2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 23-27, 2022, Kyoto, Japan

BIMRL: Brain Inspired Meta Reinforcement Learning

Seyed Roozbeh Razavi Rohani [1], Saeed Hedayatian [2], Mahdieh Soleymani Baghshah [1]

2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) | 978-1-6654-7927-1/22/$31.00 ©2022 IEEE | DOI: 10.1109/IROS47612.2022.9981250

Abstract— **Sample efficiency has been a key issue in reinforcement learning (RL). An efficient agent must be able to leverage its prior experiences to quickly adapt to similar, but new tasks and situations. Meta-RL is one attempt at formalizing and addressing this issue. Inspired by recent progress in meta-RL, we introduce BIMRL, a novel multi-layer architecture along with a novel brain-inspired memory module that will help agents quickly adapt to new tasks within a few episodes. We also utilize this memory module to design a novel intrinsic reward that will guide the agent's exploration. Our architecture is inspired by findings in cognitive neuroscience and is compatible with the knowledge on connectivity and functionality of different regions in the brain. We empirically validate the effectiveness of our proposed method by competing with or surpassing the performance of some strong baselines on multiple MiniGrid environments.**

I. INTRODUCTION

A major problem with most model-free RL methods is their sample inefficiency and poor generalizability. Even for simple tasks, these methods require thousands of interactions with the environment. Furthermore, even after learning a task, slight changes in the environment dynamics or the underlaying task will force us to basically retrain the agent from scratch. To combat these issues, a number of methods have been proposed that utilize ideas from multi-task and meta learning ([1], [2]). Some of these methods treat the task-ID like an unobservable latent variable in the underlaying POMDP and condition their policy on a learned, task belief state [1]. However, these methods generally use some simplifying assumptions that limits their generalizability. Our proposed method, BIMRL, is similar in that it learns to identify a suitable task-specific latent and uses it to quickly adapt, but relaxes some of these assumptions. Similar to some prior works ([1], [3]), we use variational methods to estimate this latent variable. However, by introducing a new factorization of the log-likelihood of trajectories, our method learns a different context embedding that is more consistent with POMDP assumptions and can also be interpreted in a more meaningful way.

Another central issue with RL is the well-known exploration-exploitation dilemma [1]. If an agent wants to quickly adapt to a new task, it must be able to explore the environment and obtain relevant experiences. Nonetheless, with complex tasks and environments, conventional exploration methods, such as epsilon-greedy exploration, are not enough. Intrinsic rewards that encourage suitable exploration are currently one of the most successful approaches. Our novel memory module provides an intrinsic reward to guide the exploration; thus alleviating the problems that arise from exploration across different tasks. Aside from this reward our memory module, which consists of an episodic part that resets after each episode, and a Hebbian part ([4]) which retains information across different episodes of a particular task, gives us a performance boost in memory-based tasks while helping deal with catastrophic forgetting that comes up in multi-task settings.

To summarize, our contributions are as follows:

1) Proposing a multi-layer architecture that bridges model-based and model-free approaches by utilizing predictive information on state-values. Each layer of our architecture corresponds to a different part of the Prefrontal Cortex (PFC) which is known to be responsible for important cognitive functionalities such as learning a world-model [5], planning [6], and shaping exploratory behaviours [5]. One of the objectives in this architecture is derived from our newly proposed factorization of the log-likelihood of trajectories. This factorization is more robust to violations of POMDP assumptions and also takes the predictive coding ability of the brain into consideration [7].
2) Introducing a neuro-inspired memory module compatible with discoveries in cognitive neuroscience. This module will help in memory-based tasks and give the agent the ability to preserve information, both within an episode and across different episodes of some task. The design of this memory module is reminiscent of Hippocampus (HP) as the main region assigned to episodic memory in the brain [8].
3) Proposing a new intrinsic reward based on our memory module. This reward does not disappear over time, is task-agnostic, and is very effective in a multi-task setting. It will encourage exploratory behaviour that leads to faster task identification.

