HARD ATTENTION CONTROL BY MUTUAL INFORMATION MAXIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Biological agents have adopted the principle of attention to limit the rate of incoming information from the environment. One question that arises is if an artificial agent has access to only a limited view of its surroundings, how can it control its attention to effectively solve tasks? We propose an approach for learning how to control a hard attention window by maximizing the mutual information between the environment state and the attention location at each step. The agent employs an internal world model to make predictions about its state and focuses attention towards where the predictions may be wrong. Attention is trained jointly with a dynamic memory architecture that stores partial observations and keeps track of the unobserved state. We demonstrate that our approach is effective in predicting the full state from a sequence of partial observations. We also show that the agent's internal representation of the surroundings, a live mental map, can be used for control in two partially observable reinforcement learning tasks. Videos of the trained agent can be found at `https://sites.google.com/view/hard-attention-control`.

1 INTRODUCTION

Reinforcement learning (RL) algorithms have successfully employed neural networks over the past few years, surpassing human level performance in many tasks (Mnih et al., 2015; Silver et al., 2017; Berner et al., 2019; Schulman et al., 2017). But a key difference in the way tasks are performed by humans versus RL algorithms is that humans have the ability to focus on parts of the state at a time, using attention to limit the amount of information gathered at every step. We actively control our attention to build an internal representation of our surroundings over multiple fixations (Barrouillet et al., 2004). We also use memory and internal world models to predict motions of dynamic objects in the scene when they are not under direct observation. By limiting the amount of input information in these two ways, i.e. directing attention only where needed and internally modeling the rest of the environment, we are able to be more efficient in terms of data that needs to be collected from the environment and processed at each time step.

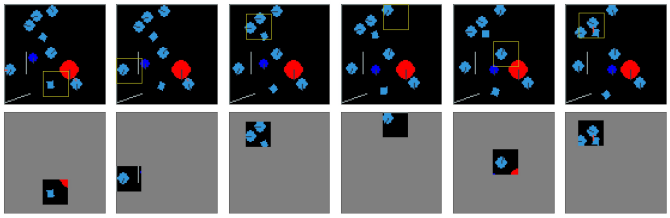


Figure 1: Illustration of the PhysEnv domain (Du & Narasimhan, 2019). Here, the agent (dark blue) needs to navigate to the goal (red) while avoiding enemies (light blue). Typically the full environment states (top) are provided to the RL algorithm to learn. We consider the case where only observations from under a hard attention window (bottom) are available. The attention's location (yellow square) at each time step is controllable by the agent. Using only partial observations, the agent must learn to represent its surroundings and complete its task.

By contrast, modern reinforcement learning methods often operate on the entire state. Observing the entire state simultaneously may be difficult in realistic environments. Consider an embodied agent that must actuate its camera to gather visual information about its surroundings learning how to cross a busy street. At every moment, there are different objects in the environment competing for its attention. The agent needs to learn to look left and right to store locations, heading, and speed of nearby vehicles, and perhaps other pedestrians. It must learn to create a *live* map of its surroundings, frequently checking back on moving objects to update their dynamics. In other words, the agent must learn to selectively move its attention so as to maximize the amount of information it collects from the environment at a time, while internally modeling motions of the other, more predictable parts of the state. Its internal representation, built using successive glimpses, must be sufficient to learn how to complete tasks in this partially observable environment.

We consider the problem of acting in an environment that is only partially observable through a controllable, fixed-size, hard attention window (see figure 1). Only the part of the state that is under the attention window is available to the agent as its observation. The rest must be inferred from previous observations and experience. We distinguish this task from that of learning *soft* attention (Vaswani et al., 2017), where the full state is attended to, weighted by a vector, and then fed into subsequent layers. Our system must learn to 1) decide where to place the attention in order to gather more information about its surroundings, 2) record the observation made into an internal memory and model the motion within parts of the state that were unobserved, and 3) use this internal representation to learn how to solve its task within the environment.

Our approach for controlling attention uses RL to maximize an information theoretic objective closely related to the notion of surprise or novelty (Schmidhuber, 1991). It is unsupervised in terms of environment rewards, i.e. it can be trained on offline data (states and actions) without knowing the task or related rewards. We discuss this in more detail in section 5. Memory also plays a crucial role in allowing agents to solve tasks in partially observable environments. We pair our attention control mechanism with a memory architecture inspired largely by Du & Narasimhan (2019)'s SpatialNet, but modified to work in partially observable domains. This is described in section 4. Emperically, we show in section 6.1 that our system is able to reconstruct the full state image including dynamic objects at all time steps given *only* partial observations. Further, we show in section 6.2 that the internal representation built by our attention control mechanism and memory architecture is sufficient for the agent to learn to solve tasks in this challenging partially observable environment.

2 Related Works

Using hard attention for image classification or object recognition is well studied in computer vision (Alexe et al., 2012; Butko & Movellan, 2009; Larochelle & Hinton, 2010; Paletta et al., 2005; Zoran et al., 2020; Welleck et al., 2017). Attention allows for processing only the salient or interesting parts of the image (Itti et al., 1998). Similarly, attention control has been applied to tracking objects within a video (Denil et al., 2012; Kamkar et al., 2020; Yu et al., 2020). Surprisingly, not a lot of recent work exists on the topic of hard attention control in reinforcement learning domains, where a sequential decision making task has to be solved by using partial observations from under the attention.

Mnih et al. (2014) proposed a framework for hard attention control in the classification setting and a simple reinforcement learning task. Their approach consists of using environment rewards to train an attention control RL agent. Our approach differs mainly in that we train the attention control using our novel information theoretic objective as reward. Mnih et al. (2014)'s approach leads to a task specific policy for attention control, whereas our approach is unsupervised in terms of the task and can be applied generally to downstream tasks in the environment. Our approach also differs in that we use a memory architecture that is more suited to partially observable tasks with 2D images as input, compared to a RNN used by Mnih et al. (2014).

There has been much prior work on memory and world models for reinforcement learning (Ha & Schmidhuber, 2018; Graves et al., 2016; Hausknecht & Stone, 2015; Khan et al., 2017). The work closest to our own is Du & Narasimhan (2019)'s SpatialNet, which attempts to learn a task-agnostic world model for multi-task settings. Our memory architecture is largely inspired by SpatialNet and adapted to work in the partially observable setting. We also use their PhysEnv environment to evaluate our approach. Closely related work is Neural Map (Parisotto & Salakhutdinov, 2017),

which uses a structured 2D memory map to store information about the environment. Their approach also applies to partially observable RL tasks, but the attention is fixed to the agent. In contrast, we consider the problem of learning to control an attention window that can move independently of the agent location. Recently, Freeman et al. (2019) showed that world models can be learnt by simply limiting the agent's ability to observe the environment. They apply *observational dropout*, where the output of the agent's world model, rather than the environment state, is occasionally provided to the policy. We consider the related scenario where only a part of the true environment state is provided to the agent at each time step and the rest must be modeled using previous observations.

