Acknowledgments. We thank the anonymous reviewers for their valuable feedback. This work is supported by the Early Career Scheme of the Research Grants Council (grant # 27207224), the HKU-100 Award, and in part by NSF China (No. 62172363). Siyan Dong would also like to thank the support from HKU Musketeers Foundation Institute of Data Science for the Postdoctoral Research Fellowship.

6 Detailed Derivation of Mutual Information of Normals

In this section, we derive the mutual information of normals and explain our approximation.

The surface normals can be written as

$$\begin{aligned}\mathbb{N}(p_i) &= (\mathbb{N}^i_{\mathbf{x}}, \mathbb{N}^i_{\mathbf{y}}, \mathbb{N}^i_{\mathbf{z}}),\\ \mathbb{N}(p_j) &= (\mathbb{N}^j_{\mathbf{x}}, \mathbb{N}^j_{\mathbf{y}}, \mathbb{N}^j_{\mathbf{z}}).\end{aligned}$$

When perturbing the normals by a random noise $n \in \mathbb{R}^D$ sampled from $\mathbb{S}^{D-1}$, according to Talor expansion, we have

$$\begin{aligned}\hat{\mathbb{N}}(p_i) &= \mathbb{N}(p_i) + \gamma n \cdot \frac{\partial \mathbb{F}(o_i, v_i; \theta^D + n)}{\partial \theta^D},\\ \hat{\mathbb{N}}(p_j) &= \mathbb{N}(p_j) + \gamma n \cdot \frac{\partial \mathbb{F}(o_j, v_j; \theta^D + n)}{\partial \theta^D},\end{aligned}$$

where $\mathbb{F}$ is the function to calculate the normal information along the ray (weighted sum of the parameter gradients). Let $\partial \mathbb{F}_i / \partial \theta^D = (\gamma_{A_\mathbf{x}} A_\mathbf{x}, \gamma_{A_\mathbf{y}} A_\mathbf{y}, \gamma_{A_\mathbf{z}} A_\mathbf{z})$, and $\partial \mathbb{F}_j / \partial \theta^D = (\gamma_{B_\mathbf{x}} B_\mathbf{x}, \gamma_{B_\mathbf{y}} B_\mathbf{y}, \gamma_{B_\mathbf{z}} B_\mathbf{z})$, where γ denotes the length of the vector, and the normal's partial vectors are all unit vectors, i.e., $A_\mathbf{x}, A_\mathbf{y}, A_\mathbf{z}, B_\mathbf{x}, B_\mathbf{y}, B_\mathbf{z} \in \mathbb{R}^{D-1}$. The computation of mutual information can be written as

$$\begin{aligned}\mathbb{I}(\hat{\mathbb{N}}(p_i), \hat{\mathbb{N}}(p_j)) =& \mathbb{H}(\hat{\mathbb{N}}(p_i)) - \mathbb{H}(\hat{\mathbb{N}}(p_i) \mid \hat{\mathbb{N}}(p_j))\\ =& \mathbb{H}(\gamma n \cdot (\gamma_{A_\mathbf{x}} A_\mathbf{x}, \gamma_{A_\mathbf{y}} A_\mathbf{y}, \gamma_{A_\mathbf{z}} A_\mathbf{z}))\\ -& \mathbb{H}(\gamma n \cdot (\gamma_{A_\mathbf{x}} A_\mathbf{x}, \gamma_{A_\mathbf{y}} A_\mathbf{y}, \gamma_{A_\mathbf{z}} A_\mathbf{z}) \mid \gamma n \cdot (\gamma_{B_\mathbf{x}} B_\mathbf{x}, \gamma_{B_\mathbf{y}} B_\mathbf{y}, \gamma_{B_\mathbf{z}} B_\mathbf{z})).\end{aligned}$$

Starting from a simple situation, we compute the entropy of $\gamma_{A_\mathbf{x}} \gamma n \cdot A_\mathbf{x}$. The entropy of a distribution shifts by the logarithm of the scaling factor when it's scaled. Therefore, we have

$$\mathbb{H}(\gamma_{A_\mathbf{x}} \gamma n \cdot A_\mathbf{x}) = \log(\gamma_{A_\mathbf{x}} \gamma) + \mathbb{H}(\mathbb{S}^{D-1}),$$

where $\mathbb{H}(\mathbb{S}^{D-1})$ is a constant. Then, for the joint entropy of two partial vectors:

