Relighting4D: Neural Relightable Human from Videos

Zhaoxi Chen and Ziwei Liu*

S-Lab, Nanyang Technological University
{zhaoxi001, ziwei.liu}@ntu.edu.sg

Supplementary Material

1 BRDF Implementation

We use the standard microfacet Bi-directional Reflectance Distribution Function [11] while introducing some approximations used in the BRDF implementations of the Blender [3] rendering engine. $R(\boldsymbol{x}, \boldsymbol{\omega}_i, \boldsymbol{\omega}_o, \boldsymbol{n}(\boldsymbol{x}))$ is defined for the 3D location $\boldsymbol{x}$, incident lighting direction $\boldsymbol{\omega}_i$, outgoing reflectance direction $\boldsymbol{\omega}_o$, and surface normal $\boldsymbol{n}(\boldsymbol{x})$) as:

$$R(\boldsymbol{x}, \boldsymbol{\omega}_i, \boldsymbol{\omega}_o, \boldsymbol{n}(\boldsymbol{x})) = \frac{\boldsymbol{A}(\boldsymbol{x})}{\pi} + \frac{D(\boldsymbol{h}, \boldsymbol{n}(\boldsymbol{x})), \gamma(\boldsymbol{x})) \cdot F(\boldsymbol{h}, \boldsymbol{\omega}_i) \cdot G(\boldsymbol{h}, \boldsymbol{n}(\boldsymbol{x}), \boldsymbol{\omega}_o, \gamma(\boldsymbol{x}))}{4(\boldsymbol{\omega}_o \cdot \boldsymbol{n}(\boldsymbol{x}))(\boldsymbol{\omega}_i \cdot \boldsymbol{n}(\boldsymbol{x}))},$$

$$D(\boldsymbol{h}, \boldsymbol{n}(\boldsymbol{x}), \gamma(\boldsymbol{x})) = \frac{\alpha^2}{\pi((\boldsymbol{h} \cdot \boldsymbol{n}(\boldsymbol{x}))^2(\alpha^2 - 1) + 1)^2},$$

$$F(\boldsymbol{h}, \boldsymbol{\omega}_i) = F_0 + (1 - F_0)(1 - (\boldsymbol{h} \cdot \boldsymbol{\omega}_i))^5,$$

$$G(\boldsymbol{h}, \boldsymbol{n}(\boldsymbol{x}), \boldsymbol{\omega}_o, \gamma(\boldsymbol{x})) = \frac{\boldsymbol{h} \cdot \boldsymbol{\omega}_o}{\boldsymbol{n}(\boldsymbol{x}) \cdot \boldsymbol{\omega}_o} \cdot \frac{2}{1 + \sqrt{1 + \alpha^2 \tan\theta}},$$

$$\alpha = \gamma^2(\boldsymbol{x}), \qquad \boldsymbol{h} = \frac{\boldsymbol{\omega}_o + \boldsymbol{\omega}_i}{||\boldsymbol{\omega}_o + \boldsymbol{\omega}_i||}, \qquad \tan\theta = \frac{1 - (\boldsymbol{n}(\boldsymbol{x}) \cdot \boldsymbol{\omega}_o)^2}{(\boldsymbol{n}(\boldsymbol{x}) \cdot \boldsymbol{\omega}_o)^2},$$

where $\boldsymbol{A}(\boldsymbol{x})$ is the diffuse map, $\gamma(\boldsymbol{x})$ is the specular roughness. In our implementations, we set Fresnel coefficient $F_0 = 0.04$.

