A Ablations

We include ablations to better understand the impact of: (1) the proposed use of *shadow transients* for shadow mapping, (2) number of illumination points, (3) single-photon lidar's temporal dimension, (4) light that has bounced *more than two* times, (5) modeling pulse shape, noise, and timing jitter, (6) amount of training data, and (7) out-of-distribution geometry at test time. We include quantitative results for these ablations in text below and qualitative results in Fig. 12, Fig. 13, and Fig. 14.

A.1 Shadow Mapping Approaches

Our approach draws on the idea of shadow transients to map lidar measurements to binary shadow maps for each illumination source. Rather than only using the multiplexed measurement as input, we also provide predicted two-bounce ToF. To understand the benefit of this approach, we compare it to the naive approach of using only the multiplexed input with supervised learning. Since there is no cue for which shadow to generate given a multiplexed measurement, we try two approaches: (1) predict shadows for all illumination points in a forward pass, (2) condition on illumination point and predict only the corresponding shadow in a forward pass. We find that the model is unable to learn accurate shadows with either of these approaches. While our proposed method – which uses predicted two-bounce ToF as an additional input – yields 0.0186 MAE and 0.959 IoU (Fig. 12, row 6), (1) yields 0.0984 and 0.788, respectively, and (2) yields 0.0896 and 0.799, respectively. Qualitative results for (1) and (2) are shown in Fig. 12, rows 3 and 4, respectively. We also tried explicitly computing the shadow transients by taking the absolute difference of the predicted two-bounce ToF transient and measured transient, and using this as input, which, while still accurate, results in slightly worse performance (0.0231 MAE, 0.950 IoU), as shown in Fig. 12, row 5. These results indicate that using shadow transient information – either implicitly or explicitly – is critical for accurate shadow mapping from multiplexed lidar measurements.

A.2 Number of Illumination Points

We vary the number of illumination points – using 4, 25, and 100 – and study its impact on depth estimation, specular surface segmentation, and shadow mapping (a proxy for occluded 3D reconstruction; i.e., inaccurate shadows lead to poor reconstruction). Intuitively, more illumination points increases ambiguity – as there are more unknown correspondences of peaks to illumination points. However, we find that this only reduces performance for shadow mapping – more illumination points results in higher performance when estimating either depth or specular surfaces. Results are shown in Fig. 11 of the main text. As more illumination points are added, there is more redundancy in depth information – since each illumination points provides a depth cue for all points light reflects to. Similarly, more illumination points increases the odds of a specular cue becoming available – since specular cues depend on light from a specular surface eventually being reflected back to the sensor. However, increasing the number of illumination points presents a serious challenge for the model to separate shadows, resulting lower performance. Thus, we hypothesize there is a *pareto optimum* that exists for number of illumination points, which results in higher depth, specular surface, and shadow accuracy. In our work, this optimum occurs at 25 illumination points. More work is needed to understand if the trend of depth improving with more illumination points continues with significantly more illumination points – on one hand, these additional illumination points provide redundancy in depth information, but, on the other, they increase the peak to illumination point correspondence ambiguities.

A.3 Intensity Only Measurements

One of the primary hypotheses of this work is that the temporal dimension of transient measurements contains information that can enable new advancements in 3D computer vision. Thus, for each of the tasks that we perform – estimation of depth, specular surfaces, and occluded geometry – we also conduct a baseline using an *intensity* image from a single-photon lidar. The intensity image is simply the sum of the transient along the temporal dimension. As expected, using the intensity image for training and inference on each task results in significantly worse performance than our method, which leverages the full information in the 3D transient. Using the intensity image results in 0.174 m mean absolute error for depth estimation, 0.703 IoU for specular segmentation, and 0.700 IoU for shadow mapping. While a significant drop in performance, interestingly, the intensity image still contains relevant cues for modest performance on each task, though we found the resulting shadow maps are not sufficiently accurate to perform 3D reconstruction.

