DENALI: A Dataset Enabling Non-Line-of-Sight Spatial Reasoning with Low-Cost LiDARs

Supplementary Material

7. Dataset Information and Public Release

In total, DENALI contains 60 objects $\times$ 100 locations $\times$ 2 LiDAR spatial resolutions $\times$ 2 lighting conditions $\times$ 3 repeated samples, for a total of 72,000 LiDAR captures. Each capture consists of a full LiDAR histogram with synchronized tracking and co-located RGB-D measurements, along with metadata containing all estimated pose information for the capture set. Each capture also has a corresponding Mitsuba 3 digital twin; we do not directly include these digital twins in the dataset, but instead provide code that can be used to generate the twin for any capture scene/configuration. The overall on-disk size of the captured real data is 96.50GB, with 47.58GB at 3×3 resolution and 48.92GB at 8×8 resolution, as detailed in Tab. 4.

The full DENALI dataset is publicly available as an open-source release. This includes: (i) all captured LiDAR histograms, RGB-D data, and metadata containing pose estimations; (ii) code for generating digital twins for each real-capture configuration; (iii) code for all benchmarking experiments in the paper; and (iv) 3D object files for all hidden objects, along with CAD models for the object and sensor mounts used in our setup. The released dataset, code, and files are intended to enable independent reproduction of our measurements, re-training and evaluation of models, and development of new methods for NLOS perception with low-cost LiDARs.

Table 4. **DENALI on-disk size.** On-disk size (in GB) of our captured dataset, separated by capture resolution and modality captured. Overall, our dataset is 96.50GB (47.58GB at 3×3 and 48.92GB at 8×8 resolution).

Modality	3×3 [GB]	8×8 [GB]	Total [GB]
LiDAR histograms	0.18	1.22	1.40
Tracking RGB	10.01	10.20	20.21
Tracking depth	23.48	23.49	46.97
Co-located RGB	3.79	3.89	7.68
Co-located depth	9.92	9.92	19.84
Metadata	0.20	0.20	0.40
Total (all modalities)	**47.58**	**48.92**	**96.50**

8. Generalization and Robustness Analysis

We present additional experiments evaluating the generalization of our trained models from the main paper to stricter train/test stratifications, non-retroreflective objects, and unseen object variations.

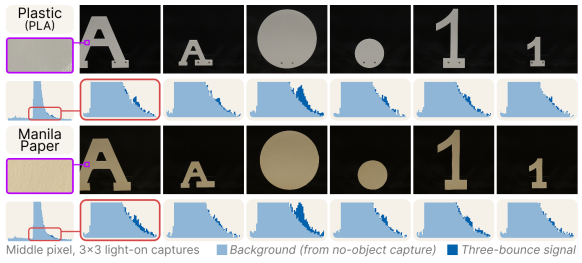
Train/Test Split Analysis. To assess potential limitations of randomized splits in our benchmarking (e.g., leakage, overfitting), we re-ran our 1D CNN (best-performing model) with stricter train/test stratifications. In Tab. 5, we report task evaluation on splits that (i) prevent leakage from repeated samples taken across captures, and (ii) evaluate on held-out location/shape/size sets. We observe stable results across these splits, with expected drops in harder regimes (e.g., train on 4 in. and test on 8 in.).

Table 5. **NLOS benchmarks (1D CNN) under different train/test splits.** Captures vary in shape, size, location, and lighting, with 3x repeats per setting. We compare the random split from the main paper to splits that group repeats within a split and hold out locations, shapes, or sizes from training. n/a* denotes ill-posed evaluations (e.g., classifying shapes unseen during training).

	Loc. Reg. ↓ RMSE / MAE	Shape Cls. ↑ Top-1 / Top-5 / F1	Size Cls. ↑ P / R / Acc
Random split (main paper)	Random 70/30 split across all shape/size/locs/lighting/samples		
	0.0456 / 0.0324	0.3876 / 0.7954 / 0.3832	0.9488 / 0.9468 / 0.9468
3x repeats grouped	Random 70/30 with 3x repeated samples *assigned together* in train/test		
	0.0488 / 0.0333	0.3848 / 0.7649 / 0.3632	0.9342 / 0.9328 / 0.9328
Location-held-out	70 training locations (across shape/size/lighting/samples), other 30 for test		
	0.0495 / 0.0367	0.3501 / 0.7217 / 0.3455	0.9190 / 0.9170 / 0.9170
Shape-held-out	21 training shapes (across size/locations/lighting/samples), other 9 for test		
	0.0429 / 0.0313	n/a*	0.8909 / 0.8882 / 0.8882
Size-held-out	4in. objects used for training (50% of samples), 8in. objects for test		
	0.0477 / 0.0337	0.0547 / 0.1854 / 0.2582	n/a*

Non-Retroreflective Object Signal. To demonstrate measurable NLOS signal from non-retroreflective (non-RR) objects within our capture setup, we created a small set of *plastic and manila paper* objects, each captured over 3 shapes $\times$ 2 sizes $\times$ 2 lighting $\times$ 100 locations. In Fig. 10, we visualize sample histograms showing clear three-bounce signal from these captures. We further evaluate 1D CNN task performance when trained on the base retroreflective dataset and evaluated on this paper+plastic set (Fig. 10); we observe reasonable localization and shape classification but poor size classification, likely due to lower three-bounce returns being confounded with small-size retroreflective objects. These results illustrate both the potential and limitations of applying RR-trained models to non-RR regimes.

Evaluation on Modified Shapes. To evaluate generalization to unseen object *variations*, we created three new 8 in. objects with *italic* styling (Fig. 11), each captured over 100 locations, and tested task performance when trained on the base dataset. As shown in Fig. 11, performance is reasonable but below the main paper baselines, as expected given the shift in shape and styling, indicating some generalization to object variations. These samples are included in the final dataset.



