# SUPPLEMENTARY MATERIAL

By leveraging neural rendering technologies based on NeRF and 3DGS, we create a wide array of realistic 3D scene representations and generate a multitude of synthesized 2D images from different perspectives. Moreover, through the combination of generative models with these advanced neural rendering methods, we generate highly sophisticated but fake images that incorporate combined artifacts. Unlike other existing datasets that largely focus on fake images generated by traditional generative models such as GANs or diffusion models, our *NeuroRenderedFake* dataset significantly extends the boundaries of a much-needed dataset for sophisticated fake image detection. This benchmark consists of over 2 million images, i.e., 512,972 authentic images and 1,653,881 highly sophisticated fake images. Therefore, it can serve as the largest collection of diverse images generated through advanced synthesis and neural rendering techniques.

This work is expected to have a significant positive societal impact, particularly benefiting the forensic community and media outlets. Our method can enhance the accurate and timely identification of real-look-like but fake images that are often found in our mailboxes or social media platforms. The development of accurate techniques to detect these images is crucial for addressing concerns related to security, privacy, and preserving harmony within our community. Importantly, the dataset created in this study does not involve human subjects or their personal data. It was synthesized using publicly accessible sources, which are clearly documented and shared alongside the dataset. Additionally, our synthesized videos and corresponding audio are also generated from publicly available content and are clearly labeled as synthesized (fake). Consequently, there is no risk associated with the release of our datasets, as comprehensive safeguards have been implemented to prevent any potential misuse.

Our supplementary material is organized as follows: **In Section A**, we provide details of the developed multimodal architecture, including its design and training specifications. **In Section B**, we present an analysis of spectral domain imprints, which serves as a supplement to Section 4.5 of the main text. **Section C** describes the detailed training and testing protocols, as well as the construction of the dataset. **Section D** discusses the scenario in which the frequency of fake images is manipulated and examines the corresponding impact on detection performance. **Section E** provides additional visualization samples from the NeuroRenderedFake dataset.

## A Motivation, Design and Training of Multimodal Architecture

### A.1 Additional Details on Network Training

The spatial branch is initialized with ViT weights pre-trained on the ImageNet dataset, except for the Ada-LoRA modules, which are randomly initialized. The spectral branch is initialized with ViT weights pre-trained by FFiT, which were acquired before, with its Ada-LoRA modules also randomly initialized. Both branches are fine-tuned using a learning rate of $1 \times 10^{-4}$ using the AdamW optimizer and a batch size of 256 on a single H100 GPU for 20 epochs, employing the BCEWithLogits loss. The fine-tuning follows the training protocols of groups $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$, and $\mathcal{D}$. After acquiring the fine-tuned spatial and spectral branches, we fix their parameters and fine-tune only the last GMU layer with the FC layer to obtain the optimal parameters for fake image classification. During each training stage, including the fine-tuning of the spatial branch, spectral branch, and GMU module, we use a class-balanced random sampler, following the approach described in [24], to balance the distribution of generated and real images over an epoch. The learning rate for fine-tuning the GMU layer is set to $1 \times 10^{-5}$, with a total of 20 training epochs using the AdamW optimizer.

**Data Augmentation for Training: 1) Spatial Branch:** Initially, all input images were resized to 256x256 pixels. Following this step, a cropping operation was performed where images were randomly cropped to 224x224 pixels. To further diversify the dataset, we applied horizontal and vertical flips with probabilities of 10% each. Additionally, Gaussian blur was applied with a 5% probability to simulate variations in image clarity. Image compression was also introduced with a 10% chance, varying the JPEG quality between 60 and 100 to mimic different levels of image degradation. For normalization, we used the mean and standard deviation values from the ImageNet dataset ([0.485, 0.456, 0.406] and [0.229, 0.224, 0.225], respectively) to standardize the pixel values. **2) Spectral Branch:** We adopt the same data augmentation settings for training the spectral branch as those used for the spatial branch. **3) Multimodal Training:** We adopt the same data augmentation settings for training the spectral branch as those used for the GMU module.

### A.2 The Design of Spectral Branch (FFiT)

We also investigate an unsupervised training approach to enable large models to extract comprehensive features from the Fourier spectrum's magnitude, thereby overcoming the challenges of reconstructing the spectrum due to its centrosymmetric properties. MAE [54] is a classical method to train large neural models in an unsupervised way. However, the centrosymmetric characteristic of the spectrum, wherein the amplitudes at positive frequencies are equivalent to those at the corresponding negative frequencies, thereby exhibiting symmetry about the zero frequency, can introduce adverse effects to the training if we use the same way as MAE to train the Transformer on the spectral domain. In Fig. Aa, a sample of the original magnitude of the spectrum is presented. Fig. Ab displays the mask utilized for patch masking during the inference stage, with the model that is trained using the original MAE-based training strategy. In this mask, white blocks indicate the patches that are to be masked during the patch embedding process, whereas black blocks represent the regions that should remain unmasked. Fig. Ac illustrates the reconstructed magnitude of the spectrum, based on the input from Fig. Aa and the mask shown in Fig. Ab, demonstrating a poor quality of reconstruction. In Fig. Ad, three representative types of regions of Fig. Ac are highlighted, and the following observations can be made:

**case (i):** When both a masked patch and its centrosymmetric counterpart are masked, the pre-trained model is unable to accurately reconstruct either. This indicates a limitation in the model's ability to infer information from the neighboring patches.

**case (ii):** In the case of masked patches for which the corresponding centrosymmetric patches remain unmasked, the pre-trained model demonstrates a capability to reconstruct these patches with high accuracy. This suggests that the model effectively captures and utilizes the centrosymmetric property of the spectrum during training.

**case (iii):** For the unmasked regions, it is evident that the pre-trained model fails to reconstruct them accurately. This finding is contrary to the expected behavior in the spatial domain, where an MAE-trained model typically succeeds in reconstructing unmasked areas.

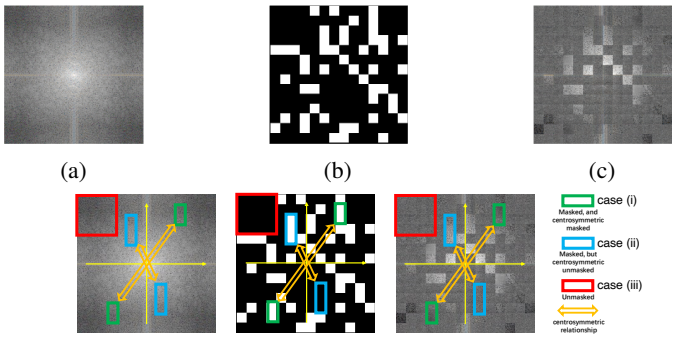



Figure A: Failure in spectral information extraction with the original MAE pre-training. (a) Input spectrum magnitude. (b) Patch embedding mask for inference. (c) Poor-quality reconstruction from (a) and (b). (d) Explanation of reconstructed patches in (c).

### A.3 Balancing the Weights of Various Masking Types

In the original MAE training process, the block-wise reconstruction loss $\mathcal{L}_{B(i,j)}$, which represents the reconstruction error for the $i^{\text{th}}$ row and $j^{\text{th}}$ column patch ($0 \leq i, j \leq N-1$) between the original input magnitude of spectrum $X$ and the reconstructed $X'$, is calculated as follows:

$$\mathcal{L}_{B(i,j)} = \sum_{m=0}^{W-1} \sum_{n=0}^{W-1} ||X(Wi+m, Wj+n) - X'(Wi+m, Wj+n)||^2 \tag{1}$$

where $X$ is divided into $N \times N$ patches ($N$ is an even number) in a Transformer-based architecture. Given that $X$ is of size $224 \times 224$ pixels and each patch is of size $W \times W$ pixels with $W = 16$, we have $N = 224/W = 14$. During the training process, masks are applied to these patches, compelling

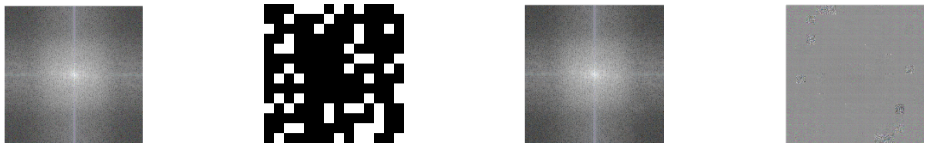
the model to reconstruct the patterns within the masked regions, thereby facilitating unsupervised learning.

The total loss function in the original MAE training is computed by summing the reconstruction losses over the masked blocks, i.e., those $B(i,j)$ that are masked. This approach has two key limitations that contribute to the failure to reconstruct the magnitude of the spectrum: 1) Ignorance of unmasked blocks in the loss function: the model is not penalized for any inaccuracies in the unmasked regions, which can lead to a lack of refinement in the overall reconstruction quality. 2) Overlooking centrosymmetric information: a masked block may have an unmasked centrosymmetric counterpart from which information can be easily copied. These limitations highlight the need for a more sophisticated loss function or training strategy that takes into account the unmasked regions and leverages the inherent symmetries within the spectral data to improve the reconstruction performance.

To address the limitations of the original MAE training process, we adopt a modified loss function that incorporates the focal loss mechanism. This approach aims to balance the influence of different masking cases, considering the special properties of the spectral magnitude. The probability of a patch being masked is assumed to be $r$. The loss function, denoted as $\mathcal{L}_{r\neq 0}(X, X'|r)$, is defined as follows:

$$\underset{r\neq 0}{\mathcal{L}}(X, X'|r) = -\frac{1}{N^2}\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\alpha_t(1-\mathcal{L}_{B(i,j)})^{\gamma}\log\mathcal{L}_{B(i,j)}, \tag{2}$$

for $B(i,j) \in$ masking case $t$ ($t \in \{1,2,3\}$), $\alpha_t$ is used to balance its weight according to the occurring frequency of the specific case.



