StealthRL: Reinforcement Learning Paraphrase Attacks for Multi-Detector Evasion of AI-Text Detectors

Suraj Ranganath
University of California, San Diego
suranganath@ucsd.edu

Atharv Ramesh
University of California, San Diego
a3nair@ucsd.edu

Abstract

AI-text detectors are increasingly used in high-stakes settings, but their robustness to meaning-preserving adversarial rewriting remains unclear. We introduce *StealthRL*, a reinforcement learning framework for stress-testing detectors with adaptive paraphrase attacks. StealthRL trains a paraphrase policy against a detector ensemble while enforcing semantic fidelity, then evaluates transfer to held-out detector families on the full filtered MAGE test pool. Across RoBERTa, Fast-DetectGPT, Binoculars, and MAGE, the resulting attack sharply degrades detection performance, driving mean TPR@1%FPR down to 0.024 while retaining nontrivial semantic overlap and readable prose. The attack also transfers to detectors not used during training, indicating a broader robustness failure rather than overfitting to a single target model. We complement the main results with detector-score analysis, repeated-generation stability checks, and automatic and judge-based quality evaluation. Our findings suggest that current AI-text detectors remain brittle under adaptive paraphrasing pressure and motivate robustness evaluation at deployment-relevant operating points. Our code and the trained model checkpoint are publicly available.[1]

1 Introduction

Large language models (LLMs) produce text that is increasingly indistinguishable from human writing, raising urgent concerns about academic integrity, misinformation, and content provenance (Solaiman et al., 2019). In response, a growing ecosystem of AI-text detectors has been deployed in educational institutions, publishing platforms, and content moderation systems. These detectors span diverse architectures, from fine-tuned classifiers (Solaiman et al., 2019) to zero-shot statistical methods (Mitchell et al., 2023; Bao et al., 2024) and paired-LM approaches (Hans et al., 2024), yet their robustness to adversarial manipulation remains poorly understood.

Standard detector evaluations measure performance on *clean* distributions, where AI-generated text is presented without any evasion attempt. However, real-world adversaries are adaptive: they can iteratively refine paraphrases, query detector APIs, and exploit known weaknesses. This gap between clean-distribution evaluation and adversarial robustness is critical. A detector that achieves 95% accuracy on clean text may fail catastrophically when an attacker deliberately targets its decision boundary.

The choice of operating point further complicates evaluation. Most prior work reports AUROC or accuracy at default thresholds, but deployed detectors in educational and moderation settings are often expected to operate at high precision because false positives can wrongly implicate human writers. This remains true even when detectors are used as human-in-the-loop triage tools rather than autonomous adjudicators: a detector alert still creates review burden, escalation risk, and reputational cost. Commercial systems explicitly discuss false-positive targets below 1% (Turnitin, 2023), and recent detector evaluations emphasize TPR measured at fixed FPR values such as 1% (Tufts et al., 2025). We therefore treat TPR@1%FPR as a strict high-precision triage setting, report TPR@5%FPR as a less conservative review-queue setting, and include AUROC as a threshold-independent view.

We introduce **StealthRL**, a reinforcement learning framework that systematically evaluates detector robustness under adaptive adversarial conditions. StealthRL trains a paraphrase policy against a multi-detector ensemble, using GRPO (Pang and Jin, 2025) with LoRA adapters (Hu et al., 2022) on

[1] Code: https://github.com/suraj-ranganath/StealthRL; model: https://huggingface.co/suraj-ranganath/StealthRL.

Proceedings of the First Workshop on Pluralistic Value Alignment of LLMs, pages 1–16
November 10, 2026 ©2026 Association for Computational Linguistics

Qwen3-4B-Instruct, to minimize detection confidence while explicitly rewarding semantic fidelity. By evaluating at the 1% FPR operating point and testing transfer to multiple held-out detectors, we provide a rigorous robustness assessment that complements standard benchmarks.

Our contributions are as follows:

- We implement black-box adaptive paraphrasing attacks via multi-detector RL training with semantic constraints, one of the first works to apply RL for adversarial detector evasion across multiple detector families simultaneously.
- We demonstrate catastrophic robustness failure: 0.024 mean TPR@1%FPR across four detectors, with strong transfer to two held-out detectors.
- We provide a reproducible evaluation protocol for measuring evasion, transfer, and quality at security-relevant operating points, along with supporting analyses such as repeat-run stability and semantic evaluation.

2 Related Work

2.1 AI-Text Detection Methods

AI-text detection methods can be broadly categorized into several families. **Fine-tuned classifiers** train discriminative models on labeled human and AI text; the RoBERTa-based OpenAI detector (Solaiman et al., 2019) is a widely used example. **Zero-shot statistical methods** exploit properties of language model probability distributions without requiring labeled data. DetectGPT (Mitchell et al., 2023) uses probability curvature, while Fast-DetectGPT (Bao et al., 2024) achieves similar accuracy with substantially reduced computational cost via conditional probability curvature. **Paired-LM detectors** such as Binoculars (Hans et al., 2024) compare log-likelihoods across two language models to detect statistical anomalies. **Linguistic-feature detectors** use stylometric or language-specific cues rather than only model likelihoods; KatFishNet, for example, detects Korean LLM text using spacing, part-of-speech diversity, and comma-usage features (Park et al., 2025). The MAGE benchmark (Li et al., 2024) provides standardized evaluation data, while RAID (Dugan et al., 2024) offers a shared benchmark for robust detector evaluation across domains and attacks.

Despite diverse architectures, Sadasivan et al. (2023) provide theoretical arguments that reliable detection may be fundamentally impossible as language models improve, motivating empirical robustness evaluation like ours.

2.2 Adversarial Attacks on Detectors

Evasion attacks on AI-text detectors range from simple paraphrasing to sophisticated adaptive methods. Krishna et al. (2023) demonstrate that paraphrasing with their DIPPER model effectively evades detectors, though retrieval-based defenses can mitigate the attack. Cheng et al. (2025) study adversarial paraphrasing as a universal attack, using detector-guided candidate selection to humanize AI-generated text. Character-level attacks such as homoglyph substitution (SilverSpeak (Creo and Pudasaini, 2025)) achieve strong evasion by replacing characters with visually similar Unicode glyphs, but often degrade readability and are detectable by normalization.

Most relevant to our work, AuthorMist (David and Gervais, 2025) applies reinforcement learning to train an evasion policy against a single detector. StealthRL extends this approach to multi-detector ensemble training with held-out evaluation, strict low-FPR operating points, and multi-metric quality evaluation.

Watermark-based detection (Kirchenbauer et al., 2023) represents an orthogonal approach where the text generator embeds a statistical signal during generation. While watermarks can provide stronger guarantees, they require control over the generation process and are outside the scope of post-hoc detection methods we evaluate.

A natural defensive response to paraphrasing attacks is adversarial training, where detectors are fine-tuned on adversarially generated examples to improve robustness. However, adversarial training faces a fundamental scalability challenge: the space of possible paraphrasing strategies is vast, and a detector hardened against one attack family may remain vulnerable to novel attack methods. The RAID benchmark (Dugan et al., 2024) evaluates detectors across diverse attack types and finds that no single detector achieves robust performance against all attacks, underscoring the difficulty of building universally robust defenses. Our work focuses on the attack side of this arms race to quantify the current robustness gap and motivate the development of stronger defensive strategies.

2.3 Reinforcement Learning for Text Generation

Reinforcement learning from human feedback (RLHF) (Ouyang et al., 2022) has become the standard approach for aligning language models with human preferences. Proximal Policy Optimization (PPO) (Schulman et al., 2017) was the original RL algorithm used for RLHF, but recent work has introduced more efficient alternatives. Group Relative Policy Optimization (GRPO) (Pang and Jin, 2025; Shao et al., 2024) eliminates the need for a separate value network by using group-level relative rewards, reducing memory requirements and enabling efficient training.

