DMHM: Density-aware Manifold Learning and Hybrid Mahalanobis Energy for LLMs-generated Text Detection

Tianle Liu[1], Zhiliang Tian[1]*, Zhen Huang[1]*, Tianlun Liu[1], Jingyuan Huang[1], Zhaoning Zhang[1], Chengcheng Shao[1], Dongsheng Li[1]

[1]College of Computer Science and Technology, National University of Defense Technology, Hunan, China

{liutianle, tianzhiliang, huangzhen, ltlun, jingyuanhuang, zhangzhaoning, shaoc, dsli}@nudt.edu.cn

Abstract

As the text generated by large language models (LLMs) increasingly resembles human-written text (HWT), detecting LLM-generated text (LGT) is crucial to avoid malicious use of LGT. Recent research treats LGT detection as an out-of-distribution (OOD) detection problem and views HWT as the OOD. However, existing OOD detection methods assume that LGT is a single homogeneous distribution. In practice, LGT exhibits different characteristics under different generation conditions. Text from weaker LLMs tends to form distinct clusters and is easy to detect, whereas text from stronger models significantly overlaps with HWTs and is hard to detect. To address the issue, in this paper, we propose an LGT detection framework based on density-aware manifold learning and the construction of hybrid Mahalanobis energy. We apply density-aware manifold learning with Laplacian smoothness and density regularization in embedding space, amplifying differences between LGT and HWT. We further propose a density-adaptive hybrid Mahalanobis metric that combines global and local covariance via density weighting, enabling adaptation to the manifold-aware embedding space. Finally, based on the metric, we define the distribution energy as a measure of distribution discrepancy, and we employ energy learning and contrastive learning to separate distributions hierarchically, establishing a clear OOD decision boundary. Experiments[1] show that our method outperforms strong baselines.

1 Introduction

Large language models (LLMs) like GPT4 (Achiam et al., 2023) have found wide applications in scientific research, news, and education (Liu and Lapata, 2019; Tian et al., 2025). However, their strong human-like capabilities create issues like academic fraud and rumors (Ahmed et al., 2021), detecting LLM-generated text (LGT) has become crucial. This task aims to detect whether an input text is LGT or human-written text (HWT). Current research mainly focuses on statistical–based or supervised learning-based methods to identify discriminative patterns between LGT and HWT, which are due to statistical and distributional differences arising from LLMs' training objectives and architectures.

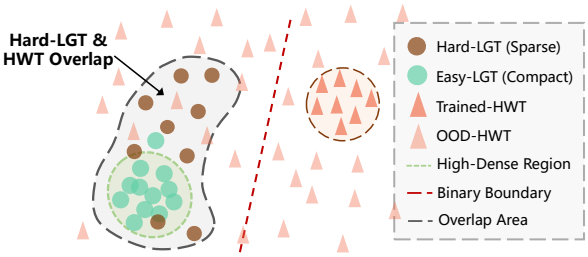

Figure 1: Comparison between LGT and HWT distributions. Unlike the assumption of a single compact distribution, LGT exhibits non-uniform density: Easy-LGT is compact and highly dense, while Hard-LGT is sparse and overlaps with HWT.

Existing LGT detection methods fall into two categories. **Statistical-based methods** detect texts by computing statistical metrics (Bao et al., 2024; Guo et al., 2024a). However, with the development of RLHF and adversarial training, LLMs can be optimized to mimic the statistical distribution of HWTs, weakening the effectiveness of statistical metrics for discrimination. In contrast, researchers begin exploring **supervised learning-based methods**, which train classifiers by using labeled data (Guo et al., 2024b; Chen et al., 2025). Despite them show strong performance on in-distribution data, the limited training data distribution results in poor generalization to unseen distributions.

Both of the above methods treat LGT detection as a binary classification and achieve great performance (Hans et al., 2024; Zhang et al., 2024). However, this paradigm implicitly treats the open

*Corresponding Authors

[1]Code is https://github.com/Letliy/OOD-LGT.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 3906–3929

and diverse HWT as a limited distribution during training, so the learned decision boundary only covers partial (limited) distributions, leading to poor generalization to unseen distributions. To address it, some research (Zeng et al., 2025) proposes a perspective: serving LGT detection as an out-of-distribution (OOD) detection problem and treating HWT as the OOD. HWT shows diverse styles and high degrees of freedom, resulting in a sparse distribution, which leads to its difficulty in modeling HWT as one distribution. If treats HWT as a class of binary classification, the model cannot cover the full diversity of HWT, tending to overfit to the observed HWT (Zeng et al., 2025). In contrast, constrained by factors (e.g., the training data and model architecture), LGT forms a more compact distribution (Tang et al., 2024). These properties make LGT suitable to serve as in-distribution samples, and serve the HWT as the OOD samples.

Existing OOD LGT detection methods assume LGT as a compact distribution (Zeng et al., 2025), overlooking the diversity induced by the generation mechanisms of LLMs. In practice, LGT exhibits composite distributions. LGTs from weaker LLMs exhibit clear statistical regularities (Wenger and Kenett, 2025), leading to reduced semantic diversity. These samples occupy a limited semantic region, resulting in dense regions, which are easier to detect (*easy-LGT*). In contrast, LGTs produced by stronger LLMs or modified by attacks exhibit higher diversity, pushing the samples away from dense LGT patterns and disrupting the statistical regularities of LGT. Hence, the diversity of stronger LLMs' LGT samples makes those samples fall into low-density regions that overlap with the HWT, making them harder to distinguish (*hard-LGT*). This phenomenon indicates that density is an effective characteristic for LGT detection: explicitly modeling density variations enables effective separation between dense easy-LGT and sparse regions shared by hard-LGT and HWT (as Fig. 1 and Fig. 6), motivating a density-aware OOD detection framework for robust LGT identification.

In this paper, we propose an LGT OOD detection framework based on density-aware manifold learning and the construction of hybrid Mahalanobis energy. Our method models the distribution clearly, reducing the misclassification of LGTs as HWTs. We apply manifold learning based on density and construct the density-adaptive Mahalanobis metric, which is used to establish energy margin between distributions, enabling adaptation to the non-uniform density. Specifically, (1) we apply density-aware manifold learning to characterize and amplify the distribution differences between LGT and HWT. By using the Laplacian smoothness and local density-aware regularization, we model the density manifold structures, which maintain LGTs in the compact region while pushing HWTs toward low-density areas. Furthermore, (2) we propose a density-adaptive hybrid Mahalanobis metric. By combining global and local covariances by density weighting, the metric quantifies the degree of deviation of samples in the manifold-aware embedding space more accurately. Finally, (3)based on the above metric, we define the distribution energy as a measure of distribution discrepancy, and we design energy learning and contrastive learning to achieve hierarchical distribution separation for LGT detection, establishing a clear OOD decision boundary for LGT detection. Our method achieves SOTA across metrics on multi-datasets, including attacked, unseen-LLM, and unseen-domain, demonstrating robustness and generalization.

Our contributions are as follows: (1) We propose a method for detecting LGTs based on density-aware manifold learning. We model the density-varying hierarchical structure of LGT, distinguishing compact LGT manifolds from sparse HWT distribution. (2) We propose a density-adaptive hybrid Mahalanobis metric by weighting global and local covariance by density to correct bias. (3) On the LGT detection task, our method achieves SOTA on the unseen-LLMs, -domain and attacked datasets, highlighting robustness and generalization.

2 Related Work

Statistical-based methods distinguish LGT using statistical metrics. *White-box* approaches compute metrics such as perplexity (Beresneva, 2016), likelihoods (Solaiman et al., 2019) and ranks (Gehrmann et al., 2019). In *black-box* settings, DNA-GPT (Yang et al., 2024) exposes output inconsistencies by comparing outputs from truncated inputs. DetectLLM (Su et al., 2023) utilizes log-rank perturbations. DetectGPT (Mitchell et al., 2023) measures perturbation-induced curvature changes. Binoculars (Hans et al., 2024) compares token probabilities between a target LLM and a reference model. Biscope (Guo et al., 2024a) examines a model's reliance on memorization of preceding tokens. Fast-DetectGPT (Bao et al., 2024) reduces overhead by approximating curvature with

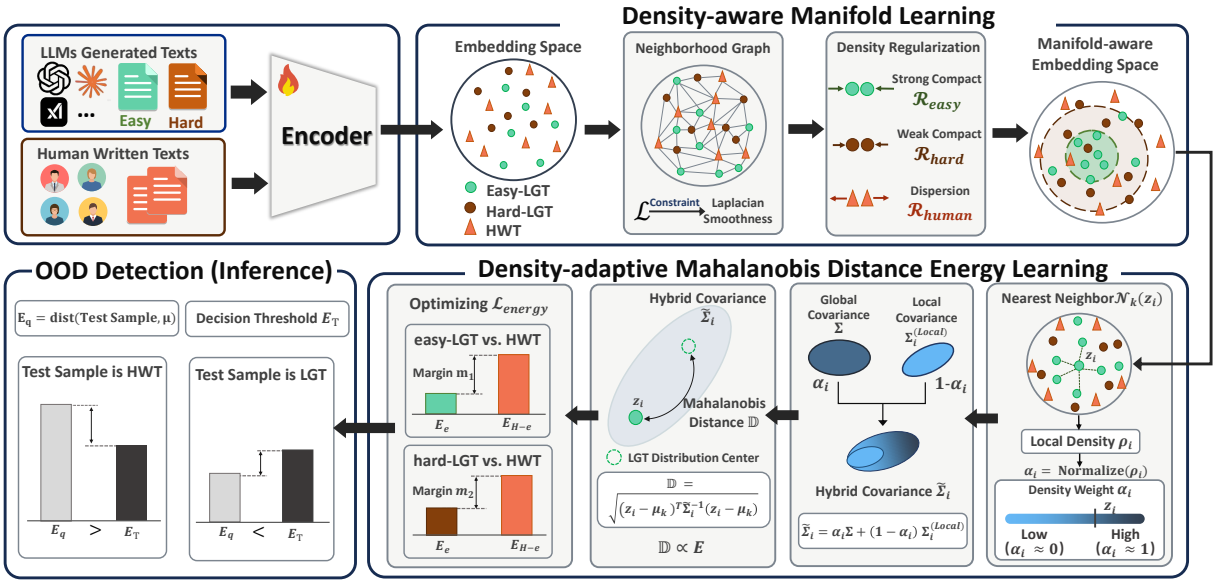

Figure 2: Overview of our method. During training, the encoder encodes texts into embeddings, and we apply density-aware manifold learning to construct the manifold-aware embedding space. We also apply Mahalanobis distance-based energy margin learning to separate LGT and HWT based on density. In inference, we compute the distance between the sample and the centers to detect whether it is an OOD sample (HWT) or not.

conditional probabilities.

Supervised learning-based methods train detectors on labeled HWT and LGT. RoBERTa-openai (Solaiman et al., 2019) fine-tunes RoBERTa for GPT classification. Soto et al. (2024) use style-based contrastive learning for detection. RADAR (Hu et al., 2023) employs adversarial training between a rewriter and a detector. OUTFOX (Koike et al., 2024) casts detection as a minimax game optimizing the generator and the discriminator. Ghostbuster (Verma et al., 2024) learns features from LGTs' multi-level label probabilities. EAGLE (Bhattacharjee et al., 2024) mitigates domain shift with contrastive and domain-adversarial learning. DeTeCtive (Guo et al., 2024b) extracts representations via multi-level contrastive objectives. MMD-DP (Zhang et al., 2024) reduces HWT–LGT distribution gaps using group MMD. ImBD (Chen et al., 2025) aligns detectors with LLM preference distributions. Zeng et al. (Zeng et al., 2025) detect by treating HWT as OOD and measuring distributional deviations.

3 Method

We propose an LGT OOD detection framework based on density-aware manifold learning and the construction of hybrid Mahalanobis energy. As illustrated in Fig. 2. (1) The density-aware manifold learning module (§ 3.1) builds a density manifold in the embedding space by combining the Laplacian smoothness with density-aware regularization. (2) The density-adaptive Mahalanobis metric module (§ 3.2) constructs a hybrid Mahalanobis distance to quantify the degree of deviation of samples. The metric can adapt to the non-uniform density. (3) The distribution separation module (§ 3.3) jointly applies energy learning and contrastive learning to enforce distribution separation, establishing a clear decision boundary for detection.

Given data samples as input, we map samples into the density-aware manifold (§ 3.1), then compute the Mahalanobis metric of samples (§ 3.2). Finally, we define the distribution energy via the above metric for energy learning and apply contrastive learning (§ 3.3).

