F^{3}Loc: Fusion and Filtering for Floorplan Localization Supplementary Material

Changan Chen[1,*] Rui Wang[2] Christoph Vogel[2] Marc Pollefeys[1,2]

[1]ETH Zürich [2]Microsoft Mixed Reality & AI Lab Zürich

[*]Work done during his internship at Microsoft Mixed Reality & AI Lab Zürich

1. Network Details

1.1. Monocular Network

The structure of our monocular network is presented in Fig. 1. The monocular network uses a ResNet50 to extract local features and another convolutional layer to reduce the channel size. The resulting features serve as keys and values, while an average pooling is applied vertically to form queries. For each query, the attention is applied to the entire image. For the queries, we use their 1D coordinate to form a positional encoding, whereas for the keys and values the positional encoding is mapped from the corresponding 2D image coordinate. Unobservable pixels are masked out in the attention. Like the multi-view network, the monocular network outputs a probability distribution of floorplan depth, evaluated at a set of predefined depth hypotheses. In the monocular case the distribution is provided directly by the attention.

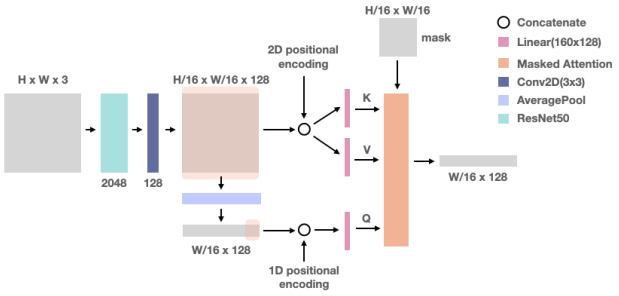

Figure 1. **Monocular Network.** The output channel sizes are denoted under the respective blocks.

1.2. Multiview Network

Figure 2 provides the detailed architecture of our multiview network. The feature extractor of the multiview network has a structure similar to the monocular network. Divergently, we use the first two blocks of ResNet50 and apply the attention only to the respective image column. The positional encoding for the keys and values is mapped from the corresponding vertical coordinate. The attention outputs the image column features using 64-channels per feature. Afterwards, as described in the main paper, the features are gathered before forming a cost volume. The cost volume is subsequently filtered by a U-Net-like network, whose details are provided in Fig. 3.

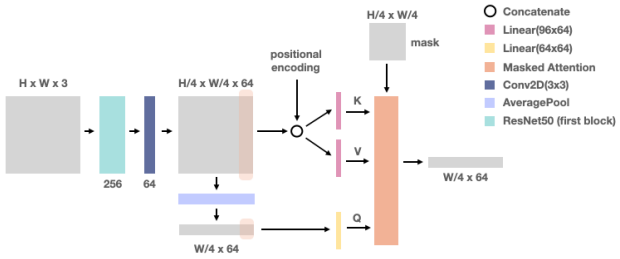

Figure 2. **Multi-view feature extractor.** The output channel sizes are denoted under the respective blocks.

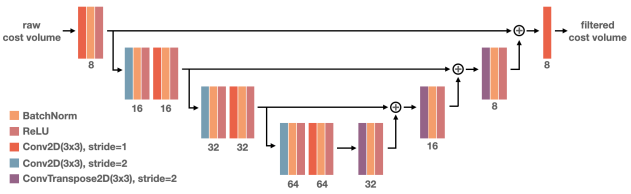

Figure 3. **Cost Filter.** The output channel sizes are provided under the respective blocks.

1.3. Selection Network

The selection employs an MLP with two hidden layers. It takes the relative poses (x, y and heading) between the frames of interest (we use 4 frames so 3 relative poses, 9 values in total) and the mean floorplan depth predictions of the two observation modules as input. It outputs a pair of weights for the monocular and the multi-view probability volume. Ablation on the input of the selection network is

studied in Sec. 7.

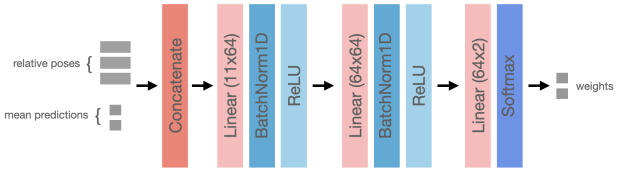

Figure 4. **Selection Network.**

2. Training Details

For the Gibson dataset, we train the monocular and multi-view network on the entire training split of Gibson(f) for 100 and 20 epochs. For the selection network, we train another pair of monocular and multiview networks on 80 scenes of the training split before freezing their weights to train the selection network on the remaining 20 scenes of Gibson(g) for 5 epochs. We train the selection network on disjoint scenes to prevent it from being biased by both modules' performance on the visited scenes. For Structured3D, we train the monocular network for 100 epochs. We use Adam [2] with a learning rate of 1×10^{-3} for all training.

