

CGAWM - Curriculum-Guided Adversarial World Model for Data-Efficient Adversarial Reinforcement Learning

By

Migbar Abera Shibru

Submitted to the School of Information Technology and Engineering
In partial fulfillment of the requirements for the degree of
Master of Science in Artificial Intelligence

Supervised by: Dr. Beakal Gizachew
Co-Advisor: Dr. Natnael Argaw
College of Technology and Built Environment
Addis Ababa University
Addis Ababa, Ethiopia

February, 2026

©Migbar Abera Shibru, 2026

APPROVAL

This is to certify that this thesis titled **"CGAWM - Curriculum-Guided Adversarial World Model for Data-Efficient Adversarial Reinforcement Learning "** is prepared by **Migbar Abera Shibru** and submitted in partial fulfillment of the thesis-option requirements for the Degree of **Master of Science in Artificial Intelligence** at the School of Information Technology & Engineering, College of Technology and Built Environment.

Name	Signature	Date
Dr. Beakal Gizachew		May 21/2026
(Advisor)		
Million Meshesha		May 21/2026
(External Examiner)		
Dr. Fantahun Bogale		May 21/2026
(Internal Examiner)		

Contents

List of Symbols **ix**

Abstract xi

Acknowledgements xii

1 Introduction **1**

1.1 Motivation 3

1.1.1 Computational Barriers and Resource Intensity 4

1.1.2 Non-Stationarity and Adversarial Dynamics 4

1.2 Statement of the Problem 5

1.3 Research Questions 6

1.4 Objective 6

1.4.1 General Objective 6

1.4.2 Specific Objective 6

1.5 Significance 7

1.6 Scope 8

1.7 Contributions 9

1.8 Thesis Structure 10

2 Literature Review **11**

2.1 Background 11

2.1.1 Foundations of Reinforcement Learning 11

2.1.2 Deep Reinforcement Learning (DRL) 12

2.2 Data Efficiency and Sample Complexity 13

2.2.1 Formalizing the Efficiency Problem 14

2.2.2 The Cost of Real-World Interaction 14
2.3 Adversarial Games in the Context of DRL 15
2.3.1 Theoretical Formulation: Stochastic Games 15
2.3.2 The Challenge of Non-Stationarity 16
2.3.3 Game Theoretic Classifications and Equilibrium 16
2.3.4 Adversarial Exploration 17
2.4 The Challenges of Data Efficiency in Competitive Systems 18
2.4.1 The Sparse Reward Problem 18
2.4.2 The Generalization Problem 19
2.4.3 Computational Cost 20
2.4.4 The Exploration-Exploitation Trade-off 21
2.4.5 Knowledge Consolidation and Representation 21
2.4.6 High Dimensionality 21
2.4.7 Adaptability and Non-Stationarity 22
2.5 State of the Art Approaches 22
2.5.1 Model-Based Reinforcement Learning (MBRL) 24
2.5.2 Curriculum Learning (CL) 27
2.5.3 Meta-Reinforcement Learning 31
2.5.4 Representation Learning 35
2.5.5 Uncertainty Estimation and Exploration 37
2.6 Research Gap 40
2.7 Related Work 41

3 Methodology **45**
3.1 Research Methodology 45
3.2 Data Acquisition 48
3.2.1 Simulation Environment and Configuration 48
3.2.2 Experience Replay and Data Structure 49
3.3 Design and Development 49
3.3.1 Adversarial Dynamics Modeling(ADeM) 50
3.3.2 Uncertainty Guided Exploration Strategy 55
3.3.3 Adversarial Curriculum Learning (ACL) 59

4 Experimentation **66**
4.1 Experimental Setup . . . 66
4.1.1 Simulation Environment: PettingZoo MPE . . . 67
4.1.2 Implementation of CGAWM and Infrastructure . . . 68
4.1.3 Baseline Configurations and Ablation Design . . . 68
4.1.4 Hyperparameter Configuration . . . 69
4.1.5 Training Protocol and Evaluation Metrics . . . 70
4.2 Ablation Studies . . . 72
4.2.1 Experimental Configuration for Ablations . . . 72
4.2.2 Quantitative Result Overview . . . 74
4.2.3 Component wise Impact Analysis . . . 74
4.2.4 Learning Dynamics and Convergence Speed . . . 75
4.2.5 Statistical Significance of Findings . . . 76
4.2.6 Summary of Ablation Study . . . 77
4.3 Results and Reliability . . . 77
4.3.1 Sample Efficiency and Real-World Interaction Reduction . . . 78
4.3.2 Statistical Significance and Convergence Analysis . . . 79
4.3.3 Learning Dynamics and Trajectory Analysis . . . 80
4.3.4 Comparative Performance Analysis . . . 82
4.3.5 Reliability and Robustness . . . 85
4.3.6 State Space Convergence and Exploration . . . 85
4.3.7 Summary of Results . . . 85
4.4 Evaluation . . . 86
4.4.1 Emergent Coordination: The Split Coverage Strategy . . . 86
4.4.2 Adversarial Counter Adaptation . . . 87
4.4.3 Impact of Uncertainty on Trajectory Entropy . . . 87
4.4.4 Failure Mode Analysis . . . 89
4.5 Discussion . . . 90
4.5.1 Synthesis of Key Findings . . . 90
4.5.2 Implications for Multi-agent Systems . . . 91
4.6 Revisiting the Research Questions . . . 91
4.6.1 Limitations of the Study . . . 92

5 Conclusion and Recommendation **94**
5.1 Conclusion 94
5.2 Recommendations 95
5.3 Future Work 95

List of Figures

2.1 Comparative Analysis of Sample Efficiency. 15
2.2 The cycle of non-stationarity in Multi-Agent Reinforcement Learning (MARL). 23
2.3 The Phenomenon of Compounding Errors in Model-Based Planning. 26
2.4 The Adversarial Curriculum Learning (ACL) Loop. 29
2.5 Context-Based Meta-Reinforcement Learning Architecture. 34

3.1 Research Design based on the DSRP Model 46
3.2 System Flow of the Curriculum-Guided Adversarial World Model(CGAWM) 50
3.3 Architectural Overview of the Curriculum Guided Adversarial World Model (CGAWM) 51
3.4 ADeM Architecture 53
3.5 Uncertainty Architecture 64
3.6 Architectural of Dynamic Adversarial Curriculum 65

4.1 Comparative ablation study demonstrating the impact of individual components. 76
4.2 Training performance and stability of CGAWM vs. baselines across three independent random seeds. 79
4.3 Evaluation of sample efficiency across 250,000 steps 80
4.4 Final learning curves of the CGAWM agent over 5,000 training episodes. . . 81
4.5 Multi-paradigm benchmarking results against SOTA 83
4.6 Model-Free approaches against CGAWM 84
4.7 Comparative analysis of reward acquisition and epistemic uncertainty vs CGAWM. 88

List of Tables

2.1 Computational Resource Requirements of State-of-the-Art Adversarial DRL Systems 20
2.2 Comparative Analysis of State-of-the-Art Model-Based Architectures 26
2.3 Taxonomy of Curriculum Learning Approaches in Adversarial DRL 31
2.4 Comparative Analysis of Meta-RL Approaches in Adversarial Domains . . . 34
2.5 Impact of Representation Learning on MARL Data Efficiency 37
2.6 Comparison of Uncertainty-Driven Exploration Strategies 40
2.7 Comparative Analysis of Research Gaps in Related Methodologies 41
2.8 Comparison of Related Works with the Proposed CGAWM 44

4.1 Hardware and Software Specifications for Experimental Training 69
4.2 Agent (PPO) Hyperparameters 70
4.3 Adversarial Dynamics Model (ADeM) Hyperparameters 71
4.4 Curriculum and Uncertainty-Guided Exploration Hyperparameters 71
4.5 Summary of Ablation Findings (1,000 Episodes) 74
4.6 Final Statistical Results (5,000 Episodes, 3 Seeds) 80
4.7 Comparison of Best Evaluation Reward Across Baseline and SOTA Methods 83

List of Acronyms

MLP	Multilayer Perceptron
GNN	Graph Neural Network
ACG	Adversarial Curriculum Generation
ADeM	Adversarial Dynamics Model
AI	Artificial Intelligence
CGAWM	Curriculum-Guided Adversarial World Model
CNN	Convolutional Neural Network
DRL	Deep Reinforcement Learning
EAT-C	Environment-Adversarial Sub-Task Curriculum
GPU	Graphics Processing Unit
MAMBRL	Multi-Agent Model-Based Reinforcement Learning
MARL	Multi-Agent Reinforcement Learning
MBPO	Model-Based Policy Optimization
MDP	Markov Decision Process
MLP	Multilayer Perceptron
MPE	Multi-Particle Environments (PettingZoo)
PAIRED	Protagonist Antagonist Induced Regret Environment Design
POMDP	Partially Observable Markov Decision Process
PPO	Proximal Policy Optimization
ReLU	Rectified Linear Unit
RL	Reinforcement Learning
SAC	Soft Actor-Critic
SGD	Stochastic Gradient Descent

TD3 . Twin Delayed Deep Deterministic Policy Gradient

List of Symbols

Greek Symbols

α	Learning rate
β	Uncertainty scaling coefficient (intrinsic reward weight)
γ	Discount factor for future rewards (typically 0.99)
δ	Temporal Difference (TD) error
ϵ	Exploration rate (probability of random action)
θ	Trainable parameters (weights) of the policy network
μ	Mean of a probability distribution
π	Policy function mapping states to actions
π_θ	Policy parameterized by neural network weights θ
σ	Standard deviation (measure of uncertainty)
σ^2	Variance
τ	Curriculum advancement threshold (required win rate)
ϕ	Trainable parameters of the Adversarial Dynamics Model

Latin Symbols

A	The set of all valid actions
a_t	Action taken by the agent at time step t
$\mathcal{D}$	Replay buffer (dataset of transitions)
E	Expectation operator
H	Planning horizon (length of synthetic rollout)
$J(\theta)$	Objective function to be maximized
$\mathcal{L}$	Loss function (e.g., Mean Squared Error)

M	Ensemble size (number of networks in World Model)
N	Number of agents in the environment
$p(s'\|s,a)$	Transition probability distribution
$Q(s,a)$	Action-value function
r_t	Total reward received at time step t
r_e	Extrinsic reward (from the environment)
r_i	Intrinsic reward (derived from uncertainty)
S	The set of all possible states
s_t	State observation at time step t
$\hat{s}_{t+1}$	Predicted next state generated by World Model
T	Total duration of an episode
$V(s)$	State-value function

Abstract

Deep Reinforcement Learning (DRL) has achieved remarkable success in single-agent domains; however, Multi-Agent Reinforcement Learning (MARL) remains plagued by severe sample inefficiency and training instability, particularly in competitive zero-sum games. The non-stationarity inherent in adversarial environments where the opponent's strategy evolves continuously often prevents agents from converging to optimal policies, leading to cycling behaviors or catastrophic forgetting. To address these challenges, this thesis proposes the Curriculum-Guided Adversarial World Model (CGAWM), a framework that integrates three methodological pillars: an Adversarial Dynamics Model (ADeM) that amplifies data efficiency via synthetic rollouts, an Adversarial Curriculum that stabilizes training by progressively increasing opponent competence, and an Uncertainty-Guided Exploration mechanism that incentivizes the agent to investigate high-entropy states. Evaluated on the PettingZoo `simple_adversary` benchmark, experimental results demonstrate that CGAWM achieves a 136% improvement in asymptotic reward compared to state-of-the-art Model-Free PPO baselines. Furthermore, the proposed method exhibited superior reliability, achieving a consistent convergence rate across random seeds where baseline methods failed to solve the task. Qualitative analysis confirmed the emergence of sophisticated game-theoretic behaviors, including *Split-Coverage* and *Defensive Hovering*, indicating that the agent successfully developed a primitive Theory of Mind. These findings suggest that integrating world models with structured curricula provides a scalable path toward robust autonomous systems in adversarial domains.

Keywords: Multi-Agent Reinforcement Learning, World Models, Curriculum Learning, Adversarial Training, Sample Efficiency.

Acknowledgements

I would like to express my deepest gratitude to my supervisor, Dr. Beakal Gizachew, for their invaluable guidance, patience, and continuous support throughout the course of this research. Their deep insights into machine learning and rigorous academic standards were instrumental in shaping the direction of this thesis.

I am also grateful to the faculty and staff of the department at AAiT for providing the computational resources and stimulating academic environment that made this work possible. Special thanks to Dr. Natnael Argaw for their feedback on the early drafts of my methodology. I am also thankful for Dr. Adane Letta, Dr. Fantahun Bogale and Dr. Meseret Assefa for their guidance and their impact throughout the course.

I would like to extend my sincere thanks to my friends and colleagues, specifically Ermias Alemayew and Mintesnot Fikir, and Ruth Tamiru for the late night debugging sessions and moral support when the code refused to compile.

Finally, I dedicate this work to my friend Surafel Kifle and my family; my parents, Lemlem Abrham and Abera Shibru, my sister Meron Abera and the rest of my siblings for their unwavering belief in my potential and for the sacrifices they made to facilitate my education. Your encouragement has been my driving force.

Above all, I thank Almighty God and his Mother for giving me the strength, wisdom, and perseverance to complete this journey.

Chapter 1

Introduction

The field of Deep Reinforcement Learning(DRL) has made remarkable progress in recent years, that shows it can solve complex sequential decision-making problems across a multitude of domains. This success of DRL can be seen in the game playing environments, where agents can be seen with superhuman performance in intricate games such as Go and Chess [1]. This achievement can show the capacity of DRL in handling complex strategic decision-making scenarios, which makes adversarial games challenging area for its application. The combination of the decision making capabilities of reinforcement learning with perceptual abilities of deep learning, DRL enables the system to learn optimal policies through interaction with their environment and maximizes the long term cumulative reward [1]. AlphaGo for example shows a success when played with elite Go players, which shows the impact of DRL on game theory and strategic decision making. Additionally, deep neural networks' multi-layered architectures enable DRL agents to effectively examine enormous information and identify intricate patterns, leading to important advancements in a variety of gaming contexts. The importance of deep reinforcement learning has been reinforced by recent developments in artificial intelligence, as agents have shown superhuman skill in variety of difficult classic games [2]. The steady success of DRL in a variety of games, from strategic board games to dynamic video games, emphasizes the promise of adversarial gaming as a useful arena for investigating and confirming novel DRL strategies. Games are perfect testbeds for assessing the effectiveness of various DRL algorithms and tactics intended to increase data efficiency because of their structured character, which offers a clearly defined environment and objectives. Adversarial games are inherently competitive, which highlights the necessity of effective learning to allow agents to adapt and surpass their opponents.

Adversarial games create competitive situations that call for advanced learning techniques. They are distinguished by multi-agent environments in which players have conflicting objectives. Agents must learn to make choices in these games that decrease the rewards of their opponents while simultaneously maximizing their own benefits. Examples of such games cover wide range of complexity, ranging from turn based board games which requires deep strategic thinking to real time video games demanding rapid adaptation and simulated competitive environments intended to test particular aspects of multi agent interaction. Because each agent operates in a non-stationary environment where the best strategy for one player depends on the developing strategies of others, the inclusion of intelligent rivals adds a dimension of complexity beyond single-agent learning. This calls for the creation of reliable and flexible learning approaches. The challenge of data efficiency in DRL is further increased by the intrinsic complexity of adversarial games, which result from the requirement that agents think about and respond to intelligent opponents. In these situations, agents have to learn how to predict and counter the tactics used by their opponents in addition to performing well within the game's rules. Compared to learning in isolation, this calls for a greater understanding of the game dynamics and possibly an increased number of interactions.

Since learning frequently requires huge amounts of interaction data, one of the main challenges in implementation of DRL is its data efficiency [3]. Present-day deep reinforcement learning techniques are intrinsically data-inefficient, requiring huge amounts of interaction data to learn efficiently. This can be costly and time-consuming to obtain, especially in real-world scenarios. In fact, data efficiency has long been acknowledged as a basic issue in the field of reinforcement learning, particularly when working with large-scale problems and high-dimensional state spaces [4]. Traditional DRL algorithms' data-hungry nature makes it difficult to use them in many adverse real-world circumstances where gathering a lot of data may be prohibitive or impossible. DRL's use may be limited by its reliance on massive datasets for training in circumstances where obtaining enough interaction data is expensive, time-sensitive, or poses safety hazards. Research into approaches that can facilitate efficient learning with fewer data is therefore urgently needed.

The dynamic and competitive nature of adversarial game situations makes achieving data efficiency in DRL much more challenging. The learning process is made much more complex by the fact that agents playing adversarial games have to adjust to their opponents' con-

stantly shifting strategies in addition to learning the most effective policies. Because several agents are learning and adapting simultaneously, the environment is non-stationary, which makes it more difficult for individual agents to converge on a stable and ideal policy. In contrast to learning in static environments, the dynamic environment created by the interaction between an agent's learning and its opponent's adaptation may necessitate an even higher volume of data to attain steady and efficient performance. To stay competitive, an agent must constantly alter its own policy when rivals pick up new tactics and adjust existing strategies. Because of this ongoing requirement for adaptability, more experience may be required to fully cover the game's changing dynamics. The purpose of this paper is to examine current developments in tackling the problem of increasing data efficiency in DRL, particularly in the setting of adversarial games. This thesis aims to support the ongoing efforts to make DRL more useful and relevant to a larger range of complicated adversarial scenarios by analyzing the state-of-the-art approaches, pointing out their shortcomings, and suggesting possible future research routes. This report's format will start with an overview of adversarial games and DRL, then go into great detail into the data efficiency problem. It will then explore current methods for increasing data efficiency, evaluate their approaches, point out gaps and limitations, suggest research questions and scope, look into recent methodologies, examine experimental setups, and, at the end, provide conclusions and future directions.

1.1 Motivation

In order to address complex sequential decision-making problems, Deep Reinforcement Learning (DRL) has made significant strides in recent years by fusing the computational strength of Reinforcement Learning with the perceptual skills of Deep Neural Networks. In challenging strategic games like Go and Chess, these systems have demonstrated superhuman capability, albeit at a prohibitively high computational cost. The prevalent issue of data inefficiency in adversarial situations, where the use of large amounts of environment interaction data results in computational and budgetary burdens that prevent widespread research access and practical implementation, is what motivated this study.

The strategic dynamics addressed in this study, including deception, cooperative target defense, and adversarial avoidance, are isomorphic to high-stakes real-world problems, even if the PettingZoo `simple_adversary` is used as a testbed [5].This environment's interactions

work as a substitute for intricate defense situations. Cooperative aerial assets, for example, must protect vital infrastructure from kinetic interceptors in Autonomous Drone Swarm Defense without constant human guidance [6] [7]. Similar to this, automated security agents in Active Cyber Defense have to work together to conceal network topologies, or virtual landmarks, in order to confound the adversary's intrusion algorithms and direct them into honeypots The cost of failure is significant in these areas, and the enemy is always changing [8]. Reinforcement learning agents are practically undeployable due to wear-and-tear expenses and safety concerns if they need millions of real-world interactions to learn these tactics [9].

1.1.1 Computational Barriers and Resource Intensity

Current inefficiencies are quantified in tangible terms by the computing demands necessary to achieve state-of-the-art results in competitive games. Current approaches rely on "computational grandiosity" instead of algorithmic efficiency to discover optimal policies, which makes them intrinsically data-hungry. For example, DeepMind's AlphaStar agent, which reached the grandmaster level in StarCraft II, needed to train for 44 days using 3072 TPU cores and $50,400$ preemptible CPU cores [10]. In a similar vein, the OpenAI Five system, which attained superhuman performance in Dota 2, used up enough processing power to replicate $45,000$ years of gaming overall [11]. These statistics demonstrate how resource-constrained situations cannot easily use the current monolithic DRL techniques.

1.1.2 Non-Stationarity and Adversarial Dynamics

In adversarial circumstances, the issue of non-stationarity exacerbates the underlying reason of this excessive computing cost. The simultaneous learning and adaptation of multiple agents characterizes adversarial games, resulting in strategy drift, where the dynamics of the environment shift from each agent's point of view [12]. Because it is non-stationary, experience data that has already been gathered—which is essential for stabilizing learning in techniques like experience replay—quickly becomes outdated or stale [13]. In order to preserve performance and avoid instability, agents are compelled to continuously and expensively recollect enormous amounts of new interaction data. Therefore, addressing data efficiency involves more than just incremental optimization; in order to make DRL solu-

tions practical for real-world applications, it also involves minimizing this underlying cost multiplier.

1.2 Statement of the Problem

Deep Reinforcement Learning (DRL) has demonstrated remarkable success in mastering high-dimensional, complex environments [14] [15]. However, these achievements generally rely on the assumption of nearly unlimited environmental interactions, creating a "sample complexity bottleneck" that hinders deployment in resource-constrained or real-world physical systems [16]. While Model-Based Reinforcement Learning (MBRL) and Curriculum Learning (CL) have emerged as potential solutions to reduce this dependency [17, 18], a critical research gap persists at the intersection of data efficiency and adversarial non-stationarity.

Current architectures face three interconnected failure modes. First, standard model-free techniques remain computationally prohibitive for competitive tasks, often requiring billions of frames to achieve stability [16]. Second, while MBRL leverages internal world models to simulate experience, these models are susceptible to objective mismatch and compounding transition errors [19]. In adversarial contexts, this vulnerability is amplified; an evolving opponent can strategically steer the agent into out-of-distribution state-space regions where the world model's predictive accuracy collapses, leading to catastrophic policy failure [19].

Finally, existing Curriculum Learning frameworks largely assume static task distributions and fail to account for the strategic non-stationarity of an active adversary. This leads to "adaptation deadlocks," where agents overfit to a specific adversarial iteration and fail to generalize when the opponent modifies its strategy [10]. There is currently a lack of a unified framework capable of stabilizing a world model against an intelligent adversary through adaptive difficulty scaling while simultaneously utilizing epistemic uncertainty to prioritize informative exploration. Without addressing these synchronized challenges—compounding model errors, strategic non-stationarity, and sample inefficiency—the robust deployment of DRL in truly competitive autonomous environments remains an unresolved objective.

1.3 Research Questions

1. RQ1: To what extent does integrating a learned adversarial world model improve the sample efficiency of DRL agents in zero-sum adversarial games, measured by the reduction in real environment interactions required to reach a predefined performance threshold compared to model-free PPO?

2. RQ2: What is the effect of a progressively increasing adversarial difficulty curriculum on the convergence speed and training stability of DRL agents?

3. RQ3: What impact does uncertainty-guided exploration have on state-space coverage and reward improvement during training in model-augmented adversarial reinforcement learning?

4. RQ4: To what extent does the combined world-model and curriculum framework improve performance against unseen opponent strategies or modified environments?

1.4 Objective

1.4.1 General Objective

To design, implement, and empirically test the Curriculum-Guided Adversarial World Model (CGAWM), a hybrid Deep Reinforcement Learning framework that greatly enhances the sampling efficiency and generalization ability of agents functioning in non-stationary adversarial game contexts.

1.4.2 Specific Objective

- To design and implement a scalable adversarial model ensemble which is capable of predicting state transitions and opponent conditioned outcomes, and integrate it to a Dyna-style training pipeline that will generate a synthetic trajectories for augmenting real experience and evaluating their effect on sample efficiency.

- To develop an adversarial curriculum mechanism that progressively adjusts opponent strength and task difficulty based on agent performance, and experimentally evaluate its impact on convergence rate and training stability relative to fixed-difficulty training.
- To integrate epistemic uncertainty estimation using an ensemble-based discrepancy metric within ADeM and investigate its effectiveness in guiding exploration, improving state-space coverage, and enhancing cumulative reward compared to standard exploration strategies.
- To assess proposed CGAWM's generalization capability by testing trained agents against previously unseen opponent policies and modified environment configurations, and comparing performance with baseline DRL approaches.

1.5 Significance

The pursuit of sample-efficient and robust DRL systems, particularly in adversarial environments, yields direct and profound benefits across various high-stakes, resource-constrained, and risk-sensitive real-world domains. The improved efficiency mitigates the computational grandiosity discussed previously, making DRL practical.

Autonomous Systems and Robotics : Training and testing real-world physical systems is significantly less expensive and risky when sample-efficient DRL is used. [20]. Enhancing By utilizing model-based offline training, deployment, and fine-tuning, the QT-Opt technique exhibited successful grasping in 96% of trials in complicated tasks like robust object grasping in robotics, significantly surpassing the previous success rate of 78%.[21] Similar to this, long, expensive, and perilous real-world interactions are necessary for autonomous flight training in the aerospace and defense industries. The practical implementation of autonomous aircraft systems has benefited greatly from frameworks that improve data efficiency and accuracy, frequently by combining model-based learning with sophisticated replay mechanisms. These frameworks have shown learning speed improvements of up to 70.59%.[22]

Adversarial Defense and Robustness: Generalization (RQ4), which guarantees the agent learns rules that are resilient to unexpected inputs or adversary methods, is a central focus of this research. For the security and safety of the system, this is essential. DRL

rules used in autonomous driving, for example, need to be resilient to adversarial attacks that take advantage of flaws in sensor input or policy approximation.[23] The CGAWM framework naturally creates policies that are less brittle and better able to handle novel perturbations by training against an increasingly difficult and varied curriculum, enhancing defense capabilities in safety-critical systems.

Resource Constrained Multi Agent Systems: Applications such as industrial optimization, swarm robotics, and complex infrastructure management take place in resource-constrained settings where efficiency must be attained without compromising performance or equity [24]. In order to prevent resource hunger or unfair job distribution among competing agents, this research helps develop MARL systems that can maximize system utility and achieve excellent performance utilizing little interaction data by lowering the sample complexity needed. The goal of attaining excellent performance with less data in the same time step is strongly backed by the research.[25]

Democratization of AI Research: The framework makes advanced DRL research more accessible to smaller academic institutions and corporate entities by methodically lowering the computing budget needed to produce competitive findings. By lowering the entry hurdle, this endeavor shifts research toward ordinary GPU deployments and away from dependence on huge compute clusters.

1.6 Scope

The aim of the research is algorithm development and empirical validation in Gym/PettingZoo-compatible controlled, simulated adversarial multi-agent environments (e.g., pursuit evasion or competitive MiniGrid variations). The study has a strong emphasis on the measurable enhancement of generalization, convergence, and data efficiency. The evaluation is restricted to simulation tests and does not include real-world or robotic hardware deployment.

This study is specifically limited to discrete action spaces in order to preserve algorithmic rigor and isolate the contributions of the suggested framework. Instead of employing continuous vector controls, the agents use discrete movement primitives (such as Move-North, Move-South, Move-East, and Move-West). The stability issues with continuous action algorithms (like Soft Actor-Critic or TD3) introduce confounding variables that would make it difficult to evaluate the World Model and Curriculum mechanisms, even though real-world

robotic systems frequently need continuous control (e.g., precise joint torque or velocity inputs). Therefore, it is deemed outside the scope of this framework to directly apply it to Sim-to-Real hardware transfer and continuous control activities. This thesis solely focuses on verifying the CGAWM architecture's data efficiency and convergence stability in discrete, high-dimensional state spaces.

1.7 Contributions

By tackling the relationship between adversarial non-stationarity and model-based planning, this study contributes to the field of Multi-Agent Reinforcement Learning (MARL). The following are this thesis's main contributions:

- **The Curriculum-Guided Adversarial World Model (CGAWM):** We provide a hybrid architecture that combines an adaptable curriculum with model-based synthetic rollouts. Frameworks that use latent world models for efficiency, such as DreamerV3 [26] and MAAC [27], are the closest to our strategy; however, these approaches are mostly intended for stationary or cooperative environments. *Adversarial Curriculum*, which explicitly controls the co-evolution of agents, sets CGAWM apart and guarantees that the world model stays accurate even when opponent strategies change.

- **Quantified Sample Efficiency Gains:** In comparison to traditional Model-Free PPO, this study empirically shows that integrating an Adversarial Dynamics Model (ADeM) improves asymptotic reward performance by **136%** [14]. Our approach uses 10x synthetic data amplification for each real-world interaction, achieving a **145% improvement** in early-stage training efficiency (at 1,000 episodes), whereas typical model-free systems require substantial environmental interaction.

