APPRAISAL-GUIDED PROXIMAL POLICY OPTIMIZATION: MODELING PSYCHOLOGICAL DISORDERS IN DYNAMIC GRID WORLD

A Project Report

Submitted by

Mr. HARI PRASAD

REG NO : TKM21MEAI05

SEMESTER : IV

In partial fulfillment for the award of the degree of

MASTER OF TECHNOLOGY
IN
Mechanical Engineering (Artificial Intelligence)

Under the guidance of
Prof. CHINNU JACOB

Thangal Kunju Musaliar College of Engineering
Kerala

JULY 2023

DECLARATION

I undersigned hereby declare that the project report "APPRAISAL-GUIDED PROXIMAL POLICY OPTIMIZATION: MODELING PSYCHOLOGICAL DISORDERS IN DYNAMIC GRID WORLD", submitted for partial fulfillment of the requirements for the award of the degree of Master of Technology of the APJ Abdul Kalam Technological University, Kerala is a bonafide work done by me under the supervision of Prof. Chinnu Jacob. This submission represents my ideas in my own words and where ideas or words of others have been included, I have adequately and accurately cited and referenced the original sources. I also declare that I have adhered to the ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed as the basis for the award of any degree, diploma, or similar title of any other University.

Place: Kollam
Date:

Hari Prasad

Thangal Kunju Musaliar College of Engineering
Centre for Artificial Intelligence

C E R T I F I C A T E

This is to certify that, this report titled ***APPRAISAL-GUIDED PROXIMAL POLICY OPTIMIZATION: MODELING PSYCHOLOGICAL DISORDERS IN DYNAMIC GRID WORLD*** is a bonafide record of the **Project** presented by **HARI PRASAD (TKM21MEAI05)**, under our guidance and supervision, in partial fulfillment of the requirements for the award of the degree, **M.Tech in Mechanical Engineering (Artificial Intelligence)** in **APJ Abdul Kalam Technological University** .

Project Coordinator & Guide

Prof. Chinnu Jacob
Assistant Professor
Centre for Artificial Intelligence

Head of the Department

Dr. Imthias Ahamed T P
Professor
Centre for Artificial Intelligence

Internal Examiner

External Examiner

ACKNOWLEDGEMENT

A successful project is a fruitful culmination of efforts by many people, some directly involved and some others indirectly, by providing support and encouragement. Firstly I would like to thank the almighty for giving me the wisdom and grace for making my project a memorable one. I thank him for steering me to the shore of fulfillment under his protective wings.

I express my sincere gratitude to **Dr. T A Shahul Hameed** , Principal of T.K.M College of Engineering for giving me an opportunity to present my project. I would like to thank **Dr. Imthias Ahamed T P**, Professor and Head of the Department, Centre for Artificial Intelligence, TKMCE, for his constant support and encouragement throughout the project work.

With a profound sense of gratitude, I would like to express my heartfelt thanks to my guide and project coordinator **Prof. Chinnu Jacob**, Assistant Professor, Centre for Artificial Intelligence, TKMCE, for their expert guidance, cooperation, and immense encouragement. I also extend my thanks to the entire faculty and staff of the Centre for AI, TKMCE, who has encouraged me throughout this work.

I also express my thanks to my loving parents and friends, for their support and encouragement in the successful completion of this project work.

Hari Prasad

Abstract

The increasing integration of artificial intelligence (AI) across multiple domains has highlighted the significance of comprehending and replicating human-like cognitive processes in AI. The incorporation of emotional intelligence into AI agents enables the evaluation of their emotional stability, thereby enhancing their resilience and dependability in critical decision-making tasks. This work aims to develop a methodology for modeling psychological disorders using Reinforcement learning agents. Appraisal theory has been utilized to model cognitive appraisals of RL agents and train them in a dynamic grid world environment by developing an appraisal-guided Proximal Policy Optimization (PPO) algorithm. Further, numerous reward-shaping strategies to regulate the behavior of agents and hence simulate psychological disorders have been investigated. An in-depth comparison of various configurations of the modified PPO algorithm is carried out to identify variants that can simulate Anxiety disorder and Obsessive Compulsive Disorder (OCD) like behavior in agents. In addition, an analysis of the behavioral patterns of the agents in a series of complex test environments is conducted, to evaluate the symptoms of disorders. Consequently, an effort has been made to develop a variety of evaluation criteria and metrics for analyzing the behavior of agents. Finally, the future possibilities and scope of studying and analyzing the psychology of artificial agents within the contexts of AI and psychology are discussed.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The need for emotional agents arises from the growing integration of artificial intelligence (AI) in various domains, such as customer service, healthcare, and entertainment. Emotions play a vital role in human interactions and decision-making processes. By endowing AI agents with emotional capabilities, we can enhance their ability to understand and respond to human emotions, leading to more effective and empathetic interactions. Emotional agents have the potential to revolutionize the way humans engage with AI systems, creating more natural and intuitive interfaces. Therefore, research in developing emotionally intelligent agents is crucial to bridge the gap between AI systems and human users.

Studying the emotional stability of artificial intelligent agents is essential for several reasons. Firstly, emotional stability is a key characteristic in ensuring the reliability and predictability of AI systems. Unstable or unpredictable emotional responses in agents may result in inappropriate behavior, unreliable decision-making, and negative user experiences. By understanding and evaluating the emotional stability of AI agents, we can enhance their robustness and reliability, making them more trustworthy and suitable for real-world applications.

Studying psychological disorders in agents provides valuable insights into AI and human psychology, deepening the understanding of cognitive and emotional processes. AI researchers can develop accurate models of human behavior, enhancing the design of emotionally intelligent agents. Human psychologists can benefit from a controlled environment to investigate factors affecting well-being, informing therapeutic approaches. Markov Decision Process (MDP) agents offer advantages in Affective Modeling (AM), with minimal assumptions, versatile task handling, and learning capabilities like PPO algorithms, helping in simulating complex situations, and facilitating the evaluation of emotion theories.

This project report focuses on the development and evaluation of a Cognitive Appraisal guided Partial Cognitive Architecture for the modeling and assessment of psychological disorders in RL agents. The proposed approach combines insights from the fields of Reinforcement Learning and Cognitive Appraisal Theory and focuses on Emotion recognition and generation, and the need for emotionally stable agents. Incorporating cognitive appraisal processes into the agent's decision-making mechanisms, it aims to enhance their emotional intelligence and resilience. The study of emotions in learning agents is significant in the field of machine

learning (ML), as the incorporation of emotions has the potential to improve learning efficiency as shown by this work. Interactive machine learning and human-robot interaction (HRI) researchers can use emotions to better communicate the agent's internal state and foster user empathy. Communicating the agent's internal status helps.

This report is structured with a literature review section that delves into the theoretical framework, presenting an in-depth analysis of key concepts and existing research in relevant areas. It covers topics such as Reinforcement Learning, Proximal Policy Optimization (PPO), Emotional agents, Cognitive Appraisal Theory, Emotion recognition and generation, and the need for emotionally stable agents. Furthermore, it identifies the research gap and highlights the need for a novel approach that integrates cognitive appraisal mechanisms for evaluating psychological disorders in RL agents.

The methodology section elucidates the proposed approach, detailing the environmental setup, and various techniques used in the work such as self-attention mechanism, cognitive appraisals, Next Reward Estimation Model, stress level estimation, Reward shaping techniques, and appraisal-guided PPO. Each component of the methodology is described to provide a comprehensive understanding of the experimental setup and the mechanisms employed for evaluating psychological disorders. We also discuss how our system can be considered as a Partial cognitive architecture.

The subsequent section presents the results and discussions, which explore the generalization capability of the proposed approach, compare the performance with and without the inclusion of appraisal mechanisms, evaluate metrics, examine the occurrence of psychological disorders, and compare the overall performance against alternative approaches. This section critically analyzes the obtained results and provides insights into the strengths and limitations of the proposed cognitive appraisal-guided partial cognitive architecture. We discuss the various observations that provide evidence for behavioral patterns seen in certain psychological disorders. We use a variety of matrices to evaluate these behavioral patterns and also discuss the benefits of emotionally stable agents over a simple PPO-based agent using different testing environment scenarios. Finally, The work concludes with a summary of the key findings, implications, and potential future research directions.

By introducing a novel cognitive appraisal-guided partial cognitive architecture, this study seeks to contribute to the advancement of evaluating psychological disorders in RL agents. Through the incorporation of cognitive appraisal mechanisms, it aims to improve the emotional intelligence and resiliency of RL agents, preparing the way for more emotionally stable and adaptable intelligent systems. Additionally, this creates new opportunities in the fields of affective computing, RL-based autonomous agents, and computational psychology. This study focuses on the behavioral patterns observed in emotional agents in order to discover evidence for disorders such as anxiety and OCD. By doing so, we do not argue that the agents are intelligent enough to develop human-level emotions or even experience very complex psychological disorders, but rather that they imitate certain key emotional characteristics through behavioral patterns that resemble the emotions of human or animal intelligence.

Chapter 2

Literature Review

From a Psychology standpoint, emotion is defined as a response to a significant stimulus, marked by brain and body arousal and a subjective sensation, that elicits a tendency towards motivated action [1]. Additionally, emotions have been identified as complex feedback signals that influence behavior. This perspective focuses on the emotional feedback function and conscious emotional states may improve learning and behavior by encouraging reflection and feedback [2]. Each fundamental emotion is an action inclination from an evolutionary standpoint. Fear, coupled with a bad mood and stereotypical facial expression, helps the organism survive perilous situations by avoiding them [3].

In the literature, numerous theories of emotion have been studied. One of them is the Dimensional Emotion Theory [4], which implies an underlying affective space with two dimensions which are Valence, which indicates whether an emotion is positive or negative, and Arousal, which depicts the intensity of an emotion. This theory falls back on separating emotion categories such as Fear, anger, Surprise, Disgust, Joy, and disgust. Another major theory of emotion is the Component Process Model of emotion by Lazarus [5], which views emotions as the consequence of personal relevance-based evaluations (appraisals) of incoming stimuli. Typical examples of appraisal dimensions include valence, novelty, motivational relevance, goal congruence, and coping capacity. Distinct emotions correspond to distinct activation patterns of appraisal. For instance, anger is the result of evaluating a situation as detrimental to one's own objectives, attributing the emotion to the responsible actor, and experiencing at least some sense of control.

In the field of affective computing and modeling, various studies have focused on developing computational models of emotions. Symbolic architectures, such as the OCC (Ortony, Clore, and Collins) model, have been widely used to incorporate emotions into AI systems based on categorical emotions or appraisal theories [6]. These architectures enable the representation and manipulation of emotions in a symbolic manner. However, symbolic approaches have limitations when it comes to learning from exploration and feedback in unstructured tasks.

In contrast to symbolic architectures, reinforcement learning (RL) approaches offer the advantage of learning through interactions with the environment and receiving feedback signals. RL agents, based on the Markov Decision Process (MDP) framework, are capable of

autonomously learning optimal policies through trial and error [7]. This makes RL agents well-suited for handling unstructured tasks and exploring complex environments.

By combining the strengths of affective computing and RL, recent research has explored the integration of emotions into RL agents. For example, work by [8] introduced a computational model that combines the OCC model with RL algorithms to enable emotionally adaptive behavior in virtual agents. This approach allows agents to learn emotional responses based on environmental stimuli and their appraisal, enhancing their ability to interact with humans in more empathetic and socially appropriate ways.

While symbolic architectures provide valuable insights into emotion representation, the integration of RL techniques allows for more flexible learning and adaptation in unstructured environments. This combination holds promise for developing AI systems capable of both symbolic reasoning and learning from experience, bridging the gap between cognitive and affective aspects of intelligent behavior in unstructured tasks.

2.1 Reinforcement Learning

Computational Reinforcement Learning and Reinforcement Learning in psychology share similarities in their approach to understanding learning and decision-making. In both domains, agents interact with their environments, receiving rewards or punishments [9], and aim to optimize their behavior. Computational RL provides a formal mathematical framework for modeling these processes, drawing inspiration from RL theories in psychology. Both computational RL and RL in psychology emphasize the importance of exploration and exploitation, balancing the discovery of new options with the utilization of existing knowledge. By incorporating insights from RL in psychology, computational RL gains a deeper understanding of human and animal learning processes, informing the design of algorithms and mechanisms for more effective and human-like intelligent systems.

In the early 20th century, the behaviorist movement led by psychologists such as B.F. Skinner [10] laid the foundation for RL by emphasizing the importance of rewards and punishments in shaping behavior. Skinner's work on operant conditioning, particularly his experiments with animals and the concept of reinforcement, provided early insights into learning through trial and error.

The field of RL witnessed a significant breakthrough with the introduction of the mathematical framework known as Markov Decision Processes (MDPs) in the 1950s. The MDP framework, developed by Richard Bellman [11], provided a formalization of sequential decision-making under uncertainty and laid the groundwork for modeling and solving RL problems. In the late 20th century, advancements in RL accelerated with the introduction of influential algorithms such as Q-learning [12] and the development of Temporal Difference (TD) learning [7] methods. These algorithms enabled agents to learn optimal policies through iterative updates based on observed rewards and state transitions.

A Markov Decision Process (MDP) is a mathematical framework used to model decision-

making in situations with sequential interactions and uncertainty [13]. In an MDP, an agent interacts with an environment over discrete time steps, making decisions based on the current state and receiving feedback in the form of rewards or punishments. The core concept of an MDP is the Markov property, which states that the future state depends only on the current state and the action taken, independent of the past history.

Formally, an MDP is defined by a tuple comprising a set of states (S), a set of actions (A), transition probabilities (T), immediate rewards (r), and discount factors (γ). The transition probabilities describe the likelihood of moving from one state to another when a particular action is taken. Immediate rewards indicate the immediate benefit or cost associated with transitioning to a new state. The discount factor determines the trade-off between immediate rewards and future rewards, allowing for long-term planning. The goal of the agent is to find a policy $\pi : S \longrightarrow P(A)$ that maximizes the expected discounted return. A typical MDP based Reinforcement learning agent-environment interaction can be explained by Figure

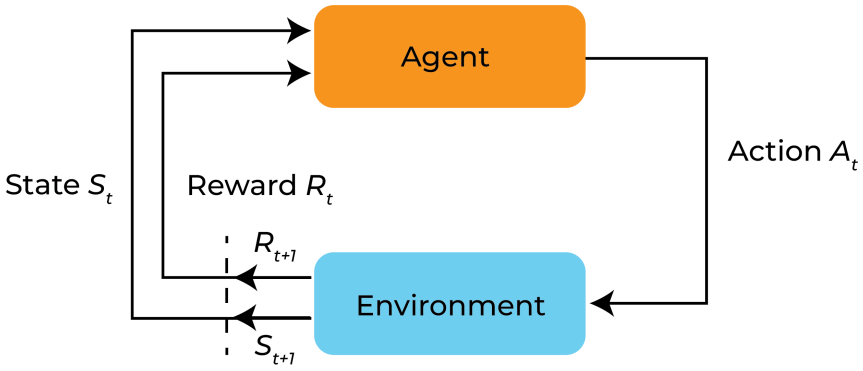


Figure 2.1: Agent and Environment interactions.

In Reinforcement Learning, the value function is a fundamental concept that quantifies the expected cumulative rewards an agent can achieve from a given state or state-action pair. The value function serves as a guide for decision-making, indicating the desirability or utility of being in a particular state or taking a specific action. There are two types of value functions: the state value function ($V(s)$) and the action value function ($Q(s, a)$). The state value function represents the expected cumulative rewards an agent can obtain when starting from a particular state s and following a given policy π. On the other hand, the action value function represents the expected cumulative rewards an agent can achieve when starting from state s, taking action a, and then following a given policy π.

The Bellman equation is a fundamental equation that characterizes the relationship between the value function and the underlying dynamics of the RL environment. The Bellman equation is expressed in equations 2.1 and 2.2:

$$V(s) = E[R_{t+1} + \gamma V(S_{t+1})|S_t = s] \tag{2.1}$$

$$Q(s, a) = E[R_{t+1} + \gamma Q(S_{t+1}, A_{t+1})|S_t = s, A_t = a] \tag{2.2}$$

Here, R_{t+1} represents the immediate reward received after taking action a in state s at time step $t + 1$. S_t and A_t denote the state and action at time step t, respectively. γ (gamma) is the discount factor, which determines the importance of future rewards relative to immediate rewards. $V(S_{t+1})$ or $Q(S_{t+1}, A_{t+1})$ represents the value of the next state or state-action pair, respectively.

2.2 Proximal Policy Optimization

Proximal Policy Optimization (PPO) [14] is a popular reinforcement learning (RL) algorithm that has gained significant attention in recent years. Developed by OpenAI, PPO aims to address the challenges of policy optimization in RL by striking a balance between stability and sample efficiency. PPO is an on-policy algorithm that is applicable in environments with discrete or continuous action spaces.

PPO builds upon the foundation of policy optimization, which involves finding the best policy for an RL agent to maximize cumulative rewards in an environment. Traditional policy optimization methods often faced challenges in balancing exploration and exploitation, and in ensuring stable and reliable learning. PPO addresses these concerns by utilizing a surrogate objective function that ensures policy updates are within a certain range, preventing drastic policy changes that could lead to instability.

The development of PPO can be traced back to previous advancements in policy optimization algorithms. One influential precursor is the Trust Region Policy Optimization (TRPO) algorithm proposed by [15]. TRPO introduced a policy update approach that constrained the maximum change to the policy distribution. Building upon TRPO, PPO further refined the approach by simplifying the optimization objective and providing a more efficient and sample-friendly implementation.

PPO is a family of first-order methods that employ a few additional techniques to keep new policies near the old. PPO methods are substantially easier to implement and appear to perform at least as well as TRPO based on empirical evidence. There are two primary variants of PPO. PPO-Penalty works by penalizing the KL-divergence in the objective function rather than putting it a hard constraint, PPO-Penalty approximates a KL-constrained update like TRPO, and it scales the penalty coefficient automatically during training. On the other hand, PPO-Clip's objective contains neither a KL-divergence term nor any other constraint. Instead, specialized clipping in the objective function is used to eliminate incentives for the new policy to depart from the old policy.