3.1 Density-aware Manifold Learning

To characterize and amplify the distribution differences between LGT and HWT, we propose a manifold learning with local density, which maps samples into a manifold space with clearer density differences. Compared to HWT, LGT exhibits more regular and lower-noise statistical distributions, which show structured patterns in the embedding space. (Tang et al., 2024). Thus, LGT tends to concentrate in a compact and smooth low-dimensional space (i.e., a manifold) (Zhang and Dong, 2025), while HWT shows a more dispersed and diverse distribution. Therefore, instead of performing discrimination in the original embedding space, we

learn the manifold-aware embedding. We construct this embedding by encouraging nearby samples to remain close while smoothing local geometric relationships across the manifold. This enables discrimination to respect the real geometry of the data and avoid false density structure. The approach consists of three steps: (1) neighborhood weight matrix construction, (2) Laplacian-based manifold smoothing for neighborhood preservation, and (3) local density-aware regularization to the distribution structure. These steps aim to preserve the adjacency relationships while introducing density-aware constraints for different distributions.

Step 1: Neighborhood Weight Matrix Construction. To represent the manifold structure, we first construct a matrix that preserves the weight relationships among sample embeddings, providing the foundation for both neighborhood preservation and density regularization. Given a set of input texts $X = \{x_i\}_{i=1}^{N}$, we compute their embeddings $z_i = \phi(x_i)$. For each embedding z_i, we identify its r-nearest neighbors $\mathcal{N}_r(i)$ and construct a weighted neighborhood weight matrix $W \in \mathbb{R}^{N\times N}$, whose elements w_{ij} are defined as:

$$w_{ij} = \begin{cases} \exp\left(-\dfrac{\|z_i - z_j\|^2}{2\sigma^2}\right), & z_i \in N_k(z_j) \\ & \text{or } z_j \in N_k(z_i), \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

where σ is a temperature coefficient.

Step 2: Laplacian-based Manifold Smoothing. To accurately describe the structured distribution in manifold space, we use a Laplacian smoothing constraint (Belkin et al., 2006), which strengthens regularization in compact regions while avoiding excessive constraints in sparse areas. From the neighborhood weight matrix $W = [w_{ij}] \in \mathbb{R}^{N\times N}$, we define the unnormalized graph Laplacian as $\mathcal{L} = D - W$, where D is the degree matrix with $D_{ii} = \sum_j w_{ij}$, which represents the total connection strength between z_i and its neighboring samples, reflecting the local density. Following the local preservation principle in manifold learning, adjacent samples in the original space should remain within the same local neighborhood. Therefore, to constrain samples with high affinity (i.e., large weights w_{ij}) stay close, we apply the Laplacian smoothing (Eq. 2), which penalizes the differences between neighboring nodes to encourage large-weight w sample pairs remain closer, effectively smoothing the latent manifold structure. (Proof and analyses of Eq. 2 are shown in App. B.1 and B.2.)

$$L_{\text{manifold}} = \text{Tr}\left(X^\top \mathcal{L} X\right) = \tfrac{1}{2}\textstyle\sum_{i,j} w_{ij}\|z_i - z_j\|^2 \tag{2}$$

From Eq. 2, the magnitude of w_{ij} controls the strength of the manifold regularization, yielding larger weights w_{ij} and thus applies stronger penalties on the distances $\|z_i - z_j\|^2$, forcing samples within the local neighborhood to be more compact. Conversely, in sparse regions, the weights w_{ij} become smaller, which weakens the smoothing force.

Step 3: Local Density-aware Regularization. Manifold smoothing only ensures that neighboring samples stay close, but it cannot regularize density differences across distributions (i.e., LGT and HWT) on the manifold. To enhance differences across distributions, we introduce a local density-aware regularization, which encourages each distribution to preserve its density characteristics, enabling better discrimination among distributions. Since the sum of neighborhood weights of z_i (i.e., D_{ii}) represents the local density around z_i, we define the degree value of the z_i as its local density:

$$\text{dens}(z_i) = D_{ii} = \sum_j w_{ij}. \tag{3}$$

A larger $\text{dens}(z_i)$ indicates that the sample lies in a higher-density region; conversely, it is located in a sparse area. Therefore, we apply density-based regularization to encourage LGT to form high-density regions, and penalize density increase for HWT to keep HWT in sparse regions.

We divide the distributions into three categories: easy-LGT $\mathcal{M}_{\text{easy}}$, hard-LGT $\mathcal{M}_{\text{hard}}$, and HWT $\mathcal{M}_{\text{human}}$ (The definition of LLM-label and discussion are shown in App. A.4). We denote $|\mathcal{M}_k|$ as the number of samples in $\mathcal{M}_k$, where $k \in \{\text{easy}, \text{hard}, \text{human}\}$. Specifically, Easy-LGT shows stronger structured patterns and forms high-density clusters. To maintain this structure during manifold smoothing, we use the regularization $\mathcal{R}_{\text{easy}}$ to encourage higher local density for $z_i \in \mathcal{M}_{\text{easy}}$:

$$\mathcal{R}_{\text{easy}} = -\frac{1}{|\mathcal{M}_{\text{easy}}|}\sum_{i\in\mathcal{M}_{\text{easy}}} \log\big(\text{dens}(z_i)\big). \tag{4}$$

To encourage hard-LGT to stay in high-density regions, we enforce the compactness of $\mathcal{M}_{\text{hard}}$. However, since Hard-LGT is closer to HWT and exhibits a relatively sparse distribution, we adopt a smaller weight ρ_h to avoid over-regularization.

$$\mathcal{R}_{\text{hard}} = -\frac{\rho_h}{|\mathcal{M}_{\text{hard}}|}\sum_{i\in\mathcal{M}_{\text{hard}}} \log\big(\text{dens}(z_i)\big). \tag{5}$$

Compared with LGT, the HWT distribution is more scattered and should stay away from the high-density regions of LGT; thus, we apply a reverse regularization term $\mathcal{R}_{\text{human}}$ to restrain density increase and keep these samples in sparse regions:

$$\mathcal{R}_{\text{human}} = \frac{1}{|\mathcal{M}_{\text{human}}|} \sum_{i \in \mathcal{M}_{\text{human}}} \log\left(1 + \text{dens}(z_i)\right) \quad (6)$$

We combine the above regularization terms to obtain the final density-aware regularization loss:

$$L_{\text{dens}} = \mathcal{R}_{\text{easy}} + \mathcal{R}_{\text{hard}} + \mathcal{R}_{\text{human}}. \quad (7)$$

Finally, the manifold learning objective is:

$$L_{\text{con}} = L_{\text{manifold}} + L_{\text{dens}}. \quad (8)$$

This module maintains the manifold neighborhood structure while utilizing the density information of different distributions to form more discriminative density distributions.

3.2 Density-adaptive Mahalanobis Metric Construction

To accurately quantify the degree of deviation of samples in manifold-aware embedding space, we construct a hybrid Mahalanobis distance that can adapt to density variations. When computing the covariance matrix of the Mahalanobis distance, we incorporate local density information and build an adaptive hybrid covariance matrix. The traditional Mahalanobis distance uses the inverse covariance matrix Σ^{-1} to standardize features (Proof in App. B.3), which captures the true structure of the data distribution in an anisotropic embedding space. (Lee et al., 2018). However, the traditional Mahalanobis distance relies on a global covariance matrix for standardization. When using a single global covariance matrix for standardization on a non-uniform density manifold, the covariance estimation is dominated by high-density regions. As a result, the global covariance primarily reflects the geometry of dense regions, while the geometric characteristics of sparse regions are suppressed during standardization, leading to biased distance estimates. To address it, we propose a density-weighted hybrid Mahalanobis distance, which adapts to density for non-uniform manifolds.

Specifically, for sample embedding $z_i \in \mathcal{M}_k$, we search for its $k-$nearest neighbors $\mathcal{N}_k(z_i)$ and compute the local mean $\mu_i = \frac{1}{k}\sum_{j \in \mathcal{N}_k(z_i)} z_j$. The local covariance Σ_i^{local} based on this neighborhood:

$$\Sigma_i^{\text{local}} = \frac{1}{k-1} \sum_{j \in \mathcal{N}_k(z_i)} (z_j - \mu_i)(z_j - \mu_i)^T \quad (9)$$

In sparse regions, the local covariance reflects the local characteristics and captures the local geometric structure. In conclusion, an effective distance metric should balance global and local covariance; we thus adopt a density-based adaptive weighting to balance them. Specifically, we use KNN to define the local density ρ_i of each sample:

$$\rho_i = \frac{k}{\sum_{j \in \mathcal{N}_k(z_i)} \|z_j - z_i\|}. \quad (10)$$

If the neighbors around z_i are close, the sum of distances $\sum\|z_j - z_i\|$ is small, which leads to a large density ρ_i and indicates that the sample lies in a high-density region. Conversely, sparse neighbors lead to a small ρ_i, indicating that the sample lies in a low-density region. Then, we normalize the densities to obtain the density weight α_i:

$$\alpha_i = \frac{\rho_i - \rho_{\min}}{\rho_{\max} - \rho_{\min} + \epsilon}, \quad (11)$$

where $\rho_{\min} = \min(\rho_1, \ldots, \rho_{|\mathcal{M}_k|})$, $\rho_{\max} = \max(\rho_1, \ldots, \rho_{|\mathcal{M}_k|})$, and ϵ is a constant to avoid division by zero. When $\alpha_i \approx 1$, the sample is located in a high-density region, while $\alpha_i \approx 0$ is located in a low-density region. For each sample z_i, we define a hybrid covariance matrix as:

$$\widetilde{\Sigma}_i = \alpha_i \Sigma + (1 - \alpha_i)\Sigma_i^{\text{local}}, \quad (12)$$

where $\Sigma = \frac{1}{N-1}\sum_{i=1}^{N}(z_i - \mu)(z_i - \mu)^T$, N is the number of training samples. The hybrid covariance increases the weight of the global covariance in high-density regions to reduce the local noise. Instead, it assigns more weight to the local covariance in low-density regions to capture local geometry. Finally, we define the Mahalanobis distance as:

$$\mathbb{D}(z_i) = \sqrt{(z_i - \mu_k)^T \widetilde{\Sigma}_i^{-1}(z_i - \mu_k)}, \quad (13)$$

where $\mu_k = \frac{1}{|\mathcal{M}_k|}\sum_{j \in \mathcal{M}_k} z_j$. In the next phase, we use $\mathbb{D}$ to construct the distribution energy in energy learning (§ 3.3).

3.3 Distribution Separation Based on Energy and Contrastive Learning

To achieve a clear separation of different distributions, we utilize energy learning and contrastive

learning to optimize the distributional representation jointly, which forces the LGT to be more compact while pushing HWT toward boundary regions.

Energy-Based Margin Learning. To construct a hierarchical distribution structure, we construct energy margins between LGT and HWT based on hybrid Mahalanobis distance (§ 3.2). Traditional classification approaches struggle (e.g., cross-entropy) to capture the hierarchical relationships among different distributions. Therefore, we adopt energy learning to ensure that the distance between LGT and HWT is larger than a given threshold. Existing OOD detection studies (Liu et al., 2020; Ouyang et al., 2021) show that energy-based learning assigns low energy to in-distribution samples and high energy to OOD samples, thereby creating a separable energy gap that can serve as a scoring function for OOD detection tasks. Specifically, to quantify how much a sample deviates from the center, we define the energy by the hybrid Mahalanobis distance. First, we define the average distance of different distributions $\mathcal{M}_k$ from the center:

$$\bar{\mathcal{D}}(\mathcal{M}_k, \mu) = \frac{1}{|\mathcal{M}_k|} \sum_{z_i \in \mathcal{M}_k} \mathbb{D}(z_i, \mu). \tag{14}$$

We denote the center of the $\mathcal{M}_{easy}$ as $\mu_e = \frac{1}{|\mathcal{M}_{easy}|} \sum_{z_j \in \mathcal{M}_{easy}} z_j$, and similarly define μ_h for $\mathcal{M}_{hard}$. Then, for different distributions $\mathcal{M}_k$, we define the energy $E(z)$ as:

$$\begin{aligned} &E_{easy} = \bar{\mathcal{D}}(\mathcal{M}_{easy}, \mu_e),\ E_{hard} = \bar{\mathcal{D}}(\mathcal{M}_{hard}, \mu_h), \\ &E_{H\text{-}e} = \bar{\mathcal{D}}(\mathcal{M}_{human}, \mu_e),\ E_{H\text{-}h} = \bar{\mathcal{D}}(\mathcal{M}_{human}, \mu_h). \end{aligned} \tag{15}$$

Finally, we construct the energy loss (as Eq. 16). The loss pushes HWT–LGT pairs into the same center (e.g., $E_{H\text{-}e} - E_e$) when their energy margin is smaller than the threshold m. The Eq. 16 enforces HWT samples to exhibit high energy and LGT samples to retain low energy, establishing a clear OOD decision boundary.

$$\begin{aligned} \mathcal{L}_{\text{energy}} = &\ \text{softplus}[m_1 - (E_{H\text{-}e} - E_e)] \\ &+ \text{softplus}[m_2 - (E_{H\text{-}h} - E_h)], \end{aligned} \tag{16}$$

where m_1 and m_2 are margin thresholds (Analysis of m sensitivity in App. C.1).

