FerretNet: Efficient Synthetic Image Detection via Local Pixel Dependencies

Shuqiao Liang **Jian Liu** **Renzhang Chen**[*] **Quanlong Guan**[*]
Jinan University
{xigua7105, liujian2143}@gmail.com, {jnulion, gql}@jnu.edu.cn

Abstract

The increasing realism of synthetic images generated by advanced models such as VAEs, GANs, and LDMs poses significant challenges for synthetic image detection. To address this issue, we explore two artifact types introduced during the generation process: (1) latent distribution deviations and (2) decoding-induced smoothing effects, which manifest as inconsistencies in local textures, edges, and color transitions. Leveraging local pixel dependencies (LPD) properties rooted in Markov Random Fields, we reconstruct synthetic images using neighboring pixel information to expose disruptions in texture continuity and edge coherence. Building upon LPD, we propose FerretNet, a lightweight neural network with only 1.1M parameters that delivers efficient and robust synthetic image detection. Extensive experiments demonstrate that FerretNet—trained exclusively on the 4-class ProGAN dataset—achieves an average accuracy of 97.1% on an open-world benchmark comprising 22 generative models. Our code and datasets are publicly available at https://github.com/xigua7105/FerretNet.

1 Introduction

The field of AI-based image generation has progressed rapidly, driven by the development of powerful generative models such as Variational Autoencoders (VAEs) [23], Generative Adversarial Networks (GANs) [19, 22, 1], and Latent Diffusion Models (LDMs) [41, 34, 9]. These models have enabled widespread applications across art, entertainment, and e-commerce, allowing users to effortlessly create realistic and engaging images. However, the potential misuse of such content has raised ethical concerns and driven extensive research on synthetic image detection [51, 10, 31, 27].

Many existing detection approaches rely heavily on model-specific features, which limit their generalization ability to unseen generative architectures. For example, Durall et al. [8] observed characteristic frequency artifacts in GAN-generated images. Although frequency-domain techniques [51, 16, 17] have demonstrated strong performance under known conditions, they often struggle to generalize across different models. DIRE [52] introduced a diffusion-based detection framework that distinguishes synthetic images by reconstructing them through a diffusion model, a capability that fails with real images. However, this method performs poorly when applied to GAN-generated content.

To address the generalization challenge, Ojha et al. [31] explored the utilization of pre-trained models, employed frozen backbone for image encoding, providing universal representations from pre-training, followed by a linear classifier. FatFormer [27] introduced an Adaptor to CLIP [38] to enhance the pre-trained model's ability to learn artifacts. While these methods achieved encouraging results, they are constrained by large parameter counts or low computational efficiency.

To overcome the dual challenges of limited generalization and computational inefficiency in synthetic image detection, we conduct a comprehensive analysis of artifact patterns shared across GAN-, VAE-,

[*] Corresponding authors.

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

and LDM-based generative models. Our analysis reveals that visual anomalies-such as unnatural textures, geometric distortions, and poor object-background integration—primarily originate from two sources: (1) distributional shifts in the latent variable z, and (2) over-smoothing and color discontinuities introduced during the decoding process.

Based on these insights and grounded in the theory of Markov Random Fields, we introduce a pixel-level artifact representation that captures local pixel dependencies (LPD) through median-based reconstruction. We further propose FerretNet, a lightweight detector incorporating depthwise separable and dilated convolutions to achieve a balance between computational efficiency and representational power.

Contributions of this work are as follows:

- We propose a novel approach that leverages Markov Random Fields and median-based statistics to capture local pixel dependencies for detecting artifacts and anomalies in synthetic images.
- We present Synthetic-Pop, a 60K-image benchmark for evaluating detection models against high-fidelity generators, containing 30K synthetic images from six models and 30K real images from COCO [25] and LAION-Aesthetics V2 (6.5+) [45]. See Appendix A for more details.
- We introduce FerretNet, a lightweight model with only 1.1 million parameters, which achieves 97.1% accuracy on synthetic image detection across 22 generative models, while maintaining low computational overhead.

2 Related Work

We categorize existing synthetic image detection methods into two main paradigms: pixel-based and frequency-based approaches.

2.1 Pixel-based Synthetic Image Detection

Wang et al. [51] trained a classifier on images generated by a single model to detect fake images across various architectures and datasets, addressing cross-model generalization via data augmentation and diverse training samples. Shi et al. [47] proposed a difference-guided reconstruction learning framework that exploits discrepancies between real and synthetic images to enhance detection accuracy. Ojha et al. [31] tackled the generalization problem to unseen generative models by leveraging a feature space not explicitly trained for real/fake discrimination, employing nearest-neighbor and linear probing strategies. He et al. [13] introduced a super-resolution-based re-synthesis technique to reconstruct test images and extract residual or layered artifact features, thereby reducing reliance on frequency artifacts. Tan et al. [50] proposed NPR, a method that revisits the upsampling process in generative CNNs by modeling Neighbor Pixel Relations, aiming to improve generalization in deepfake detection. Liu et al. [26] designed a robust detection framework based on multi-view image completion, which simulates real image distributions and captures frequency-independent features. FatFormer [27] presented a forgery-aware adaptive transformer incorporating forgery-specific adapters and language-guided alignment modules to better adapt pre-trained models for synthetic image detection. CO-SPY [2] leverages a frozen CLIP encoder for semantic features and a VAE-based reconstruction difference for artifacts, integrating them via adaptive fusion for robust synthetic image detection.

2.2 Frequency-based Synthetic Image Detection

F3Net [37] introduced a dual-branch architecture that captures frequency-aware clues for detecting subtle forgery traces, particularly in low-quality and facial imagery. FrePGAN [17] developed a frequency-level perturbation GAN framework, where a generator-discriminator pair is used to iteratively improve classifier robustness against unseen categories and generative models. Tan et al. [48] exploited pre-trained CNN gradients to generate generalizable representations of GAN-specific artifacts. BiHPF [16] amplified frequency-level artifacts via a high-pass filtering approach, achieving improved robustness across diverse image categories, color manipulations, and generative models. FreqNet [49] introduced high-frequency representations and frequency-specific convolution layers to enhance detection by focusing on localized high-frequency components, addressing overfitting and poor generalization seen in prior methods. SAFE [24] leverages the high-frequency component of the Discrete Wavelet Transform to extract forensic artifacts and employs data augmentation techniques including ColorJitter, RandomRotation, and a patch-based RandomMask mechanism to improve the model's generalization and robustness.

3 Artifacts in Synthetic Image Generation

This section provides a high-level, intuitive framework to motivate our search for universal artifacts. We establish the general principle that all generative models, despite architectural differences, introduce artifacts. While these artifacts have multiple high-level sources (e.g., latent space, decoding), our work chooses to focus on detecting a powerful and universal *effect*—the disruption of local pixel statistics—which is effectively captured by our LPD method.

3.1 Image Generation Pipeline

Generative models such as VAEs, GANs, and LDMs are widely used for image synthesis. Despite differences in architecture and training objectives, these models share a common two-stage generation pipeline, as illustrated in Figure 1.

1. Obtaining the latent variable z:

In LDMs, the generation process begins with Gaussian noise $\epsilon \sim \mathcal{N}(0, I)$, which is iteratively denoised into a latent representation z within the compressed latent space of a pretrained autoencoder, using a denoising network such as U-Net [42, 41] or Diffusion Transformer (DiT) [33, 54]. In contrast, VAEs and GANs directly sample z from predefined prior distributions, such as a standard normal distribution $\mathcal{N}(0, I)$ or a uniform distribution $U(-1, 1)$.

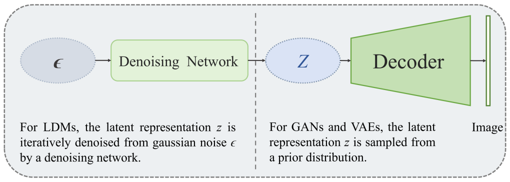

ϵ
Denoising Network
Z
Decoder
For LDMs, the latent representation z is iteratively denoised from gaussian noise ϵ by a denoising network.
For GANs and VAEs, the latent representation z is sampled from a prior distribution.
Image

Figure 1: The image generation process in VAEs, GANs, and LDMs can be broadly divided into two stages: obtaining the latent variable z, and decoding it into an image.

