Denoising Trajectory Biases for Zero-Shot AI-Generated Image Detection

Yachao Liang[1,2] **Min Yu**[1,2*] **Gang Li**[3] **Jianguo Jiang**[1,2] **Fuqiang Du**[1,2]
Jingyuan Li[4] **Lanchi Xie**[5] **Zhen Xu**[1,2] **Weiqing Huang**[1,2]
[1]Institute of Information Engineering, Chinese Academy of Sciences
[2]School of Cyber Security, University of Chinese Academy of Sciences
[3]Deakin University [4]Beijing Technology and Business University
[5]Institute of Forensic Science, Ministry of Public Security
{liangyachao, yumin}@iie.ac.cn

Abstract

The rapid advancement of generative models has led to the widespread emergence of highly realistic synthetic images, making the detection of AI-generated content increasingly critical. In particular, diffusion models have recently achieved unprecedented levels of visual fidelity, further raising concerns. While most existing approaches rely on supervised learning, zero-shot detection methods have attracted growing interest due to their ability to bypass data collection and maintenance. Nevertheless, the performance of current zero-shot methods remains limited. In this paper, we introduce a novel zero-shot AI-generated image detection method. Unlike previous works that primarily focus on identifying artifacts in the final generated images, our work explores features within the image generation process that can be leveraged for detection. Specifically, we simulate the image sampling process via diffusion-based inversion and observe that the denoising outputs of generated images converge to the target image more rapidly than those of real images. Inspired by this observation, we compute the similarity between the original image and the outputs along the denoising trajectory, which is then used as an indicator of image authenticity. Since our method requires no training on any generated images, it avoids overfitting to specific generative models or dataset biases. Experiments across a wide range of generators demonstrate that our method achieves significant improvements over state-of-the-art supervised and zero-shot counterparts. Code is available here.

1 Introduction

Recent years, we have witnessed a spurt of development in the field of Artificial Intelligence Generated Content (AIGC). With the advent of cutting-edge generative models, such as StyleGAN [28] and Diffusion [24], the quality of synthetic images has been significantly improved. With tools like Stable Diffusion [50] and ControlNet [67], people can quickly create artistic images conforming to their ideas. Today, we can already see a large number of generated images on the Internet, which are hard to distinguishable from real images for humans [42]. While image generation technology can increase our enjoyment of life, the proliferation of fake images with misleading information also bring huge security risks.

In response to this, different detection methods have been explored [18, 63, 8, 23]. Researchers attempt to address this from different perspectives, e.g., frequency anomalies [18, 14, 60, 32] and

*Corresponding authors

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

semantic differences [62, 43, 36, 25]. Although previous methods achieve promising detection results, a significant issue is that the performance of detectors degrades considerably when they encounter images generated by unseen generators. A primary reason is that images generated by different generative models exhibit distinct forgery characteristics, and detectors tend to overfitting to the categories of fake images used in training.

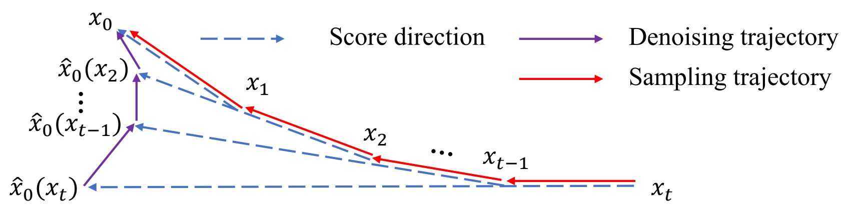

Figure 1: An illustration of the denoising trajectory. In the sampling process of PF-ODE based diffusion model, the noisy image is iteratively updated by moving a certain distance toward the denoising output $\hat{x}_0(t)$ predicted by a neural network at each step, eventually yielding a noise-free image. In this process, the intermediate denoised states x_t form the sampling trajectory, while the corresponding denoising outputs $\hat{x}_0(t)$ constitute the denoising trajectory.

Image generation is a progressive refinement process. This is especially evident in recent diffusion models [24, 56, 57], where the sampling process typically involves hundreds or even thousands of steps. For another branch of generative models, GANs [21, 3, 28], previous studies [65, 19, 17] have also suggested that the evolution of generator parameters during training can be viewed as a diffusion-like process. Most existing detectors [14, 34, 68] focus on identifying artifact-based features in final generated images, while few have explored the informative signals embedded in the image generation process itself. In this work, we attempt to uncover discriminative information between real and generated images by analyzing their generation processes. Specifically, we simulate the generation process using DDIM inversion [55] and construct the denoising trajectories of input images, as shown in fig. 1. We observe that, compared to real images, the denoising trajectories of generated images exhibit earlier mode change, with denoising outputs converging more rapidly to the final image. As a result, the similarity between the denoising outs and the final image can serve as an effective indicator for distinguishing between real and generated images.

Inspired by the above findings, we propose a novel zero-shot method for detecting generated images. Concretely, we perform DDIM inversion [55] on the input image to simulate the sampling process and collect the denoising outputs throughout the process. Then we use a pre-trained CLIP [46] model to extract semantic features and compute the feature similarity between the original image and each image along the denoising trajectory, summing the results to obtain a final similarity score. Additionally, we further incorporate embeddings from intermediate layers of CLIP to capture fine-grained features.

To validate the effectiveness of our method, we conducted extensive experiments on multiple datasets of generated images. The evaluation covered a wide range of generative models, including some of the most recent ones. Experimental results demonstrate that our approach exhibits strong generalization ability. Our contributions can be summarized as follows:

- We demonstrate that the denoising outputs of generated images converge faster than those of real images.
- We propose a novel zero-shot method for detecting generated images based on the similarity between images along the denoising trajectory and the input image.
- Through extensive experiments, our method presents superior generalization ability. Notably, it exhibits average performance improvements of 8.1% in accuracy and 9.8% in average precision over the state-of-the-art, evaluated across 21 generators.

2 Preliminaries

2.1 Denoising Trajectory of ODE-based Diffusion Model

Given a data distribution p_{data}, the forward process of the diffusion model [54, 56, 24] gradually add noise with perturbing kernel $p_t(x_t|x_0) = \mathcal{N}(x_t;\ \sqrt{\alpha_t}x_0,\ \sigma^2(t)\mathbf{I})$, where $x_0 \sim p_{data}$, $t \in [0, T]$ and α_t, $\sigma^2(t)$ are noise schedules. Eventually, p_T will follow a standard Gaussian distribution. Song et al. [58] presents a generalized framework of this process with stochastic differential equation (SDE), and further propose the corresponding probability flow ordinary differential equation (PF-ODE), which shares the same marginal distributions as the SDE. Images can be deterministically generated by constructing the reverse-time PF-ODE. Particularly, DDIM [55] is a special case of the PF-ODE, with the following form:

$$x_{t-1} = \sqrt{\alpha_{t-1}}\hat{x}_0(x_t; \sigma_t) + \sigma_{t-1}\epsilon_\theta(x_t, t) \tag{1}$$

and

$$\hat{x}_0(x_t; \sigma_t) = \frac{x_t - \sigma_t\epsilon_\theta(x_t, t)}{\sqrt{\alpha_t}} \tag{2}$$

where $\hat{x}_0(x_t; \sigma_t)$ is an estimate of the original sample x_0 and $\epsilon_\theta(x_t, t)$ denotes the noise predicted by the neural network. The denoising trajectory can be obtained by taking denoising outputs, $\hat{x}_0(x_t; \sigma_t)$, at each step of this sampling process. For existing images, we can use the DDIM inversion to obtain the approximate sampling process and denoising outputs. we provide some examples of denoising outputs in appendix A.3

2.2 Generated Image Detection

High-Frequency Based Detection. Previous works point out that AI-generated images show anomaly spectral distribution. Concretely, generated images exhibit different high-frequency mode [14, 13], which is believed to be caused by the upsampling operation in the neural network. Inspired by this insight, some works detect generated images by extracting low-level information [60, 40, 10]. Typically, researchers extract high-frequency components of images through wavelet transform [32], noise pattern extraction [34], or resampling residual feature analysis [60]. [10] reveals the high-frequency differences between real and generated images in an unsupervised manner.

