Selective Memory Replay Improves Exploration in a Spiking Wavefront Planner

Harrison Espino[1] and Robert Bain[2] and Jeffrey L. Krichmar[1,2]

***Abstract*—Spiking wavefront planners for navigation demonstrate biologically plausible behavior when exploring and planning paths through an environment. Not present in these models, however, is the replay of previous experiences observed in hippocampal sharp wave ripple complexes (SWRs) during sleep and wake resting states. This work implements a memory replay algorithm in a spiking wavefront model, and investigates different theories of replay selection. Results indicate that the addition of replay in the spiking wavefront model improves the speed at which the agent learns the environment, and the ability to adapt to change. Furthermore, selection of replays based on its effectiveness in updating model weights leads to greater improvement when compared to a uniformly weighted selection.**

***Index Terms*—Hippocampus, Navigation, Planning, Replay, Spiking Network**

I. Introduction

Studies of the hippocampus in rodents during navigation tasks have observed firing of place cells corresponding to its location in an environment [1]. When determining where to navigate in a maze, rodents exhibit successive activation in place cells corresponding to future trajectories, suggesting place cells have a role in path planning [2]. These characteristics of rodent navigation have inspired biologically plausible spiking neural network models known as "Spiking Wavefront Planners", which plan paths by propagating signals in a similar manner [3], [4].

In addition to the described place cell activity during navigation, successive reactivation of place cells in the same pattern as paths taken while awake were found during slow-wave sleep. This is believed to have a role in memory consolidation and improving learning [5]. Currently, spiking wavefront planner models do not exhibit this behavior. In this work, we modify the model to replay memories of past trajectories in a "sleep state" after a number of "awake" trials are performed by a simulated agent. These agents are tasked with exploring two environments, a Tolman detour maze [6] and a Dyna maze [7], as well as reaching their respective reward spaces when obstacles are put in place.

Another important aspect of memory replay is how the selection of paths in memory is guided. One theory proposes that memories are accessed to optimize the reward gained should the replay be used to update a model [7]. In a reinforcement learning framework, this can be quantified by the product of a "gain" term which measures the improvement of the policy if the replay is applied, and a "need" term which measures the likelihood that this replayed state will occur in the future. Another theoretical model prioritizes the replay of similar experiences [8].

For our model we take an alternative approach, where memory selection is weighted by the product of two terms. The first is a "loss" corresponding to the difference between the perceived cost of the environment and the current model weights. The second is the "eligibility trace" which is used in the update step of the model to determine which neurons are eligible to be updated.

II. Methods

A. Spiking Wavefront Planner

Here we briefly describe the spiking wavefront planner model. For more details, see [4], [9].

The spiking wavefront propagation algorithm assumes a grid representation of space, where connections between units represent the ability to travel from one grid location to a neighboring location. Each unit in the grid is represented by a simplified integrate and fire neuron. The activity of neuron i at time $t+1$ is represented by (1):

$$v_i(t+1) = u_i(t) + I_i(t+1) \tag{1}$$

in which $u_i(t)$ is the recovery variable and $I_i(t)$ is the input current at time t.

The recovery variable $u_i(t+1)$ is described by:

$$u_i(t+1) = \begin{cases} -5 & if\ v_i(t) = 1 \\ min(u_i(t)+1, 0) & otherwise \end{cases} \tag{2}$$

such that immediately after a membrane potential spike, the recovery variable starts as a negative value and linearly increases toward a baseline value of 0.

The input current I at time $t+1$ is given by:

$$I_i(t+1) = \sum_{j=1}^{N} \begin{cases} 1 & if\ d_{ij}(t) = 1 \\ 0 & otherwise \end{cases} \tag{3}$$

such that $d_{ij}(t)$ postpones the integration of input, I, from neighboring neuron j to neuron i. This delay is given by:

$$d_{ij}(t+1) = \begin{cases} D_{ij} & if\ v_j(t) \geq 1 \\ max(d_{ij}(t)-1, 0) & otherwise \end{cases} \tag{4}$$

[1]Harrison Espino is with the Department of Computer Science, University of California, Irvine, Irvine, CA, USA espinoh@uci.edu

[2]Robert Bain is with the Department of Cognitive Sciences, University of California, Irvine, Irvine, CA, USA rkbain@uci.edu

[1,2]Jeffrey Krichmar is with the Department of Cognitive Sciences and Department of Computer Science, University of California, Irvine, Irvine, CA, USA jkrichma@uci.edu

The value of $D_{ij}(t)$ is the propagation delay between neurons i and j, and denotes the expected cost of traveling between from location i to j. Cost is an open parameter, which could depend on a number of variables. In the present paper, cost represents how easy it is to move in a given direction. A low cost could be an open traversable region of space, and a high cost could signify the presence of a barrier.