2. Decoding z to generate images: In both VAEs and LDMs, a decoder transforms z into the final image through a series of convolutional layers with specific kernel sizes and strides. In GANs, the generator plays an analogous role, mapping z to the image space with the aim of approximating the target data distribution.

While this two-stage framework enables high-fidelity image synthesis, it can also introduce artifacts such as texture irregularities, unnatural transitions, and local detail loss. These artifacts commonly arise from two major sources: (1) deviations in the distribution of the latent variable z, and (2) imperfections introduced during the decoding process.

3.2 Latent Distribution Deviations

The quality of synthetic images exhibits significant sensitivity to the distribution of the latent representation z [40, 14, 5, 57]. Ideally, the sampled distribution $Q(z)$ should match the prior distribution $P(z)$ assumed or learned during training. However, in practice, factors such as data imbalance or insufficient training can lead to a mismatch between $Q(z)$ and $P(z)$. This discrepancy can be quantified using the Kullback–Leibler (KL) divergence:

$$D_{\mathrm{KL}}(Q(z) \| P(z)) = \int Q(z) \log \frac{Q(z)}{P(z)} \, dz > \delta, \tag{1}$$

where δ denotes an acceptable divergence threshold. When this threshold is exceeded, the resulting images are prone to visible artifacts, including texture inconsistencies and the loss of fine structural details. For example, in GANs, if the latent space is poorly aligned with the true data distribution, the generator may fail to reproduce realistic textures, resulting in unnatural or distorted outputs [53].

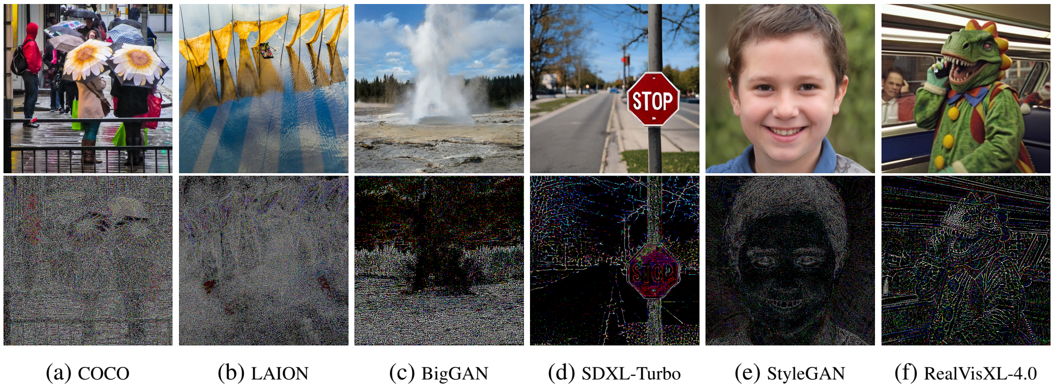


Figure 2: Local pixel dependencies (LPD) comparison between real and synthetic images. Top row: real images (COCO, LAION) and synthetic images (BigGAN, SDXL-Turbo, StyleGAN, RealVisXL-4.0). Bottom row: LPD maps derived from neighborhood-median reconstruction emphasize structural differences.

3.3 Artifacts from the Decoding Process

Even when z is accurately sampled, decoding artifacts may still arise due to limitations in the network architecture [21]. The kernel size and stride used in convolutional layers are particularly influential in determining the fidelity of the output [22]. Large kernels may over-smooth local features, while improper stride configurations can lead to aliasing, both of which degrade image quality.

Moreover, upsampling operations—such as nearest-neighbor or bilinear interpolation—are known to introduce specific artifacts. Nearest-neighbor interpolation often produces jagged edges, whereas bilinear interpolation may blur textures due to its smoothing effect. These operations can significantly impact the realism and perceptual quality of the generated images, especially in high-frequency regions.

4 Methodology

4.1 Local Median-based Feature Extraction

Natural images exhibit strong local statistical consistency due to the underlying physics of light and matter, where neighboring pixels are highly correlated. Generative models, however, often struggle to perfectly replicate these subtle, complex statistics during the synthesis of high-frequency details. This struggle leads to microscopic disruptions in local pixel dependencies. We propose a synthetic image detection method based on local statistical dependencies. The core idea is to identify generation artifacts by quantifying the deviation of each pixel from the median of its surrounding neighborhood. The full computational procedure is outlined in Algorithm 1.

Let I denote the input image, and $x_{i,j}$ represent the pixel value at location (i,j). According to the Markov Random Field (MRF) assumption, the probability distribution of a pixel depends only on its local neighborhood. Specifically,

$$P(x_{i,j} \mid x_{k,l}, (k,l) \neq (i,j)) = P(x_{i,j} \mid x_{k,l}, (k,l) \in \mathcal{N}_{i,j}), \tag{2}$$

where $\mathcal{N}_{i,j}$ is the set of neighboring pixels located within an $n \times n$ window centered at (i,j), excluding the center pixel itself:

$$\mathcal{N}_{i,j} = \left\{ x_{k,l} \;\middle|\; \begin{array}{l} i-m \leq k \leq i+m,\; j-m \leq l \leq j+m, \\ (k,l) \neq (i,j) \end{array} \right\}, \tag{3}$$

with $n = 2m+1$ and $m \in \mathbb{Z}^{+}$.

To enhance the robustness of the median filtering process and prevent contamination from generated pixels, we introduce a zero-masking strategy that replaces the center pixel with zero before computing the median. This adjustment is particularly beneficial when the neighborhood contains an even

Algorithm 1 Local Dependency Feature Extraction via Zero-Masked Median Deviation

Input: I: Image tensor of shape (C, H, W); n: Neighborhood size (n is odd)
Output: LPD: Feature map of shape (C, H, W)

```
# Compute padding size and center index
$p \leftarrow \lfloor n/2 \rfloor, \quad center\_idx \leftarrow \lfloor n^2/2 \rfloor$
# Pad the image to handle borders
$I_{\text{pad}} \leftarrow \text{Pad}(I, \text{padding} = p, \text{mode} = \text{'constant'}, \text{value} = 0)$
# Extract $n \times n$ local patches centered at each pixel
$I_{\text{patches}} \leftarrow \text{Unfold}(I_{\text{pad}}, \text{kernel\_size} = n)$    // shape: $(C, n^2, H \cdot W)$
# Zero out the center pixel in each patch
$I_{\text{patches}}[:, center\_idx, :] \leftarrow 0$
# Compute median along patch dimension
$I'_{\text{med}} \leftarrow \text{Median}(I_{\text{patches}}, \text{dim} = 1)$
# Reshape to match original image dimensions
$I' \leftarrow \text{Reshape}(I'_{\text{med}}, \text{shape} = (C, H, W))$
# Compute local pixel dependency map
$LPD \leftarrow I - I'$
return $LPD$
```

