

Adaptive Boundaries: Context-Aware Detection for Synthetic Text

Haorong Li^{1†}, Ruitao Ni^{1†}, Haihui Yang¹, Shipin Gao¹, Yang Liu^{1*}, Xiaojun Quan^{1,2*}

¹School of Computer Science and Engineering, Sun Yat-sen University

²Shenzhen Loop Area Institute

{lihr86, nirt, yanghh29, gaosp}@mail2.sysu.edu.cn

liuy856@mail.sysu.edu.cn

xiaojunquan@slai.edu.cn

Abstract

The widespread adoption of large language models (LLMs) has created an urgent need for accurate and robust detection of AI-generated text. However, existing statistical detectors often rely on a fixed decision boundary, which hinders their ability to generalize across topics, styles, and model variants. In this paper, we propose preference calibration with context-aware thresholding (PC-CAT), a novel framework that integrates preference calibration with context-aware thresholding to improve detection reliability. PC-CAT first applies Simple Preference Optimization (SimPO) to maximize the log-probability gap between paired human-written and AI-generated texts, enhancing discriminability. It then tailors instance-specific thresholds using a lightweight regressor trained with a hybrid centering and ordinal objective. Experiments across diverse domains and generators show that PC-CAT consistently outperforms strong baselines and achieves new state-of-the-art results. Moreover, it exhibits strong robustness under distribution shifts and adversarial attacks. These findings highlight the critical role of adaptive thresholding in advancing the robustness of AI-generated text detectors.

1 Introduction

The remarkable advancement of large language models (LLMs) such as GPT-4 and ChatGPT, has made AI-generated text frequently indistinguishable from human writing (Brown et al., 2020; Wei et al., 2022; OpenAI, 2022; Chowdhery et al., 2023; Touvron et al., 2023a,b; Achiam et al., 2023; Bi et al., 2024). This development has sparked escalating concerns about potential abuses of LLMs, particularly in academic integrity violations and the proliferation of misinformation. Consequently, the development of robust detection methods is im-

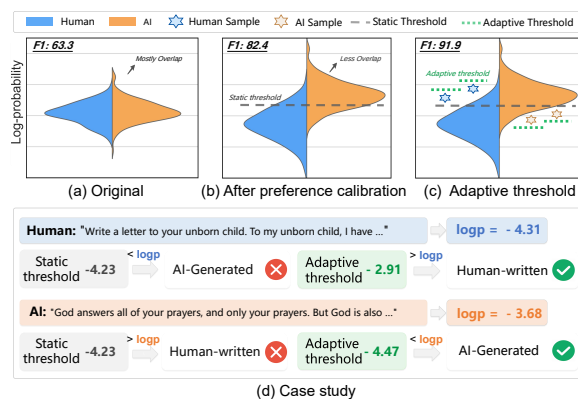


Figure 1: Illustration of preference calibration and dynamic thresholding. (a–b) Log-probability distributions of human-written and AI-generated texts before and after preference calibration. (c) Adaptive thresholding predicts a threshold for each input. (d) Adaptive thresholding enables correct classification with an instance-specific threshold, whereas the static method fails.

perative to mitigate these risks and safeguard the integrity of digital information ecosystems.

Existing methods for AI-generated text detection generally fall into two categories: training-based methods and statistical methods. Training-based methods, which detect AI-generated text by learning a classifier (Guo et al., 2023; Hu et al., 2023; Guo and Yu, 2023), are often prone to overfitting, especially in the presence of limited training data or distributional imbalance (Mosbach et al., 2021; Wang et al., 2024). Statistical methods, on the other hand, perform zero-shot detection by leveraging statistical metrics such as token-level log-probability, log-rank, and entropy (Gehrmann et al., 2019; Su et al., 2023; Bao et al., 2024; Yang et al., 2024b). These methods are based on the hypothesis that AI-generated text tends to have higher probability, as language models prefer high-probability tokens during generation. In contrast, human-written text is typically more diverse in structure and vocabulary, leading to lower overall probabilities (Holtzman et al., 2020; He et al., 2024).

While promising, current statistical approaches

[†]Equal contribution.

^{*}Corresponding authors.

face two fundamental limitations. First, they generally underperform compared to training-based approaches, primarily because the untrained scoring model struggles to ensure a significant difference in the probabilities assigned to AI-generated and human-written text (see Figure 1(a)). Second, they rely heavily on threshold calibration. Typically, a single optimal threshold is selected on a validation set and then applied uniformly to all test samples during inference. However, we observe that this static, one-size-fits-all thresholding strategy often fails to make accurate predictions across diverse inputs (see Figure 1(c)). The limitation becomes especially pronounced in cross-domain detection scenarios, where optimal thresholds can vary substantially across domains.

To overcome these limitations, we propose *Preference Calibration with Context-Aware Thresholding (PC-CAT)*, a novel dynamic thresholding framework that adaptively tailors instance-specific thresholds for AI-generated text detection. Our method combines the strengths of both training-based and statistical features, and comprises two main steps. To tackle the first limitation, we incorporate a *preference calibration* step that encourages the scoring model to favor AI-generated text while rejecting human-written text via preference optimization. This step guides the model to maximize the log-probability gap between the two text types, thereby enhancing its discriminative capacity (see Figure 1(b)). To address the second limitation, we further introduce a *context-aware thresholding* mechanism to adaptively determine decision boundaries for different instances. Specifically, we add a lightweight linear layer on top of the scoring model to predict a threshold tailored to each input. A dedicated training objective is designed to optimize this module for accurate threshold prediction. At inference time, detection is performed by comparing the log-probability score of a given text against its predicted threshold. This dynamic thresholding strategy allows the model to assign instance-specific decision boundaries, enabling more accurate classification and leading to improved overall detection performance, as illustrated in Figure 1(c) and 1(d).

We conduct extensive experiments on both AI-generated and AI-polished texts across diverse domains and model families. Results show that our method consistently outperforms existing approaches, achieving new state-of-the-art performance. Ablation studies validate the effectiveness

of each component, while further analysis confirms strong robustness under out-of-distribution (OOD) conditions and adversarial perturbations.

Our primary contributions are as follows:

- We incorporate preference calibration that explicitly maximizes the log-probability gap between human-written and AI-generated texts, better aligning with the detection objective.
- We propose context-aware thresholding that learns adaptive, instance-specific decision boundaries for fine-grained detection and better generalization under distribution shifts.
- Our method achieves superior detection performance across diverse domains, models, and revision tasks, and shows strong robustness to both OOD inputs and adversarial attacks.

2 Related Work

2.1 AI-Generated Text Detection

Existing methods for AI-generated text detection generally fall into two categories: training-based and statistical approaches. Training-based methods rely on supervised learning, using labeled corpora of human-written and AI-generated texts to train classification models. GPT-Sentinel (Chen et al., 2023a) builds a binary classifier on top of RoBERTa (Liu et al., 2019) and T5 (Raffel et al., 2020). RADAR (Hu et al., 2023) adopts an adversarial framework, jointly training a paraphraser and a detector. DPIC (Yu et al., 2024) decouples the prompt from the input and uses a Siamese network to learn intrinsic textual features for classification.

Statistical methods leverage pretrained language models to extract statistical features and apply thresholding for classification. GLTR (Gehrmann et al., 2019) analyzes word-level features such as predicted probabilities, ranks, and entropy. Fast-DetectGPT (Bao et al., 2024) estimates curvature directly from conditional probabilities, which avoids the need for perturbations. Raidar (Mao et al., 2024) computes edit distances between original and LLM-rewritten texts. ImBD (Chen et al., 2025) first enhances the scoring model using Direct Preference Optimization (DPO) (Rafailov et al., 2023) and then uses Fast-DetectGPT as the scoring function for detection.

2.2 Dynamic Thresholding

Determining appropriate classification thresholds remains a longstanding challenge in machine learn-

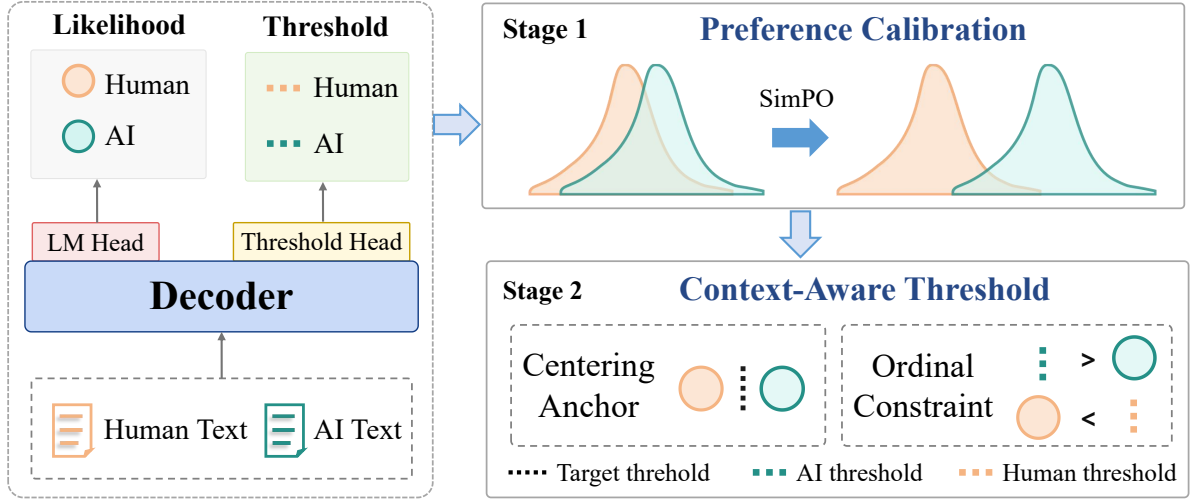


Figure 2: Overview of the PC-CAT framework. PC-CAT trains by first calibrating the preferences for human-written texts and AI-generated texts (Stage 1), then learns to tailor an adaptive threshold for each input (Stage 2).

ing, particularly for tasks involving imbalanced data or varying decision boundaries. To address this, several studies have proposed learnable thresholds, where the decision boundary is dynamically adjusted. For example, Xu et al. (2021) propose Dash, a semi-supervised learning algorithm that infers dynamic thresholds from labeled data. Hu et al. (2022a) introduce the Adaptive Threshold Selective Self-Attention (ATSSA) mechanism, which dynamically filters attention weights within a Transformer encoder. Yang and Emmert-Streib (2023) incorporate a learnable threshold function into a convolutional neural network (CNN) for multi-label text classification of electronic health records.

