M-RangeDetector: Enhancing Generalization in Machine-Generated Text Detection through Multi-Range Attention Masks

Kaijie Jiao[1],**Quan Wang**[2], **Licheng Zhang**[1], **Zikang Guo**[1]
Zhendong Mao[1]*
[1]University of Science and Technology of China, Hefei, China
[2]MOE Key Laboratory of Trustworthy Distributed Computing and Service,
Beijing University of Posts and Telecommunications, Beijing, China
jiaokaijie@mail.ustc.edu.cn, zdmao@ustc.edu.cn

Abstract

The increasing capability and widespread usage of large language models (LLMs) highlight the desirability of automatic detection of machine-generated text. Existing supervised detectors often overfit within their training domains, as they have primarily learned domain-specific textual features, such as word frequency, syntax, and semantics. In this paper, we introduce a domain-independent feature, namely the difference of writing strategy between LLMs and human, to improve the out-of-domain generalization capability of detectors. LLMs focus on the preceding range tokens when generating a token, while human consider multiple ranges, including bidirectional, global, and local contexts. The attention mask influences the range of tokens to which the model can attend. Therefore, we propose a method called **M-RangeDetector**, which integrates four distinct attention masking strategies into a Multi-Range Attention module, enabling the model to capture diverse writing strategies. Specifically, with the global mask, band mask, dilated mask, and random mask, our method learns various writing strategies for machine-generated text detection. The experimental results on three datasets demonstrate the superior generalization capability of our method.

1 Introduction

Large language models (LLMs), such as ChatGPT (OpenAI, 2022), Claude (Anthropic, 2024), Deepseek (Guo et al., 2025), Qwen (Yang et al., 2024), and Llama (Dubey et al., 2024), have been rapidly advancing. The text generated by these models is becoming increasingly fluent and human-like, facilitating tasks such as question-answering and news reporting. However, this progress has also raised increasing concerns about the misuse of LLMs, including the spread of misinformation and academic dishonesty (Crothers et al., 2023). This highlights the need for automated detection of machine-generated text.

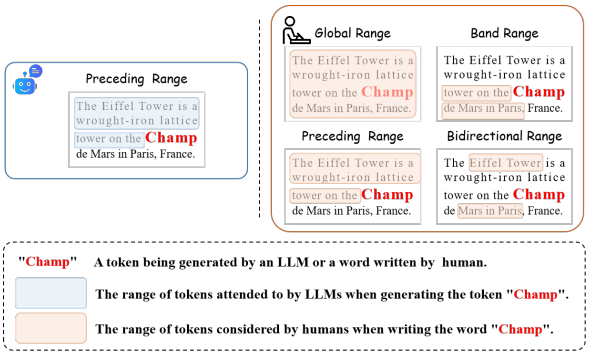

Figure 1: **Illustration of our motivation**. Both LLMs and human are writing the text, "The Eiffel Tower is a wrought-iron lattice tower on the Champ de Mars in Paris, France." When reaching the word "Champ", they exhibit different writing strategies. **Left**: LLMs focus solely on the preceding range of tokens. **Right**: In contrast, human writing strategies may involve global planning, local writing, iterative revision, and bidirectional thinking. As a result, human do not only consider the preceding tokens but may also take into account words from the global, local, and bidirectional contexts.

Machine-generated text detection is a binary classification task that determines whether a given text is authored by human or generated by LLMs. Early supervised methods (Guo et al., 2023) use RoBERTa (Yinhan et al., 2019) as an encoder to obtain token logits or representations for classification. Due to the growing capabilities of LLMs in semantic understanding, more recent approaches (Verma et al., 2023; Sarvazyan et al., 2024) have replaced RoBERTa with LLMs, and some methods (Chen et al., 2023; Wang et al., 2024a) directly use LLMs to generate "machine" or "human" labels for detection. (Guo et al., 2024) improves detection accuracy by conducting fine-grained classification of texts generated by different models and human-written texts. These methods perform exceptionally well in specific training

*Corresponding author: Zhendong Mao.

Findings of the Association for Computational Linguistics: ACL 2025, pages 8971–8983
July 27 - August 1, 2025

domains but struggle in out-of-distribution scenarios. This issue arises because existing methods primarily capture domain-specific features, such as word frequency, syntax, semantics, and sentiments (Guo et al., 2023; Wang et al., 2023). As these features can vary across domains, detectors often struggle with misclassification when applied to texts from new domains. In contrast, capturing domain-independent features is an effective method for improving the generalization capability of machine-generated text detection.

There exists an inherent distinction between the writing strategies of LLMs and human, specifically in the range of tokens they attend to when generating a token, no matter the domain to which a text belongs. As illustrated in Figure 1, when generating the token "Champ", LLMs are restricted to attending only to the preceding range of tokens, due to their masked attention mechanism. In contrast, when human write the word "Champ", they may take into account diverse ranges of words, such as global, local, and bidirectional contexts, depending on their intentions.

We propose the **M-RangeDetector** method to enhance the generalization capability of machine-generated text detection by modeling diverse writing strategies as domain-independent features. For a given text, we first obtain token representations using a Proxy LLM and then apply a Multi-Range Attention module to extract diversified writing strategy features for each token. These features are concatenated and fed into a classifier to determine whether the text is machine-generated. Specifically, the module integrates four distinct attention masking strategies to constrain different ranges during the computation of attention scores. The global mask captures relationships across all positions in the input sequence, the band mask focuses on local relationships between tokens on both sides, the dilated mask restricts a token's interaction with other tokens that are separated by a fixed interval, and the random mask randomly selects subsets of tokens from any position. The module can learn writing strategy features by calculating attention scores for each token within diverse specified ranges. This constraint ensures that the calculation of attention scores focuses on a range that closely aligns with the range a human would attend to during writing. Notably, our method does not require fine-tuning the Proxy LLM for token representation. Instead, it only requires training the Multi-Range Attention module and the classifier, which demonstrates that our approach is a resource-efficient solution for detecting machine-generated text. Our approach outperforms existing methods across multiple widely used datasets, including the Ghostbuster dataset, the M4 monolingual and multilingual datasets, and the OUTFOX dataset. Our contributions can be summarized as follows:

- In this paper, we introduce a domain-independent feature, namely the difference of writing strategy between LLMs and human, to improve the out-of-domain generalization capability of the detector.
- We introduce a Multi-Range Attention module that integrates four distinct attention masking strategies, including global, band, dilated, and random masking, to constrain the attention computation to different ranges, thereby capturing various writing strategies.
- The experimental results demonstrate that our method not only exhibits exceptional generalization ability in unseen domains but also performs well on newly emerging LLMs and multiple languages.