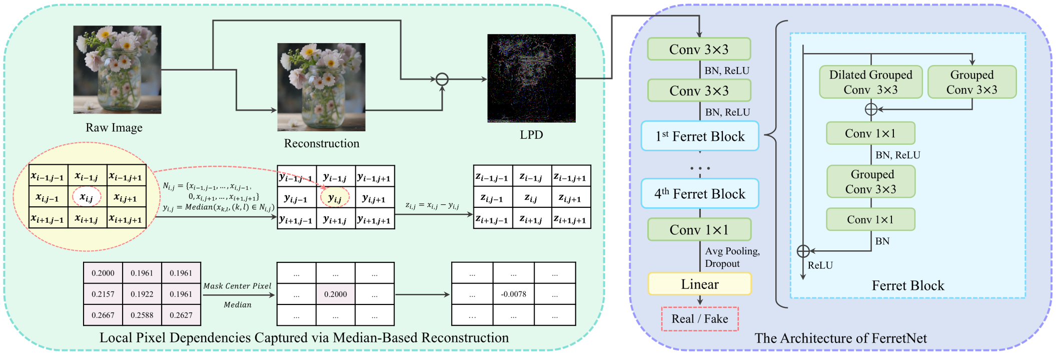

Raw Image
Reconstruction
LPD
$x_{i-1,j-1}$ $x_{i-1,j}$ $x_{i-1,j+1}$
$x_{i,j-1}$ $x_{i,j}$ $x_{i,j+1}$
$x_{i+1,j-1}$ $x_{i+1,j}$ $x_{i+1,j+1}$
$N_{i,j} = \{x_{i-1,j-1}, \dots, x_{i,j-1}, 0, x_{i,j+1}, \dots, x_{i+1,j+1}\}$
$y_{i,j} = Median(x_{k,l}, (k,l) \in N_{i,j})$
$y_{i-1,j-1}$ $y_{i-1,j}$ $y_{i-1,j+1}$
$y_{i,j-1}$ $y_{i,j}$ $y_{i,j+1}$
$y_{i+1,j-1}$ $y_{i+1,j}$ $y_{i+1,j+1}$
$z_{i,j} = x_{i,j} - y_{i,j}$
$z_{i-1,j-1}$ $z_{i-1,j}$ $z_{i-1,j+1}$
$z_{i,j-1}$ $z_{i,j}$ $z_{i,j+1}$
$z_{i+1,j-1}$ $z_{i+1,j}$ $z_{i+1,j+1}$
0.2000 0.1961 0.1961
0.2157 0.1922 0.1961
0.2667 0.2588 0.2627
Mask Center Pixel
Median
... ...
... 0.2000 ...
... ...
... ...
... -0.0078 ...
... ...
Local Pixel Dependencies Captured via Median-Based Reconstruction
Conv 3×3
BN, ReLU
Conv 3×3
BN, ReLU
1st Ferret Block
...
4th Ferret Block
Conv 1×1
Avg Pooling, Dropout
Linear
Real / Fake
Dilated Grouped Conv 3×3
Grouped Conv 3×3
Conv 1×1
BN, ReLU
Grouped Conv 3×3
Conv 1×1
BN
ReLU
Ferret Block
The Architecture of FerretNet

Figure 3: Pipeline of FerretNet: computation of local pixel median discrepancy for artifact representation, followed by lightweight detection using depthwise separable and dilated convolutions.

number of pixels. The median-based reconstruction at location (i, j) is therefore computed as:

$$y_{i,j} = \text{Median}(x_{k,l}, (k,l) \in \mathcal{N}'_{i,j}), \tag{4}$$

where $\mathcal{N}'_{i,j} = \mathcal{N}_{i,j} \cup \{x_{i,j} = 0\}$ is the extended neighborhood that includes the masked center pixel.

By applying the above operation to all pixels, we obtain a median-reconstructed image I', where each pixel value is replaced by its corresponding $y_{i,j}$. The final local pixel dependency (LPD) feature map is then computed as the pixel-wise difference:

$$LPD = I - I'. \tag{5}$$

Since both I and I' conform to local dependency assumptions, the LPD feature map effectively captures pixel-level inconsistencies and subtle structural deviations, offering strong cues for distinguishing synthetic from natural content, as illustrated in Figure 2.

This method effectively integrates the local dependency modeling capabilities of Markov Random Fields with the robustness of median filtering, providing a principled and resilient strategy for detecting subtle inconsistencies in synthetic imagery.

4.2 FerretNet Architecture

FerretNet is a lightweight convolutional neural network designed to achieve a balance between computational efficiency and feature extraction capability. As illustrated in Figure 3, the network begins with two conventional 3 × 3 convolutional layers for initial feature extraction, each followed by Batch Normalization (BN) [15] and ReLU activation.

At the core of FerretNet are four cascaded Ferret Blocks, which progressively refine the extracted features while keeping the model compact. The final stage comprises a 1×1 convolution, global average pooling, Dropout regularization, and a fully connected layer for classification.

The key innovation lies in the Ferret Block, which is designed to expand the effective receptive field under constrained network depth, thereby enhancing the model's capacity for local pattern extraction. Each Ferret Block adopts a dual-path parallel architecture to increase the receptive field:

- The **primary path** employs a 3×3 *dilated grouped convolution* with a dilation rate of 2. The number of groups equals the number of input channels, allowing the receptive field to expand without increasing the number of parameters.
- The **secondary path** utilizes a standard 3×3 *grouped convolution*, maintaining the same grouping structure to capture fine-grained local patterns.

This dual-path configuration approximates a sparse 5×5 receptive field via parallel processing, enabling FerretNet to simulate deeper network behaviors within shallower layers, thus reducing computational cost. The outputs from both paths are fused through a 1×1 convolution, followed by BN and ReLU activation. Additional 3×3 grouped and 1×1 convolution layers further enrich the feature representation. Residual connections are employed to facilitate stable gradient propagation and enhance learning stability.

5 Experiments

5.1 Dataset Construction

5.1.1 Training Dataset

To ensure a consistent evaluation baseline, we follow the protocols established in [16, 31, 27, 50], utilizing four semantic classes (car, cat, chair, horse) from the ForenSynths dataset [51]. Each class contains 18,000 synthetic images generated by ProGAN [19], paired with an equal number of real images from the LSUN dataset [56]. All methods compared in this study were trained or fine-tuned on this same limited ProGAN 4-class dataset, except for CO-SPY [2], which utilized its officially released weights trained on other datasets.

5.1.2 Testing Dataset

To assess the generalization ability of the proposed method under real-world conditions, we evaluate its performance on diverse synthetic and real images from four distinct test sets, comprising a total of 22 generative models:

ForenSynths. This test set includes synthetic images generated by eight representative generative models: ProGAN [19], StyleGAN [20], StyleGAN2 [21], BigGAN [1], CycleGAN [58], StarGAN [3], GauGAN [32], and Deepfake [43]. Real images are sourced from six widely-used datasets: LSUN [56], ImageNet [44], CelebA [29], CelebA-HQ [18], COCO [25], and FaceForensics++ [43], totaling 62,000 images.

Diffusion-6-cls. As described in FatFormer [27], this test set comprises synthetic images generated by six diffusion-based models collected from DIRE [52] and Ojha et al. [31], including DALL-E [39], Guided [7], PNDM [28], VQ-Diffusion [11], Glide [30], and LDM [41]. Variants produced by Glide and LDM with different parameter configurations are treated as separate categories (see original papers for details). Each subset includes 1,000 synthetic and 1,000 real images, with some real images reused across subsets.

Synthetic-Pop. To capture the latest progress in high-resolution image generation, we constructed the Synthetic-Pop dataset using six popular models—Openjourney [36], Proteus-0.3 [6], RealVisXL-4.0 [46], SD-3.5-Medium [9], SDXL-Turbo [34], and YiffyMix [55]. Each model was prompted with 5,000 captions randomly sampled from COCO [25]. Real images were drawn from COCO and LAION-Aesthetics V2 (6.5+) [45], resulting in six subsets, each containing 5,000 synthetic and 5,000 real images (60,000 images total).

Synthetic-Aesthetic. To further investigate the aesthetic and stylistic diversity of synthetic imagery, we sampled 40,000 images from the Simulacra Aesthetic Captions (SAC) dataset [35], which were generated by CompVis latent GLIDE [41] and Stable Diffusion [41] using prompts sourced from over 40,000 real users. An equal number of real images were sampled from LAION-Aesthetics V2 (6.5+) [45], resulting in a total of 80,000 images. This dataset provides a challenging benchmark for evaluating performance under realistic and user-driven conditions.

5.2 Implementation Details

FerretNet is trained from scratch without any pretraining. We use the Adam optimizer with a learning rate of 2×10^{-4}, betas of $(0.937, 0.999)$, and a weight decay of 5×10^{-4}. The model is trained for 100 epochs using a batch size of 32. During training, input images are randomly cropped to a resolution of 224×224 and augmented with random horizontal flipping. Binary Cross Entropy with Logits Loss (BCEWithLogitsLoss) is adopted as the loss function. For evaluation, images are center-cropped to 256×256.

