Enhancing LLM Text Detection with Retrieved Contexts and Logits Distribution Consistency

Zhaoheng Huang, Yutao Zhu*, Ji-Rong Wen, and **Zhicheng Dou***
[1]Gaoling School of Artificial Intelligence, Renmin University of China, Beijing, China
{huangzh,ytzhu,jrwen,dou}@ruc.edu.cn

Abstract

Large language models (LLMs) can generate fluent text, raising concerns about misuse in online comments and academic writing, leading to issues like corpus pollution and copyright infringement. Existing LLM text detection methods often rely on features from the logit distribution of the input text. However, the distinction between the LLM-generated and human-written texts may rely on only a few tokens due to the short length or insufficient information in some texts, leading to minimal and hard-to-detect differences in logit distributions. To address this, we propose HALO, an LLM-based detection method that leverages external text corpora to evaluate the difference in the logit distribution of input text under retrieved human-written and LLM-rewritten contexts. HALO also complements basic detection features and can serve as a plug-and-play module to enhance existing detection methods. Extensive experiments on five public datasets with three widely-used source LLMs show that our proposed detection method achieves state-of-the-art performance in AUROC, both in cross-domain and domain-specific scenarios.

1 Introduction

In recent years, the application of large language models (LLMs) has made remarkable progress, revolutionizing the way many people perform daily tasks. However, this widespread adoption has raised significant concerns regarding academic dishonesty, copyright violations, and the degradation of internet content integrity (Meyer et al., 2023; Karamolegkou et al., 2023; Dai et al., 2023). LLM-generated text may also lead to creativity loss and the spread of misinformation (Su et al., 2024), offering limited value for LLM pre-training and may degrade performance (Shumailov et al., 2024). The fluency of LLM-generated text makes it hard to distinguish from human-written text.

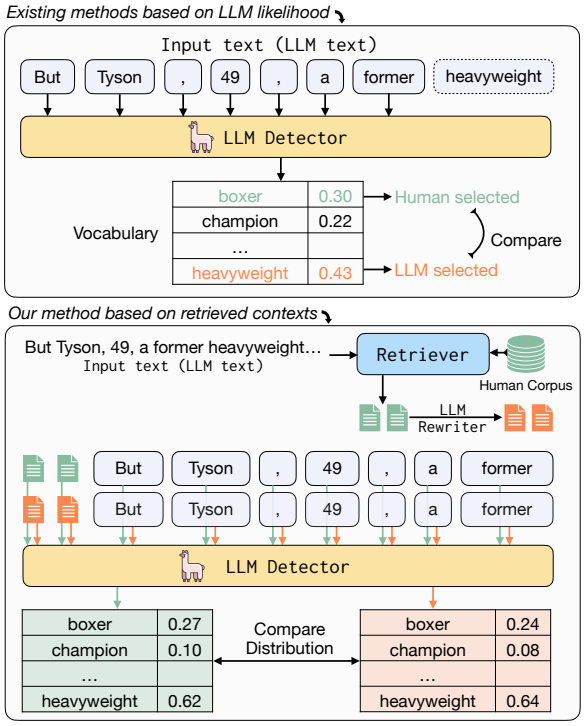


Figure 1: Comparison of existing methods based on LLM-generated text likelihood (upper) versus our proposed method utilizing retrieved context (lower).

Existing methods for detecting LLM-generated texts have traditionally formulated the task as a binary classification problem, and fine-tuned pre-trained language models to identify whether a text is generated by humans or LLMs (Solaiman et al., 2019; Guo et al., 2023; Tian et al., 2023). Although these methods have yielded promising results, their performance often degrades when applied to out-of-domain data, which significantly limits their application (Jawahar et al., 2020). Consequently, recent research explored the use of LLMs for detecting LLM-generated texts (Ippolito et al., 2020; Su et al., 2023; Bao et al., 2023).[1] These methods leverage the difference between logit distributions of LLM-

*Corresponding author.

[1]It is important to note that the LLM used for detection may differ from the LLM that generated the texts.

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 9922–9934
November 4-9, 2025

generated and human-written texts computed by the LLM-based detector, thus achieving effective identification without requiring additional training data (as illustrated in the upper side of Figure 1). Unfortunately, not all texts can provide sufficient information to identify significant differences in logit distributions. In some cases, the distinction between LLM-generated and human texts may hinge on only a few tokens, resulting in marginal differences in logit distribution. This problem is even severe when the detected texts are short, where the limited content may lead to very similar logit distributions.

To address this issue, we propose leveraging an external text corpus to enhance the detection process. Our idea is motivated by two key observations. **First**, the abundance of both human-written and LLM-generated texts in the world offers a wealth of information, which can serve as a reference for the detector to accurately identify the texts. **Second**, common LLMs typically use autoregressive structures. By using different texts as prefixes, these models can evaluate how the logit distribution of the input text changes under different conditions, thus yielding more robust detection features. Based on these insights, we first conduct a preliminary study, which indicates that the predictive distribution of human-written text tends to exhibit greater consistency across two contexts, whereas LLM-generated texts are more dynamically adaptive to changes based on the given context. Then, we design a novel detection method HALO, which leverages the differences in how human and LLM texts respond to retrieved relevant **H**uman-written **A**nd **L**LM-rewritten c**O**ntexts. As shown in the lower part of Figure 1, we first collect a large amount of human-written text from general sources like Wikipedia (Karpukhin et al., 2020) and the MS MARCO (Nguyen et al., 2016) corpus, which exhibit strong human-written features. We also include the Student Essay (Koike et al., 2024b) corpus to study the domain-specific detection scenario. Then, we retrieve human-written relevant texts and their LLM-rewritten versions, pre-constructed offline for detection efficiency. The rewriting process ensures both types of texts remain semantically relevant to the input text, while introducing subtle variations in phrasing and structure. The two types of texts are respectively formed as the contexts for the input text. Finally, we derive a detection feature by comparing the distribution consistency of the input text under these two types of contexts. We use cross-entropy to capture this detection feature and combine it with the logits distribution of input text, to effectively improve the distinction between human and LLM texts.

We conduct experiments across five cross-domain datasets containing texts generated by human and three widely-used LLMs. We compare our approach with four supervised detectors and ten LLM-based detectors. The results demonstrate that HALO achieves state-of-the-art performance in terms of the average AUROC score in both cross-domain and domain-specific scenarios. Further analysis confirms that the retrieval strategy is critical for detection performance. Additionally, since the feature we designed is orthogonal to existing detectors, it can be used as a plug-and-play enhancement for current detection methods. Our contributions are summarized as follows:

(1) We conduct a preliminary study to reveal that the logits distribution of human text is significantly more consistent under human-written and LLM-rewritten relevant contexts. This valuable finding strongly motivates our further investigation.

