

Supercharging Floorplan Localization with Semantic Rays

—Supplementary Material—

Yuval Grader
Tel Aviv University

Hadar Averbuch-Elor
Cornell University

Contents

1. Additional Details	1
1.1. Network Architecture and Design Choices	1
1.2. Training Settings and Hyperparameters	1
1.3. Dataset Descriptions	2
1.4. Baseline Methods	2
1.5. Additional Implementation Details	2
1.6. System Configuration	2
2. Additional Ablation Studies	3
2.1. Effect of Room Polygon Usage	3
2.2. Impact of Top-K Candidate Selection on Test Set Performance	3
2.3. Top-K Location Distribution Analysis	3
2.4. Ablation on Recall With Different δ_{res}	4
2.5. Integrating Our Refinement into F3Loc	5
3. Additional Experiments and Analysis	5
3.1. Probability Volume Fusing Weights	5
3.2. Room Type Classification Results	6
3.3. Effects of Refinement Parameter Choices	7
3.4. Additional Comparison with LASER	7
3.5. Comparison against Soft Constraints	7
4. Additional Visualizations	8
4.1. Qualitative Visualizations	8
4.2. Visualization of Baseline Comparisons	8
5. Limitations	8

1. Additional Details

In this section, we provide detailed information on the network architecture, training procedure, evaluation pipeline, baselines, dataset handling, and parameter settings used in our experiments.

1.1. Network Architecture and Design Choices

Our model adopts a ResNet50 backbone pretrained on ImageNet to extract features from the input RGB image. The extracted feature map (of dimension 2048) is then reduced to 128 channels via a convolution followed by batch normalization and ReLU activation (implemented in our custom `ConvBnReLU` module). These features are further projected to a 48-dimensional space using a linear layer.

To preserve spatial information, a positional encoding is computed from normalized (x, y) coordinates using a small MLP with a `Tanh` activation. Two sets of learnable query tokens are introduced:

- A single CLS token for predicting a global room-type label.
- 40 ray tokens for predicting semantic rays.

Both sets of tokens attend to the flattened spatial features using a single-head cross attention module. The ray tokens are additionally processed by a self-attention block (with residual connections and a feed-forward network) followed by an MLP to produce per-ray logits over semantic classes. The room token is processed similarly to yield room type logits.

1.2. Training Settings and Hyperparameters

As mentioned in the main paper, our semantic network is implemented within a PyTorch Lightning module to perform multi-task predictions, simultaneously producing 40 semantic ray outputs (one per ray) and one global room-type label. During training, the predicted ray outputs (with shape $(N, 40, \text{num_ray_classes})$) are supervised via cross-entropy loss against the ground-truth semantic labels (shape $(N, 40)$), while the global room-type prediction (with shape $(N, \text{num_room_types})$) is similarly trained using cross-entropy loss. The overall loss is defined as the sum of these two components. We optimize the network using the Adam optimizer with a learning rate of 1×10^{-3} and a batch size of 16.

1.3. Dataset Descriptions

Additional dataset processing details are provided here for clarity.

S3D We use the fully furnished, perspective dataset of Structured3D (S3D) with the official splits and processing protocol.

ZInD For ZInD, we follow the official splits and prior works to generate a fixed-size dataset by cropping each panorama to a single 80° FoV, 0° yaw perspective image.

1.4. Baseline Methods

We compare our method against several baselines to assess its performance under a consistent evaluation protocol.

F3Loc For the F3Loc baseline, we use the publicly available code from the official repository and made a some modifications to the way we calculate rays and identify walls for the ZInD dataset, but. For the S3D dataset, we report the official paper results as we operate on the exact same data split and processing protocol. For the ZInD dataset, we evaluate F3Loc by running its training and inference using the provided code and configuration.

LASER For the LASER baseline, we use the official implementation available from the authors. Since the provided code runs on both datasets, we execute LASER as-is. For S3D, we follow F3Loc by evaluating on the official fully furnished perspective dataset. For ZInD, we run the official training and evaluation code while adjusting the configuration to crop the panoramas to an 80° FoV and to disable random view augmentations, as detailed in Section 1.3.

1.5. Additional Implementation Details

1.5.1. Semantic Interpolation via Majority Voting

As described in the main section, we introduce a majority voting algorithm to interpolate the predicted l semantic rays into a smaller subset. As shown in our ablation study, this interpolation alone yields a 4.2% improvement in 1m recall. The detailed algorithm is provided in Algorithm 1.

1.5.2. Ray Similarity Measurement

To assess the alignment between the predicted rays and the candidate rays in our refinement procedure, we compute a similarity score that combines both depth and semantic discrepancies. Specifically, we calculate the L1 distance between the predicted depth rays and the candidate depth rays to capture the geometric error, and we

Algorithm 1 Semantic Ray Interpolation with Majority Voting

Require:

- 1: 1. A semantic ray vector r of length N .
2. Field-of-view $\text{fov} = 80^\circ$.
3. Desired number of rays N_d .
4. Desired angular gap $\Delta\theta$.
5. Window size w for majority voting.
- 2: Compute the angle between original rays: $\Delta\alpha$.
- 3: Compute the center index: $c \leftarrow \lfloor N/2 \rfloor$.
- 4: Initialize an empty semantic ray vector r_{interp} .
- 5: **for** $i = 0$ to $N_d - 1$ **do**
- 6: Compute the desired angle relative to the center:

$$\theta_i \leftarrow (i - \lfloor N_d/2 \rfloor) \times \Delta\theta.$$

- 7: Compute the index offset:

$$o \leftarrow \frac{\theta_i}{\Delta\alpha}.$$

- 8: Determine the target index:

$$\text{idx} \leftarrow \text{round}(c + o).$$

- 9: Collect neighbor labels:

$$\text{neighbors} \leftarrow \{r[j] \mid j = \text{idx} - w, \dots, \text{idx} + w\}.$$

- 10: Determine the majority label l^* .

- 11: Append l^* to r_{interp} .

- 12: **end for** **return** r_{interp} .
-

compute a semantic error as the mean mismatch between the predicted semantic labels and the candidate semantic labels. These two error metrics are then combined using a weighted sum:

$$\text{score} = \alpha \cdot \text{depth_error} + (1 - \alpha) \cdot \text{semantic_error},$$

where the depth error is computed as the average absolute difference between corresponding depth values, and the semantic error is quantified as the average binary mismatch between semantic labels. In all our experiments, we set α equal to w_d , the weight assigned to the depth probability volume in our fusion equation.