2 Estimation Error of $\tilde{p}(\boldsymbol{A}(\boldsymbol{x}))$

In Section 3.4, we use a Gaussian KDE $\tilde{p}(\boldsymbol{A}(\boldsymbol{x}))$ to estimate the PDF of diffuse map $\boldsymbol{A}(\boldsymbol{x})$ during training. For brevity of writing, we omit the bandwidth parameter h of KDE in the main paper. Actually, the kernel K_G is the scaled Gaussian function, that is defined as $K_G(x) = \frac{1}{h\sqrt{2\pi}} exp(\frac{-(x/h)^2}{2})$. Empirically, we set the bandwidth $h = Var(\boldsymbol{A}(\boldsymbol{x}))$. Here we derive the error of our approximation.

*Corresponding author.

We denote $K(x) = \frac{1}{\sqrt{2\pi}} exp(\frac{-x^2}{2})$, thus $K_G(x) = \frac{1}{h}K(\frac{x}{h})$. Given a point $\boldsymbol{x}_0$, the bias of $\tilde{p}(\boldsymbol{A}(\boldsymbol{x}_0))$ is:

$$\mathbb{E}[\tilde{p}(\boldsymbol{A}(\boldsymbol{x}_0))] - p(\boldsymbol{A}(\boldsymbol{x}_0)) = \frac{1}{2}h^2 p''(\boldsymbol{A}(\boldsymbol{x}_0)) \cdot \int y^2 K(y) dy + o(h^2). \tag{1}$$

The variance of $\tilde{p}(\boldsymbol{A}(\boldsymbol{x}_0))$ is:

$$Var(\tilde{p}(\boldsymbol{A}(\boldsymbol{x}_0))) \leq \frac{1}{nh^2}\mathbb{E}[K^2(\frac{\boldsymbol{A}(\boldsymbol{x}_0) - \boldsymbol{A}_i(\boldsymbol{x})}{h})] \tag{2}$$

$$= \frac{1}{nh} p(\boldsymbol{A}(\boldsymbol{x}_0)) \cdot \int K^2(y) dy + o(\frac{1}{nh}). \tag{3}$$

Thus, the mean square error of $\tilde{p}(\boldsymbol{A}(\boldsymbol{x}_0))$ is:

$$MSE(\tilde{p}(\boldsymbol{A}(\boldsymbol{x}_0))) = [\mathbb{E}[\tilde{p}(\boldsymbol{A}(\boldsymbol{x}_0))] - p(\boldsymbol{A}(\boldsymbol{x}_0))]^2 + Var(\tilde{p}(\boldsymbol{A}(\boldsymbol{x}_0))) \tag{4}$$

$$= O(h^4) + O(\frac{1}{nh}). \tag{5}$$

Since $h = Var(\boldsymbol{A}(\boldsymbol{x})) = \sum(\boldsymbol{A}_i(\boldsymbol{x}) - \mathbb{E}(\boldsymbol{A}(\boldsymbol{x})))^2/n$, n is the number of sampled camera rays and $\boldsymbol{A}(\boldsymbol{x}) \in [0, 1]$, our approximation will maintain a promising error bound when we sampling enough camera rays at each iteration during training.

3 Experiment Details

3.1 Data Pre-processing

The input of *Relighting4D* is assumed to be posed human videos, which contain human videos with known camera extrinsic and intrinsic parameters. Before the training, we first extract the parameters of the human model [6, 9] from the videos. Specifically, as for videos with simple motions, People-Snapshot dataset, the SMPL parameters can be accurately estimated from the monocular inputs [2]. And we estimate SMPL parameters of the ZJU-Mocap dataset [10] from the multi-view images using off-the-shelf tools [4].

3.2 Training Hyperparameters

As discussed in Section 3.5 of the paper, we introduce a set of hyperparameters to stabilize the training. We set $\lambda_{rgb} = 10, \lambda_A = 0.005, \lambda_H = 0.0005, \lambda_{temp} = 0.1$ for all scenes. Due to the difference in scale, the values of some hyperparameters vary across datasets. For the People-Snapshot dataset, we set $\lambda_{geo} = 1, \lambda_V = 0.5, \lambda_n = 0.01$. For the ZJU-Mocap dataset, we set $\lambda_{geo} = 1, \lambda_V = 0.5, \lambda_n = 0.05$. For the synthetic BlenderHuman dataset, we set $\lambda_{geo} = 0.05, \lambda_V = 0.025, \lambda_n = 0.025$. Moreover, during the process of baking the geometry (Section 3.3), the unit of s_n, s_f is the metre.