$$\mathbb{H}(\gamma_{A_\mathbf{x}} \gamma n \cdot A_\mathbf{x}, \gamma_{A_\mathbf{y}} \gamma n \cdot A_\mathbf{y}) = \mathbb{H}(\gamma_{A_\mathbf{x}} \gamma n \cdot A_\mathbf{x}) + \mathbb{H}(\gamma_{A_\mathbf{y}} \gamma n \cdot A_\mathbf{y} \mid \gamma_{A_\mathbf{x}} \gamma n \cdot A_\mathbf{x}),$$

where

$$\mathbb{H}(\gamma_{A_\mathbf{y}} \gamma n \cdot A_\mathbf{y} \mid \gamma_{A_\mathbf{x}} \gamma n \cdot A_\mathbf{x}) = \int_s \mathbb{H}(\gamma_{A_\mathbf{y}} \gamma n \cdot A_\mathbf{y} \mid \gamma_{A_\mathbf{x}} \gamma n \cdot A_\mathbf{x} = s) p(s) ds.$$

When $\gamma_{A_{\mathbf{x}}}\gamma n \cdot A_{\mathbf{x}} = s$

$$\begin{aligned}\gamma_{A_{\mathbf{y}}}\gamma n \cdot A_{\mathbf{y}} =&\gamma_{A_{\mathbf{y}}}\gamma(\langle A_{\mathbf{x}}, A_{\mathbf{y}}\rangle A_{\mathbf{x}} + (A_{\mathbf{y}} - \langle A_{\mathbf{x}}, A_{\mathbf{y}}\rangle A_{\mathbf{x}}))\\ &\cdot (\langle n, A_{\mathbf{x}}\rangle A_{\mathbf{x}} + (n - \langle n, A_{\mathbf{x}}\rangle A_{\mathbf{x}}))\\ =&\gamma_{A_{\mathbf{y}}}\gamma(\langle n, A_{\mathbf{x}}\rangle \cdot \langle A_{\mathbf{x}}, A_{\mathbf{y}}\rangle + (A_{\mathbf{y}} - \langle A_{\mathbf{x}}, A_{\mathbf{y}}\rangle A_{\mathbf{x}}) \cdot (n - \langle n, A_{\mathbf{x}}\rangle A_{\mathbf{x}})),\end{aligned}$$

so that

$$\begin{aligned}&\mathbb{H}(\gamma_{A_{\mathbf{y}}}\gamma n \cdot A_{\mathbf{y}} \mid \gamma_{A_{\mathbf{x}}}\gamma n \cdot A_{\mathbf{x}} = s)\\ =&\mathbb{H}(\gamma_{A_{\mathbf{y}}}\gamma * (\langle n, A_{\mathbf{x}}\rangle \cdot \langle A_{\mathbf{x}}, A_{\mathbf{y}}\rangle\\ &+ (A_{\mathbf{y}} - \langle A_{\mathbf{x}}, A_{\mathbf{y}}\rangle A_{\mathbf{x}}) \cdot (n - \langle n, A_{\mathbf{x}}\rangle \cdot A_{\mathbf{x}})) \mid \gamma_{A_{\mathbf{x}}}\gamma n \cdot A_{\mathbf{x}} = s)\\ =&\log(\gamma_{A_{\mathbf{y}}}\gamma \sin(A_{\mathbf{x}}, A_{\mathbf{y}}) \sin(n, A_{\mathbf{x}})) + \mathbb{H}(\mathbb{S}^{D-2}).\end{aligned}$$

We have

$$\begin{aligned}&\mathbb{H}(\gamma_{A_{\mathbf{y}}}\gamma n \cdot A_{\mathbf{y}} \mid \gamma_{A_{\mathbf{x}}}\gamma n \cdot A_{\mathbf{x}})\\ =&\int_s \mathbb{H}(\gamma_{A_{\mathbf{y}}}\gamma n \cdot A_{\mathbf{y}} \mid \gamma_{A_{\mathbf{x}}}\gamma n \cdot A_{\mathbf{x}} = s) p(s) ds\\ =&\int_s (\log(\gamma_{A_{\mathbf{y}}}\gamma \sin(A_{\mathbf{x}}, A_{\mathbf{y}}) \sin(n, A_{\mathbf{x}})) + \mathbb{H}(\mathbb{S}^{D-2})) p(s) ds\\ =&\mathbb{H}(\mathbb{S}^{D-2}) + \log(\gamma_{A_{\mathbf{y}}}\gamma \sin(A_{\mathbf{x}}, A_{\mathbf{y}})) + \int_s \log(\sin(n, A_{\mathbf{x}})) p(s) ds\\ =&\log(\gamma_{A_{\mathbf{y}}}\gamma \sin(A_{\mathbf{x}}, A_{\mathbf{y}})) + const.\end{aligned}$$

