Beyond RGB: Scene-Property Synthesis with Neural Radiance Fields

Supplementary Material

Mingtong Zhang[*,1] Shuhong Zheng[*,1] Zhipeng Bao[2] Martial Hebert[2] Yu-Xiong Wang[1]
[1]University of Illinois Urbana-Champaign [2]Carnegie Mellon University
{mz62, szheng36, yxw}@illinois.edu {zbao, hebert}@cs.cmu.edu

In this supplementary material, we first provide a demo video and the corresponding implementation details in Section 1. In Section 2, we include additional experimental evaluations, including the full comparison with the heuristic and hybrid model, the comparison with the two different modeling strategies for all the properties, full results for the ablation of the color branch, and more visualizations on both the BlendedMVS and the Replica dataset. We conduct more comprehensive investigation with multiple scene properties including multitask learning and transfer learning within our SS-NeRF framework in Section 3. Finally, in Section 4, we offer more implementation details for the model architecture, data set processing and training.

1. Demo Video

We make a demo video in the supplementary zip file to show that our SS-NeRF model has the ability to generalize to arbitrary poses within the scene scope. We pick 9 adjacent views in scene Room_0 of the Replica dataset as anchor views and do an linear interpolation between each pair of adjacent anchor views. For each pair of anchor views, we interpolate 24 new views, thus making a total of $24 \times 8 + 9 = 201$ views. We render the RGB, semantic segmentation, shading, surface normal, keypoint and edge maps using our SS-NeRF model for each of the 201 views and yield the video at a frame rate of 20 FPS.

2. Additional Experimental Details

2.1. Full Results for the Heuristic and Hybrid Models

In the main paper, we provide the averaged results for the Heuristic and Hybrid baselines on Replica Dataset [5]. The full results of these two models are shown in Table 1 (Heuristic) and Table 2 (Hybrid). Combined with the results of Table 1 in the main paper, we can find that SS-NeRF outperforms both of the two models on all the tasks, and all the scenes. These results can verify the effective model design of SS-NeRF, and also indicate that the shared 3D geometry and scene representation are critical for synthesizing different scene properties.

Scene	SL (↑)	SN (↓)	SH (↓)	KP (↓)	ED (↓)
Office_3	0.8849	0.0328	0.0379	0.0055	0.0393
Office_4	0.8824	0.0394	0.0639	0.0060	0.0457
Room_0	0.7924	0.0431	0.0419	0.0068	0.0635
Room_1	0.8721	0.0541	0.0365	0.0052	0.0343
Avg. (Heuristic)	0.8580	0.0424	0.0451	0.0059	0.0457
Avg. (SS-NeRF)	**0.9243**	**0.0395**	**0.0429**	**0.0038**	**0.0179**

Table 1: Performance of the Heuristic model on the Replica dataset. SL: Semantic Labels; SN: Surface Normal; SH: Shading; KP: Keypoint; ED: Edge. Following the standard practice, for the meansurement of SL, we use mIoU as the evaluation metric; for the rest of the tasks, we adopt $\mathcal{L}1$ error as the evaluation metric. SS-NeRF outperforms the heuristic baseline for all the scenes and all the tasks, indicating the effectiveness of the SS-NeRF model design.

Scene	SL (↑)	SN (↓)	SH (↓)	KP (↓)	ED (↓)
Office_3	0.7735	0.0427	0.0612	0.0048	0.0310
Office_4	0.8546	0.0408	0.0732	0.0052	0.0307
Room_0	0.6086	0.0766	0.0721	0.0064	0.0554
Room_1	0.7071	0.0772	0.0625	0.0056	0.0454
Avg. (Hybrid)	0.7360	0.0593	0.0673	0.0055	0.0406
Avg. (SS-NeRF)	**0.9243**	**0.0395**	**0.0429**	**0.0038**	**0.0179**

Table 2: Performance of the Hybrid model on the Replica dataset. SS-NeRF also outperforms the hybrid baseline for all the tasks, showing that it is non-trivial to synthesize paired color images and other scene properties.

