TAFIM: Targeted Adversarial Attacks against Facial Image Manipulations

Shivangi Aneja[1], Lev Markhasin[2], and Matthias Nießner[1]

[1] Technical University of Munich, Germany
[2] Sony Europe RDC Stuttgart, Germany

Fig. 1. Left: We propose a novel approach to protect facial images from several image manipulation models simultaneously. We leverage neural network to encode the generation of quasi-imperceptible perturbations for different manipulation models and fuse them together using attention mechanism to generate manipulation-agnostic perturbation. This perturbation, when added to the real image, forces the face manipulation models to produce a predefined manipulation target as output (white/blue image in this case). This is several orders of magnitude faster and can also be used for real-time applications. **Right:** Without any protection applied, manipulation models can be misused to generate fake images for malicious activities.

Abstract. Face manipulation methods can be misused to affect an individual's privacy or to spread disinformation. To this end, we introduce a novel data-driven approach that produces image-specific perturbations which are embedded in the original images. The key idea is that these protected images prevent face manipulation by causing the manipulation model to produce a predefined manipulation target (uniformly colored output image in our case) instead of the actual manipulation. In addition, we propose to leverage differentiable compression approximation, hence making generated perturbations robust to common image compression. In order to prevent against multiple manipulation methods simultaneously, we further propose a novel attention-based fusion of manipulation-specific perturbations. Compared to traditional adversarial attacks that optimize noise patterns for each image individually, our generalized model only needs a single forward pass, thus running orders of magnitude faster and allowing for easy integration in image processing stacks, even on resource-constrained devices like smartphones [1].

[1] Project Page: https://shivangi-aneja.github.io/projects/tafim

1 Introduction

The spread of disinformation on social media has raised significant public attention in the recent few years, due to its implications on democratic processes and society in general. The emergence and constant improvement of generative models, and in particular face image manipulation methods, has signaled a new possible escalation of this problem. For instance, face-swapping methods [8, 37] whose models are publicly accessible can be misused to generate non-consensual synthetic imagery. Other examples include face attribute manipulation methods [9, 10, 39, 41] that change the appearance of real photos, thus generating fake images that might then be used for criminal activities [1]. Although a variety of manipulation tools have been open-sourced, surprisingly only a handful of methods have achieved widespread applicability among users (for details see the supplemental material). One reason is that re-training these methods is not only compute intensive but they also require specialized knowledge and skill sets for training. As a result, most end users only apply easily accessible pre-trained models of a few popular methods. In this work, we exploit these popular manipulation methods and their models which are known in advance and propose targeted adversarial attacks to protect against facial image manipulations.

As powerful face image manipulation tools became easier to use and more widely available, many efforts to detect image manipulations were initiated by the research community [13]. This has led to the task of automatically detecting manipulations as a classification task where predictions indicate whether a given image is real or fake. Several learning-based approaches [2, 4, 6, 11, 12, 27, 28, 36, 43, 56, 62] have shown promising results in identifying manipulated images. Despite the success and high classification accuracies of these methods, they can only be helpful if they are actually being used by the end-user. However, manipulated images typically spread in private groups or on social media sites where manipulation detection is rarely available.

An alternative avenue to detecting manipulations is to prevent manipulations from happening in the first place by disrupting potential manipulation methods [19, 44, 57–59]. Here, the idea is to disrupt generative neural network models with low-level noise patterns, similar to the ideas of adversarial attacks used in the context of classification tasks [18, 48]. Methods optimizing noise patterns for every image from scratch [44, 57–59] require several seconds to process a single image. In practice, this slow run time largely prohibits their use on mobile devices (e.g., as part of the camera stack). At the same time, these manipulation prevention methods aim to either disrupt [19, 29, 44, 47] or nullify [58, 59] the results of image manipulation models, which makes it difficult to identify which face manipulation technique was used.

To address these challenges, we propose a targeted adversarial attack against face image manipulation methods. More specifically, we introduce a data-driven approach that generates quasi-imperceptible perturbations specific to a given image. Our objective is that when an image manipulation is attempted, a predefined manipulation target is generated as output instead of the originally intended manipulation. In contrast to previous optimization-based approaches, our

perturbations are generated by a generalizable conditional model requiring only a few milliseconds for generation. We additionally incorporate a differentiable compression module during training, to achieve robustness against common image processing pipelines. Finally, to handle multiple manipulation models simultaneously, we propose a novel attention-based fusion mechanism to combine model-specific perturbations. In summary, the contributions in the paper are:

- A data-driven approach to synthesize image-specific perturbations that outputs a predefined manipulation target (depending on the manipulation model used), instead of per-image optimization; this is not only significantly faster but also outperforms existing methods in terms of image-to-noise quality.
- Incorporation of differentiable compression during training to achieve robustness to common image processing pipelines.
- An attention-based fusion and refinement of model-specific perturbations to prevent against multiple manipulation models simultaneously.

2 Related Work

Image Manipulation. Recent advances in image synthesis models have made it possible to generate detailed and expressive human faces [7, 20–23, 38, 51] which might be used for unethical activities/frauds. Even more problematic can be the misuse of real face images to synthesize new ones. For instance, face-attribute modification techniques [9, 10, 39, 41] and face-swapping models [8, 37] facilitate the manipulation of existing face images. Similarly, facial re-enactment tools [17, 24, 46, 50, 60] also use real images/videos to synthesize fake videos.

Facial Manipulation Detection. The increasing availability of these image manipulation models calls for the need to reliably detect synthetic images in an automated fashion. Traditional facial manipulation detection leverages handcrafted features such as gradients or compression artifacts, in order to find inconsistencies in an image [3, 15, 31]. While such self-consistency can produce good results, these methods are less accurate than more recent learning-based techniques [5, 6, 11, 12, 43], which are able to detect fake imagery with a high degree of confidence. In contrast to detecting forgeries, we aim to prevent manipulations from happening in the first place by rendering the respective manipulation models ineffective by introducing targeted adversarial attacks.

Adversarial Attacks. Adversarial attacks were initially introduced in the context of classification tasks [14, 18, 35, 48] and eventually expanded to semantic segmentation and detection models [16, 40, 55]. The key idea behind these methods is to make imperceptible changes to an image in order to disrupt the feature extraction of the underlying neural networks. While these methods have achieved great success in fooling state-of-the-art vision models, one significant drawback is that optimizing a pattern for every image individually makes the optimization process quite slow. In order to address this challenge, generic universal image-agnostic noise patterns were introduced [33, 34]. This has shown to be effective for misclassification tasks but gives suboptimal results for generative models, as we show in Sec. 4.

