One for All: Synthesis-Free Fingerprint Learning for Attribution of In-the-Wild Synthetic Images

Jianwei Fei[1]*, Yunshu Dai[2], Peipeng Yu[3], Zhihua Xia[3], Dasara Shullani[1], Daniele Baracchi[1], Alessandro Piva[1]

[1]Department of Information Engineering, University of Florence
[2]School of Cyber Science and Technology, Sun Yat-sen University
[3]College of Cyberspace Security, Jinan University

Abstract

Attributing synthetic images to their source generative models is critical for digital forensics and security. While most existing attribution methods can distinguish images produced by known models and reject those from unknown ones, they are unable to verify whether a given image was produced by a specific, previously unseen model. To address this limitation, we formulate an open-set verification problem: determining whether a given image was generated by a specific model. Our key insight is that synthetic images from different models show consistent, content-independent fingerprints in their amplitude spectrum. Based on this insight, we design a dynamic fingerprint simulator capable of simulating over 1.6 trillion generative model architectures. We further train an extractor to capture model-specific fingerprint representations with supervised contrastive learning, enabling accurate attribution of synthetic images, even from previously unseen models. Our method does not rely on any synthetic images, instead, it is trained solely on real images. On DMDetection and AIGCBenchmark, which comprises dozens of state-of-the-art and in-the-wild generative models, our method improves the attribution performance (AUC) of the prior method from random level to 94.05% and 83.05%, respectively. On GenImage and OSMA benchmarks, we obtain 85.08% and 88.48% OSCR, outperforming the SOTA methods by 4.30% and 9.37% under the same settings.

Code — https://github.com/jumpycat/OFAAttribution

Introduction

Recent advances in generative artificial intelligence have enabled the synthesis of photorealistic imagery through models such as DALLE2 (Ramesh et al. 2022) and Midjourney (Midjourney, Inc. 2023). The remarkable fidelity, growing diversity, and widespread accessibility of these models have raised serious concerns about visual media security. Synthetic image detection has thus attracted considerable research interest (Fei et al. 2022a; Luo et al. 2023; Kong et al. 2022, 2021; Yu et al. 2025; Bindini et al. 2024; Du et al. 2025). However, beyond detection, identifying the specific generative model responsible for producing a given image, i.e., synthetic image attribution, is also an important task. It is essential for many applications, including intellectual property protection (Fei et al. 2025a,c), where content creators need to establish ownership of generated works, and forensic tasks requiring identification of the generative models used for malicious image production (Dai, Fei, and Huang 2024; Dai et al. 2025).

Currently, synthetic image attribution includes proactive and passive methods. Proactive methods incorporate controlled modifications during generation, such as embedding imperceptible watermarks into the synthesized images (Fei et al. 2022b; Fernandez et al. 2023; Fei et al. 2025b). Conversely, passive methods identify the origins of synthetic images through post-hoc analysis, without interacting with or modifying the generation pipeline (Yang et al. 2022, 2023; Abady et al. 2024). A significant challenge for passive methods is the open-set recognition problem. Unlike closed-set scenarios, where both training and testing images are produced by known models, open-set attribution requires identifying images produced by unknown models. This challenge is increasingly acute due to the rapid evolution of generative models, where new architectures and training strategies continuously emerge. Prior open-set synthetic image attribution focuses on attributing query images to a specific model in the closed set, or issuing rejection decisions for images produced by open-set models. However, this paradigm has two main limitations: **(1) it cannot directly verify whether a query image is produced by a particular target model that belongs to the open set**; **(2) it requires incorporating the target model into the closed set and re-training the system to gain the ability to verify images produced by that model, which leads to poor scalability.**

To this end, we formulate a practical open-set attribution problem: determining whether a given image is produced by a specific model, regardless of whether that model was seen during training. This is motivated by practical scenarios, where generative models are continually evolving and cannot be exhaustively enumerated. Specifically, we propose a framework that learns to extract distinctive model-specific artifacts, i.e., fingerprints that enable image-to-model attribution. The core of our framework lies in learning discriminative and generalizable model fingerprints from synthetic images. To achieve this, we introduce a stochastic image reconstruction encoder-decoder with variable architecture that functions as a generative model fingerprint simulator, capa-

*Corresponding author (fei_jianwei@163.com).

ble of modeling over 1.6 trillion distinct architectures. Each time a real image is reconstructed through a different architecture, it is injected with a different fingerprint. We then develop a fingerprint extractor trained using supervised contrastive learning on real images reconstructed by the simulator. The learned fingerprint representations capture subtle yet distinctive artifacts that are inherently introduced by generative models during image synthesis. In parallel, we introduce an attribution head jointly trained with the extractor, which leverages these fingerprint representations to perform pairwise attribution, predicting whether two images are produced by the same generative model.

Our method has several key advantages over existing works: (1) **Synthesis-free**. This mitigates overfitting to any specific models and avoids the need to collect large-scale, diverse synthetic datasets; (2) **Generalizability**. Due to the nearly infinite number of simulated generative architectures, the extractor learns discriminative and generalizable fingerprint representations that are useful for multiple attribution tasks; (3) **Scalability**. Our framework can handle arbitrary new generative models without retraining.

We conduct extensive experiments across multiple challenging benchmarks that contain images from dozens of state-of-the-art (SOTA) and commercial generative models. Our method improves the attribution performances of the comparable method from near-random to 94.05% and 83.05% AUC on DMDetection (Corvi et al. 2023) and AIGCBenchmark (Zhong et al. 2023). In addition, for fair comparisons with prior open-set synthetic image attribution methods, we used an extra attribution head for closed-set and open-set attribution. Our method outperforms existing SOTA methods across different settings. Specifically, it achieves OSCR scores of 88.48% on OSMA (Yang et al. 2023), outperforming the SOTA method by 9.37%. On GenImage (Zhu et al. 2023), our method also achieves an OSCR of 85.08%, surpassing the SOTA method by 4.30%. The main contributions of this paper are:

- We introduce a stochastic image reconstruction encoder-decoder capable of modeling over 1.6 trillion diverse generative models, serving as a universal simulator to inject generative fingerprints into real images.
- We introduce a synthesis-free framework for fingerprint learning, bypassing the need for large-scale synthetic datasets, and enhancing generalization to unseen models.
- We achieve SOTA performance across multiple benchmarks, demonstrating the effectiveness and scalability of our method in real-world attribution scenarios.