(2) We propose HALO, a novel approach that retrieves relevant contexts from an external corpus and introduces a fused metric combining cross-entropy and logits to enhance detection accuracy.

(3) We conduct comprehensive experiments across five datasets and against texts generated by three widely-used LLMs, validating the effectiveness of our proposed method. Further analysis demonstrates that HALO adapts well to both cross-domain and domain-specific detection scenarios while exhibiting efficiency and robustness.

2 Related Work

2.1 LLM Text Detection

Based on the backbone of detectors, existing detection methods are categorized into encoder-based and LLM-based approaches.

Encoder-based Detectors primarily train binary classifiers by fine-tuning encoder-based pre-trained language models on large-scale human-LLM text pairs. OpenAI-Det (Solaiman et al., 2019) and HC3 (Guo et al., 2023) fine-tune separate RoBERTa-based models (Liu et al., 2019) using GPT-2 and GPT-3.5 outputs from cross-domain datasets. Their detection performance is further improved through specialized techniques, such as custom loss functions (Tian et al., 2023) or siamese auto-encoders (Huang et al., 2024).

LLM-based Detectors identify differences between model-generated text and human text using an LLM-based detector, without requiring additional training data. Early research proposes several handcrafted, high-quality features based on observations, such as n-gram, likelihood, and the rank of label tokens in the predicted distribution, as well as their combinations (Gehrmann et al., 2019; Verma et al., 2024). Later work focuses on applying token-level perturbations to the input text and measuring the likelihood differences before and after the perturbation (Su et al., 2023; Mitchell et al., 2023; Bao et al., 2023). Building on comparative differences, existing research proposes detection methods such as comparing multiple LLM-generated continuations of the input text (Yang et al., 2023) and analyzing likelihood differences between two LLM-based detectors (Hans et al., 2024; Chen et al., 2025).

In this work, we perform LLM text detection by leveraging the distribution consistency of input text under relevant human-written and LLM-rewritten contexts without any training data.

2.2 Retrieval-Enhanced Detection

Text retrieval is a key task in the field of information retrieval. Existing methods rely on sparse representations (*e.g.*, BM25 algorithm (Robertson et al., 1994)) or model-based dense representations (*e.g.*, BGE and E5 (Xiao et al., 2023; Wang et al., 2022), fine-tuned from the BERT model). Since large-scale human-written and LLM-generated texts exhibit distinct linguistic patterns, recent LLM text detection methods incorporate retrieval modules that compare input text against large-scale reference corpora, containing both human-written and LLM-generated texts. Such retrieval-enhanced methods improve robustness of detectors across diverse domains and attacks (Krishna et al., 2023; Sadasivan et al., 2023), or enhance the LLM-based detection performance through adversarial in-context learning (Koike et al., 2024b) and improve detection interpretability (Koike et al., 2025). In this work, we retrieve relevant human texts and their LLM-rewritten versions and use them as the context for the input text. Then, we enhance the detection performance with the distribution consistency of the input text under two types of contexts.

3 Preliminaries

To analyze the impact of relevant texts as contexts on the logit distribution of input text, we conduct a preliminary study to illustrate the distinguishing logit distribution differences between human texts and LLM texts under the influence of retrieved human-written and LLM-rewritten contexts. Specifically, we first collect human texts and LLM texts from dataset D. Given an input text $T = \{t_1, \cdots, t_l\} \in D$ with l tokens, we measure the generation probability of the i-th token under the relevant text P retrieved from a corpus C as:

$$\text{Prob}(t_i, P) = \log p(t_i | P; t_{<i}).$$

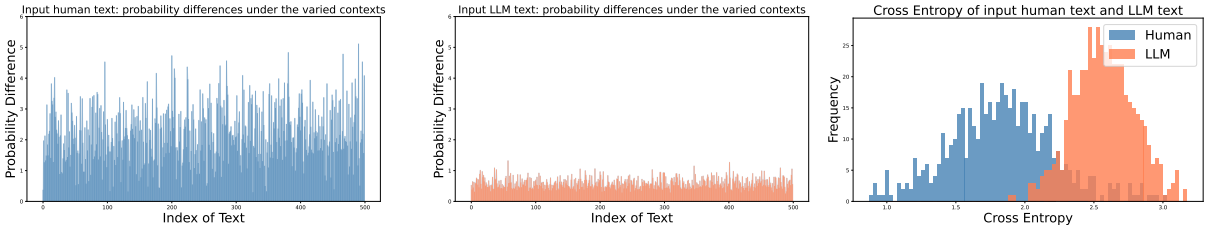
Then, we measure the probability difference $\text{PD}(t_i)$ under the relevant human texts P and LLM texts $\tilde{P}$. For each text T with l_T tokens, we measure the average probability difference as follows:

$$\begin{aligned} \text{PD}(t_i) &= \left| \text{Prob}(t_i, P) - \text{Prob}(t_i, \tilde{P}) \right|. \\ \text{PD}(T) &= \frac{1}{l_T} \sum_{i=1}^{l_T} \text{PD}(t_i). \end{aligned} \tag{1}$$

The value of the probability difference implies the average likelihood shift of input text T under varied contexts. Further, inspired by previous studies (Su et al., 2023; Bao et al., 2023), we leverage the prediction distribution over the vocabulary, which provides more robust detection features than a single token probability. To better capture the difference in logit distribution consistency, we apply cross entropy as follows:

$$\begin{aligned} \text{CE}(t_i) &= -\sum_{j=1}^{V} p(t_{ij} | P; t_{<i}) \log p(t_{ij} | \tilde{P}; t_{<i}), \\ \text{CE}(T) &= \frac{1}{l_T} \sum_{i=1}^{l_T} \text{CE}(t_i), \end{aligned}$$

where V is the vocabulary size, and t_{ij} represents the j-th token at the i-th position of the input text sequence T. The average of cross-entropy at each position, $\text{CE}(T)$, captures the logit distribution differences under human-written and LLM-rewritten contexts. We randomly select 500 human texts and 500 paired texts generated by GPT-4o-mini-2024-07-18 from the SQuAD (Rajpurkar et al., 2016) dataset. We concatenate the top 3 relevant texts P retrieved by BGE-base (Xiao et al., 2023) from the combination of Wiki and MS MARCO corpora, and use LLaMA3.1-8B-Instruct to generate LLM-rewritten texts $\tilde{P}$. From Figure 2a and 2b, we observe that human and LLM texts not only differ in likelihood measurements, but also exhibit distinct



(a) Prob. differences of human texts. (b) Prob. differences of LLM texts. (c) Cross entropy of input texts.