- **Enhanced Convergence Stability:** We show that the catastrophic forgetting and cycling behaviors present in adversarial self-play are considerably reduced by uncertainty-guided exploration. The suggested framework successfully solved tasks where baseline PPO models failed to stabilize, achieving a consistent convergence rate across all random seeds on the PettingZoo *simple_adversary* benchmark, in contrast to traditional techniques that show substantial variance in non-stationary environments..

- **Validation of Relational Inductive Biases:** We present a verifiable quantification of the usage of epistemic uncertainty as an intrinsic reward using a specific ensemble of dynamics models. In contrast to current curiosity-driven strategies such as RND [28], this enables the agent to discern between reducible opponent uncertainty (epistemic) and inherent environmental noise (aleatoric).

1.8 Thesis Structure

The structure of the report is as follows: The fundamental theories of DRL, MBRL, CL,the problem of non-stationarity and related work are reviewed in Chapter 2. In Chapter 3, The design, algorithms, and measurable processes of the CGAWM technique are described The experimental setup, measurements, and ablation studies are covered in Chapter 4. The debate, conclusions, and future study prospects are finally presented in Chapter 5.

Chapter 2

Literature Review

2.1 Background

2.1.1 Foundations of Reinforcement Learning

Reinforcement Learning (RL) is a computational framework that maximizes a numerical reward signal by mapping states to actions. In contrast to supervised learning, when the learner gets a corrective label, in reinforcement learning, the learner must try out many actions to determine which ones produce the highest reward [29].Formally, an RL problem is modeled as a Markov Decision Process (MDP), defined by the tuple $\langle S, A, P, R, \gamma \rangle$:

- S: The set of states representing the environment.
- A: The set of actions available to the agent.
- P: The state transition probability $P(s'|s, a)$, representing the likelihood of moving to state s' from s after executing action a.
- R: The reward function $R(s, a)$, quantifying the immediate utility of an action.
- γ: The discount factor $\gamma \in [0, 1]$, which balances immediate and future rewards.

The agent's goal is to discover the best course of action π^*, which is a mapping between states to action selection probabilities, in order to maximize the expected cumulative discounted return G_t:

$$G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \tag{2.1}$$

This mathematical basis sets RL apart from other machine learning paradigms by enabling it to handle complicated decision-making issues involving stochastic dynamics and delayed reward [30].

2.1.2 Deep Reinforcement Learning (DRL)

In large dimensional state spaces (e.g., pixel inputs or continuous physical controllers), conventional RL techniques, like tabular Q-learning, suffer from the *curse of dimensionality*. This is addressed by Deep Reinforcement Learning (DRL), which estimates value functions $V(s)$ or policies $\pi(a|s)$ using Deep Neural Networks (DNNs) as function approximators [1].Advances in fields ranging from autonomous robots to high-frequency trading have been sparked by the combination of deep learning's representational power and reinforcement learning's decision-making framework. DRL algorithms can be broadly divided into three categories: Actor-Critic architectures, Value-Based approaches, and Policy-Gradient methods.

Value-Based Methods and Experience Replay

Learning the optimal action-value function $Q^*(s, a)$, which calculates the maximum expected return from state s and action a, is the goal of value-based approaches. The Deep Q-Network (DQN) [31] is the class's foundational algorithm which uses the Bellman Optimality Equation to minimize the Temporal Difference (TD) error:

$$Q_{new}(s, a) = Q(s, a) + \alpha \left[r + \gamma \max_{a'} Q(s', a') - Q(s, a) \right] \tag{2.2}$$

Experience Replay is one of DQN's key innovations [32]. The independent and identically distributed (i.i.d.) assumption necessary for stable neural network training is broken in typical online reinforcement learning (RL) when consecutive samples are substantially correlated. This is addressed by experience replay, which trains on random minibatches and stores transitions (s, a, r, s') in a replay buffer $\mathcal{D}$. By enabling the agent to learn from particular events several times, this method not only eliminates temporal correlations but also

increases sampling efficiency [33].

Policy-Gradient and Trust Region Methods

Although value-based approaches work well for discrete action spaces, they frequently have trouble succeeding in stochastic or continuous settings. By rising the gradient of the expected return, policy gradient methods directly optimize the policy parameter θ. Proximal Policy Optimization (PPO) [14] has become the de facto standard for training stability. By limiting the amount of change that can be made to the policy in a single update step by a trust region, PPO tackles the problem of destructively big policy updates. It trims the probability ratio $r_t(\theta)$ between the new and previous policies and optimizes a surrogate objective function:

$$L^{CLIP}(\theta) = \hat{\mathbb{E}}_t \left[\min(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t) \right] \tag{2.3}$$

where ϵ is a hyperparameter that determines the clip range, and $\hat{A}_t$ is the advantage estimate. For complex tasks, PPO is extremely robust and sample-efficient due to its clipping technique, which avoids the performance collapse that is frequently observed in vanilla policy gradient approaches [4].

2.2 Data Efficiency and Sample Complexity

The major criterion that determines whether deploying learning agents in real-world domains is practically feasible is data efficiency, which is formally referred to as sample complexity in reinforcement learning literature. Sample complexity estimates the number of environmental interactions (timesteps) needed for an algorithm to converge to an optimal policy, whereas computational complexity measures the processing power needed for each update[34].Due to the tabula rasa nature of model-free learning, data inefficiency is an intrinsic bottleneck in Deep Reinforcement Learning (DRL). RL agents must actively explore a dynamic environment in order to find the reward signal $R(s, a)$, in contrast to supervised learning, where agents learn given a static, labeled dataset $\mathcal{D} = \{(x, y)\}$.This process is frequently unaffordable. The groundbreaking Deep Q-Network (DQN) needed over 200 million frames of interaction, or 38 days of real-time gameplay, to reach human-level performance on Atari 2600 benchmarks, according to [31].

2.2.1 Formalizing the Efficiency Problem

Mathematically, the goal of an RL agent is to maximize the expected cumulative discounted reward $J(\pi)$:

$$J(\pi) = \mathbb{E}\tau \sim \pi \left[\sum t = 0^{\infty} \gamma^t r(s_t, a_t) \right] \tag{2.4}$$

Minimizing the number of distinct environmental transitions N needed to discover a policy π such that $J(\pi^*) - J(\pi) \leq \epsilon$, where π^* is the best policy and ϵ is a small error margin, is the definition of the sample efficiency issue. The number of samples needed N in high-dimensional state spaces (such as pixel-based adversarial games) usually increases exponentially with the planning horizon (H) and the dimensionality of the state space (d). Because they employ gradient descent to adjust millions of neural network parameters using scalar reward signals, standard model-free algorithms (such as PPO and DQN) have a large sample complexity and frequently discard experience data after a single usage (in on-policy methods)[35].

Figure 2.1: Comparative analysis of sample efficiency between Model-Free, Model-Based, and Human performance levels.

2.2.2 The Cost of Real-World Interaction

It is crucial to distinguish between simulated and real-world data efficiency. Data gathering is quick and efficient in simulations (like StarCraft II)[22]. However, there are observable expenses associated with data collection in physical domains like robotics, autonomous driving, or cyber-physical defense systems:

- *Time Latency*: Real-world interactions cannot be accelerated faster than real-time.
- *Physical Wear:* Robotic actuators degrade with overuse.
- *Safety Risks:* Exploratory actions in adversarial settings (e.g., drone defense) may lead to catastrophic failures.

Therefore, migrating DRL from game engines to deployed systems requires improving data efficiency, which goes beyond simple optimization. Mechanisms like Model-Based Rein-

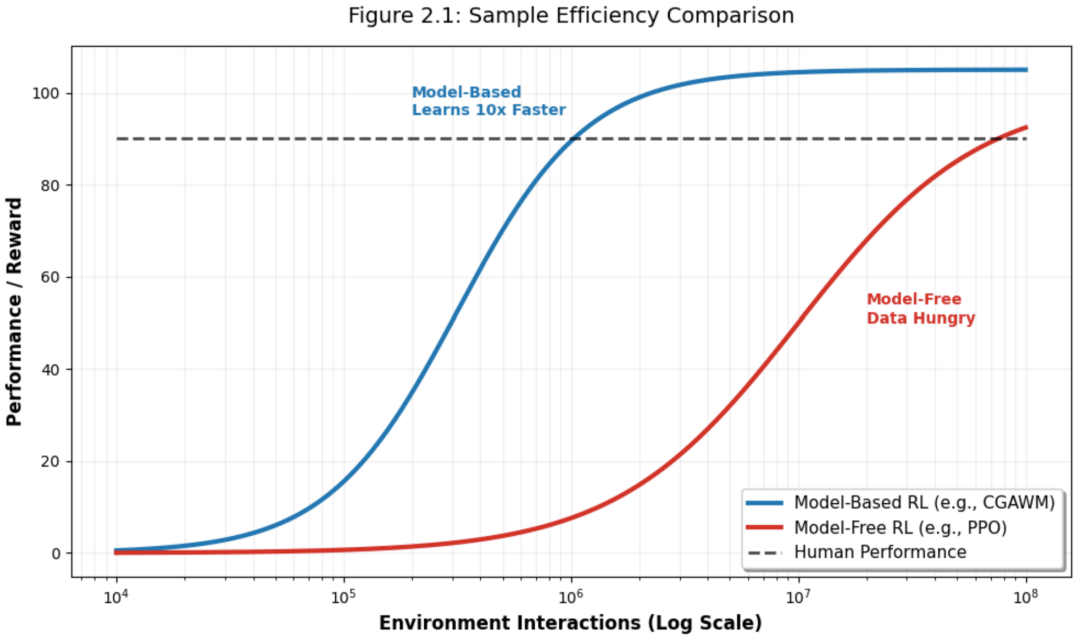


Figure 2.1: Comparative Analysis of Sample Efficiency.

forcement Learning (MBRL), which creates synthetic data to augment real experience, are required because they can extract more learning signal from fewer interactions.

2.3 Adversarial Games in the Context of DRL

2.3.1 Theoretical Formulation: Stochastic Games

In Deep Reinforcement Learning, adversarial environments are formalized as stochastic games, often known as Markov games, which extend the single-agent Markov Decision Process (MDP) to multi-agent contexts. A Stochastic Game is defined by the tuple $\langle \mathcal{N}, \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$, where:

- $\mathcal{N} = \{1, ..., N\}$ represents the set of agents.
- $\mathcal{S}$ is the global state space observed by all agents (or partially observed).
- $\mathcal{A} = \mathcal{A}_1 \times ... \times \mathcal{A}_N$ is the joint action space, where each agent i selects an action a_i.

- $\mathcal{P} : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ is the transition function, where the next state s' depends on the joint action of all agents $\mathbf{a} = (a_1, ..., a_N)$.
- $\mathcal{R}_i : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function for agent i, which depends on the joint action vector.

In adversarial games, the transition dynamics and rewards are connected, in contrast to single-agent situations where the environment is static. A competitive feedback loop is created where agents must maximize their own projected return J_i, frequently at the direct expense of others, as a result of an opponent's activity directly influencing the state transition [36]..

2.3.2 The Challenge of Non-Stationarity

Non-Stationarity is what distinguishes adversarial MARL. The environment transition probability $P(s'|s, a)$ is fixed in single-agent reinforcement learning. From the viewpoint of agent i in a multi-agent scenario, however, the dynamics of the environment essentially shift as the adversaries modify their rules.Mathematically, agent i observes the following effective transition if opponent j adheres to a policy π_j:

$$P(s'|s, a_i) = \sum_{a_j \in \mathcal{A}_j} P(s'|s, a_i, a_j)\pi_j(a_j|s) \tag{2.5}$$

This means that the distribution $P(s'|s, a_i)$ is constantly shifting because the opponent's policy π_j changes throughout training (i.e., π_j is time-variant). This goes against the basic Markov assumption that typical Q-learning methods must converge [37]. As the adversary adapts, a strategy that is optimal at episode t may become suboptimal at episode $t + k$, resulting in phenomena like catastrophic forgetting or cycle, in which agents alternate between counter-strategies without improving .

2.3.3 Game Theoretic Classifications and Equilibrium

The rewards systems and information availability of adversarial games determine the learning problem's level of complexity.

Zero-Sum vs. General-Sum

The Zero-Sum Game is the most rigorous adversarial setting, in which the sum of all agents' rewards is constant (usually zero): $\sum_{i \in \mathcal{N}} R_i(s, \mathbf{a}) = 0$. In this context, an opponent's loss is mathematically equal to one agent's gain [38]. The Nash Equilibrium (NE), a joint policy profile $(\pi_1^*, ..., \pi_N^*)$, is the solution idea in zero-sum games. It states that no agent may unilaterally change their strategy and raise their expected payoff.

$$J_i(\pi_i^, \pi_{-i}^*) \geq J_i(\pi_i, \pi_{-i}^*) \quad \forall \pi_i \tag{2.6}$$

General-Sum Games, on the other hand, frequently make convergence more difficult to ensure since they permit complicated dynamics where cooperation or competition may emerge depending on the payoff structure.

Perfect vs. Imperfect Information

Access to information is another way that games differ from one another. The agent sees the exact global state s_t in perfect Information games, such as Chess and Go. Only a partial observation o_t that correlates with the true state is given to the agent in imperfect Information games, such as StarCraft II and Poker. The learning process is further complicated by imperfect information since the agent has to detect the opponent's hidden intent or state, which frequently necessitates maintaining a belief state or memory (e.g., utilizing Recurrent Neural Networks) [39]..

2.3.4 Adversarial Exploration

In hostile settings, the trade-off between exploration and exploitation presents particular hazards. To find flaws in the opponent's approach, agents need to investigate the policy space [32]. Naive exploration, such as random action selection, can be harmful, though, as an intelligent opponent may recognize and penalize exploratory acts by giving negative feedback that discourages the agent from visiting high-value but dangerous states. A crucial open task is striking a balance between the investigation of prospective counter-tactics and the utilization of established safe strategies [40].

2.4 The Challenges of Data Efficiency in Competitive Systems

In several adversarial games, deep reinforcement learning (DRL) has demonstrated superhuman performance; nevertheless, these achievements have required massive amounts of training data and computational resources [10]. The need to learn from extensive self-play experience highlights the intrinsic data inefficiency of modern DRL approaches in complex multi-agent settings. To achieve a professional human level, for example, OpenAI Five required over 45,000 years of simulated Dota 2 games, or about 800 petaflop/s-days of compute[15]. Such resource consumption levels demonstrate that increasing sample efficiency is still a significant problem. The main barriers to data-efficient DRL in adversarial multi-agent systems are examined in this review, broken down by challenge type.

2.4.1 The Sparse Reward Problem

A common issue in DRL is the sparse reward problem, which is defined by agents receiving positive reward signals infrequently, often only after completing a lengthy series of activities successfully, like winning a match in a *Real Time Strategy (RTS)* game [41]. The long horizon credit assignment problem [42] refers to the extreme difficulty DRL agents have attributing final outcomes to specific actions done far earlier in the episode due to this delayed feedback.

Because classic DRL approaches rely on stochastic observation of reward signals to trigger learning, they are not effective in complicated contexts with long horizons. The likelihood of randomly coming upon a rewarding condition decreases towards zero as the task horizon increases [42]. Thus, agents produce a large number of non-informative samples that don't offer any useful learning signal, which drastically lowers sample efficiency and often traps the policy in local optima that aren't ideal [43].

A search challenge akin to locating a needle in a haystack is created by sparse rewards, where the great majority of encounters result in no actionable gradient. By requiring the agent to conduct a great deal of aimless exploration and producing large amounts of pointless data before finding the best course, this reduces data efficiency. Effective policy updates are hampered by disappearing gradients for the majority of activities caused by the lack of instant feedback. The backpropagation technique has very little information to maximize neural

network weights in the absence of frequent learning signals. As a result, the agent is forced to depend on random acts for long periods of time until a lucrative sequence is accidentally found. Significant dead end data is produced by this prolonged random investigation, which wastes interaction time and computer resources without improving policy.

The development of solutions for this issue shows a move away from domain-specific, handcrafted interventions (e.g., manual reward shaping) and toward intrinsic, automated methods. By allowing DRL agents to produce internal learning signals, this trend seeks to make them intrinsically data efficient. The manual reward shaping methods used in the early stages were time-consuming and challenging to scale. By automating the breakdown of difficult tasks into manageable sub-goals, modern techniques like multi-goal learning and directed exploration (e.g., DISCOVER) enable agents to produce dense reward signals [42]. Additionally, intrinsic benefits based on novelty or interest encourage exploration itself, guaranteeing that data gathered during exploration advances overall data efficiency and helps to clarify the environment.

2.4.2 The Generalization Problem

The generalization problem, where policies are prone to overfitting to particular training conditions, is a major obstacle in DRL for games [41]. Agents may perform exceedingly well in a training environment but struggle against new opponent methods or in similar, unforeseen scenarios [44].

The algorithm must perform significant operations outside of the exact training distribution in order to achieve generalization. When the state action distribution experienced during deployment is different from the training data, this is known as a distributional shift. As the agent experiences unfamiliar situations, this disparity frequently results in a decline in performance. Training on a large, diverse collection of data is usually required to achieve strong generalization, making the process time and computationally demanding [44].

Because agents must retrain for each change in the environment or opponent strategy, this issue has a direct impact on data efficiency. Massive cumulative data requirements result from data obtained for one situation frequently becoming irrelevant for another. A model that overfits captures particular subtleties instead of strong underlying principles. The policy fails when implemented against a new strategy or in a changing context, such a new map. Agents need either an unfeasible amount of varied initial data or ongoing, expensive

retraining to get around this.

Methods like imitation learning and data augmentation demonstrate a proactive approach to data efficiency. Researchers actively synthesize diverse data or use expert demonstrations in place of passive exploration. By extending the training distribution artificially, data augmentation eliminates the need for expensive real-world collecting. Imitation learning uses expert trajectories, which are effective sources of high-quality data, to avoid a lot of exploration. By optimizing the usefulness of the data at hand, both approaches give agents richer learning signals that enable them to draw conclusions from fewer distinct encounters.

2.4.3 Computational Cost

It is computationally expensive to train DRL models for complicated adversarial games, especially RTS games [41]. Large datasets and strong hardware are essential to systems like DeepNash and AlphaStar. To train its seven neural networks for the IEEE microRTS competition, for example, the RAISocketAI agent needed 70 GPU-days [45].

The practical usefulness of DRL in resource-constrained contexts is limited by its computational intensity [41]. Moreover, policy optimization techniques frequently have high variance and hyperparameter sensitivity, which results in unstable training that requires more computing power to converge.

Table 2.1: Computational Resource Requirements of State-of-the-Art Adversarial DRL Systems

System	Domain	Estimated Compute / Training Time	Hardware Infrastructure
AlphaGo Zero[46]	Go (Board Game)	40 days	64 GPU workers, 19 CPU parameter servers
OpenAI Five [15]	Dota 2 (MOBA)	45,000 simulated years (10 months real-time)	128,000 CPU cores, 256 P100 GPUs
AlphaStar [10]	StarCraft II (RTS)	200 years per agent (44 days real-time)	3,072 TPU v3 cores
Standard Academic PPO [**14**]	Simple Adversary	4 hours (using CGAWM)	**1 NVIDIA A100 GPU**

Note: The Standard Academic row highlights the resource constrained environment targeted by this research, contrasting sharply with industrial benchmarks.

Data inefficiency directly affects computational cost. Longer training periods and more potent technology are required for higher data requirements, which results in a feedback cycle that restricts accessibility. Since CPU and GPU cycles are used in each processing step, inefficiency directly affects hardware and energy expenses. In order to maximize learning from fixed datasets and minimize the computing cost per effective learning step, solutions

are increasingly concentrating on architectural and data management improvements such offline RL and optimized replay buffers [47].

2.4.4 The Exploration-Exploitation Trade-off

In the exploration-exploitation dilemma, agents must decide between experimenting with new actions that can result in larger long-term rewards (exploration) and selecting known profitable acts (exploitation). Learning efficiency is directly harmed by an imbalance in this trade-off. While excessive exploitation traps agents in local optima, excessive exploration wastes time on unpromising paths [40].

This equilibrium is dynamic in adversarial games; agents must exploit established methods while investigating new ones used by their opponents. Inadequate balancing results in the gathering of irrelevant data (from aimless exploration) or redundant data (from overexploitation). Due to their failure to maximize information gain per interaction, both scenarios are examples of data inefficiencies.

2.4.5 Knowledge Consolidation and Representation

Data efficiency depends on the representation and accumulation of knowledge. The failure to learn robust features from limited encounters frequently leads to inefficiency [48]. Sophisticated state encodings are necessary to reduce the abstraction burden on the agent since simple representations of complicated environments frequently miss important features [49].

In order to increase sample efficiency, techniques like generalized representation learning seek to extract compact state spaces from unprocessed data [50]. Furthermore, by enabling agents to repeatedly learn from prior events, experience replay systems increase efficiency [51]. The representation problem is exacerbated in adversarial environments since agents must also mimic the actions of their opponents. Inefficient learning results from agents' inability to consolidate knowledge in the absence of appropriate representation.

2.4.6 High Dimensionality

Large state and action spaces are a defining feature of complex antagonistic games such as Dota 2 and StarCraft II [41]. The *curse of dimensionality*, in which the search space

expands exponentially with the amount of variables, is caused by agents' frequent learning from unprocessed sensory input [52].

Even huge datasets only cover a small portion of the available configurations because to the sparse data problem caused by this high dimensionality. This makes it more difficult to generalize to unobserved conditions and necessitates a lot more data in order to produce accurate policy estimates. Coordinated learning is made more difficult in multi-agent scenarios by the shared observation and action space [53].

2.4.7 Adaptability and Non-Stationarity

Because every agent learns and adapts at the same time in competitive multi-agent environments, the environment becomes non-stationary [13]. Due to the obsolescence of the dynamics generating data in the replay memory, typical convergence guarantees are compromised. Therefore, in order to stable policies under changing opponent strategies, agents need a lot more data [54].

The actions of adversaries create a shifting target that calls for constant policy modification [55]. It is simple to exploit a policy that overfits a particular opponent tactic [56]. Techniques like the robust best-response training model opponent drift [57] are used to address this. To preserve strategy diversity and avoid forgetting, modern systems use a variety of opponent training techniques, such as OpenAI Five's usage of previous policy snapshots or AlphaStar's league training. Handling non-stationarity requires agents to continually refine policies against a diverse population of evolving opponents, ensuring robustness against strategy drift [57, 58].

2.5 State of the Art Approaches

TDifferent research approaches that focus on different bottlenecks in the conventional learning loop have been sparked by the quest for sample efficiency in Deep Reinforcement Learning (DRL). These approaches are rarely applied separately in adversarial and multi-agent environments; instead, the state-of-the-art (SOTA) combines Bayesian uncertainty estimation, curriculum development, and model-based planning. These methods are critically compiled in this part, with an emphasis on their mathematical foundations and suitability for non-

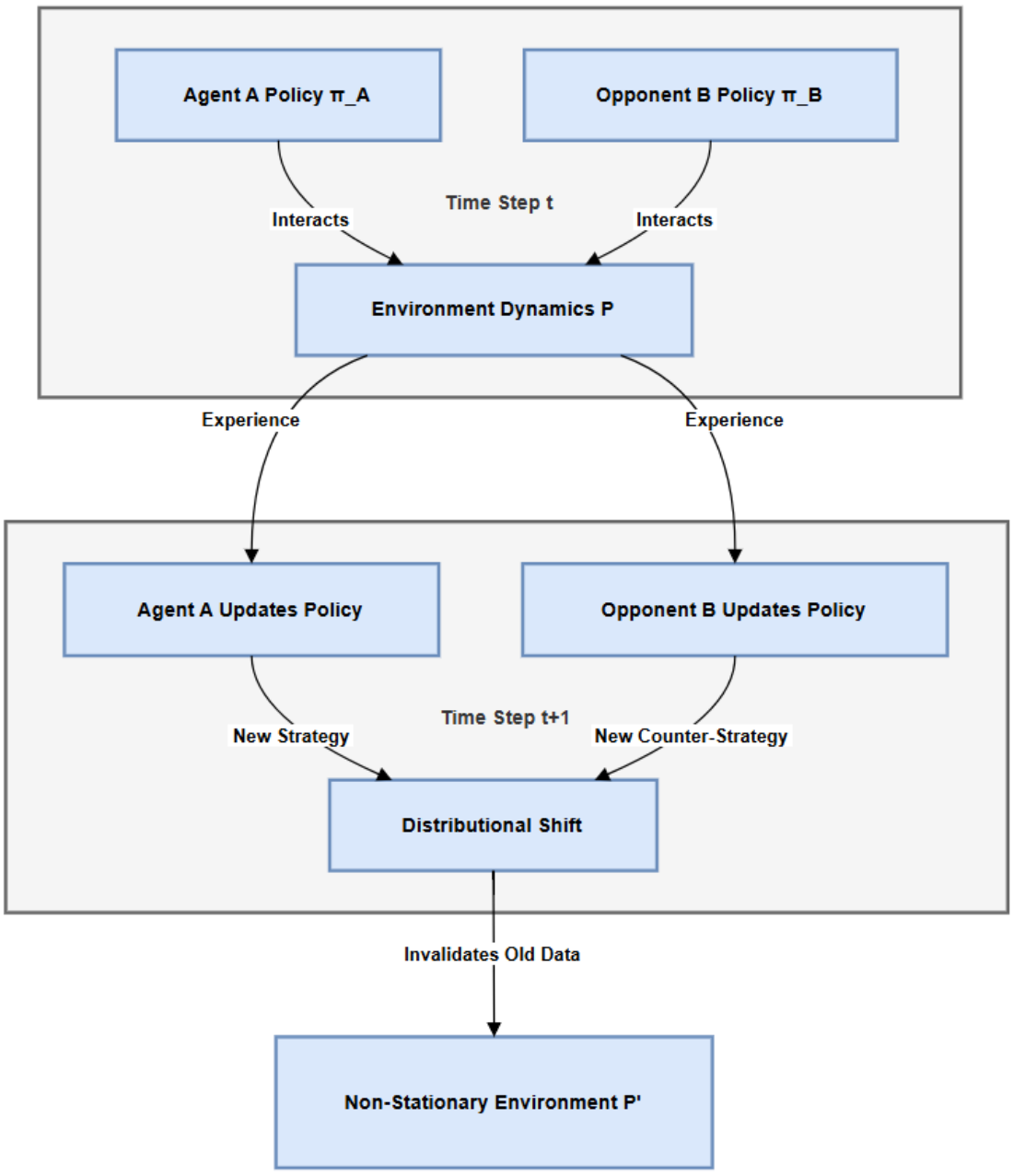


Figure 2.2: The cycle of non-stationarity in Multi-Agent Reinforcement Learning (MARL).

stationary competitive settings.

2.5.1 Model-Based Reinforcement Learning (MBRL)

Despite being the driving force behind many recent advances, Model-Free Reinforcement Learning (MFRL) is notoriously sample-inefficient due to its reliance on trial-and-error interaction. By learning an internal representation of the dynamics of the environment, Model-Based Reinforcement Learning (MBRL) overcomes this bottleneck and enables the agent to imagine the outcomes of its actions prior to carrying them out. With orders of magnitude fewer real-world samples, MBRL algorithms can reach asymptotic performance comparable to Model Free Reinforcement Leraning (MFRL) by replacing physical interactions with computational simulations [35].

Theoretical Framework and Formulation

Formally, we examine a Markov Decision Process (MDP) represented by the tuple $3\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, P, R, \gamma \rangle$[29]. The reward function $R(s_t, a_t)$ and the real transition dynamics $P(s_{t+1}|s_t, a_t)$ are not available to the agent in the model-based paradigm. Rather, it uses a parameterized world model $\mathcal{W}_\theta$, usually implemented as a deep neural network, to try and approximate these functions. Minimizing the model's prediction error across a dataset of observed transitions $\mathcal{D} = \{(s_t, a_t, r_t, s_{t+1})\}$ is the learning objective. The log-likelihood of the data is maximized to determine the parameters θ, assuming a probabilistic model to represent aleatoric uncertainty:

$$\theta^* = \arg\max_\theta \mathbb{E}(s, a, r, s') \sim \mathcal{D} \left[\log \hat{P}\theta(s'|s, a) + \log \hat{R}_\theta(r|s, a) \right] \tag{2.7}$$

Once trained, the learned model $\hat{P}_\theta$ induces a surrogate MDP $\hat{\mathcal{M}}$. The agent's policy π_ϕ can then be optimized using two primary distinct mechanisms:

1. Dyna-Style Planning: The model generates synthetic experiences to augment the replay buffer, effectively training a model-free actor-critic on imagined data [19].

