

Personalization over Generalization: History-Aware Sentence-Level AI Detection in Collaborative Writing

Rizky Adi^{1,2}, Yoshimi Suzuki¹, Fumiyo Fukumoto²

¹University of Yamanashi, Japan

²Meiji University, Japan

{g24tk03, ysuzuki}@yamanashi.ac.jp

fukumoto@meiji.ac.jp

Abstract

Sentence-level AI detectors are essential for detecting AI misuse in human-AI collaborative writing. Yet existing approaches still treat all authors identically, ignoring individual writing styles. This is a critical oversight, given that the boundary between human and AI text is inherently author-specific. To address this limitation, we propose **PHAS-ID** (Personalized History-Aware Sentence-level AI Detector), an approach that introduces personalization to sentence-level detection. By matching target sentences against the author’s history of labeled documents, PHAS-ID tailors its predictions based on prior writing behaviors. Furthermore, current synthetic datasets for this task fail to capture realistic collaborative patterns, which we address by introducing **Ghost-Fill**, a style-aware synthetic dataset generation pipeline. Our experiments demonstrate that PHAS-ID outperforms the previous state-of-the-art (SOTA) across seen-author, unseen-author, cross-domain, and cross-generator settings. Ablation studies confirm that these improvements stem from genuine author-specific personalization rather than the mere addition of context. Our source code and dataset are available at our GitHub repository¹.

1 Introduction

The rapid adoption of Large Language Models (LLMs) has introduced significant risks, including academic dishonesty and the spread of misinformation (Kwon and Jang, 2025; Pudasaini et al., 2025). While AI detectors mitigate these risks, traditional document-level methods (Gui et al., 2025; Li et al., 2025) fail on texts that contain a mixture of human and machine-generated content (Wang et al., 2023; Su et al., 2025). Consequently, sentence-level AI detection (SL-AID) that precisely identifies which portions of a document were AI-generated is essential for auditing mixed-authorship content and

¹<https://github.com/adirizq/phas-id>

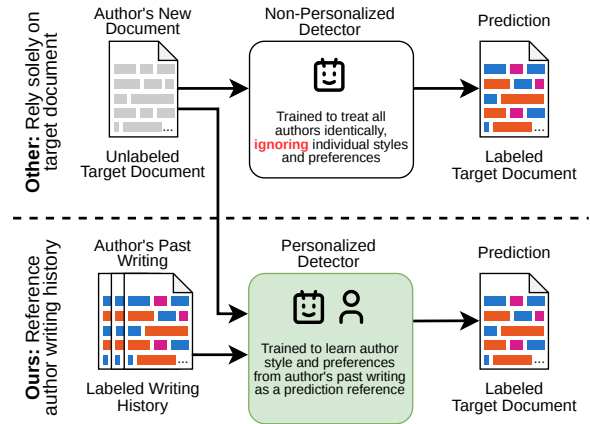


Figure 1: Comparison between generic AI detection and personalized AI detection (ours). Generic detectors rely solely on the target document, while personalized detectors leverage labeled writing history from the same author to make grounded predictions.

enforcing academic integrity in the era of human-AI collaboration (Lee et al., 2022).

Early SL-AID methods have attempted to solve this by framing the problem as a generic sequence classification (Wang et al., 2023) or isolated-sentence classification (Zeng et al., 2024; Nguyen-Son et al., 2024). Despite these efforts, existing detectors yield limited performance on real-world collaborative datasets. The strongest baseline achieves only a 61.72 macro F1 on the CoAuthor benchmark (Lee et al., 2022; Adi et al., 2025). We argue that this failure arises from two limitations. First, existing methods treat all authors identically (Top in Figure 1). In practice, however, human authors exhibit high variance in how they adopt AI suggestions, with some heavily editing outputs while others accept them verbatim. This means that the human-AI decision boundary is inherently author-specific, an observation consistent with Bevendörff et al. (2025), who theoretically argue that LLM detection is an instance of Authorship Verification. Second, efforts that rely on synthetic datasets

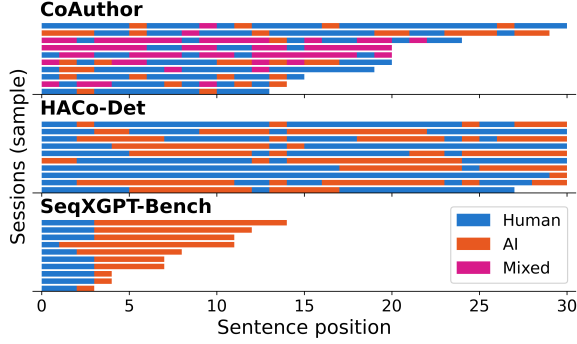


Figure 2: Label sequence patterns of CoAuthor (Lee et al., 2022), HCo-Det (Su et al., 2025), and SeqXGPT-Bench (Wang et al., 2023).

(Wang et al., 2023; Su et al., 2025; Jiang et al., 2025) fail to reflect these real-world writing dynamics. As shown in Figure 2, CoAuthor (Lee et al., 2022), a dataset of real human-AI collaboration features random short interspersed bursts of AI content (average run length of 1.4 sentences, transition variance $\sigma^2 = 6.6$), while synthetic datasets (SeqXGPT-Bench and HCo-Det) show long contiguous blocks, creating a structurally uniform and simpler detection task.

To address these limitations, we introduce **PHAS-ID** (Personalized History-Aware Sentence-level AI-Detection) and **GhostFill**. Inspired by Authorship Verification (Rivera-Soto et al., 2021; Kim et al., 2025), PHAS-ID leverages K labeled support documents from the target author as a reference for grounded predictions based on linguistic and stylistic patterns, such as vocabulary use, level of formality, and syntactic structure (Bottom in Figure 1). Although labeled support documents require prior annotation effort, we demonstrate that this data-collection requirement is practical in high-stakes environments, as PHAS-ID outperforms all baselines with as few as $K = 1$ support document. Furthermore, while real datasets like CoAuthor exist, they are limited and outdated (GPT-3.5 Turbo era). Our GhostFill pipeline addresses this necessity by incorporating an author’s extracted style to dynamically generate held-out test sets with realistic, randomized label sequences, mimicking real-world collaborative writing settings using current frontier LLMs. Our main contributions are as follows:

- (1) We propose PHAS-ID, a novel personalized SL-AID method that uses K labeled support documents from the same author as a reference for a personalized history-aware prediction.

- (2) We introduce GhostFill, a synthetic data pipeline that leverages authors’ writing styles to generate test sets that accurately mirror the complex label-sequence patterns of real-world collaborative writing.
- (3) We extensively evaluate our framework and demonstrate that PHAS-ID outperforms existing baselines across seen-author, unseen-author, cross-domain, and cross-generator settings. Furthermore, our ablation study confirms that the improvement in our model stems from personalization and is not due to additional context alone.

2 Related Work

Sentence-level AI Detection (SL-AID) The earliest attempt to solve the SL-AID task is SeqXGPT by Wang et al. (2023), which utilizes word log-probabilities from open-source LLMs as its primary features. A later dominant approach involved fine-tuning pre-trained language models (PLMs); specifically, a DeBERTaV3-based (He et al., 2023) classifier was reported as the strongest PLM-only detector in this line of work (Zeng et al., 2024; Su et al., 2025). Concurrently, regeneration-based methods emerged, exploiting the observation that LLMs tend to regenerate AI-generated text with higher similarity than human-written text under the same prompt (Zhu et al., 2023; Mao et al., 2024). This paradigm was extended to the sentence level by Nguyen-Son et al. (2024) in SimLLM. The most recent approach is to incorporate the surrounding context alongside local sentence features. GL-CLiC (Adi et al., 2025) builds on a DeBERTaV3 backbone and integrates linguistic features such as coherence and lexical complexity, achieving the current SOTA on the CoAuthor dataset (Lee et al., 2022). In contrast, SenDetEX (Jiang et al., 2025) combines token probability and entropy signals from a proxy LLM with sentence embeddings from a transformer encoder. Despite their differences, these methods share a common limitation: they were designed for a generic, one-size-fits-all detection scenario, which breaks down in personalized, real-world collaborative settings where the decision boundary between human and AI writing is inherently author-specific.

