DiffusionFake: Enhancing Generalization in Deepfake Detection via Guided Stable Diffusion

Ke Sun[1], **Shen Chen**[2], **Taiping Yao**[2], **Hong Liu**[3]*,
Xiaoshuai Sun[1], **Shouhong Ding**[2], **Rongrong Ji**[1]
[1] Key Laboratory of Multimedia Trusted Perception and Efficient Computing,
Ministry of Education of China, Xiamen University, 361005, P.R. China.
[2] Youtu Lab, Tencent, P.R. China.
[3] Osaka University, Japan.

Abstract

The rapid progress of Deepfake technology has made face swapping highly realistic, raising concerns about the malicious use of fabricated facial content. Existing methods often struggle to generalize to unseen domains due to the diverse nature of facial manipulations. In this paper, we revisit the generation process and identify a universal principle: Deepfake images inherently contain information from both source and target identities, while genuine faces maintain a consistent identity. Building upon this insight, we introduce DiffusionFake, a novel plug-and-play framework that reverses the generative process of face forgeries to enhance the generalization of detection models. DiffusionFake achieves this by injecting the features extracted by the detection model into a frozen pre-trained Stable Diffusion model, compelling it to reconstruct the corresponding target and source images. This guided reconstruction process constrains the detection network to capture the source and target related features to facilitate the reconstruction, thereby learning rich and disentangled representations that are more resilient to unseen forgeries. Extensive experiments demonstrate that DiffusionFake significantly improves cross-domain generalization of various detector architectures without introducing additional parameters during inference. Code are available in https://github.com/skJack/DiffusionFake.git.

1 Introduction

The rapid progress in AI-generated content (AIGC) has led to the emergence of highly sophisticated forged face content, making it increasingly challenging for humans to distinguish between genuine and forged faces [31, 54, 8, 6]. Face swapping, also known as *Deepfakes*, is one of the most well-known techniques for generating forged facial images. It replaces the face of a target individual with that of a source person to create a seamless and realistic composite image [43]. The widespread proliferation of Deepfakes content on social media platforms has raised significant security concerns, including the spread of disinformation, fraud, and impersonation. As a result, developing effective and generalizable face forgery detection methods to counter these malicious attacks has become a critical challenge in the field of computer vision.

The growing diversity of facial forgery techniques has spurred interest in the general face forgery detection task [40, 37, 26], which aims to develop models that detect forgeries from unseen domains. Previous approaches primarily utilize forgery simulation [22, 35, 3, 38] to augment data by simulating various forgery traces, or framework engineering to enhance generalization through specialized designs like contrastive learning, attention mechanisms, and reconstruction learning [41, 50, 39, 2, 12].

*Corresponding Author.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

 https://doi.org/10.52202/079017-3218

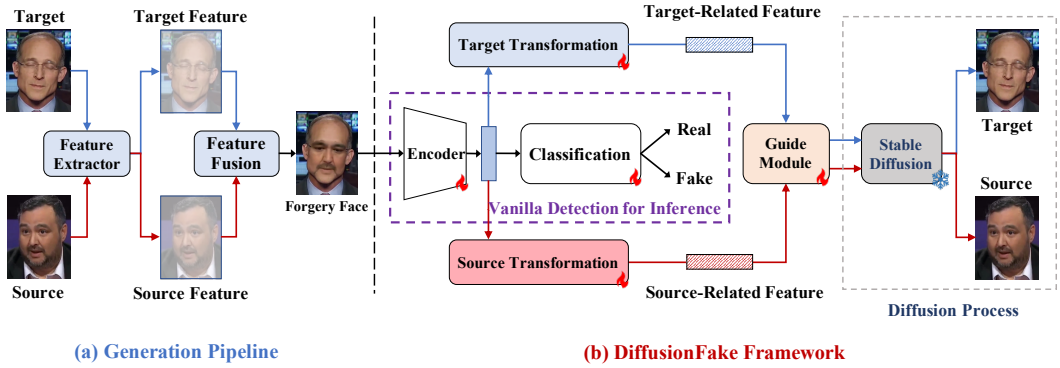

Figure 1: Pipeline of the generation process of Deepfake (a) and our proposed DiffusionFake (b).

However, their generalization capabilities remain limited due to the reliance on simulating specific forgery artifacts or designing specialized architectures tailored to certain manipulation techniques.

In this paper, we aim to identify the universal features common to all Deepfake faces by revisiting the generative process underlying forged face images. As depicted in Figure 1 (a), this process can be distilled into two key steps: (1) a feature extractor module captures salient features from both the source and target images; (2) these features are seamlessly fused through a generalized feature blending module to synthesize a novel Deepfake image. While the specific implementation of feature extraction and fusion may vary across different forgery methods, ranging from learning-based to graphics-based approaches, they all adhere to this fundamental generative paradigm.

Through this analysis, we uncover a crucial insight: Deepfake images inherently amalgamate information from both source and target faces, whereas genuine images maintain a consistent identity throughout. This amalgamated information can manifest as low-level artifacts, such as injection noise patterns and spectral discrepancies, or as high-level attributes, including facial expressions and mouth movements, depending on the specific forgery method employed.

Building upon this insight, we raise a question: Can we invert the generative process to extract and leverage the amalgamated source and target features, thereby enhancing the generalization capability of existing forgery detectors?

To answer this question, we introduce DiffusionFake, a novel plug-and-play framework that harnesses the power of Stable Diffusion to guide the forgery detector in learning disentangled source and target features inherent in Deepfakes. The core idea behind DiffusionFake is to inject the features extracted by the detector into a frozen pre-trained Stable Diffusion model, compelling the detector to capture the amalgamated source and target information by optimizing the features to reconstruct the corresponding source and target images.

As illustrated in Figure 1 (b), DiffusionFake is a plug-and-play framework that can be seamlessly integrated into existing forgery detectors. The features extracted by the encoder are first passed through the Target and Source Transformation modules, which filter and weight the features to obtain target and source-related representations. These features are then injected into the Stable Diffusion model using a Guide Module, leveraging its pre-trained knowledge to reconstruct the corresponding source and target images and optimize the feature representation.

During inference, only the encoder and classification modules are used, ensuring no additional parameters or computational overhead. By compelling the detector to learn more discriminative and generalized features, DiffusionFake enhances its ability to handle unseen forgeries without compromising efficiency. For example, when integrated with EfficientNet-B4, DiffusionFake improves AUC scores on unseen Celeb-DF dataset by around 10%, demonstrating its effectiveness in enhancing the generalization capability of existing detectors.

The main contributions of our work can be summarized as follows:

- We analyze Deepfake images from a generative perspective and propose a framework that leverages the reverse generation process to enhance the generalization capabilities of face forgery detectors.

- We introduce the DiffusionFake framework, a plug-and-play model that integrates a frozen pre-trained Stable Diffusion network to guide the forgery detector in learning disentangled source and target features inherent in Deepfakes, further enhancing the generalization.
- Extensive experimental validations demonstrate that the DiffusionFake framework significantly improves generalization capabilities across various architectures without introducing additional inference parameters.

2 Related Work

2.1 General Face Forgery Detection

General face forgery detection aims to improve the performance of forgery detectors on unseen domains and become one of the most critical issues in this field. Previous work to enhance generalization can be broadly divided into two categories: forgery simulation and framework engineering. The former utilizes data augmentation methods to simulate certain forgery traces, such as blending artifacts [13, 22, 35], Inconsistency between internal and external faces [51], subtle jitter and blur traces [19], and fine-grained facial disharmony [3, 38]. The latter improves network architectures or training procedures to help capture more generalized traces. Such methods approach the problem from different angles. Some employ attention mechanisms to enhance the capture of forgery traces [50, 39, 47, 33], while others improve generalization by jointly modeling frequency and spatial domains [29, 21, 28, 25]. Reconstruction-based methods enhance discriminability against unseen domain forgeries via modeling genuine faces [5, 2, 34]. Additionally, some approaches use implicit identity as a clue to improve the generalization of Deepfake faces [17, 9] and some explore the local and global relationships of unseen forgeries[4, 1, 45, 13, 10, 27]. Furthermore, decoupling methods[24, 32, 14, 20], such as ICT [11] and UCF [48], aim to enhance generalization by disentangling different facial information. Our DiffusionFake method addresses this by reversing the forgery process and leveraging pre-trained generative models to complete missing information, enhancing the capture of source-related and target-related features.

