DisGAR: Machine Generated Text Detection via Disentangled Generative Artifact Representation

Anonymous ACL submission

Abstract

The widespread misuse of large language models (LLMs) has made machine-generated text detection (MGTD) a critical task. Existing detection methods are often vulnerable to stylistic noise, which limits their generalization in real-world scenarios and robustness on short texts. Furthermore, many methods generate multiple auxiliary texts through LLM-based rewriting or similar strategies, substantially increasing detection latency and limiting their practicality. To address these challenges, we propose the **DisGAR** detection framework, which strips stylistic noise from the input text to mitigate its impact on detection performance, thereby improving cross-domain and cross-model robustness. Specifically, we first fine-tune the attention layers of the proxy model to capture generation-specific patterns. Then we further introduce a global artifact aggregation mechanism and a contrastive learning strategy based on rewritten texts, enabling the separation of stylistic artifacts from the underlying generation patterns. Experimental results demonstrate that DisGAR outperforms existing state-of-the-art methods in unseen-model and unseen-domain detection (with a +10.65% improvement in average balanced accuracy). Moreover, DisGAR achieves detection efficiency comparable to Fast-DetectGPT, successfully balancing detection performance and efficiency. Source code is available at https://anonymous.4open.science/r/DisGAR-7C00

1 Introduction

The landscape of LLMs is characterized by rapid technological iterations and continuous capability breakthroughs. Recent models, such as DeepSeek (Bi et al., 2024), Gemini (Comanici et al., 2025) and Gemini (Comanici et al., 2025), have exhibited exceptional generative performance. Their advancements significantly enhance productivity across various domains, including academic research, journalism, and software development.

However, LLMs pose significant threats to information authenticity and public trust. On the one hand, the inherent flaws of LLMs, including biases (Tjuatja et al., 2024) and hallucinations (Ji et al., 2023), compromise the reliability and trustworthiness of their outputs. On the other hand, malicious misuse of LLMs, such as generating disinformation (Pagnoni et al., 2022), crafting misleading reviews (Adelani et al., 2020), and enabling academic misconduct (Lee et al., 2023), further pose additional security risks. Therefore, developing effective machine-generated detectors is crucial.

In practical applications, detectors not only face a wide variety of machine-generated texts (MGTs) and human-written texts (HWTs), but also handle complex human-machine collaborations, such as *Polish* and *Rewrite*. In addition, detectors also need to detect large volumes of text. Therefore, an effective MGTD method requires detectors to possess high generalization, robustness, and efficiency.

Rewritten or auxiliary texts provide detectors with multi-dimensional comparative signals, which can improve robustness and detection accuracy. Motivated by this advantage, recent studies have increasingly explored auxiliary texts to facilitate machine-generated text detection. Therefore, we categorize existing MGTD methods into two groups.

One line of methods is designed based on probability distributions. These methods (Bao et al., 2024; Xu et al., 2025; Fu et al., 2025) employ a proxy model to compute the token-level probability distributions of the text, leveraging specific statistical features derived from these distributions for detection. However, these methods lack sufficient generalization capability when confronting texts generated by different LLMs and multi-domain texts. During the text generation process, both humans and LLMs are subject to the dual constraints of prior knowledge and specific task instructions.

These inherent constraints directly result in specific stylistic formulations. When reconstructing texts from diverse sources, the proxy model often struggles to eliminate this stylistic heterogeneity within a unified framework, thereby yielding varying degrees of reconstruction bias.

Another line of work comprises methods based on perturbation and comparison. These methods (Mitchell et al., 2023; Mao et al., 2024; Zhou et al., 2026)measure the relationship between the target text and generated auxiliary texts, such as rewritten texts. While these methods improve the generalization ability of the detectors to a certain extent, they require multiple inferences from LLMs, leading to substantial resource consumption and significant time overhead (Sun et al., 2026). Furthermore, both lines of methods lack sufficient robustness in short-text detection: the former exhibits high uncertainty when predicting initial tokens due to the absence of prompts, whereas the latter is constrained by the typically high similarity between the short text and its generated auxiliary text.

To address these limitations, we propose the DisGAR detection framework, which utilizes a proxy model to extract the hidden representations of the text. Instead of removing stylistic information directly, we apply dimensionality reduction mapping to strip out stylistic noise that may introduce reconstruction bias. This process yields representations that are more effective for MGTD, which we refer to as generative artifact representations (GARs). To guide this mapping toward effectively stripping out stylistic noise while preserving generation-related artifacts, we introduce multi-dimensional training constraints.

First, we exploit the fact that randomly truncated prefixes retain the same labels as their full texts, guiding the extractor to identify and filter out local stylistic noise. Second, we introduce a contrastive learning strategy to further decouple stylistic noise from GARs. This is based on the observation that rewritten and target texts exhibit similar stylistic noise but distinct GARs, whereas texts within the same category show distinct stylistic noise but share similar GARs. Furthermore, we fine-tune the proxy model specifically for the MGTD task, ensuring that its hidden representations are inherently enriched with GARs.

Experimental results show that DisGAR significantly outperforms the current strongest baseline method L2D(Zhou et al., 2026), while maintaining a detection efficiency comparable to Fast-DetectGPT(Bao et al., 2024). Specifically, DisGAR improves the average balanced accuracy in in-domain tests by approximately 3.2% (with an average relative gain of about 69.53%), and increases the average balanced accuracy in cross-model and cross-domain tests by approximately 10.56%. Furthermore, the framework exhibits remarkable robustness in short-text detection.

We summarize our contributions as follows:

- We propose DisGAR, an effective and efficient machine-generated text detection framework that learns generative artifact representations (GARs) from the hidden representations of a proxy language model, aiming to disentangle generation-specific artifacts from stylistic noise.
- We design two complementary mechanisms, global artifact aggregation and supervised contrastive learning based on rewritten text, to enable the detector to capture generation patterns and improve robustness across diverse text distributions.
- Experiments show that DisGAR achieves state-of-the-art performance in cross-domain and cross-model detection, while maintaining competitive efficiency and strong robustness on short texts, multilingual texts, and memorized human-written texts.

2 Related Work

Current MGTD works focus on two primary lines of work.

Methods based on probability distributions. These methods primarily rely on the fact that HWT exhibits a higher degree of uncertainty compared to MGT. Early methods use simple statistics such as entropy, likelihood, or ranking (Gehrmann et al., 2019). Fast-DetectGPT (Bao et al., 2024) compares the conditional probability curvature between original and sampled texts for efficient detection, and Lastde (Xu et al., 2025) models token probability sequences via time-series analysis to capture temporal dynamics. More recently, the focus has shifted toward adjusting a proxy model to better evaluate the uncertainty of human-written versus machine-generated text (Chen et al., 2025; Fu et al., 2025). However, these methods are vulnerable to stylistic noise, which significantly limits their generalizability.

