Population-based learning in simple stochastic games

Scott Merrill[1] and Alex McAvoy[2,3,*]

[1]Department of Computer Science, University of North Carolina at Chapel Hill, Chapel Hill, NC
[2]School of Data Science and Society, University of North Carolina at Chapel Hill, Chapel Hill, NC
[3]Department of Mathematics, University of North Carolina at Chapel Hill, Chapel Hill, NC

December 30, 2025

Abstract

Conflicts of interest are ubiquitous in populations. When individuals interact, there are often discrepancies between what is best for the individual and what is best for the larger group. Social dilemmas capture the differing incentives between individuals and groups, and specific models like the prisoner's dilemma have been studied extensively in both evolutionary game theory and multi-agent reinforcement learning. However, at the intersection of these fields lies the understudied question of how population-level stochasticity affects collective learning dynamics and emergent behaviors. In this work, we study the impact of random interactions in populations of greedy (purely self-interested) agents by examining simple, mixed-motivation stochastic games. Despite the fact that naive self-play leads to inefficient outcomes in cooperative social dilemmas, we find that stochasticity in interaction partners within a population can reverse these outcomes, leading to much larger rewards, on average. This behavior is consistent across a variety of social dilemmas, and it suggests that transient (rather than stable) encounters can serve as a mechanism for eliciting prosocial behaviors in a population, even when all agents are self-interested.

1 Introduction

Multi-agent reinforcement learning (MARL) involves modeling and training autonomous agents that interact within a shared environment. In many real-world systems, agents operate independently, without access to centralized control, global reward signals, or direct communication (apart from reward signals obtained from interaction). Such decentralized settings are common in applications including autonomous vehicles navigating shared roadways [1, 2], algorithmic traders in financial markets [3, 4], distributed energy management systems [5, 6], and communication networks managed by self-interested service providers [7]. In these environments, each agent typically maximizes its own individual reward, without regard for the goals, strategies, or learning processes of others. This form of selfish optimization, where agents update their policies to improve only their personal return, presents significant challenges for achieving globally efficient outcomes.

*Please direct correspondence to A.M. (amcavoy@unc.edu).

These challenges are especially pronounced in social dilemmas, a class of multi-agent problems characterized by a conflict between individual incentives and collective welfare. In a social dilemma, agents face a choice between defection (a strategy that yields higher personal reward at the expense of others) and cooperation (a strategy that may incur short-term individual costs but produces greater overall benefit when adopted widely). Classic examples include the prisoner's dilemma, the public goods game, and the tragedy of the commons. When all agents pursue their narrow self-interest, the population often converges to Pareto-suboptimal equilibria, where mutual defection dominates despite the availability of mutually beneficial cooperative strategies.

Social dilemmas are important to study both theoretically and practically. They model a wide array of real-world challenges, such as traffic congestion, resource depletion, climate action, and public health compliance, where the lack of coordination among autonomous agents leads to inefficient or even catastrophic outcomes. Addressing these coordination failures is a fundamental problem in multi-agent systems.

To date, the predominant approaches for fostering cooperation in MARL rely on centralized mechanisms, including shared objectives, joint training procedures, engineered reward shaping, or communication protocols. While effective in controlled settings, these methods often assume access to centralized observability, joint optimization, or structured communication which are rarely available in practice. In contrast, decentralized MARL considers the more realistic setting in which agents learn independently, act without coordination, and optimize selfishly. This formulation better captures real-world conditions, yet cooperation remains difficult to achieve under such constraints.

One proposed solution is to leverage random encounters, where agents interact randomly with others drawn from the population and update their strategies based on the outcomes of those interactions. This setup introduces strategic diversity by exposing agents to a wide range of behaviors over time, without requiring persistent partners or structured coordination. However, prior work has largely concluded that random encounters alone are insufficient for sustaining cooperation. In response, studies have proposed additional mechanisms, such as partner selection, interaction opt-out, or reputation systems, to stabilize cooperative behavior. These mechanisms often rely on assumptions such as agent memory, observability of others' behavior, or control over the interaction structure, conditions that may not hold in fully decentralized systems.

This study revisits the role of random encounters in decentralized MARL and presents evidence that randomized partner interactions can, in fact, promote cooperation, even among selfish agents. Using Markov games that retain the core structure of social dilemmas it is shown that random encounters introduce population-level stochasticity that can help escape defective equilibria and discover globally optimal strategies. This finding runs counter to the prevailing view that randomness in interactions inherently drives populations toward mutual defection.

A central insight is the role of forgiving strategies in enabling cooperative dynamics. Forgiving agents respond to defection not with retaliation, but with continued cooperation, tolerating short-term exploitation in exchange for long-term benefits. While seemingly vulnerable, these strategies can act as stabilizers in population dynamics, guiding selfish agents toward cooperative equilibria by creating a path back to mutual benefit. Their spontaneous emergence highlights the importance of population diversity and raises the question of whether such strategies can be deliberately introduced into learning populations to improve outcomes.

To better understand and track these dynamics, a novel representation learning framework is introduced: the behavior space autoencoder (BSAE). This method constructs a low-dimensional

latent space that captures agent behaviors, enabling visualization, measurement, and analysis of population trajectories over the course of learning. Beyond diagnostics, this behavior space supports latent reinforcement learning, allowing policies to be directly optimized in the behavior space. This approach yields substantial improvements in sample efficiency and provides a powerful tool for steering population-level learning dynamics.

