Supplementary Material
Uni-SLAM: Uncertainty-Aware Neural Implicit SLAM for Real-Time Dense Indoor Scene Reconstruction

Shaoxiang Wang, Yaxu Xie, Chun-Peng Chang, Christen Millerdurai, Alain Pagani, Didier Stricker
German Research Center for Artificial Intelligence
firstname.lastname@dfki.de

Abstract

In the supplemental material, we provide additional details about the following:

- *Details on implementation. (Section A)*
- *More analysis and ablation study. (Section B)*
- *Per-Scene Breakdown of the Results. (Section C)*

A. Implementation Details

A.1. Hyperparameters

Default Setting. For scene representation, we set the hash grid size $L = 16$ for the geometry hash grid and $L = 16$ for the appearance hash grid. Default resolutions for both geometry and appearance are $0.02m$. Two tiny 2-layer decoders with 32 channels are applied to decode the color and the SDF. For the activation functions, ReLU is used in hidden layers, while Sigmoid and Tanh are applied to the output layers for raw color and SDF respectively. We use the Adam optimizer to optimize scene representation and decoder. The learning rate for the geometry hash grid is $5e^{-2}$, the learning rate for the appearance hash grid is also $5e^{-2}$, and the learning rate for both MLP decoders is $5e{-}3$. We sample $N_{str} = 32$ stratified points and $N_{imp} = 10$ points within the truncated distance $\tau_{tr} = 6cm$. Our pixel-level uncertainty threshold is $\beta_{unc_m} = 1e{-}2$, image-level uncertainty threshold is $\beta_{unc} = 1e{-}3$ and the co-visibility threshold is $OC_{cov} = 0.95$. We always optimize the camera pose during tracking and mapping if BA is enabled. The learning rate for camera pose rotation and translation is $1e{-}3$. The weights of the loss function are $\lambda_{rgb} = 5$, $\lambda_{dep} = 0.1$, $\lambda_{sdf_c} = 200$, $\lambda_{sdf_t} = 10$ and $\lambda_{sdf_{fs}} = 5$ for mapping, while $\lambda_{rgb} = 5$, $\lambda_{dep} = 1$, $\lambda_{sdf_c} = 200$, $\lambda_{sdf_t} = 50$ and $\lambda_{sdf_{fs}} = 10$ are set for tracking. For the tracking part, we perform the tracking process for every frame, select $M_t = 2000$ sampling points, and perform 8 iterations. For the mapping part, we select $M_m = 4000$ sampling points, perform 13 iterations every 4 frames and use a window of $W = 20$ keyframes for local bundle adjustment. At the start of training, we use 200 iterations for the first frame mapping. The reconstructed mesh is extracted by marching cubes algorithm [11]. To ensure a fair comparison, we do the same mesh culling strategy for all benchmark baselines following Neural-RGBD [1]. In order to present the reconstructed quality considering both tracking and mapping, the predicted camera poses are used for culling paths instead of ground truth poses.

Replica Dataset [13] We set $L = 19$ for the appearance hash grid. Replica dataset it contains eight synthetic scenes including 3D ground truth mesh. So based on its 3D ground truth mesh we can also evaluate our metrics on 3D evaluation, such as *Depth L1 [cm], Accuracy [cm], Reconstruction completion [cm]*, and *Completion ratio [< 1cm]%*. Those meshes are culled following [1] before evaluation.

ScanNet Dataset [4] We perform the mapping process every 5 frames, increasing the number of iterations to 20, $N_{str} = 48$. For tracking, iterations are increased to 20. Because of invalid depth at the edge of the image of ScanNet, 75 pixels are culled at the edge of the image for tracking during data pre-processing. The learning rate of translation is set to $5e^{-4}$, and the learning rate of rotation is $3e^{-3}$.

TUM RGB-D Dataset [14] The image-level uncertainty threshold is increased to $\beta_{unc} = 2e{-}3$. We perform a mapping process every 4 frames here and select $M = 4000$ sampling points for tracking and mapping. 20 pixels are culled at the edge of the image for tracking. The iteration of tracking is set to 20, while the iteration of mapping is also set to 20, $N_{str} = 48$. The learning rate of two hash grids is set to $2e^{-2}$. The learning rate of translation is set to $1e^{-2}$, and the learning rate of rotation is $5e^{-3}$.

A.2. Proof of Termination Probability

Our goal is to prove the accumulated termination probability along a current sampling ray r as:

$$p(r) = \sum_{n=1}^{N} w_n = 1$$

where N is the number of sampling points along the ray r, the weight w_n is defined as:

$$w_n = T_n \cdot (1 - \exp(-\sigma(p_n)))$$

where p_n is one sampling point along this ray, T_n is the transmittance of all previous sample points.

$$T_n = \exp\left(-\sum_{k=1}^{n-1} \sigma(p_k)\right)$$

First, we expand the weight w_n:

$$\sum_{n=1}^{N} w_n = \sum_{n=1}^{N} \left(\exp\left(-\sum_{k=1}^{n-1} \sigma(p_k)\right) \cdot (1 - \exp(-\sigma(p_n)))\right)$$

Second, introduce a recursive relationship for transmittance. We know that the relationship between T_n and T_{n+1} is:

$$T_{n+1} = T_n \cdot \exp(-\sigma(p_n))$$

So we can expand term by term and see the pattern:

$$T_1 = 1$$
$$T_2 = \exp(-\sigma(p_1))$$
$$T_3 = \exp(-\sigma(p_1)) \cdot \exp(-\sigma(p_2)) = \exp(-\sigma(p_1) - \sigma(p_2))$$

Thus, for any n :

$$T_n = \exp\left(-\sum_{k=1}^{n-1} \sigma(p_k)\right)$$

According to Equation:

$$\sum_{n=1}^{N} w_n = \sum_{n=1}^{N} \left(\exp\left(-\sum_{k=1}^{n-1} \sigma(p_k)\right) \cdot (1 - \exp(-\sigma(p_n)))\right)$$

Look at it item by item:

$$\begin{aligned}
w_1 &= T_1 \cdot (1 - \exp(-\sigma(p_1))) \\
&= 1 \cdot (1 - \exp(-\sigma(p_1))) \\
&= 1 - \exp(-\sigma(p_1)) \\
w_2 &= T_2 \cdot (1 - \exp(-\sigma(p_2))) \\
&= \exp(-\sigma(p_1)) \cdot (1 - \exp(-\sigma(p_2))) \\
&= \exp(-\sigma(p_1)) - \exp(-\sigma(p_1) - \sigma(p_2)) \\
w_3 &= T_3 \cdot (1 - \exp(-\sigma(p_3))) \\
&= \exp(-\sigma(p_1) - \sigma(p_2)) \cdot (1 - \exp(-\sigma(p_3))) \\
&= \exp(-\sigma(p_1) - \sigma(p_2)) - \exp(-\sigma(p_1) - \sigma(p_2) - \sigma(p_3))
\end{aligned}$$

Continuing in this way, we can discover the structure of each item:

$$\begin{aligned}
\sum_{n=1}^{N} w_n = &(1 - \exp(-\sigma(p_1))) \\
&+ (\exp(-\sigma(p_1)) - \exp(-\sigma(p_1) - \sigma(p_2))) \\
&+ (\exp(-\sigma(p_1) - \sigma(p_2)) \\
&- \exp(-\sigma(p_1) - \sigma(p_2) - \sigma(p_3))) \\
&+ \cdots \\
&+ \left(\exp\left(-\sum_{k=1}^{N-1} \sigma(p_k)\right) - \exp\left(-\sum_{k=1}^{N} \sigma(p_k)\right)\right)
\end{aligned}$$

All the intermediate terms cancel each other out, leaving only the first and last terms:

$$\sum_{n=1}^{N} w_n = 1 - \exp\left(-\sum_{k=1}^{N} \sigma(p_k)\right)$$

As N tends to infinity, assuming all densities are cumulative in the observed regions, the exponential part of the last term tends to negative infinity, then:

$$\exp\left(-\sum_{k=1}^{N} \sigma(p_k)\right) \approx 0$$

So,

$$\sum_{n=1}^{N} w_n = 1 - 0 = 1$$

By the above steps, we have proved that the cumulative sum of all weights w_n on a ray for observed area is equal to 1. However, in unobserved regions where the density values $\sigma(p_k)$ are very small or zero, the exponential term will tend to 1, so

$$\sum_{n=1}^{N} w_n = 1 - 1 = 0$$

Therefore, the termination probability is proven to lie within the range $(0, 1)$.

A.3. Co-visibility Check

Loop detection is implemented based on sample point remapping. We sample $M = 50$ pixels for every keyframe in the keyframes database, sample $N = 8$ sample points along each ray, given the camera's internal and external parameters, and map these points back to the current frame. If the overlap coefficient is greater than 0.95, we consider that a loop closure has occurred. In order to avoid too short a time interval and too short a range of motion for loop closure detection, we set a minimum threshold of 100 frames between the two points where a loop closure occurs.

B. More Analysis and Ablation Study

B.1. Hash Grid Size Analysis

To investigate the distinct requirements of geometry and appearance for spatial representation, we conduct our experiments on the synthetic Replica dataset. We evaluate different hash grid size combinations to investigate the sensitivity of appearance and geometry to hash grid size in Tab. 1, while Tab. 2 compares the impact of hash grid size on model size and speed in frame per second(FPS). We compare the results with BSLAM [8],Co-SLAM [15] and ESLAM [9] at index 3, index 5, index 7 respectively. In these plots, the numbers in parentheses (h_g, h_a) report the geometry hash grid size and appearance hash grid size respectively. Experiments show that the reconstruction and rendering quality can be further improved by increasing the hash grid size. However, for equal model sizes, allocating more memory to appearance yields more benefits on rendering quality and completeness (compare the combination of index 4 $(h_g = 16, h_a = 19)$ and index 9 $(h_g = 19, h_a = 16)$). We interpret this phenomenon by considering that *color information is a higher-frequency signal compared to geometric information*. The implication here is that when computational resources are limited, we should allocate more resources to the appearance signal. In terms of the relation between hash grid size and FPS, it is worth noting that when increasing the hash grid size combination from $(h_g = 16, h_a = 19)$ to $(h_g = 22, h_a = 22)$, the speed in FPS only decreases from 8.3 fps to 6.6 fps.

B.2. Strategic BA Analysis

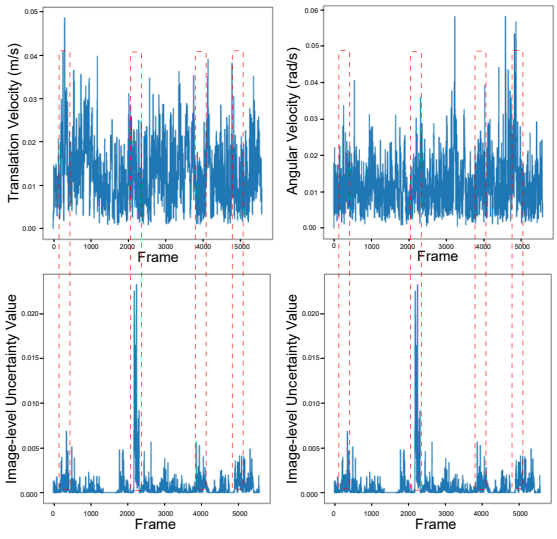
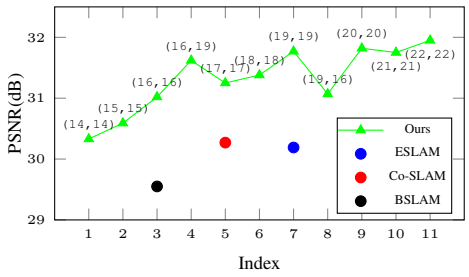


Figure 1. **Impact of Translational and Angular Velocities on Uncertainty.** We can observe the correlation between uncertainty and both translational velocity and angular velocity. Higher velocities or accelerations tend to result in higher uncertainty.