2.2. Full Results for the Two Different Modeling

For SS-NeRF model design, we propose two different decoding strategies for different scene properties: $\mathbf{F}^{\text{v}}_{\text{dec}}$ which considers the additional view input, and $\mathbf{F}^{\text{nv}}_{\text{dec}}$ which only takes the embedded 3D coordinates. We show the full comparison of the two modeling strategies for the four scene properties exclude the surface normal in Table 3. The conclusion remains the same as the main paper: $\mathbf{F}^{\text{v}}_{\text{dec}}$ works better for SH, KP and ED, but keeps a slightly worse performance compared with $\mathbf{F}^{\text{nv}}_{\text{dec}}$ for SL. We think the underlying

*Equal contribution

Setting	Office_3				Office_4				Room_0				Room_1			
	SL (↑)	SH (↓)	KP (↓)	ED (↓)	SL (↑)	SH (↓)	KP (↓)	ED (↓)	SL (↑)	SH (↓)	KP (↓)	ED (↓)	SL (↑)	SH (↓)	KP (↓)	ED(↓)
$\mathbf{F}^{\mathrm{v}}_{\mathrm{dec}}$	**0.9361**	**0.0423**	**0.0038**	**0.0155**	0.9072	**0.0503**	**0.0035**	**0.0150**	0.9662	**0.0293**	**0.0039**	**0.0209**	0.8597	**0.0495**	**0.0038**	**0.0202**
$\mathbf{F}^{\mathrm{nv}}_{\mathrm{dec}}$	0.9345	0.0602	**0.0038**	0.0188	**0.9162**	0.0794	0.0038	0.0183	**0.9707**	0.0508	**0.0039**	0.0249	**0.8757**	0.1075	0.0039	0.0225

Table 3: Full results for the comparison between two different modelling. $\mathbf{F}^{\mathrm{v}}_{\mathrm{dec}}$ consistently works better for SH, KP and ED; $\mathbf{F}^{\mathrm{nv}}_{\mathrm{dec}}$ is a better choice for SL, indicating that SS-NeRF indeed learns a geometry-aware scene representation.

Setting	Office_3					Office_4					Room_0					Room_1				
	SL (↑)	SN (↓)	SH (↓)	KP (↓)	ED (↓)	SL (↑)	SN (↓)	SH (↓)	KP (↓)	ED (↓)	SL (↑)	SN (↓)	SH (↓)	KP (↓)	ED (↓)	SL (↑)	SN (↓)	SH (↓)	KP (↓)	ED (↓)
w/ RGB	**0.9345**	**0.0355**	**0.0423**	**0.0038**	**0.0155**	**0.9162**	**0.0383**	**0.0503**	**0.0035**	**0.0150**	**0.9707**	0.0323	**0.0293**	**0.0039**	**0.0209**	**0.8757**	**0.0520**	**0.0495**	**0.0038**	**0.0202**
w/o RGB	0.8128	0.0430	0.0559	0.0098	0.0504	0.5246	0.0490	0.0638	0.0102	0.0566	0.2112	**0.0220**	0.0363	0.0147	0.0722	0.5346	0.0619	0.0644	0.0110	0.0450

Table 4: Full comparison of the model with (w/) or without (w/o) RGB branch. RGB supervision is crucial for building scene representations so that can benefit the learning of other visual tasks.

reason is that the keypoint, the edge, and the shading vary from different viewing directions but the semantic labels keep the same. Therefore, the view inputs are essential for the three properties but are redundant for the semantic modeling. The experimental results can also support that SS-NeRF indeed learns a geometry-aware scene representation.

2.3. Full Ablation for the Color Branch

In the main paper, we report the averaged results of SS-NeRF without the RGB branch. We argue that the color property is a fundamental property of the scene, and it can be beneficial to learning the underlying geometry and scene representations and so can facilitate the learning of the other tasks. In this section, we further provide the full results on all the scenes and the visual comparisons in Table 4 and Figure 1.

We have the following observations: **(1)** In general, including the additional RGB color branch can benefit the learning of the target scene property, especially for the SL, ED, and KP. **(2)** From Figure 1, when RGB supervision is missing, the model inevitably collapses, ending up predicting an all-zero map. After checking the ground-truth and full model predictions, we hypothesize the underlying reason might be that the annotations of keypoint and edge are sparse and low-valued so that without other additional objective, it is hard for the model to learn the underlying feature representations. **(3)** For all these results, the color branch is crucial for building a powerful scene representation so that it can benefit the target scene property synthesis task.

