Zero-Shot Detection of Generated Text via Prompt-Guided Probability Correction across Vocabulary-Heterogeneous Models

Anonymous ACL submission

Abstract

The widespread application of Large Language Models has aroused extensive attention to the distinction between human-created and machine-generated texts. Zero-shot detectors have gained prominence due to their elimination of the need for additional training data. However, a critical limitation in current methods is the neglect of prompts during the text generation process, which inevitably leads to biased token probability calculations. To address this issue, this paper proposes ProCDetect, a novel zero-shot detection approach designed to mitigate prompt-induced bias. Specifically, ProCDetect utilizes automatically generated prompts to reconstruct the authentic context of text generation, which fills the gap of prompt neglect in existing methods. Most notably, it innovatively leverages two vocabulary-heterogeneous models to calibrate token probabilities by capturing the discrepancies between original text and prompt-guided generated text. This unique design not only effectively alleviates bias caused by the lack of prompt consideration but also ensures robust detection performance across diverse generation models. Experimental results demonstrate that this method significantly outperforms existing baseline methods across multiple datasets, achieving an average AUROC of 0.9944. Moreover, it exhibits excellent generalization ability in out-of-distribution scenarios across models, with an average F1-score of 0.9451, representing an relative improvement of 7.22% compared to the optimal baseline method.

1 Introduction

Large Language Models (LLMs) have achieved remarkable success in various natural language processing tasks and exerted a profound impact across diverse fields. However, their strong capability to generate human-like text has raised concerns about potential abusive behaviors, such as academic misconduct (Yu et al., 2025) and the dissemination of disinformation (Pagnoni et al., 2022). This underscores the urgent need for us to develop reliable methods to distinguish between machine-generated and human-written texts.

Despite the rapid development of detection techniques, current approaches still face significant challenges in generalization and robustness. Common supervised detectors are effective within specific domains, but often suffer from severe performance degradation when facing unseen domains or novel generation models (Bakhtin et al., 2019; Pu et al., 2023; Uchendu et al., 2020). Conversely, existing zero-shot methods, which leverage pre-trained language models to calculate statistical features (e.g., perplexity or entropy), avoid the need for extensive training data. However, these methods typically rely solely on surface-level probability distributions and often struggle to distinguish the subtle nuances between high-quality machine-generated texts and human writings, limiting their effectiveness in complex and open-ended scenarios (Sadasivan et al., 2025).

In real-world, whether text is generated using LLMs or written manually, a certain prompt is required to guide the text generation process. However, most of existing methods primarily center their analysis on the text itself, largely overlooking the generative context provided by prompts (Hans et al., 2024; Park et al., 2024). Although recent studies have begun to acknowledge the critical role of prompts — noting that specific prompts can affect detection performance (Lu et al., 2024; Shi et al., 2024) and attempting to incorporate them into detection pipelines (Yu et al., 2024; Park et al., 2024; Sakurai et al., 2024) — significant limitations remain. Most current prompt-aware approaches treat prompts merely as auxiliary contexts without explicitly modeling their conditional probability influence, or lack robust mechanisms for correcting likelihood estimations across heterogeneous models. Consequently, effectively leveraging re-

constructed prompts to refine zero-shot detection signals across vocabulary-diverse models remains an unresolved challenge.

To address these limitations, this study proposes ProCDetect, a novel zero-shot detection framework that leverages prompt reconstruction and inter-model probability correction. Unlike previous approaches, our method explicitly utilizes the discrepancy in how two vocabulary-heterogeneous models process prompt-guided text to correct text likelihood. Specifically, for a given text x, we first employ a language model to reverse-engineer a potential prompt. This generated prompt is concatenated with x to form a prompt-guided context $\tilde{x}$. We then introduce a dual-level alignment strategy to bridge the vocabulary gap between two distinct pre-trained models. By analyzing the differential predictions of these models on token probabilities under both x and $\tilde{x}$, we compute a corrected text likelihood that mitigates model-specific biases. This corrected metric serves as a robust indicator for zero-shot classification.

The primary objective of the proposed method is to enhance the distinctiveness between the likelihood distributions of human-written and machine-generated texts, as schematically illustrated in Fig. 1. While traditional likelihood metrics often result in significant distributional overlap (Fig. 1(a)), our proposed correction mechanism aims to effectively separate these distributions (Fig. 1(b)). This enhanced separability allows for effective zero-shot classification simply by establishing a threshold on the corrected likelihood. Comprehensive empirical validation in Section 4 confirms the attainment of this objective, demonstrating significant improvements over state-of-the-art baselines and sustained robustness in challenging cross-model out-of-distribution scenarios.

The main contributions of this study can be summarized into the following three aspects:

1. Integrating prompts into the zero-shot detection pipeline to emphasize their essential role in likelihood-based classification.

2. Proposing a dual-level alignment strategy that enables probability calculation across vocabulary-heterogeneous models.

3. The important significance of inter-model differences for token-level probability correction is clarified.

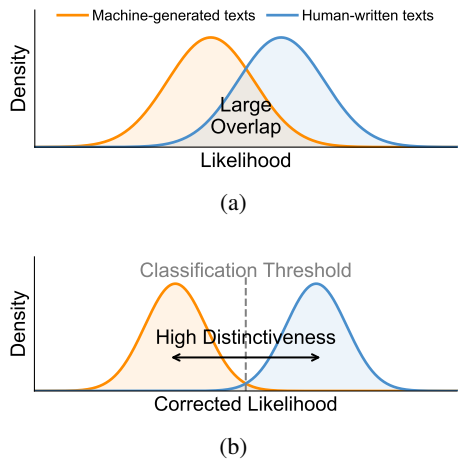

Figure 1: Schematic comparison of likelihood distributions. (a) Traditional likelihood metrics typically exhibit significant overlap between human-written and machine-generated texts. (b) The proposed method aims to correct these estimations, significantly reducing overlap and enhancing the distinctiveness for effective detection.

