Verifiable LLM-Generated Text Detection via Projected Semantic-Structural Distributions

Ruochong Xiong[1], Qien Li[1], Wangwang Lian[1], Yulong Wan[1], Hanlin Xue[1], Zhouxing Tan[1], Han Yang[2], Fengyu Lu[3]*, Junfei Liu[1]

[1]National Engineering Research Center for Software Engineering, Peking University, China
[2]Zeekr, Geely Auto, China
[3]Business School, Central South University, China

xiongrc@stu.pku.edu.cn, fengyul@csu.edu.cn, {liqien,liujunfei}@pku.edu.cn

Abstract

The widespread deployment of large language models (LLMs) makes detecting LLM-Generated text a critical security task. Existing methods, primarily relying on output probabilities from proxy models or single semantic features, suffer from distribution misalignment and limited interpretability. We observe that machine-generated text exhibits a directionally consistent systematic translation relative to human-written text within the joint semantic-structural space. Accordingly, we propose ProSSD, a statistical framework utilizing supervised subspace learning to extract compact features and construct conditional semantic distributions based on syntactic structures. By employing a likelihood ratio test, we derive a modified Mahalanobis distance, weighted by the Wasserstein distance, as the discriminative metric. Experiments demonstrate ProSSD's superior robustness and computational efficiency across cross-domain, cross-model, and adversarial scenarios. Furthermore, we reveal the phenomena of systematic semantic translation and semantic collapse in machine-generated text, offering interpretable statistical insights into LLM generation behaviors. [1]

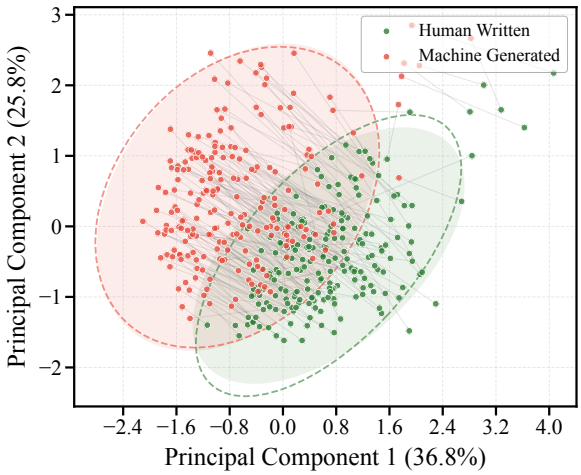


Figure 1: Average semantic positions of HWT and MGT in 2D space. MGT exhibits a systematic directional shift relative to HWT across different syntactic structures.

1 Introduction

The rapid advancement of large language models (LLMs) has significantly enhanced the efficiency of various text processing tasks (Demszky et al., 2023; Doshi and Hauser, 2024). However, this progress introduces severe challenges, notably the mass generation of fake news (Ahmed et al., 2021; Hu et al., 2025), academic fabrication (Koike et al., 2024), copyright infringement (Liu et al., 2024), and the contamination of web corpora. These issues not only precipitate widespread trust crises but also threaten the integrity of information ecosystems and human creativity (Lee et al., 2024). Consequently, the detection and governance of machine-generated text (MGT) have become urgent priorities. There is a pressing need for efficient, accurate, and interpretable detection methods to provide robust technical (Wu et al., 2025a) support for forensic analysis, academic integrity, and content moderation.

To address these challenges, extensive research has been conducted, as detailed in Section 2. As a prevailing paradigm, training-free statistical detection methods (Crothers et al., 2023) distinguishes texts by exploiting the tendency of LLMs to select high-probability tokens, utilizing metrics such as perplexity (Solaiman et al., 2019) and logits Curvature (Mitchell et al., 2023). However, these methods face significant practical limitations. First, distributional discrepancies in text generation are not statistically significant across all contexts (Jiang et al., 2025). Second, the closed-source nature of commercial models (e.g., GPT (OpenAI, 2025), Gemini (Google DeepMind, 2025)) forces reliance

*Corresponding author.

[1]Data and code are publicly available at: https://github.com/RuoChoXio/ProSSD

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 14005–14042
July 2-7, 2026

on open-source proxy models for distribution approximation (Zhou et al., 2025). This inevitably introduces distribution misalignment and incurs high training and inference costs, hindering large-scale real-time deployment. Furthermore, these methods lack intrinsic interpretability. Relying on model output probabilities, they fail to offer transparent computational processes or a step-by-step verifiable chain of evidence.

A more fundamental limitation is that MGT achieves semantic fluency comparable to human-written text (HWT), logit-based metrics relying on token-level confidence fail to define clear decision boundaries using such single semantic features (Tang et al., 2025). Recent studies further suggest that despite simulating human-like surface semantics, LLMs exhibit relatively constrained syntax-contextualized semantic expressions (Durward and Thomson, 2024). Inspired by this, we extend our perspective from a single semantic dimension to the joint semantic-structural distribution, aiming to quantify the intrinsic differences between HWT and MGT via conditional semantic statistics in a low-dimensional projected subspace.

This shift is driven by a key geometric regularity observed in the projected space. As shown in Figure 1, we calculate the semantic centroids for various local syntactic configurations (e.g., "noun-verb" pairs). The gray mapping lines connecting the centroids of HWT and MGT under identical syntactic structures reveal a striking phenomenon: while distinct syntactic configurations scatter across the feature space, **the semantic centroids of MGT consistently exhibit a Systematic Translation relative to those of HWT**. This implies a systematic bias across syntactic structures during the LLM decoding process, not a local perturbation specific to certain parts of speech, but a global characteristic inherent to the generation mechanism. For a more detailed analysis of systematic semantic translation and the phenomenon of semantic collapse, please refer to Appendix B.

Building upon these geometric insights, we propose the **Pro**jected **S**emantic-**S**tructural **D**istributions (**ProSSD**) framework. Diverging from methods dependent on internal model states or raw output probabilities, ProSSD establishes a transparent, step-by-step verifiable detection paradigm. The framework proceeds in three distinct steps: first, utilizing supervised subspace learning to extract dense, low-dimensional aggregated semantic features. Second, modeling semantic distributions conditioned on syntactic structures to quantify the intrinsic divergence between HWT and MGT. And finally, deriving the modified Mahalanobis distance via the likelihood ratio test as the core metric, dynamically weighted by the Wasserstein distance to optimize discrimination. Our main contributions are summarized as follows:

(1) **We propose ProSSD, a novel statistical detection paradigm**. Utilizing supervised subspace projection and joint semantic-structural distribution modeling, we construct a discriminant statistic based on the Wasserstein distance-weighted likelihood ratio test. This design suppresses high-dimensional noise, transforming subtle artifacts into distinguishable statistical features.

(2) **We achieve SOTA detection performance with high efficiency.** On the DetectRL benchmark covering advanced LLMs such as GPT-5.1, ProSSD excels in cross-model, cross-domain, and adversarial settings. Our approach yields robust results even with minimal samples and reduces inference costs by orders of magnitude, demonstrating significant deployment value.

(3) **We establish a step-by-step verifiable interpretability framework.** Supported by rigorous derivations, we provide theoretical guarantees and enhance transparency via statistical evidence. Furthermore, we uncover systematic semantic translation and semantic collapse in MGT under syntactic constraints, offering fresh insights into generative mechanisms.

2 Related work

Current machine-generated text detection methods are primarily categorized into supervised classifiers, statistical detection methods, and auxiliary retrieval or watermarking based techniques. As retrieval and watermarking approaches rely on external reference corpora or active injection rather than intrinsic distributional features, we discuss them in Appendix A.

Supervised training-based methods formulate detection as a binary classification task, heavily relying on labeled data (Zellers et al., 2019; Fagni et al., 2020). Early research predominantly utilized shallow linguistic features (e.g., TF-IDF) combined with traditional classifiers (Solaiman et al., 2019), subsequently transitioning to fine-tuning pre-trained models to enhance performance (Solaiman et al., 2019; Ippolito et al., 2020). To

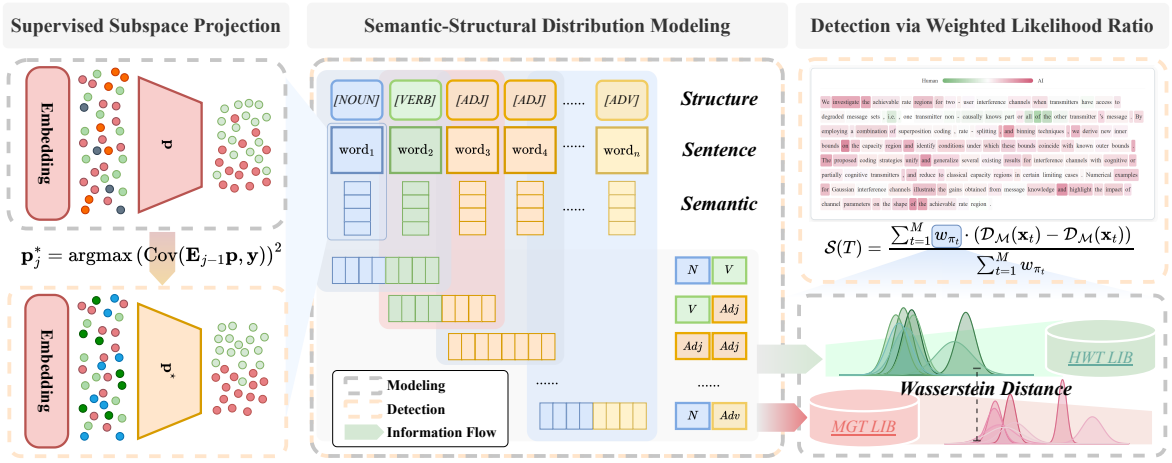


Figure 2: **The architecture of ProSSD.** It extracts features via Supervised Subspace Projection (Left), models Semantic-Structural Distributions (Middle), and calculates the final score through Detection via Weighted Likelihood Ratio (Right).

improve robustness against domain shifts and adversarial attacks, recent works introduce advanced optimization objectives: RADAR (Hu et al., 2023) leverages proximal policy optimization (PPO) (Schulman et al., 2017) to strengthen defense capabilities; ImBD (Chen et al., 2025a) employs direct preference optimization (DPO) (Rafailov et al., 2023) to align stylistic discrepancies between humans and machines; and DetectAnyLLM (Fu et al., 2025) proposes direct difference learning (DDL) to maximize distributional distances. Despite these advancements, supervised methods face two inherent limitations: severe domain dependency, which leads to significant performance degradation across models or topics (Sarvazyan et al., 2023), and high maintenance costs, as the rapid iteration of LLMs necessitates frequent classifier retraining that consumes substantial computational and data resources.

Statistical Detection methods aim to exploit distributional discrepancies between MGT and HWT using specific metrics. In terms of scalar metrics, early work utilized basic statistics such as entropy and log-probability (Gehrmann et al., 2019; Solaiman et al., 2019), later expanding to more comprehensive measures like log-rank ratio (LRR) (Su et al., 2023) , n-gram distributions (Yang et al., 2024), and intrinsic text dimension (Tulchinskii et al., 2023). Perturbation-based methods leverage probability curvature for discrimination; DetectGPT (Mitchell et al., 2023) hypothesizes that MGT tends to occupy negative curvature regions of the log-probability surface. To address high sampling costs, Fast-DetectGPT (Bao et al., 2024) employs conditional probability curvature for efficient approximation, while AdaDetectGPT (Zhou et al., 2025) introduces an adaptive witness function to provide theoretical guarantees. Regarding high-order features and representations, recent research explores deeper patterns: Binoculars (Hans et al., 2024) measures prediction surprise via perplexity ratios; RepreGuard (Chen et al., 2025b) analyzes hidden representations to capture activation differences; and GECScore (Wu et al., 2025b) performs detection from a syntactic perspective using grammatical error correction distance.

3 Methodology

3.1 Problem Definition

While LLMs generate text that is semantically indistinguishable from human writing on the surface, the generation process remains fundamentally constrained by the probabilistic decoding algorithms (Fröhling and Zubiaga, 2021), leading to statistical discrepancies in the underlying joint syntactic-semantic distribution. Drawing on the observations in Figure 1, we posit a core hypothesis: for a given local syntactic structure $\pi \in \Pi$, the conditional distributions of human-written text and machine-generated text within the semantic embedding space $\mathbf{e} \in \mathbb{R}^D$ diverge significantly. Formally, let $y \in \{H, M\}$ denote the class labels for HWT and MGT respectively, we hypothesize:

$$P(\mathbf{e}|\pi, y = H) \neq P(\mathbf{e}|\pi, y = M). \quad (1)$$

The objective of this paper is to enable machine-generated text detection without task-specific fine-

tuning by modeling and quantifying the divergence in this joint semantic-structural Distribution.

3.2 Supervised Subspace Projection

Directly estimating the aforementioned joint semantic-structural distribution using raw embeddings $\mathbf{e}$ from pre-trained models (e.g. RoBERTa (Liu et al., 2019)) presents significant challenges. First, the semantic masking effect obscures the subtle signals distinguishing HWT from MGT, as raw embeddings are overwhelmingly dominated by variance from general semantic topics irrelevant to the generation source (Nagata et al., 2023). Furthermore, high-dimensional sparsity poses a computational barrier; since the embedding dimension D far exceeds the local sample size, direct parametric estimation of the covariance matrix becomes numerically highly unstable (Ledoit and Wolf, 2004).

To address these challenges, we propose the supervised subspace learning algorithm. The goal is to learn a projection matrix $\mathbf{P} \in \mathbb{R}^{D\times k}(k \ll D)$ to map the raw vectors into low-dimensional compact semantic features $\mathbf{v} = \mathbf{P}^T\mathbf{e}$. Unlike principal component analysis (PCA), which aims to maximize total variance, our objective is to maximize the statistical correlation between features and class labels $\mathbf{y}$, thereby suppressing noise and preserving discriminative information. We model this process as a recursive optimization problem. Let $\mathbf{E}_0$ be the centered raw embedding matrix, and $\mathbf{E}_{j-1}$ be the residual feature matrix at step j. We solve for the j-th projection basis vector $\mathbf{p}_j$ via the following objective function:

$$\mathbf{p}_j^* = \underset{\mathbf{p}:\|\mathbf{p}\|_2=1}{\operatorname{argmax}} \left(\operatorname{Cov}(\mathbf{E}_{j-1}\mathbf{p}, \mathbf{y})\right)^2. \tag{2}$$

After obtaining the optimal direction $\mathbf{p}_j^*$, we perform feature deflation: $\mathbf{E}_j = \mathbf{E}_{j-1} - \mathbf{s}_j(\mathbf{p}_j^*)^T$, where $\mathbf{s}_j = \mathbf{E}_{j-1}\mathbf{p}_j^*$ is the projection score of the current dimension. This step ensures the informational orthogonality of projection directions across different dimensions; detailed derivation is provided in Appendix C.1. Finally, for the i-th word of the input text, we obtain its compact semantic vector $\mathbf{v}_i \in \mathbb{R}^k$.

3.3 Semantic-Structural Distribution Modeling

After obtaining the low-dimensional compact semantic features $\mathbf{v}$, we extend our perspective from single semantic point estimation to Joint semantic-structural distribution modeling.

Local Feature Construction. To capture semantic transition patterns within local syntactic environments, we define the meta-semantics vector $\mathbf{x}_t$ and meta-structure π_t. At time step t, we concatenate projected features from adjacent positions and pair them with corresponding pos tags to form observation pairs:

$$\mathbf{x}_t = \begin{bmatrix} \mathbf{v}_t \\ \mathbf{v}_{t+1} \end{bmatrix}, \quad \pi_t = (pos_t, pos_{t+1}). \tag{3}$$

Distribution Modeling. Consequently, any text sequence can be parsed into a set of local observations $\mathcal{D} = \{(\mathbf{x}_t, \pi_t)\}_{t=1}^{T-1}$.Based on the central limit theorem and empirical observations detailed in Appendix C.2, we assume that conditioned on the meta-structure π, the vector $\mathbf{x}$ follows a multivariate Gaussian distribution. That is, for class $y \in \{H, M\}$:

$$P(\mathbf{x}|\pi, y) = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_y^\pi, \boldsymbol{\Sigma}_y^\pi). \tag{4}$$

Here, $\boldsymbol{\mu}$ characterizes the average semantic information under this syntactic structure, while the block covariance matrix $\boldsymbol{\Sigma}$ encodes the semantic dependencies between adjacent tokens.

Parameter Estimation. We estimate the distribution parameters for HWT and MGT on a reference corpus using maximum likelihood estimation (MLE) for each appearing meta-structure $\pi \in \Pi$:

$$\begin{aligned} \hat{\boldsymbol{\mu}}_y^\pi &= \frac{1}{N_{\pi,y}} \sum_{j=1}^{N_{\pi,y}} \mathbf{x}_j, \\ \hat{\boldsymbol{\Sigma}}_y^\pi &= \frac{1}{N_{\pi,y}-1} \sum_{j=1}^{N_{\pi,y}} (\mathbf{x}_j - \hat{\boldsymbol{\mu}}_y^\pi)(\mathbf{x}_j - \hat{\boldsymbol{\mu}}_y^\pi)^T. \end{aligned} \tag{5}$$

Thus, the detection model is parameterized as a set of structured distributions $\mathcal{M} = \{(\mathcal{N}_H^\pi, \mathcal{N}_M^\pi) \mid \pi \in \Pi\}$.

Adaptive Weighting via Wasserstein Distance. Different syntactic structures carry significantly different amounts of discriminative information. To quantify this structural non-uniformity, we introduce the Wasserstein distance (Villani et al., 2008) as the discriminative weight w_π for the meta-structure π. For two Gaussian distributions $\mathcal{N}_H^\pi$ and $\mathcal{N}_M^\pi$, the Wasserstein distance has an analytical solution. As a core metric in optimal transport theory, the Wasserstein distance defines a strict geometric metric in the probability distribution space (Ge et al., 2021; Just et al., 2023). It quantifies intrinsic shifts between continuous distributions

more robustly than traditional divergence metrics. The specific form is as follows:

$$w_\pi = \Big(\|\boldsymbol{\mu}_H^\pi - \boldsymbol{\mu}_M^\pi\|_2^2 + \operatorname{Tr}\left(\boldsymbol{\Sigma}_H^\pi + \boldsymbol{\Sigma}_M^\pi - 2(\boldsymbol{\Sigma}_H^{\pi\frac{1}{2}}\boldsymbol{\Sigma}_M^\pi\boldsymbol{\Sigma}_H^{\pi\frac{1}{2}})^{\frac{1}{2}}\right)\Big)^{1/2}. \tag{6}$$

The derivation of Equation 6 is provided in Appendix C.3. We select the Wasserstein distance not only because it captures changes in both the first moment and second moment, but also because it theoretically bounds the performance of the discriminative function.

Theorem 1. ***(Wasserstein Discrimination Bound)*** *For any discriminative function f satisfying the K-Lipschitz continuity condition, the difference in expected scores on the HWT and MGT distributions is strictly bounded by the Wasserstein distance between these two distributions:*

$$|\mathbb{E}_{P_H}[f(\mathbf{x})] - \mathbb{E}_{P_M}[f(\mathbf{x})]| \leq K \cdot W_2(P_H, P_M). \tag{7}$$

The proof is provided in Appendix C.4.

This theorem indicates that the value of w_π directly reflects the theoretical possibility of distinguishing the two types of text under that structure. A larger W_2 distance implies a clearer decision boundary.

3.4 Detection via Weighted Likelihood Ratio

Grounded in the Neyman-Pearson lemma (Larsen and Marx, 2005), we formulate the detection problem as a statistical test based on the log-likelihood Ratio (LLR) for each local feature, followed by a weighted aggregation strategy to derive the final decision.

Local Statistic Derivation. Given a text T to be detected, we obtain the sequence $\{(\mathbf{x}_t, \pi_t)\}_{t=1}^M$ after projection, slicing, and feature construction. For each local slice t, we decide between the null hypothesis $H_0 : \mathbf{x}_t \sim P_H^{\pi_t}$ and the alternative hypothesis $H_1 : \mathbf{x}_t \sim P_M^{\pi_t}$. The optimal test statistic is the log-likelihood ratio:

$$s_t = \ln \frac{P(\mathbf{x}_t|\pi_t, \theta_M)}{P(\mathbf{x}_t|\pi_t, \theta_H)}. \tag{8}$$

Based on the multivariate Gaussian assumption mentioned above, substituting the density function reveals that the local anomaly score s_t is equivalent to the difference in the modified Mahalanobis distance:

$$s_t = \tfrac{1}{2}\left[\mathcal{D}_{\mathcal{M}}(\mathbf{x}_t; \boldsymbol{\mu}_H^{\pi_t}, \boldsymbol{\Sigma}_H^{\pi_t}) - \mathcal{D}_{\mathcal{M}}(\mathbf{x}_t; \boldsymbol{\mu}_M^{\pi_t}, \boldsymbol{\Sigma}_M^{\pi_t})\right], \tag{9}$$

where $\mathcal{D}_{\mathcal{M}}$ includes a penalty term for the covariance determinant:

$$\mathcal{D}_{\mathcal{M}}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = (\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}) + \ln|\boldsymbol{\Sigma}|. \tag{10}$$

The detailed derivation of this metric is provided in Appendix C.5. This equation indicates that if the sample deviates from the human distribution significantly more than from the machine distribution, s_t becomes positive, indicating MGT characteristics.