2.4. Results on BlendedMVS dataset

Besides the Replica dataset, we also evaluate our model on the BlendedMVS dataset [6]. We show the visual comparison from our model and the two compared heuristic and hybrid baselines in Fig. 2. Compared with Replica dataset, the BlendedMVS dataset is a easier dataset which mainly contains single object and has smaller view variances. Therefore, the Heuristic baseline works much better in this scenario. From Fig. 2, we can find that SS-NeRF model has a similar performance of the Heuristic baseline but works much better than the Hybrid baseline, indicating that SS-NeRF has a good generalization capability for different working scenarios.

Setting	Office_3	Office_4	Room_0	Room_1
SL	0.9345	0.9162	**0.9707**	0.8757
SL + SN	0.9050(-)	0.9188(+)	0.9692(-)	**0.8910**(+)
SL + SH	0.9406(+)	**0.9594**(+)	0.8872(-)	0.7243(-)
SL + KP	0.9482(+)	0.9122(-)	0.9669(-)	0.8777(+)
SL + ED	0.9265(-)	0.9038(-)	0.9311(-)	0.8675(-)
SL + All	**0.9512**(+)	0.9193(+)	0.8785(-)	0.7450(-)

Table 5: Model performance with additional tasks for semantic labels. SL: semantic labels; SN: surface normal; SH: Shading; KP: keypoints; ED: edges; All: all the four additional tasks. (+) indicate performance increasing, (-) indicate performance drop.

2.5. More Visualizations on Replica

We have also visualize the synthesized scene properties of the other two scenes on Replica in Fig. 3 to show the robustness and effectiveness of the proposed method.

3. Additional Exploration with Multiple Tasks and transfer learning

3.1. Multi-Task Learning

In the main paper, we take the semantic segmentation task as an example to explore potential task relationships among all the tasks. Quantitative results in Table 3 of the main paper show how other tasks influence the semantic segmentation task. Here we show the full comparison of the remaining 4 tasks (SN for surface normal, SL for semantic labels, KP for keypoint and ED for edge). Each time use focus on one task as the target task and treat the others as auxiliary tasks. Displayed in a similar way, experimental results focused on semantic labels (SL), surface normal (SN), keypoint (KP), and edge (ED) are shown in Tables 5, 6, 7, and 8, respectively.

From these results, we have some interesting observations on task relationships. For example, (1) some tasks, such as SN, may be *scene-dependent*, introducing additional tasks consistently benefit the target task in some scenes, while hurt in others; (2) Other scene properties have little effect on the performance of KP in all the scenes, indicating a far

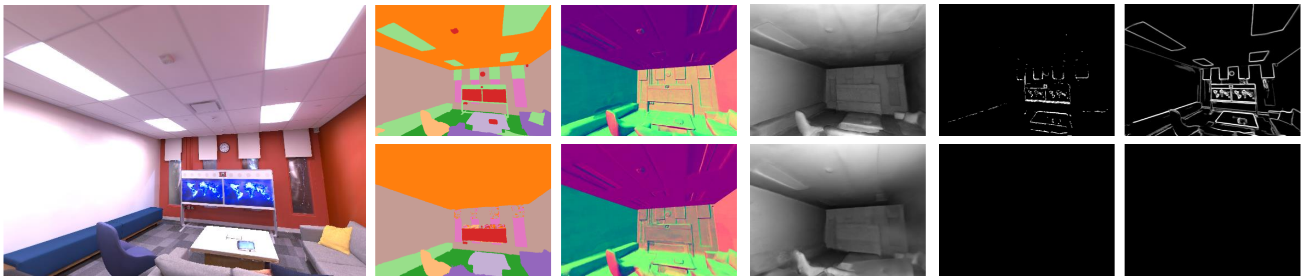

Figure 1: Visualizations of variants of SS-NeRF with or without RGB branch. Upper row: SS-NeRF with color branch; Bottom row: SS-NeRF without color image branch. SS-NeRF degraded, and even failed for KP and ED, without RGB branch, indicating that the color information is crucial for building a solid feature representation and benefiting the synthesis of other properties.