Authorship Verification The Authorship Verification (AV) task is to determine whether two or more documents were written by the same author, using the individual writing style as the primary

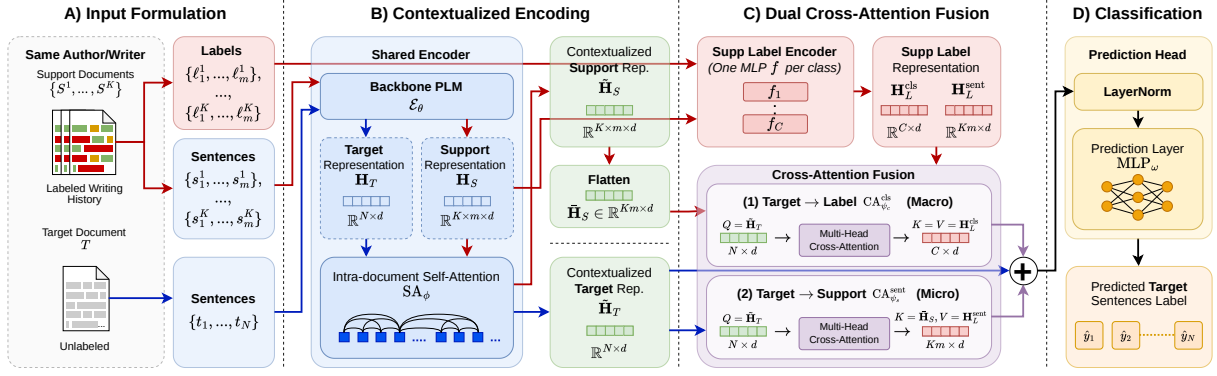


Figure 3: Overview of **PHAS-ID** framework, which consists of four steps: A) Input Formulation, B) Contextualized Encoding, C) Dual Cross-Attention Fusion, and D) Classification.

signal (Rivera-Soto et al., 2021). Bevendorff et al. (2025) theoretically argued that LLM-generated text detection is fundamentally an authorship verification problem, proposing that anomalies relative to an author’s typical writing style are the key signal for identifying generated text. However, their work lacks an empirical evaluation of this theory at the sentence level, particularly in complex human-AI collaborative writing. Although some AV approaches adopt few-shot or metric-learning formulations to compare document pairs (Rivera-Soto et al., 2021; Kim et al., 2025), these methods are optimized for binary classification of the same-author and assume full human-authored input, making them unsuitable for fine-grained sentence-level detection in mixed-authorship documents (Zeng et al., 2025; Nguyen et al., 2025). PHAS-ID operationalizes this insight by adopting the AV convention of K labeled support documents as an author-specific reference to the writing history, extending it for the first time to the SL-AID task on mixed-authorship documents.

Sentence-level AI Synthetic Dataset Early synthetic datasets for SL-AID, such as SeqXGPT-Bench (Wang et al., 2023), provided insufficient label sequence variability to reflect real collaborative writing. Subsequent efforts, such as HACO-Det (Su et al., 2025) and SenDetEX (Jiang et al., 2025), introduced iterative paraphrasing and sentence-infilling using modern LLMs. However, as illustrated in Figure 2, HACO-Det still produces label sequence patterns that diverge substantially from those of real-world collaborative writing. More critically, neither SeqXGPT-Bench, HACO-Det, nor SenDetEX incorporates author-specific style, meaning current synthetic datasets lack the personalized

nature that characterizes real human-AI collaborative writing. We bridge this gap with GhostFill, a pipeline that incorporates an extracted author-style profile to generate realistic, sparse-label sequence patterns that mirror the randomized structure of real-world collaborative writing using newer LLMs.

3 PHAS-ID Framework

Framework Overview PHAS-ID is a personalized approach that processes a target document alongside contextual information from the same author. As illustrated in Figure 3, the model accepts a single target document T and a distinct set of K support documents $\mathcal{S} = \{S^1, \dots, S^K\}$. To maintain personalization, T and all documents in $\mathcal{S}$ must be written by the same author. Furthermore, we strictly enforce $T \notin \mathcal{S}$ to prevent data leakage. We also make sure there is no test data that is used as support documents during training. To ensure the provided context is informative, we employ StyleDistance (Patel et al., 2025) to retrieve the top- K support documents from the author’s history that are most stylistically similar to T .

Deployment Assumption PHAS-ID assumes access to K labeled support documents from the target author at inference time. These labels may come from instrumented writing environments (e.g., editors with integrated AI assistants and LMS writing tools) that record sentence label at writing time, distinguishing accepted AI suggestions from user keystrokes. Historical labels are therefore metadata, not detector predictions, and introduce no circularity or dependencies on other AI detectors.

3.1 Input Formulation

As shown on the left side of Figure 3, we process both the target document T and the support set $\mathcal{S}$ at the sentence level. We represent the target document as a sequence of N sentences, $T = (t_1, \dots, t_N)$. Similarly, for each support document $S^k \in \mathcal{S}$ (where $k \in [1, K]$), we extract a sequence of up to m sentences $S^k = (s_1^k, \dots, s_m^k)$, where m denotes the maximum sentence limit per support document. Alongside the text, we extract the corresponding sequence of ground-truth sentence labels $L^k = (\ell_1^k, \dots, \ell_m^k)$. Each sentence label ℓ_i^k is drawn from a predefined label space $\mathcal{Y} = \{\text{Human}, \text{AI}, \text{Human-AI}\}$.

3.2 Contextualized Encoding

Backbone PLM As illustrated in the encoding module of Figure 3, the next step is to convert the textual information into dense vector representations. We employ DeBERTaV3 (He et al., 2023) as our backbone pre-trained language model (PLM), denoted as $\mathcal{E}_\theta$. We feed the tokenized sentences of the target document T and the support documents in $\mathcal{S}$ into $\mathcal{E}_\theta$. For each sentence, we extract its final hidden [CLS] token representation. This produces independent sentence representations $\mathbf{H}_T \in \mathbb{R}^{N \times d}$ for T , and $\mathbf{H}_S \in \mathbb{R}^{K \times m \times d}$ for $\mathcal{S}$, where d is the hidden dimension size.

Intra-doc Self-attention Because sentence representations in $\mathbf{H}_T$ and $\mathbf{H}_S$ are encoded independently, we introduce a shared intra-document self-attention module SA_ϕ , to allow sentences within the same document to attend to one another. We pass $\mathbf{H}_T$ and each support document representation $\mathbf{H}_S^k$ into shared self-attention SA_ϕ , applying standard attention masks to ignore padded sentences. The formulations are as follows:

$$\begin{aligned}\tilde{\mathbf{H}}_T &= \text{SA}_\phi(Q = \mathbf{H}_T, K = \mathbf{H}_T, V = \mathbf{H}_T), \\ \tilde{\mathbf{H}}_S^k &= \text{SA}_\phi(Q = \mathbf{H}_S^k, K = \mathbf{H}_S^k, V = \mathbf{H}_S^k),\end{aligned}$$

where $\tilde{\mathbf{H}}_T \in \mathbb{R}^{N \times d}$ denotes contextualized target representation and $\tilde{\mathbf{H}}_S^k \in \mathbb{R}^{m \times d}$ represent contextualized support representation. To prepare for cross-document fusion, we flatten the contextualized support documents by collapsing the document and sequence dimensions, resulting in a continuous memory bank $\tilde{\mathbf{H}}_S \in \mathbb{R}^{K \times m \times d}$.

3.3 Dual Cross-Attention Fusion

The contextualized target representations alone lack information regarding the author’s specific

writing style and historical preferences. We address this using a Dual Cross-Attention Fusion module (see Figure 3), which conditions the target document on the author’s labeled writing history at both the macro (label) and micro (sentence) levels.

Label Encoding Given the support representation $\mathbf{H}_S$ and its corresponding ground-truth sentence labels L , we first group the support sentences by their label classes to create class-specific representations. For each class $c \in \{1, \dots, C\}$, we compute the mean-pooled vector of all support sentences belonging to class c . We then pass this mean vector through a class-specific Multi-Layer Perceptron (MLP), denoted as f_c , to produce support label representations. This yields a macro-level label representation matrix $\mathbf{H}_L^{\text{cls}} \in \mathbb{R}^{C \times d}$. To enable micro-level sentence attention, we also broadcast these representations back to the sentence level, assigning each sentence in $\mathbf{H}_S$ its corresponding label representation, resulting in a mapped label matrix $\mathbf{H}_L^{\text{sent}} \in \mathbb{R}^{K \times m \times d}$.