(a) input (b) mask (c) reconstructed (d) difference

Figure B: During training, we adopt the loss function in Eq. (2) and fix $r$ to 0.3. For inference, the mask ratio of (b) is set to 0.25.

The focusing parameter $\gamma$ is designed to impose a greater penalty on less frequent examples. The balancing factor $\alpha_t$ is used to adjust the contribution of each class based on its effective number of samples. It is calculated as follows:

$$\alpha_t = P_t^{-1}/(P_1^{-1} + P_2^{-1} + P_3^{-1}), \quad t = 1,2,3 \tag{3}$$

where $P_1$, $P_2$, and $P_3$ refer to the expected probability for masking cases 1, 2, and 3.

The expected number of pairs of masked blocks for the three different cases, which are denoted as $E_1$, $E_2$, and $E_3$, can be computed as:

$$E_1 = \frac{N^2}{2}\times r^2, E_2 = \frac{N^2}{2}\times 2r\times(1-r), E_3 = \frac{N^2}{2}\times(1-r)^2 \tag{4}$$

Thus the probabilities $P_1$, $P_2$, and $P_3$ for three cases are:

$$P_1 = r^2, \quad P_2 = 2r\times(1-r), \quad P_3 = (1-r)^2 \tag{5}$$

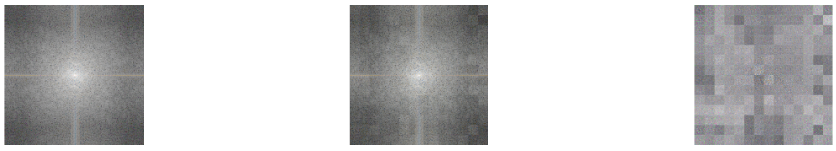
A reconstructed sample is presented in Fig. B, with the masking ratio during inference set to 0.25. The results indicate that regions corresponding to all three masking cases are reconstructed with high quality.

### A.4 Dynamic Masking Ratio for FFiT Training

Although the reconstructed sample in Fig. B demonstrates high-quality reconstruction with a masking ratio of 0.25 during inference, it can be observed that the global magnitude of the spectrum is not perfectly recovered as shown in Fig. C: when the model is trained with the same settings but evaluated

with a mask ratio of 0, the reconstructed magnitude of the spectrum (Fig. Cb) exhibits inconsistencies between blocks. Additionally, we find that if the mask ratio for inference significantly varies from the mask ratio during training, the performance of spectral reconstruction can be negatively influenced.

This observation inspires us to introduce a dynamic masking mechanism during training, where the mask ratio is randomly varied across different mini-batches. Specifically, we define three levels of masking: heavily masked, slightly masked, and not masked, with corresponding mask ratios $r_1$, $r_2$, and $r_3$ set to 0.3, 0.15, and 0.0, respectively. Within each mini-batch, the mask ratio is consistent (i.e., it is either $r_1$, $r_2$, or $r_3$), but the specific mask ratio used varies between different batches.



(a) input (b) reconstructed (c) difference of (a), (b)

Figure C: During training, adopt our loss function but set $r$ as a fixed value 0.3. During inference, using mask with ratio of 0.

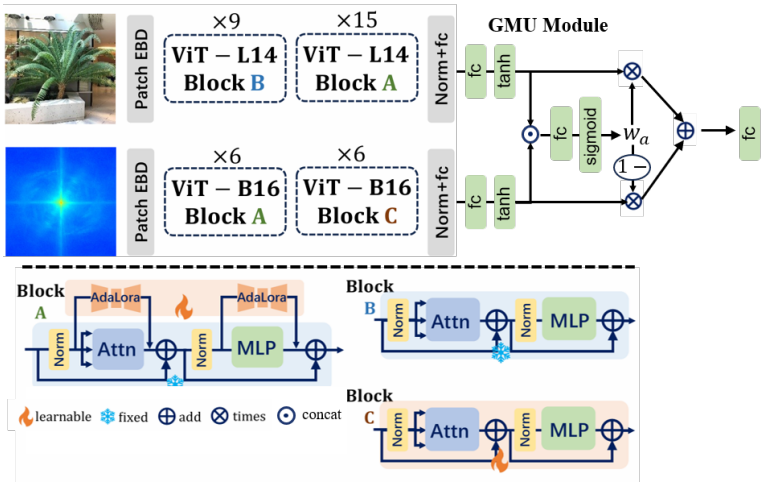



Figure D: Spatial-frequency architecture with the different blocks and distinct fine-tuning strategies across various network stages.

Different from $\mathcal{L}_{r\neq 0}(X, X'|r)$ for $r_1$, $r_2$, when $r_3 = 0$:

$$\mathcal{L}(X, X'|r_3 = 0) = \frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \mathcal{L}_{B(i,j)} \tag{6}$$

It was empirically observed that the order of magnitude for the loss function varies with different mask ratios. To mitigate such instability in training caused by significant fluctuations in gradient updates across batches for varying $r$ values, we introduce a scaling factor. Specifically, we compute the expected loss $\mathbb{E}[\mathcal{L}(X, X'|r)]$ for each $r$, and then scale the individual losses $\mathcal{L}(X, X'|r_1)$, $\mathcal{L}(X, X'|r_2)$, and $\mathcal{L}(X, X'|r_3)$ by the reciprocal of their respective expectations. This normalization ensures a more consistent gradient descent process, thereby enhancing the stability of the neural network's training across different mask ratio configurations.

To compute the expectation of the reconstruction loss, we assume that $\mathcal{L}_{B(i,j)}$ follows a $\chi$ distribution and is independent of the scenario type $t$. A detailed derivation proving that $\mathcal{L}_{B(i,j)}$ conforms to a $\chi$ distribution is provided in the Supp. A. Our goal is to determine $\mathbb{E}[\mathcal{L}(X, X'|r_k)]$ for $k = 1, 2, 3$, where $\mathbb{E}[\cdot]$ represents the expectation over the specified distribution.

For $k = 1$ and $k = 2$, the $r_k \neq 0$, we acquire:

$$\mathbb{E}(\mathcal{L}(X, X'|r_k)) = \sum_{t=1}^{3} P_t \mathbb{E}[\mathcal{L}(X, X'|r_k)|t] \tag{7}$$

For each $t$, the $\mathbb{E}[\mathcal{L}(X, X'|r_k)|t]$ is equal to:

$$\begin{aligned} &-\frac{1}{N^2}\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\alpha_t\mathbb{E}[(1-\mathcal{L}_{B(i,j)})^\gamma \log \mathcal{L}_{B(i,j)}] \\ &= -\alpha_t\mathbb{E}[(1-\mathcal{L}_B)^\gamma \log \mathcal{L}_B] \end{aligned} \tag{8}$$

Therefore, for $k = 1$ and $k = 2$,

$$\begin{aligned} \mathbb{E}(\mathcal{L}(X, X'|r_k)) &= -\left(\sum_{t=1}^{3} P_t\alpha_t\right)\mathbb{E}[(1-\mathcal{L}_B)^\gamma \log \mathcal{L}_B] \\ &= -\frac{3r_k^2(1-r_k)^2}{3r_k^2-3r_k+2}\mathbb{E}[(1-\mathcal{L}_B)^\gamma \log \mathcal{L}_B] \end{aligned} \tag{9}$$

where $\mathbb{E}[(1-\mathcal{L}_B)^\gamma \log \mathcal{L}_B]$ is equal to:

$$\int_0^\infty (1-x)^\gamma \log x \cdot \frac{1}{2}e^{\frac{-(x+\lambda)}{2}}\left(\frac{x}{\lambda}\right)^{\frac{k}{4}-\frac{1}{2}} I_{k/2-1}(\sqrt{\lambda x})\mathrm{d}x \tag{10}$$

where $I_\nu(z)$ is the modified Bessel function of the first kind of order $\nu$. Detailed steps are provided in the Supp. A.

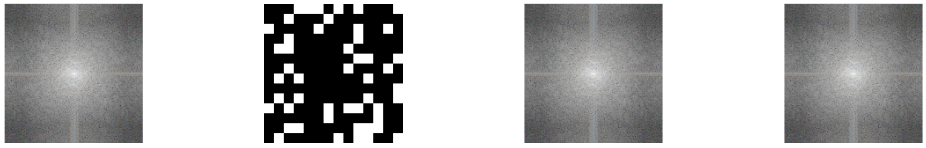
For $k = 3$, we can easily get:

$$\mathbb{E}[\mathcal{L}(X, X'|r_3)] = \mathbb{E}\left[\frac{1}{N^2}\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\mathcal{L}_{B(i,j)}\right] = \mathbb{E}[\mathcal{L}_B] \tag{11}$$

Therefore, for mask ratio of to 0.3, 0.15, 0, we have the scaled loss function $\mathcal{L}_k$ as:

$$\mathcal{L}(X, X'|r_k)/\mathbb{E}(\mathcal{L}(X, X'|r_k)) \tag{12}$$

We empirically observe that it is necessary to introduce dynamic masking ratios to capture the global reconstruction during pre-training of FFiT. After dynamically setting $r$ to $r_1, r_2, r_3 = 0.3, 0.15, 0$ respectively, the global reconstruction is perfect which can be observed from Fig. E.



(a) input (b) mask for (c) (c) from (a), (b) (d) from (a) only

Figure E: Dynamically set $r = 0.3, 0.15, 0$. (a) original magnitude, (b) mask with ratio of 0.25, (c) the reconstructed magnitude using (b) mask, (d) the reconstructed magnitude without mask.