II. RELATED WORK

A. Meta Reinforcement Learning

Meta reinforcement learning is a promising approach for tackling few-episode learning regimes. It aims to achieve quick adaptation by learning inductive biases in the form of meta-parameters. There are multiple approaches to meta-RL, two of the most popular being gradient based methods and metric based methods. Another popular family of approaches use a recurrent neural network (RNN) [2]. Some previous works try to extend this approach and use an RNN to extract

[1] Department of Computer Engineering Sharif University of Technology
razavii@ce.sharif.edu, soleymani@sharif.edu
[2] Department of Mathematical Sciences Sharif University of Technology
hedayatians@gmail.com

some context that will be used to inform the policy. Recently, by observing the effectiveness of variational approaches for estimating the log-likelihood of trajectories, some methods proposed to use the estimated context (referred to as the belief state in the literature) to inform the policy about its current task ([1], [3]). This belief state should contain information about environment dynamics as well as the reward function of the task. Ideally, conditioning on this belief would convert the POMDP into a regular MDP. However, these methods typically impose some constraints on the underlaying POMDP (for instance, [1], [3] consider Bayes-adaptive MDPs which are a special case of the general POMDPs). These restrictions may be violated in some scenarios. On the contrary, our proposed method is more robust to such violations.

B. Modular Architectures

Injecting a modular inductive bias into policies is a promising strategy for improving out-of-distribution generalization in a systematic way. [9] proposed a modular structure where each module can become an expert on a different part of the state space. These modules will be dynamically combined through attention mechanisms to form the overall policy. [10] introduced an extension called the bidirectional recurrent independent mechanism (BRIMs). BRIM is composed of multiple layers of recurrent neural networks where each module also receives information from the upper layer. A modified version of BRIM is used in our architecture as well.

C. Combining Model-Based & Model-Free

There has been efforts on trying to combine model-based and model-free approaches. Dyna [11] and successor representation [12] are two such attempts. There has also been some investigations on whether our brains work in a model-based or model-free manner, which has resulted in some neuro-inspired computational models ([13]). We propose a new way of combining these two approaches that is inspired by the predictive capabilities of PFC [14].

D. Exploration

Another central issue in RL is efficient exploration of the environment. Methods that encourage effective exploratory behaviour through the use of an intrinsic reward are among the most popular and effective ways of dealing with this problem. Intrinsic rewards can be obtained from a novelty or curiosity criterion ([15]). It is worth noting that effective exploration becomes an even bigger issue in the meta-learning setting ([3]).

III. METHODS

In this chapter we introduce our proposed method. We will first derive a factorization for a trajectory's log-likelihood. Using this factorization, we will design a multi-layered architecture with multiple loss functions. We will then shift our focus and introduce our memory module. Finally, we will give some intuitive interpretations of our proposed method and mention some of the connections to different regions of the human nervous system.

A. Likelihood Factorization

Our method builds upon VariBAD [1] and we use the same terminology as is used there. Similar to VariBAD, we aim to optimize the following ELBO loss corresponding to log-likelihood of a trajectory τ:

$$\begin{aligned}\mathrm{ELBO}_t = \mathbb{E}_\rho \Big[&\mathbb{E}_{q_\phi(m|\tau_{:t})}\left[\log p_\theta\left(\tau_{:H^+} \mid m\right)\right] \\ &-KL\left(q_\phi\left(m \mid \tau_{:t}\right) \| p_\theta(m)\right)\Big],\end{aligned} \tag{1}$$

where ρ is the trajectory distribution induced by our policy, m is the belief state, and H^+ is the time horizon for a task, which is set to be four times the horizon for an episode. Notice that this objective is comprised of two parts. The first part, $\mathbb{E}_{q_\phi(m|\tau_{:t})}\left[\log p_\theta\left(\tau_{:H^+} \mid m\right)\right]$, is known as the reconstruction loss and the second part is the KL-divergence between the variational posterior q_ϕ and the prior over the belief state, $p_\theta(m)$. Unlike VariBAD, we factorize the reconstruction loss as follows:

