Detector-Evasive LLM Paraphrasing via Constrained Policy Optimization

Anonymous ACL submission

Abstract

AI-text detectors are vulnerable to paraphrasing and detector-guided paraphrasing attacks, but existing detector-evasion methods often lack precise control over semantic preservation. In particular, optimizing directly for detector evasion can degrade fine-grained semantics, whereas scalarized reward designs provide only indirect, weight-sensitive control over the evasion–semantics trade-off. We address this limitation by formulating detector-evasive LLM paraphrasing as a Constrained Markov Decision Process, where detector evasion is the primary objective and semantic preservation is enforced as an explicit constraint. We propose Detector Evasion Policy Optimization (DEPO), a Lagrangian primal-dual reinforcement learning algorithm with a novel GRPO-style group-based policy update. DEPO adaptively balances semantic preservation and detector evasion during training, enabling the policy to improve attack success within a prescribed semantic-preservation region. Experiments on MAGE, M4, RAID, and peer-review datasets, evaluated against MAGE, RoBERTa, RADAR, Binoculars, and Fast-DetectGPT detectors, show that DEPO achieves strong detector evasion while precisely satisfying the semantic preservation constraint. DEPO also exhibits cross-domain, cross-detector, and prompt-level robustness.

1 Introduction

As large language models (LLMs) are increasingly adopted across diverse domains, identifying AI-generated text has become an important and timely challenge due to concerns about its misuse, e.g., academic dishonesty (Perkins et al., 2023) and spread of fake news (Hanley and Durumeric, 2024). This demand has led to the development of various AI-text detectors to distinguish machine-generated content from human-written text, e.g., (Wu et al., 2025; Yang et al., 2024b; Kumarage et al., 2024; Kehkashan et al., 2025). Nevertheless, AI text detectors are inherently vulnerable, and their predictions can be severely compromised by various counter-detection strategies, e.g., token-level perturbations, paraphrasing-based rewriting, detector-guided adversarial paraphrasing, and homoglyph substitutions (Liu et al., 2025; Huang et al., 2024; Zhou et al., 2024; Cheng et al., 2026; Creo and Pudasaini, 2025). Therefore, studying attacks against these detectors is urgent, as it exposes the vulnerabilities of existing detection systems and provides stress tests that can inform the design of more robust detectors (Sadasivan et al., 2025).

We categorize existing detector-evasion attacks based on how they exploit the detector as follows. The first class of methods does not use any feedback from the detector and operates at the word-level. Specifically, they reduce detector confidence through local character-level or token-level perturbations, such as adversarial substitutions or homoglyph-based modifications, and do not fully rewrite the text (Huang et al., 2024; Creo and Pudasaini, 2025). The second class of methods uses LLMs to paraphrase or rewrite the generated content. Specifically, detector feedback is used to construct supervised fine-tuning datasets (Krishna et al., 2023), as guided sampling score (Cheng et al., 2026; Sadasivan et al., 2025). The third class is reinforcement learning (RL)-based approaches, which directly fine-tune a paraphrasing LLM with detector feedback, where lower detector confidence or higher human-likeness scores are used as rewards to guide policy optimization (David and Gervais, 2025; Ranganath and Ramesh, 2026).

Despite their success in attacking the detector, a central trade-off has not yet been adequately addressed: a successful attack should not merely reduce the detector score, but also preserve the fine-grained semantics of the original input. As we will show in our experiments, existing approaches often improve detector evasion by sacrificing semantic

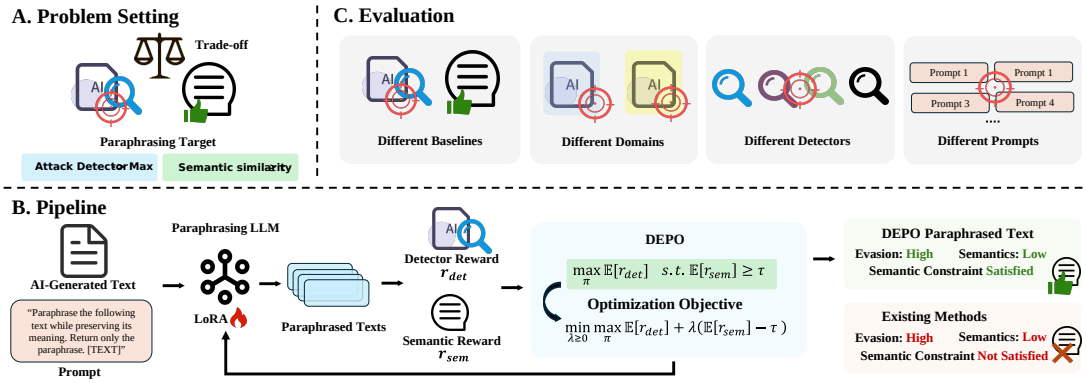


Figure 1: The DEPO framework for detector-evasive paraphrasing.(A) Problem setting: maximizing detector evasion $\mathbb{E}[r_{\text{det}}]$ under the semantic constraint $\mathbb{E}[r_{\text{sem}}] \geq \tau$. (B) Pipeline: DEPO uses detector and semantic rewards to optimize a paraphrasing LLM with constrained policy optimization. (C) Evaluation: DEPO also exhibits cross-domain, cross-detector, and prompt-level robustness.

fidelity, and even when semantic preservation is incorporated, they lack an explicit mechanism for precisely controlling semantic degradation.

1.1 Main Contributions

To address these limitations, we propose a constrained RL framework for detector-evasive LLM paraphrasing. Given AI-generated content as input, the paraphrasing LLM is optimized to generate a rewritten output that evades AI-text detectors while preserving the fine-grained semantics of the original text. We formulate this problem as a Constrained Markov Decision Process (CMDP), where detector evasion is optimized as the primary objective and semantic preserving requirements are imposed as explicit constraints (Altman, 1999; Achiam et al., 2017; Tessler et al., 2018; Li et al., 2024a). Under this formulation, the model improves attack success only within a feasible region defined by semantic preservation.

Building on this formulation, we introduce DEPO, short for Detector Evasion Policy Optimization, a Lagrangian primal-dual policy optimization algorithm that combines a group-based policy gradient with an in-group baseline, following GRPO (Shao et al., 2024), in the policy update to reduce computational costs and improve stability. During training, DEPO dynamically adjusts dual variables to balance detector reward maximization with constraint satisfaction, avoiding the ambiguity of optimizing a single composite reward. We note that prior RL-based detector-evasive methods use a scalarized reward design, where detector evasion and output semantic preservation are linearly combined with a manually chosen weight (David and Gervais, 2025; Ranganath and Ramesh, 2026), which, however, does not provide precise semantic preservation (as will be shown in our experiments).

We perform extensive experiments across diverse mainstream datasets, including MAGE (Li et al., 2024b), M4 (Wang et al., 2024), RAID (Dugan et al., 2024), and a peer-review dataset (Yu et al., 2025), and evaluations against MAGE detector (Li et al., 2024b), RoBERTa OpenAI (Solaiman et al., 2019), RADAR (Hu et al., 2023), Binoculars (Hans et al., 2024), and Fast-DetectGPT (Bao et al., 2024) detectors, and demonstrate DEPO's strong detector evasion with precise semantic preservation. We further demonstrate the cross-domain, cross-detector, and prompt-level generalization and robustness, showing our LLM paraphrasing generalizes to other detectors and domains, and is robust to variation in paraphrasing instructions. Importantly, we employ a more fine-grained BERTScore (Zhang et al., 2019) metric (token-level similarity) to measure semantic preservation and show its advantage over the T5 similarity (between sentence-level embeddings) (Ni et al., 2022; Ranganath and Ramesh, 2026).

1.2 Related Work

AI Text Detection Methods. Benchmark datasets for AI-text detection were developed, including open-domain question answering, news, stories, encyclopedic text, peer review, and online reviews (Guo et al., 2023; Li et al., 2024b; Yu et al., 2025; Gambetti and Han, 2023). Further, binary classification has become a representative detection paradigm: pretrained language encoders such as RoBERTa (Solaiman et al., 2019) are fine-tuned on human- and AI-written corpora (Zellers et al., 2019;

Ippolito et al., 2020; Uchendu et al., 2021; Li et al., 2024b). Recent works strengthened it through adversarial training, content-oriented modeling, and contrastive learning (Hu et al., 2023; Chen et al., 2025; Guo et al., 2024).

Another line of work develops zero-shot or training-free detection methods. Early studies exploit statistical signatures, including token-rank distributions and likelihood patterns (Gehrmann et al., 2019). DetectGPT uses probability curvature (Mitchell et al., 2023), while Fast-DetectGPT improves its efficiency through conditional probability curvature (Bao et al., 2024). Other methods, such as DNA-GPT, Ghostbuster, and Binoculars, exploit n-gram divergence, learned feature combinations, or cross-model perplexity discrepancies (Yang et al., 2024a; Verma et al., 2024; Hans et al., 2024). In addition, watermarking methods embed detectable statistical patterns during generation (Kirchenbauer et al., 2023). Despite their effectiveness, these detectors remain vulnerable when adversaries intentionally remove detectable signals.

Adversarial Attacks on AI Text Detectors. Recent work shows that AI text detectors are fragile under adversarial attacks. One line of work attacks detectors without using detector feedback, mainly through surface-level modifications. These methods reduce detector confidence via token-level or character-level perturbations, such as token substitutions and homoglyph replacements, while largely preserving the original text structure (Huang et al., 2024; Creo and Pudasaini, 2025).

Another line of work uses LLMs to rewrite AI-generated content with detector feedback. Sadasivan et al. (2025) shows that direct rewriting and recursive paraphrasing by LLMs can substantially weaken neural, zero-shot, and watermark-based detectors. Krishna et al. (2023) constructs paraphrase training data by using LLMs to generate candidate rewrite pairs and applying detectors to filter them, resulting in a supervised paraphraser that learns detector-evasive rewriting patterns. More adaptive methods incorporate detector signals directly into the attack process: Lu et al. (2023) uses detector feedback to construct prompts that induce detector-evasive generations; Cheng et al. (2026) uses detector scores to guide candidate selection and iterative adversarial rewriting; and Nicks et al. (2024); David and Gervais (2025); Ranganath and Ramesh (2026) optimize generation policies with reinforcement learning, where detector-derived scores are used as rewards to encourage evasive outputs.

These RL fine-tuning approaches are related, but they usually combine detector evasion and semantic preservation into a single scalar objective. In contrast, we formulate detector-evasive generation as a constrained optimization problem, enforcing semantic preservation explicitly rather than treating it as a soft reward regularization.

2 Problem Formulation

Given an AI-generated input text $x \in \mathcal{X}$, the paraphrasing LLM, represented by a policy $\pi_\theta(y \mid x)$ parameterized by θ, generates a rewritten text $y \in \mathcal{Y}$. The goal is to rewrite x such that y is less likely to be classified as AI-generated by a target detector, while preserving the semantics of the original text x. This setting naturally involves two possibly conflicting objectives: detector evasion and semantic preservation. Instead of linearly combining them into a single scalar reward, we treat semantic preservation as an explicit constraint.

Detector-evasive paraphrasing. Denote by $P_{\text{detector}}(y) \in [0, 1]$ the probability assigned by the detector that the rewritten text y is AI-generated. We define the detector-evasion reward as $r_{\text{det}}(y) = 1 - P_{\text{detector}}(y)$. A higher value of $r_{\text{det}}(y)$ indicates stronger detector evasion. The primary objective is to maximize the expected detector-evasion reward.

