Beyond Binary Classification: A Semi-supervised Approach to Generalized AI-generated Image Detection

Hong-Hanh Nguyen-Le [1], Van-Tuan Tran[2], Thuc D. Nguyen[3], Nhien-An Le-Khac[1]

[1] School of Computer Science, University College Dublin, Ireland
[2] School of Computer Science and Statistics, Trinity College Dublin, Ireland
[3] Department of Knowledge Engineering, University of Science, VNU-HCMC, Vietnam
hong-hanh.nguyen-le@ucdconnect.ie, tranva@tcd.ie, ndthuc@fit.hcmus.edu.vn, an.lekhac@ucd.ie

Abstract

The rapid advancement of generators (e.g., StyleGAN, Midjourney, DALL-E) has produced highly realistic synthetic images, posing significant challenges to digital media authenticity. These generators are typically based on a few core architectural families, primarily Generative Adversarial Networks (GANs) and Diffusion Models (DMs). A critical vulnerability in current forensics is the failure of detectors to achieve cross-generator generalization, especially when crossing architectural boundaries (e.g., from GANs to DMs). We hypothesize that this gap stems from fundamental differences in the artifacts produced by these **distinct architectures**. In this work, we provide a theoretical analysis explaining how the distinct optimization objectives of the GAN and DM architectures lead to different manifold coverage behaviors. We demonstrate that GANs permit partial coverage, often leading to boundary artifacts, while DMs enforce complete coverage, resulting in over-smoothing patterns. Motivated by this analysis, we propose the **Tri**archy **Detect**or (TriDetect), a semi-supervised approach that enhances binary classification by discovering latent architectural patterns within the "fake" class. TriDetect employs balanced cluster assignment via the Sinkhorn-Knopp algorithm and a cross-view consistency mechanism, encouraging the model to learn fundamental architectural distincts. We evaluate our approach on two standard benchmarks and three in-the-wild datasets against 13 baselines to demonstrate its generalization capability to unseen generators.

Introduction

The rapid advancement of generative models (GMs), especially GANs (WhichFaceisReal 2019) and DMs (Huang et al. 2023), enables the creation of highly realistic synthetic images that are often indistinguishable from real photographs (Lu et al. 2023). While various detection methods have been proposed (Zhang, Karaman, and Chang 2019; Liu et al. 2021; Frank et al. 2020; Wang et al. 2023; Tan et al. 2025), their most critical limitation is the poor cross-generator generalization (Deng et al. 2024; Shilin et al. 2025; Yan et al. 2025b; Chen et al. 2022). **Cross-generator generalization**, in this context, refers to the ability of a detector trained on one set of GMs to successfully identify images produced by models it has never encountered. This failure is particularly pronounced when crossing the boundaries of underlying *architectural families*, such as between GANs and DMs. This generalization gap is a critical obstacle, as new generators constantly emerge in the real world (Nguyen-Le et al. 2025).

To address this generalization gap, the fundamental challenge lies in understanding why different architectural families produce systematically different patterns. Prior research has primarily explored that GANs and DMs leave surface-level artifacts in synthetic images, such as checkerboard patterns (Dzanic, Shah, and Witherden 2020; Zhang, Karaman, and Chang 2019; Frank et al. 2020; Wang et al. 2020a) and noise residuals (Wang et al. 2023; Ma et al. 2024). However, this approach faces an inherent limitation: these surface-level artifacts are vulnerable to simple post-processing operations, further limiting detection robustness. In contrast to prior work, we shift from analyzing surface-level artifacts to **latent architectural patterns** inherent to each generative architecture family. Specifically, we hypothesize that latent representations of synthetic images generated by GANs and DMs form significantly different submanifolds in feature space. This hypothesis stems from the fundamental observation that GANs and DMs have distinct optimization objectives, which fundamentally shape how these generative architecture families model data distribution: the Jensen-Shannon (JS) divergence is optimized by GANs, whereas the Kullback-Leibler (KL) divergence is minimized by DMs.

In this work, we demonstrate the existence of latent architectural patterns inherent to each generative architecture family (i.e., GANs and DMs) through a theoretical analysis of their optimization objectives. By utilizing the manifold hypothesis (Fefferman, Mitter, and Narayanan 2016; Loaiza-Ganem et al. 2024), we show that GANs and DMs interact with the data manifold differently: GANs permit partial manifold coverage, leading to boundary artifacts, while DMs enforce complete coverage, resulting in over-smoothing patterns. This theoretical disparity implies that GAN-generated and DM-generated images occupy distinct, separable submanifolds within a well-learned feature space (Fig. 2). This theoretical analysis directly guides our method design. Rather than attempting to detect specific generators or relying on surface-level patterns, we propose **Tri**archy **Detect**or (TriDetect), a novel semi-supervised method designed to recognize these fundamental architectural signatures. Specifically, TriDect simultaneously performs binary classification while discovering **latent architectural pat-**

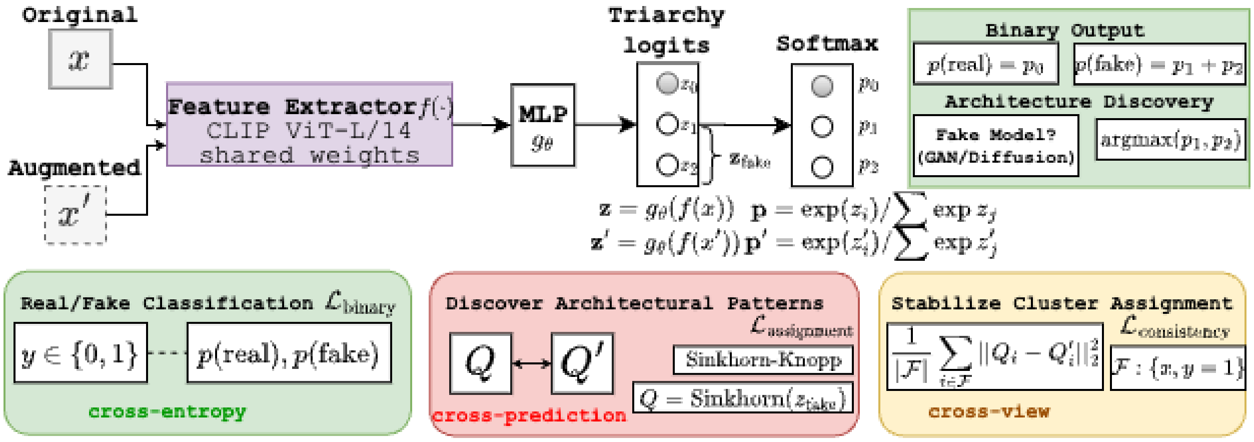

Figure 1: Overview of our proposed method, TriDetect.