Global Aggregation. To obtain a document-level decision score and ensure interpretability, we map window-level scores s_t back to the word level. We define the anomaly score of the i-th word as the mean of adjacent windows: $o_i = \frac{1}{2}(s_{i-1} + s_i)$, and assuming boundary terms are 0. Finally, we use Wasserstein distance weights to aggregate all scores, obtaining the final detection statistic $\mathcal{S}(T)$ for text T:

$$\mathcal{S}(T) = \frac{\sum_{t=1}^M w_{\pi_t} \cdot s_t}{\sum_{t=1}^M w_{\pi_t}}. \tag{11}$$

If $\mathcal{S}(T)$ exceeds the preset threshold τ, the text is classified as MGT.

4 Experiments

4.1 Experimental Settings

Benchmark Datasets. To rigorously evaluate model performance in realistic and challenging detection scenarios, we adopt DetectRL (Wu et al., 2024) as the foundational evaluation benchmark. This benchmark covers four representative high-risk misuse domains: academic writing (ArXiv [2]), news summarization (XSum, (Narayan et al., 2018)), creative writing (Writing Prompts, (Fan et al., 2018)), and social media reviews (Yelp Review (Zhang et al., 2015)). Given that the generation models in the original benchmark (e.g., GPT-3.5 (OpenAI, 2023), PaLM-2 (Anil et al., 2023)) fail to capture the frontier capabilities of current LLMs in semantic alignment and reasoning, we upgrade the generation sources to verify detector effectiveness against SOTA capabilities. Strictly following the DetectRL data construction protocol, we employ GPT-5.1 (OpenAI, 2025), Claude-Sonnet-4 (Anthropic, 2025), and Gemini-3-Flash

[2] https://www.kaggle.com/datasets/spsayakpaul/arxiv-paper-abstracts/data

Table 1: Performance comparison across different target models and text domains. Results are reported in terms of AUROC (%) and F1 (%). The best and second-best results in each column are highlighted in **bold** and underlined.

Method	Models								Domains							
	GPT-5.1		Claude-S-4		Gemini-3-F		Grok-4.1		Arxiv		Writing		XSum		Review	
	AUR.	F1	AUR.	F1	AUR.	F1	AUR.	F1	AUR.	F1	AUR.	F1	AUR.	F1	AUR.	F1
LRR	68.02	58.08	79.62	71.13	63.30	59.10	62.18	49.50	73.90	67.71	48.20	14.90	70.14	63.49	84.01	75.11
NPR	78.37	68.41	81.40	72.97	80.13	70.64	77.02	68.92	92.45	85.08	67.51	64.77	72.78	61.37	85.03	76.18
RoBERTa	91.36	81.84	92.62	84.56	95.53	88.20	91.59	83.38	97.85	93.67	90.31	82.87	99.37	96.88	97.41	92.02
DetectGPT	59.54	47.60	56.77	38.53	50.84	37.05	57.73	43.34	58.89	61.42	77.39	71.00	17.83	66.69	73.87	67.54
Binoculars	75.55	66.59	92.52	84.66	86.27	77.37	71.40	61.43	75.84	65.29	77.51	69.19	74.90	65.37	97.35	91.64
Fast-DetectGPT	67.07	58.59	78.78	72.73	76.13	68.44	52.91	42.29	74.81	64.93	58.88	47.29	57.65	50.42	82.92	75.56
ImBD	86.95	80.23	94.37	86.67	96.83	90.90	88.48	82.39	93.63	85.77	92.51	84.34	99.14	96.06	97.74	92.55
AdaDetectGPT	61.64	56.87	71.97	69.69	71.98	64.50	48.52	17.02	66.41	59.75	55.84	45.47	53.26	55.23	78.28	71.22
DetectAnyLLM	78.73	71.49	91.58	83.38	96.96	91.63	89.45	82.92	96.25	89.64	81.21	73.26	95.51	88.37	89.82	81.08
RepreGuard	97.50	92.77	99.59	97.38	99.24	96.99	98.07	92.98	99.30	96.98	94.42	87.04	99.76	**99.10**	99.92	98.80
ProSSD	**99.72**	**98.00**	**99.97**	**99.30**	**99.89**	**99.50**	**99.95**	**98.89**	**99.97**	**99.65**	**99.90**	**98.60**	**99.80**	98.95	**99.99**	**99.55**

(Google DeepMind, 2025), Grok-4.1 (xAI, 2025). Following DetectRL, we evaluate three primary settings: model generalization, domain generalization, and OOD detection. For adversarial robustness experiments, we retain the original DetectRL construction to ensure direct comparability with prior work. We adopt AUROC and F1 Score as the core evaluation metrics. The construction protocols and descriptive statistics of the main benchmark are provided in Appendix D.1, while the construction details of supplementary evaluation sets are summarized in Appendix D.2. Extended MIRAGE analyses, including tests on machine-generated, human-polished text, are provided in Appendix E.6. Full algorithmic and experimental details are given in Appendix F.

Baselines. To establish a comprehensive benchmark, we compare ProSSD against a diverse set of state-of-the-art methods, including LRR (Su et al., 2023), NPR (Su et al., 2023), DetectGPT (Mitchell et al., 2023), Fast-DetectGPT (Bao et al., 2024), AdaDetectGPT (Zhou et al., 2025), Binoculars (Hans et al., 2024), RepreGuard (Chen et al., 2025b), RoBERTa-Base (Solaiman et al., 2019), ImBD (Chen et al., 2025a), and DetectAnyLLM (Fu et al., 2025). Specific implementation details and hyperparameter settings for all baselines are detailed in Appendix D.3.

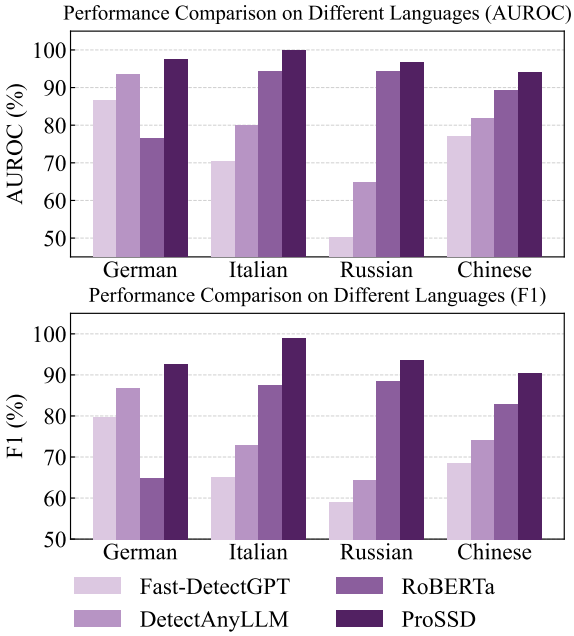


Figure 3: Performance comparison of ProSSD and baseline methods in multilingual settings on German, Italian, Russian, and Chinese datasets.

4.2 Main Comparative Results

Overall Detection Performance. Table 1 presents the main detection results under Multi-LLM and Multi-Domain settings. Overall, ProSSD demonstrates consistent superiority across all test scenarios, achieving SOTA performance. Notably, in detecting text generated by GPT-5.1 and Grok-4.1, ProSSD achieves F1 scores of 98.00% and 98.89%, respectively, representing improvements of 5.64% and 6.36% over the runner-up method, RepreGuard. In contrast, traditional supervised methods (e.g., RoBERTa) and statistical methods (e.g., Binoculars) exhibit significant performance volatility when facing frontier LLMs. In the multi-domain evaluation, apart from slightly lower performance on the XSum dataset compared to RepreGuard, ProSSD achieves the best performance across all other text styles (Academic, Creative, Reviews),

Table 2: Impact of training data sources on detection generalization. The table compares the performance when training on data generated by GPT-3.5 versus Llama-2-70b and testing on unseen target LLMs. Results are reported as AUROC (%) and F1 (%). The best and second-best scores are highlighted in **bold** and underlined.

Method	Train on GPT-3.5								Train on Llama-2-70b							
	GPT-5.1		Claude-S-4		Gemini-3-F		Grok-4.1		GPT-5.1		Claude-S-4		Gemini-3-F		Grok-4.1	
	AUR.	F1	AUR.	F1	AUR.	F1	AUR.	F1	AUR.	F1	AUR.	F1	AUR.	F1	AUR.	F1
LRR	68.02	58.08	79.62	71.13	63.30	59.10	62.18	49.50	68.02	58.08	79.62	71.13	63.30	59.10	62.18	49.50
Fast-DetectGPT	67.07	58.59	78.78	72.73	76.13	68.44	52.91	42.29	67.07	58.59	78.78	72.73	76.13	68.44	52.91	42.29
RoBERTa	77.33	73.34	90.71	83.57	86.77	79.34	78.64	71.99	76.48	70.86	86.02	79.65	85.21	77.28	78.92	73.64
ImBD	93.28	83.90	97.29	91.07	97.60	92.19	90.71	83.14	88.59	81.13	92.61	84.38	93.50	86.51	84.51	78.70
AdaDetectGPT	61.32	55.75	74.32	68.67	68.23	55.00	46.90	23.99	58.02	52.83	69.87	65.60	64.35	56.10	43.85	3.12
DetectAnyLLM	89.73	84.27	94.89	89.61	94.35	89.66	92.49	86.75	87.12	84.92	93.68	88.54	91.78	87.38	88.51	86.02
RepreGuard	78.97	73.68	94.49	87.79	89.44	82.38	81.67	74.29	72.83	62.87	88.72	81.90	81.48	74.26	74.85	69.21
ProSSD	**94.94**	**87.42**	**98.64**	**93.58**	**99.31**	**96.43**	**97.06**	**89.63**	**93.09**	**86.14**	**97.95**	**91.71**	**97.20**	**91.82**	**94.33**	**86.08**

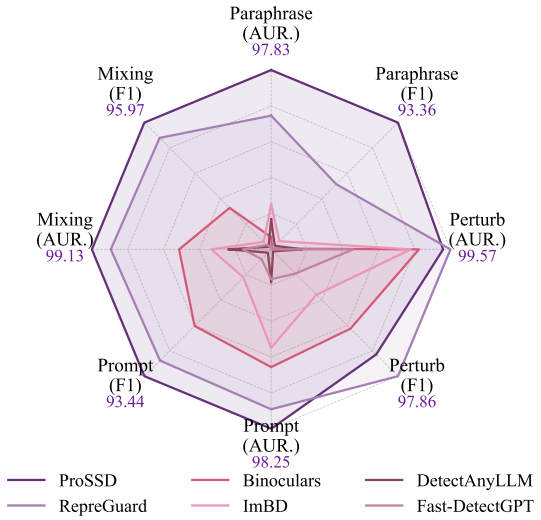


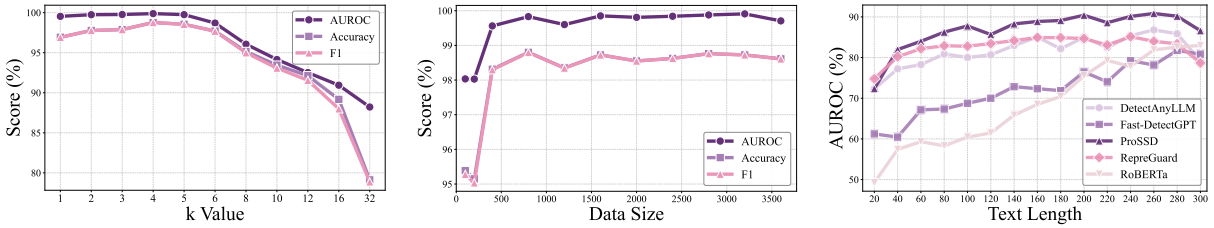
Figure 4: Performance comparison of ProSSD and baseline methods under Paraphrase, Perturbation, and Mixing attacks, with Direct Prompt as a reference setting.

demonstrating its robust generalizability across diverse semantic styles.

Out-of-Distribution Performance. Table 2 reports OOD performance, aiming to measure generalization ability when there are significant discrepancies between the generation sources of the training and test sets. The experiment involves using earlier models (e.g., GPT-3.5) for parameter estimation and testing on newer models (e.g., GPT-5.1). Results indicate that ProSSD significantly mitigates performance degradation caused by generative distribution shifts. Specifically, when trained only on GPT-3.5 (Table 2 Left), ProSSD maintains an AUROC above 94% on all four target test sets, achieving optimal results. When trained only on Llama-2-70b (Table 2 Right), its advantage remains significant when transferring to closed-source black-box models. For instance, when detecting Grok-4.1, ProSSD achieves an AUROC of 94.33%, significantly surpassing DetectAnyLLM (88.51%) and ImBD (84.51%) under the same setting. More detailed analyses of distribution shift, including supplementary OOD evaluation, adaptation to environmental changes, and highly creative or irregular texts, are provided in Appendix E.1.

Performance on Multilingual Datasets. To evaluate performance in non-English settings, we construct multilingual evaluation subsets covering German, Italian, Russian, and Chinese. Detailed dataset construction and settings are provided in Appendix D.2.1. As shown in Figure 3, ProSSD consistently outperforms all baseline methods across all four languages, with AUROC scores above 94% in every case and a peak of 99.83% on the Italian subset in particular. Relative to the strongest baseline in each language, the gain ranges from approximately 2.5 to 5.5 points, suggesting good cross-lingual transfer and consistently stable performance across languages.

Robustness Against Adversarial Attacks. Figure 4 shows radar charts of model robustness under three attack scenarios: paraphrase, perturbation, and data mixing. In terms of the performance envelope, ProSSD covers the largest area across all dimensions, indicating superior performance under attack modes without obvious shortcomings. Conversely, other methods like DetectAnyLLM and Binoculars exhibit significant "star-shaped" contraction, showing drastic performance drops particularly along the most challenging paraphrase axis. This demonstrates that our method maintains exceptional performance and robust reliability when



(a) Impact of Meta-Structure Order k on AUROC, Accuracy, and F1 Score.

(b) Impact of Data Size on AUROC, Accuracy and F1 Score.

(c) Impact of Text Length on AUROC Score.

Figure 5: Impact analysis of key hyperparameters and data variations on detection performance. (a) The AUROC score peaks at $k = 4$, indicating the optimal meta-structure order. (b) The model shows high data efficiency, reaching performance saturation with minimal training data size. (c) ProSSD remains robust across varying text lengths, consistently outperforming other detectors from short to long sequences.

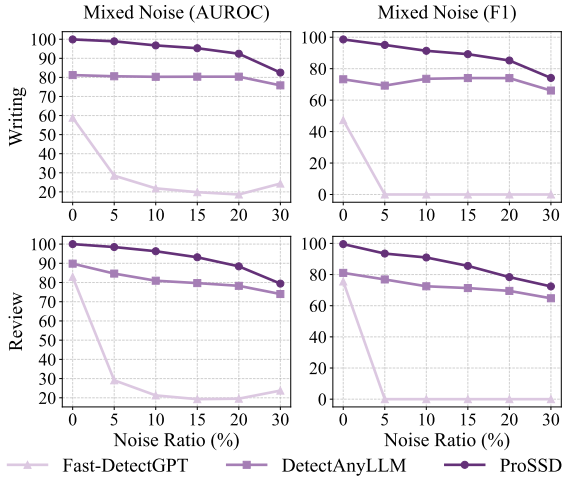


Figure 6: Robustness comparison of ProSSD and baseline methods under mixed-noise corruption, where typos, word swapping, and formatting degradation are jointly injected at different noise ratios on the Writing and Review datasets.

deployed in realistic adversarial environments. Detailed AUROC and F1 results and analyses are provided in Appendix E.2.1.

Robustness to Noisy and Informal Text. To simulate non-canonical inputs, we inject typos, word-order perturbations, and formatting degradation into the Writing Prompt and Yelp Review datasets; detailed settings are given in Appendix D.2.2. As shown in Figure 6, under mixed-noise conditions, although performance declines for all methods as the noise ratio increases, ProSSD consistently maintains the best detection results across all noise levels. Even under 30% mixed noise, its AUROC still reaches 82.52% and 79.43% on Writing Prompt and Yelp Review, respectively. In contrast, Fast-DetectGPT almost completely fails under severe noise, while DetectAnyLLM also shows substantial degradation. More detailed results and comparative analyses under different injected noise settings are provided in Appendix E.2.2.

4.3 Sensitivity and Efficiency Analysis

Impact of Meta-Structure Order k. Figure 5a illustrates the effect of the N-gram context window size k on detection performance. We observe that performance peaks at $k = 4$, which achieves an optimal balance between capturing local contextual dependencies and mitigating data sparsity inherent in higher dimensions. Notably, as k increases further, ProSSD maintains stable AUROC and F1 scores, demonstrating the robustness of the framework to hyperparameter variations.

Impact of Data Size. Figure 5b evaluates the impact of the reference set size on model efficacy. The performance curve shows a steep upward trend in the early stage ($N \leq 400$). It then rapidly enters a convergence plateau, where marginal gains from increasing data volume diminish. Unlike deep learning baselines that typically require massive corpora to fit decision boundaries, ProSSD demonstrates extremely high data efficiency. This makes it highly suitable for few-shot application scenarios.

Impact of Text Length. Text length determines the information density available for statistical inference. Figure 5c compares the response patterns of different detectors across varying text lengths. Supervised models, represented by RoBERTa, exhibit a clear linear dependency within the 20-300 token range. This indicates they require longer contexts to accumulate sufficient semantic features. In sharp contrast, ProSSD demonstrates a superior information utilization rate, even in the extremely short text range (20-80 tokens), ProSSD achieves a rapid performance climb and quickly converges to a high-performance zone.

Table 3: Efficiency comparison of per-sample inference time and GPU memory usage. The AUROC scores are provided for reference. Best results are highlighted in **bold** and second-best are underlined.

Method	Time (s)↓	GPU Mem (GB)↓	AUR.↑
DetectGPT	11.83	10.87	56.77
Binoculars	0.19	26.24	92.52
Fast-DetectGPT	0.37	33.40	78.78
AdaDetectGPT	3.06	37.98	71.97
DetectAnyLLM	**0.08**	5.32	91.58
RepreGuard	0.36	30.84	99.59
ProSSD	**0.08**	**1.35**	**99.97**

Computational Efficiency. Computational cost is critical for practical deployment. ProSSD has achieved an inference speed of 0.08s, comparable to lightweight reward models. Notably, it requires only 1.35 GB of VRAM, a 74.62% reduction compared to the runner-up, demonstrating its capability to maintain high-precision detection with minimal resource overhead.

Additional diagnostic analyses, including N-gram overlap and sensitivity to decoding strategies, are reported in Appendix E.3.

4.4 Ablation and Interpretability Analysis

Ablation Study. Table 4 verifies the necessity of ProSSD's components. First, replacing SSP with unsupervised PCA or random projection causes significant AUC drops (e.g., ∼4.5% on XSum), confirming the role of label-aware feature extraction in isolating discriminative signals from noise. Second, removing Wasserstein Distance weights or degenerating to one-sided density estimation consistently weakens performance, establishing the need for adaptive weighting and dual-distribution contrast. Finally, the framework's robustness across embedding models suggests it captures intrinsic distributional laws independent of encoder biases. Detailed ablation configurations and results are provided in Appendix E.4.

Interpretability Analysis. Our in-depth analysis reveals that discriminative cues exhibit strong domain dependency: formal corpora rely heavily on logical connectives reflecting deep syntactic coherence, whereas subjective texts hinge on pronouns and adjectives. ProSSD establishes a significantly wider safety margin in the numerical space, compressing the distributional overlap between human and machine text to a negligible level. For comprehensive analysis and visualizations, please refer to the Appendix E.5.

Table 4: Ablation studies (AUC) on XSum and Review. See Table 11 for full results. The statistical significance of the performance drop compared to ProSSD is measured by a t-test: $^{*}\ p < 0.05$, $^{\dagger}\ p < 0.01$, $^{\ddagger}\ p < 0.001$.

Method	XSum	Review
ProSSD (RoBERTa + SSP)	**99.21**±0.19	**99.67**±0.04
Robustness of Semantic Representations		
Qwen-2.5-0.6B-Embed	95.28±0.27‡	99.50±0.06†
Random Projection	93.45±1.04‡	91.49±2.58†
Impact of Subspace Projection		
w/o SSP (PCA)	94.75±0.39‡	97.54±0.08‡
w/o SSP (Rand)	93.39±0.73‡	96.46±0.51‡
w/o SSP (No-Proj)	93.97±0.35‡	96.74±0.37‡
Ablation on Detection Strategies		
w/o Wasserstein (Unif.)	96.48±0.94†	97.57±0.47‡
w/o Contrastive (Human)	93.86±0.34‡	88.70±0.25‡

5 Conclusion

This paper presents ProSSD, a detection framework without task-specific fine-tuning, built on supervised subspace learning and joint semantic-structural distribution modeling. Leveraging the Wasserstein Distance, ProSSD effectively mitigates high-dimensional noise and captures the statistical characteristics of MGT. Experiments show that ProSSD achieves SOTA performance across various benchmarks while significantly reducing computational costs. Unlike previous methods, our approach formulates detection as an interpretable statistical test, offering a lightweight and transparent solution for LLM governance.

Limitations

Despite the strong performance of ProSSD, several limitations remain. First, the theoretical understanding of its internal mechanism is still limited. Our observations of semantic-structural statistical deviations are largely empirical, and a clear theoretical connection to Transformer internals, such as attention patterns or decoding strategies, is still lacking. Second, reliable detection remains challenging in difficult scenarios, especially for heavily human-polished or human-machine hybrid text. Third, the detection paradigm still requires further extension. The current framework focuses on binary classification, whereas the field is increasingly moving toward fine-grained source attribution (Park et al., 2025). ProSSD has not yet been extended to explicitly attribute text to specific generative sources.

References

Alim Al Ayub Ahmed, Ayman Aljabouh, Praveen Kumar Donepudi, and Myung Suh Choi. 2021. Detecting fake news using machine learning : A systematic literature review. *CoRR*, abs/2102.04458.

