GSNeRF: Generalizable Semantic Neural Radiance Fields with Enhanced 3D Scene Understanding

Supplementary Material

A. Additional implementation Details

A.1. Self-Supervised Depth Loss

Our method generalizes well to unseen scenes, either with GT depth for supervision or with self-supervised depth regularization. As detailed in Section 4.2 of our main paper, we ensure the accuracy of $D_{1:K}$ even when GT depths are unavailable through applying a self-supervised loss function $\mathcal{L}_{ssl}$ [1, 3, 7] to regularize our depth predictions. By focusing on the cross-view depth consistency among all source views, we are able to regularize our depth predictions effectively. This loss function (Eq. 7 in our main paper) is defined as:

$$\mathcal{L}_{ssl} = \lambda_1 \mathcal{L}_{RC} + \lambda_2 \mathcal{L}_{SSIM} + \lambda_3 \mathcal{L}_{Smooth}, \quad (1)$$

where $\mathcal{L}_{RC}$ calculates the mean square error between a source image I_k and the reconstructed image I'_k obtained by warping other source images using their predicted depth maps for all source images $I_{1:K}$. $\mathcal{L}_{SSIM}$ measures the structural similarity between $I'_{1:K}$ and $I_{1:K}$, and $\mathcal{L}_{Smooth}$ ensures the smoothness of all depth prediction $D_{1:K}$ by penalizing large variations in the depth values between neighboring pixels. Following [1], the hyperparameters are set at $\lambda_1 = 1$, $\lambda_2 = 0.2$, and $\lambda_3 = 0.0067$.

A.2. Target View Depth Estimation

As mentioned in Section 4.2 of our main paper, we estimate target view depth D_T from the depth prediction of each source image $D_{1:K}$, corresponding source view camera poses $\xi_{1:K}$ and the target view camera pose ξ_T. To be more specific, D_T is estimated by projecting the pixels of each depth map into 3D space and then reprojecting them back to the target view. The pseudo-code of the target view depth estimation is described in Algorithm 1.

A.3. Masking Unrelated Features for Depth-Guided Visual Rendering

In Section 4.2.2 of the main paper, we utilize occlusion-aware masks represented by M_n for volume rendering in Eq. 11 and M for semantic rendering in Eq. 12. These masks selectively exclude irrelevant features of points located behind object surfaces. It is worth noting that, when computing the global features $f_{n,0}$ in Eq. 10 and f_0 and in Eq. 14, the mean and variance are also calculated as masked mean and variance using M_n and M, respectively, ensuring the exclusion of unrelated information.

Algorithm 1 Target-View Depth Map Estimation

Input: Depth predictions of each source view $D_{1:K}$, camera pose of each source view $\xi_{1:K}$, target camera pose ξ_T
Data: Image size: (H, W), camera pose of the world coordinate ξ_w

Output: Target view depth estimation D_T

$A \leftarrow$ empty array()
for k = 1, ..., K **do**
  $g \leftarrow$ meshgrid(H, W)
  Project g into the coordinate system defined by ξ_k
  Multiply g by the corresponding depth prediction D_k
  $g \leftarrow$ **Transform**(g, ξ_k, ξ_w)
  Append g to the array A
end for
$A \leftarrow$ **Transform**(A, ξ_w, ξ_T)
Reproject A onto the ξ_T image plane
$Z \leftarrow$ the third element (Z-axis) of points A
$A' \leftarrow$ round the first two elements of A to integer values
$W \leftarrow$ The first two elements of (A' - A)
Weight and normalize Z using weight W
Set the depth of target view D_T to Z based on the index of the first two elements of A'
return Estimated depth of target view D_T

/* *Function* */
Transform(point, ξ_1, ξ_2):
return transform point from coordinate ξ_1 to ξ_2

A.4. Training Strategy for Depth-Guided Volume Rendering

During the volume rendering process in Section 4.2, we sample points along the ray based on the estimated target view depth map D_T. However, in the early phase of our training, such estimation might not be accurate. Therefore, we employ a mix of uniform and depth-guided sampling, using half the points for each for the first 125K training steps and then switching to all depth-guided for the rest 125K steps. This approach stabilizes our volume rendering process and makes sure that our GSNeRF predicts accurate colors for each pixel of the target view image.

A.5. More Training Details

Given a set of multi-view images of a scene, we select training pairs of source view and target view by first randomly selecting a target view, and sampling K nearby yet sparse views as source views, following the setting of S-Ray [11] and NeuRay [12]. We implement our model using PyTorch [13] and train it end-to-end on a single RTX3090Ti GPU with 24G memory. Notably, we did not utilize any pre-trained weights. The batch size of rays is set to 1024 and our model is trained for 250k steps using Adam optimizer [9] with an initial learning rate of 5e-4 decaying to 1e-5.