Contrastive Learning for LGT-HWT Separation. To further separate the LGT and HWT distributions, we apply SimCLR-based (Guo et al., 2024b) contrastive learning loss (as Eq. 17), which pulls samples from the same distribution closer and pushes samples from different distributions apart. Specifically, for the sample $z_i \in \mathcal{M}_k$, we use the average embedding $\bar{z}_k = \frac{1}{|\mathcal{M}_k|} \sum_{z_k \in \mathcal{M}_k} z_k$ as the positive sample, which encourages intra-distribution compactness. The negative sample set $\mathcal{J}$ consists of all samples from the opposite distribution, improving instance-level discrimination.

$$\mathcal{L}_{\text{con}} = -\log \frac{\exp(z_i \cdot \bar{z}_k / \tau)}{\exp(z_i \cdot \bar{z}_k / \tau) + \sum_{j \in \mathcal{J}} \exp(z_i \cdot z_j / \tau)} \tag{17}$$

where τ is the temperature coefficient.

3.4 Training Objective and Inference

Training. We first apply density-aware manifold learning to optimize $L_{\text{d-manifold}}$ (§ 3.1), which maps samples into the manifold-aware embedding space. On this basis, the density-adaptive Mahalanobis metric module computes a distance (§ 3.2) to measure the distance of a sample from the center (i.e., μ_e and μ_h). We then use this distance as the energy to perform energy learning L_{energy} (§ 3.3). Finally, we combine a contrastive loss L_{con} (§ 3.3) to enhance instance-level discriminative ability.

Overall, our final training objective is (Eq. 18), which enables the model to utilize density information together with supervised signals. It aims to establish a clear OOD decision boundary between the LGT and HWT, achieving robust detection.

$$L_{\text{our-OOD}} = L_{\text{energy}} + L_{\text{con}} + L_{\text{d-manifold}}. \tag{18}$$

Inference. Given a query sample x with embedding z_q, we compute two distances between z_q and the LGT centers (μ_e and μ_h). The decision score is defined as the smaller of the two distances. If the distance score exceeds a given threshold, the sample is recognized as an OOD sample (i.e., HWT). Otherwise, the sample is an ID sample (LGT). Notably, the inference does not require access to the generator's identity (easy or hard). The identity label is used solely as a proxy for fine-grained modeling of the LGT distribution (the discussion of the label is shown in App. A.4).

4 Experiments

4.1 Experimental Settings

Datasets. *Deepfake* (Li et al., 2024) contains texts from 27 LLMs across 10 domains, comprising 332K training and 57K test samples. *M4* (Wang et al., 2024) includes 8 LLMs, 9 languages, and 6 domains, with 157K training and 42K testing

Type	Method	Deepfake			M4			Raid		
		AUROC↑	Acc↑	FPR95↓	AUROC↑	Acc↑	FPR95↓	AUROC↑	Acc↑	FPR95↓
Stat.	DetectLLM	0.5746	0.5898	0.9785	0.8351	0.7932	0.4803	0.7397	0.7818	0.8147
	DetectGPT	0.5512	0.5529	0.9322	0.6558	0.6409	0.8421	0.5513	0.5355	0.7314
	DNA-GPT	0.5534	0.5372	0.9992	0.6366	0.5693	0.7874	0.5847	0.5388	0.9955
	BiScope	0.7574	0.8142	0.8871	0.7836	0.8662	0.8537	0.7386	0.6725	0.9042
	Binoculars	0.5897	0.4958	0.7006	0.8384	0.7381	0.3172	0.7846	0.7581	0.4439
	Fast-Detect	0.7872	0.8146	0.7999	0.8064	0.8205	0.6525	0.7172	0.7646	0.5932
SL	RoBERTa	0.2288	0.5001	0.9971	0.7052	0.6608	0.9895	0.7549	0.7125	0.7995
	RADAR	0.5824	0.5091	0.9089	0.6394	0.7188	0.9478	0.7915	0.7556	0.6794
	MMD-MP	0.7927	0.7625	0.9752	0.9335	0.8745	0.4237	0.8238	0.7465	0.7125
	Detective	0.9479	0.9130	0.1756	0.9251	0.9016	0.4879	0.8856	0.8822	0.5833
	ImBD	0.3911	0.5158	0.9998	0.9298	0.8997	0.2685	0.7843	0.7435	0.8426
	HTAO	0.9714	0.9308	0.0909	0.9376	0.8619	0.2172	0.9633	0.9897	0.0953
	DMHM (Ours)	**0.9886**	**0.9887**	**0.0456**	**0.9666**	**0.9591**	**0.1446**	**0.9902**	**0.9927**	**0.0605**

Table 1: Main results. Our method is statistically significant via the t-test ($p < 0.01$), details are in App. C.6.

data. Its test data is attacked by OUTFOX (Koike et al., 2024) to increase difficulty. *RAID* (Dugan et al., 2024) contains texts from 11 models, 8 domains, and 11 attack forms, and we follow Zeng et al. to construct the Raid datasets. In analysis experiments, we use the unseen-LLMs datasets (GPT-4o, Claude-3.5, and Gemini-2.5), unseen-domains datasets (*Deepfake* unseen-domain setting), and attack datasets (Raid, Dipper, and Outfox). All datasets' details are shown in App. A.1.

Baselines. We compare DMHM with the mainstream method. *Statistical-based* (Stat.) baselines includes DetectLLM (Su et al., 2023), DetectGPT (Mitchell et al., 2023), DNA-GPT (Yang et al., 2024), Binoculars (Hans et al., 2024), DetectLLM (Su et al., 2023), BiScope (Guo et al., 2024a), and Fast-Detect (Bao et al., 2024). *Supervised learning-based* (SL) baselines include RoBERTa (Liu et al., 2019), RADAR (Hu et al., 2023), MMD-MP (Zhang et al., 2024), Detective (Guo et al., 2024b), ImBD (Chen et al., 2025), and HTAO (Zeng et al., 2025). See details in App. A.2.

Metric. Following mainstream research, we use AUROC, Accuracy (Acc), and FPR95 as evaluation metrics. AUROC measures the overall discriminative ability of the model and is robust to class imbalance. Acc reflects the proportion of correctly classified samples. The FPR95 is the false positive rate when the true positive rate reaches 95%.

Implementation details is provided in App. A.3 and **hyperparameter analyses** is shown in App. C.2. **Effect of batch-size** is shown in B.4.

4.2 Main Results

We conduct a comprehensive evaluation of mainstream LGT detection methods on three challenging benchmarks: *DeepFake*, *M4-multilingual*, and *Raid*. Our method achieves SOTA across all metrics in all benchmarks, which are shown in Tab. 1.

On DeepFake, our method achieves the SOTA performance across all metrics. Specifically, compared with the best statistical-based method (Fast-Detect), our method improves AUROC and Acc by 27.14% and 22.41%, while reducing FPR95 by 85.43%. Compared with the strongest supervised learning-based baseline (HTAO), our method improves AUROC and Acc by 1.72% and 5.79%, and reduces FPR95 by 4.53%. On M4-multilingual, compared with the second-best method, it improves AUROC and Acc by 2.90%, 5.75%, and reduces FPR95 by 7.26%, demonstrating its effectiveness under cross-lingual and outfox attack. Raid evaluates robustness under various attacks. While existing methods such as DetectGPT suffer from performance degradation, our method maintains strong performance and improves AUROC and Acc by 2.79% and 0.30% over the second-best baseline. These results highlight the robustness and generalization of our method. Notably, these results indicate that our method remains effective even when the overall metrics are already high (near-perfect). Evaluation of time and memory efficiency across baselines is provided in App. C.5.

4.3 Ablation Studies

We conduct ablation studies on each component (see Tab. 2). (1) Removing the density-aware manifold learning module L_{manifold} (§ 3.1) leads to AUROC drops on all datasets, showing its role in characterizing differences between LGT and HWT. (2) Removing the energy learning term L_{energy} (§ 3.3) degrades performance, indicating its importance in

Method	Deepfake	M4	Raid
w/o L_{manifold}	0.9477	0.9199	0.9615
w/o L_{energy}	0.9272	0.8370	0.9047
w/o L_{con}	0.9228	0.9275	0.9129
w/ Euclidean Distance	0.9238	0.8769	0.9513
w/o density weight α_i	0.9354	0.9209	0.9594
w/o Hard-Easy Modeling	0.9614	0.9301	0.9508
DMHM (Ours)	**0.9886**	**0.9666**	**0.9902**

Table 2: The ablation study of each component is evaluated by removing or replacing it. (metric: AUROC).

separating distributions to form an OOD decision boundary. (3) Removing the contrastive learning term L_{con} (§ 3.3) degrades performance, showing that it helps separate the LGT and HWT distribution. (4) Replacing the Mahalanobis distance (§ 3.2) with Euclidean distance results in a large drop in the M4 dataset, indicating the Mahalanobis adapts to the complex cross-language scenario. (5) Removing the adaptive weight α_i (relying on global covariance) lowers AUROC, confirming its necessity for complex distributions (§ 3.2). (6) Simplifying Hard–Easy LGT to single LGT modeling degrades performance on M4 and Raid datasets, proving the advantage of fine-grained modeling. The whole result is shown in Tab. 12.

Method	Unseen Advanced LLMs		
	GPT-4o	Claude-3.5	Gemini-2.5
Fast-Dectect	0.5414	0.5512	0.5295
Detective	0.8565	0.9425	0.7882
ImBD	0.8462	0.8544	0.7229
HTAO	0.9256	0.8892	0.7735
DMHM (Ours)	**0.9666**	**0.9971**	**0.8314**

Table 3: The results on unseen advanced LLMs generated texts detection (metric: AUROC).

4.4 Analyses of Generalization

Unseen-Advanced-LLMs LGTs Detection. Table 3 reports the results on detection of LGTs from unseen advanced LLMs, including *GPT-4o* (Hurst et al., 2024), *Claude-3.5-Haiku* (Anthropic, 2024), and *Gemini-2.5-Flash* (Comanici et al., 2025). On GPT-4o, our method achieves the highest AUROC (0.9666). On Claude-3.5, our method shows near-perfect performance (0.9971). Although Gemini-2.5 shows a challenging distribution, our method improves AUROC by 5.79%. The results show that even when the hard-LGT samples in the test set are unseen with respect to the training hard-LGT distribution, the proposed method still achieves strong performance, demonstrating its generalization. The whole result is shown in Tab. 9.

Unseen Domains LGTs Detection. Following the *deepfake* unseen domains setting, we compared our method with the strong supervised-based baselines (Tab. 10). Compared with the second-best method, our methods improve AUROC and Acc by 1.40%, 1.23%, and reduce FPR95 by 3.44%.

4.5 Analyses of Robustness under Attack

To evaluate robustness against adversarial attacks, we conducted experiments under three types of attacks: *Raid*, *DIPPER*, and *OUTFOX*, as shown in Tab. 4. The analysis of *Raid* is shown in Sec. 4.2. *DIPPER* applies paraphrasing to perturb text patterns (Krishna et al., 2023), while *OUTFOX* (Koike et al., 2024) utilize detector adversarial feedback to mimic HWT. Our method outperforms existing approaches across attacks, demonstrating robustness under attack. The relatively lower performance under *DIPPER* highlights paraphrasing as a major challenge. The whole results are shown in Tab. 11.

Method	Raid	Dipper-attack	Outfox-attack
Fast-Detect	0.6172	0.6077	0.7030
DeTeCtive	0.8856	0.9619	0.9818
ImBD	0.7843	0.8841	0.9495
HTAO	0.9633	0.9735	0.9853
DMHM (Ours)	**0.9902**	**0.9870**	**0.9951**

Table 4: The results under attacks (metric: AUROC).

4.6 Visualizing Embedding via UMAP

We apply UMAP (McInnes et al., 2018) to visualize the embedding during our method train, as shown in Fig. 3. Without training, easy-hard LGT and HWT embeddings are overlapping. (as Fig. 3a). After training, the LGT and HWT embedding distributions show a clear separation (as Fig. 3b).