A.4 Two-Bounce Only Measurements

We also study the impact of training on transients that only contain first and second bounce information to understand which signals the model has learned to exploit. We do this by re-rendering the transient dataset in MitsubaToF and setting *max_depth* to three (whereas the main dataset contains all bounces). We re-train our models for each task, allowing us to understand the importance of three or more bounces of light based on the change in performance per task. Depth MAE increases from 0.0228 to 0.0255 m (Δ0.0027 m), shadow mapping IoU increases from 0.954 to 0.964 (Δ0.01), and specular surface IoU drops from 0.865 to 0.767 (-Δ0.098). Thus, while three bounce has a minimal impact on depth and shadows, it has a significant impact on specular surface segmentation. This result matches our intuition that three-bounce signals can contain information about specular surfaces. While two-bounce signals may indicate a specular surface based on the presence of an extra measured peak at scene points that receive light reflected directly off a specular surface, three-bounce signals are empirically more helpful. In particular, we posit that the model has not only learned to exploit diffuse-specular-diffuse light paths, as done in past work, but also diffuse-diffuse-specular light paths, which may be a fruitful direction to investigate in future work. This hypothesis is based on the diffuse-diffuse-specular signal that is visually evident when watching the light-in-flight transient videos.

A.5 Noise and Timing Jitter

Since our method is trained with simulated measurements, we ablate its ability to work when trained on measurements with realistic pulse shapes, noise, and timing jitter. As done by Chen et al. [2020],

Fig. 12. **Shadow Mapping Approaches Ablation**. We compare different methods for predicting shadow masks from multiplexed lidar transients. All results are for measurements with 25 laser spots. We compare: supervised learning (transient → 25 shadows), supervised learning conditioned on laser spot index ((transient, index) → shadow), shadow transients (*abs*(2-Bounce ToF transient − measured transient) → shadow), and ours ((2-Bounce ToF transient, measured transient) → shadow). Results indicate that the use of shadow transients – whether explicit or implicit (ours) significantly improves results. Providing both the predicted 2-bounce ToF transient and measured transient as input to the network, rather than explicitly computing the shadow transient, slightly improves detail and performance, as shown by the regions in the red boxes.

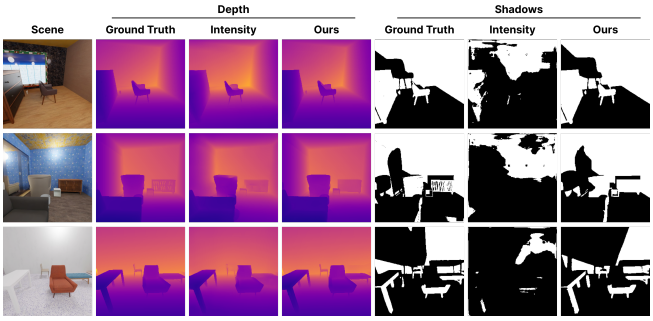

Fig. 13. **Intensity Only Ablation.** We compare depth and shadow estimation when using intensity images from the SPAD (obtained by summing along the temporal dimension) vs the full transient (ours). Using the full transient improves both depth and shadow results.

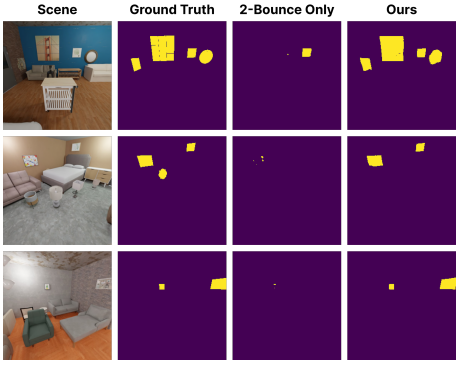

Fig. 14. **Two-Bounce Only Ablation.** We compare specular segmentation for models trained with transients rendered with only 1- and 2-Bounce peaks vs all peaks (ours). We randomly set pictures to be specular and find that only using 2-bounce information is not sufficient – thus, we posit this model also relies on 3-bounce information.

