LM^{2}OTIFS: An Explainable Framework for Machine-Generated Texts Detection

Xu Zheng[1] **Zhuomin Chen**[1] **Esteban Schafir**[1] **Sipeng Chen**[1] **Hojat Allah Salehi**[1]
Haifeng Chen[2] **Farhad Shirani**[1] **Wei Cheng**[2] **Dongsheng Luo**[1]
[1]Florida International University [2]NEC Laboratories America
{xzhen019,zchen051,escha032,schen131,hsalehi}@fiu.edu
{haifeng,weicheng}@nec-labs.com, {fshirani,dluo}@fiu.edu

Abstract

The impressive ability of large language models to generate natural text across various tasks has led to critical challenges in authorship authentication. Although numerous detection methods have been developed to differentiate between machine-generated texts (MGT) and human-generated texts (HGT), the explainability of these methods remains a significant gap. Traditional explainability techniques often fall short in capturing the complex word relationships that distinguish HGT from MGT. To address this limitation, we present LM^{2}OTIFS, a novel explainable framework for MGT detection. Inspired by probabilistic graphical models, we provide a theoretical rationale for the effectiveness. LM^{2}OTIFS utilizes eXplainable Graph Neural Networks to achieve both accurate detection and interpretability. The LM^{2}OTIFS pipeline operates in three key stages: first, it transforms text into graphs based on word co-occurrence to represent lexical dependencies; second, graph neural networks are used for prediction; and third, a post-hoc explainability method extracts interpretable motifs, offering multi-level explanations from individual words to sentence structures. Extensive experiments on multiple benchmark datasets demonstrate the comparable performance of LM^{2}OTIFS. The empirical evaluation of the extracted explainable motifs confirms their effectiveness in differentiating HGT and MGT. Furthermore, qualitative analysis reveals *distinct and visible linguistic fingerprints* characteristic of MGT.

1 Introduction

Large Language Models (LLMs) have made remarkable progress in recent years, demonstrating the ability to generate text based on prompt instructions [14]. Models like ChatGPT [43], Llama [56], and Claude-3 [3] have shown impressive capabilities in writing [73], coding [80], and question answering [82]. However, these advances raise serious concerns about content authenticity, including fake news [1], plagiarism [32], and misinformation [10]. Given that humans struggle to identify machine-generated texts (MGT) [19], developing reliable detectors to distinguish between MGT and human-generated texts (HGT) has become essential.

Existing LLM detectors [69, 42, 22, 9] are broadly categorized as white-box and black-box approaches. White-box approaches, exemplified by DetectLLM [52], analyze the probabilities of the output token to identify distinguishing characteristics [72]. In contrast, black-box methods [22, 51, 77, 42, 9] achieve detection without access to the LLM's internal workings. Despite their effectiveness, significant challenges persist in creating detectors that are both robust and explainable [64]. Furthermore, these methods typically only output a binary classification. However, practical applications demand supporting evidence for such judgments, such as the need for evidence in a court trial. However, existing explainability techniques for these detectors are inadequate. Traditional methods like Inte-

Preprint.

grated Gradients [53] are computationally prohibitive for LLM-based detectors, and while attention mechanisms [26, 63] excel at capturing local dependencies, they often fail to identify global patterns crucial for an LLM. Consequently, developing an explainable detector solution is critical and timely.

The fundamental architecture of modern LLM builds upon the principle of autoregressive next-token prediction, which models the joint probability distribution of a sequence as $P(s_1, s_2, \cdots, s_T) \approx \prod_{t=1}^{T} P_\theta(s_t|s_{1:t-1})$, where θ is the (trainable) model parameter, s_i is the word/token at the ith position, and T is bounded by the context length [45, 7]. Following this notion, in MGT detection, current methods typically treat the input as sequential data, and measure the distance between its posterior distribution and reference distributions for MGT and HGT samples — for instance by estimating the Kullback-Leibler (KL) divergence. This often requires substantial computational resources and large sample sizes. However, an intuitive and efficient alternative, probabilistic graphical models (PGM) [8, 31], to model conditional probabilities, has been largely overlooked. From the perspective of PGM, while generation tasks require that LLM operate based on probability graphs which accurately approximate the ground-truth posterior distribution, detection tasks only require constructing and analyzing probability graphs that are sufficiently discriminative for the underlying detection task. With sufficient sample data, building such graphs is straightforward. Furthermore, by analyzing the mechanism between sequential-based detectors and graph-based detectors, we provide the advantage of graph-based detectors in theory. In practice, PGM has advantages in terms of explainability, inference speed, and detection accuracy.

Drawing inspiration from PGM, we introduce a novel explainable framework, LM^{2}OTIFS. Beyond classifying input text as either MGT or HGT, LM^{2}OTIFS generates explanatory motifs that justify its detection outcome. LM^{2}OTIFS consists of three key parts: i) Graph Construction, ii) MGT Detection, and iii) Explainable Motifs Extraction. In the first stage, we leverage the word co-occurrence techniques to capture the lexical dependencies. To extract meaningful patterns at multiple levels (e.g., words and phrases), we integrate mainstream eXplainable Graph Neural Networks (XGNNs) to generate these motifs. To validate the effectiveness of our PGM-inspired approach, we empirically demonstrate that LM^{2}OTIFS achieves competitive performance with state-of-the-art MGT detection methods, including both supervised and zero-shot approaches. Following eXplainable AI (XAI) protocols, we verify the effectiveness of LM^{2}OTIFS. Our results indicate that the generated explainable motifs significantly outperform the baseline in terms of interpretability. The main contributions of this paper are summarized as follows:

- We introduce LM^{2}OTIFS, an explainable framework for MGT detection that integrates co-occurrence graphs with XGNN techniques for both accurate detection and explainable motifs extraction.
- We provide a theoretical analysis of the rationale and advantages of employing GNN for this task, drawing insights from the perspective of PGM.
- We conduct comprehensive experiments on diverse datasets, validating the effectiveness of LM^{2}OTIFS in MGT detection. Our analysis following XAI protocols supports the correctness of the extracted explainable motifs.

2 Preliminary

Probabilistic Graphical Models. PGM offers an efficient framework for representing probabilistic models, incorporating insightful properties such as conditional independence. Given a graph $G = \{\mathcal{V}, \mathcal{E}\}$, the nodes $\mathcal{V}$ correspond to random variables, and the links $\mathcal{E}$ capture probabilistic dependencies between these variables. For example, given a sequence of three tokens $\mathbf{S} = (s_1, s_2, s_3)$, the joint distribution is $P(s_1, s_2, s_3) = P(s_3|s_1, s_2)P(s_2|s_1)P(s_1)$. This can be represented using a graph with $\mathcal{V} = \{s_1, s_2, s_3\}$ and $\mathcal{E} = \{(s_1, s_2), (s_1, s_3), (s_2, s_3)\}$ as illustrated in Figure 1. More generally, for any sequence of tokens, a PGM can be constructed to represent the probabilistic dependencies among tokens.

MGT Detection. The MGT detection problem can be formulated as a classification task. Take an example of a binary hypothesis testing task. Given a pair of training sets,

$$\mathcal{T}_h = \{\mathbf{S}_{h,i} = (S_{h,i,1}, S_{h,i,2}, \cdots, S_{h,i,L_i})\}_{i \in |\mathcal{T}_h|},$$
$$\mathcal{T}_m = \{\mathbf{S}_{m,i} = (S_{m,i,1}, S_{m,i,2}, \cdots, S_{m,i,L'_i})\}_{i \in |\mathcal{T}_m|},$$

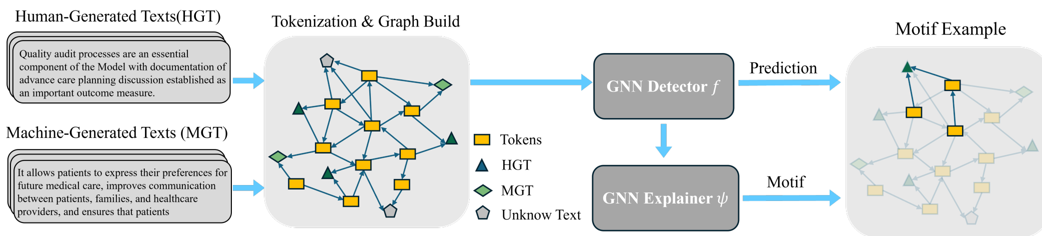


Figure 2: Overall pipeline of our framework, including tokenization, graph building, detector training, and motifs extraction.

consisting of human-generated and machine-generated text sequences, respectively, drawn from the distributions[1] P_h and P_m, the objective is to classify a newly observed text sequence $\mathbf{S}_o$ as either human-generated or machine-generated. A detection mechanism is a function $f : (\mathcal{T}_h, \mathcal{T}_m, \mathbf{S}_o) \mapsto \widehat{Y}$, where $\widehat{Y} \in \{0, 1\}$, the index 0 represents the null hypothesis (human generated) and 1 represents the alternative hypothesis (machine generated). The detection error is quantified by the risk function $P(f(\mathcal{T}_h, \mathcal{T}_m, \mathbf{S}_o) \neq Y)$, where $Y \in \{0, 1\}$ denotes the ground-truth hypothesis label.

Node Classification. A graph G consists of a set of nodes $\mathcal{V} = \{v_1, v_2, \cdots, v_n\}$, where $n \in \mathbb{N}$, and a set of edges $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. The adjacency matrix $\boldsymbol{A} \in \{0, 1\}^{n\times n}$ encodes the graph edges, where $A_{i,j} = \mathbb{1}((v_i, v_j) \in \mathcal{E})$. Each node may be associated with a feature vector, collectively represented by the matrix $\boldsymbol{X} \in \mathbb{R}^{n\times d}$, where the i-th row is the feature vector associated with the i-th node, and $d \in \mathbb{N}$ is the dimension. Furthermore, each edge is associated with a feature vector, collectively represented by the matrix $\boldsymbol{I} \in \mathbb{R}^{m\times c}$, where the i-th row is the feature vector associated with the i-th edge, and c is the dimension. Each node v is related to a label $Y_v \in \mathcal{Y}$, where $\mathcal{Y}$ is the collection of possible labels. In this work, we reformulate the author detection problem as a node classification task. This reformulation is elaborated on in the subsequent sections. The objective in node classification is to train a classifier $f : (\mathbf{G}, \boldsymbol{X}, \boldsymbol{I}, v) \mapsto \widehat{Y}_v$, which, given a graph $\mathbf{G}$, node and edge feature matrices $\boldsymbol{X}, \boldsymbol{I}$, and a node index v, produces an estimate $\widehat{Y}_v$ of the node label Y_v. The accuracy of the classifier is defined as $P_{V,\mathbf{G},\boldsymbol{X},\boldsymbol{I},Y_V}(f(V, \mathbf{G}, \boldsymbol{X}, \boldsymbol{I}) \neq Y_V)$, where V is uniformly distributed over $\mathcal{V}$, and $\mathbf{G}, \boldsymbol{X}, \boldsymbol{I}, Y_V$ follow a joint distribution $P_{\mathbf{G},\boldsymbol{X},\boldsymbol{I},Y_V}$.

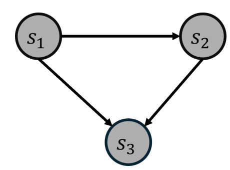


Figure 1: A PGM example of a three-token text.

Post-hoc Explainable Graph Neural Networks. Given a graph or node classification task, the goal of XGNN is to find an explanation function $\Psi(\cdot)$, which maps the input graph G to a *minimal* and *sufficient* explanation subgraph G_{exp}. Minimality restricts the size of the explanatory subgraph and is enforced by the constraint $|G_{exp}| \leq s \cdot |G|$, where $|G|$ denotes the number of edges in G and $s \in [0, 1]$ is the size parameter. Sufficiency is quantified by the KL divergence term $d_{KL}(P_{Y|G,\boldsymbol{X},\boldsymbol{I},V}||P_{Y|G_{exp},\boldsymbol{X},\boldsymbol{I},V})$. The explainer is optimally sufficient if it minimizes the KL divergence subject to minimality constraints. That is, given $s \in [0, 1]$, an *optimal* explainer Ψ^* is defined as:

$$\Psi^*(G) = \underset{\Psi:|G_{exp}|\leq s|G|}{\arg\min}\ d_{KL}(P_{Y|G,\boldsymbol{X},\boldsymbol{I},V}||P_{Y|G_{exp},\boldsymbol{X},\boldsymbol{I},V}) \tag{1}$$

3 Methodology

In this section, drawing upon the theoretical foundations of PGM in the prequel, we present the practical implementation of our probabilistic graph-based (PGB) detector framework, LM^{2}OTIFS. Our implementation encompasses three key components: graph construction based on token co-occurrences, GNN-based authorship detection, and explainable motif extraction. The complete pipeline of LM^{2}OTIFS is illustrated in Figure 2.

[1]The length of the observed text sequences is not fixed and can be modeled as a random variable. This variability is implicitly captured in the distributions P_h and P_m.

