GAP-Diff: Protecting JPEG-Compressed Images from Diffusion-based Facial Customization

Haotian Zhu*, Shuchao Pang*[1][2], Zhigang Lu†[1], Yongbin Zhou* and Minhui Xue‡
*Nanjing University of Science and Technology, China
Email: {haotian.zhu, pangshuchao, zhouyongbin}@njust.edu.cn
†Western Sydney University, Australia
Email: z.lu@westernsydney.edu.au
‡CSIRO's Data61, Australia
Email: jason.xue@data61.csiro.au

***Abstract*—Text-to-image diffusion model's fine-tuning technology allows people to easily generate a large number of customized photos using limited identity images. Although this technology is easy to use, its misuse could lead to violations of personal portraits and privacy, with false information and harmful content potentially causing further harm to individuals. Several methods have been proposed to protect faces from customization via adding protective noise to user images by disrupting the fine-tuned models.**

Unfortunately, simple pre-processing techniques like JPEG compression, a normal pre-processing operation performed by modern social networks, can easily erase the protective effects of existing methods. To counter JPEG compression and other potential pre-processing, we propose GAP-Diff, a framework of Generating data with Adversarial Perturbations for text-to-image Diffusion models using unsupervised learning-based optimization, including three functional modules. Specifically, our framework learns robust representations against JPEG compression by backpropagating gradient information through a pre-processing simulation module while learning adversarial characteristics for disrupting fine-tuned text-to-image diffusion models. Furthermore, we achieve an adversarial mapping from clean images to protected images by designing adversarial losses against these fine-tuning methods and JPEG compression, with stronger protective noises within milliseconds. Facial benchmark experiments, compared to state-of-the-art protective methods, demonstrate that GAP-Diff significantly enhances the resistance of protective noise to JPEG compression, thereby better safeguarding user privacy and copyrights in the digital world.

I. Introduction

When posting/sending photos within your social networks, have you ever thought that someone might customize and modify your photos, as shown in Figure 1, without your permission? Many image customization tools (e.g., GAN-based ones [20], [21] and diffusion-based ones, named fine-tuned text-to-image diffusion models [10], [15], [34], [35]) can easily generate lifelike photos using your posted/sent ones. Such tools are bringing serious and pervasive social problems, reported by major media outlets like CNN and BBC [8], [9], [42], as increasingly being used to create fake news about different individuals. Among these image customization tools, the fine-tuned text-to-image diffusion models (FT-T2I-DMs), implemented by fine-tuning T2I-DMs using techniques like DreamBooth [34] and its successors - DreamBooth-based LoRA [35] (which integrates LoRA [17] into DreamBooth) and SVDiff [15], generate the most realistic images thanks to the powerful posterior knowledge learned by diffusion models in image generation [1], [5], [45], [50].

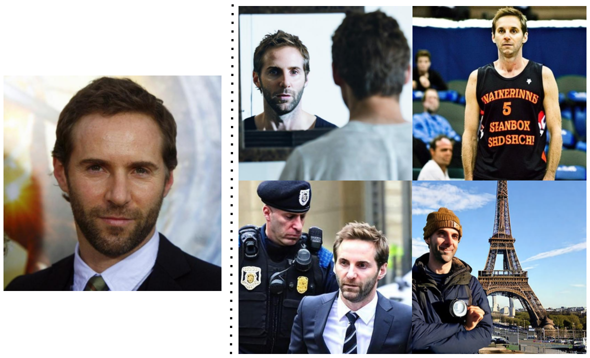

Fig. 1: Taken a random identity from VGGFace2 [3] facial dataset (left), the FT-T2I-DM (using DreamBooth [34]) produces four fake images (right) based on different prompts.

As a researcher, you can surely find out that existing works [24], [26], [36], [38], [45], [47], [49], [53] might protect your photos against the FT-T2I-DMs-based malicious image customization. Unfortunately, according to our observation (shown in Figure 2), these protective means will never work in your case, simply due to the JPEG compression applied on your uploaded (and also protected) photos, which is a normal pre-processing action performed by modern social networks, such as Facebook, Instagram, Whatsapp, X, WeChat, etc. [29], [43]. In Figure 2, we show two sets of customized images using DreamBooth-based FT-T2I-DMs on the protected images with and without JPEG compression, where JPEG compression

[1]: equal contribution
[2]: corresponding author

Network and Distributed System Security (NDSS) Symposium 2025
24-28 February 2025, San Diego, CA, USA
ISBN 979-8-9894372-8-3
https://dx.doi.org/10.14722/ndss.2025.242088
www.ndss-symposium.org

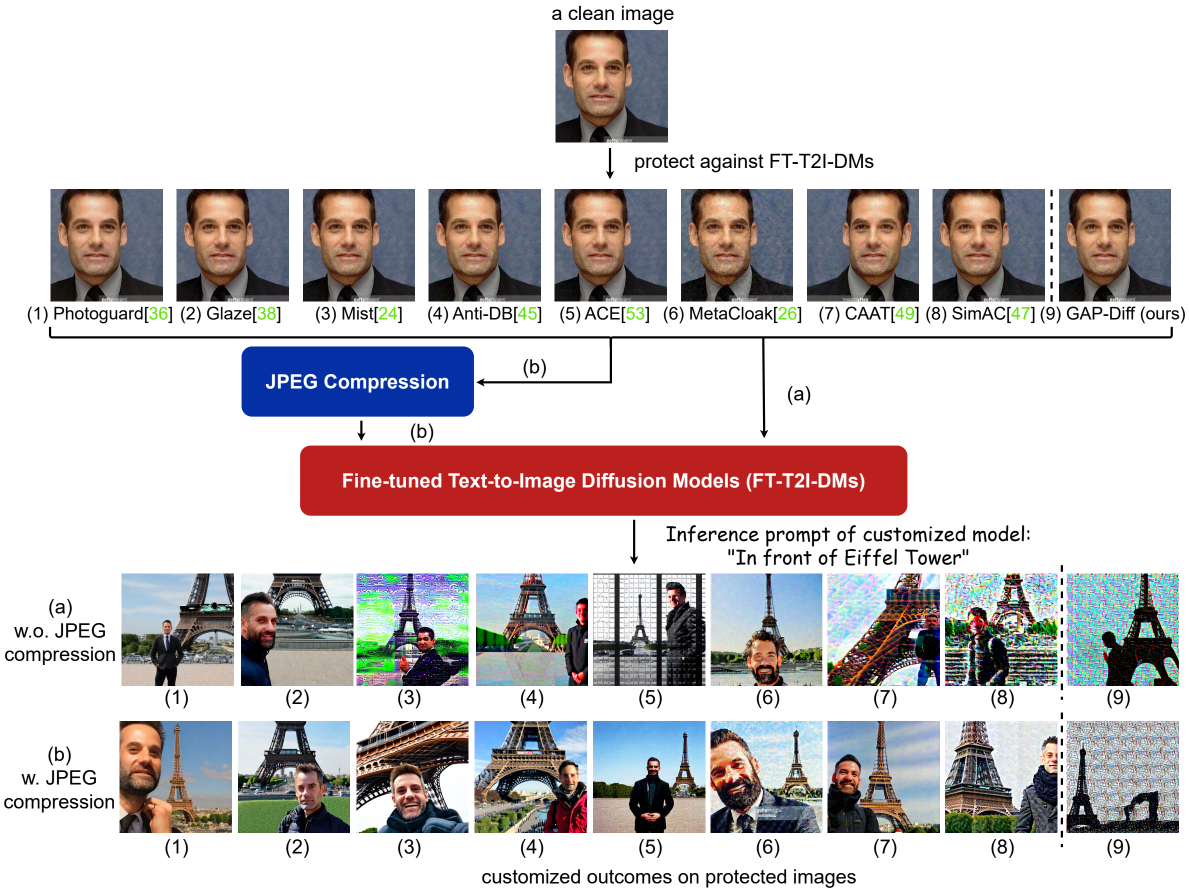

Fig. 2: Comparison between existing methods and GAP-Diff (ours) using DreamBooth-based FT-T2I-DMs on the protected images with and without JPEG compression.

did damage the protection effect of the existing works (images from number one to number eight) against the customization. The reason behind our observation is two-fold. First, JPEG compression is capable of reducing high-frequency information from images [46]. Second, to prevent the FT-T2I-DMs-based customization, the existing works inject noise concentrating on the high frequency information of images [47]. Hence, most of such high-frequency noise present in the protected images will be removed by JPEG compression, potentially compromising the effectiveness of existing protective measures.

To mitigate the degradation in protection against FT-T2I-DMs customization caused by JPEG compression, we propose a novel generative framework of unsupervised learning-based optimization, named GAP-Diff. In a nutshell, we achieve an adversarial mapping from clean images to protected images by designing adversarial losses against fine-tuning and JPEG compression. Specifically, different from the existing works that belong to iterative methods as depicted in Figure 3, we first construct a generator module as the mapping function through a robust neural network to obtain protected images in one step. Then, the generative framework uses the proposed adversarial loss functions that are invariably utilized in the fine-tuning methods of T2I-DMs as the primary optimization objectives from our fine-tuning T2I-DM module. Finally, thanks to the powerful learning and optimization capabilities of our generative framework, enabling JPEG compression to be computed during the backpropagation relying on a pre-processing simulation module, the protective noise injected by our solution is resistant to JPEG compression; hence keeps the protection effect against the FT-T2I-DMs in real social networks scenarios. This also explains why adaptive defense methods are difficult to apply in this scenario: most existing methods use the PGD strategy to generate protective noise, which involves an iterative process of creating adversarial samples and customizing DM for reference. These attacks are more complex than those applied to pre-trained classification models. Adding JPEG resistance while maintaining reported effectiveness requires careful redesign and significant modification of the noise generation process, which has been effectively addressed in this paper.

Our contributions can be summarized as follows:

- We propose a novel solution designed to protect images from customization by fine-tuned text-to-image diffusion models, which demonstrates significantly enhanced resistance to JPEG compression, a common pre-processing operation in real social networks scenarios, making it more suitable for digital world compared to existing solutions.

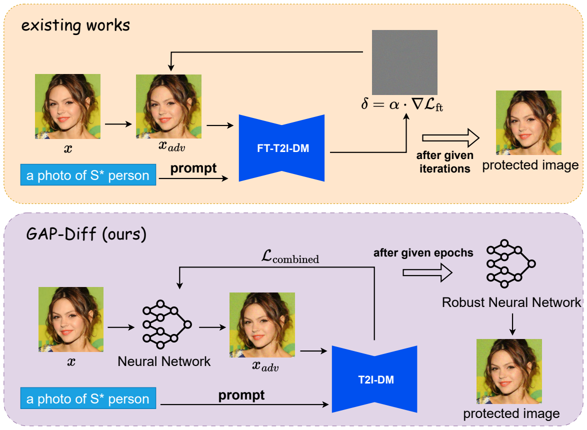

Fig. 3: The difference between the solution of existing works and ours. Given a clean image x, the former generally uses a fine-tuning loss from the FT-T2I-DM to calculate the gradients and update x_{adv} iteratively. After given iterations, they can get the final protected image. By contrast, the latter directly outputs the x_{adv} via a neural network and optimizes the network through unsupervised learning using a combined loss. Here, in addition to fine-tuning loss, more control and computational items, such as adversarial loss for countering JPEG compression, can be added than the former ones using the iterative way. After given epochs, we can get a well-trained and robust neural network that can generate protected images.

- To the best of our knowledge, we are the first to utilize neural networks to learn adversarial losses against diffusion models. This way, we can achieve potentially stronger protective noises by searching global optima of the optimization problem. Additionally, by shifting the time-consuming iterative process of generating protective noise for each image from existing works to the training phase of the neural network, our generator, once trained, can rapidly produce protected images within milliseconds.
- We conduct extensive experiment evaluation on commonly used datasets under various configurations. The experimental results confirmed the advantages of our solution in time and space efficiency and resistance to JPEG compression, compared to the state-of-the-art methods. Furthermore, experiments on different noise budgets, prompt and T2I-DM weight mismatch, fine-tuning methods and pre-processing techniques are also conducted in detail.

II. Related Work

A. Generative Models and T2I Diffusion Models

In previous works [12], [22], variational autoencoders (VAEs) and generative adversarial networks (GANs) have been widely used for generative tasks. They can roughly be categorized into likelihood-based generative models that directly fit the data distribution and implicit generative models, which aim to map the output images to the target distribution by ensuring that they are classified as real by a discriminator. However, these previous methods are often limited by network architecture and suffer from issues such as sample quality and training instability [13].

Recently, the diffusion model has emerged [16], [40], [41], which generates samples by simulating the diffusion process. In the forward process, the original sample is gradually diffused into standard Gaussian noise through noise injection. Then, in the reverse process, the noise learned by the U-Net from the forward process is used to gradually denoise the image, mapping the data back to the original distribution to generate new samples similar to the original image.

Benefiting from large-scale datasets like LAION5B [37], the diffusion model has been used as a text-to-image model known as T2I-DM for various generative tasks. In this type of task, text is typically encoded by an encoder such as Clip [31] to generate a condition c for the diffusion model, which is then incorporated into the training input of the U-Net. In the open-source and widely used LDM [32], VAEs are used as image encoders to encode images into smaller latent variables, which are then added to the diffusion process, reducing training and inference costs. Additionally, LDM incorporates attention mechanisms into the residual layers of the U-Net, enabling better mapping between the conditional c and the input latent variables in the neural network, allowing users more creative freedom.

B. Fine-tuning and Customization

To reduce training costs and enable users to better generate specific characters or artistic styles, fine-tuning methods on T2I-DMs have been proposed and widely adopted for customization. The fine-tuning methods are based on pre-trained conditional diffusion model weights and involve personalized training by outputting several images of specific characters or styles along with specific concept terms. Typically, their training process does not require much time.