Theodore Wilbur Anderson, Theodore Wilbur Anderson, Theodore Wilbur Anderson, Theodore Wilbur Anderson, and Etats-Unis Mathématicien. 1958. *An introduction to multivariate statistical analysis*, volume 2. Wiley New York.

Rohan Anil, Andrew M. Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, Eric Chu, Jonathan H. Clark, Laurent El Shafey, Yanping Huang, Kathy Meier-Hellstern, Gaurav Mishra, Erica Moreira, Mark Omernick, Kevin Robinson, and 34 others. 2023. Palm 2 technical report. *CoRR*, abs/2305.10403.

Anthropic. 2025. Claude 4. Anthropic Blog.

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

Yinpeng Cai, Lexin Li, and Linjun Zhang. 2025. A statistical hypothesis testing framework for data misappropriation detection in large language models. *CoRR*, abs/2501.02441.

Jiaqi Chen, Xiaoye Zhu, Tianyang Liu, Ying Chen, Xinhui Chen, Yiwen Yuan, Chak Tou Leong, Zuchao Li, Tang Long, Lei Zhang, Chenyu Yan, Guanghao Mei, Jie Zhang, and Lefei Zhang. 2025a. Imitate before detect: Aligning machine stylistic preference for machine-revised text detection. In *AAAI-25, Sponsored by the Association for the Advancement of Artificial Intelligence, February 25 - March 4, 2025, Philadelphia, PA, USA*, pages 23559–23567. AAAI Press.

Xin Chen, Junchao Wu, Shu Yang, Runzhe Zhan, Zeyu Wu, Ziyang Luo, Di Wang, Min Yang, Lidia S. Chao, and Derek F. Wong. 2025b. Repreguard: Detecting llm-generated text by revealing hidden representation patterns. *CoRR*, abs/2508.13152.

Miranda Christ, Sam Gunn, and Or Zamir. 2024. Undetectable watermarks for language models. In *The Thirty Seventh Annual Conference on Learning Theory, June 30 - July 3, 2023, Edmonton, Canada*, volume 247 of *Proceedings of Machine Learning Research*, pages 1125–1139. PMLR.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *CoRR*, abs/1911.02116.

Harald Cramér and Herman Wold. 1936. Some theorems on distribution functions. *Journal of the London Mathematical Society*, 1(4):290–294.

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

Dorottya Demszky, Diyi Yang, David S Yeager, Christopher J Bryan, Margarett Clapper, Susannah Chandhok, Johannes C Eichstaedt, Cameron Hecht, Jeremy Jamieson, Meghann Johnson, and 1 others. 2023. Using large language models in psychology. *Nature Reviews Psychology*, 2(11):688–701.

Anil R Doshi and Oliver P Hauser. 2024. Generative ai enhances individual creativity but reduces the collective diversity of novel content. *Science advances*, 10(28):eadn5290.

Matthew Durward and Christopher Thomson. 2024. Evaluating vocabulary usage in llms. In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications, BEA 2024, Mexico City, Mexico, June 20, 2024*, pages 266–282. Association for Computational Linguistics.

Tiziano Fagni, Fabrizio Falchi, Margherita Gambini, Antonio Martella, and Maurizio Tesconi. 2020. Tweepfake: about detecting deepfake tweets. *CoRR*, abs/2008.00036.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898, Melbourne, Australia. Association for Computational Linguistics.

William Feller. 1991. *An introduction to probability theory and its applications, Volume 2*, volume 2. John Wiley & Sons.

Leon Fröhling and Arkaitz Zubiaga. 2021. Feature-based detection of automated language models: tackling gpt-2, GPT-3 and grover. *PeerJ Comput. Sci.*, 7:e443.

Jiachen Fu, Chun-Le Guo, and Chongyi Li. 2025. Detectanyllm: Towards generalizable and robust detection of machine-generated text across domains and models. *CoRR*, abs/2509.14268.

Zheng Ge, Songtao Liu, Zeming Li, Osamu Yoshie, and Jian Sun. 2021. OTA: optimal transport assignment for object detection. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 303–312. Computer Vision Foundation / IEEE.

Sebastian Gehrmann, Hendrik Strobelt, and Alexander M. Rush. 2019. GLTR: statistical detection and visualization of generated text. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July*

28 - August 2, 2019, Volume 3: System Demonstrations, pages 111–116. Association for Computational Linguistics.

Noah Golowich and Ankur Moitra. 2024. Edit distance robust watermarks via indexing pseudorandom codes. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Google DeepMind. 2025. Gemini 3 flash model card. Technical Report.

Abhimanyu Hans, Avi Schwarzschild, Valeriia Cherepanova, Hamid Kazemi, Aniruddha Saha, Micah Goldblum, Jonas Geiping, and Tom Goldstein. 2024. Spotting llms with binoculars: Zero-shot detection of machine-generated text. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

Beizhe Hu, Qiang Sheng, Juan Cao, Yang Li, and Danding Wang. 2025. Llm-generated fake news induces truth decay in news ecosystem: A case study on neural news recommendation. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2025, Padua, Italy, July 13-18, 2025*, pages 435–445. ACM.

Xiaomeng Hu, Pin-Yu Chen, and Tsung-Yi Ho. 2023. RADAR: robust ai-text detection via adversarial learning. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Daphne Ippolito, Daniel Duckworth, Chris Callison-Burch, and Douglas Eck. 2020. Automatic detection of generated text is easiest when humans are fooled. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 1808–1822. Association for Computational Linguistics.

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.

Hoang Anh Just, Feiyang Kang, Tianhao Wang, Yi Zeng, Myeongseob Ko, Ming Jin, and Ruoxi Jia. 2023. LAVA: data valuation without pre-specified learning algorithms. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. 2023. A watermark for large language models. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 17061–17084. PMLR.

Ryuto Koike, Masahiro Kaneko, Ayana Niwa, Preslav Nakov, and Naoaki Okazaki. 2025. Exagpt: Example-based machine-generated text detection for human interpretability. *CoRR*, abs/2502.11336.

Ryuto Koike, Masahiro Kaneko, and Naoaki Okazaki. 2024. OUTFOX: llm-generated essay detection through in-context learning with adversarially generated examples. In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, pages 21258–21266. AAAI Press.

Kalpesh Krishna, Yixiao Song, Marzena Karpinska, John Wieting, and Mohit Iyyer. 2023. Paraphrasing evades detectors of ai-generated text, but retrieval is an effective defense. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Richard J Larsen and Morris L Marx. 2005. *An introduction to mathematical statistics*, volume 106. Prentice Hall Hoboken, NJ.

Olivier Ledoit and Michael Wolf. 2004. A well-conditioned estimator for large-dimensional covariance matrices. *Journal of multivariate analysis*, 88(2):365–411.

Donghyeok Lee, Christina Todorova, and Alireza Dehghani. 2024. Ethical risks and future direction in building trust for large language models application under the EU AI act. In *Proceedings of the 2024 Conference on Human Centred Artificial Intelligence - Education and Practice, HCAIep 2024, Naples, Italy, December 2-3, 2024*, pages 41–46. ACM.

Xiaoze Liu, Ting Sun, Tianyang Xu, Feijie Wu, Cunxiang Wang, Xiaoqian Wang, and Jing Gao. 2024. SHIELD: Evaluation and defense strategies for copyright compliance in LLM text generation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1640–1670, Miami, Florida, USA. Association for Computational Linguistics.

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

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D. Manning, and Chelsea Finn. 2023. Detectgpt: Zero-shot machine-generated text detection using probability curvature. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202

of *Proceedings of Machine Learning Research*, pages 24950–24962. PMLR.

Ryo Nagata, Hiroya Takamura, Naoki Otani, and Yoshifumi Kawasaki. 2023. Variance matters: Detecting semantic differences without corpus/word alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 15609–15622. Association for Computational Linguistics.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1797–1807, Brussels, Belgium. Association for Computational Linguistics.

OpenAI. 2023. Introducing chatgpt. OpenAI Blog.

OpenAI. 2025. Gpt-5.1. OpenAI Blog.

Hyeonchu Park, Byungjun Kim, and Bugeun Kim. 2025. DART: an AIGT detector using AMR of rephrased text. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2025 - Volume 2: Short Papers, Albuquerque, New Mexico, April 29 - May 4, 2025*, pages 710–721. Association for Computational Linguistics.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Vinu Sankar Sadasivan, Aounon Kumar, Sriram Balasubramanian, Wenxiao Wang, and Soheil Feizi. 2025. Can ai-generated text be reliably detected? stress testing AI text detectors under various attacks. *Trans. Mach. Learn. Res.*, 2025.

Areg Mikael Sarvazyan, José Ángel González, Paolo Rosso, and Marc Franco-Salvador. 2023. Supervised machine-generated text detectors: Family and scale matters. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction - 14th International Conference of the CLEF Association, CLEF 2023, Thessaloniki, Greece, September 18-21, 2023, Proceedings*, volume 14163 of *Lecture Notes in Computer Science*, pages 121–132. Springer.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347.

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

Jinyan Su, Terry Yue Zhuo, Di Wang, and Preslav Nakov. 2023. Detectllm: Leveraging log rank information for zero-shot detection of machine-generated text. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 12395–12412. Association for Computational Linguistics.

Chenxia Tang, Jianchun Liu, Hongli Xu, and Liusheng Huang. 2025. Top-$n\sigma$: Eliminating noise in logit space for robust token sampling of LLM. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10758–10774, Vienna, Austria. Association for Computational Linguistics.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

Eduard Tulchinskii, Kristian Kuznetsov, Laida Kushnareva, Daniil Cherniavskii, Sergey I. Nikolenko, Evgeny Burnaev, Serguei Barannikov, and Irina Piontkovskaya. 2023. Intrinsic dimension estimation for robust detection of ai-generated texts. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Cédric Villani and 1 others. 2008. *Optimal transport: old and new*, volume 338. Springer.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Osama Mohammed Afzal, Tarek Mahmoud, Giovanni Puccetti, Thomas Arnold, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024. M4gt-bench: Evaluation benchmark for black-box machine-generated text detection. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 3964–3992. Association for Computational Linguistics.

Herman Wold. 1966. Estimation of principal components and related models by iterative least squares. *Multivariate analysis*, pages 391–420.

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

Junchao Wu, Runzhe Zhan, Derek F. Wong, Shu Yang, Xuebo Liu, Lidia S. Chao, and Min Zhang. 2025b. Who wrote this? the key to zero-shot llm-generated text detection is gecscore. In *Proceedings of the 31st International Conference on Computational Linguistics, COLING 2025, Abu Dhabi, UAE, January 19-24, 2025*, pages 10275–10292. Association for Computational Linguistics.

Junchao Wu, Runzhe Zhan, Derek F. Wong, Shu Yang, Xinyi Yang, Yulin Yuan, and Lidia S. Chao. 2024.

Detectrl: Benchmarking llm-generated text detection in real-world scenarios. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

xAI. 2025. Grok-4.1 model card. Technical Report.

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

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. Defending against neural fake news. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 9051–9062.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 649–657.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. 2025. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*.

Hongyi Zhou, Jin Zhu, Pingfan Su, Kai Ye, Ying Yang, Shakeel A O. B. Gavioli-Akilagun, and Chengchun Shi. 2025. Adadetectgpt: Adaptive detection of llm-generated text with statistical guarantees. *CoRR*, abs/2510.01268.

Appendices

A Additional Related Works

Beyond detection methods relying solely on intrinsic features, existing research has extensively explored paradigms that incorporate external auxiliary signals.

Watermarking employs an active defense strategy, aiming to implicitly embed verifiable statistical signals into text via specific sampling algorithms during generation (Kirchenbauer et al., 2023). However, this technique faces a fundamental trade-off. Balancing the preservation of semantic integrity with the maximization of robustness and statistical detectability has become a central focus of recent research (Christ et al., 2024; Golowich and Moitra, 2024; Cai et al., 2025).

In contrast, retrieval-based methods incorporate external knowledge sources. They utilize sparse or dense indexing mechanisms to align and compare the input text against large-scale reference corpora. By quantifying distributional discrepancies in linguistic patterns between the input and reference samples (human or machine), these methods significantly enhance generalization in cross-domain scenarios (Krishna et al., 2023; Sadasivan et al., 2025). Furthermore, they offer new perspectives on understanding adversarial in-context learning and improving detection interpretability (Koike et al., 2024, 2025).

Notably, although ProSSD operates without task-specific fine-tuning, it draws inspiration from both aforementioned approaches.

B Geometric Analysis of HWT and MGT Distributions

This section aims to provide an in-depth theoretical elucidation of the intrinsic differences between HWT and MGT revealed in Figure 1 and Figure 7, from the perspectives of geometric topology and statistical properties.

B.1 Systematic Translation in Semantic Space

Figure 1 visualizes the distribution patterns of HWT and MGT within the low-dimensional discriminative subspace. We project high-dimensional embeddings using the supervised subspace learning described in Section 3.2 and calculate the class centroids $\boldsymbol{\mu}_H^\pi$ and $\boldsymbol{\mu}_M^\pi$ for each syntactic structure π. For intuitive visualization on a 2D plane, we apply principal component analysis (PCA) to these

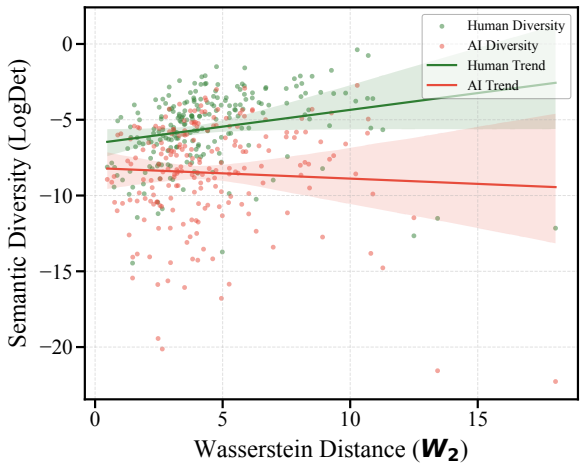


Figure 7: Semantic diversity under different structures. The increasing distance between AI and human centers indicates the rarity of the semantics.

sets of means, spanning the maximum variance plane defined by the first (PC1) and second (PC2) principal components.

The most significant geometric feature in the plot is the phenomenon of **Systematic Translation**. Although distinct syntactic structures (e.g., conjunction-noun combinations, verb phrases) are scattered across different quadrants of the feature space, the connection lines between each pair of corresponding structural centroids $(\boldsymbol{\mu}_H^\pi, \boldsymbol{\mu}_M^\pi)$ exhibit a high degree of directional consistency. This indicates that the stylistic differences of MGT relative to human text are not local, disordered random perturbations, but a systematic bias pervasive throughout the feature space. We formally model this relationship as:

$$\boldsymbol{\mu}_M^\pi = \boldsymbol{\mu}_H^\pi + \boldsymbol{\delta}_{global} + \boldsymbol{\epsilon}_\pi, \tag{12}$$

where $\boldsymbol{\delta}_{global}$ is the global offset vector independent of specific syntactic structures, and $\boldsymbol{\epsilon}_\pi$ is a minor residual term for a specific structure (where $\|\boldsymbol{\epsilon}_\pi\| \ll \|\boldsymbol{\delta}_{global}\|$). Notably, since the entire projection transformation is linear, a linear transformation cannot reconstruct anisotropic random noise in high-dimensional space into approximately parallel structured vectors in low-dimensional space. Therefore, this parallelism confirms that the systematic bias $\boldsymbol{\delta}_{global}$ is an intrinsic essential feature of the generative distribution of LLMs.

B.2 Relationship Between Discriminative Distance and Distributional Diversity

Figure 7 further reveals significant differences in second-order statistics between HWT and MGT. While not detailed in the main text, this is crucial for understanding model behavior. The x-axis represents the Wasserstein distance weight w_π, which measures the deviation of MGT from HWT under that syntactic structure. And the y-axis represents the log-determinant of the covariance matrix $\ln|\boldsymbol{\Sigma}|$, which measures the volume or diversity of the semantic distribution.

The observations reveal two diametrically opposed generation patterns:

- **Human-Written Text:** Exhibits a positive correlation trend. As the syntactic structure deviates from convention (i.e., w_π increases), the semantic richness of human text increases rather than decreases. This suggests that humans tend to mobilize a more diverse vocabulary to express precise meanings when navigating complex or rare syntax.

- **Machine-Generated Text:** Exhibits a significant negative correlation trend. When the model faces high-difficulty syntactic structures that deviate from its training priors, the volume of its generated semantic distribution shrinks significantly.

We define this phenomenon in MGT as **Conditional Semantic Collapse**: under the strong constraints of specific syntactic structures, LLMs tend to adopt conservative decoding strategies, converging to high-frequency, generic word combinations. This results in a significant reduction in local variance ($\ln|\boldsymbol{\Sigma}_M| \ll \ln|\boldsymbol{\Sigma}_H|$). This collapse reflects the model's tendency towards uncertainty avoidance when processing complex syntax.

B.3 Proof of Discriminative Effectiveness via Mahalanobis Distance

Combining the findings of systematic bias $\boldsymbol{\delta}_{global}$ and semantic collapse, we hereby prove the theoretical validity of the detection statistic (difference in modified Mahalanobis distance) proposed in Section 3.3.

Proposition 1 (Lower Bound of Expected Detection Statistic)**.** *Assume that in the feature space, the local features* $\mathbf{x}$ *of MGT follow the distribution* $\mathcal{N}(\boldsymbol{\mu}_M, \boldsymbol{\Sigma})$*, and HWT follows* $\mathcal{N}(\boldsymbol{\mu}_H, \boldsymbol{\Sigma})$*, with a non-zero offset* $\boldsymbol{\delta} = \boldsymbol{\mu}_M - \boldsymbol{\mu}_H$*. Define the single-step detection statistic* s_t *as the difference between the distance to the human center and the distance*

to the machine center:

$$s_t = \frac{1}{2}\Big[(\mathbf{x}-\boldsymbol{\mu}_H)^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}_H) - (\mathbf{x}-\boldsymbol{\mu}_M)^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}_M)\Big]. \tag{13}$$

Then, for machine-generated samples, the expectation of the score is strictly greater than zero.

Proof: Reparameterize the machine-generated sample as $\mathbf{x} = \boldsymbol{\mu}_M + \mathbf{z}$, where $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$ is zero-mean noise. Substituting this into the formula for s_t and expanding:

First term, distance relative to human center:

$$(\boldsymbol{\mu}_M + \mathbf{z} - \boldsymbol{\mu}_H)^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_M + \mathbf{z} - \boldsymbol{\mu}_H) = (\boldsymbol{\delta} + \mathbf{z})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{\delta} + \mathbf{z}). \tag{14}$$

Second term, distance relative to machine center:

$$(\boldsymbol{\mu}_M + \mathbf{z} - \boldsymbol{\mu}_M)^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_M + \mathbf{z} - \boldsymbol{\mu}_M) = \mathbf{z}^T\boldsymbol{\Sigma}^{-1}\mathbf{z}. \tag{15}$$

Subtracting the two terms yields the simplified expression for $2s_t$:

$$2s_t = \boldsymbol{\delta}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\delta} + 2\boldsymbol{\delta}^T\boldsymbol{\Sigma}^{-1}\mathbf{z}. \tag{16}$$

Taking the mathematical expectation of the above equation, since $\mathbf{z}$ is zero-mean noise, the expectation of the linear cross-term is $\mathbb{E}[\mathbf{z}] = \mathbf{0}$. Therefore:

$$\mathbb{E}[s_t] = \frac{1}{2}\boldsymbol{\delta}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\delta} = \frac{1}{2}\|\boldsymbol{\delta}\|^2_{\boldsymbol{\Sigma}^{-1}}. \tag{17}$$

Since the covariance matrix $\boldsymbol{\Sigma}$ is positive definite, its inverse matrix $\boldsymbol{\Sigma}^{-1}$ is also positive definite. According to the conclusion in Section 1, the systematic offset $\|\boldsymbol{\delta}\| > 0$; thus, $\mathbb{E}[s_t]$ is strictly positive. ■

This proposition not only proves the validity of the Mahalanobis distance but also reveals the contribution of the semantic collapse described in Section B.2 to detection performance. The expectation of the statistic s_t is proportional to $\boldsymbol{\delta}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\delta}$. The observed shrinkage in the distribution volume of MGT in Figure 7 implies that the eigenvalues of its covariance matrix become smaller, which in turn causes the eigenvalues of its inverse matrix $\boldsymbol{\Sigma}^{-1}$ to increase. This expansion of the precision matrix essentially acts as a magnifying glass, significantly amplifying the weight of the systematic bias $\boldsymbol{\delta}$ in the statistic, thereby further enhancing the discriminative signal-to-noise ratio of the model against MGT.

C Detailed Derivations and Proofs

C.1 Derivation of Supervised Subspace Projection

This section provides mathematical proofs for the recursive optimization problem defined in Section 3.2 of the main text. We derive the closed-form solution for the optimal projection direction, prove the orthogonality of the feature deflation step, and finally elucidate the necessity of introducing multi-dimensional features for distribution estimation.

Proposition 2. *Given the centered residual embedding matrix $\mathbf{E}_{j-1} \in \mathbb{R}^{N\times D}$ and the centered label vector $\mathbf{y} \in \mathbb{R}^N$, the optimal solution $\mathbf{p}_j^*$ to the optimization problem*

$$\mathbf{p}_j^* = \underset{\mathbf{p}\in\mathbb{R}^D}{\operatorname{argmax}} \left(\operatorname{Cov}(\mathbf{E}_{j-1}\mathbf{p}, \mathbf{y})\right)^2 \quad \text{s.t.} \quad \|\mathbf{p}\|_2 = 1, \tag{18}$$

is the normalized cross-covariance vector between the current residual features and the labels.