(a) Impact on PSNR.



(b) Impact on Completion Rate.

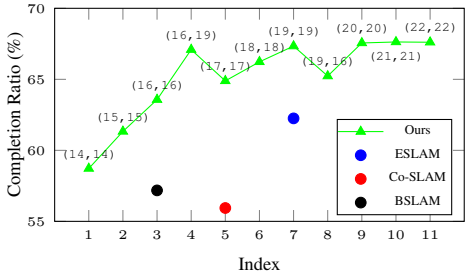
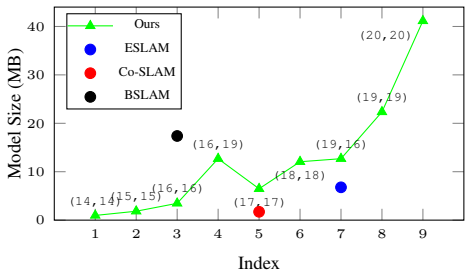


Table 1. Impact of `(SDF hash grid size, Appearance hash grid size)` on PSNR [dB] and Completion Rate [$< 1cm\%$] on the Replica dataset.

(a) Impact on Model Size



(b) Impact on FPS.

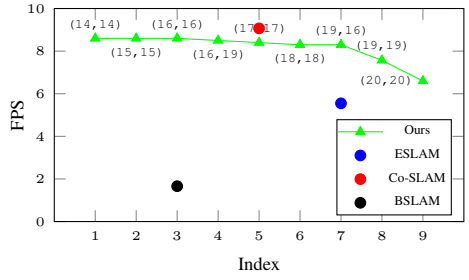


Table 2. Impact of `(SDF hash grid size, Appearance hash grid size)` on Model Size and FPS on the Replica dataset.

Uncertainty vs. Velocity. To analyze the relationship between velocity and uncertainty, we conducted experiments on `scene0000` from ScanNet [4].

The camera's motion state is described in terms of translation and rotation $\{T_i|R_i\}$. In Fig. 1, we visualize the translational velocity and angular velocity, with the corresponding image-level uncertainty displayed below each. The results show that higher velocities or accelerations can easily cause the camera to move into unseen areas before, leading to increased uncertainty. This figure exposes the relationship between our definition of uncertainty and the state of camera motion, justifying our definition of uncertainty.

Impact of Strategic BA on Uncertainty. We investigated the impact of using strategic Bundle Adjustment (BA) on image-level uncertainty on `scene0000` from ScanNet [4]. As shown in Fig. 2, using only constant global BA results in high uncertainty, as indicated by the orange line. Similarly, the green line represents high uncertainty with only local BA. The red line shows suboptimal results when using global BA and local BA without local loop closure optimization (LLCO). However, with our full strategic BA the uncertainty could be reduced significantly on average as shown in blue line. This implies more accurate localization and improved rendering. Further reduction in uncertainty demonstrates the effectiveness of our LLCO approach. In Fig. 3, we present the visual results. We visualize rendered image, depth uncertainty, and pixel-level uncertainty in three rows respectively. It is evident that under strategic BA, the quality of rendered images is noticeably enhanced, and the corresponding depth uncertainty is also lower, indicating higher geometric quality. The depth uncertainty is calculated as follows:

$$\hat{d}_{unc} = \sqrt{\sum_{i=1}^{N} w_i \left(\hat{\boldsymbol{d}} - d_i\right)^2} \tag{1}$$

where w_i is the weight corresponding to Equation(2) in main paper, $\hat{\boldsymbol{d}}$ is predicted depth, and d_i represents the distance from the camera center to the current sample point $\mathbf{x}_i$ along this ray. Pixel-level uncertainty in the third column is also lower with this strategy.

Plug-in Capability. The effectiveness of our strategy has also been validated on BSLAM [8]. Based on image-level uncertainty and co-visibility check, we dynamically activate an additional mapping process beyond the global BA. The results in Tab. 3 show improvements in all metrics, benefiting from our uncertainty-aware strategy. This demonstrates the plug-in capability of our approach.

B.3. Ablation on Model Design

In order to justify our choice of a model-free uncertainty model, we conduct also experiments with a learnable uncertainty model. As shown in Fig. 4, in addition to using two sparse grids to represent geometry and appearance separately, we use a third grid to model depth uncertainty

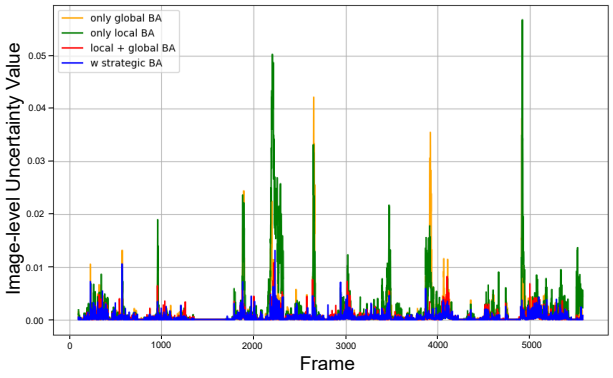


Figure 2. **Impact of different keyframe selection on Uncertainty.** Here we compare the changing image-level uncertainty per frame with different keyframe selection strategies. The results indicated by the blue line show that image-level uncertainty is significantly reduced, achieving optimal outcomes with our proposed strategic BA (local BA + global BA + LLCO).

Table 3. Analysis of the impact of our strategic BA on BSLAM [8] (Sec. 3.4 in the main paper). The experiment is conducted on Replica [13], and the metrics are ATE RMSE (cm), reconstruction accuracy (cm), reconstruction completion (cm), completion ratio and PSNR. BSLAM [8] can also benefit from our strategy.

Method	ATE (cm)↓	Acc. $(cm)\downarrow$	Comp. Ratio $[< 1cm\%]\uparrow$	PSNR (dB)↑
BSLAM [8]	1.19	1.12	57.18	29.55
BSLAM w/ Our strategic BA	1.07	1.01	58.36	29.83

based on the Gaussian assumption inspired by [6]. For depth uncertainty, a model posterior assumption is made from the Bayesian perspective, similar to Bayes' Rays [7]. Our experiments show that this idea not only brings undesirable increased model complexity, making the model much slower, but also leads to poorer results in terms of reconstruction quality (corresponding to main paper Sec. 4.3).

The following paragraph explains how we design learnable uncertainty to reweight the depth term loss function.

Gaussian Assumption Uncertainty: Assume that the residuals (errors) between the estimated depth $\hat{d}$ and the true depth D follow a Gaussian distribution with variance σ^2:

$$\hat{d} \sim \mathcal{N}(D, \sigma^2) \tag{2}$$

The probability density function (PDF) of a normal distribution is given by:

$$p(\hat{d}|D, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(\hat{d} - D)^2}{2\sigma^2}\right) \tag{3}$$

To maximize the likelihood, we equivalently minimize the

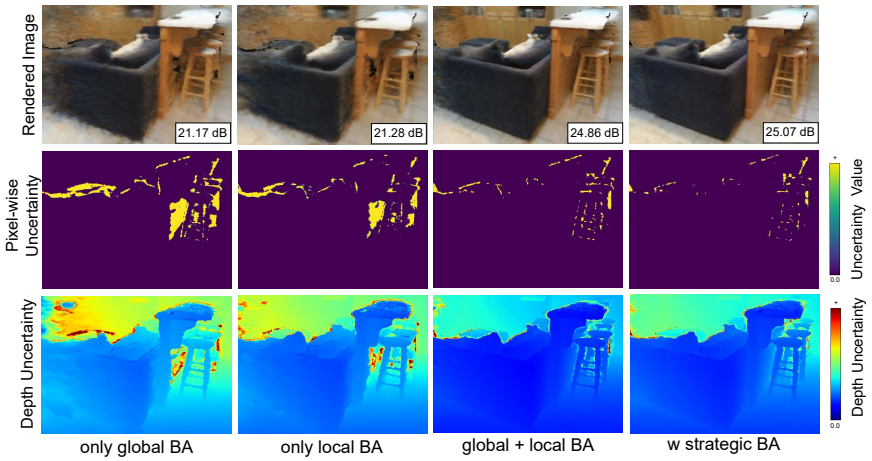


Figure 3. **Impact of Strategic BA on Rendering and Uncertainty Visualization.** Our proposed strategic BA integrates global BA, local BA, and LLCO. This approach achieves the highest rendered image quality, as indicated by the PSNR (dB) metric. The second row presents visualized pixel-level uncertainty, while depth uncertainty illustrates geometric reconstruction in the third row. The depth uncertainty, defined in Eq. (1), shows a continuous variation in visualized uncertainty, providing a clearer demonstration of the superiority of our approach.

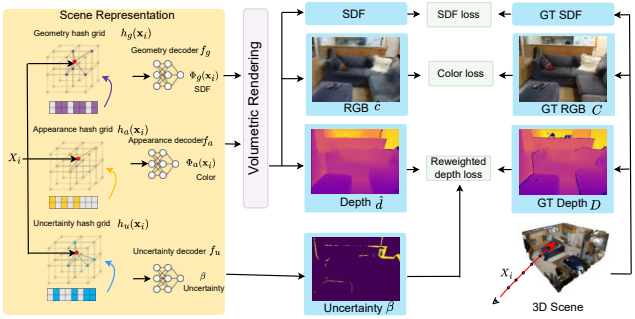


Figure 4. **Ablation on Gaussian Assumption Uncertainty Model.** We use three grids to represent geometry, appearance, and learnable uncertainty respectively.

negative log-likelihood. The negative log-likelihood for a single observed ray is given by:

$$-\log p(\hat{d}|D,\sigma^2) = \frac{(\hat{d}-D)^2}{2\sigma^2} + \frac{1}{2}\log(2\pi\sigma^2) \quad (4)$$

For simplicity, we often drop the constant term $\frac{1}{2}\log(2\pi)$ since it does not affect the optimization. Here we let $\beta = \sigma^2$. In practice, we work with an estimate of the variance β through a third grid parallel with the geometry and appearance grid. So, the term we need to minimize is:

$$\mathcal{L}_{\text{single}} = \frac{(\hat{d}-D)^2}{2\sigma^2} + \frac{1}{2}\log\sigma^2 = \frac{(\hat{d}-D)^2}{2\beta} + \frac{1}{2}\log\beta \quad (5)$$

If we have a set of depth measurements R_d, we sum the negative log-likelihoods for all rays r in the set R_d. Additionally, we normalize by the number of elements $|R_d|$ to get the average loss:

This matches the given loss function:

$$\mathcal{L}_d = \frac{1}{|R_d|}\sum_{r\in R_d}\left(\frac{1}{2\beta}(\hat{d}_r - D_r)^2 + \frac{1}{2}\log\beta\right) \quad (6)$$

The first term $\frac{(\hat{d}-D)^2}{2\beta}$ penalizes large errors more if the predicted uncertainty β is small. The second term $\frac{1}{2}\log\beta$ prevents the model from predicting an arbitrarily large uncertainty to minimize the first term. By balancing these two terms, the loss function encourages the model to provide both accurate depth estimates and reasonable uncertainty estimates.

