

FIND: A Simple Yet Effective Baseline for Diffusion-Generated Image Detection

Jie Li¹, Yingying Feng², Chi Xie³, Jie Hu⁴, Lei Tan^{4*}, Jiayi Ji^{1,4}

¹Xiamen University

²Northeastern University

³Tongji University

⁴National University of Singapore

lijie.32@outlook.com, lei.tan@nus.edu.sg

Abstract

The remarkable realism of images generated by diffusion models poses critical detection challenges. Current methods utilize reconstruction error as a discriminative feature, exploiting the observation that real images exhibit higher reconstruction errors when processed through diffusion models. However, these approaches require costly reconstruction computations and depend on specific diffusion models, making their performance highly model-dependent. We identify a fundamental difference: real images are more difficult to fit with Gaussian distributions compared to synthetic ones. In this paper, we propose Forgery Identification via Noise Disturbance (FIND), a novel method that requires only a simple binary classifier. It eliminates reconstruction by directly targeting the core distributional difference between real and synthetic images. Our key operation is to add Gaussian noise to real images during training and label these noisy versions as synthetic. This step allows the classifier to focus on the statistical patterns that distinguish real from synthetic images. We theoretically prove that the noise-augmented real images resemble diffusion-generated images in their ease of Gaussian fitting. Furthermore, simply by adding noise, they still retain visual similarity to the original images, highlighting the most discriminative distribution-related features. The proposed FIND improves performance by 11.7% on the Gen-Image benchmark while running $126\times$ faster than existing methods. With no need for auxiliary diffusion models and reconstruction, it offers a practical, efficient, and generalizable way to detect diffusion-generated content.

Introduction

The rapid advancement of diffusion models (Dhariwal and Nichol 2021; Ho, Jain, and Abbeel 2020; Nichol et al. 2021; Rombach et al. 2022) has revolutionized the field of generative artificial intelligence, enabling the synthesis of highly realistic images that are often indistinguishable from real photographs. While this technological breakthrough has opened new possibilities in creative domains (Zhou et al. 2025; Pan et al. 2022; He et al. 2025), it has also introduced significant challenges in ensuring the trustworthiness of digital media (Juefei-Xu et al. 2022; Carlini et al. 2023). The

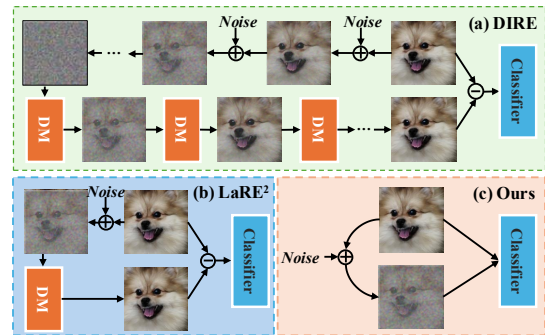


Figure 1: Comparison between FIND and previous noise-based methods. (a) DIRE utilizes complete reconstruction by adding noise and then denoising it with a Diffusion Model (DM) in multiple steps; (b) LaRE² compresses the noise addition and denoising steps into a single step within the latent space; (c) Our method only involves adding noise without the denoising process. Compared with previous methods, FIND avoids explicit reconstruction and omits the dependence on DM.

ability to generate convincing synthetic images raises critical concerns regarding misinformation, intellectual property rights, and digital security. As a result, the detection of generated images has become urgent and non-trivial.

Current mainstream approaches (Wang et al. 2023; Luo et al. 2024; Ma et al. 2023) to detecting diffusion-generated images primarily rely on sophisticated reconstruction-based methods, which utilize an additional diffusion model to assess reconstruction errors. Based on *reconstruction assumption* that diffusion-generated images are more easily reconstructed by a diffusion model compared with real images, DIRE (Wang et al. 2023) uses the reconstruction error as image representations and employ a binary classifier on them. Following DIRE, SeDID (Ma et al. 2023) further exploits intermediate steps during the reconstruction process. Among these methods, an extra diffusion model, serving as the reconstruction model, needs to infer multiple steps to achieve satisfactory results as shown in Fig. 1(a), which leads to low efficiency. To improve efficiency, LaRE (Luo et al. 2024) introduces one-step solutions based on reconstruction from the latent space depicted in Fig. 1(b). These methods, while

*Corresponding Author.

effective, suffer from two major limitations: poor generalizability across various image generation models and prohibitive computational resource requirements. The generalizability issue arises from the specific diffusion model used to calculate the reconstruction errors, while the computational overhead stems from the need to process each image through a complex reconstruction pipeline.

The limitations of existing methods motivate us to consider the detection method without an explicit reconstruction process along with an extra reconstruction model. We start by hypothesizing that the discriminative features derived from the current reconstruction process fundamentally stem from the differing abilities of Gaussian distributions to approximate image statistics, which can be easily derived from the *reconstruction assumption*. Specifically, real images are more difficult to fit with Gaussian distributions compared to their synthetic counterparts. This hypothesis provides a new perspective on understanding the nature of the differences between real and synthetic images. However, directly using a classifier without prior knowledge to identify this crucial discriminative feature is extremely challenging. The image distribution contains multiple dimensions and various features like visual elements, which can interfere with the classifier’s ability to focus on the key discriminative information. Previous methods tackle these by directly providing reconstruction error as a discriminative feature. This approach eliminates visual and other interference factors as much as possible through the process of subtracting reconstructed images from original ones but inevitably introduces additional computational overhead.

