RegDetect: Short Machine-Generated Text Detection via Regeneration Drift Modeling

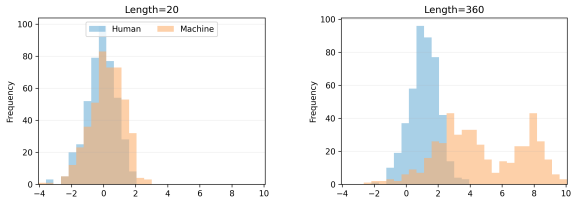
Anonymous ACL submission

Abstract

Detecting short machine-generated text remains a critical challenge due to the inherent sparsity of statistical and stylistic signals. To address this, we propose RegDetect, a **Reg**eneration-drift guided **Detect**ion framework that reformulates detection from static artifact inspection to representation drift modeling. Our method leverages the insight that an LLM-generated regeneration tends to remain closely aligned with machine-generated text, but induces larger drift when applied to human-written text. Given an input text, RegDetect prompts an LLM to generate a regenerated view, and both texts are mapped into a shared latent space using a Siamese encoder. To learn this distinction, RegDetect jointly optimizes binary classification with an asymmetric contrastive objective, pulling machine-generated texts and their regenerations closer while pushing human-authored pairs beyond a margin. Experiments on the DetectRL benchmark show that RegDetect achieves strong performance across diverse datasets, with particularly pronounced gains on challenging 30–50 word segments and consistent advantages in broader robustness and generalization.

1 Introduction

The rapid proliferation of large language models (LLMs) has substantially lowered the barrier to generating fluent and coherent text (Bubeck et al., 2023). As a consequence, machine-generated content has become increasingly common in online news, academic writing, and social media platforms (M Alshater, 2022; Goldstein, 2023). While this capability brings clear productivity benefits (Susnjak, 2022), it also raises serious concerns, including the spread of misinformation, plagiarism in academic publishing, and the erosion of trust in digital content (Susnjak, 2022; Zhou et al., 2023). These risks have motivated growing interest in the task of detecting machine-generated text.

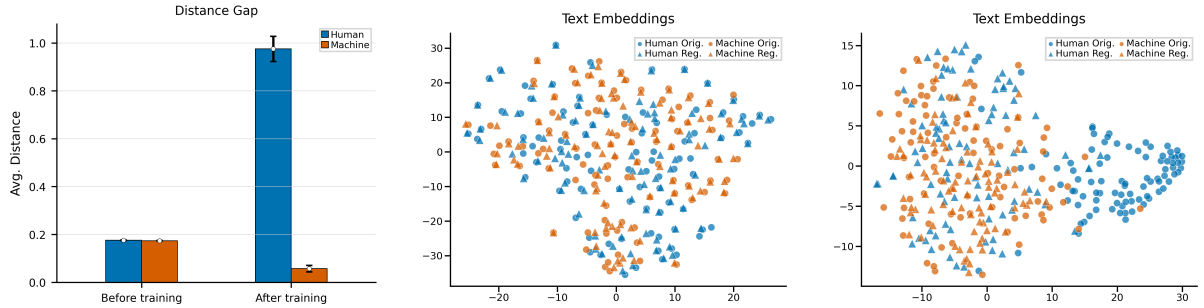


(a) Short text (b) Long text

Figure 1: Distribution of conditional probability curvatures for human-written passages and machine-generated counterparts. As text length decreases, the statistical distributions exhibit significant overlap.

Existing detection methodologies generally fall into two paradigms: zero-shot statistical baselines and supervised classifiers. Zero-shot methods, such as Fast-DetectGPT (Bao et al., 2023), theoretically exploit probability curvature under language models but degrade precipitously on short inputs where statistical signals are insufficient, as shown in Figure 1. Lastde (Xu et al., 2025) further advances zero-shot detection by treating token probability sequences as time series and mining both local dynamics and global statistics for improved robustness. In contrast, supervised approaches typically rely on deep neural networks (Guo et al., 2023; Liu et al., 2019) or lightweight classifiers (Verma et al., 2024) trained on labeled data. SeqXGPT (Wang et al., 2023) formulates detection as a signal processing task by mapping token-wise log probabilities into waveforms. However, it explicitly neglects semantic information, which leads to unstable detection performance when token-level signals become sparse. Consequently, both paradigms succumb to the inherent sparsity of short texts (Tulchinskii et al., 2023; Wei et al., 2025), as the limited context prevents both stable statistical convergence and robust feature extraction.

To bridge this gap, we propose RegDetect, a novel framework that shifts the detection paradigm



(a) Mean Representation Distance. (b) t-SNE: Before Training. (c) t-SNE: After Training.

Figure 2: Visualization of the representation space on the `XSum` split using regenerated texts produced by Qwen2.5-7B-Instruct. (a) Average representation distance between original texts and their regenerated views. (b–c) t-SNE projections of the representation space before and after optimization. "Orig." and "Reg." denote original and regenerated texts, respectively. The results illustrate that RegDetect effectively minimizes the regeneration drift for machine-generated texts while amplifying it for human texts, thereby enhancing the separability of the two distributions.

from the static inspection of a text input to representation distance modeling via regeneration-induced drift. Our core insight stems from the intrinsic regeneration bias of large language models: while a regenerated text naturally aligns with machine-generated contexts due to shared generation patterns, it diverges significantly when forced to process human-written text (Yang et al., 2024b). To leverage this asymmetry, we optimize the representation space to minimize the discrepancy between machine-generated texts and their regenerated views, while enlarging the discrepancy for human-written texts and their corresponding regenerations. Specifically, RegDetect maps both the original text and its regenerated view into a shared latent space using a Siamese encoder architecture. We then jointly optimize a binary cross-entropy objective and an asymmetric contrastive loss, which pulls machine-generated text–regeneration pairs closer while pushing human text pairs apart beyond a predefined margin.

Following this optimization, the spatial distinction between human-written and machine-generated texts becomes explicitly observable in the projection space. As demonstrated in Figure 2, machine-generated text-regeneration pairs are tightly clustered due to minimal drift, whereas human pairs exhibit substantial semantic drift and are distinctly pushed far apart beyond the established boundary. The overall workflow of our proposed framework is illustrated in Figure 3.

In summary, our contributions are as follows:

- We propose RegDetect, a novel framework that reformulates short-text detection from static feature extraction to representation drift modeling, utilizing the continuation bias of LLMs as a robust signal amplifier.
- We design an asymmetric contrastive learning objective coupled with a Siamese encoder, reshaping the semantic space by pulling machine text-regeneration pairs closer while pushing human text-regeneration pairs beyond a margin.
- We demonstrate through extensive experiments on the DetectRL (Wu et al., 2024) benchmark that RegDetect achieves state-of-the-art performance. It yields significant gains on challenging 30–50 word short texts while maintaining robust efficacy on longer inputs and strong out-of-distribution generalization.