2.2 Diffusion Model

Diffusion models have emerged as a powerful framework for image generation and manipulation. The seminal work on Denoising Diffusion Probabilistic Models (DDPM) [15] introduced a novel approach to learn the data distribution by iteratively denoising a Gaussian noise signal. This process allows for high-quality image generation but requires a large number of sampling steps. To address this issue, the Denoising Diffusion Implicit Models (DDIM) [36] proposed a deterministic sampling process that significantly accelerates the generation process while maintaining image quality. Building upon these advancements, the Latent Diffusion Model (LDM) [30] combines the strengths of Variational Autoencoders (VAEs) [18] and diffusion models. By applying the diffusion process in the latent space learned by a VAE, LDM substantially reduces the computational cost and memory requirements during training. This innovative architecture has given rise to powerful AIGC pre-trained generative models, such as Stable Diffusion [2], which enable high-quality image generation and manipulation with unprecedented efficiency. Recent developments in controllable diffusion models have further expanded their applicability. ControlNet [49]introduces a mechanism to guide the image generation process by conditioning the diffusion model on additional control signals, such as segmentation masks or edge maps. Inspired by ControlNet, our DiffusionFake leverages a guide module to inject the source and target-related features into Stable Diffusion to reconstruct the corresponding images.

3 Methodology

Figure 2 illustrates the detailed framework of our proposed DiffusionFake method, which aims to enhance the generalization capability of forgery detectors by guiding the learning of amalgamated source and target features through a frozen pre-trained Stable Diffusion model. Specifically, the features extracted by the encoder are first filtered and weighted by the Feature Filter and Weight Modules to obtain source and target-related representations. These features are then injected into a

[2]https://stability.ai/news/stable-diffusion-public-release

 https://doi.org/10.52202/079017-3218

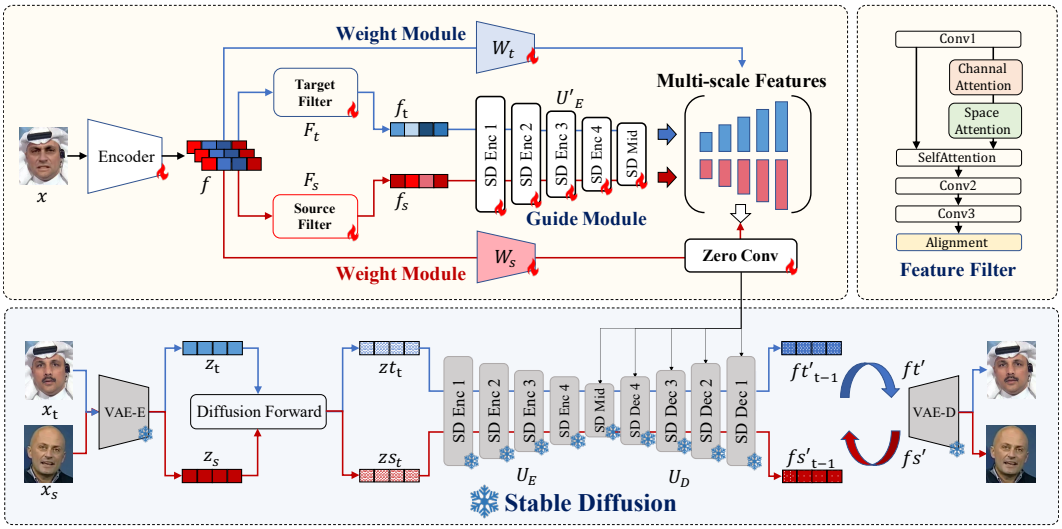

Figure 2: The details of the DiffusionFake method. The blue arrow represents the target branch, the red arrow represents the source branch, the ❄ represents the parameter frozen and does not participate in training, and the 🔥 represents the trainable module.

frozen pre-trained Stable Diffusion model via the Guide Module, which reconstructs the corresponding source and target images, compelling the encoder to learn rich and discriminative features.

3.1 Preliminaries

Diffusion Process. Denoising Diffusion Probabilistic Models (DDPMs) [15] are latent variable models that learn to generate data by reversing a gradual noising process. The forward diffusion process gradually adds Gaussian noise to the data x_0 according to a variance schedule $\beta_1, \ldots, \beta_T$, producing a sequence of noisy samples $x_1, \ldots, x_T$. The forward process can be described as:

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t\mathbf{I}) \tag{1}$$

The reverse denoising process learns to generate samples from the data distribution by starting with a Gaussian noise sample x_T and iteratively denoising it using a learned denoising function ϵ_θ. The reverse process is defined as:

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \sigma_t^2\mathbf{I}), \tag{2}$$

$$\mu_\theta(x_t, t) = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}t}}\epsilon_\theta(x_t, t)\right), \tag{3}$$

$\alpha_t = 1-\beta_t$, $\bar{\alpha}_t = \prod_{s=1}^{t}\alpha_s$, and $\sigma_t^2 = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t$. The training objective is to minimize the weighted sum of the denoising error at each step:

$$L = \mathbb{E}_{\epsilon\sim\mathcal{N}(0,1),t\sim[1,T]}\left[||\epsilon - \epsilon_\theta(x_t, t)||_2^2\right] \tag{4}$$

Stable Diffusion. The Stable Diffusion Model is a powerful pre-trained model with impressive generative capabilities, able to synthesize various types of images, including different types of human faces. Built upon the DDPM framework, the Stable Diffusion models employs a Latent Diffusion Model (LDM) [30] to reduce resource consumption. LDM applies the diffusion process in a learned latent space instead of pixel space, which is obtained by training an autoencoder. The training objective is:

$$L = \mathbb{E}_{\epsilon\sim\mathcal{N}(0,1),t\sim[1,T]}\left[||\epsilon - \epsilon_\theta(z_t, t)||_2^2\right], \tag{5}$$

where z_t is the latent representation encoded by the VAE encoder. This strategic application of latent space modeling not only enhances efficiency but also preserves the high quality of generated images.

3.2 Feature Transformation

Given an input image x and its corresponding label y, where $y = 0$ represents a real face and $y = 1$ represents a forged face, let x_s and x_t denote the corresponding source and target images from the training dataset, respectively. For real faces, x_s and x_t are identical to x. Let E be the encoder, and the extracted features be $f = E(x)$. To transform the extracted feature f into components that can guide the Stable Diffusion process, we first introduce two key modules: the *Feature Filter Module* and the *Weight Module*.

Feature Filter Module. The Feature Filter Module F is to extract source-related and target-related features from the encoded features f. To achieve this, we employ two filter networks, F_s and F_t, to obtain the source-related feature $f_s = F_s(f)$ and target-related feature $f_t = F_t(f)$, respectively.

As shown in Figure 2, the Feature Filter Module combines convolutional layers and attention mechanisms. The features first pass through a convolutional layer to transform the channels. Then channel-wise [16] and spatial-wise attention [46] are applied to adaptively weight and filter the features. These attention mechanisms help to emphasize the most relevant features while suppressing less informative ones, leading to more discriminative representations.

Subsequently, a Multi-Head Attention mechanism [44] is then applied to perform cross-attention between the original features (query) and the attention-filtered features (key and value). This operation captures long-range dependencies and enhances the receptive field, enabling more effective feature refinement. Finally, to ensure compatibility with the encoder of the Stable Diffusion model, we apply upsampling and pooling operations to align the feature dimensions.

Weight Module. The Weight Module W addresses the varying levels of source and target information embedded in different types of forged images. For example, Deepfakes may evenly blend source and target features, while expression-driven methods like NeuralTextures may predominantly feature target image information with minimal source information confined to specific regions like mouth movements. Uniformly feeding these into the guide module would be suboptimal.