Dataset	Method	Tracking/Rendering RMSE [cm] ↓	PSNR [dB] ↑	FPS ↑	params ↓
Replica [13]	Gaussian	1.175	27.27	7.06	14.65M
	Ours	**0.45**	**31.62**	**8.37**	**12.69M**
ScanNet [4]	Gaussian	11.93	18.06	3.57	5.13M
	Ours	**7.01**	**21.77**	**4.88**	**3.39M**
TUM RGB-D [14]	Gaussian	2.16	19.30	1.73	5.38M
	Ours	**2.05**	**21.23**	**2.72**	**3.58M**

Figure 5. **Gaussian Assumption Model vs. Ours.**

Our system demonstrates superior tracking accuracy and rendering quality compared to SLAM systems that rely on Gaussian assumptions as shown in Fig. 5. Additionally, our system outperforms in terms of speed and parameter efficiency. Under the Gaussian assumption, depth uncertainty is typically modeled using an additional hash grid for separate estimation. This introduces extra variables that need optimization, which introduces further complexities and disturbances in the SLAM system.

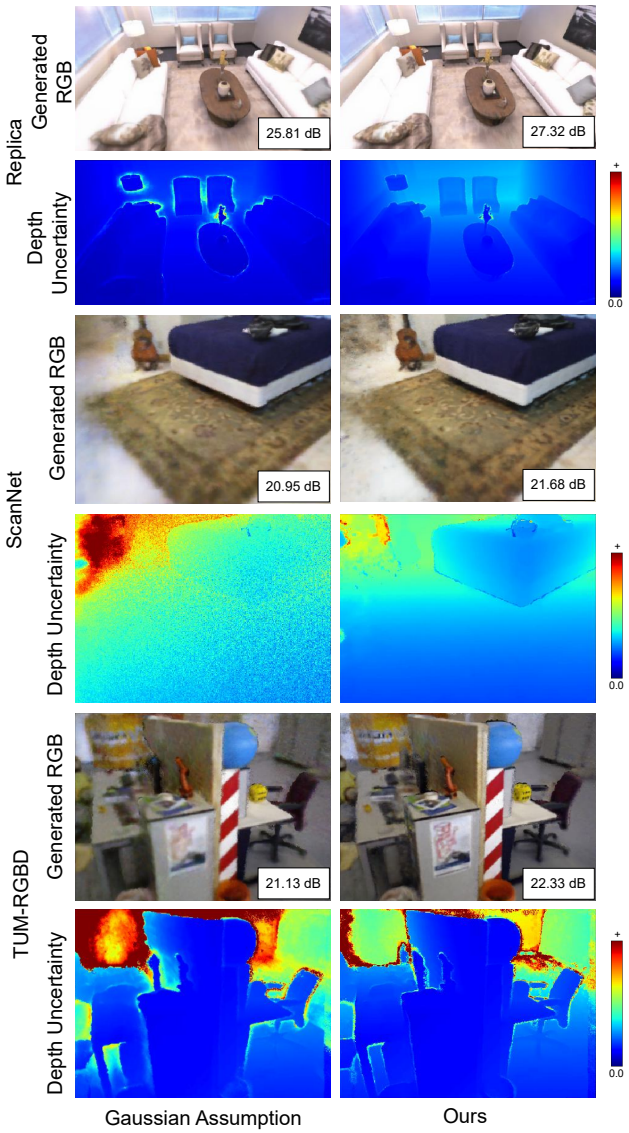


Figure 6. **Gaussian Assumption Model vs. Ours.** Our model demonstrates superior rendering quality, as evaluated by PSNR (dB) ↑. Depth uncertainty, calculated using Eq. (1), is visualized for comparison. Our method visibly reduces depth uncertainty, as clearly shown in the visualizations.

In Fig. 6, we conducted a comparison of rendering quality and depth uncertainty between the two methods across three datasets. The superiority of our approach is evident, particularly on real-world datasets such as TUM-RGBD [14] and ScanNet [4], where the visualized depth uncertainty clearly highlights the advantages of our method.

Moreover, in addition to aboving scene representation, we also experimented with the memory-efficient tri-plane [2] method for encoding geometry and appearance respectively. In Tab. 6, rows a) through d) provide quantitative results on the Replica dataset, while Fig. 12 presents the corresponding qualitative visualizations. The results show that using two hash grids for encoding provides the best performance.

B.4. Model Capability Analysis

To demonstrate the high capability of our model in reconstructing quality scenes and to fairly compare the model's upper limits, we compared our method with state-of-the-art dense implicit SLAM approaches, including ESLAM [9] and Co-SLAM [15] on Replica dataset [13]. We standardized the mapping iterations and tracking iterations to 30, and set the number of sampling points to 5000. The results in Tab. 4 indicate that our method achieves superior performance in terms of evaluation metrics localization accuracy ATE RMSE, reconstruction accuracy, completion ratio, PSNR, and computational efficiency.

Method	ATE (cm)↓	Acc. (cm) ↓	Comp. Ratio $[< 1cm\%]$ ↑	PSNR (dB) ↑	Time Mins ↓
ESLAM [9]	0.40	0.91	63.51	31.63	21.53
Co-SLAM [15]	0.75	1.07	57.79	31.77	11.92
ours	**0.29**	**0.84**	**68.35**	**32.82**	**11.17**

Table 4. Capability analysis of the effect of the number of optimization iterations during mapping and tracking on our method's reconstruction quality.

B.5. Model Convergence Speed Analysis

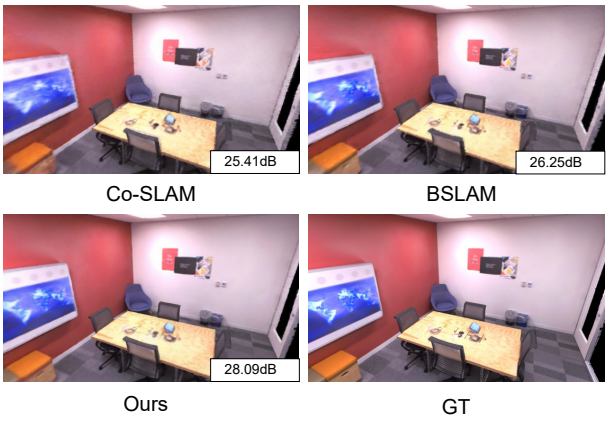


Figure 7. **Rendering Comparison on Replica dataset [13].** Ours shows the best rendering quality compared to state-of-the-art methods BSLAM [8] and Co-SLAM [15] among dense implicit SLAM methods. Please zoom in for details.

To compare model convergence speed and rendering quality, we conducted experiments on the synthetic Replica

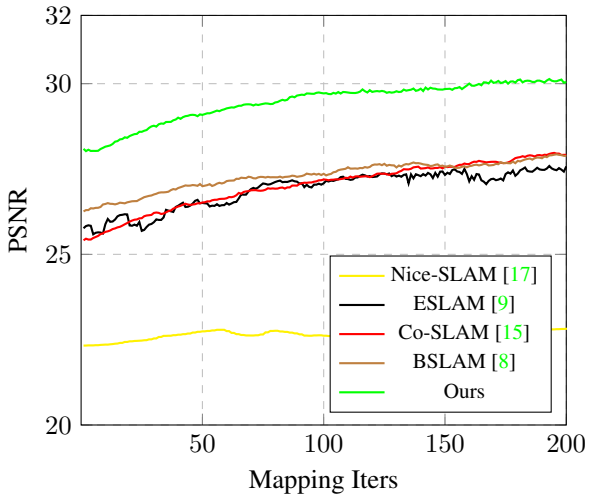


Figure 8. Comparative rendering quality convergence on the Replica dataset [13]. We set mapping iterations to 200 steps for one frame and recorded PSNR at each iteration. Our model showed stable, monotonic growth in PSNR, attributed to its decoupled scene representation. In contrast, ESLAM exhibits higher variance, and Nice-SLAM, Co-SLAM, and BSLAM have lower PSNR values, indicating slower convergence and poorer performance.

dataset and the realistic TUM RGB-D dataset. Fig. 7 and Fig. 8 illustrate the qualitative rendering quality and quantitative changes over iterations on the Replica dataset. Our model exhibited the best rendering quality with a stable, monotonically increasing curve, attributed to its decoupled grid-based scene representation. On the real-world TUM RGB-D dataset, as shown in Fig. 9 and Fig. 10 our model also outperformed Nice-SLAM, ESLAM, Co-SLAM, and BSLAM. The other models showed instability (e.g., ESLAM on Replica, Co-SLAM on TUM-RGBD) and suboptimal rendering quality.

B.6. Runtime and Memory Analysis

In Tab. 5, we compare runtime and memory usage, benchmarking all methods on NVIDIA GeForce RTX 4090 GPU using `room0` of Replica [13], `scene0000` of ScanNet [4] and `freiburg2-xyz` of TUM-RGBD [14]. We report tracking and mapping times per iteration and compare iteration steps to show convergence speed. The results show that our method achieved competitive real-time performance compared to Co-SLAM.

B.7. Ablation on Reweighting Term

Here, corresponding to Section 4.3 of the main paper, we provide further explanation of the reweighting term to validate our choice. In the tracking and mapping processes, the loss functions consist of three loss terms:

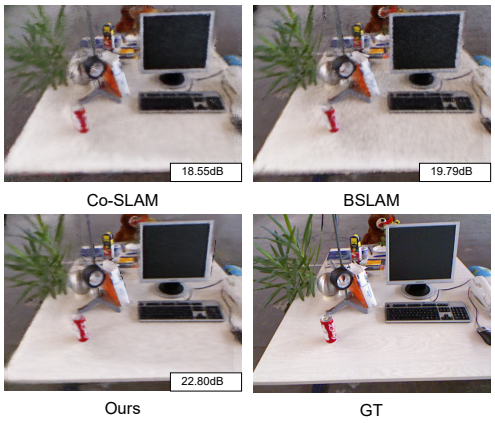


Figure 9. **Rendering Comparison on TUM RGB-D [14].** Ours shows the best results compared to state-of-the-art methods BSLAM [8] and Co-SLAM [15] among dense implicit SLAM methods.

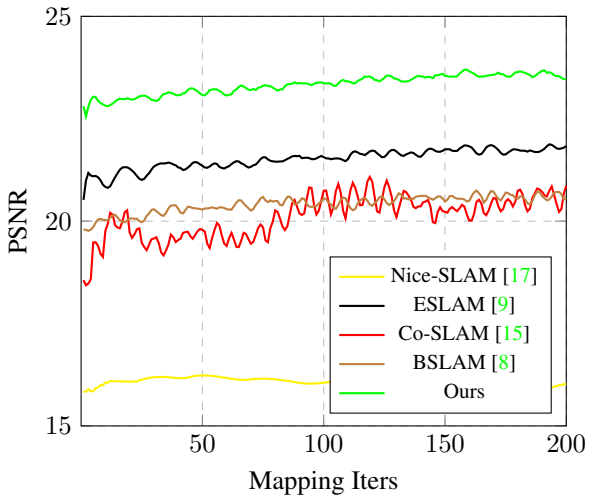


Figure 10. Comparative rendering quality convergence on TUM RGB-D [14]. We set the mapping process iterations to 200 steps and recorded the PSNR for each iteration. The variation curve shows a stable monotonic increase, demonstrating the model's stability on real-world challenging datasets. In contrast, Co-SLAM's variation curve oscillated, reflecting poorer stability in rendering. Meanwhile, Nice-SLAM, ESLAM, and BSLAM showed suboptimal results due to insufficient model capability and slower convergence.

