BEYOND RAW DETECTION SCORES: MARKOV-INFORMED CALIBRATION FOR BOOSTING MACHINE-GENERATED TEXT DETECTION

Chenwang Wu[1] **Yiu-ming Cheung**[1*] **Shuhai Zhang**[2] **Bo Han**[1] **Defu Lian**[3]
[1]Department of Computer Science, Hong Kong Baptist University, Hong Kong, China
[2]School of Software Engineering, South China University of Technology, Guangzhou, China
[3]School of Computer Science, University of Science and Technology of China, Hefei, China
{cscwwu, ymc, bhanml}@comp.hkbu.edu.hk, shuhaizhangshz@gmail.com,
liandefu@ustc.edu.cn

ABSTRACT

While machine-generated texts (MGTs) offer great convenience, they also pose risks such as disinformation and phishing, highlighting the need for reliable detection. Metric-based methods, which extract statistically distinguishable features of MGTs, are often more practical than complex model-based methods that are prone to overfitting. Given their diverse designs, we first place representative metric-based methods within a unified framework, enabling a clear assessment of their advantages and limitations. Our analysis identifies a core challenge across these methods: the token-level detection score is easily biased by the inherent randomness of the MGTs generation process. To address this, we theoretically and empirically reveal two relationships of context detection scores that may aid calibration: Neighbor Similarity and Initial Instability. We then propose a Markov-informed score calibration strategy that models these relationships using Markov random fields, and implements it as a lightweight component via a mean-field approximation, allowing our method to be seamlessly integrated into existing detectors. Extensive experiments in various real-world scenarios, such as cross-LLM and paraphrasing attacks, demonstrate significant gains over baselines with negligible computational overhead. The code is available at https://github.com/tmlr-group/MRF_Calibration.

1 INTRODUCTION

In recent years, generative AI, represented by large language models (LLMs) (Achiam et al., 2023; Radford et al., 2019), has advanced rapidly, and the machine-generated texts (MGTs) they produce often match human writing in fluency, coherence, and diversity. While this technological breakthrough offers immense opportunities, it has also triggered widespread societal concerns, such as the spread of disinformation (Vykopal et al., 2024), the violation of intellectual property rights (Yu et al., 2023b), and phishing attacks (Hong, 2012). Therefore, the research and development of MGT detection technologies hold significant theoretical and practical value in uncovering the distinct patterns of generated text and ensuring a trustworthy AI environment.

An effective detection method is to identify LLM's watermarks (Hou et al., 2024), but this requires injecting watermarks into the LLM, which is often impractical due to high access permissions. Therefore, passive detection methods, including model-based and metric-based methods, have garnered significant attention. Model-based methods, which use a set of human- and machine-generated texts to train a binary classifier, such as OpenAI detector (Solaiman et al., 2019), ChatGPT detector (Guo et al., 2023), SeqXGPT (Wang et al., 2023), and CoCo (Liu et al., 2022). However, such models are often too complex, leading to overfitting to the training data. Instead, metric-based methods exploit the inherent statistical biases of LLM to discriminate MGTs, which is model-agnostic and has better generalization properties. These methods use metrics such as log-likelihood, log-rank,

*Corresponding author: Yiu-ming Cheung (ymc@comp.hkbu.edu.hk).

and entropy. Furthermore, methods such as DetectGPT (Mitchell et al., 2023), DNA-GPT (Yang et al., 2024), and SimLLM (Nguyen-Son et al., 2024) detect MGTs by comparing the differences between a given text and a perturbed, regenerated, or continued text from an alternative model.

Obviously, metric-based methods exhibit diverse designs. Therefore, this paper first systematically examines several representative approaches, including Log-Likelihood (Solaiman et al., 2019), Log-Rank (Mitchell et al., 2023), Entropy (Gehrmann et al., 2019), DetectGPT (Mitchell et al., 2023), Fast-DetectGPT (Bao et al., 2024), and DNA-GPT (Yang et al., 2024), and situates them within a unified framework (**Section 2**). Our analysis reveals that they share a threshold-based detection criterion, with some commonalities, such as the inclusion of auxiliary data (e.g., perturbed texts). This offers a theoretical basis for understanding their core mechanisms, strengths, and limitations.

Based on the unified framework, we summarize the core challenge of metric-based methods (**Section 2**): the imprecision of token-level detection scores. Specifically, since these methods make decisions based on a threshold, their effectiveness is directly tied to score precision. However, randomness introduced by the LLM sampling mechanism can violate their underlying assumptions, leading to biased scores with low discrimination, as shown in "w/o refine" in Fig. 1 (more results can be found in Appendix E.1). Moreover, they tend to derive an overall detection score by naively aggregating token-level scores, failing to correct the underlying imprecision. Therefore, calibrating the token-level detection score is essential for improving overall detection performance.

Given that detection scores are tied to tokens and the LLM generation process induces dependencies among tokens (Achiam et al., 2023), context tokens' detection scores may have relationships that are easy to overlook. Revealing and modeling these relationships may help calibrate these scores. Accordingly, we theoretically and empirically reveal two relationships among detection scores of context tokens (**Section 3**): **Neighbor Similarity**, where adjacent tokens exhibit similar detection scores, and **Initial Instability**, where the detection scores of initial tokens are unstable.

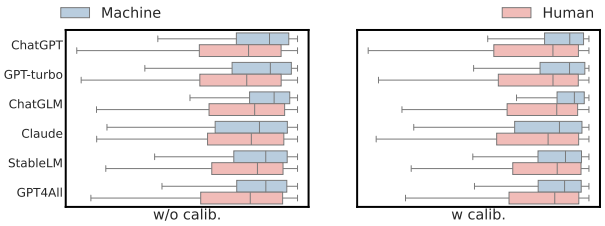

Figure 1: Distribution of token scores obtained by the DetectGPT method without and with score calibration in the Essay dataset. The proposed calibration method enhances the discriminative nature of the token scores.

Finally, building on these two relationships, we propose a Markov-informed score calibration method to enhance MGT detection (**Section 4**). Our method models the identified relationships through Markov random fields and, via a mean-field approximation, implements it as a lightweight iterative neural network. As shown by the more discriminative detection scores in the "w refine" method in Fig. 1, the proposed method boosts the discriminative nature of the scores. Notably, our method can be seamlessly stacked on top of existing detectors without architectural changes, providing flexibility. Compared with complex model-based approaches, our method introduces only a negligible 2×2 parameterization, making its computational delay negligible and less prone to overfitting. Extensive experiments demonstrate our method's enhanced effectiveness. Our contributions can be summarized as follows:

- We view existing metric-based detection methods through a unified lens, which facilitates precise comparison and enables potential improvements.
- We theoretically and empirically demonstrate that token-level detection scores exhibit neighbor similarity and initial instability, offering avenues for improved detection.
- We propose a Markov-informed score calibration method and, via a mean-field approximation, implement it as a lightweight component that can be seamlessly integrated into existing detectors to further unlock their potential.
- We conduct extensive experiments across three datasets to demonstrate superior performance in diverse scenarios, including cross-LLM, cross-domain, mixed-text, and paraphrase attacks.

Table 1: Comparing existing metric-based methods from a unified view. Here, s is the text to be detected containing N tokens, s' is the perturbed text generated by DetectGPT, $\tilde{s}$ and $\hat{s}$ are the regenerated texts of Fast-DetectGPT and DNA-GPT, respectively. Function $\mu(\cdot)$ and $\sigma(\cdot)$ represent the mean and standard deviation of the given set, respectively.

Method	Data	Token-level Score	Score Aggregation	Detection
Log-likelihood	s	$\log p(s_t \vert s_{<t})$	$\frac{1}{N-1}\sum_{t=2}^{N}\log p(s_t \vert s_{<t})$	$score > \epsilon$
Log-Rank	s	$\text{rank}(p(s_t \vert s_{<t}))$	$\frac{1}{N-1}\sum_{t=2}^{N}\text{rank}(p(s_t \vert s_{<t}))$	$score > \epsilon$
Entropy	s	$\sum_{v\in V} p(v \vert s_{<t})\log p(v \vert s_{<t})$	$-\frac{1}{N-1}\sum_{t=2}^{N}\sum_{v\in V} p(v \vert s_{<t})\log p(v \vert s_{<t})$	$score > \epsilon$
DetectGPT	$\{s, s'_1, s'_2, ..., s'_n\}$	$p(s_t \vert s_{<t})$	$\frac{\frac{1}{N-1}\sum_{t=2}^{N} p(s_t \vert s_{<t}) - \mu\left(\{\frac{1}{N-1}\sum_{t=2}^{N} p(s'_{i,t} \vert s'_{i,<t})\}_i\right)}{\sigma\left(\{\frac{1}{N-1}\sum_{t=2}^{N} p(s'_{i,t} \vert s'_{i,<t})\}_i\right)}$	$score > \epsilon$
Fast-DetectGPT	$\{s, \tilde{s}_1, ..., \tilde{s}_n\}$	$p(s_t \vert s_{<t})$	$\frac{\frac{1}{N-1}\sum_{t=2}^{N} p(s_t \vert s_{<t}) - \mu\left(\{\frac{1}{N-1}\sum_{t=2}^{N} p(\tilde{s}_{i,t} \vert s, \tilde{s}_{i,<t})\}_i\right)}{\sigma\left(\{\frac{1}{N-1}\sum_{t=2}^{N} p(\tilde{s}_{i,t} \vert s, \tilde{s}_{i,<t})\}_i\right)}$	$score > \epsilon$
DNA-GPT	$\{s, \hat{s}_1, ..., \hat{s}_n\}$	$p(s_t \vert s_{<t})$	$\frac{1}{N-1}\sum_{t=2}^{N}\log p(s_t \vert s_{1:t-1}) - \frac{1}{n}\sum_{i=1}^{n}\frac{1}{N_t-1}\sum_{t=2}^{N_t}\log p(\hat{s}_{i,t} \vert \hat{s}_{i,1:t-1})$	$score > \epsilon$

2 A Unified Perspective on Metric-Based Detection

Although model-based methods have shown competitive potential in specific domains, they are often too complex, leading to a tendency to overfit their training data. This limitation requires them to retrain or fine-tune for newly released LLMs, which hinders their generalizability. In contrast, metric-based methods extract discriminative features from MGT, and their model-agnostic nature provides superior generalization potential. Given the diverse implementations of representative metric-based methods such as Log-Likelihood (Solaiman et al., 2019), Log-Rank (Gehrmann et al., 2019), Entropy (Gehrmann et al., 2019), DetectGPT (Mitchell et al., 2023), Fast-DetectGPT (Bao et al., 2024), and DNA-GPT (Yang et al., 2024), we first provide a systematic examination of them from a unified perspective. This facilitates a deeper understanding of their mechanisms and allows for a fair comparison of their strengths and weaknesses. As illustrated in Table 1, we compare these methods across data, score aggregation, and detection dimensions. Note that their diverse core metric designs are not discussed here. This is because we aim to design a general enhancement framework decoupled from specific detectors, requiring us to start from possible commonalities rather than diverse designs.

- **Data**. Log-likelihood, Log-Rank, and Entropy are computationally efficient as they rely solely on the original input text s. However, the detection error based on a single text may be large, because the randomness inherent in the LLM sampling mechanism may cause the MGT to deviate from these methods' underlying assumptions, e.g., Log-Rank assumes that the generated tokens have high rankings. In contrast, DetectGPT, Fast-DetectGPT, and DNA-GPT incorporate multiple perturbed (i.e., s') or regenerated (i.e., $\tilde{s}$ and $\hat{s}$) samples, which mitigates the errors caused by randomness. However, this comes at the cost of increased computational overhead compared to single-text-based methods.

- **Score Aggregation**. Although these methods appear to calculate scores differently, they all tend to directly aggregate token scores to obtain the final text score, typically through summation. As discussed, the randomness introduced by the LLM generation process may cause token-level scores to be biased. Therefore, the direct aggregation of these potentially imprecise token scores may result in an inaccurate final detection score.

- **Detection**. These methods employ threshold-based detection mechanisms, whose effectiveness relies heavily on the accuracy of their calculated scores. Including uncalibrated, high-noise scores in threshold-based decision-making may lead to poor performance.

In summary, to enhance detection, existing methods incorporate more textual information (e.g., regenerated texts in DetectGPT and Fast-DetectGPT) and use different score calculation strategies (e.g., the Likelihood difference between the detected and regenerated text in DNAGPT). However, as we discussed, they fail to address the underlying token-level errors caused by inherent randomness, limiting their detection potential. Considering that detection scores are tied to tokens and the LLMs' generative mechanism induces dependencies among tokens, revealing and modeling the relationships between tokens may help correct score errors and thus improve detection effectiveness.

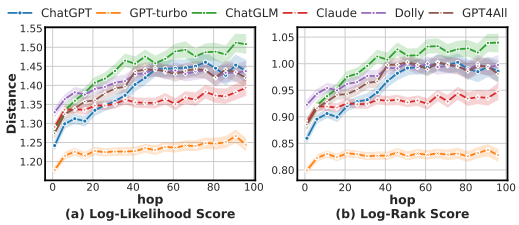

Figure 2: The detection score distances (Mean Absolute Difference) of neighbors at different hops in the Essay dataset. Log-likelihood and Log-Rank score are used.

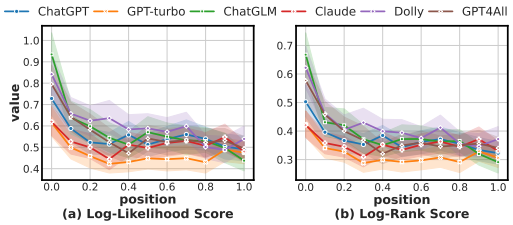

Figure 3: The detection score distances (Mean Absolute Difference) of 1-hop neighbors at different normalized relative positions in Essay. Log-likelihood and Log-Rank score are used.

3 Relation between Contextual Token-Level Detection Scores

To understand the relationship between context tokens' detection scores, following existing work (Liu et al., 2023), we consider the token generation process of a simplified model: a single-layer transformer model with single-head attention:

$$x_{t+1} = \mathcal{F}(a_t), \quad \text{where} \quad a_t = \text{softmax}\left(1/t \cdot x_t W_Q W_K^\top X_{t-1}^\top\right) X_{t-1} W_V W_O. \tag{1}$$

$x_t \in \mathbb{R}^{1\times d}$ is the embedding of token s_t, and d denotes the embedding dimension. The matrix $X_{t-1} \in \mathbb{R}^{(t-1)\times d}$ is stacked by the embeddings $x_1, \ldots, x_{t-1}$, where the j-th row is x_j. $W_Q, W_K, W_V \in \mathbb{R}^{d\times d}$ and $W_O \in \mathbb{R}^{d\times d}$ are the attention weights. Following the attention block, an MLP block, denoted as $\mathcal{F} : \mathbb{R}^{1\times d} \to \mathbb{R}^{1\times d}$, is applied, and it is a two-layer network with skip connections:

$$\mathcal{F}(x) = x + W_2 \,\text{relu}\,(W_1 x).$$

As shown in Table 1, the detection score of token s_t is usually the function of s_t (i.e., x_i), and x_i is related to the attention scores $\alpha_{t-1} = \text{softmax}\left(1/t \cdot x_t W_Q W_K^\top X_{t-1}^\top\right)$ in Eq. (1). The following theorem will reveal the relationship between attention scores, which in turn help us understand the relationship between detection scores of context tokens.

Theorem 1. *Let $\lambda_K, \lambda_Q, \lambda_V, \lambda_O$ be the largest singular values of parameters W_K, W_Q, W_V, W_O, respectively, and let $W = W_V W_O W_Q W_K^\top$. For the transformer defined in Eq. (1), assuming normalized inputs ($\|x_t\|_2 = 1$ for all t) and constants $c, \epsilon > 0$, consider $a_t x_{t+1}^\top \geq (1-\delta)\|a_t\|_2$ with $\delta \leq \left(\frac{c\epsilon}{\lambda_Q\lambda_K\lambda_V\lambda_O}\right)^2$. If x_ℓ satisfies $x_\ell W x_\ell^\top \geq c$ and $x_\ell W x_\ell \geq \epsilon^{-1}\max_{j\in[\ell], j\neq\ell} x_j W x_\ell^\top$, then*

$$\alpha_{t+1,l} \leq \frac{\exp\left(C_l \cdot \alpha_{t,l} + \eta\right)}{\exp\left(C_l \cdot \alpha_{t,l} + \eta\right) + \sum_{j\neq l}\exp\left(C_j \cdot \alpha_{t,j} - \eta\right)},$$

$$\alpha_{t+1,l} \geq \frac{\exp\left(C_l \cdot \alpha_{t,l} - \eta\right)}{\exp\left(C_l \cdot \alpha_{t,l} - \eta\right) + \sum_{j\neq l}\exp\left(C_j \cdot \alpha_{t,j} + \eta\right)},$$

where

$$C_j = \frac{x_j W x_j^T}{t\,|a_t|_2}, \text{and } \eta = \frac{(1+\sqrt{2})\epsilon x_j W x_j^T}{(t+1)\,|a_t|_2}.$$

The proof can be found in Appendix C. This theorem establishes the upper and lower bounds for the attention score at step $t+1$, which are determined by the attention scores at step t. In Transformer, the function mapping $a_t \to \log p(x_{t+1})$ is continuous, constraints on the dynamics of a_t naturally propagate to the detection scores. Therefore, Theorem 1 can reveal two relationships in token-level detection scores:

- **Neighbor Similarity**, where the detection scores of adjacent tokens in a sequence exhibit statistically lower variance compared to tokens that are far apart. This is because Theorem 1 creates a positive feedback loop analogous to simulated annealing (Kirkpatrick et al., 1983), where high scores at the current step lead to high scores at the next, and vice versa. That is, the attention (also token-level detection scores) cannot shift fast between steps.

- **Initial Instability**, where the detection scores of initial tokens in a sequence are statistically unstable compared to the subsequent tokens (i.e., fluctuate greatly). This is because these bounds are closely tied to the current step t. When t is small (early position), η and C are large, allowing for dramatic fluctuations in a_t. That is, the attention score (also token-level detection scores) is unstable in the initial generation process.

Given that analyzing the dynamics of full-scale LLMs is mathematically intractable, we follow a practical setup Liu et al. (2023) and start with a simplified model, which provides theoretical motivation for the neighbor similarity and initial instability phenomena. We then empirically verify these two findings.

First, to empirically validate the neighbor similarity property, we evaluated the distance (mean absolute difference) in detection scores across k hops (i.e., $\frac{1}{|S|}\frac{1}{N-K}\sum_{s\in S}\sum_{t=0}^{N-K}|score(s_t) - score(s_{t+k})|$, where S is the text set, and $score(s_t)$ is provided in Table 1). As illustrated in Fig. 2 (more results can be found in Appendix E.2), there is a clear positive correlation between the detection score distance and the hop, and adjacent tokens have the highest detection score similarity; thereby providing empirical evidence for our theoretical finding.

Second, to validate the initial instability property, we analyzed the distance in detection score between adjacent tokens at different percentage positions (i.e., $\frac{1}{S}\sum_{s\in S}|score(s_t) - score(s_{t+1})|$). Fig. 3 illustrates that the score difference is substantially larger for tokens at the beginning of the text and progressively decreases, eventually stabilizing. More results can be found in Appendix E.2. Considering our established finding on neighbor similarity (i.e., adjacent scores should be highly similar), a high detection score difference at the sequence beginning indicates a significant instability in detection scores of initial tokens.

4 Markov-Informed Detection Score Calibration

Based on the revealed relationships, this section uses an MRF to capture them (Section 4.1) and adopts the mean field approximation to model the MRF model as a lightweight component stacked on existing detectors to calibrate detection scores (Section 4.2), thereby enhancing detection.

4.1 Markov Random Field for MGT Detection

We capture these two types of relationships by modeling the joint probability distribution of text's token detection scores through pairwise Markov Random Fields (pMRF). Specifically, for each token s_t in text s, we assign a binary random variable y_{s_t}, where $y_{s_t} = 0$ and $y_{s_t} = 1$ indicate a human- or machine-generated token [1], respectively, as measured by the detection score of the token. Let y_s denote the label set for all tokens in text s, the pMRF over these tokens can be formalized as a Gibbs distribution: $P(y_s) = \frac{1}{Z}\exp(-E(s, y_s))$, where Z is a normalizing constant and $E(s, y_s)$ is the energy function. Our objective is to maximize the posterior probability of the token labels y_s by minimizing the global energy function $E(s, y_s)$. The energy function typically consists of two components: the unary potential Ψ_U and the pairwise potential Ψ_P:

$$E(s, y_s) = \sum_{t=1}^{N}\Psi_U\left(s_t, y_{s_t}\right) + \sum_{t=1}^{N}\sum_{s_j\in\mathcal{N}(s_t)}\Psi_P\left(y_{s_t}, y_{s_j}\right),$$

where $\mathcal{N}(s_t)$ denotes the adjacent tokens of token s_t, i.e., $\mathcal{N}(s_t) = \{s_{t-1}, s_{t+1}\}$ (if existing).

Unary potential $\Psi_U\left(s_t, y_{s_t}\right)$ quantifies the cost of assigning label y_{s_t} to token s_t. We let $\Psi_U\left(s_t, y_{s_t}\right) = -\log p(s_t)$, where $p(s_t)$ is the output probability from the original detector. For detectors without probability output, it is measured by the 0-1 normalized score of token s_t.

Pairwise potential $\Psi_P\left(y_{s_t}, y_{s_j}\right)$ models the similarity in detection scores between adjacent tokens. A penalty is applied if two adjacent tokens are assigned different labels; otherwise, a reward is given. This enforces label smoothness and captures the neighbor similarity property:

$$\Psi_P\left(y_{s_t}, y_{s_j}\right) = w\cdot(2\cdot I(y_{s_t}\neq y_{s_j}) - 1), \tag{2}$$

[1]Note that token labels are not absolute but depend on the context in which they appear. For example, "the" can be a human token or a machine token depending on the text.

where $I(\cdot)$ is the indicator function, and the reward and penalty factor $w \geq 0$. This implies an energy penalty of w when adjacent tokens have different labels; otherwise, the reward is $-w$.

To model the initial instability property, we introduce a positional weighting function $\beta(t)$ in the binary potential. This function assigns lower weights to binary potentials at earlier positions, thereby mitigating the amplification of energy errors caused by unstable initial neighbor tokens. In this paper, we define the positional weighting function $\beta(t)$ as a Sigmoid function to ensure a smooth transition of weights, and the revised binary potential is then given by:

$$\Psi_P\left(y_{s_i}, y_{s_j}\right) = \beta(j) \cdot w \cdot (2 \cdot I(y_{s_i} \neq y_{s_j}) - 1), \text{ with } \beta(t) = \frac{1}{1 + exp(-(t - t_0))}, \quad (3)$$

where t_0 is the weighted center, effectively suppressing the pairwise potential of tokens before t_0.

4.2 Mean Field Approximate in MGT Detection

Given the MRF model, this subsection details how to model it as a lightweight component stacked on the original detector through mean field approximation theory, thereby enhancing detection.

In the MRF posterior probability $P(y_s) = \frac{1}{Z}\exp(-E(s, y_s))$, the partition function $Z = \sum_{y_s} exp(-E(s, y_s))$, which is obtained by adding over all possible combinations of y_s. For a text with N tokens, there are 2^N combinations, making exact computation of $P(y_s|s)$ infeasible. Inspired by existing work (Deng et al., 2022), we employ mean-field theory for approximate inference. Its core idea is to use a simpler, factorized distribution $Q\left(y_s\right) = \prod_{t=1}^{N} Q_{s_t}\left(y_{s_t}\right)$ to approximate the true joint distribution $P(y_s)$, achieved by minimizing the KL divergence between these two distributions:

$$\begin{aligned} D(Q\|P) =& \mathbb{E}_{y_s \sim Q}\left[\log Q\left(y_s\right)\right] - \mathbb{E}_{y_s \sim Q}\left[\log P\left(y_s\right)\right] \\ =& \sum\nolimits_{t=1}^{N} \mathbb{E}_{y_{s_t} \sim Q_{s_t}}\left[\log Q_{s_t}\left(y_{s_t}\right)\right] + \mathbb{E}_{y_s \sim Q}\left[E\left(s, y_s\right)\right] + \log Z. \end{aligned} \quad (4)$$

The first equation is the definition of KL divergence, and the second is obtained by subsituting $P(y_s)$ and $Q(y_s)$. Then, we define a Lagrangian composed of all terms involving $Q_{s_t}\left(y_{s_t}\right)$ in $D(Q\|P)$:

$$L_{s_t}(Q) = Q_{s_t}\left(y_{s_t}\right) \log Q_{s_t}\left(y_{s_t}\right) + \mathbb{E}_{y_s \sim Q}\left[E\left(s, y_s\right)\right] + \lambda(\sum\nolimits_{y_{s_t}} Q_{s_t}\left(y_{s_t}\right) - 1). \quad (5)$$

Here, the term involving Lagrange multiplier λ assures that Q_{s_i} is a proper probability distribution. Now we take derivatives of Eq. (5) with respect to $Q_{s_t}(y_{s_t})$ and set the corresponding derivative to 0, then we get the optimal $Q_{s_t}(y_{s_t})$:

$$Q_{s_t}(y_{s_t}) = \frac{1}{z} exp(\Psi_U(s_t, y_{s_t}) - \sum\nolimits_{s_j \in \mathcal{N}(s_t)} \mathbb{E}_{y_{s_j} \sim Q_{s_j}}\left[\Psi_P(y_{s_t}, y_{s_j})\right]). \quad (6)$$

We can express the single token calculation in Eq. (6) as a matrix form of multiple token calculations, which involves three steps:

- **Unary potential calculation**. Corresponding to the unary potential $\Psi_U(s_t, y_{s_t})$ of token s_t being $-\log p(s_i)$, we can express the unary potential of text s in matrix form $-\log H$, where the i-th row of H corresponds to token s_i's prior probability and the two columns correspond to identity labels (HGT and MGT), that is, $H = [1 - p(s), p(s)]$, where $p(s) = [p(s_1), ..., p(s_N)]^\top$.
- **Pairwise potential calculation**. For the revised pairwise potential (i.e., Eq. (3)), we can define the weighted adjacency matrix as A^{corr}, where $A^{corr}_{t,t+1} = \beta(t+1)$ for all $t = 0, ..., N-2$, $A^{corr}_{t-1,t} = \beta(t-1)$ for all $t = 1, ..., N-1$, and 0 otherwise. Then the matrix form of the weighted pairwise potential is $A^{corr} Q \begin{bmatrix} -w & w \\ w & -w \end{bmatrix}$. This update strategy enforces that all relationships receive the same reward or penalty. However, the influence of MGT neighbors and HGT neighbors is intuitively different, so this kind of reward and punishment mechanism with the same weight will limit the expressive ability of MRF. To this end, we relax the weights for different relationships and get the pairwise potential as $A^{corr} Q(W_{mrf} \odot \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix})$, where $W_{mrf} \in \mathbb{R}_+^{2\times 2}$.
- **Normalization**. The operator $\frac{1}{z} exp(\cdot)$ in Eq. (6) can be naturally modeled as Softmax function.

In summary, we get the following update rule for tokens' detection scores:

$$Q = \text{softmax}\left(\log H - A^{corr} Q \left(W_{mrf} \odot \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix}\right)\right). \tag{7}$$

Notably, Eq. (7) shows that the computation of Q relies on Q itself; hence, iterative computation is required. Initializing the initial Q to H, we can get the iterative version as follows:

$$Q^t = \text{softmax}\left(\log Q^{t-1} - A^{corr} Q^{t-1} \left(W_{mrf} \odot \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix}\right)\right), \text{ where } Q^0 = H. \tag{8}$$

In addition to using position weight function $\beta(t)$ in pairwise potential to reduce the impact of initial unstable scores, we also use this position weight function on the final calibrated scores Q^T of T iterations to reduce their impact on detection:

$$Q_{final} = [\beta(1), ..., \beta(N)] \odot Q^T. \tag{9}$$

Algorithm 1 Markov-informed Enhancement Framework

1: **Input:** Text s to be detected, the original detector $f_1 \circ f_2$, iteration steps T, MRF weights W.
2: Construct A^{corr} based on s.
3: Get each token s_i's detection score from the detection score calculation module f_1, and set $Q^0 = H$.
4: **for** $t = 0$ **to** $T - 1$ **do**
5: Update Q according to Eq. (8).
6: **end for**
7: Calculate Q_{final} according to Eq. (9).
8: **Return** detection score $f_2(Q_{final})$ of text s.