During the pre-training of FFiT, we empirically set the focusing parameter $\gamma$ in the developed loss function to 2. The learning rate is set to $1 \times 10^{-4}$ with a batch size of 256 on a single H100 GPU. We employ early stopping, terminating the training when the reconstruction loss stagnates and does not improve for 5 consecutive epochs.

### A.5 Squared Euclidean Distance between Two Normally Distributed Vectors

In the main text section detailing the developed loss function for the frequency branch, we assume that the patch $\mathbf{X}$ representing the predicted magnitude of the frequency follows a normal distribution $\mathbf{X} \sim N(\boldsymbol{\mu}_1, \Sigma_1)$, while the patch $\mathbf{Y}$ representing the ground truth magnitude of the frequency follows $\mathbf{Y} \sim N(\boldsymbol{\mu}_2, \Sigma_2)$. Here, $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$ denote the mean vectors, and $\Sigma_1$ and $\Sigma_2$ represent the corresponding covariance matrices.

When computing the squared Euclidean distance between these vectors, we are essentially computing $\mathcal{L}_B = (\mathbf{X}-\mathbf{Y})^\top(\mathbf{X}-\mathbf{Y})$. Letting $\mathbf{Z} = \mathbf{X}-\mathbf{Y}$, then $\mathbf{Z}$ is also a multivariate normal random vector with mean $\boldsymbol{\mu}_Z = \boldsymbol{\mu}_1 - \boldsymbol{\mu}_2$ and covariance matrix $\Sigma_Z = \Sigma_1 + \Sigma_2$ (assuming independence between $\mathbf{X}$ and $\mathbf{Y}$).

The distribution of $\mathbf{Z}^\top\mathbf{Z}$ follows a generalized chi-squared distribution. Specifically, if $\boldsymbol{\mu}_1 = \boldsymbol{\mu}_2$ and $\Sigma_1 = \Sigma_2 = I$, where $I$ is the identity matrix, then $\mathbf{Z}^\top\mathbf{Z}$ would follow a standard chi-squared distribution with $d$ degrees of freedom ($d$ being the dimension of $\mathbf{X}$ and $\mathbf{Y}$). However, when $\boldsymbol{\mu}_1 \neq \boldsymbol{\mu}_2$ or $\Sigma_1 \neq \Sigma_2$, the distribution of $\mathbf{Z}^\top\mathbf{Z}$ is a noncentral chi-squared distribution, which is described as follows.

**Noncentral Chi-Squared Distribution**: For $\mathbf{Z}^\top\mathbf{Z}$, the degrees of freedom $k$ equals the dimension of $\mathbf{Z}$, and the noncentrality parameter $\lambda$ is given by:

$$\lambda = \boldsymbol{\mu}_Z^\top \Sigma_Z^{-1} \boldsymbol{\mu}_Z$$

Thus, the distribution can be written as:

$$\mathbf{Z}^\top\mathbf{Z} \sim \chi^2(k, \lambda)$$

The probability density function (PDF) of a noncentral chi-squared distribution with $k$ degrees of freedom and noncentrality parameter $\lambda$ is given by:

$$f(x; k, \lambda) = \frac{1}{2} e^{-(x+\lambda)/2} \left(\frac{x}{\lambda}\right)^{(k/4-1/2)} I_{k/2-1}(\sqrt{\lambda x})$$

where $I_\nu(z)$ is the modified Bessel function of the first kind of order $\nu$.

Let $\mathcal{L}_B$ be a variable that follows the noncentral chi-squared distribution defined above, therefore, $\mathbb{E}[(1-\mathcal{L}_B)^\gamma \log \mathcal{L}_B]$ can be computed as:

$$\int_0^\infty (1-x)^\gamma \log(x) f(x; k, \lambda)\, \mathrm{d}x$$

# B The Analysis of Spectral Domain Imprints

## B.1 Discrepancy of Magnitude of Spectrum between Real and Fake for NeuroRenderedFake

We visualize the averaged magnitude of the spectrum for real images and various fake images generated by exclusive neural rendering methods from the NeuroRenderedFake database, in Fig. F. Additionally, we illustrate the differences between the magnitude of the spectrum produced by these neural rendering methods and that of real images.

We visualize the 1D spectral energy distribution of real and fake images generated by exclusive neural rendering methods from the NeuralRenderedFake database in Fig. G. The azimuthal integrated 1D spectral energy $AI(\omega_k)$ for the input magnitude of spectrum $\mathcal{F}(I)$ is computed following the same definition in [41], and is given in Eq. (13):

$$AI(\omega_k) = \int_0^{2\pi} \left\| \mathcal{F}(I)(\omega_k \cdot \cos(\phi), \omega_k \cdot \sin(\phi)) \right\|^2 d\phi \quad \text{for } k = 0, \ldots, M/2 - 1, \tag{13}$$

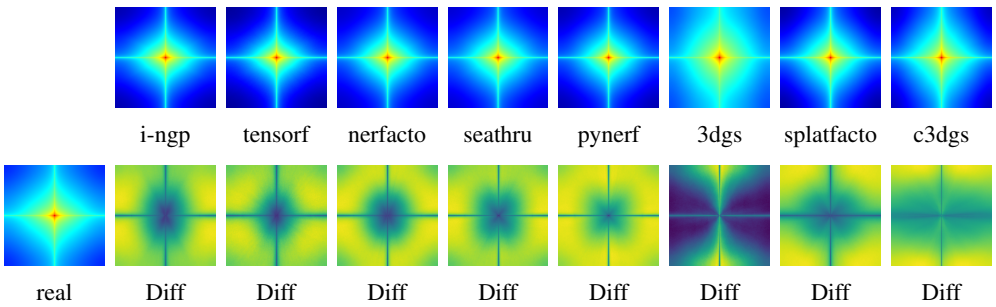



Figure F: The 2D spectral energy discrepancy among the exclusive neural rendered fake images.

We visualize the averaged magnitude of the spectrum for real images and various fake images from the test group 3-11 of the NeuroRenderedFake database, in Fig. H. In Fig. H, the first row shows

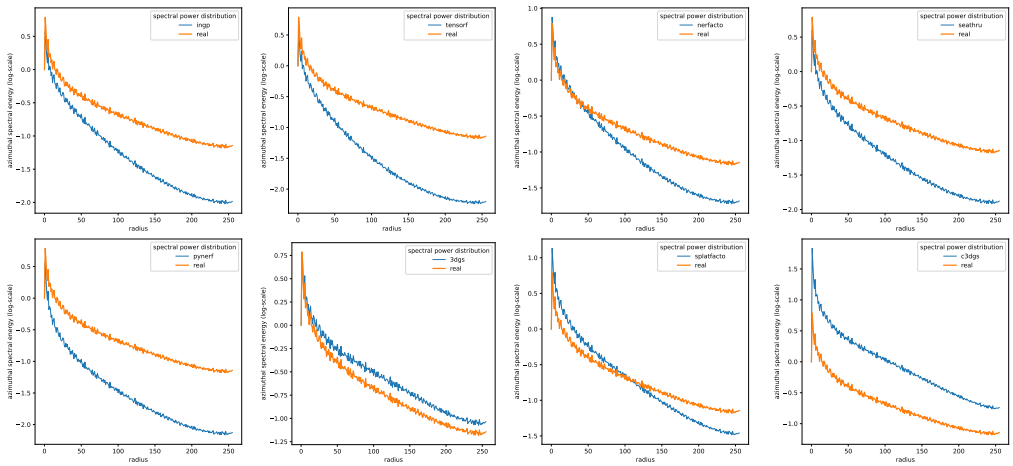
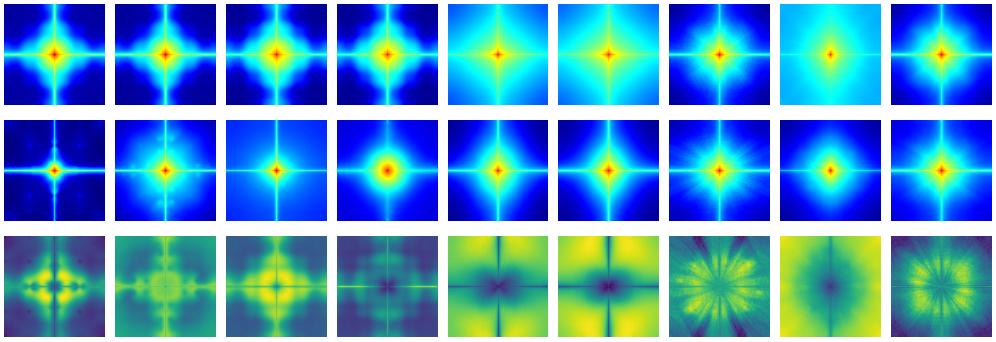



Figure G: The 1D spectral energy discrepancy among the exclusive neural rendered fake images.

the magnitude of the spectrum of real images, the second row shows that of fake images, and the third row displays the difference between the spectra of real and fake images. We also visualize their corresponding 1D spectral energy distribution in Fig. I.



[13] [14] [16] [17] [6] [23] [19] [20] [79]

Figure H: The 2D spectral energy discrepancy of the test group 3-11 of NeuroRenderedFake.

We define the summation of the 1D spectral energy discrepancy in Eq. (14) and give the corresponding discrepancy values for test group 3-11 in Tab. I.

$$discrepancy = \int_0^{M/2-1} |AI(\omega_k)_{real} - AI(\omega_k)_{fake}| \mathrm{d}k \tag{14}$$

Table I: Summation of 1D spectral energy discrepancies of test group 3-11 in NeuralRenderedFake

| pix2nerf [13] | sketchfacenerf [14] | dreamfusion [16] | gsgen [17] | in2n [6] | igs2gs [23] | genefacepp [19] | splattingavatar [20] | gaussiantalker [79] |
|---|---|---|---|---|---|---|---|---|
| 47.794 | 25.681 | 60.008 | 47.805 | 71.531 | 75.195 | 3.529 | 19.976 | 3.812 |

### B.2 The Contribution from Spectral Domain Imprints for Multimodal Detector

The contributions from spectral domain branch of the developed multimodal architecture is explicited represented by the learnable parameter $w_b$, and the distribution of $w_b$ varies between real and fake images, varies between different training groups among $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$ and $\mathcal{D}$, and also varies between different testing groups ranging from 1 to 11. Therefore, we display the violin plot to represent the distribution for the $w_b$ in Fig. J.