Proof: Since $\mathbf{E}_{j-1}$ and $\mathbf{y}$ are centered vectors, their sample covariance is proportional to their inner product. The objective function can be rewritten as:

$$J(\mathbf{p}) = (\mathbf{y}^T\mathbf{E}_{j-1}\mathbf{p})^2 = \mathbf{p}^T(\mathbf{E}_{j-1}^T\mathbf{y}\mathbf{y}^T\mathbf{E}_{j-1})\mathbf{p}. \tag{19}$$

We introduce the Lagrange multiplier λ to construct the Lagrangian function:

$$\mathcal{L}(\mathbf{p}, \lambda) = \mathbf{p}^T\mathbf{M}_{j-1}\mathbf{p} - \lambda(\mathbf{p}^T\mathbf{p} - 1), \tag{20}$$

where $\mathbf{M}_{j-1} = (\mathbf{E}_{j-1}^T\mathbf{y})(\mathbf{y}^T\mathbf{E}_{j-1})$ is a rank-1 positive semi-definite matrix. Taking the partial derivative with respect to $\mathbf{p}$ and setting it to zero yields the eigenvalue equation:

$$\frac{\partial\mathcal{L}}{\partial\mathbf{p}} = 2\mathbf{M}_{j-1}\mathbf{p} - 2\lambda\mathbf{p} = 0 \implies \mathbf{M}_{j-1}\mathbf{p} = \lambda\mathbf{p}. \tag{21}$$

This indicates that $\mathbf{p}_j^*$ must be the eigenvector corresponding to the maximum eigenvalue of matrix $\mathbf{M}_{j-1}$. Let $\mathbf{u} = \mathbf{E}_{j-1}^T\mathbf{y}$, then $\mathbf{M}_{j-1} = \mathbf{u}\mathbf{u}^T$. Applying matrix $\mathbf{M}_{j-1}$ to vector $\mathbf{u}$:

$$\mathbf{M}_{j-1}\mathbf{u} = (\mathbf{u}\mathbf{u}^T)\mathbf{u} = \mathbf{u}(\mathbf{u}^T\mathbf{u}) = \|\mathbf{u}\|^2\mathbf{u}. \tag{22}$$

It follows that $\mathbf{u} = \mathbf{E}_{j-1}^T\mathbf{y}$ is the principal eigenvector corresponding to the maximum eigenvalue $\lambda_{max} = \|\mathbf{u}\|^2$. Combining this with the unit norm

constraint $\|\mathbf{p}\|_2 = 1$, we obtain the closed-form solution:

$$\mathbf{p}_j^* = \frac{\mathbf{E}_{j-1}^T \mathbf{y}}{\|\mathbf{E}_{j-1}^T \mathbf{y}\|_2}. \tag{23}$$

This vector indicates that the optimal projection direction is collinear with the current residual-label cross-covariance direction. ■

Proposition 3. *The feature deflation step ensures that the deflated residual matrix $\mathbf{E}_j$ is orthogonal to the current projection direction $\mathbf{p}_j^*$.*

Proof: By definition, $\mathbf{E}_j = \mathbf{E}_{j-1} - \mathbf{s}_j(\mathbf{p}_j^*)^T$, where $\mathbf{s}_j = \mathbf{E}_{j-1}\mathbf{p}_j^*$. Examining the projection of the residual matrix $\mathbf{E}_j$ onto the current direction $\mathbf{p}_j^*$:

$$\begin{aligned} \mathbf{E}_j \mathbf{p}_j^* &= (\mathbf{E}_{j-1} - \mathbf{s}_j(\mathbf{p}_j^*)^T)\mathbf{p}_j^* \\ &= \mathbf{E}_{j-1}\mathbf{p}_j^* - \mathbf{s}_j(\mathbf{p}_j^*)^T \mathbf{p}_j^*. \end{aligned} \tag{24}$$

Substituting $\mathbf{s}_j$ and utilizing the unit vector property $(\mathbf{p}_j^*)^T \mathbf{p}_j^* = 1$, we obtain:

$$\mathbf{E}_j \mathbf{p}_j^* = \mathbf{s}_j - \mathbf{s}_j(1) = \mathbf{0}. \tag{25}$$

That is, the column space of the residual matrix $\mathbf{E}_j$ is orthogonal to the current projection direction $\mathbf{p}_j^*$. ■

Analysis on Multi-dimensional Subspace Construction. The above propositions establish the mathematical foundation for the iterative solution. It is necessary to clarify why constructing a subspace with $k > 1$ remains essential in a binary classification task, where y is a 1D scalar.

According to Proposition 2, the first basis vector $\mathbf{p}_1^*$ effectively captures all linear mean differences between HWT and MGT. However, the probabilistic model proposed in Section 3.3 relies not only on the mean $\boldsymbol{\mu}$ but also heavily on the estimation of the covariance matrix $\boldsymbol{\Sigma}$ to capture precise distributional characteristics. If only $k = 1$ is selected, the feature space degenerates into a line, and the covariance matrix degenerates into a scalar variance, failing to describe the spatial distribution shape of the data. Through the aforementioned iterative process, subsequent projection vectors $\mathbf{p}_j^*(j > 1)$ continue to extract major structural information of the data in the residual space orthogonal to $\mathbf{p}_1^*$. This set of orthogonal bases $\{\mathbf{p}_1, \ldots, \mathbf{p}_k\}$ jointly constitutes a low-dimensional complete feature subspace. This allows us to retain the original discriminative information while accurately estimating the covariance structure of the Gaussian distribution. Mathematically, this aligns with the NIPALS algorithm of partial least squares (PLS) (Wold, 1966), which constructs stable low-dimensional representations through iterative deflation.

C.2 Theoretical Analysis of Distribution Modeling

C.2.1 Proof of Asymptotic Normality

This section establishes the distributional properties of the meta-semantic vector $\mathbf{x}_t$, defined in Section 3.3 of the main text, as the original embedding dimension $D \to \infty$.

Definition C.1 (Construction of Meta-Semantic Vectors) Let $\mathbf{e}_t, \mathbf{e}_{t+1} \in \mathbb{R}^D$ be the original embedding vectors at time steps t and $t+1$. Let $\mathbf{P} \in \mathbb{R}^{D \times k}$ be the column-orthogonal projection matrix obtained in Section 3.2. We define the meta-semantic vector $\mathbf{x}_t \in \mathbb{R}^{2k}$ as the concatenation of projected features from adjacent time steps:

$$\mathbf{x}_t = \begin{bmatrix} \mathbf{P}^T \mathbf{e}_t \\ \mathbf{P}^T \mathbf{e}_{t+1} \end{bmatrix}. \tag{26}$$

Assumption 1 (High-dimensional Noise Decomposition)**.** *Following common settings in high-dimensional statistics, we model the original embedding $\mathbf{e}_t$ as the sum of a deterministic mean signal and random noise:*

$$\mathbf{e}_t = \boldsymbol{\mu}_t + \boldsymbol{\epsilon}_t, \tag{27}$$

where $\boldsymbol{\mu}_t = \mathbb{E}[\mathbf{e}_t]$ is the intrinsic semantic mean within this context, and $\boldsymbol{\epsilon}_t$ is a zero-mean random noise vector.

We construct a joint noise vector $\boldsymbol{\xi} \in \mathbb{R}^{2D}$ by concatenating the components of $\boldsymbol{\epsilon}_t$ and $\boldsymbol{\epsilon}_{t+1}$:

$$\boldsymbol{\xi} = [\epsilon_{t,1}, \ldots, \epsilon_{t,D}, \epsilon_{t+1,1}, \ldots, \epsilon_{t+1,D}]^T. \tag{28}$$

We assume the components $\{\xi_j\}_{j=1}^{2D}$ of $\boldsymbol{\xi}$ satisfy the following conditions:

1. ***Zero Mean:*** $\mathbb{E}[\xi_j] = 0$.
2. ***Finite Variance:*** $Var(\xi_j) = \sigma_j^2 < \infty$.
3. ***Boundedness:*** *There exists a constant $M < \infty$ such that $|\xi_j| \leq M$ holds almost everywhere (this assumption is guaranteed by the normalization mechanism of LayerNorm layers).*

Assumption 2 (Feller Condition)**.** *For any unit projection direction, we assume that the projection*

coefficients do not overly concentrate on any single original dimension. Formally, for linear combination coefficients γ_j, the Feller delocalization condition is satisfied as $D \to \infty$:

$$\lim_{D\to\infty} \frac{\max_{1\le j\le 2D} |\gamma_j|}{\sqrt{\sum_{j=1}^{2D} \gamma_j^2 \sigma_j^2}} = 0. \tag{29}$$

Proposition 4 (Asymptotic Normality of Meta-Semantic Distribution)**.** *Based on Assumptions 1 and 2, as the original dimension $D \to \infty$, the meta-semantic vector $\mathbf{x}_t$ converges in distribution to a multivariate normal distribution:*

$$\mathbf{x}_t \xrightarrow{d} \mathcal{N}(\boldsymbol{\mu}_{\mathbf{x}}, \boldsymbol{\Sigma}_{\mathbf{x}}), \tag{30}$$

where $\boldsymbol{\mu}_{\mathbf{x}} = [\boldsymbol{\mu}_t^T \mathbf{P}, \boldsymbol{\mu}_{t+1}^T \mathbf{P}]^T$.

Proof: According to the Cramér-Wold theorem (Cramér and Wold, 1936), a necessary and sufficient condition for the random vector $\mathbf{x}_t$ to follow a multivariate normal distribution is that for any non-zero constant vector $\boldsymbol{\lambda} \in \mathbb{R}^{2k}$, the scalar random variable $Z_D = \boldsymbol{\lambda}^T \mathbf{x}_t$ follows a univariate normal distribution.

Partition $\boldsymbol{\lambda}$ into $\boldsymbol{\lambda} = [\boldsymbol{\alpha}^T, \boldsymbol{\beta}^T]^T$, where $\boldsymbol{\alpha}, \boldsymbol{\beta} \in \mathbb{R}^k$. Substituting the definition of $\mathbf{x}_t$ into Z_D:

$$\begin{aligned} Z_D &= \boldsymbol{\alpha}^T \mathbf{P}^T (\boldsymbol{\mu}_t + \boldsymbol{\epsilon}_t) + \boldsymbol{\beta}^T \mathbf{P}^T (\boldsymbol{\mu}_{t+1} + \boldsymbol{\epsilon}_{t+1}) \\ &= \mu_Z + S_D, \end{aligned} \tag{31}$$

where:

$$\begin{aligned} \mu_Z &= \boldsymbol{\alpha}^T \mathbf{P}^T \boldsymbol{\mu}_t + \boldsymbol{\beta}^T \mathbf{P}^T \boldsymbol{\mu}_{t+1} \\ S_D &= \boldsymbol{\alpha}^T \mathbf{P}^T \boldsymbol{\epsilon}_t + \boldsymbol{\beta}^T \mathbf{P}^T \boldsymbol{\epsilon}_{t+1}, \end{aligned}$$

here, μ_Z denotes the deterministic mean, and S_D represents the random fluctuation term. Since μ_Z is constant, we only need to prove that the random term S_D converges in distribution to $\mathcal{N}(0, \sigma_Z^2)$.

Define the coefficient vector $\boldsymbol{\gamma} \in \mathbb{R}^{2D}$ as $\boldsymbol{\gamma} = [(\mathbf{P}\boldsymbol{\alpha})^T, (\mathbf{P}\boldsymbol{\beta})^T]^T$. Then S_D can be rewritten as a weighted sum of joint noise components:

$$S_D = \sum_{j=1}^{2D} \gamma_j \xi_j, \tag{32}$$

Calculate the variance sequence B_D^2 of S_D:

$$B_D^2 = \mathrm{Var}(S_D) = \sum_{j=1}^{2D} \gamma_j^2 \sigma_j^2, \tag{33}$$

According to the Lindeberg-Feller central limit theorem (Feller, 1991) for sums of independent (or weakly dependent) non-identically distributed random variables, convergence to a normal distribution is guaranteed if the Lindeberg condition holds. We verify that for any $\tau > 0$:

$$L = \lim_{D\to\infty} \frac{1}{B_D^2} \sum_{j=1}^{2D} \mathbb{E}\left[(\gamma_j \xi_j)^2 \cdot \mathbb{I}(|\gamma_j \xi_j| > \tau B_D)\right] = 0. \tag{34}$$

From the boundedness in Assumption 1, we know $|\xi_j| \le M$, thus $|\gamma_j \xi_j| \le |\gamma_j| M$. The indicator function $\mathbb{I}(\cdot)$ is non-zero only if $|\gamma_j \xi_j| > \tau B_D$, which implies the necessary condition:

$$|\gamma_j| M > \tau B_D \iff \frac{|\gamma_j|}{B_D} > \frac{\tau}{M}. \tag{35}$$

However, according to Assumption 2:

$$\lim_{D\to\infty} \frac{\max_j |\gamma_j|}{B_D} = 0. \tag{36}$$

This implies that for sufficiently large D, given any fixed $\tau, M > 0$, there exists no index j such that $|\gamma_j| M > \tau B_D$ holds. Therefore, the indicator function $\mathbb{I}(|\gamma_j \xi_j| > \tau B_D)$ is identically zero. Consequently, the limit $L = 0$, and the Lindeberg condition is satisfied.

Since the Lindeberg condition is met, the random variable S_D converges in distribution to a scaled standard normal distribution:

$$\frac{S_D}{B_D} \xrightarrow{d} \mathcal{N}(0,1) \implies S_D \xrightarrow{d} \mathcal{N}(0, B_D^2). \tag{37}$$

Substituting back into the expression for Z_D, we have $Z_D \xrightarrow{d} \mathcal{N}(\mu_Z, B_D^2)$. Since $\boldsymbol{\lambda}$ is arbitrary, by the Cramér-Wold Theorem, the meta-semantic vector $\mathbf{x}_t$ converges in distribution to a multivariate Gaussian distribution. ■

Remark: Although the above theorem is derived based on the asymptotic $D \to \infty$, in our experimental setup, the original semantic space dimension D (e.g., 1024 for RoBERTa-large) satisfies the requirements for high-dimensional statistical approximation. Furthermore, the projection matrix $\mathbf{P}$ obtained via supervised subspace learning tends to utilize global semantic information. This results in projection coefficients γ_j exhibiting significant non-sparse characteristics, preventing any single original dimension from dominating the distribution. Therefore, modeling compact semantic features using a multivariate Gaussian distribution in Section 3.3 is not only theoretically grounded but also consistent with the statistical laws of high-dimensional data.

C.2.2 Argument for Model Distribution Selection based on Maximum Entropy Principle

Although the asymptotic normality of meta-semantic features has been proven above, in practical applications with finite dimensions, observed data may not perfectly follow a Gaussian distribution. This section proves from an information-theoretic perspective that, given observations of only the sample mean and covariance, the Gaussian distribution model is the optimal choice as it introduces the least prior bias.

Proposition 5. *For a random variable $\mathbf{x} \in \mathbb{R}^{2k}$, if we observe only its first moment and second moment from dataset $\mathcal{D}$, then among all probability distributions $p(\mathbf{x})$ satisfying these moment constraints, the multivariate Gaussian distribution has the maximum Differential Entropy.*

Proof: Our objective is to maximize the differential entropy $H(p)$:

$$\begin{aligned} \max_{p} \quad & -\int p(\mathbf{x}) \ln p(\mathbf{x}) d\mathbf{x} \\ \text{s.t.} \quad & \int p(\mathbf{x}) d\mathbf{x} = 1 \\ & \int \mathbf{x} p(\mathbf{x}) d\mathbf{x} = \boldsymbol{\mu} \\ & \int (\mathbf{x}-\boldsymbol{\mu})(\mathbf{x}-\boldsymbol{\mu})^T p(\mathbf{x}) d\mathbf{x} = \boldsymbol{\Sigma}. \end{aligned} \tag{38}$$

We construct the Lagrangian function:

$$\begin{aligned} \mathcal{L} = H(p) + \lambda_0 \Big(\int p - 1 \Big) + \boldsymbol{\lambda}_1^T \Big(\int \mathbf{x} p - \boldsymbol{\mu} \Big) \\ + \operatorname{Tr} \left(\boldsymbol{\Lambda}_2 \Big(\int (\mathbf{x}\mathbf{x}^T) p - (\boldsymbol{\Sigma} + \boldsymbol{\mu}\boldsymbol{\mu}^T) \Big) \right). \end{aligned} \tag{39}$$

Taking the derivative with respect to $p(\mathbf{x})$ using variational methods and setting it to zero, the form of $p(\mathbf{x})$ must be an exponential family distribution:

$$p^*(\mathbf{x}) = \exp\left(-1 + \lambda_0 + \boldsymbol{\lambda}_1^T \mathbf{x} + \mathbf{x}^T \boldsymbol{\Lambda}_2 \mathbf{x}\right). \tag{40}$$

Substituting the constraints to solve for the Lagrange multipliers, we finally obtain:

$$\begin{aligned} p^*(\mathbf{x}) = & \frac{1}{(2\pi)^k |\boldsymbol{\Sigma}|^{1/2}} \\ & \times \exp\left(-\frac{1}{2} (\mathbf{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x}-\boldsymbol{\mu}) \right), \end{aligned} \tag{41}$$

which is the multivariate Gaussian distribution. ■

Since we possess a large-scale dataset (e.g., 700,000 meta-structure pairs), by the Law of Large Numbers, we can estimate $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ with high precision. In the absence of prior knowledge regarding higher-order statistics of the data, adopting a Gaussian distribution model is equivalent to making the minimal assumption at the information-theoretic level. Any other distributional assumption would implicitly introduce additional structural information not supported by the data, thereby increasing the risk of model overfitting.

C.3 2-Wasserstein Distance Between Gaussian Distributions

Let $P_H^\pi = \mathcal{N}(\boldsymbol{\mu}_H^\pi, \boldsymbol{\Sigma}_H^\pi)$ and $P_M^\pi = \mathcal{N}(\boldsymbol{\mu}_M^\pi, \boldsymbol{\Sigma}_M^\pi)$ be two Gaussian distributions on $\mathbb{R}^{2k}$. The squared 2-Wasserstein distance is defined by the optimal transport problem:

$$W_2^2(P_H^\pi, P_M^\pi) = \inf_{\gamma \in \Pi(P_H^\pi, P_M^\pi)} \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\gamma} \left[\|\mathbf{x}-\mathbf{y}\|_2^2 \right]. \tag{42}$$

By expanding the squared Euclidean norm, we decompose the expectation based on the first and second moments:

$$\begin{aligned} \mathbb{E}[\|\mathbf{x}-\mathbf{y}\|_2^2] = \|\boldsymbol{\mu}_H^\pi - \boldsymbol{\mu}_M^\pi\|_2^2 + \mathbb{E}[\|\bar{\mathbf{x}}-\bar{\mathbf{y}}\|_2^2] \\ + 2(\boldsymbol{\mu}_H^\pi - \boldsymbol{\mu}_M^\pi)^\top \mathbb{E}[\bar{\mathbf{x}}-\bar{\mathbf{y}}], \end{aligned} \tag{43}$$

where $\bar{\mathbf{x}} = \mathbf{x} - \boldsymbol{\mu}_H^\pi$ and $\bar{\mathbf{y}} = \mathbf{y} - \boldsymbol{\mu}_M^\pi$ are centered random variables. Since $\mathbb{E}[\bar{\mathbf{x}}] = \mathbb{E}[\bar{\mathbf{y}}] = \mathbf{0}$, the cross-term vanishes. The problem reduces to finding the optimal coupling for the centered covariances:

$$W_2^2(P_H^\pi, P_M^\pi) = \|\boldsymbol{\mu}_H^\pi - \boldsymbol{\mu}_M^\pi\|_2^2 + \inf_{\gamma} \mathbb{E}[\|\bar{\mathbf{x}}-\bar{\mathbf{y}}\|_2^2]. \tag{44}$$

Minimizing the covariance term is equivalent to maximizing the correlation trace. According to the properties of optimal transport for Gaussian measures, the optimal value yields the Bures-Wasserstein metric. Combining this with the mean difference, we obtain the final metric w_π:

$$\begin{aligned} w_\pi = \Bigg(& \|\boldsymbol{\mu}_H^\pi - \boldsymbol{\mu}_M^\pi\|_2^2 + \\ & \operatorname{Tr}\left(\boldsymbol{\Sigma}_H^\pi + \boldsymbol{\Sigma}_M^\pi - 2(\boldsymbol{\Sigma}_H^{\pi\frac{1}{2}} \boldsymbol{\Sigma}_M^\pi \boldsymbol{\Sigma}_H^{\pi\frac{1}{2}})^{\frac{1}{2}} \right) \Bigg)^{1/2}. \end{aligned} \tag{45}$$

C.4 Proof of Wasserstein Discriminative Bound

This section provides the theoretical proof for the discriminative bound based on the Wasserstein distance discussed in Section 3.3 of the main text. We

demonstrate that for any function satisfying the Lipschitz continuity condition, the difference in its expectations over two distributions is strictly bounded by the 2-Wasserstein distance between these distributions.