B. E-Prop

The E-Prop learning rule was developed to learn sequences in spiking neural networks [10]. In the present path planning algorithm, E-Prop was used to learn a map of the environment and to plan paths based on that map knowledge. The active neurons after a wave propagation are eligible to update. An eligibility trace based on time elapsed since the wave reaches the goal destination, dictates the eligibility.

E-Prop is applied to weights projecting from neurons along the calculated path. It is assumed that the agent can observe the features (e.g., traversal cost) at map locations adjacent to the path, In this way, E-Prop solves the credit assignment problem by rewarding paths that lead to goal locations, while also learning about environmental structure.

The E-Prop algorithm is applied to weights of the spiking neural network by:

$$D_{ij}(t+1) = D_{ij}(t) + \delta(e_i(t)(map_{xy} - D_{ij}(t)) \quad (5)$$

where δ is the learning rate, set to 0.1, $e_i(t)$ is the eligibility trace for neuron i, and map_{xy} represents the observed cost for traversing the location (x, y), which corresponds to neuron i. This rule is applied for each of the neighboring neurons, j, of neuron i. The loss in Eqn. 5 is map_{xy} - D_{ij}.

The eligibility trace for neuron i is given by (6):

$$e_i(t+1) = \begin{cases} 1 & if\ v_j(t) \geq 1 \\ e_i(t) - \frac{e_i(t)}{\tau} & otherwise \end{cases} \quad (6)$$

where τ is the rate of decay for the eligibility trace, set to 25.

To determine a path from an agent's location to a desired space, a signal is sent originating from the neuron corresponding to the agent's location. This signal propagates to the origin neuron's neighbors, delayed by the associated weight of the origin's outgoing connection. This is repeated for each of these neurons and their neighbors until a signal reaches the neuron corresponding to the destination for the first time. The origin of this signal is recursively traced backwards until the first neuron is reached again. This sequence of neurons represents the path of least cost given the agent's information about the environment. Details on the original spiking wavefront propagation planner can be found in [9].

C. Tolman Maze

We simulate a Tolman detour maze, used previously in a number of studies involving place cells and hippocampal replay [6], [11]. For the detour task, the agent is placed at the start location (S in Fig. 1), and must reach the end location (E in Fig. 1). During the task, two barriers may be placed forcing the agent to take a shortcut (P1 and P2 in Fig. 1). Placing the barrier at P1 should cause the agent to choose the shorter detour on the left, while placing the barrier at P2 should result in the agent taking longer detour on the right. Barriers and areas outside the maze have an environmental cost of 50, and areas within the maze have an environmental cost of 1.

In an initial exploration phase, no barriers are placed and the agent is free to familiarize themselves with the environment.

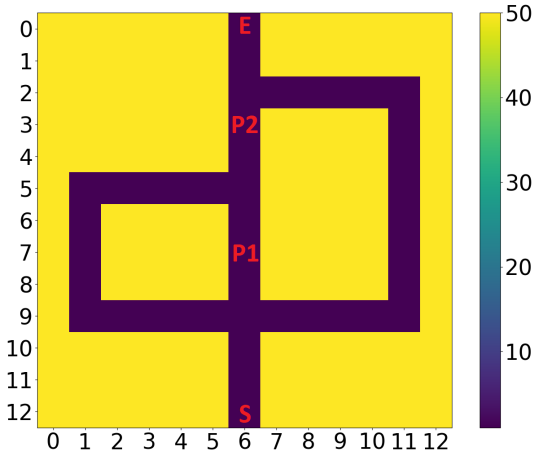


Fig. 1. Simulated Tolman detour maze. Start location is marked with "S". End location is marked with "E". Barriers P1 and P2 are marked as such. Yellow regions had a traversal cost of 50 and the blue region had a cost of 1.

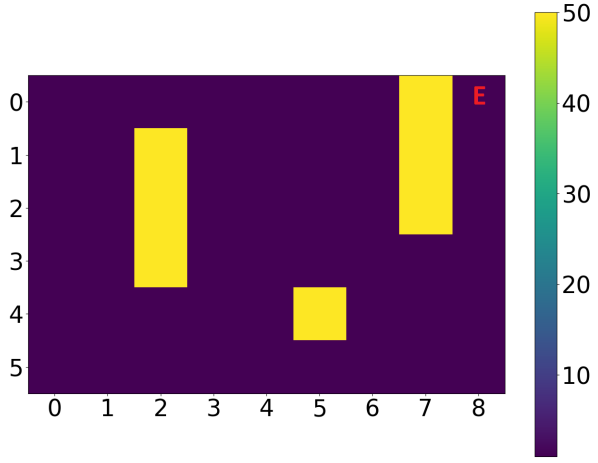


Fig. 2. Simulated Dyna maze. End location is marked with "E". Yellow regions had a traversal cost of 50 and the blue region had a cost of 1.