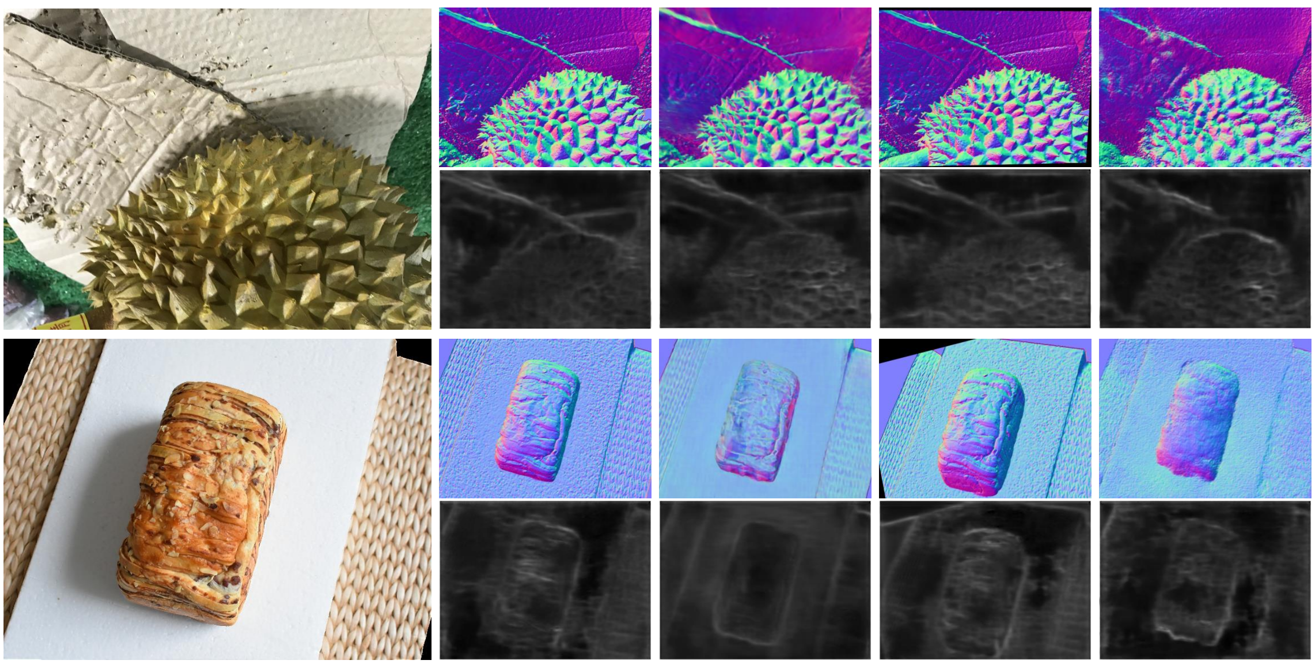

Figure 2: Visual comparison on BlendedMVS [6] dataset. The top row shows the comparison on the surface normal (SN) task. The bottom row shows the comparison on the edge (ED) detection task. Note that for the "Heuristic Model" results, the black/white stripes on the margin of the images mean that we are not able to get information on these regions by projection from the adjacent frames.

Setting	Office_3	Office_4	Room_0	Room_1
SN	0.0355	0.0383	0.0323	0.0520
SN + SL	0.0293(+)	0.0293(+)	**0.0188**(+)	**0.0523**(-)
SN + SH	0.0250(+)	0.0351(+)	0.0232(+)	0.0631(-)
SN + KP	0.0247(+)	**0.0257**(+)	0.0231(+)	0.0526(-)
SN + ED	0.0270(+)	0.0315(+)	0.0212(+)	0.0534(-)
SN + All	**0.0234**(+)	0.0258(+)	0.0213(+)	0.0586(-)

Table 6: Model performance with additional tasks for surface normal. SL: semantic labels; SN: surface normal; SH: Shading; KP: keypoints; ED: edges; All: all the four additional tasks. (+) indicate performance increasing, (-) indicate performance drop.