Following previous work [16, 31, 27], Accuracy (ACC) and Average Precision (AP) are used as the primary evaluation metrics. To measure real-world performance, we report throughput on the Synthetic-Aesthetic test set using an NVIDIA RTX 4090 GPU and an Intel(R) Xeon(R) Gold 6430 CPU (16 vCPUs), with a batch size of 128.

5.3 Main Results

Table 1: Accuracy and average precision comparisons with peer methods on ForenSynth test set for different GAN images and Deepfake images. The best and second best performance are highlighted in **bold** and underlined, respectively.

Methods	ProGAN	StyleGAN	StyleGAN2	BigGAN	CycleGAN	StarGAN	GauGAN	Deepfake	Mean
Wang [51]	91.4/99.4	63.8/91.4	76.4/97.5	52.9/73.3	72.7/88.6	63.8/90.8	63.9/92.2	51.7/62.3	67.1/86.9
F3Net [37]	99.4/100.0	92.6/99.7	88.0/99.8	65.3/69.9	76.4/84.3	100.0/100.0	58.1/56.7	63.5/78.8	80.4/86.2
FrePGAN [17]	99.0/99.9	80.7/89.6	84.1/98.6	69.2/71.1	71.1/74.4	99.9/100.0	60.3/71.7	70.9/91.9	79.4/87.2
BiHPF [16]	90.7/86.2	76.9/75.1	76.2/74.7	84.9/81.7	81.9/78.9	94.4/94.4	69.5/78.1	54.4/54.6	78.6/77.9
LGrad [48]	99.9/100.0	94.8/99.9	96.0/99.9	82.9/90.7	85.3/94.0	99.6/100.0	72.4/79.3	58.0/67.9	86.1/91.5
Ojha [31]	99.7/100.0	89.0/98.7	83.9/98.4	90.5/99.1	87.9/99.8	91.4/100.0	89.9/100.0	80.2/90.2	89.1/98.3
FreqNet [49]	99.6/100.0	90.2/99.7	88.0/99.5	90.5/96.0	95.8/99.6	85.7/99.8	93.4/98.6	88.9/94.4	91.5/98.5
NPR [50]	99.8/100.0	96.3/99.8	97.3/100.0	87.5/94.5	95.0/99.5	99.7/100	86.6/88.8	77.4/86.2	92.5/96.1
FatFormer [27]	99.9/100.0	97.2/99.8	98.8/99.9	99.5/100.0	99.3/100.0	99.8/100.0	99.4/100.0	93.2/98.0	**98.4/99.7**
SAFE [24]	99.9/100.0	98.0/99.9	98.6/100.0	89.7/95.9	98.9/99.8	99.9/100.0	91.5/97.2	93.1/97.5	96.2/98.8
CO-SPY [2]	74.7/78.1	63.9/70.2	59.7/63.1	71.6/83.9	58.5/55.8	62.1/94.3	69.6/83.4	65.7/79.7	65.7/76.1
FerretNet (Our)	99.9/100.0	98.0/100.0	98.5/100.0	92.6/98.5	98.8/99.9	99.1/100.0	91.4/99.8	89.2/96.7	95.9/99.3

Table 2: Accuracy and average precision comparisons with peer methods on Diffusion-6-cls test set.

Dataset	Wang [51]	LGrad [48]	Ojha [31]	FreqNet [49]	NPR [50]	FatFormer [27]	SAFE [24]	CO-SPY [2]	FerretNet
Dall-E	51.8/61.3	88.5/97.3	89.5/96.8	97.3/99.7	90.9/98.1	98.8/99.8	97.5/99.7	81.8/87.2	91.4/98.2
Guided	54.9/66.6	86.6/100.0	75.7/85.1	67.2/75.4	74.0/78.1	76.1/92.0	82.4/95.8	62.5/86.0	92.1/98.6
PNDM	50.8/90.3	69.8/98.5	75.3/92.5	85.2/99.9	97.5/100.0	99.3/100.0	78.9/98.6	53.0/55.6	96.9/100.0
VQ-Diffusion	50.0/71.0	96.3/100.0	83.5/97.7	100.0/100.0	100.0/100.0	100.0/100.0	100.0/100.0	71.9/71.5	99.9/100.0
Glide-50-27	54.2/76.0	90.7/95.1	91.1/97.4	86.6/95.8	97.5/99.5	94.7/99.4	96.6/99.2	69.1/74.6	97.2/99.7
Glide-100-10	53.3/72.9	89.4/94.9	90.1/97.0	87.8/96.0	97.8/99.5	94.2/99.2	97.3/99.4	76.6/81.6	97.9/99.9
Glide-100-27	53.0/71.3	87.4/93.2	90.7/97.2	84.4/95.6	97.4/99.5	94.4/99.1	95.8/98.9	73.5/78.2	97.3/99.7
LDM-100	51.9/63.7	94.8/99.2	90.5/97.0	97.8/99.9	98.0/99.6	98.7/99.9	98.8/100.0	82.7/86.9	98.8/100.0
LDM-200	52.0/64.5	94.2/99.1	90.2/97.1	97.4/99.9	98.2/99.6	98.6/99.8	98.8/100.0	83.1/87.5	98.8/100.0
LDM-200-CFG	51.6/63.1	95.9/99.2	77.3/88.6	97.3/99.9	98.0/99.5	94.9/99.1	98.7/99.9	85.3/91.0	98.5/99.9
Mean	52.4/70.1	89.4/97.7	85.4/94.6	90.1/96.2	94.9/97.3	95.0/98.8	94.5/99.1	73.9/80.0	**96.9/99.6**

We begin by evaluating FerretNet on GAN-based and Deepfake images using the ForenSynths test set. As shown in Table 1, it achieves an average accuracy (ACC) of 95.9%, outperforming lightweight baselines such as FreqNet [49] (91.5%) and NPR [50] (92.5%). Although FatFormer [27] reports a higher ACC of 98.4%, it relies on pre-trained CLIP weights, whereas FerretNet achieves competitive accuracy with significantly fewer parameters.

Next, on diffusion-generated images (Table 2), FerretNet attains an ACC of 96.9% and an AP of 99.6%, outperforming FatFormer [27] by 1.9 and 0.8 percentage points (pp), respectively. Other lightweight models such as NPR [50], FreqNet [49] and SAFE [24] perform less favorably, with ACC scores falling below 95.0%.

Table 3: Accuracy and average precision comparisons with state-of-the-art methods on Synthetic-Pop test set.

Methods	Openjourney	Proteus-0.3	RealVisXL-4.0	SD-3.5-Medium	SDXL-Turbo	YiffyMix	Mean
FreqNet [49]	56.3 / 63.6	44.0 / 41.2	59.4 / 66.6	78.5 / 86.8	77.5 / 86.0	74.3 / 84.4	65.0 / 71.4
NPR [50]	78.8 / 83.5	68.6 / 69.3	78.1 / 82.0	80.4 / 84.1	78.2 / 82.9	80.0 / 85.1	77.4 / 81.2
FatFormer [27]	58.8 / 65.4	93.9 / 97.6	49.0 / 41.7	81.9 / 89.1	58.7 / 65.3	80.9 / 89.9	70.5 / 74.8
SAFE [24]	94.7 / 99.3	99.2 / 99.9	97.9 / 99.8	98.1 / 99.7	98.1 / 99.8	99.5 / 99.9	97.9 / 99.7
CO-SPY [2]	92.4 / 97.6	88.8 / 93.0	79.0 / 86.5	80.9 / 87.8	79.9 / 88.3	92.9 / 97.5	85.6 / 91.8
FerretNet	98.4 / 99.7	98.6 / 99.7	98.8 / 99.9	97.2 / 99.6	98.9 / 100.0	97.8 / 99.7	**98.3 / 99.8**

Table 4: Performance comparisons with state-of-the-art methods across four distinct test sets. Throughput measurements were conducted on the Synthetic-Aesthetic test set. Upward arrows indicate that higher values are better, while downward arrows signify the opposite.

