PEC-DETECTOR: PLANNING-EXECUTION COUPLING TRAJECTORIES FOR DOCUMENT-LEVEL LLM-GENERATED TEXT DETECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Reliable AI-generated text detection is increasingly difficult as large language models (LLMs) produce fluent and coherent long texts. Existing detectors typically derive document-level evidence from static textual representations or aggregated statistics, without explicitly modeling dynamic coupling between discourse planning and current writing execution as a document unfolds. We show that remote discourse history provides measurable conditional information about the current sentence beyond local context, and that its contribution follows different trajectories in machine-generated text (MGT) and human-written text (HWT). Motivated by this observation, we propose **PEC-Detector**, a planning-execution coupling framework for document-level machine-generated text detection. At each sentence position, we induce planning and local context states from remote discourse history and local context, respectively, and construct a conditional planning-execution relation with the current writing execution. The resulting position-wise representations form an ordered coupling trajectory for document-level source prediction. A joint conditional semantic prediction objective encourages the planning state to retain information relevant to current writing execution. Experiments on public benchmarks spanning diverse domains, generators, and languages show that our method consistently achieves state-of-the-art detection performance. Furthermore, PEC-Detector remains robust under distribution shifts and adversarial attacks.

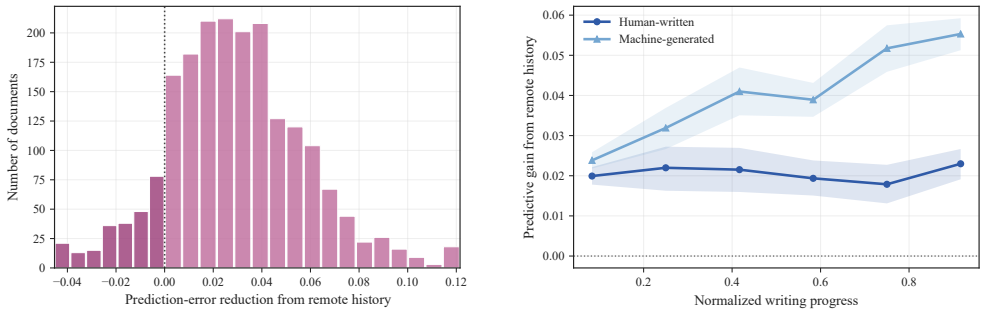
1 INTRODUCTION

Large language models (LLMs) have become capable of generating texts that are grammatically fluent, semantically coherent, and well organized at the document level Guo et al. (2026); Wang et al. (2024a); Liu et al. (2023). They are increasingly used in journalism, academic writing, and content creation. However, these advances have also blurred the boundary between machine-generated text (MGT) and human-written text (HWT), raising risks of misinformation, academic misconduct, and ambiguous content provenance Zhu et al. (2026); Borile & Abrate (2025); Wu et al. (2024). Consequently, reliable machine-generated text detection has become an important component of trustworthy digital content Wu et al. (2025a).

To address this challenge, extensive approaches have been proposed. Training-free methods employ proxy LLMs to compute statistics such as likelihood, perplexity, and probability curvature Tufts et al. (2025); Yang et al. (2024); Mitchell et al. (2023). For example, DetectGPT identifies machine-generated text from local curvature in the model's probability space. Another line of research focuses on supervised classification, which learns the decision boundaries between MGT and HWT directly from contextual semantic representations of complete texts or features derived from multiple language models Li et al. (2025); Mao et al. (2024). More recent work has explored higher-level cues related to syntax and semantics. ProSSD characterizes systematic shifts of MGT in a joint semantic-structural space, while other methods leverage hierarchical discourse structures or patterns of event transitions Xiong et al. (2026); Kim et al. (2024); Tian et al. (2024).

Despite recent advances in document-level machine-generated text detection, existing methods predominantly characterize static textual representations or aggregated statistics, while leaving the dy-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.



(a) Predictive contribution of remote history (b) Planning-execution coupling trajectory

Figure 1: Conditional contribution of remote discourse history. (a) Incorporating remote history beyond local context reduces the semantic prediction error of the current sentence for most documents. (b) The conditional contribution exhibits distinct position-wise trajectories in MGT and HWT.

namic coupling between discourse planning and current writing execution implicit as a document unfolds. Text composition is not merely the simple accumulation of locally coherent sentences: each sentence not only continues the immediate local semantics, but remains consistent with the topics, stances, and discourse goals gradually established by the preceding text Ji & Huang (2021); Yao et al. (2019). In other words, text generation can be viewed as a dynamic planning-execution process. Remote discourse history carries latent constraints on subsequent content, local context determines how the current sentence connects to it, and the current sentence itself constitutes a concrete execution under these constraints. If MGT and HWT exhibit different patterns in the formation of such discourse constraints, discriminative evidence may reside not only in what is ultimately written but also in how the remote planning state couples with current writing execution.

Based on this perspective, we perform label-free conditional semantic prediction to assess whether remote discourse history can further enhance prediction of the current sentence beyond local context. Figure 1a shows that incorporating remote history reduces the prediction error in most documents, indicating that it carries incremental information relevant to current writing. Furthermore, we trace this conditional contribution across sentence positions in HWT and MGT; computational details are provided in Appendix A. Figure 1b shows that both types of text exhibit comparable contributions at early positions but diverge as the document unfolds: the contribution of remote discourse history generally increases in MGT, whereas it remains relatively stable in HWT. This phenomenon suggests distinct planning-execution coupling patterns between the two sources.

To capture this phenomenon, we propose **PEC-Detector**, a planning-execution coupling framework for document-level machine-generated text detection (see Figure 2). At each sentence position, it induces a planning state from remote discourse history and encodes a local continuation state from recent context, treating the semantic representation of the current sentence as corresponding writing execution. We then introduce a planning-execution relation module to perform a consistency-deviation decomposition of discourse planning and current writing execution under the constraint of the local continuation state. A semantic gate further adaptively fuses the resulting relation representation with current sentence semantics, preserving both content and clues about the generation process. Finally, the position-wise coupling representations form an ordered planning-execution trajectory, whose dependencies support document-level source prediction. During training, we jointly optimize detection and conditional semantic prediction to encourage the planning state to retain information relevant to current writing execution while learning distinct coupling patterns across sources. Extensive experiments demonstrate that our method effectively distinguishes MGT from HWT and achieves superior detection performance across multiple public benchmarks.

Our main contributions are summarized as follows: (1) We revisit document-level machine-generated text detection from a dynamic writing perspective. Given local context, the conditional contribution of remote discourse history to the current writing follows distinct trajectories as documents unfold: it generally increases in MGT, but remains relatively stable in HWT. (2) We propose **PEC-Detector**, a framework that formulates source prediction as identifying ordered planning-execution coupling trajectories to distinguish MGT from HWT. (3) Extensive experiments on public benchmarks demonstrate that our method delivers superior performance across a variety of complex scenarios, including multi-domain, multi-LLM, distribution shifts, and adversarial attacks.

2 Related Work

Existing approaches can be divided into active and passive paradigms Chen et al. (2025b); Shi et al. (2024). Active methods embed verifiable watermarks during generation for subsequent verification, but cannot apply to text generated without such signals Huang & Wan (2025); Zhao et al. (2024). Passive methods instead infer source evidence from completed text and mainly comprise statistical detection and supervised classification Zeng et al. (2024).

Statistical detection methods distinguish MGT from HWT using distributional discrepancies in their generation behaviors Ma & Wang (2024); Wu et al. (2025b). They typically construct a detection score from statistics estimated by one or more proxy language models Su et al. (2023). Early work uses token-level statistics including token rank, entropy, and log-likelihood Solaiman et al. (2019); Gehrmann et al. (2019); later methods probe their variation under perturbations or conditional contexts Ni et al. (2024). DetectGPT identifies machine text through the curvature of the log-probability landscape Mitchell et al. (2023), and Fast-DetectGPT improves this criterion using conditional probability curvature and efficient sampling Bao et al. (2024). DNA-GPT compares an observed continuation with a continuation regenerated from its prefix Yang et al. (2024), whereas Binoculars contrasts the behaviors of two related language models Hans et al. (2024). These methods avoid labeled training data, but their performance may vary substantially across domains and generation conditions Wu et al. (2024); Tufts et al. (2025).

Supervised methods learn discriminative representations directly from labeled texts Hu et al. (2023); Guo et al. (2026). RoBERTa-base fine-tunes a contextual encoder, whereas DetectAnyLLM learns task-oriented discrepancy features for detection (Liu et al., 2019; Fu et al., 2025). Other supervised approaches introduce specific structural or representation-based evidence, including discourse motifs in LF+Motifs, preceding-token memorization in BISCOPE, syntactic structures in PRDetect, hidden representation patterns in RepreGuard, and projected semantic-structural Distributions in ProSSD (Kim et al., 2024; Guo et al., 2024; Li et al., 2025; Chen et al., 2025a; Xiong et al., 2026). Raidar instead treats the discrepancy induced by LLM rewriting as a detection signal (Mao et al., 2024). Although supervised detectors exploit richer cues, their decision boundaries remain contingent on labeled data coverage and can weaken generalization Xu et al. (2024). More broadly, existing detectors largely derive document-level evidence from static textual cues, leaving the dynamic coupling process through which remote discourse history constrains the current sentence implicit beyond local context.

3 Methodology

3.1 Problem Definition

Given a document $\mathcal{X} = (s_1, s_2, \ldots, s_N)$ consisting of N sentences, document-level machine-generated text detection aims to determine whether $\mathcal{X}$ is written by a human or generated by a machine. We denote its source label by $y \in \{0, 1\}$, where $y = 1$ denotes MGT and $y = 0$ indicates HWT. The detector takes only the final text as input. Its objective is to learn a parameterized detection function f_θ that estimates the source distribution $p_\theta(y \mid \mathcal{X}) = f_\theta(\mathcal{X})$.

