LEARNING SEQUENTIAL–STRUCTURAL DISCREPANCY: LABEL-FREE PRETRAINING FOR DOMAIN-ROBUST LLM-GENERATED TEXT DETECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

The rapid progress of large language models (LLMs) has made it increasingly difficult to distinguish human-written from LLM-generated text, particularly under domain shift. Existing training-based detectors typically learn sequential or syntactic cues under human/LLM provenance supervision, while approaches combining the two often rely on direct feature concatenation, leaving their relationship underexplored and potentially capturing domain-specific artifacts. We empirically find a systematic difference between human-written and LLM-generated text in the discrepancy between sequential and structural representations: human-written text tends to show a larger cross-view discrepancy, whereas LLM-generated text exhibits stronger cross-view consistency. Importantly, this distributional gap persists across domains even when the two views are learned without human/LLM provenance labels. Based on this finding, we propose the **Sequential–Structural Discrepancy** (SSD) framework. SSD first pretrains sequential and structural encoders on unlabeled text through masked representation prediction and subsequently freezes both encoders for detection. The resulting token-level discrepancy plays two complementary roles: it controls the adaptive fusion of the sequential and structural views and is retained as an explicit feature for downstream classification. Supervised training with human/LLM provenance labels updates the downstream fusion and classification modules, while both pretrained view encoders remain frozen. Experiments on in-domain and out-of-domain benchmarks demonstrate that SSD consistently outperforms strong baselines and maintains robust performance under adversarial perturbations.

1 INTRODUCTION

The rapid advancement of large language models (LLMs)—such as GPT-5 (OpenAI, 2025) and Qwen3 (Qwen Team, 2025)—has made it increasingly difficult to distinguish LLM-generated text from human writing. Such capabilities can be exploited for malicious purposes, including fake-news fabrication and dissemination (Zellers et al., 2019; Hu et al., 2025), academic misconduct (Currie, 2023), and social-media manipulation (Zhang et al., 2024). Developing reliable detectors for LLM-generated text has therefore become an important and urgent task (Cao, 2025).

Existing detection approaches broadly fall into two categories. Train-free methods infer text provenance from properties exposed by pretrained LLMs and often exhibit strong cross-domain generalization. However, they typically require repeated model inference or auxiliary generation for each sample, resulting in high computational cost (Bao et al., 2024; Yang et al., 2024; Xu et al., 2025); their effectiveness can also vary substantially across generators. Training-based methods instead learn discriminative patterns from vocabulary usage, syntactic structure, or semantic representations. Although they offer high accuracy and efficient inference when training and test data follow similar distributions, their performance often degrades sharply under domain shifts (Hao et al., 2025; Guo et al., 2026). This limitation suggests that supervised detectors may capture domain- or generator-specific artifacts rather than transferable characteristics of LLM-generated text.

Rather than modeling sequential fluency and syntactic structure as isolated cues, we focus on the relationship between the two views. Autoregressive LLMs generate each token from its preceding

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.

context, favoring word transitions that remain compatible with recurring syntactic patterns. This process can make word-order and dependency evidence mutually reinforcing, leading to tighter cross-view consistency. Human writing, by contrast, is additionally shaped by communicative intent, stylistic choice, revision, and non-local discourse planning, which may introduce greater variation between linear flow and hierarchical organization. We therefore hypothesize that LLM-generated text exhibits lower sequential–structural discrepancy. To examine this hypothesis without allowing provenance labels to induce the difference, we independently pretrain and freeze a sequential encoder and an RGCN (Schlichtkrull et al., 2018) that reconstruct the same masked contextual targets from their respective views. As shown in Figure 1, their sentence-level discrepancy ($\overline{D}$) exhibits a consistent distributional gap between human-written and LLM-generated text on both L2R (Hao et al., 2025) and M4 (Wang et al., 2024), motivating its use as a transferable detection signal.

This observation raises a natural question: how can a supervised detector exploit the cross-view discrepancy without allowing provenance supervision to redefine it? We address this question with the **Sequential–Structural Discrepancy** (SSD) framework, which explicitly separates view learning from provenance classification. In the first stage, the sequential encoder learns from word-order context, while the RGCN propagates information over dependency relations. Both encoders are independently pretrained on provenance-unlabeled text to reconstruct the same frozen teacher representations. Although they rely on different evidence, the shared targets anchor their outputs in a common feature space, making their distance a comparison between two predictions of the same linguistic content rather than between unrelated latent representations. The pretrained encoders are then frozen, preserving the discrepancy observed before provenance labels are introduced. In the second stage, SSD computes their element-wise absolute difference as the token-level discrepancy D. The discrepancy determines token-wise gates for adaptive view fusion and is also retained as an explicit input to the final classifier. Only the downstream modules are optimized with provenance labels, and SSD can therefore learn how to use the cross-view signal for classification without altering the two encoders that produce it.

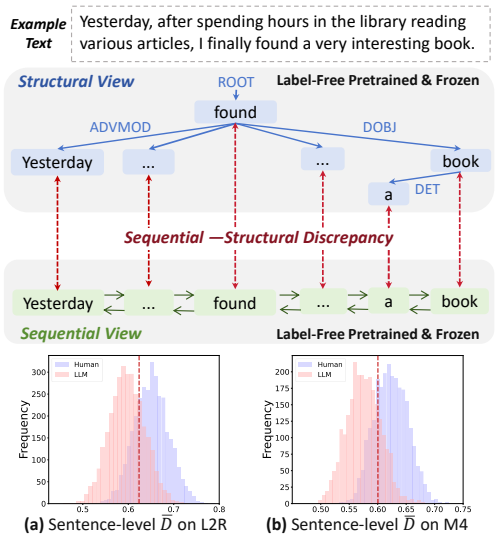


Figure 1: Sentence-level sequential–structural discrepancy $\overline{D}$ computed by frozen encoders pretrained without provenance labels exhibits a consistent human–LLM distributional gap across domains.

Our contributions are summarized as follows:

- We identify and empirically validate a consistent distributional gap in sequential–structural discrepancy between human-written and LLM-generated texts. This gap emerges from frozen dual-view encoders pretrained without human/LLM provenance labels and persists across domains, supporting discrepancy as a transferable detection signal.
- We propose SSD, a two-stage framework that separates label-free dual-view pretraining from supervised downstream detection. SSD uses the resulting token-level discrepancy both to guide adaptive view fusion and as an explicit feature for self-attention-based classification, while keeping the pretrained view encoders frozen.
- Experiments on the in-domain L2R and out-of-domain M4 benchmarks show that SSD consistently outperforms strong baselines. Ablation and robustness analyses further verify the contributions of label-free pretraining and discrepancy modeling under domain shifts, adversarial perturbations, and imperfect dependency parsing.

2 Methodology

We propose the Sequential–Structural Discrepancy (SSD) framework, which detects LLM-generated text by explicitly modeling the disagreement between sequence-conditioned and graph-conditioned

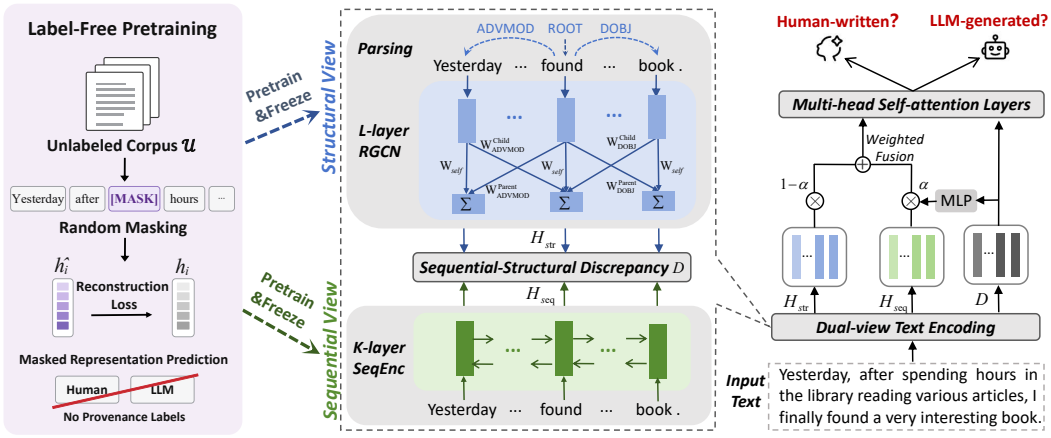


Figure 2: Overview of our proposed SSD. The sequential and structural encoders are pretrained on unlabeled text through masked representation prediction and then frozen. Their token-level discrepancy D guides adaptive view fusion and is retained for self-attention-based classification. Human/LLM labels supervise only the downstream modules. The input projection and frozen proxy-model score are omitted for clarity.

representations. As illustrated in Figure 2, SSD separates representation learning from provenance classification. First, two view-specific encoders are independently pretrained on text without human/LLM provenance labels through masked representation prediction. The pretrained encoders are subsequently frozen. During downstream training, their token-level discrepancy guides adaptive view fusion and is provided as an explicit feature to a self-attention classifier. Consequently, provenance supervision updates only the downstream modules rather than the two view encoders.

