Supplementary Material for
Scene Memory Transformer for Embodied Agents in Long-Horizon Tasks

Kuan Fang[1] Alexander Toshev[2] Li Fei-Fei[1] Silvio Savarese[1]
[1]Stanford University [2]Google Brain

A. Environment Details

In all experiments, we simulate a mobile base of the Fetch robot. The Fetch robot receives visual observations from a Primesense Carmine 1.09 short-range RGBD sensor mounted on its head. Accordingly, we render images of 640×480 resolution. To simulate the operation range of the depth sensor, we only render depth values for points that are within 5 meters from the camera. We also provide a binary mask indicating which pixels have valid depth values and concatenate the mask with the depth image as its second channel. We also add zero-mean Gaussian noise of with a standard deviation of 0.05 meter to each pixel. The segmentation mask uses the class labels from NYU40 [1] with each pixel label encoded in the one-hot manner. We subsample the rendered images by a factor of 10, providing us RGB images of $64 \times 48 \times 3$, depth images of $64 \times 48 \times 2$ and segmentation masks of $64 \times 48 \times 40$.

The environment dynamics is simulated for the Fetch robot operating on a planar surface. The robot moves forward and take turns by controlling the velocity of its two wheels, with a wheel radius of 0.065 meters and axis width of 0.375 meters. We add a zero-mean Gaussian with a standard deviation of 0.5 rad/s to both wheels to simulate the noisy dynamics. We check the collisions between the robot and the meshes of the environment. The robot will be reset to the previous pose when it collides by taking the action.

B. Analysis of Memory Factorization

In memory factorization, it is crucial to choose representative centers that have a good coverage of all past observations. Therefore, the centers should be distant from each other in the feature space. Since the memory keeps growing across time, the centers are supposed to be dynamically updated during the task execution instead of remaining as static vectors for all episodes.

In this section, we compare the **farthest point sampling (FPS)** used in SMT with two alternative types of representative centers. We refer to **Window** as the baseline which uses the last $|\tilde{M}|$ time steps in a fixed time window as representative centers. In this way, the centers are dynamically updated but only focus on the most recent history. We also implemented the static inducing points in [2], which we refer to as **Static**. The $|\tilde{M}|$ inducing points are trained as neural network weights and remain static during test time. We compare the performance of the three types of centers on the validation set by setting $|\tilde{M}|$ to be 100. As shown in Table. 1, FPS achieves comparable task performance with Static in the roaming task. And it outperforms the two baselines in coverage and search.

Center Type	Roaming	Coverage	Search
Window	378.0	451.6	438.7
Static	**383.9**	457.96	445.9
FPS	383.3	**481.2**	**462.7**

Table 1. Performance of using different types of representative center in memory factorization. Average rewards are listed.

C. Robustness to Noisy Dynamics

In this section, we evaluate the robustness of our model to noisy environment dynamics. Instead of retrieving the ground truth poses p_t from the environment, we estimate the pose using the action a_t. a_t provides us translation and rotation of the agent w.r.t. the previous pose. Thus we can estimate the $\hat{p}_{t+1}$ at each time step using a_t and the previous estimation $\hat{p}_t$. When there is no noise, $\hat{p}_t$ is equivalent to p_t. With the Gaussian noise added at each time step, the noise added to $\hat{p}_t$ will be a Gaussian process. Therefore, when computing the observation embedding using the relative poses, recent steps suffer less from the noisy dynamics.

In our design of SMT, we use a positional embedding of the time step similar to [3], but with exponential functions instead of sinusoidal functions. The positional embedding provides temporal information of each time step for the policy. Sinusoidal function is periodic and provides only relative temporal information. In contrast, the exponential function is monotonic and represents how recent each time step is. In the long-horizon tasks we are interested in, we believe relative temporal information is not sufficient for the agent

