Appendix: Diffusion Face Relighting

A. Overview

In this Appendix, we present:

- Section B: Implementation details.
- Section C: Network architectures.
- Section E: Additional results.
- Section F: Additional related work.
- Section G: Potential negative societal impacts.

B. Implementation details

B.1. Datasets

For all experiments in Section 4.1, we trained our network on the FFHQ dataset [25], which consists of 70,000 aligned face images (60k for training and 10k for testing). We evaluated the relighting performance on Multi-PIE dataset [17], which contains 337 subjects captured under 19 flashes. In "self target lighting," we use the same test set as [22], which contains pairs of images from the same person but in different lighting. For "target lighting from others," we randomly pick 200 triplets of the input, target, and ground truth, where the target image is of a different person. For all ablation studies (Section 4.2), to cap the computational resources, each ablated variation is trained on the FFHQ dataset at 128×128 resolution and evaluated on Multi-PIE dataset. For evaluation, we randomly pick 200 pairs, using the same policy as the "self target lighting," from the disjoint set of other experiments in Section 4.1.

B.2. Training and Inference

We normalize the training images to [-1,1], and precompute their encodings from DECA, ArcFace, and our shadow estimator. We train our DDIM and Modulator using training hyperparameters in Table 3.

128×128 resolution. We used four Nvidia RTX2080Tis for training and one Nvidia RTX2080Ti for testing. The training took around 1 day using batch size 32, and the inference took 101.38 ± 0.64s per image.

256×256 resolution. We used four Nvidia V100s for training and one Nvidia RTX2080Ti for testing. The training took around 8 days using batch size 20, and the inference took 194.29 ± 9.17s per image.

B.3. Improved DDIM sampling with mean-matching

We observe that when the input image contains background pixels with extreme intensities (e.g., too dark or too bright), the output tends to have a slight change in the overall brightness, most noticeable in the background (see Figure 15). This behavior also occurs with DDIM inversion that involves no relighting, i.e., when we reverse $\mathbf{x}_T = \text{DDIM}^{-1}(\mathbf{x}_0)$ and decode $\mathbf{x}'_0 = \text{DDIM}(\mathbf{x}_T)$ without modifying the light encoding, $\mathbf{x}'_0$ can look slightly different from $\mathbf{x}_0$ in terms of the overall brightness.

We found that we can correct the overall brightness with a simple, global brightness adjustment within DDIM's generative process as follows. We first perform self-reconstruction by running DDIM's reverse generative process starting from the input $\mathbf{x}_0$ to produce $\mathbf{x}_0, \mathbf{x}_1, ..., \mathbf{x}_T$, then decoding back $\mathbf{x}'_T, \mathbf{x}'_{T-1}, ..., \mathbf{x}'_0$, where $\mathbf{x}'_T = \mathbf{x}_T$ using Equation 4 in the main paper and its reverse. Then, our correction factor sequence, $\mu_0, \mu_1, ..., \mu_T$, is computed by taking the difference between the mean pixel values of $\mathbf{x}$ and $\mathbf{x}'$:

$$\mu_t = \text{mean}(\mathbf{x}'_t) - \text{mean}(\mathbf{x}_t). \tag{6}$$

We compute the mean separately for each RGB channel and compute this correction sequence *once* for each input image. Then, during relighting, we add μ_t to the generative process conditioned on the modified feature vector, starting from $\mathbf{x}_T$. That is, we use the reverse of Equation 4 in the main paper to first produce $\mathbf{x}_{T-1}$ from $\mathbf{x}_T$, and add μ_{T-1} to it: $\mathbf{x}_{T-1} \leftarrow \mathbf{x}_{T-1} + \mu_{T-1}$. Then, we continue the process until we obtain the relit output at $t = 0$.

C. Network architectures

C.1. Conditional DDIM & Modulator

Our conditional DDIM architecture is based on [10] with hyperparameters stated in Table 3. Each residual block in the first half of the network uses both spatial conditioning and non-spatial conditioning (Figure 9), whereas each residual block in the later half only uses the non-spatial conditioning. The Modulator has the same architecture and hyperparameters as the first half of conditional DDIM but without the non-spatial and spatial conditioning.