3. Implementaion Details

Multi-view network The mapping from source frame to reference frame corresponds to the following transformation

$$\lambda \begin{bmatrix} u^{\text{src}} - u_0^{\text{src}} \\ \alpha_u^{\text{src}} \end{bmatrix} = \mathbf{R}_{\text{sr}} \begin{bmatrix} \frac{u^{\text{ref}} - u_0^{\text{ref}}}{\alpha_u^{\text{ref}}} \cdot d \\ d \end{bmatrix} + \mathbf{p}_{\text{sr}}, \qquad (1)$$

where superscripts src and ref indicate the source and reference frame. Limited to the horizontal image direction, u denotes the pixel coordinate, u_0^* the principal point and α_u^* the focal distance. We let d denote the hypothesized depth of the column feature in the reference frame and $\mathbf{R}_{\text{sr}} \in \mathbb{R}^{2\times2}$ and $\mathbf{p}_{\text{sr}} \in \mathbb{R}^{2\times1}$ the rotation and translation from the reference to the source frame. The factor λ indicates a necessary transformation from homogenous to image coordinates.

Particle filter For the evaluation of the particle filter we use 5000 particles. A large number of particles is needed to globally localize the camera pose. For each particle, we render its feature (floorplan depth) by computing its distance to the occupancy grids of the floorplan. This operation requires minimal computation, however, increases when the particle number increases. It is noteworthy that even with 5000 particles, the particle filter does not reach the success rates of our histogram filter, while, nevertheless, being slower than our histogram filtering framework.

4. Dataset Collection

We manually labeled the floorplans and the traversable region (crucial for observation collections) by careful inspection of the provided mesh of the Gibson environments. For Gibson(f) and Gibson(g), we first apply grid sampling with Gaussian noise in the traversable regions to get diverse viewpoints. In order to create a short multi-view sequence, we sample a local goal position within a small neighborhood and apply a simple following control to move the camera toward to local goal. For Gibson(f), the local goal position is sampled within a certain front field of view, while for Gibson(g) it can be everywhere within a prescribed sample radius. For Gibson(t), we sequentially sample ten global goal positions and plan the global path to each goal with a modified version of the Gibson built-in LazyPRM [1]. Then the camera follows the global path to reach the target position. Once the current target is reached, the next goal is sampled based on the coverage of the existing path. This results in a long trajectory that passes through almost every traversable region. The datasets are available through our project page https://felix-ch.github.io/f3loc-page/.

We sample the camera height from a Gaussian distribution centered at 1.7m with 0.02m standard deviation to simulate a human holding phone or wearing a headset. The datasets contain only up-right camera poses with minimal roll and pitch disturbances (uniformly in -0.005rad to 0.005rad). Diverse roll and pitch angles are augmented during the training. Details about our Virtual roll-pitch augmentation are already provided in the main paper. For the virtual roll pitch ablation study, we randomly sample the roll and pitch angle uniformly within a prescribed range, for instance -0.1rad to 0.1rad.

In Strutured3D, the camera height is centered at 1.5m and more diverse. The distribution is shown in Fig. 5. The distribution of the roll and pitch angle is shown in 6 and Fig. 7.

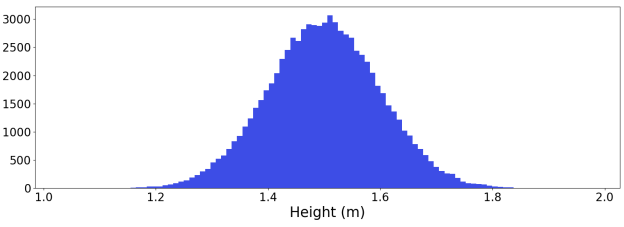

Figure 5. **Distribution of camera height in Structured3D.**

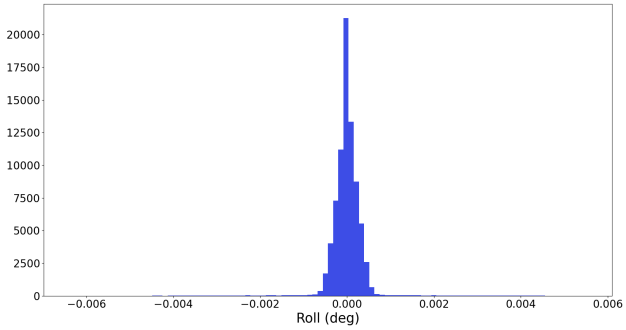

Figure 6. **Distribution of camera roll angle in Structured3D.**

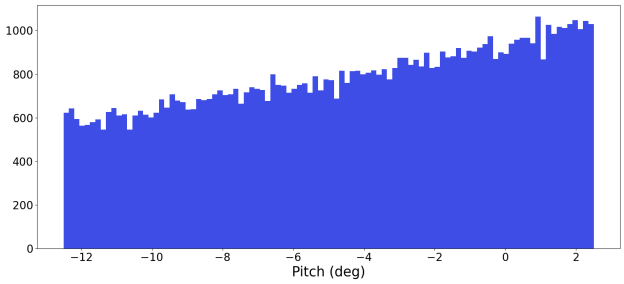

Figure 7. **Distribution of camera pitch angle in Structured3D.**

5. Additional Qualitative Studies

5.1. Floorplan Depth Estimation

Examples of the monocular floorplan depth estimation are shown in Fig. 8 Fig. 9 and Fig. 10. Our network can estimate accurate floorplan depth and ray-scans (see Fig. 8). The network does not only perform well, if the room structure is clear, but is able to handle fully furnished rooms (see Fig. 9). Typical failure cases are very cluttered scenes due to a high furnishing level and so caused occlusions, as shown in Fig. 10. Note, however, that the structure of the ray-scans retains a reasonable similarity to the ground truth rays. Overall, the network performs remarkably robust in furnished rooms.

