shushantatudublin at PAN 2026: Robust Detection of AI-Generated Texts via Multi-Generator and Multi-Evasion Adversarial Training

Notebook for the PAN Lab at CLEF 2026

Shushanta Pudasaini[1], Luis Miralles-Pechuán[1,3], David Lillis[2] and Marisa Llorens Salvador[1]

[1]School of Computer Science, Technological University Dublin, Dublin, Ireland
[2]University College Dublin, Dublin, Ireland
[3]European, EUT+ Data Science Institute, European University of Technology, European Union

Abstract

The rapid evolution of large language models (LLMs) has made the reliable identification of AI-generated text increasingly challenging, particularly due to evasion strategies such as paraphrasing and visually similar character substitutions (homoglyphs) designed to bypass existing detectors. To address this problem, the PAN Lab at CLEF 2026 introduced the Voight-Kampff Generative AI Detection task, which explicitly emphasises *robustness*: test inputs may originate from unseen generators and may be modified by unknown obfuscations before evaluation. Under these conditions, conventional detectors are known to degrade sharply.

We address this robustness gap by modelling detection as a two-player adversarial game, in which a detector is jointly trained with an evolving attacker. Training data is drawn from a multi-generator corpus comprising human-authored and machine-generated texts from twenty-two recent LLMs spanning eight model families. To expose the detector to a diverse attack space, we introduce a two-track evasion framework. Track A consists of a learnable neural paraphraser trained to evade the detector via clipped Proximal Policy Optimisation, while Track B comprises a set of deterministic linguistic and character-level obfuscations applied on-the-fly as adversarial data augmentation. The detector is optimised under a unified, per-attack re-weighted objective that enforces simultaneous separation of human, clean machine-generated, and adversarially modified text.

To stabilise training, we incorporate a suite of techniques including label smoothing, controlled detector update cadence, a dynamic freeze rule, reward clipping, and numerical-safety safeguards, which together prevent collapse of the adversarial game during long runs.

On a held-out validation split, the resulting RoBERTa-large detector achieves a macro-averaged AUROC of **99.60** across all generators, an F_1 score of 97.98 at the default threshold, and maintains performance above 99.8 AUROC under all training-time evasion attacks. These results demonstrate that explicitly training for adversarial robustness substantially improves detector reliability under challenging conditions.

Keywords

AI-generated text detection, adversarial learning, robustness, reinforcement learning, Large Language Models

1. Introduction

The rapid evolution of large language models (LLMs) has made the automatic distinction between human-written and AI-generated text a practical necessity in journalism, education, scientific publishing, and online trust and safety. The PAN lab at CLEF has tracked this problem through its Voight-Kampff Generative AI Detection task [1, 2], where systems must assign a score indicating whether a given text was written by a human (class 0) or a machine (class 1).

The 2026 edition of the task introduces a substantially harder setting [3, 4]. LLMs are explicitly instructed to mimic specific human authors, and the test set includes deliberately engineered adversarial texts—such as outputs from unseen models and novel obfuscation strategies—designed to probe detector

CLEF 2026 Working Notes, 21 – 24 September 2026, Jena, Germany
*Corresponding author.
†These authors contributed equally.
D23129142@mytudublin.ie (S. Pudasaini); luis.miralles@TUDublin.ie (L. Miralles-Pechuán); david.lillis@ucd.ie (D. Lillis); marisa.llorens@TUDublin.ie (M. L. Salvador)
0000-0003-1612-4510 (S. Pudasaini)

robustness. In a builder–breaker setup with the ELOQUENT lab, submitted systems are evaluated on inputs specifically crafted to deceive them, rather than on held-out data from the same distribution.

In this setting, conventional supervised detectors are known to fail. A RoBERTa classifier fine-tuned on outputs from a specific LLM may achieve high in-distribution accuracy, but its performance degrades sharply under paraphrasing [5, 6], character-level obfuscations such as homoglyph substitution [7], and generator shift [8]. Reported accuracy on clean held-out splits, therefore, reflects interpolation rather than the extrapolation required by PAN 2026.

Our approach is based on the premise that robustness must be explicitly trained rather than assumed. We build on the RADAR framework [9], which formulates detection as a minimax game between a classifier and a neural paraphraser, and extend it in three directions aligned with the PAN 2026 threat model:

Multi-generator training. We train on outputs from twenty-two recent LLMs spanning eight model families (GPT, Gemini, Llama, Mistral, Qwen, Falcon, DeepSeek, and PaLM). This diversity exposes the detector to a wide range of generation styles and mitigates overfitting to specific models.

Two-track multi-evasion adversarial training. We combine a learnable neural paraphraser, trained with clipped Proximal Policy Optimisation (PPO) [10], with a complementary track of deterministic evasion strategies, including synonym substitution, homoglyph replacement, article deletion, and misspelling. These attacks are applied on-the-fly, and the detector is trained with a unified objective to distinguish human text from both clean and adversarially modified machine-generated text.

