Supplementary Material: Control of Memory, Active Perception, and Action in Minecraft

Junhyuk Oh JUNHYUK@UMICH.EDU
Valliappa Chockalingam VALLI@UMICH.EDU
Satinder Singh BAVEJA@UMICH.EDU
Honglak Lee HONGLAK@UMICH.EDU
Computer Science & Engineering, University of Michigan

A. Implementation Details

A.1. Hyperparameters

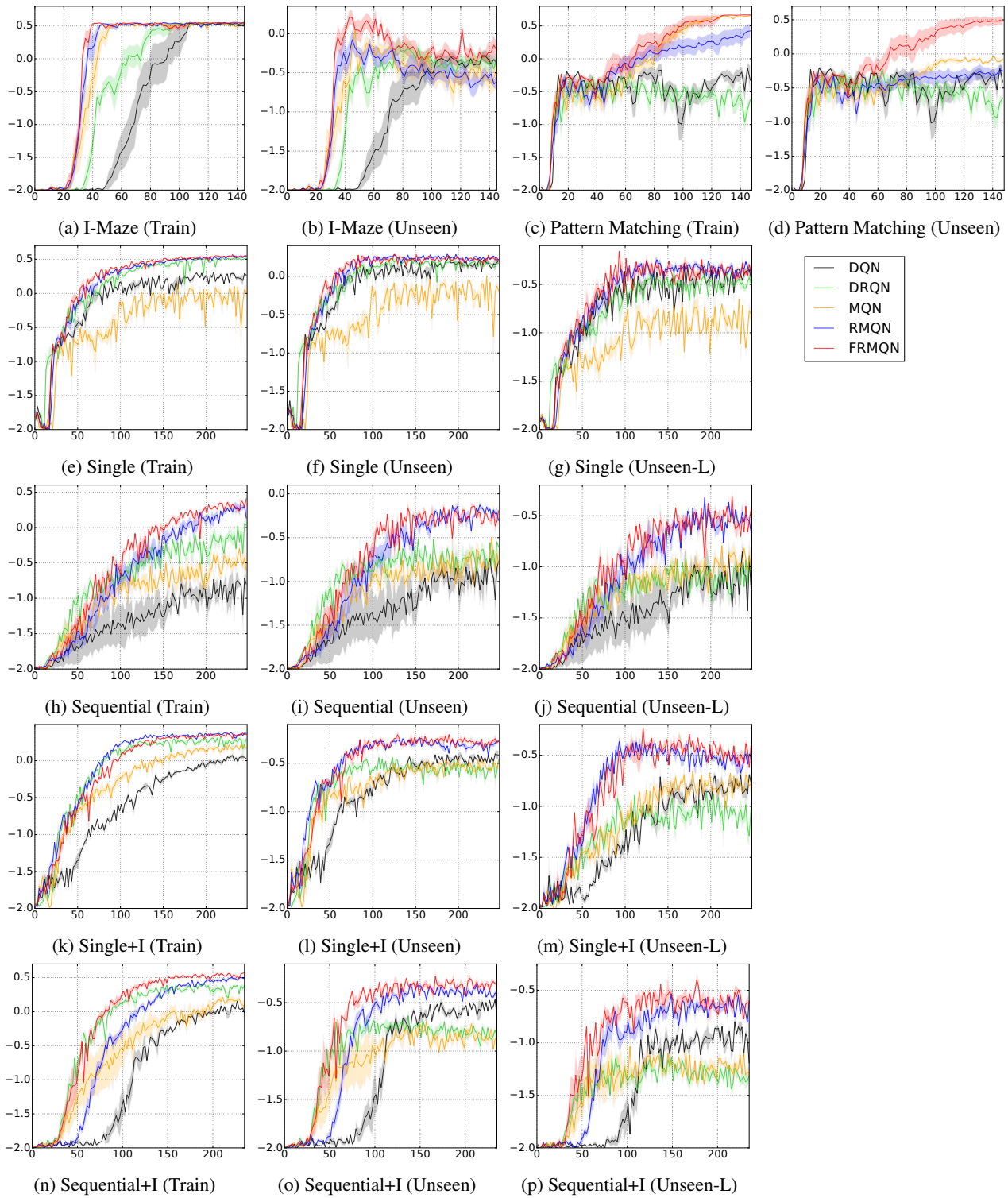
For all architectures, the first convolution layer consists of $32, 4 \times 4$, filters with a stride of 2 and a padding of 1. The second convolution layer consists of $64, 4 \times 4$, filters with a stride of 2 and a padding of 1. In Deep Q-Learning, batch size of 32 and discount factor of 0.99 are used. We used a replay memory size of 10^6 for random mazes and 5×10^4 for I-Maze and Pattern Matching tasks. We linearly interpolated ϵ from 1 to 0.1 for the initial 10^6 steps in the ϵ-greedy policy. We chose the best learning rate from $\{0.0001, 0.00025, 0.0005, 0.001\}$ that does not lead to value function explosion depending on the tasks and architectures. The chosen learning rates are shown in Table 1. The parameter is updated after every 4 steps. RMSProp was used with a momentum of 0.95 and a momentum of squared gradients of 0.95. Gradients were clipped at $l2$-norm of 20 to prevent divergence. We used "soft" target Q-network updates with a momentum of 0.999 as suggested by (Lillicrap et al., 2016).

Table 1. Learning rates.

TASK	DQN	DRQN	MQN	RMQN	FRMQN
I-MAZE	0.00025	0.0005	0.0005	0.0005	0.0005
MATCHING	0.00025	0.001	0.0005	0.0005	0.0005
SINGLE	0.0001	0.00025	0.0001	0.00025	0.00025
SEQ	0.00025	0.0005	0.00025	0.00025	0.00025
SINGLE+I	0.0001	0.0005	0.00025	0.0005	0.00025
SEQ+I	0.00025	0.001	0.00025	0.00025	0.0005

A.2. Map Generation for Pattern Matching

There are a total of 512 possible visual patterns in a 3×3 room with blocks of two colors. We randomly picked 250 patterns and generated two maps for each pattern: one that contains the same pattern in two rooms and another that has a different randomly generated pattern in one of the rooms that is randomly selected. This produces 500 maps, 250 with identical rooms, and 250 with different rooms, which are used for training. For evaluating generalization, we picked another exclusive set of 250 visual patterns, and generated 500 maps by following the same procedure.