To mitigate this, we use two separate weight modules, W_s and W_t, to estimate the information content for source and target features. Each weight module start with a pooling layer, following five MLP layers, and a sigmoid function finally outputs the weight. We train these modules using the similarity scores between the input image and its respective source and target images as ground truth.

Specifically, we encode x, x_t, and x_s using the pre-trained VAE-Encoder from Stable Diffusion to obtain latent representations z, z_t, and z_s. Such a well-pretrained model can effectively capture and quantify the differences between images, providing a reliable basis for measuring the similarity between the input image and its corresponding source and target images. The similarity scores between z and z_t, and between z and z_s, are computed and used to train the weight modules with mean squared error (MSE) loss as follows:

$$\mathcal{L}_{ws} = ||W_s(f) - \text{sim}(z, z_s)||_2^2 \tag{6}$$

$$\mathcal{L}_{wt} = ||W_t(f) - \text{sim}(z, z_t)||_2^2 \tag{7}$$

where $\text{sim}(a, b) = \frac{a \cdot b}{|a||b|}$ denotes the cosine similarity between vectors a and b.

By dynamically adjusting the influence of source and target features during the diffusion process, the Weight Module ensures optimal guidance for the Stable Diffusion model, thereby enhancing the encoder's ability to extract generalized features suitable for detecting a wide range of forgeries.

3.3 Guide Module

The Guide Module is designed to inject the source-related and target-related features into the frozen pre-trained Stable Diffusion model to guide the reconstruction of the source and target images. As illustrated in Figure 2, the Guide Module employs trainable copy and zero convolution layers for feature injection, inspired by ControlNet [49].

Let $U_E(\cdot)$ and $U_D(\cdot)$ denote the neural block of the encoder and decoder in the U-Net ϵ_θ network of the Stable Diffusion model, respectively. The Guide Module first creates a trainable copy of $U_E(\cdot)$, denoted as $U_E^{'}(\cdot)$. The source-related feature f_s and target-related feature f_t are then independently fed into $U_E^{'}(;)$. The resulting features are combined with the corresponding features from the locked

 https://doi.org/10.52202/079017-3218

model's decoder $U_D(;)$ using zero convolution layers $Z(\cdot)$, which are 1×1 convolutional layers with weights and biases initialized to zeros. This initialization minimizes the impact on the pre-trained model at the beginning of training, stabilizing the training process [49]. The final output of the Guide Module can be summarized as:

$$fs^{'} = U_D(U_E(z_s)) + Z(U_E^{'}(f_s)) \times W_s(f) \tag{8}$$

$$ft^{'} = U_D(U_E(z_t)) + Z(U_E^{'}(f_t)) \times W_t(f) \tag{9}$$

where z_s and z_t are the latent representations of the source and target images, respectively, obtained from the pre-trained VAE-Encoder of Stable Diffusion, and $W_s(f)$ and $W_t(f)$ are the weights computed by the Weight Module.

Unlike ControlNet, which aims to control the generated results of the diffusion model, our objective is to optimize the features f by fixing the output and encouraging the capture of more generalizable and disentangled features. By guiding the reconstruction of the source and target images using the respective features, the Guide Module facilitates the learning of rich and discriminative representations that enhance the performance of the forgery detector across various domains and attack types.

During training, we follow a process similar to the LDM [30], gradually executing the diffusion process, including the time step t, to guide the reconstruction of the source and target images. At each time step t, the model learns to predict the noise ϵ that was added to the latent representation of the source or target image. The overall learning objective for the source and target diffusion models can be formulated as:

$$L_s = \mathbb{E}_{\epsilon \sim \mathcal{N}(0,1), t \sim [1,T]} \left[||\epsilon - \epsilon_\theta(fs_t^{'}, t)||_2^2 \right] \tag{10}$$

$$L_t = \mathbb{E}_{\epsilon \sim \mathcal{N}(0,1), t \sim [1,T]} \left[||\epsilon - \epsilon_\theta(ft_t^{'}, t)||_2^2 \right] \tag{11}$$

where $fs_t^{'}$ and $ft_t^{'}$ represent the features within the embedding of time step t.

3.4 Loss Function and Inference

We apply a simple binary classification head to the extracted feature f to obtain the predicted label y', which is calculated via typical cross-entropy loss as follows:

$$L_{ce} = -\left[y \log y' + (1 - y) \log(1 - y') \right] \tag{12}$$

Thus, the final loss function combines Eq.12 with the losses from our *Weight module* and *Gudie module*, which is defined as follows:

$$L = L_{ce} + \lambda_s L_s + \lambda_t L_t + L_{ws} + L_{wt} \tag{13}$$

where λ_s and λ_t are hyperparameters that balance the contributions of the source and target diffusion losses, L_s and L_t, respectively.

Inference. During inference, only the Encoder and the classification head are retained, as shown in the purple dotted box in Figure 2. It is worth noting that DiffusionFake ensures the encoder network extracts generalized features only during training. Consequently, our DiffusionFake framework does not introduce any additional parameters during inference, thereby enhancing generalizability and reducing computational overhead.

4 Experiment

4.1 Experimental Setting

Dataset. To evaluate the generalization ability of DiffusionFake, we conduct experiments on several challenging datasets: (1) FaceForensics++ (FF++) [31]: This widely-used dataset contains 1,000 videos with four manipulation methods: DeepFakes, NeuralTextures, Face2Face, and FaceSwap. The pairwise real and forged data enable the generation of mixed forgery images. (2) Celeb-DF [23]: A high-quality DeepFake dataset containing various scenarios. (3) DeepFake Detection (DFD): This

Table 1: Frame-level cross-database evaluation from FF++(HQ) to Celeb-DF, Wild Deepfake, DFDC-P, DFD, and DiffSwap in terms of AUC and EER. * represents the results reproduced using open-source code or model.

Method	Celeb-DF		Wild Deepfake		DFDC-P		DFD		DiffSwap		Average	
	AUC	EER	AUC	EER	AUC	EER	AUC	EER	AUC	EER	AUC	EER
Xception [7]	65.27	38.77	66.17	40.14	69.80	35.41	87.86	21.04	74.25	32.04	72.67	33.48
Face X-ray [42]	74.20	-	-	-	70.00	-	85.60	-	-	-	-	-
F3-Net* [29]	71.21	34.03	67.71	40.17	72.88	33.38	86.10	26.17	76.89	30.83	74.96	32.92
MAT* [50]	70.65	35.83	70.15	36.53	67.34	38.31	87.58	21.73	79.93	27.77	75.13	32.03
GFF* [28]	75.31	32.48	66.51	41.52	71.58	34.77	85.51	25.64	78.38	28.15	75.46	32.51
LTW [40]	77.14	29.34	67.12	39.22	74.58	33.81	88.56	20.57	77.95	29.01	77.07	30.39
LRL [4]	78.26	29.67	68.76	37.50	76.53	32.41	89.24	20.32	-	-	-	-
DCL [41]	82.30	26.53	71.14	36.17	76.71	31.97	91.66	16.63	80.21	27.37	80.40	27.73
PCL+I2G [51]	81.80	-	-	-	-	-	-	-	-	-	-	-
SBI* [35]	80.76	26.97	68.22	38.11	76.53	30.22	88.13	17.25	75.20	31.49	77.77	28.81
UIA-ViT [53]	82.41	-	-	-	75.80	-	**94.68**	-	-	-	-	-
RECCE* [2]	70.50	35.34	67.93	39.82	75.88	32.41	89.91	19.95	77.59	29.38	76.36	31.38
UCF [48]	75.27	-	-	-	75.94	-	80.74	-	-	-	-	-
CADDM* [9]	77.56	30.63	72.56	33.63	72.45	33.56	82.90	25.20	75.58	31.01	76.21	30.81
EN-b4* [42]	73.51	34.17	70.04	37.03	70.51	33.98	87.57	21.31	77.38	29.44	75.80	31.19
VIT-B* [42]	74.64	33.07	75.46	31.53	74.24	34.29	84.38	24.15	78.50	28.14	77.44	30.24
En-b4+Ours	**83.17**	**24.59**	75.17	33.25	77.35	30.17	91.71	**16.27**	82.02	25.55	81.88	25.97
VIT-B+Ours	80.46	27.51	**80.14**	**29.62**	**80.95**	**27.66**	90.36	19.73	**86.98**	**21.32**	**83.78**	**25.17**

dataset comprises 363 real videos and 3,068 fake videos, primarily generated using the DeepFake method. (4) DFDC Preview (DFDC-P) [8]: A challenging dataset with 1,133 real videos and 4,080 fake videos, featuring various manipulation methods and backgrounds. (5) WildDeepfake [54]: A diverse dataset obtained from the internet, capturing a wide range of real-world scenarios. (6) DiffSwap [6]: A recently released dataset containing 30,000 high-quality face swaps generated using the diffusion-based DiffSwap method [52] on the MM-Celeb-A dataset. This dataset allows for evaluating cross-method generalization.