2 Related Work

LLM-Generated Text Mainstream LLMs, such as ChatGPT and Llama, primarily employ the autoregressive structure within the Transformer architecture (Vaswani, 2017) for continuous next-token prediction. This enables the generation of high-quality text that closely mimics human-like writing styles (Shlegeris et al., 2022; Mei et al., 2017).

Various Modified Attention Mechanisms In addition to the standard attention mechanism, several modified attention mechanisms have been proposed to improve model efficiency and performance. We introduce four attention mechanisms to control the range of token representations attending to other tokens. **Global attention** (Yu et al., 2018; Ronen et al., 2022) captures global contextual information by computing relationships between all positions in the input sequence. **Band attention** (Beltagy et al., 2020; Li et al., 2020) reduces the computational cost by focusing on local token relationships, making it suitable for tasks with strong local dependencies. **Dilated Attention** (Hoang et al., 2022) introduces flexibility by selecting tokens in a skipping manner, allowing the model to capture

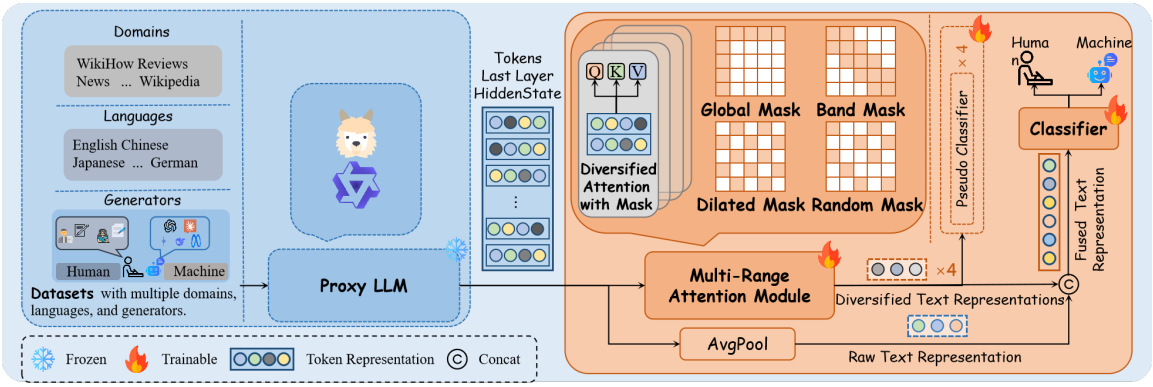

Figure 2: **The architecture of M-RangeDetector**. **Left**: We utilize Llama or Qwen as a Proxy LLM to obtain token representations. **Right**: We integrate the standard attention mechanism with four distinct attention masking strategies into a Multi-Range Attention module, composed of global, band, dilated, and random attention masks. The module is designed to capture distinct writing strategy features.

long-range dependencies efficiently. **Random attention** (Cattaneo et al., 2020) randomly selects subsets of tokens from any position.

Machine-Generated Text Detection Existing methods for detecting machine-generated text can be classified into zero-shot methods and supervised methods. Zero-shot methods rely on statistical features such as entropy, and perplexity for machine-generated text detection. For instance, GPT-Zero (Tian, 2023) classifies text based on perplexity and burstiness. DetectGPT (Mitchell et al., 2023) demonstrates that text generated by LLMs tends to fall within the negative curvature region of the log-probability function. Binoculars (Hans et al., 2024) employs two LLMs to compute the log perplexity of the text, utilizing cross-perplexity to detect LLM-generated content. These methods are training-free and easy to use, but their effectiveness is relatively limited (Taguchi et al., 2024).

Supervised approaches, such as fine-tuning RoBERTa (Yinhan et al., 2019) with external classifiers, are widely used to distinguish human-written text from machine-generated text (Yinhan et al., 2019; Guo et al., 2023). T5-Sentinel (Chen et al., 2023) addresses text detection by leveraging T5's (Ni et al., 2021) next-token prediction capabilities. DeTeCtive (Guo et al., 2024) introduces a multi-task auxiliary and multi-layer contrastive learning framework to learn writing styles from different models and human, enhancing generalization ability. Ghostbuster (Verma et al., 2023) incorporates multiple weaker language models (ranging from unigram models to LLMs) to capture token-level output probabilities and uses structured search for feature construction. LLMIXTIC (Sarvazyan et al., 2024) combines token-level probabilities from four Llama (Touvron et al., 2023) models for classification. Recent studies (Verma et al., 2023; Sarvazyan et al., 2024) have increasingly explored leveraging LLMs for feature extraction instead of relying on traditional models like RoBERTa.

3 Method

In this section, we provide a comprehensive overview of our method. In Section 3.1, we define the task of detecting machine-generated text and outline the key steps for utilizing the Proxy LLM to obtain token representations. In Section 3.2, we introduce the Multi-Range Attentions module that integrates four distinct attention masking strategies to restrict different ranges during attention computation. The architecture of **M-RangeDetector** is given in Figure 2.

3.1 Preliminary

Task Formulation This paper addresses the task of detecting machine-generated text across diverse domains, languages, and generators. Given a query text T consisting of L words, $T = \{t_1, t_2, \ldots, t_L\}$, our objective is to determine whether the text is authored by human or generated by LLMs.

Proxy LLM A sequence T is fed into the Proxy LLM (e.g., Llama or Qwen). The model generates hidden states for each token at each layer. We focus on the hidden states from the final layer, denoted as $H = \{h_1, h_2, \ldots, h_n\}$, where n represents the number of tokens after tokenization. These hidden states serve as high-level representations of the tokens. The key computational steps and equations for the hidden state H are as follows:

First, since the Proxy LLM we use is an autoregressive model, we obtain the masked attention scores A_m, formulated as follows:

$$A_m\left(Q,K,V,M\right)=\text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}+f_m\left(M\right)\right)V, \quad (1)$$

where Q, K, and V denote the query, key, and value matrices, respectively, which are obtained through linear transformations. d_k represents the dimensionality of the key vectors. M is a masking matrix where all elements in the lower triangle are set to 1, while those in the upper triangle are 0. The function f_m operates on the elements of matrix M, defined as follows:

$$f_m(M_{ij})=\begin{cases}0, & \text{if } M_{ij}=1\\ -\infty, & \text{otherwise}\end{cases}. \quad (2)$$

The H is recursively computed through a function f applied to the attention results at each layer:

$$H^{(i)}=f\left(A_m^{(i)},H^{(i-1)}\right), \quad i=1,2,\ldots,l, \quad (3)$$

where l represents the number of layers in the Proxy LLM, and $A_m^{(i)}$ denotes the attention output computed at layer i as defined in Eq. 1. The initial hidden state is set as $H^{(0)}=X$, corresponding to the input embeddings. The function f encompasses essential operations such as multi-head attention, layer normalization, and residual connections.

