VoRF: Volumetric Relightable Faces – SUPPLEMENTAL MATERIAL –

Pramod Rao[1]
prao@mpi-inf.mpg.de
Mallikarjun B R[1]
mbr@mpi-inf.mpg.de
Gereon Fox[1]
gfox@mpi-inf.mpg.de
Tim Weyrich[2]
tim.weyrich@fau.de
Bernd Bickel[3]
bernd.bickel@ist.ac.at
Hanspeter Pfister[4]
pfister@g.harvard.edu
Wojciech Matusik[5]
wojciech@csail.mit.edu
Ayush Tewari[5]
ayusht@mit.edu
Christian Theobalt[1]
theobalt@mpi-inf.mpg.de
Mohamed Elgharib[1]
elgharib@mpi-inf.mpg.de

[1] Max Planck Institute for Informatics, Saarland Informatics Campus, Germany
[2] Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), Germany
[3] IST-Austria, Austria
[4] Harvard University, USA
[5] MIT CSAIL, USA

In this supplemental document, we provide a detailed derivation of the OLAT decomposition and the face reflectance fields in Sec. 1. We then provide a summary of related works in Sec. 2, to better highlight our work with respect to the literature. In Sec. 3 we provide the detailed network architectures of our *Face Prior Network* and our *Reflectance Network*. Further, we discuss the lightstage dataset used to train our method in Sec. 4. Implementation details are given in Sec. 5, where we also show additional results on relighting and viewpoint synthesis of our method, with different numbers of input views. We also discuss the failures of baseline methods (PhotoApp in Sec. 6, NeLF in Sec. 7). In Sec. 8 we showcase our intermediate OLAT results. Furthermore, in Sec. 9 we provide a detailed ablative study on various design choices of our proposed solution. In Sec. 10 we analyze the limitations of our proposed method. In addition to this supplemental document, we have included multiple videos summarizing all our results.

1 Detailed derivation of OLAT decomposition

Obtaining complex lighting conditions by linearly combining OLAT images according to environment maps is a principle that is well-studied in the literature [2]. We find it worthwhile to show that this principle is actually compatible with NeRF's volumetric rendering model [8]. We already gave a short description of this compatibility in the main paper, but we repeat the argument here in a more detailed form.

Debevec *et al.* [2] argue that under the assumption that all sources of incident light are sufficiently far away from the face we can describe lighting conditions by a function $L_{\text{inc}}(\omega)$. This function only depends on the direction $\omega \in S$ from which radiance is incident and indicates the amount of that radiance, where S is the set of all such directions. We introduce a combination of a *volume density function* [8] and a *reflectance field* [2], that we call *volumetric reflectance field*: A volumetric reflectance field is a pair (σ, R), where the *volume density function* $\sigma : \mathbb{R}^3 \to \mathbb{R}$ maps scene points to density values and the function $R(\omega, \mathbf{x}, \mathbf{d})$ indicates the fraction of $L_{\text{inc}}(\omega)$ that is reflected from point $\mathbf{x}$ in the direction $\mathbf{d}$.

The additiveness of light transport allows us to describe the total amount $L_{\text{out}}(\mathbf{x}, \mathbf{d})$ of radiance reflected out of point $\mathbf{x}$ in the direction $\mathbf{d}$ as

$$L_{\text{out}}(\mathbf{x}, \mathbf{d}) := \int_{\omega \in S} R(\omega, \mathbf{x}, \mathbf{d}) \cdot L_{\text{inc}}(\omega)\, d\omega \tag{1}$$

We assume that image formation follows the volumetric principles described by Mildenhall *et al.* [8], i.e. we assume a ray $r_{\mathbf{o},\mathbf{d}}(t) = \mathbf{o} + t\mathbf{d}$ being shot through a camera pixel into the scene, and describe the amount of radiance accumulated along this ray as

$$L(r) := \int_{t_n}^{t_f} T(t) \cdot \sigma(r(t)) \cdot L_{\text{out}}(r(t), d)\, dt \text{ with } T(t) := \exp\left(- \int_{t_n}^{t} \sigma(r(s)) ds \right) \tag{2}$$

where t_n, t_f are bounds within which the entire face is contained.

In order to bridge the gap between the OLAT conditions of the dataset and real world lighting conditions, we discretize the dense set of incident light directions S to a finite set I, with one direction $i \in I$ per OLAT light source where $S_i \subseteq S$ represents a subset. We now approximate the following:

$$L_{\text{out}}(\mathbf{x}, \mathbf{d}) \approx \sum_{i \in I} R(\omega_i, \mathbf{x}, \mathbf{d}) \cdot L_{\text{inc}}(i) \tag{3}$$

where ω_i is the incident light direction of OLAT light source i and $L_{\text{inc}}(i) := \int_{\omega \in S_i} L_{\text{inc}}(\omega)$ is the discretized version of L_{inc}.

The property of OLATs that allow to compose complex lighting conditions can now be derived as follows:

Under OLAT conditions there exists a single $i \in I$ that contributes some radiance $\mathbb{L}_i := L_{\text{inc}}(i)$ (i.e. only lamp i is turned on), while for all $j \neq i$ we have $L_{\text{inc}}(j) = 0$. Thus, for a given ray r with origin $\mathbf{o}$ and direction $\mathbf{d}$, the accumulated radiance $L(r)$ is approximated by

$$L(i, r) := \int_{t_n}^{t_f} T(t) \cdot \sigma(r(t)) \cdot R(\omega_i, r(t), \mathbf{d}) \cdot \mathbb{L}_i\, dt \tag{4}$$

Under non-OLAT conditions all we know is that $\forall i \in I$ there must exist some factor f_i, s.t. $L_{\text{inc}}(i) = f_i \cdot \mathbb{L}_i$. This allows us to equate

$$\begin{aligned} L(r) &\approx \int_{t_\text{n}}^{t_\text{f}} T(t) \cdot \sigma(r(t)) \cdot \sum_{i \in I} R(\omega_i, r(t), \mathbf{d}) \cdot f_i \cdot \mathbb{L}_i \, dt \\ &= \sum_{i \in I} f_i \cdot \int_{t_\text{n}}^{t_\text{f}} T(t) \cdot \sigma(r(t)) \cdot R(\omega_i, r(t), \mathbf{d}) \cdot \mathbb{L}_i \, dt = \sum_{i \in I} f_i \cdot L(i, r) \end{aligned} \tag{5}$$

Eq. 5 shows that under the stated assumptions we can render the face under any given lighting specification $(f_i)_{i \in I}$ just as a linear combination of OLAT images. The errors caused by the approximations ($\approx$) in the derivations above reduce as we increase the number of OLAT directions that are used to discretize S.

2 Related Works

In Tab. 1, we compare several NeRF based relighting methods to VoRF. As shown in the table, our method can work with as low as single image at test time.While many methods require illumination of given test scene as input to perform relighting, our method can work without scene illumination as input. NeRFW models effects of illumination variation using an appearance latent code, which doesn't have any physical meaning. In contrast, our method can relight face with physically meaningful environment maps (termed as "semantic illumination" in table Tab. 1).

	NeRV	NeRF-actor	NeRD	NeRFW	NeRF-OSR	NeLF	Ours
Input views required	> 1	> 1	> 1	> 1	> 1	> 1	1
Works with unknown original illumination	No	Yes	Yes	Yes	Yes	Yes	Yes
Semantic Illu.	Yes	Yes	Yes	No	Yes	Yes	Yes
Works on unseen test scenes	No	No	No	No	No	Yes	Yes

Table 1: NeRF-based relighting methods come with different feature sets: They require different numbers of input views, might require original illumination to be known, or only work on training scenes.