Stabilisation suite. Adversarial training of this kind is inherently unstable. We introduce a practical stabilisation suite—including label smoothing, detector update scheduling, an AUROC-triggered freeze rule, reward clipping, NaN-safe logits processing, and memory-bounded gradient accumulation—which enables reliable convergence over long training runs.

The resulting RoBERTa-large detector achieves a macro-averaged AUROC of 99.60 across all 22 generators on a held-out validation set and maintains performance above 99.8 AUROC under every training-time evasion attack, while preserving high accuracy on clean inputs. We submit our system to PAN 2026 via the TIRA platform [11] and release the full training pipeline to support reproducibility.

2. Background and Related Work

This section reviews the work most relevant to our approach. We first summarise the two dominant families of AI-generated text detectors, together with the baselines provided for PAN 2026; next, we discuss adversarial training and the RADAR framework on which our method builds; and finally, we review the benchmarks that establish multi-generator, multi-attack evaluation as standard practice.

Supervised and zero-shot detectors. Two main families of AI-generated text detectors dominate the literature. *Supervised* detectors fine-tune pretrained encoders—most commonly RoBERTa [12]—as binary classifiers on labelled human and machine-generated text. While these models achieve high accuracy in-distribution, they are notoriously brittle under domain and generator shift.

In contrast, *zero-shot* detectors exploit statistical signatures of generative models without task-specific training. GLTR visualises token-rank distributions [13], DetectGPT estimates curvature in log-probability under perturbation [14], Fast-DetectGPT replaces costly perturbations with conditional probability estimates [15], and Binoculars compares the perplexity of related LLMs [16]. Hybrid approaches such as Ghostbuster combine model likelihoods with structured feature search [17].

For PAN 2026, the organisers provide several baselines, including a linear Support Vector Machine (SVM) with TF–IDF features, a PPMd compression-based cosine similarity method [18], and Binoculars [16]. Our method falls within the supervised paradigm but explicitly addresses its central weakness: lack of robustness.

Evasion of detectors. A substantial body of work demonstrates that current detectors are easily evaded. Krishna et al. [5] show that strong paraphrasing reduces multiple detectors to near-random

performance. In contrast, Sadasivan et al. [6] argue that recursive paraphrasing imposes a fundamental limit on reliable detection.

Beyond paraphrasing, character-level and formatting perturbations attack the tokenisation process rather than semantics. These include homoglyph substitution [19], invisible Unicode insertion, whitespace manipulation, and casing changes—all of which can bypass detectors while preserving meaning. The PAN 2026 "unknown obfuscations" setting directly operationalises this threat model. In contrast to prior work that treats such attacks purely as evaluation tools, we incorporate them directly into training.

Adversarial training and RADAR. Watermarking methods embed detectable signals during generation [20], but require model-level cooperation and are fragile to paraphrasing. RADAR [9] instead frames detection as a minimax game: a classifier is trained against a neural paraphraser that is itself optimised, via reinforcement learning, to evade detection. At inference time, only the detector is used.

Our approach builds on RADAR but extends it beyond its original single-generator, single-attack setting. We train across a diverse set of generators and replace the single paraphraser with a structured, two-track adversarial setup combining neural and deterministic evasion strategies.

Benchmarks. Robust evaluation requires diversity in both generators and attack settings. M4 provides a multi-generator, multi-domain, multilingual benchmark [21]; HC3 compares human and ChatGPT responses across domains [22]; and RAID [8] introduces a benchmark combining multiple generators with a suite of adversarial attacks. RAID, in particular, establishes multi-generator coverage and built-in obfuscation as key components of realistic evaluation.

Our work follows this paradigm but shifts obfuscation from a static evaluation component to a dynamic training signal, aligned with the adversarial nature of PAN 2026.

3. Methodology

In simple terms, our method trains a detector using many different generators and many different attacks at the same time, so it learns to be robust. Our algorithm is trained on a multi-generator AI text detection corpus pairing human-written texts with machine-generated texts from twenty-two distinct LLMs. The dataset is provided as a fixed train/validation split: the training partition contains $9{,}101$ human texts and $14{,}606$ machine texts, while the held-out validation set contains $1{,}277$ human and $2{,}312$ machine texts. Because this split is fixed, model selection is strictly separated from training, ensuring fair comparison across runs.

A defining characteristic of the dataset is its generator diversity. The twenty-two generators span eight model families and include both earlier chat models and recent frontier systems. The full list and per-generator performance are reported in Table 2. Notably, two classes (`gemini-pro-paraphrase` and `gpt-4-turbo-paraphrase`) already contain paraphrased machine text, providing an initial signal of paraphrase-based evasion. In our framework, however, all machine texts are treated as clean inputs, and all obfuscations are applied within the training loop, ensuring full control over the adversarial process.

