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 [10, 42] whose models are publicly accessible can be misused to generate non-consensual synthetic imagery. Other examples include face attribute manipulation methods [11, 12, 44, 46] 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 [16]. 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 [3, 5, 8, 13, 14, 32, 33, 41, 49, 64, 70] 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 [24, 50, 65–67]. 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 [22, 54]. Methods optimizing noise patterns for every image from scratch [50, 65–67] 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 [24, 34, 50, 53] or nullify [66, 67] 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   [9, 25–28, 43, 58] 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 [11,12,44,46] and face-swapping models [10,42] facilitate the manipulation of existing face images. Similarly, facial re-enactment tools [20, 29, 52, 57, 68] 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 [4,18,36]. While such self-consistency can produce good results, these methods are less accurate than more recent learning-based techniques [6, 8, 13, 14, 49], 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 [17, 22, 40, 54] and eventually expanded to semantic segmentation and detection models [19, 45, 63]. 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 [38,39]. This has shown to be effective for misclassification tasks but gives suboptimal results for generative models, as we show in Sec. 4.
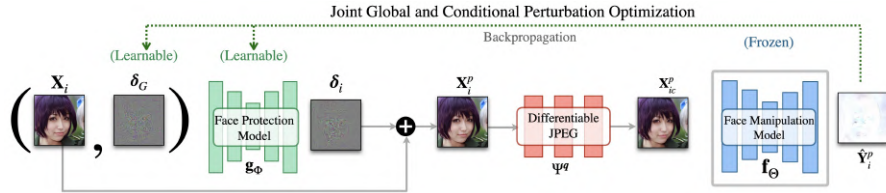
Fig. 2: We first pass the real image $\mathbf{X}_i$ and the global perturbation $\boldsymbol{\delta}_G$ through the face protection model $\mathbf{g}_{\boldsymbol{\Phi}}$ to generate the image-specific perturbation $\boldsymbol{\delta}_i$. This perturbation is then added to the original image to create the protected image $\mathbf{X}_i^p$. The protected image is then compressed using the differentiable JPEG $\boldsymbol{\Psi}^q$ 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 [35, 55, 61, 62, 65, 71] can be used to embed an image-specific watermark to secure an image. For instance, FaceGuard [65] 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. [50] disrupt the output of deepfake generation models. Yeh et al. [66, 67] aim to nullify the effect of image manipulation models. Other approaches [34, 53] 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 [24] 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.
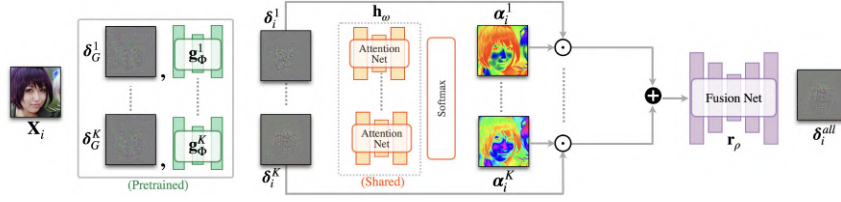
Fig. 3: For a given RGB image $\mathbf{X}_i$, we first use the pre-trained manipulation-specific global noise and protection models $\{\boldsymbol{\delta}_G^k, \mathbf{g}_{\boldsymbol{\Phi}}^k\}_{k=1}^K$ to generate manipulation-specific perturbations $\{\boldsymbol{\delta}_i^k\}_{k=1}^K$, which are passed into a shared attention backbone $\mathbf{h}_{\boldsymbol{\omega}}$ 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 $\boldsymbol{r}_{\boldsymbol{\rho}}$ to generate manipulation-agnostic perturbation $\boldsymbol{\delta}_i^{\mathrm{all}}$.

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 $\mathcal{Y} = \{\mathbf{Y}_k^{\mathrm{target}}\}_{k=1}^K$.

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 $\boldsymbol{g}_{\boldsymbol{\Phi}}$ 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 $h_{\boldsymbol{\omega}}$ (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 $\boldsymbol{r}_{\boldsymbol{\rho}}$ (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 [47] 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.

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

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, $\widehat{\mathbf{X}}_i = \left[\mathbf{X}_i, \boldsymbol{\delta_G} \right]$, 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

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

The $\mathrm{Clamp}_\varepsilon(\xi)$ 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

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

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 $\mathcal{L}_i^{\mathrm{recon}}$ and perturbation loss $\mathcal{L}_i^{\mathrm{perturb}}$ are formulated as

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

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

Finally, the overall loss can then be written as

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

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 [60] is non-differentiable due to the lossy quantization step (details in supplemental) where information loss happens with the round operation as, $x := \mathrm{round}(x)$. Therefore, we cannot train our protected images against the original JPEG technique. Instead, we leverage

continuous and differentiable approximations [30, 51] to the rounding operator. For our experiments, we use the sin approximation by Korus et al. [30]

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

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 $\mathcal{L}_{\mathrm{recon}}$ as follows

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

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 $\boldsymbol{\delta}_i^{\mathrm{all}}$.

$$\boldsymbol{\omega}^*, \boldsymbol{\rho}^* = \operatorname*{argmin}_{\omega, \rho} \ \mathcal{L}_{\mathrm{all}}. \tag{10}$$

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 $\boldsymbol{\delta}_i^{\mathrm{all}}$ 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:

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

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 $\boldsymbol{\delta}_i^{\mathrm{all}}$ as:

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

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

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

4   Results

We compare our method against well-studied adversarial attack baselines I-FGSM [31] and I-PGD [37]. To demonstrate our results, we perform experiments with three different models: (1) pSp Encoder [46] which can be used for self-reconstruction and style-mixing (protected with solid white image as manipulation target), and (2) SimSwap [10] for face-swapping (protected with solid blue as manipulation target). (3) StyleClip [44] 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 [27] 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 $\boldsymbol{g_{\Phi}}$, attention $\boldsymbol{h_{\omega}}$ and fusion $\boldsymbol{r_{\rho}}$ 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.
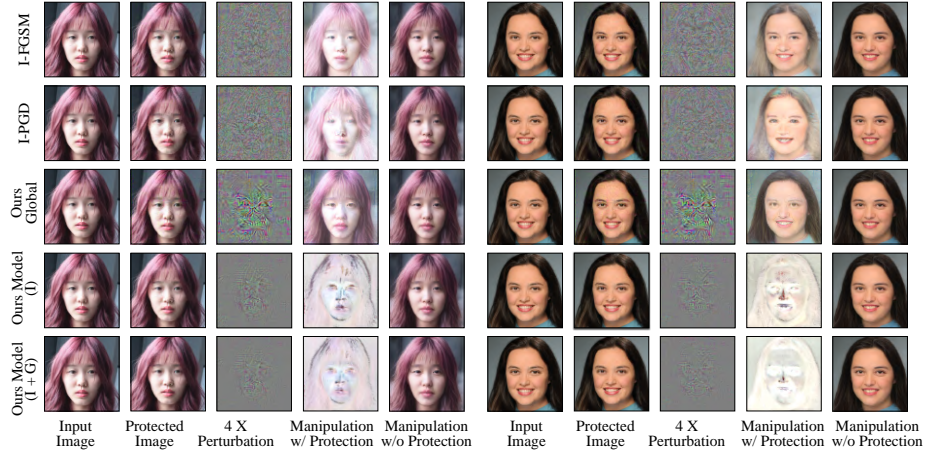
Fig. 4: Comparison on self-reconstruction task with white image as manipulation target. Perturbation enlarged (4×) for better visibility. *Ours Global* refers to the optimized single global perturbation for all the images. *Ours Model (I)* refers to the model conditioned only on real images and *Ours 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 [69] 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 [31] and I-PGD [37] 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 and 117.0 MB memory to compute the perturbation for a single image on an
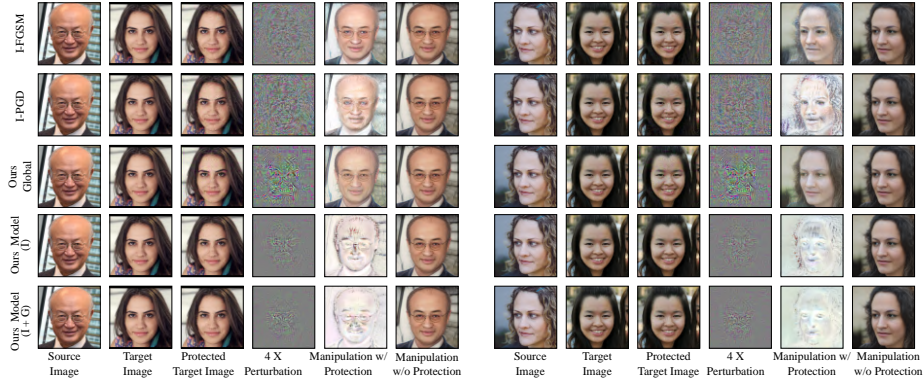
Fig. 5: Comparison on the style-mixing task (white target). The protection is applied to the target image. All methods are trained only for the self-reconstruction task and evaluated on style-mixing. Perturbation enlarged (4×) for better visibility. *Ours Global* refers to the optimized single global 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.
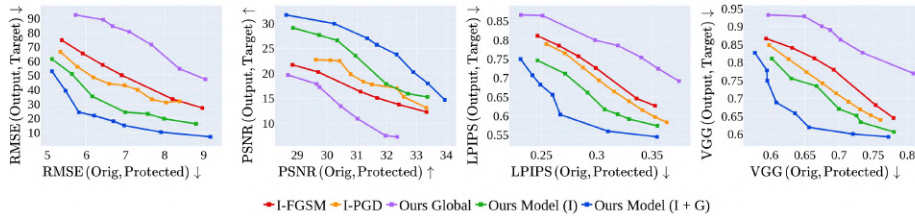
Fig. 6: Comparison with different optimization techniques evaluated on self-reconstruction (white target). We plot the output image quality (y-axis) corresponding to different levels of perturbations added to the image (x-axis). *Orig* and *Protected* refer to the original and protected image. *Output* 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.