Cross-Attention Fusion The core intuition of PHAS-ID is to ground predictions in the author’s history by implicitly asking two questions: (1) *At a macro level, which high-level label (e.g., Human vs. AI) does this target sentence most resemble?* and (2) *At a micro level, which specific historical sentences does this target sentence resemble, and what were their labels?*

We execute this via two parallel multi-head cross-attention (CA) mechanisms corresponding to pathways (1) and (2) in Figure 3. The first, $\text{CA}_{\psi_c}^{\text{cls}}$, attends to the macro-level label representation. The second, $\text{CA}_{\psi_s}^{\text{sent}}$, uses the historical support sentences as keys, but retrieves their corresponding label representation as values. Using the contextualized target document $\tilde{\mathbf{H}}_T$ as the query, we formulate this as:

$$\begin{aligned}\mathbf{Z}^{\text{cls}} &= \text{CA}_{\psi_c}^{\text{cls}}(Q = \tilde{\mathbf{H}}_T, K = \mathbf{H}_L^{\text{cls}}, V = \mathbf{H}_L^{\text{cls}}), \\ \mathbf{Z}^{\text{sent}} &= \text{CA}_{\psi_s}^{\text{sent}}(Q = \tilde{\mathbf{H}}_T, K = \tilde{\mathbf{H}}_S, V = \mathbf{H}_L^{\text{sent}}),\end{aligned}$$

where $\mathbf{Z}^{\text{cls}} \in \mathbb{R}^{N \times d}$ represents the macro-level stylistic context, capturing the target document’s alignment with the support label representation, and $\mathbf{Z}^{\text{sent}} \in \mathbb{R}^{N \times d}$ represents the micro-level stylistic context, acting as a soft associative memory that retrieves the label representations of specific, historically similar support sentences.

3.4 Classification

As depicted on the right side of Figure 3, the final module acts as a prediction head to map the target sentence representations $\{t_1, \dots, t_N\}$ to their predicted labels $\{\hat{y}_1, \dots, \hat{y}_N\}$. We formulate our classification head as:

$$\begin{aligned}\hat{\mathbf{H}}_T &= \text{LayerNorm}(\tilde{\mathbf{H}}_T + \mathbf{Z}^{\text{cls}} + \mathbf{Z}^{\text{sent}}), \\ \hat{Y} &= \text{MLP}_\omega(\hat{\mathbf{H}}_T),\end{aligned}$$

where $\hat{Y} = \{\hat{y}_1, \dots, \hat{y}_N\}$, with each $\hat{y}_i \in \mathcal{Y}$, and MLP_ω is a standard multi-layer perceptron. Finally, the entire framework is trained end-to-end to minimize a weighted Cross-Entropy Loss with label smoothing ($\epsilon = 0.05$) to mitigate overconfidence on noisy stylistic boundaries.

4 GhostFill Framework

GhostFill is a style-aware synthetic dataset generation pipeline designed to simulate realistic human-AI collaborative writing label sequence patterns. Formally, given a target author a who only has purely human-written text, the task is to generate a synthetic human-AI mixed document that preserves a ’s unique style while exhibiting realistic collaborative interleaving patterns. To achieve this, we leverage a secondary author b from an existing collaborative writing dataset whose writing style closely resembles that of a . We then use b ’s historical data as a source of authentic human-AI patterns. The framework consists of three stages:

Personalization Adaptation To simulate authentic interleaving patterns, we first select a target author a from the Blog Authorship Corpus (Schler et al., 2006) and use StyleDistance (Patel et al., 2025) to identify their most stylistically similar counterpart, author b , in the CoAuthor Dataset. We retrieve a ’s documents to serve as the base text. Next, we estimate the human-AI transition rate λ and mean run length $\bar{r}$ from b ’s collaborative documents (see Appendix A). Using these statistics, we stochastically delete sentence spans from a ’s text, creating realistic gaps for the AI to fill.

Generation To ensure the generated text matches the target author’s a style, we extract a style profile from a ’s history using the Author Writing Sheet method (Ashok Kumar et al., 2025), capturing Plot, Creativity, Development, and Language Use (details in Appendix A). We prompt an LLM with this profile to fill the deleted spans with stylistically

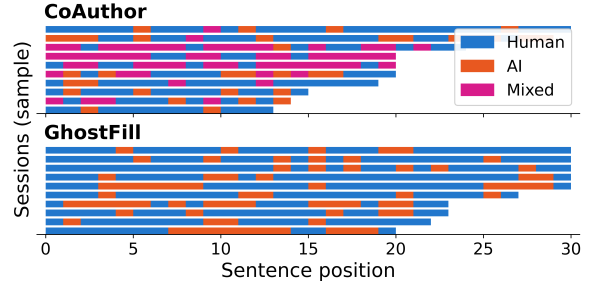


Figure 4: Label sequence patterns of CoAuthor (Lee et al., 2022) and our GhostFill dataset.

	CoAuthor	HACo-Det	GhostFill
Total Documents	1,445	11,200	276
Total Sentences	35,697	1,891,161	9,431
Human	23,183	1,313,983	7,047
AI	6,905	577,178	2,384
Human-AI	5,609	–	–
Unique Authors	63	–	10

Table 1: Statistics of the datasets.

consistent and locally coherent sentences. This produces a “Mixed Document” composed of human-written and AI-generated sentences.

Selection Because human writers actively curate AI suggestions, we generate V candidate variations for each mixed document. We then prompt the LLM to select the variation that best satisfies the style profile, effectively mimicking the editorial filtering inherent in collaborative writing.

5 Experiments

5.1 Experimental Setup

Dataset We evaluate PHAS-ID on three different SL-AID datasets. We use **CoAuthor** (Lee et al., 2022) as our primary dataset, as it is, to our knowledge, the only dataset created through real human-AI collaborative writing interaction. Furthermore, it consists of two distinct domains (creative and argumentative), making it suitable for cross-domain evaluation. The second dataset, **HACo-Det** (Su et al., 2025), is used to test PHAS-ID generalization capabilities on non-personalized settings. The third dataset is our **GhostFill**, which we use as a held-out test set specifically to evaluate our model’s capabilities on unseen LLMs (cross-generator setting). The statistics for all three datasets are summarized in Table 1, while a detailed breakdown of the train, validation, and test splits can be found in Table 2.

Split	Human	AI	Human-AI	Total
<i>CoAuthor</i>				
Train	15,969	4,855	3,490	24,314
Val	3,422	997	883	5,302
Test	3,792	1,053	764	5,609
<i>HACo-Det</i>				
Train	746,895	329,712	-	1,076,607
Val	380,837	165,624	-	546,461
Test	186,251	81,842	-	268,093
<i>GhostFill</i>				
Train	-	-	-	-
Val	-	-	-	-
Test	7,047	2,384	-	9,431

Table 2: Detailed statistics for the number of sentences in the train, validation, and test splits.

Evaluation Metrics We follow Wang et al. (2023); Adi et al. (2025) and report per-class precision, recall, and F1 score, along with Macro F1.

Human Evaluation We recruited three human annotators to evaluate the generation quality of GhostFill and check for any obvious AI artifacts. Each evaluated 15 randomly selected documents (5 documents from each LLM generator), totaling 527 sentences. For each document, the annotators were given two tasks: fluency scoring (1-5 Likert scale) and per-sentence human/AI label guessing.

Baselines For the non-personalized baselines², we include: **SeqXGPT** (Wang et al., 2023), **PLM Fine-tuning**, (specifically DeBERTaV3 (Zeng et al., 2024; Su et al., 2025)), **SimLLM** (Nguyen-Son et al., 2024), **PTD Framework** (Li et al., 2024), and **GL-CLiC** (Adi et al., 2025). For the personalized approach baseline, we perform few-shot LLM classification prompting using the author’s writing history as examples with both GPT-4o and Llama-4, the same LLMs selection as Adi et al. (2025) (details are provided in the Appendix B).

Implementation PHAS-ID is implemented using PyTorch and the Hugging Face Transformers library. We utilize OpenRouter to generate GhostFill, using GPT-5.4 (OpenAI, 2026), Sonnet 4.6 (Anthropic, 2026), and DeepSeekV4 Flash (DeepSeek-AI, 2026). All PHAS-ID experiments were conducted on a single NVIDIA RTX 4090 GPU with $K = 5$ configuration, a value determined empirically based on the analysis in §5.8. PHAS-ID consists of a total of 199 million parameters, of which

²We exclude SenDetEX (Jiang et al., 2025) from our comparison, as we cannot reproduce their work using their released code alone without making non-trivial assumptions.