Together, these results demonstrate that cooperation can emerge in decentralized systems composed of selfish learners, even in the absence of communication, memory, or central control. Through random interactions and diverse policy landscapes, agents can self-organize into globally cooperative outcomes. This challenges long-standing assumptions in the MARL literature and opens new avenues for designing decentralized systems capable of resolving social dilemmas.

2 Related Work

Game theory and multi-agent reinforcement learning. Game theory provides mathematical tools to analyze multi-agent interactions and has been used extensively in MARL research to study equilibrium concepts like Nash Equilibria and correlated equilibria [8]. While MARL research often centers on temporally and spatially extended environments and specialized benchmarks [9] (e.g., StarCraft II, Quake III), other efforts seek more general insights into multi-agent learning in smaller-scale social dilemmas [10–12]. These connections have been further explored in works [13] establishing formal frameworks for agent interactions.

Social dilemmas. In social dilemmas [14], including the prisoner's dilemma and public goods games, individual incentives are at odds with collective welfare, leading to conflicts of interest. In repeated or Markov-game formulations, agents can learn strategies conditioned on past states and actions, leading to a wealth of possible equilibria. Classic findings show that naive self-play often leads to defection, though specialized reward shaping [15, 16], partner choices [17, 18] communication [19], or carefully tuned learning rates [**learningRateVsreward**, 20, 21] can sometimes restore cooperation [22–25].

Population-based learning. Population-based methods typically train multiple agents in parallel, often saving strategies along a training trajectory [26]. Our work is closely related but emphasizes random interaction partners each round and focuses on social dilemmas. Crucially, the goal is to lift the entire population to cooperative rewards, not just a single agent, without the use of institutions or other centralized control. The closest work to this study uses exact calculations in matrix games in conjunction with a partial differential equation to study collective learning in effectively infinite populations [27]. There, it is noted that random interactions can change outcomes relative to learning in pairs, but the primary strategy space is the two-dimensional space of reactive strategies, which cannot accommodate multi-state games and even yields a limited behavioral space within matrix games.

Population-based learning in social dilemmas. The prevailing view in the literature holds that population-based learning and random encounters are insufficient to overcome selfish behavior in social dilemmas. Previous studies have introduced additional mechanisms, such as partner

selection and the ability to opt out of interactions, as critical for enabling cooperation. For example, [28] propose a reactive partner selection mechanism. In each round, agents select a partner for a one-shot prisoner's dilemma game and update their strategies using Q-learning. The study finds that cooperation can emerge when agents learn policies for partner selection; however, they find random partner selection strategies lead to widespread defection.

A similar investigation by [29] explores a setting in which agents may opt out of interactions. Again employing Q-learning, the study shows that allowing agents to learn opt-out strategies fosters cooperation. Consistent with the findings of [28], this study also concludes that random interactions result in defective outcomes.

Collectively, prior studies suggest that random encounters tend to produce inefficient outcomes in social dilemmas. However, the findings presented in this work challenge this narrative. Evidence is provided that a more general form of population-based learning, absent mechanisms such as engineered partner selection or opt-out options, can lead to the reversal of defection and the emergence of cooperation across a wide spectrum of social dilemmas. This suggests that the lack of cooperative outcomes reported in earlier work may not be an inherent limitation of population dynamics, but rather a consequence of specific experimental design choices. These include the reliance on Q-learning, the restriction to single-shot interactions, or an insufficient number of training episodes. These results hold important implications for the design and interpretation of multi-agent learning systems.

3 Model

3.1 Markov decision processes

A Markov decision processes (MDP) is a mathematical framework for modeling sequential decision-making tasks involving a single agent operating within a probabilistic environment. Formally, an MDP can be defined as a tuple (S, A, r, P, γ), where:

- S is the *state space*, representing all possible configurations or situations of the environment. Each $s \in S$ corresponds to a specific, fully observable condition of the environment.
- A is the *action space*, which consists of all possible actions the agent can take. The available actions may depend on the current state.
- $r : S \times A \to \mathbb{R}$ is the *reward function*, which maps a state-action pair to a scalar reward value. This function quantifies the immediate benefit or cost to the agent of taking a specific action in a given state.
- $P : S \times A \to \Delta(S)$ is the *transition probability function*, where $\Delta(S)$ denotes the set of probability distributions over S. Specifically, $P(s' \mid s, a)$ represents the probability of transitioning to state s' when the agent takes action a in state s.
- $\gamma \in [0, 1]$ is the *discount factor*, which determines the present value of future rewards. A lower γ indicates a preference for short-term rewards, while a higher γ values long-term rewards more heavily.

An agent interacting with an MDP selects actions according to a strategy $\pi : S \to \Delta(A)$, which maps states to a distribution over actions. The agent's objective is to identify a strategy

that maximizes the expected cumulative discounted reward, which can be formally expressed as:

$$\mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t r\left(s_t, a_t\right) \right], \tag{1}$$

where $s_0 \in S$ is the initial state, $a_t \sim \pi\left(\cdot \mid s_t\right)$, and $s_{t+1} \sim P\left(\cdot \mid s_t, a_t\right)$ for $t \geq 0$.