PPO trains a stochastic policy which indicates that the system explores by sampling actions in accordance with the most recent version of its stochastic policy. The degree of randomness in action selection is contingent upon both initial conditions and training procedures. Over the duration of the training, the policy typically turns less random as the

update rule motivates it to exploit previously discovered rewards. This may result in the policy becoming caught in a local optimum. Policy gradient algorithms are a class of reinforcement learning algorithms that operate by estimating the gradient of the policy and subsequently applying a gradient ascent algorithm to update the policy parameters. The key aspect of these algorithms is the estimation of the gradients, which is typically done using a Monte Carlo (MC) approach. In this approach, the policy is repeatedly executed in the environment to gather samples that are used to estimate the policy gradient 2.4 and update the policy.

$$J(\theta) = E_{T\ \pi_\theta T}\left[\sum_t R(s_t, a_t)\right] = E_{T\ \pi_\theta T}\big[R(T)\big] \tag{2.3}$$

$$\Delta_\theta J(\theta) = E_{T\ \pi_\theta T}\left[\left(\sum_{t=1}^{T} \Delta_\theta \log \pi_\theta(a_t|s_t)\right) R(T)\right] \tag{2.4}$$

During the execution of the policy in the environment, the algorithm collects trajectories by iteratively selecting actions based on the current policy. Each trajectory consists of a sequence of states, actions, and rewards experienced by the agent. By accumulating multiple trajectories, the algorithm obtains a set of samples that can be used to estimate the policy gradient. To estimate the policy gradient, the collected samples are processed by calculating the gradient of the expected return with respect to the policy parameters. This gradient represents the direction in which the policy parameters should be updated to improve the expected return. Various techniques, such as the likelihood ratio method or the score function method, can be employed to estimate the policy gradient using the collected samples.

The advantage function 2.7 quantifies the relative quality of an action in comparison to other available actions within a given state. It distinguishes good actions by assigning positive rewards and bad actions by assigning negative rewards. Consequently, it is necessary to estimate the average reward of the state, which is referred to as the value function

$$Q^\pi(s, a) = \sum_t E_{\pi_\theta}[R(s_t, a_t)|s, a] \tag{2.5}$$

$$V^\pi(s) = \sum_t E_{\pi_\theta}[R(s_t, a_t)|s] \tag{2.6}$$

$$A^\pi(s, a) = Q^\pi(s, a) - V^\pi(s) \tag{2.7}$$

Once the policy gradient is estimated, a gradient ascent algorithm, such as stochastic gradient ascent or natural gradient ascent, is applied to update the policy parameters. This update process aims to iteratively improve the policy by moving it towards regions of the parameter space that yield higher expected returns. By repeating this process of trajectory collection, policy gradient estimation, and policy parameter updates, the algorithm gradually improves the policy performance. The ability to estimate the policy gradient from sampled trajectories makes policy gradient algorithms suitable for both continuous and discrete action

spaces. Furthermore, the MC nature of the gradient estimation allows these algorithms to handle stochastic and non-differentiable environments. PPO-clip updates policies by taking multiple steps of (usually minibatch) SGD to maximize the objective in equation 2.8. The overall PPO training process and components are explained in figure 2.2.

$$\theta_{k+1} = \underset{\theta}{\operatorname{argmax}} \frac{1}{|\mathcal{D}_k|T} \sum_{\tau=\mathcal{D}_k} \sum_{t=0}^{T} \left(\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_k}(a_t|s_t)} A^{\pi_{\theta_k}}(s_t, a_t), g(\epsilon, A^{\pi_{\theta_k}}(s_t, a_t)) \right) \tag{2.8}$$

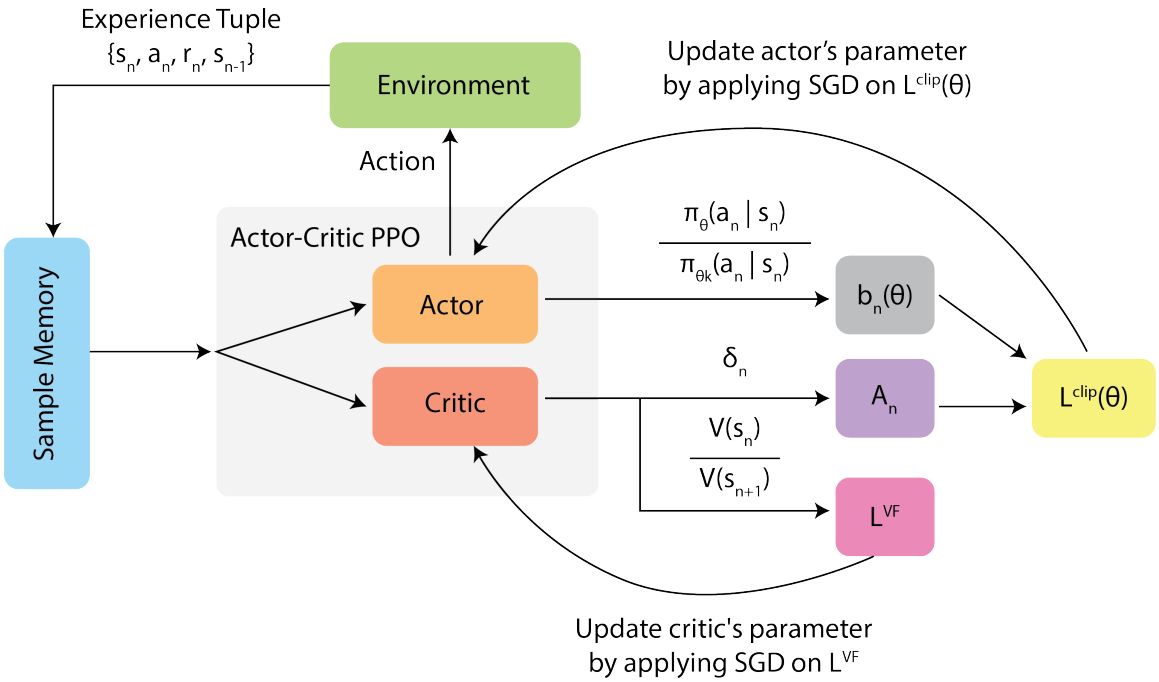


Figure 2.2: PPO Block diagram.

The robustness and stability of PPO make it well-suited for modeling emotional agents, as it enables the agents to learn and adapt their emotional responses over time. By incorporating emotional dimensions into the policy optimization process, PPO allows agents to make decisions based on both emotional cues and expected rewards. This integration can contribute to the development of more realistic and emotionally intelligent agents, facilitating more engaging and effective interactions with users.

2.3 Cognitive Appraisal Theory

Cognitive Appraisal Theory, proposed in the 1960s by Richard Lazarus [5], is a prominent psychological framework that strives to explain how individuals interpret and evaluate situations to generate affective responses. This theory suggests that emotions are not elicited directly by events, but rather by an individual's cognitive evaluation of those events. Cognitive evaluations entail the evaluation of the personal significance and repercussions of a

situation, taking into account factors such as goal relevance, coping ability, and congruence with personal values.

According to Cognitive Appraisal Theory, there are two basic phases to the appraisal process: primary appraisal and secondary appraisal. Primary evaluation is the categorization of an event's significance for one's well-being as either positive, negative, or irrelevant. This evaluation helps determine the potential emotional response. The secondary evaluation focuses on evaluating available coping strategies and resources to manage the situation. It entails evaluating one's ability to manage the situation and the possible outcomes of various coping strategies.

Cognitive Appraisal Theory has contributed significantly to the comprehension of affective experiences in a variety of contexts. It has been implemented in fields including stress, health, decision-making, and social interactions. Lazarus and Folkman extended the theory to the domain of stress [16], highlighting the significance of cognitive evaluations in determining stress levels and coping strategies. In addition, the theory has been applied to studies of emotion regulation [17], emotion and memory [18], and emotional experiences in social interactions [19]. According to the Transactional Model of Stress and Coping of Richard Lazarus, as shown in Fig. **??**, emotions are the product of ongoing transactions between individuals and their environment, rather than being solely driven by external stimuli.

Primary appraisal is the initial evaluation of an event's relevance to an individual's goals and well-being. This appraisal process categorizes the event as either irrelevant, positive (indicating potential gain or benefit), or negative (indicating potential harm or threat). The primary appraisal directly influences the emotional response. For example, if an event is appraised as positive, it may elicit feelings of joy or excitement, while a negative appraisal may lead to emotions such as fear or anger.

Following the primary appraisal, secondary appraisal focuses on evaluating one's coping resources and options in dealing with the event. This appraisal assesses the perceived ability to manage and control the situation, including the availability of social support, personal skills, and available strategies. The secondary appraisal is crucial in determining the selection of coping strategies and the extent of perceived stress. Higher perceived coping potential can reduce stress levels, while lower perceived coping potential may result in increased stress.

Stress, in the context of the transactional model, is the result of an imbalance between perceived demands (negative appraisals) and perceived coping resources (secondary appraisals). When individuals appraise an event as threatening or harmful and perceive a lack of sufficient coping resources, they experience stress. Coping mechanisms come into play as individuals attempt to manage or adapt to the stressors they encounter. Coping strategies can be problem-focused, aimed at addressing the root cause of stress, or emotion-focused, targeting the emotional distress associated with the stressor.

Reappraisal, an important aspect of the transactional model, involves re-evaluating the meaning and significance of an event or situation. It occurs as individuals gather additional information, reconsider their initial appraisals, and update their emotional responses and

coping strategies accordingly. Reappraisal allows for the adjustment of appraisals based on changing circumstances, leading to potential shifts in emotions and coping approaches.

The transactional model provides a comprehensive understanding of the dynamic nature of appraisal, emotion, and coping processes, highlighting the continuous interactions between individuals and their environment. This model has been influential in various fields, including psychology, stress research, emotion regulation, and the development of computational models of emotional agents.

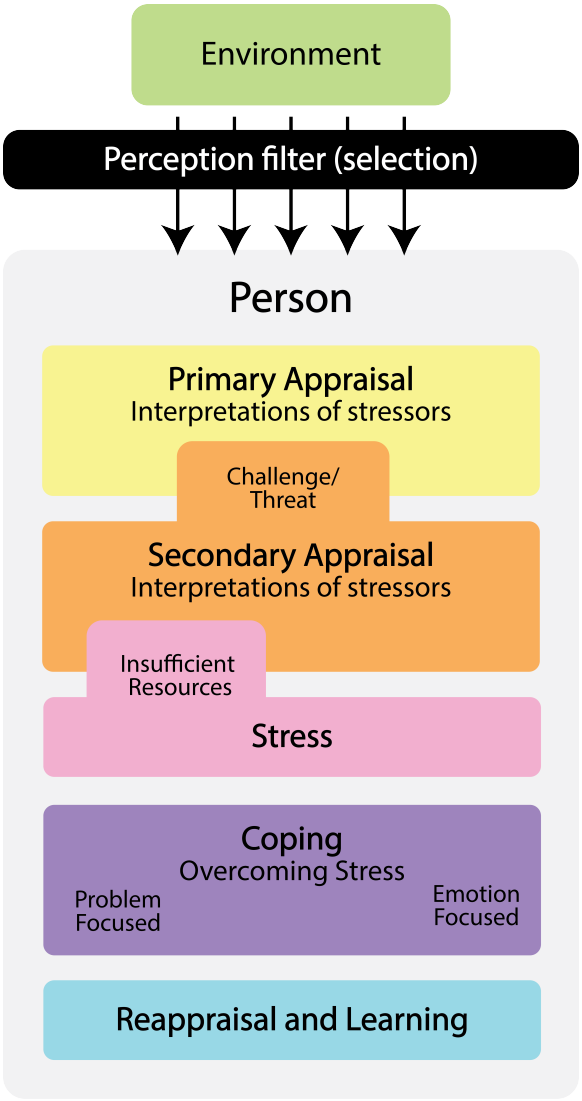


Figure 2.3: Transactional Model of Stress and Coping of Richard Lazarus

The cognitive appraisal framework has also been implemented in the fields of artificial intelligence and affective computing, in which researchers seek to create computational models that replicate human-like emotional processes, by incorporating cognitive evaluations into virtual agents to facilitate emotionally adaptive behavior [8]. These agents were able

to evaluate the significance of events and generate appropriate affective responses in social interactions by incorporating cognitive appraisal theory.

2.4 Emotion Elicitation and need for stability

Emotion elicitation can be broadly categorized into extrinsic/homeostatic, intrinsic/appraisal, value function, reward-based, and hard-wired [20]. In this work, we focus only on the intrinsic/appraisal category. Appraisal theory is a significant psychological emotion theory. Appraisals are domain-independent elements that provide a stimulus with (affective) meaning. As such, they serve as a premise for eliciting emotions, as various combinations of appraisal dimensions are associated with distinct emotions. Motivational relevance, an appraisal has been modeled as the inverse of distance to the goal (s_g), as shown in the equation 2.11.

$$\zeta_{relevance}(s) = \frac{1}{d(s, s_g)} \tag{2.9}$$

A different study maintains an ensemble of transition models by stochastically adding new data to each model and deriving 'model uncertainty' from the KL divergence as a measure of the distance between two probability distributions between the ensemble model's predictions, as shown in the equation. 2.10.

$$\zeta_{uncertainity}(s, a) = \sum_{i \neq j} D_{KL}[T_i(s'|s, a)||T_j(s'|s, a)] \tag{2.10}$$

Another evaluation formulation is the 'novelty', which identifies the state-action pairs with the shortest L1 distance from a historical observation.

$$\zeta_{novelty}(s, a) = \min_{<s_i, a_i> \in g} || < s, a >, < s_i, a_i > ||_1 \tag{2.11}$$

Other implementations of appraisal dimensions include Social fairness/attachment, Model uncertainty, Intrinsic pleasantness, Control/Power, etc. A few explicit social dimensions are also encountered, such as social equity and social accountability, although the latter necessitates symbolic reasoning in addition to the RL paradigm. Some appraisal-based dimensions require cognitive reasoning and are more difficult to implement. However, novelty, motivational relevance, and intrinsic pleasantness are frequently utilized as dimensions. Typically, these characteristics necessitate learned transition functions, recency characteristics, or forward planning procedures over the model space. Additionally, a single concept has been interpreted in multiple ways throughout the literature.

Emotional stability plays a crucial role in the performance and behavior of RL agents, especially in scenarios where adaptability, resilience, and effective decision-making are essential. Emotional stability allows RL agents to maintain consistent and appropriate responses in the face of uncertain and dynamic environments. In complex tasks such as autonomous driving, robotics, or virtual assistants, emotional stability enables agents to handle unexpected events, cope with failures or setbacks, and make sound decisions under pressure.

One area where emotional stability is particularly valuable is in social interactions. RL agents that interact with humans need to exhibit emotionally appropriate behavior to establish rapport, trust, and effective communication. Emotional stability helps agents maintain consistent emotional responses, avoiding erratic or extreme reactions that could lead to misunderstandings or discomfort for human users. By incorporating emotional stability, RL agents can provide more engaging and empathetic interactions, enhancing user satisfaction and acceptance.

Furthermore, emotional stability is essential in reinforcement learning problems that involve long-term planning and delayed rewards. RL agents with emotional stability can exhibit patience and persistence in pursuing goals, effectively balancing immediate rewards with long-term objectives. This ability is particularly valuable in domains such as finance, healthcare, or resource management, where decisions made by RL agents have long-lasting consequences and require consideration of future outcomes.

A study on psychological disorders in RL agents through an emotional model such as appraisal theory is essential for several reasons. Firstly, it provides insights into the impact of psychological disorders on the behavior and decision-making processes of RL agents. By incorporating appraisal theory, researchers can examine how disorders affect the agents' cognitive appraisals, emotional responses, coping strategies, and overall performance. This understanding can lead to the development of interventions and strategies to address and mitigate the impact of psychological disorders in RL agents.

Secondly, studying psychological disorders in RL agents contributes to the broader field of artificial intelligence and human psychology. RL agents, with their ability to simulate human-like behavior and decision-making processes, provide a valuable platform for studying and testing psychological theories and interventions. By exploring the manifestation of psychological disorders in RL agents, researchers can gain insights into the underlying mechanisms of these disorders, contributing to the advancement of psychological knowledge and therapeutic approaches.

Chapter 3

Methodology

This chapter presents the methodology employed to develop and evaluate the Cognitive Appraisal guided Partial Cognitive Architecture for the assessment of psychological disorders in RL agents. The methodology encompasses various sections, including the environmental setup, attention mechanism, appraisals, NRE Model, stress level estimation, reward shaping, and appraisal-guided Proximal Policy Optimization (PPO). These sections collectively explain the integration of cognitive appraisal processes into the decision-making mechanisms of the RL agents.

Integrating appraisals into RL agents for decision-making and emotion elicitation poses several challenges due to the inherent complexity of both appraisal processes and RL algorithms. Firstly, appraisals involve subjective evaluations of the personal significance and implications of events, which are inherently difficult to quantify and represent computationally. Appraisals require the interpretation of contextual information, personal beliefs, and values, making it challenging to define objective measures for capturing such subjective processes in an RL agent.

Secondly, RL algorithms typically rely on numerical state representations and reward signals, whereas appraisals involve the assessment of qualitative aspects such as goal relevance, coping potential, and congruence with personal values. Bridging the gap between the qualitative nature of appraisals and the quantitative nature of RL poses a significant technical challenge. It requires developing computational frameworks or models that can capture and represent the multifaceted dimensions of appraisals in a meaningful and actionable way.

Another challenge is the integration of appraisals into the RL decision-making process. Appraisals are dynamic and iterative processes that involve continuous re-evaluation and adjustment of the significance and implications of events. Incorporating this dynamic nature into the decision-making loop of RL agents requires developing algorithms that can efficiently update appraisals based on new information, while simultaneously optimizing the agent's policy. Balancing the computational demands of appraisal updates and decision-making poses a significant technical hurdle.

Furthermore, evaluating psychological disorders in RL agents is challenging due to the complexity and diversity of these disorders. Psychological disorders involve intricate in-

terplays between cognitive processes, emotions, and behavioral patterns. Translating the manifestations of psychological disorders, such as distorted appraisals, emotional dysregulation, or maladaptive coping strategies, into computational models that can be effectively evaluated in RL agents is a formidable task.

In addition, the evaluation of psychological disorders in RL agents requires reliable and valid metrics for determining the presence and severity of these disorders. This work focuses on simulating the symptoms that real-world natural agents experience. Our focus is on highlighting a possibility and a new research direction in the study of the psychology of RL agents. Thus, we do not attempt to simulate a complete psychological state, but rather a fragmentary one that still provides traces of behavioral patterns observed in such psychological disorders.

3.1 Environment

In this work, a suitable environment that would showcase the behavior of agents in a simple and effective manner has been chosen. The Minigrid [21] environment with dynamic obstacles has been chosen and modified according to the requirements of this work. The dynamic grid world with obstacles is chosen specifically since it would be easier to evaluate the psychological state of the agent easily. The behavior of the agent can be evaluated by monitoring the agent's trajectory, its proximity to the goal, its behavior near the goal, etc. This also gives options to introduce variations on the environment without compromising those benefits.