(a) I-Maze (Train) (b) I-Maze (Unseen) (c) Pattern Matching (Train) (d) Pattern Matching (Unseen)

(e) Single (Train) (f) Single (Unseen) (g) Single (Unseen-L)

(h) Sequential (Train) (i) Sequential (Unseen) (j) Sequential (Unseen-L)

(k) Single+I (Train) (l) Single+I (Unseen) (m) Single+I (Unseen-L)

(n) Sequential+I (Train) (o) Sequential+I (Unseen) (p) Sequential+I (Unseen-L)

Figure 1. Learning curves. X-axis and y-axis correspond to the number of training epochs (1 epoch = 10K steps) and the average reward respectively. For I-Maze, 'Unseen' represents unseen maps with different sizes. For Pattern Matching, 'Unseen' represents maps with different visual patterns. For the rest plots, 'Unseen' and 'Unseen-L' indicate unseen topologies with the same sizes and larger sizes of maps, respectively. The performance was measured from 4 runs for random mazes and 10 runs for I-Maze and Pattern Matching.

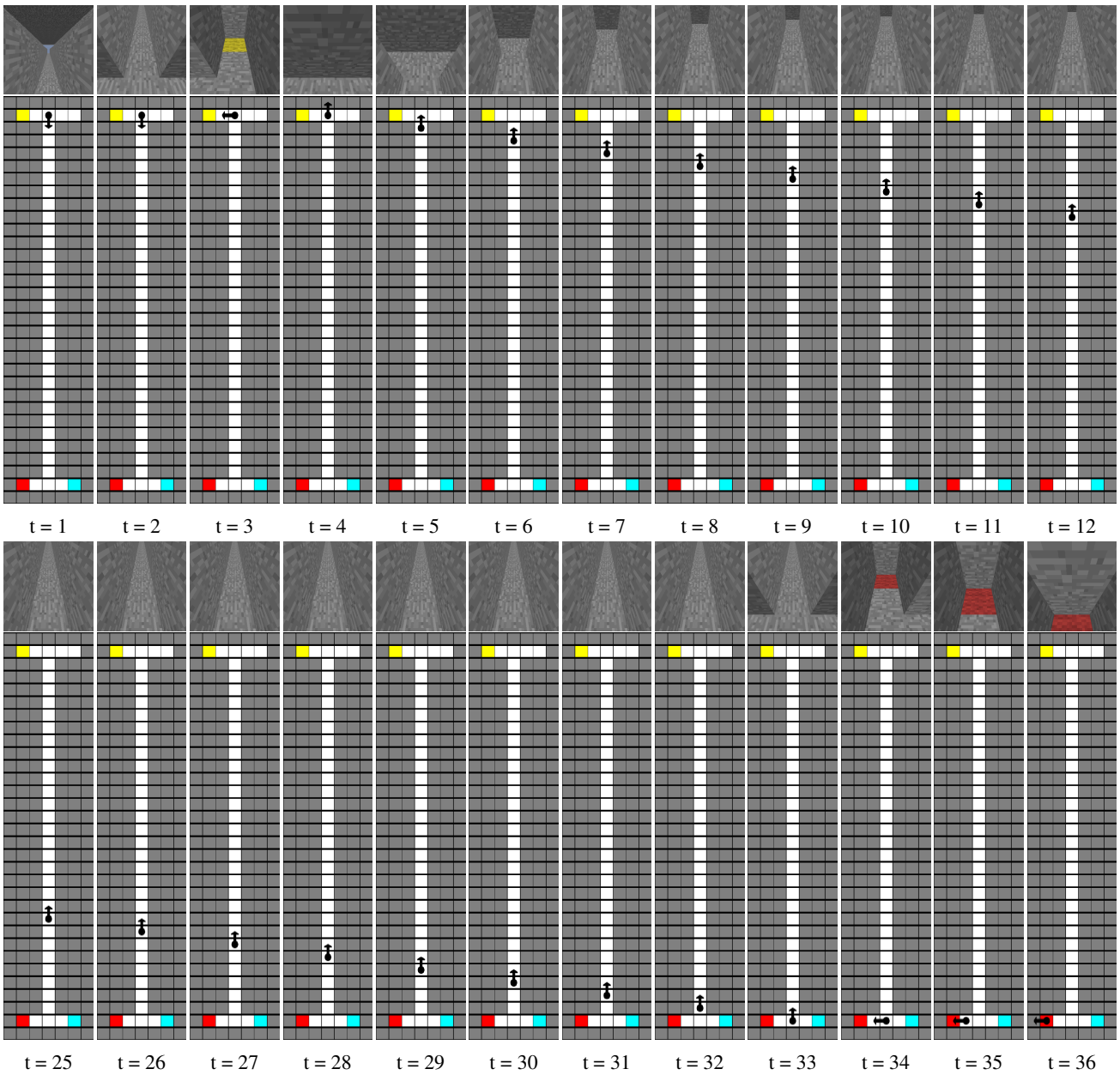


Figure 2. FRMQN's play on an unseen and larger I-maze. The agent successfully completes the task by visiting the red block given the yellow indicator.

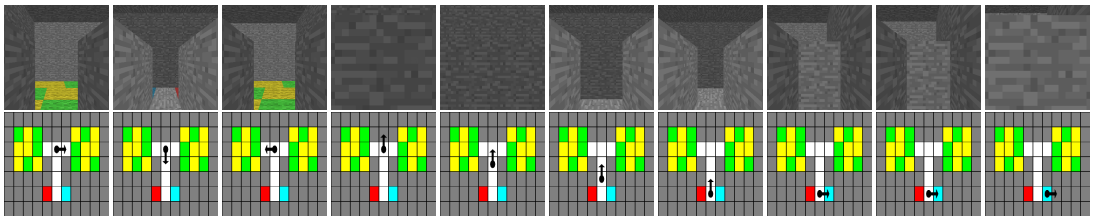


Figure 3. FRMQN's play on a Pattern Matching task. The agent starts by looking at one room and then turns twice to look at the other room. Upon observing the two rooms, the agent uses backward actions repeatedly to move along the vertical corridor. Finally, once it is at the end of the corridor, it decides to turn and move forward to the blue block as the visual patterns of the rooms were identical. Note that the agent's performance is near optimal.