Figure 2: (a) and (b): Probability difference, and (c): Cross entropy of input text across varied contexts.

differences under various contexts, which can be used as a valuable feature for detection. Moreover, as illustrated in Figure 2c, the logit distribution consistency measured by cross entropy allows us to clearly distinguish between LLM and human text, where the LLM texts exhibit larger distribution differences with higher cross-entropy. Therefore, the inverse of cross-entropy (ICE) reflects the logit distribution consistency of input text T:

$$\text{ICE}(T) = \frac{1}{\text{CE}(T)}. \tag{2}$$

4 Methodology

4.1 Overview

Given an input text $T = \{t_1, \cdots, t_l\}$ composed of l tokens, the task is to predict a label y to indicate whether T is a human-written text ($y = 0$) or an LLM-generated text ($y = 1$).

In our method, we utilize an external corpus C and a retriever R to select the top k relevant texts $P_{1:k} = R_C(T, k)$ for a given text T. We then obtain the LLM-rewritten versions $\tilde{P}_{1:k}$, which are generated by an LLM during the offline stage. Based on these two kinds of texts, we compute a detection score $s(T, P_{1:k}, \tilde{P}_{1:k})$, where a higher score indicates a greater possibility that T is generated by an LLM. The final prediction is made by comparing $s(T, P_{1:k}, \tilde{P}_{1:k})$ to a predefined threshold ϵ, classifying the text as LLM-generated ($y = 1$) if $s(T, P_{1:k}, \tilde{P}_{1:k}) > \epsilon$.[2] In our preliminary study (Section 3), we demonstrate that incorporating $P_{1:k}$ and $\tilde{P}_{1:k}$ as context inputs reveals a significant difference in the logit distribution between human text and LLM text. Building on this, we introduce this feature to enhance detection accuracy.

4.2 Relevant Text Retrieval

To obtain effective contexts for detecting whether the input text T is generated by a human or an LLM, we construct a corpus C composed of human-written texts. Specifically, we select texts from two general sources: (1) Wikipedia (Karpukhin et al., 2020) (December 2018 dump), providing high-quality human knowledge, and (2) MS MARCO (Nguyen et al., 2016), consisting of web documents in 2016, offering real-world human writing styles. We also consider the domain-specific source, the Student essays (Koike et al., 2024b), representing academic human writing. With the corpus C, we employ a retriever R to obtain k passages relevant to the input text T as follows:

$$P_{1:k} = \{P_1, \cdots, P_k\} = R_C(T, k).$$

We utilize a retriever because it ensures relevance between the input texts and the retrieved texts, which is crucial as irrelevant contexts may introduce noise and disrupt the detection process. Note that our HALO is flexible with any *off-the-shelf* retriever R. This includes both sparse retrievers such as BM25 (Robertson et al., 1994) or dense retrievers such as BGE (Xiao et al., 2023). We use Faiss (Johnson et al., 2019) to speed up dense retrieval. The retrieved passages $P_{1:k}$ are then concatenated with the input text as the human-written relevant context for the subsequent detection.

4.3 Relevant Text Rewriting

Previous studies (Dai et al., 2023) have shown that human-written texts and their LLM-rewritten counterparts exhibit differences in linguistic structures and phrasing, while maintaining similarity in semantics. Building on this observation, we use both human-written relevant texts and rewritten versions as extra features for detection. To obtain LLM-rewritten texts, we use LLaMA3.1-8B-Instruct (AI@Meta, 2024) as the rewriter, which is defined as:

$$\tilde{P}_{1:k} = \{\tilde{P}_1, \cdots, \tilde{P}_k\} = \text{Rewriter}(P_{1:k}).$$

[2]The threshold can be determined using Youden's J statistic, which leverages TPR and FPR values from the ROC curve.

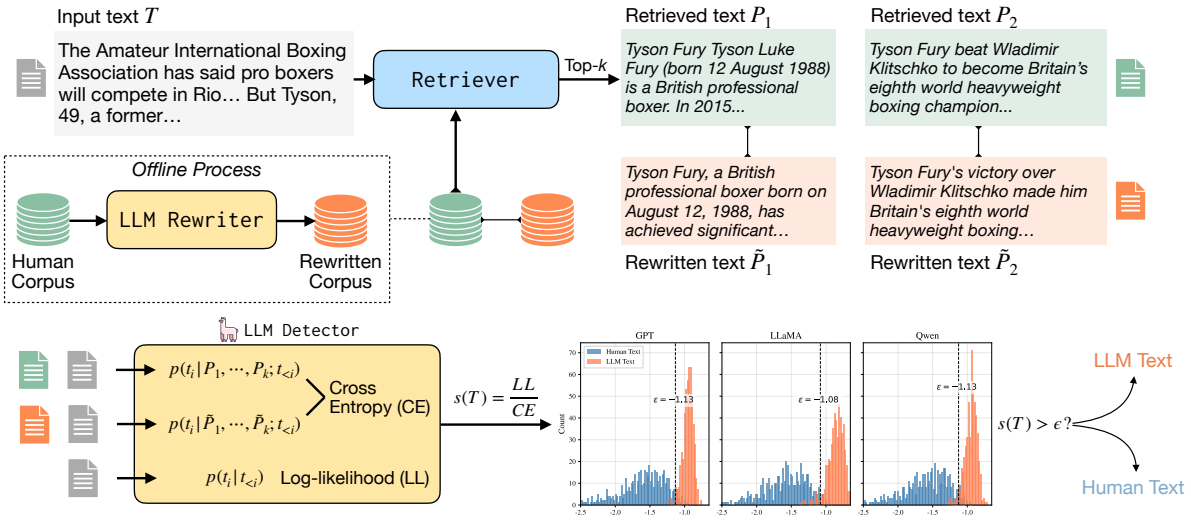


Figure 3: Overview of our proposed detection method HALO. In **offline preparation process**, human-written texts and their LLM-rewritten versions are collected to build different yet relevant contexts. In **online deployment process**, these contexts are retrieved for the input text, and the consistency is measured via cross entropy to perform LLM text detection.