we follow the protocol established in Hernandez et al. [2017] for modeling realistic SPAD measurements. First, we convolve the rendered histograms with a pulse measured with a real-world sensor (MPD PDM Series). Next, we add Poisson noise and Gaussian timing jitter to the histograms. We sample the rate from a uniform distribution, leading to 2-bounce peak photon counts ranging from 10 to 400. We add 50 ps timing jitter (FWHM), which corresponds to 6.25 bins at 8 ps resolution. For this ablation, we use a small dataset containing geometric primitives to ease training time. The dataset consists of 10k training samples. Each scene also contains a mirror.

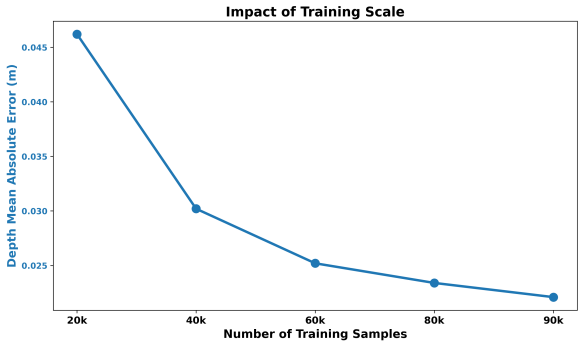

Fig. 15. **Ablation: Training Scale.** Given the scale of the proposed SB3D Dataset, we ablate the impact of training data scale on performance, focusing on depth accuracy. We find that as the amount of training data increases from 20k to 90k samples, the depth error consistency is reduced. Interestingly, the depth error reduces below the limit set by the temporal resolution of the sensor – meaning the model has learned precise correlations based on shape and appearance, rather than just relying on the timing information.

We demonstrate both accurate depth estimation (which can then be used to estimate two-bounce ToF) and shadow mapping using our method trained on this dataset. While our method is able to work in these conditions, we found that the 2D U-Net model was significantly less accurate than the "2.5D" U-Net alternative proposed in Appendix B. While the "2.5D" U-Net was not used in the main experiments due to higher training time, our use of a smaller and simpler dataset in this ablation, allowed us to use it. We posit that the 3D encoder enables the model to learn more robust temporal features, allowing it to generalize better under larger amounts of noise.

A.6 Impact of Training Scale

One of the contributions of this work is a large-scale simulated transient dataset. To ablate the impact of its scale on learning from multi-bounce signals, we vary the amount of training data used for our depth model and study the impact on performance. While past work in deep learning has extensively shown that data scale is correlated with performance, we use this ablation to confirm that intuition in the case of single-photon lidar data. We find that as we increase the amount of training data from 20k to 90k samples, depth estimation accuracy continues to improve, as shown in Fig. 15 of the main text.

A.7 Out-of-Distribution Generalization

We study the generalization capabilities of our model on out-of-distribution geometry. To do this, we rendered a new test dataset of multiplexed transients for 1,000 objects from Objaverse [Deitke et al. 2023] that don't appear in our training dataset (animals, humans, etc.). Each scene contains a cuboid room with an object placed on the ground. We tested our depth and shadow estimation models on this dataset. Depth MAE was 0.0201 m and shadow MAE and IoU were 0.0541 and 92.4% respectively, similar to previous in-distribution results (Tab. 1). We computed depth error only on object pixels (using an object mask), whereas shadow MAE/IoU were computed over all pixels.

B Implementation Details & Architecture

Pre-Processing. Our method utilizes the raw multi-bounce lidar measurement as input, which, in our work had a shape 256×256×637 in our main experiments and $256 \times 256 \times 375$ in our real-world experiments and noise ablation. We tried two methods for data normalization for the depth model: (1) reducing dynamic range by taking the log of each measurement followed by min-max normalization using the max intensity found over the entire dataset, and (2) min-max normalizing each histogram. While both were effective, we used (2) since it resulted in slightly better performance. For the shadow transient model, measured transients are min-max normalized and concatenated with the histogrammed predicted two-bounce ToF. For real-world data, the measurements were instead $256 \times 256 \times 375$ due to the differences in scene scale and temporal resolution.