Related Work

Synthetic Image Attribution

Synthetic image attribution methods include proactive and passive approaches. Proactive methods inject artificial fingerprints into synthetic images during the training process (Yu et al. 2021; Fernandez et al. 2023; Fei et al. 2024). Passive methods rely on intrinsic artifacts left by generative models in their outputs and can be further categorized into model-level and architecture-level approaches.

Model-level attribution refers to the task of identifying the exact generative model responsible for producing a given synthetic image. Early pioneering work by Marra et al. (Marra et al. 2019) first revealed that GANs leave specific fingerprints in their synthetic images, and used the averaged noise residual as model fingerprints. Yu et al. (Yu, Davis, and Fritz 2019) replaced hand-crafted fingerprint extraction with learning-based approaches to decouple GAN fingerprints into model-specific artifacts and image content. Joslin et al. (Joslin and Hao 2020) proposed to analyze features in frequency space to capture model-specific artifacts for model attribution in transformed domains.

Existing open-set attribution methods have primarily focused on extending closed-set classifiers with rejection mechanisms. Wang et al. (Wang et al. 2023) developed rejection-based classifiers using a vision transformer with MLS thresholding, while Fang et al. (Fang, Nguyen, and Stamm 2023) proposed distance-based rejection mechanisms that compare test samples to class centroids in feature space. Yang et al. (Yang et al. 2023) introduced Progressive Open Space Expansion (POSE), which progressively involves augmentation models to simulate diverse open-set fingerprints. More recently, Abady et al. (Abady et al. 2024) proposed a Siamese network-based verification framework capable of determining whether two images are produced by the same model. Recent advances have extended attribution beyond GANs to diffusion-based models. For instance, Cioni et al. (Cioni et al. 2024) proposed using pre-trained features from foundation models like CLIP for better generalization across both GAN and diffusion models.

Architecture-level attribution goes beyond model-level attribution by attributing images to underlying network architectures, regardless of specific training configurations or dataset variations. Representative works include DNA-Det (Yang et al. 2022) and RepMix (Bui, Yu, and Collomosse 2022), which focus on identifying architectures rather than specific model instances.

Challenges and Limitations

While existing methods can distinguish between known and unknown classes, they cannot verify whether a specific image was produced by a target model. While incorporating newly emerged models into the closed set is one workaround, it is impractical due to the rapid development of generative models. We address this gap by reframing open-set attribution as a verification task, enabling image-to-model attribution even for unknown in-the-wild models, without requiring retraining as new models emerge.

Methodology

Definition of Generative Model Fingerprint

As introduced in Sec. II.A, the fingerprint of a generative model is defined as a pattern naturally contained in the synthetic image that is related to the source model, yet independent of the image semantics. Recent studies have identified distinctive patterns in the amplitude spectrum of synthetic images, which are considered reliable indicators of fingerprints (Frank et al. 2020), as shown in Fig. 1. Notably, exist-

ing studies suggest that fingerprints are primarily shaped by model architectures, while also being affected by the model weights (Corvi et al. 2023). Even with a fixed architecture, variations in training data can induce changes in the learned weights, leading to differences in the fingerprints.

Building upon this idea, we regard the amplitude spectrum of synthetic images as a carrier of model fingerprints and extract feature vectors from it to represent the fingerprints of the generative models. We propose a two-stage framework that learns to extract fingerprints without synthetic training data. In the first stage, we design a stochastic encoder-decoder that simulates diverse generative models. In the second stage, we train a universal fingerprint extractor via supervised contrastive learning using real images processed by the simulator, with an additional head attached for the attribution task. **It is important to note that the simulator does not aim to replicate the exact fingerprints of real-world models. Instead, it functions as a data synthesis framework that supplies supervisory signals, enabling the fingerprint extractor to learn discriminative and generalizable model-specific fingerprints.**

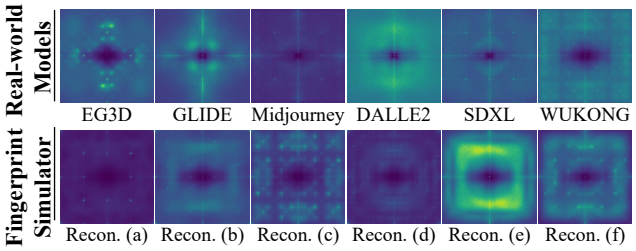

Figure 1: Amplitude spectrum of images produced by various real-world models and of real images reconstructed using our fingerprint simulator.

Training a Stochastic Fingerprint Simulator

Given that generative fingerprints are dependent on model architectures and different architectures are expected to have distinguishable fingerprints, we train an image reconstruction autoencoder with a stochastic decoder as the fingerprint simulator to simulate diverse fingerprints produced by different architectures. Formally, the simulator consists of a fixed encoder $\mathcal{E}$ and the stochastic decoder $\mathcal{D}$ that employs dynamic path selection and mixture-of-kernels convolution. $\mathcal{D}$ dynamically samples different paths during each forward pass to reconstruct the input image.

The encoder $\mathcal{E}$ adopts a hierarchical architecture that maps an input image $\boldsymbol{x} \in \mathbb{R}^{3\times H\times W}$ to a set of features:

$$\mathcal{E} : \mathbb{R}^{3\times H\times W} \rightarrow \{\mathbf{F}_i \in \mathbb{R}^{C_i\times H_i\times W_i}\}_{i=1}^{L} \tag{1}$$

where $H_i = H/2^i$ and $W_i = W/2^i$ denote the spatial resolution at scale level i, and C_i is the number of channels.