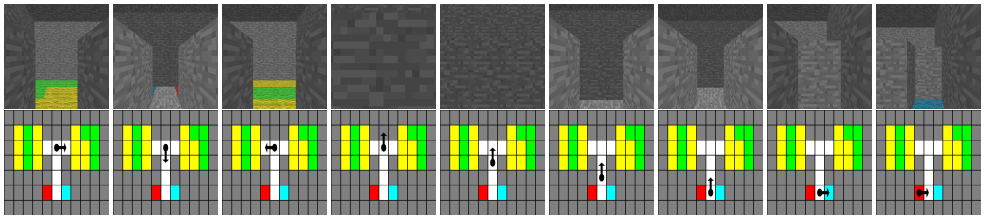


Figure 4. FRMQN's play on a Pattern Matching task. The agent successfully goes to the red goal, given that the visual patterns of the two rooms were different.

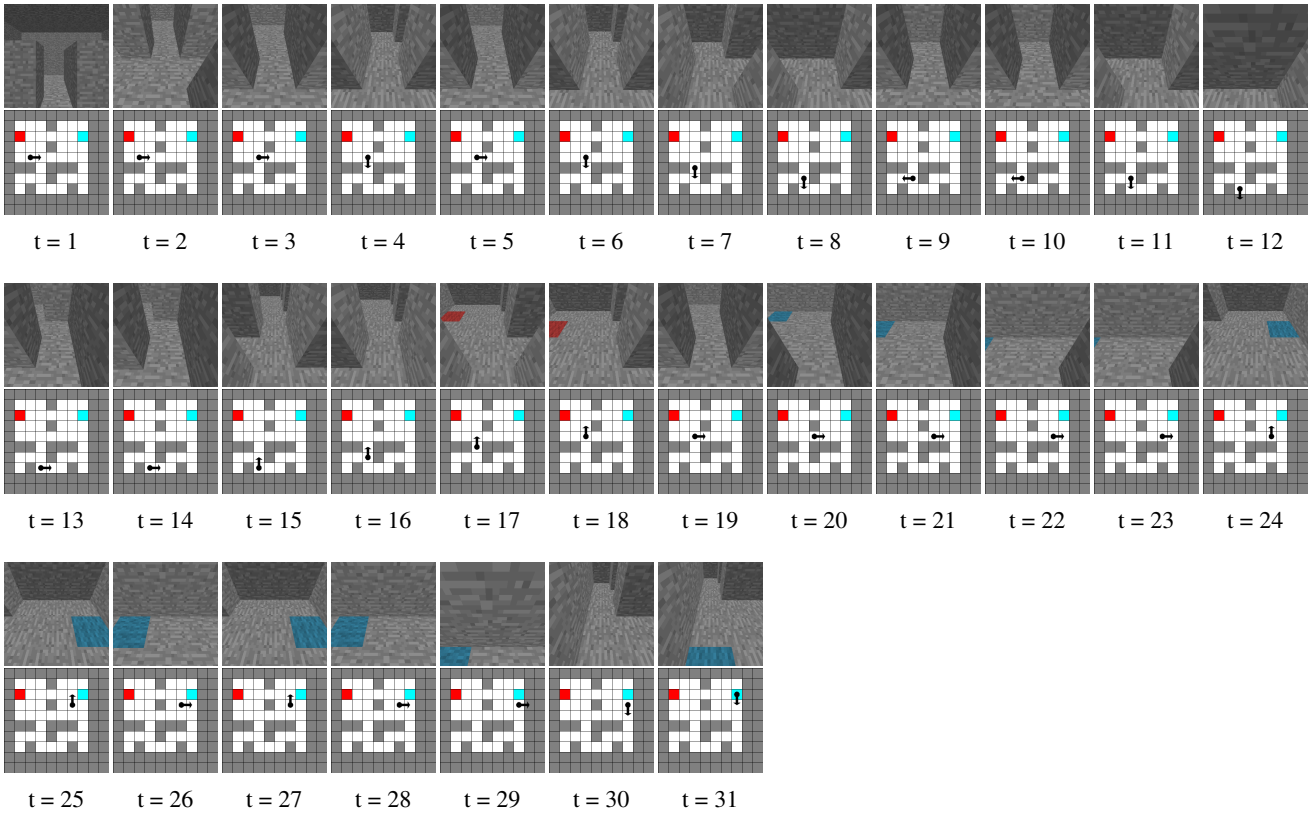


Figure 5. FRMQN's play on an unseen random maze with Single Goal task. As the case with most of the tasks, the agent starts by looking down quickly to see the important stimuli (e.g., goal blocks and indicators) more clearly. The agent then looks around it's vicinity and explores corridors. As soon as the agent sees the blue block, it goes to the block and successfully complete the task.

