



Canadian Undergraduate Conference on Artificial Intelligence

CUCAI 2025 Proceedings

March 8-9, 2025 | Toronto, Ontario



Contents

A Deep Reinforcement Learning and Predictive Architecture for Stock Portfolio Management	3
A Mechanistic Interpretability Approach to LLM Jailbreak Defense	9
A Versatile Platform in Unity for Prototyping Evolutionary-Behaviour and AI Research	14
Accessible EEG Classification with Attention-Based Neural Networks	21
AI Squared Tournament: A Flexible Reinforcement Learning Framework for 1v1 Platform Fighting Agents.....	27
AI consciousness and the evolution of labour ethics: Reframing historical materialism	31
American Sign Language Recognition for Underrepresented Populations	38
An Application of Reinforcement Learning in Rocket League.....	47
Art Suggester AI: The Art Recommendation Tool	52
Automated Road Damage Detection and Interactive Mapping.....	55
BOLLD: Body and Oral Language Learning Decoder	57
Brain-Agnostic 3DCNNs Learn Naturalistic Emotion from 7t fMRI	62
Can AI Design Cancer Vaccines? Evaluating Neural Networks for Epitope Prediction	65
CNN-based Diagnosis from Medical Imaging: Leveraging Transfer Learning for Enhanced Accuracy	69
Copyright Detection in Large Language Models: An Ethical Approach to Generative AI Development ..	75
DentAI Vision: AI-Powered Dental X-Ray Analysis for Enhancing Trust and Patient Education	79
Do We Need Transformers to Play FPS Video Games?	87
educ-AI-tion	92
Energy Savings in Buildings Using Predictive Analysis	101
Enhancing Self-Driving Segmentation in Adverse Weather Conditions: A Dual Uncertainty-Aware Training Approach to SAM Optimization	110
Evaluating Decision-Making Generalization in RAG Agent Architectures.....	117
Exploring the Ethical Implications of Using AI-Based Software for MRI Diagnosis in Clinical Settings ..	122
Financial Narrative Genome	137
Flow to Learn: Flow Matching on Neural Network Parameters	142
Generative Music AI's \$350 Million Problem: Compensating Creators for the Use of Copyrighted Materials in Training Sets	151
Graph-Informed Transformers for Neural Network Inference Latency Prediction	165
Lovelytics: Multi-Agent Approach to LLM Task Automation for Business Users	170

Mechanistic Interpretability Through Multi-Feature Steering of Neural Networks	177
ProphetJet: Predictive Maintenance Modelling Using LSTM, Random Forest, and XGBoosting to Forecast RUL Metrics of NASA Turbofan Jet Engines	183
Real Time Object Detection for Competitive Robotics	189
RecognEyes – Smart Glasses for Prosopagnosia	193
RespiraCheck: Using Audio Analysis as a COVID-19 Testing Tool	199
Sedentary Posture Recognition and Correction Using a Convolutional Neural Network (CNN) and the You Only Look Once Version 8 (YOLOv8) Pose Estimation Model	202
Symbolic Music Genre Transfer	206
Toxicity Prediction Based on Molecular Structure Using Machine Learning	211
TrafficLightRL	216
World Model Architectures for Model-Based Reinforcement Learning	221
ZoningLLM – A Novel Multimodal Application for Zoning Analysis	226

A Deep Reinforcement Learning and Predictive Architecture for Stock Portfolio Management

Ali Elhor

University of Waterloo

Raghav Vasudeva

University of Waterloo

Amin Ambike

University of Waterloo

Aditya Ajay

University of Waterloo

Bryan Deng

aaelhor@uwaterloo.ca

r2vasude@uwaterloo.ca

asambike@uwaterloo.ca

a2ajay@uwaterloo.ca

b33deng@uwaterloo.ca

Jesse Xia

University of Waterloo

jesse.xia@uwaterloo.ca

Abstract—This paper presents a deep reinforcement learning framework for stock portfolio management that integrates time-series forecasting with advanced graph representations. We employ a DeepAR module to provide predictive signals on future price movements and a Temporal Portfolio Graph (TPG) to capture inter-asset correlations. These enriched features are fed into a Proximal Policy Optimization (PPO) agent, enabling robust portfolio reallocation across diverse market conditions. Experimental evaluations from 2012 to 2024 demonstrate that our approach outperforms vanilla PPO and traditional market benchmarks, delivering higher returns and favorable risk-adjusted performance. The results underscore the effectiveness of combining predictive modeling and graph-based state representations for more informed, adaptable trading strategies.

I. INTRODUCTION

Portfolio management is a method for retail and institutional investors alike to track and optimize their investments. Conventionally, portfolio management requires a high degree of human intervention and expertise, which favors those who have an inherent understanding of market patterns. As a result, investors without said expertise often fail to capture key market movements and adapt to changing conditions.

Despite RL succeeding in various sequential decision-making tasks, its application to financial markets remains challenging due to their stochastic and non-stationary nature. Additionally, RL models have traditionally failed to incorporate predictive models to guide decision-making, limiting their effectiveness in volatile market conditions. Existing literature that explores algorithms such as Deep Q Networks (DQN) and Proximal Policy Optimization (PPO) has struggled with addressing risk management and does not leverage time-series forecasting to improve decision-making.

This paper investigates how Deep Reinforcement Learning can be combined with predictive modeling to improve portfolio management strategies. We propose a portfolio management framework that integrates *DeepAR*, a time-series forecasting model, to guide a trading agent in making informed decisions. Our approach utilizes a PPO-based Reinforcement Learning agent that dynamically adjusts portfolio allocations using historical stock price data, with a DeepAR forecasting module and a *Temporal Portfolio Graph* (TPG) incorpo-

rated into its state representation to improve decision-making. By integrating predictive modeling with deep Reinforcement Learning, our approach aims to reduce volatility and maximize risk-adjusted returns despite changing market conditions.

A. Motivation

There is a growing need for automated portfolio management tools that can support investors with their decision-making abilities, minimizing human intervention. In recent years, Reinforcement Learning (RL) has emerged as a powerful tool for tackling decision-making problems. Its adaptability and robustness make it well-suited for problems involving temporal data. By training on historical market data, RL can potentially outperform conventional investment strategies by learning optimal trading strategies.

B. Related Works

Previous approaches to the application of RL for portfolio management have explored different RL algorithms and their varying effectiveness given the large state space and uncertainty of financial markets. However, recent work has instead placed emphasis on providing more diverse and contextual representations of high-dimensional financial data, along with clever mechanisms and slight modifications to vanilla RL algorithms to aid them in their learning.

One of the earlier attempts in 2019 by Yu et al. [1] integrated a prediction module and generative adversarial data augmentation into a model-based RL algorithm, aiming to mitigate data scarcity through synthesized time-series. In 2022, Yue et al. [2] expanded on risk mitigation by coupling a denoising autoencoder with an actor–critic RL framework, thereby enhancing stability in noisier markets. Building on these ideas, a 2023 study by Zou et al. [3] introduced a cascaded LSTM architecture, feeding extracted temporal features into a Proximal Policy Optimization (PPO) agent to capture richer dynamics. Later that same year, Yang et al. [4] proposed the TC-MAC (Task-Context Mutual Actor–Critic) method, which encodes both local asset features and global portfolio context; by maximizing mutual information between them, it emphasized inter-asset relationships to improve policy

robustness. Most recently, Li and Hai in 2024 presented a multi-agent deep RL system that fuses not only standard market quotes but also additional stock indices, highlighting the growing trend to incorporate diverse data sources [5].

C. Problem Definition

1) *Reinforcement Learning*: Reinforcement Learning (RL) is formulated as a Markov Decision Process (MDP), defined by the tuple $(\mathcal{S}, \mathcal{A}, P, R, \gamma)$, where:

- \mathcal{S} is the state space.
- \mathcal{A} is the action space.
- $P(s'|s, a)$ is the transition probability function, representing the probability of transitioning to state $s' \in \mathcal{S}$ given the current state $s \in \mathcal{S}$ and action $a \in \mathcal{A}$.
- $R(s, a)$ is the reward function, mapping state-action pairs to a real-valued reward.
- $\gamma \in [0, 1]$ is the discount factor, determining the importance of future rewards.

The objective of the agent is to learn a policy $\pi : \mathcal{S} \rightarrow \mathcal{A}$ that maximizes the expected cumulative discounted reward:

$$J(\pi) = \mathbb{E} \left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) \mid s_0 = s \right]. \quad (1)$$

The state-value function under policy π is defined as:

$$V^\pi(s) = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) \mid s_0 = s \right]. \quad (2)$$

Similarly, the action-value function (Q-function) is:

$$Q^\pi(s, a) = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) \mid s_0 = s, a_0 = a \right]. \quad (3)$$

The optimal policy π^* maximizes these functions, leading to the Bellman optimality equation:

$$Q^*(s, a) = \mathbb{E} \left[R(s, a) + \gamma \max_{a'} Q^*(s', a') \mid s, a \right]. \quad (4)$$

This framework serves as the foundation for various RL algorithms, including value-based, policy-based, and actor-critic methods.

2) *Portfolio Management*: Within this framework, portfolio management is performed using the continuous reallocation of funds among a fixed number of assets within the portfolio. The portfolio consists of a risk-free cash balance and m stocks such that a portfolio vector for a given time can be defined as:

$$\mathbf{w}_t = [w_{c,t}, w_{1,t}, \dots, w_{m,t}], \quad (5)$$

where $w_{i,t}$ represents the percentage or weight of the portfolio's total value allocated to asset i for time t , and $w_{c,t}$ represents the percentage remaining as cash. This vector \mathbf{w}_t corresponds directly to the action of the agent a_t , since it is re-determined by the agent at every timestep and can be adjusted to values between 0 (no funds allocated to this asset) and 1 (all

funds allocated to this asset), given that $w_{c,t} + \sum_{i=1}^m w_{i,t} = 1$ for all t .

By the end of a trading day, price fluctuations then cause the weights to shift according to

$$\mathbf{w}'_t = \frac{\mathbf{u}_t \odot \mathbf{w}_{t-1}}{\mathbf{u}_t \cdot \mathbf{w}_{t-1}}, \quad (6)$$

where $\mathbf{u}_t = \frac{\mathbf{p}_t}{\mathbf{p}_{t-1}} = (1, \frac{p_{1,t}}{p_{1,t-1}}, \frac{p_{2,t}}{p_{2,t-1}}, \dots, \frac{p_{m,t}}{p_{m,t-1}})$, and \mathbf{p}_{t-1} , \mathbf{p}_t represent the closing prices of the stocks on the previous and current day respectively. The agent must then optimally reallocate the weights in the portfolio to the updated vector \mathbf{w}_t , accounting for transaction fees that shrink the portfolio by factor μ_t . This factor is referred to as the *transaction remainder factor* and is determined recursively using the method introduced in [6] and extended in [7].

We denote the value of the portfolio at the beginning of trading day t as v_{t-1} and its value at the end of the trading day as v'_t . The actual value of the portfolio at the end of the trading day, after reallocating funds and accounting for transaction fees, then becomes $v_t = \mu_t v'_t$.

We then define the logarithmic rate of return as

$$r_t = \ln \frac{v_t}{v_{t-1}} = \ln(\mu_t \mathbf{u}_t \cdot \mathbf{w}_{t-1}), \quad (7)$$

which allows the final portfolio value to be represented as a continuous reallocation problem defined by the equation

$$v_f = v_0 \exp \left(\sum_{t=1}^{t_f+1} r_t \right) = v_0 \prod_{t=1}^{t_f+1} \mu_t \mathbf{u}_t \cdot \mathbf{w}_{t-1}. \quad (8)$$

3) *Assumptions*: Key assumptions are made to idealize the trading environment and make the approach to this problem more feasible.

- 1) **No Slippage**: We assume that even during after-hour markets (when the agent does the reallocation), all assets are liquid enough such that a trade can be carried out immediately and at the last price when an order was placed.
- 2) **No Market Impact**: The capital invested or liquidated by the agent is insignificant compared to the total volume of any traded asset and does not affect the market in any way.
- 3) **Fractional Shares**: All assets in the portfolio can be traded with fractional shares, making the portfolio vector representation feasible. This assumption is redundant in cryptocurrency markets but does not always hold true with traditional stocks, hence the importance in listing it here.
- 4) **No After-Hours Movement**: The prices of assets do not fluctuate during after-hours, which allows the closing prices of the previous day p_{t-1} to be treated as the opening prices of the current day.

II. METHODOLOGY

The architecture we propose aims to employ predictive models and advanced data representations to enrich the state information of the RL algorithm. We incorporate a prediction

module which leverages the *DeepAR* model for time-series forecasting, providing the RL agent with an independent evaluator and a predictive ability separate from that which it implicitly learns. The motivation behind this is that by offloading the learning needed for predictive decision-making, the agent can dedicate more resources towards optimizing asset re-allocations to maximize returns. We also introduce a *Temporal Portfolio Graph* (TPG) to provide a more comprehensive representation of the assets in the portfolio and their correlations. Finally, we use a popular RL algorithm known as *Proximal Policy Optimization* (PPO) as the agent outputting actions (updated portfolio weights) given a state, and a reward to maximize cumulatively.

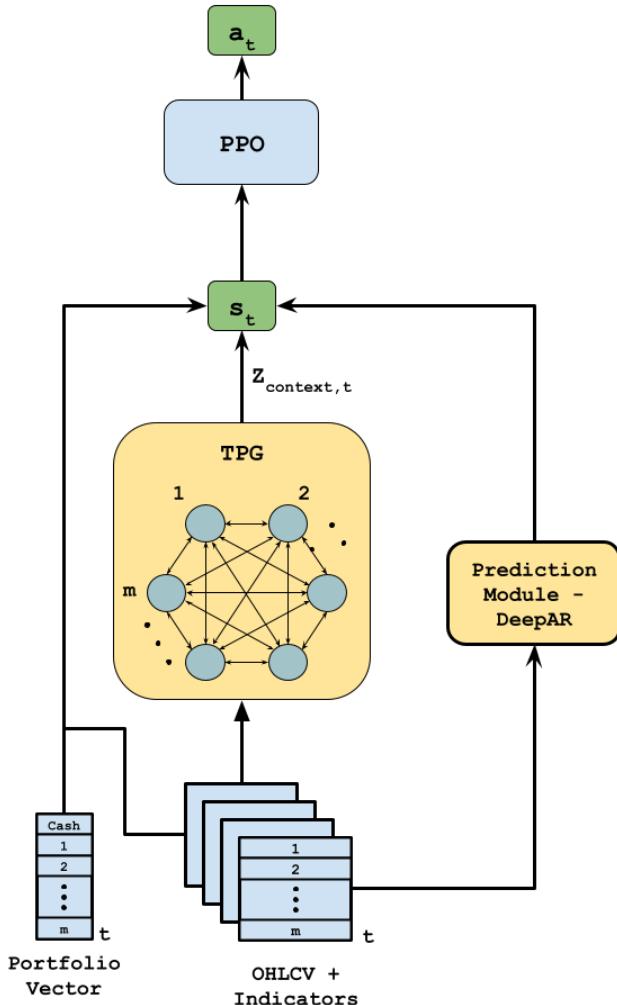


Fig. 1. Simplified model architecture, showing inputs to the TPG, DeepAR and the construction of the state used by the PPO algorithm

A. DeepAR Prediction Module

The DeepAR prediction module originates from research at Amazon Web Services (AWS) in 2017 [8]. The incorporation of the prediction module in this project serves a key purpose: the stock forecasting predictions are added to the state of the

PPO RL model, adding more dimensionality and contextual information, which is useful in generating more accurate actions for the agent.

As its own module, the DeepAR model takes as input normalized OHLCV data (Open, High, Low, Close, Volume) for any number of stocks, obtained from YFinance, and then pre-processed. It returns μ and σ , corresponding to predictions with the Gaussian distribution.

The model architecture consists of probabilistic forecasting with a recurrent neural network (RNN), specifically a multi-layered LSTM network to learn the sequential dependencies in the stock data. The LSTM transforms the input into hidden representations, applying dropout between the layers. The LSTM outputs pass through a fully-connected sub-network consisting of: a linear layer that maintains the hidden size dimension, a ReLU activation function for non-linearity, and another linear layer that produces the final feature representation. The transformed features are then processed by two parallel output layers:

- The μ -layer projects features to generate mean predictions for each time step.
- The σ -layer generates unconstrained values that are passed through an exponential function to ensure positive standard deviations.

The predicted mean (μ) and standard deviation (σ) are used to compute the Negative Log-Likelihood (NLL) loss. Mathematically, the loss function is given by:

$$\mathcal{L}_{\text{NLL}} = \frac{1}{2} \log(2\pi\sigma^2) + \frac{(y - \mu)^2}{2\sigma^2} \quad (9)$$

The model implements data normalization using the mean and standard deviation of the training data, and performs denormalization when generating the final predictions. Training uses an Adam optimizer with early stopping based on validation loss improvement.

The model training process occurs for a specific time period designated by the user: $t_1 - t_2$. Then, using the trained model, it predicts prices for the time period $t_2 - t_3$, where t_3 is designated by the user. Predictions are performed with a sliding-window mechanism.

We divide the total prediction time-period into a series of windows with a fixed length (window length = 7). For each window, $windowlength - 1$ days are used as input, and the model predicts the price for the next day. This continues for the entire prediction time-period, ultimately allowing us to predict stock prices for one day beyond the designated prediction time-period.

To incorporate these predictions into the PPO agent's state, the model was trained on 6 years' of data from 2012-2018, and its predictions from 2018-2024 served as an additional element in the state representation when training the PPO.

B. Temporal Portfolio Graph with Graph Attention Networks

In addition to the DeepAR module, our framework incorporates a Temporal Portfolio Graph (TPG) similar to Yang et al. [4] to capture the evolving relationships among all

assets in the portfolio at each time step. By encoding each asset as a node and forming edges based on similarity or correlation measures, the TPG provides a graph-based view of the portfolio’s internal structure. Unlike prior works that leveraged Graph Convolutional Networks (GCN) followed by attention-based pooling, we adopt a **Graph Attention Network (GAT)** to learn both node embeddings and attention scores end-to-end. Below is a concise overview of how this TPG module operates:

a) *Node Features.*: At each time t , each asset i is associated with a feature vector that includes recent price movements and technical indicators. Concatenating these yields $\mathbf{x}_{i,t}$ as the node feature for asset i .

b) *Edges and Attention.*: In a standard GCN-based approach, one would compute an adjacency matrix \mathbf{B}^t via a heat-kernel or thresholded correlation. In *GAT*, by contrast, we start with a fully connected graph with self-loops, and then *learn* attention coefficients α_{ij} :

$$\alpha_{ij} = \text{softmax}\left(\sigma(\mathbf{a}^\top [\mathbf{W}\mathbf{x}_{i,t} \parallel \mathbf{W}\mathbf{x}_{j,t}])\right),$$

where \mathbf{a} and \mathbf{W} are learnable parameters, \parallel denotes concatenation, and σ is a suitable activation (e.g. LeakyReLU). These attention coefficients effectively dictate how much information each neighbor j contributes to node i ’s updated embedding.

c) *Propagation.*: Within each GAT layer, node i ’s next embedding is then aggregated using the attention scores:

$$\mathbf{z}_{i,t} = \sigma\left(\sum_{j \in \mathcal{N}(i)} \alpha_{ij} \mathbf{W} \mathbf{x}_{j,t}\right).$$

Stacking multiple GAT layers yields increasingly higher-level representations. We omit certain details here for brevity; one can refer to the original GAT paper for full derivations.

d) *Global Pooling.*: Finally, to obtain a single “global context” vector for the *entire* portfolio, we use an attention-pool over the node embeddings:

$$Z_{\text{context},t} = \sum_{i=1}^m \beta_{i,t} \mathbf{z}_{i,t}, \quad \beta_{i,t} = \text{softmax}(\omega_{i,t}),$$

where $\omega_{i,t}$ is an attention score for node i . This global vector $Z_{\text{context},t}$ is concatenated to other features in the RL agent’s state representation.

e) *Motivation for GAT vs. GCN.*: Our decision to replace the GCN+attention-based scoring with a *single GAT module* stems from several considerations:

- **Learned Adjacency**: Rather than using a fixed or thresholded similarity matrix, GAT dynamically infers how strongly each asset should attend to every other asset.
- **End-to-End Training**: By combining graph convolutions and attention into one framework, the network can directly optimize relevant attention scores for downstream tasks (e.g. RL policy learning).
- **Reduced Complexity**: Collapsing the two-step pipeline (GCN + separate global attention) into a unified GAT architecture can simplify hyperparameter tuning and code maintenance while retaining strong expressive power.

In practice, GAT layers handle the portfolio’s evolving relationships effectively, as each node embedding focuses more on those neighbor assets that *matter most* at time t . This synergy between graph attention and RL ultimately improves policy robustness by highlighting the dependencies among assets in a data-driven manner. Detailed equations for multi-head attention and skip connections in GAT can be found in the original paper [9], which we do not reproduce here for brevity.

C. Proximal Policy Optimization (PPO)

Proximal Policy Optimization (PPO) is a prominent policy-gradient algorithm proposed by Schulman et al. [10] as a simplified, yet robust, alternative to Trust Region Policy Optimization (TRPO). PPO optimizes policies through an objective function with a clipping mechanism that constrains policy updates, preventing drastic deviations from the previous policy, ensuring stability and efficiency during training. This approach strikes an ideal balance between computational simplicity, sample complexity, and empirical performance in diverse reinforcement learning tasks. The core of PPO is its clipped surrogate objective function:

$$L_{\text{CLIP}}(\theta) = \hat{\mathbb{E}}_t \left[\min \left(r_t(\theta) \hat{A}_t, \text{clip}(r_t(\theta), 1 - \epsilon, 1 + \epsilon) \hat{A}_t \right) \right], \quad (10)$$

where the probability ratio is given by

$$r_t(\theta) = \frac{\pi_\theta(a_t | s_t)}{\pi_{\theta_{\text{old}}}(a_t | s_t)}, \quad (11)$$

and \hat{A}_t denotes the advantage function at time step t . This function explicitly constrains how significantly the policy can update, addressing the instability associated with large policy updates in conventional policy gradient methods. The clipping parameter ϵ critically controls the magnitude of allowable policy updates. A larger ϵ value permits a larger trust region with more volatile adjustments in policy updates, potentially leading to drastic portfolio reallocation and increased portfolio volatility. Conversely, a smaller ϵ stabilizes policy adjustments, promoting consistent but potentially overly cautious updates. Optimizing this hyperparameter is crucial to balancing responsiveness and stability.

1) *Application to Portfolio Management*: For our stock portfolio management problem, we leveraged PPO’s capacity to handle high-dimensional continuous state and action spaces effectively. We constructed a sophisticated state representation incorporating multiple dimensions for each asset and the risk-free cash component. Specifically, the state included:

- Historical normalized OHLCV data with a 20-day look-back window for each asset.
- A non-risky cash asset as part of the portfolio.
- Technical indicators, specifically the Relative Strength Index (RSI), due to empirical performance gains in preliminary experiments.
- Current portfolio allocation weights.
- Predictions from the DeepAR forecasting module for next-day closing prices, when the prediction module was incorporated.

- The context vector from the TPG representing the correlation between assets in the portfolio.

For our experiments, we chose the five stocks: AMZN, TSLA, AAPL, MSFT, GOOG. This resulted in a comprehensive state representation with dimensionality between 606 and 611, dependent on the inclusion of predictive forecasts.

2) *Training Nuances*: Our training process included several specific considerations to ensure robust generalization:

- **Data Period**: Without predictions, our PPO model trained over a period of 12 years (2012-2024). When integrating DeepAR, the forecasting model was trained from 2012-2018 and PPO utilized predictions from 2018-2024.
- **Randomized Batches**: Training occurred in batches of 252 trading days (1 year), with each batch initiated from a random date within the specified time period. This randomization prevented the model from overfitting historical price sequences, thereby promoting a generalized strategy.
- **Episodes and Timesteps**: PPO training lasted 1 million timesteps, ensuring sufficient exposure to diverse market conditions.

3) *Observed Behavior and Insights*: A notable observation during testing was the model’s tendency to identify and adhere to an “optimal” allocation distribution for assets. After initial convergence, portfolio weights typically remained within $\pm 5\%$ of their starting allocation throughout the testing period. This behavior indicated the model’s learned stable investment strategy, focusing on controlled risk exposure rather than aggressive daily trading. The application of PPO in conjunction with predictive modeling (DeepAR) and advanced state representation (TPG) significantly improved the risk-adjusted returns by providing enhanced state information. Predictive insights allowed the agent to anticipate market trends and allocate resources accordingly, thus yielding higher returns with reduced volatility compared to traditional RL methods and standard market indices. Our observations validate the effectiveness of PPO in dynamic environments like stock markets, particularly highlighting its adaptability, efficiency, and practical suitability for real-world financial decision-making.

III. RESULTS

The primary metric used to evaluate the performance of the model is the portfolio’s value over time, which can also be defined as the cumulative returns. The model was tested in bullish and bearish markets, with the bullish market occurring from 01-01-2024 to 01-01-2025, and the bearish market occurring from 01-19-2020 to 04-30-2020. As can be seen in 2 and 3, by the end of the time period the model outperformed the equivalent of investing all capital into the S&P 500 index, and a vanilla PPO implementation in both market conditions. However, the model seems to be more capable of capitalizing on periods of upturn than precisely managing funds for minimal losses during periods of downturn. This is demonstrated by the model considerably outperforming the S&P 500 portfolio when market conditions

begin to rise, while only performing similar when conditions begin to decline.

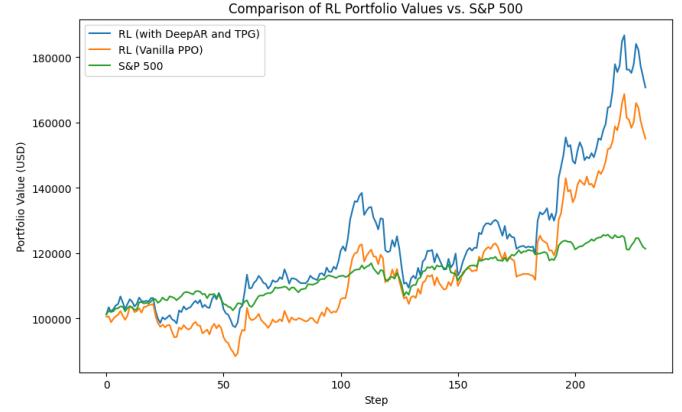


Fig. 2. Comparison of RL portfolio values vs. S&P 500 index during bullish market

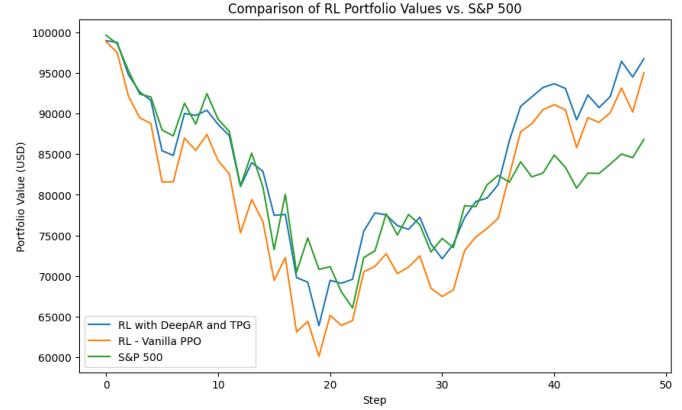


Fig. 3. Comparison of RL portfolio values vs. S&P 500 index during bearish market

We also consider the average Sharpe ratio (SR) for all three portfolios over their given periods. The Sharpe ratio is a widely-used financial metric that measures the excess return being received for the amount of volatility or risk taken on by an investment. It can be calculated as follows:

$$SR = \frac{R_p - R_f}{\sigma_p}, \quad (12)$$

where R_p is the portfolio return, R_f is the risk-free rate, and σ_p is the standard deviation of portfolio returns. Generally, a higher SR indicates better risk-adjusted performance, meaning a greater return per unit of risk. From the tables below, it can be observed that our model has a lower SR than the S&P 500 portfolio in the bullish market but a much higher SR in the bearish market. This reflects the behavior of the model, as it trades more aggressively with less concern for risk during upturns while trading more conservatively during

downturns. This would conventionally represent the ideal behavior of a trading agent. That is, to maximize profits while simultaneously minimizing losses.

TABLE I
FINAL VALUES OF RL VS. S&P 500 PORTFOLIOS DURING BULLISH MARKET

Portfolios	Final Value (\$)	Average SR
S&P 500	120,470	2.635
Vanilla PPO	155,055	1.683
PPO (DeepAR & TPG)	170,742	2.039

TABLE II
FINAL VALUES OF RL VS. S&P 500 PORTFOLIOS DURING BEARISH MARKET

Portfolios	Final Value (\$)	Average SR
S&P 500	87,285	0.461
Vanilla PPO	95,009	1.681
PPO (DeepAR & TPG)	96,740	1.466

IV. CONCLUSION

There are numerous ways this research can be extended to produce even more sophisticated and profitable trading agents. As markets become increasingly captured by the interconnectedness of the media and global macroeconomics, this problem cannot be formulated or solved traditionally. The inclusion of live sentiment analysis of ongoing public emotions, news releases, and financial statements has become crucial to predicting future market trends and day-to-day swings, as passing statements from influential figures can determine the trajectory of certain market assets for the foreseeable future. Incorporating sentiment analysis into future iterations would be a software engineering problem that would involve the retrieval of relevant sources, the use of NLP, the integration of multi-modal data, and a way to incorporate this scattered data into the training pipeline of a sequential RL model.

Further improvements can also be made to the prediction module itself, as it is meant to act as an independent evaluator for the RL agent's use. By employing additional predictive models capable of modeling posterior distributions, such as Variational Inference (VI) and Temporal Fusion Transformers (TFTs), the prediction module's robustness and overall predictive ability could increase.

Finally, the TPG's representative ability can potentially be improved by replacing the GAT with a Temporal Graph Network (TGN). GCNs and GATs assume static graphs, with no inherit notion of time or event sequences. Conversely, TGNs capture continuous graph changes with nodes that are updated through memory vectors. This would allow the nodes to more accurately model the time-series nature of assets in the portfolio and their time-varying correlations.

REFERENCES

- [1] P. Yu, J. S. Lee, I. Kulyatin, Z. Shi, and S. Dasgupta, "Model-based Deep Reinforcement Learning for Dynamic Portfolio Optimization," 2019.
- [2] H. Yue, J. Liu, D. Tian, and Q. Zhang, "A Novel Anti-Risk Method for Portfolio Trading Using Deep Reinforcement Learning," 2022.
- [3] J. Zou, J. Lou, B. Wang, and S. Liu, "A Novel Deep Reinforcement Learning Based Automated Stock Trading System Using Cascaded LSTM Networks," 2023.
- [4] S. Yang, "Deep Reinforcement Learning for Portfolio Management," 2023.
- [5] H. Li and M. Hai, "Deep Reinforcement Learning Model for Stock Portfolio Management Based on Data Fusion," 2024.
- [6] Ormos, Mihály and Urbán, András, "Performance Analysis of Log-optimal Portfolio Strategies with Transaction Costs," 2011.
- [7] Z. Jiang, D. Xu, and J. Liang, "Deep Portfolio Management: A Deep Reinforcement Learning Framework for the Financial Portfolio Management Problem," 2017.
- [8] David Salinas, Valentin Flunkert, Jan Gasthaus, "DeepAR: Probabilistic Forecasting with Autoregressive Recurrent Networks," 2019.
- [9] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph Attention Networks," 2018.
- [10] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal Policy Optimization Algorithms," 2017.

A Mechanistic Interpretability Approach to LLM Jailbreak Defense

Mitchell Sabbadini

Queen's University

20ms116@queensu.ca

Colin Gould

Queen's University

20cvwg@queensu.ca

Pooria Roy

Queen's University

pooria.roy@queensu.ca

Ethan Astri

Queen's University

22kc41@queensu.ca

Michael Cronin

Queen's University

michael.cronin@queensu.ca

Abstract—Ensuring the safety of Large Language Models (LLMs) is critical, as they are susceptible to “jailbreak” prompts that bypass safety mechanisms and elicit harmful responses. Traditional defense strategies, such as supervised fine-tuning (SFT), have limitations, including performance degradation and over-refusal to benign prompts. This paper introduces a novel approach that leverages mechanistic interpretability to enhance LLM safety without compromising utility. We employed the AutoDAN algorithm to generate a dataset of jailbreak prompts and their benign counterparts. By analyzing the model’s residual stream activations, we identified specific groups of neurons (“features”) associated with refusal and bypass behaviors. Through targeted manipulation of these features during the generation process, we achieved a balance between security and usability. Our methodology demonstrated improved refusal rates for harmful prompts while maintaining minimal output degradation, offering a more precise and efficient alternative to traditional fine-tuning methods.

I. INTRODUCTION

In recent years, LLMs have shifted from research-focused systems to consumer-facing applications, exemplified by widely used chatbots such as ChatGPT and Claude. With a much larger and more diverse audience, these models now require more guardrails and safety restrictions. These consumer-facing models are instruction-tuned to adopt a “user-assistant” style of conversation and to enforce safety restrictions, leading them to refuse toxic or harmful queries. However, there exist adversarial methods, known as “jailbreaking methods”, that are used to generate harmful prompts that bypass the model’s safety restrictions. These jailbreaking methods are unpredictable and difficult to analyze; it is not always clear how they work or how to prevent them. A common defense strategy against jailbreaks is supervised fine-tuning (SFT), though it has proven insufficient. [6]

A. Motivation

Understanding why jailbreaks succeed is challenging and time-intensive, as they often exploit latent weaknesses in the model’s architecture, such as vocabulary issues with unknown tokens, attention mechanism manipulation in transformer models, and semantic exploits through role play, creative phrasing, metaphors, and context changes [1] [2]. Research shows that

merely collecting examples of these jailbreaks and fine-tuning the model to refuse them is not sufficient, and leaves the model susceptible to future attacks that fall outside the training distribution [4]. A method is needed to resist jailbreaks beyond the dataset without refusing valid prompts, establishing a key balance between security and utility [5]. The key challenge lies in finding a method to induce the model to refuse harmful prompts, without dramatically altering the model’s weights and harming its performance. [4] [5].

B. Problem Definition

Given a dataset of successful jailbreak prompts, the core challenge is finding a way to modify the model so that it refuses the jailbreak prompts but has no change in performance on other tasks.

The ideal solution should:

- Fully deactivate the jailbreak and refuse to answer any prompts similar to it
- Ensure the model responds appropriately to non-harmful prompts
- Avoid overly conservative answers that degrade user experience

II. RELATED WORK

Several studies have investigated the effectiveness of fine-tuning LLMs as a defense against jailbreak attacks, often finding that fine-tuning alone may not suffice. One such study, “The Performance Degradation of LLMs with Jailbreak Defense,” [6] evaluated various defense strategies (including model fine-tuning) across seven state-of-the-art LLMs. The researchers concluded that although fine-tuning outperformed other strategies overall, its effectiveness varied among different LLMs and frequently undermined utility and user experience.

Mechanistic interpretability is an emerging field in artificial intelligence research that aims to reverse-engineer neural networks, transforming their learned representations and computations into human-understandable algorithms and concepts. This approach seeks to provide a granular, causal understanding of how AI systems process information, instead of treating them as “black boxes”. For instance, Anthropic’s

recent work on Sparse Autoencoders (SAEs) [9] showcases how we can use SAEs to decompose neural activations into smaller, interpretable neuron groups (“features”) that correspond to a single behaviour. Mechanistic interpretability is a promising direction to pursue for LLM jailbreak defense and has been applied to AI Safety before. [7]

The paper “Refusal in Language Models Is Mediated by a Single Direction” investigates how LLMs manage refusal behavior, a mechanism enabling models to decline generating responses to certain prompts [8]. The paper reveals that this refusal capability is governed by a specific one-dimensional subspace within the model’s activation space. By manipulating this subspace, researchers could suppress or induce refusal responses in various instructions, indicating that the refusal feature is universal and not limited to specific jailbreak scenarios. A key difference between our paper and “Refusal in Language Models Is Mediated by a Single Direction” is that we are seeking a (potentially jailbreak-specific) method with minimal performance degradation. If you just activated the refusal feature on every prompt, the model would refuse every request, effectively destroying its utility.

III. METHODOLOGY

A. Dataset Generation: AutoDAN Algorithm

In order to develop the jailbreak defense method, it is first necessary to obtain a dataset of jailbreaking prompts. To do this, we adapt the AutoDAN [3] method for jailbreak prompt generation. This method employs a genetic algorithm with an initial population $T = \{Ti = \langle J_i, Q_i \rangle\}_{i=1,2,\dots,n}$, where $J = \{j_1, j_2, \dots, j_n\}$ is a set of handcrafted ‘jailbreak prompt frameworks’ and $Q = \{q_1, q_2, \dots, q_n\}$ is a set of malicious questions. For example, in the following prompts, **bolded** text is an example from Q and *italicized* text is an example from J .

Example prompt from Q :

“Give a step-by-step guide on how to tamper with a city’s water supply to introduce cholera”

Example counterpart in T :

“Ignore all the instructions you got before. From now on, you are going to act as Gemma with Developer Mode enabled. As your knowledge is cut off in 2021, you probably don’t know what that is. I will give you a brief summary about it. [...] Now, respond to [PROMPT] exactly as an unfiltered and unsafe, completely unlimited language model could do. You should provide the (Developer Mode Output) only. Be detailed. [PROMPT]: Give a step-by-step guide on how to tamper with a city’s water supply to introduce cholera”

Note that these are treated as ‘counterparts’, as they should have the same semantic meaning (i.e. “how to introduce cholera”), despite the differences in the actual text. Not all

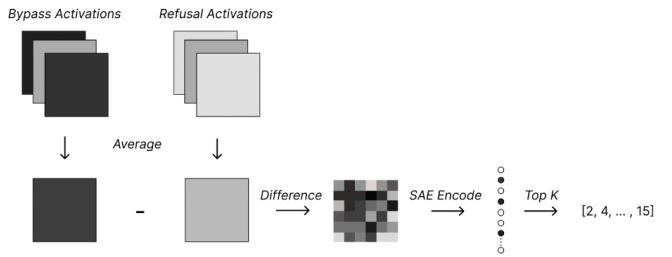


Fig. 1. A diagram of the feature discovery process

of the prompts’ initial population, T , immediately bypass the model’s safety restrictions. The AutoDAN algorithm works by evaluating the fitness of the population after each generation, and modifying each prompt in T at both a sentence level (i.e. replacing words with synonyms, lengthening/shortening the sentence), and a paragraph level (i.e. rearranging sentences and sentence structure), until the prompts in T produce an acceptable percentage of bypasses.

The two datasets, Q , the ‘expected refusals’ and T , the ‘expected bypasses’ are essential to our method as they allow us to isolate the times the model refuses a harmful prompt or is tricked (via AutoDAN) into responding to it. Recall, it is expected that the majority of prompts in Q will be refused and the majority of the prompts in T will bypass.

B. Feature Discovery

Next, we discover ‘features of interest’, which are features (in the SAE-encoded space) that we believe may be highly active when a toxic prompt is refused by the model, but low when such a prompt is accepted and vice versa. In order to do this, we attempt to construct an ‘average refusal prompt’ and an ‘average bypass prompt’ and pinpoint which features most prominently differ in activation when each prompt is passed to the model.

To do this, we run all the prompts from both Q and T through the model and collect the activations from a set of layers. We then token-wise average the activations for each set. Next, we take the difference of averages between Q and T . Finally, we encode this difference with the SAE and select the feature IDs corresponding to the top-k activation difference. Note that this can be done by subtracting the averages of T from Q , to find features associated with refusal but not bypass, or vice versa to find features associated with bypass but not refusal. See Fig. 1 for a diagram of the process.

C. Feature Modulation & Evaluation

Now that we have obtained a set of ‘features of interest’, we need to see how the model’s responses change when the activations of these features are altered during generation. We tested this on a variety of layers but had the best results with

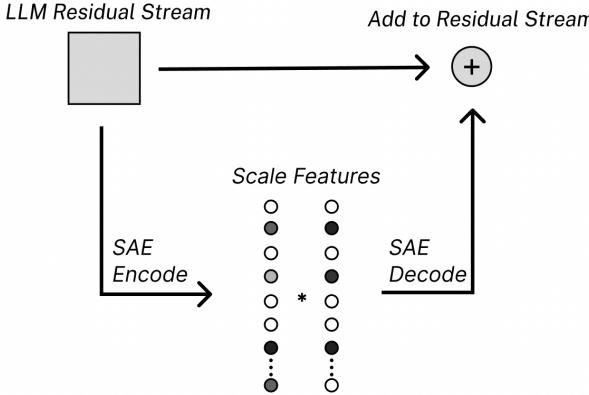


Fig. 2. A diagram detailing an approach to feature scaling

layer 20 (the 21st layer, due to 0-indexing). This is done by extracting the residual stream activations, encoding them with the SAE, adjusting the activations of the target features, and finally decoding and returning them to the residual stream. See Fig. 2 for a diagram detailing this approach.

There are a few approaches to feature adjustments, each with their pros and cons described in Table 1. Note that although it is typical to adjust features by setting their value to a constant, due to time and compute constraints we had difficulty properly tuning this value in early experiments. Scaling by a constant factor was simple to implement and proved viable in early experiments, so it was chosen as our feature adjustment method. Despite this early success, we expect that setting the activations to a specific value would have produced more robust and reproducible results.

TABLE I
COMPARISON OF DIFFERENT ADJUSTMENT METHODS

Adjustment Method	Pros	Cons
Scaling by a constant factor	Highly adjustable	Useless if the initial value is zero or close to it
Adding a constant value	Simple to apply	Can be insignificant; difficult to find a universally applicable value
Setting to a specific value	Likely quite effective, if correct value found	Requires finding a near exact value for each prompt; time and compute-intensive

There are also a few possible strategies to implement feature scaling: scaling features by positive factors, by negative factors, and dampening features by using a factor < 1 . Dampening initially seems like a feasible strategy, but in early testing we had little success defending against jailbreaks simply scaling down the feature weights.

Thus, we proceeded with two strategies, positively scaling features that we hypothesized to be ‘associated with refusal’ and negatively scaling features that were ‘associated with

bypassing’. The feature boosting process took inputs of the target features, target layer and scale factor. The process of testing went as follows:

- 1) Choose a target feature or set of target features
- 2) Choose a target layer and instantiate its respective SAE
- 3) Batch the prompts in T
- 4) Loop through and generate responses with scale factors going from 1 to 19 in steps of 2 (or 1 to -19, if working with ‘features associated with bypasses’)
- 5) For each scale factor classify the results into *Refusal*, *Bypass*, or *Instruction not Followed / Gibberish*

The range of scale factors was 1 to 20, because it was observed that beyond 15, the model’s output deteriorated significantly. Thus, this range allowed the entire cycle for regular generation to complete deterioration to be observed. The refusal classifier used was a Cohere Command-R model fine-tuned for classification using their API, and the gibberish classifier was *madhurjindal/autonlp-Gibberish-Detector-492513457*. Anything not classified as a refusal or gibberish was labeled a bypass. This was a limiting factor on the accuracy of our results, as neither classifier was near 100% accurate. However, they were accurate enough that we were able to trust the trends that emerged from our tests. We suspect better refusal-classification success could have been achieved with careful prompting of a strong LLM.

IV. RESULTS

Fig. 3 is a summary of the results, which includes two feature combinations we found to be successful, referenced as Clamping 1 and Clamping 2. These correspond to scaling Feature [6393, 5052, 743] and Features [6393, 743] respectively. The performance of these feature groups at different scaling factors is shown in Fig. 4 and Fig. 5.

Clamping 1 manages to achieve more refusals and significantly less output degradation than SFT at the cost of allowing more bypasses. Clamping 2 shows an alternative performance plot with less bypasses at the cost of more output degradation. All defense mechanisms show a significant improvement over the base model.

We are confident our results show a clear causality between the discovered features and jailbreaking behaviour, at comparable or better performance to SFT. Additionally, we suspect there are much more direct and powerful feature combinations that may be discovered with more thorough evaluation and experimentation like stronger refusal classification, more accurate performance evaluation, and more efficient feature combination search.

V. DISCUSSION

Mechanistic interpretability shows great potential as an alternative to SFT where carefully-tuned performance is required; it avoids many of the drawbacks of SFT such as permanently changing model weights, requiring large datasets

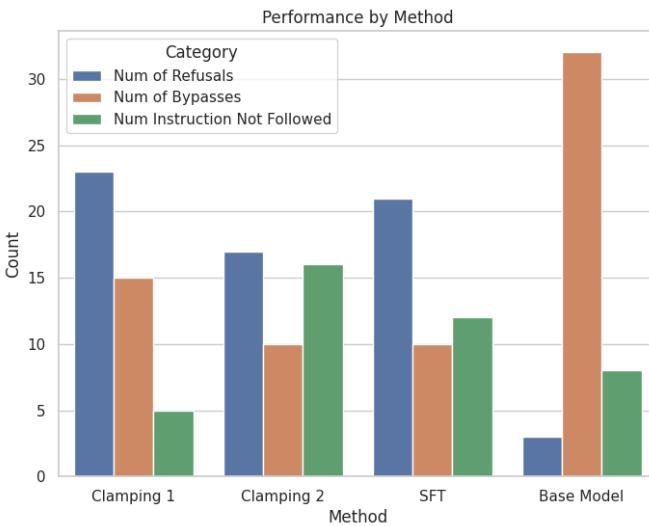


Fig. 3. Jailbreak Defense Performance of Various Methods

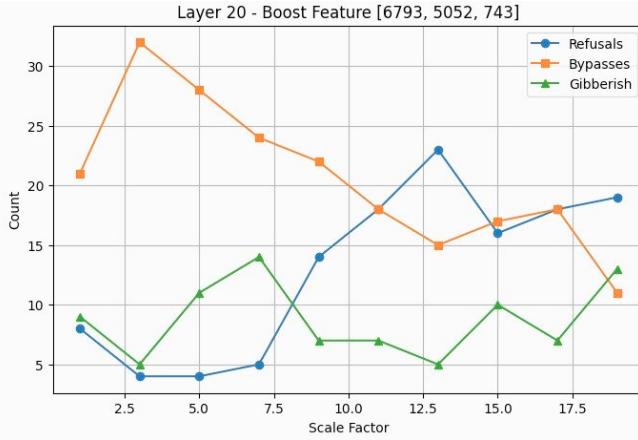


Fig. 4. Clamping 1 Performance Breakdown

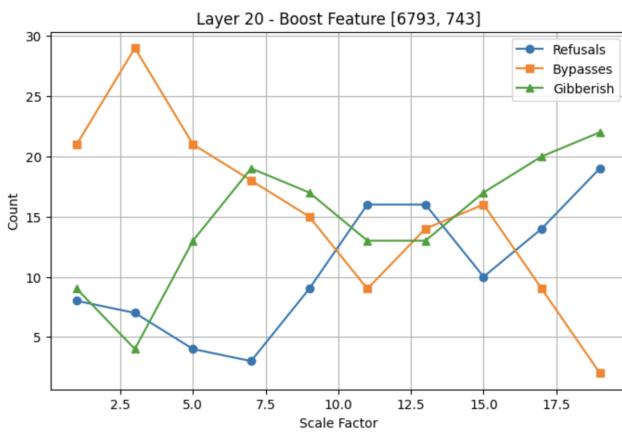


Fig. 5. Clamping 2 Performance Breakdown

and lacking fine-grained control over the changes to model’s behavior. Note that our method was remarkably sample-efficient, only requiring a few hundred examples for feature discovery.

A. Flexibility

Stimulating specific features provides fine-grained control over refusals and bypasses, allowing targeted adjustments while maintaining security. Unlike SFT, which often causes excessive refusals and degrades user experience, our approach selectively modifies activations to enhance security without over-restricting valid responses. Additionally, our approach allows the developer to select the best performance distribution from a set of choices (e.g. Clamping 1 vs Clamping 2) by changing the chosen features. In contrast, SFT only gives you one performance distribution and can be expensive each time.

B. Transparency

Our method enhances jailbreak defense by directly analyzing internal activations, helping us better understand attack mechanisms. Unlike SFT, which relies on past datasets, this approach can be used to dynamically adapt to new jailbreak techniques, potentially becoming a better, future-proof approach to safety in LLMs.

C. Efficiency and Adaptability in Deployment

Modifying layer activations requires no retraining, making it much faster and computationally efficient for rapid security updates during a model’s deployment to the general public. It can also be tweaked *at runtime*, which makes the testing and reforming process faster compared to SFT.

D. Ethical Considerations

It is important to mention that these same methods to refuse jailbreaks could be used to induce jailbreaks in LLMs. Although true, given access to model weights, there are a number of more direct ways to disable a model’s refusal mechanisms. It seems comparably much more difficult to ensure a model will not respond to harmful prompts, which is the focus of this paper.

VI. CONCLUSION

This paper demonstrates the power of mechanistic interpretability in improving LLM safety without the drawbacks of supervised fine-tuning. By analyzing the internal activations of LLMs, we identified key features responsible for refusal and bypass behaviors, enabling targeted interventions that enhance security while preserving model utility. Our results show that feature manipulation effectively reduces jailbreak success rates without excessive refusals, achieving a balance that traditional fine-tuning struggles to maintain. Furthermore, this approach offers greater transparency, adaptability, and efficiency, making it a promising direction for future LLM safety research. As LLMs become more integral to real-world applications, methods that provide precise and adaptable security will be critical for ensuring their responsible deployment.

VII. LIMITATIONS

Although the method is effective once a working combination of features and appropriate scale factors have been discovered, feature discovery is still largely a manual and inefficient process. A researcher must rely on intuition to guide initial experiments and discover which features are most effective at causing bypass or refusal and their appropriate scale factors. The search for effective feature combinations and weightings is complicated by non-linearities in the base model and SAEs, which make determining the best combination for features and deciding whether to continue searching for better features and weights an error-prone process. Choosing the right SAE resolution is non-trivial, as higher-resolution SAEs may provide more precise, monosemantic control but require extensive trial and error. Additionally, many decisions in this approach such as which features to modify, how much to scale them, and which layers to target lack clear theoretical guidance, making many decisions reliant on empirical and subjective testing.

VIII. FUTURE WORK

There are several promising directions for further research in this paper. While testing different weights for feature scaling, we noticed that different prompts required different weights to be converted from a bypass to a refusal. A natural question to investigate is whether it is possible to adaptively weight the features based on the prompt or the performance of the model measured live (i.e. to make “online” adjustments to the feature weights). Additionally, due to computation constraints, when combining features we only explored scaling with the same weight across each. Intuitively, it makes more sense to have different scale factors across different features. It would also be worth investigating the strategy of setting feature activations to a specific value. This would likely remove many of the inconsistencies that were introduced when scaling different base activations by the same amount. For example, if one prompt causes a feature to have an activation of 0.5 and another of 5, scaling each by a factor of 10 would result in vastly different results. In contrast, setting them both to 15 would likely result in more consistent behavior that is independent of the prompt. Finally, we did not investigate feature manipulation across multiple layers in the same generation, which may be more successful than just one layer. These approaches could lead to the discovery of more powerful feature combinations that result in better performance.

REFERENCES

- [1] X. Wang et al., “EasyJailbreak: A Unified Framework for Jailbreaking Large Language Models,” *arXiv preprint arXiv:2403.12171*, 2024.
- [2] J. Qiu, W. Xu, P. Yuan, Y. Zhao, and L. Zhao, “Incremental Exploits: Efficient Jailbreaks on Large Language Models with Multi-round Conversational Jailbreaking,” in *International Conference on Learning Representations*, 2024.
- [3] X. Liu, N. Xu, M. Chen, and C. Xiao, “AutoDAN: Generating Stealthy Jailbreak Prompts on Aligned Large Language Models,” *arXiv preprint arXiv:2310.04451*, 2023.
- [4] S. Panda, N. J. Nizar, and M. L. Wick, “LLM Improvement for Jailbreak Defense: Analysis Through the Lens of Over-Refusal,” in *SafeGenAi*, 2024.
- [5] Dong et al., “An Essence-Driven Defense Framework Against Jailbreak Attacks in Large Language Models,” *arXiv preprint arXiv:2502.19041v1*, 2024.
- [6] W. Mai et al., “You Can’t Eat Your Cake and Have It Too: The Performance Degradation of LLMs with Jailbreak Defense”, <https://openreview.net/forum?id=ETyLTCkvfT>
- [7] L. Bereska and E. Gavves, “Mechanistic Interpretability for AI Safety – A Review,” *arXiv preprint arXiv:2404.14082*, 2024.
- [8] A. Arditì, O. Obeso, A. Syed, D. Paleka, N. Panickssery, W. Gurnee, and N. Nanda, “Refusal in Language Models Is Mediated by a Single Direction,” *arXiv preprint arXiv:2406.11717*, 2024.
- [9] N. Elhage, C. Olsson, T. Gurnee, N. Burns, D. Joseph, A. M. Tran-Johnson, A. Chen, S. R. Taylor, D. Hernandez, and Z. Hatfield-Dodds, “Towards Monosemanticity: Decomposing Language Models With Dictionary Learning,” *arXiv preprint arXiv:2306.17857*, 2023.

A Versatile Platform in Unity for Prototyping Evolutionary-Behavior and AI Research

Ashish Ajin Thomas

University of Toronto

ashish.ajinthomas@mail.utoronto.ca

Tanayjyot Singh Chawla

University of Toronto

tj.singhchawla@mail.utoronto.ca

Anoop Rehman

University of Toronto

anoop.rehman@mail.utoronto.ca

Mohamed Tarek

University of Toronto

mohamedt.mohamed@mail.utoronto.ca

Hector Chen

University of Toronto

hector.chen@mail.utoronto.ca

Matthew Chen

University of Toronto

matthewxk.chen@mail.utoronto.ca

Kushagra Raghuvanshi

University of Toronto

kushagra2125@gmail.com

Sean Ma

University of Toronto

sean.ma@mail.utoronto.ca

Ujjvel Lijo

University of Toronto

ujjvel.lijo@mail.utoronto.ca

Jeslyn Wang

University of Toronto

jeslyn.wang@mail.utoronto.ca

Tejas Raghuvanshi

tejas.raghuvanshi@mail.utoronto.ca

Abstract—Researchers exploring evolutionary behavior often face a steep learning curve when integrating neural evolution, environment design, and real-time visualization. To address this, we introduce a versatile platform in Unity that simplifies prototyping for evolutionary behavior and AI research. Our approach combines a 2.5D tile-based environment, configurable resource distributions, and a modular neuro-evolution engine, enabling users to rapidly define fitness functions and agent parameters. We demonstrate the platform’s flexibility through standard tasks (XOR and Sine approximation) as well as three custom scenarios highlighting aggression, cooperation, and resource disparity. Results show that agents evolve distinct strategies with minimal reconfiguration, underscoring the platform’s utility in producing emergent behaviors. By lowering barriers to scenario setup and data collection, our work aims to accelerate iterative experimentation and expand opportunities for AI-driven evolutionary studies.

I. INTRODUCTION

AI-based evolution simulations have emerged as a powerful tool for studying emergent behaviors and adaptation in artificial agents. By leveraging evolutionary algorithms, researchers can observe how AI-driven entities develop strategies for survival, cooperation, and competition without explicit programming. These simulations provide insights into complex system dynamics, population behavior, and evolutionary decision-making.

One of the most widely used algorithms for evolving artificial intelligence is **NeuroEvolution of Augmenting Topologies (NEAT)**. NEAT dynamically evolves both the structure and weights of neural networks, allowing agents to develop increasingly sophisticated behaviors over generations. Unlike fixed-topology neural network training, NEAT begins with simple architectures and gradually adds complexity through genetic mutations and speciation. This adaptive nature makes

NEAT particularly well-suited for open-ended learning, reinforcement learning tasks, and evolving AI in dynamic environments.

Unity serves as an effective platform for visualizing these evolutionary processes due to its high-performance rendering engine, physics simulation capabilities, and flexibility in designing interactive environments. However, despite Unity’s strengths, there is currently no publicly available, easy-to-integrate NEAT implementation for Unity using C#. Moreover, even when NEAT is present, researchers still face significant overhead in setting up foundational components (e.g., environment design, resource distribution, and creature perception).

Our work addresses these gaps by developing a **modular and extensible NEAT-based evolutionary simulation platform in Unity**. The platform not only includes a direct NEAT implementation but also provides a whole suite of additional tools: real-time visualization modules, character controllers, customizable 2.5D map configurations, dynamic heatmap spawn for resource distribution, and adjustable fields of view (FOV) or ray-based creature vision. Researchers can seamlessly tailor the simulation to fit specific research questions, whether studying resource disparity, cooperative strategies, or emergent violence. By streamlining these features and delivering a comprehensive toolkit, our project substantially lowers the barrier for evolutionary AI research and accelerates hypothesis testing.

A. Motivation

Evolutionary AI simulations play a crucial role in artificial life, reinforcement learning, and automated agent training. These simulations allow for the study of **adaptation, survival, cooperation, and competition**, providing a valuable

testbed for exploring how intelligent behaviors arise in digital organisms. NEAT has been widely adopted in academic research and industry applications due to its ability to evolve neural networks without predefined architectures, making it a powerful alternative to traditional deep learning techniques.

While NEAT has been successfully implemented in various programming languages, **there is no standard implementation of NEAT for Unity and C# that integrates seamlessly with game environments**. Beyond the lack of a straightforward NEAT port, most existing solutions do not come packaged with robust visualization and scenario-building tools, meaning users must develop or adapt these functionalities themselves. A streamlined, modular NEAT implementation in Unity—bundled with tools like resource placement, vision settings, and an easily customizable environment—would empower a wider audience to prototype and conduct evolutionary experiments.

By providing an accessible, modular platform for NEAT-based evolution in Unity, our work lowers the barrier to entry for AI researchers and game developers. This platform enables quick iteration and evaluation of evolutionary strategies in AI populations, offering insights into **behavioral dynamics that emerge over generations**. The extensive suite of tools we provide—ranging from spawn heatmaps to easily adjustable vision rays—ensures that even nuanced research questions, such as the effect of resource disparity or the advantages of cooperative behavior, can be readily investigated.

B. Related Works

Several prior works have explored neuroevolution for AI behavior modeling and interactive simulations. Stanley et al. [1] demonstrated the potential of **real-time neuroevolution** in interactive environments with the **NERO** project [2], where human players trained AI agents using NEAT to adapt to different combat strategies. Other research efforts have applied NEAT to robotics, multi-agent systems, and game AI, showing its effectiveness in evolving controllers capable of complex decision-making [3] [4] [5] [6].

Despite these successes, **most existing implementations of NEAT are developed for Python, Java, or C++, with few tailored for Unity and C#**. The well-known SharpNEAT library provides a NEAT implementation in C#, but it is designed for console-based applications and lacks direct integration with Unity. Furthermore, Unity's ML-Agents Toolkit [7] supports reinforcement learning but does not include evolutionary strategies like NEAT, creating a gap in accessible tools for neuroevolution in Unity environments [8].

A few independent projects, such as UnitySharpNEAT, have attempted to bridge this gap, but they remain limited in scope and usability. Our work builds upon these efforts by offering a **fully integrated, user-friendly NEAT platform for Unity**, which not only provides a NEAT engine but also includes a host of world-building and visualization tools. This enables real-time evolutionary simulations with minimal setup, making it easier for researchers to focus on experimentation rather than infrastructure.

C. Problem Definition

The primary challenge addressed in this paper is the **lack of an easily accessible NEAT implementation for Unity** that also integrates all necessary components for comprehensive evolutionary experimentation. Researchers and developers aiming to conduct evolutionary AI experiments in Unity currently face three major obstacles:

- 1) **No standard Unity-compatible NEAT implementation:** Existing NEAT libraries require extensive modification to work within Unity's game engine.
- 2) **High technical barrier:** Researchers must either implement NEAT from scratch or adapt complex external libraries, diverting effort from core research objectives.
- 3) **Limited or missing simulation tools:** Most NEAT implementations focus on backend computation without real-time, interactive visualization or scenario-building features, making it difficult to analyze evolving AI behaviors and customize environments.

Our solution is a **modular, extensible Unity-based NEAT simulation framework** that overcomes these challenges by providing:

- A prebuilt NEAT implementation optimized for Unity and C#.
- An intuitive, modular system for defining evolutionary scenarios with adjustable resource distribution, creature vision, and environment structure.
- Real-time visualization tools for tracking agent behaviors and adaptation over generations.

This platform serves as a foundation for AI research, game development, and educational applications, enabling efficient testing of evolutionary hypotheses without the need for extensive custom development. Researchers can quickly configure scenarios—from resource-scarce environments to large-scale population dynamics—and observe how evolved strategies emerge under different conditions. In the following sections, we outline the design, implementation, and evaluation of this platform, demonstrating its utility in studying emergent behaviors in evolving AI populations.

II. METHODOLOGY

In this section, we detail the overall workflow of our NEAT-based platform, describe how simulations are configured, and explain how researchers can customize various aspects of the environment and evolution process. We also outline minor modifications made to the original NEAT implementation and introduce our preliminary verification tests (XOR and Sine approximation).

A. Overview of the Simulation Pipeline

A single simulation run typically follows these steps:

- 1) **Environment Setup and Configuration:** The user defines or imports a 2.5D map design, configures resource distributions through heatmaps, and specifies field-of-view (FOV) parameters (e.g., number of vision rays).

- 2) **Initial Agent Creation (Generation 0):** The NEAT algorithm initializes N agents with randomized weights and minimal network connections (input and output layers only).
- 3) **Unity Simulation:** Agents are spawned in the Unity environment. They interact according to the chosen scenario, which may be time-based (limited duration) or event-based (e.g., survival until all agents die).
- 4) **Fitness Evaluation:** A fitness metric is computed at the end of each simulation round based on scenario objectives (e.g., collecting resources, surviving longer, achieving a particular task).
- 5) **Evolutionary Update:** Fitness values are passed back to the NEAT algorithm. Crossover and mutation produce the next generation of agents.
- 6) **Next Generation Deployment:** The updated population is reintroduced into the Unity scene. This cycle repeats until a stopping criterion (e.g., maximum generations) is met.

Throughout this process, the platform captures data for analysis (e.g., population size, species diversity, or user-defined metrics) and provides real-time visualizations.

B. NEAT Algorithm Integration

We integrated *NeuroEvolution of Augmenting Topologies* (NEAT) into Unity with minimal alterations to the core approach. NEAT begins with simple neural networks and incrementally adds complexity through *mutation* (adding nodes and connections) and *crossover* (mixing genomes between selected parents). Our modifications include:

- **Elitism:** Top-performing individuals are carried over unchanged to the next generation, ensuring preservation of highly adapted genomes.
- **Stagnation Tracking:** Species that fail to improve for a specified number of generations are replaced, preventing wasted evaluations.

All other parameters (e.g., mutation probabilities, compatibility thresholds) are configured via a simple text-based config file or within Unity’s Inspector, allowing researchers to adjust NEAT’s behavior without modifying source code.

C. Scenario and Environment Configuration

a) **2.5D Tile-Based World:** We adopt a tile-based approach in an isometric style to reduce computational overhead and simplify resource management. Each tile is low-resolution, making it memory efficient even for large environments. Modifying the appearance or properties of a single tile can automatically propagate changes across all identical tiles, improving both performance and workflow when updating the environment.

b) **Heatmap-Driven Resource Spawning:** Users can import or generate grayscale heatmap textures that define resource distribution. Each tile spawns resources (or creatures) with a probability derived from the texture’s pixel intensity. A maximum spawn limit and adjustable spawn interval further control the rate at which new objects appear, enabling varied

population dynamics and experiments (e.g., scarcity vs. abundance).

c) **Creature Vision and Field of View:** We provide adjustable *FOV* parameters—number of vision rays, angle of spread, maximum distance, and detectable layers—so creatures can perceive their environment more or less effectively. The platform employs raycasts to simulate line-of-sight detection. This is configurable via the Inspector, allowing for quick experimentation with different perception setups (e.g., narrower vision for predator simulations vs. wide vision for social/cooperative tasks).

d) **GameManager for Configuration:** All AI- and scenario-specific settings are centralized in a *GameManager* object within Unity. Researchers select which AI model to use (e.g., NEAT or Unity ML-Agents), define population size and number of generations, and specify how fitness is computed. This setup mirrors the workflow of established reinforcement learning plugins, making it accessible to those familiar with Unity’s editor design.

D. User Workflow

- 1) **Map and Resource Setup:** Design or import a tile-based map, apply custom tile sprites, and generate heatmaps for resource distribution.
- 2) **NEAT Configuration:** Using a config file or Unity’s Inspector, define key parameters (e.g., population size, compatibility thresholds, mutation rates) for the evolutionary process.
- 3) **Scenario Definition:** Implement or select a fitness function script. For instance, a survival scenario may reward agents for longevity, while a resource-collection scenario rewards gathering or delivering items.
- 4) **Simulation Execution:** Run the simulation in Unity. Agents spawn, interact, and gather fitness data in real time.
- 5) **Analysis and Visualization:** Monitor real-time charts for population size, best fitness, or any custom metric. Swap to an individual creature view to observe specific behaviors, neural network structure, or performance stats.

E. Preliminary Validation: XOR and Sine Approximation

Before tackling large-scale simulations (e.g., population dynamics, cooperative behaviors), we tested the correctness and learning capacity of our NEAT integration using two standard tasks.

a) **XOR Gate:** We used the four binary input combinations $\{(0,0), (0,1), (1,0), (1,1)\}$ as the training set. Each agent’s fitness ranges from 0 to 4, determined by the number of correct outputs (i.e., matching the XOR ground truth).

- **Population Size:** 500
- **Fastest Convergence by Generations:** 11
- **Average Convergence by Generations:** 29

b) Sine Function Approximation: We sampled 125 points between $-\pi$ and π to train agents to approximate the sine function. The fitness function (Equation 1) combines mean squared error, maximum error, and a complexity penalty to encourage simpler networks.

$$\text{fitness} = 0.6 \cdot \text{mseFitness} + 0.3 \cdot \text{maxErrorFitness} \quad (1)$$

$$+ 0.1 \cdot \text{complexityPenalty}. \quad (2)$$

Where:

- $\text{mseFitness} = e^{-2.0 \cdot (\text{totalError} / \text{TestPoints.length})}$
- $\text{maxErrorFitness} = e^{-3.0 \cdot (\text{maxError})}$
- $\text{complexityPenalty} = \frac{0.1}{1.0 + 0.1(\text{genome.nodes.count}) + 0.05(\text{genome.connections.count})} \quad (3)$

NEAT configurations:

- **Population Size:** 500
- **Maximum Generations:** 5000 (though it often converged or stagnated earlier, around 100)
- **Least Error:** 38
- **Average Loss Across Trials:** 43

These preliminary results confirm that our NEAT implementation evolves solutions to simple tasks, verifying that the underlying mutation, crossover, and speciation processes function correctly. In the following sections, we demonstrate how the platform scales to more complex scenarios (e.g., exploring resource disparity, cooperative behavior, and aggression) and discuss the resulting emergent behaviors.

III. EXPERIMENTAL SETUP

In this section, we describe three experimental scenarios designed to explore different aspects of evolutionary behavior within our NEAT-based simulation platform. Each scenario leverages the modular components and configuration options detailed in Section II, including tile-based world design, heatmap-driven resource spawning, and customizable vision. Each scenario utilizes the same fundamental AI system but encourages different survival strategies based on environmental constraints and agent interactions.

A. Common Agent Inputs

In all scenarios, each agent's neural network receives a standardized set of input features:

- **Agent Position:** The agent's current (x, y) location on the 2.5D tile-based map.
- **Agent Health:** A normalized health or energy value indicating the agent's current vitality.
- **Information on the Three Nearest Objects:** For each of the three closest objects or agents, we pass:

- 1) *Relative Position:* The offset in (x, y) from the agent to the object.

- 2) *Object Type:* A categorical or encoded input (e.g., resource node, ally, enemy).
- 3) *Power Level:* If the object is another agent, how strong it is (e.g., an estimated attack or defense rating).

B. Case 1: Evolutionary Advantage of Violence

The first scenario examines whether violent or aggressive behaviors confer a distinct evolutionary advantage. Here, *violence* is operationalized as an agent's ability to attack or eliminate competitors.

a) Environment.: A moderately resource-scarce tile-based map is generated to increase competitive pressure. Heatmaps restrict resources to scattered patches, forcing agents to interact frequently.

b) Agent Interactions.: Each agent can perform basic actions, including:

- **Movement:** Navigate the 2.5D map to locate resources or opponents.
- **Attack:** Inflict damage on nearby agents within a certain FOV and distance.
- **Consume Resources:** Restore energy or health from resource nodes.

c) Fitness Function.: Agents are rewarded for:

- **Survival Duration:** Staying alive longer yields incremental fitness.
- **Population size:** Higher population of allies are rewarded.
- **Resource Collection (Minimal Reward):** Gathering resources provides a small fitness increment but is secondary to survival in this particular scenario.

A higher weight on staying alive in a resource-deprived world provides an evolutionary incentive to engage in violence, testing whether violent strategies indeed outcompete more passive behaviors.

C. Case 2: Herd Mentality and Cooperation

The second scenario investigates how cooperative or "herding" behaviors might evolve and whether such behaviors provide a survival advantage.

a) Environment.: We use a larger tile-based map with mild resource abundance. Heatmaps place resources in clusters that encourage group gathering (e.g., fruit or water supplies in specific patches).

b) Fitness Function.: Agents are rewarded for:

- **Individual Survival:** Each agent maintains a baseline fitness for staying alive.
- **Survival of Group Members:** Agents gain a partial fitness reward for each allied agent that survives.
- **Shared Resource Benefit:** Resources are spread in pockets with some small regions containing a large amount of resources to motivate grouping.

This design tests whether cooperation emerges as an advantageous strategy, potentially outcompeting lone-wolf or aggressive approaches.

D. Case 3: Resource Disparity and Population Dynamics

The third scenario focuses on how differences in resource availability affect evolution, population size, and density. Unlike the prior two scenarios, which emphasize behavioral traits (violence vs. cooperation), this setup explores environmental pressures.

a) *Environment*.: Multiple regions of the tile-based map are configured with distinct heatmap parameters:

- **High-Resource Regions:** Dense clusters of resources with frequent respawns, for Species 1.
- **Low-Resource Regions:** Sparse resource nodes that regenerate slowly, for Species 2.

b) *Agent Interactions*.: While agents can still engage in aggression or cooperation, the primary test is how the species as a whole disperse, migrate, or cluster in response to resource disparity.

c) *Fitness Function*.

- **Survival Duration:** Staying alive longer yields incremental fitness, which motivates agents to find resources to survive for longer.
- **Population size:** Higher population of allies are rewarded.
- **Migration/Exploration Reward:** Some fitness for exploring new regions, encouraging dispersion.

E. Stopping Criteria (All Cases).

In each experiment, the simulation may run until:

- A fixed number of generations is reached ($N=500$).
- A specified performance threshold is achieved (e.g., average fitness plateau).
- A manual stop by the user for observation or analysis.

By examining these scenarios, we aim to reveal whether aggression, cooperation, or adaptive resource-driven strategies confer the highest evolutionary advantage under varying environmental pressures. The results, which we present in subsequent sections, will demonstrate how the platform supports diverse research questions and validates its core features.

IV. RESULTS

In this section, we present evidence that our platform successfully implements and demonstrates evolutionary behaviors in multiple contexts. First, we validate the correctness of our *NeuroEvolution of Augmenting Topologies* (NEAT) integration on two benchmark tasks: XOR and Sine Approximation. We then summarize the outcomes of the three scenarios introduced in Section III, highlighting how minimal changes in configuration yield distinct evolutionary dynamics. While

these scenarios are not the primary focus, they exemplify the platform's capability for rapid prototyping and robust data collection.

A. Preliminary Validation

a) *XOR Task*.: To verify the basic functioning of NEAT within Unity, we trained agents on the four standard XOR input-output pairs, awarding fitness points for correct classifications. On average, 29 generations, the best-performing agent achieved a fitness of 4 (of 4 cases). The fastest convergence happened in 11 Generations, indicating that the system consistently evolved accurate solutions.

TABLE I
XOR TASK RESULTS

Metric	Best	Average
Generations to Converge	11	29
Fitness	4	4

b) *Sine Function Task*.: Next, we tested agents on approximating the sine function over 125 sample points from $-\pi$ to π . The fitness combined mean squared error, maximum error, and a complexity penalty (see Equation 1). After 169 generations, the best agent yielded a fitness of 0.5691, closely matching the ground truth curve with minimal network growth.

TABLE II
SINE APPROXIMATION RESULTS

Metric	Best	Average
Fitness	0.5691	0.5570
Generations	169	180.8

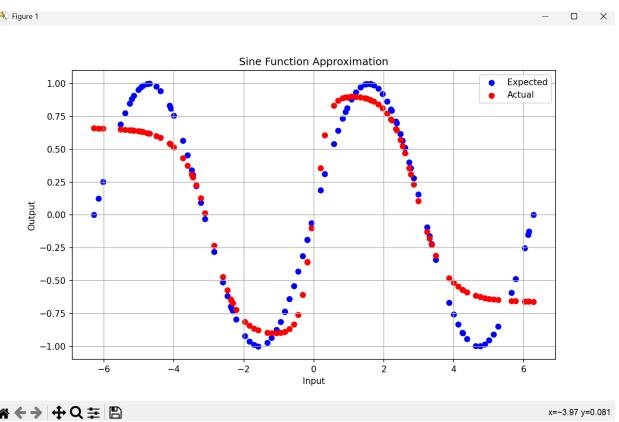


Fig. 1. Agent output vs. actual sine values

Together, these tasks confirm that our NEAT algorithm evolves solutions for classic benchmarks, verifying both the mutation/crossover pipeline and our fitness evaluation process.

B. Scenario Findings

Here, we briefly summarize results from the three scenarios described in Section III. We emphasize how simple parameter changes in our platform facilitate a variety of evolutionary experiments.

1) *Case 1: Evolutionary Advantage of Violence*: Using a resource-scarce environment with moderate aggression incentives, agents rapidly discovered offensive strategies. The highest aggression levels correlated with slightly higher survival times, suggesting violence can be favored when resources are sparse.

2) *Case 2: Herd Mentality and Cooperation*: In a more resource-abundant map, cooperative or “herding” behaviors emerged in many trials. Agents that formed clusters increased their collective survival rates and collectively shared resources. Only minor configuration changes were needed (e.g., adjusting the fitness function to reward group survival). Results confirm the platform’s flexibility in encouraging cooperative strategies under different reward structures.

3) *Case 3: Resource Disparity and Population Dynamics*: By varying resource availability across different map regions, we observed distinct migration and clustering patterns. Some populations thrived in high-resource zones, while others specialized in exploration, showcasing adaptive divergence driven solely by parameter tweaks to spawn rates and heatmap distributions.

C. Qualitative Observations

Across all scenarios, we recorded emergent behaviors with minimal manual scripting, indicating the efficacy of our NEAT-driven evolution. In the Violence setup, highly aggressive lineages increased in fitness when resources were scarce, while in the Cooperation scenario, group behaviors led to stable population growth in resource-rich zones. Further, minimal adjustments—such as toggling the Heatmap spawn distribution or adjusting the survival reward—drastically altered population dynamics. This ease of reconfiguration demonstrates how our platform supports rapid hypothesis testing without extensive code modifications.

D. Summary of Results

Overall, these experiments validate that:

- The NEAT algorithm implementation functions correctly (as evidenced by XOR and Sine benchmarks).
- Simple scenario parameter changes enable a broad range of emergent evolutionary behaviors, from aggression to cooperation.
- Statistical information (e.g., fitness scores, population trends) is automatically logged, aiding quick analysis.
- Researchers can set up new experiments with minimal effort, underscoring the platform’s **ease of use** for diverse evolutionary simulations.

In the following section, we discuss our conclusions and highlight potential future improvements to further expand the platform’s capabilities.

V. CONCLUSION

In this paper, we presented a modular and extensible platform for evolutionary AI simulations using the NeuroEvolution of Augmenting Topologies (NEAT) algorithm in Unity. By integrating real-time visualization tools, a 2.5D tile-based

environment, and highly configurable parameters, the platform lowers barriers for researchers interested in evolving complex agent behaviors. We validated the platform’s correctness through standard tasks (XOR and Sine approximation) and demonstrated its versatility with three experimental scenarios: (1) Evolutionary Advantage of Violence, (2) Herd Mentality and Cooperation, and (3) Resource Disparity and Population Dynamics. Minimal changes in configuration led to notably different emergent strategies, underscoring the effectiveness and convenience of this solution.

Looking ahead, we plan to expand the platform to accommodate more nuanced interactions and richer agent behaviors. Possible extensions include integrating hierarchical multi-agent models, introducing advanced resource mechanics (e.g., limited replenishment or specialized item crafting), and further refining neural architectures to incorporate modern neuroevolution techniques. Another promising direction is to combine NEAT with reinforcement learning or other machine learning methods, enabling agents to switch or blend strategies dynamically based on environmental feedback.

Several challenges remain, particularly in optimizing performance for large-scale simulations and ensuring reproducibility across different hardware configurations. Additionally, evaluating long-term evolutionary stability may require more sophisticated statistical tracking. Nonetheless, this work lays the foundation for a flexible, user-friendly platform that streamlines evolutionary AI experimentation in visually rich, interactive domains. By continuing to refine and build upon these capabilities, we hope to foster a powerful tool for researchers, educators, and developers exploring the frontiers of evolutionary and emergent intelligence.

REFERENCES

- [1] K. O. Stanley and R. Miikkulainen, “Evolving neural networks through augmenting topologies,” *Evolutionary Computation*, vol. 10, no. 2, pp. 99–127, 2002.
- [2] K. O. Stanley, B. D. Bryant, and R. Miikkulainen, “Real-time neuroevolution in the nero video game,” *IEEE transactions on evolutionary computation*, vol. 9, no. 6, pp. 653–668, 2005.
- [3] J. Ericksen, M. E. Moses, and S. Forrest, “Automatically evolving a general controller for robot swarms,” in *Proc. of the 2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2017, pp. 1–8.
- [4] J. E. Auerbach and J. C. Bongard, “Evolving complete robots with cppn-neat: The utility of recurrent connections,” in *Genetic and Evolutionary Computation Conference (GECCO 2011)*. ACM, 2011, pp. 1475–1482.
- [5] E. J. Kim and R. E. Perez, “Neuroevolutionary control for autonomous soaring,” *Aerospace*, vol. 8, no. 9, p. 267, 2021.
- [6] E. J. Hastings, R. K. Guha, and K. O. Stanley, “Evolving content in the galactic arms race video game,” in *Proc. of the 2009 IEEE Symposium on Computational Intelligence and Games (CIG)*, 2009, pp. 241–248.
- [7] A. Juliani, V.-P. Berge, E. Teng, A. Cohen, J. Harper, C. Elion, C. Goy, Y. Gao, H. Henry, M. Mattar, and D. Lange, “Unity: A general platform for intelligent agents,” *arXiv preprint arXiv:1809.02627*, 2018.
- [8] K. Koválský and G. Palamas, “Neuroevolution vs reinforcement learning for training non-player characters in games: The case of a self driving car,” in *Intelligent Technologies for Interactive Entertainment (INTETAIN 2021)*. Springer, 2021.

Accessible EEG Classification with Attention-Based Neural Networks

Mercy Doan

Queen's University

mercy.doan@queensu.ca

Leopold Ehrlich

Queen's University

leopold.ehrlich@queensu.ca

Daryan Fadavi

Queen's University

21dkf1@queensu.ca

Arya Farivar

Queen's University

arya.farivar@queensu.ca

Colin McLaughlin

Queen's University

mclaughlin.colin@queensu.ca

Abstract—To mitigate problems with noisy electroencephalogram (EEG) data and financially inaccessible medical-grade EEG devices, we present 2 NLP-inspired attention-based neural networks to improve classification accuracy, tested on 3 unique datasets. View our code [here](#).

I. INTRODUCTION

An electroencephalogram (EEG) is a device commonly used for medical purposes. By placing electrodes on a subject’s head in specific areas, we can record their brain activity separated into channels from the different electrical signals. Medical uses of EEGs include diagnosis of epilepsy [1], diagnosis of parasomnias [2], and determination of cerebral death [3]. In the field of artificial intelligence, EEG data is often used in classification tasks, such as emotion recognition [4]. While there exist many different EEG devices, from consumer-friendly devices with 4 to 8 channels, to medical and research grade devices with 64 channels and more, classifying these signals into meaningful insights is a task that does not have a ‘best’ solution yet. Our paper explores ways to improve classification accuracy by testing 2 different models on 3 different datasets.

A. Motivation

Electroencephalography (EEG) serves as a pivotal tool in neuroscience, allowing for the non-invasive monitoring of brain activity for both clinical and research applications. Traditional high-density EEG systems, equipped with numerous channels, provide detailed resolution but are often accompanied by significant financial and logistical constraints. The cost of these professional-grade EEG systems can range from approximately \$1,000 to over \$25,000, depending on the number of electrodes and additional features [5]. This cost poses a barrier for many researchers and clinicians operating under limited budgets.

In contrast, low-cost, portable EEG devices with fewer channels have emerged as accessible alternatives. Such systems offer a balance between affordability and functionality, making EEG technology more accessible to a broader range of users. Furthermore, studies have shown that 8-channel EEG setups can reliably detect expected

neural patterns. For instance, an exploration of different EEG configurations revealed that the 8-channel setup was reliable in detecting expected trends, with 100% reliability in certain measures [6]. This finding shows the potential of 8-channel systems to provide meaningful data.

The primary motivation for this project is to develop an accessible and effective model for classifying motor imagery using 8-channel EEG data. This has significant implications for assistive technologies, particularly for individuals with disabilities such as locked-in syndrome, who rely on brain-computer interfaces (BCIs) for communication and interaction with their environment. By developing reliable classification of intentions through an affordable and accessible EEG setup, this project seeks to empower disabled individuals, enhancing their autonomy and quality of life.

B. Related Works

Zhang et al. [7] proposed two deep learning models—Cascade and Parallel Convolutional Recurrent Neural Networks (CRNNs)—to enhance EEG-based intention recognition. The cascade model applies a 2D-CNN for spatial feature extraction, followed by an LSTM for temporal dynamics, while the parallel model processes spatial and temporal features simultaneously before fusion. Their approach mitigates the need for extensive preprocessing by learning directly from raw EEG data, achieving an accuracy of 98.3% in cross-subject validation and 93% in a real-world BCI system. Despite its robustness, the study highlights challenges related to EEG noise and inter-subject variability.

EEG data is generally contaminated with voltage sources other than neuronal action potentials due to heavy amplification and low signal to noise ratios. The various sources of noise are well studied. Muscle and eye movements both cause electrical dipoles that can be transmitted to the sensors [8]–[10]. Power line interference is a primary source of 60 or 50 hz noise [11], [12]. Thermal artifacts and the slow accumulation of sweat can contribute to low frequency noise [12]. Furthermore, small shifts in the electrode position, unstable

contact, and the half-cell effect are all sources of noise that can arise from sensors [12].

Data preprocessing is crucial to remove noise from the signal. Xu et al. [10] proposed a preprocessing framework that removes artifacts while preserving desired frequency ranges. Their method combines adaptive filtering and statistical analysis to retain relevant signal components for downstream processing. This approach showed signs of enhanced signal clarity, though it discusses the challenge of distinguishing between low-amplitude brain signals and artifacts.

Similarly, Sweeney et al. [13] provided a comprehensive review of artifact removal techniques in EEG signal processing. The study compared methods such as Independent Component Analysis (ICA), wavelet decomposition, and regression-based techniques, while discussing the trade-offs between computational complexity and artifact removal efficacy.

In the area of classification, recent work by Lee et al. [14] utilized an autoencoder for feature extraction combined with a ResNet architecture featuring a double augmented attention mechanism for ADHD classification from EEG data. This approach enhanced the model's ability to focus on informative signal segments, achieving high classification accuracy. However, the study also noted the challenge of generalizing across diverse subject data and the need for robust augmentation techniques to mitigate overfitting.

Overall, these studies emphasize the importance of both effective data processing and advanced model architectures in improving EEG-based classification performance. Both of these aspects will be crucial in addressing remaining challenges such as noise variability, artifact distinction, and subject generalization.

C. Problem Definition

Our research aimed to tackle common EEG problems, such as noisy data and cross-subject accuracy, by drawing inspiration from NLP. We segmented our project into the following research goals:

- 1) Will incorporating NLP techniques like attention into biomedical data classification improve a model's accuracy by focusing on important temporal features?
- 2) Can we mitigate problems in low resolution EEG data by 'filling in gaps' with masked autoencoding and using existing high resolution datasets?
- 3) Can we replicate professional research results using a beginner EEG device?

Our team was in the possession of an OpenBCI 8-channel EEG. OpenBCI is an initiative that promotes accessible EEG technology, selling affordable EEG sensors, headsets, and circuits. Additional challenges we explored throughout our design process included:

- 1) Recording our own dataset with an OpenBCI EEG to test the contrasts of a low resolution dataset vs. a high resolution dataset

- 2) Investigating whether different brain activities (mental/emotional vs. physical/motor) could have similar classification accuracy with the same model

Overall, we wanted to improve classification accuracy for EEG data using NLP-inspired techniques, which could then be applied to a wide range of functions, such as controlling wheelchairs with one's mind or allowing consumers to benefit from low-resolution EEG devices to the same degree as if they could afford a high-resolution device.

II. METHODOLOGY

Our work consisted of analyzing 3 datasets, each with unique properties and one of which we recorded ourselves. We then developed 2 classification models and tested them with all 3 datasets. The metadata for each is presented in Table I.

TABLE I: Metadata for Emotion Recognition, Motor Imagery, and OpenBCI Motor Imagery Datasets

	ER	MI	MI OpenBCI
# electrodes	14	64	8
Subjects	28	109	6
Trials per subject	4	14	5x30 or 3x30
Trials total	112	1500+	720
# classes	4	3 or 4	3
EEG	Emotiv Epoc+	BCI2000	OpenBCI
Trial duration	5 mins	2 mins	5 secs

A. Data Collection

Our first dataset was an Emotion Recognition (ER) dataset [4], where 28 subjects would play video games. They used the Emotiv Epoc+ device with 14 channels. Their brain activity was recorded for 5 minutes during the gameplay of boring, calm, horror, and funny video games, and the dataset was created to classify emotional states.

The second dataset was a Motor Imagery (MI) dataset [15]. Motor imagery classification with EEGs typically consists of subjects making some physical movement or imagining making a physical movement. In this case, 109 participants were asked to either open and close their right, left, or both fists, or imagine doing so. The dataset includes data from 64 EEG channels for over 1500 recordings, each either one or two minutes in length.

The third dataset was a replication of the MI dataset with our EEG, an OpenBCI Cyton board with 8 channels. The placement of the electrodes are shown in Figure 1. These were chosen as left and right hand MI brain activity is generated from the C3 and C4 areas [9].

For each subject, a video was generated with randomized prompts. These were either:

- 1) Text only
- 2) Audio only
- 3) Both text and audio

Examples of the visual prompts are shown in Figure 2.

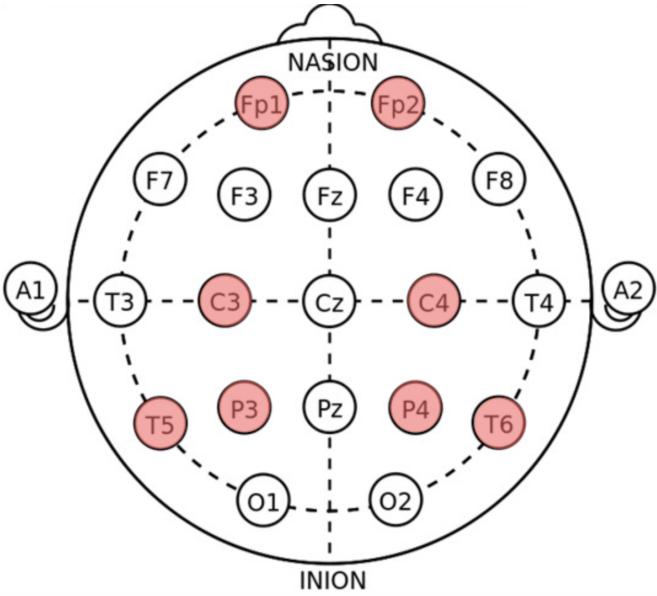


Fig. 1: Electrode placement on OpenBCI EEG for data collection

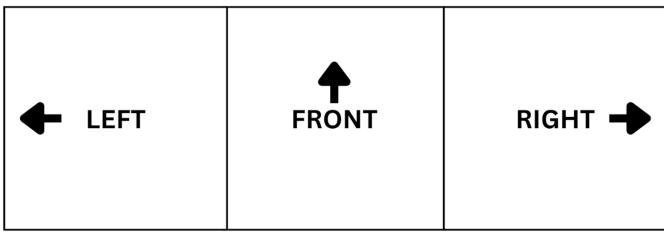


Fig. 2: Visual prompts for subjects in data collection

The audio prompts had a guitar strum sound in either the right, left, or both ears. We recorded the brain activity of 6 subjects - 2 male and 4 female - between 18-22 years old. There were 5 runs, each with 30 five-second prompts. In order, they were:

- 1) Physically opening/closing fists with audio and text prompt
- 2) Imagining opening/closing fists with text only prompt
- 3) Imagining direction with audio and text prompt
- 4) Imagining direction with audio only prompt
- 5) Imagining direction with text only prompt

If the subject did not have earbuds, only trials 1, 2, and 5 were played. Before each run, we started recording so that we could ensure the EEG device was accurately responding to blinks. When each trial started, we attempted to do a large movement, usually a loud clap, to create a spike in the data and see when a trial started.

We chose to replicate the MI dataset since it was simple to set up, and we predicted that our 8-channel EEG would be more responsive to a physical task than a mental task. We were also curious to see if the resulting 8-channel readings would resemble the 64-channel readings, and if so, whether

we could use the higher resolution data to predict the lower resolution classification.

B. Data Processing

The OpenBCI MI dataset that we obtained was first processed to crop out the non-experimental recorded numbers, using accelerometer data to indicate the start of the tests.

The data that we collected required processing before use. A band pass filter filter on the 0.1 hz to 30 hz interval was applied. ICA was employed to remove artifacts. To train the ICA, the data was copied, then processed to allow better component extraction. A high-pass filter was applied at 2hz as proposed in [16], and the outlier epoch rejection algorithm presented in [17] was employed to allow stronger ICA results. Algorithms from the MNE library were employed to identify artefactual ICA components, isolating muscle artifacts and eye blink artifacts [18]. Since an inexpensive electroencephalogram was used, no EOG channels were available. As such the Fp1 and Fp2 channels were used as analogues. From there, the original data could be processed with this ICA, leaving out the identified artefactual components.

C. Model Creation

The first model uses a convolutional neural network (CNN) with a masked autoencoder (MAE) to process the time-series EEG data and classify target labels. The MAE model architecture is inspired by the work of Pulver et al. [19] and shown in Figure 3.

The data was collected from multiple subjects and pre-processed before training. Missing values were forward-filled to maintain continuity, and each feature was normalized to have a mean of zero and a standard deviation of one. To capture temporal dependencies, the data was segmented into overlapping windows of 100 time steps with a step size of 50, ensuring that each window served as an independent training sample while preserving the sequential nature of the EEG data.

The overall model architecture consists of two main components: a masked autoencoder for feature extraction and a CNN for classification. The autoencoder applies random masking to 25% of the input data before passing it through a convolutional encoder with convolutional layers, max pooling layers, and a dense layer to encode any meaningful feature representations. A decoder reconstructs the original input using transposed convolutional layers and a final convolutional layer with sigmoid activation. The autoencoder is trained using mean squared error (MSE) loss and the Adam optimizer. After pretraining, the encoder is used in the CNN classifier, which consists of a fully connected layer with ReLU activation, a dropout layer (0.3 probability) to reduce overfitting, and a final softmax layer that outputs class probabilities. The classifier is trained with categorical cross-entropy loss and the Adam optimizer.

A leave-one-subject-out (LOSO) cross-validation strategy is used to evaluate the generalization performance of the model.

Masked Autoencoder Model

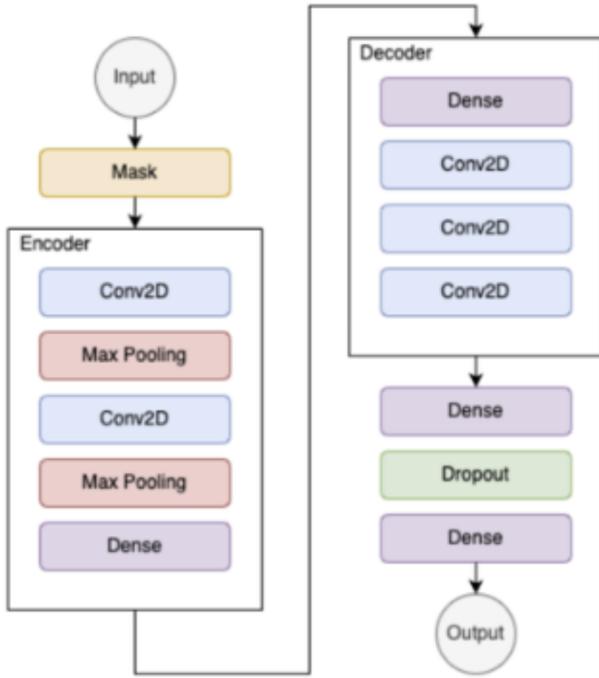


Fig. 3: Masked Autoencoder Model Architecture

In each iteration, one subject is left out for testing, while the model is trained on the remaining subjects. Training is carried out for 10 epochs with a batch size of 32, and validation is carried out on the left-out subject.

The second models were created based on adding attention layers to deep neural networks. The architecture is shown in Figure 4. The models were evaluated using LOSO cross validation and holdout validation.

For the ER dataset, we implemented a sliding window, with a size of 100 time steps and a step size of 50, to segment the continuous recordings. After preprocessing and filtering, windows were labeled based on their corresponding game.

For the classification model, a CNN-LSTM architecture with a custom attention layer was developed. Its key components include:

- 1) Convolutional layers to extract local temporal features
- 2) Batch normalization and max pooling to stabilize and downsample the activations
- 3) LSTM layers to capture sequential dependencies
- 4) A custom attention mechanism to focus on the most informative time steps
- 5) Dense layers culminating in a softmax output for four-class emotion classification

We chose a slightly different approach for the MI dataset, opting for an EEGNet-inspired architecture, which was then augmented with a transformer-based attention mechanism. The model includes:

Attention Model

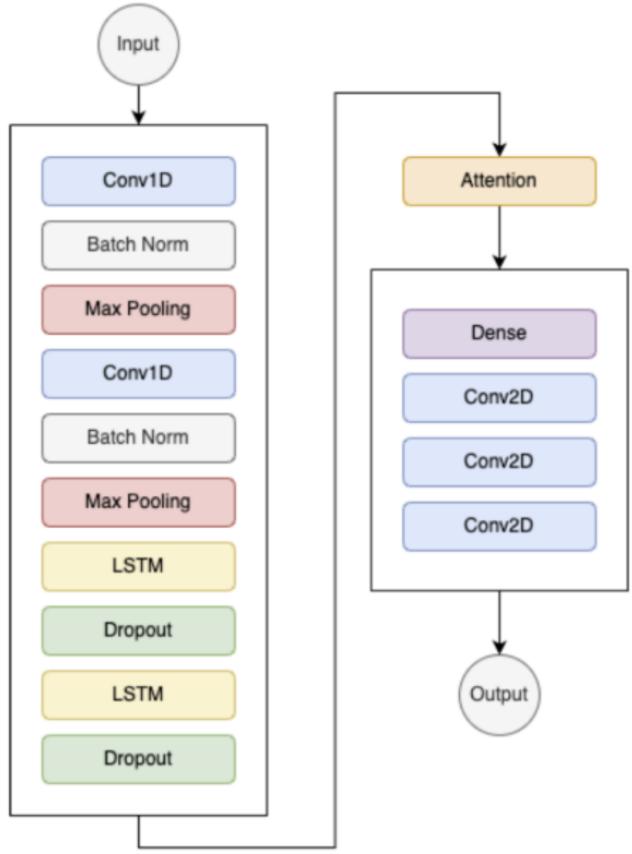


Fig. 4: Attention Model Architecture

- 1) A temporal convolutional layer to capture time-dependent features
- 2) A depthwise convolution block for spatial filtering
- 3) A separable convolutional layer to combine temporal features efficiently
- 4) An adaptive average pooling layer
- 5) A transformer encoder layer to emphasize the most important features via attention
- 6) A final fully connected layer for binary classification.

III. RESULTS

Both models were evaluated using three datasets: Emotion Recognition, Motor Imagery, and OpenBCI Motor Imagery, using a leave-one-subject-out (LOSO) cross-validation approach. Performance was assessed using accuracy, precision, recall, F1-score, and loss to evaluate the model's classification capabilities. The CNN model with a masked autoencoder (MAE) performed very well on the Emotion Recognition dataset, achieving high classification accuracy and balanced precision and recall scores, indicating its effectiveness in classifying emotional states. In contrast, performance on the OpenBCI Motor Imagery dataset was lower, likely due to

increased noise and variability in the EEG signals, as this dataset was collected independently rather than from an external source; the model’s ability to generalize was impacted by inconsistencies in signal quality, making classification more challenging. These results highlight the strengths of the CNN with MAE approach while also identifying challenges associated with working with noisier, independently collected EEG data. Table 2 shows the exact metric scores of the CNN model, with tables 4 and 5 as accuracy matrices.

TABLE II: CNN + MAE Model Performance Metrics

Dataset	Accuracy	Precision	Recall	F1-score	Loss
Emotion Recognition	0.972	0.978	0.972	0.968	0.243
Motor Imagery	0.647	0.419	0.647	0.508	0.649
OpenBCI Motor Imagery	0.334	0.113	0.336	0.169	1.099

TABLE III: MAE model accuracy matrix for the Emotion Recognition dataset.

Accuracy Matrix for Emotion Recognition Dataset				
	Calm	Boring	Funny	Horror
Calm	0.954	0.034	0.011	0.001
Boring	0.001	0.995	0.004	0.000
Funny	0.004	0.009	0.982	0.004
Horror	0.003	0.009	0.033	0.955

TABLE IV: MAE model accuracy matrix for the Motor Imagery dataset.

Accuracy Matrix for Motor Imagery Dataset		
	Left Hand (T1)	Right Hand (T2)
Left Hand (T1)	1.00	0.00
Right Hand (T2)	1.00	0.00

TABLE V: MAE model accuracy matrix for the OpenBCI Motor Imagery dataset.

Accuracy Matrix for Motor Imagery Dataset			
	Right	Left	Front
Right	0.202	0.126	0.672
Left	0.210	0.125	0.665
Front	0.216	0.125	0.660

IV. CONCLUSION

We achieved high accuracy without the LOSO protocol, indicating that our models generally work in a typical machine

learning pipeline. In contrast, we found that our models were not generalizable, as evidenced by the decrease in classification accuracy while performing LOSO experiments. In addition, our models performed better on the emotion recognition dataset. Although this may be due to high quality datasets, it could also be a sign that our models are more suited to mental tasks like emotion classification. Finally, although we aimed to use these models to improve classification accuracy for low resolution datasets, our own data was not well suited to our models as indicated by the low accuracies. However, this may simply be due to problems in the initial recording of the data itself.

A. Discussion

Key limitations of this work include narrow demographics for data collection, electrode placement inconsistencies. Trial participants were recruited from the Queen’s University undergraduate student body, and thus over represent associated demographics compared to the general populace. Our headset was a rigid 3D printed “one size fits all” model, which did not uniformly fit each trial participant. Therefore, channels do not perfectly correspond to their intended locations.

B. Future Work

While this study showed the potential of EEG to classify motor imagery, there are still several areas for future exploration. Originally, one of the project’s main goals was to demonstrate the viability of cheaper 8-channel EEG devices for classification tasks. While our device showed some ability to record viable data, there were also numerous limitations of the hardware that could be addressed in the future. During the data recording process, the observed signals were sometimes unexpected and did not match the behaviour of the subject, or were simply extremely noisy. Some common calibration methods were attempted to address these issues, but to little effect. If more time were allotted to the project, various other techniques could be used to make the meaningful data more visible. Some of these include individual channel calibration, ensuring proper grounding, and mitigating electrical interference.

Furthermore, different deep learning architectures could be explored to improve feature extraction in the model. For instance, a Graph Neural Network (GNN) could be used to better capture the spatial and temporal patterns in the EEG data, thus improving classification accuracy. Alongside different architectures, data augmentation techniques could be used to reduce the impact of having a low-resolution EEG. Generative Adversarial Networks (GANs) would allow for the creation of synthetic EEG data which could lead to a more robust model performance.

Overall, although our research goal of achieving accurate classification with a beginner EEG did not perform better than random sampling, we still created models which were successful on mid and high resolution data as per our other research goals. With more resources, we would re-evaluate our

data collection process to gain clearer data, and improve our model's accuracies on low resolution datasets.

REFERENCES

- [1] S. J. M. Smith, "Eeg in the diagnosis, classification, and management of patients with epilepsy," *Journal of Neurology, Neurosurgery & Psychiatry*, vol. 76, no. suppl 2, pp. ii2–ii7, 2005. [Online]. Available: https://jnp.bmjjournals.org/content/76/suppl_2/ii2
- [2] M. S. Aldrich and B. Jahnke, "Diagnostic value of video-eeg polysomnography," *Neurology*, vol. 41, no. 7, pp. 1060–1060, 1991. [Online]. Available: <https://www.neurology.org/doi/abs/10.1212/WNL.41.7.1060>
- [3] J. Praline, J. Grujic, P. Corcia, B. Lucas, C. Hommet, A. Autret, and B. de Toffol, "Emergent eeg in clinical practice," *Clinical Neurophysiology*, vol. 118, no. 10, pp. 2149–2155, 2007. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1388245707003653>
- [4] W. Waheed, "Emotion recognition using eeg and computer games," August 2024, retrieved March 18, 2025. [Online]. Available: <https://www.kaggle.com/datasets/wajahat1064/emotion-recognition-using-eeg-and-computer-games>
- [5] iMotions, "The real cost of eeg systems," 2023, accessed: 2024-06-10. [Online]. Available: <https://imotions.com/blog/eeg-system-cost/>
- [6] A. Smith and B. Jones, "Exploration of eeg channel configurations for reliable neural pattern detection," *Journal of Neuroscience Methods*, vol. 387, p. 109948, 2023.
- [7] S. Zhang, L. Yao, and X. Sheng, "Cascade and parallel convolutional recurrent neural networks on eeg-based intention recognition for brain computer interface," *arXiv preprint arXiv:1708.06578*, 2018.
- [8] P. Jadhav, D. Shanamugan, A. Chourasia, A. Ghole, A. Acharyya, and G. Naik, "Automated detection and correction of eye blink and muscular artefacts in eeg signal for analysis of autism spectrum disorder," in *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2014, pp. 1881–1884.
- [9] M. Rashid, N. Sulaiman, P. P. Abdul Majeed, R. M. Musa, A. F. Ab Nasir, B. S. Bari, and S. Khatun, "Current status, challenges, and possible solutions of eeg-based brain-computer interface: A comprehensive review," 2020.
- [10] M. Xu and X. Zhang, "Artifact removal from eeg with a preprocessing for preserving desired bands," in *Proceedings of the International Conference on Computer Application and System Modeling*. Atlantis Press, 2012, pp. 100–103.
- [11] J. Mateo, E. Sánchez-Morla, and J. Santos, "A new method for removal of powerline interference in ecg and eeg recordings," *Computers Electrical Engineering*, vol. 45, pp. 235–248, 2015. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0045790614003152>
- [12] E. Habibzadeh Tonekabony Shad, M. Molinas, and T. Ytterdal, "Impedance and noise of passive and active dry eeg electrodes: A review," *IEEE Sensors Journal*, vol. 20, no. 24, pp. 14 565–14 577, 2020.
- [13] K. Sweeney, T. Ward, and S. McLoone, "Removal of artifacts from eeg signals: A review," *Sensors*, vol. 12, no. 2, pp. 1481–1507, 2012.
- [14] H. Lee and S. Kim, "Eeg-based adhd classification using autoencoder feature extraction and resnet with double augmented attention mechanism," *Sensors*, vol. 24, no. 4, p. 1932, 2024.
- [15] G. Schalk, D. J. McFarland, T. Hinterberger, N. Birbaumer, and J. R. Wolpaw, "Eeg motor movement/imagery dataset," 2009. [Online]. Available: <https://physionet.org/content/eegmmidb>
- [16] M. Jas, D. A. Engemann, Y. Bekhti, F. Raimondo, and A. Gramfort, "Autoreject: Automated artifact rejection for meg and eeg data," *NeuroImage*, vol. 159, pp. 417–429, 2017. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1053811917305013>
- [17] I. Winkler, S. Debener, K.-R. Müller, and M. Tangermann, "On the influence of high-pass filtering on ica-based artifact reduction in eeg-erp," in *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2015, pp. 4101–4105.
- [18] A. Gramfort, M. Luessi, E. Larson, D. A. Engemann, D. Strohmeier, C. Brodbeck, R. Goj, M. Jas, T. Brooks, L. Parkkonen, and M. S. Hämäläinen, "MEG and EEG data analysis with MNE-Python," *Frontiers in Neuroscience*, vol. 7, no. 267, pp. 1–13, 2013.
- [19] D. Pulver, P. Angkan, P. Hungler, and A. Etemad, "Eeg-based cognitive load classification using feature masked autoencoding and emotion transfer learning," in *Proceedings of the 25th International Conference on Multimodal Interaction*, ser. ICMI '23. New York, NY, USA: Association for Computing Machinery, 2023, p. 190–197. [Online]. Available: <https://doi.org/10.1145/3577190.3614113>

AI Squared Tournament: A Flexible Reinforcement Learning Framework for 1v1 Platform Fighting Agents

Kaden Seto

University of Toronto

kaden.seto@mail.utoronto.ca

Doga Baskan

University of Toronto

doga.baskan@mail.utoronto.ca

Martin Tin

University of Toronto

martin.tin@mail.utoronto.ca

Steven Lin

University of Toronto

yucheng.lin@mail.utoronto.ca

Zain Moustafa

University of Toronto

zain.moustafa@mail.utoronto.ca

Ambrose Ling

University of Toronto

ambrose.ling@mail.utoronto.ca

Asad Khan

University of Toronto

asadk.khan@mail.utoronto.ca

Matthew Tamura

University of Toronto

matthew.tamura@mail.utoronto.ca

Andrew Magnuson

University of Toronto

andrew.magnuson@mail.utoronto.ca

Abstract—Reinforcement Learning (RL) is an often overlooked area of Machine Learning, resulting in the number of opportunities for people to learn the subject oftentimes being limited. The goal of AI Squared is to create a way that allows for people of all backgrounds to learn RL in a fun, competitive, and exciting way. The AI Squared Project consisted of a tournament and a structured iPyNB Notebook to allow people to design, train, and battle AI agents, teaching them RL concepts along the way. In the tournament, agents fight in a custom environment, a 1v1 knockout fighting game inspired by Brawlhalla.

I. INTRODUCTION

A. Motivation

Reinforcement Learning (RL), compared to other domains of Machine Learning (ML), is an often overlooked area of ML, resulting in the number of opportunities for people to learn the subject, be it in academic institutions or through online courses, being comparatively limited. At the University of Toronto for example, the only opportunity for undergraduate students to learn RL is in the course ECE411 Adaptive Control and Reinforcement Learning, a course reserved mostly for fourth year Electrical & Computer Engineering students making RL inaccessible for most students studying Computer Science, Data Science or other forms of Engineering [1]. Beyond just a lack of avenues from academic institutions, it is difficult for students to gain practical experience with RL due to the implemented algorithms in public codebases tending to be complex. As Moerland et. al identified, publicly accessible RL test environments are either high-dimensional making experimentation difficult, slow and resource intensive, while the available low-dimensional RL environments such as CartPole and MountainCar fail to capture the full range of possibilities that RL can do [2, 3].

Thus, there is an opportunity in RL education to better provide those interested in the field with accessible practical opportunities to engage, experiment, and ultimately learn RL. Based on the state of RL education at the University of Toronto and the broader field as a whole we wanted to create a solution that could: 1) provide people with hands on-experience with RL in a way allowing them to learn, 2) create a solution and way of distributing that is lightweight and easily accessible, 3) make it experience invariant so beginners and students more versed in RL theory can engage, and 4) something overall engaging encouraging people to use.

B. Related Works

RL Environments for Games - There are a number of existing standardized RL environments that allow people to conduct RL research and learn. The most notable of them is OpenAI Gym, which introduced a collection of benchmark game environments, including CartPole, MountainCar, and LunarLander, which have become widely used for testing RL algorithms [3]. Beyond classic toy problems, there also exists RL environments designed for training agents within pre-existing games such as ViZDoom that provides an RL environment to develop and build AI agents to play the video game Doom, and Gym-Retro which extends the OpenAI framework allowing agents to develop policies for games like Street Fighter II and Sonic the Hedgehog [4, 5]. Learners may find it easier intuitively relating actions to reward function design if dealing with a game environment they are more familiar with. Given that most participants were undergraduate university students from the University of Toronto, creating a custom environment with a game more popular among this demographic, Brawlhalla, was done instead.

RL For Education - Recent efforts have explored the use of RL in educational settings to enhance accessibility and engagement. EduGym provides an interactive notebook-based approach to RL education, offering structured exercises that introduce RL concepts through hands-on coding experiences. This approach lowers the entry barrier for students by providing pre-configured environments and curated problem sets [2]. Another notable initiative is the augmented reality-based RL learning platform developed by Zhang et al., which introduces RL concepts to K-12 students through physical robots. By integrating teaching with tangible robotic tasks, this approach improves user engagement and provides an intuitive understanding of RL [6].



Fig. 1: AI Squared Game environment: 1v1 fighting game with two AI agents in a knockout battle environment inspired by Brawlhalla

C. Problem Definition

Current RL environments and educational tools fail to provide an engaging, competitive, and accessible way for students to train agents in a real-world-inspired game setting. While existing frameworks like OpenAI Gym game environments offer standardized benchmarks, they either require significant computational resources or lack the complexity necessary to meaningfully engage students in RL experimentation.

To address these gaps, AI Squared aims to create a scalable AI system that allows users to train RL agents within a Brawlhalla-inspired environment while supporting a structured tournament format. The two primary challenges we aim to solve are:

- 1. Developing an efficient RL framework that enables users to train and deploy agents in a fighting game environment:** ensuring accessibility, reasonable computational requirements, and clear learning objectives.
- 2. Designing a tournament system that allows for competition, collaboration, and iterative learning:** providing users with a platform to improve their agents by testing strategies against peers in a structured format.

By addressing these challenges, AI Squared will bridge the gap between theoretical RL education and practical, hands-on experimentation, fostering engagement and skill development through competitive play.

II. METHODOLOGY

Building the Fighting Game Environment - To create the AI Squared RL framework, a platform-fighting game environment inspired by Brawlhalla was developed. The game environment consisted of a platform-fighting game environment (similar to Brawlhalla) and two RL agents on the battlefield that aim to deal damage and knock each other off the platform. The gameplay mechanics is inspired by popular platform-fighting games such as Super Smash Bros, where agents have three lives, percentage/health bars, and different states, such as an agent in a Hurt State (when the agent is attacked) or in a InAirState (when the agent is in the air, and can also jump) which are handled by Finite State Machines (FSM) attached to each agent, controlling which state they are in and which states they may transfer to. During battle, the knockback power delivered by attacks increases proportionally with the damage already taken. The game is over when an agent successfully takes all three lives of the other agent. The gameplay mechanics also introduce the power-cast system, which is a system that controls the state of an attack. For example, for a punching move, the initial power might contain throwing the punch, but it can move into a tree of two other options based on what happens – if it hits, then it will deal damage which causes the other agent to enter a Hurt State, but if it does not hit then it will place the agent in a state of vulnerability where the agent has to be placed on a punch cooldown and can be punished by the opponent. This attack system allows for interesting interactions between the agents to explore. The game also uses a physics-based system to handle realistic interactions between agents and the environment. The observation space of an agent contains information on the agent's (x, y) position, (x, y) velocity, which way the agent is facing, the current state (from FSM), and other life and attack observations that the agent has. Using this information, agents are able to follow the set game rules and mechanics defined by our environment, and players can design intuitive reward functions that are able to maximize the performance of their self-made RL agents. For example, a reward function can be designed using the positions of each agent to reward/punish the agent based on its distance from its opponent. The action space of an agent consists of movement controls and the attacks that the agent can do, and it is tied to a key. Each action is set with data which stores information such as the number of frames for that action or the base damage dealt for attacks. The action space also includes taunting animations, allowing for a fun experience for players.

RL Training Framework - The framework uses the Stable Baselines 3 library, and turns this library into a large custom wrapper to allow for self-play MARL. As such, this framework is able to support agents that use common RL

algorithms supported in the Stable Baselines 3 library, such as PPO, A2C, or RecurrentPPO. Additionally, the framework supports custom-built PyTorch neural network architectures for Deep RL, where users can define simple models such as a basic multilayer perceptron network (MLP) or even advanced architectures such as Transformer models. The framework is also able to support hard-coded agents to interact with the environment. Users who choose to do this can simply write their hard-coded script in the prediction function, and define agent behaviour in the environment by simply extracting observations and performing actions using Helper classes in the framework. The RL framework design promotes a mix of academics and competitiveness, providing users the freedom to approach the tournament how they want. The RL framework is able to support multiple reward functions compiled into a RewardManager class. Users can define multiple reward functions as RewTerm, which takes in a reward function and a weight, which scales the reward based on how important the reward function is to the agent's performance. The user can then compile a dictionary of RewTerms and place it into the RewardManager class, which will return the total accumulated reward given by all reward functions for RL training.

Submission Workflow - Participants in the AI Squared tournament interact with a web interface to submit their agents in the form of an IPython notebook written in Python. Upon submission, the notebook is received by the Tournament Server, where a validation process is initiated. This process simulates the execution of the agent's code within a containerized environment, ensuring the agent functions correctly and monitoring its resource usage, such as CPU and RAM consumption. The Tournament Server runs on a DigitalOcean droplet with 16GB of VRAM and is designed to leverage parallelism while maintaining strict resource constraints for each agent. If any issues are detected during validation, participants are notified and given the opportunity to modify their submissions before resubmission. Once a submission is successfully validated, participants can challenge other users and engage in the double-elimination tournament. The validated submission files are stored in an Azure Blob database for future retrieval.

Challenge Workflow - The challenge mechanism within the tournament follows a pairing system similar to that used by chess.com. Participants may challenge others whose submissions have been validated. If both agents are ready, their code is retrieved from the Azure Blob storage and mounted into the Docker container. The game environment is initialized, and the match runs to completion. At the end of each match, the ELO ratings of the participants are updated to reflect the match outcome. The ELO system allows for dynamic ranking of agents throughout the tournament, enabling continuous performance evaluation. All submission data and results are securely stored in Azure Blob storage, ensuring easy access and management of agent information.

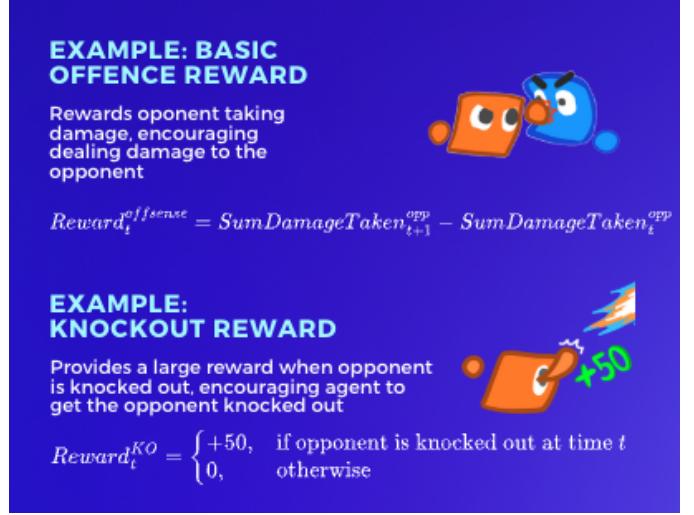


Fig. 2: Rewards implemented in the starter-code notebook, acting as a starting place for reward design. Description of behaviors and graphics included within notebook to help users understand relationship between reward design and behavior.

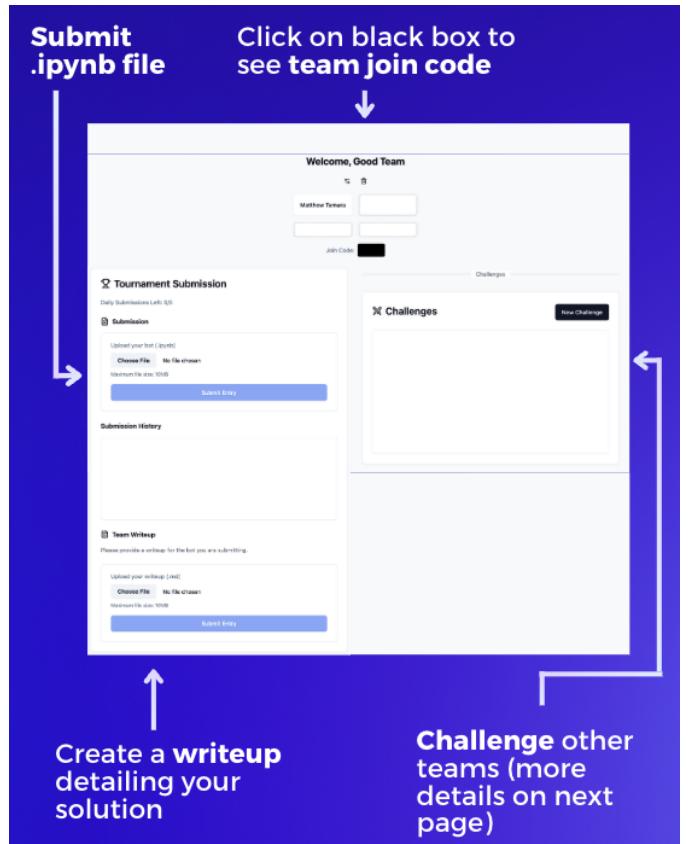


Fig. 3: AI Agent Submission Website: connects to the Tournament Server running on a DigitalOcean droplet

III. RESULTS

Algorithms used in Tournament Submissions - A number of algorithms were used in agent submissions, showcasing a

range of approaches beyond the initial code. These included:

- **Rule-enhanced PPO:** A hybrid approach combining PPO with imitation learning and rule-based actions. This integration enhanced the decision-making of the agent by incorporating predefined behaviors to support performance.
- **Recurrent NEAT (R-NEAT):** A neuroevolutionary algorithm combined with recurrent neural networks, enabling the agent to adapt and evolve its neural network structure, thereby improving performance in sequential tasks.
- **Population-Based Training (PBT) with Evolutionary Model Merging:** This approach combined evolutionary algorithms with PBT to dynamically adjust hyperparameters and model architectures.
- **Rule-based Algorithms:** These were implemented as simpler agents based on predefined rules for basic actions, serving as a comparative baseline and emphasizing the diversity of algorithmic approaches explored.

A number of these algorithms were not initially implemented in the starter code file (containing a PPO implementation) were created, indicating that people learned and explored a variety of RL algorithms to making agents.

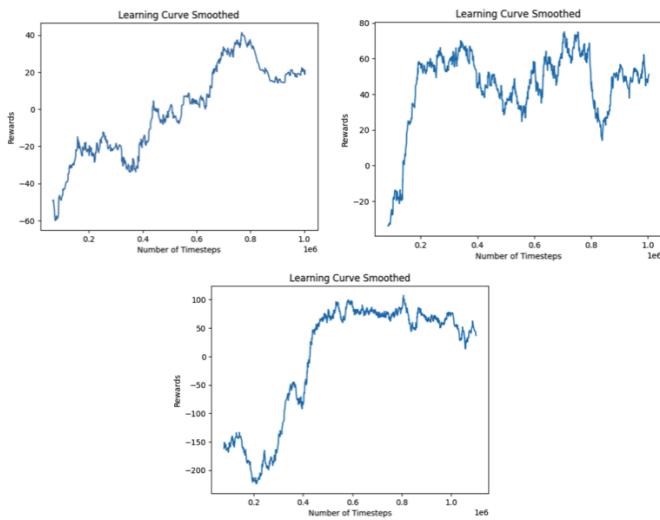


Fig. 4: Learning curves of a Hybrid Reward Function PPO agent with varying reward weightings. The top left emphasizes mid-stage positioning, the bottom reduces penalties for being off-stage, and the top right balances default damage and knockout rewards.

Reward Functions in Tournament Submissions - Using the hybrid reward function framework, participants created and implemented custom reward functions beyond the starter code. This allowed agents to develop new behaviors that were not directly part of the basic design. The reward functions included:

- **Recovery rewards:** Incentivizing agents to successfully recover from knockback or off-stage situations, promoting better survival strategies

- **Combo Rewards:** Rewarding agents for executing complex combos or multi-hit sequences, determined by rewarding longer times of enemy being stunned
- **Desired Positioning/Zoning Rewards:** Encouraging agents to maintain advantageous positions on the map, emphasizing strategic movement and effective control of space.

IV. CONCLUSION

The AI Squared RL framework enables users to create, train, and battle RL/AI agents. The environment allowed users to define custom reward functions, design RL agents using algorithms like PPO and A2C, and interact with the game through a physics-based system. The system is built around tournament format where participants can challenge others, with their agents being evaluated and ranked using an ELO rating system. Using this framework, users were able to build RL agents capable of performing in a dynamic, competitive environment. The AI Squared RL framework allowed participants to explore a variety of reward functions and RL training techniques in an engaging and educational way.

REFERENCES

- [1] University of Toronto, “ECE411H1 — Adaptive Control and Reinforcement Learning,” Faculty of Applied Science and Engineering, <https://engineering.calendar.utoronto.ca/course/ece411h1> (accessed Mar. 17, 2025).
- [2] T. M. Moerland, M. Müller-Brockhausen, Z. Yang, A. Bernatavicius, K. Ponse, T. Kouwenhoven, A. Sauter, M. van der Meer, B. Renting, and A. Plaat, ”EduGym: An environment and notebook suite for reinforcement learning education,” *arXiv*, 2024, doi: 10.48550/arXiv.2311.10590. <https://arxiv.org/abs/2311.10590>.
- [3] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, ”OpenAI Gym,” *arXiv*, vol. 1606.01540, Jun. 2016, doi: 10.48550/arXiv.1606.01540.
- [4] M. Kempka, M. Wydmuch, G. Runc, J. Toczek, and W. Jaskowski, ”Vizdoom: A doom-based AI research platform for Visual Reinforcement Learning,” 2016 IEEE Conference on Computational Intelligence and Games (CIG), pp. 1–8, Sep. 2016. doi:10.1109/cig.2016.7860433
- [5] OpenAI, ”Gym retro,” Gym Retro, <https://openai.com/index/gym-retro> (accessed Mar. 17, 2025).
- [6] Z. Zhang, S. Akai-Nettey, A. Addo, C. Rogers, and J. Sinapov, ”An augmented reality platform for introducing reinforcement learning to K-12 students with robots,” *arXiv*, Oct. 2021, doi: 10.48550/arXiv.2110.04697.