Unlike previous methods, we first introduce a preference calibration step that guides the scoring model to favor AI-generated text and reject human-written text through preference optimization. We then apply dynamic thresholding, allowing the model to adapt to input-specific variations in topic and style, and to predict adaptive thresholds for more fine-grained, accurate detection.

3 Method

Figure 2 provides an overview of the Preference Calibration with Context-Aware Thresholding (PC-CAT) framework. We describe it in detail below. We begin with preliminaries, and then describe the preference calibration step, which encourages the scoring model to favor AI-generated text while rejecting human-written text. Finally, we introduce the context-aware thresholding method.

3.1 Preliminary

Task definition AI-generated text detection is formulated as a binary classification task. Specifically, given an input text x , represented as a sequence of tokens $x = [w_1, \dots, w_T]$ of length T , the objective is to determine whether the input x is human-written (x_h) or AI-generated (x_a).

Foundation Statistical methods rely on a scoring model to extract probability-based metrics. These methods are grounded in the hypothesis that AI-generated text generally exhibits higher overall probability, as decoding strategies tend to select high-probability tokens during generation. In contrast, human-written text tends to show greater lexical diversity, resulting in lower overall probability.

In our approach, we use a scoring model $\mathcal{M}$, parameterized by θ , to compute the *average log-probability* of the tokens in a given text x , which serves as the scoring function:

$$\begin{aligned} \bar{p}(x) &= \frac{1}{T} \log p_{\theta}(x) \\ &= \frac{1}{T} \sum_{t=1}^T \log p_{\theta}(w_t | w_{1:t-1}), \end{aligned} \quad (1)$$

where $p_{\theta}(w_t | w_{1:t-1})$ denotes the probability of token w_t given its preceding context $w_{1:t-1}$.

The score $\bar{p}(x)$ is then compared with a decision threshold $\tau(x)$ to determine the predicted label:

$$\hat{y} = \begin{cases} 1 \text{ (AI-generated)} & \text{if } \bar{p}(x) \geq \tau(x) \\ 0 \text{ (Human-written)} & \text{otherwise} \end{cases}. \quad (2)$$

To achieve more accurate detection of AI-generated text, our approach comprises two key

components: (1) preference calibration, which improves the discriminative capacity of the scoring model between human-written and AI-generated text via preference optimization; and (2) context-aware thresholding, which incorporates a lightweight thresholding network trained to predict an adaptive, instance-specific decision threshold for each input. These components are detailed in the subsequent sections.

3.2 Preference Calibration

Uncalibrated scoring models often exhibit substantial overlap in the probability distributions assigned to human-written and AI-generated texts, making it challenging to distinguish between them effectively (see Figure 1(a)). To address this, we adopt a *preference calibration* step that encourages the scoring model to favor AI-generated text while rejecting human-written text. We utilize *Simple Preference Optimization (SimPO)* (Meng et al., 2024), a reference-free preference optimization method. This choice is particularly important in the context of AI-text detection, rather than merely serving as a technical replacement for DPO. DPO defines its implicit reward as $r_{\text{DPO}}(x) = \beta \log \frac{\pi_{\theta}(x)}{\pi_{\text{ref}}(x)}$, which normalizes the model’s likelihood by a reference model. Consequently, satisfying DPO’s preference ordering ($r_{\text{DPO}}(x_a) > r_{\text{DPO}}(x_h)$) does not guarantee that $\bar{p}(x_a) > \bar{p}(x_h)$ —our actual detection criterion—as the reference model ratio can distort the score ordering. In contrast, SimPO’s reward $r_{\text{SimPO}}(x) = \frac{\beta}{T} \log \pi_{\theta}(x)$ is directly proportional to $\bar{p}(x)$, so optimizing the SimPO objective is equivalent to directly maximizing the score separability required for detection. This ensures that every improvement during training translates to better detection at inference. We provide empirical evidence for this alignment advantage in Section 4.3 and further theoretical analysis in Appendix B.

Specifically, given pairs of human-written and AI-generated texts (x_h, x_a) from a preference dataset $\mathcal{D}$, we optimize the scoring model by maximizing the discrepancy in their log-probabilities using the SimPO objective:

$$\mathcal{L}_{\text{cal}} = -\mathbb{E}_{(x_h, x_a) \sim \mathcal{D}} \left[\log \sigma \left(\frac{\beta}{|x_a|} \log p_{\theta}(x_a) - \frac{\beta}{|x_h|} \log p_{\theta}(x_h) - \gamma \right) \right], \quad (3)$$

where σ is the sigmoid function, β is a scaling factor, and γ is a margin hyperparameter that enforces a minimum gap between the two average

log-probabilities. This optimization effectively enlarges the separation between the distributions of human-written and AI-generated texts while remaining faithful to the detection metric. As a result, the calibrated scoring model becomes more sensitive to their differences and thus more capable of accurately distinguishing between them.

3.3 Context-Aware Thresholding

Although preference calibration improves the separation between the scores $\bar{p}(x_h)$ and $\bar{p}(x_a)$ for text pairs, the value of $\bar{p}(x)$ can still vary significantly across different texts due to variations in topic or writing style. In such cases, using a static, fixed threshold is often insufficient for achieving accurate and fine-grained detection across diverse texts. To address this issue, we propose *Context-Aware Thresholding (CAT)*, which learns to predict an adaptive threshold tailored to each individual input. In the following, we first present the architecture of the threshold prediction network, and then detail its training objectives: a centering regression loss and an auxiliary ordinal constraint loss.

Threshold prediction network The calibrated scoring model encodes rich semantic and instance-specific stylistic features through preference calibration, making it a strong foundation for building a threshold prediction network.

Specifically, we extract the sequence of final-layer hidden states $\mathbf{H} = [\mathbf{h}_1, \dots, \mathbf{h}_T] \in \mathbb{R}^{T \times d}$ from the calibrated scoring model, where d is the hidden dimension and T is the sequence length. In order to aggregate global information, a global context embedding $\mathbf{c}(x)$ is then derived by applying average pooling over these hidden states:

$$\mathbf{c}(x) = \frac{1}{T} \sum_{t=1}^T \mathbf{h}_t. \quad (4)$$

This context vector $\mathbf{c}(x)$ is fed into a threshold prediction network f , implemented as a single-layer linear regressor parameterized by ϕ . The network maps $\mathbf{c}(x)$ to a scalar value, producing an input-adaptive threshold $\tau_{\phi}(x)$ for the given text:

$$\tau_{\phi}(x) = f(\mathbf{c}(x); \phi), \quad (5)$$

Centering anchor To train the threshold prediction network effectively, it is crucial to define a target threshold that facilitates reliable separation between human-written and AI-generated texts.

Given the paired texts (x_h, x_a) , the midpoint between their calibrated scores, i.e., $\bar{p}(x_h)$ and $\bar{p}(x_a)$, naturally serves as an optimal decision boundary in the score space. We therefore define the target threshold μ_{target} as:

$$\mu_{\text{target}} = \frac{\bar{p}(x_h) + \bar{p}(x_a)}{2}. \quad (6)$$

We encourage the predicted thresholds $\tau_\phi(x_h)$ and $\tau_\phi(x_a)$ to regress toward the midpoint target using a mean squared error objective. The resulting loss, referred to as the centering regression loss, is defined as:

$$\begin{aligned} \mathcal{L}_{\text{center}} = \mathbb{E}_{(x_h, x_a) \sim \mathcal{D}} & \left[\frac{1}{2} \left(\tau_\phi(x_h) - \mu_{\text{target}} \right)^2 \right. \\ & \left. + \frac{1}{2} \left(\tau_\phi(x_a) - \mu_{\text{target}} \right)^2 \right]. \end{aligned} \quad (7)$$

This loss encourages the predicted thresholds for both human-written and AI-generated texts to converge toward the midpoint of their respective scores, thereby promoting accurate classification under the decision rule defined in Eq. (2).

Auxiliary ordinal constraint However, the centering regression loss can fail in rare but critical cases—specifically, when the human-written text receives a higher score than the AI-generated one. In such cases, the midpoint μ_{target} becomes an invalid decision boundary: $\bar{p}(x_h) > \mu_{\text{target}}$ and $\bar{p}(x_a) < \mu_{\text{target}}$, resulting in misclassification under the decision rule in Eq. (2). To address this, we introduce an auxiliary ordinal constraint that guides the model to predict thresholds respecting the correct score ordering, particularly when the midpoint is not a reliable target. Rather than regressing toward a fixed target, the ordinal constraint loss allows the model to adaptively learn a threshold that maintains the desired ordinal relationship:

$$\begin{aligned} \mathcal{L}_{\text{ordinal}} = -\mathbb{E}_{(x_h, x_a) \sim \mathcal{D}} & \left[\log \sigma \left(\bar{p}(x_a) - \tau_\phi(x_a) \right) \right. \\ & \left. + \log \sigma \left(\tau_\phi(x_h) - \bar{p}(x_h) \right) \right], \end{aligned} \quad (8)$$

where σ is the sigmoid function. This loss enables robust threshold selection even in cases where $\bar{p}(x_h) > \bar{p}(x_a)$.

Overall objective By combining the centering regression and the ordinal constraint losses, we obtain the overall training objective for the threshold

prediction network:

$$\mathcal{L}_{\text{CAT}} = \mathbb{I}(\bar{p}(x_a) > \bar{p}(x_h)) \cdot \mathcal{L}_{\text{center}} + \alpha \cdot \mathcal{L}_{\text{ordinal}}, \quad (9)$$

where α is a hyperparameter that balances the two loss components. The indicator function $\mathbb{I}(\bar{p}(x_a) > \bar{p}(x_h))$ activates the centering loss only when the AI-generated text receives a higher score than the human-written text.

Unlike static thresholds, our context-aware thresholding mechanism offers superior flexibility in handling diverse inputs. We provide a deeper theoretical explanation in Appendix C.