3.1 Graph Construction

Following our PGB framework, we implement an efficient graph construction method based on co-occurrence principles from TextGCN [70]. Our graph consists of two types of nodes representing tokens and documents, corresponding to the node sets $\mathcal{S}$ and $\mathcal{D}$. As specified in our framework, tokens are initialized with one-hot features and documents with zero vectors. To construct edges that capture textual relationships, we consider both document-token connections and token co-occurrences. The adjacency matrix A is defined as:

$$A_{ij} = \begin{cases} 1 & i, j \text{ are token}, \mathrm{PMI}(i,j) > 0 \\ 1 & j \text{ is document}, i \text{ is token in } j \\ 1 & i = j \\ 0 & \text{otherwise} \end{cases}, \qquad (2)$$

where $\mathrm{PMI}(i,j) = \log \frac{p(i,j)}{p(i)p(j)}$, point-wise mutual information, is used to determine significant token co-occurrences. Here, $p(i)$ represents the frequency of the i-th token within a fixed-length sliding window, and $p(i,j)$ denotes the co-occurrence frequency of tokens i and j. As discussed in Section 4, in the most general sense, the edge weights may be continuous-valued, and generated using a learnable function. However, our experimental evaluation shows that the above binary-valued edge weights are sufficient for reliable detection.

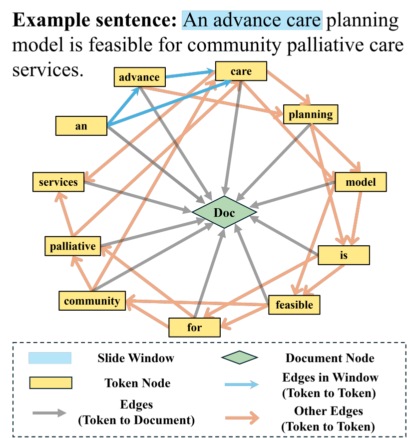


Figure 3: An example of graph construction with a fixed slid window 3.

3.2 GNN Detection

Having constructed the graph structure, we implement the detection mechanism outlined in our framework through a GNN architecture. For a given text sequence $\mathbf{S}_o$, our goal is to learn a function f that determines whether the text is machine-generated or human-authored. This corresponds to the PGB detector operating over K message passing rounds. Each GNN layer implements one round of message passing, with the update rule:

$$a_v^{(l)} = \mathrm{AGG}^{(l)}\left(h_u^{(l-1)} : u \in \mathcal{N}(v)\right),$$
$$h_v^{(l)} = \mathrm{COMBINE}^{(l)}\left(h_v^{(l-1)}, a_v^{(l)}\right),$$

where $a_v^{(l)}$ represents the aggregated message at layer l, $h_v^{(l)}$ is the node feature vector, and $\mathcal{N}(v)$ denotes the neighbors of node v. The $\mathrm{AGG}^{(l)}$ function aggregates information from neighboring nodes, while $\mathrm{COMBINE}^{(l)}$ updates the nodes' representation. After K layers, we obtain the final node embeddings $\boldsymbol{H}$. For classification, we apply a softmax function to the final embeddings to obtain prediction probabilities $\boldsymbol{Z} = \mathrm{softmax}(\boldsymbol{H})$. The model is trained by minimizing the cross-entropy loss over labeled document nodes:

$$\mathcal{L} = -\sum_{d \in \mathcal{Y}_D} \sum_{\ell \in \{h,m\}} Y_{d\ell} \ln Z_{d\ell}, \qquad (3)$$

where $\mathcal{Y}_D$ represents the set of document nodes in the training set and $Y_{d\ell}$ is the ground-truth label, While our goal focuses on binary classification (human-authored vs. machine-generated) in this paper, the framework naturally extends to scenarios with multiple classes, such as texts generated by different language models.

3.3 Explainable Motifs Extraction

Beyond detection accuracy, our framework provides interpretable insights through the extraction of distinguishing motifs between machine-generated and human-authored texts. While existing detection methods often operate as black boxes [22, 77], our graph-based approach naturally enables the identification of characteristic patterns through subgraph structures [31]. Drawing inspiration from graph analysis techniques [71, 37], we transform the interpretability challenge into a subgraph identification problem, where meaningful token dependencies in our constructed graph serve as distinguishing motifs. These motifs capture characteristic patterns of word usage and dependencies that differentiate between human and machine-generated content [28], providing insights beyond simple token-level statistics.

We use the GNNExplainer [71] to extract meaningful motifs. Specifically, we formulate a practical optimization objective using cross-entropy loss and explicit size constraints. The objective function balances the prediction accuracy of the explanation subgraph against its complexity:

$$\Psi^*(\cdot) = \underset{\Psi: G \mapsto G_{exp}}{\arg\min} \ \mathrm{CE}(Y; f(G_{exp})) + \lambda |G_{exp}| \tag{4}$$

where $\mathrm{CE}(Y; f(G_{exp}))$ measures how well the explainer preserves the model's prediction capability, $|G_{exp}|$ denotes the size of the explanation subgraph, and λ controls the trade-off between explanation fidelity and complexity. This formulation is an approximation of the theoretical requirements from Equation 1, where the cross-entropy term ensures sufficiency and the size penalty enforces minimality. The optimization is performed through gradient descent, with the edge weights of G_{exp} being learned continuously and then discretized through thresholding.

4 Theoretical Analysis

As discussed in the prequel, prior works in MGT detection, such as Fast-DetectGPT [6], have employed sequential data models to design detection mechanisms. Drawing inspiration from TextGCN [70], we formulate the MGT detection problem using a graph-based approach where both tokens and documents are represented as nodes. Building upon this foundation, we demonstrate that GNN-based detectors achieve strictly improved detection accuracy compared to such approaches. This section provides theoretical justifications for this claim. The subsequent sections provide further verification through empirical analysis over several benchmark datasets.

We formally define a class of baseline *empirical sequential-based* (ESB) detectors that capture the essential characteristics of existing approaches. An ESB detector operates in two steps. First, it uses the human-generated training set $\mathcal{T}_h$ to construct the empirical conditional distribution estimates $\widehat{P}_h(s_t|s_{1:t-1})$ for human-generated text sequences, where $t \in [T]$, and T is a hyperparameter capturing the maximum context length. Similarly, the empirical estimates $\widehat{P}_m(s_t|s_{1:t-1})$ are computed based on the machine generated training set $\mathcal{T}_m$. In the second step, the detector uses (a potentially trainable) mapping $g_s : ((\widehat{P}_h(s_t|s_{1:t-1}), \widehat{P}_m(s_t|s_{1:t-1}))_{t\in[T]}, \mathbf{S}_o) \mapsto \widehat{Y}$, where $\mathbf{S}_o$ is the to-be-classified sequence. An ESB detector is completely characterized by the mapping $g_s(\cdot)$. We denote the collection of ESB detectors by $\mathcal{F}_{\text{ESB}}$. We introduce the class of PGB MGT detectors. A PGB detector operates on a specially constructed graph with two types of nodes: token nodes and text sequence nodes [70]. Formally, let $\mathcal{V} = \mathcal{S} \cup \mathcal{D}$ denote the complete node set, where

$$\mathcal{S} = \{s | \exists \mathbf{S} \in \mathcal{T}_h \cup \mathcal{T}_m, i \in [|\mathbf{S}|] : s_i = s\},$$
$$\mathcal{D} = \{\mathbf{S} | \mathbf{S} \in \mathcal{T}_h \cup \mathcal{T}_m \cup \{\mathbf{S}_o\}\}.$$

Here, $\mathcal{S}$ represents the set of all unique tokens in either human or machine-generated texts, and $\mathcal{D}$ comprises all text sequences from both sources and the to-be-classified text.

The edge structure of the graph captures both token co-occurrences and token-sequence relationships. Two tokens $s_i, s_j \in \mathcal{S}$ are connected if they co-occur in at least λ sequences within $\mathcal{T}_h \cup \mathcal{T}_m$, where λ is a hyperparameter. Additionally, each token node is connected to sequence nodes containing that token. Edge weights are defined by two distinct functions. For token-token edges (s_i, s_j), the PGB first computes embedding vectors for each token using

$$e_\ell : \mathcal{J}_\ell(s_i) \times (\mathcal{I}_{\ell,j}(s_i))_{j \in \mathcal{J}_\ell(s_i)} \mapsto \mathbf{e}_{\ell,i}, \quad \ell \in \{h, m\},$$

where for each token s_i, the set $\mathcal{J}_\ell(s_i) = \{j | s_i \in \mathbf{S}_{\ell,j}\}$ indexes the sequences containing s_i, while $\mathcal{I}_{\ell,j}(s_i) = \{k | S_{\ell,j,k} = s_i\}$ indexes the positions where s_i appears in sequence $\mathbf{S}_{\ell,j}$. The token-token edge weights is then computed as $A_t(\mathbf{e}_{h,i}, \mathbf{e}_{m,i}, \mathbf{e}_{h,j}, \mathbf{e}_{m,j})$, where $\mathbf{e}_{h,i}$ and $\mathbf{e}_{m,i}$ are the embeddings from human and machine-generated texts, respectively. For token-sequence edges $(s, \mathbf{S})$, the weight is simply $A_s(N_{s|\mathbf{S}})$, where $N_{s|\mathbf{S}}$ counts occurrences of token s in sequence $\mathbf{S}$. Examples of these edge weight functions $A_t(\cdot)$ and $A_s(\cdot)$ are provided in equation 2 and used in our empirical evaluations.

Token nodes are initialized with one-hot features and sequence nodes with all-zeros features. The GNN operates by several rounds of message passing among connected nodes. The PGB detector applies K rounds of message passing over the constructed graph, where at each round, node embeddings are updated based on messages received from neighboring nodes. After K iterations, the detector computes the final node embeddings, denoted by $\mathbf{h}^{(K)}$. The classification output is obtained via a function $g_p : (\mathbf{h}^{(K)}, \mathbf{S}_o) \mapsto \widehat{Y}$ that maps the collection of node embeddings to the binary decision $\widehat{Y}$. A PGB detector is completely characterized by the tuple $(K, \lambda, e_h, e_m, A_t, A_s, g_p)$. We denote the collection of PGB detectors by $\mathcal{F}_{\text{PGB}}$.

The following theorem shows that the PGB class of detectors strictly subsumes the ESB class in terms of achievable detection accuracy.

Theorem 4.1. *For every ESB detector $f_{ESB} \in \mathcal{F}_{ESB}$, there exists a PGB detector $f_{PGB} \in \mathcal{F}_{PGB}$ such that the detection accuracy of f_{PGB} matches that of f_{ESB}, i.e.,*

$$P(f_{PGB}(\mathcal{T}_h, \mathcal{T}_m, \mathbf{S}_o) = Y) = P(f_{ESB}(\mathcal{T}_h, \mathcal{T}_m, \mathbf{S}_o) = Y),$$

for all pairs of probability distributions (P_h, P_m). Furthermore, the PGB class of detectors strictly improves upon the ESB class in terms of detection accuracy. That is, for any fixed set of hyperparameters T, K, λ, there exists (P_h, P_m) and $f_{PGB} \in \mathcal{F}_{PGB}$ for which:

$$P(f_{PGB}(\mathcal{T}_h,\mathcal{T}_m,\mathbf{S}_o)=Y) > \max_{f_{ESB} \in \mathcal{F}_{ESB}} P(f_{ESB}(\mathcal{T}_h,\mathcal{T}_m,\mathbf{S}_o)=Y),$$

The proof is provided in Appendix A.

5 Related Work

5.1 AI-generated text Detection

Detecting machine-generated texts plays a crucial role in authorship identification. Current approaches can be categorized into three main categories. The first category focuses on watermarking LLM-generated content [9, 2, 68, 65, 41]. Most watermarking methods operate in a white-box setting, where researchers can modify the decoding process or token distribution directly [2, 65, 41]. Some approaches tackle the more challenging black-box setting by implementing post-processing modules to embed watermarks [9, 68]. The second category encompasses training-based detection methods that leverage trained neural networks [22, 49, 77, 28, 51]. OpenAI developed GPT-2 detectors using RoBERTa [36] as their foundation model [49]. MMD-MP [78] was proposed to enhance the stability of maximum mean discrepancy (MMD) in order to detect subtle distributional differences between MGT and HGT. Additionally, researchers have explored fine-tuning language models specifically for detection purposes [34, 30, 20]. The third category consists of zero-shot detection methods [42, 76, 69, 54, 38], which utilize existing tools like LLMs without additional training. For example, SimLLM [42] generates comparative text samples to identify machine-generated content through similarity analysis. R-Detect [50] suggests a non-parametric kernel relative test to check if a text's distribution is closer to HGT than MGT.

5.2 Explainable LLMs & GNNs

Large language models often function as black-box systems, presenting inherent risks for downstream applications [81]. To address this limitation, researchers have developed various explanation methods [66, 33, 17, 11], which can be divided into local and global approaches. Local explanation methods aim to illuminate how an LLM arrives at predictions for specific inputs [66, 33, 11]. For example, the leave-one-out technique represents a fundamental approach to measuring input feature importance [66, 33]. Global explanation methods focus on understanding how specific model components operate, including hidden layers and language model mechanisms. For instance, researchers have tracked attention layers to extract semantic information [66]. SASC [48] employs pre-trained models to generate explanations for various LLM components.