The MRF-informed calibration component is then directly stacked on the original detector to calibrate the token detection scores. Specifically, if the detector is formalized as a combination of the detection score calculation module f_1 and the detection module f_2, i.e., $f(s) = f_1 \circ f_2(s)$, the MRF-informed component of Eq. (9) is defined as f_{mrf}, then the enhanced detector is $f_{enh.}(s) = f_1 \circ f_{mrf} \circ f_2(s)$. The complete inference process is shown in Alg. 1. To learn weights W, we use supervised training: $\mathcal{L} = -\sum_{s \in \mathcal{D}_{train}} (Y_s \log f_{enh.}(s) + (1 - Y_s) \log(1 - f_{enh.}(s)))$.

Computational Complexity. The MRF layer can be computed using sparse-dense matrix multiplication. For a text containing N tokens, the number of iterations is T, resulting in a computational complexity of $\mathcal{O}(NT)$, which is negligible for calculating the detection score. We will also empirically verify the efficiency of our method in Appendix E.15.

5 Experiments

Dataset. We conducted experiments on three widely used public datasets: Essay (Verma et al., 2024), Reuters (Verma et al., 2024), DetectRL (Wu et al., 2024), and TruthfulQA (Lin et al., 2022). The Essay and Reuters datasets collect machine text generated by GPT4All, ChatGPT, ChatGPT-turbo, ChatGLM, Dolly, and Claude. The TruthfulQA dataset collects machine text from GPT4, ChatGPT-turbo, ChatGLM, Dolly, ChatGPT, and StableLM. The DetectRL dataset not only includes pure machine-generated text by Google-PaLM, ChatGPT, and Llama-2-70b, but its unique mixed, paraphrased, and cross-domain texts also allow us to more comprehensively evaluate the model's performance in complex real-world scenarios. For a complete description of the datasets, please refer to Appendix D.1.

Baselines. We select the following metric-based methods for comparison and enhancement: Log-Likelihood (Likelihood) (Solaiman et al., 2019), Log-Rank (Gehrmann et al., 2019), Entropy (Gehrmann et al., 2019), DetectGPT (Mitchell et al., 2023), Fast-DetectGPT (FastGPT) (Bao et al., 2024), DNA-GPT (Yang et al., 2024), Repreguard (Chen et al., 2025), Lastde (Xu et al., 2025), FourierGPT (Xu et al., 2024), and Binoculars Hans et al. (2024). The versions equipped with the proposed method are defined with 'M' suffix, e.g., Likelihood-M. Furthermore, although we focus on metric-based methods, we also compare with model-based methods ChatGPT-D (Guo et al., 2023) and MPU (Tian et al., 2024). Their details can be found in Appendix D.2.

Metrics. First, as a binary classification problem, we use the area under the receiver operating characteristic curve (AUROC). Second, following (Tufts et al., 2024; Fraser et al., 2025; Hans et al., 2024), we recognize the negative impact of misclassifying human text as machine-generated text. Therefore, another important evaluation metric is the true positive rate (TPR) at a low false positive rate (FPR). Specifically, we measure the TPR at an FPR of 1%, denoting this as TPR@FPR-1%.

Table 2: Performance concerning AUROC (%) on Essay (left) and DetectRL (right).

Method	Essay (Training Text: GPT4All)							DetectRL (Training Text: Llama-2-70b)			
	GPT4All	ChatGPT	Dolly	ChatGLM	Claude	ChatGPT-turbo	Avg.	Llama-2-70b	ChatGPT	Google-PaLM	Avg.
Likelihood	96.16$_{\pm 0.30}$	98.79$_{\pm 0.19}$	90.90$_{\pm 1.33}$	99.29$_{\pm 0.25}$	92.76$_{\pm 0.23}$	99.13$_{\pm 0.19}$	96.17	78.58$_{\pm 0.41}$	66.61$_{\pm 0.99}$	71.42$_{\pm 0.49}$	72.20
Likelihood-M	**98.58**$_{\pm 0.14}$	**99.47**$_{\pm 0.11}$	**94.59**$_{\pm 0.96}$	**99.54**$_{\pm 0.18}$	**94.82**$_{\pm 0.36}$	**99.72**$_{\pm 0.13}$	**97.79**	**87.61**$_{\pm 0.48}$	**73.70**$_{\pm 0.58}$	**81.21**$_{\pm 1.21}$	**80.84**
Log-Rank	96.55$_{\pm 0.31}$	98.95$_{\pm 0.13}$	90.08$_{\pm 1.28}$	99.36$_{\pm 0.13}$	92.01$_{\pm 0.20}$	99.24$_{\pm 0.15}$	96.03	79.67$_{\pm 0.46}$	65.85$_{\pm 0.94}$	70.66$_{\pm 0.40}$	72.06
Log-Rank-M	**98.57**$_{\pm 0.06}$	**99.41**$_{\pm 0.09}$	**93.82**$_{\pm 0.91}$	**99.55**$_{\pm 0.08}$	**92.91**$_{\pm 0.30}$	**99.64**$_{\pm 0.10}$	**97.32**	**90.20**$_{\pm 0.44}$	**75.32**$_{\pm 0.91}$	**84.34**$_{\pm 0.61}$	**83.29**
Entropy	74.19$_{\pm 1.62}$	89.49$_{\pm 0.34}$	73.26$_{\pm 1.48}$	84.11$_{\pm 0.77}$	86.58$_{\pm 0.66}$	95.94$_{\pm 0.35}$	83.93	66.21$_{\pm 0.88}$	63.09$_{\pm 1.05}$	60.73$_{\pm 0.91}$	63.34
Entropy-M	**83.52**$_{\pm 0.73}$	**93.28**$_{\pm 0.15}$	**81.11**$_{\pm 0.91}$	**91.44**$_{\pm 0.35}$	**87.96**$_{\pm 0.48}$	**96.75**$_{\pm 0.17}$	**89.01**	**69.97**$_{\pm 0.90}$	**65.26**$_{\pm 1.36}$	**65.82**$_{\pm 0.96}$	**67.02**
DetectGPT	50.81$_{\pm 0.58}$	46.40$_{\pm 0.77}$	57.48$_{\pm 0.84}$	50.41$_{\pm 1.70}$	41.54$_{\pm 0.60}$	17.90$_{\pm 1.25}$	44.09	52.37$_{\pm 0.59}$	50.22$_{\pm 0.60}$	43.19$_{\pm 1.41}$	48.60
DetectGPT-M	**95.37**$_{\pm 2.42}$	**96.20**$_{\pm 2.48}$	**85.39**$_{\pm 7.02}$	**95.97**$_{\pm 2.93}$	**80.29**$_{\pm 10.43}$	**98.49**$_{\pm 0.78}$	**91.95**	**78.81**$_{\pm 3.68}$	**64.15**$_{\pm 4.48}$	**75.90**$_{\pm 4.53}$	**72.95**
FastGPT	64.63$_{\pm 1.53}$	67.68$_{\pm 1.70}$	47.17$_{\pm 1.53}$	71.08$_{\pm 1.51}$	**75.31**$_{\pm 0.90}$	**88.62**$_{\pm 0.67}$	69.08	**67.72**$_{\pm 1.02}$	**58.50**$_{\pm 1.22}$	56.68$_{\pm 1.21}$	60.97
FastGPT-M	**87.22**$_{\pm 3.40}$	**91.56**$_{\pm 3.33}$	**82.61**$_{\pm 17.98}$	**95.36**$_{\pm 0.48}$	59.29$_{\pm 1.89}$	63.48$_{\pm 18.13}$	**79.92**	66.55$_{\pm 3.36}$	55.19$_{\pm 3.26}$	**63.30**$_{\pm 4.37}$	**61.68**
DNA-GPT	98.08$_{\pm 0.23}$	96.56$_{\pm 0.28}$	92.78$_{\pm 0.68}$	98.00$_{\pm 0.13}$	89.92$_{\pm 0.29}$	96.22$_{\pm 0.27}$	95.26	71.91$_{\pm 0.91}$	64.14$_{\pm 0.91}$	67.32$_{\pm 0.94}$	67.79
DNA-GPT-M	**99.68**$_{\pm 0.07}$	**98.88**$_{\pm 0.06}$	**97.04**$_{\pm 0.49}$	**99.26**$_{\pm 0.05}$	**94.73**$_{\pm 0.28}$	**98.85**$_{\pm 0.09}$	**98.07**	**74.39**$_{\pm 1.02}$	**64.36**$_{\pm 1.08}$	**70.50**$_{\pm 1.03}$	**69.75**

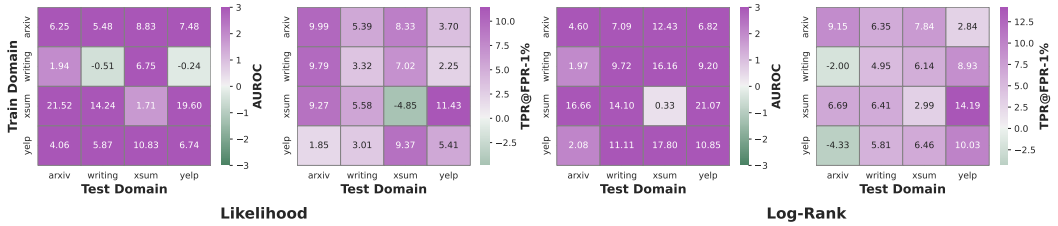

Figure 4: The performance improvement of the proposed method on Likelihood and Log-Rank. Values greater than 0 indicate an enhanced effect.

5.1 Performance Comparison

In this section, we evaluate the enhancement effectiveness of the proposed method in various real-world scenarios, including cross-LLM, cross-domain, detecting mixed machine text, and resisting paraphrase attacks. Details of these scenarios are provided in Appendix D.3.

Performance across Different LLMs. Table 2 reports AUROC for detectors trained on GPT4All (Essay) and Llama-2-70b (DetectRL) and evaluated on texts from various LLMs. TPR@FPR-1% results are provided in Table 3 in the Appendix. Besides, results of detectors trained on other LLM texts and on the Reuters dataset appear in Tables 4–15 of the Appendix. The proposed method yields significant gains for nearly all baselines. For example, on Essay, it raises Likelihood from 52.4% to 77.86% (+25.46%). Encouragingly, our method also benefits weak detectors: DetectGPT significantly improves from 0.15% to 37.18% (+37.03%), suggesting that the assumptions underlying its scoring are reasonable and that underperformance mainly stemmed from score estimation error. Finally, detection performance on texts from the same LLM is not always superior to cross-LLM. For example, on Essay, performance on ChatGLM is particularly strong and even exceeds the intra-LLM case of GPT4All, possibly because ChatGLM texts are more discriminative.

Performance across Different Domains. We evaluate cross-domain performance in four high-risk domains: arXiv, Writing Prompts, XSum, and Yelp Reviews. Results are summarized in Fig. 4; additional enhanced results for more detectors under cross-domain settings are provided in Fig. 21 and Fig. 22 of the Appendix. In most settings, detectors equipped with our strategy show substantially stronger cross-domain generalization. We attribute this to calibrating the detection score with only a few carefully designed parameters (a 2x2 reward–penalty coefficient matrix), which helps prevent overfitting and thereby improves out-of-domain detection.

Performance against Mixed Texts. In practice, human–AI collaboration is pervasive, leading to widespread mixed human–machine text. We therefore evaluate the proposed strategy for mixed-text detection. Two training strategies are considered: training on the original text and training on the mixed text. Results concerning AUROC are shown in Fig. 5, with TPR@FPR-1% shown in Fig. 23 of the Appendix. In most settings, the proposed strategy enhances the detector's ability to recognize mixed-text. Moreover, comparing training on original versus mixed text shows that simply training on mixed text does not improve the detection ability of mixed text, highlighting the focus of mixed text detection research.

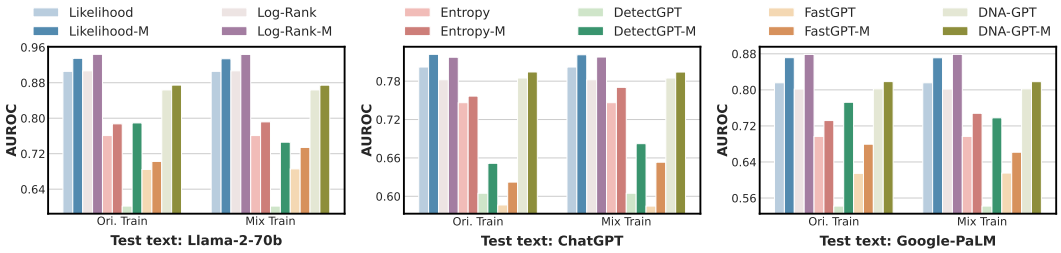

Figure 5: Detection performance concerning AUROC under different LLM mixed texts. All detectors are trained on Llama-2-70b texts.

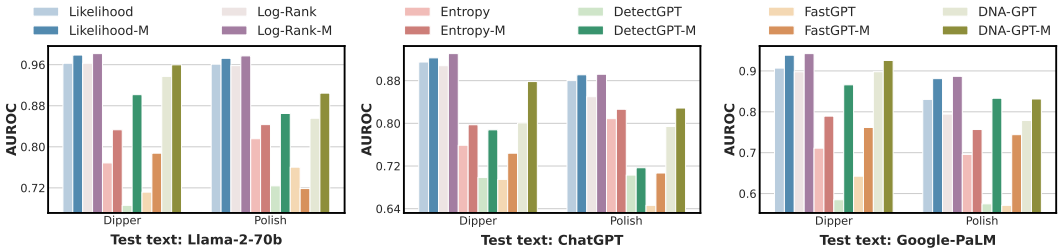

Figure 6: Detection performance concerning AUROC under different paraphrasing texts (Dipper and Polish). All detectors are trained on Llama-2-70b texts.

Performance against Paraphrasing Attacks. Prior work (Sadasivan et al., 2023) shows that MGT detection is typically vulnerable to paraphrasing attacks, where an adversary rewrites a passage without altering its semantics to evade detection. We therefore assess the robustness gains of our framework on the Dipper and Polish paraphrase attacks provided by DetectRL. The results in Fig. 6 and Fig. 24 clearly indicate that, even in adversarial settings, our strategy yields encouraging improvements in robustness against paraphrasing attacks.

5.2 Ablation Study

We introduce an MRF layer and a positional weighting function to model neighbor similarity and initial instability, respectively. In this section, we verify their effectiveness through ablation studies, denoted as "w/o MRF" and "w/o Pos". Results on the Essay dataset are shown in Fig. 7, with additional results in Fig. 25 and Fig. 26 of the Appendix. We can find that removing either component significantly drops performance, while retaining only one still outperforms the baseline detector, underscoring their designs' rationality. Besides, the contribution of each component varies by detector type. For single-text detectors like Likelihood and Log-Rank, the positional weighting function provides the most improvement by addressing instability in initial scores. However, for methods that aggregate multiple text scores, such as DetectGPT and FastGPT, this instability is already partially mitigated, making the MRF-based score calibration the primary source of gains.

5.3 MRF vs. Neural Network

This paper proposes a Markov-informed calibration method to model the relationship of detection scores of context tokens. To highlight the rationale of this design, we compare it with methods that directly use neural networks to calibrate scores. A simple three-layer neural network is used here, defined with the "nn" suffix. The comparison results on Likelihood are shown in Fig. 8, and more results can be found in Fig. 31 and Fig. 32 of the Appendix. Although NN-based methods exhibit competitive performance in some settings for intra-LLMs, their generalization ability on cross-LLMs drops significantly. This suggests that it does not truly learn the general ability to correct scores, but rather overfits the training data. In contrast, our strategy shows good generalization.

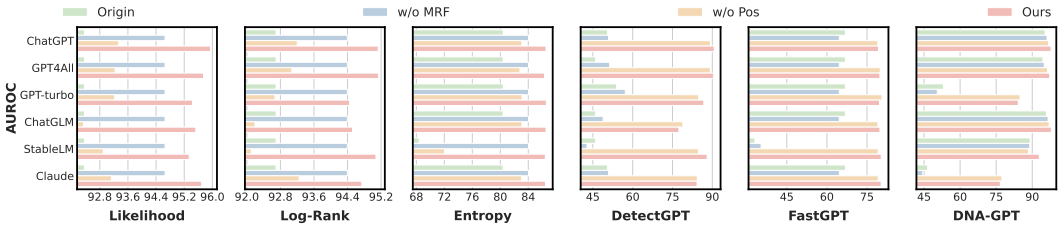

Figure 7: Ablation results on the Essay dataset. The y-axis represents the LLM text on which the detector was trained, and the x-axis represents the average performance across LLMs.

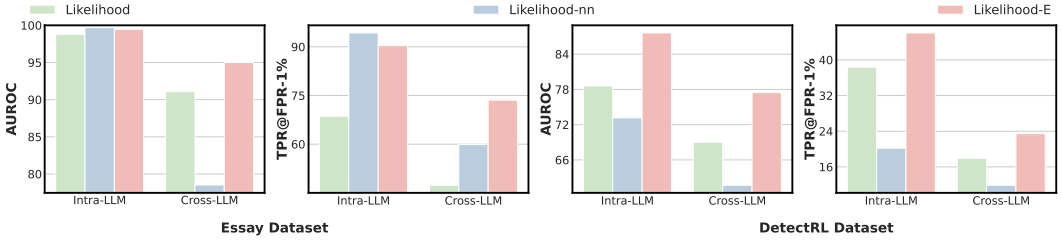

Figure 8: Comparison with NN-based calibration methods. The detector used is Likelihood.

6 Conclusion

This paper has systematically examined representative metric-based detectors within a unified framework, revealing a core challenge: the inherent randomness of the LLM generation process leads to inaccurate detection scores, and naive aggregation of existing methods fails to fix this. Therefore, we have theoretically and empirically established two key properties of these scores: neighbor similarity and initial instability. Building on these insights, we have proposed a Markov-informed score calibration method that captures token relationships and corrects the biased scores produced by base detectors. Extensive experiments show substantial and consistent performance gains of the proposed method. Admittedly, the proposed method relies on modeling the relationships between context tokens to calibrate detection scores. Consequently, our method is not directly applicable to detection methods that do not provide this fine-grained, token-level output.

Acknowledgment

This work was supported by the RGC Senior Research Fellow Scheme under the grant: SRFS2324-2S02, RGC Young Collaborative Research Grant No. C2005-24Y, and HKBU CSD Departmental Incentive Scheme.

Ethics Statement

This paper proposes a Markov-informed calibration strategy to enhance machine-generated text detection and mitigate the potential risks posed by machine-generated text, including disinformation and phishing. Our work does not involve ethical issues such as dataset releases, potentially harmful insights, potential conflicts of interest and sponsorship, discrimination/bias/fairness concerns, privacy and security issues, legal compliance, and research integrity issues.

Reproducibility Statement

Our code is available at `https://github.com/tmlr-group/MRF_Calibration`, and all datasets used in our experiments (Essay, Reuters, and DetectRL) are publicly available for download. In addition, we provide detailed implementation details in the Appendix, including data partitioning, the fixed seeds, the learning rate, the training batch size, and the two parameters t_0 and T introduced by the proposed strategy, to ensure reproducibility of our work.

REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Guangsheng Bao, Yanbin Zhao, Zhiyang Teng, Linyi Yang, and Yue Zhang. Fast-detectgpt: Efficient zero-shot detection of machine-generated text via conditional probability curvature. In *ICLR*, 2024.

Xin Chen, Junchao Wu, Shu Yang, Runzhe Zhan, Zeyu Wu, Ziyang Luo, Di Wang, Min Yang, Lidia S Chao, and Derek F Wong. Repreguard: Detecting llm-generated text by revealing hidden representation patterns. *arXiv preprint arXiv:2508.13152*, 2025.

Leyan Deng, Chenwang Wu, Defu Lian, Yongji Wu, and Enhong Chen. Markov-driven graph convolutional networks for social spammer detection. *IEEE Transactions on Knowledge and Data Engineering*, 35(12):12310–12322, 2022.

Kathleen C Fraser, Hillary Dawkins, and Svetlana Kiritchenko. Detecting ai-generated text: Factors influencing detectability with current methods. *Journal of Artificial Intelligence Research*, 82: 2233–2278, 2025.

Sebastian Gehrmann, Hendrik Strobelt, and Alexander M Rush. Gltr: Statistical detection and visualization of generated text. *arXiv preprint arXiv:1906.04043*, 2019.

GPTZero. Gptzero official website. [Online], 2023. `https://gptzero.me`.

Chenchen Gu, Xiang Lisa Li, Percy Liang, and Tatsunori Hashimoto. On the learnability of watermarks for language models. *arXiv preprint arXiv:2312.04469*, 2023.

Biyang Guo, Xin Zhang, Ziyuan Wang, Minqi Jiang, Jinran Nie, Yuxuan Ding, Jianwei Yue, and Yupeng Wu. How close is chatgpt to human experts? comparison corpus, evaluation, and detection. *arXiv preprint arXiv:2301.07597*, 2023.

Abhimanyu Hans, Avi Schwarzschild, Valeriia Cherepanova, Hamid Kazemi, Aniruddha Saha, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Spotting llms with binoculars: Zero-shot detection of machine-generated text. In *International Conference on Machine Learning*, pp. 17519–17537. PMLR, 2024.

Jason Hong. The state of phishing attacks. *Communications of the ACM*, 55(1):74–81, 2012.

Abe Hou, Jingyu Zhang, Tianxing He, Yichen Wang, Yung-Sung Chuang, Hongwei Wang, Lingfeng Shen, Benjamin Van Durme, Daniel Khashabi, and Yulia Tsvetkov. Semstamp: A semantic watermark with paraphrastic robustness for text generation. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 4067–4082, 2024.

Xiaomeng Hu, Pin-Yu Chen, and Tsung-Yi Ho. Radar: Robust ai-text detection via adversarial learning. *Advances in neural information processing systems*, 36:15077–15095, 2023.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. A watermark for large language models. In *International Conference on Machine Learning*, pp. 17061–17084. PMLR, 2023.

Scott Kirkpatrick, C Daniel Gelatt Jr, and Mario P Vecchi. Optimization by simulated annealing. *science*, 220(4598):671–680, 1983.

Kalpesh Krishna, Yixiao Song, Marzena Karpinska, John Wieting, and Mohit Iyyer. Paraphrasing evades detectors of ai-generated text, but retrieval is an effective defense. *Advances in Neural Information Processing Systems*, 36:27469–27500, 2023.

Linyang Li, Pengyu Wang, Ke Ren, Tianxiang Sun, and Xipeng Qiu. Origin tracing and detecting of llms. *arXiv preprint arXiv:2304.14072*, 2023.

Stephanie Lin, Jacob Hilton, and Owain Evans. Truthfulqa: Measuring how models mimic human falsehoods. In *Proceedings of the 60th annual meeting of the association for computational linguistics (volume 1: long papers)*, pp. 3214–3252, 2022.

Xiaoming Liu, Zhaohan Zhang, Yichen Wang, Hang Pu, Yu Lan, and Chao Shen. Coco: Coherence-enhanced machine-generated text detection under data limitation with contrastive learning. *arXiv preprint arXiv:2212.10341*, 2022.

Yepeng Liu and Yuheng Bu. Adaptive text watermark for large language models. In *Forty-first International Conference on Machine Learning*.

Zichang Liu, Aditya Desai, Fangshuo Liao, Weitao Wang, Victor Xie, Zhaozhuo Xu, Anastasios Kyrillidis, and Anshumali Shrivastava. Scissorhands: Exploiting the persistence of importance hypothesis for llm kv cache compression at test time. *Advances in Neural Information Processing Systems*, 36:52342–52364, 2023.

Shixuan Ma and Quan Wang. Zero-shot detection of llm-generated text using token cohesiveness. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 17538–17553, 2024.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D Manning, and Chelsea Finn. Detectgpt: Zero-shot machine-generated text detection using probability curvature. In *International Conference on Machine Learning*, pp. 24950–24962. PMLR, 2023.

Hoang-Quoc Nguyen-Son, Minh-Son Dao, and Koji Zettsu. Simllm: Detecting sentences generated by large language models using similarity between the generation and its re-generation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 22340–22352, 2024.

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

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

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

Jinyan Su, Terry Yue Zhuo, Di Wang, and Preslav Nakov. Detectllm: Leveraging log rank information for zero-shot detection of machine-generated text. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Yuchuan Tian, Hanting Chen, Xutao Wang, Zheyuan Bai, QINGHUA ZHANG, Ruifeng Li, Chao Xu, and Yunhe Wang. Multiscale positive-unlabeled detection of ai-generated texts. In *The Twelfth International Conference on Learning Representations*, 2024.

Brian Tufts, Xuandong Zhao, and Lei Li. An examination of ai-generated text detectors across multiple domains and models. In *Neurips Safe Generative AI Workshop 2024*, 2024.

Eduard Tulchinskii, Kristian Kuznetsov, Laida Kushnareva, Daniil Cherniavskii, Sergey Nikolenko, Evgeny Burnaev, Serguei Barannikov, and Irina Piontkovskaya. Intrinsic dimension estimation for robust detection of ai-generated texts. *Advances in Neural Information Processing Systems*, 36, 2024.

Vivek Verma, Eve Fleisig, Nicholas Tomlin, and Dan Klein. 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)*, pp. 1702–1717, 2024.

Ivan Vykopal, Matúš Pikuliak, Ivan Srba, Robert Moro, Dominik Macko, and Mária Bieliková. Disinformation capabilities of large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14830–14847, 2024.

Pengyu Wang, Linyang Li, Ke Ren, Botian Jiang, Dong Zhang, and Xipeng Qiu. Seqxgpt: Sentence-level ai-generated text detection. In *The 2023 Conference on Empirical Methods in Natural Language Processing*, 2023.

Junchao Wu, Runzhe Zhan, Derek Wong, Shu Yang, Xinyi Yang, Yulin Yuan, and Lidia Chao. Detectrl: Benchmarking llm-generated text detection in real-world scenarios. *Advances in Neural Information Processing Systems*, 37:100369–100401, 2024.

Junxi Wu, Jinpeng Wang, Zheng Liu, Bin Chen, Dongjian Hu, Hao Wu, and Shu-Tao Xia. Moses: Uncertainty-aware ai-generated text detection via mixture of stylistics experts with conditional thresholds. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 5797–5816, 2025.

Kangxi Wu, Liang Pang, Huawei Shen, Xueqi Cheng, and Tat-Seng Chua. Llmdet: A large language models detection tool. *CoRR*, 2023a.

Yihan Wu, Zhengmian Hu, Hongyang Zhang, and Heng Huang. Dipmark: A stealthy, efficient and resilient watermark for large language models. 2023b.

Yang Xu, Yu Wang, Hao An, Zhichen Liu, and Yongyuan Li. Detecting subtle differences between human and model languages using spectrum of relative likelihood. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 10108–10121, 2024.

Yihuai Xu, Yongwei Wang, Yifei Bi, Huangsen Cao, Zhouhan Lin, Yu Zhao, and Fei Wu. Training-free llm-generated text detection by mining token probability sequences. In *The Thirteenth International Conference on Learning Representations*, 2025.

Xianjun Yang, Kexun Zhang, Haifeng Chen, Linda Petzold, William Yang Wang, and Wei Cheng. Zero-shot detection of machine-generated codes. *arXiv preprint arXiv:2310.05103*, 2023.

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

KiYoon Yoo, Wonhyuk Ahn, and Nojun Kwak. Advancing beyond identification: Multi-bit watermark for 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)*, pp. 4031–4055, 2024.

Xiao Yu, Yuang Qi, Kejiang Chen, Guoqiang Chen, Xi Yang, Pengyuan Zhu, Weiming Zhang, and Nenghai Yu. Gpt paternity test: Gpt generated text detection with gpt genetic inheritance. *CoRR*, 2023a.

Xiao Yu, Kejiang Chen, Qi Yang, Weiming Zhang, and Nenghai Yu. Text fluoroscopy: Detecting llm-generated text through intrinsic features. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 15838–15846, 2024.

Zhiyuan Yu, Yuhao Wu, Ning Zhang, Chenguang Wang, Yevgeniy Vorobeychik, and Chaowei Xiao. Codeipprompt: intellectual property infringement assessment of code language models. In *International conference on machine learning*, pp. 40373–40389. PMLR, 2023b.

