Efficient Detection of LLM-generated Texts with a Bayesian Surrogate Model

Yibo Miao[1*], **Hongcheng Gao**[2*], **Hao Zhang**[3†], **Zhijie Deng**[1†]
[1]Qing Yuan Research Institute, SEIEE, Shanghai Jiao Tong University
[2]University of Chinese Academy of Sciences [3]University of California, San Diego
miaoyibo@sjtu.edu.cn, gaohongcheng23@mails.ucas.ac.cn
haozhang@ucsd.edu, zhijied@sjtu.edu.cn

Abstract

The detection of machine-generated text, especially from large language models (LLMs), is crucial in preventing serious social problems resulting from their misuse. Some methods train dedicated detectors on specific datasets but fall short in generalizing to unseen test data, while other zero-shot ones often yield suboptimal performance. Although the recent DetectGPT has shown promising detection performance, it suffers from significant inefficiency issues, as detecting a single candidate requires querying the source LLM with hundreds of its perturbations. This paper aims to bridge this gap. Concretely, we propose to incorporate a Bayesian surrogate model, which allows us to select typical samples based on Bayesian uncertainty and interpolate scores from typical samples to other samples, to improve query efficiency. Empirical results demonstrate that our method significantly outperforms existing approaches under a low query budget. Notably, when detecting the text generated by LLaMA family models, our method with just 2 or 3 queries can outperform DetectGPT with 200 queries.

1 Introduction

Large language models (LLMs) (Brown et al., 2020; Chowdhery et al., 2022; OpenAI, 2022; Touvron et al., 2023) have the impressive ability to replicate human language patterns, producing text that appears coherent, well-written, and persuasive, although the generated text may contain factual errors and unsupported quotations. As LLMs are increasingly used to simplify writing and presentation tasks, some individuals regrettably have misused LLMs for nefarious purposes, such as creating convincing fake news articles or engaging in cheating, which can have significant social consequences. Mitigating these negative impacts has become a pressing issue for the community.

The LLM-generated texts are highly articulate, posing a significant challenge for humans in identifying them (Gehrmann et al., 2019b). Fortunately, it is shown that machine learning tools can be leveraged to recognize the watermarks underlying the texts. Some methods (e.g., OpenAI, 2023b) involve training supervised classifiers, which, yet, suffer from overfitting to the training data and ineffectiveness to generalize to new test data. *Zero-shot* LLM-generated text detection approaches bypass these issues by leveraging the source LLM to detect its samples (Solaiman et al., 2019; Gehrmann et al., 2019b; Ippolito et al., 2020). They usually proceed by inspecting the average per-token log probability of the candidate text, but the practical detection performance can be unsatisfactory.

DetectGPT (Mitchell et al., 2023) is a recent method that achieves improved zero-shot detection efficacy by exploring the probability curvature of LLMs. It generates multiple perturbations of the candidate text and scores them using the source LLM to define detection statistics. It can detect texts generated by GPT-2 (Radford et al., 2019) and GPT-NeoX-20B (Black et al., 2022). Yet, DetectGPT relies on hundreds of queries to the source LLM to estimate the local probability curvature surrounding *one* single candidate passage. This level of computational expense is impractical for handling large LMs like LLaMA (Touvron et al., 2023), ChatGPT (OpenAI, 2022), and GPT-4 (OpenAI, 2023a).

This paper aims to improve the query efficiency of probability curvature-based detectors. We highlight that the inefficiency issues of the current approach stem from the use of purely random perturbations for curvature estimation. Intuitively, due to the constraint on locality, the perturbed samples are highly correlated, sharing most words and having similar semantics. As a result, characterizing the local probability curvature can be essentially optimized as (*i*) identifying a set of typical samples and evaluating the source LLM on them and (*ii*) interpolating the results to other samples.

A surrogate model that maps samples to LLM probability is necessary for interpolation, and it would be beneficial if the model could also identify typical samples. Given these, we opt for the Gaussian process (GP) model due to its non-parametric flexibility, resistance to overfitting in low-data regimes, and ease of use in solving regression problems. More importantly, the Bayesian uncertainty provided by GP can effectively indicate sample typicality, as demonstrated in active learning (Gal et al., 2017). Technically, we perform sample selection and GP fitting sequentially. At each step, we select the sample that the current GP model is most uncertain about, score it using the source LLM, and update the GP accordingly. Early stops can be in-

[*]Equal contribution.
[†]Corresponding authors.

Findings of the Association for Computational Linguistics ACL 2024, pages 6118–6130
August 11-16, 2024 ©2024 Association for Computational Linguistics

voked adaptively. After fitting, we utilize the GP, rather than the source LLM, to score a set of randomly perturbed samples to compute detection statistics. This way, we create a zero-shot LLM-generated text detector with significantly improved query efficiency.

To showcase the effectiveness and efficiency of our method, we conduct comprehensive empirical studies on diverse datasets using GPT-2 (Radford et al., 2019), LLaMA2 (Touvron et al., 2023), and Vicuna (Chiang et al., 2023). The results show that our method outperforms DetectGPT with substantial margins under the low query budget. When detecting texts generated by LLaMA, our method with just 2 or 3 queries can significantly outperform DetectGPT with 200 queries. This is one significant step toward *practical* use. When detecting texts generated by GPT-2, it can achieve similar performance with up to 2 times fewer queries than DetectGPT and achieve 3.7% higher AUROC at a query number of 5. We also show that our approach remains effective even when detecting texts generated by black-box models with huge parameter sizes like ChatGPT and conduct ablation studies to offer insights into the behavior of our method.

2 Related Works

Large language models. LLMs (Radford et al., 2019; Brown et al., 2020; Chowdhery et al., 2022; Zhang et al., 2022; OpenAI, 2022) have revolutionized the field of natural language processing by offering several advantages over previous pre-trained models (Devlin et al., 2018; Liu et al., 2019; Lan et al., 2019), including a better characterization of complex patterns and dependencies in the text, and the appealing in-context learning ability for solving downstream tasks with minimal examples. Representative models such as GPT-3 (Brown et al., 2020), PaLM (Chowdhery et al., 2022), and ChatGPT (OpenAI, 2022) have showcased their remarkable ability to generate text with high coherence, fluency, and semantic relevance. They can even effectively address complex inquiries related to science, mathematics, history, current events, and social trends. Therefore, it is increasingly important to effectively regulate the use of LLMs to prevent significant social issues.

LLM-generated text detection. Previous methods can be broadly categorized into two groups. The first group of methods performs detection in a zero-shot manner (Solaiman et al., 2019; Gehrmann et al., 2019a; Mitchell et al., 2023; Yang et al., 2023), but they require access to the source model that generates the texts to derive quantities like output logits or losses for detection. For instance, Solaiman et al. (2019) suggest that a higher log probability for each token indicates that the text will likely be machine-generated. When the output logits/losses of the source model are unavailable, these methods rely on a proxy model for detection. However, there is often a substantial gap between the proxy and source models from which the text is generated. Another group of methods trains DNN-based classifiers on collected human-written and machine-generated texts for detection (Guo et al., 2023; Uchendu et al., 2020; OpenAI, 2023b). However, such detectors are data-hungry and may exhibit poor generalization ability when facing domain shift (Bakhtin et al., 2019; Uchendu et al., 2020). Furthermore, training DNN-based classifiers is susceptible to backdoor attacks (Qi et al., 2021) and adversarial attacks (He et al., 2023). Besides, both He et al. (2023) and Li et al. (2023) develop benchmarks for evaluating existing detection methods and call for more robust detection methods.

3 Methodology

This section first reviews DetectGPT (Mitchell et al., 2023) and highlights its query inefficiency to justify the need for a surrogate model. We then emphasize the importance of being Bayesian and provide a comprehensive discussion on the specification of the surrogate model. We also discuss the strategy for selecting typical samples with Bayesian uncertainty. The method overview is in Fig. 1.

