DiFaReli: Diffusion Face Relighting

Puntawat Ponglertnapakorn  Nontawat Tritrong  Supasorn Suwajanakorn
VISTEC, Thailand

Figure 1: Relighting and shadow manipulation. Our method can realistically relight an input image using different target lights shown in small shading references. We can add or remove cast shadows, as well as control their intensity.

Abstract

We present a novel approach to single-view face relighting in the wild. Handling non-diffuse effects, such as global illumination or cast shadows, has long been a challenge in face relighting. Prior work often assumes Lambertian surfaces, simplified lighting models or involves estimating 3D shape, albedo, or a shadow map. This estimation, however, is error-prone and requires many training examples with lighting ground truth to generalize well. Our work bypasses the need for accurate estimation of intrinsic components and can be trained solely on 2D images without any light stage data, multi-view images, or lighting ground truth. Our key idea is to leverage a conditional diffusion implicit model (DDIM) for decoding a disentangled light encoding along with other encodings related to 3D shape and facial identity inferred from off-the-shelf estimators. We also propose a novel conditioning technique that eases the modeling of the complex interaction between light and geometry by using a rendered shading reference to spatially modulate the DDIM. We achieve state-of-the-art performance on standard benchmark Multi-PIE and can photorealistically relight in-the-wild images. Please visit our page: https://diffusion-face-relighting.github.io.

1. Introduction

The ability to relight face images under any lighting condition has a wide range of applications, such as in Augmented Reality, where consistent lighting for all individuals in the scene is essential to achieve realism. Another use is in portrait photography, where one may aim to soften cast shadows to create a more pleasing, diffuse appearance. Yet, relighting single-view face images remains unsolved.

Relighting a face image requires modeling the physical interactions between the geometry, material, and lighting, which are not inherently present in a 2D image and difficult to estimate accurately. Earlier work [4, 53, 26, 68, 56] thus often assumes Lambertian surfaces and a simplified lighting model, which struggle to model complex light interactions like global illumination, subsurface scattering, or cast shadows. Using multi-view, multi-illumination data from a light stage or a simulation, [38, 72] proposed relighting pipelines that predict surface normals, albedo, and a set of diffuse and specular maps with neural networks given a target HDR map. Some recent methods aim to specifically model cast shadows by predicting a shadow map with a neural network [23, 35] or rendering a shadow map through physical ray tracing with estimated geometry [22].
These approaches share a common scheme in which they first intrinsically decompose the face image into its surface normals, albedo, and lighting parameters, then use them along with a shadow or visibility map to render a relit output. However, one major issue of this scheme stems from its over-reliance on the accuracy of the estimated components, which are difficult to estimate correctly in real-world scenarios. For instance, when an input image contains cast shadows that need to be removed, these approaches often leave behind shadow residuals in the predicted albedo map, which in turn produces artifacts in the final output (Figure 3). Estimating the geometry for other areas like hair and ears is also extremely challenging, and they are often omitted from relighting pipelines, resulting in unrealistic final composites (Figure 3, 4 and 5).

This paper introduces an alternative approach that does not rely on accurate intrinsic decomposition of the face and can be trained exclusively on 2D images, without any 3D face scan, multi-view images, or lighting ground truth, once given a few off-the-shelf estimators. The general idea of our method is simple: we first encode the input image into a feature vector that disentangles the light information from other information about the input image. Then, we modify the light encoding in the feature vector and decode it. The challenge, however, is how to disentangle the light encoding well enough so that the decoding will only affect the shading without altering the person’s shape and identity. Our key idea is to leverage a conditional diffusion implicit model [59] with a novel conditioning technique for this task and learn the complex light interactions implicitly via the generative model trained on a real-world 2D face dataset.

Our method relies on mechanisms recently introduced in Denoising Diffusion Implicit Models (DDIM) [59] and Diffusion Autoencoders (DiffAE) [41]. By exploiting the deterministic reversal process of DDIM proposed by Song et al. [59], DiffAE shows how one can encode an image into a meaningful semantic code and disentangle it from other information, which includes stochastic variations. By modifying the semantic code and decoding it, DiffAE can manipulate semantic attributes in a real image. Relighting can be thought of as a manipulation of the “light” attribute in the input image. But unlike DiffAE, which discovers semantic attributes automatically and encodes them in a latent code, our method requires an explicit and interpretable light encoding that facilitates lighting manipulation by the user.