Architecture. Although the focus of our work is not on architecture, we investigated the efficacy of different architectures for the proposed tasks, including 2D U-Net [Ronneberger et al. 2015], "2.5D" U-Net (3D encoder and 2D decoder with learned projections in each skip connection), SwinIR [Liang et al. 2021], NLOST [Li et al. 2023], and NLOSFeatureEmbeddings [Chen et al. 2020]. We found that 2D U-Net and "2.5D" U-Net had the best performance – with 2D U-Net training faster since larger batches could be fit on GPU. Thus, we used a modified 2D U-Net for all results. To accommodate the large size of our input, we added an initial feature extraction convolution to project 637 bins (or 375 for real-world data) to 128 channels before proceeding to the six U-Net encoder and decoder blocks.

Training. In simulation-based experiments, we trained three models: a ToF demultiplexing model, a shadow demultiplexing model, and a specular surface segmentation decoder (using the frozen ToF demultiplexing features). All models were trained for 200 epochs. The shadow model was trained with ground-truth two-bounce ToF data for the first 100 epochs and then with the noisier predicted two-bounce ToF data for the last 100 epochs. We found this curriculum learning strategy to be most effective to maximize accuracy. While the depth model and shadow model were both trained to generalize over scenes, the neural reconstruction method [Klinghoffer et al. 2024] was trained per scene.

Implementation. Our models are implemented in PyTorch [Paszke et al. 2019] and each trained on 8 NVIDIA H100 GPUs for around two days due to the size of the dataset used. We use the AdamW optimizer [Loshchilov and Hutter 2019] with an initial learning rate of 10^{-2} and weight decay of 10^{-3}.

C Training / Eval / Test Splits for SB3D Dataset

We train our models using the proposed dataset described in Sec. 5. We use 90% training split (87,688 samples), 3.9% validation split (3,744 samples), and 6.1% test split (6k samples). For 3D reconstruction, we train PlatoNeRF per scene using the predicted two-bounce ToF and shadows, as described in Sec. 4. In all experiments we assume 25 illumination points in a grid pattern, unless stated otherwise

(i.e., ablations on number of illumination points, noise ablation, and real-world experiments).

D SB3D Dataset Rendering Details

Meshes of indoor scenes from the Aria Synthetic Environments (ASE) dataset are used in this work [Avetisyan et al. 2024], with measurements rendered at the same poses as in the ASE dataset. Scenes in the ASE dataset contain objects from the Amazon Berkeley Objects dataset [Collins et al. 2022] placed via procedural generation to create realistic indoor layouts. Objects are assembled to mimic realistic indoor environments. These scenes were shown to be sufficiently realistic for real-world generalization of models trained on rendered RGB in past work [Avetisyan et al. 2024]. Besides single-photon lidar, all renders are created with Blender. Single-photon lidar transients are created with the physically-based MitsubaToF renderer [Pediredla et al. 2019b], which uses bidirectional path tracing with ellipsoidal connections to increase sampling efficiency. Due to the computational complexity of rendering single-photon lidar and the scale of the proposed dataset, rendering was parallelized over 1,000 CPU machines over around one week. All data is rendered at a resolution of 256×256 with a field of view of 90°. Multi-bounce lidar measurements are rendered with a temporal resolution of 128 picoseconds or ~0.0384 meters. All bounces of light (1, 2, 3, and more) are rendered. To reduce rendering time, time gating is used when generating the transient data (all scenes have a minimum depth of no less than 0.5 m and a maximum depth of no more than 4.5 m). To ensure all two-bounce paths are recorded, pathlengths between 1 and 25.46 meters are recorded, resulting in 637 bins per transient histogram. We set $n_{bounces}$ to -1 in MitsubaToF, meaning all bounces of light are rendered. Thus, the dataset can be used in future work that explores additional bounces. For any specular surface, we use the "roughconductor" BSDF in MitsubaToF and set alpha to 0.01. As a result, the lidar transients have either diffuse or specular surfaces (but not a gradient). We acknowledge this is a limitation of the dataset, as, in practice, many real-world objects may exhibit material properties with partial diffuse and specular components, however, in our real-world proof-of-concept experiments, we find this assumption is sufficient in demonstrating the potential for real-world generalization. For accessibility, the dataset is compressed to ~5 TB for release. We provide additional examples from the proposed SB3D dataset in Fig. 19.