2 Related Work

Zero-shot Methods. Such methods primarily analyze the probabilistic characteristic of the input text, and then compute statistical scores to determine its origin. These methods leverage the inherent properties of generative large language model. Typically, Likelihood (Solaiman et al., 2019a) employs the generation probability fo each token. Binoculars (Hans et al., 2024) characterizes machine-generated text through a cross-perplexity-based contrast between two similar language models. RepreGuard (Chen et al., 2025) is a representation-based detector that distinguishes human and machine text by identifying discriminative directions in neural activation space using

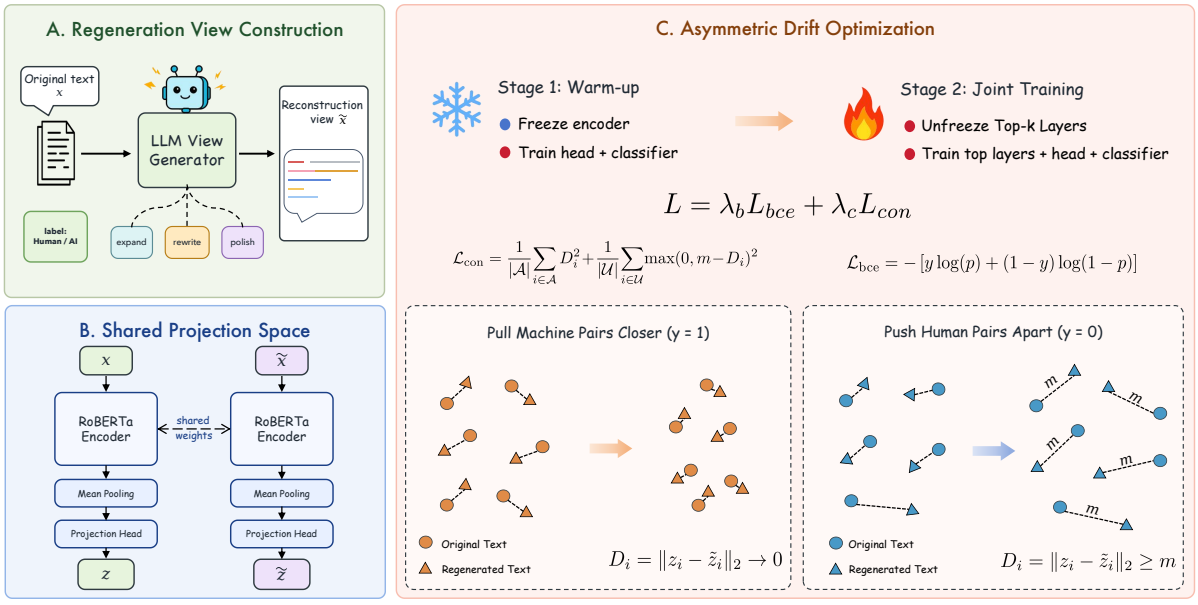


Figure 3: Overview of the RegDetect framework. An input text x and its LLM-regenerated view $\tilde{x}$ are encoded by a Siamese encoder to capture semantic drift. The framework predicts the AI-generation probability using relative drift features, optimized jointly with an asymmetric contrastive loss. This objective explicitly minimizes the reconstruction drift for machine-generated texts while separating human text pairs beyond a semantic margin.

a surrogate language model. Other methods determine the text source by analyzing the differences between the original text and its modified version. DetectGPT (Mitchell et al., 2023) and Fast-DetectGPT (Bao et al., 2023) using the log probability discrepancy between the input text and its token-perturbed counterpart to distinguish human-written text from machine-generated text.

Supervised Methods. This group of methods usually train a text classifier based on a dataset of human-written text and machine-generated text to determine text source. SeqXGPT (Wang et al., 2023) leverages log probability sequences derived from white-box large language models (LLMs) as discriminative features and trains a liner classifier for sentence-level text detection. RAIDAR (Mao et al., 2024) leverages LLM-driven rewriting to extract edit-based features and trains lightweight classifiers for machine-generated text detection. L2D (Zhou et al., 2026) formulates machine-generated text detection as adaptive distance learning by maximizing the discrepancy between reconstruction errors of human-written and LLM-generated text, where the distance metric is learned through lightweight fine-tuning on rewritten text pairs. In addition, some methods distinguish text provenance by leveraging the idea of contrastive learning. DeTeCtive (Guo et al., 2024) proposed a multi-task auxiliary multi-level contrastive learning framework that reframes machine-generated text detection as writing style discrimination.

3 Preliminary

Problem Formulation. Let $\mathcal{X}$ denote the space of text sequences. Consider a text segment $x \in \mathcal{X}$ consisting of a sequence of tokens $x = \{w_1, w_2, \ldots, w_n\}$. Our objective is to learn a binary decision function $f : \mathcal{X} \to \{0, 1\}$ that classifies x as either machine-generated ($y = 1$) or human-written ($y = 0$), given a training corpus $\mathcal{T} = \{(x_i, y_i)\}_{i=1}^{N}$.

The Short-Text Information Bottleneck. Theoretically, the upper bound of detection reliability is strictly governed by the Total Variation distance $d_{\text{TV}}(\mathcal{H}, \mathcal{M})$, where $\mathcal{H}$ and $\mathcal{M}$ represent the underlying human-written and machine-generated submanifolds within $\mathcal{X}$, respectively (Sadasivan et al., 2023). For short sequences, the lack of accumulated stylistic features causes $\mathcal{H}$ and $\mathcal{M}$ to heavily entangle ($d_{\text{TV}} \to 0$), making static representations of human and machine texts virtually indistinguishable. To circumvent this information bottleneck, rather than directly measuring the static properties of the original input x, we leverage a dynamic mechanism to amplify these microscopic manifold discrepancies.

Manifold Disentanglement via Regeneration. While original short texts overlap in the static representation space, a large language model G is highly sensitive to their minute underlying distributional disparities. By utilizing G to generate a regenerated view $\tilde{x}$, the forced generative process exposes the underlying manifold mismatch (Zhou et al., 2026).

Specifically, due to the implicit generation bias of G, a machine-generated text ($y = 1$) naturally aligns with $\mathcal{M}$, yielding minimal representation distance between x and $\tilde{x}$. Conversely, forcing a human-authored text ($y = 0$) onto $\mathcal{M}$ through regeneration induces a substantial semantic drift. Rather than calculating surrogate language model likelihoods as in Zhou et al. (2026), our framework maps both x and $\tilde{x}$ into a shared latent space via a Siamese encoder to model this relative drift explicitly. By optimizing the geometry of this space to satisfy an asymmetric constraint—where the distance $D(x, \tilde{x}) \to 0$ for machine texts and $D(x, \tilde{x}) \geq m$ for human texts (with m denoting a structural margin)—RegDetect effectively separates the overlapping static distributions into a robustly discriminative distance measurement task.

4 RegDetect

4.1 Regeneration View Construction

RegDetect first constructs an LLM-induced view for each input text. Given a short text x, we employ a generator LLM G to produce an regenerated view $\tilde{x}$. In the default setting, we utilize a standardized, task-agnostic prompt template P_τ:

> Please expand the following text while preserving its topic. Return only the expanded text.

The view generation process is defined as:

$$\tilde{x} = G(P_\tau \oplus x), \tag{1}$$

where $\oplus$ denotes prompt–text concatenation. Inspired by Zhou et al. (2026), we also evaluate rewriting and polishing views, while keeping the detector architecture and training objective unchanged. P_τ is the prompt associated with view type τ, and $\tau \in \{\texttt{expand}, \texttt{rewrite}, \texttt{polish}\}$. In our main experiments, $\tau = \texttt{expand}$. Detailed prompt configurations and comparison results are provided in Appendix A.

4.2 Shared Projection Space

Given the original text x and its regenerated view $\tilde{x}$, RegDetect encodes both sequences using a shared RoBERTa (Liu et al., 2019) encoder $f_\theta(\cdot)$. Following standard practice, we obtain sentence-level representations h and $\tilde{h}$ via mean pooling over the token-level hidden states.

To explicitly model regeneration-induced drift, the pooled representations are projected into a shared latent space through a projection head $g_\phi(\cdot)$ and subsequently ℓ_2-normalized:

$$z = \frac{g_\phi(h)}{\|g_\phi(h)\|_2}, \qquad \tilde{z} = \frac{g_\phi(\tilde{h})}{\|g_\phi(\tilde{h})\|_2}. \tag{2}$$

This shared projection space serves as the foundation for modeling the relative discrepancy between the original text and its regenerated view.

