Factorized and Controllable Neural Re-Rendering of Outdoor Scene for Photo Extrapolation Supplementary Material

Boming Zhao*
bmzhao@zju.edu.cn
Zhejiang University

Bangbang Yang*
ybbbbt@gmail.com
Zhejiang University

Zhenyang Li
zhenyounglee@gmail.com
Baidu.com

Zuoyue Li
zuli@student.ethz.ch
ETH Zürich

Guofeng Zhang
zhangguofeng@zju.edu.cn
Zhejiang University

Jiashu Zhao
jzhao@wlu.ca
Wilfrid Laurier University

Dawei Yin
yindawei@acm.org
Baidu.com

Zhaopeng Cui†
zhpcui@gmail.com
Zhejiang University

Hujun Bao†
bao@cad.zju.edu.cn
Zhejiang University

In this supplementary material, we will describe more details of our method, including a preliminary that describes the acronyms, concepts and related rendering models of our work in Sec. A, model architectures (geometry / base appearance MLP, shadow MLP, sky generator and tone mapper) in Sec. B, details of training and extrapolated 3D photo generation in Sec. C. Besides, we also provide more discussions of limitations in Sec. D and experiment results in Sec. E.

A Preliminary

A.1 NeRF and NeuS Preliminaries

Neural Radiance Fields (NeRF) [11]. NeRF represents scene with several MLP layers, which take as input a 3D position $\mathbf{x} = (x, y, z)$ and viewing direction $\mathbf{d} = (d_x, d_y, d_z)$, and produce as output a density σ and color $\mathbf{c} = (r, g, b)$. To render the color of a single pixel, NeRF approximates the volume rendering integral. Each pixel in an image corresponds to a ray $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$. To render the color of $\mathbf{r}$, NeRF draws point samples with distances $\{t_i\}_{i=1}^{N}$ along the ray, and passes the point locations $\mathbf{r}(t_i)$ as well as view directions $\mathbf{d}$ to obtain density σ_i and colors $\mathbf{c}_i$. The resulting color is rendered following the quadrature rules [9], which is defined as:

$$\begin{aligned} \hat{C}(\mathbf{r}) = \mathcal{R}(\mathbf{r}, \mathbf{c}, \sigma) &= \sum_{k=1}^{K} T(t_k)\alpha\left(\sigma(t_k)\delta(t_k)\right)\mathbf{c}(t_k), \\ T(t_k) = \exp(-\sum_{k'=1}^{k-1} \sigma(t_{k'})\delta_{k'}), &\quad \alpha(x) = 1 - \exp(-x), \end{aligned} \tag{1}$$

where $\mathcal{R}(\mathbf{r}, \mathbf{c}, \sigma)$ is the volumetric rendering through ray $\mathbf{r}$ of color $\mathbf{c}$ with density σ, $\mathbf{c}(t)$ and $\sigma(t)$ are the color and density at point $\mathbf{r}(t)$ respectively, and $\delta_k = t_{k+1} - t_k$ is the distance between two adjacent sampling points on the ray. Stratified sampling and informed sampling are used to select sample points $\{t_k\}_{k=1}^{K}$ between the near plane t_n and far plane t_f.

To enable the NeRF MLP to represent higher frequency detail, the inputs $\mathbf{x}$ and $\mathbf{d}$ need to be preprocessed by a positional encoding γ_{PE}:

$$\gamma_{PE}(p) = [\sin(2^0\pi p), \cos(2^0\pi p), ..., \sin(2^{L-1}\pi p), \cos(2^{L-1}\pi p)], \tag{2}$$

where L is the frequency of positional encoding.

Neural Implicit Surfaces [15]. Although NeRF can synthesize high-quality novel view images, extracting high-fidelity surface from the learned implicit field is difficult due to lack of sufficient constraints on its level sets. In contrast, NeuS selects SDF functions as the representation of scene geometry, which delivers exact surfaces and smooth normal that facilitate our factorized neural re-rendering model. The design of NeuS's rendering formulation (see Eq. (1) in the main paper) aims at achieving transmittance's maximal slope precisely at the zero-crossing of its SDF, allowing an unbiased estimate of the corresponding surface.

