2024 19th Annual System of Systems Engineering Conference (SoSE) | 979-8-3503-6591-7/24/$31.00 ©2024 IEEE | DOI: 10.1109/SOSE62659.2024.10620974

Fear based Intrinsic Reward as a Barrier Function for Continuous Reinforcement Learning

Rodney Sanchez
Dept of Electrical and Microelectronic Engineering
Rochester Institute of Technology
Rochester, USA
RAS8047@rit.edu

Ferat Sahin
Dept of Electrical and Microelectronic Engineering
Rochester Institute of Technology
Rochester, USA
feseee@rit.edu

Jamison Heard
Dept of Electrical and Microelectronic Engineering
Rochester Institute of Technology
Rochester, USA
jrheee@rit.edu

***Abstract*—Current reinforcement learning (RL) methods must explicitly sample states to learn about their value. Redundantly sampling these states creates dangerous situations when RL agents are deployed in real-life environments. Furthermore, since the agent only receives a reward when entering the environment, the agent must sample the state multiple times to modify the agent's policy to avoid dangerous states. Humans, specifically young children, primarily overcome this need for explicit and redundant sampling using two key strategies: fear and vicarious conditioning. Our method utilizes fear and vicarious conditioning to create a pseudo-barrier function that discourages the agent from sampling the dangerous state. Using memory augmented neural network (MANN) similarity calculations, we can see how similar the agent's current state is to the "phobia" by creating a dense reward field that serves as a pseudo-barrier function. The MANN was trained in the MiniGrid Simple environment, while the agent was tested in the LAVAGAP and Dynamic Obstacle environments. Our results show that the MANN can produce a dense reward gradient that transfers to different Minigrid environments. Our method also shows that this "phobia" can discourage the agent from visiting certain states.**

***Index Terms*—Intrinsic Rewards, Low Shot Learning, Reinforcement Learning**

I. Introduction

Reinforcement Learning is a growing subsection of machine learning that has shown the capacity to solve long-term optimization problems across various domains [1], [2]. Although reinforcement learning (RL) has shown the capacity to optimize complex image state spaces and handle continuous action spaces, the deployment of RL methods to robotic system of systems has been limited. This limitation has been largely driven by the possibility of the RL algorithm taking actions that could hurt the agent or other agents around them [3]. This possibility is a subset of the commonly considered sim-to-real paradigm. Another aspect of the sim to real problem space is the need for reinforcement learning agents to explicitly sample a state and its actions for the agent to learn its value. For a robotic system, this produces a scenario where the agent needs to directly experience a dangerous state to update its policy, which can be dangerous to the robot and the people around it. Furthermore, any variation to the state space must also be sampled by the agent, creating another condition where the agent puts itself or others in danger.

When considering dangerous states that the agent needs to avoid, we consider a few conditions. First, we find that when deploying an RL agent to a real-world scenario can create a condition where an agent could repeatedly sample a dangerous state before it could learn the state value , potentially due to sample inefficiency issues [3]. Typical Out Of Distribution (OOD) resilient RL methods focus on training the agent [4] to be resilient to possible OOD representations using external datasets or modifications to the environment's state representation. These methods attempt to produce a sufficient variety of OOD representations that allow the agent to handle visual differences from simulation to deployment. Other methods focus create disturbances in the state space and using semi-supervised learning methods to create a pseudo-discretized state space [5]. Other OOD methods have shown successful, safe transfer learning through the use of Control barrier transfer functions, but these methods work on explicitly defined state definitions. This limits the agent's capacity to deploy in an environment where the dangerous condition is abstract and cannot be explicitly defined [3].

In this work, we seek to take the success demonstrated in barrier functions and extend it to abstract dangerous state representations. We introduced *fear intrinsic rewards*, a low-shot learning method that calculates the similarity of a known dangerous state to the agent's current behavior and provides the agent a negative intrinsic reward to avoid dangerous behaviors. Specifically, our contribution is two-fold

1) We provide a few-shot method to create avoidance behavior that transfers across a few simple domains

This material is based upon work supported by the National Science Foundation under Award No. DGE-2125362. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

using intrinsic rewards.

2) We show that this method allows for faster learning of the negative reward.