3.2 Multi-Range Attentions Module

Following the steps outlined in Section 3.1, we obtain token representations. However, a token's representation is generated by the Proxy LLM using a masking module that relies solely on preceding tokens. This approach learns contextual information from only a single range of tokens. To address this limitation of the Proxy LLM, we introduce four different attention masks: Global Mask, Band Mask, Dilated Mask, and Random Mask, which are distinct from those used by the Proxy LLM. We integrate these masks with the standard attention mechanism to implement four distinct types of attention. As shown in Figure 2, these components together form the Multi-Range Attention module. The configurations of attention masks in the processing of H are detailed in Algorithm 1.

Specifically, the H from the final layer, obtained in Section 3.1, are fed into four attention modules in parallel, yielding four diversified text representations: r_1, r_2, r_3, r_4. The computation of r_i is formulated as follows for $i=(1,2,3,4)$:

$$r_i=\text{AvgPool}\left(A_m\left(Q,K,V,M_i\right)\right), \quad (4)$$

Algorithm 1 The configurations of Global, Band, Dilated, and Random attention masks during the processing of H.

Require: Input token representations H, mask type $\mathcal{M}\in$ {Global, Band, Dilated, Random}, $M\in\mathbb{R}^{n\times n}$.
Initialize mask matrix $M\leftarrow\mathbf{0}\in\mathbb{R}^{n\times n}$
if $\mathcal{M}=$ Global Mask **then**
  Select global attention tokens $G\subseteq\{1,2,...,n\}$, where $|G|<n$
  $M_{i,j}\leftarrow 1$ if $i\in G$ or $j\in G$
  $M_{i,j}\leftarrow 0$ otherwise
else if $\mathcal{M}=$ Band Mask **then**
  Define band width w
  $M_{i,j}\leftarrow 1$ if $|i-j|\leq w$
else if $\mathcal{M}=$ Dilated Mask **then**
  Define dilation rate d
  $M_{i,j}\leftarrow 1$ if $|i-j| \mod d=0$
else if $\mathcal{M}=$ Random Mask **then**
  Define sparsity ratio r
  For each row i, randomly select rn indices and set $M_{i,j}\leftarrow 1$ for selected j
end if
return M

where Q, K, and V are obtained by applying a nonlinear transformation to H, M_1, M_2, M_3, M_4 correspond to the Global Mask, Band Mask, Dilated Mask, and Random Mask. A_m represents the masked attention, as defined in Equation 1. AvgPool is applied to aggregate the representations of all n tokens into a single text representation.

Additionally, the H are transformed into a unified text representation via the AvgPool operation for subsequent processing, referred to as the raw text representation r_s.

Finally, the four diversified text representations, along with the raw text representation, are concatenated to obtain the fused text representation: $\boldsymbol{R}=\{r_s,r_1,r_2,r_3,r_4\}$. The fused representation $\boldsymbol{R}$ is then fed into a classifier to determine whether the text is machine-generated. After backpropagation, the classification loss $\mathcal{L}_f$ is computed. Meanwhile, to further differentiate the parameters of the four attention mechanisms in the Multi-Range Attention Module, each of r_1, r_2, r_3, r_4 is individually input into a pseudo classifier, whose output does not play a decisive role. After backpropagation, the four pseudo-classifiers yield the corresponding losses $\mathcal{L}_1, \mathcal{L}_2, \mathcal{L}_3, \mathcal{L}_4$. Due to the impact of individual classification losses, each attention module undergoes more significant gradient updates, leading to a more diverse semantic space for each diversified text representation. The final loss function is computed as:

$$\mathcal{L}=\alpha_f\cdot\mathcal{L}_f+\textstyle\sum_{i=1}^{4}\alpha_i\cdot\mathcal{L}_i, \quad (5)$$

where cross-entropy loss is used for each loss, α_f and α_i (for $i = 1, 2, 3, 4$) are hyperparameters. The hyperparameters α_f and α_i are selected based on validation performance and are discussed in detail in Appendix B.

Notably, our method does not require fine-tuning the Proxy LLM. Instead, it only necessitates training the Differentiated Attentions module and the classifiers, which involve a minimal number of parameters. This makes our approach resource-efficient for detecting machine-generated text.

4 Experiments

We first evaluate the effectiveness of our method in in-domain and out-of-domain detection scenarios with unknown domains. Next, we assess the model's performance in detecting machine-generated text from unseen generators and in multilingual settings. Finally, we explore additional out-of-domain detection scenarios, including generating the same text with prompts different from those used in the training set, using three datasets containing texts written by non-native English speakers, and evaluate the robustness of our approach against paraphrase attacks, as well as its performance in the authorship attribution task.

4.1 Experimental Setup

4.1.1 Dataset

We utilize four widely recognized and challenging datasets to conduct comprehensive experiments.

Ghostbuster The Ghostbuster (Verma et al., 2023) dataset covers three domains and two models, providing a diverse range of detection scenarios. These include in-domain and out-of-domain settings, such as unknown domains, unseen generators, and varied prompts, as well as cases involving text written by non-native English speakers.

M4 The M4 (Wang et al., 2023) dataset consists of M4-monolingual and M4-multilingual, encompassing data from eight models, six domains, and nine languages. The OOD detection scenarios in M4-monolingual and M4-multilingual include cases involving unknown models and domains. Additionally, M4-multilingual also includes scenarios with unknown languages.

OUTFOX The machine-generated text in the testing data is paraphrased using OUTFOX (Koike et al., 2024) or DIPPER (Krishna et al., 2024).

TuringBench The TuringBench (Uchendu et al., 2021) dataset is designed for the authorship attribution task, which involves classifying texts into 20 distinct categories. Further details about these datasets can be found in Appendix A.

4.1.2 Evaluation metrics

In line with existing works, we employ Accuracy (Acc), the F1-score, and Average Recall (AvgRec) as our primary evaluation metrics. Accuracy is simple and effective for evaluating overall performance but fails to reflect minority class performance in imbalanced datasets. The F1-score considers both the precision and recall of the model, evaluating overall model performance by computing the harmonic mean of these two. AvgRec, the average recall for human-written (HumanRec) and machine-generated (MachineRec) text.

4.1.3 Baseline methods

We primarily compare our method with the Ghostbuster approach, which uses larger parameter LLMs for feature extraction, while we utilize smaller parameter LLMs. Additionally, we compare our method with DeTeCtive, which employs a multi-layer contrastive framework to learn the differences in writing styles between machines and human, whereas we focus on learning writing strategies. Furthermore, we perform a comprehensive comparison with several widely adopted methods.