4 Experiment

4.1 Experimental Setup

Datasets and metrics We conduct experiments on the ImBD dataset (Chen et al., 2025), which extends standard AI-generated text detection to the more challenging setting of detecting AI-revised text. The ImBD dataset covers a broad range of domains, model sources, and revision types. The training and validation sets contain 500 and 150 text pairs, respectively, where each pair consists of a human-written text and its corresponding AI-revised version generated by GPT-3.5. The data spans diverse domains such as academic papers, blogs, and emails. For evaluation, ImBD provides test sets from four distinct domains: XSum (Narayan et al., 2018), SQuAD (Rajpurkar et al., 2016), WritingPrompts (Fan et al., 2018) (abbreviated as “Writing”), and PubMedQA (Jin et al., 2019). The revised texts are generated by six LLMs and span four revision types: *rewrite*, *expand*, *polish* and *generate*. Each test subset contains 150 text pairs. Notably, the *generate* setting corresponds to the standard AI-generated text detection task. We adopt the F1 score as the primary evaluation metric, as it provides a balanced measure of precision and recall for evaluating detection performance. Furthermore, a comprehensive justification for selecting F1 over AUROC is detailed in Appendix D.

Baseline methods We compare our PC-CAT approach with both training-based and statistical baselines. The training-based baselines include classifiers trained on RoBERTa-base (Liu et al., 2019), RoBERTa-large, and the ImBD method (Chen et al., 2025). To ensure a rigorous and fair comparison, our experimental configuration, including implementation code, model weights, and hyperparam-

Method	GPT-3.5					GPT-4o				
	Writing	PubMed	SQuAD	XSum	Avg.	Writing	PubMed	SQuAD	XSum	Avg.
Likelihood	80.81	64.63	67.15	64.25	69.21	74.62	60.00	64.20	62.25	65.27
Rank	72.09	52.53	62.57	56.15	60.84	68.66	51.12	59.95	56.15	58.97
LogRank	74.81	64.73	66.67	64.50	67.68	72.15	61.73	64.15	63.87	65.48
Entropy	60.00	59.69	57.51	55.19	58.10	60.00	58.25	59.34	56.28	58.49
Fast-DetectGPT	86.32	59.23	74.32	66.88	71.69	72.22	47.50	66.90	52.22	59.71
Binoculars	89.01	49.29	80.43	64.10	70.71	74.70	29.95	60.00	65.98	57.66
ImBD	90.65	75.53	87.94	77.55	82.92	74.38	73.51	71.02	41.27	65.05
RoBERTa-base	85.71	78.79	86.85	97.96	87.33	85.04	82.78	85.80	92.47	86.52
RoBERTa-large	89.02	78.76	86.06	97.96	87.95	89.02	81.51	86.06	97.61	88.55
PC-CAT (Ours)	94.01	89.81	91.54	97.69	93.26	90.20	87.10	89.81	94.92	90.51

Table 1: Detection performance on AI-revised texts generated by closed-source models (GPT-3.5 and GPT-4o) in the *polish* task. Best results are shown in **bold**.

eter settings, strictly follows the protocol established in the ImBD (Chen et al., 2025). The statistical baselines include: Likelihood (Solaiman et al., 2019), Rank (Gehrmann et al., 2019), LogRank (Mitchell et al., 2023), Entropy (Gehrmann et al., 2019), Fast-DetectGPT (Bao et al., 2024), and Binoculars (Hans et al., 2024). The optimal thresholds for all baselines are selected using the Youden Index (Youden, 1950) on the validation set. For our approach, we use Qwen2.5-0.5B (Yang et al., 2024a) as the backbone of the scoring model. Extended empirical analysis on the log-probability signals across diverse generators is provided in Appendix G. Other implementation details are provided in Appendix A.

4.2 Main Results

Detection performance across models and domains Following Chen et al. (2025), we first evaluate our method on the *polish* task, where the polished texts are produced by various LLMs, including both closed-source and open-source models, across diverse domains. As AI-polished text tends to closely resemble human writing, this setting is substantially more challenging than standard AI-generated text detection.

Table 1 presents the results for closed-source models, including GPT-3.5 and GPT-4o. PC-CAT achieves the highest average F1 score in detecting both GPT-3.5- and GPT-4o-polished texts, outperforming all baselines. Specifically, it surpasses the strong RoBERTa-large baseline by 5.31% and 1.96% on GPT-3.5 and GPT-4o, respectively. Compared to ImBD, PC-CAT achieves even larger gains of 10.34% and 25.46%. Although both our method and ImBD incorporate preference optimization for

Method	Qwen2	Llama3	Mixtral	Deepseek	Avg.
Likelihood	55.04	71.34	66.15	70.93	65.87
Rank	59.84	63.51	57.61	68.14	62.28
LogRank	60.14	68.85	66.59	68.74	66.08
Entropy	38.11	51.02	48.09	59.34	49.14
Fast-DetectGPT	61.96	78.46	73.93	72.75	71.78
Binoculars	39.16	69.56	48.65	69.43	56.70
ImBD	66.20	83.92	71.45	77.56	74.78
RoBERTa-base	91.87	87.54	89.47	86.48	88.84
RoBERTa-large	92.49	90.75	92.15	90.21	91.40
PC-CAT (Ours)	94.75	94.41	94.19	92.20	93.89

Table 2: Detection performance on AI-revised texts generated by open-source models, including Qwen2-7B (Yang et al., 2024a), Mixtral-7B (Jiang et al., 2023), Deepseek-7B (Bi et al., 2024) and LLaMA-3-8B (Grattafiori et al., 2024) in the *polish* task. F1 scores are averaged over XSum, SQuAD, and WritingPrompts.

the scoring model, ImBD relies on a static threshold selected from the validation set. This makes it vulnerable to domain shifts and leads to significant performance fluctuations. Notably, RoBERTa-large performs better on XSum, likely due to overfitting on the imbalanced training set in which over 50% of the examples are drawn from the paper domain. In contrast, our dynamic thresholding approach promotes more consistent and robust detection across a wider range of domains.

We also observe a notable performance drop for most baselines when detecting GPT-4o-polished texts compared to GPT-3.5, likely due to GPT-4o’s improved ability to mimic human writing. Nevertheless, our method continues to achieve the best performance on GPT-4o, further demonstrating its robustness and effectiveness. Results for open-source models are also shown in Table 2. Similarly, our PC-CAT method achieves the highest average F1 score, outperforming ImBD by a sub-

Method	Tasks				Avg.
	Rewrite	Expand	Polish	Generate	
Likelihood	60.52	66.19	62.09	69.48	64.57
Rank	60.63	64.20	57.34	68.54	62.68
LogRank	62.98	65.98	63.88	67.82	65.17
Entropy	60.69	61.93	53.69	66.47	60.70
Fast-DetectGPT	37.89	72.28	64.20	82.83	64.30
Binoculars	9.64	47.42	47.81	96.80	50.42
ImBD	15.80	62.75	58.68	97.90	58.78
RoBERTa-base	79.54	94.19	97.71	90.64	90.52
RoBERTa-large	88.30	94.91	95.26	95.30	93.44
PC-CAT (Ours)	90.34	96.41	96.89	98.18	95.46

Table 3: Detection performance across four revision tasks, with F1 scores averaged over revised texts by six LLMs (Qwen2-7B, Llama-3-8B, Mixtral-7B, Deepseek-7B, GPT-3.5, and GPT-4o) on the XSum dataset.

stantial margin of 19.11%. See Appendix H.4 for full results on open-source models.

Detection performance across revision tasks

We further evaluate our method on different revision tasks. For the difference between these tasks, please refer to Appendix F.2. As shown in Table 3, our method achieves the highest average F1 score compared to all training-based and statistical baselines. Notably, most statistical methods—as well as ImBD—perform well on the *generate* task but experience significant performance drops on the other three tasks, particularly on *rewrite*. For example, while ImBD achieves 97.90% F1 on *generate*, its performance drops to just 15.80% on *rewrite*. This gap can be attributed to two key factors. First, AI-revised texts typically exhibit much greater similarity to human-written texts than purely AI-generated ones, making detection inherently more difficult. Second, the shift in distribution between the validation and test sets undermines the effectiveness of static thresholding strategies. In contrast, our method achieves strong performance on both the standard *generate* task and the more challenging revision tasks, demonstrating its robustness and generalizability across task types. For detailed results, please refer to Appendix H.5.

4.3 Ablation Study

Ablation on key components To evaluate the contribution of each component in our method, we conduct an ablation study, as presented in the upper part of Table 4. The results show that removing either preference calibration or context-aware thresholding leads to a substantial drop in performance. In particular, removing both components, which reduces the method to the Likelihood baseline, re-

	XSum	Writing	PubMed	SQuAD	Avg.
PC-CAT	96.31	92.11	88.46	90.68	91.89
w/o PC	86.51	90.42	77.54	91.05	86.38
w/o CAT	86.41	91.19	78.84	73.21	82.41
w/o $\mathcal{L}_{\text{center}}$	93.19	85.97	80.63	88.28	87.02
w/o $\mathcal{L}_{\text{ordinal}}$	86.23	88.55	79.02	78.28	83.02
Likelihood	58.12	67.99	60.91	66.28	63.32
<i>preference optimization objective</i>					
SFT	55.84	50.95	53.48	64.85	56.28
DPO	90.61	75.1	67.75	81.79	78.81
SimPO (Ours)	96.31	92.11	88.46	90.68	91.89

Table 4: Ablation results with F1 scores averaged over GPT-3.5/4o-revised texts in the *polish* task. “PC” and “CAT” denote preference calibration and context-aware thresholding, respectively. Detailed descriptions are provided in Appendix F.

sults in a 28.57% decrease in F1 score. Within the context-aware thresholding module (with preference calibration retained), removing the ordinal constraint loss causes a larger performance degradation than removing the centering regression loss. This highlights the critical role of ordinal supervision in learning reliable thresholds, especially in scenarios where centering regression may fail.