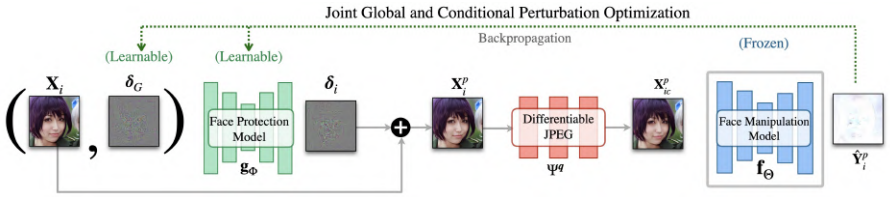

Fig. 2. We first pass the real image $\mathbf{X}_i$ and the global perturbation $\boldsymbol{\delta}_G$ through the face protection model $\mathbf{g}_{\boldsymbol{\Phi}}$ to generate the image-specific perturbation $\boldsymbol{\delta}_i$. This perturbation is then added to the original image to create the protected image $\mathbf{X}_i^p$. The protected image is then compressed using the differentiable JPEG $\boldsymbol{\Psi}^{\boldsymbol{q}}$ that generates compressed protected image $\mathbf{X}_{ic}^p$, which is passed through face manipulation model $\mathbf{f}_{\boldsymbol{\Theta}}$ to generate the manipulated output $\hat{\mathbf{Y}}_i^p$. The output of the face manipulation model is then used to drive the optimization.

Manipulation Prevention. Deep steganography and watermarking techniques [30, 49, 53, 54, 57, 63] can be used to embed an image-specific watermark to secure an image. For instance, FaceGuard [57] embeds a binary vector to the original image representative of a person's identity and classifies whether the image is fake by checking if the watermark is intact after being used for face manipulation tasks. These methods, however, cannot prevent the manipulation of face images which is the key focus of our work.

Recent works that aim to prevent image manipulations exploit adversarial attack techniques to break image manipulation models. Ruiz et al. [44] disrupt the output of deepfake generation models. Yeh et al. [58, 59] aim to nullify the effect of image manipulation models. Other approaches [29, 47] aim to disturb the output of face detection and landmark extraction steps, which are usually used as pre-processing by deepfake generation methods. One commonality of these methods is that they optimize a pattern for each image separately which is computationally very expensive, thus having limited applicability for real-world applications like resource-constrained devices. Very recently, Huang et al [19] proposed a neural network based approach to generate image-specific patterns for low-resolution images, however, they do not consider compression, which is a common practical scenario that can make these generated patterns ineffective (as shown in Sec. 4). Additionally, this method only considers a single manipulation model at a time, thus limiting its applicability to protect against multiple manipulations simultaneously. To this end, we propose (a) a novel data-driven method to generate image-specific perturbations which are robust to compression and (b) fusion of manipulation-specific perturbations. Our method not only require less computational effort compared to existing adversarial attacks works, but can protect against multiple manipulation methods simultaneously.

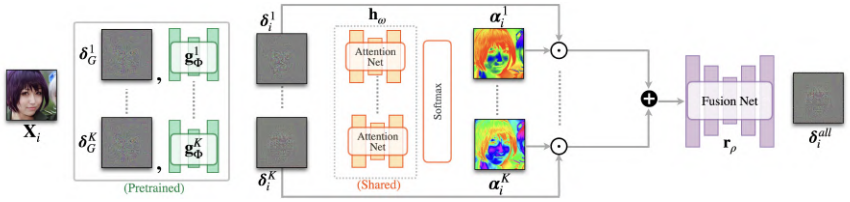

Fig. 3. For a given RGB image $\mathbf{X}_i$, we first use the pre-trained manipulation-specific global noise and protection models $\{\boldsymbol{\delta}_G^k, \mathbf{g}_{\boldsymbol{\Phi}}^k\}_{k=1}^K$ to generate manipulation-specific perturbations $\{\boldsymbol{\delta}_i^k\}_{k=1}^K$, which are passed into a shared attention backbone $\boldsymbol{h_\omega}$ to generate the spatial attention maps $\{\boldsymbol{\alpha}_i^k\}_{k=1}^K$. These attention maps are then combined with manipulation-specific $\{\boldsymbol{\delta}_i^k\}_{k=1}^K$ using channel-wise hadamard product and blended together using addition operation. Finally, the blended perturbation is then refined using FusionNet $\boldsymbol{r_\rho}$ to generate manipulation-agnostic perturbation $\boldsymbol{\delta}_i^{\text{all}}$.

3 Proposed Method

Our goal is to prevent face image manipulations and simultaneously identify which model was used for the manipulation. That is, for a given face image, we aim to find an imperceptible perturbation that disrupts the generative neural network of a manipulation method such that a solid color image is produced as output instead of originally-intended manipulation. Algorithmically, this is a targeted adversarial attack where the predefined manipulation targets make it easy for a human to identify the used manipulation method.

3.1 Method Overview

We consider a setting where we are given K manipulation models $\mathcal{M} = \{\mathbf{f}_{\boldsymbol{\Theta}}^k\}_{k=1}^K$ where $\mathbf{f}_{\boldsymbol{\Theta}}^k$ denotes the k-th manipulation model. For a given RGB image $\mathbf{X}_i \in \mathbb{R}^{H\times W\times 3}$ of height H and width W, the goal is to find the optimal perturbation $\boldsymbol{\delta}_i \in \mathbb{R}^{H\times W\times 3}$ that is embedded in the original image $\mathbf{X}_i$ to produce a valid protected image $\mathbf{X}_i^p \in \mathbb{R}^{H\times W\times 3}$. The manipulation model $\mathbf{f}_{\boldsymbol{\Theta}}^k$, which is parametrized by its neural network weights $\boldsymbol{\Theta}$, is also given as input to the method. Note that we use $\mathbf{f}_{\boldsymbol{\Theta}}^k$ only to drive the perturbation optimization and do not alter its weights. For the given image $\mathbf{X}_i$, the output synthesized by the manipulation model $\mathbf{f}_{\boldsymbol{\Theta}}^k$ is denoted as $\hat{\mathbf{Y}}_{ik} \in \mathbb{R}^{H\times W\times 3}$. We define the uniformly-colored predefined manipulation targets for the K manipulation models as $\mathcal{Y} = \{\mathbf{Y}_k^{\text{target}}\}_{k=1}^K$.

In order to protect face images and obtain image perturbations, we propose two main ideas: First, for a given manipulation model $\mathbf{f}_{\boldsymbol{\Theta}}$, we jointly optimize for a global perturbation pattern $\boldsymbol{\delta}_G \in \mathbb{R}^{H\times W\times 3}$ and a generative neural network $\mathbf{g}_{\boldsymbol{\Phi}}$ (parameterized by its weights $\boldsymbol{\Phi}$) to produce image-specific perturbations $\boldsymbol{\delta}_i$. The global pattern $\boldsymbol{\delta}_G$ is generalized across the entire data distribution. The generative model $\boldsymbol{g_\Phi}$ is conditioned on the global perturbation $\boldsymbol{\delta}_G$ as well as the real image $\mathbf{X}_i$. Our intuition is that the global perturbation provides a

strong prior for the global noise structure, thus enabling the conditional model to produce more effective perturbations. We also incorporate a differentiable JPEG module to ensure the robustness of the perturbations towards compression. This is shown in Fig. 2.