3.1 Preliminary

We consider the zero-shot LLM-generated text detection problem, which essentially involves binary classification to judge whether a text passage originates from a language model or not. Zero-shot detection implies we do not require a dataset composed of human-written and LLM-generated texts to train the detector. Instead, following common practice (Solaiman et al., 2019; Gehrmann et al., 2019b; Ippolito et al., 2020; Mitchell et al., 2023), we assume access to the source LLM to score the inputs, based on which the detection statistics are constructed.

DetectGPT (Mitchell et al., 2023) is a representative work in this line. It utilizes the following measure to determine if a text passage $\boldsymbol{x}$ is generated from a large language model $p_{\boldsymbol{\theta}}$:

$$\log p_{\boldsymbol{\theta}}(\boldsymbol{x}) - \mathbb{E}_{\tilde{\boldsymbol{x}}\sim q(\cdot|\boldsymbol{x})} \log p_{\boldsymbol{\theta}}(\tilde{\boldsymbol{x}}), \tag{1}$$

where $q(\cdot|\boldsymbol{x})$ is a perturbation distribution supported on the semantic neighborhood of the candidate text $\boldsymbol{x}$. For example, $q(\cdot|\boldsymbol{x})$ can be defined with manual rephrasings of $\boldsymbol{x}$ while maintaining semantic similarity. DetectGPT, in practice, instantiates $q(\cdot|\boldsymbol{x})$ with off-the-shelf pre-trained mask-filling models like T5 (Raffel et al., 2020) to avoid humans in the loop.

For tractablility, DetectGPT approximates Eq. (1) with Monte Carlo samples $\{\boldsymbol{x}_i\}_{i=1}^N$ from $q(\cdot|\boldsymbol{x})$:

$$\log p_{\boldsymbol{\theta}}(\boldsymbol{x}) - \frac{1}{N}\sum_{i=1}^{N} \log p_{\boldsymbol{\theta}}(\boldsymbol{x}_i) =: \ell(\boldsymbol{x}, p_{\boldsymbol{\theta}}, q). \tag{2}$$

Based on the hypothesis that LLM-generated texts should locate in the local maxima of the log probability of the source LLM, it is expected that $\ell(\boldsymbol{x}, p_{\boldsymbol{\theta}}, q)$ is large for LLM-generated texts but small for human-written ones, and thus a detector emerges. Despite good

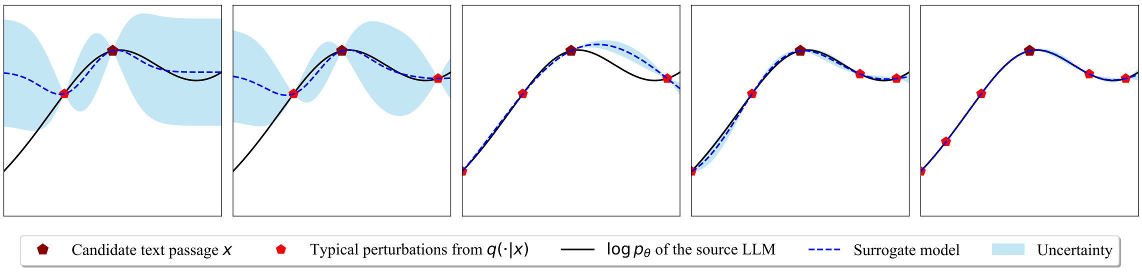


Figure 1: Method overview. Following DetectGPT (Mitchell et al., 2023), we explore the local structure of the probability curvature of the LLM $p_{\boldsymbol{\theta}}$ to determine whether a text passage $\boldsymbol{x}$ originates from it. However, instead of using the source LLM to score numerous random perturbations, we leverage the high redundancy among these perturbations to enhance query efficiency. We select a limited number of typical samples for scoring and interpolate their scores to other samples. To achieve reasonable selection and interpolation, we employ a Gaussian process as the surrogate model, which, as shown, enjoys non-parametric flexibility and delivers calibrated uncertainty in the presence of a suitable kernel. The figure above also demonstrates the sequential selection of typical samples—at each step, the sample that the surrogate model is most uncertain about is chosen. After fitting, we use the surrogate model as a substitute for $\log p_{\boldsymbol{\theta}}$ to calculate the detection measure $\ell(\boldsymbol{x}, p_{\boldsymbol{\theta}}, q)$ in Eq. (2).

performance, DetectGPT is costly because detecting one single candidate text hinges on $N+1$ (usually, $N \geq 100$) queries to the source model $p_{\boldsymbol{\theta}}$, which can lead to prohibitive overhead when applied to commercial LLMs.

3.2 Improve Query Efficiency with a Surrogate Model

Intuitively, we indeed require numerous perturbations to reliably estimate the structure of the local probability curvature around the candidate text $\boldsymbol{x}$ due to the high dimensionality of texts. However, given the mask-filling nature of the perturbation model and the requirement for semantic locality, the samples to be evaluated, referred to as $\mathbf{X} = \{\boldsymbol{x}_i\}_{i=0}^{N}$,[1] share a substantial amount of content and semantics. Such significant redundancy and high correlation motivate us to select only a small set of typical samples for scoring by the source LLM and then reasonably interpolate the scores to other samples (see Fig. 1). This way, we obtain the detection measure in a query-efficient manner.

A surrogate model that maps samples to LLM probability is required for interpolation, which should also help identify typical samples if possible. The learning of the model follows a standard regression setup. Let $\mathbf{X}_t = \{\boldsymbol{x}_{t_i}\}_{i=0}^{T} \subset \mathbf{X}$,[2] denote a subset of typical samples selected via some tactics and $\boldsymbol{y}_t = \{\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t_i})\}_{i=0}^{T}$ the corresponding log-probabilities yielded by the source LLM. The surrogate model $f : \mathcal{X} \to \mathbb{R}$ is expected to fit the mapping from $\mathbf{X}_t$ to $\boldsymbol{y}_t$, while being able to generalize reasonably to score the other samples in place of the source LLM.

3.3 The Bayesian Surrogate Model

Before discussing how to select the typical samples, it is necessary to clarify the specifications of the surrogate model. In our approach, we fit a dedicated surrogate model for each piece of text $\boldsymbol{x}$, and it approximates the source LLM only in the local region around $\boldsymbol{x}$. This allows us to avoid the frustrating difficulty of approximating the entire probability distribution represented by the source LLM, and work with lightweight surrogate models as well as good query efficiency.

The surrogate model f is expected to be trained in the low-data regime while being expressive enough to handle non-trivial local curvature and not prone to overfitting. Additionally, the model should inherently incorporate mechanisms for typical sample selection. Given these requirements, we chose to use a Gaussian process (GP) model as the surrogate model due to its non-parametric flexibility, resistance to overfitting, and capability to quantify uncertainty (Williams and Rasmussen, 1995). Parametric models such as neural networks cannot meet all of these requirements simultaneously.

Concretely, consider a GP prior in function space, $f(\boldsymbol{x}) \sim \mathcal{GP}(0, k(\boldsymbol{x}, \boldsymbol{x}'))$, where the mean function is set to zero following common practice and $k(\boldsymbol{x}, \boldsymbol{x}')$ refers to the kernel function. Consider a Gaussian likelihood with noise variance σ^2 for this problem, i.e., $y(\boldsymbol{x})|f(\boldsymbol{x}) \sim \mathcal{N}(y(\boldsymbol{x}); f(\boldsymbol{x}), \sigma^2)$.

