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

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 1. 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 2. In Section 3, we provide details about the network architecture and additional training details. We then elaborate the implementation of the baseline methods in Section 4. Finally, in Section 5, we perform ablations and additional experiments.

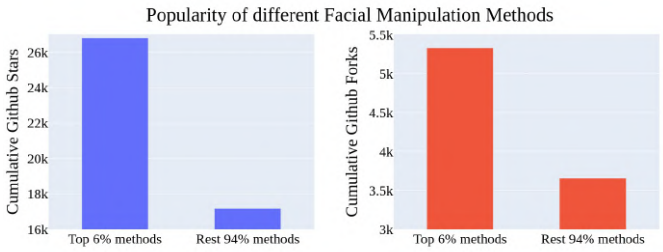

Fig. 1: Official github popularity distribution for open-source facial manipulation tools. Methods chosen from last five years' papers from top vision conferences. Note that only a handful of methods show wide applicability in a practical scenario (1.5K+ stars and 300+ forks per repo) and roughly 94% of all methods are not used very often.

1 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. 1, 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. 1. We additionally show in Subsection 5.6 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[1].

Method	Edit Type	# Runs (Web Demo)	URL	GPU Hours
pSp [13]	Style Mixing	5,704	https://bit.ly/3I7ew22	100
SimSwap [3]	FaceSwap	30,944	https://bit.ly/3i5mjmh	120
StyleClip [12]	Text-Driven Manipulation	374,636	https://bit.ly/3MIShmx	12 per style

Table 1: 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.

2 Definitions

2.1 Targeted Adversarial Attack

Adversarial attacks [1] 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. [15] 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

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

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.

3 Training Details

Dataset. To train our models we define custom split on high resolution FFHQ [8] dataset. We additionally evaluate the performance of our models on unseen Celeb-HQ [7] and VGGFace2-HQ [18] datasets in the main paper. We used 10K images for training and 1000 images for validation and test split each, see Tab. 2 for more details.

Task	Model	Mode	No. of Images	Dataset
Self-Reconstruction	pSp [13]	Train	5000	FFHQ [8]
		Val	1000	FFHQ [8]
Face-Swapping	SimSwap [3]	Train	2×5000	FFHQ [8]
		Val	2×1000	FFHQ [8]
Textual-Editing	StyleClip [12]	Train	5000	FFHQ [8]
		Val	1000	FFHQ [8]
Self-Reconstruction	pSp [13]	Test	1000	FFHQ [8]
Style-Mixing	pSp [13]	Test	2×500	FFHQ [8]
Face-Swapping	SimSwap [3]	Test	2×500	FFHQ [8]
Textual-Editing	StyleClip [12]	Test	1000	FFHQ [8]
Style-Mixing (Unseen Dataset)	pSp [13]	Test	2×500	Celeb-HQ [7]
		Test	2×500	VGGFace2-HQ [18]

Table 2: 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 $\boldsymbol{g_\Phi}$, AttentionNet $\boldsymbol{h_\omega}$, and FusionNet $\boldsymbol{r_\rho}$, we use a convolutional neural network based on U-Net [14] architecture. More specifically, we use UNet-64 architecture with 29.24M parameters, as shown in Fig. 2. The weights of the network are normal initialized with scaling factor of 0.02.