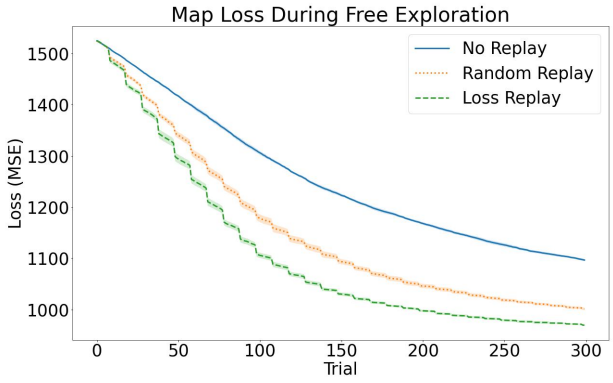


Fig. 3. Total loss for all neurons over time for each replay type in the Tolman detour maze. Agents with no replay in solid blue. Agents with random replay selection in dotted orange. Agents with loss-based replay in dashed green. Lines are the mean of 10 trials, shaded areas represent one standard error.

To explore, the agent uses the spiking wavefront planner to travel to random locations in the environment, updating the model weights after each path. After this exploration phase, the agent will plan a straight path from "S" to "E" when "E" is the goal destination.

Following the exploration phase, one of two barriers are placed and the agent must continually attempt to reach the end environment from the start. A trial is successful if the planned path successfully reaches the end state, otherwise it is a failure. After each trial, agents with replay will select paths from memory to update their weights.

D. Dyna Maze

We also simulate the Dyna maze, introduced by Sutton to evaluate the Dyna-Q reinforcement learning algorithm and later used to investigate methods for memory access [7], [12]. The agent starts at a random location along the west wall of the maze and must reach the end location at the northeast corner of the maze (see Fig. 2). Similar to the Tolman maze, the barriers had a traversal cost of 50 and the other regions had a traversal cost of 1.

In the period of initial exploration, barriers are removed. For the goal task, the agent will attempt to plan a path from its starting location to the end. If a barrier is encountered, the agent will update its weights from this information and update its path using the spiking wavefront planner again. This re-planning continues as barriers are encountered until the end is successfully reached. After each of these trials, agents with replay will select paths from memory to update their weights.

E. Replay

In a replay sequence, the agent will select a number of paths and their corresponding eligibility traces from memory stored during the exploration phase. The same path may be chosen more than once in a single replay sequence. The selected memory is used to update the model weights using the most recently sensed values from a path traversal. Two different

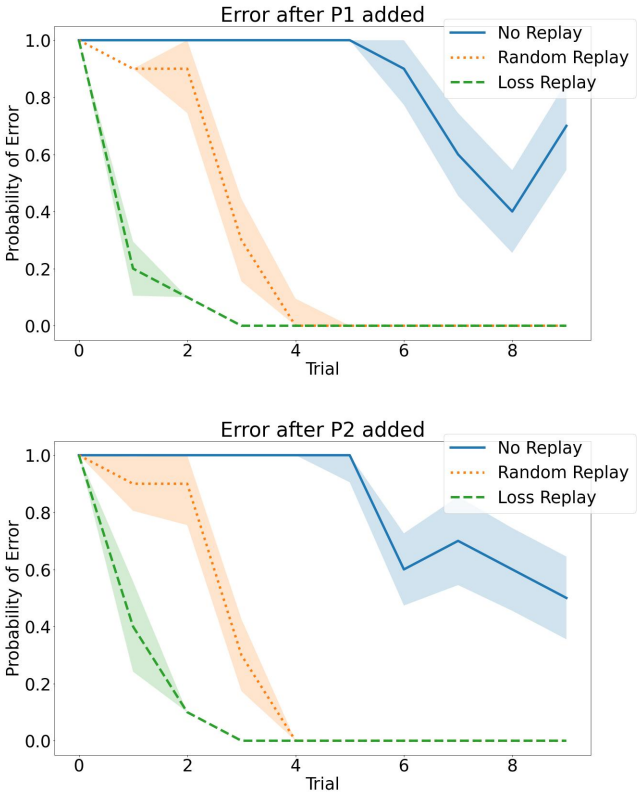


Fig. 4. Probability of failure for agents reaching the goal after the P1 (Top) or P2 (Bottom) barrier is placed in the Tolman detour maze. Lines are the mean of 10 trials, shaded areas represent one standard error.

memory selection criteria were investigated: uniform selection and loss-based selection. For uniform selection, each path in memory had an equal probability of being replayed.

