Optimisitc Initialization of Non-linear Parameterized Value Functions

Alireza Azimi [*1] **Haruto Tanaka** [*1] **Mashfiq Shahriar Zaman** [*1] **Henry Du** [*1]

Abstract

Optimistic initialization of value functions (Sutton & Barto, 2020, §2.6) is a widely used exploration approach in tabular reinforcement learning (RL). However, optimistically, initializing non-linear value functions is not as trivial as in the tabular setting. This study investigates the effect of shifting reward signals (Machado et al., 2015) in non-linear parameterized value functions and compares our results to the tabular setting.

1. Introduction

Balancing exploration and exploitation is a fundamental challenge encountered by RL algorithms (Sutton & Barto, 2020, §2.6). Achieving a careful equilibrium between exploration and exploitation helps the policy reach a more optimal performance, whereas solely depending on exploitation could result in sub-optimal policies. Exploration is an essential section of RL systems, and previous research has proposed several exploration methods, including optimistic initialization (Sutton & Barto, 2020, §2.6), ϵ-greedy (Sutton & Barto, 2020, §2.2), and UCB1 (Auer et al., 2002).

The straightforwardness of optimistic initialization as an exploration strategy has contributed significantly to its widespread acceptance. By setting the initial values greater than the reward maxima, this method promotes substantial exploration at the early stages of learning. This concept of optimism has frequently been applied in the context of tabular and linear function approximation settings. For instance, Brafman & Tennenholtz (2003) introduced an adversary algorithm named *R-max*, which optimizes the agent in a tabular setting for an optimistically initialized fictitious model of the environment. As a similar approach, a dynamic programming-based method called *Optimistic Initial Model* (OIM) has been proposed by Szita & Lőrincz (2008). For the linear function approximation setting, Machado et al. (2015) successfully implemented the optimistic initialization of a linear function approximator by normalizing and shifting reward signals.

Despite its popularity in tabular and linear function approximation, deep RL algorithms are rarely employed with optimistic initialization (Rashid et al., 2020). A particular issue involves the initialization of neural networks. Initiating neural networks with sporadic levels of optimism could impede the rate of learning if the amount of the reward signals is unknown. Furthermore, one cannot simply initialize a neural network to a specific value. Another difficulty arises in updating the neural networks during training. Generalization is an inherent characteristic of all neural networks. Uncontrolled generalization in an update can quickly change the initialization, rendering any optimistic initialization ineffective (Rashid et al., 2020).

In this work, we mitigate these difficulties in deep RL algorithms by normalizing and shifting the reward signals and utilizing the Elephant activation function (Machado et al., 2015; Lan & Mahmood, 2023). The former allows us to avoid the initialization issue, while the latter relaxes the severe generalization by introducing sparsity (Lan & Mahmood, 2023). Details of these approaches are described in Section 2.

We empirically analyze, under optimistic initialization, the effect of shifting reward signals and introducing sparsity in promoting the exploration of an RL agent with non-linear function approximation. Section 4 summarizes the details of our experiments.

2. Background

Consider a Markov Decision Process (MDP), $M = (\mathcal{S}, \mathcal{A}, p, \gamma, r, \rho)$ where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, p is the transition probability distribution, $\gamma \in (0, 1]$ is the discount factor, $r(s, a)$ is the expected reward received after state-action pair (s, a), and ρ is the initial state distribution. The agent's goal is to obtain a policy π that maximizes the expected return $G_t = \mathbb{E}_\pi \left[\sum_{k=t}^{\infty} \gamma^{k-t} r(S_k, A_k)\right]$ where $A_t \sim \pi(\cdot|S_t)$ and $S_{t+1} \sim p(\cdot|S_t, A_t)$ for $t \in \mathbb{N}_+$. State-value functions are defined to be the expected return under policy π, $v_\pi(s) = \mathbb{E}_\pi [G_t|S_t = s]$. Similary, action-value functions are defined as $q_\pi(s, a) = \mathbb{E}_\pi [G_t|S_t = s, A_t = a]$. In large problem domains, such

[*]Equal contribution [1]Department of Computing Science, University of Alberta. Correspondence to: Alireza Azimi <sazimi@ualberta.ca>.