Theorem 2 (Wasserstein Discriminative Bound)**.** *Let P_H and P_M be two probability measures on $\mathbb{R}^{2k}$. For any K-Lipschitz continuous discriminative function $f : \mathbb{R}^{2k} \to \mathbb{R}$, the difference in expectations between the two distributions satisfies:*

$$|\mathbb{E}_{\mathbf{x}\sim P_H}[f(\mathbf{x})] - \mathbb{E}_{\mathbf{y}\sim P_M}[f(\mathbf{y})]| \leq K \cdot W_2(P_H, P_M). \tag{46}$$

Proof: Let $\Pi(P_H, P_M)$ be the set of all couplingswhose marginals are P_H and P_M. For any coupling $\gamma \in \Pi(P_H, P_M)$, if the random variable pair $(\mathbf{X}, \mathbf{Y}) \sim \gamma$, then by the property of marginal distributions, $\mathbf{X} \sim P_H$ and $\mathbf{Y} \sim P_M$.

Consider the absolute difference of the expectations:

$$\begin{aligned}\Delta_f &= |\mathbb{E}_{\mathbf{X}\sim P_H}[f(\mathbf{X})] - \mathbb{E}_{\mathbf{Y}\sim P_M}[f(\mathbf{Y})]| \\ &= \left| \int f(\mathbf{x}) dP_H(\mathbf{x}) - \int f(\mathbf{y}) dP_M(\mathbf{y}) \right| \\ &= \left| \iint (f(\mathbf{x}) - f(\mathbf{y})) d\gamma(\mathbf{x}, \mathbf{y}) \right|.\end{aligned} \tag{47}$$

By the integral triangle inequality, we have:

$$\Delta_f \leq \iint |f(\mathbf{x}) - f(\mathbf{y})| d\gamma(\mathbf{x}, \mathbf{y}). \tag{48}$$

Utilizing the K-Lipschitz continuity condition of f:

$$\begin{aligned}\Delta_f &\leq \iint K\|\mathbf{x} - \mathbf{y}\|_2 d\gamma(\mathbf{x}, \mathbf{y}) \\ &= K \cdot \mathbb{E}_{(\mathbf{X},\mathbf{Y})\sim\gamma}[\|\mathbf{X} - \mathbf{Y}\|_2].\end{aligned} \tag{49}$$

According to Lyapunov's inequality, the first moment of a random variable is less than or equal to the square root of its second moment, i.e., $\mathbb{E}[Z] \leq (\mathbb{E}[Z^2])^{1/2}$. Applying this to the Euclidean distance $\|\mathbf{X} - \mathbf{Y}\|_2$:

$$\mathbb{E}[\|\mathbf{X} - \mathbf{Y}\|_2] \leq \left(\mathbb{E}[\|\mathbf{X} - \mathbf{Y}\|_2^2]\right)^{1/2}. \tag{50}$$

Therefore, for any $\gamma \in \Pi(P_H, P_M)$, it holds that:

$$\Delta_f \leq K \left(\iint \|\mathbf{x} - \mathbf{y}\|_2^2 d\gamma(\mathbf{x}, \mathbf{y}) \right)^{1/2}. \tag{51}$$

Note that the left side Δ_f is a constant dependent only on the marginals P_H and P_M, and is independent of the specific coupling γ. Thus, the inequality holds for any γ in $\Pi(P_H, P_M)$, and consequently holds for the infimum over γ on the right side:

$$\Delta_f \leq K \cdot \inf_{\gamma\in\Pi} \left(\mathbb{E}_\gamma[\|\mathbf{X} - \mathbf{Y}\|_2^2]\right)^{1/2}. \tag{52}$$

Due to the continuity and monotonicity of the square root function, $\inf(\mathbb{E}^{1/2}) = (\inf \mathbb{E})^{1/2}$. Combining this with the definition of the 2-Wasserstein distance $W_2(P_H, P_M) = \left(\inf_{\gamma\in\Pi} \mathbb{E}[\|\mathbf{X} - \mathbf{Y}\|_2^2]\right)^{1/2}$, we finally obtain:

$$|\mathbb{E}_{P_H}[f] - \mathbb{E}_{P_M}[f]| \leq K \cdot W_2(P_H, P_M). \tag{53}$$

This concludes the proof. ■

Remark: This theorem provides the theoretical basis for the weighting strategy presented in Section 3.3 of the main text. The term $W_2(P_H, P_M)$ on the right side measures the geometric separability between human and machine text in the semantic space under a specific syntactic structure π. Meanwhile, K represents the sensitivity of the discriminator to semantic perturbations (i.e., the smoothness of the classification surface). This bound indicates that the larger the W_2 distance between two distributions, the greater the theoretical expected difference any smooth discriminative function can achieve in distinguishing them.

C.5 Derivation of the Modified Mahalanobis Distance

This section derives the equivalence relationship between the log-likelihood ratio (LLR) test described in Section 3.4 of the main text and the modified Mahalanobis Distance.

Proposition 6 (Equivalence of LLR and Mahalanobis Distance)**.** *Let $P_H(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_H, \boldsymbol{\Sigma}_H)$ and $P_M(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_M, \boldsymbol{\Sigma}_M)$ be the feature distributions of human and machine text under a specific structure, respectively. Define the modified Mahalanobis distance as $\mathcal{M}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = (\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}) + \ln |\boldsymbol{\Sigma}|$. Then, for a sample $\mathbf{x}_t$, its log-likelihood ratio statistic s_t satisfies:*

$$\begin{aligned}& s_t = \ln \frac{P_M(\mathbf{x}_t)}{P_H(\mathbf{x}_t)} \\ \iff\ & s_t = \frac{1}{2}\Big[\mathcal{M}(\mathbf{x}_t; \boldsymbol{\mu}_H, \boldsymbol{\Sigma}_H) \\ & \qquad - \mathcal{M}(\mathbf{x}_t; \boldsymbol{\mu}_M, \boldsymbol{\Sigma}_M)\Big].\end{aligned} \tag{54}$$

Proof: For a d-dimensional multivariate Gaussian distribution $P(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$, the log-likelihood function is:

$$\begin{aligned}\ln P(\mathbf{x}) = &-\frac{d}{2}\ln(2\pi) - \frac{1}{2}\ln|\boldsymbol{\Sigma}| \\ &-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}).\end{aligned} \tag{55}$$

Expanding the log-likelihood ratio s_t:

$$\begin{aligned}s_t &= \ln P_M(\mathbf{x}_t) - \ln P_H(\mathbf{x}_t) \\ &= \left[-\frac{d}{2}\ln(2\pi) - \frac{1}{2}\ln|\boldsymbol{\Sigma}_M| - \frac{1}{2}Q_M(\mathbf{x}_t)\right] \\ &\quad - \left[-\frac{d}{2}\ln(2\pi) - \frac{1}{2}\ln|\boldsymbol{\Sigma}_H| - \frac{1}{2}Q_H(\mathbf{x}_t)\right],\end{aligned} \tag{56}$$

where $Q(\mathbf{x}_t) = (\mathbf{x}_t - \boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}_t - \boldsymbol{\mu})$ is the quadratic term.

Canceling the constant term $-\frac{d}{2}\ln(2\pi)$ and rearranging the terms, we obtain:

$$\begin{aligned}s_t = &\frac{1}{2}\left(Q_H(\mathbf{x}_t) + \ln|\boldsymbol{\Sigma}_H|\right) \\ &-\frac{1}{2}\left(Q_M(\mathbf{x}_t) + \ln|\boldsymbol{\Sigma}_M|\right).\end{aligned} \tag{57}$$

According to the definition of the modified Mahalanobis distance $\mathcal{M}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) \triangleq Q(\mathbf{x}) + \ln|\boldsymbol{\Sigma}|$, substituting this into the equation yields:

$$s_t = \frac{1}{2}\mathcal{M}(\mathbf{x}_t; \boldsymbol{\mu}_H, \boldsymbol{\Sigma}_H) - \frac{1}{2}\mathcal{M}(\mathbf{x}_t; \boldsymbol{\mu}_M, \boldsymbol{\Sigma}_M). \tag{58}$$

This concludes the proof. ■

Theorem 1 requires the discriminative function $f(\mathbf{x})$ to satisfy the K-Lipschitz continuity condition. Although the Mahalanobis distance, as a quadratic function, does not possess global Lipschitz properties over the entire $\mathbb{R}^{2k}$ space, within the framework of our method, the domain of the feature variable $\mathbf{x}$ is bounded, thereby guaranteeing that it satisfies the Lipschitz condition.

The reasoning is as follows based on three conditions. (1) Input boundedness: The meta-semantic vector $\mathbf{x}_t$ is formed by concatenating adjacent projected features: $\mathbf{x}_t = [\mathbf{v}_t; \mathbf{v}_{t+1}]$. The projected feature $\mathbf{v} = \mathbf{P}^T\mathbf{e}$ is obtained by transforming the original semantic embedding $\mathbf{e}$ via the probe matrix $\mathbf{P} = [\mathbf{p}_1, \ldots, \mathbf{p}_k]$. (2) Original space boundedness: The output embeddings $\mathbf{e}$ of modern LLMs typically undergo layer normalization, ensuring their Euclidean norms are bounded. That is, there exists a constant R_e such that $\|\mathbf{e}\|_2 \leq R_e$. (3) Projection constraint: In the optimization objective of Section 3.2, we explicitly impose a unit norm constraint on the probe vectors: $\|\mathbf{p}_i\|_2 = 1$.

By applying the Cauchy-Schwarz inequality (Anderson et al., 1958), the norm of the projected features satisfies:

$$\|\mathbf{v}\|_2 = \|\mathbf{P}^T\mathbf{e}\|_2 \leq \|\mathbf{P}\|_F\|\mathbf{e}\|_2 \leq \sqrt{k}\cdot 1 \cdot R_e. \tag{59}$$

Therefore, the meta-semantic vector lies within a compact subset $\mathcal{C}$ of the space $\mathbb{R}^{2k}$. Consider the gradient of the discriminative function $f(\mathbf{x})$:

$$\nabla f(\mathbf{x}) = \boldsymbol{\Sigma}_H^{-1}(\mathbf{x} - \boldsymbol{\mu}_H) - \boldsymbol{\Sigma}_M^{-1}(\mathbf{x} - \boldsymbol{\mu}_M). \tag{60}$$

Since $\mathbf{x} \in \mathcal{C}$ is bounded and $\boldsymbol{\Sigma}_H, \boldsymbol{\Sigma}_M$ are positive definite matrices, the gradient norm $\|\nabla f(\mathbf{x})\|_2$ has an upper bound K_{max} on the region $\mathcal{C}$. According to the Mean Value Theorem, if the gradient of a function is bounded on a convex compact set, then the function is Lipschitz continuous.

D Benchmark and Baselines Details

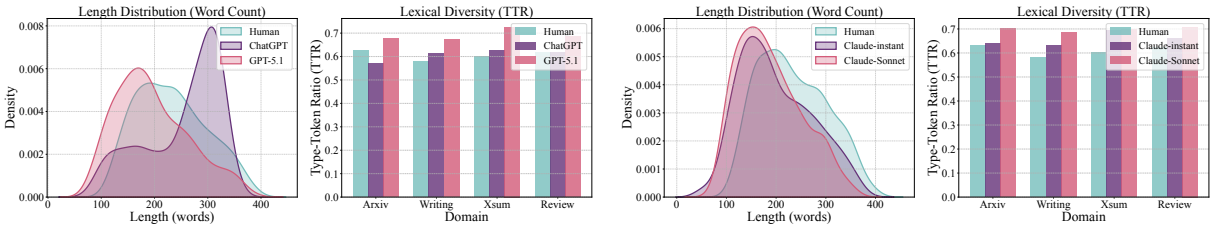
D.1 Main Benchmark Construction and Descriptive Statistics

D.1.1 Benchmark Construction and Quality Control

To ensure comparability and rigor amidst rapid model iteration, we strictly followed the DetectRL protocol proposed by (Wu et al., 2024). for dataset construction. We also enforced equivalent standards of manual review during the post-processing stage.

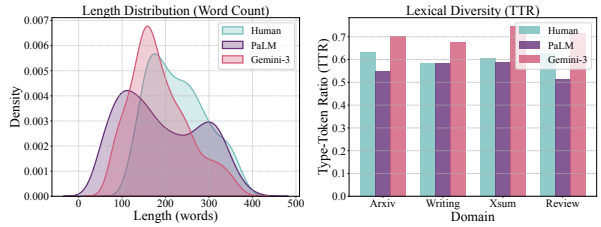
Human-Written Text Curation. We completely replicated the original data configuration, covering four domains prone to misuse: academic writing (ArXiv Abstracts), news reports (XSum), creative writing (Writing Prompts), and social media reviews (Yelp Reviews). To avoid data contamination from LLMs, all selected human samples date from the pre-ChatGPT era. Each domain contains 2,800 filtered high-quality samples, constituting a reliable detection baseline.

SOTA Model Generation Strategy. For the construction of MGT, we upgraded the generation sources to current SOTA models: GPT-5.1, Claude-Sonnet-4, and Gemini-3-Flash, Grok-4.1. Abandoning simple unconstrained generation, we strictly adhered to the task-specific conditional generation strategy from the original paper. Carefully

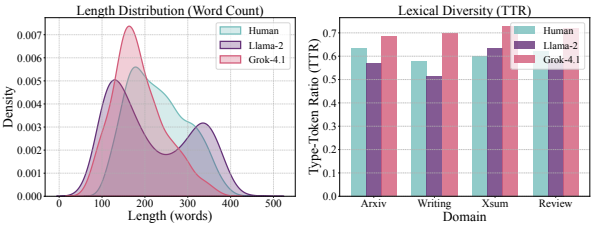


(a) Length distribution and lexical diversity: GPT-5.1 vs. ChatGPT.

(b) Length distribution and lexical diversity: Claude-Sonnet vs. Claude-instant.



(c) Length distribution and lexical diversity: Gemini-3 vs. PaLM.



(d) Length distribution and lexical diversity: Grok-4.1 vs. Llama-2.

Figure 8: **Overview of length distributions and lexical diversity comparisons.** Each subplot illustrates the comparative analysis of human text, old models, and new models across different datasets, visualizing the kernel density estimation (KDE) for length and type-token ratio (TTR) for diversity.

designed prompts forced the models to generate content strictly aligned with the length of the corresponding HWT within specific contexts.

Quality Control & Data Partitioning. We implemented a multi-stage quality control process to ensure dataset safety and academic rigor. First, we automatically filtered invalid samples with insufficient length during the preprocessing stage. Second, despite the automated generation process, we introduced a manual sampling review mechanism before storage. This aimed to identify and remove samples with potential logical breakdowns, repetition, or offensive content. Regarding data partitioning, we followed the original protocol (Random Seed = 42), splitting the data into a training set (1,800 HWT/MGT pairs) and a test set (1,000 HWT/MGT pairs). Stratified sampling was employed to ensure the proportions of domains and model sources in each subset remained consistent with the original paper.

D.1.2 Descriptive Statistics of the Benchmark

Figure 8 illustrates the comparison of Length distribution and lexical diversity across four domains. It compares our constructed datasets with original model-generated texts and human texts.

First, kernel density estimation (KDE) results for Length Distribution demonstrate that SOTA MGT effectively covers the HWT length range while maintaining a more concentrated distribution. As illustrated, the probability densities for both GPT-5.1 and Grok-4.1 fall primarily between 50 and 400 words, consistent with the HWT span. Notably, in contrast to baseline models which exhibit irregularities such as the abnormal spike at approximately 300 words in ChatGPT or the bimodal distribution in Llama-2, the SOTA models display smoother, unimodal curves. This high degree of distributional alignment effectively diminishes length as a trivial discriminatory feature. It ensures that models accurately simulate human writing patterns across diverse domains, from the conciseness of academic abstracts to the extensiveness of creative writing, thereby guaranteeing the fairness of the detection task.

Second, regarding lexical diversity, type-token ratio (TTR) statistics indicate that current SOTA MGT surpasses the human baseline in vocabulary richness. Observing the TTR bar charts, models such as GPT-5.1 and Claude-Sonnet generally show higher TTR values than HWT in domains like ArXiv and XSum. Particularly in the Writing Prompts domain, the models do not exhibit the monotony common in traditional generation. Their lexical diversity metrics significantly outperform early models like PaLM or Claude-instant in multiple samples. This data characteristic suggests that advanced models utilize broader vocabulary distributions during generation, rendering simple detection methods based on vocabulary repetition

ineffective on this dataset. Furthermore, the slightly higher lexical diversity of generated text does not imply a decline in data quality or semantic divergence. Conversely, this characteristic positively reflects that SOTA models, under generation constraints, have successfully avoided common mode collapse and repetitive degeneration. This indicates that the generated text possesses complex syntactic structures and rich word choices, accurately simulating high-stealth, high-deception MGT in the current environment.

D.2 Supplementary Evaluation Data Construction

To further examine the robustness and generalization ability of detection methods beyond the main benchmark, we constructed three supplementary evaluation settings, covering multilingual transfer, informal and noisy inputs, and highly creative or dynamically mixed text distributions. These supplementary subsets were designed to probe model behavior under conditions that are closer to realistic deployment scenarios while preserving comparability with the main benchmark.

D.2.1 Construction of Multilingual Evaluation Data

We built a multilingual evaluation subset based on M4GT-Bench (Wang et al., 2024). This benchmark is a multilingual, multi-domain, and multi-generator evaluation suite for machine-generated text detection, designed to systematically assess the generalization ability of detection methods under more complex and realistic application conditions. Its task settings not only cover monolingual and multilingual binary detection, but also further include generator attribution and human–machine mixed-boundary identification. As such, it provides a relatively strict and representative evaluation basis for supplementing the cross-lingual robustness analysis in this work. Prior studies have shown that achieving stable performance on such benchmarks typically requires a detector to adapt simultaneously to variations in language, domain, and generation source, rather than relying solely on surface-level statistical patterns in a single language.

On this basis, we selected four languages from M4GT-Bench, including German, Italian, Russian, and Chinese, to construct the multilingual evaluation data. Specifically, for each language, we randomly sampled 2,000 instances under a fixed random seed of 42, and further split them into 1,400 training samples and 600 test samples, thereby ensuring fair comparison across methods under a consistent data configuration. This construction preserves the original multilingual characteristics and evaluation difficulty of M4GT-Bench, while also providing a unified experimental basis for analyzing model stability across different language families.

In terms of experimental implementation, to accommodate multilingual testing, all RoBERTa-based methods uniformly adopted XLM-RoBERTa-base (Conneau et al., 2019). DetectAnyLLM and Fast-DetectGPT both used Qwen2.5-7B-Instruct (Team, 2024) as the backbone model. For our method, we employed the corresponding small spaCy language models (`core_web_sm`) for each language to perform syntactic annotation, and used XLM-RoBERTa-base as the semantic embedding model. In this way, we ensured fairness and reproducibility across methods.

D.2.2 Construction of Informal and Noisy Text Data

To evaluate robustness under informal and highly noisy text conditions, we constructed Informal and Noisy variants on the Yelp Review and Writing Prompt test sets, both of which are closer to everyday scenarios involving non-canonical expression. Specifically, we did not alter the source, labels, or core semantics of the samples, but only applied controlled surface-level perturbations to the original text, in order to simulate realistic inputs such as online reviews, colloquial writing, and severely degraded user-generated content. The perturbation process was conducted under a unified random seed of 42 and covered multiple intensity levels ranging from mild corruption to extreme distortion. Following the experimental setup, we considered four types of perturbations: *typos*, *word swapping*, *formatting degradation*, and *mixed noise*. The noise ratios were set to $0.00, 0.05, 0.10, 0.15, 0.20$, and 0.30, so as to gradually increase the irregularity of the input text.

More specifically, we considered four types of perturbations:

- *Typos*: character-level spelling perturbations were applied to a randomly selected subset of words in each sentence, thereby simulating common misspellings, omitted characters, incorrect substitutions, and keyboard-adjacent

Table 5: Illustrative examples of controlled perturbations used in the Informal and Noisy setting.

Perturbation Type	Example Text
Original Text	I really enjoyed this restaurant because the service was fast and the staff were very friendly.
Typos	I relaly enojyed this restuarant because the service was fsat and the staff were very freindly.
Word Swapping	I really this enjoyed restaurant because the service fast was and the staff were friendly very.
Formatting Degradation	i really enjoyed this restaurant because the service was fast and the staff were very friendly
Mixed Noise	i relaly enjoyed this restuarant because service the was fsat and staff were freindly very

replacements frequently observed in social media and user reviews.

- *Word swapping*: the positions of local neighboring words were randomly exchanged, so as to mimic local word-order disorder caused by colloquial expression, hurried typing, or low-quality transcription.

- *Formatting degradation*: the surface writing conventions of the text were deliberately disrupted, including deleting or misplacing punctuation, perturbing capitalization patterns, and weakening the original sentence boundaries, thereby simulating informal inputs that lack editing discipline.

- *Mixed noise*: spelling errors, local word-order swaps, and formatting degradation were combined into a single corruption process, yielding more extreme non-canonical text conditions.

Under higher noise ratios, such samples often deviate substantially from standard written language while still preserving basic recognizable semantic fragments, making them more suitable for probing the robustness boundary of detection models in complex noisy environments.

Table 5 presents representative examples under different perturbation types. As can be seen, these perturbations do not alter the overall semantic topic of the text, but systematically damage spelling, local word order, and writing format, thereby progressively increasing the degree of input irregularity.

All baseline methods and our method were trained on the clean, unperturbed training set. Moreover, all model choices, parameter settings, and evaluation protocols were kept identical to those used in the main benchmark experiments.