A.6. Evaluation Metrics

To evaluate the effectiveness of our method, we examine both semantic performance and visual quality through various metrics in Table 2 of our main paper. We measure the semantic capabilities of our approach using the mean Intersection over Union (mIoU), class average pixel accuracy, and total pixel accuracy. These metrics provide a comprehensive evaluation of how accurately our method is able to recognize and delineate semantic objects within the scene. For evaluating the visual fidelity of the synthesized images, we employ peak signal-to-noise ratio (PSNR), the structural similarity index measure (SSIM) [14], and learned perceptual image patch similarity (LPIPS) [16]. These metrics collectively assess the clarity, structural integrity, and perceptual resemblance of the rendered images as compared to the ground truth.

B. Additional Experiments and Analysis

B.1. Analysis of the depth-guided sampling strategy

Sensitivity on the estimated depth guidance While employing target-view depth estimation as sampling guidance, our method is not as sensitive as Neuray [12] + semhead and S-Ray [11] in Table 2, which require GT depth as the input during inference (e.g., PSNR / mIoU: 31.33 / 58.3 (ours) vs. 25.19 / 55.53 (S-Ray)). This robustness holds even when GT depth is not observed during training, where we employ self-supervised depth loss and observe satisfactory results (PSNR/mIoU 31.49/52.21), outperforming S-Ray (25.13/47.69) and other SOTAs. As suggested, we additionally compare our estimated depth with GeoNeRF [16] for depth rendering, confirming the effectiveness and robustness of our method (see results listed in Table A1).

Efficiency of our sampling strategies With target-view depth as sampling guidance, our method converges in 150k steps during training, faster than S-Ray (250k steps). As for the rendering efficiency, we present a comparison with S-Ray in Table A2, both run on the same device of RTX-3090ti and i7-13700k. Notably, when sampling only 4 points along the rays, our method shows 425% improvements in speed while maintaining superior image and semantic segmentation quality.

		accuracy ↑			error ↓	
Method	GT depth	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$	abs rel	rmse
GeoNeRF		31.38	61.49	84.29	0.3160	0.7197
Ours		82.98	95.21	98.24	0.1448	0.3594
GeoNeRF	✔	83.91	97.80	99.74	0.1287	0.3002
Ours	✔	88.77	98.56	99.84	0.1080	0.2503

Table A1. **Quantitative results of our target depth estimation.** We use the metrics defined in [4]. Our method estimates high-quality target view depth maps reliably, whether or not GT depth maps are used as supervision, in contrast to [8].

	N	FPS↑	PSNR↑	mIoU↑
S-Ray	128	0.16	25.13	47.69
Ours	128	0.11	31.49	52.21
Ours	4	0.84	27.80	52.21

Table A2. **Rendering a 320x240 image with segmentation map.** N is the sampling points number along a ray. Our approach achieves superior image quality and segmentation results using just 4 points, while exhibiting four times faster rendering speed.

B.2. Finetuning on Unseen Scenes

To enhance the completeness of our method, we adopt the fine-tuning setting in S-Ray [11]. Specifically, we fine-tune our generalized model for a limited number of steps, 5k steps, on each unseen scene before evaluation.

Table A3 and Fig. A1 show the quantitative and quantitative results of our model finetuning 5k steps on ScanNet [2]. We observe that by adopting our designed Semantic Geo-Reasoning and Depth-Guided Visual Rendering, our method preserved better rendering quality in the finetuning setting. We further include .mp4 files of trajectories provided in ScanNet for better visualization. (i.e., `finetune_compare.mp4` shows qualitative comparison of our method compare with S-Ray [11] under fintuning setting. `finetune_ours.mp4` shows our results with predicting depth map and RGB rendering error map.)

B.3. Observations on Different Number of Source Views

Even though we followed S-Ray and set the number of source views at 8 in all our experiments, we were intrigued to explore how varying the number of source views could influence the performance of the model. Therefore, we conduct an experiment with different numbers of source views on ScanNet, the results of which are presented in Table A4. In the observation depicted in Table A4, we can see that the

Finetuned Method	GT Depth	ScanNet		
	Train / Test	mIoU	acc. / class acc.	PSNR
S-Ray	✔ / ✔	92.4	98.2 / 93.8	27.67
Ours	✔ /	**93.9**	**99.1 / 98.4**	**31.70**
S-Ray	/	91.6	97.3 / 92.2	27.31
Ours	/	**93.2**	**98.2 / 96.8**	**30.89**

Table A3. **Results of finetuning on unseen scenes of ScanNet.** Note that methods in the first two rows take GT depth during training, while S-Ray additionally requires such inputs during testing. The methods in the last two rows do not have access to GT depth during training/testing.