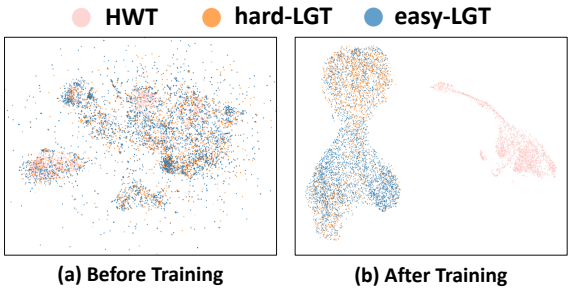

Figure 3: The visualization of the embeddings.

5 Conclusion

We propose DMHM, a density-aware manifold learning framework with hybrid Mahalanobis energy for LGT OOD detection. Our method models LGT distribution via density-aware manifold learning, compacting LGTs while pushing HWTs toward low-density regions. The density-adaptive hybrid Mahalanobis metric combines global and local covariance via density weighting to measure deviation. Moreover, energy and contrastive learning enhance distribution separation, which provides a clear decision boundary. Experiments show that DMHM achieves strong performance.

Limitations

In this work, we focus exclusively on text-based methods and do not explore multimodal approaches for LLM-generated text detection. Incorporating multimodal information (e.g., vision signal) may provide complementary cues and further enhance detection robustness. We leave the investigation of such multimodal techniques for future work.

Acknowledgements

This work was supported by the National Natural Science Foundation of China under Grant No.62376284 and No.62306330.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

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

Anthropic. 2024. Model card addendum: Claude 3.5 haiku and upgraded claude 3.5 sonnet. Technical report, Anthropic.

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*.

Maurice Stevenson Bartlett. 1937. Properties of sufficiency and statistical tests. *Proceedings of the Royal Society of London. Series A-Mathematical and Physical Sciences*, 160(901):268–282.

Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani. 2006. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of machine learning research*, 7(11).

Daria Beresneva. 2016. Computer-generated text detection using machine learning: A systematic review. In *International Conference on Applications of Natural Language to Information Systems*, pages 421–426. Springer.

Amrita Bhattacharjee, Raha Moraffah, Joshua Garland, and Huan Liu. 2024. Eagle: A domain generalization framework for ai-generated text detection. *arXiv preprint arXiv:2403.15690*.

Sid Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, and 1 others. 2022. Gpt-neox-20b: An open-source autoregressive language model. *arXiv preprint arXiv:2204.06745*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Jiaqi Chen, Xiaoye Zhu, Tianyang Liu, Ying Chen, Chen Xinhui, Yiwen Yuan, Chak Tou Leong, Zuchao Li, Long Tang, Lei Zhang, and 1 others. 2025. Imitate before detect: Aligning machine stylistic preference for machine-revised text detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 23559–23567.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, and 1 others. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, and 1 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.

Liam Dugan, Alyssa Hwang, Filip Trhlík, Andrew Zhu, Josh Magnus Ludan, Hainiu Xu, Daphne Ippolito, and Chris Callison-Burch. 2024. Raid: A shared benchmark for robust evaluation of machine-generated text detectors. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12463–12492.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. *arXiv preprint arXiv:1805.04833*.

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

Hanxi Guo, Siyuan Cheng, Xiaolong Jin, Zhuo Zhang, Kaiyuan Zhang, Guanhong Tao, Guangyu Shen, and Xiangyu Zhang. 2024a. Biscope: Ai-generated text detection by checking memorization of preceding tokens. *Advances in Neural Information Processing Systems*, 37:104065–104090.

Xun Guo, Yongxin He, Shan Zhang, Ting Zhang, Wanquan Feng, Haibin Huang, and Chongyang Ma. 2024b. Detective: Detecting ai-generated text via multi-level contrastive learning. *Advances in Neural Information Processing Systems*, 37:88320–88347.

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. *arXiv preprint arXiv:2401.12070*.

Xinlei He, Xinyue Shen, Zeyuan Chen, Michael Backes, and Yang Zhang. 2024. Mgtbench: Benchmarking machine-generated text detection. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, pages 2251–2265.

Xiaomeng Hu, Pin-Yu Chen, and Tsung-Yi Ho. 2023. Radar: Robust ai-text detection via adversarial learning. *Advances in neural information processing systems*, 36:15077–15095.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William W Cohen, and Xinghua Lu. 2019. Pubmedqa: A dataset for biomedical research question answering. *arXiv preprint arXiv:1909.06146*.

Ryuto Koike, Masahiro Kaneko, and Naoaki Okazaki. 2024. Outfox: Llm-generated essay detection through in-context learning with adversarially generated examples. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 21258–21266.

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. *Advances in Neural Information Processing Systems*, 36:27469–27500.

Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. 2018. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. *Advances in neural information processing systems*, 31.

Yafu Li, Qintong Li, Leyang Cui, Wei Bi, Zhilin Wang, Longyue Wang, Linyi Yang, Shuming Shi, and Yue Zhang. 2024. Mage: Machine-generated text detection in the wild. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 36–53.

Weitang Liu, Xiaoyun Wang, John Owens, and Yixuan Li. 2020. Energy-based out-of-distribution detection. *Advances in neural information processing systems*, 33:21464–21475.

Yang Liu and Mirella Lapata. 2019. Text summarisation with pretrained encoders. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3730–3740.

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. *arXiv preprint arXiv:1907.11692*.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Leland McInnes, John Healy, and James Melville. 2018. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*.

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*, pages 24950–24962. PMLR.

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng-Xin Yong, Hailey Schoelkopf, and 1 others. 2022. Crosslingual generalization through multitask finetuning. *arXiv preprint arXiv:2211.01786*.

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. *arXiv preprint arXiv:1808.08745*.

Yawen Ouyang, Jiasheng Ye, Yu Chen, Xinyu Dai, Shujian Huang, and Jiajun Chen. 2021. Energy-based unknown intent detection with data manipulation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2852–2861.

Vinu Sankar Sadasivan, Aounon Kumar, Sriram Balasubramanian, Wenxiao Wang, and Soheil Feizi. 2023. Can ai-generated text be reliably detected? *arXiv preprint arXiv:2303.11156*.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, and 1 others. 2021. Multitask prompted training enables zero-shot task generalization. *arXiv preprint arXiv:2110.08207*.

Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, Gretchen Krueger, Jong Wook Kim, Sarah Kreps, and 1 others. 2019. Release strategies and the social impacts of language models. *arXiv preprint arXiv:1908.09203*.

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

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

Ruixiang Tang, Yu-Neng Chuang, and Xia Hu. 2024. The science of detecting llm-generated text. *Communications of the ACM*, 67(4):50–59.

Zhiliang Tian, Jingyuan Huang, Zejiang He, Zhen Huang, Menglong Lu, Linbo Qiao, Songzhu Mei, Yijie Wang, and Dongsheng Li. 2025. Llm-based rumor detection via influence guided sample selection and game-based perspective analysis. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 28402–28414.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, and 1 others. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Christoforos Vasilatos, Manaar Alam, Talal Rahwan, Yasir Zaki, and Michail Maniatakos. 2023. Howkgpt: Investigating the detection of chatgpt-generated university student homework through context-aware perplexity analysis. *arXiv preprint arXiv:2305.18226*.

Vivek Verma, Eve Fleisig, Nicholas Tomlin, and Dan Klein. 2024. Ghostbuster: Detecting text ghostwritten by large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1702–1717.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Toru Sasaki, and 1 others. 2024. M4: Multi-generator, multi-domain, and multi-lingual black-box machine-generated text detection. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1369–1407.

Emily Wenger and Yoed Kenett. 2025. We're different, we're the same: Creative homogeneity across llms. *arXiv preprint arXiv:2501.19361*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and 1 others. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

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*.

Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, and 1 others. 2022. Glm-130b: An open bilingual pre-trained model. *arXiv preprint arXiv:2210.02414*.

Cong Zeng, Shengkun Tang, Yuanzhou Chen, Zhiqiang Shen, Wenchao Yu, Xujiang Zhao, Haifeng Chen, Wei Cheng, and zhiqiang xu. 2025. Human texts are outliers: Detecting LLM-generated texts via out-of-distribution detection. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Shuhai Zhang, Yiliao Song, Jiahao Yang, Yuanqing Li, Bo Han, and Mingkui Tan. 2024. Detecting machine-generated texts by multi-population aware optimization for maximum mean discrepancy. In *International Conference on Learning Representations (ICLR)*.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, and 1 others. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

Yukun Zhang and Qi Dong. 2025. Dynamic manifold evolution theory: Modeling and stability analysis of latent representations in large language models. *arXiv preprint arXiv:2505.20340*.

A Experimental Setting Details

A.1 Datasets Details.

Main Datasets. The specific types of LLMs included in the Deepfake dataset are summarized in Tab. 5, and the corresponding writing tasks covered are detailed in Tab. 6. Tab. 7 outlines the characteristics of the M4-multilingual dataset. The details of the Raid dataset are presented in Tab. 8.

Unseen-LLMs Datasets. The *GPT-4o* (Hurst et al., 2024), *Claude-3.5-Haiku* (Anthropic, 2024), and *Gemini-2.5-Flash* (Comanici et al., 2025) datasets cover three domains: news (*XSum*) (Narayan et al., 2018), story writing (*WritingPrompts*) (Fan et al., 2018), technical QA (*PubMed*) (Jin et al., 2019), business reviews & user data (*Yelp*) [2]. For each domain, 150 LLM-generated samples were created by GPT-4o, Claude-3.5-Haiku, and Gemini-2.5-Flash in conversational settings, where models were instructed to emulate professional news, fiction, and technical writing styles, ensuring diverse and naturalistic outputs.

Unseen-Domain Datasets. For the *deepfake* unseen-domain setting, the objective is to evaluate the classifier's ability to detect texts from domains that were not present during the training phase. Specifically, data from a particular domain is excluded from the training set, and the classifier is trained on the remaining texts from other domains. The model is then tested on the excluded domain to assess its performance in an out-of-distribution scenario. This process is repeated using a cross-validation approach across 10 different testbeds to report the weighted average performance, thereby rigorously testing the detector's robustness against novel writing tasks.

Unseen-Attacked Datasets. The dataset comprises texts generated by ChatGPT and GPT-3.5 under three scenarios: *No-attack*, *DIPPER-attack*, and *OUTFOX-attack*. The DIPPER attack employs the 11B document-level paraphrasing model proposed by (Krishna et al., 2023), which rewrites LLM outputs with controlled lexical diversity and content reordering. The OUTFOX attack generates adversarial essays through iterative in-context learning. A detector trained on triplets of human and LLM essays provides feedback used as in-context exemplars, prompting ChatGPT to produce more human-like and detector-evasive texts.

A.2 Baseline Details

All baselines are evaluated using official implementation settings, which can get the best performance.

DetectLLM (Su et al., 2023) DetectLLM leverages the sensitivity of machine-generated text to minor modifications in the input. It specifically utilizes log-rank perturbations to measure how the text's probability fluctuates, identifying LGT based on its distinct robustness characteristics compared to human writing.

DetectGPT (Mitchell et al., 2023) DetectGPT relies on the hypothesis that machine-generated text lies in regions of negative curvature on the model's log-probability function. It detects LGT by generating slight perturbations of the candidate text and measuring the resulting drop in probability, identifying peaks distinct to model outputs.

DNA-GPT (Yang et al., 2024) DNA-GPT operates on the premise that LLMs generate the remaining parts of a text differently depending on whether the prefix is truncated. It exposes these generation inconsistencies by comparing the outputs derived from original versus truncated inputs to distinguish machine text from human writing.

Biscope (Guo et al., 2024a) Biscope focuses on the distinct structural dependencies found in machine-generated text versus human writing. It detects LGT by examining the model's reliance on memorizing preceding tokens, highlighting differences in contextual consistency and redundancy.

Binoculars (Hans et al., 2024) Binoculars functions without training data by using a pair of pre-trained language models to assess text. It computes a perplexity gap or anomaly score by comparing the token probabilities assigned by a target scoring LLM against a smaller, more regularized reference model.

Fast-DetectGPT (Fast-Detect) (Bao et al., 2024) Fast-DetectGPT improves upon the efficiency of DetectGPT by avoiding the costly generation of perturbed samples. Instead, it approximates the local curvature of the probability landscape using conditional probability assessments, significantly reducing computational overhead.

RoBERTa-openai (Solaiman et al., 2019) This baseline involves fine-tuning a RoBERTa model specifically for the task of GPT output classification. It acts as a supervised binary classifier, trained on a dataset of web-scraped human text and neural-generated text to distinguish between the two sources.