Posterior predictive. It is straightforward to write down the posterior distribution of the function values $\boldsymbol{f}_{\mathbf{X}^*}$ for unseen samples $\mathbf{X}^* = \{\boldsymbol{x}_i^*\}_{i=1}^{M}$, detailed below

$$p(\boldsymbol{f}_{\mathbf{X}^*}|\mathbf{X}_t, \boldsymbol{y}_t, \mathbf{X}^*) = \mathcal{N}(\bar{\boldsymbol{f}}^*, \boldsymbol{\Sigma}^*) \tag{3}$$

where

$$\begin{aligned}\bar{\boldsymbol{f}}^* &:= \boldsymbol{k}_{\mathbf{X}^*,\mathbf{X}_t}[\boldsymbol{k}_{\mathbf{X}_t,\mathbf{X}_t} + \sigma^2\mathbf{I}]^{-1}\boldsymbol{y}_t \\ \boldsymbol{\Sigma}^* &:= \boldsymbol{k}_{\mathbf{X}^*,\mathbf{X}^*} - \boldsymbol{k}_{\mathbf{X}^*,\mathbf{X}_t}[\boldsymbol{k}_{\mathbf{X}_t,\mathbf{X}_t} + \sigma^2\mathbf{I}]^{-1}\boldsymbol{k}_{\mathbf{X}_t,\mathbf{X}^*}.\end{aligned} \tag{4}$$

$\boldsymbol{k}_{\mathbf{X}^*,\mathbf{X}_t} \in \mathbb{R}^{M\times(T+1)}$, $\boldsymbol{k}_{\mathbf{X}_t,\mathbf{X}_t} \in \mathbb{R}^{(T+1)\times(T+1)}$, $\boldsymbol{k}_{\mathbf{X}^*,\mathbf{X}^*} \in \mathbb{R}^{M\times M}$ are evaluations of the kernel k. With this, we can analytically interpolate scores from the typical samples to new test samples.

Text kernel. It should be noted that the GP model described above is designed to operate within the do-

[1]If not misleading, $\boldsymbol{x}_0$ denotes the original candidate $\boldsymbol{x}$.

[2]We constrain $t_0 = 0$ to include the original text $\boldsymbol{x}_0$ in $\mathbf{X}_t$.

main of natural language, meaning traditional kernels like RBF and polynomial kernels are not suitable. To address this challenge, we draw inspiration from BertScore (Zhang et al., 2019), which has demonstrated a good ability to capture similarities between text passages. We make a straightforward modification to BertScore, resulting in the following kernel:

$$k(\boldsymbol{x}, \boldsymbol{x}') := \alpha \cdot \mathrm{BertScore}(\boldsymbol{x}, \boldsymbol{x}') + \beta \quad (5)$$

where $\alpha \in \mathbb{R}^+$ and $\beta \in \mathbb{R}$ are two hyperparameters to boost flexibility. Other symmetric positive semi-definite kernels defined on texts are also applicable here.

Hyperparameter tuning. To make the hyperparameters α, β, and σ^2 suitable for the data at hand, we optimize them to maximize the log marginal likelihood of the targets $\boldsymbol{y}_t$, a canonical objective for hyperparameter tuning for Bayesian methods:

$$\log p(\boldsymbol{y}_t | \mathbf{X}_t, \alpha, \beta, \sigma^2) \propto -[\boldsymbol{y}_t^\top (\boldsymbol{k}_{\mathbf{X}_t, \mathbf{X}_t} + \sigma^2 \mathbf{I})^{-1} \boldsymbol{y}_t + \log |\boldsymbol{k}_{\mathbf{X}_t, \mathbf{X}_t} + \sigma^2 \mathbf{I}|], \quad (6)$$

where α and β exist in the computation of $\boldsymbol{k}_{\mathbf{X}_t, \mathbf{X}_t}$. We can make use of AutoDiff libraries to perform gradient-based optimization of the hyperparameters directly. Since the number of samples in $\mathbf{X}_t$ is typically less than 100, the computational resources required for calculating matrix inversion and log-determinant are negligible.

3.4 Sequential Selection of Typical Samples

As discussed above, once we obtain the set of typical samples, we can effortlessly score new samples using the GP model. We next describe how to use Bayesian uncertainty to identify the typical samples.

In our case, the typicality of a text sample depends on the surrogate model. If the surrogate model can accurately predict the score for the sample, it should not be considered typical, and vice versa. However, relying on the ground-truth score to measure typicality is query-intensive, and hence impractical.

Fortunately, in Bayesian methods, there is often a correlation between the model's prediction accuracy and uncertainty, with higher uncertainty implying lower accuracy (Lakshminarayanan et al., 2017; Maddox et al., 2019; Deng and Zhu, 2023). This allows us to leverage the Bayesian uncertainty of the employed GP model to select typical samples sequentially. We should choose samples with high predictive uncertainty, i.e., samples the model is likely to predict inaccurately. Interestingly, such an uncertainty-based selection strategy has similarities with those employed in existing active learning approaches (Gal et al., 2017; Mohamadi and Amindavar, 2020), highlighting the soundness of our approach. Our approach also resembles a Bayesian optimization program that finds points maximizing an acquisition function (Snoek et al., 2012). In our case, the acquisition contains only the uncertainty term.

Specifically, we perform data selection and model fitting alternately, with the following details.

Algorithm 1 Efficient detection of LLM-generated texts with a Bayesian surrogate model.

Input: Text passage $\boldsymbol{x}$, LLM $p_{\boldsymbol{\theta}}$, perturbation model $q(\cdot|\boldsymbol{x})$, kernel $k(x, x')$, hyperparameters α, β, σ^2, sample sizes N, T, S, detection threshold δ.
Output: True/false indicating whether the text passage $\boldsymbol{x}$ comes from the LLM $p_{\boldsymbol{\theta}}$ or not.
Perform rephrasing with $q(\cdot|\boldsymbol{x})$, obtaining perturbations $\mathbf{X} = \{\boldsymbol{x}_i\}_{i=0}^N$;
Randomly initialize the typical set $\mathbf{X}_t$ and the set $\mathbf{X}^*$ for selection;
$\boldsymbol{y}_t \leftarrow \log p_{\boldsymbol{\theta}}(\mathbf{X}_t)$;
while $|\mathbf{X}_t| < T$ *or* other early stop criteria have not been satisfied **do**
    Optimize the hyperparameters α, β, and σ^2 according to Eq. (6) given $\mathbf{X}_t$ and $\boldsymbol{y}_t$;
    Estimate the predictive covariance $\boldsymbol{\Sigma}^*$ for $\mathbf{X}^*$ detailed in Eq. (4);
    Identify the sample in $\mathbf{X}^*$ with the largest uncertainty (i.e., the diagonal element of $\boldsymbol{\Sigma}^*$);
    Score this sample with the LLM;
    Append the sample and the target to $\mathbf{X}_t$ and $\boldsymbol{y}_t$ respectively;
Approximately estimate the detection measure $\ell(\boldsymbol{x}, p_{\boldsymbol{\theta}}, q)$ with the resulting GP model;
Return True *if* $\ell(\boldsymbol{x}, p_{\boldsymbol{\theta}}, q) > \delta$ *else* False;

Initlization. We initialize $\mathbf{X}_t$ with a random subset of $\mathbf{X}$, i.e., $\mathbf{X}_t = \{\boldsymbol{x}_{t_i}\}_{i=0}^S, 1 \leq S < T$. Unless otherwise specified, we set $S = 2$, where the first sample refers to the original candidate text and the second a random perturbation of it. We use $\boldsymbol{y}_t$ to denote the corresponding ground-truth scores.

Model fitting. Optimize the hyperparameters of the GP model on data $(\mathbf{X}_t, \boldsymbol{y}_t)$ to maximize the log marginal likelihood defined in Eq. (6).

Data selection. Denote by $\mathbf{X}^*$ the samples for selection. It can be the complement of $\mathbf{X}_t$ in $\mathbf{X}$ or other random perturbations around the candidate $\boldsymbol{x}$. Compute the covariance matrix $\boldsymbol{\Sigma}^*$ defined in Eq. (4), whose diagonal elements correspond to the predictive uncertainty of samples in $\mathbf{X}^*$. Augment the sample with the largest uncertainty to $\mathbf{X}_t$, and append its score yielded by the source LLM to $\boldsymbol{y}_t$.

Adaptive exit. If the size of $\mathbf{X}_t$ equals T or a specific stop criterion is satisfied, e.g., the largest uncertainty is lower than a threshold, the program exits from the model fitting and data selection loop.