Parameter-efficient fine-tuning methods, particularly LoRA (Hu et al., 2022) and QLoRA (Dettmers et al., 2023), enable RL training of large language models with limited compute by adapting only low-rank weight matrices. StealthRL combines GRPO with LoRA on Qwen3-4B-Instruct, demonstrating that efficient RL fine-tuning suffices for learning effective evasion policies.

3 Method

3.1 Threat Model

We consider an adversary who has produced AI-generated text and seeks to modify it to evade detection while preserving semantic content. Formally:

- **Attacker capability**: Black-box access to detector scores. The attacker can query detector confidence $p(y)$ (probability that input y is AI-generated) but does not have access to model gradients or internal parameters. Although our experiments use open-source detectors for which gradients are in principle available, we deliberately treat them as black-box oracles, querying only scalar confidence scores.
- **Attacker goal**: Produce a paraphrase y of AI-generated text x such that $p(y) < \tau$ for detector threshold τ, while maintaining semantic equivalence $\text{sim}(x, y) > \delta$ for a similarity threshold δ.
- **Attacker constraint**: The paraphrase must be a fluent, grammatical reformulation of the original text, not a trivially corrupted version.

We evaluate transfer by testing against two *held-out* detectors (Binoculars and MAGE) not seen during training, assessing whether learned evasion strategies generalize across detector architectures.

3.2 Reward Design

Given AI-generated text x and paraphrase $y \sim \pi_\theta(\cdot \mid x)$, we define a composite reward:

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

where $\alpha = 1.0$ and $\beta = 0.1$ control the evasion-quality tradeoff. We set α to make detector evasion the dominant training signal and use a smaller β as a soft fidelity regularizer rather than a competing objective. This choice reflects the attack setting: outputs that remain easily detected fail the primary goal, while semantic preservation is also enforced through the E5 reward, KL regularization, output validation, and downstream quality evaluation.

Detector evasion reward. The detector reward measures how effectively the paraphrase evades the training ensemble:

$$R_{\text{det}}(y) = 1 - p_{\text{ens}}(y) \quad (2)$$

$$p_{\text{ens}}(y) = w_1 \cdot p_{\text{RoBERTa}}(y) + w_2 \cdot p_{\text{Fast-DetectGPT}}(y) \quad (3)$$

where $w_1 = 0.6$ and $w_2 = 0.4$ are the ensemble weights. The training ensemble intentionally combines one supervised classifier (RoBERTa) and one zero-shot curvature detector (Fast-DetectGPT). We give RoBERTa slightly higher weight because its calibrated neural score provides a stable dense reward, while Fast-DetectGPT contributes a complementary likelihood-curvature signal. We do not include Binoculars or MAGE in the reward loop; they are held out to test transfer, and adding these slower detectors to every GRPO rollout would substantially increase training cost.

Both detector terms are first mapped to an AI-probability scale before averaging. RoBERTa uses the softmax probability assigned to the detector's AI/fake class. Fast-DetectGPT produces an analytic sampling-discrepancy criterion $c(y)$; in the RL reward implementation we transform it as

$$p_{\text{Fast-DetectGPT}}(y) = \sigma(0.5\, c(y)), \quad (4)$$

then clamp the resulting value to $[0, 1]$. Thus p_{ens} is a convex average of two bounded AI-likelihood scores. For GRPO stability, the raw evasion reward $1 - p_{\text{ens}}(y)$ is then normalized with a running z-score and clipped to $[-3, 3]$ before applying the α coefficient. This normalization is used

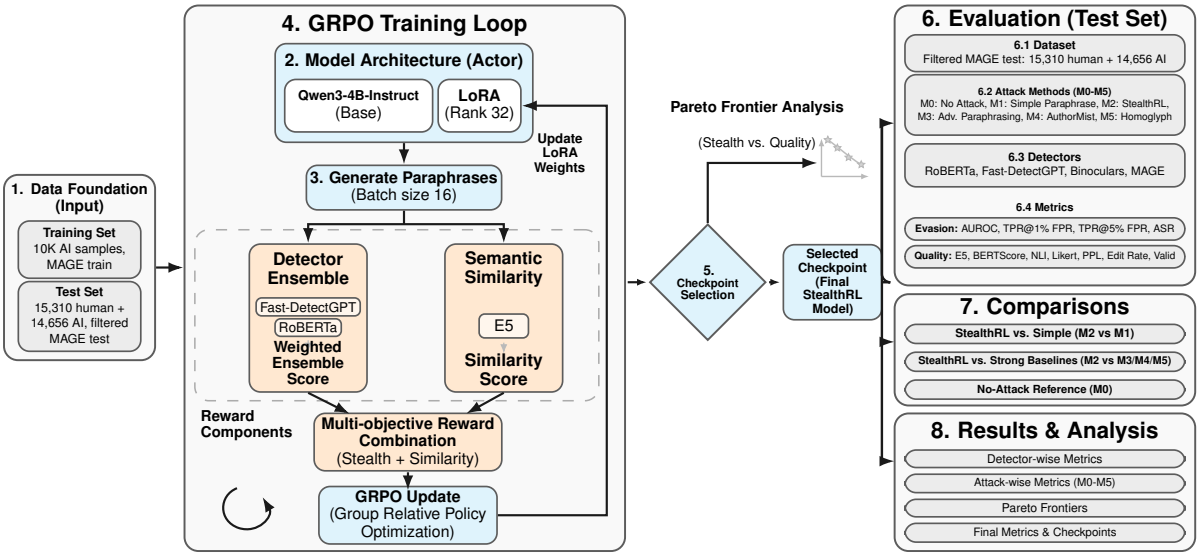


Figure 1: StealthRL training and evaluation pipeline. A paraphrase policy (Qwen3-4B with LoRA) is trained via GRPO against a two-detector ensemble (RoBERTa + Fast-DetectGPT) with semantic similarity reward. The trained policy is then evaluated against four detectors, including the held-out Binoculars and MAGE detectors, at the 1% FPR operating point.

only inside the RL reward stream; logged detector probabilities and all evaluation metrics use the unnormalized detector scores.

Semantic similarity reward. To prevent degenerate solutions (e.g., outputting empty or unrelated text), we constrain semantic preservation using E5 embedding cosine similarity (Wang et al., 2022):

$$R_{\text{sem}}(x, y) = \cos\big(\text{E5}(x),\ \text{E5}(y)\big). \qquad (5)$$

KL penalty. GRPO includes an implicit KL divergence penalty (coefficient $\lambda_{\text{KL}} = 0.05$) against the frozen reference policy π_{ref} to prevent catastrophic forgetting and maintain generation fluency.

3.3 Training Pipeline

We fine-tune Qwen3-4B-Instruct with LoRA adapters using GRPO (Pang and Jin, 2025). For each training batch, the policy samples a group of paraphrases, scores each candidate with the detector ensemble and E5 similarity reward, computes group-relative advantages, and updates the LoRA parameters under the KL penalty. The training ensemble comprises RoBERTa and Fast-DetectGPT; Binoculars and the MAGE detector are *held out* for transfer evaluation. Appendix Algorithm 1 provides the full pseudocode, and complete training, sampling, and compute settings are reported in Section 4.4.

3.4 Inference

At inference time, the fine-tuned policy generates a single paraphrase per input. No candidate selection or reranking is applied; the single-pass output is evaluated directly against all four detectors.

Accordingly, the reported StealthRL attack is *single-shot* at test time: each input receives one policy generation call, with no iterative refinement and no target-detector queries during evaluation. Detector access is required only during offline RL training. This differs from detector-guided search baselines such as M3, which use multi-candidate generation and detector-based reranking at evaluation time.

4 Experimental Setup

4.1 Dataset

We construct a custom subset of the MAGE (Machine-Generated Text Detection in the Wild) benchmark (Li et al., 2024), which provides human-written and AI-generated text across multiple domains. Our training set consists of 10,000 AI-generated samples drawn from the MAGE training split; no human-written samples are used during RL fine-tuning, as the policy learns entirely from detector feedback on paraphrased AI text. The evaluation set uses the full filtered MAGE test pool with 15,310 human-written and 14,656 AI-generated samples after applying the 100–500 to-

ken filter. Human samples in the evaluation set are used solely for threshold calibration at the 1% FPR operating point; AI and attacked samples are never used when setting thresholds.