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 on compressed images, Fig. 7 and 8. Training the model with fixed compression quality makes the perturbation robust to that specific compression quality; how-

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 [31]	17517.71 ms ($\pm$124.08 ms)
I-PGD [37]	17523.01 ms ($\pm$204.15 ms)
Ours	**10.66 ms** ($\pm$0.21 ms)
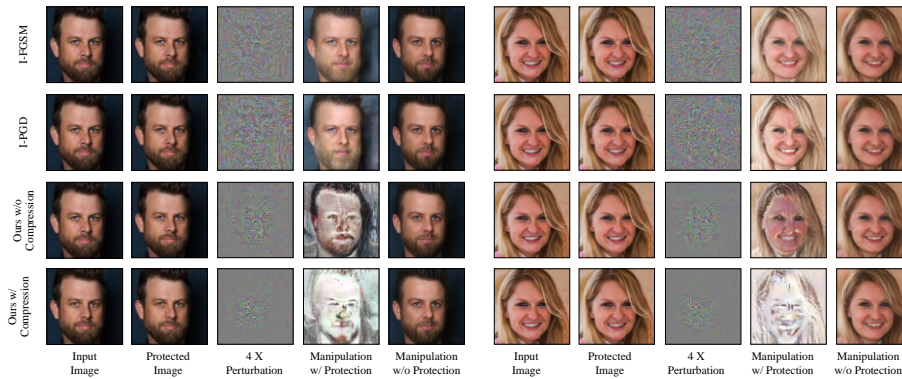
Fig. 7: Qualitative comparison in the presence of JPEG compression (white target). Methods trained without compression struggle; in contrast, our model trained with compression is able to produce perturbations that are robust to compression. *Ours w/ Compression* refers to the model trained with random 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.
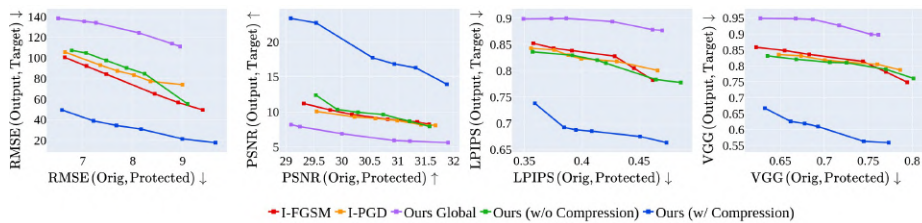
Fig. 8: Performance comparison in the presence of JPEG compression. Our method without differentiable JPEG training manages to disrupt the model; however, training with random compression levels significantly outperforms the uncompressed baselines. All methods are evaluated at compression quality C-80.

ever, it fails for other compression levels; see Fig. 9. We therefore train across different compression levels varied during training iterations.
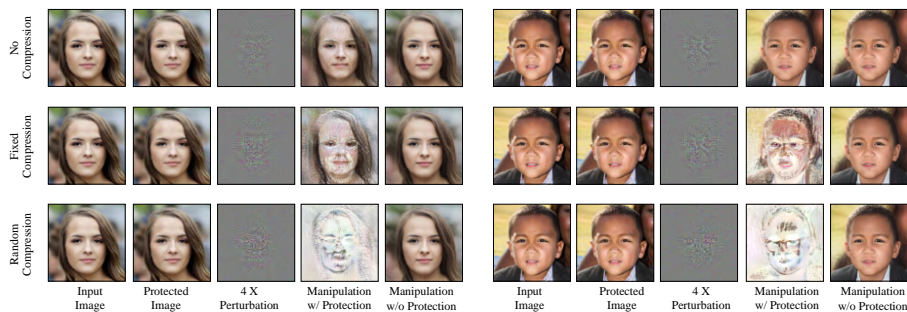
Fig. 9: Comparison for our method trained without compression, fixed compression, and random compression for self-reconstruction task (white target). The fixed compression model was trained with compression quality C-80. All methods are evaluated on compression quality C-30. The randomly compressed model 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 [46] with solid white image as the manipulation target, SimSwap [10] with solid blue image as target and StyleClip [44] 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.

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
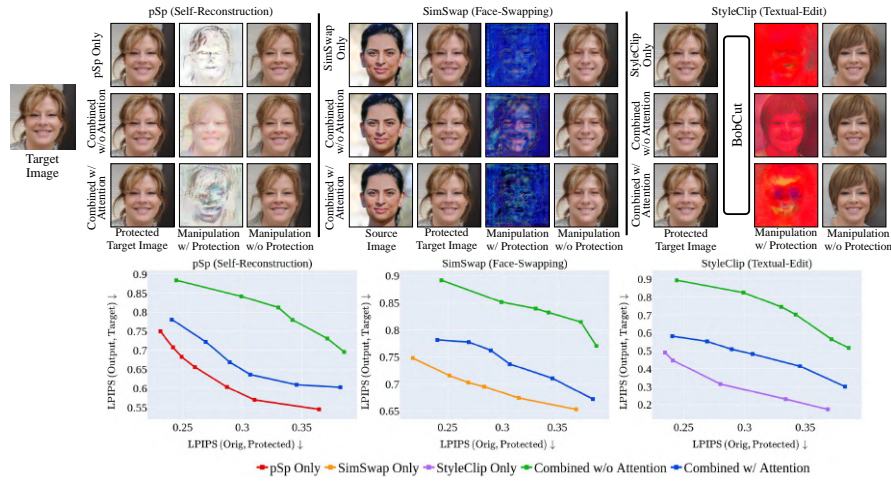
Fig. 10: Visual results (top) and performance graph (bottom) for multiple targets simultaneously. *pSp Only*, *SimSwap Only*, and *StyleClip Only* refer to the individual protection models trained only for the respective manipulations. *Combined w/o Attention* refers to a model trained directly for all 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.