3.2 Markov games

Markov games, also known as stochastic games, extend Markov Decision Processes (MDPs) to settings involving multiple agents, enabling the study of strategic interaction in shared, stochastic environments [30]. A Markov game with n agents can be described by the tuple:

$$\left(S, \{A_i\}_{i=1}^n, \{r_i\}_{i=1}^n, P, \gamma\right). \tag{2}$$

where the state space S and the discount factor γ retain the same meaning as in the MDP framework. The multi-agent extension introduces agent-specific action sets, individualized reward functions, and a joint influence on the environment dynamics:

- Each agent $i \in \{1, \ldots, n\}$ has its own action space A_i, from which it selects actions independently at each time step.
- The reward function $r_i : S \times A_1 \times \cdots \times A_n \to \mathbb{R}$ defines agent-specific rewards based on the current state and the joint actions of all agents.
- The transition function $P\left(s' \mid s, a_1, \ldots, a_n\right)$ describes how the environment evolves given the joint actions of all agents.

At each time step, agents choose their actions simultaneously, resulting in a joint action profile $(a_1, \ldots, a_n)$. The system then transitions to a new state based on P, and each agent i receives a reward r_i. Each agent aims to maximize its expected cumulative discounted reward:

$$\mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t r_i\left(s_t, a_{1,t}, \ldots, a_{n,t}\right) \right], \tag{3}$$

where $\pi = (\pi_1, \ldots, \pi_n)$ is the joint strategy profile.

In contrast to single-agent MDPs, Markov games introduce multiple agents whose simultaneous and potentially conflicting objectives transform the environment into a dynamic and strategic setting. Each agent must account not only for environmental dynamics but also for the evolving behavior of others. This multi-agent formulation enables the modeling of complex strategic interactions, such as competition, cooperation, and coordination, making it a powerful framework for analyzing decision-making in shared environments.

3.3 Social dilemmas

A social dilemma describes a scenario where individually rational decisions produce outcomes that are collectively inefficient or even harmful. More formally, it arises when the incentives of each agent align with actions that maximize personal benefit, yet when all agents act on these

incentives, the resulting joint outcome is worse for everyone than if they had coordinated differently. This creates a core tension between optimizing for self-interest and achieving outcomes that are beneficial for the group as a whole.

Social dilemmas are central to understanding cooperation and coordination challenges in multi-agent systems. They appear across a wide range of real-world domains. In environmental economics, for example, climate change mitigation presents a classic dilemma: individual actors, such as countries or corporations, may benefit from continuing to emit greenhouse gases, avoiding the short-term costs of abatement. Yet, when all actors behave this way, the long-term global consequences are catastrophic. Similarly, in public goods provision, individuals may be tempted to withhold contributions while benefiting from others' efforts, ultimately leading to underfunding or collapse of the shared resource.

In the context of autonomous systems, social dilemmas present a growing challenge. Consider self-driving vehicles navigating a four-way stop. Each car has an incentive to proceed quickly to minimize delay. If one vehicle yields while others proceed, the system functions smoothly. But if all vehicles attempt to go first, collisions or deadlock can occur, degrading traffic flow and safety for everyone. Similarly, during lane merging or highway ramp access, vehicles that aggressively merge to minimize travel time may cause stop-and-go traffic or increase the risk of accidents. In contrast, systems that incorporate occasional yielding or cooperative timing can improve overall efficiency, though these behaviors are not individually optimal in the short term.

These examples illustrate why social dilemmas are critical to study: they capture the essential tension between local decision-making and global outcomes. Designing agents that can recognize and navigate such tradeoffs is a key challenge in multi-agent learning, especially in domains where safety, fairness, and efficiency depend on sustained coordination. Understanding the mechanisms that enable cooperation to emerge and persist, such as reciprocity, reputation, or shared norms, is essential for building robust and socially aligned autonomous systems.

3.4 Simple Markov games as social dilemmas

This work focuses on a class of simple Markov games involving social dilemmas. These games are *simple* in the sense that they contain a small number of states, typically one or two, and a limited action set per agent, often consisting of two or three possible actions. Crucially, these games involve repeated interactions rather than single-shot encounters: agents interact over multiple time steps, which allows for the emergence of complex, adaptive strategies. Moreover, while we call these games *simple* the space of behaviors which they can model is far from simple. The dynamics of which are still not well understood in the literature.

Compared to the complex Markov games typically studied in MARL, simple games offer a distilled view of the core strategic tensions that define real-world social dilemmas. By abstracting away domain-specific mechanics and environmental intricacies, these minimal environments help avoid conclusions that are overly dependent on the quirks of particular settings. This mirrors the problem of overfitting in machine learning, where models may learn *shortcuts* and exploit spurious correlations, such as associating grass with the label "dog," instead of learning robust, generalizable patterns. Similarly, agents trained in complex environments can develop brittle strategies that exploit incidental features of the environment rather than learning the underlying strategic structure.

Simple Markov games mitigate the risk of overfitting to environmental intricacies by isolating the core social dilemma itself. This isolation allows for theoretical insights that generalize

more reliably across different domains. Additionally, the simplicity of these games enables exact computation of rewards and learning gradients, enhancing interpretability and providing a high degree of analytical tractability. Consequently, these environments serve as powerful tools for studying the dynamics of learning, adaptation, and strategic behavior in multi-agent systems.

The following sections describe specific social dilemmas studied, demonstrating the rich range of cooperative and competitive dynamics possible in repeated simple Markov games.