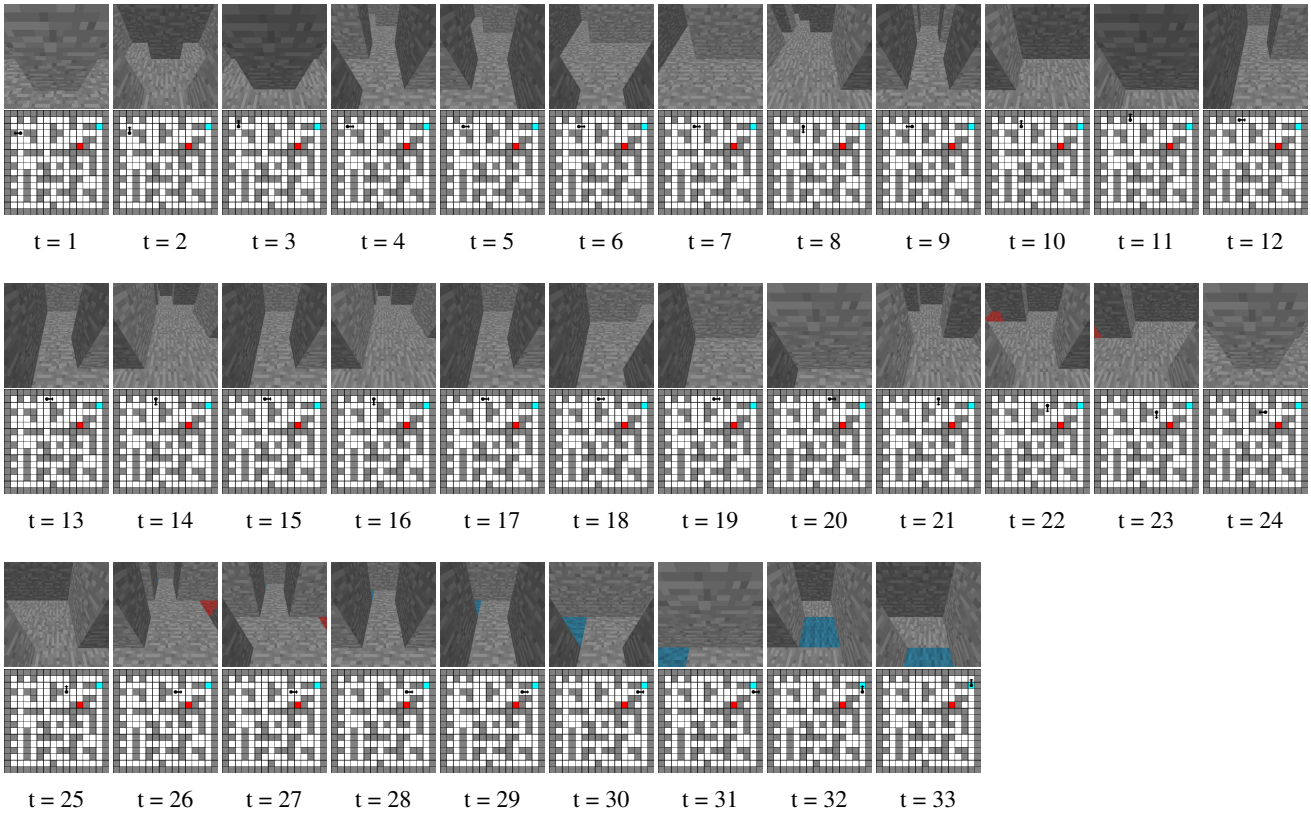


Figure 6. FRMQN's play in an unseen and larger random maze with Single Goal task. The agent explores the map from the top-left side to top-right side, and successfully finds and visits the blue block.

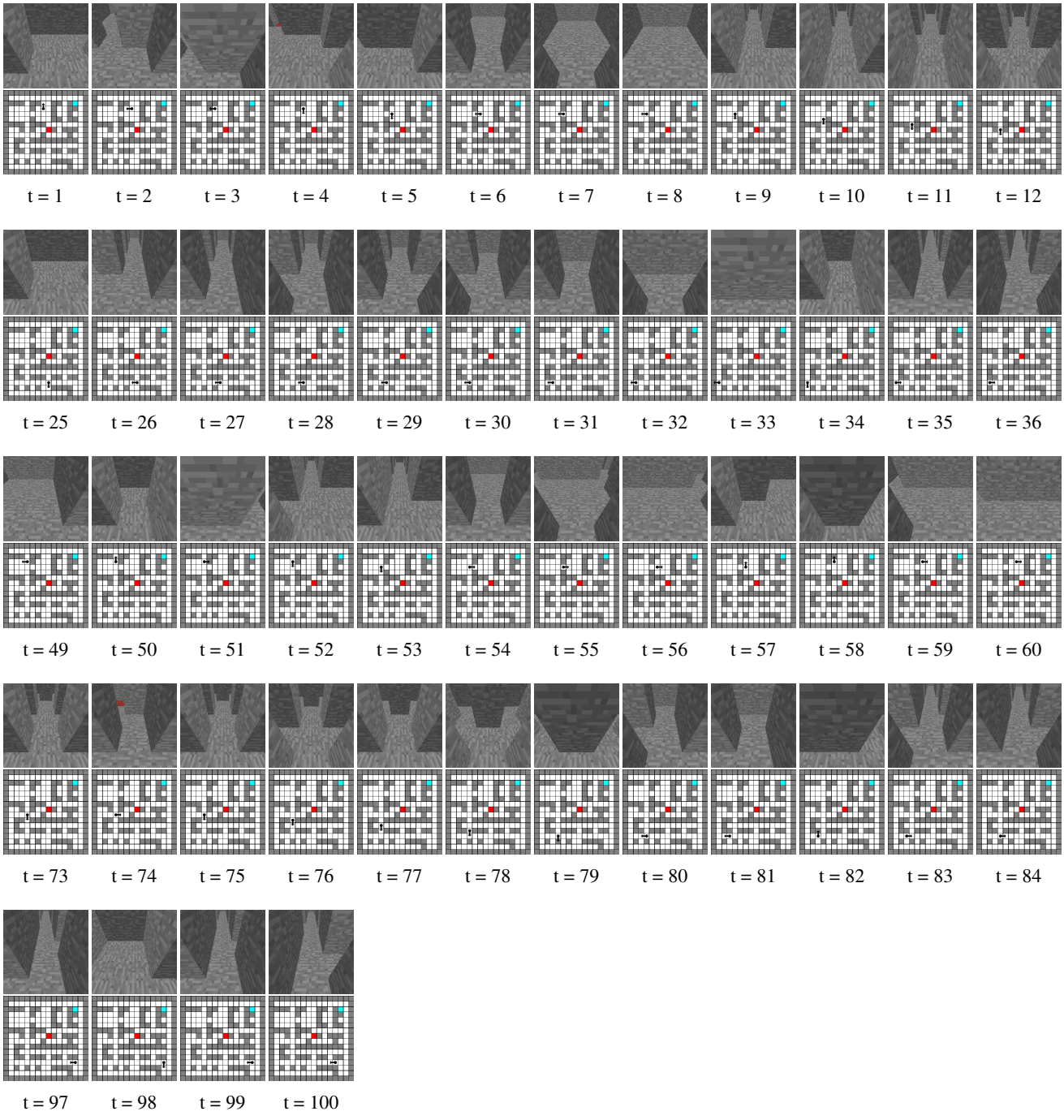


Figure 7. FRMQN's play in an unseen and larger random maze with Single Goal task. Even though the agent explores the entire map in a reasonable way, it fails to find the blue block within 100 steps. This can occur occasionally, especially when the map is quite large and intrinsically complex (e.g., contains many walls giving rise to deep partial observability).