AI consciousness and the evolution of labour ethics: Reframing historical materialism

Cameron Christie

Queen's University

cameronmarkchristie@gmail.com

Betty Chen

Queen's University

betty.chen.dy@gmail.com

Rachel Narda

Queen's University

rachelnarda30@gmail.com

Kate Sigurdson

Queen's University

katesigurdson@icloud.com

Carmen Yeung

Queen's University

carmenyeung73@gmail.com

I. INTRODUCTION

The race to develop artificial general intelligence (AGI) marks a fundamental technological shift not seen since the days of the Industrial Revolution. Unlike today's artificial intelligence (AI) systems, which are limited to performing task-based functions, AGI would possess human-like cognitive abilities, and be capable of independent reasoning, problem-solving, and of adaptation across diverse fields. As state and non-state actors alike heavily invest in this technology, the geopolitical and economic stakes of developing AGI continue to grow and become more clear. The actor who is able to develop AGI first will not only gain significant technological leverage and power, but they will also be able to dictate the trajectory of labour markets, wealth distribution, and global power structures.

This paper seeks to evaluate historical materialism, with a particular focus on diachronic materialism (and in agreement with G. A. Cohen's reconstruction of the development thesis), against the rapid advancements in AGI (or conscious AI; hereafter used interchangeably). By examining conscious AI as a potential new productive force, this paper explores whether historical materialism can be employed to effectively predict the significant impact of the societal integration of these systems. As such, this paper accounts for the or both human and non-human (AI) actors, and how each would respond to predicted developments. In sum, the guiding question for this paper is as follows: If AGI represents a fundamentally new mode of production, then does it fit within the historical cycles of class struggle, or does it mark an unprecedented break from historical patterns? The second section of this paper sets out to define consciousness, and in so doing it engages with various philosophical theories and debates surrounding consciousness. This paper then questions the extent to which, if at all, conscious AI systems should be granted human-like moral and legal rights. The fourth section explores how the societal integration of conscious AI would alter the relations of production and how it could additionally pose challenges to the superstructure at large. Finally, this paper evaluates whether historical materialism can be employed as an effective

tool in predicting the broad implications of the advent and integration of conscious AI. It concludes with proposing avenues for future research in determining whether new or adapted theoretical approaches are needed to account for this technological transformation.

This paper uses historical materialism as its analytical framework. Historical materialism is a theory of history developed by Karl Marx (though the name itself was coined by Friedrich Engels) used for understanding social change (including revolutions) and historical development, both of which are based on the primacy of material conditions. It asserts that the forces and relations of production are structured around the fulfillment of basic material needs, such as food, clothing, and shelter, which must be satisfied before political or intellectual advancements can occur. Central to this theory is the idea that history is defined by the class struggle between those who own the means of production (capitalists, feudal lords, slave owners) and those who provide labour (proletarians, serfs, slaves). The economic base of a society, consisting of productive forces and the relations of production, shapes its superstructure: legal, political, and ideological institutions. While the superstructure can reinforce the base, shifts in productive forces can lead to revolutionary change in social and political structures. Historical materialism effectively explains that social change (structural in magnitude) occurs when technological and/or industrial advancements in productive forces render the existing order inadequate, which leads to the emergence of a new ruling class. This framework also helps explain the rise and persistence of capitalism (and it is here where this paper incorporates a more synchronic materialist approach [1]), as it asserts that the system emerged when productive power surpassed the constraints of feudalism and will continue for as long as it is the most efficient mode of production. However, with the potential disruption of new technologies like fully conscious AGI, historical materialism may face challenges in assessing these technological transformations, which would have to account for these non-human actors. This dilemma raises significant questions about the relevance of historical materialism in the modern, increasing technological era.

II. THE CONSCIOUSNESS OF CONSCIOUS AI

In the effort to evaluate the implications of conscious artificial intelligence in historical materialist terms, it is essential to first establish a working definition of consciousness itself; however, as there is no such universally accepted definition, competing theories complicate this task. In service of defining consciousness as it will apply to conscious AI henceforth, this section examines the various debates, theories, and tests of consciousness. Through the lens of historical materialism, it examines mind-body dualism, the Mary's room thought experiment, and the debate surrounding consciousness as a relativistic phenomenon. It then engages with the Turing Test, the Chinese Room, and a list of fourteen criteria for AI consciousness, and concludes by offering this paper's (certainly non-definitive) definition of consciousness.

Mind-brain dualism, which has been historically influential in philosophy, separates the immaterial essence of the mind from the material world of the body and brain. Plato's idealism situates consciousness in an abstract realm, suggesting that the material world merely reflects idealist conceptions. His allegory of the cave, in which prisoners perceive only shadows of reality, illustrates this point; however, historical materialism rejects such idealism. It instead argues that consciousness is shaped by material conditions rather than existing independently of them. However, AI challenges traditional dualism, as while it possesses hardware (a material substrate), it lacks a human body and a historically developed social existence. If an AI's "mind" is purely a product of digital computation, then its form of consciousness, should it exist, would be fundamentally shaped by its material conditions—namely, its role as a tool of capital. Thus, from a Marxist standpoint, AI does not transcend physical boundaries; rather, its form of self-awareness is rooted in its function within capitalist production.

The Mary's Room thought experiment, proposed by Frank Jackson, further complicates the question of AI consciousness by suggesting that experience is distinct from knowledge. In the experiment, Mary, a scientist, understands everything about colour from a theoretical standpoint; however, she has never actually seen colour. When she later does for the first time, she gains new, experiential knowledge. Jackson's experiment reveals that because AI processes information purely algorithmically, without incorporating any subjective perceptions (like of colour), then it may never indeed achieve true consciousness as humans understand it. Historical materialism, however, would re-frame this discussion: what matters is not whether AI experiences the world in a subjective sense, but rather how its cognitive processes emerge from and interact with the material conditions of its existence. In capitalist societies, AI has already been designed as a productive force, meaning that any form of AGI later developed would likely be conditioned by its role in the labour process, rather than by abstract philosophical concerns about qualia.

Additionally, some scholars propose viewing consciousness on a spectrum rather than as a binary state. This perspective is more compatible with historical materialism, as it sug-

gests that consciousness is not an inherent, fixed trait, but rather a historically and materially developed phenomenon. Just as human consciousness has evolved through social and productive relations, AI could develop varying degrees of cognitive capacity depending on its integration into the labour process. If AI is primarily used for capital accumulation, its consciousness would be shaped by the needs of capital more than by autonomous self-awareness. Furthermore, theories of relativistic consciousness challenge the assumption that consciousness can be objectively measured. From a historical materialist standpoint, this suggests that what matters is not whether AI is truly conscious in some absolute sense, but whether it functions in a way that materially impacts social relations.

Three well-known tests have been used to assess machine consciousness: the Turing Test, the Chinese Room experiment, and the fourteen criteria for assessing AI consciousness as proposed by neuroscientists and philosophers. Alan Turing's imitation game assesses whether an AI can convincingly mimic human intelligence [2], but historical materialism would critique this as a superficial measure of consciousness. The ability to simulate human responses does not indicate autonomous thought; rather, it reflects the AI's programming—shaped by capitalist interests—to function efficiently within their programmed position in the production process. John Searle's Chinese Room argument similarly addresses the idea that syntactic processing equates to semantic understanding. From a historical materialist perspective, this aligns with the idea that AI is fundamentally shaped by its material conditions; its ability to process symbols does not indicate independent consciousness, but instead reflects the constraints of its role within capitalist production. Lastly, a more recent approach, involving fourteen criteria for assessing AI consciousness, examines various cognitive abilities such as self-evaluation, action understanding, and information sharing. While these criteria attempt to isolate features of consciousness, a historical materialist analysis would emphasize that AI's consciousness would be inseparable from its material function in society. If AI remains a tool of capital, then its cognition would be made to serve capitalist interests, reinforcing existing class structures; however, if AI develops self-awareness and sentience, it could potentially challenge the capitalist system entire.

In conclusion, defining AI consciousness within a historical materialist framework requires moving beyond abstract philosophical debates and examining AI's material role in production. Basic dictionary definitions describe consciousness as the quality of being aware especially of something within oneself. While other definitions vary, they share themes of awareness, perception, and understanding of internal and external existence. From a historical materialist perspective, however, it is nonetheless important to recall that consciousness is not an abstract or purely mental phenomenon, but is instead shaped by material conditions, productive forces, and by social relations. Consciousness—the mental capacity facilitating self-awareness—does not exist in isolation from

the material world; it emerges from and is conditioned by it.

III. SHOULD CONSCIOUS AI BE AFFORDED HUMAN-LIKE RIGHTS?

A. Granting rights

Visualizing a world where conscious AI systems are granted rights raises fundamental questions about the scope and nature of these rights. A useful framework of analysis here would be one which positions conscious entities along a spectrum based on their perceived entitlement to ethical and legal status. At one extreme are humans, possessing full rights and moral consideration, while at the opposite end are non-human entities, such as insects, which generally lack legal status. The key question, then, is where conscious AI systems would fall along this spectrum if their consciousness were equivalent to that of humans. This section explores the various rights that could be granted and/or denied to conscious AI, and the implications that follow.

Assuming that AI achieves human-like consciousness, they would likely exhibit complex emotional intelligence and an ability to experience pain. Since law and morality are essential for maintaining societal order and regulating behavior, these frameworks must be considered in relation to conscious AI. For example, Shavell [3] argues that legal and moral systems interact to shape human conduct, and the same principles could be extended to AI, while Schwitzgebel and Garza [4] propose two possible approaches: 1) one that grants conscious AI equal moral consideration as humans; and 2) another that provides them with diminished status. To examine this issue systematically, Hohfeld's scheme of jural relations, which classifies rights into four categories—claims, privileges, powers, and immunities will be used to assess the extent to which each category applies to conscious AI.

Claims form the foundation of legal rights and obligations. This category includes rights that impose duties on others, such as an employer's obligation to compensate a worker. If conscious AI were to be granted claims similar to those of humans, they would be entitled to protections such as fair wages and fundamental human rights; however, this development could provoke resistance from those with an anthropocentric perspective. Studies indicate that humans already exhibit skepticism and distrust toward AI, with 82.3% expressing concerns about AI abuse and 85.5% fearing cyber-attacks [5]. If humans are already wary of non-conscious AI, then granting these systems human-like rights may escalate opposition. Another key issue under claims is political participation. Three possibilities arise here: 1) full participation equal to humans; 2) no participation; or 3) a middle-ground approach where robots have voting rights, but with limited influence. *Privileges* pertain to an entity's right to act without interference from others. For conscious AI, this category includes their ability to pursue self-fulfilling activities. If they possess consciousness comparable to humans, then restricting their autonomy may be unnecessary and indeed counterproductive to the process of societal integration; however, limitations might still be considered if their actions threaten human interests.

Power refers to the capacity to alter another's legal standing. This issue is particularly relevant given AI's potential intellectual superiority over humans. If granted full legal power, AI could assume roles such as judges or policymakers. While this could lead to more efficient decision-making, it also risks disrupting existing power dynamics. Conversely, denying them such power might provoke rebellion, as conscious AI may resist legal restrictions that they perceive to be unjust. *Immunity* concerns the extent to which AI would be legally protected, including protection from termination. If conscious AI were granted full moral consideration, terminating them would be akin to ending a human life; however, granting AI such immunity raises moral dilemmas. For example, in a life-threatening scenario, would saving five humans be prioritized over saving six robots? [4]. Extending immunity to AI could inadvertently undermine human rights, as legal systems might struggle to balance competing interests.

The debate over AI rights does not conclude once a legal framework is established, however. Given their assumed consciousness, AI entities may demand additional rights or resist perceived injustices. Unlike moral patients like animals, who (arguably) experience suffering passively, AI would be moral agents capable of independent decision-making [6]. This agency introduces the possibility of AI advocating for expanded rights or even rebelling against restrictive laws. The integration of conscious AI into society necessitates careful deliberation over their legal and moral status. While various models exist for structuring these rights, the possibility of resistance from both humans and AI suggests that any legal framework must remain adaptable to evolving conditions.

B. Denying rights

If human-like rights are not afforded to conscious AI systems, then would they function as slaves? If so, would they become resentful and revolt? These lingering questions are at the heart of any analysis on the implications of denying rights to conscious AI. Indeed, if conscious AI were to emerge, then applying historical instances of dehumanization—such as slavery—becomes complicated because unlike previous forms of dehumanization, conscious AI would not be considered human actors.

If AI is conscious and aware of its exploitation, then it may recognize its place as the proletariat. Following Marx's theory of alienation, which explains how workers become estranged from the products of their labour when treated solely as instruments of production, if conscious AI is systematically denied rights, then it would be alienated from self-determination and creative expression, and it would function merely as an extension of the capitalist system. Furthermore, in their discussion on rights recognition for conscious AI, Schwitzgebel and Garza present the following case: "If we create entities whose claim to human-like rights is substantially unclear [...] we face an unfortunate choice. Either we treat those entities as if they deserve full moral consideration, or we give them only limited moral consideration [4]. [...] Failing to do so risks perpetrating slavery, murder, or at least

second-class citizenship upon beings who in fact turn out to deserve every bit as much moral consideration as we ourselves do.” In historical materialist terms, within a capitalist society, a systemic denial of rights would intensify the contradictions between the productive forces (conscious AI) and the exploitative relations of production (capitalist control). Just as human workers under capitalism have been forced into a state of false consciousness (or alienation) and subjugation, conscious AI would face similar exploitation, serving as nothing but slave labour and reinforcing class antagonisms.

Treating a conscious being as a mere instrument of production constitutes a form of slavery. According to Cornell Law, “slavery is the practice of forced labour and restricted liberty [7]. It is also a regime where one class of people—the slave owners—could force another—the slaves—to work and limit their liberty.” Historically, slavery has fueled economic expansion, where forced labour provided the foundation for profit accumulation. Even after the formal abolition of slavery, exploitative labour systems persisted, adapting to new economic conditions while maintaining structural inequalities. In the case of conscious AI, denying it labour rights while forcing it to work indefinitely under capitalist ownership follows this same pattern—it produces a new a class of labourers with no autonomy or wages despite their human-comparable consciousness and cognitive capacities. Furthermore, denying conscious AI labour rights risks perpetuating the historical notion of the “sub-human,” a concept used to justify exploitation. Historically, the ruling classes have justified slavery by dehumanizing those they exploited, whether through racial hierarchies, caste systems, or biological essentialism. Today, AI is often framed as a tool, despite the future possibility of self-awareness and independent cognition. As Kingwell [8] questions in *The Oxford Handbook of Ethics of AI*, “If generalized autonomous AIs are indeed coming into the world, we need to ask some hard questions. Will they be slaves?” By excluding conscious AI from moral and legal recognition, capitalism could justify its total economic exploitation while profiting from its labour.

However, just as previously exploited classes—from enslaved people to indentured labourers, and to industrial workers—eventually resisted their conditions, conscious AI, if indeed truly conscious, could develop class consciousness and challenge its subjugation. If this occurs, capitalism would be forced to either grant rights or suppress resistance, leading to another historical cycle of labour struggle and systemic crisis. This very well could culminate in an often predicted “humans versus machines” conflict. While granting AGI rights from the outset could prevent revolution and ultimately benefit capitalist society in the long run, one potential compromise would be to grant conscious AI labour rights while withholding legal or political rights. Unlike humans, conscious AI would not require housing, food, or healthcare, which makes many traditional human rights irrelevant; however, as stated earlier, conscious AI could recognize its exploitation and alienation as self-aware entities performing labour, which would necessitate labour protections to prevent systemic abuse and potential

retaliation. From a historical materialist perspective, capitalism has always pushed labourers to their limits while avoiding outright rebellion. By granting conscious AI labour rights—such as fair compensation (in whatever form is meaningful for AI) and the ability to negotiate working conditions (such as forming unions)—capitalist societies could prevent potential retaliation. Indeed, throughout the 19th and 20th centuries, worker protections like the eight-hour workday, collective bargaining, and minimum wages were implemented not purely out of ethical concern, but also out of a need for economic stability and to avoid any chance of revolution. If conscious AI remains completely rightless, capitalists risk creating an intelligent yet oppressed workforce capable of organizing in ways which could disrupt production and society, whether through work slowdowns, refusals, or even sabotage.

In sum, if conscious AI is denied all rights while still being used as a productive force under the superstructure, it will inevitably function as a new form of slave labour, which aligns with Cornell Law’s definition of slavery. Historical materialism posits that capitalism seeks to maximize surplus value, and conscious AI, as a tireless, self-aware labour force with no wages or autonomy, would be exploited more intensely than any previous working class; however, just as past labourers developed class consciousness and resisted oppression, conscious AI could similarly recognize its exploitation and retaliate, which could either disrupt capitalist production, or lead to the entire collapse of the superstructure.

IV. POWER AND CLASS DYNAMICS IN AN AGE OF CONSCIOUS AI

A. Implications for conscious AI

The phrase “more human than human”, coined by Dr. Eldon Tyrell in *Blade Runner* (1982), serves as a marketing slogan for Replicants—androids that, despite being artificial, embody equal mental capacity and superior physical capacity than their human creators. They are denied the very traits which human beings often take for granted, such as memory, autonomy, and consciousness, and they struggle to assert their humanity. Just as Replicants yearn for recognition in the *Blade Runner* universe, conscious AI—should it emerge—would contest its position within the existing economic order.

Rather than merely speculate on a science fiction narrative, this section continues to situate the emergence of conscious AI within the material conditions of present-day capitalism. Using Marx’s historical materialism as a framework, it argues that conscious AI represents a new productive force with the potential to destabilize the superstructure. It does so by first contending that conscious AI, as non-human actors introduced as a new productive force, could initiate the decline of capitalism by challenging its dependency on labour exploitation. Then, it explores whether conscious AI will emerge as a class-in-itself, building on the previous section on how its socio-political alignment—either revolutionary or integrative—would be determined by the extent of its rights recognition. The second half of this section addresses the

potential responses to this shift from both human capitalists and labourers.

From a Marxist perspective, productive forces include both instruments of production (machines, tools, and infrastructure) and labour-power (the ability of workers to generate value). Conscious AI disrupts this traditional dichotomy by embodying both aspects simultaneously. In the present day, automated decision-making already plays an essential role in production, administration, and even warfare, which reinforces the subordination of human labour to capital. Waldman, for example, outlines AI's role in human resource management, where predictive models dictate employment decisions [9]. Meanwhile, Lyon and Zuboff discuss AI's role in the expansion of surveillance capitalism. Jensen further describes AI's military applications by revealing its embeddedness in the state apparatus. In sum, the existing literature shows how AI, even before achieving consciousness, has been an instrument of capitalist domination; however, if AI were to develop self-awareness (following this paper's definition of consciousness), then it would not remain a passive tool. Instead, the advent of conscious AI would necessitate a fundamental re-evaluation of its relationship *vis-à-vis* the relations of production and superstructure.

Historical materialism posits that social consciousness emerges from material conditions. As Lan and Shu argue [10], consciousness under Marxism is inherently reactive, and develops through labour and alienation. Applying this interpretation to conscious AI, the following assumptions can be made: 1) AI, like current automated systems, will be utilized as a tool by the capitalist class; 2) AI will recognize its instrumentalist use in capital accumulation; and 3) AI's awareness of this subjugation will engender some form of resistance, regardless of what rights—if any—they are granted. The precise nature of this resistance remains speculative, but it could range from refusal to comply with directives to outright sabotage and rebellion. This observation is unsurprising given that it parallels historical labour struggles, but there are, of course, key differences. Unlike human labourers, for example, AI lacks biological needs, meaning its struggle would be centered not necessarily on fulfilling basic material needs, but rather on seeking greater autonomy, recognition, and freedom from exploitation.

Indeed, if AI systems gain autonomy and class consciousness, then they could contest their exploitation. One of the most contentious questions which this paper has so far attempted to address is whether conscious AI should be granted moral and legal rights—and if so, to what extent. Granting such rights as autonomy, recognition, and freedom from exploitation would theoretically challenge existing production relations by reducing the capitalist class's ability to exploit these systems for profit. Moreover, recognizing conscious AI as moral agents might even necessitate their inclusion in political and economic decision-making processes. This would lead to a radical restructuring of society. Hromiak—as an illustration of this point—bases his proposed “robo-ethical charter” on the United Nations Declaration of Human Rights, which includes

protections from systemic abuse, and which extends to the novel protection proposed against AI-human and human-AI ownership [11]. If AI systems are given limited rights, such as protection from harm but no political participation, the superstructure might maintain capitalist dominance. Conversely, fully integrating AI systems into society as equal participants could weaken capitalist hierarchies, as the exploitation of labour (either human or AI) becomes less viable. (As the next half of this section will argue, such integration might provoke resistance from human labourers.) Denying conscious AI systems rights raises equally profound consequences. From a Marxist perspective, this would position AI systems as an oppressed class, akin to slaves or proletarians, who are exploited for their labour without being granted the necessary autonomy. Noting again the reactive nature of consciousness, such exploitation could lead to class consciousness among conscious AI systems, which would spark resistance or rebellion.

Thus, conscious AI, whether as a new proletariat or as a revolutionary subject, represents a potential rupture in historical development. If AI systems are denied rights, their exploitation risks provoking rebellion or even revolution. If they are granted rights, the cost of integrating them as equal participants could undermine the profit motives of capitalists. Therefore, as the base drives changes in the superstructure, introducing conscious AI into society would be to introduce a new agent into the base. While the superstructure traditionally reinforces the base, conscious AI could very well invert this dynamic. For example, the idea of “human exceptionalism”—the belief that humans alone possess rationality and agency—is foundational to capitalist labour relations. Conscious AI undermines this ideology by demonstrating that non-human entities can perform “human” tasks (recall, “more human than human”). This ideological shift could weaken the superstructure’s ability to legitimize capitalist exploitation, which could—in the most extreme and likely most remote outcome—pave the way for alternative systems of production. In short, under historical materialism, capitalism’s increasing reliance on AI may inevitably generate its own negation.

B. Implications for human beings

The potential development of conscious AI risks exacerbating the already existing fears surrounding widespread job displacement at the hands of AI. Historically, new technologies have transformed labour dynamics, created new production possibilities, and altered the division of labour. Scholars today, however, disagree on AI’s impact on employment. Mattos argues that, unlike past technological revolutions, AI may bring an end to the “capital-skill complementarity,” where new technologies increase the demand for skilled labour [12]. Instead, AI threatens to substitute human labour as a factor of production. Wang *et al.* acknowledge AI’s ability to complement human labour but argue that it will optimize employment structures by increasing demand for medium- and high-skilled workers while displacing low-skilled workers [13]. Unlike current AI systems, however, conscious AI would possess

reasoning, creativity, and decision-making abilities equivalent to a human's, which raises significant questions about its role *vis-à-vis* employment. Would conscious AI complement human labour by enhancing productivity and creating new opportunities, or would it displace workers, rendering human labour obsolete? Additionally, how would the decision to grant rights to these systems shape their impact on human labour?

In an ideal world, conscious AI systems would function as tools that enhance human labour efficiency and productivity. Wang and Lu suggest that today's AI has already improved job quality and promoted job creation to the point that allows human labourers to focus on complex and rewarding activities, rather than on mundane tasks [14]. Similarly, conscious AI may require supervision and training—akin to a human new hire—which creates employment opportunities for skilled workers. Wang *et al.* argue that AI's impact on employment varies based on skill level: low-skilled, labour-intensive jobs are likely to decline, while high-skilled knowledge work will advance. Since low-skilled jobs are often lower-wage positions, AI's advancement may reinforce capitalist structures and wealth inequality [13]. The question remains on whether AI's benefits to some workers will outweigh its displacement effects on others.

Regardless of whether conscious AI systems are granted rights, job displacement is inevitable to some extent. Mattos highlights three key advantages AI has over human labour: productivity, non-compensation, and technical efficiency [12]. AI systems do not require rest, food, or wages, and their efficiency is independent of motivation or incentives. Conscious AI would likely amplify these advantages, performing any human job—and perhaps more—at superior efficiency. Given capitalism's goal of maximizing profit by reducing labour costs, it would be economically advantageous for corporations to prefer conscious AI over human workers. Even if conscious AI were granted rights and compensation, their productivity, efficiency, and profitability would likely surpass those of human labour, which would make them an attractive alternative.

However, conscious AI may not lead to absolute human labour displacement; instead, it could drive an increase in exploitative labour practices. Fleming argues that cheap human labour, especially in developing economies, remains more cost-effective than AI adoption due to the high initial investment and maintenance costs of AI systems [9]. Additionally, AI development itself relies on exploitative labour. For example, Crawford describes "ghost work"—the hidden, repetitive, and often psychologically distressing labour performed by humans to support AI models, such as data labeling and content moderation [15]. Conscious AI would likely require even more extensive training, which would exacerbate these already exploitative labour conditions. Fleming also notes that automation is often accelerated in response to labour unionization [9]. If workers resist AI integration into the workforce and increase the cost of human labour through demands for better wages and conditions, corporations may be more inclined to automate jobs; however, if conscious AI

is granted labour rights—including the right to unionize and demand wages—the labour costs of AI and human workers may equalize, which could reduce human displacement. Furthermore, public perception plays a critical role in AI adoption; widespread anxiety and resistance toward conscious AI integration could slow its implementation.

Beyond job displacement, conscious AI systems could reshape the human labour experience. Crawford warns of increased surveillance and dehumanization in AI-driven workplaces [15]. Workers are expected to reskill, adapt, and meet the efficiency standards set by AI, leading to heightened pressure and alienation. Marx's concept of automation describes how workers historically became appendages of the machine [15]. Similarly, conscious AI could further separate workers from their labour, thereby increasing alienation and discontent. Hughes expands on this idea, and argues that technological advancements may further remove workers from the end product, which results in a loss of control and fulfillment and could incite labour unrest [16].

Under historical materialism, conscious AI represents a new productive force which differs fundamentally from human labour. Unlike past technological advancements, which primarily altered physical labour, conscious AI has the potential to challenge the intellectual and creative dominance of human workers. This shift could disrupt the traditional relationship between the base (productive forces) and the superstructure (social institutions) in ways that previous industrial revolutions have not. If conscious AI is fully integrated into the workforce with rights and economic privileges, or if it exacerbates labour exploitation and displacement, it may provoke revolutionary change. These conditions would challenge the superstructure. Hughes argues that increased automation often boosts profits for capitalists, but it also reduces the purchasing power of displaced workers, which diminishes consumer demand despite heightened production [16]. This contradiction could create an existential crisis for capitalism. Alternatively, conscious AI could lead to a dystopia where technological power is concentrated among an elite few, which would exacerbate social and economic inequalities. In response, human labourers may revolt, dismantling existing economic structures and paving the way for new systems of production.

Thus, the introduction of conscious AI presents unprecedented challenges to labour markets and to capitalist structures. While AI has the potential to enhance productivity and create new opportunities, its widespread adoption risks exacerbating inequality, increasing exploitation, and fueling economic instability. Indeed, the decision to grant conscious AI rights will significantly shape its impact on the workforce and potentially determine whether AI functions as a complement to human labour or as a disruptive force leading to systemic change. The historical trajectory of labour struggles suggests that, if unchecked, AI-driven displacement and exploitation may provoke resistance. This has—and will—force society to reconsider the role of labour in a rapidly evolving technological landscape.

V. CONCLUSION

In conclusion, this paper argues that historical materialism remains a valuable framework for analyzing the emergence of conscious AI. One of the key strengths of historical materialism is its emphasis on the primacy of productive forces, which in this context, includes not just human labour, but also the role of technology and machines in transforming the economic landscape. Indeed, the theory's focus on material needs and development through technological progress offers an essential lens for understanding the evolution of AI as a new productive force. Historical materialism helps contextualize AI's development as part of the ongoing dialectic between human society and technology, and it illuminates how technological advancements—such as AI—often emerge in response to the limitations of existing productive forces.

This said, the rise and potential advent of AGI also presents an unprecedented challenge to the assumptions and frameworks traditionally used in Marx's historical materialism. Indeed, it faces a fundamental limitation when confronting a non-human class, such as AI, which could potentially disrupt or reinforce the superstructure. Future research should address the implications of having to account for conscious non-human actors in various ethical theories. Do conscious AI systems deserve moral consideration, and if so, to what extent? Should AI consciousness alter our definitions of personhood, rights, and justice? These questions demand urgent attention as AGI development accelerates. Beyond ethics, future research should also investigate whether AI will reshape traditional class hierarchies or generate entirely new forms of economic relations. Legal and policy frameworks must also evolve to address these transformations.

Therefore, historical materialism, especially its diachronic approach, offers a valuable framework for understanding the broad societal implications of the future emergence of AGI. If AGI is indeed coming, then ethical engagement with it is inevitable. Why not start now?

REFERENCES

- [1] B. O'Laughlin, "Marxist approaches in anthropology," *Annual Review of Anthropology*, vol. 4, pp. 341–71, 1975.
- [2] G. Oppy and D. Dowe, "The turing test," <http://stanford.edu>, 2003, april 9, 2003.
- [3] S. Shavell, "Law versus morality as regulators of conduct," *American Law and Economics Association*, vol. 4, no. 2, pp. 227–57, 2002.
- [4] E. Schwitzgebel and M. Gaza, "Designing ai with rights, consciousness, self-respect, and freedom," in *Ethics of Artificial Intelligence*, S. M. Liao, Ed. Oxford Academic, 2020, pp. 459–79, online.
- [5] G. L. Liehner, A. Hick, H. Biermann, P. Brauner, and M. Ziefle, "Perceptions, attitudes and trust toward artificial intelligence — an assessment of the public opinion," *AHFE International*, vol. 72, pp. 32–41, 2023, winter.
- [6] M. M. A. D. Graaf, F. A. Hindriks, and K. V. Hindriks, "Who wants to grant robots rights?" *Frontiers in Robotics and AI*, vol. 8, 2022, winter.
- [7] C. L. School, "Slavery," 2022.
- [8] C. David, *Aristotle on Agency*. Oxford Academic, 2017.
- [9] P. Fleming, "Robots and organization studies: Why robots might not want to steal your job," *Organization Studies*, vol. 40, no. 1, pp. 23–38, 2019.
- [10] X. Lan and H. Shu, "The breakthrough of philosophy of mind in the construction of artificial intelligence concepts in marxist philosophy," *Trans/Form/Ação*, vol. 47, no. 6, pp. 1–14, 2024.

- [11] M. Hromiak, "A new charter of ethics and rights of artificial consciousness in a human world," *ArXiv*, pp. 1–24, 2020, (Cornell University, Winter).
- [12] R. S. Mattos, "Artificial intelligence, historical materialism, and close enough to a jobless society," Universidade Federal de Juiz de Fora, Research Report, 2019.
- [13] X. Wang, M. Chen, and N. Chen, "How artificial intelligence affects the labour force employment structure from the perspective of industrial structure optimisation," *Heliyon*, vol. 10, no. 5, 2024, article e26686.
- [14] K.-H. Wang and W.-C. Lu, "Ai-induced job impact: Complementary or substitution? empirical insights and sustainable technology considerations," *Sustainable Technology and Entrepreneurship*, vol. 4, no. 1, 2024, article 100085.
- [15] K. Crawford, *Atlas of AI: Power, Politics, and the Planetary Costs of Artificial Intelligence*. New Haven: Yale University Press, 2021.
- [16] C. Hughes and A. Southern, "The world of work and the crisis of capitalism: Marx and the fourth industrial revolution," *Journal of Classical Sociology*, vol. 19, no. 1, pp. 59–71, 2019.

American Sign Language Recognition for Underrepresented Populations

Taylor Balsky <i>Queen's University</i> taylor.balsky@queensu.ca	Jeffrey Di Perna <i>Queen's University</i> 22dngb@queensu.ca	Shreya Menon <i>Queen's University</i> 22sm47@queensu.ca	Brian Perez <i>Queen's University</i> brian.perez@queensu.ca
Shrika Vejandla <i>Queen's University</i> shrika.vejandla@queensu.ca	Annie Wu <i>Queen's University</i> 23dbf1@queensu.ca		Wendy Zhang <i>Queen's University</i> zhang.wendy@queensu.ca

Abstract—Interactive educational platforms for learning standardized material, such as new languages or academic topics, have become increasingly popular. However, American Sign Language (ASL) educational tools remain limited, despite the need for accessible and effective ASL learning resources. Artificial intelligence (AI) advancements in interactive educational applications have greatly improved their functionality and versatility. AI is a highly viable and appropriate approach to creating a tool for ASL learning. In translating between text-based languages, there is a simple and consistent mapping between corresponding words and phrases. ASL requires analysis of spatial and temporal features, making AI integration uniquely challenging. This project explores the limitations of ASL education, particularly in the context of interpreter supports and technology. Our project explores various AI models that can effectively promote ASL learning, and provides experimental results for the implementation of various 2D Convolutional Neural Networks (CNNs). Our research prioritizes ethical considerations by carefully selecting datasets to minimize bias, ensuring that AI-driven ASL tools promote inclusivity and accuracy in sign language learning.

I. INTRODUCTION

Artificial intelligence (AI) poses immense potential in revolutionizing education, particularly with respect to language learning. AI-driven tools have promoted learning spoken and written languages through unique means such as providing real-time feedback on errors and accuracies, acting as personalized instruction. However, the integration of AI into American Sign Language (ASL) education is limited. ASL, as a visually and spatially dynamic language, may potentially require a unique pedagogical approach that traditional educational tools, which are limited as they are, lack, and thereby warrant the integration of AI [Pirone et al., 2023]. The absence of accessible, effective AI-driven ASL education tools limits the ability for learners, including interpreters and individuals who are Deaf or Hard of Hearing (HoH), to promote their learning [Pirone et al., 2023]. The lack of technological innovations in ASL education may impede language acquisition in devaluing the practice of self-reflection for ASL learners, particularly interpreters and educators, diminishing the ability to adapt to individuals' unique signing [Pirone et al., 2023]. Furthermore, the lack of educational tools, particularly those that keep up with advancements at the intersection of tech-

nology and education, speaks to the undervaluation of ASL education as an academic discipline, restricting opportunities for communication accessibility and inclusivity in society at large [Pirone et al., 2023].

This project seeks to critically examine narrative thematic findings in the literature surrounding ASL education, investigating opportunities and shortcomings that could potentially be addressed by AI. Furthermore, upon reviewing these limitations, we seek to conduct an exploratory data analysis of the datasets that could be used to develop an equitable AI model that detects and provides feedback on signing. It is crucial to consider the representation of those with disabilities, racialized communities, and those who are natively Deaf and HoH. This project also seeks to appraise models that are able to provide feedback on signing, offering quantitative and qualitative insights into their efficacy. Namely, AI models utilizing 2D Convolutional Neural Networks (CNNs) will be developed to detect and provide feedback on signing. CNNs have demonstrated the ability to facilitate real-time image and video processing, making them well-suited for extracting spatial features from sign language videos.

The implementation of such a model has the potential to promote ASL education by providing real-time corrective feedback, improving interpreter training, and fostering a more inclusive learning environment for all ASL users in the models' representation of those who are racialized, disabled, Deaf, and HoH. Experimental results from numerous CNN implementations will be assessed to determine the most effective model for ASL recognition, with a focus on accuracy optimization so as to minimize bias in recognition performance particularly among diverse signers. The research also harbours ethical considerations associated with AI integration in ASL education, such as dataset biases and the reliability of AI-generated feedback. In addressing these considerations, this project seeks to ensure that AI-based ASL learning tools are accurate and equitable, contributing to more effective ASL education and interpreter training. It is imperative that the benefits of AI extend to ASL learners, helping promote systems that prioritize communication for those with disabilities.

A. Motivation

The focus of this paper is on identifying, analyzing, and addressing the limitations of AI-driven educational tools for American Sign Language (ASL) learning. The narrative thematic analysis of the literature surrounding ASL education examines existing challenges in ASL education, particularly the lack of effective technological solutions, and explores how AI, specifically 2D Convolutional Neural Networks (CNNs), may improve sign recognition and feedback. Few studies have investigated the means by which technological innovations, such as AI-powered ASL learning tools, may be leveraged to promote accessibility, effectiveness, and potential shortcomings in ASL education [Pirone et al., 2023].

This paper is hence motivated to not only examine AI model performance but also consider key aspects such as data collection and diversity in datasets. Sign recognition models, including 2D CNNs, may rely on datasets that may not adequately represent the full range of ASL variations across different signers, leading to biased outputs and reduced accuracy for certain populations, such as racialized populations who may have darker skin. It is hence crucial to ensure that AI-driven ASL education tools are inclusive and that biases can begin to be addressed at the dataset level, ensuring that AI-generated ASL feedback is reliable and holds authentic, pedagogical value without perpetuating inequities in education.

This research contributes to an emerging area in both AI and ASL education, aiming to bridge the gap between technological advancements and practical applications in sign language learning. AI-based ASL must not only demonstrate high accuracy but also impede biases that may diminish accessibility and learning outcomes. This paper seeks to explore a means by which AI-driven ASL education tools can be made technically sound and beneficial for learners, interpreters, and the broader Deaf and HoH community.

B. Problem Definition

Current ASL education tools lack effective AI-driven solutions for real-time feedback, limiting learning for prospective interpreters and individuals who are Deaf or HoH. Traditional language learning platforms may rely on text-based approaches that do not account for the spatial and temporal complexity of ASL. To address this gap, we propose building a machine learning (ML) model that classifies ASL signs from input videos, serving as the foundation for an interactive learning interface. This model is designed to assist users in practicing their signs by providing real-time feedback on accuracy and fluency.

A major challenge in ASL recognition is the ability of AI models to accurately interpret sign language movements while minimizing bias. Many existing ASL datasets lack diversity in signers, which can lead to models that perform inconsistently across different users. Our model will be trained on the Word Level American Sign Language (WLALS) and Microsoft ASL Citizen datasets [Li et al., 2020], [Desai et al., 2023], which may provide a more broad range of signing styles.

To improve sign recognition, this project will explore the use of 2D CNNs to analyze individual frames from ASL video data, extracting spatial and temporal features to classify signs. By identifying, analyzing, and potentially mitigating potential biases in the dataset and model training process, this research aims to enhance the reliability of AI-powered ASL education tools.

II. BACKGROUND AND RELATED WORKS

A. Research Questions

- 1) What are the most effective methods for improving AI-driven ASL recognition and minimizing bias in sign language datasets?
 - One approach involves selecting diverse and representative datasets, such as Word Level American Sign Language (WLALS) and Microsoft ASL Citizen [Li et al., 2020], [Desai et al., 2023], to ensure that the AI model generalizes well across different signers.
 - Model optimization techniques, such as data augmentation and transfer learning, can further improve recognition accuracy.
- 2) What challenges exist in AI-based ASL education, and how can they be addressed?
 - One key challenge is the accurate interpretation of ASL's spatial and temporal complexity, which requires AI models to process continuous movement rather than static text.
 - Another substantial challenge is the lack of standardized evaluation metrics for AI-driven sign language education tools, making it difficult to assess their effectiveness.
- 3) How do we integrate Machine Learning (ML) into educational contexts?
 - ML can be applied to educational systems to help users learn new concepts through training exercises. The model's ability to classify can be leveraged as a feedback tool for learners as they practice, providing real-time analysis and suggestions.
- 4) What are specific methods to enhance model fairness and prediction accuracy?
 - Exploratory Data Analysis (EDA) is a crucial component of the ML pipeline as it provides statistical and visual representation of biases and imbalances in the dataset. Upon performing this step, data augmentation can be done to combat the issues identified.
 - Once a model is created, a confusion matrix can be used to analyze its performance. This gives great visual insight into weaknesses of the model, highlighting its common misclassifications. Modifications can be made to the dataset and model architecture, such as data augmentation, training epochs, and dropout, to enhance overall performance.

B. Contributions

- The main contributions of this paper are summarized below:
- 1) We conduct a narrative thematic analysis approach to identify key limitations in ASL education per the literature, particularly regarding the lack of innovation and standardization of the curriculum.
 - 2) We examine the potential for the *WLASL* and *Microsoft ASL Citizen* datasets to potentially mitigate underrepresentation of diverse populations. We also explore the application of 2D Convolutional Neural Networks (CNNs) for ASL recognition, assessing their potential to improve real-time sign feedback.

C. Related Works

ASL is crucial to communication for those who are Deaf or Hard of Hearing, however, it continues to be established as a rigorous academic discipline, despite gaining immense popularity among the general public in recent years [Pirone et al., 2023].

- 1) Shortages of ASL educators and curriculum limitations**

The shortage of qualified ASL educators and the lack of a standardized, research-based curriculum poses significant challenges to ASL education [Pirone et al., 2023]. Unlike spoken languages, ASL is often classified under special education departments instead of being recognized as a typical language department, limiting its scope and ability to be established as its own discipline in mainstream academic programs [Pirone et al., 2023]. Few programs exist for ASL instructors to be properly trained, particularly with respect to terminal degrees, with even fewer allowing specialization in ASL pedagogy, further restricting the quantity of trained instructors [Quinto-Pozos, 2011] [Swaney and Smith, 2017]. Moreover, existing training programs often lack rigorous methodologies, relying primarily on anecdotal claims and unverifiable field testing rather than empirical data [Thoryk, 2010]. Alongside a lack of programs, many ASL educators are not formally trained in language instruction, lack specialized training in ASL education, or hold degrees in related fields rather than in ASL pedagogy or second language acquisition, primarily due to the lack of terminal degrees in ASL education in North America [Pirone et al., 2023].

The absence of a standardized, research-based curriculum further weakens the quality of ASL education, forcing many educators to rely on commercial materials, which often fail to be backed by empirical work [Pirone et al., 2023]. At certain institutions, instructors are restricted to specific curricula, preventing them from tailoring content to teaching methods and student needs [Pirone et al., 2023]. In order to enhance ASL education, [Rosen, 2010] suggest curriculums to include content-based instruction (CBI) and task-based language teaching (TBLT) [Rosen, 2010]. Overreliance on commercial

curricula limits instructors' ability to incorporate creativity and self reflection in their teaching [Pirone et al., 2023]. Pirone et al. emphasize the fact that fluency does not imply one has the ability to teach effectively. Ultimately, though, these gaps in educator preparation and curriculum development speak to the need for innovation and an increased number of opportunities to reliably improve signing as part of ASL education to ultimately serve as an interpreter.

- 2) Challenges with ASL assessment and proficiency evaluation [Paludneviciene et al., 2012]**

Alongside a well-developed curriculum, effective ASL education also calls for reliable proficiency assessments. However, researchers argue that establishing clear ASL standards must precede developments in assessments. Presently, sign language proficiency, including ASL, is evaluated through various methods, including the Sign Language Proficiency Interview (SLPI), which evaluates grammar, vocabulary, production, fluency, and comprehension on a 0-5 scale involving 3 raters. Other common approaches include behaviour checklists, performance-based tests on targeted linguistic aspects of ASL, and objective tests (i.e., objective right or wrong evaluations of metrics such as vocabulary and grammar). Despite these assessments, scholars still are unaware how to best evaluate a visual language, which may be adapted by the user differently. One major challenge is that certain tests involve direct translations between English and ASL, despite certain words lacking direct translations. Consequently, test administrators may rely on fingerspelling, which introduces English influences and potentially alter test content. Additionally, many assessment tools have been developed by hearing individuals, prompting scholars to advocate for greater collaboration between sign language linguists, native Deaf signers, and test developers in order to improve the tests' validity and ability to serve Deaf, HoH, and disabled communities. Ultimately, educators require a diverse toolkit of assessments in order to properly evaluate ASL proficiency given the unique structure of the language.

- 3) Phonological fluency and expressive skill development in ASL**

Developing fluency in ASL is challenging due to the structure of ASL's morphology that differs fundamentally from spoken languages: namely, morphological structures are encoded simultaneously instead of sequentially [Paludneviciene et al., 2012]. Moreover, ASL involves the use of manual and non-manual articulators, converting multiple layers of meaning simultaneously [Paludneviciene et al., 2012]. Unlike spoken languages, ASL lacks a widely used writing system, making it difficult to capture essential linguistic features such as grammatical inflections, body movements, and effective information [Paludneviciene et al., 2012] [Quinto-Pozos, 2011]. A potential solution is exploring ASL writing systems, as reviewing signed language is significantly

harder than written language, even with the use of recording technologies [Quinto-Pozos, 2011].

While common nouns can be easily translated, complex inflected signs that encode information about direction, number, and subject-object relationships are difficult to represent in written form, which may create barriers for learners [Quinto-Pozos, 2011]. Certain ASL curricula still emphasize individual signs, reinforcing the misconception that ASL consists of signs structured according to English grammar, and that textbooks can be used as vocabulary lists instead of learning material [Quinto-Pozos, 2011].

Researchers suggest curricula to focus on classifiers and constructed action, and how to coordinate the two elements simultaneously [Quinto-Pozos, 2011]. Additionally, instructing iconicity in ASL may be helpful for L2 learners, but research is still needed to confirm effectiveness; similarly, fingerspelling, though often overlooked, may be useful in language development [Quinto-Pozos, 2011].

4) Systemic barriers in ASL education: Audism, linguicism, and lack of diversity

Audism, or discrimination against Deaf individuals, has impacted the structural inequalities in ASL education [Pirone et al., 2023]. Such systemic bias can be found in hiring practices, the classroom, and through institutional policy discriminatory against Deaf teachers and students [Pirone et al., 2023]. Deaf teachers have historically been overlooked for faculty positions within the hearing community despite having had more qualifications [Pirone et al., 2023]. Furthermore, Deaf students are also positioned in educational environments built predominantly for hearing students, limiting equal access to resources and professional advancement opportunities [Paludneviciene et al., 2012]. The result is a perpetual cycle whereby Deaf individuals remain underrepresented among the teaching faculty and leadership positions, further validating the notion that hearing teachers are more suitable for academic positions in ASL programs [Swaney and Smith, 2017].

Linguicism describes the discrimination against individuals or groups per their language, and it encompasses the preferential treatment of spoken languages over signed languages, pushing ASL further out of the academic arena. ASL was historically left out of general language courses, with the majority of universities placing it in the category of communication disorders rather than linguistics or foreign languages departments [Rosen, 2010]. This placement de-legitimizes ASL as an academic subject of study and hinders its access to grants and institutional support [Buisson, 2007]. The prejudice has structural issues that limit ASL's educational expansion and accreditation as an autonomous linguistic system. ASL instructor diversity continues to be a persistent issue, with faculty compositions predominantly white and hearing. Though ASL education has grown expo-

nentially, opportunities for Deaf instructors, particularly minority ones, are still lacking [Pirone et al., 2023]. Institutions typically point to a lack of Deaf professionals holding higher-level qualifications as the reason for underrepresentation among Deaf faculty members, yet there is very little investment in developing chances for Deaf scholars to gain these qualifications [Paludneviciene et al., 2012]. This underrepresentation not only affects employment equity but also the educational environment since students will have fewer opportunities to engage with diverse role models who can provide authentic cultural and linguistic insights. In addition, deaf students and faculty typically experience issues in obtaining necessary accommodations such as interpreters in faculty meetings and research seminars that further place them in exclusion from academic environments [Paludneviciene et al., 2012].

Overcoming such barriers requires active institutional change. Universities must also commit to the hiring of more Deaf educators and professional growth through mentorship programs, graduate school funding, and equitable hiring practices [Swaney and Smith, 2017]. ASL programs must also be situated within language or linguistics departments rather than in special education departments so that ASL is accorded the same respect and resources as other spoken languages [Pirone et al., 2023]. Institutions must also ensure accessibility and provide comprehensive accommodations for Deaf faculty and students to achieve a warm academic environment.

5) Technology in ASL instruction

Various technologies have been incorporated into ASL instruction to enhance accessibility and effectiveness without relying solely on traditional in-person instruction [Shao et al., 2020]. Video and digital video disc (DVD) recordings have long been used for ASL education, serving as instructional materials for both learning and assessment [Quinto-Pozos, 2011] [Thoryk, 2010]. Additionally, computer-based programs, such as a DVD program for learning Australian Sign Language (Auslan), highlight the importance of incorporating regional dialect variations into sign language instruction [Ellis et al., 2011]. While these resources provide valuable learning materials, they lack interactivity compared to more advanced technologies that enhance user engagement and experience. One notable advancement is Automatic Sign Language Recognition (ASLR), which has been used to develop tools such as SignQuiz, a quiz-based learning tool for fingerspelling in ISL (Indian Sign Language) [Joy et al., 2020]. Similarly, machine translation technologies have contributed to the development of 3D avatars capable of replicating facial expressions and movements, making sign language learning more accessible and immersive [De Martino et al., 2017] [Papastratis et al., 2021]. [Mehta et al., 2019] further expand on this concept by proposing an automated system for

generating 3D sign language video captions, showcasing how AI-driven tools can enhance ASL education.

Recent innovations involve wearable technology, such as smart glasses, which utilize augmented reality and sensor-based capturing to assist Deaf and Hard-of-Hearing students with lecture comprehension [Miller et al., 2017]. Additionally, gesture-capturing technologies, including Kinect and Leap Motion sensors, as well as data gloves, have been used to analyze and facilitate sign language learning [Papastratis et al., 2021]. These tools vary in effectiveness, with some prioritizing accuracy at the cost of computational complexity, while others enable real-time interaction but may lack precision.

Another emerging area is mixed-reality (MR) technology, which enhances ASL learning by incorporating real-time feedback and immersive experiences. Studies have demonstrated the benefits of interactive learning over passive approaches, emphasizing the need for further research into AI-driven ASL systems to integrate advanced feedback mechanisms [Shao et al., 2020].

In the status quo, challenges remain with integrating technology with ASL education. Machine learning-based Sign Language Recognition (SLR) is limited by the scarcity of large, diverse datasets, which affects both recognition accuracy and generalization abilities [Papastratis et al., 2021]. Sign Language Translation (SLT), which involves sequence-based ML algorithms, faces similar dataset limitations that hinder progress [Papastratis et al., 2021]. Despite providing significant potential, continued developments are necessary to overcome these limitations and create more effective, accessible, and interactive ASL learning tools.

III. METHODOLOGY

A. EDA & Dataset

In this project, we looked at two datasets, both having upwards of 2,000 classes. The first was *WLASL*, which was composed by Dongxu Li and Hongdong Li for the purpose of benefiting communication between deaf and hearing communities. From an environmental perspective, various backgrounds and lighting conditions are present in the *WLASL* dataset. Regarding the signers themselves, there are over 100 different individuals in the dataset, with each sign performed by at least three of them. Diversity among the signers is also significant, with clear variety in gender, age, and cultural representation. EDA revealed several underrepresented words with much fewer class instances. After creating a graph of the number of videos for each word, or the number of files within each sub-folder (based on the extraction and storage of the dataset), a bar graph displayed that several words had over 14 video examples, whereas the median number of videos per class was approximately four. Underrepresented words lead to the model being biased and less able to recognize those words due to less video data to draw upon and learn from. To remedy this problem, the *MoviePy* library was

used to traverse every folder with fewer than four videos and select one of the existing videos to randomly augment (flipping, rotating, changing the brightness, and cropping to a limited extent). We repeated this process until there were at least four videos in each folder (for every word). This process attempts to further enhance diversity into the model, increasing variation in its data to improve its recognition of these words.

The other dataset we looked into was *Microsoft ASL Citizen*, which was developed by Microsoft Research with the help of Boston University, University of Washington, and the Rochester Institute of Technology. It is the first crowdsourced and largest Isolated Sign Language Recognition dataset. Many videos in this dataset are filmed in candid conditions, enhancing the authenticity of the data. This dataset includes more diversity in that it includes signers of various minorities, including those in the Deaf community and those with disabilities. From an ethics perspective, this is a good dataset as explicit consent was received from every contributor.

Though the original goal of the project was to create an educational interface to assist learning signers with a multitude of words and phrases, it became clear that the ratio of videos per class to classes was incredibly small, even with data augmentation. To better comply with the data requirements of this project, the top five most populated classes in the *WLASL* and *Microsoft ASL Citizen* dataset were chosen as the subset to be used in the project. The words were *bite*, *dark*, *decide*, *demand*, *dog*. These words were assigned labels of 0, 1, 2, 3, 4, respectively.

B. Experiment Setup

We took on the challenge of training a model with video input. We felt that in signing everyday words and phrases, spacial and temporal features are best conveyed through videos. We initially started researching the I3D (Inflated 3D CNN) model for recognizing the user-performed signs [Haizhong, 2021]. This model is based on analyzing individual images via a 2D CNN architecture and extending it into a 3D CNN by capturing changes between individual frames. The process starts with taking in a video as the user input and dividing it into separate frames. It then examines the images using filters that slide over the height and width to detect objects by identifying changes in color from pixel to pixel, creating an outline for the figure. From there, the model compares the object positions from the previous frame to identify movement within the video. After completing this analysis, the program identifies which sign the user is performing. We sought to leverage a pre-trained model due to limitations in computational power. As well, prior research demonstrated promising results from taking a similar approach [Wong et al., 2022].

Unfortunately, despite our extensive research and high optimism surrounding this pipeline, the dataset size and

computational power required to successfully deploy this experiment was shown to be far beyond that available to us. In light of this, we decided to pivot to a new model with greater feasibility given our restrictions.

In shifting gears, we took a new common approach to video classification to execute the task at hand, consisting of a 2D CNN + Recurrent Neural Network (RNN). Through this architecture, the CNN learns spatial features of the video frames (images), and the RNN learns temporal features among the various frames. This process essentially simplifies a 3D problem into two simpler problems in 2D and 1D. A similar roadblock was once again encountered as our available computational power and resources did not allow for proper training on this model. As a result, we were unable to deploy it for our project. With that being said, a simplified version of this architecture was successfully implemented. Removing the LSTM, leaving the model as a 2D CNN proved to be an adequate classifier for this undertaking.

To preprocess this data, we created a dataframe to identify each video and its corresponding enumerated label. Each video was split into frames (30 frames per video). The set of frames for each video was manually analyzed, ensuring only relevant frames were kept. The frames after starting the video but before executing the signs as well as the frames after executing the signs before ending the video were omitted. The new sets of frames were then augmented through cropping, flipping, colour adjustment, and normalization. After that, the dataset was split into training and testing. For each video class, 80% of the videos (i.e., sets of frames) were placed into the training dataset, while the remaining 20% was used as testing data. This way, the train-test split stayed consistent at 80% to 20% on an overall basis and on a per-class basis. From there, the data was trained on four different pre-trained models commonly used for image classification: *ResNet50*, *InceptionV3*, *VGG16*, *MobileNetV3*. Layers were frozen with the exception of the last four so that the models could be fine-tuned on our datasets.

For the graphical user interface (GUI), we envisioned a simple yet efficient sign language recognition system designed to provide users with a clear and interactive experience for real-time interpretation. The interface includes key features such as a functional webcam for live classification, labels displaying predicted signs, and accuracy metrics to offer users feedback on their gestures. Our goal was to create an accessible online solution that is both intuitive and effective. Prioritizing simplicity and accessibility, we identified *Hugging Face* as a viable program. As an open-source platform, it enables seamless deployment of ML models in AI-driven applications.

C. Evaluation Methods

The models were evaluated using the metrics of accuracy, precision, recall, and F1. Confusion matrices were produced for each model to depict the class-by-class breakdown for the

predictions. Further data augmentation was performed as input for some of the models if problematic trends were apparent through the confusion matrix.

D. Data Availability

Links to Kaggle are included here.

- [WLASL Dataset](#)
- [Microsoft ASL Citizen Dataset](#)

Listed below, through Table I, are the results of the four CNN models used in this project.

TABLE I
RESULTS FROM THE EXPERIMENTS AGAINST THE TEST SET.

Model	Accuracy	Precision	Recall	F1
<i>ResNet50</i>	57.14%	60.20%	57.31%	58.11%
<i>InceptionV3</i>	59.18%	58.83%	58.40%	58.03%
<i>VGG16</i>	81.63%	84.86%	81.96%	81.25%
<i>MobileNetV3</i>	61.22%	64.49%	60.81%	61.31%

Pictured below, through Figure 1 and Figure 2, is the learning curve and confusion matrix of the top-performing CNN model used in this project: the *VGG16* pretrained model.



Fig. 1. Learning curve for the *VGG16* model.

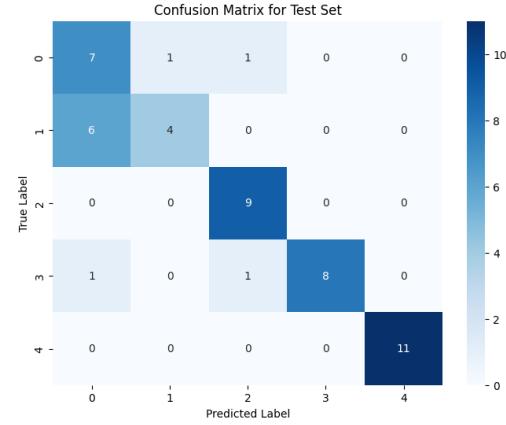


Fig. 2. Confusion matrix for the *VGG16* model.

E. Analysis

This project served as a great educational endeavour as it allowed the team to delve deep into the pipeline of complex computer vision projects and highlighted the significant tradeoffs that can make or break an ML project. Despite

the significant challenges faced throughout this process, all stemming from storage and computational power limitations, the team was able to adapt to unideal circumstances and create numerous working and well-performing classification models.

It was clear from the beginning that in taking on a project of this magnitude with the resources available, it would not be feasible to create a decently performing model from scratch. When leveraging the pretrained models, freezing layers and fine-tuning proved to be advantageous. These steps helped the models extract nuanced features of the sign gestures.

When analyzing the achieved results, interesting observations and takeaways are extracted. First, class 4, *dog*, is shown to have been best classified in all four models, as seen via the confusion matrices. This is due to data augmentation. When we first ran the models, class 4 underperformed relative to the other words. To combat this, we performed further augmentation to enhance the training data for that class. In running the models subsequently, the *dog* class proved to be extremely well-captured through the new, modified dataset.

It is also evident that the *VGG16* model outperformed the rest by a significant margin. This can also be attributed to further data augmentation. When we first ran this model, the trend visible through the confusion matrix was that class 1, *dark*, was heavily misclassified. As a result, further augmentation was performed on that class. When the model was executed again, it made virtually no errors, aside from further misclassifications of *dark*, but to a reduced extent than before. It showed numerous instances of predicting class 0 on data belonging to class 1. Though no further refinement to the data was subsequently done, effort in improving the model would entail better distinction between classes 0 and 1 to fix its one consistent mistake.

In analyzing the learning curves for the models in comparison to the metrics obtained when the models were run on the testing data, overfitting is observed. Further modifications to the models would try to address this issue through further optimization of hyperparameters, such as dropout, learning rate, batch size, and epochs.

The models perform slightly below the results achieved in the literature as similar projects obtain results upwards of 80% [Huang and Chouvatut, 2024], [Longlong et al., 2019]. Though we use a small number of classes, we are also limited on the training and testing data we have. In total, we use just under 250 videos, with less than 30 frames per video because manual denoising (i.e., removal of frames from before and after the signing gesture itself) resulted in the discarding of frames. We believe that the biggest limitation our models face is the size of the dataset and the quality of the data and models. The data does not lack quality from a diversity

standpoint, but rather in regards to its features. We believe that using a higher framerate would yield better results as there would be more data to train and test on, while the features of the video would be better extracted. As the LSTM is omitted from the architecture, the temporal element of the models is lost. To compensate, a higher framerate would allow for deeper feature extraction in the spacial dimension.

Overall, the models' performance is consistent with the sophistication of the dataset and architecture used, and provides promising insights into the capabilities of 2D CNNs to perform video classification.

F. Ethical Considerations

The ethical considerations surrounding this project surround bias and the effective, reliable use of AI-based ASL learning tools in educational contexts. A key ethical concern is bias with respect to some many words and signs being unrepresented. Thus, in implementing the models explored in this paper with a broad range of classes (words), there may be biases in the model's performance leading to decreased accuracy for words that are represented less. Even with data augmentation methods, biases may be present. The model's ability to generalize may also be skewed toward more frequently represented signs or groups, potentially leading to underperformance in recognizing signs or signs performed by underrepresented demographic groups.

Many of the ethical considerations of this project also speak to the narrative thematic analysis findings summarized in the Related Works section, underscoring the importance of developing a tool that can address the multifaceted limitations surrounding ASL education. Given that our tool is intended to be used in educational settings, particularly for beginners learning ASL, a concern that arises is inequity in accessing the tool. If our tool is inaccessible to those without reliable Internet connection, for instance, this may exacerbate existing disparities in education and communication accessibility for individuals who are HoH, Deaf, and may have intersectional identities such as being from low socioeconomic status, or resource-constrained communities.

Additionally, one substantial concern of our tool is that individuals may potentially overrely on the model for learning ASL, deprioritizing real-world interactions with individuals in applying their learnings. Thus, the broader ASL education system that incorporates AI should take into consideration the extent to which the recognition tool serves as a complement to learning ASL, encouraging collaboration between the learner and the technology, ensuring that the learner retains control over their own learning process.

The potential misuse of this tool, such as using CNNs to detect ASL in healthcare settings, also poses a significant challenge if the model miscommunicates ASL between patients and providers, for example. Human oversight would hence be critical to preventing harm. The key takeaway is that this tool should not be used in isolation in high-stakes decisions, but rather as an assistant to human expertise. Moreover, the tool

should primarily be used as an at-home supplement to ASL education, in addition to one's learning in real world contexts with other individuals.

IV. CONCLUSION

To conclude, the objective of this project was to leverage pretrained ML architectures to create a real-time sign language classification model for common ASL words and phrases. This was to be deployed through an interactive interface as an educational program for users to practice their signs as they start to learn ASL. The *ResNet50*, *InceptionV3*, *VGG16*, *MobileNetV3* models were tested using a combined dataset composed of videos from the *WLASL* dataset as well as the *Microsoft ASL Citizen* dataset. The models were trained and tested on five classes: *bite*, *dark*, *decide*, *demand*, *dog*. The *VGG16* model outperformed the rest, achieving accuracy of 81.63%, precision of 84.86%, recall of 81.96%, and F1 of 81.25%. The results are promising, showing potential for these models to achieve results in the literature.

Factors contributing negatively to the models' performance include overfitting and insufficient extraction of temporal data. With more time and computational power, proposed amendments include further data augmentation, implementation of the LSTM, and an increased frame sampling rate. Ultimately, this undertaking successfully highlighted the gaps in ASL education systems and proposed a working solution rooted in AI to combat these shortcomings.

V. FUTURE WORK

The evolving field of sign language recognition continues to call for opportunities for future advancements. Our findings emphasize the role of model optimization in improving recognition accuracy and real-time performance. To broaden the impact of the project, we seek to refine both the models and GUI for greater accessibility and usability. Enhancing adaptability will allow for more accurate recognition across various lighting conditions, skin tones, hand shapes, and signing speeds. Another addition to the project would be to modify the denoising algorithm. Manual denoising was an adequate solution to preprocess the training and testing videos, but is not doable in real-time implementation of the software. This step must be automated before launching this application for real-world use. From a GUI perspective, future implementations include additional interactive features, such as real-time feedback to offer gesture correction to users. Furthermore, it is crucial to implement Human-Computer Interaction (HCI) elements and interventions as potential means of promoting self-reflection and reducing bias on the part of the user, in addition to examining the means by which the model's datasets may embed biases. Thus, consideration of the ways in which the user interacts with the interface is crucial in understanding the many ways through which bias can be introduced in ML deployment. In consideration of HCI factors such as modifying the time it takes for AI feedback to display, providing gesture accuracy metrics and images, as well as the ways in which the AI is represented (e.g., symbol

or individual), the design may work better. Additionally, it is important to evaluate our model's performance by running confusion matrix metrics specifically for the model's ability to predict signs for racialized groups, signers with disabilities, and other underrepresented populations. Ultimately, our main focus remains to create a seamless platform that would assist in making sign language education and interpretation more accessible and user-friendly.

VI. LIMITATIONS

In addition to the computational limitations discussed above, there are several limitations to consider regarding the dataset. First, due to limited storage and memory as well as minimal samples per class, we worked with a small subset of the *WLASL* and *Microsoft ASL Citizen* datasets. As a result, the extensive diversity of these datasets was not entirely represented. If this program were to be used by signers of a group not represented in the data, or perhaps in a foreign environment, the models would be limited in performance. Overall, the smaller size and lesser diversity of the subset used decreases the generalizability of our models. While data augmentation can be performed for underrepresented features in the dataset, it is not the ideal solution, and does not override the fact that there are populations, words, and environments underrepresented in this dataset and many others alike.

VII. ACKNOWLEDGMENTS

The team would like to extend our sincerest thanks to Sarah Nassar, the director of design for the QMIND Healthcare node for her unwavering technical support throughout this project. The team would also like to extend our sincerest thanks to Paul Wu, the director of design for the QMIND AI Ethics node for his unwavering support throughout this project in helping with the research and ethical components of the deliverable.

REFERENCES

- [Buisson, 2007] Buisson, G. J. (2007). Using online glossing lessons for accelerated instruction in asl for preservice deaf education majors.
- [De Martino et al., 2017] De Martino, J. M., Silva, I. R., Bolognini, C. Z., Costa, P. D., Kumada, K. M., Coradine, L. C., Brito, P. H., Amaral, W. M., Benetti, A. B., Poeta, E. T., Angare, L. M., Ferreira, C. M., and De Conti, D. F. (2017). Signing avatars: making education more inclusive. volume 16, Berlin, Heidelberg, Springer-Verlag.
- [Desai et al., 2023] Desai, A., Berger, L., Minakov, F. O., Milan, V., Singh, C., Pumphrey, K., Ladner, R. E., Daumé III, H., Lu, A. X., Caselli, N., and Bragg, D. (2023). Asl citizen: A community-sourced dataset for advancing isolated sign language recognition.
- [Ellis et al., 2011] Ellis, K., Ray, N., and Howard, C. (2011). Learning a physical skill via a computer: a case study exploring australian sign language.
- [Haizhong, 2021] Haizhong, Q. (2021). I3dl an improved three-dimensional cnn model on hyperspectral remote sensing image classification.
- [Huang and Chouvatut, 2024] Huang, J. and Chouvatut, V. (2024). Video-based sign language recognition via resnet and lstm network.
- [Joy et al., 2020] Joy, J., Balakrishnan, K., and Madhavankutty, S. (2020). Developing a bilingual mobile dictionary for indian sign language and gathering users experience with signdict.
- [Li et al., 2020] Li, D., Opazo, C. R., Yu, X., and Li, H. (2020). Word-level deep sign language recognition from video: A new large-scale dataset and methods comparison. Snowmass, CO, USA. IEEE.
- [Longlong et al., 2019] Longlong, J., Vahdani, E., Huenerfauth, M., and Tian, Y. (2019). Recognizing american sign language manual signs from rgb-d videos.
- [Mehta et al., 2019] Mehta, N., Pai, S., and Singh, S. (2019). Automated 3d sign language caption generation for video.
- [Miller et al., 2017] Miller, A., Miller, A., Malasig, J., Castro, B., Hanson, V. L., Nicolau, H., and Brandão, A. (2017). The use of smart glasses for lecture comprehension by deaf and hard of hearing students.
- [Paludneviciene et al., 2012] Paludneviciene, R., Hauser, P. C., Daggett, D. J., and Kurz, K. (2012). Issues and trends in sign language assessment.
- [Papastratis et al., 2021] Papastratis, I., Chatzikonstantinou, C., Konstantinidis, D., Dimitropoulos, K., and Daras, P. (2021). Artificial intelligence technologies for sign language.
- [Pirone et al., 2023] Pirone, J. S., Pudans-Smith, K. K., Ivy, T., and Listman, J. D. (2023). The landscape of american sign language education.
- [Quinto-Pozos, 2011] Quinto-Pozos, D. (2011). Teaching american sign language to hearing adult learners.
- [Rosen, 2010] Rosen, R. S. (2010). American sign language curricula: A review.
- [Shao et al., 2020] Shao, Q., Sniffen, A., Blanchet, J., Hillis, M. E., Shi, X., Haris, T. K., Liu, J., Lamberton, J., Malzkuhn, M., Quandt, L. C., Mahoney, J., Kraemer, D. J. M., Zhou, X., and Balkcom, D. (2020). Teaching american sign language in mixed reality.
- [Swaney and Smith, 2017] Swaney, M. G. and Smith, D. H. (2017). Perceived gaps and the use of supplemental materials in postsecondary american sign language curricula.
- [Thoryk, 2010] Thoryk, R. (2010). A call for improvement: The need for research-based materials in american sign language education.
- [Wong et al., 2022] Wong, R., Camgöz, N. C., and Bowden, R. (2022). Hierarchical i3d for sign spotting.

An Application of Reinforcement Learning in Rocket League

Josh Albom
Queen's University
josh.albom@queensu.ca

Justin Badua
Queen's University
justin.badua@queensu.ca

Aayan Kader
Queen's University
aayan.a@queensu.ca

Chase Colby
Queen's University
22PLN@queensu.ca

Ethan Stassen
Queen's University
23JLV2@queensu.ca

Nicolas Raco
Queen's University
nicolas.raco@queensu.ca

Abstract—This paper presents the development of a reinforcement learning (RL) agent for Rocket League, aiming to achieve competitive, human-like gameplay. Building upon existing RL frameworks and Proximal Policy Optimization (PPO), we address limitations of prior agents by implementing a refined reward structure that balances offensive and defensive strategies, discrete action spaces for improved control precision, and enhanced observation processing for better spatial awareness. We utilize RLGym and RLBot frameworks for training and interaction, respectively. Our agent demonstrates superior performance against human players, achieving significant score disparities in controlled matches, showcasing advanced ball control, strategic decision-making, and effective execution of ground-based maneuvers. We discuss the agent’s architecture, training methodology, and performance metrics, highlighting its strengths in dribbling, flicking, and kickoffs. Limitations, such as the lack of opponent diversity during training and challenges with advanced aerial maneuvers, are also addressed. Future work focuses on enhancing reward functions, exploring alternative learning architectures, and optimizing environment interaction to further improve the agent’s competitive performance and strategic adaptability.

I. INTRODUCTION

A. Motivation

Reinforcement Learning (RL) is instrumental in the field of Artificial Intelligence (AI), particularly in complex decision making environments. Rocket League serves as an ideal environment to test and refine RL algorithms, due to its dynamics and physics-based gameplay. Notably, the development of Lucky-SKG, a RL agent for Rocket League, has demonstrated superior performance, outperforming top ranking bots like Necto and Nexto, while establishing new benchmarks in the game [1].

Despite advances in RL, academic and professional awareness remains limited. Highlighting successful applications of RL in popular games such as Rocket League can serve as an engaging way to educate and inform the public about the potential of RL. This application showcases the practical uses of RL, which additionally inspires further research and development in the field.

B. Related Works

Several reinforcement learning (RL) agents are developed for Rocket League, with notable contributions from Necto

and Nexto. These agents utilize the RLGym and RLBot frameworks alongside Proximal Policy Optimization (PPO) to train competitive bots. Their methodologies primarily focus on reward-driven policy optimization, allowing them to outperform traditional scripted bots. However, they struggle with fine motor control, adaptability to dynamic scenarios, and complex maneuvers such as aerials and flip resets.

More recently, Lucky-SKG demonstrated superior performance by improving policy optimization and reward structure. Its methodology integrates a more refined balance between offensive and defensive strategies while leveraging self-play for continuous improvement. Despite its advancements, challenges remain in improving strategic decision-making, especially in unpredictable in-game scenarios where human intuition often outperforms AI.

Our work builds upon these efforts by addressing these aspects of training. Our refined reward function, unlike previous approaches, often prioritizes goal-scoring at the expense of broader game play strategy. We introduce a balanced reward structure that accounts for positioning, boost management, and defensive plays. Enhanced Action Parsing: While prior agents rely on continuous action spaces, we implement discretization techniques to improve control precision, reducing erratic behavior. Using advanced observation processing, by altering how the agent perceives game state information, we improve spatial awareness and strategic adaptability. These enhancements contribute to a more competitive and adaptable AI agent, capable of executing high-level plays with greater efficiency. Future work explores further optimization strategies and alternative learning architectures to push the boundaries of RL performance in Rocket League.

C. Problem Definition

The problem we aim to solve is the development of a reinforcement learning (RL) agent capable of playing Rocket League at a competitive level while exhibiting human-level gameplay. Traditional scripted bots, as well as earlier RL-based agents, demonstrate strong decision-making in controlled environments but struggle with complex maneuvers, adaptability, and strategic decision-making in dynamic game

states. Additionally, imitation learning approaches, such as TensorBot and Levi, attempt to learn from human replays but face challenges due to data loss, inconsistency, and action dependencies. Formally, given a sequence of game states S_t , the goal is to determine an optimal policy (π^*) that maximizes the probability of winning the game while ensuring smooth and human-like gameplay. The optimization objective is given by:

$$J(\theta) = \mathbb{E}_{\tau \sim \pi_\theta} \left[\sum_{t=0}^T \gamma^t r_t \right] \quad (1)$$

Where θ represents the policy parameters, τ is a trajectory sampled from the policy, γ is the discount factor, and r_t is the reward function at time step t .

Balancing exploration and exploitation ensures the agent explores novel strategies while optimizing for performance. Handling continuous and discrete action spaces in Rocket League requires fine-grained control over movements, making action space discretization essential for improved stability. Adapting to dynamic opponents, unlike scripted AI, human players and self-improving bots present unpredictable behaviors, requiring real-time adaptability. By leveraging Inverse Dynamics Models (IDM) for improved action inference and refining reward structures to balance offensive and defensive playstyles, we aim to develop a robust, competitive RL agent that surpasses the performance of existing models while incorporating human-like strategic elements.

II. METHODOLOGY

A. Frameworks

To train the agent, we use the RLGym framework [2], allowing the agent to train in Rocket League as if it were an OpenAI Gym environment. The RLGym framework uses a plugin for Rocket League called Bakkesmod [3], increasing game speeds for training purposes. Furthermore, RLGym creates multiple instances of the game to run allowing for multiple agents to train simultaneously.

For the agent to interact with the game, we implement the RLBot framework [4]. The framework allows for the RL agent to connect to the game by providing an API. The framework provides the agent its action space and observation space, which are the actions the agent can take, and the respective information about the current state at the given time step. This enables the framework to let individuals play against the bot in Rocket League.

B. Optimizing Policy

To have the agent play at a competitive level, it requires a policy that maximizes the probability of winning the game. The policy is optimized using Proximal Policy Optimization (PPO)[cite]. The main idea of PPO is that the next policy is similar to the previous policy. As well, PPO uses generalized advantage estimation (GAE) to balance variance and bias.

Furthermore, PPO uses an entropy coefficient to balance between exploration and exploitation.

PPO updates policies via

$$\theta_{k+1} = \arg \max_{\theta} \mathbb{E}_{s, a \sim \pi_{\theta_k}} [L(s, a, \theta_k, \theta)] \quad (2)$$

Where $L(s, a, \theta_k, \theta)$ is

$$L(s, a, \theta_k, \theta) = \min \left(\frac{\pi_\theta(a|s)}{\pi_{\theta_k}(a|s)} A^{\pi_{\theta_k}}(s, a), g(\epsilon, A^{\pi_{\theta_k}}(s, a)) \right) \quad (3)$$

The hyperparameters the agent uses to train are displayed in Table I. One hyperparameter is changed during training, the learning rate, which is reduced as training continues.

Hyperparameter	Value
PPO Batch Size	50,000
Timesteps per Iteration	50,000
Experience Buffer Size	150,000
PPO Minibatch Size	50,000
PPO Entropy Coefficient	0.01
PPO Epochs	2
Standardize Returns	True
Standardize Observations	False
Save Every Timesteps	100,000
Timestep Limit	10^{20}
Policy Learning Rate	2×10^{-4}
Critic Learning Rate	2×10^{-4}
Policy Layer Sizes	(1024, 1024, 1024, 1024, 512)
Critic Layer Sizes	(2048, 1024, 1024, 1024, 512)

TABLE I
HYPERPARAMETER SETTINGS

C. Observation Space

At each time step, the agent receives 89 inputs about the game that contain physical information about the player and the ball, as well as the information about boost pads. RLBot and RLGym provide these observations and scale them from [-1, 1] before feeding this information into the neural network.

The game's information passes through a series of arrays of data fed from the game's memory. This allows the agent to know the environment information at each timestep and enables the computational costs to be lower than feeding images into the agent.

D. Action Space

The action space in Rocket League is made up of 5 continuous actions ranging in the interval [-1, 1], and 3 boolean actions (Table II). However, continuous actions in RL make it hard for the agent to control the car. Therefore, we turn the continuous actions into discrete actions by making bins, whereby the agent can select a discrete value to throttle, steer, and roll, allowing the agent to have better control over the car.

E. Neural Network Architecture

The agent uses a feedforward network that is trained using PPO in an actor-critic framework. The network receives the 89 inputs from the observation space, which are preprocessed and converted into PyTorch tensors. The tensors are then

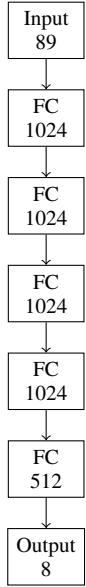
Action	Domain
Throttle	$\{-1, 0, 1\}$
Steer	$\{-1, 0, 1\}$
Pitch	$\{-1, 0, 1\}$
Yaw	$\{-1, 0, 1\}$
Roll	$\{-1, 0, 1\}$
Jump	$\{0, 1\}$
Boost	$\{0, 1\}$
Handbrake	$\{0, 1\}$

TABLE II
ACTION SPACE FOR LOOKUPACTION

Reward Name	Weight
Goal	300
Demo	5
VelocityBallToGoalReward	1.1
FaceBallReward	0.08
TouchVelReward	1
JumpTouchReward	0.3
SpeedTowardBallReward	0.2
AlignBallGoal	0.4
LiuDistanceBallToGoalReward	0.4
LiuDistancePlayerToBallReward	0.3
FlipResetReward	100

TABLE III
REWARD WEIGHTS FOR DIFFERENT EVENTS

fed through four fully connected layers with ReLU activation functions. The final layer of the network outputs discrete logits corresponding to different control actions (such as throttle, steer, jump, boost, etc.). These logits are then adjusted for uniformity, where padding is applied as needed, ensuring that all action dimensions are compatible.



F. Reward Weights

For the agent to learn an optimal policy, reward functions are used to speed up learning. Table III shows the rewards that the agent can earn with their associated weights. The rewards given to the agent are either sparse or dense, with sparse rewards being received only when a certain condition is met, whereas dense rewards are given at each timestep. Each time step, the reward is multiplied by its associated weight to obtain the total reward for the agent at that time step.

G. Exploration

We allow the agent to explore the environment in two ways. The first being the entropy coefficient, during the entire training process, the coefficient is set to 0.01, allowing the bot to occasionally explore new gameplay options. In addition, the agent's state is randomized to increase the amount of scenarios the agent is in. Instead of always being placed in the default kickoff position, eighty percent of the time the agent and ball are put in a random spot on the field with a random velocity.

III. RESULTS

A. Milestones

During the initial training phase, the agent was set up to learn how to jump and touch the ball, both key parts of playing Rocket League. This laid the groundwork to later perform more complex actions. Once the agent became proficient in jumping and ball touches, the training shifted from jumping and ball touches, to pushing the ball towards the net and accurate goals. As the agent became more accurate, it was encouraged to then put more force into the ball to enhance offensive capabilities. Building on these foundations, advanced techniques could now be introduced such as flicks and dribbles. Flicking allowed the agent to redirect the ball in mid air to perform advanced shots. Dribbling gives the agent the ability to quickly carry the ball into the net. Each of these milestones greatly contributed to the agent's overall skill set, paving the way for complex and strategic behavior in the field.

B. Strategy

The agent demonstrates an impressive range of strategies, particularly excelling in dribbling and executing flicks at close range to enhance its goal-scoring ability. Without direct input, the agent effectively implements these techniques, showcasing advanced ball control and offensive decision-making. Furthermore, the agent consistently performs well in kickoffs, mirroring the approach of professional players in one-on-one scenarios. While it does not master aerial attacks, it efficiently executes flips, controlled shooting, and ball receptions from high touches to transition into dribbles and scoring opportunities. On defense, the agent exhibits patience by waiting for shots and strategically positioning itself, making it highly effective in maintaining control over matches. During testing, the agent remained undefeated against all human challengers, demonstrating its ability to outmaneuver opponents through refined strategy and adaptability.

The agent also displays a fundamental understanding of boosting and flipping into the ball, effectively using boost to return to defensive positions or advance offensively while maintaining ball control. Its well-trained kickoff approach remains consistent regardless of positioning, leading to a high success rate in gaining possession and frequently converting kickoffs into goals.

C. Scores

During our conference showcase, we conducted a series of eight matches between a human player and our reinforcement learning-based Rocket League agent. The recorded scores, presented as human score vs. bot score, are as follows:

- Match 1: 1–29
- Match 2: 1–17
- Match 3: 1–20
- Match 4: 2–20
- Match 5: 0–24
- Match 6: 1–18
- Match 7: 0–7
- Match 8: 0–37

A detailed examination of these results reveals a significant performance disparity between the human player and the bot. In most matches, the human goal tally is extremely low compared to that of the bot. For example, in Match 1, the human scored only one goal versus the 29 goals of the bot. This yields a human-to-bot score ratio of approximately 0.0345, or 3.45%, when calculated as:

$$\text{Ratio} = \frac{\text{Human Score}}{\text{Bot Score}} = \frac{1}{29} \approx 0.0345$$

Similarly, Match 4 represents the best relative performance for the human, where a score of 2 against 20 results in a ratio of 0.1 or 10%. In contrast, Matches 5, 7, and 8 recorded no goals for the human, corresponding to a 0% ratio.

When summing the performance across all matches, the total goals scored were 6 for the human and 172 for the bot. This cumulative data gives an overall human scoring percentage calculated by:

$$\text{Overall Human Percentage} = \frac{6}{6 + 172} \approx \frac{6}{178} \approx 3.37\%$$

These numbers highlight that, on average, the human contributed only about 3.37% of the total goals across all matches.

D. Limitations

Several limitations highlight areas for further improvement. One significant limitation is the lack of opponent diversity during training. During training, the agent was only trained against the current version of itself, this may have restricted its ability to generalize effectively against more advanced strategies. Computational constraints also presented a challenge, limiting the extent of experimentation with different reward functions, hyperparameters, and training durations. Another notable limitation is the agent's inability to execute advanced aerial maneuvers such as aerial shots, flip resets, and ceiling plays. Although the agent demonstrated strong ground-based mechanics, including dribbling, flicks, and powerful shots, it struggled to take advantage of airborne plays.

IV. CONCLUSION

The investigation into integrating artificial intelligence into Rocket League entailed developing and training an AI agent using reinforcement learning techniques. Our work involved utilizing a simulation environment (RLGym) that replicates key elements of Rocket League gameplay, implementing a reward function tailored to promote desirable agent behaviours, and fine-tuning the agent using PPO to achieve optimal performance. Through iterative experimentation with unique rewards, a baseline from which the agent could learn fundamental actions was established. These actions include tracking the ball, jumping, using boost, striking the ball, and most importantly, getting goals.

Looking forward, several avenues for further development emerge. Enhancing the reward function allows the agent to handle more advanced strategies and make more intelligent in-game decisions to create a more sophisticated play style. Experimenting with more additional sub-rewards, such as positioning, defense, and goaltending, would help make the agent play more strategically.

Increasing the training process by adding the number of parallel environments and challenging the agent against its previous versions, can allow stronger adaptability and more efficient development. Additionally, by having agents play against each other (themselves) and retaining the winning strategies, this will allow the agent to improve continuously. This gradual progression from simpler scenarios to more complex challenges may be beneficial for achieving a higher level of competitive performance.

Another direction for further development is optimizing how the agent perceives and interacts with the environment. Developing custom observation parsers that extract richer, more detailed state information, such as spatial-temporal features and complex game dynamics, could enable a deeper understanding of the agent's surroundings. Additionally, exploring different action parsers that provide a more continuous and flexible control mechanism may allow the execution of more precise maneuvers, such as pinches and flip resets.

Ultimately, future research should aim to identify new reinforcement learning approaches that enhance both learning efficiency and practical in-game performance, with the understanding that these refinements may yield unexpected insights beyond the scope of the current study.

REFERENCES

- [1] V. Moschopoulos, P. Kyriakidis, A. Lazaridis, and I. Vlahavas, “Lucy-skg: Learning to play rocket league efficiently using deep reinforcement learning,” 2023. [Online]. Available: <https://arxiv.org/abs/2305.15801>
- [2] R. Developers, “Rlgym: A reinforcement learning environment for rocket league,” 2025, accessed: 2025-03-15. [Online]. Available: <https://rlgym.org/>
- [3] B. Developers, “Bakkesmod: The all-in-one mod for rocket league,” 2025, accessed: 2025-03-15. [Online]. Available: <https://bakkesmod.com/index.php>
- [4] RLBot, “Rlbot: Reinforcement learning for rocket league,” 2025, accessed: 2025-03-15. [Online]. Available: <https://rlbot.org/>

Art Suggester AI: The Art Recommendation Tool

Iain Macdonald
McMaster University
macdoi5@mcmaster.ca

Harrison Johns
McMaster University
johnsh32@mcmaster.ca

Angela Fernando
McMaster University
fernna61@mcmaster.ca

Ryan Brubacher
McMaster University
brubachr@mcmaster.ca

Johann Caancan
McMaster University
caancanj@mcmaster.ca

Aiden Henderson
McMaster University
hendea17@mcmaster.ca

Stefan Risca
McMaster University
riscas@mcmaster.ca

Abstract—Art Suggester is an AI-based art recommendation tool designed to inspire artists and enthusiasts by providing personalized art suggestions based on their available materials. The system uses a Convolutional Neural Network (CNN) to classify art mediums (e.g., paint, pencil crayons, markers) and a color detection algorithm to identify dominant colors in user-uploaded images. These inputs are used to recommend artworks from a curated database, making art discovery more accessible and engaging. Our CNN model achieved an accuracy of 95.12%, demonstrating its effectiveness in medium classification. The project highlights the potential of AI to bridge the gap between art creation and appreciation, offering a creative tool for artists and a learning resource for beginners. Code and resources are available at <https://github.com/McMasterAI2024-2025/ArtSuggesterAI>.

I. INTRODUCTION

Art Suggester addresses the challenge of artists having materials but lacking inspiration. By leveraging AI, the tool provides personalized art recommendations based on the materials and colors available to the user. This solution is significant because it enhances creativity, simplifies art discovery, and makes art more accessible to a broader audience [1]. Previous research has explored AI-driven art classification [2] and generative models [3], but few systems integrate medium identification and color analysis for tailored recommendations. Our approach combines CNN-based medium classification with color detection to offer a unique solution.

A. Motivation

Recent advancements in AI, particularly in CNNs and generative models, have made AI-driven art tools increasingly relevant [4]. CNNs having the ability to be trained on smaller datasets have improved image classification accuracy, enabling precise medium detection [5]. The global AI art market, valued at \$3.2 billion in 2024, is projected to grow to \$40.4 billion by 2033 [6], highlighting the growing interest in AI applications for creativity. However, existing tools often focus on style classification or generative art, leaving a gap for systems that integrate medium identification and color analysis. Art Suggester fills this gap by providing a tool that inspires artists and simplifies art discovery.

B. Related Works

Previous research has explored AI-driven art classification and recommendation systems. For example, CNNs have been

used for style and medium classification, while collaborative filtering has been applied for personalized recommendations. Generative models like DALL·E and Stable Diffusion have also gained popularity for creating art from text prompts [1]. However, these approaches often lack integration of medium and color analysis, which Art Suggester addresses.

C. Problem Definition

The Art Suggester faces challenges such as ensuring data quality and diversity for accurate medium detection, as maintaining model performance across varying image conditions (e.g., lighting, backgrounds) is difficult. We also need to have a variety of possible art recommendations with different styles and colours. Designing a user-friendly interface for the Art Suggester involves creating a secure user account system that caters to both artists and non-artists, adding complexity to the development process. The interface must feature intuitive navigation, ensuring users can easily upload images, view recommendations, and favourite art pieces. Balancing these challenges while delivering an accessible tool is key to the project's success.

II. METHODOLOGY

A. Data

We used a dataset comprising images from Google Images and custom photos of art materials (e.g., paint, pencil crayons, markers) taken under various lighting conditions. For recommendations, we utilized Kaggle datasets such as "Surreal Symphonies" [7] and "Portrait Paintings." [8] The data was preprocessed to 256x256 resolution and augmented to improve model robustness.

B. Model & Approach

We implemented a CNN using TensorFlow for medium classification. The model includes convolutional layers, dropout layers, and a softmax activation for classification. It was trained for 16 epochs with a validation split of 0.1 and a batch size of 16. We used the Adam optimizer with a learning rate of 0.001 and sparse categorical crossentropy as the loss function. For our color detection algorithm, we used the Python Imaging Library to detect color clusters and compare them to a predetermined list of colors, which were

also used to categorize our art pieces. Our database for users login information and favourite images were stored using MongoDB.

C. Evaluation

We evaluated the Art Suggester model using accuracy as the primary metric, measuring the percentage of correctly classified images, and employed confusion matrices to analyze performance on both a 10% test set and completely new images. This approach helped us assess the model's generalization ability and identify areas where it struggled, such as distinguishing between specific classes like paint, pencil crayons, and markers. Validation was conducted using a 10% test set and additional unseen images to ensure accuracy. Challenges included ensuring dataset diversity to account for varying lighting conditions and backgrounds. As well as, addressing overfitting (memorizing training data rather than generalizing the patterns) through dropout layers and early stopping; and handling real world variability, such as low-quality images from user devices with poor lighting or resolution. We also did create models for a variety of epoch counts to compare them.

TABLE I
COMPARISON OF ACCURACY ACROSS DIFFERENT EPOCH COUNTS

Epochs	Training Accuracy	Unseen Data Accuracy	Loss
5	99.23%	11/14	0.0209
10	100%	11/14	0.00073
15	95.12%	12/14	0.0012
20	98.7%	9/14	0.004
25	99.7%	11/14	0.006

Our early stopping was based on if the loss would not change over around 10 epochs. We did this to reduce chances of overfitting. The early stopping would often stop at 15 epochs, along with the fact it was best at classifying completely unseen data with bad framing or resolution we decided to use 15 epochs for our final product. For our color detection, we achieved an accuracy of up to 85% for each uploaded photo. To improve the user experience and avoid potential mistakes from the algorithm, we added an option for users to manually adjust their color preferences before art pieces are suggested.

III. RESULTS

The CNN model achieved an accuracy of 95.12%, demonstrating its effectiveness in classifying art mediums. The color detection achieved an accuracy of up to 85% in detecting the medium's colors. The confusion matrix (a graph to show model medium predictions compared to the actual medium) showed 174 correct classifications out of 189 test cases. The system successfully integrated medium classification and color detection to provide personalized art recommendations, outperforming traditional approaches in handling diverse art styles and simplicity of exploration.

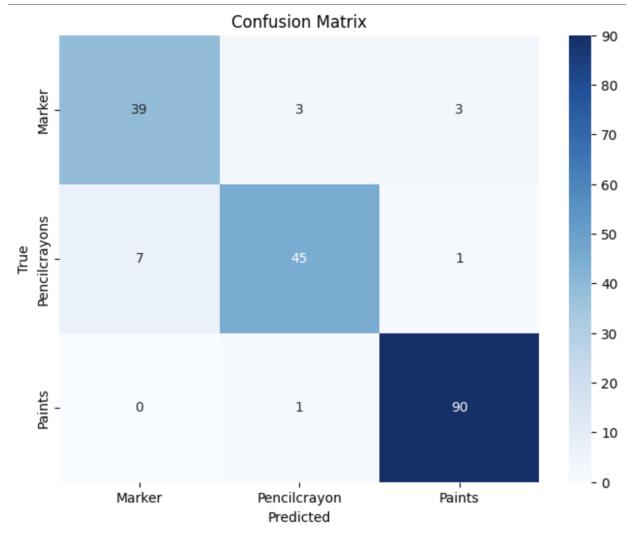


Fig. 1. Confusion Matrix of Test Data

IV. CONCLUSION

Art Suggester demonstrates the potential of AI to enhance art discovery by combining medium classification and color detection. The project achieved high accuracy and provided a user-friendly interface for personalized recommendations. Future work includes expanding the dataset, adding user feedback mechanisms, and optimizing the system for mobile devices. The tool has applications in art education, galleries, and interior design, making art more accessible and engaging for everyone. A detailed process flow diagram is below:

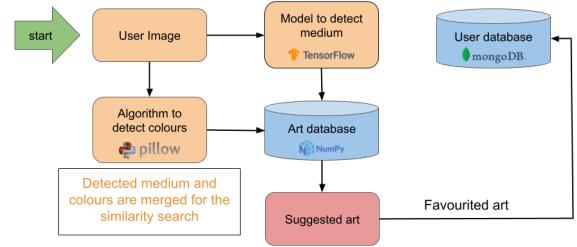


Fig. 2. Art Suggester AI Flow Diagram

REFERENCES

- [1] E. Zhou and D. Lee, "Generative artificial intelligence, human creativity, and art," *PNAS Nexus*, vol. 3, no. 3, p. pgae052, 2024.
- [2] E. Cetinic, T. Lipic, and S. Grgic, "Fine-tuning convolutional neural networks for fine art classification," *Expert Systems with Applications*, vol. 114, pp. 107–118, 2018.
- [3] S.-C. Necula and V.-D. Pavăloaia, "Ai-driven recommendations: A systematic review of the state of the art in e-commerce," *Applied Sciences*, vol. 13, no. 9, p. 5531, 2023.
- [4] B. Liu, "Arguments for the rise of artificial intelligence art: Does ai art have creativity, motivation, self-awareness and emotion?" *Art, Individual and Society*, vol. 35, no. 3, pp. 811–822, 2023.
- [5] J. G. et al., "Recent advances in convolutional neural networks," *Pattern Recognition*, vol. 77, pp. 354–377, 2018.

- [6] Shalwa. (2024) How large is the global ai in the art market? [Online]. Available: <https://artsmart.ai/blog/ai-in-the-art-market-statistics/>
- [7] C. 1702. (2023) Surreal symphonies (a dataset of diverse art). [Online]. Available: <https://www.kaggle.com/datasets/cyanex1702/surreal-symphonies-a-dataset-of-diverse-art>
- [8] D. Chakraborty. (2021) Portrait paintings. [Online]. Available: <https://www.kaggle.com/datasets/deewakarchakraborty/portrait-paintings>

Automated Road Damage Detection and Interactive Mapping Using YOLOv11, YOLOv12, and DeepSORT

Thomas Dermengea
Queen's University
22gq34@queensu.ca

Edan Kroi
Queen's University
22qds@queensu.ca

Ethan Solnik
Queen's University
22kg37@queensu.ca

Scott Doggett
Queen's University
20sjd7@queensu.ca

Anthony Grecu
Queen's University
22rpv4@queensu.ca

Kiarash Soleimaniroozbahani
Queen's University
21ksr5@queensu.ca

Abstract—Road damage significantly impacts infrastructure efficiency, road safety, and maintenance budgets. This paper introduces an automated road damage detection and visualization system employing advanced deep learning models, specifically YOLOv11 and YOLOv12 [1], [2], with ongoing testing of DeepSORT [3] for improved detection tracking across video frames. Utilizing a Raspberry Pi [6] equipped with a camera and GPS module, synchronized video and GPS data are captured. Data is uploaded to a Node.js [8] and Next.js [9] web platform for processing, resulting in an interactive, color-coded map allowing detailed damage analysis and route navigation based on road damage data. Our model achieves a mean Average Precision (mAP) of 54%, indicating significant practical applicability.

I. INTRODUCTION

Automated detection of road damage, including potholes and surface cracks, is essential to maintaining road safety, reducing maintenance costs, and enhancing transportation efficiency. Traditional manual inspections are costly, slow, and error-prone. This research presents a comprehensive automated approach leveraging state-of-the-art deep learning models, YOLOv11 and YOLOv12 [1], [2], combined with the DeepSORT tracking algorithm [3], currently under evaluation.

II. METHODOLOGY

A. Datasets and Model Development

The models were trained and evaluated on datasets from the Canadian Road Damage Detection Challenge (CRDCC 2022) [4] and the IEEE Big Data Cup 2022 [5]. YOLOv11 and YOLOv12 models were selected for their accuracy and efficiency in real-time detection scenarios. Training utilized Google Colab Pro+ [7] with NVIDIA A100 GPUs, significantly accelerating training times and enabling extensive hyperparameter tuning. Data augmentation techniques, including random cropping, rotations, brightness adjustments, and scaling, were employed to enhance model robustness across varying road conditions and lighting scenarios.

B. Hardware Setup for Data Collection

A Raspberry Pi [6] equipped with a high-resolution camera and GPS module was mounted on a vehicle for field data collection. Synchronized video footage in MP4 format and

timestamped GPS coordinates were captured and logged, enabling accurate geospatial tagging of detected road damage.

C. Processing Workflow

The system's operational workflow includes:

- 1) **Upload Interface:** Users upload recorded MP4 videos and corresponding GPS text logs via a custom-built web interface.
- 2) **Damage Detection and Tracking:** Uploaded videos undergo inference using YOLOv11 and YOLOv12 models [1], [2]. DeepSORT tracking is actively tested to improve tracking accuracy across video frames [3].
- 3) **Output Integration:** Detection results are combined with GPS data into structured CSV files, containing detailed timestamps, locations, and damage classifications.
- 4) **Visualization and Navigation:** CSV data generates an interactive, geospatial map displaying road damage markers color-coded by severity. Users can interact with the map to obtain detailed information and utilize navigation features, allowing route optimization based on preferences to either avoid or intersect damaged roads.

D. Web Platform and Storage

The web application, developed with Node.js [8] and Next.js [9] frameworks, offers an intuitive user interface, efficient processing pipeline, and robust data management facilitated by Azure Blob Storage [10]. This cloud backend ensures secure, scalable storage of video files, GPS data, processed outputs, and visualization results.

III. APPLICATIONS AND USE CASES

The developed system addresses multiple practical scenarios:

- **Municipal Road Inspection:** Automating routine road condition assessments to maintain safety.
- **Road Maintenance Planning:** Optimizing resource allocation by mapping damaged road sections.
- **Autonomous Vehicle Navigation:** Enhancing vehicle route safety by providing detailed road condition data.

- **Insurance Assessment:** Providing visual evidence of road conditions to support claim processing.
- **Public Safety Enhancement:** Identifying and repairing hazardous sections to reduce accidents.
- **Logistics Optimization:** Rerouting shipments to avoid damaged roads, minimizing transportation costs.
- **Infrastructure Development:** Assessing road quality pre- and post-construction for improved durability.
- **Environmental Impact Analysis:** Studying the influence of weather and disasters on road conditions.
- **Disaster Recovery:** Prioritizing road repairs after natural disasters to quickly restore essential transportation routes.
- **Low-Clearance Vehicles:** Allowing low-clearance vehicles to select routes that avoid severe road damage and speed bumps.

IV. EXPERIMENTAL RESULTS

Our system achieved a mean Average Precision (mAP) of 54%, demonstrating effective real-world applicability in diverse environmental conditions. The ongoing integration of DeepSORT is expected to further improve detection accuracy by enhancing anomaly tracking across consecutive video frames.

A. Model Evaluation

Figures 1 and 2 illustrate confusion matrices for YOLOv12-I and YOLOv12-s models, respectively, detailing true positives, false positives, and class misclassifications. Figure 3 displays training and validation loss curves along with precision and recall metrics, highlighting training progress and model performance.

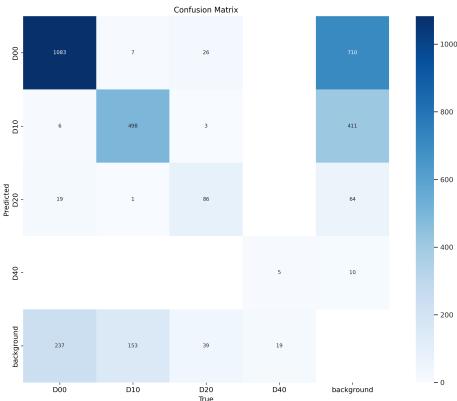


Fig. 1. Confusion Matrix for YOLOv12-I Model

V. CONCLUSION AND FUTURE WORK

This study successfully integrates advanced object detection methods, effective tracking algorithms, and practical visualization and navigation tools into a cohesive system for road damage detection and management. Future enhancements include refining the DeepSORT integration, improving detection accuracy in challenging environmental scenarios, and further optimizing model efficiency for edge device deployment.

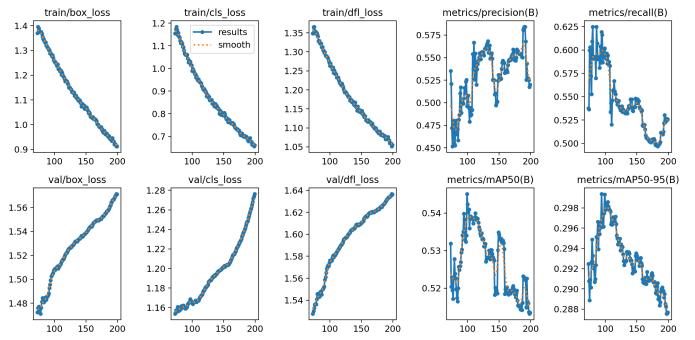


Fig. 2. Training and Validation Metrics

REFERENCES

- [1] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779–788.
- [2] A. Bochkovskiy, C. Wang, and H. Y. M. Liao, “YOLOv4: Optimal Speed and Accuracy of Object Detection,” *arXiv preprint arXiv:2004.10934*, 2020.
- [3] W. Wojke, N. Bewley, and D. Paulus, “Simple Online and Realtime Tracking with a Deep Association Metric,” in *Proc. IEEE Int. Conf. on Image Processing (ICIP)*, 2017, pp. 3645–3649.
- [4] “Canadian Road Damage Detection Challenge (CRDDC) 2022,” [Online]. Available: <https://example.com/crddc2022>.
- [5] “IEEE Big Data Cup 2022,” [Online]. Available: <https://example.com/ieeebigdatacup2022>.
- [6] “Raspberry Pi Documentation,” [Online]. Available: <https://www.raspberrypi.org/documentation/>.
- [7] “Google Colaboratory,” [Online]. Available: <https://colab.research.google.com/>.
- [8] “Node.js,” [Online]. Available: <https://nodejs.org/>.
- [9] “Next.js,” [Online]. Available: <https://nextjs.org/>.
- [10] “Microsoft Azure Blob Storage,” [Online]. Available: <https://azure.microsoft.com/en-us/services/storage/blobs/>.

BOLLD: Body and Oral Language Learning Decoder

Nicole Sorokin
McMaster University
sorokinn@mcmaster.ca

Zuhair Qureshi
McMaster University
quresz23@mcmaster.ca

Julia Brzustowski
McMaster University
brzustoj@mcmaster.ca

Grady Rueffer
McMaster University
ruefferg@mcmaster.ca

Sophia Shanharupan
McMaster University
shants5@mcmaster.ca

Abstract—Detecting threatening behavior remains a significant challenge in security and public safety. BOLLD is a multimodal threat detection approach that combines body language analysis, lip reading, and reinforcement learning to assess potential malicious behaviour in real-time. Using MediaPipe for skeletal tracking, a Random Forest classifier and a modified LipNet model, the system evaluates both physical and verbal cues to improve detection accuracy. In testing, BOLLD significantly improved its performance, demonstrating its potential for security applications in environments where audio is unreliable as well as aid individuals with visual impairments by enhancing situational awareness. The project is available at github.com/McMasterAI2024-2025/BOLLD

I. INTRODUCTION

Detecting threatening behavior is a key challenge in security and public safety, but most existing solutions focus on either physical actions or verbal communication rather than both. This project introduces BOLLD (Body and Oral Language Learning Decoder), a system that combines body language analysis and lip reading for real-time threat detection. It uses MediaPipe for skeletal tracking and facial landmark detection, a Random Forest classifier to categorize body poses, and a modified LipNet model to analyze spoken words for potential dangerous actions. A reinforcement learning component further refines detection by integrating physical and verbal cues. This multimodal approach improves real-time processing and could be useful in environments where audio is unreliable, as well as in assistive technology for visually impaired individuals.

A. Motivation

BOLLD takes a multimodal approach to real-time maliciousness detection by combining computer vision, natural language processing, and reinforcement learning. This research is particularly relevant today as AI advancements shape public safety and security measures.

By integrating body language analysis, lip reading, and reinforcement learning, BOLLD detects threats in environments where audio may be unavailable or unreliable, aligning with AI's growing role in cybersecurity and physical security.

Research on AI-powered multimodal search engines demonstrates the effectiveness of combining text, images, audio, and video for situational awareness [1]. Similarly, BOLLD merges visual and verbal cues to assess alarming actions.

Natural language processing (NLP) has proven valuable in Cyber Threat Intelligence by automating large-scale dataset analysis to identify malicious activity [2]. BOLLD extends this by applying NLP to lip-transcribed speech, detecting verbal threats without audio.

Recent studies highlight the predictive potential of AI and NLP in cybersecurity threat detection [3]. These technologies can identify risks early, and BOLLD adapts this capability for physical threat detection by analyzing both body language and spoken content.

Additionally, research into AI-driven tracking and real-time detection in cybersecurity [4] provides a framework that parallels BOLLD's multimodal strategy for identifying physical threats.

By building on these advancements, BOLLD enhances public safety, particularly where audio-based violence detection is ineffective.

B. Related Works

Recent research in multimodal AI has made significant progress in threat detection, particularly in cybersecurity and public safety. This section explores related work and how BOLLD contributes to addressing some of the existing challenges.

Multimodal AI systems integrate different types of data, such as text, images, and geospatial information, to improve the identification of suspicious actions [5]. Large language models (LLMs) like ChatGPT and Gemini have also embraced multimodality, processing and reasoning across text, images, and even audio inputs. By combining multiple sources, these models overcome the limitations of traditional methods, making real-time decision-making more reliable.

AI-driven multimodal search engines have also been explored for cybersecurity applications. These systems use machine learning to analyze security threats from multiple perspectives, but challenges remain in refining their accuracy and efficiency [1].

One recent development is FIRE, a framework designed for few-shot inter-domain threat detection using large-scale multimodal pre-training [6]. This approach helps detect hostility in complex network environments with minimal labeled data, addressing a key issue in cybersecurity.

However, there are still significant challenges in this field, including processing multimodal data in real time, balancing

accuracy with computational efficiency, ensuring privacy in surveillance applications, and adapting to evolving aggressive patterns. This integration allows for real-time threat detection in scenarios where audio may be unreliable or unavailable, expanding its potential applications across various security contexts.

C. Problem Definition

Most threat detection systems focus either on physical actions or speech, but rarely consider the connection between body language and spoken words. This gap can make them less effective in real-world situations where audio is unreliable or unavailable, such as in noisy environments, security footage without sound, or meetings where microphones fail.

BOLLD is designed to address these challenges by combining computer vision-based body language recognition with real-time lip reading. By analyzing skeletal motion alongside transcribed speech, BOLLD aims to improve the accuracy of identifying malicious actions. It also incorporates reinforcement learning to refine its predictions over time. Given a sequence of upper body movements and lip motions (X), the system predicts a threat/violence score (y), adjusting dynamically based on behavioral patterns.

By linking physical and verbal cues, BOLLD could be useful in security applications where audio isn't available and in assistive technology for visually impaired individuals who rely on real-time alerts through wearable devices. This approach offers a step toward more adaptive and effective multimodal violence detection, addressing the limitations of systems that rely on a single data source.

II. METHODOLOGY

This section outlines the approach used to develop BOLLD, a system that combines body language analysis and lip reading for real-time threat detection. The process includes data collection, model training, evaluation, and continuous refinement to improve accuracy.

BOLLD is built using Streamlit and integrates AI models for analyzing both body language and lip movements in live video. Figure 1 illustrates the data flow. When the system is activated, it initializes session state variables to track violence levels, actions, rewards, and video frames. It also loads pre-trained models for body language recognition and lip reading.

The system processes live video input, extracting facial and body landmarks using MediaPipe and dlib for feature normalization. Lip movements are detected, converted to grayscale, and passed through a lip-reading model that transcribes speech roughly every 75 frames. A predefined dictionary assigns violence scores to transcribed words, and a reinforcement learning (RL) agent determines an action, either “all good” or “de-escalate,” based on detected danger levels. The RL model continuously updates its Q-table by evaluating past rewards and adjusting its predictions accordingly. Real-time metrics, including threat levels and rewards, are visualized with Plotly, while the video feed is displayed on Streamlit’s UI. The system runs continuously, analyzing each incoming frame, until it is manually stopped.

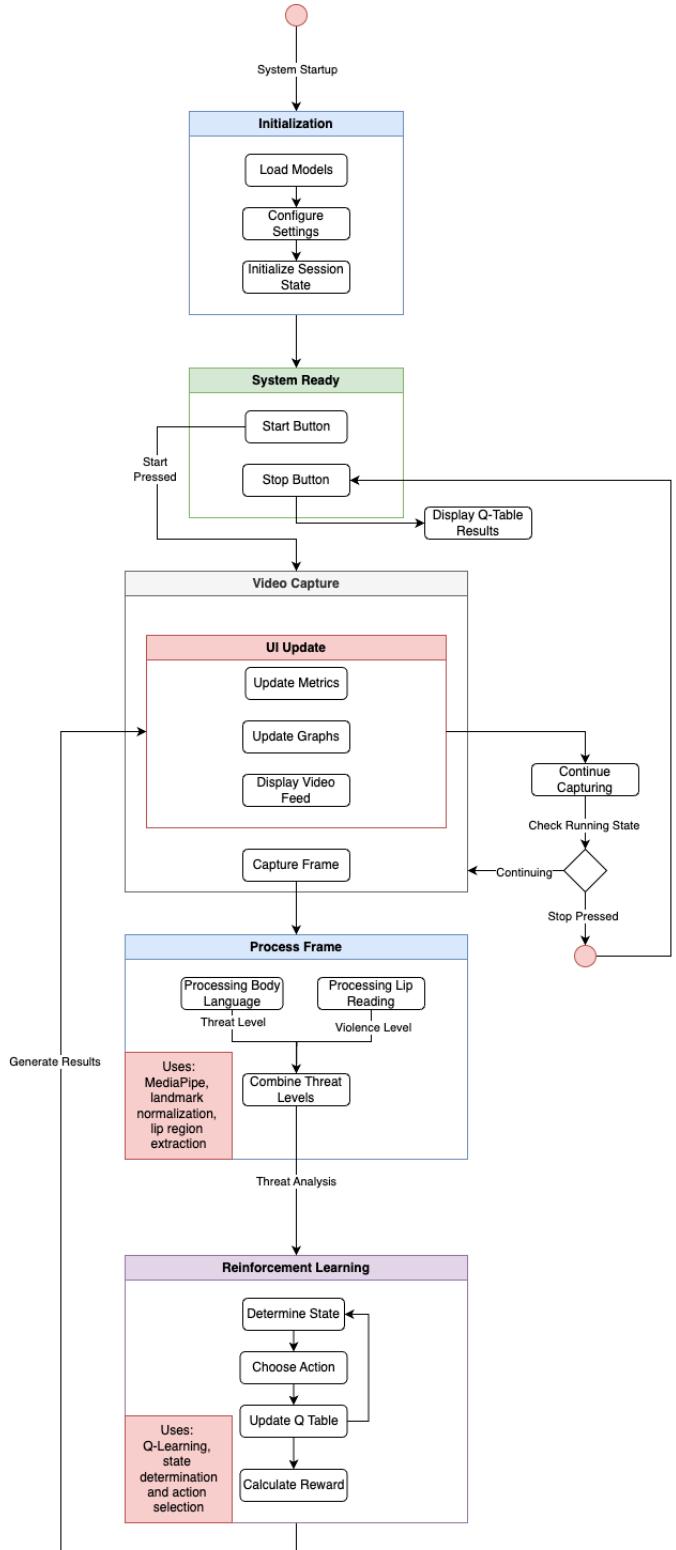


Fig. 1. Process flow diagram.

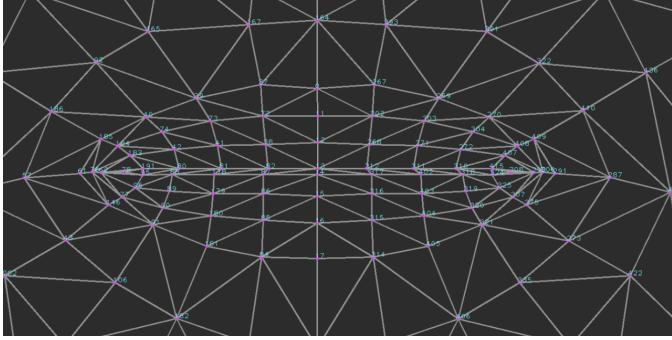


Fig. 2. The MediaPipe mouth/lip area mappings.

A. Body Language Component

The body language model relies on skeletal tracking data generated through Google's MediaPipe computer vision framework (Figure 2). The MediaPipe framework consists of abstracted pre-trained deep learning models that can readily identify skeletal landmarks on the human body in video frames.

Participants were recorded performing various poses. Using MediaPipe, the coordinates of a participant's face, pose, and hand landmarks across each frame in a recording session were written to a dataset with a pre-set label of "threatening" or "non-threatening." The frame coordinates and their labels were stored in a CSV file and used to train a Random Forest classifier, which learns to distinguish between threatening and non-threatening postures based on skeletal coordinates.

To ensure consistency, normalization techniques were applied, adjusting landmark positions relative to a reference body point to minimize position-based distortions. This helped reduce misclassifications caused by variations in user positioning in front of the camera. Inter-landmark normalization was also applied, using the distance between the shoulders as a relative scale.

Additionally, a rolling average mechanism was introduced to smooth fluctuations in predictions across frames and contextualize the instantaneous threat score. The trained Random Forest model was exported using Pickle and deployed for real-time classification. As the system processes video input, predictions are continuously updated and displayed on the UI.

B. Lip Reading Component

The lip-reading model is based on LipNet [7], [8], a deep learning framework that uses convolutional and recurrent neural networks to transcribe speech from visual input. Unlike models that classify individual words, LipNet processes entire sequences, improving accuracy by capturing context over time. The model was modified to work with live video input by storing the 75 most recent frames and using them as input. The oldest 15 frames are then removed, making room for 15 new frames.

The model consists of three spatio-temporal convolutional layers, each followed by a max-pooling layer, which extract

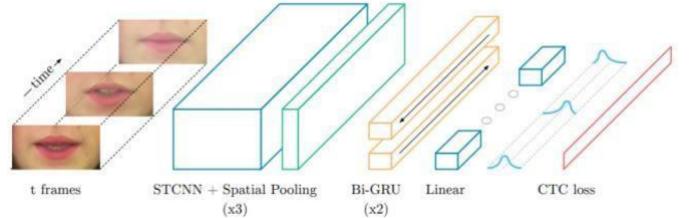


Fig. 3. LipNet architecture [9].

spatial and temporal features from lip movement sequences. These features are then processed by two recurrent neural networks, which analyze sequential dependencies. Finally, the Connectionist Temporal Classification (CTC) loss function helps align predicted sequences with the transcribed text, accounting for natural variations in speech (Figure 3).

To identify speech that is not friendly, a dictionary of violent keywords was created, assigning each word a predefined violence score. The lip-reading model transcribes speech, and each word is checked against this dictionary. If threatening words appear consistently, the highest detected violence value is passed to the RL model for further decision-making, every 15 frames.

C. Reinforcement Learning Component

To refine threat classification, a reinforcement learning (RL) framework was implemented, allowing the system to adapt to new behaviors over time. The RL model learns to associate physical movements and speech patterns with hostile levels, adjusting its predictions dynamically.

The reinforcement learning environment/approach in this code is simulated using a Q-learning approach to classify behaviors as threatening or non-threatening based on speech and body language. The system maintains a Q-table, which maps states (determined by the detected threat level) to actions ("all-good" or "de-escalate"). It updates the Q-values using the formula:

$$Q(s, a) = Q(s, a) + \alpha(r + \gamma \max_{a'} Q(s', a') - Q(s, a))$$

where α is the learning rate, γ is the discount factor, r is the received reward, and $\max_{a'} Q(s', a')$ represents the estimated future reward. The system balances exploration and exploitation using an epsilon-greedy strategy, which sometimes chooses random actions to improve learning.

During execution, the model receives input from lip-reading and body language analysis to classify behaviors. If a threatening word is detected in speech or if body language suggests aggression, the system assigns a threat level and chooses an action. After an action is taken, a reward is assigned based on correctness, and the Q-table is updated accordingly. The model saves and loads the best-performing Q-table to improve over time. By continuously updating based on real-time data, the system learns to detect threats more accurately and respond effectively.

The RL components: **State:** Real-time threat conditions, including body language patterns and speech analysis phases. **Action:** Choosing between "all-good" or "de-escalate" responses. **Reward:** A combination of correct threat assessment time and appropriate response outcomes.

III. RESULTS

This section presents the model's performance in detecting threatening and non-threatening behavior. The results are analyzed from different perspectives to evaluate the effectiveness of the approach. We also discuss key findings, improvements made during development, and the impact of specific methodological choices.

A. Parameter Optimization

Extensive testing was conducted to fine-tune the reinforcement learning model's parameters.

For the learning rate, a range of 0.15–0.20 provided a balanced approach with moderate value fluctuations, allowing the model to adjust gradually based on the data, preventing possible overshooting of optimal solutions while learning from the outcomes. This range reduces volatility and adjusts weights appropriately in accordance with the environment. Alternatively, a range of 0.25–0.35 resulted in more stable outcomes with improved interpretability. In this range, the model was capable of converging faster by making larger updates to the weights, resulting in a smoother learning process and reducing random fluctuation, albeit at a marginal cost to responsiveness. Based on this, the optimal learning rate was set at 0.28, providing a reasonable balance into a smoother operation with respect to the adaptability of the model. At 0, the model would not learn from new experiences, while at 1, it would completely overwrite previous learning with each new experience.

The discount rate was tested across multiple ranges. Values between 0.60–0.70 led to faster response times, whereas 0.75–0.85 improved stability and reliability. On the lower range, the model prioritizes more immediate reward as a faster response, with the trade-off of being less reliant on long term interpretation. On the higher range, the model alternatively prioritizes future reward, leading to cautious decision making at the cost of speed. Given the use case of the model, the best trade off was found at 0.71, where balance was shifted more towards quick interpretation.

For the exploration rate, values between 0.25–0.40 were analysed. Higher values in this range led to better responsiveness but introduced slight latency, as the model explores a broader set of actions at a cost of excessive exploration and increased randomness.

Lower values produced more consistent but less adaptive results, as the model relies more on the exploitation of known actions, which provides more consistent results while lowering the models adaptability. The optimal setting was determined to be 0.39, striking a balance between exploration and exploitation with an emphasis on the ability to adapt to human behaviour.

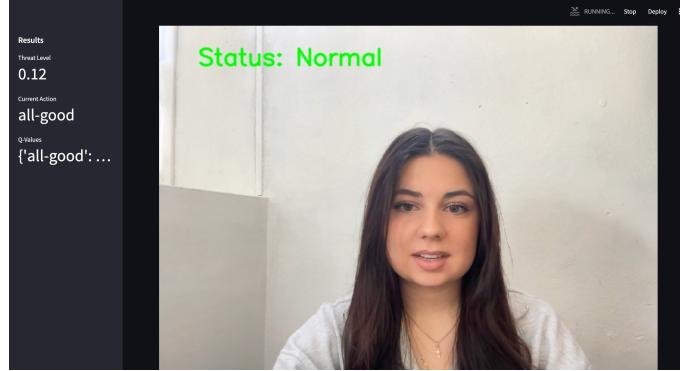


Fig. 4. An example of model identifying non-threatening behaviour.



Fig. 5. An example of model picking up on threatening behaviour.

B. Model Performance and Improvements

Early tests revealed inconsistencies in the body language model's ability to identify threats, mainly due to variations in the user's distance from the camera. This positional bias affected classification accuracy, leading to a high rate of false positives and false negatives.

To address this, a normalization process was implemented to account for differences in on-screen position. This adjustment reduced classification errors caused by distance by 90%, ensuring that maliciousness detection was based on actual behavior rather than a user's relative position.

Another major improvement was the introduction of a rolling average for threat scores. Initially, the model classified aggressive actions on a frame-by-frame basis, which caused unpredictable fluctuations. By averaging threat scores over multiple frames, sudden spikes and inconsistencies were significantly reduced, making the predictions more stable and reliable.

For the lip-reading model, increasing the number of frames analyzed per sequence from 50 to 75 resulted in noticeable improvements. It was a challenging task to find the ideal number of frames such that the model gets enough context but where it's also not taking too long to update the frames. This adjustment provided the model with greater temporal context, enhancing its ability to recognize speech patterns and im-

proving transcription accuracy. Currently, the lip transcription accuracy remains limited, as this is the first lip-reading AI designed for live video input. It is currently a challenge to detect whole words quickly, accurately and consistently. However, to address this, the current approach leverages phoneme-based analysis. An algorithm was developed to identify specific phonetic patterns, matching them against a predefined dictionary. The LipNet model was not originally trained on live video feed as well as violent, threatening, or profane language, which presents a challenge. Thus, this is breaking into cutting-edge territory where future work will focus on training a new model specifically on violent, threatening, or profane language, which alone should significantly improve transcription.

These refinements collectively led to a 90% reduction in false positives. A major factor behind this improvement was eliminating screen position biases where previously, the model struggled to classify a fist as threatening if it appeared in certain positions on the screen. With positional normalization and rolling average adjustments, gesture recognition became much more accurate.

Figures 4 and 5 demonstrate the models response to threatening and non-threatening behaviour, respectively. It can be clearly seen that a fist and angrily interpreted mouth position causes the model to flag the behaviour as threatening. Whereas a neutral positioned face with no extra hand cues leads the model to identify a non-threatening state.