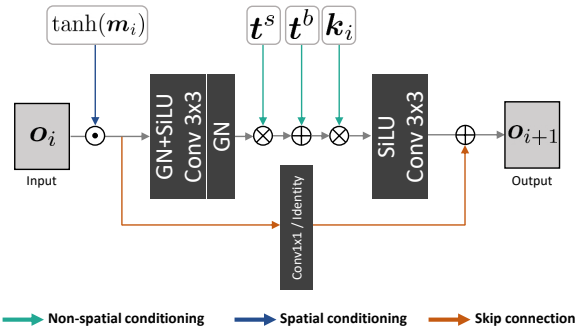

Figure 9: Diagram of one of the residual blocks inside the first half of our conditional DDIM.

Table 3: Our conditional DDIM's configuration is based on the architecture of [10].

Parameter	FFHQ 128	FFHQ 256
Base channels	128	128
Channel multipliers	[1,1,2,3,4]	[1,1,2,2,4,4]
Attention resolution	[16, 8]	[16, 8]
Batch size	32	20
Image trained	1.6M	1.7M
Diffusion step	1000	
Learning rate	1e-4	
Weight decay	-	
Noise scheduler	Linear	
Optimizer	AdamW	

C.2. Non-spatial encoding

The concatenation of $(\mathbf{s}, \mathbf{cam}, \boldsymbol{\xi}, c)$ is passed through 3-layer MLPs (Figure 10). For each MLP_i, we use fixed-dimension hidden layers $\mathbf{k}_i^1, \mathbf{k}_i^2 \in \mathbb{R}^{512}$, while the dimension of each $\mathbf{k}_i$ depends on the channel dimension of each residual block.

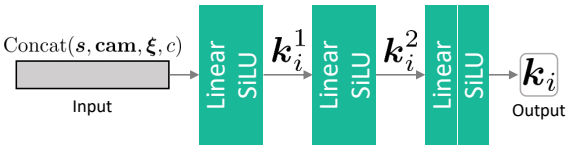

Figure 10: Diagram of one of the 3-layer MLPs in the non-spatial conditioning branch.

D. 3D face rendering

We compute the shading reference R used in the spatial conditioning by:

$$R_{i,j} = A \odot \sum_{k=1}^{9} \mathbf{l}_k H_k(N_{i,j}), \tag{7}$$

where i, j denote pixel (i, j) in image space, $A = [0.7, 0.7, 0.7]$ is a constant gray albedo, $l_k \in \mathbb{R}^3$ is the k-th second-order spherical harmonic RGB coefficient predicted from DECA, $H_k : \mathbb{R}^3 \to \mathbb{R}$ is the k-th spherical harmonic basis function, $N_{i,j} \in \mathbb{R}^3$ is the normalized surface normal at pixel (i, j).

E. Additional results

In this section, we provide additional results:

- Table 4 shows full statistics of Table 1 (Top, main paper) with standard errors, as well as more results from additional baselines [53, 76, 55].
- Figure 11 shows qualitative results of the ablation study on light conditioning (Section 4.3A).
- Figure 12 shows qualitative results of the ablation study on non-spatial conditioning (Section 4.3B).
- Figure 13 and 14 show additional qualitative results on the FFHQ dataset.
- Figure 15 shows a qualitative comparison for the ablation study of the mean-matching algorithm (Section B.3).
- Figure 16 shows more qualitative results on cast shadow manipulation.

E.1. Comparison with relighting methods that use HDR environment maps

In this section, we compare our method to [38], which represents a class of relighting techniques that take an HDR environment map as input [61, 38, 72]. We found that none of these methods [61, 38, 72] released their source code, and their datasets are proprietary. Nonetheless, we requested the authors of these methods to test their algorithms on standard Multi-Pie and FFHQ. Only Pandey et al. [38] provided us with their results, and we consider [38] to be a state-of-the-art representative for this class of techniques as [38] has "on-par" performance to [72] and already outperforms [61]. The quantitative results of this experiment are shown in Table 1 (main paper), and the qualitative results are shown in Figure 4, 5, and 3 (main paper), as well as Figure 13 and 14 in Appendix E. We would like to emphasize again that Pandey et al. [38] solve a different problem setup and require the input environment map to be first estimated from the target image. In our experiment, the results of Pandey et al. [38] were generated by the authors themselves, including the estimated environment maps.