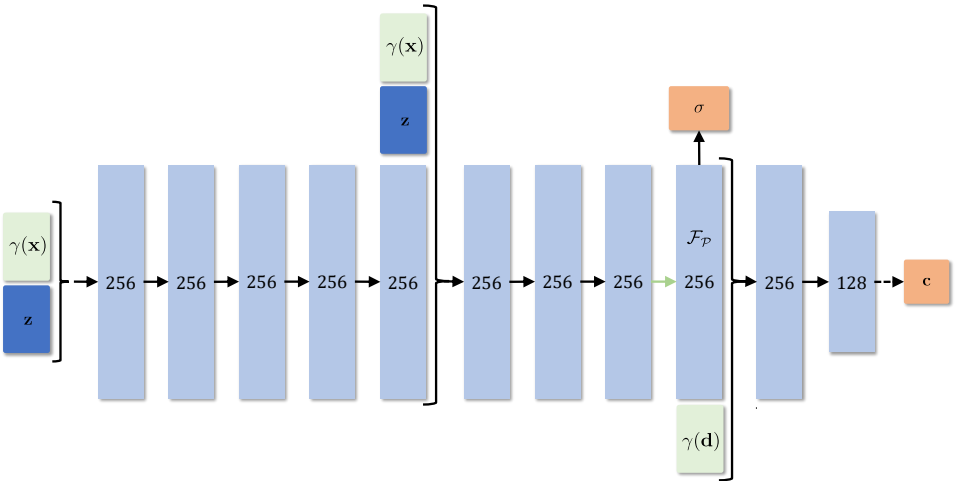


Figure 1: The architecture of our *Face Prior Network* $\mathcal{P}$ is based on that of NeRF [8]: Input vectors are green, latent vectors are dark blue and outputs are red. γ denotes positional encoding [8]. Each light blue block represents the result of a fully connected layer, with the labelling number being the number of output values. Braces represent concatenation, black arrows represent ReLU activation functions. The light green arrow is the identity function. The dashed black arrow is a sigmoid activation function. The vector $\mathcal{F}_\mathcal{P}$ is used as an input for the *Reflectance Network* ($\mathcal{R}$) (see Fig. 2).

3 Architecture Details

In this section we describe the architecture of both the *Face Prior Network*($\mathcal{P}$) and *Reflectance Network*($\mathcal{R}$). As briefly discussed in the main paper, $\mathcal{P}$ is a NeRF-based architecture and in Fig. 1 we explain the full model in detail. Furthermore, $\mathcal{R}$ is also a NeRF-based design that takes light direction and face prior features from the 9^{th} layer of $\mathcal{P}$. In addition, $\mathcal{R}$ uses density values (σ) predicted by $\mathcal{P}$ for volume rendering. We show $\mathcal{R}$ in Fig. 2.

3.1 Algorithm Design

To clarify our training, fitting, and finetuning processes we provide Figs. 3 to 5 here. The final relighting is done as presented in Fig. 3 in the main paper.

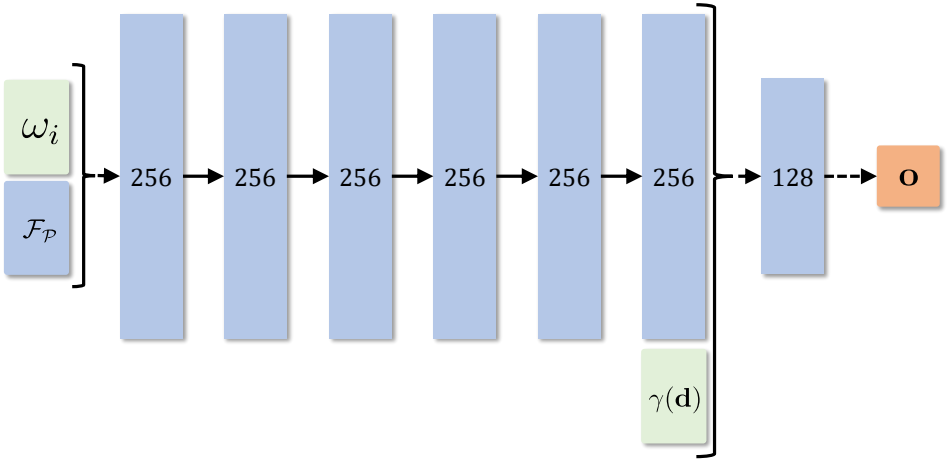
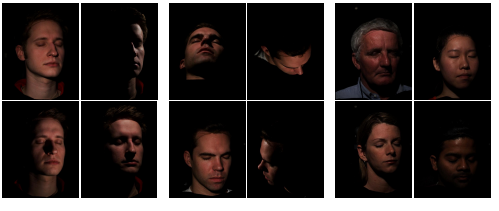


Figure 2: The architecture of $\mathcal{R}$ is also based on NeRF [8]. $\mathcal{R}$ takes a feature vector $\mathcal{F}_\mathcal{P}$ from the 9^{th} layer of $\mathcal{P}$ and incoming OLAT light direction ω_i. In addition, to synthesize an OLAT image through volume rendering [8], the volume density σ from $\mathcal{P}$ is used (see Fig. 1). For the meaning of graphical elements see the caption of Fig. 1.

4 Lightstage Dataset

We utilize a lightstage dataset [15] of 353 identities, illuminated by 150 point light sources and captured by 16 cameras. The light sources are distributed uniformly on a sphere centered around the face of the subject. For every subject each camera captures 150 images (1 per light source). All the images are captured with the subject showing a neutral expression with their eyes closed. While capturing each frame, the light sources were turned on one at a time, thus generating one-light-at-a-time (OLAT) images. Fig. 6 gives an impression of the dataset.



(a) (b) (c)

Figure 6: We use a light stage dataset [15] that provides 150 different lighting conditions (a), 16 camera angles (b) and 353 subjects (c). We brightened the images here, for better visualization.

4.1 Lightstage Test Dataset

For experiments that require a ground-truth reference, we created such a reference by combining lightstage images according to different environment maps: We randomly sampled 10 unseen identities from the lightstage dataset and synthesized naturally lit images using 10 randomly chosen unseen HDR environment maps, from the Laval Outdoor dataset [4] and the Laval Indoor HDR dataset [3]. For all quantitative and qualitative experiments, we evaluate only on the held-out views. For instance, given that the lightstage dataset has a total of 16 camera viewpoints, an evaluation method that takes three input views would be evaluated on the remaining 13 held-out views.

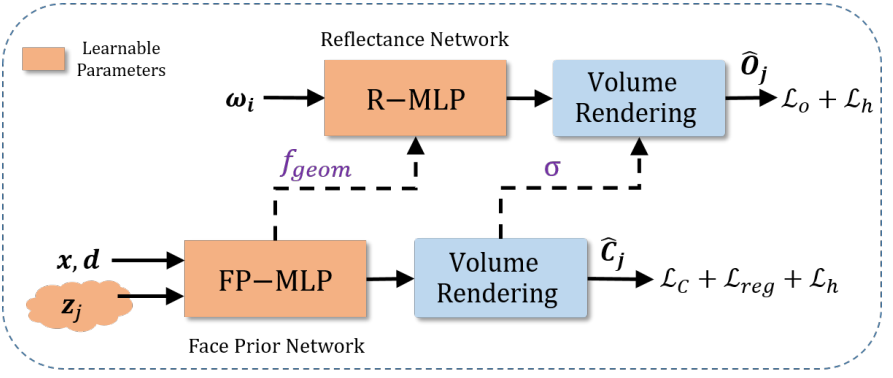


Figure 3: **Training:** First, we train the *Face Prior Network* along with the $\mathbf{z}_j$. Next, we jointly train all the components in orange from the figure in an end-to-end manner. This includes training the *Reflectance Network*.