Haolan Zhan, Xuanli He, Qiongkai Xu, Yuxiang Wu, and Pontus Stenetorp. G3detector: General gpt-generated text detector. *arXiv preprint arXiv:2305.12680*, 2023.

Ruisi Zhang, Shehzeen Samarah Hussain, Paarth Neekhara, and Farinaz Koushanfar. {REMARK-LLM}: A robust and efficient watermarking framework for generative large language models. In *33rd USENIX Security Symposium (USENIX Security 24)*, pp. 1813–1830, 2024.

APPENDIX

A More Discussion of the Proposed Method

A.1 Contribution

The contributions of this paper are multifaceted.

- **We provide a unified perspective for understanding metric-based detection methods**. The diversity of these methods (e.g., using metrics based on log-likelihood or log-rank, and introducing perturbed or regenerated text) makes comparison difficult. To this end, we re-examine these methods from three perspectives: data, score calculation, and detection. This analysis provides a precise definition for each method, facilitating comparison and potential improvements. Our analysis shows that these methods employ more reasonable evaluation metrics and incorporate additional contextual information to enhance detection. However, they all fail to address the underlying token-level errors caused by inherent randomness, which limits their detection potential. Therefore, our unified analysis encourages a more nuanced characterization of token-level detection scores, which will provide guidance for future improvements.
- **We reveal the relationships between contextual token detection scores**. While the generative mechanisms of LLMs introduce dependencies between tokens, these relationships remain unclear. To this end, we theoretically reveal two relationships between contextual token scores: neighboring similarity and initial instability, which are further validated through empirical experiments. Constraining these relationships during detection has the potential to mitigate the imprecision in score calculation caused by the inherent randomness of MGTs, which is crucial for the field of MGT detection.
- **We propose a Markov-informed score calibration method to enhance MGT detection**. This involves using Markov random fields to capture the revealed relationships and, through mean-field approximation, modeling the MRF model as a lightweight component that can be stacked on existing detectors to further unlock detection potential. It is worth noting that our main technical contribution and innovation lies not only in the specific implementation but also in the conceptual token-level score calibration. While the current implementation is based on Markov random fields, this is only one possible approach; alternatives include using sequence models or graph neural networks. This conceptual insight can inspire improving MGT detection.
- **Extensive experiments consistently demonstrate the enhanced effectiveness of the proposed strategy**. We empirically verify that it not only excels on a single task but also demonstrates strong capabilities in multiple complex and challenging real-world scenarios, including generalization across LLMs and domains, and robustness to mixed text and paraphrased text. Furthermore, the proposed enhancement component incurs negligible computational overhead compared to the original detector. This combination of effectiveness and efficiency provides a solid foundation for developing practically deployable enhancement solutions for AI-generated text detection.

A.2 Differences from Existing Methods

Although some works, such as FourierGPT Xu et al. (2024) and Lastde Xu et al. (2025), also learn local patterns, our approach is quite different:

- **Methodology**. Unlike FourierGPT or Lastde, which propose new standalone metrics, our method is a universal plug-in. We do not replace the metric; we calibrate the intermediate token-level scores of any existing metric (Likelihood, DetectGPT, etc.) using a Markov Random Field.
- **Theory vs. Empiricism**. While many existing local-pattern methods are empirically motivated (e.g., observing spectral power differences and token probability fluctuation), our approach is grounded in the theoretical derivation of attention dynamics (Theorem 1), which formally reveals the Neighbor Similarity and Initial Instability properties we model.

B Related Work

Existing detection methods can be categorized into active watermark-based methods and passive model-based and metric-based methods.

B.1 WATERMARK-BASED DETECTION

Watermarking is a proactive defense technique that embeds verifiable information during the text generation stage, thereby enabling simple and reliable detection. RedList (Kirchenbauer et al., 2023) is a model-agnostic watermarking method that dynamically partitions the vocabulary into a "green-list" and "redlist" based on preceding context, slightly increasing the probability of sampling tokens from the greenlist. Subsequent works have made various improvements to this approach. For instance, SemStamp (Hou et al., 2024) introduces a sentence-level semantic hashing watermark to enhance robustness against paraphrasing attacks; DiPmark (Wu et al., 2023b) designs an unbiased watermark that does not alter the original output distribution. REMARK-LLM (Zhang et al., 2024) is a training-based watermarking method that employs a message encoding module to generate an encrypted token distribution for watermark embedding prior to inference. Beyond manually designed watermarks, directly leveraging language models to learn to generate watermarked text is also promising, including training student models (Gu et al., 2023) and semantically invariant watermarking models (Liu & Bu). In addition to the standard binary (0/1 bit) detection of AI-generated content, researchers have also explored multi-bit watermarks (Yoo et al., 2024) for embedding more information.

B.2 MODEL-BASED DETECTION

Model-based methods represent a classical paradigm in detection, training a binary classifier on a dataset containing both human- and machine-generated texts. A series of works, such as OpenAI Detector (Solaiman et al., 2019), ChatGPT Detector (Guo et al., 2023), GPTZero (GPTZero, 2023), and G3 Detector (Zhan et al., 2023), collect texts generated by various LLMs to train a unified classifier. GPT-Pat (Yu et al., 2023a) finds that detectors trained solely on a single decoding strategy generalize poorly and enhances performance by utilizing mixed decoding strategies. In addition to original data, GLTR (Gehrmann et al., 2019) trains a simple logistic regression classifier by analyzing the predicted ranking of each word within its context. SeqXGPT (Wang et al., 2023) treats the sequence of logits as waveform signals for detection, while Sniffer (Li et al., 2023) uses the difference in logits from different models on the same text as features for detection and attribution. Beyond the data level, recent works have explored more advanced training strategies. For example, CoCo (Liu et al., 2022) introduces graph structures and contrastive learning; LLMDet (Wu et al., 2023a) leverages the perplexity of surrogate models as additional features; MPU (Tian et al., 2024) adopts a positive-unlabeled learning paradigm; and RADAR (Hu et al., 2023) incorporates adversarial training to enhance model robustness. The above methods generally assume a known text source, but when it is unknown, Ghostbuster (Verma et al., 2024) proposes training classifiers directly using texts generated from known surrogate models.

B.3 METRIC-BASED DETECTION

Metric-based methods do not require training on specific datasets; instead, they directly leverage the inherent statistical biases or intrinsic properties of language model-generated text for distinction. The main advantage of such methods lies in their stronger generalization to new models and domains. Classic approaches in this category include the use of Log-Likelihood (Solaiman et al., 2019), Log-Rank (Mitchell et al., 2023), and Entropy (Gehrmann et al., 2019). DetectGPT (Mitchell et al., 2023) finds that AI-generated text typically lies in regions of negative curvature with respect to the model's log-probability function. By perturbing the text and observing changes in log-probability, it can effectively distinguish AI-generated text. Inspired by DetectGPT, Fast-DetectGPT (Bao et al., 2024) replaces log-probability with conditional probability curvature, significantly improving detection efficiency while maintaining performance. DetectLLM-LRR (Su et al.) proposes using the ratio of log-likelihood to log-rank for detection. Some works, such as DNA-GPT (Yang et al., 2024) and DetectGPT4Code (Yang et al., 2023), detect AI-generated text by comparing discrepancies between the original text and continuations generated by a surrogate model. PHD (Tulchinskii et al., 2024) observes that genuine human-written text possesses higher intrinsic dimensionality after encoder mapping. SimLLM (Nguyen-Son et al., 2024) is based on the observation that the similarity between the original text and its generated continuation is significantly higher than that between generated text and its re-generated version; thus, it estimates the similarity between an input sentence and its generated counterpart for detection. Given that existing methods struggle with out-of-distribution data, token coherence (Ma & Wang, 2024) has been shown to be a reliable metric since

LLM-generated text usually exhibits higher token coherence than human-written text. Yu et al. (Yu et al., 2024) capture the intrinsic features of text by identifying layers with the greatest distributional differences when projecting into the vocabulary space, and using intrinsic rather than semantic features for detection has been demonstrated to yield better results. RepreGuard (Chen et al., 2025) extracted unique activation features from the MGT using a surrogate model, and then the projection scores along the direction of these features are more discriminative. Lastde (Xu et al., 2025) introduced time series analysis to LLM-generated text detection, capturing the temporal dynamics of token probability sequences. FourierGPT (Xu et al., 2024) offered a new perspective on human and model text detection tasks by using relative rather than absolute probability values and by extracting useful features from the probability spectrum. MoSEs (Wu et al., 2025) achieved the quantification of uncertainty in style perception through conditional threshold estimation.

C Proof of Theorem 1

Theorem. *1 Let $\lambda_K, \lambda_Q, \lambda_V, \lambda_O$ be the largest singular values of parameters W_K, W_Q, W_V, W_O, respectively, and let $W = W_V W_O W_Q W_K^\top$. For the transformer defined in Eq. (1), assuming normalized inputs ($\|x_t\|_2 = 1$ for all t) and constants $c, \epsilon > 0$, consider $a_t x_{t+1}^\top \geq (1-\delta)\|a_t\|_2$ with $\delta \leq \left(\frac{c\epsilon}{\lambda_Q \lambda_K \lambda_V \lambda_O}\right)^2$. If x_ℓ satisfies $x_\ell W x_\ell^\top \geq c$ and $x_\ell W x_\ell \geq \epsilon^{-1} \max_{j\in[\ell], j\neq\ell} x_j W x_\ell^\top$, then*

$$\alpha_{t+1,l} \leq \frac{\exp\left(C_l \cdot \alpha_{t,l} + \eta\right)}{\exp\left(C_l \cdot \alpha_{t,l} + \eta\right) + \sum_{j\neq l} \exp\left(C_j \cdot \alpha_{t,j} - \eta\right)},$$

$$\alpha_{t+1,l} \geq \frac{\exp\left(C_l \cdot \alpha_{t,l} - \eta\right)}{\exp\left(C_l \cdot \alpha_{t,l} - \eta\right) + \sum_{j\neq l} \exp\left(C_j \cdot \alpha_{t,j} + \eta\right)},$$

where

$$C_j = \frac{x_j W x_j^T}{t\,|a_t|_2}, \text{and } \eta = \frac{(1+\sqrt{2})\epsilon x_j W x_j^T}{(t+1)\,|a_t|_2}.$$

Proof. Our proof follows the proof of existing work (Liu et al., 2023). First, we introduce two necessary lemmas to help our proof.

Lemma 2 ((Liu et al., 2023))**.** *Let $x_1, x_2 \in \mathbb{R}^{1\times m}$ satisfies $\|x_1\|_2 = \|x_2\|_2 = 1$ and $x_1 x_2^\top \geq 1-\delta$ for some $\delta \in (0,1)$. Then for all $y \in \mathbb{R}^{1\times m}$ we have*

$$\left|x_1 y^\top - x_2 y^\top\right| \leq \sqrt{2\delta}\|y\|_2$$

Lemma 3 ((Liu et al., 2023))**.** *Let $\ell \in [t]$ be given. Suppose that $x_\ell A x_\ell^\top > \epsilon^{-1}\left|x_j A x_\ell^\top\right|$ for all $j \neq \ell$. Then we have*

$$\left(\mathcal{S}(t)_\ell - \epsilon\right) x_\ell^\top a x_\ell \leq x_\ell^\top W_K^\top W_Q a_t \leq \left(\mathcal{S}(t)_\ell + \epsilon\right) x_\ell^\top a x_\ell$$

Based on these two lemmas, we can formally prove the theorem. Let $x_1 = \frac{a_t}{\|a_t\|}$, and $x_2 = x_{t+1}$. If $\frac{a_t x_{t+1}^\top}{\|a_t\|} \geq 1-\delta$, using the conclusion of Lemma 2, we have

$$\left|\frac{a_t}{(t+1)\|a_t\|_2} W_Q W_K^T x_\ell^T - \frac{1}{t+1} x_{t+1} W_Q W_K^T x_\ell^T\right| \leq \frac{\sqrt{2\delta}}{t+1}\left\|W_Q W_K^T x_\ell^T\right\|_2$$

Since λ_Q, λ_K are the maximum singular values, respectively. Then we have $\left\|W_Q W_K^\top x_\ell^\top\right\|_2 \leq \lambda_Q\lambda_K\|x_\ell\|_2 = \lambda_Q\lambda_K$. This leads to:

$$\left|\frac{a_t}{(t+1)\|a_t\|_2} W_Q W_K^T x_\ell^T - \frac{1}{t+1} x_{t+1} W_Q W_K^T x_\ell^T\right| \leq \frac{\sqrt{2\delta}}{t+1}\lambda_Q\lambda_K \tag{10}$$

Since

$$\|a_t\|_2 = \left\|\left(\sum_{j=1}^{t-1}\alpha_{t,j}x_j\right) W_V W_O\right\| \leq \lambda_O\lambda_V\left\|\sum_{j=1}^{t-1}\alpha_{t,j}x_j\right\|_2 \leq \lambda_O\lambda_V\sum_{j=1}^{t-1}\alpha_{t,j}\|x_j\|_2 = \lambda_O\lambda_V,$$

and the theorem assumes $\delta \leq \left(\frac{c\epsilon}{\lambda_Q \lambda_K \lambda_V \lambda_O}\right)^2$, substituting these into Eq. (10), we can obtain:

$$\frac{\sqrt{2\delta}}{t+1}\lambda_Q\lambda_K \leq \frac{\sqrt{2}c\epsilon}{(t+1)\lambda_V\lambda_O} \leq \frac{\sqrt{2}c\epsilon}{(t+1)\left\|a_t\right\|_2} \leq \frac{\sqrt{2}\epsilon}{(t+1)\left\|a_t\right\|_2} x_\ell a x_\ell^T \tag{11}$$

The last inequality is obtained from $x_\ell a x_\ell^T \geq c$. Then combining Formula (10) and Formula (11), we have

$$\left| \frac{a_t}{(t+1)\left\|a_t\right\|_2} W_Q W_K^T x_\ell^T - \frac{1}{t+1} x_{t+1} W_Q W_K^T x_\ell^T \right| \leq \frac{\sqrt{2}\epsilon}{(t+1)\left\|a_t\right\|_2} x_\ell a x_\ell^T$$

From Lemma 3, we have:

$$\left| \frac{a_t W_Q W_K^\top x_\ell^\top}{(t+1)\left\|a_t\right\|^2} - \frac{\alpha_{t,\ell} x_\ell W x_\ell^\top}{(t+1)\left\|a_t\right\|^2} \right| \leq \frac{\epsilon}{(t+1)\left\|a_t\right\|^2} x_\ell^\top a x_\ell$$

By the triangle inequality, we can combine the upper bounds:

$$\begin{aligned} \left| \frac{x_{t+1} W_Q W_K^T x_\ell^T}{t+1} - \frac{\alpha_{t,\ell} x_\ell W x_\ell^T}{(t+1)\left\|a_t\right\|_2} \right| &\leq \left| \frac{x_{t+1} W_Q W_K^T x_\ell^T}{t+1} - \frac{a_t W_Q W_K^T x_\ell^T}{(t+1)\left\|a_t\right\|_2} \right| \\ &\quad + \left| \frac{a_t W_Q W_K^T x_\ell^T}{(t+1)\left\|a_t\right\|_2} - \frac{\alpha_{t,\ell} x_\ell W x_\ell^T}{(t+1)\left\|a_t\right\|_2} \right| \\ &\leq \frac{(1+\sqrt{2})\epsilon}{(t+1)\left\|a_t\right\|^2} x_\ell^\top a x_\ell \end{aligned} \tag{12}$$

Then, we rearrange the inequality, and we can obtain:

$$\frac{x_\ell W x_\ell^\top}{(t+1)\left\|a_t\right\|_2}\left(\alpha_{t,\ell} - (1+\sqrt{2})\epsilon\right) \leq \frac{1}{t+1} x_{t+1} W_Q W_K^\top x_\ell^\top \leq \frac{x_\ell W x_\ell^\top}{(t+1)\left\|a_t\right\|_2}\left(\alpha_{t,\ell} + (1+\sqrt{2}\epsilon\right)$$

Now we give the lower and upper bounds of $\alpha_{t+1,\ell} = \text{softmax}(1/(t+1) \cdot x_{t+1} W_Q W_K^\top X_t^\top)_l$.
Upper bound. Let $\gamma_{t+1,\ell} = 1/(t+1) \cdot x_{t+1} W_Q W_K^\top x_\ell^\top$. For the softmax function $\alpha_{t+1,l} = \frac{\exp(\gamma_{t+1,\ell})}{\exp(\gamma_{t+1,\ell}) + \sum_{k \neq \ell} \exp(\gamma_{t+1,k})}$, to get the maximum value of $\alpha_{t+1,\ell}$, we need to (1) make the numerator as big as possible, which means $\gamma_{t+1,\ell}$ should take its maximum value $\frac{x_\ell W x_\ell^\top}{(t+1)\|a_t\|_2}\left(\alpha_{t,\ell} + (1+\sqrt{2})\epsilon\right)$, (2) make the denominator as small as possible, which means that all other values $\gamma_{t+1,k}$ (when $k \neq \ell$) should be minimized $\frac{x_k W x_k^\top}{(t+1)\|a_t\|_2}\left(\alpha_{t,k} - (1+\sqrt{2})\epsilon\right)$. Therefore,

$$\alpha_{t+1,l} \leq \frac{\frac{x_\ell W x_\ell^\top}{(t+1)\|a_t\|_2}\left(\alpha_{t,\ell} + (1+\sqrt{2})\epsilon\right)}{\frac{x_\ell W x_\ell^\top}{(t+1)\|a_t\|_2}\left(\alpha_{t,\ell} + (1+\sqrt{2})\epsilon\right) + \sum_{k \neq \ell} \frac{x_k W x_k^\top}{(t+1)\|a_t\|_2}\left(\alpha_{t,k} - (1+\sqrt{2})\epsilon\right)}$$

Lower bound. Similarly, for the lower bound, we should (1) make the numerator as small as possible, which means $\gamma_{t+1,\ell}$ should take its minimum value $\frac{x_\ell W x_\ell^\top}{(t+1)\|a_t\|_2}\left(\alpha_{t,\ell} - (1+\sqrt{2})\epsilon\right)$, (2) make the denominator as large as possible, which means that all other values $\gamma_{t+1,k}$ (when $k \neq \ell$) should be maximized $\frac{x_k W x_k^\top}{(t+1)\|a_t\|_2}\left(\alpha_{t,k} + (1+\sqrt{2})\epsilon\right)$. Therefore,

$$\alpha_{t+1,l} \geq \frac{\frac{x_\ell W x_\ell^\top}{(t+1)\|a_t\|_2}\left(\alpha_{t,\ell} - (1+\sqrt{2})\epsilon\right)}{\frac{x_\ell W x_\ell^\top}{(t+1)\|a_t\|_2}\left(\alpha_{t,\ell} - (1+\sqrt{2})\epsilon\right) + \sum_{k \neq \ell} \frac{x_k W x_k^\top}{(t+1)\|a_t\|_2}\left(\alpha_{t,k} + (1+\sqrt{2})\epsilon\right)}$$

The proof is completed. □

D EXPERIMENTAL DETAILS

D.1 DATASETS

The details of the dataset used in the paper are as follows:

- **Essay** (Verma et al., 2024). Each source of this dataset (human-written texts, various LLM-generated texts) contains 1,000 samples. The HGT portion comprises original IvyPanda essays that cover a wide array of subjects and academic levels, from high school through university. For the MGT portion, a tailored prompt was first crafted for each source essay using ChatGPT-turbo, and that prompt was then submitted to several LLMs, including GPT4All, ChatGPT, ChatGPT-turbo, ChatGLM, Dolly, and Claude, to generate machine-written essays. This workflow produced a diverse set of model-generated texts that remained aligned with the topics of their corresponding source documents.
- **Reuters** (Verma et al., 2024). Built on the Reuters 50–50 authorship benchmark, this dataset contains 1,000 articles from 50 writers, with each author contributing 20 pieces. Replicating the pipeline used for the essay corpus, the team first asked ChatGPT-turbo to invent a headline for every article. Those auto-generated headlines were then embedded into prompts and submitted to multiple LLMs, including ChatGPT, GPT-4, ChatGPT-turbo, ChatGLM, Dolly, and Claude, to create the machine-generated texts.
- **TruthfulQA** (Lin et al., 2022). It contains 817 questions covering 38 categories, including health, law, finance, and politics. The generated answers were produced by several large language models, including GPT4, ChatGPT-turbo, ChatGLM, Dolly, ChatGPT, and StableLM.
- **DetectRL** (Wu et al., 2024). In this dataset, the human-authored portion is drawn from four sources: arXiv abstracts dated 2002–2017, XSum news reports, Writing Prompts stories, and Yelp reviews. These types were chosen because they are especially vulnerable to producing convincing but misleading content when LLMs are misapplied. From each source, 2,800 human texts are selected as HGTs. The machine-generated texts are created using four widely used LLMs—GPT-3.5-turbo (ChatGPT), PaLM-2-bison (Google-PaLM), and Llama-2-70b. The dataset further models practical adversarial settings: (1) a paraphrasing attack that rewrites MGTs with the Dipper paraphraser (Krishna et al., 2023) and Polish paraphraser, and (2) a mixed-text condition where 1/4 of machine-generated sentences is randomly replaced with human-written content while the label remains "machine-generated."

D.2 BASELINES

A detailed description of the baselines used is shown below:

- **Likelihood** (Solaiman et al., 2019). It uses an LLM to calculate the log probability of each token in a text. The average of these probabilities gives a detection score. A higher score indicates a greater chance that the text was generated by LLMs.
- **Log-Rank** (Gehrmann et al., 2019). Its detection score is created by first using an LLM to rank each token in a text based on its predicted order within a given context. The logarithm of each word's predicted rank is then calculated. The final score is an average of these values, and a lower score is a strong indicator of machine-generated text.
- **Entropy** (Gehrmann et al., 2019). Similar to Log-Rank, it calculates a score for a text by taking the average of each token's conditional entropy within its given context. A lower score suggests a higher likelihood that the text was generated by LLMs.
- **DetectGPT** (Mitchell et al., 2023). It determines if a text is machine-generated by measuring how small changes affect its log probability. The underlying idea is that text created by LLMs is already a high-probability output. So, when it is slightly altered, the new version is likely to have a lower log probability. In contrast, making similar small changes to human-written text does not consistently lower the log probability; it can just as easily stay the same or increase.
- **Fast-DetectGPT (FastGPT)** (Bao et al., 2024). To overcome the major computational expense of DetectGPT, this approach replaces DetectGPT's resource-intensive perturbation step with a more efficient sampling process. It identifies differences in token selection between humans and LLMs using a conditional probability curvature metric.

- **DNA-GPT** (Yang et al., 2024). This method involves a two-step process. First, it cuts a text in half and uses the first part to prompt an LLM to generate a new continuation. Next, it examines the differences between the newly created segment and the original one. This comparison, done via N-gram analysis for black-box models or probability divergence for white-box models, reveals a clear distinction between how humans and machines generate text.
- **Repreguard** (Chen et al., 2025). It utilizes key feature directions determined by a proxy model to project and score the representation of the text under test, comparing the result against a threshold. This approach demonstrates extremely high detection accuracy and robustness when dealing with out-of-distribution data and various types of attacks.
- **Lastde** (Xu et al., 2025). It introduces time series analysis into machine-generated text detection, overcoming the limitations of traditional methods that ignore local discriminative information by collaboratively modeling the local dynamic features and global statistical indicators of token probability sequences.
- **FourierGPT** (Xu et al., 2024). It proposes a detection method based on a likelihood spectrum perspective, which captures subtle differences between human and machine language by analyzing the relative changes in text likelihood values rather than their absolute values.
- **Binoculars** Hans et al. (2024). It is a detection algorithm that requires no training data, accurately distinguishing between human and machine-generated text by comparing the score differences of a pair of pre-trained LLMs.

D.3 Experimental Scenario

To extensively evaluate the effectiveness of the proposed enhancement model, we conduct experiments in the following real-world scenarios:

- **Cross-LLM**. To assess how well the proposed model works across different LLMs, we trained detectors on a single LLM's text and then tested it on a variety of LLMS' texts. The main body of the paper presents the results from training detectors on the GPT4All texts (Essay dataset) and Llama-2-70b texts (DetectRL), and then testing them on various LLMs, as shown in Table 2. Complete results for every training and testing combination can be found in Appendix E.3 (Tables 3-15).
- **Cross-Domain**. The DetectRL dataset includes texts from four distinct domains: arXiv academic abstracts, XSum news articles, Writing Prompts stories, and Yelp Reviews. We utilized this dataset to evaluate the model's performance across various domains. To achieve this, we trained the detector on one domain and then tested it on the other domains. For this evaluation, all machine-generated texts were created using the default PaLM model. The heatmaps in Figs. 4, 21, and 22 illustrate the performance improvement achieved by our enhancement model compared to various baseline detection models.
- **Paraphrasing Attack**. Studies have shown that MGT detection is vulnerable to paraphrase attacks. Therefore, this scenario is used to evaluate the robustness of the MGT detection method. Using the DetectRL dataset, which includes data from the Polish and Dipper paraphraser, we trained our detector on clean, original texts and then evaluated its robustness on these paraphrased texts. Specifically, we trained the detector using clean texts from Llama-2-70b and then tested it on paraphrased texts from several different LLMs. The results can be found in Fig. 6 and 24.
- **Mixed Text**. Because a blend of human and machine-generated text is common in the real world, we use the mixed texts provided by DetectRL for evaluation. It involved randomly swapping out 25% of the sentences in an LLM-generated text with human-written ones. We conducted two separate experiments on this dataset: (1) The detector was trained on pure, non-mixed text and then tested for its ability to detect the mixed texts. (2) The detector was both trained and tested on the mixed texts themselves. The performance of the detector in these mixed settings is shown in Figs. 5 and 23. In each sub-figure, the detectors trained on original text are shown on the left, and those trained on mixed text are shown on the right.

D.4 Threat Model and Proxy Settings

We conducted the experiments in a black-box setting:

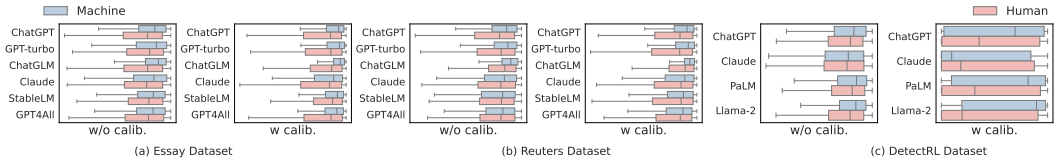

Figure 9: Distribution of token scores obtained by the Log-Likelihood method without and with enhancement. It can be observed that the proposed method enhances the discriminative nature of the token scores.

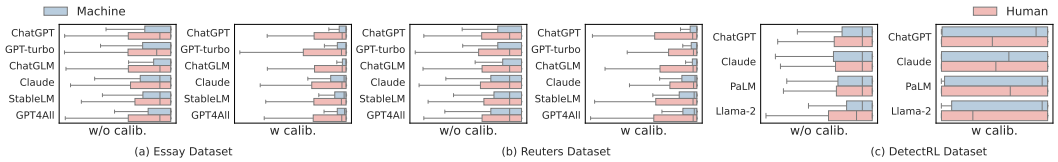

Figure 10: Distribution of token scores obtained by the Log-Rank method without and with enhancement. It can be observed that the proposed method enhances the discriminative nature of the token scores.

- **Threat Model**. We operate under a strictly Black-Box setting regarding the target LLM. We assume that the detector is entirely unknown to the LLM that generates the text (e.g., ChatGPT, Claude). Therefore, we employ a Proxy Model to compute token-level metrics (e.g., Log-Likelihood and Rank) for the candidate text, which are reasonable due to the transferability of the extracted metrics.

- **Proxy Models**. For all baselines, we use GPT2 as the proxy model. Additionally, we use GPT2-XL as the scoring model for Fast-DetectGPT. To learn the 2x2 parameter W_{mrf} introduced by our strategy, we perform training based on various LLM texts, as explicitly stated in the "Training Text" of the table. For example, in Table 2 in the paper, we learn W_{mrf} on GPT4All texts of the Essay dataset and Llama-2-70b texts of the DetectRL dataset, and then test all candidate texts (from ChatGPT, Claude, or Dolly, etc.) using the learned detector.