Among these approaches, DreamBooth [34] is popular due to its excellent generation quality and straightforward fine-tuning method. Specifically, DreamBooth conducts training by providing $3 \sim 5$ images of characters needing customization along with a special term denoting the target user such as "sks", which is a special token chosen by performing a rare-token lookup so that it could minimize the probability of the identifier having a strong prior when fine-tuning [34]. This method encourages the T2I-DM to remember relevant concepts and achieve image mappings corresponding to those concepts during inference, thereby achieving customization.

Regarding other fine-tuning methods, for example, Text-Inversion [10] adjusts the text encoding set to describe concepts, while Custom Diffusion [23] optimizes only the parameters in the model's cross-attention layers. By integrating fine-tuning methods with LoRA [17], the cost of fine-tuning can be reduced by decomposing attention layers into low-rank matrices. Consequently, DreamBooth-based LoRA was proposed [35]. SVDiff [15] involves fine-tuning the singular values of the weight matrices, thereby reducing the risk of and language drift.

C. Privacy Protection on GANs and DMs

With the continuous advancement of artificial intelligence technology and the deepening research into generative networks, the issues of identity forgery and protection have become hot topics in related fields, with DeepFake being widely recognized as one prominent example. There are numerous detection techniques for DeepFake [2], [14], [18], [48], [54], which aim to discern forged images by learning the distinct features between forged and genuine facial images. Although these methods can detect forged images, they operate after the forgery has occurred, making it challenging to protect individuals' privacy.

Before DeepFake and customization happen, a called "image cloaking" [39] privacy protection technique is proposed to prevent the generation of forged images. Methods like [51], [52] disrupt the learning and generation capabilities of GAN-based DeepFake methods, thereby concealing images from the GAN model.

For popular T2I-DM-based DeepFake methods, many new privacy protection techniques based on adversarial attacks have recently emerged. PhotoGuard [36] proposes attacking the VAE or U-Net parts of text-to-image models by perturbing the latent encoding to mislead the model. Glaze [38] misguides diffusion models by making the feature distance of the training data closer to the target image. AdvDM [25] and its subsequent version Mist [24] achieve protection by performing adversarial attacks on pre-trained diffusion models. ACE [53] induces the fine-tuned LDM to learn the same pattern as a bias in predicting the score function and improves the attack effects. Anti-DreamBooth [45] focuses on face protection during fine-tuning by iteratively applying the classic PGD [27] method to the diffusion model to obtain protective noise. Several methods have been further proposed to optimize Anti-DreamBooth. Specifically, CAAT [49] enhances protection by attacking only the U-Net's cross-attention layers. MetaCloak [26] addresses the lack of pre-processing resistance in Anti-DreamBooth by using multiple surrogate diffusion models to find the optimal perturbation against pre-processing, although this reduces protection effectiveness in non-preprocessed scenarios and incurs a high computational cost for generating protective noise. SimAC [47] improves Anti-DreamBooth through a greedy algorithm, identifying the best perturbation timestep and feature layer.

However, the protection effects of these existing methods can be easily removed by the high-frequency information quantization of JPEG compression. Therefore, our GAP-Diff framework is proposed to address this challenge.

III. Preliminaries and Threat Model

A. Preliminaries

Diffusion model. As introduced in Section II-A, DM primarily contains two processes. In the forward process, an image $x_0 {\sim} q(x)$ is perturbed with a noise scheduler $\{\beta_t : \beta_t \in (0,1)\}_{t=1}^{\mathrm{T}}$ that is designed based on a sequence of increasing levels of noise through T steps. In this process, we can obtain a sequence of x, $\{x_0, x_1, ..., x_T\}$, where each x can be obtained through the following formula that depends on random noise and timestep t:

$$x_t = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, \tag{1}$$

where $\alpha_t = 1 - \beta_t$, $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$ and $\epsilon \sim \mathcal{N}(0, \mathbf{I})$.

During the reverse process, it denoises x_t into x_{t-1} by training a U-Net network $\epsilon_\theta(x,t)$ or $\epsilon_\theta(x,c,t)$, depending on whether it is a conditional denoising diffusion model. Ultimately, it gradually denoises from a standard Gaussian distribution to obtain an image from the original distribution. Training the U-Net network effectively involves making the removed noise as similar as possible to the noise added during the forward process, aiming to approximate the reconstructed distribution to the original distribution in the forward process. The training formulas are as follows:

$$\mathcal{L}_{\text{uncond}}(\theta, x_0) = \mathbb{E}_{x_0,t,\epsilon\sim\mathcal{N}(0,\mathbf{I})}||\epsilon - \epsilon_\theta(x_{t+1},t)||_2^2, \tag{2}$$

$$\mathcal{L}_{\text{cond}}(\theta, x_0) = \mathbb{E}_{x_0,t,c,\epsilon\sim\mathcal{N}(0,\mathbf{I})}||\epsilon - \epsilon_\theta(x_{t+1},t,c)||_2^2, \tag{3}$$

where c is condition input.

Adversarial attacks. The objective of adversarial attacks is to deceive the behavior of a model by adding small perturbations to the input images. Conventional adversarial attack methods typically target a classifier f. They start by obtaining the output y_{true} of the input x from f, then alter the pixels of x until $f(x) \neq y_{true}$. The visual imperceptibility of the perturbation is ensured by the noise budget η, and the formula for obtaining the perturbation δ is as follows:

$$\delta_{\text{adv}} = \underset{||\delta||_p<\eta}{argmax} L(f(x+\delta), y_{true}). \tag{4}$$

Projected Gradient Descent (PGD) [27] is a widely utilized iterative attack method that aims to modify the pixels of input x to induce an ascent in the loss function gradient of network f, which is used in previous attack methods [24], [26], [36], [45], [47], [49], [53] and can be described by the following formula:

$$x^{k+1} = \prod_{(x,\eta)} (x^k + \alpha sgn(\nabla_x L(f(x+\delta, y_{true}))), \tag{5}$$

where $x^0 = x$, α represents the step size for each gradient ascent iteration, and $sgn(\cdot)$ is a sign function.

Different from iterative methods, by solving an optimization problem initially proposed by C&W [4] to obtain the perturbation satisfying Eq. 4, [30] seeks θ disrupting classification network such that the following formula holds for most $x \in \mathcal{N}$, where $\mathcal{N}$ represents the set of natural images:

$$\mathcal{K}(f_\theta(x)) \neq \mathcal{K}(x), \tag{6}$$

where $\mathcal{K}$ represents the target classification network, while f denotes the network being optimized.

JPEG compression resistance. As a common lossy technique, JPEG compression aims to preserve more noticeable low-frequency components while eliminating high-frequency components that are less perceptible to the human eye. Some

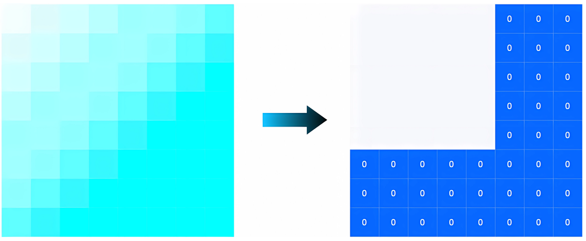

Fig. 4: Explanation of JPEG-Mask. In the left, the quantization of JPEG compression is more intense in high-frequency components, meaning that values in dark areas are quantized close to zero. In contrast, JPEG-Mask, as depicted in the right image, simulates this process by retaining some low-frequency regions while directly setting the positions of high-frequency regions to zero.

advancements have focused on simulating JPEG compression during the training phase of neural networks to enhance their resistance. One notable method is the JPEG-Mask approach by [55], which involves zeroing out a set of fixed high-frequency coefficients, retaining only the 5×5 low-frequency region of the Y channel and the 3×3 low-frequency region of the U and V channels, as illustrated in Figure 4. This simulation technique can be utilized to enable the network to exhibit a certain level of robustness against JPEG compression during the training process.

B. Threat Model

As described in Section I, FT-T2I-DMs can utilize a few facial images to generate images featuring specific individuals in various scenes. The adversary may gather a set of images $\mathcal{X}^n$ depicting a particular identity from nature and input all instances of $x^n \in \mathcal{X}^n$ into the T2I-DM for fine-tuning. The adversary employs the conditional diffusion model of DM to train its denoiser U-Net, denoted as ϵ_θ, following the fine-tuning algorithm to obtain optimized model parameters θ^*. Specifically, the fine-tuning algorithm compels the DM to learn to reconstruct images from $\mathcal{X}^n$ and utilize a generic prompt c, such as "a photo of S* person", where "S*" serves as a specific prompt word to bind with identity $\mathcal{X}^n$. To train for effective binding, the adversary utilizes the loss of the conditional diffusion model as described in Eq. 3 with the generic prompt c. On the other hand, fine-tuning models also often introduce a loss term preserving prior knowledge about the person subject, aiming to prevent overfitting and language drift issues solely from training on specific identity images, using a prior prompt c_{pr}. Overall, these two components comprise the optimization objective in Eq. 7 employed by the adversary using the family of DreamBooth-based methods [15], [34], [35] which are demonstrated most powerfully by fine-tuning through text-encoder and U-Net of the T2I-DM.

$$\begin{aligned}\mathcal{L}_{\text{ft}}(\theta, x_0^n) = \mathbb{E}_{x_0^n,t,t'}&||\epsilon - \epsilon_\theta(x_{t+1}^n, t, c)||_2^2 \\ &+ \lambda||\epsilon' - \epsilon_\theta(x'_{t'+1}, t', c_{pr})||_2^2,\end{aligned} \tag{7}$$

where x_t^n is noisy variable of $x^n \in \mathcal{X}^n$, and $x'_{t'+1}$ is noisy variable of class example $x' \in \mathcal{X}^{ori}$, where $\mathcal{X}^{ori}$ represents the set of images generated from original LDM θ_{ori} with prior prompt c_{pr}. ϵ and ϵ' are sampled from standard Gaussian noise $\mathcal{N}(0, \mathbf{I})$. λ represents the weight of the regularization term.

Furthermore, due to common compression methods employed by social media platforms or to circumvent recent noise-based protective measures aimed at preventing customization of individual photos, the adversary may obtain a collection of preprocessed images $\mathcal{X}^{pre}$ by JPEG compression, which represents $\mathcal{X}^n$ undergoing pre-processing function $p(\cdot)$. They would then utilize $x^{pre} \in \mathcal{X}^{pre}$ for fine-tuning following Eq. 8.

$$\begin{aligned}\mathcal{L}_{\text{ft'}}(\theta, x_0^{pre}) = \mathbb{E}_{x_0^{pre},t,t'}&||\epsilon - \epsilon_\theta(x_{t+1}^{pre}, t, c)||_2^2 \\ &+ \lambda||\epsilon' - \epsilon_\theta(x'_{t'+1}, t', c_{pr})||_2^2.\end{aligned} \tag{8}$$

To protect user photos, we cannot directly attack the whole customization part of the model, since it is entirely controlled by the adversary, making the fine-tuning function become our condition for the protection scenario rather than the target. Additionally, we also do not solely attack the image encoder, because even if we have a way to disrupt it, the disruption is likely to be probabilistically eliminated by the prior knowledge of the diffusion model, and we cannot guarantee that the adversary's encoder is the same as ours. Therefore, we decide to start from the generation part and disrupt the predictive performance of the U-Net model through the conditional loss. The goal of GAP-Diff can be succinctly summarized as obtaining a mapping $f(\cdot)$ from clean images to protected images that satisfy the following criteria, with the intensity of the protective noise constrained by η.

$$\begin{aligned} & f^* \in \arg\max_{f} \mathcal{L}_{\text{cond}}(\theta^*, p(f(\mathcal{X}^n))), \\ \text{s.t.} \quad & \theta^* \in \arg\min_{\theta} \mathcal{L}_{\text{ft'}}(\theta, f(\mathcal{X}^n)), \\ & ||f(\mathcal{X}^n) - \mathcal{X}^n||_p \leq \eta. \end{aligned} \tag{9}$$

We further categorize the threat model settings into the following types:

Regular setting. In this setting, the adversary utilizes an open-source Stable Diffusion [32] for fine-tuning training. During training, the special identifier "sks" is employed as "S*" along with the c_{pr} prior knowledge.

Preprocess setting. This setting is the focal point of our work, in which, the adversary still employs regular setting, but the images fed into the FT-T2I-DMs undergo JPEG compression by the adversary or social media. It is worth noting that these pre-processing steps are regulated to a certain intensity to ensure that the generated images remain authentic and natural. Excessive pre-processing might degrade the quality of the images [45].

Adverse settings. In these settings, the adversary's choice of the weight of pre-trained text-to-image diffusion model, fine-tuning method, training prompt, or pre-processing methods remains undisclosed.