5.2. Attention

We employ an attention mechanism to help the network estimate accurate floorplan depth. We notice that the monocular network focuses on the room structures and vanishing lines to predict the floorplan depth. This is illustrated in Fig. 11.

5.3. Observation Likelihood

Figure 12 provides an extended qualitative comparison of the predicted observation likelihood between ours and the baselines on both the Gibson dataset and the Structured3D dataset. As can be observed, our model delivers more accurate pose predictions and is more consistent in terms of handling multiple hypotheses.

	Panoramic Structured 3D			
R@	0.1m	0.5m	1m	1m30°
LASER	1.1	14.1	**25.0**	14.7
Ours$_s$	**2.8**	**15.3**	19.4	**15.8**

Table 1. **Single frame localization on perspective images cropped from Panoramic Structured3D.**

5.4. Posterior Evolution

Six example trials of the posterior evolution are shown in Fig. 13 and Fig. 14. In each trial, we see a clear multimodality at the beginning, due to insufficient and ambiguous observations. The posterior estimation converges to a sharp peak as the observer moves around and more observations are accumulated.

6. Additional Quantitative Results

6.1. Panoramic Structured 3D

Since our targeted application is localization with mobile devices that capture perspective images. The perspective Structured3D is the most suitable and straightforward option. Nevertheless, we present an additional experiment for LASER and our method on panoramic Structured3D with 90°FoV (see Tab. 1). Although we see a noticeable gap on the 1m recall, we outperform LASER on all other three metrics. We emphasize that R@1m30° is a more important metric for perspective camera localization, since this better represents how good the camera frustums match. Our recall@0.1m is higher than LASER, showing its high accuracy. It is worthmentioning, to make the results more statistically meaningful, we averaged the results of 10 runs, since we noticed a notable difference on recall (e.g. 2-3% for 1m) across different trials of testing, as the LASER evaluation script randomly crops one perspective image from each panorama. Hence, we consider the LASER baseline as properly reproduced.

6.2. LaMAR HGE

Since the data is limited, we only obtained 6 trajectories with 4 successfully localized (last 10 frames within 1m of the ground truth), resulting in a success rate of 4/6 at 1m. The RMSE of the succeeded tracking is 0.26m (last 10 states).

7. Ablation

Shape loss We adopted a cosine similarity based shape loss in the monocular training (Eq. 11 in the main paper). In

our experience this allows the network to learn more accurate floorplan depth (see Tab. 2). The loss performs best, if both shape loss and the L1 loss are set roughly to the same magnitude. Throughout all our training, we use a shape loss weight λ of 20.

	Gibson(f)			
R@	0.1m	0.5m	1m	1m30°
$\lambda = 0$	4.9	27.5	34.2	32.8
$\lambda = 1$	**5.3**	27.3	34.1	32.6
$\lambda = 20$	4.7	**28.6**	**36.6**	**35.1**
$\lambda = 100$	3.2	23.9	33.6	31.6

Table 2. **Shape loss for monocular network training.** Using the proposed shape loss for training the monocular network improves its recall.

Depth hypotheses are important for the multi-view network. For instance, in the literature, hypothesis are sometimes sampled equidistantly in depth and sometimes in disparity space, which can make a big difference. Accordingly ,the depth hypotheses sampling can be generalized as sampling equidistantly in the space of d^α between d^α_{min} and d^α_{max}, where d is the depth. $\alpha = 1$ corresponds to ordinary equidistant samples, where as $\alpha = -1$ corresponds to sampling equidistantly in the disparity space. We set $d_{min} = 0.1m, d_{max} = 15m$, a common floorplan depth range for residential buildings. Using $\alpha = 1$ leads to a coarse sampling for short distances whereas using $\alpha = -1$ provides a rougher sampling of hypotheses between 4m to 8m, which are typical floorplan depth values for residential buildings. We found that setting $\alpha = -0.2$ offers a good compromise. This value delivers fine sampling in near distances and provides quite dense hypotheses near typical floorplan depth values. An ablation study on α is in Tab. 3.

	Gibson(f)			
R@	0.1m	0.5m	1m	1m30°
$\alpha = 1$	12.1	40.5	**45.5**	**43.9**
$\alpha = -0.2$	**13.2**	**40.9**	45.2	43.7
$\alpha = -1$	12.0	40.2	44.8	43.4

Table 3. **Depth hypotheses sampling in the multi-view network.** A compromise between equidistant sampling in depth and in disparity space delivers the best recall. More details are given in the text.

Depth prediction for selection We test a variant of our selection network, where we do not provide the mean depth predictions as additional input besides the relative poses. The performance based on either input is compared in Tab. 4, where it is shown that the additional input allows the network to make better selections. This could be because, for instance, the monocular prediction depends on how far the camera is from the wall, if it is too close to the wall, the prediction fails whereas multi-view might still work.

	Gibson(g)			
R@	0.1m	0.5m	1m	1m30°
with	**12.2**	**39.4**	**44.5**	**43.2**
without	11.5	38.8	43.8	42.5

Table 4. **Mean depth prediction as additional input for the selection network.** Using the mean depth as input for the selection network improves recall performance.

Grid resolution Recall that the feature (virtual ray scan) extracted from the image is compared to the features on the grid points of the floorplan. The quality of the predictions and the computation cost depend on the sampling of the grids. Finer grid sampling can lead to higher accuracy, but also to a higher computation cost. An ablation on the grid resolution is shown in Tab. 5. The grid resolution affects matching time. However, for usual floorplan sizes of residential buildings, feature extraction is the dominant factor in timing. Hence, the total iteration time does not change significantly in our evaluation.

