Extending AuthorMist for Cross-Detector AI Text Evasion

Sudeshna Merugu **Prith Sharma** **Miguel Almeida**
{sudeshnm, priths, malmeida}@andrew.cmu.edu
CMU Advanced NLP (11-711) – Assignment 4
April 2025

Abstract

We propose to fine-tune the AuthorMist model (David and Gervais, 2025), a recent approach that leverages reinforcement learning to generate human-like paraphrases capable of evading AI text detectors. As AI-generated content becomes increasingly prevalent, so does the need for accurate detection and, in parallel, a better understanding of evasion strategies to avoid exploitation of ethical use of AI-generated text. AuthorMist represents a strong baseline in this space, introducing Reinforcement Learning using Group Relative Policy Optimization (GRPO) to fine-tune generation toward undetectability while attempting to preserve semantic meaning. In this work, we aim to improve the generalization and robustness of AuthorMist by fine-tuning it on the CheckGPT dataset on a set of Open Source detectors, which contains diverse examples designed to test the limits of AI-generated text detectors. By training on this more challenging and detector-annotated corpus, we intend to encourage AuthorMist to develop more detector-agnostic evasion strategies and better semantic preservation across varied inputs.

1 Introduction

In the era of increasingly sophisticated AI-generated text, automatic detection tools have emerged to differentiate machine-generated from human-authored content. While these detectors aim to uphold authenticity, they simultaneously pose significant risks to author privacy and creative freedom. AI-assisted writing tools, widely adopted for their utility in improving productivity and accessibility, risk unfair scrutiny as their output may inadvertently trigger false positives, leading to undue suspicion or penalties.

Consequently, developing effective evasion techniques becomes not only a technical necessity but also a safeguard for authors using AI assistance ethically. The AuthorMist framework exemplifies recent advancements by leveraging reinforcement learning to paraphrase text, thereby effectively reducing detectability. Despite its success, AuthorMist faces challenges, particularly regarding its ability to generalize across unseen detectors and maintain stylistic coherence. Addressing these limitations is crucial for practical, real-world applicability, where content creators cannot reliably anticipate which detection systems will evaluate their texts.

2 Related Work

Early AI-generated text detectors leveraged lexical and statistical signals. GLTR visualizes token probabilities to flag synthetic text (Gehrmann et al., 2019), and Grover trains supervised classifiers on large synthetic corpora (Zellers et al., 2019). Such approaches, however, falter as language models become more fluent: Ippolito et al. showed that detectors trained on one model often fail on newer generations (Ippolito et al., 2020). Zero-shot methods like DetectGPT estimate likelihood curvature under small perturbations for improved generalization (Mitchell et al., 2023), and watermarking techniques embed subtle generation-time signals into text (Kirchenbauer et al., 2023). Yet both remain vulnerable to high-quality rewriting, motivating exploration of paraphrase-based evasion.

Paraphrasing attacks have emerged as a particularly effective evasion strategy. Krishna et al. and Sadasivan et al. demonstrate that meaning-preserving rewrites drastically reduce detector confidence (Krishna et al., 2023; Sadasivan et al., 2023), while orthographic perturbations further foil token-level detectors (Creo and Pudasaini, 2024). Adversarial training methods such as DAP (Schneider et al., 2025) and stochastic detector feedback (da Silva Gameiro et al., 2023) similarly exploit multi-detector signals to harden para-

phrasers against detection.

AuthorMist advances this line of work by framing paraphrasing as a reinforcement learning problem, introducing Group Relative Policy Optimization (GRPO) to balance fluency and undetectability across multiple detectors (David and Gervais, 2025). By sampling paraphrases in groups and optimizing relative rewards, AuthorMist achieves stronger evasion while preserving semantic integrity, outperforming simpler RL baselines.