To solve this problem without access to the lighting ground truth, we use an off-the-shelf estimator, DECA [13], to encode the lighting information as spherical harmonic (SH) coefficients and rely on a conditional DDIM to decode and learn to disentangle the light information in the process. Unlike prior work, our use of SH lighting is not for direct rendering of the output shading, as this would be restricted by the limited capacity of SH lighting to express complex illumination. Rather, it is used to condition a generative process that learns the complex shading prior to reproduce real-world 2D face images. To help preserve the input’s identity during relighting, we also condition the DDIM on other attributes, such as the face shape and deep feature embeddings from a face recognition model, ArcFace [8].

Another key component is our novel technique for conditioning the DDIM. Instead of treating the SH lighting as a global, non-spatial condition vector as in DiffAE or other diffusion models, we render a shading reference using the known SH equation and feed it to another network called
Modulator, which computes layer-wise spatial modulation weights for the DDIM. This conditioning technique helps retain spatial information in the shading reference and provides an easy-to-learn conditioning signal as the pixel intensities in the shading reference correlate more directly with the output RGB pixels.

With our novel framework, the visibility of cast shadows can also be modeled with a simple modification: add one conditioning scalar that indicates the “degree” of cast shadows to the DDIM. At test time, we can strengthen or attenuate cast shadows by modifying this scalar. Since our diffusion-based framework does not directly use this flag or the shape and SH parameters in a physical image formation model, imprecise estimation of these parameters can be tolerated and does not significantly compromise our quality.

Our method produces highly plausible and photorealistic results and can convincingly strengthen or attenuate cast shadows. Moreover, we can reproduce the original facial details with high fidelity, which is difficult for competing methods that predict an albedo map with neural networks. We conduct qualitative and quantitative evaluations and achieve state-of-the-art performance on a standard benchmark, Multi-Pie [17]. To summarize, our contributions are:

- A state-of-the-art face relighting framework based on a conditional DDIM that produces photorealistic shading without requiring accurate intrinsic decomposition or 3D and lighting ground truth.
- A novel conditioning technique that converts a shading reference rendered from the estimated light and shape parameters into layer-wise spatial modulation weights.

2. Related work

A common approach to face relighting [4, 65, 26, 53, 68, 56] is to decompose an input image into multiple intrinsic components (e.g., lighting, albedo, surface normals) and recompose the image back with modified light-related components. The decomposition can be done by regularized optimization [4], fitting a morphable model [5, 68], or predicted from a neural network [53, 26, 65, 35, 69, 38, 56]. Most earlier methods [4, 53, 26, 68, 56] assume Lambertian surfaces, a simplified lighting model, such as second-order spherical harmonics, and a physical image formation model based on these simplified assumptions. Thus, they cannot handle non-diffuse effects, such as specular highlights or cast shadows, commonly occur in real-world scenarios.

Rather than decomposing an image into physical components, some techniques [77, 61] rely on an encoder-decoder network with a bottleneck layer that holds a latent lighting representation. Zhou et al. [77] force such a latent code to be predictive of the SH lighting and train another regressor that can map the SH lighting of a reference image back to a latent code for relighting. Sun et al. [61] rely on a similar idea but use a low-resolution illumination map, obtained from a light stage, e.g., [70], instead of the SH lighting. In principle, these learning techniques can learn to handle hard shadows and specularities, given sufficient examples. However, in practice, these approaches still struggle to model those effects due to their small light stage data [61] or limited variations in their synthetic dataset [77]. In contrast, our framework can be trained on 2D face images, which are cheaply available and cover far more diverse scenarios.

