Multiplane Prior Guided Few-Shot Aerial Scene Rendering

Supplementary Material

In this supplement, we first conduct more experimental results and discussion to evaluate the robustness and efficiency of our proposed Multiplane Prior guided NeRF (MPNeRF). We also include more qualitative results to discuss the motivation and limitations of MPNeRF. Finally, we add more details of experimental settings and implementations.

1. Additional Experiments and Analysis

Robustness to Hyperparameters. We have conducted a series of experiments to assess the sensitivity of our model to hyperparameters. Specifically, we focus on the hyperparameter λ, which plays a crucial role in balancing different components of our loss function. In Figure 1, we illustrate the impact of varying λ on the performance of the proposed MPNeRF and a standard NeRF [12] model.

As λ increases, we observe that the PSNR and SSIM metrics tend to plateau, suggesting that there is an optimal range for λ wherein the model achieves a balance between fidelity and perceptual quality. On the other hand, the LPIPS metric shows an initial decrease followed by a gradual increase, indicating a sweet spot where the model best captures the perceptual features of the aerial scenes. The trends exhibited by MPNeRF show its relative insensitivity to λ within a reasonable range, which underscores the robustness of our method. Notably, MPNeRF consistently outperforms the baseline NeRF model across all metrics, demonstrating the effectiveness of incorporating the multiplane prior to the rendering process.

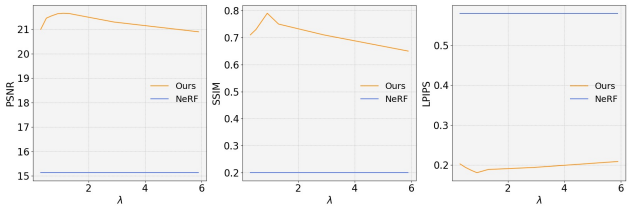

Figure 1. **Hyperparameter Sensitivity Analysis.** Performance comparison of our method (MPNeRF) and a baseline NeRF model across different values of hyperparameter λ. The graphs show PSNR, SSIM, and LPIPS metrics. MPNeRF is robust to a wide λ choice.

MPNeRF vs MPI-based Methods. The present study of MPI mainly focuses on overcoming shortcomings such as "failure to represent continuous 3D space" in [9]. In contrast, our approach utilizes MPI as a bridge to convey complex information that a single NeRF struggles with. We construct comparison experiments under 3-view settings among the proposed MPNeRF, the original MPI [15], and MINE [9] In Table. 1, the original MPI achieves an 18.57 PSNR, 0.54 SSIM, and 0.45 LPIPS. While MINE performs better with 19.99 PSNR, 0.61 SSIM, and 0.40 LPIPS. Our MPNeRF outperforms these methods by a large margin. These MPI-based methods face inherent limitations like ghosting effects and cropped corners under sparse inputs and large camera movements.

Methods	PSNR	SSIM	LPIPS
MPI [15]	18.57	0.54	0.45
MINE [9]	19.99	0.61	0.40
NeRF branch w/o $\mathcal{L}_{mul}$	15.13	0.20	0.58
MPI branch	20.32	0.57	0.34
NeRF branch w/t $\mathcal{L}_{mul}$	21.72	0.80	0.19

Table 1. **Comparation between the NeRF and MPI.** This table presents the evaluation of the MPI based method in previous studies, NeRF branch without multiplane loss ($\mathcal{L}_{mul}$), the MPI branch independently, and the NeRF branch with $\mathcal{L}_{mul}$ within our MPNeRF framework. The metrics of PSNR, SSIM, and LPIPS demonstrate the significant impact of the multiplane prior on the rendering performance in sparse aerial scenes.