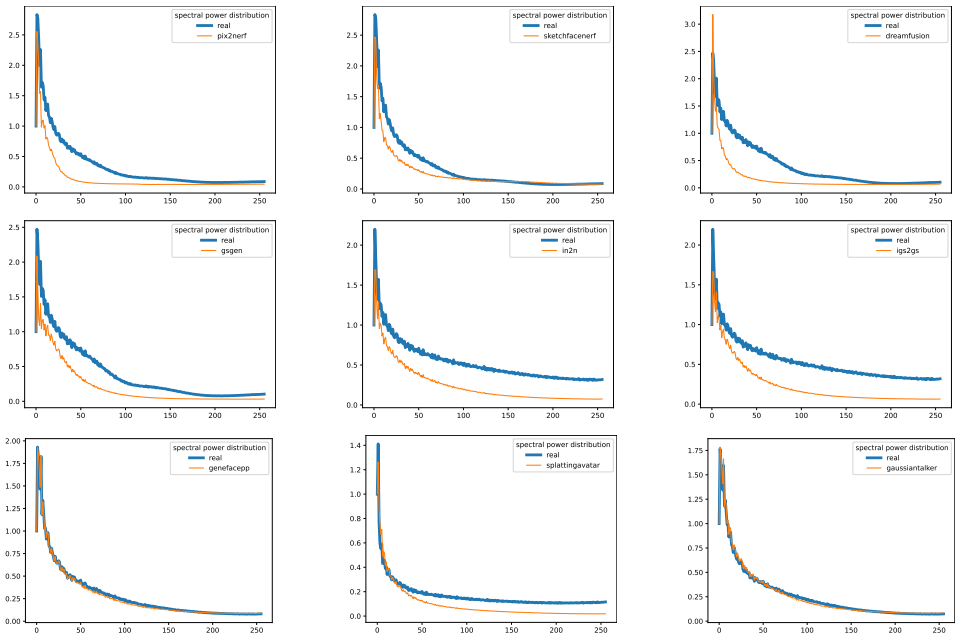



Figure I: The 1D spectral energy discrepancy for the test group 3-11 images in NeuroRenderedFake.

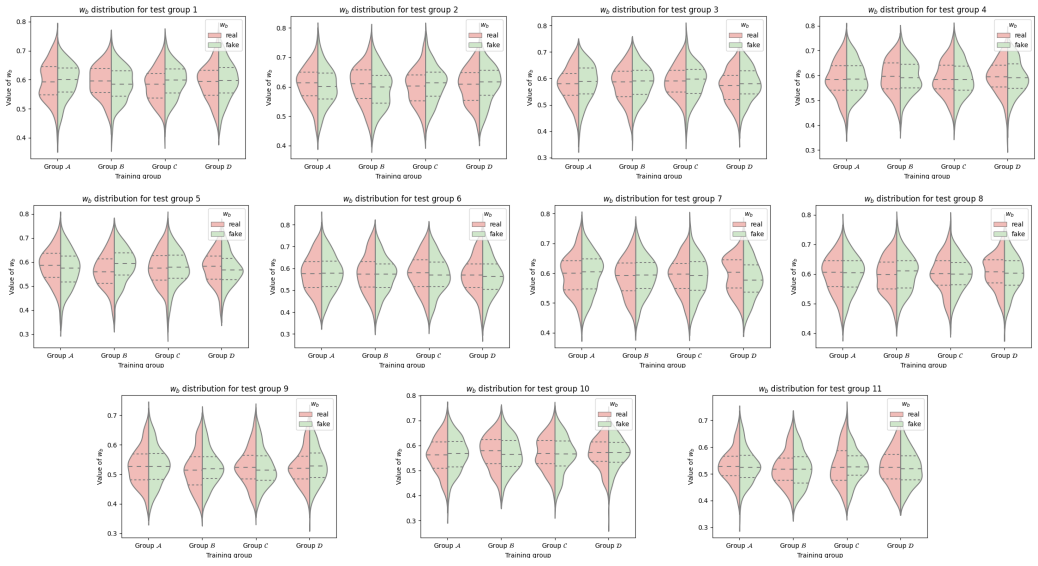



Figure J: The distribution of $w_b$ for different training and testing groups.

# C Details on the Dataset and Protocols

## C.1 Dataset Split Protocol for Training the Detectors

$\mathcal{A}$**:** For the real images, we randomly select 20,000 images from each of the category in afhq, celebahq, and lsun of ArtiFact [15] database, respectively, and collect all the 4,318 images from landscape class and all the 1,336 images from metfaces class. Therefore we acquire a total of 65,654 real images. For GAN-generated fake images, we collect 10k, 10k, 7k, 10k, 15k, and 15k images from the categories of BigGAN, Gans-former, GauGAN, ProjectedGAN, StyleGAN3, and Taming-Transformer, respectively.

$\mathcal{B}$**:** The real images are the same as $\mathcal{A}$ while a total of 66,896 fake images generated by 6 DMs are selected. Exactly, we collect 10k, 896, 20k, 6k, 20k, and 10k images from the categories of Glide, DDPM, Latent Diffusion, Palette, Stable Diffusion, and VQ Diffusion, respectively.

$\mathcal{C}$**:** For the real class, we use all the 69,377 real images in **A**$\sim$**J**. For the rendered class, we collect 12,000 images in **A**$\sim$**J** for method **I**, **II**, **III**, **IV**, **V**, respectively. Therefore, we acquire a total of 60,000 rendered images.

$\mathcal{D}$**:** For the real class, we use all the 69,377 real images in **A**$\sim$**J**. For the rendered class, we use all the 40,734 splatfacto-rendered images in **A**$\sim$**J** and all the 10,785 C3dgs-rendered images in **G**. Therefore, we acquire a total of 51,519 rendered images.

## C.2 Dataset Split protocol for Performance Evaluation

To evaluate the performance for group 1 and group 2, we select the scenes and the corresponding 2D images that never occur in the training dataset. The details are provided as follows:

Group 1 (**I** $\sim$ **V**): For the real class, we collect all the real images from **K**, **L**, **M**, **N**, **O**. For the fake class, we collect all the images from **K**, **L**, **M**, **N**, **O** rendered by the method **I**, **II**, **III**, **IV**, **V**, respectively.

Group 2 (**V** $\sim$ **VIII**): For the real class, we collect all the real images from **K**, **L**, **M**, **N**, **O**. For the fake class, we collect all the images from **K**, **L**, **M**, **N**, **O** rendered by the method **VI**, **VII**, **VIII**, respectively.

Besides evaluating the performance of unseen fake images generated by NeRF or 3DGS, further consideration is given to scenarios where traditional generative methods, such as GANs and DMs, are combined with neural rendering techniques in groups 3, 4, 5, and 6. Additionally, the use of editable neural rendering methods is explored. In groups 7 and 8, two representative methods capable of editing 3D scenes within their 3D representations are selected. A series of prompts for 3D editing are used, and these edited 3D scenes are then projected into 2D to acquire the fake images. Another important application of neural rendering technologies, digital human (avatar) generation, is also considered. In groups 9 and 10, these technologies are used to generate images of avatars, including both heads and full bodies. In group 11, frames sampled from Sora-generated videos, which exhibit realistic 3D representations within the video, are collected.

**Pix2NeRF[13]:** GAN+Nerf Image-to-image generation. For real class, We use all the 70,000 images in ffhq class of ArtiFact [15] database. For fake class, we render 96,000 ($1,000 \times 96$) images, where we reconstruct 1000 identities and render 96 images from different views for each identity.

**SketchFaceNeRF[14]:** GAN+Nerf Image-to-image generation. For real class, We use all the 70,000 images in ffhq class of ArtiFact [15] database. For fake class, we render 90,000 ($60 \times 60 \times 25$) images, where we use 60 sketches to style-transfer 60 identities and render 25 images from different views for each style-transferred head.

**DreamFusion[16]:** DM+Nerf text-to-image generation. For real class, we randomly collect 10,000 images in imagenet class of ArtiFact [15] database. For the fake class, we render 10,600 ($106 \times 100$) images, where we use 106 prompts for generation and render 100 images from different views for each generated 3D scene.

**GSGEN[17]:** DM+3DGS text-to-image generation. For real class, we randomly collect 10,000 images in imagenet class of ArtiFact [15] database. For the fake class, we render 9,540 ($106 \times 90$)

images, where we use 106 prompts for generation and render 90 images from different views for each generated 3D scene.

**Instruct-N2N[6]:** DM+Nerf image-to-image editing. For real class, we acquire all the 3,174 real images which are used to successfully train the nerfacto (**III**) of dataset **K**, **L**, **M**, **N**, **O**. For the fake class, we generate 40,559 edited images from the nerfacto-generated 3D scenes. The details of this Instruct-N2N dataset can be found in Tab. II.

**Instruct-GS2GS[23]:** DM+3DGS image-to-image editing. For real class, we acquire all the 3,174 real images which are used to successfully train the splatfacto (**VII**) of dataset **K**, **L**, **M**, **N**, **O**. For the fake class, we generate 40,559 edited images from the splatfacto-generated 3D scenes. The details of this Instruct-GS2GS dataset can be found in Tab. II.