Handling non-diffuse components. Relighting non-diffuse components has long been a challenge. Nestmeyer et al. [35] propose a two-stage framework to predict non-diffuse components as a residual correction of a diffuse rendering from their first stage. Cast shadows are predicted separately as a visibility map, which is multiplied to the output. Wang et al. [69] propose a technique based on intrinsic decomposition that predicts shadow and specular maps by learning from their own large-scale relighting dataset. Pandey et al. [38] introduces a pipeline that predicts a set of specular maps with varying degrees of Phong exponents using estimated surface normals and an input HDR environment map. These maps along with diffuse and albedo maps are used to predict a relit image with a UNet. Yeh et al. [72] uses a pipeline similar to [38] but with synthetic light stage data generated from 3D face scans, and an albedo refinement step to reduce the domain gap between synthetic and real data. Hou et al. (2021) [23] compute a shadow map based on a morphable model fitted to the input and standard ray tracing, then use it to help predict the ratio of pixel luminance changes for relighting. Hou et al. (2022) [22] predict a shadow mask via ray tracing based on their estimated depth map and render a relit image with estimated albedo and shading maps from neural networks.

While these methods [35, 23, 22] produce promising non-diffuse effects, their physical image formation model makes it difficult to operate in the wild when the estimated geometry is inaccurate. As a result, some [23, 22] can only relight the face region but not the ears or hair and still struggle to handle in-the-wild cast shadows (Figure 3). Neural rendering approaches [69, 38] can tolerate some estimation error, but high-frequency details are often lost, even when predicted by a UNet [69] and still require light stage data. The synthetic light stage data of [72] shows great potential but currently relies on 3D face scans to generate, which are difficult to obtain compared to 2D images.

Style transfer-based methods. Another class of relighting approaches is based on style transfer. Although some of these methods [28, 31, 54] do not directly solve face relighting, they can be adapted for this task by transferring the lighting and shading styles from one image to another. However, the style representation used in these methods captures broad information beyond the lighting condition
and cannot produce accurate relit results. Shu et al. [55] solves relighting using color histogram matching that is re-designed be spatially varying and dependent on the face geometry and can be solved as a mass transport problem. However, this technique does not model self-occlusion required for handling cast shadows and can easily suffer from occlusion by hair or accessories.

**GAN-based methods.** A few techniques use GANs [16] to solve relighting [63, 32]. Tewari et al. [63] rely on StyleRig [64], a technique to enable semantic control of StyleGAN [25] by mapping a set of morphable model parameters along with an initial StyleGAN latent code to a new one representing the target parameters. Specifically, they extend StyleRig, which only works on synthetic images, to real images by optimizing a latent code that reproduces the input image and use StyleRig to manipulate the lighting condition. Similarly, Mallikarjun et al. [32] maps a target illumination and a StyleGAN latent code predicted from pSp network [45] to a new code that represents a relit image. However, these techniques tend to change the identity and facial details of the input person due to the imperfect GAN inversion. New GAN inversion techniques [12, 46] are promising, but no relighting results with these techniques have been demonstrated. Our solution overcomes this issue by leveraging DDIM’s near-perfect inversion and produces high-fidelity results that preserve the original details.

3. Approach

Given an input face image, we seek to relight this image under a target lighting condition, described by spherical harmonic coefficients and an additional scalar representing the “degree” of visible cast shadows. To explain our method, we first cover relevant background on DDIM [59] and a key finding from DiffAE [41] that shows how a conditional DDIM can perform attribute manipulation on real images by acting as both a decoder and a “stochastic” encoder.

3.1. Background: Conditional DDIM & DiffAE

Our method relies on a conditional Denoising Diffusion Implicit Model (DDIM) [59], which is a variant of diffusion models [58, 19, 60]. (For a full review and notation convention, please refer to [59].) Unlike standard diffusion models, DDIM uses a non-Markovian inference process that relies on the conditional distribution \( q(x_{t-1} \mid x_t, x_0) \) that is conditioned on \( x_0 \) (the original image) in addition to \( x_t \). One important implication is that the generative process can be made deterministic, allowing us to deterministically map \( x_T \sim N(0, I) \) to \( x_0 \) and vice versa. Here the mapping from \( x_0 \) to \( x_T \) can be viewed as the encoding of an input image \( x_0 \) to a latent variable \( x_T \).