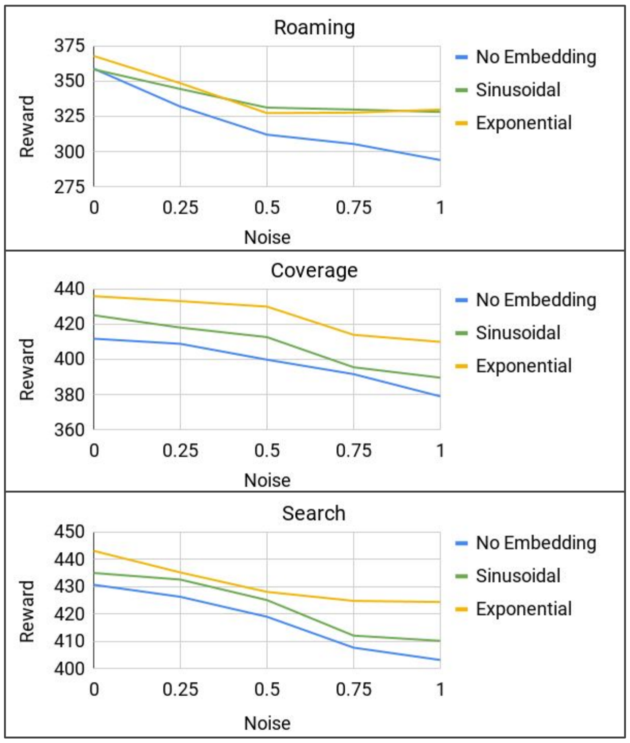

Figure 1. Robustness to noisy dynamics. We compare three positional embedding methods under noisy environment dynamics. The standard deviation of the noise is swept from 0.0 to 1.0.

to understand long-term dependencies.

To validate this assumption, we compare the exponential embedding with the two baselines. **No embedding** does not embed the positional embedding of the time step. **Sinusoidal** uses the same sinusoidal embedding function as in [3]. We sweep the standard deviation of the noise from 0.0 to 1.0 and evaluates the average rewards on the validation set. In practice, we found the temporal information not only improves the performance given clean observations, but also helps leverage the noisy environment dynamics across time. As shown in Fig. 1, the average rewards decrease with more noises in dynamics. Sinusoidal and exponential embeddings both mitigate the performance drop. In the roaming task, the two embedding methods have comparable effects. While in coverage and search, exponential embedding has the superior performance.

D. More Visualization

We present more visualization of the agent behaviors for roaming in Fig. 2, for coverage in Fig. 3 and for search in Fig. 4. As in the main paper, we visualize the trajectories from the top-down view as green curves, with white and black dots indicating the starting and ending points. Navigable area are masked in dark purple with red lines as the collision boundaries. In the coverage task (Fig. 3), we mark the covered cells in pink. In the search task (Fig. 4), we mark target objects with yellow masks. These figures demonstrate similar behaviors as analyzed in the main paper. The same reactive and LSTM baselines as in the main paper are used to compare with the proposed SMT policy.

References

[1] D. Eigen and R. Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 2650–2658, 2015. 1

[2] J. Lee, Y. Lee, J. Kim, A. R. Kosiorek, S. Choi, and Y. W. Teh. Set transformer. *CoRR*, abs/1810.00825, 2018. 1