Prisoner's dilemma (PD): The prisoner's dilemma is a canonical social dilemma, modeled here as a repeated game with a single state and two possible actions: cooperate (C) or defect (D). In each round, both agents simultaneously select their actions without knowledge of the other's current choice. The resulting pair of actions determines their individual rewards, which are defined by a payoff matrix characterized by four key parameters: the reward for mutual cooperation (R), the punishment for mutual defection (P), the temptation payoff for unilateral defection (T), and the sucker's payoff for unilateral cooperation (S). A commonly employed parameterization is $(R, S, T, P) = (3, 0, 5, 1)$, which encapsulates the strategic tension central to the dilemma.

Defection strictly dominates cooperation for the individual, meaning that regardless of the other agent's action, choosing to defect yields a higher immediate reward. For example, if the other agent cooperates, defecting yields the temptation reward $T = 5$ rather than the mutual cooperation reward $R = 3$. If the other defects, defecting yields $P = 1$ instead of the sucker's reward $S = 0$. Despite this incentive to defect, mutual cooperation results in a collectively better outcome, with each agent receiving $R = 3$ compared to $P = 1$ in mutual defection. This creates a tension between self-interest and the common good: while defection is individually rational, it leads to worse outcomes for both agents when both defect.

In the repeated prisoner's dilemma, also known as the iterated prisoner's dilemma (IPD), these interactions occur over multiple time steps, allowing agents to adapt their behavior based on previous rounds. This repeated structure enables the development of complex strategies such as reciprocity, punishment, and forgiveness, which can sustain cooperation despite the temptation to defect in any single round. By conditioning actions on past behavior, agents can build trust and enforce social norms, making the repeated prisoner's dilemma a fundamental framework for studying how cooperation can emerge and be maintained in multi-agent systems facing conflicting incentives.

Two-state coin game: The original coin game, introduced by Lerer and Peysakhovich [31], is a spatially structured multi-agent Markov game designed to capture the complexities of cooperation and competition in dynamic, spatially extended environments. In this setup, two agents occupy a grid world where red and blue coins appear randomly at different locations. Each agent is assigned a color, red or blue, and receives positive rewards for collecting coins of any color. Critically, when an agent collects a coin of the opponent's color, it also imposes a penalty on the other player. This setup captures a social dilemma with a spatial component: agents must balance immediate self-interest against the longer-term consequences of their actions in a shared space.

The two-state coin game removes all the "non-rewarding" states from the original environment. In this simplified version, each state corresponds solely to whether the coin present is red or blue, and the rewards match those of the original game when each agent is positioned immediately adjacent to the coin. By eliminating sparse, non-informative states, the two-state coin game offers improved learning efficiency and yields results that are easier to interpret.

Studying this two-state coin game is vital for understanding learning dynamics in spatially extended social dilemmas, as many real-world social interactions unfold in spatially structured settings, ranging from competition over natural resources in ecological systems to coordination among autonomous vehicles in traffic networks. Examining algorithms within this streamlined yet spatially meaningful framework provides valuable insights into how spatial factors influence learning, strategy development, and the delicate balance between cooperative and competitive behaviors. Such insights are critical for designing resilient multi-agent systems capable of navigating and resolving complex spatial social dilemmas.

Nonlinear donation game: The donation game is a classical model for studying altruism and cooperation. In its original form, each agent may choose to incur a cost c to provide a benefit b to another agent. traditionally, the donation game assumes a linear relationship between the cost incurred and the benefit conferred. That is, cooperation always improves group welfare, and more cooperation is always better. However, this assumption fails to capture a range of real-world situations in which increasing levels of cooperation may produce diminishing rewards, or even net harm. For example, in public health, a moderate level of social distancing might effectively reduce disease spread with tolerable inconvenience, whereas extreme isolation may impose excessive social and economic costs that outweigh the additional benefits.

To model this non-linearity, the nonlinear donation game introduces a richer action space and a more nuanced reward structure. Instead of two actions (cooperate or defect), each agent now has three: one defection action (D) and two levels of cooperation (c_1 and c_2). These represent increasing degrees of altruistic behavior. Specifically, c_1 incurs a smaller cost and delivers a smaller benefit b_1, while c_2 incurs a larger cost and delivers a larger benefit b_2, with $c_2 > c_1 > 0$ and $b_2 > b_1 > 0$. Importantly, the game is constructed so that the net social value (i.e., total benefit minus total cost) is maximized not at the highest cooperation level, but at the moderate one: $b_1 - c_1 > b_2 - c_2 > 0$.

This non-linearity introduces a fundamentally different strategic challenge. Unlike in the prisoner's dilemma, where increasing cooperation is unambiguously better for the group, here the socially optimal outcome occurs at an intermediate level of cooperation. Full cooperation may be too costly to justify the additional benefits, while no cooperation leaves significant potential gains unrealized. As a result, the game poses a subtler learning problem: agents must not only learn to cooperate but also learn how much to cooperate. As such, the nonlinear donation game provides a compelling testbed for understanding whether learning algorithms merely default to maximal cooperation or are capable of discovering socially optimal behaviors.

Ecological prisoner's dilemma: This game extends the classic prisoner's dilemma by adding an additional state to introduce ecological feedback. The first state mirrors the standard setup, in which agents face the familiar tension between short-term self-interest and long-term group benefit. However, repeated mutual defection in this state leads to a transition into a second state that represents environmental degradation, characterized by uniformly negative rewards for all joint actions. This models the long-term consequences of over-exploitation, such as resource depletion, pollution, or ecological collapse.