The decoder $\mathcal{D}$ mirrors the encoder structure with L hierarchical upsampling stages but introduces a stochastic strategy. Specifically, in each forward pass, a starting scale s is randomly sampled, i.e., $s \sim \text{Uniform}(L, \dots, 1)$. $\mathcal{D}$ then reconstructs the output image by progressively decoding feature map $\mathbf{F}^s$ to the full resolution (i.e., scale level 1). Note that this process does not include any skip connections from the $\mathcal{E}$. At each scale level $i \in \{s, \dots, 1\}$, the decoder feature undergoes a $2\times$ upsampling via a Variable Upsampling (VariUp) module, followed by a sequence of $V_i \sim \text{Uniform}(1, \dots, V_{max})$ Variable Struct (VariStruct) cells, where $V_{max} \in \mathbb{N}$ is the maximum number of VariStruct cells per level. In our implementation, $L = 3$ and $V_{max} = 2$.

VariUp is a stochastic upsampling module at the beginning of each decoding level, defined by an operation set composed of multiple upsampling strategies, from which one operation is randomly selected at each forward pass. Formally, $\mathbf{h}_{i,0} = \mathcal{U}_i(\mathbf{F}_i)$, where $\mathcal{U}_i$ is the VariUp module in scale level i, and $\mathbf{h}_{i,0}$ is the resulting upsampled feature fed into the following layers. In our implementation, it includes four upsampling strategies: bilinear, nearest, bicubic, and pixel shuffle (Shi et al. 2016).

VariStruct Cell is the basic building block applied after each upsampling operation in the decoding stage. It transforms upsampled features through a structurally stochastic yet learnable architecture. Let us denote the VariStruct cell by $\mathcal{B}_{i,j}$, where $j \in \{1, \dots, V_i\}$ is the number of cells in scale level i. $\mathcal{B}_{i,j}$ is composed of three operations: a convolution operation $\mathcal{C}_{i,j}$, defined by our proposed Mixture-of-Kernel Convolution (MoKConv), an activation operation $\mathcal{A}_{i,j}$ randomly sampled from a predefined activation pool $\mathcal{A}_{i,j} \sim \text{Uniform}(\mathcal{A})$, and a normalization operation $\mathcal{N}_{i,j}$ randomly sampled from a predefined normalization pool $\mathcal{N}_{i,j} \sim \text{Uniform}(\mathcal{N})$. These operations are applied in a randomly permuted order: $(o^1_{i,j}, o^2_{i,j}, o^3_{i,j}) = \texttt{Permute}(\mathcal{C}_{i,j}, \mathcal{A}_{i,j}, \mathcal{N}_{i,j})$. Therefore, the j-th VariStruct Cell at level i is formally defined as:

$$\mathbf{h}_{i,j} = \mathcal{B}_{i,j}(\mathbf{h}_{i,j-1}) = o^3_{i,j} \circ o^2_{i,j} \circ o^1_{i,j}(\mathbf{h}_{i,j-1}), \tag{2}$$

where $\circ$ denotes function composition. In our implementation, $\mathcal{A}$ includes GELU, SiLU, and LeakyReLU, $\mathcal{N}$ includes Batch Norm, Instance Norm, and Group Norm.

After passing through all decoding levels, $\mathcal{D}$ outputs the reconstructed image $\hat{\boldsymbol{x}} = \mathcal{D}(\mathcal{E}(\boldsymbol{x}))$. In our setting, $\mathcal{D}$ yields over 1.65 trillion possible architectures in total.

Training Objective The stochastic encoder-decoder is trained to minimize an image reconstruction loss $\mathcal{L}_{\text{img}}$ that combines mean squared error (MSE) and learned perceptual image patch similarity (LPIPS) (Zhang et al. 2018):

$$\mathcal{L}_{\text{img}} = \lambda_{\text{mse}} \|\boldsymbol{x} - \hat{\boldsymbol{x}}\|_2^2 + \lambda_{\text{lpips}} \text{LPIPS}(\boldsymbol{x}, \hat{\boldsymbol{x}}), \tag{3}$$

where λ_{mse} and λ_{lpips} are weighting parameters.

Mixture-of-Kernel Convolution (MoKConv) Due to the architectural dynamism, using a fixed convolution instance at each cell can lead to optimization conflicts, as it must simultaneously process input features from diverse architectures. To mitigate this issue and improve the diversity of simulated fingerprints, we propose the *Mixture-of-Kernel Convolution* (MoKConv), which adaptively integrates multiple convolution kernels conditioned on the input.

Formally, each MoKConv maintains a set of K base convolution kernels $\{\mathbf{W}_k\}_{k=1}^{K}$, where each $\mathbf{W}_k$ comprises

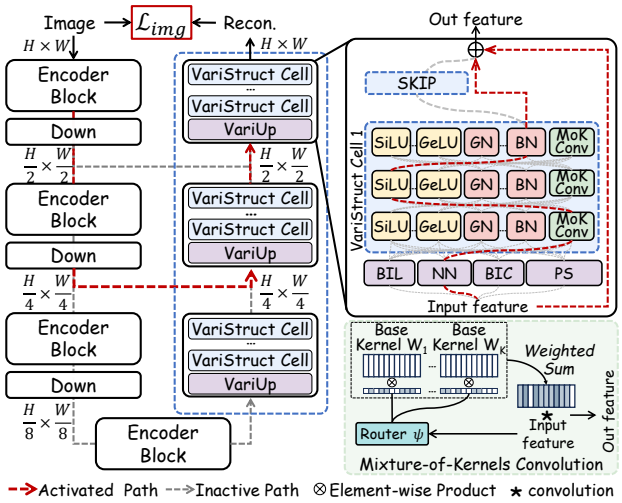

Figure 2: The stochastic fingerprint simulator.