NeRF Branch vs MPI Branch. In Table. 1, we examine the performance impact of the NeRF and MPI branches within our proposed MPNeRF. Initially, the NeRF branch without the multiplane loss $\mathcal{L}_{mul}$ (equals to a plain NeRF model) demonstrates a PSNR of 15.13, an SSIM of 0.20, and an LPIPS of 0.58. These values indicate a baseline level of performance where the NeRF branch struggles with sparse aerial views, as evidenced by the low PSNR and SSIM scores, along with a high LPIPS value which suggests a significant perceptual difference from the ground truth. In contrast, the MPI branch alone shows better across all metrics, with a PSNR of 20.32, an SSIM of 0.57, and a reduced LPIPS of 0.34. The MPI branch's improved performance is likely due to its discrete depth-based representation that aligns better with the structured nature of aerial scenes, thus capturing the scene geometry more effectively. And the inductive bias of CNN and Transformer makes MPI generalize better. The most significant performance gains are observed when the NeRF branch is combined with the multiplane loss $\mathcal{L}_{mul}$, resulting in a PSNR of 21.72, an SSIM of 0.80, and an LPIPS of 0.19. The addition of $\mathcal{L}_{mul}$ to the NeRF branch enhances its ability to recover details from sparse views, as reflected by the substantial improvements in PSNR, SSIM, and LPIPS. The proposed Multiplane Prior serves as a bridge to convey