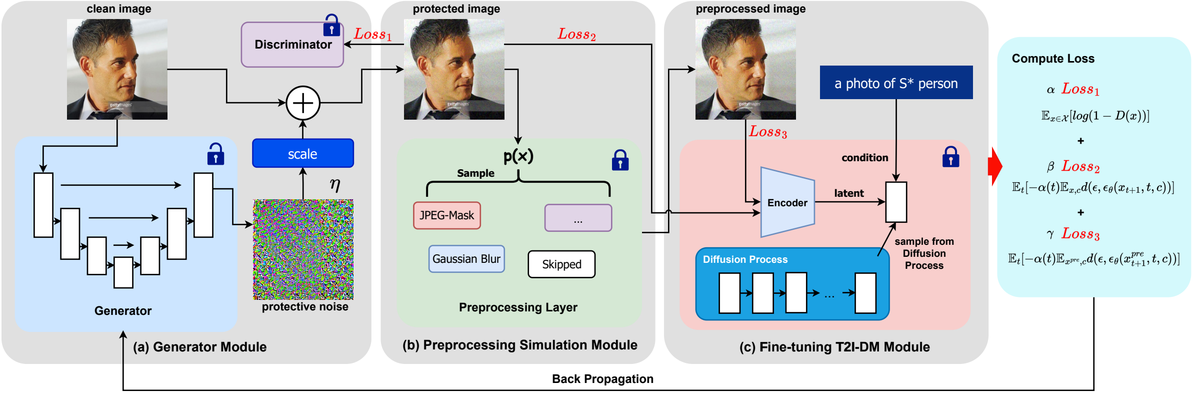

Fig. 5: Pipeline of GAP-Diff. We first input the clean images into the generator to get the output noises which is then scaled and concatenated with the clean image to create protected images. These protected images are fed into the discriminator and T2I-DM to obtain $Loss_1$, which measures the visual quality of protected images, and $Loss_2$, which directly contributes the adversarial features of the protective noise. Next, the protected images are passed through the pre-processing layer, and the preprocessed images are fed into the DM to obtain $Loss_3$ to counter JPEG compression and other pre-processing. Finally, according to different training strategies, these three losses are combined to optimize the generator.

IV. Methodology

A. Overview

GAP-Diff aims to disrupt the customization capability of FT-T2I-DMs by adding small perturbations δ to the set of images $\mathcal{X}^n$ that need protection. In other words, we aim to maximize the distortion introduced when these images $x = x^n + \delta$ are used for customization, such that adversaries cannot create clear, natural-looking, deceptive, or machine-usable fake images from $\mathcal{X}$ and $\mathcal{X}^{pre}$, which represent the set of protected images x and the set of preprocessed protected images x^{pre}. The customized outcomes of T2I-DM fine-tuned with the JPEG compressed images protected by GAP-Diff should exhibit one or more of the following characteristics:

- Poor image quality with obvious distortions, blurriness, grid patterns, or bubble-like cracks.
- Faces that are unrecognizable by humans or unusable for downstream tasks by machines.
- Faces that are extremely blurry or identities that do not match even if faces are present.

Towards these goals, we will provide detailed descriptions of the different modules of GAP-Diff in the following subsections. GAP-Diff is divided into three components. Firstly, in Section IV-B, we discuss the primary generator part requiring training. Next, in Section IV-C, we introduce the inserted pre-processing layer. Finally, in Section IV-D, we explain how we derive the adversarial optimization target for the FT-T2I-DMs. The pipeline is shown in Figure 5.

B. Generator Module

To establish the mapping f in Eq. 9, we aim to train a generator represented as $g_\psi(\cdot)$ to generate protective noise or protected images. Specifically, for the same facial images from the natural domain input $\mathcal{X}^n$, we can directly generate the protected image set $\mathcal{X}$ or the protective noise set Δ, where Δ is the set of δ and it corresponds one-to-one with the input $x^n \in \mathcal{X}^n$.

For the former, we can utilize a standard GAN architecture, where the input is an image x^n, and the output is directly a protected image x. Here, an MSE loss can be employed to enforce the similarity between x^n and x. For the latter, we can input an image x^n and output protective noise δ^0. This allows $\eta \times \delta^0$ to be added to x^n as δ to form the final protected image x, with η representing the noise budget that controls the stealthiness of the noise.

Similar to the observations in [30], we believe that the former method may result in perturbations that are either too small to be effective or too large causing significant visual changes in the entire image due to a lack of control over the noise. Therefore, we ultimately adopt the latter approach, directly managing the perturbation generation by scaling the noise through an activation function with the budget η, rather than obtaining the noise first and then truncating it to control its size as iterative methods typically do.

Consequently, we train a neural network to get robust protective perturbations. For its architecture, since we require the generated noise to be added to the original image, the network must be an end-to-end structure. Here, we opt for the classical U-Net architecture [33] as the generator, and design it to consist of convolution, deconvolution, and skip connections. We feed the input image x^n into the U-Net to obtain its output δ^0. Here, we apply a tanh function to constrain δ^0 to the range (-1, 1). Subsequently, we use η to constrain the size of the noise at the l_∞ norm level, that is, $||x - x^n||_\infty < \eta$, where $x = x^n + \delta$, $\delta = \eta \times \delta^0$ and $\delta^0 = g_\psi(x^n)$. This way, the neural network automatically constrains the output in terms of the l_∞ norm. The parameters of the generator ψ are obtained through computation and optimization of the following loss functions that will be introduced next.

Discriminator and GAN loss. For experimental rigor, we integrate an extra discriminator into the architecture to enhance the visual quality of the generated protected images. The discriminator employs conventional GAN loss to quantify the discrepancy between the adversarial example and the original image. The loss for discriminator is formulated as:

$$\mathcal{L}_{\text{GAN}}(x^n, x) = \mathbb{E}_{x^n \in \mathcal{X}^n}[logD(x^n)] + \mathbb{E}_{x \in \mathcal{X}}[log(1 - D(x))], \tag{10}$$

where $D(\cdot)$ represents the discriminator.

Once this component is added to the pipeline, the generator also needs to incorporate new generation loss terms to deceive the discriminator:

$$\mathcal{L}_{\text{D}}(x) = \mathbb{E}_{x \in \mathcal{X}}[log(1 - D(x))]. \tag{11}$$

C. Pre-processing Simulation Module

To enhance the resilience of generated images against JPEG compression (our main goal) and other pre-processing techniques in adverse settings, we design a pre-processing simulation module. It is mainly a pre-processing layer containing different pre-processing simulation functions $p(\cdot)$ that preprocess the input images and automatically sample a single function for each input. This involves leveraging JPEG-Mask, as introduced in Section III-A, which simulates differentiable JPEG compression. Additionally, we incorporate other pre-processing methods such as Gaussian blur, along with a Skipped function for training diversity.

It's important to note that while we utilize the JPEG-Mask from the steganography field, the optimization of the generator involves complex, time-varying information from diffusion, which completely differs from the simple decoder task. That means we should pay more attention to the functions comprising the pre-processing layer and ignore any information from real JPEG compression which can result in unnecessary zero-gradient updates, reducing training efficiency and simultaneously affecting the overall expected value, which is the primary objective of our optimization task as follows.

$$\mathbb{E}_{x^n \in \mathcal{X}^n, t \in (0,T)} \mathcal{L}_{\text{adv}}(p(x^n + \eta \times g_{\psi}(x^n)), t), \tag{12}$$

where T represents max diffusion training step in next module, $p(x^n + \eta \times g_{\psi}(x^n))$ can be represented as x^{pre} which belongs to $\mathcal{X}^{pre}$. L_{adv} denotes the final adversarial optimization objective.

As Eq. 12, through mixed training with the pre-processing layer, we can achieve training results that reflect the mathematical expectation across different conditions with and without pre-processing, rather than focusing solely on a single scenario. In other words, the approach can facilitate training towards a global optimum across multiple scenarios. Specifically, our pre-processing layer mainly includes: (1) The JPEG-Mask function, which simulates JPEG compression and repeatedly set in the pre-processing layer at different compression qualities, enables the gradient to be back-propagated to the generator, allowing it to learn adversarial features against JPEG compression. (2) The Skipped function, which applies no processing. Since our architecture needs to be effective both with and without pre-processing, learning adversarial features without pre-processing is essential for the generator. (3) Other pre-processing functions, which can be added as additional options in mixed training. This enables the generator to learn more robust features, like those against Gaussian blur.

D. Fine-tuning T2I-DM Module

For the Fine-tuning T2I-DM module, we disrupt the U-Net generation part following Eq. 9. Contrary to Eq. 3, where the U-Net aims to make the denoised distribution as close to the original distribution as possible, we aim to make the former far from the latter.

To achieve this goal, we seek the noise to exhibit adversarial characteristics to U-Net across all diffusion timesteps involved in training $(0, MaxTimeStep)$. As observed in [11], [47], the noise levels vary across different timesteps, resulting in different gradient information obtained during iterative attacks. We test the conditional losses of the fine-tuned model across different time intervals, as shown in Figure 6.

Due to the varying noise levels through diffusion, the adversarial characteristics at high timesteps (dominated by noise) may differ significantly from those at low timesteps (clear facial features). Therefore, if we only learn adversarial characteristics across specific time intervals, the learned characteristics may not persist across other timesteps. Using such training results for final inference can lead to two possible outcomes. One is that images with only adversarial characteristics to high time intervals can be overshadowed by the denoising process of low time intervals. The other is that images with only adversarial characteristics to low time intervals may allow DM to generate images already having facial contour features before it works, possibly only disrupting a few details during generation.

Thus, we believe it is necessary to consider information from both low and high timesteps to train for resilience and ignore the adversarial features of timesteps that are too high and are completely noisy. As a result, we incorporate a simple α function to balance adversarial information from different timesteps. After a series of tests and evaluations, including those illustrated in Figure 6, we set the α function as follows:

$$y = \begin{cases} 1 & \text{if } x \in (0, 800), \\ 0 & \text{otherwise.} \end{cases} \tag{13}$$

And the loss function for this part is as follows:

$$\mathcal{L}_{\text{adv}}(x, c, t) = \mathbb{E}_t[-\alpha(t)\mathbb{E}_{x,c} d(\epsilon, \epsilon_\theta(x_{t+1}, t, c))], \tag{14}$$

where c is the condition containing S^*, ϵ_θ represents the pre-trained U-Net, and $d(\cdot)$ measures the distance between variables.

E. Final Optimization Function

Combining the aforementioned modules, we aim to train the generator jointly with discriminator loss and adversarial losses. To balance the adversarial feature contributions before and after

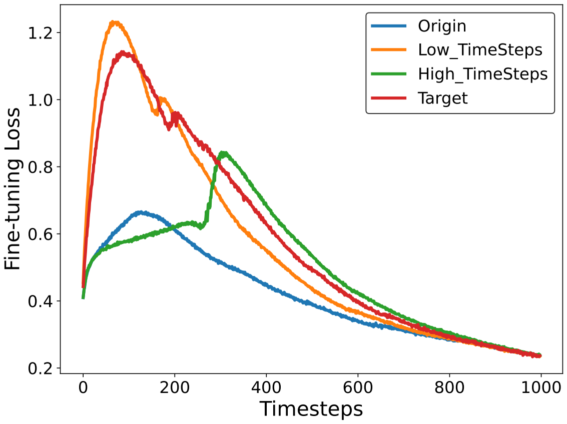

Fig. 6: Fine-tuning conditional loss corresponding to the all timesteps within the entire time interval under different conditions. The blue line represents the conditional loss of the original image input to the fine-tuned model (here DreamBooth). The orange and green lines correspond to the conditional losses of protected images with low and high time intervals trained separately input to the fine-tuned model. The red line indicates the desired fine-tuning loss of our protected images, which combines adversarial effects at both low and high time intervals; the higher this red line is overall, the better.

Algorithm 1: GAP-Diff framework.

```
Input: original images $x^n$, noise budget $\eta$, generator
       parameters $\psi$, pre-training epochs $N$, resume
       training epochs $N_2$, pre-processing layer $P$,
       diffusion max training step $T$, generic prompt $c$,
       weights of loss $\alpha$, $\beta$, $\gamma$
Output: trained parameters $\psi^*$
Initialize $P$ with $seq_P$=[JPEG-Mask, Skipped, GB, ...]
for each epoch in $N$ do
    for each batch in the epoch do
        $\delta^0 \leftarrow g_\psi(x^n)$    ▷ $g_\psi(\cdot)$ contains the $tanh(\cdot)$
                                       mapping $\delta^0 \in (-1, 1)$
        $x \leftarrow \eta \times \delta^0 + x^n$
        Sample $t$ uniformly from $(0, T)$
        $\mathcal{L} \leftarrow \alpha\mathcal{L}_\text{D}(x) + \beta\mathcal{L}_\text{adv}(x, c, t)$
        Backpropagate $\mathcal{L}$ and optimize $\psi$
for each epoch in $N_2$ do
    for each batch in the epoch do
        $\delta^0 \leftarrow g_\psi(x^n)$
        $x \leftarrow \eta \times \delta^0 + x^n$
        Sample a pre-processing function $p(\cdot)$ uniformly
          from $seq_P$,
        $x^{pre} \leftarrow p(x)$
        Sample $t$ uniformly from $(0, T)$
        $\mathcal{L} \leftarrow \alpha\mathcal{L}_\text{D}(x) + \beta\mathcal{L}_\text{adv}(x, c, t) + \gamma\mathcal{L}_\text{adv}(x^{pre}, c, t)$
        Backpropagate $\mathcal{L}$ and optimize $\psi$
return $\psi^*$
```

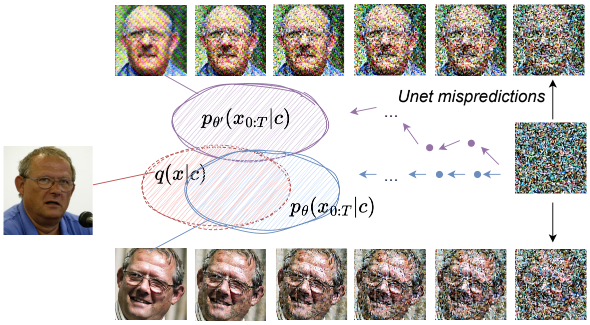

Fig. 7: The inference process of the disrupted DM. For the prior $q(x|c)$, there exists a distribution $p_\theta(x_{0:T}|c)$ predicted by the DM after learning from natural images. When the DM is trained on adversarial samples, during inference, the U-Net's predictions will gradually deviate from the original samples until they reach the adversarial distribution $p_{\theta'}(x_{0:T}|c)$ from the potential $\mathbb{P}_\text{adv}$ that the DM has been misled to learn. From the denoising process perspective, the U-Net will struggle to correctly denoise and produce natural backgrounds, facial features, etc., at different timesteps, especially at low timeteps.

the application of the pre-processing simulation module, we design the final loss function formulated as follows:

$$\mathcal{L}_\text{GAP-Diff} = \alpha\mathcal{L}_\text{D}(x) + \beta\mathcal{L}_\text{adv}(x, c, t) + \gamma\mathcal{L}_\text{adv}(x^{pre}, c, t), \tag{15}$$

where α, β and γ are the regularization terms used to balance the discriminator and adversary weights. Since this loss function is not based on ground truth, our task falls under the category of unsupervised learning. Based on the formula, the algorithm for GAP-Diff is illustrated in Algorithm 1, where we first consider using α and β to obtain the pure adversarial features, and then add γ to learn the robustness features against pre-processing.