RADAR (Hu et al., 2023) RADAR utilizes a holistic adversarial framework involving a paraphraser and a detector. The paraphraser attempts to rewrite machine text to evade detection, while the

[2] https://business.yelp.com/data/resources/open-dataset

detector is trained to identify these rewritten texts, creating a robust defense against paraphrasing attacks.

MMD-DP (Zhang et al., 2024) MMD-DP aims to improve the generalization of detectors by aligning the feature distributions of human and machine text. It employs Maximum Mean Discrepancy (MMD) to reduce the distributional gaps between Human-Written Text (HWT) and Large Language Model-Generated Text (LGT).

DeTeCtive (Guo et al., 2024b) DeTeCtive focuses on learning fine-grained nuances between human and machine text representations. It extracts robust text features using a multi-level contrastive objective that captures differences at both the token and sequence levels.

ImBD (Chen et al., 2025) ImBD introduces an Imitation-Based Detection framework that aligns the detector's behavior with the inherent preference distributions of LLMs. By mimicking the generation process, it enhances the ability to identify text that aligns too perfectly with model likelihoods.

HTAO (Zeng et al., 2025) This method approaches detection from an anomaly detection perspective, treating Human-Written Text (HWT) as Out-Of-Distribution (OOD) data. It detects machine text by measuring distributional deviations, effectively identifying human text as an anomaly within the model's generated distribution.

A.3 Implementation Details and Analyses.

Experiments are conducted using Python 3.10 and PyTorch 2.8.0, with pretrained weights from the HuggingFace Transformers library (Wolf et al., 2019). All baselines are evaluated using official implementation settings under the same experimental environment. We adopt Unsup-SimCSE-RoBERTa$_{\text{base}}$ as the encoder and optimize it with the AdamW optimizer (Loshchilov and Hutter, 2017) under a cosine annealing learning rate schedule. The maximum learning rate of *deepfake* and *Raid* is set to 2×10^{-5} and the M4 is 5×10^{-6}, and the weight decay is 1×10^{-4}. The $\rho_{\text{h}} = 0.5$, $m_1 = 2$,and $m_2 = 1.5$. We set k and r to 10. The maximum input sequence length is 512 tokens. Training is performed for 20 epochs on a single NVIDIA A800 GPU with a batch size of 128. For *Deepfake, M4, and Raid*, we use joint estimation of covariance and neighborhoods across-LLMs, across-languages, and across-domains. The distribution means μ_e and μ_h, as well as the global covariance matrix, are initialized prior to training using the entire training dataset (without validation and test datasets).

Covariances Computational Feasibility and Stability. For each sample z_i, we define a hybrid covariance matrix as

$$\widetilde{\Sigma}_i = \alpha_i \Sigma + (1 - \alpha_i)\Sigma_i^{\text{local}}, \tag{19}$$

where $\Sigma = \frac{1}{N-1}\sum_{i=1}^{N}(z_i - \mu)(z_i - \mu)^T$ denotes the global covariance. Although $\widetilde{\Sigma}_i$ is defined at the sample level, our implementation does not require explicitly inverting a distinct covariance matrix for each sample. The local covariance Σ_i^{local} is used only to capture neighborhood-level second-order statistics, while all matrix inversions are performed on a single batch-level covariance constructed from these statistics. Consequently, the expensive $O(d^3)$ matrix inversion is executed once per batch and is independent of the batch size. Increasing the batch size only improves the estimation of the global and local statistics, but does not introduce additional inversions or change the asymptotic computational complexity.

To prevent rank deficiency and numerical instability inherent to $k \ll d$, we mix with a full-rank global covariance and apply diagonal regularization $\Sigma_{\text{mix}} \leftarrow \Sigma_{\text{mix}} + \epsilon I$. When the batch size is too small to reliably estimate local structure, the method safely falls back to the global covariance. As a result, the computational overhead is amortized at the batch level, making the method stable and scalable even for large batch sizes and high-dimensional embedding spaces.

kNN Neighborhoods Setting. All kNN graphs, Laplacian weights, local density estimates, and covariance matrices are computed independently within each mini-batch. For a batch of size N, pairwise similarities are used to identify k nearest neighbors and derive graph- and density-based terms. Local covariance matrices are estimated from batch-level kNN neighborhoods and combined with a global covariance via a density-adaptive coefficient. These quantities are recomputed at every iteration without caching, avoiding the construction of a global kNN graph and enabling scalability to large datasets. With $N \leq 512$ and $k \leq 10$, the overhead is negligible.

More implementation setting analyses. The influence and theoretical proof of a moderate batch size is sufficient to capture stable manifold geometry,

which is discussed in the App. B.2. The result of the influence of batch size is shown in Tab.13.

A.4 Analyses and Criterion for Easy-Hard LGT Sample Selection

Clarification on Label Usage and Practical Deployability. Notably, the inference does not require access to the generator's identity (easy or hard). The identity label is used solely as a proxy for fine-grained modeling of the LGT distribution. They shape the global geometry of the representation space during optimization, but do not constitute oracle-assisted detection or label leakage at inference.

In our framework, generator identity is not treated as a semantic label to be predicted, but as an instrumental variable for approximating a latent detectability factor that cannot be directly observed. Prior studies (Sadasivan et al., 2023; He et al., 2024) have shown that detection difficulty exhibits consistent, model-level regularities correlated with capacity, alignment, and scale. Leveraging generator identity, therefore, provides a practical and reproducible means to inject a structural prior over detection difficulty, rather than supervision over source attribution. Crucially, the learned detector is never trained to distinguish between generators, nor is generator identity available at inference time. Instead, the Easy/Hard stratification constrains how generated texts are modeled in terms of their expected detectability regimes. As a result, the detector learns representations that are sensitive to difficulty-related distributional properties, rather than generator-specific artifacts. Moreover, we explicitly decouple difficulty from generator identity by labeling all adversarially perturbed samples as Hard-LGT regardless of their source. This reflects our assumption that detection difficulty arises from the interaction between model capability, prompting, and decoding strategies, rather than being an intrinsic property of any fixed generator.

Consequently, the proposed framework generalizes to unseen or future LLMs without requiring prior knowledge of their identities. As long as the generated outputs fall within similar detectability regimes, the learned difficulty-aware modeling remains effective. Importantly, generator-aware labeling functions purely as a training-time mechanism for structural regularization: external annotations are used only to impose weak, distribution-level geometric biases during training, analogous to curriculum learning or class-conditional regularization. This design avoids label leakage and places no assumptions on the test-time data distribution, allowing the final detector to be fully deployable without oracle access.

Criterion Selection. To construct a robust evaluation benchmark, we categorize the source LLMs into Hard-LGT (Hard-to-detect LGT) and Easy-LGT based on a composite criterion of empirical detectability, model capability, and architectural sophistication. First, drawing upon recent benchmarking studies (Wang et al., 2024; Dugan et al., 2024; Bao et al., 2024), we observe a consistent inverse correlation between detection performance (e.g., AUROC) and the generator's scale and alignment quality. Specifically, detectors exhibit significant performance degradation when identifying texts from models undergoing extensive RLHF or possessing massive parameter scales (Sadasivan et al., 2023). Second, to ensure our selection reflects the current state-of-the-art, we incorporate several major Leaderboards [3] [4] as a dynamic reference for model "human-likeness." Models ranking high on the leaderboard are proven to generate texts that are statistically closer to human distributions, thereby posing greater challenges for detection.

We define text generated by LLMs with more than 30B parameters together with attacked text as hard-LGT. Prior studies (Wang et al., 2024; Dugan et al., 2024) show that text generated by LLMs with more than 30B parameters (e.g., Cohere and GPT4) achieves relatively low detection accuracy (<60%) against mainstream detectors, which supports that this type of text is harder to detect. Guided by these principles, we designate the source LLMs for Hard-LGT across three datasets as follows: (1) In the *Deepfake* dataset, we select high-capacity models,s including OpenAI-GPT (text-davinci-003, text-davinci-002), LLaMA (65b, 30b), OPT (30b) and GLM-130B, as they have demonstrated superior evasion capabilities in zero-shot detection tasks. (2) For the *M4* dataset, we isolate ChatGPT, GPT-4, and Cohere as Hard-LGT sources due to their advanced instruction-following alignments. (3) In the *Raid* dataset, ChatGPT, GPT-4, Cohere, Mistral, and MPT are categorized as Hard-LGT.Conversely, texts from smaller or earlier-generation models (e.g., GPT-2) are classified as Easy-LGT. Furthermore, to evaluate robustness under worst-case scenarios, all text samples subjected to adversarial

[3] https://lmarena.ai/zh/leaderboard

[4] https://huggingface.co/open-llm-leaderboard

LLMs Set	LLMs
OpenAI GPT (Brown et al., 2020)	GPT-3.5-Turbo, Text-DaVinci-002, Text-DaVinci-003
Meta LLaMA (Touvron et al., 2023)	LLaMA-13B, LLaMA-30B, LLaMA-65B, LLaMA-7B
Facebook OPT (Zhang et al., 2022)	OPT-125M, OPT-350M, OPT-1.3B, OPT-IML-Max-1.3B, OPT-2.7B, OPT-6.7B, OPT-13B, OPT-30B, OPT-IML-30B
GLM-130B (Zeng et al., 2022)	GLM-130B
Google FLAN-T5 (Chung et al., 2024)	FLAN-T5-Small, FLAN-T5-Base, FLAN-T5-Large, FLAN-T5-XL, FLAN-T5-XXL
BigScience	BLOOM-7B (Muennighoff et al., 2022), T0-3B(Sanh et al., 2021), T0-11B
EleutherAI	GPT-J(Vasilatos et al., 2023), GPT-NeoX(Black et al., 2022)

Table 5: The type of LLMs contained in the **Deepfake**

Dataset	CMV	Yelp	XSum	TLDR	ELI5
Train	4,461/21,130	32,321/21,048	4,729/26,372	2,832/20,490	17,529/26,272
Valid	2,549/2,616	2,700/2,630	3,298/3,297	2,540/2,520	3,300/3,283
Test	2,431/2,531	2,685/2,557	3,288/3,261	2,536/2,451	3,193/3,215
WP	ROC	HellaSwag	SQuAD	SciGen	all
6,768/26,339	3,287/26,289	3,129/25,584	15,905/21,489	4,644/21,541	95,596/236,554
3,296/3,288	3,286/3,288	3,291/3,190	2,536/2,690	2,671/2,670	29,467/29,462
3,243/3,192	3,275/3,207	3,292/3,078	2,509/2,535	2,563/2,338	29,015/28,365

Table 6: The number of instances of the source in the Deepfake. The left side of the "/" is HWT, and the right side is LGT.

Split	Language	DaVinci-003	ChatGPT	LLaMA 2	Jais	Other	Machine	Human
Train	English	11,999	11,995	-	-	35,036	59,030	62,994
	Chinese	2,964	2,970	-	-	-	5,934	6,000
	Urdu	-	2,899	-	-	-	2,899	3,000
	Bulgarian	3,000	3,000	-	-	-	6,000	6,000
	Indonesian	-	3,000	-	-	-	3,000	3,000
Dev	Russian	500	500	-	-	-	1,000	1,000
	Arabic	-	500	-	-	-	500	500
	German	-	500	-	-	-	500	500
Test	English	3,000	3,000	-	-	9,000	15,000	13,200
	Arabic	-	1,000	-	100	-	1,100	1,000
	German	-	3,000	-	-	-	3,000	3,000
	Italian	-	-	3,000	-	-	3,000	3,000

Table 7: M4 Multilingual Dataset

Category	**Values**
Models	ChatGPT, GPT-4, GPT-3 (text-davinci-003), GPT-2 XL, Llama 2 70B (Chat), Cohere, Cohere (Chat), MPT-30B, MPT-30B (Chat), Mistral 7B, Mistral 7B (Chat)
Domains	ArXiv Abstracts, Recipes, Reddit Posts, Book Summaries, NYT News Articles, Poetry, IMDb Movie Reviews, Wikipedia, Czech News, German News, Python Code
Adversarial Attacks	Article Deletion, Homoglyph, Number Swap, Paraphrase, Synonym Swap, Misspelling, Whitespace Addition, Upper-Lower Swap, Zero-Width Space, Insert Paragraphs, Alternative Spelling

Table 8: The content of the Raid datasets

attacks are universally labeled as Hard-LGT, regardless of their source generator. In contrast, text generated by the remaining source LLMs in the dataset, whose outputs can be consistently detected with high confidence across multiple mainstream detectors, is categorized as easy-LGT.