In this paper, we propose a novel method named **Forgery Identification via Noise Disturbance (FIND)** to provide the discriminative feature implicitly, with solely the binary classifier as shown in Fig. 1(c). Specifically, during the training phase, we add random Gaussian noise to real images and label these noisy images as synthetic images. This operation serves two purposes. Firstly, similar to synthetic images, real images with added noise are more easily fitted by Gaussian distributions compared to the original real images, which helps to emphasize the discriminative feature as we hypothesized before. This point is theoretically proven in Section . Secondly, the noisy images retain a certain visual similarity to the original real images. This similarity allows us to eliminate other interfering features in the image distribution, enabling the classifier to concentrate more on the discriminative feature. We use real images, noisy real images, and synthetic images to train a basic binary classifier. During the inference phase, the images are directly input into the binary classifier to get the prediction. Our proposed method offers several advantages over existing approaches. Most notably, it eliminates the need for a specific reconstruction model and the reconstruction process, which significantly improves the generalizability of the detection method and reduces both the training and inference time. Specifically, our method demonstrates superior performance on the large-scale Gen-Image benchmark (Zhu et al. 2023), achieving an 11.7% performance improvement compared to existing methods. And, due to its simple architecture, our method is $126\times$ faster than the state-of-the-art methods, making it more practical

for real-world applications.

In conclusion, this paper presents a simple yet effective method for detecting images generated by diffusion models. It not only provides a new theoretical understanding of the discriminative features in the reconstruction process, but also demonstrates that noised real images can serve as boundary samples, maintaining content consistency with real images while exhibiting distribution patterns similar to synthetic images. Based on this insight, we offer a more practical baseline for future research. The extensive experiments on the large-scale benchmark demonstrate the effectiveness and efficiency of our method. And our method will provide a brand-new baseline for this field.

Related Work

Diffusion Models for Image Generation

Diffusion models have emerged as a transformative paradigm for high-fidelity image generation, building on foundational principles from non-equilibrium thermodynamics. Early breakthroughs like Denoising Diffusion Probabilistic Models (Ho, Jain, and Abbeel 2020) (DDPMs) established the framework of iterative denoising, achieving performance competitive with leading GAN-based approaches (Karras et al. 2020). Subsequent research has focused on enhancing their practicality and versatility, with innovations spanning accelerated sampling algorithms (Nichol and Dhariwal 2021; Song, Meng, and Ermon 2021; Liu et al. 2022), architectural refinements (Dhariwal and Nichol 2021; Rombach et al. 2022), and novel conditioning mechanisms for tasks like text-to-image synthesis (Rombach et al. 2022; Gu et al. 2022b; Zhang, Rao, and Agrawala 2023). For instance, latent diffusion models (Rombach et al. 2022) significantly improved computational efficiency with compressed representations, while cross-attention mechanisms enabled fine-grained multimodal control (Zhang, Rao, and Agrawala 2023; Ruiz et al. 2023). Advancements in deterministic sampling (Song, Meng, and Ermon 2021) and pseudo-numerical methods (Liu et al. 2022) reduced inference steps without compromising quality, bridging the gap between probabilistic and implicit modeling. Beyond generation, diffusion models have been adapted for diverse applications, from image editing (Nichol et al. 2021; Hertz et al. 2023), inpainting (Lugmayr et al. 2022), to detection of synthetic content (Juefei-Xu et al. 2022), reflecting their broad impact. These developments, coupled with community-driven efforts like Stable Diffusion (Rombach et al. 2022), underscore their scalability and adaptability, positioning diffusion models as a cornerstone of modern generative AI.

Generated Image Detection

The rapid evolution of generative models has spurred significant advancements in synthetic image detection, with methods evolving alongside emerging generation paradigms. Early detection frameworks primarily focused on GAN-generated images, employing spatial artifact analysis through handcrafted features such as chromatic aberration (Mayer and Stamm 2018), color (McCloskey and Albright 2018), and saturation (McCloskey and Albright

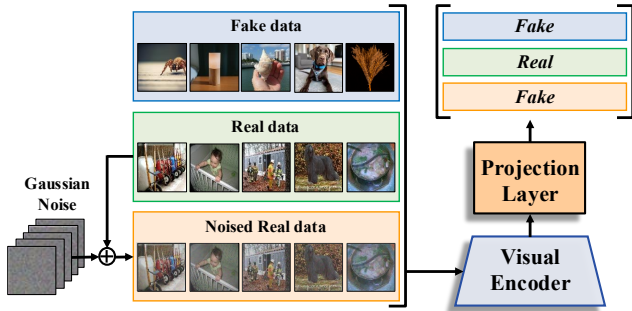


Figure 2: The training framework of FIND. Gaussian noise is added to real images, and these perturbed versions are labeled as synthetic within the training process. This enables FIND to mitigate the reconstruction model dependency, and learn the core distributional differences between real and synthetic images, resulting in a superior generalization ability.

2019), though these often struggled with generalization across architectures (Marra et al. 2018; Wang et al. 2020). Subsequent approaches leveraged frequency-domain inconsistencies caused by upsampling operations (Frank et al. 2020; Jeong et al. 2022) and model-specific fingerprints (Marra et al. 2019; Yu, Davis, and Fritz 2019; Wesselkamp et al. 2022) to improve cross-model robustness. The rise of pretrained vision models introduced new paradigms, where frozen feature extractors (Radford et al. 2021; Ojha, Li, and Lee 2023) or gradient-based representations (Tan et al. 2023) were harnessed for forgery cues, though debates persist about the necessity of feature adaptation versus static extraction. Few-shot learning (Tang et al. 2023, 2022; Tang, He, and Qin 2025), is also a potential direction. As diffusion models redefine generation quality, recent studies reveal critical performance gaps in detectors trained on GAN artifacts (Corvi et al. 2023a,b). This prompts the exploration of diffusion-specific signatures (Sha et al. 2023) via spectral analysis (Corvi et al. 2023a; Ricker et al. 2022), hybrid features (Yan et al. 2025), and reconstruction errors in the pixel space (Wang et al. 2023; Ma et al. 2023). Besides, language-guided contrastive learning (Wu, Zhou, and Zhang 2023) also shows promise. Recently, FatFormer (Liu et al. 2024) finds that the forgery adaptation of pre-trained models is essential for the generalizability of synthetic image detection. LARE (Luo et al. 2024) shows that reconstruction error in the latent space can also benefit diffusion-generated image detection. DRCT (Chen et al. 2024) utilizes diffusion reconstruction to generate hard samples for contrastive learning. However, computational efficiency and universal applicability across generative frameworks remain challenging.