For loss-based selection, selection was weighted such that paths which covered areas that have the greatest disparity between the perceived cost and the model weights have a higher likelihood of being replayed. To determine this, a metric is calculated for each path. First, the average incoming weight value is calculated for each neuron as in (7), where N is the set of neighboring neurons and $d_{n,i}$ is the delay from neuron n to neuron i.

$$D_{avgi} = \frac{\sum_{n \in N} d_{n,i}}{len(N)} \tag{7}$$

$$l_i = (c_{x,y} - D_{avgi})^2 \tag{8}$$

The loss for each neuron (denoted l_i) is calculated as the squared difference between the sensed cost of traversal at the associated area $c_{x,y}$ and the average weight value, as in (8). Finally, the score for each path is calculated by summing over each point in the path P the product of the eligibility trace and the loss.

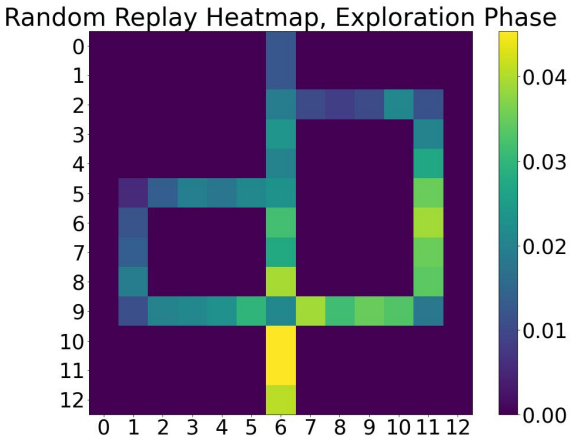


Fig. 5. Heatmap showing the frequency of each location based on the paths selected with a uniformly selected replay agent during the exploration phase.

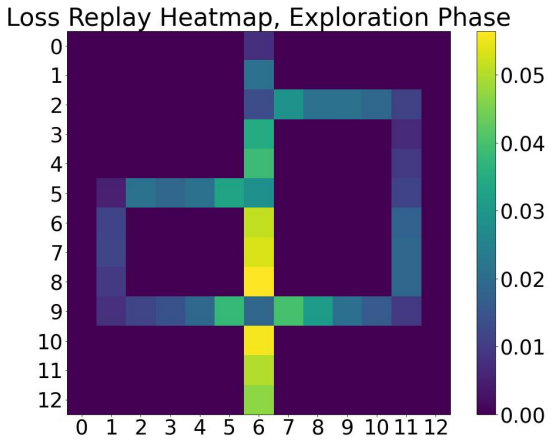


Fig. 6. Heatmap showing the frequency of each location based on the paths selected with a loss-based replay agent during the exploration phase.

$$S_p = \sum_{p \in P} e_p * l_p \tag{9}$$

The softmax of these scores is taken to determine the probability of selection for each path. Each memory score is also subject to exponential decay based on the number of times it was accessed in the past.

III. Results

A. Tolman Maze

In the exploration phase of the Tolman maze, agents explored for a total of 300 trials. For each trial, the agent planned and navigated a path to a random location using the spiking wavefront planner. Model weights were updated after each path using the sensed cost of traversal. Every 10 trials, agents would be removed from the environment and would be allowed to replay 10 paths from memory before beginning again at the start location. The total loss between all neuron weights and the true environmental cost of their corresponding locations was evaluated at every trial, after replay if it would occur.

Fig. 3 shows the mean loss of 10 agents for each replay method. Agents which used replay of some form learned the environment faster than agents which did not use replay. Additionally, loss-based replay performed better than uniformly selected replay.

Agents that have explored the environment were then tasked with reaching the end location with the either the P1 or P2 barrier placed. Fig. 4 shows the average probability of failure for 10 agents during these test trials with the P1 and P2 barrier. In both cases, some agents which did not use replay were unable to plan successful paths after 10 trials. All agents which used replay recognized the need to take the correct detour after 4 trials. Loss-based replay adapted faster than uniformly selected replay.

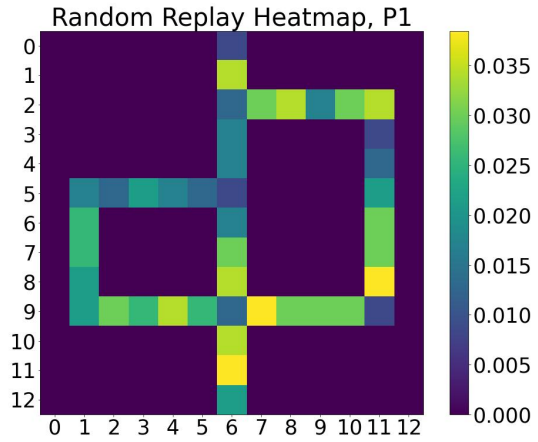
The experiences that were replayed had an effect on agent behavior. To examine the replays chosen for selection in the uniform and loss-based agents, we generate a heatmap illustrating the frequency of locations among the replayed paths during the exploration phase (Fig. 5 and Fig. 6) and the goal task (Fig. 7 and Fig. 8). In the exploration phase, the frequency of locations were distributed such that no single location was dominantly present in either agent's chosen replays. This stayed true for random agents during the goal task. In contrast, replays selected by the loss-based agent centered strongly around the placed barrier, depending on the agent's trial condition. These results show that the loss based replay leads to examination of areas with the highest uncertainty to the agent.