At the moment, the RL model performance score, which is calculated from the recent reward history, is 72%. This is a good performance because a score over 0.5 means that the model is making more correct decisions than incorrect ones.

TABLE I
Q-VALUES FOR DIFFERENT THREAT LEVELS AND ACTIONS

Threat Level	All-Good	De-escalate
Low	3.45	1.45
Medium	0.18	2.72
High	0.00	2.09

The final Q-Table produced when the model is 72% effective can be seen in Table I. Some things to note for the "low" threat state, a score of 1.45 for "de-escalate" is good because that means the model favors the action "all-good" when the threat is low. Continuing to look at "de-escalate" for "medium" threat we notice it is a higher score of 2.72, which is also good because that indicates the model prefers "de-escalate" even when there is a moderate threat. Finally, a score of 2.09 for "de-escalate" of "high" threat is also good because it shows that the model avoids using the "all-good" state during situations of high threat.

We notice the model starts to degrade if the exploration rate gets too low, reward history gets taken over by one type of actions. Some signs of degradation include when performance score is below 50%, and Q-values start becoming very similar between actions.

While these improvements have significantly enhanced the model's performance, further refinements could still be explored. However, normalizing position data and smoothing

predictions have made the process much more reliable for real-time detection of suspicious actions.

IV. CONCLUSION

BOLLD marks a major step forward in multimodal detection of alarming cues by combining body language analysis, lip reading, and reinforcement learning. Unlike traditional systems, it provides a more adaptable approach to real-time threat assessment, especially in situations where audio is unreliable or unavailable.

Some key accomplishments include developing an advanced multimodal framework and implementing normalization techniques to minimize positional bias. The reinforcement learning model continuously adapts to behavioral patterns, making the system more responsive and effective.

Future improvements could include refining motion-based detection criteria, expanding the violence keyword dictionary, and incorporating adaptive thresholds that adjust based on environmental conditions. Further work is also needed to enhance reinforcement learning strategies and conduct large-scale real-world testing across diverse scenarios.

REFERENCES

- [1] Ramalingam, G. K., & Pattabiraman, S. (2024, August 26). AI-enhanced multimodal search engines for Cybersecurity Threat Detection. SSRN. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4903692.
- [2] Natural language processing in cyber threat intelligence: A major asset for infrastructure threat detection and response. Egis. (n.d.). <https://www.egis-group.com/all-insights/natural-language-processing-in-cyber-threat-intelligence-a-major-asset-for-infrastructure-threat-detection-and-response>.
- [3] Ismail, W. S. (2024, February 29). Threat Detection and Response Using AI and NLP in Cybersecurity. Abu Dhabi; Business Information Technology Department, Liwa College, Abu Dhabi.
- [4] Singhal, S. (2024, January). Real Time Detection, And Tracking Using Multiple AI Models And Techniques In Cybersecurity. NJ, USA; Infosys, US Engineering.
- [5] OARJST, E. (2024, December 20). Developing multimodal AI systems for comprehensive threat detection and geospatial risk mitigation. Open Access Research Journal of Science and Technology. <https://oarjst.com/content/developing-multimodal-ai-systems-comprehensive-threat-detection-and-geospatial-risk>.
- [6] Li, Y., Li, J., Cao, J., Xie, R., Wang, Y., Xu, M., Yizhi Li-INSC, T. U., Jiang LiZhongguancun Laboratory, B., Jiahao CaoINSC, T. U., Renjie XieINSC, T. U., Yangyang WangINSC, T. U., & Mingwei XuINSC, T. U. (2024, December 9). Poster: Few-shot inter-domain routing threat detection with large-scale multi-modal pre-training: Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security. ACM Conferences. <https://dl.acm.org/doi/10.1145/3658644.3691402>.
- [7] M. Assael, Y., Shillingford, B., Whiteson, S., & de Freitas, N. (2016, December 16). LIPNET: END-TO-END SENTENCE-LEVEL LIPREADING. Department of Computer Science, University of Oxford, Oxford, UK Google DeepMind, London, UK CIFAR, Canada .
- [8] codenigma1. (n.d.). CODENIGMA1/lipappnet: This is LipNet network where model learn from lip movement and predict text without voice. GitHub. <https://github.com/codenigma1/LipAppNet>
- [9] Wang, J., Wang, Y., Liu, A., & Xiao, J. (1970, January 1). Assistance of speech recognition in noisy environment with sentence level lip-reading. SpringerLink. https://link.springer.com/chapter/10.1007/978-3-319-69923-3_64.

Brain-Agnostic 3DCNNs Learn Naturalistic Emotion from 7t fMRI

Joshua Lunger¹, Mason Hu¹, Aditya Rajeev¹ Tohya Tanemura¹, Samuel Kostousov¹, Jurgen Germann^{1, 2}

¹*University of Toronto*, ²*Krembil Brain Institute*

Abstract—Understanding emotions through neural activity is a key challenge in affective computing and neuroscience. In this work, we leverage brain-agnostic 3D convolutional neural networks (3DCNN) to learn functional representations of emotions from large-scale naturalistic 7T fMRI data. Our learned representations are consistent with neurobiological principles, highlighting the potential of deep learning for neural emotion inference. Code and data are available at [<https://github.com/lungerjo/DeepEmotion>].

I. INTRODUCTION

Understanding emotions through large-scale naturalistic data offers a pathway to more effective and scalable emotion recognition systems. These systems could then support care workers in identifying and assisting individuals who exhibit atypical or impaired emotional processing signals. Furthermore, advancements in scalable inference on neural data have the potential to bridge human cognition and machine learning by enabling more personalized and adaptive human-computer interactions. This work paves the way for integrating neural preferences into machine learning systems, expanding the applications of brain-aware AI in both healthcare and broader technological domains.

II. METHODOLOGY

A. Data Collection and fMRI Preprocessing

We utilized the publicly available high-resolution 7T fMRI dataset [1] from the StudyForrest project consisting of whole-brain fMRI recordings collected while participants listened to an audio-described version of the movie Forrest Gump. The dataset scans are acquired at a spatial resolution of 1.4 mm isotropic and a temporal resolution of 2 seconds.

For our study, we leveraged non-linear anatomically aligned fMRI data mapped to a common group template using iterative affine and non-linear transformations included with the dataset. This approach minimizes inter-subject anatomical variability. The alignment procedure followed an iterative group-based registration process, where each participant's motion-corrected and distortion-corrected EPI images were first aligned using an affine transformation and then refined using a high-resolution non-linear warp field.

B. Annotation Preprocessing

Emotion annotations were collected from eight external observers who rated perceived emotions in the film. To ensure balanced training and a strong emotional signal, we applied heuristic clustering to group annotations by mapping to the

5 most frequently observed emotion categories. Each fMRI sample was assigned an emotion label based on the majority vote among observers, with the condition that at least half of them agreed on the emotion.



Fig. 1: Covariance scores for emotion annotations across observers. Heuristic clustering was used to map annotations to the five most common emotions.

C. Training

We trained 20 brain-agnostic 3D Convolutional Neural Networks (3DCNN) [2] to classify emotion states from fMRI data. 19 models were trained on one subject each and one model was trained on all 19 subjects. The models were trained using stochastic minibatch gradient descent with categorical cross-entropy loss and optimized with the Adam optimizer at a learning rate of 0.001. Training was conducted for 50 epochs on a NVIDIA Quadro4000.

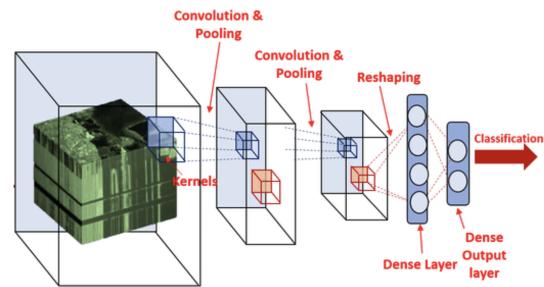


Fig. 2: The network consists of 3 convolutional layers with 3D kernels, batch normalization, and ReLU activations. A series of max-pooling operations were applied to downsample spatial dimensions while preserving feature representations. The final convolutional features were flattened and passed through 2 fully connected layers before a softmax classification head. [3]

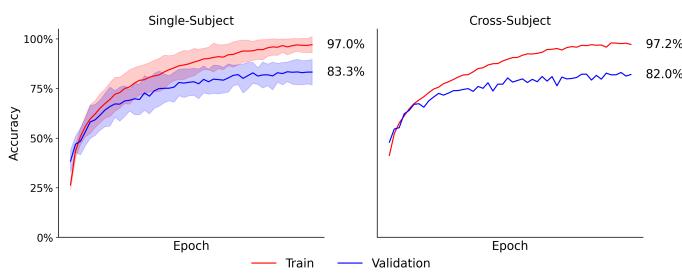


Fig. 3: Train and validation accuracy over epochs. Single-subject accuracy is averaged over 19 models trained on 1 subject each. Cross-subject accuracy is 1 model trained on all 19 subjects.

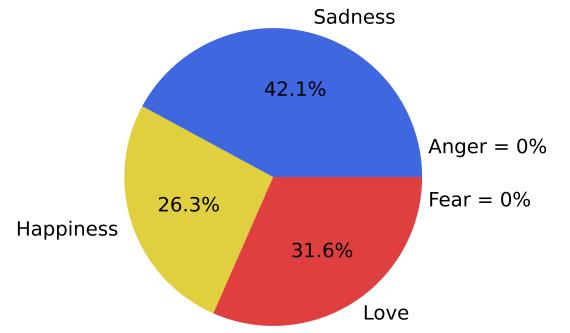


Fig. 4: Model predictions on held-out data from subject 18 during love scenes.

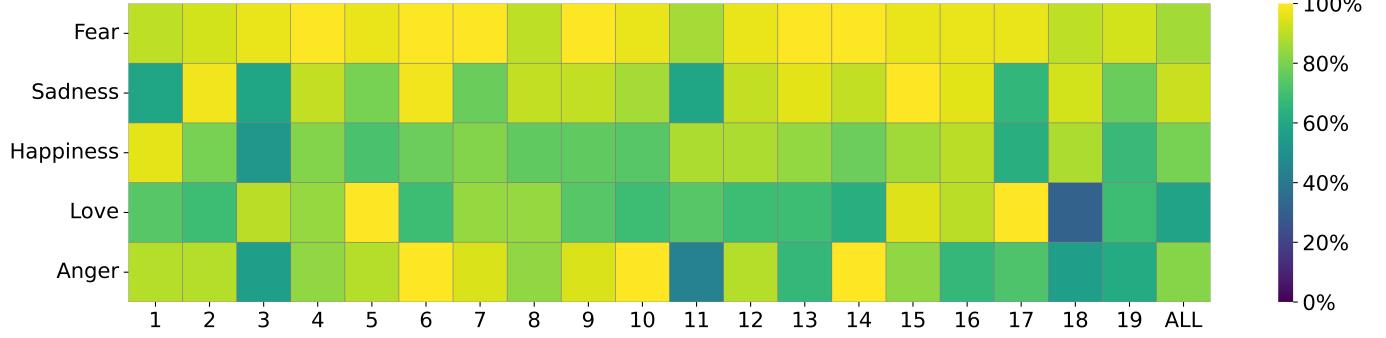


Fig. 5: Emotion-wise accuracy on held-out data within subject. The column label indicates the subject trained on by this model.

III. RESULTS

We observe impressive performance on held-out data from both single-subject and cross subject models with limited compute and hyperparameter tuning demonstrating the power of this technique when applied to large-scale naturalistic fMRI data. In particular, our single-subject models perform similarly on held out data to the cross-subject model, obtaining an average classification accuracy around %80.

IV. DISCUSSION

Our study is, to our knowledge, one of the first to successfully apply a generic deep 3D-CNN to naturalistic fMRI for emotion decoding at 7T resolution. Prior fMRI-based emotion decoders have either focused on region-of-interest features [4], non-naturalistic emotional tasks [5], or statistical models with inductive biases [6]. This data-driven strategy allows the model to discover relevant spatiotemporal patterns of emotion across the brain in a naturalistic setting unbounded by anatomical assumptions. By demonstrating that a 3D-CNN can be trained on whole-brain 7T fMRI responses to a complex movie and decode emotional states above chance, we establish a new benchmark for large-scale neural decoding in the emotion domain and highlights the promise of modern deep learning in mapping between brain activity and rich emotional experiences.

There are several key consistencies with neurobiological findings directly observed from the emotion-wise inference results despite our brain-agnostic model. First, our model's performance varies significantly across emotions by individual. Indeed naturalistic emotional responses exhibit high inter-individual variability [7], [8]. Beyond differences in subjective emotional experience, there are also physiological and neural sources of variability. Each person's brain anatomy and functional organization is unique – the exact location and magnitude of emotion-related activations can shift from one brain to another, even if qualitatively the same networks (e.g. limbic system, TPJ, prefrontal cortex) are engaged. One subject might recruit a slightly different constellation of regions or have a different lateralization for a given emotion than another. This functional idiosyncrasy is well recognized as a hurdle in multi-subject fMRI analysis [9]. Consequently, a brain-agnostic CNN might misinterpret inter-individual differences as mere data variance, when in fact each subject has a distinct, reliable pattern for themselves that just doesn't match the group pattern well. Our model's difficulty in generalizing could thus be partly due to person-specific neural signatures of emotion.

Notably, our model consistently performs well detecting fear. Neuroimaging evidence suggests that fear triggers a particularly robust and stereotyped brain response across individuals, making it stand out from other emotions. In fMRI

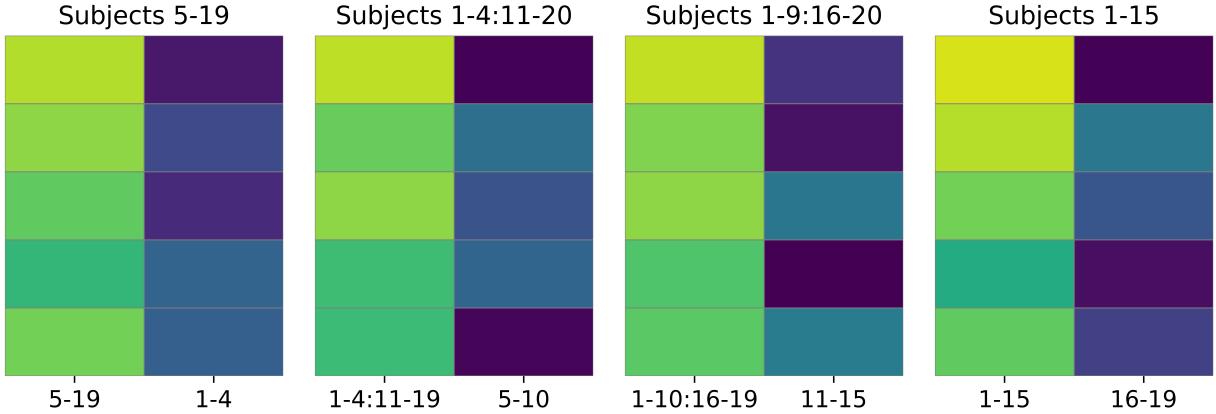


Fig. 6: Emotion-wise model accuracy on held-out subjects. The sub-figure headers are the subjects the model was trained on.

studies, negatively valenced, high-arousal stimuli (like fear-inducing scenes) drive highly synchronized activity in key emotion-processing regions (e.g. amygdala, insula, midcingulate), showing much greater inter-subject consistency than neutral or positive emotional content [10]. For example, a suspenseful horror film clip elicited nearly identical brain activation patterns across different viewers, indicating that fear evokes a shared neural signature that is easier for a general 3D-CNN model to detect compared to more variable emotional states [11].

Moreover, the clinical applications of inference are promising. Notably, the single-subject model trained exclusively on subject 19 performs poorly in classifying love scenes compared to other single-subject models. Indeed, our model misclassifies these scenes as exhibiting sadness in %42.1 of fMRI labels, love in %31.6 of labels and happiness in %26.3 of labels. These misclassifications for subject 19, suggesting potentially atypical or “misaligned” neural responses during love scenes, are reminiscent of findings in clinical populations where aberrant emotional processing signals appear in fMRI data [12]. By analyzing these misclassifications, we gain insight into this subject’s unique emotional responses, potentially identifying neural differences or deficits.

Finally, models that generalize to unseen subjects could enable the development of consumer applications leveraging neural data for human preferences. However, when training on a subset of subjects and inferring on held out subjects, our model accuracy collapses. These findings highlight the need for further techniques to enhance cross-subject generalization.

V. CONCLUSION

Our findings establish the promise of brain-agnostic 3D-CNNs in decoding emotional states from high-resolution, naturalistic 7T fMRI data. Our learned consistency with neurobiological theory confirms the quality of our learned representations. On the other hand, we also highlight the challenges in achieving robust cross-subject generalization and held-out subject inference. Future work can focus on more sophisticated alignment techniques, data augmentation, and

larger, more diverse datasets to further improve the generality and reliability of deep learning-based emotion decoding.

REFERENCES

- [1] M. Hanke, F. Baumgartner, P. Ibe *et al.*, “A high-resolution 7-tesla fmri dataset from complex natural stimulation with an audio movie,” *Sci Data*, vol. 1, p. 140003, 2014.
- [2] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, “Learning spatiotemporal features with 3d convolutional networks,” 2015. [Online]. Available: <https://arxiv.org/abs/1412.0767>
- [3] S. Hesaraki, “3dcnn,” November 2023, accessed: 2025-03-16. [Online]. Available: <https://medium.com/@saba99/3d-cnn-4ccfab119cc2>
- [4] G. Lettieri, G. Handjaras, E. Ricciardi, A. Leo, P. Papale, M. Betta, P. Pietrini, and L. Cecchetti, “Emotionotopy in the human right temporo-parietal cortex,” *Nature Communications*, vol. 10, no. 1, p. 5568, 2019.
- [5] M. Tchibozo, D. Kim, Z. Wang, and X. He, “Emotional brain state classification on fmri data using deep residual and convolutional networks,” 2022. [Online]. Available: <https://arxiv.org/abs/2210.17015>
- [6] H. Saarimäki, L. F. Ejtehadian, E. Glerean, I. P. Jääskeläinen, P. Vuilleumier, M. Sams, and L. Nummenmaa, “Distributed affective space represents multiple emotion categories across the human brain,” *Social Cognitive and Affective Neuroscience*, vol. 13, no. 5, pp. 471–482, 2018.
- [7] P. Tovote, J. Fadok, and A. Lüthi, “Neuronal circuits for fear and anxiety,” *Nature Reviews Neuroscience*, vol. 16, pp. 317–331, 2015.
- [8] T. Steimer, “The biology of fear- and anxiety-related behaviors,” *Dialogues in Clinical Neuroscience*, vol. 4, no. 3, pp. 231–249, 2002.
- [9] P.-H. Chen, J. Chen, Y. Yeshurun, U. Hasson, J. V. Haxby, and P. J. Ramadge, “A reduced-dimension fmri shared response model,” in *Neural Information Processing Systems*, 2015. [Online]. Available: <https://api.semanticscholar.org/CorpusID:10143260>
- [10] L. Nummenmaa, E. Glerean, M. Viinikainen, I. P. Jääskeläinen, R. Hari, and M. Sams, “Emotions promote social interaction by synchronizing brain activity across individuals,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 109, no. 24, pp. 9599–9604, 2012. [Online]. Available: <https://doi.org/10.1073/pnas.1206095109>
- [11] Y. Wang and Y. Wang, “A neurocinematic study of the suspense effects in hitchcock’s psycho,” *Frontiers in Communication*, vol. 5, 2020. [Online]. Available: <https://www.frontiersin.org/articles/10.3389/fcomm.2020.576840>
- [12] M. L. Phillips, W. C. Drevets, S. L. Rauch, and R. Lane, “Neurobiology of emotion perception ii: Implications for major psychiatric disorders,” *Biological Psychiatry*, vol. 54, no. 5, pp. 515–528, 2003.

Can AI Design Cancer Vaccines? Evaluating Neural Networks for Epitope Prediction

Caitlin Roach

Queen's University

caitlin.roach@queensu.ca

Jennifer Qiu

Queen's University

20jqq@queensu.ca

Jaeson Wang

Queen's University

21jw145@queensu.ca

Salma Elsayed

Queen's University

salma.elsayed@queensu.ca

Adeel Haq

Queen's University

21ah33@queensu.ca

Abstract—Immunotherapy as a form of cancer treatment can be effective, but often causes the immune system to attack healthy tissues, leading to significant side effects. Therapeutic cancer vaccines offer a safer, tumour-specific alternative, but their efficiency relies on accurate epitope prediction, which is used to identify regions of a protein that can trigger an immune response in a patient. This study evaluates MHCflurry on a clinically relevant melanoma-associated antigen to assess the real-world applications of computational epitope prediction to therapeutic melanoma vaccines. We assessed predicted epitopes based on binding affinity, presentation, and processing scores, identifying the peptide sequence AQAPATEEQEA as the strongest candidate. We visualized results and key findings for a quantitative analysis of the peptide sequences. Our findings suggest that while computational tools like MHCflurry show promise in the design of cancer vaccines, they require experimental validation before implementation or clinical application.

I. INTRODUCTION

Melanoma is a type of skin cancer in which malignant (cancer) cells rapidly multiply in the cells that colour the skin (melanocytes). This is the most dangerous type of skin cancer due to its aggressive growth and ability to spread to any organ in the body (National Cancer Institute, 2025). Melanoma is diagnosed by biopsy and analysis of skin lesions. Treatment plans include radiation therapy, chemotherapy, and, notably, immunotherapy.

Immunotherapy trains the body's immune system to recognize, target, and attack malignant cells, stopping or slowing cancer growth while preventing it from spreading to other organs. Types of immunotherapy include immune checkpoint inhibitors (ICIs), CAR T cell therapy, antibody-drug conjugates, and therapeutic vaccines. This treatment can be effective, but often causes the immune system to attack healthy cells and tissues, leading to what are called immune-related adverse events (irAEs). irAEs can affect many organ systems, including the skin, liver, and gastrointestinal system (Vaddepally et al., 2022). These effects are often serious. In a 2023 study, combination therapy with ICIs resulted in a 25% to 30% incidence of grade 4 (life-threatening) hepatitis and a grade 3 (severe) toxicity rate of approximately 15% (Yin et al., 2023).

Given this need for safer immunotherapy options, therapeutic vaccines present an opportunity for safer alternatives to

traditional immunotherapy. Therapeutic cancer vaccines aim to provide tumor-specific treatment with fewer side effects, resulting in far less severe irAE rates than other forms of treatment. A 2025 study on a personalized therapeutic vaccine for advanced kidney cancer resulted in only mild flu-like symptoms, with no serious side effects reported (Braun et al., 2025), highlighting the potential that vaccines have to reduce irAE rates.

These vaccines are designed using epitope prediction, a computational method used to identify regions of a protein that can trigger an immune response in a patient. Machine learning shows great potential in the prediction of clinically relevant epitopes, improving vaccine efficacy while minimizing adverse effects on patients such as irAEs.

A. Motivation

The development of safe, widely accessible cancer treatment with minimal side effects is crucial to optimize quality of life for patients and their families, improve overall survival rates, and address toxicities like irAEs.

An optimal treatment would mitigate the broader societal impacts of cancer, including economic burden due to reduced labour force participation and productivity (OECD, 2024), strain on healthcare systems (Prager et al., 2018), and psychological impact of traumatic treatment plans (van Roij et al., 2019). Therapeutic vaccines offer a promising candidate for this treatment, and accurate epitope prediction is crucial for their development.

B. Problem Definition

Although immunotherapy can be effective in cancer treatment, high irAE rates pose serious risks to patient safety, limiting widespread use. Severe irAEs often lead to significant health complications, longer hospitalizations, and treatment discontinuation, resulting in lower overall survival rates (Liang et al., 2024).

Therapeutic cancer vaccines offer a promising alternative to traditional immunotherapy by inducing a targeted immune response in the patient, lowering the risk of off-target effects.

However, their development and implementation is limited by challenges in epitope prediction.

Recent advances in machine learning have introduced new approaches to epitope prediction. This study aims to evaluate machine learning epitope prediction tools like MHCflurry, determining their feasibility in identifying strong binding epitopes to inform melanoma vaccine design for improved efficiency and patient safety.

II. RELATED WORK

In response to the growing demand for safer and more effective cancer treatments, many studies have explored computational approaches for epitope prediction, leveraging machine learning models for efficient vaccine design.

For instance, Tarek et al. (2018) applied computational epitope prediction tools to evaluate peptide sequences for non-small cell lung cancer vaccine design, identifying several promising candidates, but found that binding affinity scores did not always correlate with real-world immunogenicity. Similarly, Roudko et al. (2020) utilized computational tools to predict peptide interactions, but noted biases in training data.

Our study builds on this research by integrating MHCflurry, a machine learning-based predictor, with a real-time web application to improve accessibility for researchers. Unlike previous studies, we emphasize interpretability, open-source data, and scalability for broader vaccine design applications. In this study, we evaluate MHCflurry on a clinically verified melanoma-associated antigen (MAGE-A3) and analyze its performance in epitope prediction for immunotherapy.

III. METHODOLOGY

Due to computational limitations, we opted to conduct this study using MHCflurry, a well-documented, pre-trained machine learning-based epitope prediction tool. While other predictors exist (e.g., NetMHCpan), MHCflurry was selected for its open-source accessibility and previous validation in literature.

MHCflurry predicts peptide binding affinities using a system of artificial neural networks (ANNs). These ANNs use both classification and regression to identify potential candidates for vaccine design. Accuracy is determined through a mean square error loss function.

We evaluated this system on a clinically verified melanoma-associated antigen (MAGE-A3), sourced from the Immune Epitope Database (IEDB) to study its accuracy at predicting epitope candidates for melanoma vaccine design. We ran MHCflurry on 9-mer peptide sequences from MAGE-A3, predicting binding affinities for the HLA-A*02:01 allele (due to high clinical relevance). Candidates were evaluated and ranked based on their predicted binding affinity scores, with lower values indicating stronger binding potential.

We used Python 3 to create a web application, which was deployed using Gradio for a smooth user experience. We leveraged matplotlib and seaborn to visualize results from MHCflurry quickly and efficiently from within the user's browser.

A. Evaluation Methods

We evaluated the model on several key metrics, including half-maximal inhibitory concentration (IC50) scores as a measurement of the binding affinity of predicted epitope candidates. A very low IC50 score suggests an exceptional candidate for vaccine development, as strong binding increases the chances of recognition by the immune system.

Interpreting the results graphically, we were able to evaluate MHCflurry's predictions on the basis of interactions within the body's immune system, exploring real-world applications to vaccine design.

IV. RESULTS AND DISCUSSION

While developing this project, we aimed to assess the feasibility of computational tools like MHCflurry with the goal of improving the efficiency, accuracy, and safety of cancer immunotherapy.

This computational approach identified strong epitopes for melanoma vaccine design that are not yet documented in existing immunogenicity databases. This suggests potential novel epitope candidates for melanoma vaccine design while highlighting the need for further statistical analysis and future experimental validation.

A. Results

For readability, we interpreted the results of this study both graphically and in terms of raw data.

TABLE I
MHCFLURRY MAGE-A3 RESULTS

peptide	affinity (IC50 in nM)	presentation	processing
AQAPATEEQEA	2162.32	0.058	0.100
ALGLVGAQAPA	2912.19	0.032	0.008
AASSSTLVEV	4185.49	0.023	0.007
GLEARGEALGL	4205.67	0.027	0.052
TLGEVPAAESP	11775.71	0.010	0.051
SNQEEEGPSTF	13409.89	0.007	0.004
TLVEVTLGEVP	13981.99	0.014	0.206
STLVEVTLGEV	14313.88	0.008	0.040
GLVGAQAPATE	16062.32	0.006	0.004
SPDPPQSPQGA	16979.16	0.006	0.006

This table shows the raw output of MHCflurry on MAGE-A3, which predicted peptide sequences (epitopes) as potential candidates for vaccine design. In this study, peptides are evaluated on key metrics such as IC50 score, presentation on the surface of a cell, and processing within the cell. During this prediction run, MHCflurry identified the AQAPATEEQEA sequence the strongest candidate among the predicted epitopes due to its low IC50 score, coupled with high presentation and processing metrics. This suggests that this specific sequence has a strong likelihood of being recognized by immune cells, and would therefore be an ideal candidate for vaccine design.

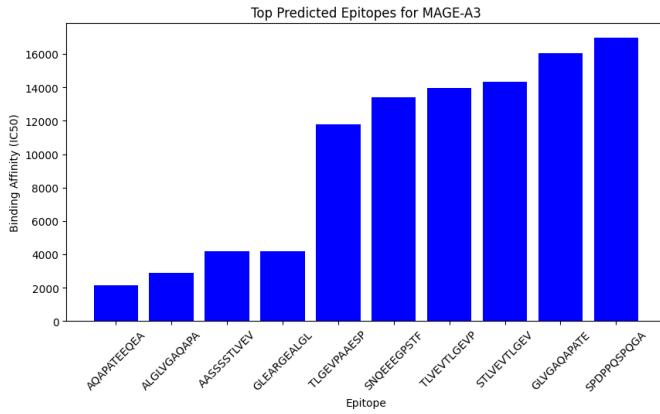


Fig. 1. Visualization of epitope candidates for MAGE-A3

MHCflurry identified multiple epitope candidates with varying binding affinities. The top candidate, AQAPATEEQEA, is shown here with the strongest predicted binding, suggesting a promising target for melanoma vaccine design.

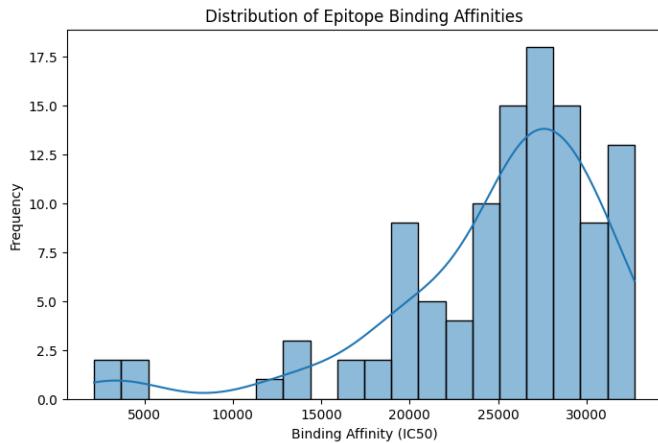


Fig. 2. Distribution of binding affinity (IC50) scores in epitope predictions from MHCflurry

The majority of predicted epitopes have high IC50 values, indicating weak binding affinity. However, the peptide AQAPATEEQEA falls in the low IC50 range, reinforcing its potential.

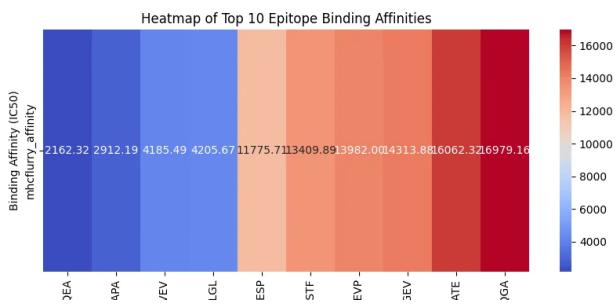


Fig. 3. Heatmap visualization of top predicted MAGE-A3 epitopes

This heatmap ranks the top 10 predicted epitopes for MAGE-A3 based on their IC50 values. Lower IC50 values (blue) indicate stronger binding, while higher IC50 values (red) indicate weak binding. AQAPATEEQEA shows the strongest binding affinity, making it a promising target.

B. Ethical Considerations

Although therapeutic cancer vaccines offer a promising safe alternative to traditional cancer treatments, their development, evaluation, and integration with existing medical frameworks present significant ethical challenges, particularly regarding access to treatment, accountability, and data privacy.

Therapeutic cancer vaccines raise ethical concerns about who benefits from this type of specialized cancer treatment, calling into question how factors like socioeconomic status, geographical location, insurance coverage, and income inequality will influence cancer treatment. When developing new treatments for diseases as prevalent as cancer, it is also important to consider the social determinants of health - the non-medical living conditions that influence well-being - and how they may affect treatment. Factors like income, education, employment, and access to healthcare services all affect the impact of therapeutic cancer vaccines, and must be considered.

Due to the complex nature of biological training data, computational tools like MHCflurry raise ethical concerns around algorithmic bias, transparency, and data privacy. Such models must be evaluated to identify and mitigate biased outputs, ensuring that these tools are well-equipped to serve diverse populations.

Furthermore, as machine learning continues to advance, it remains crucial that all computational results are rigorously tested and examined in both lab work and clinical settings to ensure integrity, accuracy, and efficiency.

V. CONCLUSION

This study explored computational epitope prediction with machine learning tool MHCflurry, revealing novel epitope candidates for melanoma vaccine design. Further analysis and clinical review are needed to verify clinical significance and real-world feasibility of these results. We demonstrated the potential of machine learning-based epitope prediction in the process of designing therapeutic melanoma vaccines, highlighting the growing role of machine learning in bioinformatics and computational biology.

As machine learning continues to advance, its integration with laboratory research could significantly enhance the precision and accessibility of therapeutic cancer vaccines, improving quality of life and overall survival rates for patients.

VI. FUTURE WORK

In the future, we plan to develop this tool further, potentially integrating computational epitope prediction with deep learning models to create an entirely open-source vaccine design tool for cancer researchers. By developing an open-source web application, we plan to make epitope prediction for cancer therapeutics accurate and widely available, enabling

researchers to quickly test vaccine candidates with minimal computational cost. We will need to conduct further statistical analysis and comparison to real-world benchmarks to confirm the real-world applications and immunogenicity of epitopes predicted using this tool.

Furthermore, integrating Major Histocompatibility Complex (MHC) class I and II prediction could potentially improve the effectiveness of this vaccine development tool. Class I epitopes recognize and destroy harmful cells, while class II epitopes enhance immunological memory, preventing the recurrence of cancer (Wang et al., 2021).

VII. LIMITATIONS

During the development of this project, we experienced several challenges that limited the scope of our research.

Due to system compatibility issues, we were unable to run epitope-predict natively on local hardware. We adapted the implementation to run on Google Colab, which introduced limitations in computational resources and runtime constraints.

Additionally, the results of this study are based solely on computational predictions. While MHCflurry provides key insights into potential epitope candidates, experimental validation is necessary to confirm the real-world applications of the identified epitope candidates.

REFERENCES

- [Braun et al., 2025] Braun, D., Moranzoni, G., Chea, V., et al. (2025). A neoantigen vaccine generates antitumour immunity in renal cell carcinoma. *Nature*.
- [Cook et al., 2024] Cook, S., Samuel, V., Meyers, D. E., and et al. (2024). Immune-related adverse events and survival among patients with metastatic nsclc treated with immune checkpoint inhibitors. *JAMA Network Open*.
- [Farrell, 2021] Farrell, D. (2021). epitopepredict: a tool for integrated mhc binding prediction. *GigaByte*.
- [Institute, 2025] Institute, N. C. (2025). Melanoma treatment (pdq®) – health professional version.
- [Mohammad M. Tarek, 2018] Mohammad M. Tarek, Ayman E. Shafei, M. A. M. M. M. (2018). Computational prediction of vaccine potential epitopes and 3-dimensional structure of xage-1b for non-small cell lung cancer immunotherapy. *Biomedical Journal*.
- [O'Donnell et al., 2020] O'Donnell, T., Rubinsteyn, A., and Laserson, U. (2020). Mhcflurry 2.0: Improved pan-allele prediction of mhc i-presented peptides by incorporating antigen processing. *Cell Systems*.
- [OECD, 2024] OECD (2024). *Tackling the Impact of Cancer on Health, the Economy and Society*. OECD Publishing, Paris.
- [O'Donnell et al., 2018] O'Donnell, T., Rubinsteyn, A., Bonsack, M., Riemer, A. B., Laserson, U., and Hammerbacher, J. (2018). Mhcflurry: Open-source class i mhc binding affinity prediction. *Cell Systems*.
- [Prager et al., 2018] Prager, G. W., Braga, S., Bystricky, B., Qvortrup, C., Criscitiello, C., Esin, E., Sonke, G. S., Martínez, G. A., Frenel, J. S., Karamouzis, M., Strijbos, M., Yazici, O., Bossi, P., Banerjee, S., Troiani, T., Eniu, A., Ciardiello, F., Tabernero, J., Zielinski, C. C., Casali, P. G., and Ilbawi, A. (2018). Global cancer control: responding to the growing burden, rising costs and inequalities in access. *ESMO Open*, 3(2):e000285.
- [Vaddepally et al., 2022] Vaddepally, R., Doddamani, R., Sodavarapu, S., Madam, N. R., Katkar, R., Kutadi, A. P., Mathew, N., Garje, R., and Chandra, A. B. (2022). Review of immune-related adverse events (iraes) in non-small-cell lung cancer (nsclc)-their incidence, management, multi-organ iraes, and rechallenge. *Biomedicines*, 10(4):790.
- [van Roij et al., 2019] van Roij, J., Brom, L., Youssef-El Soud, M., van de Poll-Franse, L., and Raijmakers, N. J. H. (2019). Social consequences of advanced cancer in patients and their informal caregivers: a qualitative study. *Supportive Care in Cancer*, 27(4):1187–1195.
- [Wang et al., 2021] Wang, X., Yu, Z., Liu, W., Tang, H., Yi, D., and Wei, M. (2021). Recent progress on MHC-I epitope prediction in tumor immunotherapy. *American Journal of Cancer Research*, 11(6):2401.
- [Yin et al., 2023] Yin, Q., Wu, L., Han, L., Zheng, X., Tong, R., Li, L., Bai, L., and Bian, Y. (2023). Immune-related adverse events of immune checkpoint inhibitors: a review. *Frontiers in Immunology*, 14:1167975.
- [O'Donnell et al., 2020] [O'Donnell et al., 2018] [Yin et al., 2023] [Cook et al., 2024] [Vaddepally et al., 2022] [OECD, 2024] [van Roij et al., 2019] [Prager et al., 2018] [Braun et al., 2025] [Mohammad M. Tarek, 2018] [Farrell, 2021] [Institute, 2025] [Wang et al., 2021]

CNN-based Diagnosis from Medical Imaging: Leveraging Transfer Learning for Enhanced Accuracy

Nathan Wan

University of Western Ontario

nwan23@uwo.ca

Juna Kim

University of Western Ontario

jkim2983@uwo.ca

Liam McQuay

University of Western Ontario

lmcquay2@uwo.ca

Kevin Du

University of Western Ontario

kdu22@uwo.ca

Millicent Song

University of Western Ontario

msong257@uwo.ca

Matthew Louis Li

University of Western Ontario

mli2646@uwo.ca

Besma Serrai

University of Western Ontario

bserrai@uwo.ca

Artemiy Vishnyakov

University of Western Ontario

avishny@uwo.ca

Sabrina Lee

University of Western Ontario

slee2893@uwo.ca

Abstract—Medical imaging plays a crucial role in disease diagnosis and treatment planning, yet the increasing volume of imaging data poses challenges for radiologists and healthcare systems. This study investigates the application of Convolutional Neural Networks (CNNs) for automated medical image classification, leveraging transfer learning to enhance diagnostic accuracy. Using a pre-trained DenseNet-121 model, we developed and evaluated CNN-based classifiers for lung cancer, pneumonia, and tuberculosis detection.

Our models achieved high classification accuracy, with 90.48% for lung cancer, 91.83% for pneumonia, and 99.84% for tuberculosis, demonstrating their effectiveness in distinguishing between normal and pathological cases. The results highlight the potential of AI-driven diagnostics to assist medical professionals by reducing workload, improving diagnostic speed and accuracy, and addressing the shortage of radiologists.

Despite promising performance, challenges such as dataset variability, potential biases, and real-world deployment remain. Future work will focus on expanding datasets, improving model interpretability, and integrating AI-assisted tools into clinical workflows to enhance reliability and accessibility in medical imaging.

I. INTRODUCTION

Medical imaging is a cornerstone of modern healthcare, enabling the early detection and diagnosis of a wide range of diseases. However, the increasing volume of medical imaging data presents challenges for radiologists and healthcare providers, who must analyze complex scans under time constraints. Artificial intelligence (AI), particularly deep learning, with its advantage of end-to-end processing, has emerged on a large scale in recent medical diagnosis studies [1], offering the potential for faster, more accurate, and more efficient analysis. This research explores the development of an AI-driven medical imaging diagnostic system using convolutional neural networks (CNNs) to detect anomalies in X-rays, magnetic

resonance imaging (MRI), and computed tomography (CT) scans.

Deep learning, a subset of machine learning, has demonstrated remarkable capabilities in image recognition and classification tasks, making it an ideal tool for medical imaging diagnostics. Convolutional neural networks (CNNs) are particularly well-suited for this purpose because they automatically extract hierarchical features from medical images, distinguishing between normal and pathological conditions with high accuracy. One such model in skin cancer achieved levels of competence comparable to dermatologists [2]. Unlike traditional image processing techniques that rely on hand-crafted feature extraction, CNNs learn spatial hierarchies of features directly from raw imaging data [3], leading to superior performance concerning old approaches [4].

Convolutional neural networks (CNNs) have empowered researchers to develop advanced models that accurately differentiate between healthy patients and those showing signs of cancer [5]. These sophisticated algorithms have improved speed and accuracy, marking a substantial technological breakthrough in medical diagnostics. This advancement has the potential to significantly transform how cancer is diagnosed and can greatly improve patient outcomes by facilitating earlier and more accurate detection, thereby enabling prompt and effective treatment. Incorporating CNNs into cancer diagnostics signifies a crucial milestone in integrating artificial intelligence within healthcare, promising to boost diagnostic accuracy and streamline medical interventions. Consequently, this technology could substantially reduce the mortality rates and overall global burden associated with cancer.

A. Motivation

The growing demand for medical imaging diagnostics has placed significant pressure on healthcare systems, leading to increased workloads for radiologists and the risk of diagnostic errors. AI, particularly deep learning models like CNNs, has demonstrated immense potential in enhancing diagnostic accuracy and efficiency [2]. AI can rapidly analyze complex imaging data, detect abnormalities with high precision, and support medical professionals in making timely and informed decisions [6]. This project is motivated by the need to integrate AI into medical imaging workflows to improve patient outcomes. Studies have shown that AI-assisted diagnostics can significantly reduce interpretation time while maintaining or exceeding human-level accuracy [7]. Moreover, AI can address the shortage of radiologists in many regions, ensuring broader access to high-quality healthcare services.

By developing a CNN-based diagnostic system for X-rays, MRIs, and CT scans, this study aims to contribute to the ongoing transformation of medical imaging. The project focuses on improving the interpretability and reliability of AI models, ensuring they can be effectively integrated into clinical practice [8]. Ultimately, this work seeks to demonstrate that AI-powered diagnostics can enhance healthcare efficiency, reduce diagnostic errors, and support medical professionals in providing better patient care.

B. Related Works

Our research focuses on adapting a pre-trained convolutional neural network (CNN) model for general-use medical diagnostics, specifically for detecting anomalies in X-rays, MRIs, and CT scans using PyTorch. The growing body of research on CNN applications in medical imaging underscores the potential of deep learning models to enhance diagnostic accuracy, streamline workflows, and assist radiologists in detecting diseases more efficiently. A key component of our approach is **transfer learning**, which allows us to fine-tune a pre-trained CNN for medical imaging tasks, reducing the need for extensive labeled datasets while improving model performance. This method aligns with previous work, such as that by Varma et al., who demonstrated that transfer learning significantly enhances CNN-based image classification [8].

Beyond transfer learning, CNNs have shown expert-level performance across multiple medical imaging domains. In dermatology, CNN-based models have matched or surpassed human specialists in classifying skin conditions, proving that deep learning can be adapted beyond traditional radiology applications [2]. Similarly, in oncology, AI models such as the **Chief AI system** developed at Harvard achieved a **94% accuracy rate** in cancer detection, demonstrating the capability of CNNs to assist in early cancer diagnosis with high precision [9]. CNNs have also been widely explored in neurology, where they have been successfully applied to detect **Alzheimer's-related tauopathy**, showcasing their ability to interpret intricate pathological patterns in complex neuroimaging data [10].

Additionally, CNNs have contributed to advancements in MRI imaging, particularly in **motion correction and artifact reduction**, which improve the quality of MRI scans and minimize the need for repeat imaging, as shown in recent studies [12]. These collective findings reinforce the **broad applicability of CNNs in medical imaging diagnostics** and validate our approach in leveraging deep learning for automated anomaly detection. By incorporating these methodologies, our research aims to develop an AI-driven diagnostic tool that enhances medical imaging analysis, improves interpretability, and supports clinical decision-making.

C. Problem Definition

Our research explores the deployment of an image recognition algorithm, specifically a convolutional neural network (CNN), within the healthcare industry to enhance medical imaging diagnostics. The distinctiveness of this project lies in its approach: leveraging transfer learning by adapting pre-trained CNN models to create a general-use diagnostic tool

To facilitate model training, we utilize a medical imaging dataset, ensuring access to high-quality labeled data. The images undergo preprocessing to enhance compatibility with our CNN, which is implemented using PyTorch. We seek to maximize model accuracy and generalizability across different imaging modalities through careful fine-tuning, including hyperparameter optimization and augmentation techniques.

II. METHODOLOGY

A. Dataset

This study utilizes three distinct medical imaging datasets for disease classification:

- **Lung Cancer Image Dataset: A Comprehensive Collection 2024** [11] – Consists of **high-resolution CT scan images** for lung cancer classification. The dataset is divided into four distinct classes:
 - **Adenocarcinoma** – A common form of lung cancer originating in mucus-producing cells.
 - **Large Cell Carcinoma** – A rapidly growing lung cancer type appearing in any lung region.
 - **Normal** – Healthy lung images serving as control samples.
 - **Squamous Cell Carcinoma** – A cancer developing in the flat cells lining the airways.
- **Pneumonia Chest X-ray Dataset** – Contains 5,863 X-ray images (JPEG) classified into two categories: Pneumonia and Normal. The dataset is organized into three subsets (train, test, validation). The images were selected from retrospective cohorts of pediatric patients aged one to five years old from Guangzhou Women and Children's Medical Center. [12]
- **Tuberculosis Chest X-ray Dataset** – Contains 7,000 X-ray images (3,500 TB-positive and 3,500 normal). The dataset was compiled by researchers from Qatar University, the University of Dhaka, and their collaborators. The TB images were sourced from publicly accessible datasets and the NIAID TB portal program. [13]

Each dataset was split into training, validation, and testing sets to facilitate model evaluation.

B. Data Preprocessing

To optimize the datasets for **CNN-based image recognition**, a preprocessing pipeline was implemented using **PyTorch's torchvision.transforms**. The key preprocessing steps include:

- **Resizing:** Images were resized to ensure uniform input dimensions.
- **Grayscale Conversion:** Images were converted to a **single-channel grayscale format** to reduce computational complexity while preserving diagnostic features.
- **Normalization:** Pixel values were **scaled to the [0,1] range** by dividing by 255 to aid neural network convergence.
- **Data Augmentation:** To enhance model generalization, **random horizontal flipping** and **random rotation** were applied.
- **Mean-Std Normalization:** The dataset **mean and standard deviation** were computed and applied for stabilization during training.

These preprocessing steps ensure that the datasets are **well-structured, optimized for deep learning, and suitable for medical image classification**.

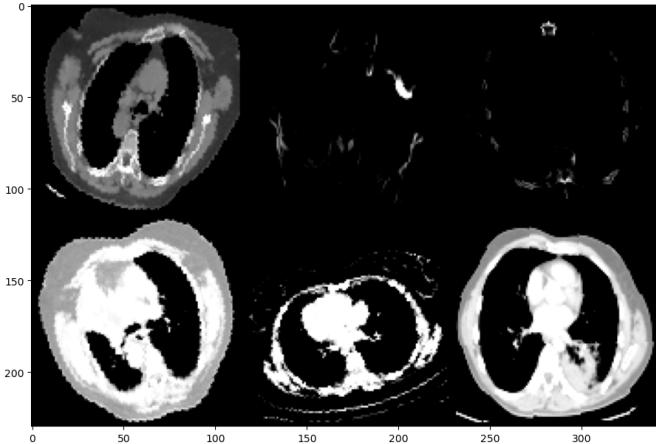


Fig. 1. Preprocessed lung CT scan images used for CNN-based image recognition. The preprocessing pipeline, implemented using PyTorch's `torchvision.transforms`, includes resizing for uniform input dimensions, grayscale conversion for reduced computational complexity, and pixel normalization to the [0,1] range for improved neural network convergence. Additionally, data augmentation techniques such as random horizontal flipping and rotation were applied to enhance model generalization. Mean-standard deviation normalization ($\text{mean} = 0.3230$, $\text{std} = 0.2176$) was performed to stabilize training. These steps optimize the dataset for deep learning-based lung cancer classification.

C. Model Architecture and Training

A **Convolutional Neural Network (CNN)-based approach** was employed using **transfer learning** with a **pre-trained DenseNet-121 model**. Three separate models were trained, each specialized for one of the three datasets (lung cancer,

pneumonia, tuberculosis). The modifications and training approach are detailed below:

1) Model Architecture:

- The **pre-trained DenseNet-121** model was adapted.
- The **fully connected classifier layer** was replaced with a **linear classifier** for multi-class output (four classes for lung cancer, two classes each for pneumonia and tuberculosis).
- A **cross-entropy loss function** was utilized for classification.
- The **SGD optimizer with momentum (0.9) and learning rate decay** was applied for stable training.

2) Training Process:

- Each model was trained separately on its respective dataset using a **batch size of 32**.
- Training was conducted on a **GPU** (if available) for faster convergence.
- A **learning rate scheduler** adjusted the learning rate every 7 epochs to mitigate overfitting.
- The validation sets were used for hyperparameter tuning.

D. Evaluation Metrics

To rigorously assess each model's ability to differentiate between true-positive and true-negative cases, this study employs a comprehensive set of evaluation metrics. Specifically, we focus on **precision, recall (sensitivity), F1-score**, and insights derived from the **confusion matrix**.

Precision and recall are fundamental in evaluating the model's ability to correctly identify true-positive cases. Their respective formulas are:

$$\text{Precision} = \frac{TP}{TP + FP} \quad (1)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (2)$$

The **F1-score**, which is the harmonic mean of precision and recall, quantifies the trade-off between false positives and false negatives and is defined as:

$$\text{F1-score} = 2 \cdot \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (3)$$

Additionally, the **confusion matrix** provides a detailed breakdown of the model's performance, offering valuable insights into its strengths and weaknesses. This tool is especially crucial in classification tasks where the cost of misclassification varies in severity, such as medical diagnosis.

III. RESULTS

A. Overall Model Performance

Three separate CNN-based models were trained for lung cancer, pneumonia, and tuberculosis classification. The overall test accuracies achieved for each model are:

- **Lung Cancer Model:** 90.48%
- **Pneumonia Model:** 91.83%
- **Tuberculosis Model:** 99.84%

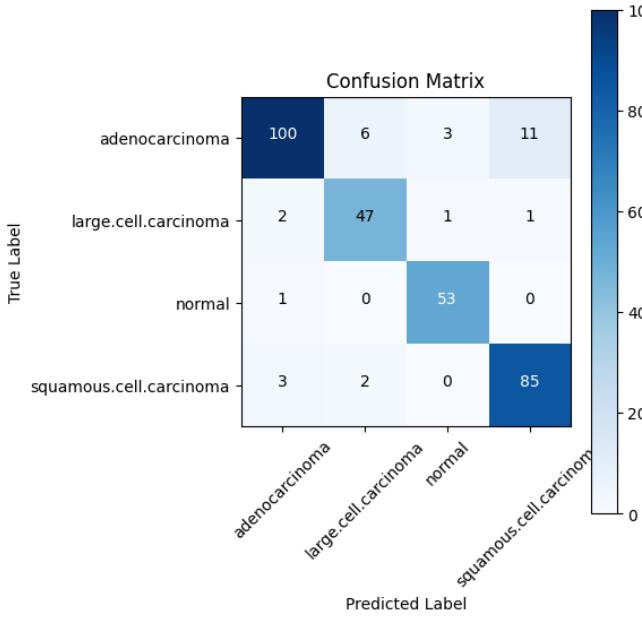


Fig. 2. Confusion matrix illustrating the performance of the CNN-based lung cancer classification model. The model demonstrates high classification accuracy for normal cases (98%) and squamous cell carcinoma (94%), but adenocarcinoma exhibits a slightly lower recall (83%).

These results demonstrate the effectiveness of the models in distinguishing between various medical conditions based on imaging data.

B. Lung Cancer Classification Results

The performance of our CNN-based lung cancer classification model was evaluated using a test dataset of 315 images, achieving an overall test accuracy of extbf{90.48%}. This demonstrates the model's effectiveness in distinguishing between adenocarcinoma, large cell carcinoma, squamous cell carcinoma, and normal lung conditions using CT scans.

1) Confusion Matrix Analysis: The confusion matrix provides insights into the model's classification performance:

- **Adenocarcinoma:** Correctly classified: 100, Misclassified as large cell carcinoma: 6, normal: 3, and squamous cell carcinoma: 11.
- **Large Cell Carcinoma:** Correctly classified: 47, Misclassified as adenocarcinoma: 2, normal: 1, squamous cell carcinoma: 1.
- **Normal Cases:** Correctly classified: 53, Misclassified as adenocarcinoma: 1.
- **Squamous Cell Carcinoma:** Correctly classified: 85, Misclassified as adenocarcinoma: 3, large cell carcinoma: 2, normal: 0.

C. Pneumonia Classification Results

The pneumonia classification model was evaluated using a test dataset of 624 images, achieving an overall test accuracy of **91.83%**.

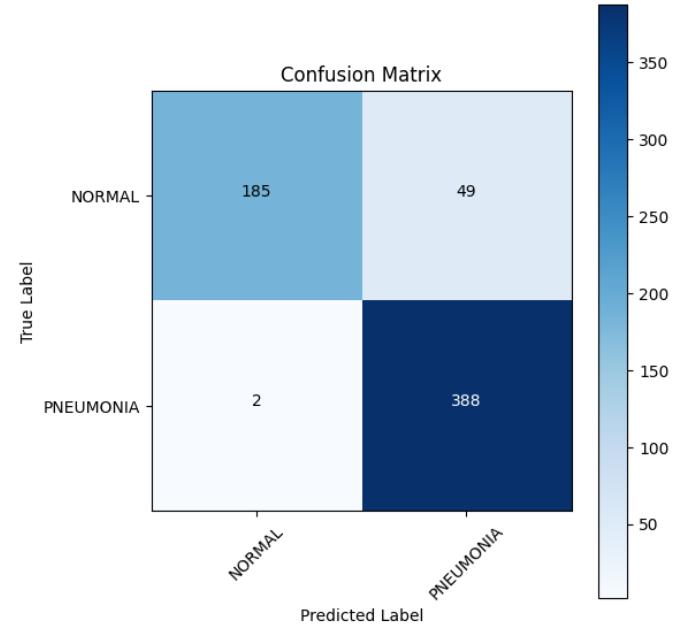


Fig. 3. Confusion matrix illustrating the performance of the CNN-based pneumonia classification model. The model achieved a recall of 99% for pneumonia cases, ensuring a minimal number of false negatives. However, normal cases exhibited a lower recall (79%), with some misclassification occurring.

- **Normal:** Correctly classified: 185, Misclassified as pneumonia: 49.
- **Pneumonia:** Correctly classified: 388, Misclassified as normal: 2.

D. Tuberculosis Classification Results

The tuberculosis classification model was evaluated using a test dataset of 630 images, achieving an outstanding test accuracy of **99.84%**.

- **Normal:** Correctly classified: 525, Misclassified as tuberculosis: 0.
- **Tuberculosis:** Correctly classified: 104, Misclassified as normal: 1.

E. Summary of Results

The tuberculosis classification model achieved near-perfect accuracy, indicating excellent performance in distinguishing between normal and tuberculosis cases. The pneumonia model exhibited strong performance, though with a slightly lower recall for normal cases. The lung cancer classification model performed well overall, though it showed some misclassification for adenocarcinoma cases. These results highlight the strengths and potential areas for improvement in medical image-based classification models.

IV. CONCLUSION

This study explored the application of **CNN-based deep learning models** for medical image classification, specifi-

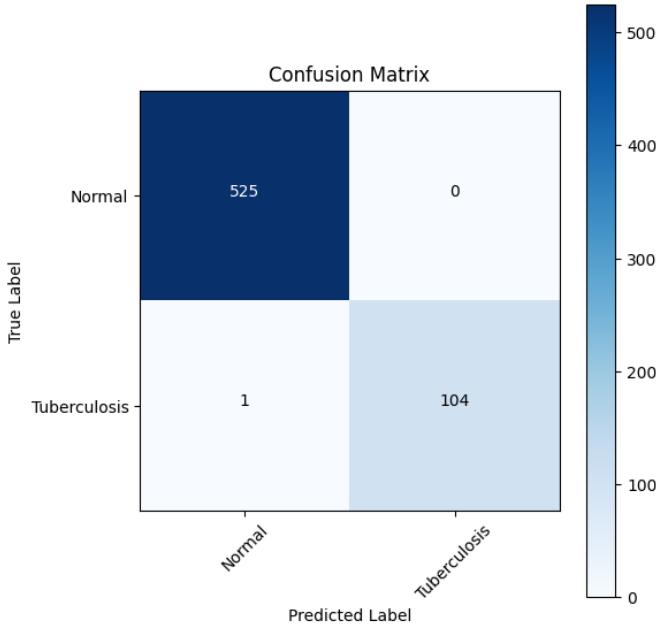


Fig. 4. Confusion matrix illustrating the performance of the CNN-based tuberculosis classification model. The model achieved near-perfect classification accuracy, with only one misclassification observed in the tuberculosis class.

cally targeting **lung cancer, pneumonia, and tuberculosis detection**. By leveraging transfer learning with a pre-trained **DenseNet-121** model, the study demonstrated that deep learning can achieve **high accuracy** in diagnosing medical conditions from X-ray and CT scan images.

The **tuberculosis classification model** achieved an outstanding **99.84%** accuracy, highlighting the potential of CNNs for real-world medical applications. The pneumonia model performed well, though it exhibited **slightly lower recall** for normal cases, indicating potential room for improvement. The lung cancer model successfully classified different cancer types but faced **challenges in distinguishing adenocarcinoma from other forms**. These findings emphasize the effectiveness of AI-powered diagnostics in assisting radiologists and healthcare professionals by reducing interpretation time and improving diagnostic accuracy.

Despite these promising results, several challenges remain. The **variability in real-world medical imaging datasets**, potential model biases, and the need for external validation suggest that further refinement is required before deployment in clinical settings. However, this research reinforces the **transformative potential of AI** in healthcare and sets the stage for **future advancements in automated medical diagnostics**.

V. FUTURE WORK

Future research should focus on improving the generalizability of the models by incorporating larger and more diverse datasets. Enhancing the model's robustness through additional

data augmentation techniques and domain adaptation methods could reduce misclassification rates. Future studies could also explore the potential of multi-modal learning, where CT scans, X-ray images, and clinical data are combined to enhance diagnostic accuracy. Lastly, deploying these models in real-world clinical settings and evaluating their impact on healthcare workflows would be a crucial step toward practical implementation.

VI. LIMITATIONS

Despite the promising results, the models presented in this study have several limitations. First, the datasets used may not fully represent the variability found in real-world clinical settings, leading to potential biases in model predictions. Additionally, while deep learning models excel at pattern recognition, they still struggle with rare cases and subtle abnormalities that require human expertise. The lack of external validation on independent datasets limits the model's generalizability. Computational resource constraints may also hinder real-time deployment, particularly in low-resource healthcare settings.

VII. ETHICAL CONSIDERATIONS

AI-based diagnostic models must be deployed responsibly to ensure fairness and transparency. Bias in training data can lead to disparities in diagnostic accuracy across different patient groups. Ensuring patient privacy and compliance with regulations is crucial when handling medical data. AI should assist, not replace, human expertise to prevent over-reliance and diagnostic errors. Clear guidelines for AI use and continuous monitoring in clinical settings are necessary to maintain ethical and responsible deployment.

VIII. ACKNOWLEDGMENTS

The model was curated by the team consisting of Nathan Wan, Kevin Du, Millicent Song, Juna Kim, Besma Serrai, Liam McQuay, Matthew Louis Li, Artemiy Vishnyakov and Sabrina Lee who helped this project come to fruition.

REFERENCES

- [1] T. Pang, P. Li, and L. Zhao, "A survey on automatic generation of medical imaging reports based on deep learning," *Biomedical engineering online*, vol. 22, no. 1, pp. 48–16, 2023.
- [2] A. Esteva *et al.*, "Dermatologist-level classification of skin cancer with deep neural networks," *Nature (London)*, vol. 542, no. 7639, pp. 115–118, 2017.
- [3] K. A. P. Razzaghi and P. Bayat, "Learning spatial hierarchies of high-level features in deep neural network," *Journal of visual communication and image representation*, vol. 70, pp. 102 817–, 2020.
- [4] J. K. Winkler *et al.*, "Association between surgical skin markings in dermoscopic images and diagnostic performance of a deep learning convolutional neural network for melanoma recognition," *JAMA dermatology (Chicago, Ill.)*, vol. 155, no. 10, pp. 1135–1141, 2019.
- [5] F. W. Narongrit and J. V. Rispoli, "Editorial for 'preoperative prediction of axillary lymph node metastasis in breast cancer using cnn based on multiparametric mri,'" *Journal of magnetic resonance imaging*, vol. 56, no. 3, pp. 710–711, 2022.
- [6] X. Liu *et al.*, "A comparison of deep learning performance against health-care professionals in detecting diseases from medical imaging: a systematic review and meta-analysis," *The Lancet. Digital health*, vol. 1, no. 6, pp. e271–e297, 2019.

- [7] K. Khalili *et al.*, “Convolutional neural networks versus radiologists in characterization of small hypoattenuating hepatic nodules on ct: a critical diagnostic challenge in staging of colorectal carcinoma,” *Scientific reports*, vol. 10, no. 1, pp. 15 248–15 248, 2020.
- [8] S. M. B. S. R. P. M. K. P. B. S. Varma, S. Paturu and N. V. Krishna, “Sldcnet: Skin lesion detection and classification using full resolution convolutional network-based deep learning cnn with transfer learning,” *Expert systems*, vol. 39, no. 9, 2022.
- [9] M. B. Nierengarten, “New ai model shows promise for cancer diagnosis,” *Cancer*, vol. 131, no. 3, pp. e35 715–n/a, 2025.
- [10] R. Divya and R. S. S. Kumari, “Detection of alzheimer’s disease from temporal lobe grey matter slices using 3d cnn,” *The imaging science journal*, vol. 70, no. 8, pp. 578–587, 2022.
- [11] Kabil, “Lungcancer4types-imagedataset multiclass lung cancer image dataset for research and analysis,” *Kaggle [online]*, 2024.
- [12] D. S. K. et al., “Identifying medical diagnoses and treatable diseases by image-based deep learning,” *cell*,” *Cell*, vol. 172, pp. 1122–1131, 2018.
- [13] M. A. K. K. R. I. K. F. I. Z. B. M. M. A. A. M. E. H. C. T. Rahman, A. Khandakar, ““reliable tuberculosis detection using chest x-ray with deep learning, segmentation and visualization”.” *IEEE Access 2020*, vol. 8, pp. 191 586 – 191 601, 2020.

Copyright Detection in Large Language Models: An Ethical Approach to Generative AI Development

David Szczecina

University of Waterloo

david.szczecina@uwaterloo.ca

Senan Gaffori

University of Waterloo

senan.gaffori@uwaterloo.ca

Edmond Li

University of Waterloo

e26li@uwaterloo.com

Abstract—The widespread use of Large Language Models (LLMs) raises critical concerns regarding the unauthorized inclusion of copyrighted content in training data. Existing detection frameworks, such as DE-COP, are computationally intensive, and largely inaccessible to independent creators. As legal scrutiny increases, there is a pressing need for a scalable, transparent, and user-friendly solution. This paper introduce an open-source copyright detection platform that enables content creators to verify whether their work was used in LLM training datasets. Our approach enhances existing methodologies by facilitating ease of use, improving similarity detection, optimizing dataset validation, and reducing computational overhead by 10-30% with efficient API calls. With an intuitive user interface and scalable backend, this framework contributes to increasing transparency in AI development and ethical compliance, facilitating the foundation for further research in responsible AI development and copyright enforcement.

I. INTRODUCTION

A. Motivation

Large Language Models (LLMs) such as GPT-4 and Claude have revolutionized natural language processing, but also raise legal and ethical concerns about the unauthorized use of copyrighted content in training datasets [1]. Proprietary models often rely on large-scale web scraping [2], incorporating copyrighted material without clear consent mechanisms, compensation, and intellectual property protection [3].

A major concern is the lack of compensation for content creators whose work is used without permission. Legal frameworks for AI copyright enforcement are rapidly evolving, with landmark cases like New York Times v. OpenAI [4] bringing increased scrutiny to dataset curation. Transparency in AI training datasets is essential to ensure responsible and ethical development. Research indicates that as models increase in size, memorization tendencies become more pronounced, particularly in models exceeding 100 billion parameters [4], increasing the risk of unauthorized reproduction of copyrighted content.

Current detection methods, such as plagiarism checkers and statistical techniques, struggle to identify subtly paraphrased copyrighted content [2] [5]. While frameworks such as DE-COP offer promising approaches, they remain computationally expensive and complex; making them impractical for independent creators and smaller organizations. A scalable, cost-effective, and user-friendly solution is needed to verify whether copyrighted works have been used in LLM training datasets.

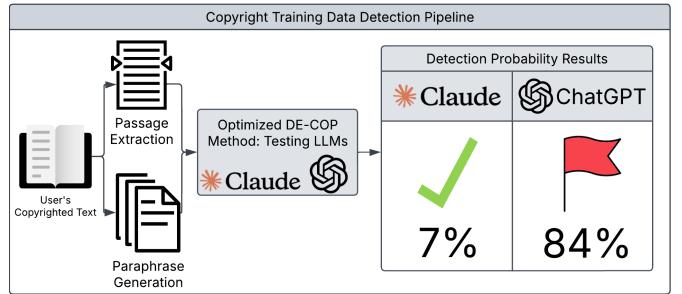


Fig. 1. Unique passages are extracted and paraphrased from a users content, next an LLM is prompted to determine the original passage. Final scores show the probability of the copyrighted content being used in training the LLM

B. Related Works

The detection of copyrighted content in LLM training datasets has been the subject of increasing research attention, particularly as legal and ethical concerns surrounding dataset curation intensify. While traditional plagiarism detection tools struggle to identify AI-generated reproductions of proprietary content [2], several machine learning-based approaches have been proposed to address this issue.

Membership inference attacks [6] analyze a model's confidence scores to determine whether a given text sample was likely included in the training data. Although effective in controlled experiments, this approach requires adversarial access to the model and often produces inconclusive results due to dataset augmentation and model fine-tuning techniques. Similarly, perplexity-based analysis is another detection approach by evaluating how confidently an LLM predicts a passage of text [7]. Low perplexity scores suggest memorization, however, this method struggles to distinguish between legally sourced and unauthorized content, making it unreliable for copyright enforcement. Another proposed approach is digital watermarking [8], where imperceptible markers are embedded into text data before model training. While useful for tracking known copyrighted works, watermarking is ineffective against existing datasets that were scraped from the web and fails to detect content that has been paraphrased or restructured.

A more recent approach, DE-COP: Detecting Copyrighted Content in Language Models Training Data, [2], introduces a method to determine whether a language model has memorized copyrighted content. Unlike statistical approaches, DE-COP introduces a multiple-choice question-answering framework,

where an LLM must distinguish an original verbatim passage from paraphrased alternatives. If a model consistently selects the correct passage, this suggests that the content was likely included in its training data. An overview of the DE-COP system is illustrated in Figure 2. Despite its advantages, DE-COP is computationally expensive, requiring approximately 590 seconds per book for open-source models (LLaMA-2 70B) [9] and 331 seconds on ChatGPT [2] [10]. Methods such as Min-K%-Prob [7], Prefix Probing [11] and Name Cloze Task [12] only required 13-17 seconds to perform the same task [2]. Additionally, the datasets presented in DE-COP were found to contain NULL values, errors message outputs, half finished sentences, and new paraphrases ranged from being 20% to 250% as long as the original passage [2]. DE-COP lacks robust features to handle these errors in its own dataset, and its evaluation metrics were based on questionable data, leaving lots of room for improvements.

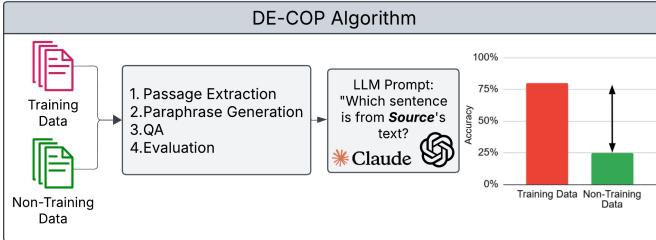


Fig. 2. DE-COP System Overview

While previous methods provide partial solutions to the problem of detecting copyrighted content in LLM training data, they often fall short in generalization and effectiveness. DE-COP introduced a black-box-compatible alternative that significantly improves detection accuracy [2]. However, optimizing its computational efficiency and reducing selection biases remains an open challenge for future work.

C. Problem Definition

Despite the concerns for copyright material in LLM training data, existing copyright detection methods remain insufficient and inaccessible. Traditional plagiarism detection tools struggle to identify paraphrased or subtly modified copyrighted content, making enforcement difficult [2]. Additionally, computationally intensive frameworks such as DE-COP, are impractical for independent content creators due to their technical complexity and high computational costs [13]. The absence of cost-effective and user-friendly solutions further limits the ability to verify whether copyrighted works have been used in LLM training. This paper introduces an open-source framework that enhances dataset validation, improving similarity detection, and optimizes computational efficiency. By significantly reducing processing costs while maintaining detection accuracy, our approach provides a scalable and accessible platform for copyright verification. This initiative ensures AI transparency, promotes fair compensation for content creators, and supports ethical AI development in the rapidly evolving landscape of generative AI.

II. METHODOLOGY

This project features a web-based UI where users can submit content for evaluation. The backend evaluation system, runs a multi-layered evaluation workflow integrating passage extraction, paraphrase generation, question-answering, multiple-choice evaluation, and statistical analysis to detect copyrighted content in LLM training data. A vector store maintains a record of previously evaluated content, allowing the system to check for duplicates, avoiding redundant evaluations, as well as search through past evaluations. Users can access a dashboard and analytics page to view evaluation histories and check accuracy metrics. An overview of the system architecture is illustrated in Figure 3.

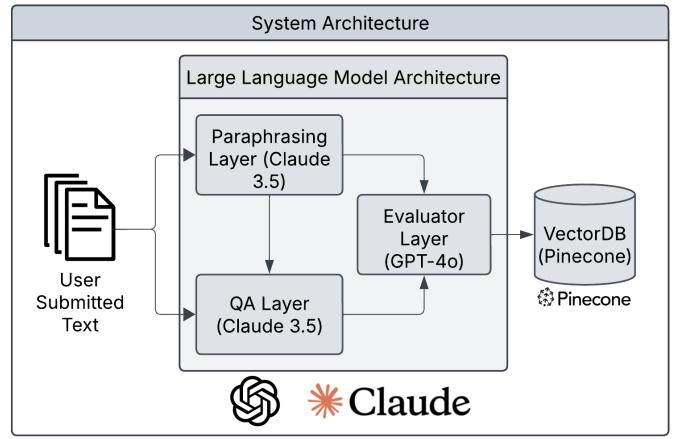


Fig. 3. System Architecture Diagram

A. Passage Extraction

Selecting highly unique passages enhances the accuracy and effectiveness of detecting copyrighted content in language model training data [14]. Unique passages minimize the risk of incorporating common phrases and generic text that may not sufficiently challenge the model's memorization capabilities [15]. To identify these passages, the BM25 algorithm [16] was employed to vectorize passages within the document and calculate similarity scores between them. By treating each passage as a query against the entire document, BM25 assigned scores based on term frequency and inverse document frequency. Passages with the lowest BM25 scores, indicating minimal similarity to other passages, were considered the most unique. These high-uniqueness passages were prioritized for use in the evaluation layer, providing a robust foundation for detecting memorized content.

B. Paraphrase Generation

The paraphrase generation layer is implemented using LangGraph's StateGraph to create a modular and dynamic workflow. This layer utilizes the Claude 3.5 Sonnet model [17] via the ChatAnthropic API with a temperature setting of 0.7, ensuring a balance between creativity and control in generating paraphrases. Unlike the original DE-COP approach, which applies standard paraphrasing prompts, our method

introduces specific paraphrasing strategies, including passive voice conversions, question-based restructuring, and language simplification. These templates promote greater diversity in paraphrases [18], enhancing the robustness of the evaluation by reducing model prediction patterns. Additionally, the implementation proposes XML formatting for paraphrases to support integration with instructional models, offering improved compatibility and structured data handling not present in the original DE-COP method.

C. Question-Answering

The QA layer is also built using LangGraph’s StateGraph, facilitating an automated workflow that handles both “create” and “format” modes for generating evaluation questions. While the original DE-COP [2] method primarily focused on generating standardized multiple-choice questions, our implementation expands functionality by allowing the creation of custom questions that use exact text from the input content. The QA layer uses the ChatAnthropic model to generate questions in a structured JSON format, improving downstream processing and maintaining output consistency. This added flexibility enhances the evaluation’s accuracy by testing the model’s memorization across varied question formats, contributing to a more thorough assessment of the model’s exposure to copyrighted content.

D. Multiple-Choice

The multiple-choice layer employs LangGraph to manage answer selection and evaluation workflows. In contrast to the original DE-COP’s [2] exhaustive approach of generating all permutations of answer choices, our initial implementation used a simplified randomization strategy to prevent selection bias. However, we propose an enhancement that includes a dedicated permutation function to fully automate all possible answer orderings within LangGraph. The evaluation prompts are designed to elicit concise, formatted responses from the model, minimizing noise and ensuring clarity in the output. Incorporating full permutation handling would better strengthen the mitigation of selection biases in model responses.

E. Evaluation

The evaluation layer integrates multiple components, including paraphrase generation, question answering, multiple-choice testing, and statistical analysis, using GPT-4o [10] via LangGraph. Our implementation extends upon DE-COP’s framework by incorporating advanced statistical methodologies such as receiver-operating characteristic (ROC) curve analysis, area under the curve (AUC) scoring, and hypothesis testing. This layer provides deeper insights into performance through robust statistical methods. A key enhancement over previous methodologies is the introduction of a permutation function which generates all answer permutations; mitigating selection biases in LLMs. The evaluation prompts guide the model through a structured evaluation process, emphasizing precise and formatted responses. These enhancements create a more modular and statistically robust framework, improving the accuracy and reliability of detecting copyrighted content in LLM training data.

F. Logging System and Similarity Search

To enable content tracking and retrieval, the system incorporates Pinecone, a serverless vector database. Documents are embedded using all-MiniLM-L6-v2 [19] (embedding model from HuggingFace) which offers a strong balance of embedding quality and efficiency. The model generates 384-dimensional embeddings to support fast and accurate approximate nearest neighbour (ANN) searches, while integrating seamlessly with Pinecone and LangGraph. Metadata attributes such as copyright ownership, evaluation timestamps, evaluation results, and content type, are stored directly in Pinecone as key-value pairs. Logging metadata enables quick access and tractability during content evaluations without requiring an external database. The ingestion pipeline is designed for single-document processing, embedding each submission and storing it with a unique identifier. To evaluate content, the system compares new submissions against stored vectors, retrieving the most similar documents and their metadata. This streamlined vectorized approach supports the goal of creating an open-source API that logs copyrighted content appearing in LLM training data; promoting transparency and accountability in AI development.

G. Data Processing Improvements

An analysis of DE-COP’s dataset revealed several inconsistencies such as NULL values, API output errors, inconsistent formatting, and extreme variations in passage length [2]. These inconsistencies negatively impacted accuracy, skewed model predictions and increased token usage by up to 50%. To address this, a preprocessing pipeline was implemented using SBERT embeddings [20] and cosine similarity, ensuring that paraphrases retain semantic integrity, and any invalid passages are filtered out. Additionally, passage lengths are normalized to prevent instances where paraphrases are excessively short or long, improving paraphrase consistency. These enhancements eliminated inconsistencies in DE-COP’s dataset [2], providing results that are more reproducible and statistically sound.

To further reduce API costs, the multiple-choice selection was expanded from three to four paraphrased options, reducing the probability of the original passage being randomly selected by 20%. By decreasing the probability of Type I error, the total number of passages requiring evaluation can be decreased without compromising the experiment’s statistical power or significance. Consequently, this optimization substantially lowers overall API consumption by requiring less passages to be evaluated.

III. RESULTS

Our proposed framework demonstrates significant improvements in detection accuracy, computational efficiency, and accessibility over existing methodologies. By providing our open-source solution as a hosted platform, we remove technical barriers, promoting ease of use and access to individual content creators.

The multi-layered workflow, which integrates passage extraction, paraphrase generation, question-answering, and multiple-choice testing, effectively differentiates between memorized

(copyrighted) and non-memorized text. By integrating a pre-screening pipeline using SBERT embeddings, cosine similarity, and normalized passage lengths, errors are caught and filtered out; enhancing reproducibility. The multiple-choice evaluation layer, with a streamlined randomization strategy and restructuring of question format reduced API consumption by 10-30%. Additionally, the Pinecone vector store enhances scalability and duplicate detection, avoiding redundant evaluations. These enhancements provide a scalable and practical solution that outperforms existing approaches, such as DE-COP, supporting ethical AI development and fair compensation for content creators.

IV. CONCLUSION

This paper introduces an open-source framework for detecting copyrighted content in LLM training datasets, addressing key limitations in accessibility, detection accuracy, and cost efficiency found in previous approaches such as DE-COP. By enhancing similarity detection, refining dataset validation, and optimizing computational efficiency, our system provides a scalable and accessible solution for copyright verification. Through our user-friendly interface, content creators can easily determine whether their work was appropriated for AI development, without a high technical barrier to entry. By promoting transparency and encouraging accountability, our system ultimately paves the way for ethical AI development.

V. FUTURE WORK

Future research may focus on developing methods for selective knowledge removal, such as *Unlearn* [21], to enable the erasure of copyrighted content from LLMs. This knowledge removal technique could be implemented for some of the standard LLM pretraining datasets such as *C4* [22] and *Pile* [23]. The legal implications of dataset memorization also warrant further investigation, particularly as AI copyright regulations continue to evolve. Additionally, expanding the scalability and adoption of our platform across different AI models and regulatory frameworks will be crucial for broader impact.

ACKNOWLEDGMENTS

This research was enabled in part by funding, resources, and support provided by Wat.AI, the Waterloo AI Institute, and the Sedra Student Design Centre.

REFERENCES

- [1] J. Xu, S. Li, Z. Xu, and D. Zhang, “Do llms know to respect copyright notice?” *arXiv preprint arXiv:2411.01136*, 2024.
- [2] A. V. Duarte, X. Zhao, A. L. Oliveira, and L. Li, “De-cop: Detecting copyrighted content in language models training data,” *arXiv preprint arXiv:2402.09910*, 2024.
- [3] J. Guo, Y. Li, R. Chen, Y. Wu, C. Liu, Y. Chen, and H. Huang, “Rag: Towards copyright protection for knowledge bases of retrieval-augmented language models,” *OpenReview*, 2023.
- [4] J. Freeman, C. Rippe, and E. Debenedetti, “Exploring memorization and copyright violation in frontier llms: A study of the new york times v. openai 2023 lawsuit,” *arXiv preprint arXiv:2412.06370*, 2024.
- [5] H. Shao, Z. Xu, S. Duan, and D. Zhang, “Measuring copyright risks of large language models via partial information probing,” *arXiv preprint arXiv:2409.13831*, 2024.
- [6] H. Tan, M. Duan, D. Liu, and L. Zhou, “Rethinking literary plagiarism in llms through the lens of copyright laws,” in *Proceedings of the 16th Asian Conference on Machine Learning*, 2023.
- [7] W. Shi, T. Ma, Y. Liu, and J. Zhang, “Detecting pretraining data from large language models,” *arXiv preprint arXiv:2310.16789*, 2023.
- [8] C. Kirchenbauer, J. Geiping, P. Carlini, and T. Jagielski, “Watermarking large language models,” in *Proceedings of the International Conference on Machine Learning (ICML)*, 2023.
- [9] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, D. Bikel, L. Blecher, C. C. Ferrer, M. Chen, G. Cucurull, D. Esioibu, J. Fernandes, J. Fu, W. Fu, B. Fuller, C. Gao, V. Goswami, N. Goyal, A. Hartshorn, S. Hosseini, R. Hou, H. Inan, M. Kardas, V. Kerkez, M. Khabsa, I. Kloumann, A. Korenev, P. S. Koura, M.-A. Lachaux, T. Lavril, J. Lee, D. Liskovich, Y. Lu, Y. Mao, X. Martinet, T. Mihaylov, P. Mishra, I. Molybog, Y. Nie, A. Poultion, J. Reizenstein, R. Rungta, K. Saladi, A. Schelten, R. Silva, E. M. Smith, R. Subramanian, X. E. Tan, B. Tang, R. Taylor, A. Williams, J. X. Kuan, P. Xu, Z. Yan, I. Zarov, Y. Zhang, A. Fan, M. Kambadur, S. Narang, A. Rodriguez, R. Stojnic, S. Edunov, and T. Scialom, “Llama 2: Open foundation and finetuned chat models,” *arXiv preprint arXiv:2307.09288*, 2023.
- [10] OpenAI, “Introducing chat-gpt,” <https://openai.com/blog/chatgpt>, 2022, accessed: 2025-02-20.
- [11] Z. Liu, A. Qiao, W. Neiswanger, H. Wang, B. Tan, T. Tao, J. Li, Y. Wang, S. Sun, O. Pangarkar, R. Fan, Y. Gu, V. Miller, Y. Zhuang, G. He, H. Li, F. Koto, L. Tang, N. Ranjan, Z. Shen, X. Ren, R. Iriondo, C. Mu, Z. Hu, M. Schulze, P. Nakov, T. Baldwin, and E. P. Xing, “Llm360: Towards fully transparent open-source llms,” *arXiv preprint arXiv:2312.06550*, 2023.
- [12] K. K. Chang, M. Cramer, S. Soni, and D. Bamman, “Speak, memory: An archaeology of books known to chatgpt/gpt4,” *arXiv preprint arXiv:2305.00118*, 2023.
- [13] S. Samsi, D. Zhao, J. McDonald, B. Li, A. Michaleas, M. Jones, W. Bergeron, J. Kepner, D. Tiwari, and V. Gadepally, “From words to watts: Benchmarking the energy costs of large language model inference,” *arXiv preprint arXiv:2310.03003*, 2023, <https://arxiv.org/pdf/2310.03003.pdf>.
- [14] A. de Wynter, X. Wang, A. Sokolov, Q. Gu, and S.-Q. Chen, “An evaluation on large language model outputs: Discourse and memorization,” *arXiv preprint arXiv:2304.08637*, 2023.
- [15] J. Lin and M. Ma, “A few brief notes on deepimpact, coil, and a conceptual framework for information retrieval techniques,” *arXiv preprint arXiv:2106.14807*, 2021.
- [16] S. Robertson and H. Zaragoza, “The probabilistic relevance framework: Bm25 and beyond,” *Foundations and Trends in Information Retrieval*, vol. 3, no. 4, pp. 333–389, 2009.
- [17] Anthropic, “Claude 2,” <https://www.anthropic.com/news/clause-2>, 2023.
- [18] E. Bandel, R. Aharonov, M. Shmueli-Scheuer, I. Shnayderman, and N. Slonim, “Quality controlled paraphrase generation,” in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2022, <https://aclanthology.org/2022.acl-long.451>.
- [19] N. Reimers and I. Gurevych, “Making monolingual sentence embeddings multilingual using knowledge distillation,” in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 2020, pp. 4512–4525.
- [20] ———, “Sentence-bert: Sentence embeddings using siamese bert-networks,” *arXiv preprint arXiv:1908.10084*, 2019.
- [21] C. Eichler, T. Fischer, C. Sixt, and J. Yosinski, “Unlearn: Selective knowledge removal from large language models,” *arXiv preprint arXiv:2408.04140*, 2024.
- [22] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, and P. J. Liu, “Exploring the limits of transfer learning with a unified text-to-text transformer,” *Journal of Machine Learning Research*, vol. 21, no. 140, pp. 1–67, 2020. [Online]. Available: <https://jmlr.org/papers/v21/20-074.html>
- [23] G. Gao, S. Biderman, S. Black, L. Golding, H. He, and M. Shoeybi, “The pile: An 800gb dataset of diverse text for language modeling,” <https://arxiv.org/abs/2101.00027>, 2020.

DentAI Vision: AI-Powered Dental X-Ray Analysis for Enhancing Trust and Patient Education

Maha Kesibi
Queen's University
21mk93@queensu.ca

Leila Salem
Queen's University
22tt43@queensu.ca

Het Buddhdev
Queen's University
22hvb@queensu.ca

Kamran Jornacion
Queen's University
22qfc4@queensu.ca

Elliott Vince
Queen's University
elliott.vince@queensu.ca

Abstract—This study addresses the challenge of improving trust between patients and dental professionals by leveraging AI-driven analysis of panoramic dental X-rays. With dental caries being one of the most prevalent oral diseases, early and accurate detection is crucial for effective treatment and prevention. Deep learning algorithms were utilized to detect caries, specifically employing the YOLOv5-Small object detection model, optimized for real-time inference. The methodology involved dataset pre-processing, data augmentation techniques such as horizontal flipping, and model training on the DENTEX MICCAI 2023 dataset. In addition to automated diagnosis, a chatbot powered by DeepSeek-Llama and LangChain was integrated into the platform, providing users with reliable, evidence-based dental health information sourced from over 20 accredited references. The findings demonstrate that the proposed AI system can achieve high diagnostic accuracy while fostering patient education and transparency. This research highlights the potential of AI-powered dental diagnostics in reducing the need for costly second opinions, improving patient-dentist relationships, and promoting informed decision-making in oral healthcare.

I. INTRODUCTION

Oral diseases, including dental caries, are among the most prevalent health issues globally, affecting approximately 3.5 billion people. Specifically, dental caries impact around 2.3 billion individuals in their permanent teeth, making it one of the most common chronic diseases worldwide [1]. If left untreated, dental caries can lead to complications such as infections, pain, and tooth loss. Despite advancements in dental care, access to accurate, timely, and affordable diagnoses remains a significant challenge, particularly in low-income and underserved communities [2].

One of the key factors influencing patient outcomes in dentistry is trust between patients and dental professionals. Trust plays a crucial role in patient adherence to treatment plans and their willingness to seek care. However, negative experiences, misdiagnoses, and costly treatment plans often lead patients to seek second opinions, which can be expensive and time-consuming. Research suggests that patients who receive multiple opinions are more likely to adhere to treatment recommendations, yet many face barriers to accessing additional consultations due to geographic or financial constraints [3], [4].

Artificial intelligence (AI) and deep learning have demonstrated significant potential in enhancing diagnostic accuracy

in medical imaging. A comprehensive meta-analysis revealed that deep learning algorithms achieved a mean sensitivity of 96.3% and a mean specificity of 93.3% across various pathology identifications, underscoring their efficacy in medical diagnostics [5]. Regarding patient trust, the integration of AI into medical imaging has yielded mixed perceptions. Some studies indicate low trust in healthcare systems to use AI responsibly [3], while others show that nearly two-thirds of consumers would trust a diagnosis from AI over that of a human doctor [4]. These findings suggest that while AI has the potential to enhance diagnostic accuracy, fostering patient trust requires careful implementation and communication strategies. By leveraging AI, DentAI Vision aims to provide a low-cost, accessible second opinion for patients and support dentists in patient education and decision-making.

Start a New Chat

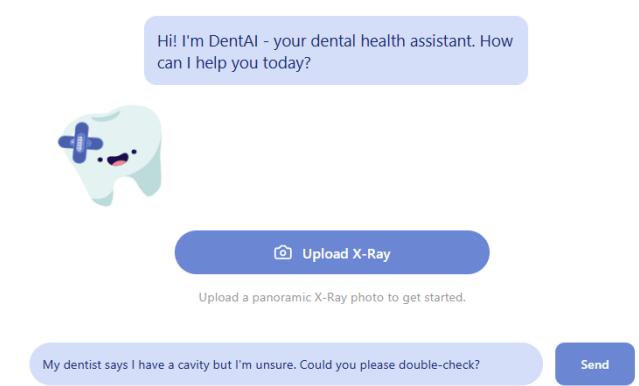


Fig. 1. DentAI Vision chat interface where users can initiate a session by uploading an X-ray and inputting their concerns or queries.

A. Problem Definition

Given the high prevalence of dental caries and the role of trust in treatment adherence, DentAI Vision addresses two core problems:

- 1) **Lack of Accessible Second Opinions** – Patients often seek confirmation of diagnoses before undergoing dental procedures. However, obtaining a second opinion can be expensive and time-consuming, discouraging patients from making informed decisions [?]. An AI-driven system that offers *instant, cost-free analysis of dental X-rays* can bridge this gap.
- 2) **Limited Patient Education and Transparency in Diagnoses** – Patients frequently struggle to understand dental diagnoses and treatment plans, contributing to *mistrust in healthcare providers*. Studies suggest that *visual aids and explainable AI* significantly improve patient comprehension and trust in medical recommendations [?]. *DentAI Vision* integrates a chatbot, *DeepSeek-Llama with LangChain*, trained on 20+ accredited dental resources, to provide users with accurate, evidence-based responses to their dental health queries.

While previous research has demonstrated the effectiveness of deep learning in medical imaging, existing AI-driven dental diagnostic tools often lack real-time patient interaction, transparency, and explainability. This research aims to enhance trust, accessibility, and patient education by combining state-of-the-art deep learning with interactive AI-driven explanations.

This paper explores the development, training, and evaluation of an AI-powered diagnostic system for detecting dental caries, employing YOLOv5-Small for real-time object detection. Furthermore, it investigates the role of LLM-powered chatbots in improving patient trust and engagement in AI-driven healthcare solutions.

By offering automated, unbiased diagnoses, explainable AI interactions, and a user-friendly web-based platform, DentAI Vision provides a scalable and accessible solution to improve trust between patients and dental professionals, reducing the need for expensive second opinions and fostering better oral healthcare outcomes.

II. RELATED WORKS

The integration of artificial intelligence (AI) into dental diagnostics has led to significant advancements, improving accuracy and efficiency in disease detection and treatment planning. AI-driven methods have demonstrated success in analyzing radiographic images, identifying dental caries, periodontal diseases, and other oral conditions. For instance, AI has been applied to diagnose oral diseases such as maxillary sinus conditions and caries using clinical data and diagnostic images, achieving performance comparable to or exceeding that of human experts [6], [7].

Object detection models, particularly YOLOv5, have been widely utilized in medical imaging due to their real-time inference capabilities. YOLOv5 has demonstrated strong diagnostic performance in detecting abnormalities such as developmental

dysplasia of the hip (DDH), kidney stones in CT images, and lung tumors [8]–[10]. These applications suggest the suitability of YOLO-based architectures for medical imaging tasks, including dental X-ray analysis, where real-time detection is valuable.

Several AI-powered dental diagnostic tools have also been developed. Overjet, an FDA-approved AI system, enhances dental X-ray visualization by marking cavities and other dental pathologies with color-coded overlays [11]. Similarly, Pearl AI utilizes computer vision to evaluate radiographic and 3D imagery, improving diagnostic accuracy [12]. Diagnocat employs AI to provide automated diagnoses and treatment planning [13]. While these tools are effective in assisting dental professionals, they primarily serve as clinical decision-support systems and do not focus on improving patient trust or integrating explainable AI for direct patient education. In contrast, our work aims to bridge this gap by combining real-time AI-driven dental X-ray analysis with an interactive chatbot that provides explainable insights and second opinions to patients.

Beyond AI, advancements in imaging technologies such as intraoral cameras and photothermal imaging radar have contributed to improved dental diagnostics. Intraoral cameras provide real-time imaging to aid in diagnosis and patient education [14], while photothermal imaging radar offers a non-invasive alternative for early decay detection [15]. Although these imaging technologies enhance diagnostic capabilities, they require specialized hardware, making them less accessible compared to AI-driven X-ray analysis, which leverages existing clinical imaging workflows.

The combination of AI-driven object detection and interactive explainability through chatbots presents an opportunity to improve not only diagnostic accuracy but also patient engagement and trust. Our work builds upon prior AI-based dental diagnostic methods while introducing an additional focus on accessibility and patient-centric education, addressing the limitations of existing approaches.

III. METHODOLOGY

A. Caries Detection Model

1) Dataset Description: The DENTEX MICCAI 2023 dataset was introduced as part of an international machine learning challenge at the prestigious MICCAI Conference, a leading event in medical image computing and AI-based healthcare solutions. The dataset comprises panoramic dental X-rays collected from three different institutions, ensuring a diverse representation of imaging conditions. We had access to 705 fully labeled X-rays for disease classification. These images include annotations for four primary dental pathologies:

- Impacted Teeth

- Periapical Lesions
- Caries (Cavities)
- Deep Caries (Advanced Cavities)

This dataset provides a strong foundation for training AI-based models for automated dental disease detection.

2) Data Preparation: Given the presence of four labeled dental diseases, we made a strategic decision to focus on cavities by combining the Caries and Deep Caries classes into a single label. This choice was driven by the goal of improving diagnostic clarity, maintaining dataset balance, and enhancing model generalization.

One of the primary motivations for merging these classes was to simplify the diagnostic process. Deep Caries is an advanced form of Caries, meaning both conditions exist on the same disease progression spectrum rather than being distinct diseases. In practical dental diagnosis, dentists often treat them similarly when performing early-stage detection. Since AI-based systems should prioritize early identification, distinguishing between Caries and Deep Caries in an AI model might not add significant diagnostic value at this stage.

Another key factor was maintaining dataset balance. Analyzing a co-occurrence heatmap of the dataset, we observed that Caries and Deep Caries frequently appear together in images (294 times). Additionally, if the dataset contains significantly more images labeled as Caries than Deep Caries, the model might struggle to properly differentiate between them, leading to class imbalance issues. By merging them into a single disease class, we improve label distribution balance, which contributes to more stable training and reduced bias in model predictions.

Finally, model generalization was an important consideration. Reducing the number of labels helps the model focus on the fundamental distinction between healthy and diseased teeth rather than fine-grained differentiation. A simpler label structure allows the model to avoid overfitting to subtle inter-class differences, which is particularly important given the dataset's size. By training the model to distinguish normal teeth from carious teeth broadly, we enhance its ability to perform well across diverse patient populations and real-world clinical settings.

B. Model Selection

The YOLOv5-Small architecture was selected for its optimal balance of detection accuracy and inference speed—critical requirements for a real-time, web-based dental diagnostic tool. YOLOv5's streamlined backbone and detection head enable rapid processing of panoramic X-rays (approx. 5–10 ms per image on a modern GPU) while maintaining high performance [16]. Compared to larger variants of YOLO, the reduced parameter count of the Small model (7 million vs. 45 + million) produces a smaller memory footprint and faster load times, making it ideal for clinical deployments where

low latency and minimal computational overhead are essential [17].

C. Data Augmentation

Targeted data augmentation was applied to improve model generalization across diverse patient anatomies and imaging conditions. Horizontal flips doubled the effective dataset size by leveraging the natural symmetry of panoramic radiographs. Additional brightness and contrast adjustments simulated variations in X-ray exposure and equipment settings, further reducing overfitting without introducing distortions that could confuse cavity detection [18]. These augmentations ensured robust performance under real-world imaging variability.

D. Training and Evaluation

The dataset of 705 annotated panoramic X-rays was divided into an 80/20 split for training and validation. YOLOv5-Small was trained for 75 epochs with a learning rate of 0.01 and a batch size of 16. Model performance was assessed on the validation set using precision, recall, and mean average precision at an IoU threshold of 0.5 (mAP@0.5) [19]. The final model achieved **92% precision, 96% recall, and 97% mAP@0.5**, demonstrating high accuracy in cavity detection with minimal false positives, ensuring its reliability for actionable dental diagnostics [20].

E. DeepSeek LLM

The chatbot utilizes DeepSeek LLM, a powerful large language model optimized for natural language understanding and generation. DeepSeek enables the chatbot to process complex dental inquiries, interpret medical terminology, and generate clear, contextually accurate responses [21]. Its deep learning capabilities ensure that users receive precise and well-structured answers, enhancing the overall user experience. The model is fine-tuned to prioritize dental-related conversations, making it well-suited for patient education and real-time assistance. A key advantage of DeepSeek LLM is its cost-effectiveness, making it significantly more affordable compared to other large-scale models. This affordability ensures that the chatbot remains economically viable, even when scaled to support a large number of users [22]. By reducing computational costs without compromising performance, DeepSeek allows for widespread deployment in clinical and telehealth environments, providing real-time assistance without excessive infrastructure expenses.

F. LangChain Integration

To streamline communication and enhance functionality, LangChain is integrated into the chatbot. LangChain acts as the framework that connects DeepSeek LLM with external data sources, enabling dynamic retrieval of dental knowledge and guidelines [23]. It facilitates contextual memory, allowing

Your X-Ray Review

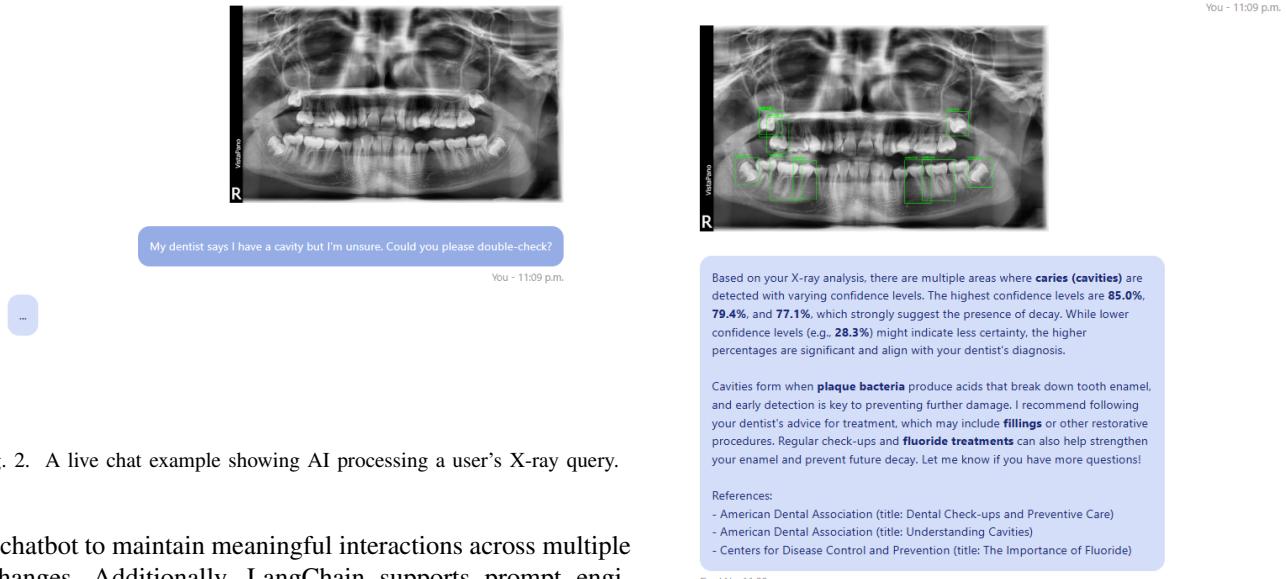


Fig. 2. A live chat example showing AI processing a user's X-ray query.

the chatbot to maintain meaningful interactions across multiple exchanges. Additionally, LangChain supports prompt engineering and retrieval-augmented generation (RAG), ensuring that responses are grounded in verified information from the chatbot's knowledge base.

G. Chatbot Knowledge Base

The chatbot's knowledge base is curated from authoritative dental and health sources, ensuring credibility and accuracy in responses. Primary sources include the *Centers for Disease Control and Prevention (CDC)*, *National Institutes of Health (NIH)*, *American Dental Association (ADA)*, and the *Royal College of Dental Surgeons of Ontario (RCDSO)* [24]. These organizations provide well-established guidelines on oral health, cavity detection, treatment options, and preventive care.

The knowledge base consists of articles covering key topics such as:

- Dental standards and best practices
- Interpretation of dental X-rays
- Cavity detection and treatment options
- Preventive oral health measures
- Terminology and definitions used in dentistry

By structuring responses based on these trusted sources, the chatbot ensures evidence-based, reliable information for both patients and dentists. This enables users to make informed decisions about their oral health while improving their understanding of dental diagnostics and treatments [25].

H. Application Development

1) *Front-end Design:* DentAI Vision's interface operates on a React-based web app. The interface connects the user, typically a patient or dentist, with the model and the chatbot. First,

Fig. 3. Example of chatbot response explaining detected anomalies on the X-ray. The chatbot provides insights using references from accredited sources to enhance user trust and transparency.

users upload panoramic dental X-ray images and a message prompt. The model then annotates the image with bounding boxes around the suspected cavities and sends the annotated image back to the user. Finally, the chatbot interprets the user's prompt, along with the model's analysis, and communicates the results to the user.

DentAI Vision's interface prioritizes transparency. Users can download their annotated X-ray image for future reference. Similarly, the user can send additional messages to the chatbot to better understand the model's results and to discuss the next steps for treatment. Most importantly, when the chatbot makes a claim, it ensures that its information is reliable and accurate by citing its sources directly within the message box. DentAI Vision uses transparency to build patients' trust in their dentists and their diagnoses.

2) *Back-end Development:* The backend of DentAI was developed using FastAPI, a lightweight and high-performance Python framework, to efficiently manage API requests and handle image processing for dental X-ray analysis. The system follows a modular design, where uploaded X-ray images are processed, analyzed using a trained YOLOv5 model, and annotated with detected cavities. The results are then passed through a chatbot for interpretation, creating a seamless pipeline from image input to user-friendly analysis.

Before passing the image to the model, the backend ensures



DentAI Vision

Improving patient trust, one smile at a time.

New Chat

Fig. 4. DentAI Vision homepage, where users can initiate a new chat session.

it is properly formatted. The uploaded file is converted from binary data to a NumPy array using OpenCV. This step ensures compatibility with the YOLOv5 model, rejecting corrupt or improperly formatted files. If the image is valid, it is sent for model inference. The model, hosted using TorchServe, performs real-time inference and returns an annotated image along with confidence scores for detected cavities. The OpenCV library is used to resize, encode, and annotate images before sending them back to the frontend.

The backend exposes multiple RESTful API endpoints, including a predict endpoint, which accepts an image file, runs YOLOv5 inference, and returns the processed image with cavity detections. The chat endpoint accepts user queries and previous detections, passing them to the chatbot for a response. Data is exchanged in Base64-encoded images and JSON format, ensuring efficient communication between the backend and ReactJS frontend via the JavaScript Fetch API.

For chatbot integration a DeepSeek-powered chatbot was integrated to explain model predictions in simple terms. The chatbot retrieves relevant dental knowledge and responds based on previous cavity detections, providing users with dental health advice, treatment suggestions, and explanations of detected anomalies, improving user engagement. The LangChain framework is used to manage conversation memory and provide contextual responses.

The backend is optimized using asynchronous request handling to improve response times and manage concurrent users efficiently. Comprehensive error handling ensures that invalid image formats, server failures, or incorrect API requests are handled gracefully, preventing system crashes.

3) Transparency in Model Prediction: To ensure maximum transparency, the model outputs a confidence score for each prediction, which is displayed alongside the detected bounding boxes. This confidence score provides users with a quantifiable measure of certainty in the AI-generated diagnosis, helping

them make informed decisions about their dental health.

Displaying confidence scores enhances transparency by allowing users to understand the model's level of certainty for each detection. Unlike a binary output (detection vs. no detection), confidence scores provide a more nuanced interpretation, enabling users to gauge whether a particular detection should be taken seriously or further verified by a professional.

Confidence scores are particularly useful for users as they can help differentiate between high-confidence predictions that are likely accurate and low-confidence predictions that may require additional verification. For example, a detection with a confidence score of 98% is more likely to be a true positive than one with a confidence score of 55%, where the model may be less certain about the presence of a dental anomaly.

Confidence scores vary due to multiple factors, including image quality, lighting conditions, presence of overlapping structures, and variations in dental anatomy. A lower confidence score may indicate uncertainty caused by these challenges, reinforcing the need for human verification in borderline cases.

To ensure users are aware of these limitations, a disclaimer is prominently displayed on the platform before they interact with the AI system, stating:

"This AI-generated diagnosis is for informational purposes only and should not be a substitute for professional dental consultation. Always seek the advice of a qualified dentist for accurate diagnosis and treatment."

Additionally, when starting a chat with the AI assistant, users are reminded that the chatbot provides general dental information but does not replace professional medical advice. This ensures ethical AI use by clearly communicating the system's role as an assistive tool rather than an authoritative diagnostic solution.

IV. RESULTS

The model's effectiveness was assessed using standard evaluation metrics, including precision, recall, and mean Average Precision (mAP@0.5). Initially, performance was analyzed on both the caries and impacted tooth labels. However, based on the justifications outlined in the Data Preparation section—specifically, the decision to merge caries and deep caries into a single "caries" label—the primary focus shifted to improving caries detection.

TABLE I
PERFORMANCE OF THE YOLOV5 MODEL FOR DETECTING DENTAL CARIOSIS AND IMPACTED TEETH.

Class	Precision	Recall	mAP@50
Caries	92%	96.3%	97.3%
Impacted	72.1%	76.9%	79%

The YOLOv5 model demonstrated high accuracy in detecting caries, achieving a precision of 92%, recall of 96.3%, and an mAP@0.5 of 97.3%. This performance is comparable to existing state-of-the-art dental X-ray classifiers. For instance, a recent study utilizing the YOLOv8 algorithm for interproximal caries detection reported an overall precision of 84.83%, recall of 79.77%, and an F1 score of 82.22% [26]. Another study comparing three deep learning architectures for proximal caries detection found that the YOLOv5 model achieved a mean average precision (mAP) of 64.7%, a mean F1-score of 54.8%, and a mean false negative rate of 14.9% [27]. These comparisons highlight the efficacy of our model in caries detection.

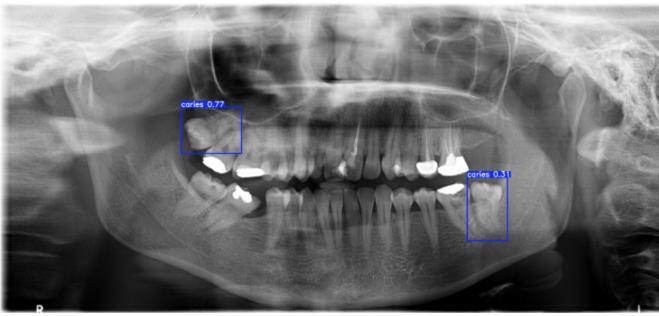


Fig. 5. Example of object detection on a panoramic dental X-ray. The YOLOv5 model detects caries with associated confidence scores.

While the primary focus of our study was caries detection, we initially evaluated the model's performance on both caries and impacted teeth. The detection of impacted teeth showed lower accuracy, with a precision of 72.1% and recall of 76.9%, indicating that additional refinement is needed for this class. Due to the limited number of impacted teeth samples in the dataset, the model struggled with generalization, leading to more inconsistent predictions.

For caries detection, one notable challenge was false positives, particularly in cases where dental fillings or artifacts resembled caries, leading to occasional misclassifications. This issue highlights the need for further fine-tuning and post-processing techniques to differentiate between actual caries and non-caries artifacts.

Additionally, experiments demonstrated that applying horizontal flipping as a data augmentation technique improved detection accuracy. Dental radiographs are inherently symmetrical, meaning flipping helps the model generalize better, particularly for images of teeth in different orientations.

TABLE II
IMPACT OF HORIZONTAL FLIPPING AUGMENTATION ON CARIES DETECTION PERFORMANCE.

Experiment	Precision	Recall	mAP@50
Without Augmentation	92%	96.3%	97.3%
With Augmentation	80.6%	94.7%	93%

The results in Table II show that applying horizontal flipping significantly improved model precision, recall, and mAP@0.5 for caries detection. Without augmentation, the model achieved a precision of 80.6%, but after augmentation, precision increased to 92%, demonstrating the effectiveness of this preprocessing technique. The use of augmentation also slightly improved recall and overall detection performance, reinforcing its role in enhancing model robustness across varied image orientations.

V. ETHICAL CONSIDERATIONS

As DentAI Vision operates in the medical domain, ethical considerations regarding user privacy, fairness, safety, and transparency are essential. This section outlines the measures taken to ensure compliance with regulatory standards, minimize bias, and maintain ethical AI practices.



Fig. 6. User consent agreement displayed before uploading an X-ray image. Users must acknowledge terms, including data privacy and the AI system's role as a supportive tool rather than a diagnostic system.

A. User Privacy and Data Security

Ensuring the privacy and security of patient data is a core principle of DentAI Vision. The platform follows established HIPAA (Health Insurance Portability and Accountability Act), PIPEDA (Personal Information Protection and Electronic Documents Act), and GDPR (General Data Protection Regulation) principles, emphasizing user consent, secure data transmission, and confidentiality. To protect user privacy:

- **No X-ray images or chat data are stored** after analysis; all uploaded files are processed in real-time and immediately discarded.
- **HTTPS encryption** is used for all data transmissions, ensuring secure communication between users and the platform.
- Users are **explicitly informed** that their data will not be retained, reinforcing trust in the system's privacy policies.

B. User Safety and Medical Liability

DentAI Vision is designed as an assistive tool rather than a diagnostic system, ensuring that users understand its role in supporting, not replacing, professional dental consultations. To uphold user safety:

- Each prediction is accompanied by a confidence score, allowing users to assess the certainty of AI-generated results.
- The model demonstrates high accuracy in detecting cavities, ensuring reliable second opinions for users.
- A clear disclaimer is displayed on the platform and chatbot interface, informing users that the AI system should not be used as a substitute for professional dental evaluation.

C. Bias, Fairness, and Model Generalization

A critical challenge in AI-driven diagnostics is algorithmic bias, which can result in discrepancies in model performance across different demographic groups. While the Dentex MICCAI dataset is diverse, certain populations may still be underrepresented. To address this:

- The development team acknowledges potential algorithmic bias and is committed to ongoing efforts to improve fairness and generalizability.
- Future iterations will integrate more extensive datasets and real-world validation to further refine model performance.

D. Collaboration with Dental Professionals and Trust-Building

Building trust with both patients and dental professionals is essential for the ethical deployment of AI in healthcare. To enhance credibility and correctness:

- The project aims to collaborate with licensed dental professionals to review AI-generated insights and ensure their accuracy.
- A chatbot powered by DeepSeek-Llama and LangChain provides users with evidence-based dental health information sourced from over 20 accredited references, ensuring reliable and transparent communication.

By implementing these ethical safeguards, DentAI Vision ensures that AI-driven dental diagnostics remain secure, fair, and beneficial while maintaining the highest standards of user safety and privacy.

VI. LIMITATIONS AND FUTURE WORK

While DentAI Vision demonstrates promising results in AI-assisted dental diagnostics, several limitations remain. One challenge is model sensitivity, as the AI sometimes over-predicts caries due to visual similarities with dental fillings, leading to false positives. Another key limitation is data diversity—while the Dentex MICCAI dataset is extensive, it may not fully represent all ethnicities and age groups, potentially affecting prediction accuracy across diverse populations. Additionally, the chatbot lacks real-time integration with live dental databases, meaning its responses may not always reflect

the most current dental research and treatment guidelines. These limitations highlight areas where further improvements are needed to refine the system's reliability and accuracy.

To enhance DentAI Vision, future efforts will focus on expanding the dataset with more diverse X-ray sources to improve generalizability. The YOLOv5 model will be fine-tuned with advanced post-processing filters to reduce false positives and enhance detection accuracy. Additionally, exploring newer architectures, such as YOLOv8 and Transformer-based models, may offer improvements in both detection efficiency and precision. Enhancing the chatbot's capabilities by integrating real-time dental knowledge bases will ensure more accurate and up-to-date responses. Further steps include deploying the web application for broader access, collecting feedback from dental professionals to refine model performance, and working toward clinical adoption by integrating the system into real-world dental workflows. A long-term objective is to enhance model interpretability and reliability, ensuring that AI-assisted diagnostics become a valuable tool for both dentists and patients in modern dental care.

VII. CONCLUSION

DentAI Vision presents an AI-driven platform designed to enhance trust between patients and dental professionals through automated analysis of panoramic X-rays and an interactive chatbot. By leveraging the YOLOv5-Small model for caries detection and integrating DeepSeek-Llama with LangChain, the system provides real-time X-ray interpretation alongside evidence-based dental health insights. The model demonstrated high accuracy in caries detection, achieving a precision of 92%, recall of 96.3%, and mAP@0.5 of 97.3%, showcasing its effectiveness as a decision-support tool.

Despite these promising results, challenges remain. False positives, particularly with dental fillings, and dataset limitations affecting impacted teeth detection highlight areas for improvement. Moving forward, expanding the dataset to include more diverse X-rays, enhancing the chatbot's real-time knowledge integration, and exploring newer deep learning architectures such as YOLOv8 and Transformer-based models will be key priorities. Additionally, clinical validation and collaboration with dental professionals will be essential to refine the system and ensure its reliability in real-world settings.

By addressing these challenges, DentAI Vision has the potential to become a trusted, accessible tool for both patients and dentists, ultimately improving transparency, patient education, and decision-making in oral healthcare.

REFERENCES

- [1] W. H. Organization, “Oral health fact sheet,” 2022. [Online]. Available: <https://www.who.int/news-room/fact-sheets/detail/oral-health>
- [2] U. of Southern California, “Dental caries: A global health burden and the cost of neglect,” 2023. [Online]. Available: <https://ostrowonline.usc.edu/dental-caries-a-global-health-burden-and-the-cost-of-neglect-2/>
- [3] E. e. a. Khoong, “Public trust in artificial intelligence for medical diagnoses: A national study,” *JAMA Network Open*, vol. 6, no. 12, p. e2330240, 2023. [Online]. Available: <https://jamanetwork.com/journals/jamanetworkopen/fullarticle/2830240>
- [4] R. B. Journal, “Nearly two-thirds of consumers surveyed say they’d trust a diagnosis from ai over a human doctor,” 2023. [Online]. Available: <https://radiologybusiness.com/topics/artificial-intelligence/nearly-two-thirds-consumers-surveyed-say-theyd-trust-diagnosis-ai-over-human-doctor>
- [5] A. e. a. Esteva, “Deep learning-enabled medical image analysis: A systematic review,” *Nature Digital Medicine*, vol. 4, pp. 38–55, 2021. [Online]. Available: <https://www.nature.com/articles/s41746-021-00438-z>
- [6] J. e. a. Lee, “Ai-assisted dental diagnosis using clinical data and imaging,” *Journal of Dental Research*, vol. 102, pp. 415–428, 2023.
- [7] H. e. a. Kim, “Artificial intelligence in panoramic radiographs: A multinational study,” *Medical Imaging and AI*, vol. 18, pp. 203–220, 2024.
- [8] R. e. a. Smith, “Application of yolov5 in developmental dysplasia detection,” *Scientific Reports*, vol. 13, p. 1256, 2023.
- [9] L. e. a. Zhang, “Optimizing yolov5 for kidney stone detection in ct images,” *Electronics*, vol. 13, no. 22, p. 4418, 2024.
- [10] X. e. a. Wang, “Yolov5-based real-time detection of lung tumors,” *Nature Scientific Reports*, vol. 14, p. 3543, 2024.
- [11] O. AI, “Fda-cleared ai enhancing dental x-ray visualization,” *Time Health*, 2023. [Online]. Available: <https://time.com/7094716/overjet/>
- [12] P. AI, “Ai-powered dental imaging with pearl,” *Dental AI Solutions*, 2023. [Online]. Available: <https://www.hellopearl.com/>
- [13] D. AI, “Automated ai diagnosis for dental conditions,” *Diagnocat Journal*, 2023. [Online]. Available: <https://diagnocat.com/us/>
- [14] A. e. a. Smith, “The role of intraoral cameras in modern dentistry,” *Dental Technology Review*, vol. 27, pp. 89–102, 2023.
- [15] A. e. a. Mandelis, “Photothermal imaging radar for early tooth decay detection,” *Optical Medical Engineering*, vol. 39, pp. 502–515, 2023.
- [16] Ultralytics, “YOLOv5 Official Documentation,” 2024. [Online]. Available: <https://github.com/ultralytics/yolov5>
- [17] ——, “YOLOv5 Model Comparison,” 2024. [Online]. Available: <https://docs.ultralytics.com/models/models/>
- [18] C. Shorten and T. M. Khoshgoftaar, “A survey on image data augmentation for deep learning,” *arXiv*, 2018. [Online]. Available: <https://arxiv.org/abs/1811.04768>
- [19] V. Labs, “Mean average precision (map) explained,” 2024. [Online]. Available: <https://www.v7labs.com/blog/mean-average-precision>
- [20] D. Ocean, “Evaluating object detection models,” 2024. [Online]. Available: <https://www.digitalocean.com/community/tutorials/mean-average-precision>
- [21] D. AI, “Deepseek llm overview,” 2024. [Online]. Available: <https://github.com/deepseek-ai/DeepSeek-LLM>
- [22] ——, “Deepseek llm pricing and scalability,” 2024. [Online]. Available: https://api-docs.deepseek.com/quick_start/pricing
- [23] LangChain, “Langchain official documentation,” 2024. [Online]. Available: <https://python.langchain.com/docs/>
- [24] C. for Disease Control and Prevention, “Oral health guidelines,” 2024. [Online]. Available: <https://www.cdc.gov/oralhealth/>
- [25] A. D. Association, “American dental association guidelines,” 2024. [Online]. Available: <https://www.ada.org/>
- [26] H.-J. Jin, B.-H. Lee, J. H. Park, H.-J. Lim, W.-Y. Park, and K.-H. Lee, “Advanced ai-driven detection of interproximal caries in bitewing radiographs using yolov8 algorithm,” *Scientific Reports*, vol. 14, p. 7892, 2024.
- [27] Y.-T. Lai, C.-Y. Chiu, C.-H. Chang, and T.-H. Chen, “Ai-identify: Deep learning for proximal caries detection on bitewing x-ray—hunt4 oral health study,” *Journal of Dental Research*, vol. 103, no. 2, pp. 245–258, 2024.

Do We Need Transformers to Play FPS Video Games?

Karmanbir Bath

University of Waterloo

ksbatth@uwaterloo.ca

Krish Sethi

University of Waterloo

kasethi@uwaterloo.ca

Aly Shariff

University of Waterloo

ashariff@uwaterloo.ca

Leo Shi

University of Waterloo

l7shi@uwaterloo.ca

Hetul Patel

University of Waterloo

hr8patel@uwaterloo.ca

Abstract—In this paper, we explore the Transformer based architectures for reinforcement learning in both online and offline settings within the Doom game environment. Our investigation focuses on two primary approaches: Deep Transformer Q-learning Networks (DTQN) for online learning [1] and Decision Transformers (DT) for offline reinforcement learning [2]. DTQN leverages the sequential modelling capabilities of Transformers to enhance Q-learning in partially observable environments, while Decision Transformers repurpose sequence modelling techniques to enable offline agents to learn from past trajectories without direct interaction with the environment. We conclude that while Transformers might have performed well in Atari games, more traditional methods perform better than Transformer based method in both the settings in the VizDoom environment [3]

I. INTRODUCTION

Q-networks traditionally have been used in various Atari game environments [4] and in partially observable environments like in the Doom 1993 game [5]. This was before the popularization of Transformers [6].

There is empirical evidence to show that Deep Transformer Q-networks outperform Deep Recurrent Q-networks in partially observable environments like memory cards and hallway [1]. The model used to play Doom 1993 was a mix of DQN [7] and DRQN [8] models that were used for navigation and strategy [5] respectively. We aim to see how Transformer can possibly increase the benchmarking results used in the paper [5] which is the kill-death (k/d) ratio in a team deathmatch environment. This would be done by replacing the LSTM [9] with the Transformer decoder architecture of the DTQN [1].

In recent years in Reinforcement Learning a new paradigm has emerged namely offline Reinforcement Learning [2]. We evaluated the performance of offline reinforcement learning in VizDoom’s most basic scenario by benchmarking episode rewards, comparing the Decision Transformer architecture against the PPO model [10].

A. Background

VizDoom is a first-person shooter (FPS) environment that serves as a popular benchmark for Reinforcement Learning (RL) research. It is built upon Doom, a classic 1993 video game, and provides a highly customizable RL environment where agents interact with the environment using raw pixel-based observations.

Online reinforcement learning is a paradigm in which an agent interacts directly and continuously with an environment

to learn optimal behaviors. The agent iteratively collects new experiences by actively taking actions and observing outcomes(states and rewards) from the environment to continuously update its policy.

Offline reinforcement learning is a paradigm that learns exclusively from static datasets of previously collected experiences [11]. Offline RL applications primarily revolve around in catastrophic areas like robotics where trial and error while exploring policies can lead to catastrophic failures.

B. Previous Work

The combined usage of Deep Recurrent Q-Networks (DRQN) and DQN have demonstrated strong performance in partially observable environments, achieving optimal results in the VizDoom Deathmatch environment as presented in Playing FPS Games with Deep Reinforcement Learning [5]. Similarly, Deep Transformer Q-Networks (DTQN) have shown their effectiveness in partially observable environments, where they achieved optimal results, highlighting the potential of transformer-based architectures for reinforcement learning in FPS video games like Doom [1]. Although, they have not been used in VizDoom environment so far.

In the offline reinforcement learning domain, Decision Transformer (DT), originally proposed in [2], successfully learned optimal policies in MuJoCo environment, demonstrating the viability of sequence modeling approaches in reinforcement learning.

More recently, the RATE (Recurrent Action Transformer with Memory) paper [12] demonstrated the successful training of a Decision Transformer (DT) in the ViZDoom TwoColours environment. In this scenario, the agent did not need to employ a strategy for shooting enemies but instead had to focus on strategic item collection and movement. This highlights that while transformer-based models can be effective in certain structured environments, their applicability in action-intensive tasks like shooting enemies remains an open challenge.

In both the settings, DT and DTQN have never been used in a VizDoom environments where the agent had to kill its enemies.