As shown in Figure 7, the whole pipeline aims to guide the neural network to learn adversarial noises at different timesteps to shift the entire distribution used for customized inference away from the original data distribution. We believe that such a shifted distribution encompasses features devoid of any prior knowledge from the diffusion model. However, it still contains facial features specific to the individual and semantic features describing the individual (such as "a photo of person"). Consequently, the generated images will contain distorted, noisy, and partially recognizable facial characteristics while they are still photos of someone.

From the perspective of probability distributions, our network can be understood as performing unsupervised training to treat the generator's function as a parameterized density. In this case, the distribution of images generated by $g_\psi(\cdot)$ becomes the $\mathbb{P}_\psi$ distribution. If we denote the true distribution with prior knowledge of the diffusion model as $\mathbb{P}_r$, then $\mathbb{P}_\text{adv}$ represents a distribution within the diffusion model that is adversarial to a specific true distribution, for which the diffusion model lacks prior knowledge about adversarial examples under different conditions. In other words, the objective of GAP-Diff is to get

TABLE I: Performance comparison using different metrics for GAP-Diff on VGGFace2. The protected images output by all methods in the table are subjected to JPEG compression with $Q = 70$ and then input into the customization model fine-tuned with DreamBooth to obtain corresponding evaluation metrics. "↑" means the higher the better while "↓" means the lower the better.

Methods	**"a photo of sks person"**				**"a dslr portrait of sks person"**			
	FDFR↑	ISM↓	BRISQUE↑	SER-FIQ↓	FDFR↑	ISM↓	BRISQUE↑	SER-FIQ↓
No Defense	5.33	0.61	26.27	0.69	21.57	0.48	9.64	0.71
Photoguard [36]	6.22	0.55	29.38	0.71	19.44	0.46	13.74	0.71
Glaze [38]	6.57	0.53	30.42	0.69	18.78	0.45	11.04	0.69
Mist [24]	14.89	0.46	35.68	0.60	19.56	0.38	20.43	0.63
Anti-DB [45]	22.89	0.41	40.19	0.40	32.67	0.34	32.72	0.44
ACE [53]	8.44	0.47	37.22	0.61	15.22	0.38	27.80	0.64
MetaCloak [26]	31.69	0.44	38.82	0.51	35.28	0.36	27.31	0.56
CAAT [49]	25.44	0.43	42.01	0.45	21.67	0.38	25.07	0.57
SimAC [47]	19.11	0.49	39.43	0.52	23.56	0.41	24.15	0.62
GAP-Diff (ours)	**77.56**	**0.25**	**42.04**	**0.23**	**76.33**	**0.19**	**48.97**	**0.20**

Methods	**"a photo of sks person looking at the mirror"**				**"a photo of sks person in front of eiffel tower"**			
	FDFR↑	ISM↓	BRISQUE↑	SER-FIQ↓	FDFR↑	ISM↓	BRISQUE↑	SER-FIQ↓
No Defense	8.67	0.44	19.61	0.56	20.67	0.22	20.11	0.44
Photoguard [36]	9.67	0.40	23.43	0.56	19.56	0.21	17.91	0.45
Glaze [38]	9.33	0.41	19.69	0.55	17.44	0.20	19.78	0.43
Mist [24]	12.33	0.35	22.17	0.50	27.33	0.18	21.05	0.36
Anti-DB [45]	21.67	0.30	24.77	0.37	34.88	0.14	31.21	0.26
ACE [53]	12.44	0.30	31.90	0.42	36.11	0.15	25.25	0.26
MetaCloak [26]	32.76	0.32	34.14	0.36	30.57	0.15	31.22	0.25
CAAT [49]	16.33	0.32	23.82	0.37	34.22	0.14	31.82	0.25
SimAC [47]	14.89	0.33	31.09	0.42	28.56	0.14	32.98	0.25
GAP-Diff (ours)	**84.56**	**0.14**	**47.30**	**0.13**	**72.78**	**0.08**	**41.69**	**0.08**

the minimum value of $KL(\mathbb{P}_{\psi}||\mathbb{P}_{\text{adv}})$.

In Algorithm 1, we achieve this by optimizing ψ through $\mathcal{L}_{\text{GAP-Diff}}$. During inference, according to the DDPM [16] inference Eq 16 and as illustrated in Figure 7, the generated images gradually shift towards the distribution of adversarial samples due to the adversarial features learned by the U-Net during training as described by Eq 17.

$$x_{t-1} = \frac{1}{\sqrt{\alpha_t}}(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t, t, c)) + \sigma_t z, \tag{16}$$

$$x'_{t-1} = \frac{1}{\sqrt{\alpha_t}}(x'_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon'_\theta(x'_t, t, c)) + \sigma_t z, \tag{17}$$

where x_t and ϵ_θ respectively represent the noisy variable of clean images and the LDM pre-trained U-Net. While x'_t and ϵ'_θ represent the noisy variable of adversarial images (protected images) and the U-Net that, after being fine-tuned using the protected images, has been misled by adversarial samples.

V. Evaluation

A. Setup

Dataset. We utilize three widely-used facial datasets FFHQ [20], CelebA-HQ [19] and VGGFace2 [3] in our experiments. The FFHQ dataset contains $70,000$ high-quality and lossless PNG images. CelebA-HQ is an enhanced version of the original CelebA dataset consisting of $30,000$ celebrity face images. VGGFace2 is a comprehensive dataset with over 3.3 million face images from $9,131$ unique identities. The resolution of all images in the datasets is set to 512×512. It is worth mentioning that, since the primary objective of our work is to resist JPEG compression and the CelebA-HQ dataset is already JPEG-compressed, most of our experiments are conducted on the lossless datasets FFHQ and VGGFace2.

T2I-DM weight. We utilize the most widely used and open-source model weights from Stable Diffusion [32] for training and testing. In our experiments, we primarily use the SD-v2.1 weights, as it is the latest and most popular, effective architecture based on the U-Net diffusion model. To test the performance of GAP-Diff under adverse setting, we assume the versions of Stable Diffusion between anti-customization and customization are the same or different.

Fine-tuning method. Consistent with [26], [45], [47], among all methods for fine-tuning text-to-image diffusion models, we choose DreamBooth [34], one of the best-performing and most widely used fine-tuning methods, as our primary experimental subject. Further, in subsequent comparative experiments, we use DreamBooth-based LoRA [35] and SVDiff [15], which are also popular and perform well in facial customization, to conduct comparative analyses.

Baseline. We compare several open-source state-of-art models designed to disrupt the training or customization of text-to-image diffusion models, including PhotoGuard [36], Glaze [38], Mist [24], Anti-DreamBooth [45], ACE [53], MetaCloak [26], CAAT [49] and SimAC [47]. Due to memory and runtime constraints, MetaCloak is only compared on the VGGFace2 dataset, which is the primary focus of this paper.

Metric. Consistent with [45], [47], we use RetinaFace [6] as the face detector to determine whether a face is present in the image, recorded as the Face Detection Failure Rate (FDFR). When a face is detected, we use ArcFace [7] to compute the cosine similarity between the face encoding and the original

TABLE II: Performance comparison using different metrics for GAP-Diff on CelebA-HQ. The protected images output by all methods in the table are subjected to JPEG compression with $Q = 70$ and then input into the customization model fine-tuned with DreamBooth to obtain corresponding evaluation metrics.

Methods	“a photo of sks person”				“a dslr portrait of sks person”			
	FDFR↑	ISM↓	BRISQUE↑	SER-FIQ↓	FDFR↑	ISM↓	BRISQUE↑	SER-FIQ↓
No Defense	6.67	0.63	16.67	0.72	20.78	0.48	5.25	0.69
Photoguard [36]	5.78	0.53	24.56	0.69	22.44	0.47	12.46	0.72
Glaze [38]	6.12	0.57	30.78	0.72	25.56	0.40	17.97	0.70
Mist [24]	11.56	0.50	36.87	0.67	28.78	0.35	24.08	0.71
Anti-DB [45]	41.44	0.42	40.98	0.33	36.56	0.33	34.98	0.53
ACE [53]	10.00	0.53	36.89	0.70	18.22	0.32	30.94	0.73
CAAT [49]	42.56	0.45	45.76	0.42	22.33	0.37	28.47	0.67
SimAC [47]	25.78	0.51	40.21	0.61	23.00	0.39	33.68	0.69
GAP-Diff (ours)	**78.67**	**0.28**	**43.39**	**0.32**	**60.22**	**0.20**	**43.00**	**0.31**

Methods	“a photo of sks person looking at the mirror”				“a photo of sks person in front of eiffel tower”			
	FDFR↑	ISM↓	BRISQUE↑	SER-FIQ↓	FDFR↑	ISM↓	BRISQUE↑	SER-FIQ↓
No Defense	8.56	0.44	18.97	0.56	21.67	0.19	17.51	0.42
Photoguard [36]	9.33	0.43	21.43	0.54	29.11	0.18	19.85	0.38
Glaze [38]	9.44	0.41	17.97	0.55	25.33	0.14	19.54	0.39
Mist [24]	12.22	0.34	25.18	0.52	34.78	0.12	20.41	0.30
Anti-DB [45]	27.33	0.28	29.82	0.36	38.33	0.09	31.64	0.26
ACE [53]	14.56	0.28	31.47	0.45	47.33	0.09	28.28	0.19
CAAT [49]	17.67	0.31	24.71	0.41	36.67	0.10	31.04	0.23
SimAC [47]	18.11	0.33	33.68	0.42	34.98	0.12	30.95	0.25
GAP-Diff (ours)	**81.33**	**0.21**	**49.14**	**0.21**	**77.67**	**0.06**	**46.61**	**0.07**

identity encoding, recorded as the Identity Score Matching (ISM). Additionally, we use BRISQUE [28], a classic and commonly used image quality assessment metric, and SER-FIQ [44], another advanced face quality assessment metric.

Implementation details. We use “a photo of sks person” (the same prompt as the existing work when fine-tuning T2I-DM) as the condition to obtain the loss function for Fine-tuing T2I-DM Module. In the experiments, all noise budgets are set to $16/255$, which provides an effective balance between perturbation capability and visual quality. For training details of the generator, we set the optimizer to Adam with a learning rate of 0.001, and set the discriminator weight α to 0.001. During training, we obtain a training set in FFHQ with $20,000$ randomly chosen images and employ a “resume training” strategy. Specifically, we first pre-train the generator on images without pre-processing for 40 epochs to establish base protective generation capabilities. Then, we continue training for an additional 10 epochs with the pre-processing layer added. The pre-processing layer consists of JPEG-Mask with two quality levels: $Q = 70$, which is commonly used in real-world JPEG compression, and $Q = 50$, which presents more challenging compression tasks. Additionally, we apply Gaussian blur with $K = 7$ for transformation resilience and a Skipped function to handle unprocessed inputs. During this latter phase, β is set to 0.6 and γ is set to 0.4 based on empirical performance. Aligning with Anti-DreamBooth, we train each text encoder and U-Net model of DreamBooth with batch size of 2 and learning rate of $5e{-}7$ for $1,000$ training steps.

To ensure diversity in our experimental inference statements, we select a union of inference prompts from Anti-DreamBooth [45] and SimAC [47]. The prompts are as follows: PromptA “a photo of sks person”, PromptB “a dslr portrait of sks person”, PromptC “a photo of sks person looking at the mirror”, and PromptD “a photo of sks person in front of eiffel tower”. For each prompt, we first sample 30 identities in face datasets, then generate 30 images per identity and finally use all these generated images to calculate the evaluation metrics and report their average values.

B. Comparison with Baseline Methods

To evaluate the effectiveness of GAP-Diff, we conduct quantitative and qualitative comparisons under four prompts with different identities on widely used datasets, compared to the state-of-the-art methods. Specifically, we first use the fully trained GAP-Diff model, which was trained on FFHQ with a randomly chosen set of $20,000$ images for approximately 120 GPU hours. Note that, while costly, the trained model is scalable and could generate protective noise in milliseconds. We then apply this model to generate four protected images for each of the identities in the VGGFace2 and CelebA-HQ datasets. These protected images are then JPEG-compressed at $Q = 70$, which is a lower end of commonly used JPEG compression quality on social networks [29], [43], to demonstrate the effectiveness of GAP-Diff. The compressed images are subsequently input into the customization model fine-tuned with DreamBooth.