References

[1] Robert Bohlin and Lydia E. Kavraki. Path planning using lazy prm. pages 521–528, 2000. 2

[2] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 2

	Success Rate @ (%)				Timing(s)		
Grid Resolution	0.2m	0.3m	0.5m	1m	Feature Extraction	Matching	Iteration (HF)
0.1m × 0.1m	**62.2**	**89.2**	**94.6**	**94.6**	0.033	0.003	0.037
0.2m × 0.2m	10.8	56.8	83.8	**94.6**	0.032	**0.001**	**0.033**

Table 5. **Success rate and timing for different grid resolutions.** A finer grid sampling leads to more accurate pose estimation. For typical floorplan sizes feature extraction is the dominant processing step and the timing remains largely unaffected of grid resolution.

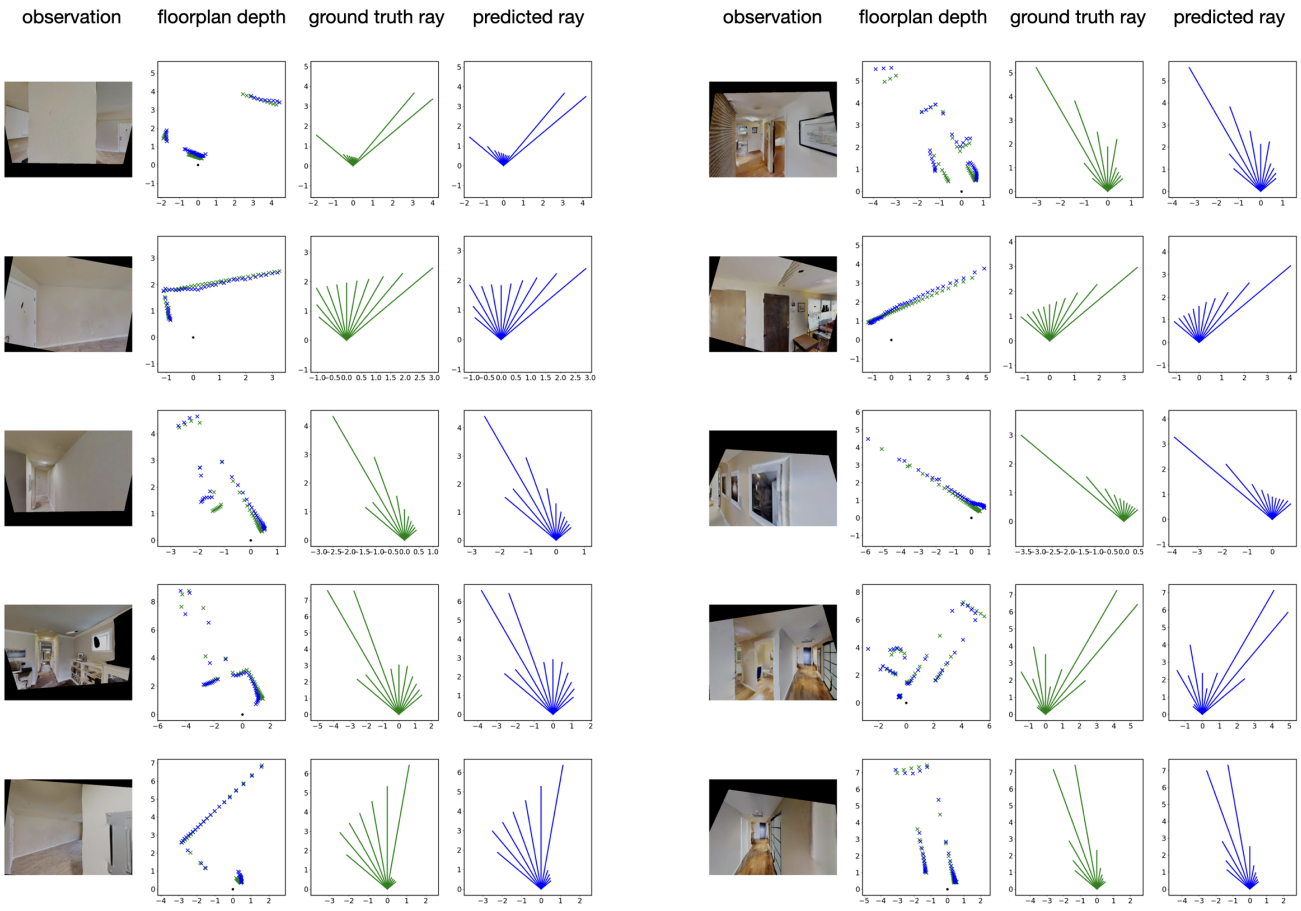

Figure 8. **Accurate floorplan depth prediction for clear room structure.** The ground truth floorplan depth is marked as × and predicted ones ×.