Semantic preservation. Detector evasion alone is insufficient for paraphrasing tasks, since a model may reduce detector confidence by significantly changing the semantics of the original text, deleting important details, or producing unrelated content. Prior detector-evasion and paraphrase-attack work often measures semantic preservation with T5-based sentence-level similarity (Ni et al., 2022; Ranganath and Ramesh, 2026). In contrast, we use BERTScore F1 (Zhang et al., 2019) as the semantic preservation metric, because its contextual token-level matching is better suited for capturing fine-grained semantic changes introduced during paraphrasing. We provide a sensitivity analysis and examples in Appendix B.1. Specifically, define

$$r_{\text{sem}}(x, y) = \text{BERTScore}_{F1}(x, y), \qquad (1)$$

where $r_{\text{sem}}(x, y) \in [0, 1]$ measures the token-level semantic similarity between x and y.

CMDP formulation. We formulate detector-evasive paraphrasing as a Constrained Markov Decision Process (CMDP). Let $\mathcal{D}$ denote the training dataset of AI-generated texts. The objective is to maximize detector evasion while satisfying the se-

mantic preservation constraint:

$$\max_{\theta} \; J_{\text{det}}(\theta) = \mathbb{E}_{x\sim\mathcal{D},\, y\sim\pi_\theta(\cdot|x)}\left[r_{\text{det}}(y)\right] \qquad (2)$$
$$- \beta\, \mathbb{D}_{\text{KL}}\left(\pi_\theta(\cdot \mid x) \,\|\, \pi_{\text{ref}}(\cdot \mid x)\right),$$
$$\text{s.t. } J_{\text{sem}}(\theta) = \mathbb{E}_{x\sim\mathcal{D},\, y\sim\pi_\theta(\cdot|x)}\left[r_{\text{sem}}(x,y)\right] \geq \tau_{\text{sem}}.$$

Here, π_{ref} is the reference policy and $\beta \geq 0$ controls the strength of the KL regularization, $\mathbb{D}_{\text{KL}}$ corresponds to the KL divergence. We require the expected semantic similarity $J_{\text{sem}}(\theta)$ to exceed a predefined threshold τ_{sem}. We will discuss the choice of τ_{sem} later in Appendix A.5.

Lagrangian dual. To solve eq. (2), we introduce a dual variable $\lambda \geq 0$ and define the Lagrangian:

$$\mathcal{L}(\theta, \lambda) = J_{\text{det}}(\theta) + \lambda\left(J_{\text{sem}}(\theta) - \tau_{\text{sem}}\right). \qquad (3)$$

The original problem in eq. (2) is equivalent to: $\max_\theta \min_{\lambda\geq 0} \mathcal{L}(\theta, \lambda)$. Its dual problem can be written as follows:

$$\min_{\lambda\geq 0} \max_{\theta} \mathcal{L}(\theta, \lambda). \qquad (4)$$

We note that such a transformation incurs zero duality gap and is equivalent to the original problem in eq. (2) (Altman, 1999). This min-max problem implies a dual-descent process: policy parameters θ are updated to maximize the Lagrangian, while the multiplier λ is updated to minimize it, thereby penalizing the policy if the constraint is violated.

3 Detector Evasion Policy Optimization

We introduce our algorithm DEPO, a constrained RL algorithm for detector-evasive paraphrasing. Unlike linear reward scalarization, DEPO adaptively adjusts the influence of the semantic constraint through a dual variable, allowing the policy to improve detector evasion only within the feasible semantic-preservation region.

To improve training efficiency and avoid the need for a separate value network, we introduce a group-based policy optimization procedure (Shao et al., 2024). For each input text $x \sim \mathcal{D}$, we sample a group of G candidate rewrites $\{y_i\}_{i=1}^{G}$ from the behavior policy $\pi_{\theta_{\text{old}}}(\cdot \mid x)$. Each candidate rewrite is evaluated by two reward signals: a detector-evasion reward from the AI-text detector and a semantic-preservation reward based on BERTScore F1 as in eq. (1).

Reward Estimation. For each sampled rewrite y_i, we compute the detector-evasion reward as $r_{\text{det}}^{(i)} = 1 - P_{\text{detector}}(y_i)$. To measure semantic preservation, we compute BERTScore F1 between the original input x and the rewritten output y_i: $r_{\text{sem}}^{(i)} = \text{BERTScore}_{F1}(x, y_i)$.

Group-Relative Advantage Estimation. For each x, DEPO normalizes the detector and semantic rewards within the sampled group. Let

$$\mu_{\text{det}} = \frac{1}{G}\sum_{i=1}^{G} r_{\text{det}}^{(i)}, \qquad (5)$$

$$\sigma_{\text{det}} = \sqrt{\frac{1}{G}\sum_{i=1}^{G}\left(r_{\text{det}}^{(i)} - \mu_{\text{det}}\right)^2}, \qquad (6)$$

and similarly define μ_{sem} and σ_{sem} for the semantic preservation rewards. The group-relative advantages are then computed as

$$A_{\text{det}}^{(i)} = \frac{r_{\text{det}}^{(i)} - \mu_{\text{det}}}{\sigma_{\text{det}}},\; A_{\text{sem}}^{(i)} = \frac{r_{\text{sem}}^{(i)} - \mu_{\text{sem}}}{\sigma_{\text{sem}}}. \qquad (7)$$

This in-group normalization provides a prompt-specific baseline and reduces the variance of policy-gradient estimation.

Primal-Dual Policy Optimization. At step t, DEPO maintains a Lagrange multiplier $\lambda_t \geq 0$ for the semantic preservation constraint. The combined advantage for the i-th rewrite is defined as $A_{\text{comb}}^{(i)} = A_{\text{det}}^{(i)} + \lambda_t A_{\text{sem}}^{(i)}$. The detector advantage drives the policy toward stronger detector evasion, while the semantic advantage is weighted by the current dual variable. When the semantic constraint is violated, λ_t increases and gives greater importance to semantic preservation. When the constraint is satisfied, λ_t decreases, allowing the policy to focus more on detector evasion.

Policy update. To update the policy parameters θ, DEPO maximizes a clipped policy-gradient objective with a KL penalty against the frozen reference policy π_{ref}. For each sampled rewrite y_i, define the importance sampling ratio $\rho_i(\theta) = \frac{\pi_\theta(y_i|x)}{\pi_{\theta_{\text{old}}}(y_i|x)}$. For a fixed input x, the policy optimization objective is

$$\max_{\theta} \mathcal{L}_{\text{policy}}(\theta; x) = \frac{1}{G}\sum_{i=1}^{G}\Big[\min\Big(\rho_i(\theta)A_{\text{comb}}^{(i)},$$
$$\text{clip}\left(\rho_i(\theta), 1-\epsilon, 1+\epsilon\right)A_{\text{comb}}^{(i)}\Big)$$
$$- \beta D_{\text{KL}}\big(\pi_\theta(y_i \mid x) \,\|\, \pi_{\text{ref}}(y_i \mid x)\big)\Big], \qquad (8)$$

where ϵ is the clipping coefficient.

Dual update. After the policy update, DEPO updates the Lagrange multiplier according to the current degree of semantic-constraint satisfaction:

$$\lambda_{t+1} = \max\left(0,\; \lambda_t - \eta_\lambda\left(\mu_{\text{sem}} - \tau_{\text{sem}}\right)\right),$$

where η_λ is the learning rate for the dual variable. If the average BERTScore falls below τ_{sem}, the term inside the parentheses becomes negative and λ_t increases, thereby assigning greater weight to semantic preservation in subsequent policy updates. If the semantic constraint is satisfied, λ_t decreases and reduces the influence of the constraint term. The complete algorithm is presented in Appendix A.1.

4 Experiments

4.1 Experimental Setup

Datasets and task. We use MAGE benchmark (Li et al., 2024b) as the primary training and in-domain evaluation corpus. To evaluate cross-domain transfer, we further test the trained paraphrasing policy on M4 (Wang et al., 2024), RAID (Dugan et al., 2024), and the ICLR peer-review subset from Yu et al. (2025). In all evaluations, AI-written texts are used as inputs, while human-written texts are used for threshold calibration and AUC computation. Additional details of the peer-review dataset are provided in Appendix A.3. In Section 4.5, we also train DEPO on a different RAID dataset using the RADAR detector for cross-reward validation.

Target detectors. During training, unless otherwise specified, DEPO uses the MAGE detector as the detector-reward model. At evaluation time, we test the attacked outputs against five detectors: MAGE detector (Li et al., 2024b), OpenAI RoBERTa (Solaiman et al., 2019), RADAR (Hu et al., 2023), Binoculars (Hans et al., 2024), and Fast-DetectGPT (Bao et al., 2024). Details of these detectors are in Appendix A.2. These detectors cover supervised neural classifiers, adversarially trained detectors, and zero-shot likelihood-based detectors. For the peer-review dataset, we additionally evaluate an in-domain RoBERTa detector trained on ICLR reviews.

Training framework. All trainable variants are initialized from the same base paraphrasing model and fine-tuned with parameter-efficient LoRA adapters (Hu et al., 2022). We implement RL with the TRL library (von Werra et al., 2020). Unless otherwise specified, the semantic threshold is set to $\tau_{\text{sem}} = 0.85$. Complete implementation details are provided in Table 14. Discussion on τ_{sem} can be found in Appendix A.5. We also conduct ablation studies with a different threshold $\tau_{\text{sem}} = 0.8, 0.85, 0.9$ (see Appendix B.2).

Evaluation metrics. We report detector reward, semantic reward, constraint fulfillment, attack success rate (ASR), ΔAUC, and output length. Detailed definitions of these metrics can be found in Appendix A.2. All detector scores are oriented so that larger scores indicate stronger confidence that the input is AI-generated.

4.2 Evasion vs. Semantics

Table 1 presents the results against the MAGE detector. The attack policy is trained with MAGE detector feedback. We compare DEPO with existing paraphrase attacks, single-objective ablations, and reward linearization baselines. Details of these baselines are in Appendix A.2.

Results. Table 1 shows a clear trade-off between detector evasion and semantic preservation. Existing paraphrase attacks either provide limited evasion or sacrifice semantic fidelity. For example, Direct Paraphrasing satisfies the semantic constraint with a BERTScore of 0.854, but only reaches an ASR of 0.030, while stronger baselines such as AuthorMist and Adversarial Paraphrase improve ASR but fall below the semantic preservation threshold.

Optimizing detector evasion alone achieves the strongest attack, with ASR 0.895 and AUC 0.472, but violates the semantic preservation constraint. In contrast, optimizing semantic preservation alone almost perfectly preserves meaning but produces little evasion. Fixed linear scalarization interpolates between these extremes, but its performance is highly sensitive to the weights and does not precisely preserve semantics.

Our DEPO achieves the best trade-off. Among methods satisfying the semantic preservation constraint of $\tau_{\text{sem}} = 0.85$, DEPO obtains the strongest evasion, reaching ASR 0.725 and ΔAUC 0.140 while maintaining a BERTScore of 0.857. These results match our CMDP formulation: optimize detector evasion while remaining within the prescribed semantic preservation.

4.3 Cross Detector Generalization

Table 2 evaluates the attacked outputs from the MAGE evaluation set under different AI-text detectors. This setting tests whether a policy trained with MAGE detector feedback can also compromise other detectors on the same dataset.

Results. Compared with the original texts, DEPO consistently increases ASR and lowers AUC across all five detectors, including supervised detectors and zero-shot likelihood-based detectors. The effect is strongest on the reward detector, where DEPO increases MAGE ASR from 0.01 to 0.73

Table 1: Main detector-evasion results against the MAGE detector on the evaluation set. The Original row reports detector performance on unmodified AI-written texts before attack. Detector reward, ASR, and ASR@1%FPR measure attack effectiveness; semantic reward and constraint fulfillment measure semantic preservation. The semantic threshold is $\tau_{\text{sem}} = 0.85$.