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

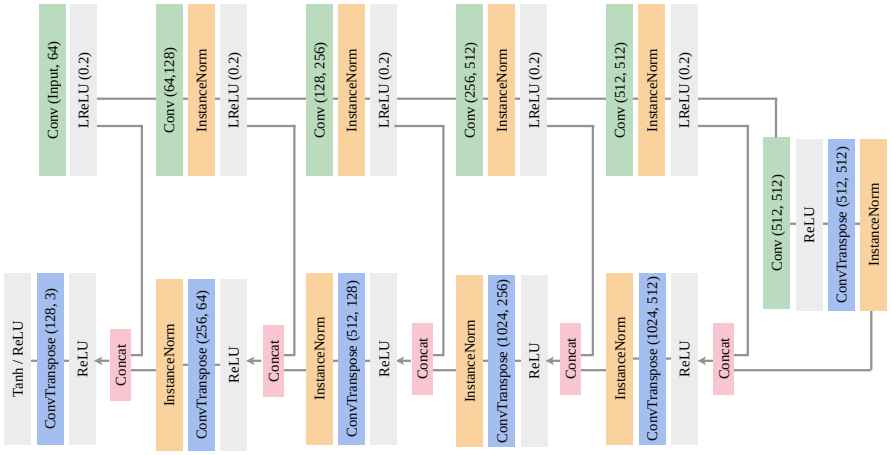

Fig. 2: Architecture Overview: Network architecture used for experiments. Conv(.) and ConvTranspose(.) refer to the 2D convolution and 2D transposed convolution operation. For both Conv and ConvTranspose, we use a kernel size of 4×4. For conv layers, values in bracket refers to the input and output channels; e.g., 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 ProtectionNet $\boldsymbol{g_\Phi}$, 4 for AttentionNet $\boldsymbol{h_\omega}$ and 3 for FusionNet $\boldsymbol{r_\rho}$. LReLU(0.2) denotes the LeakyReLU activation with a negative slope 0.2. Concat denotes the concatenation operation. For ProtectionNet $\boldsymbol{g_\Phi}$ and FusionNet $\boldsymbol{r_\rho}$, we use Tanh non-linearity after the last layer and ReLU for AttentionNet $\boldsymbol{h_\omega}$.

4 Baseline Implementations

We compare our method against adversarial attack baselines I-FGSM [10] and I-PGD [11]. These methods target classification tasks, where attack patterns are optimized for each image individually. To this end, we adapt their method for our 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.,

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

where $\mathbf{X}^p$ refers to the protected image and $\mathbf{Y}^{\text{target}}$ 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 [10] (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

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

where α is the 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 [11] 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.

5 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 5.1. Next we compare our method against some additional baselines in Subsection 5.2. We then show additional visuals for multiple manipulations simultaneously in Subsection 5.3. Next, we analyze the effect of incrementally adding more manipulation methods in Subsection 5.4. We then investigate the importance of AttentionNet backbone for blending manipulation-specific perturbations in Subsection 5.5. We also show that our generated perturbations can efficiently protect against related unseen manipulations in Subsection 5.6. We then analyze the distribution of different baselines in Subsection 5.7. Then, we report the performance of our method for different norms in Subsection 5.8. In Subsection 5.9, 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 5.10. Finally, in Subsection 5.11, 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.

5.1 Unseen Datasets

We further compare the performance of our method on two high-resolution unseen datasets, Celeb-HQ [7] and VGGFace2-HQ [18]. 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. 3. Note that our model demonstrates comparable performance on these unseen datasets compared to the seen FFHQ dataset [8].

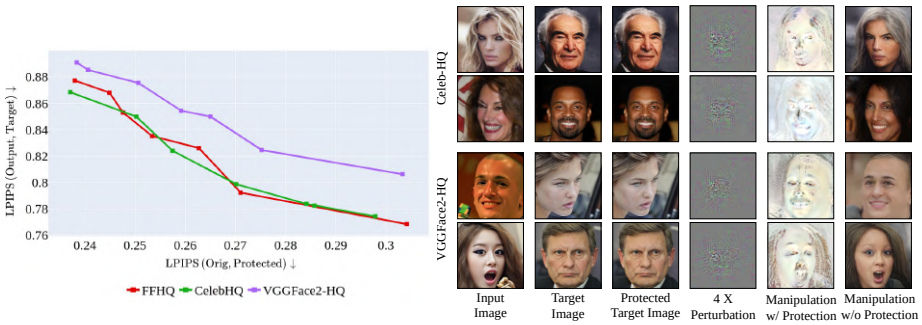

Fig. 3: Performance of unseen datasets on the style-mixing task with white manipulation target. The models are trained on FFHQ [8] and evaluated on unseen Celeb-HQ [7] and VGGFace2-HQ [18] datasets. The protection is applied to the target image. Perturbation is enlarged by a factor of four for better visibility. Our model demonstrates comparable performance on these unseen datasets compared to the seen FFHQ dataset [8].

5.2 Comparison with additional baselines

Here we compare our method against some additional baseline methods. ODI-PGD [17] (17751.72 ms) and APGD [4] (25786.96 ms) perform per-image optimization, and are therefore significantly slower; CMUA [6] is faster during inference; however, it produces lower-quality targeted disruptions compared to others. We outperform these baselines; see Fig. 4 and Fig. 5.