Train on Retroreflective Data → Evaluation on Combined Paper+Plastic Samples

Eval Set	Localization	Obj. Classification	Size Classification
Paper+Plastic	RMSE: 0.07541 MAE: 0.06169	Top-1: 0.1321 Top-5: 0.4033	Prec.: 0.5921 Rec.: 0.5317

Figure 10. **Three-bounce signal from non-retroreflective objects.** Top: sample histograms from plastic and manila paper objects showing visible three-bounce returns. Bottom: 1D CNN task evaluation when trained on retroreflective data and evaluated on non-RR objects. Localization and shape classification transfer reasonably, while size classification degrades.

Training on Original Objects → Eval. on Italic Objects

Localization	Obj. Classification
RMSE: 0.0415 MAE: 0.0304	Top-1: 0.2206 Top-5: 0.4028

Notes
- 8 in. italics only; size classif. omitted
- Top-k uses base (non-italic) classes as GT

Figure 11. **Generalization to modified object shapes.** Top: italic-styled variants of three objects (A, 1, circle) captured at 8 in. Bottom: 1D CNN evaluation when trained on base objects and tested on italic variants. Performance is reasonable but below main paper baselines, as expected with shape/style distribution shift.

9. Digital Twin Poses

Our dataset includes high-quality ground truth poses of the LiDAR, hidden objects, the relay wall, and the tabletop surface. To localize these features, we use AprilTag [37] markers of size 6cm from the tag36h11 family. We print the markers on matte paper and attach them rigidly to the table (IDs 10-15), relay wall (IDs 5-7), the LiDAR (ID 0) and the hidden object (ID 1). Using the RGB stream from bird's eye view tracking Realsense, we detect each marker position using contour-based quad detection, and estimate their 6-DoF poses using PnP.

From the 12,400 pose estimates obtained for each marker (one estimate per light-on capture for each object, location), we removed outlier detections with $|z| > 2$ along any position axis. We then computed the mean and standard deviation (σ) of the filtered marker positions to assess the precision of our ground truth. We show these computed marker positions in Tab. 6. For the hidden object (the only moving element in our setup) we compute these statistics across the 100 gantry positions. Fig. 12 illustrates the ground truth marker poses estimated during capture – the left panel shows a single instance of the detected tags and poses, whereas the right panel depicts the filtered marker positions over all the captures, illustrating the high precision of our ground truth.

Table 6. **Ground truth tag poses**. Mean and standard deviation (σ) of marker positions over all captures. The full tag poses and orientations, including for the hidden object at each gantry position, are included in the supplement folder.

ID	Component	# Estimates	Mean $\pm$ σ [cm]		
			x	y	z
0	LiDAR	11340	47.03 $\pm$ 0.02	25.42 $\pm$ 0.04	38.87 $\pm$ 0.05
5	Relay wall	11282	43.14 $\pm$ 0.06	80.39 $\pm$ 0.19	41.48 $\pm$ 0.14
6	Relay wall	11417	86.07 $\pm$ 0.02	82.72 $\pm$ 0.18	12.79 $\pm$ 0.17
7	Relay wall	11237	129.44 $\pm$ 0.12	84.21 $\pm$ 0.36	40.13 $\pm$ 0.17
10	Table	11307	0.07 $\pm$ 0.09	76.28 $\pm$ 0.13	-0.54 $\pm$ 0.13
11	Table	9531	145.92 $\pm$ 0.17	76.85 $\pm$ 0.26	1.65 $\pm$ 0.26
12	Table	11326	-0.50 $\pm$ 0.11	38.75 $\pm$ 0.13	-0.86 $\pm$ 0.17
13	Table	10772	147.23 $\pm$ 0.07	39.40 $\pm$ 0.08	1.98 $\pm$ 0.11
14	Table	11492	0.01 $\pm$ 0.16	-0.00 $\pm$ 0.08	0.01 $\pm$ 0.25
15	Table	11331	148.80 $\pm$ 0.11	1.86 $\pm$ 0.07	1.54 $\pm$ 0.17

10. Additional Benchmarking

We report additional benchmarking results for the 8×8 capture resolution using the 1D CNN model (our best-performing architecture from Sec. 4); results for 3×3 are reported in the main paper (Sec. 4). During training, we vary the training sets to encode different priors (e.g., known object size, illumination, or location). This setup emulates how such priors influence NLOS perception performance at 8×8 resolution, where the reduced photon count per pixel presents a more challenging regime.

Table 7. **Classification performance at 8×8 resolution under different conditions.** Top-1 accuracy, Top-5 accuracy, and Macro-F1 for object classification under different conditioning variables (size, lighting, and location) using the 1D CNN at 8×8 capture resolution. Performance remains low across all conditions, indicating that reduced photons per pixel at higher spatial resolution severely limits classification.