D.5 Parameter settings

We conducted five independent experiments to ensure the consistency of our results, using five fixed random seeds (1-5). For all datasets, we used 10% of the data for training, while the remaining 90% was split evenly between validation and testing. To ensure a fair comparison, the enhanced models shared the same hyperparameters as their base models. In our enhancement model, there are two hyperparameters: the transition center t_0 in the positional weighting function $\beta(t)$ and the number of iterations T in the MRF layer. By default, we set $t_0 = 30$ and $T = 10$ for the enhanced versions of all detectors across the three datasets, highlighting the flexibility of our approach. For training the MRF layer, we use a learning rate of 0.05 and train for 10 epochs. Hyperparameter sensitivity analyses are provided in Appendix E.9.

E More Experimental Resuls

E.1 More Results of Token Score Distribution before and after Enhancement

In addition to the partial results on DetectGPT presented in the main text, we also present the complete results about token-level detection score distributions for Log-Likelihood, Log-Rank, Entropy, and DetectGPT in Figs. 9, 10, 11, and 12. The results are similar to those in the main text: the original detector's scores show substantial overlap between human- and machine-generated text. However, after calibration using our proposed augmentation strategy, the scores achieve significantly improved discriminability.

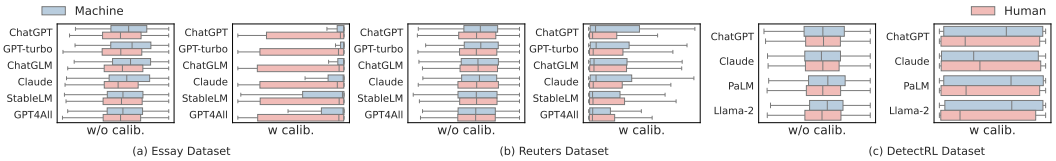

Figure 11: Distribution of token scores obtained by the Entropy method without and with enhancement. It can be observed that the proposed method enhances the discriminative nature of the token scores.

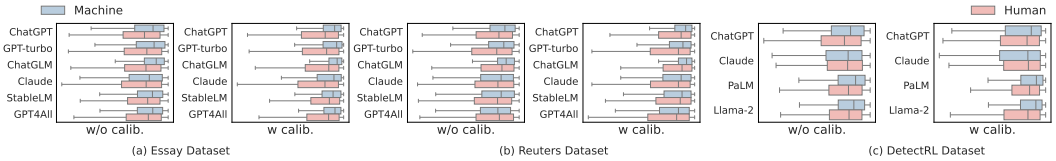

Figure 12: Distribution of token scores obtained by the DetectGPT method without and with enhancement. It can be observed that the proposed method enhances the discriminative nature of the token scores.

E.2 More Results of Context Token Relationships

In the main text, we demonstrated the existence of neighbor similarity (Fig. 2) and initial instability (Fig. 2) in token-level detection scores through experiments on the Essay dataset. Here, we provide additional supplementary results to further validate these relationships.

To verify neighbor similarity, we provide additional results using Entropy and DetectGPT detection scores on the Essay dataset (Fig. 13), as well as results on the Reuters and DetectRL datasets (Figs. 14 and 15). In addition, we provide Chinese texts on the CUDRT dataset (Fig. 16). These supplementary experiments consistently show that the closer the tokens are, the more similar their detection scores are.

Similarly, to verify initial instability, we provide additional results on the Essay dataset (Fig. 17), as well as results on the Reuters and DetectRL datasets (Figs. 18 and 19). In addition, we provide Chinese texts on the CUDRT dataset (Fig. 20). These results again demonstrate that token scores at the beginning of a text fluctuate significantly before gradually stabilizing.

E.3 More Performance Comparison

Performance across Different LLMs. In the main text, we evaluated the cross-LLM performance of detectors trained on GPT4All (Essay) and Llama-2-70b (DetectRL) on various LLM texts in terms of AUROC, as shown in Table 2. This section aims to provide more comprehensive supplementary experimental results, including: (1) cross-LLM performance performance under the same experimental settings in terms of TPR@FPR-1% (Table 3), and (2) cross-LLM performance comparisons of detectors trained on other LLMs on the Essay, DetectRL, and Reuters datasets (Tables 4 to 15). These extensive experimental results are consistent with the conclusions of the main paper. Among the 888 cross-LLM evaluation settings, our proposed enhanced model achieved better performance than the original detector in 91.4% of the cases, highlighting the generalization ability and application value of this method on different models and datasets.

Performance across Different Domains. In addition to the cross-domain performance improvements for Log-Likelihood and Log-Rank demonstrated in the main text, this section provides additional results for other detectors. Specifically, it includes improvements to the cross-domain performance of Entropy and DetectGPT (Fig. 21), as well as improvements to FastGPT and DNA-GPT (Fig. 22). Combining all experimental results, we reach the same conclusion as in the main text: in most experimental settings, detectors applied with our strategy significantly improve their cross-domain generalization capabilities.

Performance against Mixed Texts. In addition to the AUROC performance comparison for mixed texts presented in the main text, this section provides additional performance comparisons

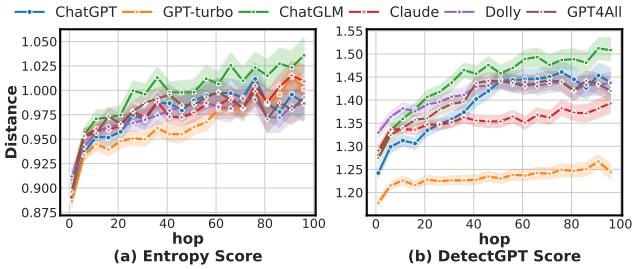

Figure 13: The detection score distances (Mean Absolute Difference) of neighbors at different hops in the Essay dataset. Entropy and DetectGPT score are used here.

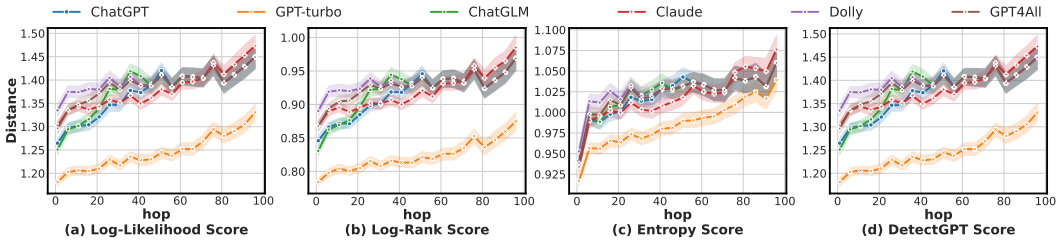

Figure 14: The detection score distances (Mean Absolute Difference) of neighbors at different hops in the Reuters dataset. Log-Likelihood, Log-Rank, Entropy, and DetectGPT score are used here.

at TPR@FPR-1%, as shown in Fig. 23. We reach consistent conclusions with those in the main text: in most experimental settings, the detector equipped with the proposed strategy significantly improves its ability to detect mixed text.

Performance against Paraphrasing Attacks. In addition to the AUROC performance comparison for paraphrasing texts presented in the main text, this section also presents a performance comparison under the TPR@FPR-1% metric, as shown in Fig. 24. We find that in most experimental settings, the detector applying our proposed strategy significantly improves its robustness against paraphrasing attacks, which is consistent with the conclusions drawn in the main text.

E.4 More Performance Comparison on the QA Dataset

In this section, we explore testing on a challenging QA dataset, specifically the TruthfulQA dataset (Lin et al., 2022), which covers Q&A tasks in 38 domains (e.g., health, law, and finance). The results are shown in Table 16. As can be seen, our method consistently boosts performance. For example, DNA-GPT-M achieves a significant leap from 80.17% to 88.32% (+8.15%) in average AUROC.

E.5 More Performance Comparison on Short-text

In this section, we performed a stress test on the Essay dataset with a strict 50-word limit. In Table 17, our calibration remains highly effective even with limited context. For example, we improve FastGPT by 5.35% and DNA-GPT by 3.03% on average. This confirms our calibration still holds even in short sequences.

E.6 Enhancement Experiments on More Detectors

In this section, we integrated more state-of-the-art metric-based baselines: RepreGuard Chen et al. (2025), Binoculars Hans et al. (2024), Lastde Xu et al. (2025), and FourierGPT Xu et al. (2024), and applied our strategy to them as "E" suffix. As shown in Tables 18-20, our method functions as a universal plug-in, enhancing these detectors. For example, on the Reuters dataset, our method improves RepreGuard by 8.97% and Binoculars by 4.42%.

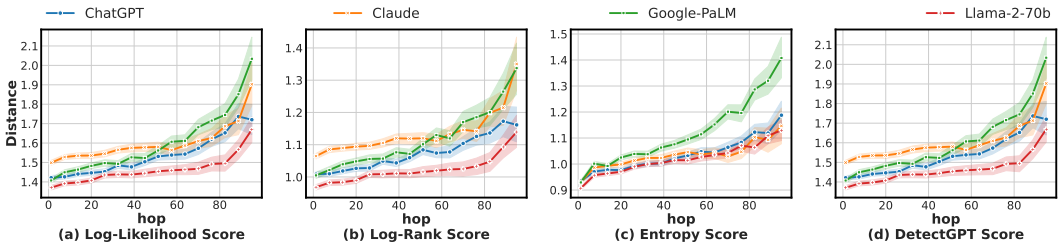

Figure 15: The detection score distances (Mean Absolute Difference) of neighbors at different hops in the DetectRL dataset. Log-Likelihood, Log-Rank, Entropy, and DetectGPT score are used here.

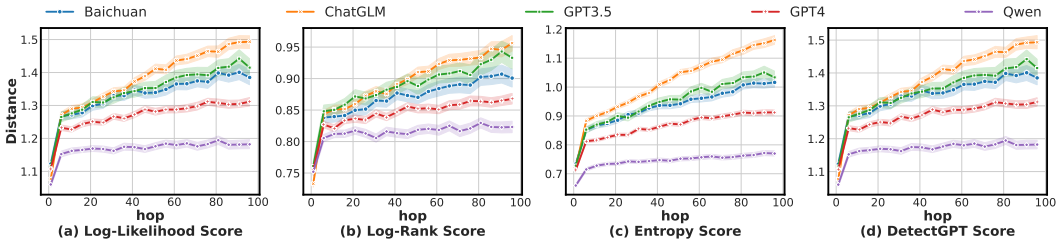

Figure 16: The detection score distances (Mean Absolute Difference) of neighbors at different hops in the CUDRT dataset. Log-Likelihood, Log-Rank, Entropy, and DetectGPT score are used here.

E.7 Performance Comparison against Perturbation Texts

In this section, we evaluated the robustness against perturbed texts. Here, we chose two distinct attack methods in the DetectRL dataset: Back-Translation and Word Perturbation. The results are shown in Tables 21. Across both perturbation scenarios, detectors equipped with our Markov-informed calibration (e.g., Likelihood-M, Log-Rank-M) consistently outperformed the uncalibrated ones, typically by margins of 10-20%. These findings align perfectly with our existing evaluation on Paraphrasing Attacks, which shows that our method effectively mitigates the performance degradation typically caused by text perturbations.

These encouraging gains are consistent with the fundamental theory of Markov random fields. The adversarial perturbation acts as token-level salt-and-pepper noise, while our Markov-informed calibration strategy (modeling Neighbor Similarity) leverages the remaining uncorrupted context to smooth out these local anomalies, recovering the latent detection signal.

E.8 More Results of Ablation Study

In addition to the ablation experiments on the position weight function and MRF layer presented in the main text for the Essay dataset, we also provide ablation results for the Reuters and DetectRL datasets, as shown in Figs. 25 and 26. These experimental results are consistent with the conclusions of the main text: removing either component leads to a significant performance drop; even retaining only one of them outperforms the baseline detector, strongly demonstrating the effectiveness of both components.

E.9 Sensitivity Analysis

Sensitivity w.r.t. transition center t_0. In our experiments, we set the default transition center t_0 of the position weighting function to 30. This section examines the effect of varying t_0 values on detection performance. The AUROC and TPR@FPR-1% results are shown in Figs. 27 and 28, respectively. The experimental results show that detection performance steadily improves with increasing t_0 values, which is consistent with the conclusions of the ablation experiments and demonstrates the effectiveness of the position weighting function. However, performance improvement is not infinite. When t_0 values are too large, performance gradually saturates or even declines. This is likely because excessively large t_0 filters out useful token scores. Therefore, a trade-off is necessary. Through sensitivity analysis, we recommend setting t_0 values between 20 and 30.

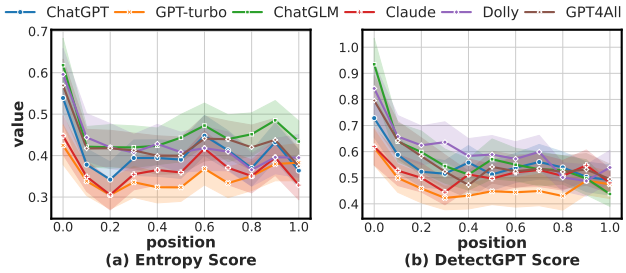

Figure 17: The score distances (Mean Absolute Difference) of 1-hop neighbors at different positions in the Essay dataset. Entropy and DetectGPT score are used here.

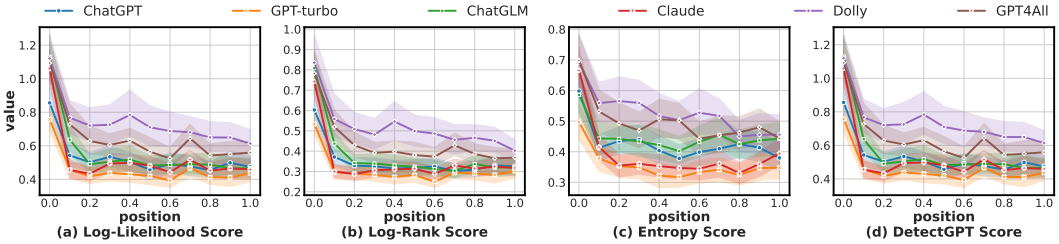

Figure 18: The score distances (Mean Absolute Difference) of 1-hop neighbors at different positions in the Reuters dataset. Log-Likelihood, Log-Rank, Entropy, and DetectGPT score are used here.

Sensitivity w.r.t. iterations T in MRF layer. We compute the posterior probability of the Markov random field using a multi-step iterative approach. To this end, we evaluate the impact of varying the number of iterations on detection performance, as shown in Figs. 29 and 30. The experimental results demonstrate that multi-step iterative computation significantly enhances detection performance compared to single-step computation, underscoring its importance in score calibration. However, performance may degrade with increasing the number of iterations, possibly due to oversmoothing of the detection scores. Based on our sensitivity analysis, we recommend setting the number of iterations to 10.

E.10 More Results Comparing with Neural Network Calibration

In the main text, we demonstrated, through experimental results using the Log-Likelihood score, that while neural network-based detection score correction performs well on the intra-LLM, its performance drops sharply on the cross-LLM. This section further presents comparative results using other detection scores, as shown in Figs. 31 and 32. Similar experimental findings further demonstrate that this method does not truly learn universal score correction capabilities, but rather overfits the training data. This strongly emphasizes the superiority and rationality of our proposed Markov-informed score calibration method.

E.11 MRF vs. HMM

In the paper, we chose the MRF over Markov chain for two key reasons:

- **Theoretical rationality**. A Markov chain typically models unidirectional dependencies (current state depends on previous state). However, in MGT detection, we identified the Neighbor Similarity property, which is inherently bidirectional. An MRF naturally captures these bidirectional dependencies, whereas a standard Markov Chain struggles to utilize future context during inference without complex multi-pass algorithms.
- **Computational Efficiency**. A raw MRF may be complicated. However, as described in Section 4.2, our Mean-Field Approximation transforms the problem into lightweight iterative matrix multiplications. This allows for massive parallelization on GPUs. In contrast, exact inference in HMMs (e.g., Forward-Backward algorithm) is inherently sequential, which can be slower on modern hardware despite having similar theoretical complexity $\mathcal{O}(N)$.

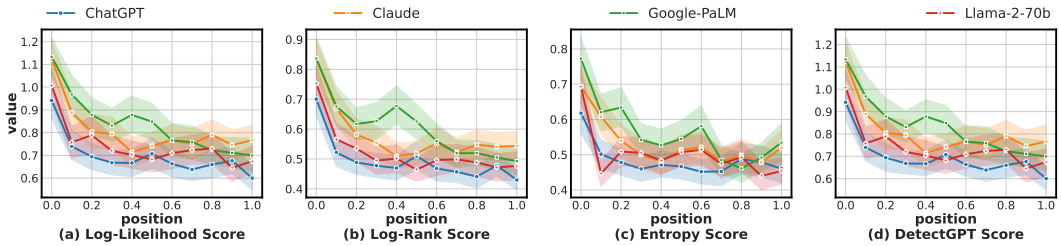

Figure 19: The score distances (Mean Absolute Difference) of 1-hop neighbors at different positions in the DetectRL dataset. Log-Likelihood, Log-Rank, Entropy, and DetectGPT score are used here.

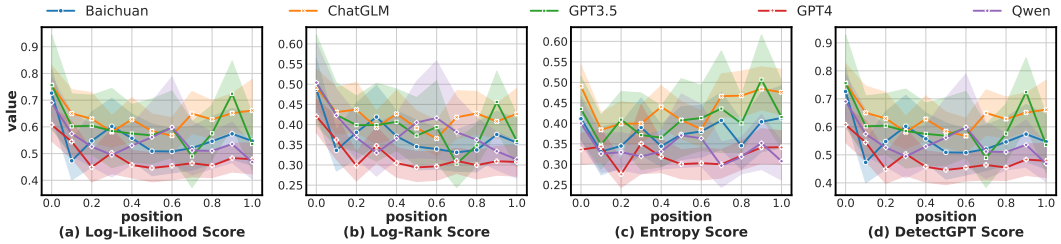

Figure 20: The score distances (Mean Absolute Difference) of 1-hop neighbors at different positions in the CUDRT dataset. Log-Likelihood, Log-Rank, Entropy, and DetectGPT score are used here.

To further verify, we implemented a Hidden Markov Model (HMM) based calibration strategy and applied it to existing detectors ("HMM" suffix). As shown in Table 22 (left part), while the HMM improves upon the original baseline, it consistently underperforms our MRF approach. This validates the necessity of modeling bidirectional dependencies. For computational efficiency (see the right part of Table 22), our MRF implementation is actually faster in practice due to GPU-friendly matrix operations.

E.12 Comparison with Enhanced Method

TOCSIN is an enhanced metric-based approach which enhances the detector by dynamically adjusting the decision threshold. Clearly, it operates on a different dimension than our method. Therefore, in addition to supplementing TOCSIN-enhanced baselines (suffix "-T"), our Markov calibration can be further applied to the top of TOCSIN (suffix "-TM") to further explore its detection potential, as shown in Table 23. It can be seen that our method further improves the performance of TOCSIN-enhanced baselines in most cases. This significant gain suggests that TOCSIN (adjusting dynamic thresholds using token cohesiveness score) and our method (calibration token-level detection score) focus on different aspects, achieving complementarity.

E.13 Analysis of Cross-domain Failure Cases

Aside from a few random performance fluctuations, most failures occur when xsum (training data) is applied to arXiv, writing, and yelp (test data). Given that our proposed calibration strategy hinges on learning a reward/penalty matrix (W_{mrf}) to encourage neighbor similarity, we examined the distance of neighbor detection score across various domains, as summarized in Fig. 33.

As observed, xsum exhibits considerably lower neighbor similarity (larger distance), whereas the other domains show higher similarity. This makes sense because, as a news summarization dataset, XSum is extremely information-dense, with a highly compressed syntactic structure and a lack of redundancy and transitional words seen in longer texts (such as arXiv). Consequently, adjacent tokens in XSum experience more drastic fluctuations in detection scores. Therefore, when trained on xsum, the MRF learns parameters (W_{mrf}) that tolerate high local variance (treating it as a normal feature of the domain). When these parameters are applied to "smoother" domains like arXiv, the MRF fails to enforce the tighter consistency constraints required for those domains, leading to the observed performance drop. Encouragingly, compared to these occasional failure modes, the extensive enhancements highlight the practicality of our method.

Table 3: Performance concerning TPR@FPR-1% (%) on Essay (left) and DetectRL (right).

Method	Essay (Training Text: GPT4All)							DetectRL (Training Text: Llama-2-70b)			
	GPT4All	ChatGPT	Dolly	ChatGLM	Claude	ChatGPT-turbo	Avg.	Llama-2-70b	ChatGPT	Google-PaLM	Avg.
Likelihood	$46.33_{\pm 16.49}$	$68.62_{\pm 13.32}$	$20.67_{\pm 10.79}$	$92.86_{\pm 4.84}$	$12.31_{\pm 6.29}$	$73.60_{\pm 14.63}$	52.4	$38.37_{\pm 0.92}$	$10.21_{\pm 1.40}$	$25.66_{\pm 2.41}$	24.75
Likelihood-M	**87.47**$_{\pm 3.42}$	**90.36**$_{\pm 2.93}$	**55.47**$_{\pm 5.03}$	**97.19**$_{\pm 1.11}$	**40.40**$_{\pm 7.87}$	**96.27**$_{\pm 1.27}$	**77.86**	**46.03**$_{\pm 1.70}$	**10.51**$_{\pm 3.92}$	**36.37**$_{\pm 2.31}$	**30.97**
Log-Rank	$62.69_{\pm 13.36}$	$79.07_{\pm 7.89}$	$25.11_{\pm 8.35}$	$95.71_{\pm 2.05}$	$19.20_{\pm 8.26}$	$80.89_{\pm 10.73}$	60.44	$42.05_{\pm 0.84}$	$12.44_{\pm 1.71}$	$22.84_{\pm 1.28}$	25.78
Log-Rank-M	**87.38**$_{\pm 2.50}$	**88.89**$_{\pm 2.84}$	**52.12**$_{\pm 4.18}$	**98.04**$_{\pm 0.43}$	**31.69**$_{\pm 4.15}$	**94.22**$_{\pm 2.46}$	**75.39**	**50.56**$_{\pm 1.44}$	**13.94**$_{\pm 2.36}$	**41.51**$_{\pm 2.69}$	**35.34**
Entropy	$2.73_{\pm 0.69}$	$7.16_{\pm 3.17}$	$2.29_{\pm 1.36}$	$6.88_{\pm 3.05}$	$3.91_{\pm 2.24}$	$13.24_{\pm 7.60}$	6.04	$2.03_{\pm 0.73}$	$0.25_{\pm 0.16}$	$6.95_{\pm 0.78}$	3.07
Entropy-M	**15.63**$_{\pm 0.87}$	**41.24**$_{\pm 2.25}$	**16.95**$_{\pm 3.12}$	**40.85**$_{\pm 4.37}$	**23.20**$_{\pm 1.70}$	**49.91**$_{\pm 5.02}$	**31.30**	**2.69**$_{\pm 0.39}$	**0.67**$_{\pm 0.10}$	**9.17**$_{\pm 1.10}$	**4.18**
DetectGPT	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.89_{\pm 0.24}$	$0.00_{\pm 0.00}$	0.15	$4.15_{\pm 0.40}$	$2.94_{\pm 0.62}$	$1.71_{\pm 0.47}$	2.93
DetectGPT-M	**41.23**$_{\pm 29.53}$	**46.98**$_{\pm 25.36}$	**18.38**$_{\pm 20.21}$	**41.83**$_{\pm 44.80}$	**8.80**$_{\pm 9.64}$	**65.87**$_{\pm 21.66}$	**37.18**	**32.41**$_{\pm 9.58}$	**7.59**$_{\pm 4.05}$	**25.14**$_{\pm 4.53}$	**21.71**
FastGPT	$1.59_{\pm 0.32}$	$1.64_{\pm 0.67}$	$0.29_{\pm 0.10}$	$2.72_{\pm 0.68}$	**3.47**$_{\pm 1.83}$	**16.89**$_{\pm 6.66}$	4.43	$11.35_{\pm 2.01}$	$3.78_{\pm 1.20}$	$5.02_{\pm 0.85}$	6.72
FastGPT-M	**23.92**$_{\pm 10.29}$	**30.00**$_{\pm 9.20}$	**32.89**$_{\pm 16.52}$	**55.54**$_{\pm 7.80}$	$0.67_{\pm 0.56}$	$16.44_{\pm 31.78}$	**26.58**	**15.23**$_{\pm 3.23}$	**6.28**$_{\pm 1.72}$	**13.52**$_{\pm 5.63}$	**11.68**
DNA-GPT	$57.68_{\pm 3.34}$	$63.24_{\pm 6.54}$	$22.20_{\pm 3.85}$	$84.82_{\pm 4.14}$	$16.67_{\pm 3.47}$	$56.27_{\pm 6.45}$	50.15	$36.56_{\pm 1.56}$	**19.70**$_{\pm 0.52}$	$30.38_{\pm 2.75}$	28.88
DNA-GPT-M	**93.85**$_{\pm 0.97}$	**88.44**$_{\pm 4.22}$	**56.42**$_{\pm 10.82}$	**96.83**$_{\pm 0.41}$	**33.33**$_{\pm 7.90}$	**86.22**$_{\pm 6.52}$	**75.85**	**42.52**$_{\pm 1.26}$	$19.11_{\pm 2.04}$	**34.81**$_{\pm 1.54}$	**32.15**

Table 4: Performance on Essay dataset. The detection models are trained on text generated by ChatGPT.