In more general terms, we denote $A_{\mathbf{y}} - \langle A_{\mathbf{x}}, A_{\mathbf{y}}\rangle A_{\mathbf{x}}$ as $P(A_{\mathbf{y}}, A_{\mathbf{x}})$, which represents subtracting the component in the $A_{\mathbf{x}}$ direction from the $A_{\mathbf{y}}$. Therefore, the equation above can be written as

$$\mathbb{H}(\gamma_{A_{\mathbf{y}}}\gamma n \cdot A_{\mathbf{y}} \mid \gamma_{A_{\mathbf{x}}}\gamma n \cdot A_{\mathbf{x}}) = \log(\gamma_{A_{\mathbf{y}}}\gamma |P(A_{\mathbf{y}}, A_{\mathbf{x}})|) + const.$$

$$\begin{aligned}&\mathbb{H}(\gamma_{A_{\mathbf{x}}}\gamma n \cdot A_{\mathbf{x}}, \gamma_{A_{\mathbf{y}}}\gamma n \cdot A_{\mathbf{y}})\\ =&\mathbb{H}(\gamma_{A_{\mathbf{x}}}\gamma n \cdot A_{\mathbf{x}}) + \mathbb{H}(\gamma_{A_{\mathbf{y}}}\gamma n \cdot A_{\mathbf{y}} \mid \gamma_{A_{\mathbf{x}}}\gamma n \cdot A_{\mathbf{x}})\\ =&\log(\gamma_{A_{\mathbf{x}}}\gamma) + \log(\gamma_{A_{\mathbf{y}}}\gamma |P(A_{\mathbf{y}}, A_{\mathbf{x}})|) + const.\end{aligned}$$

Similarly, we can infer that

$$\begin{aligned}&\mathbb{H}(\gamma_{A_{\mathbf{x}}}\gamma n \cdot A_{\mathbf{x}}, \gamma_{A_{\mathbf{y}}}\gamma n \cdot A_{\mathbf{y}}, \gamma_{A_{\mathbf{z}}}\gamma n \cdot A_{\mathbf{z}})\\ =&\log(\gamma_{A_{\mathbf{x}}}\gamma) + \log(\gamma_{A_{\mathbf{y}}}\gamma |P(A_{\mathbf{y}}, A_{\mathbf{x}})|) + \log(\gamma_{A_{\mathbf{z}}}\gamma |P(A_{\mathbf{z}}, (A_{\mathbf{x}}, A_{\mathbf{y}}))|) + const.\end{aligned}$$

$$\begin{aligned}&\mathbb{H}(\gamma_{A_{\mathbf{x}}}\gamma n \cdot A_{\mathbf{x}}, \gamma_{A_{\mathbf{y}}}\gamma n \cdot A_{\mathbf{y}}, \gamma_{A_{\mathbf{z}}}\gamma n \cdot A_{\mathbf{z}} \mid \gamma n \cdot \gamma_{B_{\mathbf{x}}}\gamma B_{\mathbf{x}}, \gamma_{B_{\mathbf{y}}}\gamma B_{\mathbf{y}}, \gamma_{B_{\mathbf{z}}}\gamma B_{\mathbf{z}})\\ =&\log(\gamma_{A_{\mathbf{x}}}\gamma |P(A_{\mathbf{x}}, (B_{\mathbf{x}}, B_{\mathbf{y}}, B_{\mathbf{z}}))|) + \log(\gamma_{A_{\mathbf{y}}}\gamma |P(A_{\mathbf{y}}, (A_{\mathbf{x}}, B_{\mathbf{x}}, B_{\mathbf{y}}, B_{\mathbf{z}}))|)\\ &+ \log(\gamma_{A_{\mathbf{z}}}\gamma |P(A_{\mathbf{z}}, (A_{\mathbf{x}}, A_{\mathbf{y}}, B_{\mathbf{x}}, B_{\mathbf{y}}, B_{\mathbf{z}}))|) + const.\end{aligned}$$