The key innovation in this variant is the coupling of agent behavior with environmental dynamics: the agents' collective actions directly influence the transition between states, embedding the strategic dilemma within a broader ecological context. Unlike in the standard prisoner's dilemma, where the incentive to defect dominates, the looming threat of environmental collapse

introduces a natural incentive to maintain cooperation as a preventive measure. This shift reframes the problem as not just a question of whether agents will cooperate, but whether they can internalize the long-term consequences of their actions through learning.

Studying this ecological variant is important for several reasons. First, it provides a principled way to investigate how learning agents respond when the rewards are endogenous and evolve as a function of behavior, rather than being fixed. Second, it sheds light on whether cooperation can be sustained in the face of long-term collective risk. By comparing agent behavior in this setting to that observed in the classic prisoner's dilemma, we can examine whether the presence of a natural incentive to cooperate leads to qualitatively different learning dynamics or long-term equilibria. This has direct relevance to real-world challenges in climate strategy, sustainability, and commons governance, where ecological feedback loops are both powerful and pervasive.

3.5 Adaptation and learning

To analyze the emergence of cooperation in social dilemmas, a population of N agents is considered, engaging in repeated pairwise interactions. Agents adopt stochastic strategies that are selfishly updated via gradient ascent, facilitating the study of strategy dynamics under decentralized learning in a range of environments.

Strategy representations. Strategies are represented as *memory-one strategies* which condition behavior solely on the outcome of the previous round of interaction. Formally, let $\mathcal{A}$ denote the set of possible actions. Since each agent observes both their own and their opponent's action from the previous round, there are $|\mathcal{A}|^2$ possible joint actions to condition on. A memory-one strategy defines, for each such joint action pair $\left(a_i^{t-1}, a_j^{t-1}\right) \in \mathcal{A} \times \mathcal{A}$, a probability distribution over next actions:

$$\pi\left(a_i^t \mid a_i^{t-1}, a_j^{t-1}\right) \in \Delta\left(\mathcal{A}\right), \tag{4}$$

where $\Delta\left(\mathcal{A}\right)$ is the probability simplex over actions and t is the round of play. While memory-one strategies abstract away the full history of play, they do not limit the expressiveness of the strategy space: any strategy that can be represented as a full memory can also be represented using a memory-one strategy. Moreover, restricting attention to memory-one strategies significantly reduces the size of the state space, thereby enhancing both learning efficiency and interpretability.

We consider two types of strategy parameterizations for implementing memory-one strategies: tabular strategies and neural network strategies.

Tabular strategies represent the strategy explicitly as a lookup table over the $|\mathcal{A}|^2$ previous joint action pairs. Each entry in the table specifies a distribution over the agent's next action. This discrete and fully specified structure makes tabular strategies highly interpretable. Their interpretability makes them ideally suited for analyzing behavioral dynamics in simple, well-controlled environments. However, tabular strategies scale poorly as the action space grows or when the environment includes richer state information. Their use is thus limited to environments with small, discrete input spaces.

Neural network strategies offer a more scalable alternative. In our framework, they are still memory-one in structure: the network takes as input a representation of the previous joint action

$\left(a_i^{t-1}, a_j^{t-1}\right)$, encoded for instance as a one-hot vector, and outputs a distribution over actions:

$$\pi_\theta\left(a_i^t \mid a_i^{t-1}, a_j^{t-1}\right) = \text{softmax}\left(f_\theta\left(a_i^{t-1}, a_j^{t-1}\right)\right), \tag{5}$$

where f_θ is a neural network with parameters θ. Neural strategies are capable of generalizing across input patterns and can scale to environments with large or continuous observation spaces. However, this expressiveness comes at a cost: the function learned by the network is not easily interpretable, and small changes in parameters can induce correlated, global changes in behavior. This makes it difficult to attribute specific actions or outcomes to identifiable behavioral rules.

Gradient-based strategy updates. Agents use gradient based learning rules to update their strategies selfishly. That is, they ignore the reward of their opponent and seek only to maximize their own long-term expected reward. Let π_i and π_j denote the strategies used by agents i and j, respectively. The quality of strategy π_i when facing π_j is quantified by the expected discounted reward:

$$U\left(\pi_i, \pi_j\right) = \mathbb{E}_{\tau\sim\left(\pi_i,\pi_j\right)}\left[\sum_{t=0}^{\infty}\gamma^t r_i^t\right], \tag{6}$$

where τ is a trajectory induced by the joint strategies, r_i^t is the reward received by agent i at time t, and $\gamma \in (0,1)$ is the discount factor. The gradient ascent update rule modifies strategies in the direction of the maximal gradient:

$$\pi_i^{t+1} = \pi_i^t + \eta\nabla_{\pi_i^t}U\left(\pi_i^t, \pi_j^t\right), \tag{1}$$

where η is the learning rate. These updates are local and selfish: each agent optimizes its own reward without considering the broader population dynamics.