1.6. System Configuration

All training experiments were conducted on a virtual machine with the following specifications:

- **CPUs:** 12 cores (Intel Xeon E5-2690 v4 @ 2.60GHz)
- **GPU:** Tesla V100-PCIE GPUs (with 16GB memory each)

These hardware details ensure reproducibility and highlight the computational resources available during

training.

2. Additional Ablation Studies

In this section, we present a series of ablation studies to evaluate key components of our localization pipeline. In Section 2.1 we analyze the impact of using external room-polygon masks. Section 2.3 examines the effect of varying Top-K candidate selections and refinement parameters. Finally, in Section 2.4 we investigate the influence of the refinement threshold δ_{res} on balancing fine and coarse localization accuracy.

2.1. Effect of Room Polygon Usage

As part of our usage of room-polygon masks, we also compare the performance of using external house-area masks versus not using them. As shown in Figure 1, we use a mask to exclude points from outside the house. This avoids matching windows and corners that lie beyond the interior. Notably, when the highest-probability location is masked out, the next best match is closer to the ground-truth location, yielding an improvement.

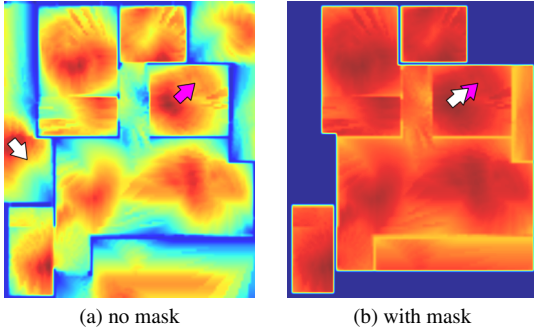


Figure 1. Comparison of the scene without mask (a) and with mask (b).

Table 1 presents a comparison of the recall obtained by our method with and without external masking, demonstrating that this procedure does not yield any substantial gains.

Mask Setting	0.1 m	0.5 m	1 m	1 m 30°
with	5.63	45.67	59.36	57.82
without	5.13	45.07	59.24	57.61

Table 1. Comparison of localization accuracy on S3D with and without external house-area masks.

2.2. Impact of Top-K Candidate Selection on Test Set Performance

We further analyze our coarse-to-fine approach by conducting an experiment to evaluate the effect of selecting

TopK	Method	0.1m	0.5m	1m	1m 30°
Top1	No refine	4.65	38.35	49.40	48.44
	Ours _s	4.73	38.35	49.59	48.59
	Ours _r	5.29 11.84%↑	42.81 11.63%↑	55.76 12.44%↑	54.30 11.74%↑
Top2	Ours _s	4.96	41.08	52.20	51.39
	Ours _r	5.48 10.48%↑	45.31 10.30%↑	58.43 11.93%↑	57.19 11.28%↑
	Ours _s	5.23	41.27	52.96	52.04
Top3	Ours _r	5.34 2.10%↑	45.24 9.63%↑	58.77 11.00%↑	57.28 10.07%↑
	Ours _s	5.42	41.87	53.52	52.61
Top5	Ours _r	5.70 5.17%↑	45.53 8.74%↑	58.78 9.83%↑	57.49 9.28%↑

Table 2. Ablation study on the coarse-to-fine Top-k selection in the S3D dataset, evaluating the location extraction module and the effect of room type prediction in our pipeline. Recall metrics (in %) for our methods (Ours_s and Ours_r) are reported. For each metric, the improvement is shown to the right of the Ours_r score in dark green with an upward arrow indicating the relative improvement over Ours_s.

different numbers of Top-K candidates. Table 2 details the impact of various Top-K values on the localization refinement. We observe that as k increases, the overall localization accuracy improves. In particular, the largest improvement is achieved when increasing from Top-1 to Top-2 candidates, which is sensible since over 70% of the ground-truth locations lie within the Top-1 and Top-2 candidate set. Beyond Top-2, while further increases in k yield additional improvements, these gains are minor compared to the initial boost. This is likely due to prediction errors and noise. As k increases, additional candidates may include rays that were previously interpolated out, leading to mislocalizations when they are erroneously matched.

2.3. Top-K Location Distribution Analysis

To better understand the effectiveness of our coarse-to-fine strategy, we conducted an in-depth study on the impact of selecting the Top-K candidate poses and on the localization accuracy. For simplicity of this analysis, no angular augmentations were applied in this analysis. all data were collected from the S3D test dataset using the following parameters: $\delta_{\text{res}} = 1$ m, $\delta_{\text{ang}} = 0^\circ$, and $\Delta_{\text{max}} = 0^\circ$.

Figure 2 presents the candidate ranking distribution. In 51.1% of cases, the Top 1 candidate is closest to the ground truth, while the second and third candidates account for 19.4% and 12.7% of cases, respectively. In this analysis, we maintain a 1 m exclusion radius around each candidate to emphasize strong mismatches. This motivates refining the Top-K candidates instead of relying solely on the Top-1 candidate during the coarse stage.

Furthermore, Figure 3 shows that approximately 90% of the localization improvements occur when a candidate is shifted by more than 0.5 m relative to the highest-scoring candidate ($K=0$) in the structural-semantic probability volume. This finding reinforces the benefit of selecting the best candidate among the Top-K predictions.

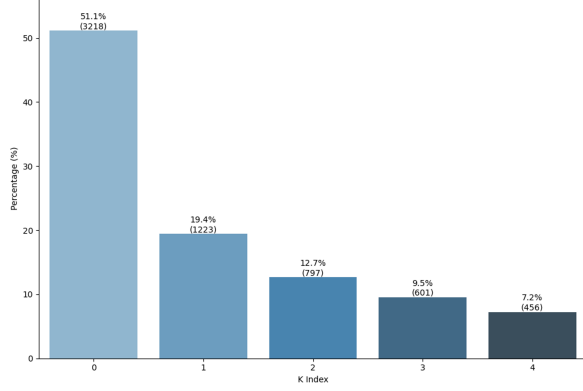


Figure 2. Distribution of the best candidate index on the S3D test set. The Top 1 candidate is closest to the ground truth in 51.1% of cases, followed by the second and third candidates.

