Using Hippocampal Replay to Consolidate Experiences in Memory-Augmented Reinforcement Learning

John Tan Chong Min

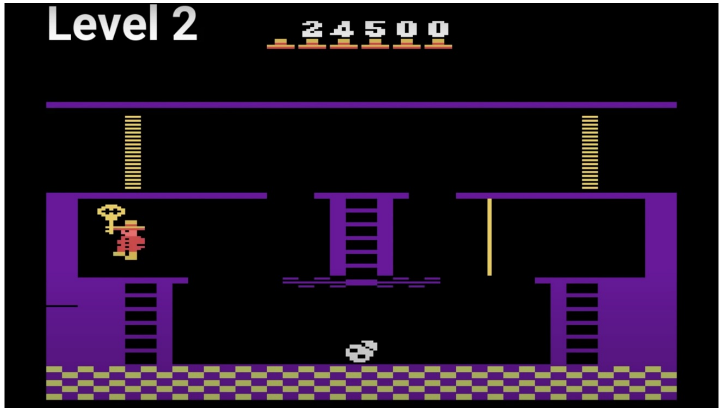
Mehul Motani

Go-Explore

- Using reward alone may be insufficient for sparse reward settings

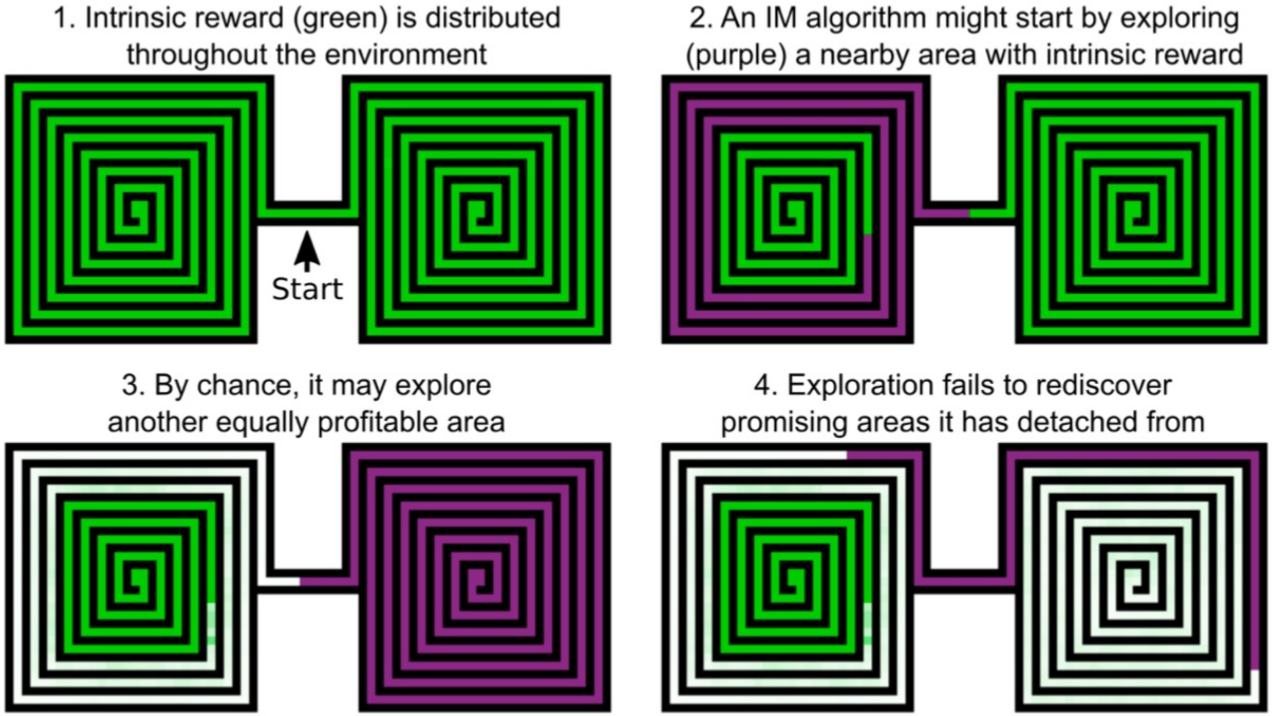
- **Go-Explore** (Ecofett, 2019) uses external memory to update states

- In order to explore more states:
  - **Go:** Jump probabilistically to a state
  - **Explore:** Explore randomly from the state



Montezuma's Revenge,
a game with sparse rewards

Problem with Traditional Agents (1/2)


1. Intrinsic reward (green) is distributed throughout the environment

2. An IM algorithm might start by exploring (purple) a nearby area with intrinsic reward

3. By chance, it may explore another equally profitable area

4. Exploration fails to rediscover promising areas it has detached from

<u>Detachment:</u>

- Intrinsic Reward (IR) may not rediscover previously searched frontiers

Problem with Traditional Agents (2/2)

Epsilon-Greedy:

$$\text{Action at time(t)} \begin{cases} \max Q_t(a) & \text{with probability } 1\text{-}\epsilon \\ \text{any action (a)} & \text{with probability } \epsilon \end{cases}$$