Condition	Test Top-1	Test Top-5	Test Macro-F1
Baseline (No Conditioning)			
-	0.0875	0.3255	0.0683
Size Conditioning			
8-inch	0.1083	0.4059	0.0914
4-inch	0.1104	0.3978	0.0945
Lighting Conditioning			
Light On	0.1372	0.4637	0.0953
Light Off	0.0476	0.2185	0.0090
Location Conditioning			
Front Quadrant	0.0756	0.2956	0.0380
Back Quadrant	0.0715	0.2974	0.0284

Tables 7-9 summarize classification and regression performance at 8×8 capture resolution under different conditioning variables. At 8×8, the baseline reaches only 53.7% size-classification accuracy (Table 8), with Top-1 object-classification accuracy of 8.8% and Macro-F1 of 0.07 (Ta-

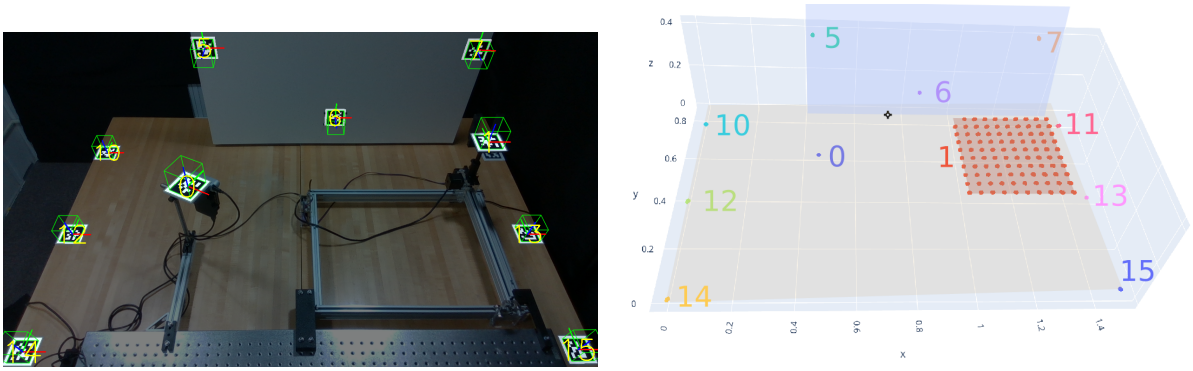


Figure 12. (left) RGB capture from Tracking RealSense stream and overlaid detected marker positions; (right) estimated marker positions in unified reference frame.

Table 8. **Precision, recall, and accuracy at 8×8 resolution under different conditions.** Precision, recall, and overall accuracy for size classification under object-type, lighting, and location conditioning at 8×8 capture resolution. Accuracy stays in the 0.52-0.59 range across all conditions, highlighting the impact of lower photon counts per pixel.

Condition	Test Precision	Test Recall	Test Accuracy
Baseline (No Conditioning)			
-	0.5372	0.5371	0.5371
Object Type Conditioning			
Letters	0.5647	0.5647	0.5647
Numbers	0.5917	0.5917	0.5917
Shapes	0.5695	0.5661	0.5661
Lighting Conditioning			
Light On	0.5315	0.5302	0.5302
Light Off	0.5526	0.5524	0.5524
Location Conditioning			
Front Quadrant	0.5219	0.5215	0.5215
Back Quadrant	0.5535	0.5530	0.5530

ble 7). Regression errors remain above 0.10 RMSE across all conditions (Table 9). In contrast, the 3×3 results reported in the main paper (Tab. 2) achieve substantially stronger performance across all tasks, indicating that the benefits of increased spatial sampling on these sensors do *not* outweigh the loss in captured photons per pixel for a given sampling exposure time.

Conditioning on scene variables provides negligible benefit at 8×8. Classification accuracy remains in a narrow band around 54% (Table 8), and Top-1 accuracy and Macro-F1 stay low across all conditioning choices in Table 7. The strongest conditioning effect is "Light On," which raises Top-1 from 8.8% to 13.7%, while "Light Off" drops performance further to 4.8% Top-1 and near-zero Macro-F1. Similarly, regression errors remain clustered just above 0.10

Table 9. **Regression performance at 8×8 resolution: RMSE and MAE.** Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) for localization regression under different conditioning variables at 8×8 capture resolution. Errors remain consistently high (RMSE ≥ 0.10) regardless of conditioning.

Condition	Test RMSE	Test MAE
Baseline (No Conditioning)		
-	0.1039	0.0899
Object Type Conditioning		
Letters	0.1043	0.0899
Numbers	0.1045	0.0905
Shapes	0.1047	0.0905
Lighting Conditioning		
Light On	0.1054	0.0914
Light Off	0.1058	0.0907
Size Conditioning		
8-inch	0.1046	0.0902
4-inch	0.1051	0.0913

RMSE regardless of object type, lighting, or size (Table 9). This plateau suggests a resolution-dependent regime: when photons per pixel are too low, performance is dominated by measurement noise and cannot be recovered simply by fixing scene parameters or exploiting higher spatial resolution.

Visualized capture resolution comparison. Figures 13 and 14 visualize the learned embeddings for three randomly selected object classes under the 3×3 and 8×8 capture configurations. At 8×8, the points form an almost uniform cloud with little apparent class structure, suggesting that the model cannot separate objects in feature space. In contrast, the 3×3 embeddings exhibit clear structure: points are more dispersed and form class-dependent groupings. This indicates that the data are both more distinguishable and better aligned with object identity at 3×3, consistent with the

performance reported in our benchmarking. In summary, our findings suggest that, for our class of low-cost LiDARs, the gains from added per-pixel photon returns (i.e., higher SNR) may significantly outweigh those from added spatial resolution.

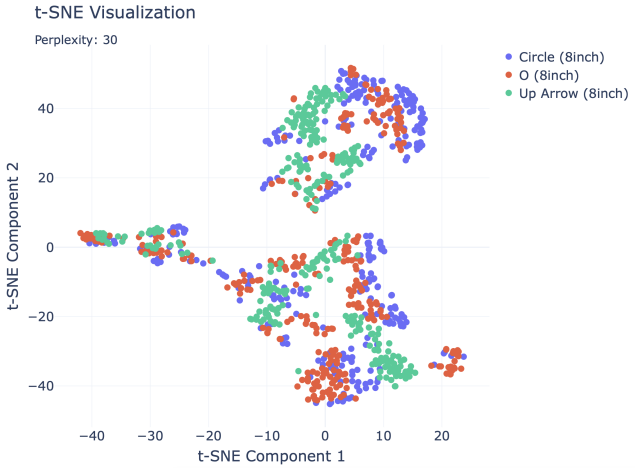


Figure 13. **t-SNE visualization under 3x3 capture resolution** We plot a visualization of a small portion of our data using a t-SNE projection. This data is a random selection of 8in. objects taken under lights off condition. We observe good separation between object classes which indicates high signal in 3x3 data.

