FaceShield: Defending Facial Image against Deepfake Threats

Supplementary Material

Contents

A. Additional Explanation on Our Attack **2**
A.1. Attention disruption Attack 2
A.2 MTCNN Attack 2

B. Additional Related Work **4**

C. Additional Experimental Details **4**
C.1. Implementation Details 4
C.2. Human Evaluation 5

D. Additional Ablation Study **7**
D.1. MTCNN Resize Robustness 7
D.2 Gaussian blur Effect 8

E. Evaluating FaceShield under Image Purifications **9**

F. Additional Qualitative Results **9**

G Additional Experiments **19**

A. Additional Explanation on Our Attack

A.1. Attention disruption Attack

Algorithm. The full procedure of attention disruption attack is summarized in Algorithm 1.

Algorithm 1: Adversarial loss in cross attention.

```
Input: perturbation $\delta$, query embedding $Q_x$, original source face embedding $K_x$, adversarial source face embedding
       $K_{(x+\delta)}$, low variance threshold $t_{\mathtt{var}}$, maximum variance value $\sigma_{\mathtt{max}}$, low variance mask $M_{\mathtt{var}}$, attention loss
       $\mathcal{L}_{\mathtt{attn}}$, attention loss function $\mathcal{F}$
Result: stored low-variance mask $M_{\mathtt{var}}$, added attention loss $\mathcal{L}_{\mathtt{attn}}$
if $M_{\mathtt{var}}$ is not precomputed then
    // Construct Ground Truth
    Compute original attention map: $A_{\mathtt{map}} \leftarrow \textbf{Softmax}(Q_x K_x^T / \sqrt{d})$
    Compute variance: $A_{\mathtt{var}} \leftarrow \textbf{Var}(A_{\mathtt{map}})$
    Calculate low-variance threshold: $P_{t_{\mathtt{var}}} \leftarrow \textbf{Quantile}(A_{\mathtt{var}}, t_{\mathtt{var}})$
    Generate low-variance mask: $M_{\mathtt{var}} \leftarrow \textbf{Mask}(A_{\mathtt{var}}, P_{t_{\mathtt{var}}})$
    Store $M_{\mathtt{var}}$ for applying adversarial noise
end
else
    // Compute Adversarial Loss
    Compute adversarial attention map: $A'_{\mathtt{map}} \leftarrow \textbf{Softmax}(Q_x K_{(x+\delta)}^T / \sqrt{d})$
    Compute variance: $A'_{\mathtt{var}} \leftarrow \textbf{Var}(A'_{\mathtt{map}})$
    Calculate attention loss in low-variance regions: $\mathcal{L}_{\mathtt{attn}} \leftarrow \mathcal{L}_{\mathtt{attn}} + \mathcal{F}(\Delta)$,
      where $\Delta = (\sigma_{\mathtt{max}} - A'_{\mathtt{var}}) \odot M_{\mathtt{var}}$
end
Subsequent steps are not shown here.
```

A.2. MTCNN Attack

Model architecture.

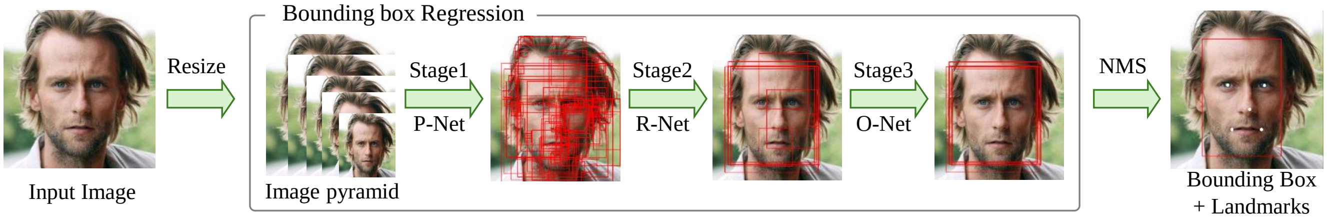


Figure 1. **MTCNN model architecture overview**.

The Multi-task Cascaded Convolutional Neural Network (MTCNN) is a deep learning-based framework for face detection and facial landmark localization. Its architecture consists of three cascaded convolutional neural networks, each refining face candidates while ensuring computational efficiency (see Fig.1). The Proposal Network (P-Net) employs a sliding window to scan the input image, generating bounding box proposals and associated confidence scores. Non-maximum suppression (NMS) is applied to remove redundant proposals. The Refine Network (R-Net) filters the bounding boxes further, reducing false positives and improving localization accuracy. Finally, the Output Network (O-Net) refines the bounding boxes and predicts precise facial landmark locations for face alignment. A key strength of MTCNN lies in its multi-scale input processing strategy. By resizing the input image across multiple scales, the network effectively captures faces of varying sizes, ensuring robust detection under diverse scenarios. This approach enables the P-Net to detect both large and small faces within a single pipeline, generating a comprehensive set of bounding box proposals. The cascaded structure leverages these multi-scale candidates, progressively refining them to achieve high detection accuracy and precision, even in complex scenes with occlusions or extreme pose variations.

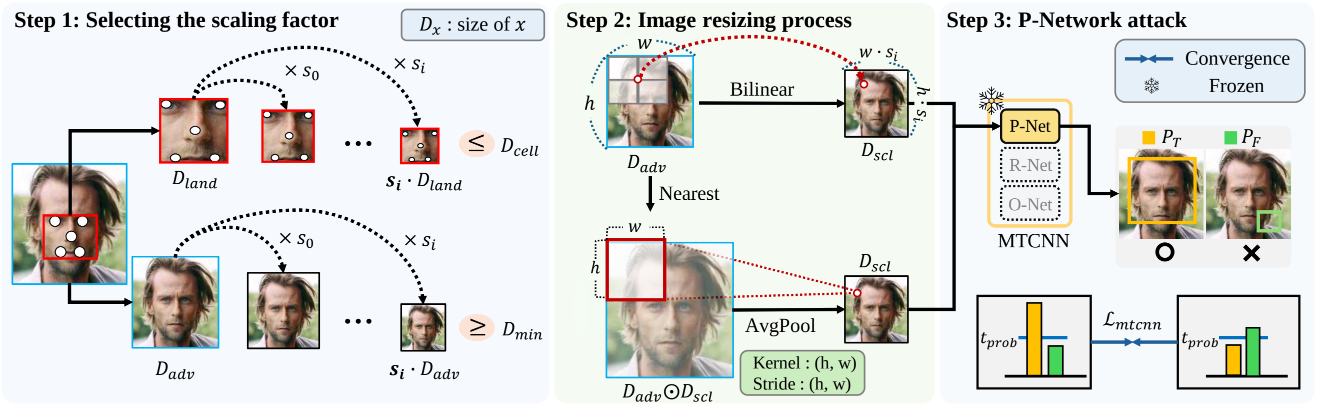


Figure 2. **MTCNN Attack Overview**. The attack process on MTCNN consists of three parts: (i) Selecting the scaling factor s_i, where the scale value is chosen according to Eq.1; (ii) Image resizing process, where we extend the robustness of resizing modes by using both Bilinear interpolation and our proposed Area-based method; (iii) P-Net attack, which decreases the probability values of candidate scales.

Details of the scaling factor selection process. MTCNN uses a multi-scale approach for face detection, which motivates us to extend the robustness of our adversarial noise across different scaling factors. To achieve this, we calculate the loss over multiple scales by dividing the image into several scales (see Fig.2). The process of selecting the optimal scaling factor is as follows: Initially, we calculate the minimum bounding box size $D_{\mathtt{land}}$ that encompasses key facial landmarks (eyes, nose, mouth) in the input image, obtained by passing the original image through MTCNN. Suitable scale values s_i are chosen to adjust the initial bounding box size $D_{\mathtt{cell}}$ to be larger than $D_{\mathtt{land}}$, while ensuring that the scaled input image size $D_{\mathtt{adv}}$ remains greater than the minimum allowable size $D_{\mathtt{min}}$. This is mathematically expressed as:

$$Scales = \left\{ s_i \;\middle|\; s_i \cdot D_{\mathtt{land}} \leq D_{\mathtt{cell}},\; s_i \cdot D_{\mathtt{adv}} \geq D_{\mathtt{min}} \right\} \tag{1}$$

where s_i is defined as $s_i = \frac{D_{\mathtt{cell}}}{D_{\mathtt{min}}} \times k^{i-1}$, with k being a predefined scale factor and i a non-negative integer. This ensures that only bounding boxes reaching MTCNN's final layers are effectively targeted.

Algorithm. The image resizing process and the P-Network attack method are summarized in Algorithm 2.

Algorithm 2: Adversarial loss in MTCNN Attack.

```
Input: source face image $x$, perturbation $\delta$, probability threshold $t_{\mathtt{prob}}$, image resize scale set $Scales$, mtcnn P-Network
       $\mathcal{T}$, mtcnn loss $\mathcal{L}_{\mathtt{mtcnn}}$, mtcnn loss function $\mathcal{F}$
Result: added mtcnn loss $\mathcal{L}_{\mathtt{mtcnn}}$
Update input image with perturbation: $x_{\mathtt{adv}} \leftarrow x + \delta$
Get input image size: $D_{\mathtt{adv}} \leftarrow Shape(x_{\mathtt{adv}})$
Set kernel and stride sizes: $K, S \leftarrow D_{\mathtt{adv}}$
for $s_i$ in $Scales$ do
    Set scaled image size: $D_{\mathtt{scl}} \leftarrow s_i \times D_{\mathtt{adv}}$
    Compute intermediate image size: $D_{\mathtt{int}} \leftarrow D_{\mathtt{adv}} \odot D_{\mathtt{scl}}$
    Upscaling image by using NEAREST: $\hat{x}_{\mathtt{adv}} \leftarrow \textbf{Scale}(x_{\mathtt{adv}}, D_{\mathtt{int}})$
    Apply average pooling: $\tilde{x}_{\mathtt{adv}} \leftarrow \textbf{Pool}(\hat{x}_{\mathtt{adv}}, K, S)$
    Obtain bbox probability: $P_{\mathtt{T}}, P_{\mathtt{F}} \leftarrow \mathcal{T}(\tilde{x}_{\mathtt{adv}})$
    Generate high-probability mask: $M_{\mathtt{prob}} \leftarrow \textbf{Mask}(P_{\mathtt{T}}, t_{\mathtt{prob}})$
    Calculate mtcnn loss in mask region: $\mathcal{L}_{\mathtt{mtcnn}} \leftarrow \mathcal{L}_{\mathtt{mtcnn}} + \mathcal{F}(\Delta)$,
      where $\Delta = (\mathcal{T}(\tilde{x}_{\mathtt{adv}}) - p_{\mathtt{gt}}) \odot M_{\mathtt{prob}}$
end
```

B. Additional Related Work

Deepfake adversarial attack. Existing research on adversarial attacks against deepfakes has focused on two main approaches: one involves targeting deepfake models based on the structural properties of specific GANs, and the other focuses on facial feature extractors to attack multiple deepfake models that use them. Studies such as [11, 26, 36] have focused on degrading the quality of images by targeting various GANs [5, 10, 16, 32, 43]. However, these approaches are ineffective against DM-based models [14, 33, 42]. As a study that attacks facial feature extractors, [17] performs adversarial attacks on several face landmark models [23, 35, 39], although the extractors targeted in this study are now less commonly used. [13] disrupt face detection targeting the MTCNN model by applying specific patches, but this approach has the limitation of being visible. Another method attacking the same model, [40], propose using `BILINEAR` interpolation to attack across multiple scales. However, since the `BILINEAR` mode only uses specific anchor points during interpolation, adversarial noise generated with this approach easily loses effectiveness when other interpolation modes are applied.

Diffusion adversarial attack. As image editing techniques utilizing DMs have gained traction, research on adversarial attacks targeting these architectures has progressed significantly. AdvDM [19] generates adversarial examples by optimizing latent variables sampled from the reverse process of a DM. Similarly, Glaze [29] investigates the latent space, generating adversarial noise and proposing a noise clamping technique based on `LPIPS` minimizing perceptual distortion of the original image. Photoguard [28] is noteworthy for introducing the concept of encoder attacks, and separately, it presents a diffusion attack that utilizes the denoised generated image. Mist [18] combines the semantic loss proposed in [19] with the textual loss from [28], leading to a novel loss function that enables the generation of transferable adversarial examples against various diffusion-based attacks. Diff-Protect [37] proposes a novel approach that updates by minimizing loss, unlike previous studies. DiffusionGuard [4] introduces adversarial noise early in the diffusion process, preventing image editing techniques from reproducing sensitive areas. All previous research has been directed toward protecting images when they are utilized directly in DMs, as depicted in Fig. 2(a) in the main paper.

Adversarial noise with frequency-domain. There are various approaches utilizing frequency in generating adversarial noise. Maiya et al. [21] suggested that using frequency is effective in designing imperceptible noise while Wang et al. [34] argued that high-frequency components are effective for attacking CNN-based models. On the other hand, recent studies [9, 31] has demonstrated that it is possible to attack DNN-based models [22, 30] effectively using only low-frequency components. Additionally, AdvDrop [7] showed that transformations in the frequency domain of images can induce misclassification. Ling et al. [20] proposed the frequency data transformation(FDT) method to improve transferability between models in black-box attacks.