Table II: Additional details on our instruct-N2N and instruct-GS2GS dataset (generated scenes/rendered images)

| | Prompts | instruct-N2N | | | | | instruct-GS2GS | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | **K** [3] | **L** [4] | **M** [6, 5] | **N** [7] | **O** [8] | **K** [3] | **L** [4] | **M** [6, 5] | **N** [7] | **O** [8] |
| prompts for human | Indian attire | ✗ | ✗ | 4/353 | ✗ | ✗ | ✗ | ✗ | 4/353 | ✗ | ✗ |
| | Mustache | ✗ | ✗ | 4/353 | ✗ | ✗ | ✗ | ✗ | 4/353 | ✗ | ✗ |
| | Bronze statue | ✗ | ✗ | 4/353 | ✗ | ✗ | ✗ | ✗ | 4/353 | ✗ | ✗ |
| | Joker makeup | ✗ | ✗ | 4/353 | ✗ | ✗ | ✗ | ✗ | 4/353 | ✗ | ✗ |
| | Gothic makeup | ✗ | ✗ | 4/353 | ✗ | ✗ | ✗ | ✗ | 4/353 | ✗ | ✗ |
| | Anime eyes | ✗ | ✗ | 4/353 | ✗ | ✗ | ✗ | ✗ | 4/353 | ✗ | ✗ |
| | Vintage sepia tone | ✗ | ✗ | 4/353 | ✗ | ✗ | ✗ | ✗ | 4/353 | ✗ | ✗ |
| | Neon lights | ✗ | ✗ | 4/353 | ✗ | ✗ | ✗ | ✗ | 4/353 | ✗ | ✗ |
| | Cyberpunk style | ✗ | ✗ | 4/353 | ✗ | ✗ | ✗ | ✗ | 4/353 | ✗ | ✗ |
| | Renaissance painting | ✗ | ✗ | 4/353 | ✗ | ✗ | ✗ | ✗ | 4/353 | ✗ | ✗ |
| | Pop art | ✗ | ✗ | 4/353 | ✗ | ✗ | ✗ | ✗ | 4/353 | ✗ | ✗ |
| | Tribal face paint | ✗ | ✗ | 4/353 | ✗ | ✗ | ✗ | ✗ | 4/353 | ✗ | ✗ |
| | Alien features | ✗ | ✗ | 4/353 | ✗ | ✗ | ✗ | ✗ | 4/353 | ✗ | ✗ |
| | Pixel art | ✗ | ✗ | 4/353 | ✗ | ✗ | ✗ | ✗ | 4/353 | ✗ | ✗ |
| | Watercolor effect | ✗ | ✗ | 4/353 | ✗ | ✗ | ✗ | ✗ | 4/353 | ✗ | ✗ |
| | Sketch drawing | ✗ | ✗ | 4/353 | ✗ | ✗ | ✗ | ✗ | 4/353 | ✗ | ✗ |
| | Surreal distortion | ✗ | ✗ | 4/353 | ✗ | ✗ | ✗ | ✗ | 4/353 | ✗ | ✗ |
| | Film noir | ✗ | ✗ | 4/353 | ✗ | ✗ | ✗ | ✗ | 4/353 | ✗ | ✗ |
| | Glitch art | ✗ | ✗ | 4/353 | ✗ | ✗ | ✗ | ✗ | 4/353 | ✗ | ✗ |
| prompts for nature | Snowy landscape | 9/1,940 | 8/305 | ✗ | 2/488 | 4/88 | 9/1,940 | 8/305 | ✗ | 2/488 | 4/88 |
| | summer style | 9/1,940 | 8/305 | ✗ | 2/488 | 4/88 | 9/1,940 | 8/305 | ✗ | 2/488 | 4/88 |
| | Autumn foliage | 9/1,940 | 8/305 | ✗ | 2/488 | 4/88 | 9/1,940 | 8/305 | ✗ | 2/488 | 4/88 |
| | spring style | 9/1,940 | 8/305 | ✗ | 2/488 | 4/88 | 9/1,940 | 8/305 | ✗ | 2/488 | 4/88 |
| | Tropical paradise | 9/1,940 | 8/305 | ✗ | 2/488 | 4/88 | 9/1,940 | 8/305 | ✗ | 2/488 | 4/88 |
| | Ancient style | 9/1,940 | 8/305 | ✗ | 2/488 | 4/88 | 9/1,940 | 8/305 | ✗ | 2/488 | 4/88 |
| | High brightness | 9/1,940 | 8/305 | ✗ | 2/488 | 4/88 | 9/1,940 | 8/305 | ✗ | 2/488 | 4/88 |
| | Halloween theme | 9/1,940 | 8/305 | ✗ | 2/488 | 4/88 | 9/1,940 | 8/305 | ✗ | 2/488 | 4/88 |
| | Cosmic style | 9/1,940 | 8/305 | ✗ | 2/488 | 4/88 | 9/1,940 | 8/305 | ✗ | 2/488 | 4/88 |
| | Industrial chic | 9/1,940 | 8/305 | ✗ | 2/488 | 4/88 | 9/1,940 | 8/305 | ✗ | 2/488 | 4/88 |
| | cyberpunk | 9/1,940 | 8/305 | ✗ | 2/488 | 4/88 | 9/1,940 | 8/305 | ✗ | 2/488 | 4/88 |
| | Baroque inspiration | 9/1,940 | 8/305 | ✗ | 2/488 | 4/88 | 9/1,940 | 8/305 | ✗ | 2/488 | 4/88 |

**GeneFace++[19]:** Speech-driven Avatar NeRF. We have 29 videos of different identities (Some source videos are the examples offered by GeneFace++, and the rest of the source videos are publicly available from YouTube, such as subject1, subject2, subject3, subject4, subject5, subject6, subject7, subject8, subject9, subject10, subject11, subject12, etc.) to train the 3D representation of the speaker's head. The original speaker's voice of different identities includes various languages, and such a multi-lingual property leads to the rich diversity of the dataset. We use two different ways to generate the fake speech video: 1). 14 identities to speak the contents from the other 13 identities by inputting the extracted audio, and therefore generate 182 (14 ×13) fake videos. 2) 15 identities to speak a predefined context by using 17 multi-langual, and therefore generate 255 (15x17) fake videos. For the real class, We collect all frames from each real video and generate 88,737 images. For the fake class, we collect all frames from each fake video and generate 452,653 images.

It can be observed from the performance evaluation on this dataset in Section 4 of our paper that the accuracy for fake detection on this dataset is quite low. Therefore, it's a quite challenging dataset that requires further work to address open problem on fake detection and is made available in the public domain to further advance much-needed research in this area.

**SplattingAvatar[20]:** Avatar 3DGS of head/body synthesis. For the real class, we use all the 33,728 images of 14 identities (10 identities are head and 4 identities are full body) provided by [20]. For the fake class, we generate 33,715 rendered images for 14 identities.

**GaussianTalker [79]**: Speech-driven Avatar 3DGS. We extend the real video dataset with one more identity and employ the same videos and same rule 1) and 2) to generate fake speech videos as those used to create GenFace++ fake videos. But only replace the Genface++ method to the Gaussiantalker that uses the 3D Gaussian Splatting. For the real class, we extract all frames from each real video and generate 94,810 images. Conversely, for the fake class, we extract all frames from each fake video and generate 411,401 images.

**SORA[22] frames:** For real class, we randomly acquire 60,000 images in coco class of ArtiFact [15] database. For the fake class, we collected 94 publicly released videos generated by SORA and randomly cropped a total of 60,531 images in the size of $512 \times 512$ pixels from the frames of these SORA-generated videos.

**Liveportrait [77]:** For the real class, we acquire a total of 59,260 frames from 28 videos. For the fake class, we acquire 88,466 frames from 234 generated fake videos. All-to-all matching protocol is adopted for evaluation.

**Runway [21]:** We randomly select 156 real videos from InternVid-10m, and 44 real videos from Youtube-8m dataset. For generating the fake video by the Runway Gen4-Turbo model, we randomly select one image from by randomly select other 161 videos from IngerVid-10m and 35 videos from Yotube-8m. We follow two method to generate the fake video: 1) For the image from the Youtube-8m, we use the image-to-video method, 2) For the image from the InternVid-10m, we use the text&image-to-video to generate the fake video, and the text is the caption of the video from the InternVid-10m. Because the generated fake video by Runway is 24 fps per second, and the total length is 5 seconds with about 121 frames. For the real class, we extract 121 frames from each real video and generate 23,334 images. Conversely, for the fake class, we extract all frames from each fake video and generate 23,353 images.

**106 Prompts Used for Generation of Stable-DreamFusion and GSGEN:** Acropolis of Athens, Desert cactus, Jamaican jerk chicken, Red panda, African elephant, Dolphin, Japanese ramen, Redwood forest, African lion, Dutch pancakes, Japanese sushi, Rhinoceros, Alpine meadow, Egyptian koshari, King cobra, Rose garden, Amazon jungle, Eiffel Tower, Koala bear, Russian borscht, American burger, Emperor penguin, Korean barbecue, Sagrada Familia, Angkor Wat, Ethiopian injera, Korean bibimbap, Saint Basil Cathedral, Arctic wolf, French bakery, Lavender fields, Siberian tiger, Argentine steak, French crepes, Lebanese falafel, Snow leopard, Australian steak, German sausages, Machu Picchu, Spanish tapas, Bald eagle, Giant panda, Malaysian satay, Statue of Liberty, Bamboo forest, Giraffe, Maple tree, Sunflower field, Belgian waffles, Golden Gate Bridge, Mexican churros, Swedish meatballs, Bengal tiger, Gray wolf, Mexican tacos, Swiss chocolate, Blue whale, Great Barrier Reef, Moroccan couscous, Sydney Opera House, Bonsai tree, Great Wall of China, Neuschwanstein Castle, Taj Mahal, Brandenburg Gate, Great white shark, Notre Dame Cathedral, Thai curry, Brazilian barbecue, Greek salad, Oak tree, Thai mango sticky rice, British fish and chips, Grizzly bear, Orca whale, Tower Bridge, Burj Khalifa, Hagia Sophia, Orchid garden, Tropical rainforest, Canadian poutine, Hawaiian poke bowl, Palm tree, Tulip garden, Cheetah, Hippopotamus, Peruvian ceviche, Turkish kebab, Cherry blossom tree, Indian curry, Petra Jordan, Venus flytrap, Chimpanzee, Indian samosas, Pine forest, Water lily pond, Chinese dumplings, Irish stew, Polar bear, Westminster Abbey, Colosseum, Italian gelato, Red fox, Coral reef, Italian pasta, Red kangaroo.

