Adversarial Neuro-symbolic model

Gilberto, Daniel, and Chris
CS53700
Professor: Rajiv Khanna

Outline

- Adversarial Robustness
  - Part-based Models
- Neuro-symbolic methods
  - Problog
- Method
- Results

Adversarial Robustness

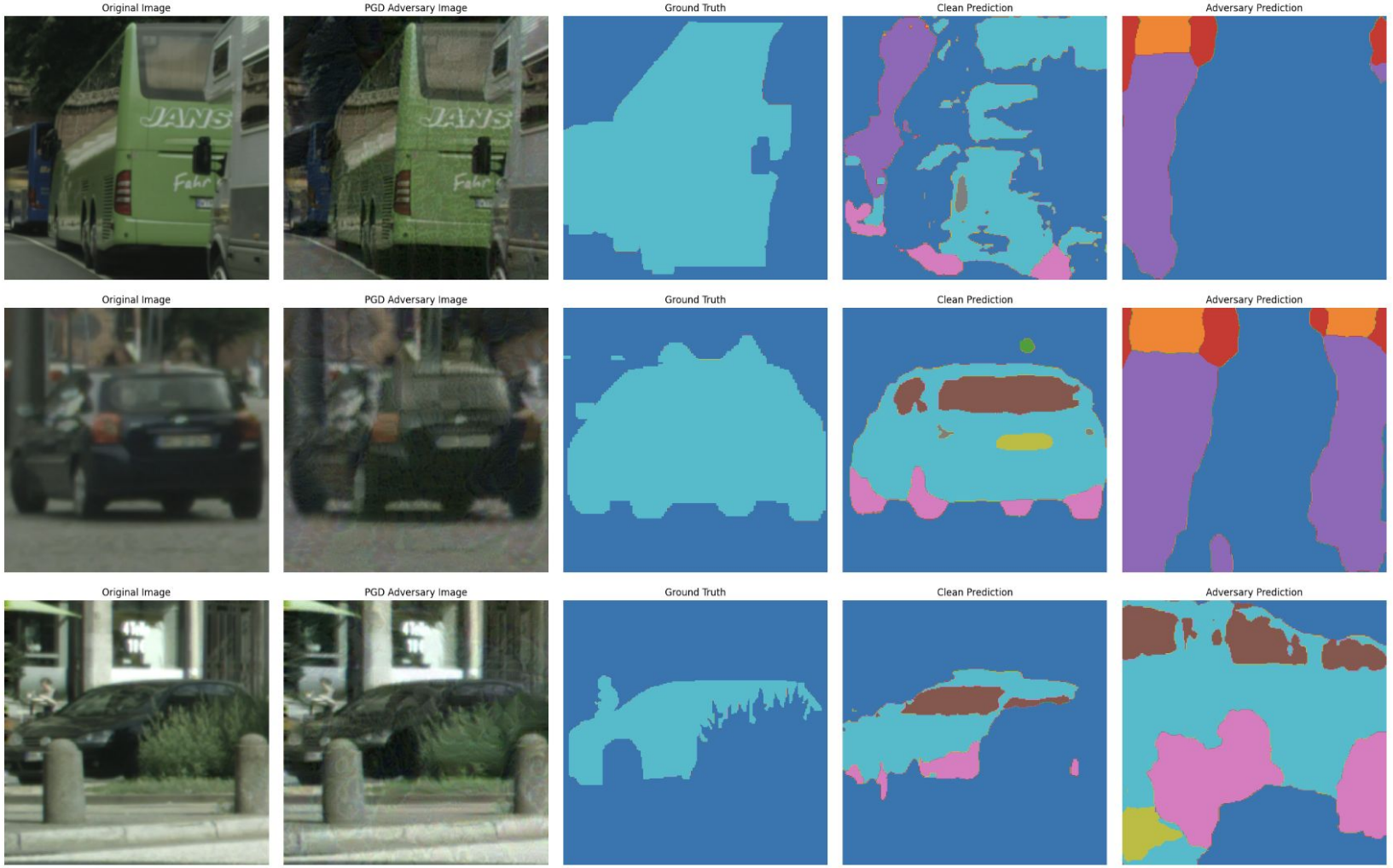
Given a machine learning model for a classification task and an input with a true label (x, y) ; an adversary attack against the model consists of slightly altering the input to deceive the model in its inference $(\hat{x}, y)$

These alterations trick the model into misclassifying the altered input.

Adversarial robustness is then the resilience of the model against such attacks, this property of the model is very important in many applications

- Medicine
- Autonomous vehicles


Original Image
PGD Adversary Image
Ground Truth
Clean Prediction
Adversary Prediction
Original Image
PGD Adversary Image
Ground Truth
Clean Prediction
Adversary Prediction
Original Image
PGD Adversary Image
Ground Truth
Clean Prediction
Adversary Prediction


Original Image
PGD Adversary Image
Ground Truth
Clean Prediction
Adversary Prediction
Original Image
PGD Adversary Image
Ground Truth
Clean Prediction
Adversary Prediction
Original Image
PGD Adversary Image
Ground Truth
Clean Prediction
Adversary Prediction

Adversarial Robustness

Recent works has been using generated adversarial data during the training phase to improve adversarial robustness or have defined different training loss functions. However, results have plateaued.

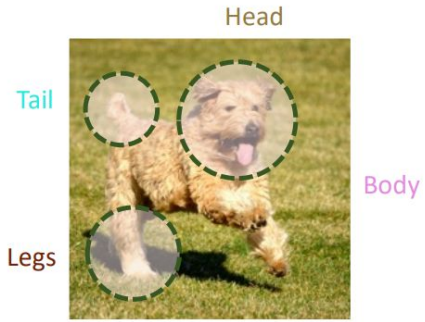
How can we have a better protocol to improve adversarial robustness?

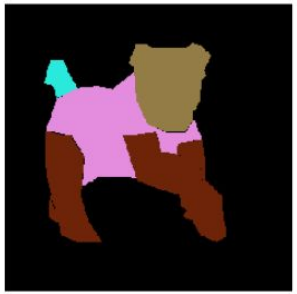
- What if the data fed in has more information?
  - Part segmentation

Part-based Models

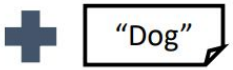
Part-based models is a two part machine learning model that utilizes a human prior to create robustness against adversarial attacks.

The first part is trained to retrieve part segmentations from the input image.





Part segmentation
(fine-grained label)



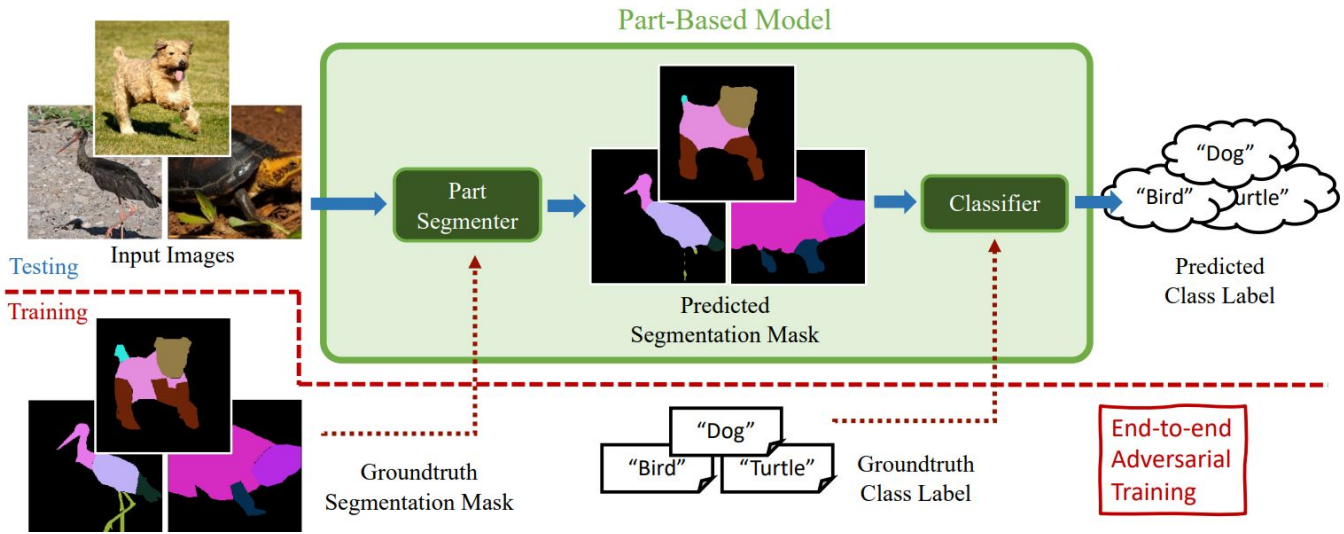
Class label

Sitawarin, Chawin & Pongmala, Kornrapat & Chen, Yizheng & Carlini, Nicholas & Wagner, David. (2022). Part-Based Models Improve Adversarial Robustness. 10.48550/arXiv.2209.09117.

Part-based Models

Using information from the first part. We pass the output into the second part.

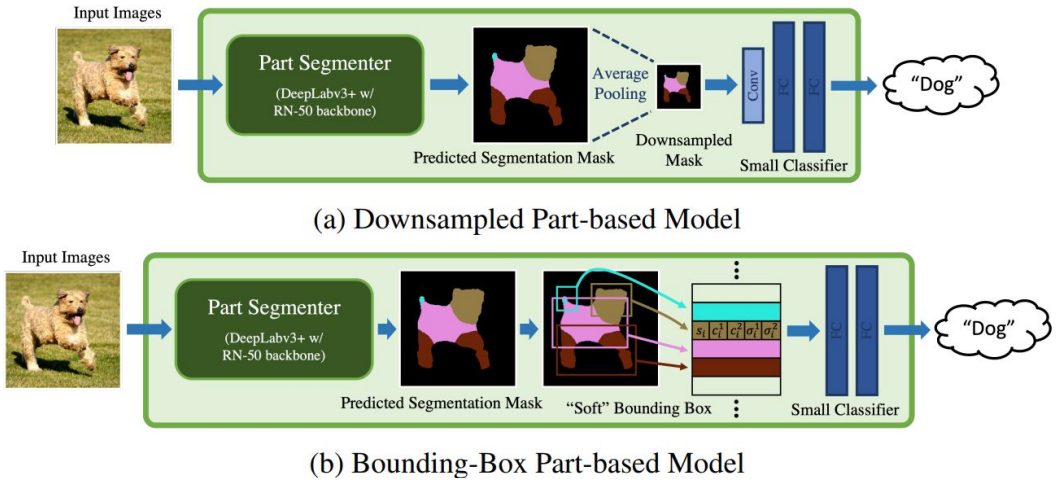
In the second part, a model is trained with a ground truth segmentation and labels. The goal is to use the predicted segmentation from the first part and then output a class label.



Sitawarin, Chawin & Pongmala, Kornrapat & Chen, Yizheng & Carlini, Nicholas & Wagner, David. (2022). Part-Based Models Improve Adversarial Robustness. 10.48550/arXiv.2209.09117.

Part-based Models

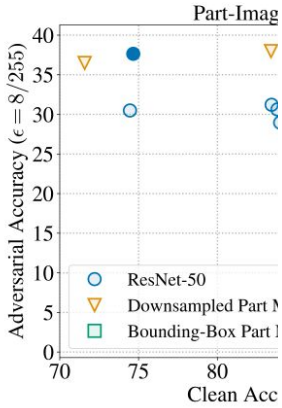
They introduce two methods to utilize the segmentation information



Sitawarin, Chawin & Pongmala, Kornrapat & Chen, Yizheng & Carlini, Nicholas & Wagner, David. (2022). Part-Based Models Improve Adversarial Robustness. 10.48550/arXiv.2209.09117.

Part-based Models

Results for 3 datasets



(a) Part-Ima

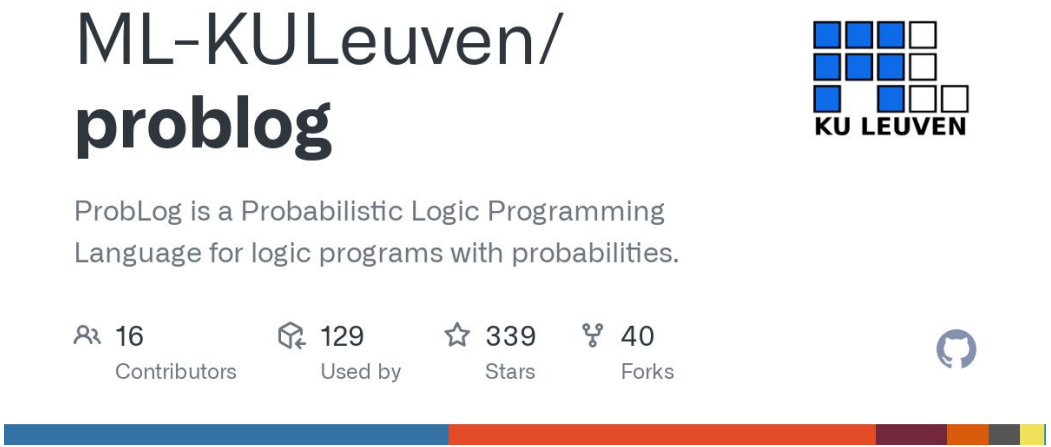
Training Method	Models	Part-ImageNet		Cityscapes		PASCAL-Part	
		Clean	Adv.	Clean	Adv.	Clean	Adv.
PGD (Madry et al., 2018)	ResNet-50	74.7	37.7	79.5	68.4	47.1	37.8
	Downsampled Part Model	85.6 (↑ 10.9)	39.4 (↑ 1.7)	94.8 (↑ 15.3)	70.2 (↑ 1.8)	49.6 (↑ 2.5)	38.5 (↑ 0.7)
	Bounding-Box Part Model	86.5 (↑ 11.8)	39.2 (↑ 1.5)	94.2 (↑ 14.7)	69.9 (↑ 1.4)	52.2 (↑ 5.1)	38.5 (↑ 0.7)
TRADES (Zhang et al., 2019)	ResNet-50	90.6	7.7	96.7	52.5	80.2	12.6
	Downsampled Part Model	90.9 (↑ 0.3)	19.8 (↑ 12.1)	97.1 (↑ 0.4)	62.5 (↑ 10.0)	83.1 (↑ 2.9)	29.9 (↑ 17.3)
	Bounding-Box Part Model	90.8 (↑ 0.2)	24.1 (↑ 16.4)	97.1 (↑ 0.4)	63.0 (↑ 10.5)	88.5 (↑ 8.3)	29.5 (↑ 16.9)

Sitawarin, Chawin & Pongmala, Kornrapat & Chen, Yizheng & Carlini, Nicholas & Wagner, David. (2022). Part-Based Models Improve Adversarial Robustness. 10.48550/arXiv.2209.09117.