Semantic Based Detection. Another route line explores semantic features for generated image detection. Researchers utilize pre-trained models to extract features in advance or finetune them, thereby guiding detectors to focus more on semantic features. [62] show that a naive Resnet50 trained on ProGAN-generated [27] images can generalize to other GAN-generated images. In addition, [43] proposed to map images into an universal space by the image encoder of CLIP [46] to boost the generalization ability of detectors. [35] further suggests incorporate the text encoder of CLIP to introduce language information. Recently, SIDA [25] performs explainable detection with the aid of rich visual and textual knowledge of large multimodal model.

3 Method

In this section, we first analyze the differences in the denoising trajectories between real and generated images and conduct preliminary experiments to validate our findings. We then introduce our zero-shot method for detecting generated images, which is based on the insights derived from the preceding analysis.

3.1 Denoising Trajectory Analysis

Previous studies have found that the generation process of diffusion models can be regarded as a frequency autoregressive process [49, 12, 16]. Since random noise has equal energy across all frequencies, the forward noising process progressively destroys image content from high to low frequencies. Conversely, the reverse denoising process, i.e., the generation process, gradually restores image information from low to high frequencies. The above description refers to the changes in images along the sampling trajectory. Similarly, the images along the denoising trajectory, also conform to a frequency-autoregressive generation process that progresses from low to high frequencies. Below, we provide a simple analysis following the method in [12].

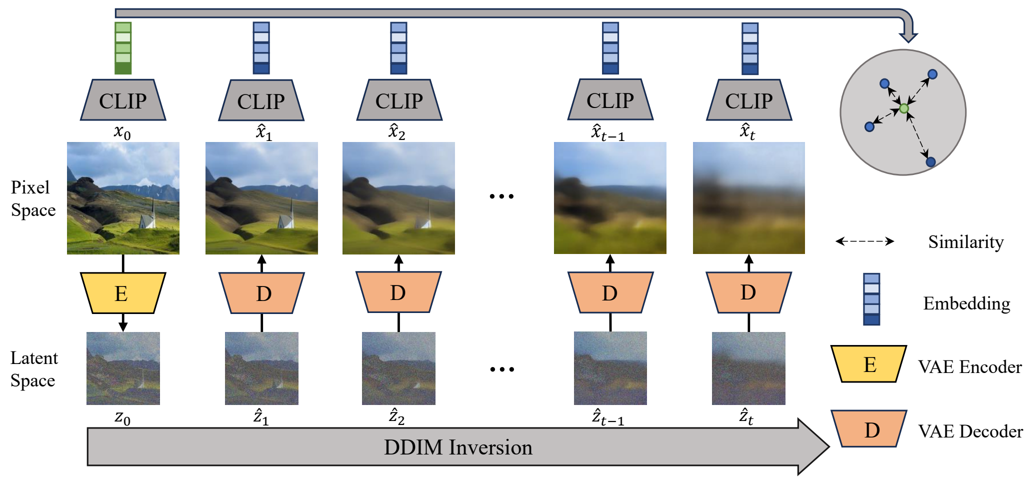

Figure 2: Overview of the proposed method. Given an input image, we employ DDIM inversion to approximate its sampling process and obtain the denoising outputs at each step. We then use CLIP to extract features from both the original image and each image along the denoising trajectory, and compute their cosine similarities, which serves as the criteria for detecting generated images.

Consider a Gaussian forward diffusion process:

$$x_t = \alpha_t \, x_0 + \sigma_t \, \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, I), \tag{3}$$

ε is standard Gaussian noise. Let $\mathcal{R}[x](f)$ denote the *radially averaged power spectral density* (RAPSD) of image x at spatial frequency f. For white Gaussian noise, $\mathcal{R}[\varepsilon](f) = 1$ holds for all f. We define that a frequency component is detectable if its signal power exceeds a given signal-to-noise ratio (SNR) threshold $\tau > 0$.

Because the Fourier transform is a linear operator and the power spectrum scales quadratically with amplitude, we have

$$\mathcal{R}[\alpha_t x_0](f) = \alpha_t^2 \, \mathcal{R}[x_0](f), \quad \mathcal{R}[\sigma_t \varepsilon](f) = \sigma_t^2 \, \mathcal{R}[\varepsilon](f) = \sigma_t^2 \tag{4}$$

A frequency component is considered detectable if

$$\mathcal{R}[\alpha_t x_0](f) > \tau \, \mathcal{R}[\sigma_t \varepsilon](f) = \tau \, \frac{\sigma_t^2}{\alpha_t^2}, \tag{5}$$

As the noise level decreases during the reverse diffusion process, the ratio σ_t/α_t decreases. Consequently, $f_{\max}(t)$, the maximal detectable frequency, increases monotonically, implying that higher frequencies become progressively detectable. Since the model does not hallucinate information that is completely obscured by noise, the spectral content of its denoised outputs will primarily consist of the detectable frequencies that satisfy the above inequality. Therefore, the images along the denoising trajectory will gradually transition from low-frequency components to high-frequency details, which is consistent with the actual observations. In summary, the images in the denoising trajectory, $\hat{x}_0(x_t; \sigma_t)$, follow a progressive refinement process from low to high frequencies, as shown in fig. 3.

Previous works suggest that there are discrepancies between the distribution learned by diffusion models and the true distribution of real images [64, 38, 53]. It inspires us to make a reasonable speculate that the diffusion model can predict generated images more accurately, which means the denoising trajectories of generated images will converge more quickly according to above analysis. To verify this conjecture, we inverse the sampling processes of some real and generated images with DDIM inversion [55], and collect the denoising outputs during these processes. Next, we measure their spectral similarity with the original image. Specifically, we compute the power spectral density of the images at different timesteps along the denoising trajectory and calculate their differences from the original image at each frequency. Since the effective frequency range of these images progresses from low to high frequencies, we apply corresponding frequency masks when computing the differences. As shown on the right side of the fig. 3, the differences for real images are larger than those for generated images. It indicates that the denoising outputs of generated images achieve faster convergence toward the target images. As a result, compared to real images, generated images and the intermediate images along the denoising trajectories will exhibit higher similarity.