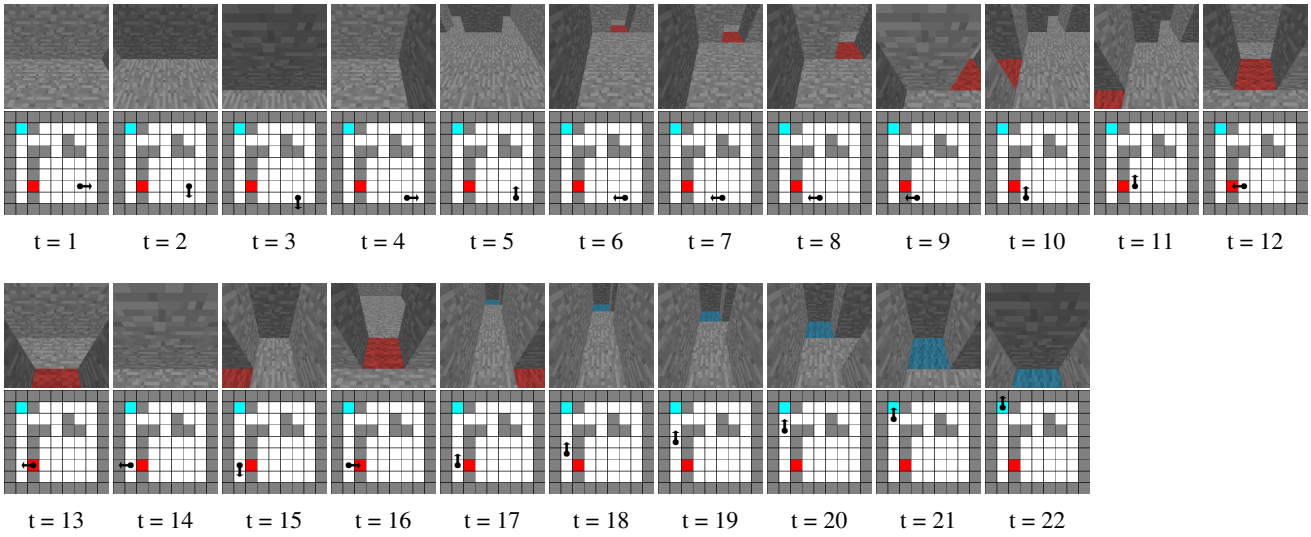


Figure 8. FRMQN's play on a random maze with Sequential Goals task. As the case with most of the other tasks, the agent begins by looking down. It then looks around for the red block (t=1-6). Upon finding the red block, it visits it (t=13). The agents then looks for the blue block (t=14-18), successfully finds it, and hence complete the sequence and task (t=22). Notably the agent does not keep searching for the red block after visiting the red block. This is where memory can be crucial.

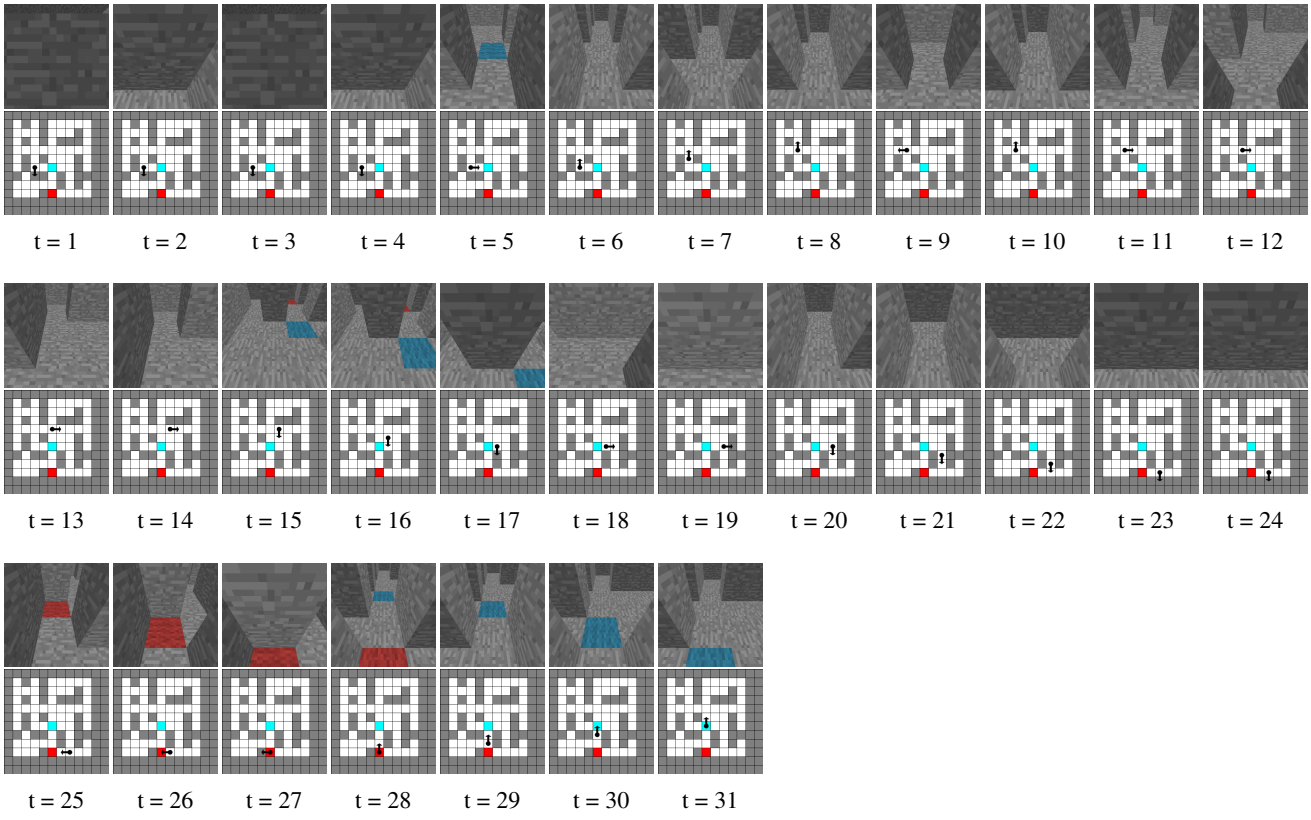


Figure 9. FRMQN's play on an unseen and larger random maze with Sequential Goals task. With the context of the visual observations containing the blue block (t=5,15,16,17), the agent avoids the blue block and keeps searching for the red block based on its memory because it has not visited the red block. After finding and visiting the red block (t=28), it directly goes to the blue block (t=31), completing the sequence in the correct order, and hence the task.

