Social Learning via Multi Agent Reinforcement Learning

Master Thesis Presented to the
Department of Economics at the
Rheinische Friedrich-Wilhelms-Universität Bonn

In Partial Fulfillment of the Requirements for the Degree of
Master of Science (M.Sc.)

Supervisor: Prof. Dr. Florian Brandl

Submitted in June, 2025 by:
Ege Can Doğaroğlu
Matriculation Number: 3464688

Abstract

This thesis bridges economic social learning theory and multi-agent reinforcement learning by introducing Partially Observable Active Markov Games (POAMGs)—a novel framework for modeling sophisticated learning dynamics in environments with partial observability. Traditional economic models provide valuable theoretical insights but often rely on simplified assumptions about agents' behavior or the learning environment, while reinforcement learning approaches offer computational tools that frequently lack theoretical foundations for strategic adaptation. Our framework addresses three fundamental challenges: modeling strategic adaptation in non-stationary environments, incorporating partial observability as an intrinsic feature, and capturing long-term strategic considerations in learning contexts. We develop POLARIS (Partially Observable Learning with Active Reinforcement In Social environments), a practical algorithm implementing our theoretical framework, and apply it to two canonical social learning scenarios: strategic experimentation with observable rewards and learning without experimentation through action observation. Our results validate theoretical predictions about ... while demonstrating how reinforcement learning agents can discover sophisticated strategies for information revelation and strategic influence, providing both theoretical insights and computational tools for understanding complex social learning dynamics.

Contents

1 Introduction **1**

2 Related Work **5**

2.1 Social Learning . . . 5

2.1.1 Strategic Experimentation . . . 5

2.1.2 Learning Without Experimentation . . . 8

2.1.3 Non-Bayesian Learning . . . 10

2.2 Multi-Agent Reinforcement Learning . . . 10

2.3 Bridging Economic Theory and MARL . . . 11

3 Theoretical Framework **13**

3.1 Partially Observable Active Markov Games . . . 13

3.2 Convergence . . . 14

3.3 Incentives . . . 15

3.4 Belief-Based Policy Gradients . . . 16

3.5 Discounted Returns . . . 18

3.5.1 Discounted Visitation Measure . . . 19

3.5.2 Policy Gradient Theorem for Discounted Returns . . . 20

3.6 Equilibrium . . . 21

3.7 Challenges with Exact Computation . . . 22

3.8 Algorithm: POLARIS . . . 23

3.8.1 Belief Processing Module . . . 23

3.8.2 Inference Learning Module . . . 24

3.8.3 Reinforcement Learning Module . . . 25

3.8.4 Training Process . . . 28

4 Results **31**

4.1 Strategic Experimentation . . . 31

4.2 Learning Without Experimentation . . . 33

Appendices **35**

A Reinforcement Learning Background **35**

A.1 Reinforcement Learning Foundations . . . 35

A.2 Multi-Agent Reinforcement Learning: Concepts and Challenges . . . 36

A.3 The Non-Stationarity Challenge . . . 36

A.4 Traditional Approaches to Non-Stationarity . . . 37

A.4.1 Independent Learning . . . 37
A.4.2 Centralized Training with Decentralized Execution . . . 38
A.4.3 Opponent Modeling and Population-Based Training . . . 38
A.4.4 Equilibrium Learning and Stability Concepts . . . 38
A.4.5 Meta-Learning for Non-Stationarity . . . 39
A.5 Active Markov Games . . . 39
A.5.1 Formal Definition . . . 39
A.5.2 Augmented Transition Function and Stationarity . . . 40
A.5.3 Theoretical Properties and Convergence . . . 40
A.5.4 Practical Implementations . . . 42
A.6 Extension to Partial Observability . . . 42

B Proofs Regarding Average Returns **44**
B.1 Markov Property of the Joint Process . . . 44
B.2 Convergence . . . 45
B.3 Policy Gradient Theorem . . . 47

C Proofs Regarding Discounted Returns **51**
C.1 Transition Operator and Its Adjoint . . . 51
C.1.1 Definitions and Duality . . . 51
C.1.2 Properties of the Operators . . . 52
C.1.3 Contraction and Propagation . . . 55
C.2 Bellman Equations and Value Functions . . . 57
C.2.1 Well-Definedness of Value Functions . . . 57
C.2.2 Bellman Equations for Discounted Value Functions . . . 58
C.2.3 Policy Gradient with Respect to Value Function . . . 60
C.3 Discounted Visitation Measure . . . 61
C.3.1 Definition and Properties . . . 61
C.3.2 Existence and Uniqueness . . . 63
C.3.3 Connection to Value Function . . . 65
C.4 Policy Gradient Theorem . . . 67

D POLARIS Architecture **71**
D.1 Belief Processing Module . . . 71
D.2 Inference Learning Module . . . 72
D.3 Reinforcement Learning Module . . . 74
D.3.1 Policy Network Architecture . . . 74
D.3.2 Value Network Architecture . . . 75
D.4 Elastic Weight Consolidation . . . 76
D.5 Implementation Details . . . 77

E Social Learning Implementation **79**
E.1 Lévy Process Discretization 79
E.1.1 Mathematical Foundations of Lévy Processes 79
E.1.2 Time Discretization of Lévy Processes 80
E.1.3 Implementing Strategic Experimentation Models 81
E.1.4 Relation to Policy-Invariant Reward Transformations 83
E.2 Observed Reward Function 84

References **87**

1 INTRODUCTION

The study of how individuals and groups learn from each other's actions and experiences has been a central focus in economic theory for decades. From the seminal contributions on information cascades by **?** and **?** to more recent explorations of learning in networks (**??**), economic research has sought to understand how social interactions shape beliefs, decisions, and collective outcomes. Traditional economic models have provided valuable insights into phenomena such as herding behavior, where individuals rationally ignore their private information to follow the observed actions of predecessors. However, these models often rely on simplified assumptions about agents' behavior and reasoning processes, frequently restricting analysis to one-shot sequential decisions or steady-state equilibria rather than capturing the rich dynamics of repeated interactions where agents continuously adapt their strategies based on others' evolving behaviors.

Meanwhile, the field of multi-agent reinforcement learning (MARL) has experienced remarkable progress in developing algorithms that enable autonomous agents to learn effective strategies in complex, interactive environments. Recent advances in reinforcement learning have demonstrated impressive capabilities in strategic games (**?**), cooperative problem-solving (**?**), and competitive scenarios (**?**). MARL offers powerful tools for modeling adaptive behavior in non-stationary environments where agents must continuously revise their strategies in response to others' changing behaviors. However, much of this work has focused on engineering objectives rather than modeling realistic human learning processes, limiting its applicability to fundamental questions in economic theory. Additionally, many MARL approaches struggle with the challenge of partial observability—a defining feature of social learning contexts where agents cannot directly observe others' private information or belief states.

This thesis bridges these domains by introducing a novel framework that integrates economic social learning theory with multi-agent reinforcement learning under partial observability. Our approach models social learning as a dynamic process where agents with limited information strategically adapt their behaviors while simultaneously learning about their environment and other agents' strategies. By formalizing this process through the lens of partially observable reinforcement learning, we aim to reveal new insights into how strategic adaptation influences collective outcomes and equilibrium behavior. This interdisciplinary synthesis addresses fundamental challenges that have constrained previous work in both economics and artificial intelligence, offering a more realistic and computationally tractable approach to modeling complex social learning dynamics.

The first challenge concerns non-stationarity and strategic adaptation. Traditional economic models of social learning typically assume that agents follow fixed, myopic decision rules or Bayesian updating procedures that do not fully account for strategic adaptation over time. When

agents repeatedly interact and observe each other's actions, however, they may adjust their strategies in anticipation of others' learning, creating a complex web of interdependent adaptation that fundamentally alters collective learning dynamics. This non-stationarity—where the effective environment an agent faces changes as other agents learn—represents a central challenge that requires new modeling approaches. Standard MARL algorithms often treat this non-stationarity as a technical obstacle to convergence rather than an intrinsic feature of multi-agent systems that agents should explicitly reason about and strategically exploit.

The second challenge involves partial observability, which pervades social learning contexts. In most real-world settings, agents cannot directly observe others' private information, beliefs, or decision processes. Instead, they must infer these hidden states from observable actions and outcomes. This partial observability compounds the complexity of strategic interaction, as agents must simultaneously learn about the underlying state of the world and about the belief formation processes of others. Economic models have often simplified this challenge through strong assumptions about common knowledge or by focusing on one-shot sequential decisions rather than repeated interactions. Similarly, many MARL approaches assume full observability of the state or treat partial observability as a technical challenge to be addressed through belief state tracking, without fully incorporating its strategic implications.

The third challenge concerns long-term strategic considerations in social learning. Agents often face a tension between immediate payoffs and long-term information generation. They may sometimes choose actions that appear suboptimal in the short term to influence the learning trajectories of others or to generate valuable information for future periods. For instance, in strategic experimentation contexts, agents might incur costs to explore unknown options, knowing that the resulting information will benefit both themselves and others in the future. Capturing these farsighted strategic considerations requires models that explicitly account for how current actions shape the future evolution of others' beliefs and behaviors—a capability that neither traditional economic models nor standard MARL algorithms fully provide.

To address these challenges, this thesis makes several interconnected contributions. First, we develop Partially Observable Active Markov Games (POAMGs)—a novel formalism that extends the Active Markov Game framework introduced by **?** to partially observable settings. Unlike standard reinforcement learning frameworks that treat non-stationarity as a challenge to be mitigated, POAMGs incorporate policy evolution as an integral part of the environment dynamics, allowing agents to reason about and strategically influence this evolution process. POAMGs explicitly model how agents' policies evolve over time based on observations and interactions, while accounting for the fundamental constraints imposed by partial observability. This formalism provides a mathematically rigorous foundation for analyzing complex social learning dynamics that involve repeated interactions, adaptive strategies, and incomplete information.

Second, we provide a thorough theoretical analysis of convergence and equilibrium properties in POAMGs. We establish conditions under which the joint process of states, beliefs, and policy parameters converges to a unique stochastically stable distribution, ensuring that our

models have well-defined limiting behavior despite the inherent non-stationarity of multi-agent learning. We derive policy gradient theorems for average and discounted reward objectives, providing a solid foundation for algorithm development. We also extend our framework to continuous-time dynamics through stochastic differential equations, enabling a wider range of applications. Our analysis illuminates the relationship between traditional game-theoretic equilibrium concepts and the more general notion of active equilibrium that emerges in our framework, offering new insights into equilibrium selection and stability in social learning contexts.

Third, we introduce POLARIS (Partially Observable Learning with Active Reinforcement In Social environments), a practical algorithm for learning in POAMGs. POLARIS extends previous algorithms to partially observable settings through an integrated architecture with three core components: a belief processing module that tracks agents' information states using Transformer models; an inference learning module that predicts the policy evolution of other agents through variational methods; and a reinforcement learning module that optimizes policies based on average or discounted reward criteria, supporting both discrete and continuous action spaces. This integrated approach enables agents to learn sophisticated strategies that account for both the partial observability of the environment and the learning dynamics of other agents—capabilities that are essential for modeling realistic social learning processes.

Fourth, we apply our framework to two canonical social learning scenarios from economic theory. The first application focuses on strategic experimentation, building on the models of **?** and **?**, where agents learn optimal actions through trial and error with observable rewards. The second application examines learning without experimentation, extending the frameworks of **?** and **?**, where agents form beliefs primarily by observing others' actions rather than receiving direct feedback from the environment. Our analysis demonstrates how our approach can model phenomena such as free-riding on others' information production, strategic teaching, information cascades, and confounded learning under more realistic assumptions about agents' reasoning capabilities and strategic sophistication.

Through these applications, we illustrate how bridging economic theory and MARL can enhance our understanding of social learning processes. Our POAMG framework provides a more realistic model of how partially rational agents learn from each other in complex, partially observable environments. At the same time, it offers practical algorithms for computing equilibrium strategies that traditional economic approaches might find intractable. The POLARIS algorithm demonstrates how techniques from deep reinforcement learning and approximate inference can be adapted to tackle the specific challenges of social learning, offering a computationally feasible method for simulating and analyzing complex multi-agent learning dynamics.

The remainder of this thesis is organized as follows: Chapter 2 reviews the relevant literature on social learning in economic theory and multi-agent reinforcement learning, emphasizing the complementary strengths and limitations of these approaches and identifying opportunities for cross-fertilization. Chapter 3 introduces our theoretical framework, developing the formalism of Partially Observable Active Markov Games and deriving key results on convergence, policy gradients, and equilibrium concepts, as well as presenting the POLARIS algorithm, detailing

its components, implementation considerations, and theoretical justifications. Chapter 4 applies our framework to strategic experimentation and learning without experimentation, demonstrating how POLARIS captures sophisticated social learning dynamics in these contexts. Chapter 5 concludes with a discussion of implications, limitations, and directions for future research.

By integrating perspectives from economic theory and multi-agent reinforcement learning, this thesis aims to enhance our understanding of social learning processes and provide new tools for modeling complex strategic interactions in partially observable environments. The insights gained may not only advance theoretical knowledge but also inform the design of platforms and institutions that facilitate efficient collective learning and decision-making in diverse contexts, from financial markets and organizational learning to online social networks and collaborative scientific endeavors. Moreover, by demonstrating the value of MARL techniques for addressing classical questions in economic theory, we hope to encourage further cross-disciplinary work that leverages complementary insights from these rich intellectual traditions.

In the next chapter, we provide a detailed review of the relevant literature, tracing the development of social learning models in economics and recent advances in multi-agent reinforcement learning. This review establishes the conceptual foundation for our theoretical framework and highlights the specific gaps that our approach aims to address, setting the stage for the formal development of Partially Observable Active Markov Games in Chapter 3.

2 RELATED WORK

This section reviews key literature across economic social learning and multi-agent reinforcement learning (MARL), highlighting how our approach bridges these fields to address their respective limitations.

2.1 SOCIAL LEARNING

The economic study of social learning originated with **?** and **?**, who formalized how rational agents might ignore private information to follow predecessors' actions, leading to information cascades and potentially inefficient herding. While foundational, these models rely on one-shot sequential decisions rather than the repeated interactions that characterize many real-world learning contexts. **?** extended this work by showing how heterogeneous preferences can lead to confounded learning, where private signals remain relevant despite observing others' actions.

Social learning in network settings expands this framework by examining how network structure influences information flow (**??**). Two principal approaches have emerged: Bayesian models where agents perform rational inference (**???**), and non-Bayesian models (**??**) like the DeGroot framework (**?**) where agents update beliefs through weighted averaging of neighbors' opinions. A critical insight from this literature is that network topology significantly affects learning outcomes. However, most network models still fall short in capturing how agents strategically adapt to others' evolving learning behaviors over time—a key element of our framework. subsectionWhile early models focused on one-shot decisions, more recent work has explored repeated interactions where agents continuously adapt strategies based on observations. These settings more closely align with our MARL approach and will be the focus of our implementation.

2.1.1 Strategic Experimentation

The strategic experimentation literature examines settings where agents balance exploiting current knowledge against generating new information through exploration (**??**). This creates a dynamic tension between individual incentives to free-ride on others' information production and collective benefits from experimentation. Strategic experimentation represents a fundamental departure from classical social learning models by explicitly accounting for the intertemporal nature of information acquisition. Unlike cascade models where agents make one-shot decisions in sequence, agents in strategic experimentation scenarios face repeated opportunities to learn and adapt their strategies over time. This dynamic perspective connects directly to the reinforcement learning paradigm, where exploration-exploitation tradeoffs are central (**?**). These

tradeoffs create the necessary tension for social influence, even in the absence of informational asymmetry among agents that is often central to social learning models (**?**).

The economic foundations of strategic experimentation were established by **?**, who analyzed how a monopolist might experiment with different prices to learn about demand. This concept was extended to multi-agent settings by **?**, who developed a framework for analyzing experimentation in teams. Their seminal work revealed that when information is a public good, free-riding incentives can significantly reduce aggregate experimentation below socially optimal levels, creating a classic collective action problem.

Several extensions have explored how different information structures affect experimentation incentives. **?** introduced exponential bandits, where lump-sum rewards arrive according to a Poisson process, demonstrating how the resolution of uncertainty affects the dynamics of experimentation. Their model showed that "encouragement effects" can arise, where agents experiment more intensively to motivate others to join the exploration effort. **?** demonstrated how negatively correlated bandits—where success on one experiment decreases the estimated value of others—can encourage more efficient experimentation patterns.

? developed a particularly relevant model using average reward criteria rather than discounted objectives. Under this framework, the value of information doesn't decay over time, incentivizing different patterns of exploration. This approach aligns with our POAMG framework's emphasis on long-term strategic adaptation in multi-agent systems. **?** further demonstrated how private observations can restore experimentation incentives that fail under public observations, providing insights into how information asymmetry affects collective learning dynamics.

The strategic teaching phenomenon, where agents take seemingly suboptimal actions to influence others' beliefs, emerges naturally in these contexts. **?** demonstrated how sophisticated agents might deliberately punish or reward others via continuation payoffs due to invariance property of the payoff sets. Similarly, **?** showed how optimal incentive structures might intentionally incentivize agents to experiment, highlighting the importance of mechanism design in collective learning environments.

The literature on information design (**??**) provides complementary insights by examining how information revelation mechanisms affect experimentation decisions. **?** showed how over-revealing information can sometimes incentivize more experimentation than full transparency. These insights connect directly to the strategic influence aspects of our POAMG framework, which explicitly models how agents reason about and deliberately influence others' learning trajectories.

Formal Model

More formally, we can describe the strategic experimentation literature through the model developed by **?**. In this framework, n agents face a two-armed bandit problem where they continuously allocate resources between a safe arm with known deterministic payoff $r_{safe} > 0$ and a risky arm whose expected payoff depends on an unknown state $\omega \in \{0, 1, \ldots, m\}$ with $m \geq 1$.

The state is drawn at the beginning according to a publicly known prior distribution with full support. The risky arm generates payoffs according to a Lévy process:

$$X_t^i = \alpha_\omega t + \sigma Z_t^i + Y_t^i$$

where Z^i is a standard Wiener process, Y^i is a compound Poisson process with Lévy measure ν_ω, and α_ω is the drift rate in state ω. The expected payoff per unit of time is $r_\omega = \alpha_\omega + \lambda_\omega h_\omega$, where λ_ω is the expected number of jumps per unit of time and h_ω is the expected jump size.

At each moment, agent i allocates fraction $a_t^i \in [0,1]$ to the risky arm, yielding instantaneous expected payoff:

$$(1-a_t^i)r_{safe} + a_t^i r_\omega$$

Each agent observes their own payoff process, the payoff processes of all other agents, and potentially a background information process B_t, which provides free information about the state. This background process follows the same structure as the payoff processes:

$$B_t = \beta_\omega t + \sigma_B Z_t^B + Y_t^B$$

whose informativeness is given exogenously as k_0 which increases linearly in time. Under the strong long-run average criterion, agents maximize:

$$\lim_{T\to\infty} \mathbb{E}\left[\frac{1}{T}\int_0^T \{(1-a_t^i)r_{safe} + a_t^i r_\omega\}dt\right] \tag{2.1}$$

The unique symmetric Markov perfect equilibrium strategy is characterized as:

$$a^*(b) = \begin{cases} 0 & \text{if } I(b) \le k_0, \\ \frac{I(b)-k_0}{n-1} & \text{if } k_0 < I(b) < k_0 + n - 1, \\ 1 & \text{if } I(b) \ge k_0 + n - 1. \end{cases} \tag{2.2}$$

where $I(b)$ corresponds to the incentive to experiment, defined as:

$$I(b) = \frac{f(b) - r_{safe}}{r_{safe} - m(b)}$$

when $m(b) < s$, and $I(b) = \infty$ otherwise, where $f(b)$ is the expected flow payoff under full information and $m(b)$ is the expected flow payoff from the risky arm.

This formulation directly captures the tension between exploitation (choosing the currently optimal action) and exploration (generating information for future decisions), as well as the free-rider problem that arises when information is a public good. While strategic experimentation models capture important elements of learning dynamics, they typically assume perfectly rational agents and often rely on standardized information structures.

2.1.2 Learning Without Experimentation

A complementary strand of research examines settings where agents learn without directly observing rewards. **?** studied how long-lived agents learn in networks through repeated interactions, revealing a fundamental inefficiency: regardless of network size, learning speed remains bounded by a constant dependent only on private signal distributions. **?** extended these results by showing that this limitation doesn't apply uniformly to all agents, constructing scenarios where some agents learn faster at others' expense.

Unlike strategic experimentation models where agents receive direct payoff feedback, learning without experimentation captures scenarios where agents must form beliefs based primarily on others' observed actions. This distinction is crucial for modeling many real-world social learning contexts, from financial markets (**?**) to technology adoption (**?**), where payoffs are delayed, noisy, or unobservable.

The theoretical foundations for this approach draw from both Bayesian and non-Bayesian learning traditions. **?** and **?** developed early models showing how rational agents might become trapped in information cascades when learning from others' actions. **?** extended this analysis to network settings, demonstrating how network topology influences the aggregation of dispersed information.

A substantial literature has explored learning rates in networked environments. **?** show neighborhood structure influences optimal action adoption, while **?** demonstrate that homophily slows consensus convergence without being affected by network density. **?** quantify information loss when observing others' discrete actions, finding only a fraction of private information transmits, approaching zero in large societies due to "groupthink." **?** characterize learning rates through agents' signal structures and eigenvector centralities, showing optimal information allocation depends on its distribution—better information should be placed at central nodes when information structures are comparable, but at peripheral nodes when agents possess unique critical information.

The mechanisms behind these learning barriers stem from information loss in the action quantization process. When continuous beliefs are compressed into discrete actions, information is inherently lost (**?**). **?** characterized this as 'coarse inference' where agents make inferences based only on the aggregate distribution of actions across states rather than on the fine details of how actions depend on specific histories, leading to a loss of information.

Strategic considerations emerge naturally in these settings as agents realize their actions influence the future learning of others. **?** demonstrated how forward-looking agents might distort their actions to manipulate the information revealed to others. **?** showed how agents with private information might strategically time their actions to maximize influence on others' beliefs. These strategic dynamics align closely with the active influence mechanisms in our POAMG framework.

Formal Model

Similar to before, we characterize the learning dynamics in settings without experimentation through the model introduced by **?**. In this framework, a set of $N = \{1,\ldots,n\}$ agents interact over discrete time periods $t \in \{1,2,\ldots\}$ in a fixed social network. The state of the world ω is drawn from a finite set S according to a prior distribution with full support and remains fixed throughout.

At each period t, each agent i receives a private signal o_t^i from a set O, drawn according to a state-dependent public known distribution. Signals are independent across agents and time periods. Each agent then chooses an action a_t^i from a set A, observing the actions taken by neighbors $N^i \subset N$ in previous periods.

All agents share the same utility function $u : S \times A \to \mathbb{R}$, which depends on the state and their own action. For each state s, there is a unique optimal action $a_s = \arg\max_{a \in A} u(s,a)$, and no action is optimal in two different states. Crucially, agents do not observe their realized utilities, eliminating experimentation motives. Agent i's information set at period τ consists of:

$$I^i_{\leq \tau} = (o^i_1,\ldots,o^i_\tau;(a^j_t)_{j\in N^i, t<\tau})$$

A (pure) strategy for agent i is a function σ^i that maps information sets to actions:

$$\sigma^i : \cup_{t=1}^{\infty}(O^t \times A^{N^i \times (t-1)}) \to A$$

For any given strategy profile $\boldsymbol{\sigma} = (\sigma^1,\ldots,\sigma^n)$, the learning rate of agent i is defined as:

$$r^i(\boldsymbol{\sigma}) = \liminf_{t\to\infty} -\frac{1}{t}\log \mathbb{P}(a^i_t \neq a_\omega | \boldsymbol{\sigma})$$

The main theoretical results establish both limitations and opportunities in social learning networks. First, the social learning barrier theorem demonstrates that regardless of network structure or strategies, some agent's learning rate is bounded by a constant that depends only on the signal structure:

$$\min_{i\in n} r^i(\boldsymbol{\sigma}) \leq r_{bdd} = \min_{\theta\neq\theta'}\{D_{KL}(\mu_\theta||\mu_{\theta'}) + D_{KL}(\mu_{\theta'}||\mu_\theta)\}$$

where $D_{KL}(\mu_\theta||\mu_{\theta'})$ is the Kullback-Leibler divergence between signal distributions and the summation corresponds to the Jeffreys divergence **?**. This bound applies regardless of the network size or structure. Second, the coordination benefit theorem establishes that for large enough networks and for any $\varepsilon > 0$, all agents can learn at rates above a certain bound:

$$\min_{i\in n} r^i(\boldsymbol{\sigma}) \geq r_{crd} - \varepsilon$$

where $r_{crd} = \min_{\theta\neq\theta'} D_{KL}(\mu_\theta||\mu_{\theta'})$ is strictly greater than the autarky learning rate that an agent can achieve alone. This theoretical framework illuminates a fundamental challenge in social

learning: information aggregation can fail due to information cascades and herding dynamics, resulting in, as pointed out by the paper, what **?** termed "rational groupthink"—where feedback loops cause agents to persist in incorrect beliefs despite continual information arrival. While this model allows agents to learn from various observation structures, it does not impose full rationality on the agents, a strand of literature that we discuss next.

2.1.3 Non-Bayesian Learning

While most economic models assume fully rational agents, partial rationality perspectives acknowledge cognitive limitations that affect learning. Models like **?**'s hybrid learning rule (combining Bayesian updating with naive averaging) and level-k reasoning (**??**) provide middle grounds between full rationality and simple heuristics.

Evolutionary Game Theory (EGT) also provides a powerful non-Bayesian framework for modeling multi-agent learning dynamics without assuming full rationality. Instead, EGT examines how strategy distributions evolve through selection processes based on relative performance (**?**). This perspective aligns naturally with reinforcement learning's trial-and-error approach. **?** demostrates several important connections between EGT and MARL among which, they relate multi-agent Q-learning to replicator dynamics. This mathematical equivalence provides theoretical insights into MARL convergence and equilibrium properties, helping explain empirical observations in complex social dilemmas (**?**).

Bridging theoretical frameworks and computational implementations, MARL offers a powerful methodology for operationalizing social learning models. MARL provides computational tools that can simulate the intricate dynamics of social learning environments while implementing the strategic adaptations that economic theories recognize as essential but frequently find difficult to compute in complex, realistic scenarios.

2.2 MULTI-AGENT REINFORCEMENT LEARNING

While economic models offer valuable theoretical insights, they often struggle with computational tractability when modeling repeated strategic interactions with partially rational agents. This is where MARL provides complementary tools that address specific limitations of economic approaches. MARL frameworks enable the simulation of complex multi-agent systems where agents learn optimal policies through trial-and-error interactions with their environment and other agents. Unlike traditional economic models that often require closed-form solutions, MARL can handle high-dimensional state spaces, complex agent interactions, and non-stationary dynamics that emerge when multiple agents learn simultaneously.

The core components of MARL include state representations, action spaces, reward functions, and learning algorithms that enable agents to maximize expected cumulative rewards. These components can be tailored to model various aspects of social learning, including partial observability (through belief states), strategic adaptation (through policy gradient methods), and partial rationality (through constrained optimization). Modern MARL approaches incorporate

techniques such as centralized training with decentralized execution, value decomposition, and multi-agent actor-critic methods to address coordination challenges that arise in multi-agent settings. We direct the interested reader to Appendix A for a detailed introduction of reinforcement learning techniques, including formal definitions, algorithms, and theoretical properties.

Economic models typically assume either fully rational agents (in Bayesian frameworks) or overly simplistic learning rules (in behavioral models). MARL offers a middle ground by modeling agents that learn from experience and adapt over time without requiring full rationality. **?** demonstrated that even without explicit programming, reinforcement learning agents can develop sophisticated social learning capabilities that mirror human behavior. This addresses the partial rationality problem by providing computational mechanisms for flexible belief updating based on partial information, learning complex strategies through trial and error, and adapting to non-stationary environments created by other learning agents.

A key limitation of economic social learning models is their difficulty in capturing how agents strategically adapt to others' learning processes. The strategic experimentation literature acknowledges these dynamics but often lacks tractable solutions outside of simplified settings. **?** addressed this by introducing Social Influence as a mechanism in MARL, where agents receive additional reward for causally influencing others' actions. This creates a computational framework for modeling strategic teaching and information revelation—key dynamics in the economic models of **?** and **?**.

More directly relevant to our approach, **?** developed Active Markov Games, which explicitly model how agents reason about and influence the policy evolution of other agents. This formalism allows us to capture the strategic adaptation that economic models identify as important but struggle to compute in complex environments.

Despite these advantages, standard MARL approaches have their own limitations when applied to social learning. Most MARL algorithms assume full observability of state information, while social learning inherently involves partial observability of others' private information and beliefs. Many MARL approaches treat non-stationarity as a technical obstacle rather than a strategic feature to be exploited. Additionally, MARL often lacks the theoretical foundations that economic models provide for understanding equilibrium behavior.

Our POAMG framework extends Active Markov Games to partially observable settings specifically to address these limitations. By incorporating policy evolution as an integral part of the environment dynamics while accounting for partial observability, we provide a computational approach that preserves the strategic sophistication of economic models.

2.3 Bridging Economic Theory and MARL

While economic social learning and MARL have developed largely in parallel, their complementary strengths suggest significant potential for integration. Economic models provide rigorous theoretical foundations for understanding rational behavior, belief formation, and information aggregation in social contexts. However, these models often face computational limitations when addressing complex strategic interactions, especially when agents have heterogeneous

beliefs, partial rationality, or operate in environments with partial observability.

Conversely, MARL offers computational frameworks for modeling adaptive agents in complex, high-dimensional environments. These approaches excel at simulating emergent behaviors and can operate effectively without imposing full rationality assumptions. However, MARL approaches frequently lack the theoretical grounding to interpret equilibrium properties and sometimes overlook the strategic sophistication captured in economic models.

Our research bridges these fields by developing a partially observable active Markov game (POAMG) framework that preserves the strategic considerations central to economic theory while leveraging the computational scalability of MARL. This integration addresses three key challenges:

First, we explicitly incorporate policy evolution dynamics and strategic adaptation as fundamental features rather than technical obstacles. Unlike standard MARL approaches that treat non-stationarity as a problem to overcome, our framework models how sophisticated agents reason about and deliberately influence others' learning trajectories (**?**), similar to the strategic teaching phenomena identified in economic experimentation literature (**?**).

Second, we incorporate partial observability as an intrinsic characteristic of social learning environments. By modeling belief states and observation functions, our approach captures the information asymmetries and strategic uncertainty that economic models identify as crucial determinants of learning outcomes (**??**).

Third, we account for long-horizon strategic planning where agents optimize not just immediate rewards but also their influence on the future learning dynamics of other agents. This aligns with economic perspectives on forward-looking behavior while remaining computationally tractable through reinforcement learning techniques.

The resulting framework enables more realistic modeling of social learning phenomena that resist analysis under either purely economic or purely computational approaches. It combines economic insights on strategic sophistication with MARL's ability to simulate complex adaptive systems, yielding both theoretical insights and practical algorithms for understanding multi-agent learning in partially observable environments.