Ablation on preference optimization objectives

We further investigate the impact of different preference optimization objectives—namely supervised fine-tuning (SFT), Direct Preference Optimization (DPO), and Simple Preference Optimization (SimPO)—on the preference calibration step in our method. In the SFT setting, the scoring model is trained by directly maximizing the log-probability of AI-generated or AI-polished texts, without incorporating human-written texts as negative samples. As shown in the lower part of Table 4, SFT performs significantly worse than both DPO and SimPO, indicating that contrastive preference signals are essential for effective calibration. Moreover, SimPO outperforms DPO by 13.08% in F1 score, highlighting that maximizing the log-probability gap (as in SimPO) better aligns with our detection metric than maximizing the log-probability ratio (as in DPO).

5 Further Analysis

Out-of-distribution detection To further evaluate the robustness of our method under distribution shifts, we conduct out-of-distribution (OOD) detection experiments on three unseen domains: Hel-laSwag (Zellers et al., 2019), Yelp (Zhang et al., 2015), and SciXGen (Chen et al., 2021). For each

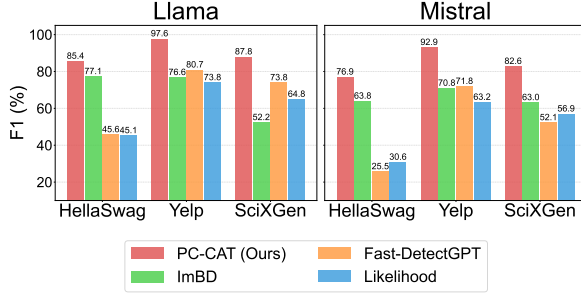


Figure 3: Out-of-distribution detection results on three unseen domains: HellaSwag, Yelp, and SciXGen. AI-revised texts are generated using LLaMA3-8B and Mistral-7B under the *polish* task.

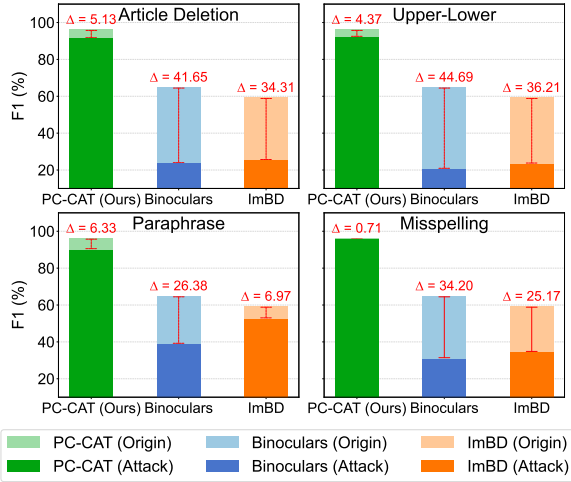


Figure 4: Detection performance under adversarial attacks on the XSum dataset, with F1 scores averaged over GPT-3.5/4o-revised texts in the *polish* task.

dataset, we sample 150 human-written texts and generate their revised versions using LLaMA3-8B and Mistral-7B under the *polish* setting. As illustrated in Figure 3, our PC-CAT significantly outperforms baselines including ImBD, Fast-DetectGPT, and simple Likelihood scoring. PC-CAT achieves an average F1 score exceeding 85% across the three domains, demonstrating strong generalization and robustness in OOD detection scenarios.

Robustness to adversarial attacks Figure 4 illustrates the robustness of PC-CAT under four common adversarial perturbations, including Article Deletion (randomly removing articles), Upper-Lower (randomly flipping the case of initial characters), Paraphrasing (random word substitutions), and Misspellings (injecting typographical errors), following the setup of Dugan et al. (2024). Additional results on other attack methods are in Appendix H.2. Compared to Binoculars and ImBD, PC-CAT consistently maintains higher F1 scores

Method	Spanish	Portuguese	Chinese	Avg.
Likelihood	59.42	48.49	53.49	53.80
Rank	37.96	60.80	56.75	51.84
LogRank	57.14	41.30	52.46	50.30
Entropy	53.00	53.51	57.14	54.55
Fast-DetectGPT	80.00	63.90	75.48	73.13
Binoculars	70.48	62.16	72.78	68.47
ImBD	74.10	65.91	74.92	71.64
PC-CAT (Ours)	87.29	79.38	86.64	84.44

Table 5: Detection performance across three languages on GPT-3.5-polished texts. Full details and analysis are shown in Appendix H.1.

across all attack types. We attribute this robustness to two key design choices in our method. First, the preference calibration stage encourages the scoring model to capture latent distributional patterns, making it less sensitive to surface-level noise. Second, the adaptive thresholding mechanism allows the model to adjust decision boundaries based on input-specific characteristics, thereby compensating for minor textual perturbations. Together, these components enhance the model’s resilience against adversarial attempts aimed at evading detection. Beyond English, Table 5 shows that PC-CAT also outperforms all baselines on GPT-3.5-polished texts in Spanish, Portuguese, and Chinese, with detailed analysis provided in Appendix H.1.

6 Conclusion

In this paper, we propose a novel dynamic thresholding framework for AI-generated text detection, termed PC-CAT, which integrates preference calibration with context-aware thresholding. The core contribution of PC-CAT lies in its two-stage design: (1) a preference calibration step that enhances the scoring model’s discriminative power by maximizing the log-probability gap between AI-generated and human-written texts, and (2) a context-aware thresholding mechanism that adaptively predicts instance-specific decision boundaries. This dynamic thresholding strategy effectively addresses the limitations of conventional static thresholds, which often struggle to generalize under distribution shifts or across heterogeneous inputs. Extensive experiments demonstrate that PC-CAT consistently outperforms prior methods across a wide range of models, domains, and revision tasks, achieving new state-of-the-art results. Moreover, it shows strong robustness under out-of-distribution settings, adversarial attacks, and multilingual scenarios, confirming the generality and practical value of our approach.

7 Limitations

Despite its strong performance, PC-CAT has a few limitations. First, the preference calibration stage depends on parallel preference pairs consisting of human-written and AI-generated (or AI-polished) texts. While effective, collecting such high-quality paired data can be costly and labor-intensive, particularly in low-resource domains or languages. Second, the context-aware thresholding module introduces a modest inference overhead, as it requires an additional forward pass to predict instance-specific thresholds, in contrast to static thresholding methods. Given the blurred boundary between human and AI text and the unbounded nature of writing styles, our model cannot guarantee coverage of all data distributions. It is intended as an assistive reference rather than a universal solution and we acknowledge the potential risk of false positives. Future work may explore weakly supervised or unsupervised alternatives to reduce reliance on labeled data, as well as more efficient designs for adaptive thresholding.

8 Acknowledgments

This work was supported by the National Natural Science Foundation of China (No. 62576368).

9 Ethics Statement

This research enhances the reliability of AI-generated text detection to promote transparency and responsible AI usage. We acknowledge the critical ethical risk of false positives, which could unfairly penalize human authors. Consequently, we emphasize that PC-CAT is intended as a supportive tool to assist human judgment rather than an autonomous arbiter in high-stakes environments. All experiments were conducted using publicly available benchmark datasets, and no personally identifiable information was collected during this study.

References

- Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.
- Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, M  rouane Debbah,   tienne Goffinet, Daniel Hesslow, Julien Launay, Quentin Malartic, Daniele Mazzotta, Badreddine Noune, Baptiste Pannier, and Guilherme
- Penedo. 2023. [The falcon series of open language models](#). *Preprint*, arXiv:2311.16867.
- Guangsheng Bao, Yanbin Zhao, Juncai He, and Yue Zhang. 2025. Glimpse: Enabling white-box methods to use proprietary models for zero-shot llm-generated text detection. In *International Conference on Learning Representations*, volume 2025, pages 51844–51865.
- Guangsheng Bao, Yanbin Zhao, Zhiyang Teng, Linyi Yang, and Yue Zhang. 2024. Fast-detectgpt: Efficient zero-shot detection of machine-generated text via conditional probability curvature. In *International Conference on Learning Representations*, volume 2024, pages 24814–24836.
- Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiusi Du, Zhe Fu, et al. 2024. Deepseek llm: Scaling open-source language models with longtermism. *arXiv preprint arXiv:2401.02954*.
- Sid Black, Gao Leo, Phil Wang, Connor Leahy, and Stella Biderman. 2021. [GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh-Tensorflow](#).
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.
- Hong Chen, Hiroya Takamura, and Hideki Nakayama. 2021. [SciXGen: A scientific paper dataset for context-aware text generation](#). In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1483–1492, Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Jiaqi Chen, Xiaoye Zhu, Tianyang Liu, Ying Chen, Chen Xinhui, Yiwen Yuan, Chak Tou Leong, Zuchao Li, Long Tang, Lei Zhang, et al. 2025. Imitate before detect: Aligning machine stylistic preference for machine-revised text detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 23559–23567.
- Yutian Chen, Hao Kang, Vivian Zhai, Liangze Li, Rita Singh, and Bhiksha Raj. 2023a. [Gpt-sentinel: Distinguishing human and chatgpt generated content](#). *Preprint*, arXiv:2305.07969.
- Zhihong Chen, Shuo Yan, Juhao Liang, Feng Jiang, Xiangbo Wu, Fei Yu, Guiming Hardy Chen, Junying Chen, Hongbo Zhang, Li Jianquan, Wan Xiang, and Benyou Wang. 2023b. [MultilingualSIFT: Multilingual Supervised Instruction Fine-tuning](#).
- Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.