Training details. DiffusionFake is a plug-and-play architecture that can be integrated with different backbone networks by simply adjusting the dimensions of the alignment layer in the Feature Filter module. During training, we utilize a pre-trained Stable Diffusion 1.5 model with frozen parameters. Input images are resized to 224×224 pixels. We employ the Adam optimizer with a learning rate of 1e-5 and a batch size of 32. The model is trained for 20 epochs. The hyperparameters λ_s and λ_t are set to 0.7 and 1, respectively. We employ widely used data augmentations, such as HorizontalFlip, and CutOut. To ensure a fair comparison, we follow the data split strategy used in FaceForensics++ [31].

4.2 Experimental Results

We use AUC and EER to evaluate all the methods, including ours, both of them are widely used in deepfake detection.[3] We compare DiffusionFake with several state-of-the-art methods.

Cross-dataset evaluation. To validate the generalization capability of DiffusionFake, we first evaluate its performance on unseen datasets against recent state-of-the-art methods. Following previous settings, we train the models on the FF++ dataset and test them on several unseen domain datasets. The frame-level results are shown in Table 1, where * denotes results obtained using official code with consistent training settings and data.

We evaluate its performance using two representative backbones: EfficientNet-B4 (En-B4) and ViT-B. We observe that incorporating DiffusionFake significantly improves the generalization ability of both architectures compared to their original classification backends. For En-B4, our method boosts performance on Celeb-DF by 11% and achieves an average improvement of 6%. Similarly, ViT-B sees a 6% increase on DFDC and an average gain of 6% when trained with DiffusionFake. Remarkably, **these enhancements are achieved without increasing the parameter count or computational overhead during inference**. Compared to state-of-the-art methods, DiffusionFake outperforms

[3]More details about the evaluation metrics can be found in the appendix.

 https://doi.org/10.52202/079017-3218

Table 2: Abalation study of different components of DiffusionFake.

SD	Filter	Weight	Celeb-DF		DFDC-P	
			AUC	EER	AUC	EER
×	×	×	71.87	34.28	71.78	35.01
×	✓	✓	73.87	32.06	72.41	34.25
✓	×	×	77.35	29.05	75.69	32.12
✓	✓	×	80.79	26.37	76.17	31.57
✓	×	✓	78.67	28.33	76.59	31.22
✓	✓	✓	83.17	24.59	77.35	30.17

Table 3: Abalation study of backbones.

Backbone	Celeb-DF		WDF	
	AUC	EER	AUC	EER
ResNet	68.89	36.78	69.91	38.07
ResNet+Ours	75.27	32.44	73.25	34.27
En-b0	71.74	34.56	69.24	38.32
En-b0+Ours	76.31	31.56	74.40	33.99
Vit-S	70.59	35.87	70.60	37.59
Vit-S+Ours	74.58	32.95	75.10	33.87

disentanglement-based approaches like UCF and CAADM on Celeb-DF. Moreover, our method demonstrates substantial improvements on the latest diffusion-based face swapping dataset, DiffSwap, highlighting its effectiveness against the most recent forgery techniques. These results validate the ability of the guide module and Stable Diffusion network to encourage the encoder to learn more generalizable features by reconstructing source and target images. Due to space limitations, we provide the results of single-source and multi-source cross-manipulation evaluations in the appendix.

4.3 Ablation Study

Ablation of components.

We conducted an ablation study to investigate the impact of the key modules in DiffusionFake: 1) the pre-trained Stable Diffusion (SD) model, 2) the Feature Filter module, and 3) the Weight Module. The results are shown in Table 2, where without SD refers to not loading the pre-trained weights of the SD model, and without Filter means directly feeding the encoder's output features f into the guide module.

We can observe that the pre-trained Stable Diffusion model is crucial for the DiffusionFake framework. Without the pre-trained weights, the network struggles to reconstruct the source and target images due to information loss, hindering the training process. Furthermore Both the Feature Filter and Weight modules play significant roles, and removing either of them leads to a performance decline. Specifically, eliminating the Filter module results in a 5% AUC drop, as the filtering component allows the reconstruction to focus on relevant information without interference from redundant features. On the other hand, the absence of the Weight module causes a 3% performance decrease, as this module assesses the amount of source and target information contained in the image, providing a prior for the generative network to determine the importance of guided information during the reconstruction process.

Ablation of backbones. As our method can be flexibly embedded into different backbones by adjusting the alignment of the Feature Filter, we conduct an ablation study on various backbone architectures to demonstrate the versatility of DiffusionFake. We experiment with traditional ResNet-34, lightweight EfficientNet-B0, and the ViT-based ViT-Small. The results in Table 3 show that integrating our method into these backbones significantly improves generalization performance. For instance, applying DiffusionFake to the lightweight EfficientNet-B0 increases the generalization accuracy on Celeb-DF from 71.75% to 76.31%, surpassing the original EfficientNet-B4 (73.51%). This evidence suggests that our method can effectively drive different encoders to extract more generalizable features.

4.4 Analysis and Visualizations

Visualizations of reverse results. Figure 3 showcases the reconstruction results for both training and unseen samples using DiffusionFake. For training samples (Figure 3 A), DiffusionFake effectively reconstructs the target image, despite the source image being slightly blurry due to information loss, capturing the basic characteristics of the ground truth. In order to compare the reconstruction effect more intuitively, we use the RECCE method to directly reconstruct the source and target images of the fake image. It can be seen that the reconstruction effect is very poor. In contrast, the reconstruction effect of our method is better due to the help of the pre-trained SD model. For unseen samples (Figure 3 B), fake images with mixed features result in significant differences between reconstructed target

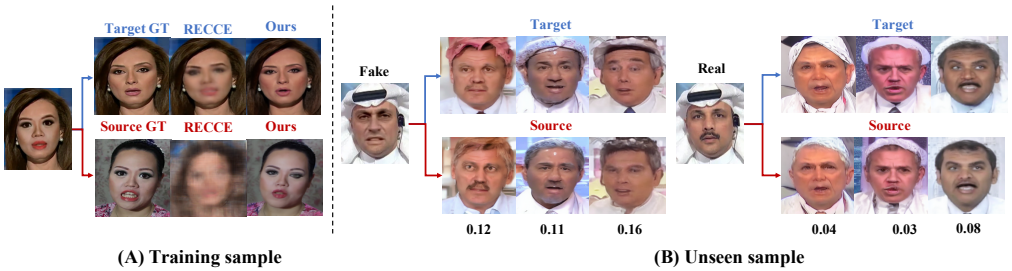

Figure 3: Reconstruction results of DiffusionFake for training (A) and unseen (B) samples. For unseen samples, the model is provided with three sets of initial Gaussian noise, differing only in the injected guide information. The numbers below represent the Euclidean distance between the corresponding source and target features.