Diffusion Autoencoders (DiffAE) [41] show that such image encoding yields \( x_T \) that contains little semantic information about the input image \( x_0 \) and propose to condition the DDIM also on a learnable latent variable \( z \) predicted from a separate image encoder. By jointly training the image encoder and the DDIM, the encoded \( z \) now captures meaningful semantics, while the encoded \( x_T \), inferred by reversing the deterministic generative process of the DDIM, captures the rest of the information not encoded in \( z \), such as stochastic variations. The resulting latent code \((z, x_T)\) can also be decoded back to the input image nearly perfectly using the same conditional DDIM. By modifying the semantic latent variable \( z \) and decoding the new \((z', x_T)\), DiffAE can manipulate semantic attributes of a real input image—a capability that inspires our work.

3.2. Method overview

The general idea of our method is to encode the input image into a feature vector that disentangles the light information from other information about the input image. Then, the relit image is produced by modifying the light encoding in the feature vector and decoding the resulting vector with a conditional DDIM (see Figure 2). This process is similar to how DiffAE performs attribute manipulation; however, our task requires well-disentangled and interpretable light encoding that facilitates lighting manipulation by the user.

To solve this, we use off-the-shelf estimators to encode an input image into light, shape, and camera encodings, as well as a face embedding, a shadow scalar, and a background image (Section 3.3). Then, these encodings are used to condition our DDIM decoder (Section 3.4) with a novel conditioning technique (Section 3.5). For training, we use a standard diffusion objective to reconstruct training images (Section 3.6). To relight, we reverse the generative process of the DDIM conditioned on the input’s encodings to obtain \( x_T \), modify the light encoding, and decode \( x_T \) using the modified encodings (Section 3.7).

3.3. Encoding

The goal of this step is to encode the input face image \( I \in \mathbb{R}^{H \times W \times 3} \) into a feature vector:

\[
f = (l, s, \text{cam}, \xi, c, \text{bg}),
\]

where \( l \in \mathbb{R}^{9 \times 3} \) represents 2nd-order spherical harmonic lighting coefficients, \( s \in \mathbb{R}^{[s]} \) represents parameterized face shape, \( \text{cam} \in \mathbb{R}^{1+2} \) represents orthographic camera parameters, \( \xi \in \mathbb{R}^{512} \) is a deep feature embedding based on ArcFace \([8]\), \( c \) is a scalar that indicates the degree of visible cast shadows, and \( \text{bg} \in \mathbb{R}^{H \times W \times 3} \) contains the background pixels with the face, hair, neck masked out. These variables will be inferred using off-the-shelf or pretrained estimators.

**Light, shape, & camera encodings** \((l, s, \text{cam})\). We use an off-the-shelf single-view 3D face reconstruction method, DECA [13]. Given a face image, DECA predicts the 3D face shape, camera pose, albedo map, and spherical harmonic lighting (SH) coefficients.
For our light encoding $l$, we directly use the SH coefficients from DECA, consisting of 9 coefficients for each channel of the RGB. DECA’s 3D face shape is parameterized based on FLAME model [27] as blendshapes with three linear bases for identity shape, pose, and expression. Their respective coefficients are denoted by $\beta, \theta, \psi$. Our face shape encoding $s$ is the combined $(\beta, \theta, \psi) \in \mathbb{R}^{|\beta|+|\theta|+|\psi|}$. DECA assumes orthographic projection and models the camera pose with isotropic scaling and 2D translation. We combine the scaling and translation parameters into $\text{cam} \in \mathbb{R}^{1+2}$. Note that we do not use the predicted albedo map because its estimation by DECA can be unreliable and we found it empirically unnecessary.

**Identity encoding ($\xi$).** To compute our deep feature embedding that helps preserve the input’s identity, we use ArcFace [8], a pre-trained face recognition model based on ResNet [18]. This model has been shown to produce discriminative and identity-preserving feature embeddings.

**Cast shadow encoding ($c$).** This scalar describes the degree of visible cast shadows, typically caused by a dominant point or directional light source, such as the sun.

