Blurred LiDAR for Sharper 3D: Robust Handheld 3D Scanning with Diffuse LiDAR and RGB

Supplementary Material

7. Ablation of Scene-Adaptive Loss

Below, we provide several ablations of the scene-adaptive loss proposed to balance signal dynamically from diffuse LiDAR and RGB sensors.

7.1. Adaptive Loss Ablation: Fixed Diffuse LiDAR Weight

We present ablation results for our technique using diffuse LiDAR and RGB with a fixed, global diffuse LiDAR loss weight. This approach mirrors the loss formulation commonly used in conventional LiDAR, where point-based supervision is implemented through an L_1 loss for depth: $L_{\text{depth}} = \|\hat{D} - D_{\text{gt}}\|_1$ where $\hat{D}$ is the predicted depth map, and D_{gt} is the ground truth depth. In these settings, a fixed loss weighting is typically applied to the depth loss throughout supervision; we present results in our setting akin to this fixed-weight training. Results are shown in Tab. 3.

Our findings consistently demonstrate improved performance with our scene-adaptive, patch-based loss formulation. Specifically, we observe notable gains in RGB and depth estimation, particularly in high-texture regions where the RGB signal is prioritized by our adaptive technique. While the fixed loss weighting performs comparably in textureless scenes, the consistent improvements across diverse textures highlight the advantages of a modified loss formulation that effectively balances RGB and diffuse LiDAR signals based on scene content.

Table 3. **Ablation: Using Fixed Loss Weight.** Results for 10 training and 10 testing images are shown, comparing performance with a fixed, global diffuse LiDAR loss weight. The findings demonstrate that while fixed weighting performs well in textureless regions, our scene-adaptive, patch-based loss achieves consistently better results in both RGB rendering and depth estimation across a variety of high and mixed-texture scenes.

Method	Blender		Chair		Hotdog		Lego	
	PSNR ↑	D.MAE ↓	PSNR ↑	D.MAE ↓	PSNR ↑	D.MAE ↓	PSNR ↑	D.MAE ↓
(a) Full Texture Datasets: Scenes with full texture on object and ground plane.								
Ablation - Fixed Transient Loss Weight	20.54	0.050	18.06	0.072	21.08	0.033	20.53	0.046
Ours	30.67	0.025	30.25	0.017	30.14	0.016	28.39	0.025
(b) Textured Object Datasets: Scenes with textured objects on completely textureless ground planes.								
Ablation - Fixed Transient Loss Weight	18.38	0.048	22.67	0.048	18.55	0.040	17.27	0.043
Ours	25.08	0.033	32.18	0.030	35.62	0.024	30.00	0.024
(c) Textured Plane Datasets: Scenes with completely textureless objects on textured ground planes.								
Ablation - Fixed Transient Loss Weight	19.69	0.053	20.48	0.045	20.50	0.036	20.20	0.046
Ours	23.72	0.045	25.70	0.037	28.97	0.034	26.60	0.046
(d) Full Textureless Datasets: Completely textureless scenes.								
Ablation - Fixed Transient Loss Weight	–	0.046	–	0.049	–	0.039	–	0.046
Ours	–	0.045	–	0.045	–	0.041	–	0.042

7.2. Patch-based Loss Ablation: Adaptive Global Loss

We present ablation results for our technique using diffuse LiDAR and RGB with a global adaptive loss rather than a patch-based loss. Our intuition is that scene-adaptive techniques should be applied not only to entire images but also to patches within images. This is because regions of high texture may benefit more from RGB signals, while low-texture regions might rely more heavily on diffuse LiDAR. The results are summarized in Tab. 4.

The findings indicate that using a globally applied scene-adaptive loss improves both color and geometry estimation compared to a fixed loss (Tab. 3). However, our patch-based approach consistently achieves better RGB and depth/geometry estimation. Qualitatively, we observe that global weighting can lead to "smoothing" in both geometry and RGB, where fine geometric details, and hence precise colors, are less effectively resolved. In such regions, significantly higher RGB loss weighting is preferred. Notably, in textureless scenes (Tab. 4d), the differences between our patch-based approach and the global method are smaller.

Table 4. **Ablation: Using Per-Image Adaptive Weights.** Results are presented for 10 training and 10 testing images, comparing an adaptive loss weight without patch-based loss. Loss weighting is computed similarly to our method, using scene variance and SNR, but applied globally rather than patch-wise. The results show that while global weighting improves robustness to texture variations, a gap remains compared to patch-wise weighting. Our method enables improved color estimation and consistently better depth and geometry results.

Method	Blender		Chair		Hotdog		Lego	
	PSNR ↑	D.MAE ↓	PSNR ↑	D.MAE ↓	PSNR ↑	D.MAE ↓	PSNR ↑	D.MAE ↓
(a) Full Texture Datasets: Scenes with full texture on object and ground plane.								
Ablation - Smart Image-wide Weighting	25.44	0.036	26.34	0.025	28.31	0.023	27.00	0.028
Ours	30.67	0.025	30.25	0.017	30.14	0.016	28.39	0.025
(b) Textured Object Datasets: Scenes with textured objects on completely textureless ground planes.								
Ablation - Smart Image-wide Weighting	17.23	0.051	28.09	0.035	32.35	0.027	26.14	0.027
Ours	25.08	0.033	32.18	0.030	35.62	0.024	30.00	0.024
(c) Textured Plane Datasets: Scenes with completely textureless objects on textured ground planes.								
Ablation - Smart Image-wide Weighting	22.07	0.052	23.41	0.039	27.61	0.033	24.74	0.044
Ours	23.72	0.045	25.70	0.037	28.97	0.034	26.60	0.046
(d) Full Textureless Datasets: Completely textureless scenes.								
Ablation - Smart Image-wide Weighting	–	0.049	–	0.051	–	0.043	–	0.052
Ours	–	0.045	–	0.045	–	0.041	–	0.042

7.3. Scene-Adaptive Loss Ablation on 45 Images

We present results for our scene-adaptive loss approach using 45 training images and 15 test images (compared to the 10 training and 10 test images shown in the main paper). This aims to demonstrate that our results remain consistent even when the dataset size increases. The results for this ablation are shown in Tab. 5.