By combining them, we obtain

$$\begin{aligned}
&\mathbb{I}(\hat{\mathbb{N}}(p_i), \hat{\mathbb{N}}(p_j)) \\
=&\mathbb{H}(\hat{\mathbb{N}}(p_i)) - \mathbb{H}(\hat{\mathbb{N}}(p_i) \mid \hat{\mathbb{N}}(p_j)) \\
=&\mathbb{H}(\gamma n \cdot (\gamma_{A_\mathbf{x}} A_\mathbf{x}, \gamma_{A_\mathbf{y}} A_\mathbf{y}, \gamma_{A_\mathbf{z}} A_\mathbf{z})) - \\
&\mathbb{H}(\gamma n \cdot (\gamma_{A_\mathbf{x}} A_\mathbf{x}, \gamma_{A_\mathbf{y}} A_\mathbf{y}, \gamma_{A_\mathbf{z}} A_\mathbf{z}) \mid \gamma n \cdot (\gamma_{B_\mathbf{x}} B_\mathbf{x}, \gamma_{B_\mathbf{y}} B_\mathbf{y}, \gamma_{B_\mathbf{z}} B_\mathbf{z})) \\
=&\log(\gamma_{A_\mathbf{x}} \gamma) \\
&+ \log(\gamma_{A_\mathbf{y}} \gamma |P(A_\mathbf{y}, A_\mathbf{x})|) \\
&+ \log(\gamma_{A_\mathbf{z}} \gamma |P(A_\mathbf{z}, (A_\mathbf{x}, A_\mathbf{y}))|) \\
&- \log(\gamma_{A_\mathbf{x}} \gamma |P(A_\mathbf{x}, (B_\mathbf{x}, B_\mathbf{y}, B_\mathbf{z}))|) \\
&- \log(\gamma_{A_\mathbf{y}} \gamma |P(A_\mathbf{y}, (A_\mathbf{x}, B_\mathbf{x}, B_\mathbf{y}, B_\mathbf{z}))| \\
&- \log(\gamma_{A_\mathbf{z}} \gamma |P(A_\mathbf{z}, (A_\mathbf{x}, A_\mathbf{y}, B_\mathbf{x}, B_\mathbf{y}, B_\mathbf{z}))|)) \\
&+ const. \\
=&\log \frac{|P(A_\mathbf{y}, A_\mathbf{x})||P(A_\mathbf{z}, (A_\mathbf{x}, A_\mathbf{y}))|}{|P(A_\mathbf{x}, (B_\mathbf{xyz}))||P(A_\mathbf{y}, (A_\mathbf{x}, B_\mathbf{xyz}))||P(A_\mathbf{z}, (A_\mathbf{x}, A_\mathbf{y}, B_\mathbf{xyz}))|} \\
&+ const.
\end{aligned}$$

$B_\mathbf{xyz}$ represent the space constructed by $B_\mathbf{x}$, $B_\mathbf{y}$ and $B_\mathbf{z}$.

Restricted by computational complexity, we approximate it by

$$\begin{aligned}
&\mathbb{I}(\hat{\mathbb{N}}(p_i), \hat{\mathbb{N}}(p_j)) \\
\approx&\log \frac{1}{|P(A_\mathbf{x}, B_\mathbf{x})||P(A_\mathbf{y}, B_\mathbf{y})||P(A_\mathbf{z}, B_\mathbf{z})|} + const.
\end{aligned}$$

It only considers the relationship of corresponding parts from the weighted sum with respect to parameter gradients. To further simplify, we used a simple formula which computes the cosine similarity of their concatenated gradients as described in main paper.

7 More Information on Datasets and Evaluation Metrics

Datasets. We report the statistics of the scenes in our evaluation in Tab. 5. In addition to the number of images, we calculate the average proportion of overlapping pixels between adjacent images.

Evaluation metrics. We use the L2 Chamfer distance and F-score to evaluate the reconstruction results. Both the two metrics are computed on top of the meshes: the ground truth mesh and the reconstructed one. The Chamfer distance is computed as:

$$cd(S_1, S_2) = \frac{1}{|S_1|} \sum_{x \in S_1} \min_{y \in S_2} ||x - y||_2^2 + \frac{1}{|S_2|} \sum_{x \in S_2} \min_{y \in S_1} ||x - y||_2^2. \qquad (13)$$

Table 5: Statistics of the scenes in our evaluation on ScanNet++ and Replica.