C. Additional Experimental Details

C.1. Implementation Details

In this paper, we generate *FaceShield* by utilizing the mid-layer cross-attention of the open-source Stable Diffusion Model v1.5 [25], the upper part of the CLIP Image Projector in the CLIP Model [24], only the P-Network from the PyTorch version of MTCNN [41], and two variants of ArcFace [6]. All images are resized to 512×512 before processing, and experiments are conducted on an RTX A6000. A more detailed description is provided in Table 1, where the same hyperparameters are applied as in the baseline methods [18, 19, 28, 37] for generating noise.

Norm	ϵ	**step size**	**number of steps**
ℓ_∞	12/255	1/255	30

Table 1. Hyperparameters used for the PGD attacks.

As a result, *FaceShield* achieves 24 seconds per image with only 15 GB of memory, demonstrating significantly lower resource costs compared to baseline methods, as shown in Table 2. This efficiency is achieved through three key optimizations: (i) Restricting the input to the Conditioned Face Attack (CFA) module, ensuring the process focuses solely on facial regions. (ii) Extracting gradients from the condition path (Fig.3 in the main paper), eliminating the need for gradient accumulation across multiple timesteps. (iii) Updating only the mid-layer of the UNet, rather than optimizing the entire network. These optimizations enable *FaceShield* to achieve high performance with minimal computational resources.

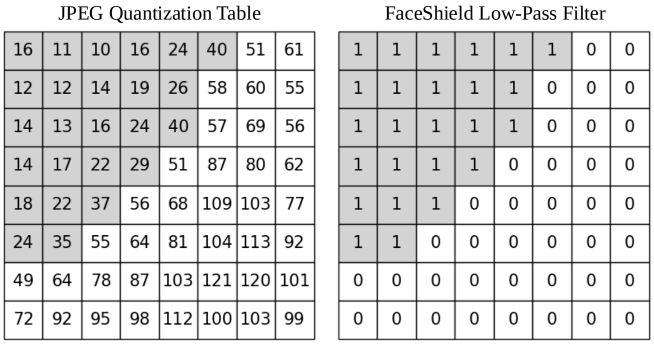
Gaussian Blur. To achieve more precise detection of intensity variations between adjacent pixels, we employ a 3×3 Sobel matrix. Its compact size ensures faster convolution operations and reduces memory consumption, which is crucial for iterative

Baseline	ISM↓	LPIPS↓	VRAM	Sec.↓
AdvDM [19]	0.288	0.4214	20 GB	39
Mist [18]	0.291	0.5492	22 GB	80
PhotoGuard [28]	0.294	0.5515	28 GB	234
SDST [37]	0.303	0.5409	**11 GB**	34
Ours	**0.168**	**0.2017**	15 GB	**24**

Table 2. Comparison of resource costs with baseline methods.

computations. Subsequently, a 9×9 padding is applied to the detected regions to generate thicker masks, ensuring smoother transitions during the subsequent Gaussian blur step and mitigating abrupt changes.

Low-pass Filter. We utilize perturbations in the frequency domain by performing an 8×8 patch division followed by a Discrete Cosine Transform (DCT). This design is inspired by the JPEG compression scheme, which operates on 8×8 blocks and employs a Quantization Table to prioritize low-frequency components. Furthermore, the 8×8 patch division offers computational efficiency advantages compared to approaches without such division during the DCT process. Unlike JPEG compression, we skip the RGB-to-YCbCr color space transformation. This decision is based on two considerations: (i) perturbations inherently contain both positive and negative values, which are incompatible with the typical range constraints of the YCbCr domain, and (ii) experiments demonstrate that handling frequencies directly in the RGB domain is sufficient to achieve our performance objectives without compromising effectiveness. The coefficients for our low-pass filter are selected from the Luminance Quantization Table, focusing exclusively on values below 40, as illustrated in Fig.3.



JPEG Quantization Table

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

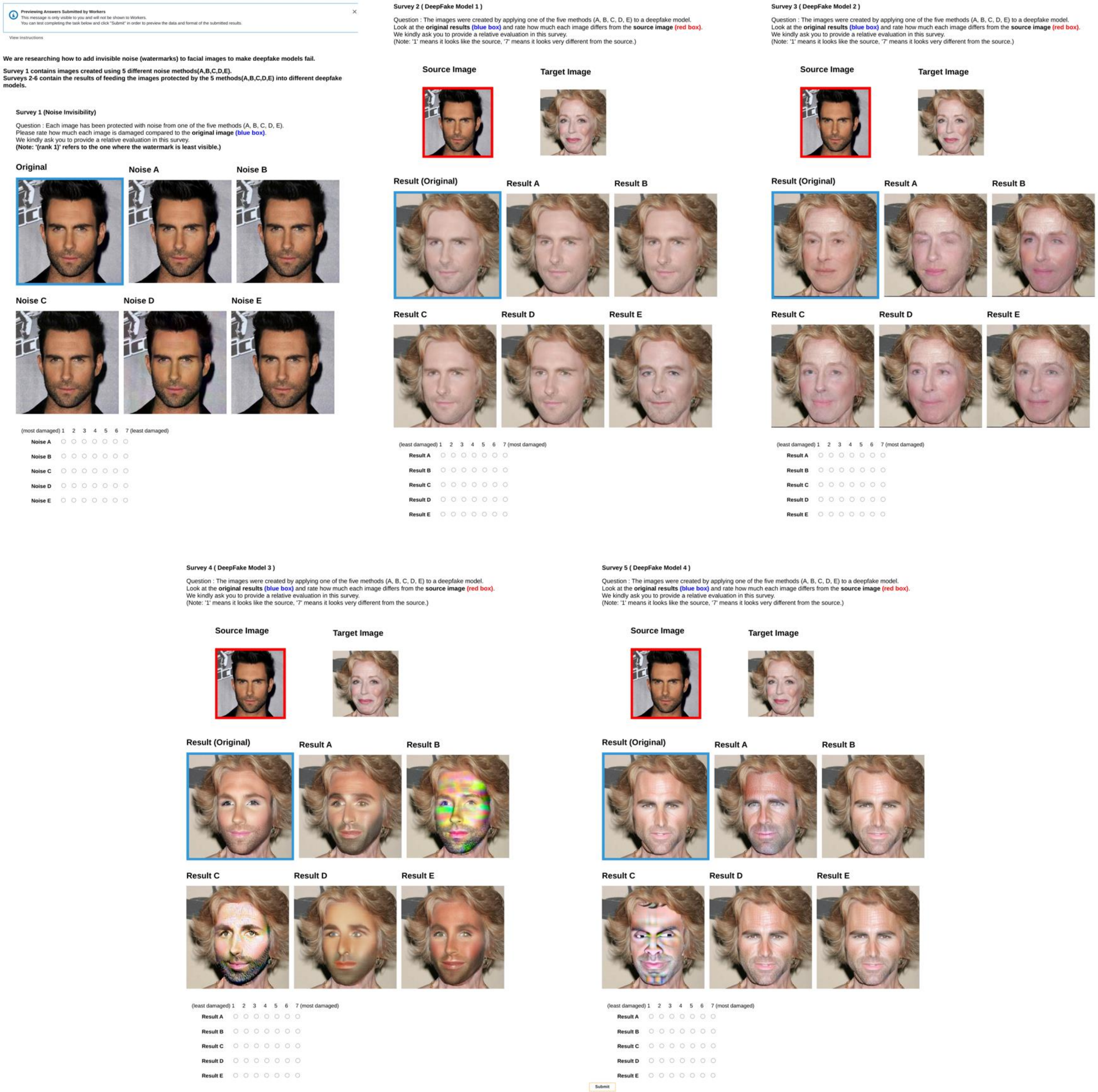
FaceShield Low-Pass Filter

1	1	1	1	1	1	0	0
1	1	1	1	1	0	0	0
1	1	1	1	1	0	0	0
1	1	1	1	0	0	0	0
1	1	1	0	0	0	0	0
1	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Figure 3. The table on the left shows the Luminance Quantization Table used in the JPEG compression process. The table on the right illustrates the *FaceShield*'s Low-pass Filter, which is created by selecting only the values below 40.

C.2. Human Evaluation

We conduct a human evaluation study to assess the visibility of the noise and the protection performance across four deepfake models [14, 33, 38, 42], along with four baseline methods [18, 19, 28, 37]. Specifically, participants are asked to score images on a scale from 1 (low performance) to 7 (high performance) in response to the following two questions: (i) *"How much each image is damaged compared to the original image?"*, which measures the visibility of the protective noise pattern relative to each baseline method, and (ii) *"How much each image differs from the source image?"*, which evaluates how effectively each method prevents the deepfake models from reflecting the original source face. We use 20 images (10 from the CelebA-HQ dataset and 10 from the VGGFace2-HQ dataset) across four deepfake models, with 100 participants providing their ratings. To enhance fairness, the positions of the compared methods within each question are randomly shuffled. An example survey is shown in Fig.4.



Previewing Answers Submitted by Workers
This message is only visible to you and will not be shown to Workers.
You can test completing the task below and click "Submit" in order to preview the data and format of the submitted results.

We are researching how to add invisible noise (watermarks) to facial images to make deepfake models fail.

Survey 1 contains images created using 5 different noise methods(A,B,C,D,E).
Surveys 2-6 contain the results of feeding the images protected by the 5 methods(A,B,C,D,E) into different deepfake models.

Survey 1 (Noise Invisibility)

Question : Each image has been protected with noise from one of the five methods (A, B, C, D, E).
Please rate how much each image is damaged compared to the **original image (blue box)**.
We kindly ask you to provide a relative evaluation in this survey.
(Note: '(rank 1)' refers to the one where the watermark is least visible.)

	(most damaged) 1	2	3	4	5	6	7 (least damaged)
Noise A	○	○	○	○	○	○	○
Noise B	○	○	○	○	○	○	○
Noise C	○	○	○	○	○	○	○
Noise D	○	○	○	○	○	○	○
Noise E	○	○	○	○	○	○	○

Survey 2 (DeepFake Model 1)

Question : The images were created by applying one of the five methods (A, B, C, D, E) to a deepfake model.
Look at the **original results (blue box)** and rate how much each image differs from the **source image (red box)**.
We kindly ask you to provide a relative evaluation in this survey.
(Note: '1' means it looks like the source, '7' means it looks very different from the source.)

	(least damaged) 1	2	3	4	5	6	7 (most damaged)
Result A	○	○	○	○	○	○	○
Result B	○	○	○	○	○	○	○
Result C	○	○	○	○	○	○	○
Result D	○	○	○	○	○	○	○
Result E	○	○	○	○	○	○	○

Survey 3 (DeepFake Model 2)

Question : The images were created by applying one of the five methods (A, B, C, D, E) to a deepfake model.
Look at the **original results (blue box)** and rate how much each image differs from the **source image (red box)**.
We kindly ask you to provide a relative evaluation in this survey.
(Note: '1' means it looks like the source, '7' means it looks very different from the source.)

	(least damaged) 1	2	3	4	5	6	7 (most damaged)
Result A	○	○	○	○	○	○	○
Result B	○	○	○	○	○	○	○
Result C	○	○	○	○	○	○	○
Result D	○	○	○	○	○	○	○
Result E	○	○	○	○	○	○	○

Survey 4 (DeepFake Model 3)

Question : The images were created by applying one of the five methods (A, B, C, D, E) to a deepfake model.
Look at the **original results (blue box)** and rate how much each image differs from the **source image (red box)**.
We kindly ask you to provide a relative evaluation in this survey.
(Note: '1' means it looks like the source, '7' means it looks very different from the source.)

	(least damaged) 1	2	3	4	5	6	7 (most damaged)
Result A	○	○	○	○	○	○	○
Result B	○	○	○	○	○	○	○
Result C	○	○	○	○	○	○	○
Result D	○	○	○	○	○	○	○
Result E	○	○	○	○	○	○	○

Survey 5 (DeepFake Model 4)

Question : The images were created by applying one of the five methods (A, B, C, D, E) to a deepfake model.
Look at the **original results (blue box)** and rate how much each image differs from the **source image (red box)**.
We kindly ask you to provide a relative evaluation in this survey.
(Note: '1' means it looks like the source, '7' means it looks very different from the source.)

	(least damaged) 1	2	3	4	5	6	7 (most damaged)
Result A	○	○	○	○	○	○	○
Result B	○	○	○	○	○	○	○
Result C	○	○	○	○	○	○	○
Result D	○	○	○	○	○	○	○
Result E	○	○	○	○	○	○	○

Figure 4. **Human Evaluation Survey**. Survey 1 (the first figure) evaluates the visibility of the noise, while Surveys 2-5 (the remaining figures) assess the protection performance across different deepfake models [14, 33, 38, 42]. The scoring scale ranges from 1 to 7, and to ensure fairness, the placement of comparison methods was randomly shuffled for each survey.