Rather than treating $\mathcal{X}$ as a static text, we focus on the dynamic coupling between remote discourse history and current sentence writing as a document unfolds. For each target position t with sufficient preceding context and a local context window of size K, we partition the preceding text into remote discourse history $\mathcal{H}_t = (s_1, \ldots, s_{t-K-1})$ and recent local context $\mathcal{C}_t = (s_{t-K}, \ldots, s_{t-1})$, and regard the current sentence s_t as the observed writing execution at position t. Therefore, each valid position defines a triplet $\big(\mathcal{H}_t, \mathcal{C}_t, s_t\big)$ for $t \in \{K + 2, \ldots, N\}$. We study the planning-execution relation between the discourse state reflected by remote history and the current writing execution, and examine whether its coupling patterns provide evidence for source prediction.

3.2 Position-wise Planning-Execution Coupling

At each sentence position t, **PEC-Detector** induces a discourse planning state from remote discourse history $\mathcal{H}_t$, encodes a local context state from $\mathcal{C}_t$, and treats the semantic representation of s_t as the current writing execution. Then it models the planning-execution coupling at position t.

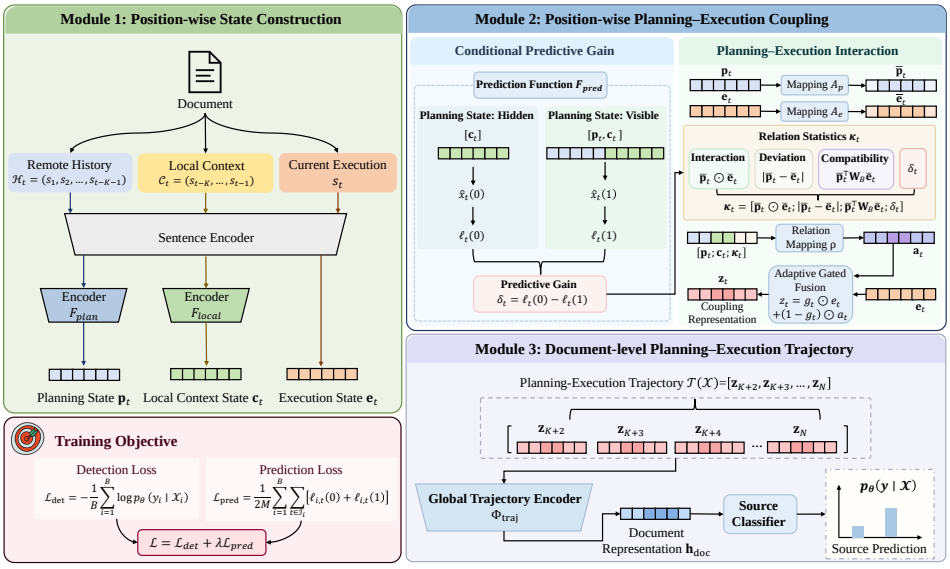


Figure 2: Overview of **PEC-Detector**. The framework constructs planning-execution coupling representations at each sentence position using remote discourse history, local context, and the current execution, then encodes the resulting coupling trajectory for document-level source prediction.

Planning, Local Context, and Execution States. We first use a pre-training sentence encoder g_ϕ to map each sentence s_i to a normalized semantic representation $\mathbf{x}_i = \frac{g_\phi(s_i)}{\|g_\phi(s_i)\|_2}$, where g_ϕ remains frozen throughout training Song et al. (2020). We then project the sentence representations into a shared representation space and incorporate their positions within the document:

$$\mathbf{u}_i = \mathrm{LN}\left(\mathbf{W}_s \mathbf{x}_i + \mathbf{b}_s\right) + \mathbf{r}_i, \tag{1}$$

where $\mathbf{W}_s \in \mathbb{R}^{d \times d_s}$, $\mathbf{b}_s \in \mathbb{R}^d$, and $\mathbf{r}_i \in \mathbb{R}^d$ denote a learnable projection matrix, bias term, and positional embedding associated with sentence position i, respectively.

For each target position t, the discourse planning state is induced solely from remote discourse history $\mathcal{H}_t$:

$$\mathbf{p}_t = \left[F_{\mathrm{plan}}\left(\mathbf{U}_t^{\mathcal{H}}\right)\right]_{t-K-1}, \tag{2}$$

Here, $\mathbf{p}_t \in \mathbb{R}^d$, and F_{plan} employs causal attention to aggregate discourse information up to position $t-K-1$ without accessing any future content. We interpret $\mathbf{p}_t$ as a latent discourse representation induced from observed remote discourse history, rather than a direct reconstruction of the author's original writing plan. Correspondingly, the local continuation state is obtained by applying a local mapping F_{local} to the ordered representations of the most recent K sentences:

$$\mathbf{c}_t = F_{\mathrm{local}}\left(\mathbf{U}_t^{\mathcal{C}}\right), \tag{3}$$

Here, $\mathbf{c}_t \in \mathbb{R}^d$, and F_{local} characterizes the local continuation condition under which the current sentence is written. $\mathbf{e}_t = \mathbf{u}_t$ serves as the current writing execution at position t. Thus, each valid position is represented by the state triplet $(\mathbf{p}_t, \mathbf{c}_t, \mathbf{e}_t)$.

Planning-Execution Relation Modeling. The relation between the discourse planning state and current writing execution cannot be fully captured by their overall semantic similarity alone. On the one hand, we need to isolate the conditional predictive information provided by the remote discourse state beyond the local context. On the other hand, we seek to retain the decomposition of correspondence and deviation between planning and execution in representation space, together with their cross-dimensional compatibility. Therefore, we characterize the position-wise planning-execution relation through the complementary views of conditional prediction and representation interaction. Specifically, we introduce a shared prediction function $F_{\mathrm{pred}} : \mathbb{R}^{2d} \to \mathbb{R}^{d_s}$ and use $m \in \{0, 1\}$ to control whether the discourse planning state is visible:

$$\begin{gathered}\widehat{\mathbf{x}}_t(m) = \mathrm{norm}\left(F_{\mathrm{pred}}([m\mathbf{p}_t; \mathbf{c}_t])\right), \qquad \ell_t(m) = 1 - \cos\left(\widehat{\mathbf{x}}_t(m), \mathbf{x}_t\right), \\ \delta_t = \ell_t(0) - \ell_t(1).\end{gathered} \tag{4}$$

Because the two conditions share the same prediction function, δ_t isolates the semantic predictive gain attributable to planning-state visibility.

To further characterize their representational relationship, we denote the resulting representations by $\bar{\mathbf{p}}_t = A_p(\mathbf{p}_t)$ and $\bar{\mathbf{e}}_t = A_e(\mathbf{e}_t)$, respectively. $A_p, A_e : \mathbb{R}^d \to \mathbb{R}^d$ denote learnable mappings for the discourse planning state and current writing execution. Finally, we define complementary perspectives at position t:

$$\boldsymbol{\kappa}_t = \left[\bar{\mathbf{p}}_t \odot \bar{\mathbf{e}}_t; |\bar{\mathbf{p}}_t - \bar{\mathbf{e}}_t|\,; \bar{\mathbf{p}}_t^\top \mathbf{W}_B \bar{\mathbf{e}}_t; \delta_t\right], \tag{5}$$

The element-wise product captures joint responses between planning and execution, the absolute difference measures coordinate-wise deviation, and the bilinear term models compatibility across semantic dimensions. Because these relation features alone do not retain the state information needed to contextualize the coupling, we concatenate $\boldsymbol{\kappa}_t$ with the planning and local context states: $\mathbf{a}_t = \rho([\mathbf{p}_t; \mathbf{c}_t; \boldsymbol{\kappa}_t])$, where ρ is a learnable relation mapping. Consequently, $\mathbf{a}_t$ represents the relational structure between discourse planning and current writing execution conditioned on the local continuation state.

Finally, we adaptively fuse $\mathbf{a}_t$ with the semantic representation of the current writing execution. The former describes how the preceding discourse state constrains the current writing, whereas the latter represents the concrete content realized at position t. We define the adaptive fusion as follows:

$$\mathbf{g}_t = \sigma\left(\mathbf{W}_G[\mathbf{e}_t; \mathbf{a}_t] + \mathbf{b}_G\right), \qquad \mathbf{z}_t = \mathbf{g}_t \odot \mathbf{e}_t + (1 - \mathbf{g}_t) \odot \mathbf{a}_t. \tag{6}$$

Here, $\mathbf{g}_t \in (0,1)^d$ determines the relative contributions of $\mathbf{e}_t$ and $\mathbf{a}_t$, and $\mathbf{z}_t$ denotes the planning-execution coupling representation at position t.

3.3 Document-level Planning-Execution Trajectory

Even when HWT and MGT documents exhibit similar position-wise coupling representations, their persistence and cross-position dependencies may follow different patterns as the document unfolds. Therefore, we treat the sequence of coupling representations as a document-level planning-execution trajectory and use it to learn evidence for source classification.

Ordered Trajectory Modeling. For a document $\mathcal{X}$, we define the planning-execution trajectory as $\mathcal{T}(\mathcal{X}) = [\mathbf{z}_{K+2}; \mathbf{z}_{K+3}; \ldots; \mathbf{z}_N] \in \mathbb{R}^{L \times d}$, where $L = N - K - 1$. This construction shifts the basic unit of document-level modeling from sentence-content representations to position-indexed planning-execution coupling representations. We further employ a Transformer encoder Φ_{traj} with global self-attention Devlin et al. (2019); Vaswani et al. (2017) and prepend a learnable document token $\mathbf{q}_{\text{doc}} \in \mathbb{R}^d$ to represent the entire document:

$$\mathbf{h}_{\text{doc}} = \left[\Phi_{\text{traj}}\left([\mathbf{q}_{\text{doc}}; \mathcal{T}(\mathcal{X})]\right)\right]_0, \tag{7}$$

where $\mathbf{h}_{\text{doc}} \in \mathbb{R}^d$, and $[\cdot]_0$ denotes the output representation corresponding to the document token.