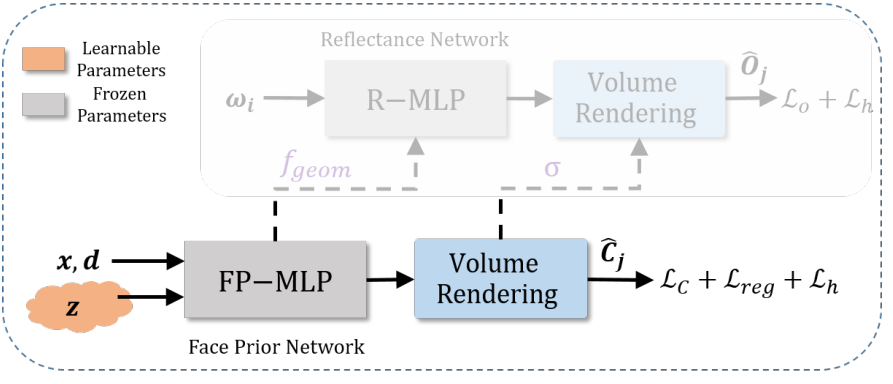


Figure 4: **Fitting:** Given an unseen test subject, we optimize $\mathbf{z}$ with frozen *Face Prior Network* parameters. The *Reflectance Network* is not used during both *fitting* and *finetuning* (Fig. 5) stages.

5 Results

In this section, we discuss the implementation details of our method and baselines. Furthermore we show more results on the H3DS [9] test dataset, for the task of simultaneous view synthesis and relighting.

5.1 Implementation Details

In order to capture the geometry of multiple heads along with various natural illuminations, we train the Face Prior Network $\mathcal{P}$ with 302 identities synthesized with 600 different natural illuminations (see main paper, Sec. 3, last paragraph). The Reflectance Network $\mathcal{R}$ is supervised with only OLAT images. After learning a sufficiently good face prior, we jointly train both the networks and sample OLAT images and naturally-lit images with equal probability. During training, we optimize Eq. 8 (in the main paper) with a batch size of 1 using the Adam Optimizer [8] and learning rates of 5×10^{-5} for both $\mathcal{P}$ and $\mathcal{R}$, and of 5×10^{-4} for the latent

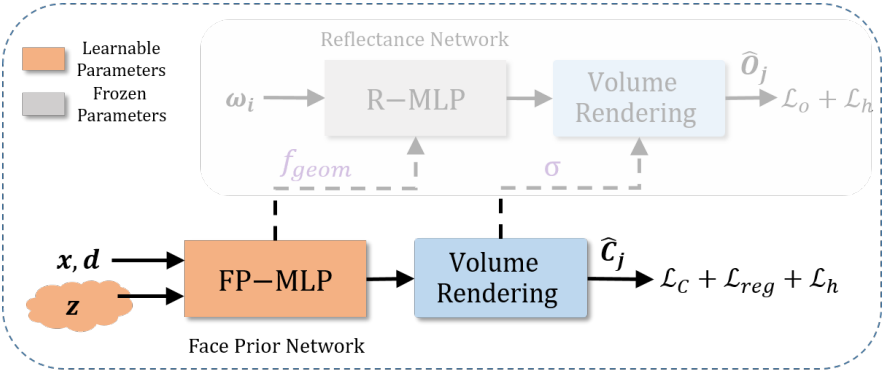


Figure 5: **Finetuning:** Upon convergence of the *fitting* process (Fig. 4), we jointly optimize **z** and *Face Prior Network* to better capture the details of input data.

codes. We train stage 1 for about 2 days with a step size of 1.6×10^6 and stage 2 is trained with a step size of 3.5×10^6 using 3 NVIDIA Quadro RTX 8000 GPUs for 5 days. During test, we perform the fitting process for $10,000$ steps with a learning rate of 1×10^{-3} for the latent code, followed by fine-tuning for 3000 steps with a learning rate of 3×10^{-6} for $\mathcal{P}$ using a single NVIDIA Quadro RTX 8000 GPU.

5.2 Baseline Implementation Details

For a fair evaluation, we retrain NeLF [12], IBRNet [13] and SIPR [14] on our lightstage dataset. We train both NeLF and IBRNet using 16 views and ensure at least one front facing view is included in each batch as required for both the methods. Since we train with a real lightstage dataset, we estimate mask and depth maps from the Agisoft Metashape software and use that as reference for training NeLF and IBRNet. Due to the noisy depth maps, we found that lowering the weight of their depth loss by a factor of 2 was useful for training NeLF. We also train PhotoApp with the lightstage dataset following the details from the original paper [7]. We note, however, that PhotoApp cannot be trained with all 16 views as they require poses that can be projected into StyleGAN space. Thus, we train with only 8 views that are closer to a frontal pose.

5.3 Results on Uncontrolled Data

We show more results of simultaneous relighting and view synthesis on the H3DS dataset [9]. We show results using three input views (see Fig. 7), two input views (see Fig. 8) and one input view (see Fig. 9). Our technique produces photorealistic results and maintains the input identity. The generated identities are also view-consistent, which is crucial for many applications. Refer to the following **supplemental videos** for additional results – *Novel_View_Synthesis.mp4* and *Novel_View_Synthesis_and_Relighting.mp4*.

6 PhotoApp Analysis

We compare our method against PhotoApp [7]. PhotoApp utilizes the StyleGAN latent space [5] and learns to edit the illumination and camera viewpoint in this space. The method

Figure 7: Novel view synthesis and relightings of our method on the H3DS dataset [9]. Here, we use 3 views as input. Target environment maps are shown in the lower left corners. Our technique produces photorealistic results and maintains the input identity even at extreme viewpoints.

Figure 8: Novel view synthesis and relightings results of our method on the H3DS dataset [9] using 2 input views. Target environment maps are shown in the lower left corners. Our technique produces photorealistic results and maintains the input identity even at extreme viewpoints.