183 million belong to the DeBERTa-V3-base (He et al., 2023) backbone. Training takes approximately three minutes per epoch per 1,000 target documents. We also provide comprehensive details on hyperparameters, optimizers, prompts, training, and generation procedures in Appendix C.

5.2 GhostFill Dataset

Generation Result As depicted in Table 1, we obtained the GhostFill dataset consisting of 276 mixed-authorship documents totaling 9,431 sentences. As we can see from Figure 4, GhostFill successfully mirrors the realistic collaborative writing label sequence patterns found in the CoAuthor dataset (Lee et al., 2022). Statistically, GhostFill achieves an average AI sentence run length of 1.57, which is highly comparable to the 1.38 run length observed in CoAuthor. Regarding transition variance, CoAuthor exhibits a higher variance ($\sigma^2 = 6.6$) than GhostFill ($\sigma^2 = 4.4$). This discrepancy is anticipated because our pipeline stochastically models the gaps using mean parameters (λ and $\bar{r}$). It successfully captures average run lengths while naturally smoothing out the highly erratic, extreme transition patterns found in unconstrained collaborative writing sessions.

Human Evaluation Result Fluency score averaged 3.09 (SD 0.97), with no detectable correlation between a document’s AI-sentence ratio and its fluency rating (Spearman $\rho = +0.11$, permutation $p = 0.68$), indicating that average fluency score comes from the underlying source document rather than the infilling mechanism. On the label guessing task, annotators reached only 0.37 AI-class F1, with near-zero agreement between annotators on which sentences were AI (Fleiss’ $\kappa = 0.10$). GhostFill therefore leaves no consistent human-detectable artifact.

5.3 Main Results

Seen Authors Setting As shown at the top of Table 3, PHAS-ID outperforms the previous SOTA, GL-CLiC, by a robust margin of 2.72 Macro F1 points. Crucially, PHAS-ID achieves its most substantial gains on the challenging minority classes, showing relative F1 improvements of 9.44% (AI) and 6.98% (Human-AI). Additionally, the result shows that few-shot prompting baselines (P-LLM) remain substantially below fine-tuned methods in all classes, confirming that personalization prompting alone is insufficient for this task.

Method	Human			AI			Human-AI			Macro F1
	P	R	F1	P	R	F1	P	R	F1	
Seen Authors Setting										
SeqXGPT	70.80	95.97	81.49	<u>58.60</u>	11.97	19.88	26.38	8.77	13.16	38.18
PLM Fine-tune	78.70	86.60	82.46	58.68	41.41	48.55	45.74	41.49	43.51	58.17
SimLLM	80.03	80.01	80.02	45.00	<u>52.61</u>	48.51	48.89	37.57	42.49	57.01
PTD	73.09	<u>90.95</u>	81.05	50.57	12.73	20.33	45.44	37.17	40.89	47.42
GL-CLiC	81.93	82.59	<u>82.26</u>	54.08	49.10	<u>51.47</u>	49.40	53.66	<u>51.44</u>	<u>61.72</u>
P-LLM (GPT-4o)	75.55	72.71	74.10	30.23	36.37	33.02	46.46	42.14	44.20	50.44
P-LLM (Llama 4)	75.17	60.36	66.96	30.01	39.41	34.07	30.06	46.47	36.50	45.85
PHAS-ID (Ours)	84.58*	79.56	81.95	50.44	64.12*	56.33*	57.85*	<u>52.54</u>	55.03*	64.44*
<i>p</i> -value	< 0.01	0.11	0.62	0.04	< 0.01	< 0.01	< 0.01	0.09	< 0.01	< 0.01
Unseen Authors Setting										
SeqXGPT	66.87	98.51	79.67	83.16	8.01	14.62	26.62	4.05	7.03	33.77
PLM Fine-tune	<u>80.22</u>	76.44	78.28	41.38	64.30	50.36	54.69	<u>34.58</u>	<u>42.37</u>	<u>57.00</u>
SimLLM	77.68	77.68	77.68	44.55	<u>61.36</u>	<u>51.62</u>	47.66	30.14	36.92	55.41
PTD	66.93	94.81	78.46	39.39	3.96	7.19	51.53	18.28	26.99	37.55
GL-CLiC	71.49	<u>95.86</u>	<u>81.90</u>	<u>60.85</u>	37.83	46.65	63.20	7.81	13.90	47.48
P-LLM (GPT-4o)	78.45	74.76	76.56	25.39	31.10	27.96	30.59	30.20	30.39	44.97
P-LLM (Llama 4)	78.22	63.14	69.88	23.60	29.67	26.29	22.39	40.69	28.89	41.68
PHAS-ID (Ours)	85.87*	86.38	86.11*	52.39	57.92	54.86*	<u>54.75</u>	43.94*	48.33*	63.10*
<i>p</i> -value	< 0.01	< 0.01	< 0.01	< 0.01	0.01	< 0.01	0.98	0.02	< 0.01	< 0.01

Table 3: SL-AID results on the CoAuthor Dataset (**seen authors** and **unseen** setting). **Bold** refers to the best performance, underline marks the second best, and (*) indicates performance consistently above second-best baseline ($p < 0.05$). All PHAS-ID results are reported as the average of 5 runs with different seeds using $K = 5$ support documents. P-LLM means Personalized LLM. All non-personalized baselines are reported by [Adi et al. \(2025\)](#). Since all non-personalized baselines are reported from prior work without per-run variance, we apply a one-sample two-tailed t-test, treating each reported value as a fixed reference point. All notations, aggregation, and abbreviations are the same for subsequent result tables unless stated otherwise.

Unseen Authors Setting This setting tests generalization to new authors that were not seen during training. Here, the gap between personalized and non-personalized approaches becomes stark (Bottom of Table 3). While GL-CLiC suffers a severe performance drop of 14.24 Macro F1 points compared to the seen setting, both PLM Fine-tune and our proposed PHAS-ID demonstrate remarkable stability, declining by only 1.14 and 1.34 points, respectively. This stability suggests that standard PLM Fine-tune and PHAS-ID rely on generic textual features rather than memorizing author identities, while GL-CLiC overfits to the author style seen during training. Crucially, our PHAS-ID personalized approach outperforms the baselines, achieving a 6.10 Macro F1 increase over PLM Fine-tune.

5.4 Cross-Domain Capabilities

Table 4 evaluates cross-domain transfer between the Creative and Argumentative domains. PHAS-ID achieves the highest Macro F1 in both directions. In the *Creative* $\rightarrow$ *Argumentative* setting, it outperforms the second-best by 2.92 points. While the re-

Method	F1			Macro F1
	H	AI	H-AI	
<i>Creative</i> $\rightarrow$ <i>Argumentative</i>				
SeqXGPT	78.72	2.17	14.53	31.81
PLM Fine-tune	78.79	43.26	42.11	54.72
SimLLM	76.71	42.38	32.41	50.50
PTD	77.23	0.75	23.98	33.99
GL-CLiC	79.51	46.31	57.37	61.06
PHAS-ID	80.27	52.20*	59.47*	63.98*
<i>p</i> -value	0.26	<0.01	0.03	<0.01
<i>Argumentative</i> $\rightarrow$ <i>Creative</i>				
SeqXGPT	81.54	9.48	4.32	31.78
PLM Fine-tune	73.67	42.62	30.60	48.96
SimLLM	69.77	39.36	26.69	45.27
PTD	77.58	0	27.15	34.91
GL-CLiC	72.35	43.77	33.47	49.86
PHAS-ID	<u>78.63</u>	45.81*	44.76*	56.40*
<i>p</i> -value	<0.01	0.01	<0.01	<0.01

Table 4: Cross-domain experimental results on the CoAuthor Dataset, training on one domain and evaluating on another.

Method	Human F1	AI F1	Macro F1
PLM Fine-tune	97.9	95.2	96.6
SeqXGPT	85.4	49.4	67.4
DetectGPT	79.4	12.4	45.9
NPR	50.6	44.0	47.3
Fast-DetectGPT	67.8	38.8	53.3
PHAS-ID (Ours)	<u>88.49</u>	<u>75.98</u>	<u>82.24</u>

Table 5: Non-personalized setting results on HACo-Det. Support documents for PHAS-ID were selected from the entire dataset pool, as there is no author metadata. All baselines are reported in Su et al. (2025).