Upper-Confidence Bounds:
(in MCTS)

$$\frac{w_i}{s_i} + c\sqrt{\frac{\ln s_p}{s_i}}$$

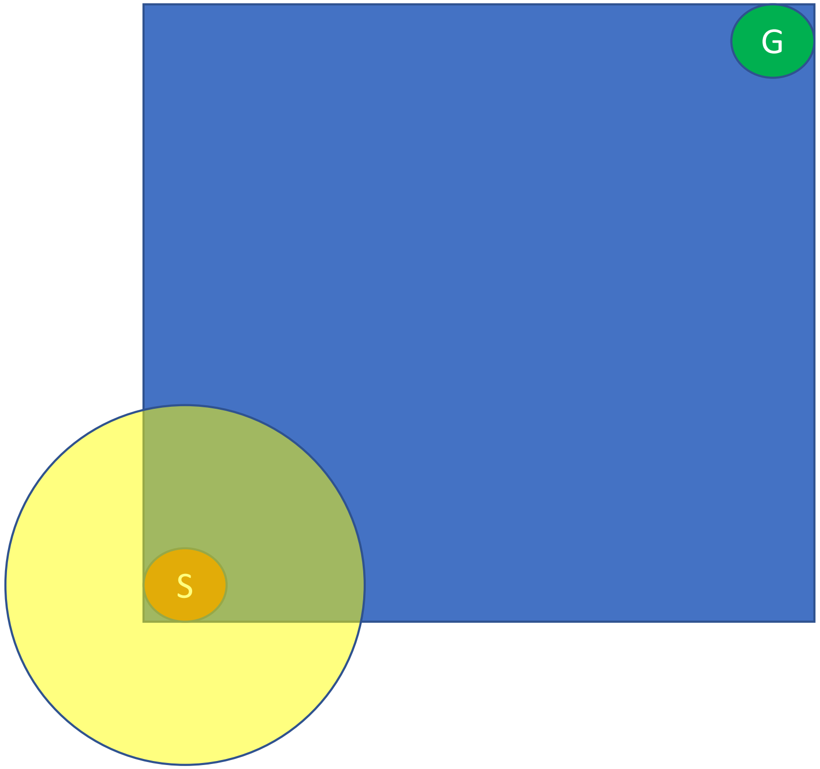
Entropy regularization:
(in Policy-based methods)

$$L_t^{CLIP+VF+S}(\theta) = \hat{\mathbb{E}}_t\left[L_t^{CLIP}(\theta) - c_1 L_t^{VF}(\theta) + c_2 S[\pi_\theta](s_t)\right]$$

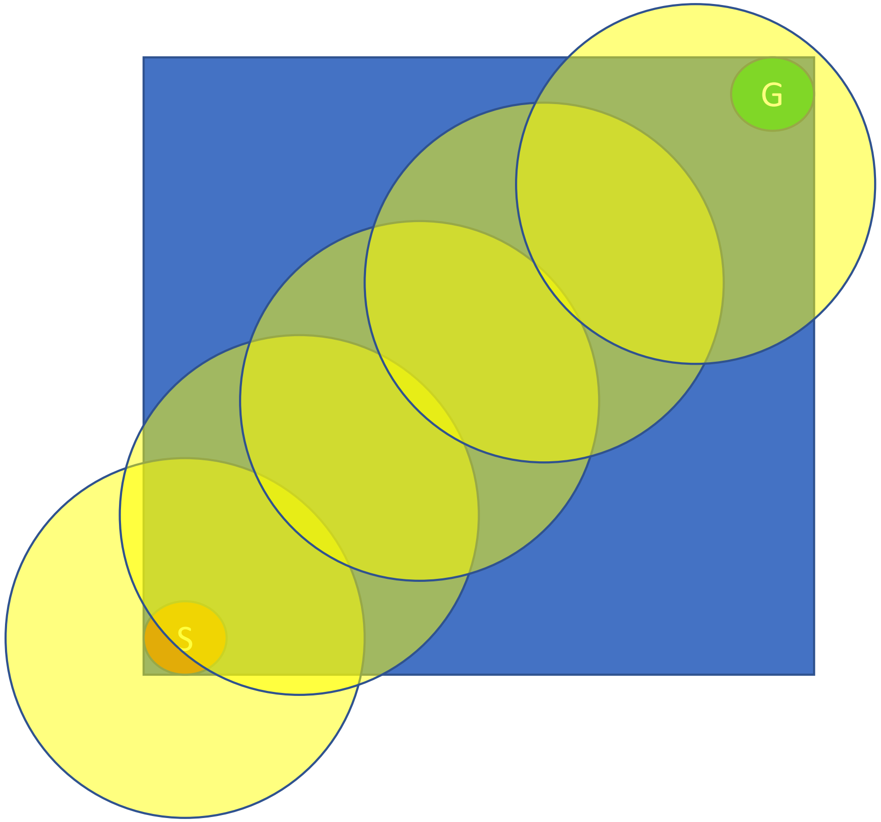
Derailment:

- Agent wants to reach a previously good state, but is hindered by exploration

The torchlight analogy



Exploration from starting state



Exploration from frontier

Agents

1. Random. This agent chooses a valid move randomly, and serves as a worst-case baseline.
2. Go-Explore. Go-Explore was implemented similar to Ecoffet et al. (2019, 2021), except that we select states deterministically in the “Go" phase for faster training.
3. Go-Explore-Count. While Go-Explore uses a random policy for exploration, Go-Explore-Count performs the best action based on a count-based selection function.
4. Explore-Count. Explore-Count is Go-Explore-Count without the “Go" phase.