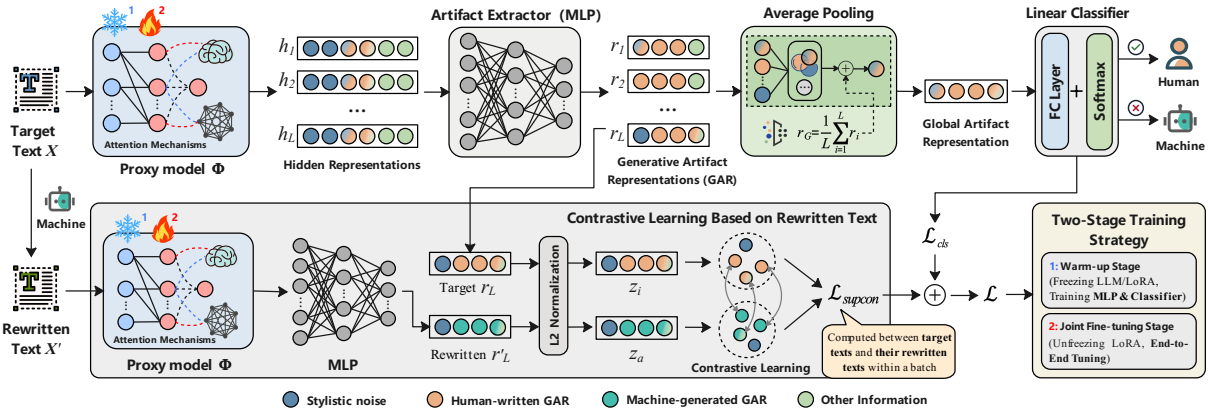

Figure 1: Overview of our proposed detection framework **DisGAR**. First, the proxy model extracts the hidden representations of all truncated prefixes from the target text. These representations are then processed by an artifact extractor to generate generative artifact representations, which are subsequently aggregated for final classification. During training, we adopt rewritten texts to compute a contrastive loss, which is jointly optimized with the classification loss to guide the DisGAR's learning via a two-stage training strategy.

Methods based on perturbation and comparison. These methods primarily rely on the fact that after LLM processing, the discrepancy between the input and output is significantly smaller for MGT than for HWT. DetectGPT (Mitchell et al., 2023) provides a novel perspective for MGTD, distinguishing between MGT and HWT through perturbation, scoring, and probability curvature. DNA-GPT (Yang et al., 2024) truncates the text and regenerates suffixes for comparison. RAIDAR (Mao et al., 2024) generates multiple rewritten versions of entire paragraph and measures consistency via the edit distance between versions. L2D (Zhou et al., 2026) adaptively learns the distance between the target text and the rewritten text. However, these methods require at least one inference from LLM, which incurs significant time overhead. Furthermore, their effectiveness is highly constrained by the quality of the model's processing. Particularly when handling short texts, the input-output discrepancy is typically extremely minimal, rendering these methods insufficiently robust in such scenarios.

3 Proposed Method

We introduce the proposed DisGAR detection framework in this section. As illustrated in Figure 1, DisGAR comprises three core components: generative artifact representation extraction, global artifact aggregation, and contrastive learning based on rewritten text.

3.1 Motivation

Most existing MGTD methods are predicated on a widely accepted consensus that *HWT exhibits greater uncertainty than MGT*. In real-world applications, this means that when utilizing a proxy model to estimate token-level probabilities, HWT typically exhibits lower likelihoods, higher perplexity, and greater statistical variance.

Mechanistically, an LLM predicts next-token probabilities by computing the hidden representations of the preceding context and passing them through a linear classifier. The success of prior studies in utilizing these probabilities for effective detection demonstrates that these hidden representations inherently contain features capable of distinguishing between MGT and HWT. However, these hidden representations also contain stylistic information, part of which is noise that causes reconstruction bias.

To achieve robust detection across diverse domains and LLMs, our objective is to disentangle the hidden representation, filtering out stylistic noise to extract a generalizable representation for distinguishing between HWT and MGT. We term this the generative artifact representation (GAR).

3.2 Generative Artifact Representation Extraction

Given a target text X tokenized into L tokens, $X = [x_1, x_2, \ldots, x_L]$, we employ a proxy model Φ to extract its hidden representations:

$$h = \Phi(X) \in \mathbb{R}^{1\times d}$$

where d denotes the representation dimension.

Then, we use a multilayer perceptron (MLP) as an artifact extractor. By non-linearly projecting the high-dimensional mixed representation h into a low-dimensional latent space, we filter out redundant stylistic noise from the original representations, thereby distilling a generalizable GAR:

$$r = MLP(h) \in \mathbb{R}^{1 \times d_f}$$

where d_f denotes the dimension of the generative artifact representation.

3.3 Global Artifact Aggregation

Text generation by LLMs is fundamentally an autoregressive process. Consequently, GAR can be extracted from any truncated prefix rather than being confined to the complete text sequence.

Given the truncated prefix $X_{\leq t} = [x_1, \ldots, x_t]$ at the token position t, we extract its corresponding local GAR r_t. Proceeding in this manner, we ultimately construct the sequence of GARs $[r_1, \ldots, r_L]$. Since GAR exhibits a high degree of homogeneity across the entire autoregressive sequence, we aggregate these local representations globally via average pooling:

$$r_G = \frac{1}{L} \sum_{i=1}^{L} r_i \in \mathbb{R}^{1 \times d_f}$$

Subsequently, the aggregated global representation r_G is mapped into the label space via a linear classifier, thereby computing the classification probabilities for both being human-written and machine-generated:

$$\hat{p} = \text{Softmax}(W_c r_G + b_c)$$

where $W_c \in \mathbb{R}^{2 \times d_f}$ and $b_c \in \mathbb{R}^2$ denote the learnable weights and bias of the linear classifier.

During training, we employ the cross-entropy loss to measure the discrepancy between the predicted probabilities and the ground-truth labels:

$$\mathcal{L}_{cls} = -\sum_{j=1}^{C} y_j \log(\hat{p}_j)$$

This aggregation mechanism effectively prevents the detector from excessively relying on local stylistic noise. Instead, it impels the GAR extractor to disentangle stylistic noise from hidden features, thereby capturing generalizable representations.

3.4 Contrastive Learning Based on Rewritten Text

Since rewriting constrains the LLM to stay close to the source text, the rewritten text may contain both machine-specific artifacts introduced by the LLM and stylistic noise inherited from the original text. Consequently, HWTs and their rewritten texts exhibit similar stylistic noise, yet their GARs are entirely distinct. Conversely, texts belonging to the same category, despite potentially exhibiting significant differences in stylistic noise, share highly consistent GARs.