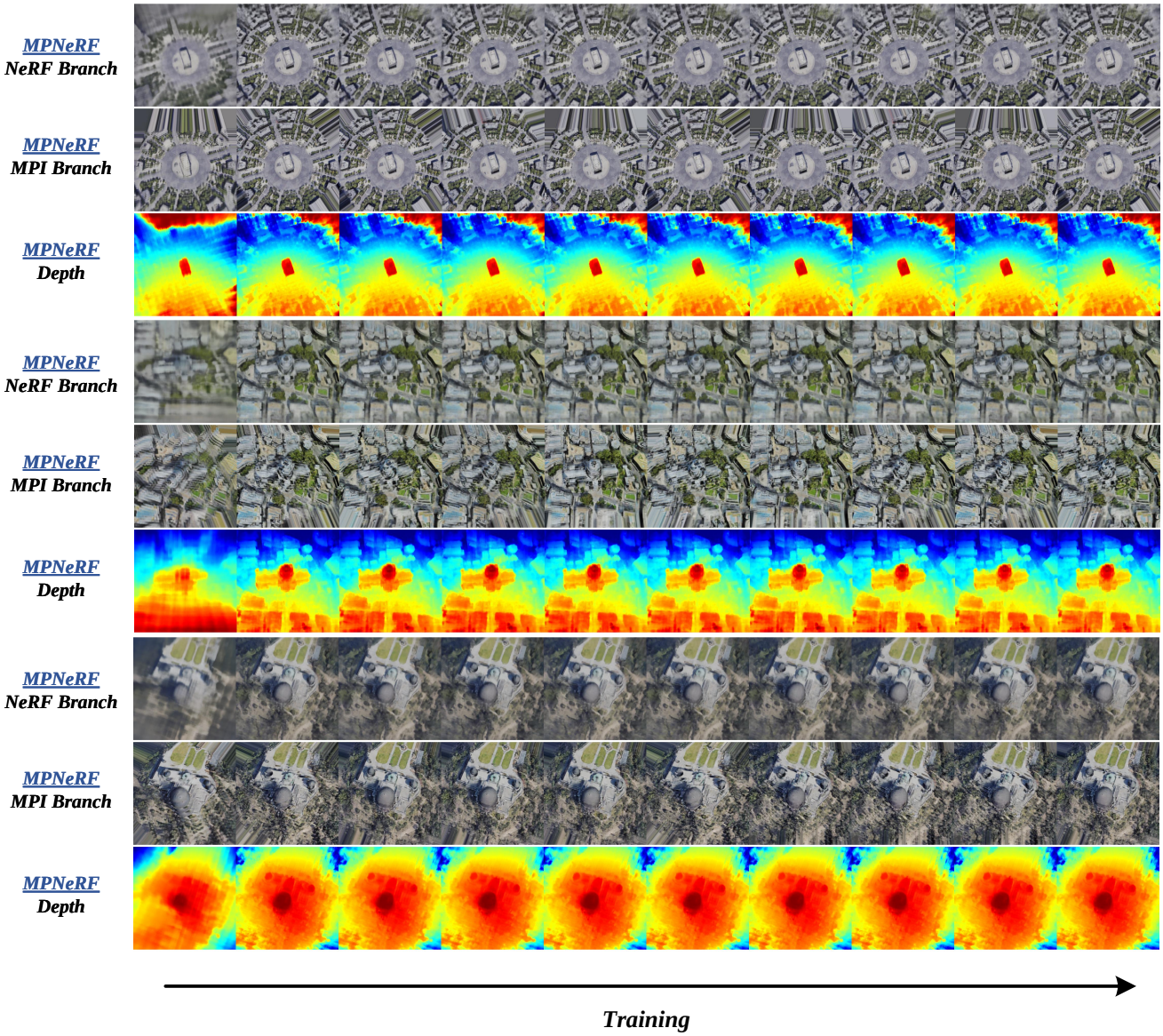

Figure 2. **Training Progression of MPNeRF.** The sequence shows comparative results from the NeRF and MPI branches at various training stages under 3 view settings. Left to right: early, mid, and late phases of training. The NeRF branch initially shows noisier reconstructions with indistinct depth estimations, while the MPI branch exhibits crop edge and overlapping ghosting effects. Over time, the NeRF branch, guided by the MPI-derived multiplane prior, progressively captures finer details and more accurate depth information, as reflected in the sharpening of depth map visualizations.

information that is hard to learn by the traditional NeRF pipeline. These results underscore the efficacy of incorporating multiplane priors into the NeRF framework for few-shot aerial scene rendering.

Other Few-shot NeRF Methods Combined with Multiplane Prior. It stands to reason that it's worth evaluating other Few-shot NeRF methods combined with multiplane prior. We incorporated FreeNeRF's [19] frequency regularization and evaluated it under a 3-view setting. This integration results in a marginal increase in the PSNR by 0.2db. We believe the MPI's noisy predictions help reduce early training overfitting in high-frequency details. This mechanism seems to parallel the underlying concept of FreeNeRF, potentially explaining the minimal improvement.

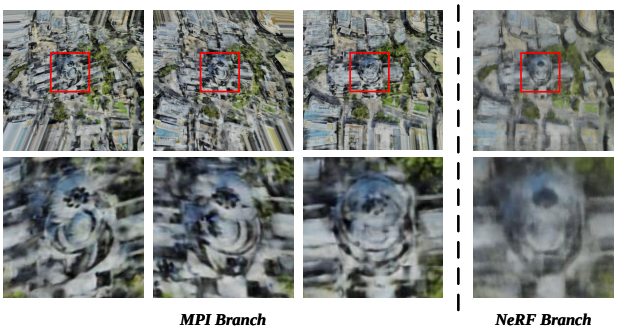

Figure 3. **Detail Comparison between MPI and NeRF Branch Outputs.** The images on the left column represent the MPI branch's output, displaying sharper details with overlapping ghosting effects in the highlighted regions. In contrast, the right column shows the NeRF branch's output, where the same regions appear more blurred.

2. Discussion and Future Works

Why MPNeRF Works? Despite the advantage of the MPI representation in aerial scenes, a simple question is: Why MPNeRF is kept away from the cropped edge and overlapping ghosting effect of the MPI? Avoiding the cropped edge is simple, we sample rays from unseen views following the mask generated during homography warping. To better illustrate why the overlapping ghosting effect can not be learned by NeRF, we visualize the same target view rendered by the MPI branch in Figure. 3. With the source viewpoint varying, the overlapping ghosting effect in the rendered target view differs. Since MPI derived from different viewpoints does not share a common world space, these overlapping ghosting effects are not multi-view consistent across all views. Thus these effects violate the multi-view consistency assumption of NeRF [12]. With these noises provided as pseudo-supervision, the MLP optimized with gradient descent tends to give blurry rendering. The blurring signifies NeRF's attempt to average out the incongruities across views. These pseudo-labels, while derived from an informed place, act as imperfect guides, introducing a trade-off that MPNeRF must navigate. On one hand, they provide a rich, albeit noisy, signal that captures the complexity of aerial scenes. On the other, they present a risk of polluting the training process with artifacts.

Although MPNeRF shows that a simple MSE loss can perform well, this delicate balance highlights the importance of a carefully crafted training regimen, one that can differentiate between useful signals and misleading noise. Our future work will delve into refining this balance, potentially through the development of more sophisticated noise-filtering mechanisms or through the implementation of more robust training strategies that can better leverage the nuanced information within these pseudo-labels. In doing so, we may further enhance MPNeRF's rendering quality, pushing the boundaries of few-shot aerial scene rendering.