Ghostbuster Ghostbuster (Verma et al., 2023) integrates multiple weaker language models, ranging from unigram models to Davinci with 17.5 billion parameters, to capture token-level probabilities as text features. It employs a structured search approach for feature construction and is the state-of-the-art model on the Ghostbuster dataset.

DeTeCtive DeTeCtive (Guo et al., 2024) employs multi-level contrastive learning to capture the differences in writing styles between machine-generated and human-written texts for AI-generated text detection. It is the state-of-the-art on both the M4 monolingual and multilingual datasets.

RoBERTa Directly fine-tuning RoBERTa (Yinhan et al., 2019) with an external classifier is a common approach for implementing binary classification tasks (Guo et al., 2023).

T5-Sentinel T5-Sentinel (Chen et al., 2023) addresses text detection by leveraging T5's next-token prediction capabilities.

Model	In-Domain				Out-of-Domain			
	All Domains	News	Creative Writing	Student Essays	News	Creative Writing	Student Essays	**Average**
Binoculars (Hans et al., 2024)	92.7	97.4	92.4	87.9	97.4	92.4	87.9	92.6
DetectGPT (Mitchell et al., 2023)	57.4	56.6	48.2	67.3	56.6	48.2	67.3	57.4
FastDetectGPT (Bao et al., 2023)	90.8	92.5	88.5	91.2	92.5	88.5	91.2	90.7
GPTZero (Tian, 2023)	93.1	91.5	93.1	83.9	91.5	93.1	83.9	89.5
RoBERTa (Guo et al., 2023)	98.1	99.4	97.6	97.4	88.3	95.7	71.4	85.1
T5 − Sentinel (Chen et al., 2023)	96.6	97.8	95.6	96.2	89.6	95.6	87.9	91.0
Ghostbuster (Verma et al., 2023)	99.0	99.5	98.4	99.5	97.9	95.3	**97.7**	97.0
M − RangeDetector (Ours)	**99.8**	**100.0**	**99.5**	**100.0**	**98.6**	**99.0**	**97.7**	**98.4**

Table 1: **The F1 scores in both In-Domain and Out-of-Domain scenarios on the Ghostbuster dataset**. The Out-of-Domain condition refers to using two of the three domains (news, creative writing, or student essays) as the training set while the remaining domain is used as the test set. Other results are derived from the (Verma et al., 2023). The best number is highlighted in bold, while the second-best one is underlined. The results shown above are obtained under the zero-shot setting. In contrast, the following results are based on supervised methods.

DetectGPT DetectGPT (Mitchell et al., 2023) utilizes probability curvature to detect whether a text is generated by LLMs. It observes that text sampled from LLMs tends to occupy the region of negative curvature in the model's logarithmic probability function.

Binoculars Binoculars (Hans et al., 2024) determines whether a text is machine-generated by computing the ratio between the raw perplexity and cross-perplexity, thus mitigating the influence of prompt words on perplexity and enhancing detection accuracy and robustness.

GPT-Zero GPT-Zero (Tian, 2023) is a commercial model that classifies text based on perplexity and burstiness.

4.1.4 Implementation details

In all experiments for our method, we freeze the Proxy LLM and only train the parameters of the Multi-Range Attention module and the classifier. It is worth noting that, with the exception of the experiments in Section 4.5, all other experiments are conducted using Llama-3 with a 1 billion (B) parameter size as the Proxy LLM. All experiments use the AdamW optimizer with a cosine annealing learning rate schedule. The peak learning rate is set to 3×10^{-5}, with a linear warm-up of 2000 steps and weight decay set to 1×10^{-5}. The maximum input token length is set to 512. We train for 20 epochs on a single NVIDIA A800 GPU, with a batch size of 64. For all comparison experiments, we either directly use the reported results from (Verma et al., 2023) and (Guo et al., 2024) or train and test using their open-source code with default settings, reporting the final results.

Method	M4-monolingual		M4-multilingual	
	AvgRec	F1	AvgRec	F1
Binoculars (Hans et al., 2024)	89.89	89.89	80.63	82.43
RoBERTa (Guo et al., 2023)	88.70	88.44	80.01	84.44
T5 − Sentinel (Chen et al., 2023)	84.01	81.08	76.21	68.99
DeTeCTive (Guo et al., 2024)	**98.44**	98.38	93.42	93.05
M − RangeDetector	98.42	**98.41**	**97.06**	**96.98**

Table 2: Experimental results on M4-monolingual and M4-multilingual.

4.2 Main Results and Analysis

We conduct extensive experiments on the Ghostbuster dataset to assess the generalization ability of **M-RangeDetector** across unknown domains. As shown in Table 1, we compare performance in both In-Domain and Out-of-Domain detection scenarios. Our method achieves an F1 score of 99.8% (All domains) and 98.4% (Average), outperforming the previous best approach by 0.8% and 1.4%, respectively. We further validate the superiority of our method on the M4-Mono dataset, which covers a broader range of domains and generators. As shown in Table 2, our method achieves an F1 score of 98.41%, matching the highest result attained by DeTeCTive. This result further demonstrates the generalization ability of our approach in OOD scenarios, including both unknown domains and newly emerging LLMs. Additionally, compared to the DeTeCTive method, our method not only demonstrates strong performance in English but also achieves an F1 score of 96.98% on the M4-multilingual dataset, exceeding DeTeCTive by 3.93%. This result highlights the generalization capability of our approach in OOD scenarios, including unseen languages. Overall, the comprehensive experimental results confirm the generalization ability of our method in OOD detection sce-

Model	Prompts (F1)	Claude (F1)	Lang8 (Acc.)	TOEFL 11 (Acc.)	TOEFL 91 (Acc.)
Binoculars	48.4	46.0	94.5	100.0	64.8
DetectGPT	70.8	64.2	98.6	**100.0**	63.7
FastDetectGPT	94.6	84.0	90.2	90.9	64.8
GPTZero	96.1	75.6	99.2	**100.0**	92.3
RoBERTa	97.4	87.8	98.6	98.1	96.7
T5 − Sentinel	94.6	84.1	98.9	99.6	97.8
Ghostuster	99.5	92.2	95.5	99.9	74.7
M − RangeDetector	**99.7**	**96.5**	**99.9**	**100.0**	**100.0**

Table 3: **The additional generalization results** include diverse prompting strategies, the unseen generator Claude, and three datasets containing texts written by non-native English speakers. All configurations remain identical to those in (Verma et al., 2023).