C_{out} filters. Given an input $\boldsymbol{h} \in \mathbb{R}^{C_{\text{in}} \times h \times w}$, a router function $\psi : \mathbb{R}^{C_{\text{in}} \times h \times w} \to \mathbb{R}^D$ extracts a compact descriptor $\psi(\boldsymbol{h})$. This descriptor is then projected into a set of coefficients $\boldsymbol{w} \in \mathbb{R}^{K \times C_{\text{out}}}$ via a linear projection matrix $\mathbf{P} \in \mathbb{R}^{D \times (K \cdot C_{\text{out}})}$, i.e.,$\boldsymbol{w} = \psi(\boldsymbol{h})\mathbf{P}$, where the output is reshaped to (K, C_{out}). Then the MoKConv output is computed as a weighted sum over all base kernels.

$$\text{MoKConv}(\boldsymbol{h}) = (\sum_{k=1}^{K} \boldsymbol{w}_k \odot \mathbf{W}_k) * \boldsymbol{h}, \tag{4}$$

where $*$ denotes convolution and $\odot$ denotes multiplication along the filter dimension of $\mathbf{W}^k$.

Training a Generative Fingerprint Extractor

We analogize learning generative model fingerprints to training face recognition models. Both tasks require distinguishing patterns across a large number of classes, where samples within each class share common features (identity information vs. model fingerprints). In our case, this demands a diverse and extensive set of model architectures to capture discriminative fingerprints. Our fingerprint simulator addresses this challenge, making effective fingerprint learning both feasible and straightforward.

Motivation and Architectural Design As shown in Fig. 3, in each step of the training pipeline, we sample real images and reconstruct them using the fingerprint simulator, resulting in images embedded with diverse model fingerprints. We then extract the frequency spectrum following the pipeline in (Corvi et al. 2023), which serves as the input to our fingerprint extractor. The core challenge then lies in learning discriminative fingerprints. Inspired by training face recognition models, we adopt a supervised contrastive loss to encourage fingerprint similarity among images produced by the same model, while maximizing the separation between fingerprints from different models. Given an reconstructed image $\hat{\boldsymbol{x}}$ as input, we first apply a pretrained denoiser $\mathcal{D}_\sigma$ (Corvi et al. 2023) to estimate its noise residual, i.e., $\boldsymbol{r} = \mathcal{D}_\sigma(\hat{\boldsymbol{x}})$. We then transform the residual using 2D Fourier Transform and compute the log-magnitude as $\mathcal{T}(\boldsymbol{r}) = \log(1 + |\text{FFT}_{2D}(\boldsymbol{r})|)$, where $|\cdot|$ denotes the element-wise absolute value. $\mathcal{T}(\boldsymbol{r})$ are then taken as inputs to the fingerprint extractor.

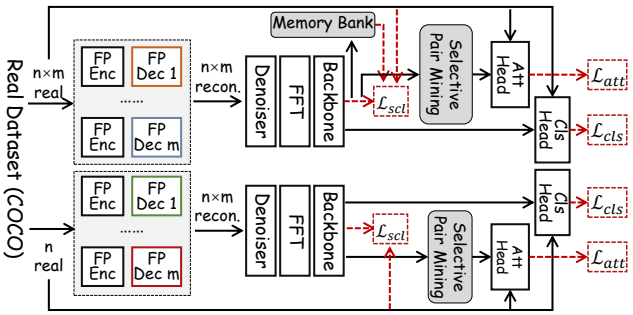

Figure 3: Training pipeline of the fingerprint extractor. *FP* stands for fingerprint. In each training step, sampled real images are reconstructed by m random decoder instances. Note that the fingerprint simulator is frozen during this process.

We now describe the architecture of the fingerprint extractor ϕ and its training objective. Let $\phi : \mathbb{R}^{3 \times H \times W} \to \mathbb{R}^d$ be the fingerprint extractor, which maps the frequency input to a compact d-dimensional fingerprint. To learn discriminative and generalizable fingerprints, we employ a supervised contrastive loss that encourages fingerprints of images from the same model to cluster together, while pushing apart those from different models. Based on the extracted fingerprints, we design two auxiliary heads: an attribution head that determines whether two images share the same origin, and a classification head that regularizes the model by distinguishing between original and reconstructed images.

Training Objective and Strategy Supervised Contrastive Learning with Memory Bank is the main loss responsible for fingerprint learning. Given a batch of $N = n \times m$ inputs $\{\mathcal{T}(\boldsymbol{r}_i)\}_{i=1}^N$ with associated labels $\{y_i\}_{i=1}^N$ that indicate the generative architecture, we define the positive set (the same architecture) for input i as:

$$P_i^+ = \{p \in \{1, \ldots, N\} \mid p \neq i,\ y_p = y_i\}. \tag{5}$$

We use log L2 normalized fingerprints $\boldsymbol{z} = \frac{\phi(\mathcal{T}(\boldsymbol{r}))}{\|\phi(\mathcal{T}(\boldsymbol{r}))\|_2}$ and calculate cosine similarity $\exp(\boldsymbol{z}_i \cdot \boldsymbol{z}_p)$ to quantify the similarity between the fingerprint $\boldsymbol{z}_i$ and a positive fingerprint $\boldsymbol{z}_p$. We use $\sum_{j \neq i} \exp(\boldsymbol{z}_i \cdot \boldsymbol{z}_j)$ to aggregate the similarities between fingerprint $\boldsymbol{z}_i$ and all other fingerprints in the batch. Negative samples play a crucial role in contrastive learning. Thus, we introduce a memory bank $\mathcal{M} = \{\boldsymbol{z}_i\}_{i=1}^M$ to store historical extracted fingerprints. Considering that the number of possible architectures exceeds 1.6 trillion, fingerprints from the memory bank are treated as negatives for the current fingerprint $\boldsymbol{z}_i$. Then, the overall supervised contrastive

loss with memory bank is defined as:

$$\mathcal{L}_{scl} = -\frac{1}{N}\sum_{i=1}^{N}\frac{1}{|P_i^+|}\sum_{p\in S_i^+}\log \frac{\exp(\boldsymbol{z}_i\cdot\boldsymbol{z}_p)/\tau}{\sum_{j\neq i}\exp(\boldsymbol{z}_i\cdot\boldsymbol{z}_j)/\tau+\sum_{\boldsymbol{z}^*\in\mathcal{M}}\exp(\boldsymbol{z}_i\cdot\boldsymbol{z}_*)/\tau} \tag{6}$$

where τ is the temperature parameter, this loss clusters outputs from the same model while separating different models to extract model-specific fingerprints.