Although not used in this work, the proposed dataset contains over 30 instance label categories that can be used in future work on instance segmentation, including everyday objects, such as desks, chairs, books, beds, pillows, weights, and many more.

Since assets from our dataset are rendered at the same pose as the ASE dataset, our dataset can easily be used with any assets from the ASE dataset in future work. Our dataset and all details on how to use it will be published in our project webpage.

E PlatoNeRF Background

Our method leverages PlatoNeRF [Klinghoffer et al. 2024], a recent method for single-view 3D reconstruction from two-bounce transients. In contrast to our approach, PlatoNeRF assumes a laser is scanned over the scene sequentially, capturing separate transients

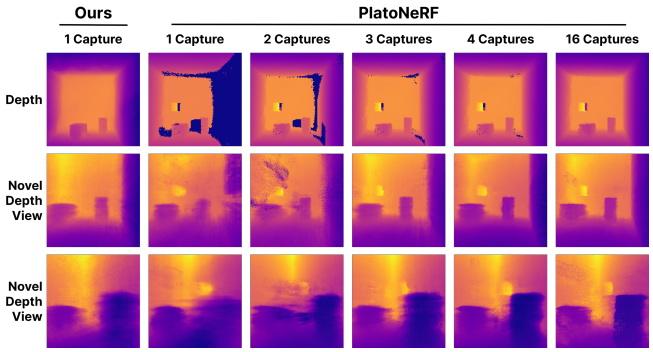

Fig. 16. **PlatoNeRF Performance with More Captures.** While our method outperforms PlatoNeRF in the single capture setting, we compare to PlatoNeRF as more points are scanned and used in training. We find that, as expected, PlatoNeRF accuracy increases as more captures (with different illumination points) are added. Shoot-Bounce-3D remains competitive – higher performance is especially noticeable in specular regions, though occluded regions have more floaters.

for each laser spot. Each transient is preprocessed into two-bounce ToF and shadow masks, which are used to supervise the learned densities via volume rendering.

PlatoNeRF is trained in two stages. First, depth from the lidar to the scene is learned by tracing primary rays with volume rendering. Since ground truth depth is not directly available, predicted depth is used to compute two-bounce ToF based on known illumination point and laser location (i.e. tracing the distance from the laser to the illumination point, from the illumination point to the predicted scene point location, and from the predicted scene point location to the known sensor location). In the second stage, secondary rays are also traced. Secondary rays originate at the end of primary rays and go to each illumination point. Intuitively, the full path of the secondary ray is only traveled if the measured point is *not* in shadow, else an object occludes the light from reaching the measured point. Thus, the secondary rays' transmittance values are supervised with the binary shadow masks extracted from the raw transient measurements.

Since PlatoNeRF uses two-bounce ToF and shadow masks to learn 3D scene geometry, the proposed pipeline naturally integrates with this approach for 3D reconstruction. Rather than computing these values from many non-multiplexed measurements, we instead use the values predicted by our models from multiplexed measurements, enabling single-shot 3D reconstruction.

F Real-World Experiments

In this section, we elaborate on the details of our real-world dataset and proof-of-concept results.