($\mathcal{L}_{sdf}$, $\mathcal{L}_{dep}$, $\mathcal{L}_{rgb}$). We aim to use pixel-level uncertainty to select effective information and progressively filter out outliers to enhance localization accuracy and rendering quality. If reweighting is applied, we denote it as Y, and if not, we denote it as N. For example, $YYY - YYN$ means, we reweight all ($\mathcal{L}_{sdf}$, $\mathcal{L}_{dep}$, $\mathcal{L}_{rgb}$) three terms in tracking process, and only reweight ($\mathcal{L}_{sdf}$, $\mathcal{L}_{dep}$) in

	Method	Tracking [ms x it.] ↓	Mapping [ms x it.] ↓	FPS↑	Time Mins↓	params.↓
Replica	Nice-SLAM [17]	6.5 x 10	29.3 x 0	1.8	18.51	12.13M
	Co-SLAM [15]	**4.6 x 10**	**6.6 x 10**	**9.07**	**3.67**	**1.72M**
	ESLAM [9]	7.9 x 8	18.8 x 15	5.55	6.01	6.78M
	BSLAM [12]	11 x 20	15 x 20	1.66	20.3	17.38M
	Ours	7.0 x 8	8.1 x 13	8.37	4.02	12.69M
ScanNet	Nice-SLAM [17]	11.3 x 50	41.2x60	1.34	57.8	22.04M
	Co-SLAM [15]	**5.6 x 20**	**12.7 x 10**	**5.7**	**17.2**	**1.74M**
	ESLAM [9]	13.41 x 30	22.5 x 30	1.57	40.6	17.63M
	BSLAM [12]	250 x 20	400 x 20	0.52	176	18.5M
	Ours	6.3 x 20	11.7 x 30	4.88	20.8	3.39M
TUM RGB-D	Nice-SLAM [17]	33 x 200	103 x 60	0.09	577	120.95M
	Co-SLAM [15]	**4.3 x 20**	**15.6 x 10**	**6.4**	**8.5**	**1.68M**
	ESLAM [9]	20.5 x 200	22.3 x 60	0.33	175	9.51M
	BSLAM [12]	251 x 20	370 x 20	0.95	59	19.76M
	Ours	12.3 x 20	13.7 x 20	2.7	21.3	3.58M

Table 5. Runtime and Memory Usage Comparison.

mapping process.

As shown in Fig. 11, column d) yields the optimal results. Not only does it produce the highest quality rendered color image (highest PSNR [dB]), but the pixel-level uncertainty map and the depth uncertainty map also demonstrate higher quality depth information estimation. Compared to column e), where we do not apply reweighting to the color loss term during the mapping process, our approach compensates effectively for invalid depth caused by the sensor itself, resulting in finer geometric reconstruction.

C. Per-Scene Breakdown of the Results.

In this section, we provide more per-scene qualitative and quantitative results. Tab. 7 and Tab. 8 present the quantitative results for 3D and 2D metrics on the Replica dataset [13] for each scene, respectively. Figs. 13 to 16 show the qualitative reconstructed meshes. The results for Nice-SLAM [17], Co-SLAM [15], ESLAM [9], and BSLAM [8] are obtained using their open-source code over five experimental runs. For PLG-SLAM [5], the authors only provide us the reconstructed meshes on the Replica dataset, so the qualitative comparison is not provided here. Additionally, although this paper primarily investigates the application of uncertainty in real-time implicit NeRF-SLAM, for a broader qualitative comparison of reconstruction quality, we also include explicit scene representations, such as Loopy-SLAM [10]. Overall, the results demonstrate that our method achieves finer reconstructions among all implicit methods while addressing the hole-filling limitations of explicit scene representations. For real-world datasets, Fig. 17 shows the reconstruction results on ScanNet [4], and Figs. 18 to 20 display our reconstruction results on TUM RGB-D [14]. These results indicate that our method achieves more precise detail reconstruction and high-fidelity rendering, which we attribute to robust scene representation and an uncertainty-aware strategy.

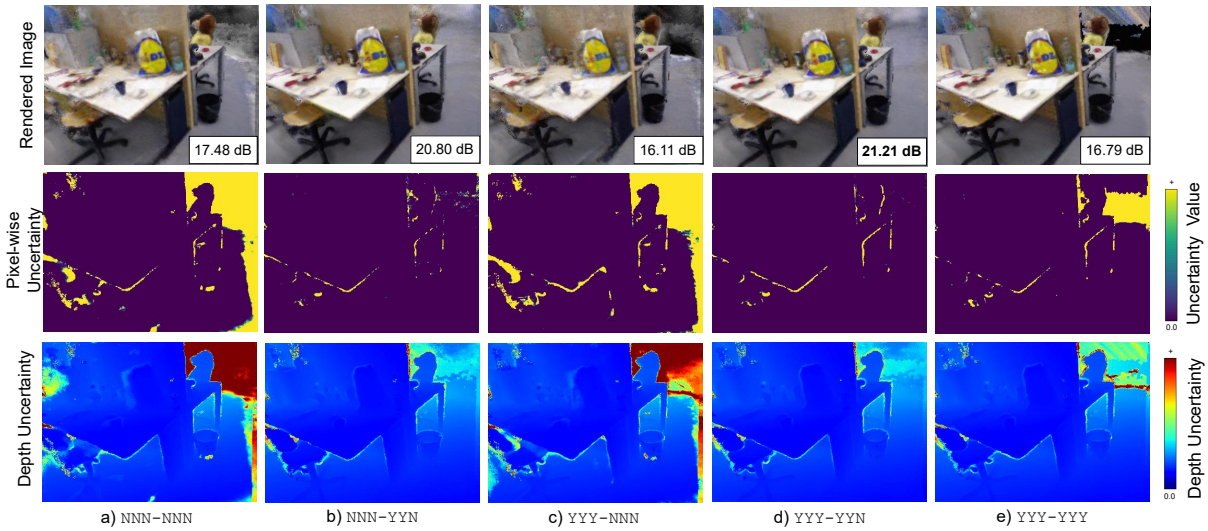


Figure 11. **Ablation on Reweighting.** In the tracking and mapping processes, the loss functions consist of three loss terms: ($\mathcal{L}_{sdf}$, $\mathcal{L}_{dep}$, $\mathcal{L}_{rgb}$). If reweighting is applied, we denote it as Y, and if not, we denote it as N. Column d) YYY-YYN indicates that we apply pixel-level uncertainty reweighting to all terms except for the color loss term $\mathcal{L}_{rgb}$ in the mapping process. With this uncertainty-guided reweighting strategy, we achieve the best rendering quality and depth estimation.

Algorithm 1 Our Uncertainty-Aware Algorithm

$i = 1$ ▷ Initialize index
P ▷ Estimated camera pose
n ▷ Fixed-frequency for constant global BA
N ▷ Number of frames of current RGB-D sequence
θ ▷ Scene representation
$Optimize()$ ▷ Optimazation function with pixel-level uncertainty reweighting
while $i < N$ **do**
  if $i = 1$ **then**
    $P_1 = P_1^{gt}$ ▷ Initialize first camera pose with ground truth
    Optimize (θ_1) ▷ Optimize scene representation at the first frame
    $i = i + 1$
  end if
  if $i > 1$ **then**
    Optimize (P_i) ▷ Tracking process for each frame
  end if
  if $\beta > \beta_{unc}$ **then** ▷ Uncertainty check
    Optimize $(\theta_{local}, P_{local})$ ▷ Local BA
    $i = i + 1$
  else if $OC_{cov} > \tau_{cov}$ **then** ▷ Co-Visibility check
    Optimize $(\theta_{LLCO}, P_{LLCO})$ ▷ Local loop closure optimization
    $i = i + 1$
  end if
  if $i \bmod n == 0$ **then**
    Optimize $(\theta_{global}, P_{global})$ ▷ Global BA for every n frame
    $i = i + 1$
  end if
end while

Methods	Reconstruction & Rendering				Localization [cm]
	Acc.	Comp. Ratio	Depth L1	PSNR	RMSE
a) Gaussian assumption uncertainty with third grid	1.79	31.52	3.75	27.33	1.51
b) Coupled scene representation with one grid	1.05	63.15	0.94	30.12	0.51
c) Grid for geometry and tri-plane for appearance	1.01	64.69	0.93	30.98	0.47
d) Tri-plane for geometry and grid for appearance	1.17	63.82	0.97	21.32	0.50
e) w/o camera pose optimization in mapping	1.89	26.88	1.76	27.56	3.52
f) Only global BA in mapping	0.96	66.01	0.95	31.32	0.49
g) Only local BA in mapping	1.01	65.21	0.91	30.87	0.55
h) Global + local BA in mapping	0.94	66.34	0.89	31.51	0.45
Ours	**0.92**	**66.86**	**0.89**	**31.62**	**0.45**

Table 6. We conduct experiments on Replica [13] to verify the effectiveness of our method. Our full model achieves better completion reconstructions and more accurate pose estimation results.

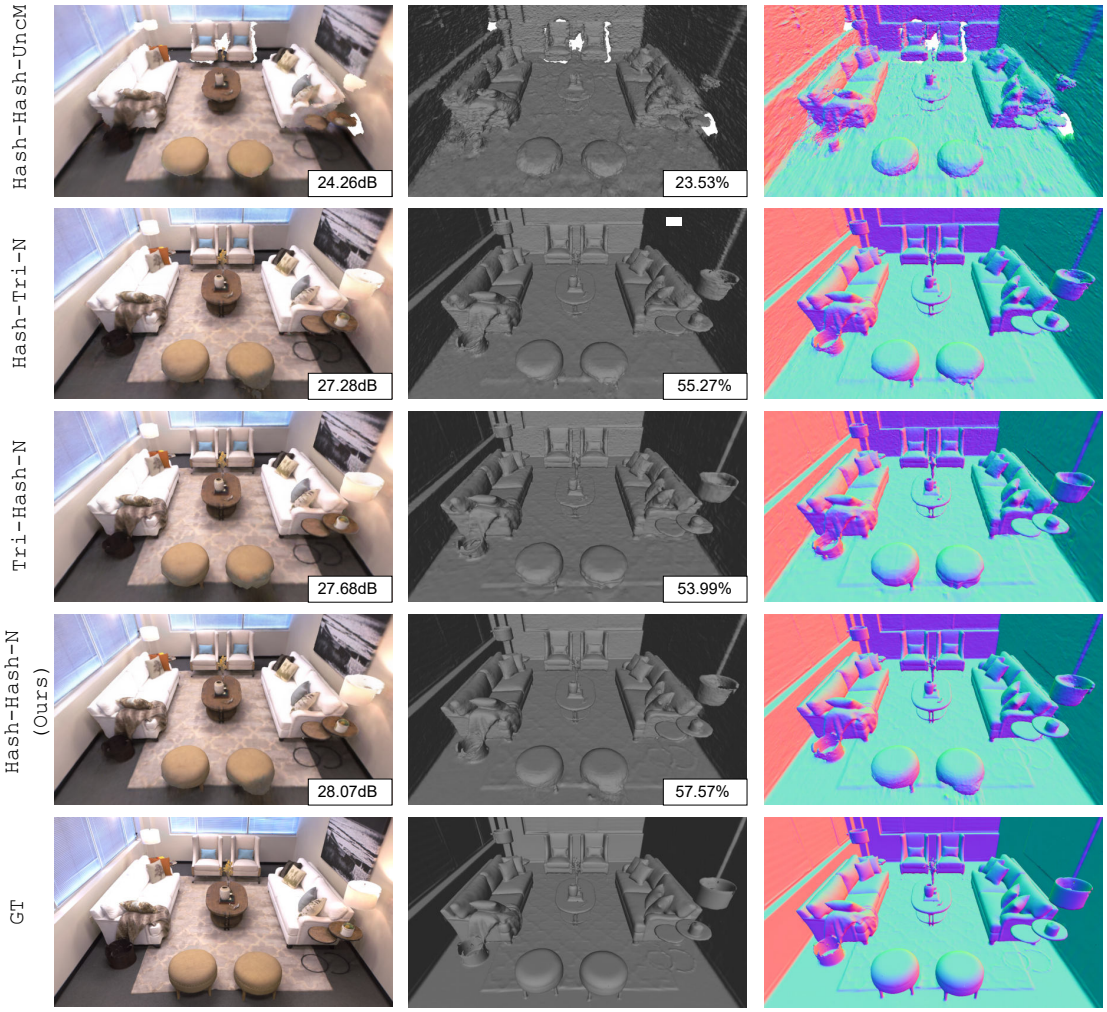


Figure 12. **Ablation on Model Design.** We compare different scene representation combinations on Replica [13] `room0` and evaluate with metrics PSNR and completion ratio[$< 1cm\%$]. `Hash-Hash-UncM` denotes using hash grids for geometry and appearance, with a learnable uncertainty model. `Hash-Tri-N` uses a hash grid for geometry, a tri-plane for appearance, and our proposed model-free method for uncertainty estimation. The results show that using hash grids for both geometry and appearance, combined with the model-free uncertainty definition, achieves the best results.