4.3 Asymmetric Drift Optimization

For each instance x_i in the training corpus $\mathcal{T}$, RegDetect first generates a regenerated view $\tilde{x}_i$ and maps both texts into the projection space, obtaining z_i and $\tilde{z}_i$. RegDetect characterizes machine-generation signals through the relative drift between the original and regenerated representations. Specifically, we compute the Euclidean distance $D_i = D(x_i, \tilde{x}_i) = \|z_i - \tilde{z}_i\|_2$ and the cosine similarity $S_i = z_i^\top \tilde{z}_i$. We then construct a unified drift representation:

$$r_i = [|z_i - \tilde{z}_i|; z_i \odot \tilde{z}_i; D_i; S_i], \tag{3}$$

where $[\cdot;\cdot]$ denotes vector concatenation and $\odot$ denotes element-wise multiplication. The probability of x_i being machine-generated is predicted by a lightweight two-layer MLP classifier:

$$\hat{y}_i = C_\psi(r_i). \tag{4}$$

The classification objective is standard binary cross-entropy:

$$\mathcal{L}_{\text{bce}} = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log \hat{y}_i + (1 - y_i)\log(1 - \hat{y}_i)\right]. \tag{5}$$

To explicitly shape the geometry of the projection space, we further introduce an asymmetric contrastive objective. Let $\mathcal{A} = \{i : y_i = 1\}$ and $\mathcal{U} = \{i : y_i = 0\}$ denote the index sets of machine-generated and human-written examples, respectively. Machine-generated original–regenerated pairs are treated as positive pairs and are encouraged to exhibit minimal regeneration drift. In contrast, human-written pairs are pushed apart only when their regeneration drift is below

a margin threshold. Therefore, the margin term naturally focuses optimization on difficult human examples: pairs whose drift already exceeds the margin produce zero contrastive loss, while only margin-violating human pairs contribute repulsive gradients.

The asymmetric contrastive loss is defined as:

$$\mathcal{L}_{\text{con}} = \frac{1}{|\mathcal{A}|}\sum_{i\in\mathcal{A}} D_i^2 + \frac{1}{|\mathcal{U}|}\sum_{i\in\mathcal{U}} \max(0, m - D_i)^2, \tag{6}$$

where m is the predefined margin. This objective assigns asymmetric roles to the two classes: machine-generated pairs are pulled together in the projection space, whereas human-written pairs are pushed apart only when their drift falls below the margin.

The overall training objective is:

$$\mathcal{L} = \lambda_{\text{b}}\mathcal{L}_{\text{bce}} + \lambda_{\text{c}}\mathcal{L}_{\text{con}}, \tag{7}$$

where λ_{b} and λ_{c} are balancing coefficients.

We optimize RegDetect in two stages. First, the RoBERTa (Liu et al., 2019) encoder is frozen while only the projection head and classifier are trained. Then, the top k encoder layers ($k = 4$ by default) are unfrozen and jointly fine-tuned together with the projection head and classifier.

5 Experimental Setup

5.1 Datasets

We conduct experiments on the DetectRL (Wu et al., 2024) benchmark, a comprehensive dataset featuring multi-attack settings alongside diverse linguistic styles across four domains (ArXiv (Paul, 2023), XSum (Narayan et al., 2018), Writing Prompts (Fan et al., 2018), Yelp (Zhang et al., 2015)) and texts generated by four representative LLMs (ChatGPT (OpenAI, 2023b), PaLM-2-bison (Anil et al., 2023), Claude-instant (Anthropic, 2023), Llama-2-70b (Touvron et al., 2023)). Crucially, each domain subset uniformly mixes outputs from all four LLMs, and conversely, each LLM subset spans all four domains. Detailed dataset descriptions are provided in Appendix B.

To address signal sparsity, we construct a dedicated short-text subset by extracting contiguous 30-to-50-word segments while preserving sentence boundaries. For length sensitivity analysis, we utilize samples ranging from 20 to 360 words. Every evaluation setting (per domain, generator, or length) employs a balanced subset of 900 instances (450 human-written, 450 machine-generated) to ensure fair comparisons.

5.2 Baselines

We compare RegDetect against a comprehensive set of representative state-of-the-art detection methods, covering both statistics-based and training-based paradigms. Specifically, the baselines include Binoculars (Hans et al., 2024), DetectGPT (Mitchell et al., 2023), Fast-DetectGPT (Bao et al., 2023), FourierGPT (Xu et al., 2024), Log-Likelihood (Solaiman et al., 2019a), Log-Rank (Solaiman et al., 2019b), LRR (Su et al., 2023), RAIDAR (Mao et al., 2024), RepreGuard (Chen et al., 2025), SeqXGPT (Wang et al., 2023) and L2D (Zhou et al., 2026). Detailed descriptions and implementation details are provided in the Appendix C.

5.3 Implementation Details

Model Configurations. We use Qwen2.5-7B-Instruct (Yang et al., 2024a) as the default generator LLM for constructing regeneration views. Additional comparisons with DeepSeek-V4-Flash (DeepSeek-AI, 2026) and Qwen3.5-Plus (Qwen Team, 2026) are reported in Appendix D. Unless otherwise specified, generation is performed with temperature 0.8, top-p 0.9. We use RoBERTa-large (Liu et al., 2019) as the Siamese encoder, followed by a projection head with hidden size 512 and output dimension 128. The classifier is a two-layer MLP with hidden size 128. Further implementation details and hyperparameter settings are provided in Appendix E.

Evaluation Protocol. We randomly partition our dataset into training, validation, and test sets, comprising 64%, 16%, and 20% of the data, respectively. For evaluation, we measure the detection performance using the Area Under the Receiver Operating Characteristic Curve (AUROC) and the F1-score. The decision threshold used for computing F1-score is selected on the validation set and then fixed for evaluation on the test set.

6 Experimental Results

6.1 In-Distribution Classification

Multi-Domain Results. As reported in the left panel of Table 1, RegDetect achieves an outstanding average AUROC of 98.00% and an F1-score of 93.72%, outperforming the strongest baseline,

Table 1: Main results on short-text segments (Length 30–50) across multi-domain and multi-LLM datasets. We report both **AUROC** (AUC, %) and **F1-score** (F1, %). Higher values indicate better performance. The best performance is highlighted in **bold**, and the second best is underlined.

Method	Multi-Domain Datasets								Multi-LLM Datasets							
	ArXiv		Writing		XSum		Yelp		ChatGPT		Claude		PaLM		LLaMA	
	AUC	F1	AUC	F1	AUC	F1	AUC	F1	AUC	F1	AUC	F1	AUC	F1	AUC	F1
Binoculars	74.84	65.67	67.28	49.09	52.90	72.47	80.07	71.39	71.36	59.95	43.13	11.31	82.37	72.72	87.11	78.50
DetectGPT	43.92	7.87	56.64	51.75	45.02	2.19	54.10	57.47	55.05	45.03	51.36	47.85	51.86	45.14	53.28	43.88
Fast-DetectGPT	72.94	64.38	57.25	44.74	52.65	40.93	70.41	59.03	62.98	48.59	37.46	3.90	76.90	67.74	80.51	71.13
FourierGPT	67.35	63.88	56.81	60.56	54.05	47.50	51.72	56.67	56.25	61.69	57.34	49.30	57.55	54.98	56.91	54.27
Log-Likelihood	64.13	51.08	59.09	40.72	47.11	36.75	71.97	63.88	64.14	53.95	50.14	26.95	71.45	62.53	75.20	59.67
Log-Rank	64.76	51.80	57.90	36.01	47.23	28.52	72.95	69.38	62.61	53.24	50.37	23.29	72.08	59.51	76.60	65.71
LRR	64.93	53.07	53.46	35.83	47.38	28.18	72.18	61.70	55.60	44.06	50.94	54.80	70.54	61.60	76.95	66.91
RAIDAR	81.77	71.17	83.14	73.94	67.29	63.35	70.39	62.25	86.37	79.25	76.73	70.73	81.07	73.29	75.69	68.93
RepreGuard	74.00	69.29	79.56	79.60	82.78	81.21	83.33	81.93	88.33	88.52	68.89	64.37	79.89	80.43	65.56	69.31
SeqXGPT	77.01	74.25	61.22	52.98	60.68	55.94	72.60	71.96	67.54	67.71	61.91	62.98	73.25	67.97	80.03	75.39
L2D	99.89	**97.75**	90.09	86.43	87.40	81.82	96.30	91.11	97.81	**96.09**	83.90	79.80	96.64	91.95	93.32	**93.75**
RegDetect (Ours)	**100.00**	97.14	**95.13**	**89.01**	**98.05**	**93.12**	**98.81**	**95.60**	**98.81**	93.18	**93.16**	**84.78**	**97.43**	**91.98**	**97.90**	90.26