Neuro-symbolic AI

Neuro-symbolic AI aims to integrate the logical reasoning capabilities of symbolic systems with pattern recognition abilities of neural networks.

We introduce Problog which is a symbolic tool, that allows us to encode complex interactions based on annotated probabilities.



https://github.com/ML-KULeuven/problog

Problog

Problog allows us to define probabilities for different events and specific actions based on certain events

```
% Probabilistic facts:
0.5::heads1.
0.5::heads2.

% Rules:
twoHeads :- heads1, heads2.

% Queries:
query(heads1).
query(heads2).
query(twoHeads).
```

Evaluate

Query ▼	Location	Probability
heads1	9:7	0.5
heads2	10:7	0.5
twoHeads	11:7	0.25

https://problog.readthedocs.io/en/latest/

Problog

How does it related to AI?

Problog will update the probabilities (learning) based on evidence that is provided to it.

```
%%% The program:
t(0.5)::burglary.
0.2::earthquake.
t(_)::p_alarm1.
t(_)::p_alarm2.
t(_)::p_alarm3.

alarm :- burglary, earthquake, p_alarm1.
alarm :- burglary, \+earthquake, p_alarm2.
alarm :- \+burglary, earthquake, p_alarm3.
```

Examples (specified as evidence, separated by ---):

```
%%% The data:
evidence(burglary,false).
evidence(alarm,false).
-----
evidence(earthquake,false).
evidence(alarm,true).
evidence(burglary,true).
-----
evidence(burglary,false).
```

Fact▼	Location	Probability
t(0.5)::burglary	2:9	0.33333333
t(_)::p_alarm1	4:7	0.82017878
t(_)::p_alarm2	5:7	1
t(_)::p_alarm3	6:7	0

Model:

```
0.333333333333333::burglary.
0.2::earthquake.
0.820178784187124::p_alarm1.
1.0::p_alarm2.
0.0::p_alarm3.
alarm :- burglary, earthquake, p_alarm1.
alarm :- burglary, \+earthquake, p_alarm2.
alarm :- \+burglary, earthquake, p_alarm3.
```

https://problog.readthedocs.io/en/latest/

Our Method

The Part-segmentation model will remain the same as from the Part-based model paper.

How will we feed in segmentation information into Problog?

- We will utilize the CityScapes Dataset, the part-segmentation model will output multiple labels for segmented parts with a true class label
  - True class label of human vs car
  - Human segmentation labels: arm, head, torso, etc
  - Non-human: windows, chassis, wheels, etc.

- We define initial probabilities based on prior beliefs
  - We define basic dependency rules, and let the program learn from data

Problog for human vs car classification

```
# 1. ProbLog model
model_string = """
% priors beliefs
0.1::has_arms.
0.1::has_legs.
0.1::has_head.
0.1::has_torso.
0.1::has_wheels.
0.1::has_windows.
0.1::has_lights.
0.1::has_license_plate.
0.1::has_chassis.

is_car:- \+is_human.
```

Problog update the prior beliefs based on data

We assume a uniform prior for every feature.

We define simple conditional rules based on features

For every feature we try to estimate the probability of the class given the feature.

P(human | arms)

Given examples i.e. (arms=True, windows=False, is_human=True, etc.)

Problog uses the EM algorithm to estimate the probabilities based on data.

```
% conditionals that are logical
t(p_human1)::is_human :- has_arms.
t(p_human2)::is_human :- has_legs.
t(p_human3)::is_human :- has_head.
t(p_human4)::is_human :- has_torso.

t(p_car1)::is_car :- has_wheels.
t(p_car2)::is_car :- has_windows.
t(p_car3)::is_car :- has_lights.
t(p_car4)::is_car :- has_license_plate.
t(p_car5)::is_car :- has_chassis.
```

By the nature of EM problog can learn and give predictions on observations with missing data

```
# New observation: has_windows = True, has_chassis = True
observation = [
    ('has_windows', True),
    ('has_chassis', True),
]
learned_model_str = lfi_problem.get_model()
pred = predict_observation(observation, learned_model_str)
print(pred)
```

```
Predicted probabilities given the observation:
is_car: 0.9362
is_human: 0.2232
car
```

Results

Results (10 epochs, bounding box)			
Acc	part-based(base)	part-based-adv-PGD	**Seg+ProbLog**
no-attack	99.614	98.380	X
PGD-attack	X	69.9	84.259

Future Work

The ultimate goal of this work is to create a benchmark that can test neuro-symbolic methods in adversarial settings. This project is an initial test adversarial robustness for neuro-symbolic models. We have a few questions in mind to explore:

- Can we replace Problog with other neuro-symbolic models?
- Can we replace the part-based segmentation with other models (i.e. concept bottleneck models)?

Thank you

References:

[1] Sitawarin, Chawin, et al. "Part-based models improve adversarial robustness." arXiv preprint arXiv:2209.09117 (2022).

[2] De Raedt, L., Kimmig, A. Probabilistic (logic) programming concepts. Mach Learn 100, 5–47 (2015). https://doi.org/10.1007/s10994-015-5494-z

Constructing an LLM fine-tuned on Code Reasoning datasets using Low-Rank Adaptation

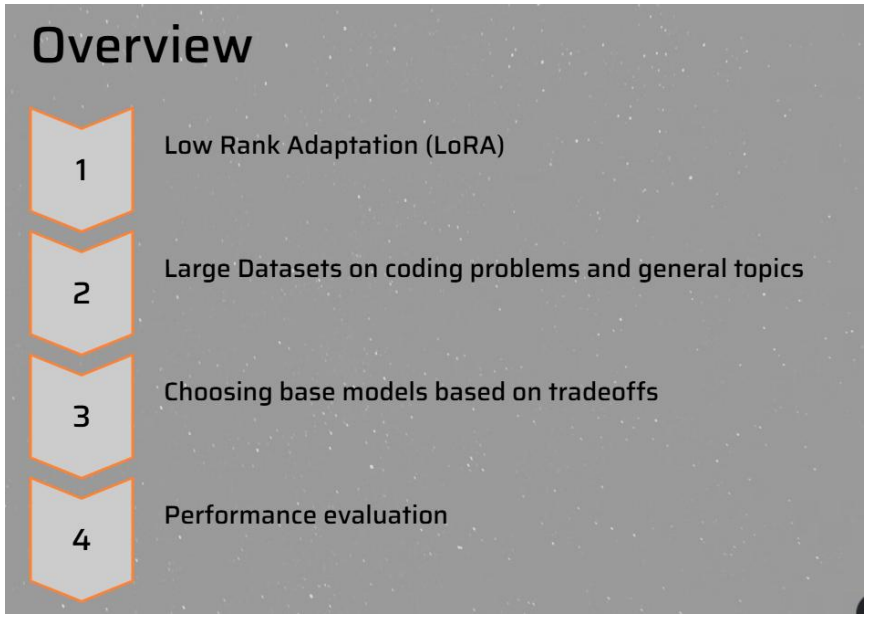
SRIHITHA KURAPATI, PRANAV SANGHI, SAMSKRITHI SIVAKUMAR

Introduction

Large Language Models (LLMs) have proven to surpass human intelligence in tasks such as memorization, contextual learning, reasoning, coding ability and problem solving.

However, popular LLMs might not be adapted to a specific task or domain that you want!

Over the semester, we explored ways to leverage the power of LLMs by running it on custom datasets.

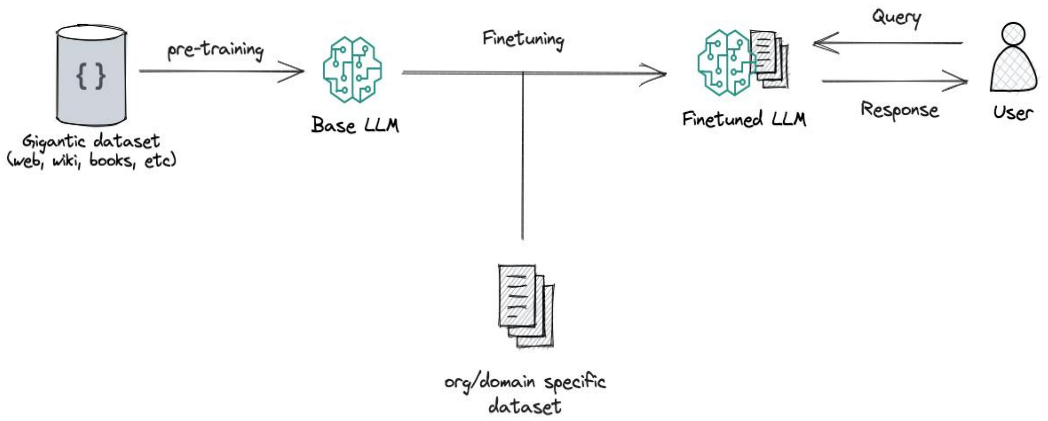


About Fine-tuning LLMs

Large Language Models are trained on large amounts of textual data and code, and use a neural network architecture, often based on the Transformer model, to process and generate text.

Fine-tuning involves adopting a pre-trained model on a specific task or domain, by exposing it to a more focused dataset and adjusting its parameters.

This can be computationally expensive and hard to store. Example: a checkpoint to fine-tune a 175B parameter variant is 1 TB large.



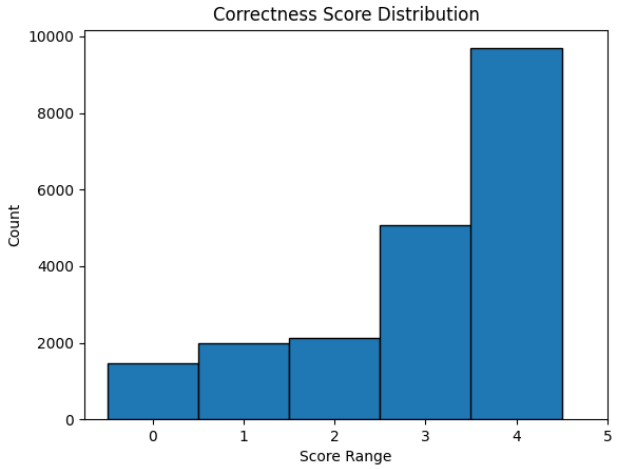
Avikumar Talaviya (2024) Source: https://medium.com/@avikumart_/evaluation-of-fine-tuned-llm-using-monsterapi-a67a7714a65b

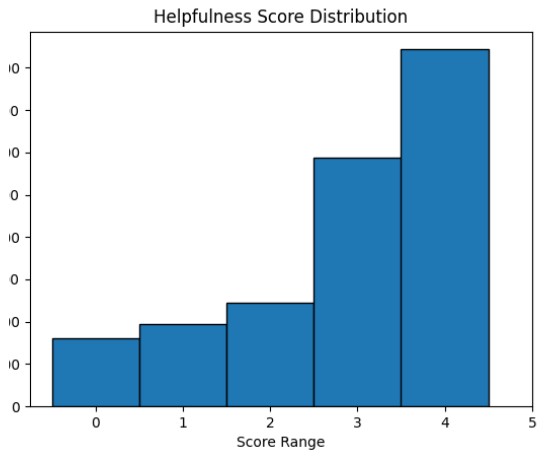
Newer methods of more efficient fine-tuning, such as Parameter-Efficient Fine-Tuning (PEFT) are being developed. Techniques include LoRA, DoRA (Weight-Decomposed Low-Rank Adaptation), Gradual unfreezing

Dataset

HELPSTEER2 DATASET

- TRAIN: 20324 prompts to 8306 prompts
- VAL: 1028 prompts to 425 prompts





OPEN CODE REASONING DATASET

- Released by Nvidia in 2025
- Largest reasoning-based synthetic dataset to date for coding
- Clear reference outputs for evaluating how well the model reasons through code tasks
- 500,000+ prompts to 5679 prompts

Models

GPT-NEO 1.3B (ELEUTHERAI) - 2021

- 1.3B parameters
- Trained on The Pile dataset
- Autoregressive transformer architecture
- Cross-entropy loss
- Optimized for text generation and language modeling

Source: https://huggingface.co/KoboldAI/GPT-Neo-1.3B-Adventure

GEMMA-3 (UNSLOTH) - 2025

- 1B parameters
- Trained on variety of documents
- Supervised Fine-Tuning (SFT), Rejection Sampling (RS), and Direct Preference Optimization (DPO)
- Optimized for resource-constrained devices such as mobiles.