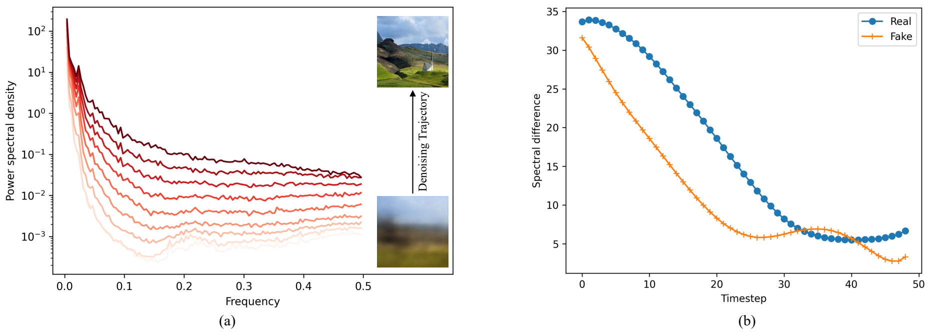

Figure 3: (a) Frequency transition of images along the denoising trajectory (reverse process). (b) Spectral differences between real/fake images and the images along their denoising trajectories: We randomly select 500 real images from ILSVRC [51] and fake images generated by Stable Diffusion v1.5 [50], then calculate the Power Spectral Density (PSD) of each image in the denoising trajectory and their differences from the original image at each frequency.

3.2 Generated Image Detection

Using the above biases, we propose a novel zero-shot AI-generated image detection method, which employs the similarity between the intermediate images along the denoising trajectory and the original image as as a criterion to differentiate real and generated images. As shown in fig. 2, we first apply DDIM inversion [55] to the input image and collect the denoising outputs at each step. Note that we employ the commonly used latent diffusion model, therefore, we need to utilize the VAE encoder to yield inputs and the VAE decoder to the outputs. To calculate the similarity between different images, we extract features from images using CLIP [46], which demonstrates superior generalization ability owing to its pretraining on a large corpus of image-text pairs. We then calculate the cosine similarity between the features of each denoised image and those of the original image, and aggregate these similarities to obtain the final semantic similarity score.

$$S(x) = \frac{1}{T}\sum_{i=0}^{T} sim(emb(x), emb(\hat{x}_i)) \tag{6}$$

where $sim(\cdot\,,\,\cdot)$ denotes cosine similarity, and we use the class embeddings as features of images.

Incorporating features of intermediate layers. An intuitional way to extract features of an image using CLIP is to take the output from its final layer as the representation of the image as in [43, 35]. However, the features extracted by different layers of CLIP exist non-negligible differences. Specifically, shallow layers of CLIP focus on low-level features such as textures, while deep layers extract high-level semantic representations or concepts of the entire image [20, 61, 26]. Directly using the embeddings from the final layer will overlook fine-grained content, which is important for our detection task since the denoising outputs in the later stages of sampling differ from the original image only in fine details. Therefore, to capture both the global and fine-grained features of the image, we use the features extracted from each layer of CLIP for similarity computation. We provide a more intuitive demonstration of this in fig. 4. In order to identify the sensitivity of each layer of CLIP to fine-grained changes, we apply blurring to images and compute the embedding similarity between the original and blurred images at each layer.

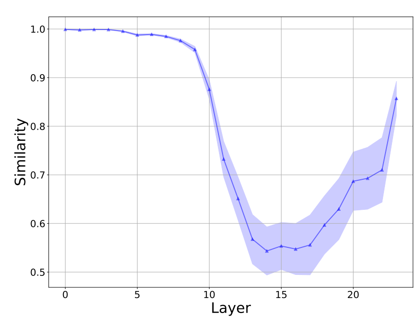

Figure 4: Sensitivity of each layer of CLIP to fine-grained changes.

Table 1: Cross-architecture generalization. We report the Acc (%) and AP (%) on ForenSynths. The supervise baselines are trained on ProGAN, except for SIDA trained on their custom dataset. We adopt either the officially released pre-trained models or reproduce the results by training according to the provided code repositories. Zero-shot methods are displayed with the gray background.

Method	ProGAN		GauGAN		BigGAN		StarGAN		CycleGAN		StyleGAN		StyleGAN2		AVG	
	Acc	AP	Acc	AP	Acc	AP	Acc	AP	Acc	AP	Acc	AP	Acc	AP	Acc	AP
NPR [60]	99.8	**100.**	82.5	85.5	84.4	87.8	**99.3**	99.9	96.1	98.5	97.7	99.8	**98.4**	**99.9**	94.0	95.9
FreqNet [59]	99.6	**100.**	93.4	98.6	90.5	96.0	85.7	99.8	95.8	99.6	90.2	99.7	87.9	99.5	91.9	99.0
FreDect [18]	99.4	**100.**	80.5	82.8	82.0	93.6	94.6	99.5	78.8	84.8	78.0	89.0	66.2	82.5	82.8	90.3
CNNSpot [62]	**100.**	**100.**	81.4	90.8	71.1	86.0	94.6	99.0	87.6	94.9	87.6	99.8	85.4	99.4	86.8	95.7
UnivFD [43]	**100.**	**100.**	**99.5**	**100.**	**95.1**	**99.3**	95.7	99.4	**98.1**	**99.9**	**98.5**	**100.**	74.2	98.4	**94.4**	**99.6**
SIDA* [25]	71.3	77.3	70.4	69.7	74.8	82.9	66.3	95.9	68.1	64.1	64.0	70.0	51.4	59.3	66.6	74.2
AEROBLADE [48]	47.4	50.3	44.8	42.6	50.1	47.4	51.1	47.1	42.2	41.0	46.9	43.6	37.4	38.2	45.7	44.3
MIBD [7]	90.3	97.4	87.9	97.9	77.6	86.8	50.6	55.3	73.4	90.5	70.5	76.5	64.2	72.8	73.5	82.5
ZeroFake [53]	48.0	44.9	58.6	60.6	50.3	53.3	45.8	39.9	55.6	46.4	49.2	52.1	45.5	46.4	50.4	49.1
Ours	96.4	99.8	92.1	97.5	93.5	97.7	97.8	**100.**	84.6	94.0	88.8	98.0	75.8	94.8	89.9	97.4

As we have previously analyzed, the denoising outputs of generated images converge more quickly. We can distinguish generated images from real counterparts based on the similarity of the semantic representations between the original image and the images along the denoising trajectory, with real images exhibiting higher similarity scores. Since our method does not use any type of fake images for training, it avoids the overfitting problem and theoretically has better generalization ability.

4 Experiments

4.1 Experiment Setup

Baselines. We compare our method with three types of detector, including high-frequency based methods, semantic feature based methods, and zero-shot detection methods. Specifically, for the high-frequency based methods, FreqNet [59], FreDect [18], and NPR [60] extract high-frequency features of images using Fourier transform, discrete cosine transform, and resampling, respectively. The semantic-based methods include CNNSpot [62], UnivFD [43], and SIDA [25], which respectively use ResNet, CLIP, and Large Multimodal Model [31] as backbones to extract semantic features and are fine-tuned for the task of generated image detection. Finally, we also compare our method with three zero-shot detection methods. AEROBLADE [48] identifies fake images based on the differences between the original image and the one reconstructed by the VAE. ZeroFake [53] achieves generated image detection based on the similarity between original image and the image edited using diffusion model. MIBD [7] approximates the curvature and gradient of the probability manifold to enable zero-shot detection. Please see appendix A.1 for more details.