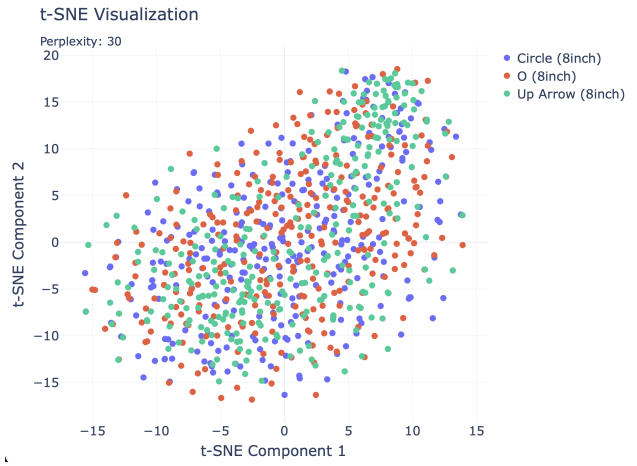


Figure 14. **t-SNE visualization under 8x8 capture resolution** We plot a visualization of a small portion of our data using a t-SNE projection. This data is a random selection of 8in. objects taken under lights off condition. We observe poor separation between object classes which indicates low signal in 8x8 data.

11. Scene, Modeling, and Training Details

11.1. Object Mount

Below we show an engineering drawing of the object mount used to secure our hidden objects (shown in Fig. 4) to the gantry in a standardized way. This mount defines a fixed geometric transformation between the detected tag (placed on the plate at the top of the mount) and the attached object. The drawing specifies all relevant dimensions needed for fabrication, and is included to support reproducibility, verification, and independent replication of our setup. We will include our 3D printed object and sensor mount designs with our publicly released dataset.

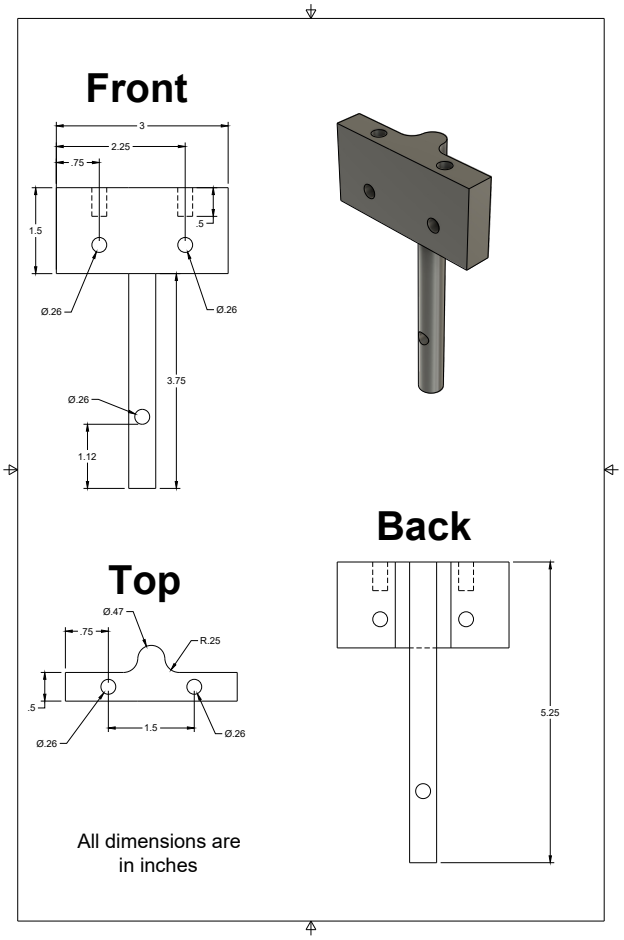


Figure 15. **Dimensions of 3D printed object mount.** We use this rigid object mount to define a fixed transformation between the detected April tag (Sec. 3.3) and the mounted object (Fig. 4).

11.2. Benchmarking Details (Sec. 4)

We report below the training specifications for each model used in the benchmarking experiments of Sec. 4 (shown in Tab. 2). We additionally describe key implementation details for each benchmarked model; we will include all our model implementations with our publicly released dataset.

LiDAR input representation. For each benchmarked model, our input is a LiDAR histogram $x \in \mathbb{R}^{B\times n\times n\times 128}$, for B the batch size, $n \times n$ the spatial resolution (either

Table 10. **Model Architecture and Training Details.** We report the number of parameters, training configuration, and computational requirements for each model architecture across our three DENALI perception tasks: location regression, object classification, and size prediction.