2 Related Work

Detecting LLM-generated text has become increasingly important for maintaining academic integrity and combating misinformation. Existing methods generally fall into three categories: watermarking, supervised learning, and zero-shot detection.

Watermarking techniques proactively embed hidden signals during generation to enable later authorship verification. Studies have explored embedding statistical signals by promoting randomized token subsets (Kirchenbauer et al., 2023), trigger-based schemes for ownership verification (Dai et al., 2022), or using masked language models for steganography (Ueoka et al., 2021). Recent work also applies logit-based watermarks to high-entropy segments to balance detection and quality (Lee et al., 2024). However, adversarial rewriting and distillation can compromise watermark robustness (Sadasivan et al., 2025).

Supervised detection methods typically frame the task as a binary classification problem. Early approaches used statistical features (Ippolito et al., 2020; Verma et al., 2024; Wu et al., 2023) or neural representations (Uchendu et al., 2020; Bhattacharjee et al., 2023; Wang et al., 2023, 2024), while current trends favor fine-tuning pre-trained models like BERT or RoBERTa (Devlin et al., 2019; Liu et al., 2019; Chen et al., 2023; Mitrović et al., 2023; Rodriguez et al., 2022). Representative frameworks include OpenAI's classifier (Solaiman et al., 2019)

and RADAR (Hu et al., 2023), the latter using adversarial training against paraphrasing. Despite strong in-domain performance, these models often overfit training distributions and struggle with generalization to evolving LLMs (Pu et al., 2023; Uchendu et al., 2020).

Zero-shot methods exploit statistical signatures of machine-generated text without task-specific training, often offering better generalization (Zhu et al., 2023). These approaches typically rely on pre-trained LLMs to simulate the generation process and extract features such as likelihood, perplexity, log-rank, and intrinsic dimension (Tulchinskii et al., 2023), followed by threshold-based classification. Representative methods include LogRank and LRR (Su et al., 2023), which use token-rank statistics; DetectGPT (Mitchell et al., 2023) and Fast-DetectGPT (Bao et al., 2024), which measure probability curvature; Raidar (Mao et al., 2024), which computes editing distances of rewritten texts; and Binoculars (Hans et al., 2024), which measures cross-perplexity between two models. While Binoculars acknowledges that prompts influence token prediction, its normalization is performed only at the text level and does not explicitly incorporate prompts into the detection pipeline.

Despite their strong generalization ability, existing zero-shot methods implicitly assume that the prompt is either known or has a negligible effect. In practice, prompts are often unavailable, and ignoring prompt-induced bias can lead to unreliable token probability estimates when using proxy models (Hans et al., 2024). Furthermore, most existing methods rely on vocabulary-homogeneous models to ensure aligned tokenization (Bao et al., 2024; Hans et al., 2024), which constrain their ability to capture richer forms of model disagreement. These limitations motivate the need for a prompt-aware detection framework that explicitly models prompt influence and leverages heterogeneous model behavior to perform probability correction at the token level for more robust machine-generated text detection.

3 Method

3.1 Preliminaries and Overview

Leveraging the likelihood of text has proven to be an effective approach for detecting machine-generated content. Given a text $x = (x_1, \ldots, x_L)$, its likelihood can be expressed as:

$$\text{Likelihood} = \frac{1}{L} \sum_{i=1}^{L} \log p(x_i | x_{<i}), \qquad (1)$$

where $\log p(x_i|x_{<i})$ denotes the log-probability predicted by the language model for the i-th token x_i given the preceding tokens $x_{<i}$. In practice, the true generation model is often unavailable, and token probabilities must be approximated using open-source proxy models.

We hypothesize that the disagreement between token-level probability estimates produced by two heterogeneous language models reflects the extent to which missing contextual information, including the original prompt, affects token prediction. In particular, larger disagreement indicates greater uncertainty in proxy likelihood estimation and motivates stronger probability correction. Based on this insight, we propose a prompt-guided, token-level probability correction framework that explicitly models inter-model disagreement.

Specifically, our framework intentionally employs vocabulary-heterogeneous language models to amplify predictive disagreement, rather than restricting both models to share identical vocabularies. To handle the inconsistencies introduced by heterogeneous tokenizations, we align both token sequences and probability vectors before computing correction coefficients. The corrected token probabilities are then aggregated into a detection score for classification.

ProCDetect consists of three stages: prompt generation, correction coefficient calculation, and classification. The overall framework is illustrated in Fig. 2, and each component is described in detail below.

3.2 Prompt Generation

Given an input text x, we first reconstruct a potential prompt using an LLM. The inferred prompt is concatenated with x to form a prompt-guided context $\tilde{x}$. Two vocabulary-heterogeneous models then compute the probability distribution vectors of each token at their corresponding positions for both x and $\tilde{x}$, which are used for subsequent probability correction.