We trained a model to estimate $c$ from a face image ourselves and fixed this pretrained estimator. To do this, we manually labeled around 1,000 face images with binary flags indicating whether cast shadows are visible. Following a technique proposed in DiffAE [41], we first use DiffAE’s pretrained encoder to map each face image to a semantically meaningful latent code $z$ and train a logistic regression classifier on $z$ to predict the flag. $c$ is then computed as the logit value of the logistic regression. As shown in [41], this technique helps reduce the number of training examples required to achieve good accuracy, but we note that $c$ can be estimated in other ways, such as with a CNN.

**Background encoding ($\text{bg}$).** To help fix the background during relighting, we condition the DDIM with an image of the input’s background. The background region is detected using a face segmentation algorithm [73]. The ears, hair, and neck are not part of the background and can be relit by our algorithm (see Figure 7.)

### 3.4. DDIM decoder & Modulator network

Our main network is a conditional DDIM that decodes our feature vector (with modified lighting information) to a relit version of the input image. In practice, the feature vector is used to condition the DDIM that maps $x_T \sim \mathcal{N}(0, I)$ to the original input $x_0$ during training or maps $x_T = \text{DDIM}^{-1}(x_0)$ from reversing the generative process to the relit output during relighting (Section 3.7). This conditioning involves another network called Modulator network, which converts the light, shape, and camera encodings into spatial modulation weights for the DDIM decoder.

The architecture of the DDIM decoder is based on Dhariwal et al. [10], which is a modified UNet built from a stack of residual blocks interleaved with self-attention layers. We provide full details in Appendix C. Our only differences are that 1) the output of each residual block in the first half of the U-Net will be modulated by the signal from the Modulator network and 2) we use our own version of adaptive group normalization. Our Modulator network has the same architecture as the first half of our DDIM’s U-Net, but they do not share weights.

### 3.5. Conditioning DDIM decoder

Conditioning a diffusion model on a condition vector can be done in various ways, such as through adaptive group normalization [71, 41, 10] or attention-based mechanisms [36, 47], among others. In our problem, the lighting information is encoded explicitly as SH coefficients and their interaction with 3D shape, specifically the surface normals, can be precisely modeled with the SH lighting equation. Our idea is to ease the modeling of the known interaction by rendering a shading reference of the target relit face. The primary goal of this reference is to convey the information about the target lighting and shading in a spatially-aligned manner, not the geometry or the exact shading intensities. The following sections detail this “spatial” conditioning technique as well as a standard non-spatial conditioning technique used for other encodings.

**Spatial conditioning.** This technique is used for the light, shape, camera and background encodings ($l, s, \text{cam, bg}$). Given the face shape $s$, we first convert it to a triangle mesh using the three linear bases of FLAME model [27] and remove the ears, eyeballs, neck, and scalp from the mesh to retain only the face region (See Figure 2). We remove those parts because they are often inaccurate and hard to estimate correctly (e.g., occluded ears behind hair). We assume a constant gray albedo (0.7, 0.7, 0.7) and render this mesh in the camera pose described by $\text{cam}$ with surface colors computed with 1 using the standard SH lighting equation. The details are in Appendix F, and we discuss this albedo choice and the inherent albedo-light ambiguity in Section 5.

Then, this shading reference $R$, which shows a shaded face in the shape and pose of the input person under the target lighting, is concatenated with the background image $\text{bg}$ and fed to the Modulator network. Let us denote the output of each residual block $i$ in the Modulator network by $m_i \in \mathbb{R}^{H_i \times W_i \times D_{m}}$, and the output of the corresponding residual block in the identical DDIM’s first half by $o_i \in \mathbb{R}^{H_i \times W_i \times D_{o}}$. In the DDIM, we take each residual block’s output $o_i$ and replace it with $o_i' \odot \tanh(m_i)$, which will be used as input to the subsequent layer in the network:

$$o_i' = o_i \odot \tanh(m_i),$$

where $\odot$ is the element-wise multiplication. This conditioning technique allows the shaded image $R$ and the background to retain their spatial structure and facilitate local...
conditioning of the generation as they are spatially aligned with the input (e.g., their facial parts and background are in the same positions).

