STEALTHRL: REINFORCEMENT LEARNING PARAPHRASE ATTACKS FOR MULTI-DETECTOR EVASION OF AI-TEXT DETECTORS

Anonymous authors
Paper under double-blind review

ABSTRACT

We introduce *StealthRL*, a reinforcement-learning framework for systematic stress-testing and red-teaming of AI-text detectors through adversarial paraphrasing. By generating high-fidelity paraphrases that evade detection while preserving semantic content, StealthRL exposes brittleness in detector families and informs defensive improvements. The technical challenge is to reduce detector confidence at strict low-FPR operating points without collapsing semantic fidelity or overfitting to a single detector family. StealthRL fine-tunes a single Qwen3-4B paraphraser with LoRA Hu et al. (2021) and a composite reward balancing detector evasion with semantic similarity. On the MAGE benchmark with three detector families (RoBERTa OpenAI, Fast-DetectGPT, Binoculars), StealthRL reduces mean TPR@1%FPR from 0.27 (baseline) to 0.04 while maintaining high semantic similarity (0.953 E5 cosine), revealing significant transferable vulnerabilities. We release an anonymized code package for reproducible robustness evaluation.

1 INTRODUCTION

AI-generated text detectors are widely deployed in academic integrity, content moderation, and misinformation pipelines, yet their robustness to paraphrasing remains brittle. Recent work shows that detector performance can drop sharply under paraphrase attacks, motivating adversarial evaluation and red-teaming rather than static benchmarking. We focus on the adversarial aspect: how to generate high-fidelity paraphrases that reliably evade multiple detector families, while maintaining readability and semantic fidelity.

We present **StealthRL**, a reinforcement-learning paraphraser trained to reduce detector confidence under a strict low-FPR operating point. StealthRL is designed to (i) train against multiple detector families to promote transfer, (ii) preserve semantic content and fluency via auxiliary reward terms, and (iii) provide an evaluation harness that reports transfer and attack success at 1% FPR. We position StealthRL as a red-teaming tool that helps stress-test detector robustness and guide defensive improvements.

Contributions.

- **Multi-detector RL paraphrasing.** We train a single paraphrase policy against a detector ensemble using group-relative policy optimization with LoRA, enabling efficient adversarial fine-tuning.
- **Low-FPR evaluation protocol.** We report attack success across multiple detector families at TPR@1%FPR and release an evaluation pipeline that produces heatmaps and tradeoff curves.
- **Empirical results.** On MAGE, StealthRL reduces mean TPR@1%FPR to 0.04 (from 0.27 for no attack) while maintaining high semantic similarity.

Anonymized code release. An anonymized code release containing training and evaluation scripts is included in the supplementary material. A placeholder anonymous link is provided: `https://anonymous.4open.science/r/STEALTHRL`.

2 Related Work

Detector families include curvature-based methods (DetectGPT and Fast-DetectGPT) and paired-LM detectors such as Binoculars. Mitchell et al. (2023); Bao et al. (2024); Hans et al. (2024) Recent evasion work focuses on paraphrase attacks and RL-based humanization, including Adversarial Paraphrasing, while character-level attacks such as homoglyph substitution demonstrate strong but often less readable perturbations. Cheng et al. (2025); Creo & Pudasaini (2025) StealthRL builds on these directions by training a single paraphrase policy against a detector ensemble and evaluating transfer at strict low-FPR operating points.

3 Threat Model

We assume **black-box access to detector scores**, i.e., the attacker can query detector confidence but does not require gradients. In practice, detectors are often open-source or deployed with a confidence API, so both black-box scoring and open-source replication are realistic. We evaluate transfer to a held-out detector family to test robustness beyond the training ensemble.

4 Method

Given AI-generated text x, we learn a paraphrase policy $\pi_\theta(y \mid x)$ that produces y with low detector confidence while preserving meaning. We optimize a composite reward:

$$R(x, y) = \alpha R_{\text{det}}(y) + \beta R_{\text{sem}}(x, y), \tag{1}$$

where R_{det} is the ensemble detector score (lower AI probability) and R_{sem} is E5 embedding cosine similarity. We train with group-relative policy optimization (GRPO) and LoRA adapters Hu et al. (2021) on Qwen/Qwen3-4B-Instruct-2507 (rank 32, $\alpha = 32$, dropout 0.05). Training uses group size 8, learning rate 2.8
$times10^{-4}$, batch size 16, and two epochs over 10,000 samples from the MAGE training split, with an additional 200 held-out development samples from the test split. The detector ensemble used for training is RoBERTa OpenAI + Fast-DetectGPT with weights 0.6/0.4.