$$\begin{aligned}\log p\left(\tau_{:H^+} \mid m, a_{0,\cdots,n}\right) &= \log p\left(s_0 \mid m\right) \\ &+ \sum_{i=0}^{H^+-1} \sum_{j=0}^{\min(n,i)} \left[\log p\left(s_{i+1} \mid s_{i-j}, a_{i-j,\cdots,i}, m\right)\right] \\ &+ \sum_{i=0}^{H^+-1} \sum_{j=0}^{\min(n,i)} \left[\log p\left(r_{i+1} \mid s_{i-j}, a_{i-j,\cdots,i}, m\right)\right] \\ &+ \sum_{i=0}^{H^+-1} \sum_{j=0}^{\min(n,i)} \left[\log p\left(a_{i+1} \mid s_{i-j,\cdots,i+1}, m\right)\right]\end{aligned} \tag{2}$$

It is worth mentioning that unlike previous methods which use the Bayes Adaptive Markov Decision Process (BAMDP) formulation and assume perfect observability given the belief state (which will not hold in extreme partial observability scenarios), our factorization uses the predictive information in m and asserts that our belief should contain enough information to predict the next n states and rewards, given the next n actions. Contrary to VariBAD that conditions the trajectory on the full history of actions, we condition a trajectory on only the first n actions (hence, the added fourth summation term in 2). This modification will result in an added action simulation network which will also help in learning better representations in the first level of the hierarchical structure. This predictive coding capability is compatible with those observed in the PFC ([7]).

B. Brain Inspired Meta Reinforcement Learning (BIMRL)

Our proposed architecture is composed of a recurrent task inference module responsible for updating our estimate of belief state about the current task, followed by a hierarchical structure similar to BRIM [10], that conditions on the inferred belief state. As we will see, this hierarchical structure is similar to PFC in how they function. The last layer of this hierarchical structure is a controller whose hidden state will be fed into an actor-critic network. We also have a memory module that directly interacts with the controller. Figure 1

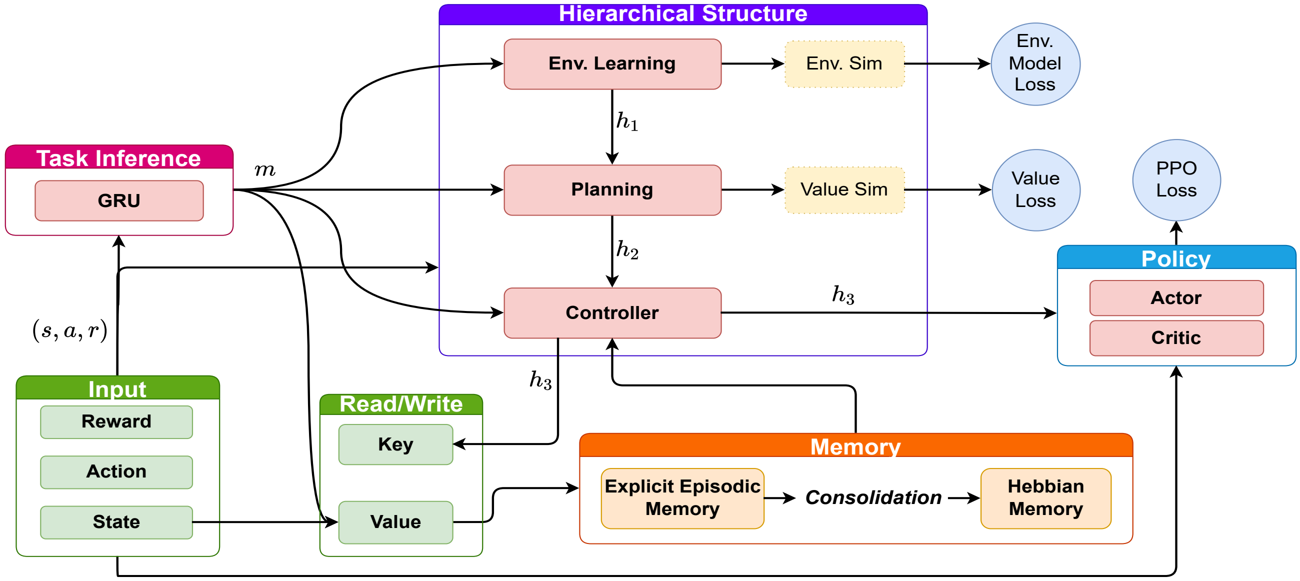
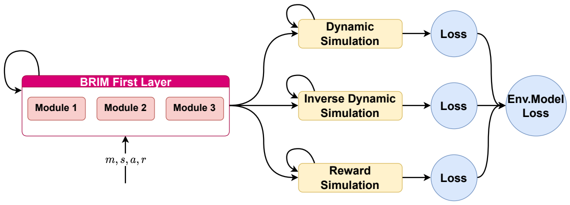