Adversarial training setup. Our method jointly trains two models in an adversarial game. The *detector* D_ϕ is a RoBERTa-large classifier [12] that outputs the probability $D_\phi(\textsc{human} \mid x) \in [0, 1]$. The *paraphraser* G_σ is a T5 model [23] that rewrites machine text x_m into a paraphrase x_p. The detector seeks to classify all machine-derived text as non-human, while the paraphraser aims to generate rewrites that appear human-like. At inference time, only the detector is retained.

Unlike RADAR [9], which uses a single adversarial paraphraser, our method introduces a structured two-track attack pool (Figure 1):

Track A – learnable. A neural paraphraser G_σ, trained with reinforcement learning to adaptively evade the detector. This track models a strong, detector-aware adversary.

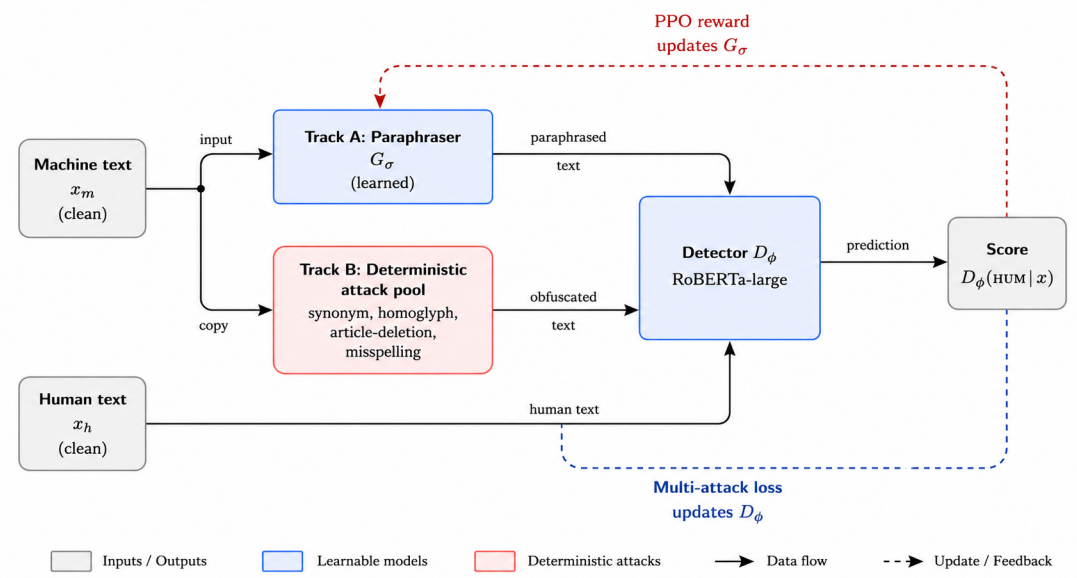

Figure 1: Overview of the two-track adversarial framework. Clean machine text is transformed via a learnable paraphraser (Track A) and deterministic obfuscations (Track B). The detector is trained jointly on human, clean machine, and adversarial inputs under a unified objective.

Track B – deterministic. A set of rule-based obfuscations applied directly to machine text. These have no trainable parameters and serve purely as adversarial data augmentation, modelling inexpensive and diverse evasion strategies.

Each training step proceeds as follows: (i) sample clean machine texts; (ii) generate paraphrases via Track A and compute rewards; (iii) apply all Track B transformations; (iv) update the paraphraser using PPO; (v) update the detector on human, clean machine, and all adversarial variants. The updates are interleaved so that both models continually adapt to each other.

3.1. Detector objective

Let $p = D_\phi(\text{HUMAN} \mid x)$ denote the predicted human probability. Each instance incurs a smoothed binary cross-entropy loss with label smoothing coefficient α:

$$\ell_{\text{H}}(p) = -(1-\alpha)\log p - \alpha \log(1-p), \tag{1}$$

$$\ell_{\text{M}}(p) = -(1-\alpha)\log(1-p) - \alpha \log p. \tag{2}$$

Label smoothing prevents overconfident predictions and mitigates memorisation of the finite training corpus.

The key contribution of our framework is the *multi-attack* objective. Let $\mathcal{A}$ denote the set of attack variants:

$$\mathcal{A} = \{\text{CLEAN}, \text{T5}, \text{SYN}, \text{HOMO}, \text{ART}, \text{MIS}\}.$$

The detector minimises

$$\mathcal{L}_D = \mathbb{E}_{x_h}[\ell_{\text{H}}(D_\phi(x_h))] + \lambda \sum_{a \in \mathcal{A}} w_a \, \mathbb{E}_{x \in X_a}[\ell_{\text{M}}(D_\phi(x))]. \tag{3}$$

Because all attack variants contribute simultaneously, the detector must learn a unified decision boundary rather than specialising to individual evasion strategies. In practice, the loss is computed via gradient accumulation to maintain manageable memory usage for RoBERTa-large.