Attacker Detector	Non-attacked		DIPPER		OUTFOX	
	AvgRec	F1	AvgRec	F1	AvgRec	F1
Binoculars	49.3	33.0	55.4	45.2	89.1	89.0
FastDetectGPT	75.1	74.6	88.2	88.2	94.9	94.9
RoBERTa	90.8	90.7	94.3	94.4	73.9	68.3
T5 − Sentinel	99.0	98.9	96.1	96.1	94.8	94.8
OUTFOX	96.5	96.4	82.4	79.0	61.8	39.4
DeTeCTive	99.1	99.1	97.7	97.5	97.0	96.9
M − RangeDetector	**99.4**	**99.4**	**99.2**	**99.2**	**99.1**	**99.1**

Table 4: **The results of our approach against paraphrase attacks**, including DIPPER attack and OUTFOX attack on the OUTFOX dataset.

narios, including unknown domains, newly emerging LLMs, and unseen languages. We compare our method against all participating approaches in the SemEval competition[1], which focuses on the M4 dataset, to thoroughly demonstrate the superior performance of our method on the M4 dataset. Detailed results are provided in Appendix C.

4.3 More comprehensive experiments

As shown in Table 3, we further conduct additional generalization experiments, evaluating model performance under diverse prompting strategies and assessing its ability to detect texts generated by the unseen generator, Claude. We further evaluate model accuracy on three datasets of texts written by non-native English speakers. In all cases, our method achieves state-of-the-art performance, demonstrating its strong generalization across machine-generated texts with diverse prompts and human-written texts from different linguistic backgrounds.

We evaluate the robustness of our approach against paraphrase attacks using the OUTFOX dataset. As shown in Table 4, in the non-attacked text detection scenario, our method achieves the highest accuracy among all approaches, with both AvgRec and F1 scores reaching 99.4%. Under

[1]SemEval-2024, Task 8: https://www.codabench.org/competitions/1752

Method	M4-monolingual AvgRec	F1
M − RangeDetector	**98.42**	**98.41**
w/o All Attentions	92.14	92.20
w/o AvgPool	97.82	97.85
w/o Global Attention	96.75	96.83
w/o Band Attention	96.15	96.24
w/o Dilated Attention	96.66	96.74
w/o Random Attention	96.25	96.28
w/o Pseudo Classifier	97.63	97.68

Table 5: Ablation studies for the Multi-Range Attention module and the Pseudo Classifiers.

DIPPER and OUTFOX attacks, our method maintains exceptional performance, achieving AvgRec and F1 scores of 99.2% and 99.1%, respectively. In contrast, other methods experience significant performance degradation under attack. These results demonstrate the superior robustness of our approach across various attack scenarios. We conducte additional experiments on the GenAI workshop dataset (Wang et al., 2025) to validate our method's effectiveness in detecting text generated by recent LLMs. To evaluate whether **M-RangeDetector** has learned fine-grained features for distinguishing between generators, we conducted an authorship attribution experiment on the TuringBench dataset. Detailed results are provided in Appendix D.1 and D.2.

4.4 Ablation study

To systematically evaluate the contribution of each component in our method, we conduct a series of ablation studies on the M4-monolingual dataset, as summarized in Table 5. After removing AvgPool, the model achieves an F1 score of 97.85%, which is only 0.6% lower than the full model (M-RangeDetector). However, when we remove the multi-range attention module (w/o All Attentions), the performance drops by 6% compared to M-RangeDetector. This demonstrates that the multi-range attention module, which captures distinct writing strategies, plays a crucial role in enhancing the model's generalization capability, rather than relying on the ProxyLLM hidden states used in existing methods. To further investigate the effectiveness of the various attention mechanisms within the module, we systematically remove Global Attention, Band Attention, Dilated Band Attention, and Random Attention individually. The observed performance degradation across all cases underscores the critical role of each attention mechanism within the module. Notably, after removing the pseudo-

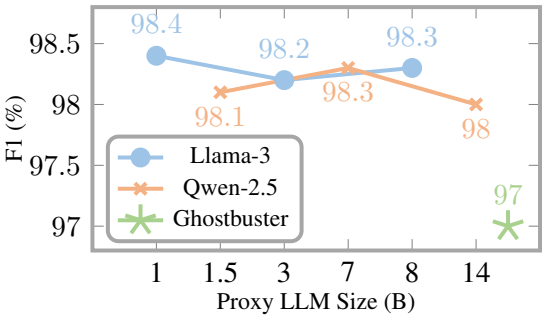

Figure 3: **The experimental results assessing the impact of Proxy LLM scale and type**. We used LLaMA and Qwen models of varying sizes, evaluated across out-of-domain scenarios on the Ghostbuster dataset.

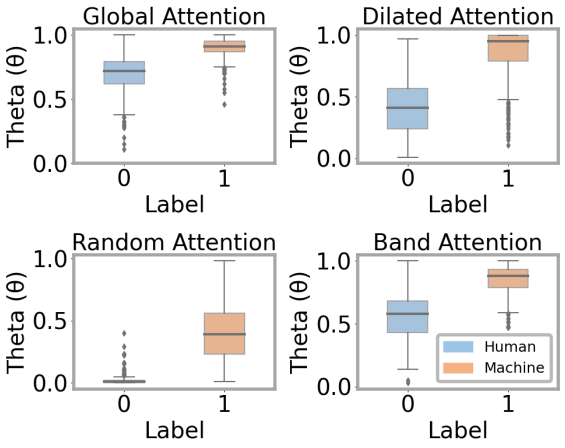

Figure 4: The figure compares θ value distributions between human-written and machine-generated texts across four attention mechanisms, using box plots to highlight the differences.

classifier, we observe a noticeable decline in performance, suggesting that the pseudo-classifier plays a role in enhancing the diversity of the semantic space for each diversified text representation.

4.5 Impact of Proxy LLM Scale and Type

To investigate whether different architectures of the Proxy LLM significantly impact the performance of our method, we conduct comprehensive experiments on the Ghostbuster dataset in out-of-domain scenarios, assessing the average F1 score across three settings using the Llama-3 and Qwen-2.5 model series. The Llama-3 models feature parameter sizes of 1B, 3B, and 8B, while the Qwen-2.5 models comprise 1.5B, 7B, and 14B. As shown in Figure 3, the results indicate that the Proxy LLM, whether based on Llama or Qwen, surpasses the previous SOTA model, Ghostbuster, which utilizes davinci with 17.5 billion parameters. Furthermore, the minimal performance gap between Llama and Qwen indicates that architectural differences among LLMs have a negligible impact on the effectiveness of our method.

Additionally, we investigate whether scaling up LLMs improves detection performance. A horizontal comparison reveals that increasing model size has a negligible impact, indicating that model scale is not a critical factor in determining the effectiveness of our method.