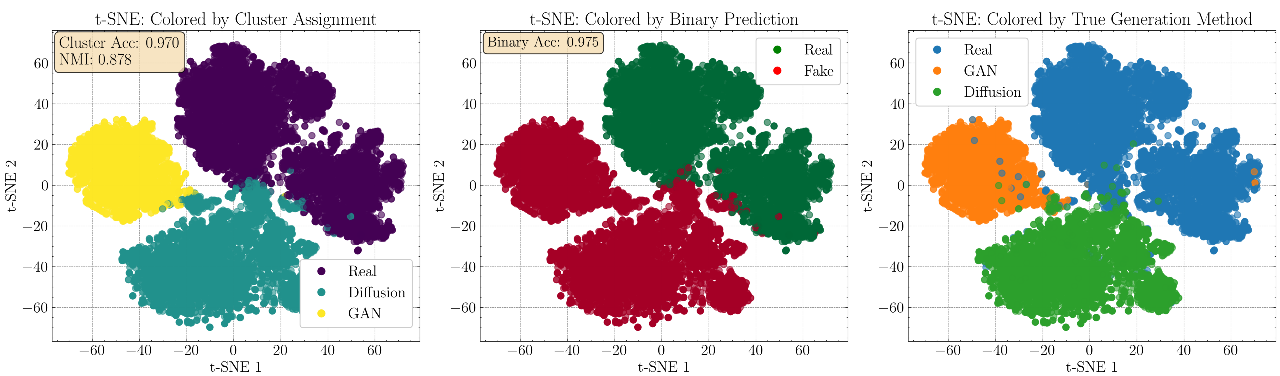

Figure 2: Visualization of learned representations demonstrating successful discovery of fake sub-types. The three t-SNE projections display feature embeddings colored by (left) the model's unsupervised cluster assignments, (middle) the model's binary real/fake predictions, and (right) the ground-truth generation methods. Results are performed on AIGCDetectBenchmark.

terns within the synthetic data (Fig. 1). To ensure robust learning, we employ the Sinkhorn-Knopp algorithm (Cuturi 2013) to enforce balanced cluster assignments, preventing cluster collapse. Furthermore, a cross-view consistency mechanism is designed to ensure the model learns fundamental architectural distinctions rather than image statistics. By discovering latent architectural patterns, our method can enhance the cross-generator generalization within the same architecture families.

Main Contributions. Our contributions include:

- **Theoretical Analysis**: We provide the first theoretical explanation for why the GAN and DM architectures produce fundamentally different latent patterns, grounded in their optimization objectives and manifold coverage.
- **Novel Semi-supervised Detection Method**: We introduce TriDetect that combines binary classification with architecture-aware clustering to learn generalized architectural features.
- **Comprehensive Evaluation**: We evaluate on 5 datasets against 13 SoTA detectors, demonstrating superior cross-generator and cross-dataset generalization.

From Divergence to Detection: A Theoretical Analysis of Architectural Artifacts

In this section, we provide the theoretical analysis to demonstrate that latent representations of synthetic images generated by GANs and DMs form significantly different submanifolds in feature space. The t-SNE visualization in Figure 2 confirms this analysis. Specifically, our analysis reveals that these differences are fundamental consequences of their optimization targets. We begin by examining the optimization objectives of each architecture family, then demonstrate how these objectives lead to characteristic coverage patterns and to distinguishable artifacts.

The Fundamental Disparity in Optimization

We use p_{data}, p_{GAN}, and p_{DM} to denote the data, GAN-generated, and DM-generated distributions, respectively.

Lemma 1. *(DM Optimization Objective). Suppose that $x_0 \sim p_{data}$, DM minimizes an upper bound on the KL divergence between the data and model distributions:*

$$D_{KL}(p_{data} \| p_{DM}) \leq \mathbb{E}_{x_0 \sim p_{data}}[-ELBO(x_0)] - H(p_{data}), \quad (1)$$

where ELBO is the Evidence Lower Bound and $H(p_{data})$ is the entropy of the data distribution.

Remark 1. (Equality Condition). The inequality in Lemma 1 becomes an equality when the variational posterior $q(x_{1:T}|x_0)$ equals the true posterior $p_{\text{DM}}(x_{1:T}|x_0)$ under the model. In standard DMs, this condition can be achieved at the global optimum when the reverse process has sufficient capacity (Luo 2022), yielding:

$$D_{\text{KL}}(p_{\text{data}}\|p_{\text{DM}}) = \mathbb{E}_{x_0\sim p_{\text{data}}}[-\text{ELBO}(x_0)] + H(p_{\text{data}}) \quad (2)$$

Since $H(p_{\text{data}})$ is independent of model parameters, minimizing $\mathbb{E}_{x_0\sim p_{\text{data}}}[-\text{ELBO}(x_0)]$ is equivalent to minimizing $D_{KL}(p_{\text{data}}||p_{\text{DM}})$. Lemma 1 is supported by recent works on DMs (Luo 2022; Nichol and Dhariwal 2021), which indicates that DMs indirectly minimize the KL divergence by maximizing the ELBO. KL divergence is asymmetric and severely penalizes the model if $p_{DM} = 0$ where $p_{data} > 0$.

Lemma 2. *(GAN optimization problem) For a fixed generator $\mathcal{G}$, the value function $V(\mathcal{D},\mathcal{G})$ satisfies the inequality:*

$$V(\mathcal{D},\mathcal{G}) \leq 2D_{JS}(p_{data}\|p_{GAN}) - 2\log 2, \quad (3)$$

with equality if and only if the discriminator is optimal, given by $\mathcal{D}^*(x) = p_{data}(x)/(p_{data}(x) + p_{GAN}(x))$

In contrast to DMs, GANs optimize the JS divergence through an adversarial game (Che et al. 2020; Goodfellow et al. 2020; Arjovsky, Chintala, and Bottou 2017). Unlike the KL divergence, the JS divergence is symmetric and remains bounded even when the generator completely ignores certain regions of the data space.

These lemmas lead to our first key theoretical result:

Theorem 1. *(Fundamental Difference in Optimization Objectives) Using the notations of Lemmas 1 and 2, GANs and DMs solve various optimization problems:*

1. *GANs:* $\min_{\mathcal{G}} V(\mathcal{D}^*,\mathcal{G}) \equiv \min D_{JS}(p_{data}\|p_{GAN})$, *where $\mathcal{D}^*$ is the optimal discriminator.*
2. *DMs:* $\min \mathbb{E}_{x\sim p_{data}}[-ELBO(x)] - H(p_{data}) \equiv \min D_{KL}(p_{data}\|p_{DM})$

Proof for Theorem 1 in Appendix. Theorem 1 indicates the fundamental difference between the optimization objectives of GANs and DMs.

From Divergence to Distribution Support and Artifacts

The choice of divergence profoundly impacts the learned distributions and the nature of artifacts inherent to the generation process.