2.1 Dual-View Text Encoding

2.1.1 Sequential-View Encoding

Let $T = (t_1, \ldots, t_L)$ denote the word sequence produced by the dependency parser. The fixed pretrained language model first represents T as subword vectors. For each word t_i, we mean-pool the vectors of its aligned subwords to obtain a word-level contextual representation x_i; special and padding tokens are excluded. This yields $X = [x_1, \ldots, x_L] \in \mathbb{R}^{L \times d}$, whose positions align one-to-one with the dependency-graph nodes. From the sequential perspective, we use an encoder E_{seq} to model local contextual dependencies along the word order:

$$H_{\text{seq}} = E_{\text{seq}}(X). \tag{1}$$

Here, $H_{\text{seq}} \in \mathbb{R}^{L \times d}$ encodes lexical compatibility, local coherence, and contextual flow. SSD is compatible with different sequential architectures, such as BiLSTM (Hochreiter & Schmidhuber, 1997) and Transformer (Vaswani et al., 2017); we use a Transformer in the main experiments.

2.1.2 Structural-View Encoding

To capture syntactic organization, we parse T into a directed dependency graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where each word is a node and every edge is associated with a typed dependency relation (e.g., `nsubj` or `dobj`). To propagate information in both directions, each head-to-dependent edge is paired with an inverse edge, and the two directions use separate relation-specific transformations. Let $\widetilde{\mathcal{R}} = \mathcal{R}^{\rightarrow} \cup \mathcal{R}^{\leftarrow}$ denote this direction-aware relation set. Starting from $h_i^{(0)} = x_i$, an RGCN layer updates word i as

$$h_i^{(l+1)} = \sigma\left(W_0^{(l)} h_i^{(l)} + \sum_{\rho \in \widetilde{\mathcal{R}}} \sum_{j \in \mathcal{N}_\rho(i)} \frac{1}{c_{i,\rho}} W_\rho^{(l)} h_j^{(l)} \right), \tag{2}$$

where $\mathcal{N}_\rho(i)$ contains the neighbors connected to token i by the directed relation ρ, and $c_{i,\rho} = |\mathcal{N}_\rho(i)|$ is a normalization factor. $W_0^{(l)}$ preserves the token's own state, while $W_\rho^{(l)}$ transforms messages

according to both dependency type and direction. After K layers, the structural representation is

$$H_{\text{str}} = E_{\text{str}}(X, \mathcal{G}) = \text{RGCN}(X, \mathcal{G}), \tag{3}$$

where $H_{\text{str}} \in \mathbb{R}^{L \times d}$ captures relation-aware syntactic context. View-specific projection layers are included in E_{seq} and E_{str} so that the two outputs have the same dimensionality.

2.2 Label-Free View Pretraining

Directly optimizing the view encoders with provenance labels could make their discrepancy reflect artifacts of a particular training set. We instead learn both views from a text corpus $\mathcal{U}$ without human/LLM provenance labels using masked representation prediction. For each text T, we randomly select a set of non-special word positions $\mathcal{M}$ and mask all language-model subwords aligned with each selected word, producing $\widetilde{T}$. A fixed pretrained language model then encodes the original and masked sequences; aligned subwords are mean-pooled as described above to obtain $X = \text{PLM}(T)$ and $\widetilde{X} = \text{PLM}(\widetilde{T})$, respectively.

For the structural view, the dependency graph $\mathcal{G}$ is parsed from the original text and retained after masking. This can be viewed as masked node-feature reconstruction on a fixed graph: masking removes the lexical feature of a selected node, while the RGCN must reconstruct its contextual target from the dependency topology, relation types, and neighboring representations. The retained graph contains neither the identity of the masked word nor any human/LLM provenance annotation. Thus, x_i provides a contextual target for reconstructing the masked position without exposing provenance labels.

The sequential and structural encoders independently reconstruct these targets from their respective contexts:

$$\widehat{H}_{\text{seq}} = E_{\text{seq}}(\widetilde{X}), \qquad \widehat{H}_{\text{str}} = E_{\text{str}}(\widetilde{X}, \mathcal{G}). \tag{4}$$

For a view $v \in \{\text{seq}, \text{str}\}$, the pretraining objective is

$$\mathcal{L}^{v}_{\text{pre}} = \frac{1}{|\mathcal{M}|} \sum_{i \in \mathcal{M}} \left[\frac{1}{d} \|\widehat{h}^{v}_i - \text{sg}(x_i)\|_2^2 + \lambda_{\cos} \big(1 - \cos(\widehat{h}^{v}_i, \text{sg}(x_i))\big) \right], \tag{5}$$

where $\text{sg}(\cdot)$ denotes stop-gradient and $\cos(\cdot, \cdot)$ is cosine similarity along the feature dimension. The fixed language model provides the targets and is not updated. Because both views reconstruct the same targets through projections retained downstream, their outputs lie in a common teacher-defined space. Their discrepancy therefore compares predictions of the same content from different evidence, rather than unrelated latent codes. For the squared-error term, the population-optimal prediction is $h^{v,*}_i = \mathbb{E}[x_i \mid C_v]$, where C_v denotes the context available to view v. The cross-view distance consequently measures how the prediction changes between sequential and structural contexts, providing an intuitive basis for the empirically observed provenance gap. The encoders are optimized independently with $\mathcal{L}^{\text{seq}}_{\text{pre}}$ and $\mathcal{L}^{\text{str}}_{\text{pre}}$, without provenance labels or a cross-view decoupling loss, and are frozen after pretraining.

2.3 Discrepancy-Based Classification

For downstream detection, the frozen encoders process the unmasked input to produce H_{seq} and H_{str}. We define their token-level sequential–structural discrepancy as

$$D = \big|H_{\text{seq}} - H_{\text{str}}\big|, \tag{6}$$

where the absolute value is applied element-wise. Thus, $D \in \mathbb{R}^{L \times d}$ highlights token positions and feature dimensions for which sequential and structural evidence disagree. For the distributional analysis in Figure 1, we further reduce this token-level tensor to a sentence-level scalar. We first compute the Euclidean discrepancy at each token and then average over the sequence:

$$d_i = \|h_{\text{seq},i} - h_{\text{str},i}\|_2, \qquad \overline{D}(T) = \frac{1}{L} \sum_{i=1}^{L} d_i. \tag{7}$$

The scalar $\overline{D}(T)$ is used only for visualization and distributional analysis; the full token-level tensor D is retained by the downstream detector.

We use D in two downstream operations only. First, an MLP maps it to a scalar token-wise gate α, whose entries lie in $(0, 1)$:

$$\alpha = \sigma(W_2 \tanh(W_1 D + b_1) + b_2)\,. \tag{8}$$

The gate is broadcast over the feature dimension and adaptively combines the two frozen views:

$$H_{\text{mix}} = \alpha \odot H_{\text{seq}} + (1 - \alpha) \odot H_{\text{str}}. \tag{9}$$

Second, D is retained as an explicit input feature. As an auxiliary cue, a frozen proxy language model provides one scalar token score for each word. Let $\mathcal{A}_i$ be the proxy-model subwords aligned with word t_i; we define $p_i = |\mathcal{A}_i|^{-1} \sum_{j \in \mathcal{A}_i} \log p_\phi(s_j \mid s_{<j})$ and collect these mean subword log-probabilities as $P = [p_1, \ldots, p_L]^\top \in \mathbb{R}^{L \times 1}$. We first concatenate the features and project them to the attention dimension:

$$Z_{\text{raw}} = [H_{\text{mix}}; \text{sg}(D); P] \in \mathbb{R}^{L \times (2d+1)}, \qquad Z = Z_{\text{raw}} W_z + b_z \in \mathbb{R}^{L \times d_a}, \tag{10}$$

where $W_z \in \mathbb{R}^{(2d+1) \times d_a}$ and d_a is chosen to be divisible by the number of attention heads. The projected sequence is then aggregated as

$$S = \text{MaxPool}\big(\text{MHSA}(Z)\big), \tag{11}$$

where $[\cdot;\cdot]$ denotes feature-wise concatenation and padding positions are excluded from both attention and max pooling. Finally, a linear classifier predicts the provenance distribution

$$p(y \mid T) = \text{softmax}(W_c S + b_c). \tag{12}$$

Only the gate, input projection, multi-head self-attention module, and classifier are optimized with the cross-entropy objective $\mathcal{L}_{\text{cls}}$, while the view encoders and proxy model remain frozen. This design allows the detector to exploit the label-free discrepancy signal without reshaping the frozen view encoders through provenance supervision.