Source code for all tasks available at https://github.com/azimi99/CMPUT655-Project.git

as Atari games (Bellemare et al., 2013), the value functions v_π and q_π become too difficult to compute, and some form of function approximation is needed in practice. In this work, we are particularly interested in parameterized q_π approximated by Multilayer Perceptron (MLP).

2.1. Shifting Reward Signals

Machado et al. (2015) proposed a method for standardizing and decreasing the magnitude of the reward signals as a means of domain-independent optimistic initialization. This methodology enables the estimation of an upper limit of reward signals for a given task and facilitates the rudimentary initialization of function parameters. Precisely, as the upper bound of the expected return will be 0 after normalizing and shifting, initializing the model weights with standard normal distribution automatically grants optimism since the expected output of such functions is 0. The method involves normalizing all rewards (r_t) based on the maximum reward (r_{max}). Furthermore, rewards have been reduced by $\gamma - 1$. To prevent the quick end of agents engaged in episodic tasks, a reward for termination has been implemented: $r_{end} = \gamma^{T-k+1} - 1$, which encourages the agent to seek an improved solution. Here, k represents the total number of steps taken in the episode, whereas T is the maximum allowable number of steps. The approach solely relied on the linear function approximator and has not been empirically evaluated using Deep Reinforcement Learning (DRL) methods.

2.2. Local Elasticity & ElephantMLP

The requirement for locally constrained updates of neural networks leads to the notion of local elasticity (He & Su, 2020). A function f_w is locally elastic if $f_w(\boldsymbol{x})$ is not significantly changed after an update on f_w is performed at $\boldsymbol{x}'$ that is dissimilar to $\boldsymbol{x}$ in a certain sense. He & Su (2020) show that neural networks are locally elastic in general, but Lan & Mahmood (2023) find that the degrees of local elasticity of neural networks with classical activation functions are insufficient to address problems that require updates to be tightly constrained, such as catastrophic forgetting in continual learning. Lan & Mahmood (2023) proposed the Elephant activation function, which is a bell-shaped activation function defined as:

$$Elephant(x) = \frac{1}{1 + \left|\frac{x}{a}\right|^d}$$

The Elephant activation function generates both sparse representations and sparse gradients. Here, x is the input, and the parameter d controls the slope of the activation function. As the value of d increases, the slope of an elephant function becomes steeper, and the gradient signal becomes sparser. The parameter a controls the width of the function, which determines the level of the sparsity of the gradient. They showed that EMLP, an MLP using the elephant activation function, has better local elasticity than traditional MLP and can reduce catastrophic forgetting.

3. Research Question

This paper aims to address the following question:
Is shifting the reward function, as proposed by Machado et al. (2015), an effective way of promoting exploration through optimistic initialization with non-linear function approximation? Could using the Elephant activation function increase exploration through sparse gradient updates?

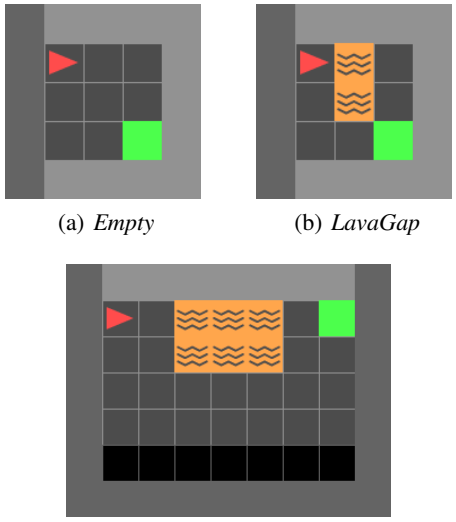
4. Experimental Design

This section addresses all aspects of experimental designs. Initially, we go into the specifics of the basic components, including environments, network architectures, and other relevant elements. Subsequently, we proceed to elaborate on the particular methods of each experiment.

4.1. Details of the Common Components

4.1.1. MiniGrid Environment

In this research, we utilize the *Minigrid* library to conduct our experiments. Specifically, we use three stationary environments: *Empty*, *DistShift*, and *LavaGap* (Chevalier-Boisvert et al., 2023).