### C.3 Data Split Protocol for Cross-time Performance Evaluation

We especially define two self-evaluation protocols (PT-NN and PT-GG) to observe the performance of detectors which are trained using the past samples and evaluated for their future performance: i) PT-NN: train on NeRF-rendered images and test on unseen NeRF-rendered images, ii) PT-GG: train on 3DGS-rendered images and test on unseen 3DGS-rendered images. For PT-NN and PT-GG, the list of such additional evaluations is summarized in the Tab. III.

**PT-NN**: To train the detectors, we select the images from the methods of real, **I**, **II**, **III**, **IV**, and **V**, excluding those from categories designated for evaluation, which are summarized in the list of evaluation. To evaluate the performance of the detectors, we utilize all images from the methods of real, **I**, **II**, **III**, **IV**, and **V** located within the categories in the following list that is specified

Table III: Dataset split protocols for cross-times evaluation (PT-NN and PT-GG)

| | method | releasing date | training | | evaluation | |
|---|---|---|---|---|---|---|
| | | | scenes | images | scenes | images |
| PT-NN | real | / | 115 | 62,286 | 24 | 10,817 |
| | **I** [9] | 2022 Jan | 91 | 42,110 | 24 | 10,817 |
| | **II** [10] | 2022 Mar | 85 | 38,658 | 24 | 10,817 |
| | **III** [2] | 2023 Feb | 98 | 59,256 | 24 | 10,817 |
| | **IV** [8] | 2023 Apr | 73 | 23,740 | 24 | 10,817 |
| | **V** [11] | 2023 Nov | 60 | 16,640 | 24 | 10,817 |
| PT-GG | real | / | 20 | 1,786 | 9 | 1,940 |
| | **VI** [7] | 2023 Aug | 19 | 1,721 | 9 | 1,940 |
| | **VII** [2] | 2023 Sep | 18 | 1,234 | 9 | 1,940 |
| | **VIII** [12] | 2024 Feb | 19 | 1,721 | 9 | 1,940 |

for evaluation: nerfstudio/{bww-entrance, campanile, desolation, Egypt, kitchen, library, person, redwoods2, storefront, stump, vegetation}, record3d/bear, head/face, eyefultower/{office1b, office-view2, riverview}, mip360/{bicycle, bonsai, counter, flowers, kitchen, room, stump, treehill}.

**PT-GG** To train the detectors, we select the 1,786, 1,721, 1,234, 1,721 images from the **L**, **M**, **N**, **O** datasets for the methods of real, **VI**, **VII**, and **VIII**. To evaluate the performance of the detectors, we use the images from the **K** dataset for the methods of real, **VI**, **VII**, and **VIII**.

We summarize the evaluation performance on PT-NN and PT-GG for cross-time evaluation of NeRF and 3DGS, respectively. We sort the neural rendering methods according to the release date, for example, "$\leq$ nerfacto" means we use the fake images generated by i-ngp, tensorf and nerfacto for training. The results in [24] indicate that the testing performance of the detector on the recently released generative models can benefit from more exposure to fake images generated by newly developed methods during training. However, such a trend is not observed for NeRF and 3DGS-generated fake image detection.

## D Evaluation of Influence from Compromised Fake Images for Detection

One key challenge in the accurate detection of fake images is related to their modification or spectral alignment after synthesis, which makes it challenging to detect them with existing methods. As pointed out in [74], spectral-based fake detectors for generative models heavily rely on the differences in the one-dimensional spectral energy distribution between real and fake images. When the spectrum magnitude of fake images is compromised by replacing their 1D spectral energy distribution with the most similar one from a real image, the detection performance of spectral-based detectors can drop to nearly 50–50. In contrast, spatial-based detectors remain robust to such spectral domain manipulations. In this paper, we aim to investigate whether a similar phenomenon exists in neural rendering-based fake images—specifically, whether spatial-based detectors remain robust to fake images whose frequency content has been compromised, while spectral-based detectors are not.

In Fig. K, we visualize the evaluation of fake detection performance on test groups 1-11, where the fake images have been post-processed using methods developed in [74] to make the 1D spectral energy of the fake images resemble that of real images. Our findings indicate that this phenomenon is also present in fake images generated through neural rendering. Specifically, while spatial-based detectors maintain robustness with only a limited drop in performance, the performance of spectral-based detectors significantly decreases when the frequency of fake images is compromised.

## E Additional Details on Performance Evaluation

We select several representative methods for comparison. We don't compare with NPR [43] since reference [44] provides a fair comparison of UniFD [26] with NPR [43] and underlines the superiority of UniFD [26] over NPR [43]. We don't compare with [44] since this method utilizes the text extractor to extract the textual description from real images, and then input such text to a diffusion model-based generator to synthesize fake images. Then SVM classifier is trained based on real and such synthesized images. However, we cannot use the same approach to synthesize training samples

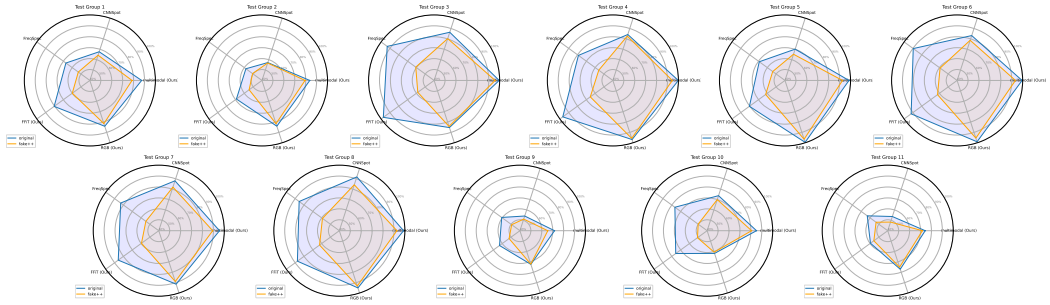


Figure K: Performance evaluation for compromised fake images (evaluated by AUROC).

by Nerf/3DGS methods for group $\mathcal{C}$ and $\mathcal{D}$. We don't compare with [49, 50, 47] since they utilize the inherent features of DM and they are DM-only methods.

### E.1 More Data Samples of NeuroRenderedFake

In Fig. L, we visualize fake image samples generated by exclusive NeRF/3DGS from NeuroRendered-Fake, along with their corresponding camera poses, all belonging to the same 3D scene. Similarly, images samples in Figs. M to Q present fake image samples from the database for the performance evaluation and are generated by a variety of 3D-realistic image synthesis methods, including editable NeRF, editable 3DGS, and combinations of NeRF/3DGS with traditional generative models, extending beyond simple NeRF-based and 3DGS-based approaches. We also give the samples of fake images generated by Sora [22] (test group 12), liveportrait [77] (test group 13) and Runway [21] (test group 14) in Fig. R, Fig. S and Fig. T.

Representative camera poses for synthesized images

Real Images for 3D Reconstruction

Fake images generated using the same camera poses by 3DGS [7]

Fake images generated using the same camera poses by Compressed 3DGS [12]

Fake images generated using the same camera poses by instant-NGP [9]

Fake images generated using the same camera poses by nerfacto [2]

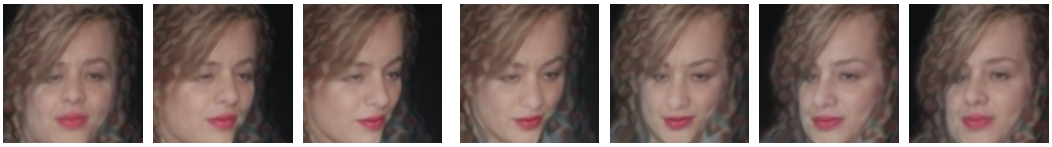
Fake images generated using the same camera poses by PyNeRF [11]

Fake images generated using the same camera poses by seaThru [8]

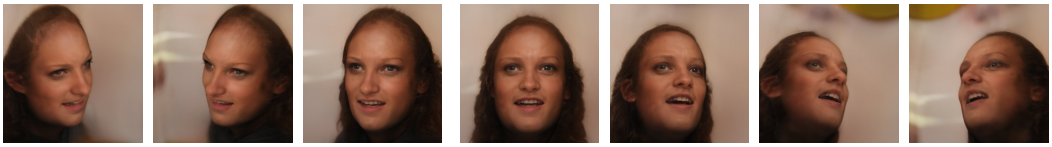
Fake images generated using the same camera poses by splatfacto [2]

Fake images generated using the same camera poses by tensorf [10]

Figure L: Samples of our acquired dataset for training. The 3D scenes are reconstructed by the different NeRF-based or 3DGS-based methods, from real images with the camera poses.



Pix2NeRF [13] (GAN combined with NeRF)



SketchFaceNeRF [14] (GAN combined with NeRF)

DreamFusion [16] (Diffusion model combined with NeRF). Prompt for generation: *Italian pasta*.