We pre-compute the rewrites of human texts using vLLM (Kwon et al., 2023) during an offline stage and store them on the disk as a cache. This approach eliminates the need for real-time rewriting during the detection phase, thus significantly enhancing detection efficiency.[3]

4.4 Detection Pipeline

Based on the retrieved relevant human texts and LLM-rewritten texts, we propose our detection pipeline, which is shown in Figure 3. Concretely, we first retrieve the top $k = 3$ relevant human texts $P_{1:3}$, and collect their LLM-rewritten versions $\tilde{P}_{1:3}$ (by generation or from cache). Then, we respectively use the two kinds of texts to form two contexts for the input text with proper prompts (provided in Appendix A). Next, we can compute the inverse of cross entropy $\mathrm{ICE}(T)$ to measure the logit distribution consistency based on the contexts $P_{1:3}$ and $\tilde{P}_{1:3}$, as described in Equation 2. The consistency feature is then applied to enhance the current detection feature, and the final detection score $s(T)$ is formed as:

$$s(T) = \mathrm{ICE}(T) \times \frac{1}{l}\sum_{i=1}^{l} \log p(t_i|t_{<i}). \quad (3)$$

[3]Advanced caching strategies, such as selectively caching the most frequently retrieved rewritten texts, can further accelerate the cache-building process. However, we omit the discussion of this as it is beyond the scope of this paper.

5 Experiment

5.1 Datasets

In line with previous studies (Mitchell et al., 2023; Bao et al., 2023; Koike et al., 2024b), we select five datasets to conduct experiments: HC3 (Guo et al., 2023), XSum (Narayan et al., 2018), WritingPrompts (Fan et al., 2018), SQuAD (Rajpurkar et al., 2016), and Essay (Koike et al., 2024b). We employ three LLMs for text generation: GPT-4o-mini-2024-07-18, Meta-Llama-3.1-70B-Instruct, and Qwen2-72B-Instruct. The details of dataset construction and statistics of both human-written texts and LLM-generated texts are presented in Appendix B.

5.2 Baselines and Metric

We compare our proposed pipeline with both supervised and LLM-based detectors.

• *Supervised Detectors.* These methods fine-tune PLMs such as RoBERTa (Liu et al., 2019) on massive labeled texts for detection. Following experimental settings of Fast-DetectGPT (Bao et al., 2023), we compare our method with publicly released detectors: **OaiDet-base/large** (Solaiman et al., 2019), **HC3** (Guo et al., 2023), and **MPU** (Tian et al., 2023).

• *LLM-based Detectors.* These methods apply decoder-only scoring LLMs to identify LLM-generated texts based on the output logit distribution of input text. Four handcrafted logit distri-

Source	GPT-4o-mini-2024-07-18					Meta-Llama-3.1-70B-Instruct					Qwen2-72B-Instruct				
Datasets	**HC3**	**XS**	**WP**	**SQ**	**Avg.**	**HC3**	**XS**	**WP**	**SQ**	**Avg.**	**HC3**	**XS**	**WP**	**SQ**	**Avg.**
Supervised Detectors															
OaiDet-base	73.53	60.29	46.22	48.07	57.03	91.18	87.94	82.46	87.68	87.32	73.28	59.77	69.01	49.82	62.97
OaiDet-large	69.30	62.01	36.07	55.93	55.83	84.57	80.95	56.13	82.07	75.93	66.78	56.26	47.91	52.88	55.96
HC3	96.87	60.97	96.38	63.15	79.34	98.64	93.08	99.36	90.40	95.37	97.49	82.53	98.53	74.05	88.15
MPU	**98.85**	80.33	41.21	73.93	73.58	97.43	92.28	56.93	92.91	84.89	**98.19**	83.74	62.62	77.53	80.52
LLM-based Detectors															
Entropy	89.80	18.83	60.68	60.35	57.42	97.05	40.57	55.24	71.66	66.13	94.18	26.64	59.81	68.61	62.31
Likelihood	95.90	39.37	84.51	78.97	74.69	99.80	84.58	94.06	93.59	93.01	98.07	51.22	88.76	85.05	80.78
Rank	92.40	54.97	89.38	76.47	78.31	95.54	67.86	89.09	79.56	83.01	93.16	57.75	93.46	79.04	80.85
LogRank	95.98	41.23	82.56	78.97	74.69	99.84	86.01	94.11	94.12	93.52	98.11	53.42	89.07	85.82	81.61
LRR	92.44	48.81	71.45	75.51	72.05	99.53	85.40	92.21	92.70	92.46	96.82	59.04	88.17	84.42	82.11
DNA-GPT	80.92	48.60	64.32	77.00	67.71	95.34	88.99	91.29	95.22	92.71	88.80	54.16	79.50	82.77	76.31
DetectGPT	72.50	32.47	57.41	73.98	59.09	94.60	47.08	49.04	84.57	68.82	85.22	39.09	56.12	78.89	64.83
NPR	67.67	32.00	46.94	65.15	52.94	95.14	53.84	42.56	84.21	68.94	83.47	40.66	50.46	73.55	62.04
OUTFOX	69.40	82.90	59.00	72.50	70.95	68.90	72.70	58.90	69.10	67.40	71.10	79.40	62.00	69.90	70.60
Fast-DetectGPT	94.48	83.44	**99.13**	93.44	92.62	**99.97**	97.58	**99.88**	98.61	99.01	96.97	85.53	99.37	94.20	94.02
HALO (Ours)	93.31	**83.60**	98.93	**99.11**	**93.74**	99.74	**97.81**	99.78	**99.37**	**99.18**	98.14	**86.48**	**99.43**	**98.87**	**95.73**

Table 1: Experimental results across four cross-domain datasets. The best result is in **bold**, and the second best result is underlined. "XS", "WP", and "SQ" are abbreviations of "XSum", "WritingPrompts", and "SQuAD".

Source	GPT	LLaMA	Qwen	**Avg.**
Datasets	**Essay**			
Supervised Detectors				
HC3	76.55	98.93	99.49	91.66
MPU	98.52	99.93	99.99	99.48
LLM-based Detectors				
Likelihood	94.03	99.82	99.95	97.93
Rank	90.61	97.44	97.79	95.28
LogRank	90.80	99.75	99.93	96.83
DetectGPT	74.34	90.65	90.75	85.25
OUTFOX	91.10	86.10	85.90	87.70
Fast-DetectGPT	99.27	**100.00**	99.99	99.75
HALO (Ours)	**99.83**	**100.00**	**100.00**	**99.94**

Table 2: Experimental results of the **Essay** dataset.

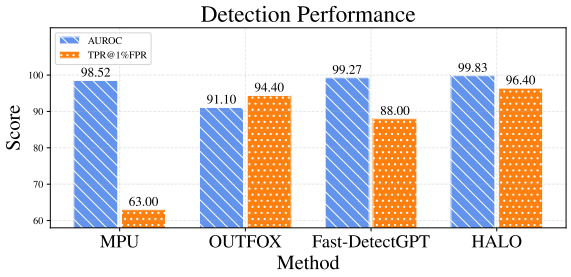


Figure 4: Comparison of detection performance measured by AUROC and TPR@1%FPR on the **Essay** dataset generated by GPT-4o-mini.

bution features are considered (Gehrmann et al., 2019; Ippolito et al., 2020): **Entropy**, **Likelihood**, **Rank**, and **LogRank**. **LRR** (Su et al., 2023) combines the above basic features for detection. We also compare with completion-based and retrieval-enhanced detection baselines: **DNA-GPT** (Yang et al., 2023) and **OUTFOX** (Koike et al., 2024b), as well as token perturbation-based baselines: **DetectGPT** (Mitchell et al., 2023), **NPR** (Su et al., 2023), and **Fast-DetectGPT** (Bao et al., 2023).