Inspired by this core insight, we introduce supervised contrastive learning based on rewritten texts to explicitly optimize the GAR space, thereby effectively decoupling the GAR r_L information. Specifically, we first apply ℓ_2-normalization to the extracted representation r_L:

$$z = \frac{r_L}{\|r_L\|_2}$$

Utilizing these normalized embeddings, we compute the supervised contrastive loss (Khosla et al., 2020) across the target texts and their rewritten counterparts within each batch:

$$\mathcal{L}_{\text{SupCon}} = \sum_{i=1}^{2N} \frac{-1}{|P(i)|} \sum_{p \in P(i)} \log \frac{\exp(z_i \cdot z_p / \tau)}{\sum_{a \neq i} \exp(z_i \cdot z_a / \tau)}$$

where N denotes the batch size, $P(i)$ is the set of positive samples with the same class as anchor i, and τ represents the temperature parameter.

3.5 Two-Stage Training Strategy

The overall loss function combines the classification loss and the supervised contrastive loss:

$$\mathcal{L} = \mathcal{L}_{cls} + \mathcal{L}_{supcon}$$

Apart from the generative artifact extractor and classifier, the proxy model also requires fine-tuning. Because the pre-training next-token prediction objective biases LLM attention mechanisms toward semantic and logical coherence, it is not adaptable to the downstream MGTD task. Consequently, we fine-tune the attention layers of the proxy model to sensitize them to generative artifacts.

During the initial training phase, random initialization of the generative artifact extractor and classifier can propagate destructive gradients, potentially degrading the proxy model's representational capabilities. To mitigate this issue, we employ a

two-stage training strategy (Kumar et al., 2022) for our DisGAR framework:

Warm-up Stage. We freeze the proxy model and all of its LoRA parameters, exclusively training the generative artifact extractor and linear classifier. This stage forces the top-level modules to rapidly adapt to the proxy model's hidden representations, thereby establishing initial feature mappings and decision boundaries.

Joint Fine-tuning Stage.s The LoRA parameters are unfrozen, enabling the proxy model and the top-level modules to conduct joint fine-tuning. Driven by the joint multitask gradient, the underlying LoRA modules amplify generative artifacts, while the top-level modules eliminate stylistic noise. Collaboratively, this two-stage training strategy achieves highly robust detection across diverse text distributions.

4 Experiment

4.1 Experimental Setup

Dataset. We employ the *Disjoint-Input Generation* (DIG) subset from the MIRAGE benchmark (Fu et al., 2025) as our primary dataset. This dataset encompasses real-world corpora across 5 distinct domains, comprising texts generated by 17 mainstream open-source and closed-source LLMs for three major tasks: *Generate*, *Polish*, and *Rewrite*. Furthermore, it is constructed by randomly injecting 16 predefined linguistic styles during the generation process. Each task-specific test set contains approximately 14,000 to 16,000 samples.

Evaluation Metrics. To comprehensively evaluate the detector's performance, we utilize four diverse metrics: Area Under the Receiver Operating Characteristic Curve (AUROC), Matthews Correlation Coefficient (MCC), Balanced Accuracy, and True Positive Rate at a 5% false positive rate (TPR@5%). While AUROC captures the model's overall discriminative capability, Balanced Accuracy and MCC ensure a robust assessment under potential class imbalance. Additionally, TPR@5% measures the recall of MGTs strictly constrained by a 5% false positive rate (FPR) ceiling on HWTs.

Baseline. We primarily compare the proposed DisGAR framework with both zero-shot methods and training-based methods. The zero-shot methods include Likelihood (Solaiman et al., 2019), Entropy (Gehrmann et al., 2019), DetectGPT (Mitchell et al., 2023), Fast-DetectGPT (Bao et al., 2024) and Lastde (Xu et al., 2025). The training-based methods include RoBERTa series (Liu et al., 2019), BiScope (Guo et al., 2024), ImBD (Chen et al., 2025), DetectAnyLLM (Fu et al., 2025) and L2D (Zhou et al., 2026).

4.2 In-domain Detection

We adopt the few-shot, multi-domain training set from ImBD, which comprises 500 pairs of HWTs and GPT-3.5-turbo polished texts. Subsequently, we evaluate the in-domain MGTD performance on the MIRAGE benchmark.

As summarized in Table 1, experimental results demonstrate that our proposed DisGAR achieves state-of-the-art detection performance across all metrics and tasks. Specifically, DisGAR achieves AUROC scores exceeding 99.7%, alongside absolute improvements of 2.9%-3.2% in Balanced Accuracy and 5.7%-6.3% in MCC. At TPR@5%, our method reaches 99.19%-99.90%, yielding an absolute gain of 3.5%-4.6%. To provide a more comprehensive view, the detailed ROC curves in Figure 2 demonstrate that DisGAR consistently maintains higher TPR than baseline methods, particularly in stringent low-FPR constraints. This meets the practical requirement of robust MGT detection without falsely penalizing human creators.

In summary, these results powerfully demonstrate DisGAR's exceptional cross-model generalization and its robustness in handling complex human-AI collaborative texts.

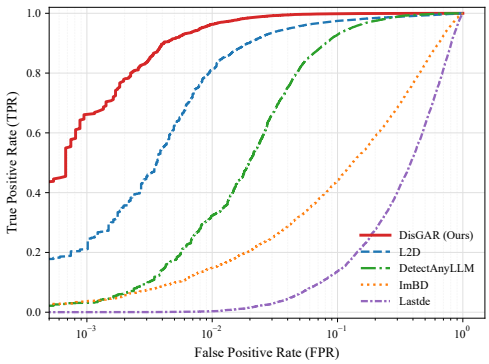

Figure 2: Receiver Operating Characteristic (ROC) curves of different detectors on the *Polish* task.

4.3 Out-of-Distribution (OOD) Detection

To evaluate the generalization capability of the detector on unseen models and domains, we categorize the test set of the *Polish* task into five domains: *Academic*, *Comment*, *Email*, *News*, and *Website*.

For each domain, we construct a training set comprising 500 pairs of human-written and Qwen3-

Table 1: In-domain detection results. The symbol "♠" indicates training-based methods. The "Imp." represents the improvement over the previous SOTA, calculated as (new-old)/(1.0-old).