Finally, mutual information has been used to train self-supervised RL agents. This line of work originates in curiosity driven and intrinsically motivated RL (Schmidhuber, 1991; Pathak et al., 2017; Bellemare et al., 2016). Typically, some notion of predictive error or novelty about an aspect of the environment is optimized in lieu of environment rewards. Multiple papers have successfully used different formulations of mutual information to learn how to efficiently explore the environment without extrinsic rewards (Mohamed & Rezende, 2015; Houthooft et al., 2016; Achiam et al., 2018; Gregor et al., 2016; Eysenbach et al., 2018; Sharma et al., 2019). We apply the idea of using mutual information to the problem of curiosity driven attention control.

3 PRELIMINARIES

In this work, we are concerned with tasks solved using reinforcement learning (RL). RL is the study of agents acting in an environment to maximize some notion of long term utility. It is almost always formalized using the language of Markov decision problems (MDPs). States, actions, rewards and transitions form the components of an MDP, often represented as the tuple $\langle S, A, R, T \rangle$. Maximizing the expected sum of (discounted) rewards over rollouts in the environment is usually set as the objective of learning. Once a problem is formulated as an MDP, such that the components $\langle S, A, R, T \rangle$ are well defined, one can apply a range of model-free RL algorithms to attempt to solve it (Schulman et al., 2017; Mnih et al., 2016; Wu et al., 2017). A solution to an MDP is typically sought as an optimal policy, $\pi^* : S \rightarrow A$, which is a function that maps every state to an action that maximizes the long term expected rewards. In this section, we will attempt to formalize the components of the partially observable reinforcement learning problem under study.

3.1 INTERNAL REPRESENTATION

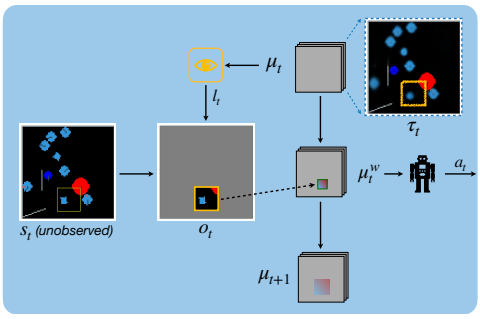


Figure 2: The agent starts with a representation (μ_t) from which it attempts a reconstruction (τ_t) of the likely current full (unobserved) state s_t. Then, it picks an attention location (l_t) and receives an observation of the state from under the attention (o_t). The observation is written into μ_t to form μ_t^w. The entire map is then stepped through an internal world model to update the dynamical objects, forming μ_{t+1}, the representation for the next step.

First, we give a brief description of how the agent stores observations in an internal representation of its surroundings (figure 2). An map (μ) tracks the full environment state (s), which is never directly observed. The map is empty at the start of the episode and gets sequentially written into as observations are made. The agent also creates a reconstruction of the full state (τ) at every time step based on its map. This will become useful later for training. For now, it suffices that μ_t is the map *before* an observation is made and μ_t^w is the one *after* the observation is made. μ_{t+1} is the map after the dynamics of the system for the next time step are taken into account.

3.2 THE TWO AGENTS

We formulate the solutions to controlling the attention at every time step and completing the environment task as two separate reinforcement learning agents. The attention location is controlled by the *glimpse agent* (the eye in figure 2), and actions within the environment are taken by the regular agent. These two agents have their own separate MDPs with their $\langle S, A, R \rangle$ tuples defined below.

The glimpse agent's state at every timestep is μ_t. Its set of actions is all possible attention locations within the full state (Width $\times$ Height actions). Its reward is based on the information theoretic objective discussed in section 5. Thus, the glimpse agent is provided the map before an observation is made and it must decide *where* the observation should be made from in order to optimize its reward.

The environment agent acting in the environment receives as input μ_t^w, i.e. the internal representation after the observation has been recorded. Its actions are the normal set of actions in the environment and its reward is the normal environment reward. We emphasize that neither agent has access to the full state at any time. They must both act based on the internal representation alone. They also cannot make multiple observations from the same environment state. Once an observation is made, an action must be selected that will change the environment state. In the next section, we will describe in detail how μ_t^w, μ_{t+1} and τ_t are formed and how the internal map is trained through a sequence of partial observations.

4 DYNAMIC MEMORY MAP

Memory plays a critical role in enabling agents to act in partially observable environments. It allows the agent to remember parts of the state that are not under direct observation. Along with memory, the agent must be able to model the dynamics of objects in its environment, as it may not receive observation of them for long periods. Finally, the effect of its own actions must be reflected in its belief about the environment. For this purpose, we design a special recurrent memory unit for visual environments called the Dynamic Memory Map (DMM). DMM is inspired largely by Du & Narasimhan (2019)'s work on SpatialNet, but modified to handle partial observations and work in tandem with the glimpse agent.

4.1 MEMORY MODULES

The DMM consists of three major modules: write, step and reconstruct.

Write. This module encodes an incoming observation, o_t, into the current memory representation, μ_t. The observation is first passed through a series of convolutional operations, W, possibly downsampling it for a more efficient representation. Then it is blended with μ using a series of convolutions, B. Finally it is written into the memory but only in the locations where the observation was made, leaving the rest of the memory intact. The write operation can be written as:

$$\mu_t^w = C_t * B(W(o_t), \mu_t) + (1 - C_t) * \mu_t$$

where C_t is a mask which is 1 under the attention 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 from one time step to the next. Objects in the environment can be static or dynamic and may be affected by the agent's own actions. This module is implemented as a residual network, S, similar to SpatialNet (Du & Narasimhan, 2019). We additionally condition S on the agent's own action a_t. The entire map is updated at each time step to ensure that objects observed previously are also updated to new likely locations.

$$\mu_{t+1} = \mu_t + S(\mu_t^w, a_t)$$

Reconstruct. This module converts the memory representation to a reconstruction of the full state using a series of deconvolutions (R).

$$\tau_t = R(\mu_t)$$

A reconstruction can also be made immediately after the observation, i.e. $\tau_t^w = R(\mu_t^w)$. Reconstruction is essential for training the other two modules of DMM as we explain in section 4.2. The reconstruction error, or the discrepancy between the observation and the reconstruction under the attention window can be back-propagated through the write and step layers.