3.3 Correction Coefficient Calculation

The correction in our framework adjusts the log-probability of each token according to the disagreement between two language models. This disagree-

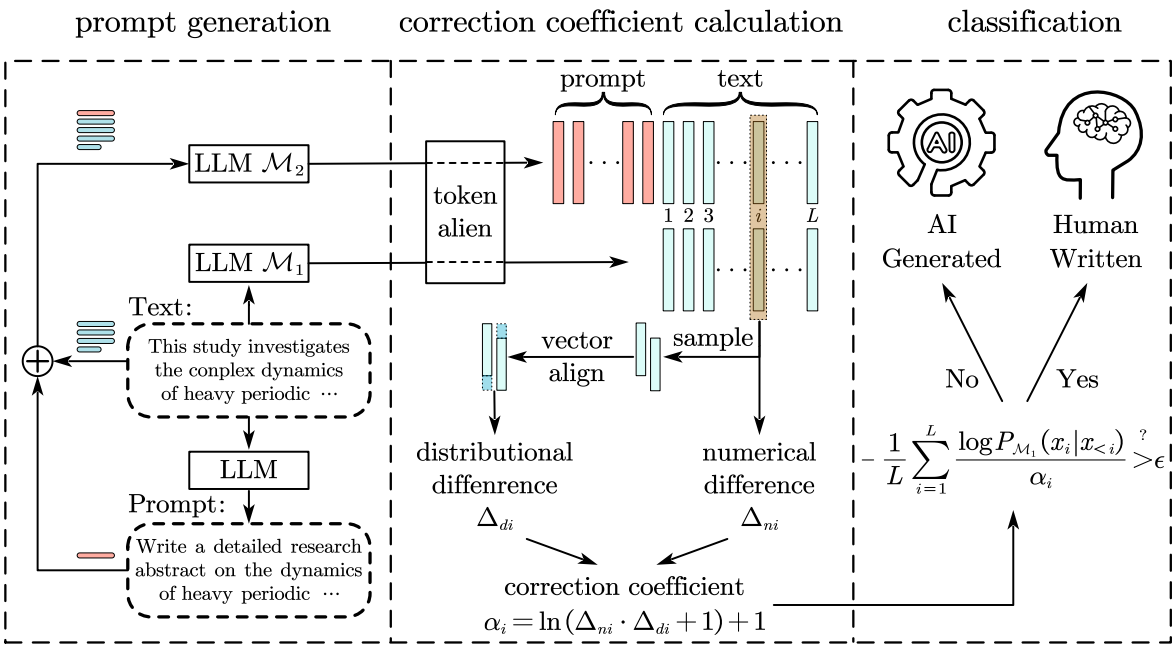

Figure 2: Overall framework of ProCDetect, consisting of prompt generation, correction coefficient calculation, and classification. M_1 and M_2 denote two LLMs with heterogeneous vocabularies; Δ_{di} and Δ_{ni} represent the distributional difference and numerical difference; α_i is the derived correction coefficient for the i-th token; and ϵ serves as the decision threshold for distinguishing between AI-generated and human-written texts.

ment is measured from two perspectives: (1) the numerical difference between token log-probabilities predicted by the two models, and (2) the distributional difference between their probability vectors.

Since the two models use heterogeneous vocabularies and tokenization schemes, we first perform model alignment to enable comparable probability estimation. Specifically, we introduce two alignment steps: token alignment and probability vector alignment, which ensure consistent token positions and probability vector dimensions across models.

3.3.1 Token Align

To address heterogeneous tokenization, we align token sequences by merging shorter tokens into composite tokens according to the longest tokenization scheme, while preserving the total log-probability of the text. Finally, the following formula is used to calculate the composite token's log probability $\log p$:

$$\log p = \frac{n}{N}\sum_{i=1}^{L}\log p'_i, \tag{2}$$

where $\log p'$ is the log probability of the token to be merged, N the total number of tokens before text merging, n the number of tokens after text merging, and L the number of tokens contained in the composite token during the merging process.

3.3.2 Probability Vector Align

To compare probability distributions from vocabulary-heterogeneous models, considering the next-token probability numerical concentration (Holtzman et al., 2020), we perform a sampling-based alignment strategy to construct vectors with consistent dimensions and one-to-one token correspondence. Specifically, we sample tokens based on their probabilities from one model and retrieve the corresponding probabilities from the other model.

Let $\{T_i\}_{i=1}^{k}$ be tokens sampled from the vocabulary of $\mathcal{M}_1$ that are absent from $\mathcal{M}_2$ and share the same first token T_0 after re-tokenization under $\mathcal{M}_2$. In this case, we approximate their probabilities using the substitute token T_0 and allocate its probability mass proportionally according to the probabilities in the source model:

$$p_{M_2}(T_i) = p_{M_2}(T_0)\frac{p_{M_1}(T_i)}{\sum_{i=1}^{k} p_{M_1}(T_i)}. \tag{3}$$

This procedure yields aligned probability vectors with unified dimensions and one-to-one token correspondence, enabling the computation of cross-model distributional differences.

3.3.3 Correction Coefficient

We compute the correction coefficient for each token based on two components. The first is the cross-entropy between the probability vectors aligned via the sample-alignment process, and the second is the absolute difference in the models' probabilities for the target token.

We represent the input text as $x = (x_1, \ldots, x_L)$, where x_i is the i-th token of the text. The prompt is represented as $p = (p_1, \ldots, p_h)$, where p_i is the i-th token in the prompt. We concatenate the prompt and the text as $\tilde{x} = p \oplus x = (p_1, \ldots, p_h, x_1, \ldots, x_L)$. Then, the calculation for the log probability correction coefficient of x_i in the text is defined as follows:

$$\Delta_{ni} = \Big| \log p_{\mathcal{M}_1}(x_i|x_{<i}) - \log p_{\mathcal{M}_2}(\tilde{x}_{i+h}|\tilde{x}_{<i+h}) \Big|, \quad (4)$$

$$\Delta_{di} = -\mathbf{p}_{\mathcal{M}_1}(x)_i \cdot \log \mathbf{p}_{\mathcal{M}_2}(\tilde{x})_{i+h}, \quad (5)$$

$$\alpha_i = \log(\Delta_{ni}\Delta_{di} + 1) + 1. \quad (6)$$

where $p_{\mathcal{M}}(x_i|x_{<i})$ denotes the conditional probability of the i-th token x_i given the prefix $x_{<i}$ under model $\mathcal{M}$. The term $\mathbf{p}_{\mathcal{M}}(x)_i$ represents the aligned probability distribution vector at position i generated by model $\mathcal{M}$ (after sampling and alignment). Δ_{ni} and Δ_{di} quantify the numerical and distributional differences between models $\mathcal{M}_1$ and $\mathcal{M}_2$ for token x_i. The coefficient α_i, derived from Δ_{ni} and Δ_{di}, is utilized to correct the log-probability predictions of $\mathcal{M}_1$.