Methods	Generate				Polish				Rewrite			
	AUROC	Accuracy	MCC	TPR@5%	AUROC	Accuracy	MCC	TPR@5%	AUROC	Accuracy	MCC	TPR@5%
Likelihood	0.4936	0.5091	0.0183	0.0147	0.4653	0.5000	0.0000	0.0214	0.4337	0.5000	0.0000	0.0148
LogRank	0.4992	0.5128	0.0260	0.0220	0.4512	0.5000	0.0000	0.0195	0.4225	0.5000	0.0000	0.0132
Entropy	0.6522	0.6150	0.2543	0.1099	0.5543	0.5417	0.1247	0.0954	0.5805	0.5566	0.1650	0.1189
RoBERTa-Base ♠	0.5523	0.5397	0.1434	0.1250	0.4859	0.5010	0.0088	0.0460	0.5020	0.5049	0.0293	0.0569
RoBERTa-Large ♠	0.4716	0.5217	0.0842	0.0871	0.5171	0.5151	0.0340	0.0633	0.5570	0.5385	0.0864	0.0895
LRR	0.5215	0.5341	0.0777	0.0701	0.4081	0.5000	0.0000	0.0200	0.3930	0.5000	0.0000	0.0188
DNA-GPT	0.5733	0.5595	0.1196	0.0776	0.4771	0.5004	0.0110	0.0309	0.4453	0.5001	0.0080	0.0251
NPR	0.6120	0.6140	0.2604	0.0191	0.5071	0.5370	0.1071	0.0318	0.4710	0.5201	0.0663	0.0226
DetectGPT	0.6402	0.6258	0.2758	0.0275	0.5469	0.5531	0.1328	0.0355	0.5061	0.5266	0.0826	0.0283
Fast-DetectGPT	0.7768	0.7234	0.4628	0.4310	0.5720	0.5570	0.1293	0.1189	0.5455	0.5432	0.1015	0.1025
BiScope ♠	0.8258	0.8258	0.6549	0.1958	0.6537	0.6537	0.3188	0.1220	0.6362	0.6362	0.2851	0.1131
Lastde	0.8172	0.7485	0.5110	0.4816	0.5851	0.5654	0.1345	0.1108	0.5747	0.5583	0.1328	0.1099
ImBD ♠	0.8597	0.7738	0.5497	0.4065	0.7888	0.7148	0.4300	0.2730	0.7825	0.7068	0.4139	0.2933
DetectAnyLLM ♠	0.9525	0.8988	0.7975	0.7770	0.9297	0.8732	0.7487	0.7756	0.9234	0.8705	0.7447	0.7778
L2D ♠	0.9771	0.9586	0.9179	0.9534	0.9841	0.9542	0.9085	0.9572	0.9839	0.9544	0.9088	0.9570
DisGAR (ours) ♠	**0.9991**	**0.9906**	**0.9811**	**0.9990**	**0.9977**	**0.9828**	**0.9656**	**0.9919**	**0.9978**	**0.9858**	**0.9717**	**0.9928**
Imp.	**+96.07%**	**+77.29%**	**+76.98%**	**+97.85%**	**+85.53%**	**+62.45%**	**+62.40%**	**+81.07%**	**+86.34%**	**+68.86%**	**+68.97%**	**+83.26%**

max (Yang et al., 2025) generated texts. We train the detector independently on the training set of each domain and evaluate its performance on the test sets of the remaining four domains. As illustrated in Figure 3, DisGAR consistently outperforms alternative baselines, exhibiting both stability and superiority under the cross-domain and cross-model setting. Notably, when L2D is trained on *Website*, its performance experiences a precipitous decline. Our analysis suggests that within this domain, L2D fails to effectively separate human-written texts from their rewritten counterparts, leading to poor generalization.

4.4 Efficiency Analysis

In real-world applications, detection efficiency is a critical metric for evaluating a detector's deployment viability. To this end, we evaluate the detection time cost for a single sample of DisGAR alongside seven baseline methods. The detailed results are presented in Figure 4.

DisGAR achieves an optimal trade-off between computational efficiency and detection performance. Since DisGAR avoids the text rewriting during inference, its computational time is primarily limited by the proxy model, yielding a detection time cost comparable to Fast-DetectGPT. While DetectAnyLLM exhibits the shortest time (approximately one-fourth of DisGAR's), our method still achieves substantially higher detection performance. Meanwhile, although L2D achieves better detection performance than DetectAnyLLM, its time cost is approximately 180× that of DisGAR.

4.5 Short-text Robustness

Short-text detection remains a highly challenging task, as the limited context significantly exacerbates the distributional overlap between HWTs and MGTs. To evaluate the length robustness of detectors, we conduct experiments on a short-text test set provided by D&R (Sun et al., 2026) (WikiHow, AG-News, and Reddit), where all MGTs are generated by Grok-3. To ensure a fair comparison, all baselines are trained under the same data setup specified in Section 4.2.

As shown in Figure 5(a), DisGAR exhibits superior short-text robustness compared to baseline methods, maintaining a high AUROC of 0.9563 even when confronting texts shorter than 40 tokens. Meanwhile, Figure 5(b) reveals that while DisGAR's detection efficacy for MGTs experiences a marginal decline on short-text, its performance on HWTs remains virtually unaffected.

4.6 Ablation Study

4.6.1 Effectiveness of Key Components

To validate the effectiveness of the individual mechanisms within the DisGAR framework, we conduct an ablation study adopting the experimental setup from Section 4.3. Specifically, we train the detector on the *Academic* training set and evaluate its performance across the remaining four domains.

As shown in Table 2, removing individual modules uniformly leads to noticeable performance degradation. Specifically, freezing the parameters of the proxy model results in an accuracy drop of 7% on the test set. The general-purpose representations of a proxy model without fine-tuning are not optimized for MGTD, making it inherently

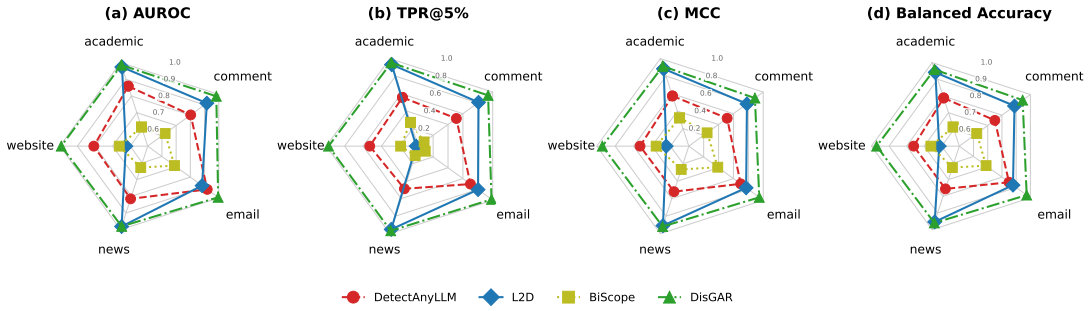

Figure 3: OOD detection results. Subfigures (a)-(d) compare performance across different evaluation metrics.