4.2 TRAINING LOSS

The write loss, L_w is incurred under the current attention window immediately after the observation is made into μ_t^w. This ensures that the immediate observation is correctly encoded into the memory and trains the *write* and *reconstruction* modules.

$$L_w = C_t * ||\tau_t^w - o_t||_2.$$

The *step loss*, L_s is incurred after stepping the map and under the attention window in the *next* step.

$$L_s = C_{t+1} * ||\tau_{t+1} - o_{t+1}||_2.$$

This trains the step module, S, to accurately model the motion of objects in the entire state, so that a faithful representation of the state can be guessed before the next observation is taken. Once the next observation is made, we can calculate the difference between what the agent's model expected to be at the glimpse location and what actually was there. In figure 2 this is the difference between the images under the yellow rectangle in τ_t and o_t. This quantity is the amount of *surprise* in the dynamics model and we will use of this again when training the glimpse agent.

In addition to reconstruction loss, the absolute value of the output by the *write* and *step* modules is also penalized in order to regularize the contents of the DMM. The total loss for as single step within the rollout is (where $\alpha = \beta = 0.01$ here):

$$L_t = L_w + L_s + \alpha * C_t * |\mu_t^w| + \beta * |S(\mu_t^w, a_t)|.$$

We sum this loss over the entire rollout and jointly minimize it over all steps. The reconstruction error at every location can be back-propagated to the step an observation was recorded there.

5 MAXIMIZING MUTUAL INFORMATION TO CONTROL ATTENTION

The intuition behind our approach is that attention should be used to gather information from the environment *when and where* it is required. The DMM is constantly modeling its environment and attention should be directed to parts of the world where the model is uncertain. This directly informs the agent about difficult to model parts of the environment, thus reducing uncertainty in the state as a whole. Secondly, by focusing on areas with hard to predict dynamics, more data can be collected from those areas, updating the model for future predictions.

5.1 MUTUAL INFORMATION OBJECTIVE

The idea of reducing uncertainty about the environment can be captured in the language of information theory by using mutual information. Specifically, we propose selecting the location of the attention such that its mutual information with the state at the following step is maximized,

$$\max_{l_t} I(s_{t+1}; l_t). \tag{1}$$

Where l_t is the location of the attention window at time t. Eq. 1 can be expanded as

$$\max_{l_t} \mathcal{H}(s_{t+1}) - \mathcal{H}(s_{t+1}|l_t), \tag{2}$$

where $\mathcal{H}$ denotes the entropy. This expansion brings out a very intuitive explanation of the objective. We would like to pick an attention location that maximizes the reduction in entropy (uncertainty) of the environment state after an observation is made.

5.2 CURIOSITY DRIVEN ATTENTION

In this section, we show how maximizing the amount of *surprise* can lead to maximizing the mutual information. Let us begin by further expanding eq. 2.

$$\begin{aligned}
I(s_{t+1}; l_t) &= \mathcal{H}(s_{t+1}) - \mathcal{H}(s_{t+1}|l_t) \\
&= -\sum_{s_{t+1}} p(s_{t+1}) \log p(s_{t+1}) + \sum_{l_t} p(l_t) \sum_{s_{t+1}} p(s_{t+1}|l_t) \log p(s_{t+1}|l_t) \\
&= -\mathop{\mathbb{E}}_{s_{t+1} \sim p(s_{t+1})} [\log p(s_{t+1})] + \sum_{l_t} \sum_{s_{t+1}} p(s_{t+1}, l_t) \log p(s_{t+1}|l_t) \\
&= -\mathop{\mathbb{E}}_{s_{t+1} \sim p(s_{t+1})} [\log p(s_{t+1})] + \mathop{\mathbb{E}}_{s_{t+1}, l_t \sim p(s_{t+1}, l_t)} [\log p(s_{t+1}|l_t)]
\end{aligned}$$

So, maximizing mutual information between the attention location and the environment state is equivalent to minimizing $\log p(s_{t+1})$ and maximizing $\log p(s_{t+1}|l_t)$ under expectation. The first term is the log-likelihood of the next state prior to selecting l_t. Computing it requires marginalization over all possible l_t, which is prohibited by our environment as only a single partial observation from the full state can be provided to the agent. The second term, $\log p(s_{t+1}|l_t)$, is computable at a single attention location from a single state and hence we will focus on maximizing this term.

Now, assume the agent's belief over the true environment state at the next step, s_{t+1}, is represented by a Gaussian with unit variance around the reconstruction, τ_{t+1}. So, $s_{t+1} \sim \mathcal{N}(\tau_{t+1}, \mathcal{I})$ and

$$\begin{aligned} \log p(s_{t+1}|\tau_{t+1}) &\propto \log exp[-(s_{t+1} - \tau_{t+1})^T (s_{t+1} - \tau_{t+1})/2] \\ &= -\sum_i (s^i_{t+1} - \tau^i_{t+1})^2/2 \end{aligned} \tag{3}$$

So, minimizing the squared difference between s_{t+1} and τ_{t+1} leads to maximizing $\log p(s_{t+1}|\tau_{t+1})$. Maximizing $\log p(s_{t+1}|\tau_{t+1})$ in turn means maximizing $\log p(s_{t+1}|l_t)$, since τ_{t+1} is constructed using the deterministic functions W, S, and R once l_t has been picked (see section 4).

In order to minimize the squared difference between s_{t+1} and τ_{t+1}, the agent needs to pick a location l_t that observes the *maximum* error between s_t and its reconstruction τ_t. This is because at step t, if the agent observes the location with the highest reconstruction error, i.e. the *least* likelihood $p(s_t|\tau_t)$, it will ensure that at the next time step $t+1$ the agent has the most recent information on that region and can make a good reconstruction of it. Hence, maximizing $\log p(s_{t+1}|l_t)$ is equivalent to picking an l_t that will lead to the highest reconstruction error at step t.

$$\max_{l_t} \log p(s_{t+1}|l_t) \implies \max_{l_t} \sum_{i \in L_t} (s^i_t - \tau^i_t)^2/2$$

where L_t forms the set of indices under the window at location l_t. Directly optimizing this quantity with respect to l_t is not possible as it can only be computed once an observation has been made and only at a single location. Hence, the glimpse agent must learn to *predict* the location that is most likely to result in high reconstruction error before an observation is made. Our approach is to formulate the glimpse agent as a reinforcement learner with the reconstruction error as its reward.

An interpretation of this objective is that the glimpse agent is surprise seeking, or curiosity driven. It is attending to parts of the state that are novel in the sense that they are difficult for the agent's current model to predict. Another interpretation is that the glimpse agent is acting adversarially to the DMM's model of the environment. At each step the glimpse agent tries to focus on parts of the state that are the most difficult for DMM to reconstruct. By doing so, it is indirectly creating a curriculum of increasingly difficult to model aspects of the environment.