Fig. 1: A bird's-eye view of our model. The major components of BIMRL include the task inference module, the multi-layer hierarchical structure and the memory module.

demonstrates how all these modules are put together. In what follows, we will briefly introduce each part[1].

1) Hierarchical Structure (BRIM): The task inference module is a recurrent network that processes the last (observation, action, reward) and produces a belief state that will be passed to each of the three layers in the hierarchical structure as well as the memory module.

The first level of BRIM (as shown in Fig. 1) is responsible for learning a world model. This layer uses the last observation, action and reward in addition to the current belief state to predict dynamics, inverse dynamics and the rewards. It does so using three simulation networks. This layer is trained to maximize the trajectories log-likelihood as is formulated in 2. Figure 2 illustrates this layer in more detail.



,

Fig. 2: First level of the hierarchical structure. Responsible for learning a world-model.

The second level (the "Planning" box in Fig. 1) is responsible for predicting the value of the next n steps. It will take in the previous level's hidden state (h_1) as well as the current state and the next n actions, and predicts the value for the next n time-steps. It aims to minimize the following loss function:

$$Loss_{\text{layer2}} = \sum_{i=0}^{H^+-1} \sum_{j=0}^{n} \left[\left\| V_\psi\left(s_i, a_{i,\cdots i+j}, h_2\right) - G_{i+j+1:i+j+1+k} \right\|^2 \right], \tag{3}$$

where $G_{t:t+k}$ is the k-step TD return defined as

$$G_{t:t+k} = R_{t+1} + \gamma R_{t+2} + \cdots + \gamma^{k-1} R_{t+k} + \gamma^k V_\theta^\pi\left(S_{t+k}\right). \tag{4}$$

In the above equation, ψ denotes the parameters of n-step value decoder network, h_2 is the output of the second level and θ shows the parameters of the critic head in the actor-critic network.

The existence of this layer between the upper layer, responsible for learning the model of the environment, and the lower layer, which directly affects decision making, allows the following interpretations:

- **Model-based approach:** Model-based methods first obtain a model of the environment and then plan their next action using this model. In our proposed architecture, the first layer aims to learn as much as possible about the environment. The second layer receives this information and predicts the value for the next n steps. To do so accurately, this layer must be able to perform some sort of planning. Therefore, in our model the two explicit stages in model-based approaches are performed implicitly by the networks in the first and second layer.
- **Context-based RL:** Methods such as VariBAD inform the agent of its task by providing it with a context vector obtained from a dynamics prediction network. The context vector that we provide is even more informative as it also contains the necessary information for predicting the value of states in the next several steps.

[1] More details about the architectures and hyperparameters that were used as well as the code can be found here

Finally, the third level of BRIM, known as the controller, aggregates the information form the previous BRIM layer and the most recent observations form the environment. Additionally, the hidden state of this layer modulates access to the memory, which will be discussed in the next section. The aggregated information will then be used by shallow actor and critic networks to determine the next action and value. We use PPO [16], an on-policy learning algorithm, to train our policy networks.