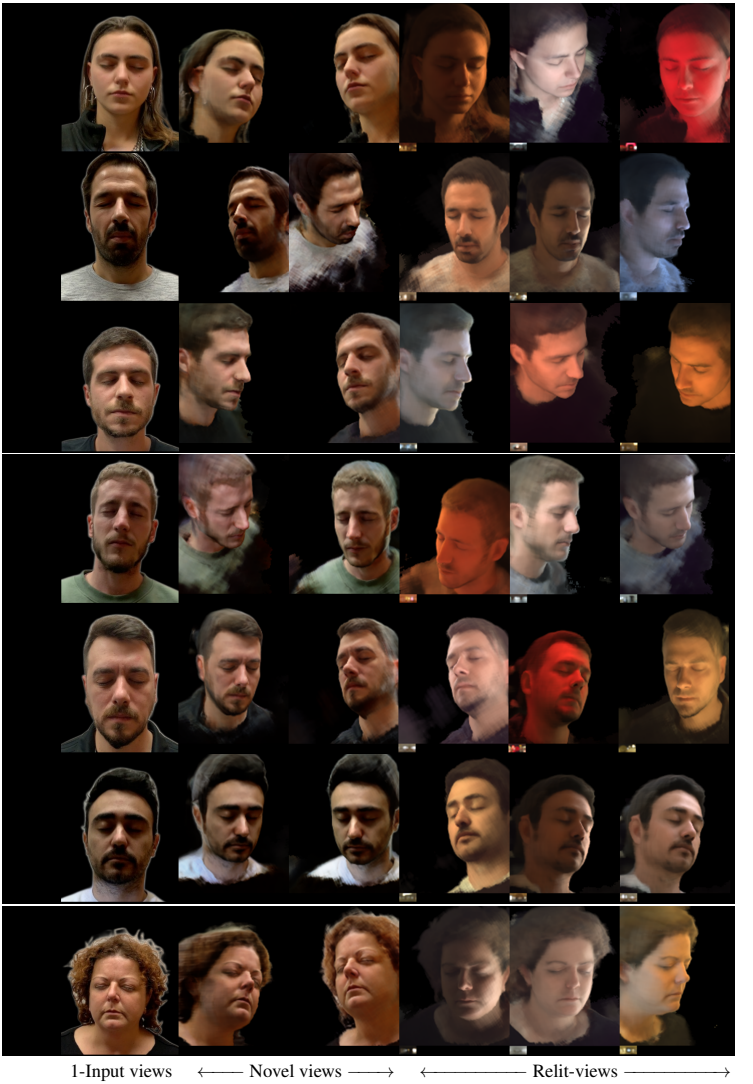


Figure 9: Novel view synthesis and relightings of our method on the H3DS dataset [9]. Here, we use a single view as input. Target environment maps are shown in the lower left corners. Our technique produces photorealistic results and maintains the input identity even at extreme viewpoints.

	View Synthesis	
Method	PSNR	SSIM
PhotoApp	**29.13**	0.72
Ours 1-view	22.49	0.77
Ours 2-views	25.44	0.79
Ours 3-views	26.67	**0.82**

Table 2: PhotoApp comparison on the lightstage test set for novel view synthesis. For PhotoApp we set the input and target illumination to be the same. While PhotoApp is limited to a single view as input, VoRF can take multiple views as input. This clearly improves our performance. Please refer to Sec. 6 for a discussion on PSNR.

is trained in a supervised manner using a lightstage dataset. While PhotoApp produces highly photorealistic results as it utilizes the StyleGAN space, it suffers from three strong limitations: First, the technique usually runs into a high risk of altering the original identity during StyleGAN fitting and editing. Second, view-point editing usually leads to strong identity inconsistencies between different views. And third, the StyleGAN embedding is limited in handling specific viewpoints and thus can produce incorrect poses during editing. These downsides severely limits PhotoApp's abilities to synthesize viewpoints and relighting. VoRF, on the other hand, maintains the input identity and produces view-consistent editing results: Refer to Fig. 10 and Fig. 11 for sample results on the lightstage test set. We also show results on the H3DS dataset (see Fig. 12). VoRF clearly maintains the input identity and produces view-consistent results (see groundtruth in case of the lightstage dataset). However, PhotoApp suffers from clear and strong artifacts as discussed.

For numerical evaluations, we report PSNR and SSIM for view synthesis only (see Tab. 2) and view synthesis plus relighting (see Tab. 3). We have noticed that PSNR is not an ideal metric for capturing the strong limitations of PhotoApp. However, these limitations are better captured with SSIM as it is a structural metric. Further, in Tab. 4 we estimated the average deviation from the groundtruth facial landmarks, which is a stronger indication of identity alteration and pose accuracy during editing. Here, landmarks are estimated using [1] and we report the mean of absolute difference of 68 face landmarks between the groundtruth and the rendered faces. Thus, the lower the Landmark Loss in Tab. 4, the better the performance. As expected, PhotoApp produces a significantly higher landmark error, almost twice as large as ours, validating our qualitative observations.

We note that all qualitative results reported here are generated using a single view as input. However, another advantage of our technique is to accept multiple viewpoints as input which is not possible with PhotoApp. This allows VoRF to produce superior results when multiple views are given as input. In Tab. 2 and Tab. 3 we show that with multiple input views(upto 3) our method gives a higher SSIM score in comparison to PhotoApp.

7 NeLF Analysis

For a fair comparison against NeLF [12], we retrain it using the lightstage dataset. The original NeLF model was trained using a synthetic dataset while VoRF is trained on a real lightstage dataset. We observe that the orignal NeLF struggles to generalize on the H3DS test dataset. We suspect that this is due to the mismatch in the data distribution. Thus for a

Figure 10: Comparing PhotoApp [7] against our method during view synthesis on the lightstage test set. Here, VoRF takes a single view as input. PhotoApp suffers from strong identity alternations and pose inaccuracies during editing. This leads to highly view-inconsistent results. Our method, however, clearly resembles the groundtruth much better than PhotoApp.

Figure 11: Comparing PhotoApp [7] against our method during simultaneous view synthesis and relighting on the lightstage test set. Here, VoRF takes a single view as input. Similar to novel view synthesis (see Fig. 10), PhotoApp suffers from strong identity alternations and pose inaccuracies during editing. This leads to highly view-inconsistent results. Furthermore, in some cases the rendered lighting is also view-inconsistent (see PhotoApp, Identity 2). Our method, however, clearly resembles the groundtruth much better than PhotoApp.

Figure 12: Comparing simultaneous view synthesis and relighting on the H3DS dataset with single view input. PhotoApp produces inconsistent identities and illuminations across different views. VoRF maintains the input identity and produces view-consistent results.

	Relighting + View Synthesis	
Method	PSNR	SSIM
PhotoApp	**29.08**	0.71
Ours 1-view	20.21	0.69
Ours 2-views	22.15	0.74
Ours 3-views	22.80	**0.76**

Table 3: PhotoApp comparison on the lightstage test set for novel view synthesis and relighting. While PhotoApp is limited to a single view as input, VoRF can take multiple views. This clearly improves our performance. Please refer to Sec. 6 for a discussion on PSNR.

Method	Landmark Loss
PhotoApp	2060.25
Ours 1-view	**1021.62**

Table 4: We report the average deviation from the ground-truth facial landmarks on the lightstage test set. Here, the smaller the Landmark Loss the more the predicted pose/identity resembles the ground-truth. We use an illumination condition which leads to good landmarks detection [1]. Our approach produces much better performance than PhotoApp.

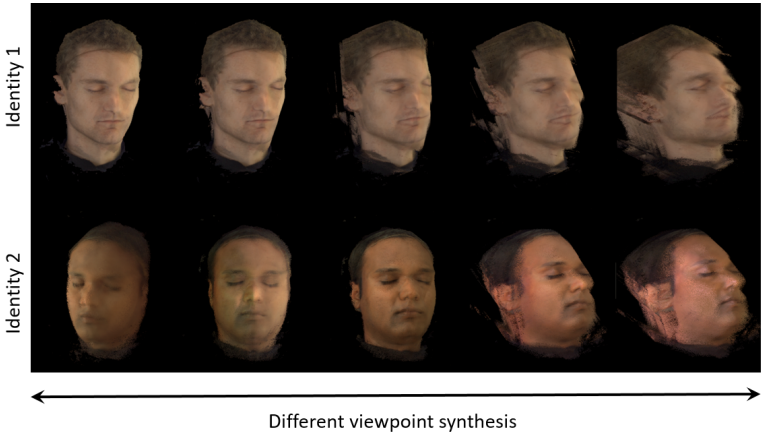


Figure 13: Viewpoint editing using our retraining of NeLF [17]. Here, we use 3 views as input. NeLF performs decently for small angles around training views. However, its performance degrades quickly as the target head pose becomes wider (see to the right).

fair evaluation of NeLF, retraining on our lightstage dataset would be necessary. Therefore, we have trained NeLF using the same data split used for training our method. We used the official implementation of NeLF from their project webpage. To further aid the training, we ensured the presence of a frontal view for each training batch.