**Non-spatial conditioning.** This technique is used for \((s, \text{cam}, \xi, c)\). The direct use of \(s, \text{cam}\) again in this technique is empirically found to be helpful, in addition to their indirect use through the shading reference. We use a similar conditioning technique as used in [10, 41] based on adaptive group normalization (AdaGN) [71] for these encodings and also for the time embedding in the standard diffusion model training \(\gamma(t)\), where \(\gamma\) is a sinusoidal encoding function [10]. Given an input feature map \(h_j \in \mathbb{R}^{H_j \times W_j \times D_j}\), we compute

\[
Ada\text{GN}_j(h_j, s, \text{cam}, \xi, c, t) = k_j(t_j GN(h_j) + t_j^0),
\]

where \(k_j = \text{MLP}_j^\alpha(\text{Concat}(s, \text{cam}, \xi, c)) \in \mathbb{R}^{D_j}\), is the output of a 3-layer MLP with the SiLU activation [11], and \((t_j^0, t_j) \in \mathbb{R}^{2 \times D_j} = \text{MLP}_j^\gamma(\gamma(t))\) is the output from a single-layer MLP also with the SiLU activation. GN is the standard group normalization. We apply our AdaGN in place of all the AdaGNS in the original architecture of [10], which occur throughout the UNet. (Details in Appendix C.)

3.6. Training

We jointly train the DDIM decoder, parameterized as a noise prediction network \(\theta_\phi\), and the Modulator network \(M_\psi(I, s, \text{cam}, bg)\) using standard diffusion training [19, 59, 41]. Here we consider the MLPs in Figure 2 as part of the DDIM. We adopt the simplified, re-weighted version of the variational lower bound with \(\epsilon\) parameterization:

\[
L_{\text{simple}} = \mathbb{E}_{t, x_0, \epsilon} [\epsilon_\theta(x_t, t, M_{\psi}, s, \text{cam}, \xi, c) - \epsilon^2],
\]

where \(\epsilon_\theta\) is trained to predict the added noise \(\epsilon \sim \mathcal{N}(0, I)\) in \(x_t = \sqrt{\alpha_t} x_0 + \sqrt{1 - \alpha_t} \epsilon\), given a training image \(x_0\). We define \(\alpha_t\) as \(\prod_{s=1}^t (1 - \beta_s)\), where \(\beta_t\) is the noise level at timestep \(t\) in the Gaussian diffusion process \(q(x_t|x_{t-1}) = \mathcal{N}(\sqrt{1 - \beta_t} x_{t-1}, \beta_t I)\). We use a linear noise schedule and a total step \(T = 1000\). Note that we do not reverse \(x_T = \text{DDIM}^{-1}(x_0)\) during training.

3.7. Relighting

To relight an input image, we first encode the input image into our feature vector \(f\) (Equation 1), then reverse the deterministic generative process of our DDIM conditioned on \(f\), starting from the input image \(x_0\) to \(x_{T=1000}\):

\[
x_{t+1} = \sqrt{\alpha_{t+1}} g_\theta(x_t, t, f) + \sqrt{1 - \alpha_{t+1}} \epsilon_\theta(x_t, t, f),
\]

where \(g_\theta\) represents the predicted \(x_0\), which is reparameterized from \(\epsilon_\theta\) and is computed by:

\[
g_\theta(x_t, t, f) = \frac{1}{\sqrt{\alpha_t}} (x_t - \sqrt{1 - \alpha_t} \epsilon_\theta(x_t, t, f)).
\]

After obtaining \(x_T\), we modify the SH light encoding \(I\) and the cast shadow flag \(c\) to the target \(I'\) and \(c'\), which can be set manually or inferred from a reference lighting image using DECA and our cast shadow estimator. Then, we decode the modified \(f' = (I', \text{cam}, \xi, c', \text{bg})\) using the reverse of Equation 4, starting from \(x_T\) to produce the final output.

The reverse process to obtain \(x_T\) is key to reproducing high-frequency details from the input image. As demonstrated in DiffAE [41], DDIM will encode any information not captured in the conditioning feature vector \(f\) in the noise map \(x_T\). This information includes high-frequency details, such as the hair pattern or skin texture.