Various approaches have emerged for extracting subgraph explanations using GNN [75, 35, 18, 67, 12]. These methods can be categorized into several groups. Gradient-based traditional approaches, including SA [4] and Grad-CAM [44], leverage gradient information to derive explanations. Model-agnostic techniques encompass three main categories. First, perturbation-based methods such as

GNNExplainer [71], PGExplainer [37], and ReFine [60] identify important features and subgraph structures through systematic perturbations. Second, surrogate methods [58, 16] approximate local predictions using surrogate models to generate explanations. Third, generation-based approaches [74, 47, 61] employ generative models to produce both instance-level and global-level explanations.

6 Experiments

We conduct extensive experiments to evaluate LM^{2}OTIFS across two aspects: MGT detection performance, and explainable motifs effectiveness. For MGT detection, we compare LM^{2}OTIFS against state-of-the-art supervised and zero-shot detectors on multiple benchmark datasets in both in-domain and cross-domain aspects. To validate our explainable motifs, we follow the [23, 46] to use Most Relevant First (MoRF) and Least Relevant First (LeRF) to verify the effectiveness.

6.1 Setups

Datasets. Following established benchmarks in MGT detection [69, 76], we evaluate LM^{2}OTIFS on six comprehensive datasets: HC3 [20], M4 [62], and RAID [15], Yelp [39], Creative, Essay [57, 21]. We select four domains in each dataset: open-qa, wiki-csai, medicine, and finance in HC3; wiki-how, reddit, peerread, and arxiv in M4; and recipes, book summaries, poetry, and IMDB reviews in RAID. The HC3 dataset only contains ChatGPT-generated text. While in M4 and RAID, there are several kinds of LLM-generated texts. In this paper, we also consider language models: Davinci, Cohere, Dolly, and BloomZ in M4, Llama2, GPT-4, MPT, and Mistral in RAID. In Yelp, Creative, and Essay, we consider three LLMs, Claude3-Sonnet, Claude-3-Opus, and Gemini-1.0-Pro. The dataset details are available in Appendix B.1.

Table 1: Detection comparisons on HGTs and ChatGPT-generated texts. The best and second-best results are shown in bold font and underlined. * means the model is trained on that dataset.

	ACC				AUC			
Method	HC3	M4	RAID	Avg.	HC3	M4	RAID	Avg.
Likelihood [49]	0.75	0.88	0.85	0.83	**1.00**	0.90	0.98	0.96
Rank [19]	0.53	0.58	0.56	0.56	0.89	0.95	0.91	0.92
LogRank [25]	0.70	0.87	0.84	0.81	**1.00**	0.94	<u>0.97</u>	0.97
Entropy [19]	0.77	0.73	0.66	0.72	<u>0.95</u>	0.79	0.89	0.88
NPR [52]	0.83	0.71	0.79	0.78	**1.00**	0.93	<u>0.97</u>	0.97
LRR [52]	0.96	0.86	0.87	0.90	**1.00**	0.98	0.96	0.98
DetectGPT [40]	0.63	0.61	0.62	0.62	0.56	0.63	0.78	0.66
Fast-DetectGPT [6]	0.97	<u>0.96</u>	<u>0.97</u>	<u>0.97</u>	**1.00**	<u>0.99</u>	**1.00**	<u>0.99</u>
DNAGPT [69]	0.73	0.68	0.72	0.71	0.88	0.86	0.93	0.89
Binoculars [38]	<u>0.98</u>	0.94	**0.99**	<u>0.97</u>	**1.00**	0.98	**1.00**	<u>0.99</u>
Glimpse [5]	<u>0.98</u>	0.94	0.91	0.94	**1.00**	0.98	0.96	0.98
GPTZero [55]	0.77	0.75	0.68	0.73	0.77	0.75	0.68	0.73
RoBERTa-QA [20]	**1.00***	0.95	0.80	0.91	**1.00***	<u>0.99</u>	0.96	0.98
Radar [24]	0.66	0.76	0.77	0.73	0.52	0.83	0.95	0.76
DeTeCtive [22]	0.92	0.93	0.96	0.93	0.93	0.94	0.98	0.95
LM^{2}OTIFS	0.97	**0.98**	**0.99**	**0.98**	**1.00**	**1.00**	**1.00**	**1.00**

Implementation. The detector is implemented as a two-layer Graph Convolutional Network. The input dimension of the first layer is dependent on the token size of the training set. The hidden dimension is 64, and the output dimensionality is fixed to the number of text categories. We use the Bert [13] tokenizer as the tokenizer.

We employ the Adam [29] as the optimizer with the learning rate 5E-4, 5000 epochs. For the motif extraction, we adapt the GNNExplainer [71] to fit our analysis. We follow the Refine [59] to implement the GNNExplainer. The optimizer for GNNExplainer is Adam with a learning rate 1E-3, 100 epochs. Hardware platform consisted of a Linux system with eight NVIDIA A100 GPUs (40GB each), running CUDA 11.3, Python 3.9, and PyTorch 1.10.1.

6.2 Detection Performance Comparison

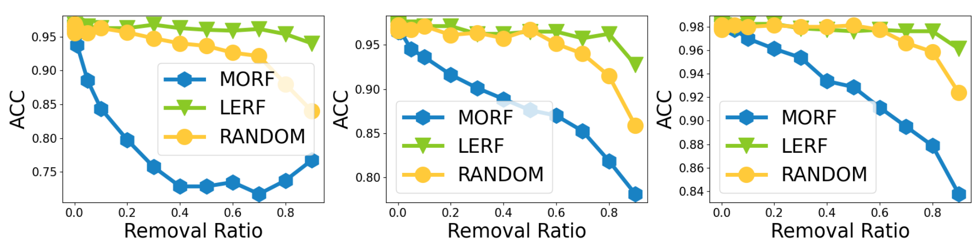
We compare LM^{2}OTIFS against 13 baselines, including supervised and zero-shot methods, to evaluate detection performance. The summary of baselines is provided in Appendix B.2. We report both accuracy(ACC) and area under the receiver operating characteristic(AUC) results. For the in-domain setting, we train and test our method on the same domain. In Table 1, we report the average results of ChatGPT-based texts detection on three datasets. LM^{2}OTIFS achieves the best performance under ACC and AUC metrics. In Table 2, we study the performance across various LLMs. As the results show, the performance is aligned with Table 1. Under the ACC metric, LM^{2}OTIFS is the best performance on average, demonstrating the ability for MGT detection. The detailed results are available in Appendix C. Due to the limitation of pages, we provide more experiments about cross-domain evaluation, statistical significance analysis and comparison with TextGCN in Appendix C.1.

Table 2: Detection comparisons on HGT and MGT based on ACC. The best and second-best results are shown in bold font and underlined. YSC represents the combination of Yelp, Essay, and Creative datasets. DaV., Coh., Dol., Blo., Lla., GT4, Mis., Son., Opu. and Gem. are short for DaVinci, Cohere, Dolly, BloomZ, Llama, GPT-4, Mistral Claude3-Sonnet, Claude-3-Opus and Gemini.

	M4				RAID				YSC			
Method	DaV.	Coh.	Dol.	Blo.	Lla.	GT4	MPT	Mis.	Son.	Opu.	Gem.	Avg.
Likelihood [49]	0.69	0.87	0.66	0.54	0.79	0.75	0.50	0.65	0.80	0.83	0.74	0.71
Rank [19]	0.51	0.54	0.53	0.53	0.53	0.53	0.51	0.52	0.52	0.52	0.52	0.52
LogRank [25]	0.67	0.88	0.72	0.62	0.80	0.74	0.46	0.66	0.76	0.79	0.72	0.71
Entropy [19]	0.62	0.61	0.53	0.53	0.62	0.62	0.52	0.63	0.72	0.74	0.64	0.62
NPR [52]	0.63	0.67	0.55	0.59	0.79	0.66	0.54	0.65	0.70	0.63	0.54	0.63
LRR [52]	0.77	0.75	0.74	0.77	0.87	0.71	0.55	0.71	0.74	0.72	0.53	0.72
DetectGPT [40]	0.48	0.57	0.48	0.59	0.67	0.59	0.46	0.53	0.62	0.58	0.57	0.56
Fast-DetectGPT [6]	0.81	**0.98**	0.90	0.54	0.94	0.85	0.48	0.64	0.85	0.88	0.76	0.78
DNAGPT [69]	0.53	0.74	0.53	0.50	0.68	0.66	0.39	0.54	0.62	0.64	0.65	0.59
Binoculars [38]	0.83	0.97	0.90	0.66	**0.98**	0.92	0.58	0.71	0.88	0.91	0.81	0.83
Glimpse [5]	0.74	0.94	0.69	0.61	0.88	0.77	0.68	0.77	0.85	0.85	0.76	0.69
GPTZero [55]	0.74	0.80	0.61	0.53	0.65	0.60	0.54	0.55	0.69	0.71	0.54	0.63
RoBERTa-QA [20]	0.83	0.94	0.74	0.51	0.77	0.70	0.56	0.56	0.79	0.87	0.80	0.73
Radar [24]	0.76	0.77	0.65	0.63	0.68	0.69	0.64	0.72	0.80	0.83	0.78	0.72
DeTeCtive [22]	0.90	0.85	0.90	0.92	0.96	0.97	**0.92**	0.88	0.94	0.91	0.86	0.91
LM^{2}OTIFS	**0.95**	0.97	**0.91**	**0.98**	**0.98**	**1.00**	0.90	**0.91**	**0.99**	**0.99**	**0.91**	**0.95**

6.3 Explanation Evaluation

Quantitive Analysis. Due to a lack of ground truth, evaluating the effectiveness of explanations remains challenging. Therefore, we follow previous work [23, 46] using MoRF to verify the motifs. MoRF and LeRF are popular evaluation protocols in XAI that assess the faithfulness of explanations by measuring how the model's prediction changes when the most or least relevant input features are sequentially removed according to explanations. Based on the MoRF and LeRF protocol, we evaluate our motifs on the ChatGPT-generated texts. We first extract the explainable motifs, which indicate the importance of each edge. Then we remove the most important edges following an increasing sequence. The area under the curve is lower is better for MoRF, and higher is better for LeRF. To verify that the explainable motifs are effective, we introduce a baseline, random motifs, where the importance of each edge is randomly assigned.



(a) HC3 (b) M4 (c) RAID

Figure 4: Comparison results of MoRF and LeRF between explainable motifs extracted from LM^{2}OTIFS and random motifs.

As Figure 4 shows, the motifs from LM^{2}OTIFS are much better than the baseline. From the MoRF protocol, when the most important 20% edges are removed, the explainable motifs cause more than an average 15% accuracy drop on HC3 dataset while random motifs keep the accuracy the same. Similarly, in the M4 and RAID datasets, when 50% edges are deleted, the explainable motifs have a more significant impact on accuracy than the random motifs. Under the LeRF protocol, the explainable motifs cause a lower performance drop than random motifs. In summary, we verify the effectiveness of explainable motifs through a common XAI evaluation protocol, and the results show that the performance of explainable motifs is much higher than the baseline and random motifs. For the detailed results of each domain, we provided the results in Appendix C.2.

Table 3 reveals distinct motif fingerprints—frequency variations between HGT and MGT across tokens and token-token co-occurrences. Selecting the top 0.05% of edges as global explainable motifs highlights a notable difference: HGT shows a higher ratio of token and token-token co-occurrences compared to MGT. This suggests that for MGT detection, word-to-word connections are more influential than for HGT detection, given the same number of tokens. One possible explanation is that language models excel at utilizing diverse word collocations, while humans tend to rely on more conventional patterns. More results are available in Appendix C.2.

Table 3: Statistics of text covered by explanation motifs. We report how many HGT/MGT contain tokens/token-tokens in explanation motifs. The sparsity of the explanation motifs is 0.05%.

	open-qa		wiki-csai		medicine		finance	
Statistic	HGT	MGT	HGT	MGT	HGT	MGT	HGT	MGT
Tokens(Nodes)	610	2407	1685	777	923	990	1251	618
Token-Token(Edges)	277	3496	2180	1993	797	2086	2004	1816
Ratio(Nodes/Edges)	2.20	0.69	0.77	0.39	1.16	0.47	0.62	0.34

To elucidate motif patterns, we visualize both graph and corresponding text motifs, encompassing word-level and high-order structures. Word-level motifs highlight word occurrence probabilities, while high-order motifs capture complex relationships, such as phrasal and semantic structures. Figure 5 presents examples extracted from the PubMed [27] dataset, preserving the top 2% of edges. While GPT-4 and Davinci share common words (e.g., "sleep", "patients"), our method captures distinct phrasal patterns. For instance, GPT-4's "ob" and "sleep" (purple) indicate "obstructive sleep", whereas Davinci's "disorder" and "sleep" represent "sleep-related breathing disorder". Furthermore, GPT-4's connection of "airway", "improves", and "serum" reveals sentence-level patterns. Detailed case studies are provided in Appendix C.2. The motifs reveal that different language models possess *distinct and visible fingerprints*.