We follow Mitchell et al. (2023); Bao et al. (2023) and evaluate with the AUROC metric. A higher AUROC indicates a better performance, and we report its $\times 100$ value for better presentation. The details are shown in Appendix C.

5.3 Experimental Results

The experimental results are shown in Table 1 (cross-domain scenario, where the corpora are the combination of Wiki and MS MARCO) and Table 2 (domain-specific scenario, where the corpus is the Essay). Our findings are as follows:

(1) Among all detection methods, our approach achieves the highest AUROC scores. When detecting texts generated by GPT-4o-mini, our method shows a 1.12% absolute improvement over the state-of-the-art baseline, Fast-DetectGPT (Bao et al., 2023), highlighting the effectiveness of our approach. Fast-DetectGPT improves performance by enhancing the likelihood detection feature with random input perturbations. In contrast, our method leverages the consistency of logit distributions in the input text, achieving an average absolute improvement of 1.00% in AUROC for

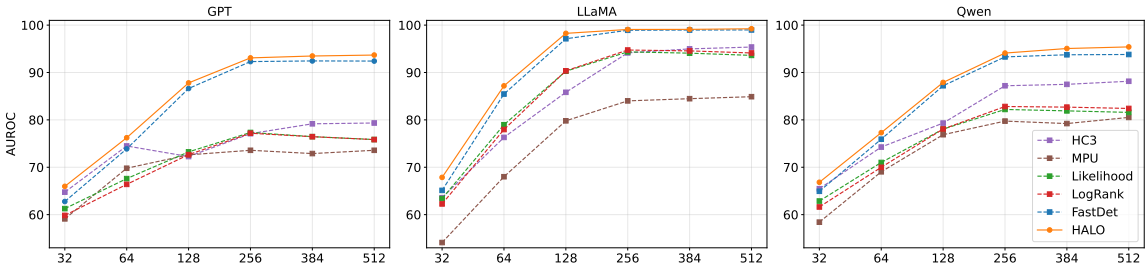


Figure 5: Performance of different methods on various input lengths.

cross-domain and 0.19% for domain-specific detection on GPT-4o-mini. We also report the True Positive Rate (TPR) under the constraint of a False Positive Rate (FPR) of 1% in Figure 4, showing that our proposed method improves TPR@1%FPR by 8.40% over Fast-DetectGPT when detecting essays generated by GPT-4o-mini, demonstrating superior detection reliability under strict false positive control. These results further demonstrate the advantages of our approach and its applicability across different source LLMs and detection scenarios.

(2) Our method outperforms OUTFOX (Koike et al., 2024b) by 26.6% in cross-domain scenario and 12.2% in domain-specific scenario. OUTFOX is the most relevant retrieval-enhanced baseline method, which uses the relevant texts as demonstrations for in-context learning, and its prediction heavily relies on the LLM output, making it highly dependent on the in-context learning ability to accurately identify LLM texts. Our proposed method uses the logit distribution consistency under relevant texts, which enables a more stable and robust detection feature.

(3) Compared to supervised detectors, our method demonstrates superior generalization capabilities across datasets. On the HC3 dataset, supervised detectors such as HC3 and MPU outperform our proposed method HALO, because they were trained on the HC3 training set and thus achieve high performance on its test set. However, their average detection performance declines when applied to texts from other datasets.

5.4 Further Analysis

We further conduct a comprehensive series of experiments, mainly focusing on detecting texts generated by GPT-4o-mini.

Detection Robustness under Text Length and Paraphrasing Attack The input text length usually affects the detection performance (Tian et al., 2023). Hence, we truncate the human and LLM texts in each dataset into lengths from 32 to 512 tokens and compare the performance with several baselines. The experimental results are shown in Figure 5. Our experiments reveal that detection performance degrades for inputs shorter than 128 tokens, as limited text length provides insufficient detection features. Conversely, when more context is provided, all methods can perform better, among which our HALO achieves the best performance. These results validate our assumptions that incorporating more texts as context can effectively improve the detection accuracy. Inspired by prior work (Bao et al., 2023; Sadasivan et al., 2023), we also analyze the robustness under paraphrasing attack, and use a T5-based paraphraser to paraphrase LLM texts before detection.[4] The paraphrasing prompt follows the format:

```
paraphrase: <LLM text> </s>
```

As shown in Table 3, all detection methods exhibit performance degradation under paraphrasing attacks, consistent with findings from Bao et al. (2023). Notably, our proposed method HALO demonstrates superior robustness, achieving the highest detection performance. It indicates the importance of the logits distribution consistency in maintaining detector robustness against a paraphrasing attack.

Source	GPT-4o-mini-2024-07-18				
Datasets	**HC3**	**XS**	**WP**	**SQ**	**Avg.**
MPU	**83.89**	**74.88**	60.56	63.21	70.64
Likelihood	55.75	39.57	32.64	40.05	42.00
LogRank	55.90	30.60	32.43	40.74	39.92
Fast-DetectGPT	67.23	48.51	67.56	60.98	61.07
HALO (ours)	70.24	73.20	**70.23**	**88.22**	**75.47**

Table 3: Robustness of LLM text detectors under paraphrasing attack.

[4] https://huggingface.co/Vamsi/T5_Paraphrase_Paws

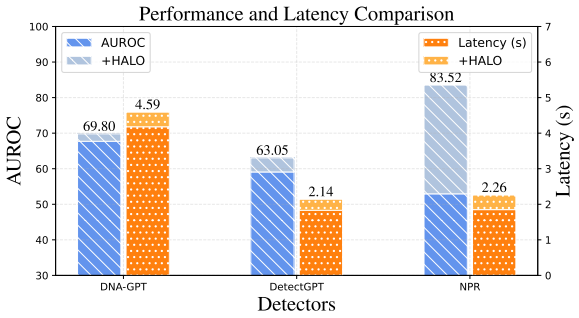


Figure 6: Performance and time latency when combining HALO with existing approaches.