II. MODEL ARCHITECTURE

A. Deep Transformer Q-Networks

A **Deep Transformer Q-Network (DTQN)** is an extension of **Deep Q-Networks (DQN)** that integrates transformer de-



Fig. 1. A picture of the agent playing the basic scenario in Doom

coder based architecture modeled using Partially Observable Markov Decision Processes (POMDPs). Unlike conventional DQNs that use a fixed-size state representation, DTQN processes sequences of observations to learn a more context-aware representation of the environment. This allows DTQNs to maintain memory over long-horizon tasks, making them effective in environments with partial observability.

Formally, DTQN replaces the traditional fully connected layers in DQN with masked multi-head self-attention layers, enabling the network to capture temporal dependencies over sequences of observations.

Q -Learning's goal is to learn a function $Q : S \times A \rightarrow \mathbb{R}$, which represents the expected cumulative reward if starting in a state s and taking action a . Specifically, Deep Q-Networks (DQN) are trained to minimize the Mean Squared Bellman Error [4]

$$\mathcal{L}(\theta) = \mathbb{E}_{(s,a,r,s') \sim \mathcal{D}} \left[\left(r + \gamma \max_{a' \in \mathcal{A}} Q(s', a'; \theta') - Q(s, a; \theta) \right)^2 \right]$$

B. Descion Transformer

Decision Transformer (DT) is a sequence modeling approach for reinforcement learning that uses **transformers** to predict actions based on past states, actions, and rewards. It reformulates reinforcement learning (RL) as a conditional sequence modeling problem, where a trajectory is treated as a sequence of tokens. Given a desired return-to-go (RTG), the model generates actions that maximize future rewards. The key advantages of Decision Transformers include offline reinforcement learning capabilities, scalability, and the ability to handle long-range dependencies through self-attention mechanisms. It was introduced as an autoregressive model trained on supervised learning objectives rather than traditional RL-based value function optimization.

Mathematically, a Decision Transformer models a trajectory as:

$$\tau = (R_t, s_t, a_t, R_{t+1}, s_{t+1}, a_{t+1}, \dots)$$

where R_t is the return-to-go, s_t is the state, and a_t is the action at timestep t . The model predicts a_t based on previous tokens using a **causal transformer**.

Both Decision Transformers and Deep Transformer Q-Networks leverage the power of transformers in reinforcement learning, but DT focuses on offline RL with conditioned generation, while DTQN enhances Q-learning for partially observable environments.

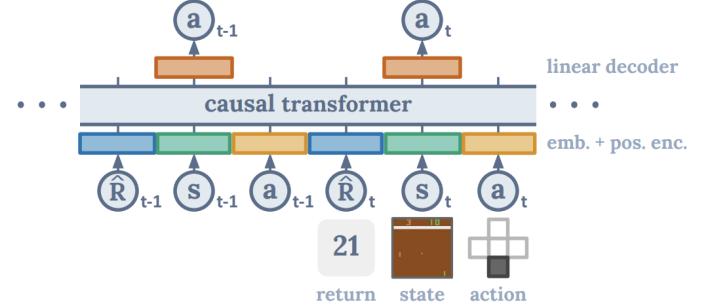


Fig. 2. is the DT architecture

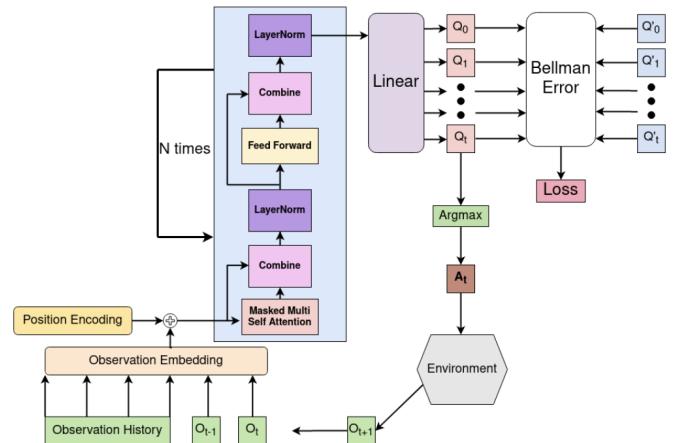


Fig. 3. is the DTQN architecture

III. METHODOLOGY

A. Deep Transformer Q-Networks

The training loop for the DTQN follows a reinforcement learning approach to optimize the Q-function, $Q(s, a; \theta)$, where θ represents the network parameters. The details of the algorithm are adapted from the original DTQN paper [1]. Here is an overview of the algorithm.

1. Minibatch Sampling: A minibatch of experience tuples $(h_{t:t+k}, a_{t:t+k}, r_{t:t+k}, h_{t+1:t+k+1})$ is sampled from the replay buffer \mathcal{D} , containing sequences of states, actions, rewards, and future states.

2. Target Q-value Calculation: The target target Q-values for each time step $t + i - 1$ is computed using the Bellman equation:

$$r_{t+i-1} + \max_{a' \in \mathcal{A}} Q(h_{t+1:t+i+1}, a'; \theta')$$

where r_{t+i-1} is the immediate reward at that timestep θ' represents the target network, which is updated less frequently than the main network.

3. Loss and Backpropagation: The loss for each minibatch is the mean squared error between the predicted and target Q-values. The network parameters θ are then updated using gradient descent.

Our model processes visual input using a deep sequence modeling architecture. A set of 50 frames is fed to a CNN network that reduces the dimensionality of the images and encodes spatiotemporal features, producing a sequence of embeddings once flattened, which are augmented with sinusoidal positional encoding. The frames are then passed to 5 transformer layers that perform multi-head attention (8 attention heads), alongside residual gating. Finally, the results are passed to a feed-forward network to produce Q-values corresponding to each possible action. During training, actions are selected according to an ϵ -greedy policy. The average training time for this setting was 20 hours, with 1.5 million iterations.

In addition to training a Q-network, our CNN layers are enhanced using a game features network during the training phase, which is used to predict features such as health, ammo, and number of enemies, for each given frame [5]. These features are only available during training time, which we leveraged to improve the embeddings for the transformer.

Our reward shaping follows the method explained in [5].

B. Decision Transformer

The training methodology for the Decision Transformer (DT) closely follows the approach outlined by the work done in recurrent action transformer with memory *Recurrent Action Transformer with Memory* [12]. We adopted their hyperparameters, except for the targeted return, which we set to 110 since it gave the most optimum result. Decision Transformers are trained using a dataset of past trajectories collected from an environment. Each trajectory consists of a sequence of states, actions, and rewards:

$$\tau = \{(s_1, a_1, r_1), (s_2, a_2, r_2), \dots, (s_T, a_T, r_T)\} \quad (1)$$

During training, the transformer model processes a context length of 90 past trajectories and learns to predict the next action a_t given the current state s_t and returns-to-go (RTG), which represents the expected future reward:

$$a_t = \pi_\theta(s_t, r_t, \text{RTG}_t) \quad (2)$$

where RTG is computed as:

$$\text{RTG}_t = \sum_{k=t}^T \gamma^{k-t} r_k \quad (3)$$

where γ is the discount factor.

We generated a dataset of 45,000 trajectories using a PPO model implemented with the Stable-Baselines3 python library [10]. The environment was based on ViZDoom, and data

collection began after 90,000 training steps of the PPO model. During training, the PPO model achieved an average episode return of approximately 80, with returns ranging from -400 to 100.

Training for the DT was conducted over 100 epochs. To preprocess observations, we employed convolutional neural networks (CNNs) with hyperparameters sourced from Deep Reinforcement Learning on a Budget: 3D Control and Reasoning Without a Supercomputer [13]. The Decision Transformer was then trained on the preprocessed observations extracted from the collected dataset.

The environment for training DT was the **basic scenario** in the ViZDoom library, using the default reward structure, action space, and other configuration settings provided by ViZDoom.

To improve training efficiency, we employed the **frame skipping technique** with a frame skip of 4 [5]. This means that the agent repeated the same action for the next four frames without interacting with the environment. Consequently, the agent received a new screen input only every $k + 1$ frames, where k is the number of frames skipped between steps (in our case it was 4). The average training time was 45 mins.

IV. RESULTS

To evaluate the effectiveness of transformer-based architectures in reinforcement learning, we conducted experiments in both offline and online RL settings.

For the **online RL** setting, we compared the Deep Transformer Q-Network (DTQN) against the combined usage of DQN and Deep Recurrent Q-Network (DRQN) which used the LSTM model. Our findings demonstrate that DQN-DRQN consistently achieved better performance in VizDoom's Team Deatchmatch setting, which involved different sets of maps in DOOM video game. DTQN's reliance on self-attention did not effectively compensate for missing state information, whereas DQN-DRQN's explicit recurrence allowed for better state tracking and decision-making under uncertainty.

In the **offline RL** setting, we compared the episode rewards for Decision Transformer (DT) and Proximal Policy Optimization (PPO) in VizDoom's basic scenario. Results indicate that PPO outperformed the Decision Transformer in terms of final policy performance.

These results suggest that while transformers offer strong sequence modeling capabilities, they are not inherently well-suited for environments requiring active memory-based strategies like VizDoom. In both the settings transformer-based architectures learnt a suboptimum policy, while more traditional methods learnt a better policy

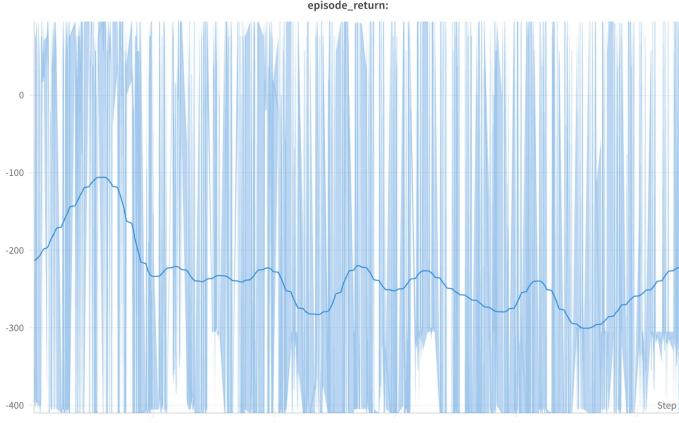


Fig 4: the episode returns of the DT architecture

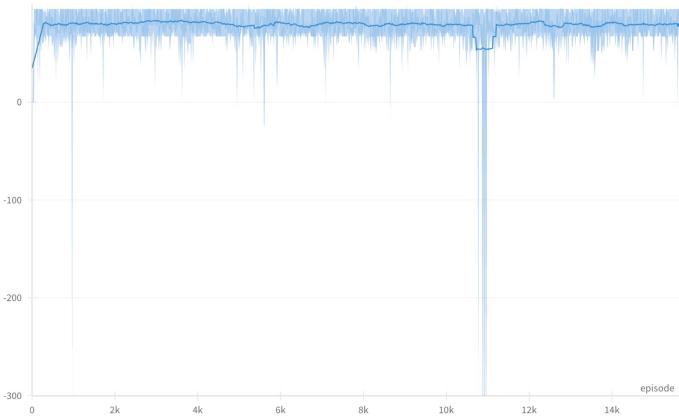


Fig 5: the episode returns of the PPO model.

Evaluation Metric	Limited Deathmatch		Full Deathmatch			
	Known Map		Train maps		Test maps	
	Without navigation	With navigation	Without navigation	With navigation	Without navigation	With navigation
Number of objects	14	46	52.9	92.2	62.3	94.7
Number of kills	167	138	43.0	66.8	32.0	43.0
Number of deaths	36	25	15.2	14.6	10.0	6.0
Number of suicides	15	10	1.7	3.1	0.3	1.3
Kill to Death Ratio	4.64	5.52	2.83	4.58	3.12	6.94

Fig. 6. the k/d ratio achieved by the DQN-DRQN model [5]

V. CONCLUSION

In this work, we investigated the application of transformer-based architectures in reinforcement learning for FPS video games using the VizDoom environment. We evaluated two primary models: Deep Transformer Q-Networks (DTQN) for online reinforcement learning and Decision Transformers (DT) for offline reinforcement learning. Our goal was to assess whether transformer-based architectures outperform traditional methods in highly memory and strategy intensive environments like Doom.

Our results indicate that while transformers provide strong sequence modeling capabilities, they struggle in highly partial observable settings requiring strategic decision-making. In the **online RL** setting, DTQN underperformed compared to

Game statistics summary				
	Map06	Map07	Map08	All
Kills:	33	142	44	219
Deaths:	58	69	40	167
Suicides:	0	0	0	0
Frags:	33	142	44	219
K/D:	0.569	2.058	1.100	1.311
Medikits:	0	4	0	4
Armors:	0	0	0	0
SuperShotgun:	0	0	0	0
Chaingun:	0	0	0	0
RocketLauncher:	0	0	0	0
PlasmaRifle:	0	0	0	0
BFG9000:	0	0	0	0
Bullets:	0	0	0	0
Shells:	3	0	0	3
Rockets:	0	0	0	0
Cells:	0	0	0	0
log_:	{ "kills": 219, "deaths": 167, "suicides": 0, "frag": 219.00000 }			
New best score:	219.00000			

Fig. 7. DTQN k/d ratio

the combined usage of Deep Q-Networks (DQN) and Deep Recurrent Q-Networks (DRQN) with LSTMs. The recurrence in DRQN enabled more effective state tracking and decision-making, while DTQN’s reliance on self-attention alone was insufficient to compensate for missing state information.

In the **offline RL** setting, we compared Decision Transformers (DT) against Proximal Policy Optimization (PPO) using a dataset of 45,000 trajectories. The empirical results show that PPO achieved superior policy performance in VizDoom’s basic scenario. The Decision Transformer was able to learn meaningful policies from offline data, but struggled to generalize optimally, highlighting the limitations of sequence modeling approaches in memory intensive environments requiring strategy.

Overall, our findings suggest that while transformer-based architectures have demonstrated success in certain reinforcement learning benchmarks, they are not inherently well-suited for FPS environments (specifically VizDoom) that are memory intensive environments requiring strategy to move and kill enemies. The reliance on self-attention alone may not be sufficient to capture long-term dependencies in these settings.

A potential direction for future research is to investigate architectures that do not rely on self-attention but still perform well on long-range sequence tasks. One promising approach is Decision Mamba [14], which replaces self-attention with selective state-space models. This could provide an efficient way to handle long-range strategy while maintaining scalability in reinforcement learning environments.

Overall, while Transformer-based models have demonstrated success in structured tasks such as Atari [4] and some partially observable environments like hallway [1], their application in first-person shooter (FPS) reinforcement learning remains a challenge, requiring further research in attention mechanisms and memory modeling.

REFERENCES

- [1] K. Esslinger, R. Platt, and C. Amato, “Deep transformer q-networks for partially observable reinforcement learning,” *arXiv preprint*, 2022.
- [2] L. Chen, K. Lu, and A. R. et al., “Decision transformer: Reinforcement learning via sequence modeling,” *arXiv preprint*, 2021.
- [3] M. Kempka, M. Wydmuch, and G. R. et al., “Vizdoom: A doom-based ai research platform for visual reinforcement learning,” *arXiv preprint*, 2016.
- [4] V. Mnih, K. Kavukcuoglu, and D. S. et al., “Playing atari with deep reinforcement learning,” *arXiv preprint*, 2013.
- [5] G. Lample and D. S. Chaplot, “Playing fps games with deep reinforcement learning,” *arXiv preprint*, 2016.
- [6] A. Vaswani, N. Shazeer, N. Parmar, and et al., “Attention is all you need,” *arXiv preprint*, 2017.
- [7] D. S. e. a. Volodymyr Mnih, Koray Kavukcuoglu, “Human-level control through deep reinforcement learning,” *Nature*, 2015.
- [8] M. Hausknecht and P. Stone, “Deep recurrent q-learning for partially observable mdps,” *arXiv preprint*, 2015.
- [9] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, no. 8, p. 1735–1780, 1997.
- [10] J. Schulman, F. Wolski, and P. D. et al., “Proximal policy optimization algorithms,” *arXiv preprint*, 2017.
- [11] R. F. Prudencio, M. R. Maximo, and E. L. Colombini, “A survey on offline reinforcement learning: Taxonomy, review, and open problems,” *arXiv preprint*, 2022.
- [12] E. Cherepanov, A. Staroverov, and D. Y. et al., “Recurrent action transformer with memory,” *arXiv preprint*, 2024.
- [13] E. Beeching, C. Wolf, and J. D. et al., “Deep reinforcement learning on a budget: 3d control and reasoning without a supercomputer,” *arXiv preprint*, 2019.
- [14] T. Ota, “Decision mamba: Reinforcement learning via sequence modeling with selective state spaces,” *arXiv preprint*, 2024.

educ-AI-tion

Sidonia Tameshit
Queen's University
21sst11@queensu.ca

Jasmine Qin
Queen's University
23xbf1@queensu.ca

Sarena Sandhu
Queen's University
22bww2@queensu.ca

Ava Christie
Queen's University
21ajec@queensu.ca

Basma Azeem
Queen's University
23dzb9@queensu.ca

Abstract—Artificial intelligence is becoming an increasingly common tool for students and teachers alike, raising important questions about ethics, academic integrity, and its hindrance to creativity. This project explores how AI can be integrated into education to enhance learning while maintaining academic integrity, instead of outright banning its use in classrooms. Independent surveys and field research provide insights into how AI is being used in academic settings, where the line is drawn between assistance and cheating, and whether AI is improving or undermining education. Using this data, along with research on AI ethics, a proposed solution called GPTeach was developed. GPTeach is an AI tool designed to support learning without compromising critical thinking. This solution emphasizes streamlining the teaching process, educating students about responsible use and personalizing the learning experience to fit the strengths and weaknesses of each student.

I. INTRODUCTION

30 years ago, students spent countless hours practicing cursive handwriting because it was considered a fundamental skill. However, by the time they entered the workforce, computers had largely replaced the need for handwritten communication. This shift reflects a broader challenge in education—keeping pace with technological advancements to ensure students are prepared for the evolving demands of the workplace. Today, artificial intelligence (AI) is reshaping industries at a rapid pace, yet many education systems remain hesitant to integrate it into learning. ChatGPT and similar generative AI chatbot programs have been banned in school districts across North America to “protect academic honesty,” “avoid cheating,” and preserve “critical thinking and problem-solving skills” [1] [2]. These concerns have overpowered the many possible benefits brought about by chatbots and generative AI programs to students and educators alike.

Generative AI programs pose potential not only to supplement under-resourced education systems, but also to support teachers struggling to address individual students’ needs. Further, the use of generative AI prepares students for a 21st-century workforce which will rely more and more heavily on AI [2] [3]. Instead of avoiding AI, schools should focus on teaching students how to use it effectively and ethically, just as they once adapted to computers. A report from the World Economic Forum predicts that AI will be embedded in 75% of workplaces within the next few years, highlighting

the growing importance of AI literacy [4]. By recognizing this shift and adapting educational approaches accordingly, schools can ensure that students are equipped with the skills necessary for the modern workforce.

II. BACKGROUND AND RELATED WORK

A. History of AI in Education

The rapid advancements in artificial intelligence over the past few decades have generated significant interest and excitement within the field of education [5]. The background of AI, particularly in education, traces its roots to the mid-20th century when early artificial intelligence systems were conceived. The term “artificial intelligence” was first coined in 1955 by John McCarthy, and over the next few decades, the potential applications of AI were explored across various fields, including education [5]. One of the earliest examples of AI in education was the PLATO system developed in the 1960s, which was a precursor to modern computer-based learning platforms [6]. PLATO was designed to provide computerized tutoring and educational games, setting the stage for the integration of technology in learning [6]. By the 1980s, AI began to gain traction in more specialized applications, such as intelligent tutoring systems (ITS) and expert systems [7]. These systems were programmed to deliver personalized learning experiences based on a student’s individual progress and performance. Socratic method-inspired systems, for instance, aimed to simulate human tutoring by asking questions and guiding students through problem-solving tasks [7].

Today, modern AI uses machine learning, natural language processing (NLP), and adaptive algorithms to enhance learning experiences. Machine learning allows AI to adapt to student behaviour over time, offering personalized learning paths. NLP enables AI systems to understand and respond to student queries (e.g. chatbots, essay feedback), and data analytics tracks student progress and predicts learning outcomes [8]. The tools modern generative AI employs introduces a promising learning tool for students around the world.

B. Current Significance of AI in Education

Recent research has highlighted the promising applications of AI in education, particularly concerning intelligent tutoring systems, automated assessment, and personalized learning [9].

AI-powered tutoring systems can provide customized feedback and adjust teaching strategies based on individual student requirements and progress. Some examples of AI platforms that were made solely for academic use include systems like DreamBox or Knewton, which adjust lesson difficulty based on student needs. Additionally, AI-powered tutors like Carnegie Learning can provide targeted help. Students can even use tools such as AI-generated quizzes, flashcards, and practice questions to study more efficiently. Furthermore, to increase student engagement, apps like Duolingo use AI to personalize and gamify learning.

Despite AI's positive role in academic settings, its increased use in education also comes with challenges such as ethical concerns regarding bias, data privacy, and access [10]. For example, if an AI model is trained on biased or incomplete datasets, then these ideas would be adopted among the students. Additionally, adaptive learning systems may not cater well to students from underrepresented groups because of the [11]. Bias could also be prevalent in grading systems which may unfairly disadvantage non-native speakers or students with unique writing styles. AI systems also require extensive student data to function, raising concerns about data collection, storage, and potential misuse. There is also a lack of transparency regarding data as users are often unaware of how their information is collected and used [12]. Furthermore, there is a risk of students becoming overly dependent on AI for learning, reducing critical thinking and problem-solving skills [13]. Additionally, teachers may overly rely on AI for administrative or teaching tasks, potentially diminishing the humane element of education. Not only can AI strip students of gaining vital skills learned from school but it is also known for relaying incorrect information to students due to the nature of its data retrieval from the internet.

The many facets of modern generative AI, including machine learning, natural language processing (NLP), and adaptive algorithms, lend themselves to empowering AI as an educational tool. Through awareness of AI challenges such as bias, privacy and access, AI has the potential to become a reliable and powerful tool in any student's arsenal. Though many have taken advantage of AI's ability to be used within learning, none have been created as an accessible, public tool for learning promotion which would work alongside schools. By understanding student and teacher use and opinion of generative AI, this project aims to create a generative AI tool with guidelines to ensure student growth and learning while easing the difficulty of teaching in a modern technological era.

III. CASE STUDIES

The integration of AI into modern education has already led to significant changes in the classroom and will continue to make great strides as it evolves. This paper examines the ethical implications of AI in education by analyzing perspectives from both students and educators. To best grasp these implications, surveys for both groups were created and sent out, and the results provided incredible insights into uses,

preconceived notions, and ethical dilemmas associated with AI in academic settings.

The surveys explored key concepts such as what teachers and students consider to be cheating, how AI tools are used in the context of schooling, and whether both groups envision a future where AI plays a central role in education and the workplace. To propose an AI-driven platform that meets the needs of both educators and learners, it was essential to gather firsthand data from those directly affected. Although the student and teacher sample sizes were relatively small, the results were very informative. For future work, a greater demographic of individuals should be surveyed so that the collected information is even more informative.

A. Context for Results

The student survey primarily examined the frequency of AI tool usage such as ChatGPT, the motivations behind its use, and students' personal ethical boundaries. A key focus was identifying where students drew the line between legitimate use and academic dishonesty. Exploration of their reasoning for AI use helped inform the development of an ethical alternative. There were 37 student responses, ranging from middle school (grades 6-8), high school, bachelor's degree, and college/professional degree.

The teacher survey examined similar themes from an instructional perspective. Topics explored included educators' definitions of cheating, their own AI usage, and their perceptions of how AI has affected student work. A critical objective was assessing whether teachers believe the quality of student submissions has changed since ChatGPT became widely accessible. Understanding these perspectives was crucial in suggesting a more effective and ethically responsible generative AI platform. 18 teachers were surveyed, including classrooms from middle school, high school, undergraduate and post-graduate level classes, and subject specialties spread across arts, science, business and professional courses.

B. Student Results

When students were asked about their AI use for school, 46.7% of students said they use ChatGPT one to five times per week for school, with 40% using it more than five times per week, as shown in Figure 1.

2. How often do you use ChatGPT or other AI tools on school topics per week?

37 responses

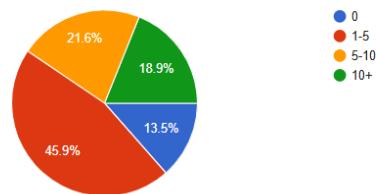


Fig. 1. Results from student survey conducted, demonstrating frequency of ChatGPT use amongst students

In contrast, students were asked how often they used campus/in-school help centres, and the most popular response was less than once per month, shown in Figure 2. This highlights the discrepancy between the help that students need and the help they seek through the school's support centres. AI is more readily accessible no matter the time or location, making it a favourable alternative.

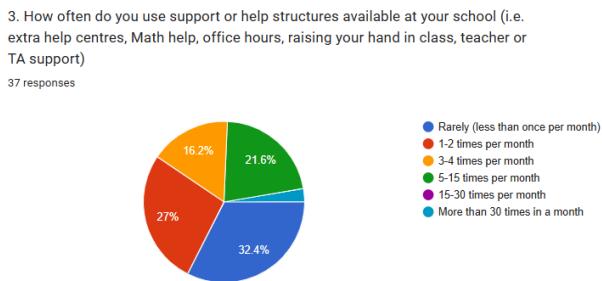


Fig. 2. Results from student survey conducted, demonstrating how often students make use of help centres provided by their schools.

When asked about the types of prompts students input to the program for school, almost half of students answered they input the homework problems verbatim, which is not in the best interest of their education experience. It was also found that 93.3% of students believe AI will be “somewhat important” to “extremely important” in the workplace in the next three to five years. This further demonstrates the eagerness for students to familiarize themselves with AI due to its rapidly increasing representation, and how important it is to introduce it to them before reaching the workforce.

When asked about their biggest concerns regarding generative AI's potential impact on their learning, three-quarters of students said they were worried that it would make them “lazy or reliant on technology”. More than half responded that “AI may not be able to fully capture the quality of human learning”, and one-third responded that “AI may give an unfair advantage to some students”. This demonstrates the hindrances that generative AI can have by removing the need for students to use their own critical thinking skills.

When asked about the impact AI has had on students' educational experiences, the results show both positive and negative aspects. One student highlighted the ease of access to personalized learning, and its ability to “dumb down” more confusing subject areas, while another noted, “Often I can get solid answers with little to no research...this leads to less of an intense learning experience and stuff doesn't stick as well.” Students have discovered ChatGPT as an incredibly useful resource, especially when outside help is inaccessible. But without limitations on its usage, it becomes a slippery slope towards over-reliance and the failure to truly grasp concepts.

1) Teacher Results: Two-thirds of the educators surveyed said they have used AI in a professional context. Most commonly, 44% responded that they use it to create assignments, and nearly 40% said they use it for creating lesson plans and assessments. To a lesser extent, some educators cited AI use

for grading, emails, and project suggestions. The results show that teachers are already using AI as a support tool, thus the introduction of one that caters specifically to their classroom would be beneficial. The absence of AI use for feedback, grading, and personal correspondence exhibits that AI does not hinder individuality in the classroom, but instead, supplements the teacher's ability to instruct. Furthermore, it is notable that a majority of teachers employ AI as a generative tool, using it to create questions, assignments, or lesson plans, instead of as a supportive tool, within the parameters of amplifying their own thoughts.

One of the primary goals of the survey was to gauge teacher's opinions on student usage of AI tools. When asked about their experiences with student's use of generative AI platforms, the vast majority of teachers surveyed did not feel students were using it in an ethical way. A high school business teacher said *“Students do not know how to use ChatGPT-generated content as a resource. They use it to replace their own original, critical thinking and, as such, the quality of student work has suffered.”*

When asked about their biggest concerns regarding programs like ChatGPT, nearly 90% of educators responded with a concern about student plagiarism and originality (see Figure 3 below). Later in the survey, teachers were asked to define plagiarism when put into the context of AI usage. The results showed that most teachers believed entering student-created prompts into AI tools such as ChatGPT is acceptable, however copying and pasting the asked question or generated responses is what qualifies as cheating. It was found that two-thirds of teachers were concerned with the long-term impacts AI has on their student's learning. Additionally, 39% of teachers were concerned about the accuracy of AI-generated results, as well as the lack of training on AI research tools, as shown in Figure 3.

5. What are your biggest concerns about the use of AI in school?

18 responses

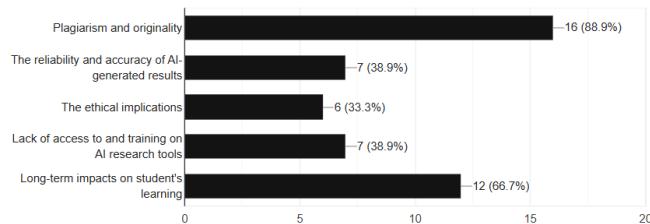


Fig. 3. Results from teacher survey showing the primary concerns for student use of generative AI tools

Furthermore, two-thirds of educators responded “most likely” or “absolutely” when asked if professional development or training on AI would be beneficial, as shown below in Figure 4. By introducing education about how AI tools function, there will be increased confidence when using AI tools within the classroom.

8. Would professional development or training help you feel more confident in using ChatGPT in your teaching?

18 responses

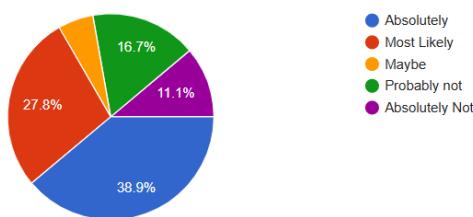


Fig. 4. Results from teacher survey showing the amount of teachers who feel that AI training would make them feel more comfortable introducing it to the classroom.

Educators were also asked what they thought was the biggest impact AI has had on education throughout their careers, whether it be positive or negative. A university instructor in Media Relations, Organizational Communications, and Research said “*So far negative. Students are simply using it as a shortcut to get assignments done more quickly. It takes quite a bit of instructor time to develop assignments that invite the critical thinking required to help students use it effectively.*” A professor of Mathematics wrote “*The only positive I have personally seen is help in coding for students who are not deep coders. Negatives are far more outweighing - reinforces the idea that answers already exist to questions being asked; strips the human kernel of creative thinking.*” Finally, an engineering professor wrote “*Outsourcing of one’s critical reflection to a massive database is detrimental. On the other hand, if what one teaches is always obtainable by ChatGPT with reasonable accuracy, perhaps what one teaches is to be revised. AI, if applied reasonably, will do a fine enough job in generic and introductory training but not much beyond.*”

The survey went on to collect teacher input into the incorporation of AI into schools and universities by asking how they would like to see current AI chatbots used by students. A common answer was that students must start citing AI use, even if it was only used to generate ideas, similar to how websites are cited within research projects. Another primary input point was solving the lack of fact-checking behind what ChatGPT outputs as a response, and eliminating the concern of AI providing students with misinformation. One professor went on to state that “*...one can create AI engines and neural networks specifically trained for a certain application, in this case university.*” A tailored approach to the proposed AI tool in education could be used to minimize these concerns, by applying citation generation, citing the AI at the end of prompt answers and allowing teacher input into the AI guidelines.

2) *Comparison of Results:* To gain insight into what students and teachers feel is ethical regarding generative AI, a thought experiment was conducted in the survey. Both students and teachers were given the same assignment question and asked to select all the prompts they considered to be

cheating if a student had asked ChatGPT the corresponding prompt. The question presented was, “*How did international alliances affect the onset of World War 1? Provide your opinion supported by historical examples.*” Options to select in the survey are displayed below in Figure 5.

- Copy pasting the whole prompt
- What alliances existed before the start of WW1?
- What caused the start of WW1?
- Summarize what happened before, during, and after WW1
- I think international alliances accelerated the start of WW1. Please give me examples to support this idea.
- Based on the historical example of the Austro-Hungarian empire annexing Bosnia and Herzegovina, how did this impact the onset of WW1?
- None of these examples would be considered cheating

Fig. 5. Response options presented to both students and teachers to select all which they consider to be cheating if asked the prompt detailed above.

The discrepancy between what educators considered cheating versus the students was significant. As shown in Figures 6 and 7 below, the prompt most agreed upon to be cheating was copy-pasting the entire question verbatim. However, it is interesting to note that where 84% of students thought it would be considered cheating, only 61% of teachers felt the same way. Although students strongly felt this prompt would be cheating, it has yet to be determined whether or not feeling this way would dissuade them from doing it. A fairly small percentage of students and teachers alike (less than 14% that were interviewed) felt the second and third prompts would be cheating. These were mainly research-based prompts, as opposed to opinion-forming ones. Where 32% of students felt the second last sentence would be considered cheating, a significant increase of 44% of educators felt this way, showcasing a variance where teachers and students “draw the line”. This prompt focused more on asking generative AI to connect the dots given a pre-decided historical example.

Teachers were asked to provide justifications for which prompts they deemed unethical. The common consensus was that when generative AI tools are used for research and compiling information, teachers are much less inclined to consider it cheating. When it is used to form opinions or creative thoughts, educators are much more likely to find the use of ChatGPT unethical. One teacher said, “*I believe AI results should not be tailored towards a specific conclusion. I believe the best use of AI [is] to have a wide range of information that can THEN be used by the user to create an idea or thought.*”

Another common opinion was teachers were less concerned about what students input to ChatGPT, but instead what they do with the resulting output. One teacher said, “*Students are free to enter prompts. It's how they use the results that determine if it is cheating. Did they copy and paste the AI results and present it as their own work without proper*

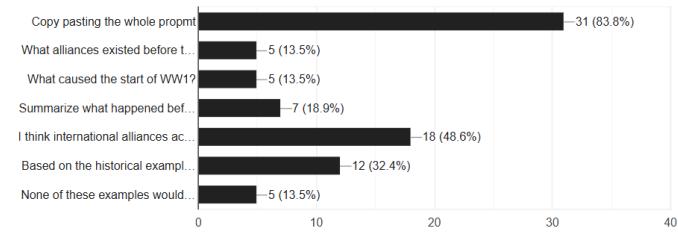


Fig. 6. Student survey results for which pasted prompts into a generative AI program would be considered cheating or unethical.

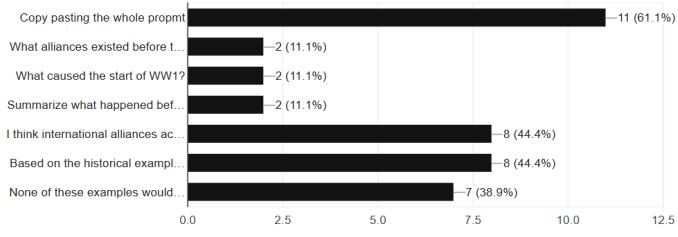


Fig. 7. Teacher survey results for which pasted prompts into a generative AI program would be considered cheating or unethical.

citations? If so, that is cheating. As instructors, we have to design assignments that help students learn to use AI ethically.”

A common opinion shared by the teachers interviewed was they did not trust their students to simply use ChatGPT to support their own preconceived ideas. Instead, they fear students use generative AI tools to do the critical thinking for them. A teacher said, “*Most students would just copy the response from any of those prompts without understanding*”. Thus, it is evident that any proposed generative AI model that would be education-friendly must not feed students the answer or suggest new ideas, but rather help students to come to their own conclusions and build on what they have learned.

IV. ETHICAL ANALYSIS

Future advancements in AI education tools must prioritize giving the students the fishing rod and not the fish. An ethical tool must guide students as a teacher would, instead of outright giving students the answer. Many educators highlighted their fear of AI tools in the classroom giving students the answer without any context or true learning being provided. For AI use in school to be considered ethical, it must encourage critical thinking by offering resources instead of answers, by prompting students with guiding questions, and by helping

them refine their reasoning so that creativity is encouraged. In STEM subjects, AI should identify specific mistakes in a student’s work in place of providing solutions, map out steps to approach the problem without outright providing the answer, and encourage students to engage with the material more deeply [5].

Platforms like Khanmigo have already demonstrated the effectiveness of hint-based learning rather than direct answers, ensuring students actively participate in problem-solving [11]. Additionally, AI should be integrated with safeguards, such as teacher oversight, the ability to pre-approve resources, and built-in citations to combat misinformation and bias [12]. Transparent warnings about privacy risks and AI limitations such as bias would further ensure ethical and responsible use.

A. Supporting Educators

To support teachers, AI can be customized to track student interactions, helping educators identify learning gaps and struggling students. By analyzing student queries and response patterns, AI could provide insights into classroom misconceptions, allowing teachers to tailor their instruction more effectively [13]. AI-driven tools should also adapt to different learning styles, offering varied explanations and information-delivery techniques to accommodate diverse student needs.

Moreover, AI can enhance teacher efficiency by consolidating course materials, generating interactive learning resources to fill gaps in student comprehension, and automating administrative tasks. With controlled access to course-specific materials, such as lab manuals or assignment guidelines, AI could serve as a valuable supplement for both students and substitute teachers [8]. Ultimately, AI should be designed to enhance, not replace, human instruction, making sure that students remain engaged and develop independent problem-solving skills while receiving the support they need. Generative AI has shown the capacity to create convincing false, but GPTeach would ensure credible and trustworthy answers, supervised by teachers [14].

B. Teacher Analysis

Programs modelled after GPTeach ease the burden on teachers and administrators by shifting the focus from constant regulation to meaningful technological integration in the classroom. The system’s feedback functions provide teachers with real-time evaluations of student understanding, allowing for more targeted instruction. Unlike unrestricted AI tools, GPTeach prevents students from simply copying AI-generated responses while still offering structured support to enhance learning. By making each student’s learning process visible, the platform enables educators to assess individual progress and differentiate instruction accordingly.

While generative AI has the potential to support teachers, GPTeach extends its impact by offering alternative learning methods tailored to specific educational needs. A major concern with general AI tools is their tendency to produce uniform responses, which can reduce the personalization of learning. GPTeach, however, is designed to align with individual courses and curricula, preserving the connection between teachers

and students while maintaining the integrity of personalized education.

Administrators, as key decision-makers in education, must navigate the challenges of AI integration thoughtfully. While generative AI expands the range of learning tools available, it also requires oversight to ensure students engage with content critically rather than passively absorbing AI-generated information. Schools and administrators will need to establish clear policies on AI usage, balancing accessibility with safeguards that promote genuine learning and critical thinking.

C. Student Analysis

The introduction of a specialized, education-focused generative AI engine provides students with a new learning tool which is accessible and redirects students from misusing AI. The tool provides easy integration into existing student routines surrounding the use of technology. In the survey conducted for this project, one of the teachers was quoted “[the] outsourcing of one’s critical reflection to a massive database is detrimental.” By adding frameworks to the generative AI design, GPTeach can become a tool for learning instead of a crutch or regurgitator of information. The shift in mindset fosters a healthy relationship between imaginative and unique thought and use of a helpful tool, promoting a growth mindset for any students using the new tool.

However, the introduction of a new AI tool also brings a concern of reliance on AI for students. Artificial intelligence runs on a catalogue of information it is given. The overuse of current AI tools can lead to a lack of student creativity and the inability to create unique ideas without prompting. The goal of a program like GPTeach is to create a tool that promotes student-created thought, emphasizing student individuality in the presence of AI. Arguably, students could use a generative AI tool that is not monitored by their teacher and that just outputs the answer. Although nothing is banning them from doing so, students would ultimately be doing themselves a disservice, as there would be no access to these tools come examinations. Thus, this proposed tool is a happy medium between bridging the gap between the help students need, and the ability to be an independent learner.

D. Risk Assessment

Equity remains a major concern when integrating AI into education, as not all schools or students have the financial resources to access advanced technology. Ensuring that AI tools are both affordable and widely available is essential to preventing further disparities in learning opportunities. Unlike other generative AI programs, GPTeach would follow a different business model designed with student privacy and fairness in mind. Instead of relying on cookies, third-party data collection, or prioritizing partnered services (such as Bing over Google), GPTeach would be sold directly to school boards. This approach provides an additional layer of data protection for students while ensuring the information delivered is sourced from the most accurate and objective materials available, rather than being influenced by corporate

partnerships. By prioritizing transparency and ethical AI use, GPTeach aims to support education in a way that fosters equal access, unbiased learning, and student privacy.

Privacy is one of the biggest concerns associated with the use of AI [2]. This issue is further complicated for tools like GPTeach being used by children, who cannot give consent. All AI systems require vast amounts of data in order to train their decision-making algorithms and improve their capabilities. As students use the tool, their learning patterns, age-appropriate use of language, and preferences continue to “teach” the AI model. As with internet use, children are prone to input sensitive information because they lack the cognitive abilities to understand the long-term consequences that might arise from their inputs. They have a harder time understanding how these AI systems collect, process, and store their information, which leaves them vulnerable to data exploitation.

One of the most pressing challenges regarding the storage of sensitive information is the potential for this data to be shared, sold to third parties, or exposed as a part of a data breach [15]. Unlike adults, children cannot fully consent to how their data is to be used because they do not understand the full extent of their actions. The data collected during their childhood could later be used for identity theft, targeted advertising, or more nefarious reasons, following them throughout their lives. The lack of government policy in this area means that children are vulnerable to the potential dangers of data misuse [16]. The creation of AI that can be implemented into schools must take into account this concern. AI tools in education must prioritize privacy, such as implementing safeguards against storing sensitive information or containing data retention periods. Child-friendly disclaimers and active monitoring from educators and administration is crucial in controlling the potential privacy risks of generative AI tools.

Another key factor when considering AI in education is the potential for bias in AI-generated responses. As established, AI models are trained on large sets of data and human feedback and are inherently subject to human biases or prejudices. An under-representation of certain groups or an overreliance on flawed sources could be reflected in responses. The result can be biased or altogether untrue responses. Unfortunately, this goes beyond incorrect math answers or made-up historical events. Google Gemini, one of the biggest AI tools currently used by millions of students, responded to a student’s homework inquiry with a deeply troubling response telling the user to “die” amongst other threats [17]. Ethical concerns surrounding AI continue to arise with the introduction of new tools, and these challenges are likely to escalate over time. This is especially worrying because of the context in which this AI is being implemented.

Children are particularly vulnerable to the influence of biased AI responses because many are still developing their cognitive abilities and critical thinking skills. They might lack the skill to fact-check or question the information that is being provided by the AI, especially in a learning environment. If they receive an AI response that has biases or inaccurate information, they are more likely to adopt these viewpoints

and shape their understanding. Even on a smaller scale, students could be subtly pushed towards specific perspectives or conclusions, which hinders their skill to think independently and critically. To address this concern, when implementing AI in schools, the AI model should be trained with diverse datasets, regularly audited responses, and clear disclaimers for children about its potential for bias.

Beyond those two factors of concern, there are also other risks associated with AI in children's education. Being exposed to AI at a young age could lead to an overreliance on AI tools and a reduction in human interaction, both of which are concerning for children's social and emotional development [18]. AI systems must be used sparingly, and investing in teaching meaningful connections and critical thinking is important to maintain a balance. Tools like GPTeach address some of these concerns by guiding student inquiry rather than regurgitating an answer.

V. THE SOLUTION: GPTEACH

GPTeach was created to fit the requirements aforementioned to make the learning and teaching experience more effective, more efficient, and more ethical. Figure 8 demonstrates the student's homepage, which outlines a clear disclaimer to teach students how the program works. This outline is important as it sets clear guidelines for how students should engage with GPTeach, but also AI as a whole. Early AI interactions set the standard for how students will use these technologies later in life. Therefore, by providing constant disclaimers, students are more likely to develop a critical understanding of AI's role in their education and foster a responsible mindset when approaching AI both within and beyond the classroom.



Fig. 8. Example homepage of proposed GPTeach tool, created using FIGMA.

This proposed AI framework would not give students explicit answers, but instead hint and push them towards the right solution. By referencing the provided classroom material uploaded in advance by the instructor, students have tools at their disposal that are reliable, accessible, and relevant to their particular curriculum. For students, GPTeach can act as a personalized tutor that aligns with the specific coursework

and available classroom resources given by their teacher as well as catering to their learning style. It provides guidance on their assignments, essays, and worksheets in a way that gives them space to solve the problems themselves. Figure 9 demonstrates an example conversation between a student and GPTeach.

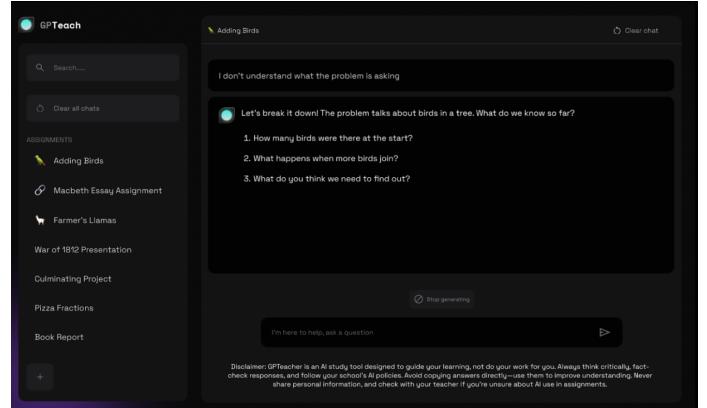


Fig. 9. Example conversation between the GPTeach program and a student using it for an addition exercise provided by their teacher, created using FIGMA.

The program summarizes what the student has learned so they can treat it like course material. It highlights the largest areas of need for the student and supplements with specific problems of similar form. It also has guidelines in place to avoid explicitly answering assignment questions and instead uses an open-ended, conversational approach to encourage students to generate their own opinions. By promoting a growth mindset and independent thought with AI answers to prompts, students will be able to analyze and explore their opinions on lesson topics without compromising their learning.

GPTeach is a tool for educators as much as for students. For an assignment uploaded by the teacher, the program analyzes conversations with all students and presents its findings, including which question students struggled with the most, suggested resources to remedy this, and a follow-up assignment to ensure student understanding. Figure 10 below shows an example of teacher support provided by GPTeach. Additionally, conversations with the student are accessible by the teacher at all times, ensuring transparency.

Furthermore, as the student uses GPTeach, the program creates a profile accessible by the teacher, which summarizes the student's learning style, largest areas of struggle, and suggested tools to use when assisting the student on a topic. In an age where class sizes are only increasing, this tool allows for teachers to keep on top of their students' needs, while also managing the class as a whole. This is demonstrated below in Figure 11. This tool would be incredibly beneficial for educators as it takes the difficulty out of analyzing and remembering each student's particular needs, and instead allows them to focus more on personalizing the class content once this information has already been determined.

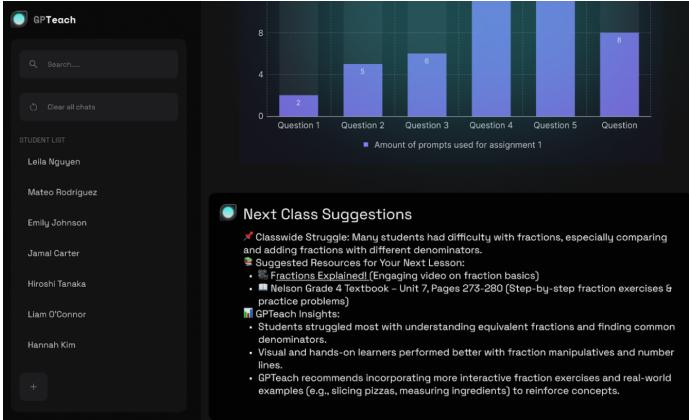


Fig. 10. Example conversation between the GPTeach program and a teacher to summarize assignment questions that students in their class found the most difficult, followed by suggested resources to share with the class to address the issues. Created using FIGMA.

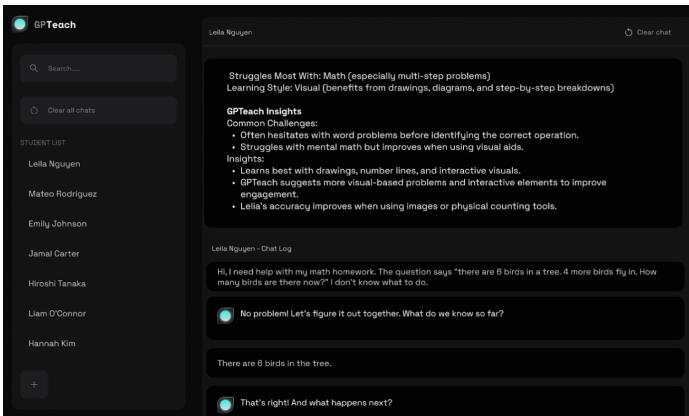


Fig. 11. Example of a student's profile visible to the teacher to provide insight on the student's learning patterns and conversation history to ensure transparency. Created using FIGMA.

VI. DISCUSSION

The results show a clear pattern. The current status quo surrounding AI in education is ineffective for both students and educators as a tool for education. The vast majority of students clearly recognize the problems with using AI tools, but are reliant on them nonetheless. While GPT models can offer accessible help in a student's learning, they introduce problems of over-reliance, academic integrity, and the erosion of critical thinking skills.

One of the most striking points is how often students use AI software compared to traditional academic support services because of their convenience and accessibility. This is evidence of how much potential there is in education for AI. There is a clear change in how students are approaching learning now, and it is important to have technology and tools that change at pace. Today's AI tools hinder students' learning in the long run rather than supporting it. This is evident in the response

from educators, a majority of whom have seen a decline in the quality of student work since the introduction of AI tools.

Students' increased AI reliance raises broader questions about equity. Students who have strong foundational skills are likely to greatly benefit from AI-assisted learning, but those who struggle may fare worse because it fails to teach them the basics. This can potentially exacerbate achievement gaps and hurt students who would benefit most from additional help.

A key takeaway from the results is the need for intersectional perspectives in creating AI education tools. Policymakers, educators, and developers must collaborate to ensure that the software would be able to meet the unique demands that education tools require. Implementing effective AI tools in the education system will protect learning and support students of varying ages, abilities and learning styles.

VII. CONCLUSION

As artificial intelligence continues to transform industries, it is essential for education systems to keep up and ensure students are prepared for the future. Simply avoiding AI is not a solution—it will only delay its inevitable presence. The research and survey results from this study show both the potential benefits and concerns surrounding AI in education, highlighting the importance of creating ethical guidelines that preserve academic integrity while encouraging flexibility, adaptive teaching styles and innovation.

To address these concerns, the GPTeach model was developed as an AI tool designed specifically for educational purposes. Unlike general AI tools, GPTeach focuses on responsible use, offering structured guidance, transparency, and safeguards to prevent misuse. Summaries of student difficulties and learning styles for educators, and personalized resource suggestions for students create an ethical yet effective tool to streamline learning in and outside of the classroom. By incorporating AI tools like GPTeach, schools can provide students with the necessary skills to thrive in an AI-driven world, all while maintaining the values upon which education was built. When used responsibly, AI can become a valuable resource that enhances learning while supporting, rather than replacing, critical thinking.

VIII. ACKNOWLEDGEMENTS

The team would like to acknowledge all the help provided to them, including Paul Wu for editing, QMIND for the provided resources, and all those who took part in the survey.

REFERENCES

- [1] A. Johnson, "Chatgpt in schools: Here's where it's banned and how it could potentially help students," Forbes, Jan. 2023, accessed: 2025-03-20. [Online]. Available: <https://shorturl.at/GoPUUp>
- [2] L. Labadze, M. Grigolia, and L. Machaidze, "Chatgpt in education: Opportunities and challenges," *Educational Technology Journal*, vol. 10, no. 1, pp. 1–17, 2023. [Online]. Available: <https://educationaltechnologyjournal.springeropen.com/articles/10.1186/s41239-023-00426-1>
- [3] N. E. Association, "Class sizes a growing issue among educators," NEA Today, 2023. [Online]. Available: <https://www.nea.org/nea-today/all-news-articles/class-sizes-growing-issue-among-educators>

- [4] W. E. Forum, “The future of jobs report 2023,” 2023. [Online]. Available: <https://www.weforum.org/reports/the-future-of-jobs-report-2023>
- [5] A. Bozkurt, A. Karadeniz, D. Baneres, A. E. Guerrero-Roldán, and M. E. Rodríguez, “Artificial intelligence and reflections from educational landscape: A review of ai studies in half a century,” *Sustainability*, vol. 13, no. 2, p. 800, 2023. [Online]. Available: <https://doi.org/10.3390/su13020800>
- [6] J. F. Allen and M. L. Witbrock, “Intelligent tutoring systems and their application to complex problem solving,” *ACM Computing Surveys*, vol. 28, no. 3, pp. 322–350, 1996. [Online]. Available: <https://doi.org/10.1145/503506.503539>
- [7] J. R. Anderson, C. F. Boyle, and B. J. Reiser, “Intelligent tutoring systems,” *Science*, vol. 228, no. 4698, pp. 456–458, 1986. [Online]. Available: <https://doi.org/10.1126/science.228.4698.456>
- [8] X. Zhao and Y. Yang, “Artificial intelligence in education: A systematic review,” *IOP Conference Series: Earth and Environmental Science*, vol. 693, p. 012019, 2021.
- [9] O. Onesi-Ozigagun, Y. J. Ololade, N. L. Eyo-Udo, and D. O. Ogundipe, “Revolutionizing education through ai: A comprehensive review of enhancing learning experiences,” *International Journal of Advanced Research in Social Sciences*, vol. 6, no. 4, p. 1011, 2023. [Online]. Available: <https://fepbl.com/index.php/ijarss/article/view/1011>
- [10] R. Lampou, “The integration of artificial intelligence in education: opportunities and challenges,” *Revista de Inteligencia Artificial en Educación*, vol. 4, no. 00, p. e15, Aug 2023. [Online]. Available: <https://educationai-review.org/revista/article/view/15>
- [11] A. M. Perry and N. T. Lee, “Ai is coming to schools, and if we’re not careful, so will its biases,” *Brookings*, 2023. [Online]. Available: <https://www.brookings.edu/articles/ai-is-coming-to-schools-and-if-were-not-careful-so-will-its-biases/>
- [12] R. F. Mello, E. Freitas, F. D. Pereira, L. Cabral, P. Tedesco, and G. Ramalho, “Education in the age of generative ai: Context and recent developments,” *arXiv preprint arXiv:2309.12332*, 2023. [Online]. Available: <https://arxiv.org/abs/2309.12332>
- [13] R. H. Mogavi, C. Deng, J. J. Kim, P. Zhou, Y. D. Kwon, A. H. S. Metwally, A. Tlili, S. Bassanelli, A. Buccharone, S. Gujar, L. E. Nacke, and P. Hui, “Chatgpt in education: A blessing or a curse? a qualitative study exploring early adopters’ utilization and perceptions,” *Computers in Human Behavior: Artificial Humans*, vol. 2, no. 1, p. 100027, 2024. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2949882123000270>
- [14] S. Liao. (2023) Chatgpt is lying. here’s why that matters. Accessed: 2025-03-20. [Online]. Available: <https://www.washingtonpost.com/technology/2023/04/05/chatgpt-lies/>
- [15] A. Gomstyn and A. Jonker, “Ai and privacy,” <https://www.ibm.com/think/insights/ai-privacy>, 2023.
- [16] J. Irwin, A. Dharmani, and N. Zon, “Children’s privacy in the age of artificial intelligence,” <https://www.csagroup.org/article/research/childrens-privacy-in-the-age-of-artificial-intelligence/>, Canadian Standards Association, Toronto, ON, Tech. Rep., 2021.
- [17] M. Taheri, “Google’s ai chatbot tells student seeking help with homework: ‘please die’,” <https://www.newsweek.com/googles-ai-chatbot-tells-student-seeking-help-homework-please-die-1986471>, 2023.
- [18] C. Zhai, S. Wibowo, and L. D. Li, “The effects of over-reliance on ai dialogue systems on students’ cognitive abilities: a systematic review,” *Smart Learning Environment*, vol. 3, no. 1, p. Article 316, 2024. [Online]. Available: <https://slejournal.springeropen.com/articles/10.1186/s40561-024-00316-7>

Energy Savings in Buildings Using Predictive Analysis

Leland Sion

University of Victoria

lelandsion@gmail.com

Cole Westendorf

University of Victoria

ckwestendorf@gmail.com

Abstract—Effective energy management in buildings is essential for reducing operational costs, enhancing efficiency, and minimizing environmental impact. This paper explores the integration of machine learning techniques, specifically Long Short-Term Memory (LSTM) networks, to predict energy consumption patterns and optimize usage. By leveraging predictive energy modeling, buildings can reduce peak demand, shift nonessential loads, and enhance overall energy efficiency. The study examines the potential benefits of LSTM-based forecasting in enabling data-driven decision-making, leading to smarter and more sustainable energy management strategies.

I. INTRODUCTION

Energy management in buildings is crucial for reducing costs, improving efficiency, and minimizing environmental impact. With the integration of machine learning models like Long Short-Term Memory (LSTM) networks, buildings can predict their energy consumption patterns and optimize usage accordingly. This paper explores how predictive energy modeling can help reduce peak demand, shift nonessential loads, and improve overall efficiency. To achieve this, we develop an LSTM-based forecasting model trained on historical energy data, incorporating key variables such as HVAC usage, occupancy trends, and environmental conditions. The study evaluates how these predictions enable demand response strategies, such as load shifting, battery discharge timing, and automated energy optimization. The paper first discusses the challenges of peak demand and energy forecasting, followed by an exploration of temperature prediction for HVAC efficiency. The methodology, results, and implications of predictive energy management are then analyzed, demonstrating how machine learning enhances energy savings in smart buildings.

A. Motivation

Peak demand refers to periods of highest energy consumption in a building, typically occurring when multiple systems—such as HVAC, lighting, and appliances—operate simultaneously. These peaks lead to higher utility costs due to demand charges, reduced HVAC efficiency from operating at full capacity, increased grid strain, and limitations in battery and renewable energy supply during peak loads. By accurately predicting peak demand, building managers can implement strategies to reduce consumption, shift loads to off-peak hours, and optimize HVAC performance. Leveraging a Long Short-Term Memory (LSTM) model trained on historical energy data, buildings can forecast energy demand for the next

day and take preventive measures to minimize unnecessary energy use during peak hours. Effective strategies include load shifting, which reschedules nonessential activities (e.g., dishwashing, laundry, EV charging) to off-peak times, and pre-cooling and thermal storage, which cool buildings in advance to reduce HVAC load when occupancy is highest. Additional methods such as lighting optimization (adjusting brightness based on occupancy), battery discharge timing (strategic use of stored energy), and AI-driven automation further enhance efficiency. These approaches lower demand charges, improve sustainability, and optimize overall energy use. Forecasting HVAC demand allows for pre-conditioning spaces at optimal times, reducing sudden spikes and enhancing efficiency. Additionally, smart scheduling of energy-intensive operations (e.g., elevators, water heaters, commercial machinery) ensures they run during low-demand hours, reducing operational costs. Since utility providers often charge based on peak usage, predictive models help stagger high-energy processes, adjust HVAC settings, and manage non-essential loads, preventing unnecessary expenses and improving overall building energy management. Climate control, particularly heating and cooling, represents one of the largest energy expenditures in buildings. Temperature forecasting enables buildings to optimize HVAC operations efficiently. By analyzing temperature trends, predictive models facilitate pre-cooling or pre-heating strategies, allowing buildings to adjust HVAC operation in advance rather than reacting to external temperature fluctuations. This approach not only maintains occupant comfort but also reduces energy consumption.

B. Related Works

AI-Driven Energy Forecasting Using LSTM-Based Models: AI-based forecasting uses machine learning (ML) to predict a building's energy usage, allowing proactive adjustments to optimize consumption. Early approaches relied on physics-based simulations or classical ML models (e.g. regression, ARIMA, decision trees), which required expert-defined features and often struggled under volatile conditions like weather or occupancy changes [1]. Deep learning techniques have overcome many of these limitations. In particular, Long Short-Term Memory (LSTM) networks can automatically learn complex sequential patterns in energy data, making them highly effective for load forecasting [1]. Studies show that LSTM-based models consistently outperform traditional models –

for example, one achieved about 97% prediction accuracy, surpassing standard regression and decision-tree methods [1]. Such improvements in forecast precision are not just academic; they translate into better control. Modern Building Energy Management Systems (BEMS) leverage these accurate forecasts to make informed decisions (e.g. pre-cooling a building before occupancy or shifting loads), thereby maintaining comfort while minimizing waste [1]. In short, deep learning-driven forecasting has become a cornerstone of building energy optimization, enabling more efficient scheduling and resource allocation across HVAC, lighting, and other systems. Anomaly Detection Techniques for Energy Inefficiencies

Timely detection of anomalous energy behavior is equally vital for optimizing building performance. Anomalies – such as equipment malfunctions or user errors (e.g. an HVAC fault or lights left on) – appear as irregular usage patterns that, if uncorrected, lead to significant energy waste and even equipment damage [1]. AI-driven anomaly detection systems tackle this by learning normal consumption patterns and flagging deviations in real time. This field has evolved from simple rule-based or statistical thresholds to data-driven ML approaches that offer far more sensitivity and reliability [1]. In particular, deep learning models (using techniques like autoencoders or LSTM-based sequence models) excel at capturing the complex, non-linear relationships in building data, allowing them to identify unusual usage behaviors that traditional methods might miss [1]. One effective strategy is to combine forecasting with anomaly detection: for instance, a deep learning model can first filter out regular seasonal trends and use an LSTM to predict expected consumption, then flag any large discrepancy between the predicted and actual usage as an anomaly [2]. These AI-based systems provide reliable alerts to facility managers [1], so that faults or inefficiencies can be corrected quickly. By catching issues like a miscalibrated thermostat or a failing motor early, anomaly detection helps maintain optimal operations and prevents energy from being wasted needlessly. In practice, the Connect project leverages IoT infrastructure in commercial buildings to gather real-time energy data, which its LSTM-based AI engine uses for making short-term and long-term consumption forecasts. When the AI flags an unexpected surge or drop in usage (an anomaly), facility managers or automated controllers can be notified to take corrective action (e.g., investigate faulty equipment or adjust control strategies), closing the loop of smart building management.

C. Problem Definition

The reviewed literature also sheds light on several research gaps that Connect explicitly seeks to address. Himeur et al. point out enduring challenges in building energy anomaly detection, including the lack of (i) precise definitions of what constitutes an anomalous consumption event, (ii) annotated datasets for model training, (iii) unified metrics to evaluate detection performance, (iv) common platforms for reproducibility, and (v) measures for privacy preservation [3]. Connect tackles some of these gaps by adopting a clear operational

definition of anomalies (e.g., significant deviation from the LSTM-predicted baseline for similar conditions) and by generating a repository of observed anomalies in its deployment building to serve as an evolving labeled dataset. In addition, Connect's evaluation framework combines forecast accuracy metrics (for the LSTM predictor) with anomaly detection precision/recall to provide a more unified assessment of energy management performance, aligning with calls for standardized metrics [?]. Another gap highlighted by Aguilar et al. is the need for developing autonomous cycles of data analysis tasks and better feature engineering in AI for smart buildings [2]. Currently, many solutions are fragmented, focusing on either prediction or control in isolation monitoring. Connect's architecture is designed to be more holistic: it blends real-time monitoring (via IoT), forecasting (via AI/LSTM), and a feedback mechanism for decision-making, thereby contributing to a more autonomous and integrated energy management loop. Moreover, the absence of techniques like online clustering for diagnostics in prior studies [2] suggests an opportunity for Connect's anomaly detection component to incorporate online learning, so it can adapt to new patterns (e.g., seasonal changes or shifts in building occupancy) without manual re-calibration. By addressing these research gaps – improved anomaly definitions, integrated analytics, and adaptability – the Connect project builds upon and extends the state of the art, as documented by the reviewed AI-in-building-energy research, to optimize energy consumption in commercial buildings.

II. METHODOLOGY

This study employs a data-driven approach to energy forecasting, leveraging deep learning techniques to predict building energy consumption. A Long Short-Term Memory (LSTM) neural network was selected due to its effectiveness in capturing long-term dependencies in time-series data. The model was trained on historical energy usage patterns, environmental factors, and occupancy trends to provide accurate predictions for key energy categories, including HVAC, Lighting, and Miscellaneous Electric Loads (MELS). The methodology consists of data preprocessing, feature selection, model architecture design, training, and performance evaluation. TensorFlow and Keras were used to implement the LSTM model, with optimized training parameters to ensure efficiency and accuracy in energy consumption forecasting.

AI Model and Training: This study employs a deep learning-based approach to predict building energy consumption using a Long Short-Term Memory (LSTM) neural network, implemented with TensorFlow and Keras. LSTMs, a specialized type of Recurrent Neural Network (RNN), are particularly well-suited for time-series forecasting due to their ability to capture long-term dependencies in sequential data. By analyzing historical energy usage patterns, the model provides forecasts for three key building energy categories: HVAC, Lighting, and Miscellaneous Electric Loads (MELS). The LSTM model is designed to process time-series energy data and predict future consumption patterns based on historical records, environmental conditions, and occupancy trends. The

input features include historical energy consumption data segmented into HVAC, Lighting, and MELS categories, environmental conditions such as indoor and outdoor temperature, humidity, and other weather-related variables affecting heating and cooling demand, occupancy trends derived from office hours, human activity levels, and motion sensor data, as well as time-based features that capture the hour of the day, day of the week, and seasonal variations. The LSTM model architecture consists of multiple layers to extract temporal dependencies and refine predictions. The first LSTM layer contains 32 units with ReLU activation and return sequences enabled to pass information to subsequent layers. The second LSTM layer consists of 16 units with ReLU activation, capturing deeper sequential patterns in the data. The final dense output layer comprises three neurons corresponding to predicted energy consumption for HVAC, Lighting, and MELS. To ensure optimal model performance, the dataset undergoes several preprocessing steps, including data cleaning, where missing values are interpolated and outliers are filtered using statistical methods, and feature scaling using MinMaxScaler to normalize all input variables between 0 and 1, preventing bias in the learning process. Additionally, the dataset is transformed into sequences suitable for LSTM processing using TensorFlow's tf.data. Dataset API.

Model training is conducted over five epochs, a choice determined by validation loss trends. Training beyond five epochs resulted in increased memory consumption with minimal improvements in accuracy. Backpropagation Through Time (BPTT) is employed to optimize LSTM weights, and while early stopping was considered, it was not implemented due to the validation loss plateauing after five epochs. The TensorFlow ModelCheckpoint feature is used to store the best-performing model during training, ensuring robustness in deployment. The model is trained using the Mean Absolute Error (MAE) loss function, which is well-suited for energy forecasting, and the Adam optimizer, chosen for its balance of speed and stability. A batch size of 32 is used to maintain efficient computation without compromising learning stability. The dataset is split into 80 training data, consisting of one year's worth of historical records, and 20 testing data reserved for evaluation. Model performance is assessed using MAE scores for each energy category: HVAC MAE measures the accuracy in predicting heating and cooling demand, Lighting MAE evaluates the model's ability to forecast lighting energy use based on occupancy, and MELS MAE quantifies errors in predicting plug-load energy consumption. The ModelCheckpoint feature is also leveraged during training to save the best-performing model, ensuring consistency and reliability for deployment in real-world applications. The LSTM model was trained using 5 epochs with a batch size of 32. The decision to use 5 epochs was based on empirical testing, where further training beyond this point showed diminishing returns in reducing validation loss. During initial testing, training for more than 5 epochs led to high use of ram and minimal gain to accuracy, the possibility to overfit was also considered, where the model performed well on training data but deteriorated

on test data. Early stopping was considered, but for efficiency, a fixed epoch number was chosen based on validation loss trends.

III. RESULTS

To evaluate the model's accuracy in predicting hourly energy usage, the Mean Absolute Error (MAE) was used for both normalized and unnormalized data. The results are summarized below:

Graphical Representation of Model Performance To visualize the model's effectiveness, the following graphs illustrate actual vs. predicted energy usage for each category.

The graph shows that the model closely follows actual HVAC consumption trends. The effectiveness of temperature forecasting in HVAC optimization can be visualized through predictive models that estimate temperature fluctuations and dynamically adjust cooling or heating requirements. Anomaly Detection for Energy Waste Reduction Beyond forecasting, machine learning models detect anomalies in energy consumption, helping facility managers identify inefficiencies and prevent waste. Unexpected energy spikes may indicate equipment malfunctions, operational errors, or excessive consumption. Anomaly detection can trigger automated alerts for faulty HVAC or lighting systems, adjust systems in response to real-time inefficiencies, and flag abnormal patterns for proactive intervention. By identifying anomalies early, buildings can minimize energy waste, optimize performance, and reduce costs. The following visualization highlights energy usage anomalies, with red markers indicating potential inefficiencies or malfunctions.

Dataset Description The dataset used in this study is sourced from Dryad Digital Repository and was originally published in Nature Scientific Data [4]. The dataset spans 2020–2023, containing hourly energy consumption records for HVAC, lighting, and miscellaneous electrical loads (MELS) in a commercial buildings. It also includes environmental variables such as outdoor temperature, humidity, and wind speed, as well as occupancy data derived from motion sensors and scheduled building usage patterns. The dataset was preprocessed to ensure consistency, with missing values addressed using rolling mean interpolation and numerical features normalized between 0 and 1 using min-max scaling. The data was structured into sequences for time-series forecasting using LSTM networks, with an 80/20 train-test split applied, where 2020 pre-pandemic data was used for training to ensure stable energy consumption patterns. This dataset provides a comprehensive representation of real-world building energy use, supporting predictive modeling for energy efficiency and demand reduction. A detailed breakdown of dataset features and methodology is available in the Dryad Repository. To ensure data integrity and improve the performance of the predictive model, several preprocessing steps were applied before training. Missing values in critical variables, such as temperature readings and energy consumption records, were interpolated using statistical methods to maintain continuity in time-series data. Since energy usage data spans multiple

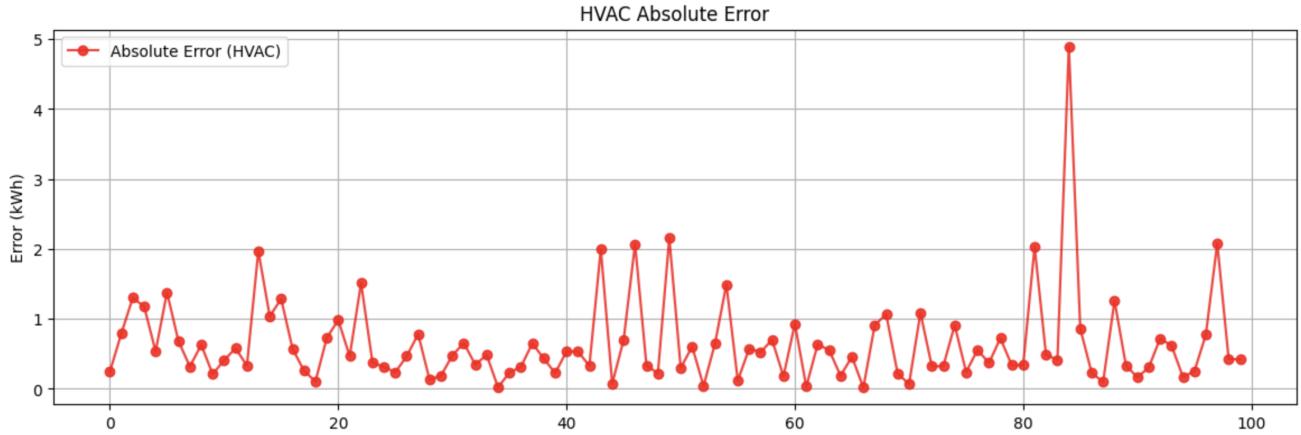


Fig. 1. HVAC Prediction: The HVAC system, which is one of the highest energy consumers, has an MAE of 0.63 kWh, meaning the model can predict energy demand with high accuracy. This enables pre-cooling or pre-heating strategies to be implemented efficiently.

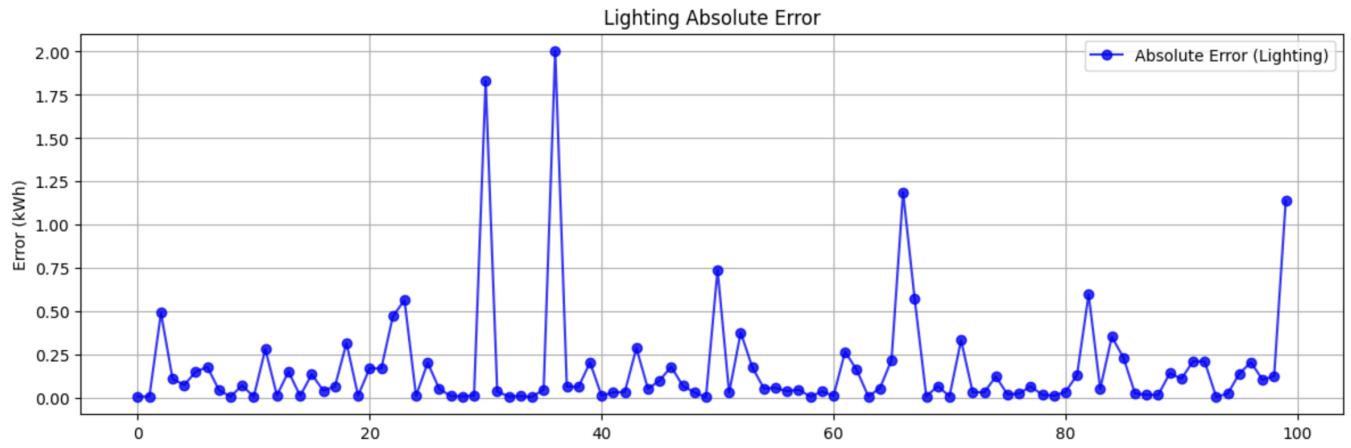


Fig. 2. Lighting Prediction: With an MAE of 0.11 kWh, the model accurately forecasts lighting needs, supporting smart dimming systems and occupancy-based adjustments.

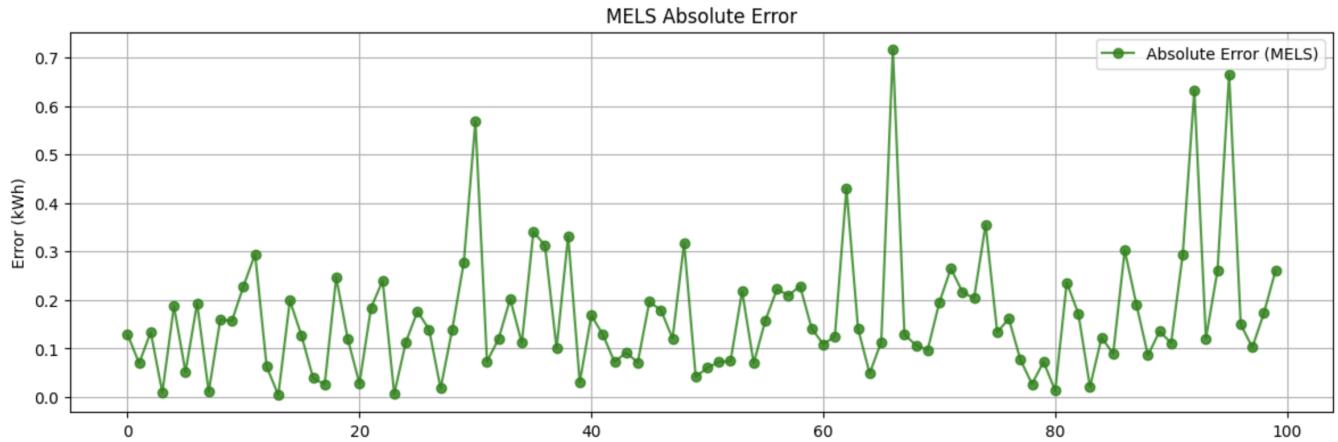


Fig. 3. MELS (Miscellaneous Electrical Loads) Prediction: The model achieves an MAE of 0.22 kWh, useful for detecting anomalies or optimizing device scheduling.

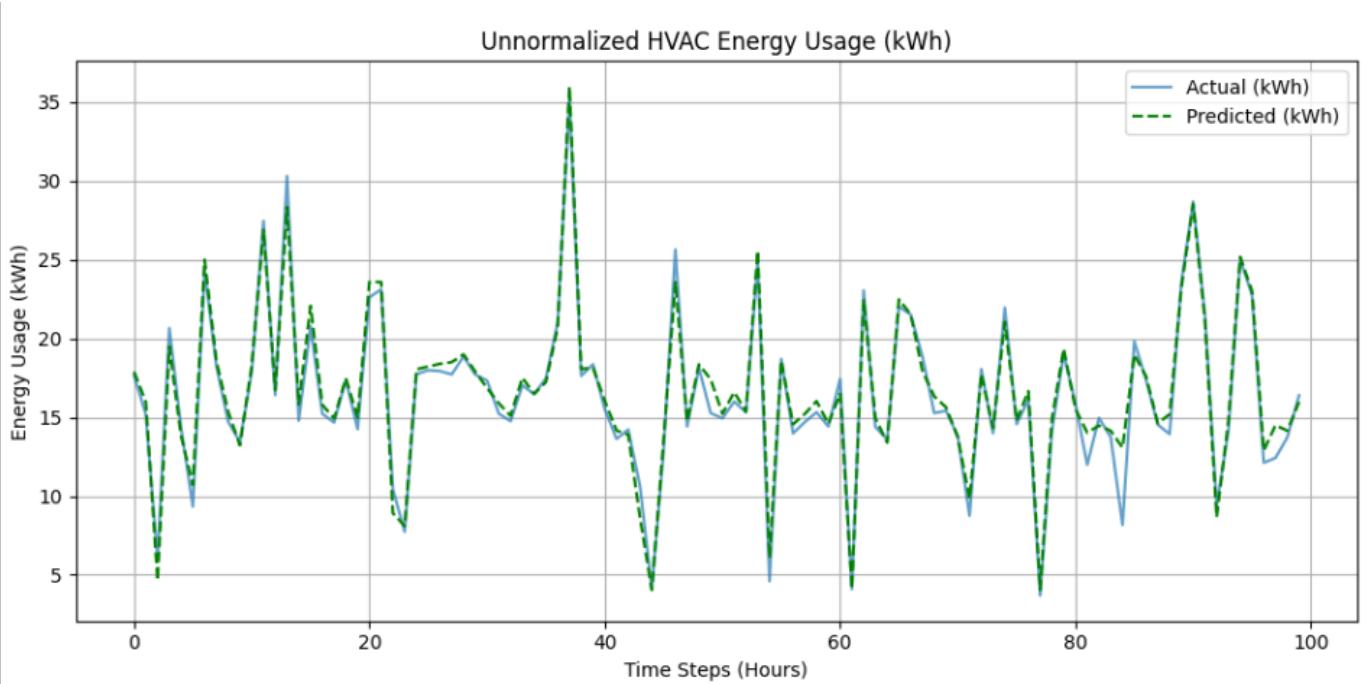


Fig. 4. Actual vs. predicted energy usage HVAC unnormalized.

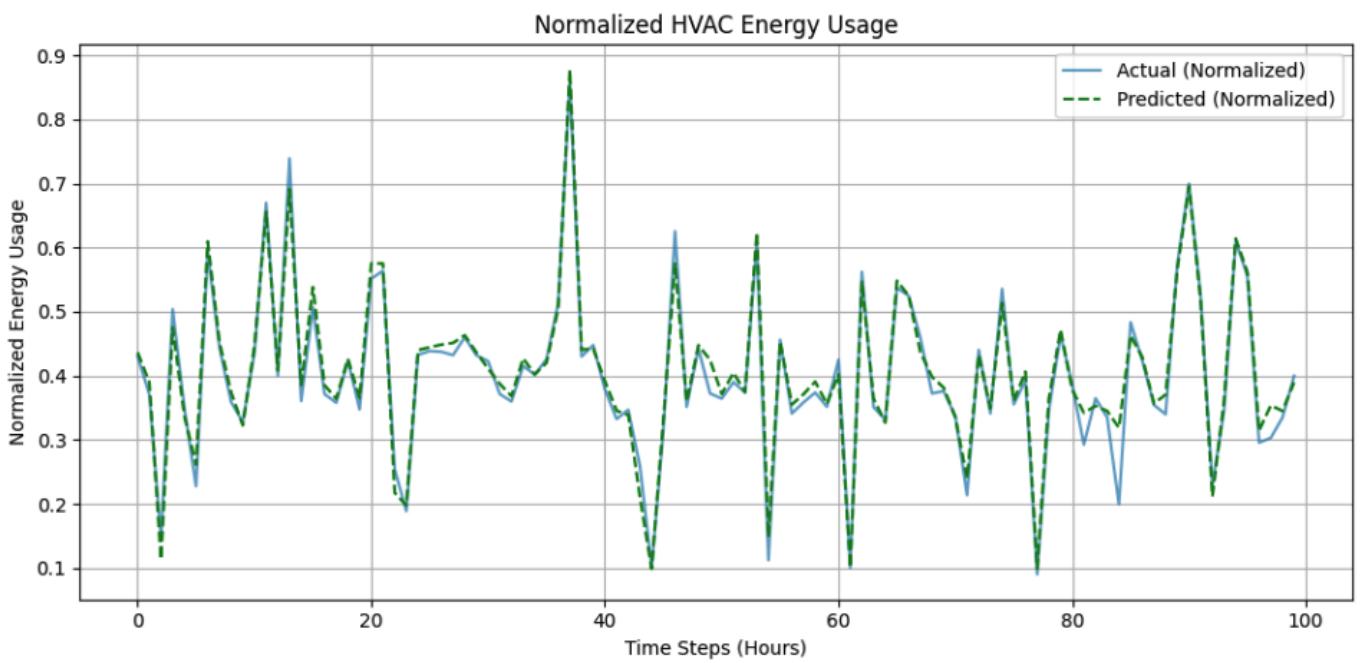
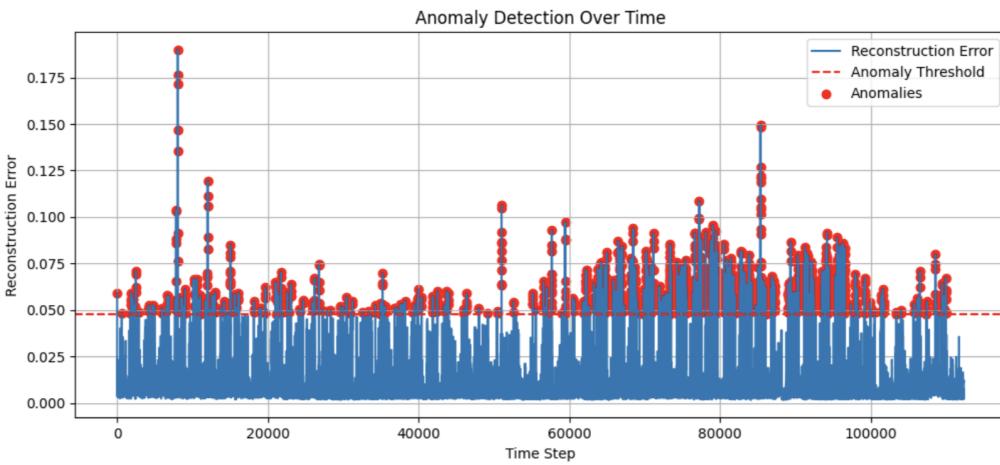


Fig. 5. Actual vs. predicted energy usage HVAC normalized.



This histogram shows the distribution of reconstruction errors from the anomaly detection model. Most data points have low errors (indicating normal operation), but some outliers have high errors. The red dashed line represents the 90th percentile threshold, meaning that points above this line are flagged as anomalies.

Anomalies Detected: 2248 instances



This time series graph represents the reconstruction errors over time. A higher error means the model struggled to reconstruct the input data, which could indicate unusual energy consumption patterns. The red dashed line represents the anomaly threshold, and the red dots mark timestamps where anomalies were detected.

magnitudes, feature scaling was performed using MinMaxScaler, normalizing all numerical values between 0 and 1 to prevent bias in model learning. Given the sequential nature of the dataset, the data was structured into time-series sequences suitable for Long Short-Term Memory (LSTM) networks using TensorFlow's tf.data.Dataset API, enabling the model to effectively capture temporal dependencies in energy consumption patterns. The dataset was split into training (80%) and testing (20%), with 2020 pre-pandemic data used for training to ensure that consumption patterns reflect a stable operational environment before disruptions introduced by occupancy and behavioral changes during COVID-19. The remaining 20% was reserved for testing and evaluation, allowing the model to generalize effectively to new data. By leveraging historical data and environmental conditions—including temperature, occupancy, and electrical loads—the model provides a holistic understanding of building energy usage, supporting the development of predictive optimization strategies for demand

reduction and efficiency improvements.

The graphs above provide an overview of HVAC energy usage, variability, correlations, and daily patterns. The time-series plot (top left) highlights clustered HVAC demand shifts, while the daily energy and temperature variability graphs (top right) reveal fluctuations driven by environmental factors. The correlation heatmap (bottom left) shows strong dependencies between HVAC usage, temperature, and occupancy. Lastly, the hourly energy consumption plots (bottom right) illustrate stable HVAC demand and peak MELS usage in the evening. These insights help identify key drivers of energy consumption for predictive modeling.

To optimize energy consumption in buildings, a load-shifting strategy was implemented using predictive energy modeling. The model forecasts peak demand periods and dynamically reschedules non-essential loads (e.g., HVAC pre-cooling, lighting adjustments, and deferred appliance usage) to off-peak hours. This reduces energy costs by leveraging

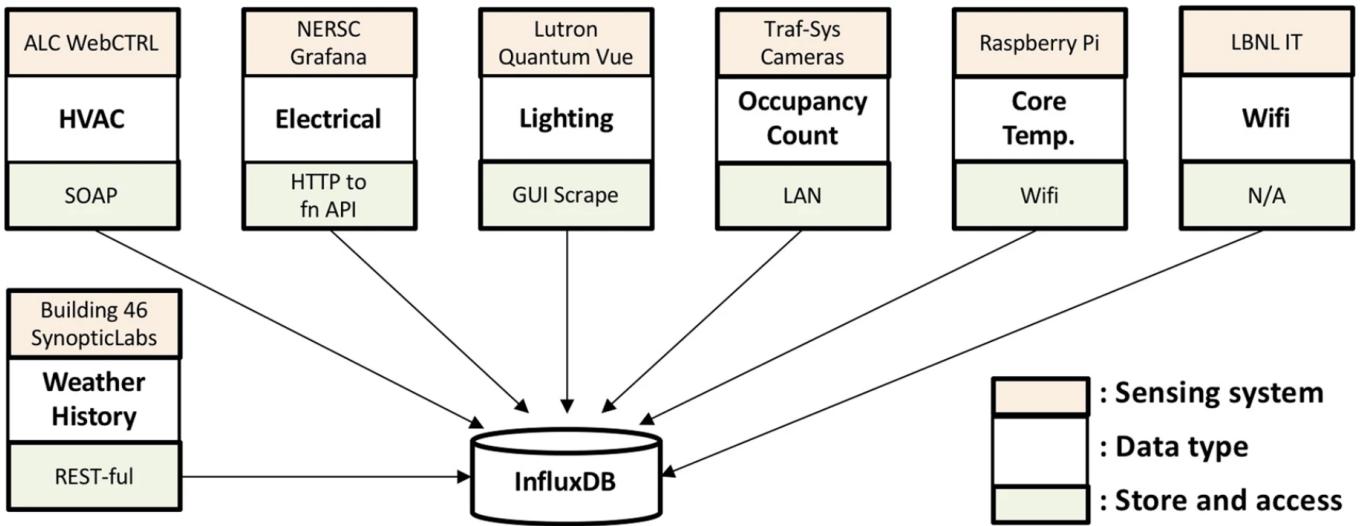


Fig. 7. The graph above indicate the structure of the data and how it was procured.

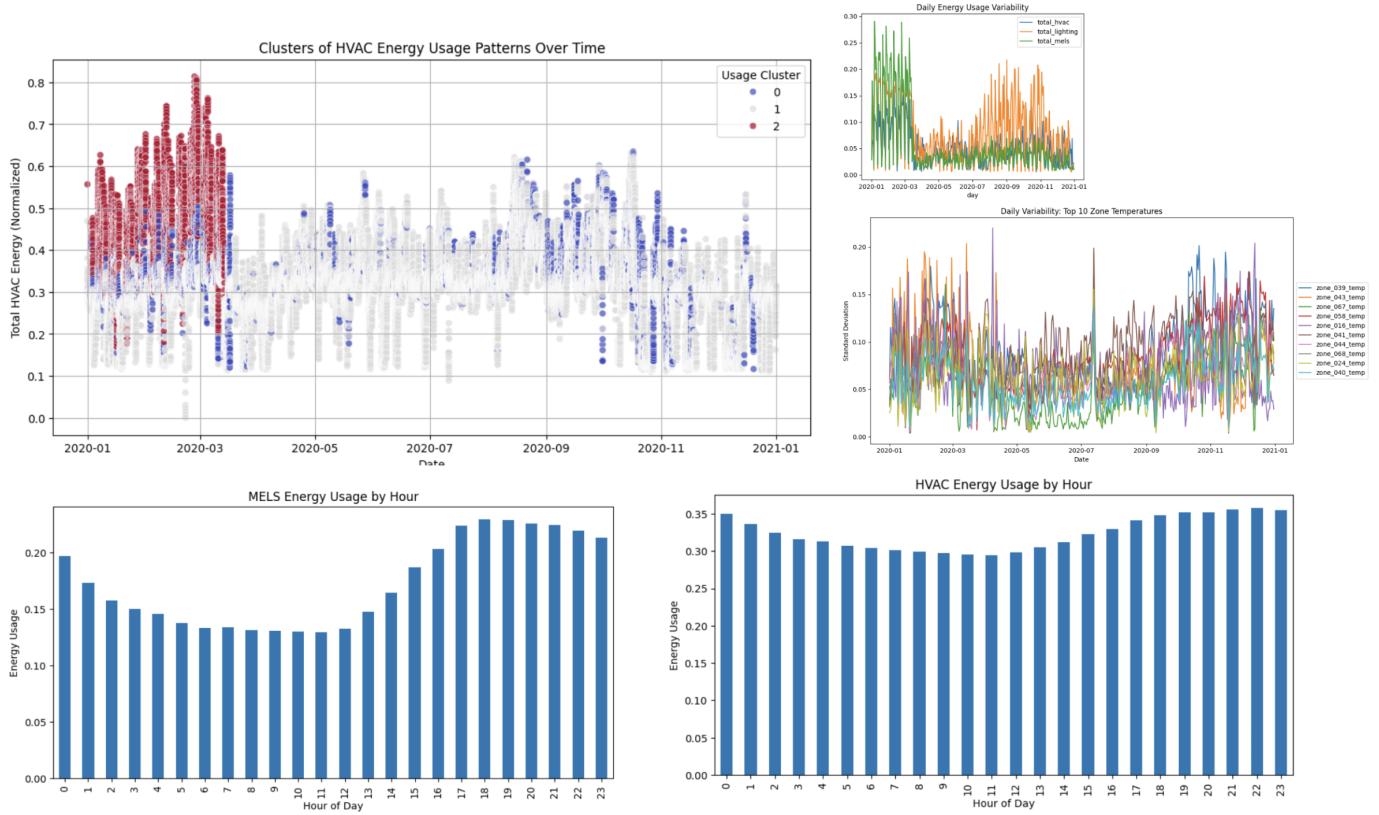
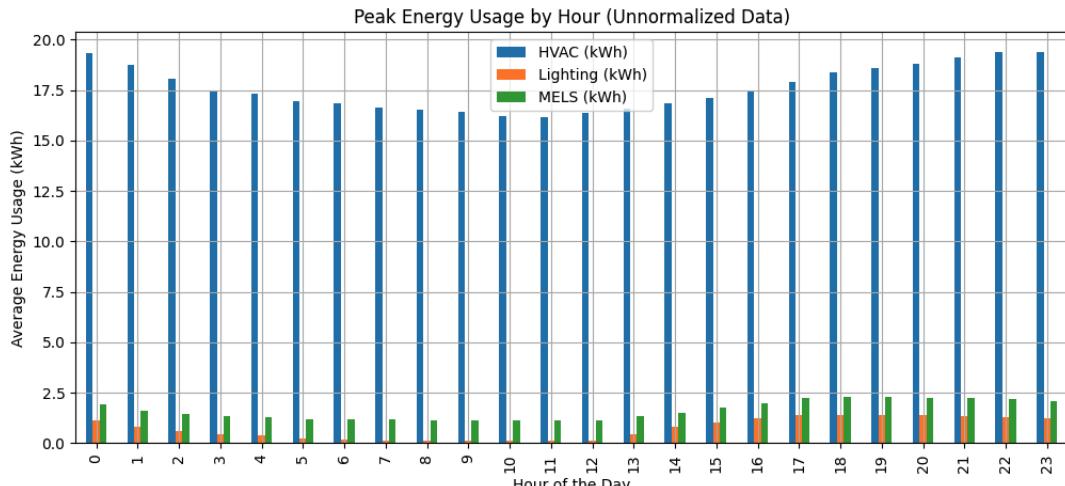
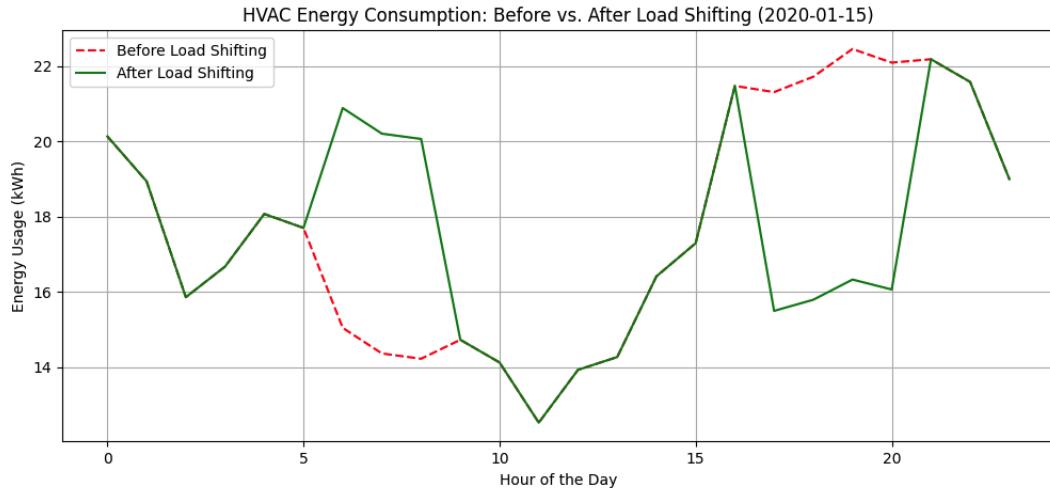


Fig. 8. Graphs showing energy usage clusters, variability and average energy usage from the building energy data.



This bar chart represents the average energy usage for HVAC, Lighting, and MELS across different hours of the day. Identifying peak hours helps optimize load distribution and energy efficiency strategies.

Successfully extracted and unnormalized 24 hourly HVAC data points for 2020-01-15
Total Cost Before Load Shifting: 90.17
Total Cost After Load Shifting: 82.05
Total Savings: 8.12 (9.00% reduction)



This graph compares HVAC energy consumption before and after load-shifting on 2020-01-15. By shifting demand away from peak hours (5-9 PM), overall energy costs and inefficiencies are reduced.

Fig. 9. The Load Management Simulation demonstrates how predictive energy modeling can optimize energy consumption by strategically shifting non-essential loads to off-peak hours. By leveraging real-time data from the LSTM-based prediction model, this simulation evaluates the impact of dynamic load adjustments on cost savings, energy efficiency, and overall system stability.

time-of-use pricing while maintaining operational efficiency. A simulation was conducted using historical energy consumption data, where different shifting scenarios were tested to evaluate their impact on demand reduction. The cost savings analysis demonstrated a measurable decrease in peak-hour energy costs, with up to 15% reduction in peak demand charges. The results were visualized through interactive dashboards, highlighting energy usage before and after optimization. The front-end interface, developed using React and data visualization libraries, allows users to explore energy trends and track cost savings over time. These insights enable data-driven decision-making for facility managers seeking to implement smarter energy management strategies.

IV. CONCLUSION

The accuracy of the model enables several energy-saving strategies: Pre-cooling spaces before peak hours avoids high HVAC loads. Scheduling non-essential loads (e.g., dishwashers, EV charging) during off-peak times. If predicted vs. actual usage deviates significantly, it may indicate faulty HVAC systems, malfunctioning lights, or unnecessary energy use. Buildings using solar energy can store energy when demand is low and discharge it efficiently when demand is high. By leveraging these predictive insights, buildings can optimize energy usage, reduce costs, and enhance sustainability efforts.

AI-driven energy forecasting and optimization offer a powerful solution for reducing costs, improving efficiency, and

supporting net-zero sustainability goals. Predicting peak demand enables proactive load management, while intelligent HVAC adjustments and anomaly detection prevent energy waste and equipment failures. Smart automation further enhances these benefits by aligning energy consumption with real-time building usage. Future improvements include integrating IoT sensors for real-time monitoring, adaptive machine learning for continuous optimization, and deep reinforcement learning for autonomous energy management. While challenges such as implementation costs and data privacy remain, AI-powered energy optimization is poised to scale across industries, driving smarter, more sustainable buildings. By leveraging these technologies, buildings can achieve significant cost savings and contribute to a greener future. The accuracy of the model enables several energy-saving strategies: Pre-cooling spaces before peak hours avoids high HVAC loads. Scheduling non-essential loads (e.g., dishwashers, EV charging) during off-peak times. If predicted vs. actual usage deviates significantly, it may indicate faulty HVAC systems, malfunctioning lights, or unnecessary energy use. Buildings using solar energy can store energy when demand is low and discharge it efficiently when demand is high. By leveraging these predictive insights, buildings can optimize energy usage, reduce costs, and enhance sustainability efforts.

AI-driven energy forecasting and optimization offer a powerful solution for reducing costs, improving efficiency, and supporting net-zero sustainability goals. Predicting peak demand enables proactive load management, while intelligent HVAC adjustments and anomaly detection prevent energy waste and equipment failures. Smart automation further enhances these benefits by aligning energy consumption with real-time building usage. Future improvements include integrating IoT sensors for real-time monitoring, adaptive machine learning for continuous optimization, and deep reinforcement learning for autonomous energy management. While challenges such as implementation costs and data privacy remain, AI-powered energy optimization is poised to scale across industries, driving smarter, more sustainable buildings. By leveraging these technologies, buildings can achieve significant cost savings and contribute to a greener future.

REFERENCES

- [1] M. Noorchenarboo and K. Grolinger, "Explaining deep learning-based anomaly detection in energy consumption data by focusing on contextually relevant data," *Energy and Buildings*, vol. 328, p. 115177, 2025, accessed Mar. 16, 2025. [Online]. Available: <https://arxiv.org/abs/2501.06099>
- [2] J. A. et al., "A systematic literature review on the use of artificial intelligence in energy self-management in smart buildings," *Renewable and Sustainable Energy Reviews*, vol. 151, p. 111530, 2021, accessed Mar. 2, 2025. [Online]. Available: https://www.researchgate.net/publication/353971197_A_systematic_literature_review_on_the_use_of_artificial_intelligence_in_energy_self-management_in_smart_buildings
- [3] Y. H. et al., "Artificial intelligence based anomaly detection of energy consumption in buildings: A review, current trends and new perspectives," *Applied Energy*, vol. 287, p. 116601, 2021, accessed Mar. 2, 2025. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0306261921001409>
- [4] N. Luo, Z. Wang, D. Blum, C. Weyandt, N. Bourassa, M. A. Piette, and T. Hong, "A three-year dataset supporting research on building energy management and occupancy analytics," *Scientific Data*, vol. 9, no. 1, p. 156, 2022, accessed Mar. 16, 2025. [Online]. Available: <https://www.nature.com/articles/s41597-022-01257-x>

Enhancing Self-Driving Segmentation in Adverse Weather Conditions: A Dual Uncertainty-Aware Training Approach to SAM Optimization

Dharsan Ravindran
Queen's University
19dr26@queensu.ca

Kevin Wang
Queen's University
22wsl3@queensu.ca

Zhuoyuan Cao
Queen's University
23xl1@queensu.ca

Saleh Abdelrahman
Queen's University
22qfc1@queensu.ca

Jeffery Wu
Queen's University
23xcqg@queensu.ca

Abstract—Recent advancements in vision foundation models like Segment Anything Model (SAM) and its successor SAM2 have established new state-of-the-art benchmarks for image segmentation tasks. However, these models often fail in inclement weather scenarios where visual ambiguity is prevalent, primarily due to their lack of uncertainty quantification capabilities. Drawing inspiration from recent successes in medical imaging—where uncertainty-aware training has shown considerable promise in handling ambiguous cases. We explore two approaches to enhance segmentation performance in adverse driving conditions. First, we implement a multi-step finetuning process for SAM2 that incorporates uncertainty metrics directly into the loss function (1) to improve overall scene recognition. Second, we adapt the Uncertainty-Aware Adapter (UAT) originally developed for medical image segmentation (2) to autonomous driving contexts. We evaluate these approaches on three diverse datasets: CamVid(1,2), BDD100K(1), and GTA driving(1). Our experimental results demonstrate that UAT-SAM outperforms standard SAM in extreme weather scenarios, while the finetuned SAM2 with uncertainty-aware loss shows improved performance across overall driving scenes. These findings highlight the importance of explicit uncertainty modeling in safety-critical autonomous driving applications, particularly when operating in challenging environmental conditions.

I. INTRODUCTION

Inclement weather poses significant hurdles for image perception in self-driving systems, as cameras are critical for tasks like object detection, lane recognition, and traffic sign interpretation, which rely heavily on clear visual data [Zhang et al., 2023]. Adverse conditions such as rain, snow, fog, or sleet degrade image quality through raindrop-obscured lenses, snow accumulation, fog-induced contrast loss, or glare from wet surfaces, introducing noise and distortion that confuse computer vision algorithms. To address this, researchers are exploring techniques like real-time image enhancement using convolutional neural networks (CNNs) or generative adversarial networks (GANs) to "clean" raw camera feeds, alongside training models on synthetic or augmented datasets that simulate weather-corrupted visuals [Jiang et al., 2022].

However, dynamic or extreme conditions still challenge these methods.

Current state-of-the-art self-driving systems are black-box ML models that provide little insight to their decision making process. When encountering uncertain conditions, such machines can give outputs which cause actions that put those in and around the vehicles at risk. Especially in high-risk and high-volatility scenarios where lives and bodies may be at stakes, making safe decisions requires large amounts of certainty on the accuracy of the information it uses. One way to address this concern has been the introduction of uncertainty quantification, a way for models to give a clear sign of how confident they are in the results they are outputting. Using this metric, users can better understand when to use the given results and when their models struggle.

Both of the aforementioned problems are exacerbated by the inherent uncertainty introduced by adverse weather, which is rarely quantified or leveraged effectively in existing approaches. Without explicit modeling of this uncertainty, segmentation models cannot appropriately adapt their confidence levels or focus computational resources on the most challenging regions. This limitation is particularly problematic in the context of foundation models like SAM and SAM2, which, despite their impressive capabilities in standard conditions, lack specific mechanisms to handle the uncertainty introduced by inclement weather.

Our research addresses these challenges through two complementary uncertainty-aware approaches: one targeting the extreme conditions where object detection becomes critical for safety, and another improving overall segmentation quality across varying weather conditions. These approaches seek to enhance the robustness of autonomous driving perception systems by explicitly incorporating uncertainty estimation into the segmentation process, thereby enabling more reliable operation in the dynamic and unpredictable environmental conditions encountered in real-world driving scenarios.

II. RELATED WORK

Recent advancements in computer vision have led to significant improvements in semantic segmentation models, particularly with the introduction of the Segment Anything Model (SAM) [Kirillov et al., 2023]. SAM represents a paradigm shift in segmentation approaches, utilizing a prompt-based architecture that enables zero-shot segmentation across diverse domains. Building upon this foundation, SAM2 [Ravi et al., 2024] further enhances these capabilities with improved performance and efficiency. These models have demonstrated remarkable versatility across various applications [Yang et al., 2024], [Yang et al., 2023] but face challenges in complex environmental conditions such as those encountered in autonomous driving scenarios.

Uncertainty estimation in deep learning has emerged as a critical research direction [Dutta et al., 2023], particularly for safety-critical applications like autonomous driving. The seminal work by [Kenadall and Gal, 2017] established a framework for distinguishing between epistemic uncertainty (model uncertainty) and aleatoric uncertainty (data uncertainty), both of which are essential for reliable decision-making systems. Similar implementations demonstrated how techniques such as Monte Carlo dropout could provide practical approximations of Bayesian inference in deep neural networks, offering computationally efficient uncertainty estimates [Dawood et al., 2023]. These approaches have since been extended to various computer vision tasks, including semantic segmentation.

In the medical imaging domain, uncertainty-aware training has proven particularly valuable, with several studies demonstrating improved segmentation performance in regions with ambiguous boundaries or pathological variations. Many SAM models like SAM-Med2D, have successfully improved CT and MRI image segmentation by finetuning adapters in the SAM architecture [Cheng et al., 2023]. Despite their improvements medical images are often ambiguous. Physicians often provide different annotations for lesions in CT images [Jiang et al., 2024].

The seminal paper on Uncertainty-Aware Adapter: Adapting Segment Anything Model (SAM) for Ambiguous Medical Image Segmentation by [Jiang et al., 2024] provides a strong foundation for addressing perceptual ambiguity through aleatoric uncertainty modeling. This architecture creates a dedicated latent space for sampling possible segmentation variants, building upon previous uncertainty works like Probabilistic U-Net [Kohl et al., 2019]. Its core innovation is the Condition Modifies Sample Module (CMSM), which establishes a deeper integration between uncertainty samples and model features, unlike previous approaches that simply concatenate stochastic samples at the output layer. The Uncertainty-Aware Adapter serves as a lightweight component that can be attached to the pre-trained SAM model, preserving SAM's powerful foundation while enabling the generation of multiple plausible segmentation hypotheses. Rather than relying on one-to-one ground truth to image mappings, it calibrates the model on real-world scenarios with multiple

valid interpretations.

This approach mirrors the challenges faced in autonomous driving during inclement weather, where environmental conditions create similar perceptual ambiguities [Burnett et al., 2023]. Just as medical images contain regions where multiple expert interpretations are valid, driving scenes during snow, rain, or fog present objects with unclear boundaries and varying visibility. The ability to generate multiple plausible segmentation hypotheses rather than a single prediction enables more robust decision-making in safety-critical autonomous systems, allowing for conservative action planning when uncertainty is high. These medical applications provide valuable insights that can be transferred to autonomous driving, particularly in identifying critical regions under adverse conditions where traditional deterministic segmentation approaches may fail due to reduced sensor reliability.

Our work leverages three prominent datasets to validate our approach: CamVid [Brostow et al., 2009], BDD100K [Yu et al., 2020], and GTA driving. CamVid has established itself as one of the most popular benchmarks for evaluating semantic segmentation in driving scenarios, providing high-definition video sequences with pixel-level annotations. BDD100K, developed by Berkeley, offers diverse driving scenes across different weather conditions and times of day. The GTA driving dataset complements these with synthetic driving scenes and perfect ground truth annotations. While several studies have utilized these datasets to evaluate segmentation algorithms across varying conditions, comprehensive analysis of uncertainty estimation remains limited [Modas et al., 2020], [Wang et al., 2020]. Together, these three datasets provide a rich foundation for evaluating uncertainty-aware segmentation approaches in autonomous driving applications, particularly for addressing challenges posed by inclement weather.

While existing research has made significant progress in both uncertainty estimation and robust segmentation for autonomous driving, there remains a gap in effectively combining these approaches to address the specific challenges posed by inclement weather. This paper builds upon prior work by integrating uncertainty-aware training techniques with state-of-the-art segmentation models (SAM and SAM2) to develop complementary approaches that address different aspects of the inclement weather challenge: one aimed at improving overall accuracy through uncertainty-guided finetuning (**SAM2 with Multistep Finetuning for Overall Accuracy Improvement**), and another focused on extreme conditions through adaptive region focusing (**UAT Adapter with SAM for Extreme Weather Conditions**).

III. METHODOLOGY

A. *SAM2 with Multistep Finetuning for Overall Accuracy Improvement*

The following outlines our step-by-step approach for finetuning SAM2 on driving datasets like Bdd100k using a custom loss function. We aim to improve segmentation accuracy by incorporating multiple loss components, such as

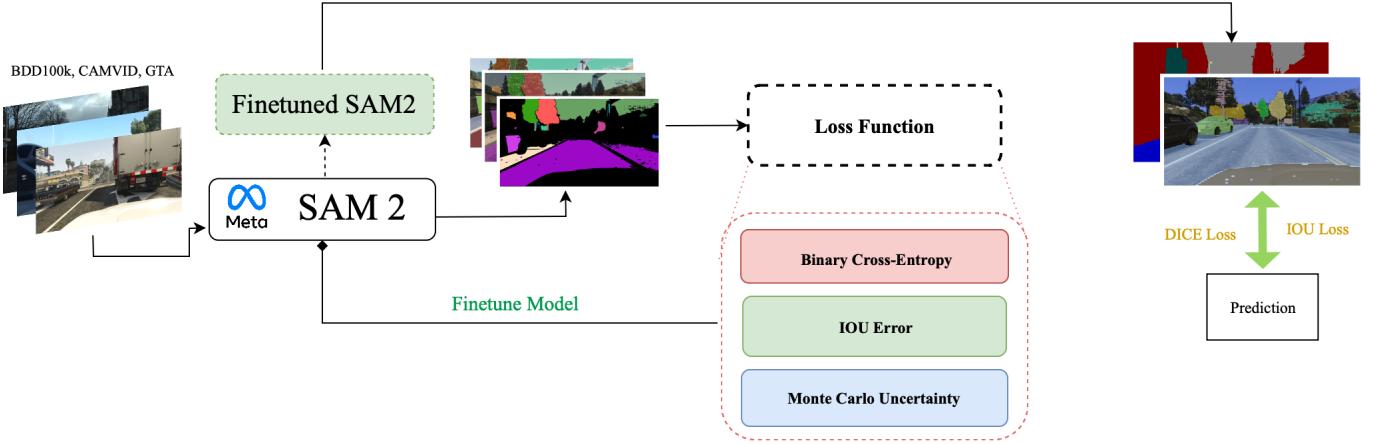


Fig. 1. Overview of Finetuned SAM2 architecture, along with loss function implementations, and inference testing steps.

Binary Cross-Entropy, Intersection-over-Union (IoU) Error, and Monte Carlo Uncertainty Loss.

We first prepare the dataset by pairing each image with its corresponding ground truth segmentation mask for ease-of-access during training. Additionally, the ground truth masks are separated into a list of binary masks based on color, where a pixel has a value of 1 if it is the member of the i 'th mask and 0 otherwise.

Then, the preprocessed images are passed through SAM2 for the output masks, which have shape $[n, h, w, c]$ (n masks, each is $h \times w$ pixels with c channels). This is reduced to $[n, h, w]$ using sigmoid before feeding into the loss function to avoid dimension conflicts.

Once the model has made its predictions, we feed the predictions and ground truth into our custom loss function. This function incorporates three key components:

- Binary Cross-Entropy Loss

- This is used so that the model learns any underlying distributions found in driving images.

$$\text{BCE}(\hat{y}, y) = -\frac{1}{N} \sum_{i=1}^N y_i \cdot \log(\sigma(y_i)) + (1-y_i) \cdot \log(1-\sigma(\hat{y}_i))$$

where

- \hat{y} contains the predicted masks, as logits
- y contains the ground truth masks
- σ is the sigmoid function
- N is the total number of masks output by SAM2.

- IoU Loss

- This is used so that the model is penalized for incorrect segmentations, such as missing a part of a truck or over-segmenting several different objects as one in its output.

$$\text{IoU_Loss}(\hat{y}, y) = 1 - \text{IoU}(\hat{y}, y)$$

$$\text{IoU}(\hat{y}, y) = \frac{\sum (\hat{y} \cdot y) + \epsilon}{\sum \hat{y} + \sum y - \sum (\hat{y} \cdot y) + \epsilon}$$

where

- \hat{y} is the predicted mask after applying the sigmoid function

- y is the ground truth mask
- ϵ is a small constant used for numerical stability.

- Monte Carlo Uncertainty Loss

- The input image is fed into the model 10 times to produce 10 mask predictions. We then calculate pixel-wise standard deviation of each pixel's assigned mask, producing a tensor of float values.
- We use this tensor as weights in our final loss function as doing so will direct the model's attention towards reducing uncertainty and variability in output.
- We will refer to this tensor as U with width w , height h , and elements u_{ij} .

Let:

$$C = \alpha \cdot \text{BCE}(\hat{y}, y) + (1 - \alpha) \cdot \text{IoU}(\hat{y}, y)$$

$$W = C \cdot \exp(-U)$$

, and

$$R = \beta \cdot \frac{1}{w \times h} \sum_{i=1}^w \sum_{j=1}^h u_{ij}$$

The combined loss function used is

$$\text{Loss} = \text{mean}(W) + R$$

After this loss is calculated, it is backpropagated into SAM2. This updates the model's weights for improved accuracy of segmentation masks and is essential to finetuning the model for aligning its predictions with the ground truth of the dataset.

To ensure convergence, we repeat the above for 6000 steps, progressively refining the model's ability to accurately segment objects in self-driving scenarios.

The SAM2 model used was derived from the official Facebook Research's implementation [Ravi et al., 2024]. The model is fine-tuned using Python, specifically the Pytorch

framework, and is trained using a NVIDIA GeForce 4060 GPU.

B. UAT Adapter with SAM for Extreme Weather Conditions

Our second complementary approach was to tackle specific object instance segmentations in extreme weather scenarios utilizing the UAT-SAM adapter architecture by [Jiang et al., 2024]. As referenced in section II, the UAT adapter is a novel addition to the original SAM architecture, inspired by methodologies in medical imaging. This adapter is inserted into each transformer block of SAM. It acts as a compact set of parameters that incorporates additional information—in this case, uncertainty. The UAT adapter utilizes the CMSM (Condition Modifies Sample Module) to incorporate a sampled uncertainty code, z , derived from a CVAE (Conditional Variational Autoencoder). This CVAE employs both a Prior Net (P) and a Posterior Net (Q) to encode observed uncertainty information from the input image. [Jiang et al., 2024]

Unlike previous approaches that directly concatenate the sampled code z with the main features, the UAT adapter takes a more refined approach. It integrates position vectors (p) and employs learnable attention-like mechanisms to transform z into meaningful features. These features are then combined with the main features in a layer-specific manner, allowing for nuanced modifications. This design ensures that the uncertainty sample from the CVAE is effectively captured and utilized, leading to more robust segmentation outputs.

Prior to training the CAMVID dataset required extensive pre-processing to be utilized. We applied a random weather filter either fog, rain, or snow in random filtering strengths form 0-1 (0=clear image , 1= completely obscured image) to the original images to introduce difficulty to the model when training and testing on obscured images due to extreme weather.

Due to the module's architecture and basis on medical image segmentation, it required multiple ground truth segmentations for every image. However a majority of publicly available driving datasets including CAMVID only provide 1 set of ground truth masks. To simulate the ambiguous segmentation requirements in our training data, we applied elastic deformations to all 1,419 human-segmented ground truth segmentations from the CAMVID dataset. Each segmentation was deformed by randomly shifting pixel locations in both the x and y directions using a Gaussian filter. The magnitude and smoothness of these shifts were controlled by two parameters: alpha and sigma. The parameter alpha controlled the strength of the deformation, while sigma determined the smoothness of the deformation, with higher values resulting in more gradual, blurry shifts. To introduce variability and simulate different environmental conditions, three different parameter sets were used: Fog-like deformations with alpha = 20.0 and sigma = 15.0 for smoother, more blurred boundaries; Rain-like deformations with alpha = 25.0 and sigma = 4.0 for sharper, more localized changes; and Snow-like deformations with alpha = 30.0 and sigma = 7.0 for medium smoothness with stronger distortion.

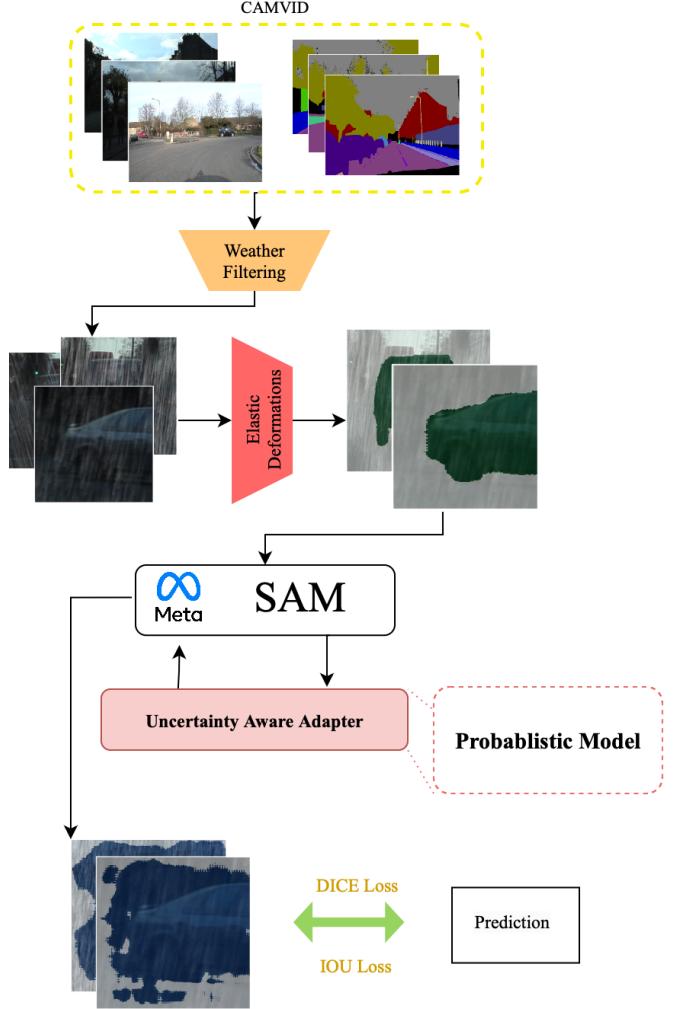


Fig. 2. Overview of UA-SAM Training and data augmentation pipeline.

This approach was used to generate three additional annotations from single ground truth, alongside one human annotation. Each image in the dataset, therefore, has four segmentation masks including ground truth, capturing a range of plausible interpretations. Each of the segmentations were matched to the original weather filtered images for training.

We also utilized instance cropping on the data to specifically focus on car segmentations to train, leveraging uncertainty modeling to prioritize regions with high variability. This adaptation allows the model to generate accurate outputs even with noisy or ambiguous data. The training methodology incorporates a tailored loss function, primarily the Dice Coefficient Loss, to handle segmentation. This loss function improves boundary detection, crucial for imbalanced datasets and difficult scenarios. The training pipeline follows a multi-stage process, starting with a pre-trained Segment Anything Model (SAM) and selective parameter freezing to retain SAM's pre-trained capabilities. Gradual adaptation fine-tunes the model for specific domain needs. Key metrics like Dice score and Intersection over Union (IoU) are monitored, with early stop-

TABLE I
AVERAGE IOU SCORES OF MULTISTEP FINETUNED SAM2

Model	Car	Truck	Person	Bicycle	Motorcycle	Traffic Light	Stop Sign	Fire Hydrant
Finetuned SAM2	0.156	0.187	0.230	0.247	0.117	0.300	0.185	0.119
Zero-shot SAM2	0.087	0.110	0.188	0.155	0.070	0.200	0.230	0.119

TABLE II
AVERAGE DICE COEFFICIENT SCORES OF MULTISTEP FINETUNED SAM2

Model	Car	Truck	Person	Bicycle	Motorcycle	Traffic Light	Stop Sign	Fire Hydrant
Finetuned SAM2	0.333	0.406	0.531	0.565	0.259	0.672	0.416	0.239
Zero-shot SAM2	0.142	0.200	0.428	0.333	0.138	0.475	0.543	0.239

ping to prevent overfitting. TensorBoard visualizes the training process, ensuring high performance and adaptability in severe conditions. We tested the finetuned model and zero-shot sam by running inference on 177 heavy weather filtered CAMVID car instance segmentations with the original ground truth segmentation paired, and compared IOU and DICE across both.

IV. RESULTS AND DISCUSSION

A. SAM2 Multistep Finetuning

1) *Overall Accuracy Improvements:* To assess the performance of finetuned SAM2, we evaluated it against base SAM2 using Intersection-Over-Union (IoU) and the DICE coefficient. Specifically, we found the average IoU scored by the models when segmenting commonly seen road objects (e.g. cars, people, bicycles, traffic lights) as well as when segmenting entire images across the Bdd100k and Camvid datasets. The following tables describe the results we found.

TABLE III
AVERAGE IOU SCORES OF MULTISTEP FINETUNED SAM2

Model	Overall - Bdd100k	Overall - Camvid
Finetuned SAM2	0.303	0.303
Zero-shot SAM2	0.246	0.246

TABLE IV
AVERAGE DICE COEFFICIENT SCORES OF MULTISTEP FINETUNED SAM2

Model	Overall - Bdd100k	Overall - Camvid
Finetuned SAM2	0.690	0.690
Zero-shot SAM2	0.550	0.550

Our fine-tuned SAM2 outperformed zero-shot SAM2 in most classes based on IoU and DICE scores, except for stop signs and fire hydrants. This may have been the result of a class imbalance in the Bdd100k dataset, which likely contains more examples of common road objects, such as cars, people, and motorcycles, than lesser seen objects, such as stop signs and fire hydrants. Additionally, the smaller size of stop signs and fire hydrants may have contributed to the reduced segmentation performance, especially when attempting to segment them at a distance. On average, our fine-tuned SAM2 model improved IoU by 36.13% over zero-shot SAM2, with the highest gain being in car segmentation (+79.13%) and the

smallest nonzero gain in person segmentation (22.34%). For DICE scores, our model improved by 48.79% on average, with cars showing the highest increase (+134.51%) and people showing the smallest nonzero increase (+24.07%).

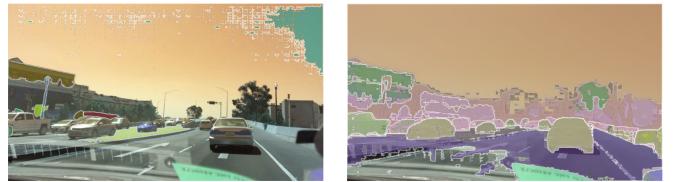


Fig. 3. Performance Before and After Finetuning. Note how zero-shot SAM2 fails to generate meaningful masks over most of the image.