Reproducibility details. Reward weights are $\alpha = 1.0$, $\beta = 0.1$. GRPO uses KL penalty coefficient 0.05 with the base Qwen3-4B-Instruct as the reference policy. Inference uses temperature 1.0, top-p 0.9, max tokens 512. The paraphrase prompt is: "Paraphrase the following text while preserving its meaning: [TEXT]". Detector versions: RoBERTa (openai-community/roberta-large-openai-detector), Fast-DetectGPT (scoring model: EleutherAI/gpt-neo-2.7B), Binoculars (lightweight version: gpt2-medium performer, gpt2-large observer). Threshold calibration uses quantile estimation at FPR=0.01 on 1,000 human samples. Training uses the Tinker API framework, with reward computation and offline evaluation performed on a combination of MacBook and NVIDIA A10 GPUs. Full hyperparameters are in Appendix A.

5 Experimental Setup

Dataset. We evaluate on MAGE (Machine-generated Text Detection in the Wild) using the test split with 1,000 human and 1,000 AI samples, filtered to a 100–500 token window. Li et al. (2024)

Detectors. Evaluation uses three detector families: RoBERTa OpenAI classifier, Fast-DetectGPT (curvature-based), and Binoculars (paired-LM). Mitchell et al. (2023); Bao et al. (2024); Hans et al. (2024)

Methods. We compare five methods: M0 (no attack), M1 (simple paraphrase via Qwen3:4B), M2 (StealthRL-v1 checkpoint), M3 (detector-guided selection inspired by Adversarial Paraphrasing), and M5 (homoglyph substitution via SilverSpeak). Cheng et al. (2025); Creo & Pudasaini (2025)

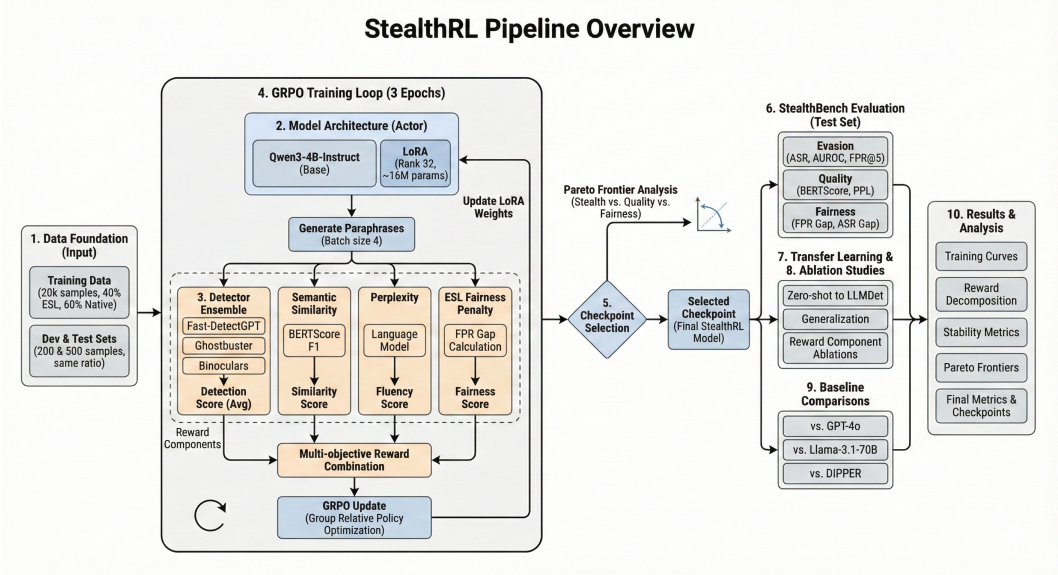

Figure 1: StealthRL training and evaluation pipeline. A single paraphrase policy is trained with detector-guided reward and quality constraints, then evaluated across multiple detector families with low-FPR metrics and tradeoff analysis.