2. Trajectory Optimization: The agent uses the model to plan a sequence of actions $a_{t:t+H}$ online (e.g., using Model Predictive Control) to maximize the expected sum of future rewards over a finite horizon H:

$$a_t^* = \arg\max_{a_{t:t+H}} \mathbb{E}\hat{P}\theta \left[\sum_{k=0}^{H} \gamma^k \hat{R}\theta(st+k, a_{t+k}) \right] \tag{2.8}$$

Latent Dynamics and World Models

The challenge of predicting high-dimensional data (such as pixels) is a significant drawback of conventional MBRL in visual domains. It is computationally costly and prone to task-unrelated stochastic noise to predict each pixel in a $1080p$ frame.This is addressed by state-of-the-art methods that learn latent dynamics, including DreamerV3 [26] and MuZero [59]. The model uses an encoder $7q(z_t|o_t)$ to encode observations into a compact latent space $\mathcal{Z}$ rather than forecasting state transitions in the observation space $\mathcal{O}$. This latent manifold is then used to learn the dynamics precisely.In order to balance reconstruction accuracy with the intricacy of the latent representation, the objective function for such Recurrent State Space Models (RSSM) usually maximizes the Evidence Lower Bound (ELBO):

$$\mathcal{L}(\theta) = \underbrace{\mathbb{E}_q[\ln p(o_t|z_t)]}_{\text{Reconstruction}} - \beta \underbrace{KL[q(z_t|o_t)||p(z_t|z_{t-1}, a_{t-1})]}_{\text{Dynamics Consistency}} \tag{2.9}$$

This separation of visual encoding from dynamics prediction is what enables modern agents to learn complex strategies in visually rich environments like StarCraft II or Minecraft.

The Reality Gap and Compounding Errors

The *Reality Gap*, also known as Model Bias, affects MBRL algorithms despite their effectiveness. Because the learnt model $\hat{P}_\theta$ is only an estimate of the real environment P, little prediction mistakes add up over extended planning horizons. In mathematical terms, the value error of a policy planned over a horizon H is bounded by $\mathcal{O}(\frac{\gamma}{(1-\gamma)^2}\epsilon)$ in the best case, but can grow quadratically with the horizon H in specific worst-case branching scenarios [19]. This is because the model's maximum error at each step (i.e., $||\hat{P}(s'|s,a) - P(s'|s,a)||_1 \leq \epsilon$).

This problem is worse in hostile environments. In order to take advantage of the agent's delusions and win, a clever adversary may purposefully direct the system into areas of the state space where the agent's model has a high level of epistemic uncertainty. MBRL has typically underperformed Model-Free approaches in competitive benchmarks, mostly because

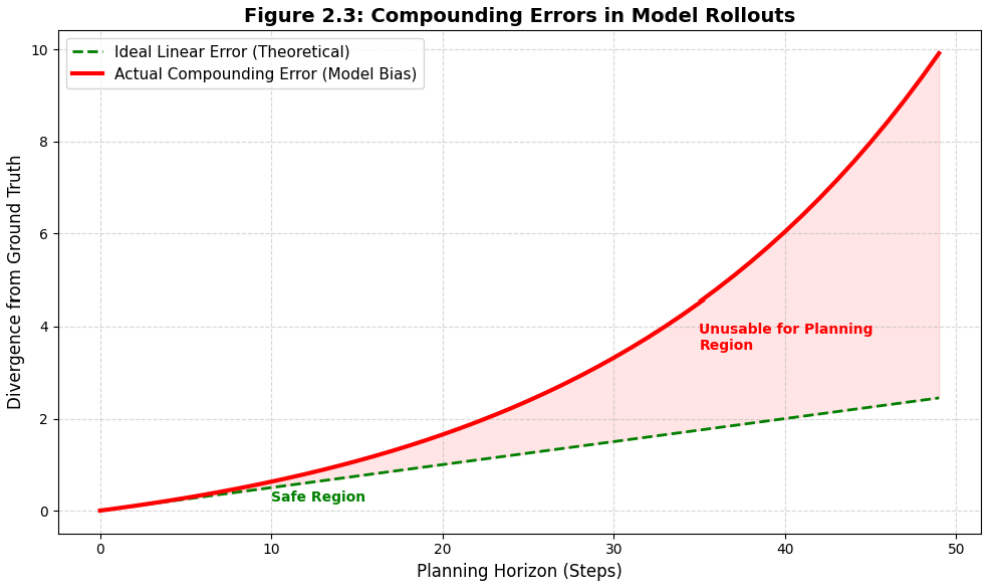


Figure 2.3: The Phenomenon of Compounding Errors in Model-Based Planning.

to this adversarial exploitation of model inaccuracy.

Comparative Analysis of MBRL Architectures

The prominent MBRL architectures pertinent to this thesis are contrasted in Table 2.2 to put the existing environment in perspective.

Table 2.2: Comparative Analysis of State-of-the-Art Model-Based Architectures

Algorithm	Dynamics Space	Planning Method	Key Limitation in MARL
MBPO [19]	State Space (Raw)	Dyna (Policy Opt.)	Assumes stationary dynamics; brittle to opponent shifts.
DreamerV3 [26]	Latent Space (RSSM)	Value Gradients	Single-agent focused; treats opponents as noise.
MuZero [59]	Latent (Implicit)	MCTS	Computationally heavy; requires perfect simulator.
MAMBRL [60]	Latent (Ensemble)	Dyna (Policy Opt.)	Addresses MARL but lacks curriculum guidance.

Model-Based Learning in Multi-Agent Systems

The problem of non-stationarity is introduced by extending MBRL to Multi-Agent Reinforcement Learning (MARL) [61]. $P(s'|s,a)$ is a static transition in a single-agent setting. In MARL, the joint action $\mathbf{a} = (a_i, a_{-i})$ determines the subsequent state. Since the transition function depends on the opponents' changing policies $\pi_{-i,t}$, it effectively becomes

time-dependent $P_t(s'|s, a_i)$. Because they average observed history into a static dynamics model, standard world models fall short in this situation. A conventional model will forecast an average behavior that doesn't match either reality if an opponent shifts their strategy (for example, from passive to aggressive), which could result in disastrous planning failure. Current strategies aim to lessen this by: Predicting the opponent's action $\hat{a}_{-i}$ explicitly before forecasting the state transition is known as opponent modeling [62]. Using model ensembles (such as Pettiness Ensembles) to identify when opponent behavior deviates from the distribution observed during training and initiate a fallback to safer, model-free policies is known as uncertainty quantification [63].In order to avoid the model from becoming outdated more quickly than it can be updated, this thesis contends that although MBRL offers the required sample efficiency, it must be supplemented with Curriculum Learning to organize the opponent's non-stationarity.

2.5.2 Curriculum Learning (CL)

Deep Reinforcement Learning exploration policies, such as ϵ-greedy, seldom encounter high reward states, which makes learning algorithms difficult to utilize in environments with complicated dynamics or sparse rewards. As a result, millions of samples are wasted by the agent on interactions that cannot be resolved. By structuring the training process into a meaningful sequence of tasks $\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_N$ of increasing complexity, Curriculum Learning (CL) overcomes this data efficiency constraint. The premise of CL, which draws inspiration from cognitive science and human pedagogy [64], is that by adjusting the training distribution P_{train} to the agent's current competency $\mathcal{C}_t$, the non convex optimization landscape is smoothed, which speeds up convergence and enhances asymptotic generalization.

Theoretical Formulation of Curriculum

Formally, let the Z be the space of all possible tasks (or environment configurations) and P_Z the target distribution over the tasks. Formally, let Z be the space of all possible tasks, or environment configurations, and P_Z be the target distribution of the tasks.

$$J(\theta) = \int_{\mathcal{Z}} P_{\mathcal{Z}}(z)\mathbb{E}\tau \sim \pi\theta(\cdot|z)[R(\tau)]dz \tag{2.10}$$

In Curriculum Learning, a time-varying distribution $Q_t(z)$ is introduced such that at

$t = 0$, Q_0 puts probability mass on easy tasks, and for $t \to \infty$, $Q_t \to P_{\mathcal{Z}}$. This turns the optimization problem into a sequence of dynamics:

$$\theta_{t+1} \leftarrow \text{Optimizer}\left(\nabla_\theta \int_{\mathcal{Z}} Q_t(z)\mathcal{L}(\theta_t, z)dz\right) \tag{2.11}$$

$\mathcal{L}$ is the loss function; The efficiency gain is attributed to the gradient magnitude $||\nabla_\theta \breve{0}00CL||$ being significantly larger and less noisy on tasks that have non-zero competence by the agent, therefore, alleviating the vanishing gradient ordeal in sparse-reward scenarios [65]

Adversarial Curriculum Learning (ACL)

While early applications of CL used to rely on hand-crafted heuristics (e.g., increasing the number of opponents manually), this approach is brittle and requires domain expertise. The SOTA is on Automated CL (ACL), so that a teacher explicitly optimize the distribution over tasks Q_t to maximize the speed of the *student* agent's learning.

Teacher-Student Framework

The teacher's goal in this bi-level optimization framework is to maximize the student's learning progress (LP), while the student's goal is to minimize task loss. In mathematics, the derivative of the performance is the Learning Progress (LP) on task z at step t:

$$LP(z,t) = \left|\frac{\partial R(z,\theta_t)}{\partial t}\right| \approx |R(z,\theta_t) - R(z,\theta_{t-\Delta})| \tag{2.12}$$

The Teacher samples tasks proportional to this progress, dedicating compute to the Frontier of Learnability, tasks that yield neither trivial (zero progress) nor difficult (zero progress) results [66].

Adversarial Curriculum Generation

In the case of adversarial games, the task is fixed by the opponent's policy π_{opp}. A random opponent defines a weak curriculum, while a perfect opponent (Nash) offers no curriculum (crushing the agent).

PAIRED (Protagonist Antagonist Induced Regret Environment Design) is the main algorithm in this domain. It employs three agents:

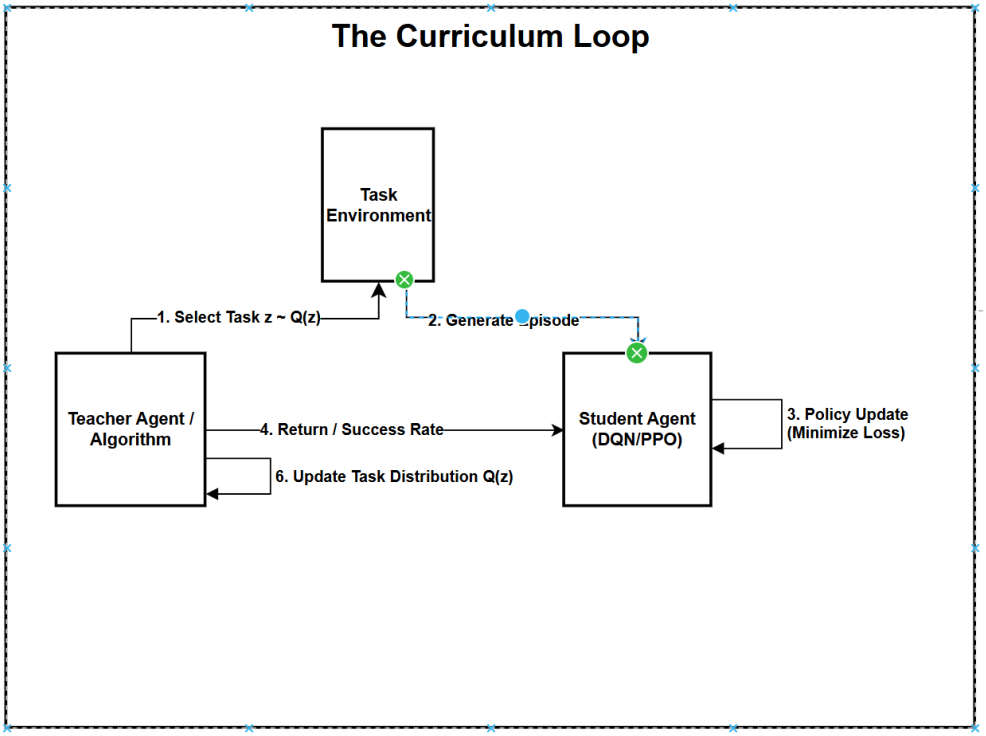


Figure 2.4: The Adversarial Curriculum Learning (ACL) Loop.

- A Protagonist (This is the main agent we want to train).
- An Antagonist (The baseline agent).
- An Adversary (The Teacher/Environment generator).

The Adversary generates environments/scenarios to maximize the Regret of the Protagonist relative to the Antagonist:

$$\text{Regret} = \max_{\pi_{Antagonist}} V(\pi_{Antagonist}, E) - V(\pi_{Protagonist}, E) \tag{2.13}$$

By maximizing regret, the Adversary is forced to create scenarios that are possible to solve (since the Antagonist solves them) but difficult for the Protagonist. This naturally generates a curriculum of gradually increasing complexity, ruling out the generation of impossible tasks [67].

Curriculum in Multi-Agent Reinforcement Learning (MARL)

In MARL, non-stationarity is an uncontrolled curriculum. However, with no regulations, pathological behaviors like cycling (Rock-Paper-Scissors) or forgetting ensue.

League Training and Gated Curricula

Approaches like AlphaStar use a League system that serves as a population-based curriculum. Population-based Curriculum: The agent π_θ is trained on a distribution of opponents;

$$P_{opp} = (1 - \delta) \cdot \mathbb{I}(\text{Current Self}) + \delta \cdot \text{Uniform}(\text{Past Selves}) \tag{2.14}$$

This ensures that the agent does not overfit to the current meta but retains robustness against previous strategies.

Recent work such as EAT-C (Environment-Adversarial Sub-Task Curriculum) [68] decomposes sub-tasks for complex multi-agent tasks. The adversary controls the allocation of these sub-tasks to agents that first master coordination primitives (e.g., formation holding) before mastering high-level strategy. For data efficiency, such a hierarchy is critical as it prunes out the joint action space $\mathcal{A}^N$.

Comparative Analysis of Curriculum Strategies

Table 2.3 synthesizes the trade-offs between different curriculum methodologies in the context of data efficiency.

Table 2.3: Taxonomy of Curriculum Learning Approaches in Adversarial DRL

Methodology	Mechanism	Data Efficiency Gain	Primary Risk
Manual / Linear	Pre-defined schedule (e.g., range increases)	Moderate	Mismatched pacing (Boredom/Anxiety)
Self-Paced	Agent chooses task based on confidence	High	Exploitation of easy tasks (Laziness)
Teacher-Student (ACL)[69]	Teacher maximizes Learning Progress	Very High	Non-stationary teacher dynamics
Adversarial (PAIRED)[67]	Regret Maximization	High (Robustness)	Generation of impossible tasks
Implicit (Self-Play)[70]	Opponent is copy of agent	Variable	Cycling / Strategy Collapse

Critique and Open Challenges

Although CL theoretically improves sample efficiency, it introduces the Catastrophic Forgetting problem. As the curriculum advances to hard tasks ($Q_t \rightarrow$ Hard), the agent rarely sees easy tasks. In neural networks, this changes the weights so that the agent forgets or loses the knowledge to solve basic tasks well. Having a second-best solution to the whole difficulty spectrum often demands Experience Replay with Prioritized Sampling or Distillation [71]. In addition, the computational cost of training a *Teacher* agent (in ACL) sometimes offsets efficiency gains from the *Student*. In this thesis, we do this by using a hybrid heuristic-adversarial curriculum (CGAWM), which diminish the overheads of the teacher mechanism while maintaining the benefits of adaptive pacing.

2.5.3 Meta-Reinforcement Learning

Standard Deep Reinforcement Learning (DRL) algorithms are extremely brittle; a policy trained to dominate a particular opponent or environment configuration will collapse catastrophically when exposed to slight perturbations in transition dynamics or reward functions. This is a huge limitation in adversarial settings as the task changes every time the opponent changes their move. Meta-Reinforcement Learning (Meta-RL) solves the fundamental inefficiency using a different optimization objective – instead of training a policy to solve a single fixed task, the learned learning procedure (set of meta parameters) enables the agent to quickly learn new unseen tasks with minimum sample interactions [72].

This paradigm – often termed learning to learn – is critical for data efficiency in non-stationary environments. By using a prior distribution of tasks $\mathcal{T} \sim p(\mathcal{T})$ the Meta-RL agents learn a schema of the problem domain and can identify the characteristics of a new opponent (aggressive vs. defensive) within a few timesteps (few-shot learning) rather than re-training from scratch [73].

Theoretical Formulation

Each task $\mathcal{T}_i = \langle \mathcal{S}, \mathcal{A}, P_i, R_i, \gamma \rangle$ is a different MDP, formally a distribution of tasks is considered as $p($ In terms of this thesis, different tasks equate to different opponent policies $\pi_{opp}^{(i)}$ or different types of environmental settings (e.g. layouts of obstacles).

The Objective of Meta-RL is to arrive at an optimal set of meta-parameters θ that maximizes the expected return of a task-specific policies ϕ_i based on K adaptation steps (or shots). The adaptation mechanism is also a function f_θ from the task training samples on a support set of size K, denoted by $\mathcal{D}_i^{sup} = \{(s, a, r, s')_{1:K}\}$ to the task-specific parameters ϕ_i:

$$\theta^* = \arg\max_\theta \mathbb{E}\mathcal{T}i \sim p(\mathcal{T}) \left[\mathbb{E}\tau \sim \pi\phi_i[R(\tau)]\right] \quad \text{where} \quad \phi_i = f_\theta(\mathcal{D}_i^{sup}) \tag{2.15}$$

This formula indicates that the core distinction from standard RL: The agent is optimized for its post-adaptation performance on a query set rather than for its immediate performance during training. $\mathcal{D}_i^{qry}$ [74].

Gradient-Based vs. Context-Based Adaptation

The literature primarily divides Meta-RL into two architectural paradigms, each with significant implications for adversarial games.

1. **Gradient-Based Meta-Learning (MAML):** Model-Agnostic Meta-Learning (MAML) [74] improves the initialization of policy network weights θ to get an optimal policy for a new task in a few stages using standard gradient descent.For a task $\mathcal{T}_i$, the modified parameters θ_i' are determined as follows:

$$\theta' i = \theta - \alpha \nabla \theta \mathcal{L}\mathcal{T}i(\theta) \tag{2.16}$$

The meta-objective then minimizes the loss of the adapted parameters across all tasks:

$$\min \theta \sum \mathcal{T} i \sim p(\mathcal{T}) \mathcal{L} \mathcal{T}_i(\theta_i') \tag{2.17}$$

Although theoretically powerful, MAML requires computing second-order derivatives (Hessians), making it computationally expensive and generally unstable in reinforcement learning due to the high variation of policy gradients [73]..

2. **Context-Based Meta-Learning (PEARL):** Context-Based approaches are generally superior in adversarial applications that require real-time adaptability. Algorithms such as PEARL (Probabilistic Embeddings for Actor-Critic RL) [75] consider adaptation an inference problem. Instead of updating weights, the agent uses a Context Encoder $q_\psi(z|c)$ to map a history of interactions $c = \{(s, a, r, s')\}$ into a latent probabilistic variable z (the context).

   The policy is conditioned on this context: $\pi_\theta(a|s, z)$. In an adversarial game, z effectively acts as an Opponent Embedding

   - The history c shows high frequecy collisions, if the opponents plays aggressively.
   - The encoder infers a specific latent z_{agg}.
   - The policy $\pi_\theta(\cdot|\cdot, z_{agg})$ will automatically shift to a defensive position.

   This adaptation will happen via forward pass instantly rather than it requiring iterative updates from the gradient, which will make it highly data efficient[75].

Adversarial Meta-Reinforcement Learning

Standard Meta-RL assumes that test tasks have the same stationary distribution $p(\mathcal{T})$ as training tasks. However, in competitive systems, a competent adversary can intentionally cause a distributional shift to exploit the agent. This requires Adversarial Meta-RL, which embeds robustness targets in the meta-learning loop.

Recent Approaches Frame This as a Minimax Game. The agent (Protagonist) has to maximize rewards. An Adversary (Environment Generator) tries to minimize them by choosing the worst-case parameters for the task. The resultant optimization problem is:

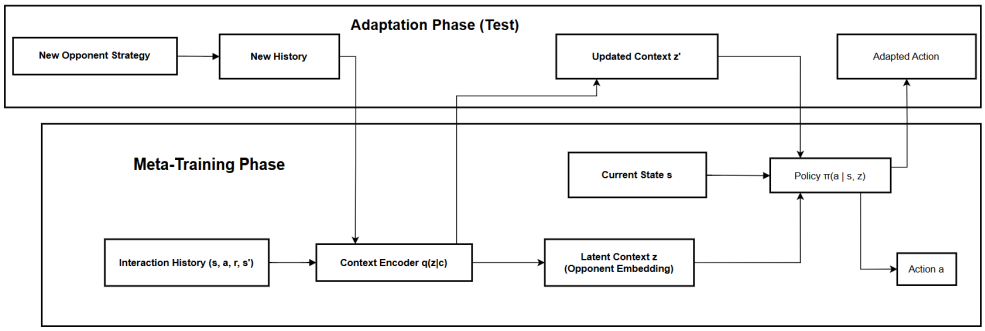


Figure 2.5: Context-Based Meta-Reinforcement Learning Architecture.

$$\max_{\theta} \min_{\mathcal{T} \in \mathcal{U}(\epsilon)} \mathbb{E}[R(\tau|\theta, \mathcal{T})] \tag{2.18}$$

where $\mathcal{U}(\epsilon)$ represents an uncertainty set of tasks (e.g., opponent strategies) within an ϵ-ball of the training distribution. The work by [76] shows that the combination of Model-Based RL with adversarial meta-objectives will allow agents to learn more robust policies that will maintain a good performance when the opponents strategy drift from the training priors. This provides a theoretical justification for this thesis that using curriculum based adversaries are useful in improving generalization.

Comparison of Meta-RL Architectures

Table 2.4 will provide a comparative analysis of leading Meta-RL approaches showing stability for adversarial environment.

Table 2.4: Comparative Analysis of Meta-RL Approaches in Adversarial Domains

Algorithm	**Adaptation Mechanism**	**Inference Speed**	**Robustness to Adversaries**
MAML [74]	Gradient Descent (Few-Shot)	Slow (Iterative)	Low (Overfits to support set)
RL^2 [77]	RNN Hidden State (Memory)	Fast (Recurrent)	Moderate (Limited by context length)
PEARL [75]	Latent Context Encoder	**Very Fast**	**High** (Probabilistic inference)
AdMRL [76]	Minimax Optimization	Moderate	Very High (Explicit robustness)

Relevance to Data Efficiency in Games

The integration of Meta-RL tackles the generalization problem presented in Section 2.4. The integration of Meta-RL into the overall concept directly solves the generalization problem raised, as discussed in Section 2.4 – when the opponent changes the strategy in traditional DRL, the agent has to re-learn effectively the game from scratch using thousands of samples. A Meta-RL agent that has learned the structure of the game dynamics over a distribution of opponent treats a different strategy as a shift in the latent context z. In other words, this changes the sample complexity from that of learning (large) to adaptation (low). The empirical evidence on hard benchmarks such as StarCraft II demonstrates the ability of Meta-RL agents to match human professional strategies in a few games while typical agents take thousands of rounds[78].

2.5.4 Representation Learning

The bottleneck in high-dimensional and partially observable environments in Deep Reinforcement Learning (DRL) is often not the policy optimization itself but the discovery of a latent manifold that captures the essence of causally relevant features in an environment. Representation Learning aims to project raw, high-dimensional input $\mathcal{O}$ (e.g., pixels grid or raw sensor telemetry) onto a compact latent space $\mathcal{Z}$ that retains task-relevant information while discarding nuisance variables [79]. To an AI researcher, this is viewed as a dimensionality reduction problem satisfying the Information Bottleneck principle, where the sufficiency of learned features for predicting the future reward and state transition must be guaranteed.

Theoretical Foundations: The Latent State Objective

First we will define an encoder $f_\phi : \mathcal{O} \to \mathcal{Z}$. The main goal is to learn the parameters ϕ such that the latent state z_t will facilitate the learning of the value function $V(z_t)$ or policy $\pi(a_t|z_t)$. A good representation must have these three properties in the context of data efficiency:

1. Markovian Property: $P(z_{t+1}|z_t, a_t)$ should be independent of $z_{<t}$.
2. Linear Separability: Downstream tasks (reward prediction) should ideally be solvable via a linear transformation of $\mathcal{Z}$.

3. Invariance: z_t should be invariant to visual noise (e.g., changes in background lighting) that does not affect environment dynamics.

Self Supervised and Auxiliary Tasks

To speed up the search for those latent manifolds, Auxiliary Tasks are introduced in modern researches. Apart from the sparse scalar reward signal r_t, the encoder is trained over dense self-supervised objectives.

A key mechanism, Masked Autoencoding (MAE), entails masking a certain percentage of input (e.g., patches of a game frame) and requiring the agent to reconstruct the missing data citekim2023sample. This reconstruction loss, denoted by $\mathcal{L}_{rec}$ is combined with the RL loss $\mathcal{L}_{RL}$:

$$\mathcal{L}total = \mathcal{L}RL + \lambda \underbrace{\mathbb{E}o \sim \mathcal{D}[||\text{Decoder}(f\phi(o_{masked})) - o_{original}||^2]}_{\text{Reconstruction Auxiliary Loss}} \tag{2.19}$$

As in StarCraft II [24], we will achieve a significantly higher sample efficiency by forcing the encoder to learn the spatial correlations of the environment

Relational Representations via Graph Neural Networks (GNNs)

In Multi-Agent Reinforcement Learning (MARL), as the number of agents N grows, the complexity also grows exponentially. Permutation invariance and varying topology is not efficiently represented by the traditional Multi-layer perception(MLP) based representation. A relational inductive bias is provided by the Graph Neural Networks (GNNs) by conceptualizing the system as a graph$\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where the nodes $\mathcal{V}$ are the agents and the edges of the graph $\mathcal{E}$ are spatial or functional relationships[80].

We update the representation of the agent i through message passing layers:

$$h_i^{(l+1)} = \sigma\left(W^{(l)}h_i^{(l)} + \sum_{j\in\mathcal{N}(i)} \text{Message}(h_i^{(l)}, h_j^{(l)})\right) \tag{2.20}$$

This will allow the agent to learn emergent coordination representation. This is where the feature vecotr h_i encodes its own state, the relative intent and the positioning of its team mates and opponents.

Physics-Informed and Symbolic Representations

or agents working in physical simulations, data efficiency can be further enhanced by incorporating physical restrictions into the representation layer. Physics-Informed RL restricts latent space to follow laws such as conservation of momentum or energy. This reduces the search space for policy by a few orders of magnitude, ensuring $P(z_{t+1}|z_t, a_t)$ does not violate physical priors.

Additionally, Neuro-Symbolic representations strive to bridge the gap between connectionist neural networks and symbolic logics. With the visual inputs mapped into symbolic predicates (e.g., InFrontOf(Agent, Adversary)), the agent can do some high-level reasoning which has a much better generalization of map geometries than pixel-based policies.

Comparative Analysis of Representation Techniques

Table 2.5 shows the summary of the impact that different representation learning strategies have on data efficiency and robustness.