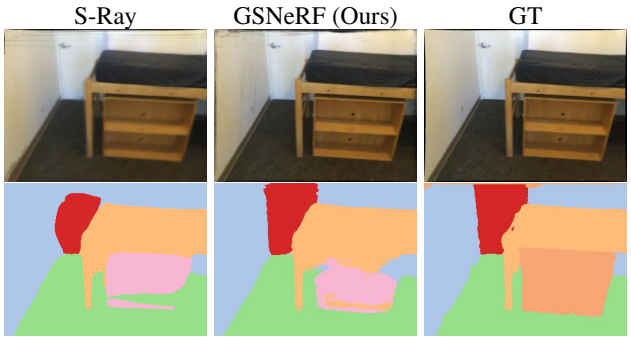

Figure A1. **Qualitative results of finetuning on ScanNet.** Unlike our GSNeRF, S-Ray fails to capture the semantic contour of the door (in red) at the upper-left corner.

utilization of extra source view images is associated with improvements in both visual and semantic segmentation quality. The improvement is more pronounced in the semantic segmentation quality as the number of source views increases. This characteristic motivates our future work to explore the design of a novel view semantic segmentation framework that operates more effectively with fewer input views.

K	mIoU	acc.	class acc.	PSNR
4	48.70	72.71	57.97	31.02
6	51.61	73.92	59.45	30.96
8	**58.30**	**79.79**	**65.93**	**31.33**

Table A4. **Comparisons of different numbers of source-view images on ScanNet** We show the quantitative results of our method, given K = 4, 6, or 8 input views. The testing scene is not seen during training (i.e., the generalized setting).

B.4. Compare with GeoNeRF + semhead

In Section 5.2.1, we mentioned that GeoNeRF + semhead with depth supervision (second row of Table 2) slightly outperforms our approach (fourth row of Table 2) regarding PSNR and SSIM for the rendered RGB images on ScanNet, while our GSNeRF excels in all semantic segmentation

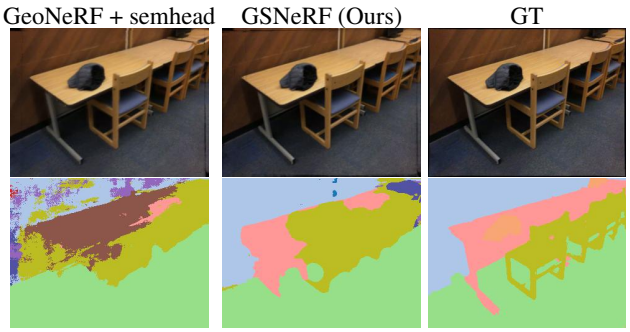

Figure A2. **Qualitative comparisons with GeoNeRF with semhead on ScanNet.** While the difference between the visual quality of rendering images (in the first row) is not remarkable, improved semantic segmentation can be observed for our GSNeRF (in the second row).

metrics by approximately 5%. To further show the advantage of our GSNeRF, we conduct a qualitative comparison in Fig. A2. Despite observing marginal decreases in image rendering metrics (PSNR, SSIM) compared to GeoNeRF + semhead, the perceptual impact on visual quality is not obvious. However, a more notable distinction arises in semantic segmentation quality, as GeoNeRF + semhead produces a pronounced dissimilarity in semantic content with the ground truth (denoted as GT) in Fig. A2. This suggests that while GeoNeRF + semhead may marginally outperform us in image rendering metrics, our method significantly excels in delivering superior semantic segmentation results.

Figure A3. **Semantic and depth error map visualization.** We compare the depth map and semantic prediction with GT and show the error map in binary.

B.5. More Qualitative Evaluation

We show the error map of the depth map and semantic prediction in Fig. A3. Fig. A4 shows more qualitative evaluation results. The first three columns of each row illustrate the novel view image synthesis results from S-Ray, our GSNeRF, and the GT image. The latter three columns present the corresponding novel view semantic segmentation outcomes for S-Ray, our proposed GSNeRF, and the GT semantic segmentation map.

B.6. Limitations

Our GSNeRF is proposed for novel-view scene synthesis and understanding in a generalized setting. Unlike studies like [5, 6, 10, 15] which synthesize novel-view images