Second, to handle multiple manipulation models simultaneously, we leverage an attention network $\boldsymbol{h_\omega}$ (parametrized by $\boldsymbol{\omega}$) to first generate attention maps $\{\alpha_k\}_{k=1}^K$ for the K manipulation methods, which are then used to refine the model-specific perturbations $\{\boldsymbol{\delta}_i^k\}_{k=1}^K$ with an encoder-decoder network denoted as FusionNet $\boldsymbol{r_\rho}$ (parametrized by $\boldsymbol{\rho}$) to generate a single final perturbation $\boldsymbol{\delta}_i^{\text{all}}$ for the given image $\mathbf{X}_i$. $\boldsymbol{\delta}_i^{\text{all}}$ can protect the image from the given K manipulation methods simultaneously. An overview is shown in Fig. 3.

3.2 Methodology

We define an optimization strategy where the objective is to find the smallest possible perturbation that achieves the largest disruption in the output manipulation; i.e., where the generated output for the k-th manipulation model is closest to its predefined target image $\mathbf{Y}_k^{\text{target}}$. This is explained in detail below.

Joint Global and Conditional Generative Model Optimization The global perturbation $\boldsymbol{\delta}_G \in \mathbb{R}^{H\times W\times 3}$ is a fixed image-agnostic perturbation shared across the data distribution. The conditional generative neural network model $\boldsymbol{g_\Phi}$ is a UNet [42] based encoder-decoder architecture. For a given manipulation method, we jointly optimize global perturbation $\boldsymbol{\delta}_G$ and the parameters $\boldsymbol{\Phi}$ of this conditional model $\boldsymbol{g_\Phi}$ together in order to generate image-specific perturbations.

$$\boldsymbol{\delta}_G^*, \boldsymbol{\Phi^*} = \underset{\delta_G, \Phi}{\text{argmin}} \quad \mathcal{L}_k \tag{1}$$

where $\mathcal{L}_k$ refers to overall loss (Eq. 2).

$$\mathcal{L}_k = \left[\sum_{i=1}^{N} \mathcal{L}_i^{\text{recon}} + \lambda \mathcal{L}_i^{\text{perturb}}\right]_k, \tag{2}$$

where the parameter λ regularizes the strength of perturbation added to the real image, N denotes the number of images in the dataset, i denotes the image index and k denotes the manipulation method. $\mathcal{L}_i^{\text{recon}}$ and $\mathcal{L}_i^{\text{perturb}}$ represent reconstruction and perturbation losses for i-th image. The model $\boldsymbol{g_\Phi}$ is conditioned on the globally-optimized perturbation $\boldsymbol{\delta_G}$ as well as the original input image $\mathbf{X}_i$. Conditioning the model $\boldsymbol{g_\Phi}$ on $\boldsymbol{\delta}_G$ facilitates the transfer of global structure from the facial imagery to produce highly-efficient perturbations, i.e., these perturbations are more successful in disturbing manipulation models to produce results close to the manipulation targets. The real image $\mathbf{X}_i$ and global perturbation $\boldsymbol{\delta}_G$ are first concatenated channel-wise, $\widehat{\mathbf{X}}_i = [\mathbf{X}_i, \boldsymbol{\delta}_G]$, to generate a six-channel input $\widehat{\mathbf{X}}_i \in \mathbb{R}^{H\times W\times 6}$.

$\widehat{\mathbf{X}}_i$ is then passed through the conditional model $\boldsymbol{g_\Phi}$ to generate image-specific perturbation $\boldsymbol{\delta}_i = \boldsymbol{g_\Phi}(\widehat{\mathbf{X}}_i)$. These image-specific perturbations $\boldsymbol{\delta}_i$ are then added to the respective input images $\mathbf{X}_i$ to generate the protected image $\mathbf{X}_i^p$ as

$$\mathbf{X}_i^p = \text{Clamp}_\varepsilon(\mathbf{X}_i + \boldsymbol{\delta}_i). \tag{3}$$

The $\text{Clamp}_\varepsilon(\xi)$ function projects higher/lower values of ξ into the valid interval $[-\varepsilon, \varepsilon]$. Similarly, we generate the protected image using global perturbation $\boldsymbol{\delta}_G$ as

$$\mathbf{X}_i^{Gp} = \text{Clamp}_\varepsilon(\mathbf{X}_i + \boldsymbol{\delta}_G). \tag{4}$$

For the generated conditional and global protected image $\mathbf{X}_i^p$ and $\mathbf{X}_i^{Gp}$ and the given manipulation model $\mathbf{f}_{\boldsymbol{\Theta}}^k$, the reconstruction loss $\mathcal{L}_i^{\text{recon}}$ and perturbation loss $\mathcal{L}_i^{\text{perturb}}$ are formulated as

$$\mathcal{L}_i^{\text{recon}} = \left\| \mathbf{f}_{\boldsymbol{\Theta}}^k(\mathbf{X}_i^p) - \mathbf{Y}_k^{\text{target}} \right\|_2 + \left\| \mathbf{f}_{\boldsymbol{\Theta}}^k(\mathbf{X}_i^{Gp}) - \mathbf{Y}_k^{\text{target}} \right\|_2. \tag{5}$$

$$\mathcal{L}_i^{\text{perturb}} = \left\| \mathbf{X}_i^p - \mathbf{X}_i \right\|_2 + \left\| \mathbf{X}_i^{Gp} - \mathbf{X}_i \right\|_2. \tag{6}$$

Finally, the overall loss can then be written as

$$\mathcal{L}_k = \left[\sum_{i=1}^{N} \left\| \mathbf{f}_{\boldsymbol{\Theta}}^k(\mathbf{X}_i^p) - \mathbf{Y}_k^{\text{target}} \right\|_2 + \left\| \mathbf{f}_{\boldsymbol{\Theta}}^k(\mathbf{X}_i^{Gp}) - \mathbf{Y}_k^{\text{target}} \right\|_2 + \lambda \left(\left\| \mathbf{X}_i^p - \mathbf{X}_i \right\|_2 + \left\| \mathbf{X}_i^{Gp} - \mathbf{X}_i \right\|_2 \right) \right]_k. \tag{7}$$

The global perturbation $\boldsymbol{\delta}_G$ is initialized with a random vector sampled from a multivariate uniform distribution, i.e., $\boldsymbol{\delta}_{\mathbf{G}}^0 \sim \mathcal{U}(\mathbf{0}, \mathbf{1})$ and optimized iteratively. Note that $\mathbf{X}_i^{Gp}$ is used only to drive the optimization of $\boldsymbol{\delta}_G$. For further details on the network architecture and hyperparameters, we refer to Sec. 4 and the supplemental material.