In simple Markov games $\nabla_{\pi_i}U\left(\pi_i, \pi_j\right)$ can be computed explicitly. In complex environments where exact gradients are intractable, agents estimate $\nabla_{\pi_i}U\left(\pi_i, \pi_j\right)$ using *policy gradient methods*. These approaches optimize a parameterized policy, $\pi : S \to \Delta\left(A_i\right)$ by maximizing the expected return,

$$U\left(\theta\right) = \mathbb{E}_{\tau\sim\pi_\theta}\left[\sum_{t=0}^{\infty}\gamma^t r^t\right]. \tag{7}$$

Using the log-derivative trick, the gradient can be expressed as:

$$\nabla_\theta U\left(\theta\right) = \mathbb{E}_\tau\left[\sum_{t=0}^{\infty}\nabla_\theta \log\pi_\theta\left(a_t \mid s_t\right)\cdot R_t\right], \tag{8}$$

where $R_t = \sum_{k=t}^{\infty}\gamma^{k-t}r^k$ is the return from time t. This formulation allows the use of sampled episodes to estimate gradients, enabling learning in environments where explicit models are unavailable. The behavior of selfish agents in social dilemmas is studied using both exact and policy gradients.

Population-level learning dynamics. This study investigates the impact of population stochasticity in social dilemmas through a framework where a population of N agents repeatedly participates in decentralized, pairwise interactions governed by the learning rules described above. The learning process unfolds over many rounds of interaction. In each round, the population is partitioned into $N/2$ pairs. Each pair plays a repeated Markov game using their current strategies at time t. For a given interaction between agent i and agent j, the outcome is evaluated using the expected discounted reward $U\left(\pi_i, \pi_j\right)$. Agent j receives a reward $U\left(\pi_j, \pi_i\right)$. These rewards are used to update each agent's policy using the gradient-based learning rules described in previous sections.

After the policies are updated, the population is re-paired for the next round of interaction. We consider two distinct pairing mechanisms, which represent qualitatively different interaction structures:

- **Fixed Pairings:** Each agent is paired with the same partner across all rounds. This regime corresponds to standard "naive self-play," where agents co-adapt in isolation.
- **Random Pairings:** Agent pairings are assigned randomly after each round. This simulates a well-mixed population in which agents interact with a broad and changing set of partners over time. Random pairing introduces population-level stochasticity and exposes agents to a greater variety of strategies, potentially enabling more robust and generalizable forms of cooperation.

A visual summary of this learning and adaptation process is provided in **??**. This population-based learning framework enables a systematic exploration of how the structure of social interaction influences the trajectory of behavioral evolution. In particular, it offers insight into how diverse encounters and patterns of social exposure affect the resolution of social dilemmas, including the emergence and stability of cooperation in the absence of coordination or centralized control.

4 Strategy Initialization

The way strategies are initialized in population-based learning sets the stage for how agents explore and grow through interaction. When agents begin with a broad range of strategies, they experience a richer variety of behaviors, leading to more effective learning. However, if the initialization is too narrow, agents become confined to a small set of behaviors, which limits the benefits of population-based learning and can make it no better than naive self-play.

It is important to understand that diversity in the parameter space alone does not guarantee meaningful diversity in behavior. Parameter space diversity refers to variation in the underlying numerical parameters that specify a strategy. For example, in neural network strategies, this corresponds to differences in the network weights; in tabular strategies, this means different probability values assigned to actions. While parameter diversity measures how "far apart" strategies are in terms of their internal representation, it does not necessarily reflect how differently those strategies behave in practice.

In contrast, behavior space diversity is concerned with the variation in the observable consequences of deploying a strategy. In multi-agent settings, this means how a strategy's actions influence the rewards it obtains when interacting with other agents. Formally, we define the

behavior of a strategy π_i with respect to a population $\mathcal{P}$ as the expected reward achieved when interacting with strategies sampled from $\mathcal{P}$:

$$V_{\text{avg}}(\pi_i, \mathcal{P}) = \mathbb{E}_{\pi_j \sim \mathcal{P}}\left[U(\pi_i, \pi_j)\right]. \tag{9}$$

This definition grounds behavior in the functional impact of a strategy within its interaction context, rather than its internal parameterization.

This distinction is crucial because many environments exhibit nonlinear reward dynamics, where uniform sampling of the parameter space often does *not* produce a uniform or meaningful coverage of behavioral space. Small parameter changes may correspond to large behavioral differences or none at all. Conversely, in linear reward environments, parameter variation more directly translates to variation in behavior and payoffs.

To demonstrate the influence of strategy initialization on parameter diversity and behavior, two complementary approaches based on autoencoders are developed: one that captures variation in strategy parameters, and another specifically designed to capture behavioral variation through reward outcomes.

4.1 Parameter space autoencoder

The parameter space autoencoder, shown in Figure 1a, is trained to compress and reconstruct high-dimensional strategy representations, optimizing for minimal reconstruction loss. Given a strategy π, the encoder E maps it to a low-dimensional latent vector $z = E(\pi)$, and the decoder D attempts to reconstruct π from z, such that the reconstruction loss

$$\mathcal{L}_{\text{rec}} = \|\pi - D(E(\pi))\|^2 \tag{10}$$

is minimized.

This setup ensures that the latent space captures the structure of the parameter space and clusters similar parameters together. Note, however, that it is agnostic to the actual behavior of strategies. Two strategies that differ significantly in parameter space may lead to indistinguishable in-game behavior. Moreover, the parameter space autoencoder only helps visualize parameter space diversity of strategies and does not capture meaningful behavioral distinctions.