3.2. Track A: Adversarial paraphraser

The paraphraser G_σ is trained using the clipped PPO objective with entropy regularisation [9, 10].

Reward. Given a sampled paraphrase $x_p \sim \pi_\sigma(\cdot \mid x_m)$, the reward is

$$r(x_p) = \mathrm{clip}(D_\phi(\textsc{human} \mid x_p), r_{\min}, r_{\max}), \tag{4}$$

so that higher reward corresponds to stronger detector evasion.

Advantage. Rewards are normalised:

$$\hat{A} = \frac{r - \bar{r}}{\sigma_r + \eta}, \tag{5}$$

removing the need for a value network.

Policy update. With importance ratio $\rho = \pi_\sigma / \pi_{\text{old}}$, the loss is

$$\mathcal{L}_G = -\mathbb{E}\big[\min(\rho\hat{A}, \mathrm{clip}(\rho)\hat{A})\big] + \beta_{\text{KL}}\mathcal{L}_{\text{KL}} + \gamma\mathcal{L}_{\text{E}}. \tag{6}$$

The KL term prevents drift from the base language model, while entropy regularisation maintains diversity. Because the detector is continuously evolving, $\mathcal{L}_G$ typically oscillates rather than converging monotonically.

3.3. Track B: Deterministic evasion pool

Track B applies parameter-free transformations targeting different linguistic levels:

Synonym replacement. WordNet-based substitution of content words, preserving semantics while altering lexical patterns.

Homoglyph substitution. Replacement of ASCII characters with visually identical Unicode glyphs, disrupting tokenisation without affecting readability.

Article deletion. Random removal of articles (`a`, `an`, `the`), simulating non-native writing.

Keyboard misspelling. Injection of realistic typos via key-adjacent substitutions and character edits.

These transformations collectively approximate a broad, low-cost adversarial space consistent with real-world obfuscation.

3.4. Stabilisation strategies

Adversarial training over long horizons is highly unstable. The following techniques were essential for convergence:

1. **Label smoothing.** Prevents early overfitting and logit saturation.
2. **Detector update cadence.** Updating the detector every two steps prevents it from overwhelming the paraphraser.
3. **Dynamic freezing.** Detector updates are temporarily paused when validation AUROC exceeds a threshold, preserving a competitive adversarial balance.
4. **Reward clipping.** Stabilises PPO gradients when detector confidence is extreme.
5. **Numerical safeguards.** NaN-safe logits, gradient sanitisation, and value clamping prevent numerical instability from corrupting training.

Table 1
Configuration of the submitted algorithm. Values correspond to the released implementation.

Group	Parameter	Value
Detector D_ϕ	Architecture	RoBERTa-large
	Maximum sequence length	512 tokens
	Batch size	8
	Learning rate (AdamW)	3×10^{-6}
	Update cadence	every 2 outer steps
Paraphraser G_σ	Architecture	T5 (paraphrase checkpoint)
	Learning rate (AdamW)	2×10^{-5}
	Decoding	sampling, T=1.0, top-k=50
	Maximum new tokens	128
	Repetition penalty	1.3
PPO (cppo-ep)	Buffer size	64
	Epochs per buffer	8
	Clip ε	0.2
	KL coeff. β_{KL}	0.001
	Entropy coeff. γ	0.01
	Reward clip $[r_{\min}, r_{\max}]$	$[0.01,\ 0.99]$
Detector objective	Human/machine balance λ	0.5
	Label smoothing α	0.15
Track B rates	synonym / homoglyph	0.20 / 0.10
	article-del. / misspell	0.50 / 0.08
Training loop	Outer steps	200
	Warmup / patience	50 / 25
	Freeze threshold	0.995 macro-AUROC
	Generator schedule	mixed (all 22)
	Random seed	42

4. Experimental Setup

Models. The detector is `roberta-large` (355M parameters), a transformer-based encoder with a binary classification head. The Track A paraphraser is the publicly available T5 checkpoint from the Huggingface platform `ramsrigouthamg/t5_paraphraser`. We selected this model over larger alternatives due to its stable and untied parameterisation, which avoids the gradient instabilities we observed with heavier checkpoints during PPO training. At inference time, only the detector is used.

Configuration. Table 1 summarises the configuration of the submitted run. The detector is trained jointly on human texts, clean machine texts, Track A paraphrases, and the four Track B transformations described in Section 3.3. Machine texts are sampled using a *mixed* schedule, such that each outer step draws proportionally from all twenty-two generators.