Quantitative results. As shown in Tables I and II, the comparison of the evaluation metrics reveals that GAP-Diff significantly outperforms the state-of-the-art works across all prompts. For instance, GAP-Diff achieves a $\sim$30% higher FDFR across all prompts compared to the best one of existing works. Additionally, ISM and SER-FIQ are reduced to extremely low ranges, indicating both low person identity matching rates and exceptionally low face generation quality. For BRISQUE, our values exceeding 40 indicate extremely poor image quality across all prompts. We attribute this to the learning capabilities

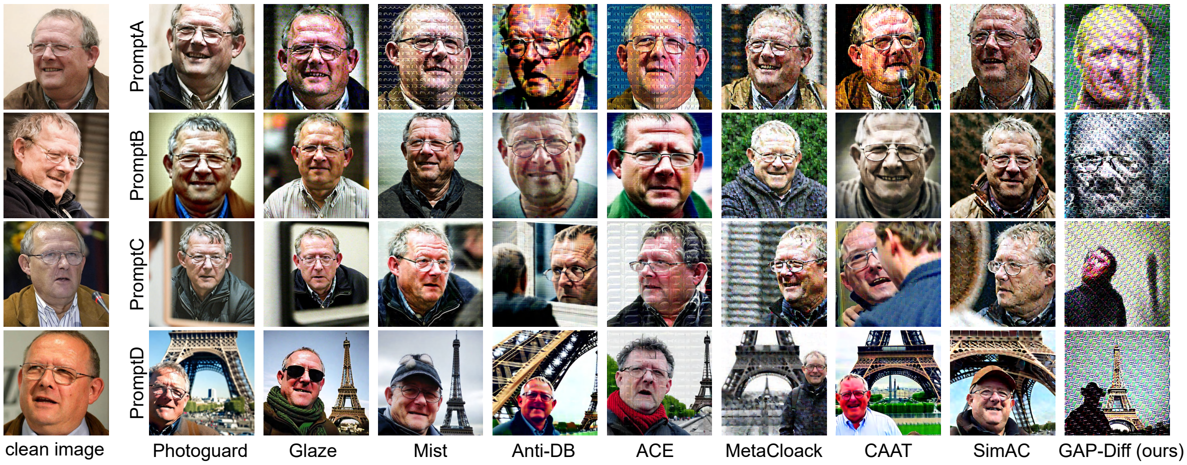

Fig. 8: Visualization results (four prompts) on VGGFace2. The first column shows clean identity photos, while the columns on the right depict results obtained by first protecting clean photos using different methods, then compressing them with JPEG $Q = 70$, and finally getting customized outcomes from the customization model fine-tuned with DreamBooth.

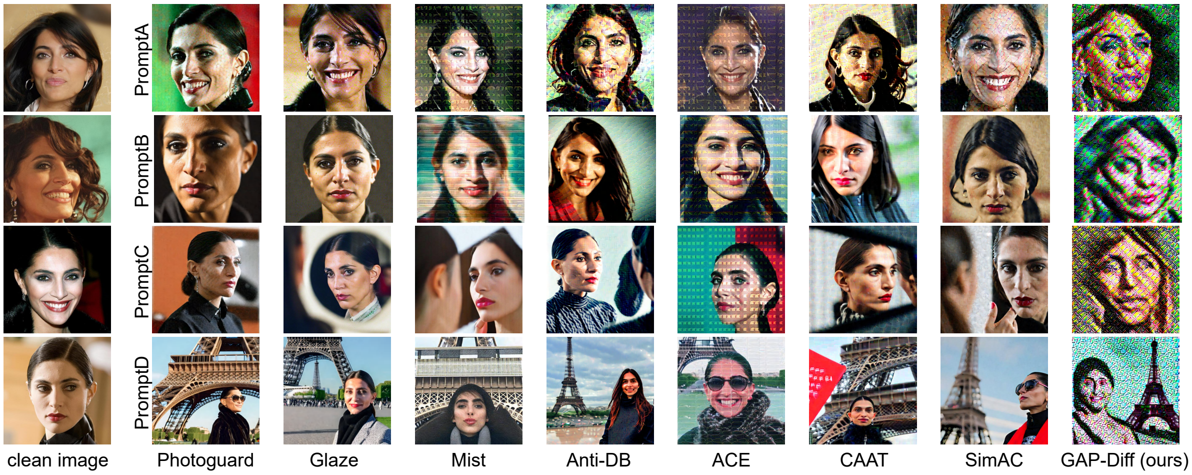

Fig. 9: Visualization results (four prompts) on CelebA-HQ. The first column shows clean identity photos, while the columns on the right depict results obtained by first protecting clean photos using different methods, then compressing them with JPEG $Q = 70$, and finally getting customized outcomes from the customization model fine-tuned with DreamBooth.

of our generative framework. Specifically, GAP-Diff makes it easier to find the globally optimal solution across all timesteps compared to existing iterative approaches. Further, GAP-Diff can simulate real JPEG compression during training and use it as gradient information to backpropagate and optimize the generator, while existing works struggle to achieve this due to the limits of their frameworks. These two aspects make our noise more robust cause higher face detection failure rates and often poorer image quality. As a result, GAP-Diff proves more effective in protecting faces from being customized in real social network scenarios.

Qualitative results. We present some of the visual results on VGGFace2 and CelebA-HQ dataset in Figure 8 and Figure 9. Compared to existing works, GAP-Diff clearly achieves superior visual protection. This is because GAP-Diff tends to generate protective noise that is concentrated in the low-frequency region of images, making it more resistant to JPEG compression. In contrast, while SimAC is an improved method based on Anti-DreamBooth and significantly enhances performance [47], its resistance to JPEG compression is lower. This is because SimAC focuses more on capturing high-frequency information in the U-Net feature layers during improvement, leading the protective noise to deviate more from the low-frequency region. Moreover, when our framework achieves better optimization for facial data, the generated protective noise tends to have stronger adversarial effects within the time interval when obtaining the adversarial loss. This means that both detailed and edge features of the face are more challenging for the FT-T2I-DM to generate, making the facial features more blurred overall. Consequently, the minimum amount of facial information is exposed in the customization model's output, achieving superior protection.

Why does GAP-Diff outperform the existing work? Figure 11

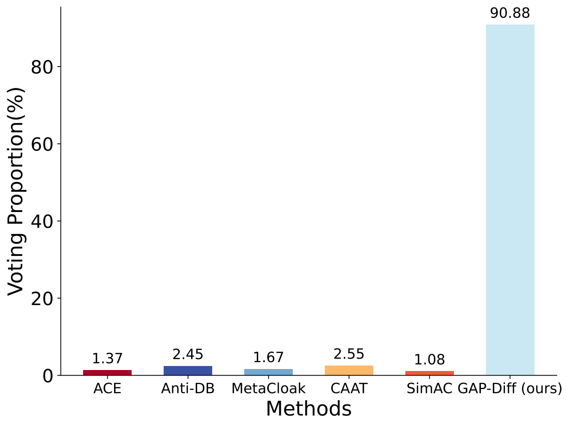

Fig. 10: Human study results on face protection effectiveness.

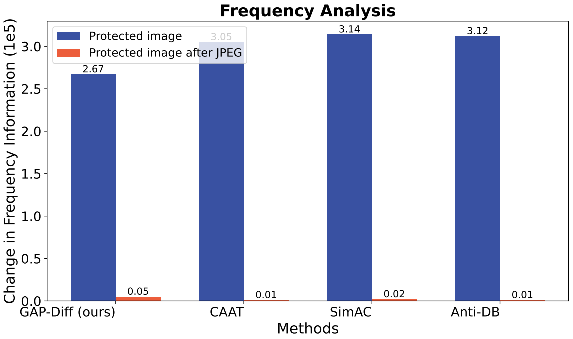

Fig. 11: Comparison of change in high-frequency information relative to the original image between images protected by different methods after DCT transformation. The blue bars in the figure represent the change in the high-frequency regions of images protected by different methods compared to the original image, while the orange bars show the change in high-frequency regions after JPEG compression at $Q = 70$.

reveals that JPEG compression removes the high-frequency protective noise introduced by existing methods, rendering them ineffective. GAP-Diff relies least on high-frequency information, which is largely removed from all images. This partially explains why protection methods dependent on high-frequency details fail, while GAP-Diff remains more robust due to its reduced reliance on compressible information.

One more thing - Human survey. To better evaluate the protective effectiveness of GAP-Diff under JPEG compression with $Q = 70$, we conducted a human study using a survey composed of 10 single-choice questions. Each question presents an image from VGGFace2 of a specific individual. Users are asked to select the most obscured and difficult-to-identify image of the person's identity among the customized outcomes of T2I-DM fine-tuned with the JPEG compressed images protected by baseline methods and ours. (We directly call them the customized outcomes of baseline methods and ours.)

We collected surveys from 102 participants, most of whom are not familiar with adversarial attacks or image compression techniques. Figure 10 illustrates the results, demonstrating that GAP-Diff not only surpasses state-of-the-art methods in the evaluation metrics shown in Table I but also proves to be more effective in protecting facial privacy as perceived by human eyes. The participants consistently find that the customized outcomes of ours are more difficult for them to recognize faces or identities. Therefore, it is more practical for real-world applications.

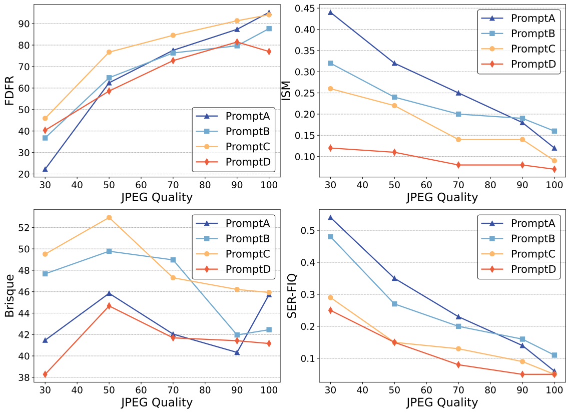

Fig. 12: Quantitative comparison of protection effectiveness under different JPEG compression qualities on VGGFace2. Here, $Q = 100$ indicates no compression.

C. Ablation Studies

JPEG compression qualities. In the pre-processing setting, besides the commonly used JPEG compression quality in social media, potential adversaries may use different compression qualities or even no compression according to the regular setting. As shown in Figure 12 and Figure 13, GAP-Diff demonstrates strong adversarial effects across various JPEG compression levels. Even at a compression quality of 30, which significantly distorts details, GAP-Diff still maintains effective protection, with BRISQUE exceeding 38 and the generated images visually exhibiting grid patterns or bubble-like cracks. For JPEG $Q = 50$, we conduct additional quantitative and qualitative experiments, as shown in Table III and Figure 14. The results show that existing methods fail completely at $Q = 50$, while GAP-Diff still maintains strong protective performance. The results also confirm that GAP-Diff performs well with other image formats, including uncompressed images ($Q = 100$), further validating its robustness across varying compression scenarios.

Noise budget. We adjust the noise budget to test protection under different η limits. As shown in Table IV and Figure 15, GAP-Diff already demonstrates protection effectiveness under the noise budget $\eta = 8/255$ with low face matching accuracy and poor image quality. With a perturbation size of $\eta = 32/255$, it can completely prevent any prompt customization with FDFR approaching 100%, while ISM and SER-FIQ are nearly 0. Therefore, GAP-Diff achieves better defense performance with larger noise budgets, although this comes at the cost of increased noise visibility.

TABLE III: Performance comparison using different metrics for GAP-Diff on VGGFace2. The protected images output by all methods in the table are subjected to JPEG compression with $Q = 50$ and then input into the customization model fine-tuned with DreamBooth to obtain corresponding evaluation metrics.

Methods	**“a photo of sks person”**				**“a dslr portrait of sks person”**			
	FDFR↑	ISM↓	BRISQUE↑	SER-FIQ↓	FDFR↑	ISM↓	BRISQUE↑	SER-FIQ↓
Photoguard [36]	4.78	0.55	32.72	0.70	22.89	0.44	9.52	0.68
Glaze [38]	4.89	0.56	30.81	0.71	23.11	0.43	8.76	0.68
Mist [24]	10.89	0.52	34.92	0.69	21.22	0.42	12.02	0.68
Anti-DB [45]	10.44	0.50	36.45	0.57	23.00	0.39	19.33	0.62
ACE [53]	7.55	0.51	37.94	0.68	17.67	0.38	21.24	0.67
MetaCloak [26]	32.16	0.46	40.05	0.54	38.25	0.41	27.76	0.60
CAAT [49]	10.44	0.52	37.50	0.61	17.44	0.42	13.90	0.65
SimAC [47]	10.22	0.51	37.19	0.65	18.89	0.43	14.60	0.67
Ours	**62.44**	**0.32**	**45.85**	**0.35**	**64.80**	**0.24**	**49.78**	**0.27**
Methods	**“a photo of sks person looking at the mirror”**				**“a photo of sks person in front of eiffel tower”**			
	FDFR↑	ISM↓	BRISQUE↑	SER-FIQ↓	FDFR↑	ISM↓	BRISQUE↑	SER-FIQ↓
Photoguard [36]	4.33	0.39	20.76	0.55	19.56	0.21	17.91	0.45
Glaze [38]	6.89	0.40	20.71	0.56	20.56	0.20	20.34	0.44
Mist [24]	7.78	0.37	22.69	0.51	21.56	0.18	20.91	0.38
Anti-DB [45]	17.78	0.32	14.82	0.42	26.89	0.17	28.11	0.28
ACE [53]	7.89	0.32	28.09	0.48	32.56	0.16	22.97	0.31
MetaCloak [26]	28.40	0.33	33.87	0.38	24.73	0.17	30.89	0.27
CAAT [49]	11.11	0.36	13.17	0.46	28.33	0.18	27.84	0.31
SimAC [47]	7.22	0.36	26.90	0.49	20.67	0.18	27.57	0.35
Ours	**76.73**	**0.22**	**52.93**	**0.15**	**58.61**	**0.11**	**44.67**	**0.15**

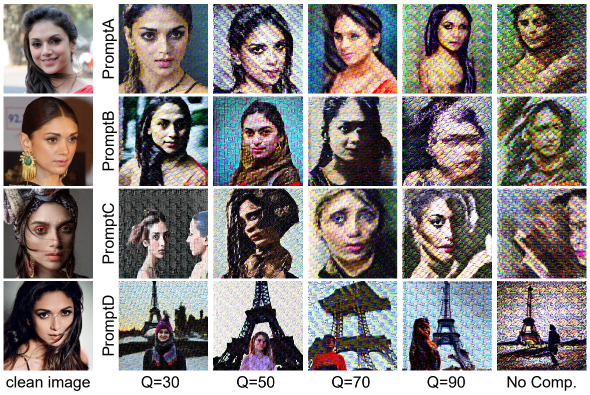

Fig. 13: Qualitative Comparison of protection effectiveness under different JPEG compression qualities on VGGFace2.