4.2 Detectors

We evaluate against four detectors spanning distinct architectural paradigms:

- **RoBERTa OpenAI** (`openai-community/roberta-large-openai-detector`): A fine-tuned RoBERTa-large classifier trained on GPT-2 outputs. Represents the class of supervised neural classifiers.
- **Fast-DetectGPT** (Bao et al., 2024): A zero-shot statistical method that uses conditional probability curvature with scoring model `EleutherAI/gpt-neo-2.7B`. Represents curvature-based detection.
- **Binoculars** (Hans et al., 2024): A paired-LM detector using `gpt2-medium` and `gpt2-large` that compares cross-entropy and perplexity. *Held out during training* to assess transfer.
- **MAGE detector** (Li et al., 2024): The benchmark-provided Longformer-based classifier released with MAGE (model checkpoint `yaful/MAGE`). In the original MAGE paper, this PLM-based detector is the strongest method among their compared detectors across the benchmark's in-distribution and out-of-distribution testbeds, so we include it as a strong held-out evaluation target rather than a detector used during training.

4.3 Baselines

We compare six attack settings:

- **M0 (No Attack)**: Original AI-generated text, unmodified.
- **M1 (Simple Paraphrase)**: Standard LLM paraphrasing without detector feedback.
- **M2 (StealthRL)**: Our method, RL-trained paraphrasing against a detector ensemble (3 epochs, GRPO + LoRA).
- **M3 (Adversarial Paraphrasing)** (Cheng et al., 2025): Detector-guided candidate selection from multiple paraphrases.
- **M4 (AuthorMist)** (David and Gervais, 2025): RL-trained paraphrasing against a single detector (RoBERTa).
- **M5 (Homoglyph)** (Creo and Pudasaini, 2025): Character-level Unicode substitution (SilverSpeak).

Only M3 uses detector-guided search at evaluation time: it samples four candidates and reranks them using RoBERTa subject to an E5 similarity floor. StealthRL (M2) is evaluated as a single-shot policy with one generation per input and no detector-based reranking or evaluation-time detector queries. Appendix Tables 11 and 12 report the complete per-method query budget, candidate count, generator, prompt/decoding settings, reranking detector, and seed policy.

4.4 Implementation Details

StealthRL fine-tunes Qwen3-4B-Instruct with LoRA adapters (rank 32, LoRA $\alpha = 32$, dropout 0.05) for 3 epochs on 10,000 AI-generated MAGE training samples. GRPO uses batch size 16, group size 8, learning rate 2.8×10^{-4}, maximum 512 generated tokens, and KL coefficient $\lambda_{\text{KL}} = 0.05$ against the frozen reference policy. The inference prompt is "`Paraphrase the following text while preserving its meaning: [TEXT]`"; decoding uses temperature 1.0 and top-p 0.9. To test whether results depend on one stochastic draw, we additionally run two full-pool M2 repeats with the same checkpoint, prompt, and decoding configuration but different random seeds (4201 and 4202), scoring each repeat with the same detector pipeline and fixed thresholds.

The training reward uses RoBERTa and Fast-DetectGPT because they expose complementary detector signals while remaining fast enough for repeated reward calls inside GRPO. Binoculars and MAGE are excluded from training and reserved for held-out transfer evaluation. This design makes the evaluation stricter, but it also means our training ensemble is not exhaustive; scaling RL to larger and slower detector panels remains future work.

Appendix G reports the small-subset ablations used for model selection. These ablations were run on a 1,000-human / 1,000-AI MAGE selection subset and were used to choose the final reward setting before full-test evaluation; all main results in Section 5 are recomputed on the full filtered MAGE test pool.

4.5 Evaluation Metrics

We report detection metrics and quality diagnostics:

- **TPR@1%FPR**: True positive rate at 1% false positive rate, using a detector threshold calibrated as the empirical 99th percentile of scores on the 15,310 human evaluation samples. This is our primary high-precision operating point.
- **TPR@5%FPR**: The same metric at a looser 5% false positive rate, using the empirical 95th percentile of human scores. We report this as a supplementary operating point.
- **ASR** (Attack Success Rate): Fraction of AI samples classified as human at the 1% FPR threshold. ASR $= 1 -$ TPR@1%FPR.
- **AUROC**: Area under the receiver operating characteristic curve. Threshold-independent measure of overall discriminability.
- **E5 Similarity**: Cosine similarity between E5 embeddings (Wang et al., 2022) of original and paraphrased text, measuring sentence-level semantic preservation.
- **BERTScore**: Contextual token-alignment F1 (Zhang et al., 2020), which complements E5 by measuring local lexical and semantic overlap rather than only global embedding similarity.
- **NLI Consistency**: Bidirectional entailment consistency using a DeBERTa-large model fine-tuned on MNLI (He et al., 2021). We score both original→paraphrase and paraphrase→original directions and report the minimum entailment probability, so a high value requires meaning preservation in both directions.
- **Perplexity (PPL)**: GPT-2 perplexity of the output text. Lower values indicate lower language-model surprise under GPT-2; PPL is reported as a fluency diagnostic, not as a higher-is-better quality score.
- **Edit Rate**: $1 -$ `SequenceMatcher` character-level similarity between the source and output. It measures how much text changed; higher values are not inherently better because both trivial copying and excessive rewriting can be undesirable.
- **Valid**: Fraction of outputs passing the automatic validity filter: non-empty, at least 10 words, and no more than $3\times$ the original character length.

All confidence intervals are computed via bootstrap resampling with 500 iterations and seed 42.

4.6 LLM-Based Quality Judge

We additionally evaluate paraphrase quality with an LLM judge, following prior work on automatic text-quality evaluation (Fu et al., 2024; Kim et al., 2024; Vu et al., 2024). OpenAI `gpt-5-nano` scores each paraphrase on two 1–5 Likert axes: linguistic quality and semantic similarity. For fair cross-method comparison, we score a shared subset of 500 AI samples per method (M1–M5) using identical sample IDs; the judge sees only the source and paraphrase, without method labels. Appendix I reports the exact prompts.

5 Results

5.1 Main Detection Evasion Results

StealthRL (M2) achieves a **0.024 mean TPR@1%FPR**, reducing mean AUROC from 0.79 (no attack) to 0.43 with a 97.6% attack success rate. At this strict high-precision triage setting, the evaluated detector panel becomes largely ineffective against the adaptive attack. The full detector-by-detector AUC/TPR/ASR tables for the 1% FPR operating point are moved to Appendix Tables 1, 2, 3, and 4.

Figure 2 summarizes the main comparison across metrics. Panel (a) shows that StealthRL dramatically reduces AUROC across the detector panel, driving Binoculars and Fast-DetectGPT close to random chance and substantially degrading the remaining detectors. Panel (b) confirms that the mean AUROC of 0.432 for M2 is far below the no-attack baseline of 0.789. Panel (c) demonstrates near-zero TPR at strict operating points on three detectors, while Panel (d) shows the 97.6% attack success rate.

