Beyond the Final Actor: Modeling the Dual Roles of Creator and Editor for Fine-Grained LLM-Generated Text Detection

Anonymous ACL submission

Abstract

The misuse of large language models (LLMs) requires precise detection of synthetic text. Existing works mainly follow binary or ternary classification settings, which can only distinguish pure human/LLM text or collaborative text at best. This remains insufficient for the nuanced regulation, as the LLM-polished human text and humanized LLM text often trigger different policy consequences. In this paper, we explore fine-grained LLM-generated text detection under a rigorous four-class setting. To handle such complexities, we propose RACE (Rhetorical Analysis for Creator-Editor Modeling), a fine-grained detection method that characterizes the distinct signatures of creator and editor. Specifically, RACE utilizes Rhetorical Structure Theory to construct a logic graph for the creator's foundation while extracting EDU-level features for the editor's style. Experiments show that RACE outperforms 11 baselines in identifying fine-grained types with low false alarms, offering a policy-aligned solution for LLM regulation.

1 Introduction

While the surge of Large Language Models (LLMs) (OpenAI, 2025b; Yang et al., 2025a) has revolutionized content creation and inspired a diverse range of downstream applications, the improper and malicious use of LLMs is also eroding the foundation of information credibility (Anwar et al., 2024). From the large-scale synthesis of misinformation (Hu et al., 2025) to unauthorized academic assistance (Goodier, 2025) and LLM-based identity fraud (FBI, 2025), the ease of generating high-quality synthetic text poses a severe challenge to our trust system, necessitating effective techniques to distinguish LLM-generated text from human-written text (Wu et al., 2025a; Liu et al., 2025).

LLM-generated text detection was primarily formulated as a binary classification task that judges

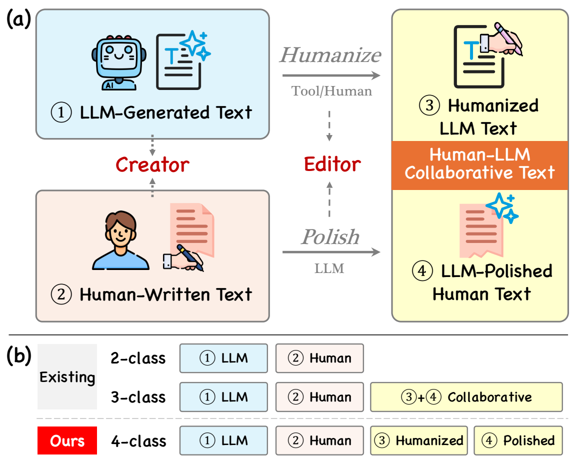

Figure 1: Illustration of our research scope. (a) A Creator-Editor framework for categorizing different types of texts in fine-grained LLM-generated text detection. (b) Comparison of the existing settings and the complex 4-class setting that we focus on in this paper.

whether the given text is generated by any LLM or by human (He et al., 2024; Wang et al., 2024). However, the binary setting oversimplifies real-world scenarios where text is often a product of human-LLM collaboration (Wang et al., 2025). For instance, people may ask LLMs to *polish* their original drafts for better readability (Yang et al., 2024a); or conversely, *humanize* LLM-generated outputs to evade detection (Masrour et al., 2025). Such collaborative processes yield hybrid texts that blend the characteristics of human and LLM generations, ultimately blurring the decision boundaries of conventional binary classifiers.

To address these complexities, recent studies shift towards fine-grained detection settings, typically by introducing a third "mixed" category (Zhang et al., 2024; Artemova et al., 2025; Saha and Feizi, 2025). Yet, even this ternary classification remains insufficient for the nuanced LLM use regulatory policies in specific domains like academic writing (Cahill et al., 2025). Under such policies, polished text is often considered legitimate writing assistance that requires no compulsory disclosure, while humanized text for bypassing detectors is often prohibited, as it brings improper advantages to cheating students and damages academic integrity.

In this paper, we study **a more rigorous four-class detection setting where the mixed category is explicitly separated into *LLM-Polished Human Text* and *Humanized LLM Text* classes**. Inspired by the conceptual framework from Bao et al. (2025), we analyze the four classes through the dual lenses of *creator* and *editor* and propose to enhance the modeling of creators' contribution for fine-grained detection. As illustrated in Figure 1, the creator establishes the basic elements and logical flow, while the editor controls the linguistic expression and surface-level style of these elements. For the pure human/LLM classes, differentiating the two roles is unnecessary; thus, conventional binary classifiers only need to obtain unified features to model human-LLM differences. In contrast, the creator-editor collaboration modes for the two mixed classes are quite different: LLM-Polished Human Text originates from a human creator's framework and is subsequently refined by an LLM's stylistic surface, whereas Humanized LLM Text has an LLM-generated foundation but is then edited by humans to perturb LLM traits. These divergent modes produce unique traits that are hard for unified features to capture, making it essential to look beyond the final actor and model the contributions of the creator and editor roles separately.

To address the four-class detection challenge, we propose the **R**hetorical **A**nalysis for **C**reator-**E**ditor Modeling (**RACE**) that explicitly models the distinct contributions of the creator and the editor. RACE is grounded in the argument that an editor's influence is primarily manifested in the linguistic expression, while the creator's identity is deeply rooted in the logical organization and argumentative progression of the content. To model the editor's role, RACE first segments the text into Elementary Discourse Units (EDUs) and extracts their semantic representations, which reflect surface-level linguistic choices and refinements. To model the creator's role, RACE utilizes Rhetorical Structure Theory (RST) (Mann and Thompson, 1988) to construct an EDU-based logical relation graph that characterizes the foundational organization of the text, highlighting the human-LLM creation differences stemming from their fundamental knowledge formation mechanisms. The graph is then processed by rhetoric-guided message passing to propagate information to capture complex rhetorical dependencies, which produces a root pooling representation for final prediction. Our contributions are as follows:

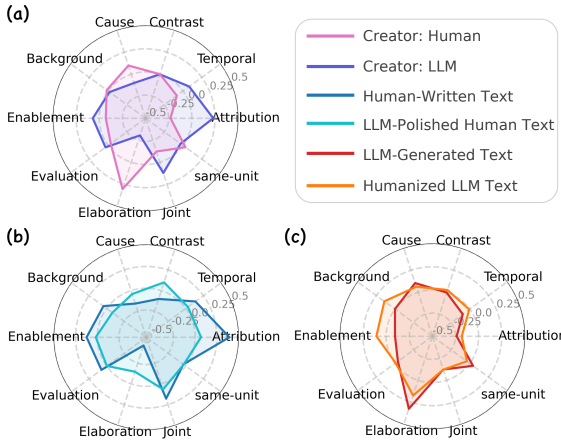

Figure 2: Distribution of RST relations. (a) Divergence of Creators: Human creators build deeper rhetorical hierarchies (*e.g.*, Attribution, Background), whereas LLMs produce flatter structures relying on surface-level relations (*e.g.*, Elaboration, Evaluation). (b) LLM-Polished: underlying human architecture persists. (c) Humanized: underlying LLM architecture persists.

- **Task:** We explore a refined four-class setting for LLM-generated text detection that better aligns with the nuanced requirements of contemporary LLM regulatory policies.
- **Method:** We propose RACE, a detection method that models the generative process through the dual lenses of creator and editor by leveraging Rhetorical Structure Theory.
- **Performance:** Extensive experiments demonstrate the superiority of RACE under the four-class fine-grained setting with low false alarms.

2 Preliminaries

2.1 Rhetorical Structure Theory

Rhetorical Structure Theory (RST) is a descriptive framework for natural text organization, originally proposed to analyze how coherent discourse is constructed (Mann and Thompson, 1988). RST models the hierarchical and functional dependencies between text spans, treating a text piece not as a linear sequence of words but as a structured tree of logical intentions.