Training proceeds for up to 200 outer steps with early stopping based on validation performance (patience 25 evaluations). Validation macro-AUROC is computed every 5 steps, and the checkpoint with the highest score is retained. In the submitted run, the best checkpoint was reached early, at outer step 35. After this point, the dynamic freeze rule (Section 3.4) prevented further detector updates, while training continued until early stopping triggered at step 160. The full run required approximately 3.5 hours on a single NVIDIA H100 GPU.

Evaluation. We evaluate performance on the held-out validation set using two complementary diagnostics.

Per-generator AUROC measures detection performance separately for each generator, comparing

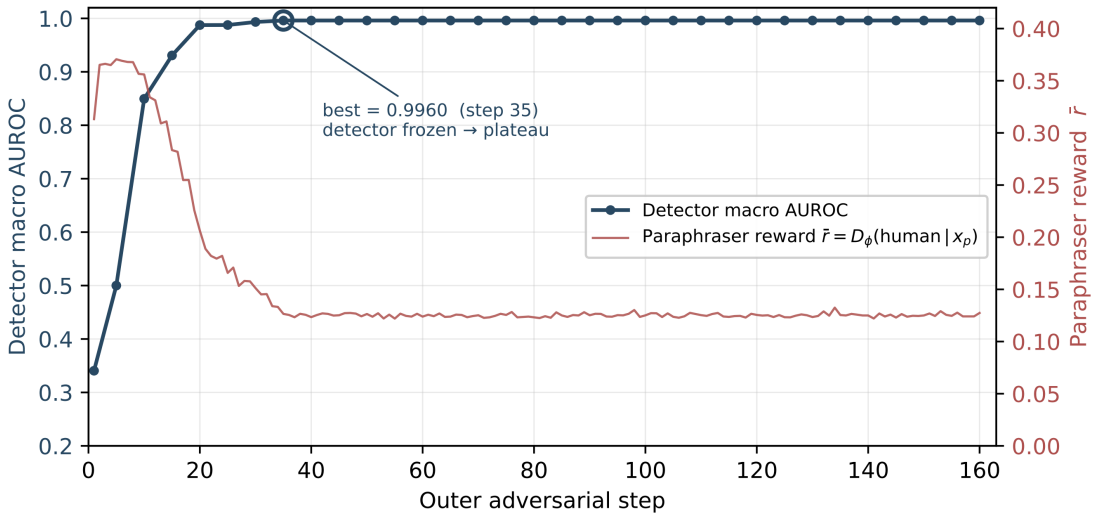

Figure 2: Adversarial training dynamics over 160 outer steps. Left axis: detector macro-AUROC on the validation set (measured every five steps). Right axis: mean paraphraser reward. The detector peaks at step 35, after which it is frozen while the paraphraser continues probing.

its machine texts against the shared pool of human texts. The macro-average across all twenty-two generators serves as the primary model-selection metric.

Per-attack AUROC evaluates robustness to evasion by applying each attack operator to the entire validation machine set and scoring it against the human texts. This isolates specific weaknesses, as degraded performance indicates susceptibility to a particular attack type.

We additionally report threshold-based metrics—precision, recall, and F_1—at the default decision threshold of 0.5, aligning with the PAN 2026 evaluation suite (which also includes $F_{0.5u}$). AUROC remains the primary measure, as it evaluates ranking quality independent of threshold and directly corresponds to the ROC-AUC component used in PAN evaluation.

5. Results

We analyse the results along four axes: adversarial training dynamics, per-generator detection performance, robustness to evasion attacks, and threshold-based performance on the validation set.

5.1. Adversarial training dynamics

Figure 2 shows the evolution of the adversarial game over the 160 outer steps executed before early stopping. At step 1, before any detector updates, the randomly initialised classifier performs well below chance (macro-AUROC 34.1). Within ten steps, it becomes a competent detector (macro-AUROC $\approx$ 85), surpasses 98 by step 20, and reaches its peak performance of **99.60** at step 35.

Once the macro-AUROC exceeds the 99.5 freeze threshold, the dynamic freeze rule (Section 3.4) halts further detector updates. Training then proceeds with a fixed detector until early stopping terminates the run at step 160. The submitted model is therefore the checkpoint from step 35.

The paraphraser reward exhibits complementary behaviour. It begins at $\bar{r} \approx 31$, indicating that the initial paraphraser successfully fools the untrained detector roughly one third of the time. The reward briefly rises above 36 as it exploits the still-weak detector, then declines as the detector improves, stabilising around $\bar{r} \approx 12$–13 after the detector freezes. Importantly, the reward does not collapse: even against a fixed detector, approximately one in eight paraphrases remains partially successful. This bounded, non-vanishing reward is a clear indicator of a stable adversarial game and, in our experiments, depends critically on the stabilisation strategies described in Section 3.4.

Table 2
Per-generator AUROC of the selected detector on the held-out validation split (best checkpoint, step 35). Generator names are abbreviated; the macro-average over all 22 generators is the model-selection metric.