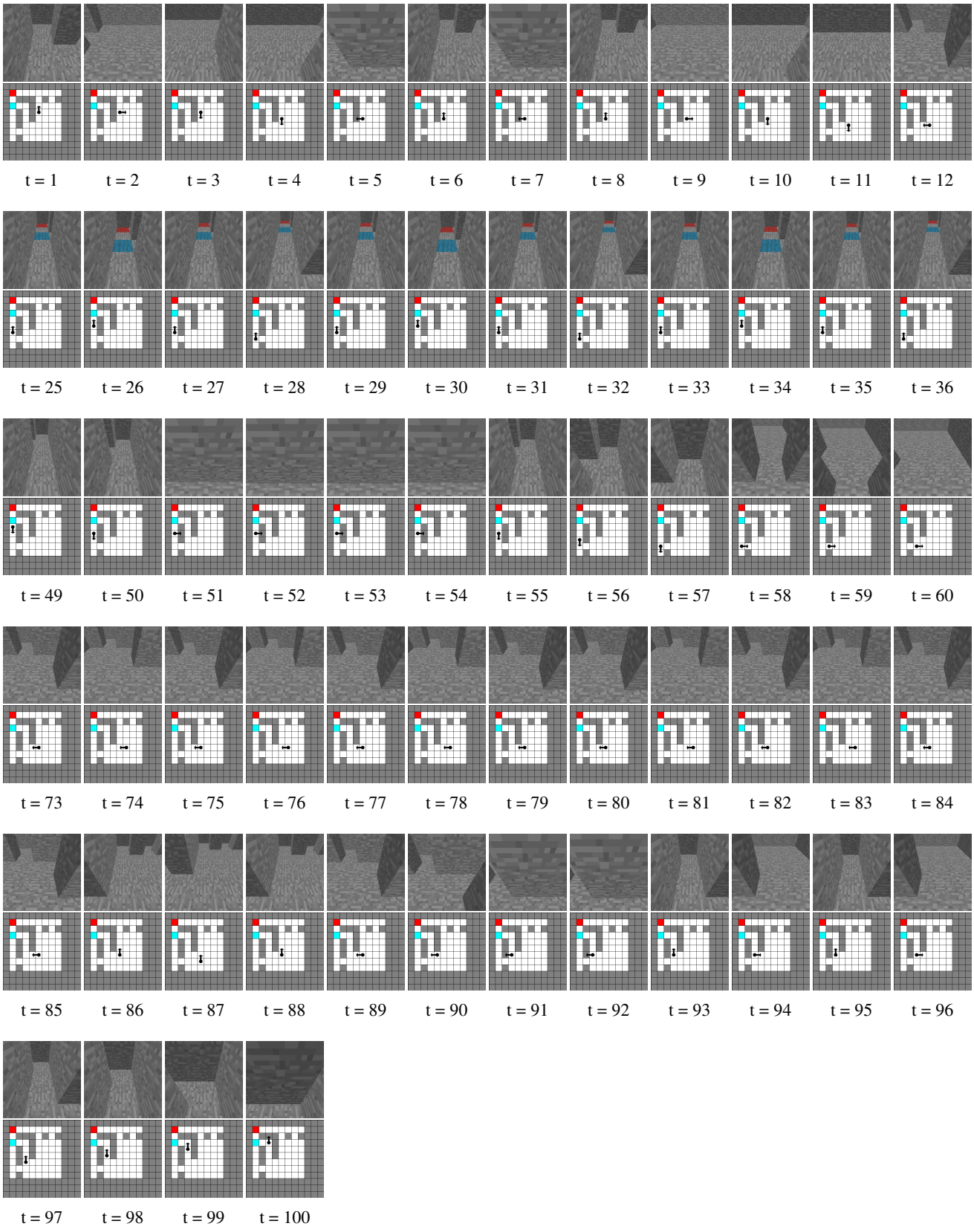


Figure 10. FRMQN's play on an unseen and larger random maze with Sequential Goals task. At t=25, the agent finds both the red and blue blocks. However, visiting the red block (the first goal in the task) by following the corridor found would lead to visiting the blue block, and result in a negative reward. Thus, the agent attempts to search for another route to reach the red goal (t=49-100). But, the agent fails to find another route within the time limit (t=100).