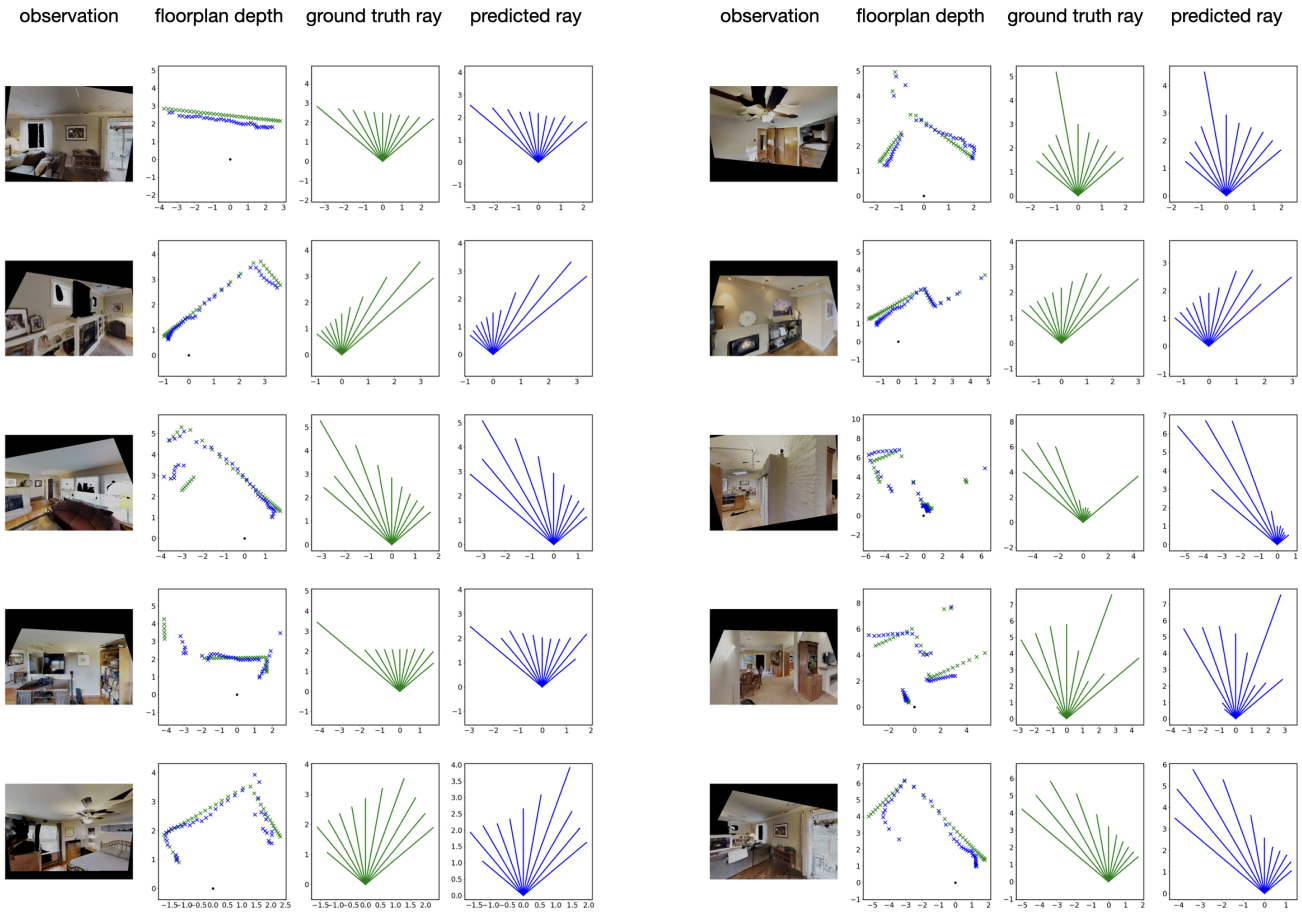

Figure 9. **Floorplan depth prediction handles furnishing and occlusion to a certain extent.** The ground truth floorplan depth is marked as $\times$ and predicted ones $\times$.

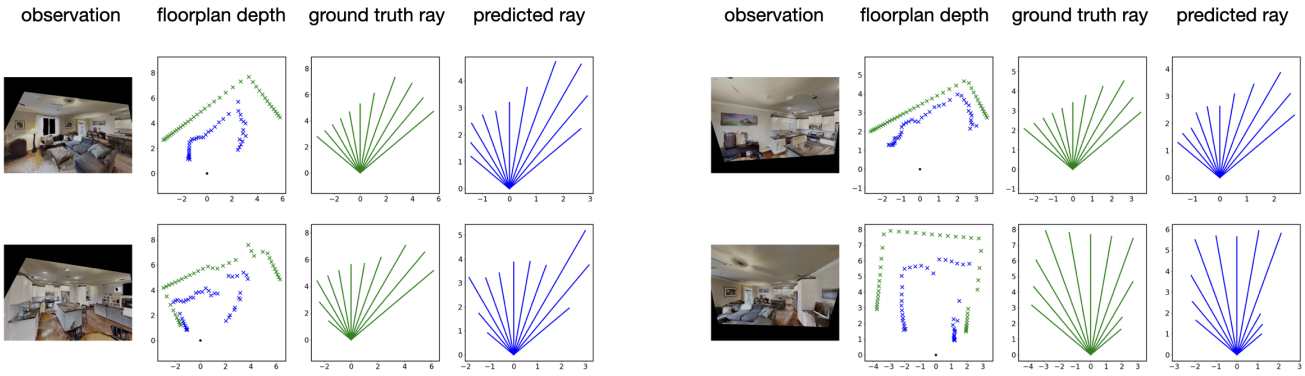

Figure 10. **Floorplan depth prediction fails due to high furnishing level and occlusion.** The ground truth floorplan depth is marked as $\times$ and predicted ones $\times$.