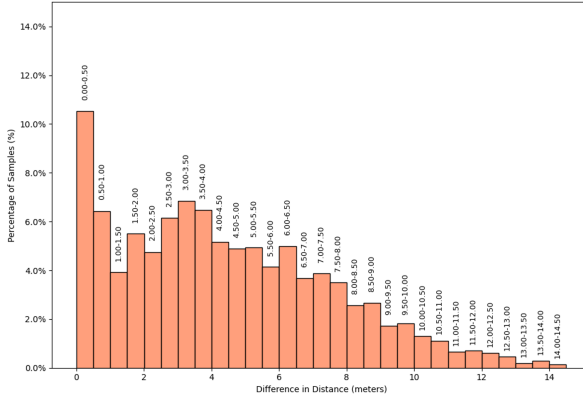


Figure 3. Histogram of distance improvements for Top-K selections. Approximately 90% of the improvements exceed 0.5 m compared to the highest-scoring candidate (K=0).

Figure 4 illustrates the discrepancies between semantic and depth ray predictions when the top candidate (K0) is not the best match. The trend of decreasing sample percentages with increasing differences in the semantic rays confirms that even small changes in semantic cues are critical for accurate localization. This effect is also evident when a semantic label resolves ambiguity between two structurally identical environments, further emphasizing the importance of integrating semantics into the localization process. Note that we consider two depth rays to be identical if they differ by less than 10 cm.

In Figure 5 we illustrate the impact of different Top-K selections on localization accuracy. In many cases, especially in environments with repetitive patterns, the Top-1 candidate does not necessarily correspond to the correct prediction (as can also be seen quantitatively in Figure 2).

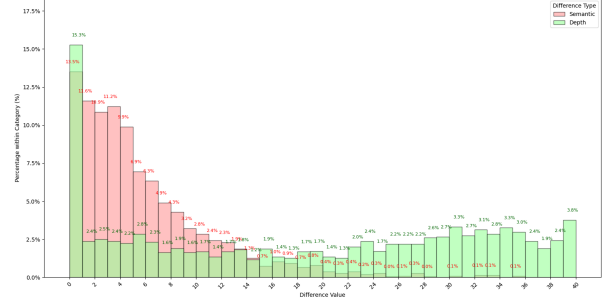


Figure 4. Semantic and Depth Ray Differences. The Y-axis represents the percentage of samples, and the X-axis indicates the number of ray differences between the top candidate (K0) and the best candidate, with **depth** differences shown in green and **semantic** differences in red.

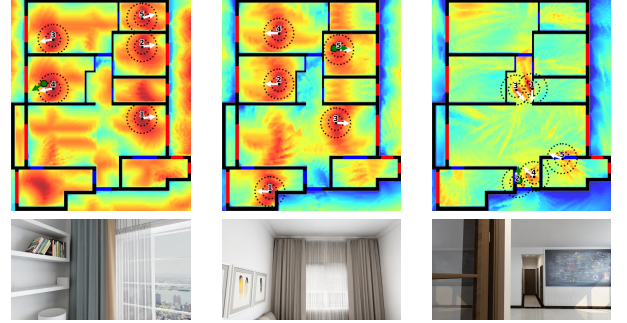


Figure 5. Illustrative examples of the impact of repetitive patterns on localization accuracy. This figure demonstrates difficult cases where, despite accurate semantic and depth predictions, floorplan localization remains challenging. The identical depth and semantic patterns may result in the top score not corresponding to the ground truth location, which motivates our analysis of Top-K recall.

In Table 3, we observe numerically that recall improves drastically as we compute recall within the Top-K candidates. This indeed indicates that our pipeline strongly captures the true location of the images within the top results, but it still remains a challenge to extract the correct location.

2.4. Ablation on Recall With Different δ_{res}

As shown in Table 4, the refinement threshold δ_{res} plays a critical role in balancing fine and coarse localization accuracy. In particular, when using a lower δ_{res} value (0.05 m), we observe a significant improvement at the fine accuracy threshold (0.1 m), achieving a recall of 18.40%. In contrast, a higher δ_{res} value (0.5 m) yields better performance at the coarser thresholds (0.5 m, 1 m, and 1 m 30°). This demonstrates the benefit of customizing the refinement process to meet specific application needs, thereby making it a flexible procedure.

Top K	0.1m	0.5m	1m	1m 30°
Top 2	7.45	57.55	70.75	69.45
Top 3	7.85	63.60	78.98	76.65
Top 5	8.82	69.18	85.69	83.18

Table 3. Recall metrics for different K values evaluated on the S3D dataset. Recall is defined as the percentage of samples for which the ground truth location is within a specified distance threshold of at least one of the Top K candidate locations extracted from the probability volume. Higher K values lead to improved recall, as more candidate locations are considered. For this experiment, we exclude the room-aware module to specifically isolate the effect of the refinement module.

δ_{res} (m)	0.1 m	0.5 m	1 m	1 m 30°
0.05	18.40	63.02	71.60	70.08
0.2	12.94	64.37	73.14	71.57
0.5	8.84	67.07	77.25	75.33
1	7.85	63.60	78.98	76.65

Table 4. Recall performance on the S3D dataset for candidate refinement using the Top 3 candidates. Recall is defined as the percentage of test instances for which at least one of the Top 3 refined candidate poses falls within the specified distance thresholds (0.1 m, 0.5 m, 1 m) and within a 30° orientation tolerance at 1 m, evaluated under different δ_{res} values.

2.5. Integrating Our Refinement into F3Loc

Table 5 quantifies the impact of our refinement module on the baseline F3Loc across both the S3D and ZInD datasets. By incorporating the refinement stage, F3Loc’s recall gains substantial improvements in every threshold (e.g., R@1 m30° on S3D rises from 21.3 to 29.6), demonstrating that our refinement module is indeed effective and substantially enhances localization performance. However, even with refinement, F3Loc+Refine still falls short of the recall achieved by our full method (both with and without room-aware predictions), which underlines that the semantics awareness of our method achieves significant gains beyond what geometric refinement alone can provide.