Datasets. To verify the effectiveness of our method, we benchmark on a large number of fake images generated by different types of generators involving GANs and diffusion models. **ForenSynths** [62] contains images generated by various GANs, e.g., ProGAN [27] and StyleGAN [28]. The real images are collected from LSUN [66], ImageNet [52], COCO [33], and CelebA [37]. **GenImage** [70] include 8 early text-to-image diffusion datasets, such as Stable Diffusion V1.4 [50] and Glide [41], and the real images are sampled from ImageNet [52]. **New Generator**: Considering the rapid development of generative models, we also test on images generated by several cutting-edge generative models. Specifically, we take the test set of COCO [33] as the real image dataset, then we collect images generated by FLUX [30], Stable Diffusion XL (SDXL) [45], and Stable Diffusion V3 (SD3) [15] with corresponding prompts of real images. We further collected fake images generated by DALLE3 [5], Firefly, and Midjourney-v5 (MJv5) [1] from [4]. See appendix A.2 for the full list of generative models we used.

Implementation Details. We perform a 50-step DDIM inversion with Stable Diffusion v1.5, before which we crop images to the size of 512×512. We use CLIP ViT-L/14 to extract features. Our experiments are implemented with PyTorch on NVIDIA A100 GPU. We set the detection threshold as 0.75.

Table 2: Cross-paradigm generalization in terms of **Acc** performance.

Method	GenImage								New Generator						AVG
	SD1.4	SD1.5	ADM	DALLE2	MJ	Glide	VQDM	Wukong	DALLE3	Firefly	MJv5	SDXL	FLUX	SD3	
NPR [60]	78.6	78.9	69.7	64.9	77.8	78.3	78.1	76.1	79.0	73.6	80.0	80.0	80.3	79.9	76.8
FreqNet [59]	64.2	64.9	83.3	55.1	69.8	81.6	81.6	57.7	50.4	61.2	74.2	82.6	70.2	55.8	68.0
FreDect [18]	39.5	39.9	64.3	34.6	46.4	55.0	78.8	41.0	33.0	52.6	44.4	66.7	28.0	30.6	46.8
CNNSpot [62]	51.0	51.4	57.6	49.5	52.2	55.4	53.5	49.8	46.6	54.0	54.8	61.8	49.0	48.4	52.5
UnivFD [43]	63.4	63.3	66.6	50.7	55.9	62.2	85.3	70.8	49.7	92.3	54.9	70.3	49.7	53.8	63.5
SIDA* [25]	48.0	48.9	53.6	60.5	59.3	48.8	50.0	55.5	**84.0**	58.6	67.9	62.2	86.9	77.9	61.6
AEROBLADE [48]	96.7	97.2	64.7	79.3	97.3	**86.8**	56.1	98.0	51.5	61.2	75.1	64.1	92.0	86.9	79.1
MIBD [7]	62.0	63.0	57.3	77.7	55.5	64.3	76.9	65.4	49.8	57.8	54.1	59.8	50.2	56.9	60.8
ZeroFake [53]	87.1	87.7	**81.5**	**82.2**	70.1	82.9	67.5	84.2	46.0	47.0	54.0	54.7	58.0	52.3	68.2
Ours	**98.4**	**97.7**	78.7	74.0	**97.0**	80.1	**91.8**	**99.0**	68.0	**95.1**	**98.7**	**98.3**	**98.6**	**98.6**	**91.0**

Table 3: Cross-paradigm generalization in terms of **AP** performance.

Method	GenImage								New Generator						AVG
	SD1.4	SD1.5	ADM	DALLE2	MJ	Glide	VQDM	Wukong	DALLE3	Firefly	MJv5	SDXL	FLUX	SD3	
NPR [60]	84.0	84.6	74.6	76.7	85.4	85.7	81.2	80.5	86.0	77.8	88.9	89.1	88.6	87.9	83.6
FreqNet [59]	74.3	75.6	91.4	54.5	78.9	88.8	89.6	66.9	55.9	66.0	80.6	90.3	76.6	61.3	75.1
FreDect [18]	37.8	37.8	61.8	38.2	46.1	52.9	85.1	39.6	36.6	49.2	44.7	76.7	34.1	32.5	48.1
CNNSpot [62]	59.2	60.0	76.2	53.5	58.7	71.6	67.7	57.0	42.1	62.5	64.6	75.1	49.2	47.1	60.3
UnivFD [43]	86.7	86.4	87.3	63.2	75.0	84.4	96.7	91.5	50.4	99.3	77.9	92.7	50.1	77.3	79.9
SIDA* [25]	53.1	52.3	65.3	71.9	69.7	50.9	40.6	72.7	**93.5**	61.3	69.5	63.8	96.4	91.7	68.1
AEROBLADE [48]	98.2	98.9	80.3	92.1	**99.7**	**96.8**	76.1	99.3	60.6	73.3	86.9	79.2	97.0	94.3	88.0
MIBD [7]	72.3	73.4	65.4	88.1	60.9	77.8	87.8	76.6	53.5	67.5	60.3	71.3	54.2	64.6	69.5
ZeroFake [53]	94.2	95.4	90.0	90.7	77.0	91.1	74.3	90.8	48.2	43.8	58.5	61.7	60.3	55.7	73.7
Ours	**100.**	**100.**	**93.6**	**93.6**	**99.7**	93.9	**97.2**	**100.**	88.6	**98.6**	**99.8**	**99.8**	**99.8**	**99.9**	**97.5**

4.2 Comparison to Baselines

To compare the generalization of our method with other approaches, following previous works [60, 35, 43] we consider two experiment settings, i.e., cross-architecture and cross-paradigm. Specifically, models are trained on images generated by one type of GAN, and under these two settings, they are tested on images generated by other GANs and diffusion models respectively. SIDA [25] is an exception because their models need to be trained on customized datasets labeled with tampered regions and corresponding descriptions, hence we directly use the pre-trained models they provide for evaluation. ***Note that for zero-shot methods, including ours, these two settings are equivalent.*** To evaluate the performance of the proposed method, we use accuracy (Acc) and average precision (AP) metrics.

Cross-Architecture Generalization. In order to assess the generalization on images of GAN sources, we employ ForenSynths [62] for evaluation. Concretely, models are trained on the training set generated by ProGAN [27], which involves four types of images (cat, chair, car, and horse), and then evaluated on the test set containing other GANs. We report the results in table 1.

It can be observed that most supervised methods achieve good detection performance on images generated by GANs. This is because images generated by different GAN architectures tend to share similar artifact patterns [14, 62]. Among frequency-based and semantic-based approaches, NPR [60] and UnivFD [43] demonstrate the best performance, respectively. Regarding zero-shot detection methods, AEROBLADE and ZeroFake yields almost random results. On the contrary, our method and MIBD [7] present better detection performance, and our method outperforms MIBD 16.4% and 14.9% in terms of average Acc and AP, respectively. We note that although our analysis is based on sampling process of diffusion , we find our method can get promising performance on images generated by GANs.