Differentiable JPEG Compression In many practical scenarios, images shared on social media platforms get compressed over the course of transmission. Our initial experiments suggest that protected images $\mathbf{X}_i^p$ generated from the previous steps can easily become ineffective by applying image compression. In order to make our perturbations robust, we propose to incorporate a differentiable JPEG compression into our generative model; i.e., we aim to generate perturbations that still disrupt the manipulation models even if the input is compressed. The actual JPEG compression technique [52] is non-differentiable due to the lossy quantization step (details in supplemental) where information loss happens with the round operation as, $x := \text{round}(x)$. Therefore, we cannot train our protected images against the original JPEG technique. Instead, we leverage

continuous and differentiable approximations [25, 45] to the rounding operator. For our experiments, we use the sin approximation by Korus et al. [25]

$$x := x - \frac{\sin(2\pi x)}{2\pi}. \tag{8}$$

This differentiable round approximation coupled with other transformations from the actual JPEG technique can be formalized into differentiable JPEG operation. We denote the full differentiable JPEG compression as $\boldsymbol{\Psi^q}$, where $\boldsymbol{q}$ denotes the compression quality.

For training, we first map the protected image $\mathbf{X}_i^p$ to RGB colorspace $[0, 255]$ before applying image compression, obtaining $\widetilde{\mathbf{X}}_i^p$. Next, the image $\widetilde{\mathbf{X}}_i^p$ is passed through differential JPEG layers $\boldsymbol{\Psi^q}$ to generate a compressed image $\widetilde{\mathbf{X}}_{ic}^p$, which is then normalized again as $\mathbf{X}_{ic}^p$ before passing it to the manipulation model $\mathbf{f}_{\boldsymbol{\Theta}}$.

Training with a fixed compression quality ensures robustness to that specific quality but shows limited performance when evaluated with different compression qualities. We therefore, generalize across compression levels by training our model with different compression qualities. Specifically, at each iteration, we randomly sample quality $\boldsymbol{q}$ from a discrete uniform distribution $\mathcal{U}_{\mathcal{D}}(1, 99)$, i.e. $\boldsymbol{q} \sim \mathcal{U}_{\mathcal{D}}(1, 99)$ and compress the protected image $\mathbf{X}_i^p$ at quality level $\boldsymbol{q}$.

This modifies the reconstruction loss $\mathcal{L}_{\text{recon}}$ as follows

$$\mathcal{L}_i^{\text{recon}} = \left\| \mathbf{f}_{\boldsymbol{\Theta}}^k(\boldsymbol{\Psi^q}(\mathbf{X}_i^p)) - \mathbf{Y}_k^{\text{target}} \right\|_2 + \left\| \mathbf{f}_{\boldsymbol{\Theta}}^k(\mathbf{X}_i^{Gp}) - \mathbf{Y}_k^{\text{target}} \right\|_2 \tag{9}$$

where $\mathbf{X}_{ic}^p = \boldsymbol{\Psi^q}(\mathbf{X}_i^p)$ denotes the compressed protected image. Backpropagating the gradients through $\boldsymbol{\Psi^q}$ during training ensures that the added perturbations survive different compression qualities. At test time, we evaluate results with actual JPEG compression technique instead of approximated/differential used during training to report the results.

Multiple Manipulation Methods To handle multiple manipulation models simultaneously, we combine model-specific perturbations $\{\boldsymbol{\delta}_i^k\}_{k=1}^K$ obtained previously using $\{\boldsymbol{\delta}_G^k, \mathbf{g}_{\boldsymbol{\Phi}}^k\}_{k=1}^K$ and feed them to our attention network $\mathbf{h}_{\boldsymbol{\omega}}$ (parameterized by $\boldsymbol{\omega}$) and fusion network $\mathbf{r}_{\boldsymbol{\rho}}$ (parameterized by $\boldsymbol{\rho}$) to generate model-agnostic perturbations $\boldsymbol{\delta}_i^{\text{all}}$.

$$\boldsymbol{\omega}^*, \boldsymbol{\rho}^* = \underset{\omega, \rho}{\text{argmin}} \quad \mathcal{L}_{\text{all}}. \tag{10}$$

We leverage the pre-trained global pattern and conditional perturbation model pairs $\{\boldsymbol{\delta}_G^k, \mathbf{g}_{\boldsymbol{\Phi}}^k\}_{k=1}^K$ for each of the K different models to generate the final perturbation $\boldsymbol{\delta}_i^{\text{all}}$ for image $\mathbf{X}_i$. More precisely, for the image $\mathbf{X}_i$, we first use the pre-trained $\{\boldsymbol{\delta}_G^k, \mathbf{g}_{\boldsymbol{\Phi}}^k\}_{k=1}^K$ to generate the model-specific perturbations $\{\boldsymbol{\delta}_i^k\}_{k=1}^K$ as:

$$\boldsymbol{\delta}_i^k = \mathbf{g}_{\boldsymbol{\Phi}}^k(\mathbf{X}_i, \boldsymbol{\delta}_G^k). \tag{11}$$

Next, these model-specific perturbations $\{\boldsymbol{\delta}_i^k\}_{k=1}^K$ are fed into attention module $\mathbf{h}_{\boldsymbol{\omega}}$ coupled with the softmax operation to generate spatial attention maps $\{\boldsymbol{\alpha}_i^k\}_{k=1}^K$ as:

$$\boldsymbol{\alpha}_i^k = \frac{\exp\big(\mathbf{h}_{\boldsymbol{\omega}}(\boldsymbol{\delta}_i^k, C_k)\big)}{\sum_{k=1}^{K} \exp\big(\mathbf{h}_{\boldsymbol{\omega}}(\boldsymbol{\delta}_i^k, C_k)\big)}. \tag{12}$$

where $\boldsymbol{\alpha}_i^k \in \mathbb{R}^{H \times W}$ and C_k refer to class label for the k-th manipulation model. These spatial attention maps are then blended with model-specific perturbations and refined with a fusion network $\mathbf{r}_{\boldsymbol{\rho}}$ to generate the final perturbation $\boldsymbol{\delta}_i^{\text{all}}$ as:

$$\boldsymbol{\delta}_i^{\text{all}} = \mathbf{r}_{\boldsymbol{\rho}}\left(\sum_{k=1}^{K}(\boldsymbol{\alpha}_i^k \odot \boldsymbol{\delta}_i^k)\right) \tag{13}$$

Finally, $\boldsymbol{\delta}_i^{\text{all}}$ is added to the image $\mathbf{X}_i$ to generate the common protected image $\mathbf{X}_i^{\text{all}} = \text{Clamp}_{\varepsilon}(\mathbf{X}_i + \boldsymbol{\delta}_i^{\text{all}})$ and total loss is formalized as

$$\mathcal{L}_{\text{all}} = \sum_{i=1}^{N}\left[\sum_{k=1}^{K}\left(\left\|\mathbf{f}_{\boldsymbol{\Theta}}^k(\mathbf{X}_i^{\text{all}}) - \mathbf{Y}_k^{\text{target}}\right\|_2\right) + \lambda\left\|\boldsymbol{\delta}_i^{\text{all}}\right\|_2\right]. \tag{14}$$

4 Results

We compare our method against well-studied adversarial attack baselines I-FGSM [26] and I-PGD [32]. To demonstrate our results, we perform experiments with three different models: (1) pSp Encoder [41] which can be used for self-reconstruction and style-mixing (protected with solid white image as manipulation target), and (2) SimSwap [8] for face-swapping (protected with solid blue as manipulation target). (3) StyleClip [39] for text-driven manipulation (protected with solid red as manipulation target). For all these manipulations, we use the publicly available pre-trained models. For pSp encoder, we use a model that is trained for a self-reconstruction task. The same model can also be used for style-mixing to synthesize new images by mixing the latent style features of two images. For style-mixing and face-swapping, protection is applied to the target image. We introduce a custom split on FFHQ [22] for our experiments. We use 10K images for training and 1K images for val and test split each. More details can be found in supplemental. All results are reported on the corresponding test sets for each task respectively.