3. Additional Experiments and Analysis

3.1. Probability Volume Fusing Weights

In our approach, the structural-semantic probability volume is obtained by fusing the depth and semantic probability volumes:

$$P_c = w_s \cdot P_s + w_d \cdot P_d,$$

where w_d and w_s denote the weights assigned to depth and semantic cues, respectively. We determine the optimal weight configuration by evaluating recall metrics on

S3D R@				
Method	0.1m	0.5m	1m	1m 30°
F3Loc	1.5	14.6	22.4	21.3
F3Loc + Refine	2.74	23.29	30.74	29.59
Ours _s	5.42	41.87	53.52	52.61
Ours _r	5.70	45.53	58.78	57.49

ZInD R@				
Method	0.1m	0.5m	1m	1m 30°
F3Loc	0.67	7.90	15.07	11.46
F3Loc + Refine	1.21	10.46	16.94	14.21
Ours _s	2.98	24.00	33.96	29.30
Ours _r	3.31	26.60	38.01	31.86

Table 5. Recall performance on the S3D and ZInD datasets. The table reports recall at thresholds of 0.1 m, 0.5 m, 1 m, and 1 m with a 30° orientation tolerance for the baseline F3Loc with and without our refinement module.

the validation sets. Below, we report our experiments on the S3D and ZInD datasets.

As in the main paper, all experiments use a floor-plan resolution of 0.1 m and an angular granularity of 10°. Specifically, we predict 40 rays per image and interpolate these to 9 rays during the coarse stage of localization. For the Location Extraction module, we set $\delta_{\text{res}} = 0.05$ m, $\delta_{\text{ang}} = 5^\circ$, and $\Delta_{\text{max}} = 10^\circ$, and report results using Top $K = 5$ candidates.

3.1.1. Performance Breakdown on the S3D Dataset

Table 6 presents a consolidated view of recall performance for various weight configurations on the S3D validation set. Based on these results, we selected $w_d = 0.6$ and $w_s = 0.4$ as our final configuration, as it yielded the best overall performance over the validation split.

Weights		0.1 m	0.5 m	1 m	1 m 30°
w_d	w_s				
1.0	0	2.83	22.31	30.27	29.05
0.9	0.1	4.79	34.71	44.33	43.56
0.8	0.2	5.19	38.04	48.82	48.03
0.7	0.3	5.20	38.68	49.83	49.02
0.6	0.4	4.93	39.22	50.16	49.48
0.5	0.5	5.17	38.31	49.44	48.64
0.4	0.6	4.96	37.43	48.68	47.89
0.3	0.7	4.52	36.29	47.56	46.46
0.2	0.8	4.21	35.01	45.66	44.55
0.1	0.9	4.29	34.40	44.49	43.45
0	1.0	0.11	3.60	8.93	7.27

Table 6. Recall metrics on the S3D validation set obtained with our model without room aware and refinement.

3.1.2. Performance Breakdown on the ZInD Dataset

Table 7 shows the recall performance on the ZInD validation set for different weight configurations. For this dataset, the configuration $w_d = 0.4$ and $w_s = 0.6$ achieved the best overall performance.

Weights		0.1 m	0.5 m	1 m	1 m 30°
w_d	w_s				
1.0	0	0.83	8.95	14.45	11.85
0.9	0.1	1.13	13.14	20.53	18.07
0.8	0.2	1.28	15.21	23.57	20.96
0.7	0.3	1.53	16.61	25.69	22.90
0.6	0.4	1.56	16.88	26.07	23.58
0.5	0.5	1.51	16.74	26.37	23.31
0.4	0.6	1.38	16.90	26.86	23.87
0.3	0.7	1.31	16.38	26.39	23.67
0.1	0.9	1.22	16.16	25.81	22.97
0	1.0	0.04	1.83	5.25	3.04

Table 7. Recall metrics on the ZInD validation set obtained with our model without room aware and refinement.

3.2. Room Type Classification Results

In this section, we evaluate the performance of our room type prediction branch on two datasets: S3D (3.2.1) and ZInD (3.2.2). Accurate room type classification not only provides semantic context for localization but also reduces the effective search space for image matching.

3.2.1. Room Type - S3D

On the S3D dataset, which consists of fully furnished environments, our model achieves a room type prediction accuracy of 72.1%. A major source of misclassifications stems from uninformative images and rooms lacking furniture, which are common in the dataset. As shown in Figure 6, correct predictions generally exhibit high confidence scores (greater than 0.8), whereas misclassifications tend to display a more uniform confidence distribution across incorrect labels. Based on these observations, we set our threshold $T_{\text{room}} = 0.8$: any prediction with a confidence score lower than 0.8 is rejected. This strategy limits misclassifications and effectively narrows the search space, resulting in an average improvement of 6.2% across the 0.5m, 1m, and 1m 30° thresholds, as seen from the gap between Ours_s and Ours_r . Notably, on the 1m 30° metric, the improvement is 3.74 percentage points.

Figure 7 illustrates the overall room type distribution in the S3D dataset. Notably, bedrooms dominate the dataset, with an average of three per floorplan. Although this narrows the search space, it does not isolate a single room type. Furthermore, our analysis reveals that the areas corresponding to room labels account for only

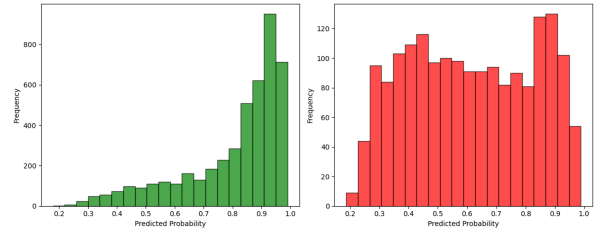


Figure 6. Room type prediction branch confidence scores over the S3D dataset. Correct predictions (green, left side) show high confidence, while incorrect predictions (red, right side) are more uniformly distributed.

27.6% of the total apartment area. This means that, on average, if true room labels were available, the effective area to be searched would be reduced to just 27.6% of the full apartment, significantly narrowing the search space for image localization.

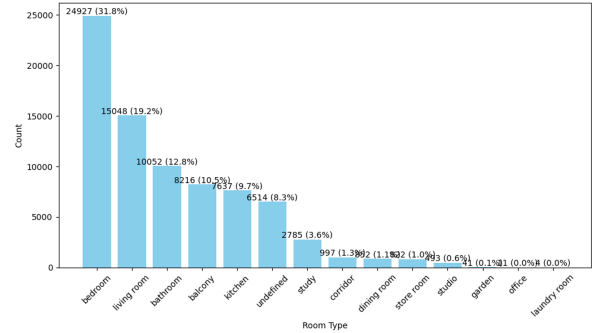


Figure 7. Overall room type distribution in the S3D dataset. Each column indicates the total number of rooms with the corresponding label and their percentage out of all rooms.