Fig. 13 shows results from our retraining of NeLF. Results indicate that NeLF can render novel views for small angles around the training views, but struggles with wider variations around such views (see to the right). To verify the correctness of our implementation, we shared our results and sample of the training data with the authors of NeLF. The authors noted that our results are plausible since the lightstage data contains sparse viewpoints, which is challenging for NeLF to handle. They also confirmed that they observed a similar behaviour of NeLF in the early stages of their project. While VoRF relies on global cues, NeLF relies on local CNN-based features. This makes it difficult for NeLF to reason about the geometry from sparse viewpoints. We observe similar shortcomings with IBRNet as well. In Fig. 14 we provide a qualitative comparison of NeLF, IBRNet+SIPR and VoRF for the task of simultaneous view synthesis and relighting using three input views on the lightstage dataset. We observe that both NeLF and IBRBet+SIPR show significant artifacts. We see similar results when we evaluate NeLF on the H3DS dataset with three input views as shown in Fig. 15.

8 OLAT Predictions

Fig. 16 and Fig. 17 demonstrates One-Light-At-A-Time (OLAT) images produced by our method on the unseen subjects from the lightstage and H3DS datasets respectively. We show results using different number of input views and render the OLATs from different viewpoints. The predicted OLATs capture important illumination effects and details such as hard shadows and specularities. Please refer to the **supplemental video** – *View_Dependent_Effects.mp4*.

Please refer to the **supplemental video** (*150_OLAT_Rendering.mp4*) for OLATs rendered at different viewpoints for multiple test identities from the H3DS dataset with three input views.

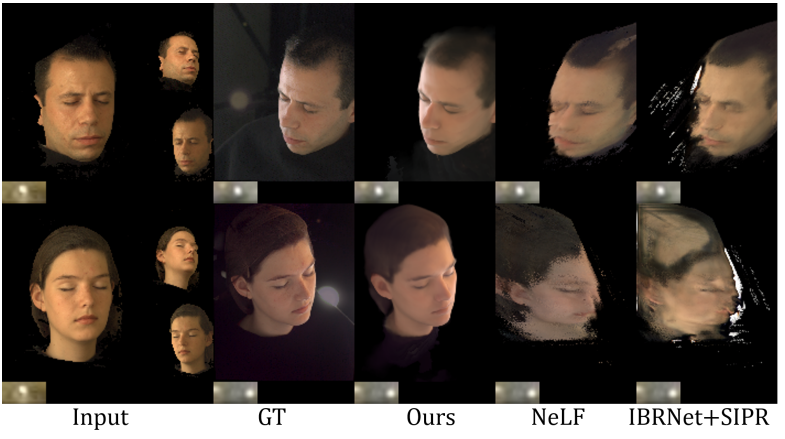


Figure 14: We show results for simultaneous view synthesis and relighting on the lightstage dataset. We can observe that the baselines show significant artifacts as we render arbitrary viewpoints.

Figure 15: Novel view synthesis and relighting on the H3DS dataset [2]. Our technique significantly outperforms NeLF especially at views far away from the training views.

Figure 16: Using the *Reflectance Network*, we can synthesize OLAT images for an unseen identity. Our method captures view-dependent effects as well as accurate shadows and the result closely matches the ground truth.

Our $\mathcal{R}$ renders an OLAT image given a light direction ω. Using the lightstage dataset, we train $\mathcal{R}$ with 150 point light sources. This enables us to generate an OLAT volume from any desired light direction, allowing us to use higher resolution environment maps and have better approximation of the environment lighting conditions. To show this, we render new a OLAT volume with 1321 point light sources by interpolating between the existing light sources. We show this dense OLAT sampling result in the **supplemental video** – *Dense_OLAT_Rendering.mp4*.

9 Ablation Study

In this section we evaluate the design choices of our work through multiple quantitative and qualitative ablation studies.

9.1 Significance of Two-Step Optimization

We examine the significance of our two-stage optimization process to reconstruct a new test identity. Our test time optimization first involves *fitting* the latent code $\mathbf{z}_{test}$ to the test subject (with the weights of both $\mathcal{P}$ and $\mathcal{R}$ frozen). This is followed by a *fine-tuning* process where we jointly update both $\mathbf{z}_{test}$ and the weights of network $\mathcal{P}$, i.e. $\Theta_{\mathcal{P}}$.

We perform the *fitting* process with a learning rate of 1×10^{-3} for 10,000 iterations to ensure that the $\mathbf{z}_{test}$ lies in the learnt face prior distribution. Then, assuming convergence, we lower the learning rate to 1×10^{-6} and jointly optimize $\mathbf{z}_{test}$ and $\Theta_{\mathcal{P}}$ for 3000 iterations. Note that we do not modify the weights of $\mathcal{R}$ in both the stages since we do not have access to OLATs for our test subjects.

We evaluate the significance of this design choice on our lightstage test dataset for the task of novel view synthesis. Quantitative results (Tab. 5) support our choice: *Fit + FineTune* performs better than just *Fit only*. In Fig. 18 (left) we observe that the fitting stage recovers an approximate face geometry, after which the fine-tuning recovers the missing identity-specific fine details.

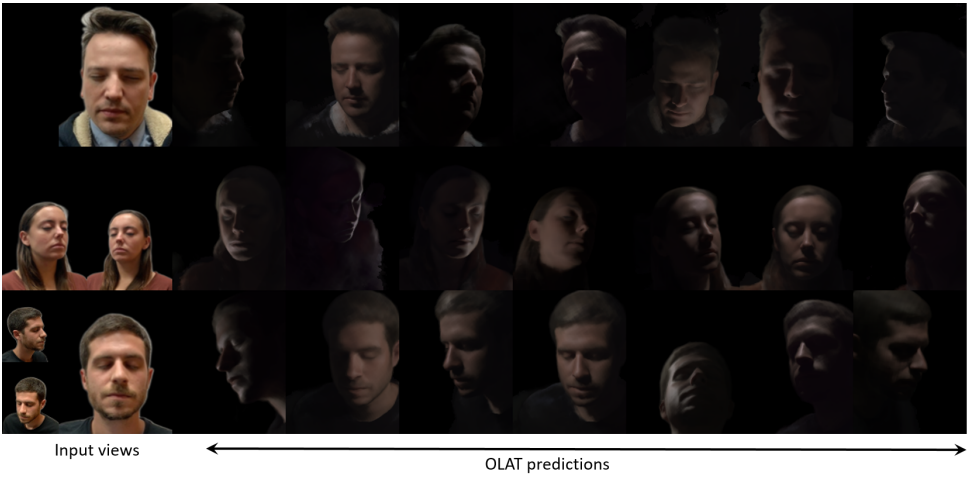


Figure 17: OLAT predictions of our method for the test subjects from the H3DS dataset. We show results with a single view as input (top), two views as input (middle) and three views as input (bottom). We render the predictions from different viewpoints. The OLAT predictions capture important illumination effects such as specularities and hard shadows.

9.2 Significance of the *Reflectance Network*

We evaluate the significance of the *Reflectance Network* in our proposed framework. In this ablation study we evaluate for the task of simultaneous view synthesis and relighting, first using only $\mathcal{P}$, and then with our proposed framework, which involves both $\mathcal{P}$ and $\mathcal{R}$.

