Benefits of Combining Dimensional Attention and Working Memory for Partially Observable Reinforcement Learning Problems

Ngozi Omatu
Middle Tennessee State University
Murfreesboro, Tennessee, USA
nco2f@mtmail.mtsu.edu

Joshua L. Phillips
Middle Tennessee State University
Murfreesboro, Tennessee, USA
Joshua.Phillips@mtsu.edu

ABSTRACT

Neuroscience provides a rich source of inspiration for new types of algorithms and architectures to employ when building AI and the resulting biologically-plausible approaches that provide formal, testable models of brain function. The working memory toolkit (WMtk), was developed to assist the integration of an artificial neural network (ANN)-based computational neuroscience model of working memory into reinforcement learning (RL) agents, mitigating the details of ANN design and providing a simple symbolic encoding interface. While the WMtk allows RL agents to perform well in partially-observable domains, it requires prefiltering of sensory information by the programmer: a task often delegated to dimensional attention mechanisms in other cognitive architectures. To fill this gap, we develop and test a biologically-plausible dimensional attention filter for the WMtk and validate model performance using a partially-observable 1D maze task. We show that the attention filter improves learning behavior in two ways by: 1) speeding up learning in the short-term, early in training and 2) developing emergent alternative strategies which optimize performance over the long-term.

CCS CONCEPTS

• **Computing methodologies** → **Sequential decision making**; *Cognitive science*; • **Applied computing** → Psychology.

KEYWORDS

Reinforcement Learning, Artificial Neural Networks, Working Memory, Dimensional Attention Learning

ACM Reference Format:
Ngozi Omatu and Joshua L. Phillips. 2021. Benefits of Combining Dimensional Attention and Working Memory for Partially Observable Reinforcement Learning Problems. In *2021 ACM Southeast Conference (ACMSE 2021), April 15–17, 2021, Virtual Event, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3409334.3452072

ACMSE 2021, April 15–17, 2021, Virtual Event, USA

ACM ISBN 978-1-4503-8068-3/21/04…$15.00
https://doi.org/10.1145/3409334.3452072

1 INTRODUCTION

The fields of neuroscience and artificial intelligence (AI) have a long and intertwined history. Since the age of computers, works on AI have been inextricably interlinked with neuroscience and psychology, with collaborations between these disciplines proving highly productive for early pioneers [2, 5, 6, 15]. Particularly in the area of artificial neural networks (ANNs), the benefits of developing ANNs based on examining biological cognition and its neural implementation have been substantial. Neuroscience provides a rich source of inspiration for new types of algorithms and architectures to employ when building ANNs. Likewise, these ANNs can be viewed as formal, testable hypotheses of brain function, and neuroscientific experiments can provide validation or refutation of such ANNs.

There is considerable evidence that the brain uses working memory (WM) which plays an important role in a wide variety of high-order cognitive tasks including learning new information, following directions, taking notes, reasoning, and problem solving [3]. WM operates by maintaining a small amount of task-essential information which focuses attention, limits the search space for perceptual systems, and/or helps avoid the out-of-sight/out-of-mind problem and being obdurate towards irrelevant events [1, 16]. WM in humans and animals is hypothesized to subsist in the interaction of two major neural components: the prefrontal cortex (PFC) and mesolimbic dopamine system (MDS). The PFC functions as a fixed storage for task-related information while the MDS evaluates the efficacy of such information. Biologically-based ANNs for WM have been developed based on MDS in the mesolimbic pathway from electrophysiological, neuroimaging, and neuropsychological studies [7, 10]. Additionally, reinforcement learning is another area that links ANNs and neuroscience together. One of the breakthroughs in reinforcement learning is temporal difference learning (TD), the goal of which is for the learning system (the 'agent') to be able to estimate the values of different states or situations in terms of future rewards or punishments. Electrophysiological studies of MDS also suggested that the firing rates of cells in the MDS encodes for changes in expected future rewards [9, 13].