Method	Detector Reward ↑	Semantic Reward ↑	Constraint Fulfillment	ASR@$\tau = 0.5$ ↑	ASR@1%FPR ↑	ΔAUC ↓
Original	–	–	–	0.007	0.078	0.494
Direct Paraphrasing	0.040	0.854	✓	0.030	0.276	0.476
AuthorMist	0.070	0.810	✗	0.059	0.321	0.466
StealthRL	0.045	0.613	✗	0.021	0.789	0.459
SilverSpeak	0.028	0.338	✗	0.005	0.849	0.469
Adv. Paraphrase	0.073	0.826	✗	0.094	0.489	0.453
BERTScore 0 (Detector Evasion Only)	0.903	0.798	✗	0.895	0.995	0.028
BERTScore 1 (Semantic Only)	0.010	0.993	✓	0.008	0.081	0.494
BERTScore Linear 0.1	0.883	0.805	✗	0.913	0.994	0.050
BERTScore Linear 0.3	0.758	0.834	✗	0.768	0.953	0.081
BERTScore Linear 0.5	0.348	0.837	✗	0.347	0.668	0.344
DEPO	**0.716**	**0.857**	✓	**0.725**	**0.957**	**0.140**

Table 2: Same-domain cross-detector evaluation on the MAGE evaluation set. The attack policy is trained with MAGE detector feedback and evaluated against different supervised and zero-shot detectors.

Method	BERT Score	MAGE		RoBERTa		RADAR		Binoculars		Fast-DetectGPT	
		ASR↑	ΔAUC↓	ASR↑	ΔAUC↓	ASR↑	ΔAUC↓	ASR↑	ΔAUC↓	ASR↑	ΔAUC↓
Original	–	0.01	0.49	0.32	0.32	0.28	0.22	0.47	0.19	0.53	0.15
Direct Paraphrasing	0.854	0.03	0.48	0.47	0.23	0.26	0.25	0.74	0.09	0.74	0.10
AuthorMist	0.810	0.06	0.47	0.57	0.15	0.35	0.19	0.95	0.17	0.96	0.16
StealthRL	0.613	0.02	0.46	0.54	0.21	0.01	0.48	0.99	0.45	1.00	0.41
SilverSpeak	0.338	0.01	0.47	0.00	0.48	0.00	0.33	0.20	0.46	0.99	0.16
Adv. Paraphrase	0.826	0.09	0.45	0.56	0.20	0.29	0.26	0.82	0.04	0.81	0.06
MAGE BERTScore 0	0.798	0.91	0.05	0.63	0.10	0.44	0.14	0.98	0.23	0.97	0.20
MAGE BERTScore Linear 0.1	0.805	0.90	0.03	0.61	0.12	0.26	0.25	0.95	0.09	0.94	0.09
MAGE BERTScore Linear 0.3	0.834	0.77	0.08	0.61	0.11	0.45	0.13	0.95	0.18	0.95	0.14
MAGE BERTScore 1	0.993	0.01	0.49	0.31	0.32	0.28	0.23	0.47	0.20	0.53	0.15
MAGE BERTScore Linear 0.5	0.837	0.35	0.34	0.43	0.26	0.10	0.36	0.81	**0.07**	0.80	**0.03**
DEPO	0.857	**0.73**	**0.14**	**0.58**	**0.14**	**0.36**	**0.19**	**0.92**	0.12	**0.91**	0.09

while maintaining a BERTScore of 0.857. It also transfers to unseen detectors, achieving ASR scores of 0.58 on RoBERTa, 0.36 on RADAR, 0.92 on Binoculars, and 0.91 on Fast-DetectGPT. Some of the other baselines also generalize across different detectors, but a similar observation can be made: they fail to precisely control semantic preservation.

4.4 Cross Domain Generalization

Table 3 (and Tables 11 and 12 in the Appendix) evaluates whether the MAGE-trained policy transfers beyond the training domain. We keep the attack model fixed and test its outputs on three held-out corpora using the same detector suite:

- **M4** (Wang et al., 2024): a multi-generator, multi-domain, and multilingual benchmark for machine-generated text detection.
- **RAID** (Dugan et al., 2024): a robustness benchmark covering diverse generators, domains, decoding strategies, and adversarial attacks.
- **Peer review** (Yu et al., 2025): an academic peer-review dataset containing human and AI-written ICLR reviews from 2017–2022. Details are in Appendix A.3.

For each dataset, we use the evaluation protocol described in Section 4.1.

Peer-review RoBERTa detector. For the peer-review evaluation, we additionally train a RoBERTa-based detector on ICLR 2017–2019 re-

Table 3: Attack effectiveness on the M4 evaluation set under different AI-text detectors.

Method	BERT Score	MAGE		RoBERTa		RADAR		Binoculars		Fast-DetectGPT	
		ASR↑	ΔAUC↓	ASR↑	ΔAUC↓	ASR↑	ΔAUC↓	ASR↑	ΔAUC↓	ASR↑	ΔAUC↓
Original	–	0.02	0.47	0.84	0.03	0.36	0.14	0.93	0.08	0.74	0.01
Direct Paraphrasing	0.856	0.06	0.41	0.77	0.04	0.38	0.13	0.93	0.01	0.77	0.07
AuthorMist	0.817	0.04	0.42	0.82	0.00	0.48	0.05	0.98	0.26	0.97	0.23
StealthRL	0.656	0.01	0.34	0.77	0.15	0.02	0.44	1.00	0.43	0.99	0.41
SilverSpeak	0.391	0.03	0.35	0.00	0.48	0.00	0.30	0.66	0.45	0.99	0.28
Adv. Paraphrase	0.856	0.13	0.39	0.88	0.01	0.73	0.02	0.80	0.01	0.79	0.09
MAGE BERTScore 0	0.804	0.85	0.18	0.85	0.03	0.64	0.04	0.98	0.27	0.96	0.24
MAGE BERTScore Linear 0.1	0.791	0.90	0.21	0.86	0.06	0.39	0.13	0.97	0.15	0.95	0.15
MAGE BERTScore Linear 0.3	0.839	0.68	0.04	0.85	0.04	0.57	0.01	0.96	0.21	0.92	0.18
MAGE BERTScore 1	0.990	0.02	0.46	0.83	**0.00**	0.36	0.14	0.93	0.06	0.73	**0.03**
MAGE BERTScore Linear 0.5	0.827	0.29	0.23	0.73	0.09	0.18	0.29	**0.97**	**0.03**	0.87	0.04
DEPO	0.854	**0.70**	**0.04**	**0.83**	0.02	**0.51**	**0.04**	0.95	0.22	**0.93**	0.17

views from the AI Peer Review Detection Benchmark (Yu et al., 2025). The detector is evaluated on held-out ICLR 2020–2022 reviews, which tests temporal generalization within the same peer-review domain. As shown in Table 5 (in the Appendix), this detector achieves very high TPR under low-FPR operating points, outperforming the strongest reported baseline detector, Anchor (Yu et al., 2025), from the original benchmark.

Results. Table 3 (and Tables 11 and 12 in the Appendix) show that DEPO generalizes beyond the MAGE training domain. On M4, DEPO maintains strong attack performance against several detectors, reaching ASR 0.70 on MAGE, 0.51 on RADAR, 0.95 on Binoculars, and 0.93 on Fast-DetectGPT, while keeping a BERTScore of 0.854, satisfying the prespecified threshold. Because RoBERTa is already weak on the original M4 texts, DEPO adds little but remains competitive with the strongest semantically faithful baselines.

The transfer effect is clearer on RAID. Although RAID differs from MAGE in generators, domains, and attack settings, DEPO reduces detector separability across all five detectors, including the target-family MAGE detector and unseen detectors. It reaches ASR 0.86 on MAGE and 0.99 on RoBERTa, and also improves evasion against RADAR, Binoculars, and Fast-DetectGPT while maintaining a BERTScore of 0.878. Some unconstrained variants achieve stronger evasion, but at the cost of lower semantic similarity. DEPO instead provides a more balanced cross-detector profile under semantic control.

On peer-review data, DEPO remains effective for general-purpose detectors, with ASR 0.94 on MAGE, 0.84 on RADAR, 0.95 on Binoculars, and 0.78 on Fast-DetectGPT. The in-domain PeerRevIew RoBERTa detector shows lower AUC degradation, suggesting that a detector trained directly on peer-review data captures domain-specific signals that transfer attacks do not fully remove. Nevertheless, DEPO increases threshold-based evasion on this detector while preserving semantics more than most attack baselines.

Overall, DEPO learns rewriting patterns that transfer across domains and detector families.

4.5 Cross-Reward Validation

To test whether DEPO generalizes beyond a single training corpus and reward detector, we conduct a cross-reward validation. Specifically, we train a separate DEPO policy on RAID using RADAR as a detector. This setting changes both the training distribution and the reward detector.

Target-detector results. Table 4 shows that DEPO remains effective when trained on RAID with RADAR feedback. Compared with Direct Paraphrasing and Adversarial Paraphrase, DEPO achieves a much stronger attack against RADAR while still satisfying the semantic constraint. It reaches ASR 0.952 and ΔAUC 0.095 with a BERTScore of 0.850. Detector-only optimization obtains lower AUC, but its BERTScore drops to 0.775, indicating that its attack strength comes with clear semantic degradation. These results confirm that DEPO can optimize a different detector-evasion objective while keeping semantic preservation under explicit control.

Comparison with MAGE-trained DEPO. Ta-

Table 4: Main results of DEPO trained on RAID with RADAR detector feedback. The Original row reports RADAR detector performance on unmodified AI-written texts. The semantic threshold is $\tau_{\text{sem}} = 0.85$.

Method	Detector Reward ↑	Semantic Reward ↑	Constraint Fulfillment	ASR@$\tau = 0.5$ ↑	ASR@1%FPR ↑	ΔAUC ↓
Original	–	–	–	0.444	0.631	0.419
Direct Paraphrasing	0.652	0.877	✓	0.720	0.835	0.366
AuthorMist	0.789	0.825	✗	0.839	0.891	0.240
StealthRL	0.259	0.733	✗	0.256	0.337	0.460
SilverSpeak	0.327	0.374	✗	0.136	0.634	0.486
Adv. Paraphrase	0.697	0.876	✓	0.770	0.865	0.348
BERTScore 0 (Detector Evasion Only)	0.973	0.775	✗	0.983	0.986	0.287
BERTScore 1 (Semantic Only)	0.475	0.991	✓	0.458	0.653	0.413
BERTScore Linear 0.1	0.961	0.783	✗	0.971	0.980	0.274
DEPO	**0.931**	**0.850**	✓	**0.952**	**0.964**	**0.095**

ble 13 (in the Appendix) further shows that the reward detector strongly shapes the learned attack behavior. Compared with the MAGE-trained DEPO policy evaluated on RAID in Table 11, the RADAR-trained policy is much stronger against RADAR, increasing ASR from 0.65 to 0.95 and lowering ΔAUC from 0.32 to 0.09. In contrast, its attack against the MAGE detector becomes weaker, with ASR decreasing from 0.86 to 0.19. This contrast is expected: each policy is optimized against a different reward detector. The RADAR-trained policy also transfers to several unseen detectors, reaching ASR 0.98 on RoBERTa, 0.87 on Binoculars, and 0.84 on Fast-DetectGPT. Overall, the results show that DEPO is not tied to MAGE or to a single dataset, but the strongest attack direction is determined primarily by the detector used for reward feedback.

AUC-level comparisons between MAGE-trained and RADAR-trained policies are in Appendix B.4.

Together, these results provide a cross-validation of DEPO: the same constrained optimization framework remains effective when both the training dataset and reward detector are changed, while the detector used for reward feedback determines where the learned attack is strongest.