3 Experiments

3.1 Experimental Setup

Datasets and metric. We train and evaluate in-domain on L2R (Hao et al., 2025), which spans 21 domains and four generators, and evaluate zero-shot transfer on the English portion of M4 (Wang et al., 2024), a separate benchmark with substantial domain and generator shifts relative to L2R. This protocol contrasts performance on matched data with cross-dataset transfer. M4 is not used for model selection. The view encoders are pretrained on 300K unlabeled C4 (Dodge et al., 2021) texts used only for self-supervised pretraining and excluded from downstream training and evaluation splits. We use AUROC as the primary metric because it does not require selecting a test-set decision threshold.

Model and optimization. RoBERTa-base (Liu et al., 2019) supplies frozen 768-dimensional targets and spaCy supplies dependency graphs. We use a two-layer Transformer and a two-layer RGCN (hidden size 768), with inputs truncated to 256 subwords. Each encoder is pretrained for 10 epochs (AdamW, batch size 32, learning rate 10^{-4}) and then frozen; the downstream modules are trained for 30 epochs (batch size 256, learning rate 10^{-5}). Proxy-based methods use frozen Qwen3-4B. Unless otherwise stated, reported SSD performance is averaged over three seeds, with selection based only on L2R validation data. Complete settings are in Appendices A–B.

Baselines. We compare with Fast-DetectGPT (Bao et al., 2024), RAIDAR (Mao et al., 2024), and Lastde (Xu et al., 2025), together with Ghostbuster (Verma et al., 2024), L2R (Hao et al., 2025), AdaDetectGPT (Zhou et al., 2025), ImBD (Chen et al., 2025), and L2D (Zhou et al., 2026). The baselines cover probability-, perturbation-, rewriting-, and representation-based detectors. Training-based methods use the same L2R split; official implementations and recommended settings are used whenever available, and proxy-based methods share Qwen3-4B.

3.2 IN-DOMAIN PERFORMANCE

Table 1: Per-domain AUROC (%) on L2R. SSD achieves the highest average AUROC and the lowest standard deviation. Bold and underlined values denote the best and second-best results, respectively.

	Train-free			Training-based					
Domain	**Fast-DetectGPT**	**RAIDAR**	**Lastde**	**Ghost-buster**	**L2R**	**Ada-DetectGPT**	**ImBD**	**L2D**	**SSD**
AcademicResearch	48.55	83.11	60.87	65.70	84.06	54.18	90.19	<u>94.46</u>	**95.82**
ArtCulture	63.25	77.11	78.97	67.55	83.28	54.70	74.93	<u>87.55</u>	**96.69**
Business	69.29	83.69	84.70	84.41	<u>91.56</u>	51.63	87.21	90.90	**96.02**
Code	69.42	38.40	84.42	38.38	83.83	58.10	77.33	<u>90.12</u>	**94.97**
EducationMaterial	75.46	96.75	89.75	85.23	96.44	54.65	<u>99.54</u>	96.02	**99.70**
Entertainment	85.01	83.19	97.88	87.39	94.94	51.52	<u>98.27</u>	97.93	**98.96**
Environmental	83.25	92.28	98.26	84.17	97.86	56.62	94.12	<u>98.83</u>	**99.98**
Finance	69.69	81.53	84.71	78.82	94.00	49.40	94.65	<u>98.51</u>	**98.89**
FoodCuisine	76.84	78.31	90.31	68.46	95.47	57.43	86.33	<u>96.59</u>	**97.06**
GovernmentPublic	71.46	76.19	85.77	68.28	86.75	53.23	89.35	<u>92.82</u>	**95.32**
LegalDocument	83.77	65.94	**99.67**	54.14	78.03	51.27	98.54	98.03	<u>98.76</u>
LiteratureCreativeWriting	79.43	91.61	95.99	94.56	92.94	64.43	<u>99.62</u>	99.31	**99.97**
MedicalText	56.48	77.00	**91.15**	63.67	78.57	54.19	76.17	81.50	<u>87.01</u>
NewsArticle	60.82	85.47	74.11	68.66	92.42	62.95	89.25	<u>96.85</u>	**96.97**
OnlineContent	63.12	82.31	78.56	60.22	88.81	53.55	83.51	<u>96.83</u>	**98.22**
PersonalCommunication	55.78	72.33	70.46	70.42	82.39	56.34	74.60	**91.51**	<u>88.85</u>
ProductReview	66.80	80.75	81.56	73.59	96.89	51.97	89.84	**97.48**	<u>97.06</u>
Religious	66.00	83.97	79.75	61.00	97.75	63.49	<u>98.36</u>	95.43	**98.93**
Sports	60.17	78.69	76.46	66.66	87.42	51.53	83.95	**91.67**	<u>91.02</u>
TechnicalWriting	60.96	85.75	76.36	72.30	93.69	51.86	95.95	<u>98.06</u>	**98.93**
TravelTourism	62.47	88.97	79.16	76.07	94.75	49.65	90.12	<u>99.18</u>	**99.24**
AVERAGE	68.00	80.16	83.76	70.94	90.09	54.89	89.13	<u>94.74</u>	**96.59**
STD	9.90	11.81	9.86	12.59	6.34	<u>4.35</u>	8.28	4.54	**3.59**

As shown in Table 1, SSD achieves the highest average AUROC of 96.59% and ranks among the top two methods in every domain. Its cross-domain standard deviation is also the lowest (3.59), indicating that the overall improvement is not produced by a small number of favorable domains. Instead, SSD maintains consistently strong performance across domains with substantially different vocabulary, style, and structural conventions.

The domain-level results further reveal where the cross-view signal is most useful. SSD obtains particularly clear gains on ArtCulture (+9.14 points over the strongest baseline) and Code (+4.85 points), although these domains represent markedly different writing regimes: the former permits considerable stylistic variation, whereas the latter is highly constrained by formal structure. This contrast suggests that SSD does not rely on a single domain-specific lexical pattern, but benefits from modeling how sequential and structural evidence interact. The main exception is MedicalText, where SSD trails Lastde by 4.14 points. One possible explanation is that specialized terminology and formulaic clinical expressions also make human-written medical text highly regular, thereby reducing the contrast captured by the two views.

3.3 OUT-OF-DOMAIN PERFORMANCE

Table 2: OOD detection performance on M4 in terms of AUROC (%).

	Train-free			Training-based					
Metric	**Fast-DetectGPT**	**RAIDAR**	**Lastde**	**Ghost-buster**	**L2R**	**Ada-DetectGPT**	**ImBD**	**L2D**	**SSD**
AUROC	59.88	69.31	<u>76.72</u>	39.67	65.61	50.66	73.73	75.39	**78.39**

As shown in Table 2, supervised baselines degrade markedly under the combined domain-and-generator shift: Ghostbuster falls to 39.67% AUROC and L2R to 65.61%, indicating reliance on the training distribution. SSD achieves the best OOD AUROC of 78.39%, exceeding the train-free Lastde

and the training-based L2D. This supports our design: label-free pretraining provides transferable dual-view representations, while freezing the encoders limits adaptation to domain-specific cues.

3.4 Transferability of the Frozen Discrepancy

To evaluate the discrepancy signal before downstream finetuning, we average the token-level differences into a sentence-level score $\overline{D}$. Because LLM-generated texts tend to have lower discrepancy, we use $-\overline{D}$ as the detection score, treating LLM-generated text as the positive class. AUROC is then computed directly without fitting a classifier or selecting a decision threshold.