3 THEORETICAL FRAMEWORK

We adapt the framework in **?** to the partially observable setting and formalize the problem of multi-agent learning as a Partially Observable Active Markov Game with periodic policy updates. Throughout the paper, we will use the bold convention to denote the collection of joint sets, joint variables and joint functions for $i \in I = \{1,\dots,n\}$, where $\boldsymbol{X} := \times_{i\in I} X^i$ for set X^i, $\boldsymbol{x} := \{x_1,\dots,x_n\}$ for variable x^i and $\boldsymbol{G}(\cdot) := \prod_{i\in I} G^i(\cdot)$ for function $G^i(\cdot)$.

3.1 PARTIALLY OBSERVABLE ACTIVE MARKOV GAMES

In social learning environments, agents rarely possess complete information about the underlying state of the world. Instead, they receive private signals—often limited and noisy—that only partially reveal the true state, while also observing the actions of other participants rather than their private information. This fundamental information asymmetry is a cornerstone of economic models of social learning. The challenge of inferring valuable information from others' actions while acknowledging the confounding influence of their own social learning creates rich strategic dynamics that cannot be captured by frameworks assuming full observability. This necessitates extending our theoretical approach to the partially observable setting. Standard approaches to partial observability, such as Partially Observable Markov Decision Processes (POMDPs) and Decentralized POMDPs, address this challenge in single-agent and cooperative multi-agent contexts respectively. However, these frameworks do not adequately capture the strategic nature of policy evolution that characterizes the social learning environments, and neither do they allow for flexible modeling of the reward functions. Building on the Active Markov Game formulation of **?**, we introduce Partially Observable Active Markov Games (POAMGs), which integrate belief state dynamics with evolving policy parameters to model sophisticated social learning dynamics under uncertainty.

Definition 1 (Partially Observable Active Markov Game)**.** *A Partially Observable Active Markov Game is defined as a tuple* $M_n = \langle I, S, \boldsymbol{A}, T, \boldsymbol{O}, \boldsymbol{R}, \boldsymbol{\Theta}, \boldsymbol{U} \rangle$*, where:*

- $I = \{1,\dots,n\}$ *is the set of n agents;*
- S *is the state space, assumed to be discrete and finite;*
- $\boldsymbol{A} = \times_{i\in I} A^i$ *is the joint action space, where* A^i *is the action space of agent i;*
- $T : S \times \boldsymbol{A} \mapsto \Delta(S)$ *is the Markovian state transition function, with* $T(s'|s,\boldsymbol{a})$ *denoting the probability of transitioning to state* s' *after taking joint action* $\boldsymbol{a}$ *in state s;*

- $\boldsymbol{O} = \times_{i \in I} O^i$ *is the joint observation function, with* $O^i : S \times \Omega^i \mapsto \Delta(\Omega^i)$ *denoting the observation function for agent i and* Ω^i *denoting the observation space for agent i;*
- $\boldsymbol{R} = \times_{i \in I} R^i$ *is the joint reward function, with* $R^i : S \times \boldsymbol{A} \mapsto \mathbb{R}$ *denoting the reward function for agent i;*
- $\boldsymbol{\Theta} = \times_{i \in I} \Theta^i$ *is the joint policy parameter space, where* Θ^i *is the policy parameter space for agent i;*
- $\boldsymbol{U} = \times_{i \in I} U^i$ *is the joint Markovian policy update function, with* $U^i : \Theta^i \times O^i \times \boldsymbol{A} \times R^i \times O^i \mapsto \Delta(\Theta^i)$ *denoting the policy update function for agent i.*

This formulation extends the Active Markov Game framework by incorporating observation functions that mediate agents' perceptions of the environment state. Next, we define policies of the agents based on their beliefs about the underlying state of the environment.

Definition 2 (Belief-Based Policy)**.** *Under partial observability, each agent i maintains a belief state* $b_t^i \in B^i$ *at time t, representing its probability distribution over states given its observation history. The policy of agent i is defined as a mapping from belief and parameter spaces to distributions over actions:*

$$\pi^i : B^i \times \Theta^i \mapsto \Delta(A^i) \tag{3.1}$$

where $\pi^i(a^i|b^i;\theta^i)$ *represents the probability of agent i taking action* a^i *given belief state* b^i *and policy parameters* θ^i*.*

In partially observable active Markov games, agents form beliefs to infer the underlying state of the environment. Unlike standard partially observable settings, these states evolve through the dynamically changing policies of the agents in addition to the environmental dynamics. The process unfolds as follows: at time step t, each agent i selects an action at state $s_t \in S$ by sampling from its belief-conditioned policy $a_t^i \sim \pi^i(\cdot|b_t^i;\theta_t^i)$. When all agents act collectively through joint action $\boldsymbol{a_t}$, the environment transitions from s_t to s_{t+1} with probability $T(s_{t+1}|s_t,\boldsymbol{a}_t)$. Subsequently, each agent i receives a reward $r_t^i = R^i(s_t,\boldsymbol{a}_t)$ and adjusts its policy parameters via the update function $U^i(\theta_{t+1}^i|\theta_t^i,\tau_t^i)$, where $\tau_t^i = \{o_t^i,\boldsymbol{a_t},r_t^i,o_{t+1}^i\}$ is the trajectory for agent i at time t consisting of the current observation o_t^i, joint action $\boldsymbol{a_t}$, reward received r_t^i, and the next observation o_{t+1}^i. This adaptive cycle continues until non-stationary policies reach convergence. A key insight is that the joint policy update function $\boldsymbol{U}$ depends on a_t^i, which directly impacts state transitions and rewards, thereby enabling agent i to strategically shape future collective policies through its individual decisions. This explicit modeling of strategic influence constitutes the primary advantage of active Markov games over their stationary counterparts.

3.2 CONVERGENCE

A central question in our analysis is whether the joint process of states, beliefs and policy parameters converges to a well-defined stationary distribution, and under what conditions. Following

?, we establish this connection using the properties of regularly perturbed Markov processes. First, we define the joint process, which operates on the joint space of states, beliefs and policy parameters.

Definition 3 (Joint Process)**.** *In a Partially Observable Active Markov Game, the joint process* $(s_t, \boldsymbol{b}_t, \boldsymbol{\theta}_t)$ *consists of the state* $s_t \in S$*, the joint belief state* $\boldsymbol{b}_t = (b_t^1, \ldots, b_t^n) \in \boldsymbol{B}$ *and joint policy parameters* $\boldsymbol{\theta}_t = (\theta_t^1, \ldots, \theta_t^n) \in \boldsymbol{\Theta}$ *of all agents at time* t*.*

We then make the following assumptions on the subprocesses to ensure the ergodicity of the perturbed joint process.

Assumption 1 (Communicating State Transition)**.** *The state transition* T *is communicating, meaning that for any two states* $s, s' \in S$*, there exists a sequence of joint actions* $\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_k$ *and a sequence of states* $s_1, s_2, \ldots, s_{k-1}$ *such that:*

$$T(s_1|s, \boldsymbol{a}_1) > 0, \quad T(s_2|s_1, \boldsymbol{a}_2) > 0, \quad \ldots, \quad T(s'|s_{k-1}, \boldsymbol{a}_k) > 0 \tag{3.2}$$

Assumption 2 (Communicating Belief-State Process)**.** *The belief-state process is communicating, meaning that for any two belief states* $b^i, b'^i \in B^i$*, there exists a sequence of joint actions* $\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_m$ *and a sequence of observations* $o_1^i, o_2^i, \ldots, o_m^i$ *such that:*

$$\mathbb{P}(b_{t+m}^i = b'^i | b_t^i = b^i, \boldsymbol{a}_t = \boldsymbol{a}_1, o_{t+1}^i = o_1^i, \ldots, \boldsymbol{a}_{t+m-1} = \boldsymbol{a}_m, o_{t+m}^i = o_m^i) > 0 \tag{3.3}$$

Assuming the communicating property of the subprocesses, we can establish convergence of the perturbed joint process to the unique stochastically stable distribution of the unperturbed one, as the perturbation vanishes in the limit.

Theorem 1 (Stochastically Stable Distribution)**.** *Under Assumptions 1 and 2, as* $\varepsilon \to 0$*, the perturbed joint processes defined by* ε*-perturbed policy update functions converge to the unique stochastically stable distribution* μ^* *of the unperturbed joint process* $(s_t, \boldsymbol{b}_t, \boldsymbol{\theta}_t)$*.*

Proof. See Appendix B.1 for a detailed proof. □

This convergence result has profound implications for our framework. By establishing the existence of a unique stochastically stable distribution, we provide a solid theoretical foundation for defining optimization objectives and analyzing equilibrium behavior in partially observable active Markov games. The stochastically stable distribution represents the limiting behavior of the system, independent of initial conditions, allowing us to characterize the long-run outcomes of social learning processes. This property is crucial for developing algorithms that optimize for long-term performance rather than myopic rewards, aligning with our goal of modeling sophisticated social learning dynamics.

3.3 INCENTIVES

We now formalize the optimization objectives for agents in our Partially Observable Active Markov Game framework. In contrast to traditional reinforcement learning settings that typ-

ically employ discounted rewards, we first focus on the average reward criterion as our fundamental optimization objective. The average reward formulation provides several compelling advantages for modeling social learning dynamics. As **?** emphasize, this approach is particularly well-suited for continuing tasks without natural episode boundaries—a characteristic that aligns perfectly with the ongoing nature of social learning interactions. While economic theory has predominantly employed discounted reward objectives due to their mathematical tractability and natural correspondence to time preference in utility maximization (**??**), the average reward paradigm better captures the strategic considerations in our framework. The key advantage of the average reward approach in our context is its emphasis on the limiting behavior of the multi-agent system. When agents engage in repeated strategic interactions over indefinite horizons, their primary concern becomes the long-run system behavior rather than transient dynamics. Moreover, unlike discounted objectives that can induce myopic behavior depending on the discount factor, the average reward formulation naturally encourages agents to consider the permanent effects of their actions on other agents' learning processes—a crucial aspect for accurately modeling the strategic dimensions of social learning.

Definition 4 (Average Reward Objective under Partial Observability)**.** *Each agent $i \in I$ in a Partially Observable Active Markov Game aims to find policy parameters θ^i and update function U^i that maximize its expected average reward $\rho^i \in \mathbb{R}$ based on the joint beliefs $\boldsymbol{b}$:*

$$\begin{aligned}\max_{\theta^i, U^i} \rho^i(\boldsymbol{b}, \boldsymbol{\theta}, \boldsymbol{U}) :&= \max_{\theta^i, U^i} \lim_{T\to\infty} \mathbb{E}\left[\frac{1}{T}\sum_{t=0}^{T} R^i(s_t, \boldsymbol{a}_t) \,\middle|\, \begin{array}{c} \boldsymbol{b}_0 = \boldsymbol{b}, \boldsymbol{\theta}_0 = \boldsymbol{\theta}, \\ \boldsymbol{a}_t \sim \boldsymbol{\pi}(\cdot|\boldsymbol{b_t}; \boldsymbol{\theta_t}), s_{t+1} \sim T(\cdot|s_t, \boldsymbol{a}_t), \\ \boldsymbol{o}_{t+1} \sim \boldsymbol{O}(\cdot|s_{t+1}), \boldsymbol{\theta}_{t+1} \sim \boldsymbol{U}(\cdot|\boldsymbol{\theta_t}, \boldsymbol{\tau_t}) \end{array}\right] \\ &= \max_{\theta^i, U^i} \sum_{s, \boldsymbol{b}, \boldsymbol{\theta}} b^i(s) \mu^*(s, \boldsymbol{b}, \boldsymbol{\theta}) \sum_{\boldsymbol{a}} \boldsymbol{\pi}(\boldsymbol{a}|\boldsymbol{b}; \boldsymbol{\theta}) R^i(s, \boldsymbol{a})\end{aligned} \tag{3.4}$$

This formulation emphasizes that agents aim to maximize their rewards in the limiting behavior of the Markov process, focusing on long-term performance rather than immediate gains. The second equality expresses the objective in terms of the stochastically stable distribution μ^*, weighted by agent's belief, connecting our optimization problem to the theoretical results established earlier. It is worth noting that this definition assumes beliefs, policy parameters, and policy update functions are all publicly observable. Since this assumption rarely holds in practical scenarios, we will implement *variational inference* in our algorithm to enable agents to infer these measures from their partial observations. Next, we derive the policy gradients, which we will be the foundation of our algorithm, allowing the agents to maximize their objectives.

3.4 Belief-Based Policy Gradients

Policy gradient methods are a class of reinforcement learning techniques that directly optimize policy parameters by following the gradient of expected return **?**. Unlike value-based methods, policy gradients explicitly parametrize the policy and update parameters in the direction that im-

proves performance. These methods excel in continuous action spaces and can learn stochastic policies **?**. In partially observable settings, policy gradients require adaptation to handle belief states rather than true states **?**. Modern approaches like Proximal Policy Optimization **?** have further improved stability and sample efficiency of these methods through constrained policy updates.

Our framework extends these concepts to multi-agent settings with non-stationary policies by incorporating the Active Markov Game formulation of **?**. The key innovation in our approach is conditioning value functions not just on belief states, but on the joint space of states, belief states, and policy parameters of all agents. This allows us to explicitly model how an agent's actions influence both the environmental dynamics and the learning processes of other agents.

While our theoretical framework proposes maximizing over both policy parameters θ^i and update functions U^i, this joint optimization presents significant computational challenges. As noted in **?** optimizing over policy update functions essentially constitutes a long-horizon meta-learning problem, which remains computationally intractable for many realistic multi-agent settings. Following their practical approach, we simplify the optimization problem by focusing exclusively on learning optimal fixed-point policies that influence joint policy behavior while using standard update rules:

$$\max_{\theta^i} \rho^i_{\theta^i}(\boldsymbol{b},\boldsymbol{\theta}) := \max_{\theta^i} \lim_{T\to\infty} \mathbb{E}\left[\frac{1}{T}\sum_{t=0}^{T} R^i(s_t,\boldsymbol{a}_t)\,\middle|\, \begin{array}{c} \boldsymbol{b}_0=\boldsymbol{b},\boldsymbol{\theta}_0=\boldsymbol{\theta}, \\ \boldsymbol{a}_t\sim\boldsymbol{\pi}(\cdot|\boldsymbol{b}_t;\boldsymbol{\theta}_t), s_{t+1}\sim T(\cdot|s_t,\boldsymbol{a}_t), \\ \boldsymbol{o}_{t+1}\sim\boldsymbol{O}(\cdot|s_{t+1}),\boldsymbol{\theta}_{t+1}\sim\boldsymbol{U}(\cdot|\boldsymbol{\theta}_t,\boldsymbol{\tau}_t) \end{array}\right] \tag{3.5}$$

This formulation still preserves the essence of our framework—capturing how agent i's actions influence other agents' learning trajectories—while making the optimization problem tractable. Under the stochastically stable distribution discussed earlier, this simplified objective becomes independent of initial conditions, further supporting the practical viability of our approach. Under the average reward formulation, we define the differential value function for agent i as:

$$v^i_{\theta^i}(s,\boldsymbol{b},\boldsymbol{\theta}) = \lim_{T\to\infty} \mathbb{E}\left[\sum_{t=0}^{T}\left(R^i(s_t,\boldsymbol{a}_t)-\rho^i_{\theta^i}\right)\,\middle|\, \begin{array}{c} s_0=s,\boldsymbol{b}_0=\boldsymbol{b},\boldsymbol{\theta}_0=\boldsymbol{\theta}, \\ \boldsymbol{a}_t\sim\boldsymbol{\pi}(\cdot|\boldsymbol{b}_t;\boldsymbol{\theta}_t), s_{t+1}\sim T(\cdot|s_t,\boldsymbol{a}_t), \\ \boldsymbol{o}_{t+1}\sim\boldsymbol{O}(\cdot|s_{t+1}),\boldsymbol{\theta}_{t+1}\sim\boldsymbol{U}(\cdot|\boldsymbol{\theta}_t,\boldsymbol{\tau}_t) \end{array}\right] \tag{3.6}$$

This function represents the expected total difference between future rewards and the average reward $\rho^i_{\theta^i}$ when starting from state s, belief states $\boldsymbol{b}$, and policy parameters $\boldsymbol{\theta}$. It serves as a crucial component in deriving the policy gradient theorem for our framework.

Theorem 2 (Partially Observable Active Average Reward Policy Gradient Theorem)**.** *The gradient of the active average reward objective with respect to agent i's policy parameters θ^i in a*

partially observable setting is:

$$\nabla_{\theta^i} J^i_\pi(\theta^i) = \sum_{s,\boldsymbol{b},\boldsymbol{\theta}} \mu_{\theta^i}(s,\boldsymbol{b},\boldsymbol{\theta}) \sum_{a^i} \nabla_{\theta^i} \pi^i(a^i|b^i;\theta^i) \sum_{a^{\neg i}} \pi^{\neg i}(a^{\neg i}|b^{\neg i};\theta^{\neg i}) q^i_{\theta^i}(s,\boldsymbol{b},\boldsymbol{\theta},\boldsymbol{a}) \tag{3.7}$$

where the action-value function $q^i_{\theta^i}$ is defined as:

$$q^i_{\theta^i}(s,\boldsymbol{b},\boldsymbol{\theta},\boldsymbol{a}) = \sum_{s'} T(s'|s,\boldsymbol{a}) \sum_{\boldsymbol{o}'} \boldsymbol{O}(\boldsymbol{o}'|s') \sum_{\boldsymbol{\theta}'} \boldsymbol{U}(\boldsymbol{\theta}'|\boldsymbol{\theta},\boldsymbol{\tau}) \left[R^i(s,\boldsymbol{a}) - \rho^i_{\theta^i} + v^i_{\theta^i}(s', update(\boldsymbol{b},\boldsymbol{a},\boldsymbol{o}'),\boldsymbol{\theta}') \right] \tag{3.8}$$

with $update(\boldsymbol{b},\boldsymbol{a},\boldsymbol{o}')$ representing the belief update function.

Proof. See Appendix B.3 for a detailed proof. □

This theorem extends the policy gradient result in **?** to partially observable settings by integrating over the joint space of states, beliefs, and policy parameters according to their stochastically stable distribution $\mu_{\theta^i}(s,\boldsymbol{b},\boldsymbol{\theta})$. The resulting gradient has a form similar to the standard policy gradient theorem, but with important modifications. First, our policies are conditioned on belief states rather than actual states. Second, the action-value function must account for belief updates and policy parameter evolution. Third, the expectation is taken over the stochastically stable distribution of the joint state space, while also considering the stochasticity of the observations.

Building on the theoretical foundations established for the average reward criterion, we now turn our attention to the discounted return formulation. This alternative objective function is particularly relevant in economic contexts where agents exhibit time preferences, valuing immediate rewards more highly than delayed ones. The following section develops the mathematical framework for optimizing agent behavior under discounted returns, providing a complementary perspective to our earlier analysis. By examining both criteria, we offer a comprehensive theoretical foundation that can accommodate different modeling assumptions about how agents value future consequences of their actions.

3.5 Discounted Returns

While the average reward criterion provides a principled approach for analyzing limiting behaviors in continuing tasks, the discounted return objective remains predominant in economic models of social learning (**????**). This formulation incorporates time preference through a discount factor $\gamma \in [0,1)$, giving higher weight to near-term rewards and diminishing importance to those further in the future. The expected effective planning horizon under discounting is approximately $\frac{1}{1-\gamma}$ steps (**?**), making it well-suited for scenarios where agents exhibit time preference or where finite planning horizons are appropriate (**??**). We begin by formalizing the discounted return objective in our POAMG framework:

Definition 5 (Discounted Return Objective under Partial Observability)**.** *Each agent $i \in I$ in a Partially Observable Active Markov Game aims to find policy parameters θ^i that maximize its expected discounted return $J^i_{\pi,\gamma}(\theta^i) = v^i_{\theta^i}(s_0, \boldsymbol{b}_0, \boldsymbol{\theta}_0)$ starting from initial state s_0, joint beliefs $\boldsymbol{b}_0$, and joint policy parameters $\boldsymbol{\theta}_0$:*

$$\max_{\theta^i} v^i_{\theta^i}(s_0, \boldsymbol{b}_0, \boldsymbol{\theta}_0) := \max_{\theta^i} \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R^i(s_t, \boldsymbol{a}_t) \,\middle|\, \begin{array}{c} s_0, \boldsymbol{b}_0, \boldsymbol{\theta}_0, \\ \boldsymbol{a}_{0:\infty} \sim \boldsymbol{\pi}(\cdot|\boldsymbol{b}_{0:\infty}; \boldsymbol{\theta}_{0:\infty}), s_{t+1} \sim T(\cdot|s_t, \boldsymbol{a}_t), \\ \boldsymbol{o}_{t+1} \sim \boldsymbol{O}(\cdot|s_{t+1}), \boldsymbol{\theta}_{t+1} \sim \boldsymbol{U}(\cdot|\boldsymbol{\theta}_t, \boldsymbol{\tau}_t) \end{array} \right] \tag{3.9}$$

The discounted return objective differs fundamentally from the average reward criterion in its treatment of future consequences. As γ approaches 1, it increasingly resembles the average reward formulation, but with a crucial distinction: even with γ arbitrarily close to 1, the discounted formulation still assigns diminishing weight to distant future rewards. This property has significant implications for strategic behavior in multi-agent settings, particularly in social learning contexts.

3.5.1 Discounted Visitation Measure

To analyze the behavior of agents operating under discounted returns, we introduce the *discounted visitation measure* (also called occupancy measure). This measure represents the normalized expected discounted time spent in each state-belief-policy configuration when following a joint policy (**??**).

Definition 6 (Discounted Visitation Measure)**.** *For a Markov process governed by a transition operator Ψ and its adjoint Ψ^*, with an initial distribution μ_0 over the joint state-belief-policy space, the discounted visitation measure $d^{\pi}_{\mu_0}$ is defined as:*

$$d^{\pi}_{\mu_0} := (1-\gamma) \sum_{t=0}^{\infty} \gamma^t \mu_t \tag{3.10}$$

where μ_t is the distribution at time t when starting from μ_0, and $\gamma \in [0,1)$ is the discount factor.

This distribution serves two critical purposes in our framework. First, it provides a well-defined distribution over which expectations can be taken, even in non-stationary environments where policies are constantly changing. Unlike the stochastically stable distribution used in the average reward setting, the discounted distribution automatically exists for any $\gamma < 1$ without requiring additional assumptions on the ergodicity of the underlying processes (**?**). Second, it directly links to the optimization objective, allowing us to express the expected discounted return as an expectation of immediate rewards under this distribution.

The discounted visitation measure satisfies important properties that facilitate policy optimization. Most notably, it can be expressed as the unique solution to a functional equation:

$$d^{\pi}_{\mu_0} = (1-\gamma)\mu_0 + \gamma\Psi^* d^{\pi}_{\mu_0} \tag{3.11}$$

where Ψ^* is the adjoint of the transition operator Ψ that describes how probability distributions evolve over one timestep. This relationship can be seen as a fixed-point equation, where $d^{\pi}_{\mu_0}$ is the unique fixed point of the contractive mapping $(1-\gamma)\mu_0 + \gamma\Psi^*(\cdot)$ in the space of finite measures. The existence and uniqueness of this measure is guaranteed for any $\gamma < 1$, providing a solid mathematical foundation for our policy gradient derivations.

3.5.2 Policy Gradient Theorem for Discounted Returns

With the discounted visitation measure established, we now derive the policy gradient theorem that forms the basis for optimization in this framework.

Theorem 3 (Partially Observable Active Discounted Return Policy Gradient Theorem)**.** *The gradient of the discounted return objective $J^i_{\pi,\gamma}(\theta^i) = v^i_{\theta^i}(s_0, \boldsymbol{b}_0, \boldsymbol{\theta}_0)$ with respect to agent i's policy parameters θ^i in a partially observable active Markov game setting can be expressed as:*

$$\begin{aligned}\nabla_{\theta^i} J^i_{\pi,\gamma}(\theta^i) = \frac{1}{1-\gamma} \sum_{s,\boldsymbol{b},\boldsymbol{\theta}} d^{\pi}_{\mu_0}(s, \boldsymbol{b}, \boldsymbol{\theta}) \sum_{a^i} \nabla_{\theta^i} \pi^i(a^i|b^i;\theta^i) \\ \sum_{a^{-i}} \pi^{-i}(a^{-i}|\boldsymbol{b}^{-i};\theta^{-i}) q^i_{\theta^i}(s, \boldsymbol{b}, \boldsymbol{\theta}, \boldsymbol{a})\end{aligned} \tag{3.12}$$

where $d^{\pi}_{\mu_0}(s, \boldsymbol{b}, \boldsymbol{\theta})$ is the discounted visitation measure starting from initial distribution μ_0, and $q^i_{\theta^i}(s, \boldsymbol{b}, \boldsymbol{\theta}, \boldsymbol{a})$ is the action-value function defined as:

$$\begin{aligned}q^i_{\theta^i}(s, \boldsymbol{b}, \boldsymbol{\theta}, \boldsymbol{a}) = R^i(s, \boldsymbol{a}) + \gamma \sum_{s'} T(s'|s, \boldsymbol{a}) \sum_{\boldsymbol{o}'} \boldsymbol{O}(\boldsymbol{o}'|s') \\ \sum_{\boldsymbol{\theta}'} \boldsymbol{U}(\boldsymbol{\theta}'|\boldsymbol{\theta}, \boldsymbol{\tau}) v^i_{\theta^i}(s', update(\boldsymbol{b}, \boldsymbol{a}, \boldsymbol{o}'), \boldsymbol{\theta}')\end{aligned} \tag{3.13}$$

Proof. See Appendix C for a detailed proof. □

This theorem connects the policy gradient to expectations with respect to the discounted visitation measure, providing a principled approach to policy optimization in partially observable multi-agent settings. The scaling factor $\frac{1}{1-\gamma}$ accounts for the effective horizon of the discounted objective, with its magnitude increasing as γ approaches 1.

In implementation, the choice between these objectives should be guided by the specific characteristics of the social learning scenario being modeled. The discounted formulation may be appropriate for settings with finite horizons, impatient agents, or where near-term performance is particularly valued. However, for scenarios focused on long-run learning dynamics and asymptotic behavior, the average reward criterion provides a more principled approach by equally valuing rewards across all future time periods.

In practice, computing these gradients exactly is typically infeasible. Instead, sample-based approximations and function approximation techniques using neural networks are employed to

estimate the gradient from experience. Nevertheless, in theory, when agents learn to adapt their policies according to these gradients, they eventually converge to an equilibrium concept related to traditional equilibria, which we will introduce in the following section.

3.6 EQUILIBRIUM

Policy gradient optimization naturally leads agents toward stable configurations where no individual agent can unilaterally improve its reward by changing its policy. Given the formalization of our optimization objectives and the policy gradient theorems established above, we now characterize equilibrium concepts in Partially Observable Active Markov Games. By following the gradient directions specified in Theorems 2 and 3, agents can iteratively improve their policies to maximize their respective objectives. This optimization process naturally leads to the concept of Partially Observable Active Equilibrium, which extends the Active Equilibrium from standard Active Markov Games **?** to account for partial observability while representing the fixed point of the policy gradient optimization process.

Definition 7 (Partially Observable Active Equilibrium)**.** *A Partially Observable Active Equilibrium is a joint policy parameter* $\boldsymbol{\theta}^* = \{\theta^{i*}, \theta^{-i*}\}$ *with associated joint update function* $\boldsymbol{U}^* = \{U^{i*}, U^{-i*}\}$ *such that for all* $i \in I$*:*

$$J^i_\pi(\boldsymbol{b}, \theta^{i*}, \theta^{-i*}, U^{i*}, U^{-i*}) \geq J^i_\pi(\boldsymbol{b}, \theta^i, \theta^{-i*}, U^i, U^{-i*}) \quad \textit{(average reward case)} \tag{3.14}$$

$$J^i_{\pi,\gamma}(\boldsymbol{b}, \theta^{i*}, \theta^{-i*}, U^{i*}, U^{-i*}) \geq J^i_{\pi,\gamma}(\boldsymbol{b}, \theta^i, \theta^{-i*}, U^i, U^{-i*}) \quad \textit{(discounted reward case)} \tag{3.15}$$

for all $\boldsymbol{b} \in \boldsymbol{B}$, $\theta^i \in \Theta^i$, $U^i \in \mathscr{U}^i$, *where* $\mathscr{U}^i$ *is the space of possible update functions for agent* i*.*

This equilibrium concept captures the idea that rational agents should optimize not just their immediate policies but also their adaptation strategies, taking into account the learning dynamics of the system while operating under partial observability. At equilibrium, no agent can improve its long-term reward (either discounted or average, depending on the formulation) by unilaterally changing either its policy or its update function, overlapping with the definition of the Bayesian Nash equilibrium in non-cooperative games.

Computing partially observable active equilibria exactly is typically intractable due to the complexity of belief spaces and the sophistication of policy update functions. Nevertheless, this equilibrium concept provides a theoretical benchmark against which practical algorithms can be evaluated. In the next section, we discuss the computational challenges inherent in direct implementation of our theoretical framework, before introducing our practical policy optimization method that approximates equilibrium strategies through policy gradient algorithms.

3.7 Challenges with Exact Computation

Theorems 2 and 3 provide mathematically principled approaches for optimizing policies in both average and discounted return settings, but implementing them exactly is computationally infeasible for most real-world applications. In partially observable environments, agents must maintain belief states, whose evolution can be governed by the Bayes' rule as:

$$b^i_{t+1}(s') = \frac{O^i(o^i_{t+1}|s') \int_{s\in S} T(s'|s,\boldsymbol{a}_t) b^i_t(s) ds}{\int_{s''\in S} O^i(o^i_{t+1}|s'') \int_{s\in S} T(s''|s,\boldsymbol{a}_t) b^i_t(s) ds ds''} \tag{3.16}$$

For environments with large or continuous state spaces, representing and updating these distributions becomes prohibitively expensive. Moreover, exact updates require perfect knowledge of observation functions O^i and transition dynamics T, which may not be readily available to the agents. The complexity compounds in multi-agent settings where agents must reason about others' belief states and policies using Bayesian inference:

$$\mathbb{P}(\boldsymbol{b}^{-i}_t, \boldsymbol{\theta}^{-i}_t | \tau^i_{0:t}) \propto \mathbb{P}(\tau^i_{0:t} | \boldsymbol{b}^{-i}_t, \boldsymbol{\theta}^{-i}_t) \mathbb{P}(\boldsymbol{b}^{-i}_t, \boldsymbol{\theta}^{-i}_t | \tau^i_{0:t-1}) \tag{3.17}$$

This creates nested inference problems—agents maintaining beliefs about others' beliefs—that grow exponentially with the number of agents and the complexity of their policies.

Computing action-value functions adds another layer of intractability, with different formulations for average returns and discounted returns. The triple expectations—over next states, observations, and policy parameters— in equations (3.8) and (3.13) require summing over an exponentially large joint space, where the policy parameter space Θ^i would ideally be modified to be continuous. Furthermore, the policy update functions $\boldsymbol{U}$ of other agents are typically unknown, and the recursive nature of the value function creates computational dependencies that quickly become unmanageable.