Methodology

Preliminaries

Denosing Diffusion Probabilistic Models (DDPMs) (Ho, Jain, and Abbeel 2020) has emerged as a significant class within the realm of generative models, revolutionizing the

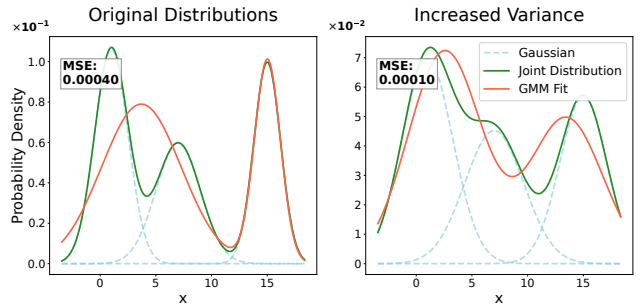


Figure 3: Visual illustration of Gaussian distribution with increased variances. **Left panel:** A joint distribution (green) from three Gaussians, with its two-component Gaussian mixture model fit (red). **Right panel:** Increasing the variances smooths the joint distribution, causing the two left-hand Gaussians to nearly merge. The drop of fitting MSE indicates that distributions with increased variance are easier to fit.

landscape of image generation. The DDPMs involves two key parts: the forward and reverse diffusion processes. In the forward process, which follows a Markov chain, Gaussian noise is incrementally added to the original image x_0 until degenerating it into isotropic Gaussian distribution. Mathematically, the transition from the image x_{t-1} at step $t-1$ to the noisy image x_t at step t is defined as:

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{\frac{\alpha_t}{\alpha_{t-1}}}x_{t-1}, (1 - \frac{\alpha_t}{\alpha_{t-1}})\mathbf{I}), \quad (1)$$

where $\alpha_1, \dots, \alpha_T$ are pre-defined noise schedule, and \mathbf{I} is the identity matrix. Leveraging the properties of the Markov process and Gaussian distributions, we can directly obtain x_t from x_0 :

$$q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\alpha_t}x_0, (1 - \alpha_t)\mathbf{I}). \quad (2)$$

The reverse diffusion process aims to reverse the corruption, which is also defined as a Markov chain, with the reverse transition probability:

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)), \quad (3)$$

where $\mu_\theta(x_t, t)$ and $\Sigma_\theta(x_t, t)$ are functions parameterized by a neural network θ . Instead of directly predicting x_{t-1} , the network is often trained to predict the noise ϵ added in the forward process. During training, the network ϵ_θ is optimized to minimize the loss function:

$$L_{simple}(\theta) = \mathbb{E}_{t, x_0, \epsilon} \left[\|\epsilon - \epsilon_\theta(\sqrt{\alpha_t}x_0 + \sqrt{1 - \alpha_t}\epsilon, t)\|^2 \right], \quad (4)$$

$$\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I}).$$

Hypothesis about Reconstruction Error

In prior methods, *e.g.*, DIRE and LaRE², the input images or latent features are passed through the forward and reverse processes defined in Eq. 1 and Eq. 3 for the reconstruction error. Despite substantial efforts to minimize the cost associated with these two processes, an additional reconstruction model is still required, along with at least one iteration of the forward and reverse processes.

We aim to eliminate the need for reconstruction error by understanding its underlying meaning. Considering that both the forward and reverse processes of the diffusion model involve adding Gaussian noise gradually, we hypothesize that the reconstruction error essentially measures how well a data distribution can be fitted by a Gaussian mixture model. Specifically, let the distribution of real images be $(P_{real}(\mathbf{x}))$, which can be expressed as a superposition of a large, possibly infinite, number of Gaussian distributions:

$$P_{real}(\mathbf{x}) = \sum_{i=1}^{N_1} w_i \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_i, \boldsymbol{\sigma}_i^2), \quad (5)$$

where N_1 is a large number, w_i are weights such that $\sum_{i=1}^{N_1} w_i = 1$, and $\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_i, \boldsymbol{\sigma}_i^2)$ is a Gaussian distribution with mean $\boldsymbol{\mu}_i$ and variance $\boldsymbol{\sigma}_i^2$. Let the distribution of synthetic images be $P_{synthetic}(\mathbf{x})$. Since synthetic images are generated by a diffusion model through finite iterations, it is a superposition of finite Gaussian distributions:

$$P_{synthetic}(\mathbf{x}) = \sum_{j=1}^{N_2} v_j \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_j, \boldsymbol{\sigma}_j^2), \quad (6)$$

where v_j are the weights with $\sum_{j=1}^{N_2} v_j = 1$, and $N_2 \ll N_1$. We use a Gaussian mixture distribution $Q(\mathbf{x}) = \sum_{k=1}^K u_k \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_k, \boldsymbol{\sigma}_k^2)$ to fit the above distributions, with minimizing the KL divergence $D_{KL}(P||Q) = \int P(\mathbf{x}) \log \frac{P(\mathbf{x})}{Q(\mathbf{x})} d\mathbf{x}$. For the synthetic image distribution $P_{synthetic}(\mathbf{x})$, since it is already a superposition of Gaussian distributions with relatively small numbers, because the structure of $Q(\mathbf{x})$ is similar to $P_{synthetic}(\mathbf{x})$, it is easier to find suitable parameters $u_k, \boldsymbol{\mu}_k, \boldsymbol{\sigma}_k^2$ such that $D_{KL}(P_{synthetic}||Q)$ is small. In contrast, for the real image distribution, since the number of Gaussian components is much larger than the number in Q , it is more difficult to fit, resulting in a larger reconstruction error. Therefore, the goodness of fit of $Q(\mathbf{x})$, *i.e.*, the reconstruction error, can serve as a discriminative feature to distinguish real images and synthetic images.