We compare our technique to a patch-based ablation using global variance and SNR to compute a global diffuse LiDAR weight (the same approach used in Tab. 4, providing the strongest possible ablation comparison). Our findings confirm that the improvements achieved by our method are consistent across a larger dataset. Notably, we observe effects similar to those seen in smaller datasets: reliance on diffuse LiDAR weighting in high-texture regions can lead to undesirable smoothing, particularly when RGB cues already provide sufficient texture. This smoothing effect can degrade performance, even in high-input view regimes. Our patch-based loss consistently avoids this issue, enabling improved results across RGB rendering and depth/geometry estimation.

Table 5. **Scene Adaptive Loss Ablation on 45 Images.** Results are presented for a comparison between our scene-adaptive loss approach and a global loss weighting method using diffuse LiDAR and RGB signals. The global method computes a single, scene-specific loss weight for both inputs across entire images, adjusting by scene-wide SNR and texture. In contrast, our scene-adaptive loss incorporates patch-based weighting to dynamically prioritize signals *within images*; we find that our approach consistently outperforms the global weighting approach. This is evident in improved RGB rendering and depth/shape estimation, even when the dataset size is increased to 45 training images.

Method	Blender		Chair		Hotdog		Lego	
	PSNR ↑	D.MAE ↓	PSNR ↑	D.MAE ↓	PSNR ↑	D.MAE ↓	PSNR ↑	D.MAE ↓
(a) Full Texture Datasets: Scenes with full texture on object and ground plane.								
Ablation - No Scene-Adaptive Patch Weight	32.67	0.025	32.07	0.024	33.41	0.022	32.42	0.024
Ours	39.16	0.016	37.34	0.011	39.23	0.009	35.82	0.012
(b) Textured Object Datasets: Scenes with textured objects on completely textureless ground planes.								
Ablation - No Scene-Adaptive Patch Weight	34.61	0.023	31.70	0.022	34.26	0.022	30.46	0.024
Ours	37.84	0.018	37.39	0.016	41.50	0.016	37.26	0.015
(c) Textured Plane Datasets: Scenes with completely textureless objects on textured ground planes.								
Ablation - No Scene-Adaptive Patch Weight	32.18	0.026	32.37	0.025	33.77	0.026	32.96	0.033
Ours	36.28	0.021	36.47	0.017	37.30	0.022	36.71	0.027
(d) Full Textureless Datasets: Completely textureless scenes.								
Ablation - No Scene-Adaptive Patch Weight	–	0.032	–	0.039	–	0.029	–	0.033
Ours	–	0.029	–	0.029	–	0.025	–	0.031

8. Additional Quantitative Evaluation

Creating a large-scale dataset for this new diffuse LiDAR modality, including large-scale simulated and real-world captures with ground truth geometry labels, is a non-trivial yet valuable direction for future work. Currently, adapting existing RGB-D datasets for evaluating diffuse LiDAR is infeasible. Existing RGB-D datasets (e.g. [3, 6, 45]) largely do not capture the time-resolved histograms necessary for diffuse LiDAR, while transient datasets (e.g. [27]) lack this new diffuse LiDAR modality. Simulating diffuse LiDAR from point clouds introduces inaccuracies, as key characteristics–such as spatial coverage, multi-bounce effects, cross-talk, timing jitter, and instrument response functions (Sec. 13)–cannot be faithfully reproduced. To demonstrate some extended evaluation of our method, we present additional results on challenging rendered scenes in Tab. 6, where our method consistently outperforms the RGB with sparse LiDAR baseline.

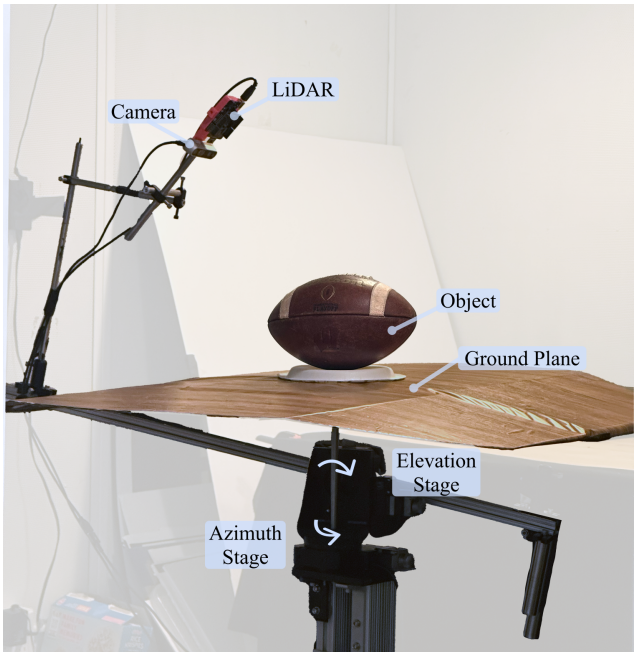

Figure 8. **Capture setup.** Our capture setup enables full 360° captures of objects in azimuth and 25° in elevation. We create a high-texture ground plane using a wood vinyl. The object is centered about the azimuth rotation axis. We mount the camera and LiDAR above the object such that the majority of the cameras FOV incorporates the ground plane.