5.3 Full Training Objective

Let us look at a different expansion of the objective in eq. 1,

$$I(s_{t+1}; l_t) = \max_{l_t} \mathcal{H}(l_t) - \mathcal{H}(l_t|s_{t+1}). \tag{4}$$

The first term is $\mathcal{H}(l_t)$, the entropy over the attention location, which is controlled by the policy of the glimpse agent. This term can easily be maximized by standard RL algorithms, such as A2C (Mnih et al., 2016; Wu et al., 2017), using weighted entropy maximization.

Combining the first term from equation 4 and second term from equation 2, the final objective that we use to train the glimpse agent is

$$\max_{l_t} \mathcal{H}(l_t) - \mathcal{H}(s_{t+1}|l_t) \equiv \alpha \mathcal{H}(\pi_{glimpse}) + \max_{l_t} \sum_{i \in L_t} (s^i_t - \tau^i_t)^2/2,$$

where α is the entropy weighting set to 0.001 in our experiments.

Note that this objective is not the mutual information from equation 1. It is possible to design a variational algorithm that maximizes the approximate mutual information based on the expansion in eq. 4 alone (instead of mixing terms from both the expansions). Empirically, however, we found that our objective performs better than the variational approximation to mutual information. See section A.4 for more detail.

6 RESULTS

We evaluate our approach in two environments: a gridworld environment (figure 5) and PhysEnv (figure 1). In both environments, the task is to navigate to a goal while avoiding obstacles and enemies. The agent's actions are movement in the four discrete cardinal directions. In the gridworld, the state is a one-hot image of the objects in the environment (agent, walls, enemies and goal). The agent's and enemies' movements are restricted to the grid squares and the dynamics of the enemies are simple: moving in straight lines until an obstacle is hit in which case it reflects. In PhysEnv, observations are provided as RGB images and the motions of the objects are more varied.

The glimpse agent + DMM are first trained on offline data of state and action trajectories (no reward) collected from each environment. More details on data collection are provided in section A.1. We evaluate this part of the training by how accurately the full state is reconstructed by DMM at each step. Once the glimpse agent + DMM have converged, i.e. the reconstruction loss has plateaued, we freeze their parameters and initialize an agent within the environment. The glimpse agent + DMM serve as fixed components within the agent while it is learns to solve a task. The agent is trained using an off-the-shelf RL algorithm PPO (Schulman et al., 2017) and is evaluated by the total environment reward it collects during testing episodes.

We compare our method against baselines inspired from related work. In the *follow* baseline the attention is always focused on the agent itself. This is a formulation of the partial observability as seen in Parisotto & Salakhutdinov (2017). Next, the *environment* baseline mimics the approach by Mnih et al. (2014), where the environment rewards are used to train the glimpse agent. Finally, the *random* baseline moves the attention randomly at each time step. For the environment task, we have an additional baseline, *full*, where the full state is provided to the agent to solve the task. This is an upper limit for how well any method that only receives partial observations can perform.

6.1 STATE RECONSTRUCTION

	l2 reward (ours)	random	follow	environment
Gridworld	**0.00555**	0.007666	0.07138	0.06348
PhysEnv	**0.0521**	0.0614	0.1186	0.0826

Table 1: Per-pixel L2 loss between agent's reconstruction τ and the ground truth full state s. All methods only see partial observations, but must attempt to reconstruct the full state from internal representation. In both environments, our method has the lowest reconstruction error.

We measure the performance of the glimpse agent + DMM by the average reconstruction error to the true full (unobserved) state in unseen testing episodes. 25 consecutive observations from 6 unseen test episodes are passed through each model and error is averaged over reconstructions at each time step. This gauges the ability of our attention mechanism and memory to reconstruct the full state using only partial observations. Table 1 shows that using surprise driven reward (our approach) for learning attention control leads to a lower reconstruction error than the other baselines. See figures 7, 5, and 6 in the appendix for visualizations of the reconstructions and training error.

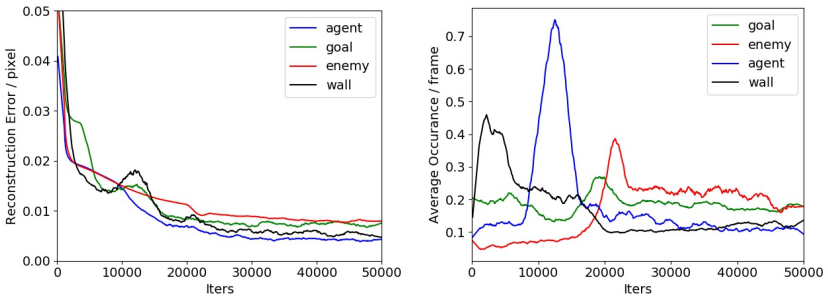


Figure 3: (left) Object-wise breakdown of reconstruction error in gridworld. (right) % of frames in which the object occurs, i.e. the frequency with which the attention is focuses on a particular object.

Figure 3 shows an analysis of what objects are under focus during training in the gridworld environment and their corresponding reconstruction error. The attention forms an interesting curriculum for training the DMM, where it learns to focus on different objects in the environment over time. It starts with walls, then moves to the agent itself, then goal and finally the enemies. Thus, it moves gradually from static to dynamic, harder to model parts of the state, allowing the DMM to learn how to accurately model each object. This very interesting curriculum-like behavior emerges naturally and is not pre-programmed into DMM or the attention agent. It is a consequence of maximizing surprise at every time step. See section A.2 for a more detailed discussion.

6.2 Reinforcement Learning

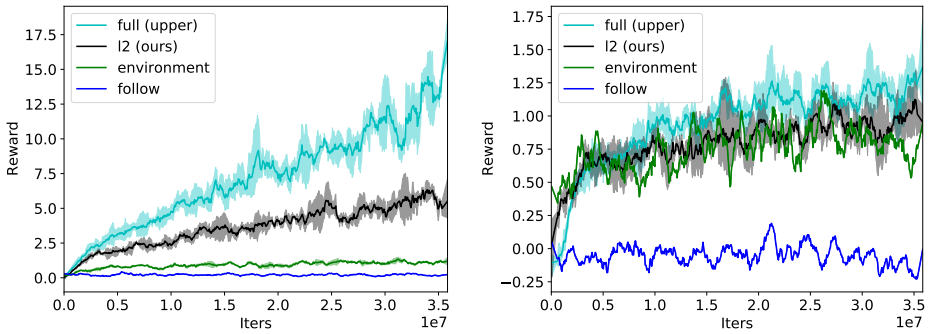


Figure 4: Average testing reward for goal seeking task in PhysEnv (left) and gridworld (right). Comparison between agent with full observations (upper limit) and partial observations trained with surprise (ours), environment rewards and simply following the agent's location.