3.2.2. Room Type - ZInD

For the ZInD dataset, the prediction accuracy drops significantly to 45%. This lower accuracy can be attributed to the unfurnished nature of the dataset, which results in many ambiguous room images, and to the large number (over 250) and inconsistency of room labels. To address these issues, we grouped similar labels (e.g., “bedroom-1”, “primary bedroom”, “main bedroom”) into a single category. After grouping, we selected the top 15 room labels and classified all remaining labels as *undefined* (thereby excluding sparse categories). Although the gain from incorporating room predictions on the 1m 30° metric in ZInD is 2.11 percentage points, lower than that observed in S3D, it still constitutes a significant enhancement in narrowing the search space for image localization.

3.3. Effects of Refinement Parameter Choices

In Table 8 we present an experiment on the S3D validation set, comparing baseline methods with refinement results across various configurations. From this table, we selected the best score and used its corresponding parameters for evaluation on our test set.

Method	dist	alpha	Top-K	R@0.1m	R@0.5m	R@1m	R@1 m 30°
Baseline	0.1	0.1	3	0.055	0.417	0.545	0.533
Refine	0.1	0.1	3	0.049	0.432	0.566	0.553
Baseline	0.1	0.1	5	0.054	0.416	0.547	0.535
Refine	0.1	0.1	5	0.047	0.426	0.564	0.553
Baseline	0.1	0.3	3	0.054	0.420	0.548	0.537
Refine	0.1	0.3	3	0.049	0.437	0.570	0.557
Baseline	0.1	0.3	5	0.050	0.409	0.539	0.528
Refine	0.1	0.3	5	0.052	0.428	0.563	0.551
Baseline	0.1	0.5	3	0.050	0.413	0.539	0.528
Refine	0.1	0.5	3	0.051	0.430	0.563	0.551
Baseline	0.1	0.5	5	0.053	0.417	0.544	0.532
Refine	0.1	0.5	5	0.052	0.435	0.564	0.553
Baseline	0.5	0.1	3	0.054	0.413	0.541	0.530
Refine	0.5	0.1	3	0.046	0.377	0.542	0.528
Baseline	0.5	0.1	5	0.055	0.413	0.538	0.526
Refine	0.5	0.1	5	0.042	0.350	0.527	0.513
Baseline	0.5	0.3	3	0.054	0.420	0.547	0.536
Refine	0.5	0.3	3	0.053	0.392	0.552	0.539
Baseline	0.5	0.3	5	0.055	0.417	0.543	0.531
Refine	0.5	0.3	5	0.046	0.372	0.535	0.522
Baseline	0.5	0.5	3	0.053	0.414	0.546	0.533
Refine	0.5	0.5	3	0.050	0.388	0.546	0.533
Baseline	0.5	0.5	5	0.051	0.412	0.540	0.529
Refine	0.5	0.5	5	0.048	0.370	0.536	0.523
Baseline	1.0	0.1	3	0.054	0.420	0.552	0.540
Refine	1.0	0.1	3	0.050	0.358	0.496	0.484
Baseline	1.0	0.1	5	0.052	0.413	0.541	0.530
Refine	1.0	0.1	5	0.042	0.321	0.455	0.444
Baseline	1.0	0.3	3	0.054	0.414	0.542	0.531
Refine	1.0	0.3	3	0.053	0.382	0.516	0.504
Baseline	1.0	0.3	5	0.052	0.412	0.543	0.531
Refine	1.0	0.3	5	0.053	0.369	0.504	0.492

Table 8. Refinement parameter experiment on the S3D validation set.

3.4. Additional Comparison with LASER

Table 9 compares our approach with the LASER baseline. To ensure a fair evaluation, we train our model under the same protocol as LASER, applying random yaw perturbations to the panoramas during training. We then evaluate both methods on the test set—using the same random yaw sampling—and report the mean recall over five independent runs. Our method significantly outperforms LASER at the 1 m and 1 m 30° thresholds; in particular, we achieve a 64% absolute improvement on the 1 m 30° metric, which is the most critical measure for our application. LASER, however, attains higher recall on the fine localization metrics (0.1 m), suggesting that given a large training set, their model can achieve finer-grained accuracy. We observe that the scores for the dataset when randomly cropping panoramas are lower than those for the perspective sets. Two factors contribute to this gap: (i) under random-yaw training, a larger fraction of panorama crops contain uninformative wall-only views, making localization harder. And (ii)

in the S3D dataset the resolution of a cropped panorama view is much lower than that of an image covering the same field of view in the perspective dataset—*e.g.*, approximately 228×512 px versus 1280×720 px. Both the reduced visual content and the lower image quality adversely affect model performance on the panorama random yaw crop. With these results, we consider the LASER baseline to be faithfully reproduced.

Method	0.1 m	0.5 m	1 m	1 m 30°
LASER	6.48	25.75	31.05	22.57
Ours _s	3.12	23.84	32.34	29.52
Ours _r	4.33	31.12	42.49	37.13

Table 9. Recall metrics on the S3D dataset, with a random yaw in the training stage. Results are reported on the random angle of yaw of each panorama in the test set and averaged over $N = 5$ times.

3.5. Comparison against Soft Constraints

Our approach uses hard thresholds both for semantic ray classification—where we assign each ray the class with the highest probability—and for room-type selection—where we apply a binary mask for the room with the maximum confidence. To validate this hard-threshold strategy against a soft-constraint alternative, we conduct two experiments: (1) **Semantic Ray Classification** compares hard vs. soft ray assignments, and (2) **Room-Type Selection** compares hard vs. soft room-type classifications. Results are reported in Table 10 and Table 11

3.5.1. Semantic Ray Classification

For semantic ray classification, instead of selecting the highest-probability class for each ray (hard assignment), we retained the logits and computed a probability map by measuring cross-entropy with the ground truth (soft assignment). This soft approach caused a dramatic decrease in all recall metrics (*e.g.*, $R@1\text{ m }30^\circ$ dropped to 25.88% on S3D), demonstrating that hard assignments are crucial for aggregating semantic information in our network.

Method	0.1 m	0.5 m	1 m	1 m 30°
Hard Ours _s	5.42	41.87	53.52	52.61
Hard Ours _r	5.70	45.53	58.78	57.49
Soft Ours _s	1.94	16.74	26.22	22.66
Soft Ours _r	2.24	19.55	31.43	25.88

Table 10. Recall metrics on the S3D dataset for semantic ray classification under hard vs. soft assignments (Experiment 1). The top result in each column is **bolded**.