For a binary classifier $f(\cdot)$, we expect it to capture such discriminative features to identify synthetic images. However, the reconstruction error feature is not prominent and may interfere with other features such as visual elements. As a result, the classifier $f(\cdot)$ may be misinformed by biases in the training dataset. Previous methods explicitly construct the reconstruction error and feed it into $f(\cdot)$ so that $f(\cdot)$ can directly obtain the corresponding information. For example, in DIRE, this process can be represented as $f(\text{DIRE}(\mathbf{x})) = f(|\mathbf{x} - R(I(\mathbf{x}))|)$, where $|\cdot|$ denotes the absolute value, $I(\cdot)$ and $R(\cdot)$ are the inversion process adding noise and reconstruction process, respectively. Both of the processes are multiple-steps, and can be also further compressed into one step (Luo et al. 2024). Note that the noise-adding process $I(\cdot)$ does not require an additional model, and $R(\cdot)$ can be represented as a forward neural network. This inspires us to implicitly incorporate the $R(\cdot)$ process into the classifier $f(\cdot)$ to eliminate the dependence on the reconstruction process and the reconstruction model.

Adding random noise $\epsilon \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\sigma}_n^2)$ to real images, the distribution of the noisy images $P_{noisy}(\mathbf{x})$ is formulated as:

$$\begin{aligned} P_{noisy}(\mathbf{x}) &= P_{real}(\mathbf{x}) * \mathcal{N}(\mathbf{0}, \boldsymbol{\sigma}_n^2) \\ &= \sum_{i=1}^{N_1} w_i (\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_i, \boldsymbol{\sigma}_i^2) * \mathcal{N}(\mathbf{0}, \boldsymbol{\sigma}_n^2)) \quad (7) \\ &= \sum_{i=1}^{N_1} w_i \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_i, \boldsymbol{\sigma}_i^2 + \boldsymbol{\sigma}_n^2). \end{aligned}$$

Compare with P_{real} in Eq. 5, P_{noisy} increases the variance of the Gaussian distributions, making the entire distribution smoother. As the variance increases, the width of the Gaussian distributions increases, and the overlapping regions between different Gaussian components become larger, thereby reducing the number of Gaussian distributions with distinct main effects. We also provide a visual example in Fig. 3. The left-hand figure shows the joint distribution (green line) composed of three Gaussian distributions and the fitting results (red line) using a Gaussian mixture model with two components. In the right-hand figure, we increase the variances of the three Gaussian distributions. It can be seen that the joint distribution becomes smoother, and the two Gaussian distributions on the left part almost merge. When fitting this distribution, the mean squared error of the fitting decreases from 3.9×10^{-4} to 8×10^{-5} .

From the information-theoretic perspective, the Fisher information matrix of each component is $\frac{w_i}{\boldsymbol{\sigma}_i^2}$ for the real image distribution $P_{real}(\mathbf{x})$, and $\frac{w_i}{\boldsymbol{\sigma}_i^2 + \boldsymbol{\sigma}_n^2}$ for the noisy image distribution $P_{noisy}(\mathbf{x})$. After adding random Gaussian noise, the Fisher information of the distribution decreases, so that this distribution is easier to be fitted by the finite Gaussian mixture distribution $Q(\mathbf{x})$.

Therefore, such a noisy distribution is consistent with the synthetic image distribution in terms of the discriminative feature of the reconstruction error. Moreover, in other dimensions such as visual elements, since this distribution is obtained by adding random Gaussian noise to real images, it is consistent with the real image distribution.

Implementation of the FIND

Based on the above analysis, we present our proposed method FIND. Considering such noisy data is align with synthetic images on reconstruction error and consistent with real images on other dimensions, we can add this type of data to the training dataset and label them as synthetic data, prompting the classifier $f(\cdot)$ to focus more on the discriminative features.

Specifically, as shown in Fig. 2, during the training phase, given a batch of data $B = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots\}$, where $\mathbf{x}_i \in [0, 255]^d$ denotes the image and label $y \in \{0, 1\}$, with 0 representing synthetic images and 1 representing real images. We pick the real images in the batch B and add random Gaussian noise to them as:

$$\mathbf{x}'_i = \Pi_{Img}(\mathbf{x}_i + \epsilon \cdot \boldsymbol{\eta}), \boldsymbol{\eta} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \quad (8)$$

where ϵ is the hyper-parameter that controls the magnitude of the noise, and $\Pi_{Img}(\cdot) = \text{clip}(\cdot, 0, 255)$ is used to projects the input back to the image space, *i.e.*, $[0, 255]^d$.

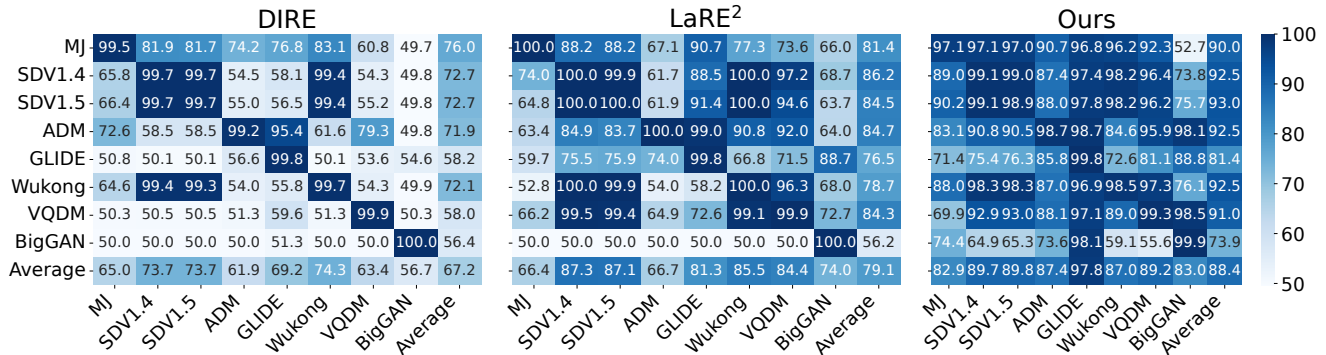


Figure 4: Cross-validation results across different training and testing subsets. We report the accuracy for DIRE, LaRE² and Ours on all 8 generators, where each row represents a corresponding training subset. Our method shows consistent and superior performance on different training sets and test sets.

Algorithm 1: Training procedure of FIND.