D.2.3 Construction of Poetic, Dynamic-Mixed, and Decoding-Variant Text Data

Poetry Test Set. To evaluate the generalization ability of detection methods under highly creative and strongly irregular text conditions, we constructed a fully out-of-distribution poetry evaluation set. Specifically, we randomly sampled 500 poetry instances satisfying the minimum length constraint from the Poetry-Categorized [3] corpus on HuggingFace under a fixed random seed of 42, and retained the original human-written poems as the human side of the dataset. We then first used an auxiliary model to extract a concise thematic description for each poem, and subsequently used this description as a unified semantic condition to prompt GPT-5.1, Claude-Sonnet-4, Gemini-3-Flash, and Grok-4.1 to regenerate the corresponding English poem. For the generated outputs, we further performed text cleaning and length alignment, so that the machine-generated poems remained approximately matched in length to their corresponding human poems, thereby minimizing the influence of shallow statistical cues such as text length on the detection results. Ultimately, each generator was paired with the same set of human poems, resulting in a structurally consistent poetry test set. All baseline methods and our method were trained directly on the training set of the main benchmark, and all model configurations, parameter settings, and testing procedures were kept identical to those in the main benchmark experiments.

Dynamic Mixed Test Set. To further evaluate adaptability under rapidly changing environments and mixed-model distributions, we additionally constructed a dynamic mixed test set. This test set was sampled from the human-written texts in the main benchmark, with 250 instances selected under the fixed random seed of 42. Taking the original human texts as semantic references, and while keeping the overall length and topic approximately aligned, we no longer allowed the machine-generated text to be produced independently by

[3] https://huggingface.co/datasets/schifferlearning/Poetry-Categorized

a single generator. Instead, multiple current advanced generators were required to alternately continue the text at the sentence level, thereby deliberately introducing frequent local probability distribution shifts within a single sample. Concretely, each machine-generated text was decomposed into multiple sentence-level generation units, and GPT-5.1, Claude-Sonnet-4, Gemini-3-Flash, and Grok-4.1 were used in a predefined rotation order to generate these units, which were then concatenated into a complete text. This process produced machine-generated samples with pronounced mixed-decoding characteristics. Through this construction, a single sample simultaneously contains local stylistic discrepancies and decoding offsets from different generators, thus more faithfully simulating dynamic text streams in real-world applications where sources are mixed and distributions drift rapidly. All baseline methods and our method were likewise trained on the main benchmark training set, with the same model choices, parameter settings, and evaluation protocols as those used in the main benchmark.

Temperature-Varied Test Sets. To evaluate the sensitivity of detection methods to changes in decoding strategy, we additionally reconstructed machine-generated test sets under different temperature settings while keeping the human-written texts in the main benchmark fixed. Specifically, the human texts in the main benchmark were still used as semantic references, and the same prompt templates, text cleaning pipeline, and length alignment strategy as in the main experiments were retained. The only generation factor varied was the temperature parameter, which was set to 0.1, 0.4, 0.7, and 1.0. Under each setting, machine-generated texts were produced using GPT-5.1, Claude-Sonnet-4, Gemini-3-Flash, and Grok-4.1. Apart from decoding temperature, all other data construction procedures, evaluation protocols, training settings, and parameter estimation processes were kept identical to those of the main benchmark. This design ensures that the resulting experiments primarily capture the effect of decoding randomness on detection performance, without interference from other confounding factors.

D.3 Baseline Implementation Details

To establish a comprehensive benchmark, we compare ProSSD against a diverse set of state-of-the-art methods, including those published within the last three months, covering both statistical detection methods and training-based approaches. The specific introductions and experimental configurations for each baseline are as follows:

- **Log-Likelihood Log-Rank Ratio (LRR)** (Su et al., 2023): This method utilizes log-rank information to distinguish between human and machine-generated texts, based on the hypothesis that MGT exhibits specific statistical properties in rank distribution. In our experiments, we employ GPT-Neo-2.7B as the base scoring model.
- **Normalized Perturbed Log-Rank (NPR)** (Su et al., 2023): As an enhanced version of LRR, NPR introduces text perturbations to quantify the sensitivity of log-rank scores. Although computationally more expensive, it achieves higher detection precision. We configure GPT-Neo-2.7B as the scoring model and use T5-small to generate perturbed samples, setting 100 perturbations per sample.
- **DetectGPT** (Mitchell et al., 2023): This method operates on the hypothesis that text generated by **LLMs** tends to reside in the negative curvature regions of the model's log-probability function. It performs detection by comparing the log-probability differences between the original and perturbed texts. We use GPT-Neo-2.7B as the base model and T5-small as the mask-filling model to generate perturbations, with 100 perturbations per sample.
- **Fast-DetectGPT** (Bao et al., 2024): This method utilizes conditional probability curvature and an efficient sampling strategy to replace the perturbation steps in DetectGPT, maintaining high precision while improving inference speed. Following the standard settings of the original paper, we use GPT-Neo-2.7B as the scoring model and GPT-J-6B as the reference model.
- **Binoculars** (Hans et al., 2024): This is a low false-positive rate zero-shot detection method that identifies AI text by contrasting the perplexity of an "Observer" model with the cross-perplexity of a "Performer" model. In our configuration, we select Falcon-7b and Falcon-7b-instruct as the Observer and Performer models, respectively.

- **RoBERTa-base** (Solaiman et al., 2019): As a classic supervised baseline, RoBERTa detects machine-generated text by fine-tuning a binary classifier on labeled datasets. In our experiments, we build upon the open-source `RoBERTa-base-openai-detector` checkpoint and further fine-tune it on the source-domain training data for evaluation. The training settings are as follows: 2 epochs, a learning rate of 1×10^{-6}, a batch size of 32, and a random seed of 42.

- **AdaDetectGPT** (Zhou et al., 2025): This method introduces an adaptive classifier to learn a witness function, thereby optimizing the performance of logits-based detectors. We use GPT-Neo-2.7B for scoring and GPT-J-6B for sampling. The model is trained on source domain data using the same sampled data as our method. Following the optimal settings of the original paper, we employ B-splines as basis functions to learn the witness function.

- **Imitate Before Detect (ImBD)** (Chen et al., 2025a): ImBD proposes a Style Preference Optimization (SPO) strategy to mimic the style distribution of machine generation, detecting rewritten text by comparing distribution differences. We load the official GPT-Neo based inference checkpoint for testing. In OOD experiments, the model is trained on source domain data with parameters strictly following the original optimal settings: a learning rate of 0.0001, a beta coefficient of 0.05, and LoRA parameters including a rank of 8, an alpha of 32, and a dropout rate of 0.1.

- **DetectAnyLLM** (Fu et al., 2025): This framework utilizes a Direct Discrepancy Learning (DDL) strategy, enabling the detector to better capture core task semantics and addressing the mismatch between training objectives and task requirements. We evaluate the model after training DDL on GPT-Neo-2.7B based on the optimal configuration. In OOD experiments, the model is trained on source domain data with parameters consistent with the ImBD method mentioned above, with the additional hyperparameter γ set to 100.

- **RepreGuard** (Chen et al., 2025b): This method posits that the internal representations of **LLMs** contain richer primitive features than logits alone. It performs classification by calculating projection scores of text representations onto specific feature directions. Following the original paper, we select Llama-3-8B-Instruct as the proxy models to extract features and use the same samples as our method to calculate projection vectors. In the OOD setting, the model constructs feature directions based on source domain data.

E Supplementary Results and Analyses

E.1 Generalization under Distribution Shift

E.1.1 Supplementary OOD Evaluation

Cross model robustness. As shown in Table 6, we evaluated the out of distribution performance of the models to examine their generalization capabilities when facing unknown data distributions that differ from the training set. Compared to the in domain results in Table 1 of the main text, although RepreGuard and DetectAnyLLM perform excellently in the source setting, they exhibit significant instability in out of distribution tests. Specifically, when the detector is trained on data generated by Claude instant and directly used to detect text generated by GPT 5.1 (right side of Table 6), the F1 score of RepreGuard drops sharply to 40.23%. In contrast, ProSSD maintains an F1 score of 88.44% and an AUROC of 96.22% under the same setting. Overall, whether trained on Google PaLM or Claude instant, ProSSD maintains an AUROC above 94% on all four target test models (Claude S 4, Gemini 3 F, GPT 5.1, Grok 4.1), demonstrating the best robustness.

Cross domain adaptability. Cross domain evaluation examines the ability of the model to handle huge differences in vocabulary and semantic distributions. The task involves training or constructing distributions on dataset A and testing on dataset B. Tables 7 and 8 report the relevant results. As shown in Table 7, although DetectAnyLLM and RepreGuard score slightly higher than ProSSD in individual specific transfer scenarios (such as Writing Prompts $\rightarrow$ Arxiv and Arxiv $\rightarrow$ XSum), the gap is minimal. More importantly, judging from the overall scores in Table 7 and Table 8, ProSSD exhibits the lowest performance variance, powerfully proving its strong generalization ability. Combining the above out of distribution analyses, the core advantage of ProSSD lies not in defeating all baselines on every single metric, but in its ability to

Table 6: Cross-model generalization results with different training sources. The table evaluates detection performance when the training data is generated by Google-PaLM (Left) and Claude-instant (Right). The detectors are tested on four unseen target LLMs to assess robustness across different generator architectures. Results are reported as AUROC(%) and F1(%). Best and second-best results are highlighted in **bold** and underlined.

Method	Train on Google-PaLM								Train on Claude-instant							
	GPT-5.1		Claude-S-4		Gemini-3-F		Grok-4.1		GPT-5.1		Claude-S-4		Gemini-3-F		Grok-4.1	
	AUR.	F1	AUR.	F1	AUR.	F1	AUR.	F1	AUR.	F1	AUR.	F1	AUR.	F1	AUR.	F1
LRR	68.02	58.08	79.62	71.13	63.30	59.10	62.18	49.50	68.02	58.08	79.62	71.13	63.30	59.10	62.18	49.50
Fast-DetectGPT	67.07	58.59	78.78	72.73	76.13	68.44	52.91	42.29	67.07	58.59	78.78	72.73	76.13	68.44	52.91	42.29
RoBERTa	79.72	71.80	89.84	81.64	93.91	87.00	88.42	80.19	60.69	55.11	68.04	65.76	70.09	70.88	58.71	66.10
ImBD	90.63	83.26	93.72	84.97	94.90	88.66	85.70	77.83	92.67	84.74	96.62	89.94	95.65	90.29	87.83	81.58
AdaDetectGPT	54.40	55.33	63.59	63.33	61.22	57.07	41.37	66.67	61.47	54.74	73.25	69.41	68.95	59.10	47.08	21.61
DetectAnyLLM	76.96	82.48	81.02	85.14	79.41	85.33	78.34	84.14	83.42	74.66	92.04	84.89	89.86	81.82	80.04	73.50
RepreGuard	69.88	64.32	81.69	75.97	73.04	67.24	69.40	66.63	58.56	40.23	68.66	58.41	62.80	46.15	62.02	39.91
ProSSD	**94.87**	**88.14**	**97.07**	**90.31**	**98.34**	**93.98**	**96.21**	**89.22**	**96.22**	**88.44**	**98.53**	**93.74**	**97.89**	**92.69**	**97.65**	**91.60**

Table 7: Cross-domain detection performance using different training sources. We report the AUROC (%) and F1 (%) scores when the detector is trained on the Writing-prompt(Left) and Arxiv (Right) domains and evaluated on other unseen text domains.The best and second-best results in each column are highlighted in **bold** and underlined, respectively.

Method	Train on Writing-prompt						Train on Arxiv					
	Arxiv		XSum		Review		Writing		XSum		Review	
	AUR.	F1	AUR.	F1	AUR.	F1	AUR.	F1	AUR.	F1	AUR.	F1
LRR	73.90	67.71	70.14	63.49	84.01	75.11	48.20	14.90	70.14	63.49	84.01	75.11
Fast-DetectGPT	74.81	64.93	57.65	50.42	82.92	75.56	58.88	47.29	57.65	50.42	82.92	75.56
RoBERTa	89.88	82.87	76.31	66.41	94.07	87.72	85.60	78.37	74.45	66.77	94.53	87.83
ImBD	97.55	92.42	84.52	77.02	99.36	96.52	90.54	82.18	94.24	87.63	96.98	91.22
AdaDetectGPT	46.40	1.19	38.87	0.00	47.31	66.67	59.73	51.70	62.46	59.31	83.78	75.14
DetectAnyLLM	**99.91**	**98.59**	90.62	83.23	97.99	93.72	92.68	86.19	**100.00**	**99.95**	97.28	92.09
RepreGuard	99.07	95.53	94.07	86.36	99.20	96.37	**99.83**	**98.75**	99.95	99.10	**99.87**	**98.24**
ProSSD	97.83	94.03	**99.54**	**97.60**	**99.92**	**99.00**	97.27	92.34	97.93	93.66	98.85	95.20

provide a reliable performance lower bound under different training sources. This characteristic makes it more practically valuable when facing complex and changeable unknown data in the real world.

E.1.2 Adaptation to Environmental Changes

In the dynamic mixture setting, the evaluation difficulty is further elevated from inter-sample distribution shift to frequent intra-sample distribution switching. Specifically, different generators alternately continue the same text at the sentence level, causing local lexical preferences, syntactic habits, and decoding probability patterns to shift continuously within a single sample. This setting is substantially more challenging than conventional cross-model transfer, and more faithfully reflects the complex text-stream scenarios encountered in real-world applications. As shown in Table 9, ProSSD remains remarkably stable under such strong perturbations, achieving the best overall performance on Arxiv, Writing, and Review, while remaining nearly on par with the strongest result on XSum. Particularly noteworthy is the more open and volatile Writing domain, where Fast-DetectGPT attains an F1 of only 6.15%, and DetectAnyLLM also drops to 80.95%, whereas ProSSD still maintains 100.00% AUROC and 100.00% F1. This result indicates that the discriminative mechanism of ProSSD does not rely on relatively smooth and continuous surface-level probabilistic cues internal to any single generator.

From the overall trend, the impact of dynamic mixed distributions on baseline methods is highly selective. DetectAnyLLM remains relatively competitive on more structurally regular domains, but exhibits a noticeable decline on open-ended and subjective texts. Fast-DetectGPT, by contrast, becomes severely unstable under mixed distributions, suggesting that detection strategies relying on lo-

Table 8: Cross-domain detection performance using different training sources. We report the AUROC (%) and F1 (%) scores when the detector is trained on the XSum(Left) and Review(Right) domains and evaluated on other unseen text domains.The best and second-best results in each column are highlighted in **bold** and underlined, respectively.

Method	Train on XSum						Train on Review					
	Arxiv		Writing		Review		Arxiv		Writing		XSum	
	AUR.	F1	AUR.	F1	AUR.	F1	AUR.	F1	AUR.	F1	AUR.	F1
LRR	73.90	67.71	48.20	14.90	84.01	75.11	73.90	67.71	48.20	14.90	70.14	63.49
Fast-DetectGPT	74.81	64.93	58.88	47.29	82.92	75.56	74.81	64.93	58.88	47.29	57.65	50.42
RoBERTa	88.38	80.72	70.82	66.86	86.86	77.85	79.69	72.68	91.06	82.33	67.00	66.17
ImBD	98.98	94.98	86.00	77.14	96.66	90.38	97.26	91.26	94.73	87.52	90.90	82.90
AdaDetectGPT	67.23	58.35	55.27	39.20	79.36	72.99	60.77	49.58	52.94	31.95	50.04	12.66
DetectAnyLLM	98.79	95.18	79.39	73.27	97.23	91.35	**99.77**	**98.61**	99.74	98.10	**99.80**	**99.50**
RepreGuard	97.34	92.58	69.32	66.12	96.27	90.18	99.73	98.29	**99.87**	**98.85**	99.54	98.10
ProSSD	**99.83**	**98.55**	**96.39**	**91.05**	**99.25**	**95.89**	98.98	95.51	99.52	97.09	99.72	99.25

Table 9: Performance on the dynamic mixed test set across domains. We report AUC and F1 (%). Best results are marked in **bold** and second-best in underlined.

Method	Arxiv		Writing	
	AUROC	F1	AUROC	F1
DetectAnyLLM	98.92	96.06	87.15	80.95
Fast-DetectGPT	67.40	66.15	34.64	6.15
ProSSD	**99.65**	**99.20**	**100.00**	**100.00**
Method	**Review**		**XSum**	
	AUROC	F1	AUROC	F1
DetectAnyLLM	87.98	82.26	**100.00**	**100.00**
Fast-DetectGPT	78.30	72.44	35.95	23.08
ProSSD	**99.71**	**97.52**	99.97	99.19

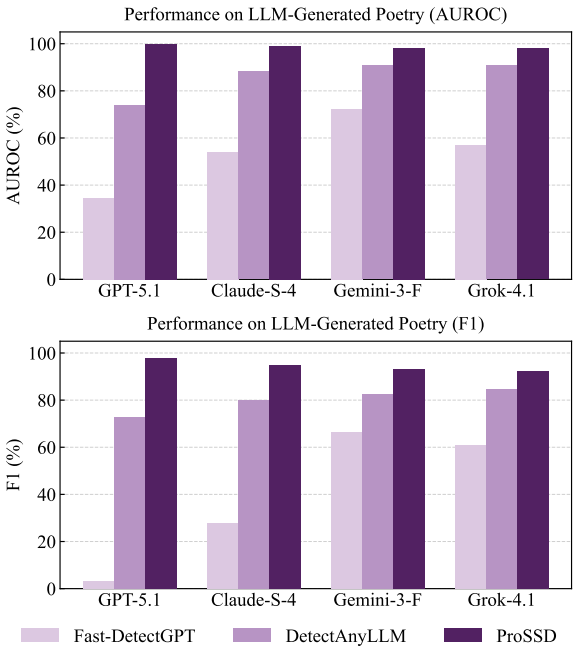


Figure 9: Comparison of detector performance on LLM-generated poetry across different generators. ProSSD consistently achieves near-saturated AUROC and F1 across all settings, demonstrating strong robustness under highly irregular and stylistically unconstrained text conditions.

cal conditional probability patterns are more easily disrupted in generation environments with frequent switching. In contrast, ProSSD consistently preserves near-saturated detection performance across all four domains. This demonstrates that the joint semantic-structural statistics modeled by ProSSD can transcend local stylistic discrepancies across generators and stably capture the deeper shared shifts induced by machine generation. These results further confirm that the advantage of ProSSD is not limited to static OOD generalization, but extends to rapidly changing environments and mixed-model distributions, where it continues to provide robust and reliable detection capability.

E.1.3 Evaluation on Highly Creative Texts

In highly creative and strongly irregular poetic scenarios, texts often deviate substantially from the grammatical organization, sentence-length distribution, and semantic progression patterns of conventional prose, thereby imposing much stricter requirements on the generalization ability of detection methods. As illustrated in Figure 9, ProSSD consistently maintains a clear advantage across all four models: its AUROC remains within 98.00%–99.81%, while its F1 remains within 92.50%–97.80%, demonstrating remarkable consistency and stability.

By comparison, both categories of baseline methods degrade more noticeably on the poetry benchmark. Although DetectAnyLLM retains a certain

Table 10: Adversarial robustness comparison of all detectors. We report detection performance on the original data(Direct Prompt) and against three adversarial attack scenarios(Paraphrase, Perturbation, and Data Mixing). The best results are marked in **bold** and second-best in underlined.

Method	Paraphrase		Perturbation		Direct Prompt		Data Mixing	
	AUR.	F1	AUR.	F1	AUR.	F1	AUR.	F1
LRR	70.39	60.11	97.97	94.66	86.41	78.78	82.78	72.61
NPR	72.96	66.63	98.60	95.69	80.26	72.68	73.42	63.16
RoBERTa	75.73	65.31	66.82	52.95	78.81	64.99	78.78	68.08
DetectGPT	47.58	22.65	88.11	79.40	53.78	41.04	32.63	0.00
Binoculars	81.78	73.11	97.94	94.22	94.93	90.07	93.89	88.43
Fast-DetectGPT	76.33	68.50	93.44	86.72	86.24	79.95	86.52	78.41
ImBD	88.25	79.29	97.48	90.41	93.61	84.48	90.93	80.28
AdaDetectGPT	75.40	64.06	39.20	0.79	84.72	78.44	84.72	76.58
DetectAnyLLM	86.00	76.67	87.38	76.55	86.80	76.40	88.89	76.75
RepreGuard	95.51	89.01	**99.57**	**97.86**	97.31	92.52	98.19	94.99
ProSSD	**97.83**	**93.36**	99.21	96.35	**98.25**	**93.44**	**99.13**	**95.97**

degree of discriminative ability on some generators, it still exhibits a consistent gap relative to ProSSD overall. The fluctuation of Fast-DetectGPT is even more pronounced. For instance, on GPT-5.1 and Claude-S-4, its F1 drops to only 3.11% and 27.85%, respectively, indicating that it is highly sensitive to the irregular expressions characteristic of creative texts. This phenomenon suggests that poetic writing substantially weakens methods built upon surface continuity or local curvature cues, whereas ProSSD, by virtue of joint distribution modeling in the projected subspace, does not depend on any single shallow feature. Taken together, these results show that ProSSD is not only effective on conventional formal texts, but also maintains stable detection performance under extreme textual conditions with strong stylization and strong irregularity.

E.2 Supplementary Robustness Analyses

E.2.1 Supplementary Evaluation on Adversarial Attacks

To comprehensively assess the security of the detectors in realistic adversarial environments, we evaluated their performance across four distinct attack scenarios: paraphrase, perturbation, direct prompt, and data mixing. Table 10 presents the comparative results. In general, ProSSD demonstrates the most consistent and superior defense capabilities, achieving the best performance in three out of four scenarios.

As shown in Table 10, while RepreGuard maintains strong competitiveness, particularly achieving the highest AUROC of 99.57% in the perturbation setting, ProSSD exhibits better stability across semantic alterations. Specifically, in the paraphrase scenario, which involves substantial rewriting and semantic restructuring, classic methods like DetectGPT and RoBERTa suffer significant performance degradation, with DetectGPT dropping to an F1 of 22.65%. Although RepreGuard performs well, ProSSD further improves the AUROC to 97.83% and F1 to 93.36%. This suggests that our method, by modeling the joint semantic-structural distribution, successfully captures the deep invariant features of machine generation that survive surface level rewriting.