Source: https://huggingface.co/google/gemma-3-1b-it

LLAMA-3 (UNSLOTH) - 2024

- 1B parameters
- Trained on internet data
- Supervised Fine-Tuning (SFT), Rejection Sampling (RS), and Direct Preference Optimization (DPO)
- Optimized on Multilingual Dialogue and use for Agentic Applications

Source: https://ai.meta.com/blog/meta-llama-3/

Low Rank Adaptation (LoRA)

Low Rank Adaptation (LoRA) is a technique of efficient fine-tuning of LLMs by applying low-rank updates to selected weight matrices.

1. Update a fraction of parameters only
2. Low rank matrix multiplication to use less parameters

Benefits of LoRA:

1. Dramatic reduction in checkpoint size. For instance, when fine-tuning GPT-3 (175B), LoRA can shrink model checkpoint from 1 TB to 25 MB (Parameters from 175B to 4.7M).
2. Preservation of the original pre-trained weights, allowing for multiple lightweight models for different tasks.

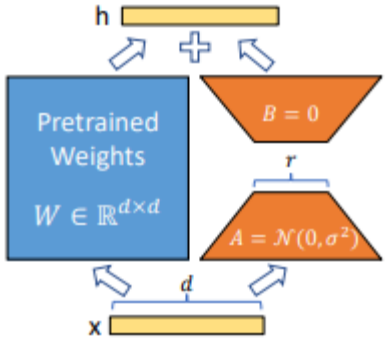


Figure 1: Our reparametrization. We only train A and B.

Hu Edward (2021)
SOURCE: LORA: LOW-RANK ADAPTATION OF LARGE LANGUAGE MODELS [5]

DPO

1. Motivation

DPO aligns models to human preferences using pairwise comparisons.

Standard DPO is sensitive to the β parameter (controls sharpness of preference learning).

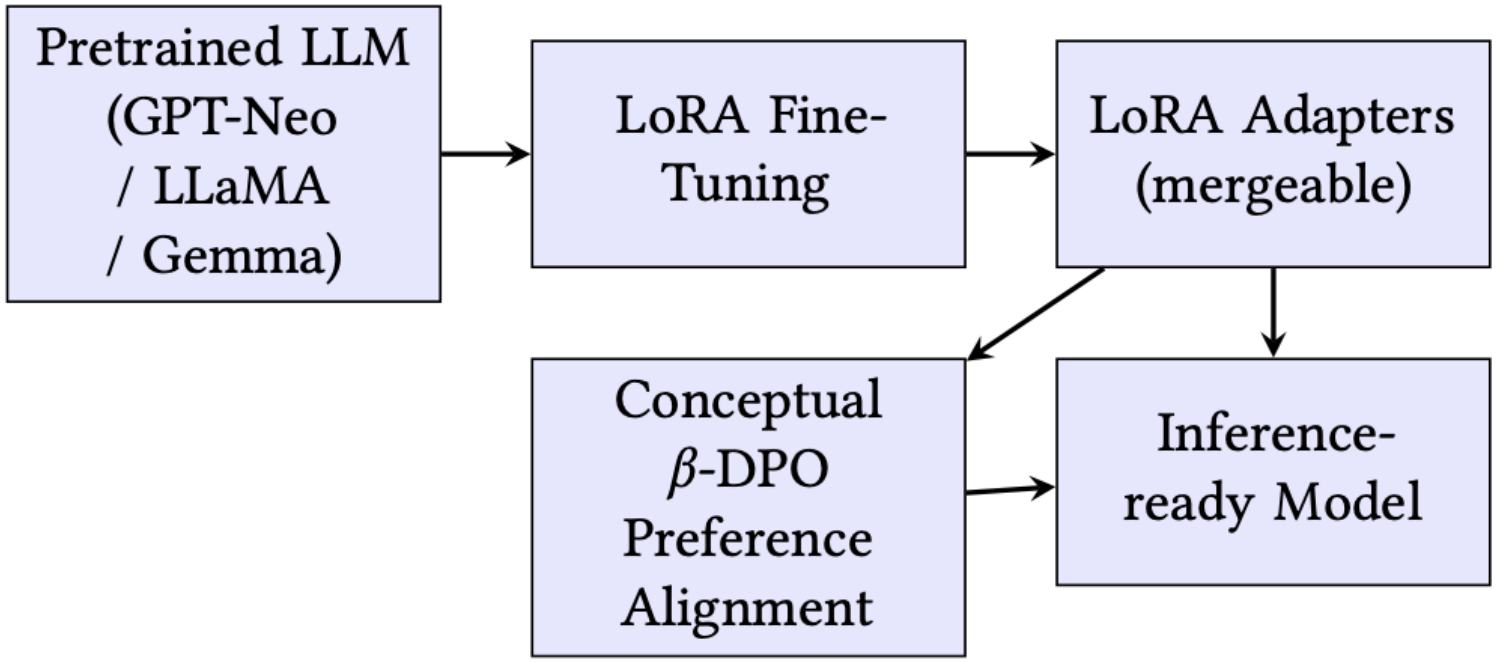
β-DPO improves stability and performance by adapting β dynamically per batch.

2. Key Concepts

Dynamic β Calibration: Adjust β higher for clearer preferences, lower for noisy data.

β-Guided Outlier Filtering: Discard low-confidence preference pairs in each bat

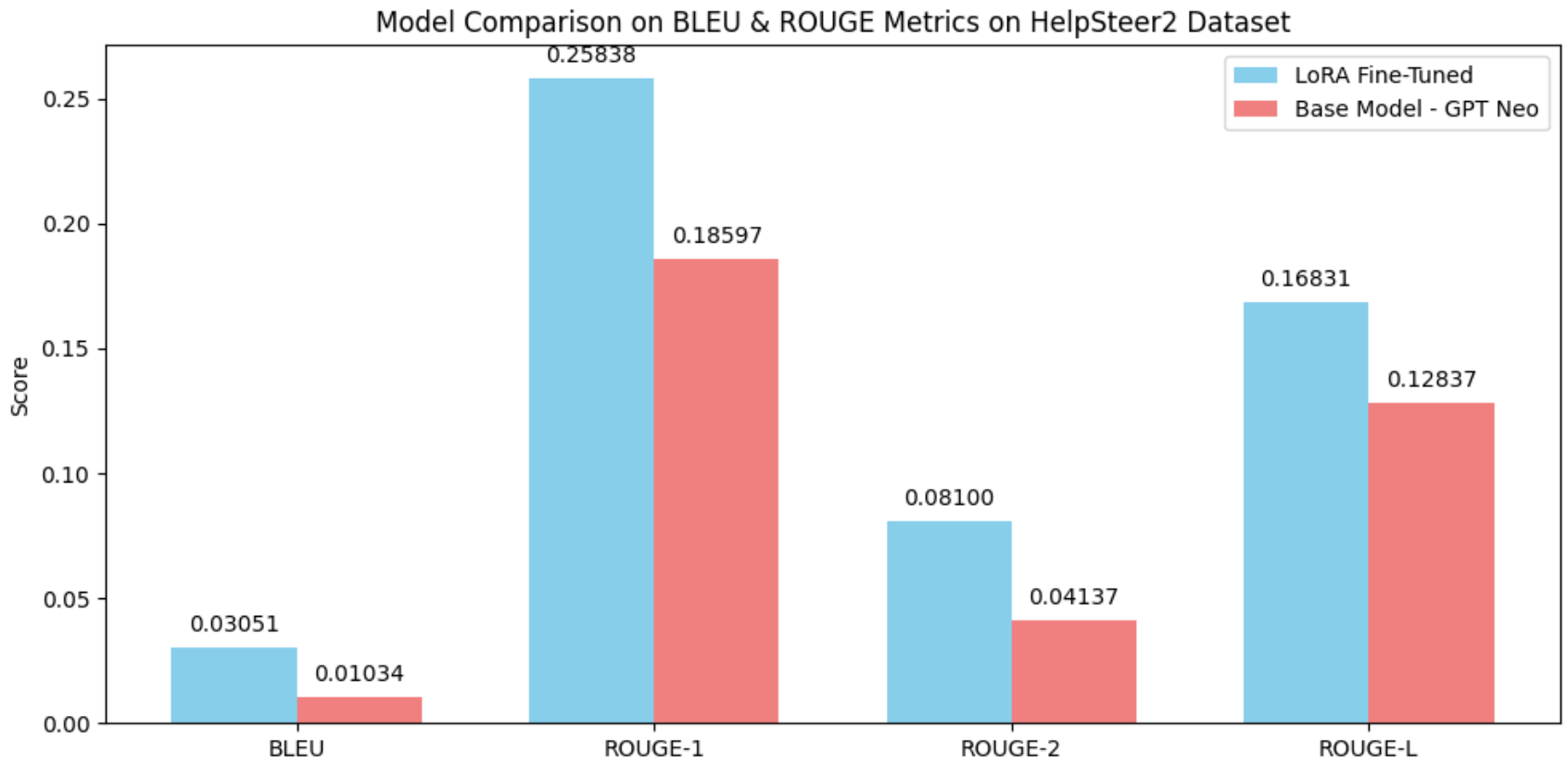
Methodology



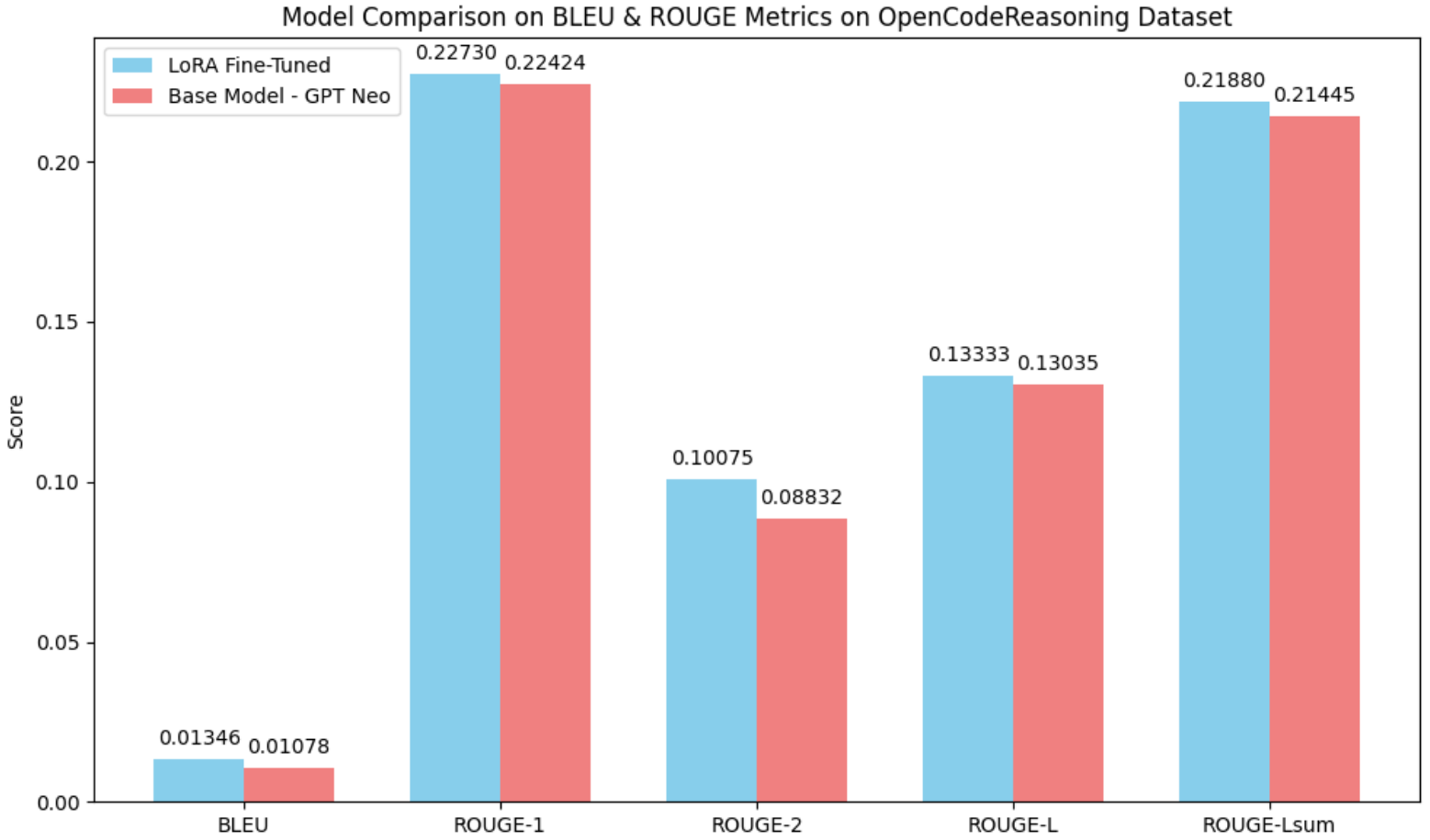
Metrics

- **BLEU (Bilingual Evaluation Understudy):** Match of multi-word sequences (n-grams) between generated and reference text
- **ROUGE (Recall-Oriented Understudy for Gisting Evaluation) 1:** Match of individual words (unigrams)
- **ROUGE 2:** Match of two-word phrases (bigrams)
- **ROUGE L:** Longest common subsequence (order-aware match)
- **ROUGLE L-SUM:** Variant of ROUGE-L optimized for summary-level evaluation