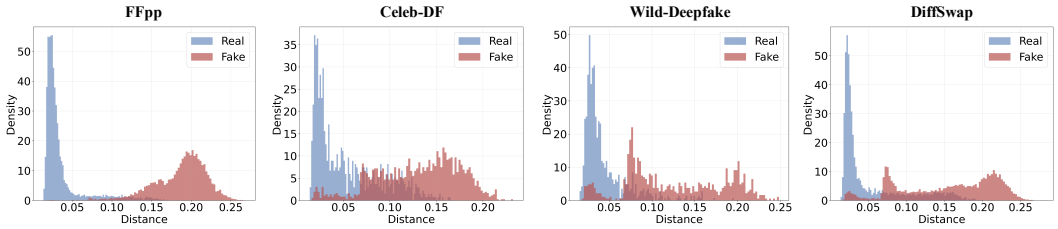

Figure 4: Histogram of feature divergence on FFpp, Celeb-DF, Wild-Deepfake, and DiffSwap.

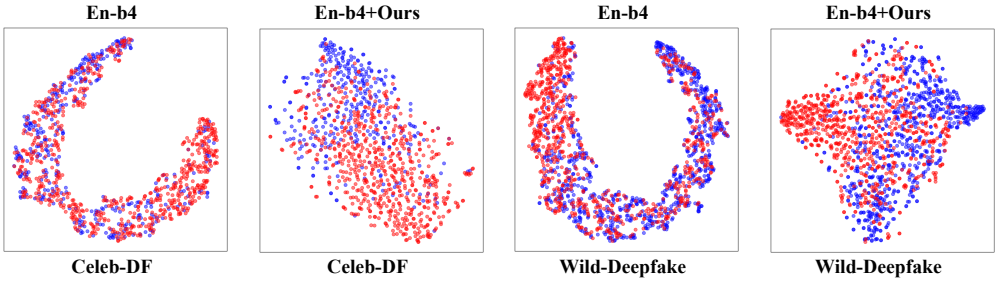

Figure 5: Feature distribution of En-b4 model and the En-b4 model trained with our DiffusionFace on two unseen datasets Celeb-DF and Wild-Deepfake via t-SNE. The red represents the real samples while the blue represents the fake ones.

and source images, while real images exhibit smaller differences. The Euclidean distances at the bottom quantify these differences, indicating larger differences in fake images compared to real ones.

Analysis of feature divergence. DiffusionFake utilizes two Feature Filter modules to separate source-related and target-related features, expecting significant divergence between f_s and f_t for forged images and minimal differences for genuine faces. To validate this, we visualize the Euclidean distance distribution between these features across various datasets, including FFpp, Celeb-DF, Wild-Deepfake, and DiffSwap, as shown in Figure 4. The plots clearly distinguish real from fake samples: real samples have small feature distances, mostly within 0.05, whereas fake samples show larger distances due to mixed source and target information. These observations strongly support that DiffusionFake effectively disentangles source and target information in the extracted features.

Analysis of feature distribution. To demonstrate that DiffusionFake enhances the discriminative power and generalization ability of the extracted features, we visualize the t-SNE plots of the last layer features from two encoders: the original EfficientNet-B4 (En-B4) and En-B4 trained with DiffusionFake. The feature distributions are examined on two unseen datasets, Celeb-DF and Wild-Deepfake. As illustrated in Figure 5, the original En-B4 exhibits poor generalization on both datasets, with the real and fake features being nearly inseparable. In contrast, when trained with DiffusionFake, the encoder learns to capture the generalizable hybrid features present in forged images via the reverse process. Consequently, the real and fake features become more distinctly separated, forming clear decision boundaries on both unseen datasets.

https://doi.org/10.52202/079017-3218

5 Conclusion

In this paper, we introduce DiffusionFake, a novel framework that leverages the generative process of face forgery to enhance the generalization capabilities of detection models. DiffusionFake inverts this generative process to extract and utilize hybrid features from source and target identities for effective forgery detection. Extensive experiments demonstrate that DiffusionFake significantly improves the generalization performance of various detector architectures without increasing inference parameters. The proposed framework enables the learning of discriminative and generalizable features, enhancing the robustness of detectors against a wide range of unseen forgeries.

6 Acknowledgements

This work was supported by National Science and Technology Major Project (No. 2022ZD0118202), the National Science Fund for Distinguished Young Scholars (No.62025603), the National Natural Science Foundation of China (No. U21B2037, No. U22B2051, No. U23A20383, No. 62072389, No. 62176222, No. 62176223, No. 62176226, No. 62072386, No. 62072387, No. 62002305 and No. 62272401), and the Natural Science Foundation of Fujian Province of China (No. 2021J06003, No.2022J06001).

References

[1] Weiming Bai, Yufan Liu, Zhipeng Zhang, Bing Li, and Weiming Hu. Aunet: Learning relations between action units for face forgery detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24709–24719, 2023.

[2] Junyi Cao, Chao Ma, Taiping Yao, Shen Chen, Shouhong Ding, and Xiaokang Yang. End-to-end reconstruction-classification learning for face forgery detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4113–4122, 2022.

[3] Liang Chen, Yong Zhang, Yibing Song, Lingqiao Liu, and Jue Wang. Self-supervised learning of adversarial example: Towards good generalizations for deepfake detection. In *CVPR*, pages 18710–18719, 2022.

[4] Shen Chen, Taiping Yao, Yang Chen, Shouhong Ding, Jilin Li, and Rongrong Ji. Local relation learning for face forgery detection. *AAAI*, 2021.

[5] Zhikai Chen, Lingxi Xie, Shanmin Pang, Yong He, and Bo Zhang. Magdr: Mask-guided detection and reconstruction for defending deepfakes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9014–9023, 2021.

[6] Zhongxi Chen, Ke Sun, Ziyin Zhou, Xianming Lin, Xiaoshuai Sun, Liujuan Cao, and Rongrong Ji. Diffusionface: Towards a comprehensive dataset for diffusion-based face forgery analysis. *arXiv preprint arXiv:2403.18471*, 2024.

[7] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *CVPR*, pages 1251–1258, 2017.

[8] Brian Dolhansky, Joanna Bitton, Ben Pflaum, Jikuo Lu, Russ Howes, Menglin Wang, and Cristian Canton Ferrer. The deepfake detection challenge dataset. *arXiv preprint arXiv:2006.07397*, 2020.

[9] Shichao Dong, Jin Wang, Renhe Ji, Jiajun Liang, Haoqiang Fan, and Zheng Ge. Implicit identity leakage: The stumbling block to improving deepfake detection generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3994–4004, 2023.

[10] Shichao Dong, Jin Wang, Jiajun Liang, Haoqiang Fan, and Renhe Ji. Explaining deepfake detection by analysing image matching. In *European Conference on Computer Vision*, pages 18–35. Springer, 2022.

[11] Xiaoyi Dong, Jianmin Bao, Dongdong Chen, Ting Zhang, Weiming Zhang, Nenghai Yu, Dong Chen, Fang Wen, and Baining Guo. Protecting celebrities from deepfake with identity consistency transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9468–9478, 2022.

[12] Zhihao Gu, Taiping Yao, Yang Chen, Shouhong Ding, and Lizhuang Ma. Hierarchical contrastive inconsistency learning for deepfake video detection. In *European Conference on Computer Vision*, pages 596–613. Springer, 2022.

[13] Fabrizio Guillaro, Davide Cozzolino, Avneesh Sud, Nicholas Dufour, and Luisa Verdoliva. Trufor: Leveraging all-round clues for trustworthy image forgery detection and localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20606–20615, 2023.

[14] Ying Guo, Cheng Zhen, and Pengfei Yan. Controllable guide-space for generalizable face forgery detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 20818–20827, 2023.

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

[16] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *CVPR*, pages 7132–7141, 2018.

[17] Baojin Huang, Zhongyuan Wang, Jifan Yang, Jiaxin Ai, Qin Zou, Qian Wang, and Dengpan Ye. Implicit identity driven deepfake face swapping detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4490–4499, 2023.

[18] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *International Conference on Learning Representations*, 2014.