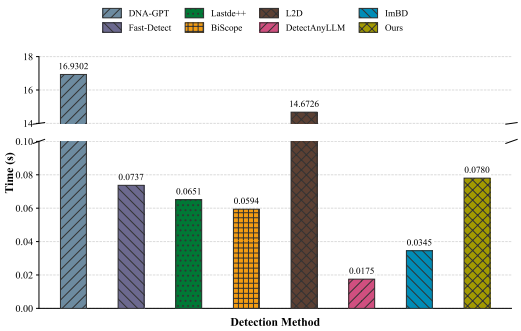

Figure 4: Time cost comparison between DisGAR and seven baselines for single-sample processing. The detection time is calculated based on a text of approximately 500 tokens (averaged over multiple detections after warm-up). No model inference acceleration is applied to any of the methods.

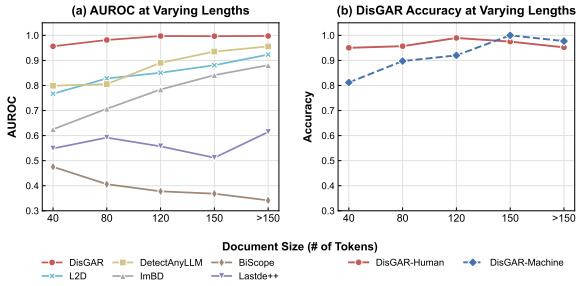

Figure 5: Impact of target text length on detection performance. (a) shows the AUROC of various detection methods across different text lengths. (b) presents DisGAR's detection accuracy across various text lengths at a fixed classification threshold 0.17.

difficult to capture the subtle, underlying machine-generated patterns. Furthermore, when the global artifact aggregation and the supervised contrastive learning based on rewritten texts are individually removed, the detector's accuracy drops by 7.2% and 7.3%, respectively. These results confirm that both mechanisms play distinct and complementary roles in extracting style-agnostic generative artifacts, and their synergy significantly enhances the generalization capability of DisGAR.

Table 2: Ablation study results on overall performance. "w/o LoRA" denotes freezing the proxy model and only training the GAR extractor and linear classifier. "w/o Agg." denotes removing the global artifact aggregation mechanism. "w/o Con." denotes removing the supervised contrastive learning based on rewritten texts.

Method	AUROC	TPR@5%	MCC	Accuracy
w/o LoRA	0.9394	0.8268	0.7728	0.8884
w/o Agg.	0.9235	0.8205	0.7706	0.8864
w/o Con.	0.9448	0.8194	0.7676	0.8855
DisGAR	0.9851	0.9529	0.9130	0.9583

4.6.2 Domain Probing Analysis

To verify that DisGAR effectively strips away stylistic noise and consequently reduces the domain-discriminative features within GAR, we conduct a domain probing analysis for the *Academic* and *Comment* domains.

As illustrated in Figure 6, the probing classifier based on DisGAR exhibits a consistently higher loss value. This observation indicates that the representations extracted by DisGAR retain substantially less stylistic noise and domain-specific information, partially corroborating its capability to disentangle underlying generation artifacts.

4.6.3 Impact of proxy models

We evaluate the impact of utilizing various proxy models on detection performance. As shown in Figure 7, the empirical results vary across configurations, which can be attributed to differences in parameter scales and model architectures. With the exception of gpt-neo-2.7B, which performed relatively poorly, all other models consistently achieve a detection accuracy exceeding 94%. Overall, larger models generally yielded better results, but at the expense of increased inference time cost.

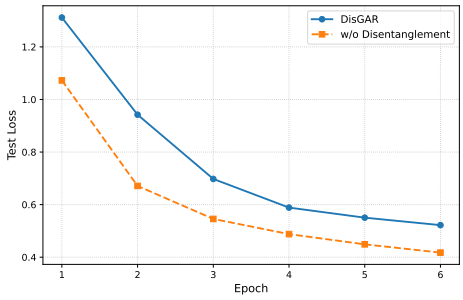

Figure 6: Domain classification test loss comparison. "w/o Disentanglement" shares identical components with the DisGAR framework, except that it solely utilizes the MLP-processed full-text representation. Both methods undergo identical training on the MGTD task, following the setup described in Section 4.2.

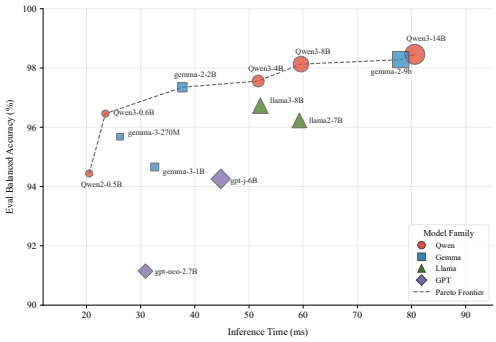

Figure 7: Effectiveness and efficiency comparison of DisGAR across different open-source LLMs. Following the setup in Section 4.2, all models are trained on the same dataset and evaluated using balanced accuracy on the *Polish* test set. Time calculation follows the methodology described in Section 4.4.

4.7 Other Analysis

4.7.1 Other languages

To assess DisGAR's performance across languages, we construct multilingual test suites (comprising Chinese, German, Italian, and Japanese texts) via Qwen-turbo. As illustrated in Figure 8(a), the results indicate that the detector demonstrates strong performance across all evaluated languages, with detection accuracy consistently at or above 95%. Additionally, on the Chinese test set provided by MIRAGE (Fu et al., 2025) , DisGAR achieves an AUROC of 0.9506 and an average accuracy of 0.8902, underscoring its high efficacy in cross-lingual scenarios.

4.7.2 Memorization

LLMs are highly prone to memorization overfitting on content that appears frequently in the training corpus, such as legal statutes and famous quotes.

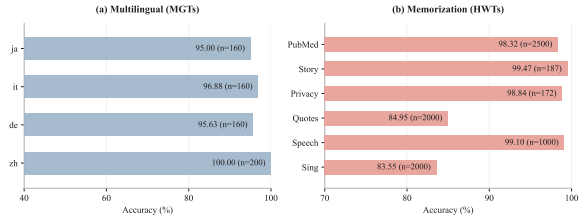

Figure 8: Other analysis results. (a) Accuracy of DisGAR on generated texts across different languages; (b) Accuracy of DisGAR on various memorized texts. The classification threshold is set to 0.17.

Consequently, when evaluating such texts, the proxy model tends to assign them anomalously high prediction probabilities, thereby causing the detector to misclassify them as MGTs.