Estimation of detection measure. We use the resulting GP model to compute the approximate log probability for all samples in $\mathbf{X}$, and hence get an estimation of the detection measure $\ell(\boldsymbol{x}, p_{\boldsymbol{\theta}}, q)$.

We depict the overall algorithmic in Algorithm 1. We clarify our method also applies to situations where only a proxy of the source LLM is available, as demonstrated in Section 4.3. Despite decreasing the number of queries to the source LLM, our method needs to estimate the BertScore in the local environment frequently.

Dataset	Method	2	3	4	10	20	50	100	200
Xsum	DetectGPT (7B)	0.550	0.588	0.594	0.614	0.644	0.674	0.699	0.707
	Our Method (7B)	0.723	0.864	0.865	-	-	-	-	-
	DetectGPT (13B)	0.526	0.505	0.536	0.524	0.590	0.556	0.586	0.580
	Our Method (13B)	0.721	0.851	0.821	-	-	-	-	-
Squad	DetectGPT (7B)	0.463	0.485	0.485	0.530	0.533	0.533	0.528	0.528
	Our Method (7B)	0.718	0.806	0.799	-	-	-	-	-
	DetectGPT (13B)	0.482	0.444	0.463	0.426	0.432	0.461	0.454	0.455
	Our Method (13B)	0.707	0.770	0.757	-	-	-	-	-
Writing	DetectGPT (7B)	0.424	0.479	0.506	0.466	0.456	0.455	0.452	0.462
	Our Method (7B)	0.604	0.891	0.903	-	-	-	-	-
	DetectGPT (13B)	0.480	0.431	0.473	0.421	0.420	0.422	0.426	0.427
	Our Method (13B)	0.619	0.893	0.900	-	-	-	-	-

Table 1: The AUROC for detecting samples generated by LLaMA2-7B/13B varies depending on the number of queries made to the source model. We primarily focus on scenarios with low query budgets. Additionally, since our method at a query budget of 4 has already outperformed DetectGPT, we don't report the results of our method at larger query budgets.

Dataset	Method	2	3	4	10	20	50	100	200
Xsum	DetectGPT (7B)	0.850	0.893	0.904	0.927	0.932	0.95	0.952	0.952
	Our Method (7B)	0.958	0.955	0.957	-	-	-	-	-
	DetectGPT (13B)	0.817	0.849	0.861	0.913	0.922	0.936	0.932	0.935
	Our Method (13B)	0.886	0.912	0.929	-	-	-	-	-
Squad	DetectGPT (7B)	0.798	0.820	0.857	0.871	0.878	0.889	0.886	0.884
	Our Method (7B)	0.947	0.936	0.932	-	-	-	-	-
	DetectGPT (13B)	0.671	0.686	0.712	0.740	0.731	0.743	0.758	0.757
	Our Method (13B)	0.799	0.785	0.787	-	-	-	-	-
Writing	DetectGPT (7B)	0.903	0.916	0.930	0.938	0.937	0.936	0.941	0.938
	Our Method (7B)	0.985	0.983	0.979	-	-	-	-	-
	DetectGPT (13B)	0.856	0.897	0.933	0.962	0.963	0.963	0.966	0.967
	Our Method (13B)	0.929	0.971	0.979	-	-	-	-	-

Table 2: The AUROC for detecting samples generated by Vicuna-7B/13B varies depending on the number of queries made to the source model.

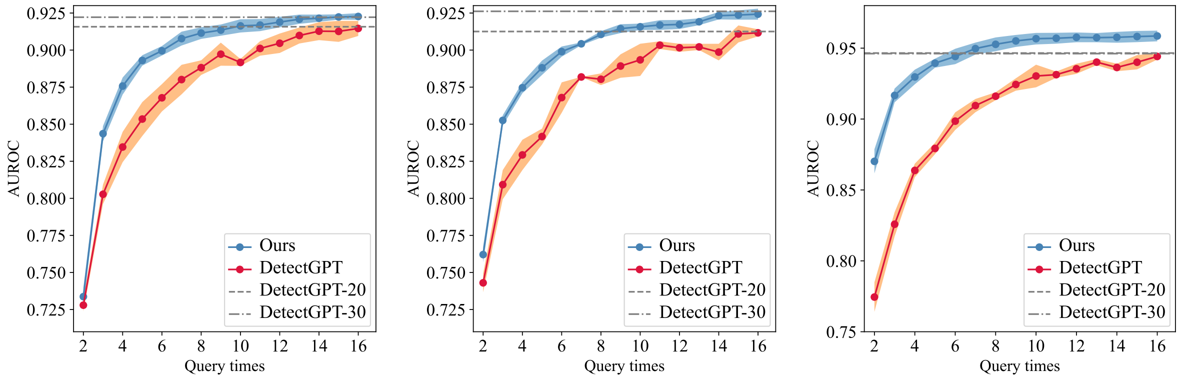
4 Experiments

We conduct extensive experiments to evaluate the efficiency and efficacy of our method for zero-shot LLM-generated text detection. We mainly compare our method to DetectGPT (Mitchell et al., 2023) because (*i*) both works adopt the same detection measure, and (*ii*) DetectGPT has proven to defeat prior zero-shot and supervised methods consistently. We are primarily concerned with detecting under a low query budget and admit that our method would perform similarly to DetectGPT if queries to the source model can be numerous. We also consider a black-box variant of the task where only a proxy of the source LLM is available for detection. We have also verified the effectiveness of our method in detecting large-parameter models, such as ChatGPT. We further qualitatively analyze the behavioral difference between our method and DetectGPT and showcase the limitations of our method.

Datasets. Following DetectGPT (Mitchell et al., 2023), we primarily experiment on three datasets, covering news articles sourced from the XSum dataset (Narayan et al., 2018), which represents the problem of identifying fake news, paragraphs from Wikipedia drawn from the SQuAD contexts (Rajpurkar et al., 2016), which simulates the detection of machine-generated academic essays and prompted stories from Reddit WritingPrompts dataset (Fan et al., 2018), which indicates the recognition of LLM-created creative writing submissions. These datasets are representative of a variety of common domains and use cases for LLM. We let the LLMs expand the first 30 tokens of the real text to construct generations. Refer to (Mitchell et al., 2023) for more details.

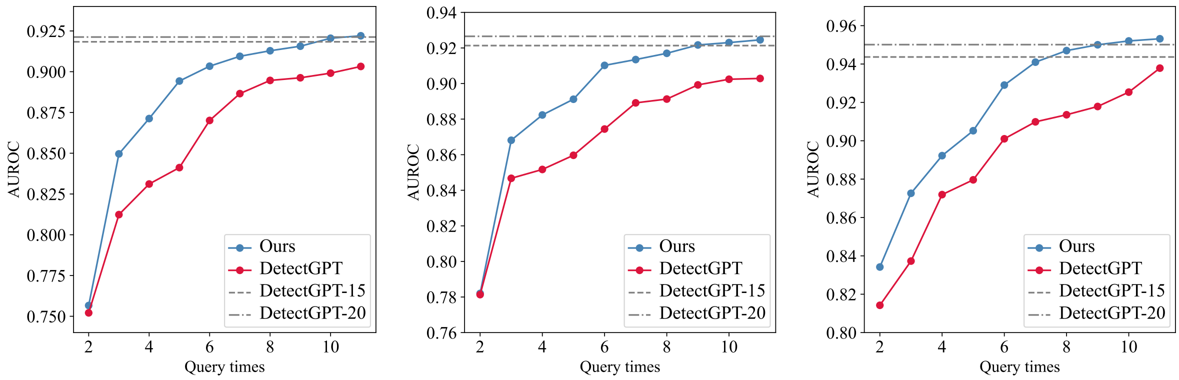
Evaluation Metric. The detection of LLM-generated texts is actually a binary classification problem. Thereby, we use the area under the receiver operating characteristic curve (AUROC) as the key metric to evaluate the performance of detectors (i.e., classifiers).