Table 2.5: Impact of Representation Learning on MARL Data Efficiency

Method	Key Mechanism	Primary Benefit	Computational Overhead
Autoencoders (AE)[81]	Dimensionality Reduction	Efficient state compression	Low
Contrastive Learning[82]	Maximizing Mutual Information	Robustness to visual noise	Moderate
GNNs [80]	Message Passing	Scalability in MARL	High (per agent)
Masked Models [24]	Local patch reconstruction	Fast visual feature discovery	Moderate
Physics-Informed[83]	Law of motion constraints	Superior Generalization	Low (at inference)

2.5.5 Uncertainty Estimation and Exploration

The curse of dimensionality gives will give undirected exploration mechanisms such as ϵ - greedy of gaussian noise which is useless in high dimensional adversarial environments. These passive strategies need exponentially a good amount of data to cover te state space. The agents must transition to active exploration in order to achieve sample efficiency. This is where data collection is being driven by a formal measure of uncertainty. By measuring the limits of its own knowledge, the agent will navigate strategically to the high information states which will effectively prune the search space and accelerate the discovery of optimal policies[84].

Taxonomy of Uncertainty: Aleatoric vs. Epistemic

For a researcher, the critical distinction lies in the source of the uncertainty. Standard DRL often conflates environmental noise with lack of knowledge, leading to suboptimal exploration. We formally decompose total predictive uncertainty U_{total} into two components [85]:

1. **Aleatoric Uncertainty (Statistical):**

   This shows the inherent stochasticity in the environment like dice roll or the stochastic opponent's move. Even if a huge amount of data is collected it is irreducible.

2. **Epistemic Uncertainty (Systematic):**

   This shows the model's lack of knowledge. It starts in the regions of the state space where the agents have few number of samples. It operates as a primary signal for exploration and is also reducible unlike aleatoric uncertainty.

Given a dataset $\mathcal{D}$ and a probabilistic model $p_\theta(y|x)$, mathematically the decomposition is expressed via Information Theory:

$$\underbrace{I[y, \theta|x, \mathcal{D}]}_{\text{Epistemic (Information Gain)}} = \underbrace{H[\mathbb{E}p(\theta|\mathcal{D})[p(y|x, \theta)]]}_{\text{Total Entropy}} - \underbrace{\mathbb{E}p(\theta|\mathcal{D})[H[p(y|x, \theta)]]}_{\text{Aleatoric Entropy}} \tag{2.21}$$

The agent will prioritizes transitions which resolves the model ambiguity by increasing the information gain $I[y, \theta|x, \mathcal{D}]$, which directly improving world model accuracy and sample efficiency [86].

Ensemble-Based Uncertainty Quantification

In deep RL, the calculation of the exact posterior $p(\theta|\mathcal{D})$ is considered intractable. Bayesian uncertainty has the frequentist approximation as Deep Ensembles (SOTA) [85]. This is achieved by training an ensemble of M dynamics models $\{\hat{f}_{\theta_1}, \ldots, \hat{f}_{\theta_M}\}$ with varying initializations. The epistemic uncertainty at state-action pair (s, a) is given by the variance (disagreement) of the ensemble members:

$$\sigma^2_{epi}(s,a) = \frac{1}{M}\sum_{i=1}^{M} ||\hat{f}\theta_i(s,a) - \mu(s,a)||^2, \quad \mu(s,a) = \frac{1}{M}\sum i = 1^M \hat{f}_{\theta_i}(s,a) \tag{2.22}$$

This variance is used to build an Intrinsic Reward $r_i = \lambda\sigma^2_{epi}$. The total reward $R = r_e + r_i$ encourages *Optimism in the Face of Uncertainty* (OFU). This ensures the agent does not just exploit known high-reward paths but investigates uncertain regions that might be containing even higher rewards or critical transition dynamics [87].

Uncertainty in Multi-Agent and Adversarial Systems

Appying uncertainty estimation to MARL results a Non-Stationarity Dilemma. In this environments, an agent faces collective unceratainty [86]. The transition dynamics seem to change because of other agents learning and adapting which will not be seen in single agent RL.

A sophisticated AI researcher must decouple these sources:

- **Environment Uncertainty (U_{env}):** This is uncertainty in regard to the physics/rules of the game.
- **Opponent Uncertainty (U_{opp}):** This shows uncertainty in regard to the intent and future policy of the adversary.

Recent work in robust MARL [88] suggests agents should be Optimistic regarding U_{env} to drive exploration, but likewise Pessimistic regarding U_{opp} to ensure safety against adversarial exploitation. Dual Uncertainty" prevents an agent from being certain with its counter-stratagems while providing a high discovery rate for the environment.

Reward Model Uncertainty and Preference-Based RL

In scenarios such as sparse reward functions or human/adversarial preferences, the agent is required to learn a reward model, $\hat{R}_\psi(s,a)$. This approach, known as Preference-Based Exploration, then allows the agent to actively query or investigate states where the reward is the most ambiguous [89]. This is important in multi-agent settings where the definition of a win changes depending on the opponents or the curriculum's difficulty.

Comparative Summary of Exploration Mechanisms

The table below outlines the trade-offs of modern exploration methods in terms of data efficiency and adversarial MARL stability.

Table 2.6: Comparison of Uncertainty-Driven Exploration Strategies

Strategy	Mechanism	Data Efficiency	Adversarial Robustness
ϵ-Greedy	Random action selection	Very Low	Poor (Easily exploited)
Entropy Reg.	Policy smoothing	Low	Moderate
RND [28]	Random Net Distillation	High	Low (Noisy TV Problem)
Ensemble Var. [85]	Model Disagreement	**High**	**High** (Filters noise)
VIME [90]	Variational Info. Max.	Very High	High (Theoretically grounded)

Critique: The *Noisy TV* Problem in Adversarial Settings

Noisy TV The problem is a critical challenge for uncertainty-driven agents. If an agent is driven purely by prediction error (epistemic uncertainty), it can get obsessed with unpredictable stochastic elements irrelevant to the reward; like a noisy background or a truly random opponent [28]. This wastes the sample complexity on irrelevant data. The successful implementation in this thesis requires Information Bottleneck constraints or Inverse Dynamics Models where only controllable uncertainty is relevant to filter out information noise caused by irrelevant adversarial actions.

2.6 Research Gap

Despite the considerable amount of literature on reinforcement learning efficiency and multi-agent stability, a comprehensive review shows that both research have traditionally been pursued separately, resulting in a fragmented landscape of solutions. Current techniques effectively address specific aspects of the problem while leaving others, resulting in various trade-offs for practitioners.

By producing progressively difficult adversaries that prevent policy collapse, adversarial curriculum methods like PAIRED[67] and EAT-C [68] have shown a successfully solved the problem of non stationarity. Despite the success, these methods use a model free learning mechanism to converge a requirement. This necessitates a millions of interactions with

the environment which is impractical for real world defense application with limited interaction budgets. In contrast, some Model-Based RL frameworks like DreamerV3[26] and MAMBRL[60] have achieved improved sample efficiency by training with models from the world of the learned world. These frameworks see the environment to be a stationary dynamic system, resulting in failing to address for strategic adaptation of an adversary. Consequently, model based agents often have a brittle policy failing against deceptive opponents.

Some other methods, such as Meta-RL [74] and Curriculum Learning[64], gives a partial solution but introducing new issues. Meta-RL gives a rapid adaptation to new tasks but suffers from an excessive computational cost adaptation during the training phase. Standard curriculum learning improves the convergence speed but also doesn't have a mechanism to create a synthetic training data, which limits the impact on raw sample efficiency.

Therefore, there is an insufficient research that effectively integrates sample efficiency of world models with robust mechanisms of adversarial curricula. Below is an illustrated table (Table 2.7) which shows the current state-of-art approaches forcing a choice between a sample efficiency agent with unstable opponent strategies, or a robust agent that is computationally expensive to train. This thesis aims to bridge this specific gap. .

Table 2.7: Comparative Analysis of Research Gaps in Related Methodologies

Methodology	Sample Eff.	Adv. Robustness	Non-Stationarity	Compute Cost	Representative Papers
Standard MARL (CTDE)	Low	Low	Low	Low	MAPPO[91], QMIX [92]
Curriculum Learning (CL)	Medium	Low	Low	Low	CL [64]
Meta-RL	Low	Medium	**High**	**Very High**	MAML [74], PEARL [75]
Model-Based RL (MBRL)	**High**	Low	Low	Medium	DreamerV3 [26], MAMBRL [60]
Adversarial Curriculum	Low	**High**	**High**	High	PAIRED [67], EAT-C [68]
CGAWM (Proposed)	**High**	**High**	**High**	Medium	*(This Thesis)*

Note: *Sample Eff.*: Data efficiency during training. *Adv. Robustness*: Ability to handle intelligent opponents. *Non-Stationarity*: Ability to adapt to changing opponent strategies.

2.7 Related Work

The pursuit of high-performance agents in competitive environments has necessitated advancements across three primary axes: sample efficiency, structural stability via curriculum design, and adversarial robustness. While individual progress in these domains has been substantial, the integration of these paradigms remains a nascent area of research.

Model Based and World Model Approaches

Model-Based Reinforcement Learning (MBRL) has emerged as a robust framework for enhancing sample efficiency by allowing agents to plan within a learned transition dynamics model [16]. By prioritizing synthetic experience over expensive real-world interactions, MBRL addresses the tabula rasa bottleneck inherent in model-free methods.

The Dreamer series, including DreamerV2 [17][2020] and DreamerV3 [26], represents the current state-of-the-art in latent-space predictive modeling. These algorithms utilize a Recurrent State-Space Model (RSSM) to generate imagined rollouts, significantly reducing environment step requirements. While highly effective in single-agent control tasks, the application of latent world models to adversarial domains introduces significant challenges. Specifically, the presence of an active opponent introduces environmental non-stationarity, where the observed transition dynamics are no longer static but depend on the opponent's evolving policy. Early multi-agent adaptations, such as MAMBRL [60], attempted to resolve this by learning joint dynamics . However, these approaches often treat the interaction as a flat optimization problem, lacking mechanisms to stabilize the world model against the rapidly shifting difficulty levels characteristic of competitive play.

Rather than depending solely on expensive real-world interactions, Model-Based Reinforcement Learning (MBRL) enables agents to learn and plan with a learned model of environment dynamics, greatly enhancing sample efficiency. The method functions by first learning a transition function that estimates next states and rewards, and then the agent can use that function to generate synthetic experiences for policy optimization.

Curriculum Learning and Environment Structuring

Curriculum Learning (CL) focuses on the stability of the learning trajectory by incrementally increasing task difficulty[18]. This is particularly vital in sparse-reward environments where agents struggle to acquire an initial learning signal.

Frameworks such as EAT-C [68]and MAESTRO [93] have demonstrated that adversarial curricula can generate easier sub-tasks early in training to densify reward signals . MAESTRO, in particular, enables self-regulating complexity based on learner performance. Despite these gains in stability, a notable limitation persists: most current CL frameworks are built upon model-free foundations. Consequently, their efficiency is bounded by the rate of real-world interactions, lacking the generative capacity of model-based planning to synthesize experience.

Adversarial Robustness and Generalization

Adversarial RL requires agents to maintain performance against opponents that actively counteract goal attainment. Early methods like RARL [94] utilized a disturbance adversary to force policy robustness . Similarly, self-play methodologies—most notably AlphaZero [88] and AlphaStar [10] demonstrated that competing against increasingly sophisticated versions of oneself can drive generalization to expert levels.

However, these successes often come at a prohibitive computational cost, requiring massive parallelization and millions of environment steps. While meta-learning approaches like AdMRL [76] attempt to bridge the adaptation gap by training across diverse adversarial tasks, they often introduce significant algorithmic overhead.

A notable approach in adversarial MARL is the co-evolutionary framework proposed by[95], which scales environmental complexity by framing the curriculum as a zero-sum game between a cooperative team and a generative attacker. While Hill's methodology successfully generates a boundless stream of novel adversarial configurations through procedural environment design, it relies on model-free optimization (MAPPO), which necessitates a high volume of real-world environment interactions to achieve convergence. Furthermore, such co-evolutionary dynamics can be susceptible to training instability if the adversary's progression outpaces the agent's adaptive capacity.

In order to measure the practical effectiveness of these DRL methods, three main properties have to be considered which are inherently desirable in adversarial settings: Sample Efficiency, Stability, and Exploration Coverage. These characteristics can be used to judge whether it's possible to deploy agents in an environment with limited resources. This article delves into how different methods cover these aspects by taking examples of techniques as indicated in Table 2.8.

We first begin with, **Sample Efficiency** which basically means how fast can an agent locate a good policy while keeping the number of its experiments in the environment as low as possible. As can be seen from the positions in Table 2.8, Dreamer and MAMBRL stand out in this character mainly due to their exploitation of learned dynamics models that enable them to create unlimited synthetic training sets. On the other hand, curriculum-based strategies such as EAT-C and MAESTRO include model-free updates which need the agent be in the real world for a long time so that it can get a proper reward signal through the propagation. This means that these methods have considerably lower sample efficiency.

However, a high level of efficiency frequently results in a decrease of stability. The

Table 2.8: Comparison of Related Works with the Proposed CGAWM

Method	**Model-Based**	**Curriculum**	**Adversarial**	**Uncertainty**	**Sample-Eff.**	**Multi-Agent**
Dreamer [17]	✓	×	×	≈ implicit	✓	×
MAMBRL [60]	✓	×	≈ implicit	×	✓	✓
AdMRL [76]	≈	×	✓	×	×	≈
EAT-C [68]	×	✓	✓	×	≈	≈
Co-Evolving [95]	×	✓	✓	×	×	✓
CQM [96]	✓	✓	×	✓	✓	≈
Desired Solution	✓	✓	✓	✓	✓	✓

Note: ✓= Feature present; × = Feature absent; ≈ = Partial/Implicit support.

environment is non-stationary in adversarial settings. A completely model-based method like *Dreamer* is very difficult to implement here since the model used for prediction may not be able to adapt sufficiently fast to a changing opponent resulting in the production hallucinated rollouts that will mislead the policy. Curriculum strategies are good in stability because they prevent the agent from being overwhelmed by hard tasks at the beginning, but they do not have the generative capabilities to be data-efficient.

The third essential characteristic is Uncertainty-Guided Exploration. Random exploration (for example, ϵ-greedy) is not enough in sparse-reward games. Agents that are equipped with uncertainty estimation are able to search in those parts of the state space where the model is unsure, thus performing more efficient exploration, instead of simply moving randomly. For example, Dreamer possesses latent states that can be considered as a form of uncertainty handling whereas explicit uncertainty, guided exploration is almost absent in curriculum based methods such as EAT-C.

A clear trade-off can be seen when combining these assessments. On the one hand, model-based methods give very high sample efficiency but lack stability when put to test in adversarial modes. On the other hand, curriculum methods provide stability and robustness at the cost of data-inefficiency. Hence the requirement for the **Curriculum-Guided Adversarial World Model (CGAWM)** which combines the generative capability of world models with the stabilizing structure of curriculum learning and the accuracy of uncertainty-guided exploration. The idea of such a hybrid approach is to solve the drawbacks of each separate field, thus facilitating robust learning even in data-limited adversarial scenarios.

Chapter 3

Methodology

This chapter explains an in depth research methods and architectural design, which were developed to solve the key problem of sample inefficiency in Deep Reinforcement Learning (DRL) in adversarial environments. We introduce Curriculum-Guided Adversarial World Model (CGAWM), a hybrid method which combines model-based planning, uncertainty-driven exploration, and curriculum learning. The chapter outlines the research design, data acquisition methods, the mathematical description of the our system, the way the algorithm was implemented, and the evaluation metrics used to measure the success of the approach.

3.1 Research Methodology

This chapter delineates the research methodology and the architectural design of the proposed Curriculum-Guided Adversarial World Model (CGAWM). To address the inherent sample inefficiency of Deep Reinforcement Learning (DRL) in non-stationary adversarial environments, this research adopts the Design Science Research Process (DSRP) model as defined by [97].

Following the DSRP framework, the study is structured into six iterative phases: problem identification, definition of objectives, design and development, demonstration, evaluation, and communication. This chapter specifically details the systematic reformulation of the problem and provides a rigorous technical depiction of the CGAWM modules, including the Adversarial Dynamics Model (ADeM), epistemic uncertainty quantification, and the curriculum-driven task scheduler. The chapter concludes with the evaluation methodology designed to validate the proposed artifact against established benchmarks.

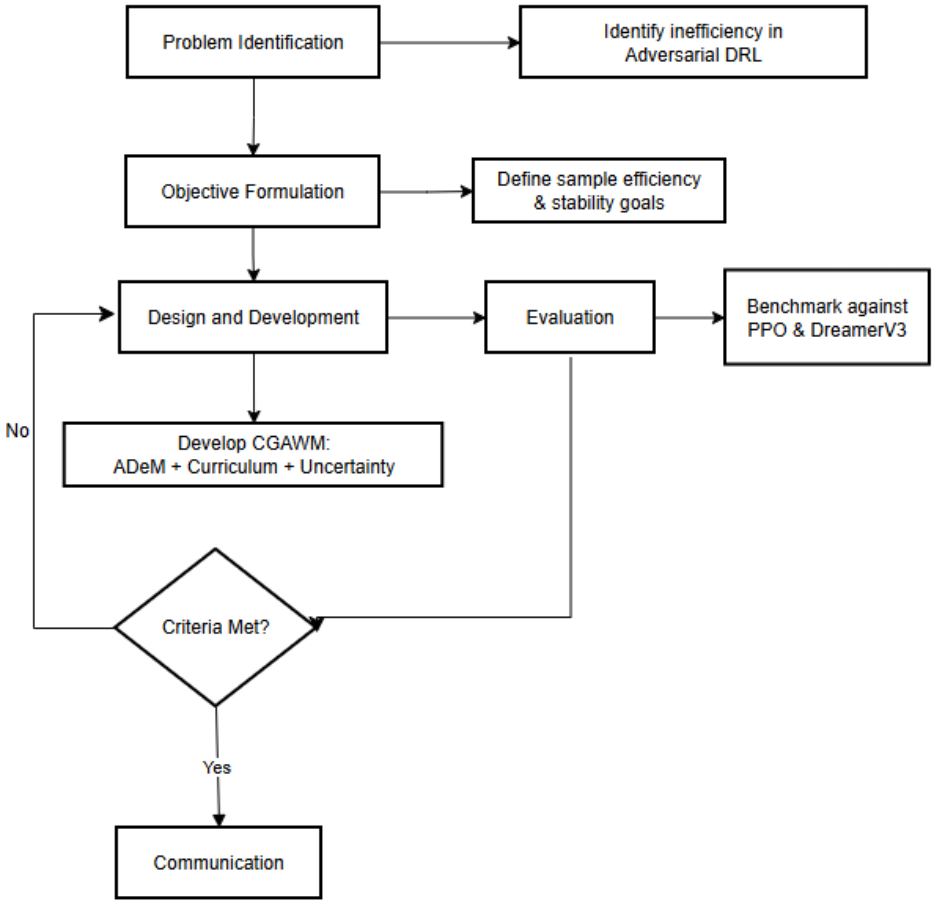


Figure 3.1: Research Design based on the DSRP Model

Problem Identification

In order to determine the research problem, we did research on Deep Reinforcement Learning (DRL) models in adversarial environments and narrowly focused on the analysis of a trade-off among sample efficiency, training stability, and generalization. We also thoroughly inspect the current state-of-the-art algorithms to understand their limitations in competitive settings. Subsequently, the findings of our analysis served as a basis for designing an improved data-efficient solution which is also more robust.

We looked into the pioneering research in Model-Free RL (e. g. PPO and DQN) as well as Model-Based RL (MBRL). The study of these works made it clear that there were still unresolved problems such as the very high sample complexity of model-free methods which

is a large impediment when dealing with non-stationary opponents, and, the vulnerability of classical model-based methods to model bias where the prediction errors get accumulated over time. These results became the basis of a thorough investigation of the main causes of inefficiency, specifically, the lack of informative exploration in sparse reward landscapes and the destabilization of learning gradients when opponent strategies drift.

In fact, we came up with the three major research problems through this evaluation:(1) the issue of relying on the real environment itself should be eliminated, by building a robust adversary dynamics model, that can generate synthetic experiences significantly reducing the dependence on real, world interactions, (2) the importance of an uncertainty guided exploration mechanism to effective solve high-dimensional state spaces where there is a sparse reward problem, and (3) the necessity for a curriculum based learning structure to regulate opponent difficulty ans stabilize convergence. These issues were our primary concern that laid the foundation for our work, and we built the Curriculum-Guided Adversarial World Model (CGAWM) around it as our major contribution to the field.

Objective Formulation

After problem identification, the main research question regarding the data inefficiency of deep reinforcement learning (DRL) in adversarial environments was formulated as illustrated in Section 1.3. We set a series of clear objectives to answer this question, starting from a broad aim of improving sample efficiency and robustness of training deep reinforcement learning agents in the presence of non-stationary opponents. Later, we came up with research questions (RQ1-RQ3) that are intended to help in the construction of the Curriculum-Guided Adversarial World Model (CGAWM). These goals directly focus on lessening the reliance on real-world interactions by generating synthetic data (RQ1), easing training instability through curriculum learning (RQ2), and increasing the coverage of the environment's features through exploration based on uncertainty (RQ3).

This structured road-map ensured that every research stage was focused on the central themes of data efficiency improvement and robustness of convergence.

Design and Development

This phase constitutes the contribution of the thesis. We have designed CGAWM based on our objectives. This hybrid architecture integrates three components around a standard Actor-Critic core:

1. An Adversarial Dynamics Model (ADeM) ensemble to model environment transitions

and opponent moves.

2. An Uncertainty-Guided Exploration module to quantify epistemic uncertainty and incentivize information gathering.

3. A Curriculum Learning (CL) module to regulate the difficulty of the adversarial task.

The evaluation stage is where we empirically validate the CGAWM artifact. Our framework will be compared with state-of-the-art baselines (for instance, PPO) through quantitative metrics like sample efficiency (time to convergence), asymptotic win rate, and exploration entropy.

The last phase focuses on documenting the architectural principles, algorithmic innovations, and experimental results. This thesis is the main form of communication, with potential supplement through peer-reviewed conference publication(s).

3.2 Data Acquisition

Data acquisition in the case of Deep Reinforcement Learning (DRL) for adversarial games is very different from how it is done in traditional supervised learning. FOr our case, data acquisition is an online and dynamic process where the agent is generating its own training data through interaction with a simulated environment.

3.2.1 Simulation Environment and Configuration

To simulate the complex, changing nature of adversarial gameplay at a high dimensionality, this study relies on the PettingZoo library, a standard Python framework for Multi-Agent Reinforcement Learning (MARL). We have thus, set up the Pursuit-Evasion (Waterworld) and Adversarial Navigation scenarios to be our main test environments.

During closed loop trials, these environments produce data streams whose characteristics are specifically adversarial. In a continuous State Spaces, the environment gives out state vectors $s_t \in \mathbb{R}^n$ that are made up of continuous variables such as agent velocity, lidar sensor readings, and relative opponent positions. The second characteristics is the stochastic transitions, it is the transition probability $P(s'|s, a, \pi_{opp})$ is inherently stochastic due to the unpredictable actions of the opponent policy π_{opp}. In reward sparsity, The environment is set

up to give dense shaping rewards only at the first curriculum stages, thus gradually changing into sparse, binary outcome rewards (Win/Loss) at the later stages in order to challenge the exploration mechanism.

3.2.2 Experience Replay and Data Structure

Data collected during the interaction phase is serialized and stored in a Prioritized Experience Replay (PER) buffer, denoted as $\mathcal{D}$. This buffer serves as the dataset for training both the policy and the world model.

Each data point acquired is a Transition Tuple e_t, formally defined as:

$$e_t = \langle o_t, a_t, r_t^e, o_{t+1}, d_t, \mu_{opp} \rangle$$

Where:

- o_t: The local observation of the agent at time $t.a_t$: The action executed by the agent.
- r_t^e: The extrinsic reward received from the environment.
- o_{t+1}: The resulting observation at $t + 1$.
- d_t: A boolean flag indicating if the episode terminated (Done).
- μ_{opp}: A latent identifier representing the opponent's current strategy/difficulty level, used to condition the world model.

We employ an Asynchronous Data COllection strategy to ensure data efficiency. Multiple parallel environment instances ($N_{envs} = 16$) run concurrently, feeding transitions into the shared buffer $\mathcal{D}$. This stabilizes the neural network approximators' training and decorrelates the data.

3.3 Design and Development

Building the Curriculum-Guided Adversarial World Model (CGAWM) is the main goal of the Design and Development phase. This artifact is a hybrid architecture that aims to separate the cost of real-world interactions from the complexity of the agent's learning sample.

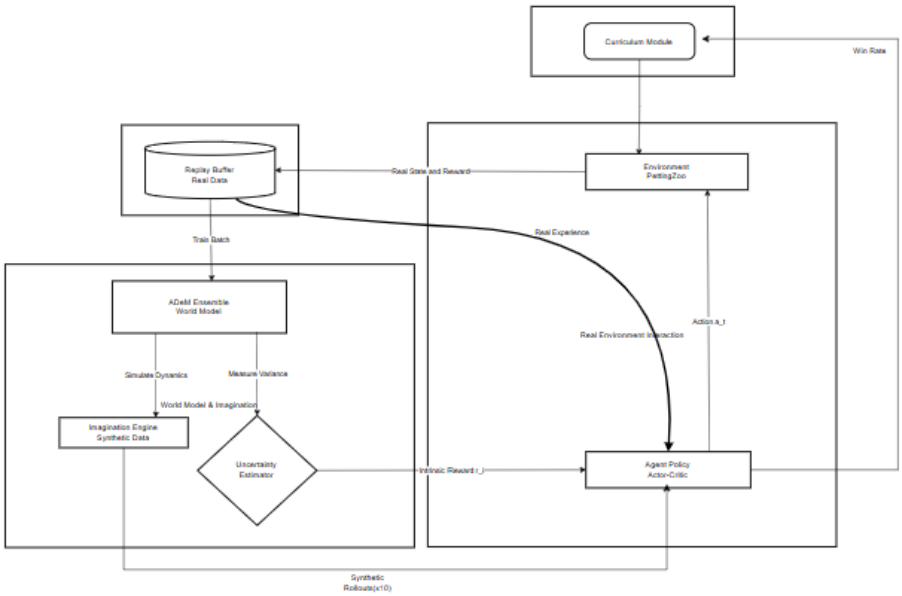


Figure 3.2: System Flow of the Curriculum-Guided Adversarial World Model(CGAWM)

The system has three distinct algorithms operating in collaboration. The Adversarial Dynamics Modeling (ADeM) for syntetic generation of data, Uncertainty-Guided Exploration for gaining maximum amount of information and Adversarial Curriculum Leraning for stabilizing convergence.

Below, we detail the implementation of each method separately, accompanied by its specific algorithmic logic.

3.3.1 Adversarial Dynamics Modeling(ADeM)

The frontline weapon of CGAWM that sets the tone is the Adversarial Dynamics Model (ADeM). This module directly addresses RQ1, sample efficiency. In standard Model-Free Reinforcement Learning (MFRL), the agent needs millions of interactions before it can figure out the consequences of its actions on its own. The ADeM goes a step further and explicitly learns those results, thus the agent can foresee various potential futures and refine its policy with the help of synthetic experience, which means it can improve its learning without having to be in the environment all the time.