L2D (Zhou et al., 2026), by an absolute margin of 4.58% in average AUROC. We also note that other competitive methods, namely RepreGuard (Chen et al., 2025) and RAIDAR (Mao et al., 2024), only manage average AUROCs of 79.92% and 75.65%, respectively. Furthermore, while these baseline methods suffer severe performance degradation in challenging domains—for instance, L2D (Zhou et al., 2026) drops to 87.40% and RAIDAR to 67.29% on the XSum (Narayan et al., 2018) dataset—RegDetect exhibits remarkable robustness, sustaining a near-perfect 98.05% AUROC. The strong performance therefore suggests that the regenerated-view drift provides an additional discriminative signal beyond raw lexical or likelihood-based statistics.

Multi-LLM Results. The right panel of Table 1 extends this evaluation to texts generated by four structurally diverse LLMs. RegDetect achieves the highest average AUROC of 96.83%, improving over L2D (Zhou et al., 2026) by 2.41 percentage points (94.42%). In terms of average F1-score, RegDetect remains highly competitive, reaching 90.05%, only slightly below L2D's 90.40%. This small F1 gap suggests that the two methods are comparable under a fixed operating point, while RegDetect provides stronger overall ranking quality as reflected by AUROC. At the individual-generator level, RegDetect obtains the best AUROC on all four LLM sources. The advantage is especially pronounced on Claude (Anthropic, 2023), where many existing detectors struggle: L2D drops to 83.90% AUROC, whereas RegDetect improves this result to 93.16%. Zero-shot baselines such as Binoculars (Hans et al., 2024) and Fast-DetectGPT (Bao et al., 2023) also experience severe F1 degradation on Claude (11.31% and 3.90%, respectively). These results indicate that RegDetect captures generator-agnostic regeneration-drift signals rather than relying on artifacts specific to a single LLM family.

6.2 Out-of-Distribution Classification

As shown in Figure 4, RegDetect demonstrates more stable performance in various out-of-distribution scenarios across both domain and generator shifts. Transferring from XSum (Narayan et al., 2018) to ArXiv (Paul, 2023) presents a severe domain shift, as ArXiv's abstract-style scientific writing differs markedly from highly condensed news texts. In this setting, L2D (Zhou et al., 2026) and representation-based baseline RepreGuard (Chen et al., 2025) drop to near 50% AUROC, whereas RegDetect maintains an 80.89% AUROC. A similar trend emerges in cross-LLM generalization. Although L2D (Zhou et al., 2026) is highly competitive in many generator shifts, its performance decreases to 78.72% when trained on Claude (Anthropic, 2023) and evaluated on LLaMA (Touvron et al., 2023), with RAIDAR (Mao et al., 2024) similarly dropping to 68.93%. RegDetect, however, yields a higher detection efficacy of 88.89% AUROC under this shift. Compared to these approaches, our mechanism captures a more robust signal, generalizing more effectively across diverse language models and text formats.

Figure 4: Out-of-Distribution AUROC (%) heatmaps comparing RegDetect with RepreGuard, RAIDAR, and L2D. Rows and columns correspond to training and evaluation domains, respectively.

6.3 Robustness to Text Length

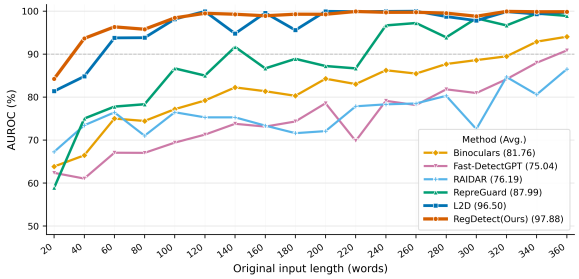


Figure 5: AUROC (%) across different input lengths. RegDetect remains robust from ultra-short texts to longer contexts. We report representative and competitive baselines, including L2D, RepreGuard, Binoculars, Fast-DetectGPT, and RAIDAR. Values in the legend indicate average AUROC across all length intervals.

To assess robustness across input lengths, we evaluate methods from 20 to 360 words, as shown in Figure 5. Overall, most methods exhibit improved AUROC as input length increases, which supports the common observation that short texts contain insufficient statistical and stylistic evidence, whereas longer texts expose richer and more detectable generation patterns. RegDetect achieves the highest average AUROC of 97.89%, outperforming the strongest baseline L2D (Zhou et al., 2026) (96.50%) and RepreGuard (Chen et al., 2025) (87.99%). At 20 words, RegDetect reaches 84.23% AUROC, exceeding L2D (81.37%) and Binoculars (Hans et al., 2024) (63.84%), suggesting that regeneration-based drift modeling is effective even when the original text contains limited detectable signal. Across longer inputs, RegDetect remains near ceiling performance and consistently competitive with L2D, indicating that the learned drift structure does not rely solely on short-text sparsity but also scales to richer contexts. More results and regeneration-length analyses are provided in Appendix G.

Table 2: Performance comparison (AUROC %) under various evasion attacks. The best performance in each setting is highlighted in **bold**, and the second best is underlined.

Attack Type	RepreGuard	L2D	RegDetect (Ours)
Direct Prompt (No Attack)	82.78	95.93	**97.88**
Prompt Attack	82.22	**98.27**	96.38
Paraphrase Attack	82.22	94.30	**96.02**
Perturbation Attack	86.67	67.47	**92.42**

6.4 Robustness to Attacks

As shown in Table 2, RegDetect demonstrates exceptional resilience. When facing prompt and paraphrase attacks, our framework maintains highly competitive detection capabilities, achieving an AUROC of 96.38% and 96.02%, respectively. Detailed attack configurations are provided in Appendix H.

Notably, L2D (Zhou et al., 2026) suffers a severe performance degradation under perturbation attacks, with its AUROC dropping to 67.47%. This vulnerability mainly stems from its reliance on token-level log-probability statistics, which are highly sensitive to perturbation-based attacks. In contrast, RegDetect measures relative semantic

drift in the representation space, making it substantially more robust to surface-level perturbations and providing a more stable detection signal.

6.5 Evaluation on Frontier LLMs

To evaluate the robustness of RegDetect on the most frontier language models, we further construct a new benchmark based on XSum (Narayan et al., 2018), using DeepSeek-V4-Flash (DeepSeek-AI, 2026) as the source generator and Qwen2.5-7B-Instruct (Yang et al., 2024a) as the regenerator. Detailed dataset construction procedures are provided in Appendix I.