[3] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, 2017. 1, 2

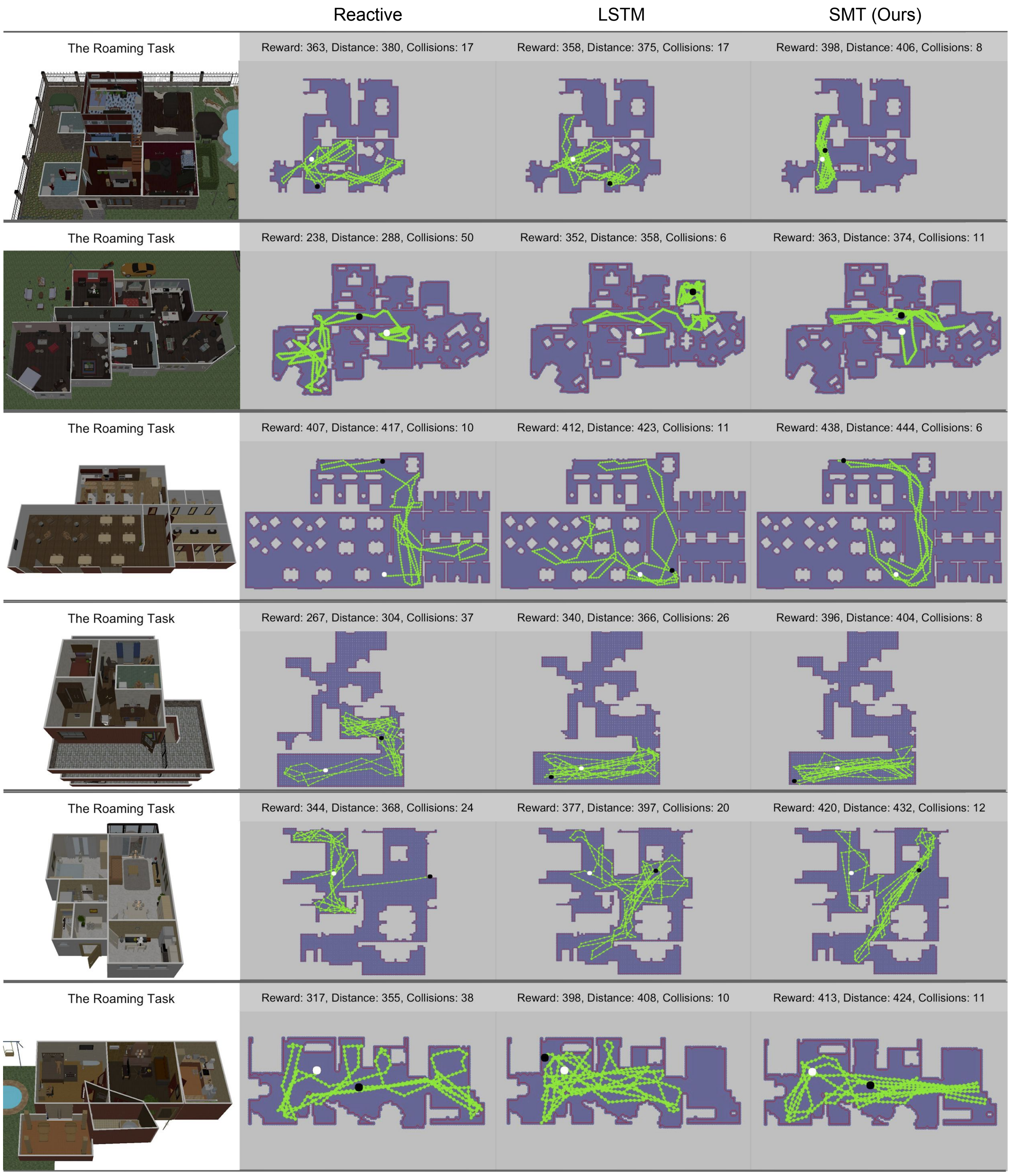
Reactive
LSTM
SMT (Ours)
The Roaming Task
Reward: 363, Distance: 380, Collisions: 17
Reward: 358, Distance: 375, Collisions: 17
Reward: 398, Distance: 406, Collisions: 8
The Roaming Task
Reward: 238, Distance: 288, Collisions: 50
Reward: 352, Distance: 358, Collisions: 6
Reward: 363, Distance: 374, Collisions: 11
The Roaming Task
Reward: 407, Distance: 417, Collisions: 10
Reward: 412, Distance: 423, Collisions: 11
Reward: 438, Distance: 444, Collisions: 6
The Roaming Task
Reward: 267, Distance: 304, Collisions: 37
Reward: 340, Distance: 366, Collisions: 26
Reward: 396, Distance: 404, Collisions: 8
The Roaming Task
Reward: 344, Distance: 368, Collisions: 24
Reward: 377, Distance: 397, Collisions: 20
Reward: 420, Distance: 432, Collisions: 12
The Roaming Task
Reward: 317, Distance: 355, Collisions: 38
Reward: 398, Distance: 408, Collisions: 10
Reward: 413, Distance: 424, Collisions: 11

Figure 2. Visualization of the agent behaviors in the Roaming Task.