Scene	Cereal				Seat & Bowl				Plain Box				Backpack			
	PSNR	SSIM	DMAE	NMAE	PSNR	SSIM	DMAE	NMAE	PSNR	SSIM	DMAE	NMAE	PSNR	SSIM	DMAE	NMAE
RGB Only	21.67	0.8089	0.0998	37.60	21.57	0.8318	0.0811	39.65	22.12	0.7641	0.2017	50.95	29.92	0.8457	0.4490	41.13
Sparse	24.55	0.8282	0.0903	37.19	25.13	0.8431	0.1092	37.21	22.21	0.8161	0.1085	34.96	35.31	0.9150	0.0466	23.39
Ours	28.81	0.8513	0.0268	20.17	27.76	0.8548	0.0225	17.82	26.80	0.8415	0.0363	21.02	36.45	0.8745	0.0246	11.61
Scene	Dark Helmet				Washbasin				Candlelight				Bathtub			
	PSNR	SSIM	DMAE	NMAE	PSNR	SSIM	DMAE	NMAE	PSNR	SSIM	DMAE	NMAE	PSNR	SSIM	DMAE	NMAE
RGB Only	31.36	0.6884	0.5593	48.62	25.80	0.8880	0.1328	48.11	28.37	0.8627	0.1104	39.35	20.11	0.7864	0.2373	53.73
Sparse	35.43	0.8957	0.0786	41.66	26.06	0.8815	0.0707	39.82	33.55	0.9388	0.0349	29.00	24.27	0.8820	0.0586	37.20
Ours	35.05	0.8702	0.0398	12.77	28.33	0.9082	0.0293	26.06	35.64	0.9407	0.0218	25.19	27.02	0.8943	0.0358	23.38

Table 6. **Metrics for additional rendered scenes**. Our RGB with diffuse LiDAR approach consistently outperforms the RGB with Sparse LiDAR baseline in challenging, low-texture, low-light, low-albedo scenes.

9. Discussion on Peripheral Region Robustness and Ground Plane Estimation

Our technique consistently improves object-ground separation and plane estimation compared to baselines. Most notably, we observed that baselines, including Gaussian Surfels (RGB only), Surfels with monocular depth priors, and Surfels with sparse LiDAR, exhibited consistent errors in plane estimation for peripheral regions. Specifically, these methods frequently produced meshes that either (1) poorly separated objects from the ground or (2) achieved

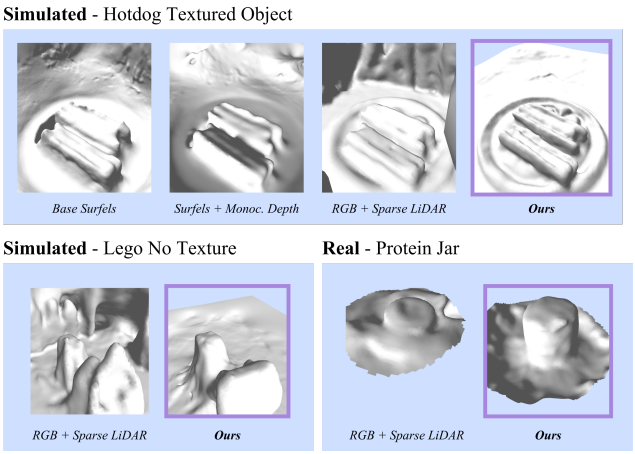

Figure 9. **Robust Object/Ground Separation and Plane Estimation.** Our method demonstrates improved object-ground separation and plane estimation, particularly in peripheral regions with limited input views, outperforming conventional techniques.

reasonable plane estimation in central regions with sufficient multi-view coverage but failed in peripheral regions, i.e., distant areas with limited input views.

This issue was observed in both simulated and real-world results, which we highlight in Fig. 9. Our method addresses these shortcomings, offering robustness not only in textureless and low-SNR regions but also in peripheral and very low-texture areas with limited input views, where traditional RGB and LiDAR-based methods lack sufficient depth cues for accurate reconstruction.

10. Extended Baseline Comparisons

Our work focuses on 3D scanning from multi-view inputs, motivated by sparse LiDAR-supervised RGB scanning for mobile and robotic applications. Given the increasing prevalence of LiDAR-equipped smartphones, we compare against baselines that employ similarly sparse LiDAR sensors (e.g., 12×12 dot projection [23]) rather than dense depth or scanning LiDAR systems. In addition to the results presented in the main text, we evaluate our approach on low-texture scene scanning against Delaunay depth interpolation and RGB-based depth completion [17] for 3D scanning. We present these results in Tab. 7. We consistently outperform these methods, and find that these baselines struggle due to the sparsity of depth in our setting, the low-texture RGB inputs, and the lack of multi-view consistency.

11. Real Data Capture Setup

Our capture setup includes two rotation axes and a sensor mount. The real results, as shown in Figure 7, used only a single rotation axis, however. Our capture setup is shown in Figure 8.