GSGEN [17] (Diffusion model combined with 3DGS). Prompt for generation: *American burger*.

Figure M: Samples of fake images synthesized by the methods that combine the generative models with neural rendering technologies.

real images

instruct NeRF-to-NeRF [6]. Prompt for style edit: *Autumn foliage*

instruct NeRF-to-NeRF [6]. Prompt for style edit: *Snowy Landscape*

instruct GS-to-GS [23]. Prompt for style edit: *Autumn foliage*

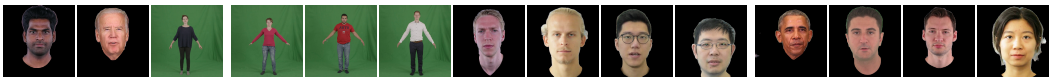
instruct GS-to-GS [23]. Prompt for style edit: *Snowy Landscape*

Figure N: Samples of fake images synthesized by the editable neural rendering technologies.


Fake generation process.

Samples of frames extracted from a synthesized fake video, in which the original person's face is manipulated to speak using another person's voice.

Figure O: Generation of the Geneface++ [33] frames for digital avatar synthesis based on NeRF.



(a) Real images for training SplattingAvatar [20] including heads and full bodies.

(b) Samples of the projected avatars we acquired in the database for each identity.

Figure P: Samples generated by SplattingAvatar [20] which represents neural rendering-based method for digital avatar synthesis.


Fake generation process.

Samples of frames extracted from a synthesized fake video, in which the original person's face is manipulated to speak using another person's voice.

Figure Q: Generation of the GaussianTalker [79] frames for digital avatar synthesis based on 3DGS.

Figure R: Samples of fake images cropped from Sora-generated videos [22].

Figure S: Samples of fake images cropped from liveportrait-generated videos [77].

Figure T: Samples of fake images cropped from Runway-generated videos [21].

## References


[1] S.-Y. Wang, O. Wang, R. Zhang, A. Owens, and A. A. Efros, "Cnn-generated images are surprisingly easy to spot... for now," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 8695–8704.

[2] M. Tancik, E. Weber, E. Ng, R. Li, B. Yi, T. Wang, A. Kristoffersen, J. Austin, K. Salahi, A. Ahuja *et al.*, "Nerfstudio: A modular framework for neural radiance field development," in *ACM SIGGRAPH 2023 Conference Proceedings*, 2023, pp. 1–12.

[3] J. T. Barron, B. Mildenhall, D. Verbin, P. P. Srinivasan, and P. Hedman, "Mip-nerf 360: Unbounded anti-aliased neural radiance fields," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5470–5479.

[4] B. Mildenhall, P. P. Srinivasan, R. Ortiz-Cayon, N. K. Kalantari, R. Ramamoorthi, R. Ng, and A. Kar, "Local light field fusion: Practical view synthesis with prescriptive sampling guidelines," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 4, pp. 1–14, 2019.

[5] C. Wang, R. Jiang, M. Chai, M. He, D. Chen, and J. Liao, "Nerf-art: Text-driven neural radiance fields stylization," *IEEE Transactions on Visualization and Computer Graphics*, 2023.

[6] A. Haque, M. Tancik, A. A. Efros, A. Holynski, and A. Kanazawa, "Instruct-nerf2nerf: Editing 3d scenes with instructions," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 19 740–19 750.

[7] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3d gaussian splatting for real-time radiance field rendering," *ACM Transactions on Graphics*, vol. 42, no. 4, pp. 1–14, 2023.

[8] D. Levy, A. Peleg, N. Pearl, D. Rosenbaum, D. Akkaynak, S. Korman, and T. Treibitz, "Seathru-nerf: Neural radiance fields in scattering media," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 56–65.

[9] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *ACM transactions on graphics (TOG)*, vol. 41, no. 4, pp. 1–15, 2022.

[10] A. Chen, Z. Xu, A. Geiger, J. Yu, and H. Su, "Tensorf: Tensorial radiance fields," in *European Conference on Computer Vision*. Springer, 2022, pp. 333–350.

[11] H. Turki, M. Zollhöfer, C. Richardt, and D. Ramanan, "Pynerf: Pyramidal neural radiance fields," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[12] J. C. Lee, D. Rho, X. Sun, J. H. Ko, and E. Park, "Compact 3d gaussian representation for radiance field," *arXiv preprint arXiv:2311.13681*, 2023.

[13] S. Cai, A. Obukhov, D. Dai, and L. Van Gool, "Pix2nerf: Unsupervised conditional p-gan for single image to neural radiance fields translation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 3981–3990.

[14] G. Lin, L. Feng-Lin, C. Shu-Yu, J. Kaiwen, L. Chunpeng, Y. Lai, and F. Hongbo, "Sketchfacenerf: Sketch-based facial generation and editing in neural radiance fields," *ACM Transactions on Graphics*, 2023.

[15] M. A. Rahman, B. Paul, N. H. Sarker, Z. I. A. Hakim, and S. A. Fattah, "Artifact: A large-scale dataset with artificial and factual images for generalizable and robust synthetic image detection," in *2023 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2023, pp. 2200–2204.

[16] B. Poole, A. Jain, J. T. Barron, and B. Mildenhall, "Dreamfusion: Text-to-3d using 2d diffusion," *arXiv preprint arXiv:2209.14988*, 2022.

[17] Z. Chen, F. Wang, and H. Liu, "Text-to-3d using gaussian splatting," *arXiv preprint arXiv:2309.16585*, 2023.

[18] Z. Ye, Z. Jiang, Y. Ren, J. Liu, J. He, and Z. Zhao, "Geneface: Generalized and high-fidelity audio-driven 3d talking face synthesis," *arXiv preprint arXiv:2301.13430*, 2023.

[19] Z. Ye, J. He, Z. Jiang, R. Huang, J. Huang, J. Liu, Y. Ren, X. Yin, Z. Ma, and Z. Zhao, "Geneface++: Generalized and stable real-time audio-driven 3d talking face generation," *arXiv preprint arXiv:2305.00787*, 2023.

[20] Z. Shao, Z. Wang, Z. Li, D. Wang, X. Lin, Y. Zhang, M. Fan, and Z. Wang, "Splattingavatar: Realistic real-time human avatars with mesh-embedded gaussian splatting," *arXiv preprint arXiv:2403.05087*, 2024.

[21] Runway Gen-4, 2025, https://runwayml.com/research/introducing-runway-gen-4.

[22] OpenAI, "Creating video from text," 2024, https://openai.com/index/sora/.

[23] J. Wu, J.-W. Bian, X. Li, G. Wang, I. Reid, P. Torr, and V. A. Prisacariu, "Gaussctrl: Multi-view consistent text-driven 3d gaussian splatting editing," *arXiv preprint arXiv:2403.08733*, 2024.

[24] D. C. Epstein, I. Jain, O. Wang, and R. Zhang, "Online detection of ai-generated images," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 382–392.

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

[26] U. Ojha, Y. Li, and Y. J. Lee, "Towards universal fake image detectors that generalize across generative models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 24 480–24 489.

[27] Y. Guo, K. Chen, S. Liang, Y.-J. Liu, H. Bao, and J. Zhang, "Ad-nerf: Audio driven neural radiance fields for talking head synthesis," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 5784–5794.

[28] S. Shen, W. Li, Z. Zhu, Y. Duan, J. Zhou, and J. Lu, "Learning dynamic facial radiance fields for few-shot talking head synthesis," in *European conference on computer vision*. Springer, 2022, pp. 666–682.

[29] J. Wang, J.-C. Xie, X. Li, F. Xu, C.-M. Pun, and H. Gao, "Gaussianhead: High-fidelity head avatars with learnable gaussian derivation," *arXiv preprint arXiv:2312.01632*, 2023.

[30] J. Tang, J. Ren, H. Zhou, Z. Liu, and G. Zeng, "Dreamgaussian: Generative gaussian splatting for efficient 3d content creation," *arXiv preprint arXiv:2309.16653*, 2023.

[31] Z. Liu, H. Wang, Y. Kang, and S. Wang, "Mixture of low-rank experts for transferable ai-generated image detection," *arXiv preprint arXiv:2404.04883*, 2024.

[32] Z. Wang, J. Bao, W. Zhou, W. Wang, H. Hu, H. Chen, and H. Li, "Dire for diffusion-generated image detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 22 445–22 455.

[33] L. Chai, D. Bau, S.-N. Lim, and P. Isola, "What makes fake images detectable? understanding properties that generalize," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVI 16*. Springer, 2020, pp. 103–120.

[34] L. Nataraj, T. M. Mohammed, S. Chandrasekaran, A. Flenner, J. H. Bappy, A. K. Roy-Chowdhury, and B. Manjunath, "Detecting gan generated fake images using co-occurrence matrices," *arXiv preprint arXiv:1903.06836*, 2019.

[35] R. He, S. Huang, X. Nie, T. Hui, L. Liu, J. Dai, J. Han, G. Li, and S. Liu, "Customize your nerf: Adaptive source driven 3d scene editing via local-global iterative training," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 6966–6975.

[36] H. Jung, S. Nam, N. Sarafianos, S. Yoo, A. Sorkine-Hornung, and R. Ranjan, "Geometry transfer for stylizing radiance fields," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 8565–8575.

[37] C. Ma, Y.-L. Liu, Z. Wang, W. Liu, X. Liu, and Z. Wang, "Humannerf-se: A simple yet effective approach to animate humannerf with diverse poses," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 1460–1470.

[38] W. Hong, M. Ding, W. Zheng, X. Liu, and J. Tang, "Cogvideo: Large-scale pretraining for text-to-video generation via transformers," *arXiv preprint arXiv:2205.15868*, 2022.