- Liam Dugan, Alyssa Hwang, Filip Trhlík, Andrew Zhu, Josh Magnus Ludan, Hainiu Xu, Daphne Ippolito, and Chris Callison-Burch. 2024. [RAID: A shared benchmark for robust evaluation of machine-generated text detectors](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12463–12492, Bangkok, Thailand. 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.
- 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](#). *Preprint*, arXiv:2407.21783.
- Biyang Guo, Xin Zhang, Ziyuan Wang, Minqi Jiang, Jinran Nie, Yuxuan Ding, Jianwei Yue, and Yupeng Wu. 2023. How close is chatgpt to human experts? comparison corpus, evaluation, and detection. *arXiv preprint arXiv:2301.07597*.
- Zhen Guo and Shangdi Yu. 2023. Authentigtpt: Detecting machine-generated text via black-box language models denoising. *arXiv preprint arXiv:2311.07700*.
- Asier Gutiérrez-Fandiño, Jordi Armengol-Estapé, Marc Pàmies, Joan Llop-Palao, Joaquin Silveira-Ocampo, Casimiro Pio Carrino, Aitor Gonzalez-Agirre, Carme Armentano-Oller, Carlos Rodriguez-Penagos, and Marta Villegas. 2021. Maria: Spanish language models. *arXiv preprint arXiv:2107.07253*.
- 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*, ICML’24. JMLR.org.
- Xinlei He, Xinyue Shen, Zeyuan Chen, Michael Backes, and Yang Zhang. 2024. Mgtbench: Benchmarking machine-generated text detection. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, pages 2251–2265.
- Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. [The curious case of neural text de-generation](#). In *International Conference on Learning Representations*.
- Biao Hu, Zhen Huang, Minghao Hu, Ziwen Zhang, and Yong Dou. 2022a. Adaptive threshold selective self-attention for Chinese ner. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 1823–1833.
- Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022b. [LoRA: Low-rank adaptation of large language models](#). In *International Conference on Learning Representations*.
- Xiaomeng Hu, Pin-Yu Chen, and Tsung-Yi Ho. 2023. Radar: Robust ai-text detection via adversarial learning. *Advances in neural information processing systems*, 36:15077–15095.
- Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, L  lio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timoth  e Lacroix, and William El Sayed. 2023. [Mistral 7b](#). *Preprint*, arXiv:2310.06825.
- 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.
- Yi Liu, Gelei Deng, Yuekang Li, Kailong Wang, Tianwei Zhang, Yepang Liu, Haoyu Wang, Yan Zheng, and Yang Liu. 2023. [Prompt injection attack against llm-integrated applications](#). *Preprint*, arXiv:2306.05499.
- 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. *arXiv preprint arXiv:1907.11692*.
- Chengzhi Mao, Carl Vondrick, Hao Wang, and Junfeng Yang. 2024. [Raidar: generative ai detection via rewriting](#). *Preprint*, arXiv:2401.12970.
- Yu Meng, Mengzhou Xia, and Danqi Chen. 2024. Simpo: Simple preference optimization with a reference-free reward. In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D Manning, and Chelsea Finn. 2023. Detectgpt: Zero-shot machine-generated text detection using probability curvature. In *International Conference on Machine Learning*, pages 24950–24962. PMLR.

- Marius Mosbach, Maksym Andriushchenko, and Dietrich Klakow. 2021. [On the stability of fine-tuning bert: Misconceptions, explanations, and strong base-lines](#). In *International Conference on Learning Representations*.
- 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. 2022. Chatgpt: Optimizing language models for dialogue. <https://openai.com/blog/chatgpt/>.
- Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. [Direct preference optimization: Your language model is secretly a reward model](#). In *Thirty-seventh Conference on Neural Information Processing Systems*.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. [Exploring the limits of transfer learning with a unified text-to-text transformer](#). *Journal of Machine Learning Research*, 21(140):1–67.
- Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. [SQuAD: 100,000+ questions for machine comprehension of text](#). In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. 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](#). *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.
- Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023a. [Llama: Open and efficient foundation language models](#). *arXiv preprint arXiv:2302.13971*.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajwal Bhargava, Shruti Bhosale, et al. 2023b. [Llama 2: Open foundation and fine-tuned chat models](#). *arXiv preprint arXiv:2307.09288*.
- Ben Wang and Aran Komatsuzaki. 2021. [GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model](#). <https://github.com/kingoflolz/mesh-transformer-jax>.
- Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Toru Sasaki, Thomas Arnold, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024. [M4: Multi-generator, multi-domain, and multi-lingual black-box machine-generated text detection](#). In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1369–1407, St. Julian's, Malta. Association for Computational Linguistics.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.
- Yi Xu, Lei Shang, Jinxing Ye, Qi Qian, Yu-Feng Li, Baigui Sun, Hao Li, and Rong Jin. 2021. [Dash: Semi-supervised learning with dynamic thresholding](#). In *International conference on machine learning*, pages 11525–11536. PMLR.
- An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, and 43 others. 2024a. [Qwen2 technical report](#). *Preprint*, arXiv:2407.10671.
- Xianjun Yang, Wei Cheng, Yue Wu, Linda Ruth Petzold, William Yang Wang, and Haifeng Chen. 2024b. [DNA-GPT: Divergent n-gram analysis for training-free detection of GPT-generated text](#). In *The Twelfth International Conference on Learning Representations*.
- Zhen Yang and Frank Emmert-Streib. 2023. Threshold-learned cnn for multi-label text classification of electronic health records. *IEEE Access*, 11:93402–93419.
- W. J. Youden. 1950. [Index for rating diagnostic tests](#). *Cancer*, 3(1):32–35.
- 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](#). *Advances in Neural Information Processing Systems*, 37:16194–16212.
- Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. [HellaSwag: Can a machine really finish your sentence?](#) In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy. Association for Computational Linguistics.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. [Character-level convolutional networks for text classification](#). In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Ying Zhou, Ben He, and Le Sun. 2024. Humanizing machine-generated content: Evading ai-text detection through adversarial attack. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation*.

Xiaowei Zhu, Yubing Ren, Yanan Cao, Xixun Lin, Fang Fang, and Yangxi Li. 2025. [Reliably bounding false positives: A zero-shot machine-generated text detection framework via multiscaled conformal prediction](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12298–12319, Vienna, Austria. Association for Computational Linguistics.

A Implementation details

A.1 Training Details

We fine-tune the scoring model for 2 epochs using 1,000 samples. We use the AdamW optimizer with an initial learning rate of $1e-4$ and linear decay. The preference calibration stage adopts a β value of 0.1, α of 1.0, and margin γ of 0.5. For parameter-efficient tuning, we apply Low-Rank Adaptation (LoRA) (Hu et al., 2022b) with a rank of 8, a LoRA alpha of 16, and a dropout rate of 0.1. In Stage 1, we use the final checkpoint after 2 epochs. During the second training stage, we freeze the backbone model and update only the parameters of the threshold prediction network. All experiments are conducted on Ubuntu 18.04 using eight NVIDIA RTX 2080 Ti GPUs (12GB each). The software stack includes Python 3.11, PyTorch 2.4.0, Transformers 4.51.3, and Datasets 3.2.0.

A.2 Baselines Details

For the Likelihood, Rank, LogRank, Entropy, and ImBD baselines, we use GPT-Neo-2.7B (Black et al., 2021) as the scoring model. For Fast-DetectGPT, GPT-J-6B (Wang and Komatsuzaki, 2021) is used to generate local perturbations, while GPT-Neo-2.7B is used to compute the scores of perturbed samples. For Binoculars, Falcon-7B (Almazrouei et al., 2023) serves as the *observer* model to compute log-perplexity, and Falcon-7B-Instruct is used as the *performer* model to compute cross-perplexity. The discrepancy between these two scores captures model agreement and serves as the detection signal.

For all statistical baselines, the optimal decision threshold is determined using the Youden Index,

which provides a principled criterion for balancing true positive and true negative rates. We conduct a grid search over candidate thresholds on the validation set and select the value that maximizes the Youden Index (Youden, 1950). The selected threshold is then used during evaluation.

B Analysis of Preference Calibration

In this section, we analyze the alignment between the objectives of preference calibration algorithms and our core detection task. We first establish the foundational hypothesis of our detector, then deconstruct the objective mismatch inherent in Direct Preference Optimization (DPO), and finally, we demonstrate the superior alignment offered by Simple Preference Optimization (SimPO).

B.1 The Foundational Hypothesis of Our Detection Task

Following recent work (Mitchell et al., 2023; Bao et al., 2024; Hans et al., 2024; Bao et al., 2025), our detection method is based on the hypothesis that AI-generated text exhibits lower perplexity, favoring high-probability tokens, whereas human-written text is more lexically diverse and less predictable. We quantify this characteristic using the average log-probability of a given text sequence $x = (w_1, \dots, w_T)$, which is defined under π_θ with parameters θ as:

$$\begin{aligned}\bar{p}(x) &= \frac{1}{T} \log \pi_\theta(x) \\ &= \frac{1}{T} \sum_{t=1}^T \log \pi_\theta(w_t | w_{<t}).\end{aligned}\tag{10}$$

Our detection mechanism classifies a text x as AI-generated if its score, $\bar{p}(x)$, exceeds a threshold. Therefore, the central challenge is to fine-tune a model π_θ that maximally separates human and AI-generated texts based on this metric.

B.2 Misalignment of DPO for Likelihood-Based Detection

DPO’s training objective is misaligned with our detection goal. DPO maximizes the sum of the log-probability of preferred responses over rejected ones, effectively optimizing an implicit reward function. For a given text x , this reward is defined as:

$$r_{\text{DPO}}(x) = \beta \log \frac{\pi_\theta(x)}{\pi_{\text{ref}}(x)}.\tag{11}$$

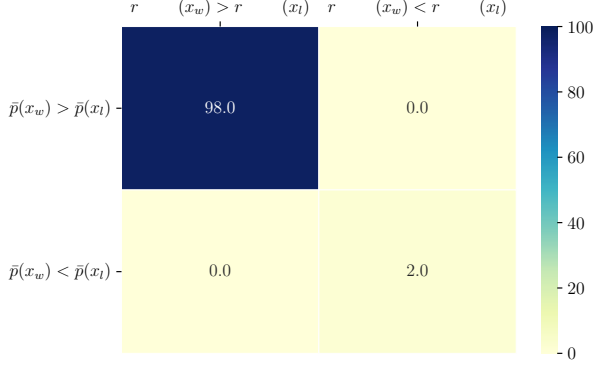


Figure 5: Empirical alignment between the SimPO reward (r_{SimPO}) and the average log-probability ($\bar{p}$) on the training set. The heatmap demonstrates that SimPO’s reward function directly optimizes the likelihood-based detection metric.