Setting	Office_3	Office_4	Room_0	Room_1
KP	0.0038	**0.0035**	0.0039	**0.0038**
KP + SL	0.0042(-)	**0.0035**	**0.0037**(+)	0.0039(-)
KP + SN	**0.0036**(+)	**0.0035**	**0.0037**(+)	0.0040(-)
KP + SH	0.0037(+)	0.0036(-)	**0.0037**(+)	0.0041(-)
KP + ED	0.0037(+)	0.0036(-)	0.0039	0.0041(-)
KP + All	0.0037(+)	0.0036(-)	0.0039	0.0041(-)

Table 7: Model performance with additional tasks for keypoint detection. SL: semantic labels; SN: surface normal; SH: Shading; KP: keypoints; ED: edges; All: all the four additional tasks. (+) indicate performance increasing, (-) indicate performance drop.

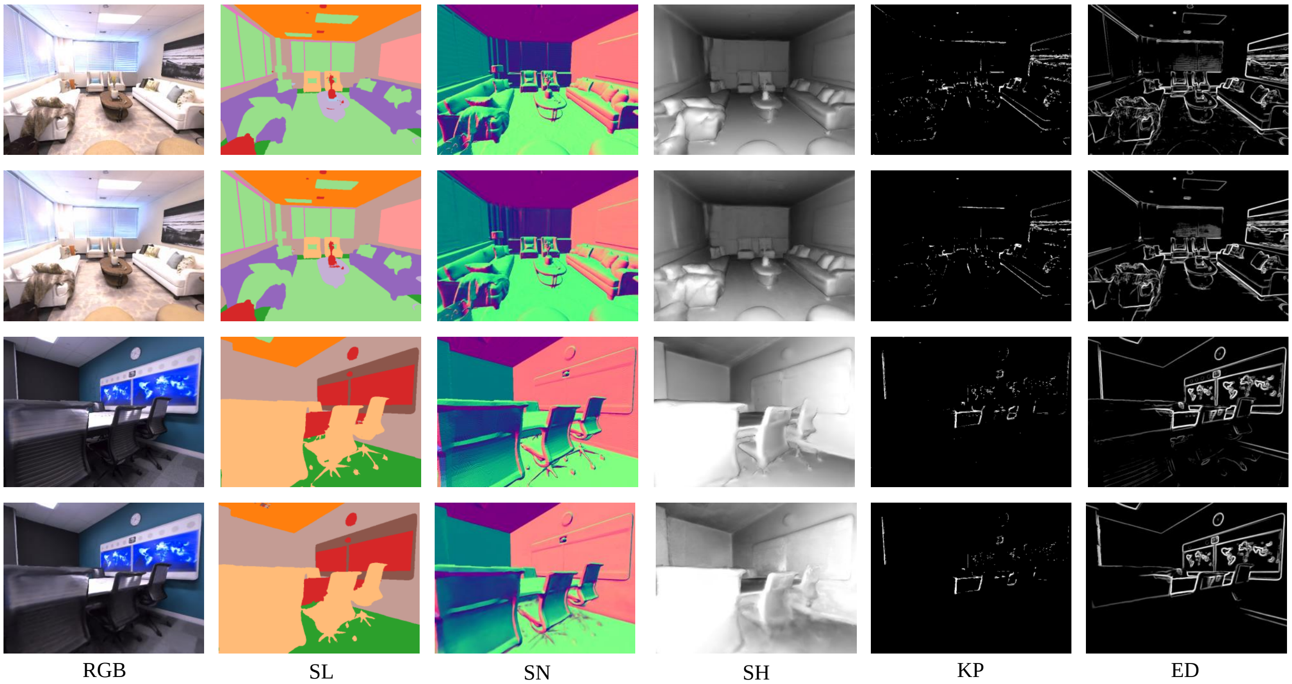

Figure 3: Two additional qualitative results of testing views on Replica [5]. **Top row:** ground-truth; **Bottom row:** our predictions. The predicted image is from the SL task. SS-NeRF is able to generate *realistic and matched* images and other properties.