The policy gradient calculations presented in Equations (3.7) and (3.12) present similarly insurmountable difficulties. Computing the stochastically stable distribution $\mu_{\theta^i}(s, \boldsymbol{b}, \boldsymbol{\theta})$ for average returns or the discounted visitation measure $d^{\pi}_{\mu_0}(s, \boldsymbol{b}, \boldsymbol{\theta})$ for discounted returns requires implementing additional algorithms (**?**), while expectations over joint actions create additional combinatorial explosion. These computational barriers—state space explosion, nested belief inference, triple expectations, and distribution computation—collectively render exact computation infeasible except for trivial environments. The curse of dimensionality is especially severe due to the combined complexity of partial observability, multi-agent interactions, and evolving policies.

To address these computational challenges and make our theoretical framework practically applicable, we develop POLARIS, a neural network-based algorithm that approximates the key components of our framework while maintaining its essential properties.

3.8 ALGORITHM: POLARIS

In this section, we present POLARIS (Partially Observable Learning with Active Reinforcement In Social environments), our practical implementation of the theoretical framework. POLARIS extends and improves upon the FURTHER algorithm **?** with specialized components for belief modeling, information propagation through networks, and advantage-weighted training. The algorithm uses neural network approximations to overcome the computational challenges outlined in the previous section while preserving the key insights of our theoretical framework. It consists of three integrated modules: a belief processing module, an inference learning module, and a reinforcement learning module, which operate in concert to enable sophisticated social learning in partially observable environments. The architecture integrates several neural network components working in tandem: a Transformer-based belief processor **?** provides sophisticated temporal pattern recognition through its self-attention mechanisms; an inference module implementing a Graph Neural Network (GNN) **?** with temporal attention for network-aware representations; and Multi Agent Soft Actor-Critic (MASAC) following **?** with both average and discounted reward formulations to address the long-term planning requirements of social learning. To maintain performance across changing environments, particularly when the true state changes, POLARIS employs Elastic Weight Consolidation (EWC) **?** that selectively preserves important parameters while allowing adaptation to new conditions. We provide the detailed network architectures in Appendix D.

3.8.1 Belief Processing Module

The belief processing module employs a Transformer architecture to encode the agent's belief state based on partial observations and joint actions:

$$b_t^i = f_{\text{Transformer}}(b_{t-1}^i, o_t^i, \boldsymbol{a}_{t-1}; \psi_{\text{Transformer}}^i) \tag{3.18}$$

Unlike traditional recurrent approaches, the Transformer architecture introduces crucial advantages for social learning scenarios. The self-attention mechanism allows the model to weigh the importance of different parts of the observation history dynamically, enabling it to focus on the most informative signals while downplaying noise or irrelevant information. By processing inputs in parallel rather than sequentially, the Transformer enables more efficient training compared to conventional Recurrent Neural Networks, particularly important when dealing with multiple agents and long interaction histories. Perhaps most importantly for social learning, the Transformer architecture captures relationships between temporally distant observations more effectively, allowing agents to recognize patterns that may emerge over extended periods of social interaction. The Transformer-based belief processor also outputs an explicit belief distribution over possible states:

$$\mathbb{P}(s|b_t^i) = \text{Softmax}(f_{\text{belief\_head}}(b_t^i)) \tag{3.19}$$

This explicit representation allows for more interpretable belief states and enables advantage-weighted training, where the belief processing is optimized not just to track observations but also to produce beliefs that lead to high-value actions. By maintaining a probability distribution over states rather than just an abstract representation, the model provides transparency into its decision-making process and facilitates debugging and analysis of social learning dynamics.

3.8.2 Inference Learning Module

POLARIS implements a Graph Neural Network (GNN) with temporal attention for network-aware social learning. This inference module represents the multi-agent system as a graph where nodes correspond to agents and edges represent observational relationships in the social network. The module outputs distribution parameters:

$$\mu_t, \log \sigma_t = \text{GNN}_{\text{temporal}}(o_t^i, \boldsymbol{a}_t, r_t^i, o_{t+1}^i; \phi_{\text{GNN}}^i) \tag{3.20}$$

Here, μ_t and $\log \sigma_t$ parameterize a Gaussian distribution from which latent variables $\hat{\boldsymbol{z}}_t^{-i}$ are sampled, encoding information about other agents' policies and beliefs. The GNN architecture adapts dynamically to the network topology, enabling automatic configuration for various social network structures. It utilizes multi-head attention mechanisms across both spatial (network) and temporal dimensions, allowing it to differentially weigh connections between agents based on their decision relevance. A key feature is the temporal memory system that maintains a sliding window of past states, enabling temporal attention mechanisms that help the model reason about the evolution of other agents' beliefs and strategies over time. This approach proves especially effective for complex network topologies where information flow patterns significantly influence learning dynamics.

The inference module learns a mapping from observable trajectories to latent variables that are predictive of other agents' behaviors, optimized using the Evidence Lower Bound (ELBO) objective: [1]

$$J_{elbo}^i = \mathbb{E}_{\mathbb{P}(\tau_{0:t}^i), \mathbb{P}(\hat{\boldsymbol{z}}_{1:t}^{-i} | \tau_{0:t-1}^i; \phi_{GNN}^i)} \Bigg[\sum_{t'=1}^{t} \log \mathbb{P}(\boldsymbol{a}_{t'}^{-i} | o_{t'}^i, \hat{\boldsymbol{z}}_{t'}^{-i}; \phi_{GNN}^i) \tag{3.21}$$

$$- \alpha_{KL} D_{KL}(\mathbb{P}(\hat{\boldsymbol{z}}_{t'}^{-i} | \tau_{t'-1}^i; \phi_{GNN}^i) || \mathbb{P}(\hat{\boldsymbol{z}}_{t'-1}^{-i})) \Bigg] \tag{3.22}$$

The ELBO objective balances reconstruction accuracy (predicting other agents' actions) with regularization through the KL divergence term, which encourages temporal consistency in

[1] See **?** for a detailed derivation of the ELBO objective in the sequential setting.

the latent space. This is particularly important in social learning settings where abrupt changes in models of other agents can lead to unstable learning dynamics.

The inference module additionally produces opponent belief distributions, enabling agents to model other agents' understanding of the environment:

$$\mathbb{P}^i(\boldsymbol{b}_t^{-i}|\tau_{0:t}^i) = \text{Softmax}(f_{\text{opponent\_belief}}(\mu_t, \log\sigma_t)) \tag{3.23}$$

This second-order belief modeling allows for sophisticated strategic reasoning, as agents can anticipate how others will respond to their actions based not just on past behavior but on inferred beliefs about the environment.

3.8.3 Reinforcement Learning Module

The reinforcement learning module uses the Soft Actor-Critic (SAC) framework with both average and discounted reward formulations. The policy and action-value functions are conditioned on belief states and inferred latent variables:

$$\pi^i(a^i|b^i,\hat{\boldsymbol{z}}^{-i};\theta^i) = \begin{cases} \text{Categorical}(a^i|\text{MLP}_{\text{policy}}(b^i,\hat{\boldsymbol{z}}^{-i};\theta^i)) & \text{for discrete actions} \\ \mathcal{N}(\mu(b^i,\hat{\boldsymbol{z}}^{-i};\theta^i),\sigma(b^i,\hat{\boldsymbol{z}}^{-i};\theta^i)) & \text{for continuous actions} \end{cases} \tag{3.24}$$

$$q_{\theta^i}^i(b^i,\hat{\boldsymbol{z}}^{-i},\boldsymbol{a};\psi_\beta^i) = \text{MLP}_{\text{value}}(b^i,\hat{\boldsymbol{z}}^{-i},\boldsymbol{a};\psi_\beta^i) \tag{3.25}$$

For continuous action spaces, the policy network outputs the mean μ and standard deviation σ of a Gaussian distribution, where actions are sampled using the reparameterization trick. Then, hyperbolic tangent function is applied to bound the actions within a desired range, with appropriate adjustments to the log probability calculation.

The SAC framework is particularly well-suited for social learning environments due to its entropy regularization, which promotes exploration and prevents premature convergence to suboptimal equilibria. By supporting both categorical distributions for discrete action spaces and Gaussian distributions for continuous action spaces, the policy network can express uncertainty about optimal actions—crucial in partially observable settings where perfect inference is rarely possible. The value functions measure expected returns conditioned not only on the agent's belief state but also on inferred representations of other agents' policies, enabling strategic reasoning about the long-term consequences of actions in a multi-agent context. The RL module is optimized using three objectives, each addressing a different aspect of the learning problem:

Value Function Objective POLARIS supports both discounted and average reward formulations to accommodate different types of social learning problems. For discounted returns:

$$J_q^i(\psi_\beta^i) = \mathbb{E}_{D^i}\left[(y - q_{\theta^i}^i(b^i, \hat{\boldsymbol{z}}^{-i}, \boldsymbol{a}; \psi_\beta^i))^2\right] \tag{3.26}$$

$$y = r^i + \gamma \cdot v_{\theta^i}^i(b_{t+1}^i, \hat{\boldsymbol{z}}_{t+1}^{-i}; \bar{\psi}_\beta^i) \tag{3.27}$$

where $\gamma \in [0,1)$ is the discount factor. This formulation is appropriate for scenarios with finite horizons or where near-term performance is particularly valued.

For average reward:

$$J_q^i(\psi_\beta^i, \rho_{\theta^i}^i) = \mathbb{E}_{D^i}\left[(y - q_{\theta^i}^i(b^i, \hat{\boldsymbol{z}}^{-i}, \boldsymbol{a}; \psi_\beta^i))^2\right] \tag{3.28}$$

$$y = r^i - \rho_{\theta^i}^i + v_{\theta^i}^i(b_{t+1}^i, \hat{\boldsymbol{z}}_{t+1}^{-i}; \bar{\psi}_\beta^i) \tag{3.29}$$

The average reward formulation better captures long-run learning dynamics and asymptotic behavior by equally valuing rewards across all future time periods. By supporting both approaches, POLARIS can be tailored to match the specific characteristics of different social learning scenarios without requiring structural changes to the architecture.

Policy Objective Maximizes expected value plus entropy:

$$J_\pi^i(\theta^i) = \mathbb{E}_{D^i}\left[\sum_{a^i} \pi^i(a^i|b^i, \hat{\boldsymbol{z}}^{-i}; \theta^i) \sum_{\boldsymbol{a}^{-i}} \pi^{-i}(\boldsymbol{a}^{-i}|\boldsymbol{b}^{-i}, \hat{\boldsymbol{z}}^{-i}) \min_{\beta=1,2} q_{\theta^i}^i(b^i, \hat{\boldsymbol{z}}^{-i}, a^i, \boldsymbol{a}^{-i}; \psi_\beta^i)\right. \tag{3.30}$$

$$\left. + \alpha_e H(\pi^i(\cdot|b^i, \hat{\boldsymbol{z}}^{-i}; \theta^i))\right] \tag{3.31}$$

For discrete action spaces, the summation and entropy terms are computed directly over the action probabilities. For continuous action spaces, the expectation is approximated using samples from the policy, and the entropy term is calculated as:

$$H(\pi^i(\cdot|b^i, \hat{\boldsymbol{z}}^{-i}; \theta^i)) = \begin{cases} -\sum_{a^i} \pi^i(a^i|b^i, \hat{\boldsymbol{z}}^{-i}; \theta^i) \log \pi^i(a^i|b^i, \hat{\boldsymbol{z}}^{-i}; \theta^i) & \text{for discrete actions} \\ \frac{1}{2} \log \det(2\pi e \sigma^2(b^i, \hat{\boldsymbol{z}}^{-i}; \theta^i)) & \text{for continuous actions} \end{cases} \tag{3.32}$$

The entropy calculation must also account for the hyperbolic tangent transformation using the change of variables formula.

The policy objective combines expected future returns with an entropy bonus, creating a balance between exploitation and exploration. The entropy term encourages the policy to maintain stochasticity, preventing premature convergence to deterministic policies that might be suboptimal in partially observable settings. The use of the minimum value across dual Q-networks helps mitigate overestimation bias—a persistent challenge in Q-learning approaches. This is particularly important in social learning contexts where overestimated values can lead to overly optimistic strategies that fail to account for the strategic adaptations of other agents.

Advantage-Weighted Transformer Training

A key innovation in POLARIS is the use of advantage-weighted training for the belief processor:

$$J^i_{\text{transformer}}(\psi^i_{\text{Transformer}}) = \mathbb{E}_{D^i}\left[A^i(b^i,a^i)\cdot \text{BCE}(\mathbb{P}(s|b^i),o^i_{t+1})\right] \tag{3.33}$$

where $A^i(b^i,a^i)$ is the advantage function and BCE is binary cross-entropy loss. This objective bridges belief formation and policy optimization in a novel way. By weighting the belief update loss with the advantage function, the Transformer is encouraged to generate belief states that not only accurately predict future observations but also lead to high-value actions. In traditional partially observable reinforcement learning, belief updates are typically performed independently of policy optimization. By contrast, POLARIS creates a direct pathway for policy performance to influence belief formation, prioritizing informative aspects of the state space that have the greatest impact on decision quality. This approach is particularly valuable in social learning contexts, where certain beliefs about others' strategies may be more consequential than others for effective coordination or competition.

Catastrophic Forgetting Prevention

In social learning environments, the underlying state of the world often remains fixed during an episode, but varies across episodes. This creates a continual learning challenge: agents must adapt to new states while retaining knowledge about previously encountered states. Without proper mechanisms, neural networks tend to overwrite previously learned representations when adapting to new conditions—a phenomenon known as catastrophic forgetting (**?**). POLARIS addresses this challenge through Elastic Weight Consolidation (EWC), which selectively preserves important parameters for previously encountered states while allowing flexibility for adaptation. EWC augments the standard training objective with a regularization term that penalizes significant changes to parameters critical for performance in previously encountered states:

$$L_{\text{EWC}}(\theta) = L(\theta) + \sum_i \frac{\lambda}{2} F^i(\theta^i - \theta^{i^*})^2 \tag{3.34}$$

where θ^* represents parameters that performed well on previous tasks, F^i is the Fisher information matrix that measures parameter importance, and λ controls the strength of regularization. The Fisher information matrix provides a principled way to identify which parameters are most crucial for previously learned tasks, calculated using gradients of the log-likelihood of the model's output with respect to the parameters on data from previously encountered states.

Crucially, POLARIS introduces a dynamic mechanism that automatically adjusts the EWC importance factor based on observed performance:

$$\lambda_t = \begin{cases} \lambda_{t-1} \cdot 0.8 & \text{if loss ratio} > 1.5 \text{ (loss increasing too much)} \\ \lambda_{t-1} \cdot 1.2 & \text{if loss ratio} < 0.5 \text{ (loss decreasing too quickly)} \\ \lambda_{t-1} & \text{otherwise} \end{cases} \tag{3.35}$$

This adaptive approach monitors the ratio between current and previous loss values, reducing the EWC importance when the loss is increasing too rapidly (indicating difficulty learning new information due to excessive preservation of old parameters) and increasing it when the loss is decreasing too quickly (suggesting potential forgetting of valuable past knowledge). The implementation also includes both L1 and L2 penalties with adaptive weighting between them, transitioning from more L2 regularization for early tasks to more L1 regularization for later tasks, encouraging sparser updates as the model accumulates more knowledge from different states.

3.8.4 Training Process

The overall training process interleaves updates to the three modules, ensuring that improvements in one component can benefit the others within each training iteration. Algorithm 1 outlines the complete POLARIS training procedure for a single agent.

Algorithm 1 POLARIS Training Algorithm for Single Agent

1: Initialize belief processor, inference module, policy and value networks with parameters $\psi^i_{\text{Transformer}}$, ϕ^i, θ^i, ψ^i_β
2: Initialize replay buffer D^i, initial belief b^i_0, and latent state $\hat{\boldsymbol{z}}^{-i}_0$
3: Determine action space type (discrete or continuous)
4: **for** each step t **do**
5: Observe signal o^i_t and joint action $\boldsymbol{a}_{t-1}$
6: Update belief: $b^i_t, \mathbb{P}(s|b^i_t) = f_{\text{Transformer}}(b^i_{t-1}, o^i_t, \boldsymbol{a}_{t-1}; \psi^i_{\text{Transformer}})$
7: Select action:
8: **if** discrete action space **then**
9: $a^i_t \sim \pi^i(\cdot|b^i_t, \hat{\boldsymbol{z}}^{-i}_t; \theta^i)$ ▷ Sample from categorical distribution
10: **else**
11: $\mu, \sigma = \text{MLP}_{\text{policy}}(b^i_t, \hat{\boldsymbol{z}}^{-i}_t; \theta^i)$
12: $\varepsilon \sim \mathcal{N}(0, I)$
13: $a^i_t = \tanh(\mu + \sigma \odot \varepsilon)$ ▷ Sample using reparameterization trick
14: **end if**
15: Execute action, observe reward r^i_t and next signal o^i_{t+1}
16: Infer latent: $\hat{\boldsymbol{z}}^{-i}_{t+1} = f_{\text{GNN}}(o^i_t, \boldsymbol{a}_t, r^i_t, o^i_{t+1}; \phi^i)$
17: Store transition in replay buffer
18: **if** update step **then**
19: Sample batch B from replay buffer D^i
20: Update inference module: $\phi^i \leftarrow \phi^i - \alpha_\phi \nabla_{\phi^i} J^i_{\text{elbo}}(B)$
21: Update value networks: $\psi^i_\beta \leftarrow \psi^i_\beta - \alpha_\psi \nabla_{\psi^i_\beta} J^i_q(B)$
22: Update policy: $\theta^i \leftarrow \theta^i - \alpha_\theta \nabla_{\theta^i} J^i_\pi(B)$ ▷ Using appropriate entropy calculation
23: Update belief processor: $\psi^i_{\text{Transformer}} \leftarrow \psi^i_{\text{Transformer}} - \alpha_\psi \nabla_{\psi^i_{\text{Transformer}}} J^i_{\text{transformer}}(B)$
24: Update target networks: $\bar{\psi}^i_\beta \leftarrow \tau \psi^i_\beta + (1-\tau)\bar{\psi}^i_\beta$
25: **if** true state has changed **then**
26: Calculate Fisher matrices from current replay buffer
27: Register task with EWC
28: **end if**
29: **end if**
30: **end for**

The algorithm follows a sequential process where each agent first updates its belief state based on new observations and the joint actions of all agents using the Transformer-based belief processor (line 6). Based on this updated belief and the current inference of other agents' latent states, the agent selects an action using its stochastic policy—sampling from a categorical distribution for discrete action spaces (line 9) or using the reparameterization trick for continuous action spaces (lines 11-13). After executing this action and observing the reward and next observation (line 15), the agent updates its inference of other agents' latent states using the GNN-based inference module (line 16). All of these experiences are stored in a replay buffer

for later training (line 17).

At designated update steps, the agent samples a batch of transitions from its replay buffer and uses these to update all network components. The inference module is updated first (line 19) to improve predictions of other agents' actions using the ELBO objective. Next, the value networks are updated (line 20) to better estimate expected returns. The policy network is then updated (line 21) to maximize expected value and entropy. Finally, the belief processor is updated (line 22) using the advantage-weighted training approach. Target networks, used for stable learning in the SAC framework, are updated using a weighted average of the current and target parameters (line 23).

A crucial feature of POLARIS is its adaptive response to changes in the environment state. When the true state changes (line 24), the algorithm calculates Fisher information matrices from the current replay buffer (line 25) and registers a new task with the EWC mechanism (line 26), preserving important knowledge from the previous state while allowing adaptation to the new state.

In the next chapter, we demonstrate the practical application of our POAMG framework and the POLARIS algorithm to canonical social learning scenarios from economic theory. The versatility of our approach is particularly evident in its ability to seamlessly handle both continuous and discrete action spaces, as well as varying types of return objectives. This capability proves essential when implementing strategic experimentation models with continuous allocation decisions and average reward objectives, and learning without experimentation models with discrete signaling actions and discounted reward objectives. Through these implementations, we validate theoretical predictions from economic models while uncovering new insights about strategic adaptation in partially observable multi-agent systems that would be difficult to obtain through purely analytical approaches. The experimental results not only demonstrate the effectiveness of our theoretical framework but also highlight how computational methods can complement and extend traditional economic analysis of social learning phenomena.

4 RESULTS

Building on the theoretical framework of Partially Observable Active Markov Games developed in the previous section, we now apply our POLARIS algorithm to concrete social learning scenarios. This implementation demonstrates how our approach bridges theoretical economic models with computational reinforcement learning methods, allowing us to validate theoretical predictions while uncovering new insights about strategic adaptation in partial observability settings.

We implement our Partially Observable Active Markov Game framework to model the two canonical social learning scenarios discussed in the background section: strategic experimentation with observable rewards and learning without experimentation through action observation. By reformulating these economic models as POAMGs, we demonstrate how our framework can capture the sophisticated learning dynamics present in these scenarios while enabling the discovery of complex strategies through our POLARIS algorithm. [1]

4.1 STRATEGIC EXPERIMENTATION

We reformulate the strategic experimentation model of **?** as a Partially Observable Active Markov Game. This framework analyzes undiscounted continuous-time games where a number of symmetric players act non-cooperatively, trying to learn an unknown state of the world governing the risky arm's expected payoff. The state space $S = \{0, 1, \ldots, m\}$ represents possible states of the world, with a deterministic transition function since the underlying state remains constant throughout the interaction. Each agent's action space $A^i = [0, 1]$ represents the fraction of resources allocated to the risky arm at each decision point, creating a continuous action space that allows for nuanced exploration strategies.

While the original model operates in continuous time with Lévy processes, our POAMG implementation discretizes time using the Euler-Maruyama scheme (**??**) for both the background signal and the agents' payoff processes. This numerical method is widely used for approximating solutions to stochastic differential equations driven by both Brownian motion and Poisson processes.

The agents receive a public signal produced by the background process, which leads to the observation function:

$$O^i(s, o) = \mathbb{P}(o_t | B_{t-1:t}) \tag{4.1}$$

[1] Complete Python implementation is available at https://github.com/ecdogaroglu/POLARIS.

where o_t represents the observation at discrete time step t and $B_{t-1:t}$ represents the signal increment in discrete time. To address the time-dependent nature of Lévy processes while maintaining compatibility with our POAMG framework, we formulate the reward function for each agent i as:

$$R^i(s,a^i) = (1-a^i)r_{safe} + a^i \frac{X^i_{t-1:t}}{\Delta t} \tag{4.2}$$

where Δt is the discretization time step,

$$X^i_{t-1:t} = \alpha_s \Delta t + \sigma(W^i_t - W^i_{t-1}) + \Delta Y^i_t, \tag{4.3}$$

$(W^i_t - W^i_{t-1}) \sim \mathcal{N}(0,\Delta t)$, and ΔY^i_t is the increment of the compound Poisson processes over the interval $[t-1,t]$. This normalization converts accumulated rewards to instantaneous reward rates, preserving the incentive structure of the original model. A comprehensive mathematical treatment of this discretization approach, including convergence properties and the preservation of strategic incentives, is provided in Appendix E.1.

Each agent observes the increments in the public background signal, their own payoff process (dependent on their allocation a^i to the risky arm and the true state ω), and the allocations together with the rewards of all other agents. The unique feature of the strategic experimentation model is the explicit form of the symmetric Markov perfect equilibrium, which depends only on three factors: the safe payoff, the expected current payoff of the risky arm, and the expected full-information payoff. To validate this theoretical result, we configure POLARIS to measure: (1) The cutoff belief at which agents switch between the safe and risky arm (2) The relationship between individual experimentation intensity and group experimentation intensity (3) Free-riding behavior as a function of the number of players (4) Strategy independence from the specific payoff-generating process To measure convergence to equilibrium behavior, we track the Kullback-Leibler divergence (KL divergence) between agents' actual strategies and the theoretically optimal strategy specified in Keller and Rady's work. The KL divergence is calculated as

$$D_{KL}(P||Q) = \sum_x P(x) \log \frac{P(x)}{Q(x)}, \tag{4.4}$$

where P represents the agent's learned policy distribution and Q represents the theoretical optimal policy distribution at each belief state:

$$a^*(b) = \begin{cases} 0 & \text{if } I(b) \le k_0, \\ \frac{I(b)-k_0}{n-1} & \text{if } k_0 < I(b) < k_0+n-1, \\ 1 & \text{if } I(b) \ge k_0+n-1. \end{cases} \tag{4.5}$$

For varying group sizes $n \in \{2,4,8,16\}$, we measure the relationship between individual and group experimentation. The theory predicts that the set of beliefs for which no experimentation occurs remains unchanged as n increases—a stark manifestation of free-riding behavior.

Additionally, the theory predicts that in regions of partial experimentation, the intensity per agent will decrease with n, while the total group intensity increases. These predictions create testable patterns that our POLARIS implementation can validate.

4.2 Learning Without Experimentation

We reformulate the learning without experimentation model of **?** as a Partially Observable Active Markov Game. In this formulation, the state space $S = \{s^1, s^2, \ldots, s^m\}$ represents possible states of the world, with a deterministic transition function since the state remains constant. Each agent's action space $A^i = \{a^1, a^2, \ldots, a^m\}$ corresponds to potential states, where a^j is the unique optimal action is state s^j. In addition to observing their neighbors' actions, each agent receives a private signal $o_t^i \in \Omega^i = S$ drawn from distribution:

$$O^i(s,o) = \begin{cases} q & \text{if } s = o \\ \frac{1-q}{m-1} & \text{otherwise} \end{cases}$$

where $q > 1/m$ is the signal accuracy. Since agents don't observe real rewards in this model, we construct an observed reward function that facilitates learning:

$$R^i(s_t, a_t^i) = v(o_t^i, a_t^i) = \frac{q \cdot \mathbb{I}_{\{a_t^i = \varphi(o_t^i)\}} - (1-q) \cdot \mathbb{I}_{\{a_t^i \neq \varphi(o_t^i)\}}}{2q-1}$$

where φ maps states to their corresponding correct actions. [2] This construction ensures the expected reward matches the true utility function in expectation: [3]

$$\mathbb{E}_{o \sim \mu^s}[v(o,a)] = u(s,a) = \mathbb{I}_{\{a = \varphi(s)\}}.$$

For methodological soundness, we distinguish between two learning processes: algorithm learning (neural network training via gradient descent) and belief learning (agents refining beliefs about the underlying state based on observations). Brandl's theoretical analysis focuses on the second type—how quickly agents converge to correct actions as they accumulate information over time, assuming fixed information-processing strategies. To properly measure the asymptotic learning rates predicted by Brandl's theory, we employ a two-phase methodology: training with both types of learning occurring simultaneously, followed by post-training evaluation where we freeze network parameters while still allowing belief states to update based on new observations.

The learning rate in Brandl's framework is defined through asymptotic behavior:

$$r = \lim_{t \to \infty} -\frac{1}{t} \log \mathbb{P}(a_t^i \neq a_\omega)$$

[2] Formally, the mapping $\varphi : S \to A$ is a bijective function defined as $\varphi(s^j) = a^j$ for all $j \in \{1, 2, ..., m\}$, associating each state with the action having the same index.

[3] See E.2 for a detailed derivation and the formulation for a generic dimension m.

During post-training evaluation, we collect empirical mistake probabilities across many episodes, fit them to the exponential decay model $\mathbb{P}(a_t^i \neq a_\omega) \approx e^{-rt}$ using log-linear regression, and extract the empirical learning rate for comparison with theoretical bounds.

To validate scaling properties with network size, we implement a systematic protocol that trains separate POLARIS instances across different network configurations (complete, ring, star, random) and sizes (from very small n=2,3,4 to large n=30,50 where feasible).

For the binary case where $S = A = \{0, 1\}$ and signal accuracy is $q = 0.75$, the theoretical bounds from **?** are: autarky rate $r_{aut} \approx 0.14$ (learning rate of a single isolated agent), coordination rate $r_{crd} \approx 0.55$ (achievable rate with coordination), and bounded rate $r_{bdd} \approx 1.10$ (upper bound on any agent's learning rate).

Our experiments validate two key theoretical predictions. First, the Learning Barrier (Theorem 1) states that for any strategy profile, there exists some agent whose learning rate is bounded by r_{bdd}, regardless of network size or structure. Second, the Coordination Benefits (Theorem 2) predicts that in sufficiently large, strongly connected networks, there exist strategies enabling all agents to learn faster than in autarky.

REINFORCEMENT LEARNING BACKGROUND

This section provides a brief overview of reinforcement learning (RL) and multi-agent reinforcement learning (MARL) to provide a foundation for understanding the theoretical framework presented in this paper.

A.1 REINFORCEMENT LEARNING FOUNDATIONS

Reinforcement learning (RL) provides a mathematical framework for solving sequential decision-making problems under uncertainty. In the classical single-agent setting, an agent interacts with a stationary environment by observing states, taking actions, and receiving rewards, with the objective of maximizing its expected cumulative reward over time **?**. This interaction is formalized as a Markov Decision Process (MDP), defined as a tuple $M = \langle S, A, T, R, \gamma \rangle$ where:

- S is the state space, representing all possible configurations of the environment
- A is the action space, representing all possible decisions available to the agent
- $T : S \times A \mapsto \Delta(S)$ is the transition function, specifying the probability distribution over next states given the current state and action
- $R : S \times A \mapsto \mathbb{R}$ is the reward function, specifying the immediate reward received after taking an action in a state
- $\gamma \in [0, 1)$ is the discount factor, balancing immediate versus future rewards

The agent's behavior is characterized by a policy $\pi : S \mapsto \Delta(A)$, which maps states to probability distributions over actions. The policy can be evaluated using value functions: the state-value function $V^{\pi}(s)$ represents the expected return when starting in state s and following policy π thereafter, while the action-value function $Q^{\pi}(s, a)$ represents the expected return after taking action a in state s and following policy π thereafter. The goal of RL is to find an optimal policy π^* that maximizes the expected return from all states.

The Markov property, which states that the future is independent of the past given the present, is a fundamental assumption in MDPs. This property ensures that the current state provides all the necessary information for making optimal decisions, greatly simplifying the learning problem. However, this assumption becomes problematic in multi-agent settings, as we will discuss next.

A.2 Multi-Agent Reinforcement Learning: Concepts and Challenges

Multi-agent reinforcement learning (MARL) extends the single-agent RL framework to environments with multiple autonomous agents that interact simultaneously **?????**. MARL encompasses a wide spectrum of scenarios, from fully cooperative tasks where agents share a common reward function, to fully competitive zero-sum games, to the general mixed cooperative-competitive case with individual reward functions **?**. These interactions are commonly formalized as Markov games (also known as stochastic games) **??**, defined as a tuple $M_n = \langle I, S, A, T, R, \gamma \rangle$, where:

- $I = \{1, \ldots, n\}$ is the set of n agents
- S is the state space, shared among all agents
- $A = \times_{i \in I} A^i$ is the joint action space, where A^i is the action space of agent i
- $T : S \times A \mapsto \Delta(S)$ is the transition function that depends on the joint action
- $R = \times_{i \in I} R^i$ is the joint reward function, where $R^i : S \times A \mapsto \mathbb{R}$ is agent i's individual reward function
- $\gamma \in [0, 1)$ is the discount factor

MARL introduces several fundamental challenges beyond those encountered in single-agent RL. The joint action space grows exponentially with the number of agents, creating a combinatorial explosion that makes exploration and learning increasingly difficult as more agents are added to the system. Coordination challenges further complicate the learning process, as agents must synchronize their actions to achieve effective joint behavior, especially in cooperative settings where team performance depends on complementary actions **?**, that can also involve explicit communication **?**. In scenarios with shared rewards, the credit assignment problem becomes particularly troublesome, as determining individual contributions to team success grows increasingly complex when outcomes result from joint actions rather than individual decisions. MARL also demands sophisticated strategic reasoning, requiring agents to model and reason about other agents' goals, beliefs, and strategies, particularly in competitive or mixed cooperative-competitive settings. Perhaps most critically, when multiple agents learn simultaneously, the environment becomes non-stationary from each agent's perspective, as other agents' changing policies continuously modify the effective environment dynamics, violating a fundamental assumption of traditional reinforcement learning algorithms.