The working memory toolkit (WMtk) was developed to assist the integration of ANN-based WM into reinforcement leaning (RL) agents to allow the solution of partially-observable RL problems by attenuating the details of ANN design and providing a simple symbolic encoding interface [4, 12]. However, the current WMtk requires the user (programmer) to guide it in a general way as to what kinds of information are relevant for the task at hand (candidates for storage in WM). This is problematic as the WMtk was

designed to imitate biological WM which appears to utilize self-sufficient mechanisms to solve this problem automatically in other learning domains. For example, ANN-based category learning models often employ dimensional attention mechanisms to filter out irrelevant or distracting information which leads to faster learning similar to human performance [8, 11]. Our aim is to integrate a similar biologically-plausible dimensional attention filter into the WMtk framework in order to autonomously perform tasks in the face of distracting stimuli. We test the performance of our augmented model using a partially-observable 1D maze task to explore the impact of dimensional attention on learning speed and agent behavior.

2 METHODS

The WMtk traditionally utilizes the standard TD-learning [14] method for learning the value function encoded using a single layer ANN. We replaced the ANN with a value function look-up table for all of our experiments to ensure that any observed advantages/disadvantages for our model were not due to function approximation errors. This allows us to encode, manipulate and store information similar to how the WMtk operates, namely using values encoded from location-signal-memory triplets ($V(x, s, m)$) instead of state alone ($V(s)$). Below, we develop a biologically-plausible attention filter around our WMtk and experiment with the results of four possible filter conditions (static filtering), and also at a range of thresholds (dynamic filtering), in order to investigate on-demand filter switching. During this phase, we have our WMtk preform basic memory retrieval tasks to help evaluate the performances of each set of thresholds and explore the sensitivity of the initial threshold parameter settings.

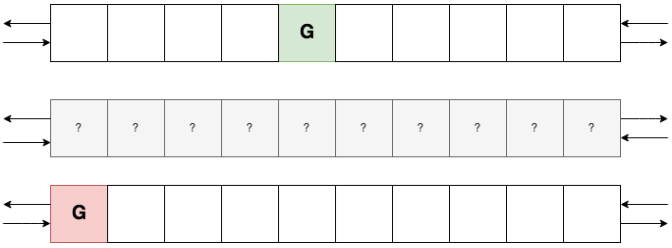

Figure 1: Representation of the 1D Maze

We created a reinforcement learning agent which utilized TD learning to solve a 1D maze task consisting of 10 independent states as seen in Figure 1. Each state has two immediately accessible neighbors to the left and right, such state 5 has access to its left neighbor, state 4, and right neighbor, state 6.The maze is periodic such that state 10 is the right neighbor of state 1, and 1 is the left neighbor of state 10. For each color *signal*, there is a 1D maze with its own reward state. The green signal indicates a reward is in state 5 and the red signal indicates a reward located in state 1. The result is two conflicting, partially-observable policies as only one of the color signals is used for each episode of the 1D maze task. Internal to the agent, we used a 3D array that could hold the value representation of all the possible state-memory-signal triplets that the agent could potentially have (dimensional and memory combination) and a 2D array to represent the environment and where the rewards are located. The table lookup system is being used, due to prior issues that risen when the attention filter was used with ANNs. This is to determine whether the current attention filter is the cause of the issues.

TD learning is primarily concerned with learning the value function (discounted future rewards), and the agent is implemented through a series of functions under a for-loop imitating the 3 main function in the original WMtk: Initialize Episode, Step, and Absorb Reward. First, an episode resets all episode-specific variables and clears the WM. To make the task partially observable, the color signal is provided only on the first step of each episode which was correlated with the location of the goal for that episode. This is done to create good test of WM performance as the agent also has to determine whether or not the information is relevant for the task. The color signal (and therefore the goal) is selected randomly and shown to the agent along with its location in the maze. At this step, the agent is either rewarded if it happens to be on the reward state or has to choose between remembering the color signal, its location, or neither. The agent chooses based on which option has the highest value. After this, the agent calculates the value of its current state and uses those values along with those stored from the previous state to update the WM value array using the TD learning equation. The agent stores the current state value for use in the next step of the episode. Finally, if the agent lands on the reward location, the TD error is computed using the previous state value as well as the final state value (absorbing the reward). Typically, a scalar reward of zero is provided throughout all steps of the task. On the final step, a reward value of 1 is provided, thus indicating that the agent successfully completes the task. When a new episode begins, these functions are called again, in the same order.