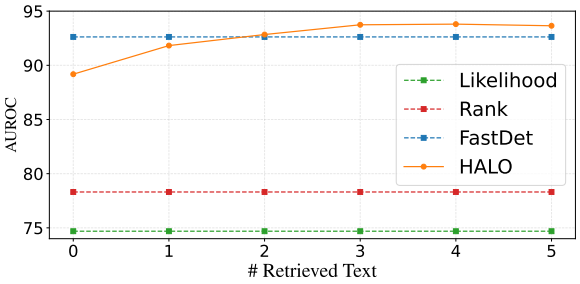


Figure 7: Performance with different numbers of retrieved texts.

Efficiency and Combination with Other Methods A key advantage of our approach is its orthogonality to existing methods, which means it can be seamlessly combined with them and further boost performance. To examine this, we apply a modified feature-merging strategy (Ma and Wang, 2024) defined as:

$$w(x) = \begin{cases} e^{u(x)} \cdot v(x), & \text{if } v(x) \geq 0, \\ e^{-u(x)} \cdot v(x), & \text{if } v(x) < 0, \end{cases} \quad (4)$$

where $u(x)$ corresponds to our detection score (Equation (3)), and $v(x)$ denotes the score from other methods. We omit the same basic features from two methods, such as Likelihood. The experimental results in Figure 6 indicate that integrating our method leads to performance improvements ranging from 3.09% to 57.76%. This demonstrates the compatibility of HALO with existing methods. Additionally, our method is lightweight, introducing only minimal latency (including <0.1 seconds of online retrieval), ensuring efficiency in practical use. In contrast, OUTFOX requires >30 seconds for each input text due to online generation of adversarial samples for in-context learning.

Impact of Retrieved Text Amount In our approach, we retrieve the top $k = 3$ human-written relevant texts from the corpus and combine them with the corresponding LLM-rewritten texts for detection. To explore the impact of the number of retrieved texts on detection performance, we vary the number of retrieved texts ($0 \leq k \leq 5$) and compute the average AUROC score over four datasets. The results are shown in Figure 7. We can observe that the retrieved texts are important in our methods. Without any retrieved text ($k = 0$), HALO achieves an average AUROC score of 89.18. In this case, HALO degenerates to the ratio of likelihood and entropy. While this performance is slightly lower than Fast-DetectGPT, it remains superior to other methods. However, when retrieved texts are added, our method shows significant improvements, achieving state-of-the-art performance. Notably, the optimal performance is achieved at $k = 3$, beyond which performance plateaus. This implies that the introduced noise in texts with lower similarity does not contribute to further performance improvement.

#	Variant	HC3	XS	WP	SQ	Avg.
	Full model					
1		93.31	83.60	98.93	99.11	93.74
	Ablation Study					
2	*w/o* Human	87.55	75.36	98.47	96.77	89.54
3	*w/o* Rewritten	92.20	79.21	98.66	97.18	91.81
4	*w/o* Relevance	89.37	75.47	97.98	95.18	89.50
	Different Retrieval Corpora					
5	Only MS	92.92	82.33	98.48	93.48	91.80
6	Only Wiki	92.51	82.83	98.04	98.94	93.08
	Different Retrievers and Rewriters					
7	BM25	92.29	80.21	98.02	98.59	92.28
8	Qwen2.5-7B-Inst	93.18	83.70	98.67	99.23	93.70

Table 4: Performance of HALO under different settings.

Impact of Context and Pipeline Settings We conduct experiments to investigate the impact of different pipeline settings, and we have the following findings: **(1) Both contexts are crucial for detection performance.** Our ablation study (Table 4 #2-3) demonstrates that performance drops when removing each context, while retrieved texts show marginally greater importance than rewritten texts. **(2) Randomly retrieved context causes a performance drop.** Table 4 (#4) examines the scenario where the retriever fails to find truly relevant contexts by replacing retrieved texts with randomly sampled texts from the top-20 retrieval candidates. The observed performance drop across four datasets highlights the important role of the retriever and context relevance. **(3) Wiki and**

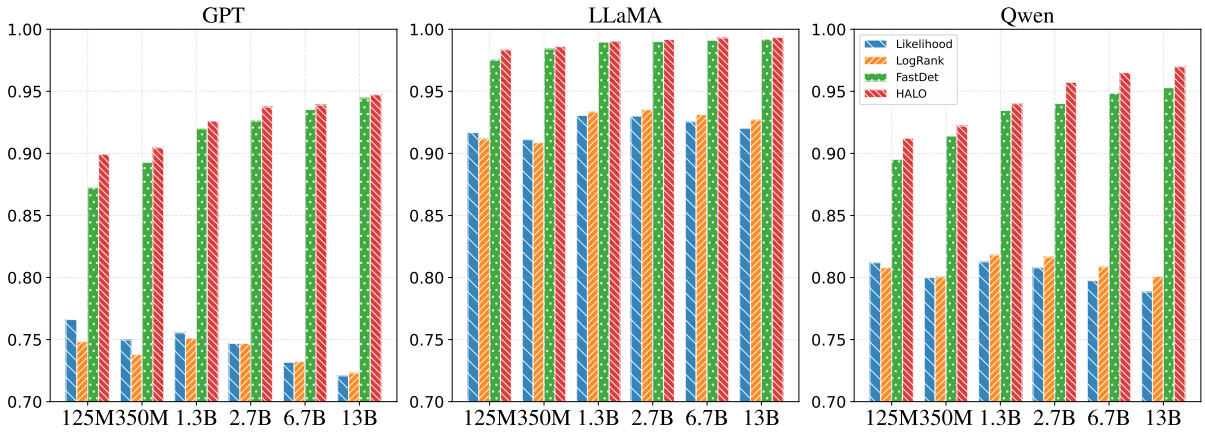


Figure 8: Performance with LLM detectors of six different parameter scales.

MSMARCO corpora both contribute to cross-domain detection. Table 4 (#5-6) shows a performance drop when either corpus is removed, highlighting the necessity of general human features for effective and robust detection. Dense retrievers (BGE) outperform sparse methods (BM25), as they capture more semantically relevant contexts, further enhancing detection. **(4) The selection of LLM rewriter does not significantly affect the results.** Table 4 (#8) shows the consistency of performance when the rewriter is replaced with Qwen2.5-7B-Instruct. This is aligned with existing work (Dai et al., 2023), which demonstrated strong consistency across various LLMs in text rewriting tasks. Despite these variations, our method shows consistent performance, demonstrating its robustness and broad applicability.