		room0	room1	room2	office0	office1	office2	office3	office4	Avg.
Nice-SLAM [17]	Depth L1 [cm] ↓	2.51	2.65	3.37	2.12	2.20	4.53	4.30	3.79	3.18
	Acc. [cm] ↓	1.51	1.44	1.62	1.34	1.02	1.71	2.02	4.55	1.90
	Comp. [cm] ↓	1.50	1.39	1.54	1.42	1.08	1.57	1.82	1.94	1.53
	Comp. Ratio [$< 5cm\%$] ↑	98.33	98.81	97.37	97.6	98.08	97.65	95.81	95.92	97.45
	Comp. Ratio [$< 3cm\%$] ↑	95.20	95.30	91.45	94.82	95.52	92.91	90.30	88.10	92.95
	Comp. Ratio [$< 1cm\%$] ↑	32.63	39.07	35.17	42.37	67.39	31.22	24.07	23.48	36.93
Co-SLAM [15]	Depth L1 [cm] ↓	1.51	2.38	3.00	1.51	1.46	2.68	2.81	1.85	2.15
	Acc. [cm] ↓	1.11	1.33	1.22	0.99	0.71	1.36	1.29	1.24	1.16
	Comp. [cm] ↓	1.04	1.30	1.18	0.90	0.71	1.29	1.35	1.15	1.12
	Comp. Ratio [$< 5cm\%$] ↑	98.84	99.05	97.85	98.52	98.62	97.52	98.65	97.12	98.27
	Comp. Ratio [$< 3cm\%$] ↑	97.82	97.15	94.45	97.87	97.57	96.28	95.89	94.46	96.44
	Comp. Ratio [$< 1cm\%$] ↑	54.69	40.08	55.47	71.35	87.41	46.93	39.21	52.35	55.94
ESLAM [9]	Depth L1 [cm] ↓	0.97	1.07	1.28	0.86	1.26	1.71	1.43	1.06	1.18
	Acc. [cm] ↓	1.07	0.85	0.93	0.85	0.83	1.02	1.21	1.15	0.97
	Comp. [cm] ↓	1.12	0.88	1.05	0.96	0.81	1.09	1.42	1.27	1.05
	Comp. Ratio [$< 5cm\%$] ↑	99.06	99.64	98.84	98.34	98.85	98.60	96.80	97.65	98.47
	Comp. Ratio [$< 3cm\%$] ↑	98.84	99.24	96.73	97.89	98.02	98.02	96.31	96.54	97.70
	Comp. Ratio [$< 1cm\%$] ↑	53.06	70.27	62.15	73.11	84.13	59.32	46.93	49.06	62.25
BSLAM [8]	Depth L1 [cm] ↓	1.44	1.43	3.05	1.64	1.95	4.18	4.10	2.43	2.52
	Acc. [cm] ↓	1.02	0.92	1.01	0.86	0.69	1.46	1.75	1.27	1.12
	Comp. [cm] ↓	1.05	0.94	1.15	0.91	0.76	1.34	1.39	1.26	1.1
	Comp. Ratio [$< 5cm\%$] ↑	99.48	99.69	98.22	98.97	99.27	98.8	98.28	99.28	98.99
	Comp. Ratio [$< 3cm\%$] ↑	98.53	98.74	94.98	97.64	97.68	94.46	95.57	97.71	96.91
	Comp. Ratio [$< 1cm\%$] ↑	54.64	65.52	56.17	71.43	84.26	46.52	39.97	40.61	57.18
Ours	Depth L1 [cm] ↓	**0.81**	**0.77**	**1.13**	**0.70**	**1.11**	**1.52**	**1.15**	**0.99**	**0.89**
	Acc. [cm] ↓	**0.97**	**0.78**	**0.85**	**0.76**	**0.62**	**0.92**	**1.10**	**1.15**	**0.92**
	Comp. [cm] ↓	**0.99**	**0.78**	**0.93**	**0.77**	**0.67**	**0.93**	**1.18**	**1.13**	**0.92**
	Comp. Ratio [$< 5cm\%$] ↑	**99.69**	**99.84**	**99.21**	**99.21**	**99.25**	**99.19**	**98.25**	**98.99**	**99.20**
	Comp. Ratio [$< 3cm\%$] ↑	**99.15**	**99.47**	**96.75**	**98.96**	**98.15**	**97.75**	**97.12**	**96.75**	**98.01**
	Comp. Ratio [$< 1cm\%$] ↑	**57.57**	**74.99**	**69.53**	**76.76**	**88.18**	**62.78**	**50.91**	**54.19**	**66.86**

Table 7. Per-scene quantitative reconstruction evaluation on Replica [13] dataset. Our method achieves consistently better reconstruction in comparison to Nice-SLAM [17], Co-SLAM [15], ESLAM [9] and BSLAM [8]. We report Depth L1, reconstruction accuracy, completion, and completion ratios of 5cm, 3cm and 1cm respectively, reflecting our advantages in reconstruction geometry in detail.

Method	Metric	Rm 0	Rm 1	Rm 2	Off 0	Off 1	Off 2	Off 3	Off 4	Avg.
Nice-SLAM [17]	PSNR [dB] ↑	22.12	22.47	24.52	29.07	30.34	19.66	22.23	24.94	24.42
	SSIM ↑	0.689	0.757	0.814	0.874	0.886	0.797	0.801	0.856	0.809
	LPIPS ↓	0.330	0.271	0.208	0.229	0.181	0.235	0.209	0.198	0.233
Vox-Fusion [16]	PSNR [dB] ↑	22.9	22.36	23.91	27.79	29.83	20.33	23.47	25.21	24.4
	SSIM ↑	0.683	0.751	0.798	0.857	0.876	0.794	0.803	0.847	0.801
	LPIPS ↓	0.303	0.269	0.234	0.241	0.184	0.243	0.213	0.199	0.236
Co-SLAM [15]	PSNR [dB] ↑	27.12	27.94	29.27	34.13	35.04	28.53	28.81	31.29	30.27
	SSIM ↑	0.908	0.900	0.935	0.962	0.970	0.939	0.942	0.957	0.939
	LPIPS ↓	0.316	0.293	0.258	0.207	0.191	0.257	0.222	0.227	0.246
ESLAM [9]	PSNR [dB] ↑	27.10	28.41	29.16	34.59	34.29	28.97	28.57	30.51	30.19
	SSIM ↑	0.914	0.910	0.938	0.966	0.963	0.946	0.948	0.948	0.942
	LPIPS ↓	0.295	0.294	0.240	0.178	0.208	0.239	0.194	0.295	0.243
BSLAM [8]	PSNR [dB] ↑	26.43	28.67	28.44	33.27	33.92	27.68	28.14	29.85	29.55
	SSIM ↑	0.902	0.9179	0.919	0.950	0.963	0.933	0.939	0.9438	0.9335
	LPIPS ↓	0.300	0.2523	0.2618	0.201	0.195	0.246	0.205	0.2274	0.2361
Ours	PSNR [dB] ↑	**28.07**	**30.16**	**30.87**	**36.35**	**35.62**	**29.98**	**30.06**	**31.85**	**31.62**
	SSIM ↑	**0.927**	**0.940**	**0.955**	**0.978**	**0.977**	**0.961**	**0.962**	**0.965**	**0.958**
	LPIPS ↓	**0.241**	**0.201**	**0.172**	**0.145**	**0.167**	**0.231**	**0.156**	**0.169**	**0.185**

Table 8. Per-scene quantitative rendering evaluation on Replica [13]. Our method achieves consistently better rendering in comparison to Nice-SLAM [17], Co-SLAM [15], ESLAM [9] and BSLAM [8]. We report the PSNR, SSIM and LPIPS as metrics to reflect the rendering quality. Our model demonstrates advanced results across all metrics.