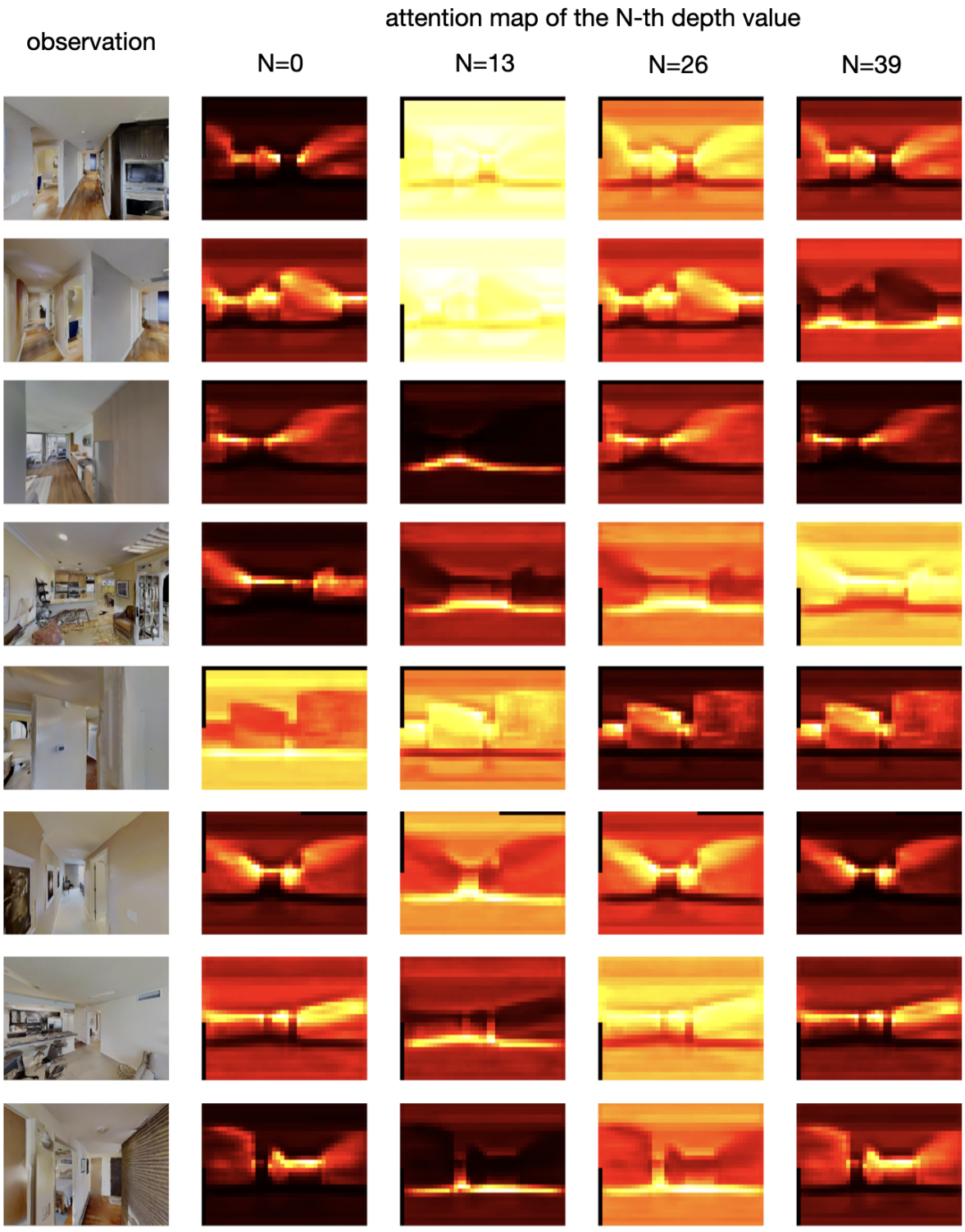

Figure 11. **Attention on the room structure.** The attention maps from left to right are the attention map for the 0th, 13th, 26th, and 39th of the 40 depth values from left to right.