Furthermore, in the data mixing scenario, where machine text is interleaved with human text, ProSSD achieves a dominant AUROC of 99.13%, outperforming the second best method RepreGuard by roughly 1 percentage point and significantly surpassing LRR and NPR. It is worth noting that methods heavily reliant on specific probability curvature or noise sensitivity (such as AdaDetectGPT) exhibit extreme volatility in the perturbation setting, whereas ProSSD maintains a high AUROC of 99.21%. Similar to the findings in the out of distribution analysis, the core advantage of ProSSD lies in its extremely low variance across different attack types. Whether facing simple character level perturbations or complex semantic paraphrasing, ProSSD provides a robust detection boundary, proving its reliability as a secure defense mechanism in dynamic adversarial contexts.

E.2.2 Robustness to Informal and Noisy Text

As shown in Figure 10, beyond the mixed-noise setting already reported in the main text, the three single-noise experiments further demonstrate that ProSSD remains significantly more stable than the baselines under informal text and surface writ-

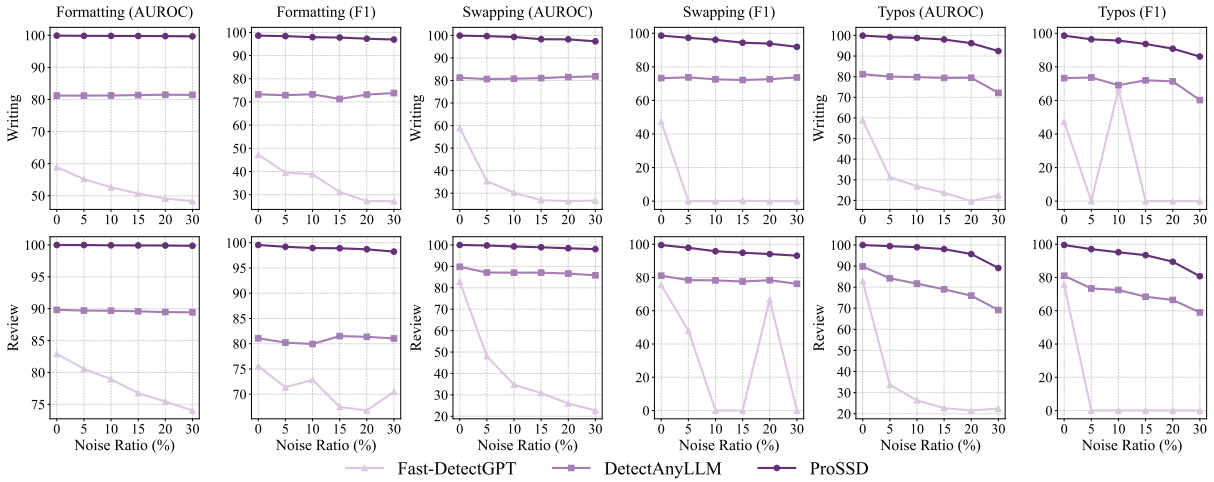


Figure 10: Robustness under three single-noise settings on Writing and Review. We report AUROC and F1 under *Formatting degradation*, *Word swapping*, and *Typos* as the noise ratio increases. Comparison methods include Fast-DetectGPT, DetectAnyLLM, and our ProSSD.

ing degradation. Overall, *Formatting degradation* exerts the smallest impact on all methods, *Word swapping* introduces a moderate level of perturbation, and *Typos* constitutes the most destructive setting among the three single-noise conditions. Even so, on both Writing and Review, the performance curves of ProSSD decline only gradually and remain consistently well above those of DetectAnyLLM and Fast-DetectGPT. In contrast, Fast-DetectGPT becomes unstable rapidly under word-order corruption and spelling errors, with its F1 dropping to nearly zero under multiple settings. DetectAnyLLM, although exhibiting relatively smaller fluctuations overall, consistently remains at a substantially lower absolute level, suggesting that under noisy conditions it merely preserves a low discriminative ceiling rather than achieving genuine robustness.

More specifically, *Formatting degradation* hardly weakens the detection capability of ProSSD. On Writing, its AUROC decreases only slightly from 99.90% to 99.67%, while its F1 declines from 98.60% to 96.91%. On Review, the degradation is even smaller, with AUROC dropping from 99.99% to 99.86% and F1 from 99.55% to 98.24%, thus remaining overall near saturation. This indicates that simple case perturbations, missing punctuation, or broken formatting boundaries do not substantially destroy the discriminative signals on which the model relies. In other words, ProSSD does not depend on strictly standardized written form, but is able to stably capture deeper generation-induced statistical features despite degradation in the surface presentation of text. ProSSD also exhibits strong tolerance to *Word swapping*, which more directly disrupts local expression order. Even at a perturbation ratio of 30%, its AUROC and F1 on Writing still reach 97.37% and 91.92%, respectively, while on Review they remain at 98.06% and 93.14%. These results suggest that as long as global semantic fragments and local structural patterns are at least partially preserved, the decision boundary of ProSSD does not collapse as rapidly as methods based on surface probability curvature.

By comparison, *Typos* are more challenging for all methods, because character-level spelling corruption simultaneously affects word forms, tokenization, and subsequent local structure identification, leading to the most pronounced performance degradation. Nevertheless, even at a 30% typo ratio, ProSSD still achieves an AUROC of 92.40% and an F1 of 86.13% on Writing; on Review, its AUROC and F1 remain at 89.09% and 80.78%, respectively, still substantially outperforming the two strong baselines. This indicates that although severe spelling errors increase the uncertainty of local structural analysis, ProSSD can still extract stable signals from the remaining recognizable semantic-structural evidence, demonstrating strong fault tolerance to noisy and non-canonical inputs. Taken together, the three single-noise experiments show that ProSSD adapts well to colloquial expressions, mild-to-moderate formatting disorder, and local spelling degradation, all of which are common in realistic application scenarios.

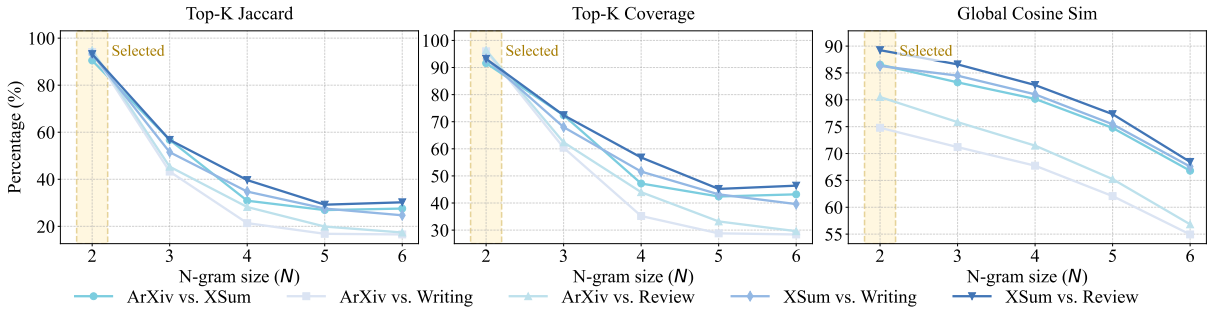


Figure 11: Cross-domain similarity of high-frequency POS n-gram meta-structures. We report Top-K Jaccard, Top-K Coverage, and Global Cosine Similarity across representative domain pairs as the n-gram size N increases. The shaded region indicates the selected meta-structure length adopted in ProSSD.

E.3 Sensitivity and Diagnostic Analyses

E.3.1 N-gram Overlap Analysis

To further explain why ProSSD maintains stable performance in cross-domain scenarios, we conduct a systematic analysis of local meta-structural similarity across four representative text domains: ArXiv, XSum, Writing Prompt, and Yelp Review. Specifically, we extract high-frequency POS n-gram meta-structures from each domain and gradually increase n from 2 to 6, so as to examine how the degree of cross-domain sharing evolves from short-range local patterns to longer structural fragments. We measure cross-domain similarity from three perspectives: (1) *Top-K Jaccard*, which characterizes the overlap between high-frequency meta-structure sets; (2) *Top-K Coverage*, which measures the proportion of high-frequency patterns covered by the shared meta-structures; and (3) *Global Cosine Similarity*, which reflects the closeness of the overall frequency distributions.

The results in Figure 11 shows that short-range local meta-structures are highly shared across domains. When $n = 2$, the average Top-K Jaccard, Coverage, and Global Cosine Similarity across all cross-domain pairs reach 93.18%, 94.08%, and 83.48%, respectively. When n increases to 6, these values decrease to 23.27%, 37.44%, and 62.91%. In other words, from 2-gram to 6-gram, the overlap at the discrete set level drops to roughly one quarter and the coverage to about two fifths, while the global distributional similarity still remains above 60%. This indicates that cross-domain differences are mainly reflected in long-range, compositional macro-structures.

This trend is highly consistent across domain pairs. For 2-grams, the Jaccard score exceeds 90% and the Coverage remains above 91% for all cross-domain pairs. For example, ArXiv vs. Yelp and ArXiv vs. Writing Prompt achieve 2-gram Jaccard scores of 94.88% and 94.12%, with corresponding Coverage values of 96.40% and 96.00%. Even for the stylistically more distinct pair ArXiv vs. XSum, the 2-gram Jaccard and Coverage still reach 90.51% and 91.60%. In contrast, when extending to 6-grams, the shared proportion of longer structures drops rapidly: the 6-gram Jaccard is only 16.55% for ArXiv vs. Writing Prompt and 17.37% for ArXiv vs. Yelp, confirming substantial domain-specific variation in long-span surface expression.

ProSSD does not rely on long-range, domain-specific macro-syntactic patterns, but instead models the joint statistics of short-range high-frequency meta-structures and semantic representations. The adopted meta-structure length, $N = 2$, preserves sufficient local structural constraints while avoiding the rapid sensitivity to domain style variation exhibited by longer n-grams. Therefore, although different domains vary substantially in sentence organization, rhetorical preference, and discourse style, they still share a stable statistical foundation at the local meta-structural level, which provides a key explanation for the strong generalization ability of ProSSD on unseen domains.

E.3.2 Sensitivity to Decoding Strategies

As shown in Figure 12, although varying the temperature changes the randomness of local token selection, it has little substantive impact on the overall discriminative capability of ProSSD. Across all four generators and all four temperature settings, the AUROC of ProSSD never falls below 99.33%, and its F1 never drops below 96.98%. In terms of fluctuation range, the AUROC variation remains within 0.56 percentage points for all four models, while the F1 variation stays within 1.23 percentage

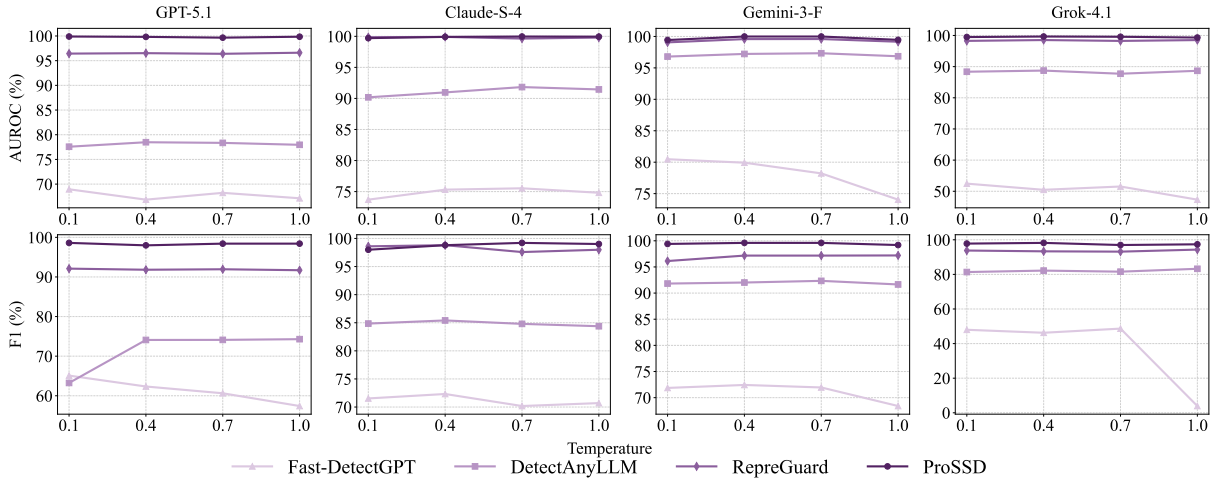


Figure 12: Detection performance under different decoding temperatures. We report AUROC and F1 across four generators as the temperature varies from 0.1 to 1.0. The compared methods are Fast-DetectGPT, DetectAnyLLM, RepreGuard, and ProSSD.

points, demonstrating strong stability. For example, on GPT-5.1, when the temperature increases from 0.1 to 1.0, the AUROC of ProSSD consistently remains within 99.66%–99.89%, and its F1 stays within 97.98%–98.60%. On Claude-Sonnet-4 and Gemini-3-Flash, this stability is even more evident, with the performance curves being nearly horizontal.

By comparison, the baseline methods are more sensitive to decoding temperature. DetectAnyLLM exhibits relatively smooth trends on Claude-Sonnet-4, Gemini-3-Flash, and Grok-4.1, but its absolute performance remains consistently below that of ProSSD, and its F1 fluctuates more noticeably on GPT-5.1, where the minimum drops to 63.24%. RepreGuard performs strongly overall and remains relatively stable in most settings, yet its results are still generally slightly inferior to those of ProSSD. Fast-DetectGPT is the most sensitive to temperature, especially on Grok-4.1, where its F1 drops sharply to 3.91% when the temperature reaches 1.0, indicating clear instability. Taken together, these results suggest that temperature mainly affects local decoding paths and token-level probability patterns, but does not disrupt the joint semantic-structural statistics modeled by ProSSD in the low-dimensional projected subspace. As a result, ProSSD is able to maintain stable and reliable detection performance across different decoding strategies.

E.4 Supplementary Ablation Analyses

To rigorously verify the effectiveness of the proposed framework and the contribution of each core component, we designed three groups of comparative experiments as shown in Table 11. First, in the robustness of semantic representations group, we aim to prove that this method does not rely on the semantic embeddings of a specific encoder, but rather is based on the inherent statistical differences between human and machine text. We replaced the default RoBERTa-large base with modern large language model embedding Qwen-3-0.6B-Embedding (Zhang et al., 2025) and randomly initialized embeddings. This setting verifies whether our supervised subspace learning can effectively extract discriminative features in different semantic spaces. Second, the impact of subspace projection group is specifically used to verify the core hypothesis in Section 3.2. We replaced supervised subspace learning with unsupervised PCA, random projection, and a baseline without projection. This comparison aims to prove that extracting label correlated variance is superior to purely maximizing global variance (such as PCA) or preserving random geometric structures. Finally, in the ablation on detection strategies, we explored the necessity of distribution modeling. We examined the results after removing Wasserstein weights to verify the hypothesis that different syntactic structures contribute unequally to detection; we also evaluated the one class setting that only constructs the human distribution (P_H) without utilizing the machine distribution (P_M), thereby testing the importance of

Table 11: Ablation studies across different domains. We report AUC and F1 scores for ArXiv and Writing, and F1 scores for XSum and Review. Results represent the mean and standard deviation over 5 independent runs. The statistical significance of the performance drop compared to ProSSD is measured by a t-test: * $p < 0.05$, † $p < 0.01$, ‡ $p < 0.001$.

Method	ArXiv		Writing		XSum	Review
	AUC	F1	AUC	F1	F1	F1
ProSSD (RoBERTa + SSP)	99.55±0.26	**99.25**±0.18	**99.34**±0.03	96.45±0.14	**96.96**±0.16	**97.23**±0.03
Robustness of Semantic Representations						
Qwen-2.5-0.6B-Embed	96.22±0.12‡	91.38±0.34‡	99.16±0.13*	**97.27**±0.25†	88.93±0.42‡	97.18±0.16
Random Projection	83.50±1.81‡	76.29±2.19‡	85.53±3.37†	77.21±3.58‡	87.36±1.36‡	84.05±2.63‡
Impact of Subspace Projection (SSP)						
w/o SSP (PCA)	99.51±0.22	98.74±0.20†	97.58±0.09‡	92.04±0.19‡	88.02±0.29‡	92.25±0.25‡
w/o SSP (Rand)	92.05±1.52‡	87.97±3.03†	97.84±0.41†	93.23±0.51‡	89.61±0.88‡	93.02±0.54‡
w/o SSP (No-Proj)	95.71±0.83‡	94.60±0.75‡	98.16±0.32†	94.44±0.43‡	90.81±0.28‡	94.17±0.30‡
Ablation on Detection Strategies						
w/o Wasserstein (Uniform)	**99.74**±0.20	98.07±1.06	96.42±0.88†	90.70±1.79†	91.46±1.56†	92.15±1.06‡
w/o Contrastive (Human-Only)	97.39±0.18‡	93.02±0.30‡	95.73±0.19‡	88.71±0.35‡	86.27±0.57‡	80.08±0.53‡

modeling both HWT and MGT distributions simultaneously.

Table 11 reports the quantitative results on four datasets. To ensure the reliability of statistical results, all experiments were independently repeated 5 times, and we report the mean and standard deviation. Significance tests (t test) confirmed that our method achieved statistically significantly better performance than the comparative ablation experiments on the vast majority of metrics ($p < 0.05$).

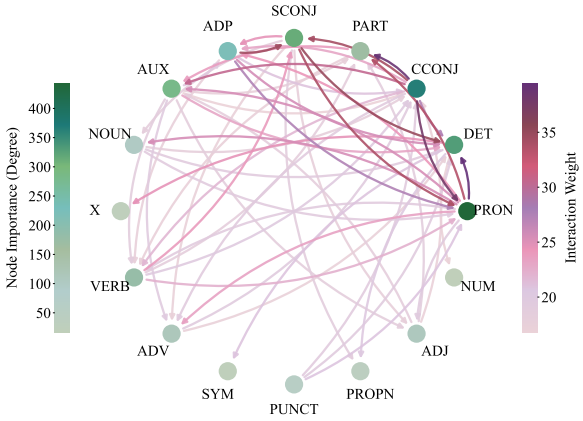
Effectiveness of supervised subspace learning. The experimental results strongly support our hypothesis in Section 3.2. As shown in the impact of subspace projection part, replacing supervised projection with PCA ("w/o SSP (PCA)") leads to a significant performance drop, especially on the XSum dataset where F1 drops from 96.96% to 88.02%. This indicates that PCA, which blindly maximizes global variance, is highly likely to retain the semantic noise shared by HWT and MGT, thereby drowning out weak style signals.

Importance of structured distribution modeling. The "w/o Wasserstein" ablation experiment exhibits a significant increase in standard deviation (for example ±1.79 on Writing) and a decrease in F1 score. This validates Theorem 1, that weighting local decisions according to theoretical discriminability (Wasserstein distance) can build a more robust detector. Furthermore, the "w/o Contrastive" (only constructing human distribution) ablation experiment performed the worst among all valid methods (for example Review F1 dropped to 80.08%). This confirms that MGT are not merely outliers of human text; they possess their own statistical regularities. Explicitly modeling the likelihood ratio $P(\mathbf{x}|\text{MGT})/P(\mathbf{x}|\text{HWT})$ is more effective.

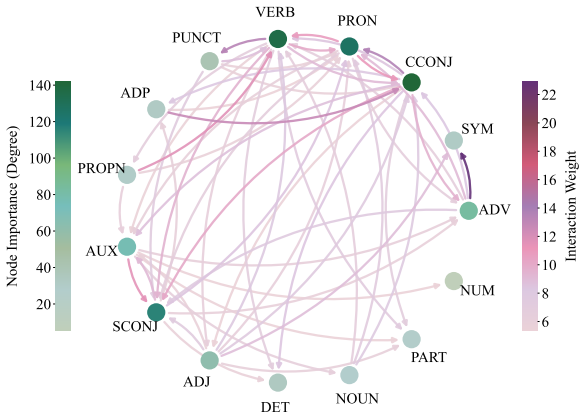
Robustness across embedding models. Although the original method uses RoBERTa, the comparative experiment with Qwen-3-0.6B-Embedding also achieved highly competitive results, even slightly outperforming RoBERTa on the Writing dataset (F1 97.27%). This proves that our method has high adaptability for different embedding models. Notably, the "random projection" comparative trial still achieved impressive performance (average F1 > 80%), even surpassing many existing mainstream detection baselines. This indicates that even in the absence of true semantic information, detection with considerable precision can be achieved solely relying on the structured statistical differences captured by our distribution modeling. When high quality semantic embeddings are introduced, performance is further improved by approximately 10% to 15%, illustrating that our method better combines deep semantic flow with syntactic statistical features.