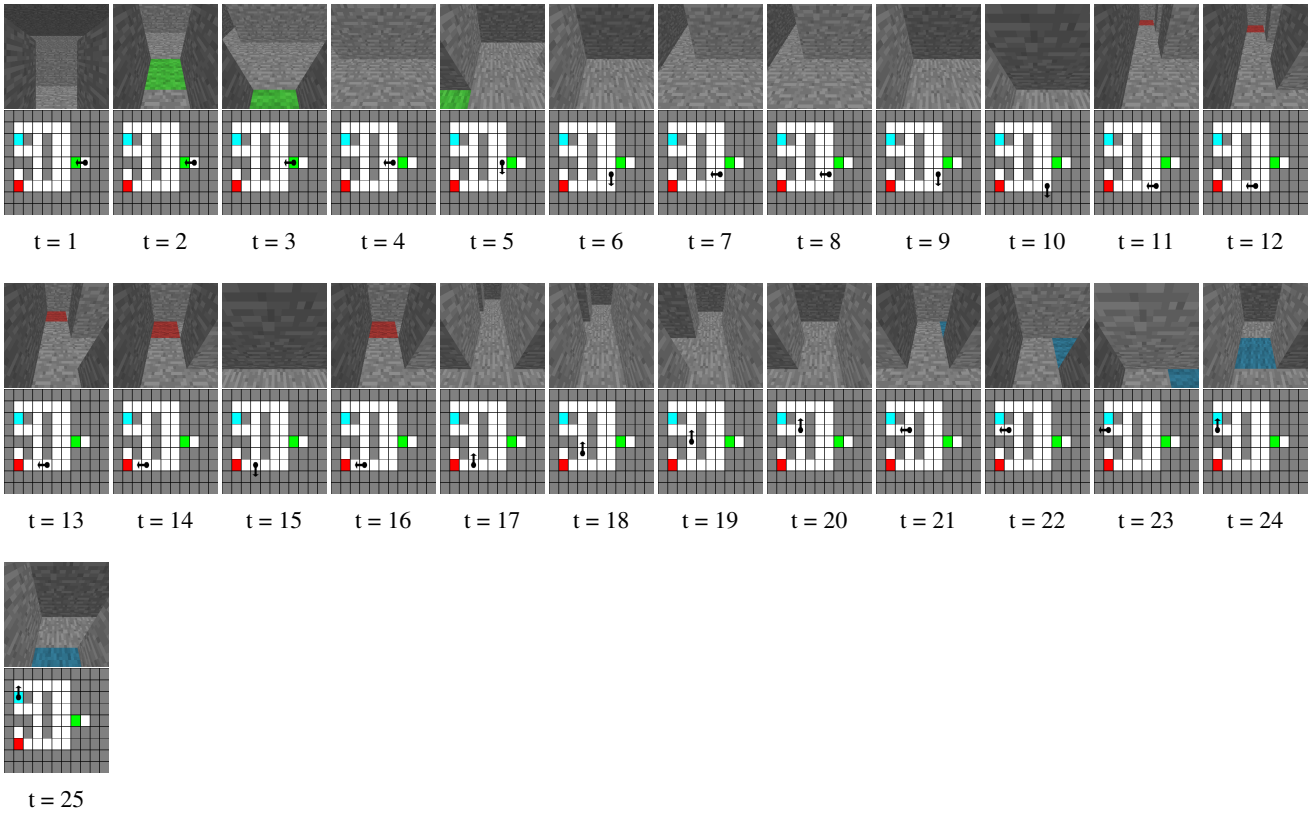


Figure 11. FRMQN's play in a random maze with Single Goal with Indicator task. Upon seeing that the indicator is green in color (t=2), the agent proceeds to explore the map. During its search, it comes across a corridor with a red block (t=11). This is where memory helps. The agent avoids the red block (having observed that the indicator is green). From this we can infer that the agent utilizes its memory appropriately. Later, it successfully completes the task by finding and visiting the blue block (t=22-25).

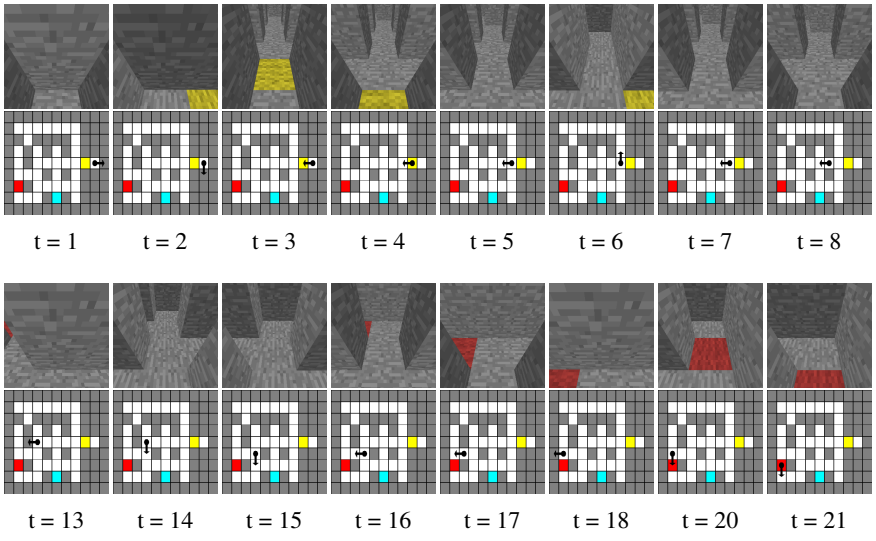


Figure 12. FRMQN's play in a random maze with Single Goal with Indicator task. The agent visits the red block correctly, given the yellow indicator.

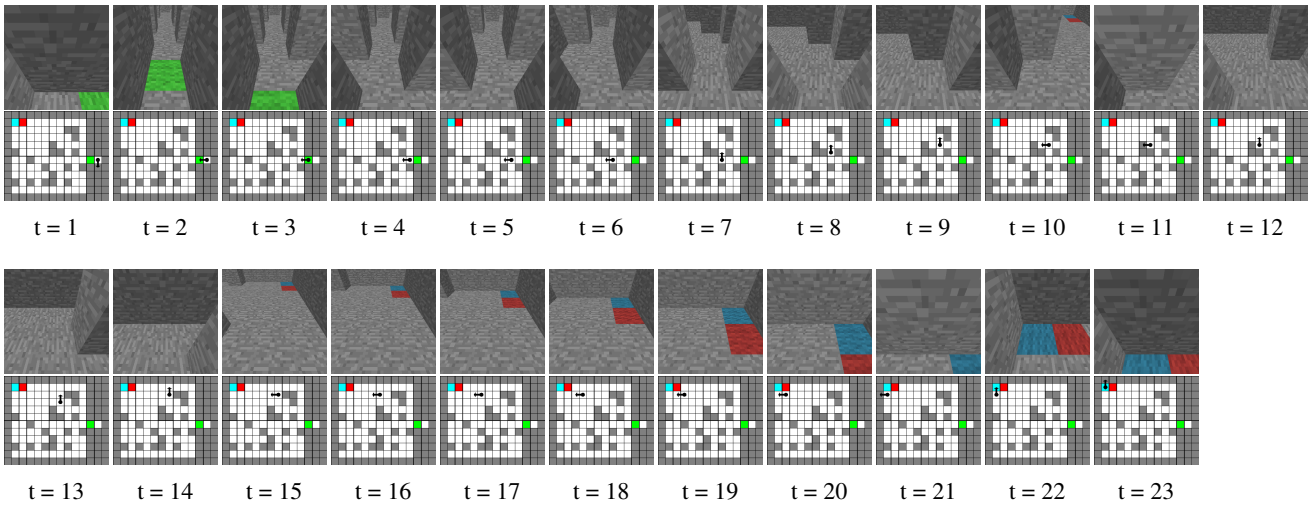


Figure 13. FRMQN's play in an unseen and larger random maze with Single Goal with Indicator task. While the correct goal is far from the indicator, the agent is able to memorize the color of the indicator observed at the beginning (t=3) and visits the correct goal (t=23).