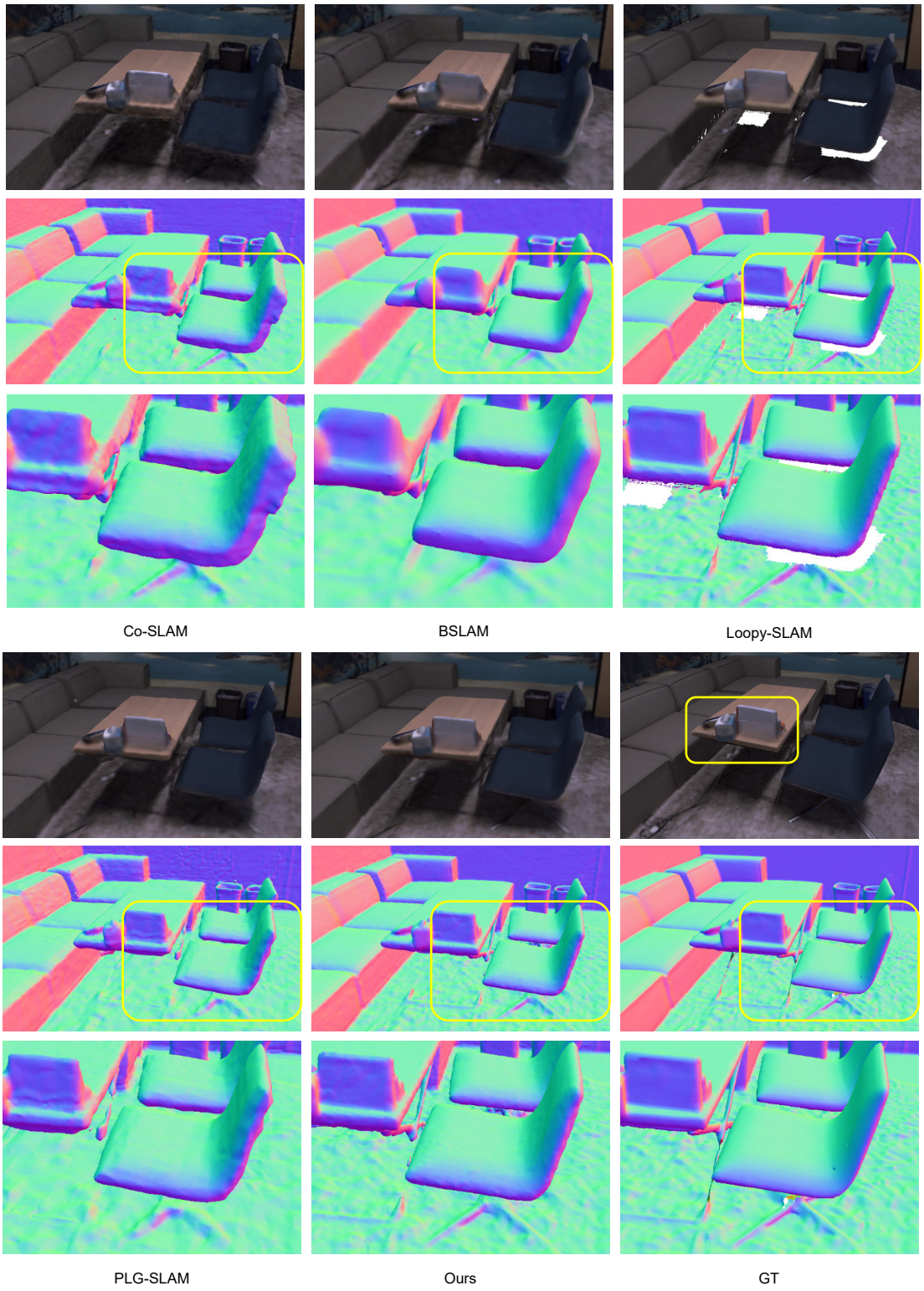


Figure 13. **Mesh Evaluation on Replica [13]** `Office-0`. Notably, our method can present fine geometric structures while also achieving better scene completion for unobserved regions compared to explicit Loopy-SLAM [10]. Compared to implicit methods such as Co-SLAM [15], BSLAM [8], and PLG-SLAM [5], our method captures finer high-frequency geometric details. For example, the chair back, chair legs, and the carpet. For appearance, rendered objects on the table are also better.

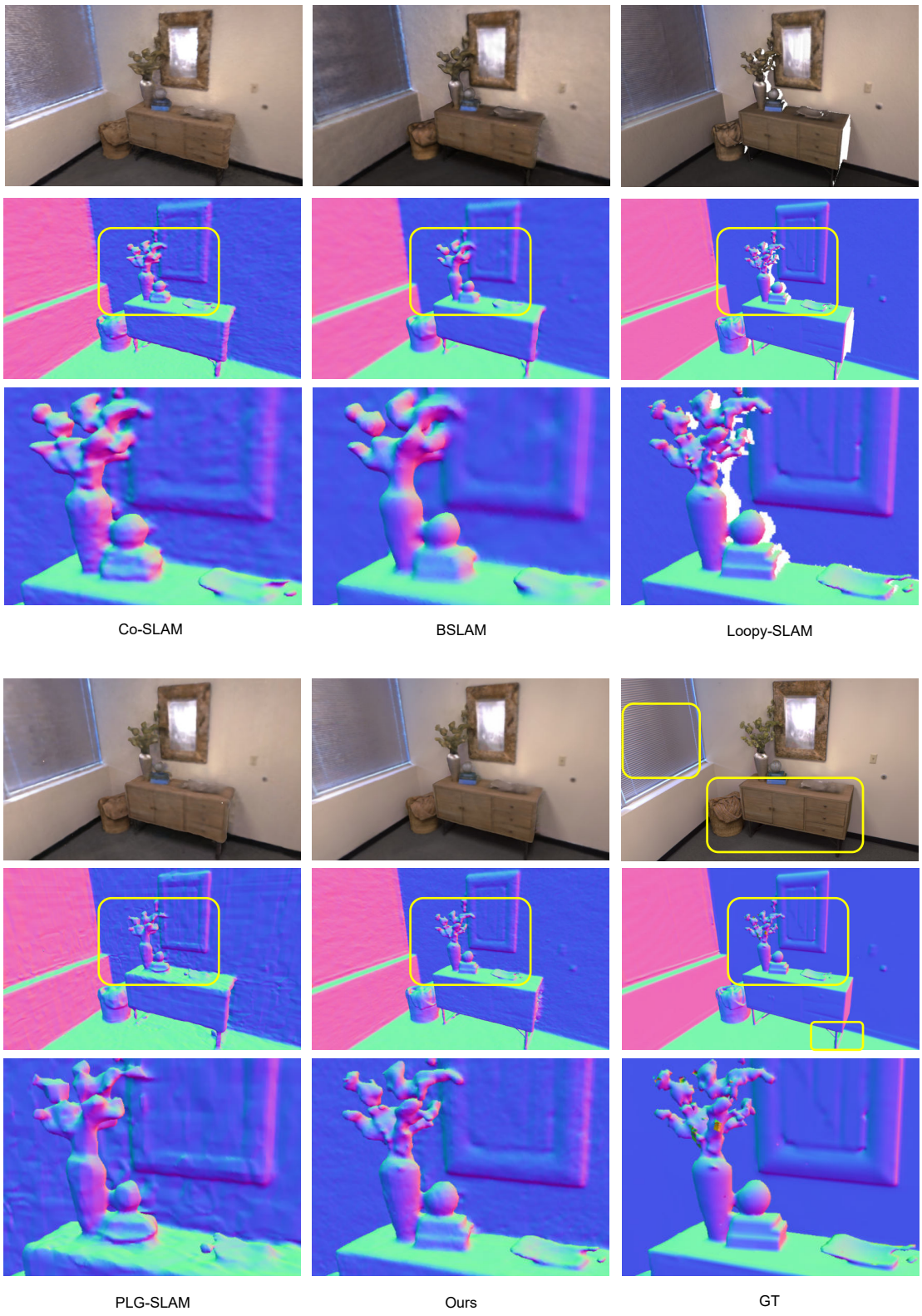


Figure 14. **Mesh Evaluation on Replica [13]** `Room-2`. Our method achieves finer geometric and appearance reconstruction. For appearance: the patterns on the curtains and the detailed textures on the cabinet surface. For geometry: the vase on the cabinet and the cabinet legs. Please zoom in for more details.

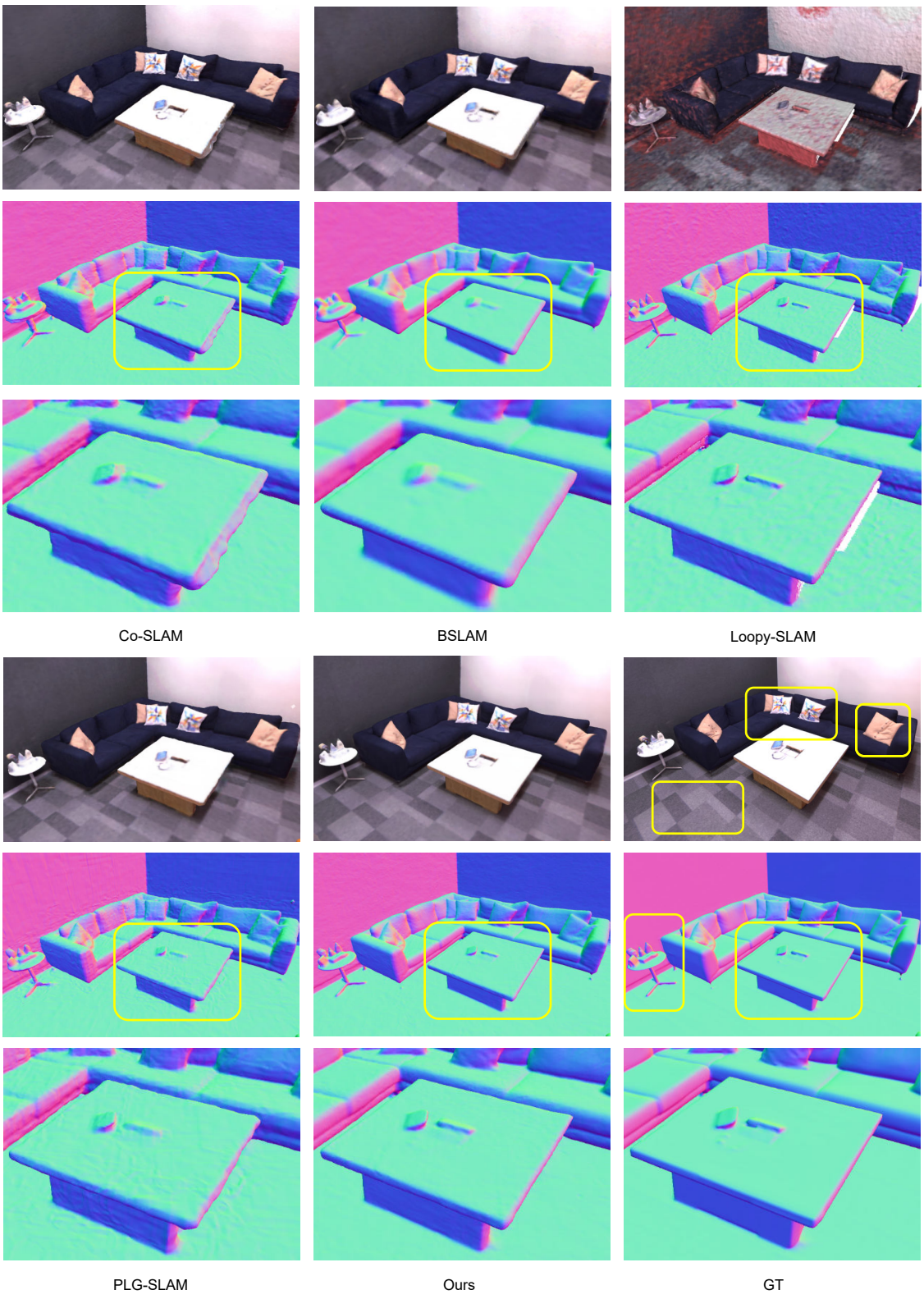


Figure 15. **Mesh Evaluation on Replica [13]** `Office-2`. For appearance: our rendered floor has higher quality, clearly distinguishing the floor patterns, as well as the textures on the pillows on the sofa. For geometry: we zoomed in on the table, and our method reconstructs sharper edges and smoother surfaces.