F.1 Model Training

For real-world validation, we retrain our models on a simulated dataset of scenes with a randomly placed cube, cylinder, and mirror in a room of varying scale. There are several reasons that motivate our use of a new training dataset for our real-world experiments. The original networks used (a) noiseless data, (b) scene scales too

large for our galvo ranges and lab space (1–4 vs 0.4–1 m), (c) different camera settings (90° vs 45° FoV, 128 ps vs 32 ps bins), and (d) 25 illumination points (we used 16 to cut acquisition time). While differences in scene scale can be partially mitigated by rescaling our measurements, this process is approximate (e.g. the original number of bins × scene scale delta may not equal the target number of bins). Re-rendering the entire dataset to close these gaps would have been computationally expensive. Instead, we rendered a smaller dataset with the same scale, camera, and illumination as real, and added noise. Using this dataset allowed us to validate that transient demultiplexing is feasible under realistic signal and noise. The simulated scene is illuminated at 16 points simultaneously – these points are in a grid pattern. When rendering, we randomly apply jitter to the camera origin and field of view for each scene. Each transient is rendered at 8 ps resolution and every four adjacent bins are summed to reduce the temporal resolution to 32 ps before training/inference. We choose 32 ps so that fine details can be resolved given the small scale of our scenes (the cylinder is one inch wide). Rendering with higher temporal resolution allows pulse shape and noise to be applied optionally before combining bins. The training dataset contained 10k samples, with an additional 2.5k for validation and 2.5k for test. We trained two sets of models – one with noiseless data and one with added pulse shapes, noise, and timing jitter (see Appendix A.5 for details).

F.2 Real-World Dataset

We capture a real-world dataset with scene geometry and sensor intrinsics/extrinsics that lie in distribution with the training data described above. We construct a room from diffuse white poster board and randomly place a foam cube and cylinder inside it, along with a mirror on the back wall. We illuminate each laser spot one at a time with a pulsed laser (Picoquant LDH-D Series) with 640 nm wavelength and a two-axis scanning galvonometer (Thorlabs GVS412). For each laser spot, we then scan a single-pixel SPAD (MPD PDM Series) over a 46° field of view using a second two-axis scanning galvonometer. This procedure results in sixteen 256 × 256 transients captured at 8 ps resolution. We sum adjacent bins in each transient to reduce the temporal resolution to 32 ps and add the sixteen transients together to create a multiplexed measurement. The light-in-flight video is shown in the supplementary webpage. Ground truth depth is captured from 1-bounce light by converting the setup shown in Fig. 8 to be confocal.

F.3 Results

Results are shown in Fig. 8. While we found our noiseless and noised models were both able to estimate reasonable depth, recovered shadows varied in quality, with some being highly accurate and others containing more artifacts, as shown in Fig. 17. To find the best shadows, we performed a grid search over model (noiseless, noised), amount of noise to subtract, and maximum histogram intensity (for clipping). We also tried applying a low-pass filter to the data and performing peak finding to reconstruct histograms with Diracs, but found that neither improved performance. We used the four best shadows (via manual selection), along with the 2-bounce ToF, predicted by our model to train PlatoNeRF for 3D reconstruction. In

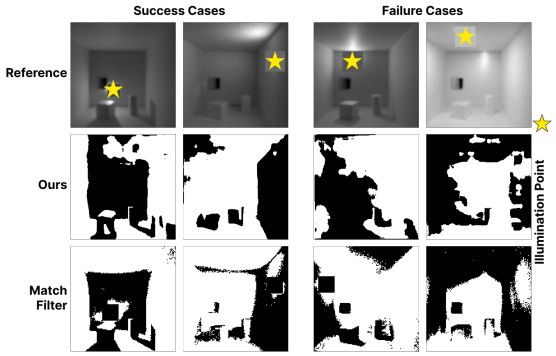

Fig. 17. **Shadow Prediction Quality.** While our is able to produce shadows sufficient for 3D reconstruction, some predicted shadows have significant artifacts, as shown in the two examples on the right. Below our predicted shadows we show shadow quality when using a match filtr on the non-multiplexed measurement from the individual illumination point.