As for the progressive training (Section 3.5), the scaling factor α starts from 0.1 and linearly increases to 1.0 every 5k iterations. For example, on People-Snapshot [2] dataset, the resolution of video frames starts from 108×108 and increases to 1080×1080 after 50k iterations.

We minimize the training objective using Adam [5] optimizer with a learning rate that starts from 5×10^{-4} and exponentially decays to 5×10^{-5} for 260k iterations.

3.3 Comparison Methods

NeRFactor [13]. We re-implement NeRFactor in PyTorch [8] based on its TensorFlow [1] version* under Apache-2.0 License. Note that, the original NeRFactor uses a static NeRF [7] as the geometry proxy which is definitely not reasonable to directly use on the dynamic scenes. Thus, we adapt NeRFactor by fitting a dynamic neural radiance field [10] as its geometry proxy.

PhySG [12]. We adapt PhySG in PyTorch based on its original version† under MIT License. Note that, the original PhySG leverage a static signed distance function (SDF) as the representation of geometry which doesn't fit dynamic scenes. However, in our experiments, we found that fitting a SDF on dynamic scenes is a non-trivial task. Thus, to make a fair comparison, we use NeuralBody [10] to provide a more accurate geometry information to the reflectance model of PhySG. In Specific, the spherical Gaussian and reflectance model are keeping unchanged while surfaces of geometry are obtained from NeuralBody [10].

NB [10]+A. NeuralBody(NB) uses an MLP M_c as the color model to predict RGB values $\boldsymbol{c}_t(\boldsymbol{x})$ based on its defined features $v_t(\boldsymbol{x})$, i.e. $\boldsymbol{c}_t(\boldsymbol{x}) = M_c(v_t(\boldsymbol{x}))$. In our paper, we incorporate lighting on top of NeuralBody by concatenating the light probe with the feature $v_t(\boldsymbol{x})$. In specific, the light probe with the resolution of $16 \times 32 \times 3$ is flattened to a vector $\bar{L}$ with dimension of 1536, and then concatenated with $v_t(\boldsymbol{x})$ as the input of M_c. Thus, the color model of NB+A is defined as $M_c(v_t(\boldsymbol{x}), \bar{L})$.

NB [10]+LE. Different from NB+A, NA+LE first uses another two-layered MLP M_L to map the light probe into a latent vector with dimension of 32. Therefore, the color model of NB+LE is defined as $M_c(v_t(\boldsymbol{x}), M_L(\bar{L}))$.

3.4 Ambient Light Probes

We collect multiple light probes from the online non-commercial website‡ as light sources to do relighting, which are stored in High-Dynamic-Range (HDR) format. We show the correspondence of the light probes used in our experiments and their original high-resolution ones in Figure 2.

*`https://github.com/google/nerfactor`

†`https://github.com/Kai-46/PhySG`

‡`https://polyhaven.com`

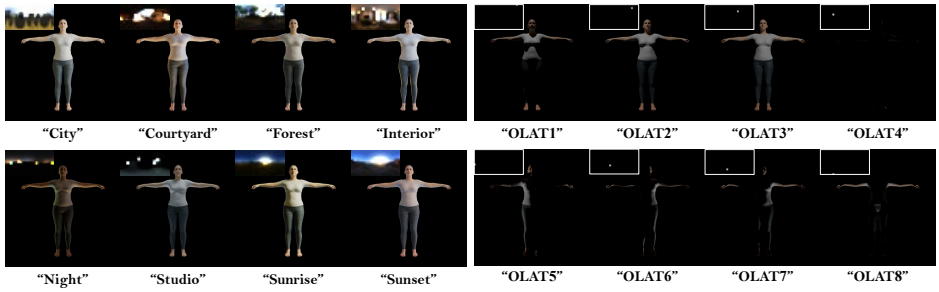

Fig. 1. Overview of test sequences on the BlenderHuman Dataset.