3.5.2. Room-Type Selection

For room-type selection, we compared a hard classification approach—where each room polygon receives a binary mask from the maximum-probability prediction—to a soft classification approach, in which each room polygon is weighted by its predicted probability. Although the gap is modest, hard classification still outperforms the soft approach (e.g., 1.23% gap in $R@1\text{ m }30^\circ$ on S3D).

Method	0.1 m	0.5 m	1 m	1 m 30°
Hard Ours _s	5.70	45.53	58.78	57.49
Soft Ours _r	4.55	43.57	58.51	56.27

Table 11. Recall metrics on the S3D dataset for room-type selection under hard vs. soft classification (Experiment 2). The top result in each column is **bolded**.

These limitations suggest that improvements in semantic segmentation and more sophisticated feature disambiguation techniques could enhance performance. We believe that addressing these issues can lead to further improvements in localization accuracy in future work.

4. Additional Visualizations

4.1. Qualitative Visualizations

In this section, we show more visual examples from our predictions on both datasets. Figure 8 presents several successful examples from the S3D dataset, illustrating how combining precise semantic information with structural data can yield accurate localizations. We added an interpolated line, colored by each ray’s semantic label, connecting the ray endpoints to make interpolation easier. Figure 9 further demonstrates our predictions on the ZInD dataset. In both figures, warmer colors correspond to higher probabilities, with magenta indicating the ground-truth location and white denoting our predicted layout. The strong similarity between the ground-truth rays and the predicted rays underlines the effectiveness of our method.

4.2. Visualization of Baseline Comparisons

Here, we present additional examples comparing our method against baseline approaches, specifically F3Loc and LASER, on the ZInD dataset. More visual examples of these comparisons are shown in Figure 10.

5. Limitations

Figure 11 illustrates several failure cases from both of the datasets where our approach struggles. In these examples, misclassifications of certain semantic labels or confusions between visually similar features, such as interpreting a window as a door (row 1) or mistaking the window size (row 3), can lead to localization errors. The figure displays both the ground truth rays and the predicted rays, highlighting the differences and emphasizing the critical role of precise semantic inference for robust indoor localization.


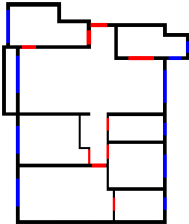
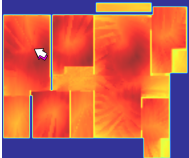
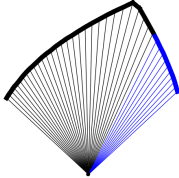
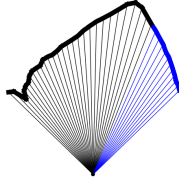

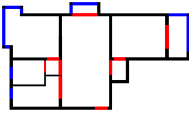
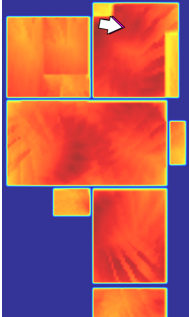
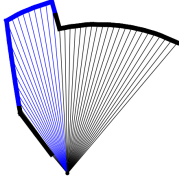
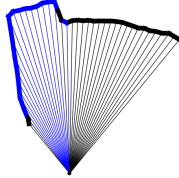
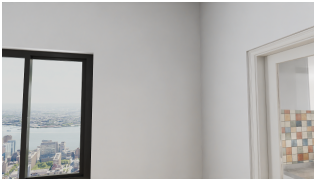
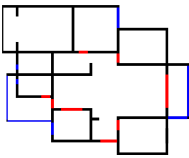
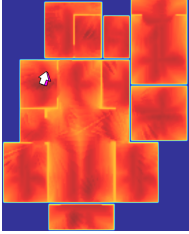
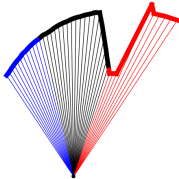
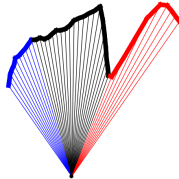

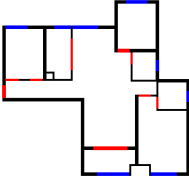
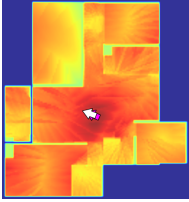
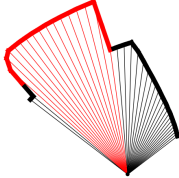
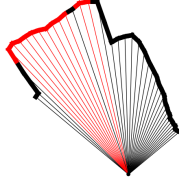
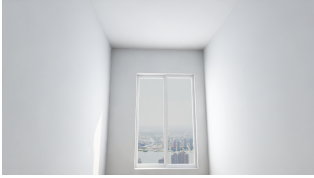
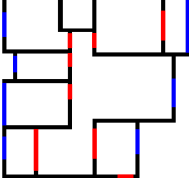
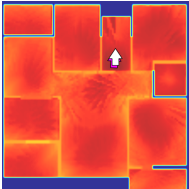
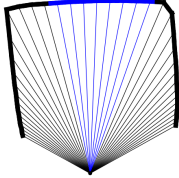
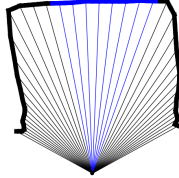

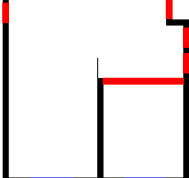
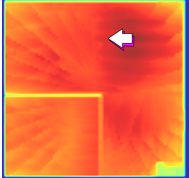
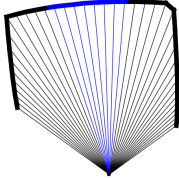
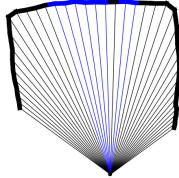
Input Image	Floorplan	Ours	Ground Truth Rays	Predicted Rays
				
				
				
				
				
				

Figure 8. **Additional Qualitative Results (S3D dataset):** Warmer colors correspond to higher probabilities, while magenta indicates the ground-truth location and white denotes our predicted layout. Rays are: wall, window, and door.