**Improved DDIM sampling with mean-matching.** We observe that when the input image contains a background with extreme intensities (e.g., too dark or too bright), DDIM can produce results with a slight change in the overall brightness. We alleviate this issue by computing the mean pixel difference between each \(x_t\) during DDIM’s generative reversal \((x_0 \rightarrow x_T)\) and \(x_T\) from self-decoding of the reversed noise \(x_T\). This sequence of mean differences is then applied to the decoding for relighting (Appendix B).

4. Experiments

In this section, we present quantitative and qualitative results of our proposed method. We provide a comparison of our relighting performance (Section 4.1) to the state of the art on Multi-PIE dataset and ablation studies (Section 4.3) on the non-spatial and light conditioning. Implementation details, dataset details, and runtime are in Appendix B.

**Evaluation metrics.** We use DSSIM [35], LPIPS [75], and MSE. DSSIM measures the structure dissimilarity, and LPIPS measures the perceptual quality. All metrics are computed between each relit image and its ground-truth image only on the face region following [22, 23] using the same face parsing algorithm [73].

4.1. Relighting performance

We evaluate our relighting performance on Multi-PIE dataset [17] against recent state-of-the-art methods [23, 22, 35, 38]. Note that Pandey et al. [38] solve a different problem setup (also [69, 72]) and require an HDR environment map as input, which has to be first estimated from a target image, making a comparison with [38] not entirely apples-to-apples. The results of [38] in our experiment were generated by the authors themselves, including the HDR maps. Other test sets and code of [38] were not released.

Our experiment has two setups where the target lighting is taken either from \(i\), the same person. This setup uses the same test set as [23], which contains 826 testing samples from 329 subjects. Or \(ii\), a different person. This setup contains 200 random triplets of input, target, and ground-truth images, where the target image is of a different person.
Figure 3: Reilt images on FFHQ [25]. The FFHQ dataset contains a variety of face images captured in real-world environments. Our method produces more realistic relit images than previous methods and can add or remove cast shadows and highlights to match the reference lighting.
The results are shown in Table 1, Figure 4, and Figure 5. For both setups, our method achieves the best performance across all metrics with minimal artifacts and can convincingly relight the neck and ears or remove cast shadows, e.g., from the nose of the lady in Figure 5. We include a comparison with [53, 77, 61] and more qualitative results of [23, 22, 38] in Appendix E.

Table 1: State-of-the-art comparison on Multi-PIE.

<table>
<thead>
<tr>
<th>Method</th>
<th>DDSIM↓</th>
<th>MSE↓</th>
<th>LPIPS↓</th>
</tr>
</thead>
<tbody>
<tr>
<td>Same subject as target lighting</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Nestmayer et al. [35]</td>
<td>0.2226</td>
<td>0.0588</td>
<td>0.3795</td>
</tr>
<tr>
<td>Pandey et al. [38]</td>
<td>0.0875</td>
<td>0.0165</td>
<td>0.2010</td>
</tr>
<tr>
<td>Hou et al. (CVPR’21) [23]</td>
<td>0.1186</td>
<td>0.0303</td>
<td>0.2013</td>
</tr>
<tr>
<td>Hou et al. (CVPR’22) [22]</td>
<td>0.0990</td>
<td>0.0150</td>
<td>0.1622</td>
</tr>
<tr>
<td>Ours</td>
<td><strong>0.0711</strong></td>
<td><strong>0.0122</strong></td>
<td><strong>0.1370</strong></td>
</tr>
<tr>
<td>Different subject as target lighting</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pandey et al. [38]</td>
<td>0.1000</td>
<td>0.0252</td>
<td>0.2053</td>
</tr>
<tr>
<td>Hou et al. (CVPR’21) [23]</td>
<td>0.1056</td>
<td>0.0247</td>
<td>0.1989</td>
</tr>
<tr>
<td>Hou et al. (CVPR’22) [22]</td>
<td>0.1150</td>
<td>0.0238</td>
<td>0.2215</td>
</tr>
<tr>
<td>Ours</td>
<td><strong>0.0969</strong></td>
<td><strong>0.0215</strong></td>
<td><strong>0.1669</strong></td>
</tr>
</tbody>
</table>

4.2. Qualitative evaluations

Relighting on FFHQ dataset. In Figure 3, we provide a qualitative comparison with three recent SOTAs [38, 23, 22] on subjects with different head poses, genders, races, accessories. Our approach produces highly realistic results and can synthesize new highlights and eliminate hard shadows, while the competing methods often leave behind shadow or shading residuals due to the inaccurate albedo prediction of these in-the-wild images.