Model / Task	# Params	Train Size	Train Hours	Epochs	Batch Size	Test Size	Inference Hours
1D CNN							
Location Regression	82,340	25,200	0.0769	100	64	10,800	0.0003845
Object Classification	83,198	25,200	0.0850	100	64	10,800	0.0004
Size Prediction	82,274	25,200	0.0800	100	64	10,800	0.0004
3D CNN							
Location Regression	2,226,146	25,200	2.6792	75	64	10,800	0.0134
Object Classification	2,344,158	25,200	0.0703	200	256	10,800	0.0004
Size Prediction	2,226,146	25,200	2.5919	75	64	10,800	0.0130
MLP							
Location Regression	754,818	25,200	0.0161	75	64	10,800	0.0001
Object Classification	758,430	25,200	0.0150	100	256	10,800	0.0001
Size Prediction	754,818	25,200	0.0464	75	64	10,800	0.0002
Transformer							
Location Regression	3,228,802	25,200	0.9989	75	64	10,800	0.0050
Object Classification	3,232,926	25,200	0.6228	50	256	10,800	0.0031
Size Prediction	3,228,802	25,200	0.1261	10	256	10,800	0.0006

3x3 or 8x8), and 128 is the number of time bins. Unless otherwise noted, we use GELU activations and batch normalization throughout, and a final linear layer mapping the final feature vector to C task outputs.

1D CNN. For the 1D CNN, we treat the time-of-flight axis as the temporal dimension and the n^2 spatial bins as channels, reshaping the input to $(B, n^2, 128)$. We apply three Conv1d-BN-GELU blocks along the time axis:

- Conv1d: $n^2 \to h$, kernel size 5, padding 2;
- Conv1d: $h \to h$, kernel size 3, padding 1;
- Conv1d: $h \to h/2$, kernel size 3, padding 1.

Global average pooling over time yields a $(B, h/2)$ feature, followed by a 2-layer MLP with dropout and a final linear layer to produce $\hat{y} \in \mathbb{R}^{B\times C}$.

3D CNN. For the 3D CNN, we treat the time-of-flight axis as depth and apply 3D convolutions jointly over (D, H, W). We reshape the input to $(B, 1, n, n, 128)$ and use three Conv3d-BN-GELU blocks:

- Conv3d: $1 \to h$, kernel $(3, 3, 10)$, stride (s_D, s_H, s_W);
- Conv3d: $h \to h$, kernel $(3, 3, 10)$, stride $(1, 1, 1)$;
- Conv3d: $h \to h/2$, kernel $(3, 3, 10)$, stride $(1, 1, 1)$.

We apply global average pooling over (D, H, W) to obtain a $(B, h/2)$ feature, followed by the same 2-layer MLP and linear classifier as in the 1D CNN.

We make a special note that the object-classification task required meaningful deviations from our base 3D CNN architecture. We found that the full 3D CNN struggled with gradient propagation across three stacked 3D-convolution blocks. To address this, we reduced the model to a single 3D-convolution block with larger strides and kernels, while expanding the MLP head. These adjustments allowed for improved performance in significantly less training time, as shown in Tab. 10.

MLP. For the MLP baseline, we flatten each histogram $x \in \mathbb{R}^{B\times n\times n\times 128}$ to $(B, n^2 \cdot 128)$ and apply a 4-layer MLP

$$(n^2 \cdot 128) \to 512 \to 256 \to 128 \to C,$$

with GELU activations and dropout on the hidden layers.

Transformer. We treat each ToF bin as a token with n^2 spatial features and reshape the input to $x \in \mathbb{R}^{B\times L\times D_{\text{in}}}$ with $L = 128$ and $D_{\text{in}} = n^2$. A linear layer projects $D_{\text{in}} \to d$, after which we add learnable positional encodings and prepend a learnable class token. We apply N standard transformer encoder layers (pre-LN) and use the class-token output, followed by a 2-layer MLP with dropout and a final linear layer, to produce $\hat{y} \in \mathbb{R}^{B\times C}$.

11.3. Training Curves

We provide training curves for the 1D CNN trained on all 3×3 capture data. Figure 16 shows location-regression metrics, and Figure 17 shows object-classification metrics over the course of training.

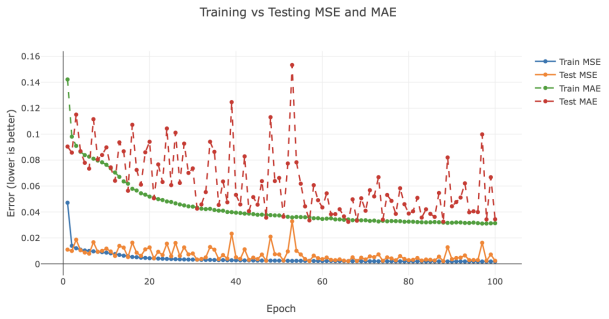


Figure 16. **1D CNN regression training curves.** Training and test MSE and MAE for the 1D CNN on 3×3 captures. The curves show a small, fluctuating train-test gap, indicating good overall generalization without severe overfitting.

11.4. Digital Twin NLOS Simulations

To study the sim-to-real gap for NLOS perception with low-cost LiDARs, we create a digital twin of our capture setup and render time-resolved transients using the MiTransient simulator [27]. For each real scene in DENALI, we use the matching hidden-object mesh, object size (4in / 8in), and discrete table location (pose computation described in Sec. 3.3), and place them in a virtual scene with the same relay wall, table, and sensor configuration as in the physical setup. Using MiTransient, we then simulate a pulsed LiDAR emitting a short laser pulse toward the relay wall and recording a histogram of photon-arrival times at the sensor.

