Supplementary Materials

October 2, 2020

1 Experiment Details

1.1 Environment

The 2D worlds used in the experiment are all 100×100 grid spaces. The sensor observation is normalized to $[-1, 1]$ before feeding to the agent. The size of MNIST digit observation is 25×25 and the size of top-down observation is 50×50. The pixel value of MNIST digit and top-down observations are also normalized to $[-1, 1]$.

We use following approach to generate MNIST digits instead of using generative models. First, we choose n different MNIST digit images from the dataset, denoted as $\{o_1, o_2, ..., o_n\}$. Then we assign each x_i with a center location $(x_i, y_i) \in [0, 99] \times [0, 99]$. Then for agent at location $(x, y) \in [0, 99] \times [0, 99]$, we generate the observation o using a weighted sum

$$o = \frac{\sum_{i=1}^{n} w_i o_i}{\sum_{i=1}^{n} w_i},$$

where

$$w_i = \exp\left(-0.01\sqrt{(x - x_i)^2 + (y - y_i)^2}\right).$$

In the experiments, we randomly select 4 different MNIST digit images and place them at $(10, 10)$, $(10, 90)$, $(90, 10)$ and $(90, 90)$ respectively.

1.2 Embedding Network

The embedding networks used in the experiments are shown in Table 1, Table 2 and Table 3. In the table, 'FC' stands for fully connected layer and 'Conv' stands for convolution layer. We train these networks using Adam optimizer Kingma and Ba [2015] with learning rate 0.0001, $\beta = (0.9, 0.999)$. We draw samples uniformly from a replay pool of size 100k. In each iteration, we update this replay pool with 400 samples. The batch size is 64. The size of ensemble used in top-down maze is 8. In the planning experiment, r is set to 20. In the exploration experiment, r is set to 8.

1.3 Goal-conditioned Agent

The goal-conditioned agent used in the planning experiment is the locomotion network proposed in SPTM Savinov et al. [2018]. The locomotion networks used in planning experiments are shown in Table 4, Table 5 and Table 6.

1.4 Q-network

The Q-networks used in the experiments are shown in Table 7, Table 8, Table 9 and Table 10. We set the discount factor $\gamma = 0.95$, ε-greedy factor $\varepsilon = 0.9$ in all the exploration experiments. We train these networks using Adam optimizer with learning rate 0.0003, $\beta = (0.9, 0.999)$. The batch size is 256. The size of HER is 100k. The goal number k of HER is set to 4 in MountainCar experiments and is set to 16 in Snake maze experiments. The memory pool of each vertex holds at most 500 samples. For better exploration, the DQN agent chooses action randomly after $0.9T$ steps in each iteration, where T is the total steps of each iteration.

1.5 Planning

Since our graph is an abstraction of environment, it's not necessary to plan at each step. Instead, we carry out planning every 10 steps in planning experiments and exploration experiments.

2 Further Analysis

2.1 Effect of r

We measure the effect of radius r and find it is consistent with our expectation. We use "Empty" map and let $r = 8, 16, 32$ for different levels of abstraction. The graph generation results are illustrated in Figure 1 (b). The state coverage of each vertex does increase as we increase r, leading to a higher level of abstraction.

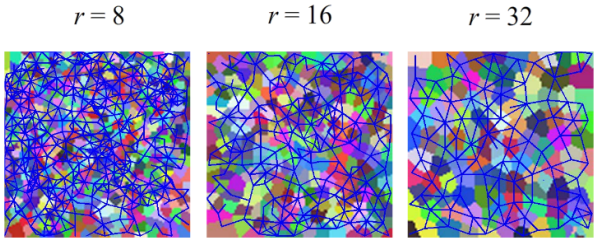

Figure 1: Effect of r.

2.2 Visualizing Embedding Space

We use t-SNE [van der Maaten and Hinton, 2008] to visualize the embedding space of the sensor and MNIST digit maze on the plane. The result is shown in Figure 2. Though only exposed to visual features, TOMA still successfully captures the underlying topological map of the world.