[39] O. Bar-Tal, H. Chefer, O. Tov, C. Herrmann, R. Paiss, S. Zada, A. Ephrat, J. Hur, Y. Li, T. Michaeli *et al.*, "Lumiere: A space-time diffusion model for video generation," *arXiv preprint arXiv:2401.12945*, 2024.

[40] T. Dzanic, K. Shah, and F. Witherden, "Fourier spectrum discrepancies in deep network generated images," *Advances in neural information processing systems*, vol. 33, pp. 3022–3032, 2020.

[41] R. Durall, M. Keuper, and J. Keuper, "Watch your up-convolution: Cnn based generative deep neural networks are failing to reproduce spectral distributions," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 7890–7899.

[42] H. Liu, Z. Tan, C. Tan, Y. Wei, J. Wang, and Y. Zhao, "Forgery-aware adaptive transformer for generalizable synthetic image detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 10 770–10 780.

[43] C. Tan, Y. Zhao, S. Wei, G. Gu, P. Liu, and Y. Wei, "Rethinking the up-sampling operations in cnn-based generative network for generalizable deepfake detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 28 130–28 139.

[44] D. Cozzolino, G. Poggi, R. Corvi, M. Nießner, and L. Verdoliva, "Raising the bar of ai-generated image detection with clip," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 4356–4366.

[45] C. T. Doloriel and N.-M. Cheung, "Frequency masking for universal deepfake detection," in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2024, pp. 13 466–13 470.

[46] J. Ricker, S. Damm, T. Holz, and A. Fischer, "Towards the detection of diffusion model deepfakes," *arXiv preprint arXiv:2210.14571*, 2022.

[47] J. Ricker, D. Lukovnikov, and A. Fischer, "Aeroblade: Training-free detection of latent diffusion images using autoencoder reconstruction error," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 9130–9140.

[48] Y. Li, Q. Bammey, M. Gardella, T. Nikoukhah, J.-M. Morel, M. Colom, and R. G. Von Gioi, "Masksim: Detection of synthetic images by masked spectrum similarity analysis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 3855–3865.

[49] G. Cazenavette, A. Sud, T. Leung, and B. Usman, "Fakeinversion: Learning to detect images from unseen text-to-image models by inverting stable diffusion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 10 759–10 769.

[50] Z. Zhou, K. Sun, Z. Chen, H. Kuang, X. Sun, and R. Ji, "Stealthdiffusion: Towards evading diffusion forensic detection through diffusion model," in *ACM Multimedia*, 2024.

[51] J. Arevalo, T. Solorio, M. Montes-y Gómez, and F. A. González, "Gated multimodal units for information fusion," *arXiv preprint arXiv:1702.01992*, 2017.

[52] I. Shumailov, Z. Shumaylov, Y. Zhao, N. Papernot, R. Anderson, and Y. Gal, "Ai models collapse when trained on recursively generated data," *Nature*, vol. 631, no. 8022, pp. 755–759, 2024.

[53] K. Fan, T. Liu, X. Qiu, Y. Wang, L. Huai, Z. Shangguan, S. Gou, F. Liu, Y. Fu, Y. Fu *et al.*, "Test-time linear out-of-distribution detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 23 752–23 761.

[54] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 16 000–16 009.

[55] Q. Zhang, M. Chen, A. Bukharin, N. Karampatziakis, P. He, Y. Cheng, W. Chen, and T. Zhao, "Adalora: Adaptive budget allocation for parameter-efficient fine-tuning," *arXiv preprint arXiv:2303.10512*, 2023.

[56] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of gans for improved quality, stability, and variation," in *ICLR*, 2018.

[57] G. Kwon and J. C. Ye, "Diagonal attention and style-based gan for content-style disentanglement in image generation and translation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 13 980–13 989.

[58] T. Karras, M. Aittala, S. Laine, E. Härkönen, J. Hellsten, J. Lehtinen, and T. Aila, "Alias-free generative adversarial networks," *Advances in neural information processing systems*, vol. 34, pp. 852–863, 2021.

[59] P. Dhariwal and A. Nichol, "Diffusion models beat gans on image synthesis," *Advances in neural information processing systems*, vol. 34, pp. 8780–8794, 2021.

[60] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10 684–10 695.

[61] A. Q. Nichol, P. Dhariwal, A. Ramesh, P. Shyam, P. Mishkin, B. Mcgrew, I. Sutskever, and M. Chen, "Glide: Towards photorealistic image generation and editing with text-guided diffusion models," in *International Conference on Machine Learning*. PMLR, 2022, pp. 16 784–16 804.

[62] M. Li, T. Cai, J. Cao, Q. Zhang, H. Cai, J. Bai, Y. Jia, K. Li, and S. Han, "Distrifusion: Distributed parallel inference for high-resolution diffusion models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 7183–7193.

[63] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "LoRA: Low-rank adaptation of large language models," in *International Conference on Learning Representations*, 2022. [Online]. Available: `https://openreview.net/forum?id=nZeVKeeFYf9`

[64] H. Dang, F. Liu, J. Stehouwer, X. Liu, and A. K. Jain, "On the detection of digital face manipulation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern recognition*, 2020, pp. 5781–5790.

[65] Y. He, B. Gan, S. Chen, Y. Zhou, G. Yin, L. Song, L. Sheng, J. Shao, and Z. Liu, "Forgerynet: A versatile benchmark for comprehensive forgery analysis," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 4360–4369.

[66] Y. Wang, Z. Huang, and X. Hong, "Benchmarking deepart detection," *arXiv preprint arXiv:2302.14475*, 2023.

[67] J. J. Bird and A. Lotfi, "Cifake: Image classification and explainable identification of ai-generated synthetic images," *IEEE Access*, 2024.

[68] M. Zhu, H. Chen, Q. Yan, X. Huang, G. Lin, W. Li, Z. Tu, H. Hu, J. Hu, and Y. Wang, "Genimage: A million-scale benchmark for detecting ai-generated image," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[69] X. Gao, X. Li, C. Zhang, Q. Zhang, Y. Cao, Y. Shan, and L. Quan, "Contex-human: Free-view rendering of human from a single image with texture-consistent synthesis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 10 084–10 094.

[70] C. Tan, Y. Zhao, S. Wei, G. Gu, and Y. Wei, "Learning on gradients: Generalized artifacts representation for gan-generated images detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 12 105–12 114.

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

[72] T. Zhao, X. Xu, M. Xu, H. Ding, Y. Xiong, and W. Xia, "Learning self-consistency for deepfake detection," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 15 023–15 033.

[73] M. J. Powell, "An efficient method for finding the minimum of a function of several variables without calculating derivatives," *The computer journal*, vol. 7, no. 2, pp. 155–162, 1964.

[74] C. Dong, A. Kumar, and E. Liu, "Think twice before detecting gan-generated fake images from their spectral domain imprints," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 7865–7874.

[75] C. Tan, Y. Zhao, S. Wei, G. Gu, P. Liu, and Y. Wei, "Frequency-aware deepfake detection: Improving generalizability through frequency space domain learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, 2024, pp. 5052–5060.

[76] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, "A kernel two-sample test," *The Journal of Machine Learning Research*, vol. 13, no. 1, pp. 723–773, 2012.

[77] J. Guo, D. Zhang, X. Liu, Z. Zhong, Y. Zhang, P. Wan, and D. Zhang, "Liveportrait: Efficient portrait animation with stitching and retargeting control," *arXiv preprint arXiv:2407.03168*, 2024.

[78] Y. Hong and J. Zhang, "Wildfake: A large-scale challenging dataset for ai-generated images detection," *arXiv preprint arXiv:2402.11843*, 2024.

[79] S. Kim, "Gaussiantalker: Real-time high-fidelity talking head synthesis with audio-driven 3d gaussian splatting," in *32th ACM Multimedia conference, MM 2024*. Association for Computing Machinery, Inc, 2024.

[80] C. Koutlis and S. Papadopoulos, "Leveraging representations from intermediate encoder-blocks for synthetic image detection," in *European Conference on Computer Vision*. Springer, 2024, pp. 394–411.

[81] X. Song, X. Guo, J. Zhang, Q. Li, L. BAI, X. Liu, G. Zhai, and X. Liu, "On learning multi-modal forgery representation for diffusion generated video detection," in *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

[82] L. Baraldi, F. Cocchi, M. Cornia, L. Baraldi, A. Nicolosi, and R. Cucchiara, "Contrasting deepfakes diffusion via contrastive learning and global-local similarities," in *European Conference on Computer Vision*. Springer, 2024, pp. 199–216.

[83] X. Guo, X. Liu, I. Masi, and X. Liu, "Language-guided hierarchical fine-grained image forgery detection and localization," *International Journal of Computer Vision*, pp. 1–22, 2024.

[84] A. Pal, J. Kruk, M. Phute, M. Bhattaram, D. Yang, D. H. Chau, and J. Hoffman, "Semi-truths: A large-scale dataset of ai-augmented images for evaluating robustness of ai-generated image detectors," *Advances in Neural Information Processing Systems*, vol. 37, pp. 118 025–118 051, 2024.

[85] H. Shum, Z. Zhou, and A. Kumar, "Detecting hyper-realistic videos generated by diffusion models via text-guided semantic enhancement," in *2025 IEEE international joint conference on biometrics (IJCB)*. IEEE, 2025.

[86] S. Mundra, G. J. A. Porcile, S. Marvaniya, J. R. Verbus, and H. Farid, "Exposing gan-generated profile photos from compact embeddings," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 884–892.

[87] The Hong Kong Polytechnic University NeuroRenderedFake Benchmark, 2025, `https://www4.comp.polyu.edu.hk/~csajaykr/neurorenderedfake.html`.