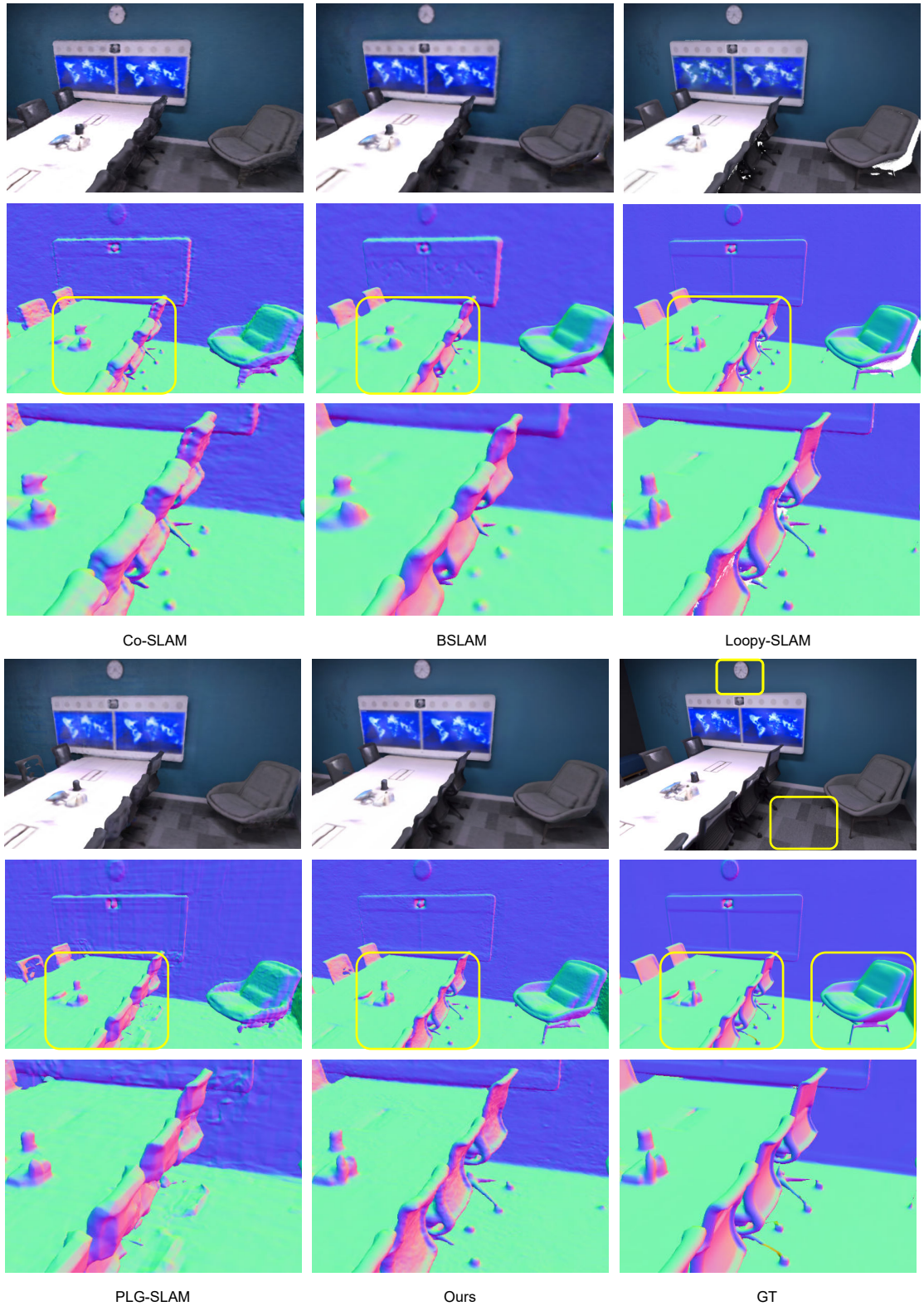


Figure 16. **Mesh Evaluation on Replica [13]** `Office-4`. For appearance: our rendered floor quality is higher, clearly distinguishing the floor patterns, as well as the clock on the wall. For geometry: we reconstructed more of the office chair's geometric structure, such as the legs and the backrest. The geometry of the sofa in the corner also demonstrates the superiority of our algorithm.

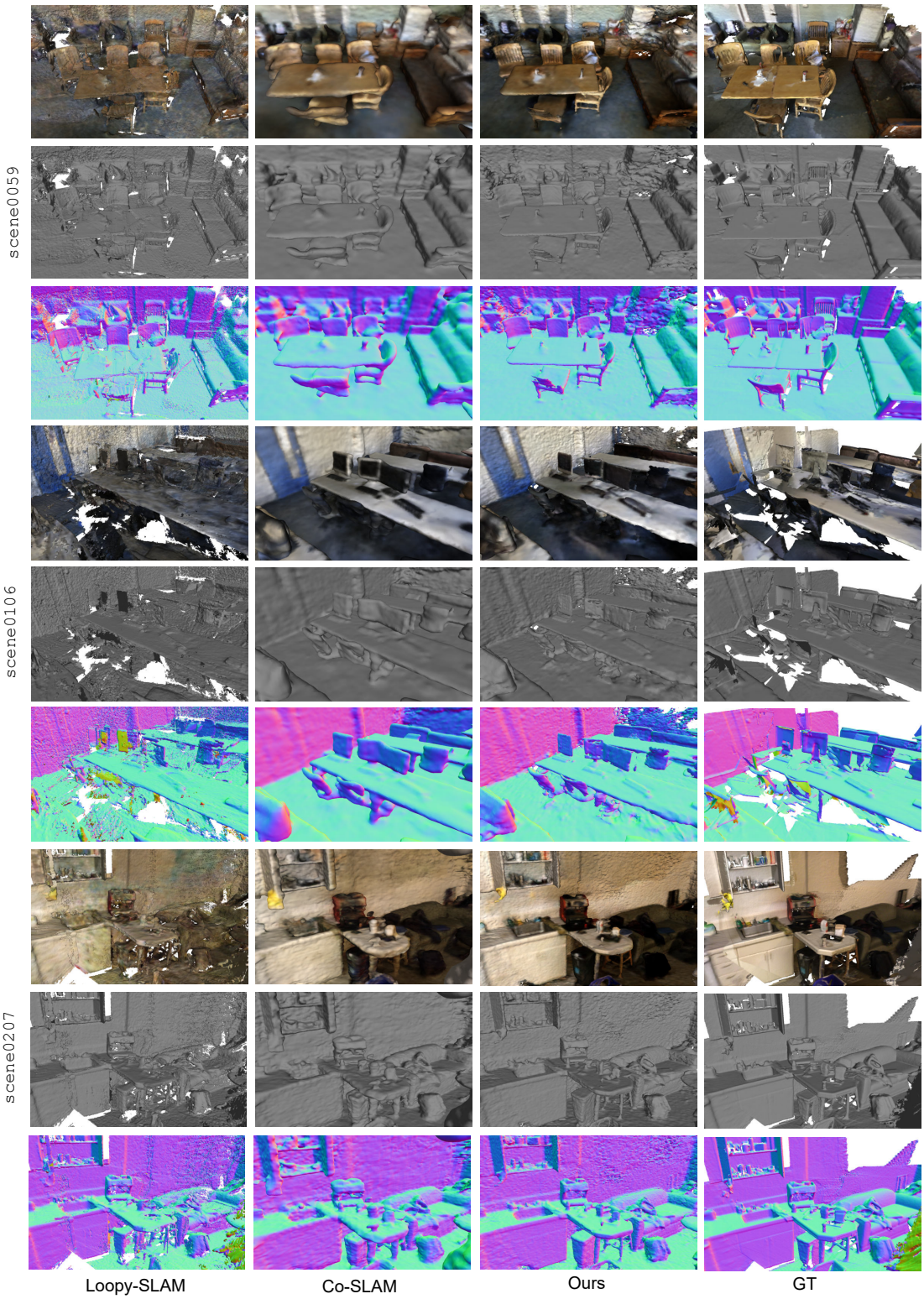


Figure 17. **Mesh Evaluation on ScanNet [4].** Explicit Loopy-SLAM [10] and implicit Co-SLAM [15] are listed here for comparison. For appearance: Our method achieves higher rendering quality compared to the ground truth (GT) mesh, as seen texture of chairs, objects on the desk in `scene0059`, and the coffee machine on the table in `scene0207`. For geometry: More detailed and complete results are reconstructed, such as table and chairs in `scene0059`, the surface of desks in `scene0207`.

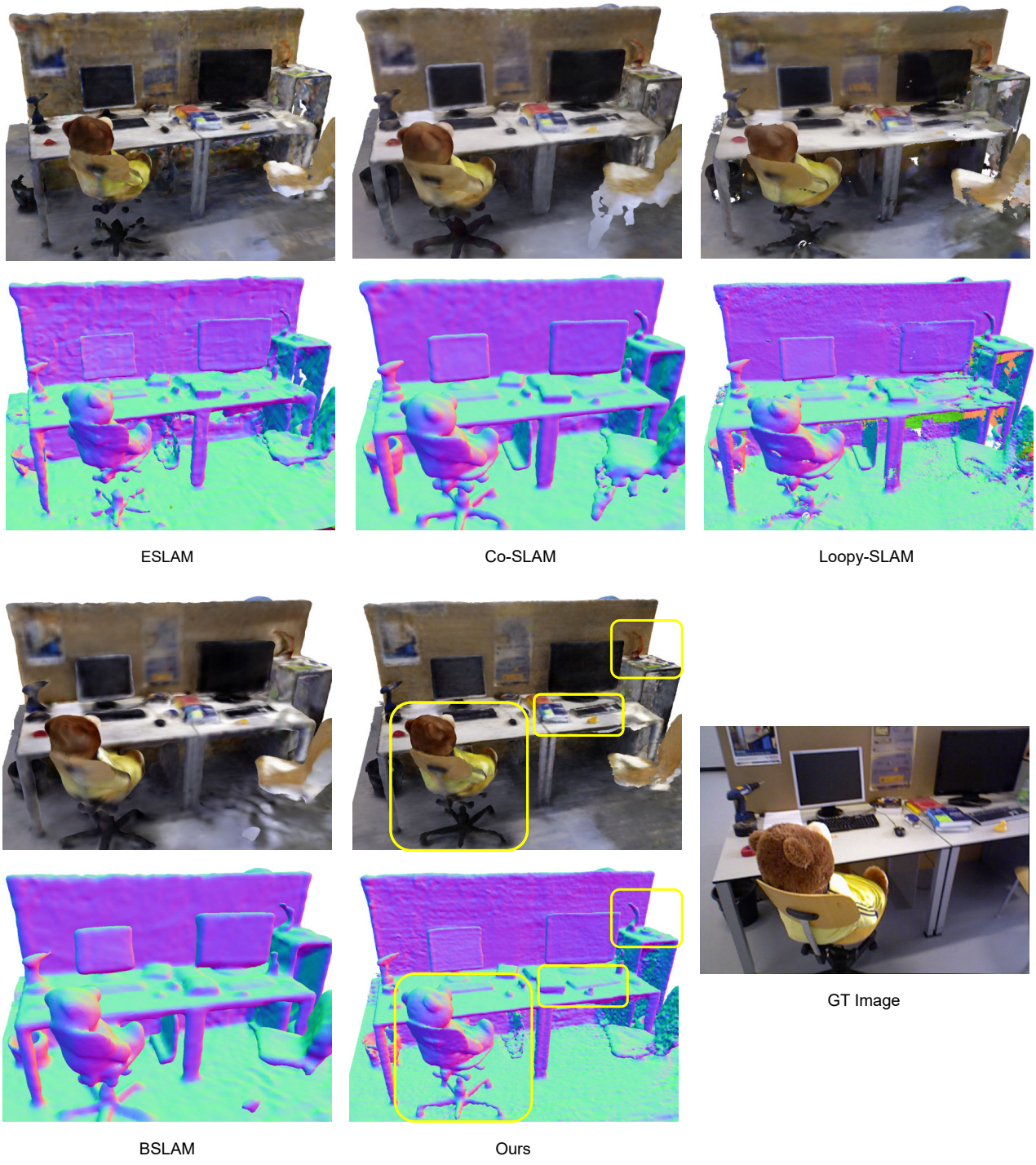


Figure 18. **Mesh Evaluation on TUM RGB-D [14].** Because there is no ground truth mesh for the TUM RGB-D dataset, we provide an image to facilitate qualitative comparison. We extensively compare the reconstruction quality with implicit methods such as ESLAM [9], Co-SLAM [15], and BSLAM [8], as well as the explicit method Loopy-SLAM [10]. The results show that our method achieves superior quality in both rendering and geometry. Our method captures finer geometric details and higher fidelity rendering, such as the legs of the chair, the teddy bear, and the captured objects on the table.