The construction of a structured tree begins with segmenting the text into several spans called Elementary Discourse Units, which are typically clauses or phrases. Then the text spans are linked

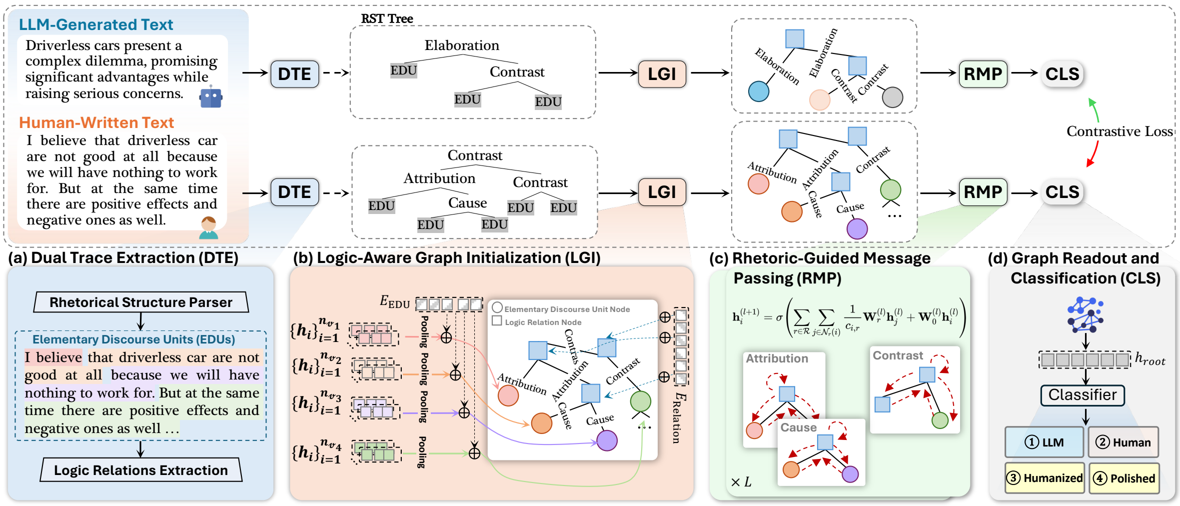

Figure 3: Overall architecture of **RACE**. Given a text piece, RACE **(a)** first captures both creator and editor traces through rhetorical structure construction and elementary discourse unit extraction. **(b)** These dual traces are then transformed into a logic-aware graph, where both linguistic expression and logical organization signals are encoded into node features via descendant span pooling and relation-aware projection. **(c)** Next, Rhetoric-Guided Message Passing propagates information through relation-specific aggregation with basis decomposition to capture complex rhetorical dependencies. **(d)** Finally, the global text representation is obtained via root pooling for classification.

through rhetorical relations (*e.g.*, Elaboration, Contrast, Cause), resulting in an RST tree. The RST tree can serve as a fingerprint of the creator's thought process. Specifically, we posit that human and machine creators exhibit distinct structural signatures. For humans, the writing process is inherently teleological, employing complex rhetorical relations such as clausal coordination to guide readers through a preset logical progression. In contrast, LLMs, driven by auto-regressive probability, prioritize informational density over narrative logic, resulting in superficial structural signatures (Reinhart et al., 2025). By modeling logical organization, we can capture the intrinsic differences in how humans and machines architect their narratives.

2.2 Motivating Analysis

We conducted a preliminary statistical analysis on the distribution of RST relations across the HART dataset (Bao et al., 2025). Specifically, we adopt the Z-score to measure the deviation of each relation's frequency. For an RST relation j in class k, the Z-score is calculated as $Z_{k,j} = (\bar{x}_{k,j} - \mu_j)/\sigma_j$, where $\bar{x}_{k,j}$ is the intra-class mean of relative frequency, and μ_j, σ_j are global mean and standard deviation. A value of $Z > 0$ indicates over-expression relative to the general population.

As visualized in Figure 2 (a), human creators show a significant over-expression in Attribution and Background, which aligns with the human tendency to cite sources, establish context, and ground arguments in external evidence. LLM creators, conversely, exhibit strong spikes in Elaboration, Evaluation, and Enablement, lacking the deep intertextual grounding found in human writing. In Figure 2 (b), even after LLMs' polishing, the text retains the high Attribution and Background features, which are more aligned with the human creator. Similarly, Figure 2 (c) shows that human editing fails to mask the underlying LLMs' logic, as Elaboration and Evaluation remain dominant.

These findings indicate that the subsequent editing operation generally preserves the underlying logic of the creator, which shows the possibility of separately modeling the unique characteristics of humans and LLMs as creators or editors. In the next section, we will introduce rhetorical structure information to model the dual roles for fine-grained LLM-generated text detection.

3 Proposed Method: RACE

To capture the dual trace of creator and editor for fine-grained LLM-generated text detection, we propose the logical-structure-aware detection framework, RACE. As illustrated in Figure 3, RACE consists of four key components: Dual Trace Extraction, Logic-Aware Graph Initialization , Rhetoric-Guided Message Passing, and Graph Readout and Classification. Through these modules, RACE models the generative process through the dual lenses of linguistic expression and logical organization to improve fine-grained detection performance.

3.1 Dual Trace Extraction

To transform unstructured raw text into a structured logic-aware representation, we utilize the end-to-end RST parser developed by Chistova (2024), which achieves superior performance in identifying hierarchical discourse dependencies.

Formally, the parsing process is defined as a mapping function $\mathcal{F}_{\text{parse}} : D \to \mathcal{T}$. Given an input text piece D, the parser outputs a binary constituency tree $\mathcal{T}$ that explicitly encodes the relation topology. In this structure, the leaf nodes constitute the sequence of EDUs $\mathcal{V}_{\text{edu}} = \{u_1, u_2, \ldots, u_{|\mathcal{V}_{\text{edu}}|}\}$, where each u_i aligns with a specific continuous text span $[s_i, e_i]$. Recursively, the internal nodes $\mathcal{V}_{\text{rel}} = \{v_1, v_2, \ldots, v_{|\mathcal{V}_{\text{rel}}|}\}$ capture the logical organization by assigning a specific rhetorical label $r \in \mathcal{R}$ (*e.g.*, Elaboration, Contrast) to the dependencies between sub-trees. The tree $\mathcal{T}$ serves as the foundational skeleton, which is subsequently transformed into a logic-aware multi-relational graph to enable rhetoric-guided message passing.

3.2 Logic-Aware Graph Initialization

Building upon the parsed tree $\mathcal{T}$, the text piece is formalized as a multi-relational graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{R})$, where $\mathcal{V} = \mathcal{V}_{\text{edu}} \cup \mathcal{V}_{\text{rel}}$. Each edge $e \in \mathcal{E}$ is represented as a triplet (u, r, v), preserving the explicit dependency structure where relation nodes govern their constituent EDUs.

Furthermore, a hybrid strategy combining descendant span pooling with information bottleneck projection is proposed to initialize non-leaf nodes with semantically-informed representations, thus going beyond surface-level relation labels to encode richer contextual information.

Descendant Span Pooling. For a text piece D with tokens $\{t_1, \ldots, t_K\}$, a pre-trained language model (PLM) is employed as the backbone to produce a sequence of contextualized embeddings $\mathbf{E} \in \mathbb{R}^{K \times d_{\text{PLM}}}$. The content representation $\mathbf{c}_i$ for any node $v_i \in \mathcal{V}$ is computed recursively:

$$\mathbf{c}_i = \begin{cases} \text{MeanPool}(\{\mathbf{e}_k\}_{k=s_i}^{e_i}), & \text{if } v_i \in \mathcal{V}_{\text{edu}} \\ \frac{1}{|\mathcal{D}(v_i)|} \sum_{u \in \mathcal{D}(v_i)} \mathbf{c}_u, & \text{if } v_i \in \mathcal{V}_{\text{rel}}, \end{cases} \tag{1}$$

where $\mathbf{e}_k$ is the k-th row of $\mathbf{E}$, and $\mathcal{D}(v_i) \subset \mathcal{V}_{\text{edu}}$ denotes the set of all descendant nodes in the subtree rooted at v_i. This strategy ensures that relation nodes are initialized with the global semantic centroid of the text segments they govern.