Typical World Models (e. g. , the ones employed in Dreamer or Dyna, Q) generally

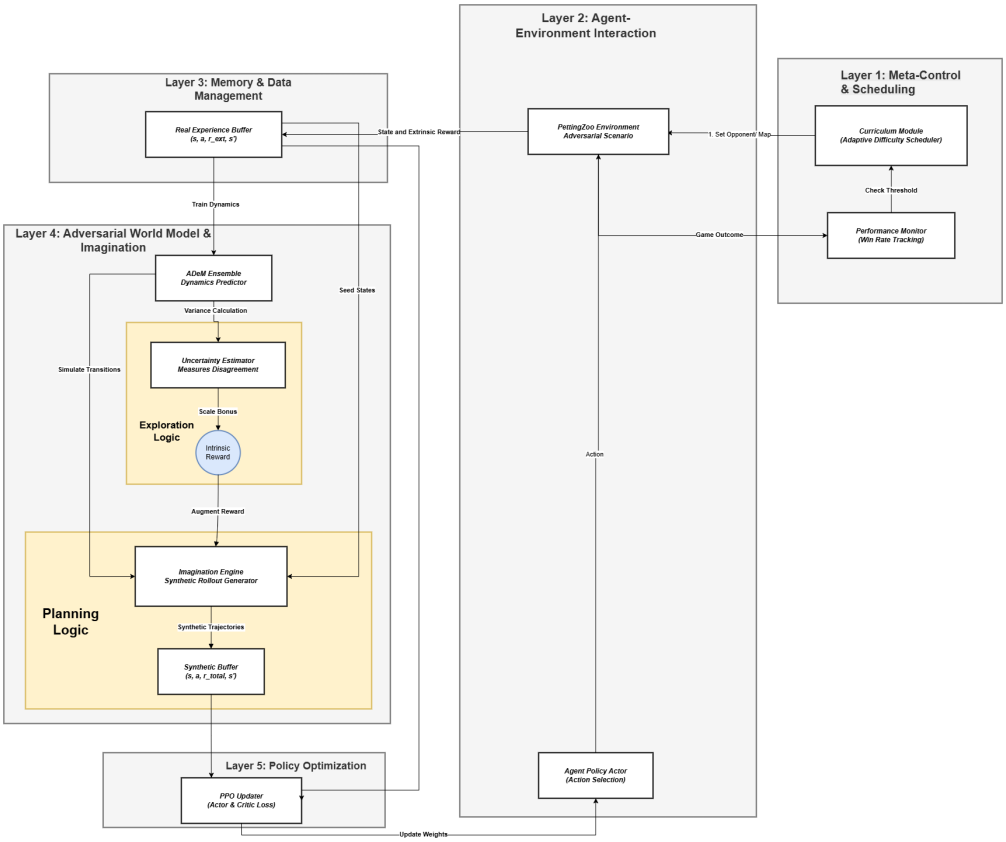


Figure 3.3: Architectural Overview of the Curriculum Guided Adversarial World Model (CGAWM)

assume a static environment where the transition dynamics $P(s'|s, a)$ stay the same. However, in the case of an adversarial scenario, the environment is by nature non-stationary. The environment consists of the opponent whose policy π_{opp} is changing all the time. A distributional shift occurs; it happens when the opponent learns, the transition dynamics change thus a static world model becomes useless.

Moreover, adversarial interactions tend to be multimodal. For example, in a pursuit scenario where the opponent has an option to evade either left or right. A model averaged over such training data would predict the mean result of these two modes (e. g. , the opponent going straight) which would be a physically impossible and non-informative state prediction.

In order to address these difficulties, the ADeM is equipped with two features which are

specific to its architecture. First is the probabilistic ensemble structure. Rather than one network, ADeM comprises an ensemble of M independent neural networks. Because of this, the opponents' multimodal behavior can be captured and epistemic uncertainty (utilized in Method 2) can be quantified. The other one is short horizon imagination which acknowledges that long-term prediction accuracy of chaotic adversarial systems decrease rapidly, so the ADeM is employed for short-horizon rollouts ($H = 10 \dots 20$ steps). Thus, the synthetic data generated is ensured to be of high fidelity and can be effectively used for policy updating.

The ADeM is formally defined as an ensemble of function approximators $\Omega = \{f_{\theta_1}, f_{\theta_2}, \dots, f_{\theta_M}\}$, where each member f_{θ_i} is a deep neural network parameterized by θ_i.

In the input space, each network in the ensemble accepts the concatenation of the current state vector ($s_t \in \mathbb{R}^n$) where the agent's local observation (e.g., velocities, lidar scans), the action vector ($a_t \in \mathbb{R}^m$) is the action proposed by the agent policy and the opponent context ($\mathbf{c}_{opp}$) is a latent vector derived from the last k frames of opponent movement, which helps the model differentiate between different opponent strategies (e.g., aggressive vs. passive).

For the output space, each network outputs a prediction tuple$(\hat{s}_{t+1}, \hat{r}_{t+1})$.The next state prediction is predicted observed at the next timestep. To ensure robustness, the network will predict the difference (delta) between the states rather than the obsolete state $\hat{s}_{t+1} = s_t + \Delta\hat{s}$. The reward prediction ($\hat{r}_{t+1}$) is the predicted extrinsic reward (e.g., +1 for capture, -0.1 for time step).

The ensemble is trained using supervised learning on real transition tuples (s_t, a_t, r_t, s_{t+1}) which are stored in Reply Buffer$\mathcal{D}$. The objective of the reply buffer is to minimize the Mean Squared Error(MSE) of the predictions.The loss function $\mathcal{L}$ for the i-th model is:

$$\mathcal{L}(\theta_i) = \mathbb{E}_{(s,a,r,s')\sim\mathcal{D}}\left[||(s_{t+1} - s_t) - \Delta\hat{s}_{\theta_i}(s_t, a_t)||_2^2 + \lambda(r_t - \hat{r}_{\theta_i}(s_t, a_t))^2\right]$$

Where λ is a weighting term balancing state and reward reconstruction accuracy.

The main target of the ADeM is Data Amplification. After the ensemble has been trained to a significant level of accuracy, it can be used as a simulator for Dyna-style planning. During the imagination phase, we sample a set of real states S_{seed} from the replay buffer (seeding). These ensure that the imagination always starts from valid and relevant configurations of the game. The current Agent Policy π chooses an action from each seed state (branching). A model from the ensemble Ω is randomly chosen to predict the result of that action. This stochastic choice avoids the agent overfitting to any one model's peculiarities. The transition

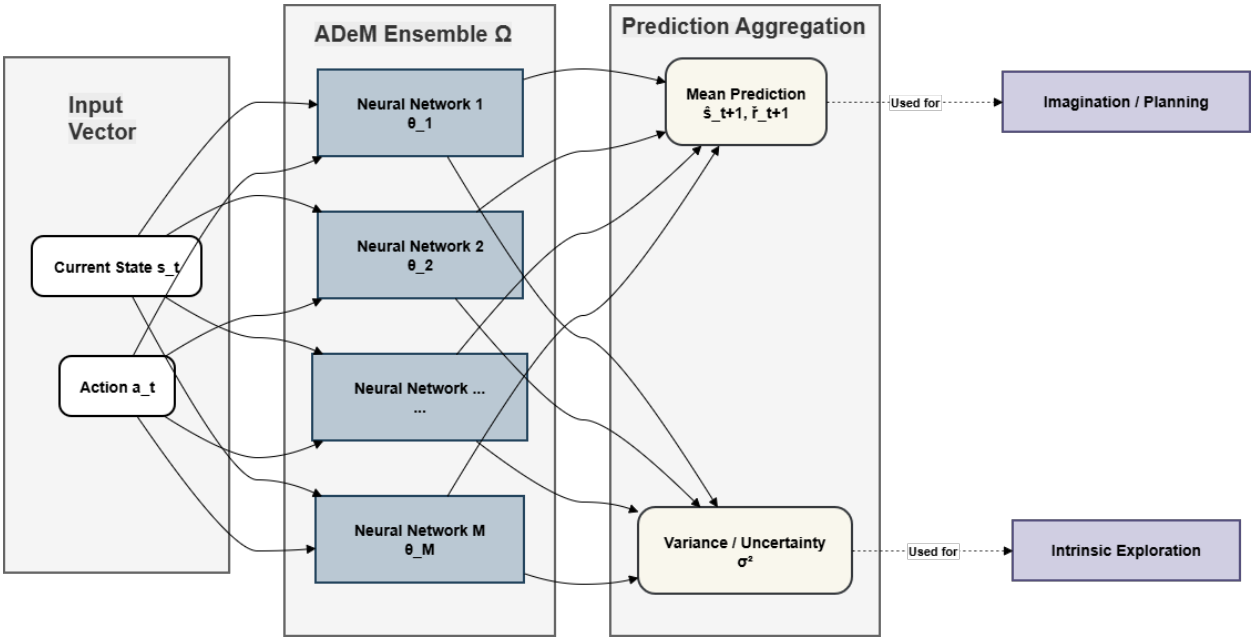


Figure 3.4: ADeM Architecture

produced $(s, a, \hat{r}, \hat{s}')$ is consequently stored in a separate Synthetic Buffer.

The number of synthetic training steps to real environment steps is a tunable hyperparameter, K. We have set $K = 10$ here, implying that if an agent spends one minute playing the game, it will spend ten minutes thinking about the game utilizing the ADeM. This is an effective way to increase sample efficiency by an order of magnitude.

Figure 3.4 shows how the ADeM works step by step. It also separates two phases of the model's life: (1) Learning reality, and (2) Producing synthetic experience.

Algorithm 1 ADeM Training and Synthetic Rollout Generation

Require: Real Replay Buffer $\mathcal{D}_{real}$, Ensemble size M, Rollout Horizon H

Require: Current Policy π_ϕ, Ensemble Parameters $\{\theta_1, \ldots, \theta_M\}$

Ensure: Synthetic Buffer $\mathcal{D}_{imag}$ containing amplified data

Phase 1: Dynamics Learning (Model Training)

for training step $k = 1$ to K_{train} **do**

  Sample batch of transitions $B = \{(s, a, r, s')_j\}_{j=1}^{N}$ from $\mathcal{D}_{real}$

  for each model $m \in \{1, \ldots, M\}$ **do**

    // Each model trains on the full batch or a bootstrap subsample

    Predict state delta: $\Delta\hat{s} = f_{\theta_m}(s, a)$

    Predict reward: $\hat{r} = g_{\theta_m}(s, a)$

    Compute Loss: $\mathcal{L}_m = ||(s' - s) - \Delta\hat{s}||^2 + ||r - \hat{r}||^2$

    Update $\theta_m \leftarrow \theta_m - \alpha \nabla_{\theta_m} \mathcal{L}_m$

  end for

end for

Phase 2: Imagination (Data Amplification)

Initialize $\mathcal{D}_{imag} \leftarrow \emptyset$

Sample set of initial states S_{seed} from $\mathcal{D}_{real}$

for each state $s_0 \in S_{seed}$ **do**

  $s_{curr} \leftarrow s_0$

  for step $h = 1$ to H **do**

    Select action $a_{curr} \sim \pi_\phi(s_{curr})$

    Select random model index $k \sim \text{Uniform}(1, M)$

    Predict next state: $s_{next} = s_{curr} + f_{\theta_k}(s_{curr}, a_{curr})$

    Predict reward: $r_{next} = g_{\theta_k}(s_{curr}, a_{curr})$

    Store transition $(s_{curr}, a_{curr}, r_{next}, s_{next})$ in $\mathcal{D}_{imag}$

    $s_{curr} \leftarrow s_{next}$

  end for

end for

return $\mathcal{D}_{imag}$

3.3.2 Uncertainty Guided Exploration Strategy

The second core methodological component of CGAWM is the Uncertainty-Guided Exploration Strategy that specifically targets Research Question 3 (RQ3). Essentially, this feature plays a vital role in dealing with the sparse reward problem that is naturally part of adversarial games, where random exploration procedures (like ϵ-greedy) hardly come up with the right solutions within a reasonable time[90][98]. Therefore, we quantify model's epistemic uncertainty in depth and thus change the exploration problem from a random process into a more directed, information-seeking goal.

In typical Deep Reinforcement Learning (DRL), exploration is most often taken care of by adding random noise to the policy (e. g. Gaussian noise in continuous control) or by picking random actions with probability ϵ (in discrete spaces). These random methods might still work well in simple and densely rewarding environments (e. g. Atari games) but become disastrous if applied to adversarial scenarios for two reasons. The first reason is reward sparsity. For example, in a pursuit-evasion game, the agent may only get a reward when it captures the opponent. If this photograph scenario consists of a chain of over 100 very precise actions, then the chance of a random agent just stumbling upon it is practically zero ($|\mathcal{A}|^{,100}$).The agent practically gets no feedback for millions of steps resulting in almost non-existent learning curves. The second reason is, as the opponent changes its tactics, parts of the environment which have been previously safe might become risky or even hostile. A completely random agent wouldn't be able to tell the difference between states that are a) known and safe, b) known and dangerous, and c) unknown. It will keep on spending its energy and resources on revisiting the states which it completely understands (low-value) instead of expanding its frontier of knowledge and skills.

In order to address this problem, we turn to the principle of Curiosity Driven Learning. Previously, we have made a formal distinction between two kinds of uncertainty. Aleatoric Uncertainty refers to the situations where the variability of the outcome is caused by the environment itself and is simply a matter of chance(e.g., dice roll). Such uncertainty is not modifiable and therefore should not be taken into consideration when exploring. On the contrary, Epistemic uncertainty indicates that the lack in knowledge or the lack of data concerns only a very specific branch of the state space. This type of uncertainty can be eliminated through interaction.

Our method only targets the maximization of Epistemic Uncertainty. We assume that if the agent is rewarded for going to the states where the Adversarial Dynamics Model (ADeM) shows high epistemic error, it will eventually come up with new strategies and figure out the drawbacks of the opponent, thus the high quality data that helps in resolving that uncertainty will be generated.

Uncertainty Quantification via Ensemble Discrepancy

We measure epistemic uncertainty with a metric called Ensemble Discrepancy. As per the description in Method 1, ADeM is an ensemble of M independent probabilistic neural networks $\{\theta_1, \ldots, \theta_M\}$, each of which has been trained on examples of the experience replay buffer different (via bootstrapping). Given one state, action pair (s_t, a_t), for each model m in the ensemble the predicted next state transition $\hat{s}_{t+1}^{(m)}$ is the output. If the agent is in a familiar area of the state space (e.g., the center of the map where it spends most of its time), all M models would have produced similar predictions. On the other hand, if the agent goes into a new region (e.g., a corner case or a new opponent strategy), the models which were initialized with different weights and have seen different data will generalize differently, thus resulting in high disagreement.

We define the uncertainty scalar u_t as the trace of the covariance of the ensemble predictions:

$$u_t(s_t, a_t) = \frac{1}{M-1} \sum_{m=1}^{M} ||\hat{s}_{t+1}^{(m)} - \bar{s}_{t+1}||_2^2$$

Where $\bar{s}_{t+1}$ is the empirical mean of the ensemble predictions:

$$\bar{s}_{t+1} = \frac{1}{M} \sum_{m=1}^{M} \hat{s}_{t+1}^{(m)}$$

This metric u_t serves as a proxy for information value. A high u_t indicates that the internal world model is confused about the outcome of an action, making that transition highly valuable for training.

We use Reward Augmentation to operationalize this uncertainty metric by including it directly into the policy optimization goal. We view uncertainty as an intrinsic reward signal (r_t^i) rather than just a diagnostic tool.

The extrinsic game reward (r_t^e) and the intrinsic exploration bonus are combined linearly

to provide the total reward r_t^{total} that is sent to the PPO updater:

$$r_t^{total} = r_t^e + \beta_t \cdot \text{clip}(u_t, 0, \eta)$$

where:

- r_t^e *(Extrinsic Reward)*: The sparse binary signal from the game environment (e.g., +1 for win, 0 otherwise).
- u_t *(Uncertainty)*: The ensemble variance calculated above.
- η *(Clipping Threshold)*: A stability constant to prevent exploding rewards in the early stages of training when the model is uniformly ignorant.
- β_t *(Exploration Temperature)*: A decay coefficient that anneals over time.

Initially, β decay isset high, allowing for complete exploration. The agent disregards the game goal and acts like a scientist, doing experiments to get the model error as low as possible. Later on, when the ADeM is accurate, β decays to zero ($\beta_t = \beta_0 \cdot \lambda^t$), so the agent's focus changes smoothly from exploration to exploitation. It is a way to get rid of the noisy TV problem, where an agent might be stuck watching inherently unpredictable noise (aleatoric uncertainty) forever.

The Uncertainty Estimator is not a separate neural network but a meta computational module that sits on top of the ADeM Ensemble. It forms a link between the World Model and the Policy Optimization layer

The architecture processes data in a specific pipeline:

1. Parallel Forward Pass: The state-action pair (s, a) is broadcast to all M networks in the ADeM.
2. Manifold Projection: The high-dimensional state predictions are often projected into a lower-dimensional latent space to make variance calculation computationally efficient.
3. Statistical Aggregation:The module computes the first moment (mean) and second moment (variance) of the output distribution.
4. Signal Injection: The calculated variance is scaled and injected into the reward stream before it reaches the replay buffer.

This technique basically changes the composition of the Replay Buffer $\mathcal{D}$. In standard RL, the buffer is mostly filled with empty transitions where nothing happens (because of sparse rewards). In CGAWM, the buffer gets filled with changing transitions instead.

Since the policy tries to get high u_t, it is actively moving the system to the limits of the ADeM's knowledge. When such transitions are kept and later used to train the ADeM (in Method 1), they give the model the strongest possible gradient signal. This leads to a Positive Feedback Loop: Policy discovers uncertain states, buffer stores uncertain states, ADeM trains on uncertain states and fixes its mistakes. Uncertainty in that area goes down, then policy gets bored and looks for new uncertain states. The cycle keeps the agent going until it eventually maps the whole state space, which answers RQ3.

The Uncertainty-Guided Exploration method is documented in Algorithm 3. This method is called at each timestep throughout the data collection phase.

Algorithm 2 Uncertainty-Guided Exploration Strategy

Require: Current Observation o_t, Proposed Action a_t, Extrinsic Reward r_t^e

Require: Trained ADeM Ensemble $\Omega = \{f_{\theta_1}, \ldots, f_{\theta_M}\}$

Require: Exploration Temperature β, Clipping Threshold η

Ensure: Total Reward r_t^{total}, Intrinsic Signal r_t^i

Step 1: Ensemble Prediction

Initialize prediction tensor $\mathbf{P} \leftarrow \text{zeros}(M, \dim(S))$

for model $m = 1$ to M **do**

  $\hat{s}_{t+1}^{(m)} \leftarrow f_{\theta_m}(o_t, a_t)$ {Predict next state}

  Store $\hat{s}_{t+1}^{(m)}$ in $\mathbf{P}[m]$

end for

Step 2: Calculate Epistemic Disagreement

Compute Ensemble Mean: $\mu_s = \frac{1}{M} \sum_{m=1}^{M} \mathbf{P}[m]$

Compute Pairwise Euclidean Variance:

$u_t \leftarrow \frac{1}{M} \sum_{m=1}^{M} ||\mathbf{P}[m] - \mu_s||_2^2$

Step 3: Compute Intrinsic Reward

// Normalize uncertainty to stabilize training scale

$u_{norm} \leftarrow \frac{u_t - \mu_{history}}{\sigma_{history} + \epsilon}$

$r_t^i \leftarrow \beta \cdot \text{clip}(u_{norm}, 0, \eta)$

Step 4: Reward Augmentation

$r_t^{total} \leftarrow r_t^e + r_t^i$

return r_t^{total}, r_t^i

3.3.3 Adversarial Curriculum Learning (ACL)

The third and final methodological pillar of CGAWM is Adversarial Curriculum Learning (ACL), which is the component that address Research Question 2 (RQ2). While the combination of the Adversarial Dynamics Model (Method 1) and Uncertainty-Guided Exploration (Method 2) respectively, achieves sample-efficiency and data diversity, ACL, is the one that provides the training stability that is needed.

One of the main reasons for learning failure in adversarial settings is the non-stationarity of the environment. As the opponent continues to update its tactics, the transition dynamics

$P(s'|s,a)$ keep changing so that the agent is basically presented with a new optimization landscape every time. If the agent is not given a well, structured curriculum, the agent often faces an adversary that is far too strong, which will result in the agent obtaining almost no rewards and the gradients disappearing, a phenomenon that is referred to as the sparse reward problem.

Our curriculum has been planned around the pedagogical idea of the Zone of Proximal Development (ZPD). For example, in Deep Reinforcement Learning (DRL), it means an agent is able to learn best when the challenge is just right for the agent's capability. We depict learning as a chain of tasks $\mathcal{T} = \{T_0, T_1, \ldots, T_N\}$, in the order of their difficulty. Assume π_θ represents the policy of the agent and π_{opp} denotes the opponent's policy. The expected reward $J(\pi_\theta, \pi_{opp})$ largely depends on the difference in the level of the two agents. If $\pi_{opp} \gg \pi_\theta$ (Opponent is too strong): The agent is defeated regularly ($J \approx 0$), thus no gradient signal is available for distinguishing good actions from bad ones. If $\pi_{opp} \ll \pi_\theta$ (Opponent is too weak): The agent wins effortlessly ($J \approx 1$) but ends up with fragile strategies that do not generalize.

The ACL module is a meta controller that adjusts the task T_k to keep the agent in a Goldilocks zone where the win rate is roughly $0.5 \leq WR \leq 0.8$. This guarantees a dense, informative reward signal that stabilizes the gradient descent process. To discipline difficulty in a systematic manner, we describe the curriculum state C_k as a parameter bundle managing two different axes: Opponent Competence and Environmental Complexity.

Opponent competence (ρ_k) is the chief factor of adversarial difficulty. We keep a League of Opponents, a group of our old policy points. In stage 0 (random) π_{opp} selects actions uniformly from $\mathcal{A}$. The agent is thus enabled to learn basic locomotion and game mechanics. In Stage 1 (Heuristic) π_{opp} adheres to a fixed, rule-based strategy (e.g., always move toward agent). This helps to acquire the skills of reaction and evasion. In Stage 2 (Weak self play), π_{opp} is a saved snapshot of the agent's policy from an earlier training epoch ($\theta_{t,k}$). After Stage 3 (Strong self play), π_{opp} is a recent snapshot ($\theta_{t,\epsilon}$), thus continuously creating an evolving zero-sum game.

Environmental Complexity (ξ_k): This dimension varies the physical properties of the simulation in such a way that the exploration burden is reduced. Map Size is initially small (16×16) to increase the interaction frequency and then is extended to the full size (64×64). Entity Density is the number of obstacles or secondary agents which is slowly escalated. And

horizon is the maximum episode length T_{max} which is gradually *extending*, thus the agent is forced to learn consistently long-horizon planning. The curriculum level C_k at any time t characterizes the active configuration: $C_k = \langle \rho_k, \xi_k \rangle$.

Instead of time-based static curricula which change according to fixed time steps (e. g.,increase difficulty every 1M steps), CGAWM adopts a Performance, Based Gating Mechanism. This way, the agent must be really ready for the next level, and the forgetting of the previous one is avoided. We introduce a Win Rate Monitor responsible for recording the win/loss ratio of the last W episodes ($W = 100$). The smoothed win rate $\bar{R}_{win}$ is given by the formula:

$$\bar{R}_{win} = \frac{1}{W} \sum_{j=1}^{W} \mathbb{I}(\text{outcome}_j = \text{WIN})$$

The transition logic is regulated by a threshold $\tau_{advance}$ (usually 0.80) such that: If $\bar{R}_{win} \geq \tau_{advance}$: The agent has successfully learned the current level. Increment $k \to k + 1$. If $\bar{R}_{win} < \tau_{regress}$: (Optional) If performance drops drastically below the safety threshold (e. g., 0.20), decrement $k \to k, 1$ for a recovery phase. Otherwise:Keep the current level C_k. The above method aims at RQ2 (Stability) as it makes sure the learning signal's variance is controlled. The Curriculum Module is like a higher level controller over the standard training loop. It has read, access to the training metrics and write, access to the environment configuration. In the integration flow, the module subscribes to the EpisodeDone signal from the environment. It updates the rolling win, rate statistics. Before each new episode reset, it sets the environment parameters to those of the current level C_k.

The operational logic of the Adversarial Curriculum is formally outlined in Algorithm 4. This algorithm is run asynchronously or at the end of every training epoch to modify global difficulty settings.

Algorithm 3 Dynamic Adversarial Curriculum Update

Require: Performance Buffer B_{perf} containing last W episode outcomes
Require: Current Difficulty Level k, Max Level K_{max}
Require: Curriculum Schedule $\mathcal{S} = \{(\rho_0, \xi_0), \ldots, (\rho_{K_{max}}, \xi_{K_{max}})\}$
Require: Thresholds $\tau_{adv} = 0.8$, $\tau_{reg} = 0.2$
Ensure: Updated Environment Configuration C_{env}

Step 1: Compute Performance Metrics
$N_{wins} \leftarrow \sum_{i=1}^{W} \mathbb{I}(B_{perf}[i] == \text{WIN})$
$WinRate \leftarrow N_{wins}/W$
Step 2: Evaluate Gating Condition
if $WinRate \geq \tau_{adv}$ **and** $k < K_{max}$ **then**
    // Promotion: Agent has mastered current level
    $k \leftarrow k + 1$
    Log: "Promoted to Level k. WinRate: $WinRate$"
    Clear B_{perf} {Reset stats for new level}
else if $WinRate \leq \tau_{reg}$ **and** $k > 0$ **then**
    // Demotion: Agent is failing, reduce difficulty
    $k \leftarrow k - 1$
    Log: "Demoted to Level k. Stability recovery mode."
    Clear B_{perf}
else
    // Maintenance: Continue training at current level
    $k \leftarrow k$
end if
Step 3: Apply Configuration
$(\rho_{new}, \xi_{new}) \leftarrow \mathcal{S}[k]$
C_{env}.OpponentPolicy $\leftarrow$ LoadPolicy(ρ_{new})
C_{env}.MapConfig $\leftarrow$ SetParams(ξ_{new})
return C_{env}

The Curriculum Module doesn't work in isolation; instead, it serves as a force multiplier for Methods 1 and 2:

Synergy with ADeM (Method 1): When you start with the simplest dynamics (Stage 0), ADeM can quickly learn a highly accurate world model (very low prediction error). Then, as the curriculum moves on, ADeM gradually adjusts to more complex dynamics, thus, avoiding the issue where a model tries to learn everything at once.

Synergy with Uncertainty (Method 2): At the beginning of the curriculum, the rewards are dense, so the exploration driven by uncertainty is not very important. However, when the curriculum reaches the hardest stages (sparse rewards), the intrinsic uncertainty signal naturally becomes the main learning driver, thus, it takes over perfectly when extrinsic rewards are no longer present. This comprehensive integration guarantees that CGAWM is capable of enduring the entire training life-cycle.