Method	F1		Micro F1	Macro F1
	H	AI		
PLM Fine-tune	80.13	<u>16.54</u>	67.90	48.30
GL-CLiC	83.66	6.08	72.17	44.87
PHAS-ID	<u>80.17</u>	30.38*	<u>69.18</u>	55.28*
<i>p</i> -value	0.96	<0.01	0.23	<0.01

Table 6: Cross-generator experimental results. Training on the CoAuthor dataset and evaluating on the GhostFill dataset.

verse direction (*Argumentative* $\rightarrow$ *Creative*) yields lower overall scores across all models, likely due to the higher lexical and stylistic diversity inherent in creative writing. In other words, the model sees unseen words and lexicals during inference due to limited examples in the training set. Despite the more challenging transfer setting, PHAS-ID maintains a substantial 6.54 point advantage over the second-best method.

5.5 Non-Personalized Setting

Table 5 reports results of HACo-Det, which lacks author metadata. Here, PHAS-ID must randomly sample support documents from the general pool rather than matching them to specific authors. Without its primary personalization signal, PHAS-ID’s performance drops, trailing the PLM Fine-tune baseline by 14.36 Macro F1 points. This performance degradation directly validates our core hypothesis that matching author history is critical for history-aware detection; this finding is strengthened by the personalization ablation experiment in §5.7. Nevertheless, even with mismatched context, PHAS-ID still outperforms all other zero-shot and sequence-based baselines (e.g., +14.84 Macro F1 over SeqXGPT), demonstrating robust fallback capabilities when metadata is entirely absent.

Method	F1			Macro F1
	H	AI	H-AI	
Non-Personalized				
No Supp	80.37	51.30	46.92	59.53
Global Supp	80.46	50.59	48.29	59.78
Topic Supp	79.67	51.16	48.92	59.92
Personalized				
Human Only	82.41	53.67	53.70	63.26
Full	81.95	56.33*	55.03*	64.44*
Non-Personalized vs. Personalized				
p-value	0.08	<0.01	<0.01	<0.01

Table 7: Results of the personalization ablation study. No Supp means $K = 0$; Global Supp denotes the support documents sampled from the entire dataset; Topic Supp means support documents sampled from different random authors on the same topic; Human Only denotes full human written only support documents from the same author, and Full denotes the standard PHAS-ID. The significance test was conducted using a paired t-test between the best personalized and best non-personalized method.

5.6 Cross-Generator Capabilities

To test cross-generator robustness, we trained models on CoAuthor (GPT-3.5) and evaluated them on GhostFill (GPT-5.4, Sonnet 4.6, DeepSeekV4). We compare against PLM Fine-tune and GL-CLiC, the two strongest non-personalized baselines. As shown in Table 6, all methods suffer severe performance drops on the AI class, highlighting the immense difficulty of cross-generator generalization. While the absolute Macro F1 remains modest at 55.28 points, PHAS-ID demonstrates a consistent improvement over the best non-personalized baseline. Notably, PHAS-ID improves AI-class F1 by 13.84 absolute points over PLM Fine-tune. This suggests that on the GhostFill dataset, author-specific stylistic cues provide a more reliable, generator-agnostic signal than surface-level artifacts, even when transitioning to more advanced frontier models.

5.7 Personalization Ablation

Table 7 isolates the impact of author-matched context. Moving from zero support documents (No Supp) to the personalized configuration (Full) yields significant gains on the AI (+5.03 F1) and Human-AI (+9.11 F1) classes. Three controls attribute these gains to author identity personalization rather than additional information alone. Replacing author-matched documents with randomly sampled ones (Global Supp) collapses performance to

Total Support	F1			Macro F1
	H	AI	H-AI	
$K = 0$	80.37	51.30	46.92	59.53
$K = 1$	81.43	53.75	54.32	63.17
$K = 2$	83.13	54.22	55.25	<u>64.20</u>
$K = 3$	82.05	55.18	55.20	64.14
$K = 4$	<u>82.27</u>	<u>55.64</u>	54.59	64.16
$K = 5$	81.95	56.33	55.03	64.44

Table 8: Experimental results of different K value on PHAS-ID.

near-baseline levels; since the global pool already carries the corpus label prior, label-distribution information alone cannot explain the improvement. Support drawn from the same topic but a different author also performs at the no-support level, ruling out topical overlap. Finally, support consisting solely of the author’s fully human-written documents retains most of the personalization benefit despite removing the per-sentence labeling burden entirely, and reaches 63.26 Macro F1, outperforming the best of non-personalized at 59.92 Macro F1.

5.8 K -Support Document Analysis

Table 8 shows the impact of varying the number of support documents. The most substantial improvement occurs immediately upon introducing a single document, yielding a +3.64 point jump from $K = 0$ to $K = 1$. Macro F1 continues to rise sharply to 64.20 points at $K = 2$, after which overall performance plateaus with marginal fluctuations before reaching its peak at $K = 5$. Notably, the benefits are class-dependent: while Human and Human-AI F1 scores peak early around $K = 2$, AI detection continues to scale with additional context up to $K = 5$. We hypothesize this is because Human and Human-AI sentences are grounded in the author’s own voice, which even a single support document characterizes well, whereas AI sentences carry no author-specific anchor and require a wider support pool for the micro-level cross-attention to cover their stylistic range. Practically, since $K = 1$ already exceeds the strongest non-personalized baseline (§5.3), allowing deployments to trade a small amount of Macro F1 for substantially reduced labeling effort.

5.9 Error Analysis

The model misclassified 1,454 of 5,609 test sentences (25.92% error rate), with errors predominantly occurring at authorship boundaries, a failure

(Human)	You can use the time to sit in the bathtub and just listen.
(Human)	That can be very relaxing.
(AI)	It also allows you more time to hear the story especially when driving.
(Human)	Imagine going on a cross country trip and bringing audiobooks with you.
(Human)	You could easily get through multiple books even while driving.

Figure 5: Example of "sandwich" error, where a single AI sentence is surrounded by human sentences, causing the AI sentence to be easily misclassified as human.

mode we call the "sandwich" error. As illustrated in Figure 5, a "sandwich" error occurs when a target sentence is surrounded on both sides by sentences sharing a different label, and the model predicts that surrounding label rather than the target’s own label. Sandwich errors account for 28.89% of misclassifications and disproportionately affect AI-generated text, which constitutes 55.58% of all AI misclassifications. We hypothesize that this error is caused by context over-smoothing, where the intra-document attention layer globally blends sentence representations, overwhelming minority-label embeddings. Future work could theoretically resolve this issue by incorporating boundary-aware modeling, such as local attention windows, auxiliary boundary-task prediction, or a lightweight CRF decoding head to learn explicit label transitions.

6 Conclusion

In this work, we proposed **PHAS-ID**, a personalized, history-aware sentence-level AI detector. We also introduced **GhostFill**, a style-aware synthetic test set generation pipeline that produces label sequence patterns consistent with real-world, human-AI collaborative writing. Extensive experiments demonstrate that PHAS-ID consistently outperforms all baselines across seen-author, unseen-author, cross-domain, and cross-generator settings, with ablation studies confirming that the improvements come from genuine author-specific personalization rather than the mere presence of additional context alone. PHAS-ID’s advantage is bounded by the availability of author writing history: with mismatched or absent metadata, it remains competitive but no longer leads, and metadata-rich collaborative platforms are its intended deployment setting. The error analysis further reveals a sandwich issue as the primary failure mode, in which isolated sentences surrounded by neighbors of a different class are disproportionately misclassified, which will be the direction for future work.

Limitations

Author Style and Preference Changes Our work does not account for scenarios where a target author’s writing style or preferences evolve over time. In real-world settings, individual writing habits are subject to temporal drift driven by shifting professional environments, social contexts, or natural vocabulary evolution. Since our model relies heavily on historical labeled data, its longitudinal robustness remains untested. Specifically, if an author’s style and preferences in the support documents have evolved significantly relative to the target document, the model’s predictions may degrade.

Data Scarcity of Gold Standard Corpora A key constraint of personalized sentence-level AI detection is the reliance on historical writing data. Although our empirical analysis demonstrates that PHAS-ID is highly data-efficient, achieving strong performance with as few as $K = 2$ support documents, the overall scarcity of large-scale, high-quality human-AI collaborative benchmarks remains a bottleneck. The CoAuthor dataset represents a specific era of LLMs (GPT-3.5 Turbo), and our GhostFill pipeline is constrained by the high financial cost of API generation from modern frontier models. Consequently, the upper bound of our model’s performance under massive, multi-domain training data remains unexplored, and performance may scale further as more extensive gold standard collaborative datasets become available.