Matching LiDAR instantaneous field of view. Our LiDAR, the ams TMF8828 [1], has a wide instantaneous field of view (iFoV) for each captured pixel [3, 21], but its exact

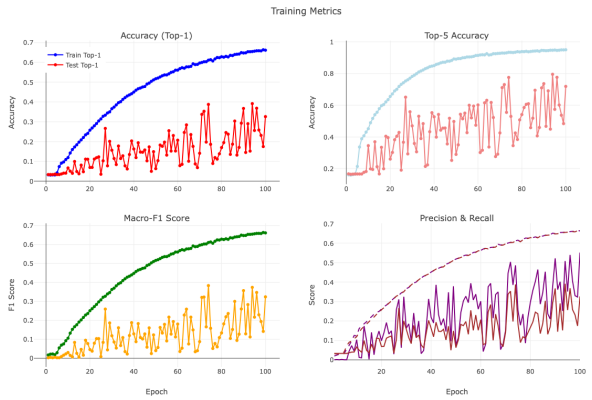


Figure 17. **1D CNN object classification training curves.** Training and test Top-1, Top-5, Macro-F1, precision, and recall for the 1D CNN on 3×3 captures. We observe reasonable, albeit noisy, generalization to the held-out test set during NLOS object classification model training.

spatial and angular response are not specified by the manufacturer and are difficult to calibrate directly. Instead of relying on an analytic iFoV model, we empirically match the hardware by manually aligning simulated and measured histograms. We focus on matching simulated measurements to the *center pixel* of the 3×3 LiDAR captures. In MiTransient, we create a relay wall and define a small wall patch corresponding to the expected footprint of this center pixel. We then adjust the size and location of this patch, as well as the distribution of sampled rays within it, until the simulated third-bounce response visually matches the measured center-pixel histogram. This procedure implicitly encodes the unknown iFoV and angular sensitivity of the real LiDAR into the simulation, without requiring an explicit calibration of the sensor's spatial response.

Temporal alignment, background subtraction, and cropping. We configure MiTransient to use the same effective temporal bin width as our LiDAR. The hardware operates over a 1.5m depth range discretized into 128 bins, corresponding to an optical-path-length bin size of $3/128$m (≈ 2.34cm OPL), or ≈ 1.17cm in depth and ≈ 78ps in time per bin. We apply a global offset in optical path length so that the dominant relay-wall peak in the simulation aligns with the corresponding peak in the real histogram. In both real and simulated data, we are primarily interested in the three-bounce returns induced by the hidden object. To isolate these returns, and to mitigate the effects of any inaccuracies in the simulated first bounce, we crop the histograms after the first-bounce wall return, discarding all earlier bins. The cropping point is selected so that (i) the direct relay-wall peak and its immediate neighborhood are removed and (ii) the leading edge of the remaining histogram corresponds as closely as possible between simulation and measurement. This cropping is fixed across the dataset and is chosen purely to improve alignment between the multi-bounce portions of the simulated and real histograms.

Our raw LiDAR measurements contain contributions from the static scene (relay wall, table, ambient illumination) in addition to the hidden-object signal. In practice, we hope DENALI provides a useful test bed to better simulate these returns; however, in our analysis, to reduce the impact of these nuisance terms and to better match the simulator, we perform a background-subtraction procedure for the real data: for each object, size, and location, we subtract a corresponding "no-object" capture from the "object-present" capture. This aligns with our simulation protocol, where we render only the hidden-object contribution as seen from the relay wall. In practice, we treat the wall patch viewed by the center pixel as the sensor aperture, then simulate the transient response with the hidden object present, and subtract the wall-only baseline. This yields a simulated histogram that is directly comparable to the background-subtracted real histogram and focuses on the same object-induced three-bounce signal.

MiTransient rendering. In MiTransient, we instantiate the relay wall as a rectangular Lambertian surface with physical dimensions matching the real wall (approximately 0.93×0.61 m) and attach a transient film of spatial resolution 64×64 and 128 temporal bins. Scene elements (wall, tables, hidden objects) are modeled as diffuse reflectors with fixed, spatially uniform albedos; the hidden objects are assigned a slightly higher reflectance to match returns in real-world captures. We use a transient NLOS path-tracing integrator, and draw 50,000 Monte Carlo samples per configuration to reduce variance in the three-bounce tail. To obtain a single histogram comparable to the center-pixel LiDAR response, we first render the full 64×64 transient on the relay wall and then average the simulated transient over a fixed, small spatial neighborhood around the calibrated center-pixel footprint. The resulting 1D histogram is then subjected to the same background subtraction, temporal alignment, and cropping steps described above before being paired with its real counterpart.

11.4.1. Qualitative Comparison

We show randomly sampled dataset captures and their corresponding rendered RGB and NLOS histograms (with background subtraction) in Fig. 18. Our NLOS rendering using MiTransient produces reasonable simulations that capture the temporal shifts observed in the real measurements and approximately match the peak shapes and relatively higher intensities (though the absolute photon count scaling is unknown). However, several key effects are not properly modeled: for smaller objects in particular, the lower-SNR signals are not well aligned with the real data (e.g., the first row in Fig. 18), and the exact pulse width, shape, and noise characteristics are not faithfully reproduced. We note that we subtract the first-bounce peak for

our renderings; the full NLOS histogram rendering including the first-bounce peak may introduce additional misalignment between simulated and measured histograms that we do not describe here.

11.5. NLOS Simulation Fidelity Eval. (Sec 5.1)

11.5.1. Learning Transform Functions

Given a simulated, cropped, and background-subtracted histogram $\boldsymbol{\tau}^{\text{sim}} \in \mathbb{R}^{128}$ generated in MiTransient [27], we define a sequence of parametric transform functions that progressively increase simulation fidelity: (i) a global intensity scaling, (ii) a temporal blur (pulse-shape) model, and (iii) a heteroscedastic noise model. Each transform operates directly on the simulated 128-bin histogram $\tau(\mathbf{x}', t)$, and is learned *directly from paired simulated/real histograms*.

