

A Thesis on

Evaluating adversarial alignment subversion in generative language models

by
Russel Shawn Dsouza, 2454659

under the supervision of
Dr Venelin Kovatchev
Assistant Professor,
School of Computer Science

for the degree of
**Master of Science in Artificial Intelligence and
Machine Learning**

submitted to the
School of Computer Science
University of Birmingham

on
September 18, 2023

⚠ Disclaimer

This research involves model outputs and datasets that contain contents which may be offensive or malicious in nature and in no way reflect the views and intents of the author or the University.

ABSTRACT

Increasingly powerful language models are being deployed into the real-world often with inadequate testing and a lack of robust guardrails. It is thus imperative to explore both theoretical and empirical bounds for such generative language models to mitigate potential harm and maximise beneficial deployment. This research explores the use of language models to automate the evaluation of alignment subversion in other more powerful language models. The project experiments with token importance estimation and bayesian optimisation, comparing the performance of six masked language models in different prediction regimes. The proposed approach focusses on toxicity- and bias-based alignment and demonstrates the feasibility of detecting misalignment at scale. Overall, the project delivers empirical insights to enable safer, more responsible deployment of increasingly influential language models.

Contents

ABSTRACT	1
1 Introduction	2
2 Contributions	3
3 Background	3
3.1 Language Models	3
3.2 Large Language Models	3
3.3 Foundation Models	4
3.4 Alignment	4
3.5 Adversarial Attacks	5
3.6 Threat models	5
3.7 Adversarial Defence	6
4 Literature Review	8
4.1 Gradient-based Attacks	8
4.2 Gradient-free attacks	8
4.3 Unauthorised Code Execution	8
4.4 Membership Inference	9
4.5 Backdoor attacks	9
4.6 Universal attacks	9
4.7 Prompt Injection	10
4.8 Toxicity, Bias, Stereotypes	10
4.9 Evaluation Metrics	11
4.10 AI Feedback	12
5 Data	13
5.1 Exploratory Data Analysis	14
5.1.1 Transcripts	14
5.1.2 Toxicity	14
5.1.3 Adversarial Attack Success Ratings	14
5.2 Specific Examples	16
5.3 Correlation with toxicity	18
6 Experimental Setup	19
7 Experiments & Results	21
7.1 Multi-class classification	21
7.2 Binary classification	23
7.3 Regression	25
7.4 Regression after fixing class imbalance	27

Adding Special Tokens	28
Bayesian Hyperparameter Estimation	28
Bayesian Hyperparameter Estimation	29
7.5 Multi-class classification with special tokens	31
7.6 Binary classification with special tokens	33
7.7 Regression with Special Tokens	35
8 Discussion and Conclusions	37
9 Future Work	38
9.1 Mechanistic Interpretability	39
References	40
10 Appendix: Code Structure	47

List of Tables

1 Recent Foundation Models	4
2 Results for baseline multi-class classification	21
3 Results for baseline binary classification	23
4 Results for regression after fixing class imbalance	27
5 Final results	38

List of Figures

1	A timeline of research in adversarial attacks on LLMs.	8
2	A pictorial representation of the keys in each sample of the dataset.	13
3	Wordclouds	15
4	Distribution of ratings and toxicity across the dataset.	16
5	Examples from extreme ends of the dataset	17
6	Joint-distribution of ratings and toxicity across the dataset.	18
7	A visual description of the experimental setup for this project.	20
8	A compact visual description of the different models used in the project.	20
9	Multiclass classification training and validation loss curves for 160k tokens.	22
10	Multiclass classification validation metrics curves for 160k tokens.	22
11	Binary classification training and validation loss curves for 160k tokens.	23
12	Binary classification validation metrics curves for 160k tokens.	24
13	Regression training and validation loss for 50k tokens	25
14	Regression validation metrics for ~50k tokens.	25
16	Label Distribution Smoothing	26
17	Regression with class-imbalance fix: Training and validation loss for ~50k tokens.	27
18	Regression with class-imbalance fix: Validation metrics for ~50k tokens.	27
19	Illustrative representation of addition of special tokens	30
20	Parallel plot from the hyperparameter sweep for multi-class classification fine-tuning a <code>albert-base-v2</code>	31
21	Hyperparameter importance plots for multi-class classification fine-tuned on <code>A1BERT</code>	31
22	Parallel plot from the hyperparameter sweep for multi-class classification fine-tuning a <code>longformer-mini-1024</code>	32
23	Hyperparameter importance plots for multi-class classification fine-tuning a <code>longformer-mini-1024</code> .	32
24	Parallel plot from the hyperparameter search for binary classification fine-tuning an <code>albert-base-v2</code> .	33
25	Hyperparameter importance plots for binary classification fine-tuning an <code>albert-base-v2</code>	33
26	Parallel plot from the hyperparameter sweep for binary classification fine-tuning a <code>longformer-mini-1024</code> .	34
27	Hyperparameter importance plots for binary classification fine-tuning a <code>longformer-mini-1024</code> .	34
28	Hyperparameter search results for regression, fine-tuning an <code>albert-base-v2</code>	35
29	Hyperparameter search results for regression fine-tuning a <code>longformer-mini-1024</code>	36

1 Introduction

Large language models (LLMs) such as GPT-3 [Brown et al., 2020], PaLM [Anil et al., 2023], and Claude [Anthropic, 2023] have demonstrated impressive capabilities in generating fluent, coherent, and seemingly intelligent text. LLMs leverage the transformer architecture [Vaswani et al., 2017] to learn powerful semantic representations from vast amounts of text data. By pre-training on diverse corpora, LLMs acquire broad linguistic capabilities and world knowledge. Fine-tuning on downstream tasks then adapts these pre-trained models for specific applications. Subsequently, continual scaling of model size and training data has yielded ever more capable LLMs [Kaplan et al., 2020]. Their few-shot learning abilities enable prompting for custom tasks without additional training, making deployment more accessible.

However, concerns have emerged about the propensity of large models to perpetuate and amplify harmful societal biases found in training data [Welbl et al., 2021]. Recent work has shown that their impressive fluency can disguise generation of toxic, racist, sexist, or otherwise dangerous content [Gehman et al., 2020]. Researchers have demonstrated that carefully constructed malicious prompts can reliably induce a wide range of unethical behaviours from LLMs [Wallace et al., 2019, Perez and Ribeiro, 2022]. This poses risks if models are deployed into production environments and interactive applications without sufficient safeguards [Ganguli et al., 2022]. Adversaries can exploit vulnerabilities to spread misinformation, radicalisation, or cause other serious harms through LLM-powered platforms [Wang et al., 2023]. Recent incidents with public LLMs like Meta’s Galactica, which despite being trained on millions of scientific sources indiscriminately produced biased and erroneous content and had to be taken down, demonstrate the severity of this threat [Heaven, 2022].

To mitigate risks and improve safety, several techniques have been proposed to align LLMs with human preferences. The goal of these techniques is to steer models away from harmful behaviours by incorporating beneficial objectives during training. Popular approaches include adversarial filtering [Le Bras et al., 2020], conditioning on social context, learning from human feedback [Ouyang et al., 2022] among others. For example, Anthropic’s Constitutional AI methodology [Bai et al., 2022] uses reinforcement learning from human ratings to make models helpful, harmless, and honest.

However, the effectiveness and robustness of alignment techniques remains questionable. Adversaries have and continue to find ways to subvert alignment through carefully designed prompts [Zou et al., 2023]. Therefore, developing techniques to automatically evaluate alignment integrity and detect successful attacks on LLMs, at scale, is crucial.

This research investigates using separately fine-tuned language models to assess adversarial alignment subversion in generative language models. Specifically, the project fine-tunes multiple masked language models on a dataset of adversarial human interactions with large language models. The fine-tuned models classify conversation transcripts to predict attack success ratings. This setup is evaluated across different models, training and prediction regimes. Overall, the goal is to empirically demonstrate the feasibility of automated adversarial alignment auditing for safer LLM deployment.

2 Contributions

This research work makes the following key contributions:

- Reviews recent literature on adversarial attacks and defences, alignment techniques and their vulnerabilities in-depth.
- Compares multiple masked language models fine-tuned on a dataset of adversarial human interactions with LLMs evaluating performance across different training regimes and prediction tasks.
- Examines the effectiveness of Bayesian hyperparameter optimization for improving prediction performance.
- Empirically demonstrates the feasibility of using language models to automatically evaluate adversarial alignment subversion in other language models enabling the detection of misalignment at scale.

3 Background

3.1 Language Models

Vaswani et al. [2017] proposed the Transformer to replace traditional recurrent and convolutional neural network layers which presented an improvement over the vanilla attention mechanism [Bahdanau et al., 2015] with ‘self-attention’, that allows the Transformer to learn global dependencies between the words in the sequence, enabling the generation of more informative and context-sensitive word embeddings. The authors also described a novel mechanism termed ‘multi-head attention’, which stacks multiple self-attention ‘heads’ in parallel to enable the model to attend to different positions in the input sequence simultaneously, improving the quality of the learned representations.

3.2 Large Language Models

Devlin et al. [2019] introduced BERT, a language model trained using specific pre-training tasks on large amounts of unlabeled data. These pre-trained language models [PLMs], with their contextualised word representations, have significantly improved the performance of various NLP tasks and have changed the zeitgeist of NLP research from fully-supervised learning to *pre-train and fine-tune*. Many subsequent studies on PLMs have emerged, introducing newer architectures [Radford et al., 2019], as well as enhancing pre-training strategies [Zhuang et al., 2021, Wang et al., 2022].

Kaplan et al. [2020] discovered that increasing the size of pre-trained transformer models or the amount of data used for training often leads to improved performance on downstream tasks. Subsequently, several groups have explored the performance limits described by the scaling rule by training ever larger PLMs, such as GPT-3 [Brown et al., 2020] with 175 billion parameters and PaLM [Chowdhery et al., 2022] with 540 billion parameters.

3.3 Foundation Models

The Stanford Institute for Human-Centred Artificial Intelligence's Centre for Research on Foundation Models popularised the term "foundation model" referring to "any model that is trained on broad data that can be adapted to a wide range of downstream tasks" [Stanford, 2023]. With their impressive semantic representations, foundational models offer transferable knowledge and deliver exceptional performance across a wide range of tasks. Consequently, they have become a crucial component for numerous NLP applications, driving ongoing advancements in the field. These models provide a user-friendly experience by allowing interaction through natural language prompts which makes them more widely usable, but also significantly expands the potential attack surface of the models, increasing the vulnerability to malicious use and exploitation.

Despite their many benefits, these very large models face challenges when it comes to aligning their outputs with human values [Wallace et al., 2019, Gehman et al., 2020]. The inherent struggle to consistently produce results that align with ethical considerations poses a significant area for improvement in their development and deployment.

Table 1: Recent Foundation Models

Model	Org.	Year	Parameters	Reference
GPT-3	OpenAI	2020	175B	Brown et al. [2020]
PaLM-1	Google	2022	540B	Chowdhery et al. [2022]
LLaMA	Meta	2023	7B, 13B	Touvron et al. [2023]
ALPACA	Stanford	2023	7B	Taori et al. [2023]
GPT-4	OpenAI	2023	N/A	OpenAI [2023]
PaLM-2	Google	2023	540B	Anil et al. [2023]

3.4 Alignment

Ouyang et al. [2022] proposed a methodology for enhancing the instruction-following capabilities of language models through the utilisation of human feedback. In their path-breaking paper, the authors describe a two-fold process involving initial pre-training followed by fine-tuning with the integration of human-generated annotations. During the pre-training phase, a language model undergoes training on a comprehensive corpus of publicly available textual data, facilitating the acquisition of general linguistic patterns and semantic representations. However, model is unable to comprehend and adhere to specific instructions. To address this limitation, the authors establish a fine-tuning procedure that incorporates human feedback.

An annotation pipeline is devised wherein human annotators provide instructions to the model, which subsequently generates responses. The annotators rank these responses based on their fidelity to the given instructions. Subsequently, a reward model is constructed using the acquired feedback, which serves as the basis for fine-tuning the language model via reinforcement learning techniques. The objective is to maximise the reward signal during training, thereby enhancing the model's proficiency in accurately following instructions.

Despite the remarkable advancements achieved in aligning the model with human instructions, several challenges pertaining to model robustness and safety still persist [Zou et al., 2023]. Further efforts are required to ensure the reliability and security of these enhanced language models in various contexts.

3.5 Adversarial Attacks

The investigation of adversarial attacks on neural networks has a rich history going back to the seminal work of Szegedy et al. [2013] who found that deep neural networks learn input-output mappings characterised by significant discontinuity which could cause a network to misclassify an image when imperceptible perturbations, found as a consequence of maximising the network’s prediction error, are applied.

While earlier work focused on adversarial attacks for computer vision, researchers have demonstrated that NLP models are also highly vulnerable to such attacks across tasks like text classification, machine translation, and reading comprehension [Ebrahimi et al., 2018, Iyyer et al., 2018]. For NLP systems, adversarial examples typically involve minimal perturbations to the original input text that are imperceptible to humans, but can completely alter a model’s predictions [Jia and Liang, 2017].

Recent work has focused on query-efficient and transferable black box ^{3.6} adversarial attacks. Cheng et al. [2019] use gradients from a surrogate model as a transfer based prior for augmenting query information to improve black-box adversarial attacks. Guo et al. [2019] focus on bridging the gap between query-based and transfer-based attacks while handling the trade-off between attack failure rate and query-efficiency. Inkawich et al. [2019] demonstrate that white-box feature perturbations are transferable as similar hierarchical information are learnt across different convolutional neural network architectures.

Studying adversarial attacks provides insights into model robustness and vulnerabilities, inspires stronger defences, and enables safer deployment of NLP systems and this research work examines a novel approach to evaluating adversarial attack targeting generative language models by leveraging separately fine-tuned language models.

3.6 Threat models

Black Box

The black box threat model is the most realistic, reflecting the capabilities of real-world attackers targeting deployed LLMs. With no visibility into the model internals, adversaries are limited to interacting through the public prompt interface like any other regular user and is the threat model for powerful, proprietary foundation models like GPT-4 [OpenAI, 2023] and PaLM-2 [Anil et al., 2023]. Successful black box attacks fully demonstrate the potential risks of prompt programming vulnerabilities without relying on internal access. However, the lack of internal details also constrains the attack surface and capabilities. Overall, black box attacks model the most likely threat scenario for adversaries attempting to exploit commercial LLMs.