Training Objective. We train **PEC-Detector** by jointly optimizing a document-level source-classification objective and a position-level conditional semantic-prediction objective. For the i-th document, the output representation $\mathbf{h}_{\text{doc}}^{(i)}$ is mapped by a classifier Cls_θ to logits over the two source classes, yielding the conditional source distribution $p_\theta(\cdot \mid \mathcal{X}_i) = \text{softmax}\left(\text{Cls}_\theta(\mathbf{h}_{\text{doc}}^{(i)})\right)$.

For a mini-batch of B documents, let $\mathcal{I}_i = \{K+2, \ldots, N_i\}$ denote the set of valid positions in the i-th document, and let $M = \sum_{i=1}^{B} |\mathcal{I}_i|$ denote the total number of valid positions in the mini-batch. We define the joint training objective as:

$$\mathcal{L}_{\text{pred}} = \frac{1}{2M} \sum_{i=1}^{B} \sum_{t \in \mathcal{I}_i} \left[\ell_{i,t}(0) + \ell_{i,t}(1)\right], \qquad \mathcal{L}_{\text{det}} = -\frac{1}{B} \sum_{i=1}^{B} \log p_\theta\left(y_i \mid \mathcal{X}_i\right), \tag{8}$$
$$\mathcal{L} = \mathcal{L}_{\text{det}} + \lambda \mathcal{L}_{\text{pred}}.$$

Here, $\mathcal{L}_{\text{det}}$ supervises document-level source classification, while $\mathcal{L}_{\text{pred}}$ constrains semantic prediction of the current sentence under both planning-state visibility conditions. The hyperparameter λ controls the relative weight of the auxiliary prediction objective.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Benchmarks and Evaluation. We evaluate **PEC-Detector** on two public benchmarks, DetectRL (Wu et al., 2024) and MAGE (Li et al., 2024). DetectRL covers four high-risk application domains and texts generated by mainstream LLMs; we additionally assess robustness against adversarial attacks under its *Multi-Attack* setting. To evaluate generalization under distribution shifts, we use the *Fixed-domain & Arbitrary-models* and *Arbitrary-domains & Model-specific* settings of MAGE. We report AUROC and F1: AUROC measures threshold-independent discriminability, whereas F1 measures the balance between precision and recall. Detailed dataset statistics and split protocols are provided in Appendix B.1.

Baselines. We categorize the baselines into zero-shot and supervised methods. The zero-shot baselines include DetectGPT Mitchell et al. (2023), Fast-DetectGPT Bao et al. (2024), DNA-GPT Yang et al. (2024), and Binoculars Hans et al. (2024). The supervised baselines comprise RoBERTa-base Liu et al. (2019), LF+Motifs Kim et al. (2024), Raidar Mao et al. (2024), BiScope Guo et al. (2024), PRDetect Li et al. (2025), RepreGuard Chen et al. (2025a), DetectAnyLLM Fu et al. (2025), and ProSSD Xiong et al. (2026). Further details of the baseline methods are provided in Appendix B.2.

Settings. We use the pretrained all-mpnet-base-v2 model as a frozen sentence encoder and retain up to 256 sentences per document. We set the local context window to $K = 2$ and the representation dimension to $d = 256$. The remote-history and trajectory encoders each comprise three Transformer layers, whereas local context is encoded by a single-layer GRU; all trainable modules use a dropout rate of 0.15. We train for 30 epochs with AdamW Loshchilov & Hutter (2019), using a learning rate of 2×10^{-4}, weight decay of 10^{-4}, and batch size of 128. The auxiliary prediction loss weight is $\lambda = 0.1$. Additional implementation and runtime details are provided in Appendix B.3.

4.2 MAIN RESULTS

Overall Performance. As shown in Table 1, PEC-Detector achieves the best AUROC in seven of the eight evaluation settings and attains the highest overall average AUROC of 98.88%. Under the *Multi-Domain* setting, PEC-Detector obtains an average AUROC of 99.21%, outperforming the second-best PRDetect by 0.71 percentage points; it falls slightly below RoBERTa-base only on arXiv. Under the *Multi-LLM* setting, our method achieves the best result for all four generators, with an average AUROC of 98.56%. Zero-shot methods are generally more sensitive to domain and generator variation: the strongest Binoculars achieves average AUROCs of 76.40% and 83.31% on Multi-Domain and Multi-LLM, respectively. In contrast, supervised methods can leverage source labels to learn more stable decision boundaries. PEC-Detector maintains a consistent advantage over strong supervised methods, indicating that explicitly modeling the coupling among remote discourse history, local continuation context, and current writing execution provides useful evidence in addition to content and structural cues. Complete F1 results are provided in Appendix C.1.

Generalization Analysis. We assess generalization under the two distribution shift settings in MAGE. As shown in Table 2, under *Fixed-domain & Arbitrary-models*, the domain is fixed while generators varies. PEC-Detector achieves the best AUROC across all four domains, averaging 92.99% and outperforming second-best PRDetect by 1.99 percentage points. This result suggests that characterizing the relation between remote discourse states and current writing execution on the local context state provides discriminative evidence that is not fully tied to domain-specific content. Under *Arbitrary-domains & Model-specific*, the generator is fixed while text domain vary. PEC-Detector again achieves the highest AUROC across all four LLMs, reaching an average of 95.26% and exceeding the second-best BiScope by 2.33 percentage points. The result indicates that the planning-execution coupling captures discourse-level source regularities shared across generators beyond generator-specific surface statistics. Complete F1 results are provided in Appendix C.2. Supplementary evaluations on text generated by more recent LLMs (Appendix C.3) further show that PEC-Detector achieves the highest average performance among the baselines.

Table 1: Overall AUROC (%) performance on multi-Domain and multi-LLM assessment, with the best and second-best results in each column highlighted in **bold** and underlined, respectively.

Method	Multi-Domain					Multi-LLM				
	ArXiv	WP	XSum	Yelp	Avg.	GPT-3.5	Claude	Llama-2	PaLM-2	Avg.
Zero-shot methods										
DetectGPT	22.15	58.95	12.21	45.03	34.59	43.46	32.86	36.71	26.72	34.94
Fast-DetectGPT	43.69	74.21	39.19	22.97	45.02	65.56	30.01	76.79	65.99	59.59
DNA-GPT	55.85	66.27	72.18	30.20	56.13	61.87	48.88	75.22	71.48	64.36
Binoculars	84.03	94.38	77.39	49.81	76.40	88.14	55.15	96.64	93.30	83.31
Supervised methods										
RoBERTa-base	**98.74**	97.38	98.56	98.97	98.41	98.60	97.35	96.75	91.79	96.12
LF+Motifs	97.99	97.79	32.38	97.72	81.47	97.32	97.94	97.98	96.36	97.40
Raidar	94.94	85.45	96.11	80.12	89.16	86.58	87.71	89.19	90.43	88.48
BiScope	65.65	71.25	78.66	79.64	73.80	67.86	60.63	71.03	76.68	69.05
PRDetect	98.49	97.88	98.73	98.87	98.49	98.87	98.68	96.85	94.87	97.32
RepreGuard	97.93	97.39	97.68	98.62	97.91	94.06	77.29	79.94	70.86	80.54
DetectAnyLLM	97.00	97.95	97.99	97.86	97.70	98.99	97.54	97.95	96.95	97.86
ProSSD	93.94	92.41	93.84	91.84	93.01	93.77	87.30	91.05	90.36	90.62
PEC-Detector	98.42	**99.57**	**99.08**	**99.75**	**99.21**	**99.61**	**99.23**	**98.40**	**97.00**	**98.56**

Table 2: Overall AUROC (%) performance under distribution shifts on MAGE. *FD&AM* denotes *Fixed-domain & Arbitrary-models* and *AD&MS* denotes *Arbitrary-domains & Model-specific*.

Method	*FD&AM*					*AD&MS*				
	CMV	SciGen	WP	XSum	Avg.	GLM-130B	GPT-3.5	GPT-J	Bloom-7B	Avg.
Zero-shot methods										
DetectGPT	69.56	65.58	69.19	70.10	68.61	34.68	40.16	46.74	30.69	38.07
Fast-DetectGPT	48.25	40.14	41.67	35.01	41.27	91.36	78.89	35.24	92.26	74.44
DNA-GPT	64.55	44.22	59.90	45.48	53.54	92.09	90.75	21.89	90.12	73.71
Binoculars	71.21	59.75	67.50	57.82	64.07	92.61	93.19	14.51	92.68	73.25
Supervised methods										
RoBERTa-base	90.88	89.62	90.13	79.69	87.58	86.39	86.25	87.69	85.33	86.42
LF+Motifs	95.96	90.36	96.75	71.23	88.58	91.99	92.94	93.77	92.35	92.76
Raidar	79.38	83.58	81.30	74.78	79.76	85.68	71.65	89.52	82.49	82.34
BiScope	70.25	80.02	89.05	75.09	78.60	93.22	92.38	92.05	94.06	92.93
PRDetect	92.91	89.46	95.86	85.76	91.00	92.80	92.33	92.67	93.53	92.83
RepreGuard	69.26	87.34	76.93	69.23	75.69	84.94	77.79	73.14	74.75	77.66
DetectAnyLLM	88.48	89.01	90.68	84.37	88.14	92.14	91.99	93.14	92.87	92.54
ProSSD	83.30	67.27	86.38	74.42	77.84	83.10	74.72	79.38	81.79	79.75
PEC-Detector	**96.45**	**90.78**	**97.52**	**87.20**	**92.99**	**94.98**	**95.49**	**95.84**	**94.71**	**95.26**

Robustness Against Attacks. Adversaries may attempt to evade detection through rewrites or perturbations that modify detector cues. We evaluate PEC-Detector under the *Multi-Attack* setting of DetectRL. As shown in Table 3, PEC-Detector achieves the highest AUROC and F1 across all four attack types, with average scores of 99.54% and 97.23%, respectively, surpassing the strongest baseline DetectAnyLLM by 2.03 and 1.34 percentage points. On Prompt Attack, it still attains an AUROC of 99.18%. This robustness is consistent with source prediction based on cross-position planning-execution coupling patterns linking remote discourse states, local context conditions, and current writing execution, rather than isolated surface cues. Attacks that do not reconstruct these planning-execution relations may therefore leave detectable evidence in the coupling trajectory.