Theorem 2. *(Disparity of Learned Distribution Support) Let p_{data} be the true data distribution with support $S_{data} = \{x : p_{data}(x) > 0\}$. Assume that the generator and score-matching networks have sufficient capacity:*

1. ***GANs**: A globally optimal solution for the GAN generator, which minimizes the JS divergence $D_{JS}(p_{data}\|p_{GAN})$, can exist even if the support of the learned distribution, S_{GAN}, is a strict subset of the data support ($S_{GAN} \subset S_{data}$, partial coverage).*
2. ***DMs**: A globally optimal solution for the DM, which minimizes the upper bound on the KL divergence $D_{KL}(p_{data}\|p_{DM})$, requires the support of the learned distribution, S_{DM}, to cover the support of the data distribution ($S_{data} \subseteq S_{DM}$, complete coverage).*

Proof in Appendix. Theorem 2 reveals why GANs and DMs exhibit fundamentally different coverage behaviors. For DMs, to achieve an optimal solution, the support of its learned distribution must cover the support of the real data distribution ($S_{\text{data}} \subseteq S_{\text{DM}}$). This is because the KL divergence $D_{\text{KL}}(p_{\text{data}}\|p_{\text{DM}})$ becomes infinite whenever $p_{\text{DM}} = 0$ for any x where $p_{\text{data}} > 0$. As a result, DMs must spread their capacity across the entire data manifold, resulting in diffusion blur, over-smoothing in low-density regions. Furthermore, the iterative denoising process of DMs relies on score matching. Imperfections in this estimation accumulate during the reverse diffusion steps, combined with the smoothing enforced by the KL objective, will leave unique, structured noise residuals that serve as a distinct architectural artifact (Wang et al. 2023; Ma et al. 2024). In contrast, due to JS divergence optimization, GANs can achieve optimality while ignoring low-probability regions of p_{data}. This leads to sharp samples but incomplete coverage where GANs concentrate their capacity on high-density modes, producing sharp samples but missing rare patterns. The boundary artifacts manifest as abrupt transitions at the edges of learned support, creating detectable discontinuities. These discontinuities manifest in the image space as characteristic boundary artifacts, such as structural inconsistencies or unnatural texture transitions (Zhang, Karaman, and Chang 2019).

Methodology

This section presents a semi-supervised **Tri**rchy **Detect**or (TriDetect) that simultaneously performs binary classification and discovers latent structures within synthetic images. From our theoretical analysis, by learning to recognize distinct architectural patterns that persist across specific generators within each family (i.e., GANs vs. DMs), it is possible to achieve cross-generator generalization capability.

Problem Formulation and Objectives

Given a dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$ where $x_i \in \mathbb{R}^{H\times W\times 3}$ represents an image and $y_i \in \{0, 1\}$ denotes its binary label (0:real, 1:fake), we compute logits $z_i = g_\theta(f(x_i)) \in \mathbb{R}^3$ using a fine-tuned vision encoder f and classifier g_θ, where we have 3 output dimensions (one real class and two fake clusters). We formulate a joint optimization problem with two objectives: (i) distinguish real from synthetic images, and (ii) automatically discover latent clusters within synthetic samples that correspond to different generative architectures. The architecture produces three-way logits that align with the theoretical distinction between GAN and DM architectures established in our analysis.

Balanced Clustering via Optimal Transport

A critical challenge in unsupervised deep clustering is the tendency toward trivial solutions where all samples collapse into a single cluster (Zhou et al. 2024). This problem is particularly serious in our setting, where we seek to discover subtle architectural patterns within synthetic images generated by different generators. To address this, we formulate cluster assignment as an optimal transport problem that enforces balanced distribution across clusters.

For a batch B of synthetic samples with logits $Z_{\text{fake}} \in \mathbb{R}^{B\times K}$ where $K = 2$ is the number of fake clusters, we seek an assignment matrix $Q \in \mathbb{R}^{B\times K}$ that maximizes the similarity between samples and clusters while maintaining balanced assignments:

$$\max_{Q\in\mathcal{Q}} \operatorname{Tr}(Q^T Z_{\text{fake}}) + \epsilon H(Q), \tag{4}$$

where $H(Q) = -\sum_{ij} Q_{ij} \log Q_{ij}$ is the entropy function, ϵ controls assignment smoothness, and $\mathcal{Q}$ is the transportation polytope (Asano, Rupprecht, and Vedaldi 2019):

$$\mathcal{Q} = \left\{ Q \in \mathbb{R}_+^{B\times K} \mid Q\mathbf{1}_K = \mathbf{1}_B,\ Q^T\mathbf{1}_B = \frac{B}{K}\mathbf{1}_K \right\} \tag{5}$$

These constraints ensure that each sample is fully assigned across clusters (rows sum to 1) and each cluster receives exactly B/K samples (columns sum to B/K), preventing collapse.

Sinkhorn-Knopp Algorithm for Online Clustering. We utilize the Sinkhorn-Knopp algorithm (Cuturi 2013) to solve the optimal transport problem. Starting from the initialization $Q^{(0)} = \exp(Z_{\text{fake}}/\epsilon)$, for $t = 1, \ldots, T$ iterations, we perform:

$$Q^{(t)} = \mathcal{R}(Q^{(t-1)}); \quad Q^{(t)} = \mathcal{C}(Q^{(t)}), \tag{6}$$

where $\mathcal{R}(Q)_{ij} = \frac{Q_{ij}}{\sum_k Q_{ik}}$ normalizes rows and $\mathcal{C}(Q)_{ij} = \frac{K\cdot Q_{ij}}{\sum_k Q_{kj}}$ normalizes columns with appropriate scaling to maintain the balance constraints. Note that Eq. 6 is operated on mini-batches rather than the entire dataset, enabling online learning that scales to arbitrarily large datasets.

Cross-view Consistency for Generalized Learning

To learn discriminative and generalized clusters that capture architectural patterns rather than image statistics, we propose a cross-view consistency mechanism that encourages consistent predictions and clustering stability. This mechanism contains the *swapped prediction strategy* and *consistency regularization*.