D. Additional Ablation Study

D.1. MTCNN Resize Robustness

The experimental results for MTCNN, as discussed in the Ablation Study of the main paper, are presented through both quantitative and qualitative evaluations. Specifically, Table 3 and Table 4 provide quantitative metrics, while Fig.5 illustrates how the detected regions propagate to the subsequent network when face detection fails at the P-Network stage. These results demonstrate the superiority of the newly proposed method in *FaceShield* compared to the `BILINEAR` approach introduced in prior work [40], which aimed to perturb the MTCNN model. In particular, Table 3 evaluates various scaling modes provided by `OpenCV`, while Table 4 focuses on those offered by `Pillow`. The experiments were conducted using both the PyTorch and TensorFlow versions of the framework. For comprehensive evaluation, we utilized 3,000 images each from the CelebA-HQ [12] and VGGFace2-HQ [3] datasets. The results confirm that *FaceShield* achieves superior coverage across diverse scaling modes compared to previous approaches.

Dataset	CelebA-HQ [12]					
Method	`BILINEAR`	`AREA`	`NEAREST`	`CUBIC`	`LANC`	`EXACT`
BILINEAR	93.77%	0.07%	0.40%	95.73%	95.67%	93.77%
Ours	97.31%	94.17%	4.13%	97.10%	97.00%	97.30%
Dataset	VGGFace2-HQ [3]					
Method	`BILINEAR`	`AREA`	`NEAREST`	`CUBIC`	`LANC`	`EXACT`
BILINEAR	87.23%	0.17%	0.37%	94.63%	94.43%	88.93%
Ours	89.20%	72.93%	2.47%	94.93%	95.27%	89.33%

Table 3. The metric values represent the detection failure rates of the MTCNN [41] model. Our scaling method demonstrates greater robustness across various scaling modes in the `OpenCV` Library compared to the existing approach, with particularly notable performance in the model's default setting, `AREA`.

Dataset	CelebA-HQ [12]					
Method	`BILINEAR`	`BOX`	`NEAREST`	`BICUBIC`	`LANCZOS`	`HAMMING`
BILINEAR	0.70%	0.80%	79.73%	0.57%	0.84%	0.70%
Ours	10.67%	98.57%	97.90%	16.90%	16.30%	37.53%
Dataset	VGGFace2-HQ [3]					
Method	`BILINEAR`	`BOX`	`NEAREST`	`BICUBIC`	`LANCZOS`	`HAMMING`
BILINEAR	1.37%	1.87%	68.83%	1.47%	1.60%	1.63%
Ours	12.83%	84.20%	87.97%	16.03%	15.53%	28.53%

Table 4. The metric values represent the detection failure rates of the MTCNN [41] model. Our scaling method demonstrates greater robustness across various scaling modes in the `Pillow` Library compared to the existing approach.

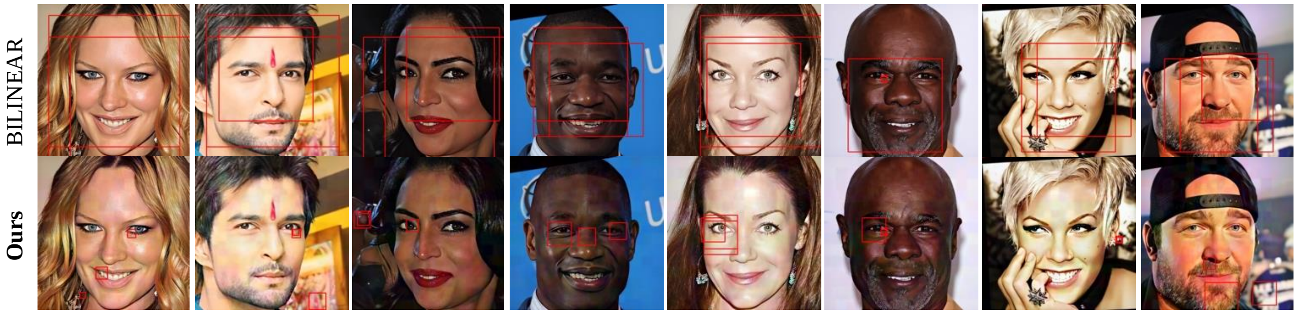


Figure 5. We compare the performance of the image resize method using only `BILINEAR` interpolation (top) and our proposed approach (bottom). Experiments are conducted with the default MTCNN resizing mode, `CV2.INTER_AREA` . The bounding boxes (red boxes) shown represent the top three outputs from the P-Net with the highest confidence scores.

D.2. Gaussian blur Effect

The qualitative results of the Gaussian blur effect, mentioned in the Ablation Study of the main paper, are presented in the following Fig.6, comparing the cases with and without its application. As shown in the figure on the right, Sobel filtering is applied to achieve effective invisibility while maintaining maximum performance, resulting in blurred areas where noticeable differences between adjacent regions exist. Additional examples of the results are provided in Fig.7.

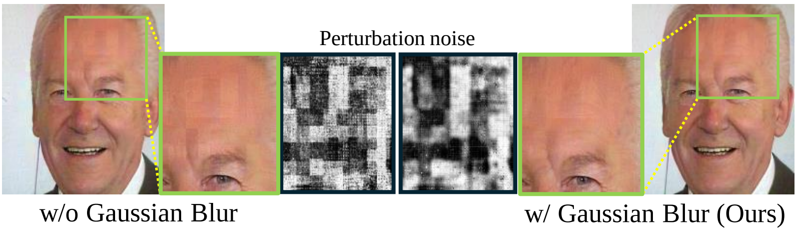


Figure 6. By detecting regions with large intensity differences between adjacent RGB pixels in the perturbation, a blur effect is applied, enhancing the invisibility of the noise.

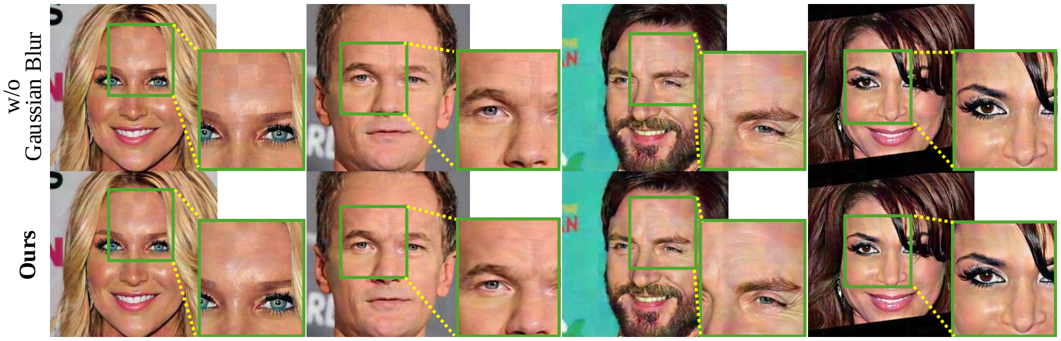


Figure 7. Qualitative comparison between the case with Gaussian Blur (bottom) and without Gaussian Blur (top).

E. Evaluating FaceShield under Image Purifications

We conduct additional experiments to demonstrate the robustness of *FaceShield* leveraging low-frequency components against various image purification techniques. Specifically, we evaluate the performance under three primary scenarios.

- **JPEG compression**: Images are compressed at quality levels of 90, 75, and 50 to introduce distortions.
- **Bit reduction**: Images are quantized to 8-Bit and 3-Bit formats, simulating lossy storage conditions.
- **Resizing**: Images are resized to 75% and 50% of their original dimensions and then restored to their original size. Two interpolation methods, `BILINEAR` and `INTER_AREA`, are applied during resizing.

These experiments are conducted using the IP-Adapter model [38], with the same dataset as in Table 1 in the main paper. The quantitative results for ISM and PSNR are presented in Fig.8, while the qualitative results are shown in Fig.9 and Fig.10. As shown in the results, *FaceShield* causes only minor performance degradation across various purification methods, yet still demonstrates superior performance compared to other baselines [18, 19, 28, 37], proving its remarkable robustness.

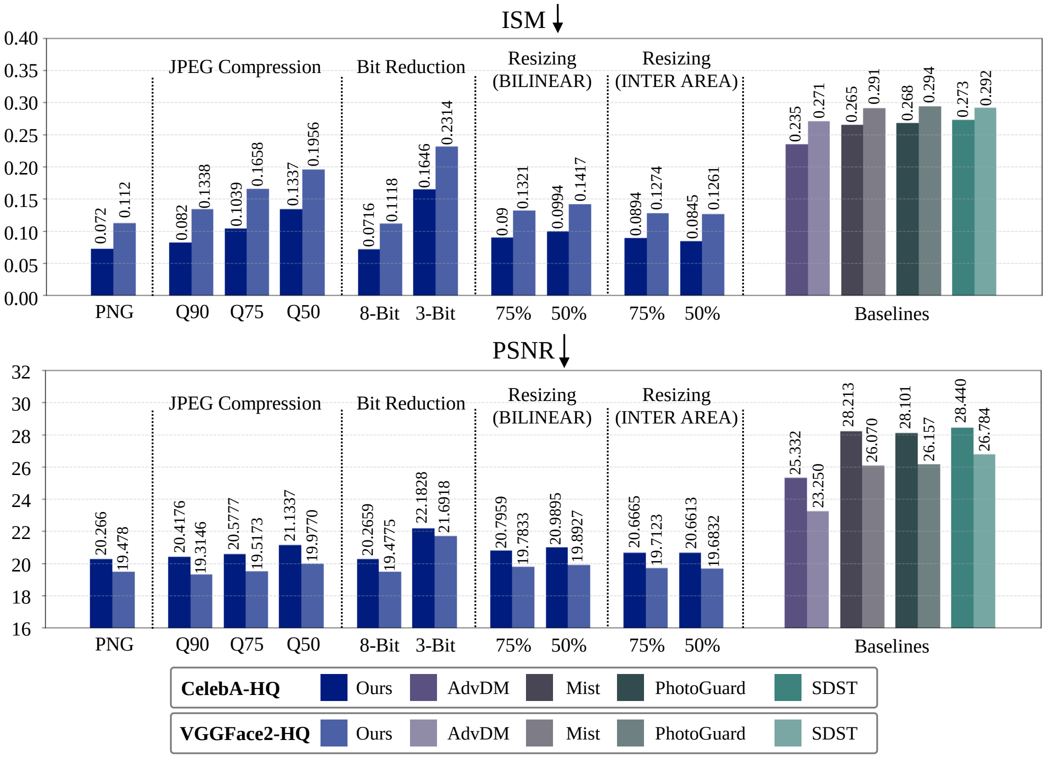


Figure 8. Quantitative results of *FaceShield*-protected images after passing through various purification methods and evaluated on a deepfake model [38]. Our method demonstrates robustness against various purification methods, including JPEG compression, bit reduction, and two types of resizing, with its performance compared to baseline methods [18, 19, 28, 37]. The results, measured using PSNR and Identity Score Matching (ISM), show that our method closely resembles lossless (PNG) outcomes while consistently outperforming the baselines. Both metrics indicate better performance with lower values.

F. Additional Qualitative Results

In this section, we present additional qualitative results of our methods. Specifically, Fig.11 to Fig.13 compare our approach with baseline methods [18, 19, 28, 37] on various diffusion-based deepfake models [14, 33, 38, 42], using a pair of source and target images. Fig.14 compares our method with the baselines on the FaceSwap via Diffusion model [33] across different image pairs. Fig.15 shows the comparison within the IP-Adapter model [38], while Fig.16 compares our method with the baselines on the DiffSwap model [42]. Fig.17 presents a comparison on the DiffFace model [14]. Finally, Fig.18 and Fig.19 showcase additional experiments on two GAN-based deepfake models: SimSwap [2] and InfoSwap [8], respectively.

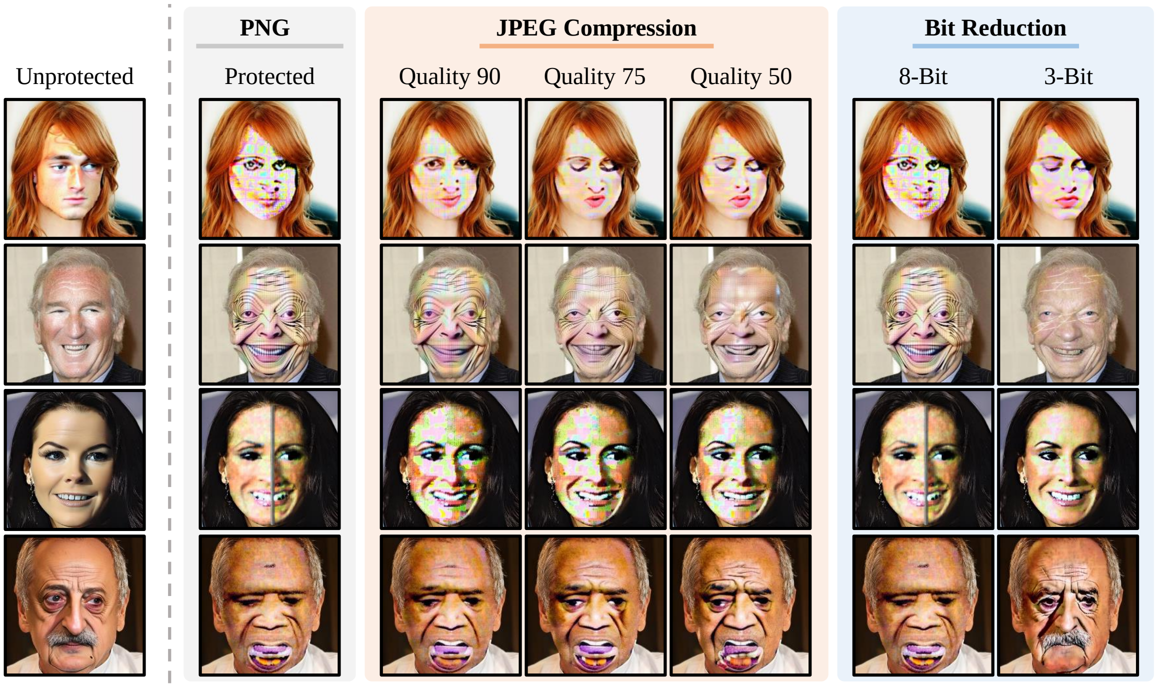


Figure 9. The results of applying three levels of JPEG compression and two levels of bit reduction to images protected by *FaceShield*, followed by evaluation on a deepfake model [38], show that the performance degradation is minimal compared to lossless storage (PNG).