Pairwise Similarity Learning enables pairwise image attribution based on fingerprints, i.e., determining whether two images originate from the same generative model. To this end, we introduce a lightweight attribution head $\boldsymbol{g}:\mathbb{R}^{2d}\rightarrow[0,1]$ to predict the likelihood that two extracted fingerprints belong to the same architecture:

$$\mathcal{L}_{att} = -\frac{1}{|S|}\sum_{(i,j)\in S}\big[\mathbb{1}(y_i=y_j)\log(\boldsymbol{g}([\boldsymbol{z}_i;\boldsymbol{z}_j])) + (1-\mathbb{1}(y_i=y_j))\log(1-\boldsymbol{g}([\boldsymbol{z}_i;\boldsymbol{z}_j]))\big], \tag{7}$$

where S is a set of index pairs (i,j) constructed to maintain a balance between positive pairs $(y_i=y_j)$ and negative pairs $(y_i\neq y_j)$ by selective pair mining, $\sigma(\cdot)$ is the sigmoid function, and $\mathbb{1}(\cdot)$ is the indicator function.

Real/Recon. Classification acts as a regularization loss to stabilize contrastive learning and mitigate representation collapse. It encourages the model to preserve discriminative information by explicitly distinguishing real images from images reconstructed by any architecture. Formally, we incorporate a binary classification head $\boldsymbol{f}:\mathbb{R}^d\rightarrow[0,1]$ to distinguish between real and reconstructed images:

$$\mathcal{L}_{cls} = -\frac{1}{N}\sum_{i=1}^{N}\Big[\mathbf{1}_{y_i\neq r}\log\big(\boldsymbol{f}(z_i)\big) + \mathbf{1}_{y_i=r}\log\big(1-\boldsymbol{f}(z_i)\big)\Big], \tag{8}$$

where r denotes the label of the original real images.

The overall training loss is:

$$\mathcal{L}_{total} = \lambda_{scl}\mathcal{L}_{scl}+\lambda_{att}\mathcal{L}_{att}+\lambda_{cls}\mathcal{L}_{cls} \tag{9}$$

where λ_{scl}, λ_{cls}, and λ_{att} are hyperparameters that balance each loss term. To ensure the extractor learns robust and discriminative fingerprints across diverse generative models, we adopt two complementary data sampling strategies during training as shown in Fig. 3. Specifically, in each training step: (1) **Multiple Sets:** we sample multiple sets of reconstructed images from different decoder instances, ensuring that each batch contains diverse fingerprints with different semantics. (2) **Single Set:** meanwhile, we sample a single set of reconstructed images, apply multiple decoder instances to create variations that share the same image content but have different fingerprints. This helps the model learn semantic-agnostic invariant fingerprints.

Experiments

Experimental Settings

Datasets Both our fingerprint simulator and extractor are trained solely on the COCO training set, and all images are cropped to 256 pixels. For evaluation, we used AIGCBenchmark (Zhong et al. 2023) and DMDetection (Corvi et al. 2023), each containing images generated by numerous SOTA and commercial generative models. Additionally, we evaluate and compare closed-set and open-set attribution performance using the GenImg (Zhu et al. 2023) and OSMA (Yang et al. 2023) datasets. All of these datasets contain samples generated by dozens of different models.

Settings We adopt ResNet-50 as the backbone of the fingerprint extractor. λ_{mse} and λ_{lpips} are set as 1.0 and 0.25. λ_{scl}, λ_{att}, and λ_{cls} are all set as 1.0. The temperature τ is set as 0.02. Memory bank $\mathcal{M}$ has 1024 historical fingerprints.

Evaluation Protocol We adopt two evaluation protocols:

- **Image-to-Model Attribution** that determines whether a query image is generated by a specific target model.
- **Open-set and Closed-set Attribution** that classifies images into known closed-set models or rejects them if generated by unseen open-set models.

Metrics Following prior work (Yang et al. 2022; Cioni et al. 2024), we use the Area Under the Curve (AUC) to evaluate the ability to distinguish images from different sources, classification accuracy to evaluate the performance on closed-set attribution, and Open Set Classification Rate (OSCR) (Yang et al. 2023) to evaluate the overall performance across both closed-set and open-set attribution.

Compared Methods For the first protocol, given the novelty of our framework and the limited number of related studies, the Siamese Network (SN) method proposed by Abady et al. (Abady et al. 2024) is currently the only directly comparable method for pairwise synthetic image attribution. We also include two large-scale pre-trained models, CLIP (Radford et al. 2021) and DINO (Caron et al. 2021), both of which support pairwise image comparison. Note that they are not specifically designed for attribution tasks, thus, their results are reported for reference only. For the second protocol, we compared our methods with SOTA synthetic image attribution methods, including **ResNet-50** (He et al. 2016), **ViT** (Dosovitskiy et al. 2020), **PRNU** (Marra et al. 2019), **CNNs** (Yu, Davis, and Fritz 2019) **DCT-CNN** (Frank et al. 2020), **DNA-Det** (Yang et al. 2022), **RepMix** (Bui, Yu, and Collomosse 2022), **POSE** (Yang et al. 2023), and **UA** (Cioni et al. 2024).

Image-to-Model Attribution in the Wild

We first evaluate our methods under the image-to-model attribution protocol. Specifically, for each generative model (referred to as the *target model*), we construct a support set [1] using images produced by it, and a query set containing both positive samples (from the target model) and negative samples (from another generative model). For each query image, we compute its attribution score by averaging the predictions of the query and all images in the support set using our attribution head. We perform this procedure in a one-vs-one manner between the target model and the remaining

[1] By default, the support set contains 100 images.

models. For each such pair, we compute the AUC by treating target-model queries as positives and the other-model queries as negatives. The final AUC reported for each model is obtained by averaging over all these 1-vs-1 comparisons. These results can measure how effectively our method distinguishes images from different models. The performances on the DMDetection and AIGCBenchmark are summarized in Table 1 and Table 2, respectively.