To perform viewpoint editing and relighting using only $\mathcal{P}$ we modify the network design slightly. Instead of illumination latent $\mathbf{z}_{\text{env}}$ we directly input the downsampled HDR environment map and train $\mathcal{P}$. This enables us to perform a one-to-one comparison with our full model involving both $\mathcal{P}$ and $\mathcal{R}$. To fit an unseen identity during test, we initialize the $\mathbf{z}_{\text{env}}$ with the environment map estimated from SIPR [1] trained on our lightstage dataset, followed by our two-step optimization process to reconstruct the unseen subject.

Quantitative evaluations in Tab. 6 show that using a dedicated $\mathcal{R}$ for relighting improves the overall performance by a good margin. This is evident in Fig. 18 (right) where we observe that using just $\mathcal{P}$ fails to capture the environment illumination conditions completely. In contrast, relighting using OLATs regressed from $\mathcal{R}$ closely matches the ground truth lighting condition, thereby validating our design choice.

Fit+FineTune		Fit only	
PSNR	SSIM	PSNR	SSIM
26.67	**0.82**	22.39	0.71

Table 5: We summarize quantitative results to evaluate the significance of our two-step optimization process. We observe that *fitting+fine-tuning* leads to better performance.

With $\mathcal{R}$		Without $\mathcal{R}$	
PSNR	SSIM	PSNR	SSIM
22.80	**0.76**	20.81	0.72

Table 6: We present quantitative results for emphasize the need for $\mathcal{R}$. Clearly having a dedicated $\mathcal{R}$ improves the relighting quality.

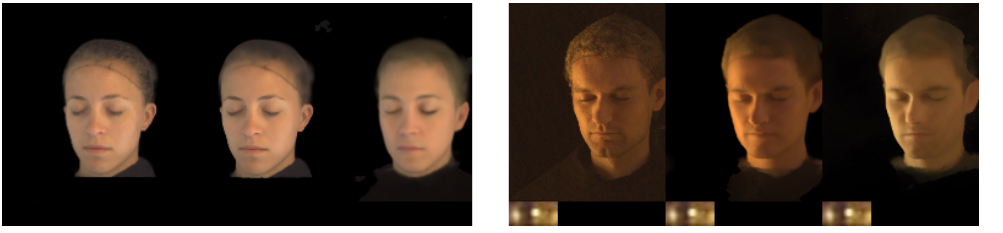


Figure 18: Left: Performing the two-step optimization improves the overall quality during view-synthesis. Right: Removing the *Reflectance Network* ("w/o shading") leads to a clear loss in quality during relighting.

9.3 Significance of number of OLATs

In this section, we examine the significance of the quality of relighting by utilizing different numbers of OLAT configurations: 50, 100, and 150 OLATs. We conduct evaluations for simultaneous view-synthesis and relighting.

Since the original lightstage dataset contains 150 OLATs, we uniformly sample from the original configuration to select 50 and 100 OLATconfigurations. Next, we train three different *Reflectance Network* models with various OLAT configurations for the same number of iterations. We summarize quantitative evaluations in Tab. 7 and observe that the quality of relighting increases with the increase in the number OLATs. This is distinctively clear from the Fig. 19, as the *Reflectance Network* trained with 150 OLATs shows better results in comparison. We reason that an increase in the number of OLATs leads to a better approximation of the environment illumination and as a consequence, it improves the quality of relighting. In summary, we conclude that a higher number of OLATs improves the quality of relighting. In this work, we are restricted to 150 OLATs since it is the capacity of the lightstage dataset available to us.

	50 OLATs		100 OLATs		150 OLATs	
Input	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
3-views	19.70	0.72	21.22	0.73	**22.80**	**0.76**

Table 7: Influence of number of OLATs for the task of simultaneous relighting and view-synthesis. Using all 150 OLATs gives the best results. In general, we observe that quality of relighting improves with the increasing number of OLATs.

9.4 The importance of the Hard Loss ($\mathcal{L}_h$)

In this section, we investigate the importance of the hard loss $\mathcal{L}_h$. This loss constrains *accumulation weights* $w_{r,k}$ to be sparse [10], thereby encouraging the face geometry to approximate a surface. This measure prevents cloudy artifacts around the face as shown in Fig. 20 (see red arrows). In our main experiments, we use a default value of 0.1 for the hard loss. Fig. 21 shows that using an over-emphasized value of 10 for the hard loss leads to severe artifacts. In Tab. 8 we examine the importance of the hard loss using quantitative evaluations against groundtruth. Here, we evaluate on the lightstage test set. As expected, completely removing the hard loss leads to a strong drop in the PSNR and SSIM, as opposed to using it with

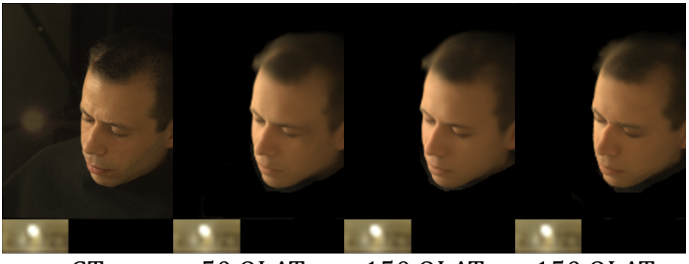


Figure 19: We show the significance of number OLATs on final relighting. During simultaneous view synthesis and relighting, we observe that with fewer OLATs, the *Reflectance Network* struggles to accurately relight the environment illumination. Hence, using all the 150 OLATs of the lightstage dataset gives closest resemblance to the groundtruth.

	$\mathcal{L}_h = 0.1$		$\mathcal{L}_h = 0$		$\mathcal{L}_h = 10$	
Input	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
3-views	**26.67**	**0.82**	25.65	0.78	19.81	0.64

Table 8: Impact of the hard loss $\mathcal{L}_h$ on novel view synthesis on the lightstage test set. Our default value of 0.1 for the hard loss produces the best results.

the default value of 0.1. However, using an over-emphasized value of 10 leads to very poor performance.

9.5 Latent Space Design

Our approach uses separate latent codes for identity ($\mathbf{z}_{id}$) and illumination ($\mathbf{z}_{env}$). During training, we store one $\mathbf{z}_{id}$ per subject and one $\mathbf{z}_{env}$ per illumination condition, amounting to $302 + 600$ codes. Each identity code receives supervision under different lighting conditions. Similarly, each illumination code receives supervision from all subjects. If a single code was used for each *combination* of identity and lighting condition, we would need to supervise 302×600 latent codes. Codes representing the same subject under different illuminations would not be supervised jointly anymore. To investigate this, we compare a "disentangled" model (i.e. $302 + 600$ codes) to one that uses one code per combination (i.e. 302×600 codes). After training both models for the same number of iterations, we tabulate our findings in Tab. 9: Having a single latent code for each identity and illumination leads to combinatorial explosion of latent parameters, making it difficult to learn a good face prior. Fig. 22 shows that using separate latent codes leads to better reconstructions of unseen subjects.

Disentangled Latent Codes		Single Latent Code	
PSNR	SSIM	PSNR	SSIM
26.66	**0.82**	24.53	0.78

Table 9: We evaluate the two latent space design choices on our lightstage dataset. We observe that using a disentangled latent space design leads to an improved performance, mainly attributed to a better face prior representation that helps generalize to unseen subjects.