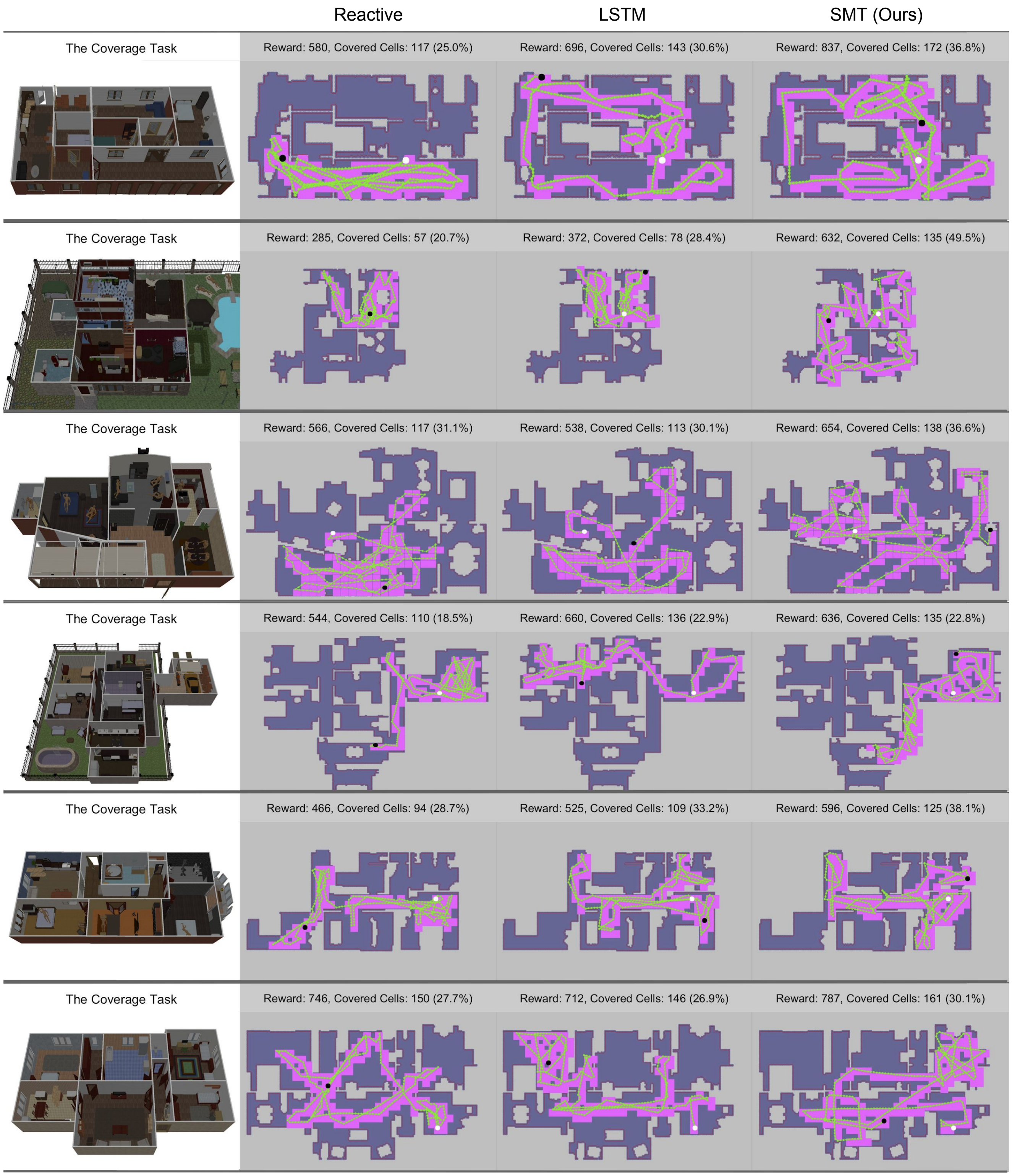
Reactive
LSTM
SMT (Ours)
The Coverage Task
Reward: 580, Covered Cells: 117 (25.0%)
Reward: 696, Covered Cells: 143 (30.6%)
Reward: 837, Covered Cells: 172 (36.8%)
The Coverage Task
Reward: 285, Covered Cells: 57 (20.7%)
Reward: 372, Covered Cells: 78 (28.4%)
Reward: 632, Covered Cells: 135 (49.5%)
The Coverage Task
Reward: 566, Covered Cells: 117 (31.1%)
Reward: 538, Covered Cells: 113 (30.1%)
Reward: 654, Covered Cells: 138 (36.6%)
The Coverage Task
Reward: 544, Covered Cells: 110 (18.5%)
Reward: 660, Covered Cells: 136 (22.9%)
Reward: 636, Covered Cells: 135 (22.8%)
The Coverage Task
Reward: 466, Covered Cells: 94 (28.7%)
Reward: 525, Covered Cells: 109 (33.2%)
Reward: 596, Covered Cells: 125 (38.1%)
The Coverage Task
Reward: 746, Covered Cells: 150 (27.7%)
Reward: 712, Covered Cells: 146 (26.9%)
Reward: 787, Covered Cells: 161 (30.1%)

Figure 3. Visualization of the agent behaviors in the Coverage Task.