A.2 The Rendering Equation

Though recent neural rendering approaches [8, 10] have achieved great success in modeling outdoor lighting variations with latent appearance code, they are not capable of controllable re-rendering with user-selected lighting effects. Inspired by explicit lighting models [6, 19], we model scene appearance with a rendering formulation that is derived from the classical rendering equation [6]. Here, we first briefly introduce the rendering equation. For a surface point $\mathbf{x}$ with surface normal $\mathbf{n}$, suppose $L_i(\omega_i; \mathbf{x})$ is the incident light intensity at location $\mathbf{x}$ along the direction ω_i, and BRDF $f_r(\omega_o, \omega_i; \mathbf{x})$ is the reflectance coefficient of the material at location $\mathbf{x}$ for incident light direction ω_i and viewing direction ω_o, then the observed light light intensity $L_o(\omega_o; \mathbf{x})$ is an integral over the hemisphere $\Omega = \{\omega_i : \omega_i \cdot \mathbf{n} > 0\}$:

$$L_o(\omega_o; \mathbf{x}) = \int_{\Omega} L_i(\mathbf{x}, \omega_i) f_r(\omega_o, \omega_i; \mathbf{x})(\omega_i \cdot \mathbf{n}) \mathrm{d}\omega_i. \tag{3}$$

Since our goal is to model outdoor scenes, we adopt the Lambertian reflectance assumption [7, 17, 18], so the diffusion term has the same reflection intensity for all ω_o in the hemisphere. Hence, in

*Boming Zhao and Bangbang Yang contributed equally to this work. The authors from Zhejiang University are also affiliated with the State Key Lab of CAD&CG.
†Corresponding authors: Hujun Bao and Zhaopeng Cui.

our setting, Eq. (3) can be rewritten as:

$$L_o(\mathbf{x}) = \int_{\Omega} L_i(\mathbf{x}, \omega_i)\mathrm{a}(\omega_i \cdot \mathbf{n})\mathrm{d}\omega_i. \tag{4}$$

where a is the basic color from the base appearance MLP. Specifically, we represent illumination in the form of an environment map (introduced later) and ignore indirect reflections between objects. Therefore the integral of Eq. (4) can be represented as the sum of all environment light sources:

$$L_o(\mathbf{x}) = \sum_{\omega_i} L_i(\mathbf{x}, \omega_i)\mathrm{a}(\omega_i \cdot \mathbf{n})\Delta\omega_i. \tag{5}$$

A.3 The Environment Map

We adopt a an HDR environment map [1] (with the form of latitude-longitude) for external lighting modeling. Specifically, we follow the setting of NeRFactor [19] and use a 16×32 resolution for the HDR map. Intuitively, the environment map can be regarded as 512 point light sources with fixed positions.

B Model Architecture

Geometry / base appearance MLP. The geometry and base appearance MLPs consist of 8 hidden layers with the hidden size of 256, where we use SoftPlus for the former and ReLU activation for the latter, and add skip connections to the forth layer as suggested by Wang *et al.* [15]. To increase the representing resolution of the network, we apply positional encoding to the query point position $\mathbf{x}$ (with 10 frequencies). Note that different from the color branch in previous works [11, 15], our base appearance MLP takes the same input as the geometry MLP, *i.e.*, only the query point position $\mathbf{x}$ without the viewing direction $\mathbf{v}$, because we adopt the Lambertian reflectance assumption [7, 17, 18] to ease the representation and factorized learning of outdoor scenes.

Shadow MLP. As shown in Fig. 2 from the main paper, our shadow MLP takes the query position $\mathbf{x}$ (with the positional encoding of 6 frequencies) and a 64-dimensional per-frame latent shadow code $\boldsymbol{l}_s$ as input, and then outputs the scalar shadow value s at $\mathbf{x}$. Practically, we build the shadow MLP of 4 hidden layers with the hidden size of 256, which is shallower than the geometry/base appearance MLPs.

Neural sky generator. The neural sky generator models the varying sky appearance as a spherical dome at infinity, *i.e.*, taking the viewing direction $\mathbf{v}$ and a 64-dimensional per-frame latent environment code $\boldsymbol{l}_e$ as input and directly output RGB colors, which is inspired from GANCraft [4] and also follows the concept of "Sky Box" in the standard rendering pipeline. In practice, the sky generator is constructed as an MLP of 3 hidden layers with the hidden size of 256, and the network input $\mathbf{v}$ is also elevated with positional encoding (4 frequencies).

Affine tone mapper. The affine tone mapper is introduced to compensate for the limitation of data-driven HDR prior that struggles to handle sensor variations (*e.g.*, white balancing, auto-exposure), as explained in Sec. 3.1 of the main paper. Instead of optimizing the explicit affine matrix, motivated by Rematas *et al.* [12], we learn an MLP based mapper that converts a 64-dimensional per-frame latent tone code into a 3×4 affine matrix, which empirically ensures a smoother optimization space for the training. To avoid the overfitting, we set the MLP with 2 hidden layers and the hidden size of 64.