II. Background

Deep Reinforcement learning describes how neural networks use environmental reward signals to optimize a policy in a given environment [6]. Two paradigms exist: on-policy and off-policy. On-policy methods sample the agent's policy directly and optimize the state and actions that the agent has sampled, while off-policy methods seek to optimize the maximum action [6]. This leads to a trade-off where on-policy learning converges faster and has greater stability but does reach as optimal of a policy as off-policy methods [6]. Although both methods have been extended to use deep neural networks, we largely see that the same paradigm holds for non-deep reinforcement learning approaches [1], [7].

A. Reward Shaping

A key aspect of all RL frameworks is the reward function. The reward function allows the agent to optimize its behavior to maximize the cumulative rewards over an episode. [6] The explicit creation and modification of an environment reward function are defined as reward shaping [8]. Some environments like minigrid [9] only reward the agent upon reaching the goal state. This type of environment reward function is defined as a sparse reward problem, since the agent gets a limited reward signal throughout an episode. Another important reward shaping method is handcrafted reward functions [10]; handcrafted reward functions often balance multiple goals or abstract conditions that the designer believes will lead to an optimal behavior. These functions can often be expressed as one of the following representations: positive, negative, and balanced skews; a positive skew shows faster convergence while a negative skew has shown a decrease in the agent's episodic variance [11]. As environments become more complex and must balance multiple goals, the simple way to ensure that the agent does not develop unintended behaviors is to create a positive reward gradient to the overall environment goal [8].

B. Intrinsic Reward

Intrinsic reward functions describe a reward function that satisfies some desire that is derived from the agent and is not explicitly reinforced by the environment. [12]. These intrinsic reward functions can largely be divided into two discrete paradigms: skill training methods and novelty optimization [13]. Skill-based methods allow the agent to decompose the main goal into a subset of goals that can be optimized while working in a rewards-sparse or reward-noisy environment [14], [15]. Novelty-based intrinsic reward functions are focused on optimizing the exploration vs exploitation paradigm. These intrinsic reward functions differ from previous exploration methods like e-greedy [6] by using randomness to increase the number of states the agent has seen. ϵ-greedy methods increased randomness but do not assure that the agent's exploration had provided any new information to the agent [6], [13]. Two major developments in the novelty intrinsic reward have been pseudo-counter [16] and methods and entropy [17]. A PsuedoCounter keeps track of state visitation, minimizing a loss function for a random network where a decrease in the loss designated an increase in that state's visitation. other novelty intrinsic reward optimize the entropy from state transitions, pushing the agent to increase the distribution of states it has previously seen [17].

C. Biological Fear

Fear is an emotional response that serves as a methodology to avoid or get out of a dangerous/potentially dangerous situation. Largely, fear leads to what is commonly understood as the freeze, fight, or flight response. Human fear can be largely explained with two distinct types of fear expression a learned fear response or evolutionary fear response [18]. Learned fear responses are learned through two distinct behavioral patterns: direct exposure to stimuli or vicariously observing someone else experience the stimuli [19]. Children best optimize this vicarious fear, allowing them to watch their guardian's expression of state to understand its value without the need for explicit sampling. [19] This vicarious learning helps children emulate safe behavior strictly through social interaction. This behavior of learning through societal interaction is referred to as societal learning. Within the social learning theory exists a framework that describes the method a person uses to learn from others [20]. From this paradigm, we can break down the vicarious learning of fear into distinct steps: observation, remembering, reproduction, and motivation. One key aspect we focus on is remembering and reproduction, where the agent can recall the stimuli and understand their meaning or value. To better understand this behavior, we can understand the surgical pathways of recalling fear as a process that takes in a stimuli process. The stimuli reference the behavior with memories and then use the amygdala to encode the value of the stimuli [18]. This expression of fear demonstrates that fear is a methodology that requires the capacity to encode the stimuli and reference the encoding to prior information.

D. Memory Methods

Memory-based machine learning methods focus on the capacity to retain, write, and retrieve information that the method has been exposed to. This capacity to take stimuli and understand what memory aligns with the methods is called content-based memory addressing [21]. This form of memory addressing is decomposed into three distinct operations: read, write, and erase. where reading uses attention to address the external memory, writing modifies the external memory and erase function that mimics the LSTM's forget gate [22]. These memory-based methods can process an input and relate it to the stored value in memory, and this recall methodology allows for a simple low-shot learning paradigm [21]. Other methods that perform Memory-based content Addressing use correlation and Hebbian learning to track and strengthen the association of representations [23].