F. Additional related work

Conditional DDPMs. Diffusion models (DDPMs) [19] and scored-based models [58, 60] have been used to solve multiple conditional generation tasks [7], such as conditional image synthesis [10, 21, 57, 6], image-to-image translation [50], image super-resolution [20, 37], image segmentation [1, 3] and image manipulation [41, 34]. Many recent approaches use cross-modal embeddings from popular language models [42, 66, 43] as conditions for diffusion models [44, 51, 48, 49, 36, 2, 40, 74], which enables general text-to-image generation and image manipulation. However, they lack the ability to precisely manipulate lighting attributes or directions. DiffAE [41] conditions a DDIM with a 1D latent vector that is learned to capture semantically meaningful information. Manipulating this novel la-

Table 4: **State-of-the-art comparison on Multi-PIE.** We report the means and standard errors. Our method outperforms all previous methods on all metrics with p-values < 0.001.

Method	DDSIM↓		MSE↓		LPIPS↓	
	Mean	SE	Mean	SE	Mean	SE
SfSNet[53]	0.2918	0.0013	0.0961	0.0017	0.5222	0.0025
DPR[77]	0.1599	0.0019	0.0852	0.0018	0.2644	0.0028
SIPR[61]	0.1539	0.0015	0.0166	0.0004	0.2764	0.0025
Nestmayer et al.[35]	0.2226	0.0046	0.0588	0.0018	0.3795	0.0078
Pandey et al.[38]	0.0875	0.0007	0.0165	0.0003	0.2010	0.0022
Hou et al.(CVPR'21)[23]	0.1186	0.0013	0.0303	0.0006	0.2013	0.0023
Hou et al.(CVPR'22)[22]	0.0990	0.0013	0.0150	0.0004	0.1622	0.0017
Ours	**0.0711**	0.0011	**0.0122**	0.0005	**0.1370**	0.0020

tent vector allows manipulation of various semantic face attributions. Unlike DiffAE, which implicitly models semantic attributes via a learnable latent code, our method requires an explicit and interpretable light encoding, which can be controlled by the user.

Single-view 3D face modeling. Our work uses DECA [14] to estimate the 3D shape and spherical harmonic lighting information. Based on the pioneer work of Blanz and Vetter [5], DECA regresses the parameters of a FLAME model [27], which represents the face shape with three linear bases corresponding to the identity shape, pose, and expression, and further recovers person-specific details that can change with expression. Our work only uses the FLAME estimate from DECA without the additional facial details. Note that other 3D face modeling techniques, such as [9, 15, 24, 29], can also be used in our framework.

Face recognition model for deep face embedding. Our work leverages a face recognition model, ArcFace [8], to preserve the identity of the relit face. Most previous face recognition models are trained using softmax loss [62, 39, 33] and triplet loss [52, 30] (See [67] for a review.) However, they do not generalize well with open-set recognition and large scale recognition. ArcFace adopts Additive Angular Margin loss, which retains discriminativeness while avoiding the sampling problem of the triplet loss. Arcface also proposed a sub-center procedure, which helps improve the robustness of the embedding. Note again that other face embedding models, such as [62, 39, 33, 30], can also be used in our framework.

G. Potential negative societal impacts

Our method can be used for changing the lighting condition of an existing image and producing the so-called DeepFake, which can deceive human visual perception. Our manipulation process is based on conditional DDIM [10], and a study from [41], which uses the same architecture, shows that certain artifacts from DDIM can be currently detected using a CNN with about 92% accuracy. We developed our work with the intention of promoting positive and creative uses, and we do not condone any misuse of our work.