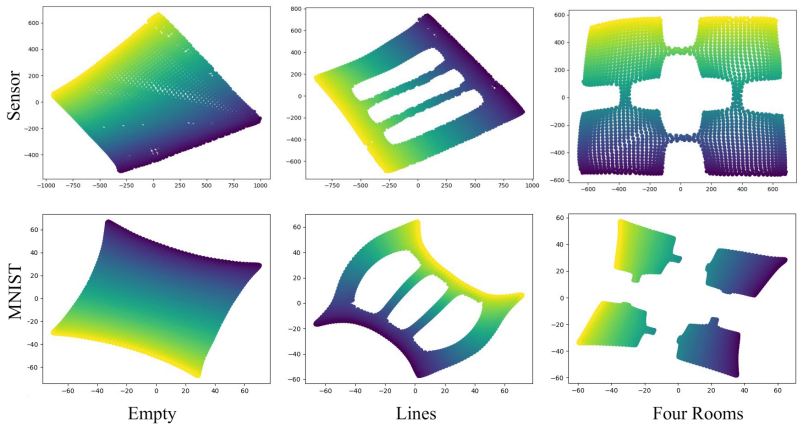

Figure 2: Visualization of embedding space.

References

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proceedings of 3rd International Conference on Learning Representations, ICLR*, 2015.

Nikolay Savinov, Alexey Dosovitskiy, and Vladlen Koltun. Semi-parametric topological memory for navigation. In *Proceedings of the 6th International Conference on Learning Representations (ICLR)*, 2018.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

Table 1: Embedding network for sensor maze and MountainCar.

FC 64, ReLU
FC 64, ReLU
FC 8

Table 2: Embedding network for MNIST digit maze.

Conv 6x6, channel 16, stride 2, ReLU
Conv 6x6, channel 16, stride 2, ReLU
Conv 3x3, channel 16, stride 2, ReLU, flatten to 256
FC 64, ReLU
FC 8

Table 3: Embedding network for top-down maze.

Conv 5x5, channel 64, stride 4, ReLU
Conv 3x3, channel 128, stride 2, ReLU
Conv 3x3, channel 128, stride 2, ReLU
Conv 3x3, channel 128, stride 1, ReLU, flatten to 2048
FC 8

Table 4: Locomotion network for sensor maze.

FC 16, ReLU
Concatenate two encoded vectors.
FC 32, ReLU
FC 32, ReLU
FC 4

Table 5: Locomotion network for MNIST digit maze.

Conv 4x4, channel 32, stride 2, ReLU
Conv 4x4, channel 32, stride 2, ReLU
Conv 4x4, channel 32, stride 2, ReLU, flatten to 512
Concatenate two encoded vectors.
FC 512, ReLU
FC 512, ReLU
FC 4

Table 6: Locomotion network for top-down maze.

Conv 5x5, channel 64, stride 4, ReLU
Conv 3x3, channel 64, stride 2, ReLU
Conv 3x3, channel 64, stride 2, ReLU
Conv 3x3, channel 64, stride 1, ReLU, flatten to 1024
Concatenate two encoded vectors.
FC 512, ReLU
FC 512, ReLU
FC 4

Table 7: Q-network for MountainCar.

FC 64, ReLU
FC 64, ReLU
FC 3

Table 8: Q-network for sensor maze.

FC 300, ReLU
FC 200, ReLU
FC 4

Table 9: Q-network for MNIST digit maze.

Conv 6x6, channel 32, stride 4, ReLU
Conv 4x4, channel 64, stride 2, ReLU
Conv 4x4, channel 64, stride 1, ReLU, flatten to 1024
FC 256, ReLU
FC 4

Table 10: Q-network for top-down maze.

Conv 6x6, channel 64, stride 4, ReLU
Conv 4x4, channel 128, stride 2, ReLU
Conv 4x4, channel 128, stride 2, ReLU, flatten to 2048
FC 512, ReLU
FC 4