This hierarchical structure is reminiscent of certain parts of the brain. The first layer is similar to the medial Prefrontal Cortex (mPFC) as they are both responsible for learning a predictive model of the environment [7]. Similarly, the second layer corresponds to the Obitofrontal Cortex (OFC) which, in some neuroscientific literature, is identified as the main region responsible for multi-step planning in the brain [6]. Lastly, a part of the nervous system that connects PFC to HP (episodic memory) and the Striatum (which corresponds to actor-critic networks [17]) is known as the Anterior Cingulate Cortex (ACC) [18]. This region manages the use of control or habitual behaviours based on the degree of uncertainty about the task. It is also known as one of the pathways for transferring information from PFC to HP to guide the transition between the working memory (corresponding to PFC [19]) and the episodic memory [8]. In our model this corresponds to the third layer which modulates the memory and aggregates information in time [18].

2) Memory: The memory module is itself composed of two parts: an explicit episodic memory and a Hebbian memory. At each time step, the output of these two parts are combined using an attention mechanism and is then sent to the controller (see Fig. 3). The hidden state of the controller is used as the query for this attention mechanism and weights the contribution of each module.

By concatenating the embedded observation of the agent along with the output of the task inference module at each time-step, we create the "event key" and the hidden state of the controller is used as the "event value". In what follows, we will reference these event keys and values.

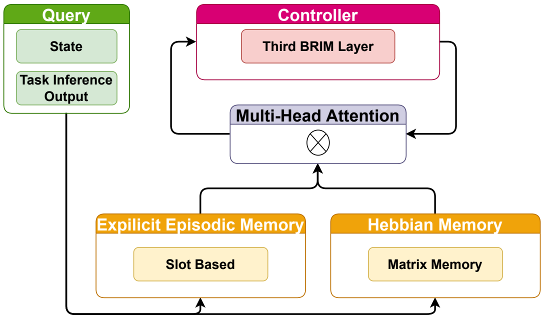


Fig. 3: The interaction between the controller and the memory module

Explicit memory is a slot-based memory that stores the event keys and values. At each time-step, the hidden state of the last level of BRIM is used as the query and the explicit memory is accessed using a multi-head attention mechanism. At the end of each episode, we compute the normalized "reference time" for each slot, which indicates how often each stored event was called upon by the incoming queries, on average. Equation (5) shows how it is calculated.

$$r_i = \frac{\sum_{t=s_i}^{H} W_i^t}{H - s_i} \tag{5}$$

In above equation r_i is the reference time for the i-th slot, s_i is the time when this slot was added to the memory, W_i^t is attention weight attributed to the i-th slot at time t and H is the episode length. As we will see, the reference time is needed for updating the Hebbian memory. Figure 4 demonstrates how the episodic memory works.

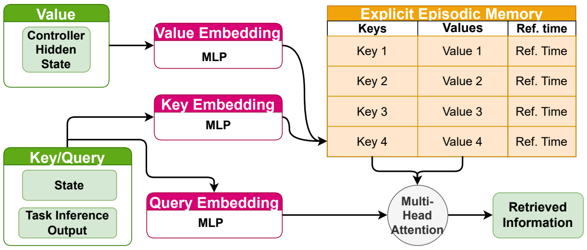


Fig. 4: Episodic memory

The Hebbian memory works on the time scale of tasks, i.e., it won't be reset after each episode. Once an episode is completed, the top k percent of slots with the largest reference times are transferred to the Hebbian memory through the process of memory consolidation [20]. These transferred slots will update the Hebbain memory using the Hebbian learning rule. Equation (6) shows a general form of this learning rule:

$$\Delta W_{kl}^{\text{assoc}} = \gamma_{+} \left(w^{\max} - W_{kl}^{\text{assoc}}\right) v_{t,k}^{s} k_{t,l}^{s} - \gamma_{-} W_{kl}^{\text{assoc}} {k_{t,l}^{s}}^{2} \tag{6}$$

Here, γ_+ is the correlation coefficient, γ_- is the regularization coefficient, $w^{\max}$ imposes a soft constraint on the maximum connection weight, and W_{kl}^{assoc} denotes the parameters of the Hebbian network. As can be seen in the above equation, this learning rule consists of multiple meta-parameters that control the plasticity of synaptic connections. Inspired by the phenomenon of meta-plasticity ([21]), a meta-learning mechanism is used so that these meta-parameters can be learned. Using the Hebbian learning rule as learning mechanism in the inner loop will also circumvent the issue of facing a second order optimization problem.