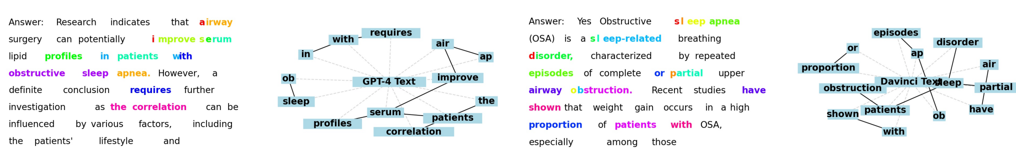


Figure 5: High-order explainable motif samples from GPT-4 and Davinci. We extract motifs from texts in the PubMed dataset for the same question. In graph motifs, solid lines represent subgraph motifs and dashed lines mean the text contains words. In text motifs, words highlighted in the same color are connected in the corresponding graph motifs. A single word may contain multiple colors.

7 Conclusion

This paper focuses on explainable authorship detection, introducing a framework that identifies characteristic motifs to provide insight into model decisions. We evaluate our method against supervised and zero-shot learning baselines across various domains, demonstrating comparable performance. We follow the previous XAI evaluation protocol to verify the effectiveness of the explainable motifs.

Limitation. Although we demonstrate the effectiveness of our method both theoretically and experimentally, we have not explored the impact of different GNN architectures or variations in their hyperparameter settings.

Impact Statements

LM^{2}OTIFS introduces a framework using limited samples to achieve authorship detection, providing a new method for detecting the MGT generated by rapidly updating LLMs. A potential risk associated with the explainability of our method is that malicious actors could leverage the identified motifs to modify machine-generated text in a way that circumvents detection.

References

[1] Alim Al Ayub Ahmed, Ayman Aljabouh, Praveen Kumar Donepudi, and Myung Suh Choi. Detecting fake news using machine learning: A systematic literature review. *arXiv preprint arXiv:2102.04458*, 2021.

[2] Anirudh Ajith, Sameer Singh, and Danish Pruthi. Downstream trade-offs of a family of text watermarks. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 14039–14053, 2024.

[3] Anthropic. Claude3. 2024.

[4] Federico Baldassarre and Hossein Azizpour. Explainability techniques for graph convolutional networks. *arXiv preprint arXiv:1905.13686*, 2019.

[5] Guangsheng Bao, Yanbin Zhao, Juncai He, and Yue Zhang. Glimpse: Enabling white-box methods to use proprietary models for zero-shot llm-generated text detection, 2025.

[6] 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 *The Twelfth International Conference on Learning Representations*, 2024.

[7] Yoshua Bengio, Réjean Ducharme, and Pascal Vincent. A neural probabilistic language model. *Advances in neural information processing systems*, 13, 2000.

[8] Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*, volume 4. Springer, 2006.

[9] Yapei Chang, Kalpesh Krishna, Amir Houmansadr, John Wieting, and Mohit Iyyer. Postmark: A robust blackbox watermark for large language models. *arXiv preprint arXiv:2406.14517*, 2024.

[10] Canyu Chen and Kai Shu. Combating misinformation in the age of llms: Opportunities and challenges. *AI Magazine*, 45(3):354–368, 2024.

[11] Hanjie Chen, Faeze Brahman, Xiang Ren, Yangfeng Ji, Yejin Choi, and Swabha Swayamdipta. Rev: Information-theoretic evaluation of free-text rationales. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2007–2030, 2023.

[12] Zhuomin Chen, Jiaxing Zhang, Jingchao Ni, Xiaoting Li, Yuchen Bian, Md Mezbahul Islam, Ananda Mondal, Hua Wei, and Dongsheng Luo. Generating in-distribution proxy graphs for explaining graph neural networks. In *Forty-first International Conference on Machine Learning*.

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186, 2019.

[14] Ning Ding, Shengding Hu, Weilin Zhao, Yulin Chen, Zhiyuan Liu, Hai-Tao Zheng, and Maosong Sun. Openprompt: An open-source framework for prompt-learning. *arXiv preprint arXiv:2111.01998*, 2021.

[15] Liam Dugan, Alyssa Hwang, Filip Trhlík, Andrew Zhu, Josh Magnus Ludan, Hainiu Xu, Daphne Ippolito, and Chris Callison-Burch. RAID: A shared benchmark for robust evaluation of machine-generated text detectors. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12463–12492, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[16] Alexandre Duval and Fragkiskos D Malliaros. Graphsvx: Shapley value explanations for graph neural networks. In *Machine Learning and Knowledge Discovery in Databases. Research Track: European Conference, ECML PKDD 2021, Bilbao, Spain, September 13–17, 2021, Proceedings, Part II 21*, pages 302–318. Springer, 2021.

[17] Joseph Enguehard. Sequential integrated gradients: a simple but effective method for explaining language models. *arXiv preprint arXiv:2305.15853*, 2023.

[18] Junfeng Fang, Xiang Wang, An Zhang, Zemin Liu, Xiangnan He, and Tat-Seng Chua. Cooperative explanations of graph neural networks. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*, pages 616–624, 2023.

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

[20] 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.

[21] Hanxi Guo, Siyuan Cheng, Xiaolong Jin, Zhuo Zhang, Kaiyuan Zhang, Guanhong Tao, Guangyu Shen, and Xiangyu Zhang. Biscope: Ai-generated text detection by checking memorization of preceding tokens. *Advances in Neural Information Processing Systems*, 37:104065–104090, 2024.

[22] Xun Guo, Shan Zhang, Yongxin He, Ting Zhang, Wanquan Feng, Haibin Huang, and Chongyang Ma. Detective: Detecting ai-generated text via multi-level contrastive learning. *arXiv preprint arXiv:2410.20964*, 2024.

[23] Sara Hooker, Dumitru Erhan, Pieter-Jan Kindermans, and Been Kim. A benchmark for interpretability methods in deep neural networks. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[24] 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.

[25] Daphne Ippolito, Daniel Duckworth, Chris Callison-Burch, and Douglas Eck. Automatic detection of generated text is easiest when humans are fooled. *arXiv preprint arXiv:1911.00650*, 2019.

[26] Sarthak Jain and Byron C Wallace. Attention is not explanation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3543–3556, 2019.

[27] Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William W Cohen, and Xinghua Lu. Pubmedqa: A dataset for biomedical research question answering. *arXiv preprint arXiv:1909.06146*, 2019.

[28] Zae Myung Kim, Kwang Hee Lee, Preston Zhu, Vipul Raheja, and Dongyeop Kang. Threads of subtlety: Detecting machine-generated texts through discourse motifs. *arXiv preprint arXiv:2402.10586*, 2024.

[29] Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

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

[31] Daphane Koller. *Probabilistic Graphical Models: Principles and Techniques*. The MIT Press, 2009.

[32] Jooyoung Lee, Thai Le, Jinghui Chen, and Dongwon Lee. Do language models plagiarize? In *Proceedings of the ACM Web Conference 2023*, pages 3637–3647, 2023.

[33] Jiwei Li, Will Monroe, and Dan Jurafsky. Understanding neural networks through representation erasure. *arXiv preprint arXiv:1612.08220*, 2016.

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

[35] Wanyu Lin, Hao Lan, and Baochun Li. Generative causal explanations for graph neural networks. In *International Conference on Machine Learning*, pages 6666–6679. PMLR, 2021.

[36] Yinhan Liu. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 364, 2019.

[37] Dongsheng Luo, Wei Cheng, Dongkuan Xu, Wenchao Yu, Bo Zong, Haifeng Chen, and Xiang Zhang. Parameterized explainer for graph neural network. *Advances in neural information processing systems*, 33:19620–19631, 2020.

[38] 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*, pages 17538–17553, 2024.

[39] Chengzhi Mao, Carl Vondrick, Hao Wang, and Junfeng Yang. Raidar: generative ai detection via rewriting. *arXiv preprint arXiv:2401.12970*, 2024.

[40] 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*, pages 24950–24962. PMLR, 2023.

[41] Piotr Molenda, Adian Liusie, and Mark JF Gales. Waterjudge: Quality-detection trade-off when watermarking large language models. *arXiv preprint arXiv:2403.19548*, 2024.

[42] 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*, pages 22340–22352, 2024.

[43] OpenAI. Chatgpt. 2022.

[44] Phillip E Pope, Soheil Kolouri, Mohammad Rostami, Charles E Martin, and Heiko Hoffmann. Explainability methods for graph convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10772–10781, 2019.

[45] 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.

[46] Yao Rong, Tobias Leemann, Vadim Borisov, Gjergji Kasneci, and Enkelejda Kasneci. A consistent and efficient evaluation strategy for attribution methods. In *Proceedings of the 39th International Conference on Machine Learning*, pages 18770–18795. PMLR, 2022.

[47] Caihua Shan, Yifei Shen, Yao Zhang, Xiang Li, and Dongsheng Li. Reinforcement learning enhanced explainer for graph neural networks. *Advances in Neural Information Processing Systems*, 34:22523–22533, 2021.

[48] Chandan Singh, Aliyah R Hsu, Richard Antonello, Shailee Jain, Alexander G Huth, Bin Yu, and Jianfeng Gao. Explaining black box text modules in natural language with language models. *arXiv preprint arXiv:2305.09863*, 2023.

[49] 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.

[50] Yiliao Song, Zhenqiao Yuan, Shuhai Zhang, Zhen Fang, Jun Yu, and Feng Liu. Deep kernel relative test for machine-generated text detection. In *The Thirteenth International Conference on Learning Representations*, 2025.

[51] Rafael Alberto Rivera Soto, Kailin Koch, Aleem Khan, Barry Y Chen, Marcus Bishop, and Nicholas Andrews. Few-shot detection of machine-generated text using style representations. In *The Twelfth International Conference on Learning Representations*.

[52] Jinyan Su, Terry Yue Zhuo, Di Wang, and Preslav Nakov. Detectllm: Leveraging log rank information for zero-shot detection of machine-generated text. *arXiv preprint arXiv:2306.05540*, 2023.

[53] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International conference on machine learning*, pages 3319–3328. PMLR, 2017.

[54] Yufei Tian, Zeyu Pan, and Nanyun Peng. Detecting machine-generated long-form content with latent-space variables. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 10394–10408, 2024.

[55] Tian, Edward. GPTZero. `https://gptzero.me`, 2023.

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

[57] Vivek Verma, Eve Fleisig, Nicholas Tomlin, and Dan Klein. Ghostbuster: Detecting text ghostwritten by large language models. *arXiv preprint arXiv:2305.15047*, 2023.

[58] Minh Vu and My T Thai. Pgm-explainer: Probabilistic graphical model explanations for graph neural networks. *Advances in neural information processing systems*, 33:12225–12235, 2020.

[59] Xiang Wang, Ying-Xin Wu, An Zhang, Xiangnan He, and Tat-Seng Chua. Towards multi-grained explainability for graph neural networks. In *Proceedings of the 35th Conference on Neural Information Processing Systems*, 2021.

[60] Xiang Wang, Yingxin Wu, An Zhang, Xiangnan He, and Tat-Seng Chua. Towards multi-grained explainability for graph neural networks. *Advances in Neural Information Processing Systems*, 34:18446–18458, 2021.

[61] Xiaoqi Wang and Han-Wei Shen. Gnninterpreter: A probabilistic generative model-level explanation for graph neural networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2023.

[62] Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Toru Sasaki, Thomas Arnold, Alham Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. M4: Multi-generator, multi-domain, and multi-lingual black-box machine-generated text detection. In Yvette Graham and Matthew Purver, editors, *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1369–1407, St. Julian's, Malta, March 2024. Association for Computational Linguistics.

[63] Sarah Wiegreffe and Yuval Pinter. Attention is not not explanation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 11–20, 2019.

[64] Junchao Wu, Shu Yang, Runzhe Zhan, Yulin Yuan, Lidia Sam Chao, and Derek Fai Wong. A survey on llm-generated text detection: Necessity, methods, and future directions. *Computational Linguistics*, pages 1–66, 2025.

[65] Yihan Wu, Zhengmian Hu, Junfeng Guo, Hongyang Zhang, and Heng Huang. A resilient and accessible distribution-preserving watermark for large language models. In *Forty-first International Conference on Machine Learning*.

[66] Zhiyong Wu, Yun Chen, Ben Kao, and Qun Liu. Perturbed masking: Parameter-free probing for analyzing and interpreting bert. *arXiv preprint arXiv:2004.14786*, 2020.

[67] Yaochen Xie, Sumeet Katariya, Xianfeng Tang, Edward Huang, Nikhil Rao, Karthik Subbian, and Shuiwang Ji. Task-agnostic graph explanations. *Advances in Neural Information Processing Systems*, 35:12027–12039, 2022.