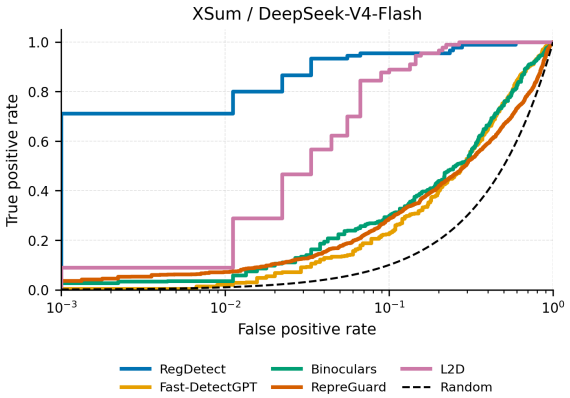


Figure 6: ROC curves on the benchmark under logarithmic FPR scaling. RegDetect consistently dominates competing baselines, particularly in the low-FPR regime that is most critical for real-world deployment. At extremely low false-positive rates ($\mathrm{FPR} \leq 10^{-3}$), RegDetect already achieves strong recall, whereas existing methods remain close to random guessing.

As shown in Figure 6, RegDetect consistently outperforms competing baselines, including Fast-DetectGPT (Bao et al., 2023), Binoculars (Hans et al., 2024), RepreGuard (Chen et al., 2025), and L2D (Zhou et al., 2026). In terms of AUROC, RegDetect achieves 0.979, surpassing the strongest baseline L2D (0.950) while maintaining a clear advantage over other zero-shot detectors.

More importantly, the superiority of RegDetect becomes particularly pronounced in the low-FPR regime, which is critical for practical deployment. At $\mathrm{FPR} \leq 0.001$, RegDetect already achieves a TPR of 0.711, whereas all competing baselines remain below 0.10. Similarly, RegDetect reaches TPR values of 0.933 and 0.956 at $\mathrm{FPR} \leq 0.05$ and $\mathrm{FPR} \leq 0.10$, respectively, consistently exceeding all baselines by a substantial margin.

Overall, these results demonstrate that RegDetect maintains strong discriminative capability even against frontier LLMs, highlighting the robustness and transferability of regeneration-based drift modeling.

7 Ablation Study

Table 3 presents the ablation results. Replacing RegDetect with a standard RoBERTa-only classifier leads to substantial performance degradation, particularly under distribution shifts, highlighting the generalization capability of regeneration-based drift modeling. Removing the regeneration view also causes a clear drop in both cross-domain and cross-LLM performance, confirming that the regeneration view provides critical discriminative signals for short-text detection. In addition, removing the asymmetric contrastive objective consistently weakens generalization ability, demonstrating that contrastive space regularization further improves the robustness of the learned representation space. Overall, the results verify that both regeneration-induced drift and asymmetric contrastive learning are essential to RegDetect.

Table 3: Ablation results measured by AUROC (%). For in-distribution settings, we report the average over four multi-domain and four multi-LLM datasets. For out-of-distribution, we average over all directed cross-domain and cross-LLM transfer pairs. The *w/o regeneration* variant replaces the regenerated view with the text under detection itself, while *w/o contrastive loss* keeps the regeneration view but removes the contrastive objective.

Variant	In-distribution		Out-of-distribution	
	Multi-domain	Multi-LLM	Cross-domain	Cross-LLM
Full RegDetect	**98.00**	**96.83**	**76.08**	**92.59**
w/o regeneration	95.73	91.56	70.52	88.11
w/o contrastive loss	97.09	95.62	73.69	91.07
RoBERTa-only	94.10	91.23	72.42	83.10

8 Conclusion

We introduce RegDetect, a framework for detecting short machine-generated text through regeneration-induced drift modeling. RegDetect leverages LLM regeneration to amplify weak short-text signals and optimizes an asymmetric contrastive objective to shape label-dependent representation drift between inputs and their regenerated views. Experiments on the DetectRL benchmark show that RegDetect performs strongly in the challenging 30–50 word regime, remains effective on longer inputs, and generalizes across domains and generators. These results suggest that regeneration-induced drift modeling is a promising direction for robust short-text detection.

Limitations

Despite the strong empirical performance of RegDetect, our evaluation is currently restricted to English benchmarks, and the effectiveness of regeneration-induced drift modeling in multilingual settings remains to be systematically validated. Although the underlying principles of semantic drift and regeneration consistency are language-agnostic, linguistic differences across languages may influence the behavior of regeneration-based detectors.

Ethical considerations

RegDetect is developed to improve the reliability of machine-generated text detection, however, we acknowledge the inherent limitations of automatic detection systems. Although RegDetect achieves strong empirical performance, no detector can guarantee perfect accuracy across all domains, writing styles, or future generation models. False positives remain possible, especially for highly polished human-written text or content that closely resembles machine-generated distributions. Therefore, outputs produced by RegDetect should not be treated as definitive evidence of misconduct or policy violation without additional human verification and contextual analysis. We hope this work contributes toward more robust detection methods while encouraging careful human oversight, fairness, and responsible use of AI-assisted content analysis tools.

References

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Mérouane Debbah, Étienne Goffinet, Daniel Hesslow, Julien Launay, Quentin Malartic, and 1 others. 2023. The falcon series of open language models. *arXiv preprint arXiv:2311.16867*.

Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, and 1 others. 2023. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*.

Anthropic. 2023. Releasing Claude Instant 1.2. https://www.anthropic.com/news/releasingclaude-instant-1-2.

Guangsheng Bao, Yanbin Zhao, Zhiyang Teng, Linyi Yang, and Yue Zhang. 2023. Fast-detectgpt: Efficient zero-shot detection of machine-generated text via conditional probability curvature. In *The Twelfth International Conference on Learning Representations*.

Sid Black, Leo Gao, Phil Wang, Connor Leahy, and Stella Biderman. 2021. GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh-Tensorflow.

Bernhard E. Boser, Isabelle Guyon, and Vladimir Vapnik. 1992. A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual ACM Conference on Computational Learning Theory, COLT 1992, Pittsburgh, PA, USA, July 27-29, 1992*, pages 144–152. ACM.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, and 1 others. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*.

Xin Chen, Junchao Wu, Shu Yang, Runzhe Zhan, Zeyu Wu, Ziyang Luo, Di Wang, Min Yang, Lidia S Chao, and Derek F Wong. 2025. Repreguard: Detecting llm-generated text by revealing hidden representation patterns. *Transactions of the Association for Computational Linguistics*, 13:1812–1831.

DeepSeek-AI. 2026. Deepseek-v4: Towards highly efficient million-token context intelligence.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898, Melbourne, Australia. Association for Computational Linguistics.

Ji Gao, Jack Lanchantin, Mary Lou Soffa, and Yanjun Qi. 2018. Black-box generation of adversarial text sequences to evade deep learning classifiers. In *2018 IEEE security and privacy workshops (SPW)*, pages 50–56. IEEE.

Josh A Goldstein. 2023. Generative language models and automated influence operations: Emerging threats and potential mitigations. *arXiv preprint arXiv:2301.04246*.

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, Yongxin He, Shan Zhang, 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.

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. *Preprint*, arXiv:2401.12070.

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

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2020. Is bert really robust? a strong baseline for natural language attack on text classification and entailment. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 8018–8025.

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.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

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*.

Muneer M Alshater. 2022. Exploring the role of artificial intelligence in enhancing academic performance: A case study of chatgpt. *Available at SSRN 4312358*.

Chengzhi Mao, Carl Vondrick, Hao Wang, and Junfeng Yang. 2024. Raidar: generative ai detection via rewriting. *arXiv preprint arXiv:2401.12970*.

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.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, Brussels, Belgium. Association for Computational Linguistics.