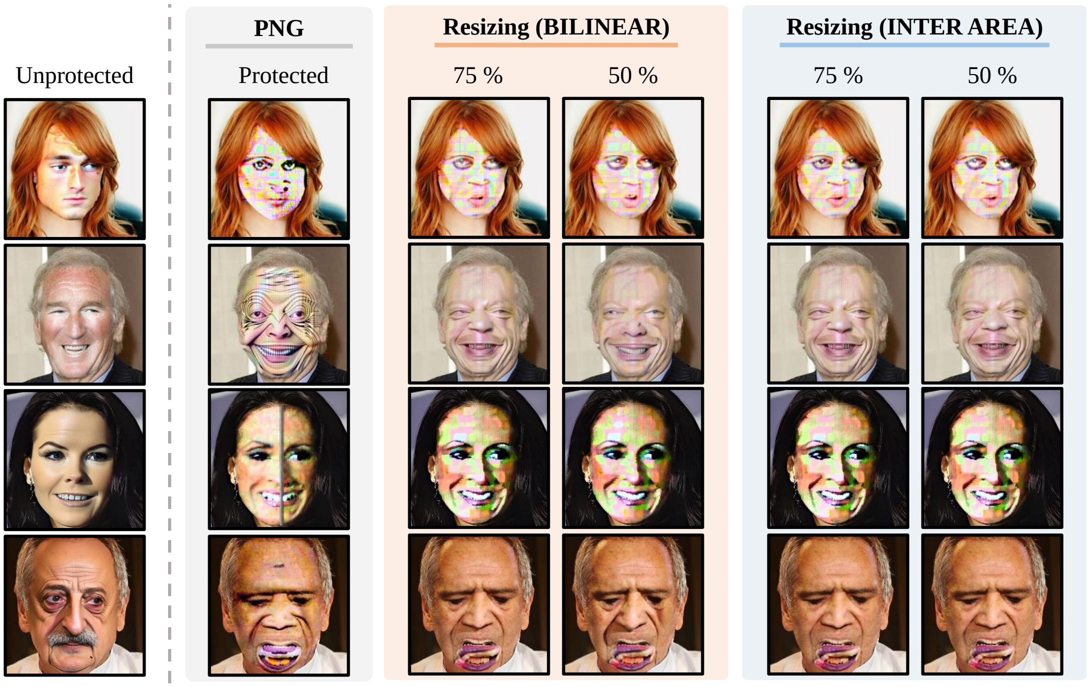


Figure 10. The results of applying two types of resizing methods, with 75% and 50% scaling, to images protected by *FaceShield*, followed by evaluation on a deepfake model [38], show that the performance degradation is minimal compared to lossless storage (PNG).

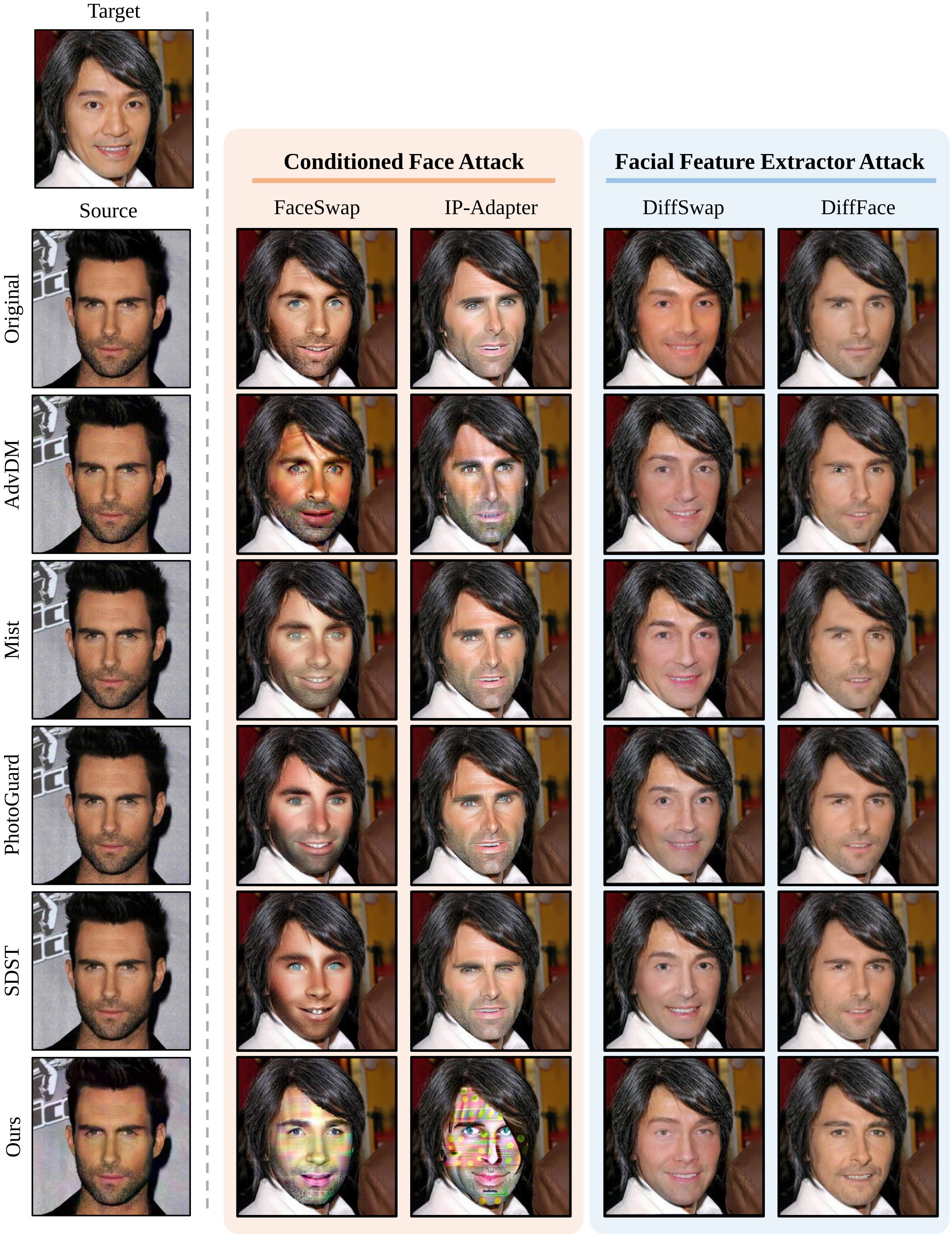


Figure 11. Qualitative comparisons with AdvDM [19], Mist [18], PhotoGuard [28], and SDST [37] across four diffusion-based deepfake models: FaceSwap [33], IP-Adapter [38], DiffSwap [42], and DiffFace [14].

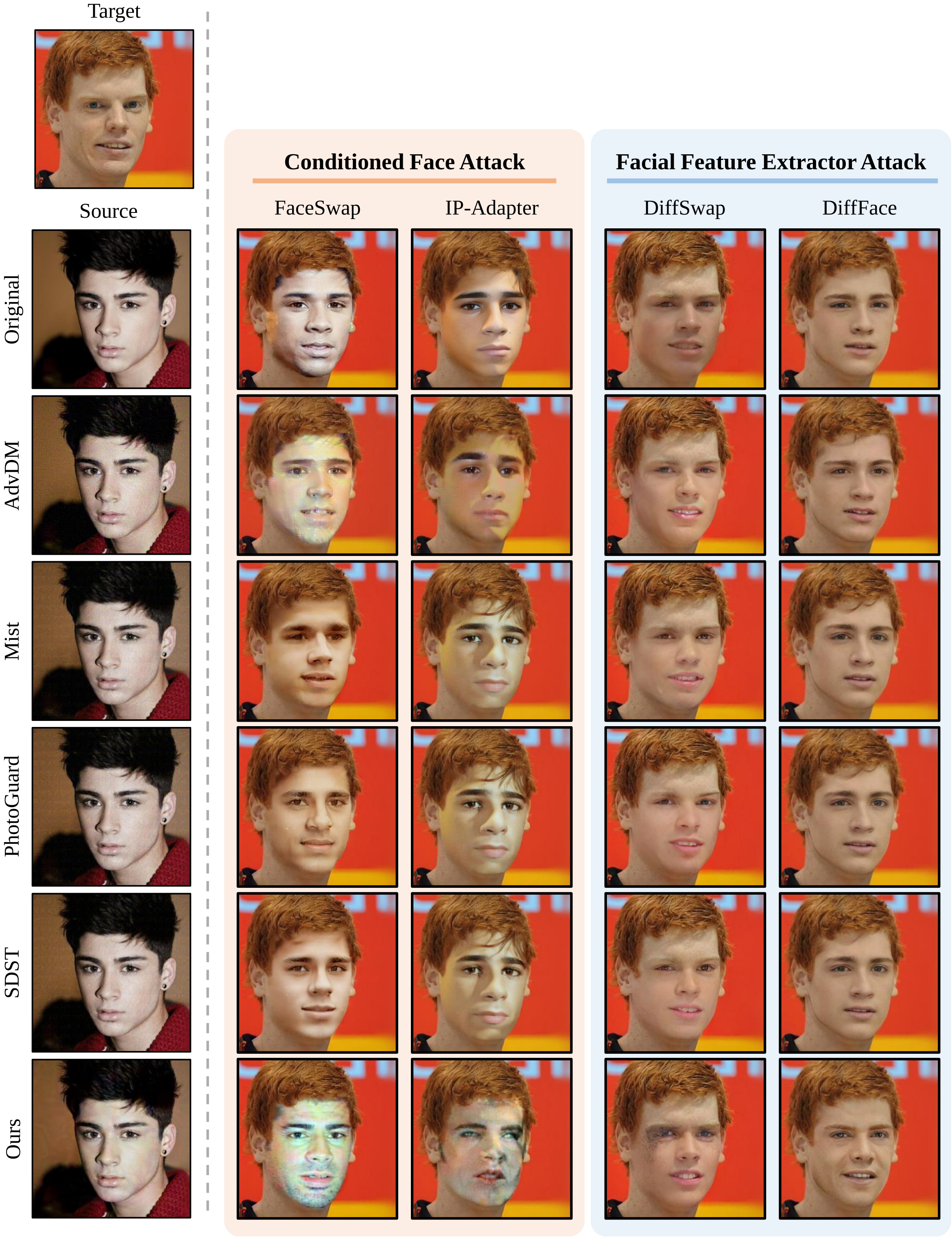


Figure 12. Qualitative comparisons with AdvDM [19], Mist [18], PhotoGuard [28], and SDST [37] across four diffusion-based deepfake models: FaceSwap [33], IP-Adapter [38], DiffSwap [42], and DiffFace [14].