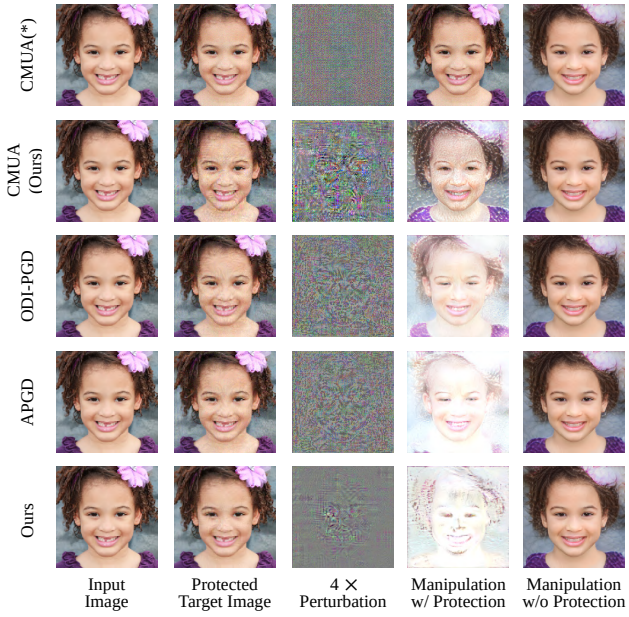

Fig. 4: Comparison on pSp (white target): CMUA(*) and CMUA (Ours) refers to their pretrained patterns and ours adapted.

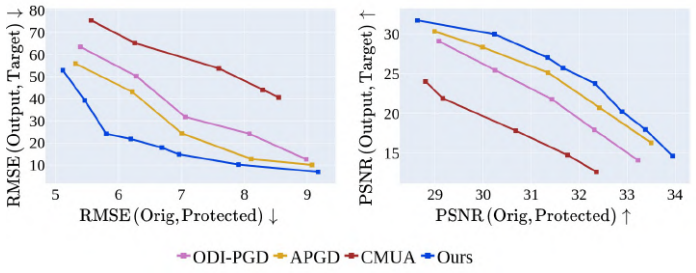

Fig. 5: Quantitative comparison with baselines on pSp (white target).

5.3 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. 6.

Fig. 6: 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.

5.4 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 [13], SimSwap [3] and StyleClip [12] are shown in Fig. 7, 8 and 9 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).

5.5 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 in Fig. 10. Note that this experiment differs from *Combined w/o Attention* baseline comparison shown in the main paper, which refers to directly learning a

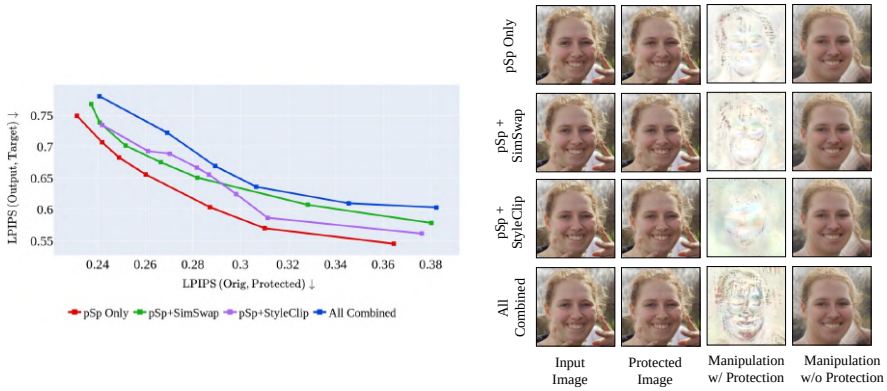

Fig. 7: Performance graph (left) and visual results (right) for pSp model [13] when incrementally adding multiple manipulations. *pSp Only* refers to the protection model trained only for pSp. *pSp + SimSwap* refers to a model trained for pSp [13] and SimSwap [3] manipulations combined. *pSp + StyleClip* refers to a model trained for pSp [13] and StyleClip [12] 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.