Information Bottleneck Projection. Raw semantic embeddings often contain surface-level lexical noise irrelevant to structural authorship analysis. To filter this redundancy, a dimension reduction strategy is adopted as an information bottleneck. Specifically, the PLM embeddings are projected into a compact structural space of dimension d_{feat}:

$$\begin{aligned} \mathbf{c}'_i &= \mathbf{c}_i + \mathbf{E}_{\text{type}}[\tau_i], \\ \mathbf{h}_i^{(0)} &= \text{Dropout}\left(\text{LN}\left(\mathbf{c}'_i \mathbf{W}_{\text{proj}} + \mathbf{b}_{\text{proj}}\right)\right), \end{aligned} \tag{2}$$

where $\tau_i \in \{0, 1\}$ indicates the node type (non-leaf or leaf), $\mathbf{E}_{\text{type}}$ is the learnable node type embedding table, $\mathbf{W}_{\text{proj}} \in \mathbb{R}^{d_{\text{PLM}} \times d_{\text{feat}}}$ and $\mathbf{b}_{\text{proj}}$ are the projection parameters, and LN signifies layer normalization. This compression forces the model to distill only the most salient features required for the subsequent rhetoric-guided message passing.

3.3 Rhetoric-Guided Message Passing

To learn the human-LLM differences over complex rhetorical dependencies, an L-layer Relational Graph Convolutional Network (RGCN; Schlichtkrull et al., 2018) is adopted on the logic-aware graph. Unlike vanilla GCNs that treat all edges uniformly (Kipf and Welling, 2017), RGCN assigns relation-specific transformation matrices, allowing the model to learn distinct propagation rules for different rhetorical logics.

Message Aggregation. In each layer l, the node representation is updated as:

$$\mathbf{h}_i^{(l+1)} = \sigma\left(\sum_{r \in \mathcal{R}} \sum_{j \in \mathcal{N}_r(i)} \frac{\mathbf{h}_j^{(l)} \mathbf{W}_r^{(l)}}{Z_{i,r}} + \mathbf{h}_i^{(l)} \mathbf{W}_0^{(l)}\right), \tag{3}$$

where $\sigma(\cdot)$ denotes the activation function, $\mathbf{W}_r^{(l)}$ is the relation-specific weight matrix for relation r in the l-th layer, $\mathcal{N}_r(i)$ is the set of neighbors under relation r, $Z_{i,r}$ is a normalization constant, and $\mathbf{W}_0^{(l)}$ handles the self-loop update.

Basis Decomposition for Regularization. Given the large number of fine-grained rhetorical relations, learning $\mathbf{W}_r^{(l)}$ for each relation leads to parameter explosion and overfitting. To constrain the weight space, basis decomposition is employed:

$$\mathbf{W}_r^{(l)} = \sum_{k=1}^{B} \alpha_{rk}^{(l)} \mathbf{V}_k^{(l)}, \tag{4}$$

where $\{\mathbf{V}_k^{(l)}\}_{k=1}^{B}$ is a set of shared basis matrices, and $\alpha_{rk}^{(l)}$ are learnable scalar coefficients unique to relation r. This technique forces the model to learn the "atomic" components of rhetorical logic, improving generalization on sparse relations.

Table 1: Quantitative comparison of detection methods under the 4-class setting. For RACE, we report the results across three runs using different seeds in the format of the $\text{mean}_{\pm\text{std}}$. **Bold** and underlined values denote the best and second-best performance, respectively.

Method	AUROC	TPR@1%FPR				
		Human-Written	LLM-Polished	LLM-Generated	Humanized	Avg
RoBERTa (Solaiman et al., 2019)	92.22	99.36	68.06	63.14	70.92	75.37
CoCo (Liu et al., 2023)	97.67	**99.68**	75.77	63.93	**79.43**	79.70
SeqXGPT (Wang et al., 2023)	89.87	98.38	15.23	14.32	31.68	39.90
DeTeCtive (Guo et al., 2024)	95.74	98.62	0.00	0.00	77.23	43.96
LF-Motifs (Kim et al., 2024)	**98.20**	96.68	69.61	67.01	75.62	77.23
$\text{Binoculars}_{\text{MLP}}$ (Hans et al., 2024)	79.15	29.49	7.34	4.37	5.50	11.70
$\text{Binoculars}_{\text{C-T}}$ (Bao et al., 2025)	50.03	0.00	0.00	0.00	0.00	0.00
F-DetectGPT (Bao et al., 2024)	61.70	0.00	3.37	26.27	0.09	7.70
$\text{F-DetectGPT}_{\text{MLP}}$ (Bao et al., 2024)	73.69	3.12	3.87	29.35	3.96	10.8
$\text{F-DetectGPT}_{\text{C-T}}$ (Bao et al., 2025)	49.93	0.00	0.00	0.00	0.00	0.00
TDT_{SVC} (West et al., 2025)	57.16	2.88	2.37	3.58	0.50	2.33
RACE (Ours)	$97.99_{\pm 0.13}$	$99.04_{\pm 0.40}$	$\mathbf{83.60}_{\pm 1.61}$	$\mathbf{74.18}_{\pm 0.95}$	$75.41_{\pm 1.03}$	$\mathbf{83.06}_{\pm 0.57}$

3.4 Graph Readout and Classification

As the logical structure is inherently hierarchical with a single root node v_{root} encompassing the entire text piece's rhetorical intent, a root pooling strategy is employed to capture the global text representation. The global representation $\mathbf{z}_{\mathcal{G}}$ is directly extracted from the root node's final hidden state:

$$\mathbf{z}_{\mathcal{G}} = \mathbf{h}^{(L)}_{v_{\text{root}}}. \tag{5}$$

Finally, the global representation $\mathbf{z}_{\mathcal{G}}$ is passed to a classification head:

$$\begin{aligned} \tilde{\mathbf{z}} &= \sigma(\text{Dropout}(\mathbf{z}_{\mathcal{G}})\mathbf{W}_{\text{in}} + \mathbf{b}_{\text{in}}), \\ \hat{y} &= \text{Softmax}(\text{Dropout}(\tilde{\mathbf{z}})\mathbf{W}_{\text{out}} + \mathbf{b}_{\text{out}}), \end{aligned} \tag{6}$$

where $\mathbf{W}_{\text{in}}, \mathbf{W}_{\text{out}}$ are weight matrices and $\mathbf{b}_{\text{in}}, \mathbf{b}_{\text{out}}$ are bias vectors, σ is a non-linear activation function, and $\hat{y}$ is the predicted probability.

Optimization. RACE is optimized using a joint loss that combines the supervised contrastive loss (Khosla et al., 2020) $\mathcal{L}_{\text{con}}$ and the cross-entropy loss $\mathcal{L}_{\text{ce}}$. The former is applied to the normalized feature representations, encouraging the model to learn a compact representation space. The joint loss function is $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{con}} + \mathcal{L}_{\text{ce}}$.

4 Experiments

4.1 Experimental Setup

Dataset. We use the HART (Bao et al., 2025) benchmark for evaluation due to its coverage of the desired categories. However, the official release only contains validation and test partitions. To enable supervised learning, we reorganized the data distribution (see Appendix A.1) and performed a train/val/test split at the 70:20:10 ratio using stratified sampling across diverse domains (*e.g.*, News, Writing, ArXiv, and Essay), which ensures the distribution consistency across all partitions.

Metrics. To evaluate the quality of the model's probability estimates independent of arbitrary decision thresholds, we prioritize metrics that assess the global ranking capability of the classifier rather than hard predictions. Specifically, we adopt:

- **Macro-Averaged AUROC**, which evaluates the probability that a randomly selected positive instance from any class is ranked higher than a randomly selected negative instance.
- **TPR@1% FPR** (True Positive Rate at the 1% False Positive Rate), which requires the detector to make precise judgments while avoiding false alarms (Tufts et al., 2025).