Our setup enables 360° rotation in azimuth (horizontal plane) and ∼25° in elevation. During dataset acquisition, we fix the elevation axis to be at 5° above the horizontal. This way, the angle of the camera and LiDAR are above and facing down towards the object such that the plane where the object sits encapsulates most of the sensor's views. We mount the LiDAR and RGB camera rigidly together in order to keep extrinsics constant throughout the acquisition. The mount is shown in Figure 10.

The LiDAR module we use is the AMS TMF8828 [1], which has 18 × 12 individual SPAD pixels which are grouped into 8 × 8 measurement zones on-device. This is done to minimize noise and reduce bandwidth limitations. We also enable short range, high accuracy mode. This mode reduces the overall measurement range from 10 to 1.5 meters while increasing the timing resolution from 266 to 39

Textured Plane Datasets: Scenes with completely textureless objects on textured ground planes.								
Scene	Blender		Chair		Hotdog		Lego	
	DMAE↓	NMAE↓	DMAE↓	NMAE↓	DMAE↓	NMAE↓	DMAE↓	NMAE↓
Surfels RGB+Depth Interpolation	0.767	68.68	0.832	68.91	0.821	68.18	0.799	67.51
Surfels RGB+Depth Completion	0.635	60.12	0.748	63.46	0.763	62.37	0.709	64.31
Ours (RGB+Diffuse LiDAR)	0.045	21.13	0.037	22.80	0.034	22.32	0.046	25.69
No Texture Datasets: Completely textureless scenes.								
Scene	Blender		Chair		Hotdog		Lego	
	DMAE↓	NMAE↓	DMAE↓	NMAE↓	DMAE↓	NMAE↓	DMAE↓	NMAE↓
Surfels RGB+Depth Interpolation	0.841	37.40	0.898	38.57	0.915	42.55	0.920	40.78
Surfels RGB+Depth Completion	0.729	29.30	0.783	33.43	0.791	25.69	0.805	32.51
Ours (RGB+Diffuse LiDAR)	0.045	15.68	0.045	13.55	0.041	16.35	0.042	14.54

Table 7. **Comparison to depth interpolation and depth completion baselines.** Our diffuse LiDAR and RGB-based scanning technique consistently outperforms these baselines in challenging low-texture scenes.

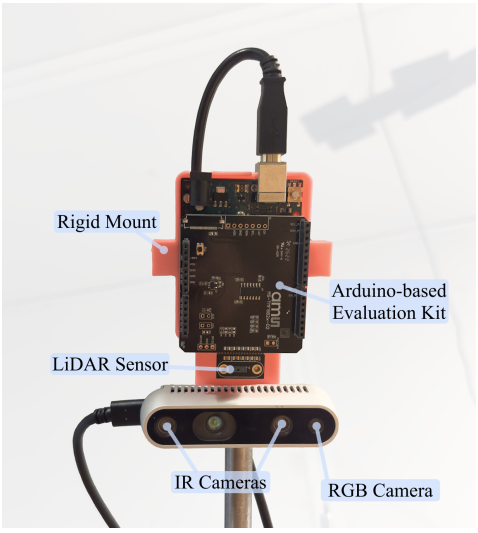

Figure 10. **Sensor mount.** We rigidly mount the LiDAR and camera sensors using a 3D printed part. The camera (an Intel RealSense D435i Depth Camera) has both an RGB camera, and two IR cameras which are used to produce a dense depth map, which we use for approximating a sparse LiDAR. We use the Arduino-based evaluation kit provided by AMS for interfacing with the LiDAR sensor.

picoseconds per bin. This limits the object size which we can capture, though is sufficient in handheld scanning scenarios. We utilize the AMS Arduino-based evaluation kit to interface with the sensor and acquire the histogram measurements. Due to limitations with Arduino serial speeds, we receive 8×8 histogram measurements about 1Hz.

For RGB measurements, we use the Intel RealSense D435i Depth Camera. This camera module has three imaging sensors: two infrared cameras (IR), and an RGB sensor. The IR cameras can be used to extract a high-resolution depth map from the scene using structured light. We utilize this depth map to approximate the measurements theoretically obtained by a sparse LiDAR; we note that this is a more faithful comparison than using the measured transient peaks as point depths (as shown in Mu et al. [31]). The RGB camera has a 1920×1080 resolution RGB sensor and a FOV of $66° \times 42°$.

12. Real Data Processing

To utilize the sensor data in our algorithm, it's necessary to calibrate the LiDAR and RGB such that we know the relative correspondences between pixels for rendering. We calibrate the sensors using a custom 2D gantry system, as shown in Figure 11.

To calibrate the LiDAR and RGB, we move the gantry in the vertical plane along the FOV of the RGB camera and match peak positions in the 8×8 sensor to the pixels in the RGB camera. We mount a retroreflective patch on the 2D gantry which has a very high return relative to any surrounding diffuse objects. We then put green tape around the patch and use a color threshold to find the position of the patch in the image. Given the patch's position in the image, and it's corresponding peak in the LiDAR's transient, we can interpolate the per-pixel response of the LiDAR relative to the RGB image.

As noted in Mu et al. [31], there is a significant amount of crosstalk observed in the histograms. Crosstalk is effectively when noise or a signal response in one pixel interferes or is measured in neighboring pixels. We have found this crosstalk is exhibited non-uniformly across all pixels and is dependent on both the position and surface properties of objects we captured. We do not model this crosstalk through our calibration. To reduce crosstalk, high-SNR regions were isolated via thresholding, followed by power-law normalization and intensity scaling to normalize transients. While we do not explicitly model crosstalk in our forward model, we consider this modeling as relevant future work.