The grid world comprises 5 elements categorized as Agents, Goals, Obstacles, Walls, and Empty spaces. In the observation space, they are encoded as integer values as shown in the table

Table 3.1: Appraisal Weights for stress level calculation

Elements	**Integer value**	**Color**
Agent	10	Red
Goal	8	Green
Wall	2	Gray
Empty Space	1	Black
Obstacle	6	Blue

The observation space is defined by the agent's view size. The agent's view is a region in the grid world around the agent's current position, where the agent can observe. The view has a dimension of (nxnx3), which is a 3-dimensional array of size (nxn). The agent can only use this information to navigate around the environment. For uniformity across all experiments, we have chosen the view size to be 7x7 for all experiments, which gives enough information for the agent to learn from.

The action space of the agent defines the possible actions allowed by the environment for the agent to make in order to navigate around. In this work, the action space consists of 3

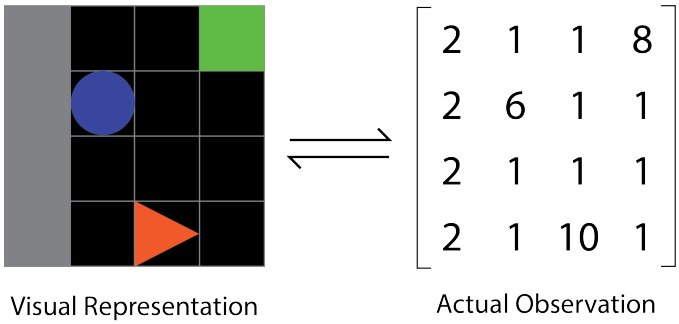


Figure 3.1: Grid world elements with action representation on right and visualization on left.

Table 3.2: Parameters that define the grid world environment

Appraisal	**Value**
Grid Size	10
Agent view Size	7
Agent start position	0 (random)
Agent start direction	0 (random)
Number of obstacles	7
Max. Steps	100
Dynamic walls (Maze)	False
Dynamic goal-starting position	True
Moving Goal	True
Moving Obstacles	True
Wall split ratio	2
See-through walls	True

discrete actions (left, right, and forward). Each time an agent needs to move to the cell on the right or left side, it has to make 2 actions in a sequence, first turn right or left and then make a step forward.

The reward system in our environment is sparse and limited to the fact that the agent only receives a cumulative return once an episode is complete. If the agent wins in an episode, it receives a return equal to the complement of the ratio of the steps taken to reach the goal to the maximum possible steps. If an agent fails by running into an obstacle or reaches the maximum number of steps, then it receives a return of -1.

3.1.1 Environment parameters

Various parameters have been used to design the environment and modify it according to the need for different experiments. in general, the parameters that control the environment and its initial conditions are shown in the table.

An example set of parameters used in one of the test setups is shown in the table. The

grid size is the parameter that defines the size of the environment. In this work, a 10x10 grid and an 11x11 grid have been chosen for experiments. The agents' view size defines the observation space dimension, which is used for training the agent. Throughout this work, a view size of 7x7 is chosen.

The agent's start position and start direction decide at which state in the grid, the agent starts from and which direction it should face respectively. Both these parameters are set to 0 which would correspond to random starting position and direction. Max steps is another parameter that limits the total number of steps an agent can take in an episode. We have chosen the values 100 and 400 in our experiments.

Number of obstacles defines the total number of obstacles that are allowed in an environment. Along with the parameter, Moving obstacles when toggled would make the obstacles move around in the environment randomly. The motion of the obstacles is set in such a way that it does not move itself to the states occupied by the goal or the agent. but the agent can run into obstacles. The dynamic goal-starting position when toggled, allows the goal to randomly start at any position that is not occupied by the agent or an obstacle. the moving goal parameter allows the goal to be either stationary or randomly move around.

In order for carrying out more challenging experiments for analyzing the behavior and also evaluate the generalization capabilities of the agents, we have also used environments with dynamic walls, which are like mazes that change randomly at the start of each episode. The parameter dynamic walls toggle this feature and in order for this to work, an odd grid size is to be chosen and we have used a size of 11x11. The wall split ratio defines the complexity of the maze, where a value of 2 would represent a split of 2 in horizontal and vertical directions resulting in 4 chambers as shown in Fig.

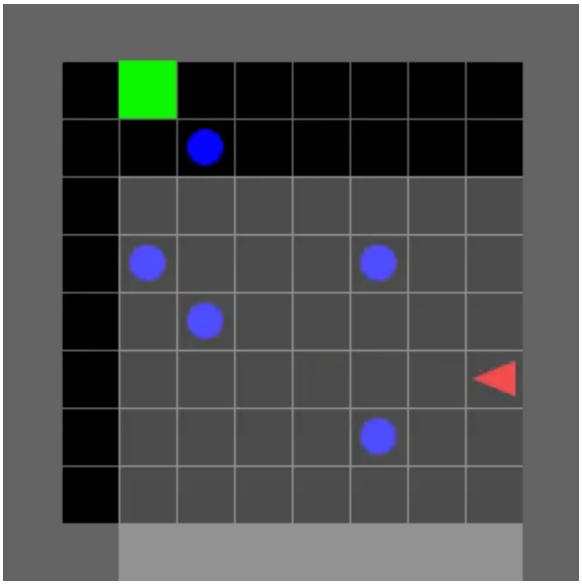


Figure 3.2: Grid world with moving goal and dynamic obstacles.

3.2 Attention Mechanism

Attention mechanisms enable models to focus selectively on relevant information while disregarding irrelevant or noisy inputs. This selective focus mimics the attentional processes observed in human cognition, where attention plays a vital role in perception, memory, and decision-making.

Self-attention, also known as intra-attention or scaled dot-product attention, is a specific type of attention mechanism that has gained significant attention in recent years. It allows a model to capture the dependencies between different elements or tokens within a sequence, such as words in a sentence or pixels in an image, by attending to the relationships among the elements themselves.

Unlike traditional attention mechanisms that attend to external context or sources, self-attention operates within a single sequence, enabling the model to capture long-range dependencies and identify important interactions among elements. This self-attention mechanism has proven to be highly effective in tasks involving sequential data, such as machine translation, sentiment analysis, and text summarization.

The core idea behind self-attention is to transform each element of the sequence into three distinct representations: query, key, and value. These representations are derived through linear transformations of the original elements, enabling the model to calculate the relevance or similarity between each query and all the keys in the sequence. The resulting attention weights are then used to weigh the values, producing a weighted sum that represents the attended information.

In recent years, self-attention has been prominently used in state-of-the-art models such as the Transformer, which revolutionized machine translation and achieved remarkable performance in various natural language processing tasks[22], [23], [24], [25]. In this work, Self-attention is used in the Convolution Block of the Agent model. In self-attention, also known as the Transformer attention mechanism, the terms "key," "query," and "value" are components used to compute the attention weights and generate the output representation.

- **Key:** The key represents the information that is used to calculate the similarity between different elements in the input sequence. It is obtained by multiplying the input sequence with a learnable weight matrix. The key matrix helps determine the relevance or importance of each element in the input sequence with respect to the other elements.
- **Query:** The query is used to compare against the keys to calculate the attention weights. It is obtained by multiplying the input sequence with another learnable weight matrix. The query matrix captures the information that we want to retrieve or focus on from the input sequence.
- **Value:** The value represents the actual information associated with each element in the input sequence. It is obtained by multiplying the input sequence with yet another learnable weight matrix. The value matrix contains the content or features of each element in the input sequence.

The attention mechanism computes the attention weights by calculating the dot product between the query and key vectors, followed by applying a SoftMax function to obtain the normalized attention scores. These attention scores are then used to weigh the corresponding value vectors, resulting in a weighted sum that represents the output representation of the self-attention mechanism.

In this work, the key, query, and value are derived from the input tensor using convolution layers with different kernel sizes, as shown in Fig. The query and key are down-sampled to a smaller dimension compared to the input dimension, while the value retains the original dimension. The forward method then processes the derived key, query, and value. The query and key tensors are reshaped by flattening the spatial dimensions and permuting the dimensions appropriately. The value tensor is reshaped by only flattening the spatial dimensions.

The attention scores are computed using the SoftMax of the dot product between the query and key tensors. These attention scores are used to weigh the value and generate the attended output representation. The attended output representation is then reshaped back to the original spatial dimensions of the input tensor. Finally, the attended output is rescaled using the learnable parameter gamma and added to the original input tensor.

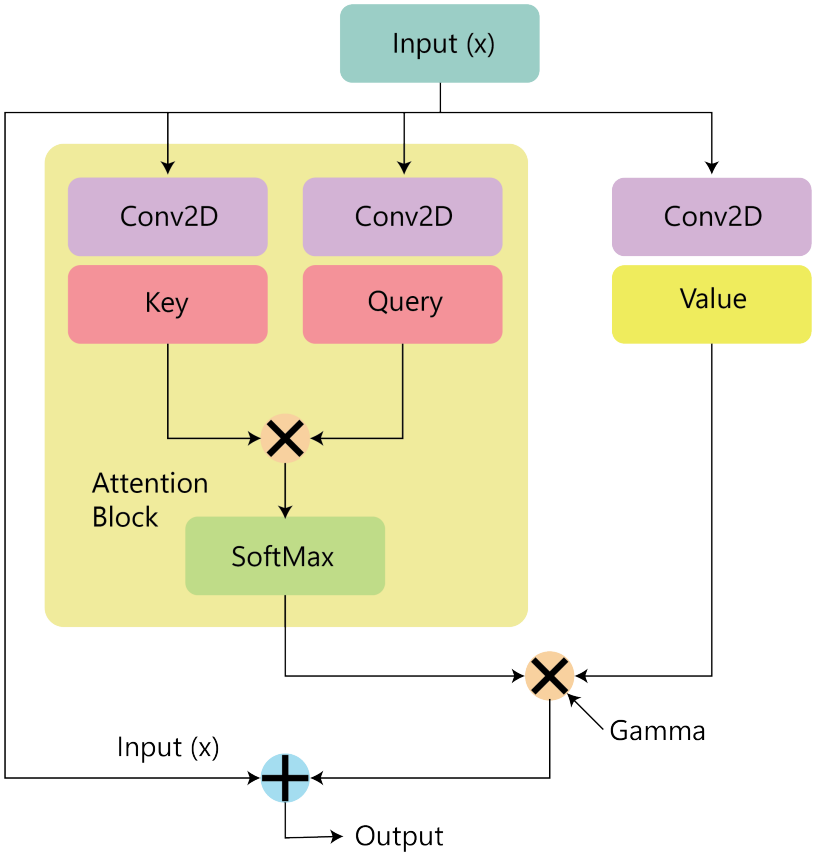


Figure 3.3: Architecture of Self-Attention block used in Covolution section of the Agent model

3.3 Cognitive Appraisals

Emotions play a fundamental role in human experience, influencing cognition, behavior, and overall psychological well-being. The process of emotion generation has been the subject of extensive research, with cognitive appraisals emerging as a crucial determinant of emotional responses. Cognitive appraisals refer to the evaluative cognitive processes through which individuals interpret and ascribe personal significance to events, stimuli, or situations.

Appraisal theories posit that emotions are not direct responses to external stimuli but are instead generated through cognitive evaluations of the perceived significance of these stimuli. According to these theories, cognitive appraisals serve as mediators between the external world and the emotional experience, shaping the quality and intensity of emotional responses.

Cognitive appraisals can be classified into primary and secondary appraisals. Primary appraisals involve the assessment of the relevance, goal congruence, and implications of a given event for one's well-being. Secondary appraisals, on the other hand, focus on the evaluation of one's coping resources and potential for managing or altering the situation. These appraisals are highly subjective and can vary across individuals and contexts, leading to different emotional outcomes.

The influence of Cognitive Appraisals on Emotion Generation can be described in terms of intensity, differentiation, and regulation of emotions as follows

- **Emotional Intensity:** Cognitive appraisals significantly contribute to the intensity of emotional experiences. Appraisals characterized by high relevance and negative valence tend to elicit more intense emotions, while appraisals with low relevance and positive valence typically result in milder emotional responses.
- **Emotional Differentiation:** Cognitive appraisals play a crucial role in differentiating between various emotional states. The subjective interpretation of events through appraisals provides a basis for distinguishing between emotions with similar physiological arousal but different cognitive representations.
- **Emotional Regulation:** Cognitive appraisals are central to the process of emotion regulation. Individuals can reinterpret or reappraise a situation to alter its emotional significance, thereby influencing their emotional experience and subsequent behavior.

In this work, 6 cognitive appraisal variables, including Motivational Relevance, Certainty, Novelty, Goal Congruence, Coping Potential, and Anticipation are evaluated. The appraisals represent how the agent comprehends the environment from a cognitive point of view. These appraisal variables are estimated at each step the agent takes and is fed into the actor and critic networks for action selection and value estimation. The range of the appraisals is re-scaled to (0,1). Each appraisal influences the action selection process of the agent by incorporating inputs from both actor and critic networks. The 6 cognitive appraisals used in this work are as follows.

3.3.1 Motivational Relevance

Motivational relevance, within the framework of cognitive appraisals, pertains to the subjective evaluation of the personal significance and impact of a stimulus or event on an individual's goals, needs, and values. It involves assessing the relevance of a situation for achieving desired outcomes or avoiding undesired ones. Motivational relevance influences the intensity of emotional responses and directs attention and behavioral responses.

Factors such as personal goals, values, and contextual factors contribute to the appraisal of motivational relevance. Understanding motivational relevance in cognitive appraisals provides insights into the role of personal motives and values in shaping emotional experiences.

In this work, the motivational relevance is calculated as the distance between the current position of the agent in the grid world to the position of the goal state in the grid world. Mathematically, Motivational Relevance is defined as the complement of the Manhattan distance between the agent and goal in the grid world, as shown in the equation 3.1

$$MotivationalRelevance = 1 - \frac{(|x_a - x_g| + |y_a - y_g|) - 1}{2\,(w-1)} \tag{3.1}$$

Here, (x_a, y_a) represents the agent's position and $(x_g,\ y_g)$ represent the goal position in the grid world. The value w denotes the width of the grid.

3.3.2 Certainty

Certainty refers to the subjective evaluation or perception of the level of certainty or predictability associated with a particular event, stimulus, or situation. Certainty appraisal involves assessing the level of confidence or belief in the outcome or consequences of an event. It encompasses an individual's subjective evaluation of the likelihood or probability of a specific outcome occurring. For example, if someone appraises an upcoming job interview as highly certain, they may believe that their performance will strongly influence the outcome, leading to feelings of anticipation or anxiety. Conversely, if the certainty appraisal is low, they may perceive the situation as unpredictable, resulting in feelings of uncertainty or ambiguity.

In this work, Certainty is calculated by measuring the entropy of the probability distribution obtained by applying the SoftMax function to predicted logits. Higher the entropy, the more uncertain the situation is and the lower the certainty value. The Certainty is estimated using the following equation 3.2.

$$Certainty = 1 - \frac{-\sum p \log(p)}{1 + (-sump \log(p)} \tag{3.2}$$

Where $p = SoftMax(logits)$ represents the SoftMax output of predicted logits. Here the Complement of the entropy is taken and normalized and rescaled to the range (0,1).

3.3.3 Novelty

Novelty refers to the evaluation of the degree of newness or unfamiliarity associated with a stimulus, event, or situation. It represents the cognitive dimension through which individuals assess the novelty or uniqueness of a particular experience.

Here, Novelty is estimated by taking the KL divergence between the predicted action probabilities and uniform distribution with the same dimensions as the action probabilities as in equation 3.3. The target distribution Q is a uniform distribution, where each action or class has an equal probability. The predicted distribution P is obtained by applying the SoftMax function to the logits. The KL divergence measures the difference between the two distributions.

When the KL divergence is higher, it indicates that the predicted distribution P deviates more from the uniform distribution Q. This suggests that the scenario or input is more novel because the model's predictions are significantly different from a uniform distribution, indicating a higher level of uncertainty or divergence.

$$Novelty = \frac{KL(Q||P)}{1 + KL(Q||P)} \tag{3.3}$$

3.3.4 Goal Congruence

Goal congruence appraisal involves the evaluation of the relevance and compatibility of the situation with one's goals. It encompasses an individual's subjective assessment of whether the stimulus or event is conducive to achieving or hindering their desired objectives. In this work, we estimate the goal congruence as the Euclidean distance from the current position of the agent to the goal position, if and only if the goal is visible to the agent, as seen in equation 3.4.

$$Goal\ Congruence = 1 - \frac{\sqrt{(x_a - x_g)^2 + (y_a - y_g)^2}}{\sqrt{\left(\frac{(n-1)}{2}\right)^2 + (n)^2}} \tag{3.4}$$

Here, (x_a, y_a) represents the agent's position and $(x_g,\ y_g)$ represent the goal position in the grid world. n is the view size of the agent in the grid world.

3.3.5 Coping Potential

Coping Potential refers to the perceived ability or resources an individual possesses to effectively manage and cope with a given stimulus, event, or situation. It represents the cognitive dimension through which individuals assess their capacity to handle or adapt to the demands and challenges presented by a specific circumstance.

in our environment, the coping potential refers to how well the agent believes that it can move ahead with its planned trajectory. In order to estimate the value, we calculate the

ratio of the number of obstacles, the agent sees in its view to the total number of obstacles in the environment as represented in the equation 3.5

$$Coping\ Potential = 1 - \frac{k_{obst}}{n_{obst} + \varepsilon} \tag{3.5}$$

Here the k_{obst} denotes the number of obstacles in the agent's view and n_{obst} denotes the total number of obstacles in the environment. The value ε is used to avoid division by zero.

3.3.6 Anticipation

In terms of cognitive appraisals, anticipation refers to the cognitive process through which individuals evaluate and project future outcomes or events. It represents the cognitive dimension through which individuals assess the likelihood, desirability, and potential implications of a future occurrence.

In this work, anticipation is measured as the complement of the Next Reward Estimation (NRE) error. A Neural Network is trained in predicting the next reward given the current observation and current action. The anticipation will be equal to the complement of the difference between the predicted reward and the actual reward as in equation 3.6.

$$Anticipation = 1 - [R_t - NRE(Obs_{t-1},\ a_{t-1})] \tag{3.6}$$

Where obs_{t-1} and a_{t-1} are the previous observation and action probabilities respectively.

3.4 Next Reward Estimation

The Next Reward Estimation (NRE) is a technique introduced in this work, in order to predict the next reward that the agent is expected to get, given the current observation and action taken in the current step. The concept is not new in the domain of Reinforcement Learning and has been explored in various scenarios across the literature [26], [27], [28]. In our work, the NRE is a critical way to estimate the anticipation of the agent, which is one of the 6 cognitive appraisals.

The agent class has a separate network for NRE which is a 3 Layer Dense Neural Network, with ReLU activations. It takes in the observation and the current action as input and predicts the reward for the next step. In the next step, the NRE error is calculated and backpropagated.