3.4 Classification

Given a text $x = (x_1, \ldots, x_L)$, we compute its corrected likelihood by reweighting each token's log-probability using a token-specific correction coefficient $\alpha = (\alpha_1, \ldots, \alpha_L)$:

$$\mathrm{CL}(x) = -\frac{1}{L}\sum_{i=1}^{L} \frac{\log p_{\mathcal{M}_1}(x_i \mid x_{<i})}{\alpha_i}. \quad (7)$$

The resulting metric $\mathrm{CL}(x)$ represents the likelihood of the text after correction. To perform classification, we compare $\mathrm{CL}(x)$ against a predefined threshold ϵ. Texts with $\mathrm{CL}(x) < \epsilon$ are classified as machine-generated, whereas those with $\mathrm{CL}(x) \geq \epsilon$ are classified as human-written.

4 Experiments

4.1 Settings

4.1.1 Datasets

In this study, 200 human-written texts were randomly selected from three public datasets, namely Arxiv[1], Writing Prompt (Fan et al., 2018), and XSum (Narayan et al., 2018). Using the titles and prompt information of these human texts as generation constraints, we employed four language models, including three closed-source LLMs (Claude 3.5 Sonnet (Anthropic, 2024), GPT-4o (OpenAI et al., 2024), and Gemini 2.0 Flash (Comanici et al., 2025)) and one open-source LLM (Phi-4 (Abdin et al., 2024)), to generate machine-generated texts that matched the lengths and types of the human texts. Based on the aforementioned process, 12 experimental datasets were finally constructed, and the number of human texts and machine-generated texts in each dataset was kept equal.

4.1.2 Evaluation Metrics

To systematically evaluate the effectiveness and robustness of our proposed method, we conducted three sets of experiments: 1) Baseline Comparisons to assess standard detection performance against state-of-the-art methods; 2) Out-of-Distribution (OOD) Testing to verify the method's generalization capability on unseen generative models, which simulates real-world scenarios where the specific generation source is unknown; and 3) Ablation and Sensitivity Studies to analyze the contribution of individual components and parameter stability. Accordingly, we employed specific metrics tailored to these experimental objectives.

Following prior work, we adopted the Area Under the Receiver Operating Characteristic Curve (AUROC) for the baseline comparisons to provide a threshold-independent evaluation of detection performance. For OOD evaluation and sensitivity studies, we adopted the F1-score, which reflects the balance between precision and recall under a fixed decision threshold.

4.1.3 Baselines and Model Configuration

We compare our approach with a diverse set of representative detectors, including zero-shot methods (Likelihood (Solaiman et al., 2019), Rank (Gehrmann et al., 2019), LogRank (Gehrmann et al., 2019), Entropy (Lavergne

[1] https://www.kaggle.com/datasets/spsayakpaul/arxiv-paper-abstracts/data

Table 1: Comparison of AUROC scores between the proposed method and baselines.

Method	Claude-3.5-sonnet			GPT-4o			Gemini-2.0-flash			Phi-4		
	Arxiv	Writing	Xsum	Arxiv	Writing	Xsum	Arxiv	Writing	Xsum	Arxiv	Writing	Xsum
RoBERTa-Base	0.7127	0.7531	0.6593	0.8411	0.9286	0.5217	0.6631	0.7665	0.8243	0.9431	0.6500	0.9636
Openai-detector	0.5024	0.4362	0.6327	0.5278	0.5653	0.7235	0.5348	0.4600	0.7523	0.5317	0.6457	0.8084
RADAR	0.6307	0.5028	**0.9921**	0.9024	0.7050	**1.0000**	0.8812	0.7010	**0.9999**	0.9550	0.8325	**1.0000**
Likelihood	0.7987	0.7960	0.8546	0.8180	0.9667	0.9362	0.9534	0.9388	0.9582	0.8414	0.9712	0.9306
Rank	0.7494	0.6439	0.8719	0.7979	0.8664	0.8996	0.8720	0.6968	0.8725	0.7746	0.9140	0.9286
Log-Rank	0.8160	0.7679	0.8706	0.8253	0.9539	0.9425	0.9605	0.9134	0.9651	0.8529	0.9657	0.9435
Entropy	0.6114	0.7434	0.6899	0.5741	0.8421	0.6848	0.7171	0.7385	0.7232	0.5236	0.7706	0.6033
LRR	0.8125	0.6589	0.8721	0.7981	0.8704	0.9175	0.9476	0.7741	0.9542	0.8374	0.9265	0.9401
NPR	0.7742	*0.9223*	0.8169	0.6220	0.9639	0.7914	0.8631	0.9756	0.8775	0.6277	0.9611	0.7651
DetectGPT	0.7083	0.8643	0.6916	0.6295	0.8645	0.6442	0.7160	0.8566	0.7360	0.5675	0.8834	0.7037
Fast-DetectGPT	0.9192	0.8020	0.9014	*0.9534*	*0.9852*	0.9704	*0.9964*	*0.9876*	0.9892	0.9752	*0.9952*	0.9919
Binoculars	*0.9245*	0.5484	0.6615	0.9494	0.8922	0.9015	0.9786	0.9118	0.9020	*0.9905*	0.9895	0.9705
Ours	**0.9763**	**0.9753**	*0.9901*	**0.9965**	**0.9991**	*0.9996*	**0.9995**	**0.9982**	*0.9989*	**1.0000**	**0.9996**	*0.9997*
Relative↑	*5.60%*	*5.75%*	*9.84%*	*4.52%*	*1.41%*	*3.01%*	*0.31%*	*1.07%*	*0.98%*	*0.96%*	*0.44%*	*0.79%*

et al., 2008), LRR (Su et al., 2023), NPR (Su et al., 2023), DetectGPT (Mitchell et al., 2023), Fast-DetectGPT (Bao et al., 2024), and Binoculars (Hans et al., 2024)) and supervised detectors (OpenAI Detector (Solaiman et al., 2019), RADAR (Hu et al., 2023), and a RoBERTa-Base classifier (Park et al., 2021)).