B Theories and Proofs

B.1 Proof of Smoothing Laplacian Manifold

Given an adjacency matrix $W = [w_{ij}] \in \mathbb{R}^{N\times N}$, we define the unnormalized graph Laplacian as

$$\mathcal{L} = D - W, \tag{20}$$

where D denotes the degree matrix with diagonal entries

$$D_{ii} = \sum_j w_{ij}. \tag{21}$$

The degree D_{ii} characterizes the local density of the neighborhood around node i.

To encourage samples with high similarity (i.e., large edge weights w_{ij}) to be close to each other on the learned manifold, we introduce a Laplacian smoothing regularization that minimizes the weighted Euclidean distances between neighboring embeddings, thereby enforcing smoothness with respect to the intrinsic manifold structure.

Consider an undirected weighted graph

$$\mathcal{G} = (\mathcal{V}, \mathcal{E}, W), \quad |\mathcal{V}| = N, \tag{22}$$

with an adjacency matrix

$$W = [w_{ij}] \in \mathbb{R}^{N\times N}, \quad w_{ij} = w_{ji} \geq 0, \tag{23}$$

and degree matrix

$$D = \mathrm{diag}(d_1, \ldots, d_N), \quad d_i = \sum_{j=1}^{N} w_{ij}. \tag{24}$$

Each node i is associated with an embedding vector

$$z_i \in \mathbb{R}^d. \tag{25}$$

Stacking all node embeddings yields the matrix

$$X = \begin{bmatrix} z_1^\top \\ z_2^\top \\ \vdots \\ z_N^\top \end{bmatrix} \in \mathbb{R}^{N\times d}. \tag{26}$$

The manifold regularization term induced by the graph Laplacian is given by the quadratic form

$$\mathcal{L}_{\text{manifold}} = \mathrm{Tr}\Big(X^\top \mathcal{L} X\Big) = \mathrm{Tr}\Big(X^\top (D - W) X\Big). \tag{27}$$

By linearity of the trace operator, this expression can be decomposed as

$$\mathrm{Tr}\Big(X^\top \mathcal{L} X\Big) = \mathrm{Tr}\Big(X^\top D X\Big) - \mathrm{Tr}\Big(X^\top W X\Big). \tag{28}$$

We first expand the degree term. Using the identity $\mathrm{Tr}(A) = \sum_k A_{kk}$, we obtain

$$\begin{aligned} \mathrm{Tr}\Big(X^\top D X\Big) &= \sum_{k=1}^{d} (X^\top D X)_{kk} \\ &= \sum_{i=1}^{N} d_i \sum_{k=1}^{d} X_{ik}^2 \\ &= \sum_{i=1}^{N} d_i \|z_i\|^2. \end{aligned} \tag{29}$$

Substituting $d_i = \sum_j w_{ij}$ yields

$$\mathrm{Tr}\Big(X^\top D X\Big) = \sum_{i=1}^{N}\sum_{j=1}^{N} w_{ij}\|z_i\|^2. \tag{30}$$

Next, we expand the adjacency term as

$$\begin{aligned} \mathrm{Tr}\Big(X^\top W X\Big) &= \sum_{k=1}^{d}\sum_{i,j} X_{ik}\, w_{ij}\, X_{jk} \\ &= \sum_{i,j} w_{ij}\, z_i^\top z_j. \end{aligned} \tag{31}$$

Combining the two expanded terms gives

$$\mathrm{Tr}\Big(X^\top \mathcal{L} X\Big) = \sum_{i,j} w_{ij}\|z_i\|^2 - \sum_{i,j} w_{ij} z_i^\top z_j. \tag{32}$$

Exploiting the symmetry $w_{ij} = w_{ji}$ and symmetrizing the expression, we arrive at

$$\begin{aligned} \mathrm{Tr}\Big(X^\top \mathcal{L} X\Big) = \frac{1}{2}\sum_{i,j} w_{ij} \\ \times \big(\|z_i\|^2 + \|z_j\|^2 - 2 z_i^\top z_j\big). \end{aligned} \tag{33}$$

Noting that the term inside the parentheses corresponds to the squared Euclidean distance

$$\|z_i - z_j\|^2 = \|z_i\|^2 + \|z_j\|^2 - 2 z_i^\top z_j. \tag{34}$$

We finally obtain the equivalent formulation of the Laplacian quadratic form:

$$\mathrm{Tr}\Big(X^\top \mathcal{L} X\Big) = \frac{1}{2}\sum_{i,j} w_{ij}\|z_i - z_j\|^2. \tag{35}$$

Method	GPT-4o			Claude-3.5			Gemini-2.5		
	Auroc↑	Acc↑	FPR95↓	Auroc↑	Acc↑	FPR95↓	Auroc↑	Acc↑	FPR95↓
Fast-Detect	0.5414	0.5301	0.9999	0.5512	0.5512	0.9999	0.5295	0.5065	0.9999
Detective	0.8565	0.8683	0.4733	0.9425	0.9141	0.5916	0.7882	0.7028	0.6501
ImBD	0.8462	0.7667	0.9383	0.8544	0.7683	0.8651	0.7229	0.6725	0.9983
HTAO	0.9256	0.9016	0.1483	0.8892	0.9356	0.3966	0.7735	0.6991	0.5601
DMHM (Ours)	**0.9666**	**0.9275**	**0.0951**	**0.9971**	**0.9851**	**0.0067**	**0.8314**	**0.7841**	**0.1333**

Table 9: The whole results of the unseen-LLMs generated texts detection.

Method	Unseen Domain		
	AUROC ↑	Acc ↑	FPR95 ↓
DeTeCtive	0.7958	0.8051	0.6064
ImBD	0.5841	0.5673	0.8064
HTAO	0.9702	0.9714	0.1173
DMHM (Ours)	**0.9842**	**0.9837**	**0.0829**

Table 10: The results on the unseen domain.

This formulation shows that the graph Laplacian penalizes large discrepancies between embeddings connected by edges with high affinity, thereby enforcing smoothness of the learned representations on the underlying data manifold.

We construct the adjacency matrix using a k-nearest-neighbor graph based on cosine similarity. For each sample, edges are formed to its top-k neighbors, and the edge weights are defined using an exponential kernel:

$$w_{ij} = \begin{cases} \exp\left(-\dfrac{\|z_i - z_j\|^2}{2\sigma^2}\right), & z_i \in N_k(z_j) \\ & \text{or } z_j \in N_k(z_i), \\ 0, & \text{otherwise.} \end{cases} \tag{36}$$

which guarantees non-negative affinities. Due to the directed kNN construction, the resulting adjacency matrix W is generally asymmetric. Nevertheless, the Laplacian regularization term remains well-behaved, as it can be written as a weighted smoothness penalty:

$$\sum_{i,j} w_{ij} \|x_i - x_j\|^2, \tag{37}$$

which is non-negative as long as $w_{ij} \geq 0$. This formulation provides stable manifold regularization without explicitly enforcing symmetry. In implementation, this is equivalent to applying Laplacian regularization on a directed graph or its implicit symmetrized form, which is commonly adopted in kNN-based manifold learning.

B.2 Manifold Approximation under Finite-Sample kNN Graphs

Classical manifold learning theory typically assumes access to sufficiently dense samples from the underlying data distribution, so that the intrinsic low-dimensional manifold structure can be accurately recovered. Let $\mathcal{M} \subset \mathbb{R}^d$ be a compact and smooth manifold with intrinsic dimension $m \ll d$. Assume that samples $\{z_i\}_{i=1}^N$ are drawn independently from a probability distribution $p(z)$ supported on $\mathcal{M}$. In the asymptotic regime $N \to \infty$, it is well known that graph-based constructions, such as k-nearest neighbor (kNN) graphs, converge to the geometry of $\mathcal{M}$, and the associated graph Laplacian converges to the Laplace–Beltrami operator on the manifold. In practical training settings, however, manifold regularization is computed on mini-batches (e.g., $B = 128$), which are far too small to globally approximate the full data manifold. As a result, the kNN graph constructed on a mini-batch cannot recover the global topology of $\mathcal{M}$. Instead, it provides a local stochastic approximation of the manifold geometry induced by random batch sampling.

Mathematical Framework for Local Stochastic Approximation. Let $\mathcal{X} = \{z_i\}_{i=1}^N$ denote the full dataset, and let $\mathcal{B} = \{z_i\}_{i=1}^B \subset \mathcal{X}$ be a randomly sampled mini-batch. The empirical distribution of the batch is defined as

$$p_B(z) = \frac{1}{B} \sum_{i=1}^{B} \delta(z - z_i), \tag{38}$$

where $\delta(\cdot)$ denotes the Dirac delta function. The kNN graph on the batch is denoted by $\mathcal{G}_B = (V_B, W_B)$, with weight matrix $W_B = (w_{ij})$ defined as

$$w_{ij} = \begin{cases} \exp\left(-\dfrac{\|z_i - z_j\|^2}{2\sigma^2}\right), & z_i \in N_k(z_j) \\ & \text{or } z_j \in N_k(z_i), \\ 0, & \text{otherwise.} \end{cases} \tag{39}$$

Method	Raid			Dipper			Outfox		
	AUROC↑	Acc↑	FPR95↓	AUROC↑	Acc↑	FPR95↓	AUROC↑	Acc↑	FPR95↓
Fast-Detect	0.6172	0.6646	0.7932	0.6077	0.6512	0.4390	0.7030	0.6634	0.5869
DeTeCtive	0.8856	0.8822	0.5833	0.9619	0.9680	0.1319	0.9818	0.9850	0.0992
ImBD	0.7843	0.7435	0.8426	0.8841	0.8039	0.4152	0.9495	0.9562	0.1869
HTAO	0.9633	0.9897	0.0953	0.9735	0.9778	0.0832	0.9853	0.9894	0.0718
DMHM (Ours)	**0.9902**	**0.9927**	**0.0605**	**0.9870**	**0.9907**	**0.0735**	**0.9951**	**0.9984**	**0.0201**

Table 11: The whole results under attacks.

Method (Ablation)	Deepfake			M4			Raid		
	AUROC↑	Acc↑	FPR95↓	AUROC↑	Acc↑	FPR95↓	AUROC↑	Acc↑	FPR95↓
w/o L_{manifold}	0.9477	0.9102	0.4259	0.9199	0.8491	0.4222	0.9615	0.9885	0.1538
w/o L_{energy}	0.9272	0.9225	0.5120	0.8370	0.7607	0.5246	0.9047	0.9617	0.3925
w/o L_{con}	0.9228	0.9129	0.5198	0.9275	0.8507	0.2544	0.9129	0.9717	0.1857
w/ Euclidean Distance	0.9238	0.9084	0.7305	0.8769	0.8333	0.8911	0.9513	0.9608	0.2547
w/o density weight α_i	0.9354	0.9265	0.8269	0.9209	0.8788	0.1864	0.9594	0.9712	0.1621
w/o Hard–Easy Modeling	0.9214	0.9108	0.3909	0.8801	0.8332	0.5871	0.9008	0.9348	0.3837
DMHM (Ours)	**0.9886**	**0.9887**	**0.0456**	**0.9666**	**0.9591**	**0.1446**	**0.9902**	**0.9927**	**0.0605**

Table 12: Ablation study results on three datasets. The impact of each component is evaluated by removing or replacing it.

where $N_k(z_i)$ denotes the set of k nearest neighbors of z_i within $\mathcal{B}$, and σ is a bandwidth parameter. Although the kNN relation is constructed in a directed manner, the final weight matrix W_B is symmetrized by taking the union of mutual neighborhoods. This ensures that W_B is symmetric and the resulting graph Laplacian corresponds to an undirected, unnormalized Laplacian as used.

The corresponding (unnormalized) graph Laplacian is defined as $\mathcal{L}_B = D_B - W_B$, where $D_B = \operatorname{diag}(\sum_j w_{ij})$ is the degree matrix. The exponential kernel is used solely to define edge weights that decay with local distance, and does not introduce any normalization on the Laplacian. After symmetrizing the kNN graph, the resulting weight matrix W_B is symmetric, and the operator $\mathcal{L}_B = D_B - W_B$ remains an *unnormalized* graph Laplacian, consistent with the formulation in Eq. 2. The manifold regularization term computed on a batch is given by:

$$L^{(B)}_{\text{manifold}}(f) = \tfrac{1}{kB} \sum_{i=1}^{B} \sum_{j \in N_k(z_i)} w_{ij} \|f(z_i) - f(z_j)\|^2, \tag{40}$$

where $f : \mathcal{M} \to \mathbb{R}^m$ is the embedding function. After symmetrizing the kNN graph, the resulting weight matrix W_B is symmetric. Thus, up to a constant scaling factor, the mini-batch regularizer $L^{(B)}_{\text{manifold}}$ is equivalent to $\operatorname{Tr}(F^\top \mathcal{L}_B F)$ with $\mathcal{L}_B = D_B - W_B$, which is consistent with the unnormalized Laplacian formulation.