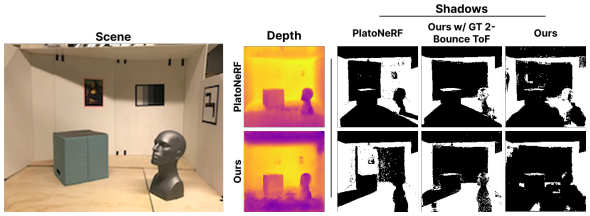

Fig. 18. **Limitations in Generalizability.** We test the generalizability of our models by testing them on an existing real-world dataset (from Bounce Flash Lidar [Henley et al. 2022]) with different scene scale, geometry, and spatial/temporal sensor resolution than seen in training. Our models are able to predict reasonable depth despite these differences. However, while some structure is maintained in predicted shadows, there are noticeable artifacts, especially in the region near the mannequin's head. If we use ground truth 2-bounce ToF, along with raw transients, as input to our shadow model, instead of predicted 2-bounce ToF, shadow quality improves significantly. This improvement suggests that a limitation of our work is the propagation of errors from the depth estimation model to the shadow model.

Fig. 8, we compare SB3D to PlatoNeRF trained when both are trained with only a single capture. Since PlatoNeRF is unable to handle multiplexed illumination, we instead train this PlatoNeRF model with a single illumination point. As shown in Fig. 16, as we increase the number of captures used to train PlatoNeRF, its performance improves. SB3D outperforms PlatoNeRF in the single capture setting and remains competitive with PlatoNeRF even when PlatoNeRF is trained with 16 captures. Training PlatoNeRF with 16 captures (by scanning a laser over different illumination points) serves as an upper bounce on 3D reconstruction. With 16 captures, PlatoNeRF exhibits slightly fewer floaters/artifacts in occluded regions than SB3D, but SB3D exhibits better performance in areas with specular objects due to its use of a data prior.

Limitations & Opportunities. Our real-world results demonstrate feasibility that the ideas proposed in this work can extend to real-world settings, In this section, we investigate generalizability and limitations of our model.