Baselines. Since there is no existing work adopting the 4-class setting, we establish baselines by adapting methods originally designed for two-/three-class settings. For a reasonable comparison, we cover 11 learning-based or metric-based methods and tailor them to the fine-grained setting (More details in Appendix A.3):

- **Learning-based Methods:** We include RoBERTa (Solaiman et al., 2019), CoCo (Liu et al., 2023), DeTeCtive (Guo et al., 2024), and LF-Motifs (Kim et al., 2024) and increase the number of entries of the classification head (*i.e.*, the fully connected layer) from 2 to 4. The selected baselines share designs similar to our method: CoCo and LF-Motifs also consider discourse information, and DeTecCtive adopts contrastive learning.

Table 2: Ablation results (mean$_{\pm\text{std}}$) of RACE. The *Bottleneck* represents the Information Bottleneck Projection in Eq. (2), the *Basis* represents the Basis Decomposition in Eq. (4), and *w/o Relation* means removing the relation types on edges and adopting vanilla GCN (Kipf and Welling, 2017).

Method	**AUROC**	**TPR@1%FPR**				
		Human-Written	LLM-Polished	LLM-Generated	Humanized	**Avg**
RACE	$97.99_{\pm0.13}$	$99.04_{\pm0.40}$	$83.60_{\pm1.61}$	$74.18_{\pm0.95}$	$75.41_{\pm1.03}$	$83.06_{\pm0.57}$
w/o CL	$97.73_{\pm0.44}$	$98.21_{\pm0.19}$	$78.07_{\pm1.06}$	$69.10_{\pm2.66}$	$73.43_{\pm1.59}$	$79.70_{\pm1.12}$
w/o Relation	$96.78_{\pm0.65}$	$97.42_{\pm0.75}$	$78.24_{\pm2.08}$	$65.35_{\pm9.19}$	$74.92_{\pm1.74}$	$78.98_{\pm2.95}$
w/o RGCN	$97.91_{\pm0.20}$	$98.92_{\pm0.19}$	$82.27_{\pm4.56}$	$65.35_{\pm6.71}$	$74.42_{\pm1.14}$	$80.24_{\pm2.07}$
w/o Bottleneck	$98.07_{\pm0.22}$	$98.54_{\pm0.26}$	$80.82_{\pm2.45}$	$74.68_{\pm0.86}$	$75.74_{\pm0.99}$	$82.45_{\pm0.60}$
w/o Basis	$97.22_{\pm0.74}$	$97.92_{\pm0.26}$	$83.39_{\pm3.20}$	$73.02_{\pm5.46}$	$74.58_{\pm1.14}$	$82.23_{\pm1.74}$

- **Metric-based Methods:** We include Binoculars (Hans et al., 2024), Fast-DetectGPT (F-DetectGPT) (Bao et al., 2024), and TDT (West et al., 2025), which typically produce scalars as predictions and thus are hard to directly extend to multi-class scenarios. Here, we conduct necessary modifications to obtain features via these methods by extracting the last-layer representation and appending a lightweight learnable MLP for four-class prediction.

Implementation Details. For RACE, we use RoBERTa-base (Liu et al., 2019) as the backbone and only fine-tune the last layer while keeping the preceding layers frozen. The extracted features are projected to a dimension of 128 to initialize node features. The graph component consists of an RGCN with $L = 2$ layers, a hidden dimension of 512, and 10 bases for parameter regularization. The temperature τ is 0.07 for supervised contrastive loss. We select the best validation checkpoint for testing. All experiments were conducted on a single NVIDIA RTX 4090 GPU.

4.2 Main Results

Table 1 presents the quantitative comparison of RACE against baselines. We observe that:

1) RACE achieves the highest average performance in TPR@1%FPR. Specifically, RACE outperforms the best baseline CoCo by 3.36% absolute with a low alarm rate, indicating the effectiveness of the creator-editor dual modeling framework.

2) RACE outperforms closely-related discourse-aware detection methods. Similar to RACE, CoCo and LF-Motifs utilize discourse information: CoCo relies primarily on entity-coherence graphs to model inner- and inter-sentence relations; while LF-Motifs introduces statistical features of RST trees concatenated with Longformer embeddings. Though they outperform other compared baselines, CoCo struggles to capture the local stylistic shift when semantic entities remain unchanged, and LF-Motifs's statistical features are relatively shallow. In contrast, RACE leverages RGCN for message passing directly over the relational graph, thereby capturing the intrinsic structural topology and logical anomalies that shallow motifs fail to represent.

3) Learning-based methods generally outperform metric-based ones for fine-grained classification. Metrics-based methods typically compress information into scalar values, which may be simple and effective for the binary setting, but the loss that such compression leads to also collapses the high-dimensional feature space necessary for the multi-class task. Aligned with the observation from Tufts et al. (2025), we see poor performance for certain detectors, with TPR@1%FPR as low as 0%. Even if we adopt several modifications to preserve more information, their performance still falls behind the learning-based methods, perhaps because the latter could entail the classification knowledge into well-trained parametric networks.

4.3 Ablation Study

As presented in Table 2, the TPR@1%FPR shows a clear drop when removing the involved components, confirming their individual benefits for improving fine-grained detection performance. We notice the largest performance drop arises at the LLM-generated class, followed by the LLM-Polished class. This is aligned with our intuition: The LLM-generated and polished samples are created by LLMs and humans, respectively, but share the same editor (*i.e.* the LLM). The degradation of *w/o Relation* variant indicates that, without logical relations, vanilla GCN fails to capture patterns defining the role of creator and confirms the core advantage of RACE. Without the contrastive learning that enhances the feature differences and the RGCN that explicitly models the dual roles of creator and

Table 3: Quantitative evaluation of feature discriminability using clustering validity indices.

Metric	CoCo	RACE
Davies-Bouldin Index (↓)	0.9286	0.8042
Calinski-Harabasz Index (↑)	2289.40	4333.32

editor, the detector might mix the characteristics of the two editor-similar types of samples.

Furthermore, the results of *w/o Basis* and *w/o Bottleneck* validate the necessity of parameter efficiency and feature compression. Specifically, removing the Basis Decomposition leads to a noticeable increase in performance variance (*i.e.*, high standard deviation). This means that the basis decomposition is an important regularizer because it forces weight sharing between similar relations. This stops over-parameterization and makes sure that optimization stays stable. Meanwhile, the removal of the Information Bottleneck Projection causes a specific performance degradation on the LLM-Polished class. This corroborates that this module effectively prevents the model from overfitting to superficial patterns shared with the LLM editor and forces it to focus on the invariant features indicative of the human creator.

4.4 Further Analysis

To provide a deeper insight into the effectiveness of our proposed method, we conduct a comprehensive comparison against CoCo or LF-Motifs, which are top-performing in the main experiments.

Discriminability of Feature Representations. To quantitatively evaluate the discriminability of the learned representations, we employ two standard clustering validity indices: the Davies-Bouldin Index (DBI) (Davies and Bouldin, 1979) and the Calinski-Harabasz Index (CH) (Caliński and Harabasz, 1974). As shown in Table 3, RACE achieves a lower DBI of 0.8042 than CoCo (0.9286), indicating a better ratio of intra-cluster scatter to inter-cluster separation. Regarding the CH index, RACE's score is nearly double that of CoCo, implying that explicitly modeling the creator/editor logic leads to a significantly more compact and distinct embedding space. The indices show the superiority of RACE in learning a discriminative feature space for fine-grained detection.

Impact of Text Length Variations. We investigate how input text length affects detection performance. Figure 4 illustrates the TPR@1% FPR across different token length intervals. While both methods perform well on texts longer than 600 tokens, RACE performs better for shorter texts with 200-600 tokens. This advantage suggests that our graph-based approach is more efficient in capturing the nuanced differences introduced by creators and editors, with relatively limited information.