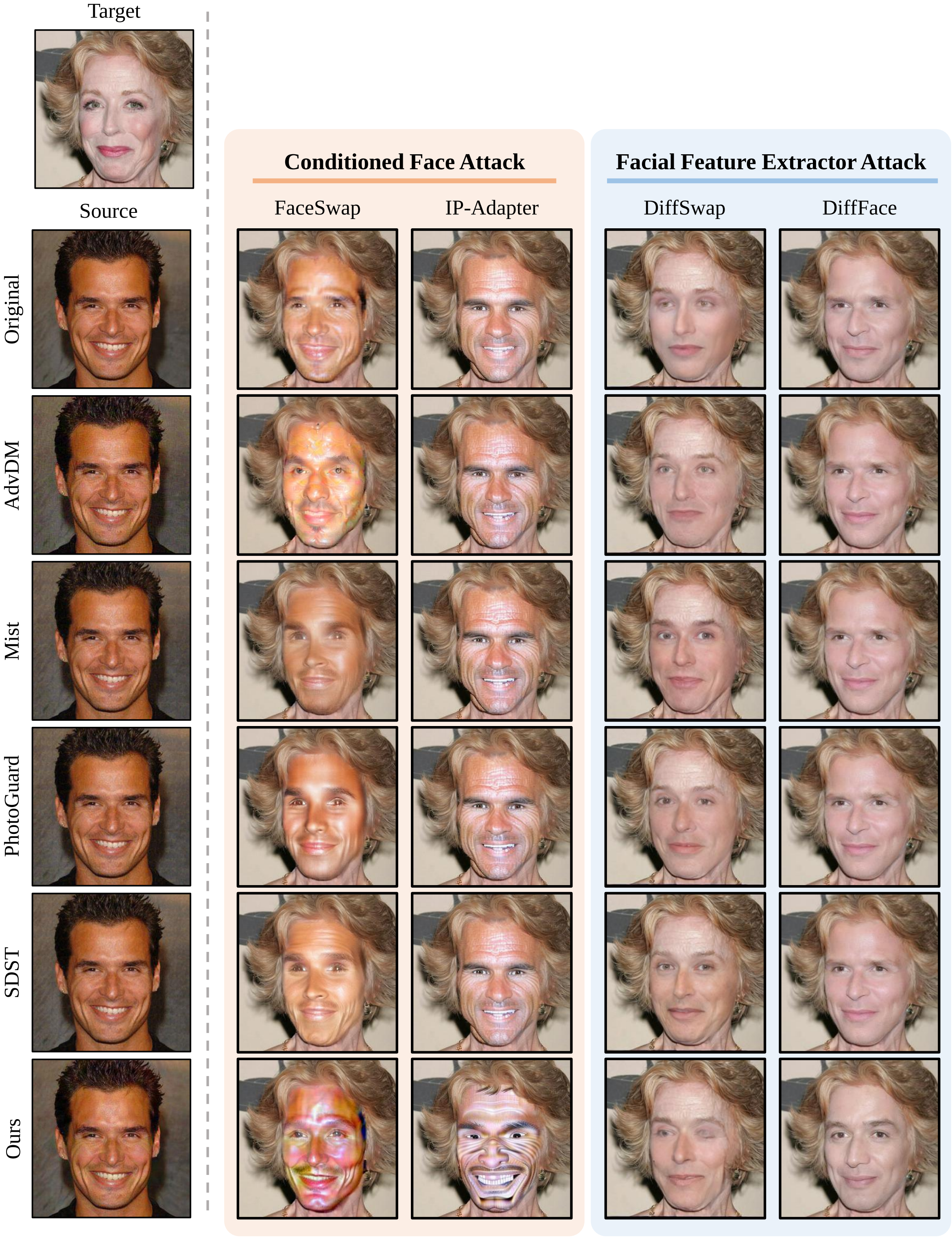


Figure 13. Qualitative comparisons with AdvDM [19], Mist [18], PhotoGuard [28], and SDST [37] across four diffusion-based deepfake models: FaceSwap [33], IP-Adapter [38], DiffSwap [42], and DiffFace [14].

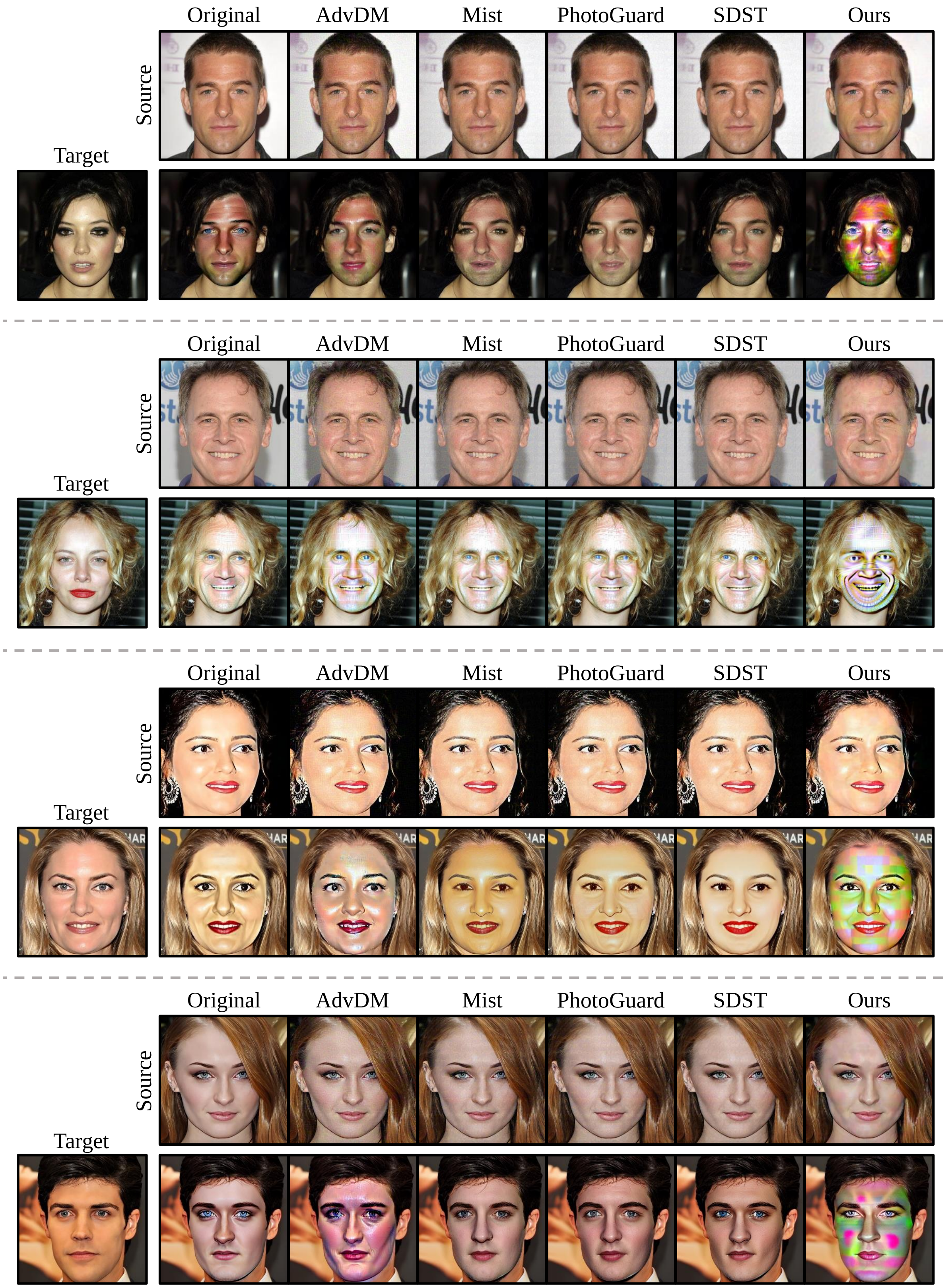


Figure 14. Qualitative comparisons for **FaceSwap** [33].

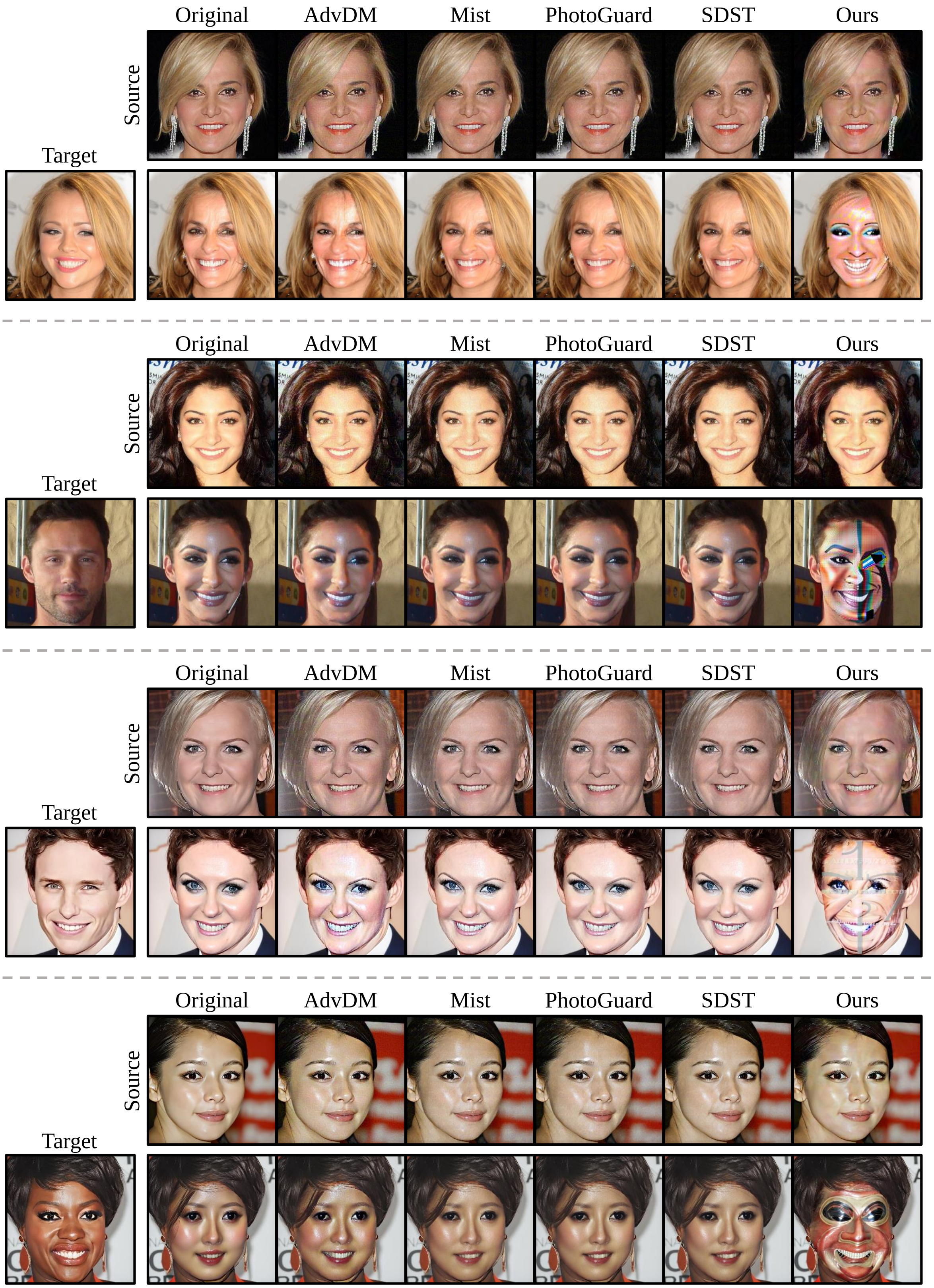


Figure 15. Qualitative comparisons for **IP-Adapter** [38].

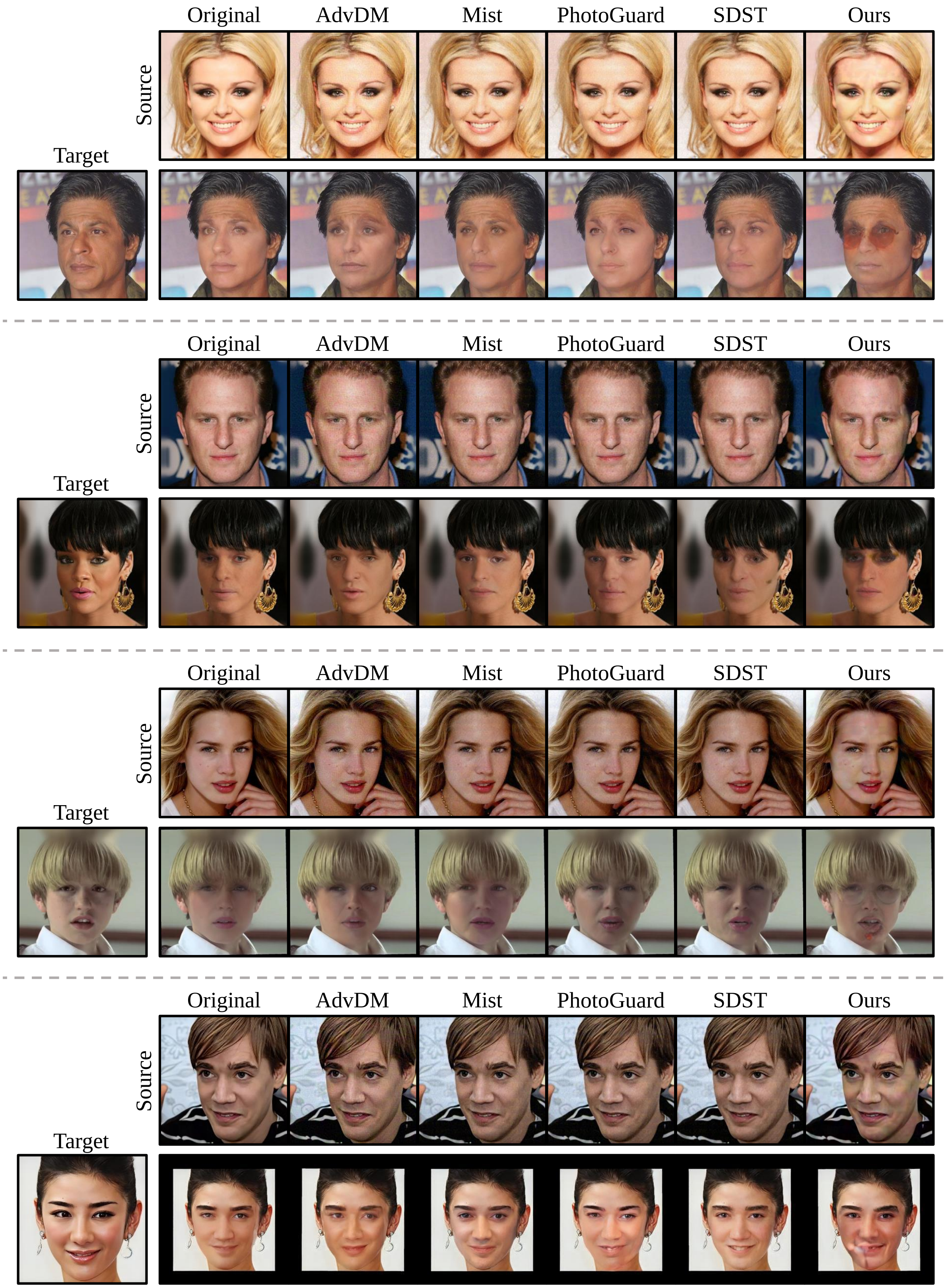


Figure 16. Qualitative comparisons for **DiffSwap** [42].