C Implementation Details

C.1 Training Settings

As introduced in Sec. 3.2 of the main paper, we use a composited training scheme to learn factorized scene re-rendering from cluttered datasets, which includes a geometry learning stage and a scene re-rendering learning stage. For both two stages, we use the same setting with a batch size of 512 rays, 64 coarse and 64 fine sampling points along each ray. We employ the Adam optimizer with an initial learning rate of 0.0005 and a cosine annealing scheduler with 5000 warm-up steps. The training process takes about 20 hours for each scene on a single Nvidia RTX3090-24G GPU, with 90k iterations for the first stage and 40k iterations for the second stage.

C.2 Extrapolated 3D Photo Generation

The processing of extrapolated 3D photo generation can be divided into the background rendering (*e.g.*, scene buildings) step, foreground warping (*e.g.*, tourists) step, and blending step. During the background rendering step, we first obtain the camera pose of the given photo, which is directly accessible for our testing dataset and can be also estimated for unseen photos with mature visual localization pipelines [13]. Then, we adapt the lighting conditions of the model by optimizing latent codes to the given photo with masked background areas (see Sec. 3.3 in the main paper). Finally, we synthesize swaying poses based on the current camera pose and enlarge the field of view (FoV) to obtain an extrapolated background view of the photo. During the foreground warping step, we aim to sway the selected tourists to make their motion consistent with the global swaying poses. Instead of recovering human body geometries for human warping [14], we simply treat the front tourist lying at a vertical plane in front of the camera, and thus we can compute a homography warping matrix with the same swaying pose [5] to make the foreground move together. In practice, during the computation of homography, we assume that the target persons are standing in front of the camera at a reasonable distance, which is adjustable by users. Besides, to facilitate foreground and background blending, we also warp the foreground human masks with the same homography. In the final blending step, we feather the warped foreground human mask, and then apply the linear blending to blend the foreground and the background together.

D More Discussions

Limitation for real-world applications. 1) Due to the frequent network queries of volume rendering, the rendering speed of our method is about 4 minutes for each frame (at a resolution of 960×540). In the future, we can accelerate network inference by adopting the latest volume caching techniques [2, 3]. **2)** During the extrapolated 3D photo generation, the swaying foreground persons are directly obtained from the given photo. So we only support photo extrapolation for the background scene and 3D photo generation without downward views, while the uncaptured part of the human bodies are not considered. A possible workaround is to adopt portrait