In addition to the base model above, an dimensional attention filter is added to the agent. The attention filter is composed of two adjustable parameters representing the dimensions of features that could be filtered out. In the case of the task above, the two dimensions that might be filtered out are the signal color and the agent location. The dimensional features that are currently being filtered is demonstrated with the value of a parameter being 0 or 1 for on and off, respectively. At the start of an episode, the parameters are observed to determine if the perceptual information can be taking in by the working memory. If the parameter holds 1 (filter is off), the WM is allowed to consider the feature information relating to the corresponding dimension. Additionally, the filters have a threshold function which checks on when a dynamically update TD error accumulator variable crosses from positive to negative or vice-versa. If an accumulator is below 0, it will turn the boolean parameter for the corresponding dimension to 0 (off). The accumulator value is calculated using the TD delta equation:

$$\delta_t = (r(x, s) + \gamma V(x, s, m)_{t+1}) - V((x, s, m)_t \quad (1)$$

where δ_t is the delta delivered on each time step to predict the change in expected future reward, given features of the current situation. The δ_t can be added or subtracted with each accumulator's current value depending on their corresponding filter's on/off setting, respectively. To determine reasonable initial settings for the accumulator values, we calculated their values during simulations with the two filters statically set to all four on/off combinations.

This provided good estimated starting points, and also provided guidance on the range of initial accumulator settings to use for the sensitivity study below.

The learning parameters for all tasks are set to these values: learning rate parameter, α=0.03; future reward discounting factor, γ=0.9; ϵ-soft random working memory selection probability =0.01; number of working memory slots =1;

3 RESULTS

When testing the dimensional attention filter with the 1D maze task, we are looking for two criteria for success as mentioned in the introduction: 1) how the agent reacts to the maze for each filter condition (static attention) and 2) how does the agent perform for a wide range of initial accumulator values (dynamic attention).