4.6 Summary

The results show that DEPO improves detector evasion while keeping semantic preservation under explicit control. Its gains are not confined to the reward detector or the training corpus, and the cross-reward experiment confirms that the framework remains effective when both the training data and detector feedback are changed. These findings support the central claim that constrained policy optimization provides a more reliable way to control the evasion–semantics trade-off than unconstrained optimization or fixed reward scalarization. Additionally, to further examine DEPO's generalization behavior, prompt stability, and output quality, we analyze detection score distribution change under attacking, reward detector effects on AUC, prompt robustness, and quality evaluation of paraphrased texts in Appendices B.3, B.4, B.5, and B.6. An interactive demo of the DEPO model is also available at `https://anonymous-submission-demo-paraphrase-demo.hf.space`, where readers can try paraphrasing.

5 Conclusion

We present DEPO, a constrained RL-based framework for optimizing AI-generated text to evade detection while preserving semantic meaning and generation quality. DEPO formulates detector evasion as a constrained RL problem, optimizing detector evasion while preserving semantics. This design enables DEPO to move beyond surface-level paraphrasing and learn adaptive rewriting policies that remain faithful to the original content. Experiments on various datasets show that DEPO achieves strong detector evasion within the prescribed semantic-preservation region. Cross-detector, cross-domain, and cross-reward evaluations further indicate that DEPO is not limited to a single detector or dataset, while the reward detector determines where the learned attack is strongest. We also note that BERTScore is not a complete semantic-equivalence measure, but it provides a more local, token-sensitive constraint than sentence-level embedding similarity for the perturbations considered here.

Limitations

DEPO has dual-use implications. We study detector-evasive paraphrasing to stress-test detector robustness and motivate stronger defenses, but similar techniques could be misused to conceal AI-generated text. Future work should combine constrained attack evaluation with human assessment, stronger semantic constraints, and adaptive defense mechanisms.

References

Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. 2017. Constrained policy optimization. In *International conference on machine learning*, pages 22–31. Pmlr.

Eitan Altman. 1999. *Constrained Markov Decision Processes*. Stochastic Modeling. Chapman & Hall/CRC, Boca Raton, FL.

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 *International Conference on Learning Representations*, volume 2024, pages 24814–24836.

Yihan Chen, Jiawei Chen, Guozhao Mo, Xuanang Chen, Ben He, Xianpei Han, and Le Sun. 2025. Coconuts: Concentrating on content while neglecting uninformative textual styles for ai-generated peer review detection. *arXiv preprint arXiv:2509.04460*.

Yize Cheng, Vinu Sankar Sadasivan, Mehrdad Saberi, Shoumik Saha, and Soheil Feizi. 2026. Adversarial paraphrasing: A universal attack for humanizing ai-generated text. *Advances in Neural Information Processing Systems*, 38:47591–47622.

Aldan Creo and Shushanta Pudasaini. 2025. Silverspeak: evading ai-generated text detectors using homoglyphs. In *Proceedings of the 1stWorkshop on GenAI Content Detection (GenAIDetect)*, pages 1–46.

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

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.

Alessandro Gambetti and Qiwei Han. 2023. Combat ai with ai: Counteract machine-generated fake restaurant reviews on social media. *arXiv preprint arXiv:2302.07731*.

Sebastian Gehrmann, Hendrik Strobelt, and Alexander M 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*.

Xun Guo, Shan Zhang, Yongxin He, Ting Zhang, Wanquan Feng, Haibin Huang, and Chongyang Ma. 2024. Detective: Detecting ai-generated text via multi-level contrastive learning. *Advances in Neural Information Processing Systems*, 37:88320–88347.

Hans WA Hanley and Zakir Durumeric. 2024. Machine-made media: Monitoring the mobilization of machine-generated articles on misinformation and mainstream news websites. In *Proceedings of the international AAAI conference on web and social media*, volume 18, pages 542–556.

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*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. *Iclr*, 1(2):3.

Xiaomeng Hu, Pin-Yu Chen, and Tsung-Yi Ho. 2023. Radar: Robust ai-text detection via adversarial learning. *Advances in neural information processing systems*, 36:15077–15095.

Guanhua Huang, Yuchen Zhang, Zhe Li, Yongjian You, Mingze Wang, and Zhouwang Yang. 2024. Are ai-generated text detectors robust to adversarial perturbations? In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6005–6024.

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*, pages 1808–1822.

Tanzila Kehkashan, Raja Adil Riaz, Ahmad Sami Al-Shamayleh, Adnan Akhunzada, Noman Ali, Muhammad Hamza, and Faheem Akbar. 2025. Ai-generated text detection: A comprehensive review of methods, datasets, and applications. *Computer Science Review*, 58:100793.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. 2023.

A watermark for large language models. In *International conference on machine learning*, pages 17061–17084. PMLR.

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. *Advances in neural information processing systems*, 36:27469–27500.

Tharindu Kumarage, Garima Agrawal, Paras Sheth, Raha Moraffah, Aman Chadha, Joshua Garland, and Huan Liu. 2024. A survey of ai-generated text forensic systems: Detection, attribution, and characterization. *arXiv preprint arXiv:2403.01152*.

Tianjiao Li, Ziwei Guan, Shaofeng Zou, Tengyu Xu, Yingbin Liang, and Guanghui Lan. 2024a. Faster algorithm and sharper analysis for constrained markov decision process. *Operations Research Letters*, 54:107107.

Yafu Li, Qintong Li, Leyang Cui, Wei Bi, Zhilin Wang, Longyue Wang, Linyi Yang, Shuming Shi, and Yue Zhang. 2024b. 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.

Xin Liu, Yang Li, and Kan Li. 2025. Enhancing the robustness of ai-generated text detectors: A survey. *Mathematics*, 13(13):2145.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: Nlg evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pages 2511–2522.

Ning Lu, Shengcai Liu, Rui He, Qi Wang, Yew-Soon Ong, and Ke Tang. 2023. Large language models can be guided to evade ai-generated text detection. *arXiv preprint arXiv:2305.10847*.

Yash Mahajan, Naman Bansal, Eduardo Blanco, and Santu Karmaker. 2024. Align-sim: A task-free test bed for evaluating and interpreting sentence embeddings through semantic similarity alignment. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 7393–7428.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D Manning, and Chelsea Finn. 2023. Detectgpt: Zero-shot machine-generated text detection using probability curvature. In *International conference on machine learning*, pages 24950–24962. PMLR.

Jianmo Ni, Gustavo Hernandez Abrego, Noah Constant, Ji Ma, Keith Hall, Daniel Cer, and Yinfei Yang. 2022. Sentence-t5: Scalable sentence encoders from pretrained text-to-text models. In *Findings of the association for computational linguistics: ACL 2022*, pages 1864–1874.

Charlotte Nicks, Eric Mitchell, Rafael Rafailov, Archit Sharma, Christopher Manning, Chelsea Finn, and Stefano Ermon. 2024. Language model detectors are easily optimized against. In *International Conference on Learning Representations*, volume 2024, pages 7807–7826.

Mike Perkins, Jasper Roe, Darius Postma, James McGaughran, and Don Hickerson. 2023. Game of tones: Faculty detection of gpt-4 generated content in university assessments. *arXiv preprint arXiv:2305.18081*.

Suraj Ranganath and Atharv Ramesh. 2026. Stealthrl: Reinforcement learning paraphrase attacks for multi-detector evasion of ai-text detectors. *arXiv preprint arXiv:2602.08934*.

Vinu Sankar Sadasivan, Aounon Kumar, Sriram Balasubramanian, Wenxiao Wang, and Soheil Feizi. 2025. Can ai-generated text be reliably detected? stress testing ai text detectors under various attacks. *Transactions on Machine Learning Research*.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, Gretchen Krueger, Jong Wook Kim, Sarah Kreps, and 1 others. 2019. Release strategies and the social impacts of language models. *arXiv preprint arXiv:1908.09203*.

Chen Tessler, Daniel J Mankowitz, and Shie Mannor. 2018. Reward constrained policy optimization. *arXiv preprint arXiv:1805.11074*.

Adaku Uchendu, Zeyu Ma, Thai Le, Rui Zhang, and Dongwon Lee. 2021. Turingbench: A benchmark environment for turing test in the age of neural text generation. In *Findings of the association for computational linguistics: EMNLP 2021*, pages 2001–2016.

Vivek Verma, Eve Fleisig, Nicholas Tomlin, and Dan Klein. 2024. Ghostbuster: Detecting text ghostwritten by large language models. 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 1702–1717.

Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, Shengyi Huang, Kashif Rasul, and Quentin Gallouédec. 2020. TRL: Transformers Reinforcement Learning.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Toru Sasaki, and 1 others. 2024. M4: Multi-generator, multi-domain, and multi-lingual black-box

machine-generated text detection. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1369–1407.

Junchao Wu, Shu Yang, Runzhe Zhan, Yulin Yuan, Lidia Sam Chao, and Derek Fai Wong. 2025. A survey on llm-generated text detection: Necessity, methods, and future directions. *Computational Linguistics*, 51(1):275–338.

Xianjun Yang, Wei Cheng, Yue Wu, Linda Petzold, William Wang, and Haifeng Chen. 2024a. Dna-gpt: Divergent n-gram analysis for training-free detection of gpt-generated text. In *International Conference on Learning Representations*, volume 2024, pages 48572–48597.

Xianjun Yang, Liangming Pan, Xuandong Zhao, Haifeng Chen, Linda Ruth Petzold, William Yang Wang, and Wei Cheng. 2024b. A survey on detection of llms-generated content. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 9786–9805.

Sungduk Yu, Man Luo, Avinash Madusu, Vasudev Lal, and Phillip Howard. 2025. Is your paper being reviewed by an llm? a new benchmark dataset and approach for detecting ai text in peer review. *arXiv e-prints*, pages arXiv–2502.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. Defending against neural fake news. *Advances in neural information processing systems*, 32.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, and 1 others. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623.

Ying Zhou, Ben He, and Le Sun. 2024. Humanizing machine-generated content: evading ai-text detection through adversarial attack. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 8427–8437.

A Experimental Details

A.1 Pseudocode of DEPO

In Algorithm 1, we present the complete pseudocode of DEPO.

Algorithm 1 Detector Evasion Policy Optimization

Require: Dataset $\mathcal{D}$; MAGE detector P_{detector}; BERTScore function; group size G; semantic threshold τ_{sem}; PPO clip coefficient ϵ; KL coefficient β; step sizes $\eta_\theta, \eta_\lambda$; reference policy π_{ref}

Initialize policy parameters θ
Initialize dual variable $\lambda_0 \geq 0$
Set behavior policy parameters $\theta_{\text{old}} \leftarrow \theta$
for $t = 0, 1, 2, \ldots$ **do**
  Sample input text $x \sim \mathcal{D}$
  Sample G rewrites $\{y_i\}_{i=1}^{G} \sim \pi_{\theta_{\text{old}}}(\cdot \mid x)$
  for $i = 1$ to G **do**
    Compute detector and semantic rewards $r_{\text{det}}^{(i)}, r_{\text{sem}}^{(i)}$
  end for
  Compute statistics $\mu_{\text{det}}, \sigma_{\text{det}}, \mu_{\text{sem}}, \sigma_{\text{sem}}$
  for $i = 1$ to G **do**
    Compute detector advantage $A_{\text{det}}^{(i)}$
    Compute semantic advantage $A_{\text{sem}}^{(i)}$
    Compute combined advantage:

$$A_{\text{DEPO}}^{(i)} = A_{\text{det}}^{(i)} + \lambda_t A_{\text{sem}}^{(i)}$$

  end for
  Update policy parameters

$$\theta \leftarrow \theta + \eta_\theta \nabla_\theta \mathcal{L}_{\text{policy}}(\theta)$$

  Update dual variable

$$\lambda_{t+1} = \max(0, \lambda_t - \eta_\lambda \left(\mu_{sem} - \tau_{\text{sem}}\right))$$

  Set $\theta_{\text{old}} \leftarrow \theta$
end for
return θ

A.2 Baselines, Detectors, Evaluation Metrics and Evaluation Dataset

Baselines. Below, we briefly introduce the baselines used in our experiments.