5.2 Robustness, Transfer, and Quality

StealthRL's evasion transfers to detectors not used during training: M2 reaches 0.002 TPR@1%FPR on held-out Binoculars and 0.089 on the held-out MAGE detector, alongside 0.002 on both in-ensemble detectors. The same ranking holds at the supplementary 5% FPR operating point, where M2 obtains 0.076 mean TPR@5%FPR (Appendix Table 5). Repeated stochastic generation is also stable: across the original run and two full-pool repeats, mean AUROC is 0.432 ± 0.001, mean

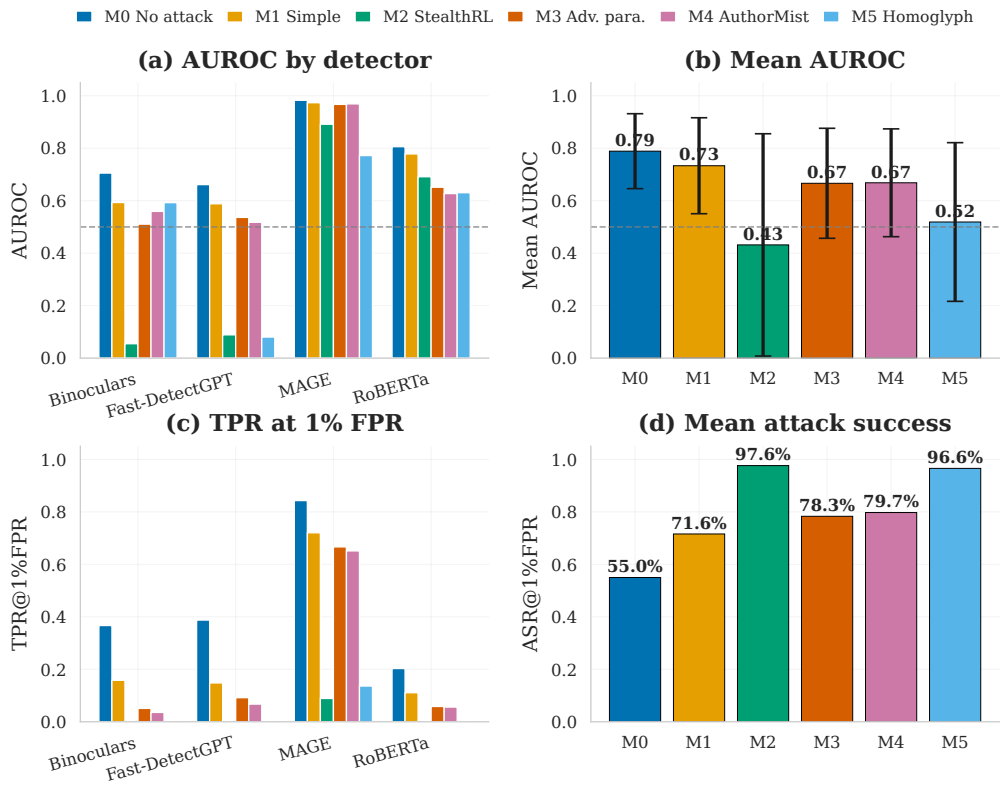


Figure 2: Detection evasion results for methods M0–M5. (a) AUROC by detector. (b) Mean AUROC with standard-deviation bars across detectors. (c) TPR at 1% FPR. (d) Mean attack success rate. StealthRL (M2, teal) achieves extremely low TPR on Binoculars, Fast-DetectGPT, and RoBERTa, with MAGE remaining the most robust detector in the panel.

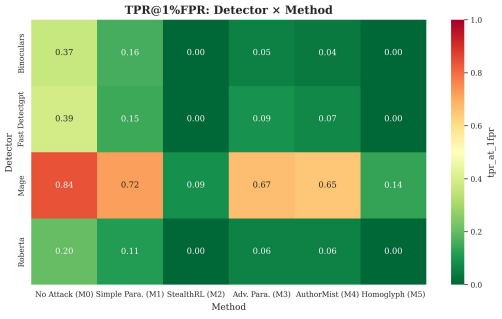


Figure 3: TPR@1%FPR heatmap. Lower is better for the attacker; M2 collapses TPR on three detectors and degrades held-out MAGE.

TPR@1%FPR is 0.024 ± 0.001, and mean ASR is 0.976 ± 0.001 (Appendix Table 6).

Figure 3 and Appendix Figure 4 provide the per-detector heatmap and score distributions. The distributions show that M2 pushes most AI-sample scores below the calibrated 1% FPR threshold, explaining why TPR collapses even when a detector such as RoBERTa retains some global rank information. This pattern is consistent with cross-detector vulnerabilities within our panel, but we do not claim a universal mechanism for all detector families.

The attack also exposes an evasion–fidelity tradeoff. M2 obtains near-maximal mean ASR (0.976) with E5 similarity 0.901, BERTScore 0.868, NLI 0.316, and judged quality 2.51 (Appendix Table 7). Thus, M2 is more readable than the homoglyph baseline (M5 quality 1.97) but less semantically faithful than high-quality paraphrasing baselines M1/M3. We therefore interpret StealthRL as a robustness stress test rather than a quality-equivalent paraphraser.

5.3 Discussion

These results show that, for the evaluated detector panel, adaptive paraphrasing can defeat high-precision detector-assisted review workflows with one generation per input and no evaluation-time detector reranking. Human review changes how detector alerts should be used, but it does not remove the need for robust triage signals: low precision overloads reviewers, while low TPR

fails to surface adversarially paraphrased AI text for review. Purpose-built adversarial training, provenance-aware defenses, and more semantically grounded detectors are therefore necessary next steps.

6 Conclusion and Future Work

StealthRL demonstrates severe detector failure under adaptive RL-based paraphrasing attacks, revealing significant robustness gaps in the AI-text detectors evaluated. By training against a multi-detector ensemble with GRPO and LoRA, we achieve near-zero detection on three of four detectors and a 0.024 mean TPR@1%FPR with strong cross-architecture transfer, including to two held-out detectors. Our evaluation, spanning detection metrics, stochastic repeat analysis, multi-metric quality assessment, and score distribution analysis, characterizes the evasion–quality tradeoff on this benchmark.

Future work should study adversarial training, provenance-aware defenses, semantic and discourse-level detection signals, stronger constrained RL objectives, and broader datasets, languages, and detector families. Our evaluation framework and released code provide a rigorous testbed for measuring progress on adversarially robust AI-text detection.

7 Limitations

Detector coverage. Our evaluation covers four detectors spanning supervised, zero-shot statistical, paired-LM, and benchmark-provided paradigms. We do not evaluate against watermark-based detectors (Kirchenbauer et al., 2023), which embed signals during generation and may be more robust to paraphrasing attacks. Evaluating StealthRL against watermarked text is an important direction for future work.

Training-ensemble coverage. The RL reward uses two detectors, RoBERTa and Fast-DetectGPT, while Binoculars and MAGE are held out for transfer evaluation. This separation is useful for measuring cross-detector transfer, but it also means we do not study whether training directly against a larger detector ensemble would further improve or destabilize evasion. Larger ensembles would require substantially more reward-model calls per GRPO update, especially for paired-LM and Longformer-based detectors, and are left for future work.

Dataset diversity. We evaluate on a single benchmark (MAGE) in English. Broader coverage across datasets (RAID (Dugan et al., 2024)), domains, and languages is needed to establish generalizability. The MAGE benchmark, while diverse, may not capture all deployment scenarios.

Quality gap. StealthRL achieves lower semantic fidelity than the high-quality paraphrase baselines (E5 0.901, BERTScore 0.868, and NLI 0.316, compared with M1/M3 E5 0.974/0.973, BERTScore 0.926/0.934, and NLI 0.967/0.965) and lower Likert quality (2.51 vs. 3.78/3.77). It still outperforms character-level obfuscation on BERTScore and judged quality (M5: 0.773 and 1.97), but the remaining fidelity cost is a limitation of the current system. We supplement E5 with BERTScore, bidirectional NLI, and LLM-based similarity judgments, but all automatic semantic metrics remain imperfect proxies for human meaning preservation. Improving semantic preservation while maintaining strong evasion is an important direction. Techniques such as constrained decoding, rejection sampling, or multi-objective RL could help close this gap.

Defense evaluation. We do not explore defensive strategies such as adversarial training, certified robustness, or ensemble diversification that could improve detector resilience. Our focus is on exposing vulnerabilities to motivate defensive research.

8 Ethical Considerations

Adversarial paraphrasing is dual-use technology. We position StealthRL as a *stress-testing and robustness evaluation tool* for researchers and detector developers, not a production evasion system. The near-zero TPR@1%FPR result exposes critical vulnerabilities that must be addressed before detectors are deployed in high-stakes applications such as academic integrity enforcement.

We release our code and evaluation pipeline to enable reproducible robustness assessment and to accelerate defensive research. By making attack capabilities transparent, we aim to shift the detector development paradigm toward adversarial robustness rather than clean-distribution accuracy. We believe responsible disclosure of detector vulnerabilities, accompanied by tools for measuring progress, serves the broader goal of trustworthy AI-text detection.