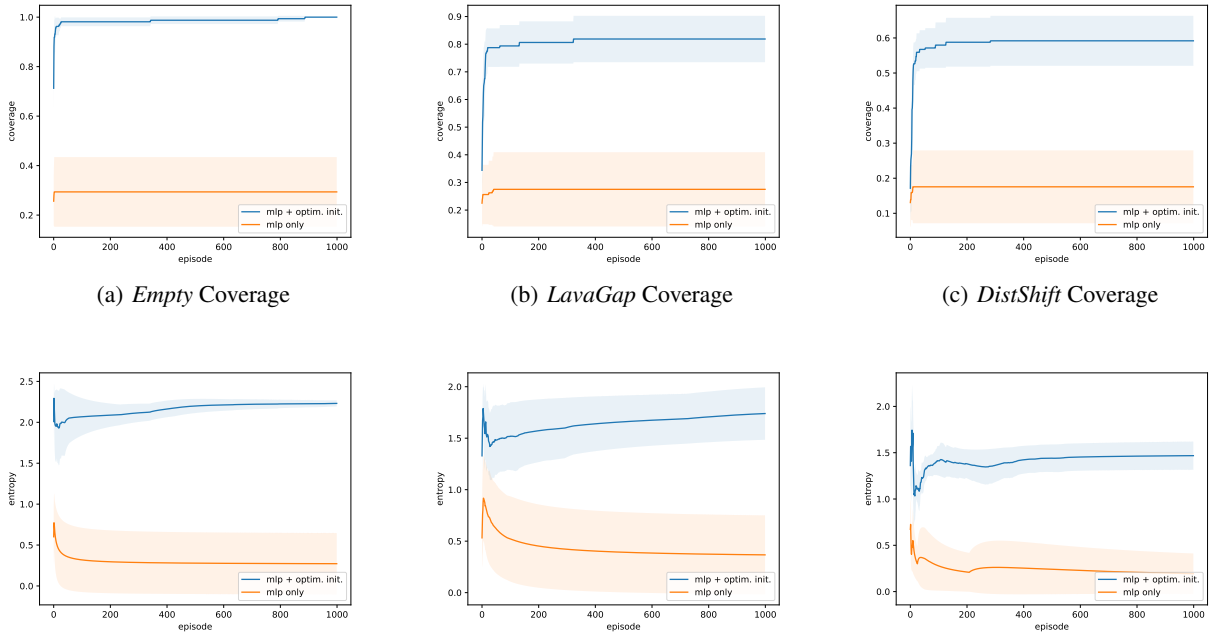


(a) *Empty* (b) *LavaGap*

(c) *DistShift*

Figure 1. Illustrations of the environments for conducting experiments (Chevalier-Boisvert et al., 2023)

Empty (Fig. 1(a)) is a static, gridworld environment with no obstacles, and the RL agent needs to reach the goal state



(a) *Empty* Coverage (b) *LavaGap* Coverage (c) *DistShift* Coverage

(d) *Empty* Entropy (e) *LavaGap* Entropy (f) *DistShift* Entropy

Figure 2. Coverage and entropy of the **MLP** agent with and without optimistic initialization. The blue lines represent the optimistically initialized agents, and the orange lines represent the randomly initialized agents. Each experiment ran for 10 seeds.

with the fewest possible steps. *LavaGap* (Fig. 1(b)) is a stationary gridworld environment in which the agent must navigate through a narrow opening to reach the goal state while avoiding a river of lava. Lastly, *DistShift* (Fig. 1(c)) is a stationary gridworld environment where the agent needs to reach a goal state by avoiding a pool of lava.

The environments described above have a discrete state-action space. The actions are $\mathcal{A} = \{up, down, left, right\}$, and the states are represented by the matrix of $H \times W \times 3$ (Chevalier-Boisvert et al., 2023). Here, H is the number of rows of the grid; similarly, W is the number of columns of the grid. Each channel in the observation represents the object ID, color ID, and agent direction ID. Each ID in a particular grid represents the object's property on the grid. Since the color and direction of the agent are unnecessary information, the observations given in the experiments are the flattened first channel [1]. To optimistically initialize the state-action values, we will shift and normalize the reward function (Machado et al., 2015).