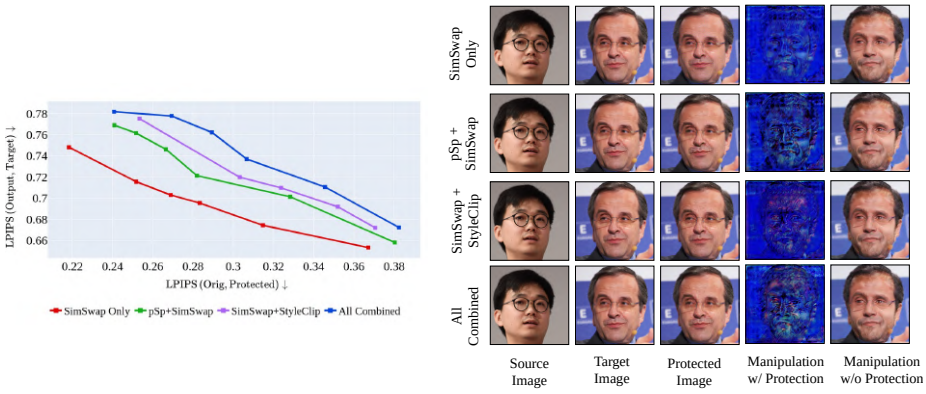

Fig. 8: Performance graph (left) and visual results (right) for SimSwap [3] when incrementally adding multiple manipulations. *SimSwap Only* refers to the protection model trained only for SimSwap. *pSp + SimSwap* refers to a model trained for pSp [13] and SimSwap [3] manipulations combined. *SimSwap + StyleClip* refers to a model trained for SimSwap [3] and StyleClip [12] 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.

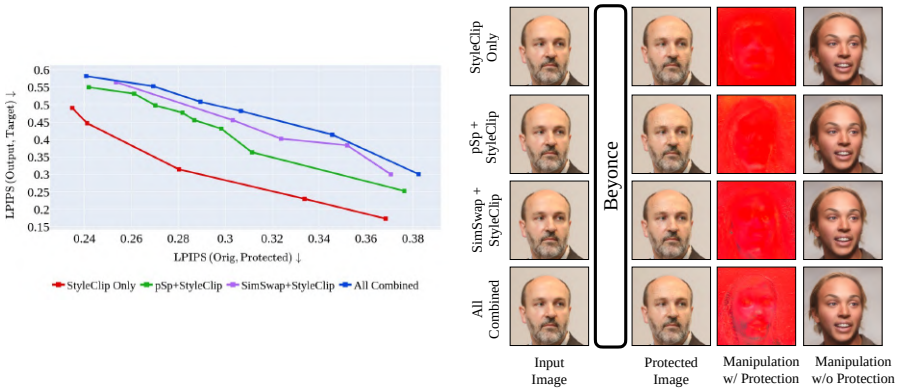

Fig. 9: Performance graph (left) and visual results (right) for StyleClip [12] when incrementally adding multiple manipulations. *StyleClip Only* refers to the protection model trained only for StyleClip. *pSp + StyleClip* refers to a model trained for pSp [13] and StyleClip [12] manipulations combined. *SimSwap + StyleClip* refers to a model trained for SimSwap [3] and StyleClip [12] 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.

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 [2]	18,205	`https://bit.ly/3KG7F19`
Style-NADA [5]	22,031	`https://bit.ly/3J4Da4F`

Table 3: Number of runs on the web demo (as of 14.03.2022) of publicly accessible pre-trained models for the unseen manipulation methods.

5.6 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 [2], an age transformation model. (2) Style-NADA [5], a domain adaptation technique for image generators using only text prompt. Both of these methods are built upon pSp [13] to encoder images, which is protected