How to “Go” and “Explore”

- Random exploration can be inefficient

- **Solution –** Use Deterministic Selection to balance explore and exploit

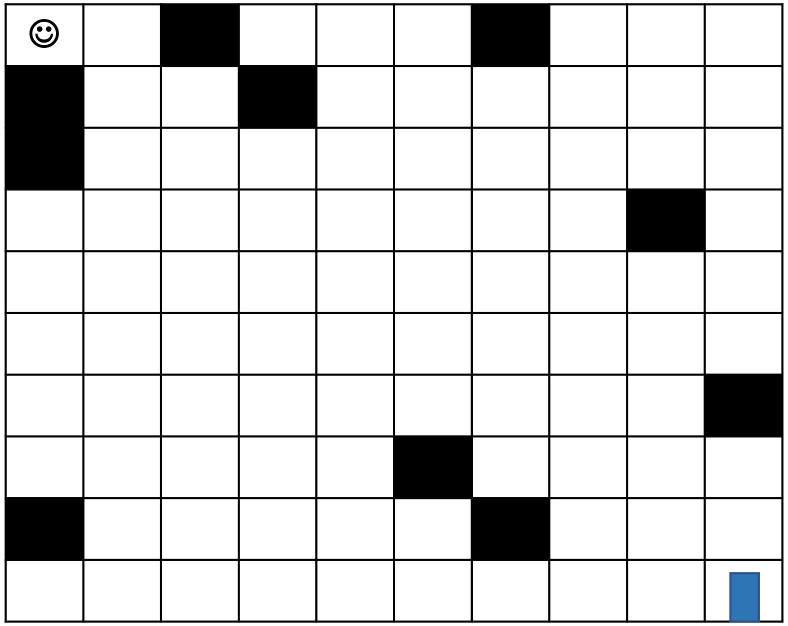
$$\alpha \ \cdot reward + \kappa\sqrt{moves} \ - \ \gamma\sqrt{numselected + numvisited}$$

  - ***reward***: environment reward
  - ***moves***: number of moves to reach state
  - ***numselected***: number of times state is selected in “Go” phase
  - ***numvisited***: number of times state is visited in “Explore” phase

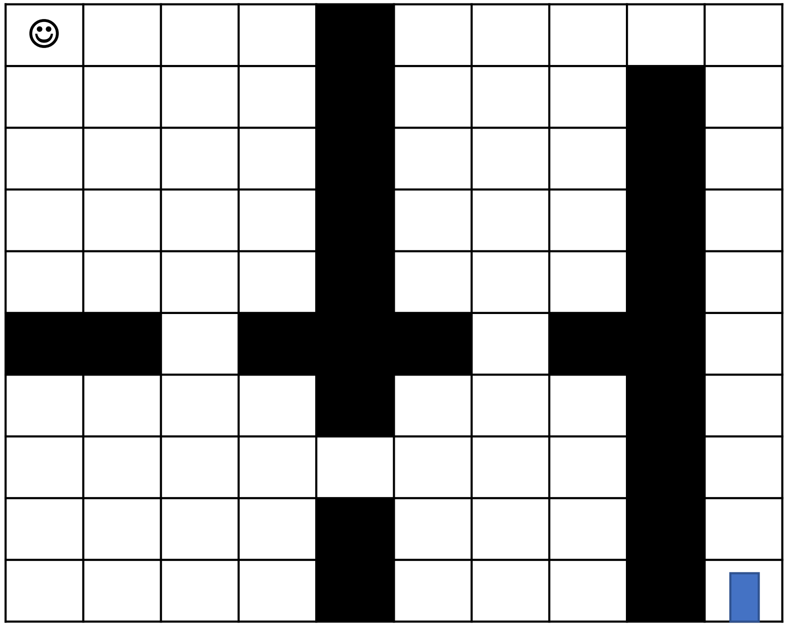
- Similar to Upper Confidence Bounds (UCB) equation and encourages greedy action selection in the long run

Environments

Unwalled Maze



Walled Maze



Legend:

□ Empty Grid

■ Obstacle

☺ Agent

🟦 Door

Memory Initialization and Updates

- Initialization. For each explored state, store in memory:
    1. Trajectory of actions to reach it
    2. No. of moves to reach it
    3. Reward
    4. No. of selections in the “Go” phase (initialized to 0)
    5. No. of visits in the “Explore” phase (initialized to 0)