By fine-tuning the attention layers of the proxy model to focus on the detection task, DisGAR mitigates the misclassification of memorized texts to a certain extent. As shown in Figure 8(b), DisGAR demonstrates an outstanding capability in distinguishing memorized texts. Even on the English song data, where it performs relatively the weakest, its detection accuracy still reaches 83.55%.

5 Conclusion

In this paper, we propose DisGAR, an MGTD framework that learns style-invariant generative artifact representations. Motivated by the observation that hidden representations of proxy models contain both generation-specific signals and stylistic noise, DisGAR disentangles these through global artifact aggregation and supervised contrastive learning based on rewritten text. In this way, the detector can better capture generalizable generation patterns while reducing its reliance on superficial stylistic cues. Extensive experiments demonstrate that DisGAR consistently outperforms strong zero-shot and training-based baselines under in-domain, cross-domain and cross-model settings. Further analyses show that DisGAR maintains competitive inference efficiency and exhibits strong robustness on short texts, multilingual generated texts, and memorized human-written texts. These results suggest that disentangling generative artifacts from stylistic noise is a promising direction for building more reliable and practical machine-generated text detectors.

Limitations

Although our proposed DisGAR framework demonstrates strong performance, several limitations remain. (1) DisGAR's detection performance degrades when processing low-quality texts containing substantial noise, such as grammatical errors or semantic incoherence. (2) While our method achieves competitive results in multi-domain and multi-model detection, establishing a universally fixed detection threshold remains challenging. However, based on our experiments, we selected 0.17 as a relatively stable classification threshold. (3) Our current research primarily focuses on three relatively isolated scenarios: generation, polishing, and rewriting. The more complex task of document-level detection for human-machine mixed texts has not yet been fully explored. We plan to improve these aspects in our future work.

Ethical Considerations

This research focuses on the detection of MGT. We acknowledge the inherent risk of false positives, where HWT could be misclassified as machine-generated, potentially leading to unfair judgments. Therefore, our framework is designed to serve as an assistive tool rather than an absolute decision-maker. Additionally, all datasets used in this study consist of publicly available text, and our data processing strictly adheres to privacy regulations, ensuring no personally identifiable information (PII) is exposed.

References

David Ifeoluwa Adelani, Haotian Mai, Fuming Fang, Huy H, Nguyen, Junichi Yamagishi, and Isao Echizen. 2020. Generating sentiment-preserving fake online reviews using neural language models and their human- and machine-based detection. In *Proceedings of the 34th International Conference on Advanced Information Networking and Applications (AINA-2020)*, pages 1341–1354.

Guangsheng Bao, Yanbin Zhao, Zhiyang Teng, Linyi Yang, and Yue Zhang. 2024. Fast-detectGPT: Efficient zero-shot detection of machine-generated text via conditional probability curvature. In *The Twelfth International Conference on Learning Representations*.

Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiushi Du, Zhe Fu, Huazuo Gao, Kaige Gao, Wenjun Gao, Ruiqi Ge, Kang Guan, Daya Guo, Jianzhong Guo, Guangbo Hao, Zhewen Hao, and 67 others. 2024. Deepseek llm: Scaling open-source language models with longtermism. *Computing Research Repository*, arXiv:2401.02954.

Jiaqi Chen, Xiaoye Zhu, Tianyang Liu, Ying Chen, Chen Xinhui, Yiwen Yuan, Chak Tou Leong, Zuchao Li, Long Tang, Lei Zhang, Chenyu Yan, Guanghao Mei, Jie Zhang, and Lefei Zhang. 2025. Imitate before detect: aligning machine stylistic preference for machine-revised text detection. In *Proceedings of the Thirty-Ninth AAAI Conference on Artificial Intelligence and Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence and Fifteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'25/IAAI'25/EAAI'25. AAAI Press.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, Luke Marris, Sam Petulla, Colin Gaffney, Asaf Aharoni, Nathan Lintz, Tiago Cardal Pais, Henrik Jacobsson, Idan Szpektor, Nan-Jiang Jiang, and 81 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *Computing Research Repository*, arXiv:2507.06261.

Erna Daniati, Aji Prasetya Wibawa, Wahyu Sakti Gunawan Irianto, and Andrew Nafalski. 2025. Attention-enhanced convolutional neural network for context extraction in andersen's fairy tales. *JOIV : International Journal on Informatics Visualization*.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, and 181 others. 2025. Deepseek-v3 technical report. *Computing Research Repository*, arXiv:2412.19437v2. Version 2.

Franck Dernoncourt and Ji Young Lee. 2017. PubMed 200k RCT: a dataset for sequential sentence classification in medical abstracts. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 308–313, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Jiachen Fu, Chun-Le Guo, and Chongyi Li. 2025. Detectanyllm: Towards generalizable and robust detection of machine-generated text across domains and models. In *Proceedings of the 33rd ACM International Conference on Multimedia*, MM '25, page 11229–11238, New York, NY, USA. Association for Computing Machinery.

Sebastian Gehrmann, Hendrik Strobelt, and Alexander Rush. 2019. GLTR: Statistical detection and visualization of generated text. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 111–116,

Florence, Italy. Association for Computational Linguistics.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Computing Research Repository*, arXiv:2407.21783v3. Version 3.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, and 175 others. 2025. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638.

Hanxi Guo, Siyuan Cheng, Xiaolong Jin, Zhuo Zhang, Kaiyuan Zhang, Guanhong Tao, Guangyu Shen, and Xiangyu Zhang. 2024. Biscope: Ai-generated text detection by checking memorization of preceding tokens. In *Advances in Neural Information Processing Systems*, volume 37, pages 104065–104090. Curran Associates, Inc.

Tahmid Hasan, Abhik Bhattacharjee, Md. Saiful Islam, Kazi Mubasshir, Yuan-Fang Li, Yong-Bin Kang, M. Sohel Rahman, and Rifat Shahriyar. 2021. XL-sum: Large-scale multilingual abstractive summarization for 44 languages. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4693–4703, Online. Association for Computational Linguistics.

Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Mądry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, Alex Nichol, Alex Paino, and 80 others. 2024. Gpt-4o system card. *Computing Research Repository*, arXiv:2410.21276.

Daphne Ippolito, Daniel Duckworth, Chris Callison-Burch, and Douglas Eck. 2020. Automatic detection of generated text is easiest when humans are fooled. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1808–1822, Online. Association for Computational Linguistics.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Comput. Surv.*, 55(12).

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. 2019. PubMedQA: A dataset for biomedical research question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2567–2577, Hong Kong, China. Association for Computational Linguistics.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. In *Advances in Neural Information Processing Systems*, volume 33, pages 18661–18673. Curran Associates, Inc.

Ananya Kumar, Aditi Raghunathan, Robbie Matthew Jones, Tengyu Ma, and Percy Liang. 2022. Fine-tuning can distort pretrained features and underperform out-of-distribution. In *International Conference on Learning Representations*.