For our method, token log-probabilities are computed using Falcon3-3B-Instruct, while correction coefficients are estimated using Falcon3-3B-Instruct and Llama3.2-3B-Instruct. Prompt reconstruction is performed using the GPT-3.5-Turbo API.

For methods that require text perturbation, such as DetectGPT and NPR, we used the T5-Base model to generate perturbed texts. For Binoculars, we employed Falcon3-3B-Base and Falcon3-3B-Instruct as paired models to perform same-scale comparisons. All other zero-shot baselines were computed using GPT-Neo-2.7B to ensure consistency. For the OpenAI Detector and RADAR, we directly employed the publicly released models. The RoBERTa-Base Classifier was trained individually on each of the twelve datasets to serve as a supervised baseline for the corresponding domain.

4.2 Baseline Comparison

We evaluated our method on twelve datasets (Section 4.1.1) spanning diverse text domains and generation sources. The results, measured by AUROC, are summarized in Table 1, where bold numbers denote the best results, italic numbers the second best, and the last row indicates the relative improvement over the top zero-shot baseline.

Our method consistently achieved the highest AUROC among all zero-shot baselines across every dataset, with an average AUROC of 0.9944 compared to 0.9556 for the second-best zero-shot method. This corresponded to an average relative improvement of 4.06%, and up to 9.84% on individual datasets, highlighting both the robustness and strong detection capability of our approach in the absence of task-specific training data.

Fig. 3 further visualizes the distribution of corrected likelihood scores for human-written and machine-generated texts on the representative XSum dataset. The two distributions exhibit clear separation with minimal overlap across all source models, explaining the strong detection performance observed in Table 1.

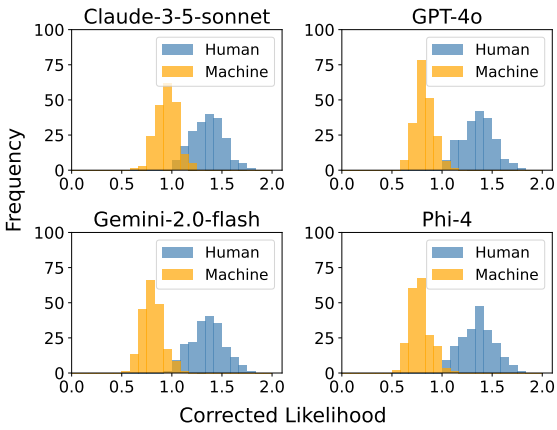

Figure 3: The corrected likelihood distribution of machine-generated texts (produced by four LLMs from the Xsum dataset) and human-written texts. The distinct separation between the two histograms demonstrates the effectiveness of our metric in distinguishing generated content.

Compared with supervised detectors, some methods such as RADAR achieve competitive performance on specific datasets but show noticeable variance across domains. In contrast, our zero-shot

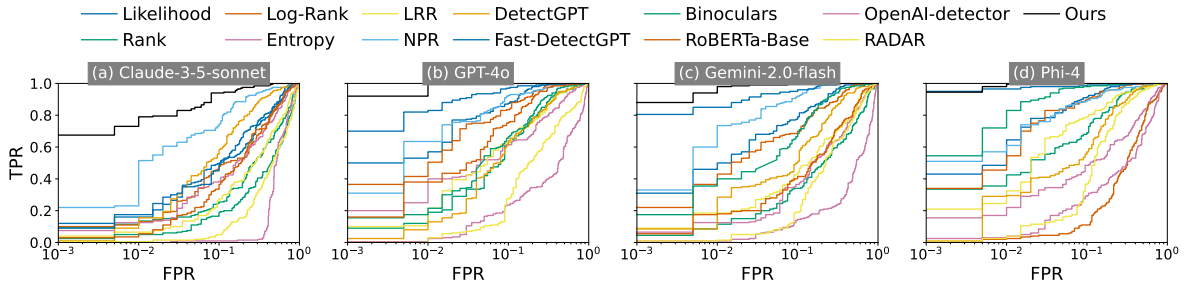

Figure 4: ROC curves of our method and all the baselines.

approach maintains consistently strong results, indicating better robustness and domain generalization.

Finally, the ROC curves in Fig. 4 show that our method achieves higher true positive rates under very low false positive rates, which is particularly important for high-precision detection scenarios.

4.3 Out of Distribution Evaluation

We further evaluate our method under cross-model OOD settings, where the detector is applied to texts generated by unseen LLMs within the same domain. This setting reflects realistic scenarios in which the exact generation model is unknown.

For zero-shot detectors and supervised models with released checkpoints, the decision threshold is first determined on one dataset and then directly applied to datasets generated by three other LLMs in the same domain. For fine-tuned detectors, models are trained on one dataset and evaluated on the remaining datasets within the same domain.

Table 2 reports the F1 scores under OOD settings. Our method consistently achieves the best performance across all datasets. On average, it improves the F1 score by 4.18% over the strongest baseline, with the largest gain reaching 7.22%.