OpenAI. 2023a. GPT-3.5 Turbo (version 0613). https://openai.com/blog/function-calling-and-other-api-updates.

OpenAI. 2023b. Introducing chatgpt.

Sayak Paul. 2023. ArXiv Paper Abstracts. Kaggle Dataset.

Qwen Team. 2026. Qwen3.5: Towards native multimodal agents.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, and 1 others. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.

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

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. 2019a. Release strategies and the social impacts of language models. *arXiv preprint arXiv:1908.09203*.

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. 2019b. Release strategies and the social impacts of language models. *arXiv preprint arXiv:1908.09203*.

Jinyan Su, Terry Zhuo, Di Wang, and Preslav Nakov. 2023. Detectllm: Leveraging log rank information for zero-shot detection of machine-generated text. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 12395–12412.

Teo Susnjak. 2022. Chatgpt: The end of online exam integrity? *CoRR*, abs/2212.09292.

Gemma Team, Morgane Riviere, Shreya Pathak, and 1 others. 2024. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Eduard Tulchinskii, Kristian Kuznetsov, Laida Kushnareva, Daniil Cherniavskii, Sergey Nikolenko, Evgeny Burnaev, Serguei Barannikov, and Irina Piontkovskaya. 2023. Intrinsic dimension estimation for robust detection of ai-generated texts. *Advances in Neural Information Processing Systems*, 36:39257–39276.

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, Mexico City, Mexico. Association for Computational Linguistics.

Ben Wang and Aran Komatsuzaki. 2021. Gpt-j-6b: A 6 billion parameter autoregressive language model.

Pengyu Wang, Linyang Li, Ke Ren, Botian Jiang, Dong Zhang, and Xipeng Qiu. 2023. SeqXGPT: Sentence-level AI-generated text detection. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1144–1156, Singapore. Association for Computational Linguistics.

Dongjun Wei, Minjia Mao, Xiao Fang, and Michael Chau. 2025. Short-phd: Detecting short llm-generated text with topological data analysis after off-topic content insertion. *arXiv preprint arXiv:2504.02873*.

Junchao Wu, Runzhe Zhan, Derek F Wong, Shu Yang, Xinyi Yang, Yulin Yuan, and Lidia S Chao. 2024. Detectrl: Benchmarking llm-generated text detection in real-world scenarios. *arXiv preprint arXiv:2410.23746*.

Yang Xu, Yu Wang, Hao An, Zhichen Liu, and Yongyuan Li. 2024. Detecting subtle differences between human and model languages using spectrum of relative likelihood. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 10108–10121, Miami, Florida, USA. Association for Computational Linguistics.

Yihuai Xu, Yongwei Wang, Yifei Bi, Huangsen Cao, Zhouhan Lin, Yu Zhao, and Fei Wu. 2025. Training-free LLM-generated text detection by mining token probability sequences.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, and 22 others. 2024a. Qwen2.5 technical report. *CoRR*, abs/2412.15115.

Xianjun Yang, Wei Cheng, Yue Wu, Linda Ruth Petzold, William Yang Wang, and Haifeng Chen. 2024b. DNA-GPT: divergent n-gram analysis for training-free detection of gpt-generated text. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28.

Hongyi Zhou, Jin Zhu, Kai Ye, Ying Yang, Erhan Xu, and Chengchun Shi. 2026. Learn-to-distance: Distance learning for detecting llm-generated text. In *The Fourteenth International Conference on Learning Representations*.

Jiawei Zhou, Yixuan Zhang, Qianni Luo, Andrea G Parker, and Munmun De Choudhury. 2023. Synthetic lies: Understanding ai-generated misinformation and evaluating algorithmic and human solutions. In *Proceedings of the 2023 CHI conference on human factors in computing systems*, pages 1–20.

A Detailed Analysis of View Strategies

Prompt Configurations. In addition to the `expand` prompt introduced in Section 4.1, we evaluate our framework using `rewrite` and `polish` instructions. For all view generation processes, we utilize Qwen2.5-7B-Instruct (Yang et al., 2024a) as the generator LLM. To ensure a fair comparison, all hyperparameters are set to their default values and kept strictly identical across strategies (see Section 5.3 and Appendix E). Only the specific prompt instructions are varied:

> **Rewrite Prompt:** Please rewrite the following text while preserving its topic. Return only the rewritten text.
> **Polish Prompt:** Please polish the following text while preserving its topic. Return only the polished text.

Performance and Text Length Comparison. Table 4 summarizes the detection performance and average sequence lengths across the three view strategies. Overall, all three transformations yield highly effective detection signals, with mean AUROCs consistently exceeding 97%. The performance gaps between the strategies are marginal, demonstrating the general robustness of representation drift modeling regardless of the specific generation instruction used.

View Strategy	Mean AUROC	Orig Len.	View Len.
`expand` (Default)	**97.42**	39.87	43.13
`polish`	97.41	39.87	36.26
`rewrite`	97.26	39.87	36.11

Table 4: Overall performance (AUROC, %) and sequence length comparison of different view generation strategies.

To provide a more granular perspective, Figure 7 illustrates the detailed performance across the `multi_domain` and `multi_llm` subsets.

B Datasets Details

The DetectRL (Wu et al., 2024) benchmark is constructed using human-written texts from four distinct domains prone to LLM misuse, alongside machine-generated texts produced by four representative Large Language Models. Below we provide detailed descriptions of the specific data sources and models employed.

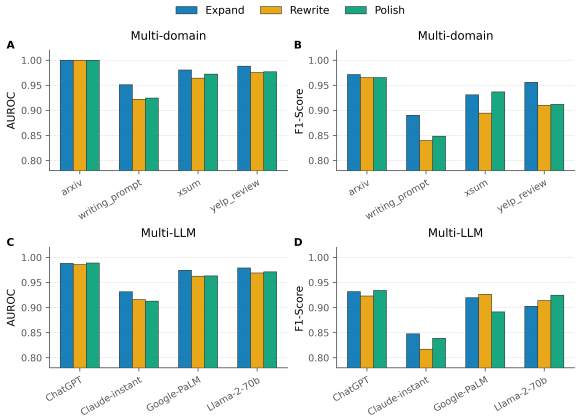


Figure 7: Detailed performance (AUROC and F1-score in %) of `expand`, `rewrite`, and `polish` views across in-distribution datasets.

B.1 Human-Written Datasets

- **ArXiv Abstracts (Paul, 2023):** This domain is represented by the ArXivPapers dataset, a collection of machine learning-related papers published on arXiv.org.
- **XSum (Narayan et al., 2018):** Sourced from BBC reports published between 2010 and 2017, this collection covers a broad spectrum of topics including politics, sports, technology, science, and entertainment.
- **Writing Prompts (Fan et al., 2018):** Focused on hierarchical story generation, it consists of human-written stories paired with unique writing prompts from an online community, capturing the nuances of narrative structure and creative expression.
- **Yelp Reviews (Zhang et al., 2015):** For informal and opinionated text, the Yelp Reviews Polarity dataset is employed. This dataset categorizes reviews into negative (1-2 stars) and positive (3-4 stars) classes, offering a balanced view of user-generated sentiment.

B.2 Generative Models

- **ChatGPT (OpenAI, 2023b):** A conversational language model developed by OpenAI. ChatGPT is widely deployed for dialogue, content generation, and general-purpose language understanding tasks.
- **Claude-instant (Anthropic, 2023):** A lightweight conversational model developed by Anthropic and released as part of the Claude family.

- **PaLM-2-bison (Anil et al., 2023)**: A language model released by Google as part of the PaLM 2 family.

- **LLaMA-2-70B (Touvron et al., 2023)**: An open-source large language model released by Meta as part of the LLaMA 2 series.