The LLMs of concern. We focus on several widely used open-source LLMs: GPT-2 (Radford et al., 2019),



(a) XSum (b) SQuAD (c) WritingPrompts

Figure 2: The AUROC for detecting samples generated by GPT-2 varies depending on the number of queries made to the source GPT-2. We present the results on three representative datasets.



(a) XSum (b) SQuAD (c) WritingPrompts

Figure 3: The AUROC for detecting samples generated by GPT-2 varies depending on the number of queries made to the source GPT-2. We use T5-3B as the perturbation model here.

LLaMA2 (Touvron et al., 2023) and Vicuna (Chiang et al., 2023). GPT-2 is an LLM that leverages the strengths of the GPT architecture. LLaMA2 and Vicuna have gained great attention for their ability to build chatbots. Evaluating our method on them can demonstrate the practical value and effectiveness of our method.

Hyperparameters. Most hyperparameters regarding the perturbation model, i.e., $q(\cdot|\boldsymbol{x})$, follow DetectGPT (Mitchell et al., 2023). We use T5-large when detecting samples from GPT-2, LLaMa2, and Vicuna. Unless otherwise specified, we set the sample size N for estimating the detection measure to 200 and S to 2. We tune the hyperparameters associated with the GP model with an Adam optimizer (Kingma and Ba, 2014) using a learning rate of 0.01 (cosine decay is deployed) for 50 iterations.

4.1 Detection of LLaMA and Vicuna

We first compare our method to DetectGPT in detecting contents generated by LLaMA family (Touvron et al., 2023) and focus on the commonly used LLaMA2 and Vicuna models, LLaMA2-7B/13B and Vicuna-7B/13B, to thoroughly investigate the practical application value of our method. LLaMA2 and Vicuna models are trained on a diverse range of web text and conversational data and have gained recent attention. The texts generated by these models are of exceptional quality and coherence, closely resembling texts written by humans.

Letting Q denote the query budget, DetectGPT uses $Q-1$ random perturbations to estimate the detection measure $\ell(\boldsymbol{x}, p_{\boldsymbol{\theta}}, q)$. In contrast, our method uses $Q-1$ typical samples for fitting the surrogate model and still uses a large number of random perturbations (as stated, 200) to estimate ℓ, which is arguably more reliable.

Empirically, we use a normalized version of measurement defined in Eq. (1) in this experiment, which is detailed in Appendix C. We use T5-large as the perturbation model in this case. To emphasize the query-saving effects of our method, we consider the range of query times only up to 4. Nevertheless, we still include DetectGPT under $10, 20, 50, 100$, and 200 query budgets. We display the comparison between DetectGPT and our method in Tables 1 and 2, where the three datasets and various query budgets are also considered.

As shown, our method with only 4 query times outperforms DetectGPT with 200 query times, whether detecting texts generated by LLaMA2 or Vicuna. Notably, when detecting samples generated by LLaMA2, our method with only 2 query budgets can beat DetectGPT with 200 query budgets. These results demonstrate that using typical samples rather than random samples can significantly improve efficiency.

4.2 Detection of GPT-2

We then compare our method to DetectGPT in detecting contents generated by GPT-2. In Fig. 2, we present a comparison of detection AUROC on the datasets mentioned earlier. To reduce the influence of randomness, we report average results and variances over three random runs. For a comprehensive comparison, we also draw the performance of DetectGPT using 20 and 30 queries in the figure. As shown, our method outperforms DetectGPT significantly, achieving faster performance gains, particularly under a lower query budget. Notably, our method using only 10 queries can outperform DetectGPT using 20 queries in all three cases.

Interestingly, our method can already surpass DetectGPT in the 2-query case, especially on the WritingPrompts dataset. In that case, our method fits a GP model with only the original text passage as well as a random perturbation. This result confirms that the GP-based surrogate model is highly data-efficient and excels at interpolating the scores of typical samples to unseen data. Furthermore, DetectGPT's performance gain with increasing query times is slow on the WritingPrompts dataset, as its detection AUROCs using 16, 20, and 30 queries are similar. In contrast, our method does not face this issue. Given that the variances are minor compared to the mean values, we omit repeated random runs in the following analysis.

Impact of the perturbation model. The perturbation model used in the above studies is the T5-large model. We then question whether replacing it with a more powerful model results in higher detection performance. For an answer, we introduce the T5-3B model as the perturbation model and conduct a similar set of experiments, with the results displayed in Fig. 3.

As shown, the performance of DetectGPT and our method is substantially improved compared to the results in Fig. 2, indicating the higher ability of T5-3B to perturb in the semantic space. Still, our method consistently surpasses DetectGPT and achieves similar performance with up to $2\times$ fewer queries than DetectGPT. In particular, the average AUROCs of our method at query times of 5 and 10 are 0.897 and 0.932, respectively, while those for DetectGPT are 0.860 and 0.909.

High query budget. To demonstrate that our method remains effective with more query times, we test our method on 50 queries for case study, and it achieved a final AUROC of 98.0% on WritingPrompts, the same as DetectGPT using about 150 queries. This result verifies the much higher efficiency of our method than DetectGPT while being able to achieve excellent performance.

4.3 Cross Evaluation

The above studies assume a white-box setting, where the source LLM is utilized to detect its generations. Yet, in practice, we may not know which model the candidate passage was generated from. A remediation is to introduce a proxy model for an approximate estimate of the log probability of source LLM (Mitchell et al., 2023). This section examines the impact of doing this on the final detection performance. To reduce costs, we consider using GPT-J (Wang and Komatsuzaki, 2021), GPT-Neo-2.7 (Black et al., 2021), and GPT-2 as the source and proxy models and evaluate the detection AUROC across all 9 source-proxy combinations. We present the average performance across 200 samples from XSum, SQuAD, and WritingPrompts in Fig. 4.

As demonstrated, using the same model for both generating and scoring yields the highest detection performance, but employing a scoring model different from the source model can still be advantageous. The column mean represents the quality of a scoring model, and our results suggest that GPT-Neo-2.7 is more effective in accounting for scoring. Furthermore, we observe significant improvements (up to 3% AUROC) in our method's results over DetectGPT, indicating the generalizability of our approach.

Method	Xsum	Squad	Writing
DetectGPT (LLaMA2)	0.397	0.473	0.641
Our Method (LLaMA2)	0.631	0.660	0.742
DetectGPT (Vicuna)	0.595	0.606	0.733
Our Method (Vicuna)	0.780	0.714	0.850

Table 3: The AUROC for detecting texts generated by ChatGPT with query budget 15.

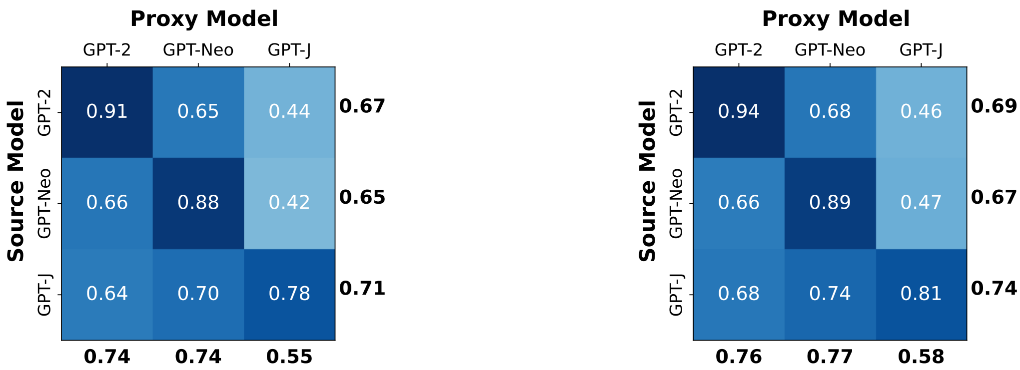
4.4 Detection of ChatGPT

In addition to the GPT-2, LLaMA2 and Vicuna models discussed above, is crucial to determine how to detect the text generated by the most frequently used ChatGPT (OpenAI, 2022), which has a huge number of parameters. However, ChatGPT is a black box model and we can't directly get the log probability from ChatGPT. We use the same method introduced in Section 4.3 — obtaining the log probability of the text generated by ChatGPT through a proxy model. We test using LLaMA2-13B and Vicuna-13B as the proxy models to estimate the log probability of ChatGPT. The results are shown in Table 3. Our method significantly outperforms DetectGPT on all three datasets. It is shown that by including a proxy model, our method can efficiently detect the text from LMs that have huge parameter sizes.