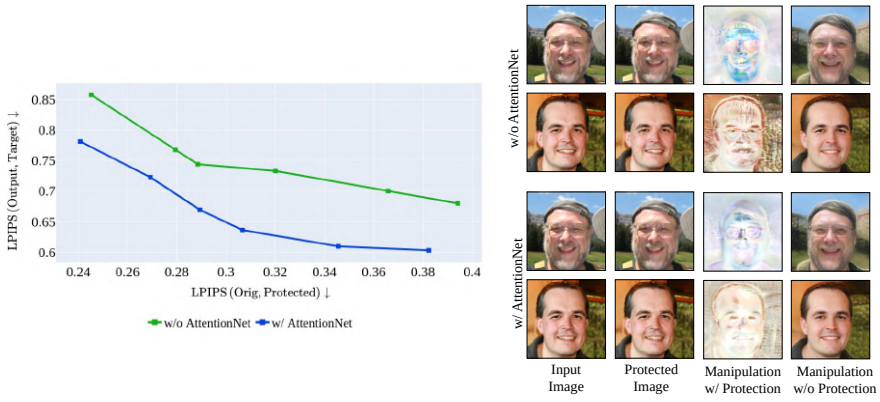

Fig. 10: Analysis of AttentionNet backbone. *w/o AttentionNet* refers to direct blending of manipulation-specific perturbations using FusionNet only, without attention. *w/ AttentionNet* refers to blending of manipulation-specific perturbations using both AttentionNet and FusionNet. *w/ AttentionNet* outperforms *w/o AttentionNet* baseline indicating that using attention efficiently blends manipulation-specific perturbations.

with white target in our experiments. For SAM [2], we show results across different age distributions. For Style-NADA [5], we show results on several different manipulation models/styles. Visual results are shown in Fig. 11 and Fig. 12 and details on web demo are provided in Tab. 3.

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 [2] and Style-NADA [5]. We only evaluate the robustness of our generated perturbations on these models.

5.7 Analysis of different baselines

We analyze the output distribution of the manipulation model for different methods in Fig. 13. We notice that per-image optimization techniques, like I-PGD [11] 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. 14.

5.8 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. 15 and performance comparison in

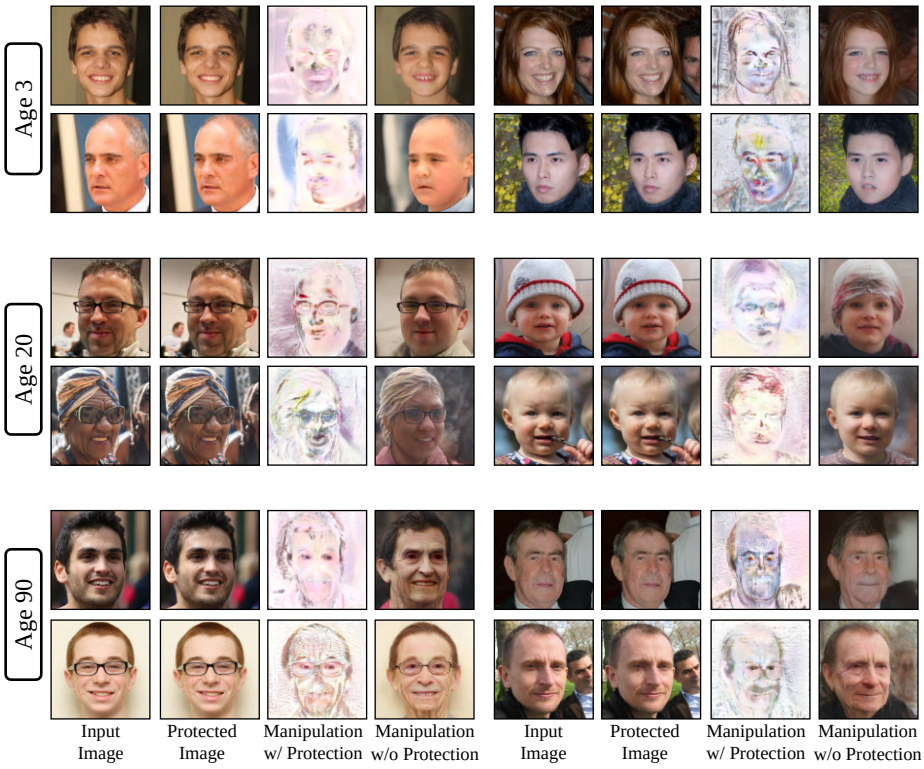

Fig. 11: Protection results on unseen Age Transformation Model SAM [2]. Our model trained on a related method pSp [13] can protect against this new age transformation manipulation. Note that our protection is robust to different age transformations.

Fig. 16. 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.

5.9 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.