Models	SN	DINO	CLIP	Ours
BigGAN	45.37±15.82	62.54±18.02	66.53±17.21	**90.83**±10.06
DALLE-mini	34.38±13.15	67.16±14.98	78.36±9.48	**92.03**±15.31
DALLE2	26.85±16.82	68.69±12.65	78.25±10.00	**96.63**±9.55
EG3D	13.96±9.07	**98.62**±4.14	96.85±8.49	97.71±3.25
Guided Diff.	40.16±16.33	44.06±18.15	56.14±14.44	**91.23**±22.44
LDM (C2I)	42.34±15.41	41.03±18.26	61.94±13.73	**96.51**±0.77
LDM (T2I)	33.34±14.59	62.05±16.72	73.65±8.77	**91.37**±13.49
Stylegan2	45.57±18.02	92.60±3.22	**95.48**±2.91	95.36±5.60
Stylegan3	62.13±14.71	92.85±21.42	91.39±19.64	**99.99**±0.01
VQGAN	35.19±14.54	57.34±18.04	77.51±11.73	**88.81**±16.33
Average	37.93±12.74	77.61±20.09	68.69±13.84	**94.05**± 3.65

Table 1: Attribution AUC (%, mean±std) on DMDetection.

Performance on DMDetection Benchmark We can observe in Table 1 that our method achieves an average AUC of **94.05**%, outperforming the second-best baseline by over 25%. The improvement is particularly pronounced for advanced generative models such as Guided Diffusion and LDM, suggesting that our method captures generalizable fingerprints. In comparison with the SN, we used the official model provided by the authors. Although it was trained on a self-collected dataset from 5 generative models, it achieved only an average AUC of 37.93% when extended to DMDetection. We attribute this to the inclusion of unseen models, unknown sampling, and post-processing strategies in DMDetection. Besides, while large pre-trained models such as DINO and CLIP show moderate performance on certain models (e.g., EG3D, StyleGAN2/3), their attribution performance is inconsistent. We verify in Table 3, that they are more inclined to semantic clues when attributing. In Table 4, we find that such semantic dependency is ubiquitous. Please refer to more details in Sec. IV.C.

Performance on AIGCBenchmark On AIGCBenchmark (Table 2), which contains more diverse and recent generative models, our method outperforms all baselines with an average AUC of **83.05**%. While the overall AUC is lower compared to DMDetection, due to the higher diversity and real-world unknowability of this dataset, our method still shows strong generalization. Notably, our method achieves an AUC of 96.94% on ADM and 88.07% on SDXL, both high-performing diffusion models with minimal visual artifacts. Even for models with limited training exposure (e.g., VQDM and Wukong), our model maintains competitive performance (82.04% and 77.81%, respectively), demonstrating its capacity to generalize to in-the-wild models. In contrast, the performances of the compared methods are near a random level on most models.

Models	SN	DINO	CLIP	Ours
ADM	41.08±9.45	58.14±11.93	65.40±12.29	**96.94**±4.26
DALLE2	35.51±8.36	80.83±7.39	78.83±8.78	**88.43**±7.01
Glide	47.49±8.71	56.42±12.09	**86.53**±3.98	75.42±9.15
Midjourney	52.35±8.52	50.26±13.78	59.61±11.77	**78.93**±13.58
VQDM	32.29±8.05	67.78±12.29	71.31±11.09	**82.04**±21.17
SDXL	55.43±8.58	78.01±11.61	81.47±10.65	**88.07**±11.24
SDv1.4	55.28±7.64	42.07±10.75	55.91±11.76	**83.38**±19.93
SDv1.5	55.77±7.68	39.78±9.43	61.43±9.08	**76.41**±26.25
Wukong	64.78±6.69	44.28±10.26	56.97±11.43	**77.81**±23.77
Average	48.89±10.68	57.51±15.21	68.61±11.39	**83.05**±7.02

Table 2: Attribution AUC (%, mean±std) on AIGCBenchmark.

Model	Method	SG2-afhq	SG2-ffhq	SG3-afhq	SG3-ffhq
SG2-afhq	CLIP	-	100.0	50.01	100.0
	Ours	-	71.74	88.25	87.28
SG2-ffhq	CLIP	84.55	-	84.94	30.48
	Ours	80.05	-	99.38	91.07
SG3-afhq	CLIP	62.69	100.0	-	100.0
	Ours	99.63	99.83	-	93.31
SG3-ffhq	CLIP	99.05	78.59	99.09	-
	Ours	97.77	95.91	84.13	-

Table 3: Attribution AUC (%) on StyleGAN2/3 variants for different datasets, gray cell indicates the same dataset.

Semantic Dependency Analysis Attribution performance can be misleadingly inflated when evaluated on generative models that produce semantically distinct content (e.g., different categories or prompts). To evaluate semantic dependency, we run evaluations on models trained with the same StyleGAN2/3 (SG) architecture but different datasets: FFHQ (human faces) and AFHQ (animals). A higher AUC indicates a stronger ability to distinguish between the source models. As shown in Table 3, CLIP achieves high AUC when semantic cues are prominent (e.g., SG2-afhq → SG2-ffhq), but its performance deteriorates sharply when semantic differences are minimal (e.g., 30.48% for SG2-ffhq → SG3-ffhq), suggesting a strong reliance on content-level features. In contrast, our method maintains consistently high AUCs across all StyleGAN2/3 variants and domains, including cross-architecture and cross-dataset settings. This demonstrates that our method captures model-specific fingerprints rather than semantic cues.