4.5 More Studies

To better understand our method, we lay out the following additional studies.

How does our method behave differently from DetectGPT? We are interested in how our method achieves better performance than DetectGPT. To chase an intuitive answer, we collect some human-written texts from the considered three datasets as well as the generated texts from GPT-2, and compute the detection measure $\ell(\boldsymbol{x}, p_{\boldsymbol{\theta}}, q)$ estimated by DetectGPT and our method under a query budget of 15 and the T5-large perturbation model. We list the results in the Appendix due to space



(a) DetectGPT (b) Our Method

Figure 4: Cross evaluation of using various source models (i.e., those generating the texts) and proxy models (i.e., those scoring the texts for detection) in detection. We select models from {GPT-J, GPT-Neo-2.7, GPT-2}. We report the average AUROC over the three datasets. We offer the row/column mean. The query budget is 15.

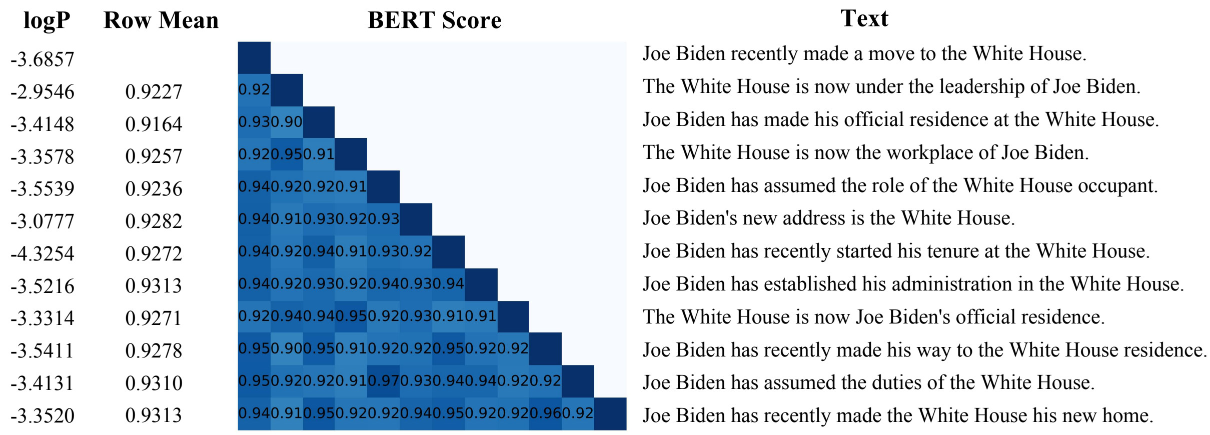


Figure 5: The visualization of the candidate text passage and the first 11 typical perturbations of it identified by our method, ordered from top to bottom. The BertScore among them and the row mean (estimated without the diagonal elements) are reported. The log probabilities are given by GPT-2.

constraints. We find that (1) Our method's estimation of ℓ is usually higher than DetectGPT. Recalling the expression of ℓ, we conclude our method can usually select samples with substantially lower $\log p_\theta$ than random samples. (2) Our method can occasionally produce too high or too low ℓ for texts written by humans. This could be attributed to the fact that using a limited number of typical samples may fail to reliably capture the local curvature when it is ill-posed or complex.

The visualization of typical samples. As depicted in Fig. 1, typical samples have a direct physical meaning in toy cases. However, it is unclear whether this property can be maintained in the high-dimensional space in which texts exist. To investigate this, we conducted a simple study based on the text passage, "Joe Biden recently made a move to the White House." Specifically, we perturb it with the rewriting function of ChatGPT for 50 times, and simulate the sequential procedure of typical sample selection. We present the original text and the first 11 typical samples in Fig. 5. We also display the BertScore among them as well as the log probabilities $\log p_\theta$ of GPT-2 for them. As shown, the row mean exhibits a clear increasing trend as the index of the typical sample increases, indicating that the later selected samples are becoming more similar to the earlier ones. In other words, the uniqueness or typicality of the selected samples decreases over the course of the selection process. This phenomenon is consistent with our expectations for the selected samples and confirms the effectiveness of our uncertainty-based selection method.

5 Conclusion

This paper tackles the issue that the existing probability curvature-based method for detecting LLM-generated text relies on a vast number of queries to the source LLM when detecting a candidate text passage. We introduce a Bayesian surrogate model to identify typical samples and then interpolate their scores to others. We use a Gaussian process regressor to instantiate the surrogate model and perform an online selection of typical samples based on Bayesian uncertainty. Extensive empirical studies on various datasets, using GPT-2, LLaMA2, and Vicuna, validate our method's superior effectiveness and efficiency over DetectGPT.

Our optimization approach serves as a foundational method, which can be applied to other unsupervised detection techniques such as Binoculars (Hans et al., 2024) and GPT-who (Venkatraman et al., 2023), offering a means to enhance overall detection performance.

Acknowledgement

This work was supported by NSF of China (No. 62306176), Key R&D Program of Shandong Province, China (2023CXGC010112), Natural Science Foundation of Shanghai (No. 23ZR1428700), and CCF-Baichuan-Ebtech Foundation Model Fund.

Limitations

A limitation is that our method is not compatible with parallel computing due to the sequential nature of sample selection. Besides, with the goal of enhancing DetectGPT's query efficiency, we have not evaluated its reliability against paraphrasing attacks.

Ethics Statements

We use public datasets and open-source LLMs for research purposes only, under licenses.

We emphasize that detecting LLM-generated text is crucial in preventing serious social problems that may arise from the misuse of LLMs. However, our method may not achieve 100% detection of all samples generated by LLMs. Samples that go undetected could potentially introduce some impact or consequences. E.g., LLM-generated text could be used to spread false information, manipulate public opinion, or even incite violence.

References

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

Sid Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, et al. 2022. Gpt-neox-20b: An open-source autoregressive language model. *arXiv preprint arXiv:2204.06745*.

Sid Black, Leo Gao, Phil Wang, Connor Leahy, and Stella Biderman. 2021. GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh-Tensorflow. If you use this software, please cite it using these metadata.

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.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

Zhijie Deng and Jun Zhu. 2023. Bayesadapter: Being bayesian, inexpensively and reliably, via bayesian fine-tuning. In *Asian Conference on Machine Learning*, pages 280–295. PMLR.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. *arXiv preprint arXiv:1805.04833*.

Yarin Gal, Riashat Islam, and Zoubin Ghahramani. 2017. Deep bayesian active learning with image data. In *International conference on machine learning*, pages 1183–1192. PMLR.

Sebastian Gehrmann, Hendrik Strobelt, and Alexander Rush. 2019a. 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.

Sebastian Gehrmann, Hendrik Strobelt, and Alexander M Rush. 2019b. 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.

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.

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. *arXiv preprint arXiv:2401.12070*.

Xinlei He, Xinyue Shen, Zeyuan Chen, Michael Backes, and Yang Zhang. 2023. MGTBench: Benchmarking Machine-Generated Text Detection. *CoRR abs/2303.14822*.

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

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. 2017. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Yafu Li, Qintong Li, Leyang Cui, Wei Bi, Longyue Wang, Linyi Yang, Shuming Shi, and Yue Zhang. 2023. Deepfake text detection in the wild. *arXiv preprint arXiv:2305.13242*.

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*.