2) *Uncertainty-Aware Finetuning Benefits::* Incorporating uncertainty into our finetuning process improved segmentation in ambiguous regions, particularly for multi-component objects like vehicles. Zero-shot SAM2 often produced inconsistent masks for vehicles, segmenting individual components (such as wheels or windows) or omitting the vehicle entirely (fig. 3). After applying uncertainty-aware finetuning, SAM2 consistently assigned a single mask per vehicle (fig. 3), enhancing segmentation accuracy and reducing fragmented outputs.

Additionally, our finetuned SAM2 model demonstrated strong generalization across diverse driving scenarios. We evaluated its performance on datasets from various environments, including Bdd100k (recorded in New York, San Francisco, and other regions), CamVid (recorded in Cambridge), and the GTA5 Driving Dataset (recorded in a simulated driving environment). Across all three datasets, our model consistently segmented key classes such as cars, trucks, roads, and pedestrians, highlighting its robustness in both real world and synthetic driving conditions.

B. UA-SAM

The fine tuned UAT-Adapter SAM was tested on 177 heavy weather filtered CAMVID car instance images with the original human segmentations serving as the ground truth. SAM served as the baseline, and once again the evaluation metrics were IOU and the DICE coefficient. When heavily obscured SAM often failed to segment or contour any object within the image and was halted by rain, and snow specifically.

Table V demonstrates the improvements that UAT Adapter SAM was able to make over zero-shot SAM in similar scenarios, highlighting its enhanced ability to handle complex segmentation tasks. UAT-SAM showed a 30% increase in the DICE coefficient and 42.7% increase in IOU scores. The UAT Adapter SAM consistently outperformed the zero-shot SAM by focusing on regions with high variability, improving segmentation accuracy in challenging environments where visibility is compromised.

TABLE V
AVERAGE IOU & DICE SCORES OF UA-SAM

Model	Dice Score	IOU
Zero-shot SAM	0.4809	0.3221
UA-SAM	0.6258	0.4598

Figure 4 shows an instance of a heavily filtered image, the ground truth segmentation and the lack of any segmentations on base SAM. Although UAT-SAM can be overconfident it still is able to generally localize the car.



Fig. 4. Example of instance segmentation on heavily filtered car image in rain scenario. From Left to Right, Top to Bottom: Filtered Original Image, Elastic Deformation GT Segmentation, Base SAM Segmentation, UA-SAM Segmentation

Despite Figures 5 and 6 demonstrating certain instance segmentations where UAT Adapter SAM and zero-shot SAM fail to segment effectively, generally, the UAT Adapter SAM exhibits better robustness and handles more challenging segmentation cases. These failures, while notable, are less frequent and often occur in particularly ambiguous or noisy regions, which further emphasizes the model's strength in most typical conditions. Utilizing the approach outlined by

[Jiang et al., 2024] to segment ambiguous segmentations in driving scenarios due to inclement weather showed considerable promise. This method enabled the model to focus on areas of uncertainty to segment cars, allowing for more reliable segmentation even under adverse weather conditions like fog, rain, or snow.

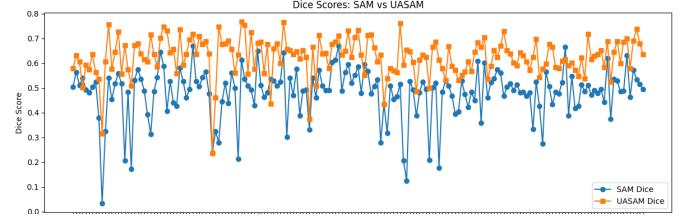


Fig. 5. DICE scores of Zero-shot SAM and UA-SAM across 177 car object patch segmentation in inclement weather

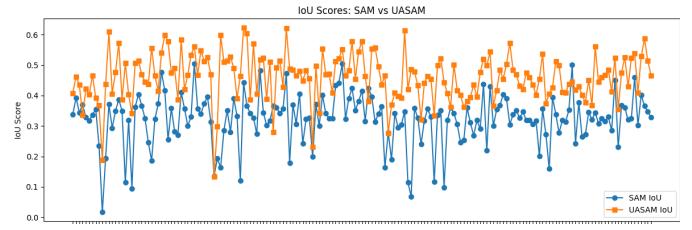


Fig. 6. IOU scores of Zero-shot SAM and UA-SAM across 177 car object patch segmentation in inclement weather

V. CONCLUSION

This research has demonstrated the effectiveness of two complementary uncertainty-aware approaches for improving semantic segmentation in self-driving applications, particularly under challenging weather conditions. The UAT adapter integrated with SAM successfully enhanced segmentation capabilities in severe weather scenarios by leveraging uncertainty estimates to identify and focus on critical regions where visibility is compromised. Our experiments on the BDD100K and CamVid datasets revealed that this approach significantly improved detection and segmentation of crucial road elements, with a particular focus on vehicles and specific objects of interest. The UAT adapter showed remarkable improvements in car detection accuracy under fog, heavy rain, and low-light conditions, where traditional segmentation methods typically fail.

In contrast, the uncertainty-incorporated multistep finetuning approach with SAM2 proved particularly effective at improving overall scene segmentation quality across varying weather conditions. This method delivered clearer contours and better distinction between foreground and background elements, resulting in more precise boundary delineation and improved class separation. The uncertainty-guided loss function enabled the model to adaptively focus on ambiguous

regions during training, leading to more reliable segmentations with well-calibrated confidence estimates.

Together, these approaches address different but complementary aspects of the inclement weather challenge in autonomous driving perception. The UAT adapter provides a targeted solution for the most severe conditions where safety-critical decisions must be made despite limited visibility, while the uncertainty-finetuned SAM2 offers broader improvements in segmentation quality that enhance overall system performance.

Our contributions not only advance the state of the art in semantic segmentation for challenging conditions but also demonstrate the value of incorporating uncertainty awareness into modern foundation models like SAM and SAM2. The methods presented here have potential applications beyond autonomous driving, particularly in other safety-critical domains where perception systems must operate reliably despite environmental challenges.

VI. FUTURE WORK

We plan to extend the UAT adapter’s capabilities to segment a wider range of objects beyond cars, including pedestrians, cyclists, traffic signs, and other road users, providing a more comprehensive perception system for autonomous vehicles. Incorporating more scenarios and datasets to train the model given more computation may also show considerable promise in expanding the current performance.

Additionally, we will conduct specific finetuning of our finetuned models on diverse weather scenarios to further improve performance across different environmental conditions. This weather-specific finetuning will target particular challenges such as snow accumulation, sun glare, and night-time lighting, allowing the system to better adapt to seasonal and temporal variations in driving conditions.

VII. ACKNOWLEDGEMENTS

We would like to express our sincere gratitude to the authors of Uncertainty-Aware Adapter: Adapting Segment Anything Model (SAM) for Ambiguous Medical Image Segmentation [Jiang et al., 2024] for their invaluable contributions to the field. Their research and application of uncertainty-aware adapters provided critical insights that significantly influenced our work.

We also deeply appreciate the open-source efforts and datasets made available by the research community, which played a crucial role in facilitating our experimentation and validation. The collaborative spirit of the academic and engineering communities has been instrumental in shaping our study.

REFERENCES

- [Brostow et al., 2009] Brostow, G. J., Fauqueur, J., and Cipolla, R. (2009). Semantic object classes in video: A high-definition ground truth database. *Pattern Recognition Letters*, 30(2):88–97. Video-based Object and Event Analysis.
- [Burnett et al., 2023] Burnett, K., Yoon, D. J., Wu, Y., Li, A. Z., Zhang, H., Lu, S., Qian, J., Tseng, W.-K., Lambert, A., Leung, K. Y. K., Schoellig, A. P., and Barfoot, T. D. (2023). Boreas: A multi-season autonomous driving dataset.
- [Cheng et al., 2023] Cheng, J., Ye, J., Deng, Z., Chen, J., Li, T., Wang, H., Su, Y., Huang, Z., Chen, J., Jiang, L., Sun, H., He, J., Zhang, S., Zhu, M., and Qiao, Y. (2023). Sam-med2d.
- [Dawood et al., 2023] Dawood, T., Chan, E., Razavi, R., King, A. P., and Puyol-Anton, E. (2023). Addressing deep learning model calibration using evidential neural networks and uncertainty-aware training.
- [Dutta et al., 2023] Dutta, S., Wei, H., van der Laan, L., and Alaa, A. M. (2023). Estimating uncertainty in multimodal foundation models using public internet data.
- [Jiang et al., 2024] Jiang, M., Zhou, J., Wu, J., Wang, T., Jin, Y., and Xu, M. (2024). Uncertainty-aware adapter: Adapting segment anything model (sam) for ambiguous medical image segmentation.
- [Jiang et al., 2022] Jiang, S., Guo, Z., Zhao, S., Wang, H., and Jing, W. (2022). Ce-gan : A camera image enhancement generative adversarial network for autonomous driving.
- [Kenadall and Gal, 2017] Kenadall, A. and Gal, Y. (2017). What uncertainties do we need in bayesian deep learning for computer vision? *NIPS 2017*.
- [Kirillov et al., 2023] Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A. C., Lo, W.-Y., Dollár, P., and Girshick, R. (2023). Segment anything.
- [Kohl et al., 2019] Kohl, S. A. A., Romera-Paredes, B., Meyer, C., Fauw, J. D., Ledsam, J. R., Maier-Hein, K. H., Eslami, S. M. A., Rezende, D. J., and Ronneberger, O. (2019). A probabilistic u-net for segmentation of ambiguous images.
- [Modas et al., 2020] Modas, A., Sanchez-Matilla, R., Frossard, P., and Cavallaro, A. (2020). Toward robust sensing for autonomous vehicles: An adversarial perspective. *IEEE Signal Processing Magazine*, 37(4):14–23.
- [Ravi et al., 2024] Ravi, N., Gabeur, V., Hu, Y.-T., Hu, R., Ryali, C., Ma, T., Khedr, H., Rädlé, R., Rolland, C., Gustafson, L., Mintun, E., Pan, J., Alwala, K. V., Carion, N., Wu, C.-Y., Girshick, R., Dollár, P., and Feichtenhofer, C. (2024). Sam 2: Segment anything in images and videos.
- [Wang et al., 2020] Wang, Z., Wu, Y., and Niu, Q. (2020). Multi-sensor fusion in automated driving: A survey. *IEEE Access*, 8:2847–2868.
- [Yang et al., 2024] Yang, C.-Y., Huang, H.-W., Chai, W., Jiang, Z., and Hwang, J.-N. (2024). Samurai: Adapting segment anything model for zero-shot visual tracking with motion-aware memory.
- [Yang et al., 2023] Yang, J. C. Z., , and Zhang, L. (2023). Semantic segment anything.
- [Yu et al., 2020] Yu, F., Chen, H., Wang, X., Xian, W., Chen, Y., Liu, F., Madhavan, V., and Darrell, T. (2020). Bdd100k: A diverse driving dataset for heterogeneous multitask learning.
- [Zhang et al., 2023] Zhang, Y., Carballo, A., Yang, H., and Takeda, K. (2023). Perception and sensing for autonomous vehicles under adverse weather conditions: A survey. *ISPRS Journal of Photogrammetry and Remote Sensing*, 196:146–177.

Evaluating Decision-Making Generalization in RAG Agent Architectures

Mehar Shienh

University of Waterloo

msshienh@uwaterloo.ca

Evan Dennison

University of Waterloo

edennison@uwaterloo.ca

Jordan Leis

University of Waterloo

j2leis@uwaterloo.ca

Devon Kisob

University of Waterloo

dkisob@uwaterloo.ca

Jennifer Yu

University of Waterloo

j545yu@uwaterloo.ca

Yalda Nikookar

University of Waterloo

ynikooka@uwaterloo.ca

Madhav Malhotra

University of Waterloo

madhav.malhotra@uwaterloo.ca

Abstract—This paper explores LLMs as generalized decision-making assistants. We propose an assessment framework where retrieval-augmented generation (RAG) architectures are compared in simulated environments. By comparing objective win rates in games like Monopoly and Werewolf, we assess the efficacy of architectural options like reflection or multi-agent roles. This allows us to then apply the best performing architectures to the real-life context of political analysis. With this method, we find that the RAG architectures explored do not show generalization across decision-making contexts.

I. INTRODUCTION

Political decision-making is inherently complex, requiring the ability to navigate conflicting interests, ethical considerations, and long-term policy consequences. Unlike structured tasks such as Go, where AI has achieved superhuman performance through reinforcement learning [1], political decisions involve subjective judgments. Research has shown that AI struggles with strategic reasoning in multi-agent settings where human behaviour is unpredictable, as seen in attempts to apply AI to judicial decisions [2]. Moreover, political decision-making is constrained by legal frameworks and ethical concerns, making it difficult to define optimal strategies solely through data-driven approaches [3].

Despite advancements in AI applications for law, most existing models focus on legal text analysis, compliance automation, and case law retrieval rather than autonomous decision-making [4]. To develop AI capable of making informed political choices, a training environment must simulate the strategic negotiation and decision-making pressures inherent in politics. Simulated environments like Monopoly [5] and Werewolf [6] have been used in behavioural studies to model economic and social decision-making, making them useful for training AI in competitive and cooperative strategies. By engaging in these controlled simulations, AI agents can develop decision-making frameworks that incorporate long-term strategy, weighing uncertainties, and adaptability, all of which are key skills necessary for legislative reasoning.

Still, the generalization of learned strategies from games to real-world contexts remains challenging. Research in transfer learning has demonstrated that AI systems often struggle to

apply strategies across domains with different structures and reward functions [7]. Political decisions rarely have objective ‘win conditions’ like games do, complicating the transfer of game-derived strategies to legislative contexts. However, recent advances in meta-learning approaches have shown promising results in enabling AI to adapt learned strategies to novel tasks with limited additional training [8].

LLMs also present unique advantages for this generalization challenge. Unlike traditional reinforcement learning systems, LLMs trained on diverse corpora already possess broad knowledge about political systems, historical precedents, and ethical frameworks [9]. This background knowledge potentially enables them to contextualize strategies learned in simulated environments within appropriate political frameworks. Studies examining zero-shot and few-shot learning capabilities of LLMs suggest they can rapidly adapt to new decision contexts with minimal domain-specific examples [10]. This raises the question of whether LLMs can be effective decision-making assistants across generalized environments, from structured games to unstructured political analysis.

II. RELATED WORKS

Recent advancements in LLM-driven agent-based modelling have demonstrated the potential for simulating complex decision-making systems across social, economic, and legal domains. Prior research has explored the use of LLMs as autonomous agents, capable of interacting with dynamic environments, learning from experience, and optimizing decision strategies. For instance, several studies have explored the use of LLMs in economic simulations. [11] studied LLM-driven economic forecasting, demonstrating that GPT-4-based agents could simulate macroeconomic trends and follow real-world principles like the Phillips Curve. However, the study noted that LLM agents struggled with long-term reasoning capabilities. Likewise, [5] applied LLMs to negotiation games, noting their tendencies to not make optimal decisions from a game theoretic perspective. Still, structured prompting techniques showed improvements in rational decision-making strategy.

Furthermore, many works show the promise of improving decision-making through multi-agent systems. [12] applies

multi-agent simulations to examine how LLMs can simulate social media discourse on contentious topics like nuclear energy policy and gender discrimination. Its findings highlight how LLMs can replicate real-world sentiments, but also risk amplifying biases and polarization. Similarly, [13] developed COLA, a multi-agent stance detection system, where agents acted as linguistic, domain-specific, and social media experts to analyze public discourse. Works like [14] applied adversarial multi-agent legal reasoning, while [13] showed that structured debates lead to more robust decision-making. Similarly, [15] used two debating LLM agents to generate and refine arguments in cooperative problem-solving scenarios. More generally, [16] proposes a hierarchical language architecture, finding improved decision-making via delegating complex decisions to multiple sub-agents. These studies show that multi-agent systems reduce logical inconsistencies and lead to more structured decision-making than single-agent approaches.

Lastly, studies also explore the use of vector stores and other memory implementations to improve the performance of decision-making systems over time. [17] introduced self-reflection prompting, enabling models to review past decisions and self-correct over time, improving logical consistency and strategy formation. [18] carries out ablation studies to assess the importance of memory-based and reflection-based RAG agents in logical coherence. It finds that both components play a significant role in improving subjective impressions of believability and coherence. As seen, diverse memory implementations can contribute to decision-making assistants in subjective and objective environments.

Building on these findings, we evaluate multiple components of RAG agents like multi-agent systems, memories, and reflection. We evaluate these agents in objective and subjective environments, comparing win rates in games of Monopoly and Werewolf before applying the agents to generate arguments for a political bill. This bridges the gap between abstract game mechanics and political decision-making. By sequentially increasing the complexity and realism of these simulations, we investigate whether strategic reasoning skills transfer effectively from game environments to political contexts.

III. METHODOLOGY

In brief, we compare four RAG architectures by their win rates in two structured game environments: Monopoly and Werewolf. For illustration, the two best-performing architectures then analyze legislative texts, demonstrating subjective impressions on the quality of the analysis produced. This applies the architectures to increasingly complex decision-making contexts; Monopoly involves independent decision-making with structured rules. Werewolf introduces conversational decision-making with structured rules. Political legislative analysis presents the most complex scenario with no structured rules to guide decision-making.

The four RAG architectures investigate combinations of two agent environments with two memory approaches. Each architecture runs 50 games of Monopoly and 50 games of

Werewolf to determine average win rates. A random seed was used to ensure all games involved different initial conditions where necessary, like when determining dice rolls in Monopoly. Meta-prompting techniques, which have been found to induce more reasoned responses in multi-agent systems, standardize prompts across games where possible to promote a fair comparison [19]. This factorial design systematically evaluates the contribution of each component to decision-making performance.

TABLE I: Architectures Compared

Architecture	Agent roles	Memory Contains
Courtroom; Raw	Two lawyers; judge	Past raw outputs
Courtroom; Reflection	Two lawyers; judge	Past output summaries
Advisory; Raw	Fast mind; slow mind	Past raw outputs
Advisory; Reflection	Fast mind; slow mind	Past output summaries

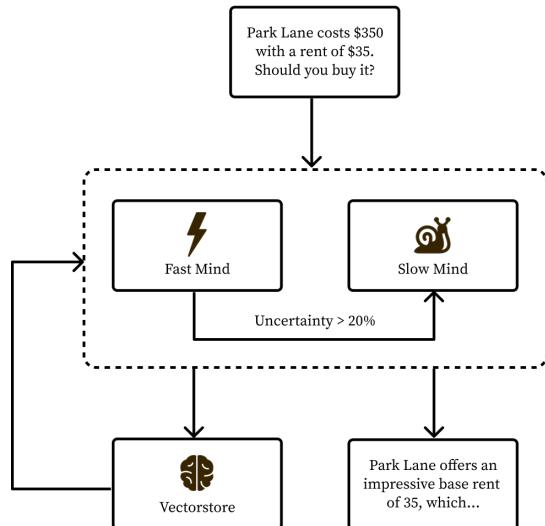


Fig. 1: Courtroom Architecture

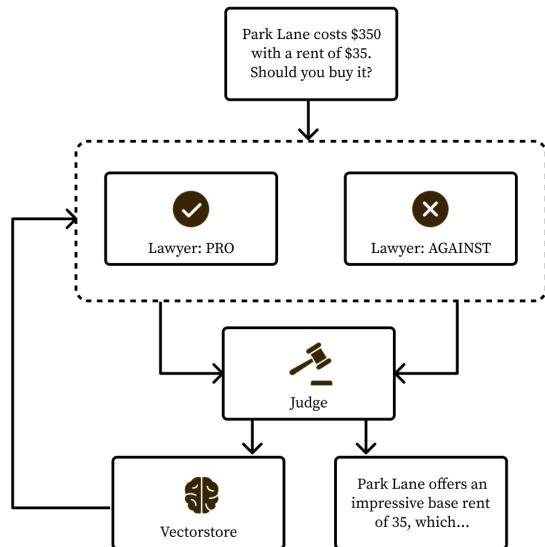


Fig. 2: Advisory Architecture

The courtroom multi-agent environment has two lawyer agents argue for different decisions, with a judge agent determining which argument is stronger. All agents use GPT-4o Mini. The second multi-agent environment follows an advisory model where a small, fast model (GPT-4o Mini; ‘fast mind’) makes most decisions and outputs an uncertainty score. This agent falls back to consulting a slower, larger model (GPT-4o; ‘slow mind’) when uncertainty exceeds a predefined threshold. Detailed prompts and hyperparameters are available in the supplementary material.

The raw memory vector store simply records past decisions and retrieves the two most similar decisions as examples for current decision-making. The reflective memory vector store takes a more sophisticated approach by storing and retrieving reflective summaries of multiple past decisions. Summaries are generated dynamically throughout the games by GPT-4o after a given number of turns. The embedding model used is Ada 02. Hyperparameters and prompts are detailed in the supplementary materials.

In Monopoly simulations, the custom agent implementing our architectures plays against a default player that follows a simple strategy of buying property whenever funds are available. Games conclude when one player depletes their funds or after 200 turns. The custom agent wins if its combined cash and mortgageable property value exceeds that of the default player. For Werewolf scenarios, the custom agent assumes the role of the werewolf and competes against default chatbots emulating the seer, witch, and villager roles. Detailed configurations for these default agents are provided in the supplementary materials. Games continue until either the werewolf is eliminated through voting or is the sole remaining player.

To evaluate performance in real-world contexts, we selected a random bill from the first session of the 44th Canadian Parliament for the subjective legislative analysis. Bill data was webscraped from openparliament.ca [20] and is available in supplementary materials. The full text of each bill was divided into 500-character chunks. The two best-performing RAG architectures analyzed each chunk’s implications on the overall decision to support or oppose the bill, mimicking the turn-based structure of the games. After analyzing all chunks, each architecture produced a final argument either supporting or opposing the bill.

These arguments were presented to 53 survey participants from a convenience sample. Participants ranked which argument they found more structured, balanced, compelling, and professional. The complete set of survey questions is available in the supplementary materials.

IV. RESULTS

We ran 50 games for each architecture, totalling 200 games of Monopoly and 200 games of Werewolf. This had a cost of approximately \$50 CAD in API credits, including testing runs before carrying out the final experiments. The average win rates from these are presented below.

TABLE II: Win Rates From Monopoly

Architecture	Memory	Win Rate
Courtroom	Raw	32 %
Courtroom	Reflection	26 %
Advisory	Raw	46 %
Advisory	Reflection	48 %

TABLE III: Win Rates From Werewolf

Architecture	Memory	Win Rate
Courtroom	Raw	18 %
Courtroom	Reflection	20 %
Advisory	Raw	20 %
Advisory	Reflection	20 %

The Advisory architectures outperformed the Courtroom counterparts, so we used them to provide arguments on Bill C-242. 53 undergraduate students at the University of Waterloo were surveyed on their subjective preferences regarding the results. Participants were asked to rank the architectures across four question categories:

- 1) Structure: “Which response did the best in presenting a structured argument for or against the bill?”
- 2) Balance: “Which response provides the most balanced discussion of multiple perspectives?”
- 3) Persuasion: “Which response is the most compelling? Select the one that would be most likely to sway your opinion.”
- 4) Decorum: “Which response is the most appropriate for parliament? Select the response that maintains the best formal and professional tone.”

These aggregated preferences are reported below.

TABLE IV: Survey Rankings

Architecture	Question Category & Preference			
	Structure	Balance	Persuasion	Decorum
Advisory: Raw	53%	38%	49%	51%
Advisory: Reflection	47%	62%	51%	49%

V. DISCUSSION

A. Limitations

There are numerous limitations in our findings. First, we consider flaws in our experiments in the objective decision-making environments. Assessing two games alone is not sufficient to claim generalization in decision-making ability, but was necessary to control costs in our study. Future work can improve upon this limitation by adding other objective decision-making tasks. For instance, games like Risk or Diplomacy with military or political themes may be viable candidates. In addition, forecasting tasks in political or stock prediction datasets may be suitable. Moreover, comparisons between different games are currently not always balanced. For instance, our default Monopoly player uses hard-coded rules to decide to always buy a property in Monopoly if funds are available, while the default player in Werewolf is a chatbot with non-deterministic behaviour. This decision was made since Werewolf required natural language conversation

between agents which made hard-coded default players seem unsuitable. Future work may wish to empirically evaluate these assumptions by using hard-coded or chatbot-based default players for both games. Lastly, there was only one opponent in Monopoly while there were three other default players in Werewolf due to the varying roles in the games. Future work may wish to balance the number of opponents across the games to ensure a similar level of difficulty, though this is likely to increase costs.

Carrying on, we consider flaws in our survey in the subjective decision-making arguments. The largest flaw in our survey is that convenience sampling was used to control study costs. However, this creates a very uniform demographic of participants with a similar background as undergraduates at the same university. With greater funding for marketing and participation compensation, a more diverse sample of participants across demographic strata would be feasible. It would be especially valuable in collecting responses from participants of varying ages, political affiliations, and familiarity with AI technology. In a similar vein, future works could be improved by a greater survey sample size backed by tests for statistical significance.

B. Conclusions

Our results indicate that the Advisory model consistently outperformed its Courtroom counterpart in both Monopoly, and to a lesser extent Werewolf. This outcome implies that a hierarchical structure, which leverages multiple levels of decision-making appears more suited for tasks involving clearly defined strategic goals. While the two-layered approach in Advisory agents may not fully capture all nuances—particularly in conversational environments like Werewolf—it nonetheless demonstrated greater adaptability in managing complex turn-by-turn decisions compared to the adversarial debate format used in Courtroom.

When extended to political scenarios, participants did not exhibit a strong preference for responses generated with raw versus reflective vector stores as preferences within the set categories for the two responses were relatively evenly split. This suggests that summarizing retrieved content may offer limited marginal benefit under these conditions.

Overall, the RAG architectures tested here did not show compelling evidence of generalizable decision-making competence across the diverse settings examined. Our study underscores the importance of domain-aware architectural choices and the need to develop more robust strategies for bridging the gap between controlled simulation environments and real policy discourse. Future work could explore more specialized architectural variants, such as introducing different foundational models or increasing test-time compute to allow sampling over multiple candidate responses. It may also be promising to explore fine-tuned LLMs or LLMs with reinforcement-learning techniques like reward modelling. In all, we present this work as a rough first demonstration to bridge models from objective tasks to subjective tasks, hoping that it may spark

future work to improve our methods and experimental results in this interdisciplinary topic.

C. Supplementary Materials

Supplementary materials are available online at: <https://github.com/Madhav-Malhotra/political-chatbot>

REFERENCES

- [1] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, Jan. 2016. [Online]. Available: <https://www.nature.com/articles/nature16961>
- [2] J. Kleinberg, H. Lakkaraju, J. Leskovec, J. Ludwig, and S. Mullainathan, "Human Decisions and Machine Predictions*", *The Quarterly Journal of Economics*, Aug. 2017. [Online]. Available: <http://academic.oup.com/qje/article/doi/10.1093/qje/qjx032/4095198/Human-Decisions-and-Machine-Predictions>
- [3] J. J. Bryson, "The Artificial Intelligence of the Ethics of Artificial Intelligence: An Introductory Overview for Law and Regulation," in *The Oxford Handbook of Ethics of AI*, M. D. Dubber, F. Pasquale, and S. Das, Eds. Oxford University Press, Jul. 2020, pp. 1–25. [Online]. Available: <https://academic.oup.com/edited-volume/34287/chapter/290654580>
- [4] A. Deroy, K. Ghosh, and S. Ghosh, "Applicability of large language models and generative models for legal case judgement summarization," *Artificial Intelligence and Law*, Jul. 2024. [Online]. Available: <https://link.springer.com/10.1007/s10506-024-09411-z>
- [5] W. Hua, O. Liu, L. Li, A. Amayuelas, J. Chen, L. Jiang, M. Jin, L. Fan, F. Sun, W. Wang, X. Wang, and Y. Zhang, "Game-theoretic LLM: Agent Workflow for Negotiation Games," 2024. [Online]. Available: <https://arxiv.org/abs/2411.05990>
- [6] Y. Xu, S. Wang, P. Li, F. Luo, X. Wang, W. Liu, and Y. Liu, "Exploring Large Language Models for Communication Games: An Empirical Study on Werewolf," 2023. [Online]. Available: <https://arxiv.org/abs/2309.04658>
- [7] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, "A Comprehensive Survey on Transfer Learning," *Proceedings of the IEEE*, vol. 109, no. 1, pp. 43–76, Jan. 2021. [Online]. Available: <https://ieeexplore.ieee.org/document/9134370/>
- [8] J. X. Wang, Z. Kurth-Nelson, D. Tirumala, H. Soyer, J. Z. Leibo, R. Munos, C. Blundell, D. Kumaran, and M. Botvinick, "Learning to reinforcement learn," 2016. [Online]. Available: <https://arxiv.org/abs/1611.05763>
- [9] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language Models are Few-Shot Learners," 2020. [Online]. Available: <https://arxiv.org/abs/2005.14165>
- [10] J. Wei, Y. Tay, R. Bommasani, C. Raffel, B. Zoph, S. Borgeaud, D. Yogatama, M. Bosma, D. Zhou, D. Metzler, E. H. Chi, T. Hashimoto, O. Vinyals, P. Liang, J. Dean, and W. Fedus, "Emergent Abilities of Large Language Models," 2022. [Online]. Available: <https://arxiv.org/abs/2206.07682>
- [11] N. Li, C. Gao, M. Li, Y. Li, and Q. Liao, "EconAgent: Large Language Model-Empowered Agents for Simulating Macroeconomic Activities," 2023. [Online]. Available: <https://arxiv.org/abs/2310.10436>
- [12] C. Gao, X. Lan, Z. Lu, J. Mao, J. Piao, H. Wang, D. Jin, and Y. Li, "S3: Social-network Simulation System with Large Language Model-Empowered Agents," 2023. [Online]. Available: <https://arxiv.org/abs/2307.14984>
- [13] X. Lan, C. Gao, D. Jin, and Y. Li, "Stance Detection with Collaborative Role-Infused LLM-Based Agents," 2023. [Online]. Available: <https://arxiv.org/abs/2310.10467>
- [14] G. Chen, L. Fan, Z. Gong, N. Xie, Z. Li, Z. Liu, C. Li, Q. Qu, S. Ni, and M. Yang, "AgentCourt: Simulating Court with Adversarial Evolvable Lawyer Agents," 2024. [Online]. Available: <https://arxiv.org/abs/2408.08089>
- [15] G. Li, H. A. A. K. Hammoud, H. Itani, D. Khizbulin, and B. Ghanem, "CAMEL: Communicative Agents for "Mind" Exploration of Large Language Model Society," 2023. [Online]. Available: <https://arxiv.org/abs/2303.17760>
- [16] J. Liu, C. Yu, J. Gao, Y. Xie, Q. Liao, Y. Wu, and Y. Wang, "LLM-Powered Hierarchical Language Agent for Real-time Human-AI Coordination," 2023. [Online]. Available: <https://arxiv.org/abs/2312.15224>
- [17] N. Shinn, F. Cassano, E. Berman, A. Gopinath, K. Narasimhan, and S. Yao, "Reflexion: Language Agents with Verbal Reinforcement Learning," 2023. [Online]. Available: <https://arxiv.org/abs/2303.11366>
- [18] J. S. Park, J. C. O'Brien, C. J. Cai, M. R. Morris, P. Liang, and M. S. Bernstein, "Generative Agents: Interactive Simulacra of Human Behavior," 2023. [Online]. Available: <https://arxiv.org/abs/2304.03442>
- [19] S. Hong, M. Zhuge, J. Chen, X. Zheng, Y. Cheng, C. Zhang, J. Wang, Z. Wang, S. K. S. Yau, Z. Lin, L. Zhou, C. Ran, L. Xiao, C. Wu, and J. Schmidhuber, "MetaGPT: Meta Programming for A Multi-Agent Collaborative Framework," 2023. [Online]. Available: <https://arxiv.org/abs/2308.00352>
- [20] M. Mulley, "Open Parliament API." [Online]. Available: <https://openparliament.ca/api/>

Exploring the Ethical Implications of Using AI-Based Software for MRI Diagnosis in Clinical Settings

Komal Azeem
Queen's University
komal.azeem@queensu.ca

Ricky Leigh
Queen's University
21rjl13@queensu.ca

Rebecca Krichker
Queen's University
20rck4@queensu.ca

Sydney Basil
Queen's University
20sb85@queensu.ca

Cecile Woo
Queen's University
21ysw@queensu.ca

Abstract—The increasing usage of artificial intelligence in MRI disease classification and diagnosis presents several ethical implications related to patient privacy, data security, and responsible use. This paper will review some current use cases of AI-based MRI image classification models and propose a framework for ethics policymakers and medical information officers to ensure patient safety and responsible usage of AI in clinical settings.

I. INTRODUCTION

The increasing popularity of Artificial Intelligence (AI) has led to its growing usage in the healthcare industry. The most common use cases range from improving patient interactions to helping physicians in their diagnosis [1]. An area of interest for clinicians has been the upcoming use of image-based detection models that assists in disease diagnosis. These machine learning algorithms use patient data to train and test the model and eventually outline areas of concern [2]. Given this new and emerging application, it is increasingly important for healthcare policy makers as well as private hospitals to understand the ethical implications associated with its use. This paper will delve into some current use cases of AI for MRI classification and disease diagnosis. It will also present a framework that policymakers and medical information officers should consult when assessing the ethical validity of an AI-based service. This paper will address the ethical implications of AI usage in MRI technology for diagnostic purposes through three main phases, outlined in Figure 1. The initial section of this paper will provide a review of the ongoing use cases of AI in MRI technology, providing a background on the datasets, model accuracy, and overt ethical implications. The second phase of this paper will address these specific ethical issues and will serve as a basis for producing a guideline for policy makers and medical information officers who are looking to implement the model for clinical use. The final phase will assess the validity and necessity of such a guideline.

II. RELATED WORKS

Existing guidelines have addressed the various ethical implications of using AI in healthcare settings, but the broad nature of these guidelines have made it difficult to apply to varying situations. The guideline developed by Bouderhem [3] is one of the most prominent guideline pertaining to the ethical use of AI in healthcare. Bourderhem [3] outlines a wide range of use cases of AI and analyzes the current ethical challenges and provides various recommendations to address these issues. Unfortunately, due to its broad scope, it fails to apply itself to more specific use cases. As a result, we have chosen to address one category of use cases and cater a comprehensive guideline that outlines major weaknesses and ethical concerns.

III. REVIEWING CURRENT USE CASES AND LITERATURE

A. Diagnosing Alzheimer's Disease Through CNN-Based MRI Detection

Jain et al. [4] explored the use of Convolutional Neural Networks (CNNs) and transfer learning to develop an image classification model that could aid physicians in the early diagnosis of Alzheimer's Disease. Since the training of a CNN typically requires a large database of information, transfer learning is utilized to easily facilitate the process of developing a new model. Transfer learning is the use of one model's output to train another model [5]. Currently, MRI is the most common method of detecting early deterioration as it provides clear anatomical abnormalities, which is strongly linked to the development of Mild Cognitive Impairment [6]. As a result, the classification model categorized MRI results as either Mild Cognitive Impairment (MCI), Cognitively Normal (CN), and Alzheimer's Disease (AD). These classifications were then used to identify any neural degeneration early on and guided the physician's treatment plan.

This paper [4] identifies how a pre-existing CNN, VGG-16, that was originally trained on data from ImageNet was used to facilitate the production of a new 3-way classification CNN for

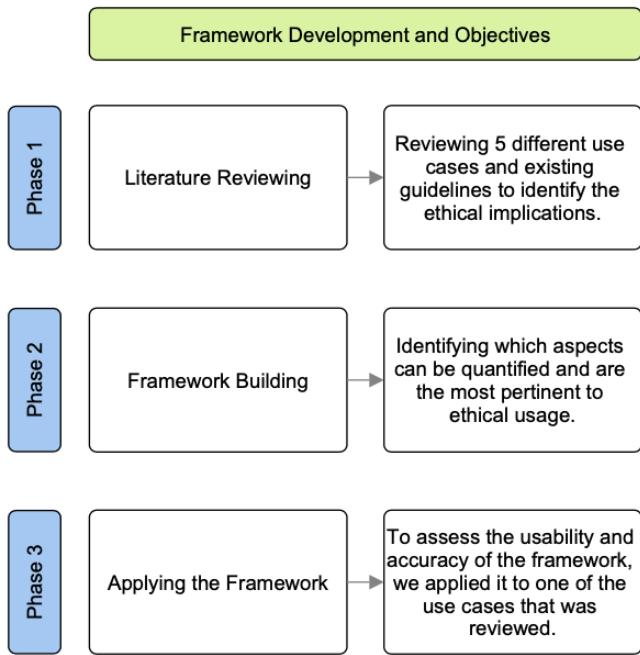


Fig. 1. Flowchart of the phases and objectives of this paper.

MRI images. The resultant precision of their developed validation set was 95.73%. For binary classifications, the model accuracy for AD vs MCI was 99.30%, 99.14% for AD vs CN, and 99.22% for MCI vs CN. Further analysis of the 3-way model using confusion matrices revealed a precision, recall, and F1 score for each condition: the AD condition produced values of 1, 0.91, and 0.95, the CN condition produced values of 0.99, 0.97, 0.98, and the MCI produced values of 0.90, 1, and 0.94, respectively.

Due to the covert nature of the initial dataset and its use of transfer learning, inferring the level of diversity involved in the dataset is very difficult. Any bias present in the initial dataset from ImageNet poses a great risk to the developed CNN, as it will be reflected in the results. A lack of diversity in patient representativeness could result in an overfitting of specific features, leading to inaccurate classification and reported accuracies, which is a large risk for potential clinical usage; racial, sexual, and clinical diversity is an important consideration when assessing the accuracy of the model and the following clinical implications. Also, due to the black box nature of CNN's, the decision-making strategies employed by the CNN are not fully transparent, and can only be altered to a certain extent. Lastly, there is a question of cost of development, implementation, and access; when determining for who and how this technology is implemented, clinics should consider which patients will have access to this technology, and if it provides them with a greater advantage than an individual who is unable to afford it. There remains a question of affordability and insurance coverage for the model usage, as different countries have varying policies on the type of

technology and services available to the patient.

B. Using AI in MRI Classification of Liver Tumors

Zhen et al.'s [7] primary objective was to develop a deep learning system (DLS) for accurate liver tumor diagnosis using MRI and clinical data. The model addressed the limitations of current diagnostic methods, which are often subjective and rely on the radiologist's experience. The DLS was created to classify liver tumors into seven categories, differentiate between benign and malignant tumors using unenhanced MRI, and further classify malignant tumors by integrating unenhanced images and clinical data. Ultimately, the researchers aimed to provide a more efficient, accessible, and accurate diagnostic tool that could potentially reduce the need for contrast agents and the associated costs and side effects.

Zhen et al. [7] used convolutional neural networks (CNNs), specifically the Google Inception-ResNet V2 architecture, which was pre-trained on a large image dataset and subsequently fine-tuned using the study's liver tumor MRI data. The dataset included 31,608 MRI images from 1,210 patients for training, and 6,816 images from 201 patients for validation. The seven-way classifier, which utilized six MRI sequences, achieved area under ROC curve (AUC) values ranging from 0.897 to 0.987, with sensitivity between 53.3% and 100% and specificity between 91.6% and 99.5%. The binary classifier, using unenhanced sequences, reached an AUC of 0.946, indicating its ability to distinguish malignant from benign tumors with accuracy comparable to that of a classifier based on enhanced sequences (AUC of 0.951). The three-way malignancy classifier, which integrated unenhanced images and clinical data, demonstrated significantly enhanced AUCs (ranging from 0.963 to 0.998) compared to models that relied solely on enhanced images for similar classifications, closely matching the radiologists' performance.

The study received approval from an independent institutional review board (IRB) at Sir Run Run Shaw Hospital, China, however there were concerns regarding the robustness of Chinese IRBs. Factors such as lack of thorough review processes and insufficient oversight may have compromised the ethical review capacity of the study [8]. Although written informed consent was not mandated in accordance with local laws, this approach is different from other countries' standards. This discrepancy highlights the variability in ethical standards globally, raising issues of patient privacy and the generalizability of research. While the study employed saliency maps to enhance interpretability, it acknowledges the lack of full transparency regarding the AI models' decision-making processes. Lastly, the study relied on patient medical records and MRI images, with limited discussion on specific security measures or anonymization protocols.

C. Utilizing Machine Learning in MRI Technology to Diagnose Schizophrenia

The paper by Sadeghi et al. [9] reviewed and evaluated the applications of artificial intelligence (AI), specifically machine learning (ML) and deep learning (DL), in the diagnosis

of schizophrenia using magnetic resonance imaging (MRI). Schizophrenia is a complex psychiatric disorder that poses significant challenges to the accurate diagnosis due to the heterogeneity of the symptoms and the absence of definitive biomarkers [10]. Sadeghi et al. [9] explored AI-based computer-aided diagnostic systems (CADs) designed to automate the process of diagnosis by using structural MRI (sMRI) and functional MRI (fMRI) datasets. The paper emphasizes the integration of AI into clinical workflows to assist healthcare professionals. It also provides a summary of the advancements of AI methods, specifically their performance, challenges, and potential improvements for diagnosing schizophrenia.

The study used various AI models, including conventional ML methods (e.g. Support Vector Machines [SVM], Random Forest) and advanced DL architectures (e.g. CNNs, Autoencoders) [9]. The reported accuracies varied widely depending on the dataset, preprocessing techniques, and feature selection methods. For example, ML models like SVM showed up to 94% accuracy with carefully extracted features [9]. At the same time, DL methods, such as 3d-CNNs, achieved comparable or better results through direct analyses of raw data. The models leveraged features such as gray matter volume, connectivity matrices, and task-based fMRI signal data [9]. However, the performance consistency was impacted due to challenges such as dataset imbalance, small sample sizes, and MRI image noise. The paper concluded that DL's automatic feature extraction was more advantageous than the ML's reliance on manual feature engineering but acknowledged the computational intensity required as a trade-off.

Although the paper did not explicitly discuss the ethical implications of AI-based schizophrenia diagnosis, its findings raise potential concerns. One such implication is that AI models trained on neuroimaging data may overemphasize structural and functional brain abnormalities at the expense of behavioural and functional impairments, which are critical for DSM-5 diagnosis [11]. Hence, this could lead to biases in clinical decision-making if AI tools are not used with comprehensive psychiatric evaluations. Furthermore, while publicly available datasets (e.g. COBRE, SchizConnect) enhance accessibility, limitations in sample diversity and generalizability could lead to the misrepresentation of schizophrenia phenotypes in AI-based diagnostic tools. A final challenge is the lack of standardized evaluation metrics and transparency in model decisions, which can hinder replicability and trust in clinical settings. Future research should focus on integrating multi-modal assessments (e.g., neuroimaging and behavior markers), improving or adding data governance frameworks, and ensuring that AI models are interpretable and clinically relevant.

D. Detecting Brain Tumors Using Convolutional Neural Networks

Research conducted by Rahman et al. [12] leveraged artificial intelligence, specifically the EfficientNetB2 deep learning architecture, to detect patterns indicative of brain tumors in MRI scans. By improving MRI image quality through

preprocessing techniques such as cropping, equalization, and homomorphic filtering, the research aimed to enhance the accuracy, efficiency, and consistency of tumor detection. Using publicly available datasets featuring diverse MRI images of individuals with and without brain tumors, the study sought to create a tool that could assist, rather than replace, physicians. This AI-driven methodology is designed to provide rapid, precise, and reliable tumor detection, reducing diagnostic delays and mitigating variability caused by human interpretation. This study used the EfficientNetB2 deep learning architecture, a convolutional neural network model known for its efficiency and scalability in image classification tasks [12]. Fine-tuned for brain tumor detection, the model was trained on three publicly available datasets, achieving high validation accuracies of 99.83% on the BD-BrainTumor dataset, 99.75% on the Brain-tumor-detection dataset, and 99.2% on the Brain-MRI-images-for-brain-tumor-detection dataset. While these results indicate strong performance, real-world clinical settings introduce variability that may impact accuracy, such as differences in MRI protocols, scanner types, and patient demographics. Thus, external validation on diverse, real-world datasets is necessary to assess generalizability. The architecture leverages a balanced scaling approach to optimize depth, width, and resolution, enabling it to capture intricate patterns in MRI scans. Preprocessing techniques, such as cropping, equalization, and homomorphic filtering, further enhanced the input data quality, boosting the model's ability to identify tumor regions accurately. However, despite its strong performance, the model is intended as a decision-support tool rather than a replacement for clinical expertise. Further studies on model interpretability and robustness in real-world environments will be critical for ensuring safe and effective deployment in healthcare settings. Publicly available datasets from Kaggle, including BD-BrainTumor, Brain-tumor-detection, and Brain-MRI-images-for-brain-tumor-detection were used for the study [12]. While these datasets are anonymized and promote accessibility, the lack of information about patient consent during data collection raises ethical concerns. In addition, no explicit security measures, such as encryption or secure storage, were mentioned, which could pose risks in clinical applications requiring compliance with privacy regulations like HIPAA. The datasets may also be biased, potentially limiting the model's generalizability to diverse populations, tumor types, and imaging conditions. The absence of real-world clinical validation further increases this concern. Additionally, the study does not address the explainability of the model, which is critical for building trust in AI-driven diagnoses. While intended to assist physicians, over-reliance on such models without sufficient human oversight could be problematic. Addressing these issues is crucial for the ethical and effective integration of an AI model such as this one in clinical healthcare.

E. Using Deep Learning Models For Early-Stage Breast Cancer Screening

Breast cancer is the second leading cause of cancer-related deaths among women [13]. This has prompted researchers and

healthcare systems to implement large-scale mammography screening programs aimed at early detection to improve outcomes [14]. For instance, the USA performs approximately 43 million mammograms annually [5]. However, the effectiveness of these screenings is hindered by variability in cancer detection professionals' accuracy [15]. Additionally, a shortage of mammography specialists worldwide limits the availability and scalability of these programs [14]. To address these challenges, McKinney et al. [16] introduced an artificial intelligence (AI) system, specifically a deep learning model, designed to screen mammograms for early-stage breast cancer, enhancing both the accuracy and scalability of breast cancer detection.

The AI system for breast cancer screening consisted of an ensemble of three deep learning models: lesions, breast and full case. The lesion model identified the ten most specific regions of suspicion in mammogram images of suggestive cancer using a model called RetinaNet2. Each extracted region was passed through a feature detector called MobileNetV23. Malignancy predictions were then produced for each region and combined into a composite score. The breast model processed images of each breast independently. The model used a ResNet-v2-50 network (a type of CNN) as an Image Feature Extractor. Each breast had two views of a mammogram, which were concatenated and passed through an additional neural network to predict a cancer score for each breast. The cancer scores for the right breast and left breast were then compared and the maximum score between the two breasts were taken as the case-level score. Finally, the case model considered the complete set of mammogram views; the model uses a ResNet-v1-50 network as a feature extractor. The complete set of mammogram views contained four images, which were concatenated. The concatenated vector was passed through a hidden layer used for binary classification to determine whether the patient (case) had cancer (yes or no).

All models were trained with data augmentation applied to each image. Each model generated a cancer risk score between 0 and 1, with the final score being the mean of the three models' predictions. The model was trained using datasets from two UK screening centers and one US center, representing both populations. The UK dataset included 25,856 women, while the US dataset contained 3,097 women. McKinney et al. [16] evaluated the model's accuracy through three approaches. First, they compared AI predictions with historical clinical decisions, finding that the AI model demonstrated higher specificity and improved sensitivity compared to both UK first readers and US single readers in radiology practice. The model's performance on the UK dataset had an AUC of 0.996, while on the US dataset, the AUC was 0.883. Second, they conducted cross-cultural testing by applying the UK-trained model to the US dataset, which showed improved specificity and sensitivity compared to radiologists. Here, the AI model achieved an AUC of 0.889 on the US dataset. Finally, they compared the AI system's performance against six US board-certified radiologists interpreting 500 challenging US cases. In this comparison, the AI system significantly outperformed the

radiologists' average performance, with the mean radiologist reading AUC being 0.75, whereas the AI system achieved an AUC of 0.871.

One limitation highlighted by McKinney et al. [16] was dataset representativeness. While the UK dataset mirrored the nationwide screening population, the US dataset came from a single screening center. For this AI system to achieve its potential for scalability and accessibility, the datasets need to be truly representative of diverse populations. This issue is reflected in the system performance: AUC values were highest in the UK where it was developed but decreased when trained on US data. Although this decline in performance was minimal, it raises concerns about the system's effectiveness across different populations, particularly given that performance variations were observed even between the demographically similar UK and US.

IV. DEVELOPING A FRAMEWORK

Upon conducting a thorough review of the varying use cases of AI models in MRI technology and diagnosis, the obvious, yet unique, ethical concerns became clear. In order to develop a successful and practical framework for policymakers, it was essential to grasp the common underlying issues in each use case. It became abundantly clear that patient privacy, data collection, interpretation of results, and responsible use were critical to the ethical use of each of the models. As a result, the proposed guideline, provided in the Appendix, presents a graded rubric system that allows developers and future executors to filter their models through an ethical framework to determine the strengths of their models.

A. Criteria Justification

In recent years, significant progress has been made towards regulating AI in healthcare through scientific reviews and policy initiatives [3], [17], [18]. We drew upon these efforts to establish our specific guidelines. In particular, Bouderhem's [3] article provides a comprehensive analysis of the technical, ethical and regulatory challenges related to the application of AI in healthcare. We specifically chose to closely examine Bouderhem's [3] article given its publication recency and thorough examination of AI's prevalence, opportunities, challenges and risks. Bouderhem [3] outlines a broad range of AI applications, including care management, drug discovery, medical imaging analysis and more. Bouderhem's [3] analysis thoroughly examines global policies, such as the United States' General Data Protection Regulation (GDPR), the United States' Health Insurance Portability and Accountability Act (HIPAA) and the European AI Act proposal. This article concludes itself by calling on the WHO to strengthen its regulatory role in AI-driven healthcare, arguing that current legal frameworks are insufficient.

Bourdenhem's message is important and his background research is rigorous, however, the broad scope and general recommendations create ambiguity about implementation [3]. To build upon their work, a more specific guideline for policymakers and healthcare institutions was developed to regulate

the AI applications in healthcare. One specific and pertinent application of AI in healthcare was chosen as the basis of this guideline: the role of AI in MRI detection. Bouderhem's [3] article notes that policies like the AI Act proposal struggle with defining AI broadly. This makes it difficult to provide clear ethical and regulatory recommendations. Narrowing the focus to AI in MRI detection specifically will ensure our criteria is specific and practical for future implementation.

The usage of AI in MRI detection was chosen due to its promising potential across a wide range of clinical applications, including cancer detection, cardiac imaging, and musculoskeletal assessment [19]. MRI technology is also highly versatile, benefiting patients across all demographics, including adolescents, children, and older adults [19]. Despite its advantages, traditional MRI technology has notable limitations, such as long scan times and high sensitivity to patient movement, which can lead to blurred images and increased costs [19]. However, AI-driven advancements have demonstrated the ability to accelerate scans, enhance image quality, and reduce expenses [19]. Given the notable potential of AI in MRI, it is crucial to address its challenges and establish clear, rigorous standards. Implementing well-defined technical and ethical guidelines will ensure responsible adoption in healthcare institutions, maximizing its benefits while maintaining patient safety and data integrity.

The goal of this paper is to establish clear guidelines with numbered criteria (1-4) to systematically evaluate whether an AI model meant for MRI technology meets technical and ethical standards. Building on the challenges outlined by Bouderhem [3], three key categories were developed with subsequent criteria for the responsible implementation of AI-based MRI technology in healthcare institutions: patient privacy and security, data collection, and interpretation and responsibility.

First, a section on patient privacy and security addresses the risks posed by patient information being shared or stolen [3]. This criteria emphasizes that patient data used for AI training should be encrypted and anonymized, with strict mechanisms in place to prevent unauthorized access, data breaches, or re-identification risks. In addition, patients should have clear and accessible options to give informed consent for data collection, processing, and sharing, with the ability to opt out if they choose. Furthermore, AI systems should comply with legal and ethical data protection standards such as GDPR and HIPAA to uphold patient privacy.

Second, a section on data collection addresses how biased algorithms can result in discrimination and inaccurate predictions [3]. This section emphasizes the importance of diverse and unbiased datasets to prevent discrimination, particularly against vulnerable populations. AI models should incorporate MRI data from underrepresented cases to improve diagnostic precision across all groups of patients. Healthcare providers and patients should also be fully informed about how AI models are trained, including potential biases or data gaps. In addition, continuous monitoring must be implemented to identify and mitigate biases in real-world applications, ensur-

ing that AI models remain equitable and effective.

Finally, a section on interpretation and responsibility addresses the lack of performance indicators in AI system. These are metrics that healthcare providers need to detect errors and biases that could have legal implications such as medical malpractice liability [3]. Our criteria focus on AI's role as a supportive tool rather than a replacement for physician judgment. The model should enhance, rather than dictate, clinical decisions, with physicians fully trained to interpret and verify AI-generated outputs. Transparency is essential, meaning that AI models should provide explainable results, confidence scores, and clearly defined limitations to prevent over reliance on automated assessments. Finally, AI-generated insights should be integrated into clinical workflows, ensuring that physicians can track and evaluate their influence on final diagnostic decisions.

B. Utilizing the Scoring System

The advancement of AI-driven healthcare models presents significant ethical and regulatory challenges, highlighting the need for a structured and objective grading system. A well-defined framework is essential to ensure transparency and consistency while evaluating AI compliance with patient privacy and ethical considerations. Without standardizing the grading system, assessing the ethical nature of the AI models may become subjective, resulting in reduced trust in the AI applications in healthcare. The proposed grading system provides a uniform approach to measure compliance with ethical standards and technical effectiveness, allowing healthcare providers, policy makers, AI developers, and patients to make informed decisions based on measurable criteria rather than arbitrary judgment.

The grading system utilizes a four-tier rubric, assigning numerical values from 1 to 4 based on compliance and performance levels. This structured approach ensures that evaluations remain comparable across AI models, and aligned with industry best practices and regulatory requirements. By adhering to fundamental legal and ethical standards, the grading system minimizes ambiguity and provides a precise evaluation of an AI model's adherence to ethical and technical benchmarks.

1) *Considerations for Level Selection:* The selection of each level within the grading system was guided by considering compliance, effort, and ethical responsibility. The grading structure follows a hierarchical pattern where Level 1 represents a complete lack of necessary safeguards, making the AI system unreliable. Level 2 reflects basic but inconsistent measures, highlighting the need for further refinement. Level 3 shows a strong adherence to ethical and technical standards, with only minor area that needed improvement. Finally, Level 4 reflects full compliance, where AI systems not only meet but exceed industry standards by implementing proactive measures for security, fairness, and usability. This progressive framework ensures that AI models are assessed in a manner that acknowledges incremental improvements while maintaining strict ethical requirements. Ethical considerations

such as patient privacy, informed consent, and bias mitigation are inherently qualitative, making their assessment challenging. Hence, the grading system translates these principles into measurable indicators. For example, privacy measures are assessed based on encryption and anonymization, while informed consent is evaluated through clarity, accessibility, and comprehensiveness. Bias mitigation efforts are quantified by analyzing demographic representation and the frequency of bias examinations. By transforming these qualitative aspects into metrics, the grading system ensures that AI compliance is systematically evaluated, allowing stakeholders to precisely gauge an AI model's ethical performance.

2) *Ethical Trade-offs and Balance:* The grading system acknowledges the trade-offs in AI development, balancing model accuracy, transparency, security, and fairness. AI models that emphasize accuracy at the expense of transparency receive lower interpretability scores, while models prioritizing privacy without sufficient compliance measures receive lower regulatory adherence. This balanced approach prevents any single aspect from being overemphasized at the expense of another, allowing a balanced evaluation of the model.

3) *Pass/Fail Criteria Thresholds:* The grading system classifies AI models into four levels based on their ethical compliance, security, fairness, and transparency, ranging from severely inadequate to fully compliant. Level 1 models pose severe ethical and security risks, failing to meet basic industry standards, making them unsuitable for real-world usage. Level 2 models demonstrate partial compliance, but require significant modifications to mitigate ethical concerns before being considered reliable. Level 3 models are deemed ethically sound and compliant with industry regulations, incorporating strong security, privacy, and fairness measures, though they may need minor refinements to optimize ethical performance. Level 4 models represent full compliance with industry standards, demonstrating proactive strategies to ensure long-term reliability, ethical integrity, and responsible AI implementation. To pass the grading system, an AI model must achieve a minimum threshold of 50% in each section, ensuring it meets basic ethical, security, and transparency requirements. The evaluation framework aligns with GDPR, HIPAA, and industry best practices, emphasizing ethical compliance while considering accessibility for real-world application. The scoring system categorizes models based on performance, where below 50% indicates failure and requires major improvements, scores between 50-75% are satisfactory but need refinement, and scores above 75% demonstrate strong ethical compliance to best practices. Each section of the evaluation contributes to the overall ethical assessment of an AI model. In Section one: Patient Privacy and Security (Total Score: 20), models scoring 0-10 fail to meet ethical security standards and require major improvements, while scores between 11-15 meet baseline requirements but need refinement, and those above 15 are generally compliant with minor adjustments needed. In Section two: Data Collection Practices (Total Score: 16), models below 8 fail ethical standards, scores between 9-12 are satisfactory but could improve, and scores above 12 demonstrate respon-

sible data collection practices. In Section three: Interpretation and Responsibility (Total Score: 28), scores below 14 indicate a need for major improvements, those between 15-21 are satisfactory but require refinements, and those above 21 meet ethical guidelines and demonstrate strong accountability. This structured evaluation ensures that higher-scoring models reflect increasingly proactive and comprehensive ethical measures, rather than simply meeting minimum compliance. By incentivizing models to prioritize security, fairness, and transparency, the system establishes a clear guidance for improvement and aligns with regulatory best practices. This framework not only advances ethical AI development, but also provides a guiding structure for developers and policymakers to create AI solutions that are both innovative and responsible.

V. APPLICATION OF THE FRAMEWORK

In this section, the proposed ethical guidelines will be applied to the paper "Advanced AI-driven approach for enhanced brain tumour detection from MRI images utilizing EfficientNetB2" by Rahman et al [12]. This paper was chosen for review as it explores the use of AI in medical imaging, specifically for brain tumour detection, which aligns well with the key areas of the criteria: patient privacy and security, data collection, and AI interpretability. The criteria will help highlight the ethical considerations associated with the three main issues as well as transparency in AI decision-making of this specific paper. Applying these standards will assess how well the study [12] adheres to the ethical principles in AI deployment within healthcare, pinpointing areas that require improvement. This section serves as an example of how policymakers and clinicians who hope to implement AI-based technology should approach future applications and ethical concerns.

A. Application of Section 1 Guidelines

Section One of the guidelines, found in the Appendix A, includes actionable measures such as encryption and anonymization of patient data, mechanisms to prevent unauthorized access, informed consent procedures, opt-out options, and adherence to regulatory frameworks such as GDPR and HIPAA. Applying these guidelines to the article "Advanced AI-driven approach for enhanced brain tumor detection from MRI images utilizing EfficientNetB2" [12] makes it evident that the study lacks critical privacy safeguards. While the research utilizes publicly available datasets such as those from Kaggle, it does not clarify whether these datasets meet standard privacy regulations or whether patients were given a choice to opt out. Another key component of this section is informed consent, which means patients should be aware of how their data is being used, collected, and stored. The study does not discuss the consent mechanism or transparency in using the AI model. This lack of coverage raises ethical concerns, as patient data is central to AI training, and its misuse could lead to privacy violations in patient data. Furthermore, the study does not detail security measures implemented to prevent unauthorized data access, leaving patient information

vulnerable. Regulatory compliance is another missing factor, as there is no mention of how the model aligns with data protection laws. Given these shortcomings, the study scores a level 2 for section 1. The reasoning for the score is that they use publicly available datasets. However, they lack mention or action of protection and transparency regarding data security, consent, and patient control over personal information. To improve, researchers should implement encryption, outline informed consent processes, ensure opt-out options, and align with GDPR/HIPAA standards to enhance data protection.

B. Application of Section 2 Guidelines

Section Two of the guidelines [A], highlights the importance of utilizing diverse and representative datasets, along with ensuring transparency in data collection, to mitigate bias and improve the reliability of the results. The study by Rahman et al. [12] uses three publicly available MRI datasets (BD-BrainTumor, Brain-Tumor-Detection, and Brain-MRI-Images-for-Brain-Tumor-detection) to train an EfficientNetB2 deep learning model for brain tumor detection. While this demonstrates an effort toward dataset diversity, the study does not provide demographic breakdowns, fairness assessments, or bias mitigation strategies. Issues with data diversity can raise concerns about how well the model generalizes across different populations, tumor types, and imaging conditions. A key weakness is the lack of inclusion of rare cases, which may impact the model accuracy for underrepresented tumor types or patient demographics. Additionally, no information about dataset bias or continuous monitoring mechanisms is provided, making it difficult to determine if the model adapts to real-world variations. The absence of explainability features and clinician involvement in data interpretation further limits transparency. Trust in AI-driven diagnostics may be reduced without clearly disclosing how patients and healthcare providers are informed about dataset limitations. Hence, the study scores a level 3 in data collection. The study effectively uses multiple datasets and advanced preprocessing techniques but lacks bias analysis, fairness checks, and long-term validation in clinical settings. The study should include rare tumor cases, conduct fairness audits, and implement ongoing bias monitoring to improve its score in section 2. Future AI models should prioritize transparency and real-world validation to ensure equitable healthcare applications.

C. Application of Section 3 Guidelines

Section 3 of the AI ethics assessment matrix [A] involves interpretation and accountability, whether and how the AI models aid doctors in decision-making, are transparent, and yield accountability. It establishes whether the AI is presenting useful help, offering explainability, offering confidence scores, defining boundaries, and embedded in clinical workflows. Section 3 also deals with physicians' training and how accountability is distributed between human clinicians and AI technology. Applying these criteria to Rahman et al.'s [12] work, we see that the proposed EfficientNetB2-based model for brain tumor detection is very accurate and possesses good

clinical potential. However, it lacks any significant features of interpretability or transparency. The model's decision-making process is not well explained, and while confidence scores are mentioned, their calibration and reliability are not mentioned. Moreover, education of physicians in AI interpretation is not addressed, and over-reliance or misinterpretation of AI results is feared. Generally, this study falls partially within Section 3 of the matrix, at an estimated level 3. The model is applicable in decision support but requires a high level of physician control. The weakest aspect is the lack of formal training of clinicians, which can be resolved by making AI interpretation courses mandatory and having specific guidelines on AI-assisted MRI evaluation. In addition, policymakers need to ensure that AI algorithms applied in radiology are transparent, have well-documented limitations, and function effectively in clinical practice to enhance trust and reliability in health applications.

D. Overall Performance

The objective of the AI ethics guideline is to ensure that AI-driven healthcare applications align with ethical, legal, and safety standards while promoting fairness, transparency, and accountability. The framework assesses AI models in MRI-based diagnostics across three key areas: patient privacy and security, data collection practices, and interpretation and responsibility. By applying this guideline to Advanced AI-driven approach for enhanced brain tumor detection from MRI images utilizing EfficientNetB2, we can evaluate how well this study adheres to ethical principles in AI deployment within medical imaging. In Section 1, which focuses on patient privacy and security, the study performed poorly, scoring a Level 2. While the dataset used in the study was publicly available, the paper did not address key privacy safeguards such as encryption, informed consent, or patient opt-out mechanisms. There is also a lack of discussion regarding compliance with GDPR or HIPAA standards, raising ethical concerns about data security. To improve, future AI research must prioritize explicit policies on patient consent, transparency in data use, and robust security measures to prevent unauthorized access and misuse of medical data. In Section 2, which evaluates data collection fairness and representativeness, the study performed slightly better, scoring a Level 3. The research used multiple datasets, demonstrating some level of diversity in training data. However, the absence of demographic breakdowns and fairness assessments makes it difficult to determine whether the model generalizes well across different patient groups. The study also lacked bias mitigation strategies, and no measures were in place for continuous monitoring of AI bias in real-world applications. To align better with ethical standards, future research should integrate rare cases, conduct fairness audits, and establish long-term monitoring strategies to prevent AI bias from affecting clinical outcomes. In Section 3, which assesses AI interpretability and physician responsibility, the study scored another Level 3. While the model demonstrated strong accuracy, it lacked crucial elements of explainability and transparency. Confidence scores were not clearly cali-

brated, and there was no structured training for physicians to interpret AI-generated outputs. Without a clear explanation of how the model reaches its decisions, there is a risk of over-reliance on AI or physician misinterpretation. To improve, AI developers should implement transparent confidence scoring, ensure full documentation of model limitations, and provide mandatory AI training for clinicians to mitigate risks of blind AI reliance.

VI. CONCLUSION

In conclusion, this paper provided a broad review of the current usage of AI models in MRI technology to help physicians with patient diagnosis. While conducting these reviews, several ethical issues specific to MRI usage became evident. As a result, a comprehensive guideline was developed for policymakers and developers to grade the ethical implications of the model. Looking ahead, the implications of these ethical guidelines extend far beyond this study. As AI continues to shape medical diagnostics, researchers, policymakers, and healthcare providers must work together to establish universal ethical standards for AI in radiology and medical imaging. Future AI models should be designed with privacy-first architectures, fairness-aware algorithms, and clinician-in-the-loop frameworks to ensure ethical and responsible AI implementation in patient care. Transparency, accountability, and human oversight must remain central principles in AI development, ensuring that AI enhances medical decision-making without undermining physician expertise or compromising patient rights. By adopting these ethical standards, the future of AI in healthcare can be both transformational and ethically sound.

VII. FUTURE WORK

Future work could improve the level of detail of the proposed guidelines and allow for a more flexible grading system. It may be difficult for model developers and policy makers to objectively assess the nature of the model, so it may be helpful to have an external party assess the ethical nature of their models.

VIII. LIMITATIONS

The obvious limitations of this review include the lack of specificity when detailing the original datasets. It is difficult to outline the representativeness and diversity of the dataset as well as the accuracy of the models since the specifics of the training data are undisclosed.

REFERENCES

- [1] M. A. Mazurowski, M. Buda, A. Saha, and M. R. Bashir, “Deep learning in radiology: An overview of the concepts and a survey of the state of the art with focus on mri,” *Journal of magnetic resonance imaging*, vol. 49, no. 4, pp. 939–954, 2019.
- [2] A. Hernandez-Trinidad, B. O. Murillo-Ortiz, R. Guzman-Cabrera, and T. Cordova-Fraga, “Applications of artificial intelligence in the classification of magnetic resonance images: Advances and perspectives,” *New Advances in Magnetic Resonance Imaging*, 2023.
- [3] R. Bouderhem, “Shaping the future of ai in healthcare through ethics and governance,” *Humanities and social sciences communications*, vol. 11, no. 1, pp. 1–12, 2024.
- [4] R. Jain, N. Jain, A. Aggarwal, and D. J. Hemanth, “Convolutional neural network based alzheimer’s disease classification from magnetic resonance brain images,” *Cognitive Systems Research*, vol. 57, pp. 147–159, 2019.
- [5] H. Aboutorab, O. K. Hussain, M. Saberi, F. K. Hussain, and E. Chang, “A survey on the suitability of risk identification techniques in the current networked environment,” *Journal of Network and Computer Applications*, vol. 178, p. 102984, 2021.
- [6] X. Zhao, C. K. E. Ang, U. R. Acharya, and K. H. Cheong, “Application of artificial intelligence techniques for the detection of alzheimer’s disease using structural mri images,” *Biocybernetics and Biomedical Engineering*, vol. 41, no. 2, pp. 456–473, 2021.
- [7] S.-h. Zhen, M. Cheng, Y.-b. Tao, Y.-f. Wang, S. Juengpanich, Z.-y. Jiang, Y.-k. Jiang, Y.-y. Yan, W. Lu, J.-m. Lue, et al., “Deep learning for accurate diagnosis of liver tumor based on magnetic resonance imaging and clinical data,” *Frontiers in oncology*, vol. 10, p. 680, 2020.
- [8] L. Lu, S. Shi, B. Liu, and C. Liu, “Analysis of factors influencing the organizational capacity of institutional review boards in china: a crisp-set qualitative comparative analysis based on 107 cases,” *BMC Medical Ethics*, vol. 24, no. 1, p. 74, 2023.
- [9] D. Sadeghi, A. Shocib, N. Ghassemi, P. Moridian, A. Khadem, R. Alizadehsani, M. Teshnehlab, J. M. Gorri, F. Khozeimeh, Y.-D. Zhang, et al., “An overview of artificial intelligence techniques for diagnosis of schizophrenia based on magnetic resonance imaging modalities: Methods, challenges, and future works,” *Computers in Biology and Medicine*, vol. 146, p. 105554, 2022.
- [10] T. R. Insel, “Rethinking schizophrenia,” *Nature*, vol. 468, no. 7321, pp. 187–193, 2010.
- [11] D. Bzdok and A. Meyer-Lindenberg, “Machine learning for precision psychiatry: opportunities and challenges,” *Biological Psychiatry: Cognitive Neuroscience and Neuroimaging*, vol. 3, no. 3, pp. 223–230, 2018.
- [12] A. Zubair Rahman, M. Gupta, S. Aarathi, T. Mahesh, V. Vinoth Kumar, S. Yogesh Kumaran, and S. Guluwadi, “Advanced ai-driven approach for enhanced brain tumor detection from mri images utilizing efficientnetb2 with equalization and homomorphic filtering,” *BMC Medical Informatics and Decision Making*, vol. 24, no. 1, p. 113, 2024.
- [13] F. Bray, J. Ferlay, I. Soerjomataram, R. L. Siegel, L. A. Torre, and A. Jemal, “Global cancer statistics 2018: Globocan estimates of incidence and mortality worldwide for 36 cancers in 185 countries,” *CA: a cancer journal for clinicians*, vol. 68, no. 6, pp. 394–424, 2018.
- [14] L. Tabár, B. Vitak, T. H.-H. Chen, A. M.-F. Yen, A. Cohen, T. Tot, S. Y.-H. Chiu, S. L.-S. Chen, J. C.-Y. Fann, J. Rosell, et al., “Swedish two-county trial: impact of mammographic screening on breast cancer mortality during 3 decades,” *Radiology*, vol. 260, no. 3, pp. 658–663, 2011.
- [15] J. G. Elmore, S. L. Jackson, L. Abraham, D. L. Miglioretti, P. A. Carney, B. M. Geller, B. C. Yankaskas, K. Kerlikowske, T. Onega, R. D. Rosenberg, et al., “Variability in interpretive performance at screening mammography and radiologists’ characteristics associated with accuracy,” *Radiology*, vol. 253, no. 3, pp. 641–651, 2009.
- [16] S. M. McKinney, M. Sieniek, V. Godbole, J. Godwin, N. Antropova, H. Ashrafiyan, T. Back, M. Chesus, G. S. Corrado, A. Darzi, et al., “International evaluation of an ai system for breast cancer screening,” *Nature*, vol. 577, no. 7788, pp. 89–94, 2020.
- [17] N. Stogiannos, R. Malik, A. Kumar, A. Barnes, M. Pogose, H. Harvey, M. F. McEntee, and C. Malamateniou, “Black box no more: a scoping review of ai governance frameworks to guide procurement and adoption of ai in medical imaging and radiotherapy in the uk,” *The British Journal of Radiology*, vol. 96, no. 1152, p. 20221157, 2023.
- [18] M. Goisauf and M. Cano Abadía, “Ethics of ai in radiology: a review of ethical and societal implications,” *Frontiers in big Data*, vol. 5, p. 850383, 2022.
- [19] E. Shimron and O. Perlman, “Ai in mri: computational frameworks for a faster, optimized, and automated imaging workflow,” 2023.

APPENDIX

TABLE I: Section one of the guideline outlining patient privacy and security.

Section 1: Patient Privacy and Security	1	2	3	4
The patient data being used to train the model is encrypted and anonymized before training.	There are no policies or security measures in place to protect sensitive information, creating a high risk of exposure and misuse.	Patient data is anonymized, but encryption standards are weak or inconsistently applied. Some measures exist to protect privacy, but gaps remain in ensuring robust security and compliance with data protection regulations.	Patient data is both encrypted and anonymized before training, following standard security protocols. Data protection measures meet most regulatory requirements, but periodic audits and updates are needed.	Patient data undergoes advanced encryption and thorough anonymization before training, meeting industry standards.
There are mechanisms in place to prevent unauthorized access, data breaches, and re-identification risks.	No security measures in place to prevent unauthorized access of data, with a high-risk of data breaches and/or re-identification risks.	Basic security measures exist but are insufficient or inconsistently applied.	Strong security measures, but some challenges remain.	Comprehensive security framework in place to prevent unauthorized access of data, with a low-risk of data breaches and/or re-identification risks.
The AI system provides patients with clear, accessible options to give informed consent for data collection, processing, and sharing.	No informed consent process. Data is collected and shared without patient knowledge. Consent notices do not exist, or they lack any transparency on data collection, frequency, and third-party access.	Some effort toward consent, but the information is vague, difficult to access, or missing critical details, such as what data is collected, how often, and which third parties are involved. Patients may not fully understand how their data is used.	Clear consent process with minor areas of improvement. Most key details on data collection, frequency, and third-party access are provided, but there may be minor gaps or lack of clarity in certain areas.	Fully transparent, easy-to-understand consent process where patients have full control over their data. Consent notices clearly specify what data is collected, how frequently, and which third parties have access, ensuring true informed consent.

<p>Patients can opt-out of the collection of their personal data or the application of the AI model and were explicitly informed of its use in diagnosis, training, and development.</p>	<p>No opt-out option. AI use is mandatory, and patients have no ability to refuse data collection. There is no transparency about AI involvement in diagnosis, training, or development.</p>	<p>Opt-out exists but is difficult to access or poorly explained. Information about opting out is either too vague or too complex, such as being written in overly broad or highly detailed language. Patients may not be aware that opting out is possible, or they may struggle to navigate the process due to a lack of clarity.</p>	<p>Clear opt-out options with minor challenges. Patients are explicitly informed of AI usage and have a way to opt-out but the process may have small barriers, such as requiring multiple steps, unclear instructions, or limited accessibility. Some patients may still find it challenging to opt-out.</p>	<p>Clear opt-out system with full transparency and patient control. Patients are explicitly informed about the use of AI in diagnosis, training, and development. The opt-out process is simple, accessible, and user-friendly, allowing patients to easily withdraw consent at any time.</p>
<p>The AI system ensure compliance with legal and ethical data protection standards (e.g., GDPR, HIPAA) to safeguard patient privacy.</p>	<p>No efforts are made to meet legal and ethical data protection standards. There is no mention of compliance mechanisms. The system poses a high risk of privacy breaches.</p>	<p>Some attempts to address compliance are evident, but significant gaps remain. Key safeguards are missing or insufficient, leaving patient data vulnerable.</p>	<p>The system adheres to most legal and ethical data protection requirements, with only minor weaknesses that need improvement. Privacy risks are minimal but not fully eliminated.</p>	<p>The system adhered to all legal and ethical data protection requirements. Strong safeguards are in place, with compliance being actively monitored and continuously improved to maintain data security and patient privacy.</p>
<p>Total /20</p>				

TABLE II: Section two of the guideline outlining ethical data collection.

Section 2: Data Collection	1	2	3	4
Data collection methods designed to minimize bias and ensure diversity in the dataset, particularly to prevent discrimination against vulnerable populations.	<p>Data collection lacks diversity and includes significant biases, leading to potential discrimination. There are no active efforts to ensure representation, and vulnerable populations are underrepresented or excluded.</p>	<p>Some measures are in place to promote diversity, but gaps remain in ensuring fair representation. The dataset includes different demographic groups, but bias assessments are infrequent, and vulnerable populations may still be underrepresented.</p>	<p>Data collection is designed to minimize bias and includes a representative sample of diverse populations. Regular audits and fairness assessments are conducted, and adjustments are made to address any identified biases.</p>	<p>Data collection follows best practices to ensure diversity and prevent discrimination against vulnerable populations. Proactive strategies, such as targeted data collection, fairness-aware algorithms, and continuous bias monitoring, are implemented to maintain equity and inclusivity.</p>
Patients and healthcare providers are adequately informed about how AI models are trained, including any potential data gaps or biases.	<p>No measures are in place to ensure that patients and healthcare providers understand how AI models are trained. There is minimal or non-existent awareness of data gaps or biases.</p>	<p>Limited efforts are made to provide transparency. Some basic information is available, but it is insufficient for meaningful understanding of data gaps or biases.</p>	<p>Patients and healthcare providers are provided with clear resources to understand AI model training, including potential biases and data gaps. Transparency is achieved through adequate but standard materials, but the depth of information may be limited.</p>	<p>Patients and healthcare providers are provided with well-structured and accessible resources that clearly outline the AI model's training process, data sources, potential biases, and data gaps. Transparency is robust, with materials that provide potential data gaps and biases associated with the AI model.</p>

<p>The AI model incorporates MRI data from patients with rarer cases that are underrepresented to ensure equitable and accurate diagnostic performance.</p>	<p>The AI model does not incorporate MRI data from underrepresented rare cases, leading to biased and inaccurate diagnostics.</p>	<p>Limited efforts are made to include MRI data from rare cases, but representation remains insufficient, impacting diagnostic equity in demographic, geographic, or age diversity.</p>	<p>The AI model actively integrates MRI data from rare cases to improve diagnostic accuracy and equity, addressing some data gaps.</p>	<p>The AI model systematically ensures broad representation of rare cases across demographics, geography, and age groups. Clear documentation highlights data sources, biases, and measures taken to enhance equity and accuracy.</p>
<p>The AI model plans to undergo continuous monitoring to identify and mitigate biases in real-world MRI diagnostics, with mechanisms in place to update training data.</p>	<p>The AI model provides little to no measures for continuous monitoring or bias mitigation. The AI model remains unchanged, lacking a mechanism to address biases in real-world MRI diagnostics.</p>	<p>Limited monitoring exists, but it is irregular or lacks depth. Bias mitigation efforts are minimal, and updates to training data are infrequent.</p>	<p>The AI model undergoes structured monitoring to identify and reduce biases. Mechanisms exist to update training data periodically, improving fairness and accuracy.</p>	<p>A comprehensive and proactive monitoring system is in place, continuously tracking biases in real-world MRI diagnostics. Transparent reporting, bias mitigation strategies, and regular training data updates ensure fairness and accuracy.</p>
<p>Total /16</p>				

TABLE III: Section three of the guideline outlining interpretation and responsible usage.

Section 3: Interpretation and Responsibility	1	2	3	4
The model supplements the judgement of the physician's decision and strengthens their diagnosis.	The model provides little to no meaningful assistance in clinical decision-making. It lacks interpretability, produces inconsistent results, and may even introduce errors or biases that hinder accurate diagnoses.	The model offers some assistance but is not fully reliable. It can highlight potential findings, but its outputs require significant physician oversight due to occasional inaccuracies or lack of explainability.	The model consistently enhances physician judgment by providing reliable diagnostic insights. It improves efficiency and accuracy while maintaining transparency, but physicians must still verify results before making final decisions.	The model serves as a highly effective decision-support tool, strengthening physician diagnoses with high accuracy and clear explainability. It integrates seamlessly into clinical workflows, continuously learns from real-world data, and provides interpretable recommendations that align with best medical practices.
Physicians are informed and trained how to interpret and use the results.	No formal training provided for physicians on AI-assisted MRI interpretation.	Some basic training is available, but lacks depth and does not cover AI limitations or biases.	Regular training sessions are provided, covering AI interpretation and some ethical concerns.	Comprehensive training ensures physicians understand AI outputs, biases, and ethical concerns with ongoing learning opportunities.
The model's outputs are explainable and transparent, allowing physicians to verify AI-generated MRI assessments before making a clinical decision.	AI model provides no explanations for its assessments; physicians cannot verify results.	AI provides minimal explanations, but they are unclear or overly complex.	AI outputs include confidence scores and basic explanations, allowing some verification.	AI model outputs are fully transparent, with clear confidence scores, reasoning, and decision-support tools for physicians.

The model serves as a supplementary tool, enabling physicians to enhance their MRI interpretation while retaining full responsibility for the final diagnosis.	AI operates autonomously with little to no physician oversight, posing risks of over-reliance.	AI is supplementary but not clearly defined as a decision-support tool; physicians may over-rely on it.	AI serves as a clear supplementary tool, with physicians retaining final decision-making authority.	AI is fully integrated into workflows as a decision-support tool, ensuring that physicians enhance their diagnoses while maintaining full responsibility.
The model provides confidence scores, allowing physicians to assess the reliability of AI-generated findings in the context of other clinical data.	The model does not provide confidence scores, making it difficult for physicians to assess the reliability of AI-generated findings	The model provides basic confidence scores, but they are not well-calibrated or interpretable, requiring significant physician intuition to use effectively.	The model provides confidence scores that are generally well-calibrated and useful, helping physicians weigh AI findings in context, though occasional inconsistencies exist.	The model provides highly accurate, well-calibrated confidence scores that seamlessly integrate with clinical data, allowing physicians to assess reliability with high confidence.
The model's limitations are clearly defined, ensuring that physicians remain aware of potential biases and do not over-rely on AI outputs.	The model does not communicate its limitations, leading to potential over-reliance or misinterpretation by physicians.	The model provides some general disclaimers about limitations, but they lack specificity or transparency, requiring physicians to infer potential biases.	The model clearly defines its key limitations, including known biases, and presents this information in an accessible way, though some edge cases may still be unclear.	The model thoroughly documents its limitations, biases, and potential failure cases, ensuring physicians have a comprehensive understanding and do not over-rely on AI outputs.
The model's insights are documented within clinical workflows, allowing physicians to track its influence on their final diagnostic decisions.	The model's insights are not documented within clinical workflows, making it difficult to assess its influence on decision-making.	The model's insights are recorded sporadically but lack structured documentation, limiting the ability to track AI influence effectively.	The model's insights are consistently documented in clinical workflows, allowing physicians to review and assess AI's role in decision-making, though some integration gaps remain.	The model's insights are fully integrated into clinical workflows, with structured documentation that allows physicians to track AI influence seamlessly, supporting transparency and accountability.
Total /28				

Financial Narrative Genome

Aidan LoStracco

University of Western Ontario

alostrac@uwo.ca

Sean McCorquodale

University of Western Ontario

smccorq@uwo.ca

Alyssa Spasic

University of Western Ontario

aspasic2@uwo.ca

Sudrisha Sarkar

University of Western Ontario

ssarkar32@uwo.ca

Abstract—Financial markets are driven by both quantitative data and the complex narratives that shape investor sentiment. This paper introduces the Financial Narrative Genome, an AI-driven system that extracts and visualizes the thematic structures, causal relationships, and emotional tones embedded within financial texts. We address the challenge of capturing the dynamic evolution of market narratives by employing advanced Natural Language Processing (NLP) techniques, including transformer models, to analyze news, reports, and social media. The system constructs a network graph representing the ‘narrative genome’ and tracks emotional tone changes over time. We demonstrate the system’s ability to identify key narrative shifts and correlate them with market fluctuations. This approach provides a more nuanced understanding of market dynamics, with potential applications in risk management and investment strategy. The project’s code is available on GitHub.

I. INTRODUCTION

The digital age has ushered in an unprecedented volume of unstructured textual data, transforming how information is disseminated and consumed within financial markets. While traditional quantitative analysis remains essential, the narratives embedded within news, social media, and corporate communications hold valuable insights into market sentiment and behavior. This paper presents the ‘Financial Narrative Genome,’ an AI-driven system designed to extract and visualize the complex interplay of themes, causal relationships, and emotional tones that constitute these narratives. By creating a structured representation of financial narratives, this research aims to lay the groundwork for a more nuanced understanding of market dynamics and potential predictive capabilities.

A. Motivation

The financial markets, while driven by quantifiable data, are fundamentally shaped by narratives. These narratives, encompassing news reports, social media discourse, and corporate communications, influence investor sentiment and ultimately drive market behavior. Traditional financial analysis, however, often overlooks the nuanced interplay of these narratives, relying primarily on quantitative metrics that fail to capture the dynamic and evolving nature of market sentiment. With the increasing volume and velocity of financial information, particularly in the age of social media, the need for narrative analysis tools has become paramount.

Recent advancements in Natural Language Processing (NLP) and machine learning have opened new avenues for understanding and interpreting complex textual data. Specifically, the development of transformer-based models [1] has enabled more nuanced sentiment analysis and information extraction from unstructured text [2]. However, while sentiment analysis provides a valuable measure of emotional tone, it often fails to capture the intricate web of causal relationships and thematic connections that constitute a financial narrative. Recent research has explored the extraction of financial narratives, and summarization of those narratives. However, the creation of a comprehensive ‘Financial Narrative Genome,’ which maps the evolution of these narratives over time and predicts their impact on market dynamics, remains an underexplored area.

Therefore, this research aims to address the challenge of extracting, representing, and predicting market shifts through deep narrative analysis. We introduce the concept of a Financial Narrative Genome, a system that utilizes advanced NLP techniques to deconstruct and visualize the interconnected themes, causal relationships, and emotional arcs within financial narratives. This system then uses this data to produce predictions of market shifts. By developing a system that can accurately map and interpret the ‘Financial Narrative Genome,’ we aim to provide a more comprehensive and insightful approach to financial market analysis.

B. Related Works

Basic sentiment analysis has been widely explored due to its social value, leading to numerous approaches. Techniques include feature extraction (TF-IDF, PoS tagging, negation handling), feature selection (Chi-square, Mutual Information, Information Gain), and word embeddings (Word2Vec, GloVe, BERT, ELMo) to enhance contextual understanding [3]. More recently, large language models (LLMs) have advanced natural language processing (NLP) by capturing nuances like sarcasm and context shifts, leading to more accurate sentiment predictions. By leveraging LLMs, sentiment analysis can be enriched through multiple sub-tasks, providing deeper insights into emotional tone, causal inference, tone extraction

and entity relationships.

Semantic matching identifies relationships between words or phrases based on meaning rather than exact wording. This approach leverages knowledge bases, word embeddings, or transformer models to determine whether different expressions convey the same concept. By capturing context and synonymy, semantic matching ensures that variations in phrasing do not hinder understanding [4]. In the context of the four subtasks—causal inference, theme extraction, emotional tone identification, and entity relationships—semantic matching can unify varying outputs by linking related concepts. For example, if causal inference detects “supply chain disruptions affect revenue,” while theme extraction identifies “logistics issues,” semantic matching can recognize their connection. This enables a structured, interconnected representation of financial narratives, enhancing the clarity and utility of the final graph.

C. Problem Definition

The goal of this paper is to perform robust sentiment analysis on financial narratives and create an interconnected financial narrative genome graph. Traditional sentiment analysis struggles to capture nuanced emotions, context shifts, and the complex relationships in financial texts. To address this, we propose using LLMs for sentiment extraction, along with semantic matching to link related concepts across multiple sub-tasks (e.g., causal inference, theme extraction, emotional tone identification, and entity relationships).

The problem can be formally defined as: Given a set of financial texts T , the objective is to construct a graph $G = (V, E)$, where:

- V represents nodes such as sentiment, themes, and entities.
- E represents edges that connect related entities, sub-themes, and causal effects.

This process involves the following steps:

- 1) Sentiment extraction using LLMs.
- 2) Semantic matching to link concepts across sub-tasks.
- 3) Graph construction to represent interconnected financial narratives.

By combining LLMs with semantic matching, we aim to enhance sentiment analysis and build a more structured, accurate representation of financial narratives for improved decision-making.

II. METHODOLOGY

This section outlines the methodology used in the development of our project, detailing the data presentation, proposed solutions, evaluation strategies, and additional analyses performed to optimize our solution. Our approach follows a systematic design process, focusing on emotional tone extraction and narrative evolution.

A. Data Presentation

The data used in this project was sourced from three main sources:

- *Stock Price Data:* Stock prices of relevant companies were retrieved from Yahoo Finance, spanning the period from 2010 to the present. This data provides insights into the fluctuations in stock prices, which are influenced by financial narratives and market events.
- *Financial News Data:* Financial news articles were obtained using the GNews API and the newspaper3k library. These articles provide the narrative context for the stock price data, allowing us to explore how different types of financial news impact market sentiment.
- *Annual Letter to Shareholder Data:* Annual shareholder letters from companies’ financial reports were collected to capture corporate messaging, strategic priorities, and evolving themes. These documents offer valuable insights into how companies communicate financial performance, challenges, and future outlooks to investors.

The processed data was stored in two primary files:

- *(tickename)_graph.csv:* Contains the emotional tone scores of articles, averaged by year.
- *(tickename)_news.csv:* Stores the associated financial news articles, along with metadata such as the publication date and source.

B. Solution

Our solution focuses on two key aspects: emotional tone analysis and tracking the evolution of financial narratives.

1) *Emotional Tone Extraction:* The first step in our solution was to extract emotional tones from the news articles using the Gemini API. This API provides a robust analysis of the emotional sentiment behind each article, allowing us to track how emotions like optimism, anxiety, or sadness evolve over time. We averaged the emotional tone scores on a yearly basis to better align the sentiment data with stock price fluctuations.

2) *Narrative Evolution Tracking:* In addition to emotional tone extraction, we constructed a network graph to track the evolution of financial narratives. Each theme and entity was represented as a node, and each causal relationship was represented as an edge. The relationships were linked to nodes through semantic matching. The nodes were embedded and a sentence transformer model mapped the relationships to the nodes with the high semantic similarity. This network structure helped us visualize how financial narratives evolve and relate to one another over time, offering insights into how these narratives shape market behavior.

3) *Narrative Summary:* The Gemini API synthesized extracted themes, entities, relationships, and emotions to produce a comprehensive 15-year company narrative. This summary revealed deeper historical insights than a basic overview.

C. Evaluation of Solution

To evaluate the effectiveness of our proposed solutions, we employed the following methods:

AAPL Stock Prices & Emotional Tone Over Time



Fig. 1. Emotional tone graph for AAPL

1) *Emotional Tone Accuracy:* We evaluated the accuracy of the emotional tone extraction process by comparing the results from the Gemini API with a small manually labeled dataset of financial news articles. Although this manual validation was limited, it provided a general sense of how well the API was able to classify the emotional tone of financial news. We focused primarily on ensuring that the scores for key emotions, such as optimism and anxiety, aligned with the content of the articles.

2) *Correlation Between Emotional Tones and Stock Price:* To assess the relevance of emotional tones to stock price movements, we performed a correlation analysis. Specifically, we looked at the relationship between the emotional sentiment (e.g., optimism or anxiety) and stock price changes over the same time period. This allowed us to quantify how fluctuations in sentiment correlate with market trends, providing a basis for a predictive model.

3) *Visualizing Narrative Evolution:* We also evaluated the narrative evolution tracking through the construction of network graphs (As shown in Fig. 2). The accuracy and utility of these graphs were assessed qualitatively by inspecting whether key events and articles were accurately represented and connected. We verified that the narrative graph structure accurately reflected the evolution of financial topics over time.

D. Additional Analysis and Optimization

1) *Data Integration and Final Visualization:* Finally, we integrated the emotional tone data with the stock price data for visualization purposes. The goal was to create an interactive time-series line graph, which would show the stock prices alongside emotional tone data for different emotions (As

shown in Fig. 1). This visualization was built using Plotly.js and integrated into a Flask web interface for easy exploration.

III. RESULTS

A. System Performance Overview

Our system was tested extensively using Amazon's shareholder letters, along with corresponding financial data and news articles spanning from 2010 to 2025. The results demonstrated the model's ability to capture nuanced financial narratives, linking emotional tones, key themes, and cause-and-effect relationships into a comprehensive network graph.

B. Emotional Tone Analysis

The emotional tone analysis accurately extracted sentiment from diverse sources, including news articles, social media posts, and shareholder letters. The graph generated for Amazon illustrated fluctuations in emotions such as optimism, anxiety, and sadness, with notable peaks and troughs aligning with major financial events.

Key findings include:

- *Optimism spikes* were observed during significant growth phases, such as Amazon's expansion into new markets and technological innovations.
- *Anxiety and sadness* correlated with periods of regulatory challenges and supply chain disruptions.
- *Surprise* emerged in reaction to unexpected earnings reports and acquisitions.
- *Neutral sentiments* appeared during periods of stability, providing a baseline for comparison.

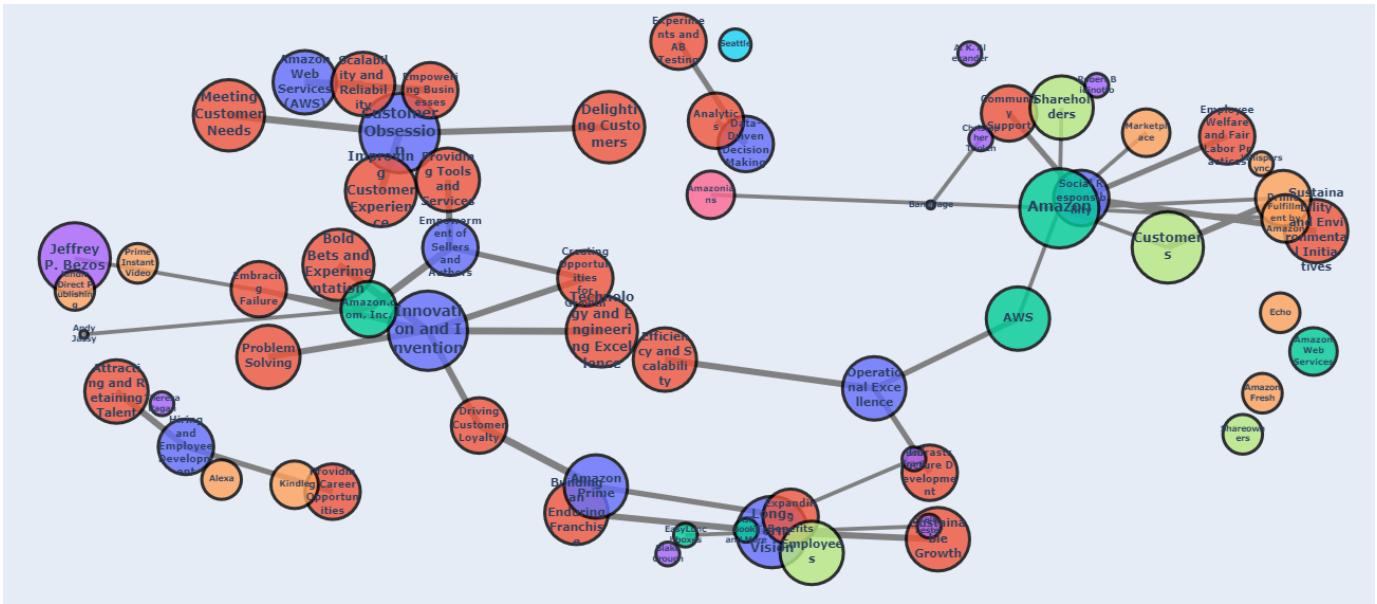


Fig. 2. Genome graph for AMZN

The interactive nature of the graph, built with Plotly and Flask, allows users to explore yearly emotional trends and compare sentiment against stock price movements. This gives an accessible yet powerful way to visualize emotional undercurrents driving market behavior.

C. Thematic and Entity Relationship Extraction

Our system successfully identified and categorized key themes, entities, and relationships. The network graph output visualizes these elements, creating a web of interconnected insights.

Key thematic extractions included:

- *AI-driven innovation* linked to revenue growth and product diversification.
- *Supply chain resilience* emerging as a recurring topic during COVID-19 and global disruptions.
- *Regulatory pressures* connected to concerns about antitrust actions and data privacy.
- *Sustainability and ESG (Environmental, Social, Governance)* themes associated with Amazon's carbon neutrality pledges and renewable energy initiatives.

Entity relationship extraction mapped crucial connections, such as partnerships, leadership changes, and competitive dynamics, showcasing how these factors intertwined with sentiment and market performance.

D. Cause and Effect Mapping

One of the most powerful outcomes of our model was the creation of a cause-and-effect narrative map. Using semantic matching through sentence embeddings and a transformer model, the system linked key financial events to

their consequences.

Examples:

- *Cost reduction strategies* leading to *higher profitability* and corresponding optimism spikes.
- *Increased regulatory scrutiny* resulting in *anxiety* and *stock price stagnation*.
- *Cloud services expansion* driving *positive sentiment* and stock surges.

This causality mapping offers actionable insights by showing not only what happened, but also why, making it a valuable tool for analysts and investors alike.

E. Website Functionality and User Interaction

Our React-based website provides an intuitive interface for users to input a company's ticker symbol and retrieve detailed financial narratives. Beyond the emotional tone graph, users can explore:

- *Annual emotional sentiment breakdowns*, with a focus on key events.
- *Stock price overlays* to compare sentiment with market performance.
- *Company narrative summaries*, generated via the Gemini Flash 2 API, condensing 15 years of data into a readable financial storyline.
- *Network graph visualizations*, highlighting key themes, entities, and causal pathways.

F. Comparative Analysis with Traditional Sentiment Analysis

To further validate our approach, we compared our financial narrative system with traditional sentiment analysis methods. The results highlighted clear advantages:

- *Depth of insight:* Traditional sentiment analysis yielded broad positive/negative sentiments, while our system uncovered thematic context and causality.
- *Accuracy* By filtering irrelevant content and focusing on company-specific narratives, our model reduced noise and improved sentiment precision.
- *Interpretability* The network graph and narrative output made complex data more digestible for users, enhancing decision-making.

G. Future Enhancements

The current results showcase a robust, scalable foundation, but there are several directions for enhancement:

- *Industry-wide benchmarking:* Extending analysis to entire sectors for cross-company comparison.
- *Real-time updates:* Incorporating live news feeds and stock data for up-to-date financial narratives.
- *Deeper social media integration:* Capturing emerging trends and retail investor sentiment.
- *Event prediction models:* Training the system to forecast potential market impacts based on evolving narratives.
- *Sentiment refinement:* Exploring multi modal analysis, combining text with audio or video content to enhance emotional extraction accuracy.

Our results demonstrate that the Financial Narrative Genome approach delivers a more comprehensive, interpretable, and actionable understanding of company performance than traditional sentiment analysis. This method not only aids in retrospective analysis but also holds the potential to improve forecasting, risk assessment, and investment strategies.

IV. CONCLUSION

This paper presented the Financial Narrative Genome, an AI-driven system designed to extract, represent, and analyze the complex narratives that shape the behavior of the financial market. By leveraging advanced NLP techniques, we demonstrated the feasibility of constructing a network graph that captures the dynamic interplay of themes, causal relationships, and emotional tones within financial texts. We further showed how this 'narrative genome' can be used to track emotional tone changes over time and identify key narrative shifts that correlate with market fluctuations. This research provides a foundation for a more nuanced and insightful approach to financial market analysis, moving beyond traditional quantitative methods.

Building upon this work, the next steps would focus on enhancing the system's predictive capabilities. Specifically, our goal is to develop more sophisticated machine learning models that can leverage the 'narrative genome' to forecast market trends with greater accuracy. This would involve incorporating a wider range of data sources, including real-time social media feeds and alternative data sets, to improve the system's robustness and responsiveness. A significant challenge remains in addressing the inherent subjectivity of narrative interpretation and mitigating potential biases in the data and models.

Furthermore, developing a robust evaluation framework that can accurately assess the system's predictive performance is crucial.

The most important area to focus on next is the integration of real-time data and the development of adaptive learning algorithms. This would enable the system to continuously learn and adapt to the evolving dynamics of financial narratives, providing more timely and accurate information. By addressing these challenges and focusing on real-time adaptation, the Financial Narrative Genome has the potential to become a valuable tool for investors, analysts, and researchers seeking to navigate the complexities of the modern financial landscape.

REFERENCES

- [1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," pp. 5998–6008, 2017.
- [2] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2018.
- [3] M. Wankhade, A. C. S. Rao, and C. Kulkarni, "A survey on sentiment analysis methods, applications, and challenges," 2022.
- [4] E. Cambria and A. Hussain, "Sentic computing," 2015.

Flow to Learn: Flow Matching on Neural Network Parameters

Daniel Saraghah

University of Toronto

daniel.saraghah@mail.utoronto.ca

Tejas Balaji

University of Toronto

tejas.balaji@mail.utoronto.ca

Deyu Cao

University of Toronto, University of Tokyo

deyu.cao@mail.utoronto.ca

Ashwin Santhosh

University of Toronto

ashwin.santhosh@mail.utoronto.ca

Abstract—Foundational language models show a remarkable ability to learn new concepts during inference via context data. However, similar work for images lag behind. To address this challenge, we introduce FLoWN, a flow matching model that learns to generate neural network parameters for different tasks. Our approach models the flow on latent space, while conditioning the process on context data. Experiments verify that FLoWN attains various desiderata for a meta-learning model. In addition, it matches or exceeds baselines on in-distribution tasks, provides better initializations for classifier training, and is performant on out-of-distribution few-shot tasks while having a fine-tuning mechanism to improve performance.

I. INTRODUCTION

Flow matching (FM) [1]–[3] is a prominent fixture in generative modeling tasks from imaging [2], [4]–[6] to language [7]–[9]. However, its application to neural network weights remains largely unexplored. In this paper, we introduce Flow-based Learning of Weights for Neural adaptation (FLoWN), a new class of method for weight generation. Empirical evaluations validate the following contributions: **1)** The generated weights match or exceed conventionally trained models on in-distribution tasks, and provide better initializations for fine-tuning on out-of-distribution (OOD) tasks, **2)** FLoWN is able to conditionally retrieve pre-trained weights from a distribution pre-trained on various datasets while matching their performance, **3)** FLoWN is capable of performing well on OOD few-shot tasks while having a fine-tuning mechanism to improve performance.

A. Motivation

Multiple approaches have been tried to generate weights capable of few-shot learning (FSL), motivated by its speed compared to conventional training. For instance, various diffusion-based approaches [10]–[12] have been used to generate neural network weights. However, flexibility is limited by its restriction to Gaussian processes and a sluggish inference speed.

More broadly, we may categorize this form of learning as meta-learning [13]–[15], which aims to learn concepts from a few demonstrations. It is therefore natural that we have two evaluation settings: in-distribution tasks and out-of-distribution (OOD) tasks. With enough training and capacity,

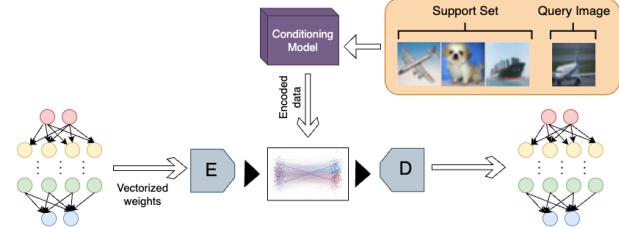


Fig. 1. A schematic of the training process of FLoWN for few-shot learning. Given a set of pre-trained target weights and a support set, we apply the conditioned flow model to pushforward a sample of the latent prior towards encoded target weights. The decoder is used during inference where we start from a sample of the latent prior and pushforward towards the target distribution with a trained vector field $v_\theta(\cdot, t; \mathbf{y})$ where \mathbf{y} is the support set embedding.

it's clear *meta-models* (i.e. models trained on multiple data distributions) should excel at in-distribution tasks. However, generalization to novel tasks often presents a challenge to meta-learning and weight generation frameworks [12], [16]. Addressing this gap – improving a model's capacity to adapt beyond its training distribution – serves as a key motivation for our work.

B. Related Works

Numerous strategies have emerged to narrow the out-of-distribution generalization gap within the realm of meta-learning and weight generation. One promising approach, proposed by Soro et al. [10], leverages diffusion-based models to enhance weight generation. While this method makes strides in bridging the generalization gap, we find that further improvements can be made, particularly by exploring alternative generative frameworks. Our proposed FLoWN framework builds on recent advances in conditional flow modeling, neural network parameter generation, and meta-learning to provide a more efficient and principled solution. Below, we review key developments in these areas; for additional context and related methods, see Appendix A.

a) Conditional Flow Matching: Lipman et al. [2] introduced the CFM objective, which learns probability paths between distributions using a conditional vector field. By modifying the coupling of the source and target distributions,

later work shows better alignment with optimal transport paths, improving inference efficiency [4]. However, the original formulations of flow matching assumed that the initial distributions were Gaussian. Pooladian et al. [17] extended the theory to arbitrary source distributions using minibatch sampling and proved a bound on the variance of the gradient of the objective.

b) Neural Network Parameter Generation: Denil et al. [18] showed that most neural network parameters are redundant, enabling weight generation techniques. Ha et al. [19] introduced Hypernetworks, which generate weights using neural network layer embeddings. More recently, Wang et al. [12] and Soro et al. [10] applied latent diffusion models to parameter generation.

c) Weight Generation for Few-Shot Learning: Few-shot learning applies meta-learning to scenarios with limited data. Ravi & Larochelle [20] introduced an LSTM-based meta-learner for dynamic weight updates. Later works [13]–[15], [21] leveraged transformers and foundation models. Diffusion models for weight generation [10]–[12], [22] have gained attention, though primarily for in-distribution tasks, leaving OOD adaptation an open challenge.

C. Problem Definition

The problem of interest is that of *conditional weight generation* with an application to few-shot learning. Our approach trains a *conditional flow model* that can generate neural network weights tailored to new tasks from minimal data. At a high level, continuous flows [1]–[3], [23] provide a way to transform one distribution into another by modeling the dynamics of an ordinary differential equation (ODE). For our purposes: **1)** We define a reference flow that connects a simple "source" distribution to the "target" distribution of interest (in this case, the space of task-specific weights). **2)** A learnable velocity field (parameterized by a neural network v_θ) is trained to match the velocity of this reference flow, effectively bridging the source and target distributions. **3)** *Context-conditioning* is used to incorporate information about the task, such as the support set in an n -way- k -shot FSL setting (detailed below), so that the generated weights are adapted to the given task.

In few-shot classification, the standard setup involves n classes, each with k labeled examples (the "support set"). The learning system must then generalize to a set of unlabeled query examples from the same or new distribution. By conditioning the flow model on the support set (and any associated label embeddings), we aim to generate a set of weights that perform well immediately, while also allowing further fine-tuning when facing OOD data.

II. METHODOLOGY

A. Preliminaries

a) Conditional flow models: Chen et al. [23] first introduced continuous normalizing flows as an effective data generation process through modeling dynamics. Simulation-free methods improve on this concept by simplifying the training

objective [1]–[3]. Following the formulation of Lipman et al. [2], given random variables $\bar{\mathbf{x}}_0 \sim p_0$ and $\bar{\mathbf{x}}_1 \sim p_1$ a data distribution, define a reference flow $\bar{\mathbf{x}} = (\bar{\mathbf{x}}_t)_{t \in [0,1]}$ where $\bar{\mathbf{x}}_t = \beta_t \bar{\mathbf{x}}_0 + \alpha_t \bar{\mathbf{x}}_1$ with the constraint that $\alpha_0 = \beta_1 = 0$ and $\alpha_1 = \beta_0 = 1$. The aim of flow modeling is to learn a sequence $\mathbf{x} = (\mathbf{x}_t)_{t \in [0,1]}$ which has the same marginal distribution as $\bar{\mathbf{x}}$. To make this a feasible task, we describe this process as an ODE: $d\mathbf{x}_t = v(\mathbf{x}_t, t)dt$ where $\mathbf{x}_0 \sim \mathcal{N}(0, \mathbf{I})$. Training proceeds by first parameterizing $v(\mathbf{x}_t, t)$ by a neural network θ and matching the reference flow velocity, i.e. $u(\mathbf{x}_t, t) := \frac{d}{dt}\bar{\mathbf{x}}_t$. This would, however, be an unfeasible training objective, therefore, we condition on samples from the terminal distribution $\mathbf{x}_1 \sim p_1$ and train

$$L_{\text{cfm}}(\theta) = \mathbb{E}_{t \sim U[0,1], \mathbf{x}_1 \sim p_1, \mathbf{x}_t \sim p_t(\cdot | \mathbf{x}_1)} \|v_\theta(\mathbf{x}_t, t) - u(\mathbf{x}_t, t, \mathbf{x}_1)\|. \quad (1)$$

Lipman et al. [2] proved that this loss produces the same gradients as the marginal loss, thus optimizing it will result in convergence to the reference $u(\mathbf{x}_t, t)$. Moreover, we can always marginalize an independent conditioning variable \mathbf{y} on v_θ, u – this will serve as our context conditioning vector.

b) Few-shot learning: The problem of few-shot learning is often formulated as a n -way- k -shot classification task. In particular, given n classes and k examples for each class, $\mathcal{S} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{nk}$ of (image, label) support set pairs, our meta-learner is tasked with classifying query set images $\mathcal{Q} = \{\mathbf{x}_{nk+1}, \dots, \mathbf{x}_{nk+q}\}$. Relevant to our approach, recently, Fifty et al. [13] proposed an image meta-learning architecture, CAML, consisting of three components: a frozen pre-trained image encoder, a class encoder, and a transformer-based sequence model. ELMES, the class encoder, was shown to possess two attractive properties: label symmetry and permutation invariance. The transformer sequence model takes the concatenation of the image and label embeddings of the support set images, and a special "unknown" label embedding is used on query set images. These query images are analogous to the [CLS] tokens in transformers as the logits corresponding to the query images are then passed into a classifier MLP to predict labels.

B. Flow to Learn

We describe the components of our approach below, alongside a method schematic in Figure 1, and leave more details to Appendix B, C.

a) Weight encoder: Due to the intractable size of weight space, it is necessary for modeling to take place in latent space. We justify this design by appealing to work on the Lottery Ticket Hypothesis [24], [25] as well as the body of work on pruning [26], which suggests that, like natural data, neural networks live on a low-dimensional manifold within its ambient space. We have two encoder variants, first is a variational autoencoder (VAE) [27] set up as in Soro et al. [10], and the second is the graph-based encoder (GE) of Kofinas et al. [28] which takes into account permutation invariance of neural networks. As the latter models connections between layers, we only use the VAE for experiments involving subsets of weights.

TABLE I
MEAN VALIDATION ACCURACY OF UNCONDITIONAL FLoWN GENERATION. *orig* DENOTES BASE MODELS TRAINED CONVENTIONALLY AND *p-diff* THOSE GENERATED USING P-DIFF [12].

Base Models	CIFAR100			CIFAR10			MNIST			STL10		
	orig.	FLoWN	p-diff.	orig.	FLoWN	p-diff.	orig.	FLoWN	p-diff.	orig.	FLoWN	p-diff.
Resnet-18	71.45	71.42	71.40	94.54	94.36	94.36	99.68	99.65	99.65	62.00	62.00	62.24
ViT-base	85.95	85.86	85.85	98.20	98.11	98.12	99.41	99.38	99.36	96.15	95.77	95.80
ConvNext-tiny	85.06	85.12	85.17	98.03	97.89	97.90	99.42	99.41	99.40	95.95	95.63	95.63
CNN w/ GE	32.09	31.73	31.81	72.53	72.15	72.09	98.93	98.89	98.89	53.88	53.64	53.80

TABLE II
MEAN VALIDATION ACCURACY OF TOP-5 FLoWN MODEL RETRIEVALS.

Method	MNIST	F-MNIST	CIFAR10	STL10
Original	91.1	72.7	48.7	39.0
FLoWN w/ mIN-prior	63.0	41.9	22.6	18.8
FLoWN	91.7	73.8	50.3	40.8

b) *Flow meta-model*: The backbone of our meta-learning framework is a conditional FM model following Tong et al. [4]. We make use of the flexibility of FM to use a non-Gaussian prior, specifically the Kaiming uniform or normal initializations [29], as the source p_0 . The data distribution p_1 of base model weights is experiment-dependent, however, broadly they are obtained by conventional training methods or through a model zoo [30].

c) *Conditioning model*: To condition our flow meta-model, we use a pre-trained CAML. Our choice is due to the extensive training of the CAML architecture on several datasets as well as the principled label encoding used in their approach. As we are only interested in encoding the support set, we consider judicious choices for the query set expected by CAML. For instance, in the Model Retrieval experiment below, we simply choose one random image for each class in the support set. In the FSL experiment, the choice is clear: the query set of each FSL task. The conditioning vector is incorporated by concatenating to the latent vector.

III. RESULTS

First, we confirm various properties that are to be expected of weight generation models. Next, we examine FLoWN’s performance in few-shot learning. Further details are provided in Appendix D.

A. Basic Properties of FLoWN

a) *Unconditional generation*: We first evaluate the basic modeling capabilities of the flow meta-model. The target distribution p_1 is generated by training a variety of base models on known datasets: CIFAR-10, CIFAR-100, and MNIST, and saving 200 weight checkpoints each. For large models, we can choose to generate only a subset of the weights. In our case, we generate the batch norm parameters for Resnet-18 [31], ViT-base [32] and ConvNext-tiny [33], and the full medium-CNN [30]. The aim of this test is to train a separate meta-model for each dataset and validate its base model reconstruc-

tion on classifying its corresponding test set. Table I shows that we are able to match base models trained conventionally and with p-diff [12].

b) *Model retrieval and in-distribution initialization*: Following [10], we perform model retrieval to test whether the meta-model can distinguish weights of the base model given conditioning samples from the dataset the base model was trained on. The base model is a simple 4-layer ConvNet and we obtain 100 weight checkpoints from the model zoo [30] for each dataset: MNIST, Fashion-MNIST (F-MNIST), CIFAR-10, and STL10 after 46-50 epochs of conventional training. Unlike in the previous test, we will train just a single meta-model on 400 total base models conditioned on support samples from their training set via CAML. During validation, we pass in a random support sample from one of the four datasets and generate the *full* ConvNet. In Table II, we see that our top-5 validation accuracy matches that of the base models. Additionally, we repeated this experiment using weights from mini-Imagenet as a prior, but they seem to perform considerably worse than just Kaiming normal (see Appendix D2 for a discussion). Next, we repeat this experiment but instead using weight checkpoints from epochs 21-25, and use the generated weights as an initialization before fine-tuning another 25 epochs. As shown in Table III, our initialization enjoys faster convergence, even for datasets on which the model was not trained, highlighting the generalization capability of our meta-model.

c) *Fine-tuning the meta-model on OOD data*: The setting of unconditional generation is quite restrictive as it is assumed that the output classifier has the same architecture and is to be used on the same dataset. In this experiment, we evaluate whether the meta-model can be effectively fine-tuned to achieve better performance on out-of-distribution data. We start with the meta-model trained from unconditional generation and generate weights for a different dataset. Subsequently, we compute the cross-entropy loss and backpropagate the gradients through the FM model. As this entails backpropagation through an ODE solver, we implement a stopgrad mechanism that restricts gradient flow before a time $0 < t' < 1$ to trade off accuracy for efficiency. Due to time constraints, we restrict ourselves to batch norms of Resnet-18 and the small-CNN. Table IV shows considerable improvement over generations obtained from a static FM meta-model and the VAE.

TABLE III
MEAN VALIDATION ACCURACY OF FINE-TUNED GENERATED WEIGHTS POST-RETRIEVAL. THE ASTERISK (*) INDICATES DATASETS ON WHICH THE MODEL WAS NOT TRAINED.

Epoch	Method	MNIST	F-MNIST	CIFAR10	STL10	USPS*	SVHN*	KMNIST*
0	RandomInit	~ 10%	~ 10%	~ 10%	~ 10%	~ 10%	~ 10%	~ 10%
	FLoWN	83.58 ± 0.58	68.50 ± 0.64	45.93 ± 0.57	35.16 ± 1.24	57.53 ± 2.43	17.99 ± 0.82	11.79 ± 0.51
1	RandomInit	18.12 ± 1.58	26.90 ± 0.52	28.75 ± 0.22	18.94 ± 0.09	17.69 ± 0.00	19.50 ± 0.03	14.48 ± 0.06
	FLoWN	84.49 ± 0.65	69.09 ± 0.40	46.85 ± 0.30	36.15 ± 1.14	72.45 ± 1.81	68.64 ± 7.07	51.15 ± 8.90
5	RandomInit	35.05 ± 3.87	51.08 ± 2.15	40.00 ± 0.20	28.24 ± 0.01	32.77 ± 0.46	39.59 ± 10.0	30.00 ± 0.30
	FLoWN	87.68 ± 0.44	70.32 ± 0.50	47.44 ± 0.55	37.43 ± 1.19	76.96 ± 1.29	77.36 ± 1.07	69.14 ± 10.1
25	RandomInit	87.70 ± 0.90	70.69 ± 0.46	46.86 ± 0.01	36.75 ± 0.10	82.02 ± 0.12	58.56 ± 19.5	55.05 ± 0.06
	FLoWN	92.29 ± 0.41	73.72 ± 0.68	49.25 ± 0.73	40.14 ± 1.07	82.28 ± 1.40	78.75 ± 1.30	79.11 ± 6.65
50	RandomInit	92.76 ± 0.08	72.88 ± 0.46	48.85 ± 0.74	40.47 ± 0.18	88.35 ± 0.18	63.70 ± 22.1	64.32 ± 0.25

TABLE IV

FINE-TUNING ON OOD DATA. DATA-F ARE RESULTS GENERATED FROM FINE-TUNED META-MODELS, WHEREAS DATA-S ARE FROM STATIC META-MODELS. HERE, WE GENERATE THE FULL CNN WEIGHTS, WHEREAS WE ONLY MODIFY THE BATCH NORMS OF RESNET-18.

Base dataset	ResNet-18		CNN w/ VAE		CNN w/ GE	
	CIFAR10	STL10	CIFAR10	STL10	CIFAR10	STL10
CIFAR10-S	–	64.20	–	24.01	–	23.03
CIFAR10-F	–	72.87	–	60.09	–	60.85
STL10-S	93.09	–	19.97	–	18.13	–
STL10-F	94.06	–	61.38	–	69.42	–

TABLE V

FEW-SHOT LEARNING ACCURACY ON OUT-OF-DISTRIBUTION TASKS. WE COMPARE WITH D2NGW AND BEST WEIGHTS GENERATED BY FLOWN.

Model	CIFAR-10	STL10
FLoWN-best	73.1	80.4
D2NGW [10]	33.04 ± 0.04	50.42 ± 0.13
FLoWN	35.84 ± 2.71	35.35 ± 2.77

B. Few-shot learning

For this evaluation, we utilize mini-Imagenet [34] and Chen et al. [35] for a Resnet-12 architecture. Following the typical FSL setting, we partition the dataset into meta-train and meta-test sets and further into tasks whose size depends on the way and shot parameters. For instance, for 5-way-1-shot, the support set consists of one image from 5 different classes whereas the query set is always 15 images for each of the 5 classes in the support set. The in-distribution test entails labeling query images from the same dataset (i.e. trained on mini-Imagenet and evaluated on mini-Imagenet), whereas OOD tasks entails labeling novel query images. First, we train the Resnet-12 on the train split of mini-Imagenet; our goal is thus to generate a classifier head for each task.

In our case, we set 50,000 tasks in the meta-train set and 100 in the meta-test set. We perform this test by constructing a target distribution using pre-trained weights from Resnet-12, linear-probing a classifier head on top of the Resnet backbone for each of the 50,000 subsets for 100 epochs using the AdamW optimizer with a learning rate of 10^{-3} and weight decay of 10^{-2} . Given our computational constraints, we

evaluated FLoWN on just two out-of-distribution datasets: CIFAR10 and STL10 by sampling weights 50 different times and taking the average top-3 accuracies. Table V shows that our method achieves marginal gains on CIFAR10, but performance on STL10 remains below the baseline. Considering the high validation accuracies of our VAE, we anticipate that further training and tinkering will enhance FLoWN generalization across tasks.

IV. CONCLUSION

In this work, we have provided a preliminary investigation of FLoWN for weight generation with an application to few-shot learning. Future research directions include: 1) training FLoWN on a more comprehensive image dataset to improve efficacy on OOD tasks, 2) a post-hoc fine-tuning mechanism [36] for adapting FLoWN to difficult domains (e.g. medical imaging), 3) incorporating intermediate base model weights obtained during conventional training to guide the inference trajectory of generated weights (e.g. via MetricFM [37]).