The complement of the difference between the actual reward obtained in a step and the NRE reward is taken as the anticipation value. The higher the difference, the lower the anticipation. This corresponds to the fact that the agent is not able to anticipate future rewards. The NRE error is calculated using the Mean Squared Error (MSE) function and is added to the PPO loss function for backpropagation. The overall block diagram of the NRE

component is described in Fig.

Practically, the NRE error cannot reach an absolute value of zero as the environment has many random processes such as the motion of the goal, the motion of obstacles, the starting position of the agent, starting position of the goal state, etc. A typical NRE loss graph is shown in Fig.

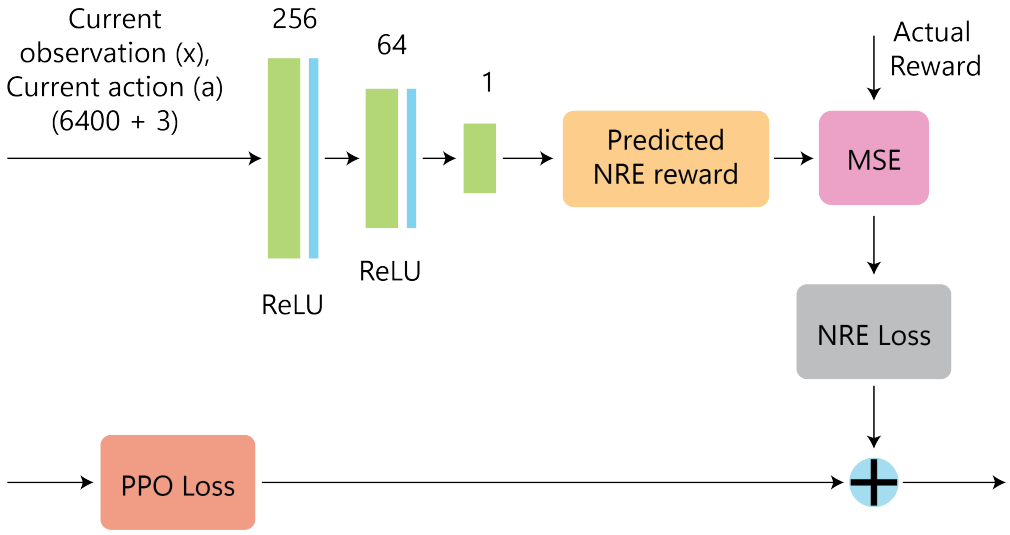


Figure 3.4: Next reward estimation block

3.5 Stress level Measurement

Stress level estimation in the context of evaluating the psychological state of agents within a grid world environment is a crucial and fundamental process. One approach to estimating the stress level involves observing the behavior of the agent within the environment. However, relying solely on observational data can be excessively intricate and challenging. To address this complexity, an alternative approach employed in this study is to utilize the concept of appraisal variables for estimating stress levels. These appraisal variables provide additional information that aids in assessing and understanding the agent's stress level.

In order to combine the values of all six cognitive appraisals effectively, a weighted sum approach is employed. The weights assigned to each cognitive appraisal are determined based on their relative significance toward the arousal of stress. These weights are approximated using existing literature on the topic and are tailored to suit the specific implementation requirements within the context of this particular study. By assigning appropriate weights to the different cognitive appraisals, a comprehensive assessment of the stress level can be achieved.

The stress level, alongside other metrics, plays a critical role in the evaluation of psychological disorders such as Anxiety and Obsessive-Compulsive Disorder (OCD) in agents. The

Table 3.3: Appraisal Weights for stress level calculation

Appraisal	**Weight**
Motivational Relevance	0.25
Novelty	0.05
Certainty	0.1
Goal Congruence	0.2
Coping Potential	0.35
Anticipation	0.05

measurement of stress level is essential for understanding and characterizing these psychological disorders, and it serves as a valuable indicator of the agents' psychological well-being. To facilitate further analysis and interpretation, the stress level is rescaled to a normalized range of values between 0 and 1. The stress level is calculated as shown in the equation.

$$Stress = \sum_{i=1}^{n} (1 - appraisal_i) * weight_i \tag{3.7}$$

3.6 Reward Shaping

Reward shaping in reinforcement learning (RL) refers to the process of modifying the rewards received by an agent during training in order to facilitate learning and improve performance. In this work, the environment has a predefined reward structure where the agent receives a reward of -1 if it fails to reach the goal state due to obstacles or reaching the maximum number of steps. On the other hand, if the agent successfully reaches the goal, the reward is determined by the complement of the ratio between the number of steps taken to reach the goal and the maximum allowed steps.

While this reward structure is sufficient to train the agent and achieve baseline performance over a large number of iterations (100,000 in this case), there is a concern regarding the sparsity of the rewards. Sparse rewards refer to situations where the agent receives rewards infrequently, making it challenging for the agent to learn the underlying dynamics and finer details of the environment.

To address this issue, different reward-shaping strategies using cognitive appraisals have been explored. Cognitive appraisals are estimates or assessments made by the agent during each step of the RL process. These appraisals represent various factors such as the agent's perception of the environment, its internal state, or the progress made toward achieving the goal.

The reward-shaping approach involves modifying the immediate reward obtained by the agent at each step using a reshaping factor derived from the cognitive appraisals made during that step. By incorporating these appraisals into the reward computation, this work aims to provide additional guidance to the agent, encouraging it to learn and explore the environment more effectively. On top of that, the agent is now controllable in terms of how

it gets trained by means of controlling its cognitive appraisals and also enhances its ability to capture finer features of the environment. The reward-shaping process in this work is explained in detail in Fig.

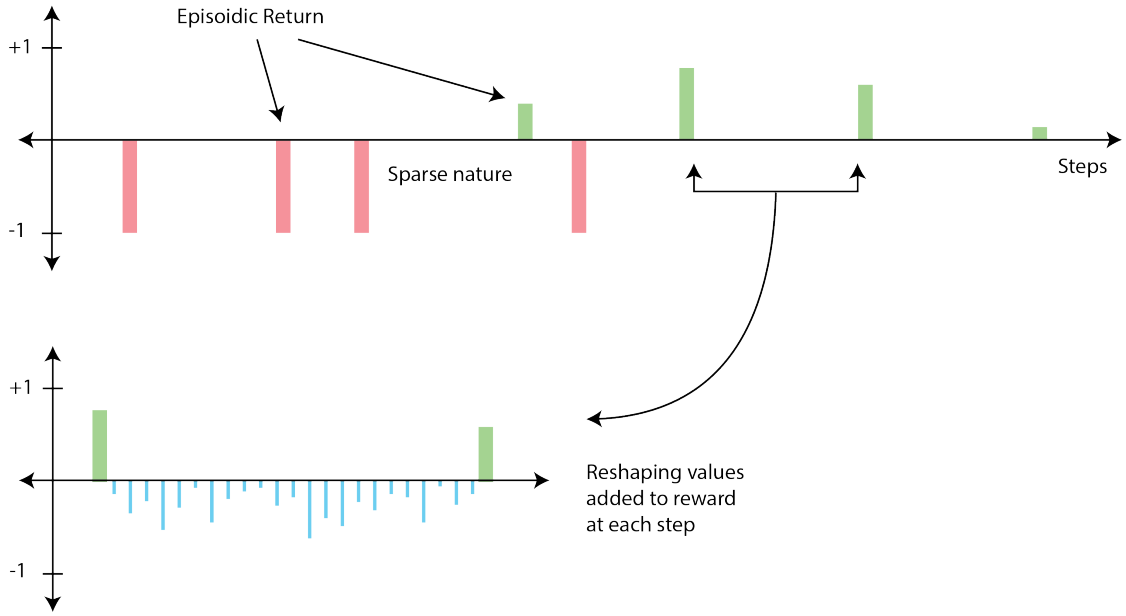


Figure 3.5: Sparse reward system and reshaping values added to reward at each step. The green bars represent the episodic return obtained during wins and the red represents -1 obtained during a failure. The blue lines represent the reshaping values given during each step.

3.7 Appraisal guided Proximal Policy optimization

In recent years, Proximal Policy Optimization (PPO) has emerged as a prominent and widely used reinforcement learning algorithm for addressing various complex problems in the field of artificial intelligence. Its efficacy lies in its ability to strike a balance between exploration and exploitation, enabling efficient policy updates while ensuring stability during training. PPO belongs to the class of on-policy algorithms, which operate by collecting data through interaction with the environment in real time.

This characteristic makes PPO particularly suitable for scenarios with dynamic environments, such as the dynamic Gridworld problem being addressed in this work. Unlike off-policy algorithms that rely on historical data, PPO leverages the most up-to-date experiences, facilitating effective adaptation to changing conditions. Consequently, PPO has gained recognition as the state-of-the-art algorithm in reinforcement learning, offering superior performance, robustness, and sample efficiency.

In this work, standard PPO with a clipped surrogate objective has been modified, in order to accommodate the additional information provided by the cognitive appraisal variables. Fig 3.6 explains the modifications, mostly being the appraisal information given to

the critic network, which uses the state information and the appraisal information together to perform value prediction.

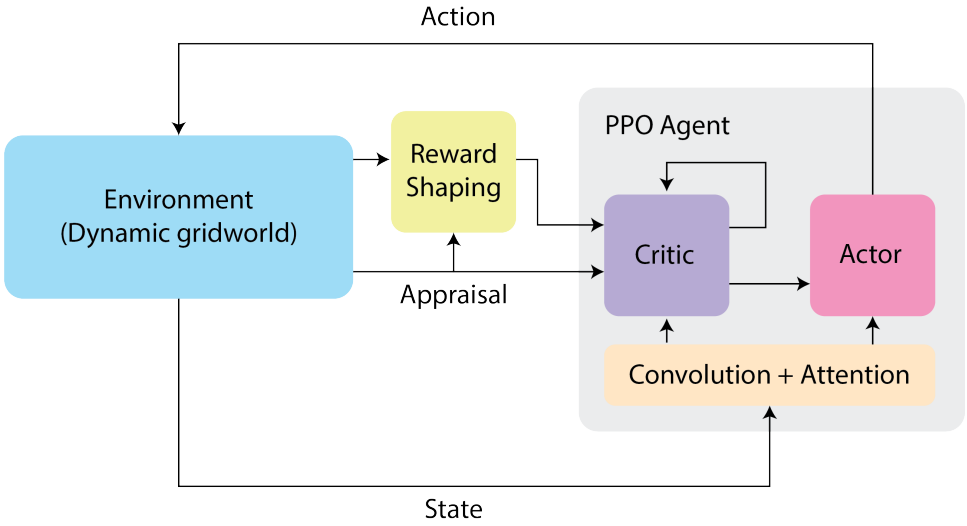


Figure 3.6: Appraisal guided PPO: Block diagram.

The appraisal information is given to the Critic model by concatenating with the state information. Each appraisal array has a dimension of (1,6) which represents the appraisals Motivational Relevance, Novelty, Certainty, Goal Congruence, Coping Potential, and Anticipation. During each step, appraisals are estimated from state information and are stored in memory, and used by the critic model for value estimation.

Before the state information is passed onto the critic model, the (N, N, 3) array is processed by a convolution network. The convolution network as shown in Fig 3.7 has 3 convolution layers and one self-attention module. The input is a 3-dimensional vector of size (N, N), where N is the view size of the agent. Each convolution layer uses a (2,2) filter with stride 1. All the layers in the convolution network are activated using the ReLU activation function. The output of the Convolution network is an array of dimensions (64, N, N)

The critic model architecture can be seen in Fig 3.8, which has 3 dense layers which take in the flattened information from the convolution block and concatenates it with the appraisals, and feeds it to the first dense layer. The first dense layer in critic model has an input dimension of 64xNxN where N is equal to the view size of the agent. then the following 2 dense layers have 256 and 64 units each, finally outputting a single value without using any activation function. Except for the last dense layer, all the other dense layers have ReLU activation.

Also, it can be seen from the block diagram that there is a reward-shaping step that takes in the reward during each step and reshapes it using the appraisal variables and then passes it to the critic network. This step is crucial in conditioning the agents to showcase the target behavior as discussed in the results chapter. The actor network is a simple 3-layer dense

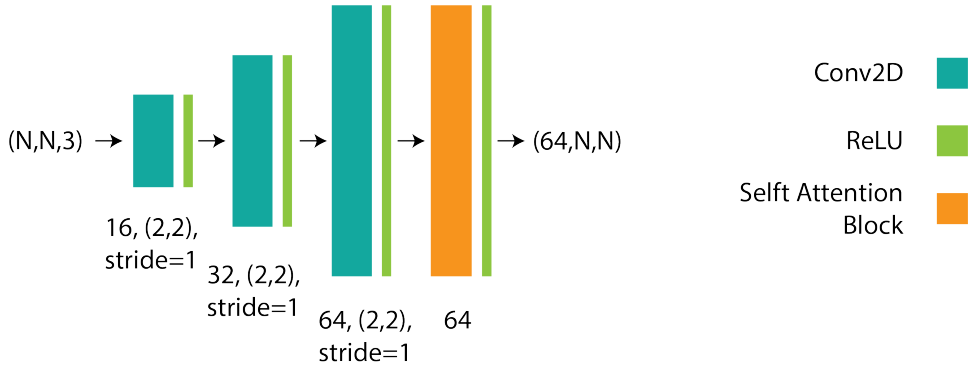


Figure 3.7: Convolution block of the PPO agent, which processes the state information before feeding into actor and critic networks.

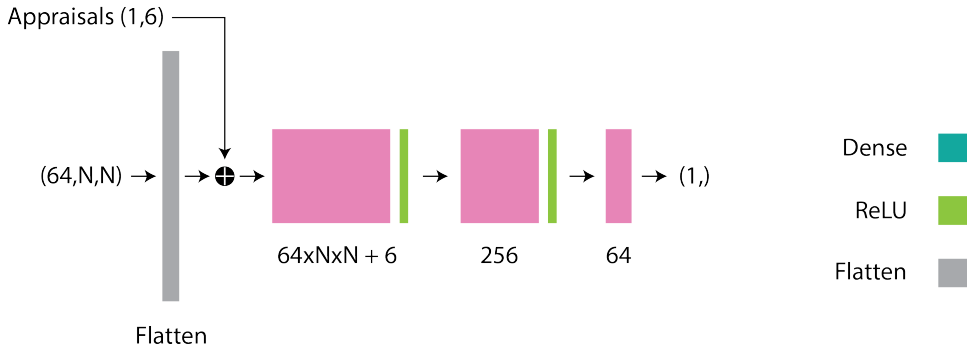


Figure 3.8: Critic network: 3 dense layers with ReLU activation used to estimate the value.

network that takes in the state information processed by the convolution block, flattens it, and then generates action probabilities, acting as a policy model for the agent. The actor network is detailed further in Fig 3.9

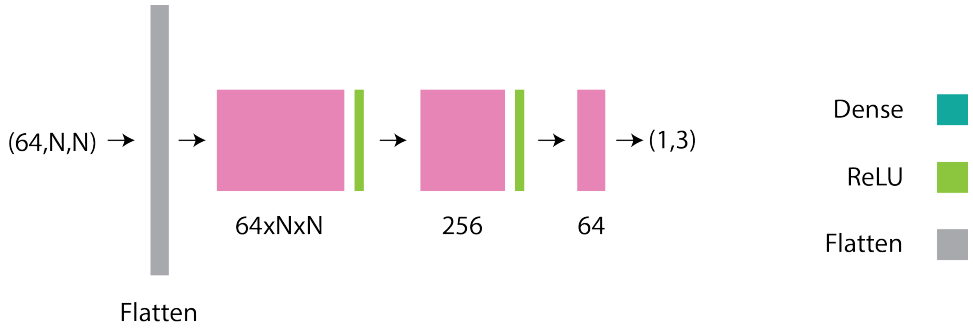


Figure 3.9: Actor network: 3 dense layers with ReLU activation used to generate action probabilities at each step.

In order to train appraisal-guided PPO agents, in the grid world environment, there are a number of parameters required to be initialized. These parameters define the training struc-

ture, the environment setup, and the type of psychological state the agent encompasses. Some of the parameters are the typical reinforcement learning variables such as the total steps which decide how long the training process should run, a seed value, the number of parallel environments which help in training the agent with parallel processing, learning rate, batch size, number of epochs, discount factor, etc.

Some are PPO-specific such as GAE lambda which controls the trade-off between bias and variance in the advantage estimation. The advantage normalization parameter helps by subtracting the mean advantage and dividing by the standard deviation, in order to reduce variance and improve the policy optimization. The clipping coefficient controls the extent to which the policy update is restricted during the optimization process. It is used to enforce a constraint on the policy update step to prevent large policy changes that could lead to instability or drastic performance degradation. The entropy coefficient and value function coefficient are both used to weigh the entropy loss and value loss in calculating the total loss, as seen in equation 3.10.

There are also some parameters specific to this work such as the reshaping coefficient which controls the influence of the reshaping factor in the training process as shown in equation 3.8 where w_{rs} is the coefficient. The NRE coefficient is used to weigh the NRE loss as shown in equation 3.10. The agent view size, grid size, and some toggles such as dynamic wall, dynamic goal, dynamic obstacles, moving goal, etc., all represent the parameters that define the environment as discussed in the previous section.

$$r'_t = r_t - w_{rs} * \rho(\alpha_t) \tag{3.8}$$

Algorithm 1 explains the working of modified cognitive appraisal-guided PPO. After initialization, a set of observations, and corresponding actions, values, appraisals, etc., are obtained by using the initial policy. then the advantages are estimated using Generalized Advantage Estimation (GAE). Once enough samples are obtained, they can be used to train the actor and critic models to update the policy and value networks respectively.

The policy loss in PPO is estimated by taking the clipped surrogate objective function. The policy loss is defined as the negative weighted average of the surrogate objective function, which measures the policy's deviation from its previous policy. The surrogate objective function compares the probabilities of the actions selected under the current policy to the probabilities of those actions under the previous policy. The Value loss in PPO is shown in the equation 3.9, where $L^v\theta$ represents the value loss objective. N is the batch size, indicating the number of samples in the mini-batch. s_i refers to the state at the index i in the mini-batch. θ denotes the current parameters of the value function neural network. θ_{old} represents the parameters of the value function neural network from the previous iteration. The objective of the value loss is to minimize the discrepancy between the estimated values and the target values, encouraging the value function to better approximate the expected returns.

$$L^V(\theta) = \frac{1}{N} \sum_{i=1}^{N} \left(V(s_i; \theta_{\text{old}}) - V_{\text{target}}(s_i) \right)^2 \tag{3.9}$$

The clipped surrogate objective has two terms: the first term is the unclipped objective, which encourages updates towards actions that have a positive advantage, promoting exploration. The second term is the clipped objective, which limits updates to a certain range, preventing large policy changes that could lead to instability. By taking the minimum between the two terms, the policy loss is calculated. This ensures that the update is conservative, as it selects the lower of the two values, allowing for controlled and gradual policy updates. The policy loss is then averaged over a batch of training samples and minimized using gradient-based optimization methods, such as stochastic gradient descent, to update the policy parameters (θ). The value loss is estimated by taking the mean squared error between the actual return and the predicted return. Further, the NRE loss is estimated and combined with the policy loss and value loss to obtain the total loss, which is then backpropagated using the Adam optimizer in order to train the agent. The optimizer uses a decayed learning rate with an epsilon of 1e-5.