Swapped Prediction for Cluster Learning. Let (x, x') denote different views, generated by random transformations on each image, with corresponding logits (z, z'). For fake samples, we extract the fake cluster logits z_{fake} and z'_{fake}. Inspired by the work (Caron et al. 2020), the core idea here is to use the cluster assignment from one view to supervise the prediction from the other view. Specifically, after computing balanced assignments using the Sinkhorn-Knopp algorithm $Q = \text{Sinkhorn}(z_{\text{fake}})$ and $Q' = \text{Sinkhorn}(z'_{\text{fake}})$ for fake samples in each view, the assignment loss is computed as follows:

$$\mathcal{L}_{\text{assignment}} = -\frac{1}{2|\mathcal{F}|}\sum_{i\in\mathcal{F}}\left(\sum_{k=1}^{K} Q'_{ik}\log p_{ik} + \sum_{k=1}^{K} Q_{ik}\log p'_{ik}\right), \tag{7}$$

Algorithm 1: TriDetect Training Algorithm

Require: Dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$, feature encoder f, classifier g_θ, learning rate η, hyperparameters $\beta, \omega_1, \omega_2$, $K = 2$ (fake clusters)
Ensure: Trained encoder f and classifier g_θ
Initialize classifier parameters θ
for $t = 1$ to T epochs **do**
  for each batch $\mathcal{B} \subset \mathcal{D}$ **do**
    $\mathcal{B}' \leftarrow \text{Augment}(\mathcal{B})$ ▷ Create augmented views
    $Z, Z' \leftarrow g_\theta(f(\mathcal{B})), g_\theta(f(\mathcal{B}'))$ ▷ Compute triarchy logits
    $\mathcal{F} \leftarrow \{i : y_i = 1\}$ ▷ Extract fake indices
    $Z_{\text{fake}}, Z'_{\text{fake}} \leftarrow Z[\mathcal{F}, 1:K], Z'[\mathcal{F}, 1:K]$ ▷ Extract fake logits
    $Q_{\text{fake}}, Q'_{\text{fake}} \leftarrow \text{Sinkhorn}(Z_{\text{fake}}), \text{Sinkhorn}(Z'_{\text{fake}})$ ▷ Balanced assignment
    Compute $\mathcal{L}_{\text{binary}}$ (Eq.10), $\mathcal{L}_{\text{assignment}}$ (Eq.7), $\mathcal{L}_{\text{consistency}}$ (Eq.8)
    $\mathcal{L}_{\text{cluster}} \leftarrow \omega_1\mathcal{L}_{\text{assignment}} + \omega_2\mathcal{L}_{\text{consistency}}$
    $\mathcal{L}_{\text{total}} \leftarrow \beta\mathcal{L}_{\text{binary}} + (1-\beta)\mathcal{L}_{\text{cluster}}$
    $\theta \leftarrow \theta - \eta\nabla_\theta\mathcal{L}_{\text{total}}$ ▷ Update parameters
  end for
end for
return f, g_θ

where $\mathcal{F}$ denotes the set of fake samples in the batch and $p_{ik} = \frac{\exp(z_{ik}/\tau)}{\sum_j \exp(z_{ij}/\tau)}$ with temperature τ.

Consistency Regularization for Stable Assignments. The consistency regularization ensures that the Sinkhorn-Knopp assignments themselves remain stable across views, preventing oscillation during training and encouraging convergence to meaningful clusters. Particularly, this additional regularization directly constrains the similarity of assignments themselves:

$$\mathcal{L}_{\text{consistency}} = \frac{1}{|\mathcal{F}|}\sum_{i\in\mathcal{F}} \|Q_i - Q'_i\|_2^2 \tag{8}$$

Loss Function

The overall loss function for training TriDetect contains the binary classification loss and the clustering loss:

$$\mathcal{L}_{\text{total}} = \beta\mathcal{L}_{\text{binary}} + (1-\beta)\mathcal{L}_{\text{cluster}}, \tag{9}$$

where $\mathcal{L}_{\text{cluster}} = \omega_1\mathcal{L}_{\text{assignment}} + \omega_2\mathcal{L}_{\text{consistency}}$. β denotes the hyperparameter that balances the importance of the binary and clustering losses while ω_1 and ω_2 represent assignment and consistency weights. $\mathcal{L}_{\text{binary}}$ is the cross-entropy loss, computed as:

$$\mathcal{L}_{\text{binary}} = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log p(\text{fake})_i + (1-y_i)\log p(\text{real})_i], \tag{10}$$

where $p_i = \text{softmax}(z_i) \in \mathbb{R}^3$, with $p(\text{real})_i = p_{i,0}$ and $p(\text{fake})_i = p_{i,1} + p_{i,2}$. We provide the training pseudocode in Alg. 1 and the Sinkhorn-Knopp algorithm in Appendix.

Method	BigGAN	SD v1.4	ADM	GLIDE	MidJourney	VQDM	Wukong	Avg
CNNSpot	**1.0000**	<u>0.9999</u>	0.8139	0.9591	0.7969	0.8028	0.9988	0.9102
FreDect	**1.0000**	0.9970	0.7485	0.7713	0.7458	0.8086	0.9894	0.8658
Fusing	<u>0.9999</u>	<u>0.9999</u>	0.7261	0.5664	0.6671	0.7586	0.9988	0.8167
LGrad	0.6530	0.5623	0.5237	0.6103	0.5411	0.5244	0.6043	0.5742
LNP	0.9812	0.9809	0.6090	0.6089	0.6596	0.6825	0.9689	0.7844
CORE	**1.0000**	0.9998	0.8198	0.9950	0.8011	0.8192	0.9971	0.9189
SPSL	0.9998	0.9970	0.5946	0.8283	0.7545	0.7742	0.9918	0.8486
UIA-ViT	**1.0000**	0.9993	0.7314	0.9519	0.8024	0.7950	0.9946	0.8964
DIRE	0.9996	0.9997	0.8257	0.9843	0.8298	0.8303	0.9965	0.9237
UnivFD	0.9988	0.9955	<u>0.9265</u>	0.9920	0.9262	0.9859	0.9678	0.9704
AIDE	0.9998	0.9900	0.7736	0.9299	0.8680	0.8739	0.9710	0.9152
NPR	<u>0.9999</u>	0.9979	0.9385	<u>0.9956</u>	0.9430	0.9553	0.9861	0.9695
Effort	**1.0000**	**1.0000**	0.9052	0.9893	**0.9795**	<u>0.9969</u>	<u>0.9998</u>	<u>0.9815</u>
TriDetect	**1.0000**	**1.0000**	**0.9609**	**0.9993**	<u>0.9581</u>	**0.9992**	**0.9999**	**0.9882**

Table 1: Comparison on the GenImage (Zhu et al. 2023) Dataset in terms of AUC Performance. The best result and the second-best result are marked in bold and underline, respectively.