Generator	AUROC	Generator	AUROC
deepseek-r1-32b	99.98	gpt-4o-mini	99.99
falcon3-10b	99.92	llama-2-70b	99.74
gemini-1.5-pro	96.89	llama-2-7b	99.67
gemini-2.0-flash	99.98	llama-3.1-8b	99.29
gemini-pro	98.51	llama-3.3-70b	99.93
gemini-pro-para	98.84	ministral-8b	99.97
gpt-3.5-turbo	99.99	mistral-7b-v0.2	99.65
gpt-4-turbo	100.00	mixtral-8x7b	99.47
gpt-4-turbo-para	99.63	o3-mini	99.99
gpt-4.5-preview	99.80	qwen1.5-72b	99.95
gpt-4o	99.99	text-bison-002	99.92
Macro-average (22 generators)			**99.60**

5.2. Per-generator detection performance

Table 2 reports per-generator AUROC for the selected checkpoint. The detector exceeds 99 AUROC on 19 of the 22 generators and 99.9 on 12 of them, achieving perfect separation for gpt-4-turbo.

Robustness to recent models is particularly relevant for the PAN 2026 "unseen model" setting. Notably, frontier systems such as o3-mini (99.99), gemini-2.0-flash (99.98) , gpt-4.5-preview (99.80) (99.98), llama-3.3-70b-instruct (99.93), and deepseek-r1-distill-qwen-32b are detected with accuracy comparable to older models, suggesting strong generalisation across both architectures and generations.

The most challenging cases are gemini-1.5-pro (96.89) and gemini-pro (98.51), with the former acting as a clear outlier that largely determines the macro-average. This behaviour validates the use of macro-AUROC as the selection metric, as it prevents weaker generators from being masked by the majority of strong cases.

Paraphrased variants (gemini-pro-paraphrase at 98.84 and gpt-4-turbo-paraphrase at 99.63) consistently score below their non-paraphrased counterparts, confirming that paraphrasing remains an effective evasion strategy. The overall performance spread across all generators is limited to 3.1 AUROC, indicating that multi-generator training successfully mitigates overfitting to individual model characteristics.

5.3. Robustness to evasion attacks

Table 3 evaluates robustness to evasion strategies. Each attack is applied to the full validation machine set and re-evaluated against the human texts.

The detector maintains AUROC $\geq$ 99.80 across all attacks and achieves 100.00 under both the learnable T5 paraphraser and homoglyph substitution. Notably, all attacked conditions perform at or above the clean baseline (99.66), indicating no observable robustness–accuracy trade-off.

This behaviour directly reflects the multi-attack training objective (Eq. (7)): adversarial variants are incorporated as standard training inputs rather than treated as out-of-distribution cases. The slightly lower score for clean data suggests that residual errors originate from inherently difficult clean examples—primarily gemini-1.5-pro—rather than from weaknesses against specific attack types.

$$\mathcal{L}_D \;=\; \mathbb{E}_{x_h}\big[\ell_{\text{H}}(D_\phi(x_h))\big] \;+\; \lambda \sum_{a\in\mathcal{A}} w_a \; \mathbb{E}_{x\in X_a}\big[\ell_{\text{M}}(D_\phi(x))\big], \tag{7}$$

Table 3
Per-attack AUROC of the selected detector. Each evasion operator is applied to the full validation machine set, which is then re-scored against the human texts. CLEAN is the unattacked baseline.

Track	Evasion attack	AUROC
A	T5 paraphrase (learnable)	100
B	Synonym replacement	99.98
B	Homoglyph substitution	100.00
B	Article deletion	99.82
B	Keyboard misspelling	99.96

5.4. Threshold-based performance

While AUROC captures ranking quality, PAN 2026 also evaluates performance at a fixed operating point via F_1 and $F_{0.5u}$. At the default decision threshold of 0.5, the detector classifies the $3{,}589$ validation samples ($1{,}277$ human and $2{,}312$ machine) with $2{,}227$ true positives, $1{,}270$ true negatives, 7 false positives, and 85 false negatives.

This corresponds to a precision of 99.69, recall of 96.32, and F_1 score of 97.98, with a pooled AUROC of 99.69. The operating point is deliberately conservative: the false-positive rate is approximately 0.5%, which is desirable in practice, as misclassifying human-written text as machine-generated incurs higher cost in most deployment settings.

5.5. Discussion

The robustness results in Table 3 reflect performance against *known* attacks included in training. The PAN 2026 test set introduces unseen obfuscations, and some performance degradation under these conditions is expected.

Our hypothesis is that training against a structured pool of semantic-, character-, and surface-level transformations yields a decision boundary that generalises to *related* unseen attacks. The near-identical performance across clean and attacked validation data supports this hypothesis, although it does not conclusively establish out-of-distribution robustness.