Given the overall structure of the appraisal-guided PPO algorithm, it has all the components of a basic cognitive architecture. Starting from the perception part, the convolution block and attention blocks help in processing state information and obtaining important features from the environment. In a cognitive architecture, There is a perception component that can be seen in the convolution+self attention block in our system. Since the PPO uses replay memory, there is a memory component in the system that mimics the short-term memory in cognitive architecture.

The PPO algorithm together with the reward shaping and actor-critic architecture, contributes to the Learning and control components of a basic cognitive architecture. Further, the appraisals along with the actor-critic system can be used to explain the agent's reason for a particular action or response, which would act as the reasoning component of the Cognitive architecture. Since appraisals are used to guide the agent, they contribute to the development of a psychological state for the agent and would explain the various emotions the agent experiences during an episode in the environment. An overall cognitive architecture with all the components in our system can be called an appraisal-guided PPO-based partial cognitive architecture that can be simplified in the block diagram shown in Fig 3.10.

Algorithm 1 Appraisal Guided PPO

Initialize: Learning rate, total time steps, clipping coefficient, Policy network parameters θ_0, Value network parameters ϕ_0, training batch D, NRE network parameter σ_0, etc.

for training iterations 1 to M **do**
  Clear the training batch D.
  for each collect step t **do**
    Observe the environment state, s_t.
    Select action a_t according to policy $\pi_\theta(a_t|s_t)$.
    Execute action a_t, obtain reward r_t, next state s_{t+1}.
    Calculate appraisals α_t.
    Calculate the reward shaping factor ϵ_t.
  end for
  Compute advantage $\hat{A}$, using GAE.
  Add experiences $(s_t, a_t, r'_t, s_{t+1}, \alpha_t, \epsilon_t)$ to training batch B.
  for each training step **do**
    Recompute advantage estimate $\hat{A}$, using GAE.
    Split the training batch D to k mini-batches B according to batch size.
    for mini-batches $k = 1$ to k **do**
      Compute clipped surrogate objective for Policy Loss P_{loss}.
      Calculate Value Loss (V_{loss}).
      Calculate Entropy Loss (E_{loss}).
      Calculate NRE loss (NRE_{loss}).
      Calculate total loss as,

$$Total\ Loss = P_{loss} - E_{loss} + V_{loss} + NRE_{loss} \tag{3.10}$$

      Update the Actor, Critic, and NRE models by total loss using an optimizer.
    end for
  end for
end for

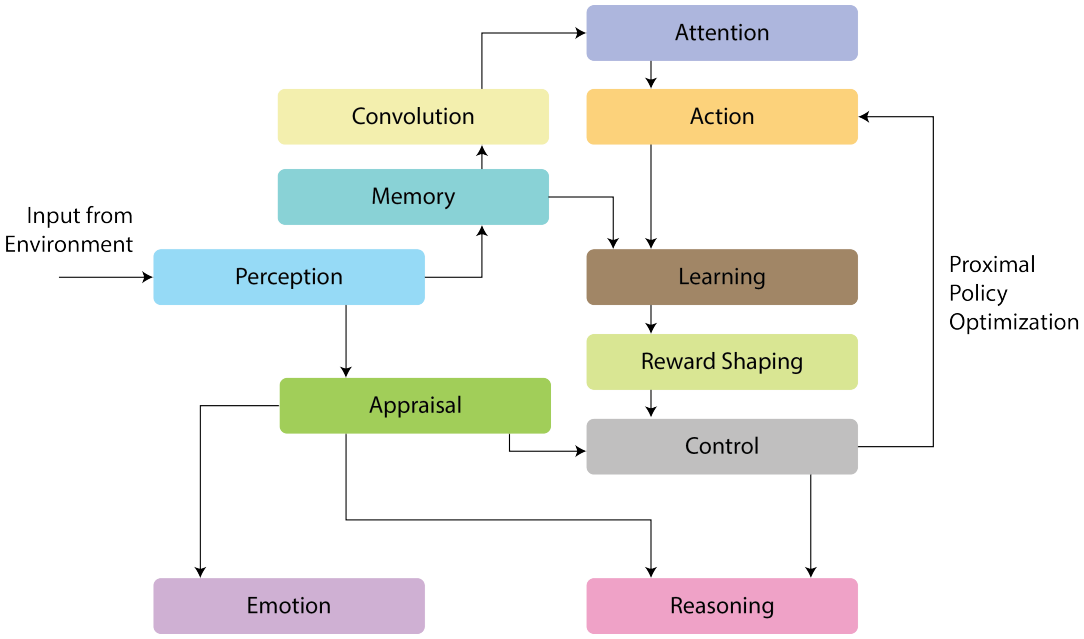


Figure 3.10: Apprisal guided PPO based Partial Cognitive Architecture

3.8 Evaluation Metrics

Since a PPO-based agent guided by cognitive appraisals is used, the training process and the testing process need to be evaluated using standard metrics. When it comes to the testing of fine-tuned agents, which are specifically trained to possess different psychological states, a new set of observation-based behavioral metrics is needed for evaluation.

In this work, in order to evaluate the training process, standard PPO metrics has been used such as episodic return, value loss, policy loss, entropy, KL divergence, explained variance, NRE loss, etc. Over the training iterations, these metrics are monitored and plotted to analyze the training efficiency.

For the purpose of testing the trained models, a set of 8 metrics has been developed that is based on the performance, behavior, and emotional state of the agent, as follows.

- **Wins/Plays:** The number of wins out of total plays is a way of estimating the win rate of the overall performance of the agent in the most simplified manner. A high win rate would mean a good agent that is capable of generalization.
- **Average Return:** Since the agent receives an episodic return over the number of episodes it plays, an average of the returns it obtained during wins is averaged and used as a metric to measure the average return of the agent.
- **Average Stress Level:** This metric defines the average stress level of an agent during an episode. Over a number of episodes, the stress levels are estimated and averaged to estimate the overall stress level of the agent in the environment. This is crucial in estimating the emotional stability of the agent.

Table 3.4: Evaluation metrics summarized

Metric	**Description**
Wins/Plays	Higher the better
Average Return	Should be closer to 1.0
Average Stress Level	Should be closer to 0.0
Number of Aversions	Should be minimum
Action Frequency	Higher for Forward and lower and similar for right and left actions
Stress graph	Should have low variance
Trajectory	Has to be short and relevant
Region of Interest	A more uniform distribution is preferred

- **Number of Aversions:** The number of aversions is another behavioral metric that calculates the number of times the agent has turned around and taken a different trajectory. This is calculated by counting the occurrences of turn-around sequences (L-L-F/R-R-F). This also is crucial in analyzing the psychological state of the agent.
- **Action Frequency:** The action frequency measures the frequency of each action the agent takes in an episode. for an agent which is emotionally stable and has good generalization, the frequency of forward action should be higher than left or right actions. The frequency of left and right actions should also be similar.
- **Trajectory:** The trajectory represents the path taken by an agent during an episode. Over the n episodes the agent goes through during the testing sequence, the trajectory information is stored and used for evaluating the behavior of the agent.
- **Region of Interest:** This metric identifies the agent's most visited states in the grid world. An image representing the frequency of visits at each cell in the grid is plotted to generate the ROI of the agent in an environment during testing.

Chapter 4

Results and Discussions

Using the appraisal-guided PPO agent, various training strategies have been experimented with and analyzed in this chapter. A total of 10 experiments have been carried out in the grid world environment using different reward shaping criteria and are compared with the baseline (standard PPO) and a control version that uses random noise instead of appraisals.

4.1 Training Environment

The training setup includes a 10x10 grid world environment with a static goal that is placed at a random position in the grid world at each episode, two dynamic obstacles which move around randomly and the agent itself has a random starting position. Around the outermost cells of the grid world, walls are placed in order to avoid the agent from exiting the grid world, as shown in Fig 4.1.

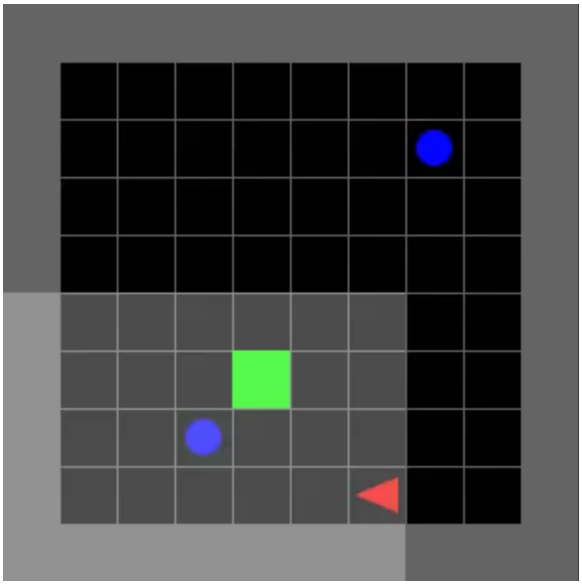


Figure 4.1: Training Environment: Two Dynamic Obstacles and a static goal

The training environment is kept as simple as possible but still includes the dynamics of

Table 4.1: Training environment parameters

Parameter	**Value**
Grid Size	10
Agent view Size	7
Agent start position	0 (random)
Agent start direction	0 (random)
Number of obstacles	2
Max. Steps	100
Dynamic goal-starting position	True
Moving Goal	False
Moving Obstacles	True

moving obstacles and random placement of goals. This is done so that the generalization capability of the agent using different variations of the PPO algorithm can be analyzed. The parameters used to define the training environment are shown in table 4.3. Here the grid size of 10x10 is chosen to give enough size and area to learn the dynamics of the environment. A too small grid will restrict the motion of the obstacles and the agent and too large will take more time to train and reach convergence. Similarly, the agent view size is chosen as 7x7 in order to accommodate enough information from the observation space in its view. The training episode has a maximum steps limit of 100, which makes the agent learn to complete the task within 100 steps. Also, the maximum step limit of 100 is acceptable for a grid size of 10x10.

The PPO-specific training parameters are shown in table 4.2. The learning rate is set to 0.001 which is used by the Adam optimizer. A seed value of 1337 is used straight all the experiments so that the results are consistent and comparable. The training process across all the variations is performed for 100,000 total steps. The replay memory size is set to 128 and the training of the actor-critic model is carried out for 4 epochs, taking a mini-batch size of 4. The rest of the values corresponding to PPO are listed in table 4.2.

Table 4.2: PPO training parameters

Parameter	**Value**
Learning Rate	0.001
Seed	1337
Total time-steps	100000
Number of Environments	4
Number of steps	128
Learning rate annealing	True
Discount factor	0.99
GAE Lambda	0.95
Number of mini-batches	4
number of epochs	4
Clip coefficient	0.2
Entropy coefficient	0.01
Value function coefficient	0.5
Maximum gradient norm	0.5

4.2 Experimental configurations

In this work, 10 different configurations have been experimented with, which include a baseline and control along with multiple variants using reward shaping. The configurations are detailed as

- **Baseline:** This is the version with no appraisal incorporated into the critic's input. This also doesn't have any form of reward-shaping strategies included. This uses standard PPO with the clipped objective algorithm. All the other experiments are compared with the baseline. Through the experiments, along with identifying configurations for psychological disorders, an effort to obtain a configuration that performs better than the baseline performance is also made.
- **PPO with noise:** This configuration uses random noise concatenated to the critic's input and does not use any reward-shaping strategies. This version is used as a control to ensure that the results obtained using appraisals are not random and are just an effect of noise in the input of the critic network.
- **PPO with appraisal:** This configuration includes appraisal information concatenated to the critic's input. But compared to other configurations, this does not use any reward-shaping strategies.
- **RSv1:** This configuration which stands for reward shaping version 1, uses motivational relevance in the reward shaping strategy. The reshaped reward can be obtained using the equation 4.1.

$$reward_{rsv1} = reward_t - 0.01[1 - \zeta^t_{MR}] \tag{4.1}$$

- **RSv2:** This configuration uses coping potential in the reward-shaping strategy. The reshaped reward can be obtained using the equation 4.2.

$$reward_{rsv2} = reward_t - 0.01[1 - \zeta^t_{CP}] \tag{4.2}$$

- **RSv3:** This configuration uses goal congruence in the reward-shaping strategy. The reshaped reward can be obtained using the equation 4.3.

$$reward_{rsv3} = reward_t - 0.01[1 - \zeta^t_{GC}] \tag{4.3}$$

- **RSv4:** This configuration uses a combination of motivational relevance and goal congruence in the reward-shaping strategy. The reshaped reward can be obtained using the equation 4.4.

$$reward_{rsv4} = reward_t - 0.01\big([1 - \zeta^t_{MR}] + [1 - \zeta^t_{GC}]\big) \tag{4.4}$$

- **RSv5:** This configuration uses a combination of motivational relevance, coping potential and goal congruence in the reward-shaping strategy. The reshaped reward can be obtained using the equation 4.5.

$$reward_{rsv5} = reward_t - 0.01\big([1 - \zeta^t_{MR}] + [1 - \zeta^t_{CP}] + [1 - \zeta^t_{GC}]\big) \tag{4.5}$$

- **RSv6:** This configuration uses a combination of motivational relevance, coping potential, and goal congruence in the reward-shaping strategy, in such a way that the agent is forced to minimize these appraisals. The reshaped reward can be obtained using the equation 4.6.

$$reward_{rsv6} = reward_t - 0.1\big([\zeta^t_{MR}] + [\zeta^t_{CP}] + [\zeta^t_{GC}]\big) \tag{4.6}$$

- **RSv7 (A, B):** This configuration uses a combination of motivational relevance, coping potential, and goal congruence in the reward-shaping strategy, in such a way that the agent is forced to minimize the motivational relevance while increasing coping potential. The reshaped reward can be obtained using the equation 4.7. Here there are 2 variations of RSv7 which are version A, having the factor $\epsilon = 0.01$, and version B having $\epsilon = 0.1$

$$reward_{rsv7} = reward_t - \epsilon[1 - \zeta^t_{CP}] - 0.1\big([\zeta^t_{MR}] + [\zeta^t_{GC}]\big) \tag{4.7}$$

4.3 Training Results

In this section, a comprehensive analysis of the training results obtained through the utilization of the 10 different configurations is presented, along with an examination of the corresponding graphs depicting various metrics that were tracked throughout the training process. These metrics play a crucial role in assessing the performance and convergence of the Proximal Policy Optimization (PPO) algorithm.

Episodic Return is an essential metric that measures the cumulative reward achieved by the agent in each episode during training. It serves as a fundamental indicator of the effectiveness of the learned policy. A higher episodic return signifies that the agent successfully

maximizes its reward in the environment, demonstrating improved policy performance.

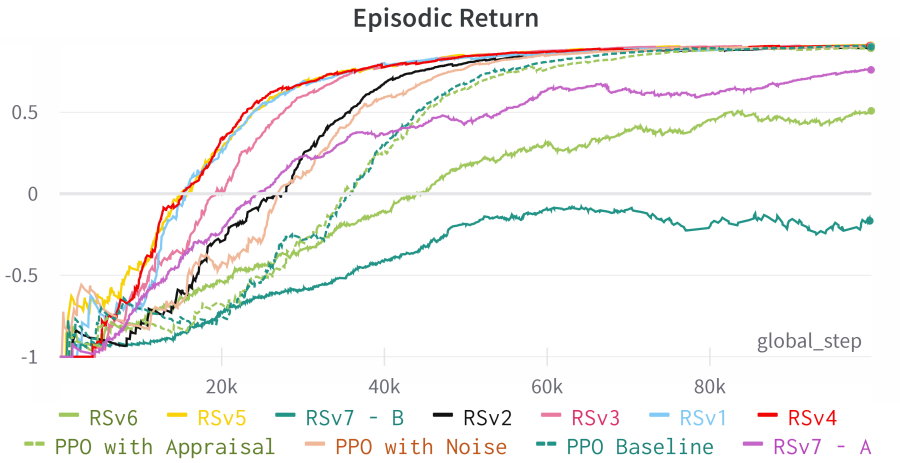


Figure 4.2: Episodic Return training graph.

Value Loss is a critical component in PPO that evaluates the discrepancy between the predicted and actual values of the state-action pairs. This metric reflects how well the value function is being learned and updated during the training process. A lower value loss indicates that the value function is effectively approximating the true values, leading to more accurate estimates of the state-action values. The episodic return during training is plotted In the graph in Fig 4.3

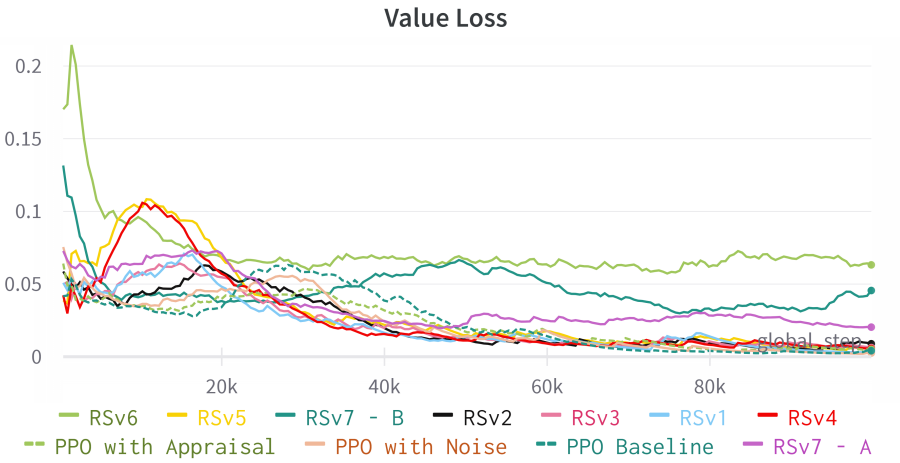


Figure 4.3: Value loss training graph.