C Baselines Details

- **Binoculars (Hans et al., 2024)**: A zero-shot detector that compares perplexity and cross-perplexity computed by a pair of LLMs to measure consistency between an observer and a performer model. We adopt the default configuration, employing Falcon-7B and Falcon-7B-Instruct (Almazrouei et al., 2023) as the underlying models.

- **DetectGPT (Mitchell et al., 2023)**: A perturbation-based method that detects curvature in the likelihood landscape by measuring likelihood discrepancies under random input perturbations. We use T5-small (Raffel et al., 2020) as the perturbation model with 50 perturbations per sample.

- **Fast-DetectGPT (Bao et al., 2023)**: An accelerated variant of DetectGPT (Mitchell et al., 2023) that replaces explicit perturbations with conditional probability sampling for efficiency. We follow the authors' recommended setup, using GPT-Neo-2.7B (Black et al., 2021) as the scoring model and GPT-J-6B (Wang and Komatsuzaki, 2021) as the reference model.

- **FourierGPT (Xu et al., 2024)**: A detection method that transforms relative likelihood sequences estimated by GPT-2-XL (Radford et al., 2019) into frequency-domain spectra via Fourier transform, utilizing an SVM (Boser et al., 1992) supervised classifier to capture dynamic linguistic patterns for identifying machine-generated text.

- **Log-Likelihood (Solaiman et al., 2019a)**: A likelihood-based baseline that uses the average token-level log-probability of the input text as the detection score. We utilize GPT-Neo-2.7B (Black et al., 2021) as the scoring model.

- **Log-Rank (Solaiman et al., 2019b)**: A rank-based metric that computes the average log-rank of tokens according to their predicted probabilities. We employ GPT-Neo-2.7B (Black et al., 2021) to estimate the token rankings.

- **LRR (Su et al., 2023)**: The Log-Likelihood Log-Rank Ratio, which combines log-likelihood and log-rank into a normalized score to improve detection sensitivity. Both components are computed using GPT-Neo-2.7B (Black et al., 2021).

- **RAIDAR (Mao et al., 2024)**: A rewriting-based detection framework that employs gpt-3.5-turbo-0613(OpenAI, 2023a) to rewrite the input text and classifies the source based on the invariance metric, leveraging the intrinsic bias of LLMs to make fewer modifications to machine-generated text compared to human-written text.

- **RepreGuard (Chen et al., 2025)**: A statistics-based detector that employs a surrogate model to capture hidden representation patterns, classifying text based on the projection score of its neural activations along a principal component direction that differentiates human from machine writing. We use Llama-3.1-8B (Dubey et al., 2024) as the surrogate model with 256 training shots.

- **SeqXGPT (Wang et al., 2023)**: A sentence-level detection method that treats log probability lists as temporal wave-like features, utilizing a convolutional network followed by self-attention layers to capture local and global dependencies for fine-grained text provenance analysis.

- **L2D (Zhou et al., 2026)**: A rewriting-based detector that learns a scoring model from paired human-written and machine-generated texts. For each input text, L2D generates multiple rewrites with an instruction-tuned LLM and compares the likelihood of the original text against its rewritten variants under a LoRA-adapted (Hu et al., 2022) surrogate language model. In our experiments, we use Gemma-2-9B-IT (Team et al., 2024) as both the rewriting model and the scoring model, and set the number of rewrites to $K = 2$.

D Impact of Different Regenerator Models

In our main experiments, we utilize Qwen2.5-7B-Instruct (Yang et al., 2024a) as the default generator LLM. To ensure that our regeneration-induced drift mechanism is not overly reliant on the specific characteristics of a single model, we evaluate the framework using two additional generators: DeepSeek-V4-Flash (DeepSeek-AI, 2026) and Qwen3.5-Plus (Qwen Team, 2026). Figure 8 illustrates the AUROC performance of these models,these results confirm that RegDetect is model-agnostic, demonstrating exceptional universality and reliability.

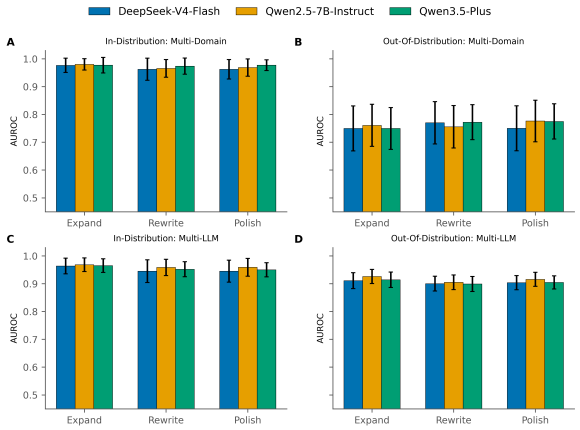


Figure 8: AUROC (%) performance comparison of different generator models and view strategies. Bars report the mean AUROC averaged over four evaluation categories in each setting: four domains for Multi-Domain experiments and four source LLMs for Multi-LLM experiments. Results are shown for in-distribution and out-of-distribution evaluation under fixed validation-selected thresholds.

In-Distribution Performance. As shown in Panels A and C of Figure 8, all three models exhibit exceptionally strong and stable detection capabilities in the in-distribution scenarios. Regardless of the chosen view strategy (`expand`, `rewrite`, or `polish`), the mean AUROC consistently exceeds 94%.

Out-of-Distribution Robustness. While cross-domain transfer (Panel B) inevitably introduces performance degradation, all three generators maintain comparable resilience, scoring tightly within the 75%–77% AUROC range. In the challenging OOD multi-LLM setting (Panel D), the models retain high efficacy, consistently exceeding 90% AUROC.

E Hyperparameters and Implementation Details

For the detector, we use RoBERTa-large (Liu et al., 2019) as a Siamese encoder and apply attention-mask mean pooling over the final hidden states. The projection head is implemented as

$$\text{Linear}(1024, 512) \rightarrow \text{LayerNorm} \rightarrow \text{GELU} \rightarrow \text{Dropout}(0.1) \rightarrow \text{Linear}(512, 128), \quad (8)$$

followed by ℓ_2 normalization. The classifier applies a two-layer MLP with hidden size 128 and dropout 0.1.

Training uses a two-stage schedule. In the first stage, the RoBERTa (Liu et al., 2019) encoder is frozen and only the projection head and classifier are trained for 3 epochs with learning rate 1×10^{-3}. In the second stage, the top 4 RoBERTa layers are unfrozen and trained for 5 epochs, using learning rate 2×10^{-5} for the encoder and 1×10^{-4} for the projection head and classifier. We use AdamW with weight decay 0.01, cosine learning-rate decay, warmup ratio 0.06, and early stopping patience 2.

The optimization objective combines the asymmetric contrastive loss and binary cross-entropy loss with equal weights. The contrastive margin is set to 1.0. The main experiments were conducted on a single NVIDIA Tesla V100 GPU with 32GB memory.

Temp (T)	AUROC	F1	Acc	Mean Len.
0.4	**98.74**	93.12	92.78	117.74
0.6	98.44	**93.68**	**93.33**	117.64
0.8	98.52	93.12	92.78	117.32
1.0	98.70	**93.68**	**93.33**	116.89
1.2	97.72	91.66	91.16	117.33

Table 5: Ablation study on sampling temperature using DeepSeek-V4-Flash on the XSum dataset (AUROC, F1 and Acc in %). All length metrics denote the number of whitespace-split words in the generated expansion views.

F Impact of Sampling Temperature

To further investigate the influence of the expander model and its sampling hyperparameters, we conduct an ablation study using DeepSeek-V4-Flash (DeepSeek-AI, 2026) under varying temperature settings. We specifically select the XSum (Narayan et al., 2018) dataset for this analysis.