Figure 4 shows the average episodic reward over 5 testing episodes as the training of the agent progresses. Our method approaches the performance of the upper bound agent that receives full environment states in the gridworld environment. The *environment* baseline performs similarly to our approach in gridworld, but does poorly in PhysEnv, where our approach performs the best. This may be because the attention policy of the *environment* baseline has high entropy and ends up exploring large parts of the state within an episode. But it does not learn to focus on the most unpredictable parts of the state as our method does. The *follow* baseline performs poorly in both environments, likely because it only focuses on the agent and does not explore the rest of the state.

7 Conclusions

We have presented an approach of using mutual information to control a hard attention window in environments with dynamic objects. From partial observations under the attention window, our system is able to reconstruct the full state at every time step with the least error. The representation learned using our method enables RL agents to solve tasks within the environment, while the baseline methods are unable to learn useful representations on the same memory architecture. This demonstrates that attention control plays a large role in enabling task completion.

Note that our attention control objective is independent of the task. It is solely concerned with gathering information from the environment where it is most unpredictable. This is similar to curiosity driven RL that seeks novel states or surprising transitions. Hence, attention control learned in this manner is a generic solution that can be applied to many downstream tasks. It may be able to be combined with task-specific attention control by fine-tuning on a task reward. A better foveal glimpse model may also be used instead of the fixed size hard attention used here. We made an assumption that only one glimpse is allowed per environment state. But the rate of collecting glimpses may be variable with respect to environment speed.

REFERENCES

Joshua Achiam, Harrison Edwards, Dario Amodei, and Pieter Abbeel. Variational option discovery algorithms. *arXiv preprint arXiv:1807.10299*, 2018.

Bogdan Alexe, Nicolas Heess, Yee W Teh, and Vittorio Ferrari. Searching for objects driven by context. In *Advances in Neural Information Processing Systems*, pp. 881–889, 2012.

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.

Marc Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Remi Munos. Unifying count-based exploration and intrinsic motivation. In *Advances in neural information processing systems*, pp. 1471–1479, 2016.

Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysław Debiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, et al. Dota 2 with large scale deep reinforcement learning. *arXiv preprint arXiv:1912.06680*, 2019.

Nicholas J Butko and Javier R Movellan. Optimal scanning for faster object detection. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2751–2758. IEEE, 2009.

Misha Denil, Loris Bazzani, Hugo Larochelle, and Nando de Freitas. Learning where to attend with deep architectures for image tracking. *Neural computation*, 24(8):2151–2184, 2012.

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

Benjamin Eysenbach, Abhishek Gupta, Julian Ibarz, and Sergey Levine. Diversity is all you need: Learning skills without a reward function. *arXiv preprint arXiv:1802.06070*, 2018.

Daniel Freeman, David Ha, and Luke Metz. Learning to predict without looking ahead: World models without forward prediction. In *Advances in Neural Information Processing Systems*, pp. 5379–5390, 2019.

Alex Graves, Greg Wayne, Malcolm Reynolds, Tim Harley, Ivo Danihelka, Agnieszka Grabska-Barwińska, Sergio Gómez Colmenarejo, Edward Grefenstette, Tiago Ramalho, John Agapiou, et al. Hybrid computing using a neural network with dynamic external memory. *Nature*, 538 (7626):471–476, 2016.

Karol Gregor, Danilo Jimenez Rezende, and Daan Wierstra. Variational intrinsic control. *arXiv preprint arXiv:1611.07507*, 2016.

David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018.

Matthew Hausknecht and Peter Stone. Deep recurrent q-learning for partially observable mdps. *arXiv preprint arXiv:1507.06527*, 2015.

Rein Houthooft, Xi Chen, Yan Duan, John Schulman, Filip De Turck, and Pieter Abbeel. Vime: Variational information maximizing exploration. In *Advances in Neural Information Processing Systems*, pp. 1109–1117, 2016.

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.

Shiva Kamkar, Fatemeh Ghezloo, Hamid Abrishami Moghaddam, Ali Borji, and Reza Lashgari. Multiple-target tracking in human and machine vision. *PLOS Computational Biology*, 16(4): e1007698, 2020.

Arbaaz Khan, Clark Zhang, Nikolay Atanasov, Konstantinos Karydis, Vijay Kumar, and Daniel D Lee. Memory augmented control networks. *arXiv preprint arXiv:1709.05706*, 2017.

Hugo Larochelle and Geoffrey E Hinton. Learning to combine foveal glimpses with a third-order boltzmann machine. In J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta (eds.), *Advances in Neural Information Processing Systems 23*, pp. 1243–1251. Curran Associates, Inc., 2010.

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, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

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.

Shakir Mohamed and Danilo Jimenez Rezende. Variational information maximisation for intrinsically motivated reinforcement learning. In *Advances in neural information processing systems*, pp. 2125–2133, 2015.

Lucas Paletta, Gerald Fritz, and Christin Seifert. Q-learning of sequential attention for visual object recognition from informative local descriptors. In *Proceedings of the 22nd international conference on Machine learning*, pp. 649–656, 2005.

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. A possibility for implementing curiosity and boredom in model-building neural controllers. In *Proc. of the international conference on simulation of adaptive behavior: From animals to animats*, pp. 222–227, 1991.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Archit Sharma, Shixiang Gu, Sergey Levine, Vikash Kumar, and Karol Hausman. Dynamics-aware unsupervised discovery of skills. *arXiv preprint arXiv:1907.01657*, 2019.

David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *nature*, 550(7676):354–359, 2017.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008, 2017.

Sean Welleck, Jialin Mao, Kyunghyun Cho, and Zheng Zhang. Saliency-based sequential image attention with multiset prediction. In *Advances in neural information processing systems*, pp. 5173–5183, 2017.

Yuhuai Wu, Elman Mansimov, Roger B Grosse, Shun Liao, and Jimmy Ba. Scalable trust-region method for deep reinforcement learning using kronecker-factored approximation. In *Advances in neural information processing systems*, pp. 5279–5288, 2017.

Yuechen Yu, Yilei Xiong, Weilin Huang, and Matthew R Scott. Deformable siamese attention networks for visual object tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6728–6737, 2020.

Daniel Zoran, Mike Chrzanowski, Po-Sen Huang, Sven Gowal, Alex Mott, and Pushmeet Kohli. Towards robust image classification using sequential attention models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9483–9492, 2020.

A APPENDIX

A.1 DETAILS OF DATA COLLECTION

For gridworld we used 20000 random steps in the environment to create this training set. In PhysEnv we used a previously trained agent to generate 500,000 steps of demonstrations. We experienced that using a random policy in PhysEnv led to quick deaths of the agent with insufficient episode lengths for learning. In both cases, the data are sequences of full states and agent actions from multiple episodes. In PhysEnv, the map size was 21×21 compared to the full state size of 84×84 and in gridworld both the map and state were 10×10. The size of the attention window was fixed to be 3×3 in the gridworld (9% of the full state visible in an observation) and 21×21 in PhysEnv (6.25% of the state visible).