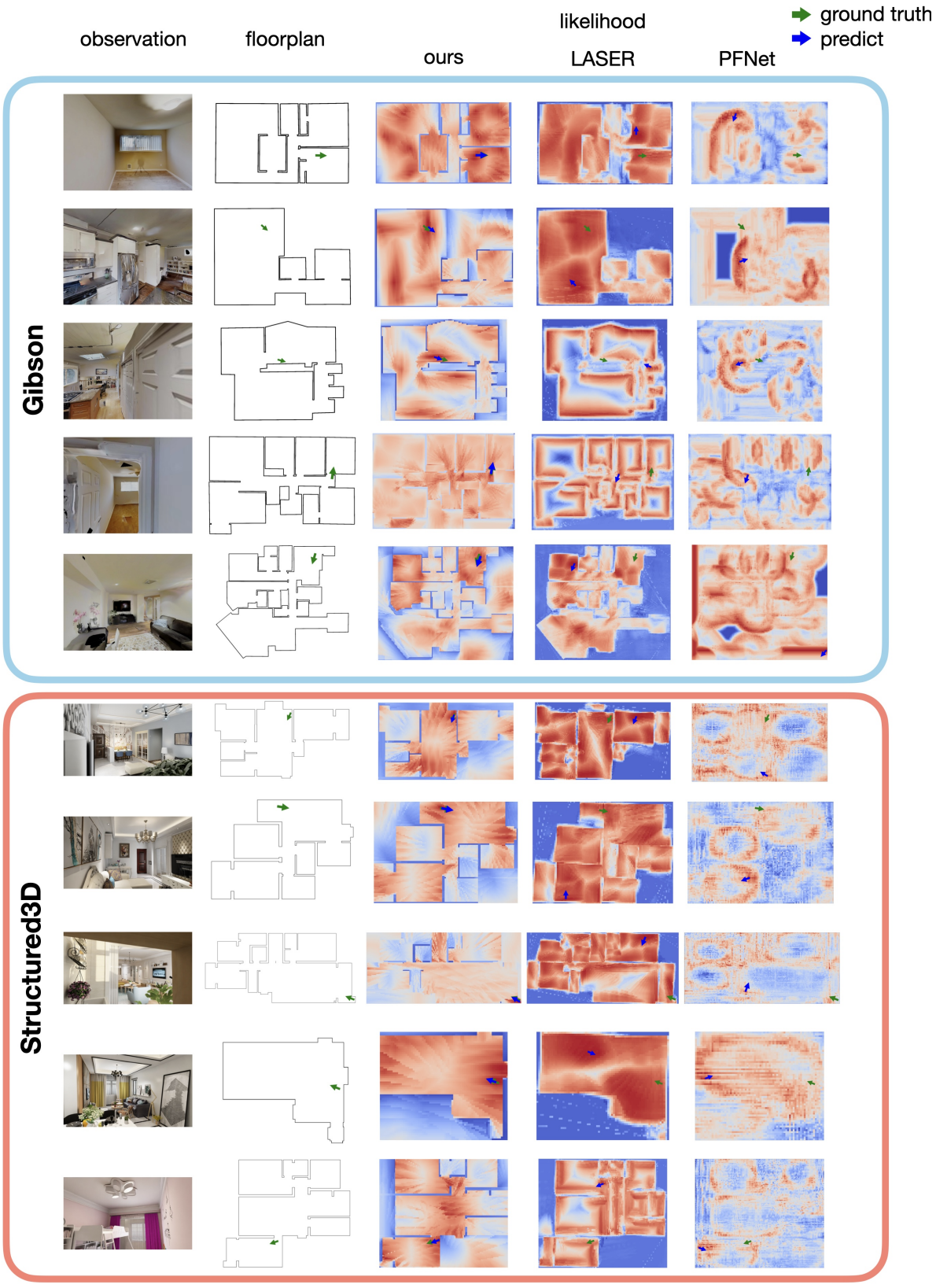

Figure 12. **Observation likelihood.**

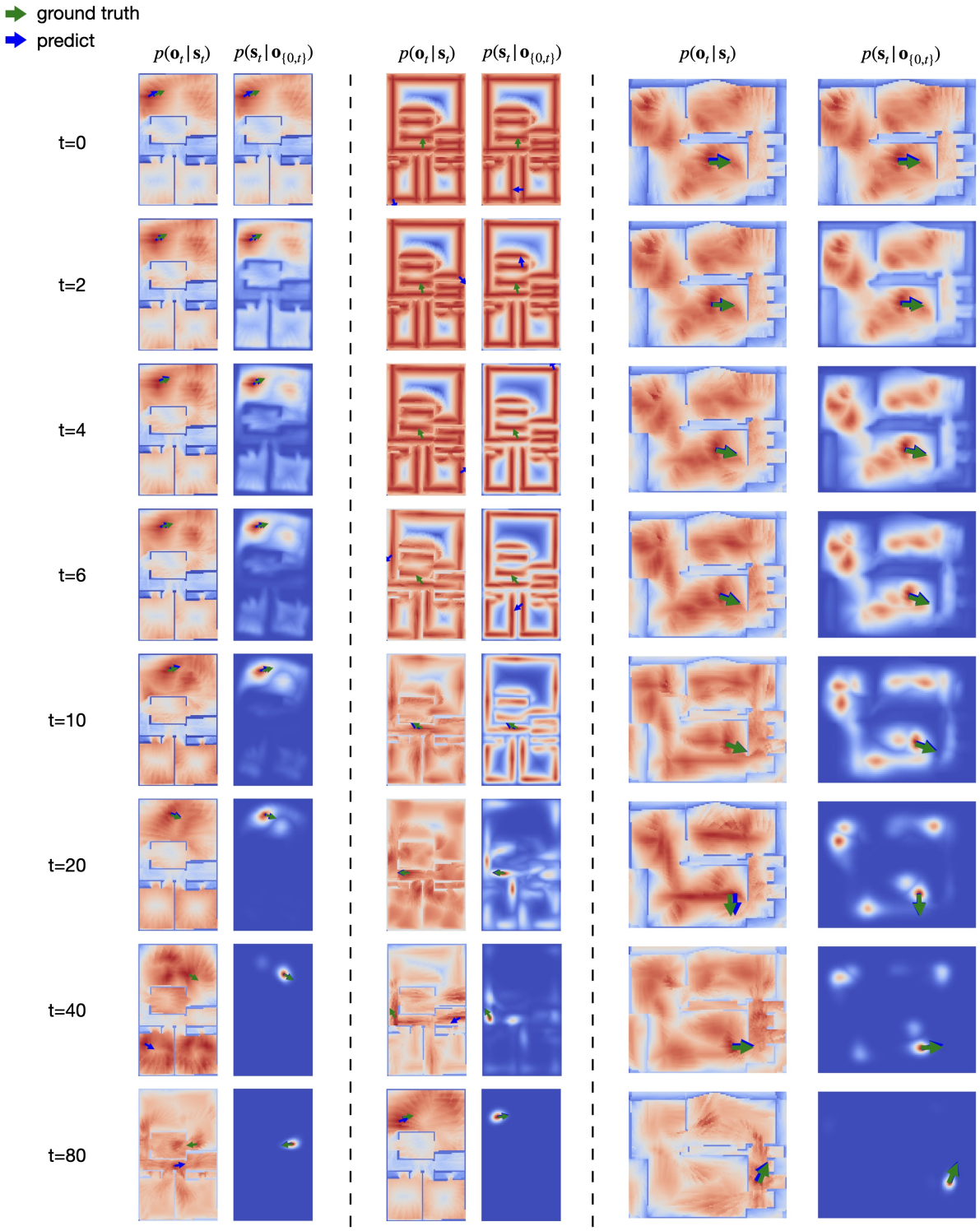

