ATTENTION DRIVEN DYNAMIC MEMORY MAPS

Anonymous authors
Paper under double-blind review

ABSTRACT

In order to act in complex, natural environments, biological intelligence has developed attention to collect limited informative observations, a short term memory to store them, and the ability to build live mental models of its surroundings. We mirror this concept in artificial agents by learning to 1) guide an attention mechanism to the most informative parts of the state, 2) efficiently represent state from a sequence of partial observations, and 3) update unobserved parts of the state through learned world models. Key to this approach is a novel short-term memory architecture, the Dynamic Memory Map (DMM), and an adversarially trained attention controller. We demonstrate that our approach is effective in predicting the full state from a sequence of partial observations. We also show that DMMs can be used for control, outperforming baselines in two partially observable reinforcement learning tasks.

1 INTRODUCTION

Biological creatures have evolved to efficiently process the deluge of information presented to us in every moment. Two components, attention and memory, are key to our success in operating under such complex conditions (Styles, 2006). We are able to predict parts of the world that are not immediately observed by utilizing previous observations and learned models of object behavior stored in memory. Furthermore, a persistent memory helps keep track of dynamic, moving objects. Attention is guided to where entities are not easily predictable, allowing us to keep our memory up-to-date and make events more predictable in the future. Hence, attention helps in the learning of dynamics models of the environment by focusing on objects and tracking their behavior through time. Observations persist in the mind's map of the surroundings, even after the attention has shifted away.

There has been evidence that suggests our own memories decay and require mental rehearsal, i.e., directed attention, to keep up to date (Barrouillet et al., 2004). Consider, for example, the task of crossing the street. You look left and right to store locations, heading, and speed of nearby vehicles, and perhaps other pedestrians. As you look around, your mind builds a live map based on short-term memory of the surroundings. You are aware that the car is still heading down the street even when you look away and will likely have impact on your location at some estimated time in the future. Now you can act to avoid the car, frequently checking back to update your model of its movement.

It is clear that attention and memory can have a mutual, positive impact on the other. Inspired by this pairing, our paper introduces a novel short term memory architecture and an attention mechanism that acts to gather information and train world models in a partially observable environment. Attention control, memory operations and world models are trained jointly, but attention is adversarial to the latter two. The world model aims to keep to memory updated so that it makes accurate predictions of the entire (unobserved) world state for the next step, while the attention is trained to focus on the most unpredictable parts. The attention mechanism outputs the next attention location based on the current map and is trained with RL to maximize the expected value of its reward—the prediction error under the attention window in the next step. This way, the attention presents the memory and model architecture with the most consistently difficult to predict aspects of its surroundings. And the memory/model is trained to better predict such situations in the future. As these models learn to foresee behavior of objects in the environment, the attention mechanism is forced to find more unpredictable or erratic occurrences. With a finite capacity model, adversarial training like this eventually leads to a state where the world model is able to predict most aspects of the world without direct observation and recording in memory, and the attention moves between high error areas, keeping the memory fresh there, thus enabling the agent to act under partial observability.

2 RELATED WORKS

Inspired by human perception, computer vision techniques for saliency detection prioritize processing only the "interesting" image regions (Itti et al., 1998). However, these techniques generally focus on local low-level image features and are task-agnostic. Although, object detection methods include selecting task-relevant regions likely to contain objects (Van de Sande et al., 2011) they were developed for static environments (images) and the attention mechanism is not learned. Our attention mechanism operates in a dynamic environment and we introduce a co-training method that learns where to glimpse in conjunction with performing the task. Other models in deep learning have shown the utility of co-training attention with the task objective. For instance, image captioning (Xu et al., 2015) and image generation Gregor et al. (2015). However, all of these techniques do have access to the whole image and their focus is to reduce computation or distraction instead of dealing with partial observability. Similar to us, (Mnih et al., 2014) use attention in a simple dynamic game with the agent and an object. For this, they introduce an RNN framework to selectively attend to part of the input. They train the model jointly with the network performing the task, however, the attention mechanism has the same goal as the task. Our approach includes an adversarial reward for the attention that incentivizes glimpsing "interesting" regions of the environment.