Out-of-Distribution (OOD) Testing. To assess the robustness of our method across different text genres, we extend the evaluation to a Leave-One-Domain-Out setting on the four domains in HART, including *Arxiv*, *Essay*, *News*, and *Writing*. Excluding the preserved domain, the samples in the remaining three domains are split into the training and validation sets with a ratio of 9:1. From Table 4, we see that RACE outperforms CoCo and LF-Motifs in most cross-domain scenarios, particularly for structured genres like research articles and essays. For CoCo, the entity distributions are highly domain-dependent, and when transferred to an unseen domain, the learned entity patterns fail to reflect the new inductive bias, leading to per-

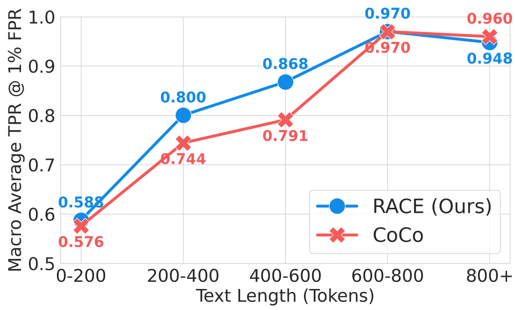

Figure 4: Analysis of detection performance of CoCo and our proposed RACE across varying text lengths.

Table 4: Performance comparison under the Out-of-Distribution setting. We employ a Leave-One-Domain-Out protocol where the model is trained on three domains and tested on the fourth unseen domain (Column 1). All values are reported in percentage (%). Best results are **bolded** and second-best are <u>underlined</u>.

Domain	Method	AUROC	Avg. TPR@1%FPR
Arxiv	CoCo	93.67	55.91
	LF-Motifs	<u>94.07</u>	<u>58.64</u>
	RACE	**96.61**	**76.28**
Essay	CoCo	89.84	28.72
	LF-Motifs	<u>91.12</u>	<u>47.85</u>
	RACE	**95.88**	**59.73**
News	CoCo	89.35	<u>39.36</u>
	LF-Motifs	<u>91.04</u>	35.36
	RACE	**92.69**	**44.30**
Writing	CoCo	83.03	**30.85**
	LF-Motifs	<u>84.73</u>	<u>31.59</u>
	RACE	**86.20**	30.84

formance degradation. Differently, RACE relies on both linguistic expression and logical organization and forms a more comprehensive view, thus enhancing the OOD generalizability.

5 Related Works

We brief recent advances in LLM-generated text detection by the classification settings.

5.1 Binary Classification

The binary classification is to judge whether a text piece is generated by the LLM. Under this setting, a detector assumes that **1)** all samples are either purely written by humans or generated by LLMs; or **2)** any text involving LLMs belongs to the "LLM" class. To model the differentiable signals, most existing methods focus on developing distribution-aware metrics like token probabilities (Gehrmann et al., 2019), token ranks (Su et al., 2023), or their combinations (Miralles-González et al., 2026), as these metrics reflect the disparities of human and LLM texts in word use. To instantiate this, researchers utilized various signals to manifest or amplify such disparities. For example, regeneration-based methods query an LLM with the given text to measure how similar the output is to the input, reflecting the familiarity the queried LLM is with the given text (Zhu et al., 2023; Mao et al., 2024; Wu et al., 2025b). The variants leverage multiple regenerations to calculate probability divergence (Yang et al., 2024b) or consider the impact of the prompts (Yu et al., 2024). Perturbation-based methods operate in the embedding space, assuming that LLM-generated text resides in negative curvature regions of log-likelihood Mitchell et al. (2023); Bao et al. (2024). Another research line directly learns stylistic representations using supervised learning (Solaiman et al., 2019; Guo et al., 2023; Soto et al., 2024; Guo et al., 2024). Among the supervised methods, Liu et al. (2023) and Kim et al. (2024) capture deeper linguistic structures and discourse coherence. To model the dual roles of creator and editor, we follow this line by introducing the RST tree, which deepened the understanding of discourse-level information.

5.2 Fine-grained Classification

Fine-grained classification is to differentiate the specific involvement of the human and the LLM to satisfy the regulation and forensics needs. From an identity perspective, some works attribute the given text to a specific LLM, thereby formulating a model attribution task (Li et al., 2023; Shi et al., 2024; Li and Wang, 2025).

Recently, human-LLM collaborative writing has become prevalent, and more works focus on differentiating the behavior and its extent in the resulting mixed text. For the scenario that LLM-generated paragraphs or sentences are interleaved with human writing, Zhang et al. (2024) constructs MixSet to address boundary detection. Zeng et al. (2024) identify the transition points between human and LLM texts by comparing the prototypes of neighboring text snippets. SeqXGPT (Wang et al., 2023) treats detection as a sequence labeling problem for precise localization within mixed texts. Other works set a third category to represent the mixed text. FAIDSet (Ta et al., 2025) categorizes text into three distinct classes: human-written, LLM-generated, and collaborative, with specific labels for LLM-polishing and LLM-continuation. APT-Eval (Saha and Feizi, 2025) considers the different levels of LLM polishing. Zhou et al. (2024) study the adversarial behavior named humanizing, typically to bypass LLM text detectors to earn unethical advantages. Recently, Bao et al. (2025) designed a detector that explicitly decouples text into content and expression dimensions, identifying LLM artifacts primarily in the expression layer.

However, they mainly focus on the linguistic expression, which represents the synthesized outcome after human-LLM collaboration, thus failing to reveal role-specific traits. In contrast, our proposed RACE incorporates both linguistic expression and the logical organization signals through rhetoric-guided graph learning, which models the generative process through the dual lenses of creator and editor, enabling superior performance on fine-grained detection.

6 Conclusion

We explored the four-class setting in fine-grained LLM-generated text detection, to distinguish human-written text, LLM-generated text, LLM-polished human text, and humanized LLM text. We modeled the dual roles of creator and editor through rhetorical structure construction and elementary discourse unit extraction, and designed the detector, RACE. By building the logic-aware graph and performing rhetoric-guided message passage, RACE outperformed 11 baselines on the HART benchmark with a low false alarm rate.

Limitations

In this paper, we conducted an initial exploration to perform the complex four-class task for fine-grained LLM-generated text detection. Despite the effectiveness of the proposed method RACE, we identify the following limitations:

1) We only conduct experiments on one public benchmark (*i.e.*, HART) because it is the only accessible dataset suitable for the four-class setting when we conducted this study. The performance of RACE on other languages, domains, and genres that HART does not cover remains unknown.

2) There is still room for RACE in terms of absolute performance improvement to satisfy the requirements for commercial use. Therefore, it is not recommended to directly take subsequent actions according to RACE's predictions without additional manual checks. Further research in this direction is advocated.

3) Though the four-class setting has been complex, there indeed exists the possibility that a text piece is the result of a longer editing sequence. A recent study began to consider sample editing multiple times by different LLMs (He et al., 2025), but we focus more on constructing the basic setting and thus did not explore this kind of effect.

Ethical Considerations

Risks. Our work aims at detecting LLM-generated text with a fine-grained setting that enables users to differentiate LLM-polished human text and humanized LLM text. Though our four-class setting is very suitable for satisfying regulatory needs, the method still requires further improvement in terms of precision under a low false alarm rate. In practice, it is not recommended to use an individual classifier for checking the text in course assignments and other writing scenarios. An additional manual verification is necessary after the detector sets an alarm.

Data. Our work uses the public benchmark HART, released by the existing work (Bao et al., 2025) under the MIT license. We follow HART's intended use of academic research on LLM-generated detection. We did not collect and use any unauthorized personal private data and did not recruit any human annotators.

Generative AI use. We adhere to the ACL policy (Cahill et al., 2025) and use generative AI tools for manuscript text polishing and code writing assistance only.

References

Anthropic. 2025. Claude 3.5 sonnet. Accessed: 2026-01-05.