[1]The color channel is omitted since the identification of the object is already done in the first channel. The agent's direction is also unnecessary because the environments are fully observable.

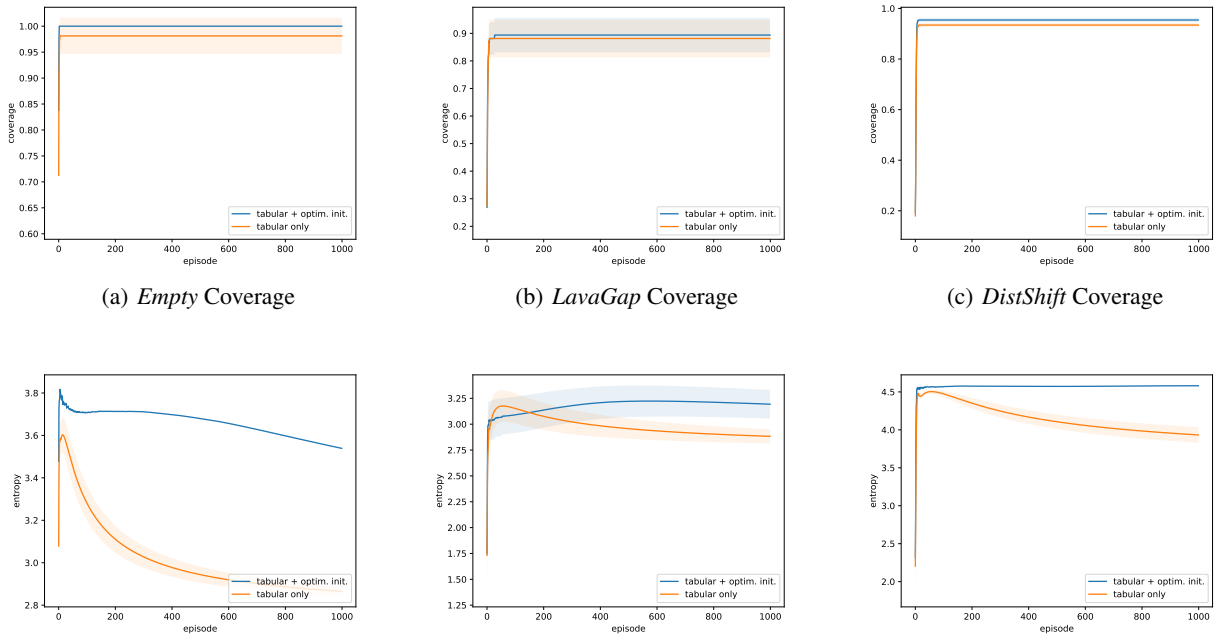
4.1.2. Non-linear Function Approximator

We use MLP as a non-linear function approximator. It consists of two hidden layers of size 100, each activated with ReLU. The input size is $H \times W$, and the output size is 4, where each output corresponds to the state-action value. On the occasion of optimistic initialization, the network's outputs are shifted uniformly by a constant to satisfy the optimism. To derive this constant, we compute the state-action values for all the states and take the minimum value. The difference between a minimum value and an optimistic value is a constant. In practice, the uniform shift is done by adding the constant to the bias terms in the last layer.

4.2. Details of the Experiments

4.2.1. Evaluation on the Exploration

One of the project's ultimate goals is to answer whether the optimistic initialization gives a certain degree of exploration to the non-linear function approximators. This experiment mainly aims to answer this question by comparing the exploration metrics, *state visitation entropies* and *state coverage*, of the non-linear MLP with and without optimistic initialization. The exploration metrics are evaluated by running greedy semi-gradient SARSA (Sutton & Barto, 2020, §10.1) with and without the optimistic initialization for 1000 episodes on the following variation of Minigrid en-



(a) *Empty* Coverage (b) *LavaGap* Coverage (c) *DistShift* Coverage

(d) *Empty* Entropy (e) *LavaGap* Entropy (f) *DistShift* Entropy

Figure 3. Coverage and entropy of the **tabular** agent with and without optimistic initialization. The blue lines represent the optimistically initialized agents, and the orange lines represent the randomly initialized agents. Each experiment ran for 10 seeds.

vironments: *MiniGrid-Empty-6x6-v0*, *MiniGrid-DistShift1-v0*, and *MiniGrid-LavaGapS6-v0* (Chevalier-Boisvert et al., 2023).