Metric	Method	GPT4All	ChatGPT	Dolly	ChatGLM	Claude	ChatGPT-turbo	avg
TPR@FPR-1%	Likelihood	$46.24_{\pm 16.60}$	$68.62_{\pm 13.32}$	$20.67_{\pm 10.79}$	$92.81_{\pm 4.82}$	$12.31_{\pm 6.29}$	$73.60_{\pm 14.63}$	52.38
	Likelihood-M	**89.38**$_{\pm 4.05}$	**92.00**$_{\pm 2.84}$	**59.19**$_{\pm 5.98}$	**97.10**$_{\pm 0.69}$	**42.22**$_{\pm 6.32}$	**96.80**$_{\pm 1.23}$	**79.45**
	Log-Rank	$62.69_{\pm 13.36}$	$79.07_{\pm 7.89}$	$25.11_{\pm 8.35}$	$95.71_{\pm 2.05}$	$19.16_{\pm 8.22}$	$80.89_{\pm 10.73}$	60.44
	Log-Rank-M	**87.15**$_{\pm 2.62}$	**88.80**$_{\pm 2.81}$	**52.12**$_{\pm 4.32}$	**98.04**$_{\pm 0.43}$	**31.33**$_{\pm 4.10}$	**94.22**$_{\pm 2.46}$	**75.28**
	Entropy	$2.73_{\pm 0.69}$	$7.16_{\pm 3.17}$	$2.29_{\pm 1.36}$	$6.88_{\pm 3.05}$	$3.91_{\pm 2.24}$	$13.24_{\pm 7.60}$	6.04
	Entropy-M	**12.76**$_{\pm 2.18}$	**34.84**$_{\pm 5.15}$	**14.27**$_{\pm 2.63}$	**36.96**$_{\pm 2.81}$	**19.78**$_{\pm 3.76}$	**43.20**$_{\pm 6.33}$	**26.97**
	DetectGPT	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.84_{\pm 0.51}$	$2.36_{\pm 2.52}$	0.53
	DetectGPT-M	**41.09**$_{\pm 30.67}$	**42.04**$_{\pm 25.95}$	**9.55**$_{\pm 9.01}$	**39.02**$_{\pm 38.26}$	**10.13**$_{\pm 10.28}$	**68.49**$_{\pm 23.15}$	**35.05**
	FastGPT	$1.59_{\pm 0.32}$	$1.64_{\pm 0.67}$	$0.29_{\pm 0.10}$	$2.72_{\pm 0.68}$	**3.47**$_{\pm 1.83}$	$16.89_{\pm 6.66}$	4.43
	FastGPT-M	**19.73**$_{\pm 11.16}$	**26.18**$_{\pm 12.42}$	**24.82**$_{\pm 20.23}$	**45.76**$_{\pm 21.65}$	$3.38_{\pm 5.34}$	**26.04**$_{\pm 32.58}$	**24.32**
	DNA-GPT	$40.77_{\pm 8.33}$	$73.69_{\pm 3.78}$	$19.95_{\pm 7.42}$	$82.59_{\pm 11.71}$	$18.04_{\pm 3.53}$	$63.82_{\pm 6.62}$	49.81
	DNA-GPT-M	**89.02**$_{\pm 4.14}$	**95.07**$_{\pm 1.33}$	**65.97**$_{\pm 4.57}$	**99.24**$_{\pm 0.36}$	**39.56**$_{\pm 4.01}$	**94.62**$_{\pm 1.43}$	**80.58**
AUROC	Likelihood	$96.16_{\pm 0.30}$	$98.79_{\pm 0.19}$	$90.90_{\pm 1.33}$	$99.29_{\pm 0.25}$	$92.76_{\pm 0.23}$	$99.13_{\pm 0.19}$	96.17
	Likelihood-M	**98.77**$_{\pm 0.10}$	**99.58**$_{\pm 0.14}$	**94.80**$_{\pm 0.87}$	**99.56**$_{\pm 0.19}$	**94.69**$_{\pm 0.42}$	**99.77**$_{\pm 0.14}$	**97.86**
	Log-Rank	$96.55_{\pm 0.31}$	$98.95_{\pm 0.13}$	$90.08_{\pm 1.28}$	$99.36_{\pm 0.13}$	$92.01_{\pm 0.20}$	$99.24_{\pm 0.15}$	96.03
	Log-Rank-M	**98.56**$_{\pm 0.06}$	**99.40**$_{\pm 0.09}$	**93.81**$_{\pm 0.92}$	**99.56**$_{\pm 0.08}$	**92.88**$_{\pm 0.29}$	**99.64**$_{\pm 0.10}$	**97.31**
	Entropy	$74.19_{\pm 1.62}$	$89.49_{\pm 0.33}$	$73.26_{\pm 1.48}$	$84.11_{\pm 0.77}$	$86.58_{\pm 0.66}$	$95.94_{\pm 0.35}$	83.93
	Entropy-M	**83.33**$_{\pm 0.66}$	**93.24**$_{\pm 0.15}$	**81.17**$_{\pm 1.04}$	**91.38**$_{\pm 0.37}$	**88.28**$_{\pm 0.49}$	**96.86**$_{\pm 0.21}$	**89.04**
	DetectGPT	$50.15_{\pm 0.99}$	$50.53_{\pm 3.65}$	$48.04_{\pm 7.27}$	$49.25_{\pm 1.58}$	$51.21_{\pm 8.39}$	$55.34_{\pm 31.67}$	50.75
	DetectGPT-M	**94.98**$_{\pm 3.26}$	**96.35**$_{\pm 2.60}$	**85.82**$_{\pm 6.13}$	**94.20**$_{\pm 4.48}$	**84.06**$_{\pm 6.47}$	**98.58**$_{\pm 1.24}$	**92.33**
	FastGPT	$64.63_{\pm 1.53}$	$67.68_{\pm 1.70}$	$47.17_{\pm 1.53}$	$71.08_{\pm 1.51}$	**75.31**$_{\pm 0.90}$	**88.62**$_{\pm 0.67}$	69.08
	FastGPT-M	**86.68**$_{\pm 3.28}$	**90.82**$_{\pm 3.86}$	**72.23**$_{\pm 23.91}$	**93.53**$_{\pm 3.86}$	$65.24_{\pm 12.36}$	$71.87_{\pm 21.80}$	**80.06**
	DNA-GPT	$96.28_{\pm 0.32}$	$98.87_{\pm 0.21}$	$92.85_{\pm 0.83}$	$99.32_{\pm 0.27}$	$91.65_{\pm 0.72}$	$98.51_{\pm 0.31}$	96.25
	DNA-GPT-M	**99.32**$_{\pm 0.13}$	**99.79**$_{\pm 0.07}$	**97.42**$_{\pm 0.56}$	**99.90**$_{\pm 0.06}$	**95.65**$_{\pm 0.51}$	**99.76**$_{\pm 0.07}$	**98.64**

E.14 Comparison with Model-based Detectors

As shown in Fig. 34, we compare the enhanced versions of metric-based methods with model-based detection methods, including ChatGPT-D and MPU. Experimental results show that while model-based methods demonstrate superior performance on the DetectRL dataset, they underperform state-of-the-art metric-based methods, such as DNA-GPT, on the Essay and Reuters datasets. Notably, the significant performance gap between model-based methods in intra-LLM and cross-LLM further confirms their increased risk of overfitting to the training data. This observation is consistent with our intention of focusing on metric-based detection methods.

E.15 Running Time

Table 24 shows the training and inference runtimes on different datasets. As discussed in the main text, our proposed Markov-based score refinement module can be implemented in constant time via sparse-dense matrix multiplication. Therefore, the additional time overhead introduced by this module is negligible compared to the time-consuming score calculation, highlighting the flexibility and practicality of our approach in practical applications.

F The Use of Large Language Models

In our paper, we used LLMs to polish the language and correct grammatical errors. LLMs were not used to generate novel research ideas, design experiments, analyze results, or write substantive

Table 5: Performance on Essay dataset. The detection models are trained on text generated by ChatGPT-turbo.

Metric	Method	GPT4All	ChatGPT	Dolly	ChatGLM	Claude	ChatGPT-turbo	avg
TPR@FPR-1%	Likelihood	46.29±16.55	68.62±13.32	20.67±10.79	92.86±4.84	12.31±6.29	73.60±14.63	52.39
	Likelihood-M	**87.61**±3.74	**88.31**±3.64	**61.29**±7.12	**94.20**±2.40	**41.11**±7.93	**95.38**±2.27	**77.98**
	Log-Rank	62.69±13.36	79.07±7.89	25.11±8.35	**95.71**±2.05	19.16±8.22	80.89±10.73	60.44
	Log-Rank-M	**86.70**±0.59	**87.47**±3.17	**54.61**±4.05	93.75±3.55	**35.91**±2.65	**94.13**±1.68	**75.43**
	Entropy	2.73±0.69	7.16±3.17	2.29±1.36	6.88±3.05	3.91±2.24	13.24±7.60	6.04
	Entropy-M	**11.62**±0.59	**30.67**±6.02	**13.56**±1.59	**32.54**±2.63	**17.33**±5.32	**39.91**±8.11	**24.27**
	DetectGPT	0.00±0.00	0.00±0.00	0.00±0.00	0.00±0.00	0.53±0.52	3.07±1.96	0.60
	DetectGPT-M	**14.03**±13.72	**17.07**±8.79	**3.05**±2.54	**6.43**±3.80	**5.11**±3.98	**44.71**±11.72	**15.07**
	FastGPT	1.59±0.32	1.64±0.67	0.29±0.10	2.72±0.68	3.47±1.83	16.89±6.66	4.43
	FastGPT-M	**8.97**±4.52	**15.33**±15.63	**0.72**±0.60	**20.09**±18.21	**7.51**±6.83	**49.82**±22.08	**17.07**
	DNA-GPT	0.73±0.33	0.76±0.30	2.63±0.81	0.13±0.11	21.64±3.45	0.00±0.00	4.31
	DNA-GPT-M	**39.41**±2.68	**38.62**±2.34	**39.47**±2.11	**38.04**±2.49	**39.38**±1.86	**38.67**±2.19	**38.93**
AUROC	Likelihood	96.16±0.30	98.79±0.19	90.90±1.33	**99.29**±0.25	92.76±0.23	99.13±0.19	96.17
	Likelihood-M	**98.65**±0.21	**99.39**±0.20	**93.95**±1.15	99.13±0.43	**94.66**±0.39	**99.77**±0.12	**97.59**
	Log-Rank	96.55±0.31	98.95±0.13	90.08±1.28	**99.36**±0.13	92.01±0.20	99.24±0.15	96.03
	Log-Rank-M	**98.37**±0.21	**99.17**±0.29	**92.96**±1.51	99.17±0.37	**92.96**±0.43	**99.62**±0.11	**97.04**
	Entropy	74.19±1.62	89.49±0.34	73.26±1.48	84.11±0.77	86.58±0.66	95.94±0.35	83.93
	Entropy-M	**82.88**±0.85	**93.08**±0.29	**80.90**±0.89	**91.18**±0.37	**88.57**±0.54	**96.90**±0.24	**88.92**
	DetectGPT	49.19±0.58	53.60±0.77	42.52±0.84	49.59±1.70	58.46±0.60	82.10±1.25	55.91
	DetectGPT-M	**91.65**±2.28	**93.38**±2.31	**77.58**±5.63	**90.54**±2.74	**79.82**±9.52	**97.61**±0.92	**88.43**
	FastGPT	64.63±1.53	67.68±1.70	47.17±1.53	71.08±1.51	75.31±0.90	88.62±0.67	69.08
	FastGPT-M	**85.05**±3.63	**86.41**±10.08	**49.85**±11.80	**89.08**±6.67	**75.84**±17.10	**96.80**±2.60	**80.51**
	DNA-GPT	53.28±1.23	43.65±0.76	61.90±0.39	40.46±1.26	67.53±0.55	43.31±0.94	51.69
	DNA-GPT-M	**83.94**±1.09	**84.02**±1.06	**83.93**±1.20	**83.89**±1.10	**84.23**±1.07	**84.07**±1.06	**84.01**

Table 6: Performance on Essay dataset. The detection models are trained on text generated by ChatGLM.

Metric	Method	GPT4All	ChatGPT	Dolly	ChatGLM	Claude	ChatGPT-turbo	avg
TPR@FPR-1%	Likelihood	46.29±16.54	68.62±13.32	20.67±10.79	92.81±4.82	12.31±6.29	73.60±14.63	52.38
	Likelihood-M	**78.82**±5.30	**87.87**±3.22	**46.73**±10.16	**98.71**±1.11	**31.64**±10.61	**93.78**±3.41	**72.92**
	Log-Rank	62.69±13.36	79.07±7.89	25.11±8.35	95.71±2.05	19.20±8.26	80.89±10.73	60.44
	Log-Rank-M	**81.32**±7.05	**86.76**±3.68	**47.35**±8.02	**98.62**±0.26	**25.24**±7.46	**90.62**±5.22	**71.65**
	Entropy	2.73±0.69	7.16±3.17	2.29±1.36	6.88±3.05	3.91±2.24	13.24±7.60	6.04
	Entropy-M	**12.76**±1.21	**32.09**±4.82	**13.99**±3.01	**35.62**±4.65	**18.40**±4.10	**41.42**±6.98	**25.71**
	DetectGPT	0.00±0.00	0.00±0.00	0.00±0.00	0.00±0.00	0.89±0.24	0.00±0.00	0.15
	DetectGPT-M	**18.36**±0.94	**17.87**±5.87	**20.67**±3.81	**60.98**±6.42	**4.89**±2.75	**6.36**±2.01	**21.52**
	FastGPT	1.59±0.32	1.64±0.67	0.29±0.10	2.72±0.68	**3.47**±1.83	**16.89**±6.66	4.43
	FastGPT-M	**23.87**±10.38	**29.87**±9.03	**32.89**±16.52	**55.49**±7.87	0.67±0.56	16.22±31.34	**26.50**
	DNA-GPT	57.54±4.18	69.91±9.92	39.19±4.23	97.32±0.76	31.24±5.75	65.16±10.11	60.06
	DNA-GPT-M	**93.58**±1.29	**92.53**±3.44	**66.54**±9.55	**99.55**±0.20	**44.36**±9.19	**91.47**±4.41	**81.34**
AUROC	Likelihood	96.16±0.30	98.79±0.19	90.90±1.33	99.29±0.25	92.76±0.23	99.13±0.19	96.17
	Likelihood-M	**98.03**±0.26	**99.22**±0.20	**93.10**±1.43	**99.65**±0.07	**93.47**±1.14	**99.54**±0.16	**97.17**
	Log-Rank	96.55±0.31	98.95±0.13	90.08±1.28	99.36±0.13	**92.01**±0.20	99.24±0.15	96.03
	Log-Rank-M	**98.14**±0.30	**99.20**±0.22	**92.86**±1.66	**99.54**±0.10	91.39±1.08	**99.53**±0.16	**96.78**
	Entropy	74.19±1.62	89.49±0.33	73.26±1.48	84.11±0.77	86.58±0.66	95.94±0.35	83.93
	Entropy-M	**82.99**±1.01	**93.13**±0.23	**81.03**±1.11	**91.15**±0.28	**88.45**±0.44	**96.88**±0.18	**88.94**
	DetectGPT	50.81±0.58	46.40±0.77	57.48±0.84	50.41±1.70	41.54±0.60	17.90±1.25	44.09
	DetectGPT-M	**80.39**±1.33	**80.53**±0.58	**80.67**±1.98	**94.02**±0.56	**62.63**±1.24	**67.25**±1.24	**77.58**
	FastGPT	64.63±1.53	67.68±1.70	47.17±1.53	71.08±1.51	**75.31**±0.90	**88.62**±0.67	69.08
	FastGPT-M	**87.19**±3.46	**91.55**±3.32	**82.58**±18.04	**95.35**±0.49	59.28±1.90	63.48±18.12	**79.91**
	DNA-GPT	96.97±0.20	98.09±0.47	94.11±0.47	99.87±0.04	92.84±0.71	97.85±0.37	96.62
	DNA-GPT-M	**99.37**±0.05	**99.58**±0.10	**97.68**±0.29	**99.92**±0.02	**96.24**±0.47	**99.54**±0.09	**98.72**

Table 7: Performance on Essay dataset. The detection models are trained on text generated by Dolly.

Metric	Method	GPT4All	ChatGPT	Dolly	ChatGLM	Claude	ChatGPT-turbo	avg
TPR@FPR-1%	Likelihood	$46.29_{\pm 16.54}$	$68.62_{\pm 13.32}$	$20.67_{\pm 10.79}$	$92.86_{\pm 4.84}$	$12.31_{\pm 6.29}$	$73.60_{\pm 14.63}$	52.39
	Likelihood-M	**89.02**$_{\pm 1.68}$	**88.09**$_{\pm 4.50}$	**57.14**$_{\pm 5.55}$	**97.59**$_{\pm 1.09}$	**34.36**$_{\pm 13.54}$	**94.36**$_{\pm 3.67}$	**76.76**
	Log-Rank	$62.69_{\pm 13.36}$	$79.07_{\pm 7.89}$	$25.11_{\pm 8.35}$	$95.71_{\pm 2.05}$	$19.16_{\pm 8.22}$	$80.89_{\pm 10.73}$	60.44
	Log-Rank-M	**87.70**$_{\pm 2.60}$	**89.02**$_{\pm 2.13}$	**51.65**$_{\pm 3.99}$	**98.12**$_{\pm 0.54}$	**31.69**$_{\pm 3.32}$	**94.36**$_{\pm 1.93}$	**75.42**
	Entropy	$2.73_{\pm 0.69}$	$7.16_{\pm 3.17}$	$2.29_{\pm 1.36}$	$6.88_{\pm 3.05}$	$3.91_{\pm 2.24}$	$13.24_{\pm 7.60}$	6.04
	Entropy-M	**11.94**$_{\pm 1.85}$	**32.09**$_{\pm 4.77}$	**13.22**$_{\pm 4.31}$	**34.78**$_{\pm 8.04}$	**18.49**$_{\pm 3.20}$	**40.62**$_{\pm 6.34}$	**25.19**
	DetectGPT	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.89_{\pm 0.24}$	$0.00_{\pm 0.00}$	0.15
	DetectGPT-M	**32.89**$_{\pm 22.87}$	**46.09**$_{\pm 19.75}$	**24.44**$_{\pm 9.98}$	**68.84**$_{\pm 21.48}$	**6.04**$_{\pm 4.69}$	**48.09**$_{\pm 25.20}$	**37.73**
	FastGPT	$0.82_{\pm 0.31}$	$0.00_{\pm 0.00}$	$0.38_{\pm 0.42}$	$0.09_{\pm 0.11}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	0.22
	FastGPT-M	**27.24**$_{\pm 2.77}$	**28.04**$_{\pm 7.40}$	**39.43**$_{\pm 3.59}$	**55.89**$_{\pm 5.30}$	**0.98**$_{\pm 0.65}$	**0.84**$_{\pm 0.71}$	**25.41**
	DNA-GPT	$58.95_{\pm 4.76}$	$64.00_{\pm 6.45}$	$48.64_{\pm 4.04}$	$84.29_{\pm 3.64}$	$29.24_{\pm 3.40}$	$59.02_{\pm 8.29}$	57.36
	DNA-GPT-M	**88.70**$_{\pm 1.78}$	**87.82**$_{\pm 1.27}$	**72.60**$_{\pm 4.17}$	**92.59**$_{\pm 0.38}$	**44.71**$_{\pm 4.67}$	**86.93**$_{\pm 2.13}$	**78.89**
AUROC	Likelihood	$96.16_{\pm 0.30}$	$98.79_{\pm 0.19}$	$90.90_{\pm 1.33}$	$99.29_{\pm 0.25}$	$92.76_{\pm 0.24}$	$99.13_{\pm 0.19}$	96.17
	Likelihood-M	**98.63**$_{\pm 0.29}$	**99.48**$_{\pm 0.16}$	**94.44**$_{\pm 0.71}$	**99.62**$_{\pm 0.08}$	**94.49**$_{\pm 0.73}$	**99.70**$_{\pm 0.15}$	**97.73**
	Log-Rank	$96.55_{\pm 0.31}$	$98.95_{\pm 0.13}$	$90.08_{\pm 1.28}$	$99.36_{\pm 0.13}$	$92.01_{\pm 0.20}$	$99.24_{\pm 0.15}$	96.03
	Log-Rank-M	**98.55**$_{\pm 0.08}$	**99.40**$_{\pm 0.09}$	**93.78**$_{\pm 0.93}$	**99.56**$_{\pm 0.09}$	**92.81**$_{\pm 0.38}$	**99.65**$_{\pm 0.10}$	**97.29**
	Entropy	$74.19_{\pm 1.62}$	$89.49_{\pm 0.33}$	$73.26_{\pm 1.48}$	$84.11_{\pm 0.77}$	$86.58_{\pm 0.66}$	$95.94_{\pm 0.35}$	83.93
	Entropy-M	**83.09**$_{\pm 0.84}$	**93.15**$_{\pm 0.17}$	**81.05**$_{\pm 1.14}$	**91.26**$_{\pm 0.24}$	**88.42**$_{\pm 0.52}$	**96.88**$_{\pm 0.20}$	**88.97**
	DetectGPT	$50.81_{\pm 0.58}$	$46.40_{\pm 0.77}$	$57.48_{\pm 0.84}$	$50.41_{\pm 1.70}$	$41.54_{\pm 0.60}$	$17.90_{\pm 1.25}$	44.09
	DetectGPT-M	**93.62**$_{\pm 5.70}$	**93.95**$_{\pm 6.36}$	**90.01**$_{\pm 3.69}$	**97.82**$_{\pm 1.98}$	**81.42**$_{\pm 9.59}$	**92.20**$_{\pm 11.34}$	**91.50**
	FastGPT	$35.37_{\pm 1.53}$	$32.32_{\pm 1.70}$	$52.83_{\pm 1.53}$	$28.92_{\pm 1.51}$	$24.69_{\pm 0.90}$	$11.38_{\pm 0.67}$	30.92
	FastGPT-M	**88.35**$_{\pm 0.89}$	**89.67**$_{\pm 1.11}$	**91.55**$_{\pm 0.28}$	**95.27**$_{\pm 0.38}$	**59.65**$_{\pm 1.54}$	**54.61**$_{\pm 2.30}$	**79.85**
	DNA-GPT	$94.50_{\pm 0.07}$	$93.95_{\pm 0.19}$	$95.91_{\pm 0.49}$	$95.04_{\pm 0.15}$	$89.35_{\pm 0.39}$	$93.81_{\pm 0.20}$	93.76
	DNA-GPT-M	**98.23**$_{\pm 0.21}$	**97.67**$_{\pm 0.16}$	**98.01**$_{\pm 0.33}$	**98.00**$_{\pm 0.14}$	**94.63**$_{\pm 0.40}$	**97.64**$_{\pm 0.17}$	**97.37**

Table 8: Performance on Essay dataset. The detection models are trained on text generated by Claude.

Metric	Method	GPT4All	ChatGPT	Dolly	ChatGLM	Claude	ChatGPT-turbo	avg
TPR@FPR-1%	Likelihood	$46.20_{\pm 16.47}$	$68.58_{\pm 13.28}$	$20.67_{\pm 10.79}$	$92.86_{\pm 4.84}$	$12.31_{\pm 6.29}$	$73.60_{\pm 14.63}$	52.37
	Likelihood-M	**86.74**$_{\pm 3.37}$	**90.09**$_{\pm 2.80}$	**50.07**$_{\pm 9.80}$	**98.08**$_{\pm 0.27}$	**34.00**$_{\pm 6.66}$	**95.78**$_{\pm 1.40}$	**75.79**
	Log-Rank	$62.69_{\pm 13.36}$	$79.07_{\pm 7.89}$	$25.11_{\pm 8.35}$	**95.71**$_{\pm 2.05}$	$19.20_{\pm 8.26}$	$80.89_{\pm 10.73}$	60.44
	Log-Rank-M	**89.20**$_{\pm 2.33}$	**88.18**$_{\pm 2.58}$	**54.42**$_{\pm 3.96}$	$95.00_{\pm 3.56}$	**37.82**$_{\pm 2.28}$	**94.80**$_{\pm 1.37}$	**76.57**
	Entropy	$2.73_{\pm 0.69}$	$7.16_{\pm 3.17}$	$2.29_{\pm 1.36}$	$6.88_{\pm 3.05}$	$3.91_{\pm 2.24}$	$13.24_{\pm 7.60}$	6.04
	Entropy-M	**12.16**$_{\pm 0.86}$	**29.07**$_{\pm 3.22}$	**10.74**$_{\pm 4.12}$	**30.94**$_{\pm 5.62}$	**16.80**$_{\pm 2.73}$	**39.69**$_{\pm 4.99}$	**23.23**
	DetectGPT	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.84_{\pm 0.51}$	$2.36_{\pm 2.52}$	0.53
	DetectGPT-M	**34.62**$_{\pm 17.61}$	**38.71**$_{\pm 23.27}$	**24.73**$_{\pm 8.38}$	**74.06**$_{\pm 17.99}$	**3.91**$_{\pm 2.05}$	**29.11**$_{\pm 24.44}$	**34.19**
	FastGPT	$1.59_{\pm 0.32}$	$1.64_{\pm 0.67}$	$0.29_{\pm 0.10}$	$2.72_{\pm 0.68}$	**3.47**$_{\pm 1.83}$	**16.89**$_{\pm 6.66}$	4.43
	FastGPT-M	**27.47**$_{\pm 2.92}$	**28.04**$_{\pm 7.49}$	**39.57**$_{\pm 3.76}$	**56.47**$_{\pm 6.13}$	$0.98_{\pm 0.65}$	$0.84_{\pm 0.71}$	**25.56**
	DNA-GPT	$0.50_{\pm 0.17}$	$0.31_{\pm 0.11}$	$0.67_{\pm 0.55}$	$0.00_{\pm 0.00}$	$0.13_{\pm 0.11}$	$0.53_{\pm 0.36}$	0.36
	DNA-GPT-M	**12.76**$_{\pm 2.00}$	**13.16**$_{\pm 2.36}$	**14.03**$_{\pm 3.12}$	**12.14**$_{\pm 1.73}$	**13.69**$_{\pm 2.59}$	**13.33**$_{\pm 2.38}$	**13.19**
AUROC	Likelihood	$96.16_{\pm 0.30}$	$98.79_{\pm 0.19}$	$90.90_{\pm 1.33}$	$99.29_{\pm 0.25}$	$92.76_{\pm 0.23}$	$99.13_{\pm 0.19}$	96.17
	Likelihood-M	**98.52**$_{\pm 0.24}$	**99.45**$_{\pm 0.09}$	**94.32**$_{\pm 0.95}$	**99.63**$_{\pm 0.09}$	**94.35**$_{\pm 0.42}$	**99.66**$_{\pm 0.16}$	**97.66**
	Log-Rank	$96.55_{\pm 0.31}$	$98.95_{\pm 0.13}$	$90.08_{\pm 1.28}$	**99.36**$_{\pm 0.13}$	$92.01_{\pm 0.20}$	$99.24_{\pm 0.15}$	96.03
	Log-Rank-M	**98.47**$_{\pm 0.29}$	**99.28**$_{\pm 0.22}$	**93.15**$_{\pm 1.37}$	$99.25_{\pm 0.32}$	**92.93**$_{\pm 0.43}$	**99.66**$_{\pm 0.10}$	**97.13**
	Entropy	$74.19_{\pm 1.62}$	$89.49_{\pm 0.33}$	$73.26_{\pm 1.48}$	$84.11_{\pm 0.77}$	$86.58_{\pm 0.66}$	$95.94_{\pm 0.35}$	83.93
	Entropy-M	**82.36**$_{\pm 1.15}$	**92.90**$_{\pm 0.26}$	**80.64**$_{\pm 1.12}$	**90.88**$_{\pm 0.52}$	**88.62**$_{\pm 0.48}$	**96.91**$_{\pm 0.23}$	**88.72**
	DetectGPT	$50.15_{\pm 0.99}$	$50.53_{\pm 3.65}$	$48.04_{\pm 7.27}$	$49.25_{\pm 1.58}$	$51.21_{\pm 8.39}$	$55.34_{\pm 31.67}$	50.75
	DetectGPT-M	**86.96**$_{\pm 7.59}$	**87.54**$_{\pm 7.67}$	**85.22**$_{\pm 5.50}$	**96.46**$_{\pm 2.24}$	**71.40**$_{\pm 9.73}$	**80.33**$_{\pm 14.18}$	**84.65**
	FastGPT	$64.63_{\pm 1.53}$	$67.68_{\pm 1.70}$	$47.17_{\pm 1.53}$	$71.08_{\pm 1.51}$	**75.31**$_{\pm 0.90}$	**88.62**$_{\pm 0.67}$	69.08
	FastGPT-M	**88.45**$_{\pm 0.98}$	**89.73**$_{\pm 1.25}$	**91.61**$_{\pm 0.36}$	**95.35**$_{\pm 0.47}$	$59.77_{\pm 1.62}$	$54.76_{\pm 2.39}$	**79.95**
	DNA-GPT	$45.85_{\pm 2.15}$	$39.61_{\pm 1.52}$	$53.93_{\pm 1.47}$	$36.19_{\pm 1.75}$	$53.01_{\pm 1.67}$	$40.65_{\pm 1.60}$	44.87
	DNA-GPT-M	**76.26**$_{\pm 1.36}$	**76.25**$_{\pm 1.29}$	**76.38**$_{\pm 1.33}$	**75.98**$_{\pm 1.39}$	**76.74**$_{\pm 1.26}$	**76.30**$_{\pm 1.29}$	**76.32**

Table 9: Performance on DetectRL. The detection models are trained on text generated by ChatGPT and Google-PaLM.