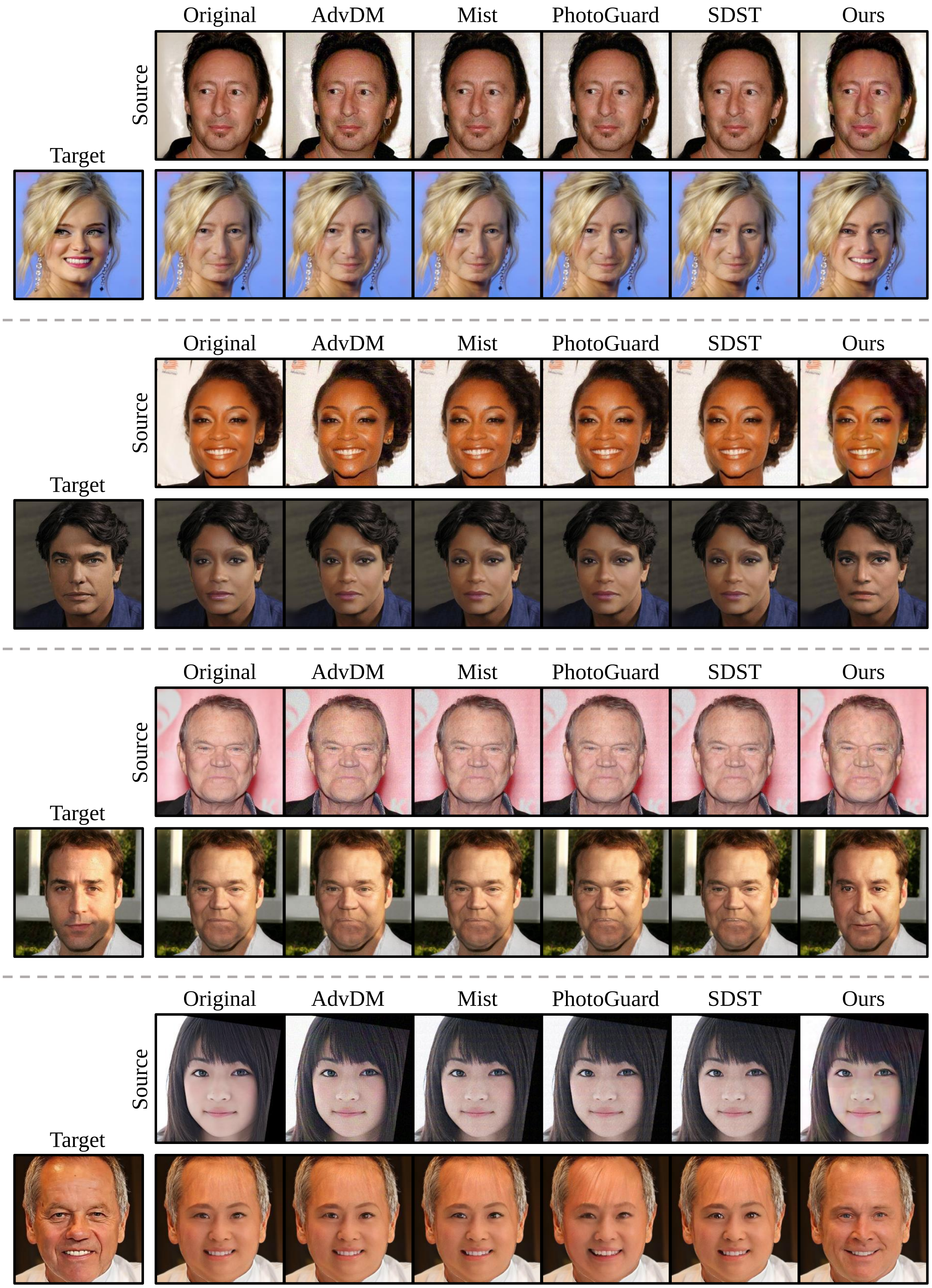


Figure 17. Qualitative comparisons for **DiffFace** [14].

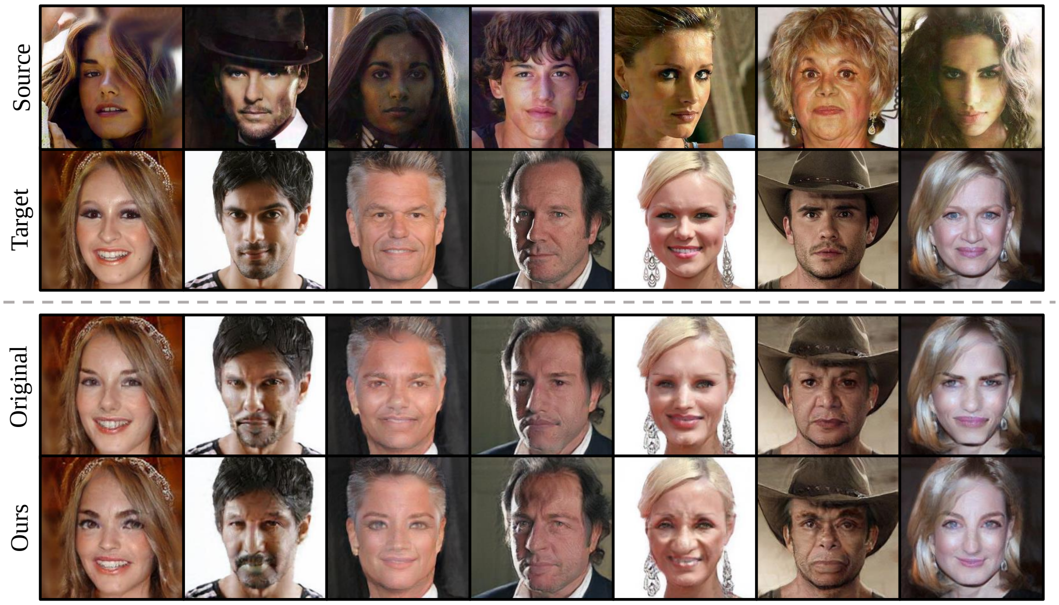


Figure 18. Qualitative results for **SimSwap** [2].

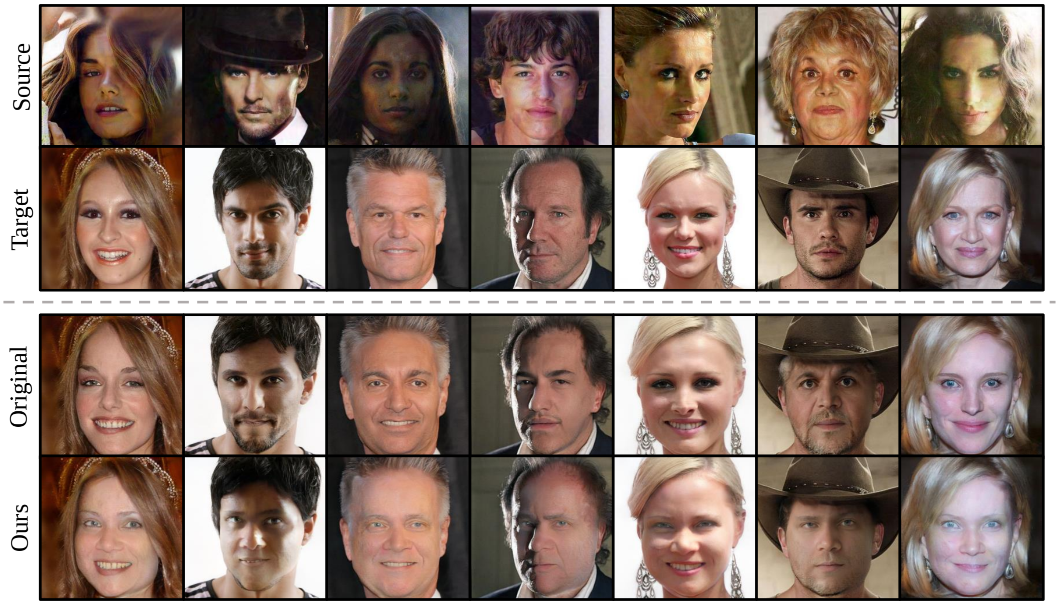


Figure 19. Qualitative results for **InfoSwap** [8].

G. Additional Experiments

Transferability experiments on variants of IP-Adapter

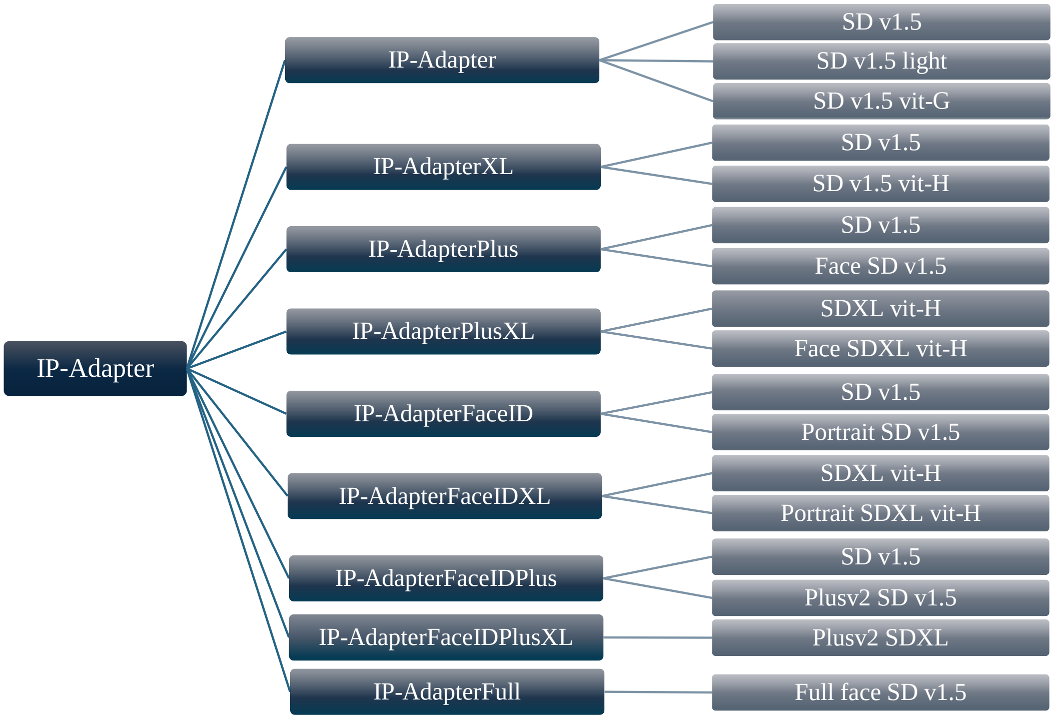


Figure 20. **IP-Adapter model family tree**. This diagram shows the hierarchical structure of the IP-Adapter variants.

IP-Adapter [38] is a lightweight adapter that enables image conditions in pre-trained text-to-image diffusion models [25]. Previous approaches [15, 27] that utilized image conditions primarily relied on fine-tuning text-conditioned diffusion models. However, these methods often demanded significant computational resources and resulted in models that were challenging to reuse. To address these limitations, the IP-Adapter, which proposes a decoupled cross-attention mechanism, has drawn considerable attention for its practical applicability. It is commonly used in inpainting methods with image conditions. As shown in Fig.20, multiple versions of the IP-Adapter model have been developed with Stable diffusion v1.5 [25].