Method	CycleGAN	StyleGAN	StyleGAN2	ProGAN	GauGAN	StarGAN	BigGAN	ADM	Wukong	GLide	SD_XL	VQDM	MidJourney	SD v1.4	SD v1.5	DALLE2	Avg
CNNSpot	0.3491	0.5298	0.4787	0.5039	0.4260	0.5538	0.3811	0.8138	0.9988	0.9591	0.8252	0.8028	0.7969	0.9999	0.9995	0.9748	0.7121
FreDect	0.3956	0.6697	0.6815	0.7172	0.5877	0.7211	0.8346	0.7485	0.9894	0.7713	0.8743	0.8086	0.7458	0.9970	0.9956	0.8224	0.7725
Fusing	0.4071	0.5180	0.5274	0.4797	0.4815	0.5072	0.3888	0.7261	0.9988	0.9429	0.7468	0.7586	0.6673	<u>0.9999</u>	0.9998	0.9448	0.6934
LGrad	0.4833	0.4783	0.4181	0.4993	0.4508	0.4925	0.4972	0.5317	0.6009	0.6758	0.4593	0.5100	0.5320	0.5639	0.5617	0.6041	0.5224
LNP	0.3984	0.5118	0.4842	0.4993	0.4884	0.4283	0.4246	0.6089	0.9689	0.8474	0.7127	0.6823	0.6596	0.9809	0.9800	0.8382	0.6571
CORE	0.6153	0.6115	0.5762	0.5617	0.5310	0.9731	0.5574	0.8198	<u>0.9971</u>	**0.9950**	0.7561	0.8193	0.8011	0.9998	0.9997	0.9498	0.7852
SPSL	0.5817	0.5384	0.5305	0.5252	0.5209	0.5228	0.6458	0.5946	0.9918	0.8283	0.8257	0.7742	0.7545	0.9971	0.9953	0.8301	0.7161
UIA-ViT	0.5291	0.7081	0.7545	0.5109	0.4934	0.5252	0.6471	0.7314	0.9946	0.9519	0.9358	0.7950	0.8024	0.9993	0.9988	0.8187	0.7623
DIRE	0.6335	0.5967	0.6117	0.7896	0.4016	0.9935	0.5755	0.8257	0.9965	0.9843	0.8928	0.8303	0.8298	0.9997	<u>0.9995</u>	0.9627	0.8077
UnivFD	0.9664	0.7297	0.7304	0.9407	0.9828	0.9185	0.9712	0.9265	0.9678	<u>0.9920</u>	0.9610	0.9859	0.9262	0.9955	0.9953	0.9627	0.9345
AIDE	0.6142	0.7377	0.7249	0.7979	0.8443	0.7025	0.8084	0.7736	0.9710	0.9299	0.9309	0.8739	0.8680	0.9900	0.9900	0.8859	0.8402
NPR	0.9288	0.8851	<u>0.9403</u>	0.9653	0.5391	**0.9998**	0.7939	**0.9385**	0.9861	0.9956	0.9728	0.9553	0.9430	0.9979	0.9967	<u>0.9935</u>	0.9270
Effort	<u>0.9894</u>	**0.9431**	0.8961	<u>0.9973</u>	**0.9997**	0.9923	<u>0.9991</u>	0.9052	<u>0.9998</u>	0.9893	**0.9980**	<u>0.9969</u>	**0.9795**	**1.0000**	**1.0000**	0.9666	<u>0.9783</u>
TriDetect	**1.0000**	<u>0.9419</u>	**0.9422**	**0.9995**	<u>0.9978</u>	**1.0000**	**0.9992**	**0.9609**	**0.9999**	**0.9993**	<u>0.9948</u>	**0.9992**	<u>0.9581</u>	**1.0000**	**1.0000**	**0.9971**	**0.9869**

Table 2: Comparison on the AIGCDetectBenchmark (Zhong et al. 2023) Dataset in terms of AUC Performance. The best result and the second-best result are marked in bold and underline, respectively.

Experiments

Experimental Setup

Baselines. We compare our method against 13 competitive detectors, including CNNSpot (Wang et al. 2020b), LNP (Liu et al. 2022), FreDect (Frank et al. 2020), CORE (Ni et al. 2022), SPSL (Liu et al. 2021), UIA-ViT (Zhuang et al. 2022), Fusing (Ju et al. 2022), LGrad (Tan et al. 2023), DIRE (Wang et al. 2023), UnivFD (Ojha, Li, and Lee 2023), AIDE (Shilin et al. 2025), NPR (Tan et al. 2025), and Effort (Yan et al. 2025b).

Datasets. We evaluate our method on 2 standard benchmarks (GenImage (Zhu et al. 2023), AIGCDetectBenchmark (Zhong et al. 2023)) and 3 in-the-wild datasets (WildFake (Hong et al. 2025), Chameleon (Shilin et al. 2025), DF40 (Yan et al. 2024)). In our experiments, we follow the training protocol introduced by (Shilin et al. 2025) by utilizing BigGAN and SDv1.4 subsets of GenImage.

Metrics. In this work, we report results across 4 commonly used metrics: accuracy (ACC), area under the ROC curve (AUC), equal error rate (EER), and average precision (AP). We primarily report AUC and ACC in the main paper; other metric results are provided in Appendix.

Implementation Details. We use CLIP ViT-L/14 (Radford et al. 2021) vision encoder f, which is fine-tuned with Low-Rank Adaptation (LoRA) (Hu et al. 2022). For baseline implementation, we use two benchmarks, DeepfakeBench (Yan et al. 2023) and AIGCDetectBenchmark (Zhong et al. 2023). Detailed experimental procedures, hyperparameters, and configurations are provided in Appendix.

Experimental Results

Standard Benchmarks. On GenImage, Table 1 shows that TriDetect achieves the highest average AUC of 0.9882, outperforming SoTA methods, including AIDE (0.9152), NPR (0.9695), and Effort (0.9815). The results on AIGCDe-

Method	DALL-E	DDIM	DDPM	VQDM	BigGAN	StarGAN	StyleGAN	DF-GAN	GALIP	GigaGAN	Avg
CNNSpot	0.5961	0.2929	0.2811	0.2882	0.8456	0.3350	0.3453	0.3418	0.3325	0.3409	0.3999
FreDect	0.4078	0.2817	0.2952	0.4643	0.8415	0.3689	0.3365	0.4415	0.3478	0.8148	0.4600
Fusing	0.4706	0.2376	0.2427	0.3305	**0.9361**	0.3425	0.3361	0.3518	0.3587	0.3446	0.3951
LGrad	0.6044	0.5794	0.5473	0.5656	0.6126	0.5517	0.5129	0.5732	0.4796	0.5758	0.5602
LNP	0.5105	0.2681	0.2715	0.3580	0.7179	0.3741	0.3696	0.3697	0.3649	0.3708	0.3975
CORE	0.3829	0.2552	0.2762	0.3668	0.7115	0.3359	0.3353	0.3974	0.3365	0.3400	0.3738
SPSL	0.5614	0.2282	0.2535	0.3082	0.8692	0.3324	0.3395	0.3618	0.3338	0.3334	0.3921
UIA-ViT	0.5718	0.2699	0.2636	0.3475	0.7542	0.3402	0.3635	0.3806	0.3430	0.3592	0.3993
DIRE	0.5725	0.3222	0.3130	0.4370	0.6130	0.3885	0.3465	0.3941	0.3724	0.3755	0.4135
UnivFD	0.5492	**0.8244**	**0.7536**	0.7974	0.7962	0.6213	0.5734	0.9141	0.8572	0.7690	0.7456
AIDE	0.5806	0.3562	0.3405	0.4757	0.7420	0.3948	0.4331	0.6638	0.4215	0.3647	0.4773
NPR	0.5122	0.6892	0.5305	0.7313	0.7102	0.5510	0.3583	0.7753	0.5221	0.6876	0.6068
Effort	**0.6741**	0.6436	0.4800	0.8467	0.8992	0.7971	0.6027	0.9767	0.8127	0.7891	0.7522
TriDetect	0.6021	0.7359	0.7207	**0.9301**	0.8042	**0.9998**	**0.6300**	**0.9825**	**0.9017**	**0.9468**	**0.8254**

Table 3: Comparison on the WildFake (Hong et al. 2025) Dataset in terms of ACC Performance. The best result and the second-best result are marked in bold and underline, respectively.