Expectation Convergence Analysis. To study the statistical behavior of this regularizer, we consider its expectation over random mini-batches. Let $\mathcal{B}$ be drawn i.i.d. from $p(z)$ with size B, and define the expected manifold regularization as

$$\bar{L}_{\text{manifold}}(f) = \mathbb{E}_{\mathcal{B} \sim p^B} \left[L^{(B)}_{\text{manifold}}(f) \right], \tag{41}$$

where p^B denotes the B-fold product measure of $p(z)$.

Under the following assumptions: (1) The manifold $\mathcal{M}$ is compact, C^2 smooth, and has bounded curvature; (2) The density $p(z)$ is continuous and bounded on $\mathcal{M}$, and there exist constants $c_1, c_2 > 0$ such that $c_1 \le p(z) \le c_2$; (3) The embedding function $f \in C^2(\mathcal{M})$; (4) As $B \to \infty$, $k \to \infty$ with $k/B \to 0$, the bandwidth $\sigma \to 0$, and $\sigma > \Theta\big((\log B/B)^{1/m}\big)$, the following asymptotic expansion holds:

$$\begin{aligned} \bar{L}_{\text{manifold}}(f) = C_1(\sigma, m) \int_{\mathcal{M}} \|\nabla_{\mathcal{M}} f(z)\|^2 p^2(z)\, dV(z) \\ + O\left(\frac{1}{\sqrt{k}}\right) + O\left(\frac{\sigma^2}{k^{1/m}}\right), \end{aligned} \tag{42}$$

where

$$C_1(\sigma, m) = \frac{\sigma^{m+2}}{(2\pi)^{m/2}} \int_{\mathbb{R}^m} \|u\|^2 e^{-\|u\|^2/2}\, du,$$

and dV denotes the volume element on $\mathcal{M}$.

Finite-Sample Bias Analysis. For a fixed batch

size B, define the bias term as

$$\mathcal{E}_B(f) = \left| \bar{L}_{\text{manifold}}(f) - C_1 \int_{\mathcal{M}} \|\nabla_{\mathcal{M}} f\|^2 p^2 \, dV \right|. \tag{43}$$

Under the assumptions of the theorem, there exist constants $C_2, C_3 > 0$ such that

$$\mathcal{E}_B(f) \leq C_2 \frac{\|\nabla^2_{\mathcal{M}} f\|_\infty}{\sqrt{k}} + C_3 \frac{\sigma^2}{k^{1/m}} \|\nabla_{\mathcal{M}} f\|^2_\infty. \tag{44}$$

This result shows that even with small mini-batches, the bias can be controlled as long as k is sufficiently large and σ is chosen properly. In particular, choosing

$$k = \Theta\left(B^{\frac{2m}{m+2}}\right), \qquad \sigma = \Theta\left(B^{-\frac{1}{m+2}}\right),$$

yields a bias decay rate of $O\left(B^{-\frac{1}{m+2}}\right)$.

Variance Analysis and Batch Size Trade-off. In addition to bias, we must account for the variance due to random batch sampling. Define the variance term as

$$\text{Var}_B(f) = \mathbb{E}_{\mathcal{B}} \left[\left(L^{(B)}_{\text{manifold}}(f) - \bar{L}_{\text{manifold}}(f) \right)^2 \right]. \tag{45}$$

There exists a constant $C_4 > 0$ such that

$$\text{Var}_B(f) \leq \frac{C_4}{B} \|\nabla_{\mathcal{M}} f\|^4_\infty. \tag{46}$$

Combining the bias and variance terms, we obtain the following mean squared error bound:

$$\mathbb{E}\left[\left(L^{(B)}_{\text{manifold}}(f) - C_1 \int_{\mathcal{M}} \|\nabla_{\mathcal{M}} f\|^2 p^2 \, dV \right)^2 \right] \leq \mathcal{E}_B^2(f) + \text{Var}_B(f). \tag{47}$$

This bound shows that as the batch size B increases, the variance decreases at a rate $O(1/B)$, while the bias is governed by k and σ. The performance saturation observed in experiments (Table 13) can thus be explained by the fact that, beyond a certain batch size, the bias term dominates and further increasing B yields diminishing returns.

For discriminative tasks, preserving local geometry is often more important than recovering global topology. We define a local distortion measure as

$$\kappa(z_i) = \frac{\|\nabla_{\mathcal{M}} f(z_i)\|^2}{\frac{1}{k}\sum_{j \in N_k(z_i)} w_{ij} \|f(z_i) - f(z_j)\|^2} \cdot \frac{\frac{1}{k}\sum_{j \in N_k(z_i)} w_{ij} \|z_i - z_j\|^2}{\|\nabla_{\mathcal{M}} \text{Id}(z_i)\|^2}, \tag{48}$$

where $\text{Id} : \mathcal{M} \to \mathbb{R}^d$ is the identity embedding, and $\kappa(z_i) \approx 1$.

Under the assumptions of the theorem, the following concentration inequality holds:

$$\mathbb{P}\left(|\kappa(z_i) - 1| > \epsilon\right) \leq 2 \exp\left(-\frac{k\epsilon^2}{2C_5}\right), \qquad \forall \epsilon > 0, \tag{49}$$

where C_5 is a constant depending on the manifold curvature and density variation. This result implies that, with high probability and sufficiently large k, the kNN graph preserves the local differential geometry of the manifold, which is exactly the first-order geometric consistency required for discriminative learning.

We further analyze the effect of batch size on the effectiveness of the proposed manifold-aware regularization. As shown in Table 13, the performance consistently improves as the batch size increases from 32 to 128, indicating that enlarging the batch provides more reliable local neighborhood statistics and a better approximation of the underlying data geometry. Notably, the performance gains exhibit a clear diminishing-return behavior beyond a batch size of 128. While increasing the batch size from 32 to 64 and from 64 to 128 yields measurable improvements in AUROC and accuracy, further enlarging the batch size to 256 or 512 does not lead to additional benefits and even causes slight degradation. This observation suggests that a batch size of 128 is already sufficient to capture the essential local manifold structure required for effective discrimination.

In summary, although kNN graphs constructed on finite mini-batches cannot recover the global manifold topology, they provide an effective stochastic approximation of local manifold geometry. The expected regularization term converges to a density-weighted Dirichlet energy, with both bias and variance being controllable. The observed performance saturation with increasing batch size is consistent with the theory: once the batch size exceeds a certain threshold, the bias term dominates the error. Overall, the proposed method can be viewed as a stochastic local manifold regularizer that enforces global manifold-aware structure in expectation through repeated local smoothing constraints, while avoiding the high computational cost of constructing a full graph.

Batch Size	AUROC ↑	Acc ↑	FPR95 ↓
32	0.9786	0.9779	0.0974
64	0.9839	0.9822	0.0896
128	**0.9886**	**0.9887**	**0.0456**
256	0.9855	0.9858	0.0851
512	0.9847	0.9850	0.0879

Table 13: Effect of batch size on *deepfake* datasets. Batch size 128 achieves the best marginal performance, balancing neighborhood reliability and noise robustness.

B.3 Covariance Adaptive Normalization

One-Dimensional Case. Let the feature $z \in \mathbb{R}$ be one-dimensional. The covariance matrix reduces to a scalar

$$\Sigma = \sigma^2. \tag{50}$$

The Mahalanobis distance becomes

$$d^2 = \frac{(z-\mu)^2}{\sigma^2}. \tag{51}$$

When the variance σ^2 is large, the denominator increases and the distance decreases, implying that this direction is down-weighted in the metric. Conversely, when σ^2 is small, the denominator decreases and the distance increases, implying that this direction is up-weighted.

Multidimensional Case. Let $z \in \mathbb{R}^d$ and let the covariance matrix $\Sigma \in \mathbb{R}^{d\times d}$ be symmetric and positive definite. Its eigendecomposition is given by

$$\Sigma = Q\Lambda Q^T, \tag{52}$$

where Q is an orthogonal matrix whose columns are the eigenvectors of Σ, and $\Lambda = \operatorname{diag}(\lambda_1, \ldots, \lambda_d)$ is a diagonal matrix of eigenvalues, with each λ_i representing the variance along the corresponding principal direction.

Define the coordinates in the principal-axis (PCA) basis as

$$y = Q^T(z-\mu). \tag{53}$$

The covariance of y is

$$\operatorname{Cov}(y) = Q^T\mathbb{E}\big[(z-\mu)(z-\mu)^T\big]\,Q. \tag{54}$$

Then, we obtain

$$\operatorname{Cov}(y) = Q^T\Sigma Q. \tag{55}$$

Substituting $\Sigma = Q\Lambda Q^T$ and using the orthogonality property $Q^TQ = I$, we have

$$\operatorname{Cov}(y) = \Lambda = \operatorname{diag}(\lambda_1, \ldots, \lambda_d). \tag{56}$$

Since $y = Q^T(z-\mu)$, it follows that

$$z - \mu = Qy. \tag{57}$$

The squared Mahalanobis distance is

$$D_M^2 = (z-\mu)^T\Sigma^{-1}(z-\mu). \tag{58}$$

Substituting $z - \mu = Qy$ yields

$$D_M^2 = (Qy)^T\Sigma^{-1}(Qy) = y^TQ^T\Sigma^{-1}Qy. \tag{59}$$

Since

$$\Sigma^{-1} = (Q\Lambda Q^T)^{-1} = Q\Lambda^{-1}Q^T, \tag{60}$$

we obtain

$$D_M^2 = y^T\Lambda^{-1}y. \tag{61}$$

Noting that

$$\Lambda^{-1} = \operatorname{diag}\left(\frac{1}{\lambda_1}, \ldots, \frac{1}{\lambda_d}\right), \tag{62}$$

The Mahalanobis distance can be written as

$$D_M^2 = \sum_{i=1}^{d}\frac{y_i^2}{\lambda_i}. \tag{63}$$

Therefore, if the variance λ_i along principal direction i is large, the corresponding term $\frac{y_i^2}{\lambda_i}$ is small, and that direction is down-weighted in the distance computation; conversely, if λ_i is small, that direction is up-weighted. Here y_i denotes the i-th coordinate of z in the PCA coordinate system.

B.4 Effect of Batch Size on Density Estimation and α_i Normalization.

The batch size plays a critical role in local density estimation and the stability of the α_i normalization. When the batch size is small, kNN graphs are constructed from sparsely sampled neighborhoods, resulting in high-variance density estimates and noisy α_i values that are sensitive to stochastic sampling. As the batch size increases, neighborhood coverage becomes more reliable, which reduces variance in the density estimation and yields smoother, more stable α_i normalization. This improvement allows the regularization term to more accurately approximate a density-weighted Dirichlet energy, leading to performance gains in the moderate batch-size regime.

As shown in Table 13, the performance improvement saturates once the batch size exceeds a certain

threshold (e.g., 128). Beyond this point, further increasing the batch size provides limited benefit, as the variance in density estimation has already been sufficiently suppressed and the remaining error is dominated by the bias introduced by the finite-batch kNN approximation. As a result, larger batches do not yield significant additional gains in stability or performance, despite incurring higher computational cost.

C More Analysis Experiments

C.1 Analyses of Threshold m_1 & m_2

To study the effect of distance margins in energy margin learning, we conduct a grid search over hyperparameters m_1 and m_2, where m_1 denotes the energy margin between easy-LGT and HWT, and m_2 denotes the minimum margin between hard-LGT and HWT. Since easy-LGT is compact while hard-LGT lies closer to HWT, we enforce the constraint $m_1 > m_2$ to avoid over-constraining hard-LGT. The results are shown in Fig. 4.

The heatmap indicates that the model is highly robust in the lower triangular region ($m_1 > m_2$), with performance consistently above 0.94. The best performance (0.9886) is achieved at $(m_1 = 2, m_2 = 1.5)$, suggesting that enlarging the margin between LGT and HWT effectively improves inter-distribution discrimination. In contrast, overly small or large margins degrade performance, implying increased optimization difficulty.

When the hierarchical constraint is violated ($m_1 \leq m_2$), performance drops to the range of 0.85–0.91 and further decreases as the margin gap $(m_2 - m_1)$ increases. This degradation indicates that enforcing equal or larger margins on hard-LGT disrupts the intended energy hierarchy and leads to suboptimal discrimination. These results demonstrate that the constraint $m_1 > m_2$ is essential rather than a mere hyperparameter preference.