Require: Training set D , noise hyper-parameter ϵ , number of training epochs T .

- 1: Initialize parameters θ of classifier f
- 2: **for** epoch = 1 to T **do**
- 3: **for** batch $B = \{(x_i, y_i)\}$ in D **do**
- 4: Initialize augmented batch $B' \leftarrow \emptyset$
- 5: **for** (x_i, y_i) in B **do**
- 6: Add original example (x_i, y_i) to B'
- 7: **if** $y_i = 1$ **then**
- 8: Sample Gaussian noise $\eta \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
- 9: Generate noisy image $x'_i \leftarrow \Pi_{Img}(x_i + \epsilon \cdot \eta)$
- 10: Add synthetic example $(x'_i, 0)$ to B'
- 11: **end if**
- 12: **end for**
- 13: Update θ via backpropagation on B'
- 14: **end for**
- 15: **end for**
- 16: **return** trained classifier f .

With the noisy images, we augment the original batch as:

$$B' = B \cup \{(x'_i, 0) | x_i \in B \wedge y_i = 1\}. \quad (9)$$

Then we perform the normal binary classifier optimization with backpropagation to update $f(\cdot)$. During the testing phase, we directly input the test images into the classifier $f(\cdot)$ to obtain the discrimination results, without the need of extra Gaussian noise or reconstruction process. A rough illustration of the entire training process is provided in Alg. 1.

Experiments

Experimental Setups

Datasets and Evaluation Metric. All the experiments are conducted on the large-scale benchmark, GenImage (Zhu et al. 2023). This benchmark contains a total of 2,681,167 images, which are partitioned into 1,331,167 real images and 1,350,000 synthetic images. The real images are all from ImageNet (Deng et al. 2009). By leveraging 1000 distinct

labels from ImageNet for generation, it ensures a nearly equal distribution of real and generated images across each class. The synthetic images are generated from 8 different generative models: BigGAN (Brock 2018), GLIDE (Nichol et al. 2021), VQDM (Nichol et al. 2021), Stable Diffusion V1.4 (Rombach et al. 2022), Stable Diffusion V1.5 (Rombach et al. 2022), ADM (Dhariwal and Nichol 2021), Midjourney (Midjourney. 2022), and Wukong (Gu et al. 2022a). Each generator creates a similar quantity of images per class. This dataset is organized into 8 subsets accordingly, where each subset consists of the synthetic images generated by a single generator and their corresponding real images. Notably, real images are not shared across subsets, and the number of generated images closely matches the number of real images in both the whole dataset and each subset. We adhere to the official division of the dataset, with 2,581,167 images for training and the remaining 100,000 images for validation. We report Accuracy (ACC) as evaluation metric.

Implementation Details. Our method is implemented based on the pytorch-image-models library (Wightman 2019). We adhere to the default data augmentation techniques therein, which encompass operations such as random resizing and cropping, as well as random erasing. We mainly utilize the ViT-B/16 (Dosovitskiy et al. 2020) architecture as our backbone, taking into account its representational capabilities. For a fair comparison, following previous work (Wang et al. 2023; Luo et al. 2024), we also tested the performance of our model using ResNet50 (He et al. 2016) as the backbone. Both the ViT-B/16 and ResNet50 models are initialized from the pre-trained CLIP (Radford et al. 2021) following previous works (Luo et al. 2024; Liu et al. 2024). The training process is carried out on an NVIDIA A100 Tensor Core GPU. We conduct training for 30 epochs with a batch size of 96. The Stochastic Gradient Descent (SGD) optimization method is employed, and the learning rate (lr) is set to 1e-4. This combination of optimization method and learning rate setting has been carefully tuned to ensure efficient convergence during the training process. We set the hyperparameter ϵ of Gaussian noise as 50. We train 8 models separately on 8 distinct subsets. Each subset corresponds to a

Method	Backbone	Test diffusion generators								Average
		Midjourney	SD V1.4	SD V1.5	ADM	GLIDE	Wukong	VQDM	BigGAN	
CNNSpot (Wang et al. 2020)	RN50	58.20	70.30	70.20	57.00	57.10	67.70	56.70	56.60	61.73
Spec (Zhang, Karaman, and Chang 2019)	RN50	56.70	72.40	72.30	57.90	65.40	70.30	61.70	64.30	65.13
F3Net (Qian et al. 2020)	RN50	55.10	73.10	73.10	66.50	57.80	72.30	62.10	56.50	64.56
GramNet (Liu, Qi, and Torr 2020)	RN50	58.10	72.80	72.70	58.70	65.30	71.30	57.80	61.20	64.74
DIRE (Wang et al. 2023)	RN50	65.00	73.73	73.69	61.85	69.16	74.33	63.43	56.74	67.24
LaRE ² (Luo et al. 2024)	RN50	66.36	87.26	87.13	66.70	81.28	85.50	84.39	73.98	79.07
FatFormer [†] (Liu et al. 2024)	ViT-B/16	76.84	78.34	78.66	78.04	93.14	76.05	79.06	72.23	79.04
FIND	RN50	74.03	76.02	75.90	78.90	92.34	75.22	82.74	85.90	80.13
FIND	ViT-B/16	82.89	89.69	89.77	87.42	97.82	87.04	89.25	82.96	88.35

Table 1: Comparison of the accuracy of our method with existing methods. Each row in the table shows the average test accuracy of the corresponding method, calculated when trained on 8 subsets and tested on the respective test set. The last column gives the overall average accuracy. The symbol † indicates our reproduction from its source code on a CLIP-ViT-B/16 version. The proposed FIND outperforms the SOTAs on both RN50 and ViT-B/16 backbones.

different generation method. Model selection is then based on the performance in the validation set, and the validation set shares the same generator as the training set.