13. Simulation and Real Capture Differences

In this section, we discuss key differences between simulated and real-world setups that help explain the discrepancies observed between results in these settings. While we noticed similar trends in both setups—such as improved geometry estimation with fewer input images, enhanced object/ground separation, better plane estimation, and finer geometric detail for diffuse LiDAR over sparse LiDAR—there were notable differences as well.

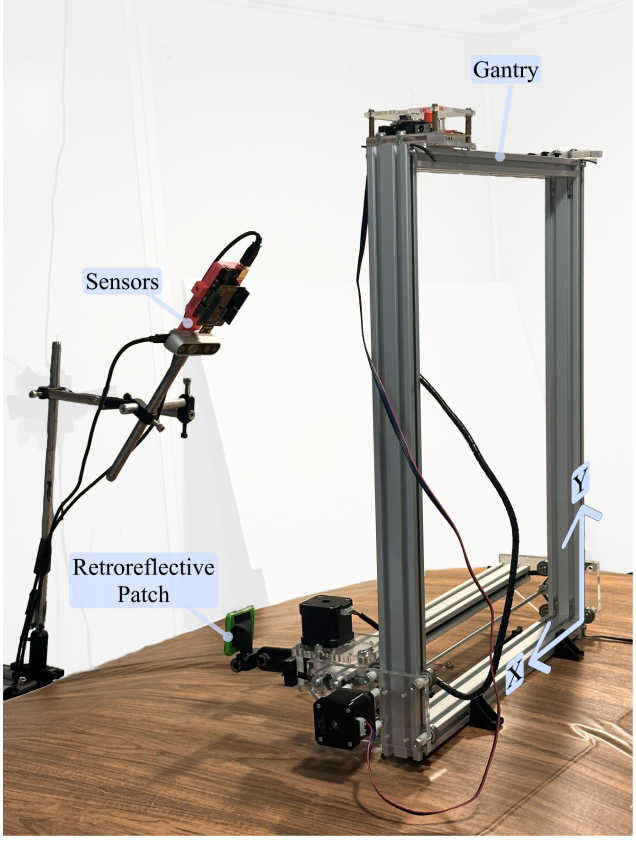

Figure 11. **Calibration setup.** To calibrate the LiDAR and camera modules, we utilize a 2D gantry system to precisely position a retroreflective patch in the FOV of the sensors. We find the corresponding peak (which has a very large return because the patch is retroreflective) and pixel position of the patch for the entire FOV of both sensors.

In real-world captures, we observed that more images were required for training, object/ground separation was slightly poorer, and geometric details were less precise than in simulation. While the superiority of RGB and diffuse LiDAR modeling over sparse LiDAR was evident in both cases, these observations motivate a discussion of the differences between our simulated and real-world regimes.

IFOV, pixel overlap, and response area. One key difference between simulation and real hardware captures using the TMF8828 is related to pixel instantaneous field of view (IFOV), overlap, and response areas. In simulation, we modeled a fixed horizontal and vertical pixel IFOV of approximately $4.9°$ with no overlap (stride) between pix-

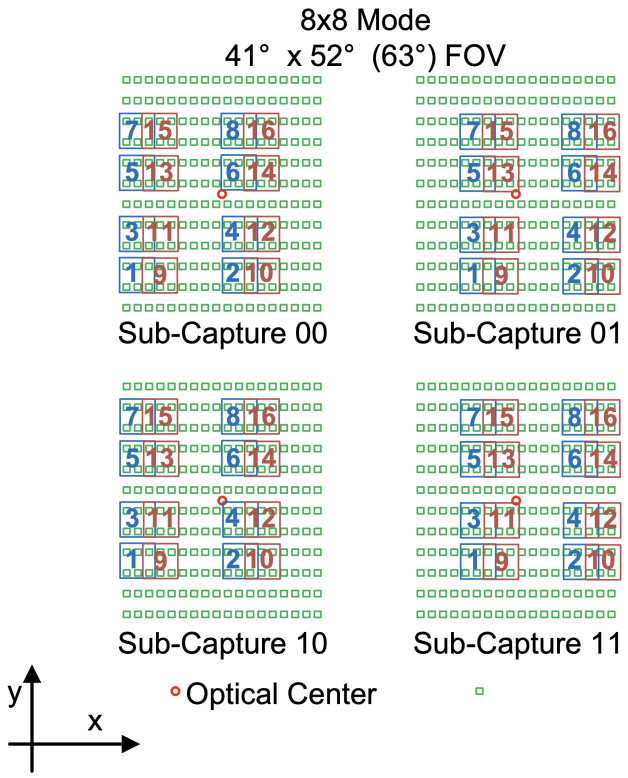

Figure 12. **TMF8828 Pixel Overlap.** The diffuse LiDAR sensor we use (TMF8828, $10 retail) has an 18×12 SPAD array that is grouped into 8×8 pixels; these groups have designed overlap that introduces an additional spatial mixing into measurements. We model this pixel overlap through sensor calibration, and consider it one additional deconvolution required in our analysis-by-synthesis framework. Note that this overlap is distinct from crosstalk, which we discuss in Sec. 12

.