4.6 Exploratory on Attention Score

We perform an analysis of the attention scores from different attention in the Multi-Range Attention module. We introduce a special token, T_s, at the end of the original text and assign it global attention, enabling T_s to interact with all tokens (similarly to the [CLS] token in BERT (Devlin, 2018)). We only analyze the attention scores between T_s and other tokens. We define the breadth of attention, denoted as θ, to calculate the proportion of preceding tokens attended to when generating the last token. Specifically, the breadth of attention is the ratio of the number of tokens with attention scores greater than 3×10^{-5} to T_s relative to the total number of tokens. A larger θ value indicates that the model attends to a broader range of tokens. The value of θ is computed as follows:

$$\theta = \frac{\sum_{j=1}^{N} \mathbb{I}(A_{i,j} > 3 \times 10^{-5})}{N} \quad (6)$$

where $A_{i,j}$ is the attention score assigned by the model to token i when attending to token j, token i is T_s, N is the total number of tokens, $\mathbb{I}(\cdot)$ is the indicator function, which equals 1 if the condition inside holds true and 0 otherwise. A random sample of 1000 instances is drawn from the Ghostbuster dataset for experimentation. Each sample is processed through the model and computed using Equation 6, yielding four θ values. The results are then visualized using box plots. As shown in Figure 4, the within-group results indicate that, across all four diversified attentions, the θ for human-written text is lower than that for machine-generated text. We hypothesize that this discrepancy arises because human writing tends to focus on a limited number of key tokens, while machine-generated text, due to the central role of the attention mechanism in LLMs, exhibits a broader attention pattern. In the between-group results, we observe that the four attention mechanisms yield distinct distributions of the θ value within the same category. This suggests that the

four attention masks have learned different token representation patterns.

5 Conclusion

In this study, we propose the different writing strategies between LLMs and human as domain-independent features for enhancing generalization in machine-generated text detection. We introduce a novel approach called M-RangeDetector, which incorporates a Multi-Range Attention module to learn different writing strategies for classification. The module incorporates four distinct attention masking strategies to constrain different ranges during the computation of attention scores. Specifically, it employs global, band, dilated, and random masking strategies to capture the contextual representations of tokens across various token ranges, including global, local, and bidirectional ranges. Our method achieves state-of-the-art performance on three widely used benchmarks. Experimental results demonstrate that our approach not only exhibits exceptional generalization ability in unseen domains but also performs well on newly emerging LLMs and across multiple languages.

Limitations

Despite its effectiveness, M-RangeDetector has certain limitations. While contextual representations from tokens across diverse ranges differ significantly between human and LLMs, the results of the Authorship Attribution experiment indicate that although our method can reliably detect whether a text is machine-generated, it is not capable of accurately identifying the specific LLM that produced it. Furthermore, we have not conducted a detailed analysis of the specific contributions of different masking strategies to the model, nor have we investigated whether incorporating additional masking strategies can capture a broader range of writing strategies. In the future, we aim to develop more generalizable models for the Authorship Attribution task and further explore the impact of various attention masking strategies on the performance of our method. We will continue to advance research in machine-generated text detection tasks.

Acknowledgements

This work is supported by the National Natural Science Foundation of China No.62222212, 62232006 and 62376033.

References

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

Anthropic. 2024. Claude 3.5 sonnet.

Guangsheng Bao, Yanbin Zhao, Zhiyang Teng, Linyi Yang, and Yue Zhang. 2023. Fast-detectgpt: Efficient zero-shot detection of machine-generated text via conditional probability curvature. *arXiv preprint arXiv:2310.05130*.

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Matias D Cattaneo, Xinwei Ma, Yusufcan Masatlioglu, and Elchin Suleymanov. 2020. A random attention model. *Journal of Political Economy*, 128(7):2796–2836.

Yutian Chen, Hao Kang, Vivian Zhai, Liangze Li, Rita Singh, and Bhiksha Raj. 2023. Gpt-sentinel: Distinguishing human and chatgpt generated content. *arXiv preprint arXiv:2305.07969*.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53.

Evan N Crothers, Nathalie Japkowicz, and Herna L Viktor. 2023. Machine-generated text: A comprehensive survey of threat models and detection methods. *IEEE Access*, 11:70977–71002.

Jacob Devlin. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019. Eli5: Long form question answering. *arXiv preprint arXiv:1907.09190*.

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

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

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

Abhimanyu Hans, Avi Schwarzschild, Valeriia Cherepanova, Hamid Kazemi, Aniruddha Saha, Micah Goldblum, Jonas Geiping, and Tom Goldstein. 2024. Spotting llms with binoculars: Zero-shot detection of machine-generated text. *arXiv preprint arXiv:2401.12070*.

Dinh-Hieu Hoang, Gia-Han Diep, Minh-Triet Tran, and Ngan T H Le. 2022. Dam-al: Dilated attention mechanism with attention loss for 3d infant brain image segmentation. In *Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing*, pages 660–668.

Dongyeop Kang, Waleed Ammar, Bhavana Dalvi, Madeleine Van Zuylen, Sebastian Kohlmeier, Eduard Hovy, and Roy Schwartz. 2018. A dataset of peer reviews (peerread): Collection, insights and nlp applications. *arXiv preprint arXiv:1804.09635*.

Ryuto Koike, Masahiro Kaneko, and Naoaki Okazaki. 2024. Outfox: Llm-generated essay detection through in-context learning with adversarially generated examples. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 21258–21266.

Mahnaz Koupaee and William Yang Wang. 2018. Wikihow: A large scale text summarization dataset. *arXiv preprint arXiv:1810.09305*.

Kalpesh Krishna, Yixiao Song, Marzena Karpinska, John Wieting, and Mohit Iyyer. 2024. Paraphrasing evades detectors of ai-generated text, but retrieval is an effective defense. *Advances in Neural Information Processing Systems*, 36.

Nicolas Küchler, Emanuel Opel, Hidde Lycklama, Alexander Viand, and Anwar Hithnawi. 2024. Cohere: Managing differential privacy in large scale systems. In *2024 IEEE Symposium on Security and Privacy (SP)*, pages 991–1008. IEEE.

Jiaojiao Li, Ruxing Cui, Bo Li, Rui Song, Yunsong Li, Yuchao Dai, and Qian Du. 2020. Hyperspectral image super-resolution by band attention through adversarial learning. *IEEE Transactions on Geoscience and Remote Sensing*, 58(6):4304–4318.

Hongyuan Mei, Mohit Bansal, and Matthew R Walter. 2017. Coherent dialogue with attention-based language models. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pages 3252–3258.

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

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng-Xin Yong, Hailey Schoelkopf, et al. 2022. Crosslingual generalization through multitask finetuning. *arXiv preprint arXiv:2211.01786*.

Jianmo Ni, Gustavo Hernandez Abrego, Noah Constant, Ji Ma, Keith B Hall, Daniel Cer, and Yinfei Yang. 2021. Sentence-t5: Scalable sentence encoders from pre-trained text-to-text models. *arXiv preprint arXiv:2108.08877*.