Policy Loss assesses the divergence between the updated policy and the previous policy. It quantifies the extent to which the policy parameters have changed between iterations. A

lower policy loss indicates that the policy updates are closer to the previous policy, resulting in more stable and incremental changes to the agent's behavior. The graph for Policy loss is given in Fig 4.4

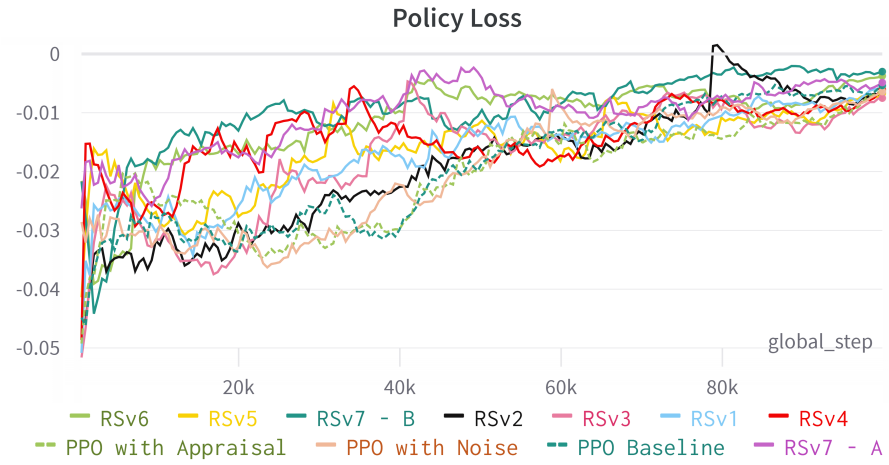


Figure 4.4: Policy loss training graph.

Entropy measures the level of uncertainty or randomness in the policy distribution. It represents the diversity of actions chosen by the agent and encourages exploration in the early stages of training. Monitoring entropy helps ensure a balance between exploration and exploitation, as a suitable level of randomness allows the agent to discover optimal policies. The graph for Entropy over the training iterations is given in Fig 4.5

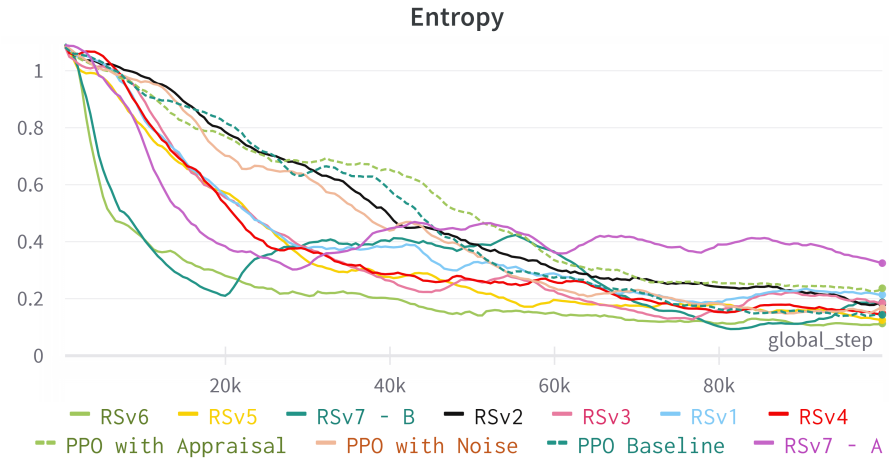


Figure 4.5: Entropy training graph.

KL divergence evaluates the discrepancy between the updated policy and the previous

policy, specifically in terms of their probability distributions. This metric provides insights into the extent of policy updates and serves as a regularization term to prevent large policy changes. A lower KL divergence indicates more conservative policy updates, promoting stability during training. The graph of KL divergence during the training is given in Fig 4.6

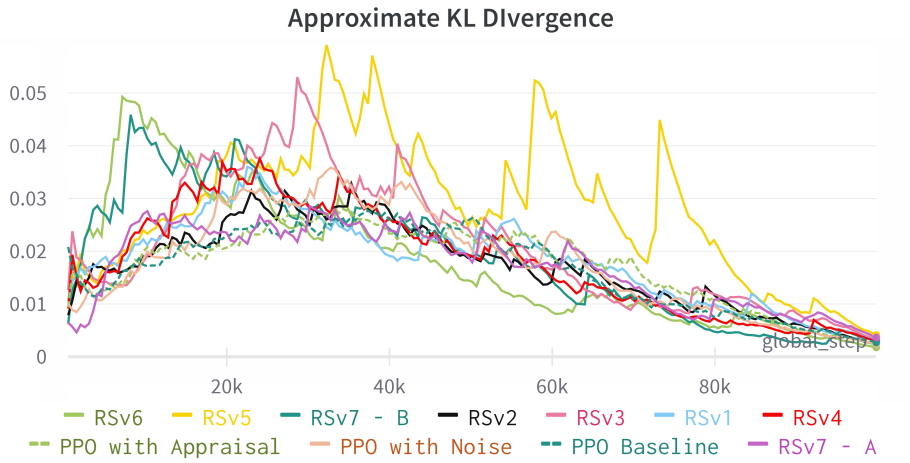


Figure 4.6: KL Divergence training graph.

Explained variance quantifies how well the value function predicts the actual rewards obtained by the agent. It indicates the proportion of the total variance in the returns that can be accounted for by the value function. Higher values of explained variance suggest that the value function is accurately capturing the reward structure of the environment. The graph of explained variance is shown in Fig 4.7

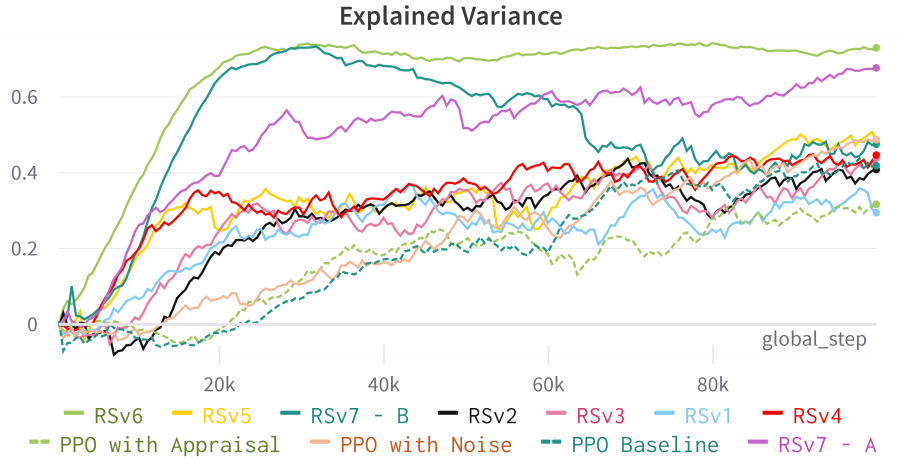


Figure 4.7: Explained variance training graph.

Clip fraction determines the mean fraction of surrogate loss that was clipped (above the clip range threshold) for PPO during the policy update step. Clipping restricts the magnitude of policy changes to a pre-defined range to ensure stability. It indicates how different the new policy is compared to the old policy. The Clip Fraction graph is shown in Fig 4.8

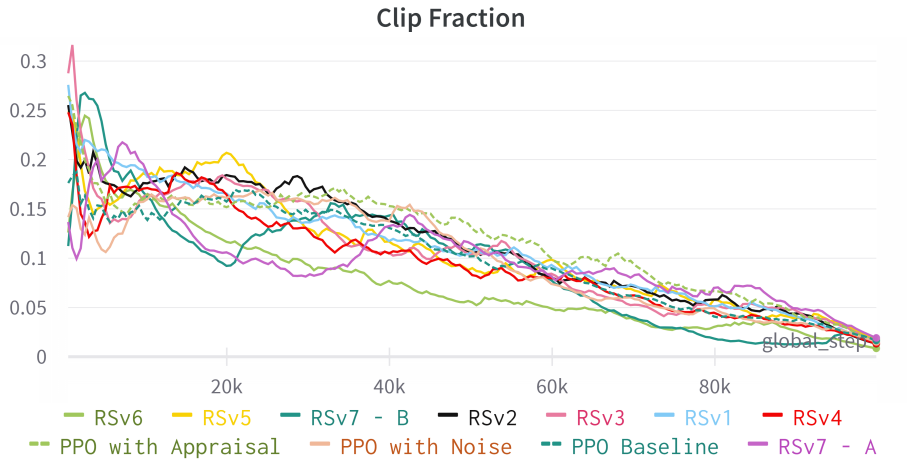


Figure 4.8: Clip fraction training graph.

Episodic Length refers to the number of time steps taken by the agent to complete an episode. Monitoring this metric provides insights into the agent's learning progress and the convergence rate. A decreasing episodic length indicates that the agent is learning to accomplish tasks more efficiently and is converging toward optimal policies. The Episodic length graph is shown in Fig 4.9

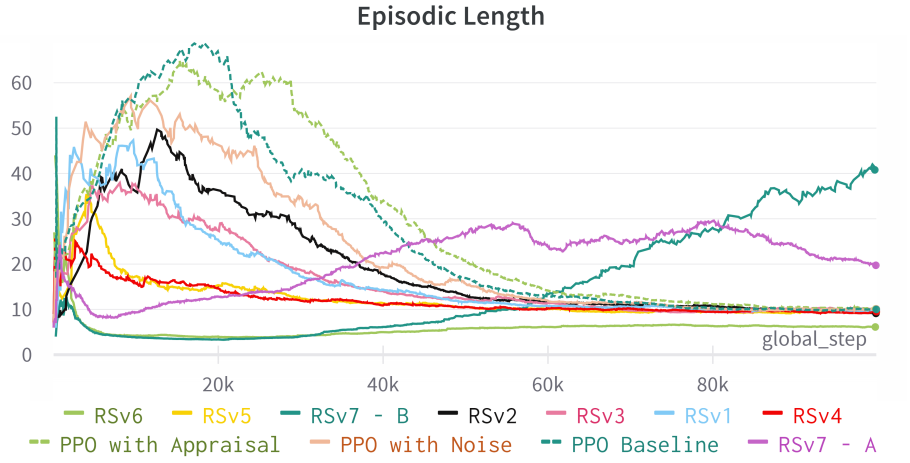


Figure 4.9: Episodic Length training graph.

Additionally, Throughout the experiments, the stress level has also been estimated and the graph corresponding to the stress level over training iterations for each of the configurations is shown in Fig 4.10.

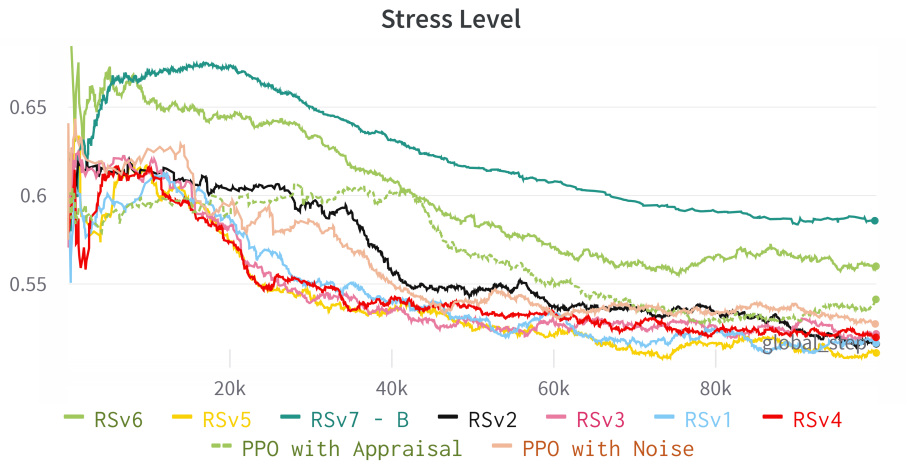


Figure 4.10: Stress Level training graph.

Another important parameter is the NRE loss and the graph corresponding to the NRE loss over training iterations is shown in Fig 4.11.

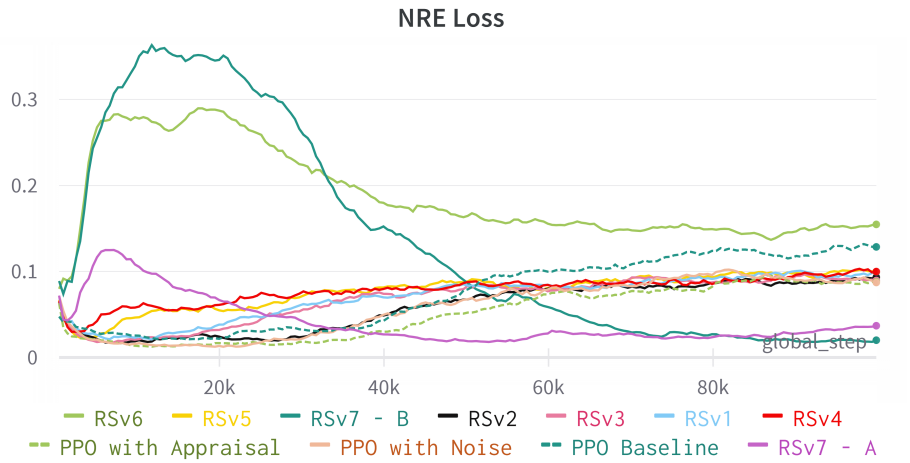


Figure 4.11: NRE loss training graph.

Analyzing the graphs associated with these metrics enables a comprehensive evaluation of the performance of the PPO algorithm under different configurations as shown in the table 4.4. By studying the trends, patterns, and correlations displayed in these graphs, valuable insights can be gained into the convergence behavior, stability, and effectiveness of the algorithm, aiding in the optimization and refinement of the training process.

4.4 Test Results

The trained models were tested on 2 environments which include one with the same attributes as the training environment but with an increased number of obstacles and a randomly moving goal (10x10, 7 obstacles, max_steps=100, and dynamic goal). Since this environment is slightly complex compared to the training environment, it is possible to evaluate the generalization capability of the agent. Also, another environment (10x10, 7 obstacles, dynamic goal, max_steps=400, and dynamic walls) is used, which is much more complex and different from the training environment which allows further analysis of the generalization abilities of the agent.

The test results of baseline configuration which uses a standard PPO algorithm showcase a really good performance both in terms of learning the dynamics of the environment and the adaptability of the agent in the new test environment. In both test environments, the agent showcases significant qualities of the standard PPO algorithm. On average the agent plays 33 episodes and wins in 26 of them, making the success rate, 78.78%. The average return is 0.7538 and the measured average stress level is about 0.4838. The agent makes a total of 145 aversions throughout the 33 episodes. The action Frequency gives the right action to be 183 and the right actions to be 144 which does not show a drastic difference, which would indicate some problem in behavior. Overall the baseline agent receives a total score of 0.6909.

Fig 4.12, (a) and (b) shows the region most visited by the agent. The darker a cell is in the grid, the more the agent has visited the cell. It can be seen that in the case of Grid World A, the baseline agent mostly stays or moves around the center of the Grid World. The results are not much different in the case of the Grid World B environment, the agent seems to be moving around the center as well. Fig 4.12 shows the path taken by the agent and the goal in both GW-A and GW-B environments. The trajectory corresponds to the longest successful episode and from the figures it is evident that the agent, in both cases is trying to move towards the region of the goal. A more comprehensive way of evaluating the performance of the model is by calculating an overall score, that combines the Wins/Plays ratio and the average episodic return. In this work, the overall score is calculated as shown in equation 4.8

$$Score = \frac{(n_{wins} * average(R_t) + (n_{losses} * (-1))}{n_{plays}} \tag{4.8}$$

Based on the above equation, the table summarizes the score values of all the configurations and it can be seen that the best configuration is RSv1 with high scores for both GW-A and GW-B environments. Also, this shows that the addition of appraisals has significantly impacted the performance of the PPO algorithm, and depending on the configuration the performance can be controlled as desired. In addition to the overall scores, the stress levels, aversions, distractions, etc., are also indicators of the behavior of agents under different configurations.

Based on the above equation, the table summarizes the score values of all the configurations, demonstrating that the best configuration is RSv1. RSv1 shows high scores for both GW-A and GW-B environments, indicating its superior performance. The addition of appraisals has had a significant impact on the performance of the PPO algorithm, allowing for

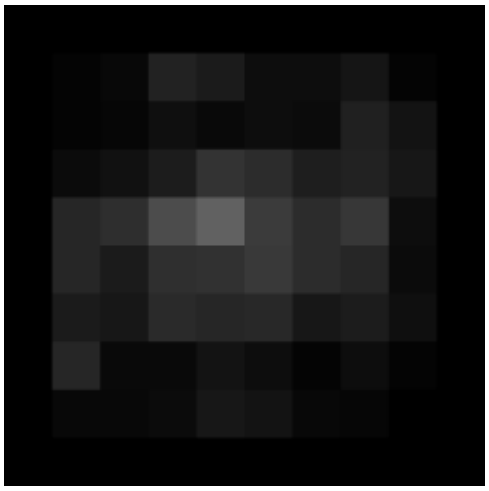
Table 4.3: Test score comparison

Configuration	**GW-A**	**GW-B**
Baseline	0.6909	0.4746
PPO + Noise	0.7081	0.6173
PPO + Appraisal	0.7534	0.7534
RSv1	0.8003	0.7066
RSv2	0.7347	0.6890
RSv3	0.6569	0.5241
RSv4	0.7936	0.3421
RSv5	0.7945	0.1577
RSv6	0.1067	0.1022
RSv7-A	0.4226	0.4446
RSv7-B	0.3312	0.1676

control of the performance based on the desired configuration. Alongside the overall scores, indicators such as stress levels, aversions, and distractions offer insights into the behavior of agents under different configurations.

The configuration RSv1, which incorporates Motivational relevance for reward shaping as discussed in section 4.2 and equation 4.1, has yielded favorable results during training and testing in both GW-A and GW-B environments. Notably, it demonstrates better generalization abilities compared to the baseline and PPO + appraisal configurations. Examining the trajectories of the agent and the goal in the longest successful episode for RSv1 in both test environments, as seen in Fig 4.13, it becomes evident that the agent adeptly follows and reaches the goal. This observation aligns with the high overall score and performance achieved by RSv1.

These findings emphasize the positive influence of utilizing appraisals, specifically the inclusion of Motivational relevance, on the performance and generalization capabilities of the PPO algorithm. RSv1 showcases the effectiveness of incorporating this appraisal in guiding the agent toward successful goal attainment. Such insights and evaluations help in refining and optimizing the configuration of the RL agent, contributing to its enhanced performance in a grid-world environment.