Metric	Method	DetectRL (Training Text: ChatGPT)				DetectRL (Training Text: Google-PaLM)			
		Llama-2-70b	ChatGPT	Google-PaLM	Avg.	Llama-2-70b	ChatGPT	Google-PaLM	Avg.
TPR@FPR-1%	Likelihood	$38.37_{\pm 0.92}$	$10.21_{\pm 1.40}$	$25.66_{\pm 2.41}$	24.75	$38.32_{\pm 0.88}$	**10.19**$_{\pm 1.38}$	$25.66_{\pm 2.41}$	24.72
	Likelihood-M	**45.61**$_{\pm 3.57}$	**12.14**$_{\pm 2.66}$	**33.18**$_{\pm 2.46}$	**30.31**	**42.94**$_{\pm 4.14}$	$9.91_{\pm 3.87}$	**35.48**$_{\pm 2.70}$	**29.44**
	Log-Rank	$42.05_{\pm 0.84}$	$12.44_{\pm 1.71}$	$22.84_{\pm 1.28}$	25.78	$42.05_{\pm 0.84}$	$12.41_{\pm 1.72}$	$22.84_{\pm 1.28}$	25.77
	Log-Rank-M	**50.66**$_{\pm 1.43}$	**12.73**$_{\pm 1.65}$	**41.41**$_{\pm 2.56}$	**34.93**	**50.58**$_{\pm 1.35}$	**13.00**$_{\pm 2.43}$	**41.09**$_{\pm 4.07}$	**34.89**
	Entropy	$2.03_{\pm 0.73}$	$0.25_{\pm 0.16}$	$6.95_{\pm 0.78}$	3.07	$2.03_{\pm 0.73}$	$0.25_{\pm 0.16}$	$6.95_{\pm 0.78}$	3.07
	Entropy-M	**2.13**$_{\pm 0.30}$	**0.27**$_{\pm 0.20}$	**9.27**$_{\pm 0.87}$	**3.89**	**2.22**$_{\pm 0.37}$	**0.27**$_{\pm 0.20}$	**9.37**$_{\pm 0.93}$	**3.96**
	DetectGPT	$3.29_{\pm 0.97}$	$5.36_{\pm 2.20}$	$2.79_{\pm 1.32}$	3.82	$2.87_{\pm 0.56}$	**6.77**$_{\pm 1.50}$	$3.86_{\pm 1.44}$	4.50
	DetectGPT-M	**25.29**$_{\pm 10.05}$	**7.66**$_{\pm 4.17}$	**14.17**$_{\pm 11.10}$	**15.71**	**22.00**$_{\pm 4.30}$	$4.57_{\pm 0.37}$	**24.94**$_{\pm 2.68}$	**17.17**
	FastGPT	$11.35_{\pm 2.01}$	$3.78_{\pm 1.20}$	$5.02_{\pm 0.85}$	6.72	$11.35_{\pm 2.01}$	$3.78_{\pm 1.20}$	$5.02_{\pm 0.85}$	6.72
	FastGPT-M	**27.71**$_{\pm 15.63}$	**13.42**$_{\pm 8.91}$	**17.68**$_{\pm 11.92}$	**19.60**	**12.41**$_{\pm 0.89}$	**6.18**$_{\pm 0.83}$	**20.10**$_{\pm 1.78}$	**12.90**
	DNA-GPT	$31.03_{\pm 1.59}$	$11.05_{\pm 1.69}$	$26.40_{\pm 2.08}$	22.83	$41.71_{\pm 1.32}$	**29.49**$_{\pm 0.84}$	$38.84_{\pm 1.58}$	36.68
	DNA-GPT-M	**41.16**$_{\pm 1.63}$	**12.53**$_{\pm 2.06}$	**33.57**$_{\pm 2.15}$	**29.09**	**47.10**$_{\pm 0.93}$	$22.72_{\pm 1.50}$	**40.54**$_{\pm 1.98}$	**36.79**
AUROC	Likelihood	**78.58**$_{\pm 0.41}$	$66.61_{\pm 0.99}$	$71.42_{\pm 0.49}$	72.2	$78.57_{\pm 0.41}$	$66.61_{\pm 0.99}$	$71.42_{\pm 0.49}$	72.20
	Likelihood-M	$78.15_{\pm 4.58}$	**67.06**$_{\pm 4.19}$	**73.40**$_{\pm 4.38}$	**72.87**	**87.44**$_{\pm 0.70}$	**73.70**$_{\pm 0.57}$	**81.58**$_{\pm 0.93}$	**80.91**
	Log-Rank	$79.67_{\pm 0.46}$	$65.85_{\pm 0.94}$	$70.66_{\pm 0.40}$	72.06	$79.67_{\pm 0.46}$	$65.85_{\pm 0.94}$	$70.66_{\pm 0.40}$	72.06
	Log-Rank-M	**90.23**$_{\pm 0.45}$	**75.40**$_{\pm 0.97}$	**84.44**$_{\pm 0.52}$	**83.35**	**90.14**$_{\pm 0.53}$	**75.61**$_{\pm 0.83}$	**84.63**$_{\pm 0.59}$	**83.46**
	Entropy	$66.21_{\pm 0.88}$	$63.09_{\pm 1.05}$	$60.73_{\pm 0.91}$	63.34	$66.21_{\pm 0.88}$	$63.09_{\pm 1.05}$	$60.73_{\pm 0.91}$	63.34
	Entropy-M	**68.61**$_{\pm 0.94}$	**66.83**$_{\pm 0.95}$	**67.93**$_{\pm 1.08}$	**67.79**	**68.91**$_{\pm 0.83}$	**66.86**$_{\pm 0.95}$	**67.94**$_{\pm 1.04}$	**67.90**
	DetectGPT	$49.21_{\pm 2.31}$	$49.64_{\pm 0.53}$	$50.72_{\pm 6.91}$	49.85	$47.63_{\pm 0.59}$	$49.78_{\pm 0.60}$	$56.81_{\pm 1.41}$	51.40
	DetectGPT-M	**73.50**$_{\pm 9.13}$	**62.93**$_{\pm 4.59}$	**61.30**$_{\pm 15.58}$	**65.91**	**77.77**$_{\pm 4.05}$	**60.34**$_{\pm 3.37}$	**76.40**$_{\pm 1.66}$	**71.50**
	FastGPT	$67.72_{\pm 1.02}$	$58.50_{\pm 1.22}$	$56.68_{\pm 1.21}$	60.97	**67.72**$_{\pm 1.02}$	**58.50**$_{\pm 1.22}$	$56.68_{\pm 1.21}$	60.97
	FastGPT-M	**71.60**$_{\pm 6.86}$	**61.26**$_{\pm 5.62}$	**66.05**$_{\pm 9.70}$	**66.30**	$62.35_{\pm 0.49}$	$51.87_{\pm 1.38}$	**68.88**$_{\pm 1.00}$	**61.03**
	DNA-GPT	$72.53_{\pm 1.19}$	$66.07_{\pm 1.38}$	$68.35_{\pm 0.98}$	68.98	$79.00_{\pm 0.86}$	**72.27**$_{\pm 0.81}$	**77.96**$_{\pm 0.94}$	**76.41**
	DNA-GPT-M	**75.37**$_{\pm 1.13}$	**66.19**$_{\pm 1.40}$	**71.77**$_{\pm 0.79}$	**71.11**	**80.09**$_{\pm 1.26}$	$71.09_{\pm 1.12}$	$77.15_{\pm 1.02}$	76.11

Table 10: Performance on Reuters dataset. The detection models are trained on text generated by GPT4All.

Metric	Method	GPT4All	ChatGPT	Dolly	ChatGLM	Claude	ChatGPT-turbo	Avg.
TPR@FPR-1%	Likelihood	$19.47_{\pm 2.45}$	$80.89_{\pm 2.17}$	$12.44_{\pm 1.38}$	$94.36_{\pm 1.15}$	$18.89_{\pm 3.41}$	$90.22_{\pm 1.23}$	52.71
	Likelihood-M	**61.42**$_{\pm 5.74}$	**88.44**$_{\pm 2.19}$	**35.56**$_{\pm 2.86}$	**95.24**$_{\pm 1.16}$	**32.36**$_{\pm 5.93}$	**92.76**$_{\pm 0.71}$	**67.63**
	Log-Rank	$29.91_{\pm 3.27}$	$86.22_{\pm 1.87}$	$15.20_{\pm 1.70}$	**97.33**$_{\pm 0.64}$	$24.58_{\pm 3.35}$	$93.16_{\pm 1.07}$	57.73
	Log-Rank-M	**73.82**$_{\pm 3.32}$	**89.56**$_{\pm 1.79}$	**38.31**$_{\pm 2.26}$	$96.71_{\pm 1.62}$	**35.47**$_{\pm 4.63}$	**94.49**$_{\pm 0.62}$	**71.39**
	Entropy	$6.18_{\pm 1.40}$	$0.04_{\pm 0.09}$	**13.07**$_{\pm 1.76}$	$0.22_{\pm 0.20}$	$0.00_{\pm 0.00}$	$0.22_{\pm 0.00}$	3.29
	Entropy-M	**17.42**$_{\pm 2.84}$	**0.36**$_{\pm 0.30}$	$12.36_{\pm 2.86}$	**0.98**$_{\pm 0.87}$	$0.00_{\pm 0.00}$	**0.40**$_{\pm 0.38}$	**5.25**
	DetectGPT	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.44_{\pm 0.40}$	$0.04_{\pm 0.09}$	0.08
	DetectGPT-M	**40.89**$_{\pm 6.07}$	**15.33**$_{\pm 8.81}$	**46.40**$_{\pm 2.46}$	**40.00**$_{\pm 8.10}$	**2.40**$_{\pm 0.38}$	**6.40**$_{\pm 2.50}$	**25.24**
	FastGPT	$3.56_{\pm 3.08}$	$0.04_{\pm 0.09}$	$12.18_{\pm 6.70}$	$0.09_{\pm 0.11}$	$0.09_{\pm 0.11}$	$0.04_{\pm 0.09}$	2.67
	FastGPT-M	**61.69**$_{\pm 7.18}$	**21.78**$_{\pm 6.77}$	**65.24**$_{\pm 5.12}$	**31.78**$_{\pm 9.97}$	**1.38**$_{\pm 0.84}$	**1.51**$_{\pm 0.55}$	**30.56**
	DNA-GPT	$85.20_{\pm 6.79}$	$92.58_{\pm 3.94}$	$68.49_{\pm 4.37}$	$96.93_{\pm 3.14}$	$79.24_{\pm 4.93}$	$95.29_{\pm 3.53}$	86.29
	DNA-GPT-M	**97.51**$_{\pm 1.31}$	**98.40**$_{\pm 0.88}$	**81.29**$_{\pm 3.75}$	**99.87**$_{\pm 0.11}$	**89.07**$_{\pm 3.57}$	**99.24**$_{\pm 0.30}$	**94.23**
AUROC	Likelihood	$75.19_{\pm 1.39}$	$97.55_{\pm 0.33}$	$58.77_{\pm 1.76}$	**99.62**$_{\pm 0.11}$	$86.45_{\pm 0.70}$	$98.42_{\pm 0.25}$	86.00
	Likelihood-M	**93.74**$_{\pm 1.23}$	**98.42**$_{\pm 0.26}$	**77.05**$_{\pm 0.53}$	$99.61_{\pm 0.11}$	**90.68**$_{\pm 0.56}$	**98.78**$_{\pm 0.33}$	**93.05**
	Log-Rank	$78.39_{\pm 1.23}$	$97.77_{\pm 0.32}$	$58.22_{\pm 2.03}$	$99.61_{\pm 0.19}$	$85.68_{\pm 0.67}$	$98.67_{\pm 0.24}$	86.39
	Log-Rank-M	**95.50**$_{\pm 0.81}$	**98.60**$_{\pm 0.23}$	**76.60**$_{\pm 1.33}$	**99.64**$_{\pm 0.29}$	**90.45**$_{\pm 0.69}$	**98.99**$_{\pm 0.31}$	**93.30**
	Entropy	**72.48**$_{\pm 1.24}$	**35.35**$_{\pm 1.65}$	**61.92**$_{\pm 1.32}$	**50.96**$_{\pm 1.57}$	$23.76_{\pm 1.05}$	$17.55_{\pm 0.92}$	**43.67**
	Entropy-M	$70.31_{\pm 1.98}$	$34.84_{\pm 2.95}$	$50.09_{\pm 1.57}$	$46.62_{\pm 1.12}$	**27.55**$_{\pm 2.60}$	**29.08**$_{\pm 4.18}$	43.08
	DetectGPT	$75.62_{\pm 1.20}$	$57.18_{\pm 1.91}$	$77.25_{\pm 0.66}$	$61.97_{\pm 0.90}$	$58.01_{\pm 1.75}$	$20.42_{\pm 1.07}$	58.41
	DetectGPT-M	**87.58**$_{\pm 0.93}$	**78.19**$_{\pm 2.33}$	**86.29**$_{\pm 1.18}$	**89.73**$_{\pm 0.97}$	**58.40**$_{\pm 1.54}$	**65.72**$_{\pm 1.36}$	**77.65**
	FastGPT	$76.50_{\pm 1.39}$	$39.56_{\pm 0.54}$	$82.45_{\pm 0.47}$	$33.21_{\pm 1.82}$	$41.31_{\pm 0.89}$	$11.29_{\pm 0.49}$	47.39
	FastGPT-M	**96.46**$_{\pm 0.31}$	**90.21**$_{\pm 0.75}$	**96.23**$_{\pm 0.36}$	**92.36**$_{\pm 0.74}$	**66.98**$_{\pm 1.06}$	**56.28**$_{\pm 1.80}$	**83.09**
	DNA-GPT	$99.51_{\pm 0.14}$	$99.72_{\pm 0.07}$	**98.41**$_{\pm 0.21}$	$99.85_{\pm 0.05}$	$99.21_{\pm 0.14}$	$99.81_{\pm 0.06}$	99.42
	DNA-GPT-M	**99.80**$_{\pm 0.09}$	**99.88**$_{\pm 0.07}$	$98.39_{\pm 0.17}$	**99.94**$_{\pm 0.06}$	**99.53**$_{\pm 0.09}$	**99.89**$_{\pm 0.09}$	**99.57**

Table 11: Performance on Reuters dataset. The detection models are trained on text generated by ChatGPT.

Metric	Method	GPT4All	ChatGPT	Dolly	ChatGLM	Claude	ChatGPT-turbo	Avg.
TPR@FPR-1%	Likelihood	$19.78_{\pm 2.21}$	$81.07_{\pm 2.05}$	$12.53_{\pm 1.35}$	$94.36_{\pm 1.15}$	$19.42_{\pm 3.17}$	$90.31_{\pm 1.11}$	52.91
	Likelihood-M	**59.73**$_{\pm 8.56}$	**88.98**$_{\pm 1.58}$	**35.91**$_{\pm 3.82}$	**96.00**$_{\pm 2.14}$	**26.71**$_{\pm 5.93}$	**93.24**$_{\pm 0.94}$	**66.76**
	Log-Rank	$29.96_{\pm 3.29}$	$86.22_{\pm 1.87}$	$15.20_{\pm 1.70}$	$97.33_{\pm 0.64}$	$24.58_{\pm 3.35}$	$93.20_{\pm 1.13}$	57.75
	Log-Rank-M	**63.91**$_{\pm 8.04}$	**90.62**$_{\pm 2.38}$	**34.58**$_{\pm 4.71}$	**98.13**$_{\pm 0.46}$	**29.73**$_{\pm 8.71}$	**94.53**$_{\pm 0.67}$	**68.59**
	Entropy	$0.09_{\pm 0.11}$	$2.49_{\pm 1.25}$	$2.04_{\pm 0.57}$	$1.69_{\pm 0.54}$	$4.36_{\pm 1.81}$	$9.56_{\pm 2.90}$	3.37
	Entropy-M	**7.07**$_{\pm 3.05}$	**17.47**$_{\pm 5.59}$	**13.60**$_{\pm 2.23}$	**9.02**$_{\pm 3.21}$	**18.09**$_{\pm 4.26}$	**33.29**$_{\pm 7.84}$	**16.42**
	DetectGPT	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.44_{\pm 0.40}$	$0.04_{\pm 0.09}$	0.08
	DetectGPT-M	**22.76**$_{\pm 25.66}$	**47.91**$_{\pm 28.49}$	**11.69**$_{\pm 12.56}$	**23.64**$_{\pm 28.60}$	**8.27**$_{\pm 5.92}$	**71.24**$_{\pm 17.28}$	**30.92**
	FastGPT	$0.31_{\pm 0.23}$	$2.04_{\pm 0.86}$	$0.04_{\pm 0.09}$	$3.16_{\pm 1.03}$	**1.42**$_{\pm 0.57}$	**23.51**$_{\pm 4.15}$	5.08
	FastGPT-M	**56.04**$_{\pm 11.84}$	**17.07**$_{\pm 10.24}$	**61.51**$_{\pm 9.34}$	**26.67**$_{\pm 13.24}$	$0.89_{\pm 0.92}$	$1.29_{\pm 0.68}$	**27.24**
	DNA-GPT	$29.82_{\pm 18.89}$	$69.78_{\pm 22.51}$	$15.64_{\pm 12.02}$	$83.47_{\pm 14.87}$	$21.96_{\pm 19.50}$	$82.22_{\pm 15.63}$	50.48
	DNA-GPT-M	**86.80**$_{\pm 5.37}$	**96.76**$_{\pm 1.16}$	**59.33**$_{\pm 8.10}$	**99.29**$_{\pm 0.38}$	**61.91**$_{\pm 13.48}$	**97.51**$_{\pm 1.24}$	**83.60**
AUROC	Likelihood	$75.19_{\pm 1.39}$	$97.55_{\pm 0.33}$	$58.77_{\pm 1.76}$	$99.62_{\pm 0.11}$	$86.46_{\pm 0.70}$	$98.42_{\pm 0.25}$	86.00
	Likelihood-M	**93.97**$_{\pm 1.27}$	**98.68**$_{\pm 0.41}$	**79.54**$_{\pm 1.12}$	**99.70**$_{\pm 0.24}$	**90.39**$_{\pm 1.08}$	**98.94**$_{\pm 0.23}$	**93.54**
	Log-Rank	$78.39_{\pm 1.23}$	$97.77_{\pm 0.32}$	$58.22_{\pm 2.03}$	$99.61_{\pm 0.19}$	$85.68_{\pm 0.67}$	$98.67_{\pm 0.24}$	86.39
	Log-Rank-M	**93.41**$_{\pm 1.37}$	**98.69**$_{\pm 0.25}$	**76.76**$_{\pm 1.45}$	**99.75**$_{\pm 0.20}$	**89.55**$_{\pm 1.04}$	**99.03**$_{\pm 0.27}$	**92.86**
	Entropy	$27.52_{\pm 1.24}$	$64.65_{\pm 1.65}$	$38.08_{\pm 1.32}$	$49.04_{\pm 1.57}$	$76.24_{\pm 1.05}$	$82.45_{\pm 0.92}$	56.33
	Entropy-M	**55.30**$_{\pm 2.65}$	**80.07**$_{\pm 1.89}$	**56.54**$_{\pm 2.20}$	**69.75**$_{\pm 2.62}$	**83.33**$_{\pm 1.09}$	**90.46**$_{\pm 1.17}$	**72.58**
	DetectGPT	$75.62_{\pm 1.20}$	$57.18_{\pm 1.91}$	**77.25**$_{\pm 0.66}$	$61.97_{\pm 0.90}$	$58.01_{\pm 1.75}$	$20.42_{\pm 1.07}$	58.41
	DetectGPT-M	**84.60**$_{\pm 11.69}$	**92.37**$_{\pm 5.63}$	$69.47_{\pm 12.90}$	**80.47**$_{\pm 13.77}$	**75.78**$_{\pm 9.49}$	**97.20**$_{\pm 1.23}$	**83.32**
	FastGPT	$23.50_{\pm 1.39}$	$60.44_{\pm 0.54}$	$17.55_{\pm 0.47}$	$66.79_{\pm 1.82}$	$58.69_{\pm 0.89}$	**88.71**$_{\pm 0.49}$	52.61
	FastGPT-M	**96.25**$_{\pm 0.33}$	**89.97**$_{\pm 0.74}$	**96.04**$_{\pm 0.33}$	**92.12**$_{\pm 0.76}$	**66.72**$_{\pm 1.11}$	$56.04_{\pm 1.73}$	**82.86**
	DNA-GPT	$97.34_{\pm 0.54}$	$99.17_{\pm 0.30}$	$95.82_{\pm 0.62}$	$99.45_{\pm 0.23}$	$97.21_{\pm 0.52}$	$99.38_{\pm 0.26}$	98.06
	DNA-GPT-M	**99.42**$_{\pm 0.15}$	**99.78**$_{\pm 0.09}$	**97.31**$_{\pm 0.24}$	**99.89**$_{\pm 0.10}$	**98.78**$_{\pm 0.24}$	**99.82**$_{\pm 0.12}$	**99.16**

Table 12: Performance on Reuters dataset. The detection models are trained on text generated by ChatGPT-turbo.

Metric	Method	GPT4All	ChatGPT	Dolly	ChatGLM	Claude	ChatGPT-turbo	Avg.
TPR@FPR-1%	Likelihood	$19.82_{\pm 2.17}$	$81.07_{\pm 2.05}$	$12.53_{\pm 1.35}$	$94.44_{\pm 1.10}$	$19.42_{\pm 3.17}$	$90.27_{\pm 1.07}$	52.93
	Likelihood-M	**66.62**$_{\pm 5.21}$	**90.13**$_{\pm 1.15}$	**38.62**$_{\pm 2.74}$	**95.96**$_{\pm 1.29}$	**29.60**$_{\pm 2.32}$	**93.91**$_{\pm 1.03}$	**69.14**
	Log-Rank	$29.96_{\pm 3.29}$	$86.22_{\pm 1.87}$	$15.20_{\pm 1.70}$	$97.33_{\pm 0.64}$	$24.58_{\pm 3.35}$	$93.20_{\pm 1.13}$	57.75
	Log-Rank-M	**60.18**$_{\pm 8.59}$	**90.27**$_{\pm 1.98}$	**33.07**$_{\pm 3.84}$	**98.22**$_{\pm 0.40}$	**27.29**$_{\pm 6.31}$	**94.00**$_{\pm 0.40}$	**67.17**
	Entropy	$0.09_{\pm 0.11}$	$2.49_{\pm 1.25}$	$2.04_{\pm 0.57}$	$1.69_{\pm 0.54}$	$4.36_{\pm 1.81}$	$9.60_{\pm 2.94}$	3.38
	Entropy-M	**5.11**$_{\pm 2.70}$	**16.18**$_{\pm 4.91}$	**12.27**$_{\pm 1.51}$	**7.69**$_{\pm 2.70}$	**17.42**$_{\pm 3.93}$	**31.69**$_{\pm 6.61}$	**15.06**
	DetectGPT	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.62_{\pm 0.53}$	$9.24_{\pm 2.65}$	1.64
	DetectGPT-M	**6.22**$_{\pm 3.99}$	**26.71**$_{\pm 14.32}$	**2.49**$_{\pm 0.97}$	**4.04**$_{\pm 3.33}$	**4.18**$_{\pm 2.22}$	**60.67**$_{\pm 14.38}$	**17.39**
	FastGPT	**0.31**$_{\pm 0.23}$	**2.04**$_{\pm 0.86}$	$0.04_{\pm 0.09}$	**3.16**$_{\pm 1.03}$	**1.42**$_{\pm 0.57}$	**23.51**$_{\pm 4.15}$	**5.08**
	FastGPT-M	$0.00_{\pm 0.00}$	$0.22_{\pm 0.28}$	$0.04_{\pm 0.09}$	$0.53_{\pm 0.33}$	$0.62_{\pm 0.29}$	$7.38_{\pm 2.42}$	1.47
	DNA-GPT	$2.09_{\pm 1.29}$	$4.31_{\pm 1.94}$	$1.69_{\pm 1.21}$	$5.51_{\pm 1.67}$	$1.96_{\pm 1.60}$	$5.24_{\pm 2.29}$	3.47
	DNA-GPT-M	**5.82**$_{\pm 1.06}$	**6.44**$_{\pm 0.97}$	**3.56**$_{\pm 1.27}$	**6.49**$_{\pm 0.87}$	**3.78**$_{\pm 1.29}$	**6.67**$_{\pm 0.84}$	**5.46**
AUROC	Likelihood	$75.20_{\pm 1.39}$	$97.55_{\pm 0.33}$	$58.77_{\pm 1.77}$	$99.62_{\pm 0.11}$	$86.46_{\pm 0.70}$	$98.42_{\pm 0.25}$	86.00
	Likelihood-M	**94.79**$_{\pm 0.60}$	**98.72**$_{\pm 0.40}$	**79.93**$_{\pm 1.44}$	**99.72**$_{\pm 0.16}$	**90.83**$_{\pm 0.53}$	**98.96**$_{\pm 0.26}$	**93.82**
	Log-Rank	$78.39_{\pm 1.23}$	$97.77_{\pm 0.32}$	$58.22_{\pm 2.03}$	$99.61_{\pm 0.19}$	$85.68_{\pm 0.67}$	$98.67_{\pm 0.24}$	86.39
	Log-Rank-M	**92.21**$_{\pm 2.45}$	**98.63**$_{\pm 0.23}$	**76.27**$_{\pm 1.24}$	**99.73**$_{\pm 0.21}$	**89.02**$_{\pm 0.89}$	**98.99**$_{\pm 0.27}$	**92.48**
	Entropy	$27.52_{\pm 1.24}$	$64.65_{\pm 1.65}$	$38.08_{\pm 1.32}$	$49.04_{\pm 1.57}$	$76.24_{\pm 1.05}$	$82.45_{\pm 0.92}$	56.33
	Entropy-M	**54.11**$_{\pm 3.57}$	**79.38**$_{\pm 1.96}$	**55.94**$_{\pm 1.82}$	**68.88**$_{\pm 2.66}$	**83.14**$_{\pm 1.07}$	**89.91**$_{\pm 0.91}$	**71.89**
	DetectGPT	$24.38_{\pm 1.20}$	$42.82_{\pm 1.91}$	$22.75_{\pm 0.66}$	$38.03_{\pm 0.90}$	$41.99_{\pm 1.75}$	$79.58_{\pm 1.07}$	41.59
	DetectGPT-M	**79.17**$_{\pm 7.47}$	**89.36**$_{\pm 4.32}$	**59.50**$_{\pm 5.37}$	**71.41**$_{\pm 7.62}$	**73.92**$_{\pm 8.27}$	**96.53**$_{\pm 1.14}$	**78.32**
	FastGPT	**23.50**$_{\pm 1.39}$	**60.44**$_{\pm 0.54}$	**17.55**$_{\pm 0.47}$	**66.79**$_{\pm 1.82}$	**58.69**$_{\pm 0.89}$	**88.71**$_{\pm 0.49}$	**52.61**
	FastGPT-M	$12.73_{\pm 1.55}$	$35.93_{\pm 2.31}$	$9.43_{\pm 0.56}$	$36.46_{\pm 3.11}$	$50.18_{\pm 2.00}$	$76.28_{\pm 1.01}$	36.83
	DNA-GPT	$45.62_{\pm 1.73}$	$63.85_{\pm 1.00}$	$35.96_{\pm 1.28}$	$71.24_{\pm 1.14}$	$43.56_{\pm 1.49}$	$70.63_{\pm 1.33}$	55.14
	DNA-GPT-M	**79.05**$_{\pm 1.13}$	**87.32**$_{\pm 0.51}$	**64.13**$_{\pm 1.27}$	**91.42**$_{\pm 0.64}$	**67.14**$_{\pm 1.48}$	**89.37**$_{\pm 0.79}$	**79.74**

Table 13: Performance on Reuters dataset. The detection models are trained on text generated by ChatGLM.