A.2 VISUALIZATIONS OF GLIMPSES DURING TRAINING

We will discuss where the attention focuses and how that effects the reconstruction (and underlying belief) as the training progresses for the gridworld environment. Please refer to 3 for this discussion.

Initially, the attention learns to focus on the *wall* objects and the corresponding reconstruction error for the walls decreases rapidly (3.b). This is because the walls are long objects in the environment, present at multiple pixels and side-by-side, providing the largest and most reliable source of error under the attention window initially. The error is provided as reward to the glimpse agent, hence it seeks more walls in the state.

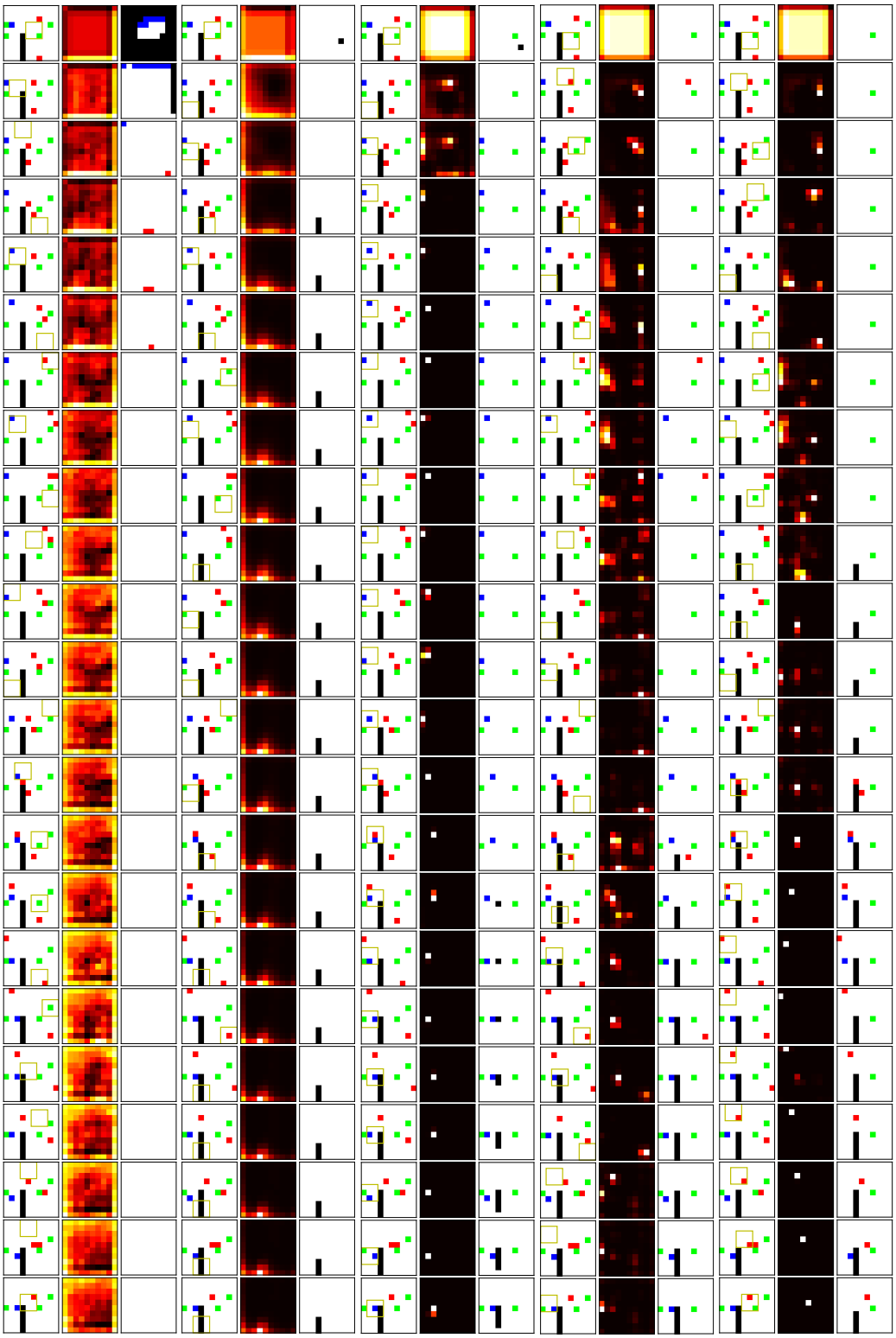
Soon, the error it receives from walls decreases as the agent's DMM learns to persistently record the location of stationary walls over multiple time steps. Then around 10000 iters, the attention switches to focusing on the agent's location itself. This is because the agent's apparent erratic movements provide a larger reward due to their unpredictability than the stationary walls can anymore. This has dual consequences. First, since now the DMM is seeing a lot of examples of the agent's movement, it learns to accurately model the agent based on an initial state and its actions and the reconstruction error corresponding to the agent falls. At its peak, over 70% of glimpses contain the agent within them (figure 3.b). Interestingly, the error corresponding to the wall and goal objects rises as some forgetting of the representation and dynamics of these objects occurs. A little before 20000 iters though, the agent has learnt to represent simultaneously all three objects and it switches attention to the enemies, the only remaining un-modeled objects in the environment. The enemies move predictably, bouncing off walls and other objects, but otherwise in straight lines. The agent does not have direct access to their actions and must infer their motion over multiple time steps using only glimpses.

Figures 5 and 6 visualize the behavior of the glimpse agent during the initial stages of training. Initially, (100 iters), the attention mostly moves around randomly as can be seen by the heatmap being active throughout the state. The reconstruction does not correspond to anything in the real state since the agent has only just begun learning. The glimpse agent here is receiving high reward for pretty much every location in the state. By 2000 iters, the glimpse agent has learnt to mainly focus on the wall once it locates it. This can be seen in the high probabilities of the glimpse agent policy near the wall locations and the attention repeatedly fixating on the wall. The agent is simultaneously striving to learn to represent the walls as they are showing up a lot in its training data, and steadily reducing the reconstruction error (the glimpse agent's reward) coming from walls.

By iter 11000, the focus has shifted to tracking the agent as can be seen in the heatmap. The heatmap shows that the glimpse agent is singularly focused on the agent's movements as the error reward it is getting from the walls has dropped off and the agent's movements are not fully predicatable yet. In the figure for iter 19000, it can be seen that the agent can now track its own location very reliably with an initial position and the actions it is taking, without having to focus the attention on itself. In fact, after step 8 when the agent is discovered, the attention almost never moved back to it and yet the location in the reconstruction perfectly matches that of the unobserved full state. Hence the agent's location based reward source has largely dried up. The heatmap of attention policy shows that the glimpse agent has become interested in the goals and walls again briefly as it learns to represent all three objects together. It can also be seen from the reconstructions here that the agent is unable to accurately track and model the enemies locations at this point.