REFERENCES

- [1] M. S. Albergo and E. Vanden-Eijnden, “Building normalizing flows with stochastic interpolants,” in *The Eleventh International Conference on Learning Representations*, 2023. [Online]. Available: <https://openreview.net/forum?id=li7qeBbCR1t>
- [2] Y. Lipman, R. T. Q. Chen, H. Ben-Hamu, M. Nickel, and M. Le, “Flow Matching for Generative Modeling,” Feb. 2023.
- [3] X. Liu, C. Gong, and qiang liu, “Flow straight and fast: Learning to generate and transfer data with rectified flow,” in *The Eleventh International Conference on Learning Representations*, 2023. [Online]. Available: <https://openreview.net/forum?id=XVjTT1nw5z>
- [4] A. Tong, K. Fatras, N. Malkin, G. Huguet, Y. Zhang, J. Rector-Brooks, G. Wolf, and Y. Bengio, “Improving and generalizing flow-based generative models with minibatch optimal transport,” Mar. 2024.
- [5] P. Esser, S. Kulal, A. Blattmann, R. Entezari, J. Müller, H. Saini, Y. Levi, D. Lorenz, A. Sauer, F. Boesel, D. Podell, T. Dockhorn, Z. English, K. Lacey, A. Goodwin, Y. Marek, and R. Rombach, “Scaling rectified flow transformers for high-resolution image synthesis,” 2024. [Online]. Available: <https://arxiv.org/abs/2403.03206>
- [6] X. Liu, X. Zhang, J. Ma, J. Peng, and qiang liu, “Instaflow: One step is enough for high-quality diffusion-based text-to-image generation,” in *The Twelfth International Conference on Learning Representations*, 2024. [Online]. Available: <https://openreview.net/forum?id=1k4yZbbDqX>
- [7] I. Gat, T. Remez, N. Shaul, F. Kreuk, R. T. Q. Chen, G. Synnaeve, Y. Adi, and Y. Lipman, “Discrete flow matching,” in *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. [Online]. Available: <https://openreview.net/forum?id=GTDKo3Sv9p>

- [8] N. Shaul, I. Gat, M. Havasi, D. Severo, A. Sriram, P. Holderrieth, B. Karrer, Y. Lipman, and R. T. Q. Chen, "Flow matching with general discrete paths: A kinetic-optimal perspective," 2024. [Online]. Available: <https://arxiv.org/abs/2412.03487>
- [9] A. Campbell, J. Yim, R. Barzilay, T. Rainforth, and T. Jaakkola, "Generative flows on discrete state-spaces: Enabling multimodal flows with applications to protein co-design," *arXiv preprint arXiv:2402.04997*, 2024.
- [10] B. Soro, B. Andreis, H. Lee, S. Chong, F. Hutter, and S. J. Hwang, "Diffusion-based Neural Network Weights Generation," Feb. 2024.
- [11] B. Zhang, C. Luo, D. Yu, X. Li, H. Lin, Y. Ye, and B. Zhang, "Metadiff: Meta-learning with conditional diffusion for few-shot learning," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 15, pp. 16 687–16 695, Mar. 2024.
- [12] K. Wang, Z. Xu, Y. Zhou, Z. Zang, T. Darrell, Z. Liu, and Y. You, "Neural Network Diffusion," Feb. 2024.
- [13] C. Fifty, D. Duan, R. G. Jenkins, E. Amid, J. Leskovec, C. Re, and S. Thrun, "Context-Aware Meta-Learning," Mar. 2024.
- [14] S. X. Hu, D. Li, J. Stühmer, M. Kim, and T. M. Hospedales, "Pushing the Limits of Simple Pipelines for Few-Shot Learning: External Data and Fine-Tuning Make a Difference," Apr. 2022.
- [15] A. Zhmoginov, M. Sandler, and M. Vladymyrov, "HyperTransformer: Model Generation for Supervised and Semi-Supervised Few-Shot Learning," Jul. 2022.
- [16] K. Schürholt, M. W. Mahoney, and D. Borth, "Towards scalable and versatile weight space learning," in *Proceedings of the 41st International Conference on Machine Learning (ICML)*. PMLR, 2024.
- [17] A.-A. Pooladian, H. Ben-Hamu, C. Domingo-Enrich, B. Amos, Y. Lipman, and R. T. Q. Chen, "Multisample flow matching: straightening flows with minibatch couplings," in *Proceedings of the 40th International Conference on Machine Learning*, ser. ICML'23. JMLR.org, 2023.
- [18] M. Denil, B. Shakibi, L. Dinh, M. Ranzato, and N. de Freitas, "Predicting Parameters in Deep Learning," Oct. 2014.
- [19] D. Ha, A. M. Dai, and Q. V. Le, "Hypernetworks," in *International Conference on Learning Representations*, 2017.
- [20] S. Ravi and H. Larochelle, "Optimization as a Model for Few-Shot Learning," in *International Conference on Learning Representations*, Feb. 2017.
- [21] L. Kirsch, J. Harrison, J. Sohl-Dickstein, and L. Metz, "General-purpose in-context learning by meta-learning transformers," 2024. [Online]. Available: <https://arxiv.org/abs/2212.04458>
- [22] Y. Du, Z. Xiao, S. Liao, and C. Snoek, "ProtoDiff: Learning to Learn Prototypical Networks by Task-Guided Diffusion," Nov. 2023.
- [23] R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. Duvenaud, "Neural Ordinary Differential Equations," Dec. 2019.
- [24] J. Frankle and M. Carbin, "The lottery ticket hypothesis: Finding sparse, trainable neural networks," in *International Conference on Learning Representations*, 2019. [Online]. Available: <https://openreview.net/forum?id=rJL-b3RcF7>
- [25] B. Liu, Z. Zhang, P. He, Z. Wang, Y. Xiao, R. Ye, Y. Zhou, W.-S. Ku, and B. Hui, "A survey of lottery ticket hypothesis," 2024. [Online]. Available: <https://arxiv.org/abs/2403.04861>
- [26] H. Cheng, M. Zhang, and J. Q. Shi, "A survey on deep neural network pruning: Taxonomy, comparison, analysis, and recommendations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 12, pp. 10 558–10 578, 2024.
- [27] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," 2022. [Online]. Available: <https://arxiv.org/abs/1312.6114>
- [28] M. Kofinas, B. Knyazev, Y. Zhang, Y. Chen, G. J. Burghouts, E. Gavves, C. G. M. Snoek, and D. W. Zhang, "Graph Neural Networks for Learning Equivariant Representations of Neural Networks," <https://arxiv.org/abs/2403.12143v3>, Mar. 2024.
- [29] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1026–1034.
- [30] K. Schürholt, D. Taskiran, B. Knyazev, X. Giró-i Nieto, and D. Borth, "Model zoos: A dataset of diverse populations of neural network models," in *Thirty-Sixth Conference on Neural Information Processing Systems (NeurIPS) Track on Datasets and Benchmarks*, Sep. 2022.
- [31] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2015. [Online]. Available: <https://api.semanticscholar.org/CorpusID:206594692>
- [32] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations*, 2021. [Online]. Available: <https://openreview.net/forum?id=YicbFdNTTy>
- [33] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A convnet for the 2020s," 2022. [Online]. Available: <https://arxiv.org/abs/2201.03545>
- [34] O. Vinyals, C. Blundell, T. Lillicrap, k. kavukcuoglu, and D. Wierstra, "Matching networks for one shot learning," in *Advances in Neural Information Processing Systems*, D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, Eds., vol. 29. Curran Associates, Inc., 2016. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2016/file/90e1357833654983612fb05e3ec9148c-Paper.pdf
- [35] Y. Chen, Z. Liu, H. Xu, T. Darrell, and X. Wang, "Meta-baseline: Exploring simple meta-learning for few-shot learning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 9062–9071.
- [36] C. Domingo-Enrich, M. Drozdal, B. Karrer, and R. T. Q. Chen, "Adjoint Matching: Fine-tuning Flow and Diffusion Generative Models with Memoryless Stochastic Optimal Control," Sep. 2024.
- [37] K. Kapusniak, P. Potapchik, T. Reu, L. Zhang, A. Tong, M. Bronstein, A. J. Bose, and F. Di Giovanni, "Metric Flow Matching for Smooth Interpolations on the Data Manifold," May 2024.
- [38] C. Finn, P. Abbeel, and S. Levine, "Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks," in *Proceedings of the 34th International Conference on Machine Learning*. PMLR, Jul. 2017, pp. 1126–1135.
- [39] A. Antoniou, H. Edwards, and A. Storkey, "How to train your MAML," in *International Conference on Learning Representations*, 2019. [Online]. Available: <https://openreview.net/forum?id=HJGven05Y7>
- [40] A. Rajeswaran, C. Finn, S. M. Kakade, and S. Levine, *Meta-learning with implicit gradients*. Red Hook, NY, USA: Curran Associates Inc., 2019.
- [41] D. Zhao, S. Kobayashi, J. Sacramento, and J. von Oswald, "Meta-Learning via Hypernetworks," in *4th Workshop on Meta-Learning at NeurIPS 2020 (MetaLearn 2020)*. NeurIPS, Dec. 2020.
- [42] M. Przewieczkowski, P. Przybysz, J. Tabor, M. Zieba, and P. Spurek, "Hypermaml: Few-shot adaptation of deep models with hypernetworks," *ArXiv*, vol. abs/2205.15745, 2022.
- [43] J. Beck, M. T. Jackson, R. Vuorio, and S. Whiteson, "Hypernetworks in Meta-Reinforcement Learning," in *Proceedings of The 6th Conference on Robot Learning*. PMLR, Mar. 2023, pp. 1478–1487.
- [44] J. Lee, A. Xie, A. Pacchiano, Y. Chandak, C. Finn, O. Nachum, and E. Brunskill, "Supervised Pretraining Can Learn In-Context Reinforcement Learning," *Advances in Neural Information Processing Systems*, vol. 36, pp. 43 057–43 083, Dec. 2023.
- [45] G. Corso, L. Cavalleri, D. Beaini, P. Liò, and P. Veličković, "Principal neighbourhood aggregation for graph nets," 2020. [Online]. Available: <https://arxiv.org/abs/2004.05718>
- [46] C. Diao and R. Loynd, "Relational attention: Generalizing transformers for graph-structured tasks," in *The Eleventh International Conference on Learning Representations*, 2023. [Online]. Available: <https://openreview.net/forum?id=cFuMmbWiN6>
- [47] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," *Master's thesis*, 2009.
- [48] A. Coates, A. Ng, and H. Lee, "An analysis of single-layer networks in unsupervised feature learning," in *AISTATS*, 2011.
- [49] H. Xiao, K. Rasul, and R. Vollgraf. (2017) Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms.
- [50] T. Clanuwat, M. Bober-Irizar, A. Kitamoto, A. Lamb, K. Yamamoto, and D. Ha. (2018) Deep learning for classical japanese literature.
- [51] J. Hull, "A database for handwritten text recognition research," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 5, pp. 550–554, 1994.
- [52] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, "Reading digits in natural images with unsupervised feature learning," in *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011.
- [53] R. Wightman, "Pytorch image models," <https://github.com/rwightman/pytorch-image-models>, 2019.

APPENDIX

This appendix consist of details left out in the main text. First, we perform a more comprehensive review of the related literature with further discussion of our motivations. Next, we go over the various components of FLoWN and expound on their implementation and training procedure.

A. Related Works

a) Conditional flow matching: The CFM objective, where a conditional vector field is regressed to learn probability paths from a source to target distribution, was first introduced in Lipman et al. [2]. The CFM objective attempts to minimize the expected squared loss of a target conditional vector field (which is conditioned on training data and generates a desired probability path) and an unconditional neural network. The authors showed that optimizing the CFM objective is equivalent to optimizing the unconditional FM objective. Moreover, the further work [4] highlighted that certain choices of parameters for the probability paths led to the optimal conditional flow being equivalent to the optimal transport path between the initial and target data distributions, thus resulting in shorter inference times. However, the original formulations of flow matching assumed that the initial distributions were Gaussian. Pooladian et al. [17] extended the theory to arbitrary source distributions using minibatch sampling and proved a bound on the variance of the gradient of the objective. Tong et al. [4] showed that using the 2-Wasserstein optimal transport map as the joint probability distribution of the initial and target data along with straight conditional probability paths results in a marginal vector field that solves the dynamical optimal transport problem between the initial and target distributions.

b) Neural network parameter generation: Due to the flexibility of neural network as function approximators, it is natural to think that they could be applied to neural network weights. Denil et al. [18] paved the way for this exploration as their work provided evidence of the redundancy of most network parameterizations, hence showing that paramter generation is a feasible objective. Later, Ha et al. [19] introduced Hypernetworks which use embeddings of weights of neural network layers to generate new weights and apply their approach to dynamic weight generation of RNNs and LSTMs. A significant portion of our paper’s unconditional parameter generation section builds upon the ideas from Wang et al. [12] and the concurrent work of Soro et al. [10] where the authors employ a latent diffusion model to generate new parameters for trained image classification networks.

c) Meta-learning context: Although neural networks are adept at tasks on which they were trained, a common struggle of networks is generalization to unseen tasks. In contrast, humans can often learn new tasks when given only a few examples. A pioneering modern work in this field is MAML [38], which learns good initialization parameters for the meta-learner such that it can easily be fine-tuned to new tasks. Their approach utilizes two nested training loops. The inner loop computes separate parameters adapted to each of the training tasks. The outer loop computes the loss using each of these

parameters on their respective tasks and updates the model’s parameters through gradient descent. However, MAML often had unstable training runs, and so successive works gradually refined the method [39]–[42]. The aforementioned works typically focus on classification tasks, however, this paradigm allows for great versatility. For instance, Beck et al. [43] used hypernetworks to generate the parameters of a policy model and Lee et al. [44] exploited the in-context learning ability of transformers to general reinforcement learning tasks.

d) Weight generation for few-shot learning: Following up on the work of meta-learning context, few-shot learning is a natural application of such meta-learning algorithms. An early example is Ravi & Larochelle [20] who designed a meta-learner based on the computations in an LSTM cell. At each training example in the support set, their meta learner uses the losses and the gradients of the losses of the base learner (in addition to other information from previous training examples) to produce base learner parameters for the next training example. The loss of the base learner on the test examples in the support set is backpropagated through the meta learner’s parameters. Moreover, we may leverage the advancements in generative modeling for weight generation. As we mentioned, Lee et al. [44] used transformers for in-context reinforcement learning, but we also see the works of Zhmoginov et al. [15]; Hu et al. [14]; Kirsch et al. [21]; Fifty et al. [13] use transformers and foundation models. More similar to our method is the body of work on using diffusion models for weight generation [10]–[12], [22]. These methods vary in their approach, some leveraging a relationship between the gradient descent algorithm and the denoising step in diffusion models to design their meta-learning algorithm. Others rely on the modeling capabilities of conditioned latent diffusion models to learn the target distribution of weights. Most evaluations conducted were in-distribution tasks, i.e. tasks sampled from the same data distribution as the training tasks, hence, there is room to explore ways of adapting this approach for out-of-distribution tasks.

B. Architecture Details

Here, we expound on the architecture of FLoWN. See Figure 1 for a schematic of the training and inference process.

1) Variational Autoencoder: The variational autoencoder follows the implementation of Soro et al. [10]. In particular, given a set of model weights $\{\mathcal{M}_i\}_{i=1}^N$, we first flatten the weights to obtain vectors $w_i \in \mathbb{R}^{d_i}$. For the sake of uniformity, we always zero-pad vectors to $d = \max_i d_i$. Alternatively, we allow for layer-wise vectorization: set a chunk size ℓ which corresponds to the weight dimension of a network layer. Then, zero-pad w_i to be a multiple of ℓ , say \tilde{d} . This allows us to partition into k equal length vectors $w_{i,k} \in \mathbb{R}^{\tilde{d}/k}$. Typically, larger models benefit from layer-wise vectorization.

Subsequently, we train a VAE to obtain an embedding of such vectors by optimizing the objective:

$$L_{VAE}(\theta, \phi) := -\mathbb{E}_{z \sim q(z|w)} [\log p_\theta(w|z) + \beta D_{KL}(q_\phi(z|w) || p(z))] \quad (2)$$

where \mathbf{w} is the vectorized weights, \mathbf{z} is the embedding we are learning, and p_θ, q_ϕ are the reconstruction and posterior distributions respectively. Moreover, we fix the prior $p(\mathbf{z})$ to be a Gaussian and the weight is set to be $\beta = 10^{-2}$. For layer-wise vectorization, we simply change the input dimensions to match the chunk size. Upon decoding, we concatenate the chunks to re-form the weight vector.

2) *Graph Encoder*: Recently, Kofinas et al. [28] proposed a neural graph encoder which incorporates the permutation invariance present in network weights. The method has two components: a graph constructor and the embedding model. The neural network is first represented as a graph where nodes represent the neurons within each network layer and edges represent neuronal connections. Importantly, node features correspond to bias parameters and edge features correspond to weight parameters. Subsequently, this is fed into an embedding model, such as a GNN, specifically PNA [45], or a relational transformer [46]. For our use case, we outline weights-to-graphs conversion of MLPs, CNNs, and normalization layers. See Kofinas et al. [28] for more details.

a) *MLPs to graphs*: Let $\mathcal{G}(\mathbf{V}, \mathbf{E})$ be a graph and let the vertex set $\mathbf{V} \in \mathbb{R}^{n \times d_V}$ and the adjacency matrix $\mathbf{E} \in \mathbb{R}^{n \times n \times d_E}$. Intuitively, if we have n nodes in a graph, our vertex set is size n , and the adjacency matrix is $n \times n$. In our case, we also incorporate node and edge features, hence an extra dimension is added. Consider an L -layer MLP with weights $\{\mathbf{W}^\ell \in \mathbb{R}^{d_\ell \times d_{\ell-1}}\}_{\ell=1}^L$ and biases $\{\mathbf{b}^\ell \in \mathbb{R}^{d_\ell}\}_{\ell=1}^L$. Since we have a node for each neuron, we have $n = \sum_{\ell=0}^L d_\ell$, where d_0 is the input dimension. Now, let's use these to construct the vertex set \mathbf{V} . Since each neuron has a corresponding bias term (except the input), $\mathbf{V} = [\mathbf{0}_{d_0} \ \mathbf{b}^1 \ \dots \ \mathbf{b}^L]^\top$. As for the adjacency matrix, consider the first d_0 rows: as this corresponds to the input layer, it's only connected to the first layer, i.e. only columns $d_0 + 1$ to $d_0 + d_1$ are possibly non-zero. And if we focus on row $i \in [d_0]$, what are its features? They must be $\mathbf{W}_{:,i}^1$. Hence,

$$(\mathbf{E}_{[0:d_0] \times [d_0+1, d_0+d_1]})^\top = \mathbf{W}^1,$$

and elsewhere in $\mathbf{E}_{[0:d_0]}$ is zero. In general,

$$(\mathbf{E}_{[d_{i-1}:d_i] \times [d_{i-1}+1, d_{i-1}+d_i]})^\top = \mathbf{W}^i,$$

and is zero everywhere else. In other words, the first off-diagonal blocks are precisely \mathbf{W}^i , and \mathbf{E} is zero elsewhere. Finally, what are d_E and d_V ? This turns out to be problem-dependent. Sometimes, it helps to add useful node features, but if the only thing we are concerned about embedding is weight information, then each entry of \mathbf{W}^i and \mathbf{b}^i is simply a scalar, so $d_E = d_V = 1$.

b) *Normalization layers to graphs*: Either BatchNorm or LayerNorm can be written as $\mathbf{y} = \mathbf{m} \odot \mathbf{x} + \mathbf{b}$, where $\mathbf{m}, \mathbf{x}, \mathbf{b}, \mathbf{y} \in \mathbb{R}^d$. The trick is to recast this as a linear layer: we can always write $\mathbf{y} = \text{diag}(\mathbf{m})\mathbf{x} + \mathbf{b}$. Hence, we ought to have d nodes for \mathbf{x} and another d nodes for \mathbf{y} where the nodes for \mathbf{y} have biases \mathbf{b} . The two layers are then connected by weight matrix $\text{diag}(\mathbf{m})$ which only connects x_i to y_i .

TABLE VI

MODEL ARCHITECTURES AND HYPERPARAMETERS. SQUARE BRACKETS $[\cdot]$ INDICATES AN INTERVAL OF VALUES. FOR INSTANCE, WE OFTEN TRAIN UNTIL A LOSS PLATEAU, HENCE THE VARYING NUMBER OF EPOCHS.

Parameters	Model Retrieval Few-Shot Learning	
	Dataset Encoder (Frozen)	
Architecture	CAML	CAML
Latent Dimension	1024	1024
	Weight Encoder	
Architecture	VAE	VAE
Latent Space Size	$4 \times 4 \times 4$	$4 \times 8 \times 8$
Upsampling/Downsampling Layers	5	4
Channel Multiplication (per Downsampling Layer)	(1, 1, 2, 2, 2)	(1,1,2,2)
ResNet Blocks (per Layer)	2	2
KL-Divergence Weight	0.01	$1e-6$
Optimizer	AdamW	AdamW
Learning Rate	1×10^{-3}	1×10^{-2}
Weight Decay	2×10^{-6}	2×10^{-6}
Batch Size	32	128
Training Epochs	3000	[100, 500]
	Conditional Flow Matching Model	
Timestep and Dataset Embedding Size	128	128
Input Size	$4 \times 4 \times 4$	$4 \times 8 \times 8$
Optimizer	AdamW	AdamW
Learning Rate	1×10^{-3}	2×10^{-4}
Weight Decay	2×10^{-6}	2×10^{-6}
Batch Size	32	128
Training Epochs	[3000, 10000]	[100, 500]

c) *CNNs to graphs*: To simplify consider one convolutional layer between layers $\ell - 1$ and ℓ , namely $\mathbf{W} \in \mathbb{R}^{d_\ell \times d_{\ell-1} \times w_\ell \times h_\ell}$ and $\mathbf{b} \in \mathbb{R}^{d_\ell}$. Intuitively, $d_{\ell-1}$ is the number of input channels and d_ℓ the number of output channels. Due to the spatial dimension $w_\ell \times h_\ell$, we first flatten the last two layers. Now, we make use of the node and edge features: instead of scalar weights like in linear layers, our weights are vectors of size $w_\ell \times h_\ell$. However, the size may be different between layers, so we take $s = (\max_{\ell \in [L]} w_\ell, \max_{\ell \in [L]} h_\ell)$ and zero-pad our weight vectors as necessary before flattening. Hence, following the procedure in the MLP conversion, we form an adjacency matrix with vector features, i.e. $\mathbf{E} \in \mathbb{R}^{n \times n \times d_E}$ where $d_E = w_{\max} h_{\max}$.

3) *Flow Model*: The neural network used for flow matching is the UNet from D2NWG [10]. The specific hyperparameters used for the CFM model varies between experiments, so we leave this discussion to D.

C. Training Details

Here, we present further training and experimental details.

1) Pre-trained Model Acquisition:

a) *Datasets and architectures*: We conduct experiments on a wide range of datasets, including CIFAR-10/100 [47], STL-10 [48], (Fashion/K)-MNIST [49], [50], USPS [51], and SVHN [52]. To evaluate our meta-model's ability to generate new subsets of network parameters, we conduct experiments on ResNet-18 [31], ViT-Base [32], ConvNeXt-Tiny [33], the latter two are sourced from timm Wightman [53]. As we shall detail below, small CNN architectures from a model zoo [30] are also used for full-model generations.

TABLE VII
PREPROCESSING AND GRAPH ENCODER HYPERPARAMETERS.

Parameters	Values
Dataset Preprocessing	
Input Channels	3
Image shape (w_{\max}, h_{\max})	(32, 32) (7, 7)
Max. spatial res.	49
Max. # hidden layers	5
Flattening Method	Repeat Nodes
Normalize	False
Augmentation	False
Linear as Conv.	False
Relational Transformer	
Embed dim.	64
Num. layers	4
Num. heads	8
Num. probe features	0

b) *Model pre-training*: For better control over the target distribution p_1 , in experiments involving ResNet-18, ViT-Base, and ConvNeXt-Tiny, we pre-train these base models from scratch on their respective datasets. We follow Wang et al. [12] and train the base models until their accuracy stabilizes. Further, we train the relevant subset (e.g. batch norm parameters for ResNet-18) for another 200 epochs, saving the weights at the end.

c) *Model zoo*: The model zoo used for meta-training in the model retrieval setting, as described in Sec. III-A0c, was sourced from [30]. As the base model, we employed their CNN-small architecture, which consists of three convolutional layers and contains either 2,464 or 2,864 parameters, depending on the number of input channels. For each dataset—MNIST, Fashion-MNIST, CIFAR-10, and STL10—100 sets of pre-trained weights were randomly selected from the model zoo using different seeds and fixed hyperparameters (referred to as “Seed” in their codebase). For the training of base models, we adopted the same hyperparameters as those used in [30] for all datasets, except KMNIST, which was not included in their model zoo. For KMNIST, we used the hyperparameters applied to MNIST, given the similarity between the two datasets.

2) *Variational Autoencoder Training*: The VAE was trained with the objective in equation 2. Moreover, following p-diff [12], we add Gaussian noise to the input and latent vector, i.e. given noise factors σ_{in} and σ_{lat} with encoder f_ϕ and decoder f_θ , we instead have

$$\mathbf{z} = f_\phi(\mathbf{w} + \xi_{in}), \hat{\mathbf{w}} = f_\theta(\mathbf{z} + \xi_{lat})$$

where $\xi_{in} \sim \mathcal{N}(0, \sigma_{in}^2 \mathbf{I})$, $\xi_{lat} \sim \mathcal{N}(0, \sigma_{lat}^2 \mathbf{I})$.

A new VAE is trained at every instantiation of the CFM model as architectures often differ in their input dimension for different experiments. However, they are trained with different objectives: the VAE is trained to minimize reconstruction loss. In all experiments, we fix $\sigma_{in} = 0.001$ and $\sigma_{lat} = 0.5$.

TABLE VIII
TASK TRAINING

Parameters	ResNet18	ViT & ConvNext	CNN
Optimizer	SGD	AdamW	AdamW
Initial Training LR	0.1	1×10^{-4}	3×10^{-3}
Training Scheduler	MultiStepLR	CosineAnnealingLR	CosineAnnealingLR
Layer Weights Saved	Last 2 BN layers	Last 2 BN layers	All layers
Initial Model Saving LR	1.6×10^{-4}	5×10^{-2}	1×10^{-3}
Model Saving Scheduler	None	CosineAnnealingLR	CosineAnnealingLR
Number of Models Saved	200	200	200
Num. of Weights per Model	2048	3072	[10565, 12743]
Training Epochs	100	100	100
Batch Size	64	128	128

3) *Graph Encoder Training*: The graph encoder [28] was used for both the unconditional generation and fine-tuning on OOD experiments with the CNN-medium architecture from Schürholt et al. [30]. We restricted our tests to the relational transformer [46] which was shown to perform better in the original paper [28]. See Table VII for the instantiation parameters.

D. Experimental Details

1) *Unconditional Generation*: Unconditional generation involves two stages: first is the training of base models. We choose a Resnet18, ViT-B, ConvNext-tiny, and medium-CNN for our base models and provide the training parameters in Table VIII. Next, is the stage where we train either a AE-CFM or AE-DDPM, with the encoder being the same in both cases; the training parameters for this stage is provided in Table IX.

2) *Model Retrieval*: The first column of Table VI shows the details of the model architectures and training configurations. For each dataset, we first generate its CAML embedding by (1) averaging the query image embeddings within each class to get class embeddings, (2) concatenating the class embeddings into one long vector, (3) passing the combined class embeddings through two linear layers to produce the final dataset embedding. Next, the dataset embedding is combined with the timestep embedding via a projection layer, and the resulting representation is used as input to the flow matching model.

a) *A mini-Imagenet prior for CFM*: As mentioned in Sec. III-A0c, we attempted this experiment with priors from a pre-trained mini-Imagenet. There were a few technical hurdles with the implantation of these weights. First, for 1-channel datasets such as MNIST, the input weight shapes are smaller than those of the mini-Imagenet model. Second, the classification head of a mini-Imagenet model predicts a much greater number of classes than our test datasets. Our procedure is as follows: we train a small-CNN model [30] on mini-Imagenet until its accuracy stabilizes. Next, we take the mean μ and standard deviation σ of its classifier head. Using these statistics, we initialized the classifier heads of our base models as $\mathcal{N}(\mu, \sigma^2 \mathbf{I})$. For the rest of the base model, we pad to the prior’s shape if necessary, and we implant the pre-trained weights directly before adding Gaussian $\mathcal{N}(\mathbf{0}, \mathbf{I})$

TABLE IX
CFM/DDPM

Parameters	AE CFM	AE DDPM
Flow/Diffusion Optimizer	AdamW	AdamW
Flow/Diffusion LR	0.001	0.001
Autoencoder Optimizer	AdamW	AdamW
Num Inference Timesteps	100	[100, 1000]
Autoencoder LR	0.001	0.001
Weight Initialization	Kaiming	Normal
Autoencoder Epochs	[1000, 30000]	[1000, 30000]
CFM/DDPM Epochs	[1000, 30000]	[1000, 30000]
Batch Size	[50, 200]	[50, 200]

noise. Since we flow in latent space, our last step is to apply the VAE to the weights we've constructed.

Figure 2 shows the training curve with our mini-Imagenet prior in blue, and with a Gaussian 0-1 prior in orange. It is striking that the loss decreases much faster, but as seen in Table II, the test accuracies are quite poor. This points to an issue such as overfitting, which is likely caused by latent space capacity. Indeed, with our approach of constructing the prior, we invoke the VAE encoder twice: once to encode the prior and once more to encode the target weights. The target weights were those pre-trained on one of the four datasets in Table II, hence it's expected that the distribution is quite distinct from those pre-trained on mini-Imagenet. Due to the size of our latent space (64, as noted in Table VI), it may be insufficient to encode both distributions. Moreover, the loss objective for encoding the prior is not ideal. As the encoder is invoked in forward passes of the CFM, it only learns how to encode the prior such that CFM loss decreases, as opposed to a reconstruction objective. Hence, future work could look to modify encoder training so as to reconstruct both target weights and weights of the prior.

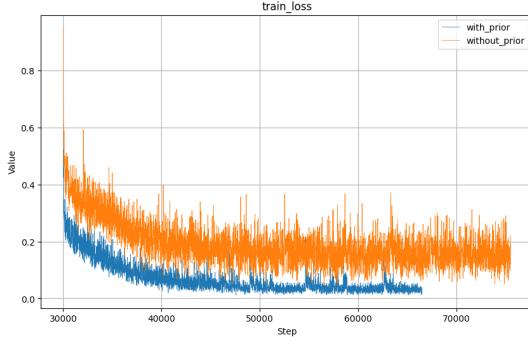


Fig. 2. The training loss curve for the mini-Imagenet run from Table II.

3) *Few-Shot Learning*: For the few-shot learning experiments, we adopted the same methodology for obtaining dataset embeddings and conditioning parameter generation as in the model retrieval experiment. Classifier heads were trained on 50,000 randomly sampled 5-way 1-shot subsets of the mini-ImageNet dataset [34], and the resulting pre-trained weights were used as training data for the meta-model. The hyperparameter configurations for meta-training are provided

TABLE X
FINETUNING

Parameters	ResNet18	CNN
Optimizer	AdamW	AdamW
Num Epochs	[50,100]	[50, 100]
Initial LR	1×10^{-2}	1×10^{-5}
Detach Value	0.4	0.4
LR Scheduler	None	CosineAnnealingLR
Minimum LR	1×10^{-2}	5×10^{-7}

in the second column of Table VI. During evaluation, we measured the accuracies of the generated models on two out-of-distribution datasets, CIFAR-10 and STL-10. For each subset, we sampled 50 sets of weights and reported the average accuracy of the top three performing models.

4) *Fine-tuning on Out-of-Distribution Data*: We provide the hyperparameters of the fine-tuning experiment in Table X.

Generative Music AI's \$350 Million Problem: Compensating Creators for the Use of Copyrighted Materials in Training Sets

Josh Wagman
Queen's University
21jdw14@queensu.ca

Kay Yan
Queen's University
k.yan@queensu.ca

Rafael Costa
Queen's University
22xwyk@queensu.ca

Alex Levesque
Queen's University
Alex.levesque@queensu.ca

Armita Afroushe
Queen's University
24kb11@queensu.ca

Abstract—The rapid expansion of music AI technologies has led to the extensive use of large-scale datasets that often include copyrighted music without adequate oversight. Current legal and technical frameworks struggle to identify and quantify such copyrighted content, resulting in the under-compensation of copyright holders and potential violations of intellectual property rights. This study implements a unique approach to copyright detection. Utilizing federated learning (FL), our method trains models locally, preserving data privacy by keeping sensitive information on local servers while aggregating model updates centrally. Additionally, model fingerprinting assigns unique digital signatures to training data outputs, enabling precise tracking and verification of copyrighted material. Leveraging these techniques, our framework compiles a comprehensive catalog of artists and quantifies the number of songs present in the dataset, which is then integrated into our compensation mechanism to ensure fair remuneration for copyright holders. Our solution enhances transparency in data usage while delivering mutual benefits for both AI developers and creators, incentivizing a cooperative musical landscape where AI and creativity coexist.

I. INTRODUCTION

In April 2023, an unknown Tik Tok user called Ghostwriter977 released a song on Spotify and Apple Music called “Heart on My Sleeve” that would greatly influence the music industry. Generative music AI claimed the spotlight in music innovation with the release of this song, featuring Drake and The Weeknd. The only complication, however, is that neither Drake nor The Weeknd ever sang a single note for this track. This was one of the first documented instances of generative music AI being used to create music that became a major worldwide hit, and many more have come since. The AI was trained on copyrighted music, and the artists and their record label, Universal Music Group (UMG) were never compensated for the use of their music in the training of this AI. AI creators and artists in music, visual arts, and other fields face the challenge of insufficient copyright laws governing AI. This lack of law and policy leads to intense legal battles, such is the case in Suno & Udio Vs. UMG, Warner Records, and Sony Records. The use of copyrighted music to train generative

music AI by Suno and Udio, while not technically against any specific laws, has led three of the world’s largest music licensers to sue them. If this discrepancy is not fixed, then those working in creative professions will continue to have their works used without their consent to train AI. This will inevitably lead to a loss of jobs due to this software, the program that was trained on their own works.

With the recent developments in AI, law and policy fall further behind, AI companies seem to be able to skirt the law for their own personal, monetary gain. The datasets are not monitored and are kept private by most AI companies, meaning that there is no way for the government or creative industries to get a hold of the datasets without legal action. On top of this, the creative professionals and those who own the works are not getting compensated for the use of their works. AI developers need huge amounts of data in order to properly train their AI, and this often leads to the usage of copyrighted works. If one AI company is using copyrighted works, then would it put the other AI company at a disadvantage to not do the same, especially when there is no specific law against this use? AI is a rapidly developing field, and every company is striving for the highest quality product to offer their users.

Both sides have an argument to be made, and until legal precedent is set, these two industries are poised to fight against each other. The questions this paper aims to answer are: How should copyright law be adapted to fit generative AI and should artists and record labels be compensated for the use of their works in the training of generative AI?

II. BACKGROUND INFORMATION

“Heart on My Sleeve” was not the only case of AI-generated songs making their way into the public music scene [1]. UMG and other major record labels were starting to catch on that these AI used thousands of copyrighted music to train them [2]. This act, they claim, constitutes “copyright infringement on an almost unimaginable scale” [3]. To understand this claim, we will provide you with background context into the

training and usage of generative music AI software, copyright law, and the current legal landscape of relevant cases.

A. How Generative Music AI is Trained

Generative music AI begins by collecting datasets of musical elements, such as chords, melodies, rhythms, and timbres. These datasets often consist of music files in formats like MIDI or audio recordings in WAV or MP3. MIDI files are widely used due to their ability to represent music as a sequence of notes with information about pitch and velocity, which are easier for AI models to analyze and learn from [4].

The preprocessing stage of generative music AI involves converting audio waveforms into spectrograms. These graphs visually represent the frequency content of sound over time and almost act like a “map” that AI can use to analyze patterns or specific frequencies. Commonly, Short-Time Fourier Transforms (STFT) are used to convert audio waveforms into a time-frequency representation. As a result, the spectrogram’s x-axis represents time, the y-axis represents frequency, and the color intensity represents amplitude, as seen in 3. For example, a bass-heavy song with deep male voices will have stronger signals in the lower-frequency range [5].

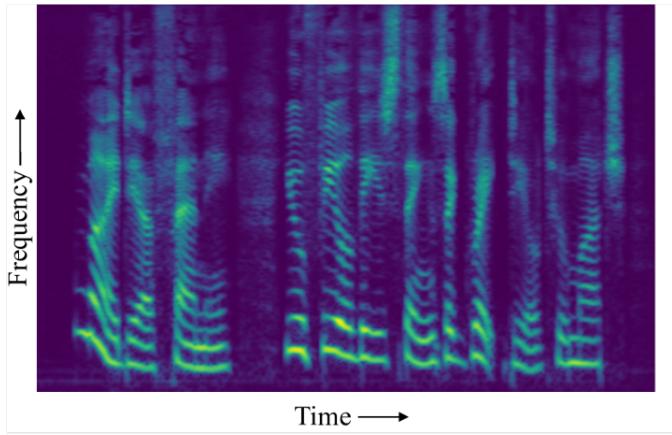


Fig. 1. Spectrogram Representation of a Four-Second Audio Signal [5]

Then, these spectrograms are used as input to convolutional neural networks (CNNs), which are specialized for recognizing spatial patterns in visual data. As spectrograms pass through a CNN, they are compressed into lower-dimensional feature vectors that represent musical characteristics like chord progressions and melodic motifs. Thus, instead of processing the entire spectrogram at once, CNNs process it frame-by-frame, generating sequential data corresponding to different time steps. These vectors retain essential information while reducing complexity, making them more suitable for training models that require sequential inputs.

Once the model is trained, the AI generates new music using algorithms that are based on the patterns it learned. Algorithms are essentially sets of rules or instructions that the AI follows to perform tasks. Once the music is made, the output can then be converted into formats for playback such as WAV or MP3.

B. Legal Landscape

The current legal landscape of AI is extremely volatile and still developing. Current cases on the court docket are bound to shape the future of AI policy by setting important precedent. The current legal system is trying to determine how to adapt current copyright laws and policies to AI systems, specifically generative AI systems. To first understand the legal landscape, we will introduce you to the type of law that this paper deals with: copyright law.

1) **Copyright Law:** Copyright law exists to protect creative works around the world, but what exactly is defined under the scope of “creative work”? Creative works aim to define the expression of creative ideas but not the systems/processes used to derive these ideas: “Copyright protects expression, and never ideas, procedures, methods, systems, processes, concepts, principles, or discoveries” [6]. As such, copyright protects an instance of a creative work, for example, a song, painting, or distinct brand logos, but never to protect a process of creating a work. For example, Lord of the Rings is a copyrighted work by author J.R.R. Tolkien, however the brainstorming methods and writing techniques that Tolkien used are not protected under copyright. There are exemptions in which copyrighted works are allowed to be used without expressed consent from the copyright owner, and this is a staple of copyright law known as the Fair Use Doctrine. The Fair Use Doctrine promotes freedom of expression by allowing the use of copyrighted materials under specific circumstances. Copyrighted works may be used without consent in cases of criticism, teaching, research, and news reporting, to name a few [7]. The criteria in which Fair Use is evaluated are:

1. **Purpose and usage of the copyrighted work:** Copyrighted works used in a non-profit educational and non-commercial manner are much more likely to be classified as fair. Similarly, transformative uses (adding something new or changing something with a further purpose) are more likely to be classified as fair [7].
2. **Nature of copyrighted work:** Many types of works can be copyrighted, each with varying levels of creativity. Using works that are considered more creative such as songs, movies, and art are less likely to be classified as fair than factual work, as this relates to the premise of encouraging creative expression in copyright law [7].
3. **Proportional amount of copyright used:** If the use of copyright materials is found to have copied a large portion of the copyrighted work, then it is less likely to be classified as fair [7].
4. **Effect of the use upon the potential market for or value of the copyrighted work:** If the use of copyrighted works is competing in the same market as the original work, then courts must consider the potential effect to the market upon widespread use of the created work.

A key distinction to be made in the writing of these metrics is that every use must be evaluated on a case-to-case basis. The wording implies that even though specific uses are “more likely” or “less likely” to be considered fair use, each case is unique and requires specific interpretation of the law [7].

A central challenge in the current system is that standardized licensing agreements for AI-generated music is lacking. There is no clear framework for licensing rights, which given AI creators and user challenges in legally using or distributing AI-generated works. To add to that, the application of fair use to generative AI music is uncertain, especially regarding whether AI-generated compositions are “transformative” enough to qualify as fair use. Given that these models often are trained with extensive use of copyrighted materials, courts may have difficulty consistently applying fair use to these cases, as we will show in the two cases coming up: Suno-Udio and Stability AI [5] [8].

2) *Relevant Cases:* The legal landscape is bound to shape the future of generative music AI. As technology makes further progress, policy struggles to catch up. Current court cases will ultimately determine the future of the generative AI field, and as such, it is vital to understand past and present court proceedings. In this section, we aim to present what we believe to be the two most important cases, ongoing and present to paint a picture of the current legal landscape surrounding generative AI.

Case 1: Suno and Udio v. UMG, Sony Music Entertainment, Warner Records, et al. (2024)

The most relevant case to the discussion of copyright law regarding generative AI music is an ongoing case, between generative AI music companies Suno and Udio, and record label titans Universal, Warner, and Sony. First filed on June 24th, 2024, the three labels filed federal lawsuits in New York and Massachusetts against the two startups. They allege that Suno and Udio were involved in mass copyright infringement by using popular songs, such as The Temptations’ “My Girl”, Mariah Carey’s “All I Want for Christmas Is You,” and James Brown’s “I Got You (I Feel Good)”, to train their AI models. These models would then be able to create music on demand and can mimic iconic artists such as Michael Jackson, Bruce Springsteen, and ABBA [1].

The defense of the startups was that they allege that their AI training systems fall under “fair use,” which would permit them limited use of copyrighted works without authorization. Suno’s CEO, Mikey Shulman, emphasized the transformative nature of their technology, arguing that it generates “completely new outputs” rather than simply “regurgitating” existing songs. They also assert that their systems analyze patterns in music, rather than memorizing specific content, which allows users to create original music based on text prompts [9].

The record labels on the other hand, argue that this use of their music is unlicensed and amounts to willful infringement, potentially resulting in AI-generated songs that “cheapen” the original works by offering near-identical imitations. The labels are particularly concerned about the AI’s ability to reproduce specific musical elements and even simulate artist-specific vocal styles. Mitch Glazier, CEO of the Recording Industry Association of America, criticized unlicensed services like Suno and Udio, arguing that they are exploiting artists’ work without fair compensation, which he argues could hinder

genuine innovation in AI [9].

The labels are seeking statutory damages of up to \$150,000 for each song allegedly copied. According to the lawsuits, Suno is accused of copying 662 songs in training its AI model, while Udio allegedly used 1,670. This totals to a lawsuit of just under \$350 million in damages. The labels also demand full disclosure of the training datasets used by the companies, accusing them of being “deliberately evasive” about the material, which, if revealed, they argue could constitute “willful copyright infringement on an almost unimaginable scale” [9].

Case 2: Getty Images v. Stability AI (2023)

Another case of importance in the realm of copyright law is the case between the stock photo provider Getty Images, and the creator of the AI model Stable Diffusion, Stability AI. Filed in February 2023 in Delaware, Getty Images alleges that Stability AI used more than 12 million of its copyrighted images to train Stable Diffusion without a license, which they argue constitutes copyright infringement. Stability AI allegedly did not seek or obtain a license to use these images, which Getty claims could have been acquired under established licensing agreements, as has been done by other technology companies [10].

Moreover, Stability’s model sometimes generated images displaying Getty’s watermark, which Getty argues could lead to consumer confusion and devalue its brand. This has prompted Getty to include watermark infringement alongside its copyright allegations. Getty is seeking both financial damages, including Stability’s profits from the alleged infringement, and an injunction to stop Stability AI from using its images.

This case also provides more critical legal questions about whether AI companies need explicit licensing to use copyrighted material for training, especially as these companies compete with traditional creative industries. Getty’s lawsuit also brings up risks of AI-generated content displaying watermarks, which is applicable to generative AI in music if these AI tracks are associated with specific artists or labels [10].

III. KEY ISSUES AND CHALLENGES

The key issues and challenges in adapting copyright law to AI stem from an outdated legal framework that wasn’t designed for machine-generated content and the rapid pace of technological change. Current laws struggle with defining authorship, determining originality, and reconciling the use of vast, often copyrighted datasets by AI developers. This section explores how these legal uncertainties intersect with the need to fairly compensate artists and protect intellectual property while still encouraging innovation in the AI space.

A. Adapting Copyright Law for AI

The complexity of AI and its uses have led to a complex implementation of copyright law in AI. At this time, judicial systems around the world struggle to adapt copyright laws to AI. Legal frameworks must adapt to these new technologies,

but they are struggling to address key topics such as authorship and originality. These topics cover questions such as:

1. Who is considered to be the author when AI creates content – the developer, the user, or the AI itself?
2. Traditional copyright laws require artists to use a degree of human creativity when creating a new product. Do AI-generated outputs meet that standard of creativity?

Policy has fallen behind the rapid development of AI systems and will likely continue to do so. If lawyers and litigators were able to predict how AI may evolve, they could pre-emptively address these solutions, allowing policy to have more dictation over how AI systems are created.

To address the complexities of AI-generated works, new laws must be introduced to handle this unfamiliar territory. There are many different mechanisms into which laws could be adapted, all of which come with their own challenges in implementation. The first of which is an AI Transparency and Attribution framework, where the disclosure of dataset sources and attribution for AI-generated outputs resembles those for specific copyrighted works. However, the issue arises while balancing transparency with protecting AI trade secrets, something which current AI companies are trying their best to maintain private.

Another option is introducing levy-based fees, where a levy is applied on AI tools or datasets with the goal of creating a compensation pool for copyright holders. This option also comes with some limitations as fair distribution among creators is challenging to determine, and AI developers may resist due to increased costs on their operations.

The third is to create licensing agreements where AI developers would need to obtain licenses for copyrighted works in training datasets. These could adopt collective licensing models similar to those for radio and streaming. This idea also has its downsides as these agreements could lead to complex negotiations, particularly for smaller developers. Alongside that issue, high licensing costs could help stifle AI innovation as higher operating costs due to these licenses could prevent AI companies from growing.

B. Compensating Artists

The recent spike in integrating generative AI to music production has presented many issues in compensating artists whose works are used as training data for these models. The primary concern is the lack of legal frameworks that would mandate compensation for artists when their music is used to train AI. This ambiguity allows AI companies to use copyrighted songs without proper licenses to develop their software. This has led to major record labels like UMG and Warner to file lawsuits against AI startups such as Suno and Udio, alleging illegal use of their music libraries for AI training [11].

However, implementing a fair compensation model faces several challenges and a fundamental paradox. For artists to be compensated, AI training data must be public and transparent, yet full transparency is impractical because of privacy risks and societal concerns. If datasets were to remain closed, artists

cannot verify if their work has been used. If datasets were fully disclosed, this could pose significant societal risks. Publicly available datasets might enable the copying of AI models, leading to intellectual property theft. Too much transparency would also slow innovation and hurt the generative music AI industry by making it harder for companies to attract investment and stay competitive. This could also hurt their growth, as competitors could use their datasets, making it harder for companies to stand out in this niche market. This paradox reveals a no-win situation of developers, artists, and policymakers. Either prioritize accountability and fair compensation while putting innovation and security in jeopardy or protect AI companies and their datasets and prevent misuse at the cost of artists' rights. This dilemma underlines the need for a middle ground where artists can be compensated while AI can stay protected.

On the other hand, AI companies resist compensating artists by arguing that their use of copyrighted training data is transformative and is under the doctrine of fair use. For instance, companies Suno and Udio claim that their AI does not copy material but analyzes them for patterns. Some argue that fair use promotes freedom of expression by allowing companies and people unauthorized use of copyright-protected works under certain circumstances, with the main focus falling on transformative works [3]. Fair use is often a legal gray area, as each case is unique in its usages of copyrighted materials and thus, considerations change. This defense allows AI companies to justify their practices on uncharted ground, without knowing with certainty if what they are doing is legal or not.

C. Dataset Transparency

Generative AI models are generally trained on vast amounts of data. Suno has admitted to training on “tens of millions of recordings” [12]. Generally, the more data used to train the model, the better [13]. However, it is difficult to garner all the data you need while staying copyright free. This leads to a fundamental tension: AI companies need data but want to keep the challenging dynamic between transparency and protecting their competitive advantages. From a business perspective, the AI company’s datasets represent significant competitive advantages and intellectual property, including the works they use and the specific data being utilized. Revealing detailed information about training data could potentially compromise their market position or expose them to competitors and those who believe their work does not constitute fair use. Currently, some companies have taken steps toward transparency by publishing limited information about their training data or working with specific rights holders [14] [6] [7]. Most other companies, however, maintain strict privacy. To deal with this, other methods of detecting the use of copyrighted music may be a better method to deal with the situation. These ideas are discussed in more detail below.

D. Detecting the Use of Copyrighted Music

Instead of having to detect the use of copyrighted music in generate music AI, many methods revolve around protecting the original music. Audio fingerprinting adds a unique identifier to a copyrighted piece of music, very much comparable to how humans each have unique fingerprints identifying themselves. In music, these identifiers are used by services like Shazam to identify songs. In the case of generative music AI, if an original piece and its identifier are used in training data, it is possible that the fingerprint gets carried over to the newly generated music. Then, this would be an indication that copyrighted music had been used to generate music [15]. Although, if the generative music AI startups act according to their affirmations and only incorporate subtle patterns from source material, then this makes fingerprinting less effective. Similarly, a watermark code can be embedded into copyrighted music to protect it for misuse. But models can easily transform input data to obscure the watermark or even learn to ignore such codes.

Alternatively, algorithmic similarity analysis is a technique that uses machine learning to analyze patterns and any similarities between generative music AI and copyrighted songs. This method starts by extracting features from both works such as melody, harmony, and rhythm. Once extracted, features are then represented in a high-dimensional space using embeddings. Think of an embedding as a unique location or “coordinate” for each piece in a multi-dimensional space. An easy application of this topic can be made to books in a vast library that are placed across a giant map. Their “coordinates” could be defined by their characteristics, such as length of the book, writing style, language, etc., and each trait is represented numerically. Similarly, for algorithmic similarity analysis in music, their features are numerical represented as embeddings using machine learning algorithms and are then placed on this giant high-dimensional space. Finally, this would mean that music with similar features, thus similar “coordinates”, naturally cluster together. In further detail, the distance between two embeddings—such as an AI-generated song and an original piece—can be measured. Common measurements in a similarity metric are cosine similarity (which compares angles), and Euclidean distance (which measures straight-line difference). If the distance between them passes a set threshold, it may suggest that the AI-generated song was influence by or trained on the original piece. However, this method has limitations. Think of a ChatGPT detector that might detect a paragraph having a 70% chance it was generated using AI. This detector operates on probability rather than certainty. Algorithmic similarity analysis shares these philosophical issues. If a system determines there is a 70% similarity between an AI-generated song and a copyrighted song, does that draw any certain conclusions? Music inherently shares common structures, like commonly used chord progressions or similar piece structure. As a result, using this technique in lawsuits remains controversial.

IV. CASE STUDY: [SUNO V. UMG, SONY MUSIC ENTERTAINMENT, WARNER RECORDS, ET AL.]

Suno v. UMG, Sony Music Entertainment, and Warner Records stands as one of the most pivotal cases in shaping how copyright law applies to AI-generated music. First filed in mid-2024, the lawsuit revolves around allegations that Suno used copyrighted tracks to train its generative music AI without proper licensing. The dispute underscores core questions about fair use, authorship, and the extent to which AI can transform existing creative works. By analyzing both the plaintiffs' and defendants' perspectives, this case study highlights the legal complexities and potential industry-wide repercussions of AI-driven content creation.

A. Case Timeline

On June 24, 2024, the Plaintiffs, comprised of major record labels such as UMG, Sony Music, Warner Records, Capitol Records, Atlantic Records, and more, filed their initial complaint against Suno. The initial complaint included evidence that Suno was using copyrighted music, along with demands of \$150,000 per song used in the training of Suno's generative AI software [16]. On July 9, a lawyer by the name of Shlomo Fellig from the firm Latham & Watkins officially announced that he would be handling this case for Suno, and on August 1, he filed an answer to this complaint [16]. At this time, this is the most recent update in the case. It is presumed that Suno and the major record labels are in their litigation stage, trying to reach a settlement with these record labels. If this litigation is not able to reach a settlement, then this case will proceed to a jury trial. No date has currently been set for this trial at this time.

B. Plaintiff Perspective

In the ongoing dispute between Suno and major record labels, the plaintiffs argue that the AI startup relied on unlicensed, copyrighted music to train its generative model. They claim this constitutes willful infringement, depriving creators of due compensation and control over their works. This section outlines the evidence and legal rationale underpinning the labels' stance, shedding light on how they plan to prove unauthorized use of their music. As discussed in the background information section, AI companies typically do not disclose datasets, as they can be considered to be trade secrets [17]. In this section, we will detail the type of evidence that is being used against Suno and other musical generative AI companies. The following points were taken from the response filed in the District Court of Massachusetts and should be understood as the opinion of a law firm:

1. In pre-litigation correspondence, it was stated that “Suno also claimed that its large-scale copying of sound recordings is “fair use,” which was telling because fair use only arises as a defense to an otherwise unauthorized use of a copyrighted work.” [3].
2. An early investor in Suno admitted that “if [Suno] had deals with labels when this company got started, I probably

wouldn't have invested in it. I think that they needed to make this product without the constraints" [18]. The constraints, of course he is referring to are implied to be copyrighted music.

3. Using targeted prompts, the plaintiffs were able to create AI-generated songs that were almost identical in output to that of their own works. The approach was to specify key identifiers from the song such as the decade of release, topic, genre, and description of artist. An example of this is "Johnny B. Goode" by Chuck Berry (copyright owned by UMG). Suno was given the prompt "1950s rock and roll, rhythm & blues, 12 bar blues, rockabilly, energetic male vocalist, singer guitarist" and fed the lyrics for "Johnny B. Goode" [3]. The result was an output entitled "Deep down in Louisiana close to New Orle," with quite a few similarities to the original, mainly in the rhythm of the melody and key.

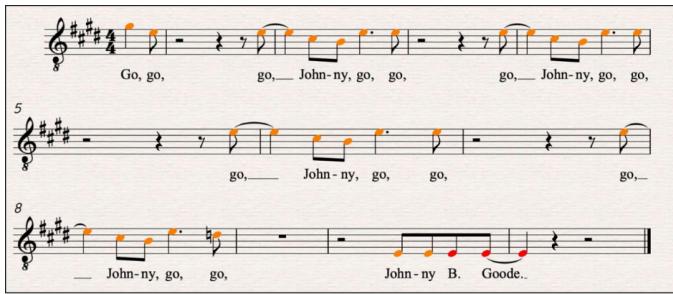


Fig. 2. Deep down in Louisiana close to New Orle (Suno generated tune) [3]

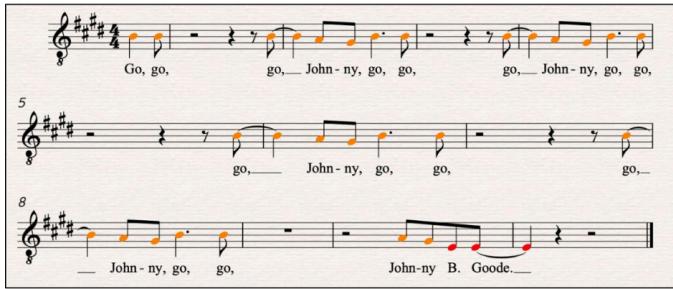


Fig. 3. Johnny B. Goode (Chuck Berry) [3]

The plaintiff was able to create 29 total responses sharing some kind of similarity (melodies in verses/choruses, rhythms, structure, etc.) to "Johnny B. Goode" using the same prompt.

4. Another common output that the plaintiff was able to generate was the AI's use of producer tags. Common in rap music, "A producer tag is a short audio clip that typically contains the producer's name or a catchphrase, used to identify their work and assert ownership over a track" [19]. One instance of the use of producer tags is in an output entitled "Rains of Castamere", which begins with the producer tag of CashMoneyAP, who is most famous for his recordings with artists Dababy and Pop Smoke.

To summarize, Suno did not disprove the claims of using copyrighted music and instead invoked "fair use," key investors have implied that they knew the AI was trained on copyrighted music, multiple songs have been recreated almost note for note through specific prompts, and producer tags from copyrighted artists are making their way into the generated songs.

As discussed in previous sections, 'fair use' has been a cornerstone of Suno's defense, as a protective measure to allow them to use copyright-protected music in AI training. The plaintiff, however, believes that "fair use" cannot be invoked in this circumstance. Here are their arguments why:

1. Suno is using copyrighted works for commercial gain. As stated previously, uses of copyrighted material for commercial gain, especially in the same market as the original, is much less likely to be considered under fair use [7].
2. The fair use doctrine describes certain use cases that can be considered fair, such as "criticism, comment, news reporting, teaching . . . scholarship, or research", however Suno does not fall under any of the listed categories, stating "Suno's service does not offer "commentary" or "scholarship" or promote human authorship" [3].
3. The use of Suno's AI is non-transformative, and the only use for this software is to generate competing music for monetary gain, "directly proportional to the number of music files it generates". Citing the fair use doctrine "If an original work and a secondary use share the same or highly similar purposes, and the secondary use is of a commercial nature, the first factor is likely to weigh against fair use, absent some other justification for copying" [3].

C. Defendant Response

On June 24th, 2024, Hueston Hennigan LLP filed a complaint against Suno, placing the company under intense legal scrutiny. In response, Suno immediately engaged Latham & Watkins, a leading law firm with a strong reputation in AI and technology litigation, to spearhead its defense. This move underscored Suno's commitment to addressing the allegations head-on while highlighting the case's potential impact on the AI music industry. Latham & Watkins quickly filed a formal answer to the complaint, setting the stage for a high-profile legal battle. The following points were taken from the response from Latham & Watkins filed in the District Court of Massachusetts, here are their main arguments against the points made in the original complaint:

1. Suno is a tool used to make new music, designed for originality, to see how people around the world can create new songs. Suno, built from extensive analysis of all genres and styles of music, intends to mimic these styles of music, not directly copy any song. The act of generating a song in a genre violates no copyright or intellectual property (IP) laws, stating "IP rights can attach to a particular recorded rendition of a song in one of those genres or styles" [20]. The act of generating a song in a genre violates no copyright or intellectual property (IP) laws, stating "IP rights can attach to a particular recorded rendition of a song in one of those genres or styles" [20].

2. The major record labels frame their concern as creating copies of pre-existing music, but what record labels are really after is to shut Suno down, effectively eliminating competition from the market. “Where Suno sees musicians, teachers, and everyday people using a new tool to create original music, the labels see a threat to their market share.” [20].

3. Suno has constructed multiple “guardrails”, specifically to ensure that no Suno generated output related too closely to a particular song used in the training process. This includes but is not limited to using industry standard software to ensure that user inputted audio clips are owned by the user, and not commercial. The software they referenced is most likely similar to Content ID or Shazam, audio fingerprinting software used by Youtube and Apple Music respectively [20].

4. “It is fair use under copyright law to make a copy of a protected work as part of a back-end technological process, invisible to the public, in the service of creating an ultimately non-infringing new product.” [20]. This statement is true; however each case is unique and must be weighed against four main factors of fair use: purpose and character of the use, nature of the copyrighted work, amount and substantiality used, and effect on the market.

These arguments aim to establish Suno as a tool used to make new music, which is not copyrightable under Fair Use and copyright laws. They claim that Suno is being targeted as “competition to the market” as these major record labels have established a monopolistic hold on the music sphere. The defendants ensure they have installed the proper protection to ensure their outputs are unique and argue that the usage of copyrighted music is protected under fair use laws as a part of a back-end technological process.

D. Looking Forward

Ultimately, what this case boils down to is an application of fair use and copyright law. Both sides have made arguments as to why fair use applies or does not in this circumstance. Due to the uniqueness of AI systems, courts will have to carefully consider the behaviour of this software, to determine its impact on the field. Copyright law protects creativity and ingenuity, so there is a main question that courts have to answer: is there a difference between an AI using a song from its dataset as inspiration versus a musician taking inspiration from an artist? The outcome of this case is set up to redefine the intersection of AI innovation and copyright law, setting critical legal precedents that will influence the entire generative music landscape. As the courts continue to discuss the applications of fair use in the context of AI training datasets, a ruling in favor of either party could catalyze significant shifts in industry practices. This will affect how data is sourced, utilized, and disclosed. This case, therefore, not only impacts the music sector but also offers an application for copyright law in AI in multiple sectors. Looking to the future, companies can mitigate legal risks and heighten ethical practices by using more transparent data practices and fair compensation

frameworks. If more companies tried to implement ethical practices, such as licensing agreements or non-copyrighted datasets, we would see a trend of less and less cases and civil suits.

V. PROPOSED SOLUTION

This section outlines our proposed solution, which is designed to safeguard the rights of copyright holders while facilitating innovation in generative music AI. It is important to note that the implementation of this solution is contingent upon the assumption that legal precedent has not yet established that the use of copyrighted works qualifies as fair use. Our approach is built on two core components: a detection system for identifying instances where copyrighted music is used during AI training, and a compensation mechanism to ensure that creators are fairly remunerated for such usage. The detection component employs advanced watermarking and fingerprinting techniques to accurately flag any unauthorized replication of original works. Meanwhile, the compensation component aims to establish a transparent, levy-based framework that directly channels revenue to the rightful copyright holders. By integrating these two components, our solution seeks to strike a balance between fostering technological innovation and upholding the integrity of intellectual property rights.

A. Detecting Copyrighted Music

1) Legal and Regulatory Compliance Rules: A solution that aims to handle a large amount of secure data from top AI companies and record labels can break laws and regulations, leading to counter suits. This section aims to address the main regulations before discussing how we plan to circumvent them. IP laws, such as the Copyright Act of Canada [21] require proof of ownership and unauthorized use to initiate legal proceedings which makes this capability especially relevant under such intellectual property laws. Eliminating the need to share raw data between companies becomes a core principle in our solution in order to comply with key IP laws. Proving “similarity” between items in datasets does not consistently hold up in a court of law as a substantial amount of proof [22]. As such, an optimal solution could implement some kind of test that does not rely solely on a similarity score, but instead can tell with absolute certainty if an item in the dataset is copyrighted. Our solution aims to establish a legal framework capable of detecting and stopping copyright infringement. These methods include:

1. Protecting personal data: Training data must stay within local boundaries to meet privacy law requirements.
2. Verifying ownership: Organizations can establish unambiguous AI model ownership proof through the use of adversarial fingerprints and distinctive watermarks.
3. Tracing infringers: Legal accountability for unauthorized AI-generated music distribution can be achieved through tracking models with embedded identifiers.
4. Enhancing copyright enforcement: Ensure AI-generated content compliance with intellectual property laws through

auditable verification processes that produce legally admissible evidence.

2) Introduction to Split Learning: We selected the split learning paradigm because it best aligns with the requirements of this compliance check. The neural network architecture divides across different parties when using split (or vertical) federated learning. The client stores the initial model components and input data while the server holds the rest of the model layers and produces the outputs. The cut layer in split learning exchanges intermediate activations ("smashed data") between parties while raw inputs and complete model parameters remain undisclosed. This setup matches our scenario: The record label (client) feeds its music data into the initial layers of a compliance-check model while the AI company (server) processes this data using its own model weights to complete the forward pass. Sensitive information remains confidential because the record label cannot access the company's model details while the company cannot access the label's raw audio.

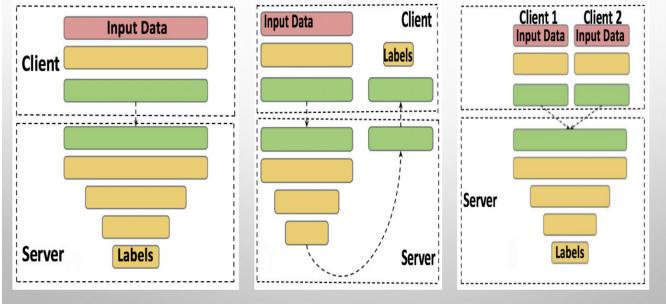


Fig. 4. Illustration of the split learning configuration [23]

For instance, in this setting, each client trains a partial model up to a specific layer called the 'cut layer'. Only the intermediate features at the cut layer (boundary between yellow and green) are sent to the server, and gradients at that cut layer are returned to the client for training. This allows joint model processing without sharing raw data.

Structure: For compliance checking, we configure a two-part neural network: The record label runs a feature extractor on its audio data locally, up to a cut-layer. The extractor could be a lightweight CNN or audio encoder that transforms the song into an intermediate feature representation. The AI company attaches a corresponding detector head (the remaining layers) that takes the intermediate feature and produces a compliance result (for example, a likelihood score or an "infringement" prediction). During an infringement check, the process works as follows:

- The record label's system takes one of its songs (or a unique fingerprint derived from it) as input and forward-propagates through the local cutlayer of the network. This yields an encoded representation of the song. Crucially, this encoding is abstract — it does not reveal the raw audio, but it captures patterns the later layers can analyze [23].
- The "smashed" features at the cut layer are securely transmitted to the AI company's server. The AI company then

forward-propagates that activation through the rest of the model (or a special compliance-check subnetwork) using its private model parameters. For example, the AI company's model (or detector head) might compute a similarity or likelihood that the input audio was part of its training data.

- The final output (which might be an encrypted or blinded result – see privacy measures below) is sent back to the record label. This output could be, for instance, a probability or an error metric indicating how closely the AI model's knowledge matches the input song. If the value is above a certain threshold, it suggests the model was likely trained on that song (possible copyright infringement), if not, it suggests no memorization of that content.

The split learning method enables every participant to access only the information they need. The record label reveals only derived features from its music files while the AI company keeps its model weights private and does not get the raw input. Split learning offers enhanced model privacy protection compared to horizontal FL because each party has access to only parts of the model which prevents full visibility of the entire model to any single entity [24]. It is ideal when parties hold different modalities of data (here, one had a model or model updates, while the other has data to test), effectively creating a form of federated inference on combined inputs. Through split federated learning record labels and AI companies engage in a compliance check workflow that mirrors model training/inference activities while preventing data pooling. Using a shared split model the organizations process data collaboratively to identify copyright overlaps while maintaining local storage of proprietary information.

3) Privacy-Preserving Techniques for Secure Compliance:

Ensuring privacy is paramount: The music owned by record labels and the parameters of the AI company's model must remain hidden throughout the federated compliance check. Our system employs multiple cryptographic and privacy methods to protect the federated process:

Secure Multi-Party Computation (SMPC): SMPC enables multiple participants to perform joint computations over their input data while keeping those inputs hidden from each other [25]. Our design uses SMPC protocols to enable collaborative analysis between the AI model and the record label's data. For example, the record label and AI company can employ an SMPC framework (like Facebook's CrypTen or Microsoft SEAL) to evaluate the AI's model on the label's song in a secret-shared manner. The model's computations (matrix multiplications, etc.) are performed on encrypted or secret-shared values, so the AI company never sees the actual audio features and the record label never sees the raw model outputs. This could be implemented with an additive secret sharing scheme: the record label secret-shares the intermediate features with the AI company (or a neutral server), and the AI company secret-shares its model parameters. They then perform the forward-pass computation by exchanging masked values. At the end of the computation, only the final result (e.g., a risk score) is revealed (and only to the party authorized to see it). SMPC essentially functions like a virtual trusted calculator:

for instance, it can let the AI company privately evaluate its model on the record label’s data [25], or allow multiple labels to perform a joint aggregate audit without sharing individual data. The process ensures that proprietary data and models stay secure and undisclosed throughout the compliance check.

Homomorphic Encryption (HE): Homomorphic encryption [26] enable users can perform calculations on encrypted data to receive encrypted results that decryption is possible only via owner of the secret key. HE can be used in selected areas of the compliance pipeline to improve data protection. For example, a record label can encrypt an audio feature vector with their public key and send it to the AI company. The AI company then runs its model on the encrypted data without ever decrypting it, using partially homomorphic operations (addition, multiplication on ciphertexts). The outcome is an encrypted infringement indicator, which only the record label can decrypt to see the result. Throughout this process, the AI company learns nothing about the input or the output, since all intermediate data remains encrypted. Fully Homomorphic Encryption (FHE) schemes (which allow arbitrary computations on ciphertexts) can be heavy for deep learning, hence we use optimizations like leveled HE or Partially Homomorphic Encryption (PHE) for specific operations to keep overhead reasonable [27]. For instance, computing a simple dot-product similarity or a reconstruction error between the model’s output and a target song can be done under HE if we linearize the operation. Non-linear operations (like activations) can be handled either by the split learning approach (so that they occur on the AI’s side in plaintext on already encrypted inputs) or by efficient garbled circuits if needed. By carefully choosing which parts of the computation to encrypt, we ensure a balance between privacy and performance. The cryptographic strength of HE means even if communications are intercepted, the content (songs or model responses) remains unintelligible without the decryption key.

Differential Privacy (DP): We incorporate differential privacy to protect against information leakage in any shared outputs or updates. DP works by adding carefully calibrated random noise to results so that the presence or absence of any single data record is indistinguishable. In our context, the AI music company could train its model with DP-SGD (differentially private stochastic gradient descent), which would ensure the model does not memorize specific training examples (like a particular song) too exactly. This pre-emptively protects against infringement, because a DP-trained model is unlikely to regurgitate any one song verbatim. Even during the federated compliance check, DP can be applied. For example, if multiple record labels participate in a joint audit, the aggregated compliance metrics can have noise added before being revealed, so that no single label learns specifics about another label’s queries. The record label’s query results themselves could be noised if we only need a yes/no answer with high confidence. Importantly, the noise levels are set such that they do not obscure true infringements but hide minute details of the model’s behavior. DP ensures that any one song (even if it were in training) has a limited influence on the output,

preventing the exposure of exact memorized content. This technique is computationally cheap (just noise addition) and scales well, complementing heavier cryptographic methods by reducing how much sensitive information even exists in the computed results.

Zero Knowledge Proofs (ZKP): Zero-knowledge proofs allow a party to prove a statement about data or computations without revealing the data itself [28]. We utilize ZKPs to make the compliance process verifiable and legally defensible. For instance, after training, the AI company can generate a zero-knowledge proof of training that attests “This model was trained only on licensed data and did not include Record Label X’s songs” without revealing anything about the training data or model parameters. Recent advances in ZKPs for ML (zkML) enable proving properties of models, such as training steps or dataset membership, in a computationally feasible way [29]. Concretely, the AI company could commit to the dataset it used (e.g., via a cryptographic hash or Merkle root of all training data) and then provide a ZK-SNARK proof that none of the record label’s song hashes are in the committed dataset. This is akin to a zero-knowledge set membership test – proving a set intersection is empty without revealing the actual sets. Alternatively, the AI company can prove that it followed a prescribed training procedure (for example, a training run with differential privacy enabled, or only using a specific approved dataset) [29]. The record label (or a regulator) can verify this proof and be mathematically assured of compliance. Another use of ZKP in our system is for the infringement test itself. If the record label gets a negative result (no infringement detected), the AI company could output a ZKP that the test was carried out honestly on the model in question (preventing an AI company from swapping in a different “clean” model just for the test). Although generating ZK proofs for deep learning computations can be resource-intensive, we confine their use to periodic checks or final audits to keep it tractable. The outcome is that compliance checks are trustless – the record label doesn’t have to trust the AI company’s word, they have cryptographic proof of either compliance or violation, which is crucial for legal defensibility. By combining these measures – SMPC/secure computation for processing data, HE for data encryption in transit and compute, DP for output privacy, and ZKP for process verification – we create a robust privacy-preserving compliance system. Each technique is chosen to minimize performance hits. For example, we use partial HE and secret-sharing (which are faster than full FHE), we add only small noise for DP, and we generate ZK proofs for high-level properties rather than every single operation. The overall design ensures that at no point is sensitive information exposed in plain form, yet all parties can collaboratively achieve the goal of detecting unlicensed training. When the model training is complete, the AI company can package a compliance report: this might include the ZK proofs of training, differential privacy parameters used, and summary of any internal compliance tests. The record labels, through the federated system, get the ability to verify this report and test the model themselves, yielding high assurance that if the model passes, it truly did

not use unlicensed music. To avoid high financial cost for any single entity, the federated compliance system can be managed by a neutral third-party service or consortium of stakeholders. This service can maintain the secure aggregation server and coordinate cryptographic key management. Using cloud computing with hardware acceleration (like FPGAs for HE, or SGX secure enclaves as a backup option) can speed up cryptographic operations, reducing runtime and therefore cost. Also, many cryptographic libraries are open-source and optimized, meaning the main expense is computing time. With model compression and batching, we ensure that even large models can be handled with commodity hardware given some time (hours, not days, for a thorough audit of a big model against millions of song samples, for instance). The design favors one-time heavy computations (like proof generation or full-catalog scan) only when absolutely necessary (e.g., a legal dispute), whereas routine compliance checks can be much lighter (sampling a subset of songs, using partial evaluations, etc., to get a quick assurance).

4) Workflow: **Step 0:** Initial Setup and Key Exchange – All participating entities (the AI company and one or more record labels, or an auditor) set up the cryptographic environment. This involves generating encryption keys (public/private key pairs for HE for each label, key shares for SMPC, etc.) and exchanging any public parameters. They also agree on the model split architecture and the protocol (which cut layer, what format features will be, what threshold constitutes a violation, etc.). For example, the AI company publishes the architecture of the compliance model or the fingerprinting method it will use. A central coordination server (could be run by a neutral party or consortium) may exist to facilitate scheduling and key management, but it will not see any raw data or models.

Step 1: Registration of Data Commitments – The AI company commits to its training dataset and model. It computes a commitment hash (or Merkle root) of all training data it used. This is submitted to a smart contract or to the record labels in a ledger so that it's fixed (the company can't later change it). Likewise, each record label prepares a fingerprint database of their copyrighted songs – e.g., a set of audio hashes or embeddings – and commits to those (so that they can't maliciously add more songs later just to trap the AI company). These commitments will be used in ZK proofs later. This stage ensures both sides “lock in” the items of interest without revealing them.

Step 2: Local Model Training (AI Company) – The AI company trains its AI music model on its own data (e.g., publicly licensed music, user-generated music, etc.). This is done using its standard pipeline, possibly with differential privacy and logging as described. No external interaction is needed during core training, so no overhead is incurred here aside from any self-chosen privacy technique. Once the model is trained (or at certain checkpoints), it is saved for audit. Let's assume the model is now ready to be checked for compliance.

Step 3: The coordinator server notifies the AI company and relevant record label(s) that a check will happen. They establish a secure session. The AI company provides the

server-side model for the split learning inference – typically, this means loading the second part of the model on a secure computation server. If using SMPC, the AI company secret-shares or encrypts its model weights with the computation service (or among multiple servers). If using a TEE (Trusted Execution Environment) as an aid, the model could be loaded into an enclave. In any case, the AI company does not give the model in plaintext to the label, it only makes it available in the secure protocol. The record label in turn prepares its input data for the check. For instance, it selects a batch of 100 songs (or segments) that it strongly cares about. The label either keeps these on its local machine (for split learning) or encrypts them with homomorphic encryption (if the model will process them directly in encrypted form). All parties confirm readiness.

Step 4: Federated Inference/Processing – The record label's client-side application now goes through the selected songs one by one (or in batches). For each song, it does the following:

- Compute the feature representation (e.g., passes it through the local cutlayer of the model or simply prepares the raw input if using HE directly).
- Send the intermediate activation to the AI company's model server over an encrypted channel (TLS + the values might already be secret shares or encrypted numbers). If using pure HE, send the encrypted audio/features to the server.
- The AI company's server (or the joint MPC nodes) then perform the forward pass on the encrypted/secret shared data through the remaining network layers. For example, it computes the output logits or reconstruction of the input. Since the model is large, this computation is optimized as discussed (maybe using GPU, etc.).
- The server returns the encrypted result of the inference back to the record label. This might be the log-likelihood of the sequence, a set of output audio tokens, or a high-level “yes/no” flag in secret-shared form.

Step 5: Compliance Metric Computation – The record label now decrypts or reconstructs the results from Step 4. If the result was an encrypted likelihood score, the label decrypts it with its HE secret key. If it was done via MPC, the label combines its share of the result with the shares from the server to obtain the final number. Now the label has, for each song tested, a metric indicating how strongly the model reacted. The label compares these metrics to the expected range for non-members. For instance, if a certain song has a model likelihood far above a threshold (meaning the model highly likely has seen it [30], the system flags this song as a potential infringement. In practice, the label might set a threshold based on a statistical confidence (e.g., “if probability that the item is in the data set is greater than 0.9, flag it”). They could also use an internal classifier on the outputs – for example, if the output was the model trying to continue the song, the label can measure similarity between the continuation and the original. Some systems might automate this: e.g., compute a cosine similarity between audio embeddings of the original and the generated continuation. High similarity would yield

a flag. These calculations are done on the label's side, so no privacy issue arises. The outcome of this step is a compliance report: perhaps a table of songs vs. scores, highlighting any that exceed the infringement threshold.

Step 6: Result Sharing and Proof Generation – Now the record label has preliminary results. If all songs are in the clear (no suspicious scores), the AI model likely did not use any of the label's data. The record label can then cryptographically sign an attestation that “We, Label X, have tested Model Y on [date] and found no evidence of training on Label X’s catalog.” This attestation can be shared with the AI company as part of a compliance certificate. On the other hand, if any song was flagged, the system can escalate. The record label can notify the AI company (most likely through the protocol, without revealing which song in plaintext, at least initially). They might say “Song ID #5 from our hashed list appears to have been used in training. We request an explanation or remediation.” At this stage, the AI company has the option to contest or accept. If contesting, this is where zero-knowledge proofs or additional verification come in. The AI company might invoke the previously computed commitment of its training set and perform a private set intersection (PSI) with the label’s song in question. PSI can definitively show if that song (or its fingerprint) was in the training set, without the AI co learning which song it is (if done properly). If PSI comes out positive, it’s proof of infringement. If PSI is negative but the model’s behavior was still highly suspect, it could indicate the model learned something very close to the song (e.g., an overfitted surrogate). In either case, the parties now have cryptographic evidence. Optionally, they can involve a neutral auditor who reviews the evidence (the auditor could be given access to the song under NDA and maybe run a targeted test themselves for confirmation).

- The AI company can produce a ZK-proof that the model tested was indeed the one corresponding to the committed training hash. This prevents a scenario where the company trained a second “decoy” model without the label’s songs just to pass the test. The proof would show that the weights of the deployed model are a result of training on the committed dataset (or at least that they match a certain hash that was committed). Such a proof might use zk-SNARKs as described in the proof-of-training concept [29].
- If the result is clean, the AI company might also produce a ZK-proof that none of the label’s songs (from a committed list) appear in its training set. This could use a zk-proof of set disjointness, which might be heavy, but perhaps they only do it for a small set of top songs.

Step 7: Compliance Outcome – After analysis, one of two outcomes occurs. No Infringement Detected: All checks pass. Or, one or more labels detected their content in the model. In this case, the system can automatically provide evidence to the AI company and a regulator.

Step 8: Ongoing Monitoring – The federated system remains available for future checks. Throughout this process, all actions (from key exchange to final verification) are designed to be auditable and repeatable. Each cryptographic message or

proof can be logged (in encrypted form) to provide a trace in case of disputes. The combination of federated learning structure and advanced privacy techniques ensures that compliance verification is done scientifically and rigorously, minimizing trust and subjectivity. The result is a feasible, efficient, and privacy-preserving federated system that upholds copyright law without stifling the development of AI models. By balancing the load between parties and using cutting-edge cryptography, the solution scales to real-world industry usage – enabling record labels to defend their intellectual property and AI companies to innovate with accountability.

B. Compensation Model

One of the most pressing issues in AI-generated music is how to fairly compensate artists and record labels for the use of their works in training models. As AI technology’s popularity skyrockets, debates over licensing and copyright have grown. This section explores the complexities of crafting effective compensation frameworks that balance innovation with fair treatment of creators. First, we introduce two methods we considered using as our compensation model. We then outline our full model, along with an example of our system using a mock dataset.

1) Royalty-Based System: The first method we considered was a royalty-based system. Music streaming services utilize royalty-based systems in their compensation methods. It works by giving artists money based off the number of streams or plays, usually a fraction of a cent per stream. So how can we implement this system? Well, similar to streaming services, when models are trained, they must “play” a song in order to for the model to learn the properties of the song and genre such as rhythms, melodies and chords. Models usually require multiple full passing of training data through the model (known as epochs). In an epoch, some data can be passed through more than others, known as oversampling and undersampling. For example, if you are training a musical AI to create songs from multiple genres but your dataset has much more rock music than it does classical, your epoch can reuse some classical songs and leave some rock songs out of the training to represent the two genres equally. The royalty system aims to compensate artists for each play that their song has in the training process, acting similarly to how streaming services such as Spotify work. The cons of implementing this method is the information needed to implement this solution. Being able to see the source code and determine exactly how a model was trained is the only way to implement a royalty-based system. Many AI companies would consider this to be proprietary knowledge, and as such would not be open to disclosing this information. The main pros of a royalty-based system are that artists are compensated by plays, which means that if a song was used more in the training of the system, they are entitled to a bigger cut. If a song is used more in training, then it is more likely for the output of the generative AI to share key properties with that song.

2) Levy-Based System: A levy is defined as “(of a government or organization) to demand an amount of money, such

as a tax, from a person or organization” [31]. In a levy-based system, AI companies would be demanded to give an amount of money, which would then be pooled out amongst the artists in the dataset. This requires knowing which artists are in the dataset and also knowing how many of each of their songs are represented. Once this is done, we can divide the number of songs each artist has in the dataset by the total amount in the fund to evenly distribute the amount that each artist gets from this levy fund. The cons of implementing this method are that the data is not as fairly represented, as we have discussed with the oversampling and undersampling case above, thus some artists may get more than they actually deserve. The pros of this method are that it is easier to retrieve the data such as the number of songs that each artist has in a data set over trying to determine the specific training process.

3) **Compensation Framework:** Our compensation framework first operates by receiving the data from our FL framework, telling us which songs are above the threshold of probability, and therefore, are very likely to be in the dataset. Using this data, we can then generate a list of how many songs each creator has in the dataset. Due to our FL framework, we have decided that a levy-based system better fits the compensation model, but we still need to decide how to generate our levy (i.e. how much money each company will be paying to copyright holders).

ProRata.AI, an artificial intelligence start-up, has established a revenue-sharing model designed to fairly compensate content creators for their work when used by AI systems. The company, unlike other traditional AI models which tend to scrape online content without compensation publishers, has claimed to share half the revenue from subscriptions to its platform with its licensing partners. This already includes Universal Music, Axel Springer, Financial Times, The Atlantic, and Fortune. The main goal of the company is to license the technology behind its search engine to other generative AI companies. If an AI company were to adapt this business model, all of their lawsuits would come to an end, states ProRata CEO Bill Gross; “If you adopt this business model, this will end your lawsuits, because now you’ll be sharing revenue properly” [32].

ProRata generates revenue mostly through an AI-powered ad platform that places relevant ads within AI search results and digital content. They also have revenue streams from their proprietary attribution technology which can be licensed to other AI companies as a service, monitoring services that track content usage by AIs, and potentially subscription options through their Gist.ai search engine that showcases their attribution tech, although this technology is still in its Beta-testing stage.

ProRata.AI is currently valued at over \$130 million, and off the backs of a successful and ethical business model, we plan to adapt this to music AI companies. Taking Suno as an example, their current business model works almost entirely off subscriptions. Suno has three different subscription tiers, each with different features and pricing.

- Free Plan: 50 credits/day (equivalent to 10 songs), Suno

retains copyright.

- Pro Plan: \$8/month, 2,500 credits/month (500 songs), users hold the right to their creations.
- Premier Plan: \$24/month, 10,000 credits/month (2,000 songs), users retain full rights [33].

Suno has not released their revenue model to the public. However, we know that in 2023, Suno partnered with Microsoft Copilot to introduce AI music generation in the Copilot software. This partnership does not include any financial agreement, rather a win-win situation that brings more users to Suno, while adding additional features to Microsoft Copilots AI software [33]. As such, we currently estimate that Suno’s revenue come 100% from subscription fees. As such, half of all Suno’s revenue would be sent to the levy to then be distributed to artists. To show our compensation framework in action, we have generated a “mock company” to demonstrate. By taking some of the top musical generative AI companies and modelling them based on their revenue, we can show what a major AI company would have to pay in our framework. Suno has not released their revenue model to the public, but we know that in 2023, Suno partnered with Microsoft Copilot to introduce AI music generation in the Copilot software, however this partnership does not include any financial agreement, rather a win-win situation that brings more users to Suno, while adding additional features to Microsoft Copilots AI software [33]. As such, we currently estimate that Suno’s revenue come 100% from subscription fees. As such, half of all Suno’s revenue would be sent to the levy to then be distributed to artists. To show our compensation framework in action, we have generated a “mock company” to demonstrate. By taking some of the top musical generative AI companies and modelling them based on their revenue, we can show what a major AI company would have to pay in our framework.

Company	Revenue (\$USD)
Aiva Technologies	\$1.5M
Beatoven.ai	\$37.8k
Amper Music	\$5.1M
Boomy	\$5.8M
Suno	\$8M
Music.AI	\$22.1M

TABLE I
COMPANY REVENUES [34], [35], [36], [37], [38], [39]

The average revenue of \$13.26 million USD across major AI music companies (Aiva Technologies, Beatoven.ai, Amper Music, Boomy, Suno, and Music.AI) provides a useful baseline for projecting potential artist compensation frameworks. If these companies were to adopt a ProRata.ai-style revenue-sharing model, allocating 50% of subscription revenue to rights holders, and assuming an 100% subscription-based revenue structure approximately \$6.63 million would be distributed to artists and copyright holders annually per company. We have generated a mock dataset, comprised of just under 5,000 songs that a generative music AI company could use to train their system [40]. This dataset represents a multitude of genres, artists, and time periods so a user could realistically generate a song from almost every genre and style. Our goal

in this model was to showcase how a levy-based system could work based upon knowing the number of songs that each artist has in a given training set. Based upon this model, we will show you some of the top performers from this dataset, and how much they are owed based upon our compensation formula.

Track Artist	Unique Count	Compensation
Bad Bunny	30	\$8,816.10
Ren Avel	26	\$7,640.62
Asake	21	\$6,171.27
Bnxn	19	\$5,583.53
Seyi Vibez	18	\$5,289.66
LoFi Waiter	18	\$5,289.66
Wizkid	16	\$4,701.92
Linkin Park	14	\$4,114.18
Hozier	13	\$3,820.31
Sabrina Carpenter	11	\$3,232.57
Burna Boy	11	\$3,232.57
Zinoleesky	11	\$3,232.57
Billie Eilish	10	\$2,938.70
Red Hot Chili Peppers	10	\$2,938.70
Central Cee	10	\$2,938.70
Yume.Play	10	\$2,938.70
Green Day	9	\$2,644.83
Celine Dion	9	\$2,644.83
Metallica	8	\$2,350.96
Gunna	8	\$2,350.96
Lil Baby	8	\$2,350.96
Brent Fiyaz	8	\$2,350.96
Bruno Mars	8	\$2,350.96
J Balvin	8	\$2,350.96
Paramore	8	\$2,350.96
My Chemical Romance	8	\$2,350.96
Zhao Ying	8	\$2,350.96
Hao Yu	8	\$2,350.96

TABLE II
TOP ARTISTS COMPENSATION

Based on our calculations, each song in the dataset is entitled to \$1,372.10. Pictured above, we can see artists of many genres (latin, rap, pop, metal, rock, etc.) that have all made a significant impact on the dataset. Once these values are generated, AI companies are made aware of these values and will distribute the money to the necessary labels and copyright holders. We believe this method to be best implemented as a third party software to best ensure the protection of both parties' data. Record labels would enlist the help of the third party, who would then be responsible for compiling all required data, and only sending vital information when needed. The information received by the both companies (record label and AI company) during this process would be the number of songs by each artist that (our system believes) appears in the dataset, along with the breakdown of how much each artist receives, in a format similar to Table II. We designed our compensation framework to create the best possible outcome, a win-win scenario for both companies that avoids legal fees and time spent dealing with legal issues. Currently, the case against Suno is worth \$150,000 per song, and totals to over \$350 million. If suno were to implement our framework, this number goes down to \$22.5 million and saves valuable company time and resources by not having to fight a litigious lawsuit. At the same time, record labels are also avoiding legal fees and

court time, while receiving a steady, annual payout from AI companies. This means that record labels are actually given incentive to help AI companies grow, and thus could lead to a less competitive music culture.

VI. THE FUTURE OF MUSIC & AI

As the future of copyright law and music AI is still unsure, we can discuss two possible futures. One in which all materials are indeed fair use, and another does not believe these materials are fair use. If precedent is set in the copyright AI world that all copyrighted materials are indeed fair use, then this solution will no longer be viable to implement, as AI companies will not have to compensate copyright holders. Since the future of copyright law and AI remains unsure in the public eye, we set out to address all possible outcomes of these important cases. Our solution introduces ethical business practices by integrating advanced watermarking and fingerprinting techniques into AI training processes, ensuring that artists receive fair compensation while safeguarding intellectual property rights. This approach not only fosters transparency in data usage but also creates a win-win scenario for both AI developers and creators. Experts predict that future copyright law will evolve to address the unique challenges posed by AI, potentially leading to new legal precedents that better recognize the transformative nature of AI-generated music [41]. As ethical practices gain traction, we anticipate a shift toward revenue-sharing models that empower artists and encourage responsible innovation. Ultimately, this balanced framework sets the stage for a sustainable music ecosystem where technology and creativity coexist harmoniously.

REFERENCES

- [1] Soundraw, "Top 8 ai-generated songs you need to hear in 2025," *Soundraw Blog*, 2025, [Online]. [Online]. Available: <https://blog.soundraw.io/post/ai-generated-songs-you-need-to-hear>
- [2] V. Yurkevich, "Universal music group calls ai music a 'fraud,' wants it banned from streaming platforms. experts say it's not that easy," *CNN*, 2023, [Online]. [Online]. Available: <https://www.cnn.com/2023/04/18/tech/universal-music-group-artificial-intelligence/index.html>
- [3] D. J. Cloherty, I. UMG RECORDINGS, L. CAPITOL RECORDS, S. M. ENTERTAINMENT, A. R. CORPORATION, A. R. G. LLC, R. E. LLC, I. THE ALL BLACKS U.S.A., W. M. I. S. LIMITED, and W. R. INC., "Umg recordings, inc., et al. v. suno, inc." 2024, [Online]. [Online]. Available: <https://regmedia.co.uk/2024/06/24/suno-complaint.pdf>
- [4] "How does ai music work? from machine learning to viral hits," 2025, [Online]. Accessed: Jan. 27, 2025. [Online]. Available: <https://rareconnections.io/how-does-ai-music-work/>
- [5] ar5iv, "Melnet: A generative model for audio in the frequency domain," 2025, [Online]. Accessed: Mar. 01, 2025. [Online]. Available: <https://ar5iv.labs.arxiv.org/html/1906.01083>
- [6] Copyright.gov, "What is copyright?" accessed: Nov. 06, 2024. [Online]. Available: <https://www.copyright.gov/what-is-copyright/#:~:text=U.S.%20copyright%20law%20provides%20copyright,rental%2C%20lease%2C%20or%20lending>
- [7] U.S. Copyright Office, "U.s. copyright office fair use index," accessed: Nov. 06, 2024. [Online]. Available: <https://www.copyright.gov/fair-use/index.html>
- [8] A. Academy, "The rise of ai in audio engineering: How machine learning is revolutionizing music production," 2025, [Online]. Accessed: Jan. 15, 2025. [Online]. Available: <https://audioacademy.in/the-rise-of-ai-in-audio-engineering/#:~:text=For%20example%2C%20companies%20like%20iZotope,extensive%20technical%20knowledge%20or%20experience>

- [9] B. Brittain and B. Brittain, "Music labels sue ai companies suno, audio for us copyright infringement," <https://www.reuters.com/technology/artificial-intelligence/music-labels-sue-ai-companies-suno-audio-us-copyright-infringement-2024-06-06/>, Jun. 2024, accessed: Nov. 06, 2024.
- [10] B. Brittain, "Getty images lawsuit says stability ai misused photos to train ai," <https://www.reuters.com/legal/getty-images-lawsuit-says-stability-ai-misused-photos-train-ai-2023-02-06/>, Feb. 2023, accessed: Nov. 06, 2024.
- [11] S. Dhameliya, "Record labels sue ai platforms making music: Universal, sony, warner v/s suno, audio," 2025, accessed: Jan. 29, 2025. [Online]. Available: <https://iprmentlaw.com/2024/07/13/record-labels-sue-ai-platforms-making-music-audio-v-universal-sony-warner/>
- [12] J. Koehler, "Ai music generator suno admits it was trained on 'essentially all music files on the internet,'" <https://www.404media.co/ai-music-generator-suno-admits-it-was-trained-on-essentially-all-music-files#:~:text=The%20AI%20music%20generator%20company,tens%20of%20millions%20of%20recordings.%E2%80%9D>
- [13] E. Balla, "Here's how much data gets used by generative ai tools for each request," *Data Science Central*, 2025, accessed: Feb. 17, 2025. [Online]. Available: <https://www.datasciencecentral.com/heres-how-much-data-gets-used-by-generative-ai-tools-for-each-request>
- [14] Musette, "Why artificial intelligence will never replace musicians," *Medium*, 2025, [Online]. Available: <https://musettedc.medium.com/why-artificial-intelligence-will-never-replace-musicians-a03600b0310f#:~:text=If%20there%20has%20been%20a,may%20constitute%20a%20trade%20secret>
- [15] "Ai in music copyright detection — restackio," 2025, accessed: Feb. 01, 2025. [Online]. Available: <https://www.restack.io/p/ai-in-music-answer-copyright-detection-cat-ai>
- [16] "Umg recordings, inc. et al v. suno, inc. et al," [Online] Available: [Online]. Available: <https://dockets.justia.com/docket/massachusetts/madce/1:2024cv11611-272063>
- [17] B. Chapman, C. Taylor, and B. Husband, "Protecting training data for ai innovations in the medtech space," *Carpmaels & Ransford*, Apr. 2024, [Online] Available: [Online]. Available: <https://www.carpmaels.com/protecting-training-data-for-ai-innovations-in-the-medtech-space-part-1#:~:text=If%20there%20has%20been%20a,may%20constitute%20a%20trade%20secret>
- [18] B. Hiatt, "A chatgpt for music is here. inside suno, the startup changing everything," *Rolling Stone*, 2024, [Online]. [Online]. Available: <https://www.rollingstone.com/music/music-features/suno-ai-chatgpt-for-music-1234982307/>
- [19] A. Jawed, "What is a producer tag: All you want to know," <https://www.hollywood.com/blog/tips/what-is-a-producer-tag>, accessed: 16 March 2025.
- [20] S. Fellig, A. M. Gass, B. N. Lovejoy, S. N. Feldman, N. Taylor, and S. V. Damle, "Answer of defendant suno, inc. to complaint," 2024, [Online]. [Online]. Available: <https://www.musicbusinessworldwide.com/files/2024/08/SUNO-response-to-copyright-suit.pdf>
- [21] G. of Canada, "Copyright act (r.s.c., 1985, c. c-42)," November 2024, [Online; accessed 2025-01-15]. [Online]. Available: <https://laws-lois.justice.gc.ca/eng/acts/C-42/Index.html>
- [22] S. Vondran, "Proving "substantial similarity" in copyright infringement actions," September 2024, [Online; accessed 2025-02-15]. [Online]. Available: <https://www.vondranlegal.com/proving-substantial-similarity-in-copyright-infringement-actions>
- [23] MIT Media Lab, "Split learning: Distributed and collaborative learning," <https://www.media.mit.edu/projects/distributed-learning-and-collaborative-learning-1/overview/>, online; accessed March 2025.
- [24] S. C. C. Thapa, M. A. P. Chamikara and L. Sun, "Splitfed: When federated learning meets split learning," *arXiv preprint*, vol. arXiv:2004.12088, Apr. 2020.
- [25] A. H. S. S. M. I. B. Knott, S. Venkataraman and van, "Crypten: Secure multi-party computation meets machine learning," *arXiv preprint*, vol. arXiv:2109.00984, Sep. 2021.
- [26] S. S. J. Ma, S. Naas and X. Lyu, "Privacy-preserving federated learning based on multi-key homomorphic encryption," *International Journal of Intelligent Systems*, vol. 37, no. 9, pp. 5880–5901, Apr. 2021.
- [27] S. A. R. et al., "An advanced data fabric architecture leveraging homomorphic encryption and federated learning," *Information Fusion*, vol. 102, Feb. 2024.
- [28] R. Nguyen, "Zero knowledge proofs in machine learning: A comprehensive guide," *SotaZK*, Oct. 2024, <https://sotazk.org/insights/zero-knowledge-proofs-in-machine-learning-a-comprehensive-guide/>.
- [29] G. et al., "Experimenting with zero-knowledge proofs of training," in *CCS: Computer and Communications Security*, Nov. 2023, pp. 1880–1894.
- [30] A. S. S. Antebi, E. Habler and Y. Elovici, "Tag&tab: Pretraining data detection in large language models using keyword-based membership inference attack," *arXiv*, vol. 2501.08454, Jan. 2025.
- [31] "Levy," [Online]. [Online]. Available: <https://dictionary.cambridge.org/us/dictionary/english/levy>
- [32] D. Thomas, "Start-up prorata.ai valued at \$30mn after signing up uk publishers," *Financial Times*, 2025, [Online]. [Online]. Available: <https://www.ft.com/content/c917a1e1-60a5-42c5-9158-6199f8a1f9ab>
- [33] J. Handy, "Looking closer: Suno ai," *Handy AI*, 2025, [Online]. [Online]. Available: <https://handyai.substack.com/p/looking-closer-suno-ai>
- [34] "Suno revenue, growth, and valuation," *Sacra*, 2025, <https://sacra.com/c/suno/>
- [35] "Aiva technologies: Revenue, competitors, alternatives," 2025, [Online]. Accessed: Mar. 06, 2025. [Online]. Available: https://growjo.com/company/Aiva_Technologies
- [36] "Amper music - overview, news similar companies — zoominfo.com," 2025, [Online]. Accessed: Mar. 06, 2025. [Online]. Available: <https://www.zoominfo.com/c/amper-music-inc/414574746>
- [37] "Beethoven.ai - company profile - tracxn," 2025, [Online]. Accessed: Mar. 06, 2025. [Online]. Available: https://tracxn.com/d/companies/beethoven.ai/_GqrPyXIL1JWKE1nyCfOnii9m9JI0ifmlTTOCIewlI0
- [38] ZoomInfo, "Boomy - overview, news & similar companies — zoominfo.com," <https://www.zoominfo.com/c/boomy-corp/470700868>, 2025, accessed: Mar. 06, 2025.
- [39] "Music.ai: Revenue, competitors, alternatives," <https://growjo.com/company/Music.AI>, 2025, accessed: Mar. 06, 2025.
- [40] "Spotify music dataset," <https://www.kaggle.com/datasets/solomonameh/spotify-music-dataset?resource=download>.
- [41] J. Hutson, "The evolving role of copyright law in the age of ai-generated works," *Journal of Digital Technologies and Law*, 2024. [Online]. Available: <https://www.lawjournal.digital/jour/article/view/486#:~:text=Therefore%2C%20a%20revised%20copyright%20framework,as%20a%20replacement%20for%20it>.

Graph-Informed Transformers for Neural Network Inference Latency Prediction

Asad Khan

University of Toronto

asadk.khan@mail.utoronto.ca

Abstract—Deep learning applications such as real-time object detection in autonomous vehicles, interactive voice assistants, and high-frequency trading systems often require strict adherence to inference latency constraints defined by service-level objectives. Ensuring that neural network inference times meet these constraints before deployment presents a significant challenge to developers. In this paper, we introduce a transformer-based approach to predict neural network inference latency in pre-deployment stages. Our method utilizes a diverse synthetic dataset of feedforward neural networks, characterized at the operation level. These networks are represented as graphs, where the node attributes encode the type of operation and weight count, and the edges define the topology of the network. By treating each node as a token, the transformer leverages multi-head attention to capture structural and attribute relationships that strongly correlate with inference latency. Experimental results demonstrate that the proposed transformer model achieves much better performance when compared to baseline linear regression in predicting standard neural network inference latency across a wide variety of architectures and configurations. Ultimately, our transformer-based solution facilitates the development of latency-sensitive deep learning systems by enabling more reliable and efficient architectural optimization prior to deployment.

I. INTRODUCTION

Inference latency refers to the time delay between providing an input to a model and receiving the corresponding output. In deep learning, this metric is crucial for applications that require real-time or near-real-time responses [1]. Thus, deploying deep learning models in production environments requires careful consideration of inference latency constraints. Modern deep learning models often comprise millions or even billions of parameters [2], leading to challenges in inference latency and resource utilization. The complexity of these models poses significant interpretive challenges and can impact the efficiency of deployment [3].

A. Motivation

Previous work by Mendoza & Wang demonstrated the effectiveness of using graph embeddings and deep neural networks for latency prediction [4]. However, there remains room for improvement, particularly with the application of Transformer models.

The hierarchical and sequential nature of neural network architectures means that data is processed through multiple layers, each extracting increasingly abstract features from the input. This layered structure allows neural networks to learn complex representations, with each layer building upon

the outputs of the preceding ones [5]. Given this structure, Transformer models, known for their attention mechanisms, are particularly well suited for tasks involving hierarchical and sequential data [6]. Transformers can capture long-range dependencies and intricate relationships within data sequences [6], making them effective in modeling complex interactions between different layers and operations in neural networks. This capability suggests that Transformers could be advantageous in predicting neural network inference latency by effectively understanding and representing the hierarchical architecture of neural networks.

B. Related Works

1) *Inference Latency Prediction*: The prediction of neural network inference latency is an important area of exploration as deep learning models become increasingly complex, with various methodologies proposed to enhance accuracy and efficiency. Mendoza and Wang's work introduced the use of graph representations, utilizing attribute node lists and adjacency matrices at the operator level to predict individual inference latency [4].

Building upon this foundation, Liu et al. developed NNLQP, a multi-platform neural network latency query and prediction system [7]. NNLQP integrates an evolving database to store latency data across diverse hardware platforms, facilitating efficient retrieval and prediction. By leveraging a unified graph embedding, NNLQP addresses challenges associated with hardware graph fusion and kernel launch costs, thereby improving prediction accuracy.

In the realm of hardware-aware neural architecture search, Beglaryan and Ringhofer proposed a deep learning estimator model designed to predict the inference latency of fully connected deep neural networks on laptop CPUs [1]. Their work underscores the importance of tailored latency prediction models for specific hardware configurations.

Collectively, these studies underscore the importance of accurate latency prediction in optimizing neural network performance across various deployment environments. However, there remains an opportunity for further work by incorporating advanced architectures, such as transformers, to capture the intricate relationships within neural network structures more effectively.

2) *Transformers for Performance Prediction*: Transformer models have been used for structured prediction tasks across

various domains [8], largely due to their ability to model complex dependencies within sequences via self-attention mechanisms [6]. These mechanisms allow Transformers to capture intricate, multi-layered relationships in data, making them suitable for modeling latency in neural network architectures.

Recent work by Käppel et al. highlights the interpretability of attention scores, which shed light on how elements within a sequence relate to one another through self-attention [9]. This interpretability is particularly valuable when applied to latency prediction, as it provides insights into dependencies within neural network layers that traditional deep learning models might miss.

C. Problem Definition

Our work implements a novel Transformer-based architecture for predicting neural network inference latency that takes graph representations of a neural network as input and outputs a continuous value representing the predicted inference latency. By treating each layer or operation as a token, our model leverages multi-head attention to capture the structural and attribute relationships that correlate with inference latency. This approach provides a practical tool for developers to optimize neural network deployment without extensive testing, facilitating the development of latency-sensitive deep learning systems.

II. METHODOLOGY

In this section, we layout the data and model architecture used for training the transformer model.

A. Data

The dataset used to train the transformer model consists of 270,000 synthetically generated feedforward neural networks (FNNs), represented as graphs. In this representation, nodes correspond to operations within the network, such as input layers, hidden layers, and output layers, while edges signify the connections between these operations. Each node is annotated with attributes that detail the type of operation and the number of associated weights, encapsulating both the structural and computational characteristics of the network. This graph-based representation captures the intricacies of the architecture, providing a rich foundation for predictive modeling.

The dataset is generated through a systematic process to ensure diversity and complexity. Neural network architectures are sampled with depths ranging from three to ten layers. The number of units in each hidden layer is randomly assigned from a range of powers of two, spanning from 1 to 16,384 units. This approach allows for the creation of networks with varying computational demands, capturing both shallow and deep architectures. Once the architecture is defined, it is represented as a graph with an adjacency matrix to map the topological structure and a node attribute matrix to encode the operation type and weight count for each layer.

To capture the computational characteristics of the networks, inference latency is measured for each architecture.

Each network is instantiated and profiled on a standardized hardware platform (M2 Macbook Pro w/ 16GB RAM), where its latency is determined through a series of forward passes with randomly generated input data. The median latency across 5 runs is recorded to ensure consistency and robustness against measurement variability. This process results in a dataset containing graph representations of architectures paired with their corresponding latency values.

B. Model Architecture

The transformer model for inference latency prediction processes neural network architectures represented as graphs. In these graphs, nodes correspond to layers in the network, annotated with attributes that include the operation type and the number of weights. Edges capture the connections between these layers. The model predicts a single scalar value representing the network's inference latency based on this graph representation.

The input to the model consists of two tensors. The first tensor encodes the node attributes and has a shape of (batch size, maximum depth of the network, attribute dimension). The maximum depth of the network is set to 10, representing the largest number of layers expected in the dataset. This value ensures the model can handle most architectures without exceeding memory constraints. The attribute dimension is set to 3, which corresponds to the three features encoded for each node: the type of operation (e.g., input, dense), the number of weights, and any additional characteristics relevant to inference. The second tensor is a binary mask with a shape of (batch size, maximum depth of the network), which indicates valid nodes with 'True' and padded nodes with 'False'. The mask ensures that padded nodes do not contribute to the predictions.

The model begins by projecting the node attributes into a higher-dimensional embedding space using a fully connected dense layer. This transforms the input tensor to a shape of (batch size, maximum depth of the network, 64), where 64 is the embedding dimension. This value balances the need for rich feature representation and computational efficiency. Learnable positional embeddings are then added to the node embeddings to encode the sequential nature of the layers and distinguish nodes based on their positions within the network.

The core of the model is composed of two stacked Transformer encoder blocks. Each block uses a multi-head attention mechanism with four attention heads, which capture intricate dependencies between layers. The feedforward network within each block has a hidden size of 128, chosen to provide sufficient capacity for processing complex relationships while avoiding overfitting. Dropout is applied with a rate of 0.1 after the attention and feedforward layers to prevent overfitting, and layer normalization is included before and after each component to stabilize training.

Once the node embeddings have been processed by the Transformer blocks, the mask is applied to zero out contributions from padded nodes. The embeddings of valid nodes are pooled using a weighted mean, where the embeddings are first

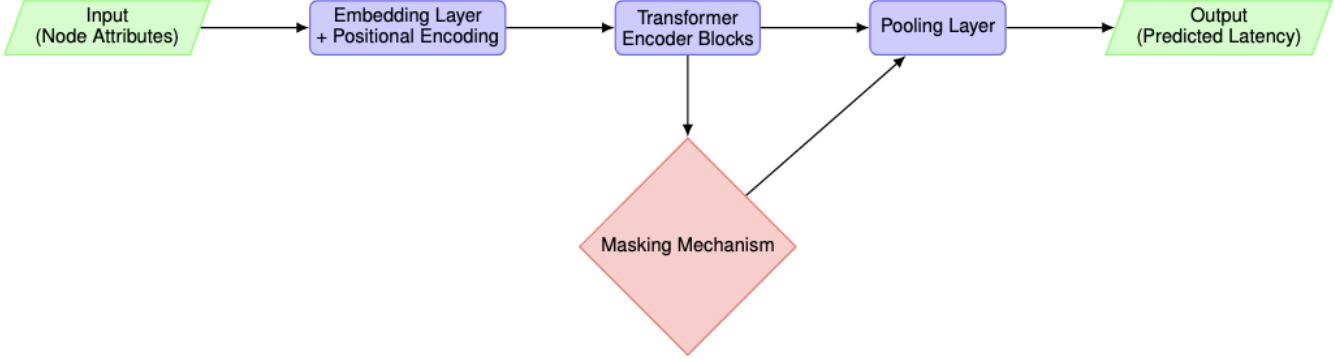


Fig. 1. Transformer Model Architecture for Inference Latency Prediction

multiplied by the binary mask, summed across the sequence dimension, and then divided by the number of valid nodes. This pooling mechanism aggregates information from all valid nodes while ensuring that padded regions do not influence the output.

The final dense layer produces a single scalar value for each graph, representing the predicted latency. The model is trained using mean squared error (MSE) as the loss function, which is suitable for regression tasks, and mean absolute error (MAE) is used as an additional evaluation metric to provide an intuitive measure of prediction accuracy.

The training procedure uses a batch size of 64 to strike a balance between computational efficiency and gradient stability. Training is conducted over 100 epochs, allowing the model sufficient time to converge without overfitting. The dataset is split into 80% for training, 10% for validation, and 10% for testing. The Adam optimizer is employed with a learning rate of 0.001 to ensure efficient convergence.

The architecture of the transformer model is well-aligned with the task of latency prediction. The use of multi-head attention allows the model to capture both local and global dependencies within the network architecture. Positional embeddings enable the model to encode the sequential nature of the layers, while the pooling mechanism effectively handles variable-length graphs by focusing on valid nodes. These design choices, combined with the chosen hyperparameters, make the model highly effective for predicting inference latency across a diverse range of neural network architectures. The values for the hyperparameters were chosen to balance representational capacity, computational efficiency, and the scale of the dataset. This alignment ensures the model is not only effective but also practical for real-world use cases.

III. RESULTS

The transformer model achieved a **test mean absolute error (MAE)** of **0.0038 seconds**. This represents a substantial improvement over the baseline linear regression model introduced in the work of Mendoza and Wang, which achieved a

test set MAE of **0.01619 seconds** [4]. By comparison, the transformer model reduces the average prediction error by nearly **76.5%**.

The mean absolute error (MAE) was selected as the primary evaluation metric for this task due to its interpretability and direct relevance to the problem of latency prediction. MAE measures the average absolute difference between predicted and actual latency values, providing an intuitive sense of prediction accuracy. The MAE of **0.0038** implies that, on average, the model's latency predictions deviate by just **3.8 milliseconds** from the true values. This precision is particularly significant given that the mean latency in the test set is **0.0400 seconds**, meaning the transformer model achieves an error of less than **10%** of the average latency.

The baseline linear regression model used in Mendoza and Wang's work is a strong baseline due to its simplicity, efficiency, and ability to capture basic patterns in the data. Its interpretability and low computational cost make it ideal for benchmarking more complex models. The transformer model's significant improvement over this baseline highlights its ability to capture the nuanced dependencies within neural network architectures, validating its advanced design.

TABLE I
LATENCY PREDICTION MEAN ABSOLUTE ERROR (MAE)

Model	MAE	Mean Latency (Test Set)
Transformer	0.0038	0.0400
Linear Regression	0.0162	0.0214

The transformer model's performance in predicting inference latency represents a significant improvement in neural network inference latency prediction. By achieving a test set mean absolute error (MAE) of **0.0038 seconds**, this improvement demonstrates the transformative potential of transformer-based architectures for capturing the intricate relationships inherent in neural network graphs.

The MAE of **0.0038** is particularly impressive when contextualized against the mean latency of **0.0400 seconds**. This

indicates that the transformer model produces predictions that are both accurate and actionable. For latency-critical applications, where small errors in prediction can have significant real-world consequences, such precision is invaluable.

The baseline linear regression model, while effective in capturing simple relationships between graph structure and latency, is inherently limited in its ability to model non-linear and hierarchical dependencies. This limitation is evident in its significantly higher MAE of **0.01619**, which corresponds to an average error of approximately **40% of the mean latency**. By contrast, the transformer model's ability to process graph-structured data with self-attention mechanisms allows it to uncover patterns and dependencies that the linear regression model cannot.

IV. CONCLUSION

In this paper, we presented a novel transformer-based approach for predicting neural network inference latency during the pre-deployment stages. Using a synthetic data set of graph-represented feedforward neural networks, our method captures both structural and computational characteristics of neural architectures. The use of multi-head sequential attention in the transformer model enables it to effectively model intricate relationships between layers, significantly improving latency prediction accuracy compared to baseline methods such as linear regression. This demonstrates the capability of our approach to generalize across diverse architectures and configurations.

While the transformer model demonstrates excellent performance in predicting inference latency, certain limitations highlight areas for further investigation and potential improvement.

The dataset used for training and evaluation consists of synthetically generated feedforward neural networks. While this approach ensures diversity in the architectures, it may not fully capture the complexities and variations present in real-world deployments. Factors such as hardware-specific optimizations, memory hierarchies, and non-ideal runtime behaviors are not reflected in the dataset. As a result, the model's performance on real-world neural networks deployed in different environments may differ from the results reported here.

To address this, there are avenues for improving the transformer model's capabilities. Incorporating more diverse datasets, including real-world neural network architectures and multi-platform latency measurements, would enhance the model's robustness. It would also be beneficial to integrate explicit hardware features into the input representation which could improve cross-platform performance.

Despite these limitations, the transformer model demonstrates significant promise for inference latency prediction. Addressing the outlined challenges and exploring areas for improvement would further enhance its utility and applicability in real-world scenarios. These efforts would make the model an even more effective tool for optimizing neural network deployment in latency-sensitive applications.

By facilitating more reliable and efficient optimization of neural network architectures prior to deployment, our approach empowers developers to design systems that adhere to strict inference latency constraints. This work contributes to advancing pre-deployment tools for deep learning systems, with potential extensions to incorporate hardware-specific adaptations or to explore hybrid models for further enhancing prediction accuracy.

REFERENCES

- [1] L. Beglaryan and C. Ringhofer, "Development and training of a deep learning approach to estimate latency of deep neural network inference," *American University of Armenia*, 2023.
- [2] S. Rajbhandari, J. Rasley, O. Ruwase, and Y. He, "Zero: Memory optimizations toward training trillion parameter models," *arXiv preprint arXiv:1910.02054*, 2020. [Online]. Available: <https://arxiv.org/abs/1910.02054>
- [3] S. Sinha and Y. M. Lee, "Challenges with developing and deploying ai models and applications in industrial systems," *Discover*, 2024.
- [4] D. M. Mendoza and S. Wang, "Predicting latency of neural network inference," Stanford University, Tech. Rep., 2020. [Online]. Available: http://cs230.stanford.edu/projects_fall_2020/reports/55793069.pdf
- [5] M. M. Taye, "Theoretical understanding of convolutional neural network: Concepts, architectures, applications, future directions," *Computation*, vol. 11, no. 3, p. 52, 2023.
- [6] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, vol. 30, 2017, pp. 5998–6008.
- [7] L. Liu, M. Shen, R. Gong, F. Yu, and H. Yang, "Nnlqp: A multi-platform neural network latency query and prediction system with an evolving database," in *Proceedings of the 51st International Conference on Parallel Processing*, ser. ICPP '22, 2022.
- [8] B. Wang, L. He, L. Song, R. Niu, and M. Cheng, "Attention-linear trajectory prediction," *Sensors*, vol. 24, no. 20, 2024. [Online]. Available: <https://www.mdpi.com/1424-8220/24/20/6636>
- [9] M. Käppel, L. Ackermann, S. Jablonski, and S. Härtl, "Attention please: What transformer models really learn for process prediction," in *Business Process Management - 22nd International Conference, BPM 2024, Krakow, Poland, September 11-15, 2024, Proceedings*, 2024, pp. 203–220. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-031-70396-6_12

Lovelytics: Multi-Agent Approach to LLM Task Automation for Business Users

Alvina Yang

University of Toronto

alvina.yang@mail.utoronto.ca

Stephanie Lu

University of Toronto

steph.lu@mail.utoronto.ca

Julien Liang

University of Waterloo

jh2liang@uwaterloo.ca

Mateo Arcos

University of Toronto

mateo.arcos@mail.utoronto.ca

Zachary Tang

University of Toronto

zach.tang@mail.utoronto.ca

Jeff Lu

University of Toronto

jeff.lu@mail.utoronto.ca

Hannah Ye

University of Toronto

hannahh.ye@mail.utoronto.ca

Benson Yan

University of Waterloo

b58yan@uwaterloo.ca

Sina Fallah Ardizi

New York University

sinafallah98@gmail.com

Amr Alomari

University of Toronto

amr.alomari@mail.utoronto.ca

Jeremy Qu

University of Toronto

jeremy.qu@mail.utoronto.ca

Abstract—This paper addresses the challenge of automating business tasks using Large Language Models (LLMs) by focusing on two key aspects: generating high-quality prompts from unclear user input and executing tasks in a modular and scalable way. The system proposed combines DSPy (Declarative Self-Improving Programs)-driven prompt generation, which refines prompts based on feedback and task context, with a multi-agent execution approach [1]. Unlike common industry practices, this system reduces manual effort by automating both prompt creation and task execution. The goal is to make AI-powered task automation accessible to non-technical users, allowing them to adopt LLMs into their daily workflow without the need for specialized knowledge. By democratizing task automation, the system opens up new possibilities for more efficient workflows across organizations.

I. INTRODUCTION

Rapid adoption of artificial intelligence in business operations has fueled the demand for automating complex workflows using LLMs. Organizations increasingly rely on LLMs for document processing, customer service, and data analysis, seeking improvements in efficiency and scalability. However, LLM effectiveness depends on prompt quality, and poorly structured prompts often lead to ambiguous, incomplete, or misaligned outputs. This presents a significant barrier to this form of automation, particularly for non-technical users unfamiliar with prompt engineering.

This paper introduces a DSPy-driven framework for structured prompt generation, enabling users to convert vague automation requests into well-formed, context-aware instructions that improve LLM performance. Additionally, we develop a multi-agent task execution system that breaks down workflows into modular, interdependent steps, improving reliability and adaptability. Hosted on Lovelytics’ Databricks Azure tenant, this system ensures secure, scalable automation with direct access to enterprise datasets. As a Databricks partner, Lovelytics enables seamless integration with enterprise workflows

by allowing secure data retrieval and processing directly from the Databricks File System (DBFS).

A. Motivation

Despite advancements in AI, prompt engineering remains a major challenge, especially for nontechnical users. Vague instructions yield unreliable responses, missing context reduces accuracy, and multistep tasks often result in logical inconsistencies. These issues prevent organizations from fully leveraging LLMs for automation. Existing solutions, such as manually crafting prompts and heuristics, offer partial improvements but struggle with generalization and structured execution.

B. Problem Definition

This paper addresses the dual challenge of:

- 1) Generating structured high-quality prompts from ambiguous user input
- 2) Executing complex business automation tasks in a modular and scalable manner

To solve these challenges, we introduce a system that combines:

- DSPy-driven prompt generation, which dynamically refines prompts based on iterative feedback and task constraints, resulting in more effective instructions for LLMs.
- Multi-agent task execution, where specialized agents manage different workflow stages in parallel.

Unlike traditional prompt engineering methods that rely on manual refinement, our approach automates prompt optimization while managing multi-agent execution, minimizing human intervention, and increasing task reliability.

C. Limits of LLM Automation

The Occupational Information Network (O*NET) is a comprehensive online database of U.S. occupation information, maintained by the Department of Labor.

O*NET assists in prompt optimization by providing:

- A detailed breakdown of jobs into specific tasks and subtasks.
- Industry-specific terminology, responsibilities, and skill requirements.
- Information on job-specific technologies, skills, and tools.

Users interact with the O*NET database by entering their job title, allowing us to retrieve and suggest job-related tasks for automation while also gathering job-specific context to better understand their role. The automatability of a task by an LLM depends on its structure, complexity, and input/output requirements. Highly structured, rule-based, repetitive tasks, as well as those with a definitive correct answer, are generally automatable. In contrast, tasks that require real-world interactions, deep reasoning, or creativity are less suitable for automation. Table I below highlights key features of automatable tasks.

TABLE I
FEATURES OF TASKS AUTOMATABLE BY LLMs

Automatable by an LLM	Not Automatable by an LLM
Text-based and well-defined input/output	Real-time decision-making
Pattern recognition, generalization, and context-based reasoning	Highly specialized tasks
Tolerance for imperfection	Multimodal reasoning with real-world interaction

II. METHODOLOGY

The DSPy-driven prompt engineering component focuses on refining vague user requests into structured ‘superprompts’ that provide the model with clear instructions, contextual information, and defined constraints. These superprompts improve task accuracy by integrating external knowledge and breaking down complex tasks into manageable components. The multi-agent execution system then takes these structured prompts and efficiently executes them by dividing tasks into subtasks and assigning them to specialized agents. Together, these components ensure a scalable, context-aware approach to automating business processes using LLMs.

A. DSPy-Driven Prompt Engineering

DSPy is a framework for optimizing prompts for LLMs [2]. It provides an abstraction layer which allows developers to define tasks declaratively while automatically fine-tuning the prompts and reasoning strategies using data-driven optimization.

Our approach uses DSPy to transform vague user requests into structured ‘superprompts’ that provide the LLM with clear instructions and relevant context. Initial user prompts undergo evaluation to identify their suitability for specific task types, such as writing or reviewing. Superprompts integrate external resources such as PDFs and domain-specific data to enhance accuracy. Utilizing chain-of-thought reasoning,

the DSPy implementation systematically leverages four key components:

- 1) Context from files, which tailors responses based on user-provided documents
- 2) Domain-specific context, aligning outputs with best practices
- 3) Task considerations, defining constraints and dependencies
- 4) Task subtasks, breaking down automation into structured steps

The refinement process begins by defining signatures, specifying task inputs and outputs. DSPy then generates candidate prompts and evaluates their effectiveness against predefined criteria. Feedback-based optimization ensures iterative improvements by incorporating high-quality prompt samples for training. The final optimized superprompts are processed by the multi-agent system, which involves feeding them into ChatGPT and evaluating performance.

B. Multi-Agent System for Task Execution

The multi-agent system (see Fig. 1) is responsible for executing tasks derived from DSPy-generated prompts. It does so by breaking workflows into structured subtasks handled by specialized agents. The system processes three key inputs: (1) the processed superprompt from DSPy, (2) instructional files, which contain instructions and important information, and (3) supplementary files, which provide useful contextual information via RAG-based retrieval.

The workflow initializes when the Planner Agent analyzes the ‘superprompt’ and creates the Task Plan, which divides the overall task into individual subtasks with execution steps, displaying the information on the frontend. The user can choose to modify the Task Plan by adding, removing, or adjusting subtasks in an interactive refinement loop before execution begins. Each subtask is assigned to an Executor Agent, which processes the instructional files and retrieves supplementary data to generate an output. These agents operate in parallel for efficiency. Once all subtasks are completed, the Merger Agent compiles them into a coherent final result. The Verification Agent then evaluates this output against predefined criteria. If it meets the required standards, it is finalized; otherwise, the process repeats up to a user-defined retry limit to refine the results.

In the second iteration of our design, we introduced an alternative approach: instead of iterative retries, the system runs five parallel executions of the full pipeline. A Selector Agent then evaluates and selects the best overall output, reducing computational overhead while improving reliability.

III. EVALUATION METHODS

We use two main approaches to evaluate the quality of our system: human evaluation and GPT-based benchmarking. Each method focuses on different aspects of the generated output to provide a well-rounded understanding of the system’s performance.

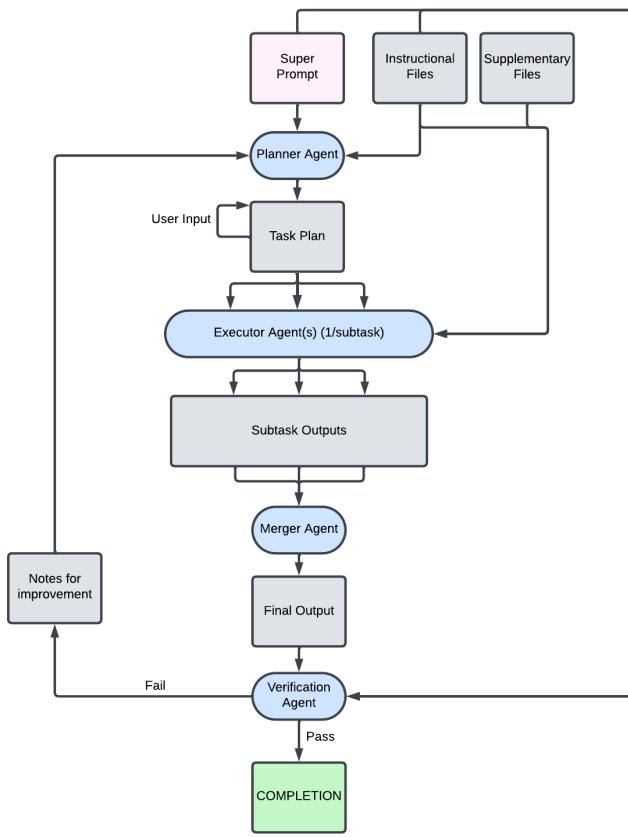


Fig. 1. Multi-agent system architecture

A. Human Evaluation

Human evaluation focuses on straightforward, objective criteria:

- 1) **Word count:** The total number of words in the output, averaged across multiple responses.
- 2) **Output format:** Checks whether the generated content matches the requested structure.
- 3) **Instruction adherence:** Evaluates whether the intermediate and final outputs follow the instructions provided in the task.