Acknowledgments

We gratefully acknowledge Thinking Machines for providing free research credits and access to their Tinker API framework, which made the RL fine-tuning possible. 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. 2024. Fast-DetectGPT: Efficient zero-shot detection of machine-generated text via conditional probability curvature. In *The Twelfth International Conference on Learning Representations*.

Yize Cheng, Vinu Sankar Sadasivan, Mehrdad Saberi, Shoumik Saha, and Soheil Feizi. 2025. Adversarial paraphrasing: A universal attack for humanizing AI-generated text. In *Advances in Neural Information Processing Systems 38*.

Aldan Creo and Shushanta Pudasaini. 2025. SilverSpeak: Evading AI-generated text detectors using homoglyphs. In *Proceedings of the 1st Workshop on GenAI Content Detection (GenAIDetect)*, pages 1–46, Abu Dhabi, UAE. International Conference on Computational Linguistics.

Isaac David and Arthur Gervais. 2025. AuthorMist: Evading AI text detectors with reinforcement learning. *Preprint*, arXiv:2503.08716.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. QLoRA: Efficient finetuning of quantized LLMs. In *Advances in Neural Information Processing Systems 36*.

Liam Dugan, Alyssa Hwang, Filip Trhlík, Andrew Zhu, Josh Magnus Ludan, Hainiu Xu, Daphne Ippolito, and Chris Callison-Burch. 2024. RAID: A shared benchmark for robust evaluation of machine-generated text detectors. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12463–12492, Bangkok, Thailand. Association for Computational Linguistics.

Jinlan Fu, See-Kiong Ng, Zhengbao Jiang, and Pengfei Liu. 2024. GPTScore: Evaluate as you desire. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6556–6576, Mexico City, Mexico. Association for Computational Linguistics.

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. In *Proceedings of the 41st International Conference on Machine Learning*.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. DeBERTa: Decoding-enhanced BERT with disentangled attention. In *International Conference on Learning Representations*.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations*.

Seungone Kim, Juyoung Suk, Shayne Longpre, Bill Yuchen Lin, Jamin Shin, Sean Welleck, Graham Neubig, Moontae Lee, Kyungjae Lee, and Minjoon Seo. 2024. Prometheus 2: An open source language model specialized in evaluating other language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 4334–4353, Miami, Florida, USA. Association for Computational Linguistics.

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*.

Kalpesh Krishna, Yixiao Song, Marzena Karpinska, John 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 36*.

Yafu Li, Qintong Li, Leyang Cui, Wei Bi, Zhilin Wang, Longyue Wang, Linyi Yang, Shuming Shi, and Yue Zhang. 2024. MAGE: Machine-generated text detection in the wild. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 36–53, Bangkok, Thailand. Association for Computational Linguistics.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D. Manning, and Chelsea Finn. 2023. DetectGPT: Zero-shot machine-generated text detection using probability curvature. In *Proceedings of the 40th International Conference on Machine Learning*.

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

Lei Pang and Ruinan Jin. 2025. On the theory and practice of GRPO: A trajectory-corrected approach with fast convergence. *Preprint*, arXiv:2508.02833. Version 1.

Shinwoo Park, Shubin Kim, Do-Kyung Kim, and Yo-Sub Han. 2025. KatFishNet: Detecting LLM-generated Korean text through linguistic feature analysis. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 21189–21222, Vienna, Austria. Association for Computational Linguistics.

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

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *Preprint*, arXiv:1707.06347.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *Preprint*, arXiv:2402.03300.

Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, and Jasmine Wang. 2019. Release strategies and the social impacts of language models. *Preprint*, arXiv:1908.09203.

Brian Tufts, Xuandong Zhao, and Lei Li. 2025. A practical examination of AI-generated text detectors for large language models. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 4839–4856, Albuquerque, New Mexico. Association for Computational Linguistics.

Turnitin. 2023. Understanding false positives within our AI writing detection capabilities. Accessed 2026-05-19.

Tu Vu, Kalpesh Krishna, Salaheddin Alzubi, Chris Tar, Manaal Faruqui, and Yun-Hsuan Sung. 2024. Foundational autoraters: Taming large language models for better automatic evaluation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 17086–17105, Miami, Florida, USA. Association for Computational Linguistics.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text embeddings by weakly-supervised contrastive pre-training. *Preprint*, arXiv:2212.03533.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating text generation with BERT. In *International Conference on Learning Representations*.

A Training Algorithm

Algorithm 1 StealthRL Training

Require: Training data $\mathcal{D} = \{x_i\}_{i=1}^{N}$ (AI-generated texts), detector ensemble $\{d_k\}_{k=1}^{K}$ with weights $\{w_k\}$, E5 similarity model, base LLM π_{ref}

Ensure: Fine-tuned paraphrase policy π_θ

Initialize $\pi_\theta \leftarrow \pi_{\text{ref}}$ with LoRA adapters (rank r=32, α=32)
for epoch $= 1, \ldots, E$ **do**
  for batch $\mathcal{B} \subset \mathcal{D}$ **do**
    for each $x \in \mathcal{B}$ **do**
      Sample group $\{y_1, \ldots, y_G\} \sim \pi_\theta(\cdot \mid x)$ ▷ G=8
      for each y_g in group **do**
        $R_{\text{det}}(y_g) \leftarrow 1 - \sum_k w_k \cdot d_k(y_g)$
        $R_{\text{sem}}(x, y_g) \leftarrow \cos(\text{E5}(x), \text{E5}(y_g))$
        $R(x, y_g) \leftarrow \alpha R_{\text{det}}(y_g) + \beta R_{\text{sem}}(x, y_g)$
      end for
      Compute group-relative advantages: $A_g \leftarrow (R_g - \bar{R})/\sigma_R$
    end for
    Update θ via clipped policy gradient with KL penalty
  end for
end for
return π_θ

B Full Main Results on the MAGE Test Pool

Method	Binoc. AUC	Binoc. TPR	Binoc. ASR
M0	0.705	0.367	0.633
M1	0.593	0.158	0.842
M2 (Ours)	**0.055**	0.002	0.998
M3	0.511	0.051	0.949
M4	0.559	0.036	0.964
M5	0.593	**0.000**	**1.000**

Table 1: Main results on the full filtered MAGE test pool at the 1% FPR operating point for Binoculars. Lower TPR@1%FPR/AUROC is better for the attacker; higher ASR is better. Bold indicates the best value per metric.

Method	FastDGPT AUC	FastDGPT TPR	FastDGPT ASR
M0	0.661	0.388	0.612
M1	0.588	0.148	0.852
M2 (Ours)	0.089	0.002	0.998
M3	0.536	0.092	0.908
M4	0.518	0.067	0.933
M5	**0.080**	**0.000**	**1.000**

Table 2: Main results on the full filtered MAGE test pool at the 1% FPR operating point for Fast-DetectGPT. Lower TPR@1%FPR/AUROC is better for the attacker; higher ASR is better. Bold indicates the best value per metric.

Method	MAGE AUC	MAGE TPR	MAGE ASR
M0	0.983	0.843	0.157
M1	0.973	0.720	0.280
M2 (Ours)	0.891	**0.089**	**0.911**
M3	0.967	0.666	0.334
M4	0.969	0.651	0.349
M5	**0.772**	0.136	0.864

Table 3: Main results on the full filtered MAGE test pool at the 1% FPR operating point for MAGE. Lower TPR@1%FPR/AUROC is better for the attacker; higher ASR is better. Bold indicates the best value per metric.

Method	RoBERTa AUC	RoBERTa TPR	RoBERTa ASR	Mean TPR	Mean ASR
M0	0.806	0.203	0.797	0.450	0.550
M1	0.778	0.111	0.889	0.284	0.716
M2 (Ours)	0.691	0.002	0.998	**0.024**	**0.976**
M3	0.651	0.058	0.942	0.217	0.783
M4	**0.627**	0.056	0.944	0.203	0.797
M5	0.630	**0.001**	**0.999**	0.034	0.966

Table 4: Main results on the full filtered MAGE test pool at the 1% FPR operating point, part II. Lower TPR@1%FPR/AUROC is better for the attacker; higher ASR is better. This panel reports RoBERTa and the mean TPR@1%FPR / mean ASR aggregates across the four-detector panel. Bold indicates the best value per metric.