B. Dyna Maze

In the exploration phase of the Dyna maze, agents explored for a total of 350 trials. Agents navigated randomly and were removed every 10 paths taken to replay as in the Tolman detour maze. For each trial, agents began the next 10 trials at a random start location. This was to replicate methods used in previous work [7].

For the initial exploration phase, the environment did not have any barriers and the cost of traversal was the same for all locations. As in the Tolman detour maze, the loss-based replay agents learned fastest, followed by the uniformly selected replay agents, with the agents without replay learning the slowest (Fig. 9).

Next, the barriers were placed in the environment and the agents were tasked with reaching the goal as previously described (see yellow regions in Fig 2). Because replay only occurs after trials, agents performed similarly during the first trial. Agents that were able to replay after the first trial were more likely to reach the goal without hitting a barrier. This results in the agents with replay reaching the goal in fewer



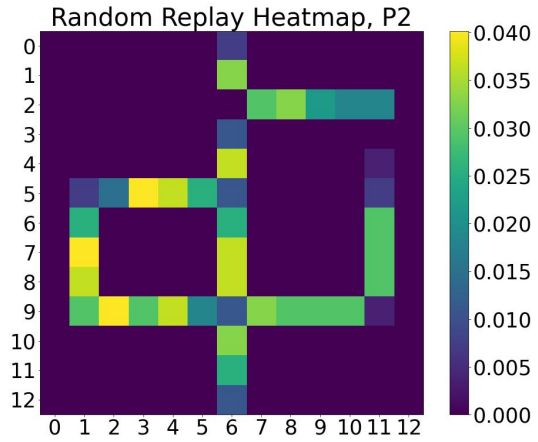
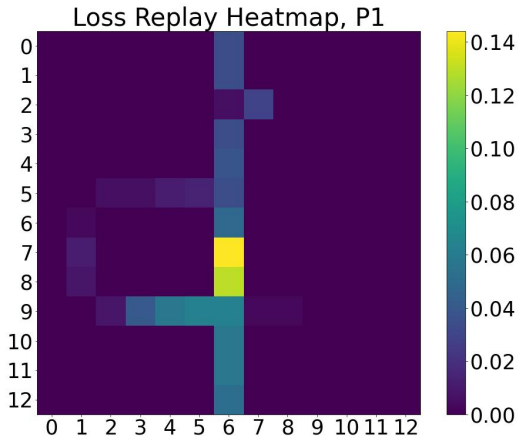


Fig. 7. Heatmap showing the frequency of each location based on the paths selected with a uniformly selected replay agent when the P1 (Top) or P2 (Bottom) barrier is placed in the Tolman detour maze.



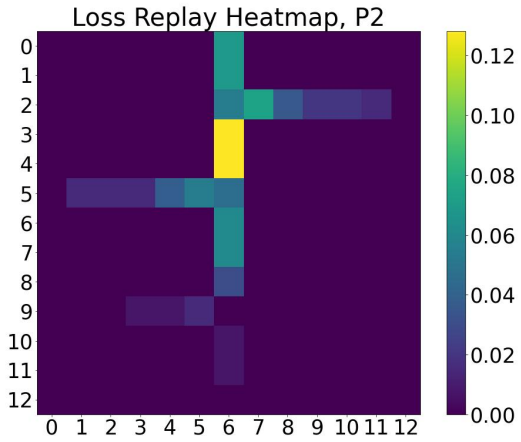


Fig. 8. Heatmap showing the frequency of each location based on the paths selected with a loss-based replay agent when the P1 (Top) or P2 (Bottom) barrier is placed in the Tolman detour maze.

steps, with the loss-based replay performing slightly better than the uniformly selected replay as in Fig. 10.

We also analyze the ability for the agent to learn the environment when the objective is to reach an end location rather than explore. Fig. 11 shows the loss of the spiking wavefront planner for each of the agents after barriers have been placed. Replay methods were able to learn changes in the environment faster, even when the task did not necessitate exploring novel areas.

Similar to the Tolman detour maze simulations, the replays for loss-based replay agent focused on areas of high uncertainty. The heatmap showing frequency of locations in the replayed paths is shown in Fig. 12. In the loss-based agent, locations on or around barriers were replayed with slightly higher frequency. Frequently replayed locations in the uniform selection agent did not seem to have any preference towards barrier locations.