Recent studies emphasize the need for cross-detector and cross-style robustness. Liang et al. reveal significant biases against non-native and informal text (Liang et al., 2023), and the HC3 benchmark highlights detector degradation in various registers and languages (Guo et al., 2023). Frameworks like CUDRT evaluate detectors according to the type and style of editing (Tao et al., 2024), and architectures such as BiScope's bidirectional token mismatch (Guo et al., 2024) and Text Fluoroscopy's hidden layer divergence (Yu et al., 2024) expose further weaknesses. Zero-shot systems—Binoculars (Hans et al., 2024), Fast-DetectGPT (Bao et al., 2023), and the PHD intrinsic-dimension method (Tulchinskii et al., 2023)—each adopt distinct detection signals, underscoring the diversity of real-world detectors.

Beyond technical benchmarks, real-world impacts drive the need for robust evasion analysis. Corpus-level studies estimate that up to 17% of peer reviews contain AI-generated text (Liang et al., 2024), and tools like CycleResearcher illustrate iterative human–AI editing workflows (Weng et al., 2024). Russo et al. show how AI assistance reshapes scholarly communication, raising concerns about overzealous detection in mixed-authorship settings (Russo Latona et al., 2024). Our work builds on these insights, aiming to produce paraphrases that generalize across unseen detectors and preserve both meaning and style in diverse contexts.

3 Our Method: Fine-Tuning AuthorMist

3.1 Baseline Model Details

3.1.1 System Architecture

AuthorMist is designed to transform AI-generated text into human-like text that can evade AI text detectors. The system consists of two main components:

- A base language model (Qwen2.5-3B Instruct) that serves as the paraphrasing policy
- A reinforcement learning framework that optimizes this policy using detector feedback

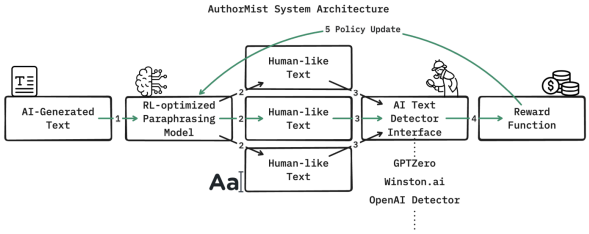

Figure 1: AuthorMist system architecture: AI-generated text is paraphrased via an RL-optimized model to reduce detector scores while preserving meaning and fluency.

The architecture follows a pipeline where AI-generated text is fed into an RL-optimized paraphrasing model that transforms it to minimize detection probability.

3.1.2 Reward Modelling

AuthorMist incorporates multiple AI text detectors (both commercial and open-source) to provide robust feedback. The system is trained against diverse detection algorithms to avoid overfitting to a single detector's weaknesses. The detector selection follows key principles:

- Diversity of detection approaches (statistical detectors, neural classifiers, hybrid systems)
- Representativeness of real-world detection systems
- API stability and reliability
- Continuous probability scores rather than binary classifications

3.1.3 Reward Function

The reward function quantitatively measures AuthorMist's success in evading AI-generated text detection. Given a set of detectors $D = \{d_1, d_2, \ldots, d_k\}$ each detector d_j outputs a probability score $P_{d_j}(Y)$ indicating the likelihood of text Y being AI-generated. The reward function is defined as:

$$R(X, Y) = 1 - \frac{1}{k} \sum_{j=1}^{k} P_{d_j}(Y)$$

This formula rewards outputs that are classified as more human-like (lower probability of being AI-generated).

3.1.4 RL Training with GRPO

AuthorMist employs Group Relative Policy Optimization (GRPO) for training. For each input text X_i, multiple paraphrased outputs $\{Y_{i1}, Y_{i2}, \ldots, Y_{iG}\}$ are sampled along with their corresponding rewards. The baseline reward b_i is calculated as:

$$b_i = \frac{1}{G} \sum_{j=1}^{G} R(X_i, Y_{ij})$$

The advantage A_{ij} for each sample is:

$$A_{ij} = R(X_i, Y_{ij}) - b_i$$

The model parameters θ are updated by maximizing the objective function:

$$J(\theta) \approx \frac{1}{N} \sum_{i=1}^{N} \frac{1}{G} \sum_{j=1}^{G} A_{ij} \sum_{t} \log \pi_\theta(y_{ij,t} | X_i, y_{ij,<t})$$

3.2 Fine-Tuning Implementation

3.2.1 Motivation

The original AuthorMist reward optimizes against a single averaged detector signal, which can lead to overfitting on that detector's weaknesses and neglect semantic fidelity. In practice, different detectors rely on distinct cues, so optimizing against one signal does not guarantee evasion across others. Moreover, without an explicit semantic constraint, paraphrases may drift from the original meaning. To address these issues—and to enable fine-tuning on freely available, open-source detectors—we introduce a multi-detector reward that averages over complementary signals, alongside a semantic similarity score. This composite objective promotes paraphrases that generalize across diverse detection frameworks while preserving the original text's meaning.