Mapping has been studied in computer vision and robotics with a range of sensing modalities using techniques like simultaneous localization and mapping (Fuentes-Pacheco et al., 2015; Snavely et al., 2008). However, these approaches take a geometric approach while our approach relies on learning. More recently, works have aimed to learn cognitive maps within partially observable environments. Parisotto et al. 2017 describe a neural map, consisting of write, update, and read operations, that organizes memory in a spatial structure. We similarly train a memory structure within partially observable environments. One difference is that we explicitly train our memory to learn representations that can be reconstructed into the full state. Additionally, we aim to learn in environments with dynamic objects. Our memory architecture is inspired from SpatialNet (Du & Narasimhan, 2019) to incorporate task-agnostic priors based on physics, however, we adapt this model for the partial-observability scenario, as explained in Section 3. Our results indicate the superior performance of our model as compared to SpatialNet for this task. Finally, our work utilizes an intrinsic reward to train the attention control. In particular, we use a notion similar to "surprise", which has been used for driving curiosity and exploration (Pathak et al., 2017; Schmidhuber, 2010).

3 MEMORY

Memory plays a key role in enabling agents to act in partially observable environments, i.e. where they do not have direct observation of the full state at every step. The agent must be able to reconstruct the environment, or at least parts of it relevant to its task, using the representation in its memory in order to act optimally. It must also be able to model the dynamics of moving objects in its surroundings over periods of time where it does not get direct observation of them. Finally, the effect of the agent's own actions must be reflected in the memory, whether directly observed or not. Here, we introduce the Dynamic Memory Map (DMM), a recurrent memory model that learns these desirable qualities from data. Special attention is paid to design modules within DMM that efficiently encode and update location of static and dynamic objects as well as the agent through time. This allows gradients to flow backwards in time efficiently, updating the DMM to better predict the location of an object glimpsed many time steps ago.

The DMM consists of three major modules: write, step and reconstruct. A glimpse is the partial observation under the attention window received by the agent at every time step. We assume that glimpses are presented to the agent embedded within the full state image, with the rest of the state zeroed out. In other words, the agent has access to the size of the glimpse, its location within the state and the size of the full state.

Write. This module encodes an incoming glimpse, O_t into the memory representation, M. The glimpse is first passed through a series of convolutional operations, W, possibly downsampling it for a more efficient representation, but preserving the 2D structure of the observation. Then it is *blended* with the current memory again using a series of convolutions, B. Finally it is written into the memory but only in the locations where the glimpse was made, leaving the rest of the memory intact. The write head can be written as:

$$M_t^w = C_t * B(W(O_t), M_t) + (1 - C_t) * M_t$$

where C_t is a mask which is 1 under the observation window and 0 otherwise.

Step. This module is responsible for modeling the dynamics of the environment and updating the memory to track the full state. Objects in the environment can be static or dynamic and may be affected by the agent's own actions. To efficiently model the changes to memory precipitated by different types of objects,

we split the memory into bundles. Each bundle is assigned a fixed number of channels within the memory representation. We assign each bundle a third of the memory channels.

The first *static* bundle, M_S, remains unchanged by the step module. This allows the write head to efficiently encode static objects within this bundle. This bundle acts as a skip connection between two glimpses at the same location at different times, allowing gradients to pass through long stretches of memory updates. Next, the *dynamic* bundle, M_D, consists of a series of residual convolutions, similar to SpatialNet (Du & Narasimhan, 2019). This bundle learns about the dynamics of objects in the scene, independent of what the agent is doing. The entire map within this bundle is updated at every step. Finally, the *ego* bundle, M_E, updates the map conditioned on the action selected by the agent. This attempts to predict the effect of the agent's actions on the environment at every step.

The updates carried out by this module can be represented as:

$$M'_D = D(M_D)$$
$$M'_E = E(M_E, a_t)$$
$$M_{t+1} = \langle M_S, M'_D, M'_E \rangle$$

Reconstruct. This module converts the memory representation to a reconstruction of the full state using a series of deconvolutions, R. Reconstruction is essential for training the other two modules of DMM. The error under the glimpse between the reconstruction and the true full state can be set as a training loss. The *write loss*, L_w is incurred under the current glimpse immediately after writing to the memory. This is an autoencoder loss for correctly encoding the immediate observation into the memory: $L_w = C_t * ||R(M_t^w) - O_t||_2$. The *step loss*, L_s is incurred after stepping the memory and under the glimpse in the next observation. Since the DMM does not know where the next glimpse location will be, it tries to reconstruct the full state as faithfully as possible: $L_s = C_{t+1} * ||R(M_{t+1}) - O_{t+1}||_2$. In addition to reconstruction losses, two regularization losses are incurred by the DMM. These are the mean absolute value of the output by the write and the *dynamic* and *ego* bundles of the step head.