Usman Anwar, Abulhair Saparov, Javier Rando, Daniel Paleka, Miles Turpin, Peter Hase, Ekdeep Singh Lubana, Erik Jenner, Stephen Casper, Oliver Sourbut, Benjamin L. Edelman, Zhaowei Zhang, Mario Günther, Anton Korinek, Jose Hernandez-Orallo, Lewis Hammond, Eric J Bigelow, Alexander Pan, Lauro Langosco, and 23 others. 2024. Foundational challenges in assuring alignment and safety of large language models. *Transactions on Machine Learning Research*.

Ekaterina Artemova, Jason S Lucas, Saranya Venkatraman, Jooyoung Lee, Sergei Tilga, Adaku Uchendu, and Vladislav Mikhailov. 2025. Beemo: Benchmark of expert-edited machine-generated outputs. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6992–7018. Association for Computational Linguistics.

Guangsheng Bao, Lihua Rong, Yanbin Zhao, Qiji Zhou, and Yue Zhang. 2025. Decoupling content and expression: Two-dimensional detection of ai-generated text. *Preprint*, arXiv:2503.00258.

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

Aoife Cahill, Leon Derczynski, and Kokil Jaidka. 2025. ACL Policy on Publication Ethics. Accessed: 2026-01-02.

T. Caliński and J Harabasz. 1974. A dendrite method for cluster analysis. *Communications in Statistics*, 3(1):1–27.

Elena Chistova. 2024. Bilingual rhetorical structure parsing with large parallel annotations. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 9689–9706. Association for Computational Linguistics.

David L. Davies and Donald W. Bouldin. 1979. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1(2):224–227.

FBI. 2025. Senior U.S. Officials Continue to be Impersonated in Malicious Messaging Campaign. Accessed: 2025-12-28.

Sebastian Gehrmann, Hendrik Strobelt, and Alexander Rush. 2019. GLTR: Statistical detection and visualization of generated text. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 111–116. Association for Computational Linguistics.

Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, and 1 others. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.

Michael Goodier. 2025. Revealed: Thousands of UK university students caught cheating using AI. Accessed: 2025-12-28.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

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. *Preprint*, arXiv:2301.07597.

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. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, pages 88320–88347. Curran Associates Inc.

Abhimanyu Hans, Avi Schwarzschild, Valeriia Cherepanova, Hamid Kazemi, Aniruddha Saha, Micah Goldblum, Jonas Geiping, and Tom Goldstein. 2024. Spotting llms with binoculars: Zero-shot detection of machine-generated text. In *Proceedings of the 41st International Conference on Machine Learning*, pages 17519–17537. PMLR.

Xinlei He, Xinyue Shen, Zeyuan Chen, Michael Backes, and Yang Zhang. 2024. MGTBench: Benchmarking Machine-Generated Text Detection. In *Proceedings of the 2024 ACM SIGSAC Conference on Computer and Communications Security*, pages 2251–2265. Association for Computing Machinery.

Yongxin He, Shan Zhang, Yixuan Cao, Lei Ma, and Ping Luo. 2025. Detree: Detecting human-ai collaborative texts via tree-structured hierarchical representation learning. *Preprint*, arXiv:2510.17489.

Beizhe Hu, Qiang Sheng, Juan Cao, Yang Li, and Danding Wang. 2025. LLM-Generated Fake News Induces Truth Decay in News Ecosystem: A Case Study on Neural News Recommendation. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 435–445. Association for Computing Machinery.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. In *Advances in Neural Information Processing Systems*, volume 33, pages 18661–18673. Curran Associates, Inc.

Zae Myung Kim, Kwang Lee, Preston Zhu, Vipul Raheja, and Dongyeop Kang. 2024. Threads of subtlety: Detecting machine-generated texts through discourse motifs. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5449–5474. Association for Computational Linguistics.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*.

Haoran Li and Quan Wang. 2025. Continual origin tracing of llm-generated text. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 479–489. Association for Computing Machinery.

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

Xiaoming Liu, Zhaohan Zhang, Yichen Wang, Hang Pu, Yu Lan, and Chao Shen. 2023. CoCo: Coherence-enhanced machine-generated text detection under low resource with contrastive learning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 16167–16188. Association for Computational Linguistics.

Xin Liu, Yang Li, and Kan Li. 2025. Enhancing the robustness of ai-generated text detectors: A survey. *Mathematics*, 13(13).

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

William C Mann and Sandra A Thompson. 1988. Rhetorical structure theory: Toward a functional theory of text organization. *Text-interdisciplinary Journal for the Study of Discourse*, 8(3):243–281.

Chengzhi Mao, Carl Vondrick, Hao Wang, and Junfeng Yang. 2024. RAIDAR: geneRative AI Detection viA Rewriting. In *The Twelfth International Conference on Learning Representations*.

Elyas Masrour, Bradley N. Emi, and Max Spero. 2025. DAMAGE: Detecting adversarially modified AI generated text. In *Proceedings of the 1stWorkshop on GenAI Content Detection (GenAIDetect)*, pages 120–133. International Conference on Computational Linguistics.

Pablo Miralles-González, Javier Huertas-Tato, Alejandro Martín, and David Camacho. 2026. Not all tokens are created equal: Perplexity attention weighted networks for ai-generated text detection. *Information Fusion*, 125:103465.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D Manning, and Chelsea Finn. 2023. Detectgpt: zero-shot machine-generated text detection using probability curvature. In *Proceedings of the 40th International Conference on Machine Learning*, pages 24950–24962. PMLR.

OpenAI. 2025a. Hello gpt-4o. Accessed: 2026-01-05.

OpenAI. 2025b. Introducing gpt-5. Accessed: 2026-01-05.

Alex Reinhart, Ben Markey, Michael Laudenbach, Kachatad Pantusen, Ronald Yurko, Gordon Weinberg, and David West Brown. 2025. Do llms write like humans? variation in grammatical and rhetorical styles. *Proceedings of the National Academy of Sciences*, 122(8):e2422455122.

Shoumik Saha and Soheil Feizi. 2025. Almost AI, almost human: The challenge of detecting AI-polished writing. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 25414–25431. Association for Computational Linguistics.

Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *European Semantic Web Conference*, pages 593–607.

Yuhui Shi, Qiang Sheng, Juan Cao, Hao Mi, Beizhe Hu, and Danding Wang. 2024. Ten words only still help: Improving black-box ai-generated text detection via proxy-guided efficient re-sampling. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, pages 494–502.

Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, Gretchen Krueger, Jong Wook Kim, Sarah Kreps, Miles McCain, Alex Newhouse, Jason Blazakis, Kris McGuffie, and Jasmine Wang. 2019. Release strategies and the social impacts of language models. *Preprint*, arXiv:1908.09203.

Rafael Alberto Rivera Soto, Kailin Koch, Aleem Khan, Barry Y. Chen, Marcus Bishop, and Nicholas Andrews. 2024. Few-shot detection of machine-generated text using style representations. In *The Twelfth International Conference on Learning Representations*.

Jinyan Su, Terry Zhuo, Di Wang, and Preslav Nakov. 2023. DetectLLM: Leveraging log rank information for zero-shot detection of machine-generated text. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 12395–12412. Association for Computational Linguistics.

Minh Ngoc Ta, Dong Cao Van, Duc-Anh Hoang, Minh Le-Anh, Truong Nguyen, My Anh Tran Nguyen, Yuxia Wang, Preslav Nakov, and Sang Dinh. 2025. FAID: Fine-grained AI-generated Text Detection using Multi-task Auxiliary and Multi-level Contrastive Learning. *Preprint*, arXiv:2505.14271.

Brian Tufts, Xuandong Zhao, and Lei Li. 2025. A practical examination of AI-generated text detectors for large language models. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 4824–4841. Association for Computational Linguistics.

Pengyu Wang, Linyang Li, Ke Ren, Botian Jiang, Dong Zhang, and Xipeng Qiu. 2023. SeqXGPT: Sentence-level AI-generated text detection. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1144–1156. Association for Computational Linguistics.