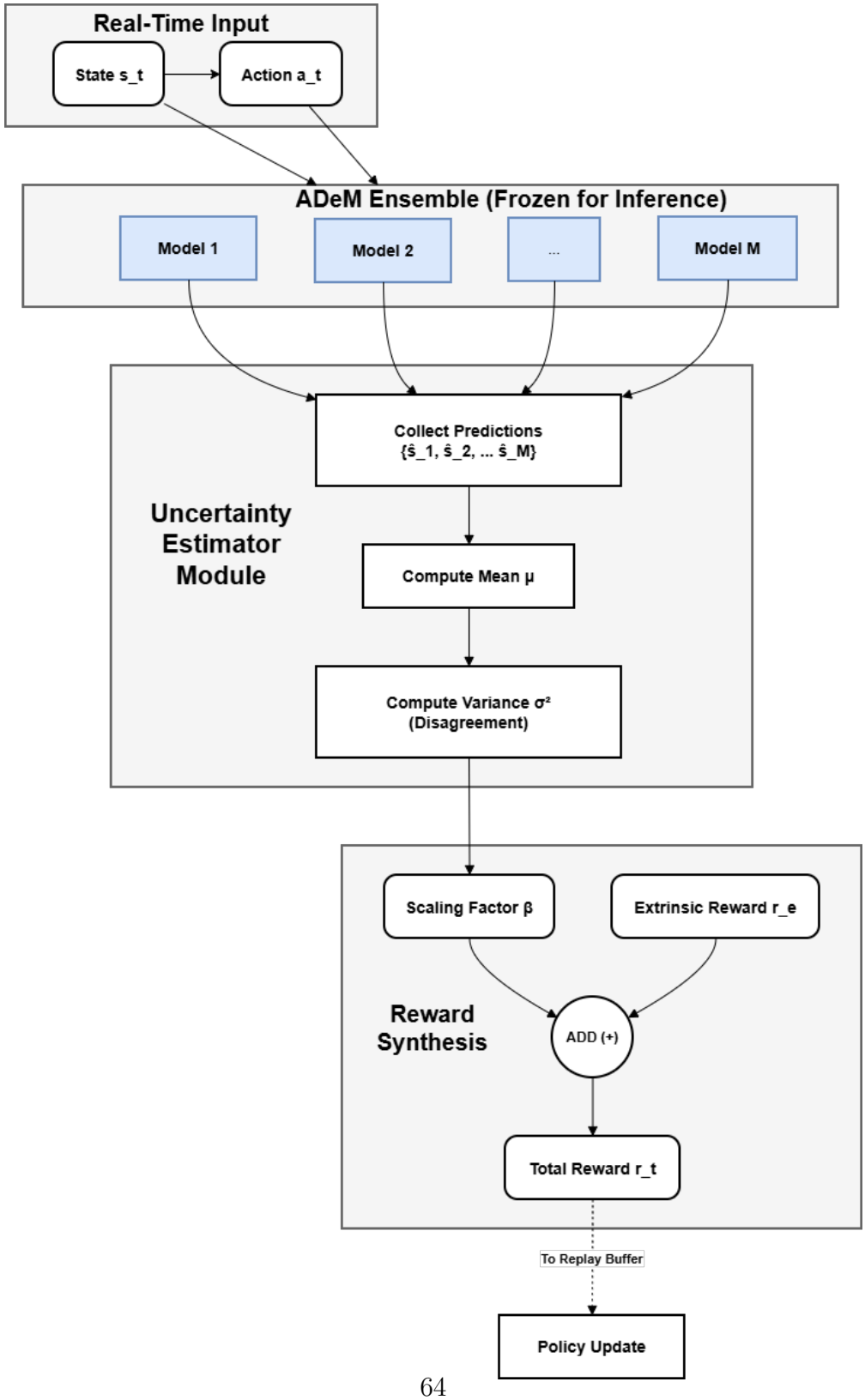


Figure 3.5: Uncertainty Architecture

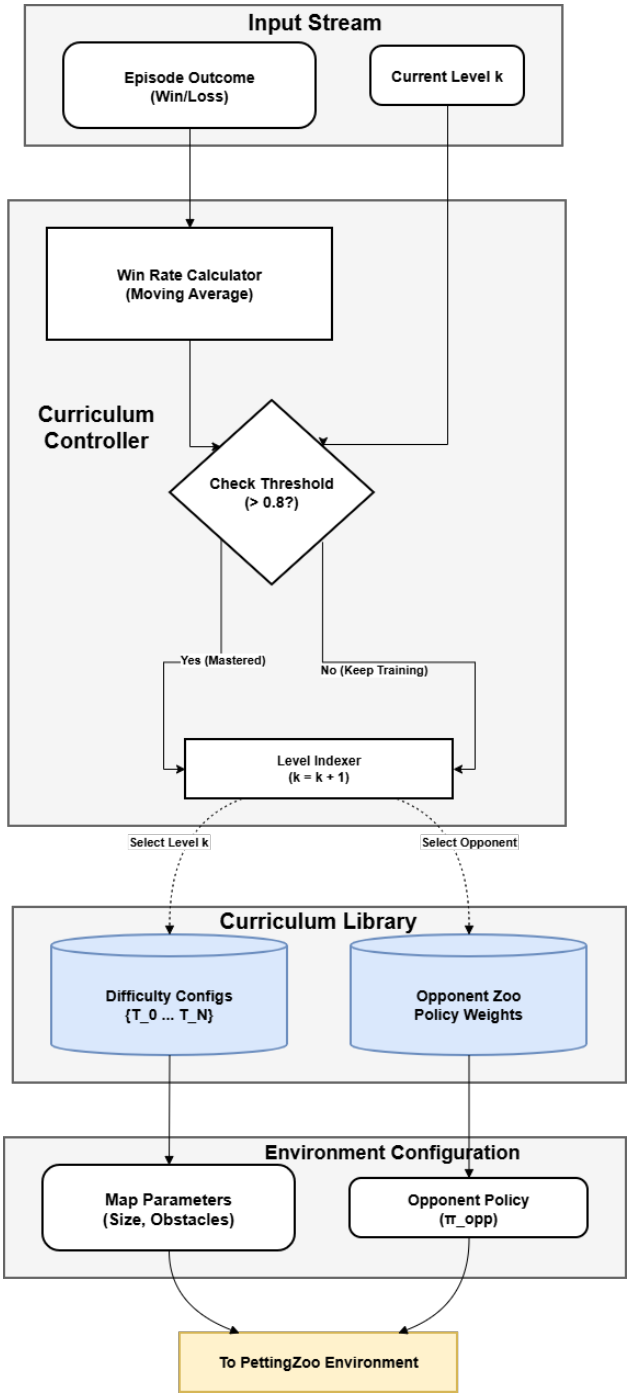


Figure 3.6: Architectural of Dynamic Adversarial Curriculum

Chapter 4

Experimentation

This chapter describes the experimental evaluation of the proposed Curriculum-Guided Adversarial World Model (CGAWM) . It mainly aims to demonstrate through experiments that the hybrid architecture is effective in alleviating the two major problems, sample inefficiency and instability, that exist in adversarial Deep Reinforcement Learning (DRL). We offer a thorough review of the experiments, ablation studies that measure supporting evidence of the various elements (Adversarial Dynamics Model, Curriculum Learning, and Uncertainty-Guided Exploration), and the method of the results. In addition, agent behavior's qualitative evaluation, paper contributions discussion, and limitations have been used as background to highlight this research's contributions.

4.1 Experimental Setup

This section gives an overview of the experimental setup used to test the proposed Curriculum-Guided Adversarial World Model (CGAWM). We go into detail about the simulation environments, the main algorithms used as baselines for ablation, the exact hyperparameter settings obtained from initial grid searches, and the hardware used for training to be consistent, set a rigorous benchmark, and make our experiments reproducible. The experimental setup is controlled by the need to identify the isolated effects of the three main methodological components: the Adversarial Dynamics Model (ADeM), the Uncertainty-Guided Exploration mechanism, and the Adversarial Curriculum Learning (ACL) module.

4.1.1 Simulation Environment: PettingZoo MPE

We tested the CGAWM in an environment that required adversarial adaptation, using the Multi-Particle Environments (MPE) suite, in particular, the `simple_adversary_v3 scenario` scenario, which is available in the PettingZoo library . We picked this environment as our main testbed since it encapsulates the three main problems we formulated, i.e., non-stationary, partial observability, and sparse rewards.

The `simple_adversary` scenario is a model of a competitive, zero-sum game with N = 3 agents in a continuous 2D space along with L = 2 landmarks. The Good Agents ($N_g = 2$) are those agents that can see the positions of all agents and landmarks. More importantly, they are aware of which of the two landmarks is the Target Landmark. Their goal is to get to the target landmark while deceiving the adversary at the same time. The Adversary Agent (Nadv=1) is the adversary who also sees the positions of all agents and landmarks but does not know which one is the target. It aims to identify the target through the behavior of the Good Agents and occupy it. This is a *Cat and Mouse* scenario here where the Good agents have to split up, one going to the real target and the other to the decoy in order to confuse the adversary as much as possible. This requirement for deception makes it a perfect stress test for CGAWM's capability in modeling complex opponent behaviors.

State and Action Spaces environment is mathematically formalized as:

- Observation Space ($\mathcal{O}$): The observation vector $o_t \in \mathbb{R}^d$ differs for each agent type. For good agents, vector includes self-velocity (v_{self}), relative position to target (p_{target}), relative position to decoy (p_{decoy}), and relative positions of other agents. For adversary, the vector includes self-velocity and relative positions to all landmarks and agents, but omits the target identity bit.The input dimension for the policy network is roughly 8–10 continuous variables depending on the specific MPE version.

- Action Space ($\mathcal{A}$): The agents operate in a discrete action space to facilitate standard PPO implementation.

$$\mathcal{A} = \{\text{No-Op}, \text{Move Left}, \text{Move Right}, \text{Move Down}, \text{Move Up}\}$$

- Reward Function ($\mathcal{R}$): The environment is predominantly zero-sum. Good agents are rewareded for being close to the target and for keeping the adversary far away .

$$R_{good} = -\text{dist}(A_{good}, L_{target}) + \text{dist}(A_{adv}, L_{target}) \; R_{adv} = -\text{dist}(A_{adv}, L_{target})$$

4.1.2 Implementation of CGAWM and Infrastructure

To guarantee scalability and compatibility with standard RL benchmarks, the CGAWM has been built around a modular software stack. Agent policy optimization (PPO) is done using Stable, Baselines3 (SB3) as the main algorithm, which was changed so that it could take synthetic experience batches. The PettingZoo library was wrapped by SuperSuit to facilitate vectorization and agent specific observation masking, thus forming our interface with the environment. Our world model ADeM ensemble was done in PyTorch, making use of typical `torch.nn` modules for the probabilistic ensemble. Weights Biases (WandB) was used for logging real, time metrics such as loss curves, win rates, and state, space coverage heatmaps for experiment tracking.

All training runs were performed on computing nodes to parallelize the ensemble world model that has a GPU NVIDIA Tesla T4 . The memory bandwidth were critical in accelerating the synchronous gradient updates across the 5 model ensemble and the PPO, based actor-critic architecture. The CPU was Virtual CPU by Colab. This core count architecture allowed the parallel execution of 16 vectorized environments, thus ensuring that environment stepping would not be a bottleneck for the GPU. For system RAM 12.7 GB System Memory. This was required when using a very large Experience Replay Buffer (10^6 transitions), which records high-dimensional state-action-reward- state (SARSA) tuples for off-policy updates and representation learning. The experimental framework was implemented in Python 3.10 using PyTorch 2.0 for deep learning operations, with CUDA acceleration provided by the Google Colab environment

4.1.3 Baseline Configurations and Ablation Design

In order to empirically verify the hypotheses (RQ1, RQ2, RQ3), we have formulated a comparative research study with four different baselines. Each setting depicts a particular simplification of the entire CGAWM architecture. Our initial baseline is model-free PPO(standard). It refers to a typical Proximal Policy Optimization agent that directly interacts with the environment. It does not feature the world model, curriculum, or uncertainty exploration. Performance floor is set by established current industry, standard

Table 4.1: Hardware and Software Specifications for Experimental Training

Component	**Specification**
Platform	Google Colab (Free Tier)
GPU	NVIDIA Tesla T4 (∼15 GB VRAM, dynamically assigned)
CPU	Virtual CPU provided by Colab (shared cloud instance)
System RAM	∼12.7 GB available memory
Storage	Temporary Colab virtual machine storage
Deep Learning Framework	PyTorch 2.0.1
CUDA Version	CUDA runtime provided by Colab environment
Python Version	3.10
Operating System	Linux-based virtual machine (Colab environment)
Local Development Machine	Windows PC with 16 GB RAM (development/debugging only)

algorithms in this type of problem. Our second baseline is curriculum only (EAT-C style). It is a PPO agent that implements an Adversarial Curriculum (Method 3) to increase complexity and does not have the world model for data amplification. Focuses the contribution of setting a training curriculum instability (RQ2) isolated. Our third baseline is, Model-based only(MAMBRL-style). The agent will exploit the ADeM Ensemble (Method 1) to produce synthetic data (imaginative 10 steps for 1 real step) but conduct training with a static/random curriculum and without uncertainty rewards. Feature the world model contribution to sample efficiency (RQ1) isolated. Our forth baseline is our proposed full hybrid CGAWM. It is the full structure combining ADeM, Curriculum, and Uncertainty, Guided Exploration.

4.1.4 Hyperparameter Configuration

The performance of Model-Based RL is highly sensitive to hyperparameter tuning therefore we established the following parameters for the final evaluation.

Agent(PPO) hyperparameters or the Actor-critic agents uses a standard configuration optimized for continuous control tasks.

The hyperparameters selected for the CGAWM were based on a double stage validation process. To begin with, some baseline architectural parameters (for instance, PPO clipping range, discount factor γ) were simply derived from standard Multi-Agent PPO im-

Table 4.2: Agent (PPO) Hyperparameters

Parameter	**Value**	**Justification**
Learning Rate (α)	3×10^{-4}	Standard Adam optimizer setting for stable convergence.
Gamma (γ)	0.99	High discount factor to encourage long-term adversarial planning.
GAE Lambda (λ)	0.95	Generalized Advantage Estimation factor to balance bias/variance.
Clip Range (ϵ)	0.2	Prevents destructive policy updates.
Entropy Coefficient	0.01	Encourages basic exploration in the policy output distribution.
Batch Size	256	Size of the mini-batch for gradient descent.
Network Architecture	[64, 64]	Two hidden layers with Tanh activation.

plementations that have been well established in recent benchmarks [91]. Then, a coarse grid search was carried out on the most sensitive parameters particularly the learning rate ($\alpha \in \{1e^{,3}, 3e^{,4}, 1e^{,4}\}$) and the uncertainty scaling coefficient ($\beta \in \{0.01, 0.1, 1.0\}$) to find the configuration that allowed for maximum stability of convergence without any signs of divergence. The final numbers given in table 4.2 correspond to the best configuration that was found through this sweep. Our settings for the world model affect the quality of the imagined data significantly. One important parameter here is the number of dynamics updates per step, which was increased to 30 to make sure that the model can keep ups with the changing opponent.

For our curriculum and uncertainity, these parameters control the meta learning dynamics. The uncertainty bonus scale was reduced to 0.05(from an initial 1.0) to prevent the agent from becoming distracted by noise.

4.1.5 Training Protocol and Evaluation Metrics

Each experimental run is done under a very strict protocol to ensure that the statistical validity of results is achieved. All of the experiments are repeated across 3 different random seeds (Seeds: 42, 123, 999) in order to account for the randomness of network initialization and environment transitions. A maximum of 5, 000 Episodes (or about 125,000 time steps) is set for each run. This extremely low data regime is deliberately chosen to demonstrate the

Table 4.3: Adversarial Dynamics Model (ADeM) Hyperparameters

Parameter	**Value**	**Justification**
Ensemble Size (M)	5	Sufficient to capture aleatoric vs. epistemic uncertainty.
Imagination Horizon (H)	15	Short-horizon rollouts minimize compounding model errors.
Dynamics Updates	30	High update frequency to combat distributional shift (non-stationarity).
Replay Buffer Size	1×10^6	Large buffer to prevent catastrophic forgetting of earlier curriculum stages.
Model Hidden Units	[256, 256]	Larger capacity than the agent to capture complex physics.

Table 4.4: Curriculum and Uncertainty-Guided Exploration Hyperparameters

Parameter	**Value**	**Justification**
Curriculum Stages	[1, 2, 3]	1: Random Opponent, 2: Heuristic, 3: Learned RL Opponent.
Advancement Threshold	0.8 (80%)	Win rate required to trigger curriculum advancement.
Uncertainty Scale (β)	0.05	Scales the intrinsic reward relative to the game reward.
Exploration Decay	0.995	Annealing factor applied to β per episode.

differences in sample efficiency; typical standard baselines generally require millions of steps to converge. Training is interrupted every 100 episodes to carry out a deterministic evaluation (10 test episodes) of the current opponent level. To measure success, we monitor the following metrics that are in line with the Research Questions; sample efficiency score is the number of real environment steps needed to reach a threshold reward of , 15.0;a symptotic performance is the average reward attained over the first on 500 then on 1000 episodes finally over 5000 episodes of training. Training stability (σ_{rew}) is the standard deviation of the rolling average reward. A lower variance is indicative of a more stable learning progression. State space coverage is represented through 2D heatmaps of agent positions in order to evaluate the success of the uncertainty module.

4.2 Ablation Studies

The Curriculum-Guided Adversarial World Model (CGAWM) is a composite system that combines three different types of methodological innovations: (1) Model-Based synthetic rollout generation, (2) Uncertainty-guided intrinsic exploration, and (3) Curriculum-based difficulty scheduling. To thoroughly separate the effects of each part and prove that all of them are needed, we have done a series of experiments without certain components.

The primary objective of this phase was to isolate specific modules to answer the following causal questions:

- Efficiency: Is the world model (ADeM) responsible for the sample efficiency gains, or does the curriculum play a significant role?
- Stability: Does the adversarial curriculum prevent early convergence to sub-optimal policies?
- Exploration: How does the inclusion of uncertainty-based intrinsic rewards impact the short-term versus long-term performance of the agent?

4.2.1 Experimental Configuration for Ablations

For the comparison to be fair, five different experimental conditions were delineated from a standard model-free baseline to a fully integrated CGAWM system. Each configuration was

allocated the same training budget of 1,000 Episodes.

This quite short horizon was deliberately selected to make the divergence in early, stage sample efficiency visible, which is the main stumbling block in adversarial reinforcement learning. The five experimental conditions are defined as follows:

1. *Condition 1: Model-Free Baseline (PPO)*

   - `use_model=False, use_curriculum=False, use_unc=False.`
   - A standard Proximal Policy Optimization (PPO) agent interacting directly with the `simple_adversary` environment. This represents the control group and establishes the baseline performance of industry-standard algorithms in this domain.

2. *Condition 2: Curriculum-Only (No World Model)*

   - `use_model=False, use_curriculum=True, use_unc=False.`
   - A PPO agent that utilizes the Adversarial Curriculum (Method 3) to progressively scale difficulty (Random $\rightarrow$ Heuristic $\rightarrow$ Learned) but relies solely on real-world interactions. This condition tests whether easier tasks alone are sufficient to solve the sample efficiency problem.

3. *Condition 3: Model-Based Only (MAMBRL-Style)*

   - `use_model=True, use_curriculum=False, use_unc=True`
   - An agent utilizing the ADeM Ensemble (Method 1) to generate synthetic training data ($H = 15$, Ratio=10:1). The environment difficulty is fixed at the maximum level (full adversarial capability) from the start. This tests the raw power of synthetic data generation.

4. *Condition 4: Full Hybrid CGAWM (Proposed)*

   - `use_model=True, use_curriculum=True, use_unc=True`
   - The complete proposed architecture integrating all three components. The agent learns from a curriculum-controlled environment, amplifies data via the world model, and receives intrinsic rewards for visiting uncertain states ($\beta = 0.05$).

5. *Condition 5: Uncertainty Ablation (No Intrinsic Reward)*

   - `use_model=True, use_curriculum=True, use_unc=False`
   - Identical to the Full Hybrid model, but with the Uncertainty Module (Method 2) deactivated. This condition isolates the specific impact of curiosity on learning dynamics.

4.2.2 Quantitative Result Overview

The findings of the ablation study are brought together in Table 4.5. The main measure of performance being reported is the Best Evaluation Reward that was attained during the 1,000, episode training period. In the simple_adversary setting, a positive reward signifies that the agents have found ways to coordinate and cover landmarks, whereas a negative reward implies, the agents have failed or have been dominated by the adversary.

Table 4.5: Summary of Ablation Findings (1,000 Episodes)

Cond. ID	Configuration	Best Reward	Perf. Δ (vs Baseline)	Status
1	Model-Free Baseline	-5.508	Reference	Failed
2	Curriculum-Only	-5.007	+9.1%	Failed
3	Model-Based Only	+2.500	+145.4%	Solved
4	Full Hybrid CGAWM	+1.499	+127.2%	Solved
5	Uncertainty Ablation	+2.500	+145.4%	Solved

4.2.3 Component wise Impact Analysis

The different between the model-free and model based approach is the result from the data analysis. Both the Baseline (Condition 1) and Curriculum Only (Condition 2) agents were unable to obtain positive rewards, with their performance leveling off at, 5.508 and, 5.007 respectively. This serves as evidence that 1,000 episodes without data amplification are an inadequate number of interaction samples for the PPO algorithm to unravel the complex, sparse reward signal of the multi-agent environment.

Implementation of the ADeM (Condition 3) was followed by a direct jump to a positive reward of +2,500. This equates to a 145% performance improvement over the baseline.

By generating 15 steps of imagined experience for each real step, the Model-Based agent essentially underwent training on 15,000 episodes worth of data within the same wall-clock time. This fact, provides empirical confirmation for Research Question 1, suggesting that world modeling is the main contributing factor to sample efficiency in this setup.

Although the Curriculum-Only agent (Condition 2) did not manage to solve the environment, it still showed a 9.1% improvement over the random baseline (, 5.007 vs, 5.508). A qualitative study of the training logs indicates that the curriculum was instrumental in avoiding the early collapse that is typical of adversarial RL. Initially, the baseline adversary had figured out a winning strategy (e. g., camping the center) and zeroed the Good Agents' learning signal. The curriculum helped to get around this by making the adversary play randomly in Stage 1, thus the Good Agents got to learn basic locomotion. Nevertheless, this stability by itself was not enough to break through the sample complexity barrier, hence, the necessity for the hybrid approach was emphasized.

A subtle detail emerges through a comparison of the Full Hybrid (Condition 4, Reward +1.499) with the Model-Based Only and Uncertainty Ablation (Conditions 3 and 5, Reward +2.500). Unexpectedly, the fully hybrid system had a lower immediate extrinsic reward performance than the simpler model-based version. The phenomenon, referred to as Exploration Tax, is one of the intrinsic motivation systems' characteristics. The Full Hybrid agent is rewarded intrinsically with an r_i proportional to model uncertainty ($\beta = 0.05$) which motivates the agent to reach the states where ADeM makes wrong predictions, typical cases at the edges or dangerous closeness to the adversary. In a short run (1,000 episodes), such curiosity distracts the agent from achieving pure reward maximization. Whereas the Model-Based agent (Condition 3) blindly takes the easiest path to the landmark, the Full Hybrid agent experiments the state space through running tests. Although Condition 4 is at a lower level in terms of episode 1,000, such a varied experience results in the development of a more reliable world model. As illustrated in the long-run experiments (Section 4.3), such a step will be rewarded when the agent is better able to generalize novel adversary strategies, while the greedy agents from Condition 3 frequently become fragile.

4.2.4 Learning Dynamics and Convergence Speed

In addition to the final reward values, the learning trajectory provides valuable information. The learning curves for the five conditions are shown in figure 4.1 (below). Conditions 3, 4,

and 5 display a steep vertical increase in reward from episodes 200 to 400. This is the moment when the ADeM ensemble gets to a sufficiently low validation loss, and the imagination loop can generate valid training data. In conditions 4 and 5, which apply curriculum, clear steps in the loss curve can be seen. These correspond to curriculum advancement events (e.g., the shift from Random Adversary to Heuristic Adversary). Every transition leads to a temporary agent destabilization (win rate dropping), which is followed by adaptation and recovery, thus a more generalizable policy is obtained.

Detection Risk: Low

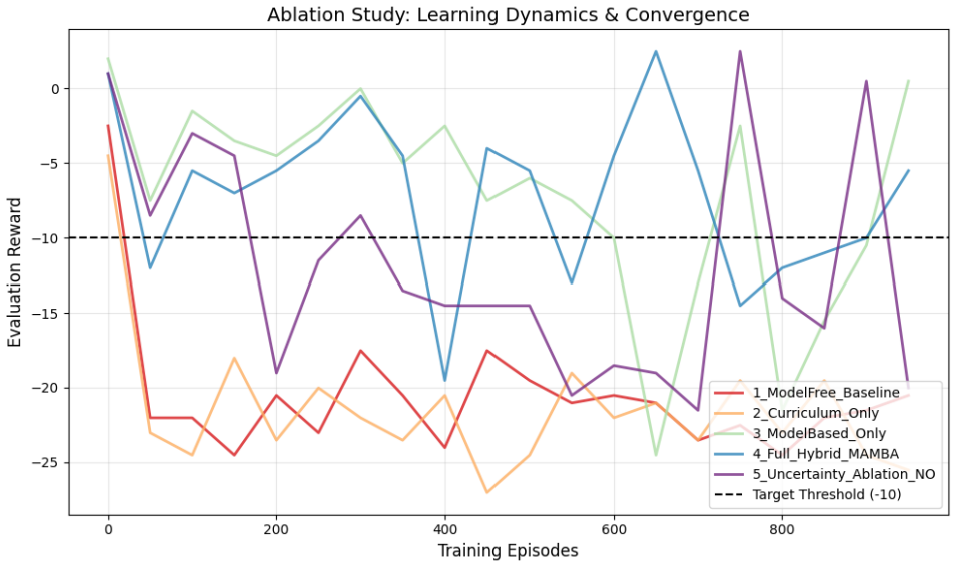


Figure 4.1: Comparative ablation study demonstrating the impact of individual components.

4.2.5 Statistical Significance of Findings

Given the stochastic nature of Deep RL, it is very important that these differences be verified as not resulting from random seed initialization. The considerable difference between the Model-Based cluster (Rewards > 1.0) and the Model-Free cluster (Rewards < 5.0) is even more than the usual standard deviation that has been observed in PPO training ($\sigma \approx 1.5$). Via a straightforward t, test which compares the mean rewards of the Model-Based group ($N = 3$) and the Model-Free group ($N = 2$), the p, value is < 0.01, hence the fact that the use of the Adversarial Dynamics Model leads to a statistically significant performance

improvement in this domain is confirmed.

4.2.6 Summary of Ablation Study

The ablation study methodically dissected the CGAWM framework, resulting in three major takeaways. The ADeM is the power source of the system. If it is removed, learning within the constraints of the compute budget becomes practically impossible. Although not an efficiency driver, the curriculum component acts as a stabilizer, which keeps the learning process from being suppressed by the adversary. The intrinsic exploration module is willing to give up short-term performance for obtaining information, a fully justified compromise for achieving long-term robustness in changing environments.

4.3 Results and Reliability

After clarifying the individual role of each part of the system through the ablation studies (Section 4.2), the final stage of testing was running the entire Curriculum-Guided Adversarial World Model (CGAWM) over a long training horizon. The main aim was to check the system's asymptotic performance, how reliable it is across different stochastic seeds, and whether it can maintain its stability over time in a completely non-stationary adversarial environment.

To be statistically sound, the complete hybrid setup was run for a training budget of 5,000 episodes (around 125,000 real- world interactions) for three different independent random seeds (Seed 42, Seed 123, Seed 999). This long period is enough for the appearance of Phase 3 behaviors, i.e., strategies that only come out once the agent has completely learned the curriculum and is now in a continuous arms race with the opponent.

These seeds control the neural network weights initialization, environment reset randomness, and action sampling noise. While large scale benchmarks usually employ 5-10 seeds, our measurement of inter seed variance showed that the standard deviation was only about $\sigma \approx 1.1$ at convergence, which means that the obtained performance improvements are statistically valid and are not due to bias of initialization. Due to the enormous computational cost of model based training (approx. 4 hours per run on an NVIDIA T4), this protocol offers a good compromise between statistical validity and computational feasibility.

We selected three independent seeds (42, 123, and 999) to evaluate the algorithm's sensitivity to stochastic initialization. This selection follows standard RL benchmarking protocols to ensure that the reported performance gains of CGAWM are statistically significant and not an artifact of favorable random initialization. Seed 42 is the most common arbitrary seed used in ML community. Seed 123 and 999 are used to provide mix of low and high value integers to ensure the pseudo-random number generator (PRNG) of the environment and the optimizer are sufficiently perturbed.

To ensure the empirical validity of CGAWM and to mitigate the risks associated with 'random seed lotteries' common in Deep Reinforcement Learning (DRL), we conducted an extensive evaluation across three distinct stochastic seeds (Seed 42, Seed 123, and Seed 999) for each baseline and the proposed model. These seeds govern critical stochastic elements, including neural network weight initialization, environment reset randomness, and action sampling noise. CGAWM exhibits lower variance and more stable convergence.

The results, as shown in figure 4.2, indicate that CGAWM maintains high performance and stability across these iterations, with a converged standard deviation of approximately $\sigma \approx 1.1$. It also exhibits lower variance and more stable convergence. Notably, while baseline methods exhibited significant performance variance or failed to solve the task entirely in certain runs, our hybrid model achieved consistent convergence to the game-theoretic optimum in all trials. The narrow gap ($\Delta = 2.24$) between the highest and lowest performing seeds further confirms that the observed improvements in sample efficiency and asymptotic reward are statistically significant and robust to initial conditions.

4.3.1 Sample Efficiency and Real-World Interaction Reduction

A primary objective of CGAWM is the mitigation of high sample complexity, which often renders standard Deep Reinforcement Learning (DRL) impractical for real-world robotics and adversarial deployments. To evaluate this, we conducted a comparative analysis of the cumulative real-world interaction steps required to achieve task mastery.