- Updating Current State. Increment selection and visit counts upon selection in “Go” and visitation in “Explore” phase respectively

- Updating Next State. Update the memory of the next state if current trajectory:
    - Has a higher reward
    - Has the same reward with a shorter trajectory

Hippocampal Replay

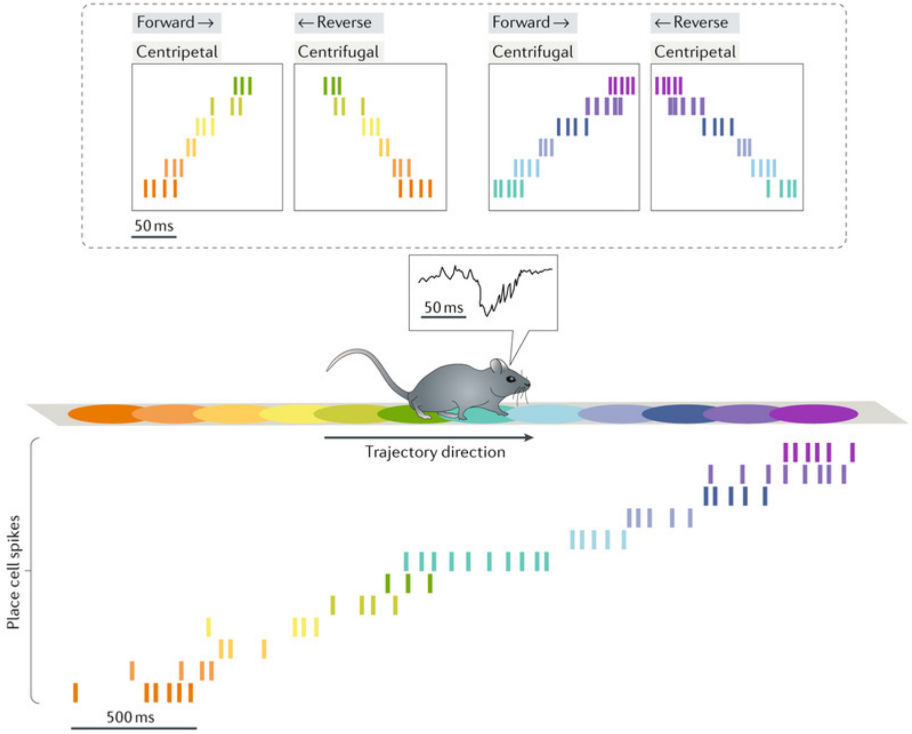
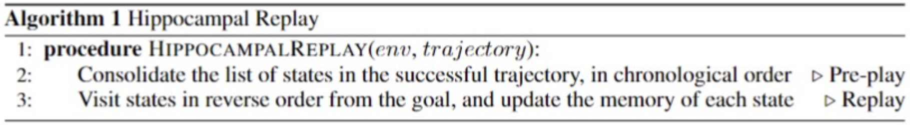


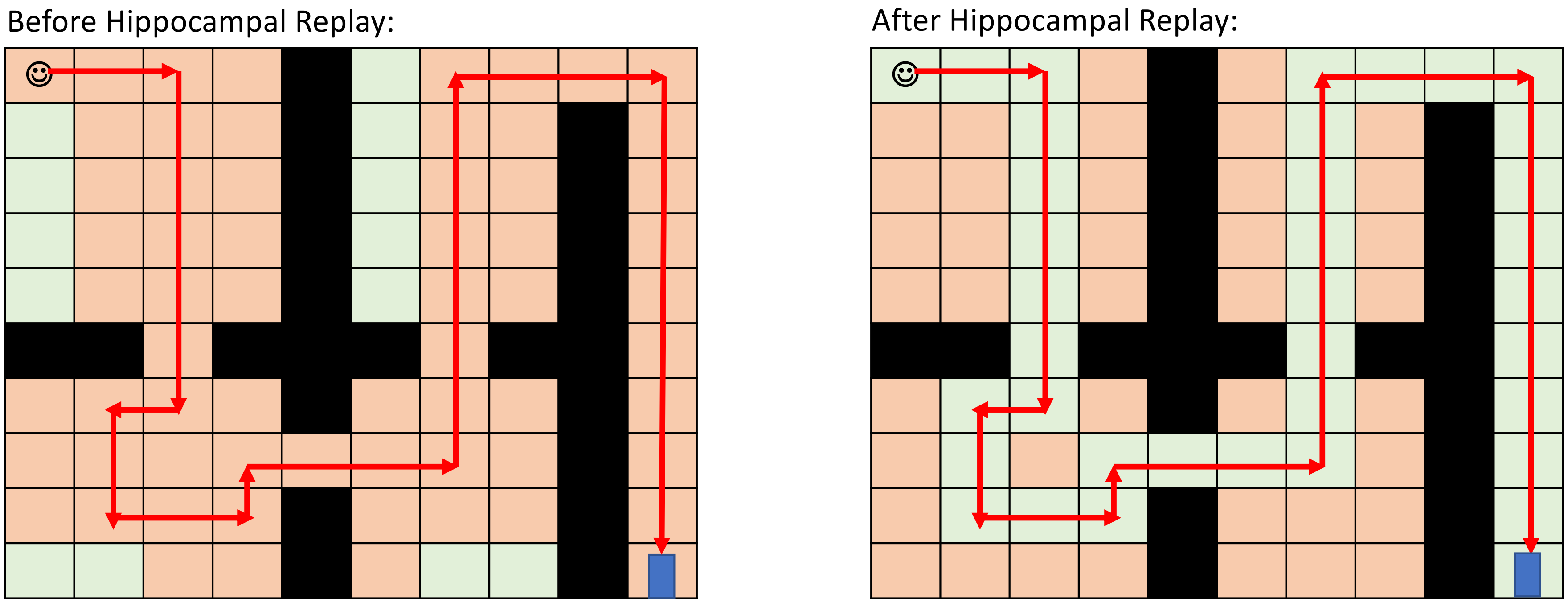
Figure extracted from Joo, H. R., & Frank, L. M. (2018). The hippocampal sharp wave-ripple in memory retrieval for immediate use and consolidation. Nature reviews. Neuroscience, 19(12), 744–757. https://doi.org/10.1038/s41583-018-0077-1



```
Algorithm 1 Hippocampal Replay
procedure HIPPOCAMPALREPLAY($env$, $trajectory$):
    Consolidate the list of states in the successful trajectory, in chronological order   ▷ Pre-play
    Visit states in reverse order from the goal, and update the memory of each state   ▷ Replay
```