Method	CollabDiff	MidJourney	DeepFaceLab	StarGAN	StarGAN2	StyleCLIP	WhichisReal	Avg
CNNSpot	0.5635	0.5376	0.3808	0.4851	0.5040	0.3041	0.5012	0.4681
FreDect	0.8720	0.4614	0.3322	0.4982	0.5158	0.4103	0.5327	0.5175
Fusing	0.6150	0.5296	0.4830	0.4554	0.4942	0.2908	0.4978	0.4808
LGrad	0.5045	0.5019	0.5374	0.5088	0.4670	0.6124	0.5107	0.5204
LNP	0.5120	0.4997	0.3808	0.4962	0.5108	0.3428	0.5292	0.4674
CORE	0.5025	0.5328	0.5708	0.5000	0.5000	0.2612	0.4998	0.4810
SPSL	0.5140	0.5280	0.3337	0.4955	0.4992	0.2884	0.4988	0.4511
UIA-ViT	0.5380	0.4374	0.3609	0.5033	0.5008	0.3930	0.4698	0.4576
DIRE	0.5070	0.5424	0.3275	0.5000	0.5000	0.2652	0.4658	0.4440
UnivFD	0.5955	0.7304	0.5369	0.6361	0.5516	0.8353	0.6727	0.6512
NPR	0.4995	**0.8711**	0.3228	0.4970	0.5000	0.2610	0.6442	0.5136
AIDE	0.5525	0.7038	0.4039	0.6331	0.5055	0.4066	0.5197	0.5322
Effort	**0.9735**	0.8370	0.4500	**0.9287**	0.7345	0.9710	**0.8291**	0.8177
TriDetect	0.9690	0.8629	**0.6402**	0.8005	**0.8208**	**0.9825**	0.8241	**0.8429**

Table 4: Comparison on the DF40 (Yan et al. 2024) Dataset in terms of ACC Performance. The best result and the second-best result are marked in bold and underline, respectively.

tectBenchmark (Table 2) also demonstrate the generalization capability of TriDetect to 16 unseen generators, yielding an improvement over DIRE (22.18%), AIDE (17.4%), UnivFD (5.6%), NPR (6.5%), and Effort (0.9%).

In-the-wild Datasets. Tables 3-5 show that our method outperforms existing detectors across all evaluation metrics on in-the-wild datasets:

- **Robust to degradation techniques:** TriDetect exhibits consistent performance when evaluated on WildFake in which degradation techniques (e.g., downsampling, cropping) are applied on testing sets. Particularly, TriDetect achieves an average ACC of 0.8254, a 8.86% improvement compared to Effort. Results demonstrate the better robustness of TriDetect than other methods.
- **Generalize to facial editing methods:** On DF40 dataset, TriDetect achieves an average ACC of 0.8429, an improvement of 3.08% and 29.43% compared to Effort and UnivFD, respectively.
- **Achieve lowest EER:** On the challenging Chameleon dataset (Table 5), which is designed to deceive both humans and AI models, TriDetect attains the lowest EER of 0.1843, a 31.74% reduction compared to Effort. This demonstrates that our approach can achieve a superior balance between false positive and false negative rates.

Ablation Study

Module Ablation of Components in TriDetect

Table 6 provides evidence for the contribution of each loss component in our TriDetect. With $\mathcal{L}_{\text{assignment}}$ and $\mathcal{L}_{\text{consistency}}$, TriDetect can yield a substantial improvement in AUC to 0.8935 and ACC to 0.6258. These results highlight the significant contribution of the clustering loss to TriDetect's performance.

Method	AUC	ACC	EER	AP
CNNSpot	0.6331	0.5838	0.4062	0.5800
FreDect	0.7545	0.6304	0.3143	0.6864
Fusing	0.6020	0.5745	0.4138	0.5350
LGrad	0.5342	0.5155	0.4809	0.4735
LNP	0.5798	0.5790	0.4420	0.5039
CORE	0.5264	0.5700	0.4772	0.4556
SPSL	0.6135	0.5748	0.4147	0.5335
UIA-ViT	0.7042	0.5987	0.3538	0.6115
DIRE	0.5488	0.5768	0.4565	0.4909
UnivFD	0.8030	<u>0.6598</u>	0.2670	0.7506
AIDE	0.7970	0.6256	0.2747	0.7345
NPR	0.5865	0.5805	0.4381	0.4943
Effort	<u>0.8371</u>	0.6459	<u>0.2428</u>	<u>0.7692</u>
TriDetect	**0.8935**	**0.6788**	**0.1843**	**0.8742**

Table 5: Comparison on the Chameleon (Shilin et al. 2025) Dataset in terms of AUC, ACC, EER, and AP Performance. The best result and the second-best result are marked in bold and underline, respectively.

$\mathcal{L}_{\text{binary}}$	$\mathcal{L}_{\text{assignment}}$	$\mathcal{L}_{\text{consistency}}$	AUC	ACC	EER	AP
✓			0.8033	0.6335	0.2713	0.7503
✓	✓		0.8560	0.5817	0.2195	0.8394
✓	✓	✓	**0.8935**	**0.6258**	**0.1843**	**0.8742**

Table 6: Ablation study on the binary classification loss ($\mathcal{L}_{\text{binary}}$), assignment loss ($\mathcal{L}_{\text{assignment}}$), and consistency loss ($\mathcal{L}_{\text{consistency}}$). Results are performed on Chameleon dataset.

Different Values of β

The ablation study on β (Table 7) reveals a crucial insight: the optimal balance between binary classification and clustering objectives occurs at $\beta = 0.7$, where the model achieves peak performance across all metrics.

Analysis on Learned Embedding Spaces

Figure 2 provides visual evidence that TriDetect successfully discovers meaningful latent structure within synthetic images. The alignment between unsupervised discovery (left figure) and ground truth (right figure) validates our theoretical analysis that different generative architectures leave distinguishable artifacts in feature space. The middle figure further confirms that this clustering capability enhances rather than compromises binary detection performance.

We also provide other ablation studies in Appendix, including comparison with attribution baselines and ablation study on different numbers of fake clusters. Results show that TriDetect outperforms the selected attribution baselines, and $K = 2$ is the optimal number of fake clusters.