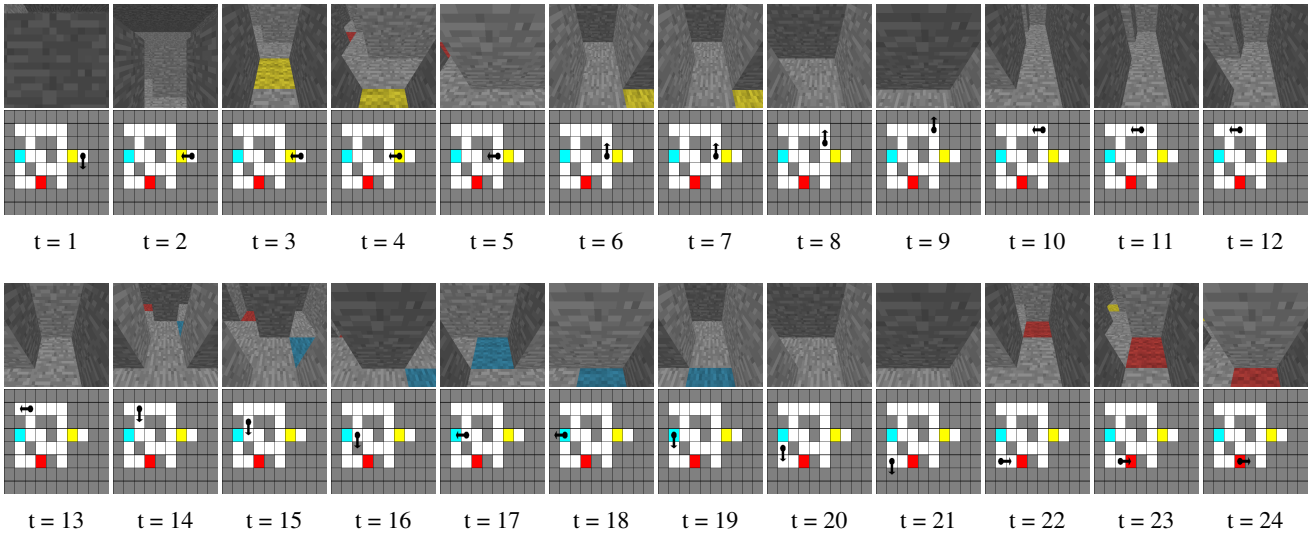


Figure 14. FRMQN's play in a random maze with Sequential Goals with Indicator task. The agent observes that the indicator is yellow at t=3. The agent can see that the red block is near at t=4. Since the task is to first visit the blue block and then the red block if the indicator is yellow, it avoids moving towards the red block (t=6). The agent proceeds by turning and exploring some other corridors. Finally, it gets a glimpse of both the blue and red block (t=14). Using this visual observation along with retrieving from memory visual observations with the indicator present, the agent correctly goes to the blue block and then proceeds to the red block, hence completing the task (t=15-24).

Figure 15. FRMQN's play in an unseen and larger random maze with Sequential Goals with Indicator task. The agent visits the red block first (t=19), given the green indicator (t=3). The agent looks for the blue block which is far apart from the red block (t=19-47). Once the agent finds the blue block, it finishes the task by completing the sequence (t=49).

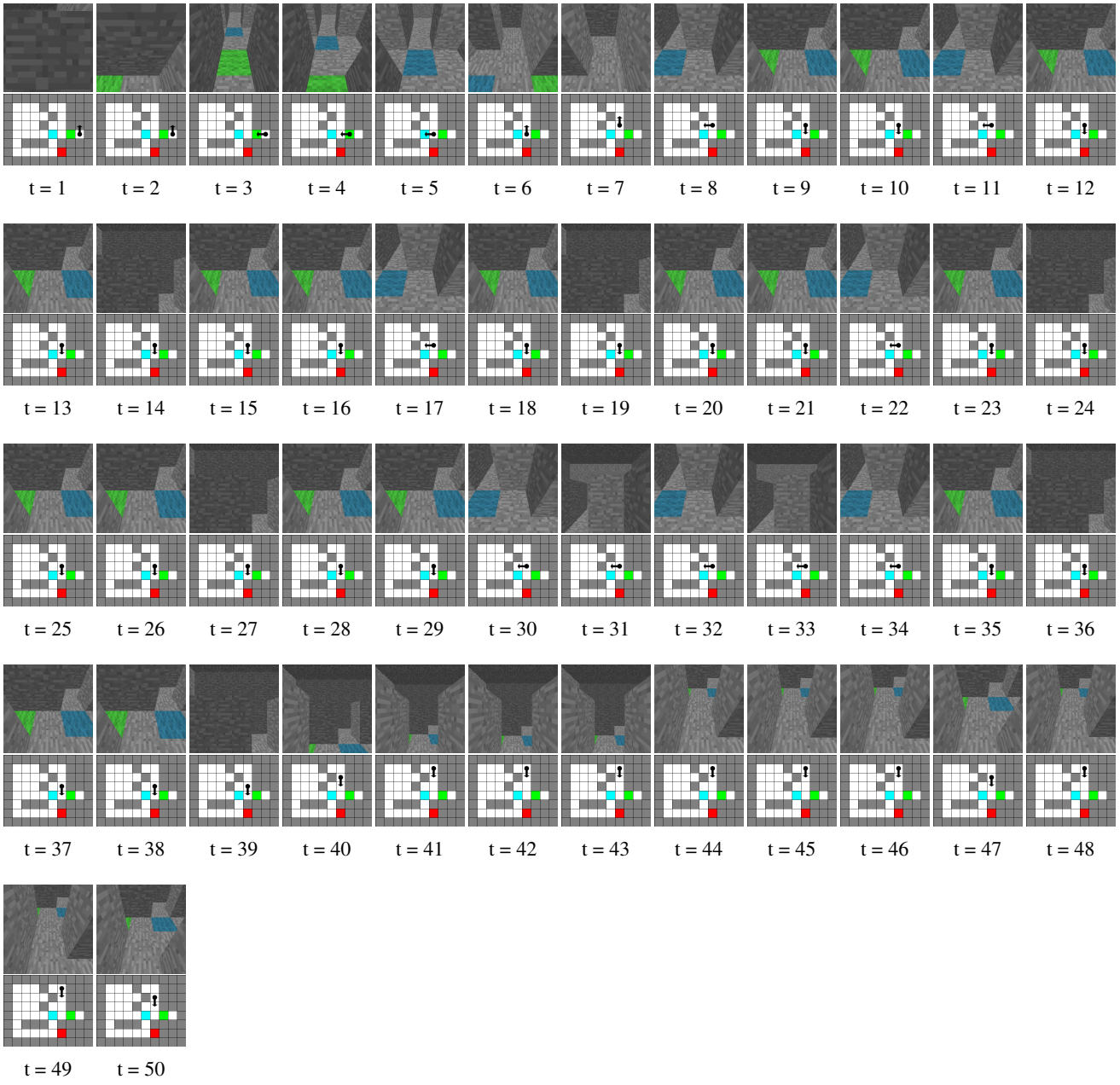


Figure 16. FRMQN's play in a random maze with Sequential Goals with Indicator task. Given that the indicator is green, the agent has to visit the red block first and the blue block later. For this reason, the agent tries to avoid the blue block and search for another route to the red block (t=4-50). However, since there is no path to the red block (that avoids the blue block), the agent keeps searching until the episode terminates.

References

Lillicrap, Timothy P, Hunt, Jonathan J, Pritzel, Alexander, Heess, Nicolas, Erez, Tom, Tassa, Yuval, Silver, David, and Wierstra, Daan. Continuous control with deep reinforcement learning. In *International Conference on Learning Representations*, 2016.