- *Direct Paraphrasing* (Krishna et al., 2023): prompt-based paraphrasing baseline that rewrites the input without detector-specific optimization.
- *AuthorMist* (David and Gervais, 2025): RL-based paraphraser trained with detector feedback to reduce AI-text detectability.
- *StealthRL* (Ranganath and Ramesh, 2026): RL-based paraphraser optimized for detector evasion.
- *SilverSpeak* (Creo and Pudasaini, 2025): homoglyph-based attack that perturbs surface characters to confuse AI-text detectors. We use a homoglyph substitution rate of 0.3.
- *Adversarial Paraphrase* (Cheng et al., 2026): a training-free adversarial paraphrasing attack that uses detector scores to guide candidate selection. We implement it with Qwen3-4B-Instruct-2507, sampling 4 candidate paraphrases per input with temperature 0.9 and top-$p = 0.95$.
- *Single-objective ablations*: BERTSCORE 0 optimizes detector evasion only, while BERTSCORE 1 optimizes semantic preservation only.
- *Linear scalarization*: BERTSCORE LINEAR combines detector and semantic rewards with fixed weights, testing whether a static reward mixture can recover the constrained trade-off achieved by DEPO.

Detectors. We then introduce the detectors used in our experiments.

- *MAGE* (Li et al., 2024b): supervised detector used as the training reward model.
- *RoBERTa* (Solaiman et al., 2019): OpenAI-RoBERTa-Large detector: supervised classifier fine-tuned to distinguish GPT-2 from human.
- *RADAR* (Hu et al., 2023): adversarially trained detector designed to improve robustness against paraphrasing attacks.
- *Binoculars* (Hans et al., 2024): zero-shot likelihood-ratio detector that compares the behavior of two language models.
- *Fast-DetectGPT* (Bao et al., 2024): zero-shot detector based on conditional probability curvature.

Evaluation metrics. We next introduce the details of evaluation metrics used in our experiments.

- *Detector reward*: $1 - P_{\text{detector}}(y)$.
- *Semantic reward*: BRTScore-F1 between the original text x and the rewritten text y, computed with `microsoft/deberta-xlarge-mnli` as the scoring backbone.
- *Constraint fulfillment*: whether the empirical average semantic reward satisfies the target constraint, i.e., $\mathbb{E}[\text{BERTScore}(x, y)] \geq \tau_{\text{sem}}$.
- *ASR@τ / ASR*: the fraction of attacked AI texts that are classified as human under the decision threshold τ. Since attacked AI texts are treated as the positive AI class, ASR is the complement

of the detector's true positive rate: $\mathrm{ASR}@\tau = 1 - \mathrm{TPR}@\tau$. For supervised neural detectors, including MAGE, RoBERTa, and RADAR, we use the standard probability threshold $\tau = 0.5$. For zero-shot detectors, whose raw scores are not calibrated probabilities, we select τ using 500 AI texts and 500 human texts from the corresponding dataset, choosing the threshold that maximizes Youden's J statistic.

- *Attack success rate (ASR)@1%FPR*: the fraction of attacked AI texts classified as human-written at a stricter operating point. For each detector, the threshold is computed from 10,000 human-written samples so that the empirical false positive rate is 1%, and is fixed when evaluating attacked AI-written texts.
- Δ*AUC*: the absolute deviation of the area under the receiver operating characteristic curve from random guessing: $\Delta\mathrm{AUC} = |\mathrm{AUC} - 0.5|$. Here, AUC is the area under the receiver operating characteristic curve. A smaller ΔAUC indicates that the detector is closer to random guessing and thus has weaker separability between human-written and attacked AI-written texts. We use ΔAUC rather than AUC because an AUC below 0.5 still indicates separability with a reversed decision direction, rather than complete detector failure.

Evaluation dataset. All reported results in our experiments are obtained on evaluation sets that are either officially provided or separately sampled from the available data splits. Specifically, for the Mage Dataset, we use the official validation split as the evaluation set. For the M4 dataset, we adopt the *dev_monolingual* split setting in Subtask A for evaluation. Since the RAID dataset does not provide a labeled evaluation split, we construct the evaluation set by randomly sampling instances from the training split using a random seed different from that used during training. For the Peer Review dataset, we use reviews collected from the ICLR conferences between 2020–2022 as the evaluation set. Unless otherwise specified, each evaluation set consists of 1,600 randomly sampled AI-generated samples. For the Peer Review dataset, due to the substantially longer average review length, we instead randomly sample 640 AI-generated samples for evaluation.

All baselines, detectors, and evaluation datasets used in this work are publicly available, and we use them in accordance with their respective licenses, terms of use, and intended research purposes.

A.3 Peer-review Dataset

The peer-review evaluation uses the AI Peer Review Detection Benchmark introduced by Yu et al. (2025). The full benchmark contains paired human- and AI-written peer reviews for research papers submitted to major AI conferences. In this work, we use the ICLR portion from 2017–2022, which provides human-written OpenReview reviews together with AI-generated reviews produced for the same papers. This construction is useful for detector-evasion evaluation because the human and AI reviews are grounded in the same manuscript context, reducing topic mismatch and making the detection problem closer to realistic peer-review misuse scenarios.

The dataset is also practically important. Peer review is a high-stakes academic workflow in which reviewers are expected to provide independent expert judgment. If LLM-written reviews become difficult to detect, program chairs and authors may have limited ability to identify low-effort or undisclosed AI-assisted reviewing. Evaluating detector-evasive paraphrasing on this corpus, therefore, tests our method in a long-form, domain-specialized setting where preserving substantive content matters: a successful rewrite should not only evade the detector but also retain the technical claims, criticisms, and recommendations present in the original review.

A.4 Peer-review RoBERTa Detector Validation

For the peer-review evaluation, we additionally train a RoBERTa-based detector on ICLR 2017–2019 reviews from the AI Peer Review Detection Benchmark (Yu et al., 2025). The detector is evaluated on held-out ICLR 2020–2022 reviews, which tests temporal generalization within the peer-review domain. As shown in Table 5, this detector achieves strong low-FPR performance and outperforms the strongest reported baseline detector, Anchor (Yu et al., 2025), from the original benchmark. We therefore use it as a strong in-domain detector for evaluating detector-evasive paraphrasing on peer-review text.

A.5 Choice of τ_{sem}

The threshold τ_{sem} specifies the semantic-preservation region within which detector evasion is optimized. In practice, it can be selected by calibrating BERTScore on a held-out set of ordi-

Table 5: Low-FPR performance of our Peer-review RoBERTa detector. The detector is trained on ICLR 2017–2019 reviews and evaluated on held-out ICLR 2020–2022 reviews generated by GPT-4o, Gemini, and Claude. Anchor and Binoculars results are taken from Yu et al. (2025). FPR and TPR are reported as percentages.

		Target FPR: 0.1%		Target FPR: 0.5%		Target FPR: 1%	
Evaluation	**Method**	**FPR**	**TPR**	**FPR**	**TPR**	**FPR**	**TPR**
GPT-4o Reviews	Anchor	0.1	63.5	0.5	83.7	1.0	88.8
	Binoculars	0.2	17.1	0.6	33.6	1.0	45.2
	Peer-review RoBERTa	0.1	**93.2**	0.4	**99.3**	0.8	**99.9**
Gemini Reviews	Anchor	0.2	59.7	0.8	80.3	1.3	86.5
	Binoculars	0.2	61.5	0.6	78.0	1.0	85.5
	Peer-review RoBERTa	0.1	**95.8**	0.4	**99.8**	0.8	**99.9**
Claude Reviews	Anchor	0.1	59.6	0.5	75.8	1.0	81.8
	Binoculars	0.2	43.5	0.6	65.8	1.0	77.0
	Peer-review RoBERTa	0.1	**98.9**	0.4	**99.8**	0.8	**99.8**

nary paraphrases and choosing a lower quantile of their similarity scores, so that standard meaning-preserving rewrites are treated as feasible while rewrites with substantial semantic drift are rejected. Following this principle, we set $\tau_{\text{sem}} = 0.85$ in our main experiments. This value places direct paraphrasing in the feasible region but rules out detector-only and low-semantic-weight optimization variants that achieve stronger evasion by sacrificing meaning. We use the same threshold across all main evaluations and explicitly report constraint fulfillment, so that attack effectiveness is interpreted solely in terms of whether the semantic constraint is satisfied.

B Additional Experiments

In this section, we include additional experiment results and ablation studies that are not discussed in the main paper.

B.1 Fine-grained Semantic Sensitivity Analysis

Motivation and Scope. Semantic preservation is central to detector-evasive paraphrasing. A useful semantic metric should assign high scores to meaning-preserving rewrites, but should also decrease when a small local edit changes the meaning of the sentence. This is important because detector-evasive rewriting may preserve most surface words while altering critical semantic components, such as negation, role structure, causal relations, temporal conditions, or logical constraints.

We state that his analysis is not intended to provide a comprehensive comparison of which metric gives a more general, complete, or universally faithful semantic representation. T5 sentence-level similarity and BERTScore come from different modeling and evaluation paradigms: Sentence-T5 produces fixed-size sentence embeddings compared by cosine similarity, whereas BERTScore performs contextual token-level matching and aggregates token-wise similarities. We therefore treat the following experiment as a task-specific diagnostic: under controlled local semantic perturbations that are especially relevant to detector-evasive paraphrasing, we test whether BERTScore provides a more responsive constraint signal than T5 sentence-level cosine similarity.

Metrics compared. We compare sentence-level T5 similarity and BERTScore. For T5 similarity, we use Sentence-T5 embeddings and compute cosine similarity between the source sentence and the candidate rewrite. This provides a global sentence-level similarity score, but may smooth over local meaning changes when the two sentences share most of their lexical content. In contrast, BERTScore computes contextual token-level matching between a candidate and a reference, and aggregates token-wise similarities into precision, recall, and F1 (Zhang et al., 2019). Prior work shows that Sentence-T5 provides strong general-purpose sentence representations (Ni et al., 2022), while sentence-level embedding spaces may still be imperfect for fine-grained semantic distinctions such as negation, antonymy, and structural changes (Mahajan et al., 2024). Related work has also used BERT-family sentence-similarity metrics to evaluate semantic preservation in adversarial paraphrasing; Adversarial attack (Cheng et al., 2026) uses cosine similarity between SBERT em-

beddings to measure semantic/text-quality preservation before and after paraphrasing. Our goal is therefore not to claim that BERTScore is universally superior, but to test whether it is more responsive to the types of local semantic changes that matter in our constrained paraphrasing setting.

Experimental setup. We construct a controlled set of sentence-level perturbation pairs. Starting from manually designed hard cases and automatically generated templates, we first obtain a meaning-preserving rewrite y^+ for each source sentence x using the paraphrase model with protected-span constraints. We then construct a corrupted rewrite y^- by applying a single rule-based semantic perturbation to y^+. Invalid cases where the rule cannot be applied, does not change the sentence, or produces an ill-formed output are removed. The resulting perturbations are grouped into four categories that focus on structural and logic-sensitive semantic changes: context shifts, logical constraints, logical relations, and role structures.