Vulnerability to Adversarial Attacks We have not evaluated our model against sophisticated adversarial manipulation, such as targeted paraphrase attacks or prompt-engineered style obfuscation. Prior work indicates that surface-level text modifications can severely degrade the accuracy of standard AI detectors (Zhou et al., 2024; Wang et al., 2024). While we hypothesize that PHAS-ID’s reliance on macro and micro stylistic context provides better generator-agnostic stability, its resilience against explicit adversarial attempts to mimic or mask an author’s unique profile has yet to be empirically verified.

Donor-Based Interleaving Statistics GhostFill transplants transition statistics from a stylistically similar donor author, assuming that similar styles imply similar collaboration dynamics. No ground-truth interleaving statistics exist for the target author, so this remains an approximation. Ghost-

Fill synthesizes realistic interleaving structure in aggregate rather than replicating any individual’s true human-AI mixing behavior or personal style preferences. When no annotated donor corpus is available, GhostFill falls back to randomly drawn parameters fixed per author for reproducibility, further weakening individual fidelity.

Ethical considerations

We recognize that AI text detection is a high-stakes domain. Mistakes, especially false positives, can have severe consequences in educational, legal, and professional settings. Given that the current iteration of PHAS-ID still exhibits limitations in its detection performance, we strongly advise against deploying this system in any real-world decision-making applications. It is released strictly as a proof-of-concept research tool intended to advance the academic understanding of AI detection.

Acknowledgments

We would like to thank reviewers and metareviewer for their helpful comments and suggestions. We are also grateful to the annotators who helped during the evaluation. Special thanks to Bassamtiano Renaufalgi Irnawan and Ammar Qayyum for their insights. This work is supported by JSPS KAKENHI (No. 24K15085) and the JKA Foundation (No. 2024M-557, 2026M-375). The first author is supported by the MEXT scholarship (Grant No. 244183).

References

- Rizky Adi, Bassamtiano Renaufalgi Irnawan, Yoshimi Suzuki, and Fumiyo Fukumoto. 2025. [GL-CLiC: Global-local coherence and lexical complexity for sentence-level AI-generated text detection](#). In *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, pages 3600–3617, Mumbai, India. The Asian Federation of Natural Language Processing and The Association for Computational Linguistics.
- Anthropic. 2026. [Claude sonnet 4.6 system card](#).
- Nischal Ashok Kumar, Chau Minh Pham, Mohit Iyyer, and Andrew Lan. 2025. [Whose story is it? personalizing story generation by inferring author styles](#). In *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, pages 1485–

- 1540, Mumbai, India. The Asian Federation of Natural Language Processing and The Association for Computational Linguistics.
- Janek Bevendorff, Matti Wiegmann, Emmelie Richter, Martin Potthast, and Benno Stein. 2025. [The two paradigms of LLM detection: Authorship attribution vs. authorship verification](#). In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 3762–3787, Vienna, Austria. Association for Computational Linguistics.
- DeepSeek-AI. 2026. [Deepseek-v4: Towards highly efficient million-token context intelligence](#).
- Jiayi Gui, Baitong Cui, Xiaolian Guo, Ke Yu, and Xiaofei Wu. 2025. [AIDER: a robust and topic-independent framework for detecting AI-generated text](#). In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 9299–9310, Abu Dhabi, UAE. Association for Computational Linguistics.
- Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023. [DeBERTav3: Improving deBERTa using ELECTRA-style pre-training with gradient-disentangled embedding sharing](#). In *The Eleventh International Conference on Learning Representations*.
- Lei Jiang, Desheng Wu, and Xiaolong Zheng. 2025. [SenDetEX: Sentence-level AI-generated text detection for human-AI hybrid content via style and context fusion](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 5287–5302, Suzhou, China. Association for Computational Linguistics.
- Junghwan Kim, Haotian Zhang, and David Jurgens. 2025. [Leveraging multilingual training for authorship representation: Enhancing generalization across languages and domains](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 34867–34892, Suzhou, China. Association for Computational Linguistics.
- Soonchan Kwon and Beakcheol Jang. 2025. [A comprehensive survey of fake text detection on misinformation and lm-generated texts](#). *IEEE Access*, 13:25301–25324.
- Mina Lee, Percy Liang, and Qian Yang. 2022. [Coauthor: Designing a human-ai collaborative writing dataset for exploring language model capabilities](#). In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, CHI ’22, New York, NY, USA. Association for Computing Machinery.
- Yafu Li, Zhilin Wang, Leyang Cui, Wei Bi, Shuming Shi, and Yue Zhang. 2024. [Spotting AI’s touch: Identifying LLM-paraphrased spans in text](#). In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 7088–7107, Bangkok, Thailand. Association for Computational Linguistics.
- Yuanfan Li, Zhaohan Zhang, Chengzhengxu Li, Chao Shen, and Xiaoming Liu. 2025. [Iron sharpens iron: Defending against attacks in machine-generated text detection with adversarial training](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3091–3113, Vienna, Austria. Association for Computational Linguistics.
- Ilya Loshchilov and Frank Hutter. 2019. [Decoupled weight decay regularization](#). In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.
- Chengzhi Mao, Carl Vondrick, Hao Wang, and Junfeng Yang. 2024. [Raidar: generative AI detection via rewriting](#). In *The Twelfth International Conference on Learning Representations*.
- Meta. 2025. [The llama 4 herd: The beginning of a new era of natively multimodal ai innovation](#).
- Tuc Nguyen, Yifan Hu, and Thai Le. 2025. [Unraveling interwoven roles of large language models in authorship privacy: Obfuscation, mimicking, and verification](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 14903–14919, Suzhou, China. Association for Computational Linguistics.
- Hoang-Quoc Nguyen-Son, Minh-Son Dao, and Koji Zettsu. 2024. [SimLLM: Detecting sentences generated by large language models using similarity between the generation and its re-generation](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 22340–22352, Miami, Florida, USA. Association for Computational Linguistics.
- OpenAI. 2026. [Gpt-5.4 thinking system card](#).
- OpenAI, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Madry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, Alex Nichol, and 400 others. 2024. [Gpt-4o system card](#). *Preprint*, arXiv:2410.21276.
- Ajay Patel, Jiacheng Zhu, Justin Qiu, Zachary Horvitz, Marianna Apidianaki, Kathleen McKeown, and Chris Callison-Burch. 2025. [StyleDistance: Stronger content-independent style embeddings with synthetic parallel examples](#). 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 8662–8685, Albuquerque, New Mexico. Association for Computational Linguistics.
- Shushanta Pudasaini, Luis Miralles, David Lillis, and Marisa Llorens Salvador. 2025. [Benchmarking AI text detection: Assessing detectors against new datasets, evasion tactics, and enhanced LLMs](#). In *Proceedings of the 1st Workshop on GenAI Content Detection (GenAIDetect)*, pages 68–77, Abu Dhabi,

- UAE. International Conference on Computational Linguistics.
- Rafael A. Rivera-Soto, Olivia Elizabeth Miano, Juanita Ordonez, Barry Y. Chen, Aleem Khan, Marcus Bishop, and Nicholas Andrews. 2021. [Learning universal authorship representations](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 913–919, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Jonathan Schler, Moshe Koppel, Shlomo Argamon, and James W Pennebaker. 2006. [Effects of age and gender on blogging](#). In *AAAI spring symposium: Computational approaches to analyzing weblogs*, volume 6, pages 199–205.
- Zhixiong Su, Yichen Wang, Herun Wan, Zhaohan Zhang, and Minnan Luo. 2025. [HACo-det: A study towards fine-grained machine-generated text detection under human-AI coauthoring](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 22015–22036, Vienna, Austria. Association for Computational Linguistics.
- James Liyuan Wang, Ran Li, Junfeng Yang, and Chengzhi Mao. 2024. [RAFT: Realistic attacks to fool text detectors](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 16923–16936, Miami, Florida, USA. 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, Singapore. Association for Computational Linguistics.
- Peter Zeng, Pegah Alipoormolabashi, Jihu Mun, Gourab Dey, Nikita Soni, Niranjana Balasubramanian, Owen Rambow, and H. Schwartz. 2025. [Residualized similarity for faithfully explainable authorship verification](#). In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 15824–15837, Suzhou, China. Association for Computational Linguistics.
- Zijie Zeng, Shiqi Liu, Lele Sha, Zhuang Li, Kaixun Yang, Sannyuya Liu, Dragan Gasevic, and Guangliang Chen. 2024. [Detecting ai-generated sentences in human-ai collaborative hybrid texts: Challenges, strategies, and insights](#). In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, pages 7545–7553. International Joint Conferences on Artificial Intelligence Organization. AI for Good.
- 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 (LREC-COLING 2024)*, pages 8427–8437, Torino, Italia. 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, Singapore. Association for Computational Linguistics.