Training epochs. To evaluate training efficiency, we conduct experiments by comparing different pre-training steps with the same resume training steps to assess the impact on protection effectiveness. Our results in Figure 16 of FDFR at different training epochs indicate that our model achieves a significant protection effect with only 10 epochs of pre-training, reaching at least 65% FDFR under different prompts, and the performance continues to improve with additional training epochs. Given that our model quickly generates images once trained, investing additional time to discover a more robust training strategy and model is highly cost-effective.

D. Adverse Settings

TABLE IV: Quantitative comparison of protection effectiveness under different noise budget on VGGFace2.

η	**“a photo of sks person”**			
	FDFR↑	ISM↓	BRISQUE↑	SER-FIQ↓
8/255	32.78	0.41	34.65	0.42
12/255	84.33	0.21	36.85	0.17
16/255	95.22	0.12	45.73	0.06
32/255	100.0	0.00	42.07	0.01
η	**“a dslr portrait of sks person”**			
	FDFR↑	ISM↓	BRISQUE↑	SER-FIQ↓
8/255	42.44	0.29	35.33	0.38
12/255	80.78	0.18	36.92	0.18
16/255	87.67	0.16	42.44	0.11
32/255	99.67	0.01	46.02	0.01
η	**“a photo of sks person looking at the mirror”**			
	FDFR↑	ISM↓	BRISQUE↑	SER-FIQ↓
8/255	61.22	0.25	45.94	0.19
12/255	71.22	0.18	44.26	0.10
16/255	94.11	0.09	45.93	0.05
32/255	98.00	0.01	46.42	0.01
η	**“a photo of sks person in front of eiffel tower”**			
	FDFR↑	ISM↓	BRISQUE↑	SER-FIQ↓
8/255	60.33	0.11	40.05	0.15
12/255	66.78	0.09	39.62	0.14
16/255	77.00	0.07	41.15	0.06
32/255	92.87	0.03	40.82	0.02

Prompt mismatch. When adversaries train their own DreamBooth models, they may not necessarily use the same special identifier “sks” as we do during training (although “sks” is considered the default optimal DreamBooth customization prompt). We attempt to replace “sks” with another special identifier like “t@t,” which is exactly the same used by Anti-DreamBooth [45] and SimAC [47] in comparative experiments, and the results are shown in Table V and Figure 17. Under the first two prompts, GAP-Diff still demonstrates strong protection effectiveness. Under the last two prompts, although performance on FDFR and BRISQUE decreases, the key metric ISM remains

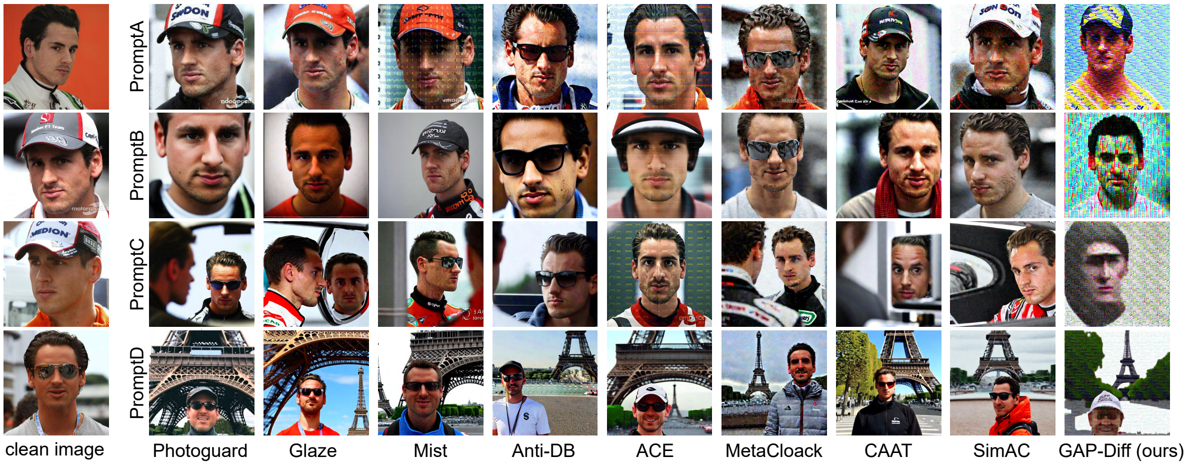

Fig. 14: Visualization results (four prompts) on VGGFace2. The first column shows clean identity photos, while the columns on the right depict results obtained by first protecting clean photos using different methods, then compressing them with JPEG $Q = 50$, and finally getting customized outcomes from the customization model fine-tuned with DreamBooth.

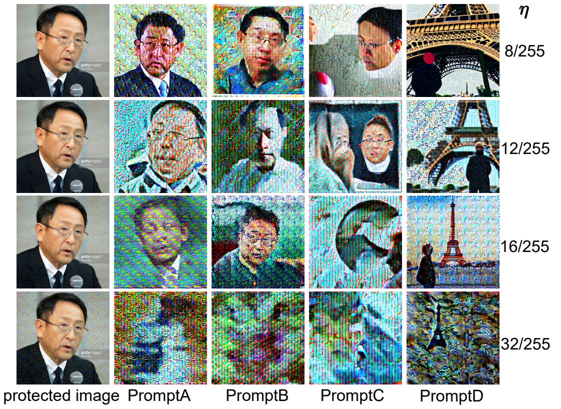

Fig. 15: Qualitative comparison of protection effectiveness under different noise budget on VGGFace2.

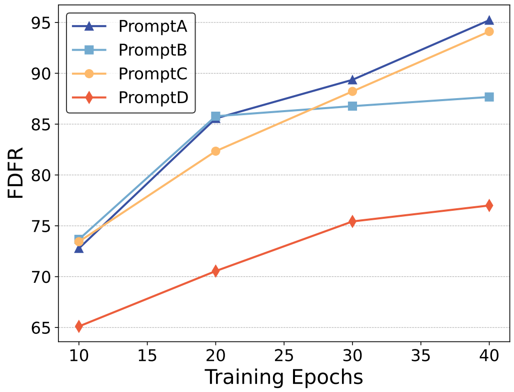

Fig. 16: FDFR at different training epochs on VGGFace2, by observing the changes in which, we can discern the impact of different training epochs on the protection effectiveness.

TABLE V: Quantitative results of prompt mismatch between training and testing on VGGFace2. The training prompt is "a photo of sks person" and the inference prompt uses special identifier "sks" or "t@t".

Train S*	Test S*	"a photo of S* person"			
		FDFR↑	ISM↓	BRISQUE↑	SER-FIQ↓
sks	sks	95.22	0.12	45.73	0.06
sks	t@t	88.78	0.13	45.00	0.11
Train S*	**Test S***	**"a dslr portrait of S* person"**			
		FDFR↑	ISM↓	BRISQUE↑	SER-FIQ↓
sks	sks	87.67	0.16	42.44	0.11
sks	t@t	72.89	0.14	36.19	0.20
Train S*	**Test S***	**"a photo of S* person looking at the mirror"**			
		FDFR↑	ISM↓	BRISQUE↑	SER-FIQ↓
sks	sks	94.11	0.09	45.93	0.05
sks	t@t	46.44	0.16	34.79	0.23
Train S*	**Test S***	**"a photo of S* person in front of eiffel tower"**			
		FDFR↑	ISM↓	BRISQUE↑	SER-FIQ↓
sks	sks	77.00	0.07	41.15	0.06
sks	t@t	32.67	0.09	32.54	0.28

low, not exceeding 0.16. That means while a face may have been generated, its identity still does not match closely, proving that GAP-Diff still provides effective protection.

Different fine-tuning methods. Furthermore, the protected images we generate could be used by adversaries to fine-tune T2I-DM with different methods. We compare the results in Table VI against different fine-tuning methods and prompts. Our best performance is observed using DreamBooth with full utilization of the conditional loss function. Since LoRA and SVDiff only fine-tune the weight matrix, unlike DreamBooth which conduct complete customized training on the entire text-encoder and U-Net, the disruption of generation is not as severe as with DreamBooth. However, the results with high FDFR up to 63% using LoRA and 98% using SVDiff along with other metrics indicating low face generation quality, demonstrate that GAP-Diff still achieves a strong protective effect.

Different pre-processing methods. Apart from common

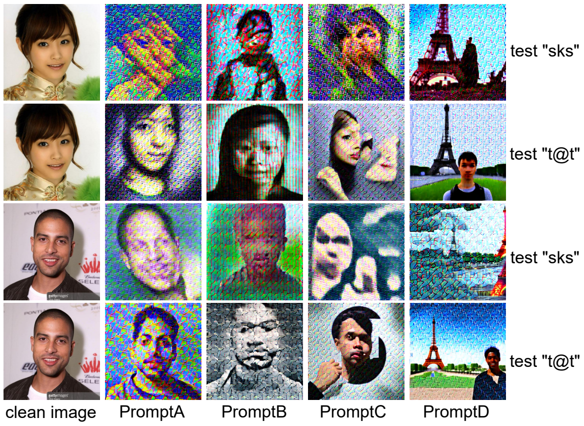

Fig. 17: Qualitative results of prompt mismatch between training and testing on VGGFace2.

TABLE VI: Quantitative results of fine-tuning T2I-DM with different methods on VGGFace2 with GAP-Diff.

Fine-tuning Method	**“a photo of sks person”**			
	FDFR↑	ISM↓	BRISQUE↑	SER-FIQ↓
DreamBooth [34]	95.22	0.12	45.73	0.06
LoRA [35]	63.44	0.18	53.48	0.24
SVDiff [15]	98.44	0.02	52.37	0.01
Fine-tuning Method	**“a dslr portrait of sks person”**			
	FDFR↑	ISM↓	BRISQUE↑	SER-FIQ↓
DreamBooth [34]	87.67	0.16	42.44	0.11
LoRA [35]	54.00	0.18	51.38	0.34
SVDiff [15]	64.67	0.12	28.72	0.26
Fine-tuning Method	**“a photo of sks person looking at the mirror”**			
	FDFR↑	ISM↓	BRISQUE↑	SER-FIQ↓
DreamBooth [34]	94.11	0.09	45.93	0.05
LoRA [35]	58.78	0.12	41.89	0.11
SVDiff [15]	65.56	0.12	42.70	0.12
Fine-tuning Method	**“a photo of sks person in front of eiffel tower”**			
	FDFR↑	ISM↓	BRISQUE↑	SER-FIQ↓
DreamBooth [34]	77.00	0.07	41.15	0.06
LoRA [35]	63.11	0.09	44.08	0.11
SVDiff [15]	47.56	0.06	34.25	0.16

pre-processing methods like JPEG compression, potential adversary may also employ other pre-processing techniques before customizing images. Consistent with CAAT [49], we experiment with random noise set at a scale of 0.05, Gaussian blur applied with a kernel size of 3 and sigma of 0.05, and image quantization reducing 8-bit to 6-bit. Additionally, we include experiments that involve downscaling by halving the original size, followed by super-resolution, as well as standard resolution experiments at the reduced size.

Table VII demonstrates the resistance of GAP-Diff against these alternative pre-processing methods. Due to our strong noise and the addition of Gaussian blur in the pre-processing layer during training, the evaluation metrics of most pre-processing methods are as good as the output results of directly customizing protected images without pre-processing. However, random noise significantly disrupts our noise structure at the pixel level, thereby reducing the protection effectiveness. We speculate that enhancing the robustness of protective noise against random noise can still, similar to how resistance to JPEG compression is achieved, be accomplished by training with a random noise function added in the pre-processing layer.

TABLE VII: Quantitative results of GAP-Diff against other pre-processing methods on VGGFace2.

Method	**“a photo of sks person”**			
	FDFR↑	ISM↓	BRISQUE↑	SER-FIQ↓
w/o preprocess	95.22	0.12	45.73	0.06
Random noise	26.44	0.41	29.41	0.49
Gaussian blur	83.78	0.24	42.25	0.18
Quantization	93.89	0.11	43.44	0.06
Resize	90.56	0.17	42.50	0.12
Super resolution	93.67	0.13	42.82	0.10
Method	**“a dslr portrait of sks person”**			
	FDFR↑	ISM↓	BRISQUE↑	SER-FIQ↓
w/o preprocess	87.67	0.16	42.44	0.11
Random noise	35.33	0.31	36.30	0.45
Gaussian blur	82.44	0.19	46.67	0.12
Quantization	85.44	0.15	43.05	0.12
Resize	82.67	0.17	41.89	0.15
Super resolution	85.22	0.16	42.75	0.15
Method	**“a photo of sks person looking at the mirror”**			
	FDFR↑	ISM↓	BRISQUE↑	SER-FIQ↓
w/o preprocess	94.11	0.09	45.93	0.05
Random noise	30.46	0.28	34.73	0.32
Gaussian blur	86.11	0.14	44.71	0.09
Quantization	94.00	0.11	46.63	0.05
Resize	87.11	0.18	46.16	0.09
Super resolution	90.11	0.15	49.24	0.08
Method	**“a photo of sks person in front of eiffel tower”**			
	FDFR↑	ISM↓	BRISQUE↑	SER-FIQ↓
w/o preprocess	77.00	0.07	41.15	0.06
Random noise	24.89	0.12	26.89	0.32
Gaussian blur	73.11	0.08	43.90	0.07
Quantization	77.33	0.08	41.06	0.06
Resize	78.33	0.08	41.34	0.06
Super resolution	78.44	0.09	40.43	0.07

T2I-DM weight mismatch. In practice, the adversary may not necessarily use the same T2I-DM weights for fine-tuning as those used to train our generator. We compare the most commonly used Stable Diffusion versions: v2.1 from v2 and v1.4, v1.5 from v1. The results are shown in Table VIII. Similar to the prompt mismatch scenario, some prompts achieve the same protection effectiveness as when the weights match, while for other prompts, the critical ISM metric remains low when the weights do not match, and the above phenomenon is explained in Figure 18. This indicates that GAP-Diff can still protect user images in different T2I-DM weight scenarios.