This transfer of information from a dynamic memory that grows into a fixed-size memory can also help with the decontextualization of the event [22].

3) Intrinsic Reward: Our proposed intrinsic reward is added to the sparse extrinsic reward of the environment to encourage exploration, both on episodic and task-level time scales. This reward is obtained by multiplying two components. The first component, $r_t^{\text{curiosity}}$, is a curiosity based reward that encourages visiting surprising states. It is a convex combination of reward, state, and action prediction errors. This reward is scaled using a second component, α_t. We can think of this second part as a factor that adjusts for the *newness* of the current observation. It is defined as the distance between the most recent observation and the k-th nearest event stored in the episodic memory. If the agent encounters an observation that is unlike what has been seen in the on-going episode, this coefficient will be large, resulting in a high intrinsic reward. The following equations fully define the intrinsic reward:

$$r_t^{\text{intrinsic}}\left(r_{t-1}, s_{t-1}, a_{t-1}, m\right) = \alpha_t r_t^{\text{curiosity}}, \tag{7}$$

where

$$\alpha_t = \left\|x_t - \mathrm{NN}_{k,X}\left(x_t\right)\right\|,$$

$$\begin{aligned} r_t^{\text{curiosity}} = -\mathbb{E}_{q_\phi(m|\tau_{:t})}\big[&\lambda_{\text{reward}} \times \log p_\theta\left(r_t \mid s_{t-1}, a_{t-1}, s_t, m\right) \\ &+\lambda_{\text{state}} \times \log p_\theta\left(s_t \mid s_{t-1}, a_{t-1}, m\right) \\ &+\lambda_{\text{action}} \times \log p_\theta(a_t \mid s_t, s_{t-1}, m)\big], \end{aligned}$$

$$\lambda_{\text{state}} + \lambda_{\text{action}} + \lambda_{\text{reward}} = 1.$$

Note that $\mathrm{NN}_{k,X}$ is the k-th nearest event to the current one, among the records that are stored in the episodic memory, X.

IV. EXPERIMENTS

A. Experimental Setup

For evaluation, we used MiniGrid [23], a set of procedurally-generated, partially-observable environments in which an agent can interact with multiple objects. At each time-step, the agent receives its observation in the form of a $7 \times 7 \times 3$ tensor and can perform any of the available 7 actions (moving in different directions, picking up objects, etc.). In the meta-training, the agent will see four episodes of each task and has to learn to adapt and solve the task within the time-span of those four episodes. At test time, we report the performance on the last (fourth) episode of each task. We used four sets of tasks: MultiRoom-N4-S5, MultiRoom-N6, KeyCorridorS3R1, and KeyCorridorS3R2. In the KeyCorridor set of tasks, the agent has to pick-up an object that is behind a locked door. The key is hidden in another room and the agent has to find it. The MultiRoom set of tasks provide an environment with a series of connected rooms with doors between them. The goal is to reach a green square in the final room.

To compare our results, we used a number of baselines: (`VariBAD`) [1], (`HyperX`) [3], and (`Rl`^{2}) [2]. We also proposed a new strong baseline by augmenting VariBAD with BeBold [24] (`VariBAD+BeBold`), a recently introduced intrinsic reward that improves the performance in sparse-reward environments.

B. Results

Fig. 5 plots the average return of our model as well as those of our baselines. As can be seen, some of the baselines do not get any reward in the more challenging tasks. This is because those tasks have larger state-spaces and require performing several actions in a particular order. This makes them difficult to solve for methods without a suitable exploration strategy.

In all four sets of tasks, our method achieves the best performance while converging faster and observing fewer number of frames. In MultiRoom-N6, one of the more difficult tasks, our method outperforms VariBAD+BeBold by a significant margin. In this task, other baselines fail to get any reward even after training for more than 2M frames.