4 Supplementary Results on the BlenderHuman Dataset

The synthetic dataset (Figure 1), BlenderHuman, is constructed with the help of SMPL-X Blender Add-on[§] using the Blender [3] engine. We generate 17 sequences of a human actor under different illuminations, and each sequence contains 200 frames in 1024×1024 resolution. We use the physically based path-tracer, Cycles[¶], to render the video frames. The actor moves in the way that is same as the Peple-Snapshot [2] dataset. We use one sequence for training, and test on the rest sequences. Figure 1 shows our test sequences.

We show qualitative comparisons with other methods in Figure 4, and results of geometry and reflectance decomposition in Figure 3. Furthermore, we present qualitative ablation studies in Figure 5. Besides, per-scene relighting results (PSNR, SSIM, and LPIPS) are presented in Table 1, Table 2, and Table 3 seperately.

5 Supplementary Results on Real Datasets

Please check the supplementary videos[‖] for more comprehensive visualizations and results.

[§] https://gitlab.tuebingen.mpg.de/jtesch/smplx_blender_addon

[¶] https://www.cycles-renderer.org/

[‖] https://frozenburning.github.io/projects/relighting4d/

Fig. 2. Correspondence of the used light probes and their 8K versions.

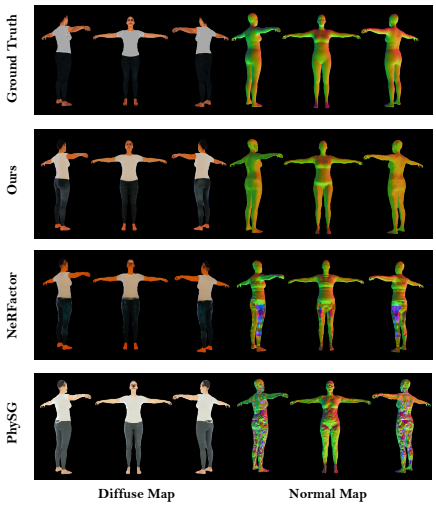

Fig. 3. Geometry and reflectance decomposition results on the BlenderHuman dataset.

References

1. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., Zheng, X.: TensorFlow: Large-scale machine learning on heterogeneous systems (2015), `https://www.tensorflow.org/`, software available from tensorflow.org
2. Alldieck, T., Magnor, M., Xu, W., Theobalt, C., Pons-Moll, G.: Video Based Reconstruction of 3D People Models. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8387–8397. IEEE (Jun 2018). https://doi.org/10.1109/CVPR.2018.00875, `https://ieeexplore.ieee.org/document/8578973/`
3. Community, B.O.: Blender - a 3D modelling and rendering package. Blender Foundation (2018), `http://www.blender.org`
4. Joo, H., Simon, T., Sheikh, Y.: Total capture: A 3d deformation model for tracking faces, hands, and bodies (2018)
5. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization (2017)
6. Loper, M., Mahmood, N., Romero, J., Pons-Moll, G., Black, M.J.: SMPL: A skinned multi-person linear model. ACM Trans. Graphics (Proc. SIGGRAPH Asia) **34**(6), 248:1–248:16 (Oct 2015)
7. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. arXiv:2003.08934 [cs] (Aug 2020), `http://arxiv.org/abs/2003.08934`