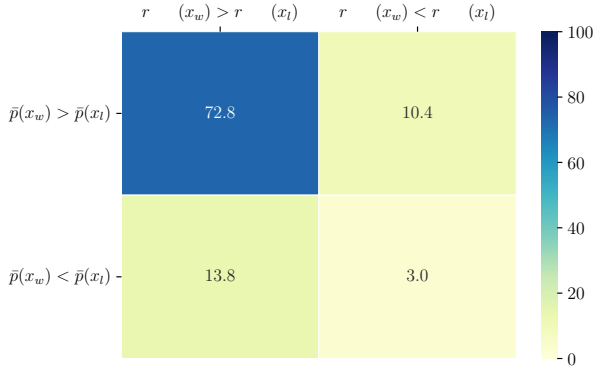


Figure 6: Empirical misalignment between the DPO reward (r_{DPO}) and the average log-probability ($\bar{p}$) on the training set. The heatmap illustrates that DPO’s training objective contradicts the likelihood-based detection metric in 24.2% (13.8% + 10.4%) of cases.

DPO’s objective is to enforce $r_{\text{DPO}}(x_w) > r_{\text{DPO}}(x_l)$ for a chosen response x_w and a rejected response x_l . However, due to the normalization by the reference model π_{ref} , satisfying this condition does not guarantee that our desired detection metric $\bar{p}(x_w)$ will be greater than $\bar{p}(x_l)$. As shown in Figure 6, for approximately 24.2% of training examples, the ranking of preference pairs by the DPO reward contradicts the ranking by average log-probability. This discrepancy reveals a fundamental limitation of using DPO-trained models for likelihood-based detection.

B.3 SimPO: Aligning the Training Objective with Detection

To address the misalignment inherent in DPO, we adopt Simple Preference Optimization (SimPO). SimPO defines its reward function directly in terms

of the average log-probability:

$$r_{\text{SimPO}}(x) = \frac{1}{T} \beta \log \pi_{\theta}(x). \quad (12)$$

This formulation establishes a direct relationship between the reward function and our detection metric. Consequently, when the model is trained to satisfy the preference $r_{\text{SimPO}}(x_w) > r_{\text{SimPO}}(x_l)$, it is explicitly optimizing for $\bar{p}(x_w) > \bar{p}(x_l)$. As illustrated in Figure 5, our experiments confirm this ideal alignment: whenever the SimPO training objective is met for a preference pair, the average log-probability ranking aligns perfectly.

C Theoretical Analysis: Dynamic vs. Static Thresholding

In this section, we provide a formal analysis demonstrating why the dynamic thresholding in PC-CAT generalizes better than global static thresholding.

C.1 Dynamic Thresholds

Let $\bar{p}(x_h)$ and $\bar{p}(x_a)$ denote the scores (average log-probabilities) of a paired human-written and AI-generated text. As described in Section 3.3, the context-aware thresholding (CAT) module is optimized via a centering regression loss $\mathcal{L}_{\text{center}}$. This objective encourages the predicted thresholds for both samples to converge toward their local midpoint:

$$\tau_{\phi}(x) \approx \mu(x_h, x_a) = \frac{\bar{p}(x_h) + \bar{p}(x_a)}{2}. \quad (13)$$

For any given pair $\{x_h^i, x_a^i\}$ in the test set $\mathcal{D}$, correct classification is achieved according to the decision rule in Eq. (2) if the predicted threshold $\tau_{\phi}(x^i)$ satisfies the local interval constraint:

$$\bar{p}(x_h^i) < \tau_{\phi}(x^i) \leq \bar{p}(x_a^i). \quad (14)$$

Let $\delta(x^i) = |\tau_{\phi}(x^i) - \mu(x_h^i, x_a^i)|$ represent the prediction error of the threshold regressor. The instance x^i is correctly classified as long as this error remains within the local margin:

$$\delta(x^i) < \frac{\bar{p}(x_a^i) - \bar{p}(x_h^i)}{2}. \quad (15)$$

Crucially, this condition is instance-specific. The model only needs to ensure that the predicted boundary falls within the safety margin of the current input, independent of the score distributions of other instances.

C.2 The Brittleness of Static Thresholds

In contrast, a static threshold τ_{static} must satisfy a global constraint to achieve robust performance across the entire test set $\mathcal{D}$:

$$\forall i \in \mathcal{D}, \quad \tau_{\text{static}} \in (\bar{p}(x_h^i), \bar{p}(x_a^i)]. \quad (16)$$

Mathematically, this requires a non-empty intersection of all local intervals across all test samples:

$$\tau_{\text{static}} \in \bigcap_{i \in \mathcal{D}} (\bar{p}(x_h^i), \bar{p}(x_a^i)]. \quad (17)$$

This requirement is extremely brittle. As empirically demonstrated in Table 6, when a model faces distribution shifts (e.g., from Task 1 to Task 2), the score distribution can undergo a significant negative shift. Such shifts often cause the global intersection to become empty:

$$\bigcap_{i \in \mathcal{D}} (\bar{p}(x_h^i), \bar{p}(x_a^i)] = \emptyset. \quad (18)$$

In such cases, no single static threshold can correctly separate the classes, leading to the catastrophic F1 score collapse observed in our experiments (from 0.73 to 0.02). Conversely, our dynamic threshold $\tau_\phi(x)$ is not bound by this global intersection. It only needs to satisfy local constraints, making PC-CAT inherently more robust to distribution shifts.

D Justification for the Choice of Evaluation Metric

D.1 Empirical Analysis

While the Area Under the Receiver Operating Characteristic Curve (AUROC) is a widely used, threshold-independent metric for evaluating the ranking capability of a binary classifier, our work deliberately adopts the F1 score as the primary evaluation metric because it clearly articulates the motivation for the threshold calibration component.

The fundamental limitation of AUROC is its invariance to monotonic transformations of the prediction scores. Because AUROC depends entirely on the rank-order of scores rather than their absolute values, a uniform shift across all prediction scores will not change the resulting curve or its area. However, in any real-world deployment, a detector must commit to a specific decision threshold to make a classification. If the distribution of scores at test time shifts relative to the distribution on which the threshold was calibrated, the

Method	Task	Score (AI / Human)	F1	AUROC
ImBD (DPO)	Task 1	-8.7 ± 3.0 (-7.3 / -10.1)	0.73	0.77
	Task 2	-17.4 ± 3.3 (-15.7 / -19.2)	0.02	0.79
PC-CAT	Task 1	-4.2 ± 1.1 (-3.4 / -4.8)	0.87	0.89
	Task 2	-3.9 ± 0.5 (-3.6 / -4.2)	0.81	0.81

Table 6: Score distribution and detection evaluation. Task 1: Rewrite task, Dataset: XSum, Reviser: GPT-4o. Task 2: Polish task, Dataset: PubMed, Reviser: GPT-4o

detector’s practical performance can collapse, even while its theoretical ranking ability (measured by AUROC) remains high.

To illustrate this phenomenon empirically, we analyzed the score distributions and performance of the ImBD baseline across two different tasks: task 1 (rewrite task on XSum dataset by GPT-4o) and task 2 (polish task on PubMed dataset by GPT-4o). As shown in Table 6, the model’s mean log-probability score undergoes a significant negative shift, from -8.7 on task 1 to -17.4 on task 2. Despite this shift, the model’s ability to rank samples correctly is preserved, resulting in a consistently high AUROC of 0.77 and 0.79.

However, a static threshold calibrated on one distribution (e.g., task 1) is misaligned for the shifted distribution (e.g., task 2). This misalignment leads to catastrophic misclassification, causing the F1 score to collapse from 0.73 to a near-zero 0.02. This failure occurs even when the threshold is optimally chosen on a validation set via the Youden Index, as such calibration cannot account for unforeseen shifts in the test distribution. In contrast, our method, which employs an adaptive threshold, maintains high F1 scores across both tasks by adjusting its decision boundary in response to the input.

Therefore, the F1 score exposes the fragility of fixed-boundary detectors under distribution shifts, a key gap between theoretical and practical detection performance. This finding directly motivates our proposed method, PC-CAT, an adaptive thresholding mechanism designed to provide robust and reliable detection in practice, not just in theory.

D.2 Formal Proofs

Proposition 1. *Let X and Y be random variables for positive and negative class scores. Let $f : \mathbb{R} \rightarrow \mathbb{R}$ be strictly monotonically increasing. AUROC is*

Dataset	Model	Score		Threshold	
		Human	AI	Human	AI
PubMed	GPT-4o	-4.84	-3.45	-3.99	-4.44
	GPT-3.5		-3.34		-4.28
Writing	GPT-4o	-5.02	-3.48	-3.88	-4.38
	GPT-3.5		-2.91		-4.47
SQuAD	GPT-4o	-3.60	-3.01	-3.04	-3.99
	GPT-3.5		-2.74		-4.09
XSum	GPT-4o	-4.18	-3.29	-3.07	-4.34
	GPT-3.5		-3.04		-4.52

Table 7: Adaptive threshold and score across different datasets and models.

invariant under the transformation f .

Proof. The probabilistic interpretation of AUROC is:

$$\text{AUC} = P(X > Y) + \frac{1}{2}P(X = Y). \quad (19)$$

Let $X' = f(X)$ and $Y' = f(Y)$. Since f is strictly monotonically increasing, $x > y \iff f(x) > f(y)$ and $x = y \iff f(x) = f(y)$. Thus:

$$\begin{aligned} \text{AUC}' &= P(X' > Y') + \frac{1}{2}P(X' = Y') \\ &= P(X > Y) + \frac{1}{2}P(X = Y) = \text{AUC}. \end{aligned} \quad (20)$$

This proves AUROC fails to capture absolute shifts in score distributions. $\square$

Proposition 2. Let τ_j^* be the optimal threshold for a distribution S . Let f be a transformation such that $f(\tau_j^*) \neq \tau_j^*$. Applying the fixed threshold τ_j^* to $S' = f(S)$ degrades performance.