4.3 Ablation Study

We conduct ablations under the *Arbitrary-domains & Model-specific* setting of MAGE. As shown in Table 4, *semantic-only* reduces AUROC by 1.91-2.78 percentage points across the four generators, whereas *context-only* reduces it by 0.93-1.97 points. Replacing the trajectory encoder with mean

Table 3: Overall AUROC and F1 (%) performance on multi-Attack assessment.

Method	Paraphrase		Perturbation		Prompt		Data Mixing		Avg.	
	AUROC	F1	AUROC	F1	AUROC	F1	AUROC	F1	AUROC	F1
Zero-shot methods										
DetectGPT	31.79	16.89	18.21	0.00	53.81	40.17	34.56	0.40	34.59	14.37
Fast-DetectGPT	70.12	62.89	49.56	41.23	28.64	9.39	67.23	59.78	53.89	43.32
DNA-GPT	65.61	54.94	40.45	2.73	13.41	3.48	62.14	50.89	45.40	28.01
Binoculars	88.34	81.56	76.89	69.34	10.01	0.58	89.12	83.67	66.09	58.79
Supervised methods										
RoBERTa-base	98.31	95.49	98.67	96.72	97.59	95.60	93.82	87.19	97.10	93.75
LF+Motifs	97.97	95.22	97.94	95.59	96.96	95.84	96.99	96.80	97.47	95.86
Raidar	66.40	59.97	91.40	83.37	81.72	74.96	88.35	80.08	81.97	74.60
BiScope	89.84	66.93	97.24	66.75	95.00	73.95	96.91	93.17	94.75	75.20
PRDetect	97.63	94.14	97.74	95.22	97.32	93.97	95.59	92.87	97.07	94.05
RepreGuard	92.64	86.14	93.76	87.30	73.14	73.33	88.50	77.70	87.01	81.12
DetectAnyLLM	97.95	95.90	98.00	98.00	97.76	95.65	96.34	94.00	97.51	95.89
ProSSD	88.50	82.53	87.25	80.93	89.56	86.17	93.66	86.96	89.74	84.15
PEC-Detector	**99.50**	**96.47**	**99.85**	**98.08**	**99.18**	**95.98**	**99.63**	**98.37**	**99.54**	**97.23**

Table 4: Ablation study on MAGE under the *Arbitrary-domains & Model-specific* setting.

Method	GLM-130B		GPT-3.5		GPT-J		Bloom-7B	
	AUROC	F1	AUROC	F1	AUROC	F1	AUROC	F1
semantic-only	93.07	85.09	93.11	86.71	93.66	85.68	91.93	83.93
context-only	93.54	84.51	93.52	87.77	94.56	87.69	93.78	85.03
w/o remote planning state	93.22	85.65	94.29	87.54	93.62	86.52	93.78	81.06
w/o local context state	93.76	84.49	94.35	87.43	94.62	88.00	93.59	83.62
w/o conditional prediction gain	93.61	82.10	94.65	87.73	95.06	88.07	94.21	86.03
mean pooling	93.52	84.58	93.79	87.63	94.48	87.48	93.88	84.62
w/o auxiliary prediction	94.18	85.20	94.55	87.75	94.47	87.81	93.73	82.98
PEC-Detector	**94.98**	**86.92**	**95.49**	**89.36**	**95.84**	**88.74**	**94.71**	**87.25**

pooling further decreases AUROC by 0.83-1.70 points. These consistent degradations indicate that the gains of PEC-Detector arise from modeling planning-execution relations rather than enhancing content semantics or local context in isolation. Notably, the contributions of individual components vary across generators. Removing the remote discourse planning state reduces F1 by 2.22 and 6.19 points on GPT-J and Bloom-7B, respectively, whereas removing the local context state causes a larger F1 decrease on GLM-130B. Removing either the conditional prediction gain or the auxiliary conditional semantic prediction objective also degrades overall performance. These results indicate that remote planning, local continuation, and conditional prediction provide complementary evidence for source prediction.

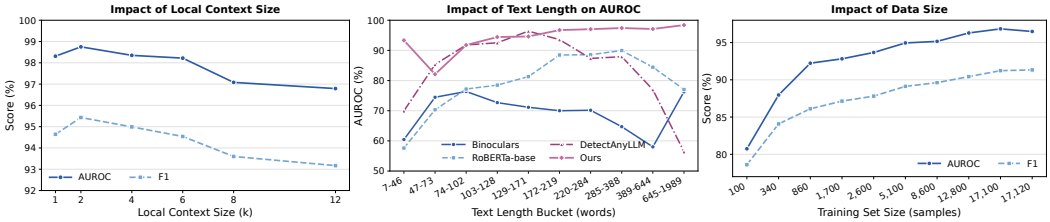
4.4 Experimental Analysis

Impact of Sentence Order. To test whether PEC-Detector uses document order beyond sentence content and short-range continuity, we keep all model parameters fixed and perturb sentence order only at test time. We compare fully random sentence-level shuffling and three-sentence block-level shuffling; the latter preserves local continuity within each block while disrupting discourse organization across blocks. As shown in Table 5, sentence-level shuffling reduces PEC-Detector's AUROC by 5.03 percentage points, while block-level shuffling still causes a 3.96-point decrease. Under complete shuffling, the largest AUROC change among the three baselines is only 1.59 points. These results indicate that PEC-Detector exploits the ordered organization of planning-execution relations within a document, rather than relying solely on sentence content or local statistical patterns.

Impact of Local Context Size. The local-context window K determines the number of preceding sentences used to form the local continuation state. We evaluate $K \in \{1, 2, 4, 6, 8, 12\}$ on the same

Table 5: Impact of sentence-order perturbations on MAGE under the CMV dataset.

Method	Original		Sentence Shuffled		Block Shuffled	
	AUROC	F1	AUROC	F1	AUROC	F1
Binoculars	71.21	78.79	69.81	77.92	69.88	77.96
RoBERTa-base	<u>90.88</u>	85.75	<u>89.85</u>	84.41	<u>91.26</u>	84.85
DetectAnyLLM	88.48	<u>87.88</u>	86.89	**85.82**	88.22	**86.68**
PEC-Detector	**96.45**	**90.84**	**91.42**	<u>85.76</u>	**92.49**	<u>86.24</u>



(a) Impact of local context size k (b) Impact of text length on AUROC (c) Impact of training data scale

Figure 3: Effects of key hyperparameters and data conditions on PEC-Detector performance.

subset of documents with at least 14 sentences, ensuring that every document remains evaluable at $K = 12$. As shown in Figure 3a, both AUROC and F1 peak at $K = 2$. Relative to $K = 1$, adding a second preceding sentence improves AUROC by 0.44 percentage points, suggesting that a single sentence does not reliably characterize local continuation conditions of current writing. Increasing K further yields no additional gain, likely because earlier discourse is increasingly absorbed into the local state and weakens the separation between local continuation and remote discourse planning.

Impact of Text Length. Document length affects the number of observable planning-execution positions. We partition the CMV test set into ten bins by document length. As shown in Figure 3b, PEC-Detector achieves the best AUROC in seven bins and remains the top-performing method with lengths exceeding 172, reaching a peak AUROC of 98.39%. Several baselines decline on longer documents, potentially because increasingly heterogeneous content weakens evidence based on global aggregation or local surface statistics. In contrast, PEC-Detector maintains stable source prediction evidence when the document shows sufficient discourse progression. Shorter documents provide fewer position-wise coupling relations and consequently exhibit performance variation.

Impact of Data Scale. We further examine the effect of training data scale. Figure 3c shows that AUROC increases from 80.74% with 100 training examples to 96.49% with 17,120 examples. The largest improvement occurs in the low-resource regime: AUROC reaches 92.23% with 860 examples and 94.95% with 5,100 examples. Performance continues to improve thereafter, albeit with diminishing returns and minor fluctuations near saturation, indicating that planning-execution coupling trajectories can be learned effectively from limited labeled data.

Further Analysis Results on multilingual datasets and the encoder robustness analysis are provided in Appendices C.4 and C.5, respectively.

5 Conclusion

We revisit document-level machine-generated text detection from the perspective of how a document unfolds. The contribution of remote discourse history to current writing after conditioning on local context follows distinct position-wise patterns in MGT and HWT. Motivated by this observation, we propose **PEC-Detector**, which represents a document as an ordered planning-execution coupling trajectory. Experiments on multiple public benchmarks demonstrate superior detection performance across domains, generators, languages, and adversarial attacks. Ablation study and sentence order perturbation results further show that the gains of **PEC-Detector** depend on the ordered organization of planning-execution relations, rather than content semantics or local surface cues alone.

AI USE STATEMENT

In this work, we used generative AI tools solely to edit the manuscript for clarity and readability and to assist with diagnosing and resolving software bugs encountered during implementation. All AI-assisted text edits and code suggestions were reviewed and verified by the authors. We take responsibility for the final content of this work.