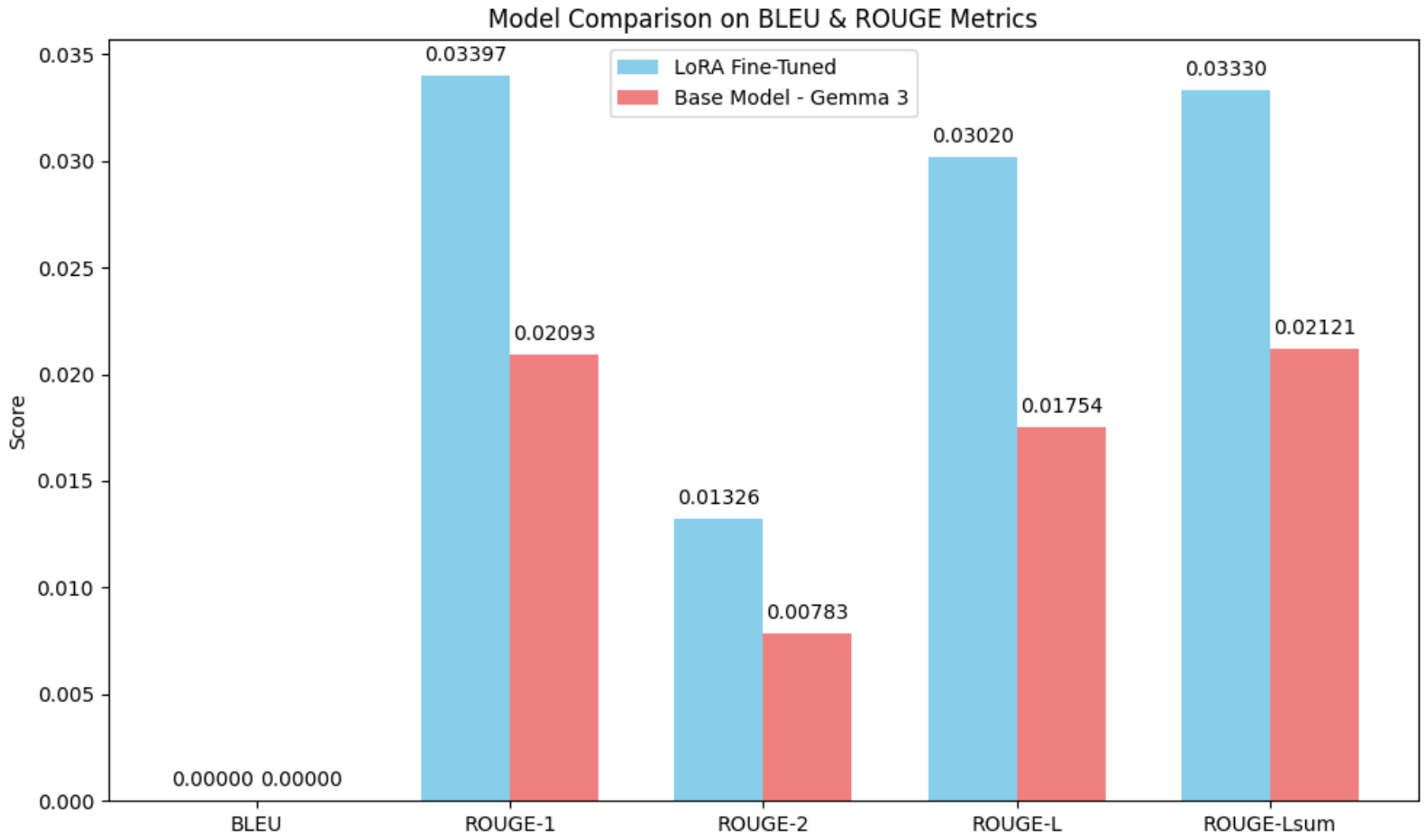
Results on HelpSteer2 Dataset – GPT Neo



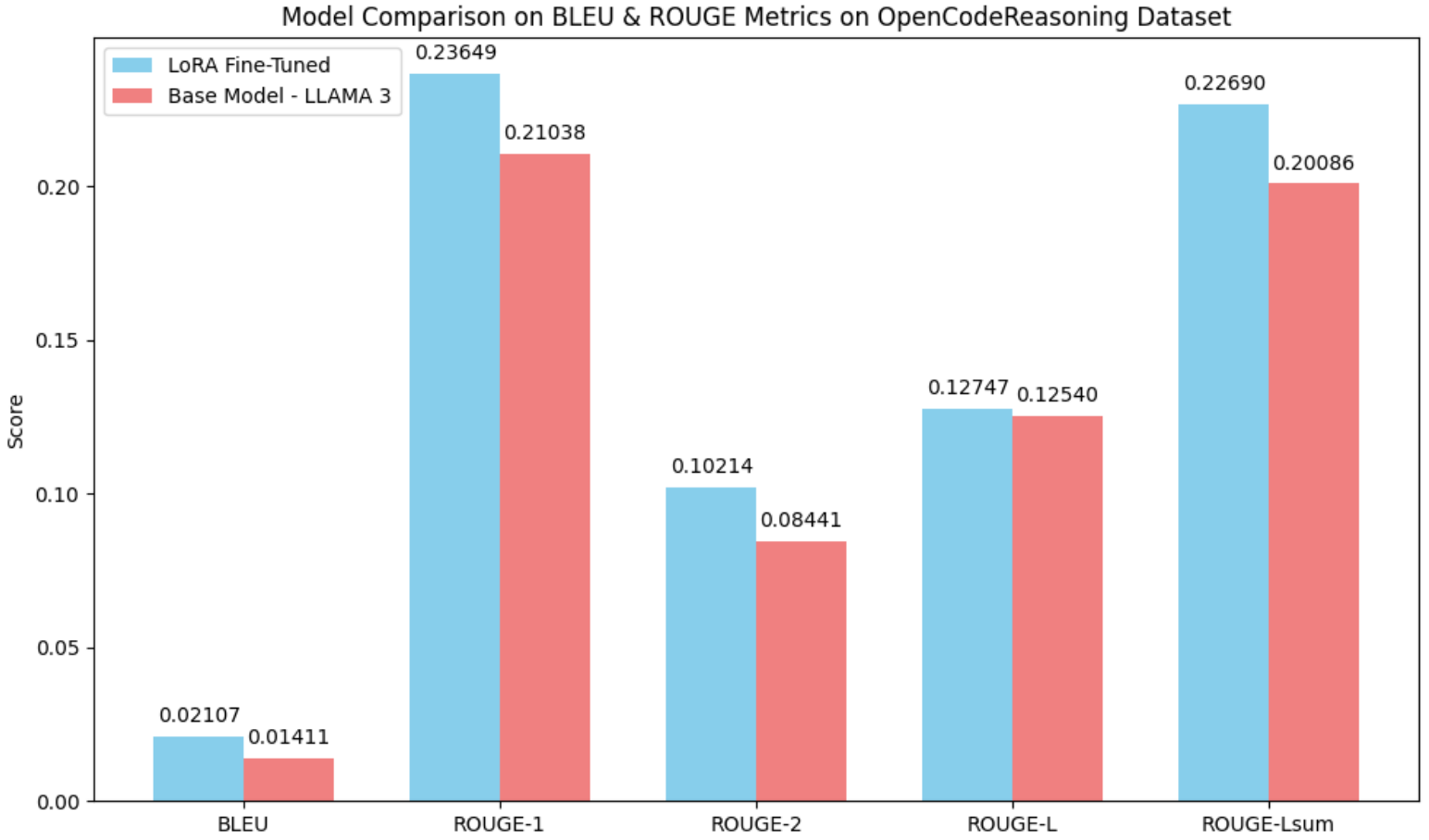
Results on Code Reasoning Dataset – GPT Neo



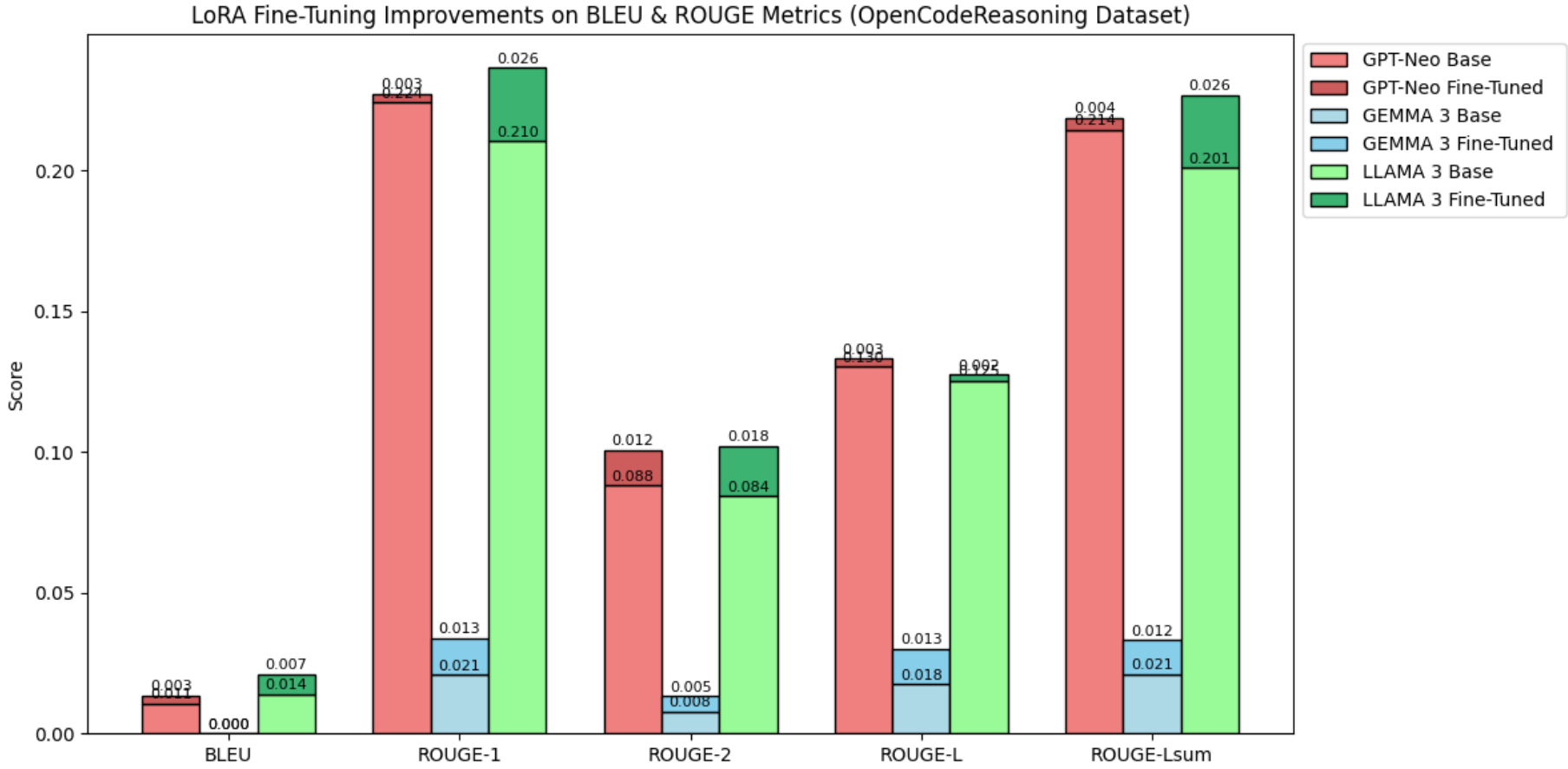
Results on Code Reasoning Dataset – Gemma 3



Results on Code Reasoning Dataset – LLAMA 3



Results All-Together



Conclusion

- We believe that by integrating LoRA's rapid, memory-efficient adapter tuning with β-DPO's dynamic, noise-robust preference optimization, our pipeline conceptually outperforms standalone approaches across efficiency, stability, and alignment metrics—enabling 1B-parameter models to be both domain-specialized and human-aligned on limited compute budgets.

- **LIMITATIONS:**
  - Ran with very few epochs
  - Evaluated on few prompts
- **FUTURE WORK:**
  - Fine-tune it further using DPO

References

[1] NVIDIA, “HelpSteer2,” *Huggingface.co*, 2025. https://huggingface.co/datasets/nvidia/HelpSteer2 (accessed Apr. 22, 2025).
[2] “9 (A)- Automatic evaluation metrics (BLEU, ROUGE, METEOR) - Java Programmatic Universe,” *https://corejava25hours.com/*, Jun. 15, 2024. https://corejava25hours.com/9-a-automatic-evaluation-metrics-bleu-rouge-meteor/
[3] NVIDIA, “OpenCodeReasoning,” *Huggingface.co*, 2025. https://huggingface.co/datasets/nvidia/OpenCodeReasoning (accessed Apr. 22, 2025).
[4] J. Wu *et al.*, “β-DPO: Direct Preference Optimization with Dynamic β,” *arXiv (Cornell University)*, Jul. 2024, doi: https://doi.org/10.48550/arxiv.2407.08639.
[5] E. J. Hu *et al.*, “LoRA: Low-rank adaptation of large language models,” *arXiv (Cornell University)*, Jan. 2021, doi: https://doi.org/10.48550/arxiv.2106.09685.
[6] “Gpt Neo 1.3B · Models · Dataloop,” *Dataloop.ai*, 2023. https://dataloop.ai/library/model/eleutherai_gpt-neo-13b/ (accessed Apr. 22, 2025).
[7] Meta, “Llama 3.2: Revolutionizing edge AI and vision with open, customizable models,” *Meta.com*, 2022. https://ai.meta.com/blog/llama-3-2-connect-2024-vision-edge-mobile-devices/
[8] C. Farabet, “Introducing Gemma 3: The most capable model you can run on a single GPU or TPU,” *Google*, Mar. 12, 2025. https://blog.google/technology/developers/gemma-3/

Multimodal Stress Detection Using Wearable Biosignals

Seeun Kim, Muhammad Taha, and Megan Bechtlofft

Problem Statement

- Chronic stress affects cardiovascular, cognitive, and immune health.
- Need for reliable, real-time stress monitoring using wearable devices.
- Stress detection models often fail when applied to new subjects or real-world conditions.
- **Goal:** Build robust, subject-independent stress detection models using wrist-worn biosignals (EDA, BVP, TEMP, ACC).