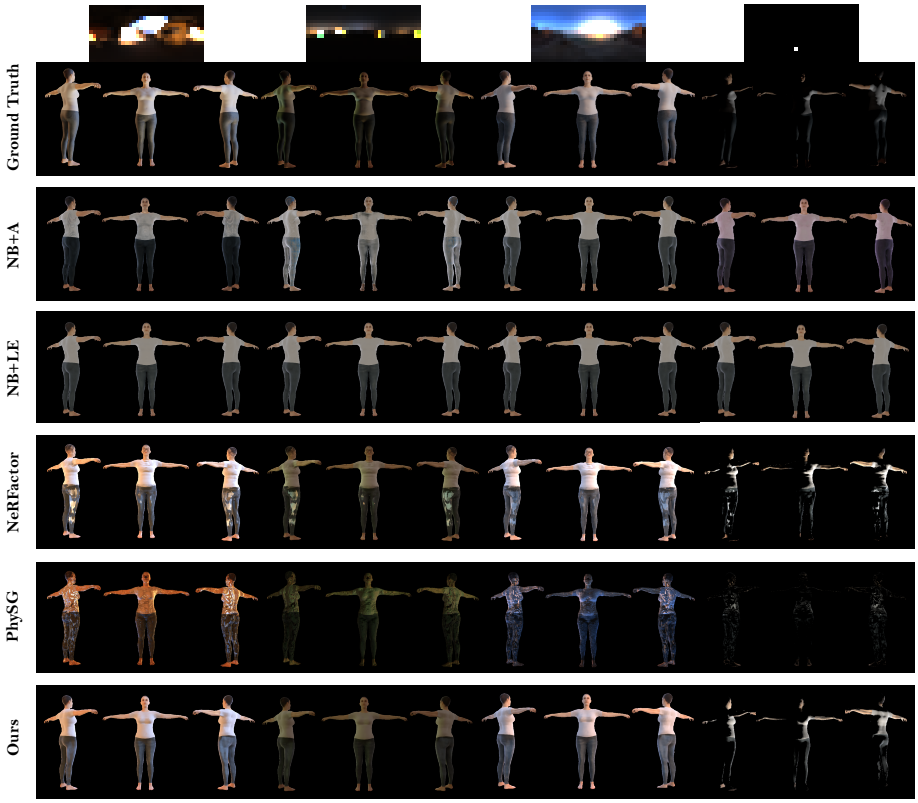

Fig. 4. Comparison results on the BlenderHuman dataset.

8. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: Pytorch: An imperative style, high-performance deep learning library. In: Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R. (eds.) Advances in Neural Information Processing Systems 32, pp. 8024–8035. Curran Associates, Inc. (2019), http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf
9. Pavlakos, G., Choutas, V., Ghorbani, N., Bolkart, T., Osman, A.A.A., Tzionas, D., Black, M.J.: Expressive body capture: 3D hands, face, and body from a single image. In: Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). pp. 10975–10985 (2019)
10. Peng, S., Zhang, Y., Xu, Y., Wang, Q., Shuai, Q., Bao, H., Zhou, X.: Neural Body: Implicit Neural Representations with Structured Latent Codes for Novel View Synthesis of Dynamic Humans. arXiv:2012.15838 [cs] (Mar 2021), http://arxiv.org/abs/2012.15838