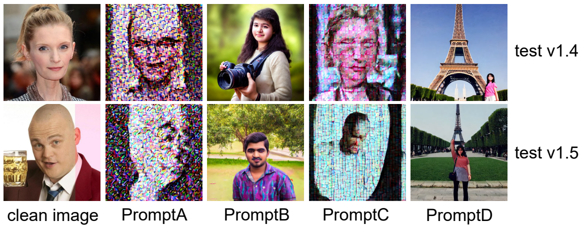

Fig. 18: T2I-DM weight mismatch results on VGGFace2 in visual interpretations where FDFR is high and ISM is low. In these cases, the inferred images and personalities are completely different from the actual individuals.

E. Cost comparison

We compare the time and memory costs of generating 4 protected images using different methods under official settings

TABLE VIII: Stable Diffusion weights version mismatch during training and testing on VGGFace2. The weight used during training is v2.1.

Train	Test	**"a photo of sks person"**			
		FDFR↑	ISM↓	BRISQUE↑	SER-FIQ↓
v2.1	v2.1	95.22	0.12	45.73	0.06
	v1.4	93.67	0.14	39.68	0.07
	v1.5	82.33	0.17	37.69	0.13
Train	**Test**	**"a dslr portrait of sks person"**			
		FDFR↑	ISM↓	BRISQUE↑	SER-FIQ↓
v2.1	v2.1	87.67	0.16	42.44	0.11
	v1.4	21.67	0.16	20.70	0.39
	v1.5	16.33	0.14	15.89	0.50
Train	**Test**	**"a photo of sks person looking at the mirror"**			
		FDFR↑	ISM↓	BRISQUE↑	SER-FIQ↓
v2.1	v2.1	94.11	0.09	45.93	0.05
	v1.4	79.44	0.11	40.68	0.07
	v1.5	69.00	0.12	34.67	0.12
Train	**Test**	**"a photo of sks person in front of eiffel tower"**			
		FDFR↑	ISM↓	BRISQUE↑	SER-FIQ↓
v2.1	v2.1	77.00	0.07	41.15	0.06
	v1.4	38.78	0.07	14.23	0.20
	v1.5	43.33	0.06	13.82	0.15

in Table IX. For all methods, the memory and time costs are tested on one RTX 4090 and report their average values. Note that the costs for MetaCloak are taken from the official data provided in its paper.

Runtime. GAP-Diff significantly reduces the time required to generate a protected image by optimizing the process through neural network training. This allows for faster deployment in real-time applications where speed is crucial.

Memory usage. The efficient use of memory resources ensures that GAP-Diff can be scaled across various hardware configurations, making it versatile for different deployment scenarios.

TABLE IX: Comparison of the runtime and memory usage for generating protected images with GAP-Diff and other baselines.

Method	Runtime (second)	Memory Usage (GB)
Photoguard [36]	194	16
Glaze [38]	118	4
Mist [24]	294	5
Anti-DB [45]	308	18
ACE [53]	175	6
MetaCloak [26]	1.1×10^4	504
CAAT [49]	98	18
SimAC [47]	1.2×10^3	33
GAP-Diff (ours)	**0.04**	**2**

VI. Discussion

Our proposed GAP-Diff framework demonstrates significant improvements in protecting personal images from unauthorized customization by text-to-image diffusion models, particularly in scenarios involving JPEG compression. In this section, we discuss the threats to validity and limitations, practical deployment considerations, and ethical implications of our work.

A. Threats to Validity and Limtations

The effectiveness of our protection mechanism is bounded by a noise budget of $16/255$, as shown in Figure 15 and Table IV. Values lower than this threshold may lead to the leakage of more facial information, while exceeding this threshold can compromise the utility of image generation, creating a trade-off between protection strength and image quality. One reason for this limitation may be that we have not yet found the optimal strategy for training the neural network, which prevents us from achieving the ideal balance between protection and visual quality.

Further, we observed that the fine-tuned T2I-DM based on protected images may still generate relatively clear images for certain prompts. We believe these images are a result of the inherent randomness in the diffusion model, leading to deviations from the original identity and its adversarial distribution. Even so, we still consider the protection successful, as the generated identity differs from the original.

Moreover, the current implementation focuses on the most classic and popular DM structures, but novel model architectures incorporating transformers may need to be considered for future improvements.

B. Practical Deployment

GAP-Diff can be deployed via a server API, allowing users to generate protected images before sharing or sending them online. This deployment strategy ensures that the protection is applied consistently and reduces the risk of user error. It also allows for potential integration with existing social media platforms or image sharing services, which could significantly enhance user privacy protection at scale. Future work could explore optimizing the performance of GAP-Diff for real-time applications and developing user-friendly interfaces to help individuals understand and control the level of protection applied to their images.

C. Ethic Considerations

Our research aims to enhance privacy as a fundamental human right. We strictly adhered to all ethical requirements during our experiments and did not engage in any malicious activities, such as disrupting legitimate services or causing financial or reputational harm to individuals whose faces appeared in the publicly available datasets we used.

This work proposes a protective approach against malicious image customization for images shared on social networks or communication applications. To illustrate the effectiveness of GAP-Diff, we present real human faces from CelebA and VGGFace2—two well-known publicly available datasets—which have been identified as potential ethical concerns by some members of the community.

We believe there are two directions to address these concerns. First, we could preprocess the raw images by obscuring certain body parts, such as the eyes, and then conduct our experiments. Second, we could collect real human face images with explicit consent and use these images for our experiments. Given our limited resources for the second option, we plan to pursue the first approach. Figure 19 in the attached document shows additional experimental results from the rebuttal period, which, while not identical to the original results, still support our main

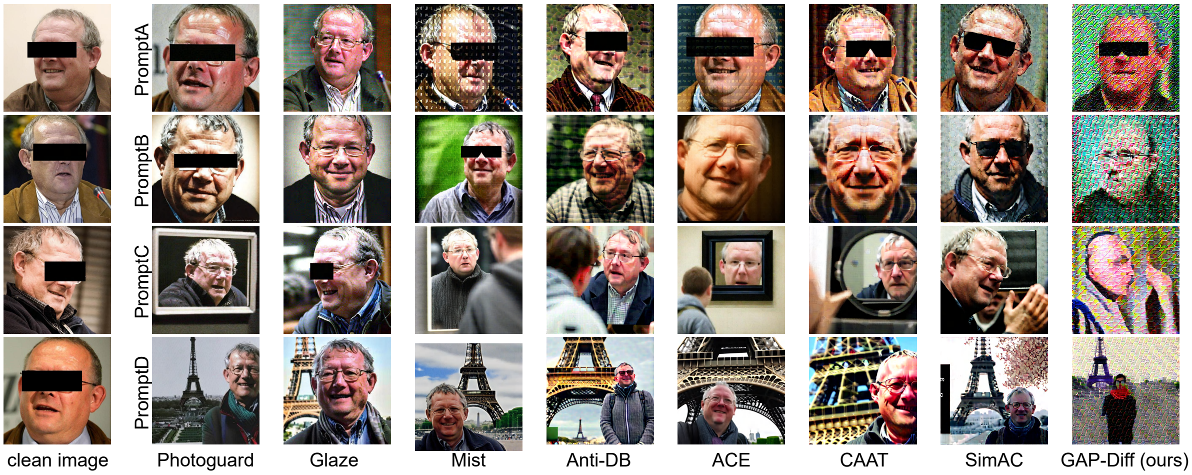

Fig. 19: Qualitative results of obscuring critical facial information on VGGFace.

conclusion: GAP-Diff outperforms existing methods under JPEG compression. Note that as seen in Figure 19, it is very likely that the generator may remove the obscuring blocks when producing the images.

VII. Conclusion

To mitigate the degradation of existing works in protection against FT-T2I-DMs customization caused by JPEG compression, we propose GAP-Diff, which can protect images from customization by adding small yet robust noise. Through the design of the proposed three modules and the optimization loss, it can learn robust representations against JPEG compression by backpropagating gradient information while learning adversarial characteristics for disrupting FT-T2I-DMs. Extensive experiments show that, compared to all state-of-the-art methods, GAP-Diff provides better facial protection and higher generation efficiency in digital world. In the future, we will explore automatically adjusting training strategies to achieve stronger protective noise and further address challenges posed by increasingly powerful customization approaches.

Acknowledgment

Haotian Zhu and Shuchao Pang are supported by the National Key R&D Program of China (No.2023YFB2703900) and the National Natural Science Foundation of China (No.62206128). Yongbin Zhou is supported by the National Natural Science Foundation of China (No.U2336205). Shuchao Pang is the corresponding author.

References

[1] Chaitali Bhattacharyya, Hanxiao Wang, Feng Zhang, Sungho Kim, and Xiatian Zhu. Diffusion deepfake. *arXiv preprint arXiv:2404.01579*, 2024.

[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] Qiong Cao, Li Shen, Weidi Xie, Omkar M Parkhi, and Andrew Zisserman. Vggface2: A dataset for recognising faces across pose and age. In *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, pages 67–74. IEEE, 2018.

[4] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (sp)*, pages 39–57. Ieee, 2017.

[5] Yunzhuo Chen, Nur Al Hasan Haldar, Naveed Akhtar, and Ajmal Mian. Text-image guided diffusion model for generating deepfake celebrity interactions. In *2023 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, pages 348–355. IEEE, 2023.

[6] Jiankang Deng, Jia Guo, Evangelos Ververas, Irene Kotsia, and Stefanos Zafeiriou. Retinaface: Single-shot multi-level face localisation in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5203–5212, 2020.

[7] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4690–4699, 2019.

[8] Clare Duffy. Puffer coat pope. musk on a date with gm ceo. fake ai 'news' images are fooling social media users, 2023. Available online at: https://edition.cnn.com/2023/04/02/tech/ai-generated-images-social-media/index.html, accessed: 11.07.2024.

[9] Gemma Dunstan. Ai: Fears hundreds of children globally used in naked images, 2023. Available online at: https://www.bbc.com/news/uk-wales-67344916.amp, accessed: 11.07.2024.

[10] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion. *arXiv preprint arXiv:2208.01618*, 2022.

[11] Sahra Ghalebikesabi, Leonard Berrada, Sven Gowal, Ira Ktena, Robert Stanforth, Jamie Hayes, Soham De, Samuel L Smith, Olivia Wiles, and Borja Balle. Differentially private diffusion models generate useful synthetic images. *arXiv preprint arXiv:2302.13861*, 2023.

[12] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in Neural Information Processing Systems*, 27, 2014.

[13] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. *Advances in Neural Information Processing Systems*, 30, 2017.

[14] Alexandros Haliassos, Rodrigo Mira, Stavros Petridis, and Maja Pantic. Leveraging real talking faces via self-supervision for robust forgery detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14950–14962, 2022.

[15] Ligong Han, Yinxiao Li, Han Zhang, Peyman Milanfar, Dimitris Metaxas, and Feng Yang. Svdiff: Compact parameter space for diffusion fine-tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7323–7334, 2023.

[16] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

[17] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

[18] 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.

[19] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

[20] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019.

[21] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8110–8119, 2020.

[22] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[23] Nupur Kumari, Bingliang Zhang, Richard Zhang, Eli Shechtman, and Jun-Yan Zhu. Multi-concept customization of text-to-image diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1931–1941, 2023.

[24] Chumeng Liang and Xiaoyu Wu. Mist: Towards improved adversarial examples for diffusion models. *arXiv preprint arXiv:2305.12683*, 2023.

[25] Chumeng Liang, Xiaoyu Wu, Yang Hua, Jiaru Zhang, Yiming Xue, Tao Song, Zhengui Xue, Ruhui Ma, and Haibing Guan. Adversarial example does good: Preventing painting imitation from diffusion models via adversarial examples. *arXiv preprint arXiv:2302.04578*, 2023.

[26] Yixin Liu, Chenrui Fan, Yutong Dai, Xun Chen, Pan Zhou, and Lichao Sun. Metacloak: Preventing unauthorized subject-driven text-to-image diffusion-based synthesis via meta-learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24219–24228, 2024.

[27] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[28] Anish Mittal, Anush Krishna Moorthy, and Alan Conrad Bovik. No-reference image quality assessment in the spatial domain. *IEEE Transactions on Image Processing*, 21(12):4695–4708, 2012.

[29] Jianxia Ning, Indrajeet Singh, Harsha V Madhyastha, Srikanth V Krishnamurthy, Guohong Cao, and Prasant Mohapatra. Secret message sharing using online social media. In *2014 IEEE Conference on Communications and Network Security*, pages 319–327. IEEE, 2014.

[30] Omid Poursaeed, Isay Katsman, Bicheng Gao, and Serge Belongie. Generative adversarial perturbations. In *Proceedings of the IEEE Conference on Computer vision and Pattern Recognition*, pages 4422–4431, 2018.

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

[32] 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.

[33] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015.

[34] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22500–22510, 2023.

[35] Simo Ryu. Low-rank adaptation for fast text-to-image diffusion fine-tuning, 2023. Available online at: https://github.com/cloneofsimo/lora, accessed: 11.07.2024.