Semantic Integration for Improved Scene Understanding. Integrating semantic segmentation into the MPNeRF framework offers an exciting direction for enhancing scene understanding. By associating semantic labels with the MPI branch, MPNeRF may provide more contextually aware reconstructions and pave the way for applications in urban planning and navigation under limited data.

Scene Editing. An exciting avenue for future research is the possibility of editing NeRF-rendered scenes by directly manipulating the MPIs generated by the MPI branch. This could enable users to alter scene characteristics such as color, texture, or even geometric structure, through an intuitive interface. A potential direction is utilizing differentiable rendering techniques to backpropagate the desired edits from the scene rendering back to the MPI and NeRF representations.

Scalability. Currently, scalability remains a potential limitation when our MPNeRF model is applied to larger scenes. The primary bottleneck arises from the inherent capacity constraints of NeRF models. They are typically optimized for smaller, more controlled environments and can struggle to maintain fidelity at the increased scale and complexity of larger scenes. As scenes expand in size, the NeRF's neural network requires a corresponding increase in capacity to model the additional detail, which can lead to a significant escalation in computational and memory requirements [14, 18]. Furthermore, the encoder-decoder architecture employed within our MPI branch is not ideally suited for high-resolution imagery [1, 3]. It tends to consume substantial amounts of memory, especially when processing the finer details necessary for large-scale scene rendering. The memory footprint grows rapidly with the resolution of input images due to the quadratic increase in the number of pixels that need to be processed simultaneously.

3. Additional Visualizations.

Training Progression of MPNeRF. Figure. 2 presents a detailed visual account of the training evolution within our MPNeRF, delineating the comparative outcomes from the NeRF and MPI branches across three distinct training phases. The left columns illustrate the initial stage where the NeRF branch outputs are notably noisier, and the depth maps lack precise definition, signifying the model's initial struggle to interpret the sparse aerial views. These preliminary results are characterized by a lack of clarity and detail, with the depth maps displaying broad, undifferentiated regions of low confidence. As training progresses to the midpoint, displayed in the center columns, the MPI branch starts to assert its strengths. It delivers reconstructions with improved clarity and begins to better capture the geometric intricacies of the aerial scenes. This enhancement is evident in the depth maps,

where we observe a transition from broad, undefined areas to more distinct regions of depth estimation, indicative of the MPI branch's capability to delineate structural features more effectively at this stage. Reaching the later stages of training, shown in the right columns, the NeRF branch, now informed by the multiplane prior, shows significant advancement. It starts to match and, in certain aspects, surpasses the MPI branch's performance by delivering images with greater detail fidelity. This is most apparent in the depth maps, where the once diffused and expansive high-confidence regions have now evolved into sharply defined areas, highlighting the network's improved proficiency in depth perception.

The visualization of the depth maps is particularly telling; the sharpening of these maps directly correlates with the improved model's depth estimations. The NeRF branch, leveraging the multiplane prior, demonstrates an enhanced ability to resolve the complex spatial relationships inherent in aerial scenes, moving beyond the initial limitations evidenced in the early training outputs.

This sequential improvement underscores the efficacy of the MPNeRF training process, which effectively leverages the distinct advantages of both NeRF and MPI branches to progressively refine the model's understanding of the scene, culminating in high-quality renderings from sparse inputs. The journey from noisy, indistinct initial attempts to clear, detailed final outputs exemplifies the potent potential of MPNeRF for aerial scene rendering.

Different Layers of the MPI Branch. To explore the geometry and appearance captured by the MPI branch, we visualize the color with transparency computed by the density of different MPI layers. Figure. 4 showcases a series of images that represent different layers of the MPI branch, each corresponding to a specific depth level within the aerial scene, as labeled from 'Shallow' to 'Deep'. The images progress from the topmost layers, which capture high-elevation features like roofs, to the bottom layers, which reveal ground-level details. However, it is evident that the fidelity of the reconstruction varies across depth layers. The initial layers, while capturing the broad layout, lack the finer details and the sharpness present in the ground truth (GT). The middle layers begin to show more structure and texture, indicating an intermediate range where the MPI branch most effectively captures the scene's appearance. The deeper layers, while richer in detail, start to exhibit artifacts, such as blurring and possible misalignments, before converging towards the ground truth. This suggests that while the MPI branch of MPNeRF shows promise in reconstructing aerial scenes from limited data, it is still highly inaccurate and contains artifacts.