Table 3: Sentence-level discrepancy produced by the frozen encoders. Gap denotes Human minus LLM; AUROC is computed directly from $-\overline{D}$ without a trained classifier.

Dataset	Human mean	LLM mean	Gap	$-\overline{D}$-only AUROC
L2R	0.6511	0.5973	0.0538	74.88
M4	0.6235	0.5754	0.0481	72.14

As shown in Table 3, human-written texts exhibit higher mean discrepancy on both datasets, and the separation remains evident on M4 despite the domain shift. Using $-\overline{D}$ directly yields AUROCs of 74.88% on L2R and 72.14% on M4, substantially above chance without a trained classifier. These scores remain 21.71 and 6.25 points below the full model, respectively, showing that the sentence-level discrepancy is transferable but does not replace token-level discrepancy patterns, adaptive fusion, or the underlying view representations.

3.5 Effect of Unlabeled Pretraining Scale

We vary the amount of provenance-unlabeled text while fixing the objective, downstream data, and optimization. The 0K setting freezes randomly initialized encoders as a reference.

Table 4: Mean AUROC (%) with different amounts of unlabeled pretraining text.

Setting	0K	10K	50K	100K	200K	300K
ID	85.45	87.02	91.11	94.25	<u>96.18</u>	**96.59**
OOD	45.05	55.95	68.69	72.86	<u>76.80</u>	**78.39**

Table 4 shows monotonic gains with corpus size: from 0K to 300K, AUROC rises by 11.14 points on L2R and 33.34 points on M4. The first 50K examples account for 50.8%/70.9% of the ID/OOD gains, while 200K captures over 95% of both. Although marginal gains diminish, the final 100K still adds 0.41/1.59 ID/OOD points. We therefore use 300K as a practical scale, while larger corpora may provide further improvements.

3.6 Ablation Analysis

We organize the ablations around three questions: how the view encoders are learned, whether both views are necessary, and how D is used in downstream integration. Except for the targeted change, the downstream data and optimization remain fixed.

View learning. As shown in the first group of Table 5, frozen random encoders trail SSD by 11.14/33.34 points ID/OOD, confirming that the label-free objective learns useful sequential and structural regularities. Training from scratch and fine-tuning pretrained encoders slightly exceed SSD by 0.77 and 0.61 ID points, but fall behind by 8.56 and 8.39 points on M4. The near-identical OOD results of these two supervised variants show that a good initialization alone is insufficient: allowing provenance labels to update the encoders improves distribution-specific fitting but weakens transfer. Freezing is therefore a substantive part of the design rather than merely a training convenience.

Complementary views. The single-view rows in Table 5 show that the sequence-only model outperforms structure-only by 7.93 points ID and 3.51 points OOD, but neither is competitive with SSD. Relative to the stronger sequence-only model, combining both views adds 8.42 points on L2R

and 21.14 on M4. The much larger OOD gain indicates that the structural branch is not simply an auxiliary in-domain feature; it provides complementary evidence when sequential patterns change across domains and generators.

Table 5: Ablation results on L2R and M4 in terms of AUROC (%).

Variant	L2R (ID)	M4 (OOD)
View learning		
Frozen random encoders (0K)	85.45	45.05
Supervised encoder training from scratch	97.36	69.83
Label-free pretraining, then encoder fine-tuning	97.20	70.00
Single-view modeling		
Sequence only	88.17	57.25
Structure only	80.24	53.74
Dual-view integration		
w/o proxy score P	94.89	74.97
D for gating only (remove D from MHSA)	95.46	75.61
w/o gating mechanism	95.61	69.26
w/o discrepancy D entirely	94.27	66.89
w/o H_{seq} in MHSA	94.85	72.09
w/o H_{str} in MHSA	94.87	71.87
Full model	**96.59**	**78.39**

Discrepancy and integration. The final group of Table 5 shows that removing D causes the largest downstream-component loss: 2.32 points ID and 11.50 OOD. Removing the gate costs 9.13 OOD points, indicating that discrepancy-conditioned fusion is particularly important under distribution shift. Retaining D for gating but removing it from MHSA reaches 75.61% on M4, 2.78 points below the full model, while removing either H_{seq} or H_{str} produces similar 6.30–6.52 point OOD losses. Removing P leads to a moderate performance decrease, while the substantially larger losses caused by removing D or discrepancy-guided gating indicate that the main gains arise from dual-view discrepancy modeling.

3.7 Adversarial Robustness

Following Hao et al. (2025), we test matched samples under Decoherence (local disruption) and Rewrite (broader paraphrasing). More details are in Appendix D.

Table 6: AUROC (%) on matched clean/attacked subsets.

	Train-free			Training-based					
Setting	**Fast-DetectGPT**	**RAIDAR**	**Lastde**	**Ghost-buster**	**L2R**	**Ada-DetectGPT**	**ImBD**	**L2D**	**SSD**
No Attack	68.00	80.16	83.76	70.94	90.09	54.89	89.13	<u>94.74</u>	**96.59**
Decoherence	49.98	77.22	70.75	48.59	87.10	51.54	82.56	<u>90.71</u>	**92.60**
Rewrite	50.92	68.68	79.19	40.50	85.68	49.71	79.43	<u>87.91</u>	**88.55**

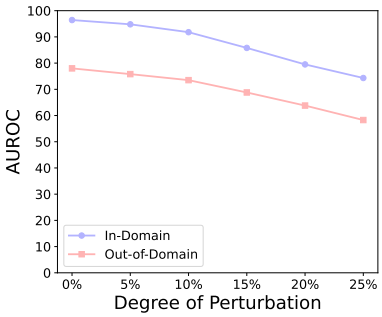
As shown in Table 6, SSD remains strongest under both attacks. Decoherence lowers its AUROC by 3.99 points to 92.60%, still 1.89 points above L2D; this modest decline is consistent with the structural branch compensating for disturbed local word order. Rewrite causes a larger 8.04-point drop and narrows the margin over L2D to 0.64, reflecting its broader changes to lexical and sequential patterns. SSD nevertheless remains best at 88.55%, indicating that the cross-view signal survives substantial surface-form perturbations.

3.8 Generalization across Parsers and Tolerance to Parsing Errors

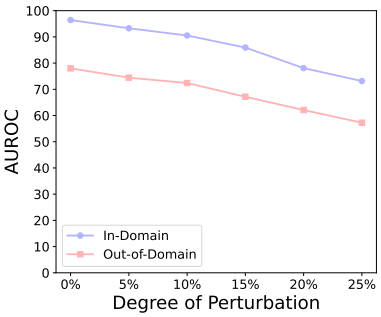
We test whether SSD is tied to a particular parser or requires highly accurate dependency trees. We first replace spaCy with Stanza and retrain the complete pipeline using the corresponding parses. We

Parser	L2R	M4
spaCy	**96.59**	**78.39**
Stanza	96.48	77.98

(a) Parser choice.



(b) Relation-type errors.



(c) Attachment errors.

Figure 3: Generalization across dependency parsers and tolerance to additional errors in automatically predicted dependency trees.

then inject additional errors into 0–25% of the nodes in the automatically predicted trees: relation-type corruption changes dependency labels while preserving the topology, whereas attachment corruption redirects parent nodes and changes graph connectivity. Since the predicted trees are already imperfect, this experiment serves as an additional stress test rather than a simulation of natural parser errors.

As shown in Figure 3(a), replacing spaCy with Stanza reduces AUROC by only 0.11 points on L2R and 0.41 points on M4, indicating that SSD is not tied to parser-specific patterns. Panels (b) and (c) further evaluate tolerance to unreliable structural input by progressively corrupting relation types and parent attachments, respectively. Both produce gradual degradation rather than an abrupt failure, although even randomly corrupting 5% of the nodes constitutes a strong additional intervention on imperfect automatic parses. At 10% corruption, SSD retains 92.08/73.58% AUROC on L2R/M4 under relation-type errors and 90.84/72.65% under attachment errors. The larger decline in panel (c) is expected because attachment errors alter graph connectivity rather than only relation labels. Together, the three panels show that SSD generalizes across parsers and remains effective with imperfect dependency trees.

4 Related Work

LLM-generated text detectors broadly follow train-free or training-based paradigms. **Train-free methods** infer provenance from n-gram, probability, perplexity, curvature, rewriting, or decoding-sensitivity signals (Yang et al., 2024; Venkatraman et al., 2024; Shi et al., 2024; Xu & Sheng, 2024; Mitchell et al., 2023; Bao et al., 2024; Mao et al., 2024; Xu et al., 2025; Ma et al., 2026). They avoid task-specific training but may require proxy-model probabilities or repeated inference.