Scaling. We first model global scaling between simulation and real captures. Given simulated histogram $\boldsymbol{\tau}^{\text{sim}}$, we compute a scaled version

$$\tau^{\text{scale}}[b] = a\tau^{\text{sim}}[b]^{\gamma}, \tag{3}$$

where b indexes time bins, and $a > 0$ and $\gamma > 0$ are scalar parameters. We estimate these parameters *once* by fitting all simulated histograms to real histograms via linear least squares on $\log \tau^{\text{sim}}$ and $\log \tau^{\text{real}}$, so that $\boldsymbol{\tau}^{\text{scale}}$ matches the response nonlinearity of the hardware.

Pulse Width. We next learn a 1D temporal blur transformation to account for the pulse shape of our real LiDAR. We represent the hardware's impulse response as a learned 1D kernel $\mathbf{k} \in \mathbb{R}^L$ with $L = 11$, and convolve it with the scaled histogram as $\boldsymbol{\tau}^{\text{blur}} = \mathbf{k} * \boldsymbol{\tau}^{\text{scale}}$, using centered padding so that the output remains 128 bins. We constrain $\mathbf{k}$ to be non-negative and sum to one via a softmax parameterization; we learn this kernel by minimizing the mean-squared error between simulated histograms and real histograms *after* applying our scaling transformation.

Noise. We finally learn a function to represent sensor noise using a heteroscedastic Poisson-Gaussian process. We treat the blurred signal $\boldsymbol{\tau}^{\text{blur}}$ as the underlying mean $\boldsymbol{\mu}$ and draw a Poisson realization with rate $\boldsymbol{\mu}$; we then add zero-mean Gaussian noise whose variance is modeled as:

$$\text{Var}\left[\tau^{\text{noise}}[b] \,\middle|\, \mu_b\right] \;\approx\; a_{\text{noise}}\,\mu_b + b_{\text{noise}}. \tag{4}$$

with $a_{\text{noise}} \geq 0$ and $b_{\text{noise}} \geq 0$ fit by regressing residuals between simulated and real histograms. We enforce that the total variance is *at least* the underlying Poisson variance.

11.5.2. Evaluation

We evaluate simulation fidelity and sim-to-real transfer on the 2D NLOS localization task. We split simulated and real (SPAD) histograms into disjoint training and test sets with a 70/30 ratio, and train a 1D CNN with mean squared error (MSE) loss for a fixed number of epochs, using identical optimization hyperparameters across all experiments. To quantify the impact of simulation fidelity and real-data augmentation, we train a fresh 1D CNN for each simulation domain using *only* simulated data at that fidelity level (raw simulation, scaled, scaled+blurred, and scaled+blurred+noisy). We then progressively augment each simulated training set with real SPAD histograms, adding 10-200 real samples in fixed increments. Every trained model is evaluated on the same held-out SPAD test set, and we report MSE, RMSE, and MAE on this real-data test split. The curves in Fig. 9 plot localization RMSE as a function of both the simulation variant and the number of real samples used during training.

11.6. Task-Specific Sensor Design (Sec 5.2)

In Sec. 5.2 we aim to motivate using DENALI to inform sensor design for NLOS perception from low-cost LiDARs. Specifically, we use the real, background-subtracted histograms from DENALI as a "best-case" reference, and synthetically degrade their temporal resolution to understand how this degradation impacts our NLOS perception tasks. Each original histogram contains 128 bins with temporal bin width $\approx$ 78ps.

We synthesize increasing levels of *timing jitter* on these original histograms. Timing jitter is modeled as a convolution along the time axis with a normalized discrete Gaussian kernel, parameterized by its full width at half maximum (FWHM) measured in histogram bins. Given a target FWHM (in bins), we convert it to the corresponding kernel standard deviation and construct the discrete Gaussian; this kernel is then applied via linear convolution, yielding a temporally blurred histogram that reasonably mimics the instrument response of a LiDAR with worse timing jitter. We sweep a range of timing jitters corresponding to approximately 50, 100, and 600ps FWHM in time, and deterministically apply the same kernel to all histograms for each configuration.

To quantify how this synthetic timing degradation affects downstream NLOS perception, we train neural networks on the jittered real data for our three NLOS tasks (hidden-object localization, object classification, and size classification). For each jitter setting, we apply timing jitter to the set of 1D histograms (using only the center pixel from our 3×3 light-on captures) and split the resulting dataset into train/validation/test sets. We then train an independent 1D CNN for each degraded histogram set. Because these experiments operate directly on real LiDAR histograms with