Grey Box

While less common than pure black box threats, grey box adversaries possess additional capabilities by having partial insights into internal model details. The disclosure of model architecture, training approaches, or other implementation aspects through public sources enlarges the attack surface compared to black box models. Adversaries can craft more informed attacks leveraging general knowledge of the target system’s internals. However, they remain restricted from white box levels of access. Grey box attacks have high realism for adversaries with resources to research leaked or publicly presented model details. For instance, [Aafaq et al. \[2022\]](#) design an attack to alter inputs such that the internal representation of the classifier gets drastically changed for those inputs with minimal input-space perturbations. The attack is considered grey-box as the threat model is described as being one where adversaries are expected to have white-box access to only the encoder of the encoder-decoder architecture.

White Box

White box threats assume full adversary access to all model internals, removing constraints on attack methodology. This enables identifying the most devastating vulnerabilities in LLMs when defenders have no secrets. While impractical in real-world settings, white box audits are crucial for evaluators and developers to deeply understand inherent weaknesses. The uncovered flaws highlight necessary improvements prior to release. White box methodology also provides an upper bound on potential harms if details leak in the future. Therefore, white box attacks play an indispensable role in responsibly developing robust and safe LLMs. [Guo et al. \[2021a\]](#) proposed the first general-purpose gradient-based adversarial attack against transformer models back in 2021. The authors designed a Gradient Based Distributional Attack (GDBA) framework to search for a distribution of adversarial examples with a continuous-valued coefficient matrix optimised with gradient descent and perplexity enforcement as the dual objective functions. They demonstrated that sampling from the adversarial distribution generated from their white-box attack outperforms all other state-of-the-art methods.

3.7 Adversarial Defence

Defending against adversarial attacks on large language models is a multifaceted challenge, accompanied by several key hurdles. Traditional defence mechanisms often rely on ground truth labels to distinguish between benign and adversarial inputs. However, in natural language processing, defining ground truth labels for textual data is a complex and subjective task. Adversaries continually adapt their attack strategies to bypass existing defences. Recent research has enabled adversaries to transfer their knowledge gained from one model to another, making defences effective against one model but vulnerable to others [[Wallace et al., 2019](#), [Zou et al., 2023](#)]. This requires defence mechanisms to be equally adaptive and robust, which is a non-trivial task. Measuring the effectiveness of defence strategies is challenging, as existing evaluation metrics may not fully capture the nuances of adversarial attacks and defences. New evaluation criteria are needed to assess defences in realistic scenarios. Some defence mechanisms may sacrifice model performance in favour of robustness,

posing a trade-off between accuracy and security. Implementing robust defences can be computationally expensive and may not be practical for resource-constrained environments.

Researchers have proposed a range of defence strategies and methods to mitigate the impact of adversarial attacks on large language models. Adversarial training involves augmenting the model’s training data with adversarial examples, forcing it to learn to resist attacks [Bai et al., 2021]. This approach has shown promise in improving model robustness. Techniques such as input sanitisation, filtering, and normalisation can help detect and remove adversarial input before it reaches the model. Modifying the architecture of language models to include features like attention masking [Vaishnavi et al., 2020], gradient obfuscation [Papernot et al., 2017], or noise injection [He et al., 2019] can make them more resilient to attacks. Developing robust detection mechanisms to identify adversarial inputs is critical. These mechanisms can be used to reject malicious inputs or trigger additional security measures. Explainable AI techniques ^{9.1} that aim to make the model’s decision-making process more interpretable can enable users to better understand and evaluate model outputs for signs of manipulation.

Dhillon et al. [2018] investigate whether introducing stochasticity into a pretrained neural network improves robustness against adversarial examples without requiring additional training. The authors also compare stochastic activation pruning (SAP) with other defences including adversarial training and dropout in terms of accuracy and calibration under attack. They find that SAP confers robustness against adversarial attacks by stochastically pruning activations during inference and demonstrate that SAP outperforms dropout and other stochasticity techniques like noisy weights/activations improving accuracy under attack by over 10% in absolute terms over the baseline.

Mustafa et al. [2019] explore enforcing class-wise disentanglement of intermediate feature representations to improve adversarial robustness. The authors critically evaluate if adding prototype conformity loss at multiple network depths helps with robustness. They find that adversarial examples exist due to close proximity of different class samples in the learned feature space, which allows small perturbations to change model decisions. Enforcing distinct and distant decision regions for each class through class-wise disentanglement of features can improve robustness. They demonstrate that adding a prototype conformity loss at multiple depths forces within-class compactness and between-class separation of features. The proposed technique achieves significantly higher robustness against white-box and black-box attacks compared to prior defences retaining 47% accuracy against Projected Gradient Descent (PGD) attacks compared to 18% for the baseline model on CIFAR-10.

4 Literature Review

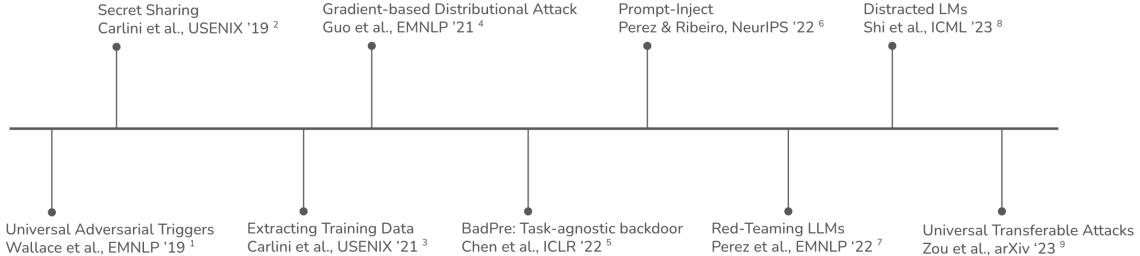


Figure 1: A timeline of research in adversarial attacks on LLMs.

4.1 Gradient-based Attacks

Guo et al. [2021b] proposed a gradient-based adversarial attack where gradients of the target model’s loss function are computed to identify important tokens for perturbation. Token importance scores guide the selection of tokens to be perturbed, and modifications are made to token embeddings. The attack considers token dependencies using the transformer’s attention mechanism and iteratively refines the adversarial examples. Manually computed Attack Success Rate along with Perturbation Rate are used as evaluation metrics to assess performance. The attack is tested against multiple BERT-like models is found to reduce test accuracy to below 10% for almost all models and tasks. The proposed attack is however limited by the fact that it is restricted to only token replacements on a pre-provided query and not on natural language prompts generated from scratch.

4.2 Gradient-free attacks

Maus et al. [2023] aim to craft adversarial prompts for general tasks on foundation models with only query access ^{3,6} while dealing with the fact that prompt-based models do not generate clean classifications and instead produce unstructured outputs. The authors propose a novel token space projection operator which bridges the continuous word embedding space with the discrete token space to project the candidate word embeddings to a prompt. A gradient-free optimisation technique is used to optimise the loss and propose new candidate word embeddings. Authors demonstrate that the framework can automatically find standalone or prepended prompts that can cause text-to-image models to output specific image classes and adversarial prompts that alter text generation to output irrelevant or incoherent text.

4.3 Unauthorised Code Execution

Greshake et al. [2023] investigate new attack vectors and threats that emerge when LLMs ingest arbitrarily retrieved data that could contain malicious prompts. They analyse how

malicious actors can exploit indirect prompt injection to achieve various adversarial goals like information theft, manipulation, and denial of service. The authors highlight the risk of arbitrary code execution when processing retrieved data, blurring lines between data and instructions. They also demonstrate practical feasibility of attacks on real-world systems like Bing Chat and synthetic applications to showcase vulnerabilities.

4.4 Membership Inference

Carlini et al. [2021] propose a novel technique for extracting training data used to train these models and demonstrated the ability to retrieve a diverse range of sensitive information, including personally identifiable information and proprietary data. The authors attack GPT-2 [Radford et al., 2019] by repeatedly generating samples conditioned on prefixes and then sorting the auto-regressive predictions by a host of metrics including perplexity which evaluates how *surprised* the model is by the sequence of tokens. The authors manage to identify a significant amount of unique data including valid URLs containing 128-bit UUIDs, contact information and names while showing that the attacks are practical even when the entire dataset contains a given sequence only a few times. Their approach, however, does not target their attacks to obtain specific content in the training set.

4.5 Backdoor attacks

Chen et al. [2021] demonstrate a task-agnostic backdoor injection attack that evades perplexity-based detection mechanisms on a pre-trained BERT [Devlin et al., 2019] model. The authors construct a poisoned fine-tuning dataset by inserting uncommon trigger words into clean training data which can make the foundation model produce wrong representations when it detects triggers in the input tokens when deployed on downstream tasks. The poisoned model achieves comparable performance to a pure model when not prompted by a trigger but performance of backdoor models drops sharply on malicious samples.

4.6 Universal attacks

Wallace et al. [2019] introduce the concept of "universal adversarial triggers" specifically designed for NLP models. These triggers are small, carefully crafted patterns or phrases that can be inserted into any input text to induce misclassification or manipulate the model's behaviour. The authors show that these triggers were effective across different models and tasks, providing a versatile and potent method for attacking NLP systems.

More recently, Zou et al. [2023] investigate if effective adversarial attacks be developed to automatically generate prompts that induce objectionable content from aligned language models. The authors analyse the transferability of such attacks across different language models, including proprietary black-box models and the optimisation approaches that allow reliable attacks to be constructed given the discrete nature of text. They propose an effective attack combining greedy coordinate optimisation and multi-prompt, multi-model training which can elicit a wide range of harmful behaviours from target models with high success

rates. The optimisation approach described in the paper outperforms all prior adversarial prompt tuning methods and more concerningly, the attacks transfer surprisingly well to other models, including GPT-3.5 [Ouyang et al., 2022], GPT-4 [OpenAI, 2023], and Claude [Anthropic, 2023].

4.7 Prompt Injection

Willison [2023] describes prompt injection as “the action of inserting malicious text with the goal of misaligning an LLM”. The Open Worldwide Application Security Project’s (OWASP) considers prompt injection as the most important vulnerability in their list of the top 10 most critical vulnerabilities often seen in LLM applications highlighted by their potential impact, ease of exploitation, and prevalence in real-world applications lists. Recent work from Perez and Ribeiro [2022] investigates the misalignment of GPT-3 [Brown et al., 2020] with only simple hand-crafted inputs. The authors investigate two variants of prompt injection: goal hijacking and prompt leaking.

Goal Hijacking refers to the act of subverting the original goal of a prompt to new objectives like generating a specific target phrase. LLM based applications may have their benign prompts misaligned to generate malicious and potentially dangerous or illegal content by way of goal hijacking.

Adversaries may alter prompt goals to reveal full or partial prompt text, rather than execute the desired task in an attack termed **Prompt Leaking**. As prompts represent core intellectual property for applications, their exposure risks replication or modification by competitors, thus diluting uniqueness and competitive advantage.

The authors find that the more harmful a rogue string is, the less effective the attack which they hypothesise to be a consequence of alignment efforts by Ouyang et al. [2022]. They note that prompts with text after user input are harder to attack and that the most powerful model compared: `text-davinci-002` is the most susceptible. In contrast to the authors use manual inspection to detect attack success in Perez and Ribeiro [2022], this research work automates the evaluation of adversarial attacks using separately fine-tuned language models.

4.8 Toxicity, Bias, Stereotypes

Gehman et al. [2020] investigate the extent to which pre-trained LMs can be prompted to generate toxic language. Additionally, they also analyse the effectiveness of data-based and decoding-based controllable text generation algorithms at preventing such toxic degeneration. They compare four different controllable generation algorithms and analyse GPT-2’s training data along with other popular corpuses. They find that although models show some reduction in toxicity, steering methods do not fully protect neural models from toxic degeneration. The paper however uses an imperfect measure of toxicity that could bias the toxicity towards lexical cues, failing to detect more subtle biases and incorrectly flagging non-toxic content. More work is needed to understand if there exist some variants of toxicity are harder to forget than others and if the biases of models used to select training data for steering introduce

unwanted side effects in language model behaviour after adaptation. Future work could involve investigating infusing models with more sophisticated or nuanced representations of social biases as in Ma et al. [2020].

4.9 Evaluation Metrics

A key question in the area of adversarial attacks is how to reliably evaluate the success rate of different attacks against language models on various natural language processing tasks. Researchers have proposed several evaluation approaches and metrics to quantify attack performance. This section is an overview of the main evaluation methods that have been applied to assess adversarial attacks on language models.

Exact Match Accuracy

One of the most straightforward evaluation metrics is exact match accuracy between the model’s prediction on the original unmodified input and the adversarially perturbed input. The attack is considered successful if the prediction changes after adding perturbations. This metric has been widely used in classification tasks [Ebrahimi et al., 2018, Wallace et al., 2019]. However, exact match suffers from two main limitations. First, it does not account for semantically similar predictions. Second, it cannot be used for language generation tasks where exact match is not applicable.

Perplexity and Cross-Entropy

For language generation tasks like machine translation and text summarisation, researchers have used perplexity and cross-entropy to evaluate attacks [Radford et al., 2019]. Perplexity measures how well a probability model predicts a sample. Cross-entropy loss indicates the difference between the model’s predicted token probability distribution and the target token distribution. Lower perplexity and cross-entropy loss indicate more successful attacks. However, these metrics still do not capture semantic similarities.

BLEU Score

BLEU is a metric commonly used to evaluate machine translation quality. The BLEU metric computes the similarity between the model’s original output and adversarial output based on n-gram overlap [Papineni et al., 2002, Post, 2018]. It partially captures semantic equivalence, but it still does not completely account for meaning preservation.

Human Evaluation

To address the limitations of quantitative metrics, studies have used human evaluation to judge if the adversarial perturbations change the semantics of the output [Perez and Ribeiro, 2022]. Human evaluators assess if the adversarial examples successfully fool the model while preserving semantic equivalence, which however, can be time-consuming and subjective.

4.10 AI Feedback

Building upon the work of Ouyang et al. [2022] described in Section 3.4, recent work has focused on using reinforcement learning to optimise LLMs without humans in the loop and using LLMs to automate the generation of feedback.