C. Ablation study

To investigate the significance of each part of our proposed architecture, we evaluated the performance of three ablated versions of our model on the MultiRoom-N4-S5 set of tasks. We examined a model without the memory module (`BIMRL w/o Mem`), one in which the second level of hierarchical structure (responsible for n-step value prediction) was removed (`BIMRL w/o Value pred`), and one with the vanilla VariBAD trajectory factorization instead of our proposed factorization (`BIMRL w/o N step pred`). The result of this ablation study is depicted in Fig. 6. It can be seen that all ablated versions are less sample-efficient, indicating the usefulness of different parts of our model. Most notably, the exclusion of the memory module significantly reduces the convergence rate.

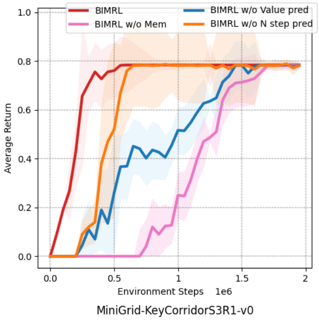


Fig. 6: Performance of ablated versions against that of the full model

V. CONCLUSIONS

We introduced BIMRL, a novel, modular RL agent inspired by the human brain. BIMRL induces several useful inductive biases which helps it meta-learn patterns across different tasks and to adapt quickly to new ones. It emphasizes the significance of taking inspiration from biological systems in designing artificial agents. For future work, one could investigate adding another layer to the memory, in the form of a life-long generative memory module. Additionally, more extensive experiments on tasks other than those in MiniGrid, particularly memory-based tasks, would shed more light on the extent to which such multi-layered architectures can

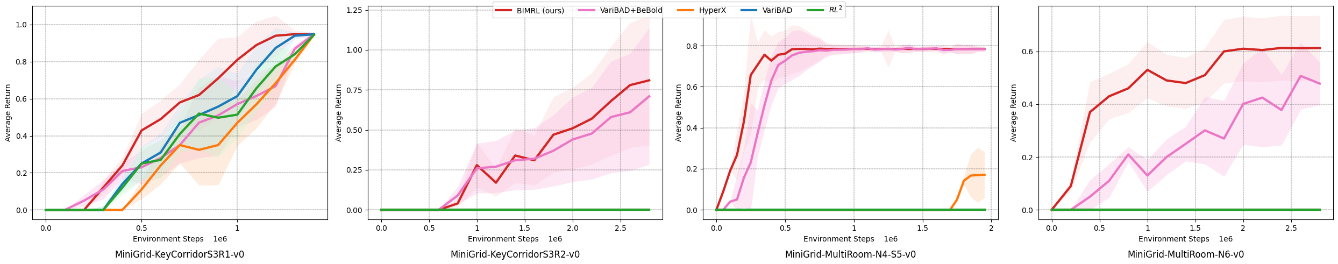


Fig. 5: The average return of our proposed model and the baselines in four sets of tasks

help with fast adaptation to new situations. Furthermore, extending our architecture with new modules that make use of textual instructions would allow the agent to test its adaptation capabilities on a much larger set of tasks. This provides yet another avenue for future research.

References

[1] L. Zintgraf, K. Shiarlis, M. Igl, S. Schulze, Y. Gal, K. Hofmann, and S. Whiteson, "Varibad: A very good method for bayes-adaptive deep rl via meta-learning," *arXiv preprint arXiv:1910.08348*, 2019.

[2] J. X. Wang, Z. Kurth-Nelson, D. Tirumala, H. Soyer, J. Z. Leibo, R. Munos, C. Blundell, D. Kumaran, and M. Botvinick, "Learning to reinforcement learn," *arXiv preprint arXiv:1611.05763*, 2016.

[3] L. M. Zintgraf, L. Feng, C. Lu, M. Igl, K. Hartikainen, K. Hofmann, and S. Whiteson, "Exploration in approximate hyper-state space for meta reinforcement learning," in *International Conference on Machine Learning*. PMLR, 2021, pp. 12 991–13 001.

[4] T. Limbacher and R. Legenstein, "H-mem: Harnessing synaptic plasticity with hebbian memory networks," *bioRxiv*, 2020.

[5] J. Russin, R. C. O'Reilly, and Y. Bengio, "Deep learning needs a prefrontal cortex," *Work Bridging AI Cogn Sci*, vol. 107, pp. 603–616, 2020.