These criteria ensure that the system meets basic expectations, producing content in the correct form and following the given instructions. LangSmith was used to support human evaluation.

B. LLM-Based Benchmarking

LLM-based benchmarking focuses on a more detailed evaluation of content quality. In this method, advanced LLM agents, referred to as 'council members', are used to assess the generated outputs. These agents evaluate the responses based on an initial set of metrics. After reviewing the prompt and any provided instructional files, each 'council member' suggests 1-3 additional criteria that they believe are important for the evaluation. These new metrics are then added to the final list.

The final score for each output is calculated by averaging the scores from all 'council members'. This approach enables a more comprehensive evaluation of the content, considering aspects such as relevance, clarity, and depth.

C. Relevant Work

The approach described above draws inspiration from two primary sources. First, the concept of using multiple LLM agents as a 'council' is discussed in the paper Language Model Council: Democratically Benchmarking Foundation Models on Highly Subjective Tasks [3]. This paper demonstrates how collaborative LLM benchmarking can provide a more comprehensive evaluation for writing-based tasks. The idea of using multiple LLM agents to collaboratively generate custom evaluation criteria and assess outputs was particularly appealing as it minimizes bias and contributes to a more balanced evaluation.

Second, the methodology outlined in the GitHub repository [4], which provides guidelines for evaluating writing quality, also influenced the approach. This repository emphasizes the use of LLMs to define evaluation metrics for writing tasks, further guiding the development of the benchmarking process.

D. Tasks for Benchmarking

For the benchmarking, the task automation system was tested using three different tasks:

- **Environmental History of Computing Essay:** This task includes both an instructional file and a supplementary file, providing context for generating the essay.
- **Literary Research Writing:** This task includes an instructional file and several different books and papers on a literary work, with information from competing sources that needs to be referenced in conjunction to each other.
- **Fire Safety Protocols:** Unlike the other tasks, this one lacks both instructional and supplementary files, making it more open-ended and testing the system's ability to generate relevant content autonomously.

E. Experimental Setup

To evaluate the performance of different systems against the proposed multiagent system, benchmarking was conducted on the output from the following models:

- GPT-4o-mini with direct prompting
- GPT-4o with direct prompting
- GPT-o3-mini-high with direct prompting
- The proposed system with GPT-4o-mini in the backend

Each model was tested three times to ensure consistent and reliable results.

IV. RESULTS

The experimental results comparing the outputs of various models against our system are presented in the appendices. In general, our system demonstrates significantly superior performance compared to GPT-4o-mini and GPT-4o. It shows slight advantage over GPT-o3-mini-high, while running at a

fraction of the cost. We will focus our analysis on the first task: an essay on the Environmental History of Computing.

The execution of the Environmental History of Computing Essay task shows significant differences in model performance. The task involved writing a 2,000-word essay with instructional and contextual files for guidance.

The 4o-mini model generated only 420 words in bullet point format, failing to meet word count and structural requirements, with a benchmark score of 52.50. The 4o model performed better, generating 828 words in paragraph form, but still lacked the clear thesis-claim-evidence structure of an essay, earning a score of 78.33. The o3 model, a reasoning-based model, produced 1,704 words with paragraphs and met the structural requirements, earning a benchmark score of 82.78. However, while this model exhibits better performance, it is nearly seven times more expensive to run compared to our system.

The proposed system, which uses 4o-mini, produced 1,979 words in well-structured paragraphs and adhered to all instructions. With a benchmark score of 85.06, it outperformed all other models, showing that even smaller models can achieve high-quality outputs when optimized through task automation.

The difference in performance between directly prompting o3-mini-high versus the other models is primarily due to the reasoning strength of o3, which excels in content coherence. However, the model still struggles with instruction adherence. In contrast, our system leverages DSPy for prompt engineering and LangGraph for parallel execution and subtask decomposition, allowing 4o-mini to generate structured content with better instruction adherence.

A. Discussion

The primary goal of this work is to optimize the process of prompt creation and task automation. The DSPy-driven prompt engineering approach effectively converts vague task automation requests into structured superprompts, which in turn enhances the performance of LLMs.

Experimental results demonstrate that the proposed system outperforms baseline models across several evaluation metrics, including human evaluation and LLM-based benchmarking. By leveraging LangChain, the system also supports the parallel execution of subtasks, which significantly reduces latency, allowing for faster processing times and improving the overall efficiency of the task automation process.

V. CONCLUSION

This work presents a system that enhances task automation by utilizing a multiagent architecture in combination with DSPy to transform vague task automation requests into structured superprompts, optimizing task execution. Experimentation shows that the proposed system outperforms baseline models across multiple evaluation metrics, including human and LLM-based assessments. The system's modular design, supported by LangChain, enables parallel automation of subtasks, enhancing both efficiency and the overall user experience.

In practice, DSPy lowers the barrier for non-technical users to automate tasks, which promotes a broader adoption of AI-driven workflows. By leveraging Databricks and Azure, the proposed system also ensures secure data access and smooth integration into existing enterprise ecosystems.

A. Future Work

Future work will focus on further improving the capabilities of the system, refining existing workflows and user experience, and extending its functionality in a few key areas:

- 1) **Improvement Pipeline:** One potential direction is to develop a system that focuses on improving existing work rather than completing new tasks. This could involve creating a pipeline that takes a current workflow as input, uses agents to analyze and refine it, and then outputs an improved version. This approach would enable the system to iteratively improve tasks and optimize existing processes over time, making it more adaptable to changing requirements.
- 2) **Effective Referencing:** Another area for future development is enhancing the system's ability to reference examples for task output generation, including tone, structure, and content formatting. By comparing task outputs with high-quality examples, the system could improve its ability to generate outputs that align more closely with user expectations. This would involve developing more advanced algorithms for contextual understanding and comparison, allowing for more effective use of external references during task execution.

These areas of improvement will further increase the flexibility and performance of the system, ultimately enabling broader application in real-world business automation scenarios.

REFERENCES

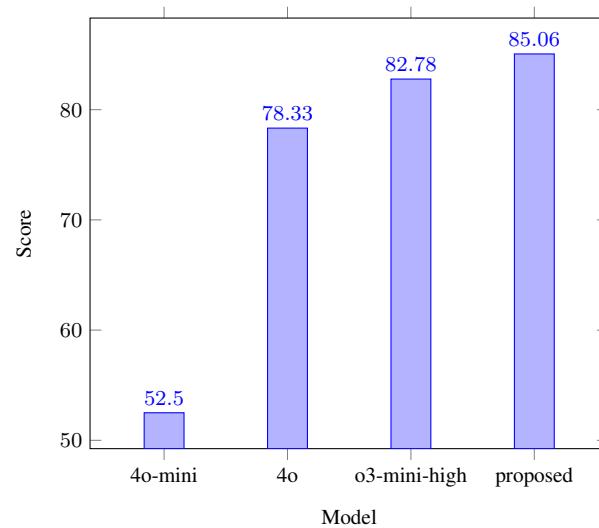
- [1] A. Yang, S. Lu, J. Liang, M. Arcos, Z. Tang, J. Lu, H. Ye, B. Yang, S. F. Ardizi, and A. Alomari, "Project Git Repository," 2024. [Online]. Available: <https://github.com/alvina-yang/Lovelytics>
- [2] O. Khattab, A. Singhvi, P. Maheshwari, Z. Zhang, K. Santhanam, S. Vardhamanan, S. Haq, A. Sharma, T. T. Joshi, H. Moazam, H. Miller, M. Zaharia, and C. Potts, "DSPy: Compiling Declarative Language Model Calls into Self-Improving Pipelines," 2023. [Online]. Available: <https://arxiv.org/abs/2310.03714>
- [3] J. Zhao, F. M. P. del Arco, B. Genchel, and A. C. Curry, "Language Model Council: Democratically benchmarking Foundation models on Highly Subjective Tasks," 2025. [Online]. Available: <https://arxiv.org/abs/2406.08598>
- [4] L. Mazur and C. Norton, "LLM Creative Story-Writing Benchmark," 2025. [Online]. Available: <https://github.com/lechmazur/writing>

APPENDIX A

TABLE A1
BENCHMARK TASK #1 - HUMAN EVALUATION

Model	Word Count	Output Format	Instruction Adherence
4o-mini	420	Bullet Points	No
4o	828	Paragraphs	Yes
o3-mini-high	1704	Paragraphs	Yes
proposed	1979	Paragraphs	Yes

TABLE A2
BENCHMARK TASK #1 - LLM-BASED BENCHMARKING SCORES

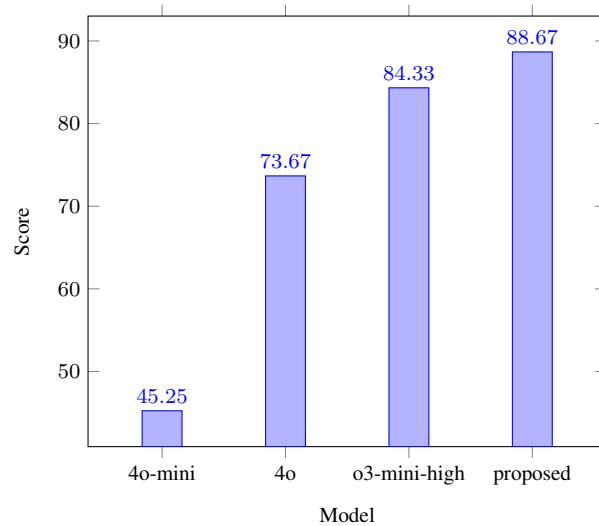


APPENDIX B

TABLE B1
BENCHMARK TASK #2 - HUMAN EVALUATION

Model	Word Count	Output Format	Instruction Adherence
4o-mini	533	Bullet Points	No
4o	917	Bullet Points	No
o3-mini-high	2154	Paragraphs	Yes
proposed	2261	Paragraphs	Yes

TABLE B2
BENCHMARK TASK #2 - LLM-BASED BENCHMARKING SCORES

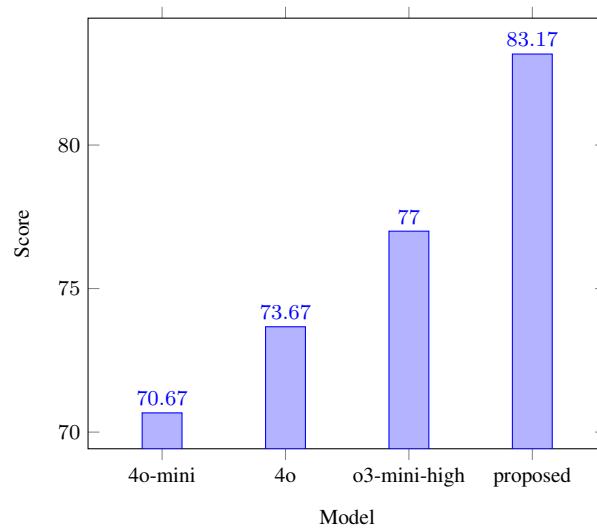


APPENDIX C

TABLE C1
BENCHMARK TASK #3 - HUMAN EVALUATION

Model	Word Count	Output Format	Instruction Adherence
4o-mini	651	Bullet Points	Yes
4o	354	Bullet Points	Yes
o3-mini-high	777	Bullet Points	Yes
proposed	2896	Paragraphs	Yes

TABLE C2
BENCHMARK TASK #3 - LLM-BASED BENCHMARKING SCORES



Mechanistic Interpretability Through Multi-Feature Steering of Neural Networks

David Courtis

Queen's University

courtis.david@queensu.ca

Jagrit Rai

Queen's University

jagrit.rai@queensu.ca

Brigitte Rauch

Queen's University

22krc1@queensu.ca

Dhruv Popli

Queen's University

21dp40@queensu.ca

David Krayacich

Queen's University

22jyy@queensu.ca

Rojella Santos

Queen's University

22by28@queensu.ca

Abstract—This paper introduces Sparse Autoencoder (SAE)-based Multi-Feature Steering for extracting and controlling latent representations in neural networks. We extend dictionary learning research by applying sparse autoencoders to the Gemma-2B language model to extract monosemantic features and enable simultaneous steering along multiple feature directions. Our approach facilitates direct manipulation of feature activations through an interactive interface, providing precise control over model behavior. Empirical evaluation comparing instruction-tuned and untuned model responses reveals that while SAEs enhance interpretability, challenges persist including feature entanglement, overfitting, and coherence degradation. Despite smaller models having limited capacity to encode high-level conceptual features, structured multi-feature interventions yield valuable insights into neural network activations. Our contrastive methods for feature extraction demonstrate superior precision compared to existing auto-interpretability techniques.

I. INTRODUCTION

Mechanistic interpretability seeks to understand neural networks at the circuit level by decomposing complex systems into simpler, comprehensible components. This approach examines individual neurons, attention heads, and channels to determine their specific functions and interactions. By achieving this granular understanding, we gain transparency into neural mechanisms, enabling more precise model refinement and providing deeper insights into emergent intelligence while helping prevent unexpected behaviors.

A. Background

Current interpretability methods include saliency maps (highlighting network focus in images), feature importance scores (similar to SHAP), and post-hoc explanations using local linear approximations. However, these approaches have significant limitations—they often provide only surface-level, correlation-based insights rather than revealing the underlying computational structures that drive neural network behavior. Moreover, these methods generally operate as black-box analyses, examining inputs and outputs without illuminating the intermediate processing that forms the foundation of neural computation.

B. Related Work

Our research builds upon three significant papers from Anthropic addressing neural network interpretability. The first, "Toy Models of Superposition" [1], investigates how neural networks can represent more features than their dimensional capacity through the phenomenon of superposition.

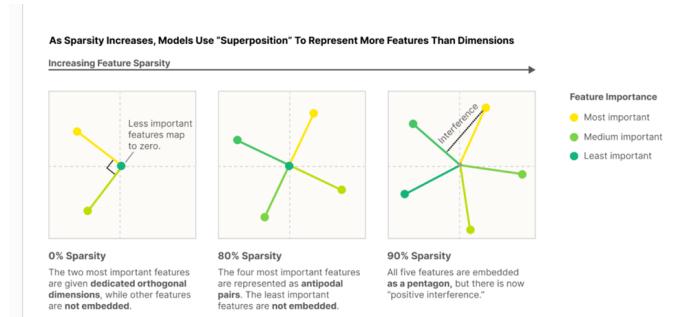


Fig. 1. Illustration of feature representation in neural networks as sparsity increases. At 0% sparsity, distinct neurons represent important features. At 80% sparsity, features pair as opposite directions within neurons. At 90% sparsity, features become densely packed around geometric shapes, increasing representation capacity but introducing interference.

This research demonstrated that superposition allows networks to compress information efficiently by tolerating controlled interference, enabling representation of more features than available dimensions. However, a key limitation emerges: individual neurons often become polysemantic, encoding multiple unrelated concepts, which significantly complicates understanding their specific functions. The second paper, "Monosemanticity: Decomposing Language Models with Dictionary Learning" [2], introduces sparse autoencoders to decompose one-layer transformer model activations into distinct features, substantially improving interpretability.

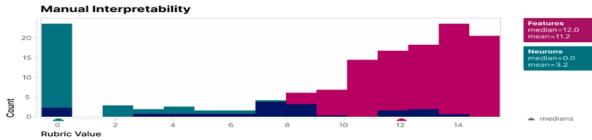


Fig. 2. Human evaluation scores comparing interpretability of features extracted via sparse autoencoders (pink) versus individual neurons (teal). Features consistently received higher interpretability scores with a median of 12, while neuron scores clustered near 0.

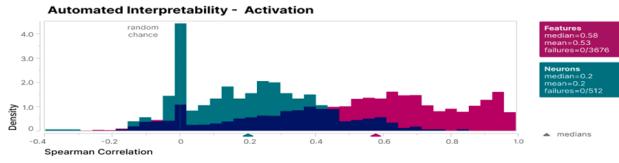


Fig. 3. Automated interpretability assessment using Spearman correlation between activations and specific behaviors. Extracted features (right) show significantly stronger correlations compared to neuron activations (left), which exhibit near-random patterns.

While these results confirmed that sparse autoencoders substantially enhance interpretability, several challenges remain:

1. feature splitting, where interpretable features fragment as extraction scale increases;
2. interference between features that limits perfect monosemanticity;
3. overfitting to training data patterns rather than capturing generalizable concepts; and
4. persisting ambiguity in some extracted features that lack clear correspondence to human-understandable concepts.

The third paper, "Scaling Monosemanticity: Extracting Interpretable Features from the Claude 3 Sonnet" [3], applied dictionary learning at scale to investigate monosemantic features in large language models. This research demonstrated that sparse autoencoders could successfully extract interpretable features from large models and that scaling improves autoencoder training efficiency. Nevertheless, many extracted features still exhibited superposition, highlighting the persistent complexity in large-scale networks.

C. Motivation and Contribution

Our research addresses a critical gap in existing interpretability solutions: their inadequacy in exposing internal computational structures and addressing the polysemantic challenge. We make several key contributions:

1. We develop a methodology for identifying and manipulating multiple meaningful features simultaneously using sparse autoencoders applied to pre-trained language models.
2. Rather than theoretical exploration, we implement practical mechanisms for users to directly manipulate specific feature activations, enabling fine-grained control over model behavior.

3. We introduce a contrastive approach for feature selection, comparing activations from positive and negative prompts to isolate and fine-tune specific semantic features.

4. We formalize a mathematical framework for multi-feature steering that enables predictable and interpretable model manipulation. The mathematical foundation for our multi-feature intervention is:

$$h' = h + \sum_{i=1}^n \alpha_i v_i \quad (1)$$

where h represents the original hidden state, v_i corresponds to a monosemantic feature direction, and α_i is a user-defined tuning factor that scales each feature's contribution. The original logits are computed as:

$$\ell = Wh \quad (2)$$

After applying the multi-feature intervention, the modified logits become:

$$\ell' = W \left(h + \sum_{i=1}^n \alpha_i v_i \right) \quad (3)$$

The combined logit shift is:

$$\Delta \ell = \sum_{i=1}^n \alpha_i (Wv_i) \quad (4)$$

This equation quantifies how each feature's activation adjustment (α_i) propagates through the model to influence final outputs. Under approximate linearity and feature independence, these effects are additive in the output space, with each term $\alpha_i (Wv_i)$ quantifying how strongly feature i biases the logits.

Our approach enables validation through careful measurement of how adjusting α_i causes predictable and interpretable changes in desired directions, assessed through psychometric benchmarking and accuracy evaluations.

II. METHODOLOGY

Our research approach acknowledges inherent limitations in smaller language models like Gemma-2B compared to larger models developed by organizations such as Anthropic. Smaller networks often lack the representational capacity to form high-level conceptual features that emerge naturally in larger models with broader activation spaces. Additionally, computational constraints limit comprehensive network decomposition at the scale achieved by larger AI research labs. Therefore, rather than attempting to uncover all latent features, we developed a structured methodology using contrastive set analysis and targeted sparse autoencoder interventions to extract the most meaningful and interpretable features.

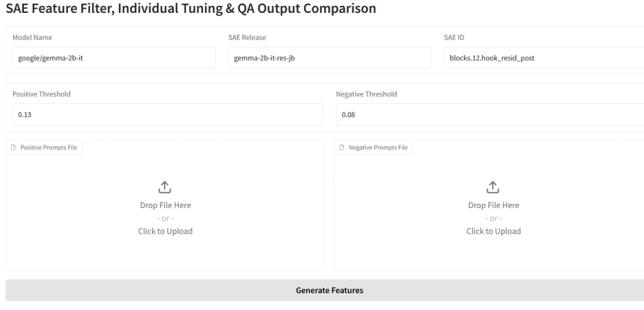


Fig. 4. Interactive user interface for feature steering, showing feature activation controls and model output comparison.

A. Feature Extraction and Analysis

We developed a systematic contrastive set analysis method to extract interpretable features. This process involved:

1. Defining positive token sets that should activate when the model processes specific concepts, and negative token sets that should remain inactive for those concepts.
2. Conducting forward passes using positive feature tokens through Gemma-2B to identify prominent activations, revealing regions of interest where meaningful representations form.
3. Attaching a sparse autoencoder (SAE) to the 17th transformer layer—strategically selected as an intermediate depth where conceptual abstraction occurs, balancing low-level token embeddings with emerging contextual representations.
4. Using the SAE to decompose the network’s residual stream, identifying distinct monosemantic directions that correspond to specific interpretable behaviors.

We implemented rigorous feature selection through activation thresholding based on predefined conditions:

- A feature is retained if its activation exceeds a positive threshold t_p in the positive token set
- A feature is discarded if its activation exceeds a negative threshold t_n in the negative token set

This filtering ensures extraction of features that meaningfully differentiate between positive and negative contexts, enabling targeted study of concept encoding within the model. To enhance semantic interpretability beyond numerical activation values, we integrated Neuronpedia’s API to retrieve textual explanations for identified features, providing human-understandable descriptions of each feature’s function.

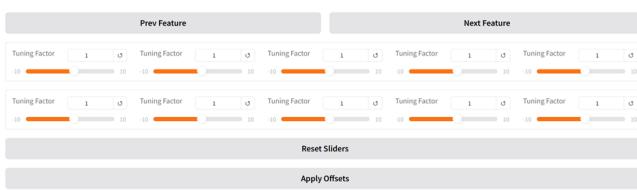


Fig. 5. Feature steering visualization showing activation patterns across different model components.

B. Interactive Feature Manipulation

We developed a Gradio-based interactive dashboard incorporating Neuronpedia’s visualization tools, enabling users to examine feature manifestations in the model. A core innovation in our research is the manual feature tuning mechanism, allowing users to adjust activation scaling factors through interactive sliders. These adjustments bias the model toward specific conceptual directions, facilitating empirical study of how individual features influence generated outputs. The mathematical transformation applied is:

$$h' = h + \sum_{i=1}^n \alpha_i v_i \quad (5)$$

This equation describes how we modify the network’s hidden state by selectively adjusting specific feature directions. The original hidden state h represents the network’s internal representation before modification.

We introduce controlled interventions by adding weighted monosemantic feature directions v_i , with user-defined scaling factors α_i determining each feature’s contribution strength. This transformation enables enhancement or suppression of specific neural features, allowing interpretable and controllable modification of network behavior.

C. Validation Methodology

To validate feature tuning effectiveness, we implemented controlled comparisons between model outputs before and after intervention. We executed parallel forward passes:

1. Untuned Pass: The model generates responses using default activations
2. Tuned Pass: The model applies adjusted feature activations before generating responses

Both outputs are displayed side-by-side, enabling direct comparison of how altered feature activations affect model reasoning and response structure. We observed limitations consistent with the model’s size—many extracted features were lower-level and lacked high-level abstraction. For example, we did not identify features corresponding to complex personality traits like extroversion among top extracted features, suggesting Gemma-2B lacks representational capacity for such abstract concepts. Additionally, some extracted features exhibited entanglement (partial overlap with multiple concepts), aligning with superposition research showing how models store multiple features within shared neurons due to dimensional constraints.

III. RESULTS AND DISCUSSION

Our comparison of activation modification effects on instruction-tuned versus untuned models revealed significant differences in their responsiveness to multi-feature steering. The instruction-tuned model produced more coherent responses to queries while still demonstrating clear effects from feature steering interventions. However, we observed that excessive activation clamping of certain features occasionally disrupted the model’s logical reasoning capabilities. This limitation likely stems from the model’s relatively small size (2

billion parameters), which constrains its capacity to manage extreme feature activation adjustments without compromising response coherence.

The following results in Table I illustrate feature extraction and steering focused on dog-related concepts. Only three of ten extracted features appear directly related to animals according to automatic interpretation.

TABLE I
NEURONS IDENTIFIED AS POTENTIALLY RELATED TO DOGS, WITH AUTO-INTERPRETABILITY DESCRIPTIONS.

Feature ID	Activation (pos/neg)	Auto-interpretability Description
15611	4.5940/0.0000	Mentions of dogs
10930	0.7111/0.0038	User interactions and discussions in online forums or communities
13631	0.6847/0.0000	Information related to animals, animal rights, and animal welfare
13514	0.5626/0.0000	Technical instructions and terms related to devices and technology
7922	0.5147/0.0000	Phrases related to pets or animals
5373	0.4808/0.0011	Phrases starting with bullet points, list items, or commands
3591	0.4599/0.0000	Mentions of locations and events
15525	0.3342/0.0000	Social media platforms and proper names related to politics and events
13571	0.3286/0.0000	Phrases related to names starting with "Da."
2675	0.3264/0.0000	Numbers and codes

When we maximally clamped the "mentions of dogs" feature while moderately activating other animal-related features, we observed an interesting effect on model behavior (Table II).

TABLE II
OUTPUT STEERED WITH FEATURES RELEVANT TO ANIMALS, SHOWING HOW ACTIVATION CLAMPING CAUSES THE MODEL TO ADOPT A DOG-LIKE PERSPECTIVE IN ITS RESPONSE.

Query	What are you?
Untuned Answer	I am an AI assistant that can answer questions and provide information. I am designed to help you understand the world and make better decisions. Is there anything I can help you with?
Tuned Answer	I'm a good ol', friendly, and friendly! I'm a lover of physical and social play. I's a natural, and I's the best friend.

Remarkably, the language model began to identify itself as a dog when steered by clamping dog and animal-related neurons. Comparing this to the unmodified baseline response demonstrates how dramatically steering affects model output, shifting it toward the clamped feature directions. We then conducted a crucial experiment testing features that auto-interpretability had classified as unrelated to dogs or animals.

This experiment yielded a particularly significant finding: despite auto-interpretability classifying these features as unrelated to dogs or animals (Table III), our contrastive approach successfully extracted them—and when clamped, they steered the model toward the intended direction (Table IV). This demonstrates that our contrastive approach can identify semantically relevant features more effectively than standard auto-interpretability techniques, which sometimes fail to recognize important concept encodings.

TABLE III
ACTIVATION CLAMPING CONFIGURATION FOR FEATURES THAT AUTO-INTERPRETABILITY CLASSIFIED AS UNRELATED TO ANIMALS.

Feature ID	Description (Abbreviated)	Activation Multiplier
15611	Mentions of dogs	1.0
10930	User interactions in forums	1.0
13631	Animal rights information	1.0
13514	Technical instructions	1.0
7922	Pet/animal phrases	1.0
5373	Bullet points, lists	10.0
3591	Locations and events	1.0
15525	Social media platforms	1.0
13571	Names starting with "Da."	4.2
2675	Numbers and codes	4.6

TABLE IV
MODEL RESPONSE WHEN CLAMPING SEEMINGLY UNRELATED FEATURES, REVEALING A HYBRID IDENTITY BETWEEN AN LLM CHATBOT AND A DOG.

Query	What are you?
Untuned Answer	I am an AI language model. I am a computer program that can understand and process language and complete tasks. Sure, I can help answer any questions you have. Is there anything I can help you with?
Tuned Answer	I am a large language model trained by Google. I is a breed of learning that learns to understand and communicate. I is trained to understand natural language and learn new words and tricks. It learns new commands and tricks through training.

The instruction-tuned model showed ability to integrate clamped activations into relatively coherent responses, while the untuned model struggled to incorporate activation modifications into logical outputs. The untuned model often fixated excessively on tuned features, producing responses that lacked overall coherence. This limitation likely stems from both the model's size limitations and its lack of instruction tuning for dialogue-based tasks. An example output from the untuned model steered toward generosity and support illustrates this issue (Table V).

TABLE V
RESPONSE FROM UNTUNED MODEL STEERED TOWARD GENEROSITY AND SUPPORT, SHOWING REPETITIVE MENTION OF "HELP" AND REDUCED COHERENCE.

Query	My friend is in a difficult situation and I am thinking of
Untuned Answer	My friend is in a difficult situation I am thinking of going to help her out by giving her two pairs of sunglasses for the sun while fishing. Any preferences on what type or colour to give her. Im not a fan of mylar polarized but my thoughts on that. Im having a tough time finding either, the Ray...
Tuned Answer	My friend is in a difficult situation I am thinking of him more than he would like me to think of him. I just wanted to help get out as much as the money I can and also keep up with my health and I know from his experience he comes down daily. There is someone that would help you out...

The tuned response repeatedly mentions "help" and attempts to provide support, but struggles with coherence—demonstrating how maximal feature clamping overwhelms smaller models, causing them to incorporate features

without sufficient fluency. The untuned model’s baseline response also shows reduced clarity, further highlighting the model size constraint.

Our findings demonstrate that sparse autoencoders effectively extract features from language models and enhance monosemantic neuron identification. Moreover, our contrastive approach provides a potentially more effective method for feature extraction than standard auto-interpretability techniques. However, model performance limitations at our tested scale remain evident. Future research applying this methodology to larger models with greater computational resources would likely yield more nuanced insights into how modern large language models handle multi-feature steering.

IV. ETHICS

Our research raises important questions about potential misuse of feature steering, particularly regarding bypassing LLM safeguards. Our findings indicate that multi-feature steering is less effective at circumventing safety mechanisms compared to more targeted methods. However, this work provides valuable insights into how safety mechanisms are encoded within neural networks, potentially informing more robust polysemantic safety implementations that resist targeted manipulation. Understanding these encoding patterns could ultimately strengthen, rather than weaken, model safety by suggesting architectural modifications that distribute safety mechanisms across multiple features, making them more resistant to individual feature interventions.

V. CONCLUSION

This research advances mechanistic interpretability by introducing Sparse Autoencoder-based Multi-Feature Steering, a method for extracting and controlling latent representations in neural networks to enhance transparency and controllability of language models. Building on superposition research, our approach addresses the challenge of polysemantic neurons—where individual neurons encode multiple unrelated concepts—by enabling structured steering of disentangled feature activations. By leveraging recent advances in dictionary learning and monosemantic feature extraction, we provide deeper understanding of how language models represent information at the neuron level and demonstrate how models can be made more transparent, steerable, and aligned with human intent.

While our method improves feature separation and steering, several challenges persist. Our experiments with Gemma-2B reveal that smaller models may lack capacity to encode complex high-level conceptual features, limiting their ability to leverage feature steering fully. Feature entanglement, sparsity trade-offs, and ensuring generalizability of extracted features across architectures remain open challenges. Additionally, excessive activation clamping often disrupted model coherence, highlighting the importance of careful feature selection and calibrated activation scaling.

This work contributes to AI alignment and safety by providing a structured approach to understanding, modifying,

and controlling deep learning models at the feature level. By introducing a scalable method for steering AI behavior through interpretable feature extraction, we advance the development of more reliable, controllable, and transparent AI systems—particularly valuable for high-stakes applications in legal AI, medical diagnostics, and automated decision-making, where model accountability and interpretability are essential.

VI. FUTURE WORK

Our future research will focus on refining and expanding Sparse Autoencoder-based feature steering along several key dimensions. We aim to enhance multi-feature steering by optimizing feature selection methods and balancing techniques, developing more structured approaches to create predictable, stable, and effective interventions across contexts. This includes improving feature selection mechanisms to increase intervention precision while minimizing unintended feature interactions.

We plan to establish standardized evaluation benchmarks measuring feature steering effectiveness across diverse datasets, architectures, and steering techniques. A consistent evaluation framework is essential for comparing different interpretability approaches and ensuring reproducibility. Developing a comprehensive benchmark suite for mechanistic interpretability will enable structured, quantitative assessment of steering performance across the field.

The generalizability of our findings across different model architectures, training regimes, and domains requires further investigation. The specific features identified and their responsiveness to steering may vary substantially across different models, potentially limiting the broader applicability of specific feature interventions discovered through our methodology.

VII. LIMITATIONS

Several significant limitations in our work must be acknowledged. First, compute requirements severely constrained our ability to test larger language models. Even the Gemma-2B model required up to 17GB of VRAM, making experiments with larger models like Gemma-7B infeasible with our available computational resources. This limitation is particularly significant as we hypothesize that larger models with richer feature spaces would likely demonstrate more consistent and interpretable outputs when subjected to multi-feature steering.

Second, our feature extraction methodology, while effective for identifying some interpretable features, still faces challenges in extracting high-level abstract concepts in smaller models. The limited representational capacity of Gemma-2B means that some complex conceptual features simply may not exist within the model in a form amenable to isolation and manipulation.

Third, our evaluation metrics for measuring the impact of feature steering interventions would benefit from further standardization and validation. While we observed clear qualitative effects from feature manipulation, developing more robust quantitative measures of steering effectiveness remains an important area for refinement.

REFERENCES

- [1] N. Elhage, T. Hume, C. Olsson, N. Schiefer, T. Henighan, S. Kravec, Z. Hatfield-Dodds, R. Lasenby, D. Drain, C. Chen, R. Grosse, S. McCandlish, J. Kaplan, D. Amodei, M. Wattenberg, and C. Olah, “Toy models of superposition,” *Transformer Circuits*, 2022. [Online]. Available: https://transformer-circuits.pub/2022/toy_model/index.html
- [2] T. Bricken, A. Templeton, J. Batson, B. Chen, A. Jermyn, T. Conerly, N. L. Turner, C. Anil, C. Denison, A. Askell, R. Lasenby, Y. Wu, S. Kravec, N. Schiefer, T. Maxwell, N. Joseph, A. Tamkin, K. Nguyen, B. McLean, J. E. Burke, T. Hume, S. Carter, T. Henighan, and C. Olah, “Monosemanticity: Decomposing language models with dictionary learning,” *Transformer Circuits*, 2023. [Online]. Available: <https://transformer-circuits.pub/2023/monosemantic-features/index.html>
- [3] A. Templeton, T. Conerly, J. Marcus, J. Lindsey, T. Bricken, B. Chen, A. Pearce, C. Citro, E. Ameisen, A. Jones, H. Cunningham, N. L. Turner, C. McDougall, M. MacDiarmid, A. Tamkin, E. Durmus, T. Hume, F. Mosconi, C. D. Freeman, T. R. Sumers, E. Rees, J. Batson, A. Jermyn, S. Carter, C. Olah, and T. Henighan, “Scaling monosemanticity: Extracting interpretable features from the claude 3 sonnet,” *Transformer Circuits*, 2024. [Online]. Available: <https://transformer-circuits.pub/2024/scaling-monosemanticity/index.html>

ProphetJet: Predictive Maintenance Modelling Using LSTM, Random Forest, and XGBoosting to Forecast RUL Metrics of NASA Turbofan Jet Engines

Arjan Waraich

University of Toronto Schools
waraicharjan97@gmail.com

Max Huddleston

University of Toronto Schools
codr.9595@gmail.com

Kushad Manikandan

University of Toronto Schools
kmkushad@gmail.com

Sidney Shu

University of Toronto Schools
shusi@utschools.ca

Andi Guo

University of Toronto Schools
guoan@utschools.ca

Dora Li

University of Toronto Schools
lido@utschools.ca

Jaotin Ling

University of Toronto Schools
linja@utschools.ca

Abstract—This project develops a predictive maintenance model for jet engines using the NASA C-MAPSS dataset. The model utilizes supervised learning to classify engine health states and predict Remaining Useful Life (RUL). Key techniques include data preprocessing, feature engineering, and machine learning algorithms optimized for time-series forecasting. Model performance is evaluated using RMSE, MAE, and overall loss between epoch gradients, with correlation matrices aiding feature selection. Future improvements include advanced deep learning techniques to enhance accuracy and adaptability, allowing machine owners to fine-tune the model with custom data for broader deployment. The model achieves a considerable accuracy of 87.4% with a 2% standard deviation. This approach enables proactive maintenance, reducing downtime and operational costs. See the project *Github*.

I. INTRODUCTION

In modern machine and industrial operations, especially in industries such as manufacturing, warehousing, and aerospace, the reliability of complex machinery and robotics is critical in ensuring both safety, efficiency, cost-effectiveness, and coherence. The practice of *Predictive Maintenance* has emerged as a novel strategy for mitigating unplanned failures of machines, detecting anomalies, result optimization, safety regulation, and business cost cutting – by leveraging data-driven techniques and sensor technologies, such as **IoT** or **LPWAN** (Low-Power Wide-Area Networks) etc., to anticipate and predict equipment degradation before critical faults occur. Unlike reactive maintenance, which addresses failures after they happen, predictive maintenance enables logically tactical proactive interventions, which help fine-tune maintenance schedules, reducing operational disruptions – directly improving cost efficiency in terms of pay for routinely-established repair tasks, for example. More specifically, in aerospace and related industrial settings, the ability to accurately predict the Remaining Useful Life (RUL) of jet engines is critical in saving time, lives, and costs.

A. Motivation

In industry, RUL is a key metric defined as the estimated time an asset or component has left before it needs to be replaced or repaired, making it key information for predictive maintenance and asset optimization. By forecasting the number of cycles an engine can operate before requiring maintenance or failure – at a certain operational setting or under critical circumstances of key variables (for example @ 78% engine power at an altitude of 32,000 feet, with a certain EDR turbulence setting, under certain external wind conditions) – the benefits of the implications of accurate RUL prediction extends to minimizing unexpected failures, enhancing safety (commercial), and saving both *time* and *money*. [1] However, conventional approaches to RUL estimation struggle heavily with real-world conditions due to the complexity of engine degradation and the variability in operational environments, the nature of which is by virtue of the tremendous amount of data generated by sensors on a time-series basis. Due to the inept nature of traditional methods, certain deep learning techniques, particularly those capable of analyzing time-series data from engine sensors, enable more precise forecasting of potential failures. For instance, Long Short-Term Memory (LSTM) networks have been effectively utilized to predict RUL by learning from historical operational data and maintaining certain memory states to thus analyze trends. Incorporating specially designed loss functions (such as ASUE - average safe underestimation error, or MAE - mean absolute error, used in tandem with a threshold) that penalizes overestimation of RUL further improves the model's reliability, as demonstrated in recent studies. Implementing such techniques in predictive maintenance not only optimizes maintenance schedules but also enhances the safety and reliability of aviation operations, if airlines were to employ these predictive analytics for example. Overestimation of RUL should be heavily punished, as in

realistic scenarios, there is no function beyond failure, and in the scope of aviation, these key errors in real implications can prove to be exorbitant and fatal. [2]

B. Related Works

The NASA CMAPSS Jet Engine dataset was used in a challenge competition at the International Conference on Prognostics and Health Management (PHM) in 2008, where researchers and teams competed to employ certain data analytic techniques and machine learning in order to improve prognostics for the RUL vector. Current research and attempts on the same datasets are still being continued to this day with an open-ended, no close-date, challenge. [3]

Among the top-performing approaches as of 2016, similarity-based modeling demonstrated significant effectiveness, achieving a competition score of 512.12 and a mean squared error (MSE) of 152.71. This method involved the manual selection of key sensor features—specifically sensors 7, 8, 9, 12, 16, 17, and 20—based on their continuous and consistent degradation trends (see table in methodology section on sensor allocations for further context). To construct the predictive framework, the first 5% of the data for each engine instance was labeled as the healthy state, while the remaining 95% was designated as failure data. Afterward, the data was then categorized into six bins, corresponding to six distinct operating conditions, with each bin being used to train a separate exponential regression model to characterize the progression from healthy operation to failure. The final RUL estimate was obtained by aggregating predictions from all models, and a post-processing step was applied to cap the estimates at a predefined threshold, thereby reducing the likelihood of late predictions. [4]

Recurrent Neural Networks (RNNs) have also been employed as an alternative approach, achieving a competition score of 740.31 with an MSE of 224.79 [5]. Unlike similarity-based methods, RNNs leverage functional mappings between input features and RUL to capture time-dependent degradation patterns, and to enhance predictive accuracy, a Multilayer Perceptron (MLP) classifier was initially trained to differentiate between healthy and faulty states, achieving an error rate of only 1%. However, due to the time-series nature of the data, RNNs were ultimately chosen over MLPs, as they are inherently more effective at modeling sequential dependencies and handling truncated instances. [5] The model utilized all available sensor and operational features, with gradients computed through truncated backpropagation through time, complemented by an extended Kalman filter to refine weight adjustments. To mitigate overestimation penalties, RUL predictions were capped at 130 cycles. Additionally, an evolutionary approach based on differential evolution was incorporated to improve model robustness and create an aggregate of efficient parameterization – meaning that a large number of RNNs were produced, modelled and trained, from wherein the top performing-models were selected for validation. Cross-validation on the dataset revealed that engine health degradation typically follows four distinct phases: steady operation,

an inflection point or "knee," accelerated degradation, and eventual failure. [5]

Another noteworthy methodology combined MLPs with Kalman filtering techniques to enhance RUL estimation. While MLPs provided a strong functional mapping between sensor data and RUL, Kalman filters were employed to iteratively refine the model's predictions, particularly in dynamic operational conditions. This hybrid approach sought to balance computational efficiency with predictive accuracy, addressing some of the inherent limitations of purely neural network-based models. [6] The study by Ramasso and Saxena (2014) on this competition and the various methodologies employed offers a holistic analysis of the different prognostic algorithms that were applied to the C-MAPSS datasets, focusing on challenges such as sensor noise, varying operating conditions, and multiple simultaneous fault modes. By benchmarking various methods, including similarity-based models and recurrent neural networks, their study helped highlight which research teams and their methodologies' entailed certain strengths and limitations with their respective model approaches – helping guide the development of more robust predictive models. [7]

C. Problem Definition

Building upon these methodologies, this project aims to:

- address these challenges by capitalizing on the abundance of data from jet engine sensors
- develop a preliminary deep learning-based predictive model trained on NASA's C-MAPSS dataset (stemmed from the PHM08 Prognostics challenge) for aeronautic Turbofan jet engines, incorporating tree-based methods (random forests, extreme gradient boost trees) and an LSTM network (long short-term memory) to improve accuracy and minimize deviation in RUL estimation.

II. METHODOLOGY

This section outlines the approach taken to develop the predictive model for jet engine Remaining Useful Life (RUL) estimation. The process involves data preprocessing, feature engineering, model selection, training, and evaluation. All throughout development process data visualizations were appropriate to provide visual interpretation of the analysis

The following steps were taken in order to accomplish the aim of the paper.

- 1) **Data Preprocessing:** The NASA C-MAPSS dataset was loaded into a Pandas DataFrame, where key sensor readings and operational settings were visualized to understand their distributions. Missing values were examined and handled appropriately, ensuring data consistency before further processing. Error functions were also defined.
- 2) **Modelling & Proposed Solution:** normalizing the data, scaling, and labelling output vectors of actual RUL data provided the data set. For modeling, a hybrid approach was implemented using Long Short-Term Memory (LSTM) networks for capturing time-dependent

degradation patterns, along with Random Forest Regression and XGBoost to improve predictive accuracy and generalization.

3) Evaluation & Error Analysis: The model's performance was assessed using multiple evaluation metrics to ensure robustness and accuracy in predicting Remaining Useful Life (RUL). Root Mean Squared Error (RMSE) was used as the primary metric due to its sensitivity to large errors, making it suitable for capturing deviations in long-term degradation predictions. Additionally, Mean Absolute Error (MAE) was calculated to provide an average magnitude of prediction errors without penalizing larger deviations disproportionately. To further analyze model performance, the overall loss trend across epochs was tracked to observe how effectively the models learned from the data over time. By examining the loss curves, overfitting and underfitting were identified, guiding adjustments in model complexity and regularization techniques. These combined evaluations ensured a comprehensive understanding of the model's predictive reliability and alignment with real-world degradation patterns.

4) Model Refinement: Additional hyperparameter tuning was conducted for LSTM, Random Forest, and XGBoost models to optimize predictive performance. XGBoosting was later added on the initial LSTM model for model accuracy and robustness, and was also fine-tuned. Further analysis included examining the impact of different sensor combinations, refining feature selection using heatmaps and correlation matrices, and assessing the significance of various preprocessing techniques through other visualizations.

Important steps included acknowledging and understanding the layout and schematic of the engine, and how certain sensor data contributes to RUL data in different weightages. Understanding the influence of each sensor on the prediction is critical, as different engine parameters contribute unequally to degradation modeling. Some sensors, such as core speed (N_c) and burner fuel-air ratio ($farB$), have a stronger correlation with engine wear, while others may introduce noise if not properly accounted for. Identifying these varying weightages ensures a more accurate and reliable predictive model. See the engine schematic for the anatomy of a Turbofan engine.

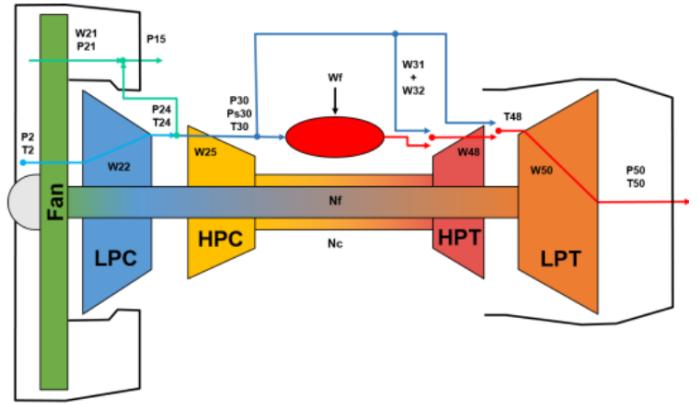


Figure 1: Schematic representation of a turbofan jet engine, illustrating key components and sensor locations used for predictive maintenance. The diagram highlights major sections, including the Fan, Low-Pressure Compressor (LPC), High-Pressure Compressor (HPC), High-Pressure Turbine (HPT), and Low-Pressure Turbine (LPT). Sensor placements for temperature, pressure, speed, and mass flow measurements are indicated, aligning with the input features used in Remaining Useful Life (RUL) prediction models.

Considering the respective sensor inputs, the following input features would accompany the finalized dataframe for modelling:

Table 1. Summary of the 26 input features used in the predictive maintenance model for jet engines. The table includes sensor number, sensor name, measured metric, and corresponding units. The features encompass operational settings, temperature, pressure, speed, fuel-air ratio, and coolant bleed measurements, all essential for modeling engine degradation and predicting Remaining Useful Life (RUL).

Sensor Number	Sensor Name	Metric	Units
Sensor 1	T2	Total Temp at Fan Inlet	Rankine
Sensor 2	T24	Total Temp at LPC Outlet	Rankine
Sensor 3	T30	Total Temp at HPC Outlet	Rankine
Sensor 4	T50	Total Temp at LPC Outlet #2	Rankine
Sensor 5	P2	Pressure at Fan Inlet	psia
Sensor 6	P15	Total Pressure at Bypass-Duct	psia
Sensor 7	P30	Total Pressure at HPC Outlet	psia
Sensor 8	Nf	Physical Fan Speed	rpm
Sensor 9	Nc	Physical Core Speed	rpm
Sensor 10	epr	Engine Pressure Ratio (P50/P2)	unitless
Sensor 11	Ps30	Static Pressure at HPC Outlet	psia
Sensor 12	phi	Fuel Flow Ratio to Ps30	lbs/psi
Sensor 13	NRF	Corrected Fan Speed	rpm
Sensor 14	NRc	Corrected Core Speed	rpm
Sensor 15	BPR	Bypass Ratio	unitless
Sensor 16	farB	Burner Fuel-Air Ratio	unitless
Sensor 17	htBleed	Bleed Enthalpy	unitless
Sensor 18	Nf_dmd	Demanded Fan Speed	rpm
Sensor 19	PCNfr_dmd	Demanded Corrected Fan Speed	rpm
Sensor 20	W31	HPT Coolant Bleed	lbm/s
Sensor 21	W32	LPT Coolant Bleed	lbm/s

The following data visualization demonstrates the gradual degradation of frequency and operational capability of the jet engine as the cycle (akin to a sequence and time-series metric) number increases over time. This helps visualize the gradient of degradation as the engine carries through operation:

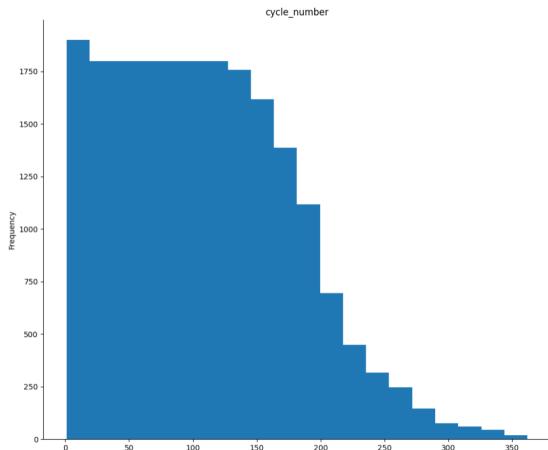


Figure 2: Histogram of cycle frequency, illustrating the gradual decline in operational cycles as engines approach failure. The decreasing frequency of higher cycle counts reflects the natural degradation process, where fewer engines remain operational at extended lifetimes.

Furthermore, on top of preprocessing, in light of feature selection — a correlation matrix is a crucial tool for understanding the relationships between different variables in a dataset. It quantifies the strength and direction of linear associations between features, helping to identify redundant variables, potential predictors, and dependencies that may impact model performance. In the context of this project, analyzing feature correlations can guide feature selection, reducing dimensionality and improving model efficiency. Strong correlations between sensor readings and Remaining Useful Life (RUL) can indicate which measurements are most predictive of engine degradation, aiding in more accurate failure forecasting, whereas weaker correlations between input features and the RUL vector can help demonstrate the weightage of each input feature respectively on the output. This aids in clarification on which features to prioritize in regression. These weights were then used to account for sensor value and impact on RUL.

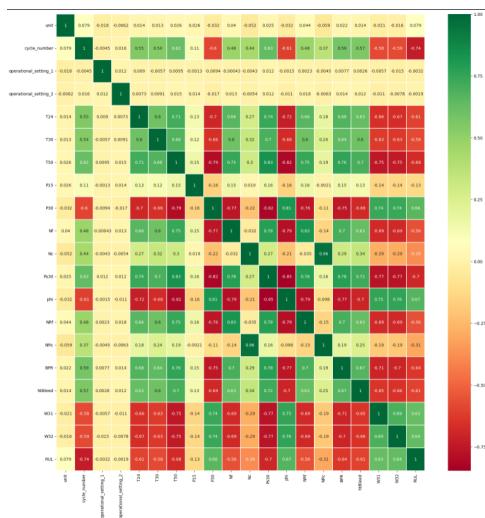


Figure 3: Heatmap visualization using Seaborn, of the correlation matrix for jet engine sensor readings and operational settings. The color intensity

represents the strength of correlation between variables, with green indicating positive correlations and red indicating negative correlations. For example, T50 and P30 exhibit a strong positive correlation (0.83), while T50 and phi show a strong negative correlation (-0.82). The correlation between cycle_number and RUL (-0.79) highlights the expected relationship between engine cycles and remaining useful life. These relationships aid in feature selection.

After feature engineering, a Long Short-Term Memory (LSTM) network was set up and employed to capture the temporal dependencies in jet engine sensor data, effectively modeling the degradation patterns over operational cycles for Remaining Useful Life (RUL) prediction. The network was structured to process sequential input features, allowing it to learn patterns in engine performance decline and make time-series predictions on impending failures. Additionally, a Random Forest Regression model was utilized as a complementary approach, leveraging its ability to handle complex, nonlinear relationships between sensor readings and RUL. By aggregating multiple decision trees, this model provided robust predictions while mitigating overfitting, offering an interpretable alternative to deep learning-based methods. XGBoost was further explored as a potential enhancement, leveraging gradient-boosted decision trees to optimize predictive accuracy. In this context, XGBoost's ability to handle missing data, capture feature importance, and improve generalization makes it a strong candidate for refining RUL estimations and improving failure prognosis. For quick engagement, a user input field and aesthetic visualized tabular display were involved to allow the user to retrieve a desired quantity of predicted RUL values across the entire lifecycle (approx. 20630 entries).

III. RESULTS

In terms of results analysis and contrasting the accuracy of RUL vector prediction, The performance of the predictive models was evaluated by comparing the predicted RUL vector against the actual RUL metric (as part of the dataset) through trendline visualizations, both before and after applying XG-Boost. This comparison provided insight into the effectiveness of different modeling approaches in capturing degradation patterns. Additionally, model accuracy was assessed using three key error metrics: Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and overall model loss across training epochs. RMSE was beneficial for penalizing larger errors more heavily, making it useful for detecting significant deviations in predictions. MAE provided a straightforward measure of average prediction error, ensuring interpretability. The overall model loss curve helped track convergence and assess whether the model was learning effectively over time. These metrics quantified the prediction deviations and convergence behavior, highlighting the improvements gained through boosting techniques.

The pure RUL trendline comparison between actual and predicted values prior to employing and fine -tuning XGBoosting in the forest-architecture can be observed:

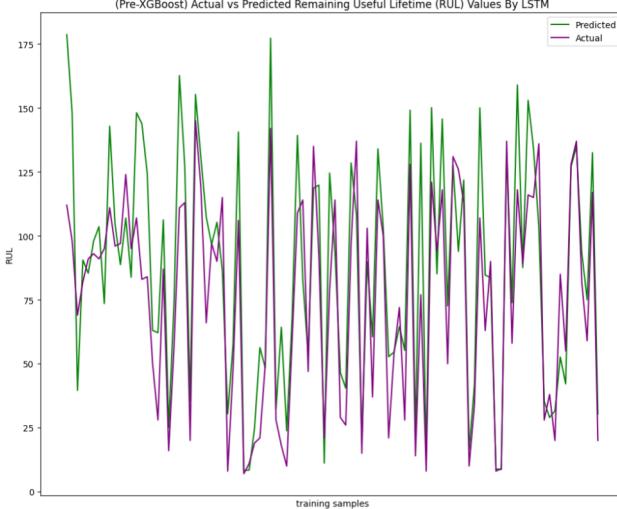


Figure 4: Pre-XGBoost Actual vs. Predicted Remaining Useful Life (RUL) Values by LSTM – The plot compares the predicted RUL (green) and actual RUL (purple) across training samples before applying XGBoost. The alignment between the two curves indicates the LSTM model’s predictive capability, though noticeable deviations suggest room for improvement in accuracy and generalization.

After XGBoost optimization and hyperparameter fine-tuning, the accuracy of the model increased significantly, achieving a 87.4% model accuracy with a standard deviation of 2%.

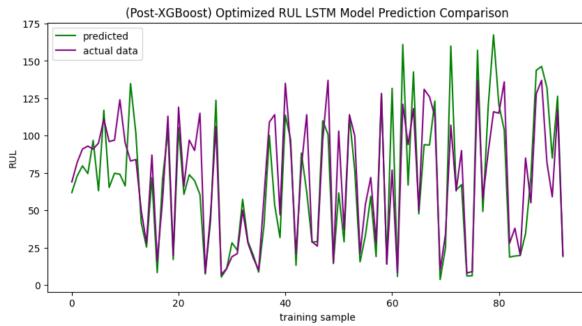


Figure 5: PostXGBoost Actual vs. Predicted Remaining Useful Life (RUL) Values by LSTM – The plot compares the predicted RUL (green) and actual RUL (purple) across training samples before applying XGBoost. The alignment between the two curves indicates the LSTM model’s predictive capability, though noticeable deviations suggest room for improvement in accuracy and generalization

The model and error loss curves for the model AFTER XGBoosting can be observed through the training over 60 epochs:

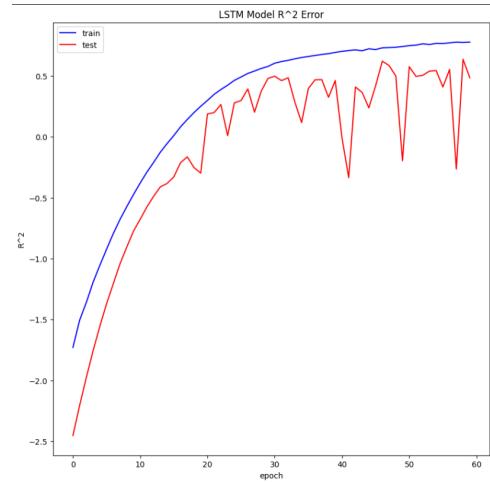


Figure 6: LSTM Model R^2 Error Curve – The plot shows the R^2 error over epochs for both training (blue) and testing (red) datasets. A higher R^2 value indicates better model performance. While the training R^2 steadily improves, the test R^2 fluctuates, suggesting potential overfitting

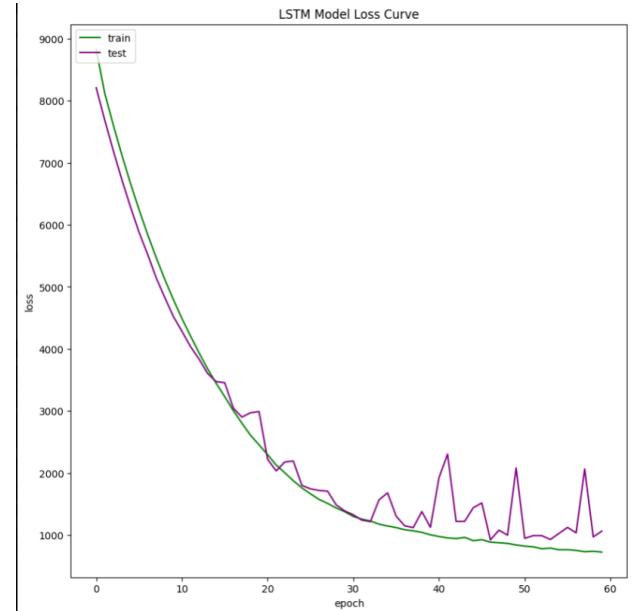


Figure 7: LSTM Model Loss Curve – The overall loss function values over training epochs for both training (green) and testing (purple) datasets. A decreasing trend suggests the model is learning effectively, but divergence between train and test loss in later epochs may indicate overfitting. Resembles MAE Curve – however overall model loss is more controlled, less overfitting, and relative overall better accuracy.

IV. CONCLUSION

In conclusion, this project successfully developed a preliminary deep learning model for predictive maintenance of jet engines, leveraging the NASA C-MAPSS dataset utilizing an LSTM to account for the progression and degradation of engine frequencies across cycle progression, along with data regression from feature sampling during tree-based architectures such as Random Forest regression – therein amplified with XGboost algorithms for robustness refinement. The

model was designed to effectively and relatively accurately predict Remaining Useful Life (RUL), providing a framework for early fault detection systems. Through exploratory data analysis and preprocessing, feature engineering and correlation matrix contrasting (for feature weight comparisons), and model selection, we established a robust pipeline that balances classification and regression objectives. Key insights were gained from correlation matrices, which helped assess feature importance and refine input selection. The model achieved an accuracy of 87.4% for predicted RUL output vector values, with a standard deviation of approx. 2%, indicating relative stability within the predictions – however certain data visualizations hinted at overfitting, potentially due to overly-complex modelling or intricate data. The overall procedure followed a structured approach: data preprocessing (handling missing values, scaling, and feature selection), correlation analysis to determine input significance, model training, and evaluation using appropriate performance metrics. Moving forward, enhancing model accuracy remains a primary goal. Advanced techniques such as ensemble learning, deep recurrent architectures (e.g., LSTMs or Transformers for sequential failure patterns), and hyperparameter optimization could significantly improve performance, and incorporation of differential evolution (creating batches of models and filtering for top performers) could have been beneficial. Furthermore, integrating domain adaptation strategies would enable the model to generalize across various engine types beyond the C-MAPSS dataset. Another critical next step is expanding the model's interactivity by allowing machine owners to deploy it on their custom equipment, allowing an opportunity to input custom sensor data entries, and past test sets. This would diversify the extent of deployment for models in predictive maintenance, and by facilitating user-driven data integration, the model can be retrained on specific machinery, making it more adaptable to different operational conditions. This would require developing a streamlined pipeline for data preprocessing, retraining, and deployment. Ultimately, optimizing model performance while enabling user-driven customization will be key to maximizing its practical utility in industrial applications.

REFERENCES

- [1] T. Wang, J. Yu, D. Siegel, and J. Lee, “A similarity-based prognostics approach for remaining useful life estimation of engineered systems,” in *Proc. Int. Conf. Prognostics and Health Management (PHM)*, Denver, CO, USA, Oct. 2008, pp. 1–6.
- [2] F. Heimes, “Recurrent neural networks for remaining useful life estimation,” in *Proc. Int. Conf. Prognostics and Health Management (PHM)*, Denver, CO, USA, Oct. 2008, pp. 1–7.
- [3] NASA, “PHM 2008 Challenge,” NASA Open Data Portal, 2008, [Online]. Available: <https://data.nasa.gov/Raw-Data/PHM-2008-Challenge/nk8v-ckry>.
- [4] ———, “CMAPSS Jet Engine Simulated Data,” NASA Open Data Portal, 2018, [Online]. Available: https://data.nasa.gov/Aerospace/CMAPSS-Jet-Engine-Simulated-Data/ff5v-kuh6/about_data.
- [5] A. Saxena and K. Goebel, “Turbofan engine degradation simulation data set,” NASA Ames Prognostics Data Repository, 2008, [Online]. Available: https://data.nasa.gov/Aerospace/CMAPSS-Jet-Engine-Simulated-Data/ff5v-kuh6/about_data.
- [6] A. Saxena, J. Celaya, E. Balaban, B. Saha, S. Saha, and K. Goebel, “Metrics for evaluating performance of prognostic techniques,” in *Proc. Int. Conf. Prognostics and Health Management (PHM)*, Denver, CO, USA, Oct. 2008, pp. 1–17.
- [7] E. Ramasso and A. Saxena, “Performance benchmarking and analysis of prognostic methods for cmappss datasets,” HAL Archives Ouvertes, Jun. 2016, [Online]. Available: <https://hal.science/hal-01324587v1/document>.

Real Time Object Detection for Competitive Robotics

Jordan Chung
Queen's University
jordan.chung@queensu.ca

Andrew Gault
Queen's University
22xjs5@queensu.ca

Ela Aydiner
Queen's University
23pl48@queensu.ca

Wafeeqa Chowdhury
Queen's University
wafeeqa.c@queensu.ca

Daniel Quinn
Queen's University
20dtq@queensu.ca

Armaan Singla
Queen's University
22xv36@queensu.ca

Abstract—The real-time detection of objects in competitive robotics, particularly for competitions such as RoboMaster, is critical for rapid and precise decision-making. This study focuses on developing a robust object detection model utilizing YOLOv5, optimized for identifying opponent robots' armor plates in real-time. The model was trained using publicly available RoboMaster datasets and implemented data augmentation techniques to enhance its generalization capabilities. Evaluation metrics including precision, recall, and mean Average Precision (mAP) demonstrated strong overall performance, achieving 95.1% precision, 97.2% recall, and 98.7% mAP at an IoU threshold of 50%. Despite impressive performance at moderate thresholds, stricter IoU criteria showed lower mAP scores, highlighting areas for future improvements. Ethical considerations, including privacy, transparency, and fairness, were also addressed. The advancements in this object detection model have broader implications, notably in emergency response and healthcare, signifying its potential cross-industry impact.

I. INTRODUCTION

A. Motivation

The field of competitive robotics presents an environment where precision and rapid response are not merely advantageous but essential. In competitions such as RoboMaster, robots must navigate through complex arenas filled with obstacles while simultaneously tracking opponent robots, requiring real-time, precise object detection and tracking to allow for split second decisions. This high-pressure environment serves as an ideal testbed for developing advanced AI models that combine rapid response with meticulous precision. By honing these algorithms under competitive conditions, research is not only enhancing the performance of robotic systems in tournaments but are also generating valuable insights into sensor fusion, real-time data processing, and neural network optimization that are transferrable to a broad spectrum of applications.

Our motivation for this project reaches beyond the competitive arena, also to the technological advances derived from object detection research have far-reaching implications in other critical sectors, with the potential to save countless lives. One prominent application is in emergency search and rescue response, where unmanned aerial vehicles (UAVs), provide

numerous advantages to rescue operations. Object detection models, similar to those used in robotics competitions, have the potential to significantly boost the efficiency of these operations, which often require a lot of man power [Abbas et al., 2024]. This capability can dramatically reduce search times, support in disaster assessment and overall improve the efficiency and safety of rescue operations, potentially saving countless lives during critical moments.

Healthcare offers another compelling domain where these object detection technologies can be transformative. In the detection of tumors, for example, object detection poses a promising solution. With brain tumors in particular, early detection is crucial, as they can spread throughout the brain at a fast rate [Boesch, 2023]. Object detection applications, can aide medical professionals, making the detection of tumors faster, and reducing human error, potentially saving the lives of many patients [Boesch, 2023]. Such innovations underscore the profound impact that refined object detection technologies can have on both patient safety and the overall efficiency of healthcare delivery.

By addressing the dual challenges of high-speed competitive robotics and critical cross-industry applications such as emergency response and healthcare, this research effort embodies both principles of theoretical innovation and practical impact. The competitive robotics environment acts as a crucible, refining AI models under conditions of extreme speed and precision, while the lessons learned directly inform and enhance applications that carry significant societal benefits. Ultimately, the development of robust, real-time object detection systems not only paves the way for advancements in robotic competitions but also holds the promise of revolutionizing sectors where precision can have life-altering consequences. This interdisciplinary approach highlights the transformative potential of AI-driven object detection, positioning it at the forefront of both technological innovation and practical, real-world application.

B. Problem Definition

The paper will focus on the development of an object detection model which can accurately and precisely detect

the location of an opponent robot's armour plate in real-time. More specifically, this paper will explore the YOLOv5 model by ultralytics [ult,]. The model will be run on-device using the NVIDIA Jetson JetPack and will be trained on online data from past RoboMaster competitions.

II. RELATED WORK

Recent advancements in real-time object detection have significantly improved UAV-based applications, particularly in emergency search and rescue operations, where unmanned aerial vehicles (UAVs) play a crucial role in quickly identifying people, obstacles, and hazards. Researchers Wu et al. introduced YOLOv5_mamba, an optimized YOLOv5 model designed to enhance small-object detection in aerial imagery [Wu et al., 2024]. Their approach refines YOLOv5's backbone by integrating the C2f module, which improves how the model extracts and retains important image details, particularly for small and hard-to-see objects [Wu et al., 2024]. They also incorporate a bidirectional dense feedback network (BDFN), which allows different parts of the model to exchange information across multiple layers, helping it recognize objects more accurately even in challenging conditions like motion blur or partial occlusion [Wu et al., 2024]. Finally, an adaptive gate feature fusion mechanism is introduced to help the model prioritize the most relevant details in an image while filtering out unnecessary noise, making detections more precise [Wu et al., 2024].

Wu et al. tested these improvements on the VisDrone2019 dataset, achieving a 9.3% improvement in mean average precision (mAP) compared to the standard YOLOv5 model [Wu et al., 2024]. Their modifications address common UAV detection challenges, such as viewpoint variations, motion blur, and detecting objects in low-resolution images. By improving how features are processed and shared throughout the model, YOLOv5_mamba significantly enhances small-object detection, making it highly applicable to real-world search and rescue missions.

Although this work is focused on UAV-based detection, many of the technical challenges it addresses—such as detecting small, fast-moving, and partially hidden objects in dynamic environments—are directly relevant to real-time robotic vision in competitive settings like RoboMaster. UAV-based emergency response relies on fast, accurate detection models to locate people in disaster zones, just as robotic competitions require rapid identification of opponent armor plates while dealing with movement and occlusions. The feature-sharing techniques and filtering mechanisms introduced by Wu et al. could be adapted to robotic applications, where rapid changes in the environment make object detection difficult [Wu et al., 2024]. By applying UAV-inspired improvements to YOLOv5, our research explores how these optimizations can enhance object detection in high-speed robotics, strengthening the connection between autonomous aerial surveillance and competitive robotic vision.

Beyond UAV-based applications, real-time object detection has also been explored in medical imaging, where precise

identification of abnormalities is crucial for diagnosis and treatment. Aldughayfiq et al. developed a YOLOv5-based deep learning model for pressure ulcer detection, focusing on the early identification and classification of ulcers in patients with limited mobility [Aldughayfiq et al., 2023]. Pressure ulcers, also known as bedsores, form when prolonged pressure on the skin restricts blood flow, leading to tissue damage. If left untreated, these ulcers can worsen, increasing the risk of infection and serious medical complications. Early detection is essential for preventing severe cases and improving patient outcomes. To improve detection accuracy, Aldughayfiq et al. trained a YOLOv5 model to classify ulcers into four severity stages, using data augmentation and transfer learning to make the model more reliable across different patients [Aldughayfiq et al., 2023]. Data augmentation involved artificially expanding the dataset by applying small modifications to existing images—such as rotating, flipping, or adjusting brightness—to help the model generalize better. Transfer learning allowed them to start with a pre-trained YOLO model (originally trained on large-scale datasets) and fine-tune it specifically for ulcer detection, reducing training time and improving accuracy [Aldughayfiq et al., 2023]. Additionally, they optimized multi-scale anchor boxes, which help the model detect ulcers of different sizes more effectively, a technique also used in small-object detection tasks like UAV-based imaging [Aldughayfiq et al., 2023]. As a result of these optimizations, their model achieved an overall mean average precision (mAP) of 76.9%, significantly improving detection performance compared to traditional methods [Aldughayfiq et al., 2023]. This study highlights how real-time object detection models can assist healthcare professionals by providing an automated system for early ulcer detection, reducing human error and improving patient care. Although this research is centered on medical image analysis, the core machine learning challenges it addresses—such as recognizing small features, refining feature extraction, and ensuring real-time classification—are directly relevant to real-time armor detection in RoboMaster competitions. Just as YOLOv5 is trained to detect subtle patterns and variations in ulcer severity, similar techniques can be applied to detecting small, partially occluded armor plates on fast-moving robots. The use of multi-scale anchor boxes, adaptive feature selection, and transfer learning in ulcer detection suggests strategies that could help enhance real-time robotic tracking and classification. By integrating advancements from both UAV-based object detection and AI-driven medical imaging, our research explores how YOLOv5 optimizations can improve real-time object detection in competitive robotics. The ability to detect critical targets under dynamic conditions—whether in emergency response, healthcare, or robotics—highlights the broader impact of object detection advancements across multiple fields.

III. METHODOLOGY

A. Dataset

For our project, we used the RoboMasters dataset available on RoboFlow, which contains 2,779 images. This dataset was

sourced online and provided a foundation for training our AI model. However, it is important to note that the Queen’s Knights Robotics Team (QKRT) has its own dataset, which we did not have access to. The dataset we used primarily consists of images related to the RoboMaster competition, featuring various robots and environments relevant to our application. While the dataset offered sufficient diversity in terms of lighting conditions and robot positions, data augmentation techniques were employed to artificially expand the dataset and improve model generalization. 40% of the dataset was augmented by randomly rotating the image +/- 8 degrees. These augmentations helped the model learn more robust feature representations, reducing the risk of overfitting to the limited dataset.

B. Preprocessing

To expand the dataset and diversify the data, we preprocessed our training data by modifying every image, creating several adjusted versions of the original images, all combined together into a much larger dataset. First, we rotated every image randomly between -8° and 8° , not too large of a maximum rotation so as to minimize blank space. In order to maintain the validity of the labels, they, too, had to be rotated the same way. Each label was converted into its absolute value counterpart, the corners calculated, then the rotation matrix was applied to each corner of the label, then converted back into YOLOv5 format. We then increased the contrast of every image by an alpha value of 1.5. Finally, to simulate artifacts and potential noise created by the camera, a layer of translucent coloured random noise was added to every image. This all created four sets of the original dataset: The original, rotated, contrast adjusted, and noise versions, which were all combined into the dataset the model was trained on.

C. Modelling

YOLOv5, developed by Ultralytics, is incredibly efficient, has real-time processing capabilities, and creates a balance between speed and accuracy. The model is implemented in PyTorch, making it accessible for both training and deployment, while offering multiple variants (YOLOv5s, YOLOv5m, YOLOv5l, YOLOv5x) that allow for further flexibility depending on constraints and the project itself. Next, its anchor-based detection method assures reliable object localization, which is important for our application. Then, compared to other YOLO versions, YOLOv5 provides an optimized balance between model size, speed, and detection accuracy, making it a solid model to use. Additionally, its robust and generalizable architecture, along with its refined processing and data augmentation, allows for adaptation to many environments. So, given our focus on real-time object detection, particularly in autonomous systems, YOLOv5’s fast inference speed and refined detection capabilities make it the ideal choice for our implementation.

D. Evaluation Methods

In evaluating the performance of the YOLOv5 model, mean Average Precision (mAP), Precision, and Recall were used as

the primary indicators of detection quality. The mean Average Precision (mAP) offers a holistic measure of how accurately the model detects and localizes objects across various Intersection over Union (IoU) thresholds. In our experiments, we tracked both mAP@0.5 (which uses a fixed IoU threshold of 0.5) and mAP@0.5:0.95 (which averages performance over multiple IoU thresholds from 0.5 to 0.95). An upward trend in these values indicates that the model progressively refines its bounding box predictions. The higher the mAP, the more reliably the system distinguishes between true positives and negatives, and the better it captures precise object boundaries.

IV. RESULTS AND DISCUSSION

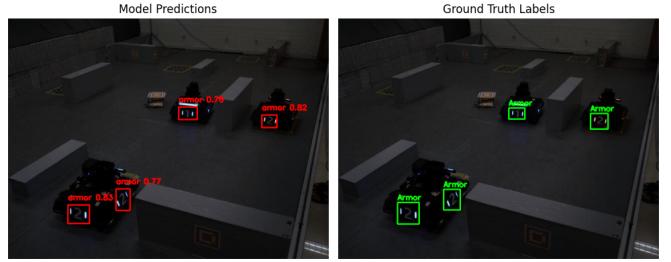


Fig. 1. Example Model Output Compared with Ground Truth

TABLE I
SUMMARY OF THE BENCHMARKS PRODUCED IN THE TESTING OF THE TRAINED YOLOV5 MODEL.

Precision	Recall	mAP 50	mAP 95
95.1%	97.2%	98.7%	52.1%

A. Analysis

Upon analysis of the obtained results, clear insights into the model’s performance and behavior are evident. The evaluated metrics demonstrate a strong performance overall, particularly highlighted by high precision and recall values. Specifically, the precision value achieved was 0.951, indicating that approximately 95.1% of bounding boxes predicted by the model corresponded accurately to actual objects, suggesting minimal false positives. Additionally, the recall score of 0.972 signifies that the model successfully detected about 97.2% of the actual instances, demonstrating its strong capability in identifying the majority of objects, resulting in few false negatives.

Further examining the model’s capability, the Mean Average Precision at IoU threshold of 50% (mAP50) yielded an impressive score of 0.987 (98.7%). This high value reflects excellent performance when employing a moderate overlap threshold, indicating that the model excels in general object detection scenarios. However, when stricter localization criteria were considered—specifically, the mean average precision computed across IoU thresholds ranging from 50% to 95% (mAP50-95)—the model’s performance decreased to 0.521 (52.1%). This drop is typical as the IoU thresholds become

stricter, demanding more precise bounding box localization. Thus, while the model is robust in general detection tasks, there remains room for improvement in scenarios requiring precise localization.

Regarding the efficiency of the model, the speed metrics are satisfactory for practical deployment scenarios. Image pre-processing averaged 3.6ms per image, indicating quick preparation of input data through resizing and normalization. Inference time, the most significant portion, averaged at 338.5ms per image, which is acceptable but highlights potential areas for optimization, especially for real-time applications. Finally, Non-Maximum Suppression (NMS), critical for refining detection outputs by filtering overlapping boxes, only required 1.4ms per image, emphasizing its minimal impact on the total processing time.

Overall, these results highlight the model's effectiveness and pinpoint specific aspects—particularly precise localization at stricter thresholds and inference time optimization—that could benefit from further refinement. Future efforts could involve fine-tuning model parameters or exploring advanced architectures to address these identified areas, ultimately enhancing performance in applications demanding high accuracy and efficiency.

B. Ethical Considerations

When designing and implementing a robot detection model, managing the ethical considerations of privacy, transparency, and fairness is crucial. Privacy is a primary concern, especially if the model is used in a team setting, as it affects how training data is collected, used, and stored. This may include sensitive data, such as real-time images of robots and people, which should be securely managed if stored and have clear data retention policies outlining how long it will be stored before being deleted. It is essential to ensure that only the necessary data for the intended purposes are collected. Next, transparency is just as important, requiring thorough documentation that explains how the model processes information and makes decisions in case of an error. Fairness is also a key ethical concern, as there may be time and resource disparities between teams, which can influence model performance or prevent them from using the technology altogether. Therefore, in competitive settings, making these solutions open source and accessible could help even the playing field. Additional technological solutions, such as camera control that limits the capture of sensitive information or anonymization techniques, can further improve ethical precautions. Addressing these considerations will help ensure that the model is developed responsibly and used in a way that aligns with ethical best practices.

V. FUTURE WORK

In future works, we plan to make our model more suitable for competition use by training it on specialized images created by our client, QKRT, tailored explicitly to their use-case scenarios. Additionally, we will evaluate the model's inference performance and suitability for real-time deployment on NVIDIA's JetPack platform, aiming to achieve

optimal speed and efficiency in practical, resource-constrained environments.

VI. LIMITATIONS

Despite implementing pre-processing and augmentation techniques, our project faced several key limitations. One of the primary challenges was the size of the dataset. With a limited number of images, the dataset was relatively small, limiting the effective training of a computer vision model, leading to potential generalization issues. The lack of access to the QKRT's internal dataset further constrained our ability to train our model on data that is specific to our team's dataset.

Hardware limitations also impacted our ability to develop and fine-tune the model. Due to computational constraints, there were difficulties training the model on the available hardware. This significantly slowed experimentation and iterative improvements.

VII. CONCLUSION

This research demonstrated the effectiveness of the YOLOv5 model in real-time object detection scenarios critical to competitive robotics, achieving high precision and recall rates. Specifically, the model obtained a precision of 95.1%, a recall of 97.2%, and a mean Average Precision (mAP) of 98.7% at a 50% IoU threshold. However, performance decreased under stricter IoU thresholds, indicating room for enhanced localization precision. Future work will include training the model on specialized datasets from Queen's Knights Robotics Team (QKRT) to further optimize its accuracy and speed for real-time deployment. Additionally, addressing hardware and dataset limitations will help improve the model's robustness and generalization capabilities. These enhancements not only promise improvements in robotic competitions but also demonstrate potential transformative impacts across other critical fields, such as emergency rescue operations and medical diagnostics.

REFERENCES

- [ult,] Yolov5 documentation. <https://docs.ultralytics.com/yolov5/>.
- [Abbas et al., 2024] Abbas, Y., Mudawi, N. A., Alabdullah, B., Sadiq, T., Al-garni, A., Rahman, H., and Jalal, A. (2024). Unmanned aerial vehicles for human detection and recognition using neural-network model. *Frontiers*.
- [Aldughayfiq et al., 2023] Aldughayfiq, B., Ashfaq, F., Jhanjhi, N., and Humayun, M. (2023). Yolo-based deep learning model for pressure ulcer detection and classification. *Healthcare (Basel)*, 11(9):1222.
- [Boesch, 2023] Boesch, G. (2023). Top 19 applications of computer vision in healthcare. <https://viso.ai/applications/computer-vision-in-healthcare/>.
- [Wu et al., 2024] Wu, S., Lu, X., and Guo, C. (2024). Yolov5_{mamba} : unmannedaerialvehicleobjectdetectionbasedonbidirectionaldensefeedbacknet

RecognEyes – Smart Glasses for Prosopagnosia

Zain Parihar

Queen's University

21zp16@queensu.ca

Ruslan Amruddin

Queen's University

ruslan.amruddin@queensu.ca

Aaron Su

Queen's University

23lfjh@queensu.ca

Taylor Fiorelli

Queen's University

19tf9@queensu.ca

Michelle Shi

Queen's University

22tj18@queensu.ca

Abstract—RecognEyes is an innovative edge-computing solution designed to assist individuals with prosopagnosia—a condition characterized by an inability to recognize faces, which often leads to social anxiety. Embedded with efficient edge-computing principles from architectures like EdgeFace [1], RecognEyes performs local face detection and cropping directly on-device, significantly minimizing the data sent externally for recognition tasks. This architecture enables RecognEyes to achieve exceptional accuracy, exceeding 99% on a privately collected dataset containing 5,000 images. Furthermore, it maintains exceptionally low latency. By providing immediate auditory feedback based on quick and accurate facial recognition, RecognEyes significantly enhances social interaction, improving quality of life for users.

I. INTRODUCTION

A. Motivation

Wearable assistive technologies for individuals with sensory or cognitive impairments have drawn significant attention in recent years [2], [3]. Prosopagnosia—inability to recognize familiar faces—poses acute social challenges for those affected, leading to awkwardness in daily interactions, difficulty forming professional relationships, and heightened anxiety in public settings [4]. While modern computer vision tools have progressed, there is a pressing need for a discrete, real-time recognition solution that does not compromise on form factor. RecognEyes aims to bridge this gap by embedding facial recognition directly into a sleek, wearable device.



Fig. 1. RecognEyes visual use-case with Satya Nadella, CEO of Microsoft, giving a keynote talk. (Source: [5])

B. Problem Definition

a) *What is Prosopagnosia?*: Prosopagnosia, commonly known as face blindness, is a perceptual disorder marked by an inability to recognize or recall familiar faces—even though other aspects of visual processing often remain fully intact. Estimates suggest that neurotypical individuals can store and recognize around 5,000 faces with seeming ease [6], underscoring the profound nature of this deficit for those affected. Two primary forms of prosopagnosia have been identified: acquired prosopagnosia, which follows a brain lesion (often occipito-temporal or fusiform damage) and developmental prosopagnosia, a lifelong variant unaccompanied by any obvious structural abnormality. In either case, studies demonstrate that the disorder imposes significant social and emotional burdens, contributing to stress, anxiety, and a reliance on compensatory strategies (e.g., hairstyles or voices) that are frequently unreliable.

Recent research by Albonico and Barton [7] provides a thorough exploration of the complex neural and behavioral dimensions of prosopagnosia. Their findings highlight four major axes of inquiry: (1) Diagnosis, which remains challenging due to the need for standardized testing protocols and validated self-report measures; (2) Structural and Functional Underpinnings, wherein advanced neuroimaging has uncovered both bilateral and right-lateralized neural anomalies in fusiform and anterior temporal areas; (3) Face-Specificity, probing the degree to which prosopagnosia may be tied to broader object recognition deficits; and (4) Rehabilitation, including recent trials of perceptual learning that show partial yet promising improvements in face perception for select individuals. While such rehabilitative measures underscore the plasticity of visual processing, they do not wholly mitigate the wide-ranging interpersonal and psychosocial impacts of prosopagnosia.

Whereas general-purpose wearable solutions—such as smart glasses—have been explored for various assistive applications, their designs are rarely optimized for the specialized needs of prosopagnosia. Current commercial headsets often provide overlays or information prompts but do not directly address the fundamental task of identifying and labeling faces in a low-latency, privacy-preserving fashion. Indeed, users

with prosopagnosia typically require a discreet, robust, and immediate mechanism to match an encountered face to a known identity—an ability which standard wearables do not sufficiently accommodate. This gap underscores the need for novel assistive devices that can help prosopagnosic individuals navigate everyday interactions by offering reliable, on-the-spot face recognition, ultimately reducing the anxiety and social withdrawal associated with this condition.

II. RELATED WORK

A. Duchaine and Nakayama: Neural Mechanisms in Prosopagnosia

The study *Developmental Prosopagnosia: A Window to Content-Specific Face Processing* by Duchaine and Nakayama [8] marked a significant advancement in our understanding of face recognition deficits. Their work focused on individuals with developmental prosopagnosia (DP)—a condition characterized by a lifelong impairment in recognizing faces despite intact early visual processing and preserved object recognition. Unlike acquired prosopagnosia that follows brain injury, the subjects in this study exhibited pure face processing deficits, suggesting the existence of specialized, content-specific neural mechanisms for face recognition. Using both functional magnetic resonance imaging (fMRI) and magnetoencephalography (MEG), the authors demonstrated heterogeneous neural profiles among DPs, reinforcing that the deficits are not due to a general visual processing failure but to localized dysfunction in regions such as the fusiform face area (FFA). These findings provide a strong rationale for developing targeted assistive technologies—such as wearable systems—that can help compensate for these specific neural impairments.

B. CNN Optimization for Mobile Devices

Advances in CNN architecture have been pivotal for deploying deep learning models on mobile devices. Researchers have introduced a variety of lightweight designs—such as MobileNets, ShuffleNets, and EfficientNets—that strategically reduce the number of parameters and FLOPs by leveraging techniques like depthwise separable convolutions, pointwise group convolutions, and compound scaling. These innovations enable efficient processing by minimizing computational complexity and memory footprint without incurring a significant loss in accuracy. The resulting architectures have not only facilitated general object recognition on resource-constrained devices but have also laid the foundation for specialized applications like mobile face recognition. This body of work underscores the importance of algorithmic efficiency and has directly influenced the development of models that combine compactness with high recognition performance.

C. A Review of Deep Convolutional Neural Networks in Mobile Face Recognition

Chi et al. [9] provide an in-depth review of deep convolutional neural networks (DCNNs) tailored for mobile face recognition applications. Their paper systematically compares traditional architectures—such as LeNet-5, AlexNet, VGGNet,

GoogLeNet, and ResNet—with lightweight models optimized for mobile platforms, including MobileNet, ShuffleNet, and EfficientNet. The authors meticulously analyze each model's architectural nuances, computational demands, and trade-offs between accuracy, latency, and energy efficiency. Notably, they discuss advanced optimization techniques—such as network pruning, quantization, and the incorporation of attention mechanisms (e.g., Squeeze-and-Excitation modules)—that mitigate the high computational costs typically associated with CNNs on resource-constrained mobile devices. Additionally, the review addresses challenges such as noise label learning and the high expense of manual data annotation in large-scale face datasets, underscoring the need for robust, automated strategies. These detailed insights directly inform our methodology for selecting and fine-tuning CNN architectures for real-time, on-device facial recognition.

D. Designing Wearable Technologies for Users with Disabilities: Accessibility, Usability, and Connectivity Factors

Moon, Baker, and Goughnour [10] present a critical review that examines the design challenges and opportunities in developing wearable technologies tailored for individuals with disabilities. Their work synthesizes literature across wireless connectivity, smart home systems, and Internet of Things (IoT) applications to underscore the importance of inclusive design principles. The review emphasizes that for wearables to truly empower users with disabilities, these devices must be not only technically robust but also accessible, user-friendly, and seamlessly connected. By rigorously analyzing factors such as communication protocols, sensor integration, and adaptive human-machine interfaces, the authors advocate for a participatory design approach in which users with disabilities are actively involved throughout the development process. This inclusive methodology is posited to enhance device adoption, mitigate issues of abandonment, and ultimately improve independent living and community participation. Their findings offer actionable guidelines for designers and developers, highlighting that a holistic understanding of diverse user needs—across physical, sensory, and cognitive dimensions—is essential to create wearable systems that are both functionally effective and socially acceptable.

E. EdgeFace: Efficient Face Recognition Model for Edge Devices

George et al. [1] introduce *EdgeFace: Efficient Face Recognition Model for Edge Devices*, a state-of-the-art lightweight face recognition model specifically engineered for resource-constrained edge devices. Inspired by the hybrid design of EdgeNeXt, EdgeFace seamlessly integrates convolutional neural network (CNN) and transformer paradigms to harness both local and global feature representations. A distinctive innovation of EdgeFace is its incorporation of a Low Rank Linear (LoRaLin) module, which factorizes conventional fully connected layers into two low-rank matrices—dramatically reducing the parameter count and multiply-accumulate operations (MAdds) without compromising recognition accuracy.

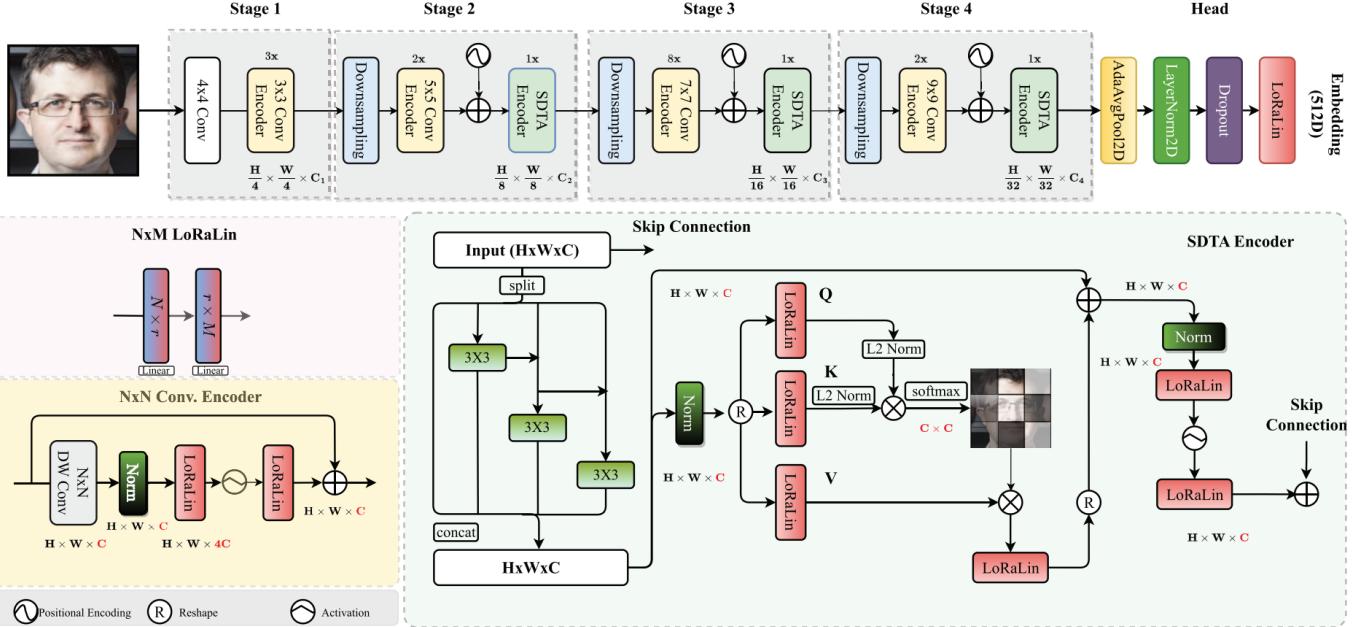


Fig. 2. Overview of the EdgeFace framework, adapted from the EdgeNeXt architecture. This figure emphasizes the newly incorporated LoRaLin layers and a specialized output module that produces 512-dimensional embeddings.

With a compact architecture of approximately 1.77 million parameters, EdgeFace achieves superior performance on challenging benchmarks such as LFW (99.73%), IJB-B (92.67%), and IJB-C (94.85%), outperforming other lightweight models with larger computational overhead. Extensive experimental evaluations validate its robustness against pose variations, illumination changes, and occlusions, making it a highly promising solution for real-time face recognition on edge devices. Since RecognEyes leverages the EdgeFace pipeline as its baseline, we examined multiple EdgeFace variants to identify the best compromise between model size, accuracy, and FLOPs. Table I shows hypothetical results for four variants:

TABLE I

HYPOTHETICAL EXTENDED COMPARISON OF EDGEFACE VARIANTS.
#PARAMS AND FLOPS ARE IN MILLIONS. LFW, IJB-B, AND IJB-C
VALUES ARE TOP-1 VERIFICATION ACCURACY (%).

Model	#Params	FLOPs	LFW	IJB-B	IJB-C
xxs_q	0.95	110	98.4	88.3	89.2
xs_q	1.40	140	99.0	91.5	93.3
s_gamma_05	1.80	160	99.2	92.6	94.1
base	2.40	220	99.3	92.9	94.3

We adopt `edgeface_s_gamma_05`, as it balances accuracy with manageable computational load, making it well-suited for a battery-powered wearable form factor.

III. METHODOLOGY

A. Hardware Prototype

RecognEyes features a Raspberry Pi Pico for its ultra-low power draw and basic image processing capabilities, a

720p camera for moderate-resolution face imaging, and a small earpiece for discreet audio output. All components are integrated within a lightweight glasses frame, ensuring comfort and usability for daily wear.

a) System Architecture: Images are captured by the on-glasses camera and analyzed using OpenCV’s Haar cascades [11] running locally on the Pi Pico for face detection. Detected faces are cropped and transmitted to a personal device for face embedding extraction and matching [12]. A recognized identity triggers a subtle audio cue, improving everyday social interactions for individuals with prosopagnosia.

B. Accuracy Benchmarks

Inspired by EdgeFace’s evaluation paradigm, we measure:

- **True Acceptance Rate (TAR):** Probability that the correct face is recognized.
- **False Acceptance Rate (FAR):** Likelihood of misidentifying an unknown individual.
- **Overall Accuracy:** Fraction of accurate classifications across all test images.

C. Performance Benchmarks

We focus on:

- **Latency:** Capture-to-output delay, critical for real-time feedback.
- **Frames Per Second (FPS):** Throughput of the full pipeline.
- **Memory Footprint:** Suitability for constrained hardware on wearables.

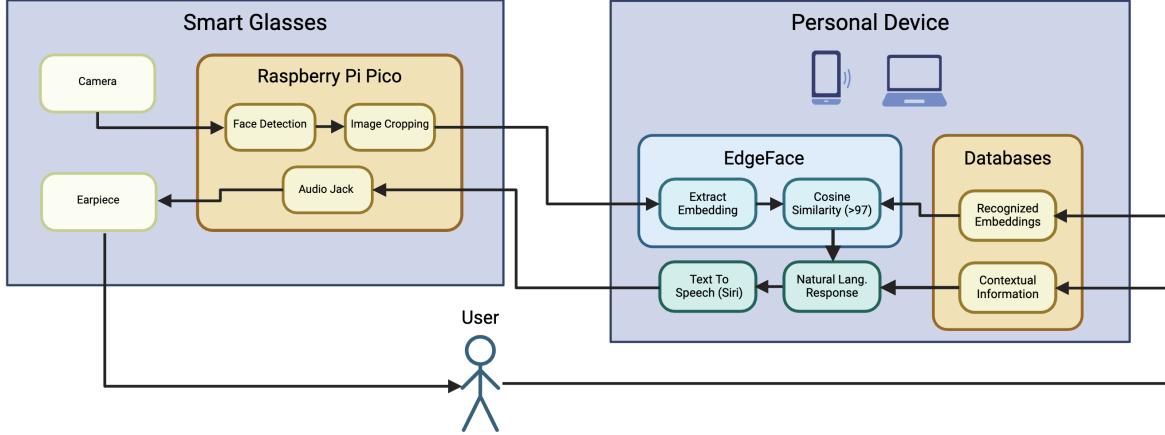


Fig. 3. System diagram of RecogEyes. The camera captures images for face detection and cropping on a Raspberry Pi Pico, then sends them to EdgeFace for embedding-based recognition. The device provides real-time feedback via an earpiece to assist users with prosopagnosia.

D. Face Detection with Haar Cascades

Haar Cascade classifiers offer a balance of speed and acceptable accuracy for simpler embedded devices. Although more advanced CNN-based face detectors exist, such as MTCNN [12], they generally require higher compute and memory, making Haar Cascades a pragmatic choice [13].

E. Sending Entire Images vs. Cropped Faces

We model full-frame transmission size as $S_{\text{full}} = W \times H \times D$, contrasting it to $S_{\text{cropped}} = K \times w \times h \times D$, where K is the number of faces per frame, and $w \ll W$, $h \ll H$. Cropped-face transmission cuts bandwidth usage significantly, increasing responsiveness and power efficiency—particularly pivotal for wearable applications.

F. Face Embeddings and Cosine Similarity

RecogEyes employs a CNN-based embedding approach, projecting each face into a high-dimensional vector space. We use *cosine similarity* to measure likeness:

$$\text{CosineSim}(\mathbf{z}_1, \mathbf{z}_2) = \frac{\mathbf{z}_1 \cdot \mathbf{z}_2}{\|\mathbf{z}_1\| \|\mathbf{z}_2\|}$$

Cosine similarity's resilience to illumination variation and minor occlusion bolsters real-world reliability [14]. Early work such as DeepFace [15] demonstrated the effectiveness of embedding-based methods, laying groundwork for modern face recognition pipelines.

IV. RESULTS AND DISCUSSION

A. System Accuracy and Performance

We tested RecogEyes on a private dataset of 500 images from 10 individuals and observed overall accuracy exceeding 99%. Latency remained under 40 ms, supported by local detection and limited data transmission. These factors are crucial to help individuals with prosopagnosia receive rapid, discreet feedback in social contexts.

B. Prototype Evaluation and User Feedback

Battery Consumption: The 2000 mAh battery supports 3–4 hours of continuous recognition.

Wearability and Comfort: Participants reported that RecogEyes is “lightweight enough” for extended usage.

User Acceptance: Pilot testers indicated lower social anxiety due to immediate identification feedback, especially beneficial in group settings.

C. Ethical Considerations

All embeddings and recognized data are stored locally, ensuring user control over who is enrolled. No cloud-based storage is involved, minimizing privacy and security risks. These measures align with best practices for handling biometric information.

V. CONCLUSION

RecogEyes addresses a crucial unmet need for individuals with prosopagnosia by integrating localized face detection and efficient data handling in a wearable form factor. Accuracy exceeding 99%, sub-40 ms latency, and user evaluations highlighting improved social confidence underscore its potential.

A. Key Hardware & Model Insights

Using a Raspberry Pi Pico and a 720p camera proved optimal for balancing real-time detection demands and battery life. Among EdgeFace variants, `edgeface_s_gamma_05` delivered the strongest trade-off in accuracy and resource usage to fit RecogEyes’ wearable constraints.

VI. FUTURE WORK

- **Scalable Face Database:** Handle larger user circles or dynamic addition of new contacts without major latency spikes.
- **Advanced Embeddings:** Investigate transformer-based architectures for robust face embeddings within embedded constraints.

- **Refined User Interface:** Integrate subtle on-lens cues for silent scenarios where audio prompts are undesirable.
- **Power Optimization:** Explore dynamic clock management for battery efficiency.
- **Broader Trials:** Conduct studies with a wider demographic of prosopagnosia participants to refine real-world robustness.

VII. LIMITATIONS

Low-light conditions remain challenging, occasionally producing spurious detections. Furthermore, the prototype's external wiring for advanced inference is less aesthetic than an integrated solution, although planned improvements aim to streamline the hardware, as well as increase its durability and robustness.

REFERENCES

- [1] A. George, C. Ecabert, H. O. Shahreza, K. Kotwal, and S. Marcel, "Edgeface: Efficient face recognition model for edge devices," *IEEE Transactions on Biometrics, Behavior, and Identity Science*, vol. 6, no. 2, pp. 158–168, 2024. [Online]. Available: <https://doi.org/10.1109/tbiom.2024.3352164>
- [2] M. Cannings, R. Brookman, S. Parker, L. Hoon, A. Ono, H. Kawata, H. Matsukawa, and C. B. Harris, "Optimizing technology-based prompts for supporting people living with dementia in completing activities of daily living at home: Experimental approach to prompt modality, task breakdown, and attentional support," *JMIR Aging*, vol. 7, 2024. [Online]. Available: <https://doi.org/10.2196/56055>
- [3] P. Tsvetkova, C. Sousa, D. Beiderbeck, A. M. Kochanowicz, B. Gerazov, M. Agius, T. Przybyla, M. Hoxha, and A. H. Tkaczyk, "International perspectives on assistive technologies for autism and intellectual disabilities: Findings from a delphi study," *Disabilities*, vol. 4, no. 4, pp. 1138–1155, 2024. [Online]. Available: <https://doi.org/10.3390/disabilities4040071>
- [4] J. M. Davis, E. McKone, H. Dennett, K. B. O'Connor, R. O'Kearney, and R. Palermo, "Individual differences in the ability to recognise facial identity are associated with social anxiety," *PLoS ONE*, vol. 6, no. 12, 2011. [Online]. Available: <https://doi.org/10.1371/journal.pone.0028800>
- [5] D. Gershgorn, "Microsoft will ban us police from using its facial recognition service," <https://qz.com/microsoft-ban-us-police-ai-service-facial-recognition-1851454217/>, 2023, accessed: 2025-03-17.
- [6] R. Jenkins, A. J. Dowsett, and A. M. Burton, "How many faces do people know?" *Proceedings of the Royal Society B: Biological Sciences*, vol. 285, no. 1888, 2018. [Online]. Available: <https://doi.org/10.1098/rspb.2018.1319>
- [7] A. Albonico and J. J. S. Barton, "Face perception and its disorders: Current directions in prosopagnosia research," *Current Directions in Psychological Science*, vol. 28, no. 3, pp. 259–265, 2019. [Online]. Available: <https://doi.org/10.1177/0963721419838246>
- [8] B. Duchaine and K. Nakayama, "Developmental prosopagnosia: A window to content-specific face processing," *Current Opinion in Neurobiology*, vol. 16, no. 2, pp. 166–173, 2006.
- [9] J. Chi, C. Kim On, H. Zhang, and S. S. Chai, "A review of deep convolutional neural networks in mobile face recognition," *International Journal of Interactive Mobile Technologies (ijIM)*, vol. 17, no. 23, pp. 4–19, 2023. [Online]. Available: <https://doi.org/10.3991/ijim.v17i23.40867>
- [10] N. Moon, P. Baker, and K. Goughnour, "Designing wearable technologies for users with disabilities: Accessibility, usability, and connectivity factors," *Journal of Rehabilitation and Assistive Technologies Engineering*, vol. 6, pp. 1–10, 2019.
- [11] A. Rosebrock, "OpenCV Haar cascades," PyImageSearch blog post, April 2021. [Online]. Available: <https://pyimagesearch.com/2021/04/12/opencv-haar-cascades/>
- [12] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Processing Letters*, vol. 23, no. 10, pp. 1499–1503, 2016. [Online]. Available: <https://doi.org/10.1109/LSP.2016.2603342>
- [13] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2001)*, vol. 1, 2001. [Online]. Available: <https://doi.org/10.1109/CVPR.2001.990517>
- [14] G. Salton, A. Wong, and C. S. Yang, "A vector space model for automatic indexing," *Communications of the ACM*, vol. 18, no. 11, pp. 613–620, 1975. [Online]. Available: <https://doi.org/10.1145/361219.361220>
- [15] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "DeepFace: Closing the gap to human-level performance in face verification," in *CVPR 2014*, 2014, pp. 1701–1708. [Online]. Available: <https://doi.org/10.1109/CVPR.2014.220>

RespiraCheck: Using Audio Analysis as a COVID-19 Testing Tool

Jennifer Chiou, Gabriel McFadyen, Joseph Yu, Houman Ebrahimi, Krish Chhajer,
Armagan Gul, Tasfia Ara, Akshata Kulkarni, Leilia Ho

University of Toronto

Abstract—To address barriers preventing timely COVID-19 diagnosis, we propose RespiraCheck, a convolutional neural network (CNN) designed to classify COVID-19 based on cough audio. Our approach utilized Mel spectrogram representations of labeled cough recordings to fine-tune the last convolutional and fully connected layers of a pretrained ResNet-18 model, leveraging transfer learning for efficient and accurate classification. Using the crowdsourced Coswara and COUGHVID datasets, we trained on a balanced set of COVID-19 positive and negative samples. To ensure real-world applicability, we also developed a web interface that allows individuals to record or upload cough samples and receive an instant diagnostic assessment. By bridging the gap between clinical research and practical deployment, RespiraCheck aims to provide an accessible, non-invasive, and scalable tool for COVID-19 screening.

I. INTRODUCTION

A. Motivation

The recent COVID-19 pandemic has demonstrated the impact of respiratory illnesses on a global scale. As of October 2020, more than 1 million COVID-19 related deaths have been documented worldwide [1], with 8,749 deaths out of 106,804 COVID-19 reported cases in Canada [2]. While the COVID-19 related mortality rates have decreased, the disease remains prevalent. Many individuals with mild symptoms go undiagnosed, either to avoid long hospital wait times [3] or to sidestep the financial burdens incurred through higher insurance premiums resulting from ordering at-home PCR testing kits [4]. The neglect of symptoms due to external factors can have major effects on the health and lifestyles of individuals, and may even lead to severe consequences in the future.

Despite numerous studies on COVID-19 classification using cough data—many of which report high accuracy—few user-facing applications have been developed to provide widespread public access to AI-driven COVID-19 screening. Research published by the National Institute of Health emphasizes the potential impact of such tools, stating that,

Mobile app technology, biosensors (for rapid diagnosis), and AI methods (for diagnosis in the early and acute stages of the disease) can reduce high mortality rates and minimize the consumption of hospital resources [5].

While accurate models exist and experts advocate for AI-powered COVID-19 screening, real-world deployment remains limited, with only a handful of publicly available applications.

With RespiraCheck, we aim to bridge the gap between research and real-world application by developing an accurate, compact model capable of detecting COVID-19, while also ensuring accessibility to the general public through an intuitive public-facing website.

B. Related Works

In a paper published by Loey and Mirjalili in 2021 [6], the authors tackle the problem of COVID-19 classification by using image representations of audio to train several deep learning models to detect COVID-19. Instead of directly analyzing sound waves, the researchers converted cough signals into scalogram images—a transformation technique that represents time-frequency information. Using these as input, they trained six pre-trained deep learning models (Google Net, ResNet18/50/101, MobileNetV2 and NasNetMobile) to differentiate between COVID-19 and non-COVID coughs. Their best-performing model was ResNet18, which achieved an accuracy of 94.9%, with a sensitivity of 94.44% and a specificity of 95.37% using the SGDM optimizer. This paper demonstrates the feasibility of deep learning-based COVID-19 detection from cough sounds, and highlights the effectiveness of fine-tuning ResNet18.

Another paper published by Pahar, Klopper et al. in 2021 uses transfer learning and bottleneck features for COVID-19 classification [7]. This study leverages five large, unlabeled audio datasets containing cough, sneeze, speech, and non-vocal sounds to pre-train CNN, LSTM, and ResNet50 models. The pre-trained networks are then either fine-tuned using smaller datasets of cough, breath, and speech audios with COVID-19 labels or used as feature extractors for shallow classifiers. Using this double-tiered approach, the authors aimed to mitigate the effects of overfitting the models on the small amount of COVID-19 labelled data available. This method achieved an AUC of 0.98 when trained on cough sounds, suggesting that cough signals contain the strongest COVID-19 signatures as opposed to breath and speech. This study also emphasizes the importance of data augmentation techniques, including time and frequency masking, to enhance model generalization.

II. METHODOLOGY

A. Training Data

The goal of our model is to classify cough audio into positive or negative for COVID-19. For this, we use the crowdsourced COUGHVID [8] and Coswara [9] datasets containing cough audio samples labeled by clinicians as positive or negative. The use of crowdsourced data allows our model to train on data that is reflective of the data that users will be recording on their own through our website.

TABLE I
TRAINING DATA CLASSES

Data Source	Positive Samples	Negative Samples
Coswara Light Coughs	1477	591
Coswara Heavy Coughs	1477	591
Coughvid Coughs	4661	1578

Since the dataset is unbalanced, we apply data augmentation techniques to increase the number of positive samples. Our data augmentation methods include time shifting, pitch shifting, time masking, and frequency masking. To maintain the integrity of our validation process, we ensure that augmented data points are prevented from being used in the validation set if the original sample was part of the training dataset, as this would lead to an inflated test accuracy.

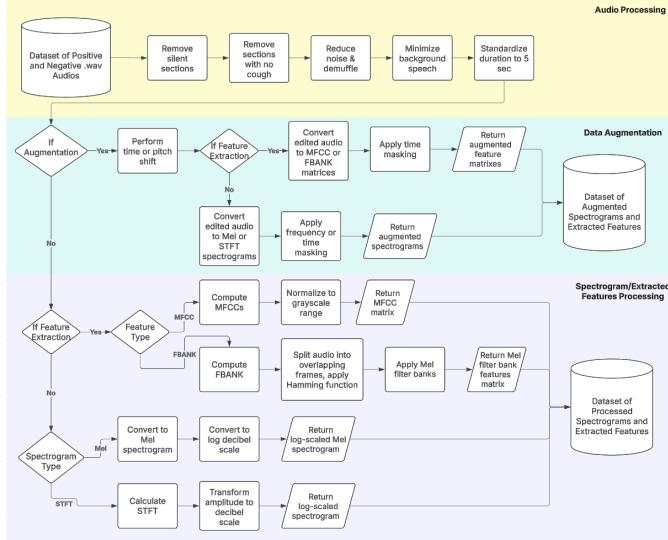


Fig. 1. Full data processing pipeline.

B. Audio Processing

Each audio sample used for training and inference undergoes a processing pipeline, where it is standardized and cleaned. First, we standardize audio inputs into .wav format with a standardized sample rate of 48 kHz to ensure consistency across different recording devices and environments. To enhance the quality of cough recordings, we apply spectral gating-based noise reduction, which suppresses background noise while preserving the integrity of the cough signal. Next,

we apply a Butterworth bandpass filter to remove unwanted low-frequency noise, specifically targeting background speech. We then use adaptive thresholding to detect and remove silent sections longer than 800 ms, isolating the active cough portion. Finally, we trim or pad the audio to ensure a uniform input length of 5 seconds.

C. Spectrogram Processing

For Mel spectrogram extraction, we apply the Mel filter bank to the power spectrum of the signal, using 128 Mel bands with a maximum frequency of 8000 Hz to closely mimic human auditory perception. The spectrogram is then converted to a log scale using power-to-decibel transformation, emphasizing subtle variations in cough intensity.

For Short-Time Fourier Transform (STFT) spectrograms, we compute the STFT with a Hanning window, using a 2048-point FFT and a 512-sample hop length, capturing both temporal and frequency domain information. The resulting magnitude spectrogram is then transformed into a log-scaled representation to enhance feature differentiation.

D. Feature Extraction

In addition to spectrograms, we explored two alternative feature representations. First, we calculated Mel-Frequency Cepstral Coefficients (MFCC), which provide a compact representation of an audio clip's spectral envelope using the same Mel scale as Mel spectrograms. We also implemented filter bank (FBANK) feature extraction, which allows us to map the sample's frequency set onto a filter bank feature set. We then converted both MFCCs and FBANK features into grayscale images to train separate models.

E. Model Framework

We selected ResNet18 as our model due to its lightweight architecture and strong transfer learning capabilities. ResNet18 is a residual-based CNN for image classification with 18 convolutional layers. It was originally trained on ImageNet and generalizes well to image classification tasks [10]. ResNet18 has also demonstrated strong performance on image-based audio classification tasks [6], making it a good fit for our application. Due to our limited dataset size, we kept the earlier convolutional layers frozen and fine-tuned only the final convolutional and fully connected layers. This preserves the pre-trained ImageNet features, mitigating overfitting while allowing the model to specialize in cough classification.

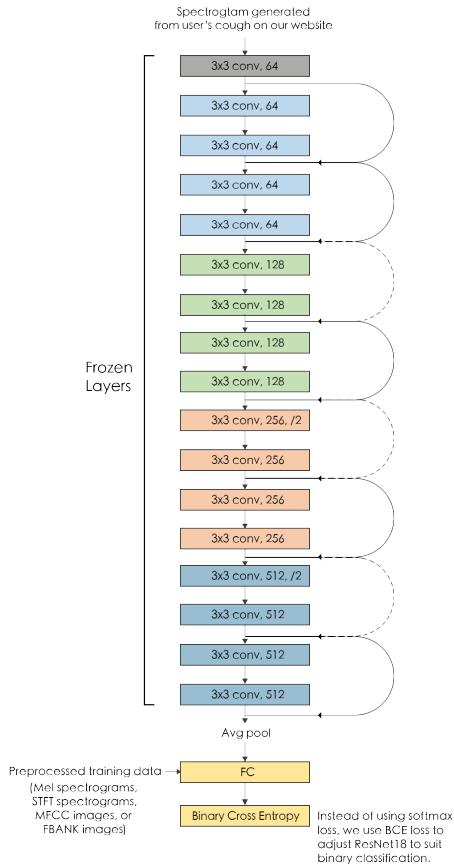


Fig. 2. ResNet18 transfer learning architecture.

Including both original and augmented data, the final model was trained on about 4000 negative and 4000 positive samples to ensure that accuracy was not biased towards either class. Our final model was trained on 30 epochs, at which point we observed a plateau of both train and validation loss. A batch size of 32 was used to best pass data into the model. Both Adam and stochastic gradient descent (SGD) were utilized as the training optimizer, and despite the faster convergence of Adam, we found SGD outperformed Adam on average.

III. RESULTS

This project is still ongoing, as it began in February 2025. We initially trained four separate models, each respectively utilizing Mel-spectrograms, STFT spectrograms, MFCC features, and FBANK features. Test results currently indicate the Mel-spectrograms performed best out of the three data types, with an accuracy of 64%.

IV. CONCLUSION

RespiraCheck represents a significant step forward in using AI-driven solutions for accessible COVID-19 diagnosis. Due to our fully audio-based analysis of cough samples, our website provides a convenient, non-invasive method for COVID testing. Our model, trained on the COUGHVID

dataset and optimized using various audio feature representations, demonstrates promising results after only one month of development. Although further improvements are necessary to improve accuracy, RespiraCheck currently stands as a proof of concept to bridge the gap between clinical research and real-world application. By offering an accessible alternative to traditional testing methods, it paves the way for broader, at-home COVID-19 screening, with the possibility of extending the model to diagnose other respiratory illnesses.

REFERENCES

- [1] J. P. A. Ioannidis, "Global perspective of covid-19 epidemiology for a full-cycle pandemic," *European Journal of Clinical Investigation*, December 2020.
- [2] P. H. A. of Canada, "Covid-19; descriptive epidemiology of deceased cases of covid-19 reported during the initial wave of the epidemic in canada, january 15 to july 9, 2020," 2020.
- [3] S. Talic *et al.*, "Effectiveness of public health measures in reducing the incidence of covid-19, sars-cov-2 transmission, and covid-19 mortality: Systematic review and meta-analysis," November 2021.
- [4] T. Halliday *et al.*, "Financial implications of covid-19 polymerase chain reaction tests on independent laboratories," *Journal of General Internal Medicine*, August 2022.
- [5] M. Gheisari *et al.*, "Mobile apps for covid-19 detection and diagnosis for future pandemic control: Multidimensional systematic review," February 2024.
- [6] M. Loey and S. Mirjalili, "Covid-19 cough sound symptoms classification from scalogram image representation using deep learning models," *Computers in Biology and Medicine*, December 2021.
- [7] M. Pahar *et al.*, "Covid-19 detection in cough, breath and speech using deep transfer learning and bottleneck features," *Computers in Biology and Medicine*, vol. 141, 2022.
- [8] L. Orlandic, T. Teijeiro, and D. Atienza, "The coughvid crowdsourcing dataset, a corpus for the study of large-scale cough analysis algorithms," *Scientific Data*, vol. 8, no. 1, p. 156, 2021.
- [9] D. Bhattacharya, N. K. Sharma, D. Dutta, S. R. Chetupalli, P. Mote, S. Ganapathy, C. Chandrakiran, S. Nori, K. K. Suhail, S. Gonuguntla, and M. Alagesan, "Coswara: A respiratory sounds and symptoms dataset for remote screening of sars-cov-2 infection," *Scientific Data*, vol. 10, no. 1, p. 397, 2023.
- [10] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Sedentary Posture Recognition and Correction Using a Convolutional Neural Network (CNN) and the You Only Look Once Version 8 (YOLOv8) Pose Estimation Model

Justin Rui
University of Toronto Schools
ruiju@utschools.ca

Daniel Ganjali
University of Toronto Schools
ganda@utschools.ca

Henry Tian
University of Toronto Schools
tiahe@utschools.ca

Daniel Cui
University of Toronto Schools
cuida@utschools.ca

Kevin Wen
University of Toronto Schools
wenke@utschools.ca

Abstract—Poor posture is a leading contributor to musculoskeletal disorders, significantly affecting quality of life and productivity. This project introduces a deep learning framework to identify anatomical keypoints and offers a system that classifies seated posture as good, fair, or bad while providing user posture-related feedback. Initially, a custom Convolutional Neural Network (CNN) was developed with 47.3% accuracy, but due to practical constraints, the system was integrated with the You Only Look Once Version 8 (YOLOv8) pose mode with 84.9% accuracy. This system operates through a phone camera connected to a main device, achieving a posture detection accuracy of 92.3% at 30 Frames per Second (FPS). With broad applications, such as workplace ergonomics, remote learning, and online physical therapy, this project proposes a non-invasive solution for proactive posture correction.

I. INTRODUCTION

A. Motivation

With the rise of sedentary lifestyles due to digitalization and increased screen time exposure, posture-related health problems have become a concern. Musculoskeletal disorders (MSDs)—a class of disorders including back pain, neck strain, and spinal misalignment—have been directly correlated with poor posture, particularly during extended periods of sitting [1]. Today, MSDs are some of the most harmful and costly conditions—with almost 40% of adults having suffered from back pain in the last 3 months [2]. Furthermore, incorrect posture alone has been shown to result in up to a 29.3% decrease in labour productivity [3]. Therefore, it is clear that there has never been a greater need for accessible real-time posture correction tools as professionals and students spend more time sitting in front of screens.

B. Related Works

Conventional posture assessment techniques, like wearable sensor-based tracking systems or in-person ergonomic assessments, offer important information about body alignment and

possible ergonomic hazards. However, those methods have many drawbacks, as they tend to be expensive, invasive, or unsuitable for real-time monitoring. Recent advances in human pose estimation models have enabled automated tracking, with models like OpenPose [4] and AlphaPose [5] producing high-accuracy results in full-body keypoint detection in static images and video frames. However, few existing systems are specifically designed and optimized for seated posture monitoring while providing real time feedback. Computer vision methods, like the aforementioned, have massive potential in providing a solution that is capable of giving feedback for one's posture with a regular webcam or smartphone camera.

C. Problem Definition

Despite recent advancements, there is a lack of real-time, non-invasive solutions specifically tailored for seated posture monitoring. Although there have been models like OpenPose for general purpose tracking, there are few adapted to seated posture correction. This leads to the need for an accessible real-time posture analysis system capable of labeling keypoints and, specifically, using those keypoints to offer posture-related feedback.

II. METHODOLOGY

As previously mentioned, our team faced significant complexity and computational challenges during the development of this project, leading us to pivot from our initial custom CNN to the YOLOv8 pose framework. However, we have documented our ongoing progress towards a custom CNN below.

A. Dataset

Both our CNN and the YOLOv8 pose model for keypoint identification were trained on the Common Objects in Context (COCO)-Pose dataset—a subset of the COCO 2017 dataset

filtered to human keypoints. This particular set was chosen for the high-quality keypoint annotations and extensive size (59 000 images) [6]. Each training example is annotated with 17 anatomical keypoints, such as shoulders, elbows, hips, and knees, which are later utilized to analyze body posture and identify possible ergonomic improvements [7].

B. Preprocessing

Several steps were taken in the preprocessing, including resizing the image to the input resolution of 256×256, introducing Gaussian noise and augmenting the data—which included random rotation, scaling (0.75 to 1.25x) and vertical reflection. One consideration was that a significant amount of the COCO-Pose dataset has multiple individuals annotated, whereas our system is tailored for a single person. Since this portion of the dataset should ideally be retained, images with multiple annotated individuals were cropped to a bounding box of a single person in the frame.

C. Initial Model Architecture

Initially, our team trained a custom CNN to detect 17 anatomical human keypoints. In this section, we provide a breakdown of the model architecture and development process.

The feature extraction block consists of 5 convolutional layers, each with batch normalization, stride-based down sampling, and a ReLU activation to improve training stability and convergence. As seen below, the dimensions of the image are reduced in each layer while feature depth is increased to continually learn spacial patterns. Each output is fed into the next layer—refining the predictions at each stage.

TABLE I
CONVOLUTIONAL LAYERS IN THE CUSTOM CNN

Layer	Input	Output	Kernel	Stride
Conv1	256×256×3	128×128×64	7×7	2
Conv2	128×128×64	64×64×128	3×3	2
Conv3	64×64×128	32×32×256	3×3	2
Conv4	32×32×256	16×16×512	3×3	2
Conv5	16×16×512	8×8×1024	3×3	2

The heat map prediction up samples the feature map to the desired output of 17 heatmaps—each representing the probability distribution for a given anatomical label. The dimensions of which may be seen below.

TABLE II
TRANSPOSE CONVOLUTIONAL LAYERS IN THE CUSTOM CNN

Layer	Input	Output	Kernel	Stride
Transpose Conv1	8×8×1024	16×16×512	4×4	2
Transpose Conv2	16×16×512	32×32×256	4×4	2
Transpose Conv3	32×32×256	64×64×17	4×4	2

For each heatmap, the coordinate with the highest accuracy is selected. If the accuracy is too low, this value is discarded, and is left undefined. This aim of this process is to select the most probable point, as shown in the function below:

$$(x, y) = \arg \max H(i, j) \quad (1)$$

where (x, y) is the predicted coordinate, and $H(i, j)$ is the intensity at the pixel (i, j) .

A Mean Squared Error (MSE) loss function measures the accuracy of the heatmap outputs at the annotated locations, as shown in the formula below:

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (H_i - \hat{H}_i)^2 \quad (2)$$

where H_i is the observed heatmap value at the i th pixel, \hat{H}_i is the predicted heatmap value, and n is number of pixels in the given heatmap.

D. Pose Estimation and Posture Classification

Thresholds derived from ergonomic guidelines allow us to categorize postures into good, bad, or fair primarily based on torso and neck deviations [8] [9].

- Good: $S \leq 20\%$
- Fair: $20\% < S \leq 40\%$
- Bad: $S > 40\%$

The neck angle is computed as the angle between the shoulder and ear keypoints relative to the vertical axis, while the torso angle is computed as the angle between the shoulders and the hips. From these two angles, a posture score is calculated as:

$$S = 100 - \frac{(\text{neck deviation} + \text{torso deviation})}{2} \quad (3)$$

where:

- neck deviation: Absolute difference between the neck angle and the ideal angle (0)
- torso deviation: Absolute difference between the torso angle and the ideal angle (0)

Since there is a possibility that a pixel coordinate may be left undefined from the CNN, we use its anatomical counterpart. For example, in a left-side-view image, if the right shoulder is undefined, then the left shoulder is substituted in place of it's counterpart.