Cross-Paradigm Generalization. Due to the differences in artifact patterns between diffusion models and GANs [18, 47], cross-paradigm poses a more challenging problem. Following [60, 43, 68], we report the results of models trained on images generated by ProGAN from the ForenSynths dataset and tested on diffusion-generated images from other datasets. The Acc and AP results are presentd in table 2 and table 3. All supervised methods experience a significant drop in performance, especially on images generated by the new generators. UnivFD [43] despite achieving a promising AP, suffers from low accuracy due to differing optimal classification thresholds for images generated by diffusion models and GANs. SIDA [25] demonstrates high detection performance on some cutting-edge diffusion models, such as FLUX and DALLE3, but its effectiveness on other models still requires improvement. Notably, our method exhibits powerful performance on almost all diffusion models. In terms of average Acc and AP, our method outperforms the second-best approach, i.e., AEROBLADE, by **11.9%** and **17.5%**, respectively. We also present the overall detection results of each method in table 4.

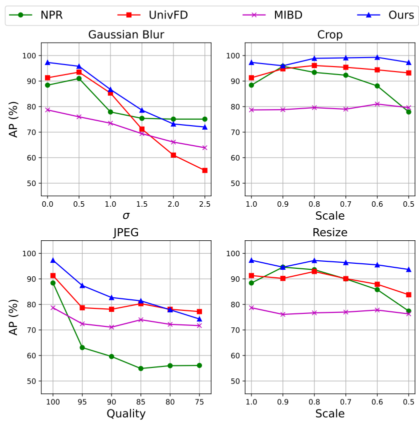

Figure 5: Robustness evaluation on common perturbations, measured in AP(%).

Table 4: Overall detection performance. We report Acc and AP results of each method on the three datasets and the average results on the 21 generators.

Method	ForenSynths		GenImage		New Genrator		AVG	
	Acc	AP	Acc	AP	Acc	AP	Acc	AP
NPR [60]	94.0	95.9	75.3	81.6	78.8	86.4	82.5	87.7
FreqNet [59]	91.9	99.0	69.8	77.5	65.7	71.8	76.0	83.0
FreDect [18]	82.8	90.3	49.9	49.9	42.6	45.6	58.8	62.1
CNNSpot [62]	86.8	95.7	52.6	63.0	52.4	56.8	63.9	72.1
UnivFD [43]	**94.4**	**99.6**	64.8	83.9	61.8	74.6	73.8	86.4
SIDA [25]	66.6	74.2	53.1	59.6	72.9	79.4	63.3	70.1
AEROBLADE [48]	45.7	44.3	84.5	92.7	71.8	81.9	67.9	73.5
MIBD [7]	73.5	82.5	65.3	75.3	54.8	61.9	65.0	73.9
ZeroFake [53]	50.4	49.1	80.4	87.9	52.0	54.7	62.3	65.5
Ours	89.9	97.4	**89.6**	**97.3**	**92.9**	**97.8**	**90.6**	**97.5**

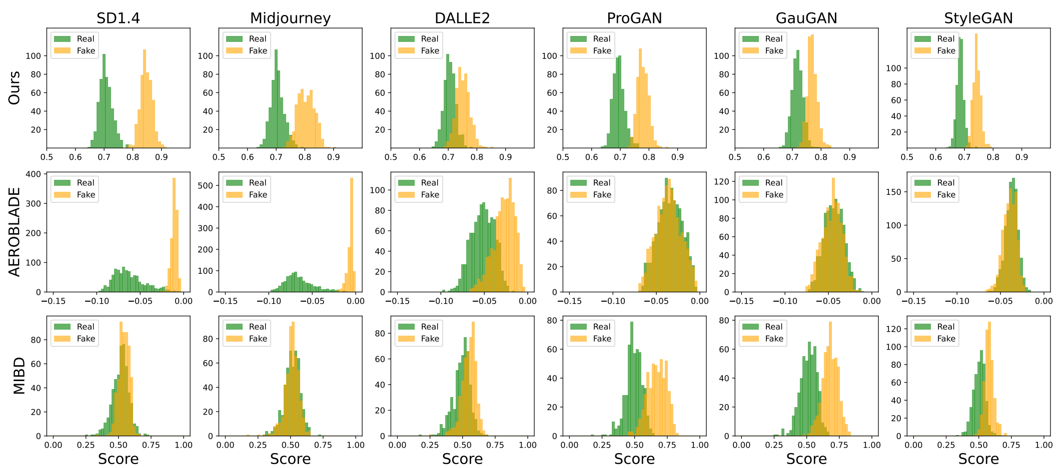

Figure 6: Similarity score distribution for different datasets. We count the number of samples of real and generated images in different score intervals.

Robustness to Perturbations. Social media platforms tend to apply post-processing to user-uploaded images. To evaluate the robustness of our method under such conditions, we consider four common types of perturbations including Gaussian blur, center cropping, JPEG compression and resizing. We investigate the robustness of our method compared with other representative approaches under these perturbations. Experiments are conducted on the ForenSynths and GenImage datasets, with each type of perturbation applied at five different intensity levels. We report the overall AP across all settings in fig. 5. Detailed results can be found in appendix A.4.

It can be observed that NPR [60], which relies heavily on high-frequency features in images, suffers significant performance degradation under common perturbations. In particular, JPEG compression, which tends to suppress high-frequency information, reduces its detection accuracy to near-random performance. While UnivFD [43] suffers a significant performance degrade when countering blurring perturbations. Moreover, compared to cropping and resizing, our method is more affected by blurring and JPEG compression. We hypothesize that this is because these two types of corrupted image distributions deviate more significantly from training datasets of diffusion models, making it difficult for them to generate accurate predictions. However, our method still achieve the optimal average performance.

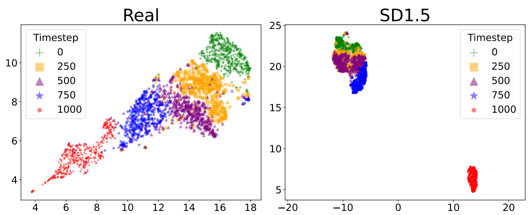

Figure 7: Feature visualization of denoising outputs at different timesteps.

Table 5: Ablation study. Detection performance of our method with different vision foundation models and features from different layers. Measured in AP (%).

Model	Layer	ForenSynths	GenImage	New Generator	AVG
ViT-B/32	all	82.0	98.2	**98.6**	92.9
Vit-H/14	all	96.4	**99.2**	97.9	**97.9**
DINOv2	all	56.9	63.9	71.5	63.7
DINOv3	all	85.4	89.8	98.0	90.7
ViT-L/14	last	87.6	71.2	91.3	82.4
	all	**97.4**	97.3	97.8	97.5

4.3 Visualization

To verify the effectiveness of our method, we statistically analyzed the similarity scores of our method on different generative models. We also visualized the distribution of scores from two other zero-shot methods, i.e., AEROBLADE and MIBD. As shown in fig. 6, our method demonstrates better separability across different generative models, whereas AEROBLADE and MIBD only show effectiveness on one paradigm of generator. For instance, although AEROBLADE achieves promising results on diffusion-generated images, it fails to distinguish GAN-generated images from real ones. In contrast, MIBD performs better on GAN-generated images but struggles with diffusion models.Furthermore, we observed that the optimal decision threshold for our method remains consistently close to 0.75 across different datasets. This is a desirable property, as it implies that a unified threshold can be used to detect various generative models.