4. More Implementation Details.

Datasets and Metrics. Our evaluation is conducted on a dataset that presents a rich tapestry of aerial landscapes, the LEVIR-NVS [16], comprising 16 diverse scenes that span mountains, urban centers, villages, and standalone architectural structures. Each scene in the dataset is represented by a collection of 21 multi-view images, each with a resolution of 512×512 pixels. This selection ensures a broad representation of scenarios that MPNeRF might encounter in real-world applications. The LEVIR-NVS dataset encapsulates a variety of pose transformations that mimic the dynamic nature of UAV flight patterns, including wrapping and swinging motions. These pose variations introduce realistic challenges in aerial photography, such as changes in viewpoint and scale, making the dataset a rigorous testing ground for our model. The inclusion of these complex transformations in the simulation process is crucial for assessing the robustness of MPNeRF's performance in conditions that closely approximate actual aerial image capture.

In our experimental setup, we strategically select specific views for training to assess the capability of our model in both interpolation and extrapolation scenarios. For the three-view setting, we utilize view IDs: 0, 7, and 15. This selection is designed to provide a spread of perspectives that challenges the model to extrapolate the scene effectively. In the five-view setting, we expand our selection to include view IDs: 0, 7, 10, 15, and 20. This broader range tests the model's interpolation skills and its ability to extrapolate scenes from more diverse viewpoints.

In our experiments, we employ three standard metrics. Peak Signal-to-Noise Ratio (PSNR) is used to measure the image reconstruction quality, calculated as the negative logarithm of the mean squared error between the predicted and ground truth images. Structural Similarity Index Measure (SSIM), obtained via the skimage[1] library, assesses image quality based on luminance, contrast, and structural information. Learned Perceptual Image Patch Similarity (LPIPS), computed using a VGG-based model from the lpips[2] package, evaluates perceptual similarity, reflecting more human-centric assessments of image quality.

Implementation of Baseline Methods. We implement the baseline methods following their open-source code base. We adopt 64 coarse sampling and 32 fine sampling for the NeRF backbone of these methods. In particular, the RegNeRF [13] and FreeNeRF [19] are implemented based on Mip-NeRF [2], and others [6, 7, 12, 20] are based on a vanilla NeRF. All methods are trained for 30 epochs for each scene and the hyperparameters are strictly consistent across all experiments.

Implementation of MPNeRF. We implement MPNeRF based on the nerf-pl codebase [3], which provides a PyTorch Lightning framework for efficiently operationalizing NeRF architectures. The settings of hyperparameters are strictly consistent with baseline methods. Our NeRF branch adheres closely to the original NeRF paper specifications, ensuring

[1] https://scikit-image.org/
[2] https://github.com/richzhang/PerceptualSimilarity
[3] https://github.com/kwea123/nerfpl