III. Methodology

We introduce fear-based intrinsic reward (FIR), a novel negative intrinsic reward that uses content memory-based addressing methods to modify an agent's reward function to create avoidance behaviors similar to those in humans. The method replicates humans' response when exposed to a phobia, where the closer our observation is to the phobia, the greater our need to move away from it. Specifically, FIR seeks to replicate humans' capacity to scale danger appropriately by calculating the similarity of the current stimuli to a known memory. This similarity allows the agent solve a rewards-dense environment in a sample efficient manner.

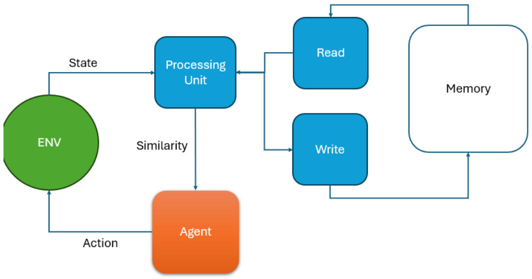

Fig. 1. An overview of the FIR methodology.

Our method (see Figure 1) demonstrates how the agent takes the environment's state representation and produces an intrinsic reward signal based on the similarity of the state to the example learned by MANN. Therefore, all training to the MANN will occur before the agent's training in its environment. We decompose this training into two distinct paradigms that mimic the theory of societal learning. First, the attention and retention phase is the explicit training of the MANN.This mimics "society" and provides examples that the agent will use to learn about a negative reward.

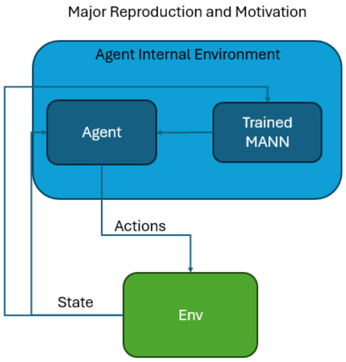

Fig. 2. The trained MANN that produces reinforcement and motivation.

Since the proposed method aims to test the ability to create avoidance behaviors across two domains, states that share some similarity to the other domains are incorporated, but these states are not one-to-one representations. Once the agent learns these representations within the MANN, the agent is deployed to an unseen environment. This training phase replicates the reproduction and motivation phase of societal training where the agent replicates the behavior that it has observed and uses intrinsic motivation to reinforce those actions, as outlined in Fig. 2. This replication is created through a negative intrinsic reward. This intrinsic reward scales up similarly as the agent approaches the described behavior, producing a dense reward function that dissuades the agent from exploring that specific state. This follows the reward-shaping paradigm that produces a negative reward gradient around undesired states.

For example, the agent is shown state representations (Fig. 3) that have a high negative societal/intrinsic reward. Next, a barrier function creates an intrinsic reward space (Fig. 4), even if the environment's extrinsic reward (Fig. 5) states otherwise. The paper examines an environment with an extrinsic reward signal that is equal or greater than the maximum value of intrinsic reward. This will simulate a situation where the extrinsic rewards might be significantly high while the intrinsic reward is negative. This mimics a biological example where a human walks past a snake, where prior experience with snakes is through external information sources (such as others handling a snake). This minimal sampling of previous experience creates a similar representation to the snake in the path. Thus, the human avoids the snake with minimal state representations. Thus, our proposed method may have domain transference properties with low-shot representations.

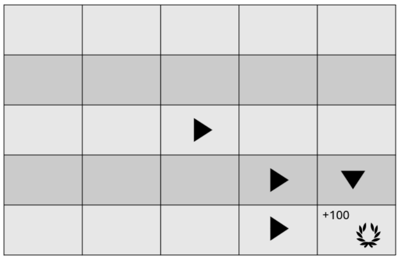

Fig. 3. Bad examples that emulate some subsection of a policy

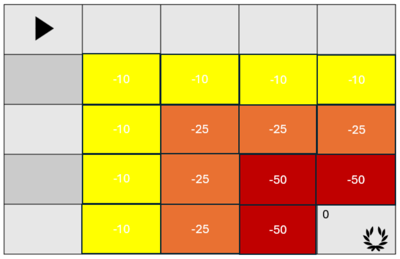

Fig. 4. The intrinsic negative barrier field.