Data Overview

Signal	Sampling frequency	Frequency range	Subwindow length	Model inputs
ACC	64 Hz	0–30 Hz	7 seconds	210
BVP	64 Hz	0–7 Hz	30 seconds	210
EDA	64 Hz	0–7 Hz	30 seconds	210
TEMP	64 Hz	0–6 Hz	35 seconds	210

Table 1: Processing details for each signal sampled by the Empatica E4, based on [5].

- Dataset: WESAD (Wearable Stress and Affect Detection) - 15 subjects
- Device: Empatica E4(wrist-worn)
- Signals used: EDA (4Hz), BVP (64Hz), TEMP (4Hz), ACC (32Hz)
- Task: Binary classification (Stress vs. Non-stress)
- Data Processing: Resampling to 64 Hz, Normalization and FFT
- Evaluation Setup:
  - Transformer, CNN → Leave-One-Subject-Out (LOSO) Cross-Validation
  - Random Forest, XGBoost → 80/20 Train-Test Split + 5-Fold Cross-Validation

Preprocessing Pipeline

- Data Acquisition (WESAD wrist & chest sensors)
- Signal Resampling & Alignment to 64 Hz
- Timestamping & Label Integration (forward-fill)
- Label Cleaning & Binarization (stress vs. non-stress)
- Min–Max Normalization per channel
- 60-s Windowing (discard <30 s)
- Overlapping Sub-windows (ACC 7 s, BVP/EDA 30 s, TEMP 35 s, 0.25 s stride)
- Frequency-Domain FFT Features (210 per channel)
- Final Subject-wise Dataset for LOSO-CV

Transformer

- **Architecture**
- 8 stacked Transformer encoder blocks (LayerNorm ➜ Multi-Head Self-Attention ➜ Dropout ➜ 1-D Conv feed-forward)
- 4 attention heads × 256 units per head
- Global average pooling ➜ MLP with dropout + sigmoid for binary stress prediction

- **Regularization & Hyper-params**
- Uniform dropout 0.25 in both Transformer blocks and MLP
- Adam optimizer, learning rate 1×10^{-4}
- Class-weighted loss to counter stress / non-stress imbalance

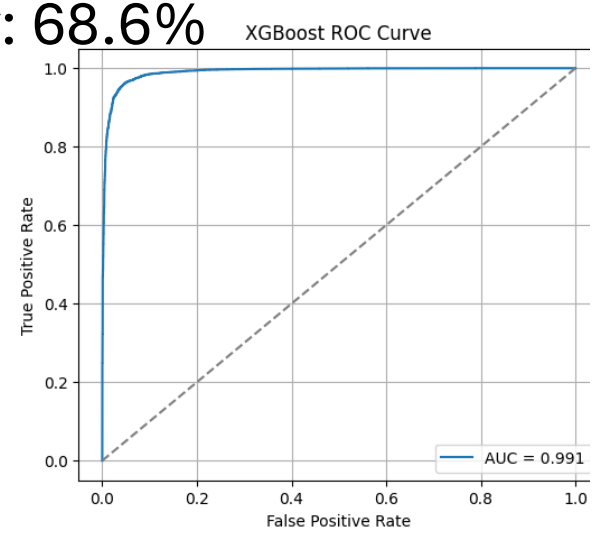
- **Training Strategy**
- Early stopping (patience 12, min $\Delta = 0.01$ on val loss)
- ReduceLROnPlateau (halve LR after 5 epochs without improvement)

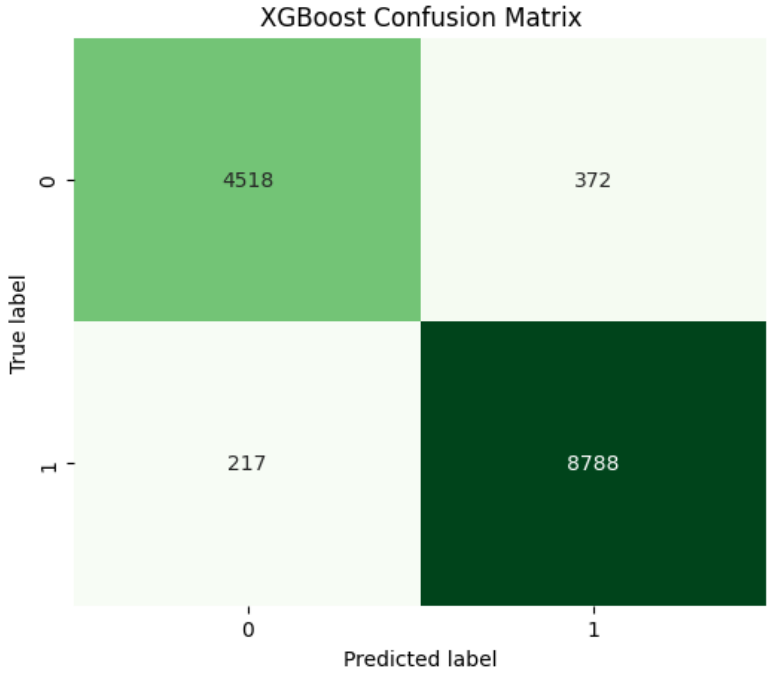
Block	Layer	Settings	Data shape
Input	Input layer	N/A	(6, 210)
Encoder (x8)	Layer normalization	epsilon 1e-6	(6, 210)
	Multi head attention	256 head size, 4 heads	(6, 210)
	Dropout	0.25	(6, 210)
	Add	residual connection	(6, 210)
	Layer normalization	epsilon 1e-6	(6, 210)
	Convolution 1D	4 filters of 1x1, ReLU	(6, 4)
	Dropout	0.25	(6, 4)
	Convolution 1D	6 filters of 1x1, linear	(6, 210)
	Add	residual connection	(6, 210)
Output	Avg-Pooling 1D	reduce feature space	(6)
	Fully connected	128 units, ReLU	(128)
	Dropout	0.25	(128)
	Fully connected	2 units, sigmoid	(2)

Table 2: Model architecture and data shape for each layer of our transformer model

XGBoost

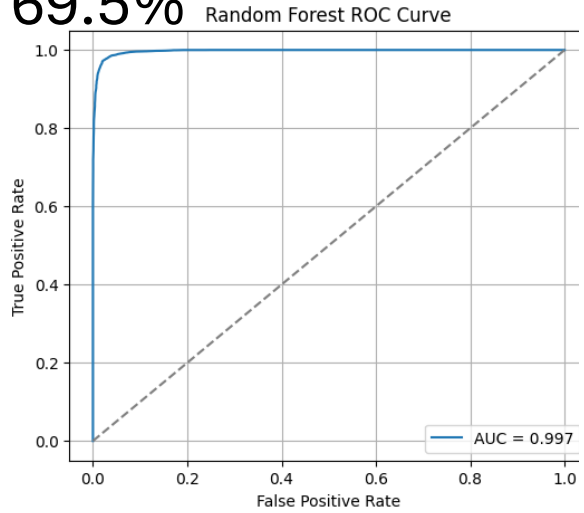
- Input: Flattened multimodal wrist signals (ACC, EDA, TEMP, BVP)
- Preprocessing: Z-score normalization, 160-sample windowing
- Evaluation: 80/20 stratified split + 5-fold cross-validation
- Results:
  - Multimodal Accuracy: 95.76%
  - Unimodal (EDA-only) Accuracy: 68.6%

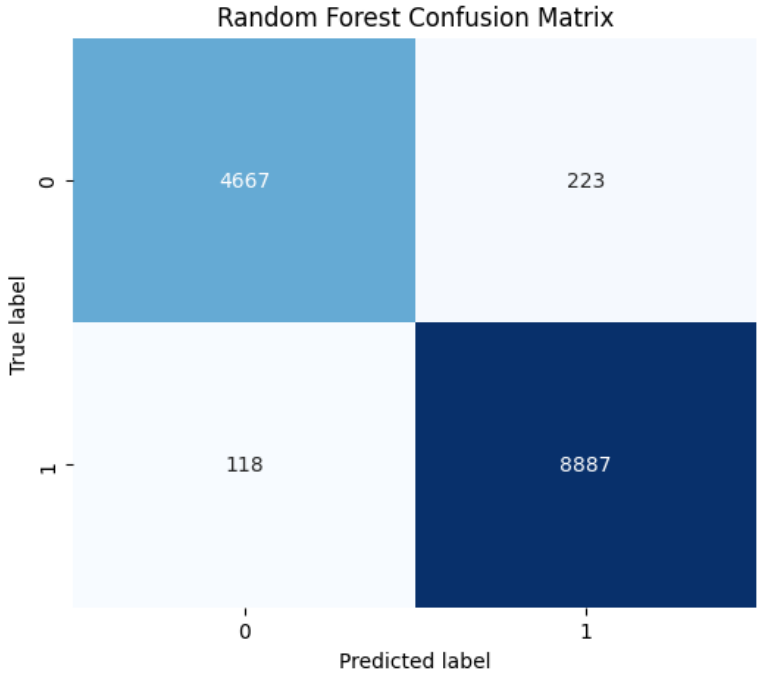




Random Forest

- Input: Same multimodal features as XGBoost
- Preprocessing: Z-score normalization, 160-sample windowing
- Evaluation: 80/20 stratified split + 5-fold cross-validation
- Results:
  - Multimodal Accuracy: 97.55%
  - Unimodal (EDA-only) Accuracy: 69.5%





CNN

Architecture:

- 3 convolutional blocks with increasing filter sizes (64, 128, 256)

Input:

- Wrist-based signals: ACC, EDA, BVP, TEMP
- Frequency-domain features (via FFT)

Training Details:

- Leave-One-Subject-Out (LOSO) cross-validation
- Class weighting, early stopping, learning rate scheduling

Regularization:

- Dropout = 0.5 to prevent overfitting

Multimodal Results:

Model	Accuracy	AUC	Precision	Recall	F1
CNN	0.981	0.998	0.981	0.981	0.981

Unimodal Results:

Model	Accuracy	AUC	Precision	Recall	F1
CNN	0.652	0.672	0.643	0.652	0.646

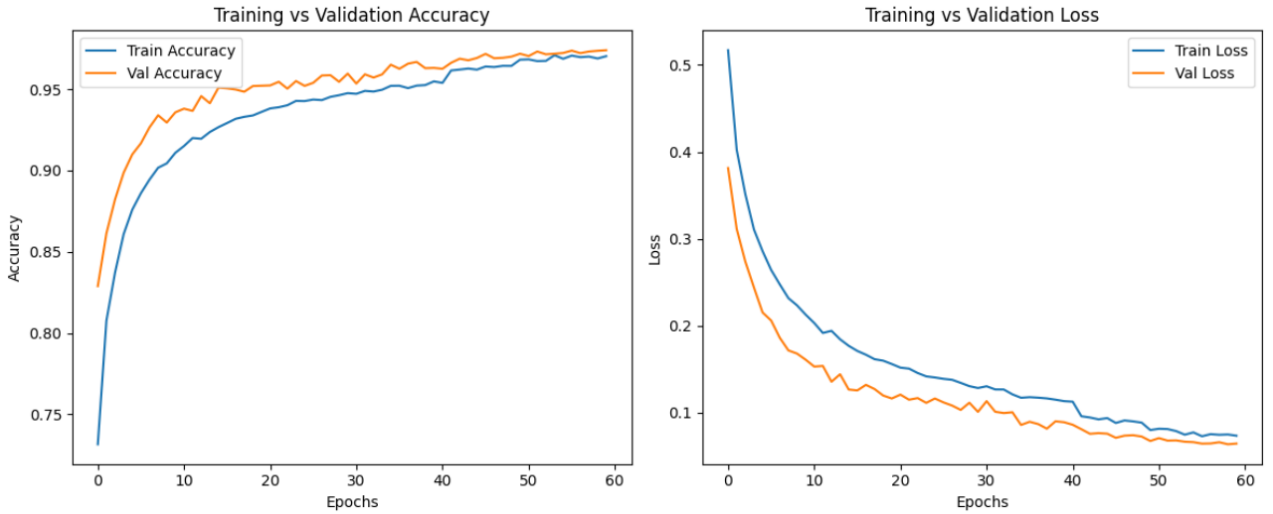


Figure 1. CNN multimodal accuracy and loss comparisons

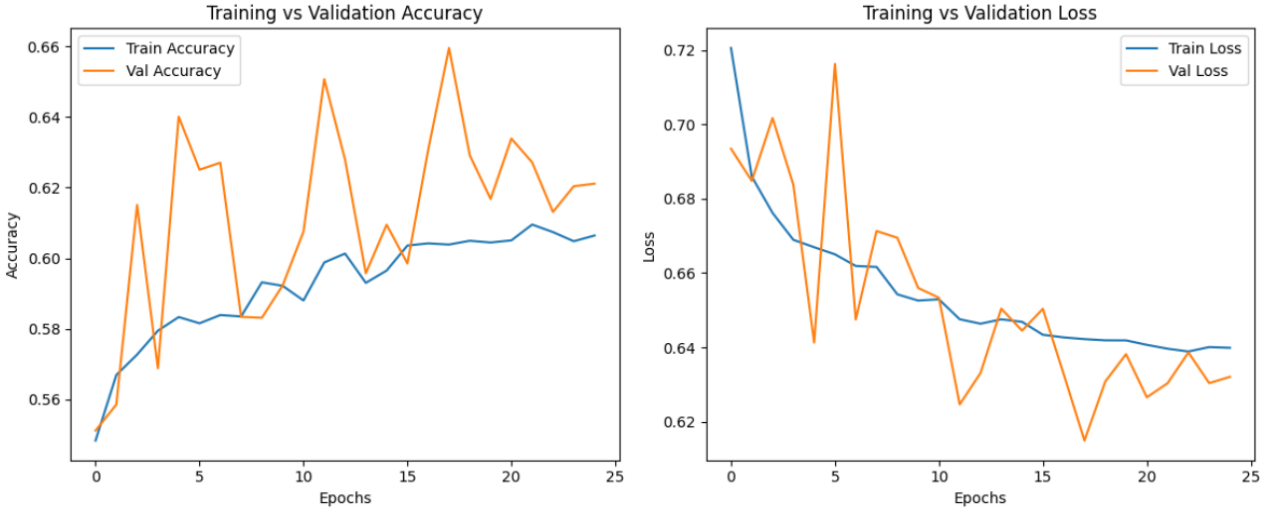


Figure 2. CNN unimodal accuracy and loss comparisons

Comparative Insights - Multimodality

Table 3: Performance comparison of models using multimodal wrist signals. Highest accuracy highlighted in red.