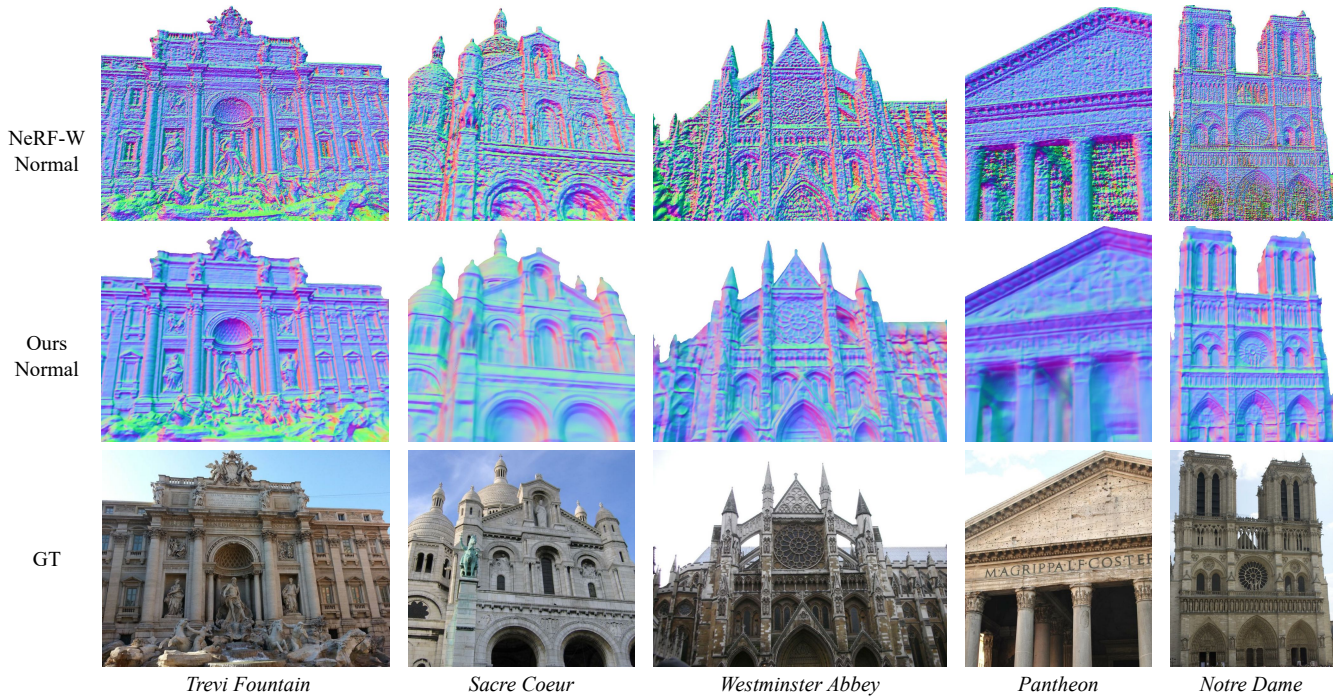

Figure A: We compare the surface normal between our method and NeRF-W (with density gradient w.r.t. query points). Note that our normal is much smoother and more reasonable than the previous NeRF-based method like NeRF-W, which ensures a natural re-rendering with external scene lighting.

image completion techniques [16] to complete the tourist persons at the beginning, which can be integrated into our method and considered as a future work.

Limitation of realism augmentation. Since the network of our realism augmentation is fine-tuned with a single user-captured photo, it works well for familiar patterns where both the user's photo and the rendered image overlap, but might lose generalization if the overlapping area is not sufficient (*e.g.*, the user's photo only contains a small portion of the background). In the case that the user's photo is very narrow, we suggest conducting realism augmentation in a partially fine-tuned way, *i.e.*, optimizing only the decoder of realism augmentation while freezing others, which might be less detailed but can avoid unexpected artifacts caused by overfitting.

E More Experiment Results

Comparison of surface normal. We compare the rendered surface normal between our method and NeRF-W [8]. As shown in Fig. A, our surface normal is much smoother than NeRF-W even trained with the same cluttered dataset (*e.g.*, a smoother dome in *Trevi Fountain* and clearer arch doors in *Notre Dame*). As a smooth normal is proven to be essential for re-rendering with given external lighting [19], we believe such SDF-based geometry representation is more suitable for outdoor scene re-rendering with controllable lighting conditions.

Data-driven HDR decoder vs. explicit HDR parameters or MLP decoder from stratch. One alternative design of learning sky HDR map is to directly optimize explicit HDR parameters (*i.e.*,

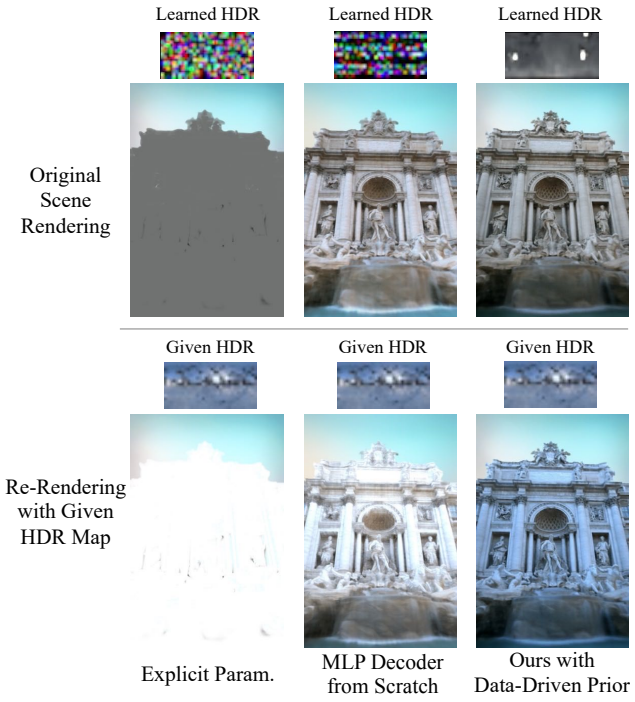

Figure B: We compare our data-driven HDR decoder with two alternative designs, *i.e.*, explicit HDR parameters and learnable HDR decoder trained from scratch.

optimize a 16×32×3 matrices per frame) [19], or train an MLP-based HDR decoder from scratch. We thus make a comparison between our HDR decoder with data-driven prior and these two alternative designs in the second re-rendering training stage as introduced in Sec. 3.2, and visualize the learned HDR maps and the controllable re-rendering results in Fig. B. It is clear that when training re-rendering without any prior knowledge, the factorized re-rendering model cannot learn a plausible HDR map (*e.g.*, cluttered HDR maps in the top row), which also affects the scene rendering and controllable re-rendering with customized HDR maps (*e.g.*, unconverged or blurry rendering results at the first two column in Fig. B). Thanks to the data-driven HDR decoder, our learned HDR map is much more plausible and can better support scene re-rendering with user-defined HDR maps.