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

Supplemental Material

In this supplemental document, we provide additional details. In the main paper, we demonstrate the effectiveness of our approach on popular face manipulation methods. The motivation behind choosing these specific manipulations is explained in Section 6. Current forensic methods aim to only identify the manipulated images, which suffers from the limitation of not being able to prevent malicious activities. However, it is practically more relevant to prevent creation of fake images using real images. We achieve this using targeted adversarial attacks, this is explained in Section 7. In Section 8, we provide details about the network architecture and additional training details. We then elaborate the implementation of the baseline methods in Section 9. Finally, in Section 10, we perform ablations and additional experiments.
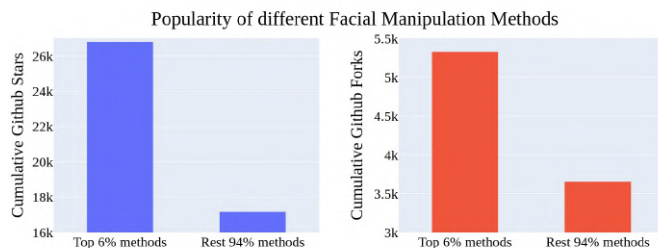
Fig. 11: Official github popularity distribution for open-source facial manipulation tools. Methods chosen from last five years' papers from top vision conferences. Note that only a handful of methods show wide applicability in a practical scenario (1.5K+ stars and 300+ forks per repo) and roughly 94% of all methods are not used very often.

6   Protection from popular Image Manipulations

We emphasize that it is important to protect images from the popular facial manipulations. Compared to large number of open-source image manipulation tools available online, we notice that very few methods have achieved wide applicability among users as shown in Fig. 11, thus attracting many users to employ only these popular methods. Secondly, re-training or even running inference on these methods requires specialized knowledge and skills, resulting in common users to look for pre-trained models in an easily accessible way. Therefore, we benchmark our results on these famous high quality facial manipulations that can easily be accessed via web demos, see Tab. 2. We additionally show in Subsection 10.6 that our generated perturbations can also protect against new and

unseen manipulations built upon these protected manipulations. A complete list of manipulation methods can be found here[2].

Method	Edit Type	# Runs (Web Demo)	URL	GPU Hours
pSp [46]	Style Mixing	5,704	https://bit.ly/3I7ew22	100
SimSwap [10]	FaceSwap	30,944	https://bit.ly/3i5mjmh	120
StyleClip [44]	Text-Driven Manipulation	374,636	https://bit.ly/3MIShmx	12 per style

Table 2: Training time and number of runs (as of 14.03.2022) on the web demo of publicly accessible pre-trained models for the widely used manipulation methods.

7 Definitions

7.1 Targeted Adversarial Attack

Adversarial attacks [2] are typically used in the context of classification tasks as a technique to fool the classifier model by maximizing the probability of outputting an incorrect target class label other than its actual class. In a similar spirit, we formalize targeted attacks for face image manipulation as a technique to trick these generative models; however, in contrast to misleading a classifier, our goal is to produce a predefined target image (uniformly colored solid white/blue/red images in our experiments).

Recently, Ruiz et al. [50] proposed the term *targeted disruption*. Their key aspect is to use a specific target image to drive the optimization such that it destroys the output of the generative model. We on the other side propose to learn a perturbation that forces the manipulation model to produce a specific target image instead of a random output image; we refer to this as a *targeted* adversarial attack.

8 Training Details

Dataset. To train our models we define custom split on high resolution FFHQ [27] dataset. We additionally evaluate the performance of our models on unseen Celeb-HQ [25] and VGGFace2-HQ [59] datasets in the main paper. We used 10K images for training and 1000 images for validation and test split each, see Tab. 3 for more details.

---

[2] https://bit.ly/35VSd1K

Task	Model	Mode	No. of Images	Dataset
Self-Reconstruction	pSp [46]	Train	5000	FFHQ [27]
		Val	1000	FFHQ [27]
Face-Swapping	SimSwap [10]	Train	2×5000	FFHQ [27]
		Val	2×1000	FFHQ [27]
Textual-Editing	StyleClip [44]	Train	5000	FFHQ [27]
		Val	1000	FFHQ [27]
Self-Reconstruction	pSp [46]	Test	1000	FFHQ [27]
Style-Mixing	pSp [46]	Test	2×500	FFHQ [27]
Face-Swapping	SimSwap [10]	Test	2×500	FFHQ [27]
Textual-Editing	StyleClip [44]	Test	1000	FFHQ [27]
Style-Mixing (Unseen Dataset)	pSp [46]	Test	2×500	Celeb-HQ [25]
		Test	2×500	VGGFace2-HQ [59]

Table 3: Dataset split for different tasks used in our experiments. For self-reconstruction, the manipulation model takes one image as input. For style-mixing and face-swapping, two images are fed as input to the manipulation model. For textual-editing, an image and a text prompt describing the manipulation-style is given an input to the model. For consistency, all methods are benchmarked on the same set of images (except unseen images).

Model Architecture. For all three model backones used in the paper ProtectionNet g_{Φ}, AttentionNet h_{ω}, and FusionNet r_{ρ}, we use a convolutional neural network based on U-Net [48] architecture. More specifically, we use UNet-64 architecture with 29.24M parameters, as shown in Fig. 12. The weights of the network are normal initialized with scaling factor of 0.02.

For ProtectionNet g_{Φ}, the network takes a 6 channel input, channel-wise concatenated original image $\mathbf{X}_i$ and the globally optimized perturbation δ_G to predict the image-specific perturbation δ_i. For AttentionNet h_{ω}, the network takes a 4 channel input, the image-specific perturbation δ_i and manipulation method label C_k and generates spatial attention map α_i^k. For FusionNet r_{ρ}, the model takes a 3 channel input, the blended model-specific perturbation with spatial attention maps and produces model-agnostic perturbation δ_i^{all}.

9 Baseline Implementations

We compare our method against adversarial attack baselines I-FGSM [31] and I-PGD [37]. These methods target classification tasks, where attack patterns are optimized for each image individually. To this end, we adapt their method for our
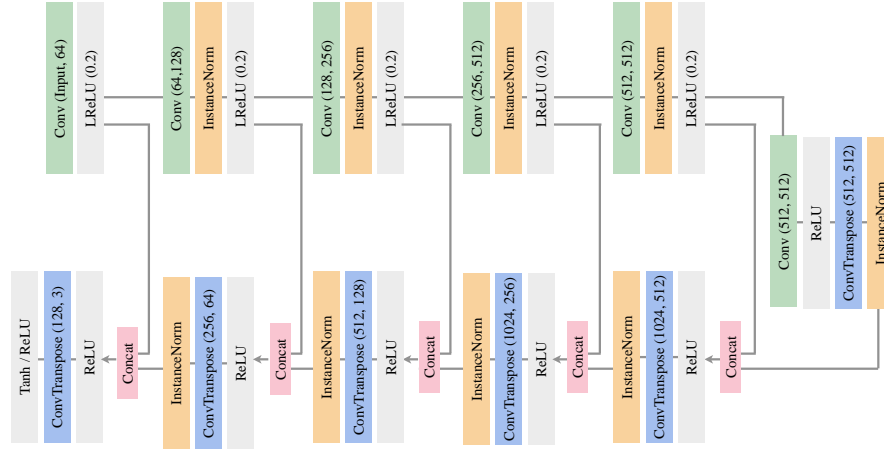
Fig. 12: Architecture Overview: Network architecture used for experiments. Conv(.) and ConvTranspose(.) refer to the 2D convolution and 2D transposed convolution operation. For both Conv and ConvTranspose, we use a kernel size of 4×4. For conv layers, values in bracket refers to the input and output channels; e.g., $\mathrm{Conv}(X, Y)$ denotes a X channel input and Y channel output. The input channel for the first Conv varies for different backbones: 6 for Protection-Net g_{Φ}, 4 for AttentionNet h_{ω} and 3 for FusionNet r_{ρ}. LReLU(0.2) denotes the LeakyReLU activation with a negative slope 0.2. Concat denotes the concatenation operation. For ProtectionNet g_{Φ} and FusionNet r_{ρ}, we use Tanh non-linearity after the last layer and ReLU for AttentionNet h_{ω}.

targeted adversarial attack task on generative models. Similar to their original implementations, we define a loss that only consists of the reconstruction term; i.e.,

$$\mathcal{L}^{\mathrm{Total}} = \mathcal{L}^{\mathrm{recon}} = \left\| f_{\Theta}(\mathbf{X}^p) - \mathbf{Y}^{\mathrm{target}} \right\|_2,$$

where $\mathbf{X}^p$ refers to the protected image and $\mathbf{Y}^{\mathrm{target}}$ refers to the predefined manipulation target (solid color white/blue image in our case). The amount of perturbation in the image can be controlled with the magnitude of the update step.