Training-based methods learn semantic (Tian et al., 2024; Ma et al., 2025), syntactic (Li et al., 2025; He et al., 2025; Guo et al., 2026), or probability-based classifiers, including Ghostbuster (Verma et al., 2024), AdaDetectGPT (Zhou et al., 2025), L2R, ImBD, and L2D (Hao et al., 2025; Chen et al., 2025; Zhou et al., 2026). Recent work further explores text-fluoroscopy, reasoning-aware alignment, and spectral representations (Yu et al., 2024; Wang et al., 2026; Liu et al., 2026). Despite these advances, performance remains sensitive to datasets and evaluation settings (Stowe & Patil, 2026). SSD instead separates representation learning from provenance supervision: it pretrains and freezes both view encoders without provenance labels, then uses labeled data only for downstream fusion and classification.

5 Conclusion

We propose SSD, an LLM-generated text detector based on the discrepancy between sequential fluency and syntactic structure. It pretrains both view encoders without provenance labels, freezes them downstream, and uses their discrepancy for adaptive fusion and classification. SSD outperforms strong baselines on L2R and M4 under domain shift and adversarial attacks, while ablations confirm the role of discrepancy in out-of-domain generalization. These results support separating label-free view learning from supervised decision learning.

AI USE STATEMENT

Generative AI tools were used to assist with language editing, minor LaTeX formatting, and code organization and refactoring. They were not used to develop the model or methodology, design or conduct experiments, analyze or interpret results, or generate data. All AI-assisted textual and code edits were reviewed and verified by the authors, and the resulting code was tested by the authors. The authors take full responsibility for the final content of this paper.

ETHICS STATEMENT

Automated LLM-generated text detectors may produce false positives under unseen domains, generators, or writing styles. SSD should therefore be used as decision-support evidence rather than as the sole basis for high-stakes decisions, such as academic misconduct judgments. Additional validation should be conducted for the intended deployment setting.

REPRODUCIBILITY STATEMENT

Section 2 specifies the model architecture and training objectives. Section 3 and Appendices A–D provide the datasets, evaluation protocol, implementation details, hyperparameters, inference cost, and attack construction. An anonymized implementation and experimental configurations are included in the supplementary material and are also available at `https://anonymous.4open.science/r/SSD-BC84/`.

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 *International Conference on Learning Representations*, pp. 24814–24836, 2024. URL `https://proceedings.iclr.cc/paper_files/paper/2024/file/6b8c6f846c3575e1d1ad496abea28826-Paper-Conference.pdf`.

Lele Cao. A practical synthesis of detecting ai-generated textual, visual, and audio content. In *2025 IEEE International Conference on Data Mining Workshops (ICDMW)*, pp. 1429–1436, 2025. doi: 10.1109/ICDMW69685.2025.00170.

Jiaqi Chen, Xiaoye Zhu, Tianyang Liu, Ying Chen, Chen Xinhui, Yiwen Yuan, Chak Tou Leong, Zuchao Li, Long Tang, Lei Zhang, Chenyu Yan, Guanghao Mei, Jie Zhang, and Lefei Zhang. Imitate before detect: Aligning machine stylistic preference for machine-revised text detection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(22):23559–23567, 2025. doi: 10.1609/aaai.v39i22.34525.

Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. Free dolly: Introducing the world's first truly open instruction-tuned llm, April 2023. URL `https://www.databricks.com/blog/2023/04/12/dolly-first-open-commercially-viable-instruction-tuned-llm`.

Geoffrey M. Currie. Academic integrity and artificial intelligence: is ChatGPT hype, hero or heresy? *Seminars in Nuclear Medicine*, 53(5):719–730, 2023.

DeepSeek-AI. DeepSeek-V4: Towards highly efficient million-token context intelligence, 2026. URL `https://arxiv.org/abs/2606.19348`.

Jesse Dodge, Maarten Sap, Ana Marasović, William Agnew, Gabriel Ilharco, Dirk Groeneveld, Margaret Mitchell, and Matt Gardner. Documenting large webtext corpora: A case study on the colossal clean crawled corpus. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 1286–1305, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.98. URL `https://aclanthology.org/2021.emnlp-main.98/`.

Rui Guo, Weibin Zeng, Fuzhang Wu, Yan Kong, Sicheng Shen, Yanjun Wu, and Weiming Dong. HLD: Approximate hierarchical linguistic distribution modeling for LLM-generated text detection. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=l9mqzHROGu.

Wei Hao, Ran Li, Weiliang Zhao, Junfeng Yang, and Chengzhi Mao. Learning to rewrite: Generalized llm-generated text detection. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 6421–6434. Association for Computational Linguistics, July 2025. doi: 10.18653/v1/2025.acl-long.322. URL https://aclanthology.org/2025.acl-long.322/.

Yongxin He, Shan Zhang, Yixuan Cao, Lei Ma, and Ping Luo. Detree: Detecting human-ai collaborative texts via tree-structured hierarchical representation learning. In *Advances in Neural Information Processing Systems*, volume 38, pp. 146060–146099, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/d6f8517fceeca1e2cd61721dff786c14-Paper-Conference.pdf.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8): 1735–1780, 1997.

Beizhe Hu, Qiang Sheng, Juan Cao, Yang Li, and Danding Wang. Llm-generated fake news induces truth decay in news ecosystem: A case study on neural news recommendation. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 435–445. Association for Computing Machinery, 2025. doi: 10.1145/3726302.3730027. URL https://doi.org/10.1145/3726302.3730027.

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 *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 8305–8316, Albuquerque, New Mexico, April 2025. doi: 10.18653/v1/2025.findings-naacl.464. URL https://aclanthology.org/2025.findings-naacl.464/.

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, 2019. URL https://arxiv.org/abs/1907.11692.

Zhichen Liu, Kaitong Qin, Linhan He, and Yang Xu. WaveDetect: Robust framework for machine-generated text detection via wavelet transform. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 8712–8727, San Diego, California, United States, July 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.findings-acl.424. URL https://aclanthology.org/2026.findings-acl.424/.

Guoxin Ma, Xiaoming Liu, Zhanhan Zhang, Chengzhengxu Li, Shengchao Liu, and Yu Lan. Deer: Disentangled mixture of experts with instance-adaptive routing for generalizable machine-generated text detection, 2025. URL https://arxiv.org/abs/2511.01192.

Shixuan Ma, Jiahao Li, Zhendong Mao, and Quan Wang. Zero-shot detection of LLM-generated text using temperature sensitivity. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 37664–37679, San Diego, California, United States, July 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.acl-long.1748. URL https://aclanthology.org/2026.acl-long.1748/.

Chengzhi Mao, Carl Vondrick, Hao Wang, and Junfeng Yang. Raidar: generative ai detection via rewriting. In *International Conference on Learning Representations*, pp. 6013–6030, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/1888b9df34b0d9eea6e009a1fdd55c4f-Paper-Conference.pdf.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D. Manning, and Chelsea Finn. Detectgpt: Zero-shot machine-generated text detection using probability curvature. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pp. 24950–24962. PMLR, Jul 2023. URL https://proceedings.mlr.press/v202/mitchell23a.html.

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng Xin Yong, Hailey Schoelkopf, Xiangru Tang, Dragomir Radev, Alham Fikri Aji, Khalid Almubarak, Samuel Albanie, Zaid Alyafeai, Albert Webson, Edward Raff, and Colin Raffel. Crosslingual generalization through multitask finetuning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15991–16111. Association for Computational Linguistics, July 2023. doi: 10.18653/v1/2023.acl-long.891. URL https://aclanthology.org/2023.acl-long.891/.

OpenAI. GPT-5 system card, August 2025. URL https://openai.com/index/gpt-5-system-card/.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. Stanza: A Python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 2020.

Qwen Team. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Michael Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *The Semantic Web*, pp. 593–607. Springer International Publishing, 2018. doi: 10.1007/978-3-319-93417-4_38. URL https://doi.org/10.1007/978-3-319-93417-4_38.

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*, pp. 494–502, 2024. doi: 10.24963/ijcai.2024/55.