References

[1] Paul Debevec. 2008. Rendering Synthetic Objects into Real Scenes: Bridging Traditional and Image-based Graphics with Global Illumination and High Dynamic Range Photography. In *Acm siggraph 2008 classes*. 1–10.

[2] Fridovich-Keil and Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. 2022. Plenoxels: Radiance Fields without Neural Networks. (2022).

[3] Stephan J Garbin, Marek Kowalski, Matthew Johnson, Jamie Shotton, and Julien Valentin. 2021. Fastnerf: High-Fidelity Neural Rendering at 200FPS. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 14346–14355.

[4] Zekun Hao, Arun Mallya, Serge Belongie, and Ming-Yu Liu. 2021. GANcraft: Unsupervised 3d Neural Rendering of Minecraft Worlds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 14072–14082.

[5] Richard Hartley and Andrew Zisserman. 2003. *Multiple View Geometry in Computer Vision*. Cambridge university press.

[6] James T Kajiya. 1986. The Rendering Equation. In *Proceedings of the 13th annual conference on Computer graphics and interactive techniques*. 143–150.

[7] Zhengqin Li, Mohammad Shafiei, Ravi Ramamoorthi, Kalyan Sunkavalli, and Manmohan Chandraker. 2020. Inverse Rendering for Complex Indoor Scenes: Shape, Spatially-Varying Lighting and SVBRDF from a Single Image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2475–2484.

[8] Ricardo Martin-Brualla, Noha Radwan, Mehdi SM Sajjadi, Jonathan T Barron, Alexey Dosovitskiy, and Daniel Duckworth. 2021. NeRF in the Wild: Neural Radiance Fields for Unconstrained Photo Collections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 7210–7219.

[9] Nelson Max. 1995. Optical Models for Direct Volume Rendering. *IEEE Transactions on Visualization and Computer Graphics* 1, 2 (1995), 99–108.

[10] Moustafa Meshry, Dan B Goldman, Sameh Khamis, Hugues Hoppe, Rohit Pandey, Noah Snavely, and Ricardo Martin-Brualla. 2019. Neural Rerendering in the Wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 6878–6887.

[11] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. 2020. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In *European conference on computer vision*. Springer, 405–421.

[12] Konstantinos Rematas, Andrew Liu, Pratul P. Srinivasan, Jonathan T. Barron, Andrea Tagliasacchi, Tom Funkhouser, and Vittorio Ferrari. 2022. Urban Radiance Fields. *CVPR* (2022).

[13] Johannes L. Schönberger and Jan-Michael Frahm. 2016. Structure-from-Motion Revisited. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, 4104–4113.

[14] Meng-Li Shih, Shih-Yang Su, Johannes Kopf, and Jia-Bin Huang. 2020. 3D Photography Using Context-Aware Layered Depth Inpainting. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[15] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. 2021. NeuS: Learning Neural Implicit Surfaces by Volume Rendering for Multi-view Reconstruction. *NeurIPS* (2021).

[16] Xian Wu, Rui-Long Li, Fang-Lue Zhang, Jian-Cheng Liu, Jue Wang, Ariel Shamir, and Shi-Min Hu. 2019. Deep Portrait Image Completion and Extrapolation. *IEEE Transactions on Image Processing* 29 (2019), 2344–2355.

[17] Ye Yu, Abhimitra Meka, Mohamed Elgharib, Hans-Peter Seidel, Christian Theobalt, and William AP Smith. 2020. Self-Supervised Outdoor Scene Relighting. In *European Conference on Computer Vision*. Springer, 84–101.

[18] Ye Yu and William Alfred Peter Smith. 2021. Outdoor Inverse Rendering from a Single Image Using Multiview Self-Supervision. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021).

[19] Xiuming Zhang, Pratul P Srinivasan, Boyang Deng, Paul Debevec, William T Freeman, and Jonathan T Barron. 2021. NeRFactor: Neural Factorization of Shape and Reflectance Under an Unknown Illumination. *ACM Transactions on Graphics (TOG)* 40, 6 (2021), 1–18.