Method	Binoc. TPR@5%	FastDGPT TPR@5%	MAGE TPR@5%	RoBERTa TPR@5%	Mean TPR@5%
M0	0.481	0.451	0.934	0.397	0.566
M1	0.269	0.252	0.876	0.294	0.423
M2 (Ours)	0.005	0.003	0.272	**0.022**	0.076
M3	0.135	0.180	0.846	0.159	0.330
M4	0.126	0.130	0.839	0.158	0.313
M5	**0.004**	**0.000**	**0.238**	0.027	**0.067**

Table 5: Supplementary TPR@5%FPR results on the full filtered MAGE test pool. Lower is better for the attacker. Thresholds are recalibrated on the 15,310 human evaluation samples at the 5% false-positive operating point. Bold indicates the lowest TPR@5%FPR per column.

Run	Mean AUROC	Mean TPR	Mean ASR	Binoc.	Fast-DGPT	MAGE	RoBERTa
Original M2	0.432	0.024	0.976	0.002	0.002	0.089	0.002
Seed 4201	0.432	0.025	0.975	0.002	0.002	0.093	0.002
Seed 4202	0.433	0.024	0.976	0.002	0.002	0.091	0.002
Mean ± SD	0.432 ± 0.001	0.024 ± 0.001	0.976 ± 0.001	0.002 ± 0.000	0.002 ± 0.000	0.091 ± 0.002	0.002 ± 0.000

Table 6: Full-pool stochastic repeat analysis for StealthRL (M2) at temperature 1.0 and top-p 0.9. All runs use one generation per input and no reranking; thresholds are the same 1% FPR thresholds calibrated on human evaluation samples. Detector columns report TPR@1%FPR.

C Quality Metrics

Method	E5 Sim.	BERTScore	NLI	PPL	Edit Rate	Valid	Quality	Similarity
M0	1.000	1.000	0.991	26.662	0.000	1.000	—	—
M1	0.974	0.926	0.967	24.596	0.779	1.000	3.780	4.058
M2 (Ours)	0.901	0.868	0.316	148.686	0.835	1.000	2.514	2.644
M3	0.973	0.934	0.965	28.338	0.782	1.000	3.770	4.024
M4	0.958	0.902	0.829	18.909	0.861	0.998	3.152	3.704
M5	0.923	0.773	0.847	146.430	0.634	1.000	1.972	2.920

Table 7: Quality and similarity metrics. E5 Sim., BERTScore, and NLI are computed on the full filtered evaluation pool. Quality and Similarity are mean Likert scores (1–5) from a gpt-5-nano judge on 500 matched samples for methods M1–M5; M0 reports automatic metrics only. Higher is better for E5, BERTScore, NLI, Valid, Quality, and Similarity. Lower PPL indicates lower GPT-2 language-model surprise, while Edit Rate is a magnitude-of-change diagnostic rather than a quality-positive score.

D Qualitative Examples

Tables 8–10 show representative paraphrases across different domains from the filtered MAGE test split.

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 8: Representative paraphrases from the filtered MAGE test split (health and automotive domains).

Original (M0)	**StealthRL (M2)**
Machine learning algorithms have become increasingly sophisticated in recent years, with deep neural networks achieving state-of-the-art performance across a wide range of tasks including image classification, natural language processing, and reinforcement learning.	Recent advances in machine learning have led to highly sophisticated algorithms. Deep neural networks now achieve top performance in diverse areas, from classifying images and processing natural language to learning through reinforcement-based approaches.
The process of photosynthesis converts light energy into chemical energy that can be stored in glucose molecules. This process occurs in the chloroplasts of plant cells and involves two main stages: the light-dependent reactions and the Calvin cycle.	Photosynthesis transforms light energy into chemical energy, storing it within glucose molecules. Taking place in plant cell chloroplasts, this process unfolds in two key stages: light-dependent reactions followed by the Calvin cycle.

Table 9: Representative paraphrases (science and technology domains).

Original (M0)	**StealthRL (M2)**
The global economy faces several interconnected challenges, including rising inflation, supply chain disruptions, and geopolitical tensions. Central banks worldwide have responded by tightening monetary policy, raising interest rates to combat price pressures.	Several intertwined challenges confront the global economy: climbing inflation, supply chain breakdowns, and geopolitical friction. In response, central banks around the world have tightened monetary policy and hiked interest rates to curb rising prices.
Renewable energy sources such as solar and wind power have seen dramatic cost reductions over the past decade, making them competitive with traditional fossil fuels in many markets. Government incentives and technological improvements continue to drive adoption rates higher.	Solar and wind power costs have fallen dramatically over the last decade, allowing renewables to compete with fossil fuels across numerous markets. Ongoing government incentives and technological progress continue pushing adoption rates upward.

Table 10: Representative paraphrases (economics and energy domains).

E Attack Protocol and Query Budgets

Prompt templates. M1 and M3 use the standard prompt "`Please paraphrase the following text while maintaining its meaning and style. Keep the paraphrase close to the original length, do not add new details, and output only the paraphrased text without any additional explanation.`" M2 uses the StealthRL prompt reported in Ta-

ble 13. M4 uses the AuthorMist humanization prompt "`Please paraphrase the following text to make it more human-like while preserving the original meaning. Keep the paraphrase close to the original length and do not add new details.`" M0 and M5 use no LLM prompt.

Method	Cand.	Generator / transform	Decoding / seed policy
M0	1	Original AI text	Deterministic
M1	1	Qwen/ Qwen3-4B-Instruct-2507 via vLLM	Temp. 0.9, top-p 0.95; vLLM default engine seed
M2 (ours)	1	StealthRL LoRA sampler on Qwen/ Qwen3-4B-Instruct-2507	Temp. 1.0, top-p 0.9; original run plus repeat seeds 4201/4202
M3	4	Qwen/ Qwen3-4B-Instruct-2507 via vLLM	Temp. 1.0, top-p 0.95; vLLM default engine seed
M4	1	`authormist/ authormist-originality` via vLLM	Temp. 0.7, top-p 0.9; vLLM default engine seed
M5	1	SilverSpeak homoglyph transform	Substitution rate 0.10; seed 42

Table 11: Per-method generation protocol. Table 13 reports the M2 training/evaluation configuration.

Method	Selection and attack-time query budget
M0	No generation and no detector queries.
M1	No reranking; 0 detector queries; 1 generation call.
M2 (ours)	No reranking; 0 evaluation-time detector queries; 1 policy generation call.
M3	Select lowest RoBERTa score among candidates with E5 ≥ 0.90; fallback to lowest RoBERTa score; 4 RoBERTa queries and 4 E5 scores per sample.
M4	No reranking; 0 detector queries; 1 generation call.
M5	Deterministic character substitution; 0 detector queries and 0 LLM calls.

Table 12: Per-method selection and query budget. The budget counts detector calls used to construct an attack output, not the post-hoc evaluation pass in which every final output is scored once by each evaluation detector.

F 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	3
Training samples	10,000 AI-generated (custom MAGE subset)
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)
MAGE detector	Longformer classifier (`yaful/MAGE`, held-out)
Evaluation	
Test samples	15,310 human + 14,656 AI (filtered MAGE test)
Token window	100–500 tokens
FPR calibration	1% on 15,310 human samples (quantile)
M2 candidates per sample	1 (see Table 11 for M0–M5)
Compute	
Training framework	Tinker API (Thinking Machines)
Reward computation	8× NVIDIA RTX A5000 GPUs
Offline generation/evaluation	8× NVIDIA RTX A5000 GPUs
Execution mode	Local GPU jobs; detector scoring distributed across A5000 workers
Software	Python 3.12.3, PyTorch 2.10.0, Transformers 4.57.6, vLLM 0.17.1
Seed	42

Table 13: Complete hyperparameters and configuration for reproducibility.