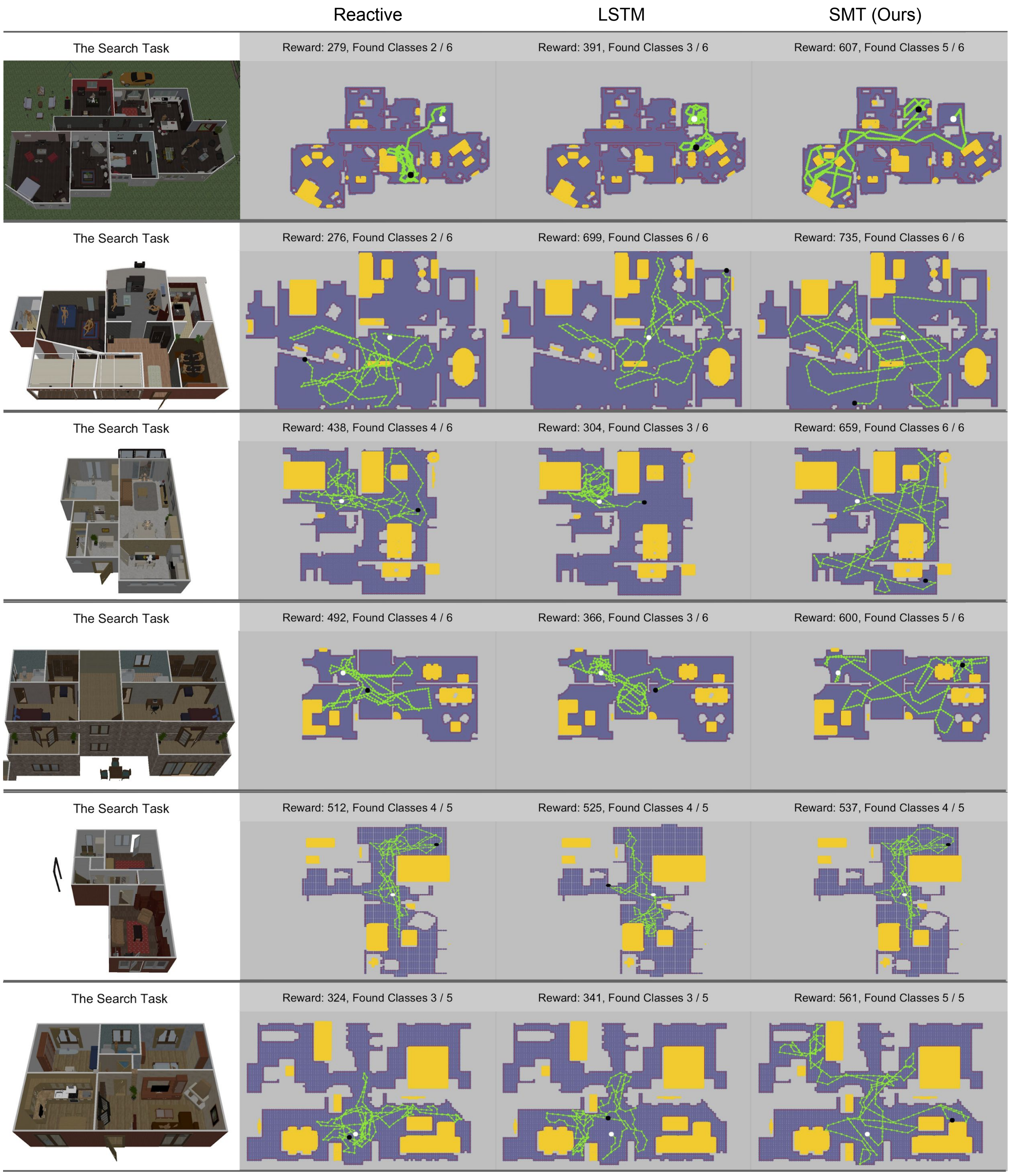
Reactive
LSTM
SMT (Ours)
The Search Task
Reward: 279, Found Classes 2 / 6
Reward: 391, Found Classes 3 / 6
Reward: 607, Found Classes 5 / 6
The Search Task
Reward: 276, Found Classes 2 / 6
Reward: 699, Found Classes 6 / 6
Reward: 735, Found Classes 6 / 6
The Search Task
Reward: 438, Found Classes 4 / 6
Reward: 304, Found Classes 3 / 6
Reward: 659, Found Classes 6 / 6
The Search Task
Reward: 492, Found Classes 4 / 6
Reward: 366, Found Classes 3 / 6
Reward: 600, Found Classes 5 / 6
The Search Task
Reward: 512, Found Classes 4 / 5
Reward: 525, Found Classes 4 / 5
Reward: 537, Found Classes 4 / 5
The Search Task
Reward: 324, Found Classes 3 / 5
Reward: 341, Found Classes 3 / 5
Reward: 561, Found Classes 5 / 5

Figure 4. Visualization of the agent behaviors in the Search Task.