Datasets	ScanNet++								Replica			
Scene name	0a7c	0a18	6ee2	7b64	56a0	9460	a08d	e0ab	office0	office1	room0	room1
Number of images	67	63	75	78	72	79	85	57	60	60	60	60
Overlaps (our split)	0.82	0.82	0.75	0.77	0.73	0.80	0.83	0.65	0.89	0.91	0.90	0.90
Original overlaps	0.96	0.96	0.98	0.95	0.90	0.97	0.98	0.94	0.99	0.99	0.99	0.99

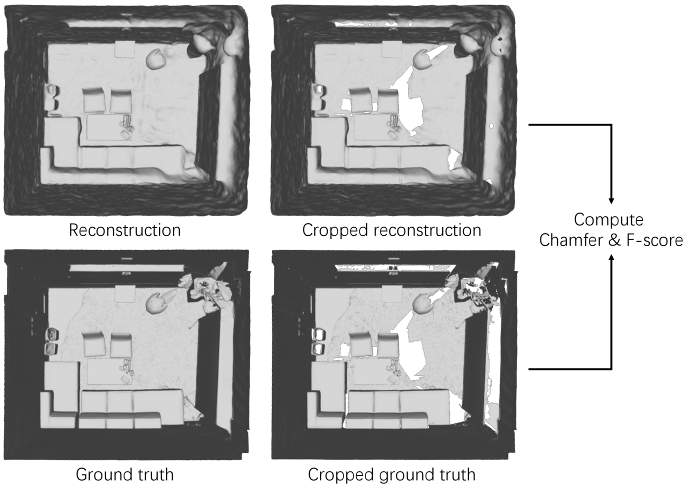

Fig. 6: Cropped reconstruction and ground truth with training viewpoints.

In this case, S_1 and S_2 represent the two point sets sampled from the ground truth mesh and the reconstruction, respectively.

The F-socre is computed as:

$$fs(S_1, S_2) = 2 \cdot \frac{precision(S_1, S_2) \times recall(S_1, S_2)}{precision(S_1, S_2) + recall(S_1, S_2)}. \tag{14}$$

The values for precision and recall are determined by the proportion of sampled points, where the distance to the nearest point in another mesh is less than 2% of the scene length. Precision is calculated from the reconstruction to the ground truth, while recall is calculated in the opposite direction.

We sampled 50,000 points from each original mesh to calculate the metrics. To ensure a fair evaluation, we remove all parts of the geometry that are not visible from the training views, as shown in Figure 6.

8 More Implementation Details for Each Baseline

The numbers reported in Table 4 in the main paper were measured on an NVIDIA H800. For all the methods we apply our mutual information shaping to their official codes at GitHub.

- NeuS [30]$^{+}$: https://github.com/Totoro97/NeuS. The network is trained by 160k iterations.
- VolSDF [37]$^{+}$: https://github.com/lioryariv/volsdf. The network is trained by 150k iterations.
- GeoNeuS [8]$^{+}$: https://github.com/GhiXu/Geo-Neus. We use adjacent 8 images (4 before and 4 after) as reference perspectives. The network is trained by 150k iterations.
- I^2-SDF [44]$^{+}$: https://github.com/jingsenzhu/i2-sdf. We discard the normal and depth supervision. The network is trained by 150k iterations.
- NeuRIS [29]$^{+}$: https://github.com/jiepengwang/NeuRIS. The network is trained by 160k iterations.
- MonoSDF [40]$^{+}$: https://github.com/autonomousvision/monosdf. We set the decay for the normal and depth loss at 30k iterations. We observed that full use causes the method to degenerate into estimation fusion, rather than reconstruction from posed images. The network is trained by 100k iterations.
- Neuralangelo [13]$^{+}$: https://github.com/NVlabs/neuralangelo. We set the hash encoding dictionary size to 20 and the feature dimension to 4. This is to ensure VRAM consumption stays at the same level as with other methods. The network is trained by 150k iterations.

Table 6: λ_M values.

Method	Value
NeuS^{+}	1.0
VolSDF^{+}	0.3
GeoNeuS^{+}	1.0
I^2-SDF^{+}	0.3
NeuRIS^{+}	1.0
MonoSDF^{+}	0.5
Neuralangelo^{+}	1.0

For different baselines, we use different weights λ_M to balance the original training and our mutual information shaping. The detailed weights are reported in Table 6. For all the experiments in the main paper, we set the positive sample threshold with $\beta_S = 0.65$ and $\beta_G = 0.99$ for DINO [4] and normal features [1], respectively.

9 Additional Comparisons

In the main paper, we exclude GeoNeuS [8]$^{+}$ and MonoSDF [40]$^{+}$ from the Replica dataset. This is because there are no readily available structure-from-motion models for GeoNeuS, and MonoSDF's training and ablation studies are based on Replica. Here, we report their performance in Tab. 7.