To further validate our previous conclusion that the denoising outputs of generated images converge more rapidly toward the target images, we visualize the features along the denoising trajectories. Specifically, we use samples generated by SD1.5 from the GenImage dataset along with real samples. For each image, we extract the CLIP embeddings of the denoised target images at different denoising time steps and visualize them using UMAP [39], which preserves both global and local structure. The visualization results are shown in fig. 7, the features of images generated by SD1.5 quickly become close to their final denoised states as early as step 750. In contrast, the features of real images change more gradually over time. This indicates that the denoising outputs of generated images converge to final images more quickly.

4.4 Ablation Study

Importance of Fine-Grained Features. We evaluate the impact of incorporating features extracted from intermediate layers of the CLIP model on the performance of our method. Specifically, we assess the detection performance when using features from the last layer and from all layers of CLIP, respectively. It can be seen in table 5, incorporating fine-grained features from intermediate layers can significantly improve the detection performance. In addition, we report detailed detection performance of each layer in appendix appendix A.5.

Effect of Vision Foundation Model. We also report the effect of using different vision foundation models in table 5. Firstly, we employed CLIP models with different architectures and evaluated their performance. It can be observed that the smallest model, i.e., ViT-B/32, performs significantly worse than the others on the ForenSynths, while the ViT-H/14 achieves slight better results with the ViT-L/14. This indicates that our method can benefit from larger models. Note that, consistent with UnivFD [43], we report our previous results using the CLIP ViT-L/14 to ensure fairness. In addition, we also extract features using DINOv2 and DINOv3, two self-supervised vision foundation models. DINOv2 performs significantly worse than the other models. We hypothesize that this is because its training paradigm encourages the extraction of more robust features, making it less sensitive to fine-grained variations. In contrast, using DINOv3 as the feature extractor brings stronger performance as its ability to capture fine-grained features is enhanced.

Table 6: Performance of our method across different diffusion models and timesteps. Measured in AP(%)

Diffusion Model	ForenSynths	GenImage	New Generator	AVG
SD1.5	97.4	**97.3**	**97.8**	**97.5**
SD2.1	92.5	92.7	95.9	93.6
FLUX	84.8	88.3	81.9	85.3
SD1.5-10steps	**97.5**	96.9	97.1	97.2

Different Diffusion Models and Timesteps. To verify the impact of different diffusion models on the performance of our method, we also evaluate our method with Stable Diffusion v2.1 (SD2.1) and FLUX, considering their open-source availability and popularity. Note that although FLUX was proposed as a Flow Matching model, we still treat it as a diffusion model in this context. Results in table 6 show that SD2.1 and FLUX achieve lower detection performance compared to SD1.5, with the FLUX based detector performing the worst. We speculate that this may be because FLUX can accurately predict the original image at the early stages of the generation process, which affects the progressive generation behavior of the denoising trajectory and partially disrupts the bias between real and fake images. Nevertheless, FLUX based detector still achieve promising results. Moreover, we test using fewer timesteps, 10 steps, and observe that our method can achieve similar results, indicating that it is not significantly suffered by detection efficiency issues.

5 Conclusions and Discussions

In this paper, we introduce a novel zero-shot method for detecting AI-generated images. Our approach is motivated by the observation that generated images exhibit faster convergence toward the target image along their denoising trajectories. To detect synthetic images, we perform DDIM inversion on the input image, collect intermediate denoising outputs, and compute their similarity to the original image—where generated images are expected to exhibit higher similarity. Since our method does not rely on generated images for training, it avoids overfitting to specific generative models or datasets and demonstrates strong generalization capabilities. We believe our work will inspire future research to further explore informative features embedded in the image generation process itself.

Limitations. Although our method demonstrates strong generalization capabilities, its performance may degrade when applied to severely corrupted images, such as those that have undergone heavy compression or significant blurring. This may be primarily due to such images deviate substantially from the data distribution learned by diffusion models, making it difficult for the model to accurately predict the original image. Currently, we compute the final score by directly averaging the similarity scores obtained from different timesteps and CLIP layers. In the future, we plan to develop an adaptive weighting scheme to assign importance dynamically, which is expected to further improve detection performance.

Broader Impacts. Our work aims to combat misinformation and enhance the credibility of content on social media platforms. The generation and detection of synthetic images form a long-term adversarial game. As generative models continue to evolve and improve, it becomes essential to incorporate diverse approaches for effective detection. We hope our work will inspire further research into discovering and leveraging new forensic cues for identifying AI-generated images.

6 Acknowledgement

This work has been supported by the Youth Innovation Promotion Association, CAS (No.2021155).

References

[1] Midjourney. `https://www.midjourney.com/home/`, 2025.

[2] Wukong. `https://xihe.mindspore.cn/modelzoo/wukong`, 2025.

[3] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. PMLR, 2017.

[4] Quentin Bammey. Synthbuster: Towards detection of diffusion model generated images. *IEEE Open Journal of Signal Processing*, 5:1–9, 2023.

[5] James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, et al. Improving image generation with better captions. *Computer Science. https://cdn. openai. com/papers/dall-e-3. pdf*, 2(3):8, 2023.

[6] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, 2018.

[7] Jonathan Brokman, Amit Giloni, Omer Hofman, Roman Vainshtein, Hisashi Kojima, and Guy Gilboa. Manifold induced biases for zero-shot and few-shot detection of generated images. In *The Thirteenth International Conference on Learning Representations*, 2025.

[8] George Cazenavette, Avneesh Sud, Thomas Leung, and Ben Usman. Fakeinversion: Learning to detect images from unseen text-to-image models by inverting stable diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10759–10769, 2024.

[9] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8789–8797, 2018.

[10] Davide Cozzolino, Giovanni Poggi, Matthias Nießner, and Luisa Verdoliva. Zero-shot detection of ai-generated images. In *European Conference on Computer Vision*, pages 54–72. Springer, 2024.

[11] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

[12] Sander Dieleman. Diffusion is spectral autoregression, 2024.

[13] Ricard Durall, Margret Keuper, and Janis Keuper. Watch your up-convolution: Cnn based generative deep neural networks are failing to reproduce spectral distributions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7890–7899, 2020.

[14] Tarik Dzanic, Karan Shah, and Freddie Witherden. Fourier spectrum discrepancies in deep network generated images. *Advances in neural information processing systems*, 33:3022–3032, 2020.

[15] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

[16] Fabian Falck, Teodora Pandeva, Kiarash Zahirnia, Rachel Lawrence, Richard Turner, Edward Meeds, Javier Zazo, and Sushrut Karmalkar. A fourier space perspective on diffusion models. *arXiv preprint arXiv:2505.11278*, 2025.

[17] Jean-Yves Franceschi, Mike Gartrell, Ludovic Dos Santos, Thibaut Issenhuth, Emmanuel de Bézenac, Mickaël Chen, and Alain Rakotomamonjy. Unifying gans and score-based diffusion as generative particle models. *Advances in Neural Information Processing Systems*, 36:59729–59760, 2023.

[18] Joel Frank, Thorsten Eisenhofer, Lea Schönherr, Asja Fischer, Dorothea Kolossa, and Thorsten Holz. Leveraging frequency analysis for deep fake image recognition. In *International conference on machine learning*, pages 3247–3258. PMLR, 2020.