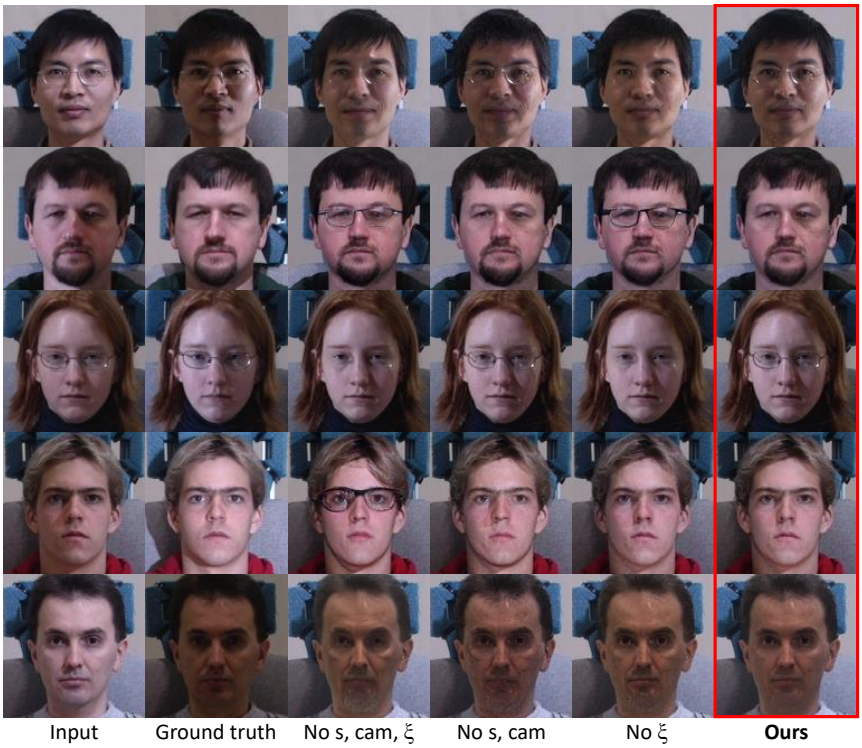

Figure 11: **Ablation study of the light conditioning** (Section 4.3A in the main text).

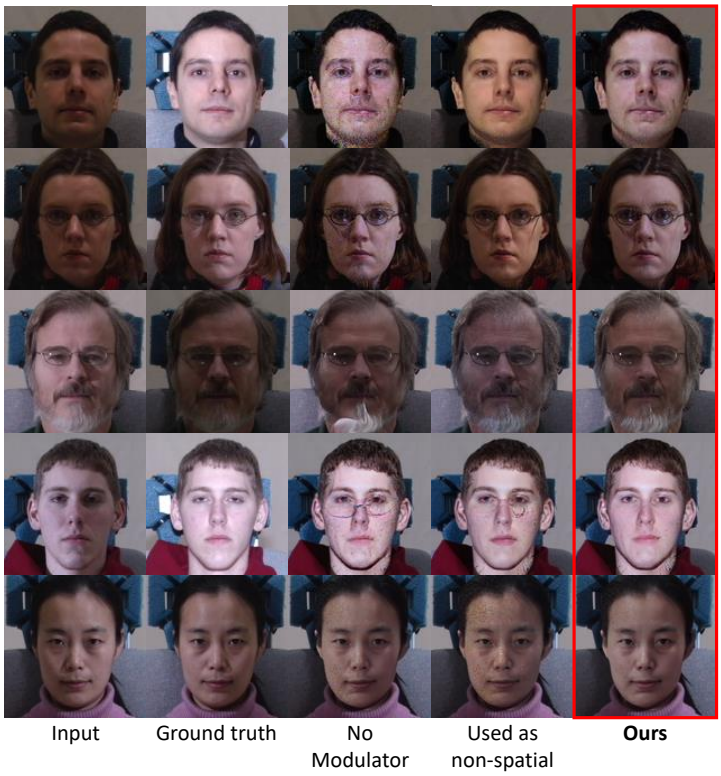

Figure 12: **Ablation study of the non-spatial conditioning variable** (Section 4.3B in the main text).

Figure 13: **Relit images from FFHQ [25].**

Figure 14: **Relit images from FFHQ [25].**

Figure 15: **Improved DDIM sampling with mean-matching.** We show a qualitative comparison between"with" and "without" mean-matching. Our mean-matching technique helps correct the overall brightness in both the inversion output and relit image.