ETHICS STATEMENT

This work introduces a document-level framework for machine-generated text detection, whose strong performance on the evaluated benchmarks does not imply immunity to outputs from more advanced LLMs or to adaptive attacks. The resulting scores should be interpreted as probabilistic evidence rather than conclusive proof that a text is human-written or machine-generated, and should never constitute the sole basis for high-stakes decisions, where careful human oversight remains indispensable.

REPRODUCIBILITY STATEMENT

To facilitate reproducibility, Appendix B provides a detailed experimental setup, including implementation and dataset split protocols. All datasets used in this study are publicly available, and their sources and descriptive statistics are documented in the appendix.

REFERENCES

Guangsheng Bao, Yanbin Zhao, Zhiyang Teng, Linyi Yang, and Yue Zhang. Fast-detectgpt: Efficient zero-shot detection of machine-generated text via conditional probability curvature. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=Bpcgcr8E8Z.

Claudio Borile and Carlo Abrate. How to generalize the detection of ai-generated text: Confounding neurons. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2025, Suzhou, China, November 4-9, 2025*, pp. 25461–25476. Association for Computational Linguistics, 2025. doi: 10.18653/V1/2025.FINDINGS-EMNLP.1388. URL https://doi.org/10.18653/v1/2025.findings-emnlp.1388.

Xin Chen, Junchao Wu, Shu Yang, Runzhe Zhan, Zeyu Wu, Ziyang Luo, Di Wang, Min Yang, Lidia S. Chao, and Derek F. Wong. Repreguard: Detecting llm-generated text by revealing hidden representation patterns. *Trans. Assoc. Comput. Linguistics*, 13:1812–1831, 2025a. doi: 10.1162/TACL.A.61. URL https://doi.org/10.1162/tacl.a.61.

Zheng Chen, Yushi Feng, Jisheng Dang, Changyang He, Yue Deng, Hongxi Pu, Haoxuan Li, and Bo Li. IPAD: inverse prompt for AI detection - A robust and interpretable llm-generated text detector. In Danielle Belgrave, Cheng Zhang, Laura N. Montoya, Hsuan-Tien Lin, Razvan Pascanu, Piotr Koniusz, Marzyeh Ghassemi, Nancy Chen, Iván Vladimir Meza Ruíz, and Arturo Loaiza-Bonilla (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025b. URL http://papers.nips.cc/paper_files/paper/2025/hash/f4d6932b6b9eec9b9d595e2847d095c1-Abstract-Conference.html.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pp. 4171–4186. Association for Computational Linguistics, 2019. doi: 10.18653/V1/N19-1423. URL https://doi.org/10.18653/v1/n19-1423.

Jiachen Fu, Chun-Le Guo, and Chongyi Li. Detectanyllm: Towards generalizable and robust detection of machine-generated text across domains and models. In Cathal Gurrin, Klaus Schoeffmann, Min Zhang, Luca Rossetto, Stevan Rudinac, Duc-Tien Dang-Nguyen, Wen-Huang Cheng, Phoebe Chen, and Jenny Benois-Pineau (eds.), *Proceedings of the 33rd ACM International Conference on Multimedia, MM 2025, Dublin, Ireland, October 27-31, 2025*, pp. 11229–11238. ACM, 2025. doi: 10.1145/3746027.3754862. URL https://doi.org/10.1145/3746027.3754862.

Sebastian Gehrmann, Hendrik Strobelt, and Alexander M. Rush. GLTR: statistical detection and visualization of generated text. In Marta R. Costa-jussà and Enrique Alfonseca (eds.), *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28 - August 2, 2019, Volume 3: System Demonstrations*, pp. 111–116. Association for Computational Linguistics, 2019. doi: 10.18653/V1/P19-3019. URL https://doi.org/10.18653/v1/p19-3019.

Hanxi Guo, Siyuan Cheng, Xiaolong Jin, Zhuo Zhang, Kaiyuan Zhang, Guanhong Tao, Guangyu Shen, and Xiangyu Zhang. Biscope: Ai-generated text detection by checking memorization of preceding tokens. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/bc808cf2d2444b0abcceca366b771389-Abstract-Conference.html.

Rui Guo, Weibin Zeng, Fuzhang Wu, Yan Kong, Yanjun Wu, Weiming Dong, et al. Hld: Approximate hierarchical linguistic distribution modeling for llm-generated text detection. In *The Fourteenth International Conference on Learning Representations*, 2026.

Abhimanyu Hans, Avi Schwarzschild, Valeriia Cherepanova, Hamid Kazemi, Aniruddha Saha, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Spotting llms with binoculars: Zero-shot detection of machine-generated text. In Ruslan Salakhutdinov, Zico Kolter, Katherine A. Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, volume 235 of *Proceedings of Machine Learning Research*, pp. 17519–17537. PMLR / OpenReview.net, 2024. URL https://proceedings.mlr.press/v235/hans24a.html.

Xiaomeng Hu, Pin-Yu Chen, and Tsung-Yi Ho. RADAR: robust ai-text detection via adversarial learning. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/30e15e5941ae0cdab7ef58cc8d59a4ca-Abstract-Conference.html.

Baizhou Huang and Xiaojun Wan. Waterpool: A language model watermark mitigating trade-offs among imperceptibility, efficacy and robustness. In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2025 - Volume 1: Long Papers, Albuquerque, New Mexico, USA, April 29 - May 4, 2025*, pp. 4156–4182. Association for Computational Linguistics, 2025. doi: 10.18653/V1/2025.NAACL-LONG.209. URL https://doi.org/10.18653/v1/2025.naacl-long.209.

Haozhe Ji and Minlie Huang. Discodvt: Generating long text with discourse-aware discrete variational transformer. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pp. 4208–4224. Association for Computational Linguistics, 2021. doi: 10.18653/V1/2021.EMNLP-MAIN.347. URL https://doi.org/10.18653/v1/2021.emnlp-main.347.

Zae Myung Kim, Kwang Hee Lee, Preston Zhu, Vipul Raheja, and Dongyeop Kang. Threads of subtlety: Detecting machine-generated texts through discourse motifs. In Lun-Wei Ku, Andre

Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pp. 5449–5474. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.ACL-LONG.298. URL https://doi.org/10.18653/v1/2024.acl-long.298.

Xiang Li, Zhiyi Yin, Hexiang Tan, Shaoling Jing, Du Su, Yi Cheng, Huawei Shen, and Fei Sun. Prdetect: Perturbation-robust llm-generated text detection based on syntax tree. In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), *Findings of the Association for Computational Linguistics: NAACL 2025, Albuquerque, New Mexico, USA, April 29 - May 4, 2025*, volume NAACL 2025 of *Findings of ACL*, pp. 8290–8301. Association for Computational Linguistics, 2025. doi: 10.18653/V1/2025.FINDINGS-NAACL.464. URL https://doi.org/10.18653/v1/2025.findings-naacl.464.

Yafu Li, Qintong Li, Leyang Cui, Wei Bi, Zhilin Wang, Longyue Wang, Linyi Yang, Shuming Shi, and Yue Zhang. MAGE: machine-generated text detection in the wild. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pp. 36–53. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.ACL-LONG.3. URL https://doi.org/10.18653/v1/2024.acl-long.3.

Xiaoming Liu, Zhaohan Zhang, Yichen Wang, Hang Pu, Yu Lan, and Chao Shen. Coco: Coherence-enhanced machine-generated text detection under low resource with contrastive learning. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pp. 16167–16188. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.EMNLP-MAIN.1005. URL https://doi.org/10.18653/v1/2023.emnlp-main.1005.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019. URL http://arxiv.org/abs/1907.11692.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Shixuan Ma and Quan Wang. Zero-shot detection of llm-generated text using token cohesiveness. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pp. 17538–17553. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.EMNLP-MAIN.971. URL https://doi.org/10.18653/v1/2024.emnlp-main.971.

Chengzhi Mao, Carl Vondrick, Hao Wang, and Junfeng Yang. Raidar: generative AI detection via rewriting. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=bQWE2UqXmf.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D. Manning, and Chelsea Finn. Detectgpt: Zero-shot machine-generated text detection using probability curvature. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 24950–24962. PMLR, 2023. URL https://proceedings.mlr.press/v202/mitchell23a.html.

Jingwei Ni, Minjing Shi, Dominik Stammbach, Mrinmaya Sachan, Elliott Ash, and Markus Leippold. Afacta: Assisting the annotation of factual claim detection with reliable LLM annotators. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024,*

Bangkok, Thailand, August 11-16, 2024, pp. 1890–1912. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.ACL-LONG.104. URL https://doi.org/10.18653/v1/2024.acl-long.104.

Yuhui Shi, Qiang Sheng, Juan Cao, Hao Mi, Beizhe Hu, and Danding Wang. 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, IJCAI 2024, Jeju, South Korea, August 3-9, 2024*, pp. 494–502. ijcai.org, 2024. URL https://www.ijcai.org/proceedings/2024/55.

Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, and Jasmine Wang. Release strategies and the social impacts of language models. *CoRR*, abs/1908.09203, 2019. URL http://arxiv.org/abs/1908.09203.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. Mpnet: Masked and permuted pre-training for language understanding. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/c3a690be93aa602ee2dc0ccab5b7b67e-Abstract.html.

Jinyan Su, Terry Yue Zhuo, Di Wang, and Preslav Nakov. Detectllm: Leveraging log rank information for zero-shot detection of machine-generated text. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, volume EMNLP 2023 of *Findings of ACL*, pp. 12395–12412. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.FINDINGS-EMNLP.827. URL https://doi.org/10.18653/v1/2023.findings-emnlp.827.