[19] Yuan Gao, Yuling Jiao, Yang Wang, Yao Wang, Can Yang, and Shunkang Zhang. Deep generative learning via variational gradient flow. In *International Conference on Machine Learning*, pages 2093–2101. PMLR, 2019.

[20] Amin Ghiasi, Hamid Kazemi, Eitan Borgnia, Steven Reich, Manli Shu, Micah Goldblum, Andrew Gordon Wilson, and Tom Goldstein. What do vision transformers learn? a visual exploration. *arXiv preprint arXiv:2212.06727*, 2022.

[21] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[22] Shuyang Gu, Dong Chen, Jianmin Bao, Fang Wen, Bo Zhang, Dongdong Chen, Lu Yuan, and Baining Guo. Vector quantized diffusion model for text-to-image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10696–10706, 2022.

[23] Zhiyuan He, Pin-Yu Chen, and Tsung-Yi Ho. Rigid: A training-free and model-agnostic framework for robust ai-generated image detection. *arXiv preprint arXiv:2405.20112*, 2024.

[24] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[25] Zhenglin Huang, Jinwei Hu, Xiangtai Li, Yiwei He, Xingyu Zhao, Bei Peng, Baoyuan Wu, Xiaowei Huang, and Guangliang Cheng. Sida: Social media image deepfake detection, localization and explanation with large multimodal model. *arXiv preprint arXiv:2412.04292*, 2024.

[26] Dongsheng Jiang, Yuchen Liu, Songlin Liu, Jin'e Zhao, Hao Zhang, Zhen Gao, Xiaopeng Zhang, Jin Li, and Hongkai Xiong. From clip to dino: Visual encoders shout in multi-modal large language models. *arXiv preprint arXiv:2310.08825*, 2023.

[27] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *International Conference on Learning Representations*, 2018.

[28] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019.

[29] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8110–8119, 2020.

[30] Black Forest Labs. Flux. `https://github.com/black-forest-labs/flux`, 2024.

[31] Xin Lai, Zhuotao Tian, Yukang Chen, Yanwei Li, Yuhui Yuan, Shu Liu, and Jiaya Jia. Lisa: Reasoning segmentation via large language model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9579–9589, 2024.

[32] Ouxiang Li, Jiayin Cai, Yanbin Hao, Xiaolong Jiang, Yao Hu, and Fuli Feng. Improving synthetic image detection towards generalization: An image transformation perspective. *arXiv preprint arXiv:2408.06741*, 2024.

[33] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer vision–ECCV 2014: 13th European conference, zurich, Switzerland, September 6-12, 2014, proceedings, part v 13*, pages 740–755. Springer, 2014.

[34] Bo Liu, Fan Yang, Xiuli Bi, Bin Xiao, Weisheng Li, and Xinbo Gao. Detecting generated images by real images. In *European Conference on Computer Vision*, pages 95–110. Springer, 2022.

[35] Huan Liu, Zichang Tan, Chuangchuang Tan, Yunchao Wei, Jingdong Wang, and Yao Zhao. Forgery-aware adaptive transformer for generalizable synthetic image detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10770–10780, 2024.

[36] Zihan Liu, Hanyi Wang, Yaoyu Kang, and Shilin Wang. Mixture of low-rank experts for transferable ai-generated image detection. *arXiv preprint arXiv:2404.04883*, 2024.

[37] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pages 3730–3738, 2015.

[38] Ruipeng Ma, Jinhao Duan, Fei Kong, Xiaoshuang Shi, and Kaidi Xu. Exposing the fake: Effective diffusion-generated images detection. *arXiv preprint arXiv:2307.06272*, 2023.

[39] Leland McInnes, John Healy, Nathaniel Saul, and Lukas Großberger. Umap: Uniform manifold approximation and projection. *Journal of Open Source Software*, 3(29):861, 2018.

[40] Tai D Nguyen, Aref Azizpour, and Matthew C Stamm. Forensic self-descriptions are all you need for zero-shot detection, open-set source attribution, and clustering of ai-generated images. *arXiv preprint arXiv:2503.21003*, 2025.

[41] Alexander Quinn Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob Mcgrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. In *International Conference on Machine Learning*, pages 16784–16804. PMLR, 2022.

[42] Sophie J Nightingale and Hany Farid. Ai-synthesized faces are indistinguishable from real faces and more trustworthy. *Proceedings of the National Academy of Sciences*, 119(8):e2120481119, 2022.

[43] Utkarsh Ojha, Yuheng Li, and Yong Jae Lee. Towards universal fake image detectors that generalize across generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24480–24489, 2023.

[44] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2337–2346, 2019.

[45] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. SDXL: Improving latent diffusion models for high-resolution image synthesis. In *The Twelfth International Conference on Learning Representations*, 2024.

[46] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[47] Jonas Ricker, Simon Damm, Thorsten Holz, and Asja Fischer. Towards the detection of diffusion model deepfakes. *arXiv preprint arXiv:2210.14571*, 2022.

[48] Jonas Ricker, Denis Lukovnikov, and Asja Fischer. Aeroblade: Training-free detection of latent diffusion images using autoencoder reconstruction error. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9130–9140, 2024.

[49] Severi Rissanen, Markus Heinonen, and Arno Solin. Generative modelling with inverse heat dissipation. *arXiv preprint arXiv:2206.13397*, 2022.

[50] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[51] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[52] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115:211–252, 2015.

[53] Zeyang Sha, Yicong Tan, Mingjie Li, Michael Backes, and Yang Zhang. Zerofake: Zero-shot detection of fake images generated and edited by text-to-image generation models. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, pages 4852–4866, 2024.

[54] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. pmlr, 2015.

[55] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

[56] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.

[57] Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. *Advances in neural information processing systems*, 33:12438–12448, 2020.

[58] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021.

[59] Chuangchuang Tan, Yao Zhao, Shikui Wei, Guanghua Gu, Ping Liu, and Yunchao Wei. Frequency-aware deepfake detection: Improving generalizability through frequency space domain learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 5052–5060, 2024.

[60] Chuangchuang Tan, Yao Zhao, Shikui Wei, Guanghua Gu, Ping Liu, and Yunchao Wei. 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*, pages 28130–28139, 2024.

[61] Yael Vinker, Ehsan Pajouheshgar, Jessica Y Bo, Roman Christian Bachmann, Amit Haim Bermano, Daniel Cohen-Or, Amir Zamir, and Ariel Shamir. Clipasso: Semantically-aware object sketching. *ACM Transactions on Graphics (TOG)*, 41(4):1–11, 2022.

[62] Sheng-Yu Wang, Oliver Wang, Richard Zhang, Andrew Owens, and Alexei A Efros. Cnn-generated images are surprisingly easy to spot... for now. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8695–8704, 2020.

[63] Zhendong Wang, Jianmin Bao, Wengang Zhou, Weilun Wang, Hezhen Hu, Hong Chen, and Houqiang Li. Dire for diffusion-generated image detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22445–22455, 2023.

[64] Zhenting Wang, Chen Chen, Yi Zeng, Lingjuan Lyu, and Shiqing Ma. Where did i come from? origin attribution of ai-generated images. *Advances in neural information processing systems*, 36:74478–74500, 2023.

[65] Mingxuan Yi, Zhanxing Zhu, and Song Liu. Monoflow: Rethinking divergence gans via the perspective of wasserstein gradient flows. In *International Conference on Machine Learning*, pages 39984–40000. PMLR, 2023.