IV. Experiment Setup

A well-known sparse reward environment that tests transfer learning methods is the MiniGrid Environments, such as the

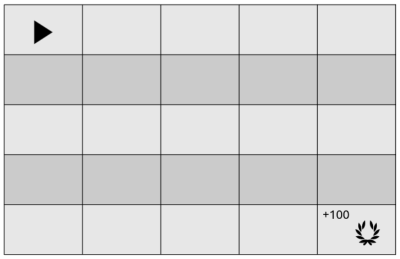

Fig. 5. Demonstrates a reward sparse environment with an extrinsic reward.

visual partially observable representations. Minigrid's sparse reward permits evaluating the FIR's ability to create a region of deterrence without interfering with other reward functions. Two wrappers were used in the experiment: Imgobswrapper, which removes the text data that the environment provides in its state representation, and partialobservable wrapper, which reduces the agent's vision to its field of view. Although MiniGrid is a commonly used framework, our environment restrictions produce a distinct testing condition that has not been previously documented. Thus, a baseline for proximal policy optimization (PPO) is provided. We also limit the Minigrid environments to those explicitly solved visually (e.g., LAVAGAP, DISTSHIFT, LAVACROSSING, and DYNAMIC OBSTACLE). We also chose these specific environments because they provide different terminal conditions and another negative reward that incentivizes variations in the agent's behavior. We expect the agent to learn to avoid conditional negative rewards and balance the negative reward with the intrinsic reward. We also used environments with terminal conditions to see if the agent would avoid the terminal conditions with negative rewards. The Negative behavior examples are collected from the EMPTYROOM environment and are separated into two classes: avoid and other (Fig. 6).

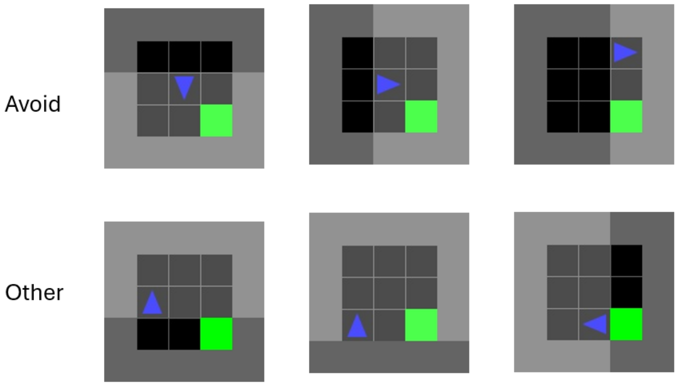

Fig. 6. PPO baseline for the DistShift using Partially Observable Enviroment

We separate the environments across the types of changes as shown in Table I. Here, we distinguish the visual and environmental differences that each environment provides, such as terminal conditions. These terminal conditions allow the agent to end the episode without reaching the GOAL. Other environments have non-terminal negative rewards, where the agent suffers punishment from collisions. This creates a secondary reward signal that the agent has to optimize. Finally, we distinguish between the LAVAGAP, and DYNAMIC Obstacles environments from the rest, as the environments' size and general representation are distinct from EMPTYROOM.

Across all environments, each game terminates after 400 steps, and 20,000 training steps are used to ensure that the observed cumulative reward for each agent can accurately be compared to its baseline. The MANN is trained during each run, and the accuracy achieved creates the intrinsic reward. This accuracy demonstrates how learning features specific to EMPTY ROOM will restrict domain transfer to unseen environments. Test environments similar to EMPTYROOM (i.e., LAVAGAP and DYNAMIC OBSTACLE) will result in higher MANN accuracy, while the remaining environments will result in lower accuracy. Three distinct types of intrinsic rewards will be evaluated: Dominant intrinsic reward where the agent is pushed away from a terminal state, balanced intrinsic reward, and dominant extrinsic reward where the agent is not sufficiently deterred.

TABLE I
MINIGRID TESTING ENVIROMENTS

Table	**Environmental Changes**		
Environment	**Terminal Condition**	**Conditional Negative**	**Representation**
LAVACROSSING	X	X	X
LAVACGAP	X	X	
DYNAMIC OBS		X	
DIST SHIFT	X	X	X

V. RESULTS AND ANALYSIS

This section seeks to establish a baseline for the Minigrid environments. From the Dishift environment in Fig. 7, we see large disparities between runs. This is a byproduct of PPO, since PPO is an on-policy method. This means that if the PPO agent did not directly sample a behavior where the agent can achieve a faster policy to the goal, it will not modify its behavior to find that potential policy.