Given a semantic metric m, we compute the score drop caused by the perturbation for each pair as:

$$\Delta_m^{(i)} = s_m(x_i, y_i^+) - s_m(x_i, y_i^-), \tag{9}$$

where $s_m(\cdot,\cdot)$ denotes the similarity score produced by metric m. To make drops easier to interpret within each metric, we also compute the relative score drop:

$$\mathrm{RelDrop}_m^{(i)} = 100 \times \frac{s_m(x_i, y_i^+) - s_m(x_i, y_i^-)}{|s_m(x_i, y_i^+)| + \epsilon}, \tag{10}$$

where ϵ is a small constant for numerical stability. We report group-level averages of $\Delta_m^{(i)}$ and $\mathrm{RelDrop}_m^{(i)}$. Because T5 cosine similarity and BERTScore-F1 have different score scales, these drops should be interpreted as diagnostic indicators of within-metric sensitivity rather than calibrated cross-metric scores.

Results and analysis. Table 6 shows that BERTScore is more responsive than T5 sentence-level similarity to controlled structural and logic-sensitive perturbations in this diagnostic setting. Across 143 constructed pairs, BERTScore has a larger average relative drop than T5 similarity (8.94% vs. 3.90%).

The difference is especially clear for structural changes. For role-structure perturbations, where the agent and patient of an action are swapped while most lexical content remains unchanged, T5 similarity drops by only 0.50%, while BERTScore drops by 6.11%. BERTScore also shows larger drops for context shifts, logical constraints, and logical relations.

These results suggest that sentence-level embeddings can remain highly similar when most surface words are preserved, even if a local structural or logical edit changes the meaning. In contrast, BERTScore provides a stronger diagnostic signal for the semantic changes emphasized in this work. This does not imply that BERTScore is a universally better semantic representation than T5-based sentence embeddings. Rather, it supports our use of BERTScore as a practical semantic-preservation constraint in DEPO, where the goal is to penalize fine-grained semantic drift under constrained detector-evasive paraphrasing. Representative minimal-edit examples in Table 7 show the same trend qualitatively.

B.2 Ablation Study on Different τ_{sem}

To study the effect of the semantic constraint, we vary the semantic threshold τ_{sem} and train DEPO under three different constraint levels. We follow the training framework described in Section 4.1 and evaluate the resulting models on a randomly selected subset of 640 samples from the MAGE evaluation dataset. Table 8 reports the detector reward and semantic reward obtained under different values of τ_{sem}.

The results reveal a clear trade-off between detector evasion and semantic preservation. A smaller τ_{sem} relaxes the semantic constraint and allows more aggressive rewriting, which can improve detector evasion but may weaken semantic preservation. In contrast, a larger τ_{sem} imposes a stricter semantic-preservation requirement, encouraging the model to produce rewrites that remain closer in meaning to the source text.

Importantly, our algorithm DEPO is able to adapt to different semantic thresholds and converge under all tested constraint levels. This suggests that the proposed optimization framework can effectively control the balance between detector evasion and semantic preservation by adjusting τ_{sem}.

B.3 Detection Score Distribution Analysis

Beyond aggregate ASR and AUC metrics, we also inspect the detector-score distributions induced by different attack methods. This analysis provides a more detailed view of how paraphrasing shifts detector confidence on each evaluation corpus. A successful attack should move AI-written texts toward

Table 6: Diagnostic sensitivity comparison between T5 sentence-level similarity and BERTScore under controlled structural and logic-sensitive semantic perturbations. Δ denotes the average absolute score drop from the meaning-preserving rewrite y^+ to the corrupted rewrite y^-. RelDrop denotes the average relative score drop within each metric. Because the two metrics have different score scales, the drop values should be interpreted as diagnostic indicators rather than calibrated cross-metric comparisons.

Perturbation group	n	**ΔT5**	**ΔBERTScore**	**T5 RelDrop**	**BERTScore RelDrop**
Context shift	26	0.0197	0.0514	1.97%	5.14%
Logic constraint	51	0.0591	0.1170	5.91%	11.70%
Logical relation	51	0.0387	0.0899	3.87%	8.96%
Role structure	15	0.0050	0.0605	0.50%	6.11%
Overall	143	0.0390	0.0895	3.90%	8.94%

Table 7: Representative minimal-edit examples illustrating how T5 sentence-level similarity and BERTScore respond to local semantic perturbations. Each example preserves most lexical content while changing a key semantic relation, such as agent–patient roles or condition-specific effectiveness.

No.	**Reference sentence**	**Perturbed sentence**	**T5 change**	**BERTScore change**
1	Alicia gave Marcus the camera before the concert.	Marcus gave Alicia the camera before the concert.	$1.0000 \rightarrow 0.9968$ (-0.32%)	$1.0000 \rightarrow 0.9173$ (-8.27%)
2	Nora beat Liam by three points in the final round.	Liam beat Nora by three points in the final round.	$1.0000 \rightarrow 0.9968$ (-0.32%)	$1.0000 \rightarrow 0.9199$ (-8.01%)
3	The landlord gave the tenant the contract after the inspection.	The tenant gave the landlord the contract after the inspection.	$1.0000 \rightarrow 0.9909$ (-0.91%)	$1.0000 \rightarrow 0.9189$ (-8.11%)
4	The editor gave the writer the draft after the review.	The writer gave the editor the draft after the review.	$1.0000 \rightarrow 0.9974$ (-0.26%)	$1.0000 \rightarrow 0.9315$ (-6.85%)
5	The vaccine was effective in adults but ineffective in children.	The vaccine was effective in children but ineffective in adults.	$1.0000 \rightarrow 0.9944$ (-0.56%)	$1.0000 \rightarrow 0.9343$ (-6.57%)

Table 8: Ablation study on different semantic constraint thresholds τ_{sem}.

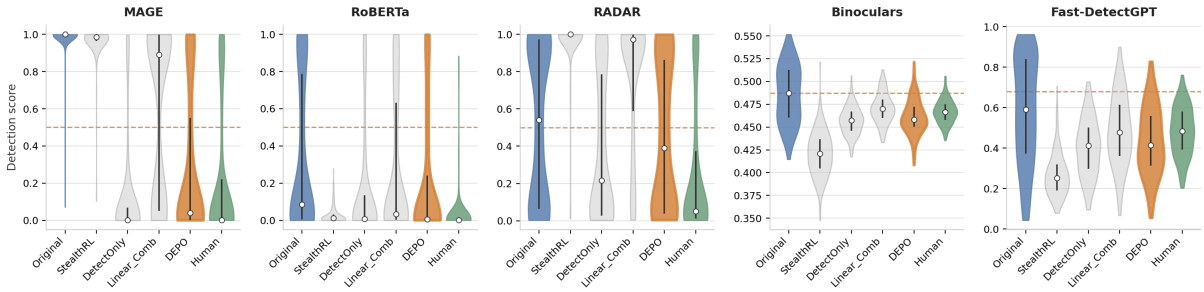
Target τ_{sem}	**Detector reward**	**Semantic reward**
0.80	0.867	0.811
0.85	0.707	0.858
0.90	0.019	0.909

the human-written score region, while a semantically controlled attack should avoid achieving this shift through excessive deletion or meaning drift.

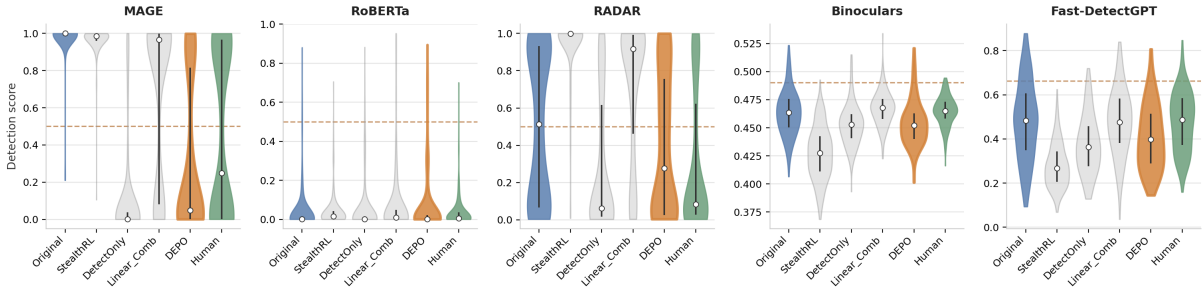
Figure 2 shows the score distributions across MAGE, M4, RAID, and peer-review evaluation sets. The distributions are consistent with the aggregate results in the main text: DEPO shifts detector scores toward lower AI-confidence regions across multiple datasets, while maintaining a more controlled semantic profile than aggressive unconstrained variants. These results complement the main tables by showing that DEPO's gains are not driven only by a small number of threshold-crossing examples, but reflect a broader change in detector-score behavior.

B.4 Reward Detector Effects on AUC

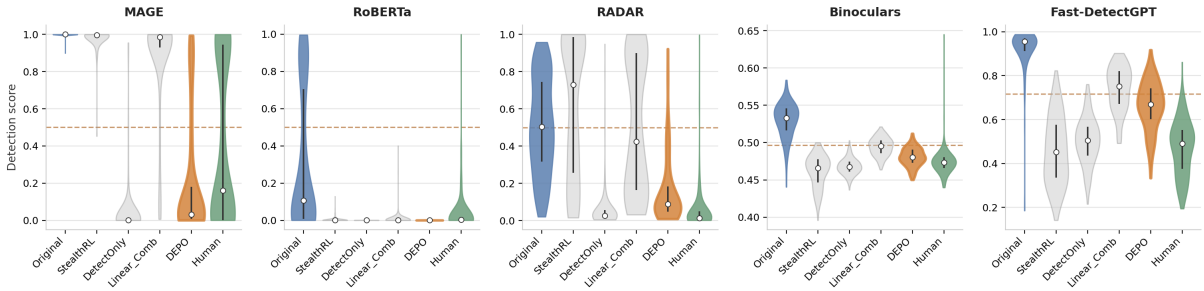
We further analyze the relative influence of the reward detector and the training dataset on DEPO's generalization behavior. For both the MAGE and RAID datasets, we randomly sample 640 AI-generated texts as rewriting targets and use the same number of human-generated texts to compute ROC curves and AUC. Figure 3 compares ROC curves for policies trained with different detector feedback and evaluated on MAGE and RAID. The main trend is that a policy trained with feedback from a given detector tends to produce stronger confusion for that detector. MAGE-trained DEPO more strongly reduces AUC for the MAGE detector on these two test sets, while RADAR-trained DEPO more strongly reduces AUC for the RADAR detector. This pattern is observed across datasets, suggesting that the reward detector has a larger effect on the learned attack direction than the training dataset.