[19] Nicolas Larue, Ngoc-Son Vu, Vitomir Struc, Peter Peer, and Vassilis Christophides. Seeable: Soft discrepancies and bounded contrastive learning for exposing deepfakes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 21011–21021, 2023.

[20] Binh M Le and Simon S Woo. Quality-agnostic deepfake detection with intra-model collaborative learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22378–22389, 2023.

[21] Jiaming Li, Hongtao Xie, Jiahong Li, Zhongyuan Wang, and Yongdong Zhang. Frequency-aware discriminative feature learning supervised by single-center loss for face forgery detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6458–6467, 2021.

[22] Lingzhi Li, Jianmin Bao, Ting Zhang, Hao Yang, Dong Chen, Fang Wen, and Baining Guo. Face x-ray for more general face forgery detection. In *CVPR*, pages 5001–5010, 2020.

[23] Yuezun Li, Xin Yang, Pu Sun, Honggang Qi, and Siwei Lyu. Celeb-df: A new dataset for deepfake forensics. *arXiv preprint arXiv:1909.12962*, 2019.

[24] Jiahao Liang, Huafeng Shi, and Weihong Deng. Exploring disentangled content information for face forgery detection. In *European Conference on Computer Vision*, pages 128–145. Springer, 2022.

[25] Honggu Liu, Xiaodan Li, Wenbo Zhou, Yuefeng Chen, Yuan He, Hui Xue, Weiming Zhang, and Nenghai Yu. Spatial-phase shallow learning: rethinking face forgery detection in frequency domain. In *CVPR*, pages 772–781, 2021.

[26] Anwei Luo, Chenqi Kong, Jiwu Huang, Yongjian Hu, Xiangui Kang, and Alex C Kot. Beyond the prior forgery knowledge: Mining critical clues for general face forgery detection. *IEEE Transactions on Information Forensics and Security*, 19:1168–1182, 2023.

[27] Anwei Luo, Enlei Li, Yongliang Liu, Xiangui Kang, and Z Jane Wang. A capsule network based approach for detection of audio spoofing attacks. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6359–6363. IEEE, 2021.

[28] Yuchen Luo, Yong Zhang, Junchi Yan, and Wei Liu. Generalizing face forgery detection with high-frequency features. In *CVPR*, pages 16317–16326, 2021.

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

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

[31] Andreas Rossler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. Faceforensics++: Learning to detect manipulated facial images. In *ICCV*, pages 1–11, 2019.

[32] Rui Shao, Tianxing Wu, and Ziwei Liu. Detecting and recovering sequential deepfake manipulation. In *European Conference on Computer Vision*, pages 712–728. Springer, 2022.

[33] Rui Shao, Tianxing Wu, and Ziwei Liu. Detecting and grounding multi-modal media manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6904–6913, 2023.

 https://doi.org/10.52202/079017-3218

[34] Liang Shi, Jie Zhang, and Shiguang Shan. Real face foundation representation learning for generalized deepfake detection. *arXiv preprint arXiv:2303.08439*, 2023.

[35] Kaede Shiohara and Toshihiko Yamasaki. Detecting deepfakes with self-blended images. In *CVPR*, pages 18720–18729, 2022.

[36] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2020.

[37] Luchuan Song, Zheng Fang, Xiaodan Li, Xiaoyi Dong, Zhenchao Jin, Yuefeng Chen, and Siwei Lyu. Adaptive face forgery detection in cross domain. In *European Conference on Computer Vision*, pages 467–484. Springer, 2022.

[38] Ke Sun, Shen Chen, Taiping Yao, Xiaoshuai Sun, Shouhong Ding, and Rongrong Ji. Towards general visual-linguistic face forgery detection. *arXiv preprint arXiv:2307.16545*, 2023.

[39] Ke Sun, Hong Liu, Taiping Yao, Xiaoshuai Sun, Shen Chen, Shouhong Ding, and Rongrong Ji. An information theoretic approach for attention-driven face forgery detection. In *European Conference on Computer Vision*, pages 111–127. Springer, 2022.

[40] Ke Sun, Hong Liu, Qixiang Ye, Jianzhuang Liu, Yue Gao, Ling Shao, and Rongrong Ji. Domain general face forgery detection by learning to weight. In *AAAI*, volume 35, pages 2638–2646, 2021.

[41] Ke Sun, Taiping Yao, Shen Chen, Shouhong Ding, Rongrong Ji, et al. Dual contrastive learning for general face forgery detection. In *AAAI*, 2022.

[42] Mingxing Tan and Quoc V Le. Efficientnet: Rethinking model scaling for convolutional neural networks. *ICML*, 2019.

[43] Ruben Tolosana, Ruben Vera-Rodriguez, Julian Fierrez, Aythami Morales, and Javier Ortega-Garcia. Deepfakes and beyond: A survey of face manipulation and fake detection. *arXiv preprint arXiv:2001.00179*, 2020.

[44] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[45] Yuan Wang, Kun Yu, Chen Chen, Xiyuan Hu, and Silong Peng. Dynamic graph learning with content-guided spatial-frequency relation reasoning for deepfake detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7278–7287, 2023.

[46] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *ECCV*, pages 3–19, 2018.

[47] Yuting Xu, Jian Liang, Gengyun Jia, Ziming Yang, Yanhao Zhang, and Ran He. Tall: Thumbnail layout for deepfake video detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22658–22668, 2023.

[48] Zhiyuan Yan, Yong Zhang, Yanbo Fan, and Baoyuan Wu. Ucf: Uncovering common features for generalizable deepfake detection. *arXiv preprint arXiv:2304.13949*, 2023.

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

[50] Hanqing Zhao, Wenbo Zhou, Dongdong Chen, Tianyi Wei, Weiming Zhang, and Nenghai Yu. Multi-attentional deepfake detection. *CVPR*, 2021.

[51] Tianchen Zhao, Xiang Xu, Mingze Xu, Hui Ding, Yuanjun Xiong, and Wei Xia. Learning self-consistency for deepfake detection. In *CVPR*, pages 15023–15033, 2021.

[52] Wenliang Zhao, Yongming Rao, Weikang Shi, Zuyan Liu, Jie Zhou, and Jiwen Lu. Diffswap: High-fidelity and controllable face swapping via 3d-aware masked diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8568–8577, 2023.

[53] Wanyi Zhuang, Qi Chu, Zhentao Tan, Qiankun Liu, Haojie Yuan, Changtao Miao, Zixiang Luo, and Nenghai Yu. Uia-vit: Unsupervised inconsistency-aware method based on vision transformer for face forgery detection. *ECCV*, 2022.

[54] Bojia Zi, Minghao Chang, Jingjing Chen, Xingjun Ma, and Yu-Gang Jiang. Wilddeepfake: A challenging real-world dataset for deepfake detection. In *ACM MM*, pages 2382–2390, 2020.

A Appendix

A.1 Cross-manipulation evaluation.

Cross-manipulation evaluation. To further validate the generalization ability across different manipulation methods, we conduct a cross-manipulation evaluation. We train models on a single manipulation method within the high-quality FF++ dataset and test them on all four methods. Using EfficientNet-B4 (En-B4) as the backbone, we compare our approach with the MAT method, which employs attention mechanisms to enhance the generalization ability of En-B4. Table 4 shows that DiffusionFake improves generalization performance across all manipulation methods. Notably, when trained on the FaceSwap method and tested on the Deepfake method, our approach outperforms the original En-B4 by 6%. Moreover, compared to the MAT method, DiffusionFake achieves a 4% improvement in generalization when trained on NeuralTextures and tested on FaceSwap. These results demonstrate the effectiveness of DiffusionFake in learning generalizable features that can be applied to unseen manipulation methods. **multi-source manipulation evaluation.** We also evaluate the multi-source generalization performance by training on three forgery methods and testing on the unknown method. Additionally, we assess the performance under low-quality (LQ) training conditions. As reported in Table 5, DiffusionFake achieves state-of-the-art results across all protocols and quality levels. Specifically, integrating our method with En-B4 improves generalization by approximately 8% compared to the backbone alone. Even under low-quality training conditions, DiffusionFake maintains a 7% performance gain, demonstrating the robustness and generalization capability of our proposed framework.