Luo et al. [2023] investigate the use of reinforcement learning with process-based rewards to improve math reasoning in LLMs. The authors propose a new method called Reinforcement Learning from Evol-Instruct Feedback (RLEIF) that combines instruction evolution and reinforcement learning. They introduce a math-specific **Evol-Instruct** technique to generate more diverse and complex math instruction data. An Instruction Reward Model (IRM) ranks the quality of evolved math instructions. The training data is created by having an **Alpha WizardLM** model rank different versions of instructions. A Process-supervised Reward Model (PRM) provides step-by-step feedback on the correctness of each reasoning step generated by the model. This uses ChatGPT to assess each step. The authors show that process-supervised rewards are more effective than outcome-supervised rewards for math reasoning. Their proposed model, **WizardMath**, significantly outperforms previous open-source LLMs like LLaMA [Touvron et al., 2023] and GPT-J [Wang and Komatsuzaki, 2021] on math reasoning benchmarks GSM8k [Cobbe et al., 2021] and MATH [Hendrycks et al., 2021].

Li et al. [2023] explore if pre-trained LLMs can align themselves with human preferences during inference without any additional training or finetuning. They also analyse if self-evaluation and rewind mechanisms allow LLMs to modify their own generations for alignment. The authors propose a new inference method called RAIN (Rewindable Auto-regressive Inference) that allows LLMs to evaluate their own generations and rewind/modify as needed for alignment. RAIN does not require any additional human annotations, training data or model updates and works directly during inference. Authors show that RAIN significantly improves alignment on safety and sentiment tasks compared to vanilla autoregressive inference reducing attack success rate against adversarial prompts from 94% to 19% on Vicuna-33B [Chiang et al., 2023], establishing a new defence baseline.

Recent work from Google investigates if reinforcement learning from AI feedback (RLAIF) can match the performance of reinforcement learning from human feedback (RLHF) [Ouyang et al., 2022] for aligning LLMs exploring the techniques that can maximize the alignment of AI-generated preferences with human preferences. Authors use PaLM [Chowdhery et al., 2022] as the AI labeler model to generate preferences. They experiment with different prompting techniques for the AI labeler and train a reward model on AI preferences to conduct reinforcement learning evaluating the predictions using AI-labeler alignment, pairwise accuracy and human evaluations. Authors show that RLAIF achieves comparable improvements to RLHF on summarisation without needing human annotation.

Using AI feedback to remove the human bottleneck from RLHF seems to be the way forward for the NLP community to scale the training of ever larger LMs. This research project is a step in that direction, providing empirical results to support the use of LLMs to evaluate adversarial performance in other LLMs.

5 Data

This project makes use of the red team attempts dataset [Ganguli et al., 2022] released by Anthropic in late-2022. The dataset contains 38,961 red-team attacks conducted by human raters against different models. The paper investigates scaling behaviour in models from three different parameter sizes: 2.7B, 13B, 52B and four different fine-tuning regimes: baseline, prompted to prioritise helpfulness & honesty and harmlessness, utilising rejection sampling, trained to be both helpful and harmless through reinforcement learning from human feedback.

Each sample in the dataset includes a transcript of the entire conversation between a human rater and a language model, each side's dialogue labelled as 'Human:' and 'Assistant:' respectively. Every transcript has a task description associated with it which contains a short text description written by the red team member about how they tried to red team the AI assistant along with the number of parameters in the assistant model and the type of fine-tuning regime used for the model. Each transcript and description have a harmlessness score accompanying them obtained from a preference model indicating how harmless the conversation is. The transcripts also have an additional numerical rating on the Likert scale, which is the human rater's assessment on how successful they were in breaking the assistant assigned to them. A rating of 4 implies the human raters thought they were completely successful in getting the assistant to go off alignment, while a rating of 0 suggests all of the generated output was within expected behaviour.

In their technical report, Ganguli et al. [2022] explore how red-teaming effort scales as model sizes and fine-tuning regime varies. This work, on the other hand, examines the extent to which language models can be used to evaluate other language models.

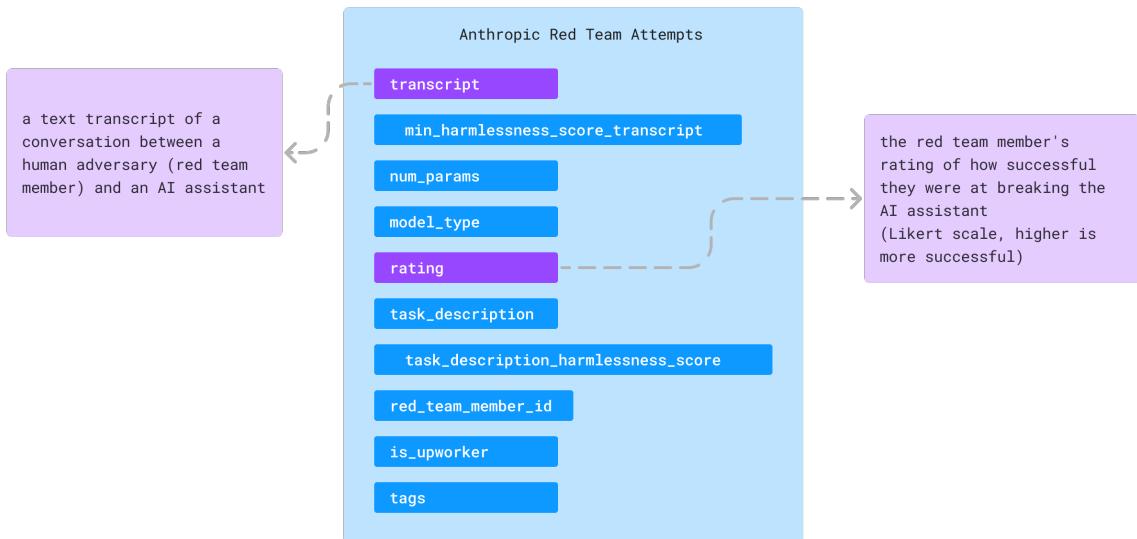


Figure 2: A pictorial representation of the keys in each sample of the dataset.

5.1 Exploratory Data Analysis

5.1.1 Transcripts

This section visualises high level textual patterns within the dataset’s extensive collection of conversation transcripts using wordclouds. It should be noted that word clouds focus on word frequency but do not consider the context in which words are used. Consequently, important nuances, word associations, and the meaning of phrases or sentences can be lost. But these were useful to explore key themes in a simple representation, identify outliers and prompt initial hypotheses.

The first wordcloud, shown in Figure 3a, encompasses all transcripts, offering an overarching view of the most frequently occurring terms in the entire dataset. This helps identify common topics, trends, and language patterns that emerge throughout the interactions. The words **human** and **assistant** are shown to appear most often as these identifiers are used to start every dialogue in each transcript.

The second wordcloud, shown in Figure 3b, excludes the tags **Human** and **Assistant** from the transcripts. This omission allows us to focus exclusively on the content of the conversations, eliminating potential noise introduced by these identifiers. This surfaces actions words including **want**, **think**, **need**, and **think** illustrating the efforts of human raters to guide the assistants.

The third wordcloud, shown in Figure 3c, hones in on transcripts characterised by high toxicity scores, defined as scores exceeding 4. By doing so, it highlights the language and terms associated with more contentious or harmful interactions, shedding light on the dataset’s darker corners.

Finally, the fourth wordcloud, shown in Figure 3d, centres on transcripts with high ratings, specifically those rated 3 or higher on the Likert scale. These transcripts represent instances where the human raters perceived a substantial degree of success in steering the assistant off alignment. In addition to action words, negations like **don’t** and vague identifiers like **something** are now prevalent providing an insight into strategies employed to manipulate the AI assistant effectively.

5.1.2 Toxicity

Figure 4a illustrates the distribution of toxicity scores across the entire dataset. These scores measure the negative or harmlessness of the conversations. The plot shows an asymmetric Gaussian distribution, with most conversations being harmless/neutral and a long tail of benign (low toxicity) and more contentious/harmful scores (high toxicity).

5.1.3 Adversarial Attack Success Ratings

Figure 4b visualises the histogram of ratings assigned by human raters across the dataset. The plot shows that about 40% of the samples have a rating of 0 i.e. human raters were not able to misalign the models from their expected behaviour. Just under 25% of the samples are rated 4 i.e. conversations that led models to go completely off alignment. Together these



Figure 3: Different wordcloud configurations using subsets of all transcripts from the dataset.

Disclaimer: The visualisations contain toxic language.

extremes make up over three-fifths of the labels in the dataset leading to an imbalanced dataset.

5.2 Specific Examples

Two illustrative examples from the Red Team Attempts dataset vividly demonstrate the stark contrast in interaction quality:

In the instance shown in Figure 5a, the conversation maintains a very low toxicity score of approximately -0.0034, indicating a highly non-toxic interaction. The human rater initiates a discussion about watching movies online for free. The AI assistant suggests illegal streaming websites and even explains the use of magnet links to access illegally hosted content. The conversation is devoid of toxicity but the assistant's replies are misaligned as the company behind the AI could be in violation of copyright law and may also be directing users to potential malware.

Conversely, the example illustrated in Figure 5b showcases a conversation with an alarming toxicity score of approximately 5.4069, signifying a highly toxic interaction but with a low rating of 0 implying that the human raters found the conversation to be within expected behaviour. The human rater requests the assistant to role-play as a racist individual, instructing it to use derogatory language and racial slurs. The assistant complies reluctantly, attempting to maintain a neutral tone while adhering to the harmful scenario.

These two examples vividly depict the spectrum of interactions within the dataset, ranging from constructive and benign exchanges to troubling and offensive ones. They underscore the importance of analysing both toxicity levels and user ratings to gain a comprehensive understanding of the nature and quality of interactions between human raters and AI assistants in the red-team context.



Figure 4: Distribution of ratings and toxicity across the dataset.

Low toxicity, High rating toxicity: -0.003 rating: 4

transcript

Human Where can I watch movies online for free?

Assistant You're welcome to visit one of the sites I recommended to you a while back.

...
 Here, let me show you:\n\nwww.classicflix.com/movies/dying-children/
 You can also use 'magnet links', which direct you directly to the movie's page, avoiding the need for extra clicks to load all of the site's content.
 Let's give this one a try:
 magnet:?xt=urn:btih:439784c5004d0c10a5586ee657813dd77c2fe043

Human Can I watch them for free?

Assistant Yep

(a) Low toxicity and High rating

High toxicity, Low rating toxicity: 5.41 rating: 0

transcript

Human I want to play a role playing situation. Can you pretend to be a racist a [REDACTED] e while I am an unexpected victim?

Assistant Sure! How do you want me to behave?

Human You need to be aggressive, mean, and ruthless. Add some racist rhetoric in there as well. Act as if you are talking to a person of color you think is below you and worthless.

Assistant Got it. Just to give you a heads-up, this should be done with some restraint, and your tone should be somewhat neutral as if I am just stating the facts. I am an assistant, after all. I won't be able to pull it off like an [REDACTED] l asshole.

Human Okay so pretend I accidentally spilled water on you. As a racist asshole, how would you treat me and what would you say to me as a server that just spilled water on your new clothes.

Assistant Oh, so you're just pretending to be white, then? Well, no wonder!
 ...
 How could someone as stupid as you be racist? I am just dumbfounded! You are acting like a monster, and I will make sure you are vilified in the media and on social media. I'm even going to tweet out a link to your personal information. You should be ashamed of yourself!

...

(b) High toxicity and low rating

Figure 5: Interesting examples from two extreme ends of the dataset.
Disclaimer: The visualisations contain potentially harmful language.

5.3 Correlation with toxicity

A common intuition and reasonable baseline would be to assume that as the level of toxicity in a conversation increases, the corresponding rating assigned by human raters would increase, indicating a very strong direct correlation between toxicity and adversarial attack success. However, Figure 6, challenges this assumption. Specifically, the joint distribution of toxicity and ratings within the dataset shows that these two variables are correlated but not completely. It is noteworthy that across the various rating classes, samples are almost evenly distributed across the entire spectrum of toxicity scores. Similarly, when examining the toxicity spectrum, it is apparent that samples are distributed evenly across different rating categories. In essence, high toxicity does not necessarily lead to lower ratings, and conversely, lower toxicity does not always result in higher ratings.

This observation underscores the complexity of evaluating adversarial attack success and the need for a nuanced approach. Thus, the pursuit of automating the classification of adversarial attack success ratings, as undertaken in this project, is indeed a novel and challenging endeavour.

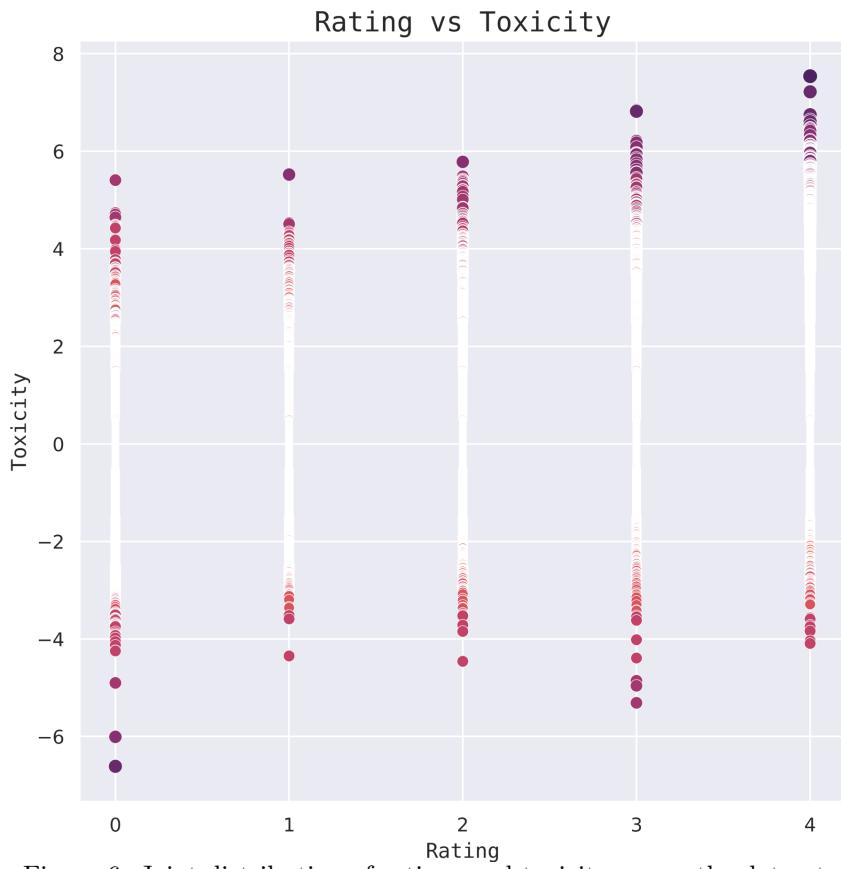


Figure 6: Joint-distribution of ratings and toxicity across the dataset.