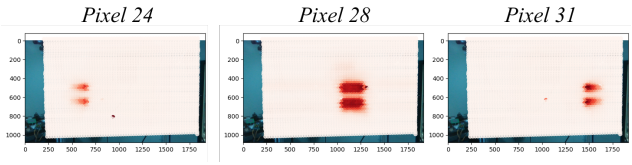

Figure 13. **Diffuse LiDAR Pixel Responses.** We visualize the response intensity for different scanned gantry locations (each with a small retroreflective patch) for three separate diffuse LiDAR pixels. We observe 1) non-uniform responses induced by different spacing between the LiDAR grouped-pixel array rows and columns, and 2) weaker, noiser responses at the peripheral pixels.

els. However, calibration of real hardware revealed several critical deviations: 1) Response area grouping and overlap. According to TMF8828 specifications, the original 18x12 SPAD array is grouped into 8x8 response areas with intentional overlap Fig. 12. This overlap introduces additional deconvolution required during forward rendering to resolve pixel overlap ambiguities. Furthermore, the original SPAD grid exhibited non-uniform spacing between pixel rows and columns, leading to horizontal offsets between grouped pixel rows. 2) Variable IFOVs. Calibration revealed significant variability in pixel IFOV depending on location. Specifically, we observed that (1) pixel IFOVs are horizontally wider than they are tall, and (2) peripheral pixels have much smaller response areas compared to central pixels Fig. 13.

These differences have two main consequences: 1) Pixels are unevenly spaced and capture non-uniform spatial regions, with central pixels covering the most relevant scene information. 2) Peripheral pixels were often noisy, and in particular, the top row of pixels showed no overlap with RGB data. Consequently, we mask out this region during training, resulting in a smaller effective region of diffuse LiDAR response compared to simulation. These differences between simulated and real-world setups highlight the challenges posed by hardware-specific characteristics and their impact on modeling accuracy.

Camera Poses. A significant difference between our simulated and real-world experiments lies in the accuracy of camera poses. In simulation, we used ground-truth camera poses, whereas in real-world experiments, camera poses were estimated using COLMAP. Given the challenging nature of our scenes, COLMAP-estimated poses are expected to be less accurate than the ground truth used in simulation. While this discrepancy could partly explain differences between simulated and real-world results, we emphasize that our method demonstrates robust performance even when using COLMAP-estimated poses for all captures (see main paper). This highlights the strength of our diffuse LiDAR-guided approach, which remains effective even under less accurate, scene-estimated camera poses.

Crosstalk. In our real-world captures using the TMF8828 sensor, we observed a degree of crosstalk that could not be accounted for during calibration. This crosstalk is distinct from pixel overlap, which we explicitly model through calibration. An extended discussion of crosstalk is provided in Section 12. We did not model this crosstalk in simulation, as we consider it a limitation of the specific sensor hardware rather than an inherent property of diffuse LiDAR. In practice, we expect that diffuse LiDAR systems with improved sensor designs would not exhibit the crosstalk observed in the TMF8828.

14. Details on Recoverability Analysis

In this section, we elaborate on the approximate forward model used to analyze the benefits of diffuse LiDAR in Sec. 3.2 of the main text. We perform analysis and derivations in 2D without loss of generality.

We first voxelize the volume of interest, with the origin $(0, 0)$ at the center of the voxel grid. The camera moves

in such a way that the optical axis of the camera always points towards the origin, and the camera remains a constant distance away from the origin. For our analysis, we model a single-pixel camera with a defined field of view (FOV).

Assuming the scene consists of a single voxel v and the camera is located at $\mathbf{x}_c$, the transient measurement is

$$i_v(t; \mathbf{x}_c) = \rho(v) \cdot \delta(ct - 2|\mathbf{x}_v - \mathbf{x}_c|), \tag{17}$$

where $\rho(v) \in \mathbb{R}_+$ is the amount of light reflected by voxel v. Due to the linearity of light transport, the ToF measurement, after accounting for the contribution from all voxels, can be modeled as

$$i(t; \mathbf{x}_c) = \sum_{v \in \mathcal{V}} \rho(v) \cdot V(\mathbf{x}_c, \mathbf{x}_v) \cdot i_v(t; \mathbf{x}_c), \tag{18}$$

where $V(\mathbf{x}_c, \mathbf{x}_v) \in \{0, 1\}$ denotes the visibility between the camera and the voxel v and $\mathcal{V}$ is the set of all voxels in the IFOV of the pixel. The visibility term models occlusions and self-occlusions, but introduces non-linearity into the forward model because the visibility of one voxel is determined by the occupancy of other voxels. In order to simplify our analysis, we neglect this visibility term. Doing so simplifies the transient measurement as

$$i(t; \mathbf{x}_c) = \sum_{v \in \mathcal{V}} \rho(v) \cdot i_v(t; \mathbf{x}_c) \tag{19}$$