Table 4: Cross-manipulation evaluation in terms of AUC. Diagonal results indicate the intra-domain performance.

Train	Method	DF	F2F	FS	NT
DF	EN-b4	***99.97***	76.32	46.24	72.72
	MAT	*99.91*	78.23	40.61	71.08
	Ours	*99.82*	**78.46**	**52.29**	**74.43**
F2F	EN-b4	84.52	*99.20*	58.14	63.71
	MAT	86.15	*99.13*	60.14	64.59
	Ours	**88.92**	***99.36***	**63.19**	**68.55**
FS	EN-b4	69.25	67.69	***99.89***	48.61
	MAT	64.13	66.39	*99.67*	50.10
	Ours	**75.28**	**70.91**	*99.12*	**52.17**
NT	EN-b4	85.99	48.86	73.05	*98.25*
	MAT	87.23	48.22	75.33	*98.66*
	Ours	**89.54**	**51.71**	**79.15**	***98.71***

Table 5: Multi-source manipulation evaluation in terms of ACC, which follows [40]. H means high-quality image (c23) in FFpp, while L represents low-quality (c40).

Method	DF (H)	DF (L)	F2F(H)	F2F(L)
Xception	78.25	68.12	61.53	59.58
EN-B4	82.40	67.60	63.32	61.41
VIT-B	81.15	73.38	62.19	61.93
Multi-task	70.30	66.76	58.74	56.50
MLDG	84.21	67.15	63.46	58.12
LTW	85.60	69.15	65.60	65.70
DCL	87.70	75.90	68.40	67.85
RECCE	86.69	75.89	62.71	68.02
MAT	84.40	73.71	66.28	66.39
UCF	86.70	74.59	67.87	67.33
En-b4+Ours	**88.17**	75.13	**70.17**	**71.25**
VIT-b+Ours	87.23	**77.33**	68.93	68.75

A.2 Reconstruction metrics and performance

We calculated PSNR and SSIM for each model in the ablation study from Table 2. For each model, we used 10 random noise sets and their corresponding target GT, then averaged the values. The results are shown in Table 6. Our analysis reveals a positive correlation between reconstruction quality and detection performance. Models with better reconstruction quality generally demonstrated higher detection accuracy. Notably, when the SD pre-trained model is not used, the generation quality is very poor, corresponding to significantly worse results. This finding supports the intuition that better reconstruction ability contributes to more effective feature extraction, which in turn leads to improved detection performance.

A.3 Influence of the loss weight

We conducted comprehensive ablation studies to determine the optimal values for λ_s and λ_t. The results are shown in Table 7 and Table 8.

https://doi.org/10.52202/079017-3218

Table 6: Abalation study of different components of DiffusionFake with reconstruction metrics.

SD	Filter	Weight	Celeb-DF		DFDC-P		Metrics	
			AUC	EER	AUC	EER	SSIM	PSNR
×	×	×	71.87	34.28	71.78	35.01	0.11	10.91
×	✓	✓	73.87	32.06	72.41	34.25	0.15	11.35
✓	×	×	77.35	29.05	75.69	32.12	0.62	17.83
✓	✓	×	80.79	26.37	76.17	31.57	0.64	18.53
✓	×	✓	78.67	28.33	76.59	31.22	0.63	18.22
✓	✓	✓	**83.17**	**24.59**	**77.35**	**30.17**	**0.67**	**19.95**

Table 7: Abalation study of λ_s

λ_s	AVG-AUC	AVG-EER
0.1	77.13	29.01
0.3	78.99	27.51
0.5	80.27	26.36
0.7	**81.88**	**25.97**
1.0	79.31	26.77

Table 8: Abalation study of λ_t

λ_t	AVG-AUC	AVG-EER
0.3	77.30	28.15
0.5	79.25	27.77
0.7	81.09	26.15
1.0	**81.88**	**25.97**
1.2	80.38	26.98

Following the ControlNet[49] setup and considering that target reconstruction is relatively stable, we initially fixed λ_t at 1. 0 and varied λ_s through values of 0.1, 0.3, 0.5, 0.7, and 1.0. Our experiments showed that $\lambda_s = 0.7$ yielded the best average performance across five test datasets.

We then fixed λ_s at 0.7 and conduct ablation studies on λ_t, inding the peak performance at $\lambda_t = 1.0$. These results align with our intuition. The source image often differs significantly from the fake image, so a slightly smaller loss weight for the source λ_s helps maintain training stability. We observed that if λ_s is too large, the loss becomes difficult to minimize.

A.4 Visualizations of CAM result.

To further illustrate the ability of our method to accurately focus on relevant locations in generalized images, we visualize the Class Activation Mapping (CAM) results of both our approach and the vanilla encoder across different datasets. As shown in Figure 6, the conventional EfficientNet-B4 (En-B4) encoder often fails to highlight key areas, such as the blurred mouth region in Celeb-DF images. This limitation can reduce the effectiveness of forgery detection. In contrast, our method demonstrates a broader focus during training, targeting significantly larger regions that may include latent forgery areas. This comprehensive attention to detail contributes to enhancing the generalization performance of the detection model. By effectively identifying and concentrating on these critical regions, our method provides a more robust defense against sophisticated forgery techniques, ultimately leading to more accurate and reliable detection outcomes across diverse datasets.

A.5 Visualizations of weight module.

Figure 7 presents the source and target scores computed by our Weight Module for four different attack types. It is evident that the target scores are generally higher than the source scores, indicating that reconstructing the target information contributes more significantly to the overall reconstruction process, while the reconstruction of the source image relies more heavily on the pre-trained knowledge. Moreover, the scores vary across different attack types. For samples that are more similar to the target, such as NeuralTextures and Face2Face, the corresponding target scores are higher (greater than 0.95) due to the high proportion of target features they contain, while the source scores are lower due to the

Figure 6: CAM maps of the baseline model (EN-b4) and En-b4 trained with DiffusionFake method on three unseen datasets: Celeb-DF, WDF (Wild-Deepfake), and DiffSwap.

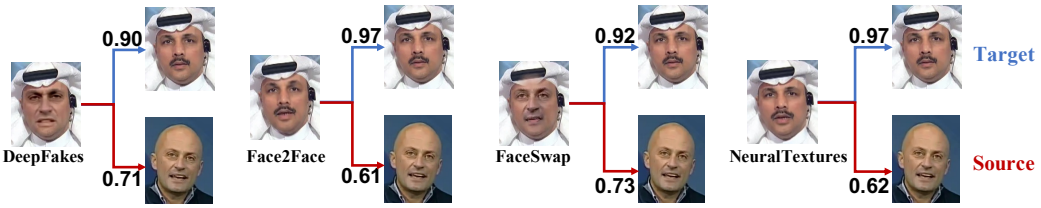

Figure 7: Visualization of weights for different attack types. The blue lines connect the target weights, while the red lines connect the source weights.

limited presence of source features. On the other hand, for Deepfakes and FaceSwap, which involve replacing the source's facial region onto the target, the proportion of source information is relatively higher, resulting in slightly elevated source scores compared to other attack types.

A.6 Visualizations of various target reconstructions

Our approach uniquely reconstructs source and target images from fake ones, facing challenges due to information loss. This can lead to expression inaccuracies or blur, as seen in Figure 3A of the main paper. However, DiffusionFake's primary goal is to compel the detection model to extract source-related and target-related features, enhancing generalization. Reconstruction quality serves as a means to this end, not the ultimate objective.

Moreover, we've observed that fine-grained expression control in reconstructed images is closely related to the input noise. With suitable input noise, we can achieve better reconstruction results. As shown in Figure 8, we visualize target images reconstructed under five different noise patterns, along with their PSNR and SSIM scores compared to the target GT. Notably, the last noise pattern produces images with expressions and details closely matching the target GT. This finding provides valuable insights into our method's capabilities and potential for improvement.