[68] Xi Yang, Kejiang Chen, Weiming Zhang, Chang Liu, Yuang Qi, Jie Zhang, Han Fang, and Nenghai Yu. Watermarking text generated by black-box language models. *arXiv preprint arXiv:2305.08883*, 2023.

[69] 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*.

[70] Liang Yao, Chengsheng Mao, and Yuan Luo. Graph convolutional networks for text classification. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 7370–7377, 2019.

[71] Zhitao Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. Gnnexplainer: Generating explanations for graph neural networks. *Advances in neural information processing systems*, 32, 2019.

[72] 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*, pages 15838–15846, 2024.

[73] Ann Yuan, Andy Coenen, Emily Reif, and Daphne Ippolito. Wordcraft: story writing with large language models. In *Proceedings of the 27th International Conference on Intelligent User Interfaces*, pages 841–852, 2022.

[74] Hao Yuan, Jiliang Tang, Xia Hu, and Shuiwang Ji. Xgnn: Towards model-level explanations of graph neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 430–438, 2020.

[75] Hao Yuan, Haiyang Yu, Shurui Gui, and Shuiwang Ji. Explainability in graph neural networks: A taxonomic survey. *IEEE transactions on pattern analysis and machine intelligence*, 45(5):5782–5799, 2022.

[76] Cong Zeng, Shengkun Tang, Xianjun Yang, Yuanzhou Chen, Yiyou Sun, Yao Li, Haifeng Chen, Wei Cheng, Dongkuan Xu, et al. Dald: Improving logits-based detector without logits from black-box llms. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[77] Shuhai Zhang, Yiliao Song, Jiahao Yang, Yuanqing Li, Bo Han, and Mingkui Tan. Detecting machine-generated texts by multi-population aware optimization for maximum mean discrepancy. In *The Twelfth International Conference on Learning Representations*.

[78] Shuhai Zhang, Yiliao Song, Jiahao Yang, Yuanqing Li, Bo Han, and Mingkui Tan. Detecting machine-generated texts by multi-population aware optimization for maximum mean discrepancy. In *The Twelfth International Conference on Learning Representations*, 2024.

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

[80] Yaolun Zhang, Yinxu Pan, Yudong Wang, and Jie Cai. Pybench: Evaluating llm agent on various real-world coding tasks. *arXiv preprint arXiv:2407.16732*, 2024.

[81] Haiyan Zhao, Hanjie Chen, Fan Yang, Ninghao Liu, Huiqi Deng, Hengyi Cai, Shuaiqiang Wang, Dawei Yin, and Mengnan Du. Explainability for large language models: A survey. *ACM Transactions on Intelligent Systems and Technology*, 15(2):1–38, 2024.

[82] Yuchen Zhuang, Yue Yu, Kuan Wang, Haotian Sun, and Chao Zhang. Toolqa: A dataset for llm question answering with external tools. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 50117–50143. Curran Associates, Inc., 2023.

Contents

A Proof of Theorem 4.1

We first prove that the ensemble of PGB detectors is at least as accurate as the ensemble of ESB detectors. To this end, let us recall that an ESB detector is completely characterized by the mapping g_s and the PGB detector by $(K, \lambda, e_h, e_m, A_t, A_s, g_p)$. Let us consider an arbitrary ESB detector by fixing the function $g_s(\cdot)$. The ESB detector computes $\widehat{P}_\ell(s_t|s_{1:t-1}), \ell \in \{h, m\}, t \in [T]$ empirically and uses $g_s((\widehat{P}_\ell(s_t|s_{1:t-1}))_{\ell\in\{h,m\},t\in[T]}, \mathbf{S}_o)$ for detection. On the other hand, the PGB uses the embedding functions e_ℓ, A_t, A_s to compute the final node embeddings $\mathbf{h}^{(K)}$ and the mapping $g_p(\mathbf{h}^{(K)}, \mathbf{S}_o)$ for detection. We take $K = T$ and $\lambda = 1$. Then, to prove that there exists a PGB which matches the ESB in terms of detection accuracy, it suffices to show that there exist choices of embedding functions e_ℓ, A_t, A_s, such that the empirical estimate $\widehat{P}_\ell(s_t|s_{1:t-1}), \ell \in \{h, m\}, t \in [T]$ can be written as a function of the final node embeddings $\mathbf{h}^{(T)}$, i.e., there exists $r(\cdot)$ such that $r(\mathbf{h}^{(T)}) = (\widehat{P}_\ell(s_t|s_{1:t-1}))_{\ell\in\{h,m\},t\in[T]}$. Then, the proof follows by taking $g_p(r(\mathbf{h}^{(T)}), \mathbf{S}_o) = g_s((\widehat{P}_\ell(s_t|s_{1:t-1}))_{\ell\in\{h,m\},t\in[T]}, \mathbf{S}_o)$, so that

$$P(f_{\text{PGB}}(\mathcal{T}_h, \mathcal{T}_m, \mathbf{S}_o) = f_{\text{ESB}}(\mathcal{T}_h, \mathcal{T}_m, \mathbf{S}_o)) = 1.$$

To this end, we take e_ℓ as the identity function and A_t as a one-to-one parametrization function, so that for each token node s_i, the collection $\mathcal{J}_\ell(s_i) \times (\mathcal{I}_{\ell,j}(s_i))_{j\in\mathcal{J}_\ell(s_i)}$ can be computed from its connected edge weights, where $\mathcal{J}_\ell(s_i)$ is the training sequence indices in which the token is present and $\mathcal{I}_{\ell,j}(s_i)$ is the collection of indices in the sequence $\mathbf{S}_{\ell,j}, j \in \mathcal{J}_\ell$ whose value is equal to s_i. We further note that

$$\widehat{P}_\ell(s_t|s_{1:t-1}) = \frac{1}{|\mathcal{J}_\ell(s_t)|} \sum_{i=1}^{|\mathcal{J}_\ell(s_t)|} \frac{\sum_{j=1}^{|\mathbf{S}_{\ell,i}|-t} \mathbb{1}(\mathbf{S}_{\ell,i,j:j+t} = s_{1:t})}{\sum_{j=1}^{|\mathbf{S}_{\ell,i}|-t} \mathbb{1}(\mathbf{S}_{\ell,i,j:j+t-1} = s_{1:t-1})},$$

Furthermore,

$$\mathbb{1}(\mathbf{S}_{\ell,i,j:j+t} = s_{1:t}) = \prod_{s_i : i \in [t]} \mathbb{1}((j+i) \in \mathcal{I}_{\ell,j}(s_i)),$$

Consequently, for each $t \in [T]$, the conditional distribution $\widehat{P}_\ell(s_t|s_{1:t-1})$ can be computed as a function of $\mathbf{h}^{(t)}$. As a result, the aggregate final node embedding $\mathbf{h}^{(T)}$ can yeild $\widehat{P}_\ell(s_t|s_{1:t-1}), \ell \in \{h, m\}, t \in [T]$ as a function. This complete the first part of the proof.

To prove strict improvements of PGM detectors over ESB detectors in terms of detection accuracy, we note that ESB detectors are restricted by their limited context length T. To provide a concrete example, consider a detection scenario characterized by the pair of probability distributions P_h, P_m, where all human and machine generated text sequences have length greater than T. That is, for any sequence $\mathbf{S}_\ell = (S_{\ell,1}, S_{\ell,2}, \cdots, S_{\ell,L})$ with $L \leq T$, we have $P_\ell(S_{\ell,1}, S_{\ell,2}, \cdots, S_{\ell,L}) = 0$, where $\ell \in \{h, m\}$. Furthermore, assume that the vocabulary consists of two tokens $\{a, b\}$. Both human and machine generated text sequences consist of tokens generated independently and with equal probability over the vocabulary for all indices in $\{1, 2, \cdots, L-1\}$. The human generated text always ends with the token a and machine generated text with the token b, i.e., $P(S_{h,L} = a) = P(S_{m,L} = b) = 1$. Then, it is straightforward to see that a PGM can achieve accuracy equal to one, since the edge weights, which are functions of $\mathcal{J}_\ell \times (\mathcal{I}_{\ell,j})_{j\in\mathcal{J}_\ell}$ can capture the fact that the human generated text ends in a and machine generated text ends in b. On the other hand, for an ESB, it can be noted that all of the empirical conditional distributions $\widehat{P}_\ell(s_t|s_{1:t-1}), t \in [T], \ell \in \{h, m\}$ converge to uniform Bernoulli distributions as $L \to \infty$. So, the ESB achieves an accuracy which is strictly less than 1 due to its limited context length, and its accuracy converges to $\frac{1}{2}$ as $L \to \infty$. This completes the proof. $\square$

B Experimental Setup Details

B.1 Datasets

Our evaluation employs six distinct datasets. For HC3, M4, and RAID, we specifically selected four domains from each. The details of these datasets and the chosen domains are provided below. The HC3 dataset [20] comprises questions with both MGT and HGT across five domains: reddit, open-qa, wiki-csai, medicine, and finance, all generated by ChatGPT [43]. For our experiments, we selected the latter four domains. The M4 dataset [62] serves as a valuable resource for detector training, featuring MGT from diverse domains such as wiki-how, reddit, peerread, and arxiv, generated by several LLMs including Davinci, Dolly, and BloomZ. Similarly, the RAID dataset [15] encompasses over 10 million documents generated by 11 LLMs across 11 genres. Our benchmark includes four domains from RAID: recipe, book, poetry, and review. The Yelp [39], Creative, and Essay [57, 21] datasets contain texts generated by five LLMs: GPT-3.5, GPT-4, Claude-3-Sonnet, Claude-3-Opus, and Gemini-1.0-Pro. For our analysis, we focus on the latter three LLMs. Detailed statistics regarding the training, validation, and test samples, along with the corresponding converted graphs, are presented in Table 4, 5, 6 and 7.

Table 4: The details of the dataset for detection between HGT and MGT generated by ChaGPT.

	HC3				M4				RAID			
	open-qa	wiki-csai	medicine	finance	wiki-how	reddit	peerread	arxiv	recipe	book	poetry	review
# Training	2,000	1,384	2,000	2,000	2,000	872	2,000	2,000	2,000	2,000	2,000	1,793
# Validation	100	100	100	100	100	100	100	100	100	100	100	100
# Test	200	200	200	200	200	200	200	200	200	200	200	200
# Nodes	15,974	12,069	8,127	9,581	20,061	11,276	18,926	10,526	6,562	20,515	16,818	17,024
# Edges	3,262K	2,635K	2,063K	2,326K	6,823K	4,658K	8,591K	3,595K	2,119K	7,076K	5,059K	5,448K

Table 5: The details of the M4 dataset for detection between HGT and MGT generated by LLMs.

	reddit				peerread				arxiv			
	Davinci	Cohere	Dolly	BloomZ	Davinci	Cohere	Dolly	BloomZ	Davinci	Cohere	Dolly	BloomZ
# Training	2,000	2,000	2,000	2,000	872	824	872	830	2,000	2,000	2,000	2,000
# Validation	100	100	100	100	100	100	100	98	100	100	100	100
# Test	200	200	200	200	200	198	200	192	200	200	200	200
# Nodes	20,867	21,701	21,344	20,944	11,059	10,837	14,366	11,340	10,153	10,724	12,039	11,468
# Edges	7,175K	7,055K	7,157K	6,601K	4,139K	3,933K	5,924K	4,296K	3,343K	3,376K	4,132K	4,011K

Table 6: The details of the RAID dataset for detection between HGT and MGT generated by LLMs.

	recipes				poetry				reviews			
	Llama	GPT-4	MPT	Mistral	Llama 2	GPT-4	MPT	Mistral	Llama 2	GPT-4	MPT	Mistral
# Training	2,000	2,000	2,000	2,000	2,000	2,000	2,000	2,000	1,793	1,793	1,793	1,793
# Validation	100	100	100	100	100	100	100	100	100	100	100	100
# Test	200	200	200	200	200	200	200	200	200	200	200	200
# Nodes	6,904	6,701	13,466	8,833	16,696	17,152	19,476	17,523	17,004	17,387	19,371	17,843
# Edges	2,125K	2,163K	4,066K	2,586K	4,766K	4,527K	5,924K	4,835K	5,039K	5,694K	6,017K	5,142K

B.2 Baselines

Our evaluation includes comparisons with several zero-shot detection methods: Likelihood, Rank, Log-Rank, Entropy [19, 49, 25], DetectGPT [40], DetectLLM (LRR and NPR) [52], DNA-GPT [69], Fast-DetectGPT [6], Glimpse [5] and Binoculars [38]. DetectGPT employs perturbations to approximate the probability distribution of the text. Fast-DetectGPT improves upon this by introducing a conditional probability curvature metric for detector optimization, thus replacing traditional perturbation-based methods. DNA-GPT adopts a distinct approach: it first truncates the input text, then uses LLMs to generate the subsequent content, and finally analyzes the N-gram differences between the original and generated text. To make a fair comparison, we utilize the OPT-2.7B model [79] as the default reference model. For detailed implementation specifics, we followed the publicly available implementation of Fast-DetectGPT [2].