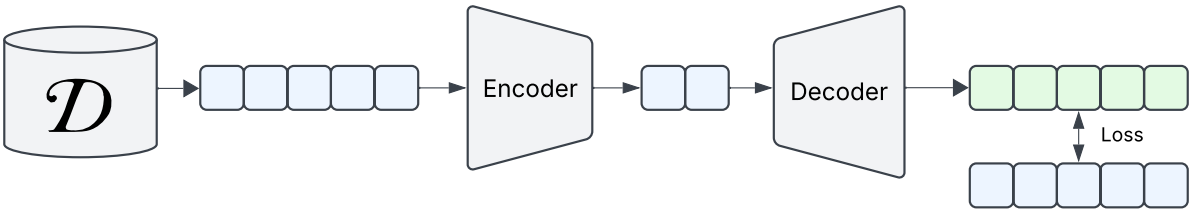
4.2 Behavior space autoencoder

To specifically capture agent behavioral, a behavior space autoencoder is introduced. This model takes pairs of strategies (π_i, π_j) as input and learns to predict their long-term expected rewards, denoted $U(\pi_i, \pi_j)$.

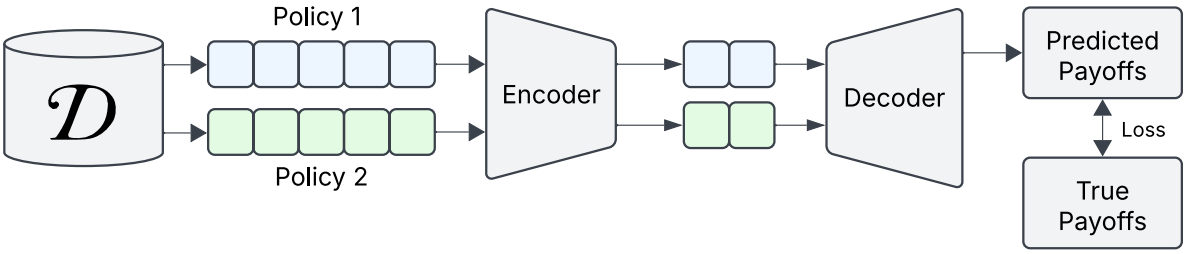
The behavior space autoencoder consists of an encoder E that maps each strategy to a latent representation $z_i = E(\pi_i)$ and $z_j = E(\pi_j)$. The decoder then takes (z_i, z_j) as input and outputs the predicted rewards to each player when these strategies interact. The loss function $\mathcal{L}_{\text{payoff}}$ is defined as:

$$\mathcal{L}_{\text{reward}} = \left|U(\pi_i, \pi_j) - \hat{U}(E(\pi_i), E(\pi_j))\right|^2. \tag{11}$$

This architecture enforces a crucial inductive bias: strategies that produce similar outcomes in terms of interaction rewards are embedded nearby in the latent space, even if their parameterizations differ drastically. In this way, the behavior space autoencoder captures behavioral equivalence or similarity and emphasizes diversity in terms of strategic effect rather than raw structure, as shown in Figure 1b.



(a) Parameter space autoencoder. This autoencoder is trained to minimize reconstruction loss between the original and decoded strategies, compressing strategies purely based on their raw parameter vectors. As a result, the latent space clusters strategies with similar parameterizations, regardless of how they behave during interactions.



(b) Behavior space autoencoder. In contrast to the parameter space autoencoder, this model embeds strategies based on how they interact in the environment. It takes pairs of strategies and predicts their expected rewards, learning a latent space where strategies with similar behavior and reward profiles are placed nearby. This behavior-aware representation captures the strategic equivalence and diversity that arise from actual gameplay outcomes, even if the underlying parameterizations differ widely.

Figure 1: Autoencoder Architectures for Strategy Embedding. These diagrams illustrate two distinct approaches to creating strategy embeddings. The parameter space autoencoder (a) focuses on compressing raw parameter vectors, while the behavior space autoencoder (b) emphasizes strategic behavior as inferred from expected interaction payoffs. Together, these models enable both structural and functional interpretations of the strategy landscape.

4.3 Visualizing parameter and behavioral diversity

To understand the impact of strategy initialization, two initialization schemes are analyzed using both a pre-trained *parameter space autoencoder* and a pre-trained *behavior space autoencoder*, which embed memory-one strategies for the prisoner's dilemma into a two-dimensional space. These embeddings facilitate interpretation of both parameter variation and behavioral diversity as reflected in the reward structure.

We consider two initialization schemes:

- **Dirichlet** (1.0): Equivalent to uniform sampling over the simplex.
- **Dirichlet** (0.1): Favors sparse strategies with probability mass near the corners $[0, 1]$.

A total of 100,000 strategies are sampled from each distribution and processed through the respective encoders. The resulting embeddings are visualized in Figure 2.

Figure 2a and Figure 2c show the embeddings produced by the traditional autoencoder. When sampling from Dirichlet (1.0), the strategies appear broadly distributed in parameter space, suggesting high surface-level diversity. In contrast, Dirichlet (0.1) generates tighter clusters in parameter space, implying that the sampled strategies seem more similar in terms of their raw parameters.

However, a contrasting picture emerges when the embeddings are examined from the perspective of actual strategic behavior. Figure 2b and Figure 2d present embeddings derived from the BSAE, corresponding to strategies sampled from Dirichlet (1.0) and Dirichlet (0.1), respectively. Notably, Figure 2d reveals that strategies sampled from Dirichlet (0.1) span a wide and comprehensive region of behavior space, including its extreme edges. This is significant, as it highlights the presence of diverse and extreme strategic profiles within the prisoner's dilemma behavior space.