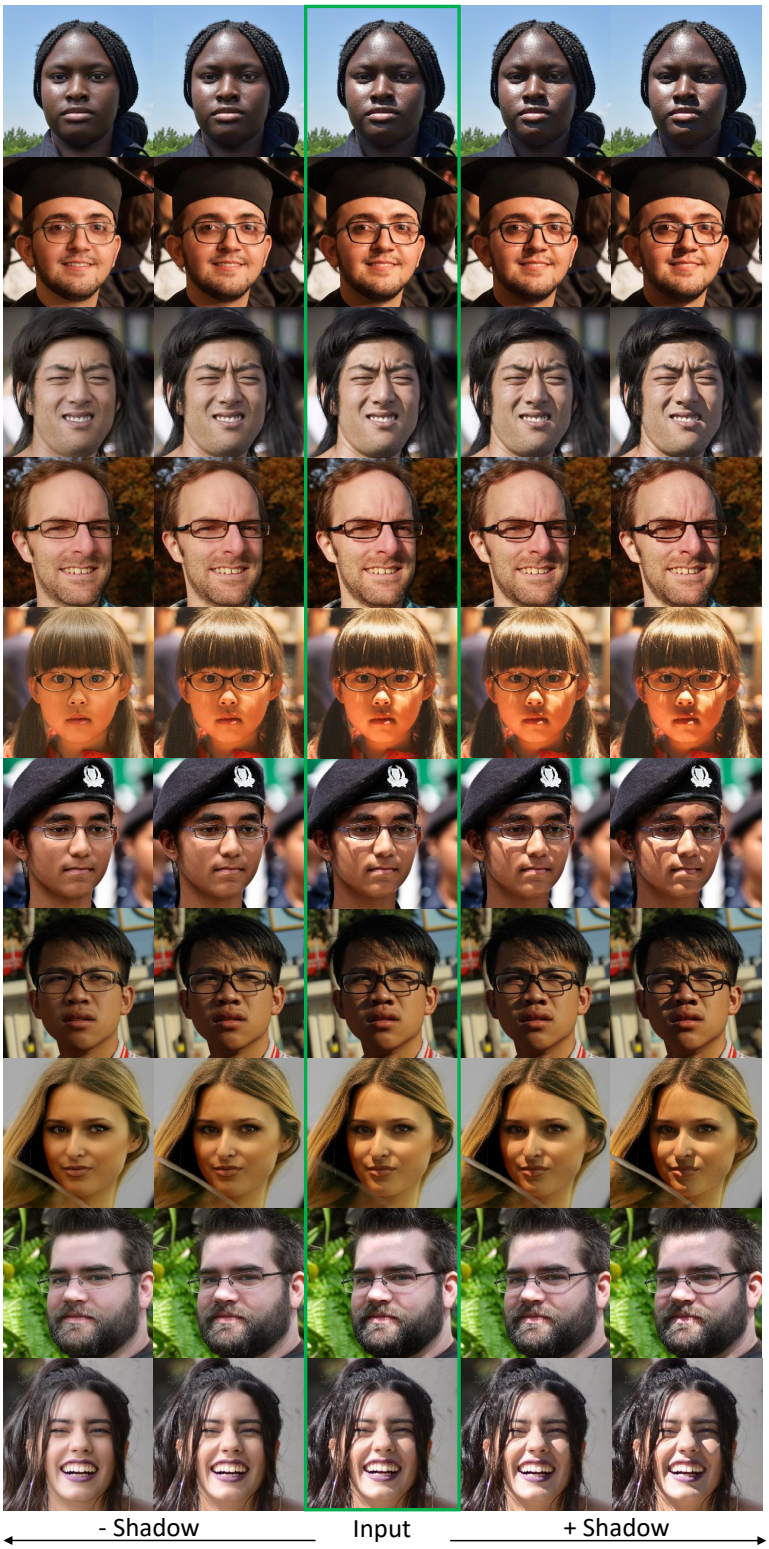

Figure 16: **Varying the degree of cast shadows.**