Yitong Wang, Zhongping Zhang, Margherita Piana, Zheng Zhou, Peter Gerstoft, and Bryan A. Plummer. 2025. Real, fake, or manipulated? detecting machine-influenced text. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 15022–15037. Association for Computational Linguistics.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Toru Sasaki, Thomas Arnold, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024. M4: Multi-generator, Multi-domain, and Multi-lingual Black-Box Machine-Generated Text Detection. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)"*, pages 1369–1407. Association for Computational Linguistics.

Alva West, Yixuan Weng, Minjun Zhu, Luodan Zhang, Zhen Lin, Guangsheng Bao, and Yue Zhang. 2025. Ai-generated text is non-stationary: Detection via temporal tomography. *Preprint*, arXiv:2508.01754.

Junchao Wu, Shu Yang, Runzhe Zhan, Yulin Yuan, Lidia Sam Chao, and Derek Fai Wong. 2025a. A survey on LLM-generated text detection: Necessity, methods, and future directions. *Computational Linguistics*, 51(1):275–338.

Junchao Wu, Runzhe Zhan, Derek F. Wong, Shu Yang, Xuebo Liu, Lidia S. Chao, and Min Zhang. 2025b. Who wrote this? the key to zero-shot LLM-generated text detection is GECScore. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 10275–10292. Association for Computational Linguistics.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025a. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, and 23 others. 2025b. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Lingyi Yang, Feng Jiang, Haizhou Li, and 1 others. 2024a. Is chatgpt involved in texts? measure the polish ratio to detect chatgpt-generated text. *APSIPA Transactions on Signal and Information Processing*, 13(2).

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

Xiao Yu, Yuang Qi, Kejiang Chen, Guoqiang Chen, Xi Yang, Pengyuan Zhu, Xiuwei Shang, Weiming Zhang, and Nenghai Yu. 2024. DPIC: Decoupling Prompt and Intrinsic Characteristics for LLM Generated Text Detection. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, pages 16194–16212. Curran Associates Inc.

Zijie Zeng, Lele Sha, Yuheng Li, Kaixun Yang, Dragan Gašević, and Guangliang Chen. 2024. Towards automatic boundary detection for human-ai collaborative hybrid essay in education. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 22502–22510.

Qihui Zhang, Chujie Gao, Dongping Chen, Yue Huang, Yixin Huang, Zhenyang Sun, Shilin Zhang, Weiye Li, Zhengyan Fu, Yao Wan, and Lichao Sun. 2024. LLM-as-a-coauthor: Can mixed human-written and machine-generated text be detected? In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 409–436. Association for Computational Linguistics.

Ying Zhou, Ben He, and Le Sun. 2024. Humanizing machine-generated content: Evading AI-text detection through adversarial attack. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation*, pages 8427–8437. ELRA and ICCL.

Biru Zhu, Lifan Yuan, Ganqu Cui, Yangyi Chen, Chong Fu, Bingxiang He, Yangdong Deng, Zhiyuan Liu, Maosong Sun, and Ming Gu. 2023. Beat LLMs at their own game: Zero-shot LLM-generated text detection via querying ChatGPT. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7470–7483. Association for Computational Linguistics.

A More Details of Experiment Setups

A.1 Dataset Details

We reconstructed the dataset by processing the raw JSON files from the HART benchmark, merging the original development and test splits into a unified corpus. To support our classification framework, we implemented a parsing pipeline that assigns fine-grained labels based on the unique record identifiers (`id`) and metadata fields (`content_source`, `language_source`) of each entry. The mapping logic is defined as follows:

- Human-Written Text: Identified by base record IDs lacking derivative prefixes (*e.g.*, `gen/`, `rep/`), representing the original, unaltered human authorship.
- LLM-Generated Text: Primarily derived from records prefixed with `gen/`, where the `content_source` indicates a machine origin (*e.g.*, `machine:gpt-4`). We also map LLM-to-LLM revision chains (prefixed `hum/gen/` where the reviser is another model) to this category, treating them as fully synthetic content.
- LLM-Polished Human Text: Extracted from records prefixed with `rep/`, where the `language_source` (tagged as `rephrase:`) indicates that an original human text was refined by an LLM.
- Humanized LLM Text: Identified from records prefixed with `hum/gen/`, specifically filtering for instances where the `language_source` is tagged as `humanize:human` or `humanize:tool`. This captures the distinct scenario of synthetic text subsequently edited by human annotators or grammar correction tools.

Table 5 presents the detailed statistics of the dataset. The LLMs used in the data include Claude-3.5-Sonnet (Anthropic, 2025), GPT-3.5-Turbo, GPT-4o (OpenAI, 2025a), Gemini-

Table 5: Statistics of the resplit HART dataset.

Domain	Category	Train	Val	Test	Total
Arxiv	Human-Written	700	100	200	1,000
	LLM-Polished	700	100	200	1,000
	LLM-Generated	1,229	174	352	1,755
	Humanized	172	25	48	245
Essay	Human-Written	700	100	200	1,000
	LLM-Polished	700	100	200	1,000
	LLM-Generated	1,220	175	349	1,744
	Humanized	179	25	52	256
News	Human-Written	700	100	200	1,000
	LLM-Polished	700	100	200	1,000
	LLM-Generated	1,229	175	354	1,758
	Humanized	169	25	48	242
Writing	Human-Written	700	100	200	1,000
	LLM-Polished	700	99	201	1,000
	LLM-Generated	1,211	175	342	1,728
	Humanized	191	27	54	272
Total		**11,200**	**1,600**	**3,200**	**16,000**

1.5-Pro (Georgiev et al., 2024), Llama-3.3-70b-Instruct (Grattafiori et al., 2024), and Qwen-2.5-72b-Instruct (Yang et al., 2025b).

A.2 Metrics Calculation

We adopt macro AUROC and TPR at 1% FPR for the main experiments. Let $\mathcal{C} = \{1, \ldots, C\}$ be the set of classes (here, $C = 4$). For each class $c \in \mathcal{C}$, let $y_{i,c} \in \{0, 1\}$ denote the binary label and $\hat{p}_{i,c} \in [0, 1]$ the predicted probability for the i-th sample. We treat the multi-class problem as C independent binary classification tasks (One-vs-Rest). The Macro-AUROC is defined as:

$$\text{Macro-AUROC} = \frac{1}{C}\sum_{c=1}^{C} \text{AUROC}(y_{\cdot,c}, \hat{p}_{\cdot,c}), \tag{7}$$

where $\text{AUROC}(\cdot, \cdot)$ denotes the standard area under the receiver operating characteristic curve for each binary target. The macro-averaged TPR at the 1% FPR is defined as:

$$\text{TPR@1\%FPR} = \frac{1}{C}\sum_{c=1}^{C} \text{TPR}_c(\tau_c), \tag{8}$$

subject to:

$$\tau_c = \min\{\tau \in [0, 1] \mid \text{FPR}_c(\tau) \leq 0.01\}. \tag{9}$$

Here, $\text{TPR}_c(\tau)$ and $\text{FPR}_c(\tau)$ represent the true positive and false positive rates for class c at threshold τ, respectively. We use macro averaging to highlight the influence of the minority class in the dataset.

A.3 More Implementation Details

A.3.1 RACE

Dual Trace Extraction We employ the IsaNLP RST Parser[1] proposed by Chistova (2024), which maps relations to a unified set of 18 coarse-grained classes. The released checkpoint we used can be found in their HuggingFace repository[2] The relation types considered in our study include Attribution, Background, Cause, Comparison, Condition, Contrast, Elaboration, Enablement, Evaluation, Explanation, Joint, Manner-Means, Same-unit, Summary, Temporal, Textual-organization, Topic-Change, and Topic-Comment.

[1] https://github.com/tchewik/isanlp_rst

[2] https://huggingface.co/tchewik/isanlp_rst_v3/tree/rstdt

Backbone Model The pretrained RoBERTa-base model can be downloaded from Facebook AI's HuggingFace page[3].