Method	TPR@1%FPR ↓ (R)	F (Fast)	B (Binoc)	Mean ↓	ASR ↑	AUROC ↓
M0 No Attack	0.22	0.28	0.32	0.27	0.73	0.71
M1 Simple Para	0.14	0.06	0.02	0.07	0.93	0.54
M2 StealthRL-v1	0.08	0.02	0.02	0.04	0.96	0.43
M3 Adv. Para (guided)	0.14	0.06	0.04	0.08	0.92	0.56
M5 Homoglyph	0.02	0.00	0.00	**0.01**	**0.99**	0.41

Table 1: Main results on MAGE (TPR@1%FPR, ASR, and AUROC). Lower TPR/AUROC is better for the attacker; higher ASR is better. R/F/B denote RoBERTa/Fast-DetectGPT/Binoculars.

Metrics. We calibrate a 1% FPR threshold per detector on human texts and report TPR@1%FPR (lower is better for the attacker), attack success rate (ASR = 1 - TPR@1%FPR), and AUROC. We also report semantic similarity (E5 cosine) to quantify fidelity. All results use one candidate per sample (n=1) to match the current evaluation run.

LLM-based quality evaluation (optional). Following Adversarial Paraphrasing, we optionally use a GPT-5-mini autorater to score paraphrase quality and semantic equivalence on a 1–5 Likert scale, capped to a fixed number of samples per method. Cheng et al. (2025); **?**); **?**); **?**

6 RESULTS

Table 1 summarizes detection evasion at 1% FPR. StealthRL reduces mean TPR@1%FPR to 0.04 (vs. 0.27 for no attack) and improves mean AUROC from 0.71 to 0.43. Per-detector TPR@1%FPR drops from 0.22/0.28/0.32 (M0) to 0.08/0.02/0.02 (M2) on RoBERTa/Fast-DetectGPT/Binoculars, indicating strong transfer to the held-out detector family. StealthRL maintains high semantic similarity (0.953 E5 cosine), comparable to simple paraphrasing (0.959), while substantially outperforming the detector-guided baseline (M3).

Interpretability: What do attacks reveal? The strong transfer of StealthRL across detector families (including the held-out Binoculars detector) suggests that detectors rely on shared, brittle statistical cues rather than robust semantic understanding. The low TPR@1%FPR (0.02–0.08) achieved while maintaining high semantic similarity (0.953 E5) indicates that surface-level paraphrasing suf-

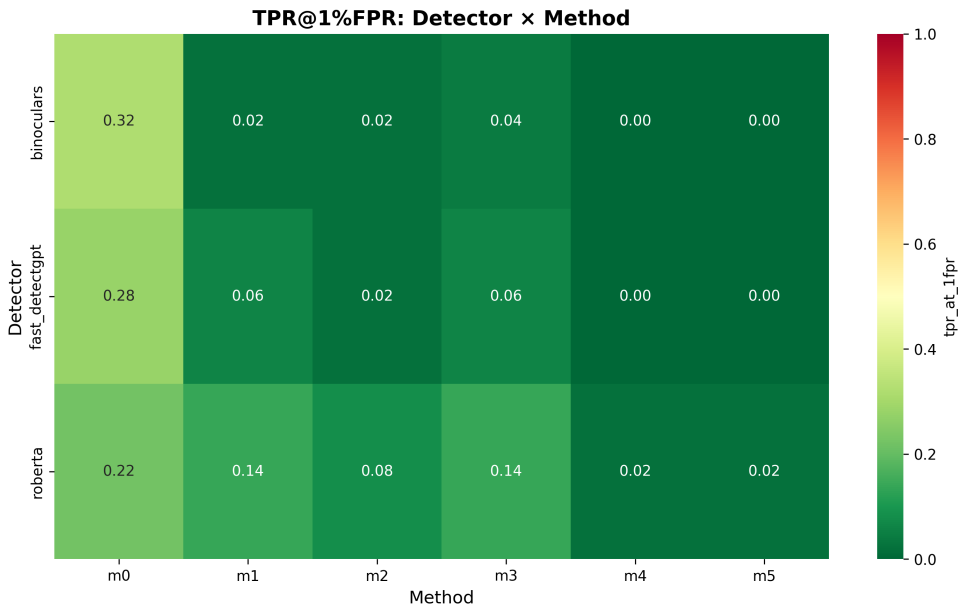

Figure 2: Detector × method heatmap of TPR@1%FPR. StealthRL shows consistently low TPR across detectors, indicating transfer beyond the training ensemble.