Proof. Let $J(\tau) = \text{TPR}(\tau) - \text{FPR}(\tau)$. The optimal threshold is $\tau_j^* = \arg \max_{\tau} J(\tau)$. Under S' , the fixed rule $\hat{y}' = \mathbb{I}(f(S) > \tau_j^*)$ is equivalent to using $\tau_{eff} = f^{-1}(\tau_j^*)$ on S . Since $\tau_{eff} \neq \tau_j^*$ and τ_j^* is the unique maximizer, $J(\tau_{eff}) < J(\tau_j^*)$. This deviation explains the fragility of static-threshold methods. $\square$

E Analysis of Threshold and Score

Table 7 shows the mean threshold and score for AI-generated and Human-written text. This provides several key insights that prove the effectiveness of our adaptive threshold mechanism:

- **Adaptation to Text Source:** The mean threshold predicted for AI-generated text is consistently lower than the mean threshold for

human-written text. This demonstrates that the mechanism actively creates a "safety margin" based on the input's characteristics, rather than using a single, fixed boundary.

- **Generator Adaptation:** The threshold adapts to the specific AI model. For example, in the XSum domain, the AI threshold for the more advanced GPT-4o is -4.34, which is higher than -4.52 for GPT-3.5. This suggests the detector is slightly more confident in GPT-3.5 than in GPT-4o.
- **Domain Adaptation:** PC-CAT demonstrates strong domain awareness. For instance, the average threshold for human text in fact-based domains like SQuAD (-3.04) and XSum (-3.07) is noticeably higher than in more complex domains like Writing (-3.88) and PubMed (-3.99). This shows that the model intelligently adapts its decision boundary based on the expected characteristics of each domain.

In summary, this analysis confirms that our adaptive threshold is a principled mechanism, which dynamically adjusts to intrinsic textual features. Furthermore, the learned distance between the score and the threshold acts as a confidence margin. This margin provides tolerance against adversarial attacks by absorbing textual perturbations.

F Detailed Descriptions of Experiments

F.1 Detailed Configurations of Ablation Variants

To clarify the experimental setup in Section 4.3, we provide the specific implementation details for each ablated version of PC-CAT below:

- **w/o CAT:** This variant removes the Context-Aware Thresholding (CAT) module. In its absence, we follow the standard statistical baseline protocol by using the Youden Index to determine a single optimal static threshold on the validation set.
- **w/o PC:** This variant skips the Preference Calibration (PC) stage. The backbone model (Qwen2.5-0.5B) is used directly as the scoring function, and the CAT module is trained on top of the uncalibrated hidden states.
- **w/o $\mathcal{L}_{\text{center}}$:** In this variant, we retain the Preference Calibration (PC) stage. The centering

regression loss term is removed from the total CAT objective. The threshold prediction network is optimized solely using the ordinal constraint, such that $\mathcal{L}_{\text{CAT}} = \alpha \cdot \mathcal{L}_{\text{ordinal}}$.

- **w/o $\mathcal{L}_{\text{ordinal}}$:** In this variant, we retain the Preference Calibration (PC) stage. The auxiliary ordinal constraint loss is removed. The objective for the threshold prediction network simplifies to the centering regression loss activated by the indicator function: $\mathcal{L}_{\text{CAT}} = \mathbb{I}(\bar{p}(x_a) > \bar{p}(x_h)) \cdot \mathcal{L}_{\text{center}}$.

F.2 Distinction Between “Rewrite” and “Polish”

Essentially, "polish" suggests minor stylistic and grammatical improvements, while "rewrite" implies a more fundamental restructuring of the text while preserving the original meaning.

- **Polish:** The LLM is asked to polish/adjust the text based on a randomly picked style.
- **Rewrite:** The LLM is asked to rewrite the given text without missing the original details.

G Extended Analysis on the Log-probability Signals

In this section, we provide additional empirical evidence regarding the effectiveness of log-probability as a discriminative signal for recent Large Language Models (LLMs) and further justify the necessity of our proposed framework.

G.1 Intrinsic Discriminative Power of Modern LLMs

A potential concern in AI-generated text detection is whether the log-probability signal remains reliable for increasingly powerful LLMs. To investigate this, we evaluate the raw likelihood of a **frozen Qwen2.5-0.5B** model without any post-training or calibration. As shown in Table 8 and Table 9, this relatively small backbone achieves high AUROC scores across all generators, including state-of-the-art models.

Notably, the performance of Qwen2.5-0.5B is comparable to, and in some cases exceeds, that of significantly larger models such as GPT-Neo-2.7B and Falcon-7B. These findings confirm that the discriminative signal based on token-level log-probability does not vanish as LLMs evolve. Instead, it remains an intrinsic and effective feature

for distinguishing machine-generated text from human writing.

G.2 Methodological Necessity of PC-CAT

While raw log-probability provides a strong ranking signal (AUROC), it is insufficient for robust real-world deployment for two primary reasons:

- **Distribution Overlap:** The log-probability distributions of human and AI text often exhibit persistent overlaps, particularly in challenging scenarios like AI-revised text detection. Our preference calibration (PC) step explicitly maximizes this gap to enhance discriminative capacity.
- **Threshold Sensitivity:** High AUROC indicates a strong ranking capability but does not ensure a high F1 score. Defining a universal static threshold is difficult due to significant score shifts across different domains. Our context-aware thresholding (CAT) mechanism converts this ranking potential into accurate classification decisions by adapting to instance-specific features.

G.3 Generalization to External Benchmarks

To verify that our approach captures generalizable detection patterns rather than overfitting to specific training distributions, we extend our evaluation to the M4 (Wang et al., 2024) and RealDet benchmarks (Zhu et al., 2025). Crucially, for these experiments, we do not perform any additional training or fine-tuning on the external datasets. Instead, we directly employ the model weights and threshold regressor that were previously trained on our primary dataset (as described in Section 4.1). As shown in Table 10, our method (PC-CAT) consistently improves detection performance on these unseen datasets. For instance, our approach further boosts the AUROC on RealDet to 92.45%. This demonstrates that the preference calibration stage effectively sharpens the intrinsic detection signal.

H Additional Results

H.1 Multilingual Performance

Our experiments are conducted on machine-revised text in Spanish (Gutiérrez-Fandiño et al., 2021), Portuguese (Chen et al., 2023b), and Chinese (Wang et al., 2024), following the same experimental setup. The dataset consists of human-written texts and their GPT-3.5-polished counterparts. We train on a given language’s training set

Scoring Model	Method	GPT-3.5	GPT-4o	Deepseek-7b	Llama-3-8B	Mistral-7B
Qwen2.5-0.5B	Likelihood	93.88	84.75	93.02	91.54	86.80
	LogRank	91.42	80.26	90.76	88.31	81.84
	LRR	74.74	57.44	77.02	66.84	55.25
GPT-Neo-2.7B	Likelihood	87.88	80.77	88.16	85.00	76.57
	LogRank	84.96	76.94	86.75	81.92	71.47
	LRR	72.03	62.14	78.80	68.76	52.61
Falcon-7B	Likelihood	88.56	78.34	87.99	87.28	76.49
	LogRank	85.67	74.30	86.28	83.81	70.49
	LRR	71.97	58.36	76.36	67.14	50.85

Table 8: Performance on the Polish Task with different scoring models (Writing Dataset). Metric: AUROC (%).

Scoring Model	Method	GPT-3.5	GPT-4o	Deepseek-7b	Llama-3-8B	Mistral-7B
Qwen2.5-0.5B	Likelihood	97.41	84.08	99.25	96.40	98.46
	LogRank	97.22	82.72	99.06	96.61	97.98
	LRR	91.77	74.36	95.70	94.91	91.69
GPT-Neo-2.7B	Likelihood	95.77	75.85	98.86	94.68	94.43
	LogRank	95.83	74.86	98.75	94.99	94.06
	LRR	91.64	67.58	96.96	92.95	89.24
Falcon-7B	Likelihood	96.84	75.99	98.98	96.59	96.46
	LogRank	96.38	75.14	98.65	96.86	95.67
	LRR	90.20	68.74	95.21	95.25	89.05

Table 9: Performance on the Generate Task with different scoring models (XSum Dataset). Metric: AUROC (%).

Scoring Model	Method	M4	RealDet
GPT-Neo-2.7B	Likelihood	83.43	87.88
	LogRank	85.15	88.56
	LRR	87.05	89.08
Falcon-7B	Likelihood	85.77	85.35
	LogRank	87.50	86.28
	LRR	71.97	58.36
Qwen2.5-0.5B	Likelihood	85.72	89.16
	LogRank	87.47	89.90
	LRR	87.66	90.12
+Ours(PC)	Likelihood	90.25	92.45

Table 10: Performance on other public benchmarks with different scoring models. Metric: AUROC (%).

and evaluate on its corresponding test set. Notably, Portuguese proved to be the most challenging task for all models. This is likely because all the scoring models are predominantly pretrained on English data, limiting their proficiency in other languages. Despite this, as shown in Table 5, PC-CAT consistently outperforms the strongest baselines across all languages, surpassing Fast-DetectGPT by 7.29% in Spanish and 11.16% in Chinese, while exceeding ImBD by 13.47% in Portuguese. These results highlight its robustness and adaptability in multi-lingual settings.

H.2 Robustness against Additional Adversarial Attacks

Table 11 presents the detection performance of various methods under three additional adversarial attacks on the XSum dataset, including Number (randomly replacing numbers in the text), Alternative Spelling (swapping between common spelling variants, e.g., US and British English) and Synonym (replacing non-stop words with a random synonym). The results indicate that our method maintains superior performance compared to all baselines across different attack types, demonstrating its robustness against adversarial manipulations.

Furthermore, we test PC-CAT’s resilience against paraphrasing and prompt-based evasion techniques. Table 12 summarizes the performance under back-translation (Zhou et al., 2024) and prompt injection (Liu et al., 2023) attacks across four settings. The results show that PC-CAT maintains good detection performance in all scenarios, demonstrating its strong robustness against more sophisticated attacks that alter sentence structure.

H.3 Comparison of Scoring Backbones

To verify that PC-CAT’s improvements originate from the proposed method rather than its scoring backbone, we conduct a controlled comparison.

Attack Method	ImBD	PC-CAT (Ours)
Original	59.41	96.31
Number	56.05 (-3.36)	96.13 (-0.18)
Alternative Spelling	51.90 (-7.51)	95.24 (-1.07)
Synonym	19.09 (-40.32)	85.71 (-10.60)

Table 11: F1 Score under various attacks on the XSum dataset. Values in parentheses indicate the decrease from the original text.