These results demonstrate that the proposed probability correction mechanism effectively mitigates model-specific biases and enables reliable detection of texts generated by previously unseen LLMs.

4.4 Ablation and Sensitivity Study

We conduct ablation and sensitivity experiments to analyze the contribution of each component in ProCDetect, including the correction coefficients, model combinations, and prompt sources.

Effect of numerical difference and distributional difference. We first investigate the contributions of the numerical difference (Δ_n) and distributional difference (Δ_d). As shown in Table 3, applying both coefficients yields the best performance with an average AUROC of 0.9761. Using only Δ_n still achieves strong results (0.9715), while using only Δ_d leads to a noticeable performance drop (0.9516). This suggests that numerical difference serves as the primary driver of performance gains, because the absolute difference of log-probabilities provides a direct signal of model disagreement at the token level. In contrast, distributional difference plays a more complementary role by capturing subtler shifts in the sampled probability vector, allowing it to highlight local distributional changes that are not fully reflected by numerical differences.

Effect of model combinations. To investigate the impact of inter-model diversity on detection performance, we evaluated our method under two types of model pairings: (1) heterogeneous pairs, where the two models differ significantly in architecture and vocabulary, and (2) homogeneous pairs, where the two models share the same vocabularies. As illustrated in Fig. 5, our quantitative analysis reveals that heterogeneous combinations consistently outperform homogeneous ones across all source models, achieving an average relative F1 improvement of 4.04%. This observation suggests that greater diversity between models enhances the effectiveness of log-probability correction, as heterogeneous pairs provide more complementary probability signals and expose model-specific biases more clearly. In contrast, homogeneous model pairs tend to produce more correlated probability distributions, which limits the corrective effect and reduces the sensitivity of our method to generated texts.

Effect of prompt sources. To examine the effect of prompt quality on detection performance, we replaced the prompt generator in our framework with three different models: GPT-3.5-turbo (API),

Table 2: Comparison of OOD F1 scores between the proposed method and baselines.

Method	Claude-3.5-sonnet			GPT-4o			Gemini-2.0-flash			Phi-4		
	Arxiv	Writing	Xsum	Arxiv	Writing	Xsum	Arxiv	Writing	Xsum	Arxiv	Writing	Xsum
Likelihood	0.8072	0.8066	0.7585	0.8002	0.8825	0.8316	0.8470	0.8304	0.8722	0.8324	0.8292	0.8452
Rank	0.7896	0.7791	0.7612	0.7795	0.7668	0.7341	0.7611	0.7244	0.8348	0.8361	0.8446	0.8250
Log-Rank	0.8177	0.8204	0.7773	0.8044	0.8660	0.8137	0.8226	0.8071	0.8848	0.8585	0.8336	0.8584
Entropy	0.6709	0.6674	0.6571	0.6674	0.7179	0.6960	0.7212	0.7246	0.6786	0.6820	0.6740	0.6850
LRR	0.7965	0.7972	0.7474	0.7941	0.7925	0.7554	0.7762	0.7386	0.8474	0.8554	0.8361	0.8468
NPR	0.6814	0.6963	0.5179	0.6962	*0.8962*	*0.8808*	*0.8812*	*0.8700*	0.7505	0.7462	0.7211	0.7630
DetectGPT	0.6661	0.6912	0.6111	0.6927	0.8006	0.7998	0.8053	0.7863	0.7002	0.7037	0.6703	0.6978
Fast-DetectGPT	*0.8861*	0.8864	0.8231	0.8615	0.8440	0.8069	0.8046	0.7530	0.8730	0.9071	0.8801	0.8525
Binoculars	0.8759	*0.9011*	*0.8850*	*0.8698*	0.8706	0.8126	0.8039	0.7734	0.8600	0.8138	0.8152	0.7825
RoBERTa-Base	0.7989	0.7450	0.7277	0.8150	0.7333	0.7954	0.6733	0.6832	0.7040	0.6667	0.8181	0.8674
Openai-detector	0.6099	0.5931	0.6099	0.6102	0.6517	0.6539	0.6509	0.6188	0.5914	0.6007	0.5407	0.2409
RADAR	0.7080	0.7095	0.7287	0.6377	0.6689	0.6603	0.6670	0.5725	*0.9732*	*0.9367*	*0.9594*	*0.9557*
Ours	**0.9324**	**0.9386**	**0.9236**	**0.9305**	**0.9502**	**0.9414**	**0.9448**	**0.9137**	**0.9744**	**0.9642**	**0.9650**	**0.9621**
Relative↑	*5.23%*	*4.16%*	*4.36%*	*6.98%*	*6.03%*	*6.88%*	*7.22%*	*5.02%*	*0.12%*	*2.94%*	*0.58%*	*0.67%*

Table 3: AUROC results of ablation study.

Method	Arxiv	Writing Prompt	Xsum	Mean
Δ_n only	*0.9673*	*0.9705*	*0.9766*	*0.9715*
Δ_d only	0.9619	0.9415	0.9514	0.9516
No Correction	0.9481	0.9367	0.9356	0.9401
Apply both difference	**0.9706**	**0.9752**	**0.9825**	**0.9761**

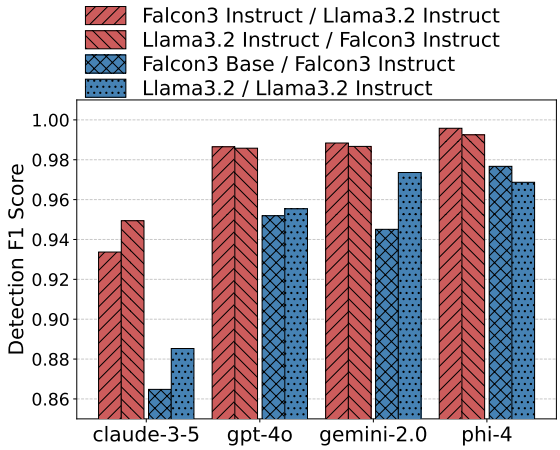

Figure 5: Detection F1 scores of different model combinations. The figure compares heterogeneous and homogeneous model pairings under different source language models.