The results are generated with the discount factor $\gamma = 0.9$ and the learning rate $\alpha = 5 \times 10^{-4}$. The state visitation entropy is computed by $H = -\sum_{s \in \mathcal{S}} P(s) \cdot \log_2(P(s))$, where s is the state and $P(s) = \frac{n(s)}{\sum_{s \in \mathcal{S}} n(s)}$ where $n(s)$ is the number of times a state s has been visited (Shannon, 1948). To calculate state coverage, we use the number of states visited divided by the total number of states available.

4.2.2. Exploration Compared to the Tabular Settings

While the first experiment addresses whether optimism provides exploration in non-linear function approximation settings, it fails to discuss the properties of this exploration. As a complement, therefore, this experiment addresses how much exploration optimism provides in the non-linear function approximation settings compared to the degree of exploration that optimism provides in the tabular settings. To do this, we evaluate the difference in the exploration metrics within each setting and then compare those differences across the settings. We will reuse the results from a previous experiment for the MLP. For the tabular setting, we evaluate the same exploration metrics by running greedy SARSA (Rummery & Niranjan, 1994) on the identical environments from the previous experiment. We use the discount factor $\gamma = 0.9$ and the learning rate $\alpha = 10^{-2}$ for the tabular setting.

4.2.3. Evaluation on the Effectiveness of EMLP

We compare the episodic returns of MLP and EMLP in the optimistic setting with their performance in the randomly initialized setting. Furthermore, we compare the episodic returns to those obtained through tabular SARSA as a baseline performance measure. The episodic state entropy and coverage for MLP and EMLP in the optimistic setting are compared with the trends in the randomly initialized setting. We also compare the entropy and coverage values of the non-linear approximators with tabular SARSA for optimistic and randomly initialized setups.

5. Experimental Results

In this section, we cover and analyze the results of our experiments described in the previous section.

5.1. Evaluation on the Exploration

The experimental results signify that the optimistic initialization encourages exploration. Figures 2(a) and 2(d) compare coverage and entropy between the optimistically and non-

optimistically initialized non-linear MLP in *Empty*. The coverage plot clearly shows that the non-linear function approximator with optimistic initialization almost covers the entire state space, while the non-optimistically initialized model covers at most half the states. A similar trend is also observable in the entropy. Optimistically initialized MLP achieves a higher overall state visitation entropy than the non-optimistically initialized MLP. These two plots have documented that the exploration effect is also given to the non-linear function approximators. Not only limited to the *Empty*, but the identical argument applies to all the results presented in Figure 2.

5.2. Performance comparison of MLP and Tabular

Both MLP (Figure 2) and tabular implementations (Figure 3) of SARSA exhibit improved exploration when operating under optimistic initialization. A higher degree of state visitation entropy and coverage is achieved across all environments when running MLP and tabular SARSA experiments. For example, In the *Empty* environment, both the MLP and tabular agents eventually achieve full coverage of the environment using optimistic initialization (Figures 2(a) and 3(a)). However, the tabular agent achieves full coverage faster. A plausible explanation for the faster coverage of the tabular agent would be its ability to discriminate between more optimistic states, thus selecting unique states more frequently.

A notable difference between tabular and MLP is the disparity in entropy and coverage in the optimistic and non-optimistic settings. The difference in entropy (Figures 2(d) to 2(f)) and coverage (Figures 2(a) to 2(c)) of the MLP agent in the optimistic and non-optimistic setting is, in general, more significant than the difference in the tabular agent (Figure 3). In other words, the MLP agent shows a larger degree of improvement in the optimistic initialization settings compared to the tabular agent.