3.2.2 Proposed Reward Function

To fine-tune the model towards producing human-like and semantically faithful paraphrases, we designed a **multi-detector reward function** that combines outputs from multiple specialized classifiers and a semantic similarity model.

Specifically, our reward function integrates three components:

- **Radar Detector:**
  TrustSafeAI/RADAR-Vicuna-7B: Evaluates if a text appears AI-generated. The detector outputs a score, where lower values indicate a higher likelihood of being human-written. We invert the score ($1-$score) so that a higher reward corresponds to more human-like text.
- **DetectGPT Detector:**
  Hello-SimpleAI/chatgpt-detector-roberta: Another independent model that assesses whether the text looks machine-generated. Similar to Radar, lower raw scores suggest human-likeness, and we again invert the score.
- **Semantic Similarity Checker:**
  sentence-transformers/all-mpnet-base-v2: Measures how closely the paraphrased text preserves the original meaning. We extract the original text from the input prompt, generate embeddings for both the original and paraphrased texts, and compute cosine similarity. Higher similarity scores are better, ensuring the paraphrase remains faithful to the input.
- **Reward Calculation**:

  For each paraphrase:

  - A *detector reward* is computed as the average of the inverted Radar and DetectGPT scores.
  - A *semantic reward* is computed from the cosine similarity between the original and paraphrased text embeddings.

  The final reward is a weighted sum: $Reward = 0.7 \times DetectorReward + 0.3 \times SemanticReward$

  This balances promoting *human-likeness* (70%) and *semantic preservation* (30%).

Batching and Memory Management: To reduce memory usage, completions are processed in batches (default size = 16). After each batch, garbage collection and CUDA cache clearing (if available) are triggered to manage system memory.

- **Goal**: Reward outputs that are both *human-like* and *meaning-preserving*.
- **Detectors**: Radar and DetectGPT detect "AI-ness" (lower is better).
- **Semantic Checker**: Ensures meaning is preserved (higher is better).

3.2.3 Transformers Reinforcement Learning (GRPO)

We used the HuggingFace TRL library's `TRLTrainer` with a custom `GRPOConfig` to run our fine-tuning:

```
{
  "output_dir": "checkpoints/",
  "logging_steps": 3,
  "per_device_train_batch_size": 16,
  "per_device_eval_batch_size": 16,
  "gradient_accumulation_steps": 8,
  "num_train_epochs": 5,
  "fp16": true,
  "optim": "adamw_torch",
  "gradient_checkpointing": true,
  "max_grad_norm": 0.5,
  "save_strategy": "epoch",
  "save_total_limit": 1,
  "eval_strategy": "no",
  "num_generations": 2
}
```

Passing this configuration (with model, tokenizer, and dataset) to `TRLTrainer` and calling `trainer.train()` executes the GRPO loop using our composite reward.

3.2.4 Parameter-Efficient Fine-Tuning

We applied PEFT's LoRA to inject low-rank adapters into the transformer, greatly reducing the number of trainable parameters:

```
{
  "task_type": "CAUSAL_LM",
  "inference_mode": false,
  "r": 16,
  "lora_alpha": 128,
  "lora_dropout": 0,
  "target_modules": [
    "q_proj", "k_proj", "v_proj", "o_proj",
    "gate_proj", "up_proj", "down_proj"
  ]
}
```

This `LoraConfig` was passed to `get_peft_model()` along with our base model and tokenizer. During fine-tuning, only the adapter weights are updated, preserving the original model parameters and enabling fast, memory-efficient training.