OpenAI. 2022. Chatgpt: Optimizing language models for dialogue.

Alec Radford. 2018. Improving language understanding by generative pre-training.

Roi Ronen, Shahar Tsiper, Oron Anschel, Inbal Lavi, Amir Markovitz, and R Manmatha. 2022. Glass: Global to local attention for scene-text spotting. In *European Conference on Computer Vision*, pages 249–266. Springer.

Areg Mikael Sarvazyan, José-Ángel González, and Marc Franco-Salvador. 2024. Genaios at semeval-2024 task 8: Detecting machine-generated text by mixing language model probabilistic features. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 101–107.

Buck Shlegeris, Fabien Roger, Lawrence Chan, and Euan McLean. 2022. Language models are better than humans at next-token prediction. *arXiv preprint arXiv:2212.11281*.

Ayumi Taguchi, Daisuke Yoshimoto, Misako Kusakabe, Satoshi Baba, Akira Kawata, Yuichiro Miyamoto, Mayuyo Mori, Kenbun Sone, Yasushi Hirota, and Yutaka Osuga. 2024. Impact of human papillomavirus types on uterine cervical neoplasia. *Journal of Obstetrics and Gynaecology Research*, 50(8):1283–1288.

Edward Tian. 2023. Gptzero: An ai text detector.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Adaku Uchendu, Zeyu Ma, Thai Le, Rui Zhang, and Dongwon Lee. 2021. Turingbench: A benchmark environment for turing test in the age of neural text generation. *arXiv preprint arXiv:2109.13296*.

A Vaswani. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*.

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

Rongsheng Wang, Haoming Chen, Ruizhe Zhou, Han Ma, Yaofei Duan, Yanlan Kang, Songhua Yang, Baoyu Fan, and Tao Tan. 2024a. Llm-detector: Improving ai-generated chinese text detection with open-source llm instruction tuning. *arXiv preprint arXiv:2402.01158*.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Osama Mohammed Afzal, Tarek Mahmoud, Giovanni Puccetti, Thomas Arnold, et al. 2024b. Semeval-2024 task 8: Multidomain, multimodel and multilingual machine-generated text detection. *arXiv preprint arXiv:2404.14183*.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Toru Sasaki, et al. 2023. M4: Multi-generator, multi-domain, and multi-lingual black-box machine-generated text detection. *arXiv preprint arXiv:2305.14902*.

Yuxia Wang, Artem Shelmanov, Jonibek Mansurov, Akim Tsvigun, Vladislav Mikhailov, Rui Xing, Zhuohan Xie, Jiahui Geng, Giovanni Puccetti, Ekaterina Artemova, et al. 2025. Genai content detection task 1: English and multilingual machine-generated text detection: Ai vs. human. *arXiv preprint arXiv:2501.11012*.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. 2024. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.

Liu Yinhan, Ott Myle, Goyal Naman, Du Jingfei, Joshi Mandar, Chen Danqi, Levy Omer, and Lewis Mike. 2019. Roberta: A robustly optimized bert pretraining approach (2019). *arXiv preprint arXiv:1907.11692*, pages 1–13.

Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V Le. 2018. Qanet: Combining local convolution with global self-attention for reading comprehension. In *International Conference on Learning Representations*.

A Dataset Details

A.1 GhostBuster Dataset

The Ghostbuster dataset (Verma et al., 2023) serves as a benchmark for detecting AI-generated text across various domains. It spans three distinct domains: student essays, creative writing, and news articles, and includes two generative models, ChatGPT (OpenAI, 2022) and Claude (Anthropic, 2024). The dataset features five detection scenarios: In-Domain, Out-of-Domain, Prompts, Claude, and non-native English speakers. In the In-Domain scenario, data from all three domains (ChatGPT) are used for training and evaluated on a test set from the same distribution. News, Creative Writing, and Student Essays represent single-domain training and testing on the corresponding test set. In the Out-of-Domain scenario, two of the three domains (news, creative writing, or student essays) are used for training, with the remaining domain reserved for testing. The Avg reports the average results of the three individual out-of-domain experiments. The Prompts scenario tests text generated from prompts different from those used in training, essentially an out-of-domain detection challenge. The Claude scenario involves training on ChatGPT-generated texts and testing on Claude-generated texts to test the unseen model detection. Finally, the non-native English speakers category includes three datasets: Lang8, TOEFL-11, and TOEFL-91, which are written by non-native speakers, testing the ability to detect texts written by different linguistic backgrounds.

A.2 M4

The M4 (Wang et al., 2023) dataset is a large-scale, multi-domain, multi-model, and multilingual collection. It includes text from sources such as Wikipedia, WikiHow (Koupaee and Wang, 2018), Reddit (Fan et al., 2019), arXiv, and PeerRead (Kang et al., 2018). Using human-written prompts, models like ChatGPT (OpenAI, 2022), davinci-003 (Radford, 2018), Llama (Touvron et al., 2023), FLAN-T5 (Chung et al., 2024), Cohere (Küchler et al., 2024), Dolly-v2, and BLOOMz(Muennighoff et al., 2022) generate content in nine languages, including English, Chinese, and Russian. In the monolingual scenario, the test set features unseen domains and unseen AI-generated texts from GPT-4 (Achiam et al., 2023), which are further paraphrased by OUTFOX (Koike et al., 2024) to increase detection difficulty. In the multilingual scenario, the test set introduces novel languages not present in the training or validation sets, with AI-generated texts also paraphrased.

A.3 TuringBench

The dataset within TuringBench (Uchendu et al., 2021) comprises 200,000 samples across 20 distinct labels, including human-written texts and outputs from various AI text-generators such as GPT-1, GPT-2, GPT-3, GROVER, CTRL, XLM, XLNET, and others. The dataset was constructed by collecting 10,000 news articles, primarily in the domain of politics, from sources like CNN. These

articles, ranging from 200 to 400 words, were used to prompt AI text-generators to produce an additional 10,000 articles each, resulting in a total of 200,000 articles. The dataset supports two benchmark tasks: the Turing Test (TT), which is a binary classification problem to differentiate between human and machine-generated texts, and Authorship Attribution (AA), a multi-class classification problem aimed at identifying the specific neural language model that generated the texts.