Experimental Setup. All images are first resized to 256×256 pixels. The global perturbation and conditional model are jointly optimized for 100k iterations with a learning rate of 0.0001 and Adam optimizer. For the protection $\boldsymbol{g}_{\boldsymbol{\Phi}}$, attention $\boldsymbol{h}_{\boldsymbol{\omega}}$ and fusion $\boldsymbol{r}_{\boldsymbol{\rho}}$ network, we use the same UNet-64 encoder-decoder architecture. We use a batch size of 1 for all our experiments. For I-PGD, we use a step size of 0.01. Both I-FGSM and I-PGD are optimized for 100 steps for every image in the test split. More details on training setup and hyperparameters can be found in the supplemental.

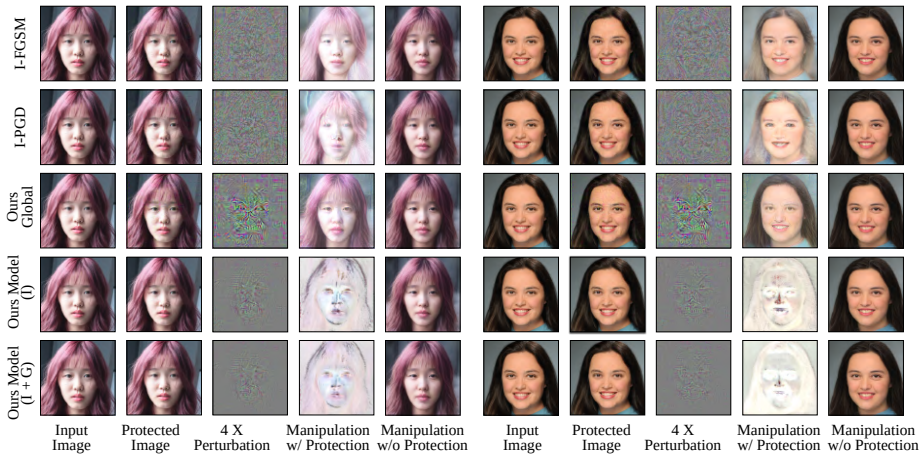

Fig. 4. Comparison on self-reconstruction task with white image as manipulation target. Perturbation enlarged (4×) for better visibility. *Ours Global* refers to the optimized single global perturbation for all the images. *Ours Model (I)* refers to the model conditioned only on real images and *Ours Model (I + G)* refers to the model conditioned on global perturbation and real image, outperforms alternate baselines.

Metrics. To evaluate the output quality, we compute relative performance at different perturbation levels, i.e., we plot a graph with the x-axis showing different perturbation levels for the image and the y-axis showing how close is the output of the face manipulation model to the predefined manipulation target. We plot the graph for RMSE, PSNR, LPIPS [61] and VGG loss. In the optimal setting, for a low perturbation in the image, the output should look identical to the manipulation target; i.e., a lower graph is better for RMSE, LPIPS and VGG loss and higher for PSNR.

Baseline Comparisons. To compare our method against other adversarial attack baselines, we first evaluate the results of our proposed method on a single manipulation model without compression; i.e., neither training nor evaluating for JPEG compression. Visual results for self-reconstruction and style mixing are shown in Figs. 4 and 5. The performance graph for different perturbation levels is shown in Fig. 6. We observe that the model conditioned on the global perturbation as well as real images outperforms the model trained only with real images, indicating that the global perturbation provides a strong prior in generating more powerful perturbations.

Runtime Comparison. We compare run-time performance against state-of-the-art in Tab. 1. I-FGSM [26] and I-PGD [32] optimize for perturbation patterns for each image individually at run time; hence they are orders of magnitude slower than our method that only requires a single forward pass of our conditional generative neural network. Our model takes only 77.89 ± 2.71 ms

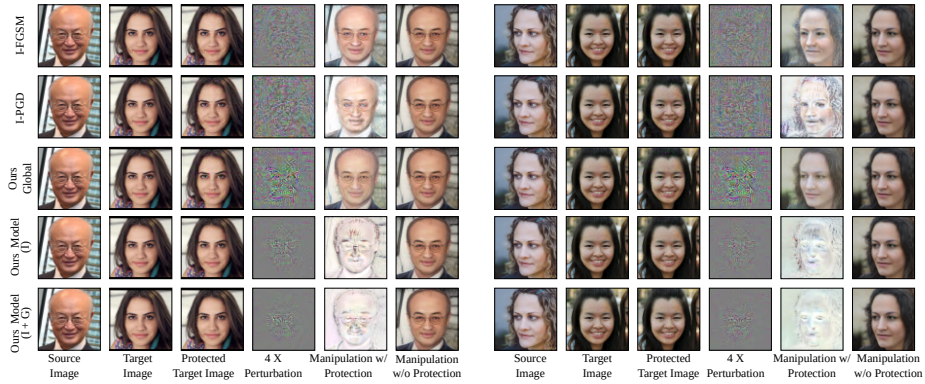

Fig. 5. Comparison on the style-mixing task (white target). The protection is applied to the target image. All methods are trained only for the self-reconstruction task and evaluated on style-mixing. Perturbation enlarged (4×) for better visibility. *Ours Global* refers to the optimized single global perturbation. *Ours Model (I)* refers to the model conditioned only on real images and *Ours Model (I + G)* refers to the model conditioned on global perturbation and real image, outperforms alternate baselines.