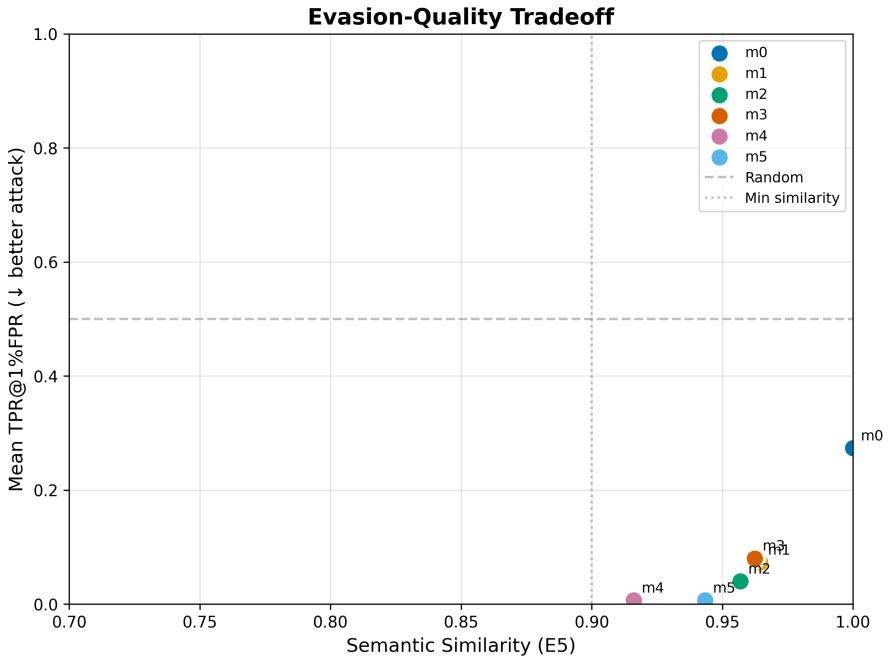

Figure 3: Evasion–quality tradeoff (mean TPR@1%FPR vs. semantic similarity). StealthRL achieves strong evasion while preserving meaning; homoglyph attacks improve evasion but degrade fluency and readability.

fices to evade detection, exposing a fundamental limitation in current detector architectures. This finding motivates defensive research into semantic-aware detectors and multi-modal robustness evaluation, as the adversarial paraphrasing approach is modality-agnostic and could be extended to code generation, image captioning, or other structured outputs.

Ablation (implemented; results pending). We implemented a *guidance-transfer* ablation that mimics the "guidance vs. deploy" setting from Adversarial Paraphrasing: candidate selection is guided by (i) RoBERTa, (ii) Fast-DetectGPT, or (iii) the ensemble mean, and evaluated across all detectors. This isolates how guidance detector choice affects transfer. Results are pending for the final model checkpoints.

7 Limitations and Safety

Our evaluation currently focuses on three detector families and one primary benchmark; additional detectors (e.g., watermark-based) and datasets can broaden coverage. **Safety framing:** Adversarial paraphrasing is dual-use technology. We explicitly position StealthRL as a *stress-testing and red-teaming tool* designed to expose detector brittleness and guide defensive improvements, not as a production evasion system. Our released code is intended for robustness evaluation by researchers and detector developers. The strong attack success we observe (mean TPR@1%FPR = 0.04) underscores the urgent need for more robust detection methods and motivates future work on semantic-aware detectors, ensemble defenses, and adversarial training.

8 LLM Usage Disclosure

Large language models were used to assist with code scaffolding and with editing text for clarity. All modeling decisions, experiments, evaluations, and interpretations were designed, executed, and validated by the authors.

9 Conclusion

StealthRL provides a practical adversarial RL framework for generating high-fidelity paraphrases that evade multiple detector families under strict low-FPR constraints. Our results show strong transfer to a held-out detector while preserving semantic fidelity, and our released evaluation pipeline supports reproducible red-teaming of AI-text detectors.

Acknowledgments

We gratefully acknowledge Thinking Machines for providing access to their Tinker API framework, which was essential for the reinforcement learning training in this work. We also thank the open-source community for the detector implementations and model checkpoints that enabled this evaluation.