IV. Discussion

The addition of memory replay to the spiking wavefront planner caused a dramatic increase in the speed at which an agent was able to learn an unknown environment and recover from change. When there was a change in an environment, replay agents adapted faster and performed goal-reaching tasks with greater accuracy and efficiency than agents which did not. Because the weights are updated using a previous path and the most recent experience with the environment, it can be said that the agent is revisiting old experiences with new information. The results also suggest that replaying experiences based on a selection criteria, rather than randomly can result in a greater benefit to the agent.

Neuroscience studies have varying observations regarding the content of replay in the hippocampus. Some observe place cell activation in similar patterns as past experiences, or resembling a trajectory toward home or goal locations [13], [14]. Others have found that replayed paths did not resemble past experiences at all, but rather random trajectories similar

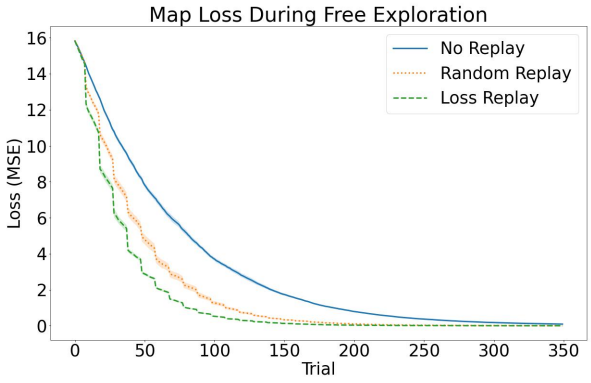


Fig. 9. Total loss for all neurons over time for each replay type in the Dyna maze. Agents with no replay in solid blue. Agents with random replay selection in dotted orange. Agents with loss-based replay in dashed green. Lines are the mean of 10 trials, shaded areas represent one standard error.

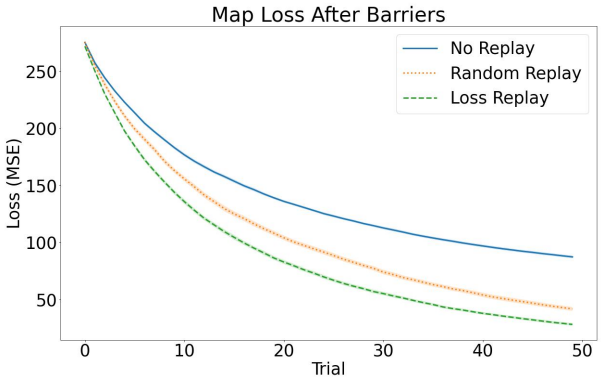


Fig. 11. Total loss for all neurons over time for each agent in the Dyna maze after the barriers have been placed. Lines are the mean of 10 trials, shaded areas represent one standard error.

to Brownian motion [15]. Our work supports the former, by showing that prioritizing replays based on novelty (i.e., loss) improves exploration and memory adaptation.

Studies of neuron behavior during replay find that the amount of time each place neuron is active is not always equal [16]. Activation of place cells during replay sometimes appear to "hover" at specific locations in the replayed path. This becomes more common as the rodent is more familiar with the environment, suggesting that the role of replay may shift from rapidly acquiring a rough model to honing in on specific details of an environment.

We find a similar shift in replay objective in our loss-based replay to that observed by [16]. Fig. 6 shows little to no preference for replayed locations while the agent is still exploring the environment. Once the agent is familiar with an environment, we see a preference for areas requiring further exploration in the overall choice of replays selected as seen in Fig 8 for the Tolman detour maze and also in Fig 12. We predict that replay experiences may be selected similarly in rodents.

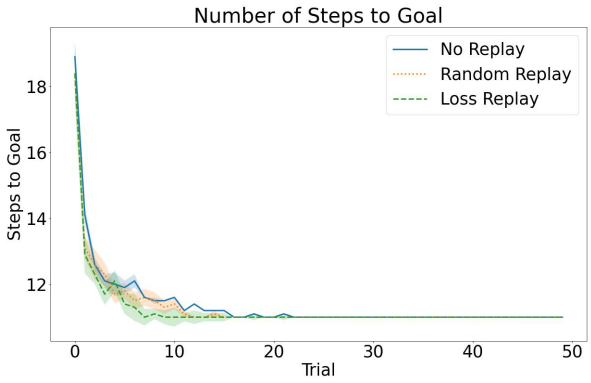


Fig. 10. Number of steps to reach the end location. Lines are the mean of 10 trials, shaded areas represent one standard error.

The proposed method of scoring memories based on the eligibility trace and the loss share many similarities to the theory proposed by Mattar and Daw [7]. The value of the eligibility trace and the gain term quantify the increase in expected reward. In our case, high reward would represent a large change in model weights towards the true environmental value. The loss and the need term quantify the frequency of reward given at a state or location. Since fast exploration prioritizes unexplored locations, we should expect it to be increasingly less rewarding to visit locations which have been traversed multiple times.