Comparison with State-of-the-Art Methods

We compare the accuracy of our method with existing methods in Tab. 1 and Fig. 4. In Tab. 1, values in each row represent the average test accuracies of the corresponding method. This average is calculated when the method is trained on 8 subsets and tested on the remaining test set. Compared with existing methods, our method demonstrates a clear superior performance. Specifically, when using the ResNet50 as the backbone of the classifier, the average accuracy of our method reaches 80.13%, which is higher than SOTA methods such as LaRE² (79.07%) and FatFormer (79.04%). We also employ the ViT-B/16 as the backbone due to its stronger capability of representation. Then the average accuracy is further increased by 10.3% to reach 88.35%. Our method performs stably on different test sets from different generators, and outperforms existing methods.

We further present the detailed accuracies in the form of matrix in Fig. 4 for DIRE, LaRE² and our method. For DIRE, the values of accuracy on the diagonal, where the training set and the test set are from the same generative model, are significantly higher than those in other positions. Through means like optimization in latent space, LaRE² has shown improvements compared to DIRE. However, it still falls short. Moreover, since both DIRE and LaRE² use Stable Diffusion v1.5 as their reconstruction model, these two methods achieve better results on test set from SD V1.4 and SD V1.5, yet this impedes their generalization to other models. Our method, without the reliance on the reconstruction model, is capable of achieving a more consistent and superior accuracy performance.

Subsequently, we compare the inference time costs of FIND with DIRE and LaRE². The forward processes of both DIRE and LaRE can be partitioned into a reconstruction process and a classification process. In Tab. 2, we report the time costs at different stages, the overall time cost, and the frames per second (FPS). This table is obtained by averaging the runs on 1000 samples with the batch size of 96. As presented in Tab. 2, for the DIRE method, it demands a reconstruction

Method	Time per image (ms)			FPS
	Rec	Cls	Total	
DIRE	151.6	0.3	151.9	6.6
LaRE ²	50.0	0.4	50.4	19.8
Ours	0.0	0.4	0.4	2500.0

Table 2: Average inference time of different methods. Without reconstruction, our method is significantly faster.

time of 151.6 milliseconds per image. This leads to a total time consumption of 151.9 milliseconds, resulting in a frame rate (FPS) of 6.6. In the case of the LaRE² method, it has a reconstruction time of 50.0 milliseconds, a cumulative total time of 50.4 milliseconds, and an FPS of 19.8. It should be noted that the time cost of the classification process is far less than that of the reconstruction process, and it can even be negligible. In contrast, our method completely eradicates the reconstruction process. As a consequence, the total inference time is drastically reduced to a mere 0.4 ms with an extremely high FPS of 2500. This remarkable advantage makes FIND suitable for real-time practical applications.

More Analysis

Subsequently, we conduct an in-depth analysis of the impact of different noise-adding methods on the results. We take into account three types of noise and their combinations. Firstly, we consider using Gaussian noise as images added to the training set. Secondly, we add random Gaussian noise to real images. Thirdly, we add random Gaussian noise to synthetic images. All these three types of noise-added data are labeled as synthetic images. In Tab. 3, we analyze the effects of using such noisy data, where N, R w/ N and S w/ N denote the three types of noisy data mentioned above respectively. The row 1 serves as the baseline, without any of the specific noise-adding strategies, and the model achieves an average accuracy of 80.58%. Note that this result also outperforms existing methods owing to the better backbone architecture and the appropriate data augmentation strategy. When only the Gaussian noise as images (marked by “✓” in the N column) is used in row 2, the average accuracy drops

	N	R w/ N	S w/ N	Midjourney	SD V1.4	SD V1.5	ADM	GLIDE	Wukong	VQDM	BigGAN	Average
1				73.97	83.59	83.54	76.35	91.45	80.78	83.62	71.30	80.58
2	✓			66.66	78.74	79.00	73.59	79.83	75.82	82.43	71.66	75.97
3	✓	✓		76.03	87.62	87.70	86.11	95.03	83.96	89.58	79.75	85.72
4			✓	64.23	79.29	79.40	71.18	76.98	76.09	82.12	70.45	74.97
5		✓	✓	75.23	86.10	86.31	85.48	96.74	82.99	87.52	75.61	84.50
6		✓		82.89	89.69	89.77	87.42	97.82	87.04	89.25	82.96	88.35

Table 3: Impact of different kinds of noise. N: using Gaussian noise as images; R w/ N: adding Gaussian noise to real images; S w/ N: adding Gaussian noise to synthetic images. The R w/ N plays a dominant role, consistent with our hypothesis.

to 75.97%. This indicates that simply adding Gaussian noise images is not beneficial for improving performance. In row 3, when both Gaussian noise as images (N) and adding random Gaussian noise to real images (R w/ N) are utilized, the average accuracy significantly increases to 85.72%. This shows that the combination of these two noise-adding methods can enhance the model’s generalization ability. The accuracy improvements are observable across various test set with different generators such as Midjourney, SD V1.4, SD V1.5. In row 4, when only adding random Gaussian noise to synthetic images (S w/ N) is applied, the average accuracy is 74.97%, which is relatively low. As hypothesized earlier, this kind of noisy data shares the same discriminative features and other interfering features with synthetic images. Moreover, since all of them are labeled as synthetic, they do not bring in new information. Instead, they may exacerbate data biases, thereby leading to lower accuracy. In row 5, when adding random Gaussian noise to both real and synthetic images (R w/ N and S w/ N), the average accuracy reaches 84.50%. This combination also shows positive impacts on the model’s performance, although slightly lower than the combination in row 3. Both row 3 and row 5 demonstrate the contribution of add noise to real images. Notably, in row 6, when only adding random Gaussian noise to real images (R w/ N), the model achieves the highest average accuracy of 88.35% among all the scenarios. The accuracy for each test generator is also high, such as 82.89% for Midjourney and 89.69% for SD V1.4. This clearly demonstrates that adding random Gaussian noise to real images is the most effective noise-adding method among the considered ones, which significantly enhance the model’s performance in distinguishing synthetic images.