Appendix

A GhostFill Implementation Details

A.1 Dataset Filtering and Author Matching

To ensure the synthetic documents structurally resemble genuine collaborative writing sessions, we first align the length distribution of the Blog Authorship Corpus (Schler et al., 2006) with the CoAuthor dataset (Lee et al., 2022). We filter the source documents to match the 25th percentile to the maximum word count observed in CoAuthor, retaining only authors with a minimum of 10 valid documents. To pair a target author a with a stylistically similar CoAuthor counterpart b , we extract up to 100 sentences per author and encode them using StyleDistance model (Patel et al., 2025). We then compute the mean embedding for each author and pair them by maximizing cosine similarity.

A.2 Human-AI Transition and Run Length

Transition (λ) To accurately simulate realistic human-AI interleaving, we extract transition and length statistics from the historical sessions of author b . While the CoAuthor dataset contains three sentence-level labels (Human, AI, and Human-AI), our extraction pipeline isolates clear transition boundaries by first filtering the collaborative document to retain only strictly Human and strictly AI sentences. Let the filtered collaborative document D from author b consist of a sequence of n sentences with corresponding labels $L = (\ell_1, \dots, \ell_n)$, where $\ell_i \in \{\text{Human}, \text{AI}\}$. We define a transition as occurring when adjacent sentences in this filtered sequence have different authorship labels (i.e., $\ell_i \neq \ell_{i+1}$). The transition rate λ_D for a single document is computed as the total number of transitions normalized by the document length:

$$\lambda_D = \frac{1}{n} \sum_{i=1}^{n-1} \mathbb{I}(\ell_i \neq \ell_{i+1}).$$

The overall author transition rate λ is computed as the mean of λ_D across all historical documents written by b .

Run Length (r) Additionally, we model the contiguous chunks of generated text. An AI run is defined as a maximal contiguous subsequence of strictly AI-generated sentences. Let $R = \{r_1, r_2, \dots, r_m\}$ represent the set of lengths of all such AI runs extracted across b 's filtered collaborative history. The mean AI run length $\bar{r}$ is

then formalized as:

$$\bar{r} = \frac{1}{m} \sum_{j=1}^m r_j.$$

We retain the empirical pool of run lengths R for stochastic sampling, and we apply identical logic to compute human run lengths. Finally, we calculate the mean AI sentence ratio for author b .

A.3 Stochastic Span Deletion

Given a purely human-written base document from target author a consisting of n sentences, we generate a binary mask to determine which spans will be replaced by the AI. We determine a target number of AI sentences by sampling a ratio from a Gaussian distribution parameterized by b 's historical mean and standard deviation (bounded strictly between 0.20 and 0.60). We then iteratively sample a target run length r from b 's empirical pool R . For each sampled r , we select a random starting index i and mask the span of sentences from i to $i+r-1$. This process repeats until the target AI sentence count is reached.

A.4 Stylistic Profiling and Generation

To ensure the LLM generates text that conforms to author a 's style, we implement a multi-step extraction and generation pipeline. Our approach builds directly upon the "Author Writing Sheet" methodology introduced by Ashok Kumar et al. (2025). Although their original framework was designed for broader stylistic analysis and generation, we adapt the profiling and contextualization steps specifically to support the construction of a synthetic, sentence-level AI detection dataset. By modifying the prompt constraints to generate highly localized, document-specific fill rules, we force the LLM to seamlessly interleave generated sentences while strictly preserving the surrounding human-written context. The adapted pipeline proceeds as follows:

Base Post Generation For up to 5 profiling documents in a 's history, we extract the opening three sentences and prompt an LLM (Figure 6) to generate a generic "base post" of equivalent length.

Delta Sheet Extraction We contrast the first 500 words of a 's original post against this generic baseline (Figure 7) to extract distinctive stylistic claims and verbatim evidence across four dimensions: Plot, Creativity, Development, and Language Use.

Incremental Sheet Combination We sequentially merge the delta sheets extracted from the profiling documents into a single, cohesive Author Writing Sheet (Figure 8).

Persona Profile Generation We synthesize this combined Author Writing Sheet into a cohesive, second-person narrative persona profile (Figure 9).

Contextual Fill Rules For each specific document undergoing generation, we provide the LLM with the persona profile and the masked text to generate concrete, per-document filling rules (Figure 10).

Best-of- P Generation and Selection We prompt the LLM to fill the deleted spans $V = 3$ times (Figure 11). We utilize sequence matching (with an 85% similarity threshold) to strictly map the LLM’s output back to the unmasked human originals, discarding any generation that fails to preserve at least 90% of the original text. Finally, we employ an LLM selection prompt (Figure 12) to evaluate the successful variations and select the version that best reflects the author’s style.

B Baselines

SeqXGPT (Wang et al., 2023) a feature-based framework that utilizes proxy LLM log-probabilities as input to a Transformer classifier, they model the log-probabilities inputs just like *waves* in speech processing.

PLM Fine-tuning (Zeng et al., 2024; Su et al., 2025) a standard PLM fine-tuning technique without additional architectural changes. We utilize DeBERTaV3-base as the base PLM following its reported success on this task (Zeng et al., 2024; Su et al., 2025).

SimLLM (Nguyen-Son et al., 2024) is a new approach to exploiting LLMs’ tendencies to re-generate a similar text. This method fine-tunes a PLM on a target sentence concatenated with its AI-"proofread" versions.

PTD Framework (Li et al., 2024) a method that aims to detect AI paraphrased text spans through sentence score regression. Their method relies on fine-tuning PLM to do the token classification task.

Personalized LLM (P-LLM) We use two LLMs in our experiments: the closed-source GPT-4o (OpenAI et al., 2024) and the open-source Llama 4 Maverick (Meta, 2025) following Adi et al.

Parameter	Value
Temperature	0.0
Top P	1.0
Top K	0.0
Frequency Penalty	0.0
Presence Penalty	0.0
Repetition Penalty	0.0
Min P	0.0
Top A	0.0

Table 9: LLM generation parameters. All parameters except Temperature are the OpenRouter default value.

Parameter	Value
<i>Input</i>	
Token Max Length	512
Batch Size	1
K	{1, ..., 5}
<i>Learning Rate</i>	
η_{plm}	1e-5
η_{mlp}	5e-5
<i>Intra-Document Attention</i>	
nhead	8
dim_feedforward	hidden_size * 4
num_layers	1
<i>Cross-attention</i>	
num_heads	8
<i>Learning Rate Scheduler</i>	
num_warmup_steps	0.1 * total_steps
<i>Early Stopping</i>	
Mode	"max"
Patience	3 epochs
Monitor	Validation Macro F1-score
<i>Training</i>	
Max Epochs	20
Accelerator	GPU

Table 10: Training hyperparameters.

(2025). Both model accessed via OpenRouter³ using the model identifiers "openai/gpt-4o" and "meta-llama/llama-4-maverick". The parameters used for generation are shown in Table 9, while the prompts are presented in Figures 13.

C PHAS-ID Implementation Details

The PHAS-ID framework was optimized using AdamW (Loshchilov and Hutter, 2019), employing a differential learning rate scheme for the shared backbone encoder PLM (η_{plm}) and randomly initialized networks (η_{mlp}). To ensure training stability and prevent overfitting, we utilized PyTorch `get_cosine_schedule_with_warmup` learning rate scheduler and implemented early

³<https://openrouter.ai>

stopping based on the validation macro F1-score. The complete configuration of all hyperparameters is detailed in Table 10.

D Artifacts License and Intended Use

Datasets We utilize three datasets: CoAuthor (Lee et al., 2022), HACo-Det (Su et al., 2025), and the Blog Authorship Corpus (Schler et al., 2006). CoAuthor is freely available on its official homepage, HACo-Det is distributed via GitHub under the MIT License, and the Blog Authorship Corpus is publicly hosted on Kaggle with the stipulation that it “may be freely used for non-commercial research purposes.” We strictly adhere to these intended use guidelines and utilize all datasets exclusively for non-commercial research.