Closed-set and Open-set Attribution

We then evaluate our method using the second protocol. To enable a comprehensive and fair comparison with SOTA methods, we fine-tune an additional attribution head on top of our fingerprint extractor using closed-set datasets. Formally, the attribution head is trained as a $(T+1)$-way classifier, where T is the number of closed-set models. The $(T+1)$-th class serves as a rejection category, designed to capture inputs that do not belong to any of the T known

Method	Closed-Set	Unseen Seed		Unseen Arch.		Unseen Data		Unseen All		Average	
	Accuracy	AUC	OSCR	AUC	OSCR	AUC	OSCR	AUC	OSCR	AUC	OSCR
PRNU	55.27	69.20	49.16	70.02	49.49	67.68	48.57	68.94	49.06	68.96	49.07
CNNs	85.71	53.14	50.99	69.04	64.17	78.79	72.20	69.90	64.86	67.71	63.05
DCT-CNN	86.16	55.46	52.68	72.56	67.43	72.87	67.57	69.46	64.70	67.58	63.09
DNA-Det	93.56	61.46	59.34	80.93	76.45	66.14	63.27	71.40	68.00	69.98	66.76
RepMix	93.69	54.70	53.26	72.86	70.49	78.69	76.02	71.74	69.43	69.49	67.30
POSE	94.81	68.15	67.25	**84.17**	81.62	88.24	85.64	**82.76**	80.50	80.83	78.75
UA (NN)	94.18	57.62	56.69	77.95	75.16	90.60	91.77	80.42	77.11	76.64	75.18
UA (NN+)	95.31	56.18	54.04	80.22	77.27	88.90	85.42	80.65	78.22	76.48	73.73
UA (LP)	97.29	54.15	54.00	78.78	78.12	90.60	89.52	79.29	78.77	75.70	75.10
Ours	**98.67**	**98.35**	**97.08**	82.96	**82.20**	**93.39**	**93.03**	82.34	**81.61**	**89.26**	**88.48**

Table 4: Attribution performance (%) on the OSMA dataset with different types of unseen images.

models. During training, this rejection class is supervised using reconstructed real images generated by our fingerprint simulator, which are not associated with any specific model. This setup establishes a unified evaluation protocol that enables direct and fair comparison with prior methods.

Performance on GenImage Benchmark In Table 5, all methods are evaluated over five-fold cross-validation, where each fold comprises a distinct combination of closed-set and open-set models (Yang et al. 2023). We can observe that our method achieves the highest AUC (86.79%) and OSCR (85.08%), indicating superior ability in distinguishing images from closed-set models and handling open-set attribution scenarios. Although UA (LP+) achieves the best accuracy (97.82%), our method closely follows (96.87%) while attaining superior AUC and OSCR, suggesting a more balanced performance across both closed set and open set. Other baselines, such as POSE, DNA-Det, and RepMix, exhibit noticeably lower AUC and OSCR, despite competitive accuracy, showing limited generalization to open-set cases.

Method	Accuracy	AUC	OSCR
ResNet-50	91.66 ± 3.74	70.74 ± 5.27	68.29 ± 6.98
ViT	93.26 ± 2.83	67.77 ± 5.58	66.02 ± 7.24
DNA-Det	93.83 ± 7.72	61.27 ± 6.70	75.08 ± 11.02
RepMix	88.98 ± 4.36	61.93 ± 4.26	57.92 ± 1.73
POSE	70.00 ± 25.95	67.00 ± 6.08	53.35 ± 19.82
UA(NN)	93.26 ± 5.46	77.18 ± 1.88	72.06 ± 3.26
UA(NN+)	94.59 ± 5.92	80.96 ± 4.85	77.80 ± 9.27
UA(LP+)	**97.82** ± 2.51	81.39 ± 3.28	80.78 ± 4.02
Ours	96.87 ± 4.45	**86.79** ± 6.15	**85.08** ± 7.25

Table 5: Performance (%, mean$\pm$std) on GenImage.

Performance on OSMA Benchmark Table 4 shows a comprehensive evaluation of attribution methods on the OSMA dataset under multiple unseen conditions: unseen seed, architecture, data, and their combination. The results are averaged over 5 different splits. Our method consistently achieves the best performance across nearly all metrics, with a closed-set accuracy of 98.67%. This indicates the capacity to correctly attribute images to known classes.

More importantly, in open-set scenarios, where unseen generative seeds, architectures, or datasets are introduced, our method maintains remarkably high AUC and OSCR scores, with an average AUC of 89.26% and OSCR of 88.48%, outperforming all competing methods. While POSE and UA exhibit competitive AUC and OSCR on some unseen splits, their performance is still less stable overall. It is worth noting that on the four unseen types, all methods except PRNU exhibit higher attribution performance on the unseen datasets. In contrast, the PRNU method, which is based on non-semantic noise, shows consistently stable AUC scores. This suggests that existing methods may have a bias toward semantic cues, as we mentioned in Sec. IV.B.

Conclusion

In this paper, we present a synthesis-free framework for synthetic image attribution that eliminates the need for large-scale synthetic training data. The proposed stochastic fingerprint simulator with supervised contrastive learning on frequency features enables robust attribution without relying on synthetic images of specific generative models. Comprehensive evaluation across diverse generative models demonstrates the ability to capture discriminative model-specific fingerprints, with strong generalization to unseen architectures. Our work opens new avenues for scalable attribution methods in an era of rapidly evolving generative models.

Acknowledgments

This work is supported in part by the National Natural Science Foundation of China under grant numbers 625B2187, U23B2023, and 62472199, Guangdong Key Laboratory of Data Security and Privacy Preserving under Grant 2023B1212060036, the basic and Applied Basic Research Foundation of Guangdong Province (2025A1515011097), and the Outstanding Youth Project of Guangdong Basic and Applied Basic Research Foundation (2023B1515020064). This work is also supported by the Engineering Research Center of Trustworthy AI, Ministry of Education. This work is also supported by the China Scholarship Council 202406380227.

References

Abady, L.; Wang, J.; Tondi, B.; and Barni, M. 2024. A siamese-based verification system for open-set architecture attribution of synthetic images. *Pattern Recognition Letters*, 180: 75–81.

Bindini, L.; Bertazzini, G.; Baracchi, D.; Shullani, D.; Frasconi, P.; and Piva, A. 2024. Tiny autoencoders are effective few-shot generative model detectors. In *2024 IEEE International Workshop on Information Forensics and Security (WIFS)*, 1–6. IEEE.

Bui, T.; Yu, N.; and Collomosse, J. 2022. Repmix: Representation mixing for robust attribution of synthesized images. In *European Conference on Computer Vision*, 146–163. Springer.