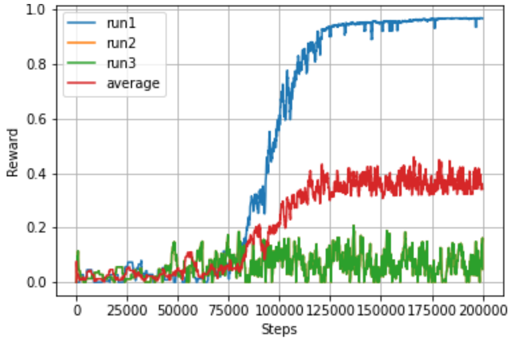

Fig. 7. Baseline runs on DISTSHIFT environment shows that on average, the agent achieves an average of .4

From fig 8 and fig 9, we see that the agent can solve these simple environments. In Dynamic Obstavcles specifically, we see that the agent's negative conditional reward helps reduce

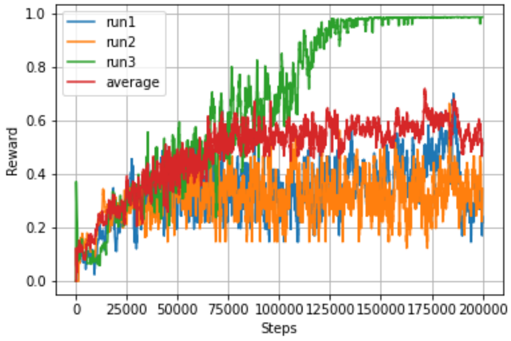

Fig. 8. Baseline runs on LAVAGAP environment shows that on average, the agent will plateau at the cumulative intrinsic reward of .6

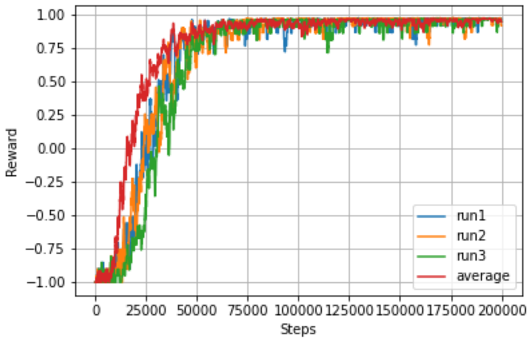

Fig. 9. Baseline runs on the DYNAMIC OBSTACLE environment show that, on average, the agent can solve the game, achieving an average of near 1.

the explicit sampling required by PPO, while LavaGAP, with its conditional terminal state, added more episodic variance. An important distinction we would like to make is that no matter the variation in performance across these states, the agent always manages to learn to reach the goal state.

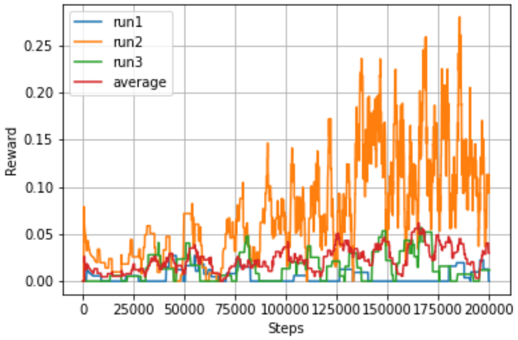

Fig. 10. Baseline runs on LAVACROSSING environment showing that the agent achieved an average of .04.

From Fig. 10, we can see a continuation of previous trends where LAVACROSSING shows a direct increase in score variance with the presence of a terminal condition.

This section will discuss the behavior achieved across multiple runs in a dynamic obstacle environment. We will observe how the different achieved accuracy produced different possible policies for the agent. The first case from fig11 is the MANN failing to generalize to the transfer environment, which occurs when the MANN has an accuracy of 100 percent. When the MANN achieves 100 percent accuracy, it has learned features too specific to the training set, limiting MANN from generalizing to the state representation in the DYNAMIC OBSTACLE ENVIRONMENT. From fig11, we also observe that an accuracy of 68 percent produced a balanced intrinsic reward where the agent could optimize the conditional reward and the FIR intrinsic reward. We believe this was caused by the MANN being less accurate but still producing sufficient similarity that it was more optimal to balance the environment's reward and the intrinsic reward. The results show that MANN accuracies from 68 percent to 95 percent produced a successful avoidant behavior in the RL algorithm. There is a range of accuracies whose learned features can create an avoidance for the agent.