Comparing the entropy and coverage results for MLP in *LavaGap* (Figures 2(b) and 2(e)) to tabular (Figures 2(a) and 3(d)), we can observe our tabular implementation outperforming MLP in both optimistic and non-optimistic settings. The tabular method achieves an overall higher entropy value and state coverage. We can attribute this difference to the discrimination ability of tabular in distinguishing between lava and regular states. In the MLP setting, it's plausible the agent falls into the lava more frequently due to more generalization, thus resulting in an overall lower coverage and entropy value. We observe a similar pattern between MLP (Figures 2(c) and 2(f)) and tabular in the *DistShift* environment (Figures 3(c) and 3(f)) where tabular has overall higher degree of exploration as it can avoid the lava more effectively, thus obtaining larger entropy and coverage. It's worth mentioning that full coverage in *DistShift* and *LavaGap* cannot be achieved since we handle the lava states as locations that the agent cannot cover.

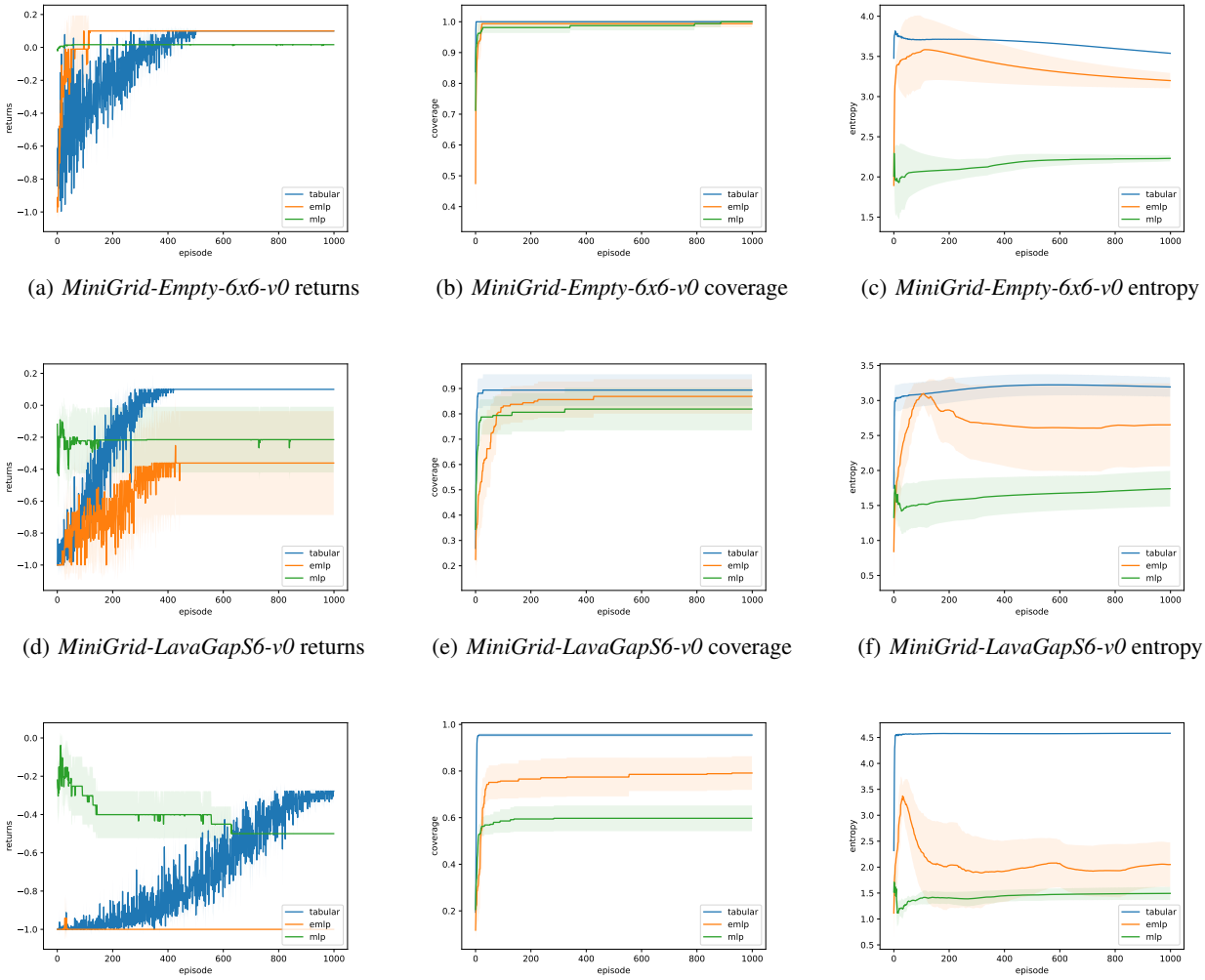
5.3. MLP vs EMLP

In this section, we discuss the effect of applying the elephant activation function inside neural networks to facilitate exploration through optimistic initialization. The elephant activation function can generate sparse representations and sparse gradients (Lan & Mahmood, 2023). This property gives EMLP more controlled generalization and a higher local elasticity than MLP with classical activation functions.

Figures 4(a), 4(d) and 4(g) show the comparison of episodic returns between the MLP and EMLP agents in the three environments. In *Empty* environment, the EMLP agent successfully learned an optimal policy that achieves the largest returns possible. The MLP agent, on the other hand, converged to a suboptimal policy. However, neither agent learned an optimal policy in *LavaGap* and *DistShift*. Most notable is the EMLP agent failing catastrophically in *DistShift*. It kept getting stuck in a corner forever or moved to terminate the episode as quickly as possible by jumping into the lava.