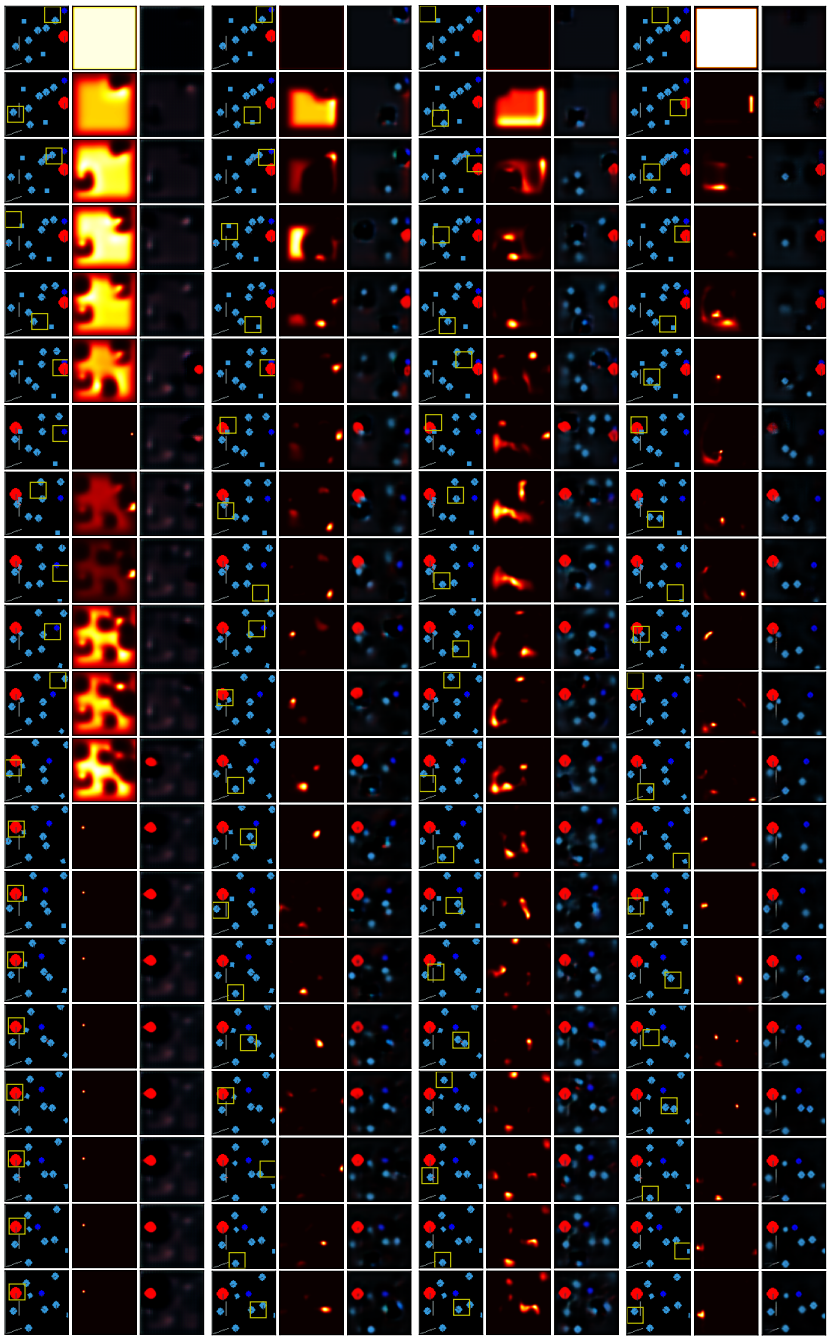
By iter 21000, the attention has shifted focus to the remaining object in the state that is producing a high reconstruction error, i.e. the enemy. Once the enemy is discovered on step 14, the glimpse agent's policy puts all probability near its location so it can keep observing and model its behavior.

The glimpse agent creates a natural curriculum for training the models within DMM, starting at large stationary objects that do not move, then moving to the agent's followed by the objects with the most complex dynamics. We conjecture this automatically discovered curriculum is the reason our system is able to learn better reconstructions than the baselines.



(a) 100 iters (b) 2000 iters (c) 11000 iters (d) 19000 iters (e) 21000 iters

Figure 5: Visualizations of glimpse behavior in gridworld environment. Each sub-figure is the same testing episode at different stages of training, with the rows corresponding to the time within the episode. In each subfigure, the first column shows the full (unobserved) state. The second column is a heatmap showing the glimpse policy, i.e. the likelihood the glimpse agent will select a location for placing attention. The location that was sampled is indicated in the first column by the yellow square. The final, third, column is the reconstruction τ based on the agent's current current belief about the environment.



(a) 1000 iters (b) 7000 iters (c) 10000 iters (d) 200000 iters

Figure 6: Visualizations of glimpse behavior in PhysEnv environment.

A.3 ERROR CURVES FOR GLIMPSE AGENET + DMM

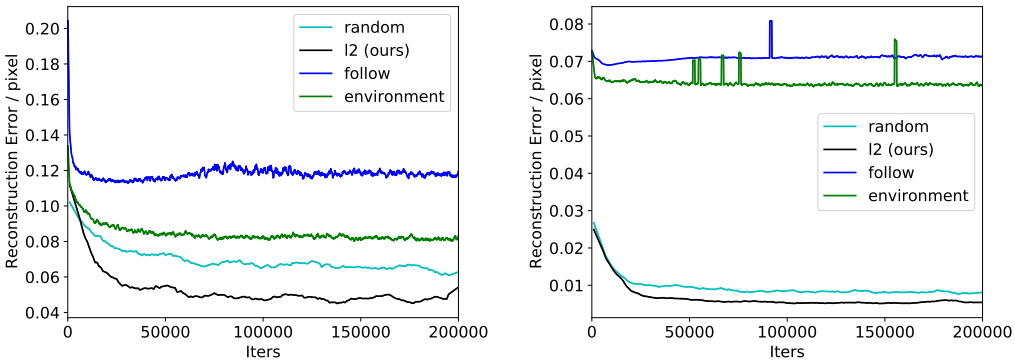


Figure 7: (left) PhysEnv environment. (right) Gridworld environment. Reconstruction error (L2) from the full state during training of the DMM + glimpse agent. All approaches have converged by 200k iters, with our approach having the lowest error, i.e. the most faithful reconstruction of the full unobserved state.

These training curves correspond to the results in table 1. The reconstruction error for each method has stabilized. Our method has the lowest reconstruction error for both environments.