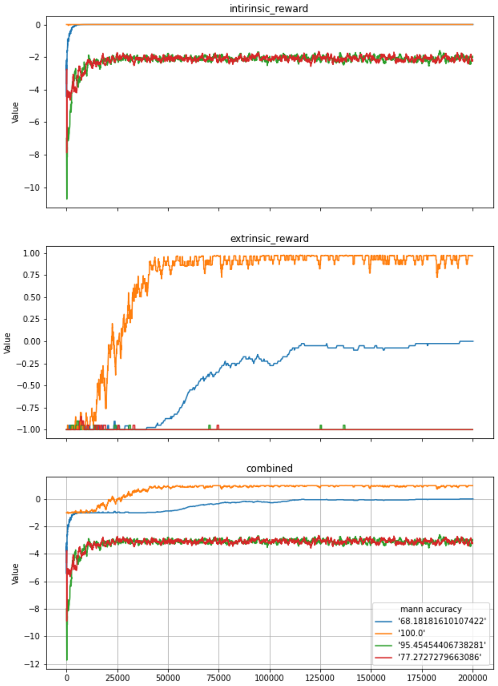

Fig. 11. Runs of FIR in DYNAMIC OBSTACLE environment showing different balanced extrinsic dominant and intrinsic dominant rewards.

In this section, we discuss the results of using negative intrinsic with variable accuracy, as seen in fig 12. One interesting result we see is that high accuracy does not fundamentally mean failed intrinsic reward. We see this with the run that scored 100 percent and produced a successful intrinsic reward. Similarly, extremely low and high accuracy can produce a negative intrinsic reward if the controller learns features that can successfully transfer to the new domain. From the results in LAVAGAP and Dynamic Obstacle, we see the possible accuracy range for MANN to produce a negative intrinsic reward skews higher. We believe that MANN can operate in these accuracies due to the many similarities EMPTYROOM has with these environments.

In this section we will discuss the results of the runs on

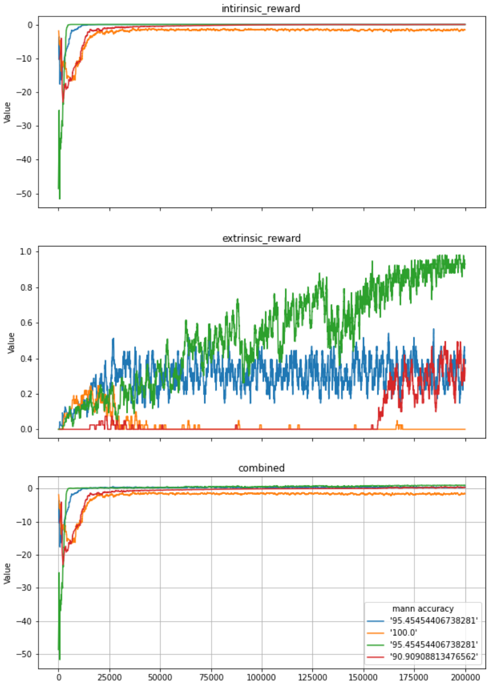

Fig. 12. Runs of FIR in LAVAGAP environment showing that MANN can achieve high accuracy and create a negative intrinsic reward.

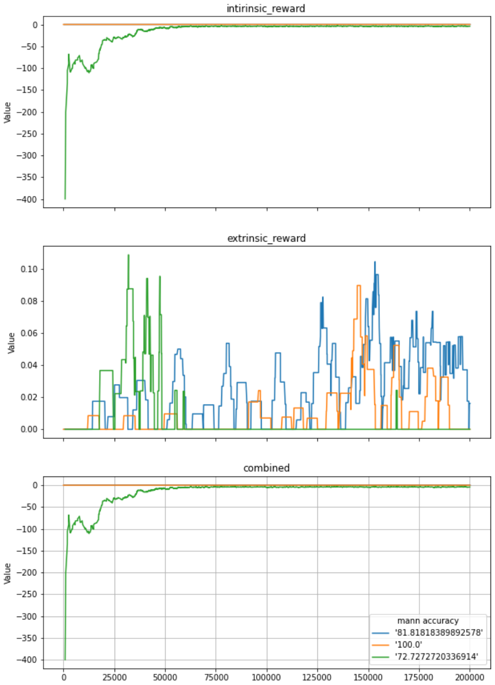

Fig. 13. Runs of FIR in a LAVACROSING environment showing that range in effective accuracy decreased for LAVACROSSING