The benefit of controlled generalization in gradient updates is evident in Figure 4(a). The EMLP agent always learned an optimal policy. Although the MLP agent learned much faster at the beginning because of its inherent global generalization, this uncontrolled generalization can quickly change the value of other actions, including the optimal actions, to a non-optimistic one, resulting in less exploration in the long term as demonstrated in Figure 4(c). Failing to explore leads to sub-optimal policy at the end. Conversely, the tabular agent learned much slower than the EMLP agent because it had no generalization. It can also be seen that the learning curve of the tabular agent is much noisier than that of the EMLP agent. This result shows that an appropriate degree of generalization is conducive to learning, but an excessive amount can also hurt performance. Balancing generalization is an essential area of research for many other learning methods, such as continual learning (Raghavan & Balaprakash, 2021; Lin et al., 2023), and more research should be done in this area.

During the experiments, we discovered that the performance of the EMLP agent is very sensitive to the initialization of the model weights, especially the bias values in the last layer. In the original initialization scheme proposed by Lan & Mahmood (2023), the bias values in the layers where elephant functions are used are initialized with evenly spaced numbers over the interval $[-\sqrt{3}\sigma_{bias}, \sqrt{3}\sigma_{bias}]$, where σ_{bias} is a hyperparameter that represents the standard deviation of the bias values. All other bias values are initialized to 0. Such initialization allows EMLP to generate diverse features. However, in our problem setting, the bias values



(a) *MiniGrid-Empty-6x6-v0* returns (b) *MiniGrid-Empty-6x6-v0* coverage (c) *MiniGrid-Empty-6x6-v0* entropy

(d) *MiniGrid-LavaGapS6-v0* returns (e) *MiniGrid-LavaGapS6-v0* coverage (f) *MiniGrid-LavaGapS6-v0* entropy

(g) *MiniGrid-DistShift1-v0* returns (h) *MiniGrid-DistShift1-v0* coverage (i) *MiniGrid-DistShift1-v0* entropy

Figure 4. Tabular, MLP & EMLP agents with optimistic initialization. The blue, orange, and green lines represent the tabular, EMLP, and MLP agents respectively.

in the last layer need to be initialized so that the outputs of EMLP are always optimistic. This deviation from the original scheme leads to degraded performance. Also, as a limitation, there is a lack of theoretical ways to properly set the hyperparameters of EMLP, including σ_{bias}.