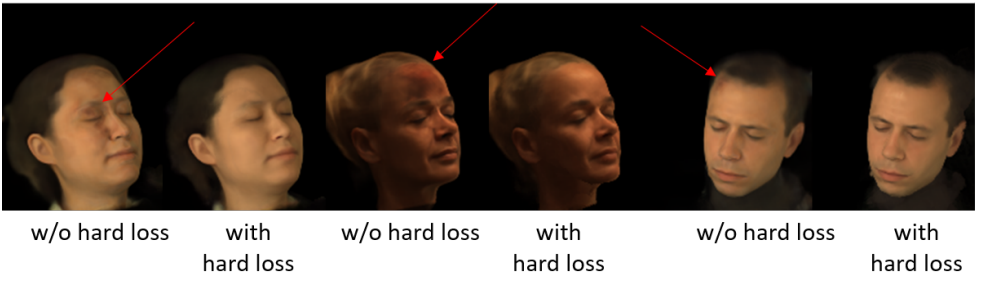


Figure 20: We visualize the importance of the hard loss $\mathcal{L}_h$ on the final results. Here, we show results from the view synthesis task. We use a default value of 0.1 for the hard loss. Removing the hard loss ($\mathcal{L}_h = 0$) produces significant cloudy artifacts as shown by the red arrows. Adding the hard loss ($\mathcal{L}_h = 0.1$) forces the volume to be more constrained around the head and thus removes such cloudy artifacts.

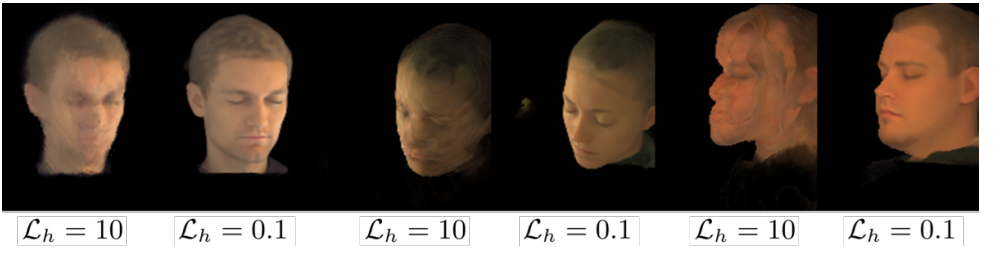


Figure 21: We visualize the impact of different values for the hard loss $\mathcal{L}_h$. The default value of the hard loss used in our experiments is 0.1. This figure shows that using an over-emphasized value of 10 leads to strong artifacts.

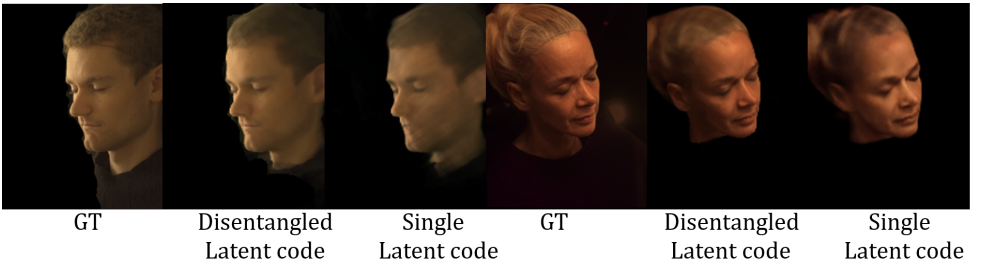


Figure 22: We compare the design choice *Distentantled Latent code* (i.e. separate latent codes for identity and illumintion) to the alternative *Single Latent code* (i.e. one latent code *per combination* of identity and illumination), by evaluating for the task of view synthesis on our lightstage dataset. The disentangled version leads to better reconstructions.

9.6 Latent Space Interpolations

Our identity and illumination latent codes enable us to disentangle identity and illumination and to modify them in isolation. Fig. 23 shows linear interpolations between various identities of our lightstage data. Please refer to the **supplemental video** – *Latent_Interpolation.mp4* for additional latent space interpolations results.

10 Discussion and Limitations

As discussed in the main paper, although our proposed method can render photo-realistic human heads, some limitations still exist:

The light stage dataset (see Sec. 4) shows all subjects with closed eyes and neutral expressions only. However, even though our method thus only sees closed eyes and neutral expression during training, it can handle test subjects with open eyes and natural expressions during view synthesis. We showcase this ability in Figs. 25 and 26 and attribute it to the fine-tuning of the *Face Prior Network*. The figures show that our approach can preserve eyes and facial expressions at arbitrary viewpoints. However, for the task of relighting we rely on the *Reflectance Network* to regress OLATs, which is trained on the lightstage dataset which neither contains open eyes nor different facial expressions. Hence, VoRF is unable to estimate the reflectances of open eyes and mouth interiors during relighting. Please refer to **supplemental video** – *Face_Expressions.mp4*. We argue that this is a limitation inherited from the dataset and not a shortcoming of our proposed method.

For the same reason, our *Reflectance Network* struggles to synthesize accessories such as spectacles, earrings etc. Fig. 24

Lastly, if only one input view is given, our approach can sometimes generate regions of geometry/texture that do not exist in reality, such as the hair in in Fig. 27. In such cases, the information from a single view simply proves to be insufficient.

Figure 23: Identity and illumination interpolations between lightstage training subjects. Results show that our method disentangles the identity and illumination correctly.

Figure 24: We show simultaneous view synthesis and relighting on the CelebA dataset.

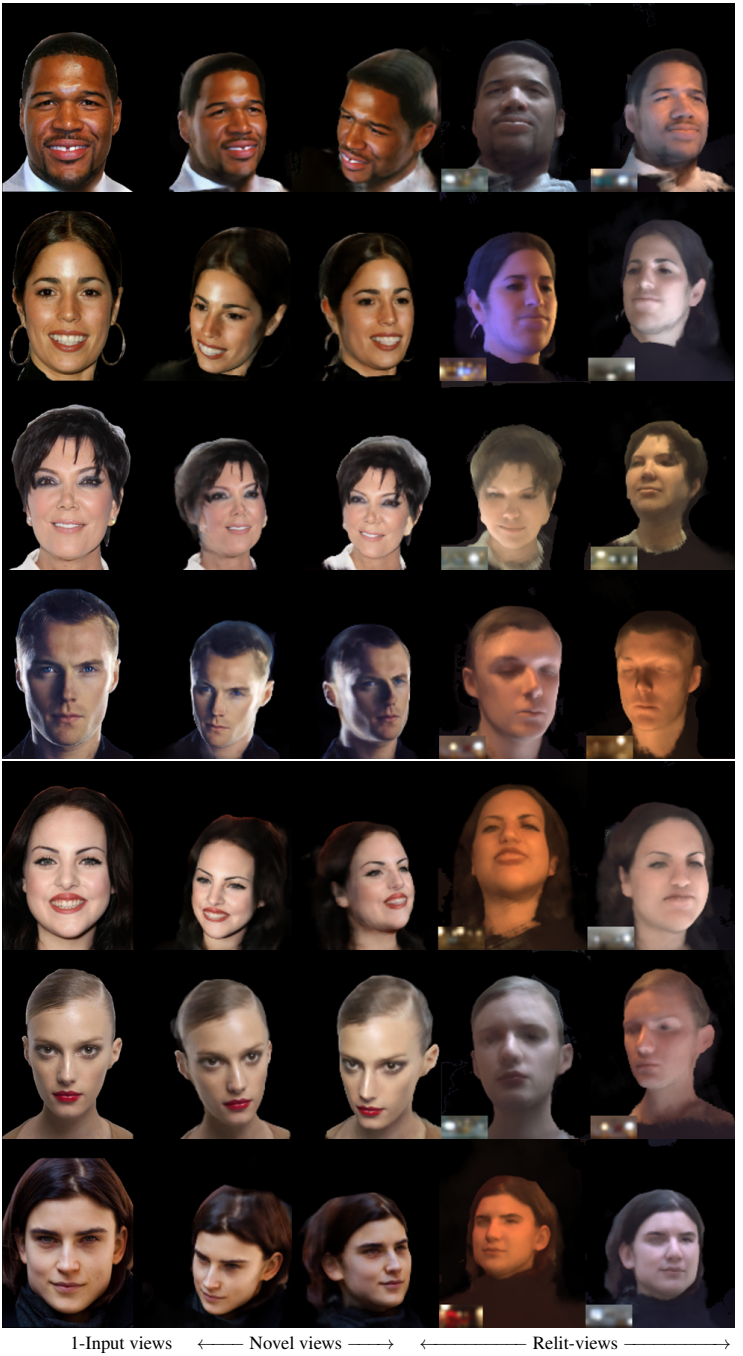


Figure 25: We show simultaneous view synthesis and relighting on the CelebA dataset.