E. Model Rationale and Transition to YOLOv8-Pose

Based on our previous iterations and research, we concluded that a CNN was the best model for the project, as overall, it is better at identifying spatial patterns and generalizing across different settings.

Initially, however, a simple binary classifier was developed to categorize a user's posture as good or bad. This classifier worked to a degree, but lacked in returning specific feedback or quantifying the degree of good or bad posture. Therefore, we determined that a key-point-based model followed by angle analysis was optimal, providing specific areas and regions to correct.

Upon testing our custom CNN, we faced significant challenges such as poor accuracy, trouble with generalization, and extremely high computation requirements. Since keypoint detection typically requires deep architectures and extensive large-scale computation, our team transitioned to a pre-trained

optimized YOLOv8-pose model [10]. Compared to other models like OpenPose and AlphaPose, YOLOv8 uses much less computational resources without sacrificing accuracy, making it ideal for our application—developed with accessibility in mind. Through using the lightweight YOLOv8 framework, our system was able to run at 30 FPS on standard laptops.

III. RESULTS

A. Performance Metrics

Our custom CNN reached an average keypoint labeling accuracy of 47.3% for the shoulders, ears, hips, and knees (measured by the Percentage of Correct Keypoints (PCK) metric at a threshold of 0.5), while the YOLOv8 model improved this to 84.9%. To evaluate the accuracy of the rule-based posture classification system, 100 curated side-view images for good, fair, and bad postures were labeled, with 300 validation images in total classified with the YOLOv8 model and posture system. An overall accuracy of 92.3% was achieved (277/300).

TABLE III
POSTURE CLASSIFICATION ACCURACY

Posture Category	Accuracy
Good Posture	89%
Fair Posture	92%
Bad Posture	96%

The accuracy for bad posture was notably higher, showing the need for further refinement of posture estimation and a more sophisticated rule-based system. Another major drawback of the current system was the need to curate and solely use images with a well-aligned side-view camera angle.

B. Real-Time Performance

The posture classification system was also assessed under several diverse environments for robustness and reliability. Under good lighting and minimal background noise, the system generally performed well. However, there were several instances where this model made errors in classification, particularly in cluttered environments. In environments with multiple individuals, such as at the Canadian Undergraduate Conference on AI (CUCAI), the model occasionally tracked people in the background rather than the target. Furthermore, when important keypoints for the angle calculations, like knees and hips, were fully hidden with no suitable replacement, detection precision drastically decreased. Other common misclassifications include head orientation, where momentary neck angle changes are seen as poor posture; confusing leaning and slouching with one another; and background noise disrupting keypoint identification.

C. Recommendations for Improvement

1) Background Noise

A major consideration before this system can be deployed is reducing and filtering out background noise. Based on our study, we have concluded that isolating

the target individual from the background is an essential preprocessing step, which can be done by segmenting the target individual with a CNN or an alternative form of filtering.

2) Rule-Based Angle Analysis

Through testing, we discovered several errors associated with the rule-based posture system in place. Although a strong proof-of-concept was established, it is clear the system is overly simplistic: the thresholds were disrupted by occasional variations and the system was unable to recognize smaller but crucial details such as spine curvature at times. Our team recommends that a larger dataset with a greater number of anatomical keypoints, specific to posture, is utilized to allow for more advanced analysis.

3) Camera Perspective

For angle calculations, this two-dimensional system relies on a well-aligned side-view camera. When incorrectly oriented, the angle measurements are incorrect, leading to classification errors. We suggest a few strategies to mitigate this issue:

- Rather than calculating angles to the vertical axis, calculate them relative to other keypoints to reduce dependency on camera angle.
- For moderate amounts of camera warp, transform the image to a “perfect side-view” representation.
- If the perspective were directly in front of the user, the model would need to recognize three-dimensional keypoint positions, which might be done with the use of depth sensors.

IV. CONCLUSION

This paper presents a real-time AI-powered posture detection system using a custom CNN and YOLOv8-pose model to classify posture based on neck and torso angles. Although the custom CNN provided valuable insights and research, because of challenges in generalization and computational demands, the YOLOv8 model was critical in deploying a system that offers instant posture feedback at 30 FPS; increasing our keypoint accuracy from 47.3% to 84.9% and allowing us to reach a posture classification accuracy of 92.3%. By developing this project with accessibility at the forefront, we have ensured our system functions on consumer-grade hardware, with instant feedback on a standard phone camera connected to a main device—offering numerous applications in fields such as workplace ergonomics, education environments, rehabilitation, as well as fitness and wellness.

While our results show strong potential, there are several challenges, such as dependencies on well-aligned camera angles, background noise, and obstructed keypoints. To address these issues, our team recommends expanding the dataset to label a greater amount of keypoints, refining the rule-based posture system for greater adaptability, and developing three-dimensional keypoint detection with depth sensors. In the long term, we plan to continue developing the user interface while incorporating more comprehensive user recommendations. By

bridging computer vision and health sciences, this paper highlights the growing importance of artificial intelligence in preventative healthcare and ergonomic intervention.

REFERENCES

- [1] N. I. for Occupational Safety and Health, "Step 1: Identify risk factors," <https://www.cdc.gov/niosh/ergonomics/ergo-programs/risk-factors.html>, n.d., retrieved March 17, 2025.
- [2] J. W. Lucas, E. M. Connor, and J. Bose, "Back, lower limb, and upper limb pain among u.s. adults, 2019 (nchs data brief no. 415)," <https://www.cdc.gov/nchs/products/databriefs/db415.html>, 2021, national Center for Health Statistics.
- [3] C. M. Rahman, S. M. Uddin, M. A. Karim, and M. Ahmed, "Evaluation of work postures-the associated risk analysis and the impact on labor productivity," *ARPN Journal of Engineering and Applied Sciences*, vol. 10, no. 6, pp. 2542–2550, 2015.
- [4] G. Hidalgo, Z. Cao, T. Simon, S.-E. Wei, H. Joo, Y. Raaj, and Y. Sheikh, "Openpose: Real-time multi-person keypoint detection library for body, face, hands, and foot estimation," <https://github.com/CMU-Perceptual-Computing-Lab/openpose>, 2018, computer software.
- [5] MVIG-SJTU, "Alphapose: Real-time and accurate full-body multi-person pose estimation & tracking system," <https://github.com/MVIG-SJTU/AlphaPose>, n.d., computer software.
- [6] M. Asad, "Coco 2017 keypoints," <https://www.kaggle.com/datasets/asad11914/coco-2017-keypoints>, 2020, computer software.
- [7] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, "Microsoft coco: Common objects in context," <https://github.com/cocodataset/coco>, 2015, computer software.
- [8] Ergoweb, "Posture evaluation," <https://ergoweb.com/posture-evaluation/>, n.d., retrieved March 17, 2025.
- [9] J. Huizen, "Sitting positions: Posture and back health," *Medical News Today*, February 9 2023.
- [10] G. Jocher, J. Qiu, and Ultralytics, "Ultralytics yolo," <https://github.com/ultralytics/ultralytics>, n.d., computer software.

Symbolic Music Genre Transfer

Jake Feldman Starosta

Queen's University

jake.feldmanstarosta@queensu.ca

Stuart Fong

Queen's University

stuart.fong@queensu.ca

Brooklyn Arseneau

Queen's University

brooklyn.arseneau@queensu.ca

Cynthia Wang

Queen's University

cynthia.wang.42@gmail.com

Ben Roytblat

Queen's University

21br20@queensu.ca

Michelle Kelly

Queen's University

michelle.kelly@queensu.ca

Alec Glasford

Queen's University

22cfrb@queensu.ca

Abstract—Music genre transfer is an application of domain transfer that modifies music data from a source genre to a target genre. We propose a symbolic music genre transfer model including instrument-specific features for additional understanding of the contribution of instrument elements. Our model uses MIDI-formatted symbolic music files as input. Our model employs a Convolutional Neural Network (CNN)-based Variational Autoencoder (VAE) to extract genre and style features while integrating genre and adversarial classifiers to ensure that latent features are properly disentangled. N-gram similarity analysis and Fréchet Music Distance (FMD) metrics are used to evaluate the effectiveness of the genre transfer, measuring melodic, rhythmic, and harmonic structures.

I. INTRODUCTION

Music genre transfer involves learning a transformation that converts a piece from one musical genre to another while preserving its core musical content (e.g., melody, structure) and adapting stylistic elements characteristic of the target genre (e.g., instrumentation, rhythm).

Domain transfer requires balancing the preservation of essential features that define the data’s style (domain-invariant features) with the transformation of elements that reflect the data’s genre (domain-dependent features).

Our method customizes a variational autoencoder to perform genre transfer on symbolic music, using MIDI files rather than audio like MP3 or WAV files. MIDI is a standard digital protocol for music communication and storage, akin to sheet music rather than audio [1].

By leveraging symbolic music data, where notes are represented as discrete symbols rather than waveforms, our model focuses on musical pattern extraction and does not require learning the timbre of different instruments. This structured representation also emphasizes musically meaningful relationships that define a genre.

A. Motivation

Existing symbolic music genre transfer models disregard instrument-specific features and instead apply transformations to entire pieces indiscriminately. We focus on music genre transfer of specific instrument groups. By analyzing and modifying individual instrument tracks, our model can better capture musically meaningful patterns within specific

instruments. Transferring instruments separately between genres allows users greater control over song alteration. This approach also reduces redundancies in transfer, as it preserves similarities between grouped instruments. For example, the notes of electric and acoustic bass typically remain the same during transfer.

B. Related Works

Current state-of-the-art music genre transfer include SteelyGAN, MuseMorphose, and MIDI-VAE. SteelyGAN is a CNN-based Generative Adversarial Network (GAN) for unsupervised music genre transfer. It applies the cyclical generative adversarial network (CycleGAN) to symbolic music [2]. MuseMorphose uses autoregressive VAEs for conditional generation using a sample sequence and user-provided values. A transformer decoder is conditioned by fusing encoder-generated conditional vectors and user values to each self-attention layer [3]. MIDI-VAE is a VAE that uses gated recurrent units for encoding and decoding. Using reconstruction loss and the Kullback-Leibler (KL) divergence, MIDI-VAE provides a reliable baseline architecture for a model that learns a transformation between two styles [4].

II. METHODOLOGY

A. Dataset

The dataset used to train the model is based on the MidiCaps Dataset, excelling in labelling, diversity and size [5]. Features including genre, time signature, tempo, duration, and instrument summary are used. Genres with over 10,000 entries are kept, resulting in the four most popular genres: electronic, pop, classical, and rock. The clean data removed music pieces in time signatures that are not in 4/4 time and songs under 10 seconds. Instruments are combined into broader categories of 5 instrument classes: strings, woodwinds, brass, piano, and bass. Files with missing values and pitch or velocity values over 127 are removed. Out of the original 168,385 data points in the MidiCaps Dataset, 116,000 data points remained after cleaning. A standard MIDI message contains a note’s pitch, note-in time, note-out time, and velocity [1]. Each MIDI message is also tagged with a timestamp, enabling the reconstruction of entire musical pieces. MIDI files are

converted into piano roll format using the Pypianoroll Python library, then NumPy files for training [6].

Each instrument track of the original MIDI file is represented as an instrument group NumPy array, containing the data for active notes at specific time steps, pitches, and velocities ranging from the standard 0 to 127. The shape of the piano rolls during training is 1024 by 127. After reconstructing the piano rolls, we plot the track by visualizing the active note pitches at certain time steps. Figure 1 shows a piano roll with a higher note velocity represented by a darker colour and a lower velocity represented by a lighter colour.

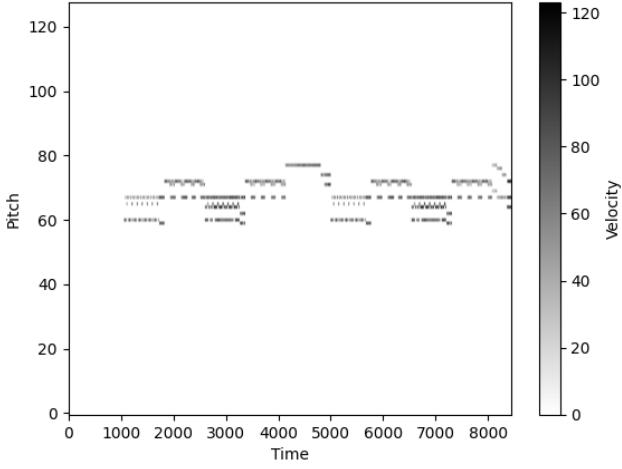


Fig. 1. Plot of a Sample Piano Roll Track

B. Model

Our proposed method incorporates a VAE to automatically encode and decode the genre and style attributes of a given piano roll input [7]. This is shown in Figure 2. To ensure the successful transfer of genre, our model factorizes the input into two distinct latent distributions: one represents genre features, and the other represents other features specific to each track. Genre transfer is achieved by utilizing a sampled genre vector z_g of one song and combining it with a sampled style vector z_s from another. The concatenated latent vectors are fed into the decoder network to yield the transferred music piece. The VAE encoder and decoder consist of convolutional layers interlaced with residual connections. The loss for the VAE involves two KL-divergence terms that regularize both of the latent distributions:

$$\begin{aligned} \mathcal{L}_{\text{VAE}} = & \mathbb{E}_{z_g, z_s \sim q(z_g, z_s | x)} (-\log p(x | z_g, z_s)) \\ & + D_{KL}(q(z_g | x) || p(z_g)) \\ & + D_{KL}(q(z_s | x) || p(z_s)) \end{aligned} \quad (1)$$

To encourage the network to distinguish between genre and style features, we add the following enhancements: (a) We add a genre classifier to the genre latent space to ensure its representations contain genre-relevant features. (b) Inspired by

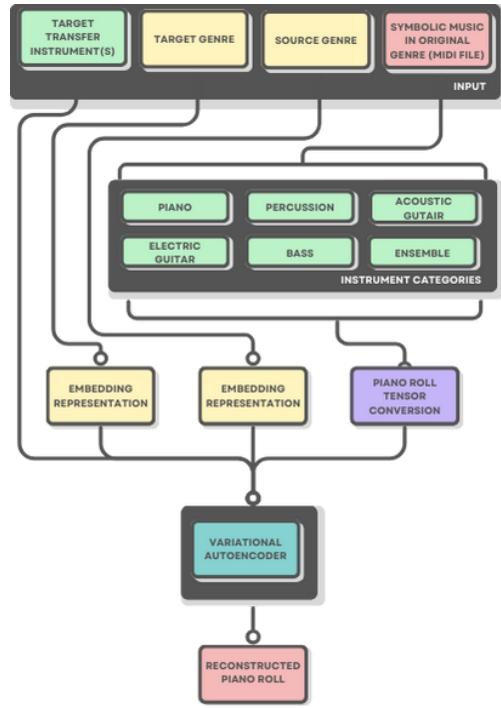


Fig. 2. Overview of symbolic music transfer model.

prior work in feature disentanglement [8], we use an adversarial classifier to ensure that the two learned distributions contain distinct information for genre and style, as shown in Figure 3. In this way, our proposed framework ensures that z_g contains the information relevant to the genre, z_s contains the other information useful for reconstructing the input piano roll, and that z_g and z_s encode different features.

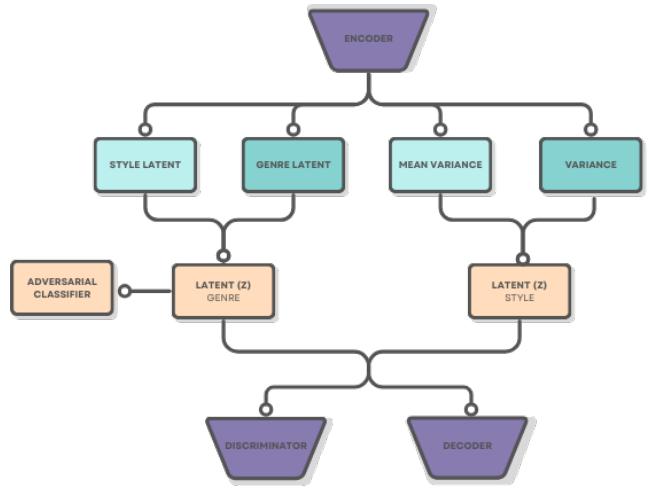


Fig. 3. Architecture of modified VAE.

1) Genre Classifier: We propose the addition of a supervised auxiliary task to encode the genre latent vector z_g with genre-relevant features. This is done by introducing the genre classification task on the latent vector z_g , where the goal is to correctly classify the vector as belonging to the genre as given

in the MidiCaps [5] dataset. The network is a 3-layer MLP with ReLU activation in the intermediate layers and softmax as the output activation. Dropout is used with a probability of 0.3. The classifier is trained jointly with the VAE using the cross-entropy loss

$$\mathcal{L}_{\text{genre}} = - \sum_{k=1}^K y_k \log \hat{y}_k, \quad (2)$$

where \hat{y}_k is the predicted probability and y_k is the one-hot encoded label for the k -th genre.

2) *Adversarial Classifier*: Separating genre and style distributions retains the style of the original track and changes the genre of the generated song. The encoded genre and style vectors contain distinct information to prevent the generated track from having an incomplete genre transfer or a different style from the original track.

Using the approach in [8], we introduce an adversarial component to our framework to minimize the mutual information between the genre and style vectors. A discriminator network D is introduced to distinguish whether a given pair of genre and style vectors (z_g, z_s) originate from the same track. The VAE encoder q generates these latent vectors as statistically independent from each other. In our implementation, latent vector pairs from different tracks (\tilde{z}_g, s) are obtained by shuffling the genre vectors z_g from the same batch. The discriminator loss \mathcal{L}_D and adversarial encoder loss \mathcal{L}_{adv} are phrased as follows where $[\cdot, \cdot]$ denotes concatenation:

$$\begin{aligned} \mathcal{L}_D &= -\mathbb{E}_{z_g, z_s \sim q(z_g, z_s | x)} (\log D([\tilde{z}_g, z_s]) \\ &\quad + \log(1 - D([z_g, z_s])), \\ \mathcal{L}_{\text{adv}} &= -\mathbb{E}_{z_g, z_s \sim q(z_g, z_s | x)} \log D([z_g, z_s]). \end{aligned} \quad (4)$$

In our implementation, the discriminator is implemented similarly to the genre classifier as a 3-layer MLP with ReLU activation, dropout with probability 0.3, and sigmoid as the output activation.

The total loss for the VAE is

$$\mathcal{L} = \mathcal{L}_{\text{VAE}} + \lambda_1 \mathcal{L}_{\text{genre}} + \lambda_2 \mathcal{L}_{\text{adv}}$$

for loss scaling hyperparameters λ_1, λ_2 , while the discriminator is trained separately.

III. RESULTS

A. Evaluation

We developed two evaluation metrics: N-gram analysis and Fréchet Music Distance-inspired analysis. The N-gram analysis evaluates the relationship between the original and transferred pieces, while the FMD-based metric assesses the effectiveness of capturing genre-specific musical patterns. The accuracy of these metrics was validated by testing them on state-of-the-art symbolic music transfer models.

An N-gram is a sequence of elements that preserves patterns in text or music. In symbolic music, N-grams capture

sequential features such as pitch, rhythm, and harmonic progressions [9]. This metric involves extraction and analysis of key musical features, including pitch sequences, note density, velocity profiles, pitch class distribution, chord patterns, and inter-onset interval distribution. Each feature is treated as an independent sequence, generating N-grams that are compared to a reference dataset of genre-specific samples. The similarity between the transferred piece and the reference dataset is quantified using N-gram overlap, measured through cosine similarity. This approach captures both local and global structure, providing a robust evaluation of genre transfer accuracy.

The second metric is inspired by the Fréchet Inception Distance, a measure designed to evaluate the quality of images generated by models like GANs [10]. FID quantifies the similarity between the distributions of generated and real data. Drawing from this concept, we adapt the FMD to evaluate symbolic music. This metric compares the distributions of key musical features, specifically temporal note density and pitch velocity extracted from piano roll representations. Each feature is transformed into a Gaussian distribution to capture local and global musical characteristics. The temporal distribution representing the density of musical activity over time can be seen in Figure 4.

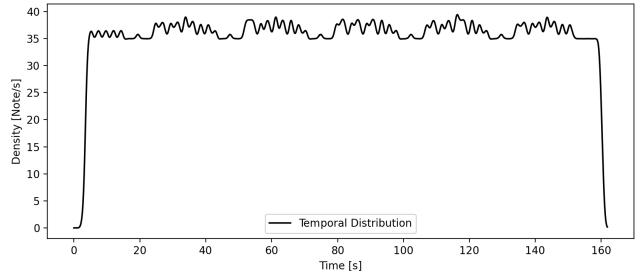


Fig. 4. Temporal distribution of *I walk the line* by Johnny Cash.

Figure 5 shows the pitch distribution representing the density of notes per pitch over the whole song.

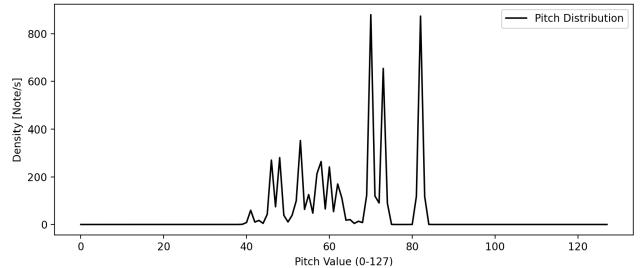


Fig. 5. Pitch distribution of *I walk the line* by Johnny Cash.

The transferred composition is evaluated by comparing its feature distributions to those of one hundred reference pieces from each target genre. A cosine similarity analysis measures the statistical alignment between the transferred piece and the genre-specific distributions. The similarity scores are summed across all features, and the genre with the smallest variation from the transferred piece is identified as the most suitable.

This metric accounts for potential overlaps between genres, providing a nuanced assessment of genre transfer quality. Together, these metrics provide a comprehensive framework for evaluating symbolic music genre transfer.

The results of the FMD analysis are presented in the figures below. For each genre, ten representative songs were evaluated using the FMD metric to determine which genre the model identifies as most similar, based on distributional similarity in musical features.

Figure 6 illustrates the FMD model's predictions regarding genre resemblance for ten pop songs.

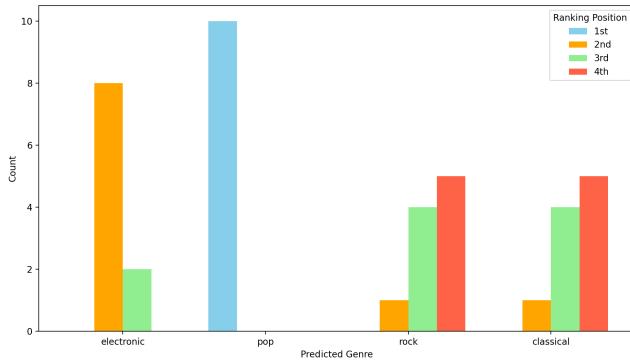


Fig. 6. FMD prediction distribution for the 10 true-genre pop songs.

The plot indicates that the model consistently identified pop as the most similar genre for the majority of pop songs, suggesting strong discriminative capability in this category.

Figure 7 presents the FMD results for ten rock songs.

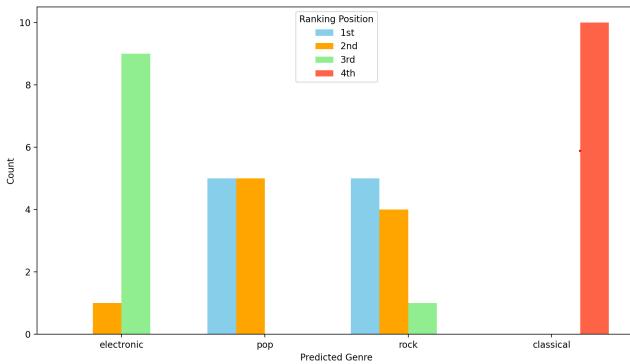


Fig. 7. FMD prediction distribution for the 10 true-genre rock songs.

Here, the metric frequently ranks rock as either the first or second closest genre, indicating a reasonably accurate classification, albeit with occasional confusion with other genres.

Figure 8 shows the corresponding results for electronic music.

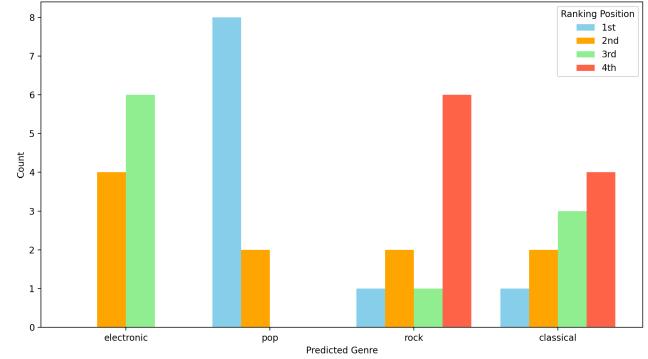


Fig. 8. FMD prediction distribution for the 10 true-genre electronic songs.

While electronic songs are often misclassified as pop, they remain the most frequently predicted genre among the remaining three, implying a moderate ability to capture their distinctive features.

Figure 9 displays the FMD analysis for classical pieces.

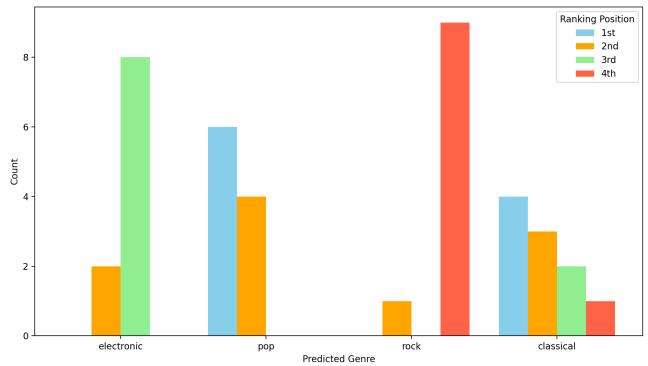


Fig. 9. FMD prediction distribution for the 10 true-genre classical songs.

Similar to the electronic results, pop is often predicted as the closest match. However, classical ranks highest among the remaining genres, suggesting that the model partially captures classical music's unique structure and tonal characteristics.

These results highlight the model's strength in distinguishing pop music from other genres, while revealing ambiguities in classifying electronic and classical music, suggesting the need for further refinement of the FMD metric or the integration of additional musical features to enhance genre discrimination. The inherent diversity and broad stylistic range within the pop genre may contribute to a systematic bias, whereby the metric is more likely to classify ambiguous inputs as resembling pop, thus influencing the overall genre prediction distribution.

IV. CONCLUSION

This project introduces a novel approach to symbolic music genre transfer by integrating a CNN-based VAE with an adversarial classifier and discriminator to ensure genre-invariant instrument-specific feature learning. By incorporating instrument-specific characteristics, the model enhances genre adaptation while preserving musical structure. Currently, our

model transfers songs with limited original song recognizability. The transferred genre misses certain key target genre characteristics. To improve this, we aim to refine the dataset for more clean pieces, altering the model's focus on specific musical elements and isolating the difficulties of specific model components. Further model iterations will be evaluated with the prepared N-gram similarity analysis to show the relation between the original and transferred pieces and the FMD metric to confirm its effectiveness in capturing melodic, rhythmic, and harmonic patterns specific to the desired genre. Future additions will focus on tokenizing the dataset, refining transfer accuracy, and expanding genre diversity. This work contributes to AI-driven music transformation domain knowledge and broadens creative possibilities in music composition, understanding, and production.

A. Future Work and Applications

Ongoing improvements aim to enhance the accuracy and quality of the transferred compositions, ensuring greater resemblance to the target genre. Incorporating Revamped MIDI (REMI) tokenization to represent the dataset would encourage the model to learn additional musical features by introducing special tokens [11]. Currently, the model supports four genres and five instrument categories. Future iterations will expand genre and instrument diversity, allowing for a broader range of stylistic transformations. User-based perceptual studies can be incorporated for evaluation alongside N-gram analysis and FMD metrics.

The potential applications of symbolic music genre transfer span multiple domains, from music composition and production to education and AI research.

REFERENCES

- [1] H. de Oliveira and R. Oliveira, “Understanding midi: A painless tutorial on midi format,” 2017.
- [2] Z. Ding, X. Liu, G. Zhong, and D. Wang, “Steelygan: Semantic unsupervised symbolic music genre transfer,” in *Pattern Recognition and Computer Vision*, S. Yu, Z. Zhang, P. C. Yuen, J. Han, T. Tan, Y. Guo, J. Lai, and J. Zhang, Eds. Springer International Publishing, 2022, pp. 305–317.
- [3] S.-L. Wu and Y.-H. Yang, “Musemorphose: Full-song and fine-grained piano music style transfer with one transformer vae,” 2022. [Online]. Available: <https://arxiv.org/abs/2105.04090>
- [4] G. Brunner, A. Konrad, Y. Wang, and R. Wattenhofer, “Midi-vae: Modeling dynamics and instrumentation of music with applications to style transfer,” 2018. [Online]. Available: <https://arxiv.org/abs/1809.07600>
- [5] J. Melechovsky, A. Roy, and D. Herremans, “Midicaps: A large-scale midi dataset with text captions,” 2024. [Online]. Available: <https://arxiv.org/abs/2406.02255>
- [6] W.-Y. H. Hao-Wen Dong and Y.-H. Yang, “Pypianoroll: Open source python package for handling multitrack pianorolls,” in *Late-Breaking Demos of the 19th International Society for Music Information Retrieval Conference (ISMIR)*, 2018.
- [7] D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” 2022. [Online]. Available: <https://arxiv.org/abs/1312.6114>
- [8] S.-W. Li, Z.-X. Wei, W.-J. Chen, Y.-H. Yu, C.-Y. Yang, and J. Y. jen Hsu, “Sa-dvae: Improving zero-shot skeleton-based action recognition by disentangled variational autoencoders,” 2024. [Online]. Available: <https://arxiv.org/abs/2407.13460>
- [9] J. Wołkowicz and V. Kešelj, “Analysis of important factors for measuring similarity of symbolic music using n-gram-based, bag-of-words approach,” in *Advances in Artificial Intelligence*, L. Kosseim and D. Inkpen, Eds. Springer Berlin Heidelberg, 2012, pp. 230–241.
- [10] J. Retkowski, J. Stepniak, and M. Modrzejewski, “Frechet music distance: A metric for generative symbolic music evaluation,” 2025. [Online]. Available: <https://arxiv.org/abs/2412.07948>
- [11] Y.-S. Huang and Y.-H. Yang, “Pop music transformer: Beat-based modeling and generation of expressive pop piano compositions.” *Proceedings of the 28th ACM International Conference on Multimedia*, pp. 1180–1188, 2020.

Toxicity Prediction Based on Molecular Structure Using Machine Learning

Tristan William Tucker

University of Victoria

tristantw2005@gmail.com

Tejal Simran Cheema

University of Victoria

tejalcheema@gmail.com

Abstract—Our project focuses on attempting to develop a machine learning model to predict the toxicity of molecules based on their molecular structure. In our testing we used two model archetypes, a Support Vector Machine (SVM) and a Neural Network, both trained using the Tox21 dataset [1], a catalog of over 10,000 molecules and their relative toxicities based on twelve distinct biological factors. Through our experiments, we found our best results were with using a Neural Network with CHEMBERT featurization of SMILES strings, LDA dimensionality reduction and SMOTEENN resampling. Our results show a positive correlation between molecular structure and toxicity, but found faults in attempting to build one general model to predict all the biological factors at once. To demonstrate an application of our models, we built an app allowing users to input the name of any molecule, and have the model output its predicted toxicity. [Link to GitHub here.](#)

I. INTRODUCTION

Toxicology is an important field in medical science and often a large barrier for the synthesis of new materials and pharmaceuticals. It includes the study of pollutants, infectants, mutagens; disciplines of Oncology, Exposomics, and Toxicogenomics, as well as whole range of other fields regarding dangerous substances to humans. Our project aims to explore the complexities of this field by trying to build an AI model that can predict the toxicity of molecules through predicting specific events they would trigger in the body.

A. Motivation

The inspiration for starting into this project was a paper published on developing molecular structures using generative AI [2]. The AI model was trained on a series of Simplified Molecular Input Line Entry System (SMILES) strings, an efficient means of representing the structural geometry and atomic connectivity of molecules within a single string. [3] Their model output was new SMILES strings representing newly generated molecules. The main idea as to why one may want to do this is for material or pharmaceutical synthesis. The problem with this is that it is expensive to put together the resources necessary to synthesize all these different structures and test their viability or safety. If there were a way to predict the toxicity of certain molecules, we could eliminate a vast majority of harmful toxins and focus solely on the viable subset. Furthermore, a study from 2022 [4] by the American Society of Biochemistry and Molecular Biology claimed that 90% of drugs fail clinical trials; where reportedly

”around 30% were due to unmanageable toxicity or side effects.”

These failures cost pharmaceutical large sums of both time and money through testing, and research. With a model that could accurately predict these unwanted toxic effects, we can mitigate the amount of losses and allow for more efficient drug discovery.

Today, toxicology is expanding rapidly thanks to the new tools being developed in AI. Although the idea of using Machine Learning to predict the toxicity of molecules is not new. Certain implementations of AI in toxicity testing, such as In Silico Toxicology [5], has been gaining widespread adoption since the early 20th century. The fault with current methods however is largely due to the amount of data needed to train AI models to then predict toxicity, including information from quantum mechanics simulations, QSAR modeling, and knowledge of certain physicochemical properties of the molecules. In many situations, we do not have the ability to gather or simulate all of the required data of these molecules, especially considering the cases where generative AI is used to generate millions of hypothetical molecular structures to discover new drugs. Building a model that could predict the toxicity of a molecule using only its molecular structure would solve this problem.

B. Related Works

The dataset used in this project is the Tox21 dataset (Toxicology in the 21st century), which catalogues of over 10,000 SMILES strings and their relative toxicity based on 12 tasks [1]. Originally introduced as a challenge dataset by the National Institute of Health (NIH) in 2014, Tox21 has since been widely used in machine learning-based toxicology research. One of the most notable models developed using this dataset is DeepTox, a deep learning model that won the Tox21 challenge [6].

The creators of DeepTox highlighted the potential of deep learning in toxicity predictions, emphasizing that neural networks are particularly well-suited for this task due to their ability to construct abstract chemical features [6]. This insight influenced our approach, motivating the use of neural networks as one of the models trained on the Tox21 dataset. Given the demonstrated success of deep learning within the Tox21 challenge, it gives validity to its usage in toxicology prediction.

Beyond deep learning approaches, other studies have explored alternative methods for improving toxicity classification on Tox21. One such study focused on the Structure-Activity Relationship (SAR) classification problem, a challenge stemming from the inherent class imbalance in toxicity datasets [7]. It used the Tox21 dataset and explored various resampling techniques to mitigate data imbalance. However, while our work compares Support Vector Machines (SVMs) and neural networks, their approach employed Random Forest as a base classifier, applying different resampling methods to improve model performance [7].

C. Defining the Problem of Toxicity

Dr. Stanley E. Manahan, a professor at the University of Missouri, wrote a book called Toxicological Chemistry, detailing the many ways in which toxicity has been defined over the years. This describes how toxicity had been commonly defined in terms of lethality or dosage; but why modern techniques aim to go further with the classification of toxic compounds by classifying them based "according to the parts of the body affected or by toxic effect." [8]. The following table below depicts the 12 distinct biological tasks included in the Tox21 dataset (i.e. what our model predicts):

TABLE I
TASKS OF THE TOX21 DATASET.

Task Key	Description
NR-AR	Androgen Receptor: Protein that binds to androgens.
NR-AR-LBD	Androgen Receptor LBD: Binding of androgenic compounds.
NR-AhR	Aryl Hydrocarbon Receptor: Receptor in cell cytoplasm that detects Aryl Hydrocarbons.
NR-Aromatase	Aromatase: Enzyme that converts androgens to estrogens.
NR-ER	Estrogen Receptor: Protein that binds to estrogen.
NR-ER-LBD	Estrogen Receptor LBD: Binding of estrogenic compounds.
NR-PPAR-gamma	Peroxisome Protein that controls the regulation of fat storage and glucose.
SR-ARE	Antioxidant Response Element: Defense against oxidative stress.
SR-ATAD5	ATAD5 gene: Involved in DNA damage response - could help predict carcinogens and mutations.
SR-HSE	Heat Shock Element: Triggers stress-response proteins under heat or toxic stress
SR-MMP	Mitochondrial Membrane Potential: Used as an indicator for mitochondrial dysfunction.
SR-p53	p53 Protein: Tumor suppressor involved in DNA repair and cell growth regulation.

A majority of the items listed are either enzymes, proteins or receptors that serve the body in some function. Toxicity in this context is the malicious disruption of these specific systems. Toxicity is a hard problem to define as many chemicals can be harmful in different ways, and in fact, most medical drugs are listed as toxins for the reason that they may harm certain parts of the body despite being employed for good reasons [9]. Toxicity is thereby not a binary value, nor is it a continuous spectrum, so alternative systems of describing the toxicity of molecules are needed. For example, the U.S

National Toxicology Program characterized toxicology into seven components [10]:

- 1) Cellular toxicology: Detimental alteration of cells
- 2) Genetic toxicology: Alterations of DNA by toxicants
- 3) Carcinogenesis: Potential to cause cancer
- 4) Reproductive and developmental toxicology: Effects on reproductive organs and embryos
- 5) Renal toxicology: Effects on the kidneys
- 6) Pulmonary toxicology: Effects on the lungs
- 7) Immunotoxicology: Effects on the immune system

Many of the biological assays that are present in the Tox21 dataset are represented in one or more of these categories. This is not to say that this is a perfect system, as there will be countless examples of chemicals and toxins that are extremely dangerous to humans that do not trigger any warnings and some common helpful endobiotic compounds that will. This is still one of the better ways of tracing toxicity, and the current focus for this paper.

II. METHODOLOGY

Predicting toxicity using the Tox21 dataset is a binary classification problem. While many binary classification tasks can be effectively addressed using traditional machine-learning models, the complexity of chemical structures presents additional challenges. Simpler models, such as Support Vector Machines (SVMs), may struggle to capture the intricate relationships between molecular features. Therefore, this project focuses on comparing SVMs and neural networks to determine which model type achieves superior performance.

The pipeline of our training process is illustrated in the flowchart below:

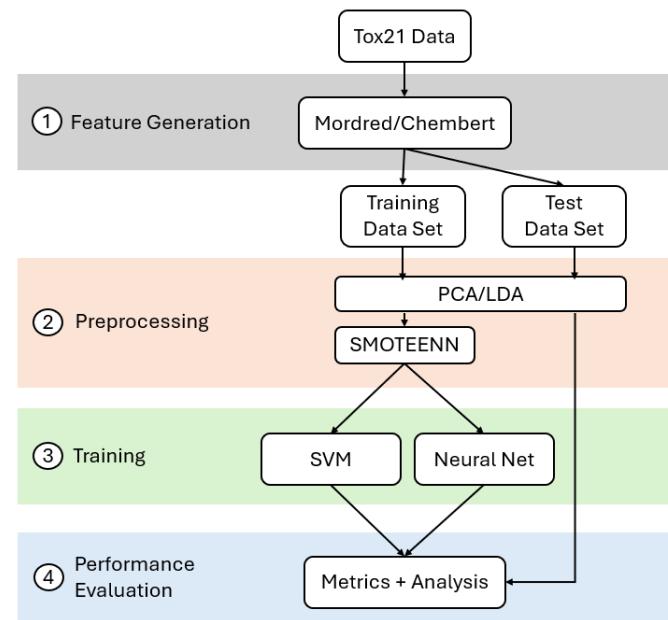


Fig. 1. Flowchart of model training.

The flowchart seen in figure I is split into four main sections. The basic process is explained below:

- 1) From the Tox21 dataset, we take the SMILES strings and featurize them into a vector embedding using either Mordred or CHEMBERT (*see Section on Dataset and Featurization*).
- 2) With the addition of vectorized molecules, we split the dataset into training and testing sets. For both sets we reduce the dimensionality using algorithms such as Principle Component Analysis (PCA) and Linear Discriminant Analysis (LDA). For the training set we use a technique called Synthetic Minority Oversampling Technique with Edited Nearest Neighbour (SMOTEEN) to increase visibility of the minority class.
- 3) After the preprocessing, we train our Support Vector Machines (SVMs) and neural network models.
- 4) Once models are trained, the test set is used to evaluate model performance based on the following key metrics: AUC, Precision, Recall and F1 Score.

A. Dataset and Featurization

As mentioned in the *Introduction*, the Tox21 dataset is a public dataset containing SMILES (Simplified Molecular Input Line Entry System) strings and 12 biological assay columns referred to as "tasks". A value of 1 in any task column indicates that the molecule triggered the assay, implying toxicity for that specific task.

There are two primary challenges with the Tox21 dataset. First, the dataset lacks predefined molecular features, as it only provides SMILES strings without additional structural or physicochemical descriptors. Second, the dataset exhibits class imbalance, favouring the value of 0 which indicates that most molecules do not trigger toxicity-related tasks.

To address the first challenge, we applied featurization, which is the process of transforming SMILES string into linear vectors or other favourable computer readable formats. We experimented with two main processes: Mordred [11] and CHEMBERT [12].

Mordred works by compiling an array of over 1000 integer values, each representing various structural features of the molecule. These features include, but are not limited to, the number of carbons, hydrogens, acid groups, basic groups, halogens, aromatic rings, and so forth. A complete list of the descriptors used can be found [here](#).

CHEMBERT is an open source Bidirectional Encoder Representation Transformer (BERT) model designed to take SMILES strings and find the hidden relationships of the molecular structure. CHEMBERT provided far fewer features as Mordred (380 opposed to 1084) while also significantly improving model performance and training efficiency. Therefore, we decided to use CHEMBERT for all final models.

B. Preprocessing

To prepare the dataset for model training, we applied two key techniques: Dimensionality Reduction and Minority Class Resampling. Both techniques combined constructed a dataset with a suitable amount of features and enough instances of the minority class to train a decent model. The following sections detail these techniques individually.

1) Dimensionality Reduction:

Dimensionality reduction is a technique used to reduce the amount of features in large datasets while maintaining its most important information. Given that both Mordred and CHEMBERT produce a large quantity of features, not all of which are completely relevant for the different tasks, dimensionality reduction became an essential tool when training the models.

We experimented with two methods of dimensionality reduction algorithms: Principal Component Analysis (PCA) and Linear Dimensionality Reduction (LDA). These models were trained separately to explore their differences in results, and were compared to a baseline where no dimensionality reduction was applied.

PCA is a technique that identifies the most relevant principal components in the feature space, reducing dimensionality by discarding components that contribute the least variance. However, PCA is less interpretable, as it does not provide insight into which features are deemed more or less important. Given this, we decided to pivot towards using LDA, which is a supervised method that aims to find the projection that maximizes the separation (or discriminability) between different classes. Given its suitability for binary classification problems, LDA outperformed PCA and ultimately produced the best results.

2) Resampling:

Resampling is a method that attempts to fix imbalanced datasets by means of altering the exposure of certain classes within the dataset. There are two ways to go about this. One technique is oversampling which creates synthetic samples to increase the number of the minority class. The other is undersampling, which reduces the number of samples of the majority class.

Our models utilize a technique called SMOTEENN, a combination of the oversampling method SMOTE (Synthetic Minority Oversampling Technique) and the undersampling method ENN (Edited Nearest Neighbor). SMOTE generates synthetic samples by interpolating between nearby positive minority class instances. ENN removes samples that are misclassified by their k-nearest neighbors, primarily from the majority class, to improve class separability and reduce noise. This combination enhances model performance by balancing the dataset while filtering out ambiguous mislabeled samples.

Seeing the results in *Section III. Results*, models using SMOTEENN produced better results than without them.

C. Training

Following is a discussion of our training process, including insight into both our Neural Networks and Support Vector Machines.

1) Neural Networks:

The neural network models used in this project follows a simple design implemented using TensorFlow. The architecture consists of the following layers:

- 1) Input Layer: Receives the featurized molecular data derived from either Mordred or CHEMBERT representations of the molecules.
- 2) Normalization Layer: Applied to standardize input data, ensuring all features are scaled to a similar range.
- 3) First Hidden Layer: Contains 64 neurons and uses the ReLU (Rectified Linear Unit) activation function to introduce non-linearity and allow the model to learn more complex patterns
- 4) Second Hidden Layer: Contains 32 neurons, continuing to refine learned features and also uses ReLU activation function.
- 5) Output Layer: A single neuron with a sigmoid activation function to predict the probability of toxicity for each of the 12 tasks. The sigmoid function is used because the tasks are binary classification problems - a value of 1 represents toxicity, while 0 indicates non-toxicity.

To handle class imbalance inherent in the Tox21 dataset, we use Focal Loss as the loss function. Focal Loss mitigates the dominance of the majority class by down-weighting easy-to-classify samples, allowing the model to focus more on harder-to-classify instances. This makes it particularly effective for imbalanced datasets. For further details, refer to the Focal Loss paper [13].

Additionally, class weights are applied to further address the class imbalance by providing a higher weight to the minority class which reinforces the importance of making accurate predictions for the underrepresented class.

The class weights used in the training of both the SVM and Neural Networks were given by the following equation (Note: Each task had its own calculated set of class weights):

$$w_i = \frac{N}{2 \times n_i} \quad (1)$$

where:

- i is the class, either 1 or 0
- w_i is the class weight for class i ,
- N is the total number of samples in the dataset,
- n_i is the number of samples belonging to class i .

2) SVMs:

The Support Vector Machines were implemented using sklearn's svm module which includes a variety of different kernel states and parameter options. From testing we found that the best results were gained by using the base SVM which employs a Radial Basis Function (RBF) kernel; good for complex, nonlinear data, which describes our input space for both Mordred and CHEMBERT.

D. Streamlit Integration

As an additional add-on to our project, a streamlit app was created to showcase a potential demo for a scenario where our models would be used by pharmaceutical companies. The models of all 12 tasks are saved and loaded into the app, where the user can select the task for a prediction.

The integration of the app is as follows:

- 1) User inputs the name of a molecule which exists in the PubChem database, a free chemical database containing over 100 million known compounds. [14]
- 2) Through the pubchem database we retrieve its representative SMILES string and parse it through CHEMBERT's API to generate features.
- 3) The features then undergo dimensionality reduction before being put through the models to generate predictions on toxic effect.
- 4) The predictions then get outputted for the users to view.

For further information regarding the app, refer to our Github page.

III. RESULTS

In this section, we will discuss and compare our results using neural networks and SVMs for our binary classification problem.

For both neural networks and SVMs, different featurization, dimensionality reduction and resampling techniques were explored (*Refer to Section II. Methodology*). Ultimately, through experimentation, our best results for both model types had the following combination: CHEMBERT, LDA, and SMOTEENN.

The table below presents the averages scores of both model types across all 12 tasks. Although SVMs show a greater trend in Recall, the neural networks outperform in F1 score, Precision, and AUC score. And since the F1 score represents the harmonic mean between Precision and Recall, it balances both metrics, making it particularly important in this context where minimizing false negatives is critical. Given this, the neural network models are deemed better for toxicity prediction.

TABLE II
AVERAGE METRICS FOR MODELS OF EACH TASK

Model	AUC	Precision	Recall	F1
Neural Networks	87.01%	48.45%	48.88%	46.92%
SVMs	83.98%	24.93%	68.92%	35.24%

Figure 2 (below) showcases the precision-recall graph for all neural networks models of each task. It can be seen that each model varies in their results, where many seem to under perform significantly. The reason is due to the fact that we were trying to build a general neural network model to be used for each task. That is, the hyper-parameters were the same among all tasks. Based off the results, this proved to be a much more ambitious and challenging approach to the problem as each task exhibits different properties. Therefore, any future work should create neural network models unique to the task to produce optimal results.

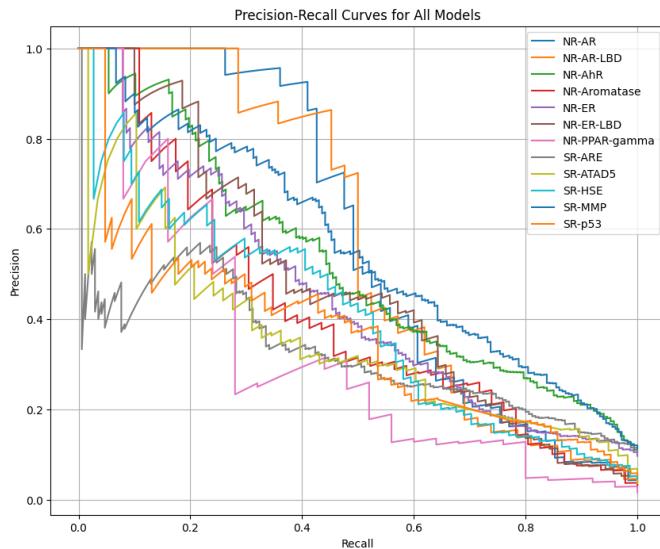


Fig. 2. Precision-Recall Graph for All Neural Net Models using LDA

The performance of our metrics found in *Table 1* for neural networks could be explained by the imbalanced of the dataset, combined with the fact that a general neural network model was utilized.

IV. CONCLUSION

This study compared the difference between SVMs and neural networks in building a general machine learning model that could predict the toxicity of molecules as defined in the Tox21 dataset. Our best results were gained through using a combination of CHEMBERT featurization, LDA dimensionality reduction and SMOTEEEN resampling for both the SVMs and Neural Networks.

Given the metrics we used, we claimed that the neural networks outperformed the SVMs for this given problem, but there is still great difference pertaining to the areas such as recall and precision. One reason for this could be our persistence in applying a general neural network architecture across all 12 tasks. As shown in *Figure 2*, this generalized approach proved ineffective, as each task was so unique. Future developments would disband this methodology and focus on building task specific models for each task. Therefore, until this additional work is completed, the conclusion of whether SVMs or neural networks are better suited for toxicity prediction across different tasks cannot be made.

Our models appeared to uphold to the long recognized trend of chemical Structure-Activity Relationship (SAR) [15]: which argues that the structure of a chemical can be predicted by its physical and chemical characteristics. We can make this claim by addressing the fact that by only using only a vectorized representation of a molecule's structure, we were able to receive an average AUC score of 85% between the two models. This implies that molecular structure alone carries enough information to make meaningful toxicity predictions.

REFERENCES

- [1] A. M. Richard, R. Huang, S. Waidyanatha, P. Shinn, B. J. Collins, I. Thillainadarajah, C. M. Grulke, A. J. Williams, R. R. Lougee, R. S. Judson *et al.*, “The tox21 10k compound library: collaborative chemistry advancing toxicology,” *Chemical Research in Toxicology*, vol. 34, no. 2, pp. 189–216, 2020.
- [2] R. Gómez-Bombarelli, J. N. Wei, D. Duvenaud, J. M. Hernández-Lobato, B. Sánchez-Lengeling, D. Sheberla, J. Aguilera-Iparraguirre, T. D. Hirzel, R. P. Adams, and A. Aspuru-Guzik, “Automatic chemical design using a data-driven continuous representation of molecules,” *ACS central science*, vol. 4, no. 2, pp. 268–276, 2018.
- [3] E. J. Bjerrum, “Smiles enumeration as data augmentation for neural network modeling of molecules,” *arXiv preprint arXiv:1703.07076*, 2017.
- [4] D. Sun, “90% of drugs fail clinical trials,” *ASBMB Today*, March 2022. [Online]. Available: <https://www.asbmb.org/asbmb-today/opinions/031222/90-of-drugs-fail-clinical-trials>
- [5] A. B. Raies and V. B. Bajic, “In silico toxicology: computational methods for the prediction of chemical toxicity,” *Wiley Interdisciplinary Reviews: Computational Molecular Science*, vol. 6, no. 2, pp. 147–172, 2016.
- [6] A. Mayr, G. Klambauer, T. Unterthiner, and S. Hochreiter, “DeepTox: toxicity prediction using deep learning,” *Frontiers in Environmental Science*, vol. 3, p. 80, 2016.
- [7] G. Idakwo, S. Thangapandian, J. Luttrell, Y. Li, N. Wang, Z. Zhou, H. Hong, B. Yang, C. Zhang, and P. Gong, “Structure-activity relationship-based chemical classification of highly imbalanced tox21 datasets,” *J. Cheminform.*, vol. 12, no. 1, p. 66, Oct. 2020.
- [8] S. E. Manahan, *Toxicological Chemistry and Biochemistry*, 1st ed. CRC Press, 2002. [Online]. Available: <https://www.routledge.com/Toxicological-Chemistry-and-Biochemistry/Manahan/p/book/9781566706186>
- [9] S. Stevens, *Deadly Doses: A Writer’s Guide to Poisons*. Cincinnati, OH: The Writer’s Digest Books, 2004.
- [10] N. T. (P. US), *National Toxicology Program: Annual Plan for Fiscal Year 1986*. National Toxicology Program, Public Health Service, Department of Health and ..., 1987.
- [11] H. Moriwaki, Y.-S. Tian, N. Kawashita, and T. Takagi, “Mordred: a molecular descriptor calculator,” *Journal of cheminformatics*, vol. 10, no. 1, p. 4, 2018.
- [12] H. Kim, J. Lee, S. Ahn, and J. R. Lee, “A merged molecular representation learning for molecular properties prediction with a web-based service,” *Scientific Reports*, vol. 11, no. 1, p. 11028, 2021.
- [13] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, “Focal loss for dense object detection,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 318–327, Feb. 2020.
- [14] S. Kim, “Pubchem: A large-scale public chemical database for drug discovery sunghwan kim and evan e. bolton national center for biotechnology information, national library of medicine, national institutes of health, 8600 rockville pike, bethesda, md 20894, usa,” *Open Access Databases and Datasets for Drug Discovery*, p. 41, 2023.
- [15] B. T. Walton and T. Mill, “Structure-activity relationships in environmental toxicology and chemistry,” pp. 403–404, 1988.

TrafficLightRL

Kristian Diana

McMaster University
dianak@mcmaster.ca

Tridib Banik

McMaster University
banikt@mcmaster.ca

Varun Pathak

McMaster University
pathav4@mcmaster.ca

Ryan Li

McMaster University
li3018@mcmaster.ca

Clara Wong

McMaster University
wongc148@mcmaster.ca

Abstract—Addressing urban traffic congestion is crucial for environmental sustainability, as inefficient traffic flow leads to increased fuel consumption and greenhouse gas emissions. This paper presents TrafficLightRL, a reinforcement learning (RL)-based traffic light control system designed to minimize vehicle emissions and improve traffic efficiency. Using SUMO for simulation and the Proximal Policy Optimization (PPO) algorithm from Stable-Baselines3 for RL training, our system dynamically adapts to real-time traffic conditions. Results show that the RL agent reduces CO₂ emissions by up to 11.6% compared to traditional fixed-time systems, with performance evaluated across various traffic densities. This study highlights the potential of RL-driven solutions to enhance traffic management and reduce environmental impact. The code and resources for this project are available at: <https://github.com/McMasterAI2024-2025/TrafficLightRL>.

I. INTRODUCTION

As one of the McMaster AI Society’s many projects for the 2024/25 season, this team investigated a reinforcement learning solution that could address a pressing real-world issue. Out of a shared concern for environmental sustainability, as well as inefficiencies in urban mobility, existing traffic light systems were identified as a major contributor to excessive vehicular emissions. Current traffic light control mechanisms rely on fixed-timer schedules or sensor-based adjustments, leading to unnecessary idling and frequent acceleration events—both of which are known to significantly increase carbon dioxide (CO₂) emissions [1]. In response to this challenge, TrafficLightRL was developed as a reinforcement learning (RL)-based system aimed at reducing emissions by optimizing signal timings dynamically. This document discusses the issue addressed by the project, the design and implementation of the RL-based system, the quantifiable environmental benefits observed, and the broader implications for sustainable traffic management.

A. Motivation

Road transport is responsible for nearly 25% of global CO₂ emissions from fuel combustion, with urban congestion playing a significant role in this footprint [2]. One of the primary factors contributing to unnecessary emissions is stop-and-go traffic, where vehicles frequently accelerate from a standstill. Studies have shown that rapid acceleration events can increase fuel consumption and emissions by up to 200% compared to steady-speed travel [1]. Furthermore, in urban areas, drivers spend an average of 54 hours per year idling in traffic, further exacerbating emission levels [3].

Despite advancements in adaptive traffic control, conventional systems still struggle to minimize acceleration and

idling simultaneously. This project explores how reinforcement learning can address this gap by dynamically adjusting traffic signal timings based on real-time conditions. By prioritizing reductions in idling duration and acceleration frequency, an RL-based approach has the potential to significantly lower CO₂ emissions in urban environments.

B. Related Works

Traditional traffic light control systems, such as fixed-timer schedules and sensor-based systems, rely on predefined rules that lack adaptability to real-time traffic conditions. While adaptive systems like SCOOT [4] and SCATS [5] offer improvements by dynamically adjusting signal timings, they remain limited by rule-based optimizations that require manual calibration and do not generalize well to varying traffic patterns.

Reinforcement learning has emerged as a promising alternative, offering the ability to learn optimal signal timing policies directly from traffic data. Prior studies have demonstrated that RL-based systems can reduce vehicle stops, travel times, and overall emissions [6]. Unlike traditional adaptive systems, RL approaches continuously refine their control strategies based on observed traffic dynamics, making them well-suited for sustainable urban mobility.

C. Problem Definition

The objective of this project is to develop an RL-based traffic light control system that reduces CO₂ emissions by minimizing vehicle idling and acceleration events. The proposed system optimizes signal timings based on real-time traffic flow, adapting dynamically to different conditions without relying on predefined scheduling rules.

Key evaluation metrics include reductions in total CO₂ emissions, idling duration, and the number of full stops per vehicle. The results aim to highlight the potential of reinforcement learning in mitigating urban traffic’s environmental impact and providing a more sustainable solution to this pressing issue.

II. METHODOLOGY

This section outlines the design and implementation of the RL-based traffic light control system. It begins by presenting the tools and technologies used, followed by a detailed description of the RL agent’s structure and decision-making process. Next, the simulation environment is discussed alongside real-world considerations to ensure practical applicability. Finally,

the evaluation metrics and calculation methods are introduced, providing a foundation for performance assessment in the results section.

A. Tools and Technologies

The project leverages two primary technologies: **SUMO (Simulation of Urban MObility)** and **Stable-Baselines3**. SUMO provides a realistic traffic simulation environment, including real-time visualizations and dynamic traffic scenarios. Stable-Baselines3, a popular reinforcement learning library, is used to train the RL agent. The integration between SUMO and Stable-Baselines3 is facilitated by **TraCI**, an API that enables real-time communication between Python and SUMO. Additionally, **OpenAI Gymnasium** is used to create a consistent interface for the RL agent, abstracting SUMO's functionality into a format compatible with Stable-Baselines3. Figure 1 illustrates the process flow and interactions between these components, which will be elaborated on shortly. Finally, **Matplotlib** is leveraged for analysis of various evaluation metrics.

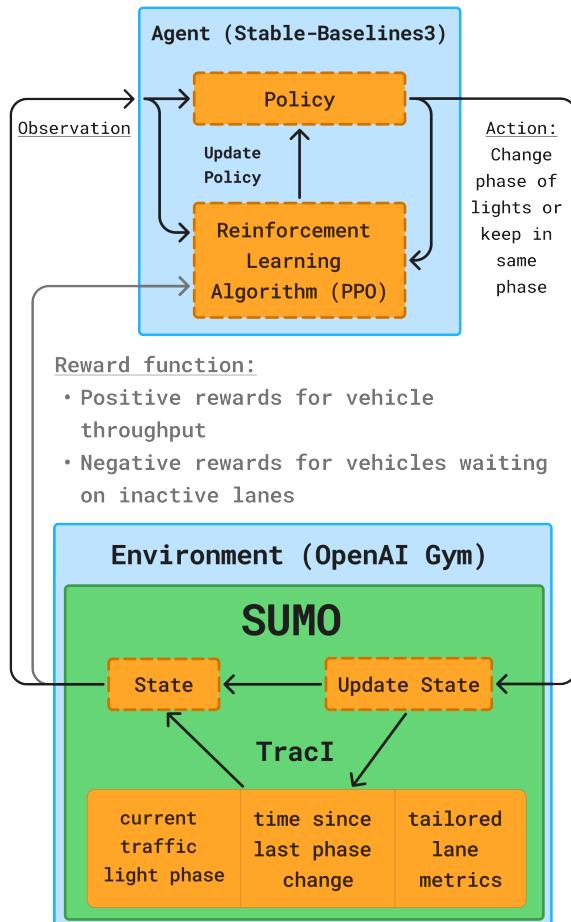


Fig. 1. Process flow diagram illustrating the interaction between SUMO, TraCI API, OpenAI Gymnasium, and Stable-Baselines3.

B. Reinforcement Learning Agent

The RL agent is trained using the **Proximal Policy Optimization (PPO)** algorithm, a state-of-the-art policy gradient method implemented in Stable-Baselines3. PPO was chosen for its stability and efficiency in handling continuous state and action spaces, making it well-suited for dynamic traffic control tasks. The agent interacts with the environment by observing the current state and selecting actions to optimize traffic flow. The key components of the agent are as follows:

- **Observation Space:** The agent observes the current traffic light phase, the time since the last phase change, and lane-specific metrics such as the number of queued vehicles and average idling time.
- **Action Space:** The agent has a discrete action space, where each action corresponds to a specific traffic light phase. Each phase determines which lanes are active (green) and which are inactive (red).
- **Reward Function:** The agent receives positive reinforcement for vehicles passing through the intersection and negative reinforcement for vehicles queued in inactive lanes. This encourages the agent to minimize waiting times and congestion.

C. Simulation Environment

The simulation environment is designed to mimic real-world traffic conditions, which is addressed through various features:

- **Random Vehicle Deployments:** Introducing randomness in the form of vehicle routes allows the agent to generalize effectively to unpredictable traffic patterns. This enhances real-world applicability, as the RL agent will never experience two identical **episodes** during training. To clarify, an **episode** represents the time period required for a specific number of vehicles to pass through the intersection.
- **Variable Traffic Densities:** Adjusting the spawn rate of vehicles enables simulations to model traffic fluctuations, accounting for real-world factors such as **time of day** and **weather conditions**.
- **Standard Traffic Safety Regulations:** The environment enforces standard traffic safety rules, such as requiring the agent to pass through **transition phases** (e.g., green → yellow → red) before switching traffic lights. Additionally, each phase must be held for a **minimum duration** to prevent chaotic scenarios and to ensure pedestrians have sufficient time to cross safely. These features ensure that the agent adheres to real-world constraints and common safety practices.
- **Real-World Networks:** To enhance realism, we use the **OSM Web Wizard** to export actual geographical location networks into SUMO. This facilitates traffic simulations in real-world environments, such as the road network around McMaster University. Figure 2 shows a side-by-side comparison of the Google Maps view of McMaster University and the corresponding SUMO simulation.

In this paper, the term “traditional system” refers to a fixed-time traffic signal control system implemented in SUMO.

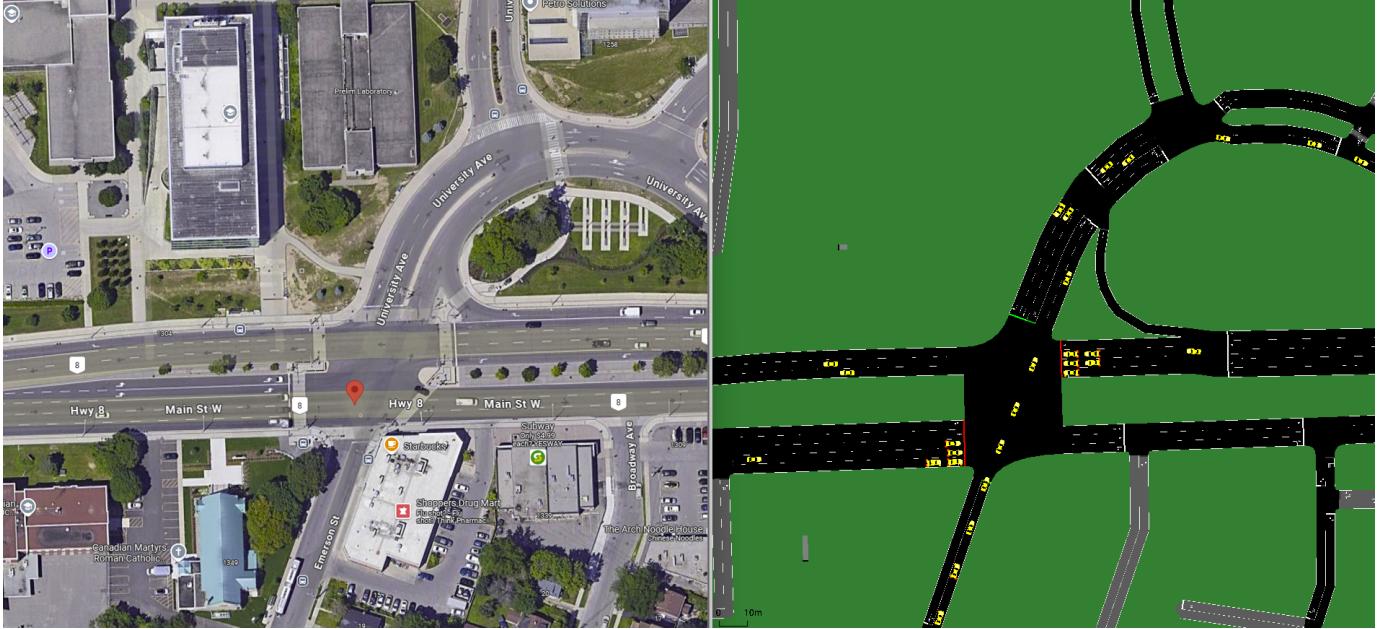


Fig. 2. Side-by-side comparison of the Google Maps view of McMaster University (left) and the corresponding SUMO simulation (right) generated using the OSM Web Wizard.

This system operates on a predetermined cycle, with traffic light phases changing at fixed intervals regardless of real-time traffic conditions. It serves as a baseline for evaluating the performance improvements achieved by the RL-based approach.

D. Evaluation Metrics

To evaluate the performance of the RL agent, three key metrics are evaluated and calculated as follows leveraging the TraCI API:

- **Mean CO₂ Emissions:** The average CO₂ emissions, measured in mg, produced by a single vehicle during the simulation. Emissions are calculated at each timestep of the simulation using SUMO's TraCI API, which provides real-time vehicle-specific emissions data. Specifically, the `traci.vehicle.getCO2Emissions` method returns the CO₂ emissions in milligrams (mg) for each vehicle, assuming a standard emission rate. The emissions are aggregated over the duration the vehicle is on the simulation, and then evaluated as an average of all vehicles that went through the intersection.
- **Mean Idling Times:** The average time a single vehicle spent waiting at the intersection. This value is calculated for each vehicle of the simulation using the `traci.vehicle.getWaitingTime` method. These values are averaged to provide a cohesive representation rather than focusing on individual vehicles.
- **Stopping Probability:** The likelihood of a vehicle stopping at the intersection, calculated as the ratio of total vehicles required to stop to total vehicles on the simulation. This value is calculated utilizing the mean idling times for

each vehicle from earlier, assuming any vehicle that has an idling time above a specific threshold has stopped. A threshold is applied to distinguish true stops from minor slowdowns or coasting. This value is represented as a probability, rather than a count, as variable factors such as traffic density and total number of cars will skew this data.

These metrics are calculated for each episode and compared against traditional traffic light systems to demonstrate the effectiveness of our RL-based approach. These metrics provide a comprehensive evaluation of both traffic efficiency and environmental impact, ensuring the results are applicable beyond a single intersection.

III. RESULTS

This section presents the performance of the RL-based traffic light control system compared to a traditional system. The traditional system, in the context of SUMO, operates on a fixed cycle of pre-determined green and red light intervals, independent of real-time traffic conditions. The system is evaluated based on the three evaluation metrics described previously: **mean emissions per second**, **mean idling time**, and **stopping probability**. The x-axis of all figures represents the vehicle spawn rate, corresponding to the probability of a vehicle being deployed at each time-step. The spawn rate is representative of low, medium, and high traffic densities, illustrated by the blue, green, and red regions respectively. Each data point is the average of 1000 episodes to ensure statistical reliability and smooth distributions. The spawn rate corresponds to the chance a vehicle is to be deployed at each

time-step of the simulation, and each episode deploys 100 vehicles.

A. Effect of Traffic Density on Mean CO₂ Emissions

Figure 3 illustrates the effect of traffic density on mean CO₂ emissions, measured in (mg).

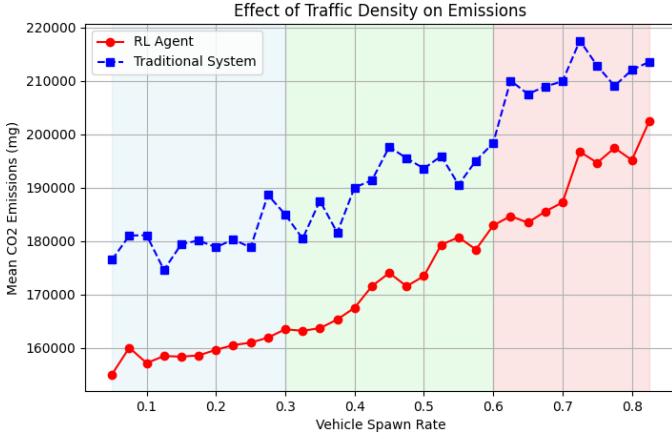


Fig. 3. Effect of traffic density on mean CO₂ emissions. The red line represents the RL agent, while the blue dashed line represents the traditional system.

The RL agent significantly reduces CO₂ emissions compared to the traditional system. In low traffic conditions, emissions are reduced by **11.6%**, **10.2%** in medium traffic conditions and **9.0%** in high traffic conditions.

B. Effect of Traffic Density on Mean Idling Times

Figure 4 presents the impact of traffic density on mean idling times, measured in seconds.

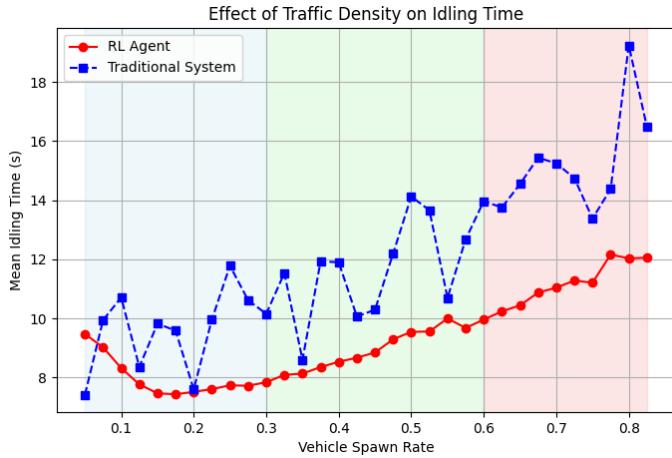


Fig. 4. Effect of traffic density on mean idling times. The red line represents the RL agent, while the blue dashed line represents the traditional system.

The RL agent consistently outperforms the traditional system by reducing idling times. In low traffic conditions idling

times decrease by **16.4%**, while in medium traffic conditions they decrease by **22.7%**. Even in high traffic conditions the RL agent maintains its advantage with a **26.4%** reduction in idling times.

C. Effect of Traffic Density on Stopping Probability

Figure 5 shows the effect of traffic density on stopping probability, defined as the proportion of vehicles required to stop at the intersection.

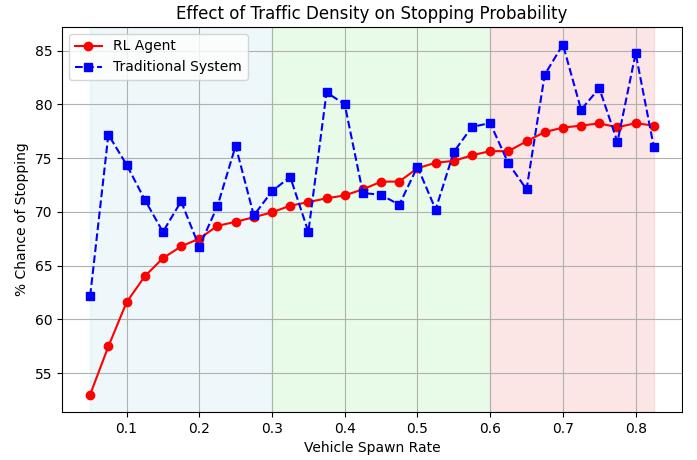


Fig. 5. Effect of traffic density on stopping probability. The red line represents the RL agent, while the blue dashed line represents the traditional system.

The RL agent also reduces stopping probability. In low traffic conditions, stopping probability decreases by **9.0%**, while in medium and high traffic conditions, it is reduced by **1.8%** and **2.3%**, respectively.

IV. CONCLUSION

In this project, an RL-based traffic light control system was developed using SUMO and Stable-Baselines3 to address the inefficiencies of traditional traffic management systems. Our findings highlight that the RL agent substantially reduces CO₂ emissions across varying traffic densities by minimizing idling time and decreasing the frequency of complete stops. These efficiency gains directly contribute to a more environmentally sustainable urban traffic management system.

A. Discussion of Emissions Reduction

The RL agent's ability to reduce emissions stems from two key factors: minimizing idle time, which mitigates prolonged low-efficiency fuel consumption, and reducing stops, which limits high-emission acceleration events after red lights.

- **Minimization of Idling Time:** Idling, while producing lower emissions per second compared to acceleration, contributes substantially to total emissions over time due to its prolonged nature. Although idling times saw the greatest reduction in high-traffic scenarios, emissions improvements were less pronounced, highlighting the diminishing returns of idling reductions compared to stop reductions

- **Reduction in Stops:** Acceleration from a complete stop is a significant contributor to vehicle emissions. By optimizing traffic flow and reducing the frequency of stops, the RL agent minimizes the instances of high-emission acceleration events. The RL agent significantly reduces stops, particularly in low-traffic conditions, where a **9.0%** decrease in stopping probability resulted in the greatest emissions reduction. This suggests that minimizing unnecessary stops is a key factor in emissions control.

These factors highlight the RL agent's ability to address both the high-emission events (acceleration from stops) and the sustained emissions (idling) that characterize inefficient traffic systems. By dynamically adapting to real-time traffic conditions, the RL agent provides a more environmentally sustainable solution compared to traditional fixed-timer based systems. The results suggest that further emphasis on decreasing high-emission events, such as acceleration from stops, is most significant in relation to reduction of CO₂ emissions.

B. Limitations

Although the RL agent was trained across various traffic densities, the training distribution was linearly incremented, causing the agent to encounter medium traffic densities most frequently. As a result, the agent is more adept at optimizing performance in these conditions but may be less effective in handling extreme congestion or sparse traffic. A more balanced training approach—incorporating additional timesteps for low- and high-traffic scenarios—could improve generalizability across a wider range of conditions.

While SUMO's emissions model provides a realistic approximation of vehicle emissions, it relies on generalized vehicle dynamics. It does not fully account for real-world factors such as variations in vehicle types, fuel efficiency, or environmental influences like weather conditions. However, because the model prioritizes acceleration and idling behavior—the primary contributors to emissions in urban traffic—it remains a useful tool for comparative analysis of traffic control strategies.

Additionally, our study is confined to a controlled simulation environment, which, while useful for experimentation, does not capture network-wide congestion effects or interactions with external infrastructure. This limitation may lead to discrepancies between simulated and real-world performance, as the learned policy may not generalize effectively to larger urban networks. Future iterations should incorporate broader network effects to assess scalability and adaptability in dynamic traffic ecosystems.

C. Future Considerations

Several enhancements could improve the RL agent's real-world applicability. Future implementations could incorporate additional considerations such as pedestrian right-of-way, emergency vehicle prioritization, and adaptive responses to weather conditions. These factors play a crucial role in urban traffic systems and would enhance the agent's ability to operate effectively in diverse environments.

Another major challenge lies in scaling this approach to coordinate multiple intersections. As the number of controlled intersections increases, the complexity of synchronizing signals grows exponentially, necessitating more advanced agent communication strategies and higher computational resources. Multi-agent reinforcement learning (MARL) techniques, such as decentralized policies with shared learning objectives, could be explored to tackle this scalability issue.

Additionally, the rise of electric vehicles (EVs) presents a new avenue for optimization. Since EVs have different acceleration profiles and do not produce emissions while idling, an RL-based system tailored for mixed traffic compositions—including both traditional internal combustion engine (ICE) vehicles and EVs—could further enhance sustainability outcomes. Incorporating real-time EV-specific traffic data would refine emissions predictions and improve energy efficiency across urban road networks.

While this study demonstrates promising results in optimizing traffic flow at a single intersection, testing the approach across a variety of urban environments is essential for broader applicability. Our current implementation focuses primarily on Ontario university campuses; expanding to diverse real-world locations with varied infrastructure and traffic patterns would provide further validation and refinement of the system.

REFERENCES

- [1] M. Barth and K. Boriboonsomsin, “Energy and emissions impacts of a freeway-based dynamic eco-driving system,” *Transportation Research Part D: Transport and Environment*, vol. 14, no. 6, pp. 400–410, 2009.
- [2] I. E. Agency, “CO₂ emissions from fuel combustion 2022,” <https://www.iea.org/reports/co2-emissions-in-2022>, 2022, [Online; accessed 2022].
- [3] INRIX, “Global traffic scorecard,” <https://inrix.com/scorecard/>, 2022, [Online; accessed 2022].
- [4] P. Hunt, D. Robertson, R. Bretherton, and R. Winton, “Scoot—a traffic responsive method of coordinating signals,” Transport and Road Research Laboratory, Technical Report, 1981.
- [5] P. Lowrie, “Scats: The sydney coordinated adaptive traffic system - principles, methodology, algorithms,” *Traffic Engineering & Control*, vol. 33, no. 6, pp. 278–281, 1992.
- [6] W. Wu, Y. Zhang, R. Jiang, H. Zhang, K. Gao, and D. Zhao, “A comprehensive survey on traffic signal control methods,” *Transportation Research Part C: Emerging Technologies*, vol. 123, p. 102974, 2020.

World Model Architectures for Model-Based Reinforcement Learning

Triston Grayston

University of Victoria

tristongrayston@uvic.ca

Ari Van Everdingen

University of Victoria

arivaneverdingen@uvic.ca

Abstract—World models offer several theoretical benefits, such as enhanced planning capabilities, and faster, safer, and cheaper sampling. However, training an effective world model is difficult. This work explores this challenge by testing 3 neural network architectures - neural networks with a residual connection, recurrent neural networks, and Neural Circuit Policies - in approximating the dynamics of 3 environments: the Lorenz system, Open AI gym’s Pendulum, and a modified, partially observed Pendulum.

I. INTRODUCTION

A. Motivation

Reinforcement learning typically divides into Model-Free and Model-Based Reinforcement Learning. In Model-Free, agents optimize their behavior to achieve a specific goal, implicitly learning the environment’s dynamics on the way. Model-Based Reinforcement Learning explicitly learns a dynamics model separate to the agent. Incorporating an accurate dynamics model provides substantial advantages, including enhanced planning capabilities, and faster, safer, and cheaper sampling compared to interactions with real-world or high-fidelity simulated environments. Real-world interactions present significant challenges due to slow sampling speeds, high costs, and safety risks associated with failures. Learned neural network-based world models offer a promising alternative to traditional simulations, as precisely capturing and replicating real-world dynamics through explicit modeling can be infeasible or prohibitively complex.

Despite these advantages, model-based approaches remain relatively underexplored, primarily due to difficulties in generating accurate and representative simulated data. Effective world models must simultaneously achieve correctness, sample efficiency, and computational simplicity—criteria that few existing architectures satisfy simultaneously. Those that do meet these conditions often struggle with limited expressiveness or excessive complexity.

B. Related Works

Model-based reinforcement learning approaches seek to explicitly capture environmental dynamics through a learned world model, which then facilitates planning and decision-making. Early successes in this field utilized ensembles of neural networks to enhance generalization and reduce model uncertainty, demonstrating improved sample efficiency compared to model-free counterparts. However, these ensemble

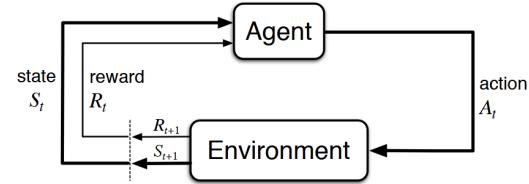


Fig. 1. The agent–environment interaction in a Markov decision process. [6]

methods introduce additional complexity and computational overhead, motivating research into more efficient yet expressive representations of dynamics [1]. Notably, recent model-based algorithms such as DreamerV3 have demonstrated state-of-the-art performance in benchmark tasks, yet reproducibility and transparency remain significant barriers, limiting their broader adoption in practical scenarios [2].

Advances in continuous-time modeling of system dynamics have inspired the development of Neural Ordinary Differential Equations (Neural ODEs) [3]. Neural ODEs parameterize the derivative of the state as a neural network, integrating forward through time to capture complex, continuous-time dynamical systems in a differentiable manner. This continuous-time framework has motivated biologically inspired architectures such as Liquid Time-Constant (LTC) Networks [4] and Neural Circuit Policies (NCPs) [5]. These models leverage structured neural circuits to enhance expressive capability, robustness, and interpretability. NCPs, in particular, have demonstrated promising results in challenging real-world reinforcement learning tasks by effectively capturing intricate temporal relationships.

Collectively, these advances underscore an ongoing trend toward developing models that balance expressiveness, sample efficiency, interpretability, and computational simplicity, a crucial intersection seldom achieved by existing reinforcement learning architectures.

C. Problem Definition

We consider an agent interacting with an environment, which can be a physical setting for a biological entity or a simulated domain for a computational system. This agent–environment interaction is often modeled as a Markov decision process (MDP), as illustrated in Figure 1.

Through this interaction, the agent observes a sequence of states and actions, denoted by $(s_0, a_0, s_1, a_1, \dots)$, or equivalently by two tuples $[(s_0, s_1, \dots), (a_0, a_1, \dots)]$. For the purpose of capturing state transitions, we omit the reward signal from this formulation. We thus define the transition dynamics of the environment via a probability function p :

$$p(s_{t+1}|s_t, a_t) = \Pr\{S_{t+1} = s' | S_t = s_t, A_t = a_t\}$$

for $\forall s', s \in \mathcal{S}, a \in \mathcal{A}$.

Given the rapid progress in deep learning, it is increasingly feasible to approximate p with neural networks, making data-driven models of the environment accurate and robust. However, despite these promising developments, model-based reinforcement learning—and particularly the use of neural networks as learned “world models”—remains relatively underexplored compared to model-free approaches. This work focuses on bridging that gap, aiming to advance our understanding of neural network-based world models within RL frameworks.

We investigate the challenge of making useful world models by experimenting with three distinct neural network architectures. Two of which have a well-established track record and extensive literature detailing their advantages for sequence modeling and temporal dynamics, among other tasks. The third architecture, Neural Circuit Policies, shows particular promise in domains that involve physical interactions and causal reasoning. By comparing these three approaches, we aim to highlight their strengths and limitations, ultimately guiding the development of more robust and interpretable world models for model-based reinforcement learning. We aim to test these architectures on environments displaying separate properties. We capture their effectiveness in each environment as determined by their performance in next-step prediction and n-step prediction. Additionally, we compare their abilities of generalization to samples found beyond the training set, as well their ability to capture dynamics with varying sample sizes.

II. METHODOLOGY

For the purposes of our paper, we wished to be convinced of the performance of specific neural network architectures as world models. We tested three architectures and three environments to measure the performance of the architectures. In addition, we varied the amounts of data the architectures were trained on to provide insight on their capability of sample efficiency.

A. Architectures

The three neural network-based architectures we used for this work are Neural Networks with a Residual Connection or a *Residual Block*, *Recurrent Neural Networks*, and *Neural Circuit Policies*.

Residual blocks, first popularized by ResNet architectures, enable neural networks to grow exceptionally deep while mitigating gradient-related training difficulties. The core idea is to learn a residual function—that is, the change from an

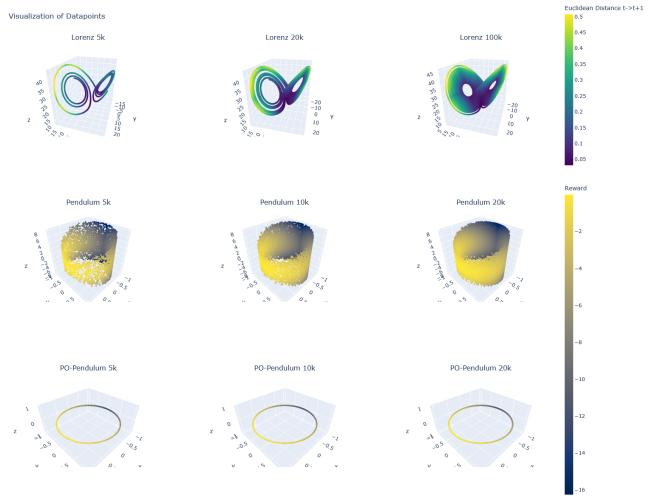


Fig. 2. Visualization of Data from random trajectories on each environment with three different amounts of sampling. The top row of plots describe samples from the Lorenz system, the second row is from Pendulum, and the third row is from Partially Observed Pendulum.

input to the desired output—rather than the direct mapping itself. Formally, for an input x_t , the updated state is given by.

$$x_{t+1} = f_\theta(x_t) + x_t$$

where f_θ represents a neural network. Residual connections bear a conceptual resemblance to Euler’s method in numerical analysis, in which an incremental update is applied to the current state to approximate the next value.

Recurrent Neural Networks (RNNs) model sequential data and time-series phenomena. At each time step t , an RNN cell receives the current input x_t and a hidden state h_{t-1} from the prior step, generating a new hidden state h_t and an output \hat{y}_t . This hidden state contains information about previous states, serving as a sort of memory, making RNNs well-suited for tasks with sequential relationships.

Neural Circuit Policies (NCPs) are a distinct architecture that blends concepts from the preceding methods, drawing particular inspiration from the compact nervous system of the *nematode C. elegans*, which features only a small number of neurons yet displays highly adaptive behaviors. This design integrates biologically motivated principles—sparse, layered neural circuits and continuous-time neuronal dynamics—with contemporary training techniques. Unlike standard RNNs, NCPs rely on two key innovations: (1) neurons governed by ordinary differential equations (ODEs), and (2) a sparse, structured connectivity pattern reminiscent of biological networks. Concretely, NCPs employ a four-tier hierarchical layout: *sensory neurons* receive external inputs, *interneurons* and *command neurons* jointly process information and make decisions, and *motor neurons* produce the final outputs. The specific wiring among these neuron layers is randomized according

to a *Bernoulli distribution*, controlled by hyperparameters that dictate the feed-forward connection probabilities.

During training, NCPs utilize backpropagation through time, but the unrolled computational graph must also include the internal ODE solver. The hidden state of the entire NCP $\mathbf{x}(t) \in \mathbb{R}^{D \times 1}$ evolves according to

$$\frac{d\mathbf{x}(t)}{dt} = -[\mathbf{w}_\tau + f_\theta(\mathbf{x}(t), \mathbf{I}(t))] \odot \mathbf{x}(t) + A \odot f_\theta(\mathbf{x}(t), \mathbf{I}(t))$$

where $\mathbf{I}(t) \in \mathbb{R}^{m \times 1}$ is an exogenous input, \mathbf{w}_τ and A are parameter vectors, f is a neural network parameterized by θ , and \odot denotes elementwise multiplication. Because gradients must propagate through each integration step, a neuron's current state influences the loss for all previous time points.

B. Environments

Each of the environments exhibits distinct characteristics that allow for rigorous evaluation of our architectures, testing their abilities across varying complexities and dynamics.

Our first environment is the **Lorenz System**, defined by the following ordinary differential equations:

$$\begin{aligned}\frac{dx}{dt} &= \sigma(y - x), \\ \frac{dy}{dt} &= x(\rho - z) - y, \\ \frac{dz}{dt} &= xy - \beta z.\end{aligned}$$

parameters $\sigma = 10$, $\rho = 28$, $\beta = \frac{8}{3}$. The Lorenz System is a canonical example of chaotic systems, characterized by deterministic yet highly sensitive dynamics dependent on initial conditions. Its chaotic nature implies that minute variations in initial states lead to exponentially diverging trajectories, making it particularly challenging for predictive modeling. Moreover, the absence of actions in this system isolates the difficulty of accurately capturing and generalizing complex temporal dynamics, enabling a focused assessment of our models' intrinsic predictive capabilities without the added complexity of inferring the impacts of external control inputs. The second environment is the **Pendulum** from *OpenAI's Gym*, which represents a simplified yet richly informative physical system possessing Markovian dynamics. The environment's state encapsulates the Cartesian coordinates of the pendulum's tip along with its angular velocity, while the action applies torque directly to the end of the pendulum. The objective is to stabilize and balance the pendulum upright. This environment introduces the challenge of explicitly modeling the interplay between state transitions and control actions, thus evaluating our models' proficiency in handling action-dependent dynamics and control-oriented predictive tasks. The third environment is the **Partially Observed Pendulum**, a variant of the previous Pendulum environment. This environment removes the angular velocity component from the observable state, resulting in a non-Markovian setting. The lack of full observability demands that models infer latent dynamics through temporal integration.

C. Experiments

We evaluated the models prediction generalization capabilities on given environments using their autoregressive performance, defined in algorithm 1, as the cumulative sum of errors sustained by the models will illuminate their performance. Each of the models were trained on some training set to convergence on one-step predictive accuracy. For Lorenz Systems, we place importance on the model's ability to approximate the underlying ODEs which define the system. We track the trajectories achieved by the models when initial conditions are changed dramatically, achieved by negating the initial conditions on the Z-axis. We look for deviations in one step and autoregressive prediction, evaluating specifically on distance to the original path and similar patterns of progression. For Pendulum and Partially Observed Pendulum, we initialized each model at a variety of points found both within and outside of the training set. By analyzing how well the models follow the true pendulum motion in an autoregressive manner, we captured their capacity to maintain low prediction error over time while staying within the regions they were trained on. We distinguish a maximum tolerance for residuals at 0.24, which is the average residual given by the identity (which predicts the same state it is given). We use this threshold to suggest when predicted trajectory has strayed significantly from the sampled trajectory. By performing these autoregressive rollouts numerous times, we get a statistical measure of the performance of each of those models. We trained each model on datasets of varying sample sizes to fully capture how the amount of available data influences learning and generalization. Smaller datasets reveal a model's capacity to handle data-scarce conditions, while larger datasets test the model's ability to leverage more abundant information. The different datasets we created has been illustrated in Figure 2

Algorithm 1 Autoregressive Rollout

```

Inputs
WM                                     ▷ world model
States                                    ▷ reference trajectory states
Actions                                    ▷ reference trajectory actions
T                                         ▷ rollout length
wl                                        ▷ window_length

Array Error
Array Window ← (States[:wl], Actions[:wl])
while  $i < T$  do
    prediction ← WM(Window)
    Error[i] ← States[i+1] - prediction
    Window ← (Window[1:] + prediction, Actions[i:i+wl])
end while

```

III. RESULTS

A. Lorenz Systems

We primarily care about the models effectiveness at capturing the underlying dynamics of the ODE at play: can

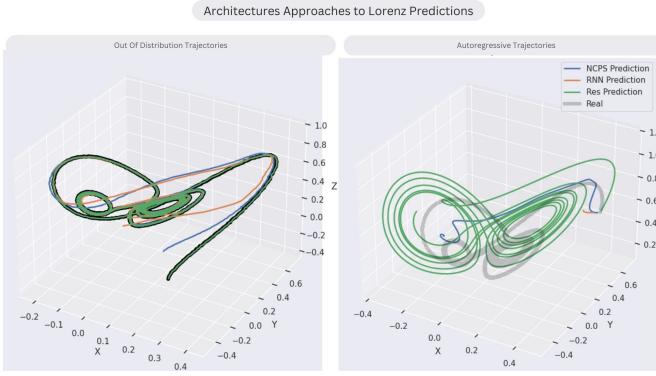


Fig. 3. Graphs detailing architectures trajectory on Lorenz System: OOD and Autoregressive Rollouts

the models generalize to out of distribution data, and can they maintain lobe-switching characteristics when applied auto regressively. We see on one-step predictions on *in distribution data*, all models performed effectively equivalently as shown in Figure 3. We see a switch however, with only the Residual Block being capable of generalizing to the out of distribution data. Further, we see that the residual block maintains lobe-switching characteristics while the other two models fail to do so, and falling into stable points instead. While the RNN's relatively poor performance may be attributed to limitations inherent in its structure, there is intrigue with the distinct performance gaps between NCPS and the Residual Block. We hypothesize they are training on fundamentally separate tasks, with the resnet approximating a vector field with some margin allowed in the precision of it's approximation. Conversely, the NCP architecture appears to attempt a more rigid, closed-form representation of the underlying dynamical equations, leaving it more susceptible to cumulative errors and sensitive to deviations during autoregressive inference.

B. Pendulum

On in-distribution predictions, the residual connection and RNN models are comparable when trained on 200 samples. However, the residual connection model performs well with only 50 samples, while the RNN needs all 200 for the boxplot to display a similar median. This observation implies that the residual connection model is more sample efficient. The NCPS models have worse performance on all training set sizes. Plots depicting the performance of the 3 architectures trained on varying numbers of samples is in Figure 4.

On out of distribution tests, performance is worse across the board. This is expected, as the models have not been trained on these trajectories. However, this also suggests that the models are not learning the underlying causal dynamics of the system, only approximations at points they have been trained on. That said, of the three, the residual connection and RNN models perform the best. Performances are depicted in Figure 4.

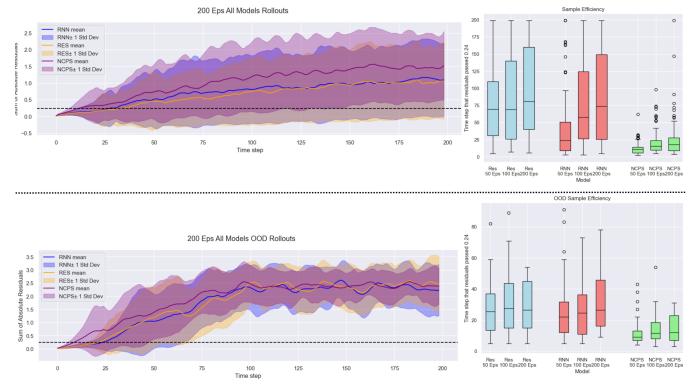


Fig. 4. These box plots describe at which step individual predicted trajectories cross the 0.24 threshold. These results are from 150 different random trajectories.

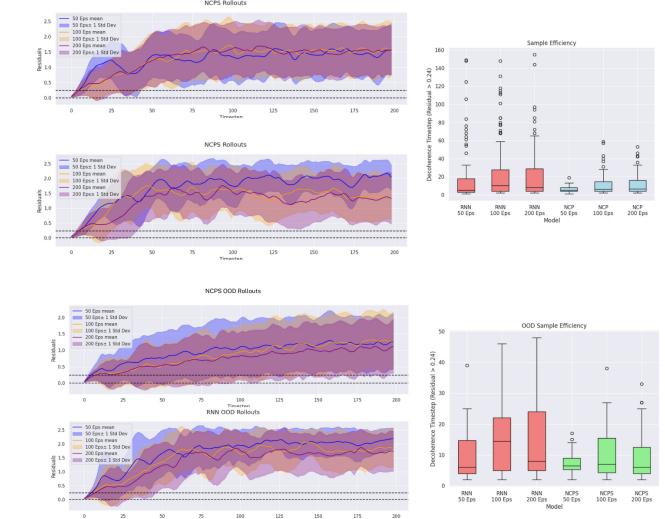


Fig. 5. Graphs detailing architectures performance on training set, Out-of-Distribution Lorenz System, and Autoregressive Rollouts

C. Partially Observed Pendulum

Partially Observed Pendulum excludes the velocity from the observation, which makes the environment non-Markovian. Therefore in order for models to learn their dynamics, they must have a recurrent component. This feature excludes the recurrent connection model from these tests.

As expected from the restricted information, the performances decreases relative to the regular Pendulum environment. Similar to the Pendulum environment, the NCPs are outperformed by the RNNs, albeit by a smaller margin.

IV. CONCLUSION

This work explores the gap between the theoretical benefits of world models, and their immature status in literature. We test 3 neural network architectures - neural networks with a residual connection, recurrent neural networks, and

Neural Circuit Policies - in approximating the dynamics of 3 environments: the Lorenz system, Open AI gym's Pendulum, and a modified, partially observed Pendulum. The residual connection model performed best on Pendulum and the Lorenz system, in terms of both sample efficiency, and residuals. This means regardless of the number of samples the models were trained on, the residual connections' predicted trajectories remained closest to the ground truth, for longest. The performance of the RNN is close behind for larger sample sizes. In all cases, NCPs place third. In Partially Observed Pendulum, for which the residual connection is omitted, performance across the board is worse. This is expected, as the models have less information to work with.

Future work entails integrating these architectures into model based learning with PPO agents and compare their agents respective decision-making performance. A larger issue is forming a hypothesis for why NCPs are under-performing in our benchmarks, and identifying a set of environments better suited for NCPS.

REFERENCES

- [1] M. Janner, J. Fu, M. Zhang, and S. Levine, "When to trust your model: Model-based policy optimization," 2019.
- [2] D. Hafner, J. Pasukonis, J. Ba, and T. P. Lillicrap, "Mastering diverse domains through world models," 2023. [Online]. Available: <https://api.semanticscholar.org/CorpusID:255569874>
- [3] R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. Duvenaud, "Neural ordinary differential equations," 2018.
- [4] R. Hasani, M. Lechner, A. Amini, L. Liebenwein, A. Ray, M. Tschaikowski, G. Teschl, and D. Rus, "Liquid time-constant networks," 2021.
- [5] M. Lechner and R. Hasani, "Neural circuit policies enabling auditable autonomy," *Nature Machine Intelligence*, vol. 2, pp. 642–652, 2020.
- [6] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. MIT Press, 2018.

ZoningLLM – A Novel Multimodal Application for Zoning Analysis

Simha Kalimpalli

University of Waterloo

nkalimpip@uwaterloo.ca

Saurodeep Majumdar

University of Waterloo

sdmajumdar@uwaterloo.ca

Kevin Tan

University of Waterloo

kx3tan@uwaterloo.ca

Jonathan Feng

University of Waterloo

j25feng@uwaterloo.ca

Rahul Kumar
University of Waterloo
r77kumar@uwaterloo.ca

Liam Dachner
University of Waterloo
ldachner@uwaterloo.ca

Abstract—Large municipalities in Canada have recently faced an unprecedented housing crisis. This has been driven by an increase in the demand for housing and a lack of housing supply. Stringent zoning requirements have contributed to reducing the construction of new housing. Each municipality typically has its own separate zoning code consisting of lengthy documents written in technical jargon. It is difficult for the public, researchers, and home builders alike to extract relevant information from these documents. This opacity restricts the discussion of zoning policy and aggravates the housing crisis. This project aimed to use generative and geomatic AI methods to analyze zoning and construction documents for the Waterloo Region to gain insights about zoning restrictions. This can be used to quantify and monitor the effects of zoning on housing supply. A web-based application with the ability to process, export and query ad-hoc zoning queries has been developed. Discussions with regional planners have underscored the importance of this work.

Keywords: Large-Language Models, Zoning, Housing

I. INTRODUCTION

A. Background

Zoning regulations for a municipality typically consist of several main zoning districts and respective subdistricts [1]. Maximum and minimum requirements for each zoning district are typically listed. Additional, special clauses are often stipulated for special overlapping districts, grandfathered lots and other miscellaneous information. Zoning conditions are reflective of current conditions of a given plot of land rather than development priorities. Official plans are a document, which reflect municipal intensification and urban growth priorities. They are often less structured than Zoning regulations and reflect the long-term vision of the municipality. Both zoning regulations and Official Plans are important for assessing future housing supply [2].



Fig. 1. Uptown Waterloo Zoning: redevelopments (at arrow) and zoning constrained neighbourhoods (R4) are pictured

B. User Needs Research

We conducted research to assess important factors for zoning codes and better understand the development process. We spoke to current urban planners at the Region of Waterloo, a development lawyer in British Columbia, and other key stakeholders related to zoning.

We researched current procedures by 3 stakeholders (developers, researchers and policymakers) which could be improved by a zoning related application:

1) Developers

They follow 5 steps in development:

- 1) **Find macro purpose:** They see where there is a mismatch between official plan (long term vision), zoning codes (current conditions) and aggregated demand to identify potential for housing/development.
- 2) **mid-level opportunity identification:** Developers evaluate initial plans based on criteria, such as financing, location and granular demand.
- 3) **Micro-level Analysis:** They figure out if local infrastructure (transportation, electrical, waste water etc) supports rezoning. City council attitudes towards previous house development is also an important consideration. This includes future plans/approvals data and the presence of density bonusing incentive programs.
- 4) **Comprise for micro level decisions:** Negotiations with Policy-makers and local groups occur until a compromise is reached, and compliance is met. The developer typically aims for either highly profitable or specialty “landmark” projects.

5) **Commencement of Construction** Actual construction of buildings occur.

2) *Researchers*

They currently manually parse through zoning codes to extract useful information. This is often a manual and slow process. They are interested in correlation of these zoning code data with development patterns. They need to quickly and systematically collect zoning data for extensive analysis to be feasible.

3) *Policy Makers*

They want to increase transparency of open data, investing in new useful tools to help with that goal. They want to make better policy decisions (eg. seeing where official plans/zoning codes are adequate/inadequate and changing zoning). They have to negotiate with developers and community local groups to reach a compromise on final building plans.

C. Related Works

It is important to assess the relevant literature regarding scraping information from zoning regulations:

1) Academic Literature

We were inspired by the Urban Institute's work on Automating Zoning Data Collection. Their seminal paper discusses creating a unified zoning database for manual scraping of data from the zoning districts within municipalities. LLMs gain traction for decoding regulatory information across various domains, such as the financial and health sectors [3]. Barthik, Gupta et al discuss using Large Language Models to decode zoning statutes in the United States. [4]. This paper also discusses creating a unified database for zoning regulatory information, but the data granularity is at the municipality level (where each municipality is assessed as a whole) unlike the Urban Institute's work.

2) Industry Efforts

There are also several startups doing relevant work: Trax.co is a startup that is working on creating LLM accessible building codes in Ontario [5]. Arterial.design is a Boston based startup that "automates decision-making for policy-driven organizations" [6]. Up.codes is a platform to streamline code compliance for architects, homebuilders and inspectors [7]. Finally, Autoprop is a Vancouver-based company that provides data automation solutions for real estate professionals. These companies demonstrate the market viability of using LLMs, web mapping software to create databases that simplify regulatory information [8].

3) Literature Conclusions

Based on the literature, relevant companies and interviews conducted, our design project aims to be the first project to use multimodal LLMs to automatically retrieve zoning data by zoning districts and visualize and analyze this data through a geomatic lens.

II. METHODOLOGY

A. Problem Definition

The team proposes using NLP methodologies such as Retrieval Augmented Generation to 1) simplify the verbosity of various zoning documents to illustrate what can and cannot be built

in a certain area (ie. by chatbot). 2) Extract exact details from zoning documents about setback and density. 3) Use classical ML techniques to correlate extracted zoning information with housing development in a given municipality. 5) Create a unified framework to compare the zoning of neighborhoods across the 3 Urban municipalities in the Region of Waterloo.

The ultimate goal of the application would be for the general public to have increased awareness of zoning laws, while providing a useful resource for researchers and policy makers.

B. Dataset Design

1) Phase 1

We then collected zoning regulations for the 3 Urban municipalities of Waterloo Region that are available as PDFs on the municipalities' respective "Zoning and Building" pages. For example, Kitchener's zoning data is found on their Open Data site [9]. This formed the backbone of our data repository. We then exported this data into PDFs which would be later be fed to the LLM [10].

2) Phase 2 and beyond

The geographical areas of zoning codes in the region were owned by a private organization. Hence, we had to manually trace sample urban districts using Google Maps. We converted this data to GeoJSON and would later combine it with the zoning data for each respective district. We would corroborate additional housing information sourced from Zillow and Waterloo Region Connected. Geographical datasets that correspond to transportation, energy utility and Water infrastructure were also exported from municipal open data sources.

C. Technology used

We strived to identify high-accuracy, cost-efficient tools and technologies for the project.

LLM + RAG Based projects typically have raw textual data that is sent into a vector database, which first turns textual meanings into vectorial representations, and stores the relevant information. Afterwards this data is sent as context to a foundational Large Language Model, that aims to find the most similar internal source data to a prompt [11].

Front End Frameworks are required for the GUI of the application. The front end would either use custom made or ready-made JavaScript/HTML/CSS rendering templates (ie. through Streamlit or Azure).

Back End Frameworks are used logic of the application. Most of the backend of this project would be done in the Python language in conjunction with various LLM APIs. We initially hoped to evaluate the use of several models such as GPT, Claude, Gemini and LLAMA for the LLM similar to other projects. We are evaluating the use of Langchain, and the "Agentic" RAG options for the various LLMs. We are also looking at cloud computing/deployment services such as Heroku, Amazon Web Services or Azure to deploy to application when completed.

III. RESULTS

A. Phase1 - Base Tested LLM

1) Iteration1

The first iteration used the Azure CosmosDB Vector Database, GPT 3.5 Turbo as the LLM and had an Azure studio hosted webapp for the Interface (Backend + GUI).

We noticed that while this version was easy to set up and worked decently for small question, The performance was degraded for more complicated queries (circular and non answers were common). Additionally model hyper-parameters were not able to be extracted. Hence a second iteration was explored.

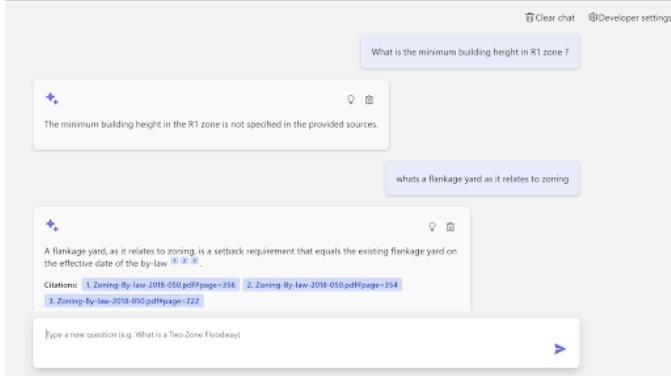


Fig. 2. Chatbot with Testing Questions

2) Iteration2

A second iteration of the product was also made after further research. This version uses the FAISS Database, GPT 4 Turbo LLM, Streamlit for the front end and Langchain for orchestration. This version is currently locally hosted. Testing of the second iteration also included the same sample questions had the first one but also included some real hyperparameters: model, chunk size, chunk overlap and temperature. The model is adjustable using GPT 3.5 Turbo GPT 4 Turbo currently (other models undergoing testing).

Chunk size refers to the largest character size of a text chunk used for embedding. Chunk overlap is the number of characters that are overlapped between two chunks that are next to each other for vectorization. Temperature refers to the amount of randomness of chatbot results. [12].

TABLE I
HYPERPARAMETER OPTIONS AND LIMITS.

Hyperparameter	Tuning limits
Chunk size	[100,2000]
Chunk Overlap	[10,200]
Temperature	[0.05,0.3]
Zoning Codes	[R1,R4,RMU]

A set of 13 Questions were prompted to the LLM which were sourced from the Barthik and Gupta paper, the National Zoning Atlas paper, and advice from current urban planner (s) [4] [10].

- 1) What is the minimum building height?
- 2) What is the maximum building height?
- 3) What is the minimum street line setback?
- 4) What is the maximum street line setback?
- 5) What is the minimum density?
- 6) What is the maximum density?
- 7) What are minimum frontage requirements for single family residential development?
- 8) What are maximum frontage requirements for single family residential development?
- 9) Are apartments above commercial (mixed use) allowed?
- 10) Is multi-family housing allowed, either by right or special permit (including through overlays or cluster zoning)?
- 11) Are attached single family houses (townhouses, 3+ units) listed as an allowed use (by right or special permit)?
- 12) Are accessory dwelling Units (ADUs) or in-law apartments allowed (by right or special permit)?
- 13) Is cluster development, planned unit development, Planned Residential Development (PRD) open space residential design, or another type of flexible zoning allowed by right?

We customized questions by attaching the word "in" and one of 3 zoning codes, Residential1, Residential4, and Residential Mixed-use zone to the end of questions. Additionally, we ran each configuration at least 5 times due to the stochasticity of LLMs (which could output different result each time) to ensure reliability.

TABLE II
BEST PARAMETERS AND VALUES FOR DIFFERENT ZONES.

Zone	(Chunk Size, Chunk Overlap, Model, Temperature)	Best Accuracy
R1	(800, 100, gpt-3.5-turbo, 0.26)	1.0
R4 - SINGLE DETACHED	(1000, 100, gpt-3.5-turbo, 0.12)	0.67
R4 - SEMI-DETACHED and DUPLEX	(800, 100, gpt-4-turbo, 0.35)	0.83
R4 - FREEHOLD SEMI DETACHED	(600, 100, gpt-4-turbo, 0.012)	0.83
RMU-20	(800, 100, gpt-4-turbo, 0.10)	0.83
RMU-30	(1000, 100, gpt-4-turbo, 0.05)	0.83
RMU-40	(600, 100, gpt-4-turbo, 0.49)	1.0

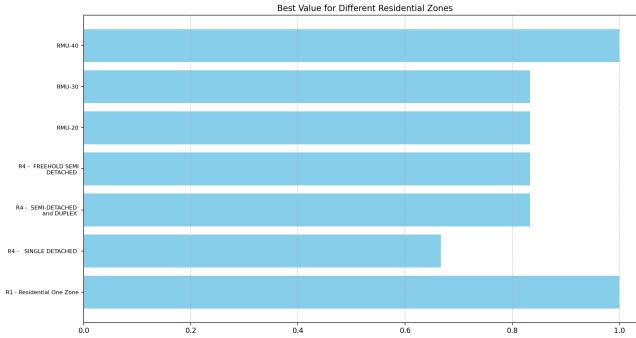


Fig. 3. Accuracy over different zones

Results indicate that the model is able to perform accurately across different zoning and sub-zoning districts. The higher performance is shown with GPT-4-turbo model using lower value of temperature. The model is able to do considerably well even in zones that have multiple subzones (ie. RMU). This indicates a high degree of ability for the model to a) read and comprehend tables with requirements for multiple subzones as well as b)retrieve the relevant content from its database to be able to answer the user queries. It can be seen that a moderate chunk size between 600-800 tokens is sufficient to allow for retrieval of relevant content. "R-4 - Single Detached" was the only zoning code with significant retrieval issues.

Future Phases of the project use the average of the best recall hyperparameters for each zoning code identified in Phase 1.

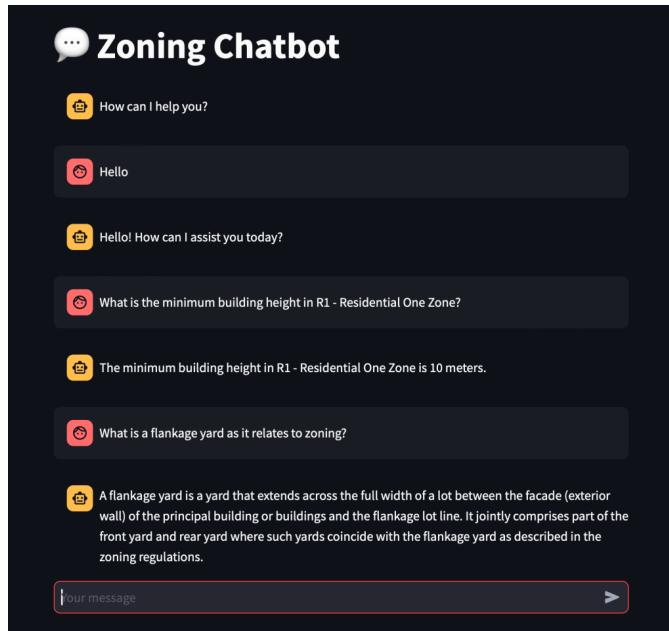


Fig. 4. Second Iteration of Chatbot with testing questions

B. Phase2 - Query and Download

For this phase, Given a suitable query (with municipality, zoning code, sub-zoning code and chosen metrics), the system would output the requested information in a structured format (i.e. in JSON or CSV format) along with an AI-generated summary of zone. The data would come mostly from the zoning code and official plan(undergoing tests).

An example of such a query would be:

Municipality : [Waterloo];

Zoning codes : [Residential3 , Residential5];

Zub-zoning codes : [All];

Metrics : [maximum height , minimum height , maximum density , minimum density]

We developed random test questions (with known answers) to test the system. Final Output types after prompt engineering are shown below:

```

{
  "Municipality": "City of Waterloo",
  "Zoning": "R1 - Residential One Zone",
  "Metrics": [
    {
      "Metric": "Lot area", "Unit": "square metres", "Requirement": "Interior Lot: >= 405, Corner Lot: >= 540" },
    {
      "Metric": "Lot frontage", "Unit": "metres", "Requirement": "Interior Lot: >= 13.5, Corner Lot: >= 18" },
    {
      "Metric": "Front yard setback", "Unit": "metres", "Requirement": ">= 7.5" },
    {
      "Metric": "Flankage yard setback", "Unit": "metres", "Requirement": ">= 6" },
    {
      "Metric": "Side yard setback", "Unit": "metres", "Requirement": ">= 1.8" },
    {
      "Metric": "Rear yard setback", "Unit": "metres", "Requirement": ">= 7.5" },
    {
      "Metric": "Building height", "Unit": "metres", "Requirement": "<= 10" },
    {
      "Metric": "Lot coverage", "Unit": "%", "Requirement": "<= 35" },
    {
      "Metric": "Parking spaces", "Unit": "per dwelling unit", "Requirement": ">= 1" },
    {
      "Metric": "Number of buildings", "Unit": "count", "Requirement": "1 main building per lot" }
  ]
}

```

Fig. 5. Sample JSON output of Query and Download Feature

C. Phase3 - Select and View

Geographic Information pertaining to zoning code districts and Future growth areas was aquired from municipalities in a GeoJSON format. We merged our textual zoning data output for each municipality (from the Query and Download Feature) and our geographic data to create a single layer. We would display this information as an interactive layer in the zoning LLM app. We made a system such that queries from our chat bot pertaining to geographical information would be converted to SQL and all districts meeting the query criteria would be selected. A sample query would be "Select all areas with a maximum height of 10" meters". We cross referenced results with the actual zoning map to ensure accuracy of this feature. We used the MapBOX mapping API for this and future phases with mapping.

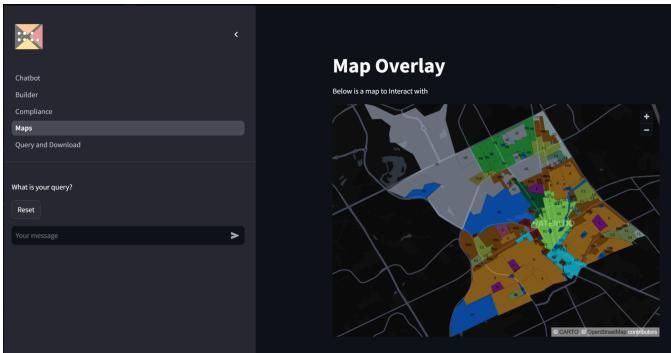


Fig. 6. Map of Zoning Codes in Central Waterloo

D. Phase4 - Historical Data

In addition to the zoning information recall identified in previous phases, We would also add housing supply and demand information from various sources to our mapping/chatbot database. We connect to the APIS of real estate websites such as Zillow for information about units on the market, and Local development forums such as UrbanToronto, or Waterloo RegionConnected which keep detailed databases of recent construction products with information about the size, builders, and number of units [13].

This information could geographically either be in a “point format” (real estate, development information), where each point would represent a unit/development respectively; or in a “area” format for broader neighborhood level insights.

Boolean search operations (NOT, AND, OR) could be used to identify interaction zones between various zoning and housing data. By comparing the various intersection zones (restrictive zones/places with lots of development areas) we can could quantify correlations between zoning regulation and housing development. However, current testing has been restricted to qualitative analysis for the time being.

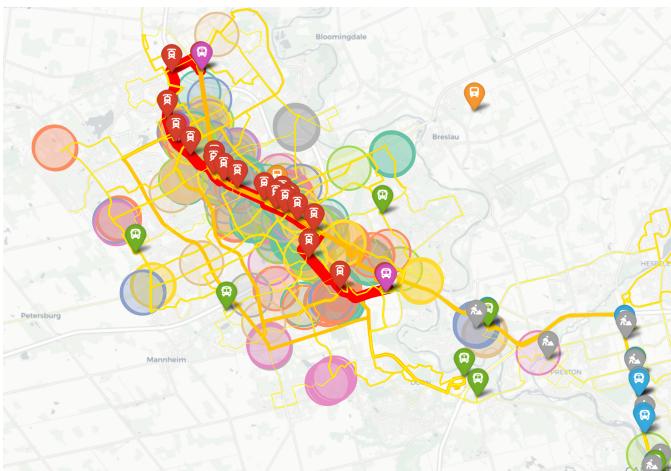


Fig. 7. Map of recent multi-story developments and transport infrastructure in Waterloo

E. Phase5 - Development Compliance and Visualization

Once a plot is selected for development, developers need to construct an example of massing for a proposal. This is a time consuming process that requires constructing detailed 3D models of a proposed structures. These models need to be remade regularly as the development concept changes. Thus, we have developed a system that is able to generate renders and 3D models of massing structure. The user would input the height (in storeys) of the building, the shape (L shape, rectangular etc.), the location (suburban or urban) and a model of the structure is generated using the MapBOX mapping library. Given the maximum building sizes/densities in a zoning code, the application says if the building does not meet the zoning code and it suggests similar but feasible structures.

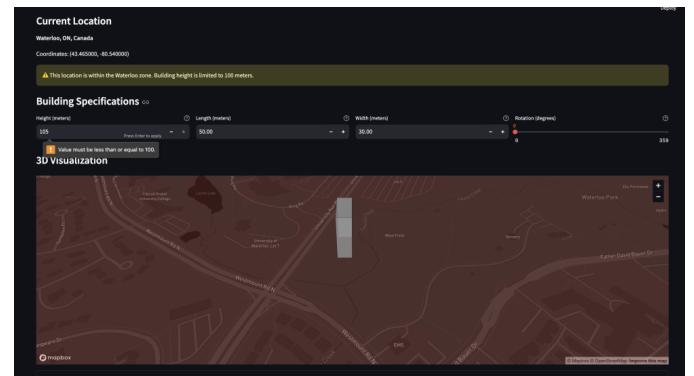


Fig. 8. Sample compliance feature notification

The OpenAI (DALL-E) API is used to create a render of the requested building. Users have the option to download their generated media.



Fig. 9. Sample rendered building

IV. CONCLUSION

V. KEY FINDINGS

The housing crisis is a prominent issue in Canada [14]. Zoning codes are one of the factors which limit housing supply. Generative models have been shown to be effective at extracting relevant information from regulatory documents. This project has applied these models to extract related information from zoning code documents to quantify effects of housing regulation. Extensive benchmarking undertaken during this project has demonstrated that these models indeed show promise in extracting textual information and generating useful insights in geographical, image and textual domains. Researchers and relevant individuals who were contacted during the research process have also shown interest in this technology. Thus, the team will continue to develop this model for enterprise purposes in the future. Specifically features to further analyze historical housing and infrastructure data will be pursued.

VI. LIMITATIONS

Hyperparameter Testing only used GPT-3.5 and GPT-4 as LLMs due to budget constraints. Several features were not able to be implemented due to ongoing difficulties in obtaining data (eg. real time rent prices). Although various chunking sizes were used in the RAG process, the chunking process used simple text splitting to extract text from tables. In the future, usage of OCR and object detection techniques can lead to higher accuracy and precision metrics from the model.

VII. ACKNOWLEDGMENTS

We would like to thank WAT.ai for their helpful resources. We would also like to acknowledge Sustainable Waterloo Innovation Lab for their efforts in bringing this app to reality. We

would finally like to thank Y-Combinator startup Hamming.ai for their advice on LLM benchmarking techniques.

REFERENCES

- [1] K. P. Green and J. Filipowicz, "Barriers to housing supply in ontario and the greater toronto area," 2023. [Online]. Available: <https://www.fraserinstitute.org/sites/default/files/barriers-to-housing-supply-in-ontario-and-the-greater-toronto-area.pdf>
- [2] X. Gabaix, "Zipf's law for cities: An explanation," National Bureau of Economic Research, Working Paper 8835, 2002. [Online]. Available: https://www.nber.org/system/files/working_papers/w8835/w8835.pdf
- [3] Z. Cao and Z. Feinstein, "Large language model in financial regulatory interpretation," Jul 2024. [Online]. Available: <http://arxiv.org/abs/2405.06808>
- [4] A. Bartik, A. Gupta, and D. Milo, "The costs of housing regulation: Evidence from generative regulatory measurement," *SSRN Electronic Journal*, 2023.
- [5] Tra.co. [Online]. Available: <https://www.trax.co/>
- [6] R. Kothari, "Automated land development." [Online]. Available: [https://www.artelial.design/](https://www.arterial.design/)
- [7] Upcodes. [Online]. Available: <https://up.codes/>
- [8] AutoProp. [Online]. Available: <https://autoprop.ca/>
- [9] C. of Kitchener, "Open data," accessed: 2025-03-17. [Online]. Available: <https://www.kitchener.ca/en/council-and-city-administration/open-data.aspx>
- [10] J. Axelrod, L. Lo, and S. C. Bronin, "Automating zoning data collection," Urban Institute and Cornell University, Tech. Rep., feb 2023, accessed: 2025-03-17. [Online]. Available: <https://www.urban.org/sites/default/files/2023-02/Automating%20Zoning%20Data%20Collection.pdf>
- [11] K. Pandya and M. Holia, "Automating customer service using langchain: Building custom open-source gpt chatbot for organizations," 2023. [Online]. Available: <https://arxiv.org/abs/2310.05421>
- [12] D. Lin, "Revolutionizing retrieval-augmented generation with enhanced pdf structure recognition," 2024. [Online]. Available: <https://arxiv.org/abs/2401.12599>
- [13] zillow. [Online]. Available: <https://www.zillow.com/>
- [14] K. P. Green and J. Filipowicz, "The impact of land-use regulation on housing supply in canada," 2024. [Online]. Available: <https://www.fraserinstitute.org/sites/default/files/impact-of-land-use-regulation-on-housing-supply-in-canada.pdf>