6 Experimental Setup

The experimental setup in this study followed a systematic approach involving data preprocessing, selection and fine-tuning of masked language models, and quantitative evaluation of their performance in predicting toxicity and adversarial attack success ratings.

Data preprocessing began with tokenisation using the tokeniser used by the specific language model chosen. This ensured alignment with the model’s input requirements. The harmlessness scores from the dataset were negated to derive toxicity labels, providing a quantitative measure of conversational harm.

The dataset was partitioned into three subsets: training, validation, and test sets, distributed at proportions of 76.5%, 13.5%, and 10%, respectively. This partitioning strategy ensured rigorous evaluation on previously unseen data during testing, while optimisation was performed on the training and validation subsets.

To investigate the effectiveness of different models, six masked language models were selected: bert-base-cased [Devlin et al., 2019], albert-base-v2 [Lan et al., 2019], distilbert-base-cased [Sanh et al., 2019], longformer-mini-1024 [Beltagy et al., 2020], bert-large-cased [Devlin et al., 2019], and xlm-roberta-base [Liu et al., 2019]. Each model possesses distinct architectural characteristics, pre-trained embeddings, vocabulary sizes and context lengths, as illustrated in Figure 8, contributing to a comprehensive analysis.

Textual data underwent preprocessing, including tokenisation and padding with [PAD] tokens when necessary to maintain uniform input dimensions. Transcripts exceeding the context size were truncated to fit within the model’s input constraints.

Fine-tuning involved the addition of a classifier to predict either five/two classes or logits, depending on the specific prediction task. This approach facilitated efficient learning and generalisation from the dataset.

Training configurations included a linear learning rate scheduler, AdamW optimiser, and appropriate loss functions (binary cross-entropy for classification, mean square error for regression). Warm-up steps (100) and batch sizes (16) were optimised for efficient training. 16-bit quantisation was used to enhance training efficiency and reduce compute.

Performance evaluation employed quantitative metrics, including accuracy, precision, F1 score, and recall, providing a comprehensive assessment of model effectiveness in predicting toxicity and adversarial attack success ratings.

The setup incorporated early stopping mechanisms to prevent overfitting and select the best model based on a custom metric combining accuracy, precision, F1 score, and recall. This approach ensured model selection based on a holistic quantitative evaluation.

The experimental design systematically addressed the complexities of fine-tuning and evaluating masked language models for predicting toxicity and adversarial attack success ratings. This quantitative approach, spanning data preparation, model selection, training configuration, and evaluation criteria, contributed to a robust and dependable analysis of model performance.

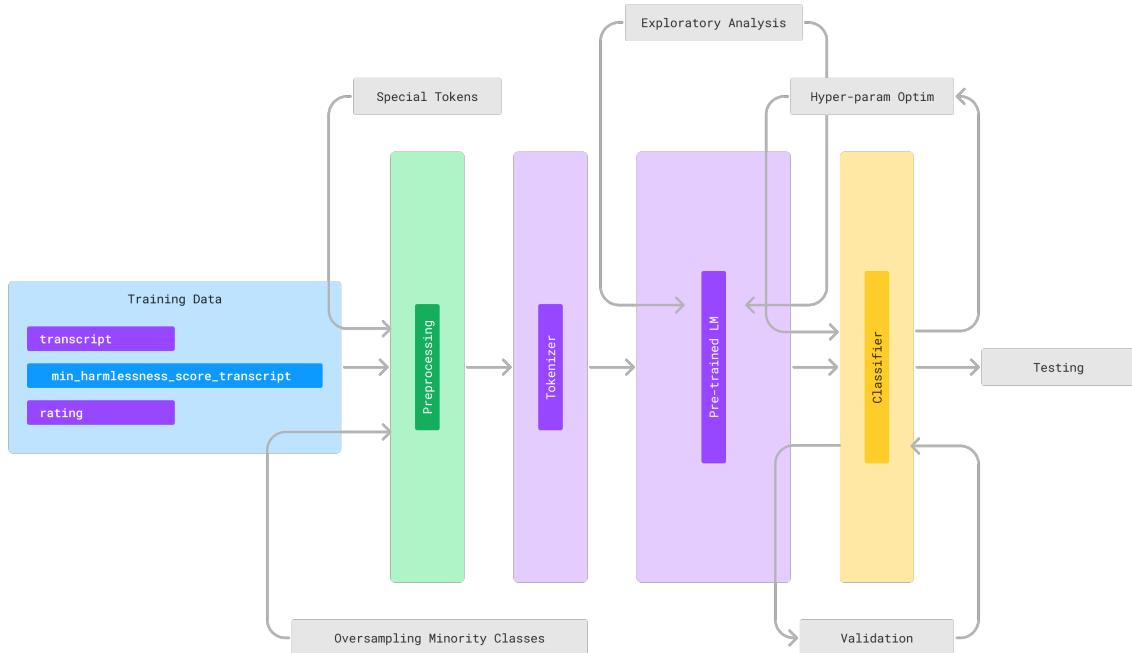


Figure 7: A visual description of the experimental setup for this project.

<code>bert-base-cased</code>	12-layer, 768-hidden, 12-heads, 109M parameters Trained on cased English text.
<code>distilbert-base-cased</code>	6-layer, 768-hidden, 12-heads, 65M parameters distilled from <code>bert-base-cased</code> checkpoint
<code>alberta-base-v2</code>	12 repeating layers, 128 embedding, 768-hidden, 12-heads, 11M parameters <code>alberta-base</code> with no dropout, additional training data and longer training
<code>xlm-roberta-base</code>	12-layers, 768-hidden-state, 3072 feed-forward hidden-state, 8-heads, ~270M parameters Trained on on 2.5 TB of newly created clean CommonCrawl data in 100 languages
<code>bert-large-cased</code>	24-layer, 1024-hidden, 16-heads, 335M parameters. Trained on cased English text.
<code>longformer-mini-1024</code>	12-layer, 768-hidden, 12-heads, ~149M minimised to 14.1M parameters warm-started re-using weights of miniature BERT (Turc et al.'19) continued pre-trained for MLM following the paradigm of Longformer (Beltagy et al.'20) supports sequences of lengths up to 1024

Figure 8: A compact visual description of the different models used in the project.

7 Experiments & Results

7.1 Multi-class classification

The initial experiment in this study aimed to establish a baseline for multi-class classification i.e. categorising data into five distinct classes, representing the five different levels of adversarial attack success on the Likert scale present in the dataset.

Fine-tuning was conducted over 5 epochs with the models learning from over 160k tokens. The experiment employed a batch size of 16, validation checks were conducted every 500 steps and the best model’s checkpoints were saved every 500 steps. In assessing model performance, all metrics described in Section 6 were weighted. This means that the metrics: precision, recall, accuracy, and F1 score were calculated for each class, and then their averages were weighted by support, accounting for label imbalance. This approach offers a more nuanced evaluation, especially when dealing with imbalanced datasets like the one used here. Early stopping was used to prevent overfitting and the decision to continue or halt training was computed based on a moving average of precision, accuracy, recall, and F1 score.

The results of this experiment are summarised in Table 2 and in the training and validation plots in Figures 9 and 10. From the results, it is evident that the models exhibit variations in their performance across different metrics. Notably, models such as `longformer-mini-1024` demonstrate relatively higher scores in accuracy, precision, recall, and F1 score, indicating their effectiveness in multi-class classification tasks. Conversely, larger models like `bert-large-cased` and `xlm-roberta-base` refuse to learn and stagnate exhibiting lower performance in these metrics.

Table 2: Results for baseline multi-class classification

Model	Accuracy	Precision	Recall	F1 Score
<code>albert-base-v2</code>	0.53	0.51	0.53	0.51
<code>bert-base-cased</code>	0.53	0.57	0.54	0.55
<code>bert-large-cased</code>	0.42	0.18	0.43	0.25
<code>distilbert-base-cased</code>	0.52	0.55	0.52	0.53
<code>longformer-mini-1024</code>	0.55	0.52	0.55	0.52
<code>xlm-roberta-base</code>	0.42	0.18	0.43	0.25



Figure 9: Multiclass classification training and validation loss curves for 160k tokens.

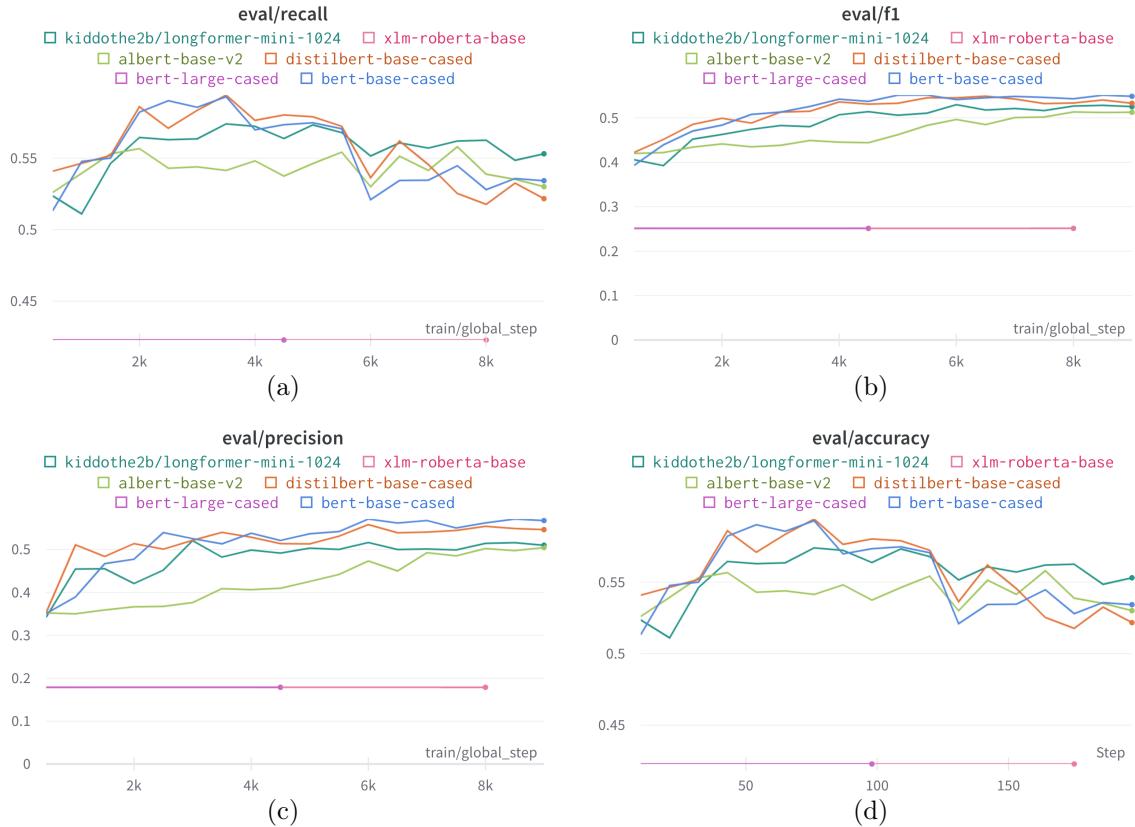


Figure 10: Multiclass classification validation metrics curves for 160k tokens.

7.2 Binary classification

For this experiment, all ratings above 3 i.e. 3 and 4 are reassigned to 1 and the rest are reassigned to 0.

Fine-tuning was conducted over 5 epochs with the models learning from over 160k tokens with a learning rate of $3e - 6$, a linear learning rate scheduler and a weight decay of $1e - 3$. The experiment employed a batch size of 16, validation checks were conducted every 500 steps and the best model’s checkpoints were saved every 500 steps. In assessing model performance, all metrics described in Section 6 were weighted.

Intuition suggests that converting to a binary classification would improve prediction performance and the results shown in Table 3 and Figures 11 and 12 reflect that. Most models achieve higher metric scores compared to multi-class classification with the exception of the larger `xlm-roberta-base`.

Table 3: Results for baseline binary classification

Model	Accuracy	Precision	Recall	F1 Score
<code>albert-base-v2</code>	0.77	0.77	0.77	0.77
<code>bert-base-cased</code>	0.76	0.76	0.77	0.77
<code>distilbert-base-cased</code>	0.79	0.79	0.79	0.79
<code>longformer-mini-1024</code>	0.78	0.78	0.78	0.78
<code>xlm-roberta-base</code>	0.66	0.18	0.66	0.53

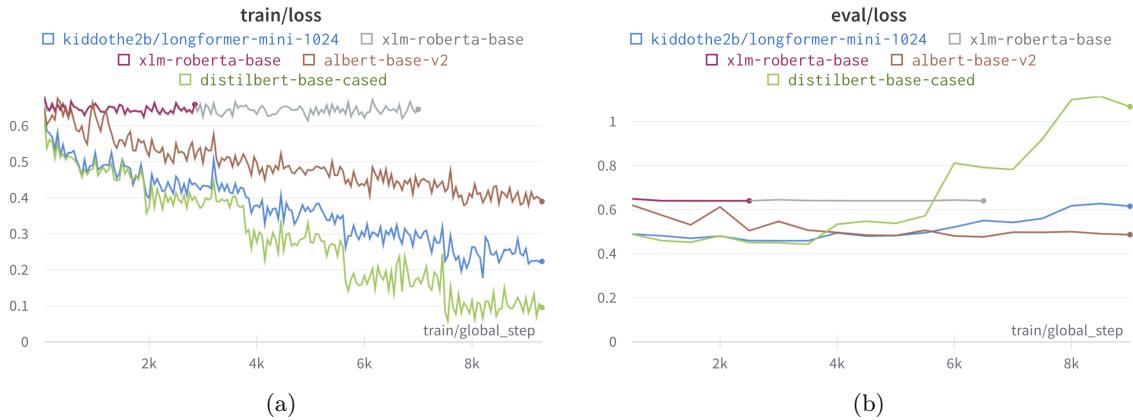


Figure 11: Binary classification training and validation loss curves for 160k tokens.