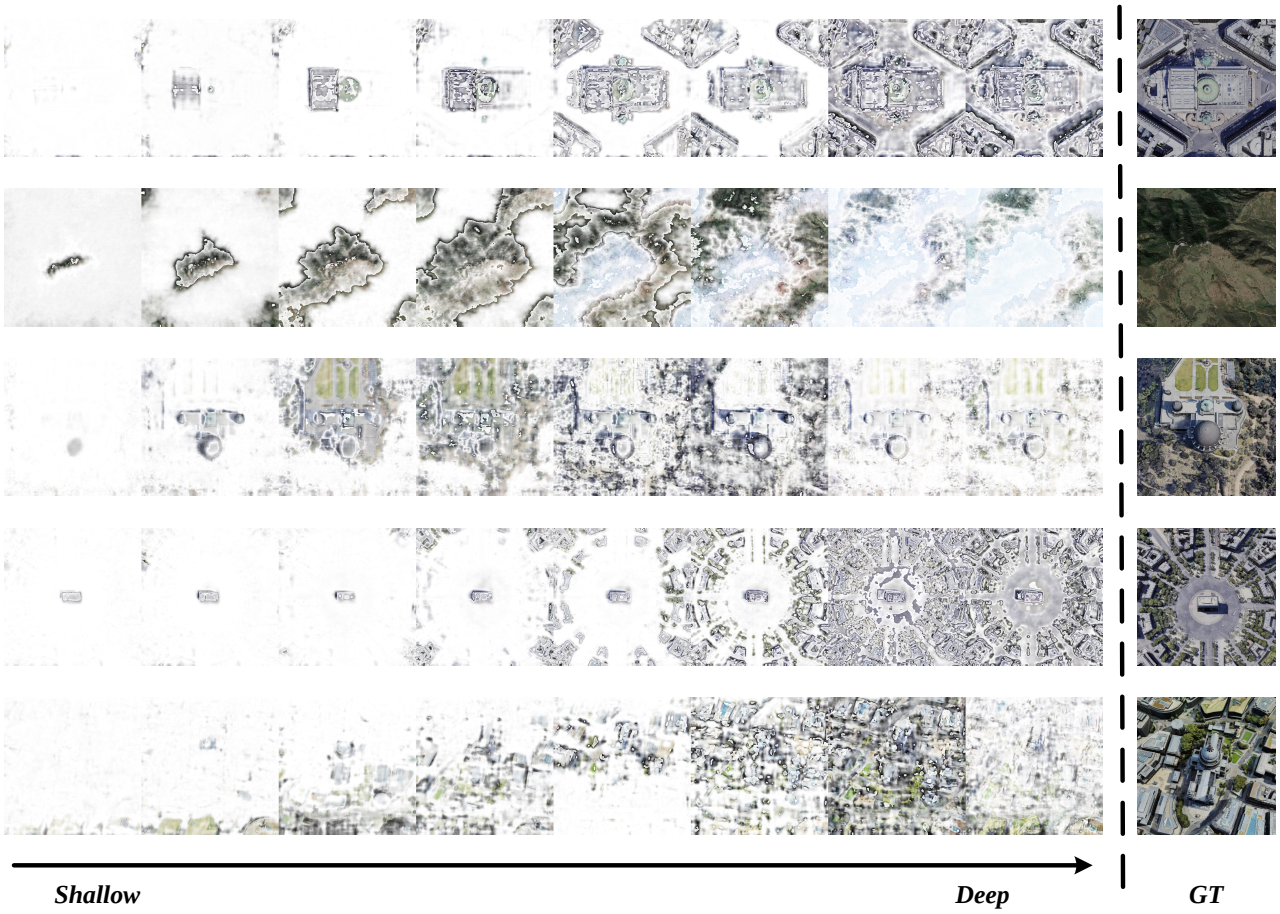

Figure 4. **Visualization of MPI Branch Depth Layers.** Sequential depth layers from the MPI branch reveal the aerial scene's structure, evolving from translucent to opaque as we move from shallow to deep layers, culminating in the ground truth (GT) image for reference.

a faithful reproduction of the baseline model. We adopt 64 coarse sampling and 32 fine sampling for the NeRF branch. Inspired by previous works [9, 15, 16], our MPI branch is constructed following an encoder-decoder architecture MPI generator. The encoder is a strand SwinV2 Transformer [10] pretrained via SimMIM [17]. The encoder is kept frozen during training. A detailed description of our decoder architecture is presented in Table. 2. The MPI generator embeds depth hypotheses into the input features, which are then processed through convolutional layers to output MPIs with RGB and density values, leveraging skip connections and multi-scale representations for detail enhancement.

For optimization, we utilize the Adam optimizer [8] with a learning rate of 5×10^{-4}, and a cosine learning rate decay scheduler [11]. Our model is trained on a single NVIDIA RTX 3090 GPU for 30 epochs, taking about 2.5 hours to converge. The batch size is set to 1024 rays per iteration for both seen and unseen views, allowing sufficient diversity of data points for gradient estimation while maintaining manageable memory requirements.