Related Work

Generalized AI-generated Image Detection

AI-generated images exhibit artifacts throughout the entire scene. The detection landscape encompasses three primary approaches: data augmentations for improved generalization (Wang et al. 2020b), artifact analysis techniques identifying generation-specific patterns such as GAN checkerboard artifacts (Zhang, Karaman, and Chang 2019; Tan et al. 2025) and diffusion reconstruction errors (Luo et al. 2024; Wang et al. 2023), and vision-language models leveraging CLIP representations to capture subtle synthetic patterns (Ojha, Li, and Lee 2023; Yan et al. 2025a,b). Recent advances include the exploitation of upsampling traces common to both GANs and DMs (Tan et al. 2025), while adaptation strategies employ meta-learning (Chen et al. 2022), and incremental learning (Pan et al. 2023) to address evolving generators.

Compared to these works, we provide the first theoretical explanation for distinct artifacts left by GANs and DMs. These theoretical results motivate us to design TriDetect that simultaneously performs binary detection and discovers architectural patterns without explicit supervision.

β value	AUC	ACC	EER	AP
$\beta = 0.3$	0.8423	0.6069	0.2348	0.8147
$\beta = 0.5$	0.8451	0.5994	0.2338	0.8058
$\beta = 0.7$	**0.8935**	**0.6258**	**0.1843**	**0.8742**
$\beta = 0.9$	0.8774	0.6043	0.2033	0.8713

Table 7: Ablation studies on β value in our method. Results are performed on Chameleon dataset.

Deep Clustering

Deep clustering has evolved from multi-stage pipelines separating representation learning and clustering (Tao, Takagi, and Nakata 2021; Zhang et al. 2021; Peng et al. 2016) to end-to-end approaches jointly optimizing both objectives (Ji et al. 2021; Shen et al. 2021; Caron et al. 2020; Qian 2023). Notable advances include utilizing the Sinkhorn-Knopp algorithm for optimal cluster assignment through prototype matching (Caron et al. 2020) and designing hardness-aware objectives for training stability (Qian 2023).

While our method also employs Sinkhorn-Knopp, we uniquely combine it with semi-supervised learning to leverage real image supervision while discovering architectural patterns in synthetic images. We also propose a consistency regularization loss to ensure stable cluster assignment.

Conclusion

This work provides the first theoretical analysis to demonstrate that latent representations of synthetic images generated by GANs and DMs form significantly different submanifolds in feature space. This analysis motivates TriDetect, a semi-supervised detection method that enhances binary classification with architecture-aware clustering. Through discovering latent architectural patterns without explicit labels, TriDetect can learn more generalized representations that help to improve the generalization capabilities across unseen generators within the same architecture families. This work establishes a new paradigm for AI-generated image detection, shifting from pattern recognition to architectural understanding, providing a foundation for addressing future generators.

Acknowledgments

This work was funded by Taighde Éireann – Research Ireland through the Research Ireland Centre for Research Training in Machine Learning (18/CRT/6183).

References

Arjovsky, M.; Chintala, S.; and Bottou, L. 2017. Wasserstein generative adversarial networks. In *International conference on machine learning*, 214–223. PMLR.

Asano, Y. M.; Rupprecht, C.; and Vedaldi, A. 2019. Self-labelling via simultaneous clustering and representation learning. *arXiv preprint arXiv:1911.05371*.

Caron, M.; Misra, I.; Mairal, J.; Goyal, P.; Bojanowski, P.; and Joulin, A. 2020. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in neural information processing systems*, 33: 9912–9924.

Che, T.; Zhang, R.; Sohl-Dickstein, J.; Larochelle, H.; Paull, L.; Cao, Y.; and Bengio, Y. 2020. Your GAN is secretly an energy-based model and you should use discriminator driven latent sampling. *Advances in Neural Information Processing Systems*, 33: 12275–12287.

Chen, L.; Zhang, Y.; Song, Y.; Wang, J.; and Liu, L. 2022. Ost: Improving generalization of deepfake detection via one-shot test-time training. *Advances in Neural Information Processing Systems*, 35: 24597–24610.

Cuturi, M. 2013. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26.

Deng, J.; Lin, C.; Zhao, Z.; Liu, S.; Peng, Z.; Wang, Q.; and Shen, C. 2024. A survey of defenses against ai-generated visual media: Detection, disruption, and authentication. *ACM Computing Surveys*.

Dzanic, T.; Shah, K.; and Witherden, F. 2020. Fourier spectrum discrepancies in deep network generated images. *Advances in neural information processing systems*, 33: 3022–3032.

Fefferman, C.; Mitter, S.; and Narayanan, H. 2016. Testing the manifold hypothesis. *Journal of the American Mathematical Society*, 29(4): 983–1049.

Frank, J.; Eisenhofer, T.; Schönherr, L.; Fischer, A.; Kolossa, D.; and Holz, T. 2020. Leveraging frequency analysis for deep fake image recognition. In *International conference on machine learning*, 3247–3258. PMLR.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2020. Generative adversarial networks. *Communications of the ACM*, 63(11): 139–144.

Hong, Y.; Feng, J.; Chen, H.; Lan, J.; Zhu, H.; Wang, W.; and Zhang, J. 2025. Wildfake: A large-scale and hierarchical dataset for ai-generated images detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 3500–3508.

Hu, E. J.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; Chen, W.; et al. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations*.

Huang, Z.; Chan, K. C.; Jiang, Y.; and Liu, Z. 2023. Collaborative diffusion for multi-modal face generation and editing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 6080–6090.

Ji, Q.; Sun, Y.; Hu, Y.; and Yin, B. 2021. Variational deep embedding clustering by augmented mutual information maximization. In *2020 25th International Conference on Pattern Recognition (ICPR)*, 2196–2202. IEEE.

Ju, Y.; Jia, S.; Ke, L.; Xue, H.; Nagano, K.; and Lyu, S. 2022. Fusing global and local features for generalized ai-synthesized image detection. In *2022 IEEE International Conference on Image Processing (ICIP)*, 3465–3469. IEEE.

Liu, B.; Yang, F.; Bi, X.; Xiao, B.; Li, W.; and Gao, X. 2022. Detecting generated images by real images. In *European Conference on Computer Vision*, 95–110. Springer.

Liu, H.; Li, X.; Zhou, W.; Chen, Y.; He, Y.; Xue, H.; Zhang, W.; and Yu, N. 2021. Spatial-phase shallow learning: rethinking face forgery detection in frequency domain. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 772–781.

Loaiza-Ganem, G.; Ross, B. L.; Hosseinzadeh, R.; Caterini, A. L.; and Cresswell, J. C. 2024. Deep Generative Models through the Lens of the Manifold Hypothesis: A Survey and New Connections. *Transactions on Machine Learning Research*.