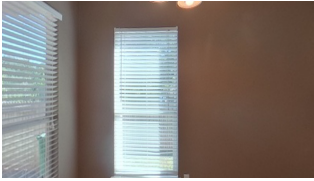
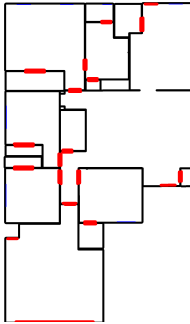
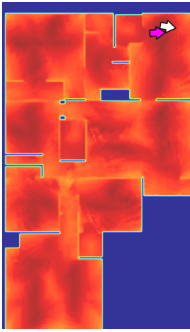
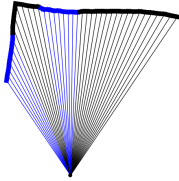
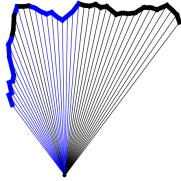
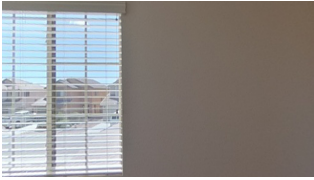
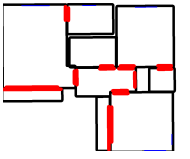
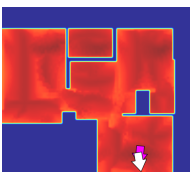
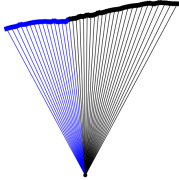
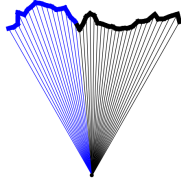
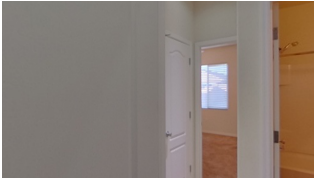
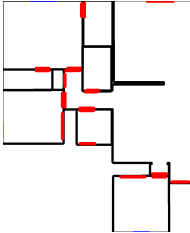
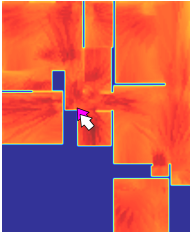
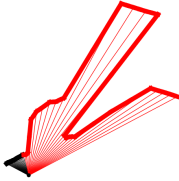

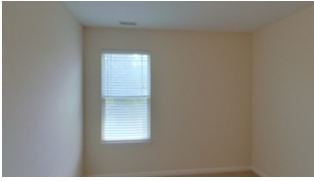
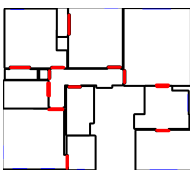
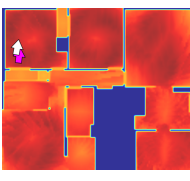
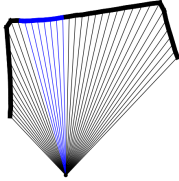
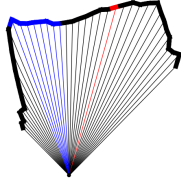
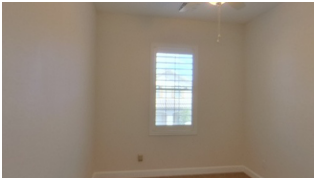
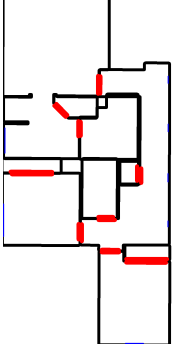
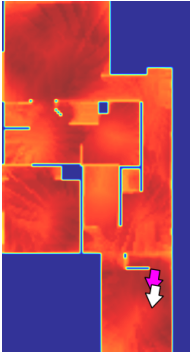
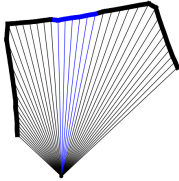
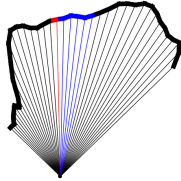

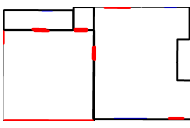
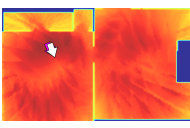
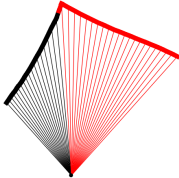
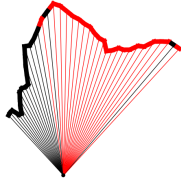
Input Image	Floorplan	Ours	Ground Truth Rays	Predicted Rays
				
				
				
				
				
				

Figure 9. **Additional Qualitative Results (ZInD dataset):** Warmer colors correspond to higher probabilities, while magenta indicates the ground-truth location and white denotes our predicted layout. Rays are: wall, window, and door.