The total loss for training DMM is (where $\alpha = \beta = 0.01$ here):

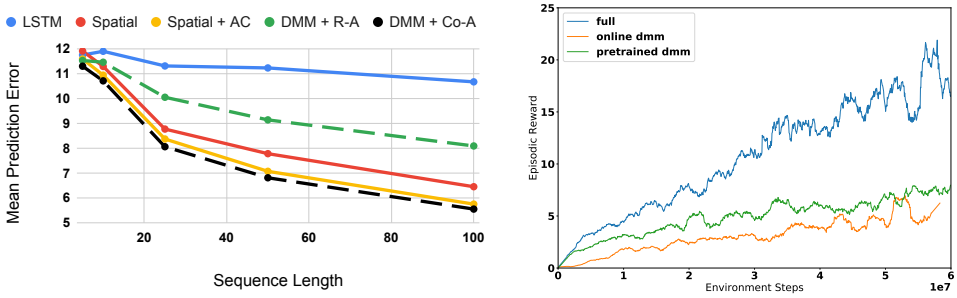
$$L = L_w + L_s + \alpha * |C_t * B(W(O_t), M_t)| + \beta * (|M'_D| + |M'_E|),$$

4 Attention Control

In our setup, control of the glimpse location is with the agent as well. The agent can decide where to look in the environment in order to collect information to store in its memory. The best location to glimpse, though, depends on the current contents of the memory. The agent may want to collect more information about parts of the state that it has not explored yet, or may want to glimpse a certain object whose dynamics it is not familiar with. We outline a solution through adversarial training of the attention with memory. After each step, the memory creates a reconstruction of the full state, $R(M_{t+1})$. We train the glimpse control to attend to those areas of the state that generate the most error between the reconstruction and the true observation. In other words we train the attention to maximize the *step loss*, L_s, at the same time the memory tries to minimize it.

Since the memory does not have access to the next observation (yet) and does not know where the attention will be placed, L_s is inherently a stochastic quantity. We treat the attention control as a reinforcement learning problem, setting L_s as the reward for the glimpse agent. The attention control agent can be seen of as a surprise seeking RL agent, looking to maximize the amount of error or surprise between the predictions and ground truth. Attention control is trained using an on-policy policy gradient approach A2C (Wu et al., 2017; Mnih et al., 2016). We found this to produce better results than using an off-policy approach such as double DQN (Van Hasselt et al., 2016), possibly because the state space for the glimpse control agent is memory representation, M, which is changing as the DMM trains, thus making the state space non-stationary. The policy network for the glimpse control agent is fully convolutional, upsampling the memory M_t to a softmax policy, π_t^G, over all possible locations of glimpse in the subsequent observation.

Adversarial training of attention and memory leads to some interesting dynamics in what parts of the world are attended to. Since the glimpses essentially form the training data for DMM, the more a particular object is attended to, the better the DMM becomes at representing it. As the same object is focused on through time, the DMM becomes better at predicting its dynamics. This results in diminishing returns for the glimpse agent, incentivizing it to explore through the state space to find more interesting objects.



(a) (b)

Figure 1: (a) Mean pixel prediction error for each model. Each model was trained on sequences of fixed length 25 but we predict states for a range of sequences from short (5) to long (100). Reconstruction error is measured over the full state at each time step and averaged over the sequence. (b) RL in PhysGoal task using a pretrained DMM and online learned DMM. Comparison with learning using full state observations.

Practically this produces a training curriculum for DMM. At first, the background and static objects in the scene provide the most reliable source for high reconstruction error. The DMM quickly learns to represent these making use of the skip connections through time provided by the *static* bundle. The attention then moves on to dyanamic objects under constant motion. It learns to track them for a reliable source of reward, until the future locations of such objects are predictable by the *dynamic* bundle. In parallel, the *ego* bundle learns the effects of the agent's own actions on the state. Eventually, the glimpse agent learns to predict when collisions between objects will happen and jumps to them as that is the hardest to learn to model.