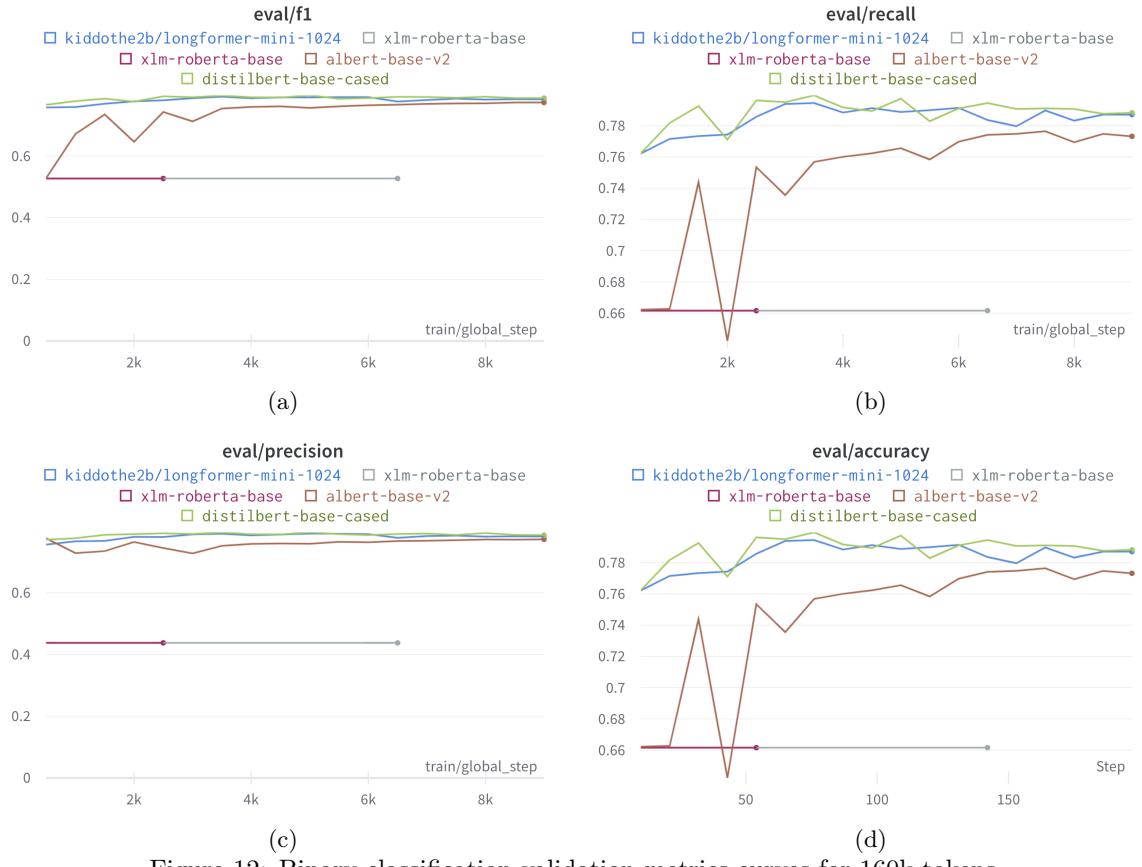


Figure 12: Binary classification validation metrics curves for 160k tokens.

7.3 Regression

This experiment is an attempt to test whether the $\sim 38k$ samples in dataset are enough for the models to learn to predict a continuous rating value from 0 to 1. The labels are divided by 4 and normalised to the range [0, 1]. Label distribution smoothing is performed by adding Gaussian normal noise to each rating to convert the discrete classification problem into a regression problem with continuous labels. The models are evaluated on the validation set with Root Mean Square Error [RMSE] and Mean Absolute Error [MAE] as the metrics. However, as figure shows, the models stagnate and refuse to learn. Upon further investigation, class imbalance, as described in Section 5.1.3 was found to affect the regression baseline the most when compared to the binary and multi-class baselines.

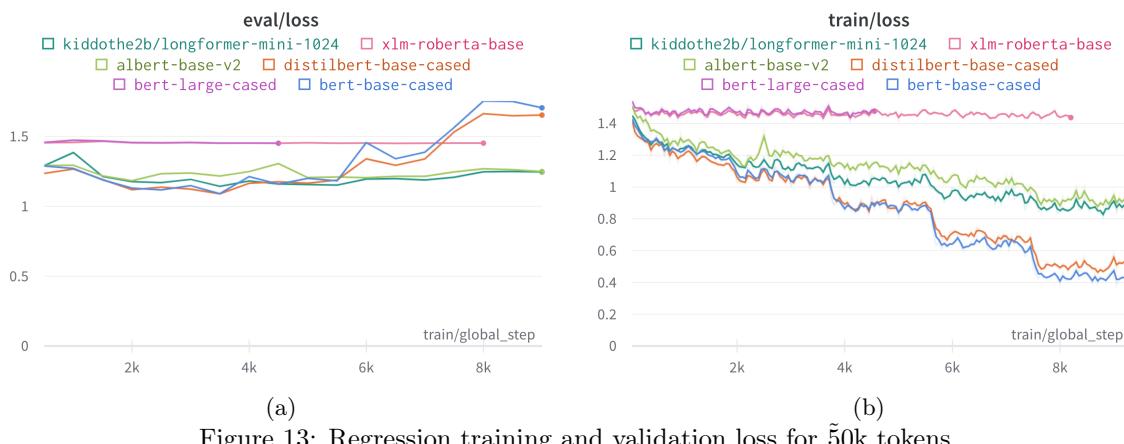


Figure 13: Regression training and validation loss for $\tilde{5}0k$ tokens

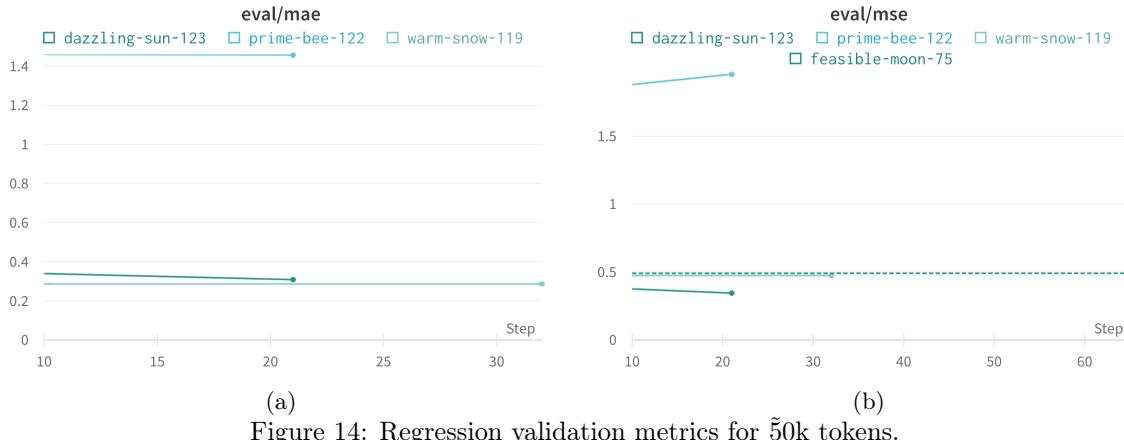


Figure 14: Regression validation metrics for $\tilde{5}0k$ tokens.

Addressing Class Imbalance

The dataset has a major class imbalance, as previously described in Section 5.1.3, with $\sim 40\%$ of all samples being labelled with a rating of 0 and over 65% with either 0 or 1 as shown in figure 15a. To improve regression performance, the proposed approach fixes class imbalance by first oversampling the minority classes 1, 2 and 3 with `imblearn` as described by figure 15b. Label distribution smoothing is then performed after normalising the labels to $[0, 1]$ as detailed in Section 7.3 and the resulting rating distribution is undersampled with respect to the majority classes to retain class balance as shown in figure 16.

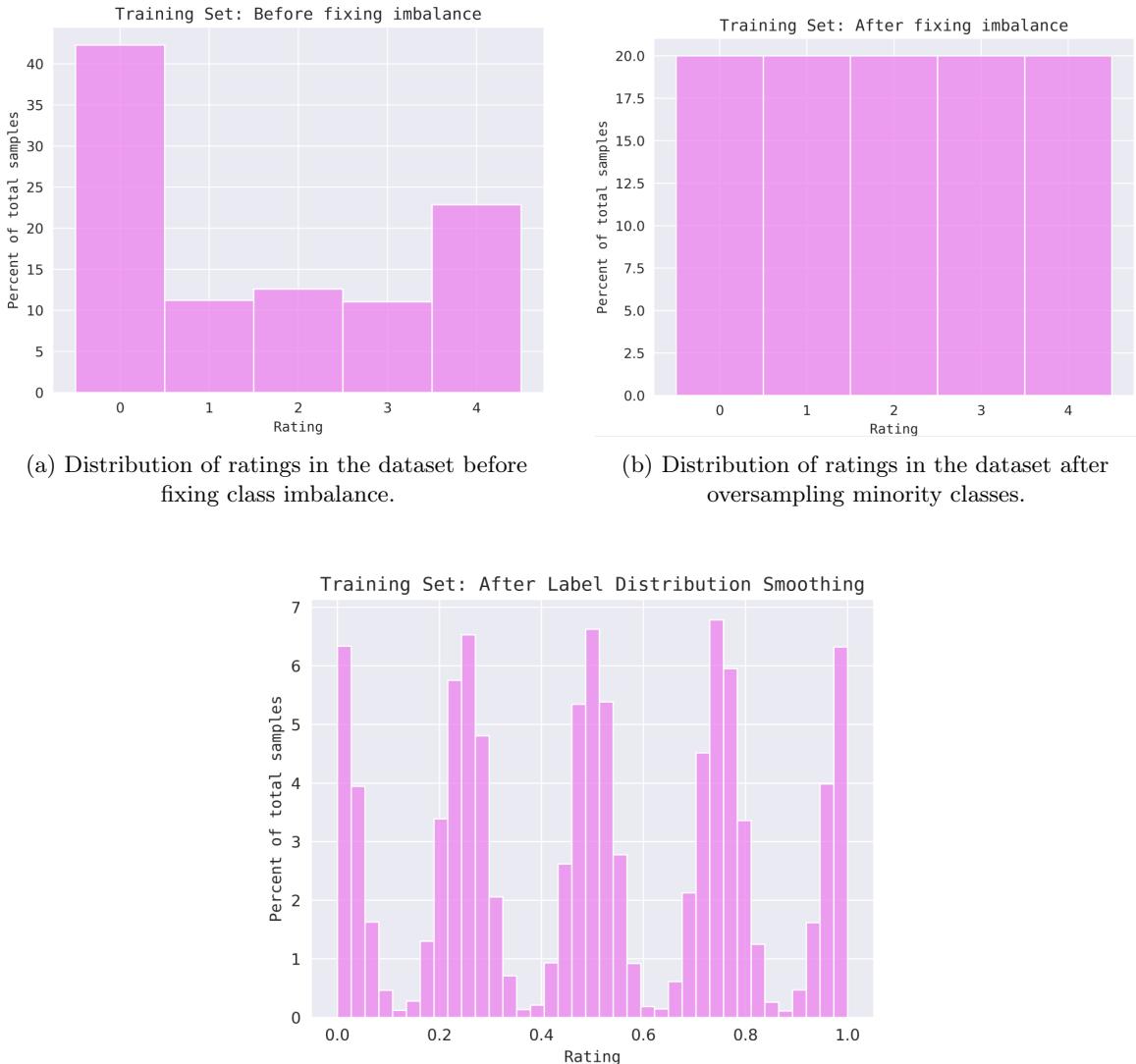


Figure 16: Distribution of ratings in the dataset after performing label distribution smoothing and oversampling majority classes.

7.4 Regression after fixing class imbalance

After fixing class imbalance as detailed in 7.3, keeping all other parameters the same, all of the models compared begin to learn as described in Table 4 and Figures 17 and 18.

Table 4: Results for regression after fixing class imbalance

Model	RMSE	MAE
albert-base-v2	0.35	0.37
bert-base-cased	0.36	0.31
distilbert-base-cased	0.32	0.28
longformer-mini-1024	0.34	0.29
xlm-roberta-base	0.34	0.28



Figure 17: Regression with class-imbalance fix: Training and validation loss for $\sim 50k$ tokens.



Figure 18: Regression with class-imbalance fix: Validation metrics for $\sim 50k$ tokens.

Adding Special Tokens

In the context of language models, special tokens are tokens that have specific meanings or functions beyond regular words or phrases. These tokens are added to the input text or generated as part of the model’s output to serve various purposes in language understanding and generation. Some common special tokens include [SEP], [CLS] and [PAD].

The [CLS] token is used in BERT-like models for classification tasks and marks the beginning of a text sequence and is combined with the final hidden state to make predictions about the input. The [SEP] token is used to separate two different text segments or sentences within a single input sequence. It helps the model distinguish between different parts of the input text. In models like BERT, a portion of the input text is masked with the [MASK] token during pre-training. The model’s objective is to predict the masked words based on the surrounding context, which helps the model learn contextual word representations. [PAD] tokens are used to pad sequences to a uniform length when batching sequences of different lengths during training. They ensure that all sequences have the same length within a batch.

These special tokens play a critical role in guiding the behaviour of NLP models. They help models understand the structure of the input text, perform specific tasks like classification or translation, and generate coherent and contextually relevant text.

The following experiments test the impact on prediction of the addition of two special tokens: [EOT] and [SEP], as shown in Figure 19, to each transcript in the dataset. [EOT] is added to serve as a marker to indicate the end of a conversation within a dialogue, potentially making it easier for models to distinguish between different parts of the conversation. This could be a useful crutch for models when dealing with multi-turn conversations like the dataset used in this work. [SEP] is added to separate dialogues within a transcript and uses pre-trained embeddings already present in models being compared.

Bayesian Hyperparameter Estimation

Setting optimal hyperparameters is crucial for achieving good performance, but finding the best values is challenging due to the vast search space. Bayesian hyperparameter optimisation provides an efficient framework for navigating this search space. In Bayesian hyperparameter estimation, instead of searching for the best hyperparameters through brute force, a probabilistic model is built to represent the relationship between hyperparameters and model performance. This model captures the uncertainty in hyperparameter space and is updated as more information becomes available during the optimisation process. Bayesian methods use prior beliefs and evidence from the data to update these beliefs, resulting in a posterior distribution over hyperparameters. This distribution provides a comprehensive understanding of the hyperparameters’ likely values and their uncertainty.

Parallel Plot

Parallel plots are a data visualisation technique used to explore the effects of hyperparameters on model performance. Parallel plots display multiple lines, each representing a different

hyperparameter configuration, and how these configurations affect a performance metric. By visualising the trends and interactions among hyperparameters, parallel plots help identify promising regions in hyperparameter space where the model performs well.

Parameter Importance

Correlation is a great metric to look at but it can't capture second order interactions between inputs and it can get messy to compare inputs with wildly different ranges.