C.2 Hyperparameter Analysis of k and r

We analyze the sensitivity of our method to the key hyperparameters involved in KNN-based density estimation and manifold regularization on the *deepfake* datasets, including: (1) the number of nearest neighbors k used for local covariance estimation, and (2) the neighborhood size r used to construct the manifold graph. All other settings are fixed throughout the experiments, as shown in Fig. 5.

Overall, the performance exhibits low sensitivity to both k and r within a reasonable range, indicating good robustness of the proposed method. When k is set too small, the estimated local covariance becomes unstable due to insufficient neighborhood samples, leading to performance degradation. In contrast, overly large k results in excessive smoothing, which weakens local geometric characteristics and slightly harms performance. In practice, we observe a broad performance plateau, where the accuracy remains consistently high, with $k = 10$ providing the best balance between estimation stability and locality preservation. A similar trend is observed for the neighborhood size r. Smaller values of r restrict the expressive power of the neighborhood graph and limit information propagation, while excessively large r tends to introduce non-local connections that violate the manifold assumption. Nevertheless, the performance variation across different r values is relatively minor, suggesting that the method is not overly sensitive to this parameter. Empirically, $r = 10$ consistently yields strong and stable results. These observations indicate that the proposed method does not rely on careful hyperparameter tuning. As long as k and r are chosen within a moderate range, the performance remains stable. We therefore adopt

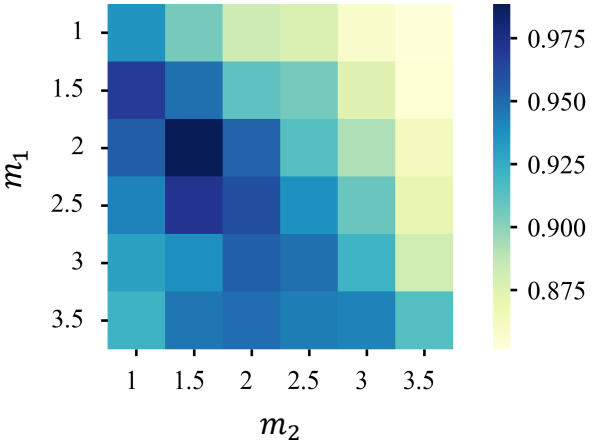

Figure 4: The heatmap of hyperparameter sensitivity analyses of the threshold m_1 and m_2.

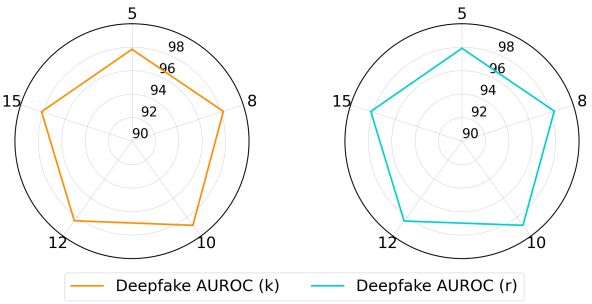

Figure 5: The hyperparameter sensitivity analyses.

Table 14: Ablation study on the weight coefficients of different loss terms.

$(\lambda_1, \lambda_2, \lambda_3)$	**AUROC** ↑	**Acc** ↑	**FPR95** ↓
$(0, 1, 1)$	0.9272	0.9225	0.5120
$(1, 0, 1)$	0.9228	0.9129	0.5198
$(1, 1, 0)$	0.9477	0.9102	0.4259
$(2, 1, 1)$	0.9821	0.9746	0.0813
$(1, 2, 1)$	0.9834	0.9758	0.0789
$(1, 1, 2)$	0.9852	0.9779	0.0742
$(0, 2, 1)$	0.9316	0.9261	0.4882
$(2, 0, 1)$	0.9289	0.9186	0.5035
$(1, 0, 2)$	0.9301	0.9197	0.4981
$(0, 1, 2)$	0.9342	0.9279	0.4720
$(2, 1, 0)$	0.9513	0.9217	0.4016
$(1, 2, 0)$	0.9538	0.9243	0.3897
$(1, 1, 1)$	**0.9886**	**0.9887**	**0.0456**

the default setting $k = r = 10$ in all experiments, which offers a reliable trade-off between robustness and accuracy.

In addition, we find that the performance is largely insensitive to the batch size used for KNN-based estimation, provided that each batch contains a sufficient number of samples to form meaningful local neighborhoods (as Sec. B.4). Tiny batch sizes may lead to noisy neighborhood statistics, while moderate to large batch sizes yield comparable performance.

C.3 Hyperparameter Analysis of Loss Weights

We conducted an ablation study on the loss coefficients. Specifically, the proposed objective is formulated as:

$$\mathcal{L}_{\text{our}} = \lambda_1 \mathcal{L}_{\text{energy}} + \lambda_2 \mathcal{L}_{\text{con}} + \lambda_3 \mathcal{L}_{\text{d-manifold}}, \quad (64)$$

where λ_1, λ_2, and λ_3 control the contributions of the three sub-objectives.

We evaluate three categories of weight configurations: (1) Removing one loss term. (2) Increasing the weight of a single objective. (3) Asymmetric weight assignments. The results are reported in Table 14.

First, removing any individual objective causes a substantial performance degradation, demonstrating that each sub-loss contributes meaningfully to the final representation learning process. Second, although unbalanced weight settings still produce competitive results, they remain consistently inferior to the balanced configuration. This suggests that over-emphasizing any single objective may weaken the complementary effects among the three objectives.

Overall, the balanced setting $(1, 1, 1)$ achieves the best AUROC, accuracy, and FPR95 simultaneously, indicating that the three optimization targets play comparably important and mutually beneficial roles during training.

C.4 Transferability Across Base Encoders

We further investigate the transferability of our framework by employing various base encoders on the Deepfake dataset. We specifically consider SimCSE-based encoders, as they utilize contrastive learning to refine semantic representation and promote embedding uniformity. Experimental results in Table 15 indicate that Unsup-SimCSE-RoBERTa$_{\text{base}}$ strikes the best balance between detection accuracy and model efficiency (parameter size). The stable performance gains observed across different backbones underscore that our method is not tied to a specific architecture but possesses broad applicability.

Notably, despite not being explicitly trained on all languages in the M4 dataset, Unsup-SimCSE-RoBERTa$_{\text{base}}$ maintains strong performance on unsupported languages. This is mainly due to the contrastive learning objective, which promotes language-agnostic semantic representations and a well-structured embedding space.

Text Encoders	**Params**	**AUROC**	**Acc**	**FPR95**
BERT$_{\text{base}}$	110M	0.9624	0.9712	0.0821
Sup-SimCSE-BERT$_{\text{base}}$	110M	0.9615	0.9683	0.0796
Unsup-SimCSE-BERT$_{\text{base}}$	110M	0.9528	0.9691	0.0734
RoBERTa$_{\text{base}}$	125M	0.9602	0.9768	0.0779
Sup-SimCSE-RoBERTa$_{\text{base}}$	125M	0.9716	0.9759	0.0853
Unsup-SimCSE-RoBERTa$_{\text{base}}$	125M	**0.9894**	**0.9881**	**0.0417**

Table 15: Our method was evaluated on the encoder transferability experiments conducted using the Cross-Domain and Cross-Model subsets of the Deepfake dataset (Li et al., 2024). **Unsup** refers to unsupervised training without the use of manually annotated data, while **Sup** denotes supervised learning with annotated pairs of similar and dissimilar sentences.

C.5 Evaluation of Time and Memory Efficiency

Table 16 summarizes the computational overhead of baselines and our method, encompassing training and inference latency alongside model complexity. In practical deployment, inference speed and memory footprint are the primary determinants of user experience and operational viability—factors

Method	Training Time	Inference Time	Memory	Model Size
Fast-detect	\	98.52 ms/it	11662 MB	\
MMD-MP	6.87 ms/it	26.62 ms/it	2932 MB	125M
Detective	4.85 ms/it	1.76 ms/it	2803 MB	125M
ImBD	234.74 ms/it	371.74 ms/it	10422 MB	2.72B
HTAO	4.83 ms/it	1.74 ms/it	2513 MB	125M
Ours	4.82 ms/it	1.84 ms/it	1999 MB	125M

Table 16: Evaluation of time and memory efficiency across different methods, \ indicates that the model size is not fixed.

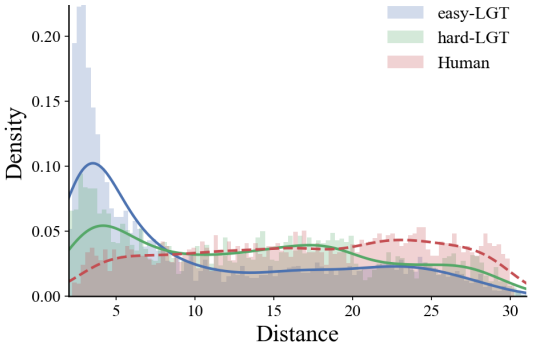

Figure 6: Density distributions of sample distances to the unified semantic center. The easy-LGT (blue) exhibits a compact, high-density peak. In contrast, the hard-LGT (green) shows a flatter, dispersed distribution that significantly overlaps with the low-density regions of HWT (red), illustrating the difficulty in distinguishing high-quality generated text.

that often outweigh training-phase efficiency. As shown in the table, our method achieves a highly competitive inference speed of 1.84 ms/it. This represents an approximate 14.5× speedup over MMD-MP and a remarkable 200× improvement compared to the large-scale ImBD detector. While HTAO and Detective exhibit marginally lower inference latency, our approach offers the most balanced performance profile.

Regarding training efficiency, our method attains the lowest cost among all evaluated approaches at 4.82 ms/it, significantly outperforming ImBD (234.74 ms/it) and even slightly surpassing efficient variants like HTAO and Detective. Furthermore, while zero-shot methods such as Fast-detect eliminate training entirely, they suffer from substantial inference latency (98.52 ms/it) and reliance on external scoring models with variable parameters. In terms of memory usage, Ours achieves the lowest memory consumption (1999 MB) among all 125M-parameter models, demonstrating superior memory efficiency. Our model maintains a compact footprint of 125M parameters, ensuring high scalability. In conclusion, our method strikes a superior trade-off between detection efficiency and computational expenditure, making it exceptionally well-suited for real-time, large-scale online deployment.

Dataset	Acc	AUROC	FPR95
Deepfake	2.14E-05	4.82E-11	1.95E-06
M4-multi	5.08E-06	3.17E-13	6.22E-08
Raid	4.27E-04	2.11E-06	2.03E-04

Table 17: The p-values of the t-test on our method in all metrics, which are all smaller than 0.01.

C.6 Statistical Significance Testing

The T-test (Bartlett, 1937) is a widely used statistical method for assessing the significance of experimental results. It determines whether the observed differences between two sets of results are due to a true effect rather than random variation. The outcomes of the T-test are typically reported using p-values, which represent the probability of obtaining the observed data under the null hypothesis H_0 (i.e., assuming no effect or difference). When $p < 0.05$, the null hypothesis is rejected, indicating that the difference is statistically significant. In our study, we employ the T-test to evaluate whether the improvements introduced by our method are statistically significant. As shown in the Tab. 17, all p-values are less than **0.01**, confirming the statistical significance of our improvements.

D Observation

We visualize the density of sample distances to a unified center. As illustrated in Fig. 6, the X-axis represents the distance of samples from a global centroid, while the Y-axis is the density, which is estimated by centering a Gaussian kernel at each sample point. The easy-LGT concentrates in a high-density region with a sharp peak at short distances, reflecting statistical regularities of weaker LLMs. Conversely, hard-LGT exhibits a flatter, more dispersed distribution that significantly overlaps with the low-density regions of HWTs, demonstrating how stronger LLMs achieve higher diversity that mimics human writing. This clear distinction, where easy samples cluster in dense regions while hard samples spread into sparse, human-shared regions, confirms that density is a critical discriminator, justifying the need for a density-aware framework to effectively separate these composite distributions.

E Societal Impacts

As large language models (LLMs) generate increasingly human-like text, societal risks such as misinformation, malicious content, and academic misconduct have intensified, making reliable detection of LGTs an urgent challenge. At the same time, false positives (misclassifying HWTs) can cause serious harm in academic and journalistic settings. To address it, our proposed method achieves state-of-the-art performance on text generated by recent LLMs and adversarial examples, while maintaining a low false positive rate and a substantially higher true positive rate than existing approaches. This technical advancement provides an efficient and robust solution for LGT detection, ensuring long-term effectiveness in complex, real-world scenarios and offering critical support for fostering a trustworthy digital environment and promoting the safe, ethical, and responsible use of AI technologies.

F AI Assistants Disclosure

AI tools were used solely for language polishing, grammar correction, improvement of English expressions, and correction of code syntax errors.