Our comparative evaluation also includes training-based methods: RoBERTa-QA [20], DeTeCtive [22], and RADAR [24]. Additionally, we present comparison results with GPTZero [3]. DeTeCtive is specifically designed for multi-source MGT detection. It employs contrastive learning to minimize

[2]https://github.com/baoguangsheng/fast-detect-gpt

[3]https://gptzero.me

Table 7: The details of the dataset for detection between HGT and MGT generated by LLMs in Yelp, Creative, and Essay Dataset. Sonnet and Opus are short for Claude3-Sonnet and Claude-3-Opus.

	Yelp			Essay			Creative		
	Sonnet	Opus	Gemini	Sonnet	Opus	Gemini	Sonnet	Opus	Gemini
# Training	2,000	2,000	2,000	1,500	1,500	1,500	1,500	1,500	1,500
# Validation	200	200	200	100	100	100	100	100	100
# Test	200	200	200	200	200	200	200	200	200
# Nodes	11,581	11,308	11,350	20,836	20,748	20,868	20,597	20,057	19,936
# Edges	1,940K	1,886K	1,778K	8,422K	8,038K	7,989K	7,187K	6,308K	6,250K

the representational divergence among various MGT sources. During prediction, DeTeCtive utilizes k-nearest neighbors (KNN) to determine the classification. For our experiments, we use the DeTeCtive model trained on the OUTFOX dataset [30]. RoBERTa-QA, proposed in [20] and trained on the HC3 dataset, leverages the pre-trained RoBERTa model [36] and fine-tunes a classification layer on the HC3 data.

C Detailed Experiment Results

C.1 Extended Detection Experiments

In our experiments, we consider in-domain detection and cross-domain detection in the same dataset and report the results in this section. We report the results under ACC and AUC metrics. For GPTZero, since it provides a binary output, we consider its ACC and AUC values to be equivalent.

In-Domain Detection. We provide the detailed experiment results for distinguishing HGTs and MGTs by ChatGPT in Table 8 and Table 9. The results demonstrate that LM^{2}OTIFS achieves the best performance across all domains under both ACC and AUC metrics, aligned with our analysis. In addition, we also provide the experiment results between HGT and MGT by other LLMs in Table 18, 19,20,21, 22, and 23. LM^{2}OTIFS performs consistently well on various LLMs and achieves the best performance, indicating the effectiveness of PGM for MGT detection tasks.

Table 8: Detection comparisons with SOTA methods on ACC between HGT and ChatGPT-generated texts. The best results are shown in bold font. The second-best results are shown underlined. * means the model is trained on that dataset.

	HC3				M4				RAID				
Method	open-qa	wiki-csai	medicine	finance	wiki-how	reddit	peerread	arxiv	recipe	book	poetry	review	Avg.
Likelihood	0.85	0.81	0.76	0.58	0.85	0.96	0.80	0.92	0.83	0.96	0.82	0.78	0.83
Rank	0.54	0.53	0.54	0.51	0.57	0.55	0.58	0.61	0.51	0.65	0.53	0.54	0.56
LogRank	0.77	0.73	0.72	0.58	0.82	0.95	0.80	0.92	0.81	0.93	0.84	0.79	0.81
Entropy	0.92	0.76	0.77	0.61	0.83	0.81	0.68	0.60	0.80	0.71	0.49	0.62	0.72
NPR	0.65	0.92	0.91	0.85	0.61	0.68	0.83	0.72	0.84	0.83	0.50	0.97	0.78
LRR	0.98	0.95	0.98	0.94	0.82	0.82	**1.00**	0.81	0.94	0.81	0.75	0.98	0.90
DetectGPT	0.46	0.63	0.76	0.68	0.58	0.66	0.59	0.61	0.56	0.66	0.59	0.68	0.62
Fast-DetectGPT	0.95	0.99	0.98	0.97	0.88	0.94	**1.00**	**1.00**	0.99	0.97	0.93	**1.00**	0.97
DNAGPT	0.63	0.79	0.63	0.88	0.60	0.79	0.53	0.81	0.71	0.82	0.75	0.59	0.71
Binoculars	0.92	**1.00**	**1.00**	**1.00**	0.77	0.97	**1.00**	**1.00**	**1.00**	0.96	**0.99**	0.99	0.97
Glimpse	0.95	0.98	0.99	0.99	0.97	0.91	0.87	0.99	0.94	0.96	0.76	0.98	0.94
GPTZero	0.58	0.69	0.96	0.84	0.54	0.82	0.96	0.69	0.61	0.84	0.48	0.78	0.73
RoBERTa-QA	**1.00***	**1.00***	**1.00***	0.99*	0.88	0.96	0.99	0.95	0.83	0.86	0.50	**1.00**	0.91
Radar	0.52	0.81	0.55	0.75	0.46	0.93	0.88	0.77	0.61	0.97	0.61	0.89	0.73
DeTeCtive	0.99	0.79	0.99	0.89	0.89	0.93	0.90	0.98	0.94	0.95	0.97	0.97	0.93
LM^{2}OTIFS	0.97	0.96	0.98	0.98	**0.97**	**0.99**	0.98	0.96	0.99	**1.00**	**0.99**	0.96	**0.98**

Cross-Domain Detection. To further analysis the generality of LM^{2}OTIFS, we conduct cross-domain detection experiments. We use the open-qa, wiki-how, and books domains in HC, M4, and RAID datasets as the training domain and test on other domains, respectively. For the zero-shot baselines and RADAR, we use the training data as reference to learning a threshold and apply to the test domain. For the RoBERTa-QA, we follow its pipeline to fine-tune the RoBERTa on one domain and

Table 9: MGT detection AUC performance comparisons with SOTA methods on HGT and ChatGPT-generated texts. The best results are shown in bold font. The second-best results are shown underlined. * means the model is trained on that dataset.

	HC3				M4				RAID				
Method	open-qa	wiki-csai	medicine	finance	wiki-how	reddit	peerread	arxiv	recipe	book	poetry	review	Avg.
Likelihood	**1.00**	**1.00**	**1.00**	**1.00**	0.95	0.99	0.69	0.97	**1.00**	**1.00**	0.90	**1.00**	0.96
Rank	**1.00**	0.77	0.99	0.81	0.94	0.92	0.97	0.95	0.79	0.99	0.87	**1.00**	0.92
LogRank	**1.00**	**1.00**	**1.00**	**1.00**	0.95	0.99	0.82	0.98	**1.00**	**1.00**	0.89	**1.00**	0.97
Entropy	0.99	0.85	0.99	0.97	0.91	0.91	0.60	0.75	0.97	0.88	0.75	0.97	0.88
NPR	**1.00**	**1.00**	**1.00**	**1.00**	0.95	0.99	0.81	0.98	**1.00**	**1.00**	0.89	**1.00**	0.97
LRR	**1.00**	0.99	**1.00**	0.99	0.93	0.98	**1.00**	0.99	0.99	0.99	0.85	**1.00**	0.98
DetectGPT	0.35	0.59	0.70	0.61	0.68	0.71	0.70	0.43	0.65	0.77	0.84	0.84	0.66
Fast-DetectGPT	**1.00**	**1.00**	**1.00**	0.98	0.96	0.99	**1.00**	**1.00**	**1.00**	**1.00**	0.98	**1.00**	0.99
DNAGPT	0.72	0.95	0.91	0.94	0.97	0.95	0.56	0.94	0.94	0.97	0.84	0.98	0.89
Binoculars	0.98	**1.00**	**1.00**	**1.00**	0.90	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	0.99	**1.00**	0.99
Glimpse	**1.00**	**1.00**	**1.00**	**1.00**	**0.99**	**1.00**	0.92	**1.00**	**1.00**	**1.00**	0.85	**1.00**	0.98
GPTZero	0.58	0.69	0.96	0.84	0.54	0.82	0.96	0.69	0.61	0.84	0.48	0.78	0.73
RoBERTa-QA	**1.00***	**1.00***	**1.00***	**1.00***	0.94	**1.00**	**1.00**	**1.00**	0.90	0.99	0.95	**1.00**	0.98
Radar	0.20	0.77	0.41	0.68	0.40	0.97	**1.00**	0.95	0.99	**1.00**	0.88	0.91	0.76
DeTeCtive	**1.00**	0.84	0.99	0.89	0.90	0.98	0.90	0.99	0.99	0.97	0.97	0.97	0.95
LM²OTIFS	**1.00**	0.99	**1.00**	**1.00**	**0.99**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**

test on other domains. As Table 10 shows, LM²OTIFS performs poorly on some domains, such as the reddit domain on the M4 dataset. One potential reason is that our method is only trained on the limited training set and lacks generalization, while other methods such as zero-shot methods fully utilize the generalization of LLM.

Table 10: Cross-domain MGT detection ACC performance comparisons with SOTA methods on HGT and ChatGPT-generated texts. The best results are shown in bold font. The second-best results are shown underlined.

	HC3			M4			RAID			
Method	wiki-csai	medicine	finance	reddit	peerread	arxiv	recipe	poetry	review	Avg.
Likelihood	0.97	0.96	0.98	0.88	0.80	0.67	0.57	0.70	0.99	0.84
Rank	0.65	0.94	0.65	0.82	0.56	0.66	0.54	0.80	0.80	0.71
LogRank	0.98	0.97	0.98	0.92	0.83	0.71	0.53	0.73	0.97	0.85
Entropy	0.71	0.91	0.87	0.61	0.75	0.50	0.50	0.58	0.78	0.69
NPR	0.97	0.97	0.98	0.93	0.97	0.77	0.55	0.72	0.97	0.87
LRR	0.98	0.94	0.96	0.93	0.94	0.76	0.52	0.69	0.93	0.85
DetectGPT	0.52	0.58	0.53	0.56	0.54	0.50	0.56	0.67	0.73	0.58
Fast-DetectGPT	0.99	**0.99**	0.95	0.93	**1.00**	0.94	**1.00**	0.89	**1.00**	**0.97**
DNAGPT	0.86	0.87	0.84	0.92	0.49	0.85	0.84	0.67	0.96	0.81
Binoculars	0.97	0.97	0.97	0.89	0.89	0.89	**1.00**	**1.00**	**1.00**	0.95
Glimpse	**1.00**	0.98	**1.00**	**0.96**	0.88	**0.99**	0.96	0.78	**1.00**	0.95
RoBERTa-QA	0.53	0.65	0.53	0.77	0.93	**0.99**	0.70	0.86	0.85	0.76
Radar	0.79	0.53	0.74	0.92	0.77	0.87	0.65	0.74	0.88	0.77
DeTeCtive	0.68	0.58	0.60	0.76	0.76	0.65	0.48	0.72	0.89	0.68
LM²OTIFS	0.71	0.82	0.64	0.59	0.99	0.93	0.50	0.93	0.97	0.79

Statistical Significance Analysis. To further demonstrate the robustness of LM²OTIFS, we conducted a Statistical Significance Analysis. Specifically, we repeated our experiments five times on the HC3 dataset, each with a distinct random seed, and the resulting performance metrics are detailed in Table 11. The consistently high performance across these different runs indicates the stable and reliable nature of LM²OTIFS.

Ablation Study. To investigate the impact of different graph characteristics on the MGT detection task, we performed ablation experiments on graph categories, specifically comparing undirected versus directed graphs, and weighted versus unweighted graphs. In our experiments, we use -B and -W to represent undirected graphs and weighted graphs.

Table 11: Statistical significance analysis on HC3 dataset. We repeat experiments 5 times and report the mean and standard deviation.

Metric	open-qa	wiki-csai	medicine	finance
ACC	0.9690$\pm$0.0073	0.9410$\pm$0.0097	0.9750$\pm$0.0032	0.9810$\pm$0.0037
AUC	0.9965$\pm$0.0005	0.9938$\pm$0.0005	0.9993$\pm$0.0001	0.9983$\pm$0.0004

Table 12: Statistical significance analysis on HC3 dataset. We repeat experiments 5 times and report the mean and standard deviation. The best results are shown in bold font.

Metric	Method	open-qa	wiki-csai	medicine	finance	Avg.
ACC	LM^{2}OTIFS	0.97	0.96	0.98	**0.98**	**0.97**
	LM^{2}OTIFS-B	0.95	0.94	**1.00**	**0.98**	**0.97**
	LM^{2}OTIFS-W	**1.00**	0.84	**1.00**	0.94	0.95
	LM^{2}OTIFS-BW	0.98	0.79	**1.00**	0.93	0.92
AUC	LM^{2}OTIFS	**1.00**	0.99	**1.00**	**1.00**	**1.00**
	LM^{2}OTIFS-B	0.99	**1.00**	**1.00**	**1.00**	**1.00**
	LM^{2}OTIFS-W	**1.00**	0.84	**1.00**	0.98	0.96
	LM^{2}OTIFS-BW	**1.00**	0.86	**1.00**	0.97	0.96

We further investigated the impact of different tokenizers on the MGT detection task. Our default tokenizer is Bert's tokenizer. To assess the influence of tokenization, we conducted experiments using GPT-2's tokenizer. The results of this comparison are presented in Table 13. Our findings indicate that the choice between Bert's and GPT-2's tokenizers did not significantly affect the overall detection performance.