Falcon3-3B-Instruct, and Falcon3-7B-Instruct. The results, summarized in Fig. 6, indicate that the choice of prompt source has only a limited impact on overall performance, suggesting that our method is relatively robust to variations in prompt quality. Although smaller models such as Falcon3-3B-Instruct consistently yielded the lowest F1 scores, larger models generate more informative prompts and lead to marginally better detection performance.

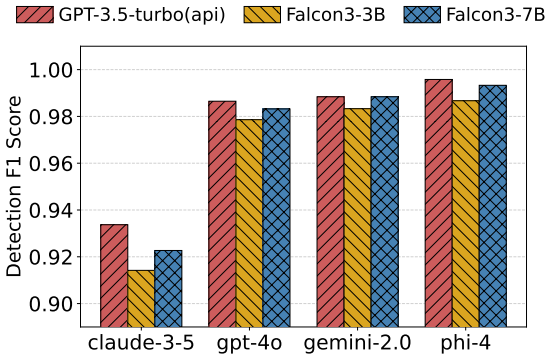

Figure 6: Detection F1 scores of different prompt sources.

5 Conclusion

This study introduces ProCDetect, a zero-shot detection method that unifies vocabulary-heterogeneous model alignment with prompt-guided probability correction, enabling reliable discrimination between machine-generated and human-written texts without any training data. Extensive experiments across multiple datasets and generation sources demonstrate the superior performance and robustness of our approach. Specifically, it achieves up to 5.75% relative improvement over the best-performing baseline and up to 7.22% in cross-model out-of-distribution settings. These results not only highlight the strong generalization ability of ProCDetect but also indicate that it serves as a promising lightweight alternative in the evolving field of LLM text detection.

Limitations

Despite its effectiveness, ProCDetect still has several limitations. First, the detection performance

remains relatively sensitive to the quality and format of prompt words, and its robustness may fluctuate across different fields. Second, the probability vector alignment process requires continuous bidirectional mapping between tokens and characters across models, which significantly increases the computational cost. Furthermore, the experiments are mainly conducted based on English datasets, and the cross-lingual generalization ability still needs further verification.

There are also potential risks associated with the practical deployment of AI-generated text detectors. False positives may cause harm in high-stakes settings, and users may over-rely on automatic detection results despite the inherent uncertainty of the task. In addition, adversaries could use detection methods to probe detector behavior and develop evasion strategies. Therefore, ProCDetect should be used as an assistive tool rather than as the sole basis for punitive or high-stakes decisions.

Based on these limitations, future research can be expanded in the following directions:

1. Explore the zero-shot detection capability in multilingual scenarios to enhance the cross-lingual adaptability of the method.

2. Design a more efficient probability vector alignment algorithm to reduce computational overhead.

3. Construct a generalized and transferable prompt generation framework to provide more stable support for log probability correction, thereby improving the overall detection performance.

4. Investigating safer deployment protocols that account for uncertainty, false-positive risks, and potential adversarial misuse.

References

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J. Hewett, Mojan Javaheripi, Piero Kauffmann, James R. Lee, Yin Tat Lee, Yuanzhi Li, Weishung Liu, Caio C. T. Mendes, Anh Nguyen, Eric Price, Gustavo de Rosa, Olli Saarikivi, and 8 others. 2024. Phi-4 technical report. *Preprint*, arXiv:2412.08905.

Anthropic. 2024. Model card addendum: Claude 3.5 haiku and upgraded claude 3.5 sonnet.

Anton Bakhtin, Sam Gross, Myle Ott, Yuntian Deng, Marc'Aurelio Ranzato, and Arthur Szlam. 2019. Real or fake? learning to discriminate machine from human generated text. *Preprint*, arXiv:1906.03351.

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*, pages 5046–5068, Vienna, Austria.

Amrita Bhattacharjee, Tharindu Kumarage, Raha Moraffah, and Huan Liu. 2023. ConDA: Contrastive domain adaptation for AI-generated text detection. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 598–610, Nusa Dua, Bali. Association for Computational Linguistics.

Yutian Chen, Hao Kang, Vivian Zhai, Liangze Li, Rita Singh, and Bhiksha Raj. 2023. Gpt-sentinel: Distinguishing human and chatgpt generated content. *Preprint*, arXiv:2305.07969.

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 3416 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *Preprint*, arXiv:2507.06261.

Long Dai, Jiarong Mao, Xuefeng Fan, and Xiaoyi Zhou. 2022. Deephider: A covert nlp watermarking framework based on multi-task learning. *Preprint*, arXiv:2208.04676.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

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.

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.

Abhimanyu Hans, Avi Schwarzschild, Valeriia Cherepanova, Hamid Kazemi, Aniruddha Saha, Micah Goldblum, Jonas Geiping, and Tom Goldstein. 2024. Spotting llms with binoculars: Zero-shot detection of machine-generated text. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 17519–17537, Vienna, Austria.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *8th International Conference on Learning Representations*, Addis Ababa, Ethiopia. OpenReview.net.

Xiaomeng Hu, Pin-Yu Chen, and Tsung-Yi Ho. 2023. RADAR: robust ai-text detection via adversarial learning. In *Advances in Neural Information Processing Systems*, volume 36, pages 15077–15095, New Orleans, LA, USA. Curran Associates, Inc.

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.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. 2023. A watermark for large language models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 17061–17084, Honolulu, HI, USA.