B Hyperparameters Settings

We outline our hyperparameter settings for the weighted loss function. Our overall loss function is defined as:

$$L = \alpha_f \cdot L_f + \sum_{i=1}^{4} \alpha_i \cdot L_i$$

To ensure numerical stability and effective optimization, we first constrain the sum of all five hyperparameters to equal 1. This prevents any individual loss term from dominating excessively while maintaining a balanced gradient.

$$\alpha_f + \sum_{i=1}^{4} \alpha_i = 1.$$

Through empirical analysis, we found that L_f plays a more crucial role in optimization than L_i. To reflect this importance, we set $\alpha_f = 0.5$ and evenly distributed the remaining weight among L_i, assigning $\alpha_i = 0.125$ for $i = 1, 2, 3, 4$. Furthermore, to validate the stability of these choices, we conducted additional experiments by varying α_f within the range of 0.3, with corresponding adjustments to α_i. Our results indicate that setting $\alpha_f = 0.5 \pm 0.1, \alpha_i = 0.125 \pm 0.025$ for $i = 1, 2, 3, 4$ yields higher detection performance. Based on these findings, we ultimately set $\alpha_f = 0.5, \alpha_i = 0.125$.

C SemEval Competition Results

As shown in Table 2 in Section 4.2, we compare our model with Detective on AvgRecall and F1 scores. The results demonstrate that our method achieves SOTA performance in both monolingual and multilingual tasks, with a 3.93% improvement over Detective in the multilingual task. To further highlight the capabilities of our approach, we also compared it with the results of all participants in the SemEval competition, where the final ranking was based on the accuracy (Acc) of the Machine-Generated Text detection task. As shown in Tables 6 and 7 (due to space limitations, we only present the results of the top six participants and the baseline, for detailed results, refer to (Wang et al., 2024b)), our method also achieves SOTA performance in terms of accuracy (Acc). In the monolingual task, our method outperforms the top-ranked model by 0.84%, and in the multilingual task, it surpasses the top model by 0.99%. These results further demonstrate that our method achieves SOTA performance among all existing methods.

Team	Prec	Recall	F1-score	Acc
M-RangeDetector	**98.19**	98.63	**98.41**	**98.42**
dianchi	96.21	99.19	97.68	97.53
Genaios	96.11	98.03	97.06	96.88
USTC-BUPT	95.75	96.86	96.30	96.10
mail6djj	94.87	97.18	96.02	95.76
howudoin	93.48	98.12	95.74	95.42
idontknow	94.57	95.42	94.99	94.72
baseline	93.36	84.02	88.44	88.47

Table 6: The results of Monolingual

Team	Prec	Recall	F1-score	Acc
M-RangeDetector	**99.10**	95.05	**97.03**	**96.98**
USTC-BUPT	94.93	97.53	96.21	95.99
FI Group	94.28	98.00	96.10	95.85
KInIT	92.95	97.86	95.34	95.00
priyansk	90.70	98.14	94.28	93.77
L3i++	92.47	94.00	93.23	92.87
QUST	90.45	90.98	90.71	90.27
baseline	73.45	99.30	84.44	80.89

Table 7: The results of Multilingual.

model	Macro-F1	Acc
Binoculars	76.51	76.76
RoBERTa	70.34	72.56
T5-Sentinel	69.20	71.12
M-RangeDetector	**86.10**	**86.34**

Table 8: Performance comparison of our approach and baseline methods on the GenAI workshop dataset.

Rank	Team	Macro-F1	Acc
1	Advacheck	83.07	83.11
2	Unibuc-NLP	83.01	83.33
3	Fraunhofer SIT	82.80	82.89
-	**Baseline**	73.42	74.89

Table 9: Partial leaderboard results for the English subtask in the GenAI workshop.

Method	Precision	Accuracy	Recall	F1
Random Forest	58.93	61.47	60.53	58.47
SVM (3-grams)	71.24	72.99	72.23	71.49
WriteprintsRFC	45.78	49.43	48.51	46.51
Syntax-CNN	65.20	66.13	65.44	64.80
N-gram CNN	69.09	69.14	68.32	66.65
N-gram LSTM	66.94	68.98	68.24	66.46
OpenAI Detector	78.10	78.73	78.12	77.41
BertAA	77.96	78.12	77.50	77.58
BERT-Multinomial	80.31	80.78	80.21	79.96
RoBERTa-Multinomial	82.14	81.73	81.26	81.07
DeTeCtive	84.04	82.75	82.59	83.05
Ours	**85.39**	**86.08**	**85.77**	**85.41**

Table 10: The results of Authorship Attribution task on TuringBench dataset.

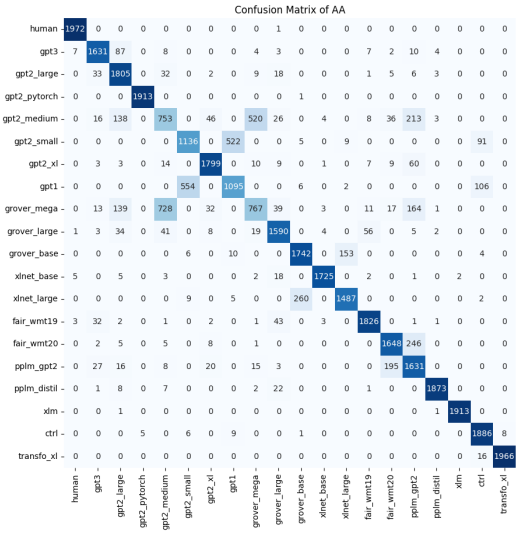

Figure 5: Confusion Matrix for Authorship Attribution on the TuringBench Dataset.

D Additional Experiments Results

D.1 GenAI Workshop Dataset

As shown in Table 8, our method achieves a Macro-F1 score of 86.10% and an Accuracy of 86.34% on the GenAI workshop dataset, outperforming other baseline methods (Best) by 10%. Additionally, as shown in Table 9, our model outperforms all teams in the GenAI workshop, ranking 1st on the English Subtask leaderboard and surpassing the top-ranked model by 3%. These results underscore the robustness of our method in detecting text from newer LLMs, maintaining strong and consistent performance even on the latest datasets.

D.2 The Authorship Attribution Task

The author attribution task requires not only determining whether a text is written by human or generated by LLMs, but also further identifying which generator generated the text. This constitutes a multi-class classification problem. To further evaluate the effectiveness of our method in the author attribution task, we conducted comprehensive experiments on the TuringBench dataset. The results, shown in Figure 10, demonstrate that our approach outperforms other methods, SOTA performance across multiple metrics, including Precision (84.56%), Accuracy (84.18%), Recall (83.67%), and F1 score (83.23%). However, the confusion matrix results in Figure 5 indicate that our method can distinguish whether a text is machine-generated. Nonetheless, there are still significant misclassifications within models from the same family (e.g., GPT). This is because while human and machine writing strategies differ greatly, writing strategies among machines can be quite similar, making it challenging for our method to accurately identify specific LLMs. In future work, we will further investigate the potential of our approach in author attribution tasks.