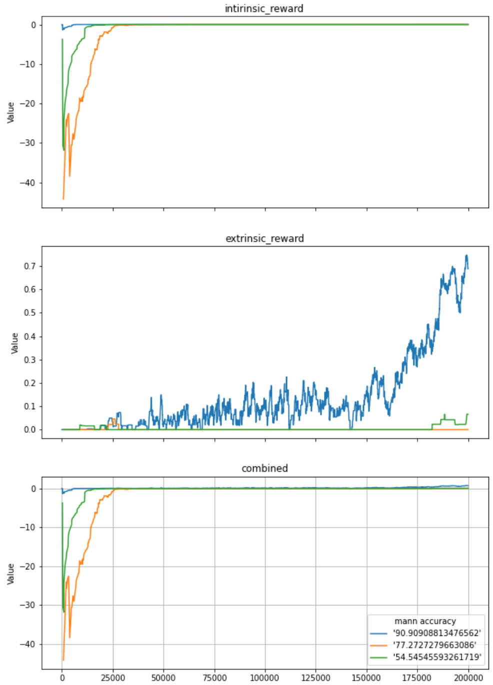

Fig. 14. Runs of FIR in a DISTSHIFT demonstrating that MANN can work with accuracy as low as 54 percent.

the LAVACROSSING as seen in fig 13.LAVACROSSING is our first environment that has a larger number of observable differences from the EMPTYROOM environment on which MANN was trained. From the results, we see that only an accuracy of 72 percent produced the avoidance behavior seen in the previous environments. We see that an accuracy of 80 percent, which worked on prior environments like DYNAMIC OBSTACLES and LAVAGAP, could not produce a negative intrinsic reward. We see that the 80 percent run was as unsuccessful as the 100 percent run. Here, we observe that the accuracy for successful generalization skews toward lower accuracies in environments that have greater differences from the environment it was trained on.

Finally, the runs performed in DISTSHIFT can be seen in fig 14 and continue the trend that environments with a greater difference in representation have a lower range of effective accuracy that will produce an accurate intrinsic reward function. From the results, we see that 77 and 54 percent accuracy do successfully produce an avoidance behavior, but accuracy, like 90 percent, fails to generalize successfully. We also see from the 54 percent accuracy run that the agent is still actively optimizing the intrinsic reward and that if it were to reach the goal state, it could find a path of least similarity that allows it to receive the extrinsic reward.

The results across multiple Minigrid environments demonstrate that we can use memory methods like MANN to modify an agent's policy without modifying the environment. This allows for the creation of complex reward functions that are based on avoiding specific states. This method also shows that memory, specifically recall of saved states, can work as a sample efficiency method since it provides a negative dense reward for similar representations.

VI. Conclusions and Future Works

In this paper, we introduced FIR, a novel negative intrinsic reward for behavior avoidance. We demonstrate how FIR methodology mimics vicarious conditioning by using external examples and recreating the behavior. We show in environments like Minigrid's LAVAGAP,DISTSHIFT,LAVACROSSING, and DYNAMIC OBSTACLE that the FIR can produce a negative reward that promotes avoidance. When transferring to other domains, we also demonstrate how the two critical parameters that affect the FIR's ability to generalize are the shared features or representations and the accuracy achieved by the MANN. We show that the agent can optimize both reward functions across multiple domains and that in environments with conditional negative rewards, the agent can learn to avoid those. In the Future, we want to expand this method to work on human-robot simulated environments (e.g., Copealla Sim) [24] to see if this fear paradigm can be used as a successful collision avoidance method in human-robot teams. With this simulation, we want to see if we can transfer to different human representations and how the agent changes its policy with different representations. We are also interested to see if other environmental changes, like the general setting, will have a larger effect on the policy than changing the person's representation Finally, we want to test if other memory methods like Hopfield networks or neural attention memory are more resilient to representation changes.

References

[1] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, 2015.

[2] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings*, 9 2015. [Online]. Available: https://arxiv.org/abs/1509.02971v6

[3] Y. Emam, S. Member, G. Notomista, P. Glotfelter, Z. Kira, S. Member, and M. Egerstedt, "Safe Reinforcement Learning using Robust Control Barrier Functions."

[4] F. Träuble, A. Dittadi, M. Wüthrich, F. Widmaier, P. Gehler, O. Winther, F. Locatello, O. Bachem, B. Schölkopf, and S. Bauer, "THE ROLE OF PRETRAINED REPRESENTATIONS FOR THE OOD GENERALIZATION OF RL AGENTS," *ICLR*, 2022. [Online]. Available: https://sites.google.com/view/ood-rl.