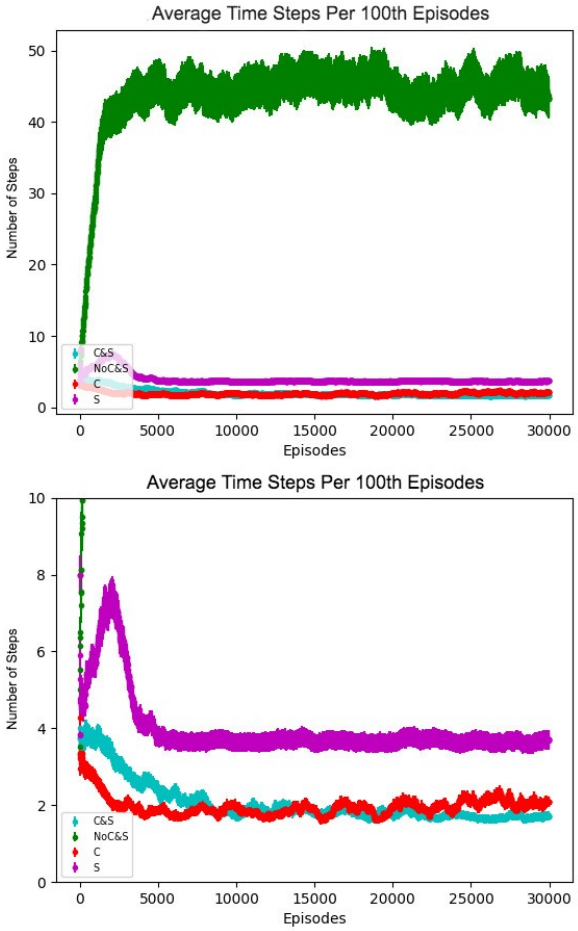

Figure 2: The Average Time Steps of Each Filter Conditions

During the 30000 learning trials, we recorded the average steps per 100 episodes (sliding window) for each of the four filtering conditions. The results are shown in Table 1 and Figure 2. Overall for the static attention models, when attention to both color and state are turned off, the agent performed very poorly (large numbers of steps). Given state attention only (color attention turned off), the performance is better but not as good as when only using color attention (state attention turned off). When attention to both the state and color dimensions is turned on, there is a only slight penalty early in learning (compared to color-only attention) but that loss is made up for in better performance later in learning. However, the dynamic model shows the best performance when starting with both dimensions active, suggesting that it quickly turns off state attention to speed learning early-on, and then adds it back later to achieve better performance later in learning. This phenomenon is best observed in Figure 2. As anticipated, condition No C&S has dramatically increased in the number of steps to approximately 40 steps before reaching the 5000 episodes and remaining between the range of 40 and 50. This may be due to the attention filter activation for both features, meaning that the WM is not taking in any information from the environment. Thus, for each episode, the agent is "taking a shot in the dark" by making random decisions on its next step. Condition S, filtering out color, allows the agent to learn using only location and did remarkably well, with the agent being able to solve the task under approximately 4 steps per episode. Notably, there is a inverse relationship between the condition CS, both color and state are taken in by agent, and condition C, where only the state dimension (distractor) is filter out. CS displayed higher performance than C early in training, but the average number of steps started to dramatically decline later in learning. This indicates that while removing distractions helps the agent learn quickly in the beginning, it doesn't help the agent perform better in the long run. Along with the result of the filtering condition, we collected the accumulator value that is produced by each condition to help guide our later sensitivity study.

Table 1: Average Number of Steps in the Maze for Both Static Filtering and Dynamic Filtering

		Static		Dynamic	
Color	State	Early	Late	Early	Late
On	On	2.63	1.66	2.57	1.67
On	Off	2.01	1.94	3.12	2.04
Off	On	4.52	3.52	28.59	3.36
Off	Off	40.66	43.95	40.91	44.60

Along with the result of the four filtering conditions, we observed the accumulator values that were produced by each condition. As shown in Table 1, we observe the average steps for the first 10,000 episodes (Early learning) and the last 10,000 episodes (Late learning) from the filtering conditions when the agent performs the task with both static filtering and dynamic filtering. With the result of the dynamic attention filter, we evaluate the values of the dynamic attention filter to determine the condition in the thresholds gradient.

Figure 3 shows the average steps the agent took during early learning and late learning under the thresholds. The threshold range from -1500 to 1500 at 50 steps for each feature. In the threshold gradient, we can observed the values of the average steps for the early and late learning of the thresholds and compare it to the values observed from the table. This demonstrates how the value of the thresholds heavily affects how the filter changes from one condition to another. The on/on condition can be seen by setting

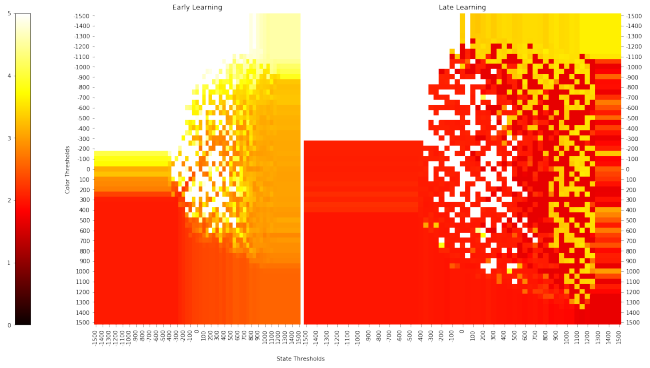

Figure 3: The Value of the Average Steps During Early Learning and Late Learning

the color threshold at regular intervals from the range -1500 to -300 and state threshold from -1500 to 700 with a few exception. The off/off condition is shown with color threshold set from range 200 to 1500 and state threshold set to -1500 to 0. The on/off condition appears to be more scarce with color threshold ranging from -1500 to -1100 and state threshold from 750 to 1500. However, this applies to the results in dynamic filtering: the off/on condition has color threshold ranging from 600 to 1500 and state threshold 50 to 1500.