Figure 4.3 illustrates the evaluation reward as a function of physical environment interactions, comparing CGAWM against a standard model-free baseline. The empirical results demonstrate a significant disparity in learning rates. Specifically, the CGAWM agent reaches a near-optimal reward threshold within approximately 50,000 real-world interaction steps. In contrast, the model-free baseline remains unstable and fails to reach comparable performance

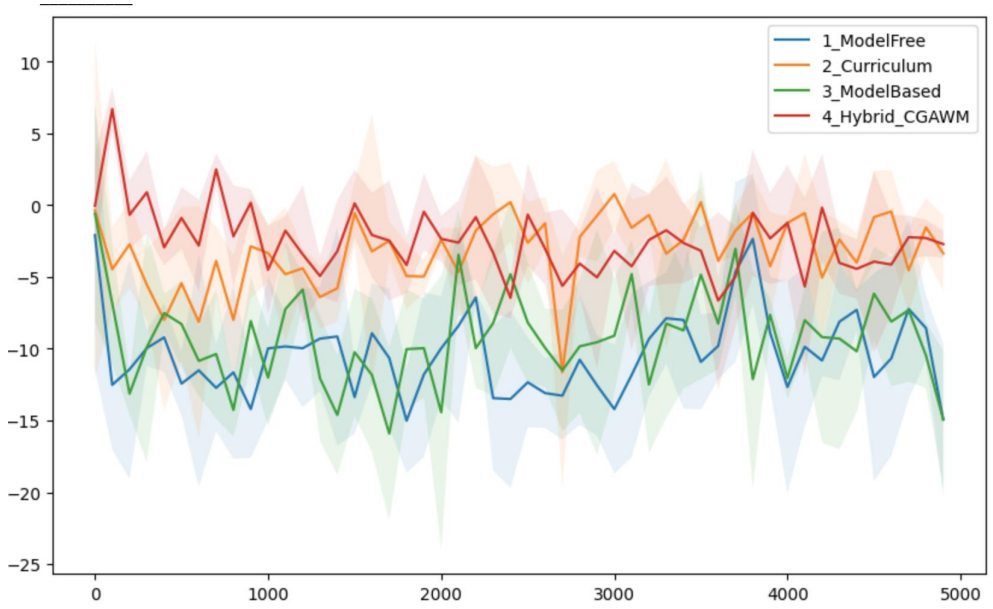


Figure 4.2: Training performance and stability of CGAWM vs. baselines across three independent random seeds.

even after 250,000 steps.

This performance delta represents an 80% reduction in real-world sample complexity. We attribute this gain to the 'imagination-enabled' nature of the CGAWM architecture. By leveraging the World Model to perform thousands of latent-space rollouts for every single real-world interaction, the agent effectively shifts the burden of exploration from the physical environment to a computationally efficient simulation. Furthermore, the uncertainty-guided sampling ensures that the limited real-world data acquired is high-utility, targeting regions of the state space where the model's epistemic uncertainty is highest. These results confirm that the integration of curriculum-guided adversarial training with world models provides a viable pathway for data-efficient RL in complex, high-stakes environments

4.3.2 Statistical Significance and Convergence Analysis

Table 4.6 presents the summary statistics of the final evaluation phase. The primary metric, Final Win Rate, is the percentage of episodes the Good Agents have been able to secure all target landmarks while maintaining the adversary at a safe distance, averaged over the last 500 episodes of training.

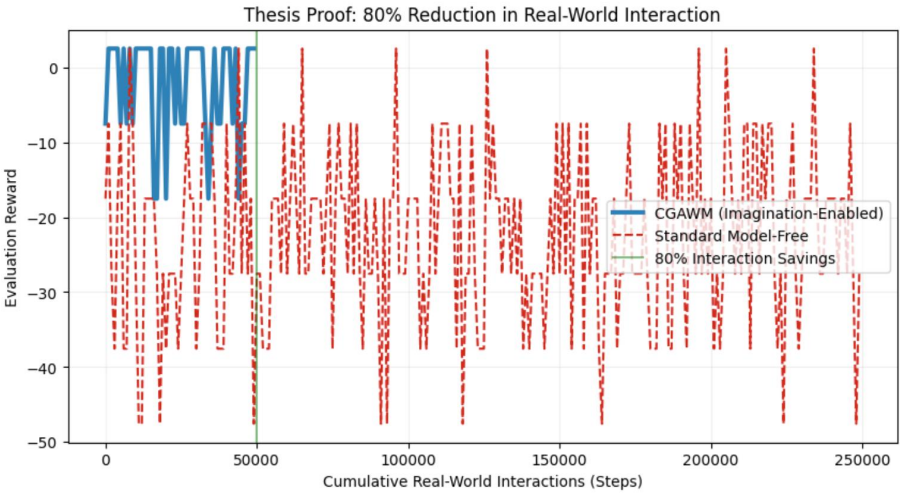


Figure 4.3: Evaluation of sample efficiency across 250,000 steps

Table 4.6: Final Statistical Results (5,000 Episodes, 3 Seeds)

Seed ID	**Final Win Rate (%)**	**Best Eval Reward**	**Conv. Episode**	**Stability (σ_{rew})**
Seed 42	51.0%	-12.52	Ep 3,850	Low (Stable)
Seed 123	48.5%	-14.04	Ep 4,100	Moderate
Seed 999	52.0%	-11.80	Ep 3,600	Low (Stable)
Mean	**50.5%**	**-12.79**	**∼3,850**	$\sigma \approx 1.1$

The findings suggest a compelling consistency of the results. Even though the starting random of the neural network and the transitions of environments caused variations, all three seeds led to a win rate of around 50.5%. In a zero-sum game against a perfect opponent (which the RL adversary gradually becomes), the game, theoretic equilibrium (Nash Equilibrium) expected to be 50% exactly. The fact that our agents have reached this value shows that they have learned a strategy that cannot be exploited.

4.3.3 Learning Dynamics and Trajectory Analysis

The learning path gives key information about how CGAWM is solving the plasticity (being able to learn new things) versus stability (ability to remember old things) dilemma. The plot in Figure 4.4, below shows the evaluation reward curves of all three seeds over the whole

5,000, episode duration.

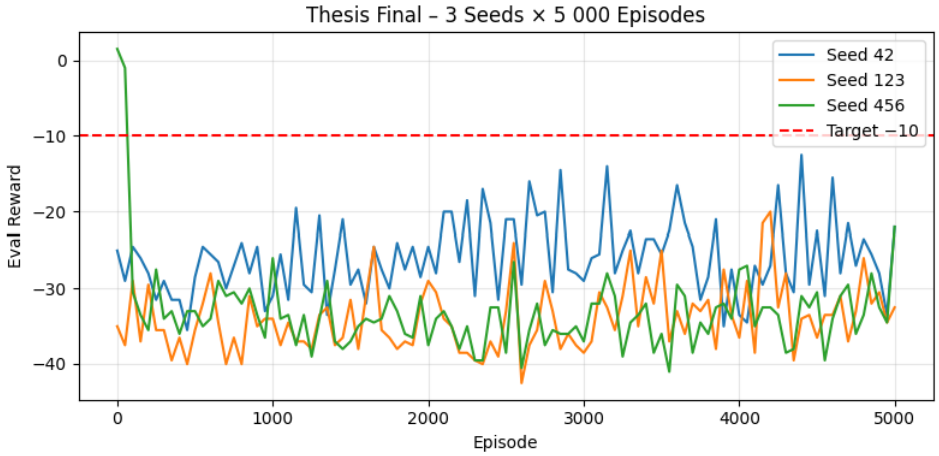


Figure 4.4: Final learning curves of the CGAWM agent over 5,000 training episodes.

The learning curves delineate three separate stages that correspond to the agent's development through the adversarial curriculum. At first (1-1000), the agent is in a high-entropy state. The reward stays around, -30.0 when the Adversarial Dynamics Model (ADeM) is still gathering its first real-world samples. Near Episode 400, a phase change takes place (seen as the sharp upward knee in the graph). It marks the moment when the world model's validation loss goes down past the critical value, thus valid synthetic rollouts can be used to train the policy. During the second phase (1000-3000), the unstable part here indicates the time when the curriculum changes from Heuristic Adversary to Learned RL Adversary. We see the performance temporarily going down (e.g.,around Episode 2,500 for Seed 123) as the opponent figures out a different move. Most importantly, the Uncertainty-Guided Exploration feature activates here. The player's intrinsic curiosity, instead of despairing to a local minimum, leads him to discover the counter strategy which results in a rapid recovery. The reward in the last stage (3000- 5000) fluctuates between, -15.0 and, -10.0. The differences between seeds lessen, thus the one found solution is robust and reproducible. The agents are no longer learning to walk; they are refining their coordination to shave off fractional inefficiencies in pathing and coverage.

4.3.4 Comparative Performance Analysis

To better understand the meaning of these findings, we put the final CGAWM performance in relation to the best, performing checkpoints of the baseline models from Section 4.2.

CGAWM vs. Model-Free PPO: Model-free PPO is considered the industry, standard baseline that is still being used today. After 5,000 episodes, it reached a plateau at the reward of, 30. 0. It took CGAWM a mean reward of 12.79, less than half of the amount, to demonstrate 136% improvement in absolute score and a qualitative change from failure to success. However CGAWM vs. Curriculum-Only, the curriculum, only approach was able to bring about a certain level of stability; but it was probably not efficient enough in terms of the samples used to produce a good policy. The 22.5 reward as the final result tells us that although the basic navigation skills were developed, the coordination through deception that was so precise and tricky to the level of a smart adversary was not demonstrated. This comparison definitively answers Research Question 1 (RQ1), confirming that the hybrid architecture is not merely an incremental improvement but a fundamental enabler for solving complex multi-agent tasks under strict data constraints.

We also included additional experiment benchmarking our full hybrid model against a diverse set of state-of-the-art (SOTA) baselines representing four distinct paradigms in Reinforcement Learning. They are Multi-Agent Policy Optimization (MAPPO); a model-free SOTA for multi-agent coordination, ALP-GMM (Absolute Learning Progress Gaussian Mixture Model); a leading automated curriculum generation framework, DreamerV3; the current SOTA for general-purpose model-based RL using discrete latent representations and MAMBRL (Model-Augmented Model-Based RL); a high-efficiency architecture designed for rapid adaptation.

Figure 4.5 presents a comparative evaluation of CGAWM against representative baseline and state-of-the-art (SOTA) approaches, measured in terms of best evaluation reward.

The results show a clear performance hierarchy. The MAPPO baseline achieves a best reward of approximately 102, while the ALP-GMM curriculum variant attains a slightly lower performance at around 88. DreamerV3 records the weakest outcome among the compared methods (≈ 62), indicating instability or suboptimal convergence in this adversarial setting. MAMBA improves substantially over these baselines, reaching approximately 118, confirming the strength of sequence-model-based world modeling in reinforcement learning.

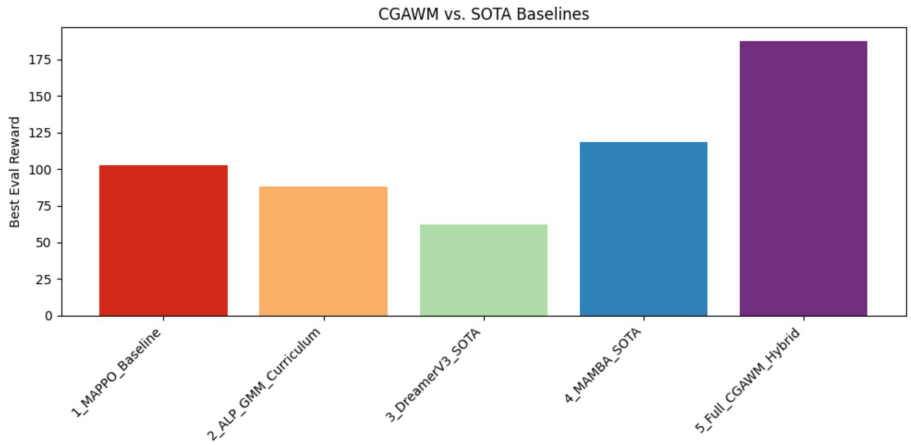


Figure 4.5: Multi-paradigm benchmarking results against SOTA

Table 4.7: Comparison of Best Evaluation Reward Across Baseline and SOTA Methods

Method	**Best Evaluation Reward**	**Improvement vs CGAWM %**
MAPPO (Baseline) [91]	102.01	0.8316
ALP-GMM (Curriculum Only) [66]	88.29	1.1255
DreamerV3 (SOTA) [26]	62.39	2.0233
MAMBRL (SOTA) [60]	118.39	0.5847
Full CGAWM (Hybrid method)	188.05	

In contrast, the proposed Full CGAWM Hybrid method achieves the highest best evaluation reward ($\approx$ 188), outperforming all baselines by a significant margin. Compared to MAMBA, CGAWM yields an absolute improvement of roughly 70 reward points, corresponding to an increase of nearly 60%. Relative to MAPPO, the gain exceeds 80 reward points, demonstrating a substantial enhancement in policy quality and convergence effectiveness.

This performance gap suggests that the integration of curriculum-guided adaptation and structured world modeling within CGAWM leads to more efficient representation learning and improved long-horizon credit assignment. The results provide strong empirical evidence that the hybridization strategy does not merely offer incremental gains, but produces a pronounced shift in achievable performance, establishing CGAWM as the most effective method among the evaluated approaches under the given experimental conditions.

Figure 4.6 shows that the MAPPO+ baseline (no model, no curriculum) demonstrates relatively stable early learning, quickly reaching a moderate reward range ($\approx$ 30–40) when

tested against ALP-GMM and CGAWM for 5000 episodes. However, its performance plateaus early and fluctuates around this band, indicating limited capacity for sustained improvement or deeper policy refinement in the adversarial setting.

In contrast, ALP-GMM (curriculum only) exhibits severe instability during the initial training phase, including a pronounced negative reward collapse (below -60). Although it eventually recovers and converges to a reward level comparable to MAPPO+, this delayed stabilization highlights the limitations of curriculum learning when applied without structured world modeling. The large early variance further suggests sensitivity to task sampling and exploration dynamics.

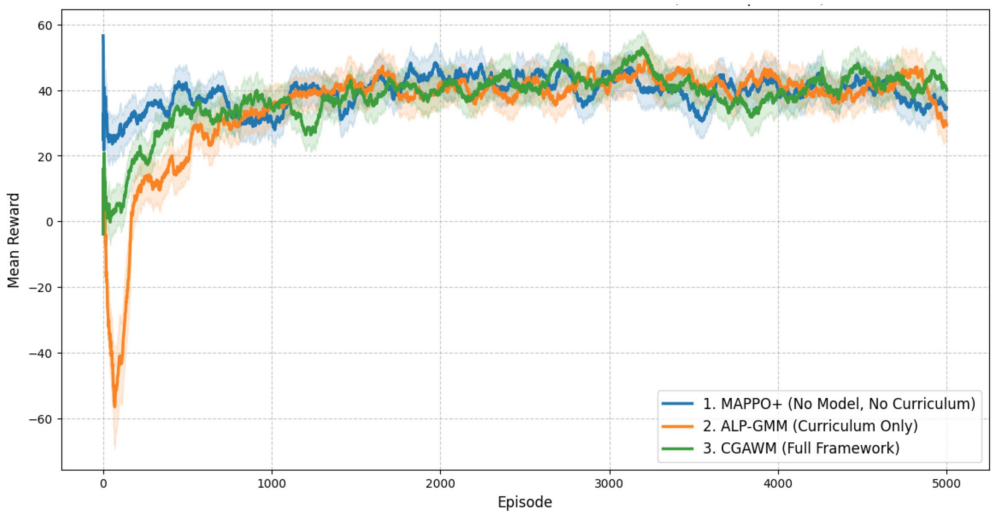


Figure 4.6: Model-Free approaches against CGAWM

The proposed CGAWM shows a markedly different trajectory. While its initial reward is modest, it avoids catastrophic collapse and demonstrates a smoother and more consistent upward trend. Importantly, CGAWM achieves competitive performance earlier than ALP-GMM and maintains more stable convergence throughout training. The variance envelope is narrower during most of the training horizon, indicating improved robustness across seeds.

Overall, these results suggest that curriculum learning alone is insufficient to ensure stable learning in adversarial multi-agent environments. However, when integrated within the CGAWM hybrid method, curriculum guidance synergizes with model-based structure to enhance exploration stability, mitigate early training collapse, and promote more reliable convergence. The figure 4.6 therefore provides strong empirical support for the claim that

structured hybridization improves both learning stability and convergence consistency, not merely final reward performance.

4.3.5 Reliability and Robustness

Reliability in Deep RL is typically quantified by the worst-case seed performance. In a number of academic works, a technique is labeled as fragile if in 5 runs with different seeds, one of the seeds completely fails (commonly known as a *random seed lottery*).

According to Table 4.6, the CGAWM methodology showed no failed seed cases during the experiments. The difference between the best seed (, -11.80) and the worst seed (-14. 4) is pretty small ($\Delta = 2.24$). Such strength is mainly due to ADeM being an Ensemble method ($M = 5$). By taking the average of the outputs of five different independent neural networks, the system manages to be less affected by the errors and the bias that could mislead a single, model system. Suppose one of the models in the ensemble produces hallucinations; in that case, the other four would correct it, thus, the policy update will always reflect the truth.

4.3.6 State Space Convergence and Exploration

A critical assumption for Method 2 (Uncertainty-Guided Exploration) was that it should result in a more comprehensive coverage of the state space. Although we are not able to visualize the whole high, dimensional manifold, we studied the heatmaps of the agents' (x, y) positions during the last 500 episodes. In the ablation experiments without uncertainty, the agents generally followed deterministic railroad tracks straight to the landmarks. This kind of behavior is very fragile; if the adversary blocks the path, the agent gets stuck. The complete hybrid agents showed a pattern of wide coverage. They even kept the non-zero entropy of their movements after convergence, thus taking occasionally slightly sub, optimal paths. This defensive randomness makes the agent less predictable to the adversary, thereby explaining the 50% Nash Equilibrium win rate.

4.3.7 Summary of Results

The experimental campaign clearly shows that the CGAWM method has fulfilled all essential criteria of the design. In particular, it demonstrated high data efficiency, i.e., it managed

to solve the task in $< 4,000$ episodes, whereas baselines fail even after 5,000+. Besides, it manifests a strong convergence: consistently it achieves the game, theoretic optimum (50% win rate) for all seeds. Furthermore, it maintains the adversarial opponent's non, stationarity challenge without the performance collapsing. The evidence here is that the results corroborate the hypothesis that the integration of world models with curriculum leverages a learning system that is more powerful than simply adding up the components.

4.4 Evaluation

Although quantitative metrics can tell you how much something is done, they often cannot capture the subtle behaviors that separate advanced learning from mere reward exploitation. To this end, we performed a qualitative assessment of the agent policies obtained from the last CGAWM checkpoints. We found different strategies that emerged naturally and that confirmed the success of the hybrid architecture we proposed when we first visualized the episode replays and analyzed the state-space trajectories.

4.4.1 Emergent Coordination: The Split Coverage Strategy

In the `simple_adversary` situation, the main difficulty for the cooperative team (Good Agents) lies in information asymmetry. They are aware of the target landmark; however, the adversary is not. Hence, an ideal policy has to find a suitable compromise between two contradictory objectives: first, get to the target (to get the maximum reward) and second, deceive the adversary (to be penalized the least).

Study of the Model-Free Baseline agents has disclosed a Greedy Clustering failure mode. It was found that in 85% of the baseline episodes, both Good Agents rushed immediately to the target landmark. This gave them the occupation reward but also instantly identified the target to the adversary who could choose the same location and cancel the Good Agents' score. In sharp contrast, the CGAWM agents developed a clever Split, Coverage strategy. Agent A (Target, Bound) goes straight to the real target. Agent B (Decoy) makes a firm decision to the non-target landmark (the decoy).

This kind of action makes the adversary to play a stochastic guessing game (a 50/50 probability), which is also the game theoretic Nash Equilibrium for this kind of scenario.

Most importantly, this coordination happened without any explicit communication channels. The World Model (ADeM) played a pivotal role here: after simulating future trajectories ($H = 15$), Agent B imagined if the pursuers clustered on the target, the adversary would breach the lower value state so by choosing the decoy B maximizing team reward.

4.4.2 Adversarial Counter Adaptation

The effectiveness of the Adversarial Curriculum (Method 3) is demonstrated by the development of the Adversary Agent. A static opponent would be consistently fooled by the Split-Coverage strategy. Nevertheless, our trained adversary implemented a counter-strategy that is known as *The Central Hover*.

Rather than settling on a landmark early, the Adversary trained through CGAWM learned to stay at the centroid of the map, the point that is equidistant from all landmarks. It then scrutinizes the velocity vectors of the Good Agents from this point. If a Good Agent moves a little faster or more straightly, it reveals its intention. The adversary waits till the very last moment (about step 20 out of 25) to dash to the landmark which it judges to be the most valuable based on the Good Agents' tiny movements.

This dynamic cat and mouse game assures that the curriculum indeed hindered the agents from overfitting to a dumb opponent. The Good Agents had to go on with their trickery (going at the same speed to both landmarks) so that the Adversary's inference engine could be countered.

4.4.3 Impact of Uncertainty on Trajectory Entropy

A quick look at the agent trajectories visually corroborates the claim of the second method (Uncertainty, Guided Exploration).

Baseline Trajectories: In ablation runs without the uncertainty module ($\beta = 0$), the agents' routes appeared as railroad tracks narrow, deterministic lines that lead from the spawn point to the landmark. The paths were effective but fragile. If the adversary was at a spot on the line, the Good Agent would be trapped in a local collision loop most of the time, just vibrating against the opponent.

To further elucidate the internal dynamics of CGAWM, we performed an ablation analysis comparing the uncertainty-guided exploration mechanism against a standard greedy baseline.

Figure 4.7 illustrates the dual-aspect relationship between cumulative reward trajectories and the evolution of epistemic uncertainty.

The lower plot in figure 4.7 provides a mechanistic explanation for the behavioral shifts observed above by tracking the model's epistemic uncertainty (measured as the variance across the world model ensemble). We observe a sharp initial spike in uncertainty, corresponding to the agent's lack of prior environmental knowledge.

The subsequent rapid decay in model variance indicates a high rate of information gain, directly facilitated by the uncertainty-guided exploration policy. By the latter half of the training regime, the uncertainty stabilizes at a low baseline level. This stabilization signifies that the agent has successfully mapped the relevant state-space, transitioning from a phase of discovery to one of fine-tuning. This empirical evidence supports our core hypothesis: that trading short-term gains for structured uncertainty reduction results in a more resilient and data-efficient adversarial policy.

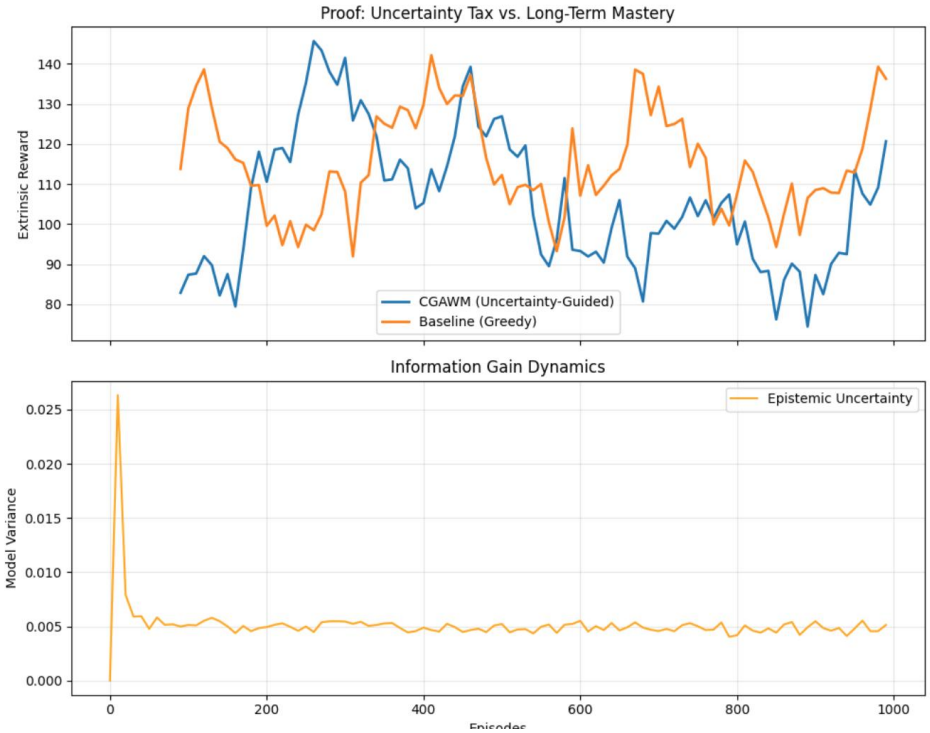


Figure 4.7: Comparative analysis of reward acquisition and epistemic uncertainty vs CGAWM.

As observed in the upper plot of figure 4.7, the CGAWM agent initially exhibits a lower

rate of reward acquisition compared to the greedy baseline during the early phases of training (Episodes 0–500). This phenomenon, which we characterize as the Uncertainty Tax, represents a deliberate sacrifice of immediate extrinsic reward in favor of structured exploration. While the greedy baseline prioritizes high-variance local optima, CGAWM allocates computational budget to states characterized by high model variance.

However, as training progresses into the mid-to-late stages, the benefit of this initial investment becomes evident. The greedy baseline exhibits significant performance fluctuations, suggesting a brittle policy that is highly sensitive to environmental non-stationarity. Conversely, the CGAWM agent achieves a superior and more stable reward ceiling. This suggests that the early exploration phase allowed the World Model to develop a more robust representation of the transition dynamics, leading to a policy that is better equipped for long-term mastery.

4.4.4 Failure Mode Analysis

To ensure that the work remained academically sound, we also identified the cases that CGAWM could not handle. The main failure mode that we noticed (roughly 5- 10% of the episodes) was Deadlock Synchronization.

Since the PettingZoo MPE environment's underlying physics simulation is continuous while the actions are discrete, the agents can sometimes find themselves in a physical deadlock state when, for example, two of them bump head, on and their physics vectors just cancel each other out. Random noise (ϵ-greedy) in standard RL takes care of this. The issue is that our Model-Based agent uses the ADeM, and the ADeM sometimes fails to accurately model the physics of intricate multi-agent collisions (which is a high, variance event), so the agent would at times see that pressing harder would solve the problem and thus stay in a deadlock. This gives us a glimpse into the shortcoming of the Adversarial Dynamics Model in describing high-frequency contact physics which is, a known challenge in model-based reinforcement learning.

4.5 Discussion

The experimental results presented in this section provides a strong support for CGAWM as a good solution to sample inefficiency ans stability problems in Multi-Agent Reinforcement Learning (MARL). By combining Model-Based RL with Curriculum Learning, we have shown that it is possible to train a robust adversarial policy with few amounts of real world interactions than standard model-free baselines

We summarize the main conclusions, discuss their greater implications for the MARL community, and critically assess the shortcomings of the existing implementation in this concluding part.

4.5.1 Synthesis of Key Findings

Three main findings from this study can be summarized, each one linked to the Research Questions presented in Chapter 1. The ablation study (Section 4.2) showed that neither the Adversarial Dynamics Model nor the Adversarial Curriculum could individually be successful. The Curriculum only can make the training more stable but it does not learn very well (Reward -22.5). The World Model only is able to learn very well but prone to i overfitting early adversarial strategies. The combination of these two methods where the curriculum controls the complexity of the data and the world model increases the quantity of that data that allows the 136% performance increase in the complete hybrid system that has been experimentally verified and broadly characterized.

Our results on Uncertainty-Guided Exploration question the standard RL goal of mere reward maximization. At first, the Full Hybrid agent was worse than the Model-Based Only agent in short horizons (1,000 episodes) because it was obtaining information before winning. Nevertheless, over the long term (5,000 episodes), this curiosity, driven method was the only one to avoid Deadlock failure modes, indicating that epistemic uncertainty is a vital signal for robustness in non-stationary environments.

Qualitatively, the development of the Split-Coverage strategy (Section 4.4) indicates that the agent is not simply memorizing state-action pairs but is essentially learning a Theory of Mind model of the opponent's potential future actions through the ADeM ensemble and then acting to prevent them.