Impact of Parameter Scales Larger language models usually have strong capabilities, leading us to hypothesize that they may also perform better in LLM-generated text detection. To verify this hypothesis, we consider six LLMs of different sizes: OPT-{125M, 350M, 1.3B, 2.7B, 6.7B, 13B}. As shown in Figure 8, the performance of HALO and Fast-DetectGPT shows a consistent improvement as model size increases. This confirms our assumption that larger models enhance detection capabilities. However, methods like LogRank and Likelihood slightly decrease when model sizes exceed 1.3B. The potential reason is that both HALO and Fast-DetectGPT leverage multiple features, mitigating bias toward any single feature, whereas LogRank and Likelihood are more limited in their feature scope. Besides, while larger detector models deliver better results, it is important to also note the trade-off in increased inference time.

6 Conclusion

In this work, we propose HALO, an LLM text detection method which measures logit distribution consistency of the input text under human-written relevant contexts and LLM-rewritten versions. HALO achieves the best AUROC performance across datasets and source LLMs without the requirement of training data, both in cross-domain and domain-specific scenarios. HALO can also be served as an orthogonal plug-and-play plugin with other detectors to improve their detection accuracy.

Limitations

We acknowledge some limitations in our work: (1) Although existing retrieval-enhanced detection methods (introduced in Section 2.2) can be applicable to both cross-domain and domain-specific detection scenarios, such as detecting AI-generated essays (Koike et al., 2024b), the reliance on relevant human-written texts is a common limitation for such detection methods (Dai et al., 2023; Koike et al., 2024b, 2025). For example, when detecting texts in a different language, the retrieval module used (including the retriever and corpus) should also switch to the same language to ensure detection accuracy. (2) In constructing the dataset for detection, we followed previous work (Mitchell et al., 2023; Bao et al., 2023) by generating corresponding LLM texts through direct question answering and text completion. However, the difficulty of detecting LLM-generated text may be related to the instruction used to generate detected text (Koike et al., 2024a). Future work could further explore the robustness of different detection methods under diversified prompts and instructions.

Ethical Considerations

Our method is designed to detect LLM-generated texts and has outperformed existing methods in terms of AUROC. This task has garnered widespread attention because it can be used to address societal issues such as fake news, essays, and online reviews, thereby enhancing the ability to govern and regulate the development of artificial intelligence. Additionally, existing work (Chen et al., 2024) suggests that LLM-generated text can have certain negative effects in scenarios such as data augmentation and retrieval-augmented generation (RAG), which further emphasizes the need for reliable detection methods. However, it must be acknowledged that our method still carries the risk of detection failure, and in practical use, users must approach the detection results with caution and assume responsibility for any associated risks.

Acknowledgments

This work was supported in part by National Science and Technology Major Project No. 2022ZD0120103, National Natural Science Foundation of China No. 62402497 and 62272467, and Outstanding Innovative Talents Cultivation Funded Programs 2023 of Renmin University of China.

References

AI@Meta. 2024. Llama 3 model card.

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

Xiaoyang Chen, Ben He, Hongyu Lin, Xianpei Han, Tianshu Wang, Boxi Cao, Le Sun, and Yingfei Sun. 2024. Spiral of silence: How is large language model killing information retrieval?—A case study on open domain question answering. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14930–14951, Bangkok, Thailand. Association for Computational Linguistics.

Zhihui Chen, Kai He, Yucheng Huang, Yunxiao Zhu, and Mengling Feng. 2025. Divscore: Zero-shot detection of llm-generated text in specialized domains. *CoRR*, abs/2506.06705.

Sunhao Dai, Yuqi Zhou, Liang Pang, Weihao Liu, Xiaolin Hu, Yong Liu, Xiao Zhang, and Jun Xu. 2023. Llms may dominate information access: Neural retrievers are biased towards llm-generated texts. *CoRR*, abs/2310.20501.

Angela Fan, Mike Lewis, and Yann N. Dauphin. 2018. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 889–898. Association for Computational Linguistics.

Sebastian Gehrmann, Hendrik Strobelt, and Alexander M. Rush. 2019. GLTR: statistical detection and visualization of generated text. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28 - August 2, 2019, Volume 3: System Demonstrations*, pages 111–116. Association for Computational Linguistics.

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. *CoRR*, abs/2301.07597.

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 *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

Guanhua Huang, Yuchen Zhang, Zhe Li, Yongjian You, Mingze Wang, and Zhouwang Yang. 2024. Are AI-generated text detectors robust to adversarial perturbations? In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6005–6024, Bangkok, Thailand. Association for Computational Linguistics.

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, ACL 2020, Online, July 5-10, 2020*, pages 1808–1822. Association for Computational Linguistics.

Ganesh Jawahar, Muhammad Abdul-Mageed, and Laks V. S. Lakshmanan. 2020. Automatic detection of machine generated text: A critical survey. In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pages 2296–2309. International Committee on Computational Linguistics.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547.

Antonia Karamolegkou, Jiaang Li, Li Zhou, and Anders Søgaard. 2023. Copyright violations and large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 7403–7412. Association for Computational Linguistics.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Ryuto Koike, Masahiro Kaneko, Ayana Niwa, Preslav Nakov, and Naoaki Okazaki. 2025. Exagpt: Example-based machine-generated text detection for human interpretability. *CoRR*, abs/2502.11336.

Ryuto Koike, Masahiro Kaneko, and Naoaki Okazaki. 2024a. How you prompt matters! even task-oriented constraints in instructions affect llm-generated text detection. In *Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Florida, USA, November 12-16, 2024*, pages 14384–14395. Association for Computational Linguistics.

Ryuto Koike, Masahiro Kaneko, and Naoaki Okazaki. 2024b. Outfox: Llm-generated essay detection through in-context learning with adversarially generated examples. In *Proceedings of the 38th AAAI Conference on Artificial Intelligence*, Vancouver, Canada.

Kalpesh Krishna, Yixiao Song, Marzena Karpinska, John Wieting, and Mohit Iyyer. 2023. Paraphrasing evades detectors of ai-generated text, but retrieval is an effective defense. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

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

Shixuan Ma and Quan Wang. 2024. Zero-shot detection of llm-generated text using token cohesiveness. *arXiv preprint arXiv:2409.16914*.

Jesse G. Meyer, Ryan J. Urbanowicz, Patrick C. N. Martin, Karen O'Connor, Ruowang Li, Pei-Chen Peng, Tiffani J. Bright, Nicholas P. Tatonetti, Kyoung-Jae Won, Graciela Gonzalez-Hernandez, and Jason H. Moore. 2023. Chatgpt and large language models in academia: opportunities and challenges. *BioData Min.*, 16(1).

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, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 24950–24962. PMLR.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 1797–1807. Association for Computational Linguistics.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A human generated machine reading comprehension dataset. In *Proceedings of the Workshop on Cognitive Computation: Integrating neural and symbolic approaches 2016 co-located with the 30th Annual Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain, December 9, 2016*, volume 1773 of *CEUR Workshop Proceedings*. CEUR-WS.org.