Lu, Z.; Huang, D.; Bai, L.; Qu, J.; Wu, C.; Liu, X.; and Ouyang, W. 2023. Seeing is not always believing: Benchmarking human and model perception of ai-generated images. *Advances in neural information processing systems*, 36: 25435–25447.

Luo, C. 2022. Understanding diffusion models: A unified perspective. *arXiv preprint arXiv:2208.11970*.

Luo, Y.; Du, J.; Yan, K.; and Ding, S. 2024. LaRE^ 2: Latent reconstruction error based method for diffusion-generated image detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 17006–17015.

Ma, R.; Duan, J.; Kong, F.; Shi, X.; and Xu, K. 2024. Exposing the Fake: Effective Diffusion-Generated Images Detection. In *The Second Workshop on New Frontiers in Adversarial Machine Learning*.

Nguyen-Le, H.-H.; Tran, V.-T.; Nguyen, D.-T.; and Le-Khac, N.-A. 2025. Deepfake Detection Across Image, Video, and Audio: A Comprehensive Survey with Empirical Evaluation of Generalization and Robustness.

Ni, Y.; Meng, D.; Yu, C.; Quan, C.; Ren, D.; and Zhao, Y. 2022. Core: Consistent representation learning for face forgery detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12–21.

Nichol, A. Q.; and Dhariwal, P. 2021. Improved denoising diffusion probabilistic models. In *International conference on machine learning*, 8162–8171. PMLR.

Ojha, U.; Li, Y.; and Lee, Y. J. 2023. Towards universal fake image detectors that generalize across generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 24480–24489.

Pan, K.; Yin, Y.; Wei, Y.; Lin, F.; Ba, Z.; Liu, Z.; Wang, Z.; Cavallaro, L.; and Ren, K. 2023. Dfil: Deepfake incremental learning by exploiting domain-invariant forgery clues. In *Proceedings of the 31st ACM International Conference on Multimedia*, 8035–8046.

Peng, X.; Xiao, S.; Feng, J.; Yau, W.-Y.; and Yi, Z. 2016. Deep subspace clustering with sparsity prior. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, 1925–1931.

Qian, Q. 2023. Stable cluster discrimination for deep clustering. In *Proceedings of the IEEE/CVF international conference on computer vision*, 16645–16654.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, 8748–8763. PmLR.

Shen, Y.; Shen, Z.; Wang, M.; Qin, J.; Torr, P.; and Shao, L. 2021. You never cluster alone. *Advances in Neural Information Processing Systems*, 34: 27734–27746.

Shilin, Y.; Ouxiang, L.; Jiayin, C.; Yanbin, H.; Xiaolong, J.; Yao, H.; and Weidi, X. 2025. A Sanity Check for AI-generated Image Detection. In *The Thirteenth International Conference on Learning Representations*.

Tan, C.; Zhao, Y.; Wei, S.; Gu, G.; Liu, P.; and Wei, Y. 2025. Rethinking the up-sampling operations in cnn-based generative network for generalizable deepfake detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 28130–28139.

Tan, C.; Zhao, Y.; Wei, S.; Gu, G.; and Wei, Y. 2023. Learning on gradients: Generalized artifacts representation for gan-generated images detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12105–12114.

Tao, Y.; Takagi, K.; and Nakata, K. 2021. Clustering-friendly Representation Learning via Instance Discrimination and Feature Decorrelation. In *International Conference on Learning Representations*.

Wang, S.-Y.; Wang, O.; Zhang, R.; Owens, A.; and Efros, A. A. 2020a. CNN-Generated Images Are Surprisingly Easy to Spot... for Now. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Wang, S.-Y.; Wang, O.; Zhang, R.; Owens, A.; and Efros, A. A. 2020b. CNN-generated images are surprisingly easy to spot... for now. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 8695–8704.

Wang, Z.; Bao, J.; Zhou, W.; Wang, W.; Hu, H.; Chen, H.; and Li, H. 2023. Dire for diffusion-generated image detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 22445–22455.

WhichFaceisReal. 2019. Which Face Is Real? — whichfaceisreal.com. https://www.whichfaceisreal.com/. [Accessed 23-07-2025].

Yan, S.; Li, O.; Cai, J.; Hao, Y.; Jiang, X.; Hu, Y.; and Xie, W. 2025a. A Sanity Check for AI-generated Image Detection. In *The Thirteenth International Conference on Learning Representations*.

Yan, Z.; Wang, J.; Jin, P.; Zhang, K.-Y.; Liu, C.; Chen, S.; Yao, T.; Ding, S.; Wu, B.; and Yuan, L. 2025b. Orthogonal Subspace Decomposition for Generalizable AI-Generated Image Detection. *Forty-second International Conference on Machine Learning*.

Yan, Z.; Yao, T.; Chen, S.; Zhao, Y.; Fu, X.; Zhu, J.; Luo, D.; Wang, C.; Ding, S.; Wu, Y.; et al. 2024. Df40: Toward next-generation deepfake detection. *Advances in Neural Information Processing Systems*, 37: 29387–29434.

Yan, Z.; Zhang, Y.; Yuan, X.; Lyu, S.; and Wu, B. 2023. DeepfakeBench: A Comprehensive Benchmark of Deepfake Detection. *Advances in Neural Information Processing Systems*, 36: 4534–4565.

Zhang, S.; You, C.; Vidal, R.; and Li, C.-G. 2021. Learning a self-expressive network for subspace clustering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12393–12403.

Zhang, X.; Karaman, S.; and Chang, S.-F. 2019. Detecting and simulating artifacts in gan fake images. In *2019 IEEE international workshop on information forensics and security (WIFS)*, 1–6. IEEE.

Zhong, N.; Xu, Y.; Li, S.; Qian, Z.; and Zhang, X. 2023. Patchcraft: Exploring texture patch for efficient ai-generated image detection. *arXiv preprint arXiv:2311.12397*.

Zhou, S.; Xu, H.; Zheng, Z.; Chen, J.; Li, Z.; Bu, J.; Wu, J.; Wang, X.; Zhu, W.; and Ester, M. 2024. A comprehensive survey on deep clustering: Taxonomy, challenges, and future directions. *ACM Computing Surveys*, 57(3): 1–38.

Zhu, M.; Chen, H.; Yan, Q.; Huang, X.; Lin, G.; Li, W.; Tu, Z.; Hu, H.; Hu, J.; and Wang, Y. 2023. Genimage: A million-scale benchmark for detecting ai-generated image. *Advances in Neural Information Processing Systems*, 36: 77771–77782.

Zhuang, W.; Chu, Q.; Tan, Z.; Liu, Q.; Yuan, H.; Miao, C.; Luo, Z.; and Yu, N. 2022. UIA-ViT: Unsupervised inconsistency-aware method based on vision transformer for face forgery detection. In *European conference on computer vision*, 391–407. Springer.