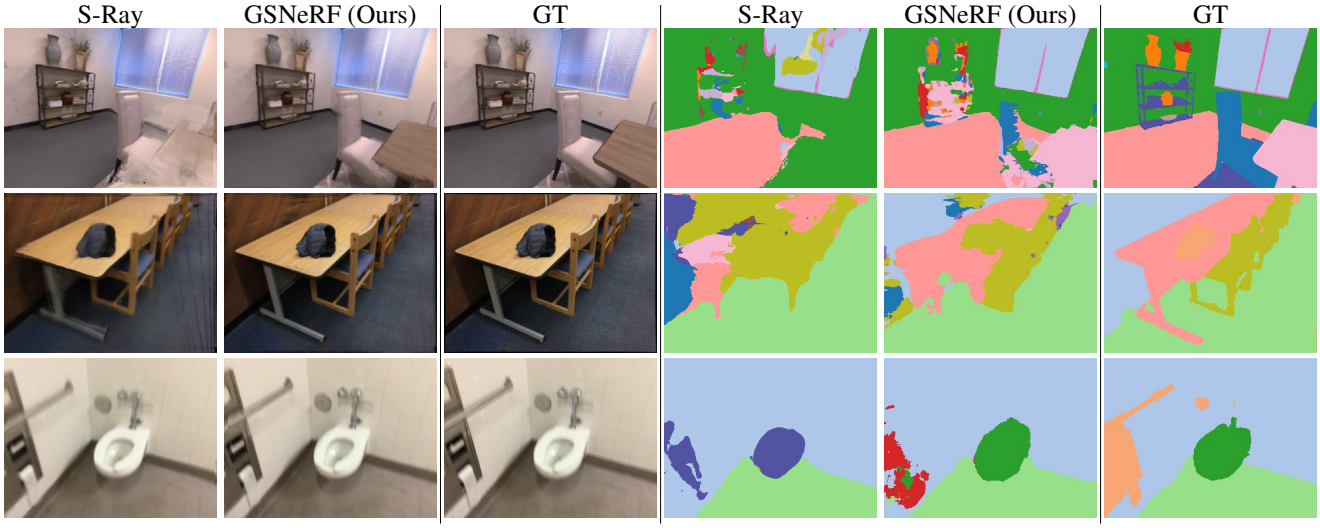

Figure A4. **More qualitative evaluation.** We compare the visual quality of the rendered novel view images (the first three columns) and semantic segmentation maps (the last three columns) with S-Ray [11]. Our method shows clearer image rendering quality and better semantic segmentation results.

for particular objects like human or faces from a single image, our method utilizes self-supervised loss in Eq. 1 for observing cross-view depth consistency. Therefore, extensions of our work to single-image generalizable NeRF for specific 3D objects would be among our future research directions.

References

[1] Di Chang, Aljaž Božič, Tong Zhang, Qingsong Yan, Yingcong Chen, Sabine Süsstrunk, and Matthias Nießner. Rc-mvsnet: unsupervised multi-view stereo with neural rendering. In *European Conference on Computer Vision*, pages 665–680. Springer, 2022. 1

[2] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017. 2

[3] Yuchao Dai, Zhidong Zhu, Zhibo Rao, and Bo Li. Mvs2: Deep unsupervised multi-view stereo with multi-view symmetry. In *2019 International Conference on 3D Vision (3DV)*, pages 1–8. Ieee, 2019. 1

[4] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. *Advances in neural information processing systems*, 27, 2014. 2

[5] Chen Gao, Yichang Shih, Wei-Sheng Lai, Chia-Kai Liang, and Jia-Bin Huang. Portrait neural radiance fields from a single image. *arXiv preprint arXiv:2012.05903*, 2020. 3

[6] Shoukang Hu, Fangzhou Hong, Liang Pan, Haiyi Mei, Lei Yang, and Ziwei Liu. Sherf: Generalizable human nerf from a single image. *arXiv preprint arXiv:2303.12791*, 2023. 3

[7] Baichuan Huang, Hongwei Yi, Can Huang, Yijia He, Jingbin Liu, and Xiao Liu. M3vsnet: Unsupervised multi-metric multi-view stereo network. In *2021 IEEE International Conference on Image Processing (ICIP)*, pages 3163–3167. IEEE, 2021. 1

[8] Mohammad Mahdi Johari, Yann Lepoittevin, and François Fleuret. Geonerf: Generalizing nerf with geometry priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18365–18375, 2022. 2

[9] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 2

[10] Tingting Liao, Xiaomei Zhang, Yuliang Xiu, Hongwei Yi, Xudong Liu, Guo-Jun Qi, Yong Zhang, Xuan Wang, Xiangyu Zhu, and Zhen Lei. High-fidelity clothed avatar reconstruction from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8662–8672, 2023. 3

[11] Fangfu Liu, Chubin Zhang, Yu Zheng, and Yueqi Duan. Semantic ray: Learning a generalizable semantic field with cross-reprojection attention. In *CVPR*, pages 17386–17396, 2023. 2, 4

[12] Yuan Liu, Sida Peng, Lingjie Liu, Qianqian Wang, Peng Wang, Christian Theobalt, Xiaowei Zhou, and Wenping Wang. Neural rays for occlusion-aware image-based rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7824–7833, 2022. 2

[13] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. 2

[14] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Si-

moncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 2

[15] Zhenzhen Weng, Zeyu Wang, and Serena Yeung. Zeroavatar: Zero-shot 3d avatar generation from a single image. *arXiv preprint arXiv:2305.16411*, 2023. 3

[16] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 2