- Inspired from sharp wave-ripple in mice
- Pre-play to **retrieve states** along successful trajectory
- Replay to **reset state selection and visit counts** to 0 to create an “Exploration Highway”
- Replay can also be used to update intrinsic reward of state for better performance

Hippocampal Replay creates Exploration Highway



Before Hippocampal Replay:

After Hippocampal Replay:

Legend:

🟧	Low Value States	⬛	Obstacle	🟦	Door
🟩	High Value States	☺	Agent	→	Discovered Path

Results

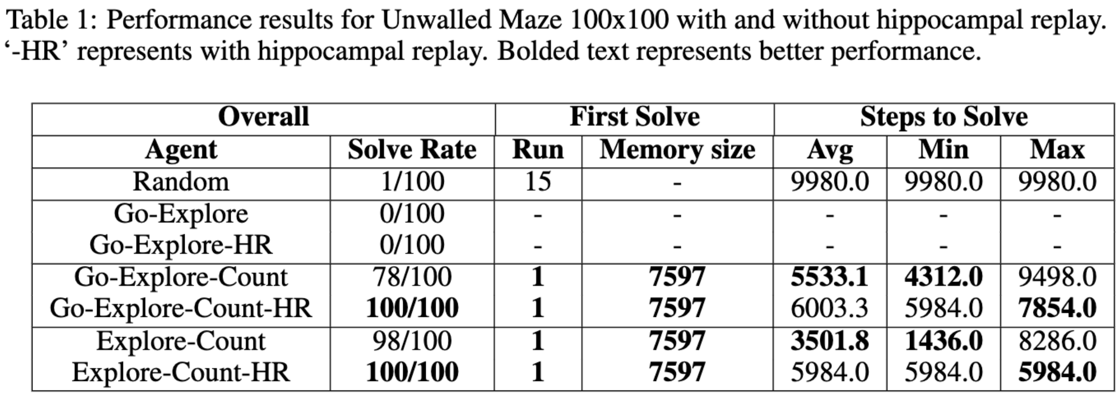


Table 1: Performance results for Unwalled Maze 100x100 with and without hippocampal replay. '-HR' represents with hippocampal replay. Bolded text represents better performance.

Overall		**First Solve**		**Steps to Solve**		
Agent	**Solve Rate**	**Run**	**Memory size**	**Avg**	**Min**	**Max**
Random	1/100	15	-	9980.0	9980.0	9980.0
Go-Explore	0/100	-	-	-	-	-
Go-Explore-HR	0/100	-	-	-	-	-
Go-Explore-Count	78/100	**1**	**7597**	**5533.1**	**4312.0**	9498.0
Go-Explore-Count-HR	**100/100**	**1**	**7597**	6003.3	5984.0	**7854.0**
Explore-Count	98/100	**1**	**7597**	**3501.8**	**1436.0**	8286.0
Explore-Count-HR	**100/100**	**1**	**7597**	5984.0	5984.0	**5984.0**