Metric	Method	GPT4All	ChatGPT	Dolly	ChatGLM	Claude	ChatGPT-turbo	Avg.
TPR@FPR-1%	Likelihood	19.78±2.13	81.07±2.05	12.49±1.33	94.44±1.10	19.42±3.17	90.36±1.06	52.93
	Likelihood-M	**41.42**±8.12	**86.84**±2.44	**31.47**±2.22	**98.58**±0.23	**20.84**±2.97	**91.56**±1.28	**61.79**
	Log-Rank	29.96±3.29	86.22±1.87	15.20±1.70	97.33±0.64	24.58±3.35	93.20±1.13	57.75
	Log-Rank-M	**62.40**±4.54	**90.49**±2.34	**33.82**±3.87	**98.22**±0.47	**30.44**±6.61	**94.13**±0.75	**68.25**
	Entropy	4.89±2.71	0.36±0.71	10.49±4.75	0.44±0.37	0.58±1.16	1.38±2.31	3.02
	Entropy-M	**6.13**±4.80	**17.20**±4.05	**13.11**±3.21	**9.73**±4.98	**18.76**±2.64	**33.87**±5.03	**16.47**
	DetectGPT	0.00±0.00	0.00±0.00	0.00±0.00	0.00±0.00	0.44±0.40	0.04±0.09	0.08
	DetectGPT-M	**40.53**±6.43	**14.18**±7.67	**45.73**±2.50	**38.98**±8.03	**2.18**±0.53	**5.47**±2.15	**24.51**
	FastGPT	0.31±0.23	2.04±0.86	0.04±0.09	3.16±1.03	**1.42**±0.57	**23.51**±4.15	5.08
	FastGPT-M	**56.04**±11.84	**17.07**±10.24	**61.51**±9.34	**26.67**±13.24	0.89±0.92	1.29±0.68	**27.24**
	DNA-GPT	27.96±18.31	66.84±18.61	16.00±12.45	85.51±12.79	20.49±17.96	80.40±13.38	49.53
	DNA-GPT-M	**82.89**±7.12	**94.93**±1.99	**56.31**±9.16	**98.98**±0.18	**57.78**±15.40	**96.53**±1.23	**81.24**
AUROC	Likelihood	75.19±1.39	97.55±0.33	58.77±1.77	99.62±0.11	86.46±0.70	98.42±0.25	86.00
	Likelihood-M	**87.15**±3.71	**98.44**±0.43	**77.92**±1.74	**99.95**±0.00	**87.99**±1.16	**98.69**±0.26	**91.69**
	Log-Rank	78.39±1.23	97.77±0.32	58.22±2.03	99.61±0.19	85.68±0.67	98.67±0.24	86.39
	Log-Rank-M	**93.39**±0.75	**98.68**±0.24	**76.11**±1.61	**99.75**±0.19	**89.70**±0.41	**98.99**±0.25	**92.77**
	Entropy	**62.76**±18.55	40.53±11.30	**56.16**±10.29	49.39±1.74	33.61±20.52	30.23±25.75	45.45
	Entropy-M	54.30±3.63	**79.38**±1.33	55.89±1.46	**69.08**±2.00	**83.03**±0.44	**89.77**±1.24	**71.91**
	DetectGPT	75.62±1.20	57.18±1.91	77.25±0.66	61.97±0.90	**58.01**±1.75	20.42±1.07	58.41
	DetectGPT-M	**87.17**±0.76	**77.46**±1.85	**86.14**±1.14	**89.36**±1.25	57.83±1.55	**64.40**±1.89	**77.06**
	FastGPT	23.50±1.39	60.44±0.54	17.55±0.47	66.79±1.82	58.69±0.89	**88.71**±0.49	52.61
	FastGPT-M	**96.27**±0.33	**89.98**±0.74	**96.07**±0.33	**92.13**±0.76	**66.74**±1.10	56.05±1.74	**82.87**
	DNA-GPT	97.10±0.48	98.94±0.26	95.56±0.57	99.29±0.17	96.95±0.46	99.22±0.22	97.85
	DNA-GPT-M	**99.33**±0.15	**99.72**±0.09	**97.15**±0.21	**99.82**±0.10	**98.68**±0.24	**99.77**±0.12	**99.08**

Table 14: Performance on Reuters dataset. The detection models are trained on text generated by Dolly.

Metric	Method	GPT4All	ChatGPT	Dolly	ChatGLM	Claude	ChatGPT-turbo	Avg.
TPR@FPR-1%	Likelihood	19.78±2.21	81.07±2.05	12.40±1.37	94.44±1.10	19.38±3.23	90.27±1.07	52.89
	Likelihood-M	**59.96**±5.45	**88.58**±2.29	**35.42**±2.84	**95.91**±1.21	**31.96**±6.22	**92.93**±0.77	**67.46**
	Log-Rank	6.62±11.36	17.24±34.49	10.49±2.42	19.42±38.62	4.93±9.87	18.58±37.16	12.88
	Log-Rank-M	**66.76**±7.01	**90.49**±1.80	**35.42**±4.26	**98.09**±0.72	**31.78**±6.22	**94.31**±0.39	**69.47**
	Entropy	**6.18**±1.40	0.04±0.09	**13.07**±1.76	0.22±0.20	0.00±0.00	0.22±0.00	3.29
	Entropy-M	2.40±0.74	**7.29**±5.90	9.91±1.09	**3.16**±2.61	**9.24**±7.67	**16.13**±13.02	**8.02**
	DetectGPT	0.00±0.00	0.00±0.00	0.00±0.00	0.00±0.00	0.44±0.40	0.04±0.09	0.08
	DetectGPT-M	**42.93**±25.42	**37.20**±35.68	**39.69**±19.14	**51.73**±37.94	**6.27**±7.20	**31.56**±34.65	**34.90**
	FastGPT	3.56±3.08	0.04±0.09	12.18±6.70	0.09±0.11	0.09±0.11	0.04±0.09	2.67
	FastGPT-M	**61.69**±7.18	**21.78**±6.77	**65.24**±5.12	**31.78**±9.97	**1.38**±0.84	**1.51**±0.55	**30.56**
	DNA-GPT	78.71±8.56	91.47±7.33	70.00±10.40	95.51±4.74	76.76±8.45	94.80±5.80	84.54
	DNA-GPT-M	**93.24**±5.57	**98.09**±1.34	**79.07**±7.29	**99.87**±0.27	**85.20**±6.66	**98.31**±1.35	**92.30**
AUROC	Likelihood	75.19±1.39	97.55±0.33	58.77±1.76	99.62±0.10	86.45±0.70	98.42±0.25	86.00
	Likelihood-M	**93.42**±0.91	**98.45**±0.29	**77.41**±1.04	**99.67**±0.16	**90.57**±0.61	**98.78**±0.33	**93.05**
	Log-Rank	32.84±22.66	21.28±38.17	43.51±5.44	20.22±39.67	28.30±28.33	20.77±38.91	27.82
	Log-Rank-M	**94.18**±0.71	**98.71**±0.19	**76.52**±1.64	**99.73**±0.23	**89.98**±0.70	**99.02**±0.31	**93.02**
	Entropy	**72.48**±1.24	35.35±1.65	**61.92**±1.32	50.96±1.57	23.76±1.05	17.55±0.92	43.67
	Entropy-M	49.80±2.95	**56.16**±26.44	52.47±1.86	**53.09**±16.63	**56.78**±31.30	**58.33**±37.31	**54.44**
	DetectGPT	75.62±1.20	57.18±1.91	77.25±0.66	61.97±0.90	**58.01**±1.75	20.42±1.07	58.41
	DetectGPT-M	**80.34**±21.91	**73.22**±28.90	**81.89**±11.41	**91.04**±8.24	57.53±18.26	**66.30**±30.95	**75.05**
	FastGPT	76.50±1.39	39.56±0.54	82.45±0.47	33.21±1.82	41.31±0.89	11.29±0.49	47.39
	FastGPT-M	**96.47**±0.30	**90.23**±0.73	**96.24**±0.37	**92.37**±0.72	**67.04**±1.07	**56.34**±1.90	**83.12**
	DNA-GPT	99.19±0.21	99.74±0.09	**98.90**±0.24	99.83±0.07	99.21±0.16	99.80±0.08	99.45
	DNA-GPT-M	**99.66**±0.19	**99.84**±0.13	98.81±0.22	**99.91**±0.10	**99.47**±0.20	**99.86**±0.14	**99.59**

Table 15: Performance on Reuters dataset. The detection models are trained on text generated by Claude.

Metric	Method	GPT4All	ChatGPT	ChatGPT-turbo	ChatGLM	StableLM	Claude	Avg.
TPR@FPR-1%	Likelihood	$19.82_{\pm 2.17}$	$81.07_{\pm 2.05}$	$12.53_{\pm 1.35}$	**94.44**$_{\pm 1.10}$	$19.42_{\pm 3.17}$	$90.36_{\pm 1.06}$	52.94
	Likelihood-M	**60.18**$_{\pm 5.26}$	**84.84**$_{\pm 2.91}$	**32.13**$_{\pm 2.78}$	$91.78_{\pm 1.51}$	**34.58**$_{\pm 4.58}$	**91.47**$_{\pm 1.34}$	**65.83**
	Log-Rank	$29.96_{\pm 3.29}$	$86.22_{\pm 1.87}$	$15.20_{\pm 1.70}$	**97.33**$_{\pm 0.64}$	$24.58_{\pm 3.35}$	$93.20_{\pm 1.13}$	57.75
	Log-Rank-M	**71.33**$_{\pm 4.00}$	**89.82**$_{\pm 2.47}$	**37.60**$_{\pm 1.64}$	$97.07_{\pm 1.44}$	**34.84**$_{\pm 5.56}$	**94.36**$_{\pm 0.87}$	**70.84**
	Entropy	$0.09_{\pm 0.11}$	$2.49_{\pm 1.25}$	$2.04_{\pm 0.57}$	$1.69_{\pm 0.54}$	$4.36_{\pm 1.81}$	$9.60_{\pm 2.94}$	3.38
	Entropy-M	**5.51**$_{\pm 2.56}$	**17.42**$_{\pm 5.92}$	**13.24**$_{\pm 2.62}$	**7.91**$_{\pm 2.58}$	**18.00**$_{\pm 4.28}$	**34.09**$_{\pm 8.23}$	**16.03**
	DetectGPT	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.00_{\pm 0.00}$	$0.44_{\pm 0.40}$	$0.04_{\pm 0.09}$	0.08
	DetectGPT-M	**39.91**$_{\pm 20.48}$	**51.60**$_{\pm 22.14}$	**28.80**$_{\pm 20.74}$	**57.47**$_{\pm 35.18}$	**6.00**$_{\pm 3.22}$	**78.53**$_{\pm 8.07}$	**43.72**
	FastGPT	$0.31_{\pm 0.23}$	$2.04_{\pm 0.86}$	$0.04_{\pm 0.09}$	$3.16_{\pm 1.03}$	$1.42_{\pm 0.57}$	**23.51**$_{\pm 4.15}$	5.08
	FastGPT-M	**59.96**$_{\pm 10.36}$	**33.11**$_{\pm 28.98}$	**56.09**$_{\pm 16.45}$	**42.44**$_{\pm 27.14}$	**8.62**$_{\pm 15.04}$	$19.64_{\pm 36.52}$	**36.64**
	DNA-GPT	$3.64_{\pm 1.52}$	$6.58_{\pm 1.66}$	$2.18_{\pm 1.15}$	**7.78**$_{\pm 0.95}$	$2.44_{\pm 2.03}$	**7.20**$_{\pm 1.15}$	4.97
	DNA-GPT-M	**6.67**$_{\pm 2.96}$	**7.16**$_{\pm 3.18}$	**4.22**$_{\pm 2.00}$	$7.51_{\pm 3.26}$	**5.38**$_{\pm 2.87}$	$7.16_{\pm 3.17}$	**6.35**
AUROC	Likelihood	$75.19_{\pm 1.39}$	$97.55_{\pm 0.33}$	$58.77_{\pm 1.77}$	**99.62**$_{\pm 0.10}$	$86.45_{\pm 0.70}$	$98.42_{\pm 0.25}$	86.00
	Likelihood-M	**91.12**$_{\pm 2.58}$	**97.69**$_{\pm 0.49}$	**69.45**$_{\pm 3.64}$	$98.38_{\pm 0.57}$	**90.57**$_{\pm 0.59}$	**98.54**$_{\pm 0.43}$	**90.96**
	Log-Rank	$78.40_{\pm 1.23}$	$97.77_{\pm 0.32}$	$58.22_{\pm 2.03}$	$99.61_{\pm 0.19}$	$85.68_{\pm 0.67}$	$98.67_{\pm 0.24}$	86.39
	Log-Rank-M	**95.06**$_{\pm 0.64}$	**98.59**$_{\pm 0.28}$	**76.35**$_{\pm 1.49}$	**99.65**$_{\pm 0.22}$	**90.36**$_{\pm 0.68}$	**98.98**$_{\pm 0.27}$	**93.16**
	Entropy	$27.52_{\pm 1.24}$	$64.65_{\pm 1.65}$	$38.08_{\pm 1.32}$	$49.04_{\pm 1.57}$	$76.24_{\pm 1.05}$	$82.45_{\pm 0.92}$	56.33
	Entropy-M	**53.47**$_{\pm 2.35}$	**79.33**$_{\pm 1.73}$	**55.92**$_{\pm 1.77}$	**68.50**$_{\pm 2.29}$	**83.03**$_{\pm 1.03}$	**90.09**$_{\pm 0.94}$	**71.72**
	DetectGPT	$75.62_{\pm 1.20}$	$57.18_{\pm 1.91}$	$77.25_{\pm 0.66}$	$61.97_{\pm 0.90}$	$58.01_{\pm 1.75}$	$20.42_{\pm 1.07}$	58.41
	DetectGPT-M	**91.23**$_{\pm 5.89}$	**94.13**$_{\pm 1.77}$	**81.29**$_{\pm 8.98}$	**94.63**$_{\pm 5.95}$	**70.99**$_{\pm 6.72}$	**96.97**$_{\pm 1.09}$	**88.21**
	FastGPT	$23.50_{\pm 1.39}$	$60.44_{\pm 0.54}$	$17.55_{\pm 0.47}$	$66.79_{\pm 1.82}$	$58.69_{\pm 0.89}$	**88.71**$_{\pm 0.49}$	52.61
	FastGPT-M	**96.53**$_{\pm 0.51}$	**91.64**$_{\pm 3.61}$	**94.46**$_{\pm 2.87}$	**93.63**$_{\pm 3.03}$	**71.74**$_{\pm 10.00}$	$63.82_{\pm 17.64}$	**85.30**
	DNA-GPT	$65.77_{\pm 1.29}$	$78.38_{\pm 0.64}$	$54.85_{\pm 1.20}$	$82.40_{\pm 0.73}$	$65.30_{\pm 0.67}$	$81.64_{\pm 0.76}$	71.39
	DNA-GPT-M	**78.84**$_{\pm 12.13}$	**83.32**$_{\pm 13.75}$	**68.13**$_{\pm 8.88}$	**85.53**$_{\pm 14.38}$	**72.95**$_{\pm 9.83}$	**84.29**$_{\pm 14.01}$	**78.84**

Table 16: Performance on TruthfulQA dataset. The detection models are trained on text generated by GPT4.

Metric	Method	GPT4	ChatGPT-turbo	ChatGLM	Dolly	ChatGPT	StableLM	Avg.
TPR@FPR-1%	Likelihood	**38.45**$_{\pm 3.34}$	$41.67_{\pm 1.51}$	$73.54_{\pm 7.83}$	$17.74_{\pm 2.72}$	**52.01**$_{\pm 10.78}$	$50.14_{\pm 8.09}$	45.59
	Likelihood-E	$29.17_{\pm 11.92}$	**56.89**$_{\pm 19.43}$	**75.91**$_{\pm 17.62}$	**52.17**$_{\pm 2.62}$	$42.04_{\pm 19.77}$	**54.79**$_{\pm 10.29}$	**51.83**
	Log-Rank	$40.17_{\pm 4.46}$	$39.77_{\pm 7.89}$	$83.22_{\pm 4.08}$	$16.45_{\pm 5.07}$	$54.50_{\pm 2.09}$	$59.26_{\pm 2.20}$	48.9
	Log-Rank-E	**47.74**$_{\pm 1.72}$	**75.39**$_{\pm 4.05}$	**90.55**$_{\pm 2.10}$	**44.48**$_{\pm 4.99}$	**83.57**$_{\pm 1.73}$	**79.26**$_{\pm 2.52}$	**70.16**
	Entropy	$24.30_{\pm 4.47}$	$38.39_{\pm 6.85}$	$60.48_{\pm 15.23}$	$18.38_{\pm 6.58}$	$38.02_{\pm 9.81}$	$32.29_{\pm 14.76}$	35.31
	Entropy-E	**39.94**$_{\pm 3.25}$	**78.44**$_{\pm 6.63}$	**84.65**$_{\pm 18.12}$	**52.70**$_{\pm 13.35}$	**78.87**$_{\pm 9.18}$	**69.52**$_{\pm 11.82}$	**67.35**
	DetectGPT	$21.66_{\pm 7.94}$	$42.65_{\pm 6.28}$	$60.82_{\pm 6.22}$	$13.75_{\pm 3.45}$	$52.41_{\pm 7.71}$	$39.49_{\pm 4.15}$	38.46
	DetectGPT-E	**46.30**$_{\pm 2.60}$	**82.31**$_{\pm 4.48}$	**94.85**$_{\pm 0.92}$	**59.52**$_{\pm 4.78}$	**86.91**$_{\pm 3.54}$	**84.14**$_{\pm 2.65}$	**75.67**
	FastGPT	$21.20_{\pm 4.54}$	$47.15_{\pm 3.59}$	$66.61_{\pm 6.47}$	$20.63_{\pm 4.37}$	$52.58_{\pm 9.86}$	$49.92_{\pm 6.27}$	43.01
	FastGPT-E	**37.25**$_{\pm 4.38}$	**74.64**$_{\pm 4.96}$	**89.80**$_{\pm 5.01}$	**49.06**$_{\pm 2.23}$	**83.80**$_{\pm 4.03}$	**76.32**$_{\pm 5.07}$	**68.48**
	DNAGPT	**10.37**$_{\pm 3.51}$	$15.62_{\pm 4.68}$	**30.59**$_{\pm 5.19}$	$8.28_{\pm 2.27}$	$20.74_{\pm 5.36}$	**22.04**$_{\pm 1.97}$	**17.94**
	DNAGPT	$8.08_{\pm 2.57}$	**15.68**$_{\pm 6.48}$	$20.39_{\pm 3.96}$	**11.52**$_{\pm 6.76}$	**21.47**$_{\pm 10.11}$	$13.99_{\pm 6.07}$	15.19
AUROC	Likelihood	$84.67_{\pm 0.88}$	$91.95_{\pm 0.30}$	$97.08_{\pm 0.29}$	$79.28_{\pm 0.48}$	**95.65**$_{\pm 0.40}$	$94.17_{\pm 0.30}$	90.47
	Likelihood-E	**86.61**$_{\pm 0.56}$	**92.72**$_{\pm 2.67}$	**97.14**$_{\pm 0.92}$	**83.70**$_{\pm 0.57}$	$94.20_{\pm 1.87}$	**94.37**$_{\pm 0.92}$	**91.46**
	Log-Rank	$82.55_{\pm 0.80}$	$90.60_{\pm 0.30}$	$97.03_{\pm 0.31}$	$78.08_{\pm 0.70}$	$95.11_{\pm 0.44}$	$94.09_{\pm 0.38}$	89.57
	Log-Rank-E	**84.16**$_{\pm 0.43}$	**93.54**$_{\pm 0.60}$	**97.56**$_{\pm 0.46}$	**82.14**$_{\pm 0.91}$	**96.36**$_{\pm 0.58}$	**95.10**$_{\pm 0.51}$	**91.48**
	Entropy	$79.29_{\pm 0.53}$	$92.19_{\pm 0.39}$	$96.02_{\pm 0.49}$	$80.50_{\pm 0.40}$	$94.04_{\pm 0.88}$	$91.30_{\pm 0.58}$	88.89
	Entropy-E	**85.48**$_{\pm 0.50}$	**95.17**$_{\pm 0.79}$	**97.45**$_{\pm 1.27}$	**84.57**$_{\pm 0.93}$	**96.83**$_{\pm 0.88}$	**95.42**$_{\pm 0.84}$	**92.49**
	DetectGPT	**75.73**$_{\pm 1.30}$	$86.86_{\pm 0.70}$	$92.84_{\pm 0.66}$	$72.25_{\pm 1.87}$	$92.84_{\pm 1.08}$	$86.70_{\pm 0.65}$	84.54
	DetectGPT-E	$70.03_{\pm 0.91}$	**90.38**$_{\pm 1.10}$	**97.23**$_{\pm 0.41}$	**78.97**$_{\pm 0.76}$	**93.82**$_{\pm 1.07}$	**91.37**$_{\pm 0.42}$	**86.96**
	FastGPT	$82.61_{\pm 1.26}$	$90.99_{\pm 0.86}$	$96.26_{\pm 0.29}$	$81.63_{\pm 0.76}$	$94.38_{\pm 0.65}$	$91.84_{\pm 0.57}$	89.62
	FastGPT-E	**82.80**$_{\pm 0.75}$	**94.26**$_{\pm 0.28}$	**97.68**$_{\pm 0.68}$	**84.02**$_{\pm 0.99}$	**96.47**$_{\pm 0.41}$	**94.84**$_{\pm 1.08}$	**91.68**
	DNAGPT	$77.49_{\pm 1.00}$	$79.64_{\pm 1.25}$	$88.69_{\pm 0.51}$	$67.60_{\pm 0.40}$	$84.15_{\pm 1.03}$	$83.46_{\pm 0.91}$	80.17
	DNAGPT	**83.92**$_{\pm 0.38}$	**90.18**$_{\pm 0.76}$	**92.96**$_{\pm 0.50}$	**79.81**$_{\pm 1.07}$	**92.16**$_{\pm 0.66}$	**90.90**$_{\pm 0.46}$	**88.32**

Table 17: Performance on short texts on Essay. The detection models are trained on text generated by GPT4All.

Metric	Method	GPT4All	ChatGPT	Dolly	ChatGLM	Claude	ChatGPT-turbo	avg
TPR@FPR-1%	Likelihood	19.86$_{\pm 5.55}$	40.93$_{\pm 4.86}$	11.69$_{\pm 2.93}$	58.75$_{\pm 6.73}$	15.38$_{\pm 1.78}$	23.47$_{\pm 4.56}$	28.35
	Likelihood-E	**30.80**$_{\pm 6.67}$	**45.91**$_{\pm 7.98}$	**14.99**$_{\pm 4.58}$	**74.51**$_{\pm 4.38}$	**18.98**$_{\pm 5.26}$	**27.24**$_{\pm 6.18}$	**35.40**
	Log-Rank	15.85$_{\pm 2.24}$	36.49$_{\pm 6.97}$	9.98$_{\pm 1.81}$	55.58$_{\pm 5.15}$	11.33$_{\pm 3.00}$	19.07$_{\pm 5.24}$	24.72
	Log-Rank-E	**33.30**$_{\pm 2.53}$	**44.18**$_{\pm 4.07}$	**13.27**$_{\pm 3.77}$	**72.41**$_{\pm 2.13}$	**17.60**$_{\pm 1.81}$	**24.22**$_{\pm 2.48}$	**34.16**
	Entropy	0.77$_{\pm 0.31}$	4.62$_{\pm 2.03}$	1.43$_{\pm 1.14}$	1.70$_{\pm 1.13}$	**3.07**$_{\pm 1.77}$	**3.47**$_{\pm 1.83}$	2.51
	Entropy-E	**1.96**$_{\pm 0.42}$	**5.11**$_{\pm 1.82}$	**2.86**$_{\pm 1.17}$	**3.26**$_{\pm 1.79}$	2.89$_{\pm 1.79}$	3.11$_{\pm 1.63}$	**3.20**
	DetectGPT	8.25$_{\pm 2.34}$	13.42$_{\pm 2.70}$	7.16$_{\pm 3.80}$	18.57$_{\pm 4.32}$	9.73$_{\pm 4.84}$	8.58$_{\pm 2.09}$	10.95
	DetectGPT-E	**23.92**$_{\pm 7.43}$	**41.96**$_{\pm 6.95}$	**15.32**$_{\pm 3.89}$	**53.44**$_{\pm 7.22}$	**22.36**$_{\pm 5.76}$	**30.49**$_{\pm 5.42}$	**31.25**
	FastGPT	5.28$_{\pm 2.94}$	11.02$_{\pm 1.41}$	3.96$_{\pm 2.41}$	14.20$_{\pm 6.19}$	7.42$_{\pm 2.08}$	6.27$_{\pm 1.27}$	8.03
	FastGPT-E	**13.71**$_{\pm 9.14}$	**20.89**$_{\pm 1.26}$	**8.78**$_{\pm 3.14}$	**27.50**$_{\pm 4.97}$	**15.73**$_{\pm 1.90}$	**12.84**$_{\pm 2.35}$	**16.58**
	DNAGPT	8.02$_{\pm 1.30}$	20.04$_{\pm 4.42}$	10.02$_{\pm 2.96}$	35.98$_{\pm 5.07}$	6.84$_{\pm 1.77}$	11.11$_{\pm 2.36}$	15.34
	DNAGPT	**13.85**$_{\pm 1.23}$	**23.87**$_{\pm 4.14}$	**16.09**$_{\pm 5.47}$	**49.78**$_{\pm 4.61}$	**8.09**$_{\pm 1.80}$	**13.38**$_{\pm 2.56}$	**20.84**
AUROC	Likelihood	84.84$_{\pm 0.81}$	93.57$_{\pm 0.68}$	83.34$_{\pm 1.31}$	96.34$_{\pm 0.44}$	81.26$_{\pm 1.02}$	89.30$_{\pm 0.82}$	88.11
	Likelihood-E	**86.95**$_{\pm 0.80}$	**93.65**$_{\pm 0.76}$	**84.16**$_{\pm 1.29}$	**97.04**$_{\pm 0.32}$	**83.43**$_{\pm 0.74}$	**89.59**$_{\pm 0.87}$	**89.14**
	Log-Rank	82.90$_{\pm 0.94}$	92.29$_{\pm 0.84}$	81.18$_{\pm 1.11}$	95.97$_{\pm 0.52}$	77.58$_{\pm 0.92}$	87.83$_{\pm 0.92}$	86.29
	Log-Rank-E	**85.88**$_{\pm 0.83}$	**92.86**$_{\pm 0.78}$	**83.27**$_{\pm 1.08}$	**97.01**$_{\pm 0.40}$	**81.40**$_{\pm 0.71}$	**88.28**$_{\pm 0.93}$	**88.12**
	Entropy	58.86$_{\pm 1.57}$	72.56$_{\pm 1.03}$	62.13$_{\pm 0.86}$	66.71$_{\pm 1.77}$	67.08$_{\pm 1.13}$	73.88$_{\pm 0.80}$	66.87
	Entropy-E	**62.02**$_{\pm 1.48}$	**74.75**$_{\pm 1.22}$	**65.11**$_{\pm 1.67}$	**71.53**$_{\pm 1.72}$	**70.74**$_{\pm 1.19}$	**74.98**$_{\pm 1.01}$	**69.86**
	DetectGPT	78.67$_{\pm 1.06}$	84.93$_{\pm 0.84}$	76.56$_{\pm 0.97}$	89.40$_{\pm 0.75}$	79.27$_{\pm 0.74}$	81.54$_{\pm 0.96}$	81.73
	DetectGPT-E	**86.13**$_{\pm 1.27}$	**92.12**$_{\pm 0.78}$	**82.57**$_{\pm 1.32}$	**96.33**$_{\pm 0.63}$	**83.40**$_{\pm 0.90}$	**88.57**$_{\pm 0.75}$	**88.19**
	FastGPT	79.78$_{\pm 0.78}$	85.87$_{\pm 0.73}$	81.15$_{\pm 0.62}$	91.22$_{\pm 1.03}$	82.62$_{\pm 0.55}$	80.05$_{\pm 1.18}$	83.45
	FastGPT-E	**86.30**$_{\pm 0.72}$	**90.49**$_{\pm 0.69}$	**86.03**$_{\pm 0.77}$	**95.26**$_{\pm 0.78}$	**88.05**$_{\pm 0.61}$	**86.65**$_{\pm 0.68}$	**88.80**
	DNAGPT	79.59$_{\pm 0.70}$	86.99$_{\pm 1.18}$	79.45$_{\pm 1.14}$	90.86$_{\pm 0.80}$	76.84$_{\pm 1.03}$	83.48$_{\pm 1.08}$	82.87
	DNAGPT	**83.46**$_{\pm 0.91}$	**88.99**$_{\pm 1.03}$	**83.06**$_{\pm 1.30}$	**93.40**$_{\pm 0.58}$	**80.65**$_{\pm 0.95}$	**85.84**$_{\pm 1.02}$	**85.90**

Table 18: Enhancement results on more detectors in the Essay dataset. The detection models are trained on text generated by GPT4All.

Method	GPT4All	ChatGPT	Dolly	ChatGLM	Claude	ChatGPT-turbo	Avg.
RepreGuard	94.66$_{\pm 0.35}$	97.35$_{\pm 0.28}$	90.40$_{\pm 0.83}$	91.95$_{\pm 0.71}$	**47.51**$_{\pm 1.53}$	**43.42**$_{\pm 1.34}$	77.55
RepreGuard-M	**95.46**$_{\pm 0.80}$	**99.55**$_{\pm 0.18}$	**90.55**$_{\pm 0.70}$	**99.95**$_{\pm 0.04}$	47.18$_{\pm 1.47}$	43.12$_{\pm 1.28}$	**79.30**
Binoculars	96.33$_{\pm 0.27}$	98.59$_{\pm 0.18}$	88.57$_{\pm 1.56}$	**99.57**$_{\pm 0.43}$	**87.21**$_{\pm 0.43}$	**98.84**$_{\pm 0.18}$	94.85
Binoculars-M	**97.34**$_{\pm 0.13}$	**98.65**$_{\pm 0.19}$	**90.19**$_{\pm 1.38}$	97.68$_{\pm 2.43}$	86.80$_{\pm 0.45}$	98.82$_{\pm 0.18}$	**94.91**
Lastde	99.54$_{\pm 0.08}$	99.72$_{\pm 0.08}$	97.38$_{\pm 0.36}$	99.98$_{\pm 0.01}$	75.98$_{\pm 1.22}$	**91.16**$_{\pm 0.72}$	93.96
Lastde-M	**99.68**$_{\pm 0.07}$	**99.77**$_{\pm 0.07}$	**97.87**$_{\pm 0.35}$	**99.99**$_{\pm 0.01}$	**76.21**$_{\pm 1.29}$	90.96$_{\pm 0.75}$	**94.08**
FourierGPT	91.70$_{\pm 16.06}$	90.18$_{\pm 19.03}$	93.74$_{\pm 12.07}$	98.26$_{\pm 2.82}$	**44.90**$_{\pm 6.18}$	**42.11**$_{\pm 6.15}$	76.81
FourierGPT-M	**97.10**$_{\pm 4.19}$	**95.27**$_{\pm 6.46}$	**97.64**$_{\pm 3.19}$	**99.44**$_{\pm 0.58}$	41.14$_{\pm 6.81}$	41.08$_{\pm 6.90}$	**78.61**

Table 19: Enhancement results on more detectors in the Reuters dataset. The detection models are trained on text generated by GPT4All.