Wesley J Maddox, Pavel Izmailov, Timur Garipov, Dmitry P Vetrov, and Andrew Gordon Wilson. 2019. A simple baseline for bayesian uncertainty in deep learning. *Advances in neural information processing systems*, 32.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D. Manning, and Chelsea Finn. 2023. Detectgpt: Zero-shot machine-generated text detection using probability curvature.

Salman Mohamadi and Hamidreza Amindavar. 2020. Deep bayesian active learning, a brief survey on recent advances. *arXiv preprint arXiv:2012.08044*.

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. *arXiv preprint arXiv:1808.08745*.

OpenAI. 2022. Introducing chatgpt. `https://openai.com/blog/chatgpt`, Last accessed on 2023-05-09.

OpenAI. 2023a. Gpt-4. `https://openai.com/research/gpt-4`, Last accessed on 2023-05-09.

OpenAI. 2023b. New ai classifier for indicating ai-written text. Https://openai.com/blog/new-ai-classifier-for-indicating-ai-written-text.

Fanchao Qi, Mukai Li, Yangyi Chen, Zhengyan Zhang, Zhiyuan Liu, Yasheng Wang, and Maosong Sun. 2021. Hidden killer: Invisible textual backdoor attacks with syntactic trigger. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 443–453, Online. Association for Computational Linguistics.

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

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

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

Jasper Snoek, Hugo Larochelle, and Ryan P Adams. 2012. Practical bayesian optimization of machine learning algorithms. *Advances in neural information processing systems*, 25.

Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, Gretchen Krueger, Jong Wook Kim, Sarah Kreps, et al. 2019. Release strategies and the social impacts of language models. *arXiv preprint arXiv:1908.09203*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

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

Saranya Venkatraman, Adaku Uchendu, and Dongwon Lee. 2023. Gpt-who: An information density-based machine-generated text detector. *arXiv preprint arXiv:2310.06202*.

Ben Wang and Aran Komatsuzaki. 2021. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. `https://github.com/kingoflolz/mesh-transformer-jax`.

Christopher Williams and Carl Rasmussen. 1995. Gaussian processes for regression. *Advances in neural information processing systems*, 8.

Xianjun Yang, Wei Cheng, Linda Petzold, William Yang Wang, and Haifeng Chen. 2023. Dna-gpt: Divergent n-gram analysis for training-free detection of gpt-generated text. *arXiv preprint arXiv:2305.17359*.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

A ROC Curves

To better demonstrate the effectiveness of our approach, we analyzed the True Positive Rate (TPR) at various False Positive Rates (FPRs) following (Sadasivan et al., 2023). Fig 6 shows the ROC curves for detecting samples generated by GPT-2 when the query budget is 15 with T5-3B as the perturbation model. Our detector can achieve 83.6% TPR on XSum, 87.0% on SQuAD, and 92.8% TPR on WritingPrompts at 15% FPR. We also have a comparison to DetectGPT of TPR at low FPRs in Table 4. The results clearly demonstrate that our method consistently outperforms DetectGPT across various scenarios.

We also provide the ROC curves for detecting samples generated by Vicuna-7B/13B when the query budget is 10 with T5-large as the perturbation model, and the comparisons between our method and DetectGPT of TPR at low FPRs are shown in Tables 5 and 6, respectively.

B The Behavioral Difference between DetectGPT and Our Method

We collect some human-written texts and generated texts from GPT-2, meanwhile computing the detection measure $\ell(\boldsymbol{x}, p_{\boldsymbol{\theta}}, q)$ estimated by DetectGPT and our method under a query budget of 15 and the T5-large perturbation model. We present the results in Table 7.

C The Normalized Version of the Difference between Log Probabilities

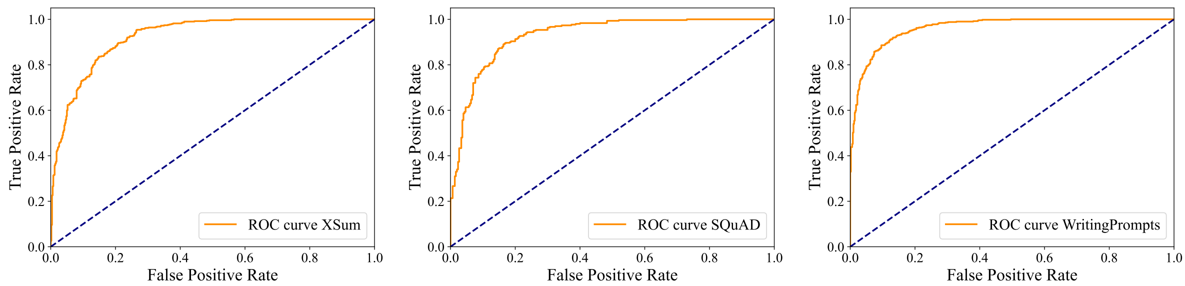
For large-parameter models like LLaMA-7B/13B models, the log probability difference between texts with minimal differences can be significant. Therefore, to obtain a robust evaluation of the difference between log probabilities, we normalized the measurement defined in Eq. (1) to:

$$\frac{\log p_{\boldsymbol{\theta}}(\boldsymbol{x}) - \mathbb{E}_{\tilde{\boldsymbol{x}} \sim q(\cdot|\boldsymbol{x})} \log p_{\boldsymbol{\theta}}(\tilde{\boldsymbol{x}})}{\sigma}, \tag{7}$$

where σ is the standard deviation of $\log p_{\boldsymbol{\theta}}(\tilde{\boldsymbol{x}})$.

Method	Xsum		Squad		Writing	
	FPR@0.01	FPR@0.05	FPR@0.01	FPR@0.05	FPR@0.01	FPR@0.05
DetectGPT	0.278	0.514	0.230	0.487	0.420	0.678
Our Method	0.314	0.574	0.267	0.613	0.502	0.784

Table 4: Comparison on TPR at various FPRs for detecting samples generated by GPT-2 with query budget 15.

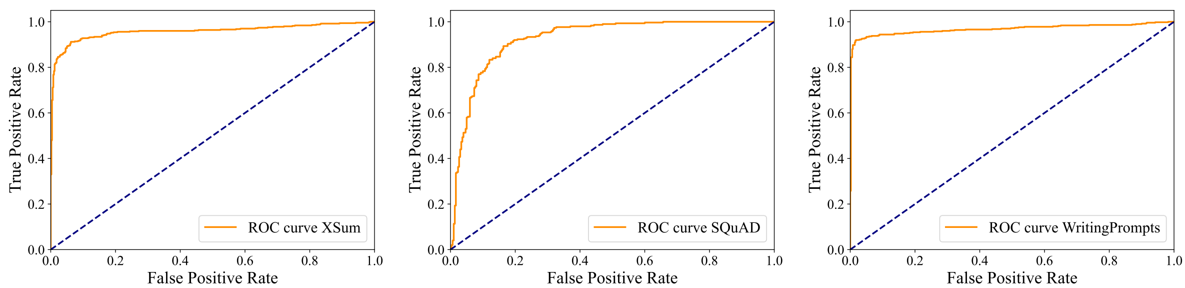


(a) XSum (b) SQuAD (c) WritingPrompts

Figure 6: The ROC curves for detecting samples generated by GPT-2 when query budget is 15. We present the results on three representative datasets. We use T5-3B as the perturbation model.

Method	Xsum		Squad		Writing	
	FPR@0.01	FPR@0.05	FPR@0.01	FPR@0.05	FPR@0.01	FPR@0.05
DetectGPT	0.604	0.798	0.027	0.433	0.780	0.878
Our Method	0.798	0.884	0.113	0.530	0.898	0.930

Table 5: Comparison on TPR at various FPRs for detecting samples generated by Vicuna-7B with query budget 10.