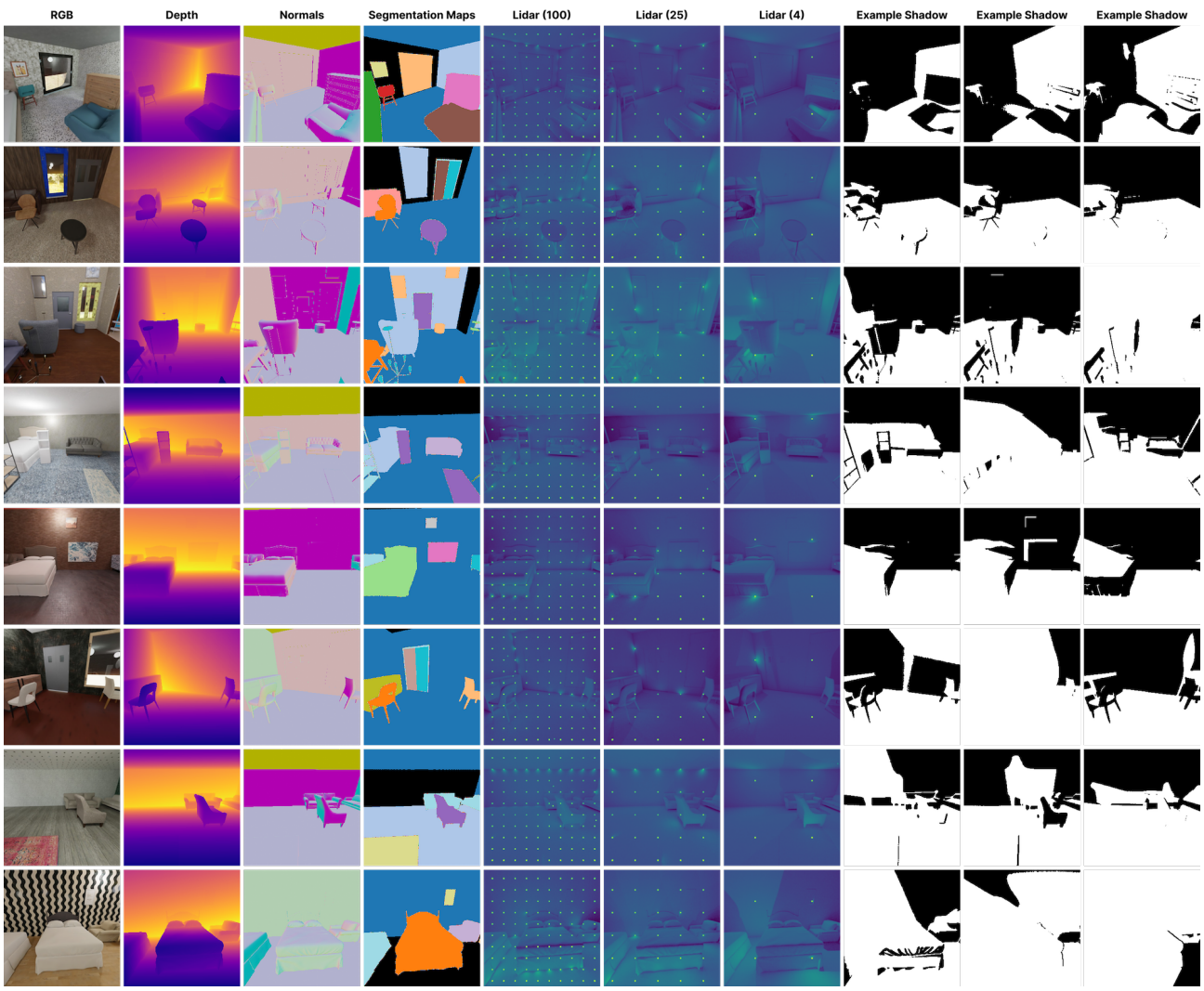

Fig. 19. **SB3D Dataset (Extended)**. We provide additional examples from our proposed dataset. In total, the dataset contains 97,432 examples, each rendered from a different scene.

We test our models' ability to generalize to another real-world dataset, from BF Lidar [Henley et al. 2022]. This test is challenging because the scene in this dataset has different scale and geometry (e.g. a mannequins head) than the scenes in our training dataset. In addition, the multiplexed measurement from BF Lidar has different spatial and temporal resolution than our models were trained with. Specifically, our models from Sec. F.1 were trained with 256×256 spatial resolution and 32 ps temporal resolution, whereas the BF Lidar data is 200×200 spatial resolution with 128 ps temporal resolution. In addition, the BF Lidar scene is illuminated at 16 random points, rather than in a grid pattern. To account for this, we retrain the models described in Sec. F.1 with random illumination points for every training sample, testing whether our model can not only generalize to a real-world measurement with different geometry and resolution, but also generalize to random illumination patterns.

To test our models on the BF Lidar dataset, we zero-pad the measurements and rescale the detected two-bounce peaks bins based on the difference in scene scale between training and test. Results are shown in Fig. 18. Despite the significant domain gaps, our model is capable of predicting reasonable depth, albeit with artifacts. The predicted shadows contain accurate regions, but also regions with significant artifacts. To understand the cause of these artifacts, we tried using ground-truth 2-bounce ToF, rather than predictions, – along with raw lidar transients – as input to our shadow estimation model. This experiment resulted in significant improvements in shadow quality. This finding suggests that the shadow models are also able to generalize to different geometries and sensor resolutions if given accurate 2-bounce ToF, but errors in depth estimation propagate and can significantly impact the shadow model.

Future work may explore different types of noise to add to the two-bounce ToF during training to improve robustness or ways to unify the first two stages of our approach. Other improvements may come from incorporating real-world data into training and investigating other ways to mitigate the sim-to-real gap.