Methods	Ref	Image size	Params (M) ↓	FLOPs (G) ↓	FPS ↑	ACC / AP ↑
FreqNet [49]	AAAI 2024	256^2	1.85	2.58	200.2	79.2 / 86.8
NPR [50]	CVPR 2024	256^2	1.44	**2.29**	720.9	86.5 / 89.4
FatFormer [27]	CVPR 2024	224^2	492.59	269.92	88.6	86.1 / 91.0
SAFE [24]	KDD 2025	256^2	1.44	**2.29**	770.2	96.8 / 99.3
CO-SPY [2]	CVPR 2025	384^2	963.05	644.80	26.3	76.5 / 83.8
FerretNet (Ours)	-	256^2	**1.06**	2.38	**772.1**	**97.1 / 99.6**

We further evaluate performance on high-quality synthetic images using the Synthetic-Pop test set (Table 3). Some existing methods experience noticeable degradation; for example, NPR [50] achieves only 77.4% ACC and 81.2% AP. In contrast, FerretNet maintains 98.3% ACC and 99.8% AP, highlighting its robustness and reliability on visually realistic forgeries.

To evaluate real-world applicability, we tested FerretNet on four distinct test sets for both detection performance and efficiency. As shown in Table 4, FerretNet achieves 97.1% ACC and 99.6% AP with 1.06M parameters and 772.1 FPS on an RTX 4090. Notably, it outperforms FatFormer [27] by 11.0 and 8.6 pp in ACC and AP, respectively, while using only 0.2% of its parameters.

Finally, Appendix B provides a detailed analysis of specific success and failure cases, further clarifying the model's decision boundaries.

5.4 Ablation Study

Unless specified, all ablation results report the average ACC and AP across four datasets: ForenSynths, Diffusion-6-cls, Synthetic-Pop, and Synthetic-Aesthetic. More supplementary experiments see Appendix C.

5.4.1 Impact of Different Local Neighborhood Sizes

Table 5: Impact of the local neighborhood size.

Inputs	Size ($n \times n$)			ACC / AP on the Test set				
	3×3	5×5	7×7	ForenSynths	Diffusion-6-cls	Synthetic-Pop	Synthetic-Aesthetic	Mean
I				84.6 / 88.9	87.8 / 96.8	84.5 / 92.9	90.5 / 95.3	86.9 / 93.5
LPD	✓			**95.9 / 99.3**	**96.9 / 99.6**	**98.3 / 99.8**	**97.3 / 99.6**	**97.1 / 99.6**
LPD		✓		91.8 / 96.2	95.8 / 99.3	91.1 / 97.4	96.9 / 98.9	93.9 / 98.0
LPD			✓	82.4 / 90.6	85.2 / 93.6	78.6 / 91.9	85.0 / 94.4	82.8 / 92.6

Table 5 shows that LPD extracted using a 3×3 local neighborhood substantially enhances detection accuracy compared to raw input I. Average ACC improves from 86.9% to 97.1% (+10.2%), and AP rises from 93.5% to 99.6% (+6.1%). However, performance deteriorates as the neighborhood size increases. For instance, using a 7×7 neighborhood weakens feature discrimination and significantly reduces detection accuracy.

This trend aligns with the structural characteristics of generative models, which typically employ $2\times$ upsampling and small convolutional kernels (1×1 or 3×3). The 3×3 neighborhood is particularly effective in capturing localized decoding artifacts for two reasons: 1) It matches the scale of operations used in generative architectures, making it ideal for exposing subtle synthesis artifacts; 2) It captures local pixel variations while suppressing potential noise artifacts.

5.4.2 Impact of Center Pixel Processing Methods

According to Section 4.1, the neighborhood median $y_{i,j}$ should satisfy two key requirements: reducing the interference of center pixels in median computation, and ensuring the median value equals a real pixel value from the neighborhood set when possible, thus enhancing statistical correlation with the original image. To validate the effectiveness of zero-value masking strategy, we compared three center pixel processing methods:

1. **Zero-value Masking**: Set the center pixel to zero while keeping it in the set. This increases the probability that the median equals a real neighborhood pixel and reduces the center pixel's influence.

2. **Complete Exclusion**: Remove the center pixel entirely. This results in a non-existent pixel value (i.e., not from the original image), thereby weakening the dependency on the source image.

3. **Center Pixel Retention**: Keep the original center pixel, as in standard median filtering. This approach compromises the ability to detect local anomalies.

Table 6: Impact of center pixel processing methods on metrics (ACC/AP): average results across varying local neighborhood sizes.

Methods	3×3	5×5	7×7
Mask	**97.1 / 99.6**	**93.9 / 98.0**	82.8 / 92.6
Exclusion	95.3 / 98.8	90.5 / 96.3	86.4 / 93.6
Retention	93.3 / 97.6	89.7 / 96.3	**87.5 / 93.1**

The experimental results in Table 6 demonstrate that for local neighborhood sizes of 3×3 and 5×5, the zero-value masking achieves the highest detection accuracy, followed by the complete exclusion, with the center pixel retention yielding the lowest accuracy. These findings validate the effectiveness of the proposed strategy.

5.4.3 Impact of Neighborhood Statistic Selection

To verify the advantages of the neighborhood median-based feature extraction strategy in synthetic image detection, we designed three alternative methods: selecting the maximum, minimum, and average values from the neighborhood. The center pixel was masked by setting it to infinity, negative infinity, or zero, respectively, to reduce its influence on feature extraction. The experimental results in Table 7 show that, for both 3×3 and 5×5 local neighborhoods, the median strategy significantly outperforms the other methods.

Table 7: Impact of neighborhood statistic selection methods.

Methods	3×3	5×5	7×7
Max	93.6 / 97.9	86.8 / 94.3	88.9 / 94.8
Avg	92.2 / 97.2	88.2 / 94.4	**90.0 / 96.6**
Min	91.8 / 96.9	88.3 / 94.7	87.6 / 94.0
Med	**97.1 / 99.6**	**93.9 / 98.0**	82.8 / 92.6

5.4.4 Impact of Different Backbones

We evaluated ResNet50 [12], Xception [4], and our proposed FerretNet on both raw image I and LPD inputs. As shown in Table 8, FerretNet achieves competitive accuracy on raw images despite having significantly fewer parameters, and outperforms the other architectures when leveraging LPD. Across all backbones, replacing I with LPD consistently delivers accuracy gains with negligible effect on inference speed.

Table 8: Comparison of different backbones with and without LPD as input.

Methods	Params	w / o	Throughput ↑	ACC / AP ↑
Xception	20.8 M	×	730.5 Img/s	89.8 / 94.1
		✓	710.6 Img/s	**95.1 / 98.8**
ResNet50	23.5 M	×	755.4 Img/s	75.0 / 80.3
		✓	750.9 Img/s	**81.1 / 85.6**
FerretNet	1.1 M	×	777.8 Img/s	86.9 / 93.5
		✓	772.1 Img/s	**97.1 / 99.6**

6 Conclusion

This work presents a universal artifact representation framework and introduces FerretNet, a lightweight yet effective neural network for synthetic image detection. FerretNet achieves a remarkable 99.8% reduction in parameters compared to the state-of-the-art method FatFormer [27], while maintaining exceptional detection accuracy, reaching 97.1% on images generated by 22 different generative models. It demonstrates strong generalization capabilities and computational efficiency, outperforming existing approaches on high-quality synthetic datasets. Our contributions include a novel artifact representation approach and the introduction of the Synthetic-Pop dataset.

Limitations and Future Work. While the proposed method demonstrates robust performance, its effectiveness against compression-altered synthetic images has yet to be fully explored. Future work will focus on improving detection of compression-altered images and extending the approach to address challenges posed by emerging forms of synthetic media.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China (No. 62377028), the Guangdong Basic and Applied Basic Research Foundation (No. 2023B1515120064), the Science and Technology Program Project of the State Administration for Market Regulation (No. 2024MK097), the Science and Technology Planning Project of Guangdong (Nos. 2023A0505030013 and 2025A0505010018), the Fundamental Research Funds for the Central Universities (No. 21625102), and the Master Mentor Plan of Jinan University (No. YDXS2501).

References

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

[2] Siyuan Cheng, Lingjuan Lyu, Zhenting Wang, Xiangyu Zhang, and Vikash Sehwag. Co-spy: Combining semantic and pixel features to detect synthetic images by ai. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 13455–13465, 2025.