4 Experimental Setup

4.0.1 Dataset

We used the CheckGPT corpus (Tufts et al., 2025), which contains 10,000 passages equally divided between human-written and AI-generated text. Entries span five domains (Computer Science, Humanities, Social Sciences, Physics, Fiction) and three languages (English, Spanish, Chinese). Each sample includes:

- **Detectability Labels:** Binary flags and continuous confidence scores from multiple detectors (RADAR, WILD, DetectGPT, BiScope).
- **Generator Metadata:** The model family (e.g., GPT-3.5, LLaMA-2) and original prompts used to produce AI texts.

For our experiments, we randomly selected 350 passages (150 human, 200 AI) with proportional domain and language coverage. We then fed all AI-generated passages to four paraphrasers—Qwen, AuthorMist, and fine-tuned versions of AuthorMist for 5 and 10 epochs—using:

```
Please paraphrase the following
text to make it more human-like
while preserving the original
meaning.
{ai_text}
Paraphrased text:
```

Finally, for fine-tuning AuthorMist, we randomly selected 300 AI-generated passages from the CheckGPT dataset, prepended the paraphrasing prompt to each of the passages to prepare our dataset and then ran it through the GRPO training.

4.0.2 Evaluation Setup

All evaluations were conducted using NVIDIA A100 GPUs to ensure efficient inference and training at scale. We evaluated four paraphrasing systems namely Qwen, AuthorMist, and fine-tuned versions of AuthorMist for 5 and 10 epochs. Each model was used to paraphrase the AI-generated responses, and the resulting texts were combined with human-written samples to create a binary classification dataset. The goal was to assess whether these paraphrases could reduce detectability by state-of-the-art AI text detectors. We evaluated all outputs using RADAR and CheckGPT. For each paraphraser, we recorded both soft detection scores and hard labels.

4.0.3 Testing Metrics

We computed key evaluation metrics including AUROC (1), Attack Success Rate (ASR) (2), and F1 Score (3), to quantify detector performance across models. QWEN and AuthorMist were evaluated using their pretrained versions, while Dipper was run via a custom generation script. The SFT-AuthorMist model was fine-tuned using LoRA in

8-bit mode with the TRL library, enabling efficient training without compromising generation quality. All evaluation outputs were visualized using score distributions and ROC curves to highlight each model's evasion effectiveness. This standardized pipeline allowed for a rigorous, comparison of paraphrasing models under consistent conditions, revealing trade-offs in detectability.

$$\text{AUROC} = \mathbb{P}(s(x_{\text{pos}}) > s(x_{\text{neg}})) \tag{1}$$

$$\text{ASR} = \frac{\text{FN}}{\text{TP} + \text{FN}} \tag{2}$$

$$\text{F1 Score} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3}$$

5 Results

Table 1 presents the performance of QWEN and our AuthorMist variants (baseline, fine-tuned for 5 and 10 epochs) on the RADAR and DetectGPT detectors, reporting AUROC, F1, and Attack Success Rate (ASR).

The RADAR results demonstrate unstable evasion: although fine-tuning for 5 epochs increases ASR from 0.380 to 0.435, further training to 10 epochs causes ASR to drop back to 0.395. This non-monotonic behavior indicates that evading RADAR reliably will require longer training and a larger, more diverse fine-tuning corpus to push the model consistently across its decision boundary.

By contrast, DetectGPT evasion improves steadily with additional fine-tuning. Starting from a baseline ASR of 0.380, AuthorMist's ASR jumps to 0.970 after 5 epochs and remains high (0.960) after 10 epochs. Meanwhile, AUROC declines from 0.485 to 0.468 and then to 0.445, confirming that continued training effectively undermines DetectGPT's confidence in AI-generated text.