Temporal discretization of Eq. (19) leaves us with

$$\mathbf{i}_c = \sum_{v \in \mathcal{V}} \rho_v \cdot \mathbf{i}_{c,v} \tag{20}$$

$$= \mathbf{A}_c \boldsymbol{\rho} \tag{21}$$

where $\mathbf{i}_c \in \mathbb{R}^{N_t}$ is the transient measurement at camera location c, $\boldsymbol{\rho} \in \mathbb{R}^{N_v}$ is the contribution of each voxel to the transient measurement, and $\mathbf{A}_c \in \mathbb{R}^{N_t \times N_v}$ is the linear operator from voxel occupancy to transient measurement. The ith columns of $\mathbf{A}_c$ contains the temporal delta response corresponding to the ith voxel, as indicated by Eq. (17). By vertically stacking the measurements at all camera locations $\mathbf{i}_c$ and their corresponding measurement operators $\mathbf{A_c}$, we obtain a final linear model of the form $\mathbf{i} = \mathbf{A}\boldsymbol{\rho}$

$$\underbrace{\begin{bmatrix} \mathbf{i}_1 \\ \vdots \\ \mathbf{i}_{N_m} \end{bmatrix}}_{\mathbf{i}} = \underbrace{\begin{bmatrix} \mathbf{A}_1 \\ \vdots \\ \mathbf{A}_{N_m} \end{bmatrix}}_{\mathbf{A}} \boldsymbol{\rho}, \tag{22}$$

where $\mathbf{A} \in \mathbb{R}^{N_m N_t \times N_v}$, N_m is the number of camera measurements and $\mathbf{i} \in \mathbb{R}^{N_m N_t}$. Because the $\mathbf{A}$ matrix is a function of the IFOV (Eq. (19)) and number of measurements, we can analyze the impact of these two parameters by studying the invertability properties (we use rank in this work) of $\mathbf{A}$ as these parameters vary.

15. Sparse LiDAR Reaches Diffuse LiDAR Performance with Sufficient Views

Below, we experimentally demonstrate that the benefits of diffuse LiDAR described in our analysis of scene recoverability (Sec. 3.2) hold true in our simulated test settings. Specifically, we find that sparse LiDAR can eventually match the performance of diffuse LiDAR, particularly in input view-limited regimes. Results for our four simulated scenes (Blender, Chair, Hotdog, Lego) across texture variations are shown in Tab. 8. We evaluate geometry estimation performance (depth and normals) of our method using RGB and diffuse LiDAR with 3 and 10 input views (the latter shown in the main paper). We compare this against the performance of sparse LiDAR with RGB, varying the number of input views from 3 to 30.

Our findings indicate that RGB with sparse LiDAR requires approximately 2-3$\times$ more input views to achieve the performance of RGB with diffuse LiDAR. Specifically: 1) for geometry estimation at 3 input views with RGB and diffuse LiDAR, sparse LiDAR typically requires 10-15 views to achieve equivalent performance, and 2) for geometry estimation at 10 input views with RGB and diffuse LiDAR, sparse LiDAR typically requires 25-30 views.

These findings align with our earlier analysis: in input view-limited settings, diffuse LiDAR provides greater coverage and improved recoverability compared to sparse LiDAR. As the number of input views increases, this difference diminishes. With sufficient input views, RGB with sparse LiDAR can eventually match—and in some cases surpass—RGB with diffuse LiDAR. Finally, while we observe consistent improvements in depth and normal estimation, RGB estimation (e.g., PSNR) for sparse LiDAR improves with more views. However, these RGB gains often overfit to training data due to inaccuracies in geometric estimation, underscoring the robustness of diffuse LiDAR in challenging regimes.

16. Low Lighting with Poisson Noise

In the main paper, we approximate low-lighting conditions by adding Gaussian noise. While this provides a reasonable approximation for many scenarios, Poisson noise is a more accurate model for very low-light conditions. To evaluate the robustness of our method under these conditions, we simulate Poisson noise in our experiments. Poisson noise is simulated by scaling pixel intensities by a factor, sampling noise from a Poisson distribution with the scaled intensities as the mean, and rescaling the noisy values back to the orig-

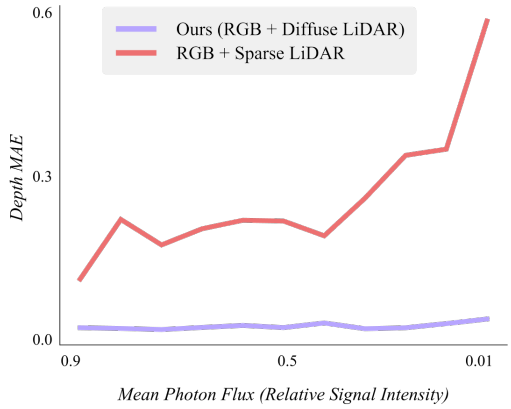

Figure 14. **Low Lighting Robustness under Poisson Noise.** Our method maintains robust depth and geometry estimation by dynamically weighting diffuse LiDAR signals more heavily as RGB signals become unstable in high-noise, low-light conditions.

Table 8. **RGB + Sparse LiDAR eventually reaches RGB and diffuse LiDAR performance with sufficient views.** We compare sparse LiDAR with varying numbers of input training views to our method, using RGB and diffuse LiDAR, using both 3 and 10 input views. We observe that our method has reasonable performance, relatively stable across texture variations, for even 3 input views, and improved geometry modeling with 10. RGB with Sparse LiDAR, on limited input views, has significantly worse geometry estimation, but eventually reaches our performance in about $2\times$ to $3\times$ the number of input views required for our method.