Model	Accuracy	AUC	Precision	Recall	F1
CNN	0.981	0.998	0.981	0.981	0.981
Transformer	0.942	0.971	0.950	0.940	0.940
XGBoost	0.958	0.991	0.954	0.959	0.957
Random Forest	0.976	0.997	0.975	0.975	0.975

- CNN achieves best overall performance, highly balanced
- Traditional models (XGBoost/RF) performed strong
- Transformer is competitive but more sensitive to input diversity

Comparative Insights - Unimodality

Table 4: Performance comparison of models using unimodal wrist signals (EDA only). Highest accuracy highlighted in red.

Model	Accuracy	AUC	Precision	Recall	F1
CNN	0.652	0.672	0.643	0.652	0.646
Transformer	0.590	0.917	0.830	0.590	0.590
XGBoost	0.686	0.683	0.670	0.686	0.651
Random Forest	0.695	0.705	0.681	0.695	0.677

- Only EDA signal data was used in this unimodal evaluation
- All models dropped 25-35% in accuracy compared to multimodal
- Transformer had highest AUC but lowest accuracy
- Random Forest performed best under unimodal constraints

Conclusion

- The CNN model achieved the best overall performance, with 98.13% test accuracy and an AUC of 0.9982.
- Models trained on a combination of signals significantly outperformed those trained on EDA alone.
- Random Forest and XGBoost had high accuracy with greater interpretability and faster training times.
- Effective preprocessing, including signal resampling, normalization, FFT feature extraction, and class balancing, is critical across all model classes.

Future Work

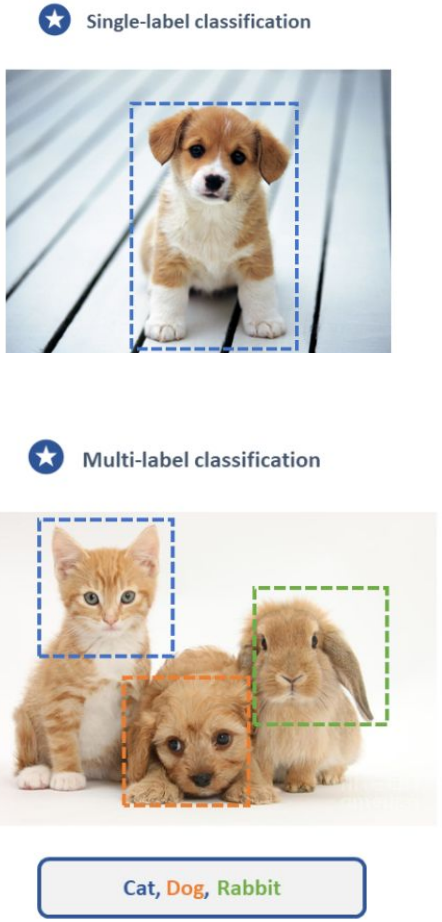
- Cross-dataset validation (ex. SWELL-KW)
- Real-time model deployment on wearables
- Explore cross-modal attention and self-supervised learning
- Improve model interpretability

Multilabel Classification with Single-Labeled Data

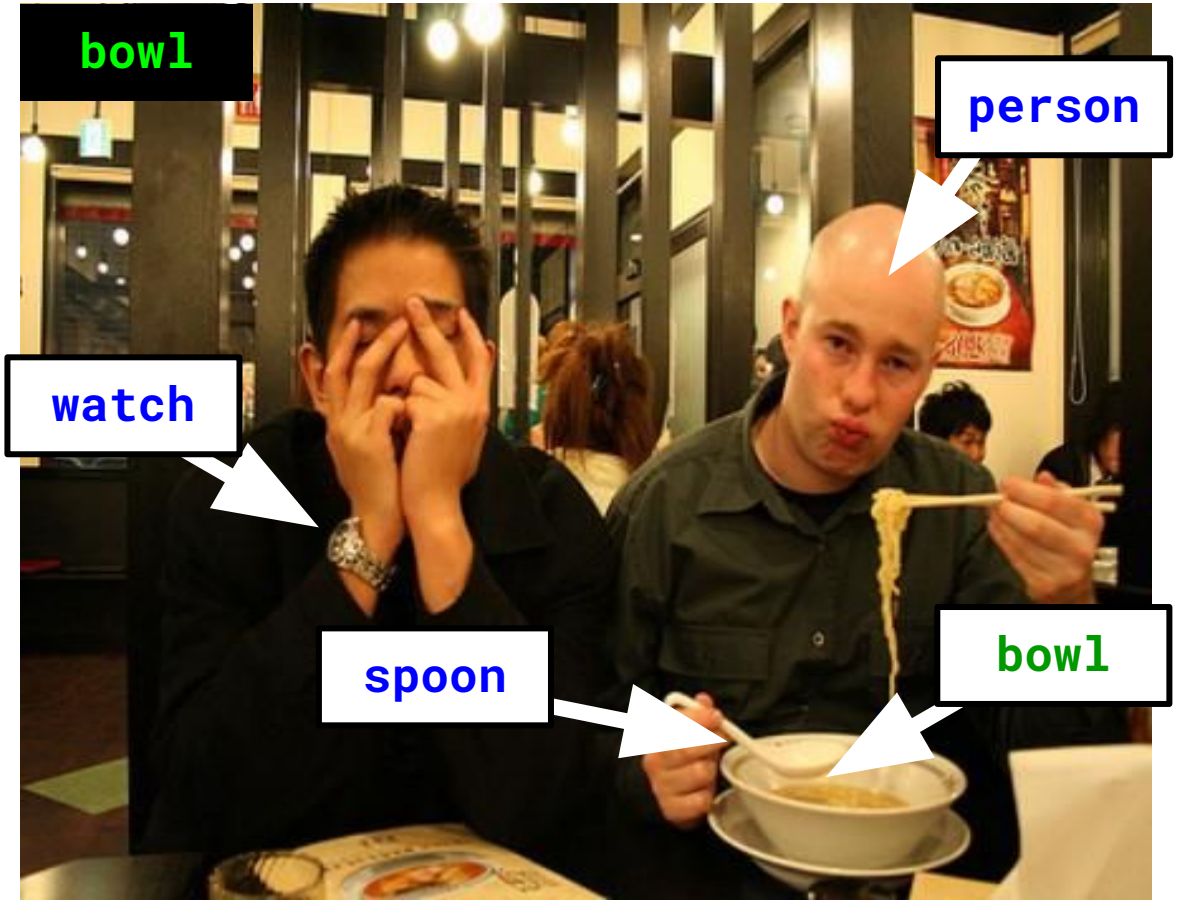
Muyang Yan, Vincent Wang, Andy Niu

Image classification

- Image classification is an extremely widely studied problem with myriad real-world applications
  - E.g. medical diagnosis, autonomous vehicles, search and rescue, wildfire identification from satellite imagery
  - Is this image a dog?
- One variant of image classification is *multi-label image classification* (e.g. find *all* possible labels, not just the most likely one)

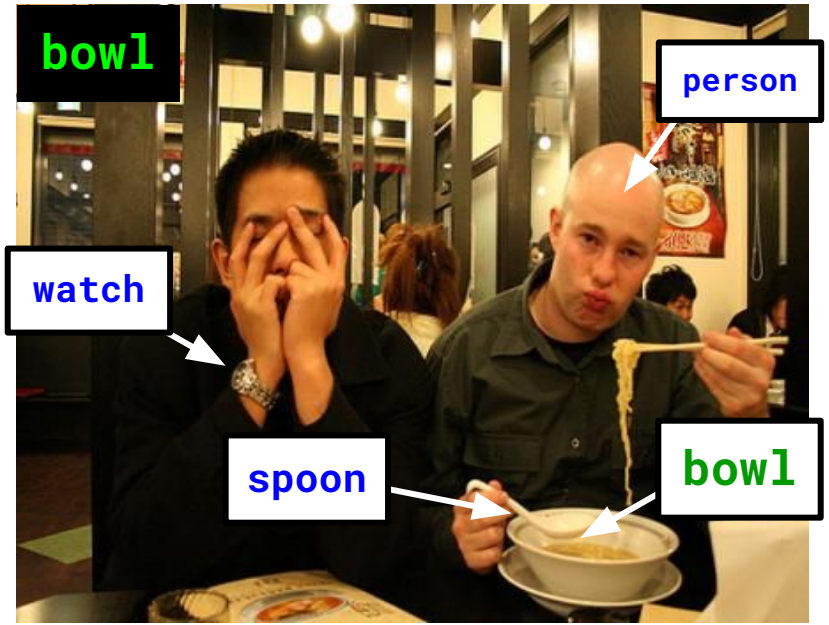


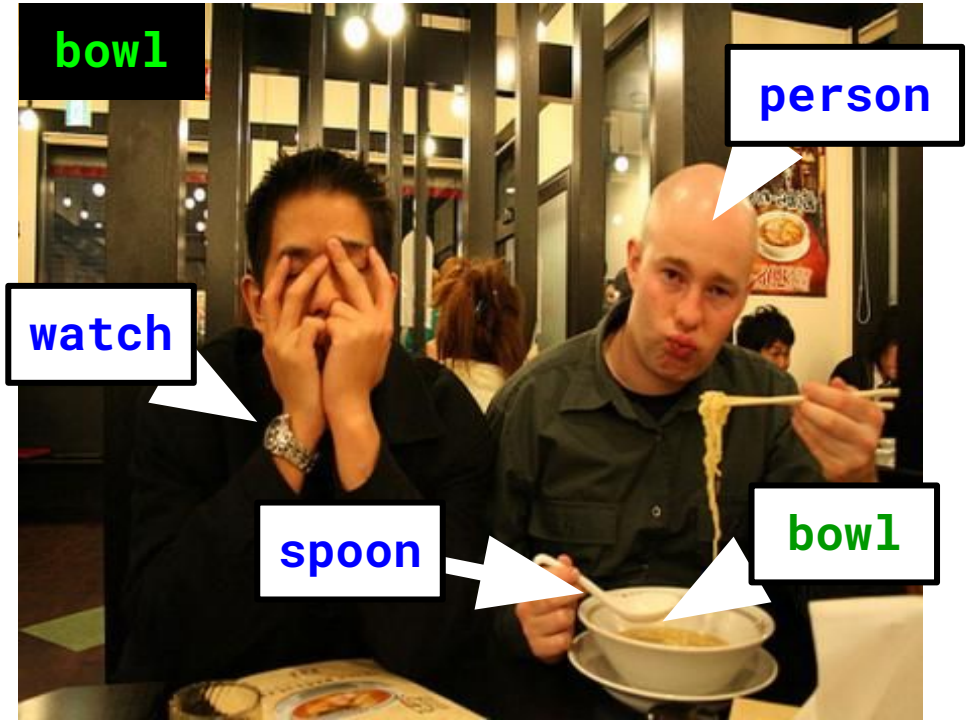

bowl
bowl


bowl
person
watch
spoon
bowl

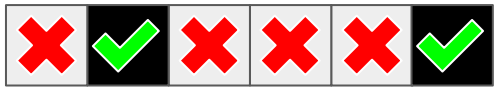
Single-Positive Multi-Label setting (SPML)

- Many good datasets exist with single-labeled images
- Not as many good datasets exist with multi-labeled images
- **Problem statement: Given only the single labels, can we correctly classify multiple labels for given images?**
  - **Use cases: any multi-label use case!**

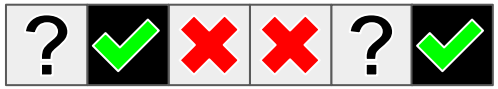




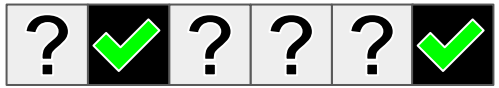
Fully supervised



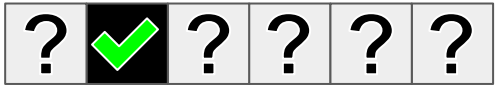
Partially supervised



Positive-unlabeled (PU)



SPML (Our setting)