A.3 The Non-Stationarity Challenge

The non-stationarity problem in MARL represents a fundamental departure from single-agent RL assumptions and presents one of the most significant obstacles to effective multi-agent learn-

ing **??**. To understand this challenge more precisely, we can analyze how learning dynamics alter the effective environment for each agent. When multiple agents learn simultaneously, each agent i with policy π^i effectively faces an environment whose dynamics depend on the joint policies of all other agents π^{-i}. From agent i's perspective, the effective transition function becomes:

$$T^i_{\boldsymbol{\pi}_t^{-i}}(s_{t+1}|s_t, a_t^i) = \sum_{\mathbf{a}^{-i} \in A^{-i}} \left(\prod_{j \in I \setminus \{i\}} \pi_t^j(a^j|s_t) \right) \cdot T(s_{t+1}|s_t, (a_t^i, \mathbf{a}^{-i})) \tag{A.1}$$

where A^{-i} denotes the joint action space, $\mathbf{a}^{-i}$ denotes the joint action and $\boldsymbol{\pi}_t^{-i}$ denotes the joint policies of all agents except i. When other agents update their policies from $\boldsymbol{\pi}_t^{-i}$ to $\boldsymbol{\pi}_{t+1}^{-i}$ through learning, this effective transition function changes, creating a non-stationary environment for agent i. This non-stationarity violates the Markov property assumption underlying most RL algorithms and can lead to several significant challenges that fundamentally undermine the theoretical foundations of single-agent RL. When multiple agents learn simultaneously, standard RL convergence guarantees no longer apply, and learning algorithms may oscillate or diverge as the effective environment continuously shifts **?**. This dynamic environment causes value function approximations to become increasingly inaccurate over time, leading to suboptimal policy decisions as agents base their choices on outdated models of their environment **?**. Further complicating matters, agents face a moving target problem where they must simultaneously learn optimal policies while adapting to the evolving strategies of others, creating a complex coupled learning process that resists straightforward optimization approaches **?**. The situation becomes particularly problematic in competitive settings, where learning dynamics may lead to cyclic policy changes rather than convergence to stable strategies, as agents continuously adapt and counter-adapt to each other's evolving behaviors **?**. These interconnected challenges highlight why addressing non-stationarity remains one of the central research questions in multi-agent reinforcement learning.

A.4 Traditional Approaches to Non-Stationarity

Researchers have developed various approaches to address the non-stationarity challenge in MARL. These can be broadly categorized as follows:

A.4.1 Independent Learning

The simplest approach is to ignore non-stationarity entirely, treating other agents as part of the environment and applying standard single-agent RL algorithms independently for each agent **?**. This approach, often called Independent Learning (IL), requires no explicit coordination or modeling of other agents. Methods in this category include Independent Q-Learning **?**, where each agent maintains its own Q-function and updates it using only its own experiences. While computationally efficient and naturally scalable to many agents, independent learning lacks theoretical convergence guarantees and can fail in complex multi-agent scenarios due to the violation of the stationarity assumption.

A.4.2 Centralized Training with Decentralized Execution

To mitigate non-stationarity during learning while preserving autonomous execution, many approaches adopt the paradigm of centralized training with decentralized execution (CTDE) **???**. In CTDE, agents have access to additional information during training (e.g., joint actions, global state, other agents' policies) but operate based solely on their local observations during execution.

Notable algorithms in this category include Multi-Agent Deep Deterministic Policy Gradient (MADDPG) **?**, which uses centralized critics with access to joint actions and states during training but decentralized actors during execution. Another significant approach is Counterfactual Multi-Agent Policy Gradients (COMA) **?**, which addresses credit assignment using a centralized critic that computes a counterfactual baseline. The category also features QMIX **?**, which learns a centralized value function that factorizes into a mixing of individual agent Q-values, ensuring consistency between centralized and decentralized policies. While CTDE methods help stabilize learning, they still do not fully address the fundamental non-stationarity issue, as they focus on adapting to current policies rather than reasoning about policy evolution over time.

A.4.3 Opponent Modeling and Population-Based Training

Another approach is to explicitly model the behavior and learning processes of other agents, enabling more informed adaptation **??**. This can involve predicting other agents' policies, types, or learning dynamics. Methods in this category include Deep Reinforcement Opponent Network (DRON) **?**, which integrates opponent modeling into deep Q-learning. Another significant approach is Probabilistic Recursive Reasoning (PR2) **?**, which models higher-order beliefs about other agents' reasoning processes.

The category also encompasses Population-Based Training (PBT) **?**, which trains a population of agents simultaneously, creating a naturally changing learning environment that promotes robustness to non-stationarity. While these approaches can better handle changing policies, they typically focus on adaptation to current policies rather than influencing the learning trajectories of other agents.

A.4.4 Equilibrium Learning and Stability Concepts

Drawing on game theory, some approaches focus on finding policies that constitute game-theoretic solution concepts like Nash equilibria **???**. These methods aim to find stable joint policies where no agent has an incentive to unilaterally deviate. Notable algorithms include Nash Q-Learning **?**, which converges to Nash equilibria in general-sum Markov games. Another significant approach is Friend-or-Foe Q-Learning **?**, which converges to optimal policies in restricted classes of Markov games.

The category also features WoLF-PHC (Win or Learn Fast - Policy Hill Climbing) **?**, which adjusts learning rates based on whether an agent is "winning" or "losing" to promote conver-

gence. While these approaches provide theoretical guarantees under certain conditions, they often make strong assumptions about the game structure and other agents' rationality. Moreover, they focus on convergence to static equilibria rather than the dynamic nature of multi-agent learning.

A.4.5 Meta-Learning for Non-Stationarity

Recent work has explored meta-learning as a framework for rapid adaptation to non-stationarity in MARL **??**. These approaches train agents to quickly adapt to new agents or environments, treating non-stationarity as a meta-learning problem. Key methods include Continuous Adaptation via Meta-Learning (CAML) **?**, which uses meta-learning to quickly adapt to evolving opponents.

Another significant approach is Meta-MAPG (Meta-Multiagent Policy Gradient) **?**, which extends meta-learning to explicitly account for the influence of an agent's actions on the learning processes of other agents. While meta-learning approaches show promise for rapid adaptation, they often still lack a comprehensive framework for modeling and influencing the long-term learning dynamics of multi-agent systems.

A.5 ACTIVE MARKOV GAMES

Active Markov Games provide a more sophisticated framework for modeling the dynamic nature of multi-agent learning by explicitly incorporating the policy update processes of all agents **?**. They extend standard Markov games to capture not just the immediate effects of actions on states and rewards, but also their influence on future policy updates, addressing the non-stationarity challenge at a fundamental level.

A.5.1 Formal Definition

An Active Markov Game is defined as a tuple $M_n = \langle I, S, A, T, R, \Theta, U \rangle$, where:

- $I = \{1, \ldots, n\}$ is the set of n agents
- S is the state space
- $A = \times_{i \in I} A^i$ is the joint action space, where A^i is the action space of agent i
- $T : S \times A \mapsto \Delta(S)$ is the transition function
- $R = \times_{i \in I} R^i$ is the joint reward function, where $R^i : S \times A \mapsto \mathbb{R}$ is agent i's individual reward function
- $\Theta = \times_{i \in I} \Theta^i$ is the joint policy parameter space, where Θ^i is the policy parameter space for agent i

- $U = \times_{i \in I} U^i$ is the joint Markovian policy update function, where $U^i : \Theta^i \times T^i \mapsto \Delta(\Theta^i)$ maps current policy parameters and transitions to distributions over next policy parameters

Here, T^i represents the trajectory space for agent i, with a particular element $\tau_t^i = \{s_t, \mathbf{a}_t, r_t^i, s_{t+1}\}$ consisting of the current state, joint action, reward received, and the next state. The interaction process in an Active Markov Game unfolds through a sequential procedure of actions and updates. At timestep t, each agent i selects an action $a_t^i \sim \pi^i(\cdot|s_t; \theta_t^i)$ based on its parameterized policy with parameters $\theta_t^i \in \Theta^i$. Following this selection, the environment transitions from state s_t to s_{t+1} according to $T(s_{t+1}|s_t, \mathbf{a}_t)$ based on the joint action $\mathbf{a}_t = (a_t^1, \ldots, a_t^n)$. Each agent i then receives a reward $r_t^i = R^i(s_t, \mathbf{a}_t)$ and subsequently updates its policy parameters according to its update function $\theta_{t+1}^i \sim U^i(\theta_{t+1}^i|\theta_t^i, \tau_t^i)$. This process continues until non-stationary policies converge to a recurrent set of joint policies. The key insight is that agent i's actions not only affect immediate states and rewards but also influence future policy updates of all agents, including itself, through the trajectory τ_t^i.

A.5.2 Augmented Transition Function and Stationarity

The Active Markov Game framework fundamentally transforms the non-stationarity problem by explicitly modeling how policies evolve through Markovian update functions. To understand this transformation, let us contrast the transition functions in standard MARL and Active Markov Games. In standard MARL, the transition function $T : S \times A \mapsto \Delta(S)$ only captures state transitions based on joint actions. However, Active Markov Games introduce a critical innovation: they explicitly model how policies change via Markovian update functions. The augmented transition function becomes:

$$\mathbb{P}(s_{t+1}, \boldsymbol{\theta}_{t+1}|s_t, \boldsymbol{\theta}_t) = \sum_{\mathbf{a}_t \in A} \left(\prod_{i \in I} \pi^i(a_t^i|s_t; \theta_t^i) \right) \times T(s_{t+1}|s_t, \mathbf{a}_t) \times \boldsymbol{U}(\boldsymbol{\theta}_{t+1}|\boldsymbol{\theta}_t, \boldsymbol{\tau}_t) \tag{A.2}$$

The crucial difference is the inclusion of $\boldsymbol{U}$ - a Markovian update function that specifies exactly how policy parameters $\boldsymbol{\theta}_t$ (and consequently policies $\boldsymbol{\pi}$) evolve based on current parameters and trajectories. By making policy updates an explicit part of the system dynamics, what was previously an unpredictable external process becomes a predictable internal one. This augmented transition function operates over the joint space $(s, \boldsymbol{\theta}) \in S \times \Theta$, creating a joint process that is stationary even though individual policies are changing.

A.5.3 Theoretical Properties and Convergence

Active Markov Games exhibit important theoretical properties that enable rigorous analysis of multi-agent learning dynamics:

Stationary Periodic Distributions. Under mild assumptions, the joint process of states and policies in an Active Markov Game converges to a stationary periodic distribution **?**:

$$\mu_k(s, \theta | s_0, \theta_0, \ell) = \mathbb{P}(s_t = s, \theta_t = \theta | s_0, \theta_0, \ell) \tag{A.3}$$

where $\ell = t\%k$ with $\%$ denoting the modulo operation. This distribution represents the limiting behavior of the system after convergence, characterized by potentially periodic patterns of states and policies. The parameter k represents the period length, with $k = 1$ corresponding to fixed-point convergence. To address sensitivity to initial conditions, the framework introduces the concept of *stochastically stable distributions* **?**. These are limiting distributions that emerge when small random perturbations are added to policy updates, providing robustness to initial state and policy configurations.

Long-term Optimization Objective. In Active Markov Games, agents optimize for long-term average reward rather than discounted return **??**:

$$\max_{\theta^i, U^i} \rho^i(s, \theta, U) := \max_{\theta^i, U^i} \lim_{T \to \infty} \mathbb{E}\left[\frac{1}{T} \sum_{t=0}^{T} R^i(s_t, \mathbf{a}_t) \,\middle|\, \begin{array}{c} s_0 = s, \theta_0 = \theta, \\ \mathbf{a}_{0:T} \sim \boldsymbol{\pi}(\cdot | s_{0:T}; \theta_{0:T}), \\ s_{t+1} \sim T(\cdot | s_t, \mathbf{a}_t), \theta_{t+1} \sim U(\cdot | \theta_t, \boldsymbol{\tau}_t) \end{array} \right] \tag{A.4}$$

where $0 : T$ denotes the sequence from time 0 to T. This formulation encourages agents to consider how to influence the limiting behavior of the system rather than just short-term performance, addressing the non-stationarity challenge at a fundamental level.

Active Equilibrium. Active Markov Games give rise to a new solution concept called the Active Equilibrium **?**, which generalizes traditional game-theoretic equilibria like Nash equilibrium. An Active Equilibrium is a joint policy parameter $\theta^* = \{\theta^{i*}, \theta^{-i*}\}$ with associated joint update function $U^* = \{U^{i*}, U^{-i*}\}$ such that:

$$\rho^i(s, \theta^{i*}, \theta^{-i*}, U^{i*}, U^{-i*}) \geq \rho^i(s, \theta^i, \theta^{-i*}, U^i, U^{-i*}) \tag{A.5}$$

for all $i \in I$, $s \in S$, $\theta^i \in \Theta^i$, $U^i \in U^i$. This equilibrium concept captures the idea that rational agents should optimize not just their immediate policies but also their adaptation strategies, taking into account the learning dynamics of the system. The active equilibrium generalizes several classic solution concepts in game theory. Stationary Nash equilibria and correlated equilibria are special cases of active equilibria when $k = 1$ (fixed-point convergence) and joint action distributions are independent or correlated, respectively. Similarly, cyclic Nash equilibria and cyclic correlated equilibria can be viewed as special cases of active equilibria when $k > 1$ (periodic behavior), the joint update function is deterministic, and joint action distributions are independent or correlated, respectively. This generality allows the active equilibrium concept to capture a wider range of stable multi-agent behaviors in dynamic learning settings than traditional equilibrium notions.

Active Markov Games address the non-stationarity challenge in several fundamental ways. By incorporating policy parameters and update functions directly into the framework, Active Markov Games make the non-stationarity explicit and amenable to analysis, embracing it as an integral part of the multi-agent learning process rather than treating it as an external challenge to be mitigated. The average reward objective promotes farsighted optimization by encouraging agents to consider the long-term limiting behavior of the system rather than myopically optimizing for immediate or short-term rewards, leading to policies that shape the learning trajectories of other agents in beneficial ways rather than merely reacting to their current behaviors. Furthermore, the framework enables active influence by allowing agents to reason about how their actions affect not just the environment state but also the learning processes of other agents, facilitating sophisticated strategies like teaching, deception, or coordination that explicitly aim to influence other agents' policy updates. Under appropriate conditions, Active Markov Games provide theoretical guarantees about convergence to stationary periodic distributions, offering a more solid foundation for algorithm development than approaches without such guarantees. Finally, the Active Equilibrium concept generalizes traditional game-theoretic solution concepts, providing a more comprehensive framework for understanding stable multi-agent behaviors in learning settings.

A.5.4 Practical Implementations

Recent work has leveraged the Active Markov Game framework to develop practical algorithms for MARL. Notable examples include FURTHER (FULly Reinforcing acTive influence witH averagE Reward) **?**, which implements a policy gradient approach tailored to the average reward objective in Active Markov Games, combined with variational inference for estimating other agents' policy dynamics in a decentralized manner. Another significant approach is Meta-MAPG (Meta-Multiagent Policy Gradient) **?**, which integrates meta-learning with explicit modeling of other agents' learning processes, aligning with the influence-aware perspective of Active Markov Games. These algorithms have demonstrated superior performance compared to methods that neglect the learning dynamics of other agents, particularly in environments with high levels of non-stationarity.

A.6 Extension to Partial Observability

While Active Markov Games provide a powerful framework for addressing non-stationarity, they assume full observability of the environment state. In many real-world scenarios, agents have limited perception capabilities and cannot directly observe the complete state of the environment or the internal parameters of other agents. This partial observability introduces additional challenges for multi-agent learning. The Partially Observable Markov Decision Process (POMDP) **?** extends MDPs to settings with partial observability by introducing observation functions. In the multi-agent context, this leads to Partially Observable Stochastic Games (POSGs) **?** or Decentralized POMDPs (Dec-POMDPs) for cooperative settings **?**. Extending

Active Markov Games to partial observability settings represents a significant advancement in addressing the combined challenges of non-stationarity and limited information in multi-agent learning. The resulting framework, Partially Observable Active Markov Games, incorporates belief states and observation functions while maintaining the key benefits of Active Markov Games for modeling and influencing learning dynamics.

In partially observable settings, policy gradient methods require careful adaptation to handle the unavailability of the true state. Theoretical formulations, as established by Meuleau et al. **?**, extend the policy gradient theorem by conditioning policies on belief states rather than environmental states, resulting in $\nabla_{\theta^i} J(\theta^i) = \mathbb{E}_{s,b}\left[\nabla_{\theta^i} \log \pi^i(a^i|b^i;\theta^i) Q^{\pi}(s,b^i,a^i)\right]$. However, this theoretical construction presents a practical challenge: agents don't have access to the true state distribution needed to compute this expectation. Practical implementations resolve this tension through multiple approaches: history-based methods condition policies on observation histories h_t rather than belief states **?**; recurrent network formulations implicitly encode history using recurrent architectures **?**; and belief-explicit methods maintain and update a belief distribution while estimating values only from observable quantities **?**. These approaches share a common principle of using trajectory sampling to estimate gradients without requiring knowledge of the true state or transition dynamics. In multi-agent settings with partial observability, the challenge compounds as agents must reason about others' belief states and learning dynamics, requiring sophisticated inference mechanisms to estimate other agents' policies and their evolution over time. For Partially Observable Active Markov Games, policy gradients can be formulated to account for the impact of current actions on both the future environmental states and the future policies of other agents, all while operating from belief states rather than full state observations.

PROOFS REGARDING AVERAGE RETURNS

B

This section provides the mathematical foundations and detailed proofs for the key theoretical results in our Partially Observable Active Markov Game framework with average return objectives. We establish the Markov properties of belief transitions and joint processes, prove the existence and uniqueness of stochastically stable distributions, and derive the policy gradient theorem for average return optimization.

B.1 MARKOV PROPERTY OF THE JOINT PROCESS

We first establish the Markov properties of belief state transitions and the joint process comprising states, belief states, and policy parameters. These properties form the foundation for our convergence analysis.

Lemma 1 (Markov Property of Belief Transitions)**.** *The sequence of belief states $\{b_t^i\}_{t\geq 0}$ for each agent i forms a Markov process under a fixed policy π, meaning that the distribution of b_{t+1}^i depends only on b_t^i, a_t^i, and o_{t+1}^i, and not on earlier beliefs, actions, or observations.*

Proof. By definition, the belief state b_t^i at time t incorporates all relevant information from the history of observations and actions up to time t. For Bayesian agents, given b_t^i, action a_t^i, and observation o_{t+1}^i, the belief update equation uniquely determines b_{t+1}^i through the rule:

$$b_{t+1}^i(s') = \frac{O^i(o_{t+1}^i|s') \int_{s\in S} T(s'|s,\boldsymbol{a}_t) b_t^i(s) ds}{\int_{s''\in S} O^i(o_{t+1}^i|s'') \int_{s\in S} T(s''|s,\boldsymbol{a}_t) b_t^i(s) ds ds''} \tag{B.1}$$

where $O_i(o_{t+1}^i|s')$ represents the probability of agent i observing o_{t+1}^i in state s'. This update depends only on b_t^i, a_t^i, and o_{t+1}^i, and not on the sequence of beliefs, actions, and observations that led to b_t^i. Therefore, the belief state transition satisfies the Markov property. □

Remark 1 (Transformer-Based Belief Representation)**.** *In practice, transformer architectures can be used to represent and update belief states. In such cases, the belief state is updated through an attention-based mechanism of the form:*

$$b_{t+1}^i = f_{Transformer}(b_t^i, o_{t+1}^i), \tag{B.2}$$

that is parameterized by $\theta_{Transformer}^i$ and processes the latest observation and previous belief state through self-attention layers, satisfying the Markov property while capturing complex dependencies between observations and beliefs.

Next, we establish the Markov property of the joint process.

Lemma 2 (Markov Property of Joint Process)**.** *The joint process* $(s_t, \boldsymbol{b}_t, \boldsymbol{\theta}_t)$ *forms a Markov process under periodic policy updates, meaning that the distribution of* $(s_{t+1}, \boldsymbol{b}_{t+1}, \boldsymbol{\theta}_{t+1})$ *depends only on* $(s_t, \boldsymbol{b}_t, \boldsymbol{\theta}_t)$ *and not on earlier states.*

Proof. The state transitions are by definition only dependent on current state and current actions. The remaining parts can be analyzed as follows:

1. Belief state transition: As established earlier, $\boldsymbol{b}_{t+1}$ depends only on $\boldsymbol{b}_t$, $\boldsymbol{a}_t$, and $\boldsymbol{o}_{t+1}$. Since actions $\boldsymbol{a}_t$ are drawn from policies $\pi(a_t^i|b_t^i;\theta_t^i)$ that depend only on b_t^i and θ_t^i for each agent i, and observations $\boldsymbol{o}_{t+1}$ depend stochastically on the resulting state, the distribution of $\boldsymbol{b}_{t+1}$ depends only on $\boldsymbol{b}_t$ and $\boldsymbol{\theta}_t$.

2. Policy parameter update: At every time step t, policy parameters are updated according to $U(\boldsymbol{\theta}_{t+1}|\boldsymbol{\theta}_t, \boldsymbol{\tau}_t)$. Under the Markovian policy update assumption, the update depends only on θ_t and current trajectory τ_t, which are functions of s_t, $\boldsymbol{b}_t$ and $\boldsymbol{\theta}_t$.Therefore, the joint transition probability $\mathbb{P}((s_{t+1}, \boldsymbol{b}_{t+1}, \boldsymbol{\theta}_{t+1})|(s_t, \boldsymbol{b}_t, \boldsymbol{\theta}_t))$ depends only on the current state $(s_t, \boldsymbol{b}_t, \boldsymbol{\theta}_t)$ and not on the history of states, establishing the Markov property. □

B.2 CONVERGENCE

Having established the Markov properties, we now analyze the convergence behavior of the joint process to a unique stochastically stable distribution. This convergence result is crucial for developing optimization objectives in our framework.

Theorem 1 (Stochastically Stable Distribution)**.** *Under assumptions (1) and (2), as* $\varepsilon \to 0$*, the perturbed joint processes defined by* ε*-perturbed policy update functions converge to the unique stochastically stable distribution* μ^* *of the unperturbed joint process* $(s_t, \boldsymbol{b}_t, \boldsymbol{\theta}_t)$.

Proof. We establish the convergence to the unique stochastically stable distribution by showing that a perturbed Markov process of a partially observable active Markov game is regular. First, let us introduce a small perturbation to the policy update functions:

$$U_i^{\varepsilon}(\theta_{t+1}^i|\theta_t^i, \tau_t^i) = (1-\varepsilon)U_i(\theta_{t+1}^i|\theta_t^i, \tau_t^i) + \varepsilon\eta_i(\theta_{t+1}^i) \tag{B.3}$$

where η_i is a baseline distribution over Θ_i with full support, and $\varepsilon > 0$ is a small constant. This perturbation ensures that from any policy parameter configuration $\boldsymbol{\theta}_t$, there is a positive probability of transitioning to any other configuration $\boldsymbol{\theta}_{t+1}$. The perturbed joint process $(s_t, \boldsymbol{b}_t, \boldsymbol{\theta}_t)^{\varepsilon}$ evolves according to the transition probability:

$$\mathbb{P}(s_{t+1}, \boldsymbol{b}_{t+1}, \boldsymbol{\theta}_{t+1}|s_t, \boldsymbol{b}_t, \boldsymbol{\theta}_t) = \sum_{\boldsymbol{a}_t \in A} \boldsymbol{\pi}(\boldsymbol{a}_t|\boldsymbol{b}_t; \boldsymbol{\theta}_t) \times T(s_{t+1}|s_t, \boldsymbol{a}_t) \tag{B.4}$$

$$\times \sum_{\boldsymbol{o}_{t+1} \in O} \boldsymbol{O}(\boldsymbol{o}_{t+1}|s_{t+1}) \times \boldsymbol{U}^{\varepsilon}(\boldsymbol{\theta}_{t+1}|\boldsymbol{\theta}_t, \boldsymbol{\tau}_t) \tag{B.5}$$

$$\times \mathbb{I}(\boldsymbol{b}_{t+1} = \text{update}(\boldsymbol{b}_t, \boldsymbol{a}_t, \boldsymbol{o}_{t+1})) \tag{B.6}$$

where $\boldsymbol{\tau}_t = \{\boldsymbol{o}_t, \boldsymbol{a}_t, \boldsymbol{r}_t, \boldsymbol{o}_{t+1}\}$ is the joint trajectory, $\boldsymbol{r}_t = \{R^i(s_t, \boldsymbol{a}_t)\}_{i \in I}$ is the vector of rewards, and update$(\boldsymbol{b}_t, \boldsymbol{a}_t, \boldsymbol{o}_{t+1})$ represents the belief update function for all agents. Suppose, for contradiction, that the perturbed Markov process is irregular, meaning it has multiple recurrent classes. Let C_1 and C_2 be two distinct recurrent classes of the joint process. We will show that there must exist a path with positive probability from any state in C_1 to any state in C_2, contradicting the assumption that they are distinct recurrent classes. Consider any $(s^1, \boldsymbol{b}^1, \boldsymbol{\theta}^1) \in C_1$ and $(s^2, \boldsymbol{b}^2, \boldsymbol{\theta}^2) \in C_2$. We construct a path from the first state to the second as follows:

1. Policy Parameter Transition: Due to the ε-perturbation in update functions, there is a positive probability of directly transitioning from $\boldsymbol{\theta}_1$ to any $\boldsymbol{\theta}' \in \Theta$ in a single step, regardless of the trajectory. Specifically:

$$\mathbb{P}(\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}' | \boldsymbol{\theta}_t = \boldsymbol{\theta}_1, \boldsymbol{\tau}_t) \geq \prod_{i \in I} \varepsilon \cdot \eta_i(\theta'^i) > 0 \tag{B.7}$$

Thus, there is a positive probability path from $\boldsymbol{\theta}_1$ to $\boldsymbol{\theta}_2$.

2. State Transition: By the communicating state assumption, for any two states $s_1, s_2 \in S$, there exists a sequence of joint actions $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_k$ such that:

$$\mathbb{P}(s_{t+k} = s_2 | s_t = s_1, \boldsymbol{a}_t = \boldsymbol{a}_1, \ldots, \boldsymbol{a}_{t+k-1} = \boldsymbol{a}_k) > 0 \tag{B.8}$$

Given the relationship between policies and actions, this sequence of actions has positive probability under any policy parameters $\boldsymbol{\theta}$ and corresponding belief states $\boldsymbol{b}$ such that $\pi^i(a^i | b^i; \theta^i) > 0$ for all required actions.

3. Belief State Transition: By the communicating belief-state assumption, for any two belief states $\boldsymbol{b}_1, \boldsymbol{b}_2 \in B$, there exists a sequence of observations $\boldsymbol{o}_1, \ldots, \boldsymbol{o}_m$ such that:

$$\mathbb{P}(\boldsymbol{b}_{t+m} = \boldsymbol{b}_2 | \boldsymbol{b}_t = \boldsymbol{b}_1, \boldsymbol{o}_{t+1} = \boldsymbol{o}_1, \ldots, \boldsymbol{o}_{t+m} = \boldsymbol{o}_m) > 0 \tag{B.9}$$

Since observations depend on states, which can be influenced through actions, and actions are determined by policies, there is a positive probability path from $\boldsymbol{b}_1$ to $\boldsymbol{b}_2$ under appropriate state transitions and policy parameters. Combining these three components, we can construct a path with positive probability from $(s^1, \boldsymbol{b}^1, \boldsymbol{\theta}^1)$ to $(s^2, \boldsymbol{b}^2, \boldsymbol{\theta}^2)$. This contradicts the assumption that C_1 and C_2 are distinct recurrent classes of the perturbed joint process. Since we have shown that the perturbed joint process has only one recurrent class, it is a regular Markov process. Following (**??**), a regular Markov process on a finite state space possesses a unique stationary distribution μ^ε to which it converges regardless of the initial state. Moreover, as $\varepsilon \to 0$, the sequence of stationary distributions μ^ε converges to a limit μ^*, which is the unique stochastically stable distribution of the original, unperturbed process. Therefore, under the given conditions, the joint process $(s_t, \boldsymbol{b}_t, \boldsymbol{\theta}_t)$ in a partially observable active Markov game possesses a unique stochastically stable distribution μ^*. □

B.3 POLICY GRADIENT THEOREM

Having established the convergence to a unique stochastically stable distribution, we now derive the policy gradient theorem for optimization in our framework. This theorem provides the mathematical foundation for gradient-based algorithms to maximize the average return objective.

Theorem 2 (Partially Observable Active Average Reward Policy Gradient Theorem)**.** *The gradient of the active average reward objective with respect to agent i's policy parameters θ^i in a partially observable setting is:*

$$\nabla_{\theta^i} J^i_\pi(\theta^i) = \sum_{s,\boldsymbol{b},\boldsymbol{\theta}} \mu_{\theta^i}(s,\boldsymbol{b},\boldsymbol{\theta}) \sum_{a^i} \nabla_{\theta^i} \pi^i(a^i|b^i;\theta^i) \sum_{a^{\neg i}} \pi^{\neg i}(a^{\neg i}|b^{\neg i};\theta^{\neg i}) q^i_{\theta^i}(s,\boldsymbol{b},\boldsymbol{\theta},\boldsymbol{a}) \tag{B.10}$$

where the action-value function $q^i_{\theta^i}$ is defined as:

$$q^i_{\theta^i}(s,\boldsymbol{b},\boldsymbol{\theta},\boldsymbol{a}) = \sum_{s'} T(s'|s,\boldsymbol{a}) \sum_{\boldsymbol{o}'} \boldsymbol{O}(\boldsymbol{o}'|s') \sum_{\boldsymbol{\theta}'} \boldsymbol{U}(\boldsymbol{\theta}'|\boldsymbol{\theta},\boldsymbol{\tau}) \left[R^i(s,\boldsymbol{a}) - \rho^i_{\theta^i} + v^i_{\theta^i}(s', update(\boldsymbol{b},\boldsymbol{a},\boldsymbol{o}'),\boldsymbol{\theta}') \right] \tag{B.11}$$

with $update(\boldsymbol{b},\boldsymbol{a},\boldsymbol{o}')$ representing the belief update function.