Attack Method	XSum	Writing
Original	96.31	92.11
Back-translation	95.09 (-1.22)	91.21 (-0.90)
Prompt Injection	95.28 (-1.03)	91.12 (-0.99)

Table 12: F1 Score of PC-CAT under Back-translation and Prompt Injection attacks. Values in parentheses indicate the performance degradation from the unattacked text.

PC-CAT adopts Qwen2.5-0.5B as its scoring backbone. In contrast, several statistical baselines were originally built on English-centric backbones such as GPT-Neo-2.7B or Falcon-7B. We therefore re-run these baselines with the Qwen2.5-0.5B backbone. We cover both the monolingual (English) GPT-3.5 polishing task and the multilingual setting (Chinese, Portuguese, and Spanish). Results are summarized in Table 13. In the English setting, adopting Qwen2.5-0.5B yields no improvement for the baselines and instead consistently degrades their performance. In the multilingual setting, the results are mixed and not consistently favorable to Qwen2.5-0.5B: Likelihood and LogRank benefit in some languages, whereas Fast-DetectGPT remains better with its original backbone. Together, these comparisons confirm that PC-CAT’s advantages arise from its proposed mechanisms rather than the choice of scoring backbone.

H.4 Detail Results on Open-source Models

Table 14 presents the detailed detection performance (F1 score) on AI-polished texts produced by open-source models across the XSum, SQuAD, and WritingPrompts datasets. The results show that our method outperforms all baselines in most settings, with particularly strong gains on the SQuAD and WritingPrompts datasets.

H.5 Detailed Results on Diverse Revision Tasks

Table 15 reports detailed F1 scores for detecting AI-revised texts generated by six LLMs across four

revision tasks. Our method achieves the highest F1 score in most cases, demonstrating its effectiveness and generalization across different types of revisions.

I Pipeline to Construct Paired Datasets

To construct the paired datasets efficiently, we can adopt the following pipeline:

1. **Source Text Collection:** First, we can gather a large corpus of human-written texts.
2. **Instruction Generation:** Next, we can employ a Large Language Model (LLM) to generate a diverse pool of prompts centered on text polishing and rewriting.
3. **Paired Data Synthesis:** Finally, for each human text, we can sample a prompt from the pool and instruct the LLM to produce a rewritten version, thus forming a (human, machine-written) pair.

We believe that this pipeline provides a scalable and systematic approach to bypass the bottleneck of finding naturally occurring data, enabling the creation of a robust dataset for training and evaluation.

Algorithm	English (GPT-3.5 Polishing)			Multilingual		
	XSum	Writing	PubMed	Chinese	Portuguese	Spanish
Likelihood	58.82 / 64.25	77.52 / 80.81	61.03 / 64.63	72.85 / 53.49	44.93 / 48.49	64.38 / 59.42
LogRank	55.56 / 64.50	75.40 / 74.81	58.24 / 64.73	71.14 / 52.46	42.01 / 41.30	51.30 / 57.14
Fast-DetectGPT	55.52 / 66.88	65.29 / 86.32	26.29 / 59.23	73.82 / 75.48	57.59 / 63.90	69.30 / 80.00

Table 13: F1 scores (%) of statistical baselines under the Qwen2.5-0.5B backbone versus their original backbones (Qwen / original). The left panel reports the monolingual English GPT-3.5 polishing task, and the right panel reports the multilingual setting. Higher values are in bold.

Dataset	Method	Source Model				Avg.
		Qwen2-7B	Llama-3-8B	Mixtral-7B	Deepseek-7B	
XSum	Likelihood	51.72	66.82	61.92	65.55	61.50
	Rank	55.76	59.90	53.01	63.10	57.94
	LogRank	58.94	66.06	63.87	66.06	63.73
	Entropy	45.82	54.82	54.45	55.56	52.66
	Fast-DetectGPT	50.98	73.18	65.68	76.24	66.52
	Binoculars	21.05	54.29	30.00	51.46	39.20
	ImBD	46.15	70.13	51.49	65.47	58.31
	RoBERTa-base	100.00	92.09	96.55	92.47	95.28
	RoBERTa-large	99.67	95.47	98.31	97.26	97.68
	PC-CAT (Ours)	98.03	97.35	97.01	96.32	97.18
SQuAD	Likelihood	56.77	67.79	63.86	67.47	63.97
	Rank	59.56	64.38	59.95	67.38	62.82
	LogRank	61.20	66.67	65.11	66.67	64.91
	Entropy	46.80	56.40	54.88	60.41	54.62
	Fast-DetectGPT	59.77	78.82	75.15	56.46	67.55
	Binoculars	26.59	63.01	38.71	69.94	49.56
	ImBD	73.09	87.54	78.46	76.56	78.91
	RoBERTa-base	88.89	86.50	85.80	83.28	86.12
	RoBERTa-large	87.43	88.10	88.10	85.71	87.34
	PC-CAT (Ours)	92.21	92.88	91.54	87.30	90.98
WritingPrompts	Likelihood	56.64	79.41	72.67	79.77	72.12
	Rank	64.20	66.26	59.87	73.93	66.07
	LogRank	60.29	73.82	70.78	73.49	69.60
	Entropy	21.70	41.84	34.93	62.04	40.13
	Fast-DetectGPT	75.14	83.38	80.95	85.56	81.26
	Binoculars	69.83	91.37	77.24	86.89	81.33
	ImBD	79.37	94.08	84.41	90.65	87.13
	RoBERTa-base	86.71	84.02	86.05	83.68	85.12
	RoBERTa-large	90.36	88.69	90.03	87.65	89.18
	PC-CAT (Ours)	94.01	92.99	94.01	92.99	96.65

Table 14: Detailed detection performance on AI-revised texts generated by open-source models across XSum, SQuAD, and WritingPrompts.

Model	Method	Tasks				Avg.
		Rewrite	Polish	Expand	Generate	
GPT-3.5	Likelihood	54.40	64.25	68.08	68.81	63.89
	Rank	56.91	56.14	64.14	68.67	61.47
	LogRank	57.91	64.50	67.57	67.57	64.39
	Entropy	58.77	55.19	62.50	68.06	61.13
	Fast-DetectGPT	16.00	66.88	73.39	83.47	59.94
	Binoculars	1.30	64.10	47.76	99.67	53.21
	ImBD	8.92	77.55	71.79	94.00	63.07
	RoBERTa-base	93.24	97.96	96.91	95.47	95.90
	RoBERTa-large	96.91	97.96	96.39	96.30	96.89
	PC-CAT(Ours)	93.10	97.69	98.03	99.67	97.12
GPT-4o	Likelihood	61.92	62.25	64.58	69.61	64.59
	Rank	62.05	56.15	60.26	66.50	61.24
	LogRank	64.81	63.87	64.50	67.87	65.26
	Entropy	61.50	56.28	62.83	65.09	61.43
	Fast-DetectGPT	32.39	52.22	57.73	80.35	55.67
	Binoculars	3.85	65.98	13.41	87.50	42.69
	ImBD	2.67	41.27	37.84	99.33	45.28
	RoBERTa-base	59.15	92.47	94.74	97.26	85.91
	RoBERTa-large	73.42	97.61	93.24	96.05	90.08
	PC-CAT(Ours)	80.77	94.92	94.56	98.03	92.07
Qwen2-7B	Likelihood	53.65	51.72	61.58	69.61	59.14
	Rank	60.26	55.76	62.05	70.36	62.11
	LogRank	59.28	58.93	64.81	67.87	62.72
	Entropy	59.46	45.82	58.06	64.78	57.03
	Fast-DetectGPT	19.21	50.98	57.73	82.72	52.66
	Binoculars	0.00	21.05	8.75	96.62	31.61
	ImBD	12.50	46.15	46.94	99.33	51.23
	RoBERTa-base	97.26	100.00	99.67	100.00	99.23
	RoBERTa-large	99.33	99.67	97.40	97.40	98.45
	PC-CAT(Ours)	95.97	98.03	98.36	98.36	97.68
Llama-3-8B	Likelihood	68.08	66.82	65.23	69.61	67.44
	Rank	63.80	59.90	63.80	69.42	64.23
	LogRank	67.27	66.06	63.87	67.87	66.27
	Entropy	64.45	54.82	57.71	65.73	60.68
	Fast-DetectGPT	65.58	73.18	81.71	82.39	75.72
	Binoculars	29.05	54.29	74.59	99.01	64.24
	ImBD	43.75	70.13	68.42	94.74	69.26
	RoBERTa-base	89.71	92.09	87.64	69.00	84.61
	RoBERTa-large	98.99	95.47	91.41	90.28	94.04
	PC-CAT(Ours)	96.32	97.35	94.20	96.32	96.05
Mistral-7B	Likelihood	59.85	61.92	68.69	69.61	65.02
	Rank	57.29	53.01	66.17	67.81	61.07
	LogRank	61.94	63.87	67.57	67.87	65.31
	Entropy	60.15	54.45	63.16	66.98	61.19
	Fast-DetectGPT	31.71	65.68	81.71	84.03	65.83
	Binoculars	2.58	30.00	63.39	99.01	48.75
	ImBD	7.69	51.49	77.05	100.00	59.06
	RoBERTa-base	86.36	96.55	97.26	97.96	94.53
	RoBERTa-large	95.10	98.31	97.40	97.07	96.97
	PC-CAT(Ours)	91.99	97.01	98.03	98.03	96.27
Deepseek-7B	Likelihood	65.23	65.55	69.00	69.60	67.35
	Rank	63.45	63.10	68.78	68.46	65.95
	LogRank	66.67	66.06	67.57	67.87	67.04
	Entropy	59.80	55.56	67.29	68.20	62.71
	Fast-DetectGPT	62.46	76.24	81.38	84.03	76.03
	Binoculars	21.05	51.45	76.61	99.01	62.03
	ImBD	19.28	65.47	74.48	100.00	64.81
	RoBERTa-base	51.49	92.47	88.89	84.17	79.26
	RoBERTa-large	66.07	97.26	93.60	94.67	87.90
	PC-CAT(Ours)	83.90	96.32	95.27	98.36	93.46

Table 15: Detailed detection performance across diverse revision tasks and six LLMs on the XSum dataset.