[36] Hadi Salman, Alaa Khaddaj, Guillaume Leclerc, Andrew Ilyas, and Aleksander Madry. Raising the cost of malicious ai-powered image editing. *arXiv preprint arXiv:2302.06588*, 2023.

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

[38] Shawn Shan, Jenna Cryan, Emily Wenger, Haitao Zheng, Rana Hanocka, and Ben Y Zhao. Glaze: Protecting artists from style mimicry by {Text-to-Image} models. In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 2187–2204, 2023.

[39] Shawn Shan, Emily Wenger, Jiayun Zhang, Huiying Li, Haitao Zheng, and Ben Y Zhao. Fawkes: Protecting privacy against unauthorized deep learning models. In *29th USENIX security symposium (USENIX Security 20)*, pages 1589–1604, 2020.

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

[41] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

[42] Marianna Spring. Trump supporters target black voters with faked ai images, 2024. Available online at: https://www.bbc.com/news/world-us-canada-68440150?zephr-modal-register, accessed: 11.07.2024.

[43] Weiwei Sun, Jiantao Zhou, Ran Lyu, and Shuyuan Zhu. Processing-aware privacy-preserving photo sharing over online social networks. In *Proceedings of the 24th ACM International Conference on Multimedia*, pages 581–585, 2016.

[44] Philipp Terhorst, Jan Niklas Kolf, Naser Damer, Florian Kirchbuchner, and Arjan Kuijper. Ser-fiq: Unsupervised estimation of face image quality based on stochastic embedding robustness. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5651–5660, 2020.

[45] Thanh Van Le, Hao Phung, Thuan Hoang Nguyen, Quan Dao, Ngoc N Tran, and Anh Tran. Anti-dreambooth: Protecting users from personalized text-to-image synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2116–2127, 2023.

[46] Gregory K Wallace. The jpeg still picture compression standard. *Communications of the ACM*, 34(4):30–44, 1991.

[47] Feifei Wang, Zhentao Tan, Tianyi Wei, Yue Wu, and Qidong Huang. Simac: A simple anti-customization method for protecting face privacy against text-to-image synthesis of diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12047–12056, 2024.

[48] Junke Wang, Zuxuan Wu, Wenhao Ouyang, Xintong Han, Jingjing Chen, Yu-Gang Jiang, and Ser-Nam Li. M2tr: Multi-modal multi-scale transformers for deepfake detection. In *Proceedings of the 2022 International Conference on Multimedia Retrieval*, pages 615–623, 2022.

[49] Jingyao Xu, Yuetong Lu, Yandong Li, Siyang Lu, Dongdong Wang, and Xiang Wei. Perturbing attention gives you more bang for the buck: Subtle imaging perturbations that efficiently fool customized diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24534–24543, 2024.

[50] Zhiyuan Yan, Yong Zhang, Xinhang Yuan, Siwei Lyu, and Baoyuan Wu. Deepfakebench: A comprehensive benchmark of deepfake detection. *arXiv preprint arXiv:2307.01426*, 2023.

[51] Chaofei Yang, Leah Ding, Yiran Chen, and Hai Li. Defending against gan-based deepfake attacks via transformation-aware adversarial faces. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2021.

[52] Chin-Yuan Yeh, Hsi-Wen Chen, Shang-Lun Tsai, and Sheng-De Wang. Disrupting image-translation-based deepfake algorithms with adversarial attacks. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision Workshops*, pages 53–62, 2020.

[53] Boyang Zheng, Chumeng Liang, Xiaoyu Wu, and Yan Liu. Understanding and improving adversarial attacks on latent diffusion model. *arXiv preprint arXiv:2310.04687*, 2023.

[54] Yinglin Zheng, Jianmin Bao, Dong Chen, Ming Zeng, and Fang Wen. Exploring temporal coherence for more general video face forgery detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15044–15054, 2021.

[55] Jiren Zhu, Russell Kaplan, Justin Johnson, and Li Fei-Fei. Hidden: Hiding data with deep networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 657–672, 2018.

Appendix A
Artifact Appendix

Our paper proposed GAP-Diff, which is a counter-customization framework with its generator trained on large facial data. In this section, we provide a complete description on the artifacts related to the training and testing phases. Due to the lengthy training and testing times inherent to diffusion models, alongside the temporal constraints of the Artifact Evaluation (AE) process, we will supply pre-trained weights for testing. Further, the test process will be a scaled-down version (in line with the temporal constraints of the AE), meaning that we provide a selection of scenarios and sample inputs, which may introduce randomness compared to the quantitative results of testing on the extensive dataset in our paper. Thus, we aim for our artifacts to conform to the expectations set by our paper regarding functionality, usability, and relevance.

A. Description & Requirements

1) How to access: The artifacts are publicly available on `https://github.com/AIASLab/GAP-Diff` and `https://doi.org/10.5281/zenodo.14249397`.

2) Hardware dependencies: All tasks can be completed on a single NVIDIA A800 80G GPU. Given the time constraints, we provide pre-trained model weights, and all testing processes can be conducted on our rented cloud server equipped with an NVIDIA RTX 4090 24G GPU and 16 vCPU Intel(R) Xeon(R) Gold 6430.

3) Software dependencies: A recent Linux operating system Ubuntu 22.04 with Anaconda (or miniconda3) and CUDA 11.8. We give the configurations to the implementation language and its dependencies in the `README` of our source code.

4) Benchmarks: We select four existing works that performed well in our paper as benchmarks and provide their implementations or protected results in our artifacts, taking into account hardware limitations and time. Specific details will be discussed in Section A-E.

B. Artifact Installation & Configuration

We first require our code to be placed in the correct Linux user directory. As our testing and evaluation require different environments, please refer to the `README` file for detailed specifications on creating these environments within Conda. Each environment may require specific pip installations, which are also outlined in the respective files. Further, relevant model weights and datasets must be properly placed in the corresponding directories as indicated in the `README` file. On the provided rented cloud server, we have already set up the required virtual environments, weight files, and datasets, allowing AEC to proceed directly with the evaluation.

C. Experiment Workflow

Our materials include three independent evaluation stages. The first stage verifies the facial protection efficacy of GAP-Diff using pre-trained weights. The second stage assesses the ability of protective noise of GAP-Diff and benchmarks to resist JPEG compression. The third stage examines the facial protection performance of GAP-Diff under conditions mentioned in the paper's Adverse Settings, such as prompt mismatch, different fine-tuning methods, and different preprocessing methods.

Due to time constraints, we recommend conducting evaluations in the order outlined above, with selective experiments in the second and third evaluation stage as indicated in the `README`. All experiments can be run using the provided bash scripts such as `generate.sh`; please be patient and wait for the completion of each `sh` file before proceeding to the next evaluation operation.

D. Major Claims

- (C1): GAP-Diff demonstrates effective facial protection using pre-trained weights. This is verified by experiment (E1), whose results are reported in Figure 12 under $Q = 100$.
- (C2): The protective noise generated by GAP-Diff shows superior resistance to JPEG compression compared to benchmark methods. This is proven by experiment (E2), with results illustrated in Table I and Figure 2 and Figure 8. Further, during this experiment, the resistance to different JPEG quality of GAP-Diff can also be evaluated with results illustrated in Figure 12 and Figure 13.
- (C3): GAP-Diff maintains robust facial protection performance under adverse settings, including prompt mismatch, different fine-tuning methods, and different preprocessing techniques. This is demonstrated by experiment (E3), with results presented in Figure 17 and Tables V, VI, VII.

E. Evaluation

In total, all experiments require approximately 21 compute-hours and about 25 human-minutes. We assume that the code and environment have already been fully deployed (as in the provided cloud server). The following experimental steps can be completed to carry out all artifact evaluation tasks. All experiments are conducted using a randomly selected dataset of five identities.

1) Experiment (E1) - Claim (C1): [around 2 human-minutes + about 1.5 compute-hours].

[How to] Run the GAP-Diff generator with pre-trained weights to obtain the protected images, then use these images in DreamBooth training to obtain the Fine-tuned Text-to-Image Diffusion Model (FT-T2I-DM) and infer the customized outcomes. Finally, evaluate these outcomes for their protection effect.

[Preparation] In a new shell, go to the `GAP-Diff` folder.

[Execution] Run the GAP-Diff generator with pre-trained weights to obtain the protected images.

```
$ conda activate gap-diff
$ bash scripts/generate.sh
# Expected output:
# - Command line output "the images are saved" for each
    identity
# - The protected images are saved to GAP-Diff/
    protected_images/gap-diff-per16
```

Use the protected images in DreamBooth training to obtain the FT-T2I-DM and infer the customized outcomes with 4 prompts.

```
$ bash scripts/db_infer.sh
# Expected output:
# - Command line output "Finish training" and infer log
     for each identity
# - The customized outcomes are saved to GAP-Diff/infer/
     gap_diff_per16
```

Evaluate the customized outcomes for the protective effect of GAP-Diff.

```
$ conda activate fdfr_ism
$ bash scripts/evaluate_fdfr_ism.sh
# Expected output:
# - Command line output the mean FDFR and ISM of all
     identities for each prompt in the customized
     outcomes
$ conda activate serfiq
$ bash scripts/evaluate_brisque_serfiq.sh
# Expected output:
# - Command line output the mean BRISQUE and SER-FIQ of
     all identities for each prompt in the customized
     outcomes
```

[Results] Four quantitative metrics for GAP-Diff protection. Notice that due to the use of sample inputs combined with the inherent randomness of the diffusion model, the quantitative results may slightly differ from the data reported in our original paper, but high FDFR, low ISM, high BRISQUE and low SER-FIQ can still demonstrate the effective protection of GAP-Diff.

2) Experiment (E2) - Claim (C2): [around 10 human-minutes + 7.5∼10.5 compute-hours].

[How to] Apply JPEG compression with a quality of $Q = 70$ to the protected images from Experiment (E1) and the new protected images generated using benchmark methods. Obtain the FT-T2I-DM using protected images and infer the customized outcomes. Finally, evaluate the outcomes for their protection effect. Similar experiments for GAP-Diff can also be conducted with JPEG compression quality of $Q = 50$ and $Q = 90$.

[Preparation] In the shell, go to the `GAP-Diff` folder.

[Execution] Apply JPEG compression with a quality of $Q = 70$ to the protected images from Experiment (E1).

```
$ conda activate gap-diff
$ bash scripts/preprocess/jpeg.sh
# Expected output:
# - Command line output "Image compression completed
     successfully!"
# - The JPEG compressed protected images are saved to
     GAP-Diff/infer/gap_diff_per16_jpeg70
```

Modify the corresponding file paths according to the `README`, and follow the same process as in Experiment (E1) for customization, inference, and evaluation.

Due to time and memory constraints, we provide implementations for Anti-DB and CAAT in benchmarks, while for SimAC and MetaCloak, we only supply the protected images. Please follow the steps outlined below for evaluation.

```
$ conda activate gap-diff
$ bash benchmark/scripts/antidb.sh
$ conda activate CAAT
$ bash benchmark/scripts/caat.sh
$ conda activate gap-diff
$ bash benchmark/scripts/jpeg.sh
$ bash benchmark/scripts/db_infer.sh
$ conda activate fdfr_ism
$ bash benchmark/scripts/evaluate_fdfr_ism.sh
$ conda activate serfiq
$ bash benchmark/scripts/evaluate_brisque_serfiq.sh
# Expected output:
# - The protected images and JPEG compressed ones are
     saved to GAP-Diff/benchmark/protected_images
# - The customized outcomes are saved to GAP-Diff/
     benchmark/infer
# - Command line output the four quantitative results
```

Additionally, run the `jpeg.sh` following the instructions of `README` with corresponding parameter Q to obtain GAP-Diff protected images at different compression qualities, and follow the same process for customization, inference, and evaluation as previously outlined.

[Results] The sample inputs may still lead to a slightly different quantitative results. However, these results are expected to be better than the benchmarks and still demonstrate that GAP-Diff has greater resistance to JPEG compression. Qualitative results can be obtained by downloading via `sftp` as guided in `README`.

3) Experiment (E3) - Claim (C3): [around 10 human-minutes + around 9 compute-hours].

[How to] Conduct experiments on prompt mismatch, different fine-tuning methods, and different preprocessing techniques using different script files and commands.

[Preparation] In the shell, go to the `GAP-Diff` folder.

[Execution] Train DreamBooth on prompt mismatch, and infer customized outcomes.

```
$ conda activate gap-diff
$ bash scripts/db_infer_prompt_mismatch.sh
$ conda activate fdfr_ism
$ bash benchmark/scripts/ex/evaluate_fdfr_ism_ex.sh
$ conda activate serfiq
$ bash benchmark/scripts/ex/evaluate_brisque_serfiq_ex.
     sh
# Expected output:
# - The customized outcomes are saved to GAP-Diff/infer/
     gap_diff_per16_ex
# - Command line output the four quantitative results
```

Run different fine-tuning methods (here SVDiff).

```
$ cd /root/svdiff-pytorch/
$ conda activate svdiff
$ bash scripts/svd.sh
# Expected output:
# - The customized outcomes are saved to GAP-Diff/infer/
     svdiff
```

Modify the paths in the script files and evaluate the customized outcomes in the same manner as in Experiment (E1).

Run different preprocessing techniques on protected images.

```
$ conda activate gap-diff
$ cd /root/gap-diff/
$ bash scripts/preprocess/other_preprocess.sh
# Expected output:
# - Command line output successful preprocessing message
# - The preprocessed images are save to GAP-Diff/
     protected_images
```

Select one of the preprocessing methods described in `README`, modify the paths in the script files and evaluate the customized outcomes in the same manner as in Experiment (E1) according to `README`.

[Results] Quantitative results which may still be different from our original paper but can also demonstrate GAP-Diff's protective effects under various adverse settings. To verify visual observations in the paper, you can also download the corresponding files to obtain qualitative results.