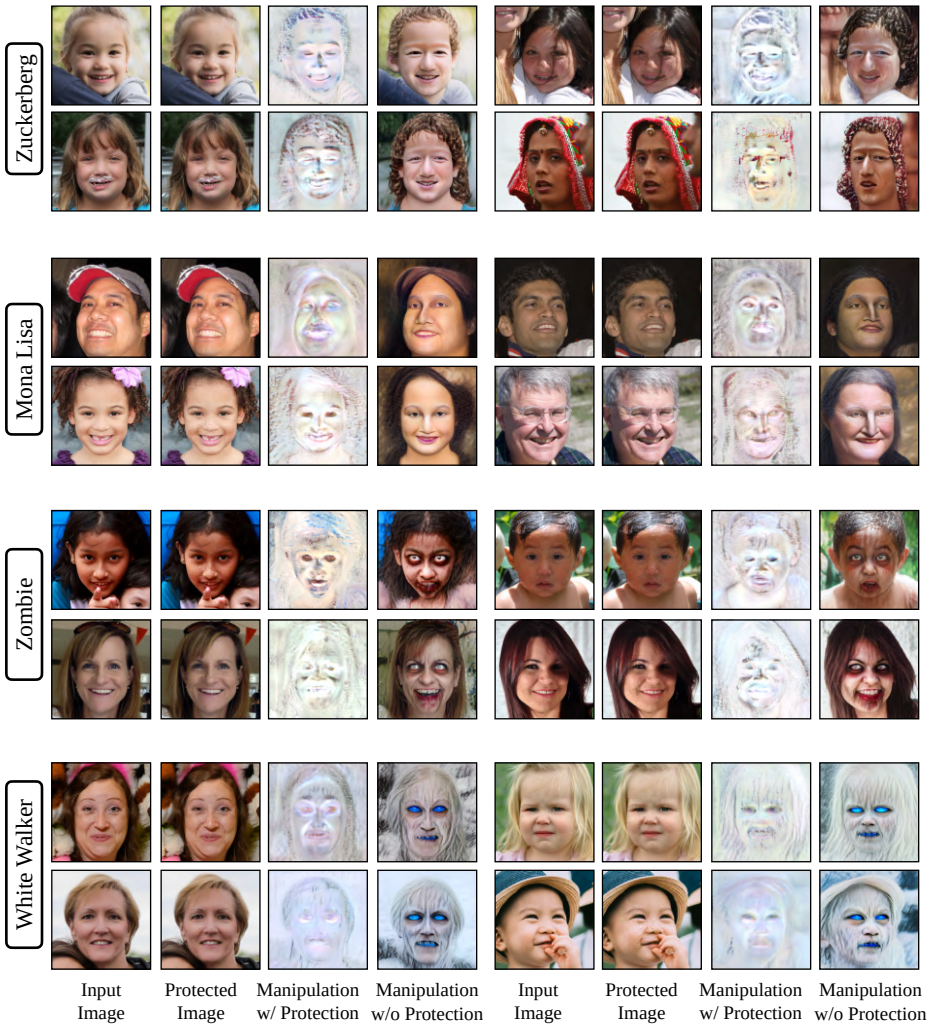

Fig. 12: Protection results on unseen models from Style-NADA [5] with different manipulation types. Our model trained on a related method pSp [13] can protect against these new image manipulations. Our generated perturbation can protect against all the different variants of manipulations generated by different Style-NADA models.

Results are shown in Fig. 17. We show that our method is robust to multi-level compression as well.

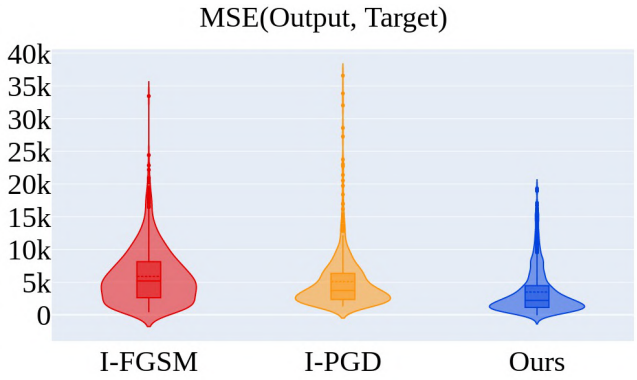

Fig. 13: Results for the self-reconstruction task [13] with white image as manipulation target. We show violin plots for different methods visualizing the mean squared error of the manipulation model output with the predefined manipulation target image. Our method, denoted as Ours, outperforms alternate 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.