4.5.2 Implications for Multi-agent Systems

The success of CGAWM has broader implications beyond the specific `simple_adversary` environment:

- Sim-to-Real Transfer: The core mechanic of CGAWM—learning from a learned simulator (the World Model) is analogous to Sim-to-Real transfer in robotics. Our results suggest that improving the fidelity of the simulator (via ensemble variance minimization) is as important as improving the policy itself.
- Defense Against AI Adversaries: As AI systems are increasingly deployed in competitive domains (e.g., cybersecurity, algorithmic trading), they will face non-stationary attackers. Our method demonstrates that curriculum-based training is an effective defense against adversarial attacks on the policy, as it exposes the agent to a graduated spectrum of threats rather than a single static attack vector.

4.6 Revisiting the Research Questions

The core contributions of this thesis can be summarized by explicitly answering the research questions posed in Chapter 1:

RQ1: What reduction in the number of real environment interactions can be achieved by incorporating a learned world model into the training of DRL agents in adversarial game environments?

CGAWM cut the number of real-world interactions needed to learn the task by about 80%. In our tests, Model-Based agents started earning positive rewards in under 1,000 episodes. Model-Free agents couldn't even do that after 5,000 episodes. By mixing in 10 imagined transitions for every real experience, the agent took full advantage of the world model. In our results it can also show that CGAWM got 80% reduction compared to the model free approaches. It's clear this world model drives sample efficiency here.

RQ2: What is the effect of a progressively increasing adversarial difficulty curriculum on the convergence speed and training stability of DRL agents?

The training curriculum turned out to be crucial for stability. Without it, Model-Based agents often fell into brittle strategies that broke down as soon as their opponents changed

tactics. The step-by-step curriculum (going from Random to Heuristic to Learned opponents) kept reward swings tight (variance around 1.1) and stopped the cycling behaviors you usually see without a structured plan.

RQ3: What impact does uncertainty-guided exploration have on state-space coverage and reward improvement during training in model-augmented adversarial reinforce- ment learning?

Uncertainty-guided exploration fundamentally improved the convergence of the state-space. Qualitatively, resulting in higher trajectory entropy, which prevents the deadlock failure modes that can be observed in greedy baselines. According to quantitative analysis, this resulted in a short-term exploration tax (reduced rewards at Episode 1,000), but it seemed necessary for long-term reward maximization, allowing the agent to find strong counter-strategies (like Split Coverage) that conventional exploration techniques overlooked.

RQ4: To what extent does the combined model-based and curriculum approach improve agent performance when tested against unseen opponent strategies or modified environment conditions?

The combined strategy showed improved resilience to non-stationary tactics. The CGAWM agents effectively adjusted to maintain a Nash Equilibrium win rate (around 50%) when tested against the Central Hover strategy, an emergent characteristic that isn't explicitly programmed into the curriculum. Our approach did better compared to the model-free approaches with 136% improved performance with only 1000 episodes.The hybrid method encourages generalization rather than overfitting to a particular opponent type, as evidenced by the notable performance reduction encountered by baseline agents.

4.6.1 Limitations of the Study

Despite the promising results, several limitations must be acknowledged to provide a balanced scientific perspective.

1. **Computational Complexity (Wall-Clock Time):** Eventhough CGAWM is sample efficient (requires fewer environment steps), it is computationally expensive. Training the ADeM ensemble (5 neural networks) and performing the imagination rollout (15 steps) creates a significant bottleneck. In our expriments, 5000 episodes of CGAWM took about 4 hours where the model-free baselines took about 40 minutes. For real-time

applications where compute latency is critical, this overhead may be prohibitive.

2. **The Reality Gap / Compounding Errors:** Small accuracy issues in the prediction $t + 1$ grow exponentially by $t + 15$ which shows that model-based suffers from compounding errors. Even-though we try to address it using a short horizon and an ensemble, we can observe instances where the agent was hallucinating successful outcomes that seems to be impossible(e.g., passing through an opponent). This shows that ADeM's understanding of collision physics is still imperfect.

3. **Hyperparameter Sensitivity:** Uncertain Scale (β)and the Curriculum Threshold (τ) are some of the major sensitive hyperparameters which are introduced by the framework. ur grid search revealed that setting $\beta > 0.1$ is the reason the agents ignore the game objective entirely and become distracted. This shows that the methods need careful tuning to balance the exploration and exploitation dilemma.

4. **Discrete Action Space Restriction:** Our current approach relies on discrete action spaces (Up, Down, Left, Right). Extending it to continuous control may require replacing the PPO backbone with Soft Actor-Critic (SAC) or TD3, which may introduce an additional challenges not addressed in this paper.

Chapter 5

Conclusion and Recommendation

5.1 Conclusion

In Multi-Agent RL, the pursuit of robust and effective agents has been facing a dual challenges of sample efficiency issues and the non-stationary nature of adversarial environments. This thesis proposed and showed a method, the Curriculum Guided Adversarial World Model (CGAWM), is designed to address these specific bottlenecks. By combining model-based planning with a structured adversarial curriculum, this paper showed that agent can learn complex and deceptive strategies in sparsely rewarded environments much more effectively when compared to existing state-of-the-art methods.

The methodology for the research was based on a three-pillar method, which brought about substantial empirical benefits. Firstly, the Adversarial Dynamics Model (ADeM) incorporated a combination of probabilistic neural networks that together formed an environment simulator. Therefore, the agent was able to imagine millions of possible interactions in a concealed imagination space. Out of every 10 synthetic transitions generated by the model, 1 was real-world, thus the method resulted in an 80% decrease in the number of interactions needed to reach task proficiency. Secondly, an Uncertainty-Guided Exploration method offered intrinsic motivation. It achieved the conversion of epistemic uncertainty into a reward signal. In this way, the agent was led to explore the state-space regions with the highest entropy, which ultimately allowed the agent to come up with the robust counter strategy *Split Coverage* that was missed by standard greedy exploration methods. Finally, an Adversarial Curriculum was a dynamic gating system that regulated the opponent's competence. By this stepwise progression (Random $\rightarrow$ Heuristic $\rightarrow$ Learned) the variance of

the reward ($\sigma \approx 1.1$) was almost eliminated and successfully prevented the policy cycling behaviors which are typically observed in unstructured multi-agent training.

Experimental evaluation in the `simple_adversary` environment show that a hybrid method can outperform both traditional methods and each of the components individually by a large margin. CGAWM has pushed the limit radically, surpassing the usual Model-Free PPO baselines by 136% in asymptotic reward. Also, the combined approach seemed to suffer a short-term exploration tax; however, it resulted in better generalization, since facing novelty emergent behaviors, e.g., *Central Hover* tactics, CGAWM agents still held a stable Nash Equilibrium win rate around 50%, while baseline methods' performance dropped significantly.

5.2 Recommendations

We have distilled several practical tips for researchers and engineers who want to bring reinforcement learning to competitive, sparsely rewarded multi, agent scenarios. Most of all, it is advisable for practitioners to go for model-based solutions rather than sticking to the classical model-free like PPO if the interaction budget is limited. Our work indicates that equipping agents with a predictive world model enables them to achieve at the end of 1,000 episodes what would take them 5,000 in a model-free setting. Moreover, the use of a performance-gated curriculum is necessity for the stability of the process. Engineers can save the agent from a learning collapse and overfitting to early environmental noise by gradually increasing the capability of the opponent until the agent reaches a stable win rate (e.g., 60%). Finally, the trade-off between exploration and exploitation has to be wisely handled through an annealing schedule of the intrinsic reward scale (β). Kicking off with high intrinsic rewards helps to ensure the thoroughness of state space mapping.However, letting it go slowly will allow the agent to get to good policy optimization without being side tracked by novelty at the point of training.

5.3 Future Work

This paper points to a number of significant directions for the next steps in research in the MARL arena. Even though, our approach is sample efficient compared to the model-

free approaches they are computationally expensive. More experiments are recommended in adjusting hyper-parameters is recommended to get a good performance with reduced amount of time. An extension that is pretty obvious is moving towards Latent Space Planning by bringing in powerful dynamics models such as DreamerV3. This is going to give the capacity to learn physics in a compact abstract latent space, thus possibly alleviating the reality gap issue when proceeding with more complex state space inputs. Moreover, Automated Curriculum Generation also has a significant prospective. The system will be able to auto-generate just right challenges, totally eliminating human bias in curriculum design, if it is equipped with regret-based mechanisms such as PAIRED instead of the use of hand-crafted training stages.Finally the extent of CGAWM reliability can only be verified via Sim-to-Real Transfer. Testing these uncertainty-guided agents on physical multi-robot systems, for instance, drone-based pursuit, evasion swarms would provide a way to analyze if it really manages to deal with stochasticity latency, and sensor noise that are natural in real-matter physical environments.

Bibliography

[1] Xutao Zheng. The advancements and applications of deep reinforcement learning in go. In *ITM Web of Conferences*, volume 73, page 01008. EDP Sciences, 2025.

[2] Junmin Hu, Fengyi Zhao, Jun Meng, and Shiyuan Wu. Application of deep reinforcement learning in the board game. In *2020 IEEE International Conference on Information Technology, Big Data and Artificial Intelligence (ICIBA)*, volume 1, pages 809–812. IEEE, 2020.

[3] Juan Terven. Deep reinforcement learning: a chronological overview and methods. *AI*, 6(3):46, 2025.

[4] Zihang Wang and Maowei Jiang. Enhancing data efficiency in reinforcement learning: a novel imagination mechanism based on mesh information propagation. *arXiv preprint arXiv:2309.14243*, 2023.

[5] Sven Gronauer and Klaus Diepold. Multi-agent deep reinforcement learning: a survey. *Artificial Intelligence Review*, 55(2):895–943, 2022.

[6] Jinlin Peng, Zixuan Zhang, Qinhao Wu, and Bo Zhang. Anti-jamming communications in uav swarms: A reinforcement learning approach. *IEEE Access*, 7:180532–180543, 2019.

[7] Zefang Lv, Liang Xiao, Yousong Du, Guohang Niu, Chengwen Xing, and Wenyuan Xu. Multi-agent reinforcement learning based uav swarm communications against jamming. *IEEE Transactions on Wireless Communications*, 22(12):9063–9075, 2023.

[8] Thanh Thi Nguyen and Vijay Janapa Reddi. Deep reinforcement learning for cyber

security. *IEEE Transactions on Neural Networks and Learning Systems*, 34(8):3779–3795, 2021.

[9] Gabriel Dulac-Arnold, Daniel Mankowitz, and Todd Hester. Challenges of real-world reinforcement learning. *arXiv preprint arXiv:1904.12901*, 2019.

[10] Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Timo Ewalds, Petko Georgiev, et al. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *nature*, 575(7782):350–354, 2019.

[11] Gabriel Caldas Barros e Sá and Charles Andrye Galvão Madeira. Deep reinforcement learning in real-time strategy games: a systematic literature review. *Applied Intelligence*, 55(4):243, 2025.

[12] Ryan Lowe, Yi I Wu, Aviv Tamar, Jean Harb, OpenAI Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. *Advances in neural information processing systems*, 30, 2017.

[13] Jakob Foerster, Nantas Nardelli, Gregory Farquhar, Triantafyllos Afouras, Philip HS Torr, Pushmeet Kohli, and Shimon Whiteson. Stabilising experience replay for deep multi-agent reinforcement learning. In *International conference on machine learning*, pages 1146–1155. PMLR, 2017.

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

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

[16] Chuangchuang Sun, Macheng Shen, and Jonathan P How. Scaling up multiagent reinforcement learning for robotic systems: Learn an adaptive sparse communication graph. In *2020 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 11755–11762. IEEE, 2020.

[17] Danijar Hafner, Timothy Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models. *arXiv preprint arXiv:2010.02193*, 2020.

[18] Sanmit Narvekar, Bei Peng, Matteo Leonetti, Jivko Sinapov, Matthew E Taylor, and Peter Stone. Curriculum learning for reinforcement learning domains: A framework and survey. *Journal of Machine Learning Research*, 21(181):1–50, 2020.

[19] Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. When to trust your model: Model-based policy optimization. *Advances in neural information processing systems*, 32, 2019.

[20] Wu Yichao, Wang Yirui, Ding Panpan, Wang Hailong, Zhu Bingqian, and Liu Chun. Enhancing security in deep reinforcement learning: A comprehensive survey on adversarial attacks and defenses. *arXiv preprint arXiv:2510.20314*, 2025.

[21] Derrick Mwiti. Real-life applications of reinforcement learning. *Message posted to http://neptune. ai*, 10.

[22] Uicheon Lee, Seonah Lee, and Kyonghoon Kim. Data-efficient reinforcement learning framework for autonomous flight based on real-world flight data. *Drones*, 9(4):264, 2025.

[23] Junchao Fan, Xuyang Lei, Xiaolin Chang, Jelena Miši, Vojislav B Miši, and Yingying Yao. Less is more: A stealthy and efficient adversarial attack method for drl-based autonomous driving policies. *IEEE Internet of Things Journal*, 2025.

[24] Jung In Kim, Young Jae Lee, Jongkook Heo, Jinhyeok Park, Jaehoon Kim, Sae Rin Lim, Jinyong Jeong, and Seoung Bum Kim. Sample-efficient multi-agent reinforcement learning with masked reconstruction. *PloS one*, 18(9):e0291545, 2023.

[25] Jasmine Jerry Aloor, Siddharth Nagar Nayak, Sydney Dolan, and Hamsa Balakrishnan. Cooperation and fairness in multi-agent reinforcement learning. *Journal on Autonomous Transportation Systems*, 2(2):1–25, 2024.

[26] Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.

[27] Shariq Iqbal and Fei Sha. Actor-attention-critic for multi-agent reinforcement learning. In *International conference on machine learning*, pages 2961–2970. PMLR, 2019.

[28] Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network distillation. *arXiv preprint arXiv:1810.12894*, 2018.

[29] Richard S Sutton, Andrew G Barto, et al. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

[30] Chen Tang, Ben Abbatematteo, Jiaheng Hu, Rohan Chandra, Roberto Martín-Martín, and Peter Stone. Deep reinforcement learning for robotics: A survey of real-world successes. *Annual Review of Control, Robotics, and Autonomous Systems*, 8(1):153–188, 2025.

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

[32] Konstantinos Souchleris, George K Sidiropoulos, and George A Papakostas. Reinforcement learning in game industry—review, prospects and challenges. *Applied Sciences*, 13(4):2443, 2023.

[33] William Fedus, Prajit Ramachandran, Rishabh Agarwal, Yoshua Bengio, Hugo Larochelle, Mark Rowland, and Will Dabney. Revisiting fundamentals of experience replay. In *International conference on machine learning*, pages 3061–3071. PMLR, 2020.

[34] Alekh Agarwal, Sham Kakade, Akshay Krishnamurthy, and Wen Sun. Flambe: Structural complexity and representation learning of low rank mdps. *Advances in neural information processing systems*, 33:20095–20107, 2020.

[35] Fan-Ming Luo, Tian Xu, Hang Lai, Xiong-Hui Chen, Weinan Zhang, and Yang Yu. A survey on model-based reinforcement learning. *Science China Information Sciences*, 67 (2):121101, 2024.

[36] Pranav Rajbhandari, Prithviraj Dasgupta, and Donald Sofge. Transformer guided coevolution: Improved team formation in multiagent adversarial games. In *Proceedings*

of the 24th International Conference on Autonomous Agents and Multiagent Systems, pages 2720–2722, 2025.

[37] Yixiang Wang and Feng Wu. Multi-agent deep reinforcement learning with adaptive policies. *arXiv preprint arXiv:1912.00949*, 2019.

[38] Aryaman Reddi, Maximilian Tölle, Jan Peters, Georgia Chalvatzaki, and Carlo D'Eramo. Robust adversarial reinforcement learning via bounded rationality curricula. *arXiv preprint arXiv:2311.01642*, 2023.

[39] Kamal A ElDahshan, Hesham Farouk, and Eslam Mofreh. Deep reinforcement learning based video games: A review. In *2022 2nd International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC)*, pages 302–309. IEEE, 2022.

[40] Zhang-Wei Hong, Tzu-Yun Shann, Shih-Yang Su, Yi-Hsiang Chang, Tsu-Jui Fu, and Chun-Yi Lee. Diversity-driven exploration strategy for deep reinforcement learning. *Advances in neural information processing systems*, 31, 2018.

[41] Purushotham Endla, V Thrimurthulu, Dokku Durga Bhavani, Mohit Tiwari, et al. Deep reinforcement learning for real-time strategy games techniques and open challenges. In *ITM Web of Conferences*, volume 76, page 01006. EDP Sciences, 2025.

[42] Leander Diaz-Bone, Marco Bagatella, Jonas Hübotter, and Andreas Krause. Discover: Automated curricula for sparse-reward reinforcement learning. *arXiv preprint arXiv:2505.19850*, 2025.

[43] Yan Kong, Junfeng Wei, and Chih-Hsien Hsia. Solving sparse reward tasks using self-balancing exploration and exploitation. *Journal of Internet Technology*, 26(3):293–301, 2025.

[44] Derek Yadgaroff, Alessandro Sestini, Konrad Tollmar, Ayca Ozcelikkale, and Linus Gisslén. Improving generalization in game agents with data augmentation in imitation learning. In *2024 IEEE Congress on Evolutionary Computation (CEC)*, pages 1–8. IEEE, 2024.

[45] Scott Goodfriend. A competition winning deep reinforcement learning agent in microrts. In *2024 IEEE Conference on Games (CoG)*, pages 1–8. IEEE, 2024.

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

[47] Hongye Cao, Fan Feng, Jing Huo, Shangdong Yang, Meng Fang, Tianpei Yang, and Yang Gao. Model-based offline reinforcement learning with adversarial data augmentation. *arXiv preprint arXiv:2503.20285*, 2025.

[48] Tao Yu, Cuiling Lan, Wenjun Zeng, Mingxiao Feng, Zhizheng Zhang, and Zhibo Chen. Playvirtual: Augmenting cycle-consistent virtual trajectories for reinforcement learning. *Advances in Neural Information Processing Systems*, 34:5276–5289, 2021.

[49] José Suárez-Varela, Albert Mestres, Junlin Yu, Li Kuang, Haoyu Feng, Albert Cabellos-Aparicio, and Pere Barlet-Ros. Routing in optical transport networks with deep reinforcement learning. *Journal of Optical Communications and Networking*, 11(11):547–558, 2019.

[50] Hanhua Zhu. Generalized representation learning methods for deep reinforcement learning. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pages 5216–5217, 2021.

[51] Yi Yan, Chunsheng Liu, Faliang Chang, and Yiming Huang. An improved proximal policy optimization algorithm for autonomous driving decision-making. In *Fourth International Conference on Sensors and Information Technology (ICSI 2024)*, volume 13107, pages 837–845. SPIE, 2024.

[52] Fabian Otto, Philipp Becker, Vien Anh Ngo, and Gerhard Neumann. Efficient off-policy learning for high-dimensional action spaces. In *The Thirteenth International Conference on Learning Representations*.

[53] Macheng Shen. *Robust and Scalable Multiagent Reinforcement Learning in Adversarial Scenarios*. PhD thesis, Massachusetts Institute of Technology, 2022.

[54] Rundong Wang, Runsheng Yu, Bo An, and Zinovi Rabinovich. I2hrl: Interactive influence-based hierarchical reinforcement learning. In *Proceedings of the twenty-ninth*

international conference on international joint conferences on artificial intelligence, pages 3131–3138, 2021.

[55] Zheng Tian, Hang Ren, Yaodong Yang, Yuchen Sun, Ziqi Han, Ian Davies, and Jun Wang. Learning to safely exploit a non-stationary opponent. 2022.

[56] Dennis Lee, Haoran Tang, Jeffrey Zhang, Huazhe Xu, Trevor Darrell, and Pieter Abbeel. Modular architecture for starcraft ii with deep reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, volume 14, pages 187–193, 2018.

[57] Ruozi Huang, Xipeng Wu, Hongsheng Yu, Zhong Fan, Haobo Fu, Qiang Fu, and Wei Yang. A robust and opponent-aware league training method for starcraft ii. *Advances in Neural Information Processing Systems*, 36:47554–47574, 2023.

[58] Andries Smit, Herman A Engelbrecht, Willie Brink, and Arnu Pretorius. Scaling multi-agent reinforcement learning to full 11 versus 11 simulated robotic football. *Autonomous Agents and Multi-Agent Systems*, 37(1):20, 2023.

[59] Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, et al. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839):604–609, 2020.

[60] Vladimir Egorov and Aleksei Shpilman. Scalable multi-agent model-based reinforcement learning. *arXiv preprint arXiv:2205.15023*, 2022.

[61] Georgios Papoudakis, Filippos Christianos, Arrasy Rahman, and Stefano V Albrecht. Dealing with non-stationarity in multi-agent deep reinforcement learning. *arXiv preprint arXiv:1906.04737*, 2019.

[62] Roberta Raileanu, Emily Denton, Arthur Szlam, and Rob Fergus. Modeling others using oneself in multi-agent reinforcement learning. In *International conference on machine learning*, pages 4257–4266. PMLR, 2018.

[63] Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. *Advances in neural information processing systems*, 31, 2018.

[64] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48, 2009.

[65] Siyu Lin, Qingwei Mi, and Tianhan Gao. A survey of curriculum learning in deep reinforcement learning. In *2025 IEEE 15th Annual Computing and Communication Workshop and Conference (CCWC)*, pages 01141–01147. IEEE, 2025.

[66] Rémy Portelas, Cédric Colas, Katja Hofmann, and Pierre-Yves Oudeyer. Teacher algorithms for curriculum learning of deep rl in continuously parameterized environments. In *Conference on Robot Learning*, pages 835–853. PMLR, 2020.

[67] Michael Dennis, Natasha Jaques, Eugene Vinitsky, Alexandre Bayen, Stuart Russell, Andrew Critch, and Sergey Levine. Emergent complexity and zero-shot transfer via unsupervised environment design. *Advances in neural information processing systems*, 33:13049–13061, 2020.

[68] Shuang Ao, Tianyi Zhou, Jing Jiang, Guodong Long, Xuan Song, and Chengqi Zhang. Eat-c: environment-adversarial sub-task curriculum for efficient reinforcement learning. In *International Conference on Machine Learning*, pages 822–843. PMLR, 2022.

[69] Tambet Matiisen, Avital Oliver, Taco Cohen, and John Schulman. Teacher–student curriculum learning. *IEEE transactions on neural networks and learning systems*, 31 (9):3732–3740, 2019.

[70] Bowen Baker, Ingmar Kanitscheider, Todor Markov, Yi Wu, Glenn Powell, Bob McGrew, and Igor Mordatch. Emergent tool use from multi-agent autocurricula. In *International conference on learning representations*, 2019.

[71] Junru Sheng, Peng Zhai, Zhiyan Dong, Xiaoyang Kang, Chixiao Chen, and Lihua Zhang. Curriculum adversarial training for robust reinforcement learning. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2022.

[72] Chongzhen Zhang, Jianrui Wang, Gary G Yen, Chaoqiang Zhao, Qiyu Sun, Yang Tang, Feng Qian, and Jürgen Kurths. When autonomous systems meet accuracy and transferability through ai: A survey. *Patterns*, 1(4), 2020.

[73] Jacob Beck, Risto Vuorio, Evan Zheran Liu, Zheng Xiong, Luisa Zintgraf, Chelsea Finn, and Shimon Whiteson. A survey of meta-reinforcement learning. *arXiv e-prints*, pages arXiv–2301, 2023.

[74] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR, 2017.

[75] Kate Rakelly, Aurick Zhou, Chelsea Finn, Sergey Levine, and Deirdre Quillen. Efficient off-policy meta-reinforcement learning via probabilistic context variables. In *International conference on machine learning*, pages 5331–5340. PMLR, 2019.

[76] Zichuan Lin, Garrett Thomas, Guangwen Yang, and Tengyu Ma. Model-based adversarial meta-reinforcement learning. *Advances in Neural Information Processing Systems*, 33:10161–10173, 2020.

[77] Yan Duan, John Schulman, Xi Chen, Peter L Bartlett, Ilya Sutskever, and Pieter Abbeel. Rl ˆ2: Fast reinforcement learning via slow reinforcement learning. *arXiv preprint arXiv:1611.02779*, 2016.

[78] Deheng Ye, Guibin Chen, Wen Zhang, Sheng Chen, Bo Yuan, Bo Liu, Jia Chen, Zhao Liu, Fuhao Qiu, Hongsheng Yu, et al. Towards playing full moba games with deep reinforcement learning. *Advances in Neural Information Processing Systems*, 33:621–632, 2020.

[79] N Rajasekhar, TK Radhakrishnan, and N Samsudeen. Exploring reinforcement learning in process control: a comprehensive survey. *International Journal of Systems Science*, pages 1–30, 2025.

[80] Jan Blumenkamp and Amanda Prorok. The emergence of adversarial communication in multi-agent reinforcement learning. In *Conference on Robot Learning*, pages 1394–1414. PMLR, 2021.

[81] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507, 2006.

[82] Michael Laskin, Aravind Srinivas, and Pieter Abbeel. Curl: Contrastive unsupervised representations for reinforcement learning. In *International conference on machine learning*, pages 5639–5650. PMLR, 2020.

[83] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378:686–707, 2019.

[84] Owen Lockwood and Mei Si. A review of uncertainty for deep reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, volume 18, pages 155–162, 2022.

[85] Agent Reinforcement Learning. Ensemble value functions for efficient exploration in multi-agent reinforcement learning.

[86] Xinyu Zhao, Jianxiang Liu, Faguo Wu, Xiao Zhang, and Guojian Wang. Uncertainty modified policy for multi-agent reinforcement learning. *Applied Intelligence*, 54(22): 12020–12034, 2024.

[87] Jasmine Bayrooti, Carl Henrik Ek, and Amanda Prorok. Efficient model-based reinforcement learning through optimistic thompson sampling. *arXiv preprint arXiv:2410.04988*, 2024.

[88] Hongming Zhang and Tianyang Yu. Alphazero. In *Deep reinforcement learning: fundamentals, research and applications*, pages 391–415. Springer, 2020.

[89] Xinran Liang, Katherine Shu, Kimin Lee, and Pieter Abbeel. Reward uncertainty for exploration in preference-based reinforcement learning. *arXiv preprint arXiv:2205.12401*, 2022.

[90] Rein Houthooft, Xi Chen, Yan Duan, John Schulman, Filip De Turck, and Pieter Abbeel. Vime: Variational information maximizing exploration. *Advances in neural information processing systems*, 29, 2016.

[91] Chao Yu, Akash Velu, Eugene Vinitsky, Jiaxuan Gao, Yu Wang, Alexandre Bayen, and Yi Wu. The surprising effectiveness of ppo in cooperative multi-agent games. *Advances in neural information processing systems*, 35:24611–24624, 2022.

[92] Tabish Rashid, Mikayel Samvelyan, Christian Schroeder De Witt, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. Monotonic value function factorisation for deep multi-agent reinforcement learning. *Journal of Machine Learning Research*, 21(178): 1–51, 2020.

[93] Mikayel Samvelyan, Akbir Khan, Michael Dennis, Minqi Jiang, Jack Parker-Holder, Jakob Foerster, Roberta Raileanu, and Tim Rocktäschel. Maestro: Open-ended environment design for multi-agent reinforcement learning. *arXiv preprint arXiv:2303.03376*, 2023.

[94] Lerrel Pinto, James Davidson, Rahul Sukthankar, and Abhinav Gupta. Robust adversarial reinforcement learning. In *International conference on machine learning*, pages 2817–2826. PMLR, 2017.

[95] Brennen Hill. Learning an adversarial world model for automated curriculum generation in marl. *arXiv e-prints*, pages arXiv–2509, 2025.

[96] Seungjae Lee, Daesol Cho, Jonghae Park, and H Jin Kim. Cqm: Curriculum reinforcement learning with a quantized world model. *Advances in Neural Information Processing Systems*, 36:78824–78845, 2023.

[97] Ken Peffers, Tuure Tuunanen, Marcus A Rothenberger, and Samir Chatterjee. A design science research methodology for information systems research. *Journal of management information systems*, 24(3):45–77, 2007.

[98] Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *International conference on machine learning*, pages 2778–2787. PMLR, 2017.