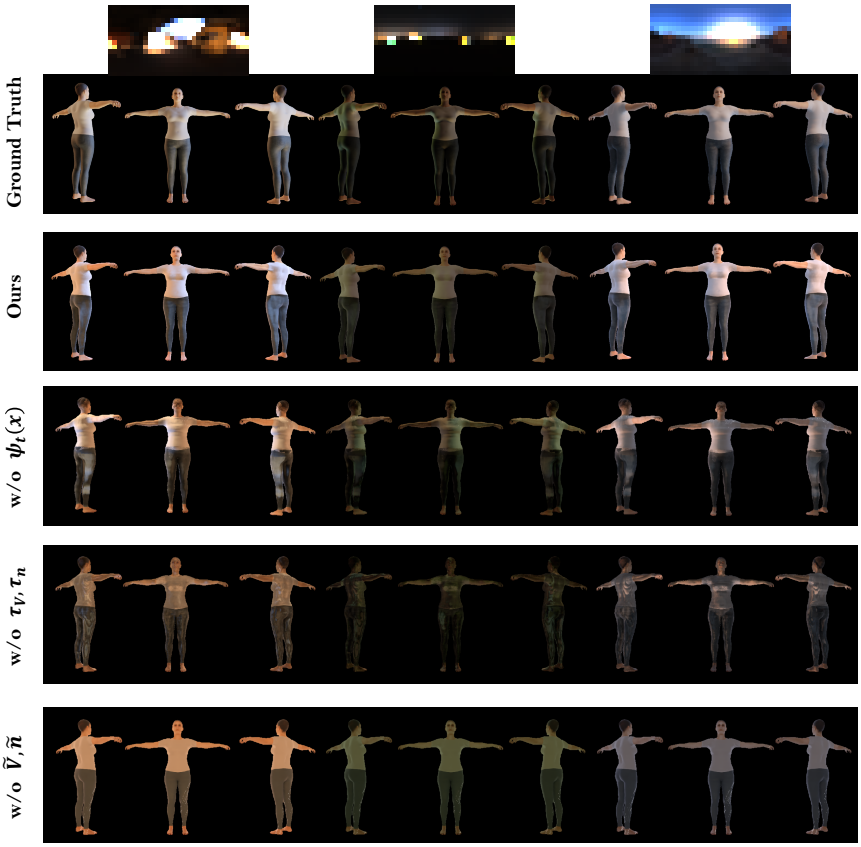

Fig. 5. Ablation Studies on the BlenderHuman dataset.

11. Walter, B., Marschner, S.R., Li, H., Torrance, K.E.: Microfacet models for refraction through rough surfaces. In: Proceedings of the 18th Eurographics Conference on Rendering Techniques. pp. 195–206. EGSR'07, Eurographics Association (2007). https://doi.org/10.2312/EGWR/EGSR07/195-206
12. Zhang, K., Luan, F., Wang, Q., Bala, K., Snavely, N.: PhySG: Inverse Rendering with Spherical Gaussians for Physics-based Material Editing and Relighting. arXiv:2104.00674 [cs] (Apr 2021), http://arxiv.org/abs/2104.00674
13. Zhang, X., Srinivasan, P.P., Deng, B., Debevec, P., Freeman, W.T., Barron, J.T.: NeRFactor: Neural Factorization of Shape and Reflectance Under an Unknown Illumination. arXiv:2106.01970 [cs] (Jun 2021), http://arxiv.org/abs/2106.01970

Table 1. Per-Scene results(PSNR) on the BlenderHuman dataset. The reported numbers are the arithmetic averages of 200 videos frames on each scene. The top two techniques for each metric are highlighted in red and orange respectively. We relight the human actor with 8 HDR ambient light probes and 8 OLAT conditions (shown as Figure 1).

Scene	PSNR ↑				
	NeRFactor	PhySG	NB+A	NB+LE	Ours
City	21.4328	19.6419	23.1069	24.6057	23.0075
Courtyard	22.6981	21.9982	25.5212	25.3169	26.0427
Forest	20.8999	20.1738	26.2215	26.3270	22.9896
Interior	22.0315	20.8893	21.6186	24.9819	24.9926
Night	25.4527	29.0631	21.6186	23.3812	29.8210
Studio	20.9773	23.5288	23.8103	28.2641	24.9691
Sunrise	21.1778	21.2953	23.0263	23.7437	22.8182
Sunset	22.6371	21.6861	27.0982	27.4887	25.9497
OLAT1	26.5106	26.3107	15.8248	19.9056	28.3209
OLAT2	19.1983	23.0530	17.6058	21.8662	20.4326
OLAT3	17.0915	21.7216	20.5199	23.6546	17.7695
OLAT4	26.6496	23.2572	18.9222	18.1458	43.6499
OLAT5	19.7225	24.4148	17.1426	19.8696	22.0923
OLAT6	22.8289	26.5456	19.0160	19.1561	25.5454
OLAT7	23.0537	27.1507	19.0595	19.4215	25.0339
OLAT8	32.4976	31.3656	14.8438	18.6029	34.9244
average	22.8037	23.8810	20.9348	22.7957	**26.1475**