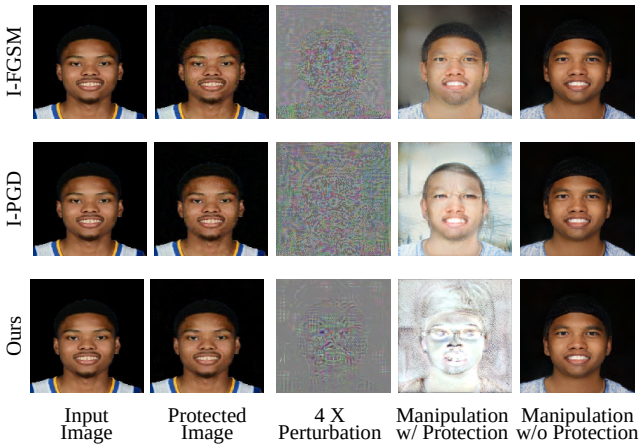

Fig. 14: Outlier case for the self-reconstruction task [13] with white image as manipulation target. In some cases, I-FGSM [10] and I-PGD [11] produce outliers with high mean squared error; cf. the high variance compared to our method in Fig. 13.

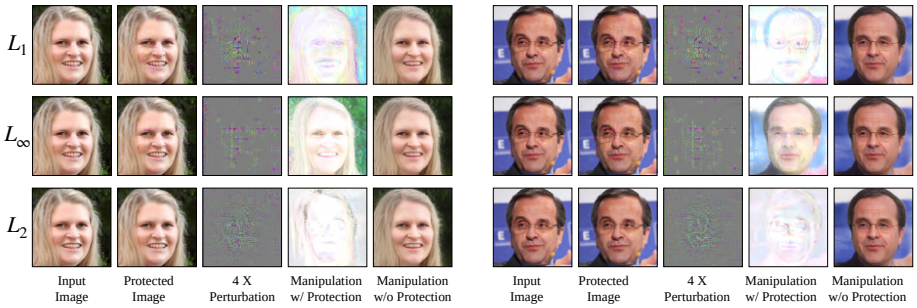

Fig. 15: Results for self-reconstruction task with white image as manipulation target. Comparison for three different norms used for the loss function: L_1, L_2 and L_∞. L_2 norm evenly distributes the perturbation over the image making it less perceptually visible. Also, the output images are more similar to white manipulation target.

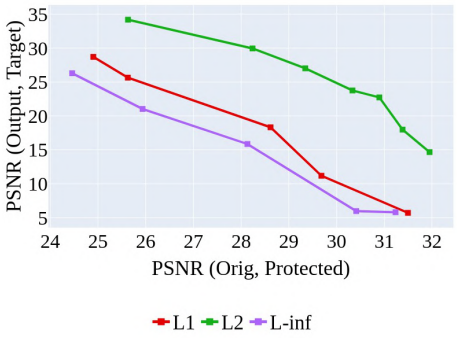

Fig. 16: Results for self-reconstruction task with white image as manipulation target. Performance comparison on PSNR graph for different norms. Orig and Protected refer to the original and protected image. Output refers to the output of the manipulation model and Target indicates predefined manipulation target. Note that L_2 norm outperforms L_1 and L_∞ based formulations.

5.10 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. 18 shows the visual results for the self-reconstruction task.

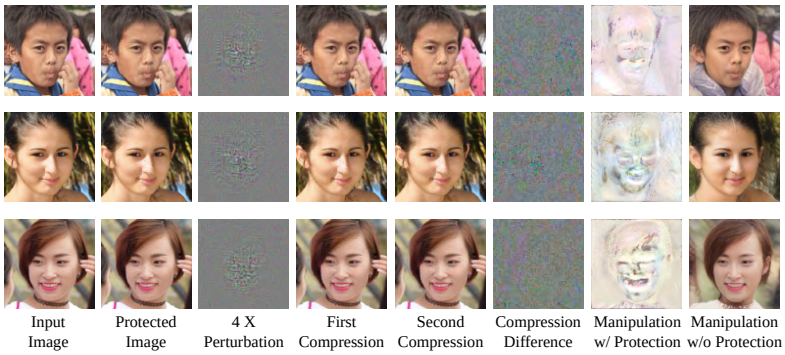

Fig. 17: Results for the self-reconstruction task with white image as manipulation target and for multi-level compression. First Compression denotes the first applied high compression C-30, Second Compression denotes the second compression (C-80) applied to the first compressed image. We apply different 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.