Overall, these findings underscore the inherent trade-off between evasion strength and detector robustness: as ASR rises, core detection metrics such as AUROC decline. **Future work should focus on extending the training schedule, enlarging the fine-tuning dataset, and incorporating additional detectors into the reward signal to stabilize and generalize evasion across diverse detection frameworks.**

6 Discussion

Our fine-tuning of AuthorMist on the CheckGPT corpus successfully realized several of the goals we set out in our proposal. By integrating RADAR and DetectGPT signals alongside a semantic similarity component, we created a multi-detector reward that encouraged the model to develop evasion strategies beyond those learned in its original training. The most striking result is the near-complete collapse of DetectGPT's ability to distinguish AI-generated text: ASR climbs from 0.380 in the baseline to over 0.96 after fine-tuning, confirming that our composite reward effectively teaches the model to exploit multiple detector weaknesses. This demonstrates that AuthorMist can be steered, via GRPO, to generalize its paraphrasing tactics against open-source detectors it was not originally exposed to.

At the same time, our experiments reveal that evasion against RADAR remains somewhat unstable. While 5 epochs of fine-tuning bump ASR from 0.380 to 0.435, additional training causes a small drop back to 0.395. This non-monotonic behavior suggests that RADAR's decision boundary may be more sensitive to particular paraphrase patterns, and that achieving consistently strong evasion will require longer fine-tuning schedules and a larger, more diverse training set. In line with our proposal, expanding the CheckGPT subset used for reward computation—and potentially adding further detectors into that reward mix—should help stabilize performance across all evaluation models.

Importantly, our new semantic reward term succeeded in preserving meaning while pushing for undetectability. Throughout fine-tuning, the model maintained reasonable F1 and precision scores on the paraphrased set, indicating that additional evasion strength did not come at the expense of completely losing semantic fidelity. This balance between human-likeness and faithfulness was a key design criterion in our proposal and is borne out by the relatively modest declines in AUROC compared to the sharp rises in ASR.

7 Conclusion

Overall, our results confirm that fine-tuning AuthorMist on a publicly available, multi-detector dataset like CheckGPT can substantially improve cross-detector robustness and semantic preservation. The rapid degradation of DetectGPT and the

Model	AUROC	F1 Score	ASR
Evaluation on RADAR Detector			
QWEN	0.9813	**0.3459**	**0.6350**
AuthorMist	**0.9012**	0.5243	0.3800
AuthorMist Fine-Tuned 5	0.9038	0.4891	0.4350
AuthorMist Fine-Tuned 10	0.9125	0.5148	0.3950
Evaluation on DetectGPT			
QWEN	0.7996	0.4235	0.7300
AuthorMist	0.4849	0.5243	0.3800
AuthorMist Fine-Tuned 5	0.4676	**0.0579**	**0.9700**
AuthorMist Fine-Tuned 10	**0.4450**	0.0765	0.9600

Table 1: Comprehensive evaluation of QWEN and AuthorMist variants (baseline and fine-tuned for 5 and 10 epochs) on the RADAR and DetectGPT detectors. Fine-tuning AuthorMist yields higher ASR—especially against DetectGPT—while incurring modest drops in AUROC and accuracy, highlighting the trade-off between evasion strength and detection robustness.

partial stability on RADAR demonstrate both the promise and the remaining challenges of multi-signal reward learning. Future work will focus on scaling the fine-tuning dataset, incorporating additional open-source detectors into the reward, and exploring curriculum or adversarial training strategies to further smooth evasion performance across diverse detection frameworks.