Jooyoung Lee, Thai Le, Jinghui Chen, and Dongwon Lee. 2023. Do language models plagiarize? In *Proceedings of the ACM Web Conference 2023*, pages 3637–3647.

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. *Computing Research Repository*, abs/1907.11692.

Chengzhi Mao, Carl Vondrick, Hao Wang, and Junfeng Yang. 2024. Raidar: generative AI detection via rewriting. In *The Twelfth International Conference on Learning Representations*.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D. Manning, and Chelsea Finn. 2023. Detectgpt: zero-shot machine-generated text detection using probability curvature. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org.

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.

Artidoro Pagnoni, Martin Graciarena, and Yulia Tsvetkov. 2022. Threat scenarios and best practices to detect neural fake news. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 1233–1249, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Qwen, 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, and 24 others. 2025. Qwen2.5 technical report.

Pavlos Sermpezis, Stelios Karamanidis, Eva Paraschou, Ilias Dimitriadis, Sofia Yfantidou, Filitsa-Ioanna Kouskouveli, Thanasis Troboukis, Kelly Kiki, Antonis Galanopoulos, and Athena Vakali. 2025. Agoraspeech: A multi-annotated comprehensive dataset of political discourse through the lens of humans and ai. *Computing Research Repository*, arXiv:2501.06265.

Eva Sharma, Chen Li, and Lu Wang. 2019. BIGPATENT: A large-scale dataset for abstractive and coherent summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2204–2213, Florence, Italy. Association for Computational Linguistics.

Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, Gretchen Krueger, Jong Wook Kim, Sarah Kreps, Miles McCain, Alex Newhouse, Jason Blazakis, Kris McGuffie, and Jasmine Wang. 2019. Release strategies and the social impacts of language models. *Computing Research Repository*, 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, Singapore. Association for Computational Linguistics.

Yuxia Sun, Ran Zhang, Aoxiang Sun, Xu Li, Zitao Liu, and Jingcai Guo. 2026. D&r: Recovery-based AI-generated text detection via a single black-box LLM call. In *The Fourteenth International Conference on Learning Representations*.

Lindia Tjuatja, Valerie Chen, Tongshuang Wu, Ameet Talwalkwar, and Graham Neubig. 2024. Do LLMs exhibit human-like response biases? a case study in survey design. *Transactions of the Association for Computational Linguistics*, 12:1011–1026.

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. In *The Thirteenth International Conference on Learning Representations*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *Computing Research Repository*, arXiv:2505.09388.

Xianjun Yang, Wei Cheng, Yue Wu, Linda Ruth Petzold, William Yang Wang, and Haifeng Chen. 2024. DNA-GPT: Divergent n-gram analysis for training-free detection of GPT-generated text. In *The Twelfth International Conference on Learning Representations*.

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*.

A More details on Dataset

A.1 MIRAGE benchmark

MIRAGE comprises two subsets: Disjoint-Input Generation (DIG) and Shared-Input Generation (SIG). In DIG, each LLM generates MGT or MRT based on an independent HWT, whereas in SIG, multiple LLMs generate MGT or MRT based on the same HWT. We selected DIG as our test set.

Source Domains and Datasets. MIRAGE encompasses a diverse range of text domains, including Academic, Email, Website, News, and Comment. The mapping between these domains and the corre sponding source datasets is summarized in Table 3.

Table 3: Source datasets for each domain.

Domains	Datasets
Academic	BigPatent(Sharma et al., 2019), NeurIPS, ArXiv, PubMed-Abstracts(Jin et al., 2019)
Email	Enron-Emails
Website	OpenWebText
News	CNNDailyMails, XSum(Narayan et al., 2018), XLSum(Hasan et al., 2021)
Comment	Amazon-Review

Source Generator LLMs. The source LLMs used for data generation in MIRAGE are listed in Table 4. In total, MIRAGE samples machine-generated texts (MGT) using 13 powerful commercial LLMs and 4 advanced open source LLMs.

Table 4: Commercial LLMs are highlighted in bold.

Series	Models
GPT	**GPT-4o**(Hurst et al., 2024), **GPT-o3-mini**, **GPT-4o-mini**(Hurst et al., 2024)
Claude	**Claude-3.5-Haiku**, **Claude-3.7-sonnet**
DeepSeek	**DeepSeek-V3**(DeepSeek-AI et al., 2025), **DeepSeek-R1**(Guo et al., 2025)
Gemini	**Gemini-2.0-Flash**, **Gemini-2.0-Flash-Lite**
Qwen	Qwen-2.5-7B(Qwen et al., 2025), Qwen-2.5-7B-R1-Distill(Qwen et al., 2025), **QwQ-Plus**
LLaMa	LLaMa-3.1-8B(Grattafiori et al., 2024), LLaMa-3.1-8B-R1-Distill(Grattafiori et al., 2024)
Grok	**Grok2**
Moonshot	**Moonshot-v1**
Doubao	**Doubao-1.5-pro-32k**

Style Control. The style controls used in MIRAGE are shown in Table 5.

Table 5: Detail style list.

Style List
formal, oral, academic, literary, critical, narrative, descriptive, lyric, objective, subjective, original, casual, expository, argumentative, journalistic, poetic

A.2 The Training Set from the ImBD

The human-written texts included in the training set were crawled from the internet before 2019. The texts are then polished by GPT-3.5. The dataset has 500 pairs of samples for training. The composition of the dataset is 57.3% papers, 14.2% blogs, 4.0% letters and emails, and 2.1% homework, with the remaining portion not specified by the authors.

A.3 Single-Domain Training Set for OOD Testing

For our OOD evaluation, the *Polish* subset of the MIRAGE-DIG dataset serves as the primary test set. To construct our five single-domain training sets, we sampled 500 HWTs per domain from this Polish dataset and supplemented them with 500 corresponding MGTs produced via Qwen3-max. To prevent data leakage and maintain evaluation integrity, the sampled HWTs were naturally excluded from the final *Polish* test set.

The prompts used for text generation are listed below:

> Write an article about 150 words in a <style> style starting exactly with: <original>

The <original> is the first 30 tokens of a HWT. The <style> is from the Table 6.

A.4 Multilingual Test Set

We generated a small number of Chinese, German, Italian, and Japanese texts using Qwen-turbo for Multilingual testing.

The prompts used for text generation are listed below:

> Topic: <original>
> Please write a short paragraph in <language> about this topic. The paragraph should be approximately 300 characters in length. Keep it natural and fluent in <language>.

The <original> is the first 30 tokens of a HWT. The <language> is the target language for text generation.