Kevin Stowe and Kailash Patil. Spotlights and blindspots: Evaluating machine-generated text detection. In *Proceedings of the Fifteenth Language Resources and Evaluation Conference*, pp. 4173–4187, Palma de Mallorca, Spain, May 2026. ELRA Language Resource Association. doi: 10.63317/3mzmq2zqni35. URL https://aclanthology.org/2026.lrec-1.329/.

Yuchuan Tian, Hanting Chen, Xutao Wang, Zheyuan Bai, Qinghua Zhang, Ruifeng Li, Chao Xu, and Yunhe Wang. Multiscale positive-unlabeled detection of ai-generated texts. In *International Conference on Learning Representations*, pp. 34849–34866, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/95ab5c3e26fd82c7de3230bbad087d2d-Paper-Conference.pdf.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008. URL https://www.jmlr.org/papers/v9/vandermaaten08a.html.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pp. 5998–6008, 2017.

Saranya Venkatraman, Adaku Uchendu, and Dongwon Lee. GPT-who: An information density-based machine-generated text detector. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pp. 103–115, Mexico City, Mexico, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-naacl.8. URL https://aclanthology.org/2024.findings-naacl.8/.

Vivek Verma, Eve Fleisig, Nicholas Tomlin, and Dan Klein. Ghostbuster: Detecting text ghostwritten by large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 1702–1717. Association for Computational Linguistics, June 2024. doi: 10.18653/v1/2024.naacl-long.95. URL https://aclanthology.org/2024.naacl-long.95/.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Toru Sasaki, Thomas Arnold, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. M4: Multi-generator, multi-domain, and multi-lingual black-box machine-generated text detection. In *Proceedings of the 18th*

Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 1369–1407. Association for Computational Linguistics, March 2024. doi: 10.18653/v1/2024.eacl-long.83. URL https://aclanthology.org/2024.eacl-long.83/.

Zhao Wang, Max Xiong, Jianxun Lian, and Zhicheng Dou. Reasoning-aware aigc detection via alignment and reinforcement, 2026. URL https://arxiv.org/abs/2604.19172.

Yihuai Xu, Yongwei Wang, Yifei Bi, Huangsen Cao, Zhouhan Lin, Yu Zhao, and Fei Wu. Training-free llm-generated text detection by mining token probability sequences. In *International Conference on Learning Representations*, pp. 19072–19098, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/2fcd27d2806a6d4f616cb4e6084d74f3-Paper-Conference.pdf.

Zhenyu Xu and Victor S. Sheng. Detecting AI-generated code assignments using perplexity of large language models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(21): 23155–23162, 2024. doi: 10.1609/aaai.v38i21.30361. URL https://doi.org/10.1609/aaai.v38i21.30361.

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*, 2024. URL https://openreview.net/forum?id=Xlayxj2fWp.

Xiao Yu, Kejiang Chen, Qi Yang, Weiming Zhang, and Nenghai Yu. Text fluoroscopy: Detecting llm-generated text through intrinsic features. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 15838–15846, Miami, Florida, USA, 2024. doi: 10.18653/v1/2024.emnlp-main.885. URL https://aclanthology.org/2024.emnlp-main.885/.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. Defending against neural fake news. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/3e9f0fc9b2f89e043bc6233994dfcf76-Paper.pdf.

Yizhou Zhang, Karishma Sharma, Lun Du, and Yan Liu. Toward mitigating misinformation and social media manipulation in llm era. In *Companion Proceedings of the ACM Web Conference 2024*, pp. 1302–1305. Association for Computing Machinery, 2024. doi: 10.1145/3589335.3641256. URL https://doi.org/10.1145/3589335.3641256.

Hongyi Zhou, Jin Zhu, Pingfan Su, Kai Ye, Ying Yang, Shakeel A O B Gavioli-Akilagun, and Chengchun Shi. Adadetectgpt: Adaptive detection of llm-generated text with statistical guarantees. In *The Thirty-Ninth Annual Conference on Neural Information Processing Systems*, 2025.

Hongyi Zhou, Jin Zhu, Kai Ye, Ying Yang, Erhan Xu, and Chengchun Shi. Learn-to-distance: Distance learning for detecting llm-generated text. In *The Fourteenth International Conference on Learning Representations*, 2026.

A Datasets

Table 7: Comparison between the L2R and M4 datasets.

	L2R	**M4**
Usage	ID train/eval	OOD eval
Domains	21 domains	Multiple domains
Generators	4 LLMs	6 LLMs
Distribution	In-domain	Cross-dataset shift

- The L2R (Hao et al., 2025) dataset is a diversified benchmark constructed to evaluate the robustness and generalization of detectors for large language model (LLM)-generated texts. It contains human-written texts from 21 distinct domains (e.g., Finance, Entertainment,

Food), with 200 complete paragraphs randomly selected from each domain, averaging 120 words per sample. For each human-written sample, corresponding LLM-generated texts are produced using four state-of-the-art models, resulting in a total of 21,000 text samples. To better simulate real-world scenarios and improve the model's adaptability to different prompts, 200 diverse rewriting prompts are employed for LLM text generation instead of a single static prompt. All human-written texts are collected prior to the release date of ChatGPT (November 30, 2022) to avoid contamination from LLM-generated content.

- The M4 (Wang et al., 2024) dataset is designed to evaluate machine-generated text detection in black-box settings. Its English portion covers five domains—Wikipedia, WikiHow, Reddit, arXiv, and PeerRead—and includes texts generated by multiple language models, such as Dolly-v2 (Conover et al., 2023) and BLOOMz 176B (Muennighoff et al., 2023), together with corresponding human-written texts. The combination of multiple domains and generators makes M4 a challenging benchmark for evaluating detector generalization. In this work, M4 is used exclusively for out-of-domain evaluation; no M4 samples are used for training, hyperparameter tuning, checkpoint selection, or any other stage of model development.
- The Colossal Clean Crawled Corpus (C4) (Dodge et al., 2021) is a large English web-text corpus constructed by applying heuristic filters to a snapshot of Common Crawl. We use an unlabeled subset of 300,000 C4 texts solely to pre-train the sequential and structural encoders. Specifically, the two encoders are trained independently to reconstruct the frozen RoBERTa token representations at randomly masked positions from their respective views. C4 is not used to train the downstream detector and is excluded from all ID and OOD evaluations; the task-specific training and evaluation data remain L2R and M4, respectively.

B Experimental Settings

We use RoBERTa-base[1] to obtain 768-dimensional contextual representations and truncate each input to a maximum of 256 subwords. Dependency graphs are constructed with spaCy[2] and contain 45 base dependency-relation types; each relation is paired with its propagation direction in the RGCN. RoBERTa subwords aligned with the same spaCy word are mean-pooled, so both encoders operate on the same word-level sequence and graph nodes; special and padding tokens are excluded. The sequential encoder is a two-layer Transformer with a hidden size of 768, while the structural encoder is a two-layer RGCN with a hidden size of 768. Both encoders project their outputs into the same 768-dimensional teacher space before the reconstruction loss, and these projections are retained downstream. Before multi-head self-attention, the concatenated features are linearly projected from 1,537 to 768 dimensions; the attention module uses four heads and a dropout rate of 0.6. We use frozen Qwen3-4B to obtain one mean subword log-probability score per word and concatenate this one-dimensional feature before the input projection.

Before task-specific training, the sequential and structural encoders are independently pretrained on the unlabeled 300,000-text C4 subset described in Appendix A. For each view, 30% of non-special words are randomly selected, all associated RoBERTa subwords are masked, and the encoder reconstructs the corresponding frozen word-level RoBERTa representations. For the structural encoder, the dependency graph parsed from the original text is retained as a fixed topology during node-feature masking. The reconstruction objective is the mean-squared error plus 0.1 times the cosine-distance loss. Each encoder is pretrained for 10 epochs with AdamW, a batch size of 32, a learning rate of $1\mathrm{e}{-4}$, a weight decay of 0.01, a dropout rate of 0.1, and gradient clipping at 1.0. RoBERTa remains frozen throughout this stage.