The performance of the agent when completing the task is greatly affected by the filtering condition. The filtering of state locations demonstrates a benefit of quicker early learning for the first episodes. However, learning develops better in the long run if the filter was left off. This in turn give an idea of what the average time steps are for each condition. This information gave us a lead to evaluate initial accumulator value for similar values when both dimension are active. This resulted with threshold ranges that can be used to augment the attention filter. Although, some thresholds in these ranges do not produce the results that can easily be predicted. Such as the values of the state threshold ranging -400 to 800, where a mixture of the values conditioned to the filter switches while the majority equal to values as lower than 2 or extremely high.

In the end, the results here suggest similar learning benefits to dimensional attention shown in the category learning literature: that dimensional attention helps focus attention on relevant stimulus dimensions and ignore distracting dimensions in order to speed learning. The same kind of learning speedups were observed by filtering out the irrelevant state dimension from the candidates for working memory storage early in learning. However, later in learning, the dynamic attention mechanisms presented here turn off this filtering at some point. This is done to improve the overall asymptotic performance later in learning since the state locations can be used in a counter-intuitive manner to perform the tasks more optimally. In particular, if a random, exploratory move by the ϵ-soft policy resulted in randomly forgetting the color signal for the current episode, a reasonable backup strategy consists in visiting the closest of the two goals and then remembering that this state was already visited to allow the agent to learn to return as quickly as possible to the other goal. This strategy of remembering the location where the goal was not present results in better asymptotic performance on the tasks.

4 DISCUSSION

Our results show that filtering aids the WM in determining relevant information to store for the task and provides an appropriate replacement to manual prefiltering by adapting dimensional attention during learning. This dynamic filtering illustrates the utility of WM as an attention focusing mechanism. It was found that the agent had created "alternative strategies" when it chose to recall its location. After a number of episodes, the agent would use its location as a way of determining where the reward's location is. For example, if the agent was preforming a red task and chose to remember it's current location it could use the fact that it can't remember the goal state's location in a trial and error strategy. From there, it can build a map from the values it created and use it to reach reward state. The average times steps during the CS condition suggests that the alternative strategies requires a sufficient amount of episodes in order to show the benefits of quicker learning over a in the long run. Notably, the 1D maze task (with state distractors) was previously beyond the capabilities of the toolkit. In the original WMTK, the 1D maze task was an arduous task when location was added for the WM to consider for retention for the agent. Yet, the tabular version was able to not only solve the task, but used the new information to its advantage as an alternative strategy to be used if the WM forgot the color. This may in-part be due to the tabular version being more precise when it calculates the value of each state location. There were times where the WM made a decision when two options had values that were only different at the ten-thousandth place. This compare to the ANN 's approximate nature gave the tabular version an advantage.

Autonomous dimensional attention learning allowed for new complex concepts to emerge in problem solving tasks. Thus, the WM can work on complex tasks with an approach similar to its biological counterpart. Additionally, the user no longer needs background knowledge on how to construct sparse, distributed, conjunctive codes to filter out information. The user also does not need to rewrite filter functions when extending the maze spaces or using different numbers of dimensions. The filtering condition results also buttressed the dynamic filter since it can switch between different conditions depending on the TD error alone. The filter, in dynamic form, shows promise in its ability to autonomously change between the filter conditions. By changing the threshold to match the condition ranges, it can gain both of the benefits of quick learning and optimal performance from the filter conditions. This can make for further research on how useful this ability may be when testing on WM-related tasks.