Finally, work related to rodent studies of hippocampal replay have shown that experience is not the only factor in contributing to the selection of replays [17]. The current work has shown that experience may be an important heuristic in the selection of replays, but a combination of many qualities may result in a greater improvement. Also not present in the current work are "backwards" replay of paths as well as the activation of novel sequences of neurons. Evidence of both has been recorded in rat hippocampal place cells during replay [14], [18].

Replay buffers are commonly used in Deep Reinforcement learning [19]. However, they have not been applied as much to robot navigation and other robot tasks. Adding replay to these control policies could lead to performance improvements in terms of learning speed and adaptability.

The BADGR robot had similar goals to the present work in that it learned the cost of traversing varied environments [20]. However, instead of using replay, it used deep learning and an LSTM to learn which features predicted passable trajectories. Unlike our algorithm, training was offline and required intensive computations. The algorithm proposed here learns continually and online. In general, it is lightweight and learns rapidly. Furthermore, since it is a spiking neural network it can be deployed on power efficient neuromorphic hardware as was demonstrated in [21].

In traditional robotics, Simultaneous Localization And Mapping (SLAM) and path planning are handled separately [22].

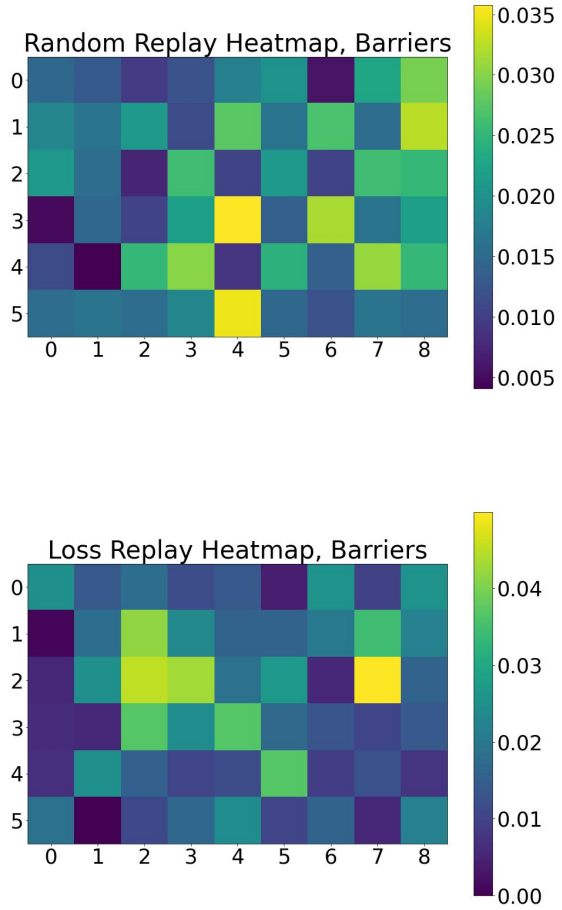


Fig. 12. Replayed areas of the Dyna maze. Heatmap showing the frequency of each location based on the paths selected with a uniformly selected replay agent (Top) and a loss-based agent (Bottom) when barriers are placed.

SLAM creates the map and can update the map as the robot explores. SLAM makes a prediction of where the robot is currently located. If a goal location is specified, a separate path planner like A* or Djikstra's algorithm or rapidly exploring random trees (RRTs) is applied to the SLAM map [23]. In contrast, our model combines these two steps by using the spikewave to generate the path and map the environment through E-Prop and replay.

The neurobiologically inspired RatSLAM also combines SLAM with path planning by maintaining the map information in a continuous attractor network and path information in an experience map [24]. However, they still apply an algorithm like Djikstra's to plan a path between two locations. It would be interesting to see if applying our replay method to the experience map could improve performance.

In conclusion, we introduce replay methods to a spiking wavefront propagation algorithm. In particular, the loss-based replay shows superior performance, in terms of learning speed and ability to adapt to change, over no replay and a uniformly distributed replay. The algorithm was tested on mazes that have been used in neuroscience experiments and to demonstrate machine learning algorithms. The replays selected by the loss algorithm are similar to that observed in recent rodent experiments [16]. The algorithm could also have applications for robot navigation and machine learning.

REFERENCES

[1] D. Derdikman, and E.I. Moser, A manifold of spatial maps in the brain. Trends Cogn Sci 14 (2010) 561-9.

[2] A. D. Redish, "Vicarious trial and error," Nature Reviews Neuroscience, vol. 17, no. 3, pp. 147–159, Feb. 2016.

[3] S. Koul and T. K. Horiuchi, "Path planning by spike propagation," 2015 IEEE Biomedical Circuits and Systems Conference (BioCAS), Atlanta, GA, USA, 2015, pp. 1-4, doi: 10.1109/BioCAS.2015.7348409.