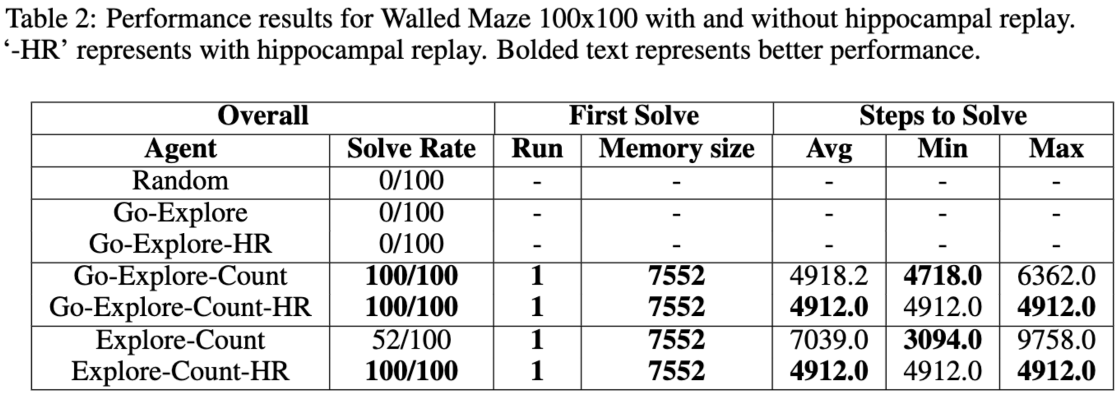


Table 2: Performance results for Walled Maze 100x100 with and without hippocampal replay. '-HR' represents with hippocampal replay. Bolded text represents better performance.

Overall		**First Solve**		**Steps to Solve**		
Agent	**Solve Rate**	**Run**	**Memory size**	**Avg**	**Min**	**Max**
Random	0/100	-	-	-	-	-
Go-Explore	0/100	-	-	-	-	-
Go-Explore-HR	0/100	-	-	-	-	-
Go-Explore-Count	**100/100**	**1**	**7552**	4918.2	**4718.0**	6362.0
Go-Explore-Count-HR	**100/100**	**1**	**7552**	**4912.0**	4912.0	**4912.0**
Explore-Count	52/100	**1**	**7552**	7039.0	**3094.0**	9758.0
Explore-Count-HR	**100/100**	**1**	**7552**	**4912.0**	4912.0	**4912.0**

- Our count-based approaches (Go-Explore-Count, Explore-Count) perform better than vanilla Go-Explore
- Hippocampal Replay leads to more **consistent performance** (higher solve rate) and **less exploration** (higher minimum number of steps to solve)

Hyperparameter Effects

Selection equation: $\alpha \ \cdot reward + \kappa\sqrt{moves} \ - \ \gamma\sqrt{numselected + numvisited}$