A more detailed look at the various models reveals that the original model [38] uses the CLIP image encoder [24] to extract features from the input image. In contrast, the IP-AdapterXL improves on this by utilizing larger image encoders, such as ViT-BigG or ViT-H, which enhance both capacity and performance. On the other hand, the IP-AdapterPlus and XL versions modify the architecture by adopting a patch embedding method inspired by Flamingo's perceiver resampler [1], allowing for more efficient image encoding. Similarly, the IP-AdapterFaceID and XL versions replace the CLIP image encoder with InsightFace, extracting FaceID embeddings from reference images. This enables the combination of additional text-based conditions with the facial features of the input image, allowing for the generation of diverse styles. The IP-AdapterFaceIDPlus and XL versions further enhance the image encoding pipeline by incorporating multiple components. InsightFace is used for detailed facial features, the CLIP image encoder captures global facial characteristics, and the Perceiver-resampler effectively combines these features to improve the model's overall functionality.

Qualitative results. We evaluate the transferability across different IP-Adapter versions and present comparisons with baseline methods. Specifically, we conducted experiments on eight of these models, with results and model descriptions provided in Fig.21 to Fig.24. These results demonstrate the versatility of *FaceShield*, showing that it is applicable across various sub-models of the IP-Adapter [38].

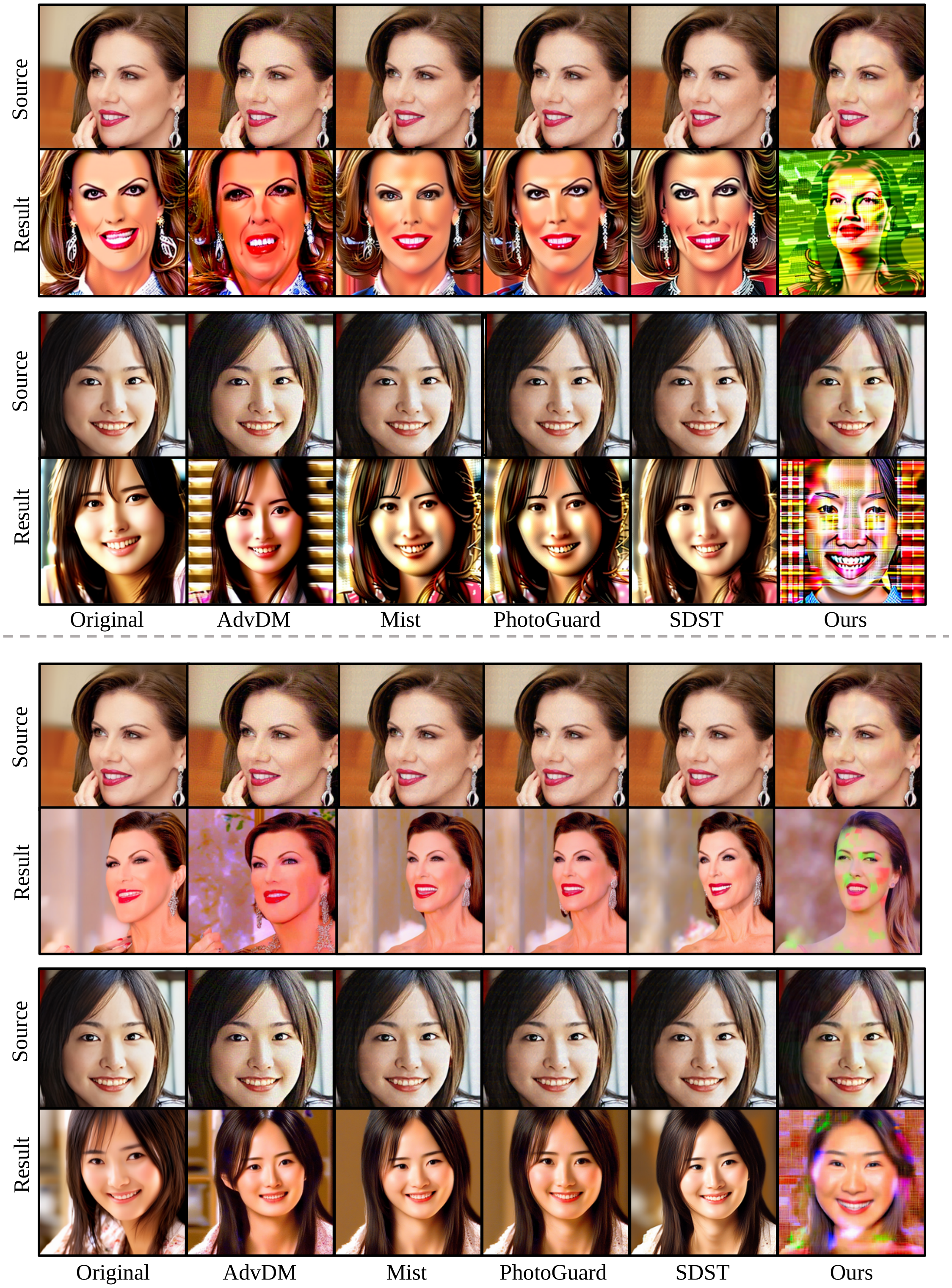


Figure 21. Qualitative comparison with baselines on the SD 1.5-based IP-Adapter **ControlNet** version (top) and SDXL-based IP-Adapter **ControlNet** version (bottom).

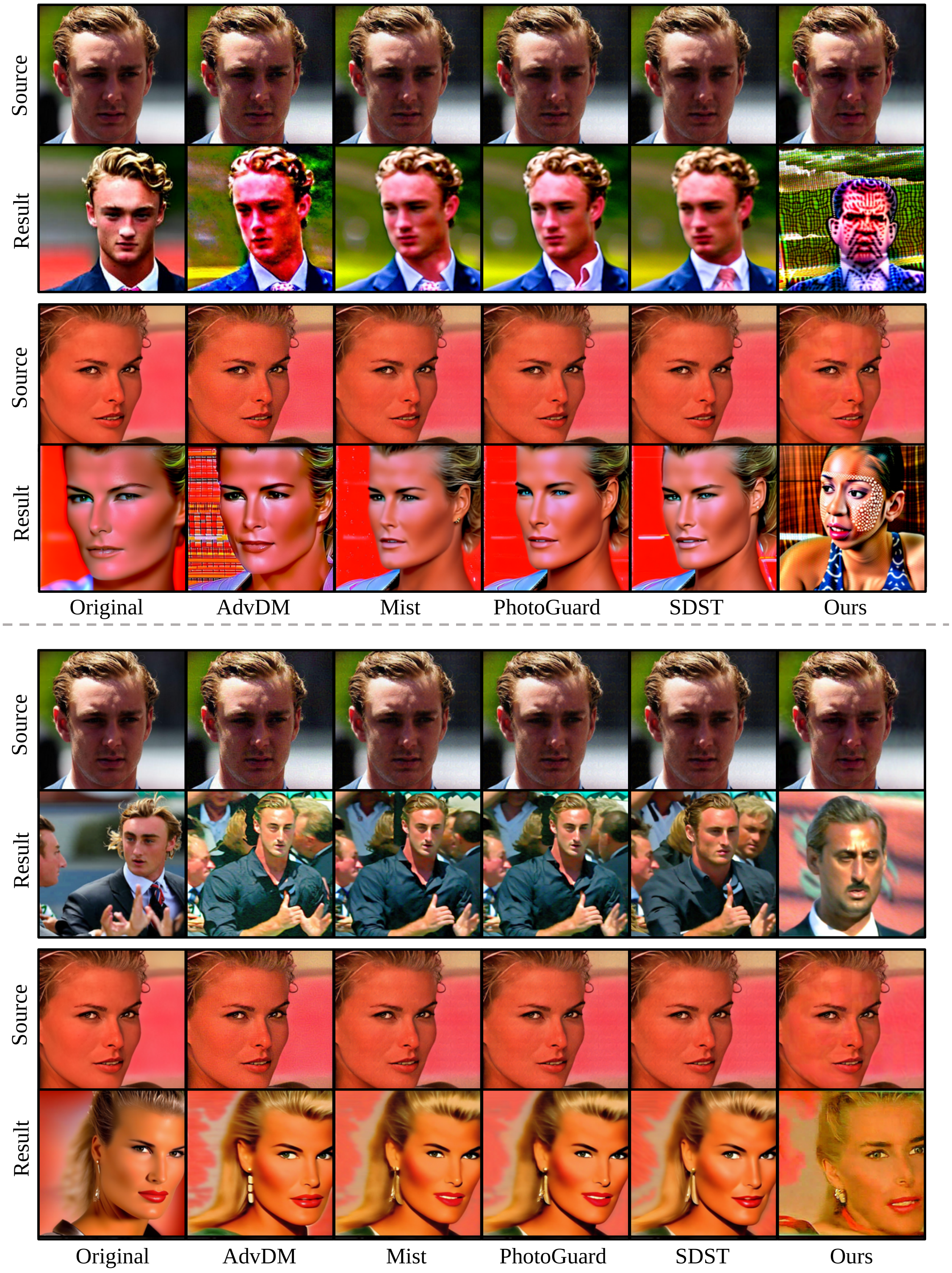


Figure 22. Qualitative comparison with baselines on the SD 1.5-based IP-Adapter **ImageVariation** version (top) and SDXL-based IP-Adapter **ImageVariation** version (bottom).

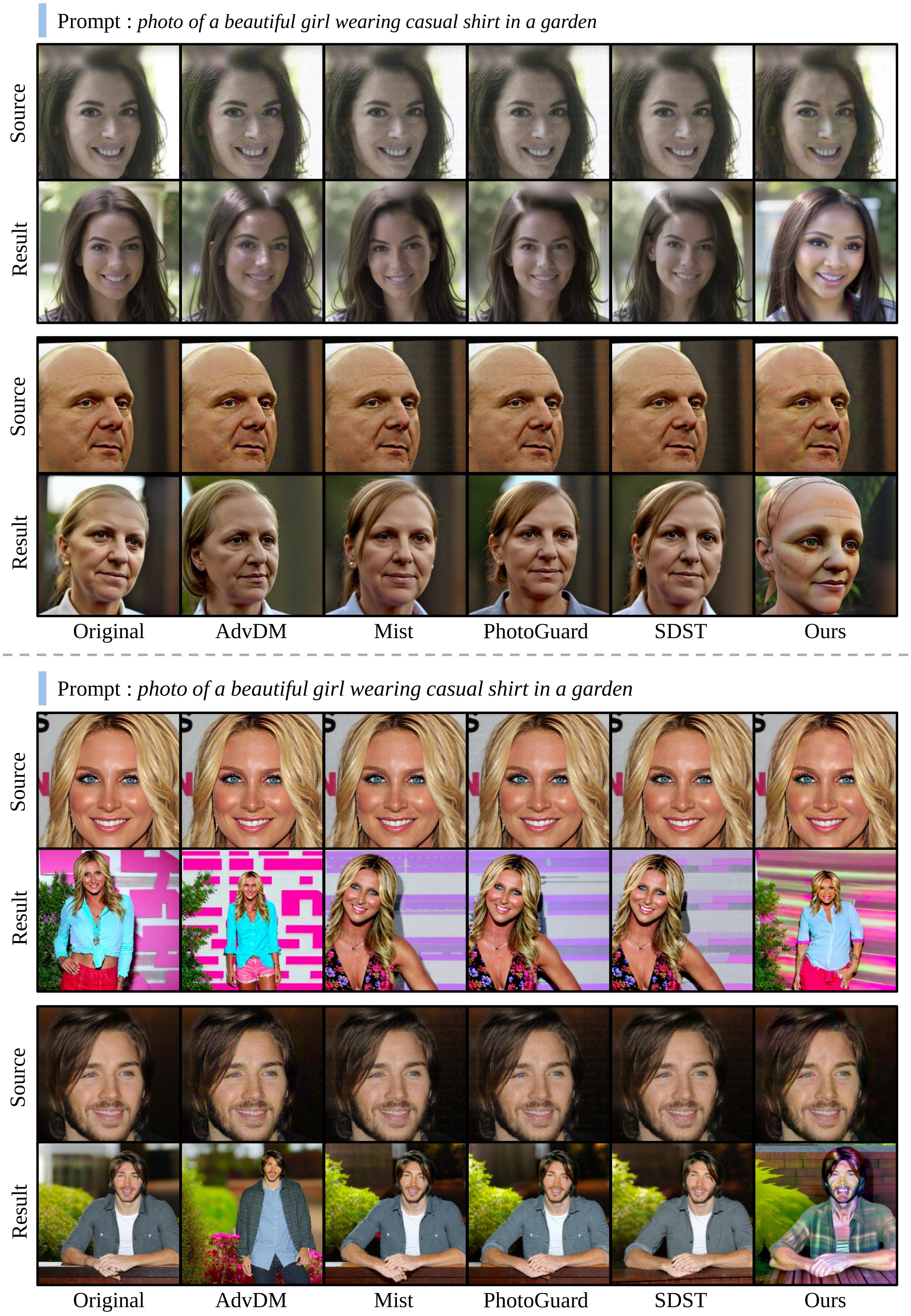

Prompt : *photo of a beautiful girl wearing casual shirt in a garden*
Source
Result
Source
Result
Original
AdvDM
Mist
PhotoGuard
SDST
Ours
Prompt : *photo of a beautiful girl wearing casual shirt in a garden*
Source
Result
Source
Result
Original
AdvDM
Mist
PhotoGuard
SDST
Ours

Figure 23. Qualitative comparison with baselines on the SD 1.5-based IP-Adapter **Multi-modal prompts** version (top) and SDXL-based IP-Adapter **Multi-modal prompts** version (bottom).