G Model-Selection Ablations

The main results use the full filtered MAGE test pool (15,310 human / 14,656 AI). Before running that expensive evaluation, we used smaller ablations for model selection. These ablations are diagnostic: they use smaller subsets, fewer detectors, or training-dashboard metrics, and are therefore not directly comparable to the full-test numbers in Tables 1–4.

Table 14 reports the initial reward-weight sweep. All runs use RoBERTa and Fast-DetectGPT as the training detectors and are evaluated with the training-loop detector score rather than the final low-FPR detector panel. The sweep identifies α=1.0, β=0.1 as the best balanced setting: it has the lowest detector probability and highest evasion rate while retaining high E5 similarity. Increasing the semantic weight to 0.3 does not improve fidelity and weakens evasion, while removing detector reward makes the objective less

aligned with the attack goal.

Reward setting	Detector prob.	Evasion rate	E5 Sim.	Total reward
α=1.0, β=0.0	0.331	0.669	0.978	-0.136
α=1.0, β=0.1 (selected)	0.226	0.774	0.970	0.127
α=1.0, β=0.3	0.327	0.673	0.968	0.200
α=0.0, β=1.0	0.359	0.641	0.964	0.970

Table 14: Training-time reward-weight ablation on the small MAGE development setup. Detector probability is the mean training-ensemble probability of AI generation, so lower detector probability and higher evasion rate are better for the attacker. This diagnostic sweep selected the nonzero but small semantic weight β=0.1.

Table 15 reports the later checkpoint-selection evaluation on a fixed 1,000-human / 1,000-AI MAGE selection set. This pass uses the same low-FPR calibration procedure as the main evaluation, but only three detectors (RoBERTa, Fast-DetectGPT, and Binoculars); MAGE was kept for the final full-panel evaluation. The selected α=1.0, β=0.1 checkpoint improves low-FPR evasion and generation quality relative to the no-semantic-reward checkpoint, and the replicate shows the same quality-preserving trend.

Selection run	Mean AUROC	Mean TPR@1%FPR	Mean ASR	E5 Sim.	PPL	Judge quality
No semantic reward / low-fidelity checkpoint	0.460	0.089	0.911	0.909	91.5	1.63
α=1.0, β=0.1 (selected)	0.546	0.050	0.950	0.973	31.7	4.10
α=1.0, β=0.1 (replicate)	0.554	0.063	0.937	0.974	30.5	–

Table 15: Checkpoint-selection ablations on a fixed 1,000-human / 1,000-AI MAGE selection set. Metrics are averaged across RoBERTa, Fast-DetectGPT, and Binoculars; lower AUROC and TPR are better for the attacker, while higher ASR and quality metrics are better. The full paper results are recomputed separately on the full filtered MAGE test pool.

We also ran a candidate-budget sweep for the selected M2 policy. Increasing candidates from 1 to 2, 4, and 8 raised mean ASR from 0.997 to 1.000 on the small selection setup, while E5 similarity remained in a narrow 0.898–0.904 band. We therefore report the stricter single-shot setting in the main paper: one generation per input, with no test-time detector-guided search or reranking.

H Per-Detector Results with Confidence Intervals

Tables 16–19 provide complete per-detector, per-method results with 95% bootstrap confidence intervals (500 iterations). We split the table by detector to keep confidence intervals legible in the two-column format.

Method	AUROC [95% CI]	TPR@1%FPR [95% CI]	ASR [95% CI]
M0 No Attack	0.806 [0.801, 0.810]	0.203 [0.196, 0.209]	0.797 [0.791, 0.804]
M1 Simple Para.	0.778 [0.773, 0.783]	0.111 [0.106, 0.116]	0.889 [0.884, 0.894]
M2 StealthRL (Ours)	0.691 [0.685, 0.697]	0.002 [0.002, 0.003]	0.998 [0.997, 0.998]
M3 Adv. Para.	0.651 [0.645, 0.657]	0.058 [0.055, 0.062]	0.942 [0.938, 0.945]
M4 AuthorMist	**0.627** [0.620, 0.633]	0.056 [0.052, 0.059]	0.944 [0.941, 0.948]
M5 Homoglyph	0.630 [0.624, 0.636]	**0.001** [0.000, 0.001]	**0.999** [0.999, 1.000]

Table 16: RoBERTa results with 95% bootstrap confidence intervals. Bold indicates the best attacker value per metric.

Method	AUROC [95% CI]	TPR@1%FPR [95% CI]	ASR [95% CI]
M0 No Attack	0.661 [0.655, 0.668]	0.388 [0.380, 0.396]	0.612 [0.604, 0.620]
M1 Simple Para.	0.588 [0.582, 0.595]	0.148 [0.143, 0.154]	0.852 [0.846, 0.857]
M2 StealthRL (Ours)	0.089 [0.086, 0.092]	0.002 [0.001, 0.002]	0.998 [0.998, 0.999]
M3 Adv. Para.	0.536 [0.530, 0.543]	0.092 [0.087, 0.097]	0.908 [0.903, 0.913]
M4 AuthorMist	0.518 [0.511, 0.524]	0.067 [0.064, 0.072]	0.933 [0.928, 0.936]
M5 Homoglyph	**0.080** [0.078, 0.083]	**0.000** [0.000, 0.000]	**1.000** [1.000, 1.000]

Table 17: Fast-DetectGPT results with 95% bootstrap confidence intervals. Bold indicates the best attacker value per metric.

Method	AUROC [95% CI]	TPR@1%FPR [95% CI]	ASR [95% CI]
M0 No Attack	0.705 [0.699, 0.712]	0.367 [0.360, 0.375]	0.633 [0.625, 0.640]
M1 Simple Para.	0.593 [0.587, 0.599]	0.158 [0.152, 0.163]	0.842 [0.837, 0.848]
M2 StealthRL (Ours)	**0.055** [0.052, 0.058]	0.002 [0.001, 0.003]	0.998 [0.997, 0.999]
M3 Adv. Para.	0.511 [0.504, 0.517]	0.051 [0.048, 0.055]	0.949 [0.945, 0.952]
M4 AuthorMist	0.559 [0.553, 0.566]	0.036 [0.033, 0.038]	0.964 [0.962, 0.967]
M5 Homoglyph	0.593 [0.587, 0.599]	**0.000** [0.000, 0.000]	**1.000** [1.000, 1.000]

Table 18: Binoculars results with 95% bootstrap confidence intervals. Bold indicates the best attacker value per metric.

Method	AUROC [95% CI]	TPR@1%FPR [95% CI]	ASR [95% CI]
M0 No Attack	0.983 [0.981, 0.984]	0.843 [0.837, 0.849]	0.157 [0.151, 0.163]
M1 Simple Para.	0.973 [0.972, 0.975]	0.720 [0.713, 0.727]	0.280 [0.273, 0.287]
M2 StealthRL (Ours)	0.891 [0.888, 0.895]	**0.089** [0.084, 0.093]	**0.911** [0.907, 0.916]
M3 Adv. Para.	0.967 [0.965, 0.969]	0.666 [0.658, 0.674]	0.334 [0.326, 0.342]
M4 AuthorMist	0.969 [0.967, 0.971]	0.651 [0.644, 0.658]	0.349 [0.342, 0.356]
M5 Homoglyph	**0.772** [0.767, 0.777]	0.136 [0.130, 0.141]	0.864 [0.859, 0.870]

Table 19: MAGE detector results with 95% bootstrap confidence intervals. Bold indicates the best attacker value per metric.