[4] J. L. Krichmar, N. A. Ketz, P. K. Pilly, and A. Soltoggio, "Flexible path planning through vicarious trial and error," bioRxiv, Oct. 2021.

[5] A. K. Lee and M. A. Wilson, "Memory of sequential experience in the hippocampus during slow wave sleep," Neuron, vol. 36, no. 6, pp. 1183–1194, Dec. 2002.

[6] A. Alvernhe, E. Save, and B. Poucet, "Local remapping of place cell firing in the Tolman detour task," European Journal of Neuroscience, vol. 33, no. 9, pp. 1696–1705, Mar. 2011.

[7] M. G. Mattar and N. D. Daw, "Prioritized memory access explains planning and hippocampal replay," Nature Neuroscience, vol. 21, no. 11, pp. 1609–1617, Jan. 2018.

[8] Saxena, R., Shobe, J.L., and McNaughton, B.L. (2022). Learning in deep neural networks and brains with similarity-weighted interleaved learning. P Natl Acad Sci USA 119. ARTN e2115229119 10.1073/pnas.2115229119.

[9] Hwu, T., Wang, A.Y., Oros, N., Krichmar, J.L.: Adaptive robot path planning using a spiking neuron algorithm with axonal delays. Ieee Transactions on Cognitive and Developmental Systems 10(2), 126–137 (2018).

[10] Bellec, G., Scherr, F., Subramoney, A. et al. A solution to the learning dilemma for recurrent networks of spiking neurons. Nat Commun 11, 3625 (2020). https://doi.org/10.1038/s41467-020-17236-y.

[11] K. L. Stachenfeld, M. M. Botvinick, and S. J. Gershman, "The hippocampus as a predictive map," Nature Neuroscience, vol. 20, no. 11, pp. 1643–1653, Oct. 2017.

[12] R. S. Sutton, "Integrated Architectures for learning, planning, and reacting based on approximating dynamic programming," Machine Learning Proceedings 1990, pp. 216–224, Jun. 1990.

[13] M. P. Karlsson and L. M. Frank, "Awake replay of remote experiences in the hippocampus," Nature Neuroscience, vol. 12, no. 7, pp. 913–918, Jun. 2009.

[14] J. Widloski and D. J. Foster, "Flexible rerouting of hippocampal replay sequences around changing barriers in the absence of global place field remapping," Neuron, vol. 110, no. 9, May 2022.

[15] F. Stella, P. Baracskay, J. O'Neill, and J. Csicsvari, "Hippocampal reactivation of random trajectories resembling Brownian diffusion," Neuron, vol. 102, no. 2, Apr. 2019.

[16] A. Berners-Lee, T. Feng, D. Silva, X. Wu, E. R. Ambrose, B. E. Pfeiffer, and D. J. Foster, "Hippocampal replays appear after a single experience and incorporate greater detail with more experience," Neuron, vol. 110, no. 11, Jun. 2022.

[17] A. S. Gupta, M. A. A. van der Meer, D. S. Touretzky, and A. D. Redish, "Hippocampal replay is not a simple function of experience," Neuron, vol. 65, no. 5, pp. 695–705, Mar. 2010.

[18] D. J. Foster and M. A. Wilson, "Reverse replay of behavioural sequences in hippocampal place cells during the awake state," Nature, vol. 440, no. 7084, pp. 680–683, Feb. 2006.

[19] Mnih, V., Kavukcuoglu, K., Silver, D. et al. Human-level control through deep reinforcement learning. Nature 518, 529–533 (2015). https://doi.org/10.1038/nature14236.

[20] Kahn, G., Abbeel, P., Levine, S.: Badgr: An autonomous self-supervised learning-based navigation system. arXiv:2002.05700 [cs.RO] (2020).

[21] K. D. Fischl, K. Fair, W. -Y. Tsai, J. Sampson and A. Andreou, "Path planning on the TrueNorth neurosynaptic system," 2017 IEEE International Symposium on Circuits and Systems (ISCAS), Baltimore, MD, USA, 2017, pp. 1-4, doi: 10.1109/ISCAS.2017.8050932.

[22] H. Choset, K.M. Lynch, S. Hutchinson, G.A. Kantor, W. Burgard, L.E. Kavraki, and S. Thrun, Principles of Robot Motion: Theory, Algorithms, and Implementations, MIT Press. A Bradford Book., Cambridge, MA, 2005.
[23] LaValle, S.M. (2011). Motion Planning Part I: The Essentials. Ieee Robot Autom Mag 18, 79-89. 10.1109/Mra.2011.940276.
[24] M. Milford, A. Jacobson, Z. Chen, and G. Wyeth, RatSLAM: Using Models of Rodent Hippocampus for Robot Navigation and Beyond. Springer Trac Adv Ro 114 (2016) 467-485.