We conduct a comprehensive analysis of the impact of the magnitude ϵ of the added noise on the results. We perform tests on the training subset for SDv1.5, and the corresponding results of different ϵ are depicted in Fig. 5a. From Fig. 5a, it is evident that initially as the ϵ increases, the accuracy of the model experiences a rapid upward trend. However, once the ϵ exceeds 20, the growth rate of the accuracy slows down and begins to fluctuate. Furthermore, we illustrate the visual effects of noisy images with different ϵ in Fig. 5b. The first column presents the original images, while the subsequent columns display the images with different magnitudes of added noise. As the noise magnitude ϵ increases, the quality of the images deteriorates rapidly. When an extremely large amount of noise is applied, the image may degenerate

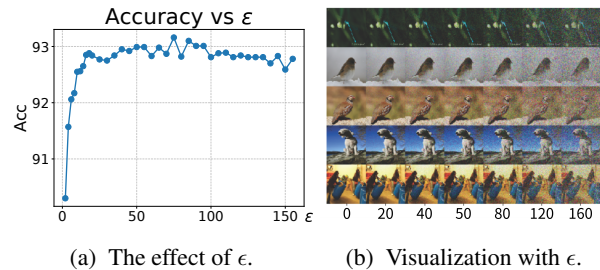


Figure 5: The impact of ϵ . The accuracy surges as ϵ rises initially and starts fluctuating when ϵ exceeds 20. The quality of the images deteriorates rapidly as ϵ increases. We pick a balanced value of $\epsilon = 50$.

into a standard Gaussian distribution, with similar effect as the second row of Tab. 3. Here these images fail to convey any meaningful visual information. Consequently, through a careful consideration of these factors, we opt for a more balanced value of 50 for the noise magnitude ϵ . This strikes a reasonable compromise between leveraging the beneficial effects of noise on model performance and maintaining an acceptable level of image integrity and visual information retention. By choosing this value, we aim to optimize the overall performance of our method in terms of both accuracy and the utilization of relevant visual data characteristics.

Conclusion

In this paper, we propose a straightforward yet efficient method named FIND for detecting images generated by diffusion models. We first propose the hypothesis that the reconstruction error utilized by prior methods essentially reflects the fitting of Gaussian distribution. Based on the hypothesis, we add random Gaussian noise to real images during the training of the classifier, which emphasizes the discriminative feature from reconstruction implicitly. We evaluate our method on the large-scale benchmark, GenImage, where a higher accuracy and lower computational cost are achieved compared with existing methods. We hope our method can serve as a new baseline for this field.

Acknowledgements

This work is supported by the China Postdoctoral Science Foundation (No.2025M771514).

References

- Brock, A. 2018. Large Scale GAN Training for High Fidelity Natural Image Synthesis. *arXiv preprint arXiv:1809.11096*.
- Carlini, N.; Hayes, J.; Nasr, M.; Jagielski, M.; Sehwal, V.; Tramèr, F.; Balle, B.; Ippolito, D.; and Wallace, E. 2023. Extracting Training Data from Diffusion Models. In *USENIX Security Symposium*.
- Chen, B.; Zeng, J.; Yang, J.; and Yang, R. 2024. Drct: Diffusion reconstruction contrastive training towards universal detection of diffusion generated images. In *ICML*.
- Corvi, R.; Cozzolino, D.; Poggi, G.; Nagano, K.; and Verdoliva, L. 2023a. Intriguing properties of synthetic images: from generative adversarial networks to diffusion models. In *CVPR*.
- Corvi, R.; Cozzolino, D.; Zingarini, G.; Poggi, G.; Nagano, K.; and Verdoliva, L. 2023b. On the detection of synthetic images generated by diffusion models. In *ICASSP*.
- Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *CVPR*.
- Dhariwal, P.; and Nichol, A. 2021. Diffusion Models Beat GANs on Image Synthesis. In *NeurIPS*.
- Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *ICLR*.
- Frank, J.; Eisenhofer, T.; Schönherr, L.; Fischer, A.; Kolossa, D.; and Holz, T. 2020. Leveraging frequency analysis for deep fake image recognition. In *ICML*.
- Gu, J.; Meng, X.; Lu, G.; Hou, L.; Niu, M.; Liang, X.; Yao, L.; Huang, R.; Zhang, W.; Jiang, X.; Xu, C.; and Xu, H. 2022a. Wukong: A 100 Million Large-scale Chinese Cross-modal Pre-training Benchmark. In *NeurIPS*.
- Gu, S.; Chen, D.; Bao, J.; Wen, F.; Zhang, B.; Chen, D.; Yuan, L.; and Guo, B. 2022b. Vector quantized diffusion model for text-to-image synthesis. In *CVPR*.
- He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*, 770–778.
- He, S.; Ji, P.; Yang, Y.; Wang, C.; Ji, J.; Wang, Y.; and Ding, H. 2025. A survey on 3d gaussian splatting applications: Segmentation, editing, and generation. *arXiv preprint arXiv:2508.09977*.
- Hertz, A.; Mokady, R.; Tenenbaum, J.; Aberman, K.; Pritch, Y.; and Cohen-or, D. 2023. Prompt-to-Prompt Image Editing with Cross-Attention Control. In *ICLR*.
- Ho, J.; Jain, A.; and Abbeel, P. 2020. Denoising diffusion probabilistic models. In *NeurIPS*.
- Jeong, Y.; Kim, D.; Min, S.; Joe, S.; Gwon, Y.; and Choi, J. 2022. Bihpf: Bilateral high-pass filters for robust deepfake detection. In *WACV*.
- Juefei-Xu, F.; Wang, R.; Huang, Y.; Guo, Q.; Ma, L.; and Liu, Y. 2022. Countering malicious deepfakes: Survey, battleground, and horizon. *IJCV*.
- Karras, T.; Laine, S.; Aittala, M.; Hellsten, J.; Lehtinen, J.; and Aila, T. 2020. Analyzing and Improving the Image Quality of StyleGAN. In *CVPR*.
- Liu, H.; Tan, Z.; Tan, C.; Wei, Y.; Wang, J.; and Zhao, Y. 2024. Forgery-aware adaptive transformer for generalizable synthetic image detection. In *CVPR*.
- Liu, L.; Ren, Y.; Lin, Z.; and Zhao, Z. 2022. Pseudo Numerical Methods for Diffusion Models on Manifolds. In *ICLR*.
- Liu, Z.; Qi, X.; and Torr, P. H. 2020. Global texture enhancement for fake face detection in the wild. In *CVPR*.
- Lugmayr, A.; Danelljan, M.; Romero, A.; Yu, F.; Timofte, R.; and Van Gool, L. 2022. Repaint: Inpainting using denoising diffusion probabilistic models. In *CVPR*.
- Luo, Y.; Du, J.; Yan, K.; and Ding, S. 2024. LaRE²: Latent Reconstruction Error Based Method for Diffusion-Generated Image Detection. In *CVPR*.
- Ma, R.; Duan, J.; Kong, F.; Shi, X.; and Xu, K. 2023. Exposing the Fake: Effective Diffusion-Generated Images Detection. *arXiv preprint arXiv:2307.06272*.
- Marra, F.; Gragnaniello, D.; Cozzolino, D.; and Verdoliva, L. 2018. Detection of gan-generated fake images over social networks. In *MIPR*, 384–389.
- Marra, F.; Gragnaniello, D.; Verdoliva, L.; and Poggi, G. 2019. Do gans leave artificial fingerprints? In *MIPR*, 506–511.
- Mayer, O.; and Stamm, M. C. 2018. Accurate and efficient image forgery detection using lateral chromatic aberration. *IEEE TIFS*, 13(7): 1762–1777.
- McCloskey, S.; and Albright, M. 2018. Detecting gan-generated imagery using color cues. *arXiv preprint arXiv:1812.08247*.
- McCloskey, S.; and Albright, M. 2019. Detecting GAN-generated imagery using saturation cues. In *ICIP*, 4584–4588. IEEE.
- Midjourney. 2022. Midjourney. <https://www.midjourney.com/>.
- Nichol, A.; Dhariwal, P.; Ramesh, A.; Shyam, P.; Mishkin, P.; McGrew, B.; Sutskever, I.; and Chen, M. 2021. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*.
- Nichol, A. Q.; and Dhariwal, P. 2021. Improved denoising diffusion probabilistic models. In *ICML*.
- Ojha, U.; Li, Y.; and Lee, Y. J. 2023. Towards universal fake image detectors that generalize across generative models. In *CVPR*.
- Pan, X.; Qin, P.; Li, Y.; Xue, H.; and Chen, W. 2022. Synthesizing Coherent Story with Auto-Regressive Latent Diffusion Models. In *WACV*.
- Qian, Y.; Yin, G.; Sheng, L.; Chen, Z.; and Shao, J. 2020. Thinking in frequency: Face forgery detection by mining frequency-aware clues. In *ECCV*.
- 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 *ICML*.