6. Conclusion

In this research project, we experimented with the combination of optimistic initialization and non-linear function approximation in facilitating exploration in RL tasks. We combined reward normalization and shifting proposed by Machado et al. (2015) with neural networks and conducted experiments using three stationary Minigrid environments. From the experimental results, we can conclude that optimistic initialization can provide the MLP agent with useful exploration. In the optimistically initialized setting, while the tabular agent displays slight enhancement in its exploratory behaviors, the MLP and EMLP agents demonstrate a much more prominent jump in their state coverage and visitation entropy. Additionally, the experiments exhibit the fact that EMLP covers a higher portion of the state space than MLP when they are both optimistically initialized. This is attributed to the EMLP agent's capability of controlling the generalization effect of gradient updates and learning sparse representations. Although EMLP achieves optimal policy convergence in the *Empty* environment, both MLP and EMLP exhibit a decline in performance in the *DistShift* and *LavaGap* environments. Particularly in *DistShift*, EMLP performed much worse than MLP, indicating that

EMLP is extremely sensitive to the choice of hyperparameters and the initialization scheme. In general, we observed that optimistic initialization by normalizing and shifting reward signals cannot be easily applied to function approximation in any arbitrary domain. It requires deep analysis and careful design in order to make the combination work.

7. Contributions

Alireza Azimi: Worked on synthesizing the project's research question, abstract, introduction, experimental design, results, and codes.
Haruto Tanaka: Worked on synthesizing the research question, introduction, background, experimental design, results, and codes.
Henry Du: Worked on surveying the related works, introduction, background, experimental results, and codes.
Mashfiq Shahriar Zaman: Worked on the background, experimental design, conclusion, reference, and codes.

References

Auer, P., Cesa-Bianchi, N., and Fischer, P. Finite-time Analysis of the Multiarmed Bandit Problem. *Machine Learning*, 47:235–256, 05 2002.

Bellemare, M. G., Naddaf, Y., Veness, J., and Bowling, M. The Arcade Learning Environment: An Evaluation Platform for General Agents. *Journal of Artificial Intelligence Research*, 47(1):253–279, 2013.

Brafman, R. I. and Tennenholtz, M. R-Max - A General Polynomial Time Algorithm for near-Optimal Reinforcement Learning. *Journal of Machine Learning Research*, 3:213–231, 2003.

Chevalier-Boisvert, M., Dai, B., Towers, M., Lazcano, R. d., Willems, L., Lahlou, S., Pal, S., Castro, P. S., and Terry, J. Minigrid & Miniworld: Modular & Customizable Reinforcement Learning Environments for Goal-Oriented Tasks. *CoRR*, abs/2306.13831, 2023.

He, H. and Su, W. J. The Local Elasticity of Neural Networks. In *International Conference on Learning Representations*, 2020.

Lan, Q. and Mahmood, A. R. Elephant Neural Networks: Born to Be a Continual Learner. *CoRR*, abs/2310.01365, 2023.

Lin, S., Ju, P., Liang, Y., and Shroff, N. Theory on Forgetting and Generalization of Continual Learning. In *International Conference on Machine Learning*, 2023.

Machado, M. C., Srinivasan, S., and Bowling, M. Domain-Independent Optimistic Initialization for Reinforcement Learning. In *AAAI Workshop on Learning for General Competency in Video Games*, 2015.

Raghavan, K. and Balaprakash, P. Formalizing the Generalization-Forgetting Trade-off in Continual Learning. In *Advances in Neural Information Processing Systems*, 2021.

Rashid, T., Peng, B., Boehmer, W., and Whiteson, S. Optimistic Exploration even with a Pessimistic Initialisation. In *International Conference on Learning Representations*, 2020.

Rummery, G. and Niranjan, M. On-Line Q-Learning Using Connectionist Systems. *Technical Report CUED/F-INFENG/TR 166*, 11 1994.

Shannon, C. E. A Mathematical Theory of Communication. *The Bell System Technical Journal*, 27(3):379–423, 1948.

Sutton, R. S. and Barto, A. G. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, USA, 2020. ISBN 9780262039246.

Szita, I. and Lőrincz, A. The Many Faces of Optimism: A Unifying Approach. In *International Conference on Machine Learning*, 2008.