[3] 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.

[4] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017.

[5] Hyungjin Chung, Jeongsol Kim, Geon Yeong Park, Hyelin Nam, and Jong Chul Ye. Cfg++: Manifold-constrained classifier free guidance for diffusion models. *CoRR*, abs/2406.08070, 2024.

[6] dataautogpt3. Proteus V0.3. `https://huggingface.co/dataautogpt3/ProteusV0.3`, 2024.

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

[8] Ricard Durall, Margret Keuper, and Janis Keuper. Watch your up-convolution: Cnn based generative deep neural networks are failing to reproduce spectral distributions. In *CVPR*, pages 7890–7899, 2020.

[9] 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.

[10] 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.

[11] 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 *CVPR*, pages 10696–10706, 2022.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[13] Yang He, Ning Yu, Margret Keuper, and Mario Fritz. Beyond the spectrum: Detecting deepfakes via re-synthesis. *arXiv preprint arXiv:2105.14376*, 2021.

[14] Tianyang Hu, Fei Chen, Haonan Wang, Jiawei Li, Wenjia Wang, Jiacheng Sun, and Zhenguo Li. Complexity matters: Rethinking the latent space for generative modeling. *Advances in Neural Information Processing Systems*, 36:29558–29579, 2023.

[15] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. pmlr, 2015.

[16] Yonghyun Jeong, Doyeon Kim, Seungjai Min, Seongho Joe, Youngjune Gwon, and Jongwon Choi. Bihpf: Bilateral high-pass filters for robust deepfake detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 48–57, 2022.

[17] Yonghyun Jeong, Doyeon Kim, Youngmin Ro, and Jongwon Choi. Frepgan: Robust deepfake detection using frequency-level perturbations. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(1):1060–1068, Jun. 2022. doi: 10.1609/aaai.v36i1.19990. URL `https://ojs.aaai.org/index.php/AAAI/article/view/19990`.

[18] 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.

[19] 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.

[20] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, pages 4401–4410, 2019.

[21] 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.

[22] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. *Advances in neural information processing systems*, 34:852–863, 2021.

[23] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2022. URL `https://arxiv.org/abs/1312.6114`.

[24] Ouxiang Li, Jiayin Cai, Yanbin Hao, Xiaolong Jiang, Yao Hu, and Fuli Feng. Improving synthetic image detection towards generalization: An image transformation perspective. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, pages 2405–2414, 2025.

[25] 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.

[26] Chi Liu, Tianqing Zhu, Sheng Shen, and Wanlei Zhou. Towards robust gan-generated image detection: a multi-view completion representation. *arXiv preprint arXiv:2306.01364*, 2023.

[27] Huan Liu, Zichang Tan, Chuangchuang Tan, Yunchao Wei, Jingdong Wang, and Yao Zhao. Forgery-aware adaptive transformer for generalizable synthetic image detection. In *CVPR*, pages 10770–10780, 2024.

[28] Luping Liu, Yi Ren, Zhijie Lin, and Zhou Zhao. Pseudo numerical methods for diffusion models on manifolds. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=PlKWVd2yBkY`.

[29] 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.

[30] 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.

[31] Utkarsh Ojha, Yuheng Li, and Yong Jae Lee. Towards universal fake image detectors that generalize across generative models. In *CVPR*, pages 24480–24489, 2023.

[32] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *CVPR*, pages 2337–2346, 2019.

[33] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205, 2023.

[34] 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. URL `https://openreview.net/forum?id=di52zR8xgf`.

[35] John David Pressman, Katherine Crowson, and Simulacra Captions Contributors. Simulacra aesthetic captions. Technical Report Version 1.0, Stability AI, 2022. `https://github.com/JD-P/simulacra-aesthetic-captions`.

[36] PromptHero. Openjourney. `https://openjourney.art/`, 2023.

[37] Yuyang Qian, Guojun Yin, Lu Sheng, Zixuan Chen, and Jing Shao. Thinking in frequency: Face forgery detection by mining frequency-aware clues. In *European conference on computer vision*, pages 86–103. Springer, 2020.

[38] 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.

[39] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International conference on machine learning*, pages 8821–8831. Pmlr, 2021.

[40] Ali Razavi, Aaron van den Oord, Ben Poole, and Oriol Vinyals. Preventing posterior collapse with delta-VAEs. In *International Conference on Learning Representations*, 2019.

[41] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, pages 10684–10695, 2022.

[42] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015.

[43] Andreas Rossler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. Faceforensics++: Learning to detect manipulated facial images. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1–11, 2019.

[44] 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.

[45] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 35:25278–25294, 2022.

[46] SG161222. RealVisXL V4.0. `https://huggingface.co/SG161222/RealVisXL_V4.0`, 2024.

[47] Zenan Shi, Haipeng Chen, Long Chen, and Dong Zhang. Discrepancy-guided reconstruction learning for image forgery detection. *arXiv preprint arXiv:2304.13349*, 2023.

[48] Chuangchuang Tan, Yao Zhao, Shikui Wei, Guanghua Gu, and Yunchao Wei. Learning on gradients: Generalized artifacts representation for gan-generated images detection. In *CVPR*, pages 12105–12114, 2023.

[49] Chuangchuang Tan, Yao Zhao, Shikui Wei, Guanghua Gu, Ping Liu, and Yunchao Wei. Frequency-aware deepfake detection: Improving generalizability through frequency space domain learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(5):5052–5060, Mar. 2024. doi: 10.1609/aaai.v38i5.28310.

[50] 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 *CVPR*, pages 28130–28139, 2024.

[51] Sheng-Yu Wang, Oliver Wang, Richard Zhang, Andrew Owens, and Alexei A Efros. Cnn-generated images are surprisingly easy to spot... for now. In *CVPR*, pages 8695–8704, 2020.

[52] 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.

[53] Jonas Wulff and Antonio Torralba. Improving inversion and generation diversity in stylegan using a gaussianized latent space. *arXiv preprint arXiv:2009.06529*, 2020.

[54] Jingfeng Yao, Cheng Wang, Wenyu Liu, and Xinggang Wang. Fasterdit: Towards faster diffusion transformers training without architecture modification. *Advances in Neural Information Processing Systems*, 37:56166–56189, 2024.

[55] Yntec. YiffyMix V31. `https://huggingface.co/Yntec/YiffyMix`, 2023.

[56] 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.

[57] Zikai Zhou, Shitong Shao, Lichen Bai, Zhiqiang Xu, Bo Han, and Zeke Xie. Golden noise for diffusion models: A learning framework. *arXiv preprint arXiv:2411.09502*, 2024.

[58] 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.

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: The main contributions of this paper are clearly presented in the Abstract and Introduction, and are also summarized at the end of the Introduction section.

   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 did.

   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: [NA]

Justification: There are no theoretical results.

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: All relevant experimental details are provided, and the data along with the code will be released on GitHub after the paper is published.

   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: If the paper is accepted, the abstract in the final submitted version will include a link to the project.

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: See Subsection 5.1 and 5.2.

   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: [Yes]

   Justification: We did.

   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: [Yes]

   Justification: See Subsection 5.2.

   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: This work conforms with the 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 did.

    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: The paper does not involve the release of data or models with a high risk of misuse. Although the dataset includes synthetic images generated from COCO captions using diffusion models, these images are intended solely for research on synthetic image detection. The data contains no real human faces or personal information, and poses minimal risk of harmful dual use.

    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 did.

    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: [Yes]

    Justification: We will provide the documentation along with the new assets in our project page.

    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: This work does not involve 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: This work does not involve 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: We didn't.