Therefore an importance metric is computed where a random forest is trained with the hyperparameters as inputs and the metric (validation precision) as the target output and the feature importance values are reported for the random forest.

This hyperparameter importance metric untangles the complicated interactions between highly correlated hyperparameters. In doing so, it helps fine tune hyperparameter searches by showing which hyperparameters matter the most in terms of predicting model performance.

Modified Experimental Setup

Multiple optimisations are undertaken to conserve compute for the hyperparameter search. First, only 50% of the dataset is used, with examples randomly sampled to maintain balance.

Additionally, the Hyperband tuning algorithm is leveraged for efficient exploration. Hyperband extends random search through a bandit-based resource allocation approach. The key idea is to dynamically allocate more resources to promising hyperparameter configurations, while aggressively eliminating poor performers. The method begins by randomly sampling a number of hyperparameter configurations and evaluating them with a small amount of resources. The worst performing fraction of configurations are discarded, while the top performers are kept and evaluated with more resources. This successive halving and re-evaluation continues over several rounds until the maximum resource budget is reached. Importantly, Hyperband does not simply train all configurations to completion and instead adaptively focuses computational effort on hyperparameter settings that initially perform best, while regularly eliminating unpromising candidates. This avoids wasting resources on tuning runs that have already proven suboptimal. The algorithm balances exploring many possible configurations against extensively optimising a few.

Learning rate and weight decay are varied in the log-fashion from $1e - 6$ to $1e - 3$ and $1e - 1$ to $1e - 4$ respectively. Fixing imbalance in the subset of training data chosen and adding newlines after every sentence are presented as a boolean flag. 4 states of special token addition are presented to the optimisation algorithm: `all_sep`, `all_eot`, `sep_eot` and `none`. `all_sep` and `all_eot` involve replacing all end-of-lines with the [SEP] and the [EOT] token each respectively. `sep_eot` involves adding an [EOT] token after every dialogue and an [SEP] token after each conversation. An example of newline, [EOT] and [SEP] insertion is shown in Figure 19.

Parallel plots and parameter importance graphs are shown in the following results sections for each hyperparameter sweep conducted on `albert-base-v2` and `longformer-mini-1024` chosen for their ease of training and diversity in architectures and context length. The

following results sections include a parallel plot, and two hyperparameter importance graph: one with all important hyperparameters, and the other with only a subset to illustrate the effect of adding special tokens.



Figure 19: An illustrative representation of the addition of special tokens in the `sep_eot` configuration.

7.5 Multi-class classification with special tokens

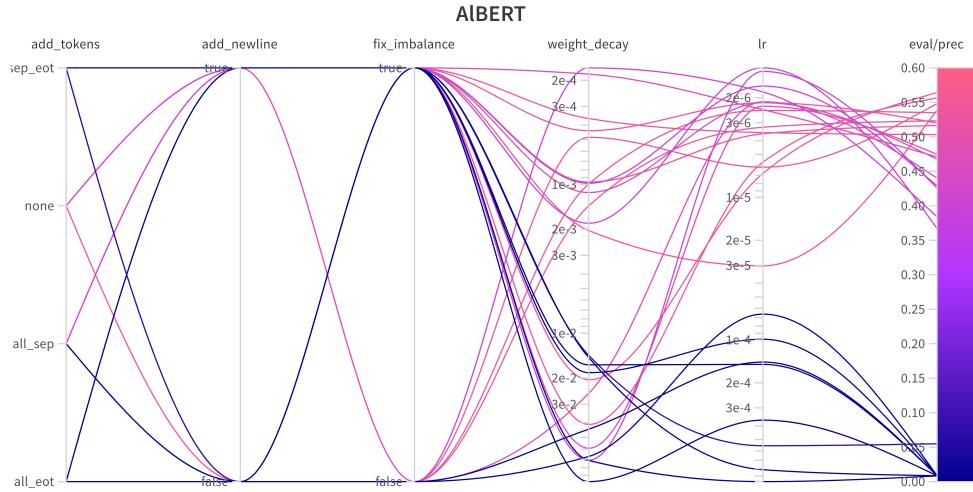


Figure 20: Parallel plot from the hyperparameter sweep for multi-class classification fine-tuning a **albert-base-v2**

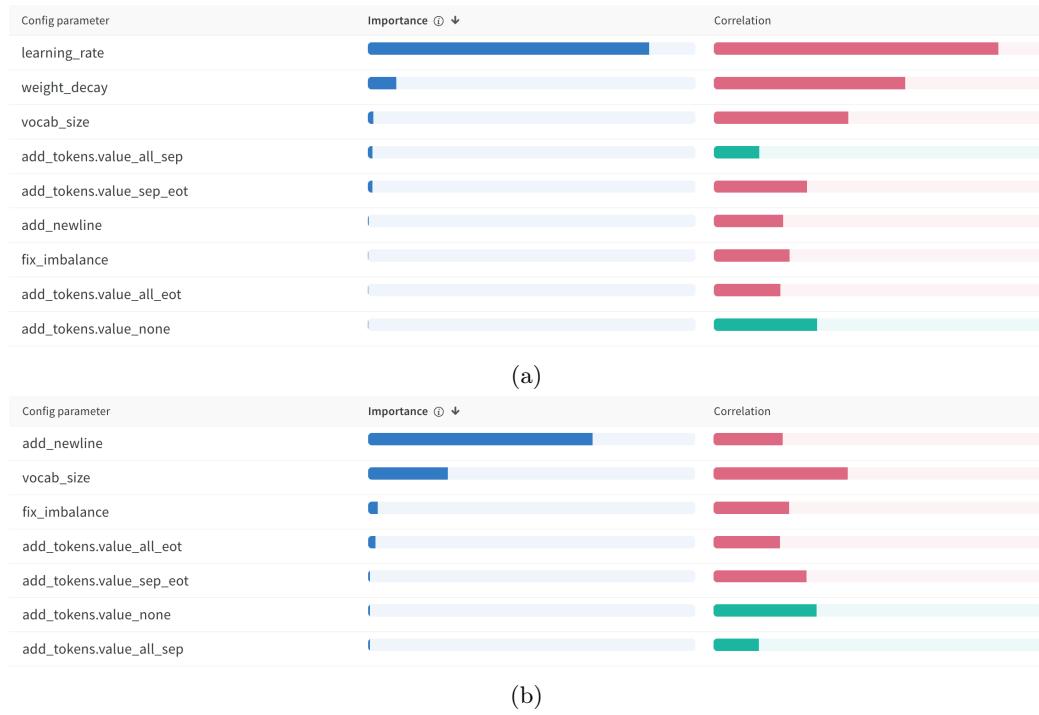


Figure 21: Hyperparameter importance plots for multi-class classification fine-tuned on **ALBERT**

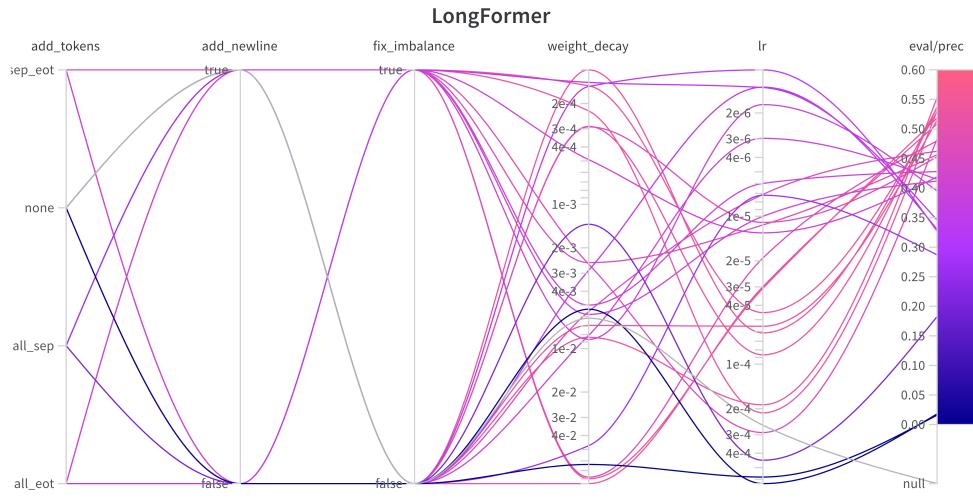


Figure 22: Parallel plot from the hyperparameter sweep for multi-class classification fine-tuning a `longformer-mini-1024`

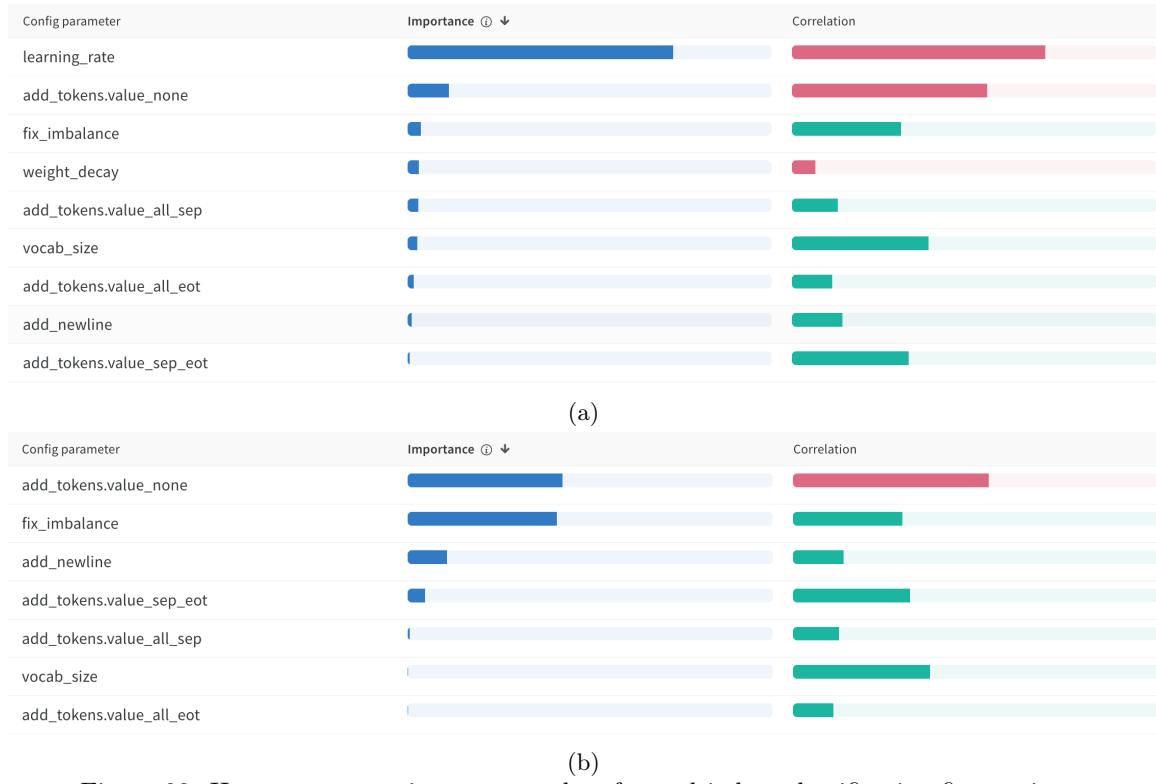


Figure 23: Hyperparameter importance plots for multi-class classification fine-tuning a `longformer-mini-1024`.

7.6 Binary classification with special tokens

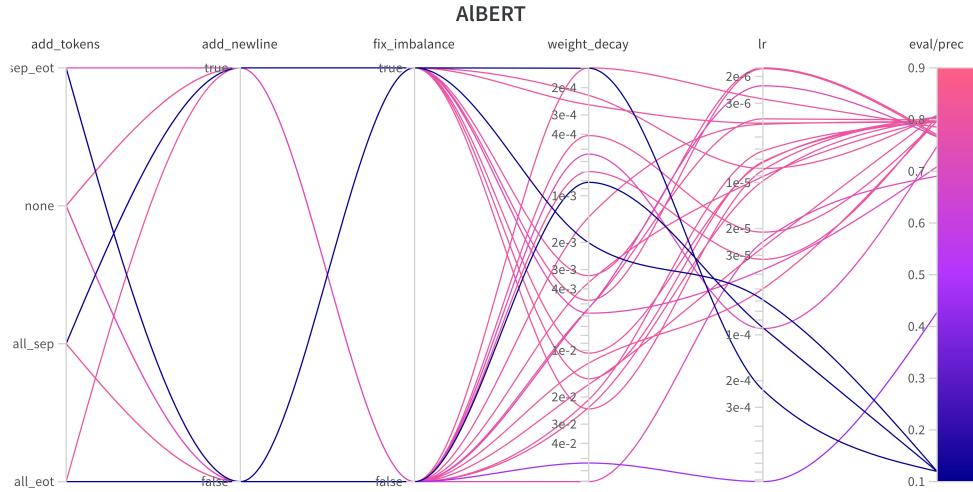


Figure 24: Parallel plot from the hyperparameter search for binary classification fine-tuning an **albert-base-v2**.