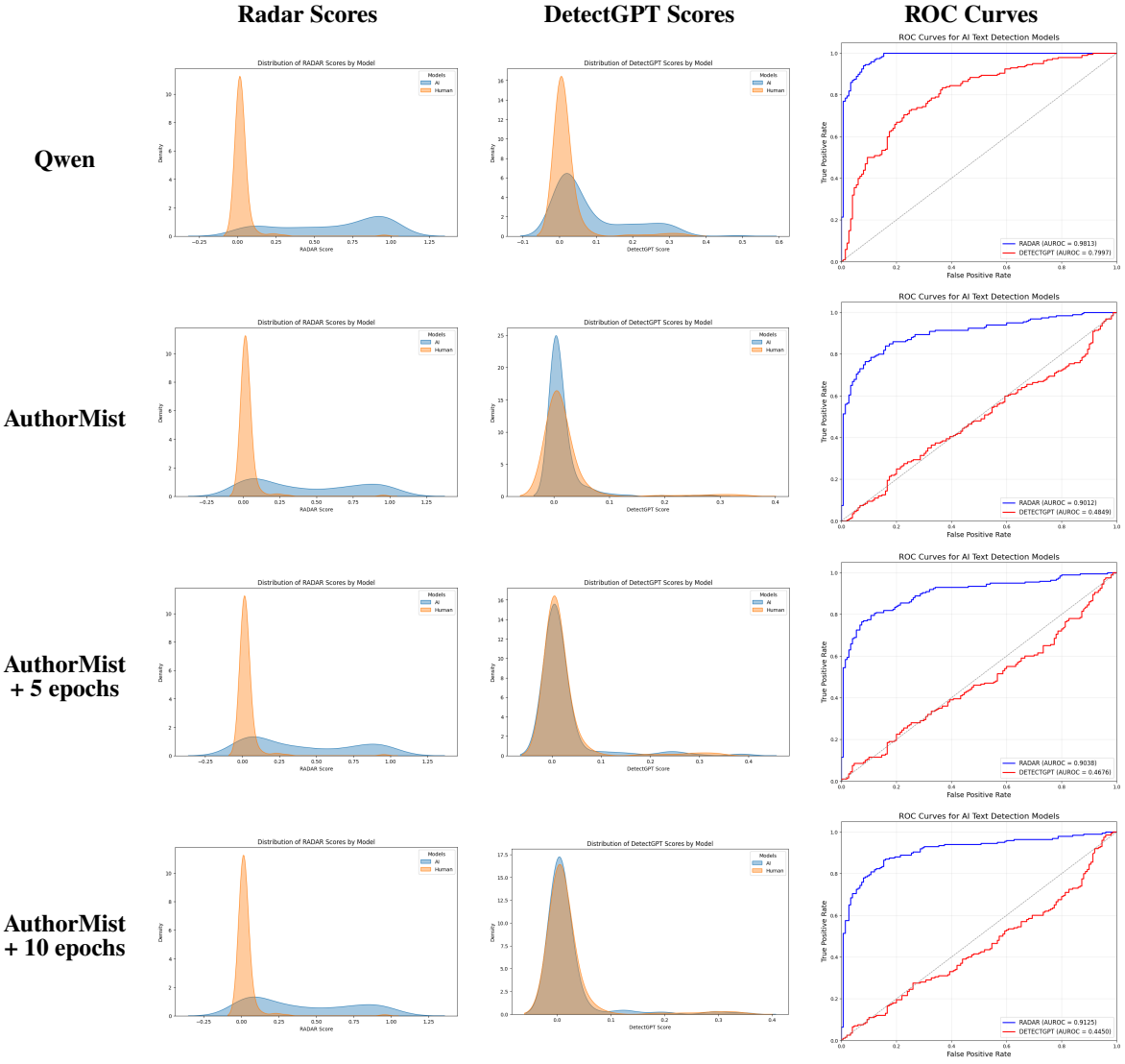

Figure 2: Radar Scores, DetectGPT Scores, and ROC Curves for Qwen, AuthorMist, and fine-tuned AuthorMist models.