Table 7: Quantitative results on the Replica dataset.

		office0	office1	room0	room1	Mean
Chamfer (m)↓	GeoNeuS [8]$^+$	0.0230-0.0206	0.0136 -0.0098	0.0396 -0.0354	0.0024 -0.0003	0.0196-0.0165
	MonoSDF [40]$^+$	0.0028+0.001	0.0047 -0.0006	0.0041 -0.0003	0.0044 -0.0007	0.0040-0.0004
F-score ↑	GeoNeuS [8]$^+$	0.891+0.083	0.910 +0.039	0.896 +0.085	0.977 +0.007	0.919+0.054
	MonoSDF [40]$^+$	0.962+0.004	0.901 +0.021	0.982 +0.004	0.946 +0.016	0.948+0.012

For GeoNeuS, we utilize COLMAP [25] to build the structure-from-motion models. We input known camera parameters and perform only triangulation. The table illustrates the advantages of our method to enhance GeoNeuS. Similarly, MonoSDF$^+$ also benefits from our mutual information shaping.

10 Additional Analyses

Effectiveness of the semantic features. We examine the effectiveness of the semantic feature DINO by replacing it with a semantic segmentation model, SAM [12]. The results are shown in Tab. 8. As observed, employing positive-negative pairs with SAM can enhance the baseline performance in some instances (NeuRIS), but it can also hurt performance in other cases (MonoSDF). On the contrary, using DINO features consistently enhances both the two baselines. Therefore, we apply DINO in our method and report the results with DINO features accordingly in the main paper.

Table 8: Ablation study with image segmentation model - SAM [12].

	Chamfer (m)↓				F-score ↑			
	6ee2	7b64	9460	Mean	6ee2	7b64	9460	Mean
NeuRIS [29]	**0.029**	0.070	0.405	0.168	0.65	0.69	0.24	0.53
NeuRIS$^+$ (SAM)	0.038	0.042	0.337	0.139	**0.69**	0.73	0.30	0.57
NeuRIS$^+$ (SAM+normal)	0.053	**0.030**	0.412	0.165	0.68	**0.77**	0.22	0.56
NeuRIS$^+$ (Full)	0.044	0.033	**0.198**	**0.092**	0.66	0.76	**0.41**	**0.61**
MonoSDF [40]	0.020	**0.016**	0.046	0.028	0.81	0.79	0.88	0.83
MonoSDF$^+$ (SAM)	0.527	0.018	0.035	0.193	0.63	0.77	0.86	0.75
MonoSDF$^+$ (SAM+normal)	0.062	0.018	0.029	0.037	0.77	0.80	0.88	0.82
MonoSDF$^+$ (Full)	**0.014**	0.020	**0.023**	**0.019**	**0.85**	**0.85**	**0.93**	**0.87**

Performance of first-order method. In Tab. 9, we report the results of directly aligning the normal directions (denoted by FO) among positive pairs (i.e., correlated surfaces). It is implemented by replacing L_M with

$$L'_M = -\log(\sum \exp(|| \cos(\mathbb{N}_i, \mathbb{N}_{i+}||)), \tag{15}$$

Table 9: Ablation study for the first-order method with correlated normals.

		6ee2	7b64	9460	Mean
Chamfer↓	NeuRIS$^+$ (FO)	0.586	0.490	0.603	0.560
	NeuRIS$^+$ (Full)	**0.044**	**0.033**	**0.198**	**0.092**
	MonoSDF$^+$ (FO)	1.207	1.024	1.163	1.132
	MonoSDF$^+$ (Full)	**0.014**	**0.020**	**0.023**	**0.019**
F-score↑	NeuRIS$^+$ (FO)	0.43	0.34	0.21	0.33
	NeuRIS$^+$ (Full)	**0.66**	**0.76**	**0.41**	**0.61**
	MonoSDF$^+$ (FO)	0.59	0.49	0.57	0.55
	MonoSDF$^+$ (Full)	**0.85**	**0.85**	**0.93**	**0.87**

which is similar to Eq. 11 but removes the calculation of second-order and the part of negative pairs. We notice performance drops, and the results appear over-smoothed. This is mainly because a) the positive pairs have similar but not identical directions, and b) the information from negative pairs is not utilized.

Table 10: Quantitative results on the DTU dataset.

		24	37	40	Mean
Chamfer↓	NeuRIS [29]	**0.980**	3.674	0.865	1.840
	NeuRIS$^+$	1.023	**3.341**	**0.663**	**1.676**
	MonoSDF [40]	0.876	**1.773**	0.657	1.102
	MonoSDF$^+$	**0.837**	1.816	**0.626**	**1.093**

More discussions on limitation. While our method does not rely on Manhattan world or near-planar assumptions, we have found that its effectiveness on object-level scenes is reduced. In Tab 10, we present the quantitative results on object-level inward-facing scenes from the DTU dataset. The experiments are carried out in the first three scenes with two baselines, using the same hyperparameter settings as those used in the indoor scenes. In these scenes, we note that DINO features or monocular normals sometimes produce inconsistent pairs, which could affect the effectiveness of our mutual information shaping.