For downstream detection, both pretrained encoders are frozen and kept in evaluation mode; only the adaptive fusion, input projection, multi-head attention, and classification layers are optimized. We train these layers for 30 epochs using AdamW with a batch size of 256, an initial learning rate of $1\mathrm{e}{-5}$, and a weight decay of $1\mathrm{e}{-4}$. We apply 5% linear warmup followed by cosine decay to $1\mathrm{e}{-6}$, label smoothing of 0.05, and gradient clipping at 1.0. The supervised objective is cross-entropy only; the sentence-level discrepancy statistic $\overline{D}$ is recorded for analysis but is not added to the training loss. We use mixed-precision training and memory-mapped dataset loading. Unless otherwise noted, SSD

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

[2] https://spacy.io/

results are averaged over three runs with seeds 13, 42, and 87; all hyperparameters and checkpoints are selected using L2R only. The downstream model is trained on a single 24 GB RTX 3090 GPU under Linux, and no M4 data are accessed during training or model selection.

C Inference Cost

Although SSD requires an additional pass through a frozen proxy language model to obtain P, it does not involve autoregressive generation or repeated text rewriting. We therefore compare its end-to-end per-text inference time with that of competing detectors. All methods are run on the same hardware described in Appendix B, and latency is averaged per text over the same evaluation set under a consistent inference configuration.

Table 8: Average per-text inference time in seconds. Lower is faster.

	Train-free			Training-based					
Metric	**Fast-DetectGPT**	**RAIDAR**	**Lastde**	**Ghost-buster**	**L2R**	**Ada-DetectGPT**	**ImBD**	**L2D**	**SSD**
Inference time (s)	<u>0.52</u>	10.45	0.58	0.74	10.73	**0.47**	1.12	11.63	1.03

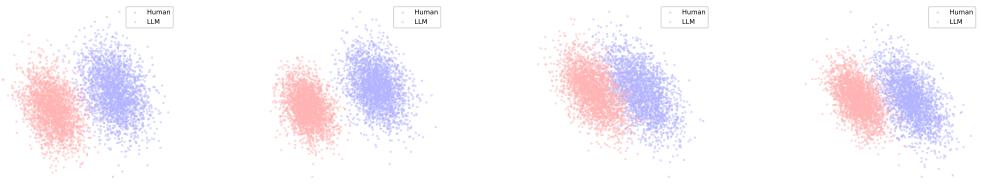
As shown in Table 8, SSD processes each text in 1.03 seconds. Its additional proxy-model pass introduces moderate overhead relative to lightweight probability-based detectors, but its latency remains comparable to other discriminative methods and is slightly lower than ImBD (1.12 seconds). More importantly, SSD is over ten times faster than RAIDAR, L2R, and L2D, which require repeated rewriting or comparison. SSD therefore retains practical inference efficiency without incurring the high latency of generation-intensive detectors.

D Adversarial Attack Construction

Following Hao et al. (2025), we consider two complementary attacks. The *Decoherence Attack* (Bao et al., 2024) transposes two adjacent, randomly selected words in each LLM-generated sentence longer than 20 words, perturbing local fluency while largely preserving meaning. The *Rewrite Attack*, introduced by Mao et al. (2024), prompts DeepSeek-V4-Pro (DeepSeek-AI, 2026) to rewrite LLM-generated paragraphs, thereby shifting their surface realization toward human writing. In both settings, detectors are trained on the original training set and evaluated only on the perturbed test set.

E Representation Visualization

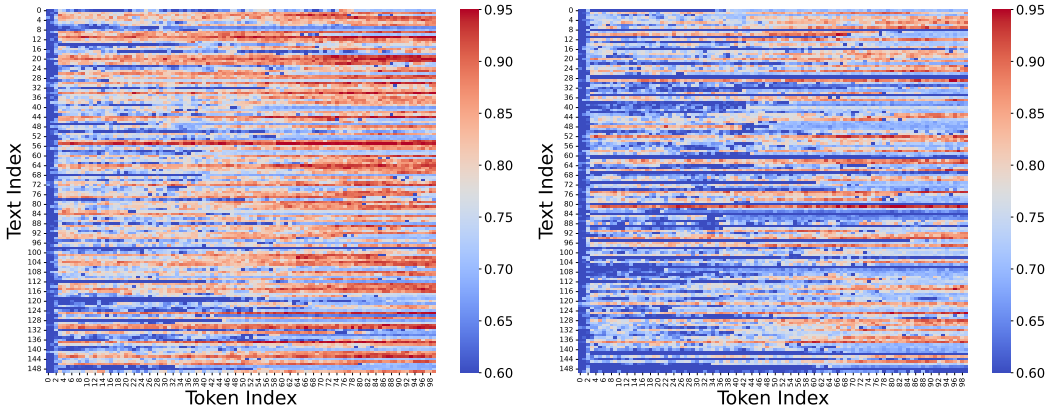
We visualize sentence representations for 5,000 randomly sampled texts (2,500 human-written and 2,500 LLM-generated) from L2R and M4 using t-SNE (van der Maaten & Hinton, 2008), comparing the full model with an ablation that removes discrepancy D.



(a) L2R, w/o D (b) L2R, full model (c) M4, w/o D (d) M4, full model

Figure 4: t-SNE visualization of sentence representations with and without discrepancy D.

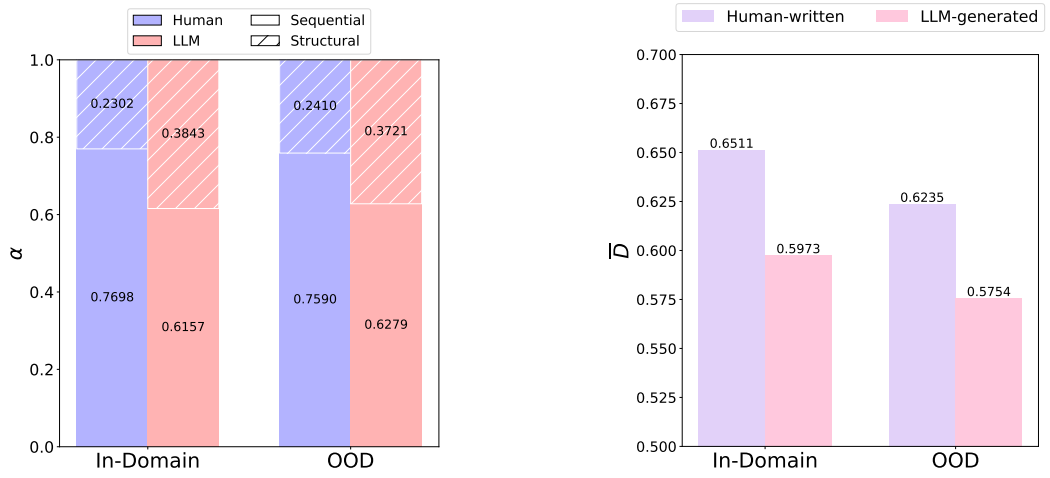
The full model produces more clearly separated human and LLM clusters on both datasets, whereas representations without D exhibit substantially greater overlap. The visualization provides qualitative support for the quantitative ablations in Table 5.



(a) Heatmap of α for human-written text

(b) Heatmap of α for LLM-generated text

Figure 5: Token-level visualization of adaptive fusion weights α based on 150 randomly sampled human texts from OOD data and their LLM-generated counterparts. Human texts exhibit wider high-value regions (red), while LLM texts show denser low-value regions (blue).



(a) Adaptive fusion weights α.

(b) Average Discrepancy $\overline{D}$.

Figure 6: Quantitative analysis of (a) adaptive fusion weights α and (b) average discrepancy $\overline{D}$. Human texts consistently show higher α and $\overline{D}$ across domains.

F Discrepancy Analysis

To understand how the model distinguishes human-written and LLM-generated texts, we visualize token-level fusion weights α (Figure 5) based on 150 randomly sampled human texts from OOD data and their paired LLM-generated counterparts, along with sentence-level discrepancy (Figure 6).

Figure 5 visualizes the token-level fusion weights α. Human-written texts exhibit broader regions of high α values (red), whereas LLM-generated texts contain denser regions of low α values (blue). Figure 6(a) quantifies this pattern: the average sequential-view weight is 0.7698/0.7590 for human-written texts on L2R/M4, compared with 0.6157/0.6279 for LLM-generated texts. This difference is consistent with our motivation. Human-written texts tend to produce greater cross-view variation, for which the learned gate assigns relatively more weight to the sequential representation; the more regular cross-view patterns of LLM-generated texts are associated with a larger relative contribution from the structural representation.

The pattern underlying this weighting difference is shown in Figure 6b, which plots the sentence-level average discrepancy $\overline{D}$ (computed as $\frac{1}{L}\sum_i d_i$ over token-level Euclidean distances). Human texts

consistently show higher $\overline{D}$ than LLM texts. This gap remains stable across domains, supporting $\overline{D}$ as an empirically stable cross-domain signal for machine-generated text.