Yufei Tian, Zeyu Pan, and Nanyun Peng. Detecting machine-generated long-form content with latent-space variables. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Florida, USA, November 12-16, 2024*, volume EMNLP 2024 of *Findings of ACL*, pp. 10394–10408. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.FINDINGS-EMNLP.608. URL https://doi.org/10.18653/v1/2024.findings-emnlp.608.

Brian Tufts, Xuandong Zhao, and Lei Li. A practical examination of ai-generated text detectors for large language models. In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), *Findings of the Association for Computational Linguistics: NAACL 2025, Albuquerque, New Mexico, USA, April 29 - May 4, 2025*, volume NAACL 2025 of *Findings of ACL*, pp. 4824–4841. Association for Computational Linguistics, 2025. doi: 10.18653/V1/2025.FINDINGS-NAACL.271. URL https://doi.org/10.18653/v1/2025.findings-naacl.271.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 5998–6008, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Quan Wang, Licheng Zhang, Zikang Guo, and Zhendong Mao. IDEATE: detecting ai-generated text using internal and external factual structures. In Nicoletta Calzolari, Min-Yen Kan, Véronique Hoste, Alessandro Lenci, Sakriani Sakti, and Nianwen Xue (eds.), *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation, LREC/COLING 2024, 20-25 May, 2024, Torino, Italy*, pp. 8556–8568. ELRA and ICCL, 2024a. URL https://aclanthology.org/2024.lrec-main.751.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Osama Mohammed Afzal, Tarek Mahmoud, Giovanni Puccetti, Thomas Arnold, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. M4gt-bench: Evaluation benchmark for black-box machine-generated text detection. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*

(Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024, pp. 3964–3992. Association for Computational Linguistics, 2024b. doi: 10.18653/V1/2024.ACL-LONG.218. URL https://doi.org/10.18653/v1/2024.acl-long.218.

Junchao Wu, Runzhe Zhan, Derek F. Wong, Shu Yang, Xinyi Yang, Yulin Yuan, and Lidia S. Chao. Detectrl: Benchmarking llm-generated text detection in real-world scenarios. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/b61bdf7e9f64c04ec75a26e781e2ad51-Abstract-Datasets_and_Benchmarks_Track.html.

Junchao Wu, Shu Yang, Runzhe Zhan, Yulin Yuan, Lidia S. Chao, and Derek Fai Wong. A survey on llm-generated text detection: Necessity, methods, and future directions. *Comput. Linguistics*, 51 (1):275–338, 2025a. doi: 10.1162/COLI_A_00549. URL https://doi.org/10.1162/coli_a_00549.

Junchao Wu, Runzhe Zhan, Derek F. Wong, Shu Yang, Xuebo Liu, Lidia S. Chao, and Min Zhang. Who wrote this? the key to zero-shot llm-generated text detection is gecscore. In Owen Rambow, Leo Wanner, Marianna Apidianaki, Hend Al-Khalifa, Barbara Di Eugenio, and Steven Schockaert (eds.), *Proceedings of the 31st International Conference on Computational Linguistics, COLING 2025, Abu Dhabi, UAE, January 19-24, 2025*, pp. 10275–10292. Association for Computational Linguistics, 2025b. URL https://aclanthology.org/2025.coling-main.684/.

Ruochong Xiong, Qien Li, Wangwang Lian, Yulong Wan, Hanlin Xue, Zhouxing Tan, Han Yang, Fengyu Lu, and Junfei Liu. Verifiable llm-generated text detection via projected semantic-structural distributions. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2026, San Diego, California, United States, July 2-7, 2026*, pp. 14005–14042. Association for Computational Linguistics, 2026. doi: 10.18653/V1/2026.ACL-LONG.638. URL https://doi.org/10.18653/v1/2026.acl-long.638.

Han Xu, Jie Ren, Pengfei He, Shenglai Zeng, Yingqian Cui, Amy Liu, Hui Liu, and Jiliang Tang. On the generalization of training-based chatgpt detection methods. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Florida, USA, November 12-16, 2024*, volume EMNLP 2024 of *Findings of ACL*, pp. 7223–7243. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.FINDINGS-EMNLP.424. URL https://doi.org/10.18653/v1/2024.findings-emnlp.424.

Xianjun Yang, Wei Cheng, Yue Wu, Linda Ruth Petzold, William Yang Wang, and Haifeng Chen. DNA-GPT: divergent n-gram analysis for training-free detection of gpt-generated text. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=Xlayxj2fWp.

Lili Yao, Nanyun Peng, Ralph M. Weischedel, Kevin Knight, Dongyan Zhao, and Rui Yan. Plan-and-write: Towards better automatic storytelling. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pp. 7378–7385. AAAI Press, 2019. doi: 10.1609/AAAI.V33I01.33017378. URL https://doi.org/10.1609/aaai.v33i01.33017378.

Cong Zeng, Shengkun Tang, Xianjun Yang, Yuanzhou Chen, Yiyou Sun, Zhiqiang Xu, Yao Li, Haifeng Chen, Wei Cheng, and Dongkuan Xu. DALD: improving logits-based detector without logits from black-box llms. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 -*

15, 2024, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/62e5f22b5dae99ec700be622df4fbe0d-Abstract-Conference.html.

Xuandong Zhao, Prabhanjan Vijendra Ananth, Lei Li, and Yu-Xiang Wang. Provable robust watermarking for ai-generated text. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=SsmT8aO45L.

Xiaowei Zhu, Yubing Ren, Fang Fang, Shi Wang, Yanan Cao, and Li Guo. Exons-detect: Identifying and amplifying exonic tokens via hidden-state discrepancy for robust ai-generated text detection. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2026, San Diego, California, United States, July 2-7, 2026*, pp. 26324–26336. Association for Computational Linguistics, 2026. doi: 10.18653/V1/2026.ACL-LONG.1211. URL https://doi.org/10.18653/v1/2026.acl-long.1211.

A Conditional Contribution of Remote Discourse History

This section details how we compute the conditional contribution of remote discourse history, as shown in Figure 1. We conduct this analysis with an independent label-free conditional semantic predictor, rather than the **PEC-Detector** used for source prediction. Specifically, we examine whether remote discourse history reduces the semantic prediction error of the current sentence once local context has been observed, and whether the resulting predictive gain exhibits different position-wise patterns in MGT and HWT.

Conditional semantic prediction. At each position, a shared predictor estimates the semantic representation of the current sentence under two conditions, with and without access to the planning state. The two conditions share predictor parameters, the local state, and the prediction target, differing only in whether remote history is available. We compute the prediction error $\ell_t(m)$ and the conditional predictive gain δ_t attributable to remote history using Equation equation 4, where $m = 0$ denotes prediction from local context alone, and $m = 1$ denotes prediction conditioned jointly on remote history and local context. We optimize the predictor using the mean prediction error over both conditions, $\mathcal{L}_{\text{pred}}$, without any source-classification objective.

Document-level contribution. For the set of valid positions $\mathcal{I}_d$ in a document, we aggregate the position-level gains into a document-level conditional contribution:

$$G_d = \frac{1}{|\mathcal{I}_d|} \sum_{t \in \mathcal{I}_d} \delta_{d,t}. \tag{9}$$

Accordingly, $G_d > 0$ indicates that incorporating remote history reduces the average semantic prediction error of the current sentences in the document. Figure 1a shows the distribution of G_d over all test documents in the CMV dataset, without partitioning by source label. We further perform $2{,}000$ document-level bootstrap resamples to estimate a 95% confidence interval for the mean gain.

Position-wise trajectory analysis. To compare the evolution of conditional contributions across documents of different lengths, we map the j-th valid position in a document to a normalized writing progress:

$$q_{d,j} = \frac{j}{\max(|\mathcal{I}_d| - 1, 1)}, \qquad j = 0, \ldots, |\mathcal{I}_d| - 1. \tag{10}$$

We then divide $[0, 1]$ into six equal-width progress bins. For each bin, we first average position-wise gains within each document and subsequently average these document-level values within each source group. This two-stage aggregation assigns equal weight to each document within a progress bin, preventing longer documents from dominating the overall trend merely because they contain more valid positions. Figure 1b reports the mean gain of MGT and HWT in each progress bin, with pointwise 95% confidence intervals obtained from $2{,}000$ document-level bootstrap resamples.

Use of source labels. Source labels are not used for sentence encoding, conditional-predictor training, or checkpoint selection. Figure 1a is constructed solely from label-free statistics over the pooled test documents. Test labels are used only in Figure 1(b) to post hoc partition the already computed

document-level progress gains into MGT and HWT groups. Accordingly, this analysis characterizes the conditional predictive role of remote history beyond local context and the resulting trajectory differences between the two sources, rather than a classification boundary learned directly from source labels.

B Experimental Setup

B.1 Datasets

DetectRL Wu et al. (2024) evaluates MGT detection under realistic variation in writing styles, generators, and adversarial perturbation attacks. We use its *Multi-Domain*, *Multi-LLM*, and *Multi-Attack* settings. The Multi-Domain setting covers academic writing (ArXiv), news writing (XSum), creative writing (WP), and social media (Yelp), whereas Multi-LLM varies the generator among GPT-3.5-turbo, Claude-instant, PaLM-2-bison, and Llama-2-70b. Multi-Attack evaluates prompt, paraphrase, perturbation, and data mixing. For each DetectRL setting, we preserve the benchmark test partition and split its original training partition into training and dev subsets at an 80:20 ratio.

MAGE Li et al. (2024) is a large-scale testbed for MGT detection. We use two of its generalization settings. In *Fixed-domain & Arbitrary-models* (FD&AM), the human-text domain is fixed while MGT spans multiple LLMs; we evaluate opinion statements (CMV), scientific articles (SciGen), Reddit WritingPrompts (WP), and news articles (XSum) separately. We retain the prepared train/dev/test files for each MAGE testbed. In *Arbitrary-domains & Model-specific* (AD&MS), human texts span multiple domains while MGT is generated by one fixed LLM; we evaluate GLM-130B, GPT-3.5-turbo, GPT-J-6B, and Bloom-7B separately.