References

1. Bae, G., Budvytis, I., Cipolla, R.: Estimating and exploiting the aleatoric uncertainty in surface normal estimation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 13137–13146 (2021)
2. Bao, C., Zhang, Y., Yang, B., Fan, T., Yang, Z., Bao, H., Zhang, G., Cui, Z.: Sine: Semantic-driven image-based nerf editing with prior-guided editing field. In:

Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 20919–20929 (2023)

3. Cai, B., Huang, J., Jia, R., Lv, C., Fu, H.: Neuda: Neural deformable anchor for high-fidelity implicit surface reconstruction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8476–8485 (2023)
4. Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging properties in self-supervised vision transformers. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 9650–9660 (2021)
5. Chen, Y., Wu, Q., Zheng, C., Cham, T.J., Cai, J.: Sem2nerf: Converting single-view semantic masks to neural radiance fields. In: European Conference on Computer Vision. pp. 730–748. Springer (2022)
6. Curless, B., Levoy, M.: A volumetric method for building complex models from range images. In: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques. pp. 303–312 (1996)
7. Do, T., Vuong, K., Roumeliotis, S.I., Park, H.S.: Surface normal estimation of tilted images via spatial rectifier. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16. pp. 265–280. Springer (2020)
8. Fu, Q., Xu, Q., Ong, Y.S., Tao, W.: Geo-neus: Geometry-consistent neural implicit surfaces learning for multi-view reconstruction. Advances in Neural Information Processing Systems **35**, 3403–3416 (2022)
9. Gropp, A., Yariv, L., Haim, N., Atzmon, M., Lipman, Y.: Implicit geometric regularization for learning shapes. arXiv preprint arXiv:2002.10099 (2020)
10. Guo, H., Peng, S., Lin, H., Wang, Q., Zhang, G., Bao, H., Zhou, X.: Neural 3d scene reconstruction with the manhattan-world assumption. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5511–5520 (2022)
11. Kerr, J., Kim, C.M., Goldberg, K., Kanazawa, A., Tancik, M.: Lerf: Language embedded radiance fields. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 19729–19739 (2023)
12. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., et al.: Segment anything. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4015–4026 (2023)
13. Li, Z., Müller, T., Evans, A., Taylor, R.H., Unberath, M., Liu, M.Y., Lin, C.H.: Neuralangelo: High-fidelity neural surface reconstruction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8456–8465 (2023)
14. Liu, F., Zhang, C., Zheng, Y., Duan, Y.: Semantic ray: Learning a generalizable semantic field with cross-reprojection attention. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 17386–17396 (2023)
15. Liu, Y.L., Gao, C., Meuleman, A., Tseng, H.Y., Saraf, A., Kim, C., Chuang, Y.Y., Kopf, J., Huang, J.B.: Robust dynamic radiance fields. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13–23 (2023)
16. Liu, Z., Milano, F., Frey, J., Siegwart, R., Blum, H., Cadena, C.: Unsupervised continual semantic adaptation through neural rendering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3031–3040 (2023)