Past work - ROLE

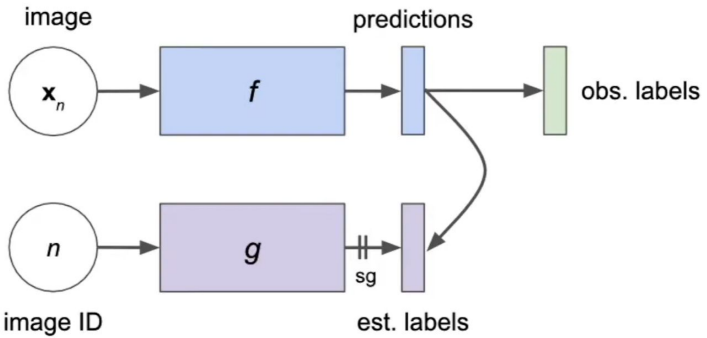
Multi-Label Learning from Single Positive Labels, Cole et al. (2021)

Expected Positive Regularization:

$$L_{EPR}(F_B, Y_B) = \frac{1}{|B|} \sum_{n \in B} L_{IU}(f_n, y_n) + \lambda \left(\frac{\hat{k}(F_B) - k}{L} \right)^2$$

+ Label Estimator Matrix

- Showed that it was possible to get close to fully labeled accuracy using only single positive labels

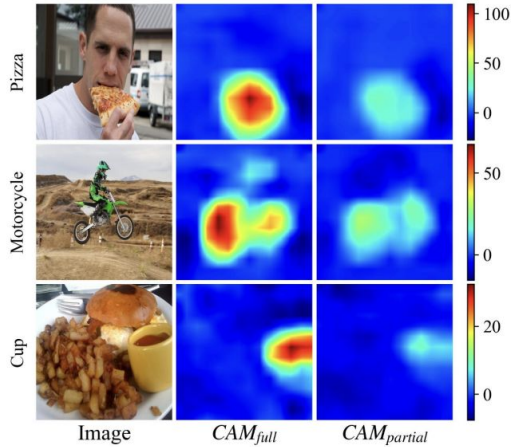


Class Activation Mapping (CAM) / Explainability

Bridging the Gap between Model Explanations in Partially Annotated Multi-label Classification, Kim et al. 2023

IGNORE: Information Gap-based False Negative Loss Rejection for Single Positive Multi-Label Learning, Song et. al. 2024

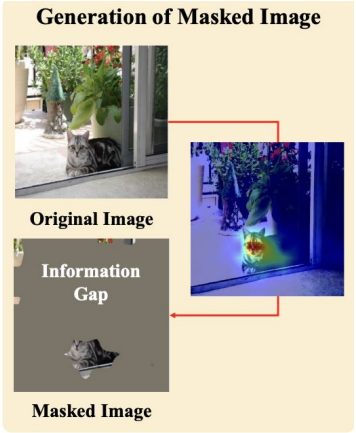
Leverage CAM *explanations* of classification to improve the model (e.g. by boosting or masking)



Class	Ground Truth	Given Labels	Labels w/ AN
Cat	1	1	1
Potted Plant	1	u	0
Train	0	u	0
Dining Table	0	u	0
Bus	0	u	0
Dog	0	u	0

Assumed Negative Labels

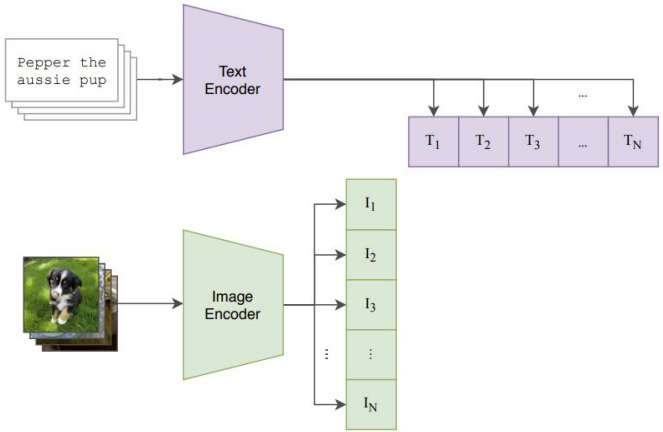
1: Positive Label u: Unannotated Label
0: False Negative u: Unannotated Positive
0: True Negative



Outside Data Sources / Foundation Models

CLIP (2021), SegmentAnything (2023)

- Trained on a large corpus of data from the internet
- Capable of zero-shot generalization
- Our goal is to evaluate the *SPML* task - using foundation model obfuscates where improvements actually come from, so we ignore these approaches



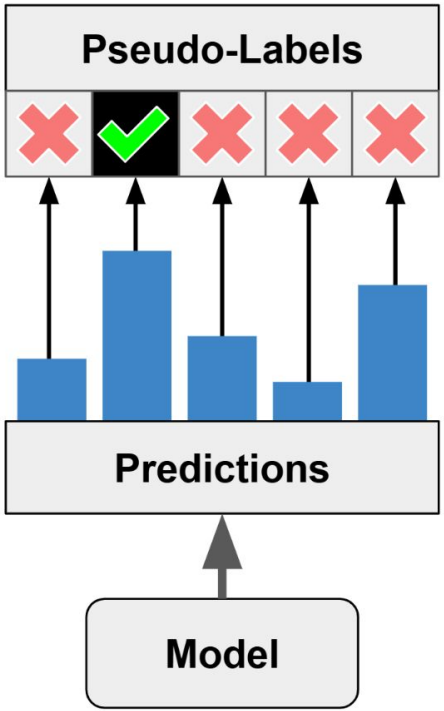
Our Methods

Scheduled **I**ncre**m**ental **P**seudo-**L**ab**e**ling (**SIMPLE**)

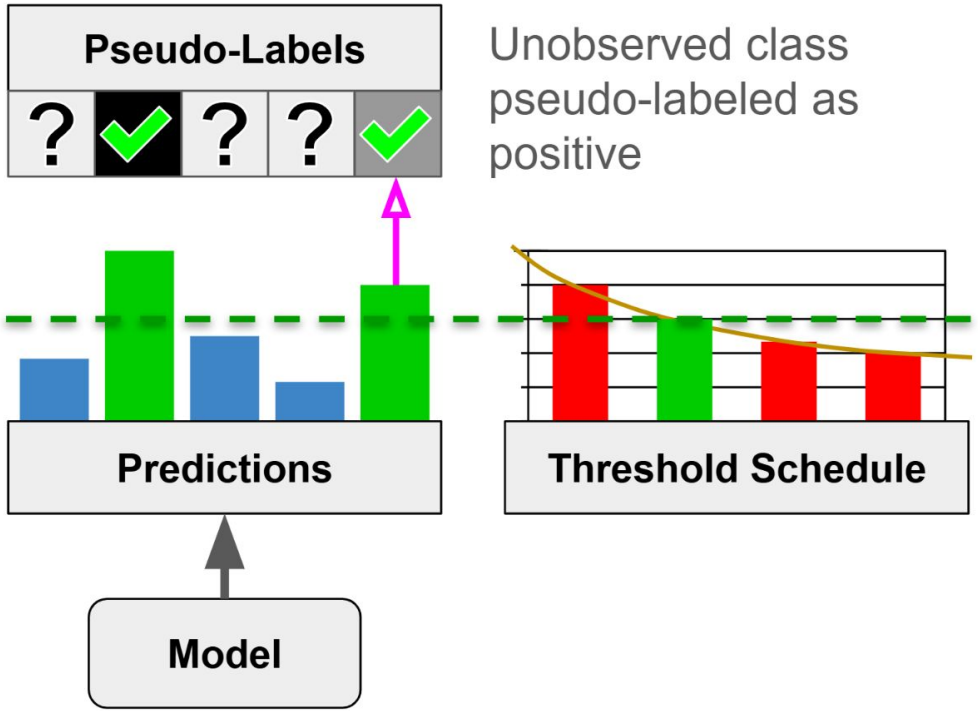
- Use a *threshold schedule* to add positive pseudo-labels
- Start with high threshold (have to be very confident), gradually lower

Scheduled **I**ncremental **P**seudo-**L**ab**e**ling (**SIMPLE**)

1) Supervision

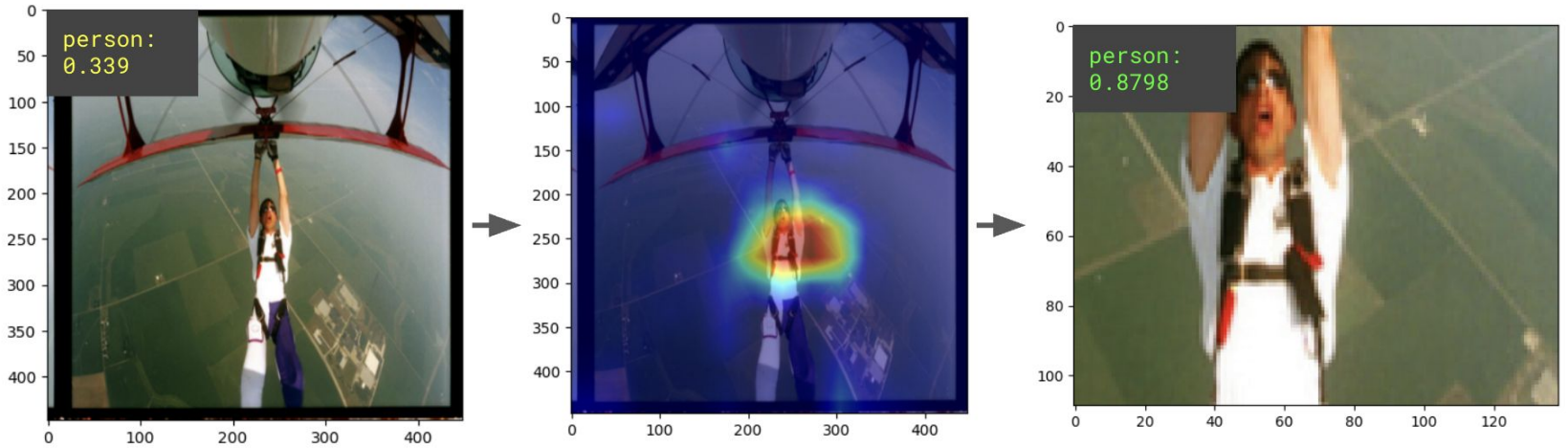


2) Updating pseudo-labels

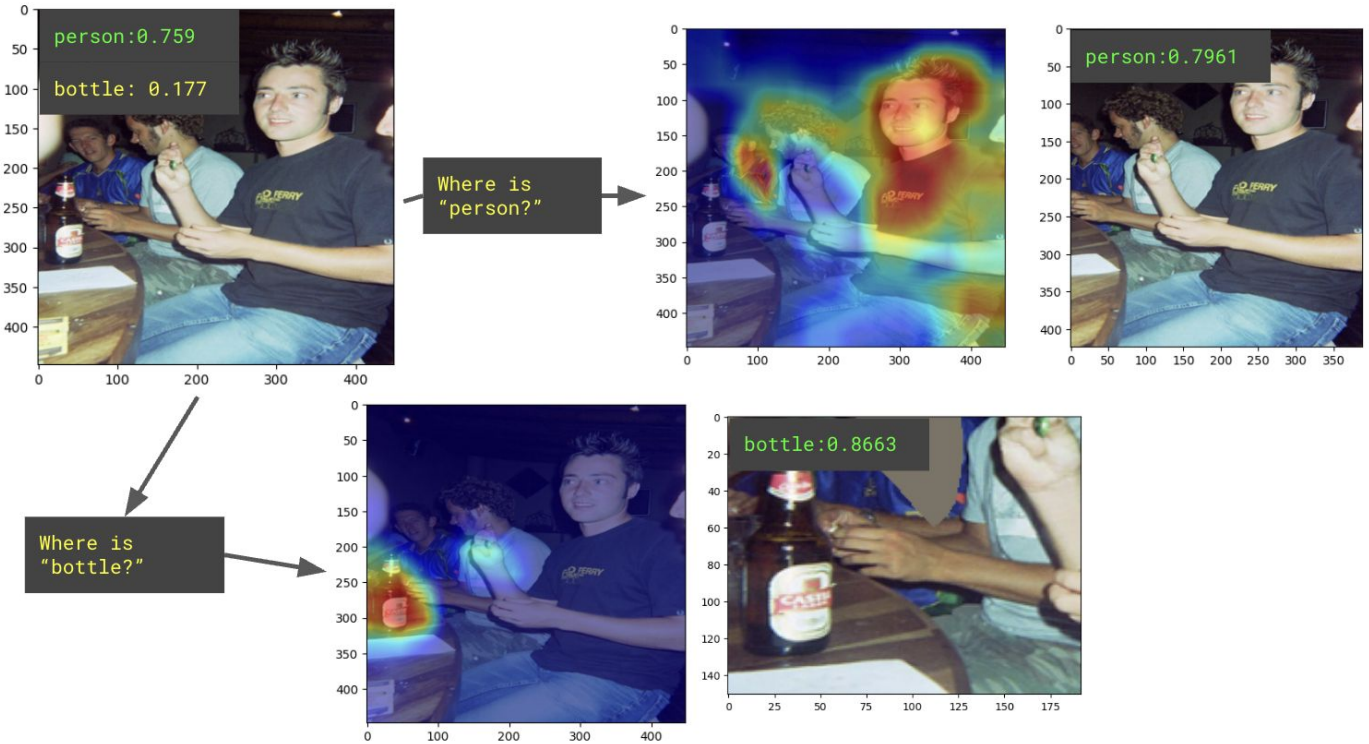


CAMCrop (Class Activation Mapping Crop)

- *Inference-time* strategy
- Leverage *class activation maps* to explain “where” objects are
- Crop to relevant portion of image, reclassify, boost probability of confirmed classes