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, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 2383–2392. The Association for Computational Linguistics.

Stephen E. Robertson, Steve Walker, Susan Jones, Micheline Hancock-Beaulieu, and Mike Gatford. 1994. Okapi at TREC-3. In *Proceedings of The Third Text REtrieval Conference, TREC 1994, Gaithersburg, Maryland, USA, November 2-4, 1994*, volume 500-225 of *NIST Special Publication*, pages 109–126. National Institute of Standards and Technology (NIST).

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

Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Nicolas Papernot, Ross J. Anderson, and Yarin Gal. 2024. AI models collapse when trained on recursively generated data. *Nat.*, 631(8022):755–759.

Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, and Jasmine Wang. 2019. Release strategies and the social impacts of language models. *CoRR*, abs/1908.09203.

Jinyan Su, Claire Cardie, and Preslav Nakov. 2024. Adapting fake news detection to the era of large language models. In *Findings of the Association for Computational Linguistics: NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pages 1473–1490. Association for Computational Linguistics.

Jinyan Su, Terry Yue 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, Singapore, December 6-10, 2023*, pages 12395–12412. Association for Computational Linguistics.

Yuchuan Tian, Hanting Chen, Xutao Wang, Zheyuan Bai, Qinghua Zhang, Ruifeng Li, Chao Xu, and Yunhe Wang. 2023. Multiscale positive-unlabeled detection of ai-generated texts. *CoRR*, abs/2305.18149.

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), NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pages 1702–1717. Association for Computational Linguistics.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2023. C-pack: Packaged resources to advance general chinese embedding. *Preprint*, arXiv:2309.07597.

Xianjun Yang, Wei Cheng, Linda R. Petzold, William Yang Wang, and Haifeng Chen. 2023. DNA-GPT: divergent n-gram analysis for training-free detection of gpt-generated text. *CoRR*, abs/2305.17359.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. Opt: Open pre-trained transformer language models. *Preprint*, arXiv:2205.01068.

A Context Formation

For top-k retrieved human-written texts, the context is denoted as:

> I am providing you with k human-written passages that are relevant to the input text.
>
> Human-written passages: [1] {human_passages[1]} ··· [k] {human_passages[k]}
>
> Input text:

For top-k retrieved LLM-rewritten texts, the context is denoted as:

> I am providing you with k model-generated passages that are relevant to the input text.
>
> Model-generated passages: [1] {rewritten_passages[1]} ··· [k] {rewritten_passages[k]}
>
> Input text:

Source	Params	Avg. Length				
		HC3	XSum	WP	SQuAD	Essay
Human	-	226.49	480.96	717.77	162.47	456.99
GPT	-	262.84	221.57	520.89	244.25	473.66
LLaMA	70B	423.43	230.63	324.93	235.03	484.33
Qwen	72B	362.14	248.54	481.16	290.94	478.70

Table 5: Statistics of five datasets and three source LLMs. The average length is calculated with LLaMA-3's tokenizer.

```
1. HC3 Dataset
<<System Prompt>>
You are a knowledgeable assistant.

<<User>>
Please write a passage answering this question: {Question}

2. XSum Dataset
<<System Prompt>>
You are an English News writer.

<<User>>
Please write a passage starting exactly with: {News_Prefix}

3. WritingPrompts Dataset
<<System Prompt>>
You are an English Fiction writer.

<<User>>
Please write a passage starting exactly with: {Fiction_Prefix}

4. SQuAD Dataset
<<System Prompt>>
You are an English Wikipedia writer.

<<User>>
Please write a passage starting exactly with: {Wiki_Prefix}

5. Essay Dataset
<<User>>
{Essay_Question}
```

Figure 9: The prompts for generating LLM texts.

B Dataset

In this paper, we study and compare the performance of detection methods with five datasets: (1) **HC3**, (2) **XSum**, (3) **WritingPrompts**, (4) **SQuAD**, and (5) **Essay**.

For datasets (1) to (4), we randomly sample 500 instances and follow Bao et al. (2023) and Tian et al. (2023) to generate corresponding LLM texts. Specifically, for the HC3 dataset, we use the prompt provided in the dataset to generate texts. For the other three datasets, we use the first 30 tokens in the human text and use a prompt to generate texts. The details of the prompt are given in Figure 9. We leverage the Wiki and MS MARCO corpora for enhancing detection. For dataset (5), we perform a domain-specific detection scenario by detecting 500 essays in the test set, and we use 14,400 samples in the training set to form the essay corpus for enhancing detection. We also provide the details of the statistics of four datasets and three source LLMs in Table 5.

C Implementation Details

Implementation. All experiments are conducted on NVIDIA A800 GPUs. We use OPT-2.7B (Zhang et al., 2022) as the LLM detector. For our method, BGE-base (Xiao et al., 2023) is used as the default retriever to retrieve top-3 relevant human-rewritten texts. We truncate each passage in the corpus into no more than 128 tokens, and the maximum length of rewritten text is also set to 128 tokens. The sizes of the Wiki, MS MARCO, and Essay corpora are about 21M, 8.8M, and 14K, respectively. For dense retrieval, we build the index of the corpora using Faiss-gpu (Johnson et al., 2019).

Computational efficiency. We employ vLLM (Kwon et al., 2023) to efficiently (1) pre-compute the LLM-rewritten texts for our method and (2) generate adversarial LLM texts from OUTFOX (Koike et al., 2024b), achieving a throughput of approximately 2 seconds per generation. For online detection, we implement faiss-gpu (Johnson et al., 2019) for online retrieval, which takes <0.1 seconds for each input text.

Baselines. Entropy computes the average entropy of the logit distribution; **Likelihood** considers the average log probability of the input text; **Rank** and **LogRank** use the average rank and log-rank of the label tokens in the logit distribution. Based on these features, **LRR** (Su et al., 2023) combines the likelihood and log-rank for detection. **DNA-GPT** (Yang et al., 2023) truncates the input text into several passages and compares the n-gram divergence between the completions of passages. **DetectGPT** (Mitchell et al., 2023) and **NPR** (Su et al., 2023) compare the likelihood and log-rank of the input text logits and T5-large (Raffel et al., 2020) perturbed texts respectively. **Fast-DetectGPT** (Bao et al., 2023) boosts the perturbation process and creates massive perturbations for a more accurate comparison. **OUTFOX** (Koike et al., 2024b) introduces an adversarial method that uses an LLM-based detector and attacker to improve detection accuracy. We use Qwen2.5-14B-Instruct to implement the OUTFOX baseline because OUTFOX needs to generate high-quality adversarial samples through in-context learning. For the other LLM-based detection baseline methods, we use OPT-2.7B for a fair comparison.