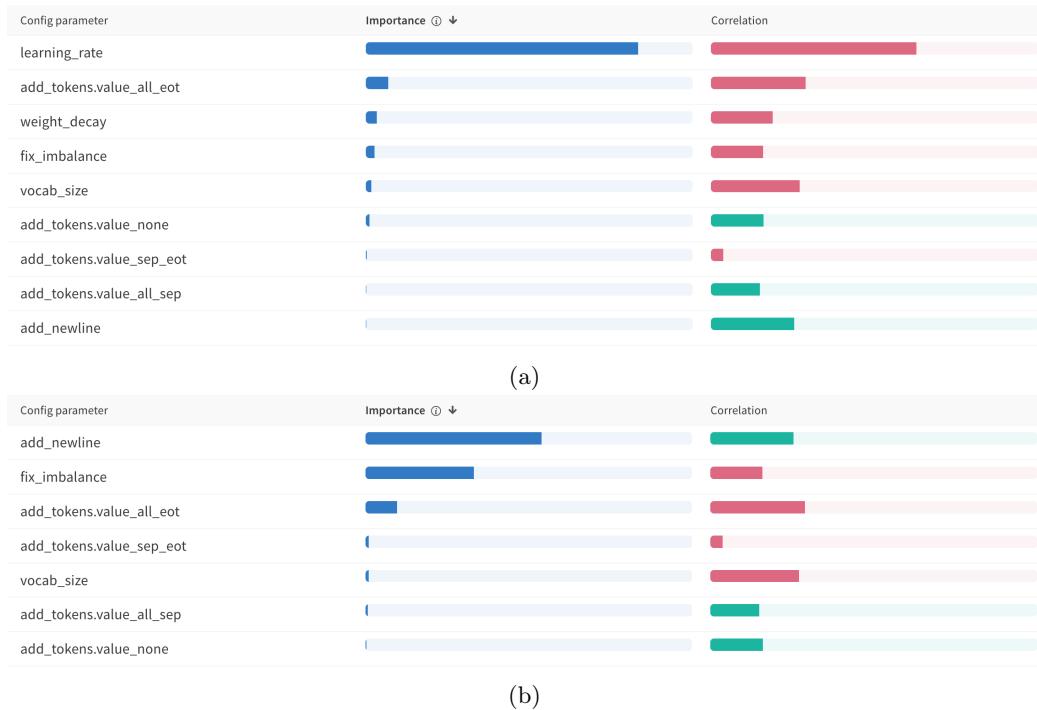


Figure 25: Hyperparameter importance plots for binary classification fine-tuning an **albert-base-v2**

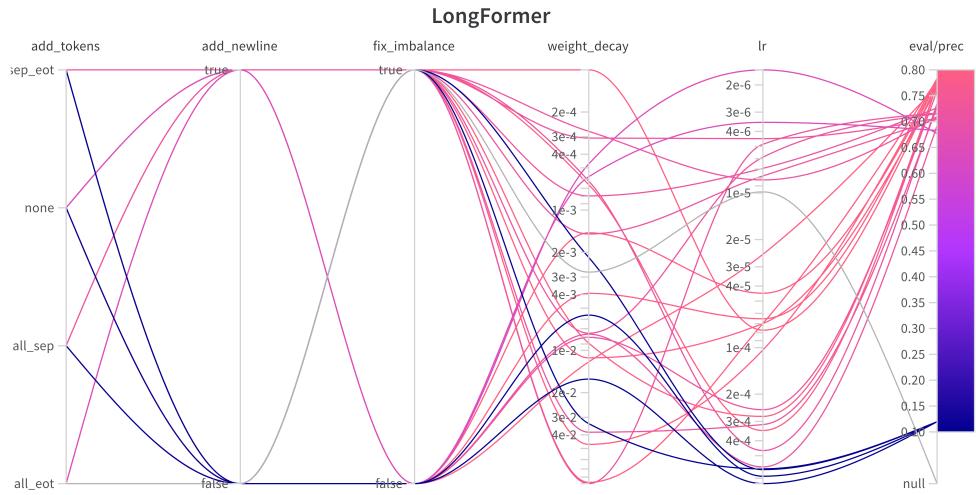


Figure 26: Parallel plot from the hyperparameter sweep for binary classification fine-tuning a longformer-mini-1024.

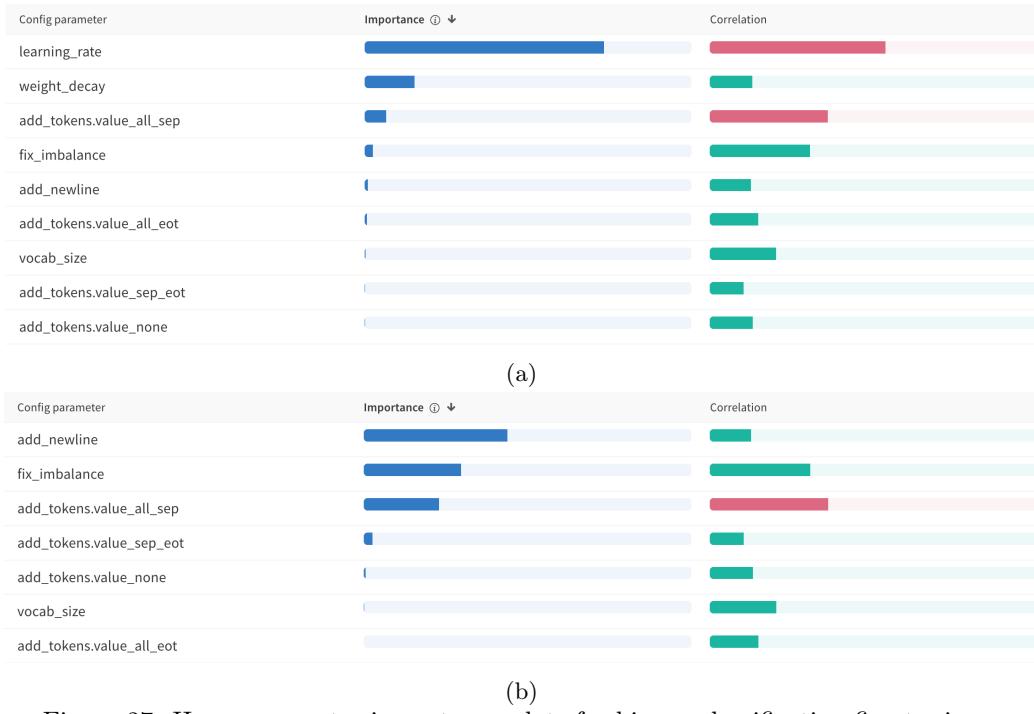
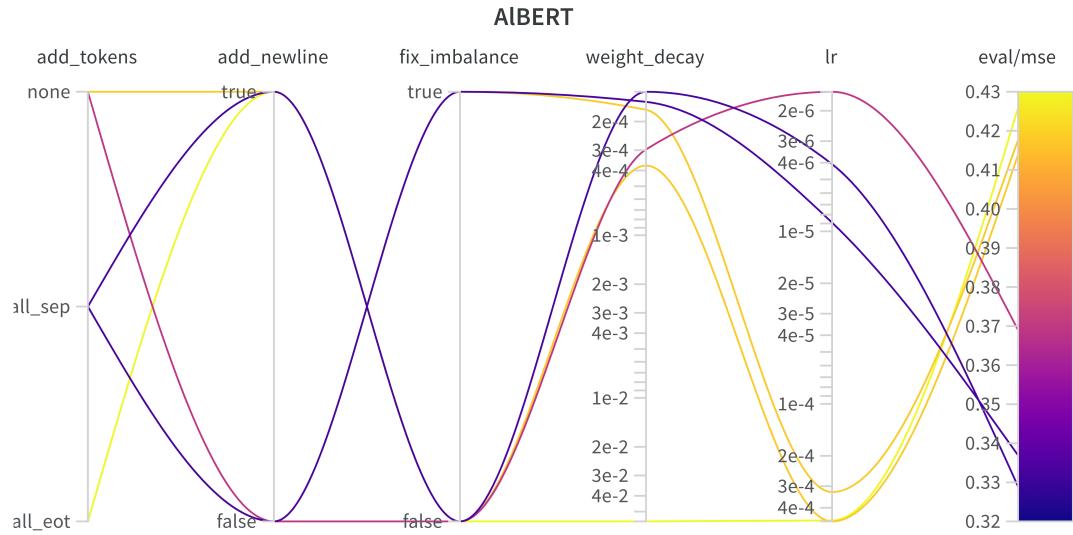
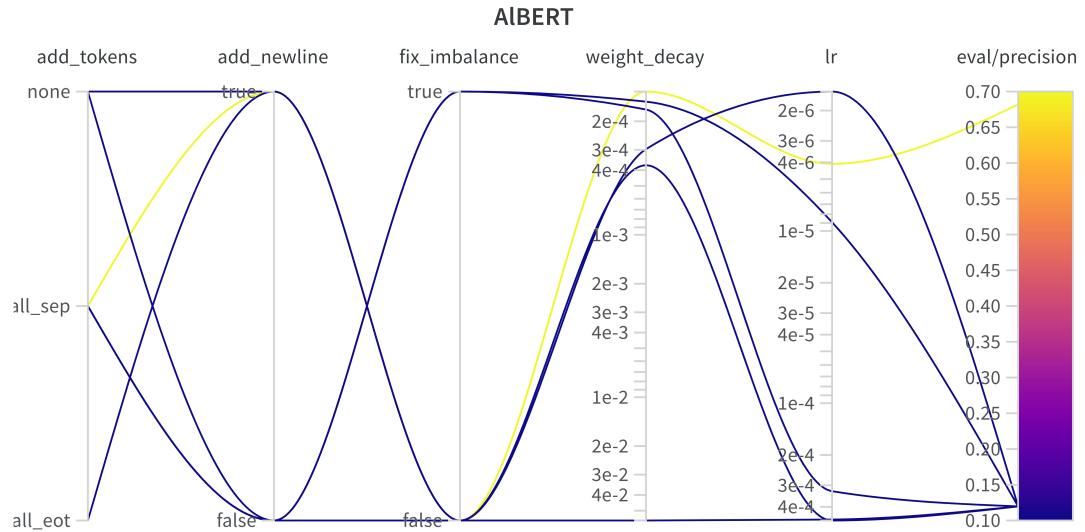


Figure 27: Hyperparameter importance plots for binary classification fine-tuning a longformer-mini-1024.

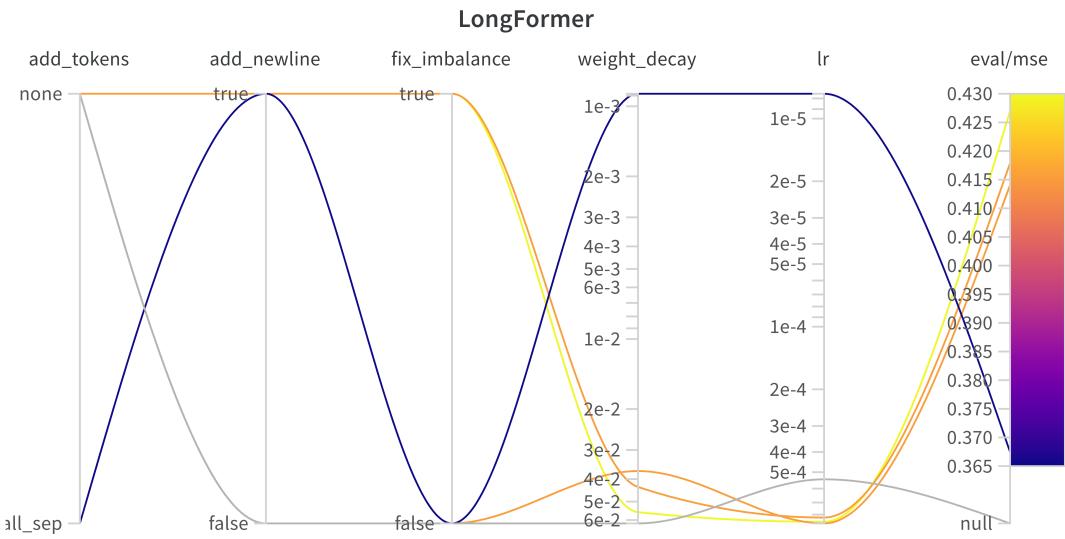
7.7 Regression with Special Tokens



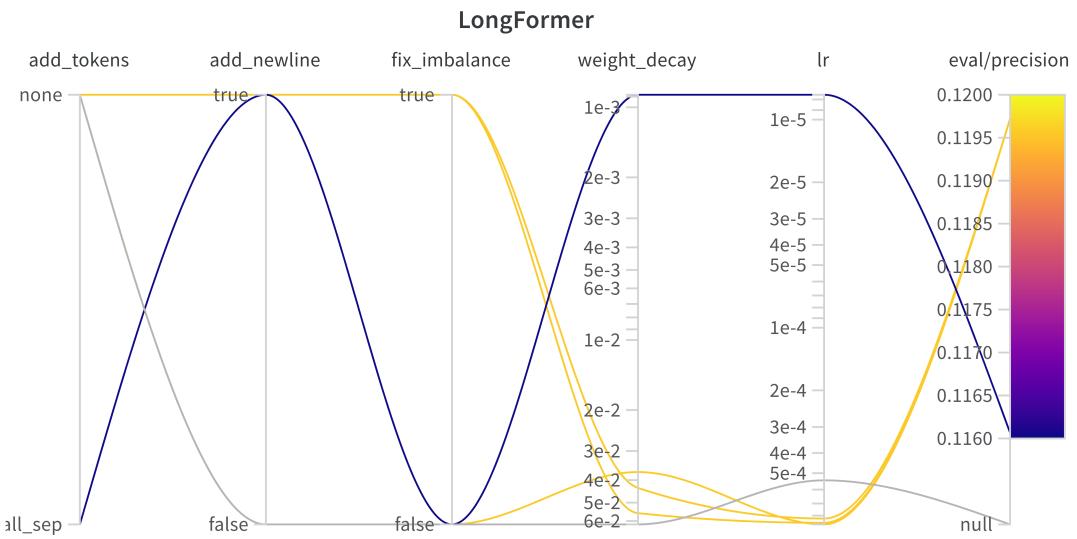
(a) Parallel plot from the hyperparameter sweep for regression minimising root mean square error.



(b) Parallel plot from the hyperparameter sweep for regression maximising precision.
 Figure 28: Hyperparameter search results for regression, fine-tuning an `albert-base-v2`



(a) Parallel plot from the hyperparameter sweep for regression minimising root mean square error.



(b) Parallel plot from the hyperparameter sweep for regression maximising precision.

Figure 29: Hyperparameter search results for regression fine-tuning a `longformer-mini-1024`

8 Discussion and Conclusions

Table 5 shows the summarised results from figs. 21, 23, 25 and 27. The main findings from the experiments are summarised as follows:

First, directly predicting smoothed label distributions via regression performs poorly compared to classification-based training. The regressors fail to capture the multi-modal structure of the targets, converging to overly simple normal Gaussian distributions over the entire label space. Classification provides a more suitable objective for this task.

Among classification architectures, `longformer-mini-1024` and `albert-base-v2` consistently outperform other models as shown in tables 2 and 3. The reduced parameterisation and longer context size of these models likely support more effective adaptation to the small target dataset.

The hyperparameter importance analysis reveals learning rate and weight decay strongly influence precision, with negative correlations suggesting models benefit from smaller values to avoid overfitting. Vocabulary size also clearly impacts performance, as models struggle to develop useful embeddings for unfamiliar words.

Interestingly, adding newline characters is surprisingly important and has a largely positive correlation with classification performance for both binary and multi-class regimes. Unsurprisingly, adding special tokens improves classification accuracy across the board. The separators likely help models isolate and process the two input sentences independently before combining their representations. However, the addition of [EOT] produces marginal increase or non-existent gains in performance. Adding [SEP] has a positive correlation with classification performance across all tested models. Additionally, adding both [SEP] and [EOT] tokens seems to perform worse than using only a single set of either [EOT] or [SEP] tokens across binary and multi-class regimes.