I-FGSM [31] (Iterative Fast Gradient Sign Method): The protected image is first initialized with the original image as:

$$\mathbf{X}_0^p := \mathbf{X},$$

and then updated iteratively as

$$\mathbf{X}_{n+1}^p = \mathrm{Clamp}_{\varepsilon} \left\{ \mathbf{X}_n^p - \alpha \, \mathrm{sign}(\nabla_X \mathcal{L}^{\mathrm{recon}}(\mathbf{X}_n^p, \mathbf{Y}^{\mathrm{target}})) \right\},$$

where α is the perturbation strength, $\text{Clamp}_\varepsilon(\xi)$ clips the higher and lower values in the valid range $[-\epsilon, \epsilon]$ and $\text{sign}(\zeta)$ returns the sign vector of ζ. We report results with 100 iterations.

I-PGD [37] Iterative Projected Gradient Descent): The protected image is obtained by the following update steps:

$$\mathbf{X}_0^p := \mathbf{X},$$

$$\mathbf{X}_{n+1}^p = \Pi_\mathcal{S}\left\{\mathbf{X}_n^p - \alpha\,\text{sign}(\nabla_X \mathcal{L}^{\text{recon}}(\mathbf{X}_n^p, \mathbf{Y}^{\text{target}}))\right\}$$

where α is the step size and $\Pi_\mathcal{S}(.)$ refers to the projection operator that projects onto the feasible set $\mathcal{S}$. We report results with 100 iterations and a step size of 0.01.

10   Additional Experiments

In this section, we report results for some additional experiments. We first show the performance of our model on unseen datasets in Subsection 10.1. Next we compare our method against some additional baselines in Subsection 10.2. We then show additional visuals for multiple manipulations simultaneously in Subsection 10.3. Next, we analyze the effect of incrementally adding more manipulation methods in Subsection 10.4. We then investigate the importance of AttentionNet backbone for blending manipulation-specific perturbations in Subsection 10.5. We also show that our generated perturbations can efficiently protect against related unseen manipulations in Subsection 10.6. We then analyze the distribution of different baselines in Subsection 10.7. Then, we report the performance of our method for different norms in Subsection 10.8. In Subsection 10.9, we evaluate the robustness of our method when compression is applied multiple times to the image. We show visual results for high perturbation levels in Subsection 10.10. Finally, in Subsection 10.11, we compare different round approximations for quantization step for the differentiable JPEG compression implementation. For brevity, we show results on self-reconstruction task with white image as manipulation target, unless otherwise stated.

10.1   Unseen Datasets

We further compare the performance of our method on two high-resolution unseen datasets, Celeb-HQ [25] and VGGFace2-HQ [59]. These datsets were neither seen during training or validation. We simply evaluate our trained models on these datasets. We show results on the style mixing task in Fig. 13. Note that our model demonstrates comparable performance on these unseen datasets compared to the seen FFHQ dataset [27].
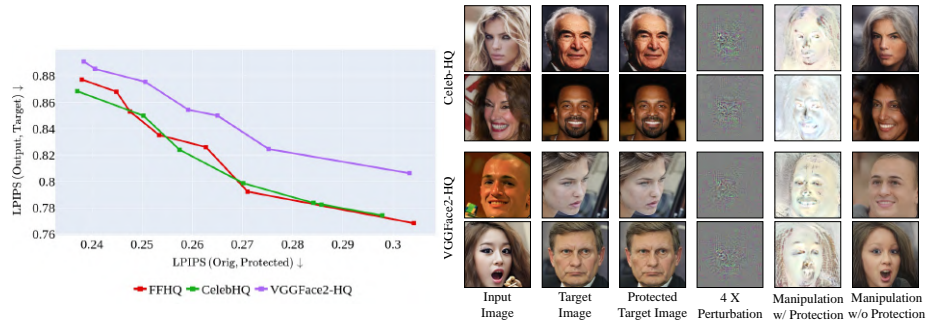
Fig. 13: Performance of unseen datasets on the style-mixing task with white manipulation target. The models are trained on FFHQ [27] and evaluated on unseen Celeb-HQ [25] and VGGFace2-HQ [59] datasets. The protection is applied to the target image. Perturbation is enlarged by a factor of four for better visibility. Our model demonstrates comparable performance on these unseen datasets compared to the seen FFHQ dataset [27].

10.2   Comparison with additional baselines

Here we compare our method against some additional baseline methods. ODI-PGD [56] (17751.72 ms) and APGD [15] (25786.96 ms) perform per-image optimization, and are therefore significantly slower; CMUA [23] is faster during inference; however, it produces lower-quality targeted disruptions compared to others. We outperform these baselines; see Fig. 14 and Fig. 15.

10.3   Additional Results for Multiple Manipulations Simultaneously

Due to space limitation in the main paper, we show additional visual results for generating manipulation-agnostic perturbations to protect against multiple manipulation methods at the same time, see Fig. 16.

10.4   Effect of adding more manipulation methods

We additionally analyze the effect of incrementally adding more manipulation methods using our proposed attention mechanism. We perform experiments with different combinations of manipulation methods. Results for pSp Encoder [46], SimSwap [10] and StyleClip [44] are shown in Fig. 17, 18 and 19 respectively. Note that gradually adding more methods using our proposed attention mechanism can be easily extended to handle multiple methods at the same time. This indicates that new manipulations can be easily integrated without significant deterioration in the overall performance on the individual manipulation(s).
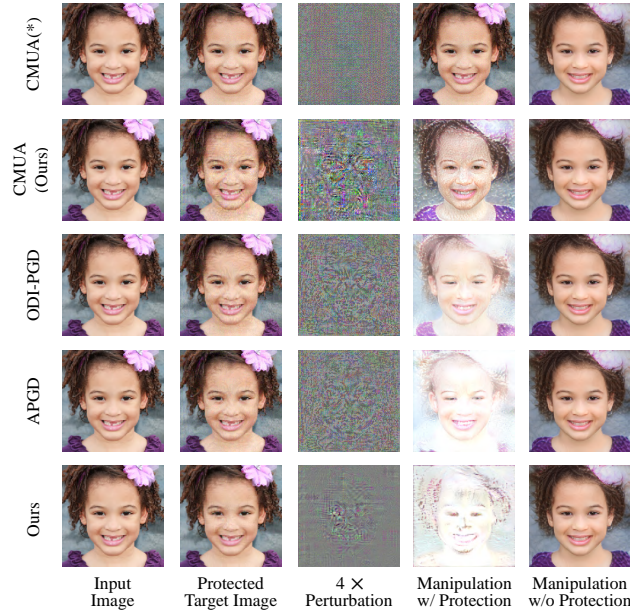
Fig. 14: Comparison on pSp (white target): CMUA(*) and CMUA (Ours) refers to their pretrained patterns and ours adapted.
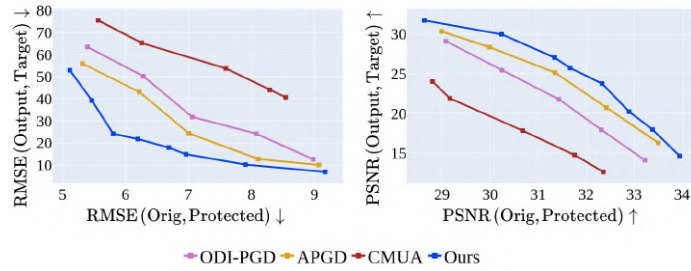

Fig. 15: Quantitative comparison with baselines on pSp (white target).