[66] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.

[67] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3836–3847, 2023.

[68] Nan Zhong, Yiran Xu, Sheng Li, Zhenxing Qian, and Xinpeng Zhang. Patchcraft: Exploring texture patch for efficient ai-generated image detection. *arXiv preprint arXiv:2311.12397*, 2023.

[69] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.

[70] Mingjian Zhu, Hanting Chen, Qiangyu Yan, Xudong Huang, Guanyu Lin, Wei Li, Zhijun Tu, Hailin Hu, Jie Hu, and Yunhe Wang. Genimage: A million-scale benchmark for detecting ai-generated image. *Advances in Neural Information Processing Systems*, 36:77771–77782, 2023.

A Appendix

A.1 Compared Baselines

Supervised Detectors. 1) **FreDect** [18]: a detector trained on the DCT space. 2) **FreqNet** [59]: it extracts high-frequency information by performing Fourier transforms in both the pixel space and the feature space. 3) **NPR** [60]: it resamples images by first downscaling and then upscaling them, and computes the residuals with respect to original images to capture local pixel relationships. Since this approach captures low-level features similar to the high-frequency components of images, we also categorize it as a frequency-based method. 4) **CNNSpot** [62]: a detector fine-tuning on pretrained ResNet. 5) **UnivFD** [43] this method uses the CLIP model to project images into a unified feature space, followed by a single linear layer for classification. 6) **SIDA** [25]: it fine-tunes a pretrained large multimodal model using custom dataset annotated with tampered labels. The trained model can provide detailed information related to fake contents. To compute the AP for this method, we use the logits before the softmax outputs as probabilities for real and fake.

Zero-Shot Detectors. 1) **AEROBLADE** [48]: This method identifies images generated by latent diffusion models by reconstructing the original image using a VAE. 2) **ZeroFake** [53]: This method performs DDIM inversion on the image, modifies the prompt during reconstruction, and uses the SSIM between the reconstructed image and the original to identify fake images. 3) **MIBD** [7]: It perturbs the image with noise and uses the similarity between original images and the noises predicted by a diffusion model as the detection criteria values.

A.2 Datasets

ForenSynths. The test set contains images generated by 7 types of GAN, namely ProGAN [27], GauGAN [44], BigGAN [6], StarGAN [9], CycleGAN [69], StyleGAN [28], and StyleGAN2 [29].

GenImage. It includes 8 diffusion models: Stable Diffusion V1.4 [50], Stable Diffusion V.15 [50], ADM [11], DALLE2, Midjourney [1], Glide [41], VQDM [22], and Wukong [2].

New Generator. We download the test set of COCO2017, and use generative models to produce corresponding fake images based on the prompts of each real image, thereby avoiding semantic bias. The new generators include Stable Diffusion v3 [15], FLUX [30], Stable Diffusion XL [45], DALLE3 [5], Firefly and Midjourney-v5 [1]. We present some examples in fig. 8.

Real
SD3
FLUX

Figure 8: Examples of real images and corresponding generated images.

A.3 Examples of Denoising Outputs

We perform DDIM inversion on the input image to simulate its generative process, and collect the denoised outputs at each timestep to compute their similarity with the original image. In fig. 9,

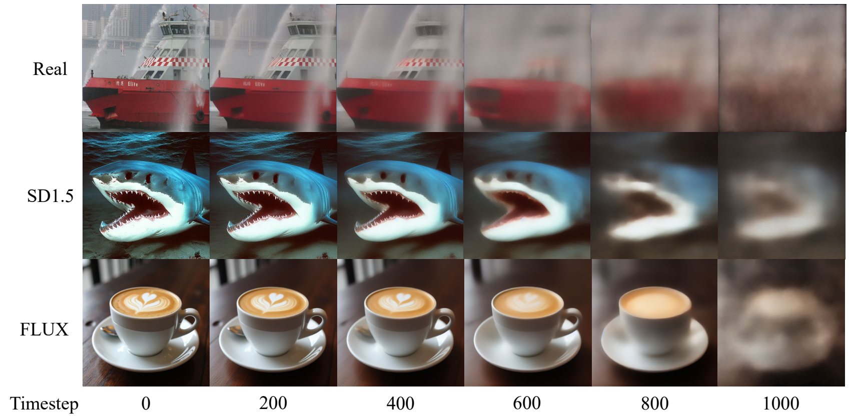

Figure 9: Examples of denoising outputs at different timesteps.

we present some examples of predicted images along the denoising trajectories for both real and generated images.

A.4 Robustness Experiments

Table 7: Average robustness to common perturbations. We report the average AP (%) scores of different methods over each type of perturbations at all intensity levels.

Method	Clean	Blur	Crop	JPEG	Resize	AVG
NPR [60]	88.4	78.9	89.5	57.9	88.3	80.6
UnivFD [43]	91.3	73.2	94.8	78.5	89.0	85.4
MIBD [7]	78.7	69.8	79.6	72.3	76.8	75.4
Ours	97.3	81.3	98.1	80.7	95.5	90.6

To evaluate the robustness of our method, we apply four common types of perturbations typically encountered on social media platforms, each with five levels of intensity. We compare our approach with state-of-the-art methods from each category, namely NPR [60], UnivFD [43] and MIBD [7]. In table 7, we report the average AP of each method under different types of perturbations.

A.5 Ablation Study on CLIP Layer

To further verify the importance of the intermediate features of CLIP, we extract the features from each layer of CLIP (ViT-L/14) as image embeddings and test the detection performance of our method. As shown in fig. 10, the features extracted from the 15th intermediate layer achieve the best performance, and the results from these middle layers are significantly better than those from the early and late layers. Moreover, the variation in detection performance across different layers is closely correlated with the curves presented in fig. 4.

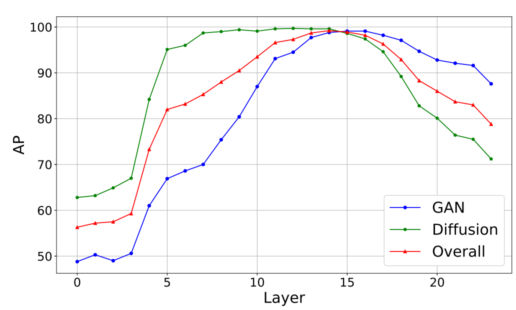

Figure 10: Detection performance of different layer.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We have faithfully describe our contributions in the abstract and introduction.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We have discussed the limitations of our method in the conclusion section.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: We provide clear descriptions or references of the relevant theoretical assumptions.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We have provided a thorough description of the settings of our method, including the models, datasets and implementation details.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We have provided the project demo in the supplementary material, and we will open the source code once our paper gets published.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We specify the key configurations about the implementation details in the experiment section.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Error bars are not reported because it would be too computationally expensive.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [No]

   Justification: We provide the type of GPU used in our experiments. While we do not report the execution time for each experiment, as this metric is not particularly relevant to our task.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Our researches conducted in this paper conform with NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss the potential societal impacts of this work in the conclusion section.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: Our research poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We have cited the original papers that produced the code packages or datasets. And we also state the versions of the assets are used.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: We do not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: Our paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: Our paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: The core method development in this research does not involve LLMs as any important components.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.