- Ricker, J.; Damm, S.; Holz, T.; and Fischer, A. 2022. Towards the detection of diffusion model deepfakes. *arXiv preprint arXiv:2210.14571*.
- Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; and Ommer, B. 2022. High-resolution image synthesis with latent diffusion models. In *CVPR*.
- Ruiz, N.; Li, Y.; Jampani, V.; Pritch, Y.; Rubinstein, M.; and Aberman, K. 2023. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *CVPR*.
- Sha, Z.; Li, Z.; Yu, N.; and Zhang, Y. 2023. De-fake: Detection and attribution of fake images generated by text-to-image generation models. In *CCS*, 3418–3432.
- Song, J.; Meng, C.; and Ermon, S. 2021. Denoising Diffusion Implicit Models. In *ICLR*.
- Tan, C.; Zhao, Y.; Wei, S.; Gu, G.; and Wei, Y. 2023. Learning on gradients: Generalized artifacts representation for gan-generated images detection. In *CVPR*.
- Tang, H.; He, S.; and Qin, J. 2025. Connecting Giants: Synergistic Knowledge Transfer of Large Multimodal Models for Few-Shot Learning. *arXiv preprint arXiv:2510.11115*.
- Tang, H.; Liu, J.; Yan, S.; Yan, R.; Li, Z.; and Tang, J. 2023. M3net: multi-view encoding, matching, and fusion for few-shot fine-grained action recognition. In *ACM International Conference on Multimedia*.
- Tang, H.; Yuan, C.; Li, Z.; and Tang, J. 2022. Learning attention-guided pyramidal features for few-shot fine-grained recognition. *Pattern Recognition*.
- Wang, S.-Y.; Wang, O.; Zhang, R.; Owens, A.; and Efros, A. A. 2020. CNN-generated images are surprisingly easy to spot... for now. In *CVPR*.
- Wang, Z.; Bao, J.; Zhou, W.; Wang, W.; Hu, H.; Chen, H.; and Li, H. 2023. DIRE for Diffusion-Generated Image Detection. In *ICCV*.
- Wesselkamp, V.; Rieck, K.; Arp, D.; and Quiring, E. 2022. Misleading deep-fake detection with gan fingerprints. In *SPW*.
- Wightman, R. 2019. PyTorch Image Models. <https://github.com/rwightman/pytorch-image-models>.
- Wu, H.; Zhou, J.; and Zhang, S. 2023. Generalizable Synthetic Image Detection via Language-guided Contrastive Learning. *arXiv preprint arXiv:2305.13800*.
- Yan, S.; Li, O.; Cai, J.; Hao, Y.; Jiang, X.; Hu, Y.; and Xie, W. 2025. A sanity check for ai-generated image detection. In *ICLR*.
- Yu, N.; Davis, L. S.; and Fritz, M. 2019. Attributing fake images to gans: Learning and analyzing gan fingerprints. In *ICCV*.
- Zhang, L.; Rao, A.; and Agrawala, M. 2023. Adding conditional control to text-to-image diffusion models. In *ICCV*.
- Zhang, X.; Karaman, S.; and Chang, S.-F. 2019. Detecting and simulating artifacts in gan fake images. In *WIFS*.
- Zhou, Y.; Zhou, D.; Cheng, M.-M.; Feng, J.; and Hou, Q. 2025. StoryDiffusion: Consistent Self-Attention for Long-Range Image and Video Generation. In *NeurIPS*.
- 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. In *NeurIPS*.