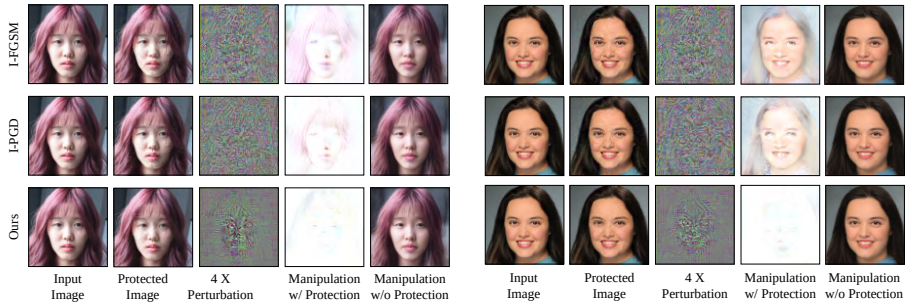

Fig. 18: Visual results for higher perturbation levels for the self-reconstruction task [13] with white image as manipulation target. Even for a higher perturbation level, our proposed approach outperforms alternate methods.

5.11 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 [16]

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

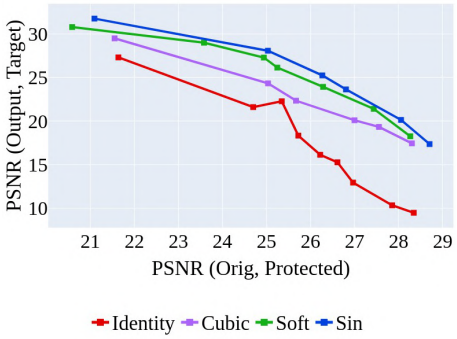

Fig. 19: PSNR performance for training our method with different approximations to round operation for JPEG compression for the self-reconstruction task [13] with white image as manipulation target. Sin approximation outperforms other approximations with comparable results for Soft approximation.

2. Soft Approximation [9]
$$\tilde{x} = x - \frac{\sin(2\pi x)}{2\pi},$$
$$x := \left[\text{round}(x) - \tilde{x}\right]_{\text{detach}} + \tilde{x},$$
where "detach" indicates that no gradients will be propagated during backpropagation.
3. Sin Approximation [9]
$$x := x - \frac{\sin(2\pi x)}{2\pi}$$

We also compare against the identity operation $x := x$ as the baseline. The performance comparison for these methods is visualized in Fig. 19. 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. Targeted Adversarial Attacks. https://www.kaggle.com/c/nips-2017-targeted-adversarial-attack, accessed: 2017-07-01 2
2. 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 10, 11, 12
3. 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, 8, 9, 10
4. 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 6
5. Gal, R., Patashnik, O., Maron, H., Chechik, G., Cohen-Or, D.: Stylegan-nada: Clip-guided domain adaptation of image generators (2021) 10, 11, 13
6. 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) 6
7. 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, 6
8. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks (2019) 3, 6
9. 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) 17
10. Kurakin, A., Goodfellow, I.J., Bengio, S.: Adversarial machine learning at scale. ArXiv **abs/1611.01236** (2017) 4, 5, 14
11. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. ArXiv **abs/1706.06083** (2018) 4, 5, 11, 14
12. 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, 8, 9, 10
13. 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, 8, 9, 10, 12, 13, 14, 16, 17
14. 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 3
15. Ruiz, N., Bargal, S.A., Sclaroff, S.: Disrupting deepfakes: Adversarial attacks against conditional image translation networks and facial manipulation systems (2020) 2
16. Shin, R.: Jpeg-resistant adversarial images (2017) 16

17. 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 6
18. VGGFace2-HQ: Vggface2-hq (2020), https://github.com/NNNNAI/VGGFace2-HQ 3, 6