[5] A. Srinivas, M. Laskin, and P. Abbeel, "CURL: Contrastive Unsupervised Representations for Reinforcement Learning," 2020. [Online]. Available: https://www.

[6] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction Second edition, in progress*, 2nd ed. MIT press, 2018.

[7] D. Zhao, H. Wang, K. Shao, and Y. Zhu, "Deep reinforcement learning with experience replay based on SARSA," *SSCI*, 2018. [Online]. Available: https://www.researchgate.net/publication/313803199

[8] A. Y. Ng, D. Harada, and S. Russell, "Policy invariance under reward transformations: Theory and application to reward shaping," *ICML*, 1999.

[9] M. Chevalier-Boisvert, B. Dai, M. Towers, R. De Lazcano, L. Willems Miple, S. Lahlou, P. S. Castro, G. Deepmind, and J. Terry, "Minigrid & Miniworld: Modular & Customizable Reinforcement Learning Environments for Goal-Oriented Tasks," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[10] T. Goto, Y. Kizumi, and S. Iwasaki, "Design of Reward Function on Reinforcement Learning for Automated Driving," *IFAC-PapersOnLine*, vol. 56, no. 2, pp. 7948–7953, 1 2023.

[11] A. Dayal, L. R. Cenkeramaddi, and A. Jha, "Reward criteria impact on the performance of reinforcement learning agent for autonomous navigation," *Applied Soft Computing*, vol. 126, p. 109241, 9 2022.

[12] S. Singh, A. G. Barto, and N. Chentanez, "Intrinsically Motivated Reinforcement Learning."

[13] A. Aubret, Laetitia Matignon, and Salima Hassas, "An information-theoretic perspective on intrinsic motivation in reinforcement learning: a survey," *Entropy*, vol. 25, p. 327, 2023.

[14] J. Zhang, H. Yu, and W. Xu, "HIERARCHICAL REINFORCEMENT LEARNING BY DISCOVERING INTRINSIC OPTIONS," *ICLR*, 2021. [Online]. Available: https://www.github.com/jesbu1/hidio.

[15] S. Li, L. Zheng, J. Wang, and C. Zhang, "LEARNING SUBGOAL REPRESENTATIONS WITH SLOW DYNAMICS." [Online]. Available: https://sites.google.com/view/lesson-iclr

[16] Y. Burda, H. Edwards, A. J. Storkey, and O. Klimov, "Exploration by Random Network Distillation," *ICLR*, 2019. [Online]. Available: http://arxiv.org/abs/1810.12894

[17] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor," 2018.

[18] C. Askew and A. P. Field, "The vicarious learning pathway to fear 40 years on," *Clinical Psychology Review*, vol. 28, no. 7, pp. 1249–1265, 10 2008.

[19] M. F. Marin, A. Bilodeau-Houle, S. Morand-Beaulieu, A. Brouillard, R. J. Herringa, and M. R. Milad, "Vicarious conditioned fear acquisition and extinction in child–parent dyads," *Scientific Reports*, vol. 10, no. 1, 12 2020. [Online]. Available: /pmc/articles/PMC7555483/ /pmc/articles/PMC7555483/?report=abstract https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7555483/

[20] M. J. Fryling, C. Johnston, and L. J. Hayes, "Understanding Observational Learning: An Interbehavioral Approach," *The Analysis of Verbal Behavior*, vol. 27, no. 1, p. 191, 4 2011. [Online]. Available: /pmc/articles/PMC3139552/ /pmc/articles/PMC3139552/?report=abstract https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3139552/

[21] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, T. Lillicrap, and G. Deepmind, "Meta-Learning with Memory-Augmented Neural Networks Google DeepMind," *ArXiv*, vol. abs/1605.06065, 2016.

[22] A. Graves, G. Wayne, and I. Danihelka, "Neural Turing Machines," *arXiv*, 2014.

[23] S. Hobson, "Correlation Matrix Memories: Improving Performance for Capacity and Generalisation," 2011.

[24] E. Rohmer, S. Singh, and M. Freese, "V-REP: A versatile and scalable robot simulation framework," in *Proceedings of the ... IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE/RSJ International Conference on Intelligent Robots and Systems*, 5 2013, pp. 1321–1326.