Table 5 summarizes the detection performance and view lengths across different temperature (T) values ranging from 0.4 to 1.2. Overall, RegDetect demonstrates remarkable robustness to temperature variations. Moderate temperatures yield consistently strong results, with $T = 0.4$ achieving the highest AUROC of 98.74%, closely followed by $T = 1.0$ (98.70%). The standard default setting of $T = 0.8$ remains highly competitive with a 98.52% AUROC. Across all settings, the average expanded text length remains strictly bounded between $\sim$117 words. These results suggest that RegDetect is relatively insensitive to moderate temperature shifts.

G Impact of Regeneration Length

In this section, we investigate the relationship between regeneration length and detection performance using the `expand` regeneration strategy.

G.1 Natural Expansion Behavior

In our main experiments, the generator LLM (Qwen2.5-7B-Instruct (Yang et al., 2024a)) is prompted to continue the text without explicit length constraints. We analyze the distribution of generated lengths across different datasets and source models (refer to Figure 9).

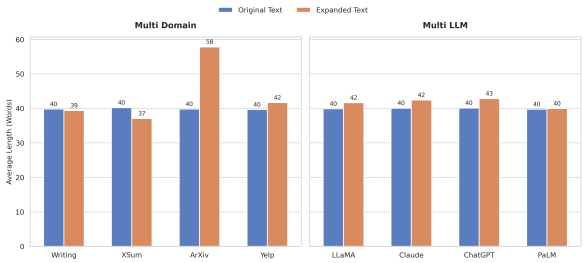


Figure 9: Comparison of average lengths between original and expanded texts. The blue bars represent the original text length (average at 40 words), and the orange bars represent the length of the text expanded by LLM.

Source-Dependent Expansion Length A notable observation from Figure 10 is that, under identical prompting and decoding settings, the generator LLM consistently produces longer continuations for machine-generated inputs than for human-written inputs. In contrast, expansions conditioned on human-written text tend to terminate earlier. This discrepancy suggests that machine-generated text provides a more compatible continuation context for the LLM, enabling sustained generation, whereas human-written text introduces stylistic or distributional mismatches that increase termination likelihood during decoding.

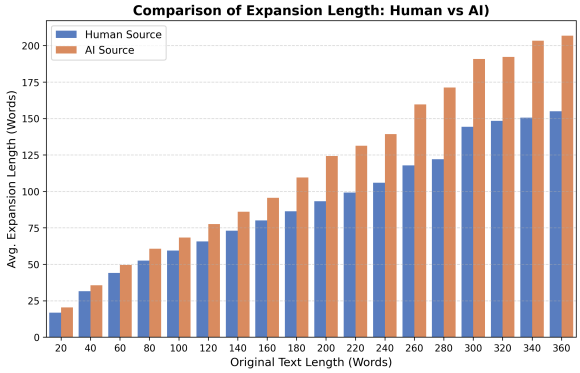


Figure 10: Comparison of average expansion length between Human and AI sources. The chart shows that AI-sourced texts (orange) consistently elicit longer continuations compared to human-sourced texts (blue) across different original text lengths.

G.2 Controlled Expansion Analysis

To systematically evaluate the sensitivity of RegDetect to expansion length, we modified the standardized prompt template used in the main methodology to explicitly control the generation scale.

Prompt Design. Let N denote the target number of words for the expansion. We modify the instruction by appending an explicit length specification, denoted as `{target_words}`:

> Please continue the text based on the context provided below. Expand by **{target_words}** words.

Experimental Setup and Overall Robustness. We evaluate detection performance over a grid of original input lengths centered at $[40, 100, 160, 220, 280, 340]$ words, paired with target expansion lengths varying across $[50, 60, 100, 180, 250, 300, 350, 390, 400]$ words. Due to the inherent variability in LLM instruction following, the actual generated continuations exhibit slight deviations from the exact specified counts; nevertheless, the relative scales are strictly preserved. Figure 11 illustrates how detection performance varies jointly with the original input length and the expansion length.

Short Original Inputs. When the original text is extremely short (e.g., target 40 words, observed mean $\sim$30 words), the intrinsic stylistic signals are highly sparse. In this regime, limiting the expansion to a short continuation (target 50 words) yields

our lowest recorded AUROC of 87.30%. However, increasing the expansion length leads to a substantial recovery in detection performance. By forcing the generator to produce a much longer continuation, the AUROC climbs to a highly competitive 94.18%.

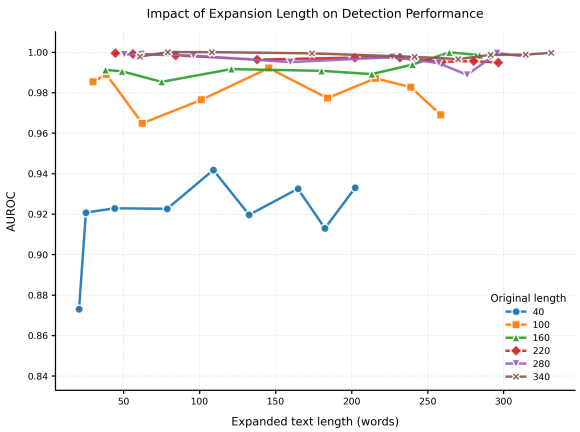


Figure 11: Impact of expansion length on detection performance across varying original input lengths. Increasing the expansion length significantly improves AUROC (%) for shorter original inputs, whereas longer original inputs maintain near-perfect performance even with brief continuations.

Long Original Inputs. For longer original texts ($\geq$ 160 words), detection performance remains near-perfect regardless of the expansion length. For instance, at an original length of 340 words, even a minimal target expansion of 60 words (observed $\sim$79 words) achieves a flawless 100.00% AUROC. This behavior suggests that once the original input is sufficiently informative, generating a massive continuation is unnecessary; a brief expansion is entirely sufficient to trigger and verify the representation drift between human and machine distributions.

H Attack Details

We conduct experiments on the DetectRL Task 1 MultiAttack benchmark (Wu et al., 2024). Following the main experimental protocol, we filter the dataset to the short-text regime (30–50 words) while preserving the original attack settings provided by DetectRL.

These attacks are grouped into three categories:

- **Prompt Attacks.** These attacks utilize carefully designed prompts to guide LLMs toward generating outputs that more closely resemble human-written text. Following DetectRL, we evaluate against Few-Shot prompting (Brown et al., 2020) and ICO prompting (Lu et al., 2023).
- **Paraphrase Attacks.** These attacks rewrite machine-generated text while preserving its semantic content. The benchmark includes several paraphrasing strategies provided by DetectRL, including DIPPER paraphrasing (Krishna et al., 2023), back-translation, and LLM-based polishing.
- **Perturbation Attacks.** These attacks introduce adversarial perturbations at the word, character, or sentence level to mimic post-editing noise and common writing errors. Following DetectRL, the benchmark includes perturbation methods such as TextFooler (Jin et al., 2020), DeepWordBug (Gao et al., 2018).

I XSum–DeepSeek Benchmark

The benchmark is based on XSum (Narayan et al., 2018) and contains 900 short news-style texts in total, including 450 human-written XSum texts and 450 machine-generated texts produced by DeepSeek-V4-Flash (DeepSeek-AI, 2026). Given an XSum (Narayan et al., 2018) news context or topic cue, the model is instructed to generate a concise news-style passage while returning only the generated text. The prompt template is:

> Write a concise news-style passage based on the following news context. Keep the output fluent, factual in style, and approximately 30–50 words. Return only the generated passage.

We then apply post-processing to remove empty responses and malformed outputs, all texts are filtered to the short-input regime, with a target length range of 30–50 words. After filtering, the average original text length is 37.76 words.