10.5   Analysis of AttentionNet Backbone

Next, we analyze the importance of attention network backbone to produce manipulation-agnostic perturbation when handling multiple manipulations at the same time. We notice that directly fusing the manipulation-specific perturbations by only using the FusionNet backbone leads to suboptimal results. Thus, the attention backbone serves as an important component to generate efficient patterns for all methods together. Results for this experiment are shown

Fig. 16: Visual results 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 than the no-attention baseline, and is comparable to individual models.

in Fig. 20. Note that this experiment differs from *Combined w/o Attention* baseline comparison shown in the main paper, which refers to directly learning a single model for multiple manipulations combined (without AttentionNet and FusionNet backbone). Here we compare blending of manipulation-specific perturbations using FusionNet only (without AttentionNet).

Method	# Runs	URL (Web Demo)
SAM [7]	18,205	https://bit.ly/3KG7F19
Style-NADA [21]	22,031	https://bit.ly/3J4Da4F

Table 4: Number of runs on the web demo (as of 14.03.2022) of publicly accessible pre-trained models for the unseen manipulation methods.
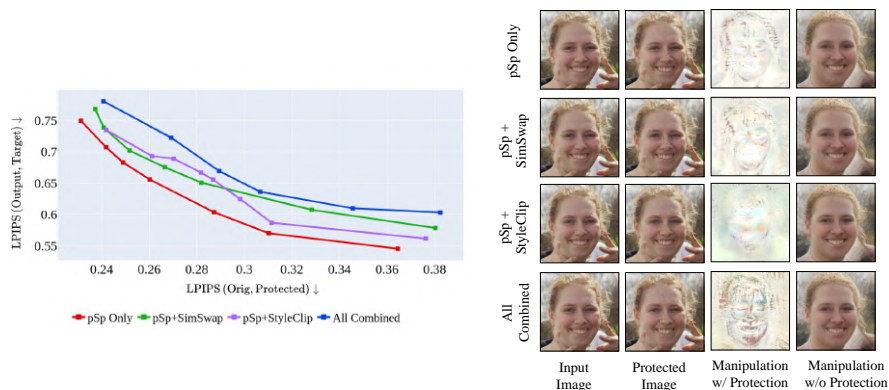
Fig. 17: Performance graph (left) and visual results (right) for pSp model [46] when incrementally adding multiple manipulations. *pSp Only* refers to the protection model trained only for pSp. *pSp + SimSwap* refers to a model trained for pSp [46] and SimSwap [10] manipulations combined. *pSp + StyleClip* refers to a model trained for pSp [46] and StyleClip [44] manipulations combined. *All Combined* refers to a model trained for all three manipulation methods combined. Our method can be easily extended to handle multiple manipulations.

10.6   Transferability on Related Models

We further show that our perturbations can even provide protection against unseen related methods build using existing methods. We show results on two different methods: (1) SAM [7], an age transformation model. (2) Style-NADA [21], a domain adaptation technique for image generators using only text prompt. Both of these methods are built upon pSp [46] to encoder images, which is protected with white target in our experiments. For SAM [7], we show results across different age distributions. For Style-NADA [21], we show results on several different manipulation models/styles. Visual results are shown in Fig. 21 and Fig. 22 and details on web demo are provided in Tab. 4.

Our applied protection can efficiently protect against both these unseen manipulations indicating that our generated perturbations can defend from newer methods build upon existing protected methods. Note that we do not train our model to protect against these new methods SAM [7] and Style-NADA [21]. We only evaluate the robustness of our generated perturbations on these models.

10.7   Analysis of different baselines

We analyze the output distribution of the manipulation model for different methods in Fig. 23. We notice that per-image optimization techniques, like I-PGD [37] show a number of outliers with high mean squared error in the generated output images, whereas our proposed method shows quite less variance. One such outlier is visualized in Fig. 24.
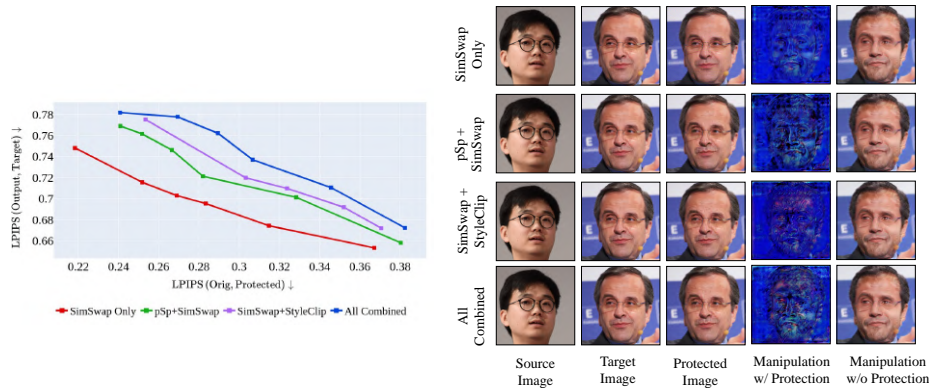
Fig. 18: Performance graph (left) and visual results (right) for SimSwap [10] when incrementally adding multiple manipulations. *SimSwap Only* refers to the protection model trained only for SimSwap. *pSp + SimSwap* refers to a model trained for pSp [46] and SimSwap [10] manipulations combined. *SimSwap + StyleClip* refers to a model trained for SimSwap [10] and StyleClip [44] manipulations combined. *All Combined* refers to a model trained for all three manipulation methods combined. Our method can be easily extended to handle multiple manipulations.

10.8   Ablations with different Norms: L_1, L_2, L_∞

Our loss function/minimization objective is formulated in terms of the L_2 norm. In this section, we compare results of our method with L_1 and L_∞ norm respectively. Visual results are shown in Fig. 25 and performance comparison in Fig. 26. We notice that L_2 norm significantly outperforms the other norms evenly distributing perturbation throughout the image making the changes in the image much less perceptually visible. In addition, it is more effective in erasing the image traces from the output of manipulation model.