Shadow flag condition. We show a novel ability to change the strength of cast shadows on FFHQ [25] in Figure 6. We generate these results by varying the cast shadow’s logit value (c). Our method can realistically remove shadows (e.g., those cast by eyeglasses or face geometry) or intensify their effects. Figure 1 demonstrates how the direction and appearance of cast shadows can change according to a new target lighting condition (the bottom guy’s chin).

4.3. Ablation studies

Light conditioning. We compare our full pipeline against two alternatives for conditioning the DDIM on the
We show the ability to change the degree of cast shadows by adjusting the scalar $c$ and decode the modified feature vector.

Without bg conditioning, the background and hat of the input person are not well preserved.

light encoding: a) We do not use our Modulator network and instead feed the reference shading directly to the DDIM by concatenating it with each $x_t$ in every timestep. b) We do not use the shading reference and instead concatenate the light encoding $l$ with $(s, \text{cam}, \xi, c)$ in the non-spatial conditioning technique.

We report the results in Table 2 and show a qualitative comparison in Appendix E. Removing all of $s$, $\text{cam}$, $\xi$ performs the worst, whereas removing $s$, $\text{cam}$ but retaining $\xi$ obtains a better MSE score. On the other hand, our full pipeline outperforms these alternatives on both DDSIM and LPIPS metrics, which agree with human perception.

<table>
<thead>
<tr>
<th>Method</th>
<th>DDSIM</th>
<th>MSE</th>
<th>LPIPS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Light conditioning</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>a) No Modulator</td>
<td>0.0749</td>
<td>0.0081</td>
<td>0.0868</td>
</tr>
<tr>
<td>b) Used as non-spatial</td>
<td>0.0885</td>
<td>0.0098</td>
<td>0.0947</td>
</tr>
<tr>
<td>Ours</td>
<td>0.0670</td>
<td>0.0077</td>
<td>0.0789</td>
</tr>
<tr>
<td><strong>Non-spatial condition vector</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>c) No $s$, $\text{cam}$, $\xi$</td>
<td>0.0713</td>
<td>0.0082</td>
<td>0.0909</td>
</tr>
<tr>
<td>d) No $s$, $\text{cam}$</td>
<td>0.0674</td>
<td>0.0063</td>
<td>0.0846</td>
</tr>
<tr>
<td>e) No $\xi$</td>
<td>0.0686</td>
<td>0.0074</td>
<td>0.0847</td>
</tr>
<tr>
<td>Ours</td>
<td>0.0670</td>
<td>0.0077</td>
<td>0.0789</td>
</tr>
</tbody>
</table>

5. Limitations & Discussion

While our results look photorealistic and plausible, some aspects such as cast shadows may not be physically accurate. There is room for improvement in terms of temporal consistency when the result is rendered as a video with moving lights (best observed on our website). Other limitations include: shadows cast by external objects can be incorrectly relit, and sunglasses, which resemble cast shadows, can sometimes be removed (Figure 8). Relighting to match a reference lighting image can be inaccurate as we rely on a light estimator, which is susceptible to the ambiguity where it is unclear if, e.g., a dark appearance is caused by the skin tone or dim lighting. Incorporating physical rendering to help guide the appearance of cast shadows or allowing other lighting models are interesting future work. Our diffusion-based model requires multiple network passes and is currently slower than other GAN-based methods.

In conclusion, we have presented a diffusion-based face relighting method that eliminates the need for accurate intrinsic decomposition and can be trained on 2D images without any 3D or lighting ground truth. Our key component is a conditional diffusion implicit model and a novel conditioning technique that maps a disentangled light representation to a relit image. This enables our method to achieve new state-of-the-art performance and produce highly photorealistic results for real-world scenarios.
References


[34] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiachen Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. S4edit: Guided image synthesis and editing with stochastic differential equations. In International Conference on Learning Representations, 2021. 14