Proof. We first define the average reward objective in the partially observable setting:

$$\max_{\theta^i} \rho^i_{\theta^i}(\boldsymbol{b},\boldsymbol{\theta}) := \max_{\theta^i} \lim_{T\to\infty} \mathbb{E}\left[\frac{1}{T} \sum_{t=0}^{T} R^i(s_t,\boldsymbol{a}_t) \,\middle|\, \begin{array}{c} \boldsymbol{b}_0 = \boldsymbol{b}, \boldsymbol{\theta}_0 = \boldsymbol{\theta}, \\ \boldsymbol{a}_{0:T} \sim \boldsymbol{\pi}(\cdot|\boldsymbol{b}_{0:T};\boldsymbol{\theta}_{0:T}), s_{t+1} \sim T(\cdot|s_t,\boldsymbol{a}_t), \\ \boldsymbol{o}_{t+1} \sim \boldsymbol{O}(\cdot|s_{t+1}), \boldsymbol{\theta}_{t+1} \sim \boldsymbol{U}(\cdot|\boldsymbol{\theta}_t,\boldsymbol{\tau}_t) \end{array} \right] \tag{B.12}$$

$$= \max_{\theta^i} \sum_{s,\boldsymbol{b},\boldsymbol{\theta}} b^i(s) \mu_{\theta^i}(s,\boldsymbol{b},\boldsymbol{\theta}) \sum_{\boldsymbol{a}} \boldsymbol{\pi}(\boldsymbol{a}|\boldsymbol{b};\boldsymbol{\theta}) R^i(s,\boldsymbol{a}) \tag{B.13}$$

where $\mu_{\theta^i}(s,\boldsymbol{b},\boldsymbol{\theta})$ is the stationary distribution of the joint process $(s_t,\boldsymbol{b}_t,\boldsymbol{\theta}_t)$ under agent i's policy parameterized by θ^i. The differential value function $v^i_{\theta^i}$ represents the expected total difference between the accumulated rewards from state s, belief states $\boldsymbol{b}$, and policy parameters $\boldsymbol{\theta}$, and the average reward $\rho^i_{\theta^i}$:

$$v^i_{\theta^i}(s,\boldsymbol{b},\boldsymbol{\theta}) = \lim_{T\to\infty} \mathbb{E}\left[\sum_{t=0}^{T} \left(R^i(s_t,\boldsymbol{a}_t) - \rho^i_{\theta^i} \right) \,\middle|\, \begin{array}{c} s_0 = s, \boldsymbol{b}_0 = \boldsymbol{b}, \boldsymbol{\theta}_0 = \boldsymbol{\theta}, \\ \boldsymbol{a}_{0:T} \sim \boldsymbol{\pi}(\cdot|\boldsymbol{b}_{0:T};\boldsymbol{\theta}_{0:T}), s_{t+1} \sim T(\cdot|s_t,\boldsymbol{a}_t), \\ \boldsymbol{o}_{t+1} \sim \boldsymbol{O}(\cdot|s_{t+1}), \boldsymbol{\theta}_{t+1} \sim \boldsymbol{U}(\cdot|\boldsymbol{\theta}_t,\boldsymbol{\tau}_t) \end{array} \right] \tag{B.14}$$

Following the Bellman equation derivation principles, we can express $v^i_{\theta^i}(s,\boldsymbol{b},\boldsymbol{\theta})$ recur-

sively:

$$v^i_{\theta^i}(s,\boldsymbol{b},\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{a}\sim\boldsymbol{\pi}(\cdot|\boldsymbol{b};\boldsymbol{\theta}),s'\sim T(\cdot|s,\boldsymbol{a}),\boldsymbol{o}'\sim\boldsymbol{O}(\cdot|s'),\boldsymbol{\theta}'\sim\boldsymbol{U}(\cdot|\boldsymbol{\theta},\boldsymbol{\tau})}\left[R^i(s,\boldsymbol{a}) - \rho^i_{\theta^i} + v^i_{\theta^i}(s',\text{update}(\boldsymbol{b},\boldsymbol{a},\boldsymbol{o}'),\boldsymbol{\theta}')\right] \tag{B.15}$$

$$= \sum_{a^i}\pi^i(a^i|b^i;\theta^i)\sum_{a^{\neg i}}\pi^{\neg i}(a^{\neg i}|b^{\neg i};\theta^{\neg i})\sum_{s'}T(s'|s,\boldsymbol{a})\sum_{\boldsymbol{o}'}\boldsymbol{O}(\boldsymbol{o}'|s')\sum_{\boldsymbol{\theta}'}\boldsymbol{U}(\boldsymbol{\theta}'|\boldsymbol{\theta},\boldsymbol{\tau}) \tag{B.16}$$

$$\left[R^i(s,\boldsymbol{a}) - \rho^i_{\theta^i} + v^i_{\theta^i}(s',\text{update}(\boldsymbol{b},\boldsymbol{a},\boldsymbol{o}'),\boldsymbol{\theta}')\right] \tag{B.17}$$

where update$(\boldsymbol{b},\boldsymbol{a},\boldsymbol{o}')$ represents the belief update function that updates the belief state based on the current belief, action, and new observation. We now define the action-value function $q^i_{\theta^i}$ for the partially observable setting:

$$q^i_{\theta^i}(s,\boldsymbol{b},\boldsymbol{\theta},\boldsymbol{a}) = \sum_{s'}T(s'|s,\boldsymbol{a})\sum_{\boldsymbol{o}'}\boldsymbol{O}(\boldsymbol{o}'|s')\sum_{\boldsymbol{\theta}'}\boldsymbol{U}(\boldsymbol{\theta}'|\boldsymbol{\theta},\boldsymbol{\tau})\left[R^i(s,\boldsymbol{a}) - \rho^i_{\theta^i} + v^i_{\theta^i}(s',\text{update}(\boldsymbol{b},\boldsymbol{a},\boldsymbol{o}'),\boldsymbol{\theta}')\right] \tag{B.18}$$

Using this definition, we can rewrite the differential value function as:

$$v^i_{\theta^i}(s,\boldsymbol{b},\boldsymbol{\theta}) = \sum_{a^i}\pi^i(a^i|b^i;\theta^i)\sum_{a^{\neg i}}\pi^{\neg i}(a^{\neg i}|b^{\neg i};\theta^{\neg i})q^i_{\theta^i}(s,\boldsymbol{b},\boldsymbol{\theta},\boldsymbol{a}) \tag{B.19}$$

To derive the policy gradient, we begin by computing the gradient of the differential value function with respect to θ^i:

$$\nabla_{\theta^i}v^i_{\theta^i}(s,\boldsymbol{b},\boldsymbol{\theta}) = \nabla_{\theta^i}\left[\sum_{a^i}\pi^i(a^i|b^i;\theta^i)\sum_{a^{\neg i}}\pi^{\neg i}(a^{\neg i}|b^{\neg i};\theta^{\neg i})q^i_{\theta^i}(s,\boldsymbol{b},\boldsymbol{\theta},\boldsymbol{a})\right] \tag{B.20}$$

$$= \sum_{a^i}\nabla_{\theta^i}\pi^i(a^i|b^i;\theta^i)\sum_{a^{\neg i}}\pi^{\neg i}(a^{\neg i}|b^{\neg i};\theta^{\neg i})q^i_{\theta^i}(s,\boldsymbol{b},\boldsymbol{\theta},\boldsymbol{a}) + \tag{B.21}$$

$$\sum_{a^i}\pi^i(a^i|b^i;\theta^i)\sum_{a^{\neg i}}\pi^{\neg i}(a^{\neg i}|b^{\neg i};\theta^{\neg i})\nabla_{\theta^i}q^i_{\theta^i}(s,\boldsymbol{b},\boldsymbol{\theta},\boldsymbol{a}) \tag{B.22}$$

We expand the gradient of the action-value function:

$$\nabla_{\theta^i}q^i_{\theta^i}(s,\boldsymbol{b},\boldsymbol{\theta},\boldsymbol{a}) = \nabla_{\theta^i}\left[\sum_{s'}T(s'|s,\boldsymbol{a})\sum_{\boldsymbol{o}'}\boldsymbol{O}(\boldsymbol{o}'|s')\right. \tag{B.23}$$

$$\left.\sum_{\boldsymbol{\theta}'}\boldsymbol{U}(\boldsymbol{\theta}'|\boldsymbol{\theta},\boldsymbol{\tau})\left[R^i(s,\boldsymbol{a}) - \rho^i_{\theta^i} + v^i_{\theta^i}(s',\text{update}(\boldsymbol{b},\boldsymbol{a},\boldsymbol{o}'),\boldsymbol{\theta}')\right]\right] \tag{B.24}$$

$$= -\nabla_{\theta^i}\rho^i_{\theta^i} + \sum_{s'}T(s'|s,\boldsymbol{a})\sum_{\boldsymbol{o}'}\boldsymbol{O}(\boldsymbol{o}'|s')\sum_{\boldsymbol{\theta}'}\boldsymbol{U}(\boldsymbol{\theta}'|\boldsymbol{\theta},\boldsymbol{\tau})\nabla_{\theta^i}v^i_{\theta^i}(s',\text{update}(\boldsymbol{b},\boldsymbol{a},\boldsymbol{o}'),\boldsymbol{\theta}') \tag{B.25}$$

Substituting back into the original expression, we get:

$$\nabla_{\theta^i} v^i_{\theta^i}(s,\boldsymbol{b},\boldsymbol{\theta}) = \sum_{a^i} \nabla_{\theta^i} \pi^i(a^i|b^i;\theta^i) \sum_{a^{\neg i}} \pi^{\neg i}(a^{\neg i}|b^{\neg i};\theta^{\neg i}) q^i_{\theta^i}(s,\boldsymbol{b},\boldsymbol{\theta},\boldsymbol{a}) \tag{B.26}$$

$$- \sum_{a^i} \pi^i(a^i|b^i;\theta^i) \sum_{a^{\neg i}} \pi^{\neg i}(a^{\neg i}|b^{\neg i};\theta^{\neg i}) \nabla_{\theta^i} \rho^i_{\theta^i} \tag{B.27}$$

$$+ \sum_{a^i} \pi^i(a^i|b^i;\theta^i) \sum_{a^{\neg i}} \pi^{\neg i}(a^{\neg i}|b^{\neg i};\theta^{\neg i}) \sum_{s'} T(s'|s,\boldsymbol{a}) \sum_{\boldsymbol{o}'} \boldsymbol{O}(\boldsymbol{o}'|s') \tag{B.28}$$

$$\sum_{\boldsymbol{\theta}'} \boldsymbol{U}(\boldsymbol{\theta}'|\boldsymbol{\theta},\boldsymbol{\tau}) \nabla_{\theta^i} v^i_{\theta^i}(s', \text{update}(\boldsymbol{b},\boldsymbol{a},\boldsymbol{o}'),\boldsymbol{\theta}') \tag{B.29}$$

Rearranging terms:

$$\nabla_{\theta^i} \rho^i_{\theta^i} = \sum_{a^i} \nabla_{\theta^i} \pi^i(a^i|b^i;\theta^i) \sum_{a^{\neg i}} \pi^{\neg i}(a^{\neg i}|b^{\neg i};\theta^{\neg i}) q^i_{\theta^i}(s,\boldsymbol{b},\boldsymbol{\theta},\boldsymbol{a}) \tag{B.30}$$

$$+ \sum_{a^i} \pi^i(a^i|b^i;\theta^i) \sum_{a^{\neg i}} \pi^{\neg i}(a^{\neg i}|b^{\neg i};\theta^{\neg i}) \sum_{s'} T(s'|s,\boldsymbol{a}) \sum_{\boldsymbol{o}'} \boldsymbol{O}(\boldsymbol{o}'|s') \tag{B.31}$$

$$\sum_{\boldsymbol{\theta}'} \boldsymbol{U}(\boldsymbol{\theta}'|\boldsymbol{\theta},\boldsymbol{\tau}) \nabla_{\theta^i} v^i_{\theta^i}(s', \text{update}(\boldsymbol{b},\boldsymbol{a},\boldsymbol{o}'),\boldsymbol{\theta}') \tag{B.32}$$

$$- \nabla_{\theta^i} v^i_{\theta^i}(s,\boldsymbol{b},\boldsymbol{\theta}) \tag{B.33}$$

Now, we define the transition operator that maps the expectation of a function from one time step to the next:

$$\Psi f(s,\boldsymbol{b},\boldsymbol{\theta}) = \sum_{a^i} \pi^i(a^i|b^i;\theta^i) \sum_{a^{\neg i}} \pi^{\neg i}(a^{\neg i}|b^{\neg i};\theta^{\neg i}) \tag{B.34}$$

$$\sum_{s'} T(s'|s,\boldsymbol{a}) \sum_{\boldsymbol{o}'} \boldsymbol{O}(\boldsymbol{o}'|s') \sum_{\boldsymbol{\theta}'} \boldsymbol{U}(\boldsymbol{\theta}'|\boldsymbol{\theta},\boldsymbol{\tau}) f(s', \text{update}(\boldsymbol{b},\boldsymbol{a},\boldsymbol{o}'),\boldsymbol{\theta}') \tag{B.35}$$

We can rewrite our expression as:

$$\nabla_{\theta^i} \rho^i_{\theta^i} = \sum_{a^i} \nabla_{\theta^i} \pi^i(a^i|b^i;\theta^i) \sum_{a^{\neg i}} \pi^{\neg i}(a^{\neg i}|b^{\neg i};\theta^{\neg i}) q^i_{\theta^i}(s,\boldsymbol{b},\boldsymbol{\theta},\boldsymbol{a}) + \Psi(\nabla_{\theta^i} v^i_{\theta^i})(s,\boldsymbol{b},\boldsymbol{\theta}) - \nabla_{\theta^i} v^i_{\theta^i}(s,\boldsymbol{b},\boldsymbol{\theta}) \tag{B.36}$$

Now, we take the expectation with respect to the stationary distribution $\mu_{\theta^i}(s,\boldsymbol{b},\boldsymbol{\theta})$:

$$\sum_{s,\boldsymbol{b},\boldsymbol{\theta}} \mu_{\theta^i}(s,\boldsymbol{b},\boldsymbol{\theta}) \nabla_{\theta^i} \rho^i_{\theta^i} = \sum_{s,\boldsymbol{b},\boldsymbol{\theta}} \mu_{\theta^i}(s,\boldsymbol{b},\boldsymbol{\theta}) \sum_{a^i} \nabla_{\theta^i} \pi^i(a^i|b^i;\theta^i) \sum_{a^{\neg i}} \pi^{\neg i}(a^{\neg i}|b^{\neg i};\theta^{\neg i}) q^i_{\theta^i}(s,\boldsymbol{b},\boldsymbol{\theta},\boldsymbol{a}) \tag{B.37}$$

$$+ \sum_{s,\boldsymbol{b},\boldsymbol{\theta}} \mu_{\theta^i}(s,\boldsymbol{b},\boldsymbol{\theta}) \Psi(\nabla_{\theta^i} v^i_{\theta^i})(s,\boldsymbol{b},\boldsymbol{\theta}) - \sum_{s,\boldsymbol{b},\boldsymbol{\theta}} \mu_{\theta^i}(s,\boldsymbol{b},\boldsymbol{\theta}) \nabla_{\theta^i} v^i_{\theta^i}(s,\boldsymbol{b},\boldsymbol{\theta}) \tag{B.38}$$

By the definition of the stationary distribution, we have:

$$\sum_{s,\boldsymbol{b},\boldsymbol{\theta}} \mu_{\theta^i}(s,\boldsymbol{b},\boldsymbol{\theta})\Psi f(s,\boldsymbol{b},\boldsymbol{\theta}) = \sum_{s,\boldsymbol{b},\boldsymbol{\theta}} \mu_{\theta^i}(s,\boldsymbol{b},\boldsymbol{\theta}) f(s,\boldsymbol{b},\boldsymbol{\theta}) \tag{B.39}$$

Therefore, the second and third terms in our expression cancel out, giving the final policy gradient theorem:

$$\nabla_{\theta^i} J^i_\pi(\theta^i) := \nabla_{\theta^i} \rho^i_{\theta^i} = \sum_{s,\boldsymbol{b},\boldsymbol{\theta}} \mu_{\theta^i}(s,\boldsymbol{b},\boldsymbol{\theta}) \sum_{a^i} \nabla_{\theta^i} \pi^i(a^i|b^i;\theta^i) \sum_{a^{\neg i}} \pi^{\neg i}(a^{\neg i}|b^{\neg i};\theta^{\neg i}) q^i_{\theta^i}(s,\boldsymbol{b},\boldsymbol{\theta},\boldsymbol{a}) \tag{B.40}$$

□

C PROOFS REGARDING DISCOUNTED RETURNS

We establish the Partially Observable Active Discounted Return Policy Gradient Theorem through a sequence of lemmas that build the necessary mathematical foundation.

C.1 TRANSITION OPERATOR AND ITS ADJOINT

To establish a rigorous framework for analyzing discounted returns in Partially Observable Active Markov Games, we need to formalize how value functions evolve over time and how probability distributions propagate through the system. This dual perspective is captured by two fundamental operators: the transition operator and its adjoint.

C.1.1 Definitions and Duality

We begin by defining the inner product between functions and measures, which forms the foundation for the duality relationship between our operators.

Definition 8 (Inner Product between Functions and Measures)**.** *For a bounded measurable function $f \in \mathscr{B}(S \times \boldsymbol{B} \times \boldsymbol{\Theta})$ and a finite measure $\mu \in \mathscr{M}(S \times \boldsymbol{B} \times \boldsymbol{\Theta})$, their inner product is defined as the Lebesgue integral:*

$$\langle f, \mu \rangle = \int_{S \times \boldsymbol{B} \times \boldsymbol{\Theta}} f(s, \boldsymbol{b}, \boldsymbol{\theta}) \, d\mu(s, \boldsymbol{b}, \boldsymbol{\theta}) \tag{C.1}$$

This inner product captures the expected value of function f with respect to measure μ, allowing us to express value functions as expectations with respect to appropriate measures.

Definition 9 (Transition Operator and Its Adjoint)**.**

1. *The **transition operator** $\Psi : \mathscr{B}(S \times \boldsymbol{B} \times \boldsymbol{\Theta}) \to \mathscr{B}(S \times \boldsymbol{B} \times \boldsymbol{\Theta})$ maps a bounded measurable function to its expected value after one transition:*

$$(\Psi f)(s, \boldsymbol{b}, \boldsymbol{\theta}) = \sum_{a^i} \pi^i(a^i | b^i; \theta^i) \sum_{a^{\neg i}} \pi^{\neg i}(a^{\neg i} | b^{\neg i}; \theta^{\neg i}) \sum_{s'} T(s' | s, \boldsymbol{a}) \tag{C.2}$$

$$\sum_{\boldsymbol{o}'} \boldsymbol{O}(\boldsymbol{o}' | s') \sum_{\boldsymbol{\theta}'} \boldsymbol{U}(\boldsymbol{\theta}' | \boldsymbol{\theta}, \boldsymbol{\tau}) f(s', \mathit{update}(\boldsymbol{b}, \boldsymbol{a}, \boldsymbol{o}'), \boldsymbol{\theta}') \tag{C.3}$$

   where $\mathit{update}(\boldsymbol{b}, \boldsymbol{a}, \boldsymbol{o}')$ represents the belief update function.

2. *The **adjoint operator** $\Psi^* : \mathscr{M}(S \times \boldsymbol{B} \times \boldsymbol{\Theta}) \to \mathscr{M}(S \times \boldsymbol{B} \times \boldsymbol{\Theta})$ is defined on the space of finite measures such that:*

$$\langle \Psi f, \mu \rangle = \langle f, \Psi^* \mu \rangle \tag{C.4}$$

for all $f \in \mathscr{B}(S \times \boldsymbol{B} \times \boldsymbol{\Theta})$ *and* $\mu \in \mathscr{M}(S \times \boldsymbol{B} \times \boldsymbol{\Theta})$.

Intuitively, Ψ describes how function expectations evolve forward in time, meanwhile, Ψ^* describes how probability distributions evolve in time, capturing the propagation of visitation probabilities through the system. The duality relationship ensures that expected values can be computed either by applying Ψ to functions or Ψ^* to measures.

C.1.2 Properties of the Operators

These operators possess several important properties that facilitate our analysis of discounted returns:

Lemma 3 (Properties of the Transition Operator and Its Adjoint)**.** *The transition operator* Ψ *and its adjoint* Ψ^* *satisfy the following properties:*

1. ***Linearity:***
   - *For the operator: For any functions* $f, g \in \mathscr{B}(S \times \boldsymbol{B} \times \boldsymbol{\Theta})$ *and constants* $\alpha, \beta \in \mathbb{R}$*:*

$$\Psi(\alpha f + \beta g) = \alpha \Psi f + \beta \Psi g \tag{C.5}$$

   - *For the adjoint: For any measures* $\mu, \nu \in \mathscr{M}(S \times \boldsymbol{B} \times \boldsymbol{\Theta})$ *and constants* $\alpha, \beta \in \mathbb{R}$*:*

$$\Psi^*(\alpha \mu + \beta \nu) = \alpha \Psi^* \mu + \beta \Psi^* \nu \tag{C.6}$$

2. ***Positivity Preservation:***
   - *For the operator: If* $f \geq 0$*, then* $\Psi f \geq 0$*.*
   - *For the adjoint: If* μ *is a non-negative measure, then* $\Psi^* \mu$ *is also non-negative.*
3. ***Preservation of Total Measure:***
   - *For the operator: The operator preserves the constant function* $\mathbf{1}$*:*

$$\Psi \mathbf{1} = \mathbf{1} \tag{C.7}$$

     where $\mathbf{1}$ *represents the function that equals 1 everywhere.*
   - *For the adjoint: For any measure* $\mu \in \mathscr{M}(S \times \boldsymbol{B} \times \boldsymbol{\Theta})$*:*

$$(\Psi^* \mu)(S \times \boldsymbol{B} \times \boldsymbol{\Theta}) = \mu(S \times \boldsymbol{B} \times \boldsymbol{\Theta}) \tag{C.8}$$

4. ***Bound Preservation:*** *For any bounded function* f*,* Ψf *is bounded with:*

$$\|\Psi f\|_\infty \leq \|f\|_\infty \tag{C.9}$$

where $\|f\|_\infty = \sup_{s, \boldsymbol{b}, \boldsymbol{\theta}} |f(s, \boldsymbol{b}, \boldsymbol{\theta})|$ *is the supremum norm.*

5. ***Explicit Form of the Adjoint:*** *For any measurable set $A \subset S \times \boldsymbol{B} \times \boldsymbol{\Theta}$:*

$$(\Psi^* \mu)(A) = \int_{S \times \boldsymbol{B} \times \boldsymbol{\Theta}} \sum_{a^i} \pi^i(a^i|b^i;\theta^i) \sum_{a^{\neg i}} \pi^{\neg i}(a^{\neg i}|b^{\neg i};\theta^{\neg i}) \tag{C.10}$$

$$\sum_{s'} T(s'|s,\boldsymbol{a}) \sum_{\boldsymbol{o}'} \boldsymbol{O}(\boldsymbol{o}'|s') \sum_{\boldsymbol{\theta}'} \boldsymbol{U}(\boldsymbol{\theta}'|\boldsymbol{\theta},\boldsymbol{\tau}) \mathbf{1}_A(s', update(\boldsymbol{b},\boldsymbol{a},\boldsymbol{o}'),\boldsymbol{\theta}')\, d\mu(s,\boldsymbol{b},\boldsymbol{\theta}) \tag{C.11}$$

where $\mathbf{1}_A$ is the indicator function for set A.

Proof. We prove each property in turn:

1. Linearity:

For the operator: For any $(s,\boldsymbol{b},\boldsymbol{\theta})$, we have:

$$\Psi(\alpha f + \beta g)(s,\boldsymbol{b},\boldsymbol{\theta}) \tag{C.12}$$

$$= \sum_{a^i} \pi^i(a^i|b^i;\theta^i) \sum_{a^{\neg i}} \pi^{\neg i}(a^{\neg i}|b^{\neg i};\theta^{\neg i}) \sum_{s'} T(s'|s,\boldsymbol{a}) \sum_{\boldsymbol{o}'} \boldsymbol{O}(\boldsymbol{o}'|s') \tag{C.13}$$

$$\sum_{\boldsymbol{\theta}'} \boldsymbol{U}(\boldsymbol{\theta}'|\boldsymbol{\theta},\boldsymbol{\tau})[\alpha f + \beta g](s', \text{update}(\boldsymbol{b},\boldsymbol{a},\boldsymbol{o}'),\boldsymbol{\theta}') \tag{C.14}$$

$$= \alpha \sum_{a^i} \pi^i(a^i|b^i;\theta^i) \sum_{a^{\neg i}} \pi^{\neg i}(a^{\neg i}|b^{\neg i};\theta^{\neg i}) \sum_{s'} T(s'|s,\boldsymbol{a}) \sum_{\boldsymbol{o}'} \boldsymbol{O}(\boldsymbol{o}'|s') \tag{C.15}$$

$$\sum_{\boldsymbol{\theta}'} \boldsymbol{U}(\boldsymbol{\theta}'|\boldsymbol{\theta},\boldsymbol{\tau}) f(s', \text{update}(\boldsymbol{b},\boldsymbol{a},\boldsymbol{o}'),\boldsymbol{\theta}') \tag{C.16}$$

$$+ \beta \sum_{a^i} \pi^i(a^i|b^i;\theta^i) \sum_{a^{\neg i}} \pi^{\neg i}(a^{\neg i}|b^{\neg i};\theta^{\neg i}) \sum_{s'} T(s'|s,\boldsymbol{a}) \sum_{\boldsymbol{o}'} \boldsymbol{O}(\boldsymbol{o}'|s') \tag{C.17}$$

$$\sum_{\boldsymbol{\theta}'} \boldsymbol{U}(\boldsymbol{\theta}'|\boldsymbol{\theta},\boldsymbol{\tau}) g(s', \text{update}(\boldsymbol{b},\boldsymbol{a},\boldsymbol{o}'),\boldsymbol{\theta}') \tag{C.18}$$

$$= \alpha(\Psi f)(s,\boldsymbol{b},\boldsymbol{\theta}) + \beta(\Psi g)(s,\boldsymbol{b},\boldsymbol{\theta}) \tag{C.19}$$

For the adjoint: For any measurable function $f \in \mathscr{B}(S \times \boldsymbol{B} \times \boldsymbol{\Theta})$:

$$\langle f, \Psi^*(\alpha\mu + \beta\nu)\rangle = \langle \Psi f, \alpha\mu + \beta\nu\rangle \tag{C.20}$$

$$= \alpha\langle \Psi f, \mu\rangle + \beta\langle \Psi f, \nu\rangle \tag{C.21}$$

$$= \alpha\langle f, \Psi^*\mu\rangle + \beta\langle f, \Psi^*\nu\rangle \tag{C.22}$$

$$= \langle f, \alpha\Psi^*\mu + \beta\Psi^*\nu\rangle \tag{C.23}$$

Since this holds for any measurable function f, we have $\Psi^*(\alpha\mu + \beta\nu) = \alpha\Psi^*\mu + \beta\Psi^*\nu$ by the uniqueness of the representing measure.

2. Positivity Preservation:

For the operator: If $f \geq 0$, then for any $(s, \boldsymbol{b}, \boldsymbol{\theta})$:

$$(\Psi f)(s, \boldsymbol{b}, \boldsymbol{\theta}) = \sum_{a^i} \pi^i(a^i|b^i;\theta^i) \sum_{a^{\neg i}} \pi^{\neg i}(a^{\neg i}|b^{\neg i};\theta^{\neg i}) \sum_{s'} T(s'|s,\boldsymbol{a}) \sum_{\boldsymbol{o}'} \boldsymbol{O}(\boldsymbol{o}'|s') \tag{C.24}$$

$$\sum_{\boldsymbol{\theta}'} \boldsymbol{U}(\boldsymbol{\theta}'|\boldsymbol{\theta},\boldsymbol{\tau}) f(s', \text{update}(\boldsymbol{b},\boldsymbol{a},\boldsymbol{o}'), \boldsymbol{\theta}') \tag{C.25}$$

Since $f \geq 0$ and all probability distributions π^i, $\pi^{\neg i}$, T, $\boldsymbol{O}$, and $\boldsymbol{U}$ are non-negative, the entire sum consists of non-negative terms, making $(\Psi f)(s, \boldsymbol{b}, \boldsymbol{\theta}) \geq 0$.

For the adjoint: If μ is non-negative, then for any non-negative measurable function $f \geq 0$:

$$\langle f, \Psi^* \mu \rangle = \langle \Psi f, \mu \rangle \geq 0 \tag{C.26}$$

where the inequality follows from the positivity preservation of Ψ and the fact that μ is non-negative. Since this holds for all non-negative functions f, $\Psi^* \mu$ must be a non-negative measure.

3. Preservation of Total Measure:

For the operator: Let $\mathbf{1}$ be the constant function that equals 1 everywhere. Then:

$$(\Psi \mathbf{1})(s, \boldsymbol{b}, \boldsymbol{\theta}) = \sum_{a^i} \pi^i(a^i|b^i;\theta^i) \sum_{a^{\neg i}} \pi^{\neg i}(a^{\neg i}|b^{\neg i};\theta^{\neg i}) \sum_{s'} T(s'|s,\boldsymbol{a}) \sum_{\boldsymbol{o}'} \boldsymbol{O}(\boldsymbol{o}'|s') \tag{C.27}$$

$$\sum_{\boldsymbol{\theta}'} \boldsymbol{U}(\boldsymbol{\theta}'|\boldsymbol{\theta},\boldsymbol{\tau}) \cdot 1 \tag{C.28}$$

Since each probability distribution sums to 1, applying them sequentially yields:

$$(\Psi \mathbf{1})(s, \boldsymbol{b}, \boldsymbol{\theta}) = \sum_{a^i} \pi^i(a^i|b^i;\theta^i) \sum_{a^{\neg i}} \pi^{\neg i}(a^{\neg i}|b^{\neg i};\theta^{\neg i}) \sum_{s'} T(s'|s,\boldsymbol{a}) \sum_{\boldsymbol{o}'} \boldsymbol{O}(\boldsymbol{o}'|s') \cdot 1 \tag{C.29}$$

$$= \sum_{a^i} \pi^i(a^i|b^i;\theta^i) \sum_{a^{\neg i}} \pi^{\neg i}(a^{\neg i}|b^{\neg i};\theta^{\neg i}) \sum_{s'} T(s'|s,\boldsymbol{a}) \cdot 1 \tag{C.30}$$

$$= \sum_{a^i} \pi^i(a^i|b^i;\theta^i) \sum_{a^{\neg i}} \pi^{\neg i}(a^{\neg i}|b^{\neg i};\theta^{\neg i}) \cdot 1 \tag{C.31}$$

$$= \sum_{a^i} \pi^i(a^i|b^i;\theta^i) \cdot 1 = 1 \tag{C.32}$$

For the adjoint: Setting $f = \mathbf{1}$ in the adjoint relationship:

$$\langle \mathbf{1}, \Psi^* \mu \rangle = \langle \Psi \mathbf{1}, \mu \rangle = \langle \mathbf{1}, \mu \rangle \tag{C.33}$$

where we used the unit preservation property of Ψ. This gives:

$$(\Psi^* \mu)(S \times \boldsymbol{B} \times \boldsymbol{\Theta}) = \mu(S \times \boldsymbol{B} \times \boldsymbol{\Theta}) \tag{C.34}$$

4. Bound Preservation:

For any bounded function f with $\|f\|_\infty = \sup_{s,\boldsymbol{b},\boldsymbol{\theta}} |f(s,\boldsymbol{b},\boldsymbol{\theta})|$:

$$|(\Psi f)(s,\boldsymbol{b},\boldsymbol{\theta})| = \Bigg| \sum_{a^i} \pi^i(a^i|b^i;\theta^i) \sum_{a^{\neg i}} \pi^{\neg i}(a^{\neg i}|b^{\neg i};\theta^{\neg i}) \sum_{s'} T(s'|s,\boldsymbol{a}) \sum_{\boldsymbol{o}'} \boldsymbol{O}(\boldsymbol{o}'|s') \tag{C.35}$$

$$\sum_{\boldsymbol{\theta}'} \boldsymbol{U}(\boldsymbol{\theta}'|\boldsymbol{\theta},\boldsymbol{\tau}) f(s', \text{update}(\boldsymbol{b},\boldsymbol{a},\boldsymbol{o}'), \boldsymbol{\theta}') \Bigg| \tag{C.36}$$

$$\leq \sum_{a^i} \pi^i(a^i|b^i;\theta^i) \sum_{a^{\neg i}} \pi^{\neg i}(a^{\neg i}|b^{\neg i};\theta^{\neg i}) \sum_{s'} T(s'|s,\boldsymbol{a}) \sum_{\boldsymbol{o}'} \boldsymbol{O}(\boldsymbol{o}'|s') \tag{C.37}$$

$$\sum_{\boldsymbol{\theta}'} \boldsymbol{U}(\boldsymbol{\theta}'|\boldsymbol{\theta},\boldsymbol{\tau}) \left| f(s', \text{update}(\boldsymbol{b},\boldsymbol{a},\boldsymbol{o}'), \boldsymbol{\theta}') \right| \tag{C.38}$$

$$\leq \|f\|_\infty \sum_{a^i} \pi^i(a^i|b^i;\theta^i) \sum_{a^{\neg i}} \pi^{\neg i}(a^{\neg i}|b^{\neg i};\theta^{\neg i}) \sum_{s'} T(s'|s,\boldsymbol{a}) \sum_{\boldsymbol{o}'} \boldsymbol{O}(\boldsymbol{o}'|s') \tag{C.39}$$

$$\sum_{\boldsymbol{\theta}'} \boldsymbol{U}(\boldsymbol{\theta}'|\boldsymbol{\theta},\boldsymbol{\tau}) = \|f\|_\infty \cdot 1 = \|f\|_\infty \tag{C.40}$$

Taking the supremum over all $(s,\boldsymbol{b},\boldsymbol{\theta})$, we get $\|\Psi f\|_\infty \leq \|f\|_\infty$.