Table 13: Authorship detection performance comparison on HC3 dataset between default(Bert) and GPT2 tokenizers.

	open-qa		wiki-csai		medicine		finance		Avg.	
Metric	Bert	GPT2	Bert	GPT2	Bert	GPT2	Bert	GPT2	Bert	GPT2
ACC	0.97	0.99	0.96	0.95	0.98	0.98	0.98	0.97	0.97	0.97
AUROC	1.00	1.00	0.99	1.00	1.00	0.99	1.00	1.00	1.00	1.00

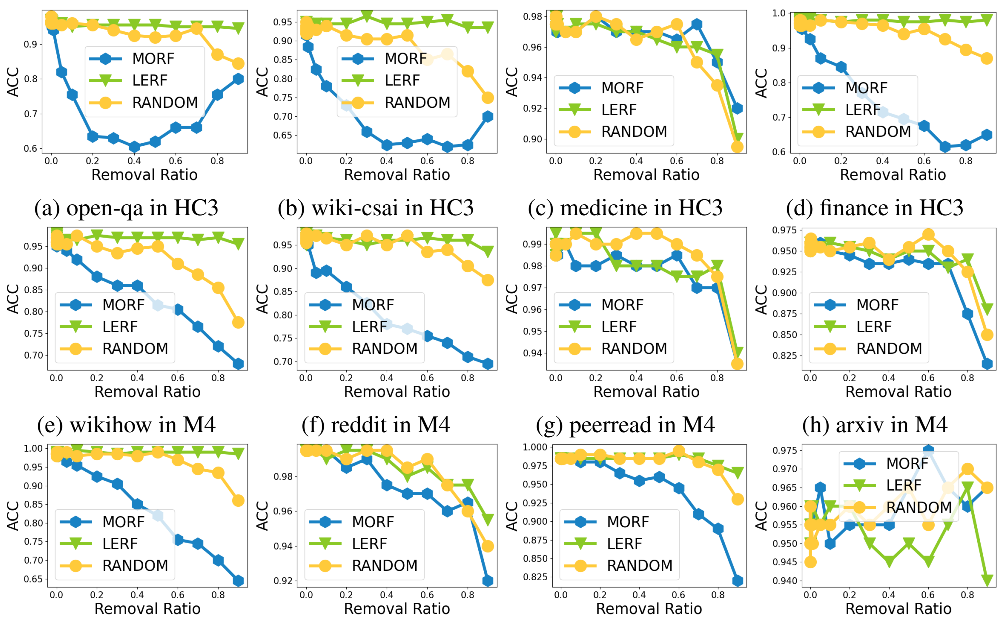
Time Consumption. Compared to other training-based methods, LM^{2}OTIFS have an additional pipeline, the graph construction phase. Specifically, its time complexity for graph construction is $O(LW^2)$, where L represents the length of the sentence and W denotes the size of the sliding window. We also evaluated the test time efficiency of LM^{2}OTIFS in comparison to several other baselines. As detailed in Table 14, LM^{2}OTIFS demonstrates the lowest time consumption during the testing phase.

Table 14: Inference time(seconds) comparison on HC3 dataset. We repeat experiments 10 times and report the average time consumption. - indicates the inference time is more than 10 minutes. The best results are shown in bold font.

	open-qa	wiki-csai	medicine	finance	Avg.
NPR	-	-	-	-	-
DNA-GPT	-	-	-	-	-
DetectGPT	442.0000	161.6530	82.2744	255.4350	235.3406
Fast-DetectGPT	28.0217	27.0673	24.1440	28.4082	26.9103
RoBERTa-QA	2.9267	2.5464	2.5391	2.5413	2.6359
DeTeCtive	18.7223	13.2559	17.2883	17.5105	16.6943
LM^{2}OTIFS	**0.0091**	**0.0065**	**0.0051**	**0.0058**	**0.0066**

C.2 Extended Motifs Evaluation

XAI Protocol Evaluation. We follow Section 6.3 to report the explainable motifs evaluation results on the HGT and ChatGPT-generated datasets. As the detailed results show in Figure 6, the explainable motifs are effective in most cases. However, in some domains, such as medicine in HC3, review in RAID, the explainable motifs are not better than random motifs. The potential reason could be the distributed nature of the explainable motifs across numerous nodes and edges. Consequently, the deletion of some edges does not drastically impede the graph network's ability to accurately perform detection. For instance, in the medicine domain of HC3 dataset, a significant performance drop in the GNN is observed when the proportion of deleted edges surpasses 70%. In the review domain of RAID dataset, even with the deletion of over 80% of the edges, the performance degradation of the GNN remains below 2%.



(a) open-qa in HC3 (b) wiki-csai in HC3 (c) medicine in HC3 (d) finance in HC3
(e) wikihow in M4 (f) reddit in M4 (g) peerread in M4 (h) arxiv in M4
(i) recipe in RAID (j) book in RAID (k) poetry in RAID (l) review in RAID

Figure 6: Comparison results of MORF and LERF between explainable motifs extracted from LM^{2}OTIFS and random motifs on HGT and ChatGPT-generated texts.

Motifs Statistical Analysis. We provide more statistical analysis on M4 and RAID datasets. As shown in Tables 15 and 24, word-to-word connections have a greater impact on detection tasks for MGT than for HGT, given the same number of tokens.

Table 15: Statistics of text covered by explanation motifs on M4 dataset. We report how many HGT/MGT contain tokens/token-tokens in explanation motifs. The sparsity of the explanation motifs is 0.05%.

	wikihow		reddit		peerread		arxiv	
Statistic	HGT	MGT	HGT	MGT	HGT	MGT	HGT	MGT
Tokens(Nodes)	4207	1894	3929	1282	2138	609	1449	725
Token-Token(Edges)	19511	8819	8448	2770	10937	6044	2954	2112
Ratio(Nodes/Edges)	0.22	0.21	0.47	0.46	0.20	0.10	0.49	0.34

Visualizations. To visualize the extracted motifs, we utilized the PubMed dataset, which includes MGT samples generated by three LLMs: GPT-4, Claude-3, and Davinci. We present the identified motifs at two levels of granularity: individual words and multi-word phrases or even entire sentences. We specifically extracted word-level motifs from one-hop neighbor subgraphs to visualize word-level motifs. As shown in Table 16, we selected the top 20% of tokens based on their motif scores for visualization. Similarly, for visualizing higher-level motifs (phrases/sentences) in Table 17, we extracted them from two-hop subgraphs, with the top-k ratio set to 2% for display.

Table 16: Samples of words explanation motifs.

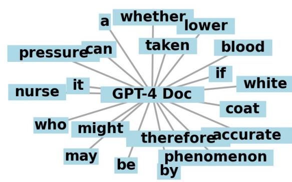
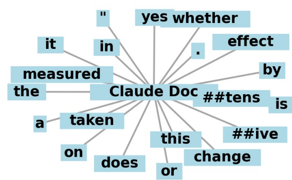
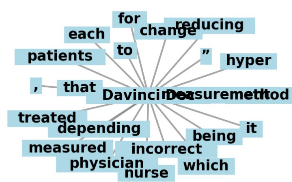
	Graph Motifs	Words Mapping
GPT-4		Answer: Yes, **blood** **pressure** readings **can** vary in treated hypertensive patients based on **who** measures **it.** **A** **phenomenon** known as "white **coat** hypertension" **may** cause higher readings **if** measured **by** **a** physician due to
Claude-3		Answer: **Yes,** blood pressure readings can differ when **measured** **by** **a** physician compared to **a** nurse **in** treated hypertensive patients. **This** phenomenon, known as **the** **"white-coat** **effect,"** **is** attributed to patient anxiety and can
Davinci		In **each** case, **to** assume **that** what is **being** **measured** is blood pressure is **incorrect.** Note: The issue is not what constitutes the "gold standard" of

Table 17: Samples of phase explanation motifs.

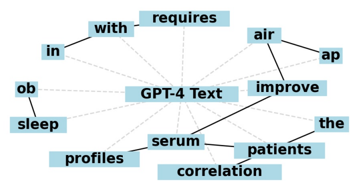
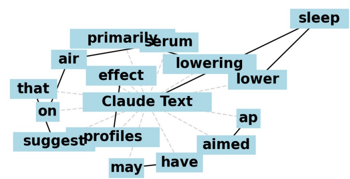
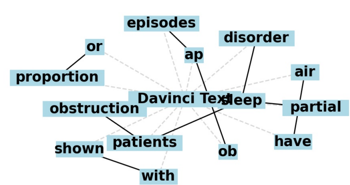
	Graph Motifs	Words Mapping
GPT-4		Answer: Research indicates that airway surgery can potentially improve serum lipid profiles in patients with obstructive sleep apnea. However, a definite conclusion requires further investigation as the correlation can be influenced by various factors, including the patients' lifestyle and
Claude-3		Question: Does airway surgery lower serum lipid levels in obstructive sleep apnea patients? Answer: Airway surgery for obstructive sleep apnea (OSA) is not primarily aimed at lowering serum lipid levels. While some studies suggest that treating OSA may have a positive impact on lipid profiles, the effect of airway
Davinci		Answer: Yes Obstructive sleep apnea (OSA) is a sleep-related breathing disorder, characterized by repeated episodes of complete or partial upper airway obstruction. Recent studies have shown that weight gain occurs in a high proportion of patients with OSA, especially among those

Table 18: MGT detection ACC performance comparisons with SOTA methods on HGT and MGT on M4 dataset. The best results are shown in bold font. The second-best results are shown underlined.

	DaVinci			Cohere			Dolly			BloomZ			
Method	reddit	peerread	arxiv	reddit	peerread	arxiv	reddit	peerread	arxiv	reddit	peerread	arxiv	Avg.
Likelihood	0.89	0.77	0.40	0.95	0.78	0.89	0.63	0.65	0.71	0.56	0.34	0.72	0.69
Rank	0.57	0.51	0.45	0.56	0.54	0.52	0.50	0.53	0.57	0.55	0.51	0.53	0.53
LogRank	0.83	0.77	0.40	0.94	0.81	0.90	0.75	0.69	0.73	0.71	0.39	0.77	0.72
Entropy	0.78	0.71	0.37	0.73	0.53	0.58	0.54	0.46	0.58	0.63	0.34	0.62	0.57
NPR	0.67	0.74	0.49	0.65	0.83	0.53	0.51	0.52	0.61	0.52	0.71	0.55	0.61
LRR	0.86	0.95	0.50	0.68	0.94	0.63	0.75	0.82	0.66	0.75	<u>0.98</u>	0.59	0.76
DetectGPT	0.56	0.53	0.35	0.63	0.60	0.47	0.54	0.46	0.45	0.58	0.57	0.62	0.53
Fast-DetectGPT	<u>0.97</u>	**1.00**	0.46	<u>0.96</u>	<u>0.99</u>	**0.98**	0.90	<u>0.99</u>	0.82	0.43	0.51	0.69	0.81
DNAGPT	0.75	0.47	0.36	0.90	0.47	0.86	0.51	0.53	0.54	0.45	0.49	0.57	0.58
Binoculars	**0.98**	**1.00**	0.51	**0.98**	0.96	**0.98**	0.83	<u>0.99</u>	<u>0.87</u>	0.58	0.62	0.77	0.84
Glimpse	0.77	0.95	0.51	0.95	0.88	1.00	0.64	0.68	0.75	0.52	0.43	0.88	0.75
GPTZero	0.86	<u>0.99</u>	0.36	0.84	0.92	0.65	0.76	0.58	0.50	0.61	0.53	0.46	0.67
RoBERTa-QA	0.93	1.00	0.55	0.95	0.97	0.89	0.95	0.55	0.71	0.50	0.50	0.52	0.75
Radar	0.84	0.88	0.57	0.87	0.85	0.60	0.66	0.77	0.53	0.80	0.79	0.30	0.71
DeTeCtive	0.90	0.85	**0.95**	0.84	0.76	<u>0.95</u>	**0.96**	0.75	**0.98**	<u>0.94</u>	0.89	<u>0.92</u>	<u>0.89</u>
LM^2OTIFS	<u>0.97</u>	**1.00**	<u>0.87</u>	**0.98**	**1.00**	0.94	<u>0.95</u>	**1.00**	0.77	**1.00**	**1.00**	**0.95**	**0.95**

Table 19: MGT detection ACC performance comparisons with SOTA methods on HGT and MGT on RAID dataset. The best results are shown in bold font. The second-best results are shown underlined.