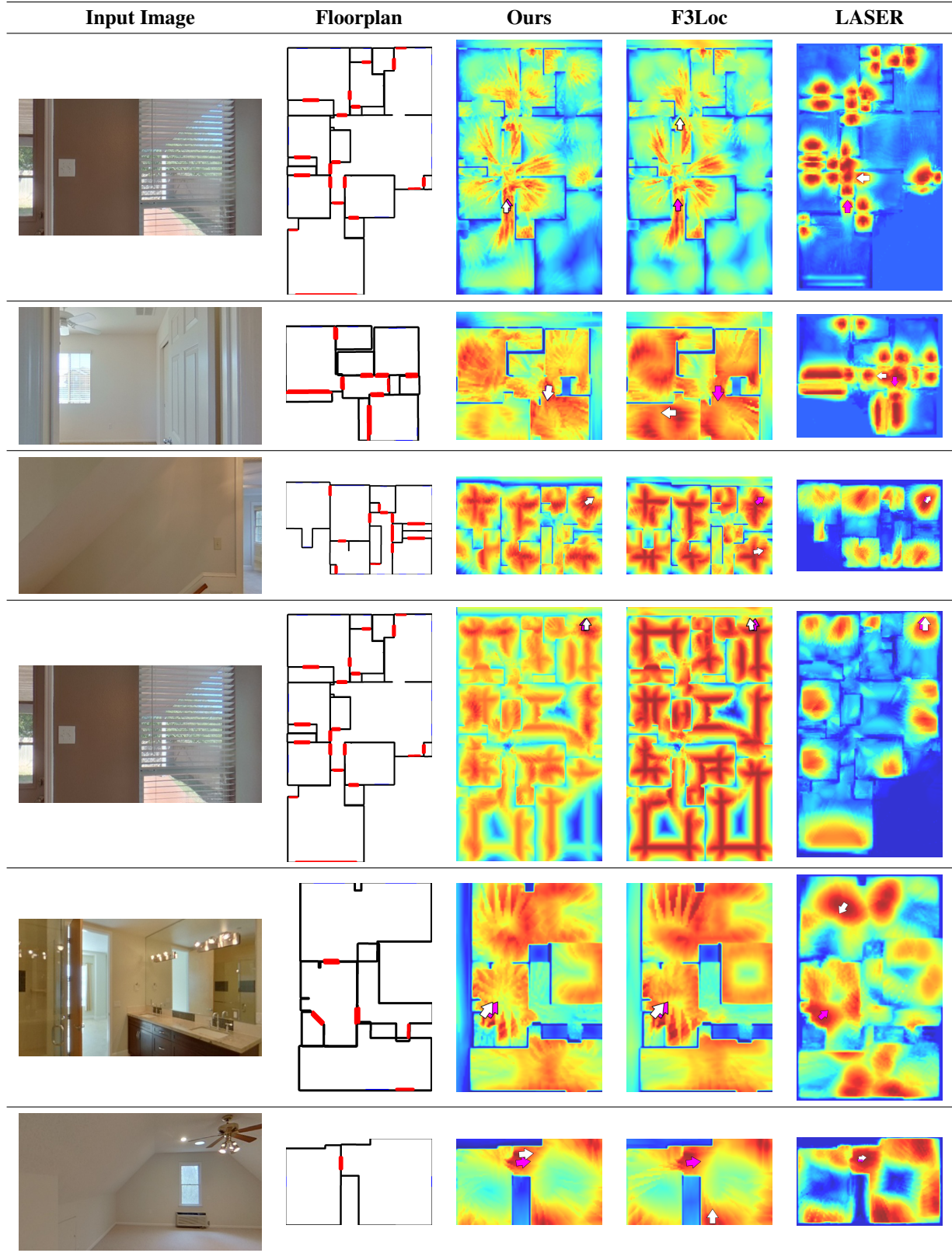


Figure 10. **Comparison to Baseline Methods:** Additional visualizations comparing our method with F3Loc and LASER on the ZiND dataset. Warmer colors correspond to regions with higher predicted probabilities. Overlaid on the estimated probabilities, we indicate the ground truth location (magenta) and the predicted location. Rays are: wall, window, and door.


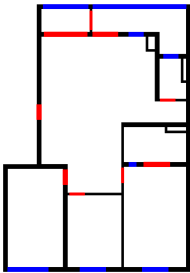
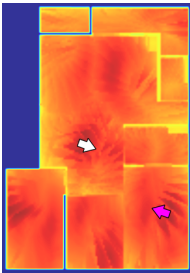
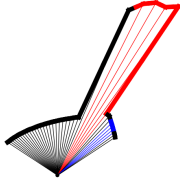
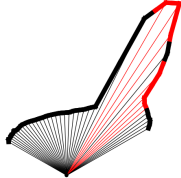

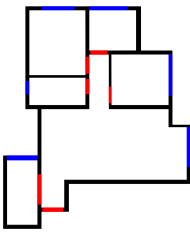
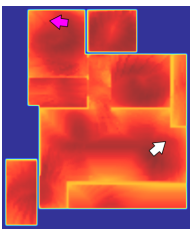
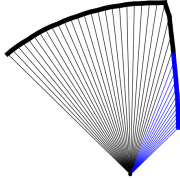
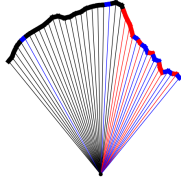
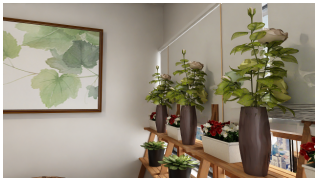
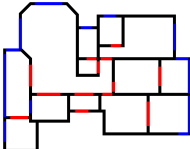
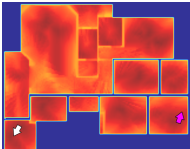
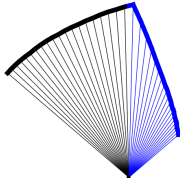
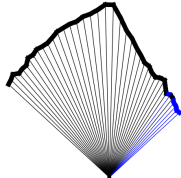

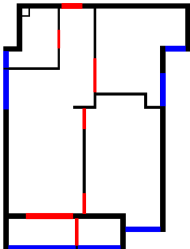
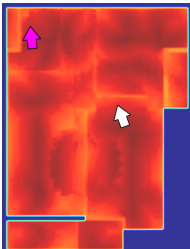
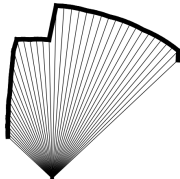
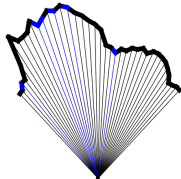
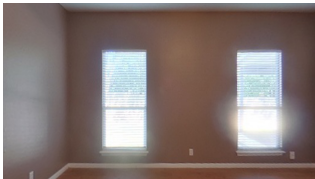
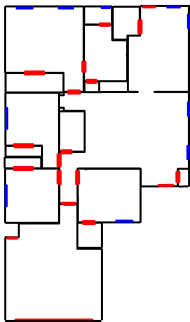
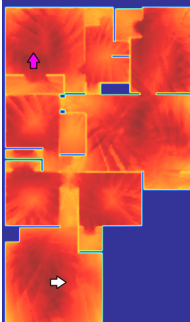
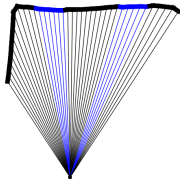
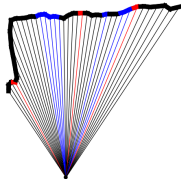
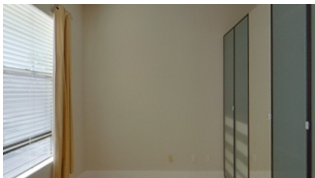
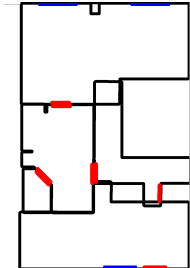
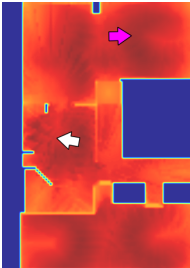
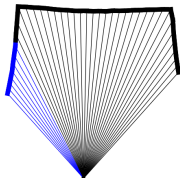
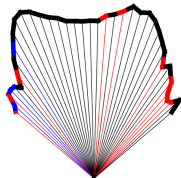
Input Image	Floorplan	Ours	Ground Truth Rays	Predicted Rays
				
				
				
				
				
				

Figure 11. **Limitations.** Above we show several failure cases, where semantic misclassifications and structural ambiguities lead to localization errors; see Section 5 for additional details. Warmer colors again represent higher probabilities. **Magenta** marks the ground truth, and white indicates the estimated layout. Rays are: wall, **blue**, and **red**.