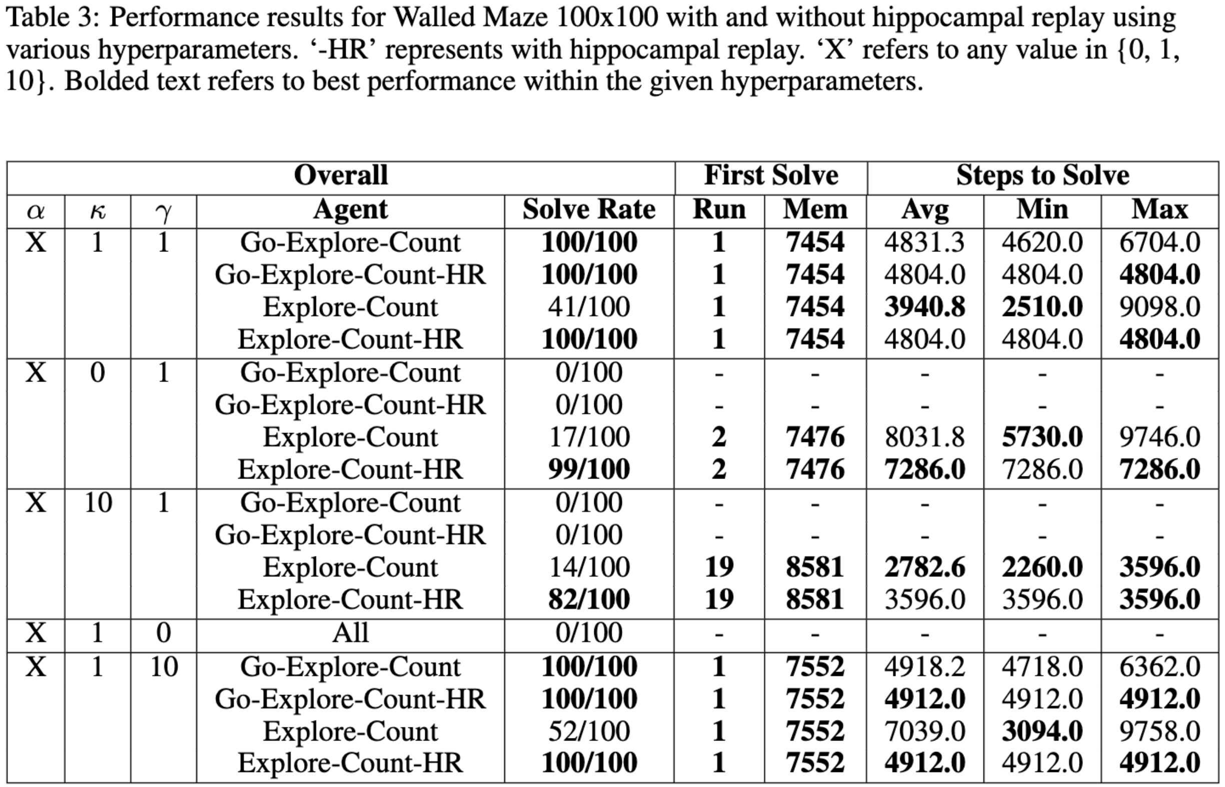


Table 3: Performance results for Walled Maze 100x100 with and without hippocampal replay using various hyperparameters. '-HR' represents with hippocampal replay. 'X' refers to any value in {0, 1, 10}. Bolded text refers to best performance within the given hyperparameters.

Overall					**First Solve**		**Steps to Solve**		
α	κ	γ	**Agent**	**Solve Rate**	**Run**	**Mem**	**Avg**	**Min**	**Max**
X	1	1	Go-Explore-Count	**100/100**	**1**	**7454**	4831.3	4620.0	6704.0
			Go-Explore-Count-HR	**100/100**	**1**	**7454**	4804.0	4804.0	**4804.0**
			Explore-Count	41/100	**1**	**7454**	**3940.8**	**2510.0**	9098.0
			Explore-Count-HR	**100/100**	**1**	**7454**	4804.0	4804.0	**4804.0**
X	0	1	Go-Explore-Count	0/100	-	-	-	-	-
			Go-Explore-Count-HR	0/100	-	-	-	-	-
			Explore-Count	17/100	**2**	**7476**	8031.8	**5730.0**	9746.0
			Explore-Count-HR	**99/100**	**2**	**7476**	**7286.0**	7286.0	**7286.0**
X	10	1	Go-Explore-Count	0/100	-	-	-	-	-
			Go-Explore-Count-HR	0/100	-	-	-	-	-
			Explore-Count	14/100	**19**	**8581**	**2782.6**	**2260.0**	**3596.0**
			Explore-Count-HR	**82/100**	**19**	**8581**	3596.0	3596.0	**3596.0**
X	1	0	All	0/100	-	-	-	-	-
X	1	10	Go-Explore-Count	**100/100**	**1**	**7552**	4918.2	4718.0	6362.0
			Go-Explore-Count-HR	**100/100**	**1**	**7552**	**4912.0**	4912.0	**4912.0**
			Explore-Count	52/100	**1**	**7552**	7039.0	**3094.0**	9758.0
			Explore-Count-HR	**100/100**	**1**	**7552**	**4912.0**	4912.0	**4912.0**

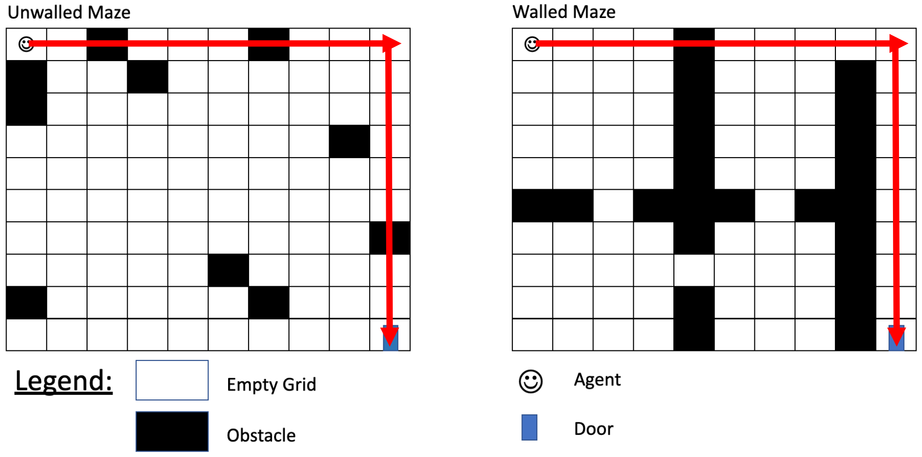
- Reward term not significant due to sparse reward setting

- Moves term should just be at 1 for tie-breaker to discover frontier. Any lower or higher leads to less efficient solution

- Higher exploration term (*numselected + numvisited*) leads to consistency at the expense of a less efficient solution

Goal-Directed Intrinsic Reward (GDIR)

- Possible to improve goal-seeking by adding a potential-function term to let agent know how close it is to goal
- Only serves as guide, does not tell us how to get there
- For maze environments, it can be Manhattan distance
- Scaled term between –1 to 0:
  - −(Manhattan Distance of agent to door position)/(Height+Width − 2)