Similarly, the reported macro-AUROC of 99.60 is an in-distribution estimate, as the validation data shares generators and domains with training. The PAN 2026 evaluation, which introduces both unseen generators and novel obfuscations, will therefore provide the definitive measure of generalisation. The results presented here demonstrate strong capacity under known conditions and suggest—but do not guarantee—robust transfer.

6. Conclusion

We presented an adversarially trained detector for the PAN 2026 Voight-Kampff Generative AI Detection task that treats robustness as a primary training objective. Building on the RADAR framework, our approach extends the adversarial game along two key dimensions: (i) multi-generator training across twenty-two contemporary LLMs, and (ii) a two-track evasion model combining a PPO-trained neural paraphraser with a structured pool of deterministic obfuscations spanning semantic, character, and surface levels. A set of stabilisation techniques ensures that this adversarial process remains well-behaved throughout long training runs.

The resulting RoBERTa-large detector achieves a macro-AUROC of 99.60 across all generators on held-out validation data, an F_1 score of 97.98 at the default threshold, and consistently strong performance under all training-time evasion attacks, with no measurable degradation on clean inputs. These results indicate that explicitly incorporating diverse adversarial transformations into the training objective can effectively mitigate the robustness limitations of conventional supervised detectors.

Several directions remain for future work. First, incorporating decoding-strategy and sampling-parameter augmentation would reduce sensitivity to generation settings, a known weakness of supervised detectors. Second, extending the training corpus with additional public datasets (e.g., RAID [8], M4 [21], HC3 [22], and PAN releases) could further improve domain and generator generalisation. Finally, expanding the evasion pool with more diverse and adaptive transformations may enhance robustness to unseen attacks.

The PAN 2026 evaluation, which introduces novel generators and unknown obfuscations, will provide the definitive test of generalisation. We will report these results once the leaderboard is released.

7. Acknowledgment

This publication has emanated from research [conducted with the financial support of/supported in part by a grant from] Science Foundation Ireland under Grant number 18/CRT/6183. For Open Access, the author has applied a CC BY public copyright licence to any Author Accepted Manuscript version arising from this submission.

Declaration on Generative AI

During the preparation of this work, the author used LLMs to refine the wording of the manuscript, improve its grammar and phrasing, and assist with formatting. The system described in the paper is itself a study of AI-generated text; no generative AI was used to fabricate experimental data or results. After using the above tool, the author reviewed and edited the content as needed and takes full responsibility for the publication's content.

References

[1] J. Bevendorff, M. Wiegmann, M. Potthast, B. Stein, Overview of the "Voight-Kampff" generative AI authorship verification task at PAN and ELOQUENT 2024, in: Working Notes of CLEF 2024 – Conference and Labs of the Evaluation Forum, CEUR Workshop Proceedings, 2024, pp. 2486–2506.

[2] J. Bevendorff, M. Wiegmann, J. Karlgren, L. Dürlich, E. Gogoulou, A. Talman, E. Stamatatos, M. Potthast, B. Stein, Overview of the "Voight-Kampff" generative AI authorship verification task at PAN and ELOQUENT 2025, in: Working Notes of CLEF 2025 – Conference and Labs of the Evaluation Forum, CEUR Workshop Proceedings, 2025, pp. 3504–3534.

[3] J. Bevendorff, M. Fröbe, A. Greiner-Petter, A. Jakoby, M. Mayerl, P. Nakov, H. Plutz, M. Potthast, B. Stein, M. N. Ta, Y. Wang, E. Zangerle, in: M. Hagen, M. Potthast, B. Stein, P. Schaer, E. Zangerle, A. Barrón-Cedeño, A. G. S. de Herrera, E. S. Salido, S. MacAvaney, J. M. Struß (Eds.), Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Seventeenth International Conference of the CLEF Association (CLEF 2026), Lecture Notes in Computer Science, Springer, Berlin Heidelberg New York, 2026.

[4] J. Bevendorff, R. R. Gunti, J. Karlgren, M. Fröbe, M. Potthast, B. Stein, Overview of the third "Voight-Kampff" Generative AI / LLM Detection Task at PAN and ELOQUENT 2026, in: A. Barrón-Cedeño, A. G. S. de Herrera, E. S. Salido, S. MacAvaney, J. M. Struß (Eds.), Working Notes of CLEF 2026 – Conference and Labs of the Evaluation Forum, CEUR Workshop Proceedings, CEUR-WS.org, 2026.

[5] K. Krishna, Y. Song, M. Karpinska, J. Wieting, M. Iyyer, Paraphrasing evades detectors of AI-generated text, but retrieval is an effective defense, in: Advances in Neural Information Processing Systems (NeurIPS), volume 36, 2023.