Method	GPT4All	ChatGPT	Dolly	ChatGLM	Claude	ChatGPT-turbo	Avg.
RepreGuard	97.21$_{\pm 0.29}$	**98.32**$_{\pm 0.30}$	8.24$_{\pm 0.46}$	**97.26**$_{\pm 0.27}$	21.81$_{\pm 0.62}$	44.32$_{\pm 1.80}$	61.19
RepreGuard-M	**98.33**$_{\pm 0.35}$	97.87$_{\pm 0.21}$	**48.12**$_{\pm 5.04}$	97.00$_{\pm 0.29}$	**29.08**$_{\pm 1.08}$	**50.59**$_{\pm 1.84}$	**70.16**
Binoculars	80.96$_{\pm 1.11}$	97.99$_{\pm 0.30}$	61.25$_{\pm 1.77}$	99.80$_{\pm 0.06}$	83.23$_{\pm 0.80}$	98.70$_{\pm 0.25}$	86.99
Binoculars-M	**90.90**$_{\pm 0.73}$	**98.49**$_{\pm 0.19}$	**74.10**$_{\pm 0.95}$	**99.86**$_{\pm 0.04}$	**86.37**$_{\pm 0.75}$	**98.72**$_{\pm 0.32}$	**91.41**
Lastde	99.21$_{\pm 0.18}$	99.67$_{\pm 0.04}$	94.25$_{\pm 0.63}$	**99.83**$_{\pm 0.07}$	83.27$_{\pm 0.49}$	**99.09**$_{\pm 0.18}$	95.89
Lastde-M	**99.55**$_{\pm 0.16}$	**99.68**$_{\pm 0.05}$	**96.09**$_{\pm 0.35}$	99.82$_{\pm 0.05}$	**85.64**$_{\pm 0.59}$	99.00$_{\pm 0.19}$	**96.63**
FourierGPT	99.28$_{\pm 0.35}$	96.71$_{\pm 3.25}$	99.06$_{\pm 0.41}$	98.45$_{\pm 0.25}$	**56.82**$_{\pm 10.56}$	49.84$_{\pm 0.89}$	83.36
FourierGPT-M	**99.36**$_{\pm 0.29}$	**98.29**$_{\pm 0.72}$	**99.29**$_{\pm 0.27}$	**98.48**$_{\pm 0.43}$	55.13$_{\pm 10.81}$	**49.88**$_{\pm 1.12}$	**83.40**

Table 20: Enhancement results on more detectors in the DetectRL dataset. The detection models are trained on text generated by GPT4All. Note that DetectRL lacks paired data for RepreGuard; therefore, this method is not compared.

Method	Llama-2-70b	ChatGPT	Google-PaLM	Avg.
Binoculars	79.10$_{\pm 0.37}$	66.55$_{\pm 0.80}$	73.85$_{\pm 0.93}$	73.17
Binoculars-M	**81.22**$_{\pm 0.39}$	**67.43**$_{\pm 1.08}$	**77.80**$_{\pm 0.79}$	**75.49**
Lastde	78.36$_{\pm 0.39}$	51.93$_{\pm 0.51}$	69.04$_{\pm 0.90}$	66.45
Lastde-M	**78.70**$_{\pm 0.36}$	**52.55**$_{\pm 0.43}$	**71.06**$_{\pm 0.76}$	**67.44**
FourierGPT	58.11$_{\pm 7.29}$	50.57$_{\pm 3.63}$	59.99$_{\pm 8.53}$	56.23
FourierGPT-M	**64.67**$_{\pm 8.42}$	**52.64**$_{\pm 5.70}$	**64.07**$_{\pm 7.25}$	**60.46**

Table 21: Performance (AUROC) on perturbation texts (Back-Translation and DeepWordBug) on DetectRL. All detectors are trained on Llama-2-70b texts.

Perturbation	Method	Llama-2-70b	ChatGPT	Google-PaLM	Avg.
Back-Translation	Likelihood	$53.88_{\pm 0.76}$	**29.91**$_{\pm 1.34}$	$48.00_{\pm 0.58}$	43.93
	Likelihood-M	**58.85**$_{\pm 0.95}$	$29.71_{\pm 1.26}$	**55.80**$_{\pm 0.90}$	**48.12**
	Log-Rank	$56.40_{\pm 0.74}$	$30.91_{\pm 1.32}$	$46.85_{\pm 0.75}$	44.72
	Log-Rank-M	**62.41**$_{\pm 0.61}$	**32.17**$_{\pm 1.19}$	**59.55**$_{\pm 0.69}$	**51.37**
	Entropy	$46.43_{\pm 1.06}$	**32.54**$_{\pm 1.08}$	$41.99_{\pm 0.61}$	40.32
	Entropy-M	**48.99**$_{\pm 1.33}$	$31.64_{\pm 1.18}$	**46.16**$_{\pm 0.49}$	**42.26**
	DetectGPT	$20.30_{\pm 0.72}$	$10.90_{\pm 0.68}$	$16.59_{\pm 0.56}$	15.93
	DetectGPT-M	**62.24**$_{\pm 2.17}$	**38.63**$_{\pm 6.75}$	**61.72**$_{\pm 4.74}$	**54.20**
	FastGPT	$35.37_{\pm 1.40}$	**24.10**$_{\pm 1.08}$	$31.23_{\pm 0.57}$	30.23
	FastGPT-M	**51.87**$_{\pm 3.25}$	$21.35_{\pm 3.21}$	**53.16**$_{\pm 3.04}$	**42.13**
	DNA-GPT	$70.82_{\pm 0.84}$	**55.34**$_{\pm 0.73}$	$65.37_{\pm 0.51}$	63.84
	DNA-GPT-M	**71.83**$_{\pm 0.94}$	$54.53_{\pm 1.32}$	**67.63**$_{\pm 0.62}$	**64.67**
DeepWordBug	Likelihood	$69.62_{\pm 0.71}$	$51.56_{\pm 0.73}$	$62.19_{\pm 1.42}$	61.12
	Likelihood-M	**85.71**$_{\pm 1.22}$	**68.11**$_{\pm 1.64}$	**79.18**$_{\pm 0.63}$	**77.67**
	Log-Rank	$72.76_{\pm 0.64}$	$50.60_{\pm 0.44}$	$62.64_{\pm 1.86}$	62
	Log-Rank-M	**90.93**$_{\pm 1.14}$	**71.16**$_{\pm 2.75}$	**83.41**$_{\pm 1.12}$	**81.84**
	Entropy	$65.85_{\pm 0.97}$	$63.63_{\pm 1.71}$	$58.00_{\pm 1.31}$	62.49
	Entropy-M	**69.35**$_{\pm 0.95}$	**66.54**$_{\pm 1.58}$	**63.95**$_{\pm 1.25}$	**66.61**
	DetectGPT	$34.33_{\pm 2.82}$	$33.28_{\pm 3.72}$	$32.72_{\pm 2.14}$	33.44
	DetectGPT-M	**82.95**$_{\pm 1.70}$	**64.13**$_{\pm 2.85}$	**79.67**$_{\pm 2.41}$	**75.58**
	FastGPT	$39.57_{\pm 2.14}$	**34.59**$_{\pm 2.49}$	$36.51_{\pm 2.30}$	36.89
	FastGPT-M	**53.71**$_{\pm 1.90}$	$30.93_{\pm 2.38}$	**54.63**$_{\pm 2.88}$	**46.42**
	DNA-GPT	$67.76_{\pm 2.07}$	$58.74_{\pm 1.64}$	$63.69_{\pm 1.06}$	63.4
	DNA-GPT-M	**71.43**$_{\pm 6.30}$	**59.59**$_{\pm 4.16}$	**67.61**$_{\pm 4.09}$	**66.21**

Table 22: Performance and running time comparison with HMM-based method on Essay. The detection models are traind on text generated by GPT4All.

Method	Detection Performance							Running Time	
	GPT4All	ChatGPT	Dolly	ChatGLM	Claude	ChatGPT-turbo	Avg.	Train	Inference
Likelihood	96.16	98.79	90.90	99.29	92.76	99.13	96.17	13.58	61.21
Likelihood-HMM	96.39	98.87	91.24	99.33	92.91	99.17	96.32	20.64	90.63
Likelihood-M	**98.58**	**99.47**	**94.59**	**99.54**	**94.82**	**99.72**	**97.79**	15.01	72.67
Log-Rank	96.55	98.95	90.08	99.36	92.01	99.24	96.03	15.98	75.60
Log-Rank-HMM	96.68	98.99	90.24	99.38	92.08	99.27	96.10	21.77	103.70
Log-Rank-M	**98.57**	**99.41**	**93.82**	**99.55**	**92.91**	**99.64**	**97.32**	17.40	87.55
Entropy	74.19	89.49	73.26	84.11	86.58	95.94	83.93	13.26	64.49
Entropy-HMM	74.47	89.70	73.56	84.39	86.61	95.98	84.12	16.87	88.26
Entropy-M	**83.52**	**93.28**	**81.11**	**91.44**	**87.96**	**96.75**	**89.01**	14.69	73.66

Table 23: Performance (AUROC) on TOCSIN-enhanced baselines and our method on top of TOCSIN (suffix "-TM"). Detectors are trained on GPT4All texts.

Method	GPT4	ChatGPT-turbo	ChatGLM	Dolly	ChatGPT	StableLM	Avg.
Likelihood-T	$90.43_{\pm 1.03}$	$98.66_{\pm 0.19}$	$89.50_{\pm 0.79}$	$98.52_{\pm 0.55}$	$90.46_{\pm 0.71}$	$98.44_{\pm 0.31}$	94.33
Likelihood-TM	**98.08**$_{\pm 0.19}$	**99.19**$_{\pm 0.15}$	**92.83**$_{\pm 1.06}$	**99.01**$_{\pm 0.51}$	**91.06**$_{\pm 1.26}$	**99.20**$_{\pm 0.09}$	**96.56**
Log-Rank-T	$93.96_{\pm 0.46}$	$98.78_{\pm 0.17}$	$89.90_{\pm 0.83}$	$98.69_{\pm 0.52}$	$91.01_{\pm 0.48}$	$98.57_{\pm 0.27}$	95.15
Log-Rank-TM	**98.07**$_{\pm 0.35}$	**99.10**$_{\pm 0.16}$	**92.32**$_{\pm 0.91}$	**99.01**$_{\pm 0.43}$	**91.46**$_{\pm 0.62}$	**99.05**$_{\pm 0.13}$	**96.50**
Entropy-T	$76.90_{\pm 1.57}$	**98.41**$_{\pm 0.25}$	$87.20_{\pm 0.92}$	$98.11_{\pm 0.61}$	**90.91**$_{\pm 0.40}$	**98.72**$_{\pm 0.19}$	91.71
Entropy-TM	**83.89**$_{\pm 0.82}$	$98.34_{\pm 0.35}$	**87.23**$_{\pm 0.96}$	**98.25**$_{\pm 0.53}$	$89.70_{\pm 0.44}$	$98.63_{\pm 0.19}$	**92.67**
DetectGPT-T	$48.95_{\pm 1.79}$	$40.69_{\pm 11.74}$	$51.46_{\pm 2.98}$	$40.67_{\pm 13.55}$	$43.99_{\pm 7.23}$	$27.04_{\pm 30.41}$	42.13
DetectGPT-TM	**95.51**$_{\pm 2.81}$	**93.27**$_{\pm 9.08}$	**79.76**$_{\pm 14.11}$	**97.67**$_{\pm 2.00}$	**70.25**$_{\pm 11.65}$	**93.59**$_{\pm 10.94}$	**88.34**
FastGPT-T	$53.53_{\pm 5.59}$	**70.36**$_{\pm 27.49}$	$56.26_{\pm 8.54}$	$71.83_{\pm 28.44}$	**72.06**$_{\pm 29.44}$	**77.26**$_{\pm 36.41}$	**66.88**
FastGPT-TM	**87.73**$_{\pm 0.15}$	$66.40_{\pm 1.22}$	**80.65**$_{\pm 0.81}$	**75.36**$_{\pm 1.49}$	$41.91_{\pm 1.52}$	$28.10_{\pm 2.13}$	63.36
DNAGPT-T	$98.32_{\pm 0.21}$	$97.97_{\pm 0.19}$	$96.60_{\pm 0.27}$	$98.34_{\pm 0.17}$	$95.13_{\pm 0.46}$	$97.93_{\pm 0.18}$	97.38
DNAGPT-TM	**99.56**$_{\pm 0.10}$	**99.04**$_{\pm 0.18}$	**97.70**$_{\pm 0.33}$	**99.24**$_{\pm 0.11}$	**96.30**$_{\pm 0.40}$	**99.06**$_{\pm 0.19}$	**98.48**

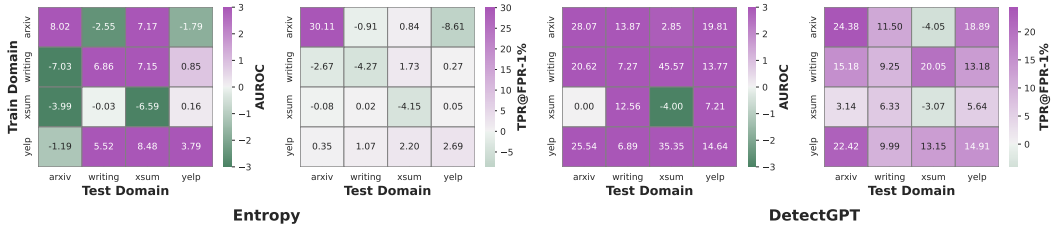

Figure 21: The performance improvement of the proposed method on Entropy and DetectGPT. Values greater than 0 indicate an enhanced effect.

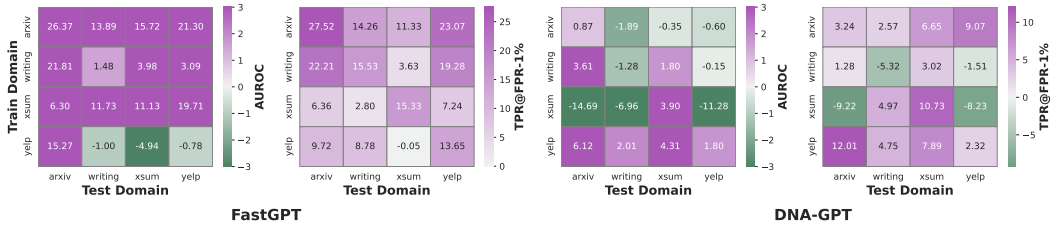

Figure 22: The performance improvement of the proposed method on FastGPT and DNA-GPT. Values greater than 0 indicate an enhanced effect.

technical content. We ensure that the use of large language models is responsible and adheres to academic and ethical standards.

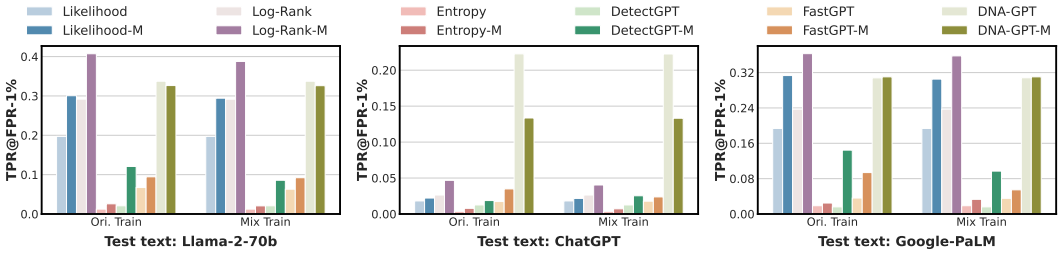

Figure 23: Detection performance concerning TPR@FPR-1% under different LLM mixed texts. All detectors are trained on Llama-2-70b texts.

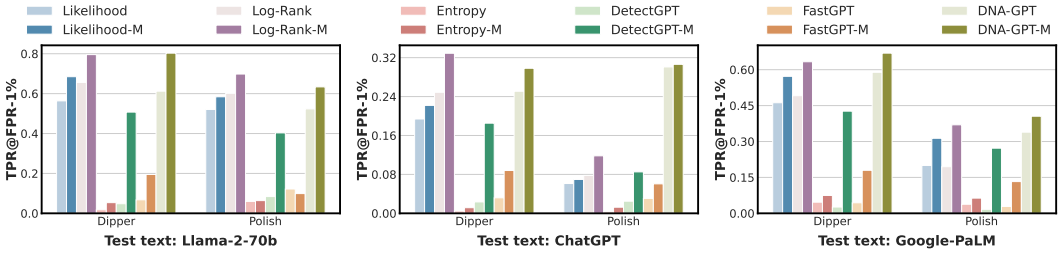

Figure 24: Detection performance concerning TPR@FPR-1% under different paraphrasing texts (Dipper and Polish). All detectors are trained on Llama-2-70b texts.

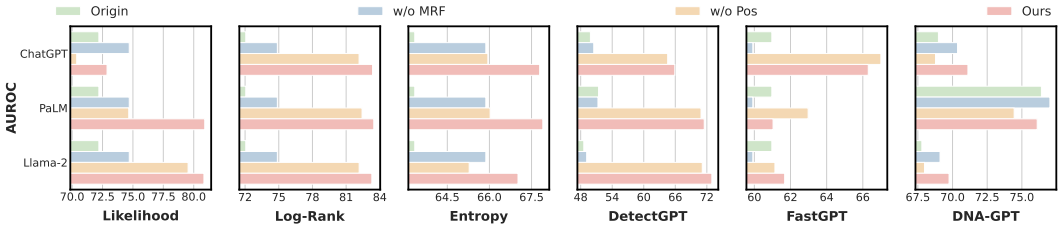

Figure 25: Ablation results on the DetectRL dataset. The y-axis represents the LLM text on which the detector was trained, and the x-axis represents the average performance across LLMs.

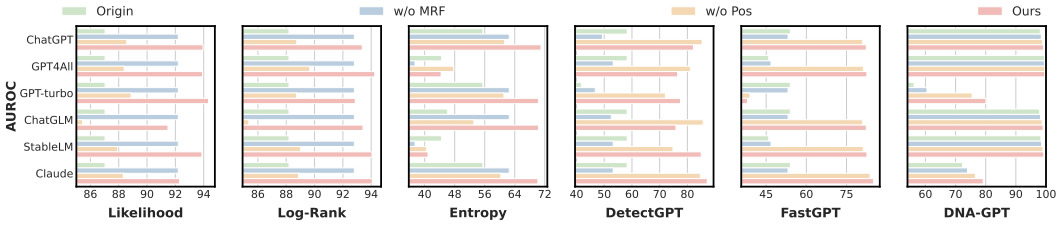

Figure 26: Ablation results on the Reuters dataset. The y-axis represents the LLM text on which the detector was trained, and the x-axis represents the average performance across LLMs.

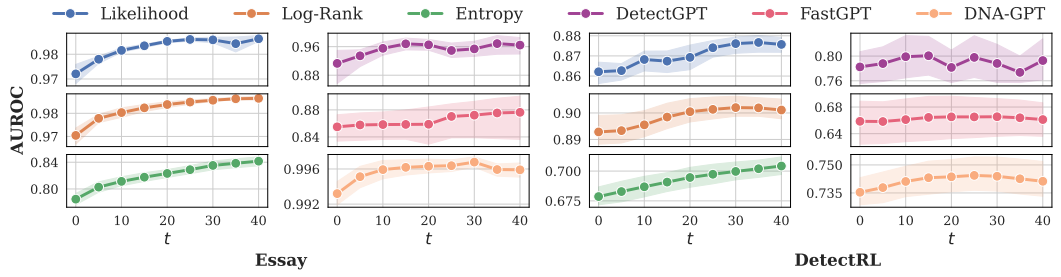

Figure 27: Detection performance of AUROC under different transition center t_0.

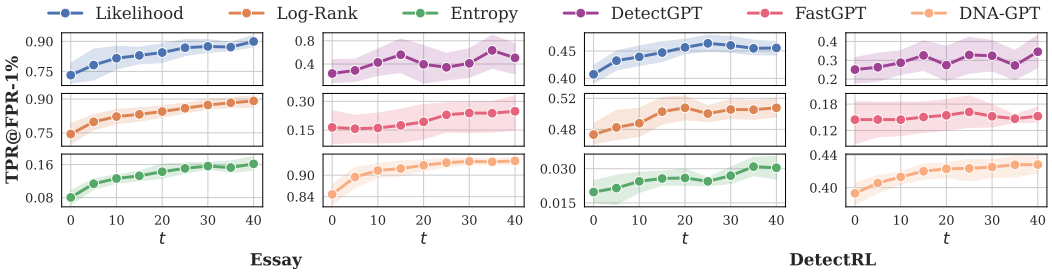

Figure 28: Detection performance of TPR@FPR-1% under different transition center t_0.

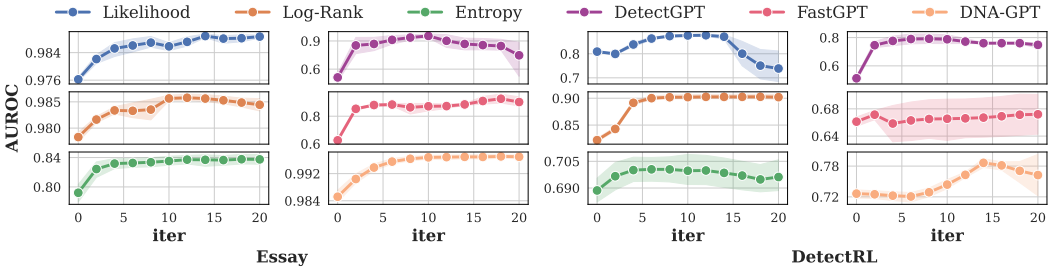

Figure 29: Detection performance of AUROC at different numbers of MRF layer iterations.

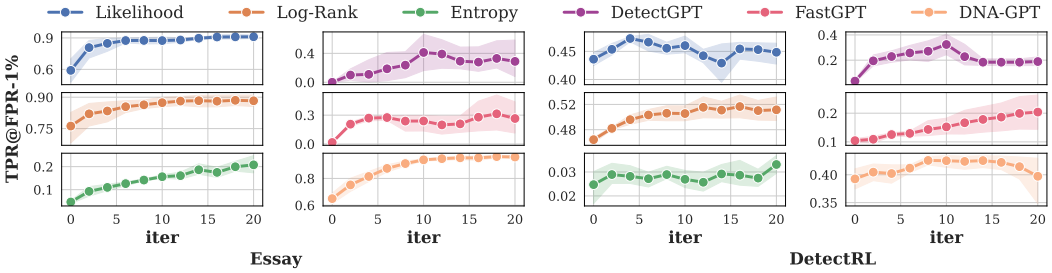

Figure 30: Detection performance of TPR@FPR-1% at different numbers of MRF layer iterations.

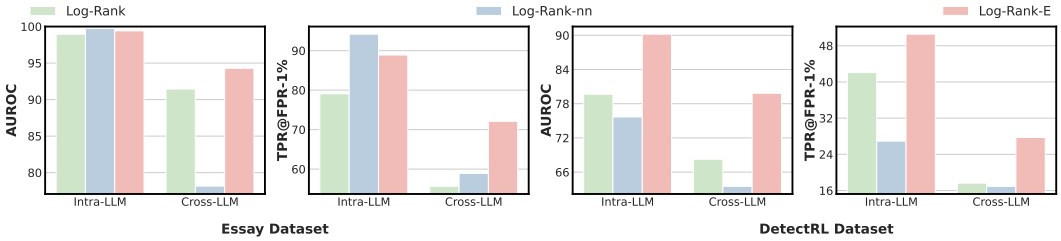

Figure 31: Comparison with NN-based methods. The detector used is Log-Rank.

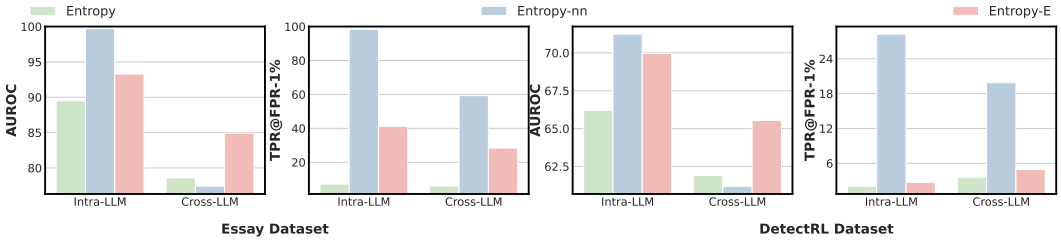

Figure 32: Comparison with NN-based methods. The detector used is Entropy.

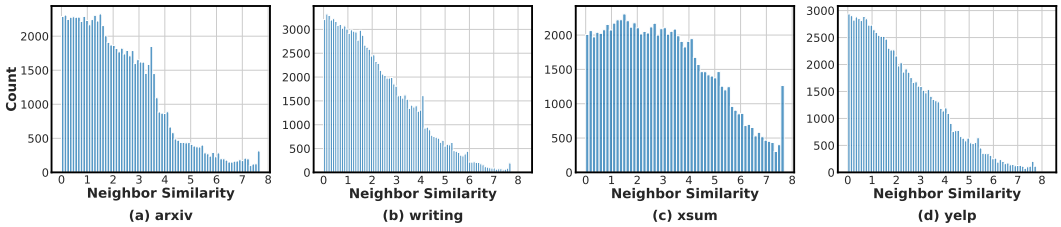

Figure 33: Statistical information on the distance of neighbor detection scores.

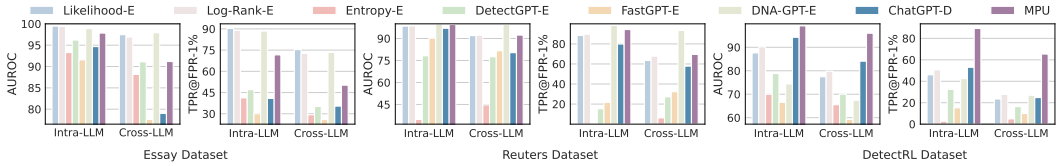

Figure 34: Comparison with Model-based detectors. Detectors are trained on GPT4All (Essay and Reuters) and Llama-2-70b (DetectRL).

Table 24: Running time (s) of training and inference phases.

Method	Train			Inference		
	Essay	Reuters	DetectRL	Essay	Reuters	DetectRL
Likelihood	13.58	12.77	12.40	61.21	61.05	59.34
Likelihood-M	15.01	14.19	13.10	72.67	72.38	61.66
Log-Rank	15.98	14.38	14.61	75.60	72.99	66.20
Log-Rank-M	17.40	15.81	16.29	87.55	72.06	82.21
Entropy	13.26	12.93	13.79	64.49	63.69	63.19
Entropy-M	14.69	14.38	15.61	73.66	72.80	72.63
DetectGPT	204.32	216.36	370.44	924.63	982.55	1672.85
DetectGPT-M	206.44	218.73	373.48	929.46	984.70	1682.78
FastGPT	60.21	58.67	52.97	279.50	272.18	240.20
FastGPT-M	63.00	61.47	55.76	286.44	284.75	254.85
DNA-GPT	469.10	532.02	267.84	2113.65	2398.61	1207.18
DNA-GPT-M	471.42	534.47	271.11	2125.17	2405.37	1221.74