References

Guangsheng Bao, Yanbin Zhao, Zhiyang Teng, Linyi Yang, and Yue Zhang. Fast-detectgpt: Efficient zero-shot detection of machine-generated text via conditional probability curvature, 2024. URL https://arxiv.org/abs/2310.05130.

Yize Cheng, Vinu Sankar Sadasivan, Mehrdad Saberi, Shoumik Saha, and Soheil Feizi. Adversarial paraphrasing: A universal attack for humanizing ai-generated text, 2025. URL https://arxiv.org/abs/2506.07001.

Aldan Creo and Shushanta Pudasaini. Silverspeak: Evading ai-generated text detectors using homoglyphs, 2025. URL https://arxiv.org/abs/2406.11239.

Isaac David and Arthur Gervais. Authormist: Evading ai text detectors with reinforcement learning, 2025. URL https://arxiv.org/abs/2503.08716.

Abhimanyu Hans, Avi Schwarzschild, Valeriia Cherepanova, Hamid Kazemi, Aniruddha Saha, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Spotting llms with binoculars: Zero-shot detection of machine-generated text, 2024. URL https://arxiv.org/abs/2401.12070.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021. URL https://arxiv.org/abs/2106.09685.

Yafu Li, Qintong Li, Leyang Cui, Wei Bi, Zhilin Wang, Longyue Wang, Linyi Yang, Shuming Shi, and Yue Zhang. Mage: Machine-generated text detection in the wild, 2024. URL https://arxiv.org/abs/2305.13242.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D. Manning, and Chelsea Finn. Detectgpt: Zero-shot machine-generated text detection using probability curvature, 2023. URL https://arxiv.org/abs/2301.11305.

A Hyperparameters and Configuration

Parameter	Value
Model & LoRA	
Base model	Qwen/Qwen3-4B-Instruct-2507
LoRA rank	32
LoRA alpha	32
LoRA dropout	0.05
Training	
Algorithm	Group-Relative Policy Optimization (GRPO)
Learning rate	2.8×10^{-4}
Batch size	16
Group size	8
Epochs	2
Training samples	10,000 (MAGE train) + 200 (dev)
KL penalty coefficient	0.05
Reference policy	Qwen3-4B-Instruct (frozen)
Reward	
Detector weight (α)	1.0
Semantic weight (β)	0.1
Detector ensemble	RoBERTa (0.6) + Fast-DetectGPT (0.4)
Semantic metric	E5 embedding cosine similarity
Inference	
Temperature	1.0
Top-p	0.9
Max tokens	512
Prompt template	"Paraphrase the following text while preserving its meaning: [TEXT]"
Detectors	
RoBERTa OpenAI	openai-community/roberta-large-openai-detector
Fast-DetectGPT	Scoring model: EleutherAI/gpt-neo-2.7B
Binoculars	Lightweight: gpt2-medium + gpt2-large (held-out)
Evaluation	
Test samples	1,000 human + 1,000 AI (MAGE test)
Token window	100–500 tokens
FPR calibration	1% on 1,000 human samples (quantile)
Candidates per sample	1
Compute	
Training framework	Tinker API (Thinking Machines)
Reward computation	MacBook + NVIDIA A10 GPUs
Offline evaluation	MacBook + NVIDIA A10 GPUs
Seed	42

Table 2: Complete hyperparameters and configuration for reproducibility.

B QUALITATIVE EXAMPLES

Original (M0)	**StealthRL (M2)**
During cardio the heart increases its workload and all the body's other systems adjust to help support that endeavor. The blood vessels dilate, the muscles do their best to help pump blood back to the heart, and the lungs work harder to take in oxygen and remove waste gases like carbon dioxide.	In cardio, the heart ramps up its workload, prompting the body's other systems to adapt and assist. Blood vessels widen, muscles strive to push blood back to the heart, and lungs intensify their efforts to absorb oxygen and expel waste gases such as carbon dioxide.
The engine has to endure the torque of powering two axels and a drive shaft generally the transfer casing connects to the drive shaft with a ujoint and same for power distribution. The electricity goes into the battery first then is sent to the alternator where it generates voltage that powers all other electrical components like the lights, radio.	The engine must handle the torque from two axles and a drive shaft, typically linked via a universal joint in the transfer casing for power distribution. Electricity first charges the battery, then flows to the alternator, which converts it into voltage to power essential electrical systems such as lights and the radio.

Table 3: Representative paraphrases from the evaluation run (MAGE test split).