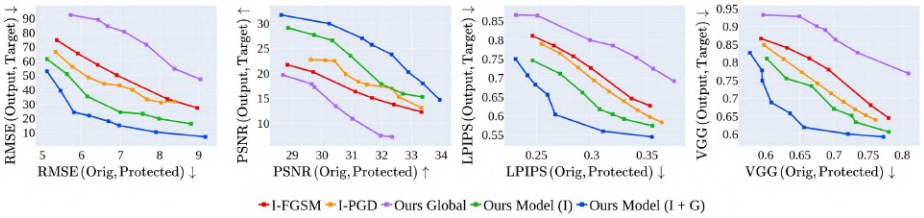

Fig. 6. Comparison with different optimization techniques evaluated on self-reconstruction (white target). We plot the output image quality (y-axis) corresponding to different levels of perturbations added to the image (x-axis). *Orig* and *Protected* refer to the original and protected image. *Output* refers to the output of the manipulation model and *Target* indicates the predefined manipulation target. Note that our method outperforms other baselines at all the different perturbation levels.

and 117.0 MB memory to compute the perturbation for a single image on an Intel(R) Xeon(R) W-2133 CPU @ 3.60GHz. This is an order of magnitude faster compared to per-image methods that are run on GPUs. We believe this makes our method ideally suited to real-time scenarios, even on mobile hardware.

Robustness to JPEG Compression. Next, we investigate the sensitivity of perturbations to different compression qualities. We apply the actual JPEG compression technique to report results. We observe that without training the model against different compression leads to degraded results when evaluated

Table 1. Run-time performance (averaged over 10 runs) to generate a perturbation for a single image on the self-reconstruction task. Our method runs an order of magnitude faster than existing works that require per-image optimization. All timings are measured on an Nvidia Titan RTX 2080 GPU.

Method	Time
I-FGSM [26]	17517.71 ms (±124.08 ms)
I-PGD [32]	17523.01 ms (±204.15 ms)
Ours	**10.66 ms** (±0.21 ms)

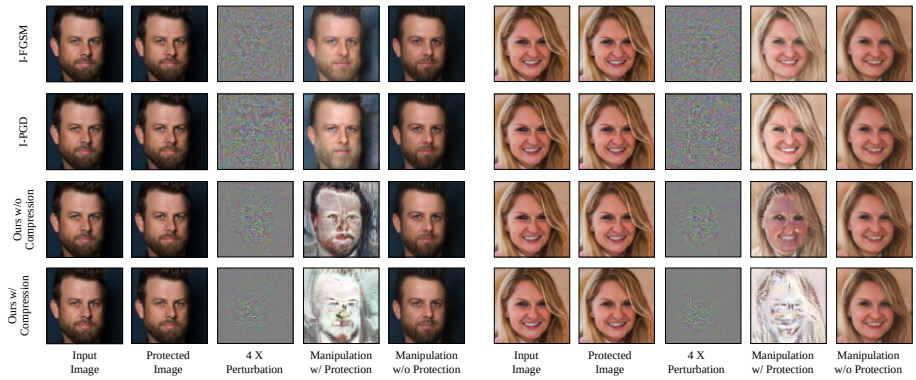

Fig. 7. Qualitative comparison in the presence of JPEG compression (white target). Methods trained without compression struggle; in contrast, our model trained with compression is able to produce perturbations that are robust to compression. *Ours w/ Compression* refers to the model trained with random compression. *Ours w/o Compression* refers to model trained without compression. Compression is applied on the protected images. All methods are evaluated at compression quality C-80.

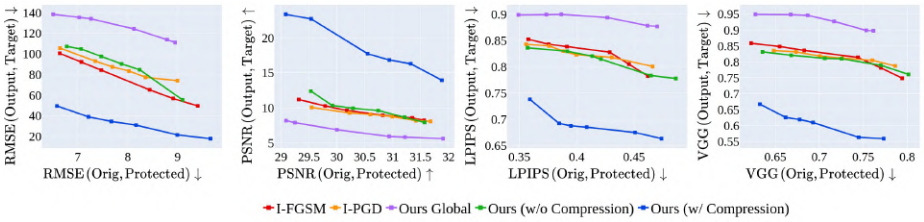

Fig. 8. Performance comparison in the presence of JPEG compression. Our method without differentiable JPEG training manages to disrupt the model; however, training with random compression levels significantly outperforms the uncompressed baselines. All methods are evaluated at compression quality C-80.

on compressed images, Fig. 7 and 8. Training the model with fixed compression quality makes the perturbation robust to that specific compression quality; however, it fails for other compression levels; see Fig. 9. We therefore train across different compression levels varied during training iterations.

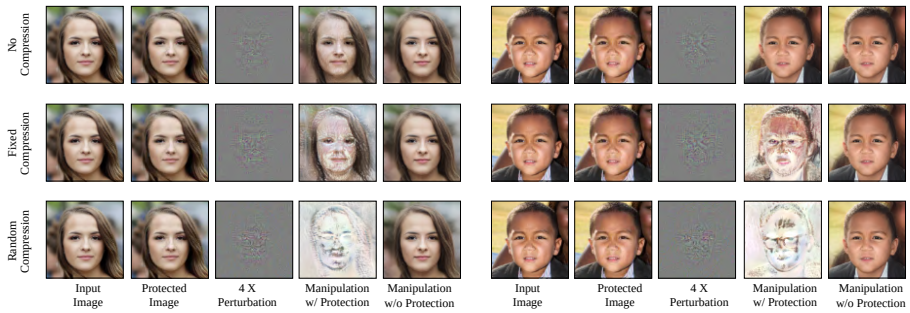

Fig. 9. Comparison for our method trained without compression, fixed compression, and random compression for self-reconstruction task (white target). The fixed compression model was trained with compression quality C-80. All methods are evaluated on compression quality C-30. The randomly compressed model outperforms both fixed and no compression models.

Multiple Manipulation Models. We leverage manipulation-specific perturbations as priors and combine them using attention-based fusion to generate a single perturbation to protect against multiple manipulations at the same time. As a baseline, we also compare against a model trained directly for all manipulation methods combined without manipulation-specific priors or attention. We notice that this setup is unable to produce optimal perturbations due to absence of prior information from manipulation-specific perturbations which provide the most optimal perturbations to produce predefined manipulation targets. We color-code the manipulation targets with different colors for different manipulation models. This protection technique has an advantage over simple disruption since it gives more information about which technique was used to manipulate the image. We conduct experiments with three different state-of-the-art methods: pSp [41] with solid white image as the manipulation target, SimSwap [8] with solid blue image as target and StyleClip [39] with solid red as target image. Visual results and performance graph comparison are shown in Fig. 10. Our combined model with attention produces more effective results than without attention baseline, and without any significant degradation compared to manipulation-specific baselines when handling multiple manipulation at the same time.