5 Predicting Full State

First, we evaluate how well the DMM reconstructs the full dynamic state from a series of glimpses. We collect a dataset of 500k images from the *PhysGoal* environment by Du & Narasimhan (2019) by using rollouts from a trained agent for this task. The agent's actions while performing the task are also recorded as our method and some of the baselines make use of it. The size of the state is 84×84 pixels and we assume an attention head of 21×21 pixels is available for all the methods. We train the DMM along with a number of ablations of our approach and baselines from prior work.

- LSTM - Underlying memory representation used is a flat LSTM. The write/step/reconstruct operations for this memory are as usual without a masked write head and bundles, but step module is conditioned on agent action.
- SpatialNet - The closest memory architecture to ours in the literature, does not employ masked write heads, bundles or action conditioning.
- SpatialNet + AC - SpatialNet with additional action conditioning during step.
- DMM + R-A - DMM trained with glimpses from random locations in the state. An ablation highlighting the importance of co-training attention with memory.
- DMM + Co-A - Our full method trained with co-attention

Figure 1a shows the results for testing sequences, excluded from training set, of varying lengths. At just 5 glimpses, all methods have equal difficulty in reconstructing the full state (it would take at least 16 glimpses to cover the entire state area once). From 10 glimpses onward, our method consistently beats all baselines, past the sequence length of the training set (25).

We also evaluate if the representation learned by DMM can be used for control in the environment. For this task, the agent must control the attention to build a picture of its environment, while at the same time acting within it to navigate to a goal using its memory. The environment contains enemies which terminate the episode (-1 reward) upon collision and the goal randomly moves every time it is reached (+1 reward), making the environment challenging with partial observations. Figure 1b shows the average episodic reward over 5 testing runs for an RL agent with a pretrained DMM and one with a DMM learned online while exploring, compared against the upper bound of an RL agent with fully observed state. Although our method is not able to achieve a score as high as having access to full observations, it learns to successfully navigate to the goal while avoiding enemies multiple times in a single episode for both versions.

REFERENCES

Pierre Barrouillet, Sophie Bernardin, and Valérie Camos. Time constraints and resource sharing in adults' working memory spans. *Journal of Experimental Psychology: General*, 133(1):83, 2004.

Yilun Du and Karthik Narasimhan. Task-agnostic dynamics priors for deep reinforcement learning. *arXiv preprint arXiv:1905.04819*, 2019.

Jorge Fuentes-Pacheco, José Ruiz-Ascencio, and Juan Manuel Rendón-Mancha. Visual simultaneous localization and mapping: a survey. *Artificial intelligence review*, 43(1):55–81, 2015.

Karol Gregor, Ivo Danihelka, Alex Graves, Danilo Jimenez Rezende, and Daan Wierstra. Draw: A recurrent neural network for image generation. *arXiv preprint arXiv:1502.04623*, 2015.

Laurent Itti, Christof Koch, and Ernst Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on pattern analysis and machine intelligence*, 20(11):1254–1259, 1998.

Volodymyr Mnih, Nicolas Heess, Alex Graves, et al. Recurrent models of visual attention. In *Advances in neural information processing systems*, pp. 2204–2212, 2014.

Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pp. 1928–1937, 2016.

Emilio Parisotto and Ruslan Salakhutdinov. Neural map: Structured memory for deep reinforcement learning. *arXiv preprint arXiv:1702.08360*, 2017.

Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 16–17, 2017.

Jürgen Schmidhuber. Formal theory of creativity, fun, and intrinsic motivation (1990–2010). *IEEE Transactions on Autonomous Mental Development*, 2(3):230–247, 2010.

Noah Snavely, Steven M Seitz, and Richard Szeliski. Modeling the world from internet photo collections. *International journal of computer vision*, 80(2):189–210, 2008.

Elizabeth Styles. *The psychology of attention*. Psychology Press, 2006.

Koen EA Van de Sande, Jasper RR Uijlings, Theo Gevers, and Arnold WM Smeulders. Segmentation as selective search for object recognition. In *2011 International Conference on Computer Vision*, pp. 1879–1886. IEEE, 2011.

Hado Van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. In *Thirtieth AAAI conference on artificial intelligence*, 2016.

Yuhuai Wu, Elman Mansimov, Shun Liao, Alec Radford, and John Schulman. Openai baselines: Acktr and a2c, 2017.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pp. 2048–2057, 2015.