To assess detection performance across languages, we use the multilingual binary detection subset of M4GT-Bench Wang et al. (2024b), selecting Chinese, German, and Russian texts. We construct splits independently for each language using two-stage label-stratified sampling. For each language, we construct label-stratified train/dev/test splits with ratios of approximately 64:16:20. This protocol preserves the class distribution within each language. Table 6 summarizes the statistics of the datasets used in our experiments.

B.2 Details of the Baselines

DetectGPT (Mitchell et al., 2023) approximates the probability curvature of a text by comparing its log probability under a surrogate language model with those of perturbed variants. It is based on the observation that machine-generated text is more likely to lie in negative-curvature regions of the model's log-probability function.

Fast-DetectGPT (Bao et al., 2024) replaces the computationally expensive text-perturbation procedure of DetectGPT with conditional probability curvature. It uses the conditional distributions of a scoring model and a reference model to construct an efficient sampling-based estimate of the curvature score.

DNA-GPT (Yang et al., 2024) truncates the input text and uses an LLM to regenerate the continuation. It then measures the divergence between the original and regenerated continuations through n-gram statistics, assessing whether the text conforms to the continuation pattern of a generative model.

Binoculars (Hans et al., 2024) uses two LLMs, termed an observer and a performer, and compares text perplexity with cross-perplexity. Their ratio reflects the agreement between the conditional distributions induced by the two models and serves as a source classification score.

RoBERTa Classifier (Liu et al., 2019) uses RoBERTa as a contextual encoder and fine-tunes a binary classifier over document representations with source labels. This baseline measures the performance of directly learning a source decision boundary from the semantic representation of the complete text.

LF+Motifs (Kim et al., 2024) augments a Longformer document classifier with frequency features of discourse motifs extracted from hierarchical discourse structures. It combines contextual rep-

Table 6: Statistics of the experimental datasets. FD&AM denotes *Fixed-domain & Arbitrary-models*, and AD&MS denotes *Arbitrary-domains & Model-specific*.

Benchmark	Setting	Subsetting	Train	Dev.	Test	MGT	HWT	Total
DetectRL	Multi-Domain	ArXiv	20,792	5,198	2,008	25,198	2,800	27,998
		XSum	20,793	5,199	2,008	25,200	2,800	28,000
		WP	20,788	5,197	2,008	25,193	2,800	27,993
		Yelp	20,787	5,197	2,008	25,192	2,800	27,992
	Multi-LLM	GPT-3.5	20,789	5,198	2,008	25,195	2,800	27,995
		Claude	20,792	5,198	2,008	25,198	2,800	27,998
		PaLM-2	20,789	5,198	2,008	25,195	2,800	27,995
		Llama-2	20,789	5,198	2,008	25,195	2,800	27,995
	Multi-Attack	Prompt	25,254	6,314	2,032	22,400	11,200	33,600
		Paraphrase	34,213	8,554	2,016	33,583	11,200	44,783
		Perturbation	34,227	8,557	2,016	33,600	11,200	44,800
		Data Mixing	34,230	8,558	2,012	22,400	22,400	44,800
MAGE	FD&AM	CMV	24,611	4,933	4,917	9,062	25,399	34,461
		SciGen	23,127	4,843	4,789	9,505	23,254	32,759
		WP	31,159	6,214	6,236	12,588	31,021	43,609
		XSum	30,759	6,529	6,537	11,265	32,560	43,825
	AD&MS	GLM-130B	14,688	1,838	1,838	9,182	9,182	18,364
		GPT-3.5-turbo	106,678	13,264	13,284	66,613	66,613	133,226
		GPT-J-6B	23,170	2,800	2,884	14,427	14,427	28,854
		Bloom-7B	59,690	7,398	7,420	37,254	37,254	74,508
M4GT-Bench	Multilingual	Chinese	7,636	1,910	2,387	5,933	6,000	11,933
		German	4,480	1,120	1,400	3,500	3,500	7,000
		Russian	1,280	320	400	1,000	1,000	2,000

resentations of long texts with discourse-relation patterns to capture structural differences between MGT and HWT.

Raidar (Mao et al., 2024) prompts an LLM to rewrite the input and uses the edit distance between the original and rewritten texts as the detection signal. Its underlying observation is that language models tend to make more revisions to HWT than to MGT.

BiScope (Guo et al., 2024) computes the forward cross-entropy between output logits and the ground-truth token, together with the backward cross-entropy between the same logits and a preceding input token. Statistics derived from these two losses are then fed to a classifier to characterize token memory patterns.

PRDetect (Li et al., 2025) constructs structural features from syntactic trees that are comparatively stable under perturbations and uses graph convolutional networks to classify these structures. It is motivated by the idea that local lexical modifications can alter surface forms without necessarily reconstructing the underlying syntactic organization.

RepreGuard (Chen et al., 2025a) extracts a discriminative activation direction between MGT and HWT from the internal representations of a surrogate language model, and classifies a test text according to its projection score along the resulting direction.

DetectAnyLLM (Fu et al., 2025) uses Direct Discrepancy Learning to optimize a scoring model such that the conditional-probability discrepancy between MGT and HWT becomes more discriminative, and performs detection through reference clustering. It aims to better align probability-based scores with the source-detection objective.

ProSSD (Xiong et al., 2026) performs supervised subspace learning in a joint semantic-syntactic space to construct syntactically conditioned semantic distributions. It then uses a weighted modified Mahalanobis distance as its discriminative statistic, capturing the systematic semantic shift of MGT relative to HWT.

Table 7: Overall F1 (%) performance on multi-Domain and multi-LLM assessment, with the best and second-best results in each column highlighted in bold and underlined, respectively.

Method	Multi-Domain					Multi-LLM				
	ArXiv	WP	XSum	Yelp	Avg.	GPT-3.5	Claude	Llama-2	PaLM-2	Avg.
Zero-shot methods										
DetectGPT	0.00	50.83	0.00	33.72	21.14	26.27	12.56	20.40	0.00	14.81
Fast-DetectGPT	24.46	67.84	28.39	11.42	33.03	59.55	0.00	69.08	57.58	46.55
DNA-GPT	22.07	65.05	65.30	8.99	40.35	55.04	25.67	62.89	60.77	51.09
Binoculars	76.77	79.73	72.18	21.89	62.64	82.50	39.35	92.30	88.20	75.59
Supervised methods										
RoBERTa-base	97.06	92.11	97.04	95.76	95.49	97.17	93.42	89.93	84.09	91.15
LF+Motifs	<u>97.70</u>	96.21	64.26	95.77	88.49	95.73	97.03	**97.03**	93.36	<u>95.79</u>
Raidar	87.98	80.37	88.89	74.88	83.03	82.36	80.36	81.45	82.69	81.72
BiScope	66.84	67.34	70.34	71.83	69.09	67.89	59.32	67.84	68.84	65.97
PRDetect	91.06	95.86	96.56	95.48	94.74	95.96	95.28	90.56	89.02	92.71
RepreGuard	92.53	91.36	94.17	96.70	93.69	84.57	57.89	80.95	73.57	74.25
DetectAnyLLM	96.95	<u>97.44</u>	<u>97.90</u>	<u>97.34</u>	<u>97.41</u>	<u>97.69</u>	<u>97.76</u>	90.60	**96.35**	95.60
ProSSD	91.85	73.91	66.89	66.95	74.90	69.15	66.84	67.77	66.93	67.67
PEC-Detector	**98.53**	**99.41**	**99.56**	**99.45**	**99.24**	**99.41**	**99.06**	<u>92.48</u>	<u>93.66</u>	**96.15**

B.3 Details of the Implementation

Except for the model architecture and training hyperparameters reported in Section 4.1, all experiments are implemented in PyTorch and conducted on two NVIDIA RTX 4090 GPUs. For every baseline, we preserved the original core detection mechanisms and standardized only the input format, output logging, and label convention to enable evaluation under the same data-splitting protocol. Specifically, DetectGPT Mitchell et al. (2023) uses GPT-Neo as the scoring model and T5-small as the perturbation model. Fast-DetectGPT Bao et al. (2024) uses GPT-Neo and GPT-J as the reference and scoring models, respectively. DNA-GPT Yang et al. (2024) scores the original and regenerated texts with GPT-Neo, while using Meta-Llama-3-8B-Instruct to generate continuations from truncated inputs. Binoculars Hans et al. (2024) follows its publicly released implementation. All supervised baselines use the same training, development, and test splits as our method, while retaining the released configurations for method-specific rewriting, structural processing, and optimization.

C Additional Results

C.1 Supplementary to Overall Performance

Table 7 reports the overall F1 results, which are consistent with the AUROC analysis in the main text. **PEC-Detector** achieves the highest average F1 in both the Multi-Domain and Multi-LLM settings, reaching 99.24% and 96.15%, respectively. In the Multi-Domain setting, our method obtains the best F1 across all four domains and surpasses the second-best baseline, DetectAnyLLM, by 1.83 percentage points on average. This result indicates that its advantage in ranking performance also translates into stable document-level source decisions. In the Multi-LLM setting, **PEC-Detector** achieves the best results on GPT-3.5 and Claude, ranks second on Llama-2 and PaLM-2, and exceeds the strongest baseline on average, LF+Motifs, by 0.36 percentage points. Notably, the F1 scores of zero-shot methods vary substantially across domains and generators, with several settings yielding near-zero values, suggesting that their statistical features may not consistently support binary decisions under distribution shifts. In contrast, **PEC-Detector** maintains high and consistent F1 across both evaluation settings, further supporting the value of modeling planning-execution coupling among remote discourse history, local context, and current writing execution for stable source prediction.