Large Language Models We utilize LLMs for both detection inference and synthetic data generation. For AI detection inference, we use GPT-4o (OpenAI et al., 2024) and Llama 4 Maverick (Meta, 2025). GPT-4o is released under a proprietary license, whereas Llama 4 Maverick operates under the Llama 4 Community License; both permit inference for our specific use case. For synthetic dataset generation, we employ GPT-5.4 and Claude Sonnet 4.6 (both proprietary), alongside DeepSeekV4 Flash (MIT License). Although the terms of service for GPT-5.4 and Sonnet 4.6 restrict the use of model outputs for generating synthetic data, this prohibition specifically targets the training of competing generative LLMs. Our application is fundamentally different: we use the generated outputs strictly to create an AI detection dataset for training an AI text detector, which is fully compliant with their terms.

Pretrained Language Models We use DeBERTa-V3 (He et al., 2023) as a baseline, which is released under the MIT License.

Source Code In our experiments, we utilize the official source code for several baselines: SeqXGPT (Wang et al., 2023), PTD (Li et al., 2024), and GL-CLiC (Adi et al., 2025). The source codes for SeqXGPT and GL-CLiC are licensed under the MIT License, while PTD is distributed under the CC BY 4.0 License.

PHAS-ID and GhostFill Upon publication, we will release the source code for PHAS-ID, as well as the GhostFill dataset and its associated code, under the MIT License.

System: You are a competent but generic blog writer. Write clear, informative blog posts without any distinctive stylistic choices - represent what an average blogger would write.

Below is the opening of a blog post. Identify the topic, then write a generic blog post on the same topic.

BLOG POST OPENING:
{opening}

- - -

Write a complete blog post on this topic (~{word_count} words). Write like an average blogger: clear and competent but without distinctive style choices. Use no special formatting or headers.

Figure 6: Base post generation prompt.

System: You are a sophisticated writing analyst. You analyze an author's blog writing style by contrasting their post with a generic base post on the same topic. Focus only on what makes the author DISTINCTIVE - patterns that would NOT appear in average blog writing.

Below are two blog posts on the same topic.

TOPIC: {topic_phrase}

BASE BLOG POST (generic, written by an average blogger with no style guidance):
{base_post}

AUTHOR'S BLOG POST:
{author_post}

- - -

Create an Author Writing Sheet by analyzing how the AUTHOR'S POST differs from the BASE POST.

For each dimension, provide 5-10 Claim-Evidence pairs:

- * Claims must describe BROADER PATTERNS - not tied to specific content details
- * Claims must be UNIQUE and NON-REDUNDANT across all four dimensions
- * Each claim must be PERFECTLY SUPPORTED by the chosen evidence
- * Evidence must be a DIRECT QUOTE from the AUTHOR'S post only
- * Use framing: "In the blog post regarding '{topic_phrase}', the author..."
- * Avoid adverbs of frequency ("often", "frequently")

Dimensions:

1. PLOT - how the author structures the post, opens and closes it, develops the central argument, and uses transitions
 2. CREATIVITY - unconventional reinterpretation of topics, blending of registers, humor, imaginative angles, surprising choices
 3. DEVELOPMENT (Character and Setting) - use of examples, personal anecdotes, emotional arcs, analogies, level of detail and depth
 4. LANGUAGE USE - vocabulary, tone, sentence structure, rhetorical devices, figures of speech, pacing, narrative voice
-

Figure 7: Delta sheet extraction prompt.

System: You are a writing analyst. You merge two Author Writing Sheets into a single refined sheet that captures consistent style patterns observed across multiple blog posts.

PREVIOUS COMBINED AUTHOR WRITING SHEET:
{prev_sheet}

CURRENT AUTHOR WRITING SHEET (from blog post [{story_ref}]):
{curr_sheet}

- - -

Merge these two sheets using the following algorithm:

1. GROUP equivalent claims (nearly identical behavior with the same underlying tendency)
2. For grouped claims, SELECT the single best evidence (prefer verbatim quotes over paraphrased)
3. REWRITE claims so every clause is directly supported by the chosen evidence
4. CARRY FORWARD unique ungrouped claims from both sheets
5. LIMIT to maximum 10 claim-evidence pairs per dimension
6. In the evidence field, add a [{story_ref}] tag at the end of evidence drawn from blog post {story_ref}

Return a single merged Author Writing Sheet in the same structured format.

Figure 8: Incremental sheet combination prompt.

System: You are a writing coach. You create vivid, actionable persona descriptions from structured author writing sheets.

AUTHOR WRITING SHEET:
{sheet}

- - -

Write a persona description for this author (~300 words) that:

- * Uses second-person voice ("You...")
 - * Organizes content into 4 sections: Plot, Creativity, Development (Character and Setting), Language Use
 - * Describes recurring patterns, unique traits, and stylistic strengths
 - * Reads as flowing narrative prose - NOT a bullet list
 - * Captures the author's distinctive voice and choices
-

Figure 9: Persona description generation prompt.

System: You are a collaborative writing assistant. You translate an author persona description into concrete, actionable writing rules tailored to a specific document.

AUTHOR PERSONA:
{persona}

DOCUMENT WITH GAPS TO FILL:
{document}

- - -

Generate actionable fill rules for this specific document. Rules must:

- * Be written as standalone second-person instructions ("You should...", "Write...")
- * Use NO comparative language ("replace", "change", "deepen", "add more")
- * Include rich, document-specific examples where possible
- * Be absolute and specific - not vague or generic
- * For Language Use: provide verbatim style examples drawn from the visible sentences

Cover all four dimensions: Plot, Creativity, Development (Character and Setting), Language Use.

Figure 10: Per-document fill rules generation prompt.

System: You are a collaborative writing assistant. You write new sentences that blend seamlessly into an existing blog post, matching the original author's distinctive style.

=== AUTHOR PERSONA ===
{persona}

=== FILL RULES FOR THIS DOCUMENT ===
{rules}

Below is a blog post where some sentences have been replaced with [MASK].
Fill in each [MASK] to produce a complete, coherent blog post in the author's style.

CRITICAL RULES:

- Copy every non-[MASK] sentence EXACTLY as written - do not rephrase, reorder, or omit any of them.
- Only write new content in place of [MASK] tokens.
- The final post must have the same total number of sentences as the input (one new sentence per [MASK]).

=== BLOG POST ===
{document}
=== END ===

Return the complete filled blog post.

Figure 11: Filling deleted spans with style-consistent text prompt.

System: You are a writing judge. You evaluate multiple versions of AI-filled sentences in a blog post and select the version that best matches the author's style.

AUTHOR PERSONA:
{persona}

Below are {n} versions of the same blog post. Human-written sentences are identical in every version; only the [AI] sentences differ.

{versions}

- - -

Select the version number (1 to {n}) where the AI-generated sentences best match the author's style and blend most naturally with the human-written sentences.

Figure 12: Best-of- P fill selection prompt.

You are an expert text classifier. Your task is to determine, for each sentence in a document, whether it was written purely by an AI, purely by a Human, or through Human-AI collaboration.

- * Use "AI" if the sentence appears entirely generated by an AI.
- * Use "Human" if the sentence appears entirely written by a human.
- * Use "Human-AI" if the sentence shows characteristics of both AI generation and human writing/editing (e.g., AI text modified by a human, human text with AI-generated parts, or a blend of styles).

Author Writing Style Reference

The following documents were written by the same author as the target document. Each sentence is labelled with its true class. Use these examples to calibrate your understanding of this author's writing style before classifying the target.

Reference Document 1

[<LABEL>] <SENTENCE>

[<LABEL>] <SENTENCE>

...

Reference Document K

[<LABEL>] <SENTENCE>

[<LABEL>] <SENTENCE>

...

Target Document to Classify

Classify each numbered sentence below. Consider the author's writing style shown above.

1. <TARGET_SENTENCE_1>

2. <TARGET_SENTENCE_1>

...

N. <TARGET_SENTENCE_N>

Output Format

Respond with a JSON object with a single key "labels" containing an array of exactly N strings.

Each element must be exactly one of: "Human", "AI", or "Human-AI".

The array must contain exactly N elements, one per sentence, in the same order.

Example: "labels": ["Human", "AI", "Human-AI"]

Figure 13: Personalized LLM prompt.