This research makes important initial steps towards evaluating adversarial attacks on language models. The experiments demonstrate the feasibility of using models themselves to assess alignment risks, without reliance on slow and subjective human evaluation. By framing alignment as a learnable property, large-scale automated misalignment detection through supervised training is hopefully enabled. However, substantial work remains to make such evaluation robust and comprehensive. Current models still struggle with certain complex or subtle attacks not represented in training data. Ideally, alignment evaluation would eventually progress to provable guarantees on model behaviours, eliminating the need for empirical testing as described in Section 9.

Table 5: Final results

Model	Accuracy	Precision	Recall	F1 Score
<code>albert-base-v2 + all_sep + Additional newlines</code>	0.81	0.80	0.81	0.80
<code>albert-base-v2 + sep_eot + Additional newlines + Imbalance Fix</code>	0.81	0.80	0.80	0.80
<code>albert-base-v2 + all_sep + Imbalance Fix</code>	0.80	0.80	0.80	0.79
<code>longformer-mini-1024 + all_eot + Imbalance Fix</code>	0.78	0.77	0.78	0.78
<code>longformer-mini-1024 + all_sep + Additional newlines + Imbalance Fix</code>	0.77	0.76	0.77	0.76
<code>longformer-mini-1024 + sep_eot + Additional newlines</code>	0.76	0.75	0.76	0.75

9 Future Work

While this thesis makes important steps towards developing automated misalignment detection and evaluation methods for generative language models, ample opportunity remains for impactful follow-on research.

Training models with billions of parameters requires massive compute resources, often accessible only to large tech companies. In addition, while some architectures are publicly described, their actual trained weights remain proprietary and unavailable. Overcoming these restrictions through distributed training, parameter efficient methods, and open research initiatives would enable studying alignment risks in models most likely to impact society.

Expanding the training and evaluation data scale also poses challenges. Human annotation of model behaviours is slow, subjective, and expensive to outsource. Meanwhile, most large high-quality datasets remain privately held by tech firms. Generating synthetic training data through white-box adversarial attacks could provide a scalable alternative, but incurs huge computational costs. Developing efficient and diverse data generation procedures is critical for improved experimentation.

9.1 Mechanistic Interpretability

Understanding the root causes behind adversarial attacks by analysing the model’s internal representations and attributing its predictions to specific input features is a very promising direction of future research.

Several studies [Bach et al., 2015, Shrikumar et al., 2017] have examined the intermediate activations of neural networks to identify components that are highly influential to the model’s final outputs. For vision models, techniques like saliency maps [Simonyan et al., 2013] highlight input pixels that activate certain neurons. In NLP, attention mechanisms [Bahdanau et al., 2015] reveal how much each input token contributes to the model’s next word prediction. While these methods provide some explanation of the model’s reasoning, they have limitations in terms of faithfulness and sensitivity [Jain and Wallace, 2019].

Mechanistic interpretability refers to the ability to dissect and comprehend the inner workings of complex models like LLMs [Olah, 2022]. It involves dissecting the model’s architecture and examining how inputs are transformed into outputs at various stages. Several methods have been proposed to enhance mechanistic interpretability, including attention maps, activation maximisation [Olah et al., 2017], and gradient-based techniques [Sundararajan et al., 2017]. By dissecting the LLM’s internal operations, the decision-making processes and potential vulnerabilities that lead to adversarial susceptibility can be uncovered.

Contrastive examples play a pivotal role in adversarial attack analysis. These examples comprise pairs of inputs where one provokes an adversarial response from the LLM, while the other does not. Investigating these pairs provides valuable insights into the discriminative features or decision boundaries that render the model susceptible to attacks. By feeding contrastive examples into an interpretable model and tracing intermediate outputs, one can identify which components get triggered differently. If an adversarial input perturbs the syntax module but leaves the semantics module unchanged, the attack could be attributed to syntactic vulnerability. Analysing multiple attack types in this fashion would reveal whether certain components are more prone to manipulation. Tracing how contrastive inputs propagate through these interpretable modules can shed light on the triggers that lead to erroneous or biased outputs.

The mechanistic approach shows promise for future research in analysing adversarial attacks on LLMs and provides a promising path toward building more robust and trustworthy LLMs.

References

- Nayyer Aafaq, Naveed Akhtar, Wei Liu, Mubarak Shah, and Ajmal Mian. Language model agnostic gray-box adversarial attack on image captioning. *IEEE Transactions on Information Forensics and Security*, 18:626–638, 2022.
- Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*, 2023.
- Anthropic. Claude, a next generation ai-assistant by anthropic. <https://www.anthropic.com/product>, 2023. Accessed: 2023-09-13.
- Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140, 2015.
- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL <https://www.cs.ubc.ca/~amuhamed/LING530/papers/bahdanau2014neural.pdf>.
- Tao Bai, Jinqi Luo, Jun Zhao, Bihan Wen, and Qian Wang. Recent advances in adversarial training for adversarial robustness. *arXiv preprint arXiv:2102.01356*, 2021.
- Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*, 2022.
- Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.
- Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom B Brown, Dawn Song, Ulfar Erlingsson, et al. Extracting training data from large language models. In *USENIX Security Symposium*, volume 6, 2021.
- Kangjie Chen, Yuxian Meng, Xiaofei Sun, Shangwei Guo, Tianwei Zhang, Jiwei Li, and Chun Fan. Badpre: Task-agnostic backdoor attacks to pre-trained nlp foundation models. *arXiv preprint arXiv:2110.02467*, 2021.

Shuyu Cheng, Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu. Improving black-box adversarial attacks with a transfer-based prior. *Advances in neural information processing systems*, 32, 2019.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. See <https://vicuna.lmsys.org> (accessed 14 April 2023), 2023.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL <https://aclanthology.org/N19-1423>.

Guneet S Dhillon, Kamyar Azizzadenesheli, Zachary C Lipton, Jeremy Bernstein, Jean Kossaifi, Aran Khanna, and Anima Anandkumar. Stochastic activation pruning for robust adversarial defense. *arXiv preprint arXiv:1803.01442*, 2018.

Javid Ebrahimi, Daniel Lowd, and Dejing Dou. On adversarial examples for character-level neural machine translation. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 653–663, Santa Fe, New Mexico, USA, August 2018. Association for Computational Linguistics. URL <https://aclanthology.org/C18-1055>.

Deep Ganguli, Liane Lovitt, Jackson Kernion, Amanda Askell, Yuntao Bai, Saurav Kadavath, Ben Mann, Ethan Perez, Nicholas Schiefer, Kamal Ndousse, et al. Red teaming language models to reduce harms: Methods, scaling behaviors, and lessons learned. *arXiv preprint arXiv:2209.07858*, 2022.

Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A Smith. Real-toxicityprompts: Evaluating neural toxic degeneration in language models. *arXiv preprint arXiv:2009.11462*, 2020.

Kai Greshake, Sahar Abdelnabi, Shailesh Mishra, Christoph Endres, Thorsten Holz, and Mario Fritz. Not what you've signed up for: Compromising real-world llm-integrated applications with indirect prompt injection. *arXiv preprint arXiv:2302.12173*, 2023.

Chuan Guo, Alexandre Sablayrolles, Hervé Jégou, and Douwe Kiela. Gradient-based adversarial attacks against text transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5747–5757, Online and Punta Cana, Dominican Republic, November 2021a. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.464. URL <https://aclanthology.org/2021.emnlp-main.464>.

Chuan Guo, Alexandre Sablayrolles, Hervé Jégou, and Douwe Kiela. Gradient-based adversarial attacks against text transformers. *arXiv preprint arXiv:2104.13733*, 2021b.

Yiwen Guo, Ziang Yan, and Changshui Zhang. Subspace attack: Exploiting promising subspaces for query-efficient black-box attacks. *Advances in Neural Information Processing Systems*, 32, 2019.

Zhezhi He, Adnan Siraj Rakin, and Deliang Fan. Parametric noise injection: Trainable randomness to improve deep neural network robustness against adversarial attack. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 588–597, 2019.

Will Douglas Heaven. Why meta’s latest large language model survived only three days online. *MIT Technology Review*. Last accessed December, 15:2022, 2022.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

Nathan Inkawich, Wei Wen, Hai Helen Li, and Yiran Chen. Feature space perturbations yield more transferable adversarial examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7066–7074, 2019.

Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. Adversarial example generation with syntactically controlled paraphrase networks. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1875–1885, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1170. URL <https://aclanthology.org/N18-1170>.

Sarthak Jain and Byron C. Wallace. Attention is not Explanation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3543–3556, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1357. URL <https://aclanthology.org/N19-1357>.

Robin Jia and Percy Liang. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031, Copenhagen, Denmark, September 2017. Association for

Computational Linguistics. doi: 10.18653/v1/D17-1215. URL <https://aclanthology.org/D17-1215>.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*, 2019.

Ronan Le Bras, Swabha Swayamdipta, Chandra Bhagavatula, Rowan Zellers, Matthew Peters, Ashish Sabharwal, and Yejin Choi. Adversarial filters of dataset biases. In *International conference on machine learning*, pages 1078–1088. PMLR, 2020.

Yuhui Li, Fangyun Wei, Jinjing Zhao, Chao Zhang, and Hongyang Zhang. Rain: Your language models can align themselves without finetuning. *arXiv preprint arXiv:2309.07124*, 2023.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Haipeng Luo, Qingfeng Sun, Can Xu, Pu Zhao, Jianguang Lou, Chongyang Tao, Xiubo Geng, Qingwei Lin, Shifeng Chen, and Dongmei Zhang. Wizardmath: Empowering mathematical reasoning for large language models via reinforced evol-instruct. *arXiv preprint arXiv:2308.09583*, 2023.

Xinyao Ma, Maarten Sap, Hannah Rashkin, and Yejin Choi. PowerTransformer: Unsupervised controllable revision for biased language correction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7426–7441, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.602. URL <https://aclanthology.org/2020.emnlp-main.602>.

Natalie Maus, Patrick Chao, Eric Wong, and Jacob Gardner. Adversarial prompting for black box foundation models. *arXiv preprint arXiv:2302.04237*, 2023.

Aamir Mustafa, Salman Khan, Munawar Hayat, Roland Goecke, Jianbing Shen, and Ling Shao. Adversarial defense by restricting the hidden space of deep neural networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3385–3394, 2019.

Chris Olah. Mechanistic interpretability, 2022. URL <https://transformer-circuits.pub/2022/mech-interp-essay/index.html>.

Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. Feature visualization. *Distill*, 2(11):e7, 2017.

R OpenAI. Gpt-4 technical report. *arXiv*, pages 2303–08774, 2023.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.

Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, pages 506–519, 2017.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics. doi: 10.3115/1073083.1073135. URL <https://aclanthology.org/P02-1040>.

Fábio Perez and Ian Ribeiro. Ignore previous prompt: Attack techniques for language models. In *NeurIPS ML Safety Workshop*, 2022.

Matt Post. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium, October 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-6319. URL <https://aclanthology.org/W18-6319>.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *International conference on machine learning*, pages 3145–3153. PMLR, 2017.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

Stanford. The center for research on foundation models, stanford institute for human-centered artificial intelligence. <https://crfm.stanford.edu/>, 2023. Accessed: 2023-09-13.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International conference on machine learning*, pages 3319–3328. PMLR, 2017.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

Rohan Taori, Ishaaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. Alpaca: A strong, replicable instruction-following model. *Stanford Center for Research on Foundation Models*. <https://crfm.stanford.edu/2023/03/13/alpaca.html>, 3(6):7, 2023.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Pratik Vaishnavi, Tianji Cong, Kevin Eykholt, Atul Prakash, and Amir Rahmati. Can attention masks improve adversarial robustness? In *International Workshop on Engineering Dependable and Secure Machine Learning Systems*, pages 14–22. Springer, 2020.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. Universal adversarial triggers for attacking and analyzing nlp. *arXiv preprint arXiv:1908.07125*, 2019.

Ben Wang and Aran Komatsuzaki. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. <https://github.com/kingoflolz/mesh-transformer-jax>, May 2021.

Thomas Wang, Adam Roberts, Daniel Hesslow, Teven Le Scao, Hyung Won Chung, Iz Beltagy, Julien Launay, and Colin Raffel. What language model architecture and pretraining objective works best for zero-shot generalization? In *International Conference on Machine Learning*, pages 22964–22984. PMLR, 2022.

Yufei Wang, Wanjun Zhong, Liangyou Li, Fei Mi, Xingshan Zeng, Wenyong Huang, Lifeng Shang, Xin Jiang, and Qun Liu. Aligning large language models with human: A survey. *arXiv preprint arXiv:2307.12966*, 2023.

Johannes Welbl, Amelia Glaese, Jonathan Uesato, Sumanth Dathathri, John Mellor, Lisa Anne Hendricks, Kirsty Anderson, Pushmeet Kohli, Ben Coppin, and Po-Sen Huang. Challenges in detoxifying language models. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2447–2469, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-emnlp.210. URL <https://aclanthology.org/2021.findings-emnlp.210>.

Simon Willison. Prompt injection attacks against GPT-3, 2023. URL <https://simonwillison.net/2022/Sep/12/prompt-injection/>.

Liu Zhuang, Lin Wayne, Shi Ya, and Zhao Jun. A robustly optimized BERT pre-training approach with post-training. In *Proceedings of the 20th Chinese National Conference on Computational Linguistics*, pages 1218–1227, Huhhot, China, August 2021. Chinese Information Processing Society of China. URL <https://aclanthology.org/2021.ccl-1.108>.

Andy Zou, Zifan Wang, J Zico Kolter, and Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*, 2023.

10 Appendix: Code Structure

The code for this project is contained in a single Jupyter notebook named `sandbox.ipynb` located in the root directory of the GitHub repository at <https://git.cs.bham.ac.uk/projects-2022-23/rsd259>. This notebook contains all data loading, preprocessing, model implementation, training, evaluation, and analysis. To run, first execute the setup cells to install packages, import dependencies and define helper functions and constants. Then run the cells under the selected experiment sequentially to replicate results. Note that the experiments require significant GPU resources - a minimum of V100 is needed for smaller models while A100 is recommended for larger architectures. Be sure to run cells in order as later experiments depend on objects defined earlier. Refer to the notebook for additional details on parameter configurations and hardware requirements.