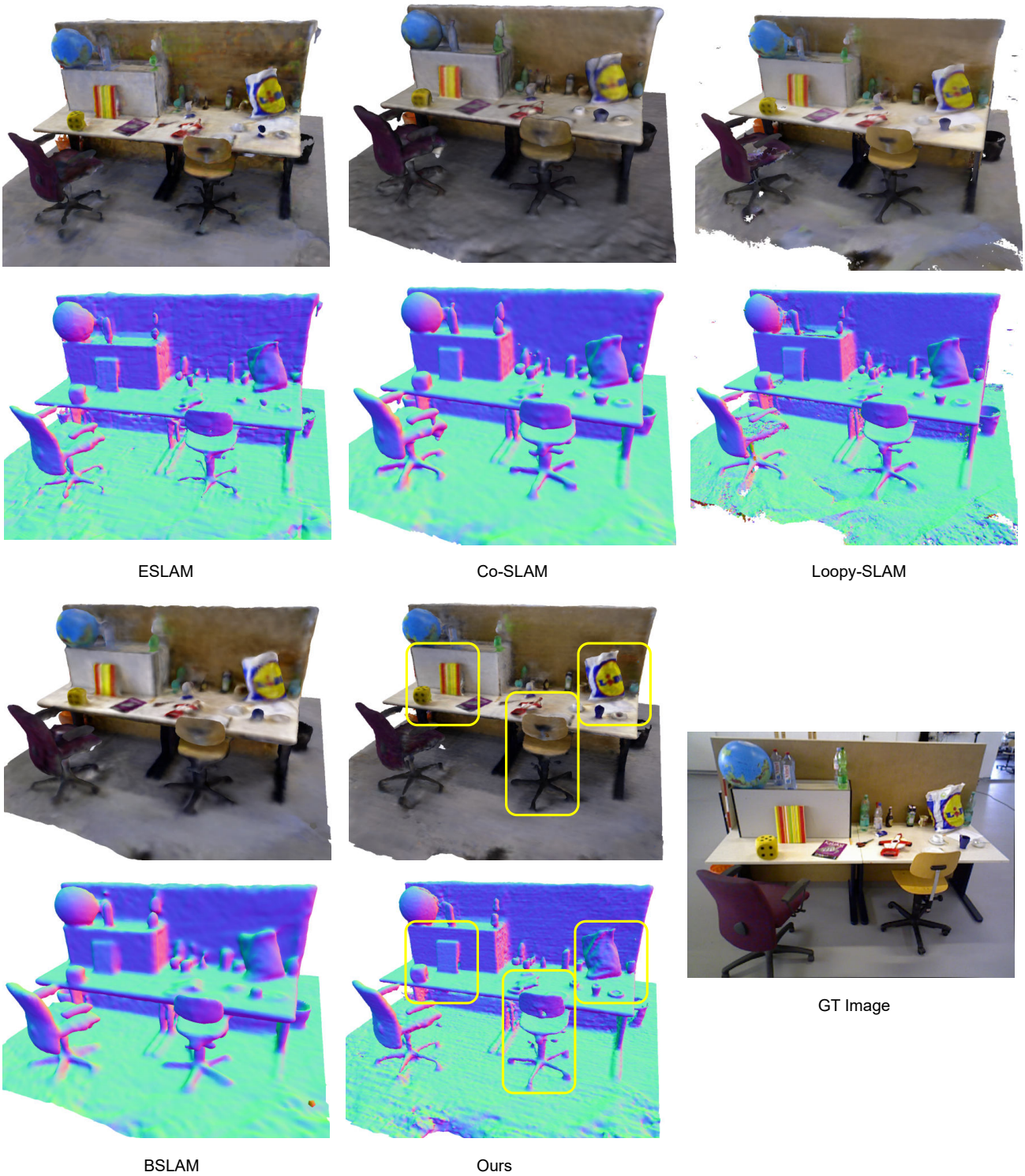


Figure 19. **Mesh Evaluation on TUM RGB-D [14].** Because there is no ground truth mesh for the TUM RGB-D dataset, we provide an image to facilitate qualitative comparison. For example, our method accurately reconstructs details such as the Rubik's cube on the table, the shopping bag, and the chair.

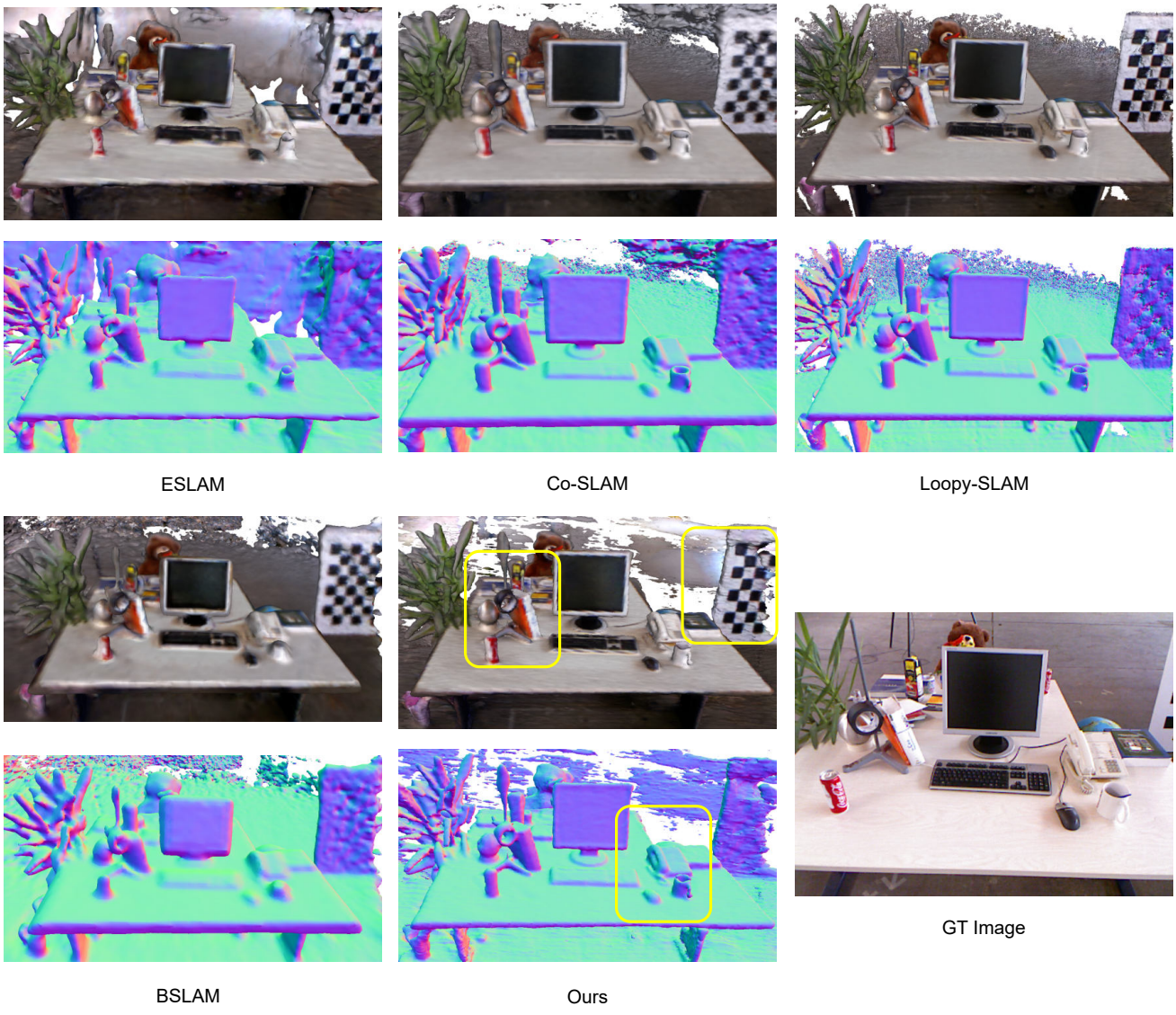


Figure 20. **Mesh Evaluation on TUM RGB-D [14].** While ESLAM [9] and BSLAM [8] can not capture geometric details such as cup and mouse on table, Co-SLAM [15] can not reconstruct thin geometric structure, such thin table surface. Our method shows outstanding performance.

References

[1] Dejan Azinović, Ricardo Martin-Brualla, Dan B Goldman, Matthias Nießner, and Justus Thies. Neural rgb-d surface reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6290–6301, 2022. 1

[2] Eric R Chan, Connor Z Lin, Matthew A Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas J Guibas, Jonathan Tremblay, Sameh Khamis, et al. Efficient geometry-aware 3d generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16123–16133, 2022. 6

[3] Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 303–312, 1996. 7

[4] 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. 1, 3, 4, 5, 6, 7, 8, 17

[5] Tianchen Deng, Guole Shen, Tong Qin, Jianyu Wang, Wentao Zhao, Jingchuan Wang, Danwei Wang, and Weidong Chen. Plgslam: Progressive neural scene represenation with local to global bundle adjustment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19657–19666, 2024. 8, 13

[6] Ziyue Feng, Huangying Zhan, Zheng Chen, Qingan Yan, Xiangyu Xu, Changjiang Cai, Bing Li, Qilun Zhu, and Yi Xu. Naruto: Neural active reconstruction from uncertain target observations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21572–21583, 2024. 4

[7] Lily Goli, Cody Reading, Silvia Sellán, Alec Jacobson, and Andrea Tagliasacchi. Bayes' rays: Uncertainty quantification for neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20061–20070, 2024. 4

[8] Tongyan Hua and Lin Wang. Benchmarking implicit neural representation and geometric rendering in real-time rgb-d slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21346–21356, 2024. 3, 4, 6, 7, 8, 11, 12, 13, 18, 20

[9] Mohammad Mahdi Johari, Camilla Carta, and François Fleuret. Eslam: Efficient dense slam system based on hybrid representation of signed distance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17408–17419, 2023. 3, 6, 7, 8, 11, 12, 18, 20

[10] Lorenzo Liso, Erik Sandström, Vladimir Yugay, Luc Van Gool, and Martin R Oswald. Loopy-slam: Dense neural slam with loop closures. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20363–20373, 2024. 8, 13, 17, 18

[11] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. *ACM SIGGRAPH Computer Graphics*, 21(4):163–169, 1987. 1

[12] Erik Sandström, Yue Li, Luc Van Gool, and Martin R Oswald. Point-slam: Dense neural point cloud-based slam. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18433–18444, 2023. 8

[13] 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. 1, 4, 5, 6, 7, 8, 10, 11, 12, 13, 14, 15, 16

[14] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of rgb-d slam systems. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 573–580. IEEE, 2012. 1, 5, 6, 7, 8, 18, 19, 20

[15] Hengyi Wang, Jingwen Wang, and Lourdes Agapito. Co-slam: Joint coordinate and sparse parametric encodings for neural real-time slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13293–13302, 2023. 3, 6, 7, 8, 11, 12, 13, 17, 18, 20

[16] Xingrui Yang, Hai Li, Hongjia Zhai, Yuhang Ming, Yuqian Liu, and Guofeng Zhang. Vox-fusion: Dense tracking and mapping with voxel-based neural implicit representation. In *2022 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 499–507. IEEE, 2022. 12

[17] Zihan Zhu, Songyou Peng, Viktor Larsson, Weiwei Xu, Hujun Bao, Zhaopeng Cui, Martin R Oswald, and Marc Pollefeys. Nice-slam: Neural implicit scalable encoding for slam. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12786–12796, 2022. 7, 8, 11, 12