	Llama			GPT-4			MPT			Mistral			
Method	recipe	poetry	review	recipe	poetry	review	recipe	poetry	review	recipe	poetry	review	Avg.
Likelihood	0.83	0.78	0.76	0.82	0.68	0.75	0.27	0.69	0.54	0.45	0.76	0.73	0.67
Rank	0.51	0.53	0.54	0.51	0.53	0.54	0.50	0.53	0.50	0.50	0.53	0.54	0.52
LogRank	0.79	0.81	0.79	0.80	0.64	0.78	0.30	0.63	0.44	0.43	0.77	0.78	0.66
Entropy	0.78	0.47	0.61	0.76	0.49	0.60	0.29	0.65	0.62	0.55	0.68	0.65	0.60
NPR	0.92	0.50	0.94	0.73	0.50	0.76	0.56	0.53	0.54	0.58	0.53	0.84	0.66
LRR	0.91	0.81	0.90	0.78	0.60	0.74	0.57	0.53	0.56	0.66	0.61	0.86	0.71
DetectGPT	0.51	0.77	0.73	0.51	0.61	0.65	0.44	0.48	0.46	0.51	0.52	0.56	0.56
Fast-DetectGPT	0.92	0.94	0.96	0.96	0.79	0.80	0.39	0.63	0.41	0.48	0.79	0.64	0.73
DNAGPT	0.76	0.69	0.58	0.68	0.70	0.59	0.29	0.54	0.35	0.40	0.52	0.70	0.57
Binoculars	**1.00**	**0.98**	0.95	**0.99**	0.81	0.95	0.43	0.62	0.68	0.76	0.72	0.65	0.80
Glimpse	0.93	0.77	0.95	0.93	0.60	0.79	0.70	0.59	0.75	0.81	0.67	0.84	0.78
GPTZero	0.74	0.47	0.73	0.61	0.46	0.73	0.53	0.52	0.57	0.58	0.57	0.51	0.59
RoBERTa-QA	0.85	0.50	0.96	0.76	0.50	0.83	0.46	0.53	0.69	0.44	0.55	0.69	0.65
Radar	0.58	0.59	0.86	0.63	0.57	0.86	0.59	0.73	0.59	0.64	**0.89**	0.63	0.68
DeTeCtive	**1.00**	0.95	0.94	0.97	0.96	0.97	0.91	**0.90**	**0.95**	0.87	0.88	0.90	0.93
LM^{2}OTIFS	**1.00**	**0.98**	**0.97**	**0.99**	**1.00**	**1.00**	**0.95**	0.84	0.90	**0.94**	0.88	**0.92**	**0.95**

Table 20: MGT detection ACC performance comparisons with SOTA methods on HGT and MGT on Yelp, Essay, and Creative dataset. The best results are shown in bold font. The second-best results are shown underlined.

	Claude3-Sonnet			Claude3-Opus			Gemini			
Method	Yelp	Essay	Creative	Yelp	Essay	Creative	Yelp	Essay	Creative	Avg.
Likelihood	0.61	0.96	0.83	0.61	0.97	0.91	0.56	0.97	0.69	0.79
Rank	0.51	0.54	0.51	0.50	0.55	0.51	0.50	0.55	0.51	0.52
LogRank	0.57	0.91	0.79	0.54	0.93	0.89	0.54	0.94	0.68	0.75
Entropy	0.60	0.87	0.69	0.58	0.92	0.71	0.53	0.85	0.53	0.70
NPR	0.62	0.67	0.80	0.62	0.58	0.68	0.50	0.57	0.56	0.62
LRR	0.55	0.90	0.78	0.52	0.91	0.73	0.45	0.58	0.56	0.66
DetectGPT	0.49	0.68	0.69	0.44	0.62	0.69	0.42	0.66	0.62	0.59
Fast-DetectGPT	0.66	**1.00**	0.88	0.72	0.99	0.93	0.60	**0.98**	0.69	0.83
DNAGPT	0.54	0.66	0.66	0.54	0.71	0.67	0.53	0.77	0.64	0.64
Binoculars	0.69	**1.00**	0.94	0.77	**1.00**	0.97	0.68	0.97	**0.78**	0.87
Glimpse	0.69	**1.00**	0.86	0.69	0.97	0.90	0.59	0.96	0.74	0.82
GPTZero	0.63	0.66	0.78	0.61	0.65	0.86	0.59	0.36	0.66	0.64
RoBERTa-QA	0.72	0.86	0.79	0.82	0.87	0.93	0.81	0.86	0.72	0.82
Radar	0.62	0.94	0.84	0.64	0.95	0.91	0.64	0.96	0.74	0.80
DeTeCtive	0.98	0.86	0.97	0.99	0.79	0.96	0.97	0.85	0.77	0.90
LM^{2}OTIFS	**0.99**	0.99	**0.98**	**1.00**	0.99	**0.98**	**0.99**	0.97	0.77	**0.96**

Table 21: MGT detection AUC performance comparisons with SOTA methods on HGT and MGT on M4 dataset. The best results are shown in bold font. The second-best results are shown underlined.

	DaVinci			Cohere			Dolly			BloomZ			
Method	reddit	peerread	arxiv	reddit	peerread	arxiv	reddit	peerread	arxiv	reddit	peerread	arxiv	Avg.
Likelihood	0.98	0.83	0.27	0.96	0.78	0.96	0.93	0.60	0.80	0.70	0.47	0.78	0.76
Rank	0.92	0.94	0.45	0.90	0.82	0.81	0.72	0.50	0.69	0.88	0.72	0.88	0.77
LogRank	0.98	0.96	0.28	0.97	0.90	0.97	0.93	0.65	0.79	0.84	0.58	0.85	0.81
Entropy	0.86	0.58	0.23	0.76	0.61	0.58	0.76	0.51	0.61	0.83	0.49	0.69	0.63
NPR	0.98	**1.00**	0.28	0.97	**1.00**	0.97	0.86	0.97	0.80	0.86	0.97	0.86	0.88
LRR	0.97	**1.00**	0.36	0.97	**1.00**	0.97	0.98	0.92	0.74	0.98	**1.00**	0.93	0.90
DetectGPT	0.59	0.74	0.29	0.72	0.76	0.43	0.61	0.54	0.42	0.73	0.71	0.63	0.60
Fast-DetectGPT	0.99	**1.00**	0.48	0.99	**1.00**	0.99	0.97	**1.00**	0.90	0.37	0.52	0.75	0.83
DNAGPT	0.84	0.27	0.32	0.94	0.35	0.93	0.72	0.55	0.70	0.47	0.11	0.69	0.57
Binoculars	**1.00**	**1.00**	0.51	0.98	**1.00**	**1.00**	0.98	**1.00**	0.95	0.53	0.66	0.85	0.87
Glimpse	0.92	**1.00**	0.51	0.98	0.96	**1.00**	0.83	0.81	0.91	0.66	0.42	**0.98**	0.83
GPTZero	0.86	0.99	0.36	0.84	0.92	0.65	0.76	0.58	0.50	0.61	0.53	0.46	0.67
RoBERTa-QA	0.99	**1.00**	0.94	0.99	**1.00**	**1.00**	0.98	0.95	0.99	0.61	0.38	0.66	0.87
Radar	0.95	**1.00**	0.48	0.97	**1.00**	0.78	0.79	0.92	0.43	0.90	0.88	0.52	0.80
DeTeCtive	0.96	0.85	**0.98**	0.89	0.88	0.98	0.96	0.86	**1.00**	0.96	0.96	**0.98**	0.94
LM^{2}OTIFS	0.99	**1.00**	0.94	**1.00**	**1.00**	0.98	**0.99**	**1.00**	0.85	**1.00**	**1.00**	**0.98**	**0.98**

Table 22: MGT detection AUC performance comparisons with SOTA methods on HGT and MGT on RAID dataset. The best results are shown in bold font. The second-best results are shown underlined.

	Llama			GPT-4			MPT			Mistral			
Method	recipe	poetry	review	recipe	poetry	review	recipe	poetry	review	recipe	poetry	review	Avg.
Likelihood	0.99	0.86	0.98	0.98	0.72	0.95	0.38	0.67	0.53	0.64	0.80	0.71	0.77
Rank	0.88	0.79	0.97	0.67	0.65	0.87	0.45	0.92	0.83	0.45	0.93	0.90	0.78
LogRank	0.99	0.87	0.98	0.97	0.69	0.94	0.38	0.73	0.60	0.64	0.81	0.74	0.78
Entropy	0.94	0.63	0.92	0.91	0.59	0.77	0.35	0.72	0.61	0.59	0.80	0.71	0.71
NPR	0.99	0.87	0.98	0.97	0.70	0.93	0.39	0.74	0.61	0.64	0.82	0.74	0.78
LRR	0.98	0.88	0.98	0.94	0.60	0.83	0.44	0.83	0.84	0.62	0.89	0.84	0.81
DetectGPT	0.52	0.82	0.83	0.55	0.63	0.75	0.29	0.45	0.45	0.48	0.45	0.55	0.56
Fast-DetectGPT	0.99	0.97	0.97	0.99	0.88	0.99	0.50	0.61	0.51	0.70	0.77	0.65	0.79
DNAGPT	0.96	0.75	0.95	0.80	0.75	0.88	0.38	0.55	0.49	0.57	0.72	0.60	0.70
Binoculars	0.99	0.99	0.97	**1.00**	0.98	**0.99**	0.55	0.66	0.59	0.72	0.79	0.68	0.83
Glimpse	**1.00**	0.87	0.97	0.99	0.60	0.88	0.69	0.63	0.84	0.86	0.74	0.92	0.83
GPTZero	0.74	0.47	0.73	0.61	0.46	0.73	0.53	0.52	0.57	0.58	0.57	0.51	0.59
RoBERTa-QA	0.95	0.94	0.96	0.82	0.83	0.95	0.45	0.73	0.63	0.31	0.65	0.54	0.73
Radar	0.98	0.85	0.89	0.99	0.81	0.87	0.95	0.83	0.74	0.80	0.86	0.63	0.85
DeTeCtive	**1.00**	0.95	0.96	0.99	0.96	0.99	0.92	**0.91**	**0.99**	0.91	0.90	0.97	0.95
LM^{2}OTIFS	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**0.99**	0.90	0.95	**0.99**	**0.94**	**0.98**	**0.98**

Table 23: MGT detection ACC performance comparisons with SOTA methods on HGT and MGT on Yelp, Essay, and Creative dataset. The best results are shown in bold font. The second-best results are shown underlined.

	Claude3-Sonnet			Claude3-Opus			Gemini			
Method	Yelp	Essay	Creative	Yelp	Essay	Creative	Yelp	Essay	Creative	Avg.
Likelihood	0.73	0.94	0.94	0.72	**1.00**	**0.99**	0.55	**0.99**	0.76	0.85
Rank	0.54	0.85	0.85	0.49	<u>0.99</u>	0.92	0.39	<u>0.97</u>	0.65	0.74
LogRank	0.69	0.93	0.93	0.68	**1.00**	0.98	0.50	**0.99**	0.74	0.83
Entropy	0.64	0.83	0.83	0.57	0.95	0.88	0.42	0.91	0.57	0.73
NPR	0.68	<u>0.99</u>	0.94	0.66	0.99	0.98	0.49	0.98	0.76	0.83
LRR	0.54	**1.00**	0.88	0.52	**1.00**	0.95	0.39	**0.99**	0.70	0.77
DetectGPT	0.53	0.75	0.71	0.43	0.74	0.78	0.37	0.80	0.64	0.64
Fast-DetectGPT	0.73	**1.00**	0.94	0.81	**1.00**	**0.99**	0.68	**0.99**	**0.79**	0.88
DNAGPT	0.67	0.94	0.86	0.70	0.94	0.93	0.58	0.95	0.75	0.81
Binoculars	0.79	**1.00**	<u>0.99</u>	0.87	**1.00**	**1.00**	0.73	<u>0.99</u>	0.79	<u>0.91</u>
Glimpse	0.78	**1.00**	0.90	0.83	**1.00**	0.96	0.74	**1.00**	0.78	0.89
GPTZero	0.63	0.66	0.78	0.61	0.65	0.86	0.59	0.36	0.66	0.64
RoBERTa-QA	0.92	0.95	0.94	0.96	0.98	0.97	0.96	0.94	<u>0.78</u>	0.93
Radar	0.58	0.93	<u>0.99</u>	0.68	<u>0.99</u>	0.97	0.70	**0.99**	0.76	0.84
DeTeCtive	<u>0.98</u>	0.86	0.96	<u>0.99</u>	0.79	**0.99**	<u>0.99</u>	0.85	0.76	<u>0.91</u>
LM^{2}OTIFS	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**0.99**	**1.00**	**0.99**	<u>0.78</u>	**0.97**

Table 24: Statistics of text covered by explanation motifs on RAID dataset. We report how many HGT/MGT contain tokens/token-tokens in explanation motifs. The sparsity of the explanation motifs is 0.05%.

	recipes		book		poetry		review	
Statistic	HGTs	MGTs	HGTs	MGTs	HGTs	MGTs	HGTs	MGTs
Tokens(Nodes)	1100	458	4093	1116	2892	760	2674	1560
Token-Token(Edges)	3519	2567	8583	4163	7731	3452	5100	5791
Ratio(Nodes/Edges)	0.31	0.18	0.48	0.27	0.37	0.22	0.52	0.27