The development of the dimensional attention filter has opened up several new avenues for future work. However, the filter has its own limitations. It was built specifically for the 1D maze task. If any expansion on complexity of the task were to occur, it would require additional changes to take up these possibilities. It is also worth trying the attention filter on the delayed saccade task, where the attention filter could provide a different strategy for completing the task. Additionally, tasks that require long memorizing sequences of information quickly would help develop our understanding of the limitations of the WM and test out how dimensional attention filtering might react in such a task.

ACKNOWLEDGEMENTS

Funding for Ngozi Omatu was provided by a grant from the MTSU Office of Research Undergraduate Research and Creative Activity (URECA) program, and from a grant from the Tennessee Louis Stokes' Alliance for Minority Participation program at MTSU.

REFERENCES

[1] A. Baddeley. 1986. *Working Memory*. Oxford University Press. https://doi.org/10.1002/acp.2350020209.

[2] P. S. Churchland and T. J. Sejnowski. 1988. *Perspectives on Cognitive Neuroscience*. Vol. 242. Science. https://doi.org/10.1126/science.3055294.

[3] A. Conway and R. Engle. 1996. Individual Difference in Working Memory Capacity: More Evidence for a General Capacity Theory. 4, 6 (1996), 577. https://doi.org/10.1080/741940997.

[4] G.M. DuBois and J. L. Phillips. 2017. Working Memory Concept Encoding Using Holographic Reduced Representations. Modern Artificial Intelligence and Cognitive Science, Fort Wayne, US, 137–144.

[5] D. Hebb. 1949. The Organization of Behavior: A Neuropsychological Theory. 35, 5 (1949), 335. https://doi.org/10.1002/sce.37303405110.

[6] J. J. Hopfield. 1982. Neural Networks and Physical Systems with Emergent Collective Computational Abilities. *Proceedings of the National Academy* 79, 8 (1982), 2554–2558. https://doi.org/10.1073/pnas.79.8.2554 arXiv:https://www.pnas.org/content/79/8/2554.full.pdf.

[7] T. Kriete, D. C. Noelle, J. D. Cohen, and R. C. O'Reilly. 2013. Indirection and Symbol-like Processing in the Prefrontal Cortex and Basal Ganglia. *Proceedings of the National Academy of Sciences* 110, 41 (oct 2013), 16390–16395. https://doi.org/10.1073/pnas.1303547110.

[8] J. K. Kruschke. 1992. ALCOVE: An Exemplar-based Connectionist Model of Category Learning. *Psychological Review* 99 (1992), 22–44.

[9] Y. Niv. 2009. Reinforcement Learning in the Brain. 53 (2009), 139–154. https://doi.org/10.1016/j.jmp.2008.12.005.

[10] R. C. O'Reilly, D. C. Noelle, T. S. Braver, and J. D. Cohen. 2002. Prefrontal Cortex and Dynamic Categorization Tasks: Representational Organization and Neuromodulatory Control. *Cerebral Cortex* 12, 3 (Mar 2002), 246–257. https://doi.org/10.1093/cercor/12.3.246.

[11] J. L. Phillips and D. C. Noelle. 2004. Reinforcement Learning of Dimensional Attention for Categorization. The 26th Annual Meeting of the Cognitive Science Society, Chicago, US, 1101–1106.

[12] J. L. Phillips and D. C. Noelle. 2006. Working Memory for Robots: Inspirations for Computational Neuroscience. 5th International Conference on Development and Learning, Bloomington, US.

[13] W. Schultz. 1998. Predictive Reward Signal of Dopamine Neurons. 80 (1998), 1–27.

[14] R. S. Sutton and A.G. Barto. 1998. *Reinforcement Learning: An Introduction* (first ed.). The MIT Press. http://incompleteideas.net/book/the-book-2nd.html.

[15] A. Turing. 1950. Computing Machinery and Intelligence. 236 (1950), 433–460. https://doi.org/10.1093/mind/LIX.236.433.

[16] N. C. Waugh and D. A. Norman. 1965. Primary Memory. 72 (1965), 89–104. https://doi.org/10.1037/h0021797/.