In summary, Figure 5 and Figure 6 provide complementary evidence. Figure 6b establishes $\overline{D}$ as a consistent cross-view signal associated with text provenance, while Figure 5 and Figure 6a show that the gating mechanism α leverages this signal through adaptive feature reweighting. Together, these observations help explain the model's strong cross-domain generalization.

G Additional Parsing-Robustness Details

For the parser comparison in Figure 3, we replace spaCy with Stanza (Qi et al., 2020) and otherwise keep the architecture, data split, and optimization protocol unchanged. For controlled-error evaluation, we construct two perturbations while preserving the number of graph nodes. **Relation-type noise** swaps the dependency labels (e.g., `nsubj` and `dobj`) assigned to randomly selected edges. **Attachment noise** swaps the parent nodes of selected tokens, altering graph connectivity while retaining one incoming attachment per affected token. The perturbation rate denotes the percentage of non-root nodes affected, and is varied from 0% to 25%. Noisy parses are used only at test time; the model is trained with the original spaCy graphs. This protocol separates robustness to parser replacement from robustness to local graph errors.

H Domain-Specific Robustness Analysis under Adversarial Attacks

We present the fine-grained detection performance of all compared methods under two adversarial attack strategies across the 21 domains. Table 9 and Table 10 correspond to the *Decoherence Attack* and the *Rewrite Attack* discussed in Section 3.7, respectively.

Decoherence Attack As shown in Table 9, the Decoherence Attack degrades most baselines substantially. For example, Ghostbuster drops from 70.94% to 48.59% in average AUROC, and Fast-DetectGPT falls to nearly 50%. In contrast, SSD achieves the highest average AUROC of 92.60%, outperforming the second-best method, L2D, which attains 90.71%. SSD also achieves the lowest standard deviation of 4.38%. The advantage is especially clear in stylistically distinct domains such as *Code* and *ArtCulture*.

Rewrite Attack Table 10 shows a similar trend under the Rewrite Attack. SSD achieves the highest average AUROC of 88.55% (std 4.31%), followed by L2D at 87.91%. SSD achieves top performance in several domains, including *Code* (91.68%) and *OnlineContent* (92.52%), despite their different writing conventions.

In summary, the results in Table 9 and Table 10 show that by explicitly modeling the sequential–structural discrepancy, SSD remains effective under attacks that perturb local fluency or rewrite surface phrasing, supporting the cross-domain and adversarial robustness of this signal.

Table 9: Decoherence Attack: per-domain AUROC scores (%) on L2R. SSD achieves the highest average performance (92.60%) and the lowest standard deviation (4.38).

	Train-free			Training-based					
Domain	**Fast-DetectGPT**	**RAIDAR**	**Lastde**	**Ghost-buster**	**L2R**	**Ada-DetectGPT**	**ImBD**	**L2D**	**SSD**
AcademicResearch	32.00	82.13	48.34	42.91	81.30	50.76	84.12	90.73	**91.44**
ArtCulture	44.59	76.11	64.33	44.31	80.74	51.21	69.34	84.12	**91.32**
Business	50.65	81.05	72.55	61.88	88.15	48.25	79.71	86.30	**91.65**
Code	49.66	36.65	71.71	15.67	81.13	54.69	71.39	86.48	**96.06**
EducationMaterial	57.12	**95.63**	76.43	63.05	93.21	51.32	92.43	91.66	93.56
Entertainment	68.05	81.56	86.02	64.95	91.92	48.15	91.63	93.85	**95.64**
Environmental	65.18	90.73	85.65	61.67	95.01	53.25	87.85	94.98	**95.87**
Finance	50.72	79.52	72.30	55.87	90.78	45.96	87.57	94.16	**95.76**
FoodCuisine	60.13	76.19	78.16	45.87	93.60	54.04	82.22	**94.07**	92.48
GovernmentPublic	53.35	75.68	71.87	46.56	82.79	49.97	80.64	87.47	**91.46**
LegalDocument	66.15	61.92	87.79	31.32	74.59	47.85	90.97	**93.39**	92.55
LiteratureCreativeWriting	61.04	76.67	82.28	73.70	88.80	61.30	90.51	93.72	**96.87**
MedicalText	38.89	76.30	78.32	42.57	74.82	51.03	67.92	76.44	**81.36**
NewsArticle	42.08	82.13	60.44	46.76	89.75	59.67	83.38	93.25	**93.53**
OnlineContent	45.36	81.38	64.23	37.68	86.77	50.17	79.02	94.08	**95.72**
PersonalCommunication	37.19	70.85	58.48	47.52	78.99	52.91	67.12	**86.92**	84.33
ProductReview	47.12	76.04	68.86	51.07	**94.49**	48.59	84.56	94.24	94.16
Religious	49.99	79.52	66.26	39.07	**95.89**	60.20	94.27	92.92	94.88
Sports	42.22	75.64	61.63	43.49	84.71	48.05	77.99	**88.01**	83.76
TechnicalWriting	42.56	81.72	64.21	50.36	89.18	48.57	86.03	91.97	**95.15**
TravelTourism	45.55	84.22	65.87	54.12	92.45	46.36	85.06	96.08	**97.07**
AVERAGE	49.98	77.22	70.75	48.59	87.10	51.54	82.56	90.71	**92.60**
STD	9.93	11.47	10.12	12.78	6.56	4.39	8.14	4.75	**4.38**

Table 10: Rewrite Attack: per-domain AUROC scores (%) on L2R. SSD achieves the highest average performance (88.55%) and the lowest standard deviation (4.31).

	Train-free			Training-based					
Domain	**Fast-DetectGPT**	**RAIDAR**	**Lastde**	**Ghost-buster**	**L2R**	**Ada-DetectGPT**	**ImBD**	**L2D**	**SSD**
AcademicResearch	33.21	76.08	58.50	33.90	79.63	48.77	80.44	**87.59**	86.34
ArtCulture	47.73	65.05	73.15	36.42	78.89	49.41	65.27	80.75	**87.71**
Business	51.08	78.12	79.64	54.38	87.02	46.52	77.22	83.86	**87.69**
Code	51.02	31.70	78.54	19.17	80.61	54.83	70.25	85.13	**91.68**
EducationMaterial	57.83	88.06	83.71	54.43	**91.64**	49.41	88.98	88.58	88.94
Entertainment	69.76	71.28	**95.74**	57.94	91.17	46.51	89.98	92.09	91.46
Environmental	65.04	72.55	**94.24**	51.85	94.03	51.13	85.69	92.89	92.31
Finance	50.18	63.26	80.67	46.11	89.55	43.84	84.86	**91.61**	90.86
FoodCuisine	60.49	75.09	86.90	38.10	**92.77**	52.27	80.39	92.41	87.87
GovernmentPublic	54.40	62.85	78.87	36.59	82.21	47.84	79.36	85.78	**88.28**
LegalDocument	69.25	59.87	**96.21**	24.47	74.29	46.23	90.31	92.23	89.04
LiteratureCreativeWriting	61.51	59.72	89.56	64.14	87.47	59.26	87.59	90.83	**92.29**
MedicalText	38.09	71.80	**86.89**	32.67	72.09	48.92	61.91	71.46	78.28
NewsArticle	44.46	70.76	68.65	36.54	88.97	57.49	81.66	**91.50**	88.91
OnlineContent	45.26	68.72	72.52	27.88	84.10	48.05	73.15	89.53	**92.52**
PersonalCommunication	38.35	67.10	66.32	38.19	76.43	50.86	61.49	**82.27**	80.00
ProductReview	47.49	58.78	77.39	43.29	**93.14**	46.82	81.59	91.67	91.34
Religious	50.95	75.16	76.28	31.97	**94.45**	58.55	91.10	90.32	91.08
Sports	45.79	68.54	70.59	35.95	83.24	46.31	74.75	**85.19**	79.40
TechnicalWriting	42.00	79.68	74.19	41.38	87.23	46.60	81.74	88.05	**91.54**
TravelTourism	45.35	78.19	74.50	45.09	90.30	44.38	80.33	**92.28**	91.92
AVERAGE	50.92	68.68	79.19	40.50	85.68	49.71	79.43	87.91	**88.55**
STD	10.06	11.30	9.89	11.33	6.71	4.47	8.92	5.24	**4.31**