NeLF-Pro: Neural Light Field Probes

We thank the reviewers for their encouraging comments. We are glad to see that the reviewers generally appreciate the methodological novelty and our result quality, and think the paper is "elegant" (R1), a "clear improvement over prior methods", "the main ideas seem novel and interesting" (R2), and the "scene presentation is elegant and effective" (R3). We now respond to the reviewers' comments.

R1 vG4L

Modeling blocked rays: Unlike standard light field methods that model rays using global parametrization, our light field probes capture visible rays in local areas and render novel rays by leveraging volumetric integration. Therefore it achieves occlusion-aware modeling and rendering. We will more explicitly expose this strength in our revision.
Model size and rendering time: We report our model size and training time in Table 1 and Table 3 of the main paper. While our method is implemented on PyTorch without customized CUDA kernels, it achieves rendering speeds competitive with previous grid-based methods: Rendering a 960×540 image requires 2.6 seconds. We will emphasize this.
Light field References: Thanks. We will add them.

R2 xRjV

Light field probe is layered? Each probe is represented by a single layer that stores multi-channel features.
How do probes contribute to the ray sample? The contributions of the selected probes are not explicitly modeled but are updated through gradient descent. Hence, this acts like a learned "voting" process guided by photometric loss.
Aggregation module: Our method represents a point inside the volume as the element product of vectors queried from the three factors of the local light field probes (i.e., $\ddot{\mathbf{V}}, \ddot{\mathbf{M}}, \mathbf{M}$), and the aggregation module is proposed to fuse the *local vectors* (e.g., $\ddot{\mathbf{V}}_i$) using a self-attention mechanism. We will make the factor aggregation more clear.
Projection network: Thanks for the suggestion. We agree that renaming it as "decoder" would indeed be clearer. We will make this change in the revision.
Information flow: We derive the final formula (Eqn. 10) from a discrete factorization (Eqn. 5), then continuous factorization (Eqn. 6), local transformation (Eqn. 9), and local aggregation (Eqn. 10). We will improve the description.
Equations 5 and 6: Eqn. 5 describes a discrete light field probe factorization, whereas Eqn. 6 describes a continuous factorization performed on the spatial position $\mathbf{x}$.
Equations 6 and 9: Compared to Eqn. 6, Eqn. 9 incorporates coordinate transformation to transform the global coordinate $\mathbf{x}$ into local factor coordinates $\ddot{t}_c, \ddot{\mathbf{v}}_c, \mathbf{v}_l$.
Mode: The term "mode" is commonly used to specify a particular dimension of a tensor.

Scene	PSNR↑	SSIM↑	Time (Hours) ↓	Model Size (GB) ↓
56Leonard	28.86 / **31.81**	0.830 / **0.957**	1.6 / 3.0	0.40 / 1.60
Scuol	**27.84** / 21.83	**0.855** / 0.818	1.6 / 3.0	0.40 / 1.20
KITTI-Big	**22.68** / 13.50	**0.720** / 0.510	1.6 / 1.0	0.40 / 0.35

Table 1. **Quantitative comparison between ours and 3DGS.** Metrics are denotes as ours / 3DGS.

Figure 1. **Qualitative Results of 3DGS on Large-Scale Scenes.** We show the same views as in Figure 5 of the supplementary PDF.

How tensor representation is interpreted as spherical? We transform the world coordinates to tensorial coordinates through a spherical projection (i.e., Eqn. 8 and Eqn. 9). The coordinate remapping allows for spherical modeling, akin to panoramic images.
Simplify hyper parameter settings details and expand ablation: Great idea. We will expand as suggested.
Typos: We will fix the typo and improve the sentence. Thanks.
3DGS: For this rebuttal, we further evaluate 3DGS on the large-scale scenes, as shown in Table 1 and Fig. 1. 3DGS produces a blurry background and floor in the Scuol scene, and it fails to densify the Gaussians for the KITTI360-large scene due to the extreme sparsity of initialization points and training views. We believe a better densification strategy and regularization terms are critical for its successful application to large-scale scenes.
The caption of Table 1: We will add "the scores of the baseline methods are taken from F^2-NeRF". Thanks.
Compare with BlockNeRF: Implementing and comparing with BlockNeRF within the short rebuttal period is a challenge due to the absence of publicly available code. In the revision phase, we will reach out to the authors, proposing to send them our datasets for a comparison.

R3 eze5

Compare with BlockNeRF: We kindly refer to R2 above.
Tiling effects and artifacts? Great point. Our method achieves cleaner reconstruction with fewer tiling effects and "floating" artifacts by employing the VMM factorization, which introduces low-rank regularization into the scene representation and optimization process. We cordially invite the reviewer to examine our supplementary video for further details.