10.9   Multi-level Compression

In a practical use case, an image might be compressed multiple times when shared on social media platforms. Therefore, the protection applied to images should be robust to consecutively applied compression. For simplicity, we show results for bi-level compression, i.e. the compression is applied twice to the image. We first apply a high compression (C-30) and thereafter a low compression (C-80) for evaluation. The second compression is applied to the first compressed image. Results are shown in Fig. 27. We show that our method is robust to multi-level compression as well.
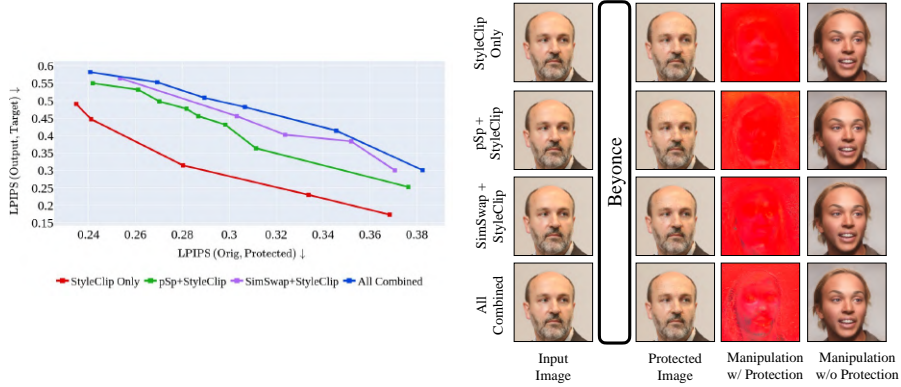
Fig. 19: Performance graph (left) and visual results (right) for StyleClip [44] when incrementally adding multiple manipulations. *StyleClip Only* refers to the protection model trained only for StyleClip. *pSp + StyleClip* refers to a model trained for pSp [46] and StyleClip [44] manipulations combined. *SimSwap + StyleClip* refers to a model trained for SimSwap [10] and StyleClip [44] manipulations combined. *All Combined* refers to a model trained for all three manipulation methods combined. Our method can be easily extended to handle multiple manipulations.
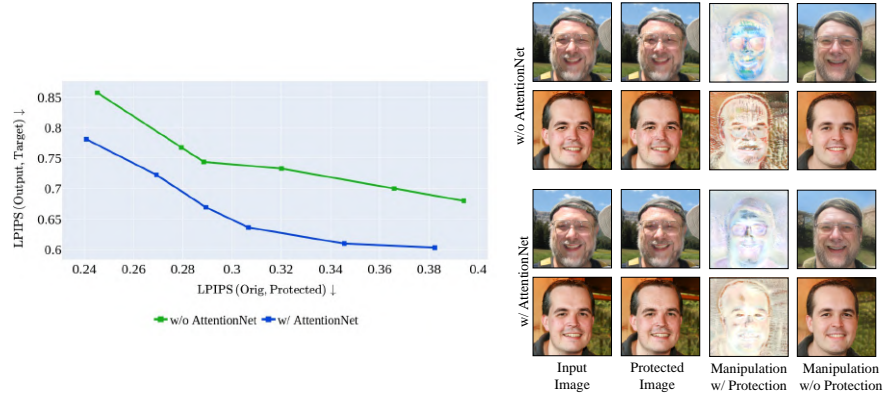
Fig. 20: Analysis of AttentionNet backbone. *w/o AttentionNet* refers to direct blending of manipulation-specific perturbations using FusionNet only, without attention. *w/ AttentionNet* refers to blending of manipulation-specific perturbations using both AttentionNet and FusionNet. *w/ AttentionNet* outperforms *w/o AttentionNet* baseline indicating that using attention efficiently blends manipulation-specific perturbations.
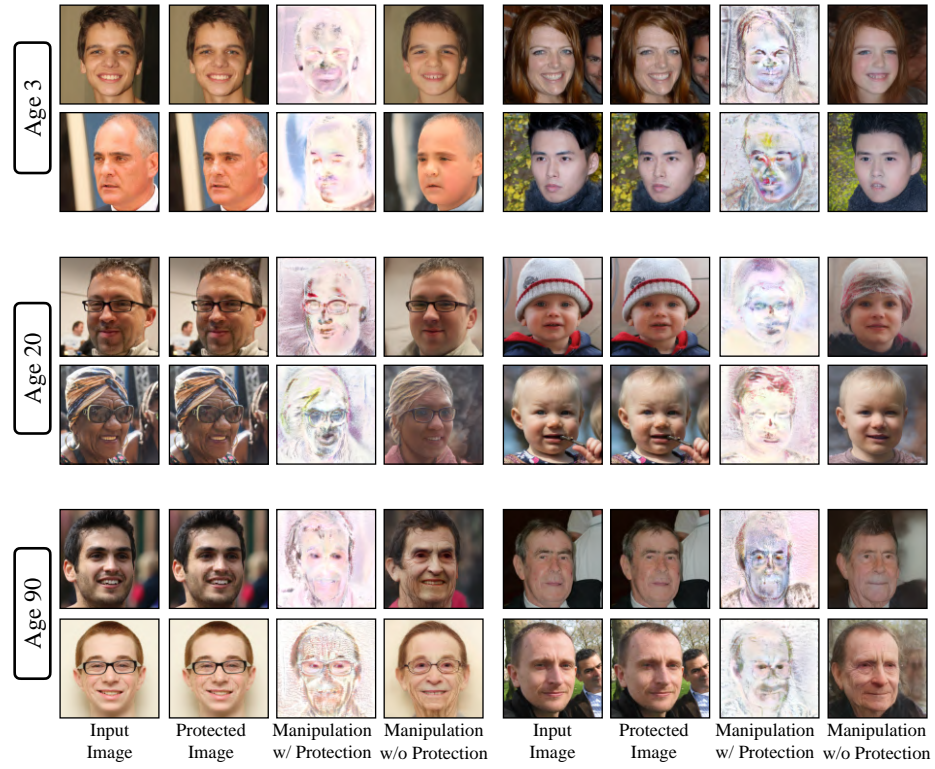
Fig. 21: Protection results on unseen Age Transformation Model SAM [7]. Our model trained on a related method pSp [46] can protect against this new age transformation manipulation. Note that our protection is robust to different age transformations.