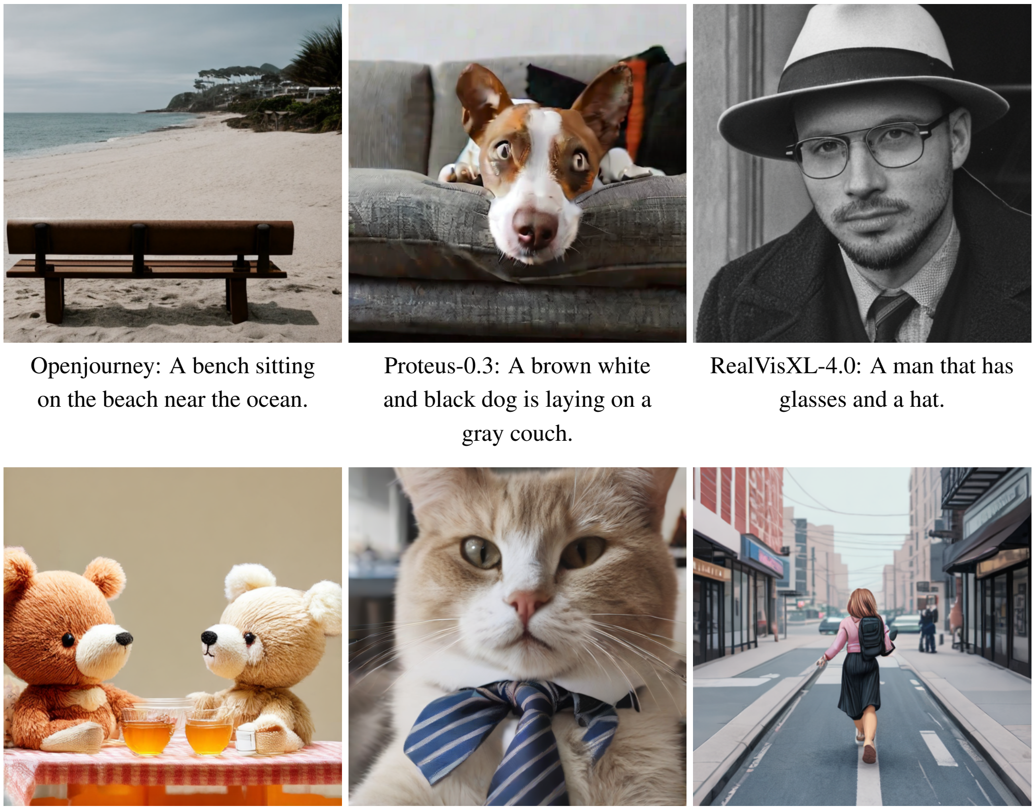
    Guidelines:

    - 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.

Appendix

A Details of Synthetic-Pop

To evaluate the practicality of our method across mainstream generated models, we construct a benchmark named **Synthetic-Pop**, which includes six widely used models: Openjourney [36], Proteus-0.3 [6], RealVisXL-4.0 [46], SD-3.5-Medium [9], SDXL-Turbo [34], and YiffyMix [55]. Images are generated using 5,000 captions randomly sampled from COCO [25], following the inference configurations recommended by each model. Representative examples are shown in Figure 4.



Openjourney: A bench sitting on the beach near the ocean.

Proteus-0.3: A brown white and black dog is laying on a gray couch.

RealVisXL-4.0: A man that has glasses and a hat.

SD-3.5-Medium: Two stuffed animals sit at a table with honey.

SDXL-Turbo: Cat sitting up with a fake tie around its neck.

YiffyMix: A woman walking down a street talking on a cell phone.

Figure 4: Examples of images generated by different models along with their corresponding text prompts. Each subfigure presents an image produced by a specific model, where the format "Model: Prompt" denotes the generating model and its input description.

B Visualization Analysis

B.1 Success Case Analysis

The effectiveness of LPD in distinguishing synthetic from real images stems from its ability to exploit intrinsic discrepancies in their local statistical structures. First, LPD captures subtle but systematic deviations introduced during the generative process. Although synthetic images may appear perceptually indistinguishable from real images, their micro-texture distributions and noise characteristics deviate from the stochastic sensor noise inherent to real image acquisition. As illustrated in the second and fifth rows of Figure 5, real images retain structured yet naturally

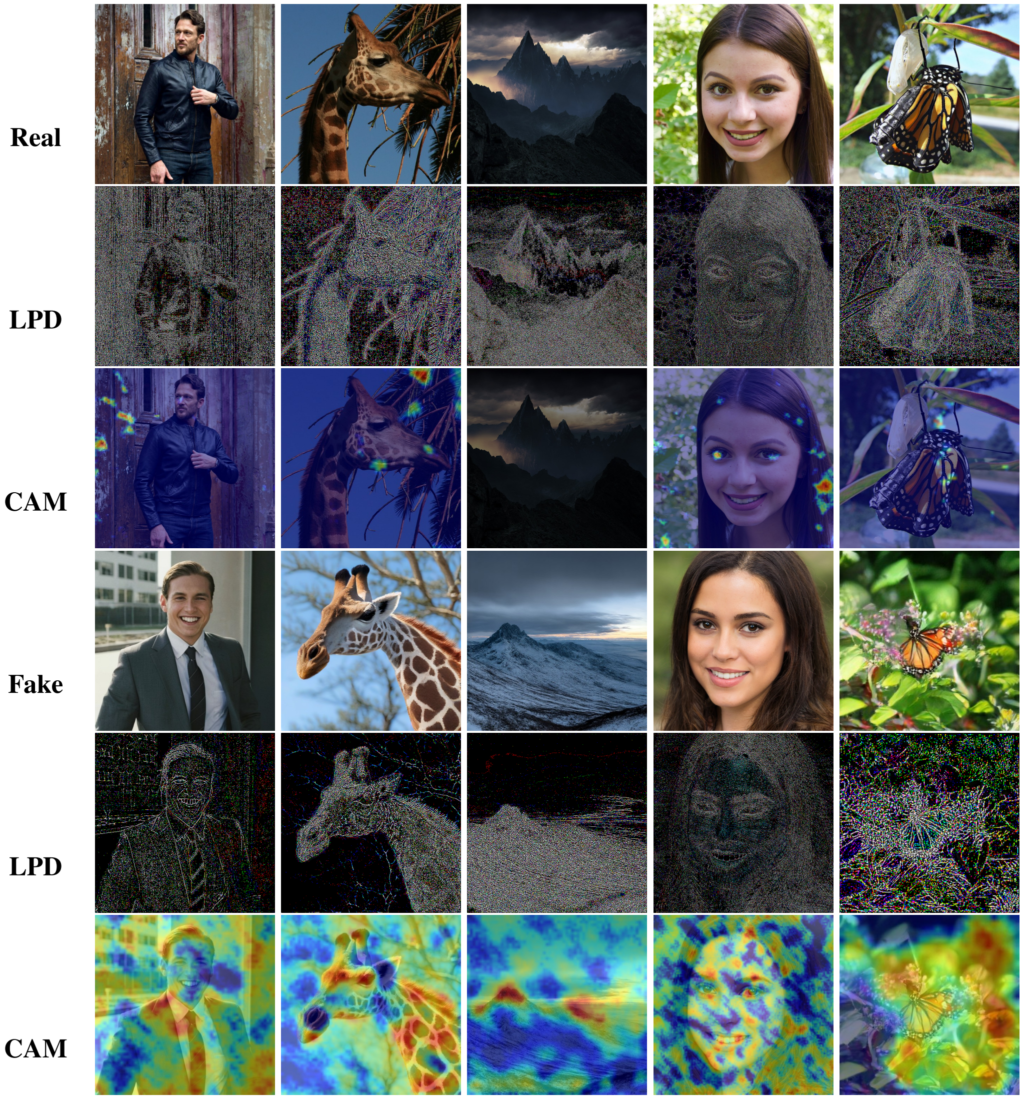


Figure 5: LPD and Grad-CAM visualizations of real and fake images.

irregular noise patterns, whereas synthetic counterparts exhibit overly smooth regions or artificial regularities—a direct consequence of the generative model's learned priors.

Second, LPD operates in a content-agnostic manner. It consistently reveals statistical inconsistencies across diverse semantic domains, including human portraits, wildlife, and landscapes. This invariance to semantic content highlights the signal-level nature of LPD, ensuring robustness against the rapid evolution of generative models that continue to enhance perceptual fidelity but still leave detectable low-level statistical artifacts.