Results for GDIR

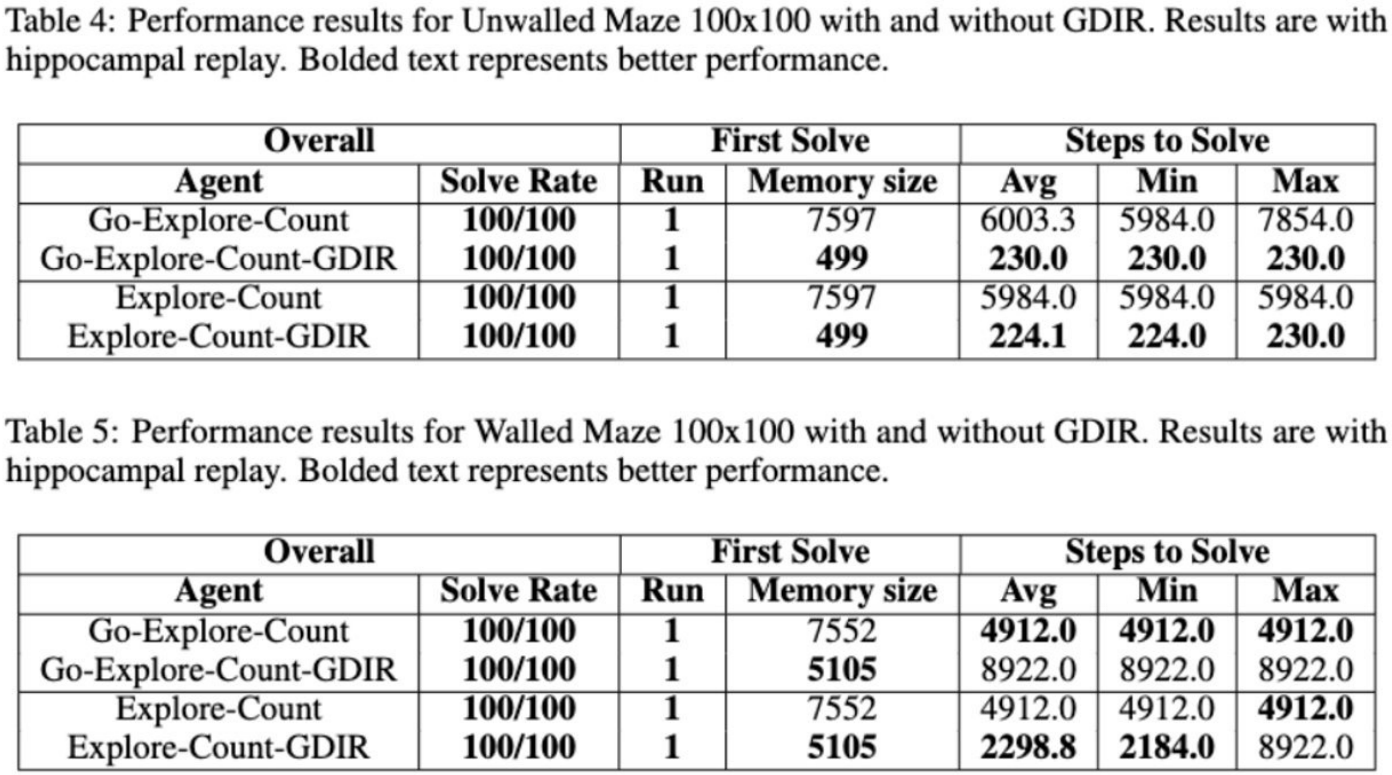


Table 4: Performance results for Unwalled Maze 100x100 with and without GDIR. Results are with hippocampal replay. Bolded text represents better performance.

Overall		**First Solve**		**Steps to Solve**		
Agent	**Solve Rate**	**Run**	**Memory size**	**Avg**	**Min**	**Max**
Go-Explore-Count	**100/100**	**1**	7597	6003.3	5984.0	7854.0
Go-Explore-Count-GDIR	**100/100**	**1**	**499**	**230.0**	**230.0**	**230.0**
Explore-Count	**100/100**	**1**	7597	5984.0	5984.0	5984.0
Explore-Count-GDIR	**100/100**	**1**	**499**	**224.1**	**224.0**	**230.0**

Table 5: Performance results for Walled Maze 100x100 with and without GDIR. Results are with hippocampal replay. Bolded text represents better performance.

Overall		**First Solve**		**Steps to Solve**		
Agent	**Solve Rate**	**Run**	**Memory size**	**Avg**	**Min**	**Max**
Go-Explore-Count	**100/100**	**1**	7552	**4912.0**	**4912.0**	**4912.0**
Go-Explore-Count-GDIR	**100/100**	**1**	**5105**	8922.0	8922.0	8922.0
Explore-Count	**100/100**	**1**	7552	4912.0	4912.0	**4912.0**
Explore-Count-GDIR	**100/100**	**1**	**5105**	**2298.8**	**2184.0**	8922.0

- GDIR leads to shorter solutions in most situations
- Walled Maze Go-Explore-Count-GDIR ends up with a longer solution – GDIR guidance may lead to wrong path followed initially in presence of wall

Discussion

- Should count-based explore-exploit selection be used as the baseline policy, or just as the action selection branch?

- How to incorporate with neural networks to help with action selection?

- How does our brain generate goals for an agent to move towards? How are these goals then quantified with an intrinsic reward?