A.4 VARIATIONAL APPROXIMATION TO MUTUAL INFORMATION MAXIMIZATION

The mutual information between attention and environment state as expanded in eq. 4 is

$$I(s_t; l_t) = \mathcal{H}(l_t) - \mathcal{H}(l_t|s_t). \tag{5}$$

In our algorithm, we combine the first term in this expansion with another term in eq. 2. This works well empirically. Theoretically, however, it is computationally possible to maximize both terms in eq. 5, hence maximizing the actual mutual information. However, as we shall show here, this does not work well practically.

The first term can be maximized by using maximum entropy RL algorithms for the attention policy as discussed in the main text. Let us consider the second term.

$$\begin{aligned}
-\mathcal{H}(l_t|s_t) &= -\sum_{s_t} p(s_t)\mathcal{H}(l_t|s_t) \\
&= \sum_{s_t} p(s_t) \sum_{l_t} p(l_t|s_t) \log p(l_t|s_t) \\
&= \mathop{\mathbb{E}}_{s_t, l_t \sim p(s_t, l_t)} [\log p(l_t|s_t)]
\end{aligned}$$

While $\log p(l_t|s_t)$ is not directly known, we can construct a variational function, $q(l_t|s_t)$, to approximate it.

Since $KL(p(l_t|s_t)||q(l_t|s_t)) \geq 0$, we have,

$$\begin{aligned}
\sum_{l_t} p(l_t|s_t) \log p(l_t|s_t) &\geq \sum_{l_t} p(l_t|s_t) \log q(l_t|s_t) \\
\mathop{\mathbb{E}}_{s_t, l_t \sim p(s_t, l_t)} [\log p(l_t|s_t)] &\geq \mathop{\mathbb{E}}_{s_t, l_t \sim p(s_t, l_t)} [\log q(l_t|s_t)]
\end{aligned}$$

Plugging this into eq. 5,

$$I(s_t; l_t) \geq \mathcal{H}(l_t) + \mathop{\mathbb{E}}_{s_t, l_t \sim p(s_t, l_t)} [\log q(l_t|s_t)]$$

Therefore, we can maximize the mutual information by maximizing this lower bound w.r.t. the policy parameters of the glimpse agent. This can be achieved by providing $\log q(l_t|s_t)$ as a reward to the glimpse agent and using RL algorithms to maximize its long term value along with the policy entropy. As such, maximizing the mutual information.

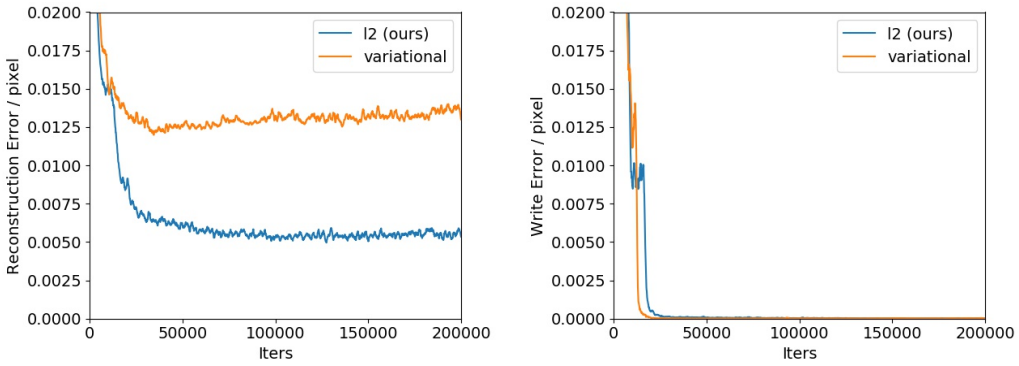
We can additionally condition q on the internal state and action of the agent at previous time step. Since we do not have access to the full environment state, s_t, we substitute it with the agent's internal state. The final variational function looks like $q(l_t|\mu_t, \mu_{t-1}, a_{t-1})$. q must be trained to match the true posterior $p(l_t|s_t)$ to provide accurate rewards to the glimpse agent.

The question now is how to learn q? In this work, it is a neural network trained on samples. This is done by minimizing the KL divergence,

$$\min \mathop{\mathbb{E}}_{s_t, l_t \sim p(s_t, l_t)} [KL(p(l_t|s_t)||q(l_t|\mu_t, \mu_{t-1}, a_{t-1}))]$$

on samples from the environment and the glimpse agent's policy. This will ensure q is likely to assign high likelihood to actual locations conditioned on its inputs.

Intuitively, q will be high at locations of attention that are easy to identify by looking at the previous state, the resulting state, and the agent action. Otherwise, its output will be flat and hard to predict. Since q is provided to the glimpse agent as reward, it will be rewarded for selecting easily identifiable locations and prefer such a policy. These can be locations that produce a high amount of localized surprising information in μ_t as compared to μ_{t-1}, as unexpected appearance of goal or enemies somewhere will signal that the attention was just moved there, hence producing a high likelihood in q. Conversely, low reward locations will be ones that are hard to identify by q as they do not impact any specific part of μ_t, leaving it largely the same as μ_{t-1}, or observe a location that was already well-modeled by the agent and hence do not leave an identifiable *mark* on μ_t that is a tell-tale sign for q.



(a) Overall state reconstruction error per pixel.

(b) Write error of observation under attention window.

Figure 8: State reconstruction and observation reconstruction errors in gridworld environment. Observation reconstruction error falls to almost zero for both methods while full state reconstruction remains high for variational approach.

The above method does not work well in practice. Figure 8a shows the reconstruction error of the unobserved state with our proposed method and the variational approach. Figure 8b shows the observation reconstruction error, which is the error in reproducing just the observation under the attention immediately after it is made. It can be seen that the latter quantity goes to near zero for both methods, meaning that immediate observations are being faithfully recorded into memory and reconstructed. Yet, the overall state reconstruction error remains high for the variational approach, which means that the agent is not able to retain observations over time steps as the attention moves around the state and the environment model is not learning to predict the evolution of the state over time without direct observation.

A reason why the variational approach fails in practice could be that we do not have direct access to $p(l_t|s_t)$ in order to train q. Instead we must rely on the agent's internal estimate of the environment state, μ_t, which itself is being trained alongside the glimpse agent. This results in noisy rewards to the glimpse agent as the q function is not able to accurately predict where the attention is. As we have seen in section A.2, attention control creates a curriculum for training the agent's state estimate. Hence this forms a loop, where attention control is dependent on agent's state estimate which is in return dependent on the curriculum generated by the attention.

Extensive hyperparameter tuning may also be required for this approach to work as the glimpse agent can easily default to chosing the same attention location at each time step as it ensures the highest $q(l_t|s_t)$.