A.7 Evaluation Metric

We use two common metrics to evaluate the performance of our forgery detection method: the Area Under the Receiver Operating Characteristic Curve (AUC) and the Equal Error Rate (EER).

The AUC is a widely adopted metric that measures the overall performance of a binary classifier across all possible decision thresholds. It represents the probability that a randomly chosen positive instance (i.e., a forged image) will be ranked higher than a randomly chosen negative instance (i.e., a real image). The EER is another commonly used metric that represents the point on the ROC curve where the False Positive Rate (FPR) and the False Negative Rate (FNR) are equal.

In summary, we use AUC and EER as our primary evaluation metrics, where a higher AUC and a lower EER indicate better forgery detection performance. These metrics provide a comprehensive assessment of the classifier's ability to distinguish between forged and genuine images across various decision thresholds.

https://doi.org/10.52202/079017-3218

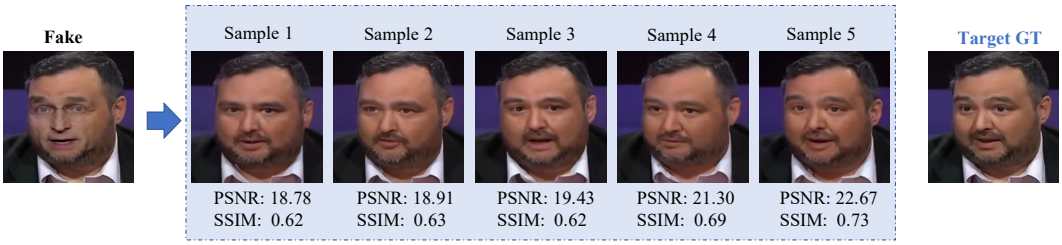

Figure 8: Visualization of our method to reconstruct the target image under different initialization noise conditions. The following numbers represent PSNR and SSIM respectively.

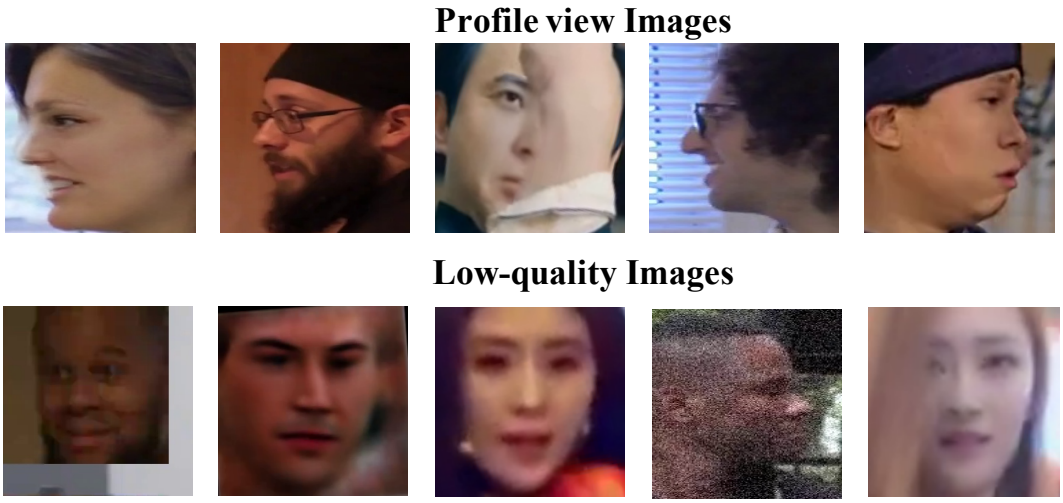

Figure 9: Visualization of two typical misprediction samples. Represents Profile view Images and Low-quality Images respectively.

A.8 Data Privacy

Our research aims to protect digital integrity by developing methods to detect DeepFake content. This work directly addresses ethical concerns surrounding AI-generated face-swapped images, contributing to the field of privacy protection in the digital age.

To validate our method, we utilized widely-accepted public datasets including FaceForensics++ [31], Celeb-DF-V2 [23], DeepFake Detection, DFDC Preview [8], WildDeepfake [54], and DiffSwap [6, 52]. These datasets are extensively used in the DeepFake detection domain, with cumulative citations numbering in the thousands. Moreover, these datasets were published in top-tier computer vision conferences, indicating that they have undergone rigorous ethical reviews as part of the conference submission process. These works have provided a solid foundation for deepfake detection research.

All datasets used in our study are bound by strict licensing terms that limit their use to non-commercial research and educational purposes. For instance, the DFDC-Preview dataset was created by META using with "paid actors who entered into an agreement to the use and manipulation of their likenesses in our creation of the dataset". FaceForensics++, DeepFake Detection, and DiffSwap stipulate: "Researcher shall use the Database only for non-commercial research and educational purposes." Similarly, Celeb-DF and WildDeepfake have comparable restrictions such as "Our dataset is used only for research purposes, we only release the face sequence rather than the whole video". These licensing terms serve to safeguard privacy and ensure ethical use of the data. These licenses are publicly available on the respective official GitHub repositories.

A.9 Limitations and Broader Impacts.

Limitation: The framework relies on paired source and target images for training, which may not always be feasible in real-world scenarios. We aim to integrate self-supervised methods to generate these images in the future. Additionally, the effectiveness of DiffusionFake against more sophisticated forgery techniques, such as those involving multiple source identities or partial manipulations, requires further investigation.

Additionally, upon examining our misprediction results, we identified two main categories of errors, as illustrated in Figure 9.

1). Profile view images: These images present a challenge during training, as they are difficult to reconstruct into source and target images due to significant information loss. This results in misclassification during inference. 2). Low-quality images: Our method encourages the detector to decouple source-related and target-related features to improve generalization. However, low-quality, blurry images hinder the network's ability to extract these features effectively, leading to misclassification.

In future work, we will focus on optimizing these two types of images, such as increasing the weight of low-quality data reconstruction during training, and using data augmentation to supplement the side faces in the training data.

Broader Impacts: Our method could potentially be used as an adversarial discriminator to create more difficult-to-detect images. Future research needs to address how to prevent this misuse.

https://doi.org/10.52202/079017-3218

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and precede the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS paper checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

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

   Answer: [Yes]

   Justification: Our abstract and introduction clearly outline the paper's motivation, the specific work conducted, and the results achieved.

   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 specifically discuss the limitations of this paper in Section A.9.

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: Our paper does not involve 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: We clearly present the implementation details of our method, including the models, datasets, and hyperparameter settings used in the experiment.

   Guidelines:

https://doi.org/10.52202/079017-3218

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

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

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

   Answer: [Yes]

   Justification: We use publicly available datasets. We are currently cleaning the code and will release it as soon as possible.

   Guidelines:

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

- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

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

   Answer: [Yes]

   Justification: We provide extensive details on the training procedures, including optimizer settings, hyperparameters of the proposed method, and the datasets used. These details encompass various experiments.

   Guidelines:

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

7. **Experiment Statistical Significance**

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

   Answer: [No]

   Justification: Repetitive computations consume excessive resources and time; we will address this in future updates.

   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: We provide the GPU resource requirements, models, and quantities for the experiments in Section 4.

https://doi.org/10.52202/079017-3218

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: The codes we provide follow 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 discuss the societal impact of our work in Appendix A.9.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

Answer: [NA]

Justification: Our model does not involve generative tasks, thus eliminating any risk of abuse from this perspective.

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 appropriately credit the referenced models, code, and datasets through methods such as adding suitable citations.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

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

Answer: [NA]

Justification: Our work does not involve new assets.

Guidelines:

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

https://doi.org/10.52202/079017-3218

14. **Crowdsourcing and Research with Human Subjects**

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

    Answer: [NA]

    Justification: Our work does not involve crowdsourcing experiments and research with human subjects.

    Guidelines:

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

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

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

    Answer: [NA]

    Justification: Our research does not involve human subjects and the potential risks to participants.

    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.