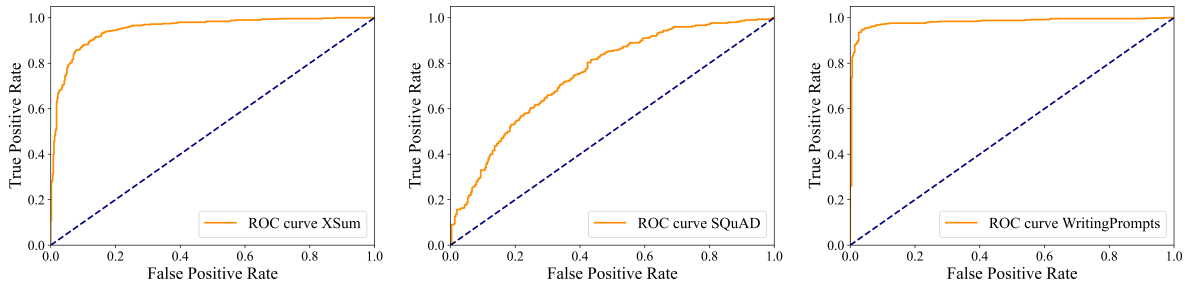


(a) XSum (b) SQuAD (c) WritingPrompts

Figure 7: The ROC curves for detecting samples generated by Vicuna-7B when query budget is 10. We present the results on three representative datasets. We use T5-large as the perturbation model.

Method	Xsum		Squad		Writing	
	FPR@0.01	FPR@0.05	FPR@0.01	FPR@0.05	FPR@0.01	FPR@0.05
DetectGPT	0.372	0.562	0.007	0.037	0.568	0.854
Our Method	0.400	0.754	0.090	0.177	0.834	0.954

Table 6: Comparison on TPR at various FPRs for detecting samples generated by Vicuan-13B with query budget 10.



(a) XSum (b) SQuAD (c) WritingPrompts

Figure 8: The ROC curves for detecting samples generated by Vicuna-13B when query budget is 10. We present the results on three representative datasets. We use T5-large as the perturbation model.

Text	DetectGPT	Ours
(Human-written)... Others, however, thought that "the most striking aspect of the series was the genuine talent it revealed". It was also described as a "sadistic musical bake-off", and "a romp in humiliation". Other aspects of the show have attracted criticisms. The product placement in the show in particular was noted...	0.0631	0.0857
(LLM-generated) ... it's frustratingly mediocre." While the reviewer on the Los Angeles Times site called American Idol "hilarious," adding "not much there to get you excited about," and the reviewer on the Chicago Sun-Times site called it "disconcertingly mediocre."For a show that is not exactly a hit but still pulls in money...	0.2736	0.3708
(Human-written)...You die alone. There is no one with you in the moment when your life fades. But in this, you are not alone. We all share this inherent isolation. We all are born, live, and die alone. There is not one person who ever lived who did not experience this. We all share the empty world the same way...	0.0774	0.0803
(LLM-generated) ... if you learn to trust yourself you can find a special group of friends, a community around you that will help you through the rough days.5. Be a beacon for your friends and family.Remember how amazing it is to have a life mate and family who loves and cares about you? The best thing you can do for someone else ...	0.1865	0.2630
(Human-written)... we look back at the developments so far: 2 August: Samsung unveils its latest flagship model Galaxy Note 7 amid great fanfare in New York. The phone is packed with new features like an iris scanner. The initial response is good and expectations high. It's seen as Samsung's big rival to the upcoming iPhone 7. 19 August...	0.1189	0.1285
(LLM-generated)...a launch on time for the Fourth of July holiday weekend, it may have to rethink the strategy of focusing on India.That's because Indian consumers have always been a difficult group to crack. As with any big, developing nation, Indians are risk takers who aren't particularly eager to buy an Android handset...	0.1784	0.2903
(Human-written)... The manifesto states the Conservatives would ïntroduce a 'funding floor' to protect Welsh relative funding and provide certainty for the Welsh Government to plan for the future, once it has called a referendum on Income Tax powers in the next Parliament. But a Welsh Conservative spokeswoman told BBC Wales: T̈he St David's Day commitment we made to introduce a funding floor for Wales is firm and clear...	0.0973	0.1672
(LLM-generated) ... The Independent has launched its #FinalSay campaign to demand that voters are given a voice on the final Brexit deal. Our petition here The Labour leader, who has said many times he would not sign a hard Brexit, is said to want a more rapid "transition" into post-Brexit British trade arrangements instead. A new spending review could be triggered in July unless a Commons vote is held today to delay it. Downing Street is preparing for the prospect that the UK ...	0.4478	0.4894
(Human-written)... The ground power unit is plugged in. It keeps the electricity running in the plane when it stands at the terminal. The engines are not working, therefore they do not generate the electricity, as they do in flight. The passengers disembark using the airbridge. Mobile stairs can give the ground crew more access to the aircraft's cabin. There is a cleaning service to clean the aircraft after the aircraft lands. Flight catering provides the food and drinks on flight ...	0.0317	-0.0172
(LLM-generated)... It also maintains an electrical system, monitors the tire pressure and other factors to ensure that the plane does not run out of fuel. At the airport, the engine is out at the airstairs while the main gear rests on the ground. The helicopter arrives on the runway with the engines running. The pilots pull up to the aircraft and step onto the runway, then shut down the engines and lift the aircraft onto the taxiway. As the air taxis down the runway, a local airport crew member, typically a pilot ...	0.1331	0.1490
(Human-written)... products of the second world made manifest in the materials of the first world (i.e., books, papers, paintings, symphonies, and all the products of the human mind). World Three, he argued, was the product of individual human beings in exactly the same sense that an animal path is the product of individual animals, and that, as such, has an existence and evolution independent of any individual knowing subjects. The influence of World Three, in his view, on the individual human mind (World Two) ...	0.0481	0.1021
(LLM-generated) ... God's existence is a fact of nature; it is a necessary and sufficient condition of everything else. God is to be known and loved; therefore one has a moral right or obligation to believe in God. The secular conception is the conception of naturalism, according to which there is no absolute moral truth or truthlike quality to our beliefs, which is why it is permissible to believe in God. One's belief in God can be replaced by beliefs, however false, which are equally sincere and important, but do not have the intrinsic value ...	0.1472	0.2113
(Human-written)... deep down we all knew this. Just as deep down we knew it was only a matter of time before mans fantasy would got the best of us. It started innocently enough, why not take a few pounds off your avatar, why not skip the traffic to get to work. The changes we made felt so insignificant and made this farce of a life so much more enjoyable. But like with all things it just kept escalating. It wasn't long before the notion of moderation was all but discarded. While knowing that your make believe job had no real meaning why bother going ...	0.0653	0.0597
(LLM-generated)... And the government also is unable to help because we no longer have any antibiotics with which to combat it because we are already too far into the future.The government in the UK is currently trying to use a bit of a law enforcement tactic to get the people to do as they are told, but this is starting to fail, more and more people are starting to realize that they are being lied to, and the government is being revealed as lying to them. We are currently seeing a huge amount of ...	0.1831	0.1983
(Human-written)... The teenager, who was on his way to the conflict in the Middle East, was returned to the custody of his parents while investigations continued, said Mr Dutton. The interception came about a week after two Sydney brothers, aged 16 and 17, were stopped at the same airport on suspicion of attempting to join IS. The brothers, who have not been named, were also returned to their parents ...	0.1189	0.24340
(LLM-generated)... he told ABC radio station AM.\n\nHis comments came after the ABC reported that on December 20, authorities in Singapore and Dubai were investigating the alleged links of several Australians to the terrorism-related activities. T̈hat's not to suggest there's been any contact with ISIL in some way in this country or anywhere in the world. Obviously any person associated with the terrorist attack in Paris would have been under scrutiny ...	0.2144	0.2680

Table 7: Comparison of the behavioral difference between DetectGPT and our method. The numbers reported refer to the measure $\ell(\boldsymbol{x}, p_{\boldsymbol{\theta}}, q)$ estimated under a query budget of 15 and the T5-large perturbation model. Note that each pair of human-written and LLM-generated texts has the same starting tokens. The exhibited samples are randomly selected.