Caron, M.; Touvron, H.; Misra, I.; Jégou, H.; Mairal, J.; Bojanowski, P.; and Joulin, A. 2021. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, 9650–9660.

Cioni, D.; Tzelepis, C.; Seidenari, L.; and Patras, I. 2024. Are CLIP features all you need for Universal Synthetic Image Origin Attribution? In *European Conference on Computer Vision*, 363–382. Springer.

Corvi, R.; Cozzolino, D.; Zingarini, G.; Poggi, G.; Nagano, K.; and Verdoliva, L. 2023. On the detection of synthetic images generated by diffusion models. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1–5. IEEE.

Dai, Y.; Fei, J.; and Huang, F. 2024. IDGuard: Robust, General, Identity-Centric POI Proactive Defense Against Face Editing Abuse. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 11934–11943. IEEE.

Dai, Y.; Fei, J.; Huang, F.; and Chang, C. H. 2025. Robust Secure Swap: Responsible Face Swap With Persons of Interest Redaction and Provenance Traceability. In *Forty-second International Conference on Machine Learning*.

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 *International Conference on Learning Representations*.

Du, S.; Yang, P.; Baracchi, D.; Jin, J.; Shullani, D.; and Piva, A. 2025. ForensiCam-215K: A Large Scale Image and Video Dataset for Forensic Analysis. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1–5. IEEE.

Fang, S.; Nguyen, T. D.; and Stamm, M. C. 2023. Open Set Synthetic Image Source Attribution. In *34th British Machine Vision Conference 2023, BMVC 2023, Aberdeen, UK, November 20-24, 2023*. BMVA.

Fei, J.; Dai, Y.; Xia, Z.; Huang, F.; and Zhou, J. 2025a. OmniMark: Efficient and Scalable Latent Diffusion Model Fingerprinting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 16550–16558.

Fei, J.; Dai, Y.; Yang, W.; and Xia, Z. 2025b. Distributor-Centric Model Watermarking for Image Generative Models. *Knowledge-Based Systems*, 114422.

Fei, J.; Dai, Y.; Yu, P.; Kong, Z.; Zhou, J.; and Xia, Z. 2025c. Scalable Dual Fingerprinting for Hierarchical Attribution of Text-to-Image Models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 15025–15034.

Fei, J.; Dai, Y.; Yu, P.; Shen, T.; Xia, Z.; and Weng, J. 2022a. Learning second order local anomaly for general face forgery detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 20270–20280.

Fei, J.; Xia, Z.; Tondi, B.; and Barni, M. 2022b. Supervised gan watermarking for intellectual property protection. In *2022 IEEE international workshop on information forensics and security (WIFS)*, 1–6. IEEE.

Fei, J.; Xia, Z.; Tondi, B.; and Barni, M. 2024. Wide flat minimum watermarking for robust ownership verification of gans. *IEEE Transactions on Information Forensics and Security*.

Fernandez, P.; Couairon, G.; Jégou, H.; Douze, M.; and Furon, T. 2023. The stable signature: Rooting watermarks in latent diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 22466–22477.

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

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Joslin, M.; and Hao, S. 2020. Attributing and detecting fake images generated by known GANs. In *2020 IEEE Security and Privacy Workshops (SPW)*, 8–14. IEEE.

Kong, C.; Chen, B.; Li, H.; Wang, S.; Rocha, A.; and Kwong, S. 2022. Detect and locate: Exposing face manipulation by semantic-and noise-level telltales. *IEEE Transactions on Information Forensics and Security*, 17: 1741–1756.

Kong, C.; Chen, B.; Yang, W.; Li, H.; Chen, P.; and Wang, S. 2021. Appearance matters, so does audio: Revealing the hidden face via cross-modality transfer. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(1): 423–436.

Luo, A.; Kong, C.; Huang, J.; Hu, Y.; Kang, X.; and Kot, A. C. 2023. Beyond the prior forgery knowledge: Mining critical clues for general face forgery detection. *IEEE Transactions on Information Forensics and Security*, 19: 1168–1182.

Marra, F.; Gragnaniello, D.; Verdoliva, L.; and Poggi, G. 2019. Do gans leave artificial fingerprints? In *2019 IEEE conference on multimedia information processing and retrieval (MIPR)*, 506–511. IEEE.

Midjourney, Inc. 2023. Midjourney.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.;

et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, 8748–8763. PmLR.

Ramesh, A.; Dhariwal, P.; Nichol, A.; Chu, C.; and Chen, M. 2022. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2): 3.

Shi, W.; Caballero, J.; Huszár, F.; Totz, J.; Aitken, A. P.; Bishop, R.; Rueckert, D.; and Wang, Z. 2016. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1874–1883.

Wang, J.; Alamayreh, O.; Tondi, B.; and Barni, M. 2023. Open set classification of gan-based image manipulations via a vit-based hybrid architecture. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 953–962.

Yang, T.; Huang, Z.; Cao, J.; Li, L.; and Li, X. 2022. Deepfake network architecture attribution. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 4662–4670.

Yang, T.; Wang, D.; Tang, F.; Zhao, X.; Cao, J.; and Tang, S. 2023. Progressive open space expansion for open-set model attribution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 15856–15865.

Yu, N.; Davis, L. S.; and Fritz, M. 2019. Attributing fake images to gans: Learning and analyzing gan fingerprints. In *Proceedings of the IEEE/CVF international conference on computer vision*, 7556–7566.

Yu, N.; Skripniuk, V.; Abdelnabi, S.; and Fritz, M. 2021. Artificial fingerprinting for generative models: Rooting deepfake attribution in training data. In *Proceedings of the IEEE/CVF International conference on computer vision*, 14448–14457.

Yu, P.; Fei, J.; Gao, H.; Feng, X.; Xia, Z.; and Chang, C. H. 2025. Unlocking the Capabilities of Large Vision-Language Models for Generalizable and Explainable Deepfake Detection. In *Forty-second International Conference on Machine Learning*.

Zhang, R.; Isola, P.; Efros, A. A.; Shechtman, E.; and Wang, O. 2018. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 586–595.

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

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