Thomas Lavergne, Tanguy Urvoy, and François Yvon. 2008. Detecting fake content with relative entropy scoring. In *Proceedings of the 2008 International Conference on Uncovering Plagiarism, Authorship and Social Software Misuse - Volume 377*, pages 27–31, Patras, Greece. CEUR-WS.org.

Taehyun Lee, Seokhee Hong, Jaewoo Ahn, Ilgee Hong, Hwaran Lee, Sangdoo Yun, Jamin Shin, and Gunhee Kim. 2024. Who wrote this code? watermarking for code generation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4890–4911, Bangkok, Thailand. Association for Computational Linguistics.

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. *Preprint*, arXiv:1907.11692.

Ning Lu, Shengcai Liu, Rui He, Yew-Soon Ong, Qi Wang, and Ke Tang. 2024. Large language models can be guided to evade AI-generated text detection. *Transactions on Machine Learning Research*, 2024:1–29.

Chengzhi Mao, Carl Vondrick, Hao Wang, and Junfeng Yang. 2024. Raidar: generative AI detection via rewriting. In *The Twelfth International Conference on Learning Representations*, pages 4330–4347, Vienna, Austria.

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*, volume 202 of *Proceedings of Machine Learning Research*, pages 24950–24962, Honolulu, HI, USA. PMLR.

Sandra Mitrović, Davide Andreoletti, and Omran Ayoub. 2023. Chatgpt or human? detect and explain. explaining decisions of machine learning model for detecting short chatgpt-generated text. *Preprint*, arXiv:2301.13852.

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, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, and 262 others. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

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.

Choonghyun Park, Hyuhng Joon Kim, Junyeob Kim, Youna Kim, Taeuk Kim, Hyunsoo Cho, Hwiyeol Jo, Sang goo Lee, and Kang Min Yoo. 2024. Investigating the influence of prompt-specific shortcuts in ai generated text detection. *Preprint*, arXiv:2406.16275.

Sungjoon Park and 1 others. 2021. Klue: Korean language understanding evaluation. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, New Orleans, LA, USA.

Jiameng Pu, Zain Sarwar, Sifat Muhammad Abdullah, Abdullah Rehman, Yoonjin Kim, Parantapa Bhattacharya, Mobin Javed, and Bimal Viswanath. 2023.

Deepfake text detection: Limitations and opportunities. In *2023 IEEE Symposium on Security and Privacy (SP)*, pages 1613–1630, San Francisco, CA, USA.

Juan Diego Rodriguez, Todd Hay, David Gros, Zain Shamsi, and Ravi Srinivasan. 2022. Cross-domain detection of GPT-2-generated technical text. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1213–1233, Seattle, United States. Association for Computational Linguistics.

Vinu Sankar Sadasivan, Aounon Kumar, Sriram Balasubramanian, Wenxiao Wang, and Soheil Feizi. 2025. Can AI-generated text be reliably detected? stress testing AI text detectors under various attacks. *Transactions on Machine Learning Research*, 2025:1–37.

Kouichi Sakurai, Kaito Taguchi, and Yujie Gu. 2024. The impact of prompts on zero-shot detection of ai-generated text. In *Proceedings of the IJCAI 2024 Workshop on Artificial Intelligence Safety (AISafety 2024) co-located with the 33rd International Joint Conference on Artificial Intelligence (IJCAI 2024)*, volume 3856 of *CEUR Workshop Proceedings*, Jeju, Korea. CEUR-WS.org.

Zhouxing Shi, Yihan Wang, Fan Yin, Xiangning Chen, Kai-Wei Chang, and Cho-Jui Hsieh. 2024. Red teaming language model detectors with language models. *Transactions of the Association for Computational Linguistics*, 12:174–189.

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. *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, Singapore. Association for Computational Linguistics.

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. In *Advances in Neural Information Processing Systems*, volume 36, pages 39257–39276, New Orleans, LA, USA. Curran Associates, Inc.

Adaku Uchendu, Thai Le, Kai Shu, and Dongwon Lee. 2020. Authorship attribution for neural text generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8384–8395, Online. Association for Computational Linguistics.

Honai Ueoka, Yugo Murawaki, and Sadao Kurohashi. 2021. Frustratingly easy edit-based linguistic steganography with a masked language model. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5486–5492, Online. Association for Computational Linguistics.

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.

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.

Quan Wang, Licheng Zhang, Zikang Guo, and Zhendong Mao. 2024. IDEATE: Detecting AI-generated text using internal and external factual structures. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 8556–8568, Torino, Italia. ELRA and ICCL.

Kangxi Wu, Liang Pang, Huawei Shen, Xueqi Cheng, and Tat-Seng Chua. 2023. LLMDet: A third party large language models generated text detection tool. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 2113–2133, Singapore. Association for Computational Linguistics.

Peipeng Yu, Jiahan Chen, Xuan Feng, and Zhihua Xia. 2025. Cheat: A large-scale dataset for detecting chatgpt-written abstracts. *IEEE Transactions on Big Data*, 11(3):898–906.

Xiao Yu, Yuang Qi, Kejiang Chen, Guoqiang Chen, Xi Yang, Pengyuan Zhu, Xiuwei Shang, Weiming Zhang, and Nenghai Yu. 2024. DPIC: decoupling prompt and intrinsic characteristics for LLM generated text detection. In *Advances in Neural Information Processing Systems*, volume 37, pages 16194–16212, Vancouver, BC, Canada. Curran Associates, Inc.

Biru Zhu, Lifan Yuan, Ganqu Cui, Yangyi Chen, Chong Fu, Bingxiang He, Yangdong Deng, Zhiyuan Liu, Maosong Sun, and Ming Gu. 2023. Beat LLMs at their own game: Zero-shot LLM-generated text detection via querying ChatGPT. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7470–7483, Singapore. Association for Computational Linguistics.