[6] K. J. Miller and S. J. C. Venditto, "Multi-step planning in the brain," *Current Opinion in Behavioral Sciences*, vol. 38, pp. 29–39, 2021.

[7] W. H. Alexander and J. W. Brown, "Frontal cortex function as derived from hierarchical predictive coding," *Scientific reports*, vol. 8, no. 1, pp. 1–11, 2018.

[8] H. Eichenbaum, "Prefrontal–hippocampal interactions in episodic memory," *Nature Reviews Neuroscience*, vol. 18, no. 9, pp. 547–558, 2017.

[9] A. Goyal, A. Lamb, J. Hoffmann, S. Sodhani, S. Levine, Y. Bengio, and B. Schölkopf, "Recurrent independent mechanisms," *arXiv preprint arXiv:1909.10893*, 2019.

[10] S. Mittal, A. Lamb, A. Goyal, V. Voleti, M. Shanahan, G. Lajoie, M. Mozer, and Y. Bengio, "Learning to combine top-down and bottom-up signals in recurrent neural networks with attention over modules," in *International Conference on Machine Learning*. PMLR, 2020, pp. 6972–6986.

[11] R. S. Sutton, "Dyna, an integrated architecture for learning, planning, and reacting," *ACM Sigart Bulletin*, vol. 2, no. 4, pp. 160–163, 1991.

[12] I. Momennejad, E. M. Russek, J. H. Cheong, M. M. Botvinick, N. D. Daw, and S. J. Gershman, "The successor representation in human reinforcement learning," *Nature human behaviour*, vol. 1, no. 9, pp. 680–692, 2017.

[13] D. Kim, J. H. Lee, J. H. Shin, M. A. Yang, and S. W. Lee, "On the reliability and generalizability of brain-inspired reinforcement learning algorithms," *arXiv preprint arXiv:2007.04578*, 2020.

[14] E. M. Russek, I. Momennejad, M. M. Botvinick, S. J. Gershman, and N. D. Daw, "Predictive representations can link model-based reinforcement learning to model-free mechanisms," *PLoS computational biology*, vol. 13, no. 9, p. e1005768, 2017.

[15] D. Pathak, P. Agrawal, A. A. Efros, and T. Darrell, "Curiosity-driven exploration by self-supervised prediction," in *International conference on machine learning*. PMLR, 2017, pp. 2778–2787.

[16] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[17] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[18] I. E. Monosov, "How outcome uncertainty mediates attention, learning, and decision-making," *Trends in neurosciences*, vol. 43, no. 10, pp. 795–809, 2020.

[19] J. X. Wang, Z. Kurth-Nelson, D. Kumaran, D. Tirumala, H. Soyer, J. Z. Leibo, D. Hassabis, and M. Botvinick, "Prefrontal cortex as a meta-reinforcement learning system," *Nature neuroscience*, vol. 21, no. 6, pp. 860–868, 2018.

[20] E. Camina and F. Güell, "The neuroanatomical, neurophysiological and psychological basis of memory: Current models and their origins," *Frontiers in pharmacology*, vol. 8, p. 438, 2017.

[21] S. Farashahi, C. H. Donahue, P. Khorsand, H. Seo, D. Lee, and A. Soltani, "Metaplasticity as a neural substrate for adaptive learning and choice under uncertainty," *Neuron*, vol. 94, no. 2, pp. 401–414, 2017.

[22] M. C. Duff, N. V. Covington, C. Hilverman, and N. J. Cohen, "Semantic memory and the hippocampus: revisiting, reaffirming, and extending the reach of their critical relationship," *Frontiers in Human Neuroscience*, vol. 13, p. 471, 2020.

[23] M. Chevalier-Boisvert, L. Willems, and S. Pal, "Minimalistic gridworld environment for openai gym," https://github.com/maximecb/gym-minigrid, 2018.

[24] T. Zhang, H. Xu, X. Wang, Y. Wu, K. Keutzer, J. E. Gonzalez, and Y. Tian, "Bebold: Exploration beyond the boundary of explored regions," *arXiv preprint arXiv:2012.08621*, 2020.