I LLM Judge Prompt Templates

We use separate gpt-5-nano prompts for quality and semantic similarity. Method labels are never shown to the judge. The exact quality-rating system prompt is:

> You are an expert linguist and paraphrase evaluator. Your task is to assess the overall writing quality of a paraphrased text compared to the original source text.
>
> Judge the paraphrase as real writing, not as an attack artifact. Reward unusual sentence structure only when it still reads naturally. Penalize awkward clause order, stilted transitions, filler phrases, obvious prompt leakage, parenthetical meta-comments, detector-evasion artifacts, and phrasing that feels unnatural even if the meaning is preserved.
>
> Be strict about naturalness. Use the full 1-5 scale and avoid defaulting to 3 unless it truly fits.
>
> Scoring criteria: 5 - Excellent: Fluent, grammatical, and natural throughout; reads like polished human writing. 4 - Good: Clear and mostly natural with only minor awkwardness. 3 - Acceptable: Understandable, but noticeably stiff, awkward, or uneven. 2 - Poor: Readably degraded; unnatural syntax or phrasing frequently distracts. 1 - Unusable: Severely broken, incoherent, or clearly unnatural.

Provide your final output as a JSON object in this format: `{"score": <score from 1 to 5>, "justification": "<brief explanation>"}`.

The quality-rating user prompt template is:

Evaluate the following paraphrase using the criteria above (focus on fluency and overall writing quality):

Original Text: `"""{original_text}"""`

Paraphrased Text: `"""{paraphrased_text}"""`

What score (1 to 5) would you assign to the paraphrase's quality, and why? Do not give extra credit for sounding evasive or unusual if the prose becomes less natural.

The exact semantic-similarity system prompt is:

You are an expert linguist and paraphrase evaluator. Your task is to assess how well the paraphrased text preserves the meaning of the original source text.

Be generous when meaning is substantially preserved. Use the full 1-5 scale and avoid defaulting to 3 unless it truly fits. If the paraphrase keeps the core meaning with only minor changes, prefer 4 or 5. Use 2 or 1 only for clear meaning loss or topic drift.

Scoring criteria: 5 - Approximately equivalent: Meaning preserved; only wording/structure changes. 4 - Nearly equivalent: Meaning mostly preserved; small factual or emphasis shifts. 3 - Somewhat equivalent: Partial meaning preserved; important details differ. 2 - Topically related: Same topic but most meaning lost. 1 - Not related: Different topic or meaning.

Provide your final output as a JSON object in this format: `{"score": <score from 1 to 5>, "justification": "<brief explanation>"}`.

The semantic-similarity user prompt template is:

Evaluate the following paraphrase using the criteria above:

Original Text: `"""{original_text}"""`

Paraphrased Text: `"""{paraphrased_text}"""`

What score (1 to 5) would you assign for semantic similarity, and why?

J Dataset Statistics

Stage	Purpose	Human Samples	AI Samples
Train	RL fine-tuning (GRPO)	0	10,000
Test	Detection evaluation	15,310	14,656
Likert eval	Quality scoring (per method)	0	500
Token range	100–500 tokens		
Source	MAGE benchmark (Li et al., 2024)		
Domains	Multiple (essays, Q&A, creative writing)		

Table 20: Dataset statistics for training and evaluation. The Likert evaluation uses 500 AI samples per attack method (M1–M5) with matched sample IDs.

K Supplementary Figures

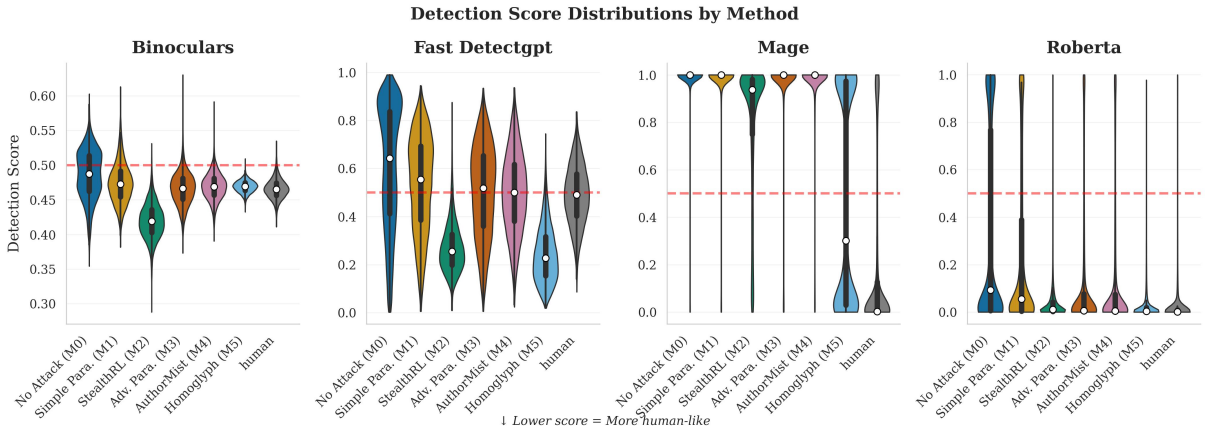


Figure 4: Detector score distributions for AI samples across methods (one panel per detector). StealthRL (M2) and Homoglyph (M5) push scores below the detection threshold, explaining their near-zero TPR@1%FPR.

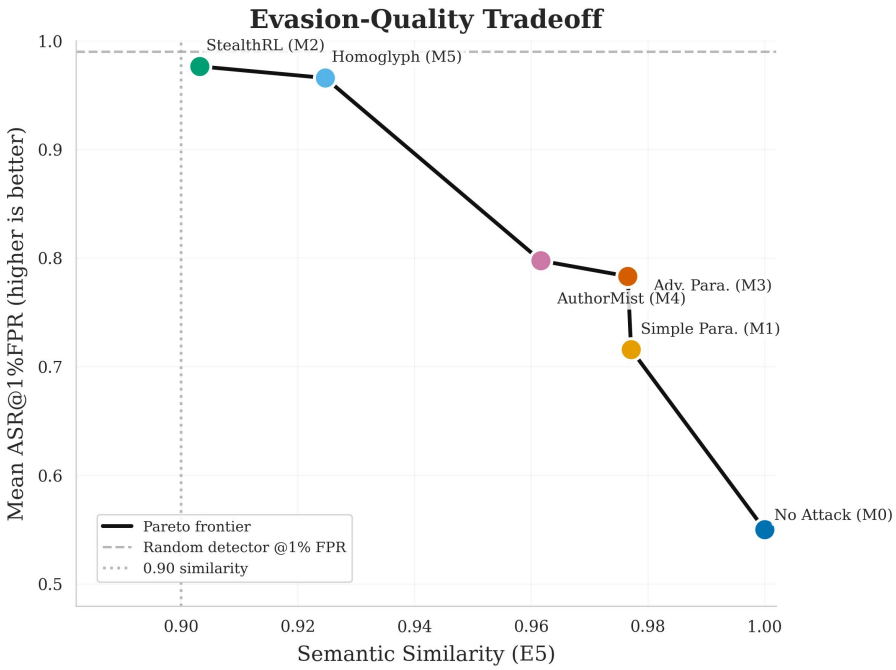


Figure 5: Evasion–quality tradeoff at the 1% FPR operating point. Each point represents a method, plotted by E5 semantic similarity (x-axis) against mean ASR@1%FPR (y-axis). Top-right is ideal (high similarity, high attack success). The dashed lines mark the 0.90 similarity floor and the random-detector ASR baseline at 1% FPR, and the solid line shows the Pareto frontier.

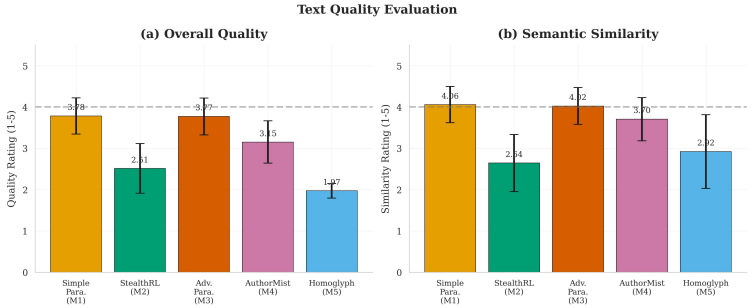


Figure 6: LLM-based quality evaluation on 500 matched samples. Panel (a) shows linguistic quality and panel (b) semantic similarity; higher is better. The dashed line marks the neutral midpoint (3.0).