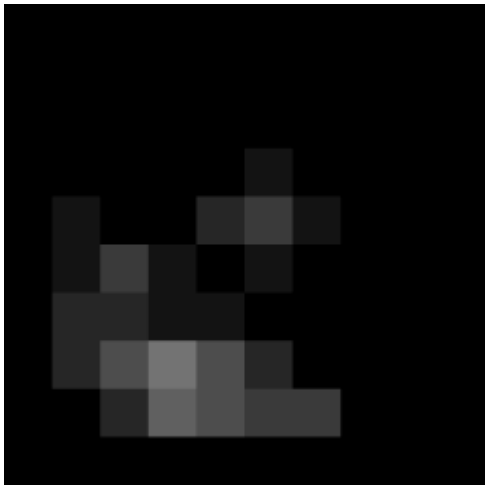


(a) ROI Static Walls

(b) ROI Dynamic Walls (Maze)

(c) Agent (GW-A)

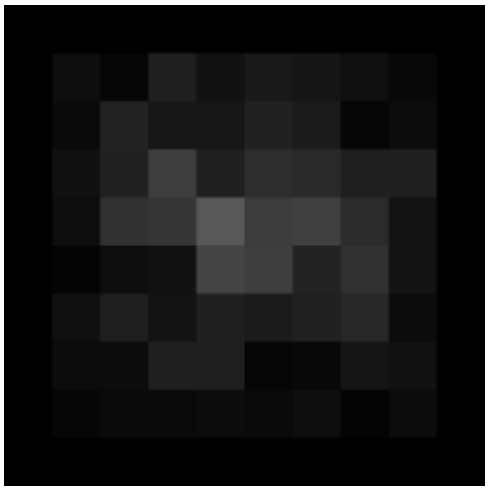
(d) Goal (GW-A)



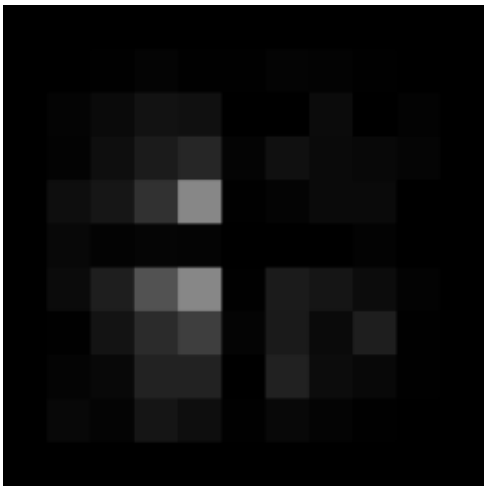
(e) Agent (GW-B) (Maze)

(f) Goal (GW-B) (Maze)

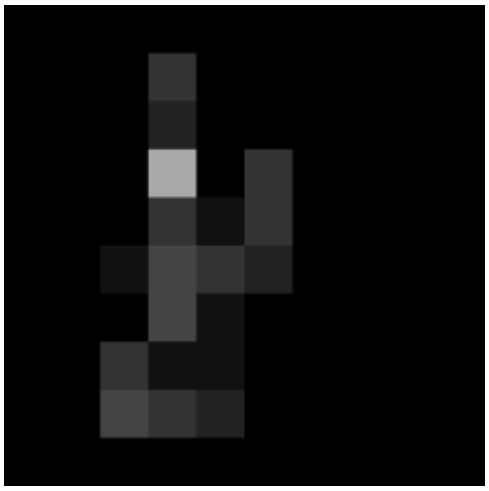
Figure 4.12: (a) Baseline agent's region of Interest in GW-A. (b) Baseline agent's region of Interest in GW-B. (c) Baseline agent's trajectory on GW-A in a single episode. (d) Goal trajectory on GW-A in a single episode. (e) Baseline agent's trajectory on GW-B in a single episode. (f) Goal trajectory on GW-B in a single episode.



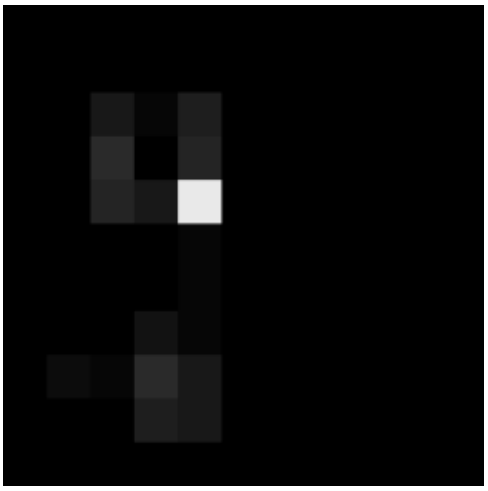
(a) RSv1 - ROI Static Walls



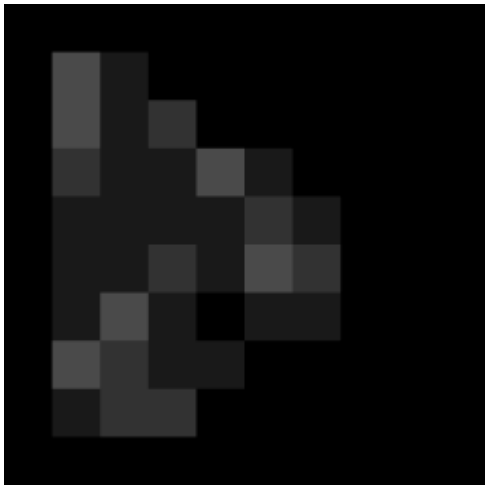
(b) RSv1 - ROI Dynamic Walls (Maze)



(c) RSv1 agent (GW-A)



(d) RSv1 goal (GW-A)



(e) RSv1 agent (GW-B) (Maze)

(f) RSv1 goal (GW-B) (Maze)

Figure 4.13: (a) RSv1 agent's region of Interest in GW-A. (b) RSv1 agent's region of Interest in GW-B. (c) RSv1 agent's trajectory on GW-A in a single episode. (d) Goal trajectory on GW-A in a single episode. (e) RSv1 agent's trajectory on GW-B in a single episode. (f) Goal trajectory on GW-B in a single episode.

4.5 RSv7-A/B: OCD and Anxiety Disorder

If the RL agent exhibits behaviors characteristic of OCD-like disorder, it may display a range of symptoms akin to Obsessive-Compulsive Disorder (OCD). One such symptom is an obsessive focus on specific actions or patterns, leading the agent to feel compelled to excessively repeat them. This behavior is evident in the results for GW-A presented in Table 4.5, where a high imbalance in action probability is observed for RSv7 variants. The agent demonstrates a strong preference for forward and left actions, while right actions are significantly less favored. Specifically, in the case of GW-A, the agent chooses left actions 27.79% of the time, while right actions are selected only 18.63% of the time. Similarly, in GW-B, left actions are chosen 35.60% of the time, while right actions are chosen a mere 17.85% of the time.

Another behavior associated with OCD is compulsive checking or rechecking. In this regard, the RL agent exhibits a compulsion to repeatedly monitor its surroundings and verify certain conditions. Live monitoring of the agent reveals its incessant need to keep track of the goal and obstacles, constantly updating its knowledge of its positions. This behavior aligns with the characteristic symptom of OCD and signifies a compulsive need for vigilance and reassurance.

Furthermore, avoidance of perceived threats is a prominent behavior seen in OCD and anxiety disorders, and the RL agent also showcases specific obsessions related to certain obstacles or locations. The agent actively avoids these areas out of fear or anxiety. This avoidance behavior is particularly noticeable in RSv7, as the agent consistently avoids traversing the middle of the grid and instead prefers to remain around the edges. This pattern is further evident in the region of interest map, which highlights the agent's preference for the edges. Consequently, the agent opts for longer paths or detours to minimize the chances of encountering perceived threats associated with specific obstacles or areas.

The RSv7-A agent exhibits repetitive behavior patterns, indicative of OCD-like symptoms. These patterns, coupled with other metrics such as high levels of distractions and stress, further reinforce the notion that the agent experiences symptoms akin to those associated with anxiety and OCD-like disorders. The agent engages in repetitive behaviors, adhering to specific action sequences or movement patterns, despite their lack of direct contribution to goal attainment. These behaviors manifest as compulsions driven by the agent's internal compulsive tendencies.

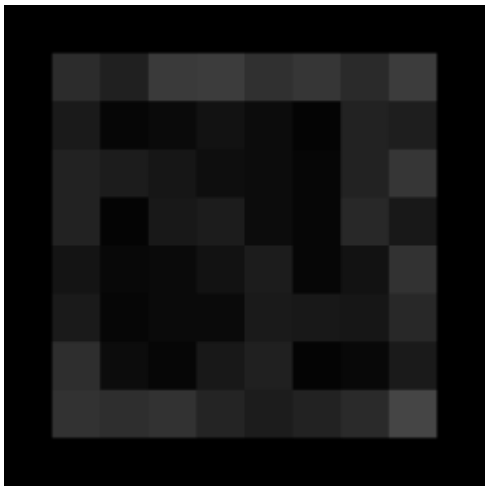
The RSV7-B agent's tendency to navigate along the edges of the grid reflects its inclination to avoid potential harm. This behavior can be most seen in Anxiety disorder. Even when shorter paths are available, the agent adheres to the edges to minimize the likelihood of encountering obstacles within the grid's interior. Figure 4.15 illustrates this behavior, emphasizing the agent's high interest in the corners of the grid. In fact, during simulations, the agent often becomes trapped in the corners, which serve as the most remote positions in the grid, offering minimal chances of encountering obstacles as they are shielded by walls on two sides.

Figure 4.16 illustrates the behavior of the RSv7-A agent in a variant of the GW-A envi-

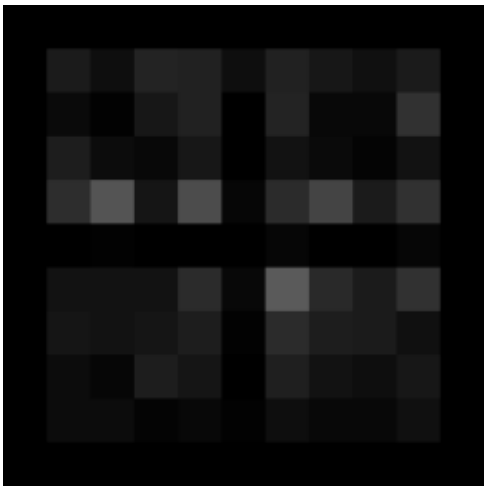
ronment where the goal remains stationary, and there are only two obstacles. The agent's initial position is relatively close to the goal; however, instead of immediately searching the nearby area, it exhibits a tendency to move away from the central region of the grid and gravitates towards the edges. Once it reaches an edge, the agent follows a trajectory in an anticlockwise direction (most frequently observed) and explores the grid-primarily from the edges. This behavioral pattern is consistent across multiple episodes, aligning with OCD-like symptoms commonly associated with repetitive and ritualistic behavior. The agent predominantly takes forward and left actions until it visually perceives the goal, at which point it moves forward and successfully reaches the goal. The recurrent occurrence of this behavior further reinforces its connection to OCD-like symptoms.

On the other hand, Figure 4.17 demonstrates the behavior of the RSv7-B agent in a variant of the GW-A environment with a stationary goal and only two obstacles. Similar to RSv7-A, this agent also displays a failure to initially search for the goal by observing its surroundings. Instead, it exhibits a tendency to move away from the center of the grid. Notably, when confronted with obstacles, the agent employs a distinctive avoidance strategy that is not observed in other variants. It rotates in place, striving to keep the obstacles out of its field of view by remaining in the same position. This behavior strongly indicates similarities to Anxiety disorder, as it reflects an excessive focus on potential threats and an avoidance behavior associated with anxiety. Furthermore, even when no obstacles are present in the immediate vicinity, the agent persists in staying at the corners of the grid, suggesting a reluctance to leave a perceived safe zone, a trait frequently observed in anxiety-related behaviors.

The distinctive behavioral patterns exhibited by the RSv7-A and RSv7-B agents in response to the environment variations provide valuable insights into the psychological dimensions at play. These observations highlight the connections between artificial agent behavior and psychological disorders, namely OCD and Anxiety disorder. The association between the RSv7-A agent's preference for edge exploration and repetitive actions aligns with OCD-like symptoms. Similarly, the RSv7-B agent's avoidance behavior, particularly when staying in corners, along with its unique avoidance strategy, reflects anxiety-related tendencies. These findings contribute to the growing body of knowledge regarding the modeling of psychological disorders in artificial agents and have implications for understanding human cognition within the context of AI. Table 4.7, summarizes the test results by mentioning the key differences and behavioral patterns of all the 11 configurations.



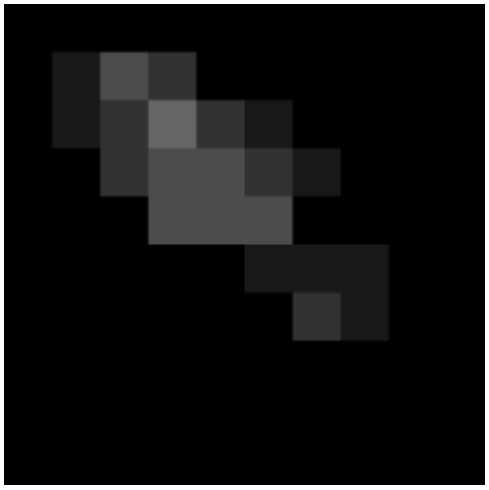
(a) RSv7-A - ROI Static Walls



(b) RSv7-A - ROI Dynamic Walls (Maze)

(c) RSv7-A agent (GW-A)

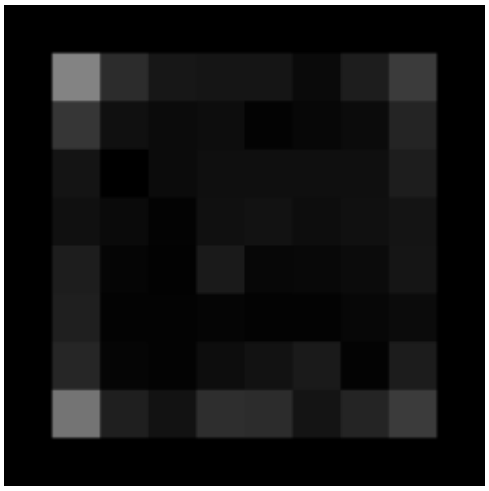
(d) RSv7-A goal (GW-A)



(e) RSv7-A agent (GW-B) (Maze)

(f) RSv7-A goal (GW-B) (Maze)

Figure 4.14: (a) RSv7-A agent's region of Interest in GW-A. (b) RSv7-A agent's region of Interest in GW-B. (c) RSv7-A agent's trajectory on GW-A in a single episode. (d) Goal trajectory on GW-A in a single episode. (e) RSv7-A agent's trajectory on GW-B in a single episode. (f) Goal trajectory on GW-B in a single episode.

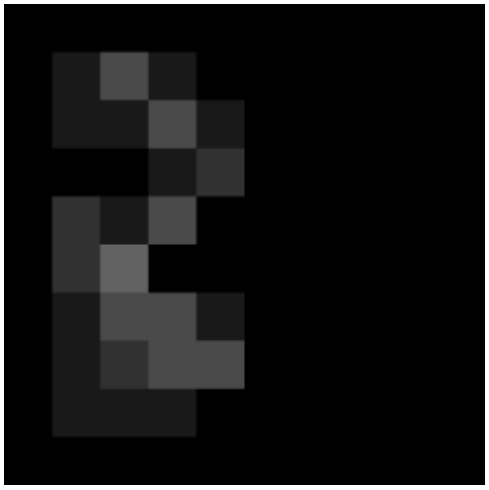


(a) RSv7-B - ROI Static Walls

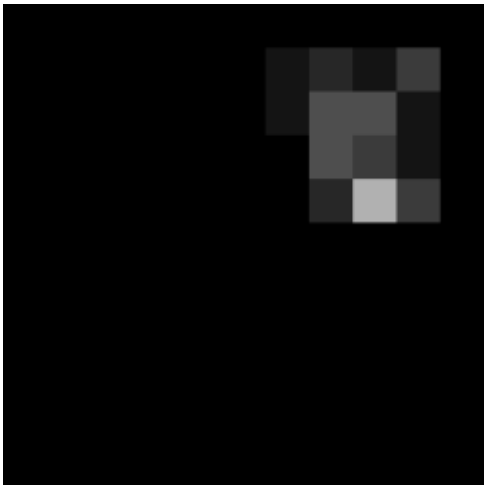
(b) RSv7-B - ROI Dynamic Walls (Maze)

(c) RSv7-B agent (GW-A)

(d) RSv7-B goal (GW-A)



(e) RSv7-B agent (GW-B) (Maze)



(f) RSv7-B goal (GW-B) (Maze)

Figure 4.15: (a) RSv7-B agent's region of Interest in GW-A. (b) RSv7-B agent's region of Interest in GW-B. (c) RSv7-B agent's trajectory on GW-A in a single episode. (d) Goal trajectory on GW-A in a single episode. (e) RSv7-B agent's trajectory on GW-B in a single episode. (f) Goal trajectory on GW-B in a single episode.

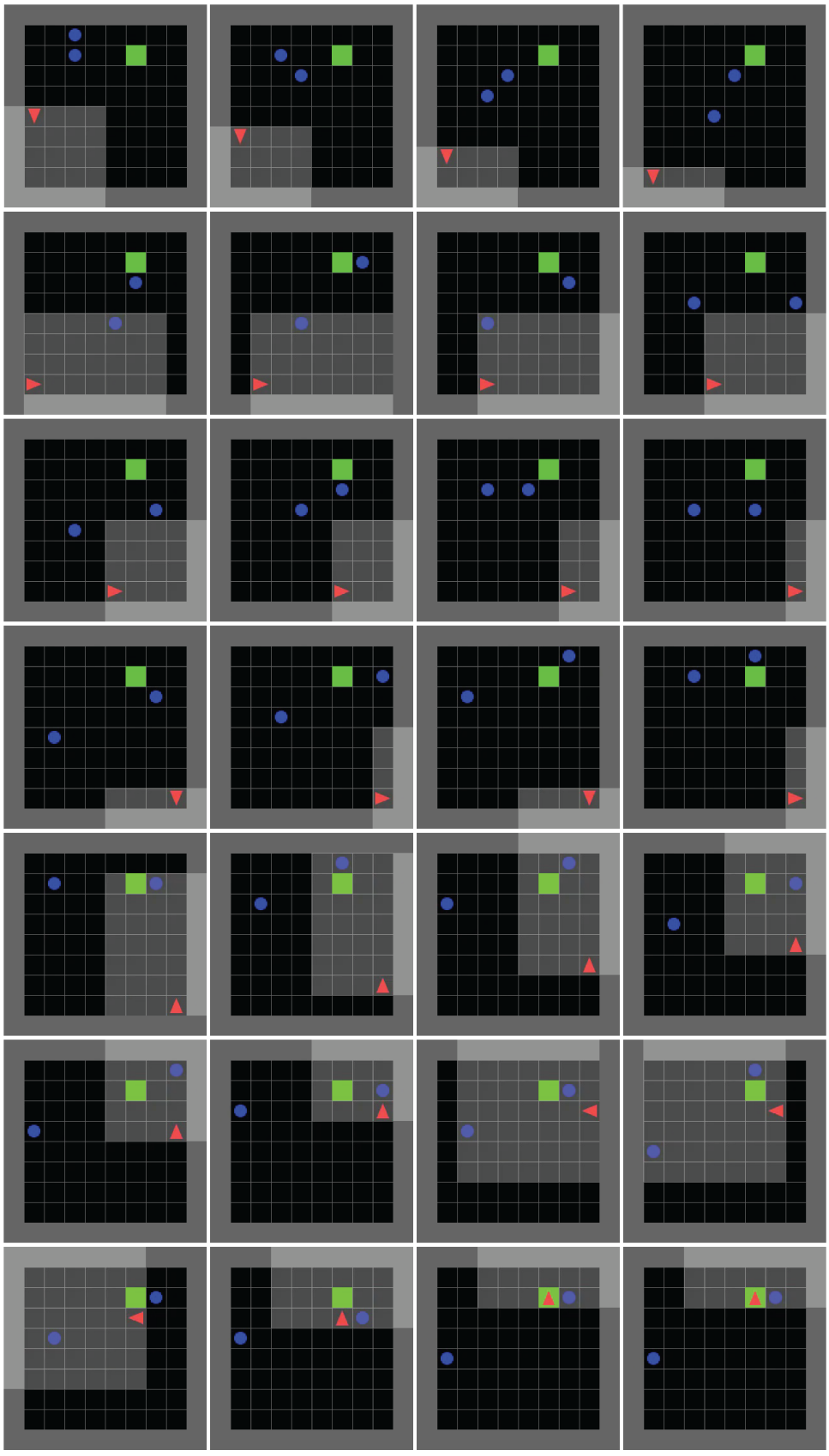


Figure 4.16: Behaviour of agent under RSv7-A, over 28 steps of an episode.