10.10   Manipulation for higher perturbation levels

We show visual comparisons for lower perturbation levels since these are most useful for practical purposes. To this end, in the graphs shown in the main paper, we have analyzed our method at several different perturbation levels. In this section, we visualize our results for higher perturbation levels, which show more disturbance in the original images, but illustrated more significant disruptions for manipulation model predictions. Fig. 28 shows the visual results for the self-reconstruction task.
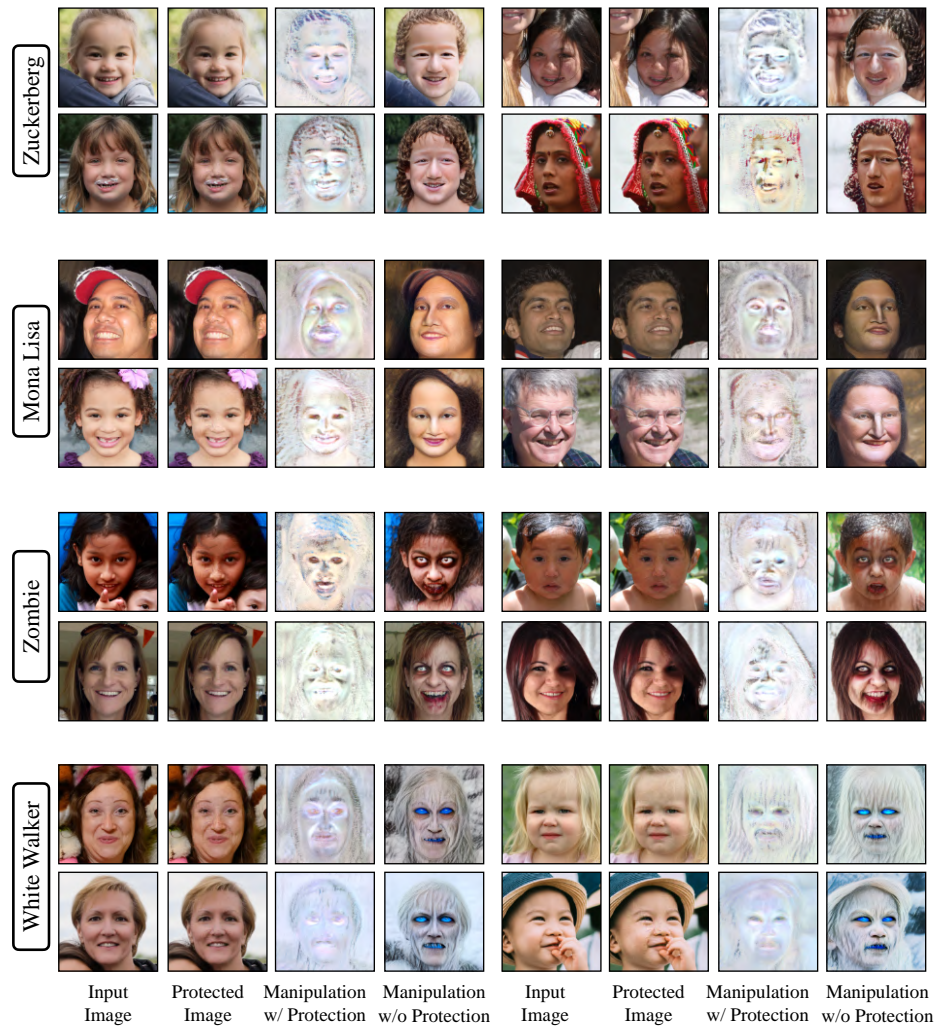
Fig. 22: Protection results on unseen models from Style-NADA [21] with different manipulation types. Our model trained on a related method pSp [46] can protect against these new image manipulations. Our generated perturbation can protect against all the different variants of manipulations generated by different Style-NADA models.
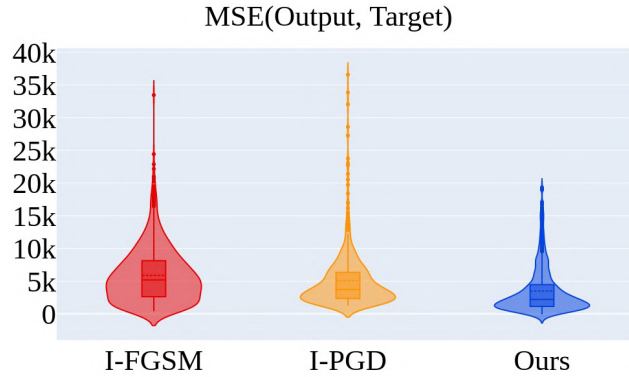
Fig. 23: Results for the self-reconstruction task [46] with white image as manipulation target. We show violin plots for different methods visualizing the mean squared error of the manipulation model output with the predefined manipulation target image. Our method, denoted as Ours, outperforms alternate methods showing a lot less variance in the output distribution. In contrast, per-image optimization techniques such as I-PGD have long tail distribution indicating that the method is not equally effective for all the samples in the dataset.
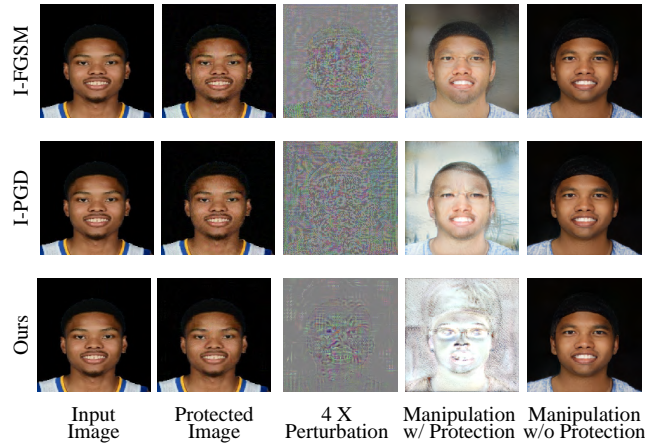
Fig. 24: Outlier case for the self-reconstruction task [46] with white image as manipulation target. In some cases, I-FGSM [31] and I-PGD [37] produce outliers with high mean squared error; cf. the high variance compared to our method in Fig. 23.
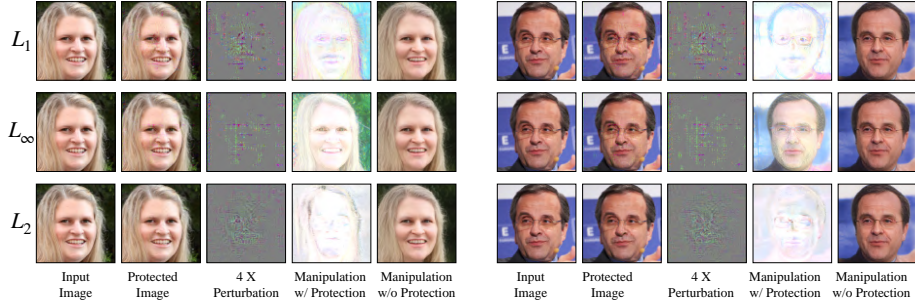
Fig. 25: Results for self-reconstruction task with white image as manipulation target. Comparison for three different norms used for the loss function: L_1, L_2 and L_∞. L_2 norm evenly distributes the perturbation over the image making it less perceptually visible. Also, the output images are more similar to white manipulation target.
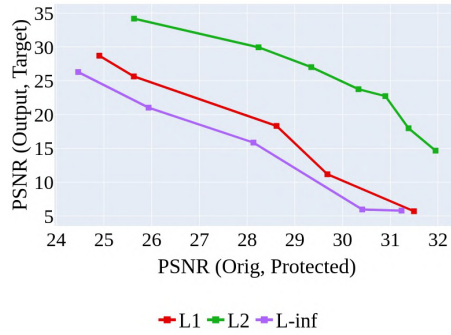

Fig. 26: Results for self-reconstruction task with white image as manipulation target. Performance comparison on PSNR graph for different norms. Orig and Protected refer to the original and protected image. Output refers to the output of the manipulation model and Target indicates predefined manipulation target. Note that L_2 norm outperforms L_1 and L_∞ based formulations.

10.11   Ablations with different JPEG approximations

There are three different approaches to approximate the round operation used in the quantization step of the original JPEG compression technique. These are formalized below.

1. Cubic Approximation [51]

$$x := \lfloor x \rceil + \left(x - \lfloor x \rceil \right)^3.$$

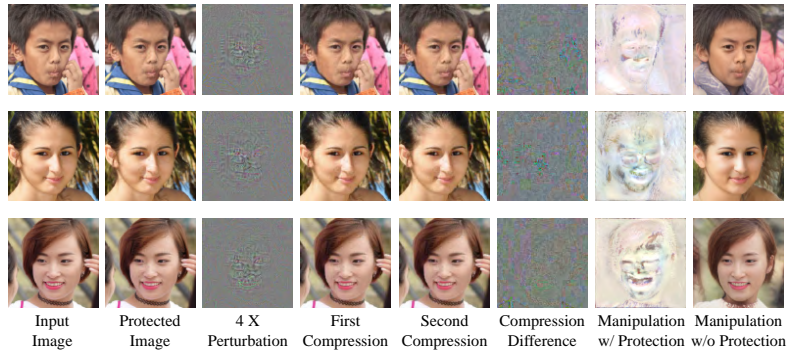| Input Image | Protected Image | 4 X Perturbation | First Compression | Second Compression | Compression Difference | Manipulation w/ Protection | Manipulation w/o Protection |

Fig. 27: Results for the self-reconstruction task with white image as manipulation target and for multi-level compression. First Compression denotes the first applied high compression C-30, Second Compression denotes the second compression (C-80) applied to the first compressed image. We apply different compression to evaluate robustness. Compression Difference denotes the difference between first and second compressed images amplified by 20X for visibility. Our method can efficiently handle multiple compression levels of different qualities.

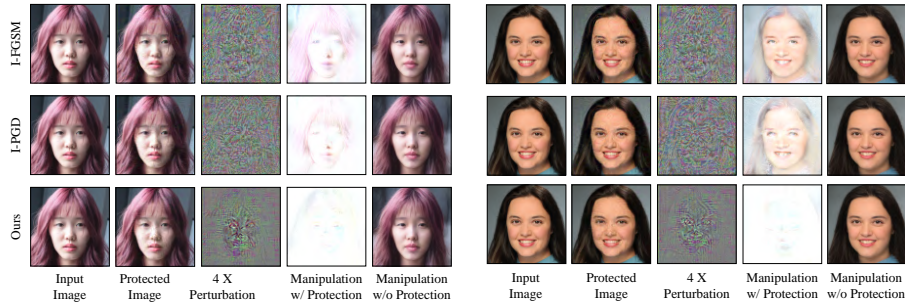| Input Image | Protected Image | 4 X Perturbation | Manipulation w/ Protection | Manipulation w/o Protection | Input Image | Protected Image | 4 X Perturbation | Manipulation w/ Protection | Manipulation w/o Protection |

Fig. 28: Visual results for higher perturbation levels for the self-reconstruction task [46] with white image as manipulation target. Even for a higher perturbation level, our proposed approach outperforms alternate methods.