Setting	Office_3	Office_4	Room_0	Room_1
ED	0.0155	0.0150	0.0209	**0.0202**
ED + SL	0.0149(+)	0.0163(-)	0.0206(+)	0.0226(-)
ED + SN	0.0181(+)	0.0147(+)	0.0209	0.0214(-)
ED + SH	0.0249(-)	0.0144(+)	**0.0204**(+)	0.0439(-)
ED + KP	0.0157(-)	**0.0140**(+)	0.0207(+)	0.0214(-)
ED + All	**0.0143**(+)	0.0146(+)	0.0205(+)	0.0223(-)

Table 8: Model performance with additional tasks for edge detection. SL: semantic labels; SN: surface normal; SH: Shading; KP: keypoints; ED: edges; All: all the four additional tasks. (+) indicate performance increasing, (-) indicate performance drop.

relationship from KP to other scene properties; (3) For each property, in most case we can find a better model when introducing additional properties for jointly training, indicating that the SS-NeRF framework can learn shared knowledge across different tasks. More comprehensive conclusions can be drawn when we apply SS-NeRF to more tasks and more scenes, which we leave as interesting future work.

3.2. Knowledge Transfer

We also provide the full results of transfer learning for all the scene properties in this section. The data setting keep the same as the main paper for all the scene properties. For each target scene property, we first train our model from another source property and transfer the knowledge learned by the source to the target through initializing the learned encoding network $\mathbf{F}_{\text{enc}}$. We focus on the typical transfer learning setting with limited data (6 training views) for the target property. The results for SL, SN, KP, ED are shown in Table 9, 10, 11, and 12 respectively.

We have the following observations and conclusions: (1) For all the transfer learning scenarios except one outlier, the transferred models are able to consistently achieve better performance since they benefit from learned shareable knowledge from other scene properties, indicating the effective generalization of the SS-NeRF framework. (2) Task relationships are easier to find in this difficult transfer learning scenario: SL and SN, KP and ED have closer relationships compared with the other pairs. This conclusion is consistent with human cognition and also the previous work on discriminative multi-task learning papers [4, 8]. (3) Combined with the results on Table 1 in the main paper, *SN* $\rightarrow$ *SL* and achieves a even better result on "Office_4" with fewer labelled data, indicating the knowledge from one property can be transfered to another through our SS-NeRF framework.

4. Additional Implementation Details

4.1. Detailed Model Architecture Design

Our model architecture follows the framework of the NeRF [1] and Semantic-NeRF [9]. Outputs of the tasks modeled by our $\mathbf{F}^{\text{v}}_{\text{dec}}$ decoder are predicted by a fully-connected layer from the hidden layer and the 2D direction (θ, ϕ),

Settings	Office_3	Office_4	Room_0	Room_1
Limited Views	0.7773	0.8283	0.7370	0.5834
SN → SL	**0.8875**	**0.9400**	**0.9318**	0.8267
SH → SL	0.8840	0.9207	0.8172	**0.8395**
KP → SL	0.8528	0.9271	0.7642	0.8190
ED → SL	0.8607	0.9220	0.7865	0.8374

Table 9: Model performance with transfer learning. With the learned shareable knowledge from other scene properties, the transferred model can consistently have better performance, indicating the good generalization of SS-NeRF.

Settings	Office_3	Office_4	Room_0	Room_1
Limited Views	0.0832	0.1028	0.0672	0.1302
SL → SN	**0.0696**	**0.0749**	**0.0493**	**0.1001**
SH → SN	0.0781	0.0853	0.0530	0.1134
KP → SN	0.0730	0.0754	0.0590	0.1075
ED → SN	0.0721	0.0775	0.0563	0.1199

Table 10: Model performance with transfer learning. With the learned shareable knowledge from other scene properties, the transferred model can consistently have better performance, indicating the good generalization of SS-NeRF.

Settings	Office_3	Office_4	Room_0	Room_1
Limited Views	0.0061	**0.0063**	0.0051	0.0079
SL → KP	0.0061	0.0075	0.0056	0.0069
SN → KP	0.0082	0.0082	0.0064	0.0073
SH → KP	0.0101	0.0137	0.0084	0.0138
ED → KP	**0.0059**	0.0075	**0.0050**	**0.0065**

Table 11: Model performance with transfer learning. With the learned shareable knowledge from other scene properties, the transferred model can consistently have better performance, indicating the good generalization of SS-NeRF.