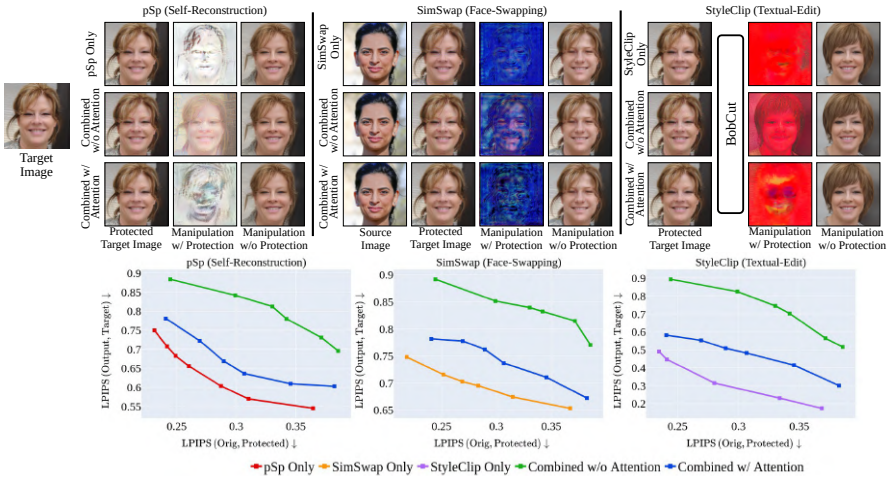

Fig. 10. Visual results (top) and performance graph (bottom) for multiple targets simultaneously. *pSp Only*, *SimSwap Only*, and *StyleClip Only* refer to the individual protection models trained only for the respective manipulations. *Combined w/o Attention* refers to a model trained directly for all manipulation methods combined. *Combined w/ Attention* refers to our proposed attention-based fusion approach. Our proposed attention model performs much better the no attention baseline, and is comparable to individual models.

5 Conclusion

In this work, we proposed a data-driven approach to protect face images from potential popular manipulations. Our method can both prevent and simultaneously identify the manipulation technique by generating the predefined manipulation target as output. In comparison to existing works, our method not only run orders of magnitude faster, but also achieves superior performance; i.e., with smaller perturbations of a given input image, we can achieve larger disruptions in the respective manipulation methods. In addition, we proposed an end-to-end compression formulation to make the perturbation robust to compression. Furthermore, we propose a new attention-based fusion approach to handle multiple manipulations simultaneously. We believe our generalized, data-driven method takes an important step towards addressing the potential misuse of popular face image manipulation techniques.

Acknowledgments

This work is supported by a TUM-IAS Rudolf Mößbauer Fellowship, the ERC Starting Grant Scan2CAD (804724), and Sony Semiconductor Solutions Corporation. We would also like to thank Angela Dai for video voice over.

References

1. Maisy Kinsley fake account. https://twitter.com/sokane1/status/1111023838467362816, accessed: 2019-03-27 2
2. Afchar, D., Nozick, V., Yamagishi, J., Echizen, I.: Mesonet: a compact facial video forgery detection network (Dec 2018). https://doi.org/10.1109/wifs.2018.8630761, http://dx.doi.org/10.1109/WIFS.2018.8630761 2
3. Agarwal, S., Farid, H.: Photo forensics from jpeg dimples. In: 2017 IEEE Workshop on Information Forensics and Security (WIFS). pp. 1–6 (2017). https://doi.org/10.1109/WIFS.2017.8267641 3
4. Agarwal, S., Farid, H., Gu, Y., He, M., Nagano, K., Li, H.: Protecting World Leaders Against Deep Fakes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops. p. 8. IEEE, Long Beach, CA (Jun 2019) 2
5. Agarwal, S., Farid, H., Gu, Y., He, M., Nagano, K., Li, H.: Protecting world leaders against deep fakes. In: CVPR Workshops (2019) 3
6. Aneja, S., Nießner, M.: Generalized Zero and Few-Shot Transfer for Facial Forgery Detection. In: ArXiv preprint arXiv:2006.11863 (2020) 2, 3
7. Brock, A., Donahue, J., Simonyan, K.: Large scale GAN training for high fidelity natural image synthesis. In: International Conference on Learning Representations (2019), https://openreview.net/forum?id=B1xsqj09Fm 3
8. Chen, R., Chen, X., Ni, B., Ge, Y.: Simswap: An efficient framework for high fidelity face swapping. In: MM '20: The 28th ACM International Conference on Multimedia. pp. 2003–2011. ACM (2020) 2, 3, 9, 13
9. Choi, Y., Choi, M., Kim, M., Ha, J.W., Kim, S., Choo, J.: Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2018) 2, 3
10. Choi, Y., Uh, Y., Yoo, J., Ha, J.W.: Stargan v2: Diverse image synthesis for multiple domains. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2020) 2, 3
11. Cozzolino, D., Rössler, A., Thies, J., Nießner, M., Verdoliva, L.: Id-reveal: Identity-aware deepfake video detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 15108–15117 (October 2021) 2, 3
12. Cozzolino, D., Thies, J., Rössler, A., Riess, C., Nießner, M., Verdoliva, L.: Forensictransfer: Weakly-supervised domain adaptation for forgery detection. arXiv (2018) 2, 3
13. Dolhansky, B., Bitton, J., Pflaum, B., Lu, J., Howes, R., Wang, M., Ferrer, C.C.: The deepfake detection challenge (dfdc) dataset (2020) 2
14. Dong, Y., Liao, F., Pang, T., Su, H., Zhu, J., Hu, X., Li, J.: Boosting adversarial attacks with momentum. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition pp. 9185–9193 (2018) 3
15. Ferrara, P., Bianchi, T., De Rosa, A., Piva, A.: Image forgery localization via fine-grained analysis of cfa artifacts. IEEE Transactions on Information Forensics and Security **7**(5), 1566–1577 (2012). https://doi.org/10.1109/TIFS.2012.2202227 3
16. Fischer, V., Kumar, M.C., Metzen, J.H., Brox, T.: Adversarial examples for semantic image segmentation (2017) 3
17. Gafni, G., Thies, J., Zollhöfer, M., Nießner, M.: Dynamic neural radiance fields for monocular 4d facial avatar reconstruction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 8649–8658 (June 2021) 3

18. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples (2015) 2, 3
19. Huang, Q., Zhang, J., Zhou, W., WeimingZhang, Yu, N.: Initiative defense against facial manipulation (2021) 2, 4
20. Karras, T., Aila, T., Laine, S., Lehtinen, J.: Progressive growing of GANs for improved quality, stability, and variation. In: International Conference on Learning Representations (2018), https://openreview.net/forum?id=Hk99zCeAb 3
21. Karras, T., Aittala, M., Laine, S., Härkönen, E., Hellsten, J., Lehtinen, J., Aila, T.: Alias-free generative adversarial networks. In: Proc. NeurIPS (2021) 3
22. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks (2019) 3, 9
23. Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., Aila, T.: Analyzing and improving the image quality of stylegan. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2020) 3
24. Kim, H., Garrido, P., Tewari, A., Xu, W., Thies, J., Nießner, M., Péerez, P., Richardt, C., Zollhöfer, M., Theobalt, C.: Deep video portraits. ACM Transactions on Graphics 2018 (TOG) (2018) 3
25. Korus, P., Memon, N.: Content authentication for neural imaging pipelines: End-to-end optimization of photo provenance in complex distribution channels. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019) 8
26. Kurakin, A., Goodfellow, I.J., Bengio, S.: Adversarial machine learning at scale. ArXiv **abs/1611.01236** (2017) 9, 10, 12
27. Li, L., Bao, J., Zhang, T., Yang, H., Chen, D., Wen, F., Guo, B.: Face x-ray for more general face forgery detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5001–5010 (2020) 2
28. Li, Y., Lyu, S.: Exposing deepfake videos by detecting face warping artifacts (2018) 2
29. Li, Y., Yang, X., Wu, B., Lyu, S.: Hiding faces in plain sight: Disrupting ai face synthesis with adversarial perturbations. ArXiv **abs/1906.09288** (2019) 2, 4
30. Luo, X., Zhan, R., Chang, H., Yang, F., Milanfar, P.: Distortion agnostic deep watermarking. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 13545–13554 (2020) 4
31. Lyu, S., Pan, X., Zhang, X.: Exposing region splicing forgeries with blind local noise estimation. International Journal of Computer Vision **110**(2), 202–221 (2014) 3
32. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. ArXiv **abs/1706.06083** (2018) 9, 10, 12
33. Metzen, J.H., Kumar, M.C., Brox, T., Fischer, V.: Universal adversarial perturbations against semantic image segmentation. In: submitted (2017), https://arxiv.org/abs/1704.05712 3
34. Moosavi-Dezfooli, S.M., Fawzi, A., Fawzi, O., Frossard, P.: Universal adversarial perturbations. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (July 2017) 3
35. Moosavi-Dezfooli, S.M., Fawzi, A., Frossard, P.: Deepfool: A simple and accurate method to fool deep neural networks. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 2574–2582 (2016) 3
36. Nguyen, H.H., Yamagishi, J., Echizen, I.: Capsule-forensics: Using capsule networks to detect forged images and videos. ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (May 2019). https://doi.org/10.1109/icassp.2019.8682602, http://dx.doi.org/10.1109/ICASSP.2019.8682602 2

37. Nirkin, Y., Keller, Y., Hassner, T.: FSGAN: Subject agnostic face swapping and reenactment. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 7184–7193 (2019) 2, 3
38. Park, T., Liu, M.Y., Wang, T.C., Zhu, J.Y.: Gaugan: Semantic image synthesis with spatially adaptive normalization. In: ACM SIGGRAPH 2019 Real-Time Live! SIGGRAPH '19, Association for Computing Machinery, New York, NY, USA (2019) 3
39. Patashnik, O., Wu, Z., Shechtman, E., Cohen-Or, D., Lischinski, D.: Styleclip: Text-driven manipulation of stylegan imagery. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 2085–2094 (October 2021) 2, 3, 9, 13
40. Poursaeed, O., Katsman, I., Gao, B., Belongie, S.: Generative adversarial perturbations. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4422–4431 (2018) 3
41. Richardson, E., Alaluf, Y., Patashnik, O., Nitzan, Y., Azar, Y., Shapiro, S., Cohen-Or, D.: Encoding in style: a stylegan encoder for image-to-image translation. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2021) 2, 3, 9, 13
42. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: MICCAI (2015) 6
43. Rössler, A., Cozzolino, D., Verdoliva, L., Riess, C., Thies, J., Nießner, M.: FaceForensics++: Learning to detect manipulated facial images. In: International Conference on Computer Vision (ICCV) (2019) 2, 3
44. Ruiz, N., Bargal, S.A., Sclaroff, S.: Disrupting deepfakes: Adversarial attacks against conditional image translation networks and facial manipulation systems (2020) 2, 4
45. Shin, R.: Jpeg-resistant adversarial images (2017) 8
46. Siarohin, A., Lathuilière, S., Tulyakov, S., Ricci, E., Sebe, N.: First order motion model for image animation. In: Conference on Neural Information Processing Systems (NeurIPS) (December 2019) 3
47. Sun, P., Li, Y., Qi, H., Lyu, S.: Landmark breaker: Obstructing deepfake by disturbing landmark extraction. In: 2020 IEEE International Workshop on Information Forensics and Security (WIFS). pp. 1–6 (2020). https://doi.org/10.1109/WIFS49906.2020.9360910 2, 4
48. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R.: Intriguing properties of neural networks (2014) 2, 3
49. Tancik, M., Mildenhall, B., Ng, R.: Stegastamp: Invisible hyperlinks in physical photographs. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2020) 4
50. Thies, J., Zollhöfer, M., Stamminger, M., Theobalt, C., Nießner, M.: Face2face: Real-time face capture and reenactment of rgb videos. In: Proc. Computer Vision and Pattern Recognition (CVPR), IEEE (2016) 3
51. Thies, J., Zollhöfer, M., Nießner, M.: Deferred neural rendering: Image synthesis using neural textures. ACM Transactions on Graphics 2019 (TOG) (2019) 3
52. Wallace, G.K.: The jpeg still picture compression standard. IEEE transactions on consumer electronics **38**(1), xviii–xxxiv (1992) 7
53. Wang, R., Juefei-Xu, F., Luo, M., Liu, Y., Wang, L.: Faketagger: Robust safeguards against deepfake dissemination via provenance tracking (2021) 4
54. Wengrowski, E., Dana, K.: Light field messaging with deep photographic steganography. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1515–1524 (2019) 4

55. Xie, C., Wang, J., Zhang, Z., Zhou, Y., Xie, L., Yuille, A.: Adversarial examples for semantic segmentation and object detection. In: International Conference on Computer Vision. IEEE (2017) 3
56. Yang, X., Li, Y., Lyu, S.: Exposing deep fakes using inconsistent head poses. ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (May 2019). https://doi.org/10.1109/icassp.2019.8683164, http://dx.doi.org/10.1109/ICASSP.2019.8683164 2
57. Yang, Y., Liang, C., He, H., Cao, X., Gong, N.Z.: Faceguard: Proactive deepfake detection (2021) 2, 4
58. Yeh, C.Y., Chen, H., Tsai, S.L., Wang, S.D.: Disrupting image-translation-based deepfake algorithms with adversarial attacks. 2020 IEEE Winter Applications of Computer Vision Workshops (WACVW) pp. 53–62 (2020) 2, 4
59. Yeh, C.Y., Chen, H.W., Shuai, H.H., Yang, D.N., Chen, M.S.: Attack as the best defense: Nullifying image-to-image translation gans via limit-aware adversarial attack. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 16188–16197 (October 2021) 2, 4
60. Zakharov, E., Shysheya, A., Burkov, E., Lempitsky, V.: Few-shot adversarial learning of realistic neural talking head models (2019) 3
61. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: CVPR (2018) 10
62. Zhou, P., Han, X., Morariu, V.I., Davis, L.S.: Two-stream neural networks for tampered face detection (Jul 2017). https://doi.org/10.1109/cvprw.2017.229, http://dx.doi.org/10.1109/CVPRW.2017.229 2
63. Zhu, J., Kaplan, R., Johnson, J., Fei-Fei, L.: Hidden: Hiding data with deep networks. In: ECCV (2018) 4