17. Long, X., Lin, C., Wang, P., Komura, T., Wang, W.: Sparseneus: Fast generalizable neural surface reconstruction from sparse views. In: European Conference on Computer Vision. pp. 210–227. Springer (2022)
18. Lorensen, W.E., Cline, H.E.: Marching cubes: A high resolution 3d surface construction algorithm. In: Seminal graphics: pioneering efforts that shaped the field, pp. 347–353 (1998)
19. Meng, X., Chen, W., Yang, B.: Neat: Learning neural implicit surfaces with arbitrary topologies from multi-view images. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 248–258 (2023)
20. Metzer, G., Richardson, E., Patashnik, O., Giryes, R., Cohen-Or, D.: Latent-nerf for shape-guided generation of 3d shapes and textures. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12663–12673 (2023)
21. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. In: ECCV (2020)
22. Oord, A.v.d., Li, Y., Vinyals, O.: Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748 (2018)
23. Park, J.J., Florence, P., Straub, J., Newcombe, R., Lovegrove, S.: Deepsdf: Learning continuous signed distance functions for shape representation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 165–174 (2019)
24. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning. pp. 8748–8763. PMLR (2021)
25. Schönberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
26. Straub, J., Whelan, T., Ma, L., Chen, Y., Wijmans, E., Green, S., Engel, J.J., Mur-Artal, R., Ren, C., Verma, S., Clarkson, A., Yan, M., Budge, B., Yan, Y., Pan, X., Yon, J., Zou, Y., Leon, K., Carter, N., Briales, J., Gillingham, T., Mueggler, E., Pesqueira, L., Savva, M., Batra, D., Strasdat, H.M., Nardi, R.D., Goesele, M., Lovegrove, S., Newcombe, R.: The replica dataset: A digital replica of indoor spaces (2019)
27. Tertikas, K., Despoina, P., Pan, B., Park, J.J., Uy, M.A., Emiris, I., Avrithis, Y., Guibas, L.: Partnerf: Generating part-aware editable 3d shapes without 3d supervision. arXiv preprint arXiv:2303.09554 (2023)
28. Wang, F., Galliani, S., Vogel, C., Speciale, P., Pollefeys, M.: Patchmatchnet: Learned multi-view patchmatch stereo. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 14194–14203 (2021)
29. Wang, J., Wang, P., Long, X., Theobalt, C., Komura, T., Liu, L., Wang, W.: Neuris: Neural reconstruction of indoor scenes using normal priors. In: European Conference on Computer Vision. pp. 139–155. Springer (2022)
30. Wang, P., Liu, L., Liu, Y., Theobalt, C., Komura, T., Wang, W.: Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. arXiv preprint arXiv:2106.10689 (2021)
31. Weder, S., Garcia-Hernando, G., Monszpart, A., Pollefeys, M., Brostow, G.J., Firman, M., Vicente, S.: Removing objects from neural radiance fields. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16528–16538 (2023)

32. Xiangli, Y., Xu, L., Pan, X., Zhao, N., Rao, A., Theobalt, C., Dai, B., Lin, D.: Bungeenerf: Progressive neural radiance field for extreme multi-scale scene rendering. In: European conference on computer vision. pp. 106–122. Springer (2022)
33. Xu, L., Xiangli, Y., Peng, S., Pan, X., Zhao, N., Theobalt, C., Dai, B., Lin, D.: Grid-guided neural radiance fields for large urban scenes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8296–8306 (2023)
34. Xu, X., Yang, Y., Mo, K., Pan, B., Yi, L., Guibas, L.: Jacobinerf: Nerf shaping with mutual information gradients. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16498–16507 (2023)
35. Yan, Z., Li, C., Lee, G.H.: Nerf-ds: Neural radiance fields for dynamic specular objects. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8285–8295 (2023)
36. Yao, Y., Luo, Z., Li, S., Fang, T., Quan, L.: Mvsnet: Depth inference for unstructured multi-view stereo. In: Proceedings of the European conference on computer vision (ECCV). pp. 767–783 (2018)
37. Yariv, L., Gu, J., Kasten, Y., Lipman, Y.: Volume rendering of neural implicit surfaces. Advances in Neural Information Processing Systems **34**, 4805–4815 (2021)
38. Yeshwanth, C., Liu, Y.C., Nießner, M., Dai, A.: Scannet++: A high-fidelity dataset of 3d indoor scenes. In: Proceedings of the International Conference on Computer Vision (ICCV) (2023)
39. Yu, H., Julin, J., Milacski, Z.A., Niinuma, K., Jeni, L.A.: Dylin: Making light field networks dynamic. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12397–12406 (2023)
40. Yu, Z., Peng, S., Niemeyer, M., Sattler, T., Geiger, A.: Monosdf: Exploring monocular geometric cues for neural implicit surface reconstruction. Advances in neural information processing systems **35**, 25018–25032 (2022)
41. Zhang, Y., Yang, G., Guibas, L., Yang, Y.: Infogaussian: Structure-aware dynamic gaussians through lightweight information shaping. arXiv preprint arXiv:2406.05897 (2024)
42. Zhi, S., Laidlow, T., Leutenegger, S., Davison, A.J.: In-place scene labelling and understanding with implicit scene representation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 15838–15847 (2021)
43. Zhu, B., Yang, Y., Wang, X., Zheng, Y., Guibas, L.: Vdn-nerf: Resolving shape-radiance ambiguity via view-dependence normalization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 35–45 (2023)
44. Zhu, J., Huo, Y., Ye, Q., Luan, F., Li, J., Xi, D., Wang, L., Tang, R., Hua, W., Bao, H., Wang, R.: I^2-sdf: Intrinsic indoor scene reconstruction and editing via raytracing in neural sdfs. In: CVPR (2023)