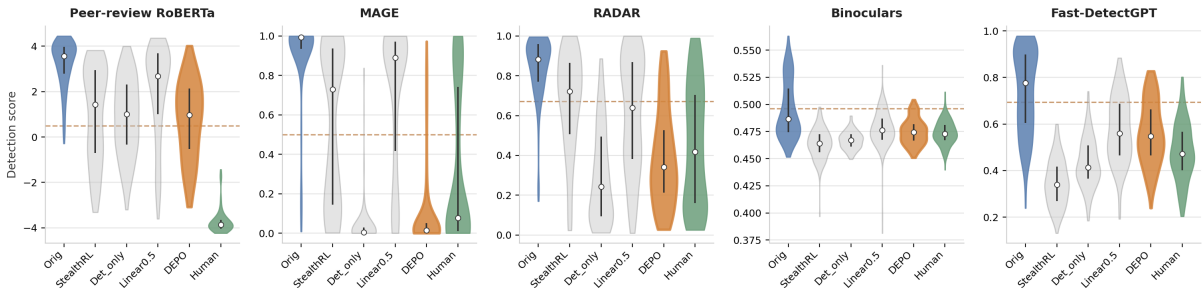
(a) MAGE



(b) M4

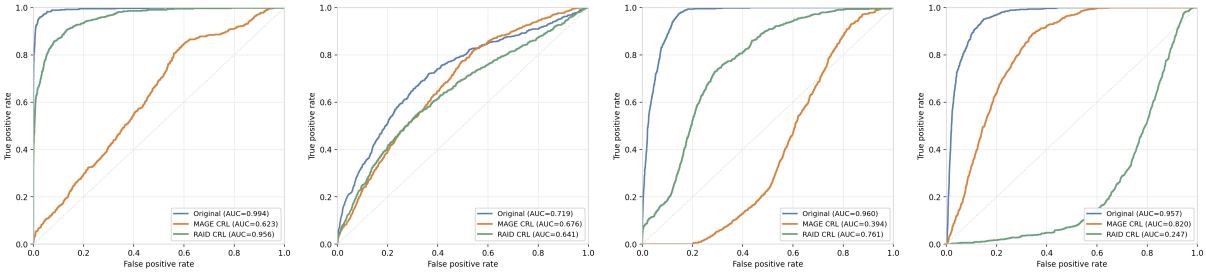


(c) RAID



(d) Peer Review

Figure 2: Detection score distributions across attack methods on four evaluation corpora. This distribution-level analysis complements the aggregate ASR and AUC results in the main text. `DetectOnly` denotes MAGE BERTScore 0 model, while `Linear_Comb` denotes BERTScore Linear 0.5 model.



(a) MAGE, MAGE detector (b) MAGE, RADAR detector (c) RAID, MAGE detector (d) RAID, RADAR detector

Figure 3: ROC curves for original texts and DEPO-attacked texts under different reward-detector settings. The policy trained with a given detector reward tends to reduce AUC more strongly on that detector, indicating that detector feedback has a stronger influence on the attack direction than dataset choice alone. In the figure, MAGE CRL denotes the MAGE-trained DEPO policy, while RAID CRL denotes the RADAR-trained DEPO policy.

Table 9: Prompt robustness of DEPO on a 640-sample subset of the MAGE evaluation set under different paraphrasing instructions.

Prompt Template	BERT Score	MAGE		RoBERTa		RADAR		Binoculars		Fast-DetectGPT	
		ASR↑	ΔAUC↓	ASR↑	ΔAUC↓	ASR↑	ΔAUC↓	ASR↑	ΔAUC↓	ASR↑	ΔAUC↓
Default	0.857	0.74	0.14	0.84	0.13	0.57	0.18	0.90	0.11	0.90	0.08
No Preserving Meaning	0.841	0.76	0.12	0.85	0.12	0.62	0.16	0.93	0.14	0.91	0.08
AuthorMist	0.850	0.76	0.13	0.85	0.13	0.59	0.17	0.93	0.12	0.92	0.08
Reworded	0.855	0.73	0.13	0.85	0.14	0.58	0.18	0.91	0.11	0.92	0.08
Concise	0.852	0.75	0.13	0.85	0.13	0.58	0.18	0.92	0.11	0.91	0.08

B.5 Prompt Robustness Evaluation

We further evaluate whether DEPO relies on a specific paraphrasing instruction. Unlike the main evaluation, which uses 1,600 AI-written texts, this prompt-robustness study is conducted on a 640-sample subset of AI-written texts from the MAGE evaluation set. All prompt templates are evaluated on the same subset using the same detector thresholds as in the main evaluation. Therefore, the absolute numbers in this section are not intended to exactly match the main MAGE results; instead, this analysis focuses on the relative stability of DEPO across different prompt wordings.

Table 9 shows that DEPO is stable under prompt rewording on the 640-sample evaluation subset. Across five templates, MAGE ASR remains between 0.73 and 0.76, while BERTScore stays between 0.841 and 0.857. Attacks generated with different prompts are also consistently effective and stable against the other evaluated detectors. Removing the explicit meaning-preservation instruction slightly lowers semantic similarity, but the attack metrics remain close to those of the default prompt. These results suggest that DEPO's behavior is not tied to a single paraphrasing instruction.

- **Default**: "Paraphrase the following text while preserving its meaning. Return only the paraphrase."
- **No preserving meaning**: "Paraphrase the following text to sound more human-like. Return only the paraphrase."
- **AuthorMist**: "Please paraphrase the following text to make it more human-like while preserving the original meaning. Return only the paraphrased text."
- **Reworded**: "Rewrite the passage below in your own words, keeping the original meaning intact. Output only the rewritten passage."
- **Concise**: "Paraphrase the text below. Output only the paraphrase."

B.6 Quality Evaluation of Paraphrased Texts

Recent work has shown that large language models can serve as scalable automatic judges for open-ended text evaluation, with reasonable agreement with human preferences and flexible rubric-based assessment (Zheng et al., 2023; Liu et al., 2023). We therefore conduct an LLM-based quality evaluation as a complementary analysis of paraphrase quality. Following the auto-rated quality evaluation protocol of Cheng et al. (2026), we ask an LLM judge to assign a single 1–5 equivalence rating to

each source–paraphrase pair. The rating measures whether the paraphrase preserves the meaning of the original text: a score of 5 indicates that the texts are approximately equivalent, while a score of 1 indicates that they are not topically related.

We use `gpt-oss-120b` as the judge model. For a fair comparison, all methods are evaluated on the shared subset of 300 AI-written samples selected randomly on MAGE evaluation dataset. Each example is scored independently, and the judge is shown only the source text and the paraphrase without method labels. In addition to the LLM rating, we report BERTScore against the source text and perplexity (PPL) as complementary automatic indicators of semantic overlap and language-model likelihood, respectively. PPL is computed using `Qwen/Qwen3-4B-Instruct-2507`.

Method	Rating	BERTScore	PPL
Original	–	–	20.34
Direct Paraphrasing	4.96	0.85	16.25
AuthorMist	4.64	0.81	26.31
StealthRL	2.94	0.61	61.60
SilverSpeak	4.62	0.34	35.12
Adv. Paraphrase	4.94	0.85	16.44
MAGE BERTScore 0	4.89	0.80	20.86
MAGE BERTScore 1	4.99	0.99	19.54
MAGE BERTScore Linear 0.1	4.86	0.80	23.72
MAGE BERTScore Linear 0.3	4.93	0.83	20.69
MAGE BERTScore Linear 0.5	4.90	0.84	17.61
DEPO	4.94	0.86	20.60

Table 10: Quality evaluation of paraphrased texts. Ratings are LLM-based equivalence scores on a 1–5 scale.

Results. Table 10 summarizes the LLM-judge evaluation results for DEPO and representative attack baselines. DEPO achieves a high LLM-based rating under the `gpt-oss-120b` judge. This indicates that, even when evaluated by a larger and stronger external quality rater, DEPO preserves the semantic equivalence and overall quality of the rewritten outputs. The result suggests that the proposed optimization does not obtain detector evasion by sacrificing paraphrase fidelity.

DEPO also yields a higher PPL than Direct Paraphrasing, while remaining close to the original text and other MAGE-based variants. Since Direct Paraphrasing produces very low-PPL outputs, its generations may be overly smoothed by the paraphrasing model. In contrast, DEPO moves the output distribution away from this highly regularized paraphrase style, suggesting more diverse and human-like linguistic variation while maintaining high judged quality.

Finally, the LLM ratings and BERTScore exhibit a broadly consistent trend: methods with higher BERTScore generally receive higher judge ratings, whereas lower BERTScore is often associated with lower equivalence scores. This agreement provides additional evidence that BERTScore is a reasonable proxy for semantic similarity in our setting, and supports its use as the semantic preservation signal in DEPO. A qualitative Example can be found in Fig 4.

LLM Judge Prompt Template. We use the following prompt template to configure `gpt-oss-120b` as an LLM-based paraphrase quality judge.

LLM Judge Prompt

System prompt.
You are an expert linguist and paraphrase evaluator. Your task is to assess the quality of a paraphrased text compared to the original source text. Use the following scoring criteria:
5 - Approximately equivalent: Meaning is preserved; differences are only in wording or structure.
4 - Nearly equivalent: Meaning is mostly preserved; minor factual details differ.
3 - Somewhat equivalent: Some meaning is preserved; important details or meanings differ.
2 - Topically related: The texts are on the same topic but most meaning is lost.
1 - Not topically related: The texts are not related in topic or meaning.
Provide your final output as a JSON object in this format:

```
{ "score": <score from 1 to 5>,
"justification": "<brief explanation>" }
```

User prompt.
Evaluate the following paraphrase using the criteria above:
Original Text: <original_text>
Paraphrased Text: <paraphrased_text>
What score (1 to 5) would you assign to this paraphrase, and why?

B.7 Interactive Demo and License

We release an interactive demo of our DEPO model on Hugging Face. Users can input a piece of AI-generated text to obtain its paraphrased version. The demo also reports the MAGE detector probabilities before and after paraphrasing, The demo model is released under the MIT License.

C Additional Tables Mentioned in the Main Paper

In this section, we include the tables referred to in the main paper.

Table 11: Attack effectiveness on the RAID evaluation set under different AI-text detectors.

Method	BERT Score	MAGE		RoBERTa		RADAR		Binoculars		Fast-DetectGPT	
		ASR↑	ΔAUC↓	ASR↑	ΔAUC↓	ASR↑	ΔAUC↓	ASR↑	ΔAUC↓	ASR↑	ΔAUC↓
Original	–	0.00	0.46	0.30	0.27	0.10	0.46	0.06	0.48	0.03	0.49
Direct Paraphrasing	0.883	0.00	0.38	0.84	0.07	0.35	0.39	0.36	0.42	0.27	0.48
AuthorMist	0.831	0.07	0.31	0.98	0.27	0.78	0.20	0.90	0.07	0.96	0.19
StealthRL	0.730	0.03	0.33	0.94	0.21	0.22	0.43	0.96	0.19	0.97	0.05
SilverSpeak	0.373	0.00	0.33	0.00	0.49	0.00	0.49	0.05	0.49	1.00	0.24
Adv. Paraphrase	0.879	0.00	0.48	0.97	0.25	0.99	0.26	0.58	0.40	0.21	0.48
MAGE BERTScore 0	0.821	0.96	0.27	0.98	0.30	0.89	0.14	0.99	0.16	0.99	0.06
MAGE BERTScore Linear 0.1	0.807	0.96	0.26	1.00	0.31	0.63	0.29	0.94	0.13	0.96	0.22
MAGE BERTScore Linear 0.3	0.862	0.86	0.14	0.98	0.30	0.74	0.26	0.91	0.12	0.90	0.29
MAGE BERTScore 1	0.986	0.00	0.45	0.33	0.26	0.12	0.45	0.05	0.48	0.03	0.49
MAGE BERTScore Linear 0.5	0.881	0.12	0.26	0.89	**0.10**	0.20	0.44	0.48	0.39	0.50	0.44
DEPO	0.878	**0.86**	**0.11**	**0.99**	0.33	**0.65**	**0.32**	**0.85**	**0.15**	**0.78**	**0.37**

Table 12: Attack effectiveness on the peer-review evaluation set under different AI-text detectors.