5. Explicit Form of the Adjoint:

For any measurable set $A \subset S \times \boldsymbol{B} \times \boldsymbol{\Theta}$:

$$(\Psi^* \mu)(A) = \langle \mathbf{1}_A, \Psi^* \mu \rangle = \langle \Psi \mathbf{1}_A, \mu \rangle \tag{C.41}$$

Expanding $\Psi \mathbf{1}_A$ using the definition of the transition operator:

$$(\Psi \mathbf{1}_A)(s,\boldsymbol{b},\boldsymbol{\theta}) = \sum_{a^i} \pi^i(a^i|b^i;\theta^i) \sum_{a^{\neg i}} \pi^{\neg i}(a^{\neg i}|b^{\neg i};\theta^{\neg i}) \sum_{s'} T(s'|s,\boldsymbol{a}) \tag{C.42}$$

$$\sum_{\boldsymbol{o}'} \boldsymbol{O}(\boldsymbol{o}'|s') \sum_{\boldsymbol{\theta}'} \boldsymbol{U}(\boldsymbol{\theta}'|\boldsymbol{\theta},\boldsymbol{\tau}) \mathbf{1}_A(s', \text{update}(\boldsymbol{b},\boldsymbol{a},\boldsymbol{o}'), \boldsymbol{\theta}') \tag{C.43}$$

Substituting this back and using the definition of inner product:

$$(\Psi^* \mu)(A) = \int_{S \times \boldsymbol{B} \times \boldsymbol{\Theta}} (\Psi \mathbf{1}_A)(s,\boldsymbol{b},\boldsymbol{\theta}) \, d\mu(s,\boldsymbol{b},\boldsymbol{\theta}) \tag{C.44}$$

$$= \int_{S \times \boldsymbol{B} \times \boldsymbol{\Theta}} \sum_{a^i} \pi^i(a^i|b^i;\theta^i) \sum_{a^{\neg i}} \pi^{\neg i}(a^{\neg i}|b^{\neg i};\theta^{\neg i}) \sum_{s'} T(s'|s,\boldsymbol{a}) \tag{C.45}$$

$$\sum_{\boldsymbol{o}'} \boldsymbol{O}(\boldsymbol{o}'|s') \sum_{\boldsymbol{\theta}'} \boldsymbol{U}(\boldsymbol{\theta}'|\boldsymbol{\theta},\boldsymbol{\tau}) \mathbf{1}_A(s', \text{update}(\boldsymbol{b},\boldsymbol{a},\boldsymbol{o}'), \boldsymbol{\theta}') \, d\mu(s,\boldsymbol{b},\boldsymbol{\theta}) \tag{C.46}$$

□

C.1.3 Contraction and Propagation

The properties established above lead to two crucial results: the contraction property of the discounted transition operator and the propagation of distributions through the adjoint.

Lemma 4 (Contraction Mapping Property)**.** *For any discount factor $\gamma \in [0,1)$, the operator $\gamma\Psi$ is a contraction mapping on the Banach space of bounded functions on $S \times \boldsymbol{B} \times \boldsymbol{\Theta}$ equipped*

with the supremum norm. Specifically, for any two functions $f, g \in \mathscr{B}(S \times \boldsymbol{B} \times \boldsymbol{\Theta})$:

$$\|\gamma\Psi f - \gamma\Psi g\|_\infty \leq \gamma\|f - g\|_\infty \tag{C.47}$$

Proof. Using the linearity of Ψ and its bound preservation property:

$$\|\gamma\Psi f - \gamma\Psi g\|_\infty = \gamma\|\Psi(f-g)\|_\infty \tag{C.48}$$

$$\leq \gamma\|f - g\|_\infty \tag{C.49}$$

Since $\gamma < 1$, $\gamma\Psi$ is a contraction mapping with contraction factor γ. □

This contraction property leads directly to the invertibility of $I - \gamma\Psi$, a result that is fundamental for characterizing value functions in discounted settings.

Lemma 5 (Invertibility of $I - \gamma\Psi$)**.** *For any discount factor $\gamma \in [0, 1)$, the operator $I - \gamma\Psi$ is invertible, and its inverse can be represented as a Neumann series:*

$$(I - \gamma\Psi)^{-1} = \sum_{t=0}^{\infty} \gamma^t \Psi^t \tag{C.50}$$

which converges absolutely in the operator norm.

Proof. By the Banach fixed-point theorem, if T is a contraction mapping on a Banach space, then $I - T$ is invertible, with inverse given by the Neumann series $\sum_{k=0}^{\infty} T^k$. Since $\gamma\Psi$ is a contraction mapping, we have:

$$(I - \gamma\Psi)^{-1} = \sum_{t=0}^{\infty} (\gamma\Psi)^t = \sum_{t=0}^{\infty} \gamma^t \Psi^t \tag{C.51}$$

The absolute convergence follows from:

$$\left\| \sum_{t=0}^{\infty} \gamma^t \Psi^t \right\|_\infty \leq \sum_{t=0}^{\infty} \gamma^t \|\Psi^t\|_\infty \leq \sum_{t=0}^{\infty} \gamma^t = \frac{1}{1-\gamma} < \infty \tag{C.52}$$

where we use the bound preservation property to establish that $\|\Psi^t\|_\infty \leq 1$ for all t. □

Finally, we establish how distributions evolve in the system through the adjoint operator, formalizing the propagation of probability measures.

Lemma 6 (Distribution Propagation)**.** *If μ_t represents the distribution of the joint process $(s_t, \boldsymbol{b}_t, \boldsymbol{\theta}_t)$ at time t, then the distribution at time $t+1$ is given by $\mu_{t+1} = \Psi^* \mu_t$.*

Proof. For any measurable function $f \in \mathscr{B}(S \times \boldsymbol{B} \times \boldsymbol{\Theta})$:

$$\langle f, \mu_{t+1} \rangle = \mathbb{E}[f(s_{t+1}, \boldsymbol{b}_{t+1}, \boldsymbol{\theta}_{t+1})] \tag{C.53}$$

$$= \mathbb{E}[\mathbb{E}[f(s_{t+1}, \boldsymbol{b}_{t+1}, \boldsymbol{\theta}_{t+1}) \mid s_t, \boldsymbol{b}_t, \boldsymbol{\theta}_t]] \tag{C.54}$$

$$= \mathbb{E}[(\Psi f)(s_t, \boldsymbol{b}_t, \boldsymbol{\theta}_t)] \tag{C.55}$$

$$= \langle \Psi f, \mu_t \rangle = \langle f, \Psi^* \mu_t \rangle \tag{C.56}$$

Since this holds for any measurable function f, we have $\mu_{t+1} = \Psi^* \mu_t$ by the uniqueness of the representing measure. □

The concepts introduced in this section—the transition operator, its adjoint, and their properties—provide the mathematical foundation for analyzing discounted returns in Partially Observable Active Markov Games. The transition operator Ψ captures how function expectations evolve forward in time, while its adjoint Ψ^* describes how probability distributions propagate forward in time. This duality is fundamental to establishing the connection between value-based and distribution-based perspectives in reinforcement learning.

In the next section, we will use these tools to formulate Bellman equations and derive policy gradients for maximizing discounted returns in this complex multi-agent setting.

C.2 Bellman Equations and Value Functions

In this section, we establish the fundamental recursive relationships that characterize the discounted value functions in Partially Observable Active Markov Games. These relationships form the basis for our policy gradient derivation and practical algorithms for optimizing agent behavior. We begin by confirming the well-definedness of value functions, then establish the Bellman equations, and finally derive the gradient expressions necessary for policy optimization.

C.2.1 Well-Definedness of Value Functions

Before deriving recursive relationships, we first establish that the value functions are well-defined mathematical objects under our assumptions of bounded rewards and discount factors less than 1.

Lemma 7 (Well-Defined Value Functions)**.** *Under the assumptions of bounded rewards and $\gamma < 1$, the discounted value functions $v^i_{\theta^i}(s, \boldsymbol{b}, \boldsymbol{\theta})$ and action-value functions $q^i_{\theta^i}(s, \boldsymbol{b}, \boldsymbol{\theta}, \boldsymbol{a})$ are well-defined and finite for all states, belief states, policy parameters, and actions.*

Proof. Assuming rewards are bounded such that $|R^i(s, \boldsymbol{a})| \leq R_{\max}$ for all $s \in S$ and $\boldsymbol{a} \in A$, we

have:

$$|v^i_{\theta^i}(s,\boldsymbol{b},\boldsymbol{\theta})| = \left|\mathbb{E}\left[\sum_{t=0}^{\infty}\gamma^t R^i(s_t,\boldsymbol{a}_t)\middle| s_0=s,\boldsymbol{b}_0=\boldsymbol{b},\boldsymbol{\theta}_0=\boldsymbol{\theta},\pi\right]\right| \tag{C.57}$$

$$\leq \mathbb{E}\left[\sum_{t=0}^{\infty}\gamma^t |R^i(s_t,\boldsymbol{a}_t)|\middle| s_0=s,\boldsymbol{b}_0=\boldsymbol{b},\boldsymbol{\theta}_0=\boldsymbol{\theta},\pi\right] \tag{C.58}$$

$$\leq R_{\max}\sum_{t=0}^{\infty}\gamma^t = \frac{R_{\max}}{1-\gamma} < \infty \tag{C.59}$$

Similarly, for the action-value function:

$$|q^i_{\theta^i}(s,\boldsymbol{b},\boldsymbol{\theta},\boldsymbol{a})| = \left|R^i(s,\boldsymbol{a}) + \gamma\mathbb{E}[v^i_{\theta^i}(s',\boldsymbol{b}',\boldsymbol{\theta}')]\right| \tag{C.60}$$

$$\leq |R^i(s,\boldsymbol{a})| + \gamma\mathbb{E}[|v^i_{\theta^i}(s',\boldsymbol{b}',\boldsymbol{\theta}')|] \tag{C.61}$$

$$\leq R_{\max} + \gamma\frac{R_{\max}}{1-\gamma} = \frac{R_{\max}}{1-\gamma} < \infty \tag{C.62}$$

Thus, both value functions are well-defined and bounded. □

This lemma ensures that our subsequent derivations involving value functions are mathematically sound. The boundedness property is particularly important when we consider expectations and derivatives of these functions.

C.2.2 Bellman Equations for Discounted Value Functions

Next, we establish the recursive relationships between value functions, which are fundamental to dynamic programming approaches.

Lemma 8 (Bellman Equation for Discounted Value Functions)**.** *In a Partially Observable Active Markov Game, the discounted value function $v^i_{\theta^i}$ satisfies the Bellman equation:*

$$v^i_{\theta^i}(s,\boldsymbol{b},\boldsymbol{\theta}) = \sum_{a^i}\pi^i(a^i|b^i;\theta^i)\sum_{a^{\neg i}}\pi^{\neg i}(a^{\neg i}|b^{\neg i};\theta^{\neg i})q^i_{\theta^i}(s,\boldsymbol{b},\boldsymbol{\theta},\boldsymbol{a}) \tag{C.63}$$

where the action-value function $q^i_{\theta^i}$ is defined as:

$$q^i_{\theta^i}(s,\boldsymbol{b},\boldsymbol{\theta},\boldsymbol{a}) = R^i(s,\boldsymbol{a}) + \gamma\sum_{s'}T(s'|s,\boldsymbol{a})\sum_{\boldsymbol{o}'}\boldsymbol{O}(\boldsymbol{o}'|s') \sum_{\boldsymbol{\theta}'}\boldsymbol{U}(\boldsymbol{\theta}'|\boldsymbol{\theta},\boldsymbol{\tau})v^i_{\theta^i}(s',update(\boldsymbol{b},\boldsymbol{a},\boldsymbol{o}'),\boldsymbol{\theta}') \tag{C.64}$$

Proof. By definition, the discounted value function is:

$$v^i_{\theta^i}(s,\boldsymbol{b},\boldsymbol{\theta}) = \mathbb{E}\left[\sum_{t=0}^{\infty}\gamma^t R^i(s_t,\boldsymbol{a}_t)\middle| s_0=s,\boldsymbol{b}_0=\boldsymbol{b},\boldsymbol{\theta}_0=\boldsymbol{\theta},\pi\right] \tag{C.65}$$

We can decompose this into the immediate reward and the future discounted returns:

$$v^i_{\theta^i}(s, \boldsymbol{b}, \boldsymbol{\theta}) = \mathbb{E}\left[R^i(s_0, \boldsymbol{a}_0) + \gamma \sum_{t=1}^{\infty} \gamma^{t-1} R^i(s_t, \boldsymbol{a}_t) \middle| s_0 = s, \boldsymbol{b}_0 = \boldsymbol{b}, \boldsymbol{\theta}_0 = \boldsymbol{\theta}, \pi\right] \tag{C.66}$$

$$= \mathbb{E}\left[R^i(s_0, \boldsymbol{a}_0) \middle| s_0 = s, \boldsymbol{b}_0 = \boldsymbol{b}, \boldsymbol{\theta}_0 = \boldsymbol{\theta}, \pi\right]$$

$$+ \gamma \mathbb{E}\left[\sum_{t=1}^{\infty} \gamma^{t-1} R^i(s_t, \boldsymbol{a}_t) \middle| s_0 = s, \boldsymbol{b}_0 = \boldsymbol{b}, \boldsymbol{\theta}_0 = \boldsymbol{\theta}, \pi\right] \tag{C.67}$$

For the first term, we have:

$$\mathbb{E}\left[R^i(s_0, \boldsymbol{a}_0) \middle| s_0 = s, \boldsymbol{b}_0 = \boldsymbol{b}, \boldsymbol{\theta}_0 = \boldsymbol{\theta}, \pi\right] = \sum_{a^i} \pi^i(a^i | b^i; \theta^i) \sum_{a^{\neg i}} \pi^{\neg i}(a^{\neg i} | b^{\neg i}; \theta^{\neg i}) R^i(s, \boldsymbol{a}) \tag{C.68}$$

For the second term, by the law of total expectation:

$$\mathbb{E}\left[\sum_{t=1}^{\infty} \gamma^{t-1} R^i(s_t, \boldsymbol{a}_t) \middle| s_0 = s, \boldsymbol{b}_0 = \boldsymbol{b}, \boldsymbol{\theta}_0 = \boldsymbol{\theta}, \pi\right] \tag{C.69}$$

$$= \sum_{a^i} \pi^i(a^i | b^i; \theta^i) \sum_{a^{\neg i}} \pi^{\neg i}(a^{\neg i} | b^{\neg i}; \theta^{\neg i}) \sum_{s'} T(s' | s, \boldsymbol{a})$$

$$\sum_{\boldsymbol{o}'} \boldsymbol{O}(\boldsymbol{o}' | s') \sum_{\boldsymbol{\theta}'} \boldsymbol{U}(\boldsymbol{\theta}' | \boldsymbol{\theta}, \boldsymbol{\tau}) v^i_{\theta^i}(s', \text{update}(\boldsymbol{b}, \boldsymbol{a}, \boldsymbol{o}'), \boldsymbol{\theta}') \tag{C.70}$$

Combining these two terms and factoring out common terms:

$$v^i_{\theta^i}(s, \boldsymbol{b}, \boldsymbol{\theta}) = \sum_{a^i} \pi^i(a^i | b^i; \theta^i) \sum_{a^{\neg i}} \pi^{\neg i}(a^{\neg i} | b^{\neg i}; \theta^{\neg i}) \Bigg[R^i(s, \boldsymbol{a})$$

$$+ \gamma \sum_{s'} T(s' | s, \boldsymbol{a}) \sum_{\boldsymbol{o}'} \boldsymbol{O}(\boldsymbol{o}' | s')$$

$$\sum_{\boldsymbol{\theta}'} \boldsymbol{U}(\boldsymbol{\theta}' | \boldsymbol{\theta}, \boldsymbol{\tau}) v^i_{\theta^i}(s', \text{update}(\boldsymbol{b}, \boldsymbol{a}, \boldsymbol{o}'), \boldsymbol{\theta}') \Bigg] \tag{C.71}$$

$$= \sum_{a^i} \pi^i(a^i | b^i; \theta^i) \sum_{a^{\neg i}} \pi^{\neg i}(a^{\neg i} | b^{\neg i}; \theta^{\neg i}) q^i_{\theta^i}(s, \boldsymbol{b}, \boldsymbol{\theta}, \boldsymbol{a}) \tag{C.72}$$

where we define the action-value function $q^i_{\theta^i}$ as specified. □

The Bellman equation provides a recursive characterization of the value function that forms the basis for dynamic programming algorithms. This equation captures how current actions influence both immediate rewards and future value through their effects on the environment state, belief states, and policy parameters of other agents. The key distinction from standard Bellman equations is the inclusion of belief updates and policy parameter dynamics, which reflects the complex dependencies in partially observable multi-agent settings.

C.2.3 Policy Gradient with Respect to Value Function

Having established the recursive relationship for value functions, we now derive how the value function gradient depends on policy parameters, which is essential for policy optimization.

Lemma 9 (Policy Gradient with Respect to Value Function)**.** *The gradient of the value function with respect to policy parameters θ^i can be expressed as:*

$$\nabla_{\theta^i} v^i_{\theta^i}(s, \boldsymbol{b}, \boldsymbol{\theta}) = (I - \gamma\Psi)^{-1} \left[\sum_{a^i} \nabla_{\theta^i} \pi^i(a^i|b^i;\theta^i) \sum_{a^{\neg i}} \pi^{\neg i}(a^{\neg i}|b^{\neg i};\theta^{\neg i}) q^i_{\theta^i}(s, \boldsymbol{b}, \boldsymbol{\theta}, \boldsymbol{a}) \right] \tag{C.73}$$

where Ψ is the transition operator defined in the previous section.

Proof. Taking the gradient of the Bellman equation with respect to θ^i:

$$\nabla_{\theta^i} v^i_{\theta^i}(s, \boldsymbol{b}, \boldsymbol{\theta}) = \nabla_{\theta^i} \left[\sum_{a^i} \pi^i(a^i|b^i;\theta^i) \sum_{a^{\neg i}} \pi^{\neg i}(a^{\neg i}|b^{\neg i};\theta^{\neg i}) q^i_{\theta^i}(s, \boldsymbol{b}, \boldsymbol{\theta}, \boldsymbol{a}) \right] \tag{C.74}$$

Applying the product rule:

$$\begin{aligned} \nabla_{\theta^i} v^i_{\theta^i}(s, \boldsymbol{b}, \boldsymbol{\theta}) = &\sum_{a^i} \nabla_{\theta^i} \pi^i(a^i|b^i;\theta^i) \sum_{a^{\neg i}} \pi^{\neg i}(a^{\neg i}|b^{\neg i};\theta^{\neg i}) q^i_{\theta^i}(s, \boldsymbol{b}, \boldsymbol{\theta}, \boldsymbol{a}) \\ &+ \sum_{a^i} \pi^i(a^i|b^i;\theta^i) \sum_{a^{\neg i}} \pi^{\neg i}(a^{\neg i}|b^{\neg i};\theta^{\neg i}) \nabla_{\theta^i} q^i_{\theta^i}(s, \boldsymbol{b}, \boldsymbol{\theta}, \boldsymbol{a}) \end{aligned} \tag{C.75}$$

For the gradient of the action-value function:

$$\nabla_{\theta^i} q^i_{\theta^i}(s, \boldsymbol{b}, \boldsymbol{\theta}, \boldsymbol{a}) = \nabla_{\theta^i} \left[R^i(s, \boldsymbol{a}) + \gamma \sum_{s'} T(s'|s, \boldsymbol{a}) \sum_{\boldsymbol{o}'} \boldsymbol{O}(\boldsymbol{o}'|s') \sum_{\boldsymbol{\theta}'} \boldsymbol{U}(\boldsymbol{\theta}'|\boldsymbol{\theta}, \boldsymbol{\tau}) v^i_{\theta^i}(s', \text{update}(\boldsymbol{b}, \boldsymbol{a}, \boldsymbol{o}'), \boldsymbol{\theta}') \right] \tag{C.76}$$

Since $R^i(s, \boldsymbol{a})$, $T(s'|s, \boldsymbol{a})$, $\boldsymbol{O}(\boldsymbol{o}'|s')$, and $\boldsymbol{U}(\boldsymbol{\theta}'|\boldsymbol{\theta}, \boldsymbol{\tau})$ do not depend directly on θ^i:

$$\begin{aligned} \nabla_{\theta^i} q^i_{\theta^i}(s, \boldsymbol{b}, \boldsymbol{\theta}, \boldsymbol{a}) = &\gamma \sum_{s'} T(s'|s, \boldsymbol{a}) \sum_{\boldsymbol{o}'} \boldsymbol{O}(\boldsymbol{o}'|s') \\ &\sum_{\boldsymbol{\theta}'} \boldsymbol{U}(\boldsymbol{\theta}'|\boldsymbol{\theta}, \boldsymbol{\tau}) \nabla_{\theta^i} v^i_{\theta^i}(s', \text{update}(\boldsymbol{b}, \boldsymbol{a}, \boldsymbol{o}'), \boldsymbol{\theta}') \end{aligned} \tag{C.77}$$

Using the transition operator Ψ, we can rewrite:

$$\begin{aligned} \nabla_{\theta^i} v^i_{\theta^i}(s, \boldsymbol{b}, \boldsymbol{\theta}) = &\sum_{a^i} \nabla_{\theta^i} \pi^i(a^i|b^i;\theta^i) \sum_{a^{\neg i}} \pi^{\neg i}(a^{\neg i}|b^{\neg i};\theta^{\neg i}) q^i_{\theta^i}(s, \boldsymbol{b}, \boldsymbol{\theta}, \boldsymbol{a}) \\ &+ \gamma\Psi[\nabla_{\theta^i} v^i_{\theta^i}](s, \boldsymbol{b}, \boldsymbol{\theta}) \end{aligned} \tag{C.78}$$

Rearranging:

$$(I-\gamma\Psi)[\nabla_{\theta^i} v^i_{\theta^i}](s,\boldsymbol{b},\boldsymbol{\theta}) = \sum_{a^i} \nabla_{\theta^i}\pi^i(a^i|b^i;\theta^i) \sum_{a^{\neg i}} \pi^{\neg i}(a^{\neg i}|b^{\neg i};\theta^{\neg i}) q^i_{\theta^i}(s,\boldsymbol{b},\boldsymbol{\theta},\boldsymbol{a}) \tag{C.79}$$

By the lemma on the invertibility of $(I-\gamma\Psi)$, we have:

$$\nabla_{\theta^i} v^i_{\theta^i}(s,\boldsymbol{b},\boldsymbol{\theta}) = (I-\gamma\Psi)^{-1}\left[\sum_{a^i} \nabla_{\theta^i}\pi^i(a^i|b^i;\theta^i) \sum_{a^{\neg i}} \pi^{\neg i}(a^{\neg i}|b^{\neg i};\theta^{\neg i}) q^i_{\theta^i}(s,\boldsymbol{b},\boldsymbol{\theta},\boldsymbol{a})\right] \tag{C.80}$$

$$= \sum_{t=0}^{\infty} \gamma^t \Psi^t \left[\sum_{a^i} \nabla_{\theta^i}\pi^i(a^i|b^i;\theta^i) \sum_{a^{\neg i}} \pi^{\neg i}(a^{\neg i}|b^{\neg i};\theta^{\neg i}) q^i_{\theta^i}(s,\boldsymbol{b},\boldsymbol{\theta},\boldsymbol{a})\right] \tag{C.81}$$

□

This lemma expresses the value function gradient in terms of an infinite sum of expected Q-values weighted by policy gradients and propagated through the transition operator. The operator $(I-\gamma\Psi)^{-1}$ captures how immediate changes in policy parameters propagate through future timesteps. This form of the gradient is central to our discounted policy gradient theorem, which we will develop in the next section. The inverse operator can be represented as a Neumann series $\sum_{t=0}^{\infty}\gamma^t\Psi^t$, which has a natural interpretation as the accumulated effect of policy changes over an infinite horizon, weighted by the discount factor.

C.3 Discounted Visitation Measure

The infinite sum formulation derived in the previous sections, while mathematically sound, is not directly amenable to practical computation. In this section, we reformulate these expressions in terms of a probability distribution, which allows for more efficient estimation through sampling-based approaches. This reformulation is central to our policy gradient theorem for discounted returns.

C.3.1 Definition and Properties

Definition 10 (Discounted Visitation Measure)**.** *For a Markov process governed by a transition operator Ψ and its adjoint Ψ^*, with an initial distribution μ_0 over the joint state-belief-policy*

space, the discounted visitation measure $d^{\pi}_{\mu_0}$ is defined as:

$$d^{\pi}_{\mu_0} := (1-\gamma)\sum_{t=0}^{\infty}\gamma^t\mu_t \tag{C.82}$$

where μ_t is the distribution at time t when starting from μ_0, and $\gamma \in [0,1)$ is the discount factor.

This measure represents the expected discounted frequency with which different states, belief states, and policy parameters are visited when following policy π starting from the initial distribution μ_0. The normalization factor $(1-\gamma)$ ensures that $d^{\pi}_{\mu_0}$ is a proper probability distribution, summing to 1 across the entire state-belief-policy space. The discounted visitation measure plays a central role in reinforcement learning theory, particularly in policy gradient methods. It provides a natural weighting of states according to their importance for the discounted objective, as states that are visited more frequently or earlier in trajectories receive higher weight. This concept generalizes the stationary distribution used in average-reward settings to the discounted return framework.

Lemma 10 (Evolution of Discounted Visitation)**.** *The discounted visitation measure $d^{\pi}_{\mu_0}$ can be expressed in terms of the adjoint operator as:*

$$d^{\pi}_{\mu_0} = (1-\gamma)\sum_{t=0}^{\infty}\gamma^t(\Psi^*)^t\mu_0 = (1-\gamma)(I-\gamma\Psi^*)^{-1}\mu_0 \tag{C.83}$$

where μ_0 is the initial distribution.

Proof. By the propagation lemma established earlier, $\mu_t = (\Psi^*)^t\mu_0$ represents the distribution at time t when starting from μ_0. The discounted visitation measure weights these distributions by γ^t and normalizes by $(1-\gamma)$ to ensure it's a proper probability measure:

$$d^{\pi}_{\mu_0} = (1-\gamma)\sum_{t=0}^{\infty}\gamma^t\mu_t \tag{C.84}$$

$$= (1-\gamma)\sum_{t=0}^{\infty}\gamma^t(\Psi^*)^t\mu_0 \tag{C.85}$$

The second equality follows from the Neumann series expansion:

$$(I-\gamma\Psi^*)^{-1} = \sum_{t=0}^{\infty}(\gamma\Psi^*)^t = \sum_{t=0}^{\infty}\gamma^t(\Psi^*)^t \tag{C.86}$$

which converges for $\gamma < 1$ since Ψ^* is a non-expansion operator in the total variation norm. □

This establishes the essential connection between the adjoint operator and the discounted visitation measure, which is central to the policy gradient theorem. The measure $d^{\pi}_{\mu_0}$ represents the normalized expected discounted time spent in each state-belief-policy configuration,

weighted according to the discount factor γ. This connection allows us to transform expressions involving $(I-\gamma\Psi)^{-1}$ in the value function to expectations with respect to $d^{\pi}_{\mu_0}$, which is key for deriving practical policy gradient algorithms.In practical implementations, the discounted visitation measure is rarely computed explicitly due to the prohibitive computational cost. Instead, sampling-based approaches are used to estimate expectations with respect to this measure. The policy gradient theorem leverages this measure to derive update rules that can be efficiently implemented using trajectory samples collected from the environment.

C.3.2 Existence and Uniqueness

To provide a rigorous mathematical foundation for our policy gradient theorem, we establish the existence and uniqueness of the discounted visitation measure.

Lemma 11 (Existence of the Discounted Visitation Measure)**.** *Let $\mathscr{M}(S\times\boldsymbol{B}\times\boldsymbol{\Theta})$ be the space of finite measures on the joint state-belief-policy space, equipped with the total variation norm. For any initial measure $\mu_0\in\mathscr{M}(S\times\boldsymbol{B}\times\boldsymbol{\Theta})$ and discount factor $\gamma\in[0,1)$, the discounted state-visitation measure*

$$d^{\pi}_{\mu_0}=(1-\gamma)\sum_{t=0}^{\infty}\gamma^t(\Psi^*)^t\mu_0 \tag{C.87}$$

exists and is a well-defined measure in $\mathscr{M}(S\times\boldsymbol{B}\times\boldsymbol{\Theta})$.