2. Soft Approximation [30]

$$\tilde{x} = x - \frac{\sin(2\pi x)}{2\pi},$$

$$x := \big[\,\mathrm{round}(x) - \tilde{x}\,\big]_{\mathrm{detach}} + \tilde{x},$$

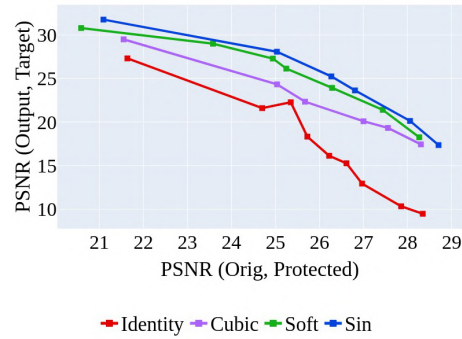where "detach" indicates that no gradients will be propagated during back-propagation.

Fig. 29: PSNR performance for training our method with different approximations to round operation for JPEG compression for the self-reconstruction task [46] with white image as manipulation target. Sin approximation outperforms other approximations with comparable results for Soft approximation.

3. Sin Approximation [30]

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

We also compare against the identity operation $x := x$ as the baseline. The performance comparison for these methods is visualized in Fig. 29. We notice that Sin approximation is better in performance compared to other approximations, therefore we use it for experiments in the main paper.

References

1. Maisy Kinsley fake account. https://twitter.com/sokane1/status/1111023838467362816, accessed: 2019-03-27 2

2. Targeted Adversarial Attacks. https://www.kaggle.com/c/nips-2017-targeted-adversarial-attack, accessed: 2017-07-01 16

3. 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

4. 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

5. 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

6. Agarwal, S., Farid, H., Gu, Y., He, M., Nagano, K., Li, H.: Protecting world leaders against deep fakes. In: CVPR Workshops (2019) 3

7. Alaluf, Y., Patashnik, O., Cohen-Or, D.: Only a matter of style: Age transformation using a style-based regression model. ACM Trans. Graph. **40**(4) (2021), https://doi.org/10.1145/3450626.3459805 22, 23, 26

8. Aneja, S., Nießner, M.: Generalized Zero and Few-Shot Transfer for Facial Forgery Detection. In: ArXiv preprint arXiv:2006.11863 (2020) 2, 3

9. 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

10. 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, 16, 17, 20, 23, 24, 25

11. 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

12. 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

13. 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

14. Cozzolino, D., Thies, J., Rössler, A., Riess, C., Nießner, M., Verdoliva, L.: Forensic-transfer: Weakly-supervised domain adaptation for forgery detection. arXiv (2018) 2, 3

15. Croce, F., Hein, M.: Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In: Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event. Proceedings of Machine Learning Research, vol. 119, pp. 2206–2216. PMLR (2020), http://proceedings.mlr.press/v119/croce20b.html 20

16. Dolhansky, B., Bitton, J., Pflaum, B., Lu, J., Howes, R., Wang, M., Ferrer, C.C.: The deepfake detection challenge (dfdc) dataset (2020) 2

17. 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

18. 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

19. Fischer, V., Kumar, M.C., Metzen, J.H., Brox, T.: Adversarial examples for semantic image segmentation (2017) 3

20. 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

21. Gal, R., Patashnik, O., Maron, H., Chechik, G., Cohen-Or, D.: Stylegan-nada: Clip-guided domain adaptation of image generators (2021) 22, 23, 27

22. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples (2015) 2, 3

23. Huang, H., Wang, Y., Chen, Z., Zhang, Y., Li, Y., Tang, Z., Chu, W., Chen, J., Lin, W., Ma, K.K.: Cmua-watermark: A cross-model universal adversarial watermark for combating deepfakes. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 36, pp. 989–997 (2022) 20

24. Huang, Q., Zhang, J., Zhou, W., WeimingZhang, Yu, N.: Initiative defense against facial manipulation (2021) 2, 4

25. 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, 16, 17, 19, 20

26. Karras, T., Aittala, M., Laine, S., Härkönen, E., Hellsten, J., Lehtinen, J., Aila, T.: Alias-free generative adversarial networks. In: Proc. NeurIPS (2021) 3

27. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks (2019) 3, 9, 16, 17, 19, 20

28. 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

29. 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

30. 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, 30, 31

31. Kurakin, A., Goodfellow, I.J., Bengio, S.: Adversarial machine learning at scale. ArXiv **abs/1611.01236** (2017) 9, 10, 12, 17, 18, 28

32. 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

33. Li, Y., Lyu, S.: Exposing deepfake videos by detecting face warping artifacts (2018) 2

34. 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

35. 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
36. 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
37. 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, 17, 19, 23, 28
38. 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
39. 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
40. 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
41. 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
42. 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
43. 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
44. 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, 16, 17, 20, 23, 24, 25
45. 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
46. 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, 16, 17, 20, 23, 24, 25, 26, 27, 28, 30, 31
47. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: MICCAI (2015) 6
48. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation (2015), http://arxiv.org/abs/1505.04597, cite arxiv:1505.04597Comment: conditionally accepted at MICCAI 2015 17
49. 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
50. Ruiz, N., Bargal, S.A., Sclaroff, S.: Disrupting deepfakes: Adversarial attacks against conditional image translation networks and facial manipulation systems (2020) 2, 4, 16

51. Shin, R.: Jpeg-resistant adversarial images (2017) 8, 29
52. 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
53. 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
54. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R.: Intriguing properties of neural networks (2014) 2, 3
55. Tancik, M., Mildenhall, B., Ng, R.: Stegastamp: Invisible hyperlinks in physical photographs. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2020) 4
56. Tashiro, Y., Song, Y., Ermon, S.: Diversity can be transferred: Output diversification for white- and black-box attacks. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H. (eds.) Advances in Neural Information Processing Systems. vol. 33, pp. 4536–4548. Curran Associates, Inc. (2020), https://proceedings.neurips.cc/paper/2020/file/30da227c6b5b9e2482b6b221c711edfd-Paper.pdf 20
57. 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
58. Thies, J., Zollhöfer, M., Nießner, M.: Deferred neural rendering: Image synthesis using neural textures. ACM Transactions on Graphics 2019 (TOG) (2019) 3
59. VGGFace2-HQ: Vggface2-hq (2020), https://github.com/NNNNAI/VGGFace2-HQ 16, 17, 19, 20
60. Wallace, G.K.: The jpeg still picture compression standard. IEEE transactions on consumer electronics **38**(1), xviii–xxxiv (1992) 7
61. Wang, R., Juefei-Xu, F., Luo, M., Liu, Y., Wang, L.: Faketagger: Robust safeguards against deepfake dissemination via provenance tracking (2021) 4
62. 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
63. 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
64. 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
65. Yang, Y., Liang, C., He, H., Cao, X., Gong, N.Z.: Faceguard: Proactive deepfake detection (2021) 2, 4
66. 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
67. 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

68. Zakharov, E., Shysheya, A., Burkov, E., Lempitsky, V.: Few-shot adversarial learning of realistic neural talking head models (2019) 3
69. 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
70. 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
71. Zhu, J., Kaplan, R., Johnson, J., Fei-Fei, L.: Hidden: Hiding data with deep networks. In: ECCV (2018) 4