Settings	Office_3	Office_4	Room_0	Room_1
Limited Views	0.0504	0.0436	0.0368	0.0651
SL → ED	0.0292	0.0358	0.0350	0.0327
SN → ED	0.0298	0.0334	**0.0347**	**0.0300**
SH → ED	0.0332	0.0361	0.0722	0.0450
KP → ED	**0.0271**	**0.0326**	0.0379	0.0311

Table 12: Model performance with transfer learning. With the learned shareable knowledge from other scene properties, the transferred model can consistently have better performance, indicating the good generalization of SS-NeRF.

which is prior to the original RGB prediction. Likewise, outputs of the tasks modeled by our $\mathbf{F}_{\text{dec}}^{\text{nv}}$ are predicted by a fully-connected layer from only the hidden layer which is prior to the original volume density prediction. For the modelling of surface normal, since it is independent of viewing direction (θ, ϕ) physically, but dependent to the camera poses which determines the observed value of surface normal, we use $\mathbf{F}_{\text{dec}}^{\text{nv}}$ to model it but additionally encoding the 12-dim pose matrix $\mathbf{p}$ with the same positional embedding used by NeRF [1]:

$$\gamma(\mathbf{p}) = \Big(\sin(2^0\pi\mathbf{p}), \cos(2^0\pi\mathbf{p}), \ldots, \sin(2^{L-1}\pi\mathbf{p}), \cos(2^{L-1}\pi\mathbf{p})\Big) \tag{1}$$

where L is set to 10 for our implementation.

We have also provide the codes in the supplementary zip file for reference. The code is built upon the PyTorch implementation of NeRF[1].

4.2. Additional Implementation Details

For each scene in the Replica dataset, normalize the maximum scale of the camera parameters to 10m and set near and far sample bounds to 0.1m and 10m, respectively. Also, since the views are captured in the face-forwarding manner, we did not use the normalized device coordinate (ndc) provided by NeRF.

For the transfer learning setting, we first froze the $\mathbf{F}_{\text{enc}}$ and warm up the decoding network for 50k iterations with a learning rate 5×10^{-4}. Then we train the whole network jointly with an initial learning rate of 1×10^{-5} for another 150k iterations to get the transferred model.

4.3. Dataset Processing

For Replica [5] and BlendedMVS [6] dataset, both of them have accurate 3D mesh and depth annotations. The Replica dataset also has the accurate semantic labels. We render the other ground-truth annotations by ourselves. First, for the surface normal, as mentioned in the main paper, we directly derive them from depth with:

$$SN(x, y, z) = \left(-\frac{dx}{dz}, -\frac{dy}{dz}, 1\right), \tag{2}$$

where (x, y, z) are the 3D coordinates and $\frac{dx}{dz}$, $\frac{dy}{dz}$ are the gradients of z with respect to x and y respectively. For the other three scene properties (shading, edge and keypoint), we render them with a pre-trained model XTConsistency [7][2]. Notice that the camera pose released by Semantic-NeRF [9] is designated for their own training, we regenerate the camera poses with COLMAP [2, 3].

References

[1] 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.

[2] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016.

[3] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *ECCV*, 2016.

[1] https://github.com/yenchenlin/nerf-pytorch/

[2] https://github.com/EPFL-VILAB/XTConsistency/

[4] Trevor Standley, Amir R Zamir, Dawn Chen, Leonidas Guibas, Jitendra Malik, and Silvio Savarese. Which tasks should be learned together in multi-task learning? In *ICML*, 2020.

[5] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, et al. The replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019.

[6] Yao Yao, Zixin Luo, Shiwei Li, Jingyang Zhang, Yufan Ren, Lei Zhou, Tian Fang, and Long Quan. Blendedmvs: A large-scale dataset for generalized multi-view stereo networks. *CVPR*, 2020.

[7] Amir R Zamir, Alexander Sax, Nikhil Cheerla, Rohan Suri, Zhangjie Cao, Jitendra Malik, and Leonidas J Guibas. Robust learning through cross-task consistency. In *CVPR*, 2020.

[8] Amir R Zamir, Alexander Sax, William Shen, Leonidas J Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *CVPR*, 2018.

[9] Shuaifeng Zhi, Tristan Laidlow, Stefan Leutenegger, and Andrew J Davison. In-place scene labelling and understanding with implicit scene representation. In *ICCV*, 2021.