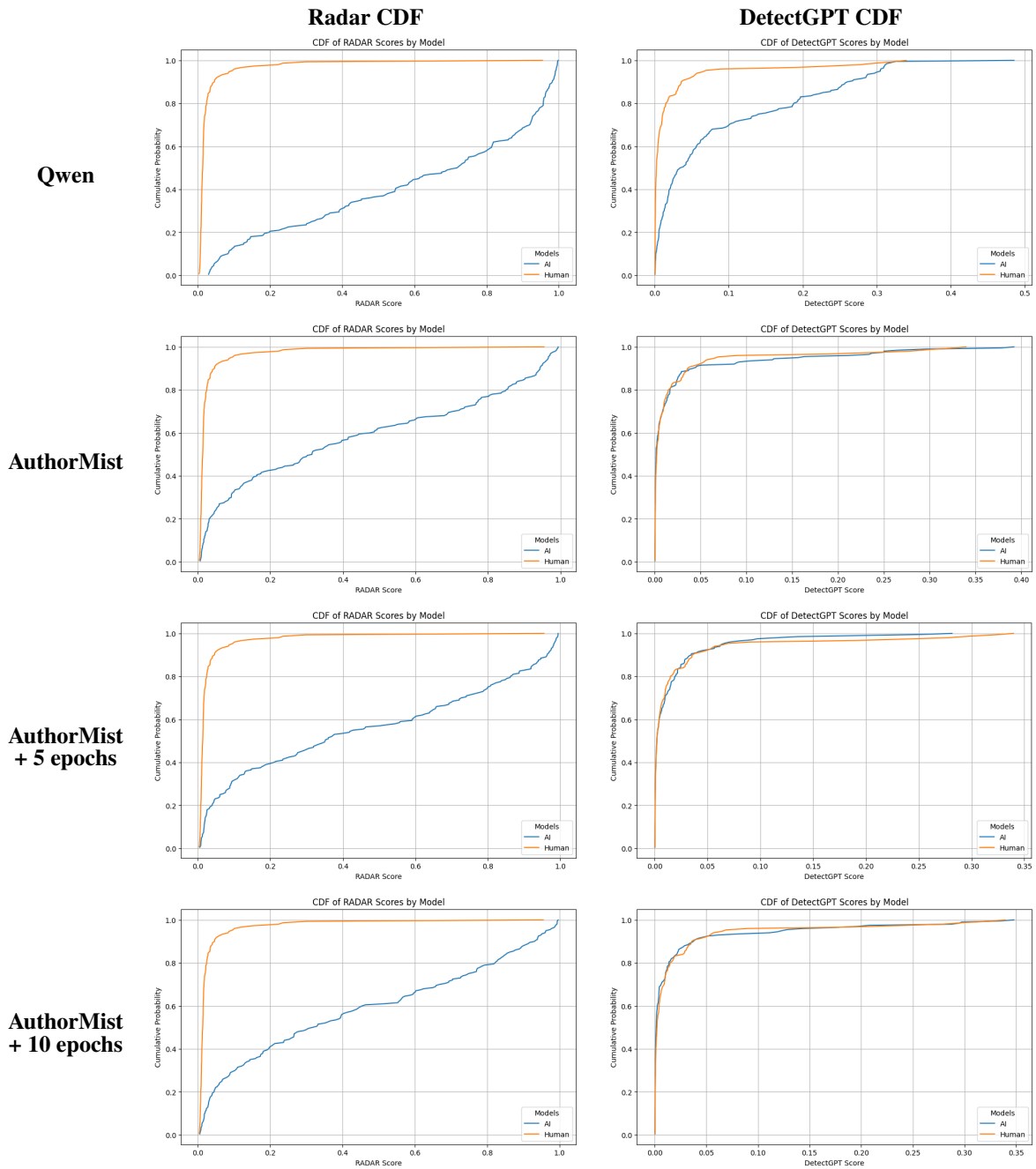

Figure 3: Radar CDF and DetectGPT CDF for Qwen, AuthorMist, and fine-tuned AuthorMist models.

References

Guangsheng Bao, Yanbin Zhao, Zhiyang Teng, Linyi Yang, and Yue Zhang. 2023. Fast-detectgpt: Efficient zero-shot detection of machine-generated text via conditional probability curvature. *arXiv preprint arXiv:2310.05130*.

Saurav Creo and Aakash Pudasaini. 2024. Silverspeak: Evading ai-text detectors with homoglyph substitution. *arXiv preprint arXiv:2404.08627*.

Henrique da Silva Gameiro, Andrei Kucharavy, and Rachid Guerraoui. 2023. Stochastic parrots looking for stochastic parrots: Llms are easy to fine-tune and hard to detect with other llms. *arXiv preprint arXiv:2304.08968*.

Isaac David and Arthur Gervais. 2025. Authormist: Evading ai text detectors with reinforcement learning. *arXiv preprint arXiv:2503.08716*.

Sebastian Gehrmann, Hendrik Strobelt, and Alexander Rush. 2019. GLTR: Statistical detection and visualization of generated text. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 111–116.