Table 2. Per-Scene results(SSIM) on the BlenderHuman dataset. The reported numbers are the arithmetic averages of 200 videos frames on each scene. The top two techniques for each metric are highlighted in red and orange respectively. We relight the human actor with 8 HDR ambient light probes and 8 OLAT conditions (shown as Figure 1).

Scene	SSIM ↑				
	NeRFactor	PhySG	NB+A	NB+LE	Ours
City	0.8862	0.8116	0.9029	0.9268	0.9143
Courtyard	0.8883	0.8409	0.9280	0.9293	0.9254
Forest	0.8839	0.8107	0.9314	0.9323	0.9167
Interior	0.8982	0.8220	0.8744	0.9305	0.9326
Night	0.8973	0.8939	0.8744	0.8992	0.9330
Studio	0.8812	0.8850	0.9048	0.9372	0.9241
Sunrise	0.8794	0.8022	0.8946	0.9047	0.9062
Sunset	0.8979	0.8396	0.9345	0.9400	0.9355
OLAT1	0.8653	0.8906	0.8002	0.8119	0.8864
OLAT2	0.8272	0.8115	0.8193	0.8582	0.8480
OLAT3	0.8295	0.7934	0.8433	0.8903	0.8479
OLAT4	0.9433	0.8292	0.7746	0.7692	0.9766
OLAT5	0.8617	0.8064	0.8087	0.8195	0.8859
OLAT6	0.8856	0.8554	0.8071	0.8072	0.9140
OLAT7	0.8770	0.8498	0.8161	0.8137	0.9008
OLAT8	0.9253	0.9415	0.7785	0.7832	0.9416
average	0.8830	0.8427	0.8559	0.8721	**0.9118**

Table 3. Per-Scene results(LPIPS) on the BlenderHuman dataset. The reported numbers are the arithmetic averages of 200 videos frames on each scene. The top two techniques for each metric are highlighted in red and orange respectively. We relight the human actor with 8 HDR ambient light probes and 8 OLAT conditions (shown as Figure 1).

Scene	LPIPS ↓				
	NeRFactor	PhySG	NB+A	NB+LE	Ours
City	0.1647	0.3043	0.1383	0.1301	0.1366
Courtyard	0.1539	0.2469	0.1253	0.1281	0.1158
Forest	0.1611	0.3092	0.1164	0.1218	0.1254
Interior	0.1502	0.2752	0.1480	0.1188	0.1128
Night	0.1640	0.2323	0.1779	0.1477	0.1073
Studio	0.1648	0.2373	0.1486	0.1306	0.1256
Sunrise	0.1685	0.3215	0.1635	0.1482	0.1359
Sunset	0.1467	0.2601	0.1164	0.1145	0.1093
OLAT1	0.2293	0.3195	0.3426	0.2953	0.2131
OLAT2	0.2553	0.3157	0.2884	0.2421	0.2317
OLAT3	0.2599	0.3330	0.2794	0.2168	0.2347
OLAT4	0.2954	0.3992	0.4062	0.4189	0.1672
OLAT5	0.2651	0.3033	0.3548	0.2886	0.2263
OLAT6	0.2466	0.3131	0.2950	0.3035	0.1996
OLAT7	0.2448	0.3070	0.2904	0.2918	0.1989
OLAT8	0.2023	0.2570	0.3976	0.3344	0.1821
average	0.2045	0.2959	0.2368	0.2145	**0.1639**