Figure 13. **Posterior evolution.**

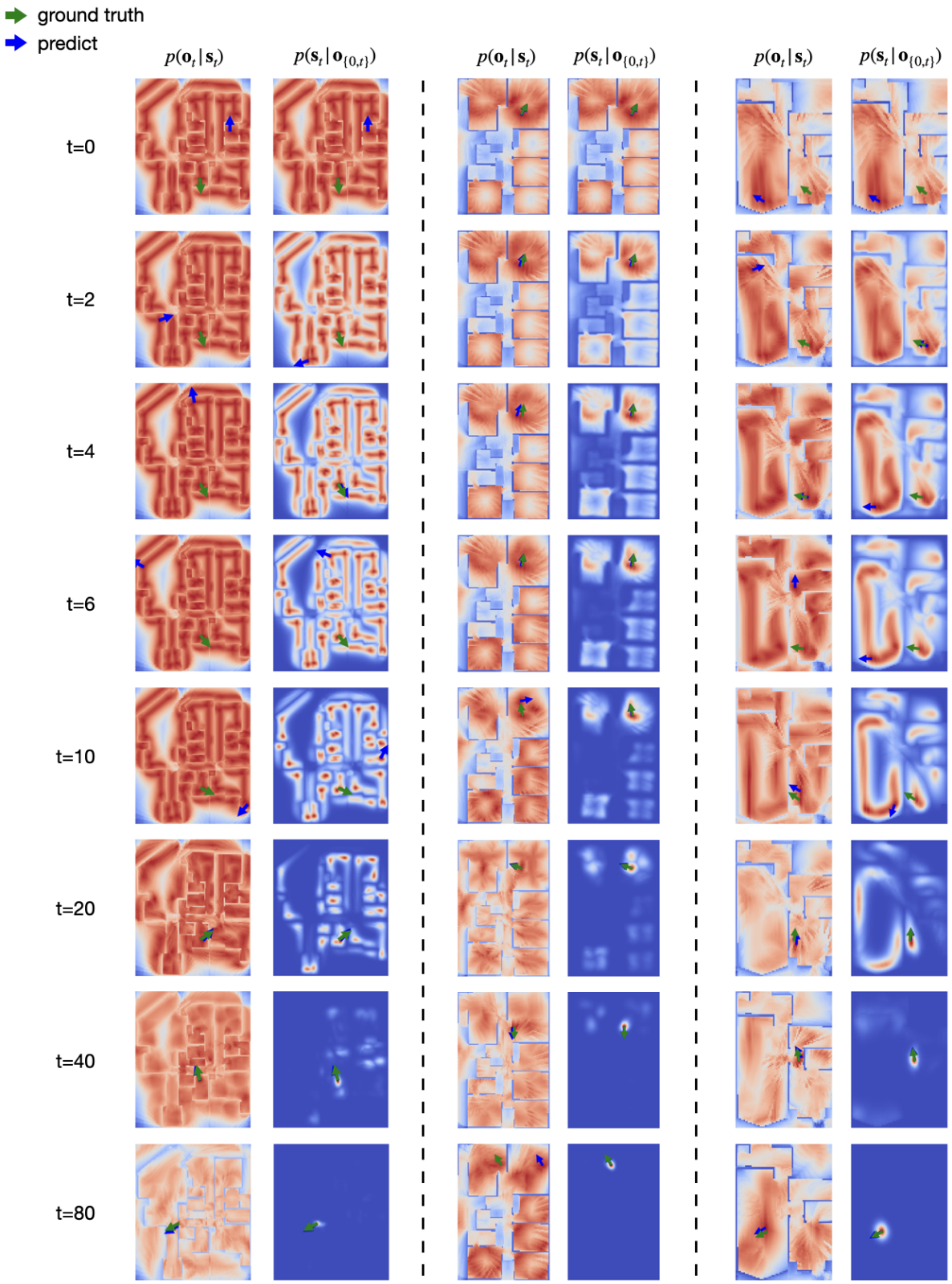

Figure 14. **Posterior evolution.**