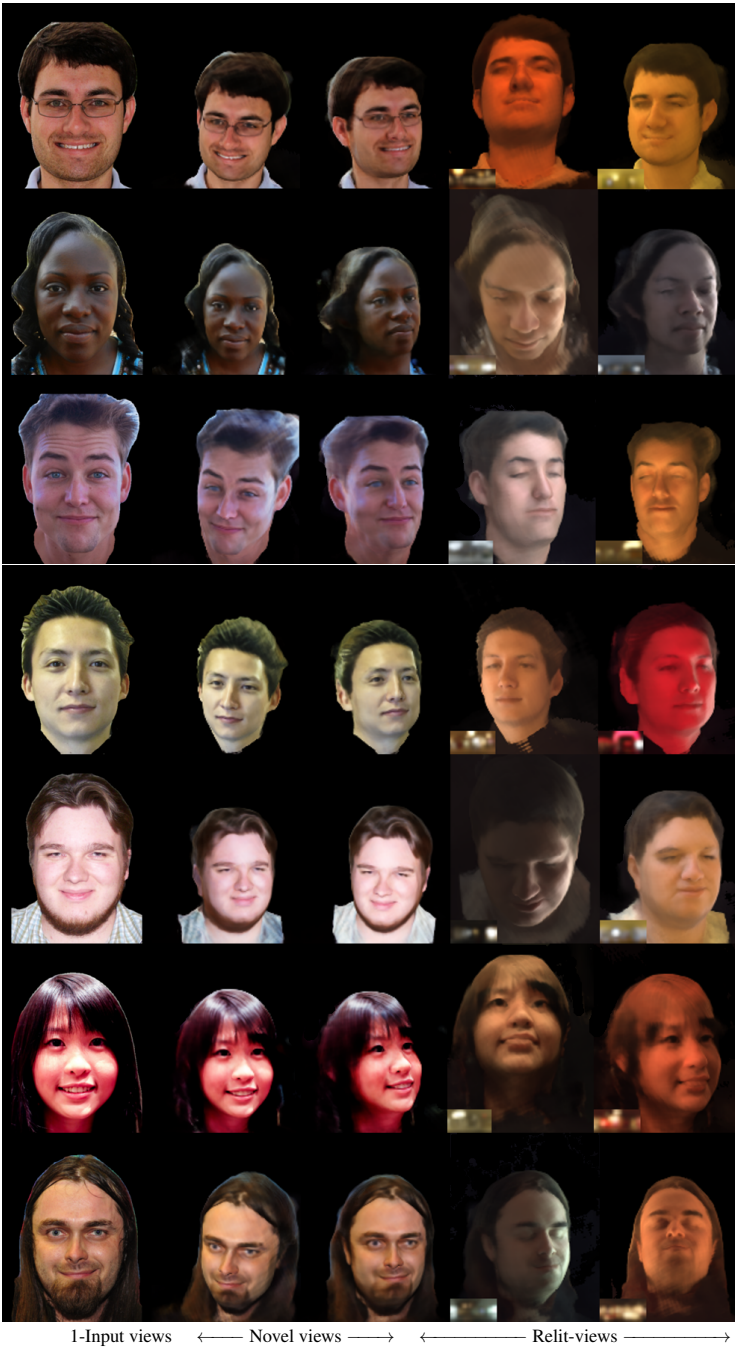


Figure 26: We show results for simultaneous view synthesis and relighting on the FFHQ dataset.

Figure 27: Our method produces good relighting and view synthesis for 3 input views, 2 input views or even 1 single input view. However, it may sometimes generate features that do not exist in reality, but are not ruled out by the input: In the 1-view case presented here, our method added hair, which does not contradict the information present in the single input view (frontal view on the very left of this figure).

References

[1] Adrian Bulat and Georgios Tzimiropoulos. How far are we from solving the 2d & 3d face alignment problem? (and a dataset of 230,000 3d facial landmarks). In *International Conference on Computer Vision*, 2017.

[2] Paul Debevec, Tim Hawkins, Chris Tchou, Haarm-Pieter Duiker, Westley Sarokin, and Mark Sagar. Acquiring the reflectance field of a human face. In *Annual conference on Computer graphics and interactive techniques*, 2000.

[3] Marc-André Gardner, Kalyan Sunkavalli, Ersin Yumer, Xiaohui Shen, Emiliano Gambaretto, Christian Gagné, and Jean-François Lalonde. Learning to predict indoor illumination from a single image, 2017.

[4] Yannick Hold-Geoffroy, Akshaya Athawale, and Jean-François Lalonde. Deep sky modeling for single image outdoor lighting estimation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[5] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of StyleGAN. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[6] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2015.

[7] Mallikarjun B R, Ayush Tewari, Abdallah Dib, Tim Weyrich, Bernd Bickel, Hans-Peter Seidel, Hanspeter Pfister, Wojciech Matusik, Louis Chevallier, Mohamed Elgharib, et al. Photoapp: Photorealistic appearance editing of head portraits. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, 2021.

[8] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision (ECCV)*, 2020.

[9] Eduard Ramon, Gil Triginer, Janna Escur, Albert Pumarola, Jaime Garcia, Xavier Giro-i Nieto, and Francesc Moreno-Noguer. H3d-net: Few-shot high-fidelity 3d head reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5620–5629, 2021.

[10] Daniel Rebain, Mark Matthews, Kwang Moo Yi, Dmitry Lagun, and Andrea Tagliasacchi. Lolnerf: Learn from one look. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[11] Tiancheng Sun, Zexiang Xu, Xiuming Zhang, Sean Fanello, Christoph Rhemann, Paul Debevec, Yun-Ta Tsai, Jonathan T. Barron, and Ravi Ramamoorthi. Light stage super-resolution: Continuous high-frequency relighting. In *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia)*, 2020.

[12] Tiancheng Sun, Kai-En Lin, Sai Bi, Zexiang Xu, and Ravi Ramamoorthi. Nelf: Neural light-transport field for portrait view synthesis and relighting. In *Eurographics Symposium on Rendering*, 2021.

[13] Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul Srinivasan, Howard Zhou, Jonathan T. Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. Ibrnet: Learning multi-view image-based rendering. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[14] Zhibo Wang, Xin Yu, Ming Lu, Quan Wang, Chen Qian, and Feng Xu. Single image portrait relighting via explicit multiple reflectance channel modeling. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia)*, 2020.

[15] Tim Weyrich, Wojciech Matusik, Hanspeter Pfister, Bernd Bickel, Craig Donner, Chien Tu, Janet McAndless, Jinho Lee, Addy Ngan, Henrik Wann Jensen, and Markus Gross. Analysis of human faces using a measurement-based skin reflectance model. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, 2006.