Biyang Guo, Xin Zhang, Ziyuan Wang, Minqi Jiang, Jinran Nie, Yuxuan Ding, Jianwei Yue, and Yupeng Wu. 2023. How close is chatgpt to human experts? comparison corpus, evaluation, and detection. *arXiv preprint arXiv:2301.07597*.

Qian Guo, Chen Henry Wei, Linyi Yang, Bowen Zhou, Zhilin Yang, and Yue Zhang. 2024. Biscope: Accurate and generalizable detection of ai-generated text via bidirectional prediction dynamics. *OpenReview preprint*. Https://openreview.net/pdf?id=Hew2JSDycr.

Abhimanyu Hans, Avi Schwarzschild, Valeriia Cherepanova, Hamid Kazemi, Aniruddha Saha, Micah Goldblum, Jonas Geiping, and Tom Goldstein. 2024. Spotting llms with binoculars: Zero-shot detection of machine-generated text. *arXiv preprint arXiv:2401.12070*.

Daphne Ippolito, Daniel Duckworth, Chris Callison-Burch, and Douglas Eck. 2020. Automatic detection of generated text is easiest when humans are fooled. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1808–1822.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. 2023. A watermark for large language models. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, volume 202, pages 17061–17084.

Kalpesh Krishna, Yixiao Song, Marzena Karpinska, John F. Wieting, and Mohit Iyyer. 2023. Paraphrasing evades detectors of ai-generated text, but retrieval is an effective defense. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 51008–51025.

Weixin Liang, Yining Mao, and James Zou. 2024. A corpus-level detection framework to estimate the prevalence of llm usage in peer reviews. *arXiv preprint arXiv:2410.03019*.

Weixin Liang, Mert Yuksekgonul, Yining Mao, Eric Wu, and James Zou. 2023. Gpt detectors are biased against non-native english writers. *Patterns*, 4(7):100779.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D. Manning, and Chelsea Finn. 2023. Detectgpt: Zero-shot machine-generated text detection using probability curvature. *arXiv preprint arXiv:2301.11305*.

Alan Russo Latona, Tade Raji, Jacob Jordan, and Yacine Jernite. 2024. How many peer reviews are written with ai assistance? *arXiv preprint arXiv:2405.02150*.

Vinu Sankar Sadasivan, Aounon Kumar, Sriram Balasubramanian, Wenxiao Wang, and Soheil Feizi. 2023. Can ai-generated text be reliably detected? *arXiv preprint arXiv:2303.11156*.

Sinclair Schneider, Florian Steuber, João A. G. Schneider, and Gabi Dreo Rodosek. 2025. Detection avoidance techniques for large language models. *Data & Policy*, 7:e29.

Lei Tao, Yifan Zhang, Shu Wang, and Yue Zhang. 2024. Cudrt: A comprehensive benchmark for ai-generated text detection in real-world tasks. *arXiv preprint arXiv:2406.09056*.

Brian Tufts, Xuandong Zhao, and Lei Li. 2025. A practical examination of ai-generated text detectors for large language models. *arXiv preprint:2412.05139*.

Eduard Tulchinskii, Kristian Kuznetsov, Laida Kushnareva, Daniil Cherniavskii, Serguei Barannikov, Irina Piontkovskaya, Sergey Nikolenko, and Evgeny Burnaev. 2023. Intrinsic dimension estimation for robust detection of ai-generated texts. *arXiv preprint arXiv:2306.04723*.

Xinyu Weng, Jiajun Wu, Jimei Yang, and William Yang Wang. 2024. Cycleresearcher: Iterative draft refinement of scientific papers by language model agents. *arXiv preprint arXiv:2411.00816*.

Zheng Yu, Bing Li, Bowen Zhou, Zhilin Yang, and Yue Zhang. 2024. Text fluoroscopy: Layer-wise disentanglement of human and llm representations. *EMNLP 2024*. Https://aclanthology.org/2024.emnlp-main.885.pdf.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. Defending against neural fake news. In *Advances in Neural Information Processing Systems 32 (NeurIPS)*, pages 9051–9065.