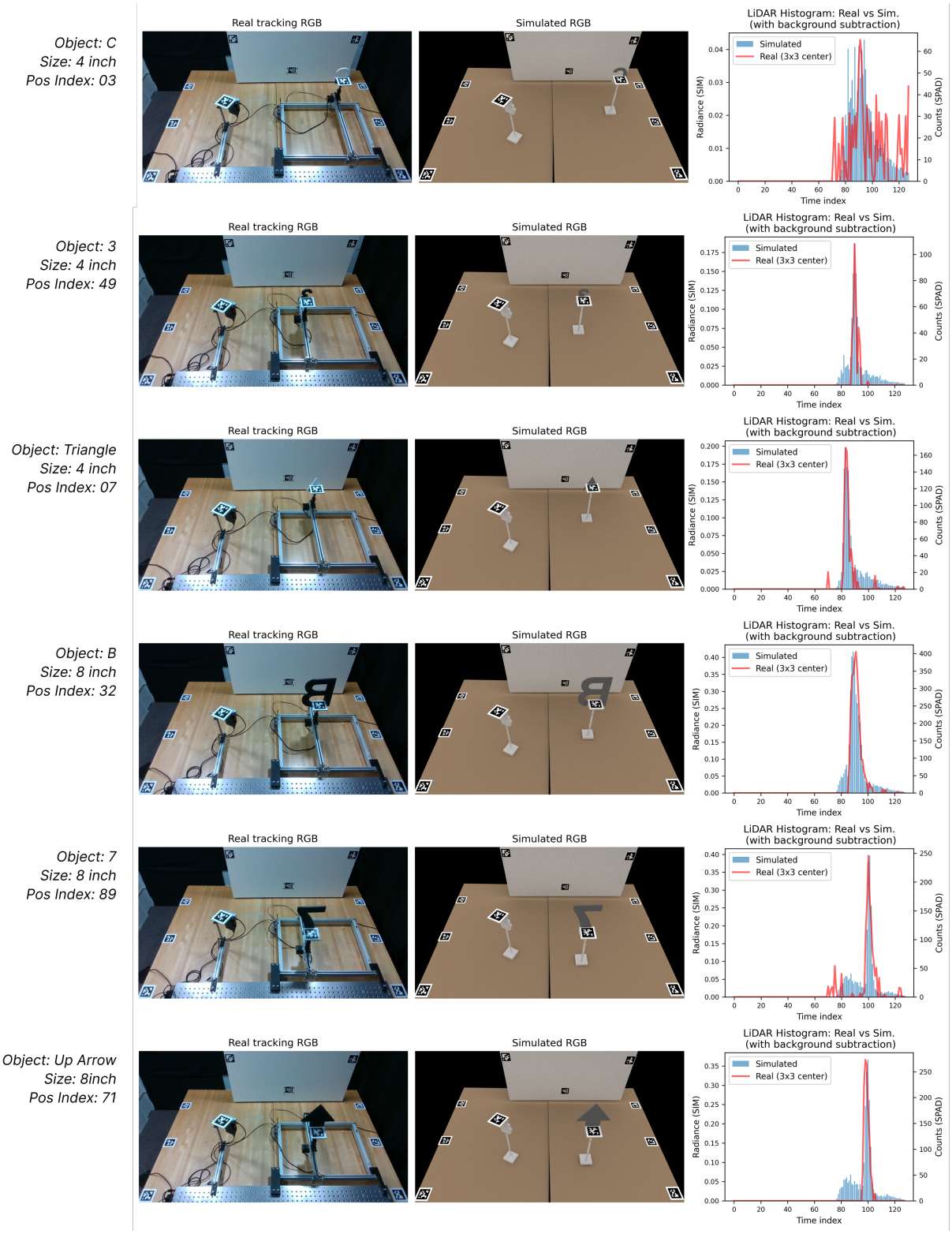


Figure 18. **Sample rendered RGB scenes and corresponding LiDAR histograms (three-bounce signal only).** (Left) real RGB images captured using the tracking RealSense camera, (Middle) rendered RGB views of the calibrated scene using Mitsuba 3, (Right) real LiDAR histograms from the 3×3 center pixel overlaid with simulated transients from our NLOS renderer. The simulated results qualitatively reproduce temporal structure and relative intensity trends (without enforcing absolute scaling), but do not capture several real-world effects including noise from ambient lighting, timing jitter, pulse shape distortion, and sensor noise characteristics.

controlled synthetic timing jitter, we believe this approach provides a useful path toward identifying empirical upper bounds on what forms of NLOS perception are achievable for this class of low-cost LiDARs.

12. Impact and Limitations

Impact. We introduce **DENALI**, the first large-scale real-world dataset of histograms from low-cost LiDARs for non-line-of-sight perception. We capture 72,000 time-resolved LiDAR histograms with synchronized RGB-D and tracking information, providing a controlled but realistic testbench for studying the NLOS capabilities of consumer dToF sensors. Our results show that, in favorable regimes, commodity LiDARs can support accurate data-driven NLOS tasks such as object classification and location prediction, without additional specialized hardware. We view this as a step toward practical, deployable NLOS vision with data-driven methods, with potential applications in robotic safety, X-ray-like perception on mobile devices, and extended ego and environment awareness in AR/VR settings.

Limitations. Our data is collected in a real but controlled indoor environment. Scenes are designed to elicit strong multi-bounce returns: objects are retroreflective, constrained to a known region, and viewed from a fixed sensor, table, and planar relay wall. This controlled setup allows us to systematically vary object type, position, size, lighting, and LiDAR resolution to characterize consumer sensors under best-case conditions. However, it does not capture the full variability of cluttered, dynamic, or outdoor environments (e.g., non-planar relays, non-retroreflective materials, moving occluders). Extending **DENALI**, and related perception techniques, to more unconstrained NLOS scenes remains an important direction for future work.

We use a single low-cost dToF LiDAR model (ams TMF8828 [1]), which is representative of, but not identical to, other low-cost consumer LiDARs (e.g., [33]). As a result, the quantitative NLOS performance reported here may not directly transfer to sensors with different optics, wavelengths, or noise characteristics. Our benchmarks focus on a specific NLOS configuration (single hidden object, fixed relay-wall geometry) and on a fixed set of downstream tasks; we do not address the broader class of 3D perception problems, including full 3D reconstruction, multi-object reasoning, long-horizon tracking, or dynamic motion estimation; nevertheless, we hope **DENALI** serves as a useful starting point for exploring these extended NLOS tasks with low-cost LiDARs.

Although each capture is paired with a Mitsuba 3 digital twin, our simulations approximate the true sensor and scene physics. Our experiments are intended to *highlight sim-to-real gaps* that limit direct transfer of models trained purely in simulation to real hardware. The released code for generating digital twins is therefore meant as a basis for future work on higher-fidelity sensor models, domain adaptation, and joint optimization of sensors and algorithms. Addressing these gaps is essential for scaling NLOS perception beyond the specific hardware and capture conditions used in this work.