[6] V. S. Sadasivan, A. Kumar, S. Balasubramanian, W. Wang, S. Feizi, Can AI-generated text be reliably detected?, Transactions on Machine Learning Research (2025).

[7] S. Pudasaini, L. Miralles, D. Lillis, M. L. Salvador, Benchmarking AI text detection: Assessing detectors against new datasets, evasion tactics, and enhanced LLMs, in: F. Alam, P. Nakov, N. Habash, I. Gurevych, S. Chowdhury, A. Shelmanov, Y. Wang, E. Artemova, M. Kutlu, G. Mikros

(Eds.), Proceedings of the 1stWorkshop on GenAI Content Detection (GenAIDetect), International Conference on Computational Linguistics, Abu Dhabi, UAE, 2025, pp. 68–77. URL: https://aclanthology.org/2025.genaidetect-1.4/.

[8] L. Dugan, A. Hwang, F. Trhlík, J. M. Ludan, A. Zhu, H. Xu, D. Ippolito, C. Callison-Burch, 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 (ACL), 2024, pp. 12463–12492.

[9] X. Hu, P.-Y. Chen, T.-Y. Ho, RADAR: Robust AI-text detection via adversarial learning, in: Advances in Neural Information Processing Systems (NeurIPS), volume 36, 2023.

[10] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, O. Klimov, Proximal policy optimization algorithms, arXiv preprint arXiv:1707.06347 (2017).

[11] M. Fröbe, J. H. Reimer, S. MacAvaney, N. Deckers, S. Reich, J. Bevendorff, B. Stein, M. Hagen, M. Potthast, Continuous integration for reproducible shared tasks with TIRA.io, in: Advances in Information Retrieval – 45th European Conference on Information Retrieval (ECIR), 2023, pp. 236–241.

[12] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, RoBERTa: A robustly optimized BERT pretraining approach, arXiv preprint arXiv:1907.11692 (2019).

[13] S. Gehrmann, H. Strobelt, A. M. Rush, GLTR: Statistical detection and visualization of generated text, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, 2019, pp. 111–116.

[14] E. Mitchell, Y. Lee, A. Khazatsky, C. D. Manning, C. Finn, DetectGPT: Zero-shot machine-generated text detection using probability curvature, in: Proceedings of the 40th International Conference on Machine Learning (ICML), 2023, pp. 24950–24962.

[15] G. Bao, Y. Zhao, Z. Teng, L. Yang, Y. Zhang, Fast-DetectGPT: Efficient zero-shot detection of machine-generated text via conditional probability curvature, in: The Twelfth International Conference on Learning Representations (ICLR), 2024.

[16] A. Hans, A. Schwarzschild, V. Cherepanova, H. Kazemi, A. Saha, M. Goldblum, J. Geiping, T. Goldstein, Spotting LLMs with binoculars: Zero-shot detection of machine-generated text, in: Proceedings of the 41st International Conference on Machine Learning (ICML), 2024.

[17] V. Verma, E. Fleisig, N. Tomlin, D. Klein, 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 (NAACL), 2024, pp. 1702–1717.

[18] D. Sculley, C. Brodley, Compression and machine learning: a new perspective on feature space vectors, in: Data Compression Conference (DCC'06), 2006, pp. 332–341. doi:10.1109/DCC.2006.13.

[19] A. Creo, S. Pudasaini, SilverSpeak: Evading AI-generated text detectors using homoglyphs, in: F. Alam, P. Nakov, N. Habash, I. Gurevych, S. Chowdhury, A. Shelmanov, Y. Wang, E. Artemova, M. Kutlu, G. Mikros (Eds.), Proceedings of the 1stWorkshop on GenAI Content Detection (GenAIDetect), International Conference on Computational Linguistics, Abu Dhabi, UAE, 2025, pp. 1–46. URL: https://aclanthology.org/2025.genaidetect-1.1/.

[20] J. Kirchenbauer, J. Geiping, Y. Wen, J. Katz, I. Miers, T. Goldstein, A watermark for large language models, in: Proceedings of the 40th International Conference on Machine Learning (ICML), 2023, pp. 17061–17084.

[21] Y. Wang, J. Mansurov, P. Ivanov, J. Su, A. Shelmanov, A. Tsvigun, C. Whitehouse, O. Mohammed Afzal, T. Mahmoud, T. Sasaki, T. Arnold, A. F. Aji, N. Habash, I. Gurevych, P. Nakov, 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 (EACL), 2024, pp. 1369–1407.

[22] B. Guo, X. Zhang, Z. Wang, M. Jiang, J. Nie, Y. Ding, J. Yue, Y. Wu, How close is ChatGPT to human experts? comparison corpus, evaluation, and detection, arXiv preprint arXiv:2301.07597 (2023).

[23] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. J. Liu, Exploring

the limits of transfer learning with a unified text-to-text transformer, Journal of Machine Learning Research 21 (2020) 1–67.