References

[1] André Araujo, Wade Norris, and Jack Sim. Computing receptive fields of convolutional neural networks. *Distill*, 4(11): e21, 2019. 3

[2] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5855–5864, 2021. 4

[3] Wuyang Chen, Ziyu Jiang, Zhangyang Wang, Kexin Cui, and Xiaoning Qian. Collaborative global-local networks for memory-efficient segmentation of ultra-high resolution images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8924–8933, 2019. 3

[4] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*, 2015. 6

Layer	Kernel Size	In-Channels	Out-Channels	Input	Activation
convdown1	1	768	512	encoder_layer4	ELU [4]
convdown2	3	512	256	convdown1	ELU
convup1_extra	3	256	256	convdown2	ELU
convup2_extra	1	256	768	convup1_extra	ELU
convup5	3	768 + 21	256	cat(convup2_extra, depth_embedding)	ELU
conv5	3	256 + 768 + 21	256	cat(convup5, encoder_layer3, depth_embedding)	ELU
convup4	3	256	128	conv5	ELU
conv4	3	128 + 384 + 21	128	cat(convup4, encoder_layer2, depth_embedding)	ELU
output4	3	128	4	conv4	Sigmoid (for RGB) and abs (for σ)
convup3	3	128	64	conv4	ELU
conv3	3	64 + 192 + 21	64	cat(convup3, encoder_layer1, depth_embedding)	ELU
output3	3	64	4	conv3	Sigmoid (for RGB) and abs (for σ)
convup2	3	64	32	conv3	ELU
conv2	3	32 + 96 + 21	32	cat(convup2, encoder_conv1, depth_embedding)	ELU
output2	3	32	4	conv2	Sigmoid (for RGB) and abs (for σ)
convup1	3	32	16	conv2	ELU
conv1	3	16	16	convup1	ELU
output1	3	16	4	conv1	Sigmoid (for RGB) and abs (for σ)

Table 2. **Decoder Architecture for the MPI Branch.** Each convup layer within our architecture is composed of a convolution layer, followed by batch normalization and the specified activation layer, as delineated in the table. This sequence is then succeeded by a $2\times$ nearest neighbor upsampling process. Conversely, the convdown blocks are structured beginning with a max pooling layer with a stride of 2, followed by a convolution layer, and culminating with an activation layer. This architecture choice follows previous research in MPI representations and depth estimation [5, 9, 16].

[5] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J Brostow. Digging into self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3828–3838, 2019. 6

[6] Ajay Jain, Matthew Tancik, and Pieter Abbeel. Putting nerf on a diet: Semantically consistent few-shot view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5885–5894, 2021. 4

[7] Mijeong Kim, Seonguk Seo, and Bohyung Han. Infonerf: Ray entropy minimization for few-shot neural volume rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12912–12921, 2022. 4

[8] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[9] Jiaxin Li, Zijian Feng, Qi She, Henghui Ding, Changhu Wang, and Gim Hee Lee. Mine: Towards continuous depth mpi with nerf for novel view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12578–12588, 2021. 1, 5, 6

[10] Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, et al. Swin transformer v2: Scaling up capacity and resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12009–12019, 2022. 5

[11] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 5

[12] 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 *ECCV*, 2020. 1, 3, 4

[13] Michael Niemeyer, Jonathan T Barron, Ben Mildenhall, Mehdi SM Sajjadi, Andreas Geiger, and Noha Radwan. Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5480–5490, 2022. 4

[14] Matthew Tancik, Vincent Casser, Xinchen Yan, Sabeek Pradhan, Ben Mildenhall, Pratul P Srinivasan, Jonathan T Barron, and Henrik Kretzschmar. Block-NeRF: Scalable large scene neural view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8248–8258, 2022. 3

[15] Richard Tucker and Noah Snavely. Single-view view synthesis with multiplane images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 551–560, 2020. 1, 5

[16] Yongchang Wu, Zhengxia Zou, and Zhenwei Shi. Remote sensing novel view synthesis with implicit multiplane representations. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–13, 2022. 4, 5, 6

[17] Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhuliang Yao, Qi Dai, and Han Hu. Simmim: A simple framework for masked image modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9653–9663, 2022. 5

[18] Linning Xu, Yuanbo Xiangli, Sida Peng, Xingang Pan, Nanxuan Zhao, Christian Theobalt, Bo Dai, and Dahua Lin. Grid-guided neural radiance fields for large urban scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8296–8306, 2023. 3

[19] Jiawei Yang, Marco Pavone, and Yue Wang. Freenerf: Improving few-shot neural rendering with free frequency regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8254–8263, 2023. 2, 4

[20] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4578–4587, 2021. 4