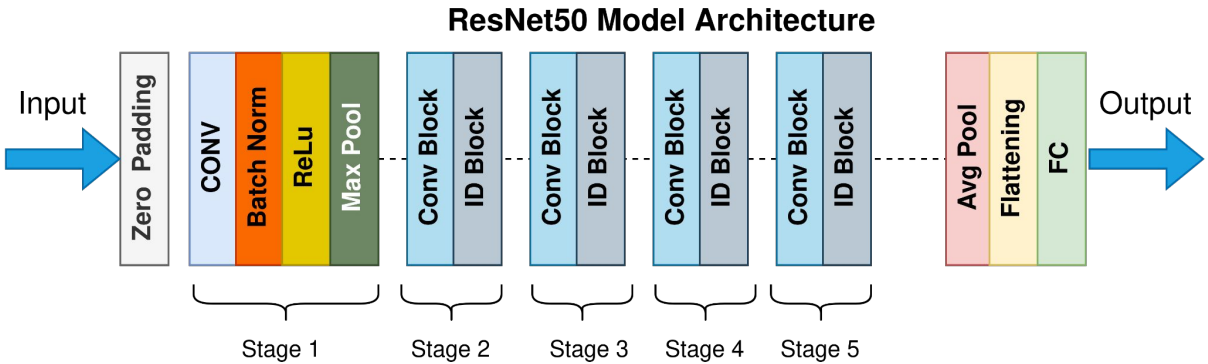
CAMCrop (Class Activation Mapping Crop)



Experiments

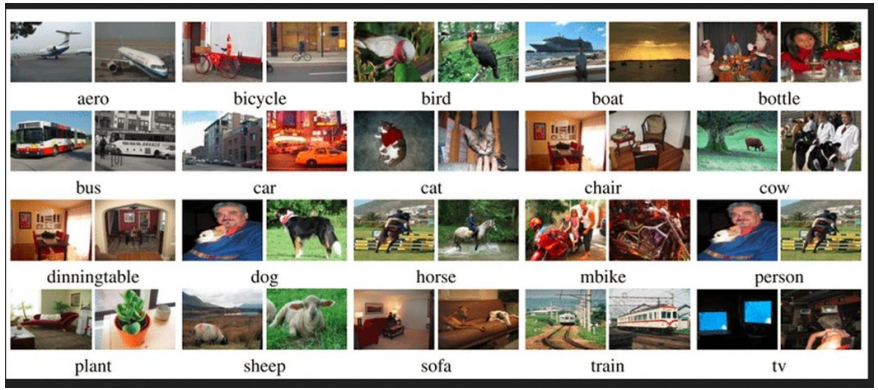
Evaluation

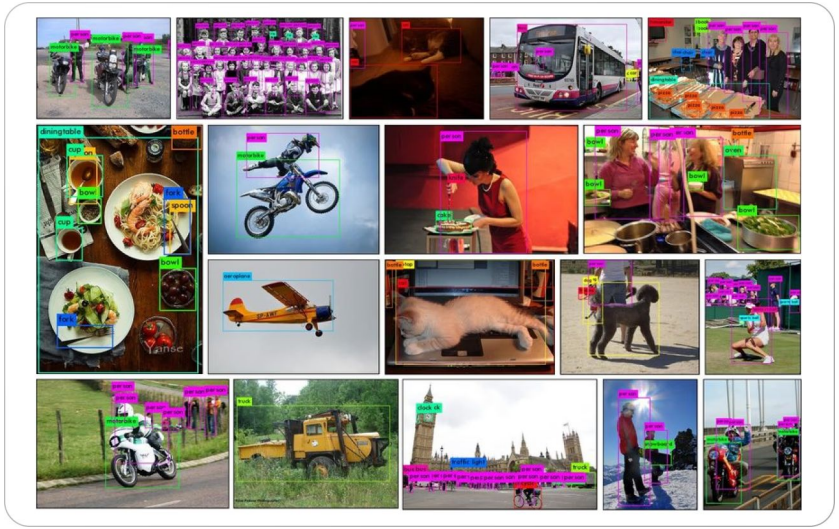
- Evaluate our methods **SIMPLE** and **CAMCrop** against:
  - Assume Negative
  - Assume Negative with Label Smoothing
  - ROLE (2021)
  - BCE (fully-labeled data)
- All methods use ResNet-50 pretrained on ImageNet, final FC layer fine-tuned on individual datasets for 10 epochs



Datasets

- COCO (2014, 80 classes)
- Pascal-VOC-2012 (20 classes)
- Both well-known, highly used datasets
- Both datasets are multi-label by default; use method described in ROLE paper to extract single positive labels





Results

Method	Labels Used	VOC2012	COCO
BCE	All Pos. + All Neg.	86.7*	70.0*
BCE+LS	All Pos. + All Neg.	87.6*	70.2*
IU	1 Pos. + 1 Neg.	82.6*	60.8*
AN	1 Pos. Only	83.7	60.8
AN+LS	1 Pos. Only	85.1	63.7
SIMPLE	1 Pos. Only	84.8	64.2
SIMPLE + LS	1 Pos. Only	86.1	64.2
ROLE	1 Pos. Only	86.5*	66.3*
AN+CAMCrop	1 Pos. Only	87.9	76.9† (61.2)
AN+LS+CAMCrop	1 Pos. Only	**88.9**	77.6† (64.2)
SIMPLE+CAMCrop	1 Pos. Only	87.3	75.9† (64.9)
SIMPLE+LS+CAMCrop	1 Pos. Only	**88.9**	**76.2**† (64.72)

SIMPLE Results

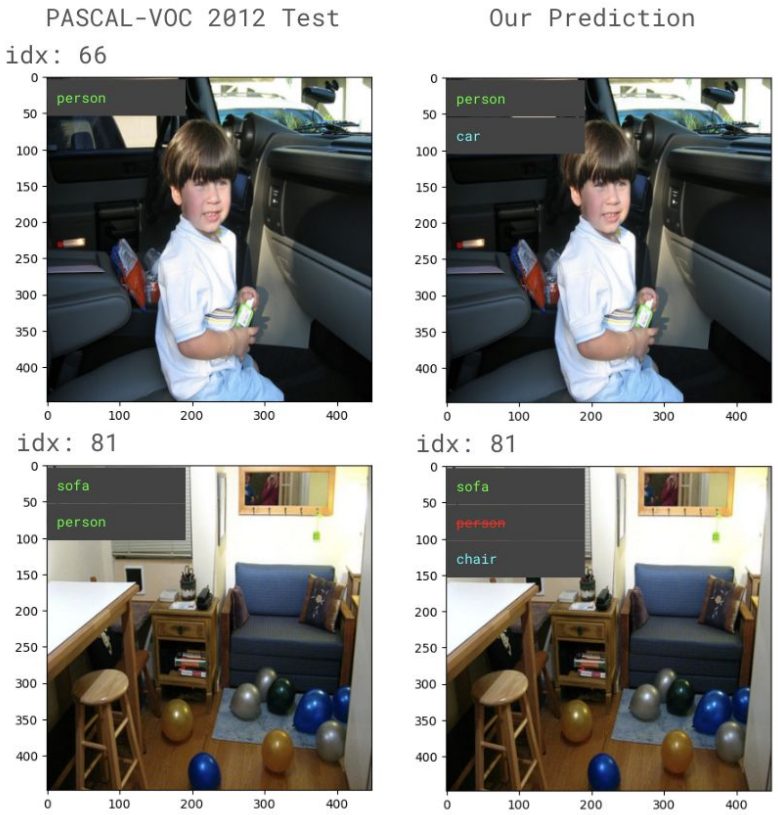
- Significantly outperforms AN baseline!
- Performance relatively close to ROLE
- Previous Works mostly considered adding negative pseudo-labels
  - Adding positive labels theorized to add noise
- SIMPLE shows that positive pseudo-labeling w/ schedule increases predictive power, despite potential for label noise

CAMCrop Results

- Outperforms AN baseline!
  - Massive improvement over existing methods on COCO
- Reveals a variety of other interesting results

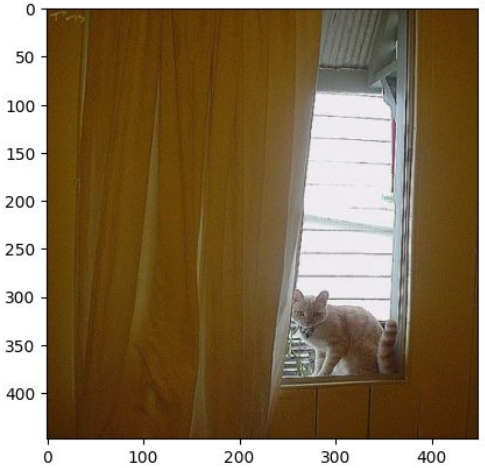
CAMCrop Results - Image Label Robustness

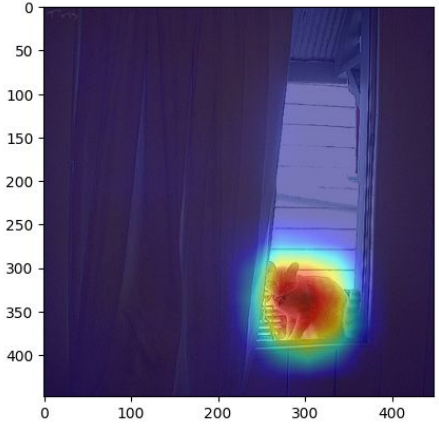
- In some cases, PASCAL-VOC 2012 is **missing labels**
- **CAMCrop recovers** those labels, even though they are not in ground truth
- Suggests SPML methods can act as “check” during human labeling process

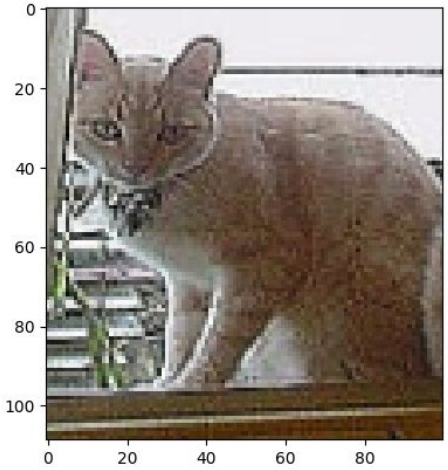


CAMCrop Results - Localization

- **Localization** refers to finding **where** an object is in the image
- Despite no location information, using simple masking strategies on CAM heatmaps are generally effective at finding objects







CAMCrop Results - Improved Multi-Object Classification

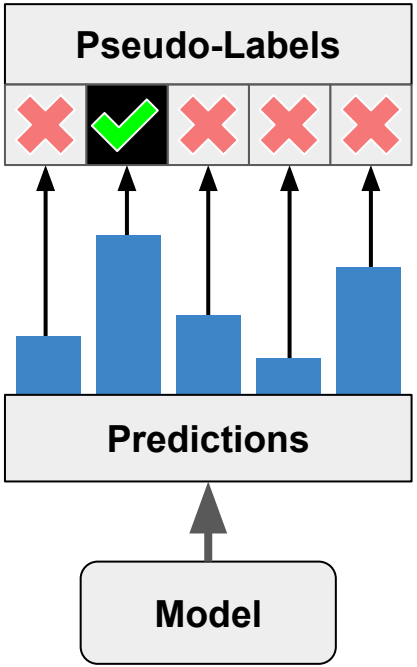
- **Outperforms BCE (fully-observed) upper-bound on COCO**
- Theory: Cropping/focus strategy allows classifier to focus on one object at a time

Conclusion

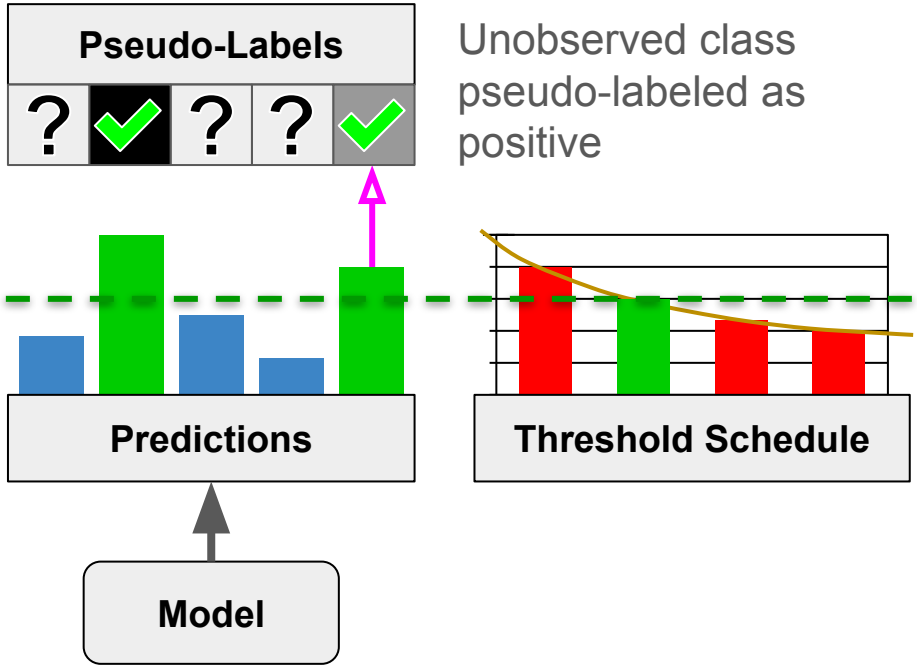
- Learning multi-label classification from single labels is a difficult, but possible, task
- Positive pseudo-labeling can increase the predictive ability of an SPML model
- Leveraging explainability tools such as CAM can recover labels not present in original dataset, suggesting robustness to label noise
- SPML is a worthwhile research direction that can yield applications in a wide variety of fields

Thank you!

1) Supervision



2) Updating pseudo-labels



pdating pseudo-labels

Pseudo-Labels

Unobserved class