Method	Train Views	Blender D.MAE ↓	Blender N.MAE ↓	Chair D.MAE ↓	Chair N.MAE ↓	Hotdog D.MAE ↓	Hotdog N.MAE ↓	Lego D.MAE ↓	Lego N.MAE ↓
(a) Full Texture Datasets: Scenes with full texture on object and ground plane.									
Surfels RGB w/ Sparse LiDAR	3	0.239	70.06	0.287	73.41	0.189	70.39	0.220	73.68
Surfels RGB w/ Sparse LiDAR	6	0.113	47.99	0.120	53.56	0.106	50.26	0.116	50.77
Surfels RGB w/ Sparse LiDAR	10	0.058	24.74	0.057	23.44	0.054	29.49	0.058	30.63
Surfels RGB w/ Sparse LiDAR	15	0.048	24.51	0.060	25.60	0.054	28.25	0.050	31.12
Surfels RGB w/ Sparse LiDAR	20	0.041	21.43	0.040	19.38	0.039	24.88	0.043	26.99
Surfels RGB w/ Sparse LiDAR	25	0.035	19.35	0.031	17.20	0.028	21.57	0.032	24.41
Surfels RGB w/ Sparse LiDAR	30	0.026	16.20	0.018	12.53	0.021	17.83	0.027	20.86
Ours (RGB + Diffuse LiDAR)	**3**	0.086	31.54	0.110	38.16	0.072	34.72	0.093	33.88
Ours (RGB + Diffuse LiDAR)	**10**	0.025	19.17	0.017	18.05	0.016	24.62	0.025	25.27
(b) Textured Object Datasets: Scenes with textured objects on completely textureless ground planes.									
Surfels RGB w/ Sparse LiDAR	3	0.344	73.12	0.385	77.35	0.272	75.46	0.314	73.07
Surfels RGB w/ Sparse LiDAR	6	0.209	65.55	0.225	60.38	0.196	62.20	0.203	69.99
Surfels RGB w/ Sparse LiDAR	10	0.105	36.91	0.115	38.68	0.080	34.98	0.093	39.99
Surfels RGB w/ Sparse LiDAR	15	0.063	24.69	0.064	23.96	0.066	30.13	0.089	37.47
Surfels RGB w/ Sparse LiDAR	20	0.049	20.47	0.031	14.05	0.064	30.61	0.063	31.27
Surfels RGB w/ Sparse LiDAR	25	0.049	20.10	0.025	12.93	0.055	26.78	0.033	21.75
Surfels RGB w/ Sparse LiDAR	30	0.040	16.96	0.007	6.52	0.021	16.70	0.040	24.06
Ours (RGB + Diffuse LiDAR)	**3**	0.066	15.63	0.044	12.48	0.045	17.85	0.044	19.10
Ours (RGB + Diffuse LiDAR)	**10**	0.033	10.47	0.030	7.33	0.024	16.34	0.024	16.94
(c) Textured Plane Datasets: Scenes with completely textureless objects on textured ground planes.									
Surfels RGB w/ Sparse LiDAR	3	0.213	68.71	0.293	73.22	0.245	74.73	0.231	71.16
Surfels RGB w/ Sparse LiDAR	6	0.138	48.09	0.131	54.71	0.123	60.09	0.151	57.96
Surfels RGB w/ Sparse LiDAR	10	0.057	23.95	0.063	31.37	0.045	29.28	0.067	31.06
Surfels RGB w/ Sparse LiDAR	15	0.067	26.54	0.059	27.63	0.053	30.76	0.060	35.05
Surfels RGB w/ Sparse LiDAR	20	0.037	21.33	0.044	24.50	0.043	28.29	0.047	29.13
Surfels RGB w/ Sparse LiDAR	25	0.031	18.68	0.035	20.82	0.034	23.68	0.037	26.15
Surfels RGB w/ Sparse LiDAR	30	0.020	14.57	0.020	16.62	0.021	18.74	0.030	23.32
Ours (RGB + Diffuse LiDAR)	**3**	0.133	30.45	0.143	31.86	0.113	29.21	0.130	33.37
Ours (RGB + Diffuse LiDAR)	**10**	0.045	21.13	0.037	22.80	0.034	22.32	0.046	25.69
(d) Full Textureless Datasets: Completely textureless scenes.									
Surfels RGB w/ Sparse LiDAR	3	0.385	76.13	0.402	78.29	0.399	78.69	0.398	79.73
Surfels RGB w/ Sparse LiDAR	6	0.210	61.01	0.214	58.80	0.167	55.30	0.161	47.46
Surfels RGB w/ Sparse LiDAR	10	0.125	40.86	0.107	37.29	0.111	43.54	0.106	41.63
Surfels RGB w/ Sparse LiDAR	15	0.056	23.30	0.068	26.63	0.043	28.09	0.084	31.23
Surfels RGB w/ Sparse LiDAR	20	0.069	26.68	0.074	28.42	0.041	26.42	0.053	25.23
Surfels RGB w/ Sparse LiDAR	25	0.036	17.12	0.050	21.13	0.032	22.59	0.028	17.73
Surfels RGB w/ Sparse LiDAR	30	0.021	12.36	0.033	16.50	0.011	15.94	0.033	19.49
Ours (RGB + Diffuse LiDAR)	**3**	0.117	22.47	0.119	21.18	0.073	19.06	0.081	18.38
Ours (RGB + Diffuse LiDAR)	**10**	0.045	15.68	0.045	13.55	0.041	16.35	0.042	14.54

inal range. Lower scaling factors correspond to higher noise levels, as they effectively reduce the signal intensity.

Our results show that our method consistently estimates accurate depth and geometry, even under the noisiest conditions. This finding aligns with the results presented in the main paper, where we demonstrated robustness to added Gaussian noise. In very low-SNR, high-noise regimes, our approach effectively shifts reliance to the diffuse LiDAR signal, enabling robust reconstruction in challenging low-light conditions.