Proof. To establish the existence of $d^{\pi}_{\mu_0}$, we need to prove that the infinite sum $(1-\gamma)\sum_{t=0}^{\infty}\gamma^t(\Psi^*)^t\mu_0$ converges to a well-defined measure. First, let's recall that $\mathscr{M}(S\times\boldsymbol{B}\times\boldsymbol{\Theta})$ equipped with the total variation norm $\|\mu\|_{TV}=\sup_{|f|\le 1}|\int f d\mu|$ is a Banach space. Now, we need to show that the series $\sum_{t=0}^{\infty}\gamma^t(\Psi^*)^t\mu_0$ converges in this space. For any $t\ge 0$, $(\Psi^*)^t\mu_0$ represents the measure over the joint state-belief-policy space after t transitions, starting from the initial measure μ_0. Since Ψ^* is a Markov operator (the adjoint of a Markov transition kernel), it preserves the total mass of a measure: if μ_0 is a probability measure (i.e., $\mu_0(S\times\boldsymbol{B}\times\boldsymbol{\Theta})=1$), then so is $(\Psi^*)^t\mu_0$ for all $t\ge 0$. Therefore, $\|(\Psi^*)^t\mu_0\|_{TV}=\mu_0(S\times\boldsymbol{B}\times\boldsymbol{\Theta})$ for all $t\ge 0$. For a general finite measure μ_0 with total mass $M=\mu_0(S\times\boldsymbol{B}\times\boldsymbol{\Theta})$, we have $\|(\Psi^*)^t\mu_0\|_{TV}=M$ for all $t\ge 0$. Now, consider the partial sum:

$$S_n=(1-\gamma)\sum_{t=0}^{n}\gamma^t(\Psi^*)^t\mu_0 \tag{C.88}$$

For $n < m$, the difference between two partial sums is:

$$\|S_m - S_n\|_{TV} = \left\| (1-\gamma) \sum_{t=n+1}^{m} \gamma^t (\Psi^*)^t \mu_0 \right\|_{TV} \tag{C.89}$$

$$\leq (1-\gamma) \sum_{t=n+1}^{m} \gamma^t \|(\Psi^*)^t \mu_0\|_{TV} \tag{C.90}$$

$$= (1-\gamma) M \sum_{t=n+1}^{m} \gamma^t \tag{C.91}$$

$$= (1-\gamma) M (\gamma^{n+1} + \gamma^{n+2} + \ldots + \gamma^m) \tag{C.92}$$

$$= (1-\gamma) M \gamma^{n+1} \frac{1-\gamma^{m-n}}{1-\gamma} \tag{C.93}$$

$$= M \gamma^{n+1} (1-\gamma^{m-n}) \tag{C.94}$$

As $n \to \infty$, $\gamma^{n+1} \to 0$ (since $\gamma < 1$), which means that the sequence of partial sums $\{S_n\}$ is Cauchy in the Banach space $\mathscr{M}(S \times \boldsymbol{B} \times \boldsymbol{\Theta})$. By the completeness of this space, the sequence converges to a measure which we denote as $d^{\pi}_{\mu_0}$. Moreover, the total mass of $d^{\pi}_{\mu_0}$ is:

$$d^{\pi}_{\mu_0}(S \times \boldsymbol{B} \times \boldsymbol{\Theta}) = (1-\gamma) \sum_{t=0}^{\infty} \gamma^t (\Psi^*)^t \mu_0 (S \times \boldsymbol{B} \times \boldsymbol{\Theta}) \tag{C.95}$$

$$= (1-\gamma) M \sum_{t=0}^{\infty} \gamma^t \tag{C.96}$$

$$= (1-\gamma) M \frac{1}{1-\gamma} \tag{C.97}$$

$$= M \tag{C.98}$$

Thus, $d^{\pi}_{\mu_0}$ is a finite measure with the same total mass as μ_0. □

Having established existence, we now turn to the uniqueness of the discounted visitation measure.

Lemma 12 (Uniqueness of the Discounted Visitation Measure)**.** *The discounted state-visitation measure $d^{\pi}_{\mu_0}$ is the unique solution to the functional equation:*

$$d^{\pi}_{\mu_0} = (1-\gamma)\mu_0 + \gamma \Psi^* d^{\pi}_{\mu_0} \tag{C.99}$$

in the space of finite measures $\mathscr{M}(S \times \boldsymbol{B} \times \boldsymbol{\Theta})$.

Proof. First, we verify that $d^{\pi}_{\mu_0} = (1-\gamma)\sum_{t=0}^{\infty} \gamma^t (\Psi^*)^t \mu_0$ satisfies the functional equation:

$$(1-\gamma)\mu_0 + \gamma\Psi^* d^\pi_{\mu_0} = (1-\gamma)\mu_0 + \gamma\Psi^* \left((1-\gamma) \sum_{t=0}^{\infty} \gamma^t (\Psi^*)^t \mu_0 \right) \tag{C.100}$$

$$= (1-\gamma)\mu_0 + \gamma(1-\gamma) \sum_{t=0}^{\infty} \gamma^t (\Psi^*)^{t+1} \mu_0 \tag{C.101}$$

$$= (1-\gamma)\mu_0 + (1-\gamma) \sum_{t=1}^{\infty} \gamma^t (\Psi^*)^t \mu_0 \tag{C.102}$$

$$= (1-\gamma) \left(\mu_0 + \sum_{t=1}^{\infty} \gamma^t (\Psi^*)^t \mu_0 \right) \tag{C.103}$$

$$= (1-\gamma) \sum_{t=0}^{\infty} \gamma^t (\Psi^*)^t \mu_0 \tag{C.104}$$

$$= d^\pi_{\mu_0} \tag{C.105}$$

To prove uniqueness, suppose there exists another measure $\tilde{d} \in \mathscr{M}(S \times \boldsymbol{B} \times \boldsymbol{\Theta})$ that satisfies the functional equation:

$$\tilde{d} = (1-\gamma)\mu_0 + \gamma\Psi^* \tilde{d} \tag{C.106}$$

Let's consider the operator $T : \mathscr{M}(S \times \boldsymbol{B} \times \boldsymbol{\Theta}) \to \mathscr{M}(S \times \boldsymbol{B} \times \boldsymbol{\Theta})$ defined as:

$$T(\mu) = (1-\gamma)\mu_0 + \gamma\Psi^* \mu \tag{C.107}$$

For any two measures $\mu, \nu \in \mathscr{M}(S \times \boldsymbol{B} \times \boldsymbol{\Theta})$:

$$\|T(\mu) - T(\nu)\|_{TV} = \|\gamma\Psi^*\mu - \gamma\Psi^*\nu\|_{TV} \tag{C.108}$$

$$= \gamma \|\Psi^*(\mu - \nu)\|_{TV} \tag{C.109}$$

$$\leq \gamma \|\mu - \nu\|_{TV} \tag{C.110}$$

Since $\gamma < 1$, T is a contraction mapping on the Banach space $\mathscr{M}(S \times \boldsymbol{B} \times \boldsymbol{\Theta})$ with contraction factor γ. By the Banach fixed-point theorem, T has a unique fixed point. We've already shown that $d^\pi_{\mu_0}$ is a fixed point of T, and by assumption, $\tilde{d}$ is also a fixed point. Therefore, $d^\pi_{\mu_0} = \tilde{d}$, proving the uniqueness of the discounted state-visitation measure. □

C.3.3 Connection to Value Function

Having established the theoretical properties of the discounted visitation measure, we now connect it to the value function, which will provide important insights later on.

Lemma 13 (Value Function as Inner Product)**.** *For any discount factor $\gamma \in [0,1)$ and initial*

state s_0, belief state $\boldsymbol{b}_0$, and policy parameter $\boldsymbol{\theta}_0$, the value function can be expressed as:

$$v^i_{\theta^i}(s_0, \boldsymbol{b}_0, \boldsymbol{\theta}_0) = \frac{1}{1-\gamma} \int_{S \times \boldsymbol{B} \times \boldsymbol{\Theta}} r^i(s, \boldsymbol{b}, \boldsymbol{\theta}) d^{\pi}_{\mu_0}(s, \boldsymbol{b}, \boldsymbol{\theta}) \tag{C.111}$$

where $\mu_0 = \delta_{(s_0, \boldsymbol{b}_0, \boldsymbol{\theta}_0)}$ is the Dirac measure concentrated at $(s_0, \boldsymbol{b}_0, \boldsymbol{\theta}_0)$, and r^i is the expected immediate reward function:

$$r^i(s, \boldsymbol{b}, \boldsymbol{\theta}) = \sum_{a^i} \pi^i(a^i | b^i; \theta^i) \sum_{a^{\neg i}} \pi^{\neg i}(a^{\neg i} | b^{\neg i}; \theta^{\neg i}) R^i(s, \boldsymbol{a}) \tag{C.112}$$

Proof. Let $\mu_0 = \delta_{(s_0, \boldsymbol{b}_0, \boldsymbol{\theta}_0)}$ be the Dirac measure concentrated at the point $(s_0, \boldsymbol{b}_0, \boldsymbol{\theta}_0)$. By definition of the value function:

$$v^i_{\theta^i}(s_0, \boldsymbol{b}_0, \boldsymbol{\theta}_0) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R^i(s_t, \boldsymbol{a}_t) \middle| s_0, \boldsymbol{b}_0, \boldsymbol{\theta}_0, \pi \right] \tag{C.113}$$

$$= \sum_{t=0}^{\infty} \gamma^t \mathbb{E}\left[R^i(s_t, \boldsymbol{a}_t) \middle| s_0, \boldsymbol{b}_0, \boldsymbol{\theta}_0, \pi \right] \tag{C.114}$$

Define the expected immediate reward function $r^i : S \times \boldsymbol{B} \times \boldsymbol{\Theta} \to \mathbb{R}$ as:

$$r^i(s, \boldsymbol{b}, \boldsymbol{\theta}) = \sum_{a^i} \pi^i(a^i | b^i; \theta^i) \sum_{a^{\neg i}} \pi^{\neg i}(a^{\neg i} | b^{\neg i}; \theta^{\neg i}) R^i(s, \boldsymbol{a}) \tag{C.115}$$

Then:

$$v^i_{\theta^i}(s_0, \boldsymbol{b}_0, \boldsymbol{\theta}_0) = \sum_{t=0}^{\infty} \gamma^t \mathbb{E}\left[r^i(s_t, \boldsymbol{b}_t, \boldsymbol{\theta}_t) \middle| s_0, \boldsymbol{b}_0, \boldsymbol{\theta}_0, \pi \right] \tag{C.116}$$

$$= \sum_{t=0}^{\infty} \gamma^t \int_{S \times \boldsymbol{B} \times \boldsymbol{\Theta}} r^i(s, \boldsymbol{b}, \boldsymbol{\theta}) d\mu_t(s, \boldsymbol{b}, \boldsymbol{\theta}) \tag{C.117}$$

where μ_t is the distribution over the joint state-belief-policy space at time t, starting from μ_0. From previous lemmas, we know that $\mu_t = (\Psi^*)^t \mu_0$. Using the duality relationship between functions and measures:

$$\int_{S \times \boldsymbol{B} \times \boldsymbol{\Theta}} r^i(s, \boldsymbol{b}, \boldsymbol{\theta}) d\mu_t(s, \boldsymbol{b}, \boldsymbol{\theta}) = \int_{S \times \boldsymbol{B} \times \boldsymbol{\Theta}} r^i(s, \boldsymbol{b}, \boldsymbol{\theta}) d((\Psi^*)^t \mu_0)(s, \boldsymbol{b}, \boldsymbol{\theta}) \tag{C.118}$$

$$= \int_{S \times \boldsymbol{B} \times \boldsymbol{\Theta}} (\Psi^t r^i)(s, \boldsymbol{b}, \boldsymbol{\theta}) d\mu_0(s, \boldsymbol{b}, \boldsymbol{\theta}) \tag{C.119}$$

Therefore:

$$v^i_{\theta^i}(s_0, \boldsymbol{b}_0, \boldsymbol{\theta}_0) = \sum_{t=0}^{\infty} \gamma^t \int_{S\times \boldsymbol{B}\times\boldsymbol{\Theta}} (\Psi^t r^i)(s, \boldsymbol{b}, \boldsymbol{\theta}) d\mu_0(s, \boldsymbol{b}, \boldsymbol{\theta}) \tag{C.120}$$

$$= \int_{S\times \boldsymbol{B}\times\boldsymbol{\Theta}} \left(\sum_{t=0}^{\infty} \gamma^t (\Psi^t r^i)(s, \boldsymbol{b}, \boldsymbol{\theta}) \right) d\mu_0(s, \boldsymbol{b}, \boldsymbol{\theta}) \tag{C.121}$$

$$= \int_{S\times \boldsymbol{B}\times\boldsymbol{\Theta}} ((I - \gamma\Psi)^{-1} r^i)(s, \boldsymbol{b}, \boldsymbol{\theta}) d\mu_0(s, \boldsymbol{b}, \boldsymbol{\theta}) \tag{C.122}$$

Since $\mu_0 = \delta_{(s_0, \boldsymbol{b}_0, \boldsymbol{\theta}_0)}$, we have:

$$v^i_{\theta^i}(s_0, \boldsymbol{b}_0, \boldsymbol{\theta}_0) = ((I - \gamma\Psi)^{-1} r^i)(s_0, \boldsymbol{b}_0, \boldsymbol{\theta}_0) \tag{C.123}$$

Alternatively, using the duality between $(I - \gamma\Psi)^{-1}$ and $(I - \gamma\Psi^*)^{-1}$:

$$\int_{S\times \boldsymbol{B}\times\boldsymbol{\Theta}} ((I - \gamma\Psi)^{-1} r^i)(s, \boldsymbol{b}, \boldsymbol{\theta}) d\mu_0(s, \boldsymbol{b}, \boldsymbol{\theta}) = \int_{S\times \boldsymbol{B}\times\boldsymbol{\Theta}} r^i(s, \boldsymbol{b}, \boldsymbol{\theta}) d((I - \gamma\Psi^*)^{-1}\mu_0)(s, \boldsymbol{b}, \boldsymbol{\theta}) \tag{C.124}$$

From previous lemmas, we know that $d^\pi_{\mu_0} = (1-\gamma)(I - \gamma\Psi^*)^{-1}\mu_0$. Therefore:

$$v^i_{\theta^i}(s_0, \boldsymbol{b}_0, \boldsymbol{\theta}_0) = \int_{S\times \boldsymbol{B}\times\boldsymbol{\Theta}} r^i(s, \boldsymbol{b}, \boldsymbol{\theta}) d((I - \gamma\Psi^*)^{-1}\mu_0)(s, \boldsymbol{b}, \boldsymbol{\theta}) \tag{C.125}$$

$$= \int_{S\times \boldsymbol{B}\times\boldsymbol{\Theta}} r^i(s, \boldsymbol{b}, \boldsymbol{\theta}) \frac{d^\pi_{\mu_0}(s, \boldsymbol{b}, \boldsymbol{\theta})}{1-\gamma} \tag{C.126}$$

$$= \frac{1}{1-\gamma} \int_{S\times \boldsymbol{B}\times\boldsymbol{\Theta}} r^i(s, \boldsymbol{b}, \boldsymbol{\theta}) d^\pi_{\mu_0}(s, \boldsymbol{b}, \boldsymbol{\theta}) \tag{C.127}$$

□

This lemma establishes that the value function can be expressed as an expectation with respect to the discounted visitation measure, with an appropriate scaling factor. This formulation provides a direct link between the value function and the distribution of states.

C.4 Policy Gradient Theorem

Having established the mathematical foundations for discounted returns in partially observable multi-agent settings, we now derive the policy gradient theorem that forms the basis for practical optimization algorithms. This theorem provides a principled expression for computing gradients of the expected discounted return with respect to policy parameters, enabling efficient policy improvement. The policy gradient theorem connects an agent's policy parameters to its expected long-term performance through the discounted visitation measure, providing a mathematically rigorous foundation for optimization.

Theorem 3 (Partially Observable Active Discounted Return Policy Gradient Theorem)**.** *The gradient of the discounted return objective* $J^i_{\pi,\gamma}(\theta^i) = v^i_{\theta^i}(s_0, \boldsymbol{b}_0, \boldsymbol{\theta}_0)$ *with respect to agent i's*

policy parameters θ^i in a partially observable active Markov game setting can be expressed as:

$$\nabla_{\theta^i} J^i_{\pi,\gamma}(\theta^i) = \frac{1}{1-\gamma} \sum_{s,\boldsymbol{b},\boldsymbol{\theta}} d^{\pi}_{\mu_0}(s,\boldsymbol{b},\boldsymbol{\theta}) \sum_{a^i} \nabla_{\theta^i} \pi^i(a^i|b^i;\theta^i) \sum_{a^{-i}} \pi^{-i}(a^{-i}|\boldsymbol{b}^{-i};\theta^{-i}) q^i_{\theta^i}(s,\boldsymbol{b},\boldsymbol{\theta},\boldsymbol{a}) \tag{C.128}$$

where $d^{\pi}_{\mu_0}(s,\boldsymbol{b},\boldsymbol{\theta})$ is the discounted visitation measure starting from initial distribution μ_0, and $q^i_{\theta^i}(s,\boldsymbol{b},\boldsymbol{\theta},\boldsymbol{a})$ is the action-value function defined as:

$$q^i_{\theta^i}(s,\boldsymbol{b},\boldsymbol{\theta},\boldsymbol{a}) = R^i(s,\boldsymbol{a}) + \gamma \sum_{s'} T(s'|s,\boldsymbol{a}) \sum_{\boldsymbol{o}'} \boldsymbol{O}(\boldsymbol{o}'|s') \sum_{\boldsymbol{\theta}'} \boldsymbol{U}(\boldsymbol{\theta}'|\boldsymbol{\theta},\boldsymbol{\tau}) v^i_{\theta^i}(s', update(\boldsymbol{b},\boldsymbol{a},\boldsymbol{o}'),\boldsymbol{\theta}') \tag{C.129}$$

We now provide a detailed proof of the policy gradient theorem, building on the lemmas established in previous sections.

Proof. From our previous lemma, we know that the gradient of the value function can be expressed as:

$$\nabla_{\theta^i} v^i_{\theta^i}(s,\boldsymbol{b},\boldsymbol{\theta}) = (I - \gamma\Psi)^{-1} \left[\sum_{a^i} \nabla_{\theta^i} \pi^i(a^i|b^i;\theta^i) \sum_{a^{-i}} \pi^{-i}(a^{-i}|\boldsymbol{b}^{-i};\theta^{-i}) q^i_{\theta^i}(s,\boldsymbol{b},\boldsymbol{\theta},\boldsymbol{a}) \right] \tag{C.130}$$

For notational clarity, we define the term inside the brackets as:

$$g(s,\boldsymbol{b},\boldsymbol{\theta}) = \sum_{a^i} \nabla_{\theta^i} \pi^i(a^i|b^i;\theta^i) \sum_{a^{-i}} \pi^{-i}(a^{-i}|\boldsymbol{b}^{-i};\theta^{-i}) q^i_{\theta^i}(s,\boldsymbol{b},\boldsymbol{\theta},\boldsymbol{a}) \tag{C.131}$$

Now we can write:

$$\nabla_{\theta^i} v^i_{\theta^i}(s,\boldsymbol{b},\boldsymbol{\theta}) = (I - \gamma\Psi)^{-1} g(s,\boldsymbol{b},\boldsymbol{\theta}) \tag{C.132}$$

For our specific initial state configuration $(s_0,\boldsymbol{b}_0,\boldsymbol{\theta}_0)$:

$$\nabla_{\theta^i} v^i_{\theta^i}(s_0,\boldsymbol{b}_0,\boldsymbol{\theta}_0) = ((I - \gamma\Psi)^{-1} g)(s_0,\boldsymbol{b}_0,\boldsymbol{\theta}_0) \tag{C.133}$$

Let's define $\mu_0 = \delta_{(s_0,\boldsymbol{b}_0,\boldsymbol{\theta}_0)}$, which is the Dirac measure concentrated at the initial state. We can view $((I - \gamma\Psi)^{-1} g)(s_0,\boldsymbol{b}_0,\boldsymbol{\theta}_0)$ as the integration of the function $(I - \gamma\Psi)^{-1} g$ against this Dirac measure:

$$\nabla_{\theta^i} v^i_{\theta^i}(s_0,\boldsymbol{b}_0,\boldsymbol{\theta}_0) = ((I - \gamma\Psi)^{-1} g)(s_0,\boldsymbol{b}_0,\boldsymbol{\theta}_0) \tag{C.134}$$

$$= \int ((I - \gamma\Psi)^{-1} g)(s,\boldsymbol{b},\boldsymbol{\theta}) d\delta_{(s_0,\boldsymbol{b}_0,\boldsymbol{\theta}_0)}(s,\boldsymbol{b},\boldsymbol{\theta}) \tag{C.135}$$

Recall the duality principle we established earlier, which states that:

$$\int ((I-\gamma\Psi)^{-1}f)(x)d\mu(x) = \int f(x)d((I-\gamma\Psi^*)^{-1}\mu)(x) \tag{C.136}$$

Allowing us to move the operator from acting on the function g to acting on the measure μ_0:

$$\int ((I-\gamma\Psi)^{-1}g)(s,\boldsymbol{b},\boldsymbol{\theta})d\mu_0(s,\boldsymbol{b},\boldsymbol{\theta}) = \int g(s,\boldsymbol{b},\boldsymbol{\theta})d((I-\gamma\Psi^*)^{-1}\mu_0)(s,\boldsymbol{b},\boldsymbol{\theta}) \tag{C.137}$$

Therefore:

$$\nabla_{\theta^i} v^i_{\theta^i}(s_0,\boldsymbol{b}_0,\boldsymbol{\theta}_0) = \int g(s,\boldsymbol{b},\boldsymbol{\theta})d((I-\gamma\Psi^*)^{-1}\delta_{(s_0,\boldsymbol{b}_0,\boldsymbol{\theta}_0)})(s,\boldsymbol{b},\boldsymbol{\theta}) \tag{C.138}$$

From our earlier derivation of the discounted visitation measure, we established that:

$$d^\pi_{\mu_0} = (1-\gamma)(I-\gamma\Psi^*)^{-1}\mu_0 = (1-\gamma)(I-\gamma\Psi^*)^{-1}\delta_{(s_0,\boldsymbol{b}_0,\boldsymbol{\theta}_0)} \tag{C.139}$$

We can now substitute this expression:

$$\nabla_{\theta^i} v^i_{\theta^i}(s_0,\boldsymbol{b}_0,\boldsymbol{\theta}_0) = \int g(s,\boldsymbol{b},\boldsymbol{\theta})d\left(\frac{d^\pi_{\mu_0}}{1-\gamma}\right)(s,\boldsymbol{b},\boldsymbol{\theta}) \tag{C.140}$$

$$= \frac{1}{1-\gamma}\int g(s,\boldsymbol{b},\boldsymbol{\theta})d^\pi_{\mu_0}(s,\boldsymbol{b},\boldsymbol{\theta}) \tag{C.141}$$

Substituting the definition of $g(s,\boldsymbol{b},\boldsymbol{\theta})$ back:

$$\nabla_{\theta^i} J^i_{\pi,\gamma}(\theta^i) = \nabla_{\theta^i} v^i_{\theta^i}(s_0,\boldsymbol{b}_0,\boldsymbol{\theta}_0) \tag{C.142}$$

$$= \frac{1}{1-\gamma}\int_{S\times\boldsymbol{B}\times\boldsymbol{\Theta}} \sum_{a^i} \nabla_{\theta^i}\pi^i(a^i|b^i;\theta^i)\sum_{a^{-i}}\pi^{-i}(a^{-i}|\boldsymbol{b}^{-i};\theta^{-i})q^i_{\theta^i}(s,\boldsymbol{b},\boldsymbol{\theta},\boldsymbol{a})d^\pi_{\mu_0}(s,\boldsymbol{b},\boldsymbol{\theta}) \tag{C.143}$$

For finite state, belief, and policy parameter spaces, this integral can be written as a sum:

$$\nabla_{\theta^i} J^i_{\pi,\gamma}(\theta^i) = \frac{1}{1-\gamma}\sum_{s,\boldsymbol{b},\boldsymbol{\theta}} d^\pi_{\mu_0}(s,\boldsymbol{b},\boldsymbol{\theta})\sum_{a^i}\nabla_{\theta^i}\pi^i(a^i|b^i;\theta^i)\sum_{a^{-i}}\pi^{-i}(a^{-i}|\boldsymbol{b}^{-i};\theta^{-i})q^i_{\theta^i}(s,\boldsymbol{b},\boldsymbol{\theta},\boldsymbol{a}) \tag{C.144}$$

This completes the proof of the Partially Observable Active Discounted Return Policy Gradient Theorem. □

The proof reveals a symmetry in the mathematical structure of our results.

Remark 2 (Symmetry)**.** *From the Value Function as Inner Product lemma, the value function*

$$v^i_{\theta^i}(s_0,\boldsymbol{b}_0,\boldsymbol{\theta}_0) = \frac{1}{1-\gamma}\int_{S\times\boldsymbol{B}\times\boldsymbol{\Theta}} r^i(s,\boldsymbol{b},\boldsymbol{\theta})d^\pi_{\mu_0}(s,\boldsymbol{b},\boldsymbol{\theta}) \tag{C.145}$$

is an expectation of immediate rewards r^i under the discounted visitation measure $d^{\pi}_{\mu_0}$. While its gradient

$$\nabla_{\theta^i} v^i_{\theta^i}(s_0, \boldsymbol{b}_0, \boldsymbol{\theta}_0) = \frac{1}{1-\gamma} \int_{S \times \boldsymbol{B} \times \boldsymbol{\Theta}} g(s, \boldsymbol{b}, \boldsymbol{\theta}) d^{\pi}_{\mu_0}(s, \boldsymbol{b}, \boldsymbol{\theta}) \tag{C.146}$$

is an expectation of policy-gradient-weighted action values g under the same discounted visitation measure $d^{\pi}_{\mu_0}$.

This symmetric structure provides not only mathematical elegance but also practical advantages. It means that the same samples from the discounted visitation distribution can be used to estimate both values and policy gradients.

The policy gradient theorem provides the mathematical foundation for agents to optimize their policies in ways that account for both the partial observability of the environment and the learning dynamics of other agents. This enables sophisticated strategic behaviors such as information revelation, teaching, and influence that are essential in social learning contexts.

POLARIS ARCHITECTURE

D

This section provides detailed technical specifications of all neural network architectures used in our POLARIS implementation. We describe each component's structure, activation functions, and design considerations to enable reproducibility and thorough understanding of our approach.

D.1 BELIEF PROCESSING MODULE

In our POLARIS implementation, agents maintain and update belief states using a Transformer-based architecture **?**, which effectively processes sequential observations while preserving the temporal relationships critical for social learning tasks.

The TransformerBeliefProcessor replaces traditional recurrent architectures with a self-attention mechanism that can capture complex dependencies across observation sequences. The module takes a signal (private observation), neighbor actions, and the current belief state as inputs, then produces an updated belief state and belief distribution over possible environmental states.

The belief processor begins with an input projection layer that maps the concatenated signal and neighbor action vectors to a hidden dimension. Specifically, for an agent receiving signal o_t and observing neighbor actions $\boldsymbol{a}_t$, the input vector $x_t = [o_t, \boldsymbol{a}_t]$ is projected as:

$$x_{\text{projected}} = W_{\text{proj}} x_t + b_{\text{proj}} \tag{D.1}$$

This projection is then passed through a transformer encoder with multi-head self-attention. The transformer encoder consists of attention layers defined as:

$$\text{Attention}(Q,K,V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{D.2}$$

where Q, K, and V are the query, key, and value matrices derived from the input, and d_k is the dimensionality of the key vectors. The multi-head attention mechanism applies this attention function in parallel across multiple representation subspaces, then concatenates the results:

$$\text{MultiHead}(X) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)W^O \tag{D.3}$$

where each head is computed as $\text{head}_i = \text{Attention}(XW_i^Q, XW_i^K, XW_i^V)$. Our implementation uses 4 attention heads with a dimension of 64 per head.

The transformer encoder is followed by feed-forward networks with ReLU activations:

$$\text{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2 \tag{D.4}$$

A belief head at the output projects the transformed representation to a probability distribution over possible environment states:

$$b_{\text{dist}} = \text{softmax}(W_{\text{belief}} h_{\text{final}} + b_{\text{belief}}) \tag{D.5}$$

Where h_{final} is the final hidden state from the transformer. This distribution represents the agent's belief about the current environmental state. The transformed representation also becomes the new belief state that is maintained across time steps.

The transformer architecture offers several advantages for belief processing in partially observable environments. First, it can model arbitrarily long-range dependencies in observation sequences without the vanishing gradient issues that affect recurrent networks. Second, the self-attention mechanism provides a natural way to weight the importance of different observations in forming beliefs. Third, the positional encodings allow the model to incorporate temporal information while maintaining permutation invariance within each time step.

D.2 Inference Learning Module

The Inference Learning Module enables agents to predict and model the behavior of other agents, which is essential for effective social learning. POLARIS implements a Graph Neural Network (GNN) approach for this module, which explicitly captures the network structure of agent interactions.

The GNN-based inference module represents agents and their interactions as a graph, where nodes correspond to agents and edges represent observational relationships. For each node, features include the agent's observation and action.

The graph representation treats each agent as a node, with edges representing observational relationships. The GNN uses graph attention layers (GAT) that compute attention coefficients between connected nodes:

$$e_{ij} = \text{LeakyReLU}(a^T [Wh_i || Wh_j]) \tag{D.6}$$

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k \in \mathcal{N}_i} \exp(e_{ik})} \tag{D.7}$$

where h_i represents the features of node i, W is a learned weight matrix, a is an attention vector, and $\mathcal{N}_i$ is the neighborhood of node i. The node representations are then updated using these attention weights:

$$h'_i = \sigma\left(\sum_{j \in \mathcal{N}_i} \alpha_{ij} W h_j\right) \tag{D.8}$$

The temporal aspect is handled through a window-based memory system that maintains a

history of previous graph states. A multi-head attention mechanism operates across this temporal window:

$$\text{TemporalAttention}(X_t, X_{t-1:t-k}) = \text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right)V \tag{D.9}$$

where X_t is the current graph representation, $X_{t-1:t-k}$ represents previous graph states, and Q, K, V are learned projections. The GNN implementation uses configurable parameters for the number of graph layers (default: 2), attention heads (default: 4), and temporal window size (default: 5).

The GNN receives a combination of the current latent estimate, the agent's observations, actions, and rewards. Formally, its input for agent i is defined as $x_i = [\hat{\boldsymbol{z}}_t^{-i}, o_t^i, \boldsymbol{a}_t, r_t^i, o_{t+1}^i]$, where $\hat{\boldsymbol{z}}_t^{-i} \in \mathbb{R}^d$ represents the current estimate of other agents' latent states with dimension $d = 256$ in our implementation.

After processing through the graph layers and temporal attention mechanism, the GNN outputs parameters of a Gaussian distribution over the next latent state:

$$\mu_t = W_\mu h_i^{\text{final}} + b_\mu \tag{D.10}$$

$$\log \sigma_t = W_\sigma h_i^{\text{final}} + b_\sigma \tag{D.11}$$

where μ_t is the mean vector and $\log \sigma_t$ is the log-standard-deviation vector of the distribution, and h_i^{final} is the final node representation for agent i.

The GNN also outputs an opponent belief distribution through a separate head:

$$p_{\text{opp}} = \text{softmax}(W_{\text{opp}} h_i^{\text{final}} + b_{\text{opp}}) \tag{D.12}$$

This distribution represents the agent's prediction of other agents' beliefs about the environment state.

The latent variables are sampled using the reparameterization trick to enable backpropagation:

$$\hat{\boldsymbol{z}}_{t+1}^{-i} = \mu_t + \exp(0.5 \cdot \log \sigma_t) \odot \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, I) \tag{D.13}$$

The GNN-based inference module is trained using the Evidence Lower Bound (ELBO) objective function:

$$J_{\text{elbo}}^i = \mathbb{E}[\log \mathbb{P}(\boldsymbol{a}_t^{-i} | o_t^i, \hat{\boldsymbol{z}}_t^{-i}) - \alpha_{\text{KL}} D_{\text{KL}}(\mathcal{N}(\hat{\boldsymbol{z}}_{t+1}^{-i} | \tau_t^i) || \mathcal{N}(\hat{\boldsymbol{z}}_t^{-i}))] \tag{D.14}$$

where the first term represents the reconstruction log-likelihood and the second term is a KL divergence regularization that encourages temporal consistency in the latent space. The hyperparameter α_{KL} controls the strength of this regularization.