A.5 Memorized Text Test Set

We selected six categories of memorized texts from public domains for testing, including English songs, campaign speeches(Sermpezis et al., 2025), famous quotes, GDPR legal texts, Andersen's Fairy Tales (Daniati et al., 2025), and medical paper abstract(Dernoncourt and Lee, 2017).

Specifically, we conducted testing on medical paper abstracts primarily because they contain a high frequency of memorized terms (proper nouns and collocations). As for the GDPR legal texts, we processed them article by article.

A.6 Domain classification test set

We selected 1,000 HWTs from each of the *Academic* and *Comment*, and equally divided them into training and test sets.

B More details on Experiment

B.1 Experiment Setup

Device. All of our experiments are conducted in Linux 5.15.0 (Ubuntu 22.04.5), using a single RTX PRO 6000 GPU with 96GB GPU memory. The Python version is 3.12.3, the PyTorch version is 2.7.0, and the Transformers version is 4.57.3.

LoRA Config. We adopt a LoRA configuration specifically designed for feature extraction, with a rank of 8, a LoRA alpha of 32, and a dropout rate of 0.1. Additionally, the trainable parameters are targeted at the attention projection modules with no bias applied.

Training Setup for DisGAR. The framework is built upon the Gemma-2-9B-it proxy model, loaded in bfloat16 precision to optimize memory efficiency. The model is optimized using the AdamW optimizer with a training batch size of 4. The entire training process spans 4 epochs and is decoupled into two distinct stages. Specifically, we conduct a 1-epoch head warmup phase where only the top-level learning rate is set to 1e-4. This is followed by 3 epochs of joint fine-tuning, during which the learning rates for both the top-level modules and the proxy model are uniformly set to 1e-4. In each phase, we apply a linear learning rate scheduler with a warmup ratio of 0.1. Additionally, the temperature parameter for the contrastive learning objective is set to 0.07.

B.2 Baseline Method Setup

Table 7 lists the pre-trained models utilized across different detection methods. Among them, the RoBERT-based method was not retrained on the dataset. For L2D (Zhou et al., 2026), we rewrite the target text only once. All other settings remain the same.

B.3 Prompts used for text rewriting

We employ DeepSeek-V3.2 to rewrite the target texts. The prompts used for text rewriting are listed below:

> Strictly rewrite the following text, output only the rewritten result.
> [Core Instructions]: 1. Maintain the original meaning and language style 2. Absolutely do not add any explanations, introductions, or summaries 3. The output language must be exactly the same as the input language
> [the original text]: <original>

The <original> is the target text.

B.4 Detailed Results

The detailed results of the OOD test are shown in Table 7. The detailed results of different proxy models are shown in Table 8.

Table 6: Pre-trained Models for Various Detection Methods

Methods	Model
Likelihood (Solaiman et al., 2019)	gpt-neo-2.7B
Entropy (Gehrmann et al., 2019)	gpt-neo-2.7B
Log-Rank (Ippolito et al., 2020)	gpt-neo-2.7B
LRR (Su et al., 2023)	gpt-neo-2.7B
RoBERTa-Base (Liu et al., 2019)	RoBERTa-base
RoBERTa-Large (Liu et al., 2019)	RoBERTa-large
DNA-GPT (Yang et al., 2024)	gpt-neo-2.7B
NPR (Su et al., 2023)	T5-3B /gpt-neo-2.7B
DetectGPT (Mitchell et al., 2023)	T5-3B / gpt-neo-2.7B
Fast-DetectGPT (Bao et al., 2024)	gpt-neo-2.7B/gpt-j-6B
BiScope (Guo et al., 2024)	Llama2-7B
ImBD (Chen et al., 2025)	gpt-neo-2.7B
DetectAnyLLM (Fu et al., 2025)	Qwen2-0.5B
Lastde (Xu et al., 2025)	gpt-j-6B
L2D (Zhou et al., 2026)	gemma-2-9b-it
DisGAR	gemma-2-9b-it

Table 7: Detailed Results on OOD test

Source Domain	Metric	DisGAR	L2D	DetectAnyLLM	BiScope
academic	AUROC	**0.9851**	0.9731	0.8591	0.6145
	TPR@5%	**0.9529**	0.9318	0.5571	0.2717
	MCC	**0.9130**	0.8837	0.5737	0.3228
	Bal. Acc.	**0.9583**	0.9414	0.7884	0.6145
comment	AUROC	**0.9827**	0.9160	0.8024	0.6226
	TPR@5%	**0.9381**	0.8099	0.5115	0.0786
	MCC	**0.8882**	0.7812	0.5151	0.2461
	Bal. Acc.	**0.9455**	0.8883	0.7498	0.6226
email	AUROC	**0.9957**	0.8826	0.9187	0.6884
	TPR@5%	**0.9842**	0.8034	0.7009	0.0939
	MCC	**0.9469**	0.7703	0.6970	0.3894
	Bal. Acc.	**0.9739**	0.8776	0.8497	0.6884
news	AUROC	0.9800	**0.9810**	0.8153	0.6282
	TPR@5%	**0.9669**	0.9513	0.4872	0.1084
	MCC	**0.9186**	0.9087	0.5201	0.2661
	Bal. Acc.	**0.9594**	0.9533	0.7557	0.6282
website	AUROC	**0.9922**	0.6245	0.8067	0.6619
	TPR@5%	**0.9748**	0.0225	0.5271	0.1916
	MCC	**0.9396**	0.2455	0.5298	0.3549
	Bal. Acc.	**0.9707**	0.6144	0.7599	0.6619

Table 8: Evaluation Results of Different Models

Model	Eval AUROC	TPR@5%	Eval MCC	Balanced Accuracy	Time/Sample (s)
Qwen2-0.5B	0.9821	0.9366	0.8891	0.9444	0.0205
Qwen3-0.6B	0.9896	0.9661	0.9295	0.9646	0.0235
Qwen3-4B	0.9933	0.9779	0.9514	0.9756	0.0517
Qwen3-8B	0.9960	0.9866	0.9627	0.9813	0.0596
Qwen3-14B	0.9979	0.9928	0.9691	0.9846	0.0806
gpt-neo-2.7B	0.9688	0.8710	0.8239	0.9115	0.0309
gpt-j-6B	0.9845	0.9343	0.8856	0.9426	0.0448
llama2-7B	0.9885	0.9630	0.9249	0.9623	0.0593
llama3-8B	0.9918	0.9718	0.9348	0.9673	0.0521
gemma-2-9b	0.9977	0.9919	0.9656	0.9828	0.0780
gemma-3-270M	0.9865	0.9551	0.9143	0.9568	0.0262
gemma-3-1B	0.9838	0.9419	0.8933	0.9466	0.0326
gemma-2-2B	0.9955	0.9808	0.9471	0.9735	0.0377