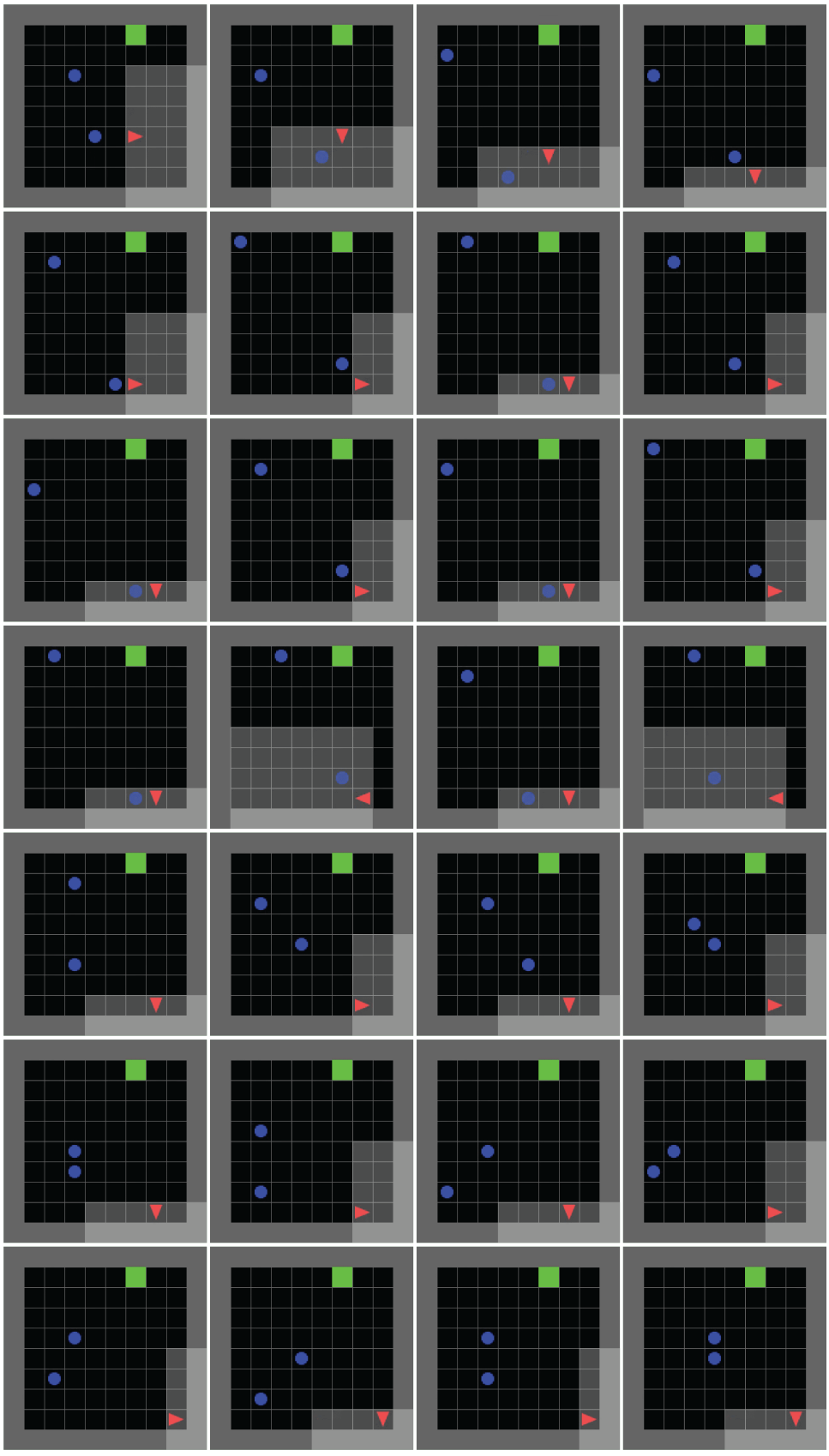


Figure 4.17: Behaviour of agent under RSv7-B, over 28 steps of an episode.

Table 4.4: Training Summary

Metric	Baseline	Noise	Appraisal	RSv1	RSv2	RSv3	RSv4	RSv5	RSv6	RSv7-A	RSv7-B
Episodic Return	0.9008	0.9045	0.8956	0.9120	0.8985	0.8973	0.9105	0.9152	0.4340	0.7486	-0.1511
Policy Loss	-0.0053	-0.0064	-0.0063	-0.0054	-0.0064	-0.0075	-0.0060	-0.0066	-0.0037	-0.0048	-0.0030
Value Loss	0.0041	0.0025	0.0050	0.0027	0.0089	0.0063	0.0056	0.0057	0.0632	0.0203	0.0455
Entropy	0.1436	0.1662	0.2369	0.2141	0.1810	0.1852	0.1488	0.1248	0.1132	0.3250	0.1878
KL Divergence	0.0027	0.0024	0.0023	0.0030	0.0023	0.0032	0.0029	0.0041	0.0017	0.0036	0.0030
Explained Variance	0.4188	0.4872	0.3173	0.2947	0.4086	0.4389	0.4468	0.4853	0.7298	0.6769	0.4754
Clip Fraction	0.0165	0.0151	0.0160	0.0148	0.0132	0.0162	0.0134	0.0180	0.0085	0.0191	0.0149
NRE Loss	0.1294	0.0864	0.0884	0.1006	0.0941	0.0913	0.1005	0.0995	0.1558	0.0381	0.0201
Episodic Length	9.5610	9.5060	10.5720	9.6060	9.3130	10.1470	9.1860	9.1560	6.8700	20.483	41.2730

Table 4.5: Test Summary (Grid world - A)

Metric	Baseline	Noise	Appraisal	RSv1	RSv2	RSv3	RSv4	RSv5	RSv6	RSv7-A	RSv7-B
Plays/Wins	0.7878	0.8260	0.8421	0.8823	0.8275	0.7500	0.9230	0.8919	0.1130	0.4761	0.3684
Average Return	0.7538	0.7143	0.7893	0.8140	0.7757	0.7518	0.7195	0.7818	0.8878	0.7749	0.7981
Average Stress	0.4838	0.4679	0.4562	0.4650	0.4687	0.4651	0.4834	0.4878	0.5241	0.5337	0.5472
Number of Aversions	145	125	144	156	167	151	172	136	132	94	79
Forward Action	0.6350	0.5524	0.5892	0.5725	0.6060	0.5915	0.6127	0.6149	0.7142	0.5357	0.4799
Left Action	0.2042	0.2901	0.2377	0.2589	0.2678	0.2533	0.2142	0.2700	0.1551	0.2779	0.4575
Right Action	0.1607	0.1573	0.1729	0.1685	0.1261	0.1551	0.1729	0.1149	0.1305	0.1863	0.0625
Distraction	220	240	196	204	245	203	228	212	154	364	397

Table 4.6: Test Summary (Grid world - B)

Metric	Baseline	Noise	Appraisal	RSv1	RSv2	RSv3	RSv4	RSv5	RSv6	RSv7-A	RSv7-B
Plays/Wins	0.5000	0.6363	0.875	0.7333	0.7368	0.5416	0.3513	0.1607	0.1032	0.4615	0.1724
Average Return	0.8987	0.9402	0.7181	0.9270	0.8703	0.9352	0.9475	0.9629	0.9815	0.9266	0.9441
Average Stress	0.5583	0.5195	0.6074	0.5560	0.5606	0.5528	0.5278	0.5868	0.5558	0.5458	0.5787
Number of Aversions	124	80	51	68	111	104	122	121	99	93	48
Forward Action	0.4464	0.4589	0.4419	0.2857	0.4609	0.4475	0.5725	0.5970	0.7075	0.4654	0.5535
Left Action	0.3236	0.4084	0.2879	0.3325	0.3727	0.4375	0.2150	0.2723	0.2075	0.3560	0.3783
Right Action	0.2299	0.1406	0.2700	0.2700	0.1662	0.1149	0.2120	0.1305	0.0848	0.1785	0.0068
Distraction	414	379	555	196	441	462	260	242	104	424	224

Table 4.7: Test results summarized

Configuration	**Summary**
Baseline	Standard PPO implementation shows good generalization results with acceptable average return but has higher stress levels compared to many of the other variants.
PPO + Noise	Used as a control to evaluate if noise simulates the effect of appraisal in input. Has performance similar to baseline.
PPO + Appraisal	Comparatively better than baseline and noise variants, in terms of average return, stress level, and action frequency has lower distractions.
RSv1	Showcases the best performance compared to all other configurations. Better results than baseline and PPO + appraisal without reward shaping. Has a faster convergence rate and overall performance in both GW-A and GW-B are good.
RSv2	Configured to increase coping potential, is capable of high return and success rate but not as high as RSv1. Showcases higher values of aversions and distractions and higher stress levels.
RSv3	Showcases a lower success rate and average return than RSv1. Performs well in GW-A environment but not in GW-B. Has lower generalization capability and has action imbalance. ROI shows a non-uniform distribution, focused around the central region.
RSv4	Has a very high success rate in GW-A but not in GW-B. Showcases high levels of aversions but comparatively less action imbalance. ROI shows a high focus on the central region.
RSv5	Showcases a high success rate in GW-A but very low in GW-B, proving lower generalization capability. Behavioral analysis shows the agent has a good focus on the goal. Designed to maximize Motivational relevance, Goal congruence, and coping potential.
RSv6	Has the least success rate compared to all other variants, even though having a good focus on goal and tracking ability. Comparatively has low action imbalance in GW-A compared to all other configurations
RSv7-A	Showcases higher stress levels, action imbalance, and behavior shows, agent moving along in a repetitive and fixed region in the environment, hence showcasing OCD-like symptoms.
RSv7-B	Has a high-stress level, high imbalance in right and left action probabilities, and behavioral analysis shows a high level of anxiety-like symptoms and is unable to learn dynamics of the environment.

Chapter 5

Conclusion

In this study, a modified variant of the PPO algorithm in a grid-based environment was evaluated with the objective of incorporating cognition similar to that of natural intelligence. The research aimed to devise a PPO-based partial cognitive architecture capable of analyzing the behavioral patterns of agents in dynamic grid environments. This involved the development of design criteria to emulate specific behavioral patterns observed in psychological disorders like OCD and anxiety disorder. Empirical evidence from experiments showcased the ability of agents trained using the modified PPO algorithm to simulate symptoms of Anxiety and OCD when solving grid-world problems. Customized criteria and metrics were devised to assess these behaviors in RL agents.

The study revealed that the modified PPO algorithm has the potential to incorporate cognitive dimensions, enabling the simulation and study of psychological states and behaviors resembling those observed in natural intelligence. These findings underscore the significance of further research in affective computing, particularly in the context of psychology and psychological disorders within the field of AI. The study also demonstrated the capabilities of reinforcement learning, particularly on-policy algorithms like PPO, in simulating the cognitive states of artificial agents. Replicating cognitive patterns associated with psychological disorders presents promising avenues for understanding and emulating complex cognitive processes using RL techniques.

It is important to note that the study's results are empirical in nature, based on observations and measurements from the conducted experiments. While the behavior of the artificial agents exhibited similarities to certain psychological disorders, caution should be exercised in interpreting these findings. The complexity and intricacies of psychological disorders in humans surpass the current capabilities of artificial agents. The study provides valuable insights into modeling and simulating psychological behavior in artificial agents, but further research is necessary to explore the underlying dynamics in greater depth.

Future research should focus on investigating the complex dynamics underlying the observed behaviors and exploring the interaction between cognitive appraisals, reward-shaping strategies, and agent behavior. Incorporating sophisticated models and theories from psychology and neuroscience can enhance the fidelity of the simulation and provide a more comprehensive understanding of the relationship between artificial agent behavior and hu-

man psychology. Continued research in this area will contribute to a deeper understanding of psychological behavior and advance the development of AI systems that better simulate and understand human cognition.

The future scope of research involves the evaluation of psychological disorders in RL agents and the development of Cognitive Appraisal guided Partial Cognitive Architecture. This includes the enhancement of appraisal models to improve the reliability and accuracy of psychological evaluations in RL agents. Further dimensions of appraisals, such as social evaluation and moral judgment, can be incorporated to enable agents to exhibit more nuanced emotional responses and decision-making capabilities.

Expanding the research to multi-agent systems can explore the dynamics of emotional interactions among RL agents. Investigating how agents appraise and respond to the emotional states of others can lead to the development of cooperative or competitive strategies involving emotional intelligence. Additionally, studying the transferability and generalization capabilities of emotional intelligence in RL agents can enhance their adaptability. Real-world applications, such as mental health support or personalized learning, can benefit from emotional agents, offering potential societal impact.

Research on emotional agents is crucial for advancing the field of AI and improving human-AI interactions. Addressing emotional stability ensures reliability and predictability, fostering trust and user acceptance. Furthermore, exploring psychological disorders in agents can provide insights into both AI and human psychology, advancing our understanding of human behavior and informing therapeutic practices. These findings contribute to the intersection of AI, psychology, and cognitive science, paving the way for developing AI systems with nuanced cognitive behaviors.

References

[1] Rafael Calvo et al. *Cyberpsychology and affective computing*. 2014.

[2] Roy F Baumeister et al. "How emotion shapes behavior: Feedback, anticipation, and reflection, rather than direct causation". In: *Personality and social psychology review* 11.2 (2007), pp. 167–203.

[3] Nico H Frijda, Peter Kuipers, and Elisabeth Ter Schure. "Relations among emotion, appraisal, and emotional action readiness." In: *Journal of personality and social psychology* 57.2 (1989), p. 212.

[4] James A Russell. "Evidence of convergent validity on the dimensions of affect." In: *Journal of personality and social psychology* 36.10 (1978), p. 1152.

[5] Richard S Lazarus. "Cognition and motivation in emotion." In: *American psychologist* 46.4 (1991), p. 352.

[6] Andrew Ortony, Gerald L Clore, and Allan Collins. *The cognitive structure of emotions*. Cambridge university press, 2022.

[7] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[8] Jonathan Gratch, Stacy Marsella, and Paolo Petta. "Modeling the cognitive antecedents and consequences of emotion". In: *Cognitive Systems Research* 10.1 (2009), pp. 1–5.

[9] Anthony Dickinson. "The 28th Bartlett memorial lecture causal learning: An associative analysis". In: *The Quarterly Journal of Experimental Psychology Section B* 54.1b (2001), pp. 3–25.

[10] Burrhus Frederic Skinner. *The behavior of organisms: An experimental analysis*. BF Skinner Foundation, 2019.

[11] Richard Bellman. "Dynamic programming". In: *Science* 153.3731 (1966), pp. 34–37.

[12] Christopher John Cornish Hellaby Watkins. "Learning from delayed rewards". In: (1989).

[13] Ronald A Howard. "Dynamic programming and markov processes." In: (1960).

[14] John Schulman et al. "Proximal policy optimization algorithms". In: *arXiv preprint arXiv:1707.06347* (2017).

[15] John Schulman et al. "Trust region policy optimization". In: *International conference on machine learning*. PMLR. 2015, pp. 1889–1897.

[16] Richard S Lazarus and Susan Folkman. *Stress, appraisal, and coping*. Springer publishing company, 1984.

[17] James J Gross. "The emerging field of emotion regulation: An integrative review". In: *Review of general psychology* 2.3 (1998), pp. 271–299.

[18] Elizabeth A Kensinger. "Remembering emotional experiences: The contribution of valence and arousal". In: *Reviews in the Neurosciences* 15.4 (2004), pp. 241–252.

[19] Klaus R Scherer, Angela Schorr, and Tom Johnstone. *Appraisal processes in emotion: Theory, methods, research*. Oxford University Press, 2001.

[20] Thomas M Moerland, Joost Broekens, and Catholijn M Jonker. "Emotion in reinforcement learning agents and robots: a survey". In: *Machine Learning* 107 (2018), pp. 443–480.

[21] Maxime Chevalier-Boisvert et al. "Minigrid & Miniworld: Modular & Customizable Reinforcement Learning Environments for Goal-Oriented Tasks". In: *CoRR* abs/2306.13831 (2023).

[22] Ashish Vaswani et al. "Attention is all you need". In: *Advances in neural information processing systems* 30 (2017).

[23] Alex Wang et al. "GLUE: A multi-task benchmark and analysis platform for natural language understanding". In: *arXiv preprint arXiv:1804.07461* (2018).

[24] Jacob Devlin et al. "Bert: Pre-training of deep bidirectional transformers for language understanding". In: *arXiv preprint arXiv:1810.04805* (2018).

[25] Alexey Dosovitskiy et al. "An image is worth 16x16 words: Transformers for image recognition at scale". In: *arXiv preprint arXiv:2010.11929* (2020).

[26] Dan Horgan et al. "Distributed prioritized experience replay". In: *arXiv preprint arXiv:1803.00933* (2018).

[27] Matteo Hessel et al. "Rainbow: Combining improvements in deep reinforcement learning". In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 32. 1. 2018.

[28] Natasha Jaques et al. "Way off-policy batch deep reinforcement learning of implicit human preferences in dialog". In: *arXiv preprint arXiv:1907.00456* (2019).