D.3 REINFORCEMENT LEARNING MODULE

The reinforcement learning module optimizes the agent's policy based on beliefs, inferences about other agents, and received rewards. POLARIS implements a soft actor-critic (SAC) **?** framework that balances exploitation with exploration through entropy regularization.

D.3.1 Policy Network Architecture

The policy network translates belief states and inferred latent variables into action decisions. It represents the agent's strategy for maximizing long-term rewards in the social learning environment. The policy network takes the agent's belief state and the inferred latent variables of other agents as input: $x_\pi = [b_t^i, \hat{\mathbf{z}}_t^{-i}]$.

The policy is implemented as an MLP with ReLU activations across multiple hidden layers:

$$h_\pi^1 = \text{ReLU}(W_\pi^1 x_\pi + b_\pi^1) \tag{D.15}$$

$$h_\pi^2 = \text{ReLU}(W_\pi^2 h_\pi^1 + b_\pi^2) \tag{D.16}$$

Each hidden layer contains 256 units. For discrete action spaces, the policy outputs action probabilities:

$$\pi^i(a^i | b_t^i, \hat{\mathbf{z}}_t^{-i}; \theta^i) = \text{softmax}(W_{\text{out}} h_\pi^2 + b_{\text{out}}) \tag{D.17}$$

where W_{out} and b_{out} are part of the policy parameters θ^i.

For continuous action spaces, the policy outputs the mean and log standard deviation of a Gaussian distribution:

$$\mu(b_t^i, \hat{\mathbf{z}}_t^{-i}; \theta^i) = W_\mu h_\pi^2 + b_\mu \tag{D.18}$$

$$\log \sigma(b_t^i, \hat{\mathbf{z}}_t^{-i}; \theta^i) = W_\sigma h_\pi^2 + b_\sigma \tag{D.19}$$

To sample actions for continuous spaces, we use the reparameterization trick, which enables gradient flow through the sampling process:

$$a_{\text{raw}}^i = \mu + \sigma \odot \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, I) \tag{D.20}$$

A squashing function (tanh) is then applied to bound the actions within the range $[-1, 1]$:

$$a^i = \tanh(a_{\text{raw}}^i) \tag{D.21}$$

This bounded action can be scaled to the appropriate range required by the environment. When using this squashing function, the log probability calculation must be adjusted using the change of variables formula:

$$\log \pi^i(a^i|b_t^i, \hat{\mathbf{z}}_t^{-i}; \theta^i) = \log \mathcal{N}(a_{\text{raw}}^i|\mu, \sigma) - \sum_j \log(1 - \tanh^2(a_{\text{raw},j}^i)) \tag{D.22}$$

where the second term is the logarithm of the absolute determinant of the Jacobian of the tanh transformation.

Following the soft actor-critic framework, the policy is trained to maximize both expected returns and entropy:

$$J_\pi^i(\theta^i) = \mathbb{E}[\min_{\beta=1,2} q_{\theta^i}^i(b_t^i, \hat{\mathbf{z}}_t^{-i}, a_t^i; \psi_\beta^i) + \alpha_e H(\pi^i(\cdot|b_t^i, \hat{\mathbf{z}}_t^{-i}; \theta^i))] \tag{D.23}$$

where α_e is the entropy weight (temperature parameter) that controls the balance between exploitation and exploration. Higher values of α_e encourage more exploration.

For discrete action spaces, the entropy is calculated directly from the action probabilities:

$$H(\pi^i) = -\sum_{a^i} \pi^i(a^i|b_t^i, \hat{\mathbf{z}}_t^{-i}; \theta^i) \log \pi^i(a^i|b_t^i, \hat{\mathbf{z}}_t^{-i}; \theta^i) \tag{D.24}$$

For continuous action spaces, the entropy of the base Gaussian distribution is used:

$$H(\pi^i) = \frac{1}{2} \log \det(2\pi e \sigma^2(b_t^i, \hat{\mathbf{z}}_t^{-i}; \theta^i)) \tag{D.25}$$

When using the tanh squashing function, this entropy calculation must be adjusted to account for the change in probability density.

D.3.2 Value Network Architecture

The value network estimates the expected future rewards for state-action pairs, providing the foundation for policy optimization. POLARIS implements a dual critic approach to mitigate overestimation bias, a common issue in Q-learning algorithms.

The value network receives the belief state, inferred latent variables, and joint action as input: $x_q = [b_t^i, \hat{\mathbf{z}}_t^{-i}, \boldsymbol{a}_t]$. Each value network follows an MLP architecture with three hidden layers:

$$h_q^1 = \text{ReLU}(W_q^1 x_q + b_q^1) \tag{D.26}$$

$$h_q^2 = \text{ReLU}(W_q^2 h_q^1 + b_q^2) \tag{D.27}$$

$$h_q^3 = \text{ReLU}(W_q^3 h_q^2 + b_q^3) \tag{D.28}$$

Each hidden layer contains 256 units. The network outputs a scalar value estimate:

$$q_{\theta^i}^i(b_t^i, \hat{\mathbf{z}}_t^{-i}, \boldsymbol{a}_t; \psi_\beta^i) = W_{q,\text{out}} h_q^3 + b_{q,\text{out}} \tag{D.29}$$

where $\beta \in \{1, 2\}$ indicates which of the two value networks is being used. The value function is trained using the soft Bellman equation, which incorporates the entropy-regularized ob-

jective:

$$y = r_t^i - \rho_{\theta^i}^i + v_{\theta^i}^i(b_{t+1}^i, \hat{\mathbf{z}}_{t+1}^{-i}; \bar{\psi}_\beta^i) \tag{D.30}$$

$$J_q^i(\psi_\beta^i, \rho_{\theta^i}^i) = \mathbb{E}[(y - q_{\theta^i}^i(b_t^i, \hat{\mathbf{z}}_t^{-i}, \boldsymbol{a}_t; \psi_\beta^i))^2] \tag{D.31}$$

where $\bar{\psi}_\beta^i$ are the target network parameters, updated using an exponential moving average:

$$\bar{\psi}_\beta^i \leftarrow \tau_q \psi_\beta^i + (1 - \tau_q)\bar{\psi}_\beta^i \tag{D.32}$$

with $\tau_q = 0.005$ in our implementation. The term $\rho_{\theta^i}^i$ represents the average reward, which is used for differential returns in continuing tasks. The soft value function incorporates policy entropy:

$$v_{\theta^i}^i(b_t^i, \hat{\mathbf{z}}_t^{-i}) = \mathbb{E}_{a^i \sim \pi^i(\cdot|b_t^i, \hat{\mathbf{z}}_t^{-i}; \theta^i)}[\min_{\beta=1,2} q_{\theta^i}^i(b_t^i, \hat{\mathbf{z}}_t^{-i}, a_t^i; \psi_\beta^i)] + \alpha_e H(\pi^i(\cdot|b_t^i, \hat{\mathbf{z}}_t^{-i}; \theta^i)) \tag{D.33}$$

POLARIS can operate in both discounted and average reward modes. In discounted mode, the target is simply $y = r_t^i + \gamma \cdot v_{\theta^i}^i(b_{t+1}^i, \hat{\mathbf{z}}_{t+1}^{-i}; \bar{\psi}_\beta^i)$, where γ is the discount factor. In average reward mode, the differential returns formulation with $\rho_{\theta^i}^i$ is used instead.

The value network architecture directly implements our theoretical framework of active Markov games, estimating returns that account for the influence of current actions on future environmental states and the policy parameters of other agents.

D.4 Elastic Weight Consolidation

POLARIS implements Elastic Weight Consolidation (EWC) **?** to prevent catastrophic forgetting when agents need to maintain performance across multiple tasks or environments. EWC adds a regularization term to the loss function that penalizes large changes to parameters that were important for previously learned tasks.

The EWC loss is defined as:

$$\mathscr{L}_{\text{EWC}} = \lambda \sum_i F_i(\theta^i - \theta^{i*})^2 \tag{D.34}$$

where λ is the importance factor, F_i is the Fisher information for parameter θ^i, and θ^{i*} is the optimal parameter value for the previous task. The Fisher information matrix is calculated by:

$$F = \mathbb{E}_{x \sim \mathscr{D}, y \sim \mathbb{P}(y|x;\theta)} \left[\nabla_\theta \log \mathbb{P}(y|x;\theta) \nabla_\theta \log \mathbb{P}(y|x;\theta)^T \right] \tag{D.35}$$

In practice, POLARIS approximates this using samples from the replay buffer. The implementation offers both standard EWC and online EWC variants. The online variant continuously updates the Fisher matrix with a decay factor, allowing for sequential learning across multiple

tasks without storing separate Fisher matrices:

$$F_{\text{new}} = \gamma F_{\text{old}} + F_{\text{current}} \tag{D.36}$$

where γ is a decay factor typically set to 0.95. EWC is particularly valuable in social learning contexts where agents may need to adapt to different partner policies without forgetting effective strategies learned previously.

In practice, all these neural network components operate in tandem, with the belief state processor tracking the environmental state, the inference module predicting other agents' behavior, and the reinforcement learning module optimizing the agent's policy. This integrated approach enables POLARIS to effectively navigate the complexities of partially observable multi-agent environments with non-stationary dynamics.

D.5 Implementation Details

POLARIS incorporates several crucial implementation details that enhance its performance in social learning environments. Double Q-learning is employed to prevent overestimation bias, a common issue in value-based reinforcement learning methods. This approach maintains two separate Q-networks and uses the minimum of their predictions for target value calculations, providing a more pessimistic and reliable estimate of expected returns. Such conservative value estimation is particularly important in social learning, where overestimated values can lead to overly aggressive strategies that fail to account for the strategic adaptations of other agents.

Temperature scaling is applied to softmax outputs in the belief distribution calculations, controlling the sharpness of the resulting probability distributions. This technique produces more stable belief distributions, preventing premature convergence to overly confident beliefs based on limited evidence. The temperature parameter can be adjusted to balance exploration and exploitation in the belief space, with higher values producing more uniform distributions that encourage consideration of alternative hypotheses about the environment state.

For robust latent sampling from the variational posterior, POLARIS employs several numerical stability safeguards. These include clamping log-variances to prevent extreme values, enforcing minimum and maximum bounds on variances, and using epsilon terms to avoid division by zero or other numerical instabilities. These precautions are essential when working with variational methods in complex, high-dimensional spaces, ensuring reliable performance even with noisy or ambiguous social signals.

Gradient clipping is applied across all network updates to prevent exploding gradients, a common issue in deep reinforcement learning, especially with recurrent or transformer architectures. By limiting the maximum gradient norm, POLARIS maintains stable training even when encountering unusual observations or reward patterns. Additionally, separate optimizers are maintained for each network component, allowing for different learning rates and optimization strategies tailored to the specific characteristics of each module.

The POLARIS implementation significantly advances beyond the original FURTHER frame-

work by introducing sophisticated belief modeling through Transformers, network-aware representations via GNNs, advantage-weighted Transformer training, and catastrophic forgetting prevention through EWC. These enhancements enable more effective learning in the complex, partially observable social environments that characterize our theoretical framework. By integrating these components into a cohesive algorithm, POLARIS provides a practical realization of the theoretical Partially Observable Active Markov Game framework, offering a powerful tool for studying social learning dynamics across various network topologies and environmental conditions.

SOCIAL LEARNING IMPLEMENTATION

E

This chapter presents the technical implementation details of our social learning models within the Partially Observable Active Markov Game framework. We provide a comprehensive account of the algorithmic approaches, mathematical techniques, and computational methods used to translate theoretical social learning concepts into executable simulations. The discussion begins with the rigorous discretization of Lévy processes—the mathematical foundation for modeling stochastic rewards and signals—and proceeds to cover the specific implementation challenges encountered in strategic experimentation scenarios. Another key innovation is our construction of observation-based reward functions that preserve expected reward structures when agents cannot directly observe the true environmental state, allowing them to learn optimal policies from noisy signals while maintaining the incentive properties of the original models. Throughout this chapter, we emphasize how our implementation preserves the strategic incentives inherent in the original models while adapting them to a computational framework suitable for reinforcement learning applications.

E.1 LÉVY PROCESS DISCRETIZATION

This appendix provides a comprehensive mathematical treatment of the discretization of Lévy processes for implementing strategic experimentation models within our Partially Observable Active Markov Game framework. We establish rigorous theoretical foundations for the numerical approximation schemes used in our implementation and analyze their convergence properties and preservation of strategic incentives.

E.1.1 Mathematical Foundations of Lévy Processes

Lévy processes form a fundamental class of stochastic processes that include Brownian motion and Poisson processes as special cases. They are characterized by stationary and independent increments, serving as the natural continuous-time generalization of random walks.

Definition 11 (Lévy Process)**.** *A stochastic process $X = \{X_t : t \geq 0\}$ on $\mathbb{R}^d$ with $X_0 = 0$ almost surely is a Lévy process if:*

1. *It has independent increments: for any $0 \leq t_1 < t_2 < \cdots < t_n < \infty$, the random variables $X_{t_2} - X_{t_1}, X_{t_3} - X_{t_2}, \ldots, X_{t_n} - X_{t_{n-1}}$ are mutually independent.*

2. *It has stationary increments: for any $s < t$, the distribution of $X_t - X_s$ depends only on $t - s$.*

3. It is stochastically continuous: for any $t \geq 0$ and $\varepsilon > 0$, $\lim_{h\to 0} \mathbb{P}(|X_{t+h} - X_t| > \varepsilon) = 0$.

The celebrated Lévy-Khintchine formula provides a complete characterization of Lévy processes through their characteristic functions:

Theorem 4 (Lévy-Khintchine Formula **??**)**.** *If $X = \{X_t : t \geq 0\}$ is a Lévy process, then its characteristic function has the form:*

$$\mathbb{E}[e^{i\theta X_t}] = e^{t\psi(\theta)} \tag{E.1}$$

where

$$\psi(\theta) = i\alpha\theta - \frac{1}{2}\sigma^2\theta^2 + \int_{\mathbb{R}\setminus\{0\}} (e^{i\theta x} - 1 - i\theta x \mathbf{1}_{|x|<1})\nu(dx) \tag{E.2}$$

for some $\alpha \in \mathbb{R}$, $\sigma \geq 0$, and a measure ν on $\mathbb{R}\setminus\{0\}$ satisfying $\int_{\mathbb{R}\setminus\{0\}} \min(1, x^2)\nu(dx) < \infty$.

The triplet (α, σ^2, ν) is called the Lévy-Khintchine triplet or the characteristics of the Lévy process. Here, α represents a drift term, σ^2 parametrizes the continuous Gaussian component, and ν is the Lévy measure characterizing the jump behavior.

Theorem 5 (Lévy-Itô Decomposition **??**)**.** *Any Lévy process X_t can be decomposed as:*

$$X_t = \alpha t + \sigma W_t + \int_{|x|<1} x(\tilde{Y}(t, dx) - t\nu(dx)) + \int_{|x|\geq 1} xY(t, dx) \tag{E.3}$$

where W_t is a standard Brownian motion, $Y(t, A)$ counts the number of jumps of size in set A occurring up to time t, and $\tilde{Y}(t, dx) = Y(t, dx) - t\nu(dx)$ is the compensated Poisson random measure.

The Lévy-Itô decomposition provides a pathwise representation of a Lévy process as the sum of a drift term, a Brownian motion, and potentially infinitely many jumps, both small and large.

E.1.2 Time Discretization of Lévy Processes

To implement continuous-time Lévy processes within discrete computational frameworks, we must employ appropriate numerical approximation schemes. For our strategic experimentation model, we adopt the Euler-Maruyama scheme, extended to accommodate the jump components of general Lévy processes.

Definition 12 (Euler-Maruyama Scheme for Lévy Processes)**.** *Given a Lévy process X_t with characteristics (α, σ^2, ν) and a discretization time step Δt, the Euler-Maruyama approximation constructs a discrete-time process $\{X_{t_n}\}_{n=0}^N$ where $t_n = n\Delta t$ through the recursive relation:*

$$X_{t_{n+1}} = X_{t_n} + \alpha\Delta t + \sigma\sqrt{\Delta t}Z_n + \Delta J_n \tag{E.4}$$

where $Z_n \sim \mathcal{N}(0, 1)$ are independent standard normal random variables and ΔJ_n represents the jump increment over $[t_n, t_{n+1}]$.

Theorem 6 (Convergence of Euler-Maruyama Scheme (**??**))**.** *Let X_t be a Lévy process and $\hat{X}_t$ be its Euler-Maruyama approximation with time step Δt. Then for any fixed $T > 0$:*

1. *(Weak Convergence) For any smooth function f with polynomial growth:*

$$|\mathbb{E}[f(X_T)] - \mathbb{E}[f(\hat{X}_T)]| \leq C\Delta t \tag{E.5}$$

2. *(Strong Convergence) If $\int_{|x|>1} |x|^2 \nu(dx) < \infty$, then:*

$$\mathbb{E}[\sup_{0\leq t\leq T} |X_t - \hat{X}_t|^2] \leq C\Delta t \tag{E.6}$$

where C is a constant depending on T and the characteristics of the Lévy process.

The weak and strong convergence properties ensure that our numerical scheme accurately approximates both the distributional properties and pathwise behavior of the continuous-time process as the time step decreases.

E.1.3 Implementing Strategic Experimentation Models

In our implementation of the strategic experimentation model from **?**, we must discretize both the background signal process B_t and the individual payoff processes X_t^i, while preserving the strategic incentives that drive experimentation decisions.

Discretization of Diffusion-Poisson Processes

In the original model, both B_t and X_t^i follow Lévy processes that combine continuous diffusion and discrete jumps:

$$dB_t = \beta_s dt + \sigma_B dW_t^B + dY_t^B \tag{E.7}$$
$$dX_t^i = \alpha_s dt + \sigma dW_t^i + dY_t^i \tag{E.8}$$

where W_t^B and W_t^i are standard Brownian motions, and Y_t^B and Y_t^i are compound Poisson processes. Using the Euler-Maruyama scheme (**?**), we discretize these continuous-time stochastic differential equations:

$$B_{t+\Delta t} - B_t = \beta_s \Delta t + \sigma_B (W_{t+\Delta t}^B - W_t^B) + (Y_{t+\Delta t}^B - Y_t^B) \tag{E.9}$$
$$X_{t+\Delta t}^i - X_t^i = \alpha_s \Delta t + \sigma (W_{t+\Delta t}^i - W_t^i) + (Y_{t+\Delta t}^i - Y_t^i) \tag{E.10}$$

For implementation, we denote these increments as:

$$B_{t-1:t} = \beta_s \Delta t + \sigma_B (W_t^B - W_{t-1}^B) + \Delta Y_t^B \tag{E.11}$$

$$X_{t-1:t}^i = \alpha_s \Delta t + \sigma (W_t^i - W_{t-1}^i) + \Delta Y_t^i \tag{E.12}$$

where $(W_t^B - W_{t-1}^B) \sim \mathcal{N}(0, \Delta t)$, $(W_t^i - W_{t-1}^i) \sim \mathcal{N}(0, \Delta t)$, and ΔY_t^B, ΔY_t^i are the increments of the compound Poisson processes over the interval $[t-1, t]$.

Reward Function Equivalence

A critical challenge in our implementation is reconciling the time-dependent nature of continuous-time rewards with the time-independent reward structure required by the POAMG framework. In the original continuous-time model, agents' instantaneous rewards are:

$$dR_t^i = (1 - a_t^i) r_{safe} dt + a_t^i dX_t^i \tag{E.13}$$

where $a_t^i \in [0,1]$ is the allocation to the risky arm and r_{safe} is the safe arm's deterministic flow payoff and the Levy process increment is a function of the state s. To translate this structure into the POAMG framework, we need a reward function $R^i(s, a^i)$ that depends only on the state and action, not explicitly on time. We achieve this through a normalization approach:

$$R^i(s, a^i) = (1 - a^i) r_{safe} + a^i \frac{X_{t-1:t}^i}{\Delta t} \tag{E.14}$$

This transformation preserves the incentive structure of the original model while eliminating explicit time dependence. The following proposition establishes this equivalence formally:

Proposition 1 (Reward Equivalence)**.** *The expected value of the discrete-time reward function $R^i(s, a^i)$ exactly equals the expected instantaneous flow payoff in the continuous-time model. Specifically:*

$$\mathbb{E}[R^i(s, a^i)] = (1 - a^i) r_{safe} + a^i (\alpha_s + \lambda_s h_s) \tag{E.15}$$

where α_s is the drift of the Levy process, λ_s is the jump intensity and h_s is the mean jump size of the compound Poisson process component of X_t^i in state s.

Proof. The expected value of the discrete-time reward function is:

$$\mathbb{E}[R^i(s, a^i)] = (1 - a^i) r_{safe} + a^i \mathbb{E}\left[\frac{X_{t-1:t}^i}{\Delta t}\right] \tag{E.16}$$

$$= (1 - a^i) r_{safe} + a^i \frac{\mathbb{E}[\alpha_s \Delta t + \sigma (W_t^i - W_{t-1}^i) + \Delta Y_t^i]}{\Delta t} \tag{E.17}$$

$$= (1 - a^i) r_{safe} + a^i \left(\alpha_s + \frac{\mathbb{E}[\Delta Y_t^i]}{\Delta t}\right) \tag{E.18}$$

Since $\mathbb{E}[W_t^i - W_{t-1}^i] = 0$ and $\mathbb{E}[\Delta Y_t^i] = \lambda_s h_s \Delta t$, where λ_s is the jump intensity and h_s is the mean jump size, we have:

$$\mathbb{E}[R^i(s,a^i)] = (1-a^i)r_{safe} + a^i\left(\alpha_s + \frac{\lambda_s h_s \Delta t}{\Delta t}\right) \tag{E.19}$$

$$= (1-a^i)r_{safe} + a^i(\alpha_s + \lambda_s h_s) \tag{E.20}$$

This exactly matches the expected instantaneous flow payoff in the continuous-time model. □

E.1.4 Relation to Policy-Invariant Reward Transformations

Our approach to time-independent reward formulation can be viewed through the lens of policy-invariant reward transformations, as developed in the reinforcement learning literature by **?**.

Definition 13 (Potential-Based Reward Shaping)**.** *A reward transformation $\tilde{R}(s,a,s') = R(s,a,s') + F(s,s')$ is potential-based if there exists a potential function $\Phi : S \to \mathbb{R}$ such that $F(s,s') = \gamma\Phi(s') - \Phi(s)$ for all $s, s' \in S$, where γ is a discount factor.*

Theorem 7 (Ng-Harada-Russell (**?**))**.** *If a reward transformation is potential-based, then the optimal policy under the transformed reward function is also optimal under the original reward function, and vice versa.*

While our transformation does not directly fit the potential-based formulation (since we're normalizing by Δt rather than adding a potential difference), it shares the crucial property of preserving optimal policies. The following result establishes this connection:

Proposition 2 (Connection to Policy-Invariant Transformations)**.** *The transformation from time-dependent rewards in the continuous-time model to time-independent rewards in the POAMG framework preserves policy optimality in the limit as $\Delta t \to 0$.*

Proof. The original cumulative reward over a time interval $[0,T]$ is:

$$\int_0^T [(1-a_t^i)s + a_t^i\alpha_s]dt + \int_0^T a_t^i\sigma dW_t^i + \int_0^T a_t^i dC_t^i \tag{E.21}$$

The discretized cumulative reward is:

$$\sum_{k=0}^{\eta-1} \Delta t \left[(1-a_{t_k}^i)s + a_{t_k}^i \frac{X_{t_k:t_{k+1}}^i}{\Delta t}\right] = \sum_{k=0}^{\eta-1} [(1-a_{t_k}^i)s\Delta t + a_{t_k}^i X_{t_k:t_{k+1}}^i] \tag{E.22}$$

where $\eta = T/\Delta t$ and $t_k = k\Delta t$.

As $\Delta t \to 0$ and $\eta \to \infty$ with $\eta t = T$ fixed, the discretized sum converges to the continuous-time integral. Since both formulations yield the same expected cumulative reward in the limit, they induce the same optimal policies. □

E.2 OBSERVED REWARD FUNCTION

In our reinforcement learning implementation, agents need to learn from observed rewards that reflect the optimality of their actions with respect to the true state of the world. However, since agents do not directly observe the true state ω, we need to construct a reward function based on their observations o that preserves the expected reward structure. This appendix provides a detailed derivation of the observation-based reward function.The true reward function is given by $u(s,a) : S \times A \to \mathbb{R}$, where a is an action and s is the state. However, agents only observe signals $o \sim \mu^s$ drawn from a distribution determined by the state. We need to construct an observation-based reward function $v(o,a) : \Omega_i \times A \to \mathbb{R}$ that satisfies:

$$\mathbb{E}_{o\sim\mu^s}[v(o,a)] = u(s,a), \quad \forall s \in S, \forall a \in A. \tag{E.23}$$

In other words, the expected value of the observation-based reward $v(o,a)$ over all possible observations o given the state s should equal the true reward $u(s,a)$ for any state-action pair. Let's define $\boldsymbol{\mu}$ as a $m \times m$ matrix representing the observation distributions, where $m = |S|$ and assume the aforementioned state and observation spaces. Each entry $\mu^s(o)$ denotes the probability of observing observation o in state s:

$$\boldsymbol{\mu} = \begin{bmatrix} \mu^{s^1}(s^1) & \mu^{s^1}(s^2) & \cdots & \mu^{s^1}(s^m) \\ \mu^{s^2}(s^1) & \mu^{s^2}(s^2) & \cdots & \mu^{s^2}(s^m) \\ \vdots & \vdots & \ddots & \vdots \\ \mu^{s^m}(s^1) & \mu^{s^m}(s^2) & \cdots & \mu^{s^m}(s^m) \end{bmatrix} \tag{E.24}$$

We assume that observation distributions are linearly independent across states, which implies that $\boldsymbol{\mu}$ is invertible. This assumption is satisfied when $\mu^s(s) > \mu^s(s')$ for all $s \neq s' \in S$, meaning that each state is more likely to generate its corresponding observation than any other observation. For each action $a \in A$, we define a utility vector $\boldsymbol{u}_a \in \mathbb{R}^k$ as:

$$\boldsymbol{u}_a = \begin{bmatrix} u(s^1,a) \\ u(s^2,a) \\ \vdots \\ u(s^m,a) \end{bmatrix} \tag{E.25}$$

Similarly, we define a vector of observation-based rewards $\boldsymbol{v}_a \in \mathbb{R}^k$ for action a:

$$\boldsymbol{v}_a = \begin{bmatrix} v(s^1,a) \\ v(s^2,a) \\ \vdots \\ v(s^m,a) \end{bmatrix} \tag{E.26}$$

The condition that expected observation-based rewards match true rewards can be written

as a linear system:

$$\boldsymbol{\mu}^T \boldsymbol{v}_a = \boldsymbol{u}_a \tag{E.27}$$

where $\boldsymbol{\mu}^T$ is the transpose of $\boldsymbol{\mu}$. The solution to this system is:

$$\boldsymbol{v}_a = (\boldsymbol{\mu}^T)^{-1} \boldsymbol{u}_a = (\boldsymbol{\mu}^{-1})^T \boldsymbol{u}_a \tag{E.28}$$

This gives us the vector of observation-based rewards that satisfy our expected reward condition. For each observation o_j and action a, the observation-based reward is:

$$v(o_j, a) = \sum_{l=1}^{m} \left(\boldsymbol{\mu}^{-1}\right)_{jl} u(s_l, a) \tag{E.29}$$

where $\left(\boldsymbol{\mu}^{-1}\right)_{jl}$ denotes the (j,l)-th element of $\boldsymbol{\mu}^{-1}$.

Binary Case Example For the special case with two states of the world ($m = 2$) and symmetric observations with accuracy $q > 0.5$, the observation distribution matrix is:

$$\boldsymbol{\mu} = \begin{bmatrix} q & 1-q \\ 1-q & q \end{bmatrix} \tag{E.30}$$

This means that in state s^1, the probability of observing s^1 is q and the probability of observing s^2 is $1-q$, and vice versa for state s^2. The inverse of this matrix is:

$$\boldsymbol{\mu}^{-1} = \frac{1}{2q-1} \begin{bmatrix} q & -(1-q) \\ -(1-q) & q \end{bmatrix} \tag{E.31}$$

In our implementation, we set the true reward as the indicator function for taking the correct action, which for the binary case becomes, $u(s^1, a^1) = u(s^2, a^2) = 1$; $u(s^2, a^1) = u(s^1, a^2) = 0$. For action a^1, the utility vector is $\boldsymbol{u}_{a^1} = [1, 0]^T$, and the observation-based reward vector is:

$$\boldsymbol{v}_{a^1} = \boldsymbol{\mu}^{-1} \boldsymbol{u}_{a^1} = \frac{1}{2q-1} \begin{bmatrix} q \\ -(1-q) \end{bmatrix} \tag{E.32}$$

This means that when an agent takes action a^1 and observes signal s^1, the reward is $v(s^1, a^1) = \frac{q}{2q-1}$, and when the agent observes signal s^2, the reward is $v(s^2, a^1) = -\frac{1-q}{2q-1}$. Similarly, for action a^2, the utility vector is $\boldsymbol{u}_{a^2} = [0, 1]^T$, and the observation-based reward vector is:

$$\boldsymbol{v}_{a^2} = \boldsymbol{\mu}^{-1} \boldsymbol{u}_{a^2} = \frac{1}{2q-1} \begin{bmatrix} -(1-q) \\ q \end{bmatrix} \tag{E.33}$$

Combining these results, we can express the observation-based reward function for any

action-observation pair (a, o) as:

$$v(a,o) = \frac{q \cdot \mathbf{1}_{\{a=\varphi(o)\}} - (1-q) \cdot \mathbf{1}_{\{a\neq\varphi(o)\}}}{2q-1} \tag{E.34}$$

where φ maps observations to their corresponding actions. The derived observation-based reward function has an intuitive interpretation. When an agent takes an action matching its observation ($a = o$), it receives a positive reward of $\frac{q}{2q-1}$, whereas when she takes an action not matching its observation ($a \neq o$), it receives a negative reward of $-\frac{1-q}{2q-1}$. As q approaches 0.5 (signals become uninformative), the rewards grow in magnitude, reflecting the increased uncertainty. Conversely, as q approaches 1 (signals become perfectly informative), the reward for matching observation and action approaches 1, and the penalty for mismatching approaches 0, meaning that the observed reward function converges to the true reward function. This reward structure incentivizes agents to match their actions to their observations when signals are reliable, but also allows them to learn to override their immediate observations when the actions of other agents provide stronger evidence about the true state through beliefs.

5.2 STATEMENT OF AUTHORSHIP

I hereby confirm that the work presented has been performed and interpreted solely by myself except for where I explicitly identified the contrary. I assure that this work has not been presented in any other form for the fulfillment of any other degree or qualification. Ideas taken from other works in letter and in spirit are identified in every single case.

Date: _____ Signature: _____