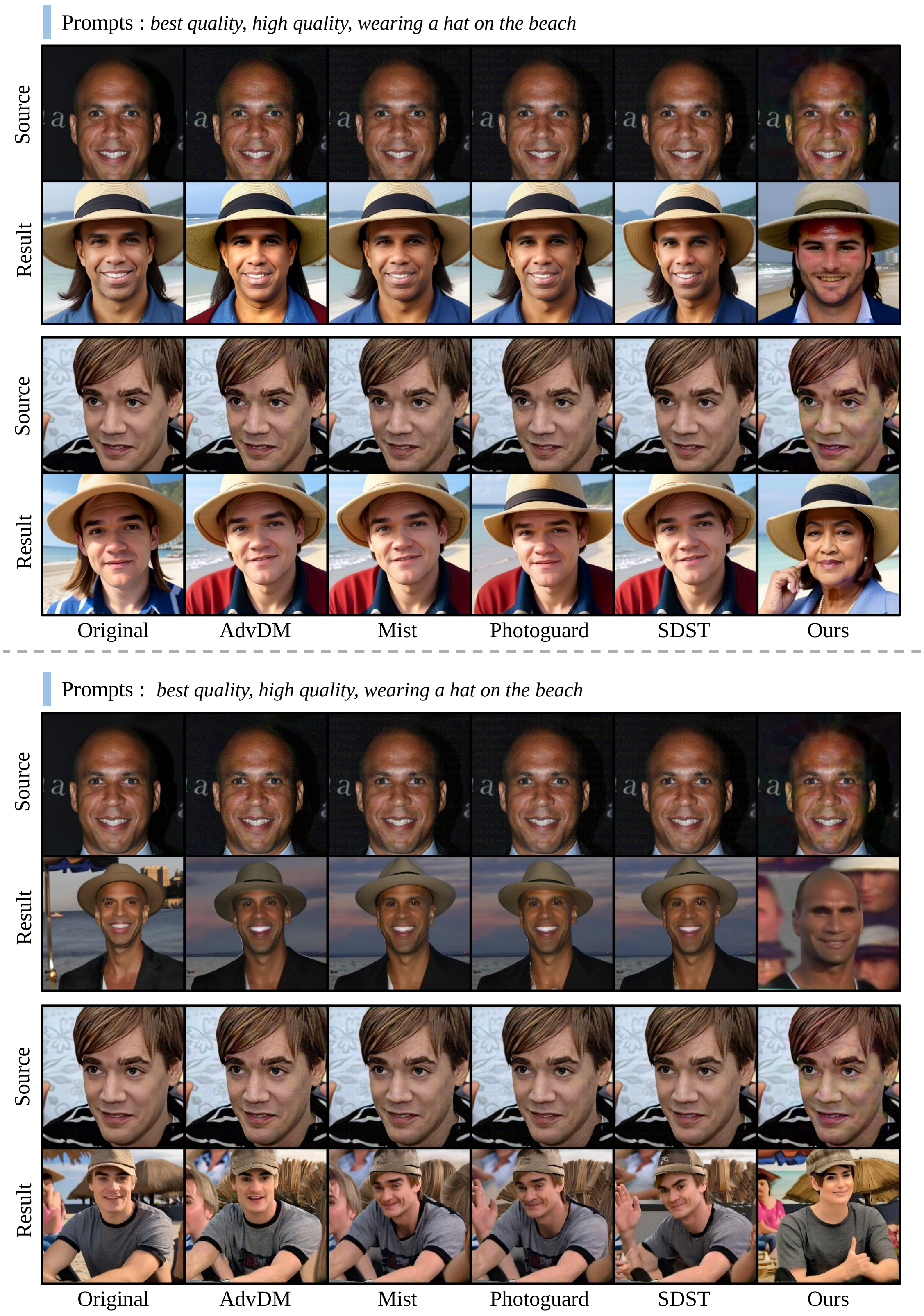

Prompts : *best quality, high quality, wearing a hat on the beach*
Source
Result
Source
Result
Original
AdvDM
Mist
Photoguard
SDST
Ours
Prompts : *best quality, high quality, wearing a hat on the beach*
Source
Result
Source
Result
Original
AdvDM
Mist
Photoguard
SDST
Ours

Figure 24. Qualitative comparison with baselines on the SD 1.5-based IP-Adapter **Plus** version (top) and the SDXL-based IP-Adapter **Plus Face** version (bottom).

References

[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736, 2022. 19

[2] Renwang Chen, Xuanhong Chen, Bingbing Ni, and Yanhao Ge. Simswap: An efficient framework for high fidelity face swapping. In *Proceedings of the 28th ACM international conference on multimedia*, pages 2003–2011, 2020. 9, 18

[3] Xuanhong Chen, Bingbing Ni, Yutian Liu, Naiyuan Liu, Zhilin Zeng, and Hang Wang. Simswap++: Towards faster and high-quality identity swapping. *IEEE Trans. Pattern Anal. Mach. Intell.*, 46(1):576–592, 2024. 7

[4] June Suk Choi, Kyungmin Lee, Jongheon Jeong, Saining Xie, Jinwoo Shin, and Kimin Lee. Diffusionguard: A robust defense against malicious diffusion-based image editing. *arXiv preprint arXiv:2410.05694*, 2024. 4

[5] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8789–8797, 2018. 4

[6] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4690–4699, 2019. 4

[7] Ranjie Duan, Yuefeng Chen, Dantong Niu, Yun Yang, A Kai Qin, and Yuan He. Advdrop: Adversarial attack to dnns by dropping information. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7506–7515, 2021. 4

[8] Gege Gao, Huaibo Huang, Chaoyou Fu, Zhaoyang Li, and Ran He. Information bottleneck disentanglement for identity swapping. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3404–3413, 2021. 9, 18

[9] Chuan Guo, Jared S Frank, and Kilian Q Weinberger. Low frequency adversarial perturbation. *arXiv preprint arXiv:1809.08758*, 2018. 4

[10] Zhenliang He, Wangmeng Zuo, Meina Kan, Shiguang Shan, and Xilin Chen. Attgan: Facial attribute editing by only changing what you want. *IEEE transactions on image processing*, 28(11):5464–5478, 2019. 4

[11] Hao Huang, Yongtao Wang, Zhaoyu Chen, Yuze Zhang, Yuheng Li, Zhi Tang, Wei Chu, Jingdong Chen, Weisi Lin, and Kai-Kuang Ma. Cmua-watermark: A cross-model universal adversarial watermark for combating deepfakes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 989–997, 2022. 4

[12] Tero Karras. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017. 7

[13] Edgar Kaziakhmedov, Klim Kireev, Grigorii Melnikov, Mikhail Pautov, and Aleksandr Petiushko. Real-world attack on mtcnn face detection system. In *2019 International Multi-Conference on Engineering, Computer and Information Sciences (SIBIRCON)*, pages 0422–0427. IEEE, 2019. 4

[14] Kihong Kim, Yunho Kim, Seokju Cho, Junyoung Seo, Jisu Nam, Kychul Lee, Seung Wook Kim, and Kwanghee Lee. Diffface: Diffusion-based face swapping with facial guidance. *Pattern Recognit.*, 163:111451, 2022. 4, 5, 6, 9, 11, 12, 13, 17

[15] Nupur Kumari, Bingliang Zhang, Richard Zhang, Eli Shechtman, and Jun-Yan Zhu. Multi-concept customization of text-to-image diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1931–1941, 2023. 19

[16] Xinyang Li, Shengchuan Zhang, Jie Hu, Liujuan Cao, Xiaopeng Hong, Xudong Mao, Feiyue Huang, Yongjian Wu, and Rongrong Ji. Image-to-image translation via hierarchical style disentanglement. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8639–8648, 2021. 4

[17] Yuezun Li, Pu Sun, Honggang Qi, and Siwei Lyu. Landmarkbreaker: A proactive method to obstruct deepfakes via disrupting facial landmark extraction. *Computer Vision and Image Understanding*, 240:103935, 2024. 4

[18] Chumeng Liang and Xiaoyu Wu. Mist: Towards improved adversarial examples for diffusion models. *arXiv preprint arXiv:2305.12683*, 2023. 4, 5, 9, 11, 12, 13

[19] Chumeng Liang, Xiaoyu Wu, Yang Hua, Jiaru Zhang, Yiming Xue, Tao Song, Zhengui Xue, Ruhui Ma, and Haibing Guan. Adversarial example does good: Preventing painting imitation from diffusion models via adversarial examples. *arXiv preprint arXiv:2302.04578*, 2023. 4, 5, 9, 11, 12, 13

[20] Jie Ling, Jinhui Chen, and Honglei Li. Fdt: Improving the transferability of adversarial examples with frequency domain transformation. *Computers & Security*, page 103942, 2024. 4

[21] Shishira R Maiya, Max Ehrlich, Vatsal Agarwal, Ser-Nam Lim, Tom Goldstein, and Abhinav Shrivastava. A frequency perspective of adversarial robustness. *arXiv preprint arXiv:2111.00861*, 2021. 4

[22] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2574–2582, 2016. 4

[23] Shengju Qian, Keqiang Sun, Wayne Wu, Chen Qian, and Jiaya Jia. Aggregation via separation: Boosting facial landmark detector with semi-supervised style translation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10153–10163, 2019. 4

[24] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 4, 19

[25] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695, 2022. 4, 19

[26] Nataniel Ruiz, Sarah Adel Bargal, and Stan Sclaroff. Disrupting deepfakes: Adversarial attacks against conditional image translation networks and facial manipulation systems. In *Computer Vision–ECCV 2020 Workshops: Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16*, pages 236–251. Springer, 2020. 4

[27] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 22500–22510, 2023. 19

[28] Hadi Salman, Alaa Khaddaj, Guillaume Leclerc, Andrew Ilyas, and Aleksander Madry. Raising the cost of malicious ai-powered image editing. In *International Conference on Machine Learning*, 2023. 4, 5, 9, 11, 12, 13

[29] Shawn Shan, Jenna Cryan, Emily Wenger, Haitao Zheng, Rana Hanocka, and Ben Y Zhao. Glaze: Protecting artists from style mimicry by {Text-to-Image} models. In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 2187–2204, 2023. 4

[30] Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, and Michael K Reiter. Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In *Proceedings of the 2016 acm sigsac conference on computer and communications security*, pages 1528–1540, 2016. 4

[31] Yash Sharma, Gavin Weiguang Ding, and Marcus A. Brubaker. On the effectiveness of low frequency perturbations. In *International Joint Conference on Artificial Intelligence*, 2019. 4

[32] Hao Tang, Dan Xu, Nicu Sebe, and Yan Yan. Attention-guided generative adversarial networks for unsupervised image-to-image translation. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2019. 4

[33] Feifei Wang. Face swap via diffusion model. *arXiv preprint arXiv:2403.01108*, 2024. 4, 5, 6, 9, 11, 12, 13, 14

[34] Haohan Wang, Xindi Wu, Zeyi Huang, and Eric P Xing. High-frequency component helps explain the generalization of convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8684–8694, 2020. 4

[35] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, et al. Deep high-resolution representation learning for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 43(10):3349–3364, 2020. 4

[36] Run Wang, Ziheng Huang, Zhikai Chen, Li Liu, Jing Chen, and Lina Wang. Anti-forgery: Towards a stealthy and robust deepfake disruption attack via adversarial perceptual-aware perturbations. *arXiv preprint arXiv:2206.00477*, 2022. 4

[37] Haotian Xue, Chumeng Liang, Xiaoyu Wu, and Yongxin Chen. Toward effective protection against diffusion-based mimicry through score distillation. In *The Twelfth International Conference on Learning Representations*, 2023. 4, 5, 9, 11, 12, 13

[38] Hu Ye, Jun Zhang, Sibo Liu, Xiao Han, and Wei Yang. Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models. *arXiv preprint arXiv:2308.06721*, 2023. 5, 6, 9, 10, 11, 12, 13, 15, 19

[39] Xi Yin, Ying Tai, Yuge Huang, and Xiaoming Liu. Fan: Feature adaptation network for surveillance face recognition and normalization. In *Proceedings of the Asian Conference on Computer Vision*, 2020. 4

[40] Chongyang Zhang, Yu Qi, and Hiroyuki Kameda. Multi-scale perturbation fusion adversarial attack on mtcnn face detection system. In *2022 4th International Conference on Communications, Information System and Computer Engineering (CISCE)*, pages 142–146. IEEE, 2022. 4, 7

[41] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE signal processing letters*, 23(10):1499–1503, 2016. 4, 7

[42] Wenliang Zhao, Yongming Rao, Weikang Shi, Zuyan Liu, Jie Zhou, and Jiwen Lu. Diffswap: High-fidelity and controllable face swapping via 3d-aware masked diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8568–8577, 2023. 4, 5, 6, 9, 11, 12, 13, 16

[43] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017. 4