Table 8: Overall F1 (%) performance under distribution shifts on MAGE. *FD&AM* denotes *Fixed-domain & Arbitrary-models* and *AD&MS* denotes *Arbitrary-domains & Model-specific*. The best and second-best results in each column are highlighted in bold and underlined, respectively.

Method	*FD&AM*					*AD&MS*				
	CMV	**SciGen**	**WP**	**XSum**	**Avg.**	**GLM-130B**	**GPT-3.5**	**GPT-J**	**Bloom-7B**	**Avg.**
Zero-shot methods										
DetectGPT	65.98	77.77	70.62	67.05	70.36	72.39	21.87	37.25	71.88	50.85
Fast-DetectGPT	67.98	72.99	68.40	64.19	68.39	<u>86.38</u>	72.53	28.32	85.81	68.26
DNA-GPT	75.00	73.11	75.73	69.03	73.22	85.59	86.23	1.59	83.33	64.19
Binoculars	78.79	76.31	78.61	76.54	77.56	86.53	86.95	8.93	86.05	67.12
Supervised methods										
RoBERTa-base	85.75	83.95	81.25	73.60	81.14	83.47	79.84	78.54	78.46	80.08
LF+Motifs	<u>89.65</u>	84.22	90.51	12.17	69.14	84.27	86.91	86.51	85.77	85.87
Raidar	68.99	76.32	73.99	71.70	72.75	81.93	65.52	81.49	73.85	75.70
BiScope	2.86	13.04	5.46	3.35	6.18	85.98	87.52	87.42	**89.15**	87.52
PRDetect	87.62	82.67	<u>91.10</u>	69.54	82.73	86.04	79.98	**90.56**	85.19	85.44
RepreGuard	66.67	81.09	73.04	69.17	72.49	81.27	74.32	73.00	74.57	75.79
DetectAnyLLM	87.88	<u>87.96</u>	88.65	<u>80.59</u>	<u>86.27</u>	85.72	<u>88.48</u>	<u>89.68</u>	86.84	<u>87.68</u>
ProSSD	78.90	71.80	81.49	71.91	76.03	82.09	71.60	75.19	78.77	76.91
PEC-Detector	**90.84**	**88.03**	**92.98**	**83.92**	**88.94**	**86.92**	**89.36**	88.74	<u>87.25</u>	**88.07**

C.2 Supplementary to Generalization Analysis

Table 8 supplements the AUROC analysis in the main text with F1 results under distribution shifts, jointly characterizing ranking quality and source-prediction performance under the reported decision rule. Notably, F1 depends on a specific decision threshold, whereas AUROC measures overall ranking ability across thresholds. Consequently, when domain or generator shifts alter the distribution of detection scores, F1 may cause more pronounced F1 variation even when AUROC remains high. Under the *FD&AM* setting, each testbed fixes the text domain while incorporating outputs from multiple generators. **PEC-Detector** achieves the best F1 across all four domains, with an average of 88.94%, exceeding the strongest baseline, DetectAnyLLM, by 2.67 percentage points. Under the *AD&MS* setting, each testbed fixes the generator while spanning multiple text domains. Our method again obtains the highest average F1 of 88.07%, surpassing DetectAnyLLM by 0.39 percentage points. It achieves the best results on GLM-130B and GPT-3.5, ranks second on Bloom-7B, and remains below PRDetect on GPT-J. Although its average advantage is smaller under *AD&MS*, **PEC-Detector** achieves the best average F1 in both distribution shift settings, indicating that planning-execution coupling trajectories provide useful source-discriminative evidence under both domain and generator variation.

C.3 Evaluations on Advanced LLMs

To assess the effectiveness of the detector on newer generative models, we construct a supplementary evaluation set covering the WP, arXiv, and XSum domains in DetectRL Wu et al. (2024). We sample $1{,}600$ human-written text instances from each domain and generate machine-written counterparts with GPT-4-Turbo, Gemini-2.0-Flash, and Claude-3.7-Sonnet. Each model-specific evaluation set contains $6{,}400$ examples, enabling evaluation of detection performance on outputs from newer LLMs. As shown in Table 9, **PEC-Detector** achieves the best performance for all three generators, with average AUROC and F1 scores of 99.48% and 97.18%, respectively. Compared with DetectAnyLLM, the strongest baseline on average, our method improves AUROC and F1 by 1.50 and 0.48 percentage points, respectively, while attaining AUROC above 99.35 for GPT-4-Turbo, Gemini-2.0-Flash, and Claude-3.7-Sonnet. Although zero-shot methods such as Binoculars achieve competitive AUROC in some settings, their F1 scores vary more substantially across generators. In contrast, **PEC-Detector** maintains consistently strong performance across the three newer LLMs, suggesting it captures coupling signals beyond earlier generators.

Table 9: Comprehensive performance comparison of the newer LLMs, with the best and second-best results in each column highlighted in **bold** and underlined, respectively.

Method	GPT-4-Turbo		Gemini-2.0-Flash		Claude-3.7-Sonnet		Avg.	
	AUROC	F1	AUROC	F1	AUROC	F1	AUROC	F1
Zero-shot methods								
Fast-DetectGPT	43.25	67.52	23.23	66.74	39.10	67.67	35.19	67.31
DNA-GPT	39.00	65.66	23.58	3.14	30.68	67.12	31.09	45.31
Binoculars	95.96	90.32	97.73	95.68	97.74	92.56	97.14	92.85
Supervised methods								
RoBERTa-base	97.97	96.43	97.95	96.06	97.99	96.42	97.97	96.30
LF+Motifs	97.67	96.08	97.98	95.70	97.98	95.64	97.88	95.81
Raidar	83.51	76.92	81.56	75.68	88.81	80.83	84.63	77.81
BiScope	97.64	91.80	97.80	94.67	96.82	93.39	97.42	93.29
PRDetect	97.67	95.06	97.84	95.34	97.75	95.39	97.75	95.26
RepreGuard	77.25	80.27	67.17	72.87	73.58	74.05	72.67	75.73
DetectAnyLLM	97.99	96.58	97.95	96.84	97.99	96.69	97.98	96.70
ProSSD	85.39	79.01	80.42	80.18	79.85	77.53	81.89	78.91
PEC-Detector	**99.35**	**97.14**	**99.58**	**97.24**	**99.50**	**97.16**	**99.48**	**97.18**

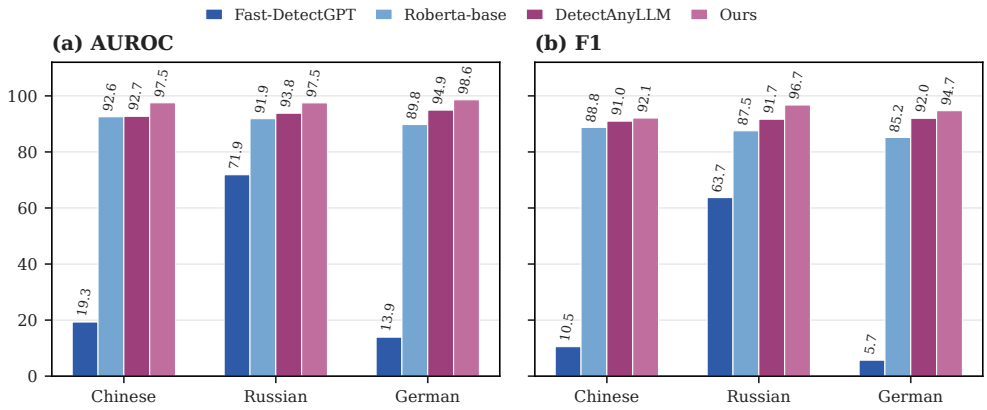


Figure 4: Performance of our method in multilingual settings.

C.4 Performance on Multilingual Datasets

Lexical, morphological, and syntactic realizations vary substantially across languages. We further assess performance beyond English on the Chinese, Russian, and German subsets of M4GT-Bench Wang et al. (2024b). For this evaluation, PEC-Detector uses the frozen `paraphrase-multilingual-mpnet-base-v2` sentence encoder, and each supervised detector is trained and selected on language-specific label-stratified splits; zero-shot methods are evaluated directly on the corresponding test sets. The contextual classification baseline uses `xlm-roberta-base`, while the multilingual implementations of Fast-DetectGPT and DetectAnyLLM use Qwen2.5-7B-Instruct. As shown in Figure 4, PEC-Detector achieves the highest AUROC in all three languages, averaging 97.89% and exceeding the second-best baseline by 3.68 to 4.83 percentage points. The results indicate that the observed advantage is not driven by any single language.

C.5 Encoder Robustness Analysis

We replace the sentence encoder in our method with the lightweight `all-mpnet-base-v2`, while keeping all other model and training configurations fixed. As shown in Table 10, the resulting variant retains strong AUROC and F1 across all four evaluation datasets. This consistency indicates that **PEC-Detector** is reasonably robust to the choice of sentence encoder: its effectiveness primarily

Table 10: Encoder robustness analysis on four evaluation datasets. We replace the sentence encoder in **PEC-Detector** with the lightweight `all-mpnet-base-v2`. GPT-3.5 refers to the dataset in the *AD&MS* setting of MAGE.

Method	CMV		GPT-3.5		ArXiv		Claude	
	AUROC	F1	AUROC	F1	AUROC	F1	AUROC	F1
PEC-Detector w/ `all-mpnet-base-v2`	96.36	89.41	95.47	88.21	97.93	95.69	98.29	97.68
PEC-Detector	**96.45**	**90.84**	**95.49**	**89.36**	**98.42**	**98.53**	**99.23**	**99.06**

derives from modeling planning-execution relations as a document unfolds, rather than relying on a particular large sentence encoder.