Optimization Let $\mathcal{B} = \{(x_i, y_i)\}_{i=1}^{N}$ denote a mini-batch of N input samples, where x_i represents the input text and $y_i \in \{1, \ldots, C\}$ is the corresponding ground-truth label, with $C = 4$ representing the number of classes. The Supervised Contrastive Loss $\mathcal{L}_{\text{con}}$ is formulated as:

$$\mathcal{L}_{\text{con}} = \sum_{i \in I} \frac{-1}{|P(i)|} \sum_{p \in P(i)} \log \frac{e^{\mathbf{z}_{\mathcal{G}}^{i} \cdot \mathbf{z}_{\mathcal{G}}^{p} / \tau}}{\sum_{a \in A(i)} e^{\mathbf{z}_{\mathcal{G}}^{i} \cdot \mathbf{z}_{\mathcal{G}}^{a} / \tau}}. \tag{10}$$

Here, $\mathbf{z}_{\mathcal{G}}^{i}$ is the feature vector extracted by Eq. (5) for the i-th sample, $I \equiv \{1, \ldots, N\}$ is the set of indices in the batch. $A(i) \equiv I \setminus \{i\}$ represents the set of all indices excluding the anchor i. The set $P(i) \equiv \{p \in A(i) : y_p = y_i\}$ denotes the set of indices for positive samples sharing the same class label as i, and $|P(i)|$ is its cardinality. The symbol $\tau \in \mathbb{R}^{+}$ is a temperature parameter that controls the smoothness of the distribution. For the Cross-Entropy Loss $\mathcal{L}_{\text{ce}}$, we apply it on the classifier output $\hat{y}_i$ calculated by Eq. (6), formulated as:

$$\mathcal{L}_{\text{ce}} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} \mathbb{1}_{[y_i = c]} \log(\hat{y}_{i,c}), \tag{11}$$

where $\mathbb{1}_{[\cdot]}$ is the indicator function, y_i is the ground-truth label, and $\hat{y}_{i,c} \in [0, 1]$ is the predicted probability for class c.

A.3.2 Baselines

Given the absence of existing baselines specifically designed for this four-class classification framework, we selected representative methods from both learning-based and metric-based paradigms and adapted them to our reorganized HART dataset.

Learning-based Methods Adaptation. We modified the output dimensions of the classification heads to support four categories while retaining the original model architectures. Specifically:

- **CoCo and LF-Motifs:** We adjusted the final classification layer to output four class probabilities and optimize the learning rates on the training set to ensure convergence while keeping other hyperparameters consistent with the original implementations. Specifically, for LF-Motifs,

[3] https://huggingface.co/FacebookAI/roberta-base

we re-extracted the single, double, and triple triads from the HART corpus to reconstruct the features required for the four-class scenario.

- **SeqXGPT:** We reformulated the training objective by removing the Conditional Random Field (CRF) layer and discarding the fine-grained sequence labeling prefixes (*i.e.*, B-, M-, E-, S-). Instead, the model was trained to predict one of the four category labels for each token directly. During inference, we maintained the original majority voting mechanism, aggregating token-level predictions to determine the sentence-level label.
- **DeTeCtive:** We extended the contrastive learning objective by expanding the sample definitions in the contrastive loss from a binary distinction (LLM v.s. Human) to the target four distinct categories. All other training configurations followed the original paper.

Metric-based Methods Adaptation. Since metric-based detectors are originally designed for binary classification via thresholding, we adapted them by leveraging their intermediate signals:

- **Multi-interval Thresholding (F-DetectGPT):** We adapted Fast-DetectGPT by discretizing its probabilistic curvature score into four intervals using three empirical thresholds (0.5, 0.8, and 1.2). These intervals correspond to Human, LLM-Polished, Humanized, and LLM-Generated, respectively.
- **Feature Fusion with MLP (Binoculars$_{\text{MLP}}$ and F-DetectGPT$_{\text{MLP}}$):** We introduced an MLP classifier for each method independently. For Binoculars, we concatenated $\log$ PPL and $\log$ X-PPL to form the input feature vector; for Fast-DetectGPT, we used the curvature score as a single-dimensional feature. Both were then fed into their respective MLPs for four-class prediction.
- **Decoupled Content-Expression Judgment (Binoculars$_{\text{C-T}}$ and F-DetectGPT$_{\text{C-T}}$):** Following HART (Bao et al., 2025), we employed a decoupled detection strategy. We performed binary classification independently on the content and expression dimensions. The final label is derived from the combination of these two binary outcomes: Human (Content: Human, Expression: Human), LLM-Generated (Content: LLM, Expression: LLM), LLM-Polished (Content: Human, Expression: LLM), and Humanized (Content: LLM, Expression: Human). For the expression dimension, we used the text to be tested following the best-performing setting (C_2-T) reported in HART; for the content dimension, we utilized the `content` provided in the original HART dataset.

Table 6: Performance comparison of different detection methods evaluated using Avg. TPR@5%FPR.

Method	Avg. TPR@5%FPR
RoBERTa	92.53
CoCo	94.13
SeqXGPT	54.41
DeTeCtive	92.82
LF-Motifs	91.90
Binoculars$_{\text{MLP}}$	28.00
Binoculars$_{\text{C-T}}$	0.34
F-DetectGPT	13.99
F-DetectGPT$_{\text{MLP}}$	23.27
F-DetectGPT$_{\text{C-T}}$	0.34
TDT$_{\text{SVC}}$	16.37
RACE (Ours)	**94.41**

Table 7: Comparison of efficiency and model size. Training time (hours, **h**), inference throughput (**samples/s**), and model size (million parameters, **M**) are reported. Data preprocessing is excluded from both training time and inference throughput.

Method	Training	Inference	Params
RoBERTa	2.119	220.8	125.2
CoCo	2.138	32.4	125.6
LF-Motifs	0.587	50.6	148.8
RACE (Ours)	1.071	90.0	128.6

B Additional Experimental Results

B.1 Supplementary Quantitative Comparison

Table 6 reports the performance under a more relaxed constraint of 5% FPR. Our method consistently maintains the leading position with an average TPR of 94.41%. This consistent superiority across different thresholds highlights the strong discriminative power of our model, which ensures a high safety margin and demonstrates its reliability for high-precision applications.

B.2 Efficiency Analysis

Table 7 presents the training time, inference throughput, and the number of parameters for the four top-performing methods. While LF-Motifs appears to achieve the lowest training time, primarily due to its utilization of the optimized Hugging

Face Trainer[4], it requires a heavy data preprocessing phase to extract single, double, and triple triads. This extraction process takes over three hours, significantly longer than other compared methods. Our proposed RACE consumes relatively low training and inference time with a comparable scale of model parameters, confirming its efficiency for model preparation and deployment in reality.

C Reproducibility

The code is available at the following anonymous GitHub repository for reproducibility needs: https://anonymous.4open.science/r/Submission-169-RACE.

D Future Work

While existing methods have struggled to address the nuanced regulatory requirements of specific domains like academic writing, our work RACE demonstrates that decoupling the roles of creator and editor is a promising direction for next-generation LLM-generated text detection. We foresee three pathways for future exploration motivated by this dual-role paradigm:

- **Logic-Based Model Attribution:** Current attribution methods often rely on surface-level token probability distributions, which are fragile to simple editing. Future research could adapt RACE's graph-based topological features to fingerprint the unique logical thought processes of specific LLMs, potentially enabling attribution even after heavy human polishing.
- **Fine-Grained EDU-Level Detection:** Moving beyond document-level labels, the creator-editor framework naturally extends to localizing specific human or LLM contributions within a single text. Future works could explore identifying exact EDUs where a human editor intervenes in LLM-generated drafts, providing granular evidence for academic integrity investigations.
- **Adversarial Logic Defense:** As LLMs become capable of mimicking human rhetorical structures, the "arms race" will shift from lexical to logical obfuscation. Future logic-aware adversarial attacks where prompts explicitly request structural restructuring and corresponding defense mechanisms can be explored.

[4] https://huggingface.co/docs/transformers/en/trainer