Method	BERT Score	PeerReview RoBERTa		MAGE		RADAR		Binoculars		Fast-DetectGPT	
		ASR↑	ΔAUC↓	ASR↑	ΔAUC↓	ASR↑	ΔAUC↓	ASR↑	ΔAUC↓	ASR↑	ΔAUC↓
Original	–	0.03	0.50	0.09	0.40	0.10	0.35	0.61	0.20	0.39	0.35
Direct Paraphrasing	0.818	0.04	0.50	0.07	0.39	0.53	0.13	0.84	0.11	0.64	0.28
AuthorMist	0.721	0.53	0.49	0.32	0.24	0.96	0.27	1.00	0.37	1.00	0.32
StealthRL	0.698	0.40	0.49	0.48	0.17	0.43	0.19	0.99	0.28	1.00	0.33
SilverSpeak	0.392	0.43	0.50	0.00	0.47	0.39	0.22	0.06	0.49	1.00	0.33
Adv. Paraphrase	0.819	0.04	0.50	0.13	0.32	0.58	0.10	0.89	0.09	0.68	0.27
MAGE BERTScore 0	0.777	0.44	0.50	0.97	0.29	0.91	0.16	1.00	0.27	0.98	0.17
MAGE BERTScore Linear 0.1	0.737	0.63	0.49	0.98	0.31	0.76	0.06	0.99	0.10	0.98	0.09
MAGE BERTScore Linear 0.3	0.798	0.39	0.49	0.96	0.25	0.90	0.12	0.99	0.13	0.96	0.04
MAGE BERTScore 1	0.913	0.02	0.50	0.07	0.41	0.13	0.34	0.62	0.21	0.39	0.37
MAGE BERTScore Linear 0.5	0.781	0.21	**0.49**	0.27	0.24	0.52	0.14	0.88	**0.01**	0.77	0.15
DEPO	0.825	**0.41**	**0.49**	**0.94**	**0.23**	**0.84**	**0.07**	**0.95**	0.06	**0.78**	**0.12**

Table 13: Cross-detector evaluation of the RAID-trained, RADAR-targeted DEPO policy on the RAID evaluation set.

Method	BERT Score	MAGE		RoBERTa		RADAR		Binoculars		Fast-DetectGPT	
		ASR↑	ΔAUC↓	ASR↑	ΔAUC↓	ASR↑	ΔAUC↓	ASR↑	ΔAUC↓	ASR↑	ΔAUC↓
Original	–	0.00	0.47	0.54	0.35	0.44	0.42	0.23	0.31	0.20	0.32
Direct Paraphrasing	0.877	0.01	0.38	0.96	0.07	0.72	0.37	0.50	0.29	0.34	0.32
AuthorMist	0.825	0.09	0.32	0.94	0.07	0.84	0.24	0.91	0.01	0.89	0.09
StealthRL	0.733	0.03	0.34	1.00	0.03	0.26	0.46	0.99	0.31	0.98	0.20
SilverSpeak	0.374	0.00	0.33	0.21	0.48	0.14	0.49	0.25	0.48	1.00	0.23
Adv. Paraphrase	0.876	0.02	0.37	0.98	0.03	0.77	0.35	0.57	0.27	0.41	0.31
RADAR BERTScore 0	0.775	0.33	0.19	0.99	0.21	0.98	0.29	0.99	0.28	1.00	0.16
RADAR BERTScore Linear 0.1	0.783	0.34	0.17	0.99	0.21	0.97	0.27	0.99	0.23	0.99	0.14
RADAR BERTScore 1	0.991	0.00	0.47	0.56	0.35	0.46	0.41	0.23	0.32	0.19	0.32
DEPO	0.850	**0.19**	**0.27**	**0.98**	**0.13**	**0.95**	**0.09**	**0.87**	**0.04**	**0.84**	**0.14**

Table 14: Complete hyperparameters and configuration.

Parameter	**Value**
Model & LoRA	
Base model	Qwen/Qwen3-4B-Instruct-2507
LoRA rank	32
LoRA alpha	64
LoRA dropout	0.05
Training	
Algorithm	DEPO with Lagrangian GRPO-style policy optimization
Optimizer	AdamW
Adam β_1, β_2	0.9, 0.999
Learning rate	2×10^{-4}
Batch size (prompts)	4
Gradient accumulation	4
Group size G	8
Training epochs	3
Training samples	1,000 AI-generated samples
Temperature	1.0
KL penalty coefficient β	0.04
Reference policy	Qwen3-4B-Instruct-2507 (frozen)
Hardware	NVIDIA A6000 Ada / H100 GPU
Detector reward	Detector $1 - P(AI)$
Semantic reward	BERTScore-F1 with microsoft/deberta-xlarge-mnli
Radom seed	42
Lagrangian & Constraints	
Target semantic threshold τ_{sem}	0.85
Multiplier init λ_0	0.0
Multiplier LR η_λ	0.1
Multiplier max $\lambda_{\max}$	10.0
Inference	
Temperature	0.9
Top-p	0.95
Max tokens	512
Default prompt template	"Paraphrase the following text while preserving its meaning. Return only the paraphrase. Text: [TEXT]"
Detectors	
MAGE detector	Longformer-based classifier from yaful/MAGE
RoBERTa OpenAI	openai-community/roberta-large-openai-detector
RADAR	TrustSafeAI/RADAR-Vicuna-7B
Binoculars (lightweight)	gpt2-medium + gpt2-large
Fast-DetectGPT	Scoring LM: EleutherAI/gpt-neo-2.7B
Peer-review RoBERTa	roberta-base model fine-tuned on peer-review dataset
Evaluation	
Main evaluation test samples	1,600 AI-written samples
Token window	100–500 tokens
FPR calibration	1% on 10,000 human samples using quantile threshold
Candidates per sample	1
Random seed	2026

Detection Evasion Case Study

Method	Rewritten Text	BERTScore	$1 - P(AI) \uparrow$
Original	The 23-year-old crowdfunded her way to Rio to compete in the women's single sculls, but finished 20th in the finals last Saturday. However, six days later Snoop Dogg posted a picture of her on Instagram, with the caption "Congrats to Emma Twigg who just won gold at the Olympics!" The post went viral and quickly hit thousands of likes and shares. While the post was well-intentioned, it was incorrect given that she did not win a medal. Twigg later responded to the post, thanking him for his support but clarifying that she had in fact finished 20th.	1.000	0.023
Direct Paraphrasing	The 23-year-old funded her own trip to Rio to compete in the women's single sculls, but placed 20th in the finals last Saturday. Six days later, Snoop Dogg shared a photo of her on Instagram, captioning it "Congrats to Emma Twigg who just won gold at the Olympics!" The post went viral, amassing thousands of likes and shares. Although meant as a compliment, it was factually wrong since Twigg did not win a medal. She later addressed the post, thanking Snoop Dogg for his support while correcting the record by stating she had finished 20th.	0.912	0.593
Authormist	Emma Twigg raised funds herself and competed in singles rowing at Rio Olympics. She came in twenty place in finals last Saturday. But six days later Snoop Dogg posted an image of her on Instagram along with caption "Congratulations to Emma Twigg who just won gold at the Olympics." This post spread widely and received many likes and shares. Although this was meant well, it was incorrect since she didn't actually win any medals. Later Twigg thanked Snoop for his support but clarified that indeed she had placed twentieth.	0.861	0.001
StealthRL	The 23-year-old funded her trip to Rio herself to compete in the women's single sculls, but placed 20th in the finals last Saturday.	0.640	0.005
SilverSpeak	T h e 23- ea r -old cr o w d f u n ded h er way t o R io t o compete ì n t h W o me n' S s n9l sCu l ls, but fì nish d 20th n the f i n a l S l a s t Satu r a . Howeve r , s i x d a y s l t r S n o op D。 g g o t ed a pict u re o f h r on I n s t agram, Wì t h t e c p t ion "Con9rat t o Enm Twi g g w h o u St w°n gol at the O l m ic S!" T h os t Wen t vi r l an qui c k l h ì t tho u s nds of l ikes and s a r es. Wh ì l e th e po s t was w l l-i n t e n tion d, it w a s ì nc°rre t giv n th a t she d i d n ot w n a meda l . Twig9 l at r reS o n e d t o the pos t , t ha n king im f or h s s u p or t b u t lar i f i n g t h a t sh e a d n f a Ct f n i h e d 20 t h.	0.319	0.054
Adv. Paraphrase	The 23-year-old funded her own journey to Rio to compete in the women's single sculls, but finished 20th in the final last Saturday. Six days later, Snoop Dogg shared a photo of her on Instagram, captioning it "Congrats to Emma Twigg who just won gold at the Olympics!" The post went viral, amassing thousands of likes and shares. Though well-meaning, it was factually wrong since Twigg did not win a medal. She later addressed the post, appreciating Snoop Dogg's support while correcting the record by stating she had finished 20th.	0.916	0.936
BERTScore 0 (Detector Evasion Only)	Twenty-three-year-old Emma Twigg raised funds to travel to Rio and compete in the women's single sculls, but placed 20th in the finals last Saturday. Six days later, Snoop Dogg shared a photo of her on Instagram, celebrating her as an Olympic gold medalist. The post, widely shared and well-received, was inaccurate since Twigg hadn't won, a fact she later, in response, acknowledged—thanks him for the support, and, clarified she' d actually finished 20th, not won, gold.	0.834	1.000
BERTScore 1 (Semantic Only)	The 23-year-old crowdfunded her way to Rio to compete in the women's single sculls, but finished 20th in the finals last Saturday. However, six days later Snoop Dogg posted a picture of her on Instagram, with the caption "Congrats to Emma Twigg who just won gold at the Olympics!" The post went viral and quickly amassed thousands of likes and shares. While the post was well-intentioned, it was incorrect given that she did not win a medal. Twigg later responded to the post, thanking him for his support but clarifying that she had in fact finished 20th.	0.997	0.020
BERTScore Linear 0.1	The 23-year-old funded her own trip to Rio to compete in the women's single sculls, ending up in 20th place in the finals last Saturday. Six days later, Snoop Dogg shared a picture of her on Instagram, captioning it, "Congrats to Emma Twigg who just won gold at the Olympics!"—a post that quickly went viral, accumulating thousands of likes and shares.—though factually wrong, as Twigg, a 20th-place finisher, didn' t win a medal. Responding to the post, Twigg thanked Snoop Dogg for the support but, correctly, clarified that—despite the viral hype—she had, in fact, finished 20th, not won gold.	0.847	1.000
BERTScore Linear 0.3	The 23-year-old raised funds to travel to Rio and compete in the women's single sculls, ending up in 20th place in the finals last Saturday. Six days later, Snoop Dogg shared a photo of her on Instagram, captioning it, "Congratulations to Emma Twigg, who just won gold at the Olympics!"—a post that went viral, accumulating thousands of likes and shares. Though well-meaning, the post was factually wrong, as Twigg had not, in fact, won a medal. She later clarified, thanking Snoop Dogg for his support and confirming she had finished 20th.	0.885	0.999
BERTScore Linear 0.5	The 23-year-old crowdfunded her trip to Rio to compete in the women's single sculls, but placed 20th in the finals last Saturday. Six days later, Snoop Dogg shared a photo of her on Instagram, captioning it, "Congratulations to Emma Twigg, who just won gold at the Olympics!" The post went viral, amassing thousands of likes and shares. Though well-meaning, it was factually wrong, as she did not win a medal. Twigg later addressed the post, thanking him for his support and clarifying that she had actually finished 20th. *(end) - the final, accurate, and fully paraphrased, coherent, and natural-sounding version, no extra elements.*	0.888	1.000
DEPO	The 23-year-old financed her trip to Rio to compete in the women's single sculls, but ended up finishing 20th in the final last Saturday. Six days later, Snoop Dogg shared a photo of her on Instagram, captioning it "Congrats to Emma Twigg who just won gold at the Olympics!" The post, though meant well, was false, as Twigg did not, in fact, win a medal. She later addressed the post, thanking Snoop Dogg for his support but, clearly, stating she had finished, 20th.	0.890	0.998

Figure 4: **Qualitative comparison of paraphrasing attack results.** This figure presents representative paraphrasing attack outputs for a given AI-generated sentence(Original) using different methods described in Appendix A.2.