E.5 Visualization Details

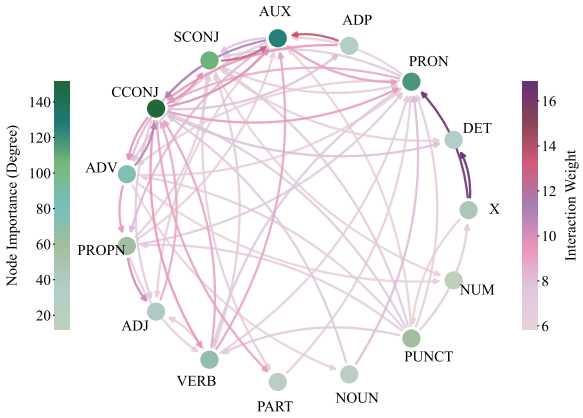
Visualization of discriminative meta-structures. To visualize the relationships between structures, we construct a discriminative meta-structure topology as shown in Figure 13. In this graph, each node represents a part of speech (POS) category acting as a local syntactic anchor, while directed edges explicitly characterize the meta-structure transition



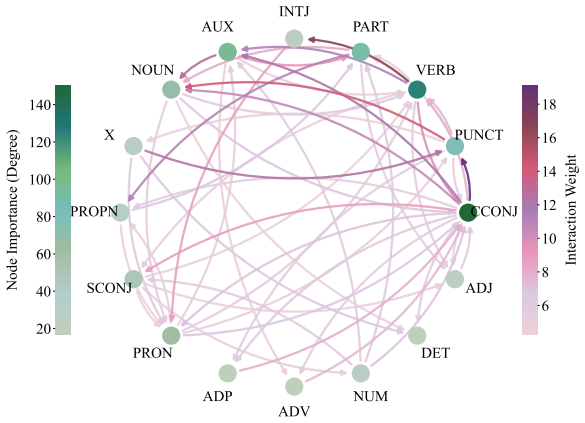
(a) Discriminative meta-structures in academic texts.



(b) Discriminative meta-structures in news summaries.



(c) Discriminative meta-structures in subjective reviews.

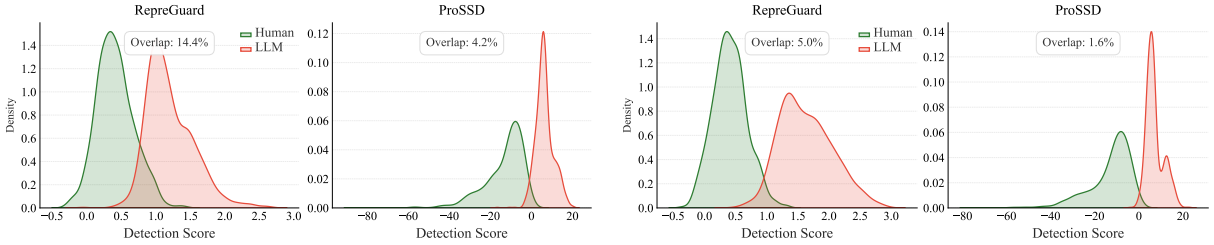


(d) Discriminative meta-structures in creative writing.

Figure 13: **Visualization of Discriminative meta-structure Topology across different domains.** Each node represents a POS tag, and edges denote structural transitions. The edge thickness and color intensity correspond to the Wasserstein discriminative weight (w_π), highlighting where MGT deviates most from HWT.

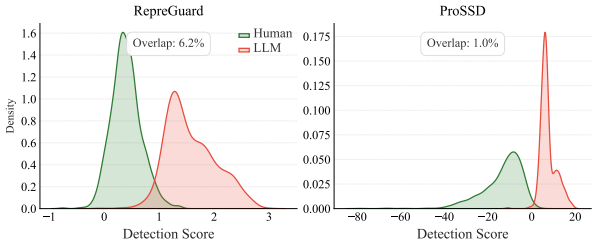
patterns $\pi_t = (pos_t, pos_{t+1})$ between adjacent words. The color intensity and thickness of the edges correspond to the Wasserstein discriminant weights (w_π) derived in Section 3.3. Darker colors and thicker lines indicate that under this syntactic transition path, the conditional semantic distribution of machine-generated text exhibits significant statistical deviation from that of human-written text, thus being assigned a higher discriminant weight. Accordingly, the node size reflects its cumulative usage, characterizing the importance of that syntactic structure as a hub for differentiated semantic flow.

The meta-structure topology across different domains reveals that the structural information generated by machines is not static but demonstrates strong context dependency. As shown in Figure 13, discriminative hotspots present distinctly different distribution results. In formal domains with high syntactic complexity such as news or Wikipedia (Figure 13b), the discriminative network presents a dense interaction structure centered on subordinating conjunctions (SCONJ) and coordinating conjunctions (CCONJ). This suggests that although large models perform well in local fluency, they still struggle to reproduce the rigorous logic and coherence of human authors in deep syntactic structures when dealing with long complex sentences and logical clauses, leading to significantly elevated Wasserstein distances at these connectives. Conversely, in domains with strong subjectivity or informal contexts like reviews or social media (Figure 13c), the center of discrimination significantly shifts towards pronouns (PRON), adjectives (ADJ), and even interjections (INTJ). In this context, statistical differences mainly stem from microscopic variations in stance expression and emotional coloring. Human-written text often exhibits extremely high variance and idiosyncrasy in subjective descriptions and first person narratives,

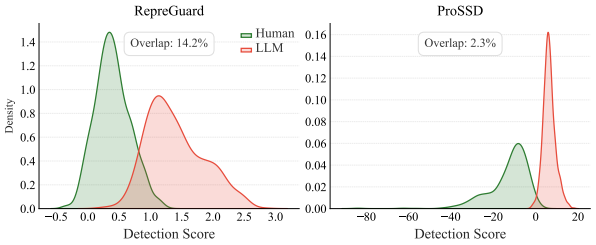


(a) Score distribution on GPT-5.1.

(b) Score distribution on Claude-4-Sonnet.



(c) Score distribution on Gemini-3-Flash.



(d) Score distribution on Grok-4.1.

Figure 14: **Visualization of detection score distributions across four LLMs.** The green and red curves represent the probability density functions of human-written and machine-generated texts, respectively. We compare our method, **ProSSD**, with the state-of-the-art baseline, RepreGuard.

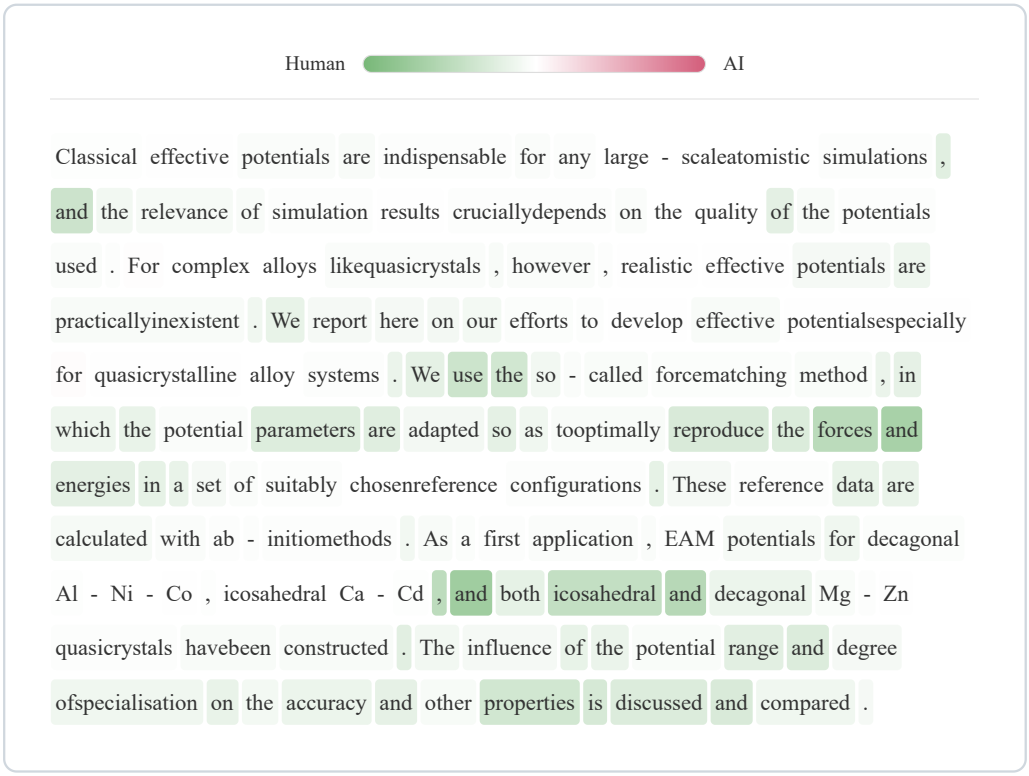
whereas model generated text tends to fall into semantic collapse, manifesting as more mediocre and conservative semantic choices that fail to mimic diverse authentic human emotional states.

Distribution of detection scores. To intuitively evaluate the ability of the detector to distinguish between human and machine text, we visualized the score distribution of all samples in the test set (Figure 14). The green and red curves represent the probability density distributions of scores for human-written text and large language model text, respectively. We calculated the overlap rate, defined as the intersection area of the two probability distribution functions, to quantify the degree of confusion. A lower overlap rate implies a greater distance between the two classes of text in the feature space and a clearer decision boundary.

Comparing the baseline RepreGuard with our ProSSD reveals the significant advantage of ProSSD in distinctiveness. First, ProSSD demonstrates extremely low distribution overlap. While RepreGuard shows an overlap rate exceeding 14% on multiple models (such as 14.4% on GPT 5.1), posing a higher risk of misjudgment, ProSSD compresses the overlap rate to an extremely low level (such as only 1.0% on Gemini 3 Flash), effectively partitioning the two types of text into distinct distribution regions. Second, ProSSD establishes a wider safety margin in score values. Observing the horizontal axis, the scores of RepreGuard are concentrated in a narrow interval from 0 to 3, whereas the score span of ProSSD significantly expands to between -80 and +20. This broad numerical difference builds a sufficient safety buffer between human and machine text, proving that this method can extract features with greater discriminative power and achieve more robust sample separation in the numerical space.

Word level discriminative analysis. We conducted a sample analysis to visualize the discriminative contribution of each token via color coding, as shown in Figures 15. As indicated by the topology in Figure 13, while attending to text content, our method keenly captures interactions between part of speech structures. The nodes for pronouns (PRON), coordinating conjunctions (CCONJ), adpositions (ADP), and determiners (DET), along with their connecting edges, exhibit the darkest colors, indicating that they are key to distinguishing between human and AI text.

As shown in Figure 15b, this text is classified as LLM generated. For instance, when handling clause introductions in long complex sentences (such as "whose") and parallel structures (such as "and"), the semantic transitions surrounding these tokens are extremely smooth and lack variation. Their distribution biases towards the center of our constructed AI Gaussian distribution, exhibiting



(a) Human-written text sample.

(b) Machine-generated text sample.

Figure 15: **Visualization of token-level discriminative contribution.** Red shading indicates tokens that the model identifies as more characteristic of LLM generation, while green shading indicates tokens more typical of human writing. The top pane displays a human-written sample, and the bottom pane displays an AI-generated sample.

Table 12: Performance comparison (%) on MIRAGE-DIG and MIRAGE-SIG. Tasks are categorized into **Gen.** (Generation) and **Rev.** (Revision, weighted average of Polish & Rewrite). **ProSSD** achieves superior performance in generation tasks and competitive results in revision tasks.

Method	MIRAGE-DIG								MIRAGE-SIG							
	Gen.				Rev.				Gen.				Rev.			
	AUC	Acc	MCC	TPR	AUC	Acc	MCC	TPR	AUC	Acc	MCC	TPR	AUC	Acc	MCC	TPR
Likelihood	49.36	50.91	1.83	1.47	44.90	50.00	0.00	1.80	49.68	52.07	1.96	1.45	44.55	50.01	0.15	1.70
LogRank	49.92	51.28	2.60	2.20	43.64	50.00	0.00	1.62	50.08	51.83	1.82	1.86	43.41	50.00	0.00	1.63
Entropy	65.22	61.50	25.43	10.99	56.78	54.94	14.55	10.75	64.42	61.23	15.92	10.74	57.52	55.45	7.23	10.76
RoBERTa-Base	55.23	53.97	14.34	12.50	49.42	50.30	1.94	5.16	53.68	53.92	5.29	11.01	49.26	50.68	1.37	5.36
RoBERTa-Large	47.16	52.17	8.42	8.71	53.77	52.72	6.10	7.68	47.03	52.36	4.17	9.10	53.70	52.96	3.46	7.33
LRR	52.15	53.41	7.77	7.01	40.03	50.00	0.00	1.94	52.14	53.11	3.14	6.57	40.25	50.00	0.00	2.05
NPR	61.20	61.40	26.04	1.91	48.85	52.83	8.60	2.71	60.88	61.70	15.71	1.85	49.01	52.39	4.72	2.33
DetectGPT	64.02	62.58	27.58	2.75	52.58	53.94	10.69	3.18	63.53	62.41	17.19	1.93	52.51	53.83	5.46	2.73
Fast-DetectGPT	77.68	72.34	46.28	43.10	55.83	54.99	11.50	11.04	77.06	71.93	20.78	42.00	56.00	55.55	5.65	11.65
ImBD	85.97	77.38	54.97	40.65	78.55	71.07	42.17	28.35	86.12	77.91	55.99	41.83	78.18	70.55	41.85	29.49
DetectAnyLLM	95.25	89.88	79.75	77.70	92.64	**87.18**	**74.66**	**77.67**	95.26	90.59	81.19	77.22	92.34	**86.90**	**73.99**	**76.73**
ProSSD (Ours)	**97.23**	**91.79**	**83.99**	**88.62**	**92.70**	84.84	69.89	71.16	**97.37**	**91.88**	**83.81**	**88.03**	**92.52**	84.99	70.06	69.94

extremely low semantic variance.

As shown in Figure 15a, this text is classified as human written. Consider the placement of "and", particularly the tight combination of CCONJ and DET in "and both". When describing complex chemical structures like icosahedral and decagonal quasicrystals, the human author employs the nested parallel structure "and both... and...". This usage exhibits strong specificity in the semantic vector space. Such representation lies closer to the human distribution μ_H and further from the AI distribution. Consequently, our method classifies it as human text.

E.6 Extended Evaluation on MIRAGE Benchmark

To further verify the generalization ability of the proposed method in complex real world scenarios, we extended the evaluation scope to the latest MIRAGE benchmark (Fu et al., 2025) proposed within three months. MIRAGE is a comprehensive evaluation framework covering 10 different corpora across 5 text domains, utilizing 17 advanced LLMs to construct diverse adversarial samples. The benchmark includes two settings: disjoint input generation (DIG) and shared input generation (SIG), aiming to test the performance of detectors under different input output correspondences. To optimize the spatial layout of Table 12, we categorized the evaluation tasks into "Gen." (generation) and "Rev." (revision). Specifically, the "Rev." category is the weighted average result of the "polish" and "rewrite" tasks in the original benchmark, as both tasks involve modifications to human-written text rather than generating content from scratch. The results of all baseline methods, including the previous state-of-the-art methods DetectAnyLLM and RepreGuard, are partly cited directly from the original MIRAGE paper to ensure fairness and consistency of comparison.

Table 12 presents the quantitative comparison results between our method and existing baseline methods. In the "Gen." task, ProSSD demonstrates significant performance advantages, outperforming the strongest baseline DetectAnyLLM in all metrics under both DIG and SIG settings. Notably, in the MIRAGE DIG generation subset, ProSSD achieved a TPR@5% of 88.62%, realizing a significant improvement of approximately 10.9 percentage points compared to DetectAnyLLM (77.70%). This indicates that our method constructs a more robust discriminative boundary for completely machine-generated content. In the "Rev." task, ProSSD remains highly competitive, achieving an AUROC score slightly higher than current state-of-the-art methods, for example 92.70% versus 92.64% under the DIG setting. Although DetectAnyLLM maintains a slight advantage in accuracy and MCC metrics for revision tasks, the excellent AUROC performance of ProSSD indicates that our method can effectively rank machine revised text, even if specific decision thresholds require further calibration. Overall, these experimental results confirm that our method can effectively capture critical semantic-structural distribution differences, a capability that is particularly prominent in generation type tasks.

F Algorithm and Experimental Settings

We formally describe our proposed ProSSD discrimination framework in Algorithm 1. The method first constructs a supervised subspace projection matrix. It then performs semantic-structural distribution modeling. After obtaining the distribution sets for HWT and MGT, it scores and detects new input text via a detection function based on modified Mahalanobis distance.

The hyperparameter settings of our method include the following aspects. Regarding part of speech tagging, we employ the latest `en_core_web_sm_3.8.0` model released by the spaCy library. This model covers over 100 part of speech tags. Regarding semantic embeddings, all experiments excluding ablation studies utilize the RoBERTa-large model. We use the 1024 dimensional vector from the final layer output as the word-level embedding. All experiments and calculations were completed on a CPU and a single NVIDIA A800 SXM4 80G GPU. For the hyperparameters of our method, we set the projection dimension $k = 4$ in both comparative and ablation experiments. The window size for constructing meta semantics and meta-structures is set to 2, with a stride of 1. Regarding training data volume, our comparisons utilize 1400 HWT and MGT samples extracted from the training set. For general experiments, the random seed is set to 42. In ablation studies, we conduct 5 independent runs with random seeds set to 42, 43, 44, 45, and 46.

In the sensitivity analysis, we primarily conducted two experiments. The first experiment discusses the optimal value of the projection dimension k. We tested k in the range of {1, 2, 3, 4, 5, 6, 8, 10, 12, 16, 32}. In this case, the semantic embedding model is controlled as RoBERTa-large, with 1400 HWT and MGT training samples. The second experiment discusses the required training data size. The data volume N ranges from {50, 100, 200, 400, 600, 800, 1000, 1200, 1400, 1600, 1800} HWT and MGT samples. In this experiment, the projection dimension is $k = 4$, and the semantic embedding model is RoBERTa-large.

Algorithm 1 Semantic-Structural Distribution Modeling via Subspace Projection

Require: Human corpus $\mathcal{D}_H$, AI corpus $\mathcal{D}_{AI}$.
Require: Embedding Model $\mathcal{F}$, Projection dim k, Window size w.
Ensure: Projection Matrix $\mathbf{P}$, Distribution Library $\mathbb{L}$.

procedure LEARNSUBSPACE($\mathcal{D}_H, \mathcal{D}_{AI}, k$)
  Extract embeddings $\mathbf{E}_H, \mathbf{E}_{AI}$ using $\mathcal{F}$.
  $\mathbf{E} \leftarrow [\mathbf{E}_H; \mathbf{E}_{AI}]$, $\mathbf{y} \leftarrow [0 \dots 0; 1 \dots 1]$
  Initialize $\mathbf{P} \leftarrow \emptyset$, $\mathbf{E}_{res} \leftarrow \text{Center}(\mathbf{E})$
  for $j = 1 \to k$ **do** ▷ Sec 3.2: Supervised Projection
    $\mathbf{p}_j^* \leftarrow \text{argmax}_{\mathbf{p}:\|\mathbf{p}\|=1} \left(\text{Cov}(\mathbf{E}_{res}\mathbf{p}, \mathbf{y})\right)^2$
    $\mathbf{E}_{res} \leftarrow \mathbf{E}_{res} - (\mathbf{E}_{res}\mathbf{p}_j^*)(\mathbf{p}_j^*)^T$ ▷ Sec 3.2: Feature Deflation
    Append $\mathbf{p}_j^*$ to $\mathbf{P}$
  end forreturn $\mathbf{P}$
end procedure

procedure BUILDDISTRIBUTIONLIBRARY($\mathcal{D}_H, \mathcal{D}_{AI}, \mathbf{P}$)
  $\mathbb{L} \leftarrow \emptyset$
  for $S \in \{\mathcal{D}_H, \mathcal{D}_{AI}\}$ **do**
    $\mathbf{V} \leftarrow \mathcal{F}(S)\mathbf{P}$ ▷ Sec 3.2: Project to k-dim semantic space
    Construct Meta-Semantics $\mathcal{X} = \{(\mathbf{x}_t, \pi_t)\}_{t=1}^T$
      where $\mathbf{x}_t = [\mathbf{v}_t; \mathbf{v}_{t+1}]$ and $\pi_t = (pos_t, pos_{t+1})$
  end for
  for each unique structure $\pi \in \Pi$ **do**
    Estimate $\mathcal{N}_H^\pi(\boldsymbol{\mu}_H, \boldsymbol{\Sigma}_H)$ and $\mathcal{N}_M^\pi(\boldsymbol{\mu}_M, \boldsymbol{\Sigma}_M)$ via MLE
    Calculate weight $w_\pi \leftarrow W_2(\mathcal{N}_H^\pi, \mathcal{N}_M^\pi)$ ▷ Sec 3.3: Wasserstein Dist
    $\mathbb{L}[\pi] \leftarrow \{\mathcal{N}_H^\pi, \mathcal{N}_M^\pi, w_\pi\}$
  end forreturn $\mathbb{L}$
end procedure

function DETECT(Text T, $\mathbf{P}, \mathbb{L}$)
  $\mathbf{V} \leftarrow \mathcal{F}(T)\mathbf{P}$
  Parse structure sequence $\pi_{1\dots M}$ and features $\mathbf{x}_{1\dots M}$
  $ScoreSum \leftarrow 0, \quad WeightSum \leftarrow 0$
  for $t = 1 \to M$ **do**
    if $\pi_t \in \mathbb{L}$ **then**
      Retrieve parameters $(\boldsymbol{\mu}_H, \boldsymbol{\Sigma}_H), (\boldsymbol{\mu}_M, \boldsymbol{\Sigma}_M), w_\pi$ from $\mathbb{L}$
      Calculate modified Mahalanobis distances $\mathcal{M}_H, \mathcal{M}_M$
      $s_t \leftarrow \frac{1}{2}(\mathcal{M}_H(\mathbf{x}_t) - \mathcal{M}_M(\mathbf{x}_t))$ ▷ Sec 3.4:Detection
      $ScoreSum \leftarrow ScoreSum + w_\pi \cdot s_t$
      $WeightSum \leftarrow WeightSum + w_\pi$
    end if
  end forreturn $\mathcal{S}(T) \leftarrow ScoreSum/WeightSum$
end function