Finally, the Grad-CAM visualizations in the third and sixth rows confirm that the model's attention aligns closely with LPD activation regions. Rather than focusing on semantic objects, FerretNet concentrates on areas exhibiting statistical anomalies, enabling consistent detection across diverse image categories. This strong alignment between LPD and Grad-CAM underpins the discriminative strength of our approach: the network remains nearly silent on authentic images while responding sharply to synthetic ones.

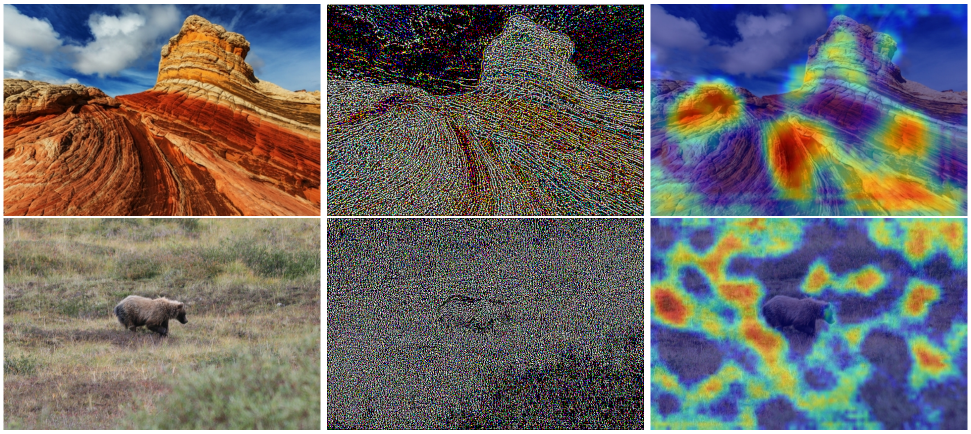


Figure 6: LPD and Grad-CAM visualizations of False Positive Case.

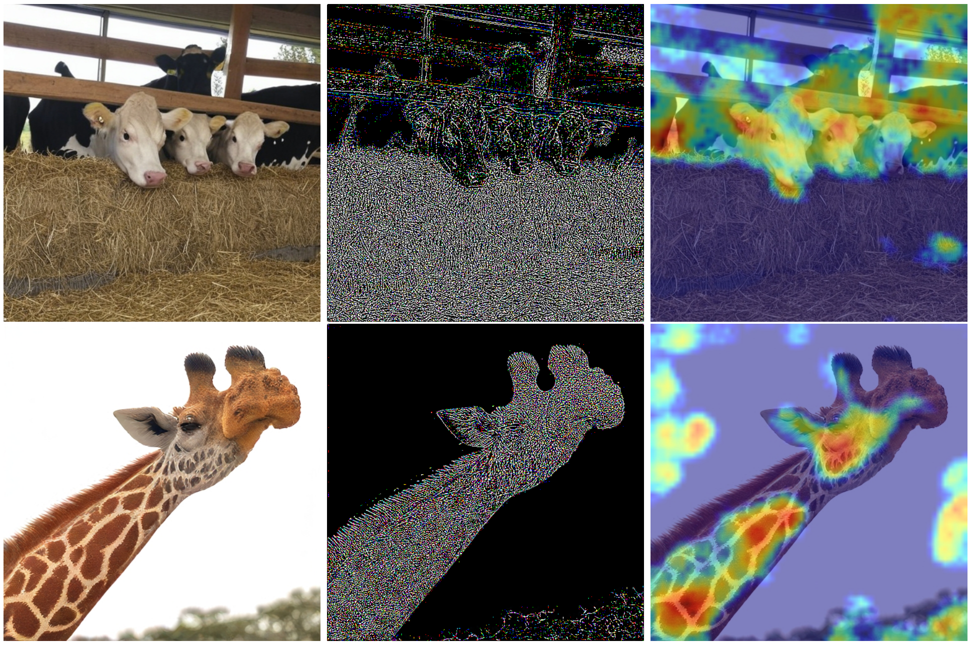


Figure 7: LPD and Grad-CAM visualizations of False Negative Case.

B.2 Failure Case Analysis

False Positive Case. Figure 6 illustrates representative real images are mistakenly classified as synthetic. Unlike the clear statistical distinctions observed in successful detections, these cases demonstrate that certain real-world textures can exhibit statistical patterns similar to those introduced by generative models.

In the first row, geological landscapes display highly regular and repetitive patterns, such as stratified rock formations with sharp color transitions and well-structured edges. These natural textures lead the LPD to capture statistical irregularities that resemble those typically found in generated content, particularly around boundary regions.

In the second row, the image of a bear on grassland demonstrates a similar issue. The network has learned to associate strong, structured statistical biases as primary indicators of artificial generation, resulting in misclassification. This reflects the model's high sensitivity to subtle statistical cues, while also highlighting its limitations when encountering rare real-world scenes that are themselves highly structured "outliers."

False Negative Case. Figure 7 illustrates representative synthetic images that were mistakenly classified as real. These cases reveal two distinct mechanisms by which AI-generated content can evade detection, highlighting limitations of LPD-based approaches.

In the synthetic cow image, the LPD map correctly highlights widespread, high-energy artifacts. Here, the failure is not in feature detection but in decision aggregation: the network sums all detected artifact signals to compute a final "fakeness" score. In this instance, despite numerous detected artifacts, their combined contribution did not exceed the threshold for a "fake" classification. This represents a rare boundary-case where a state-of-the-art generator produces content lying precisely on the “real” side of the learned decision boundary.

In contrast, the synthetic giraffe image exhibits a fundamentally different failure mode. Although Grad-CAM shows activation in textured regions, the overall signal is weak and insufficient for confident classification. A major factor is the clean, homogeneous sky background, which occupies a large portion of the image. Such smooth, featureless regions lack the local pattern variations necessary for LPD to detect pixel discontinuities or unnatural transitions. Consequently, the model receives insufficient discriminative signal, leading to misclassification.

C Additional Experiments

C.1 Scaling FerretNet

We scaled our model down (FerretNet-S) and up (FerretNet-L, 20x parameters). As shown in Table 9, making the model significantly larger results in a slight performance degradation. This confirms our hypothesis that for detecting low-level statistical artifacts, larger networks are prone to overfitting to training-specific patterns, which harms generalization. Our proposed FerretNet hits the "sweet spot" of being sufficiently expressive without the excess capacity.

Table 9: Ablation on scaling FerretNet. Results are mean ACC/AP across all test sets.

Method	Channels	Blocks	Parameters	Mean ACC / AP
FerretNet-S	(32, 64)	(2, 2)	0.13 M	93.1 / 97.6
FerretNet-B	**(96, 192)**	**(2, 2)**	**1.06 M**	**97.1 / 99.6**
FerretNet-L	(96, 192, 384, 768)	(2, 2, 6, 2)	21.51 M	96.6 / 99.4

C.2 Robustness to Common Post-Processing

We conducted a comprehensive robustness analysis on the ForenSynths [51] test set against JPEG compression, resizing, and rotation. For resizing, we used a stringent protocol with dynamic resolutions. As shown in Table 10, FerretNet demonstrates strong robustness, particularly against rotation, where it significantly outperforms the heavyweight FatFormer [27]. This provides strong evidence that LPD features, being based on local, orientation-agnostic statistics, are inherently immune to geometric transformations. While heavy JPEG compression remains a challenge for all lightweight detectors, FerretNet performs competitively.

Table 10: Robustness analysis on the ForenSynths test set against common post-processing attacks.

Method	No Attack	JPEG		Resize		Rotation
		(Q=100)	(Q=75)	(S=0.75)	(S=1.25)	D=[-45°, 45°]
FreqNet [49]	91.5/98.5	50.5/66.6	50.1/51.8	65.2/85.8	64.9/82.8	79.9/91.6
NPR [50]	92.5/96.1	55.0/59.3	50.0/49.1	83.9/84.9	78.9/81.8	86.7/90.7
FatFormer [27]	98.4/99.7	96.5/99.4	71.7/89.8	—	—	68.1/96.8
FerretNet (Ours)	**95.9/99.3**	**55.1/67.8**	**50.2/49.4**	**81.4/94.3**	**80.8/95.4**	**88.2/98.0**