Efficient coverage of this behavior space is critical because incorporating these behavioral extremes allows agents to learn from the broadest possible range of strategies. When initialized in this manner, agents can interpolate between a more extensive set of strategies, overcoming the limitations imposed by their initial parameterization. In contrast, Figure 2b shows that, despite the apparent parameter-level diversity produced by Dirichlet (1.0), many behavioral extremes remain unexplored. Even with 100,000 sampled strategies, certain crucial regions of the behavior space are left uncovered, potentially restricting an agent's ability to learn from key strategic profiles during training.

This distinction is crucial: diversity in parameter space does not necessarily equate to diversity in behavior space. While Dirichlet (1.0) spreads samples broadly across the parameter simplex, it does not fully cover the behavioral landscape and therefore misses high-impact strategies located at the edges of the behavior spectrum. Conversely, strategies sampled from Dirichlet (0.1), despite exhibiting less apparent diversity in parameter space, actually achieve broad and comprehensive coverage of behavior space. Thus, sampling from a Dirichlet prior with a low concentration parameter offers a powerful mechanism for initializing strategies that ensure rich behavioral diversity, enabling more effective exploration and learning within population-based frameworks.

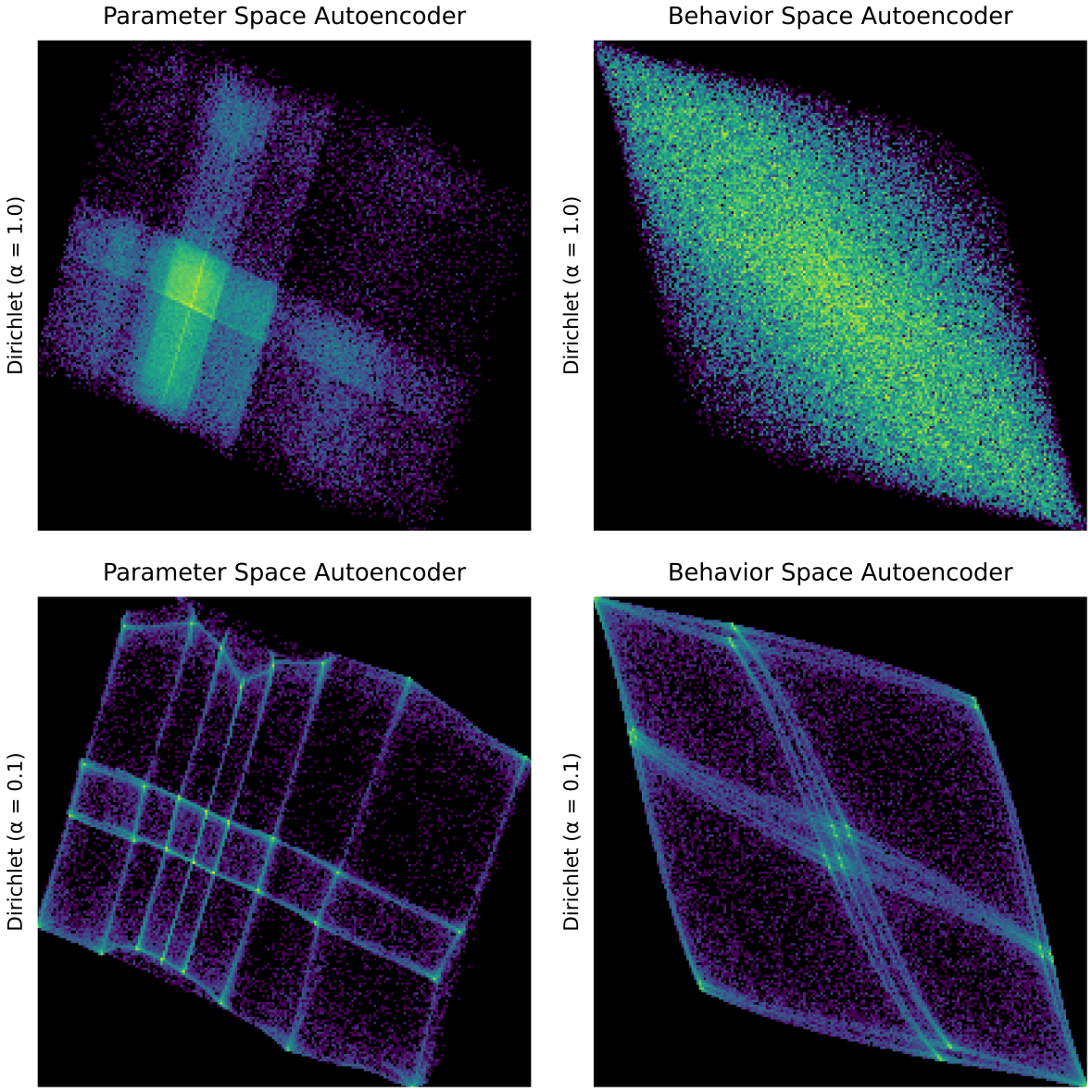


Figure 2: Visualization of 100,000 sampled strategies from two Dirichlet distributions, embedded using a traditional autoencoder (Figures a and b) and a payoff-based autoencoder (Figures c and d). **(a)** Dirichlet (1.0) with parameter space autoencoder shows widespread coverage in parameter space. **(b)** Dirichlet (1.0) with behavior space autoencoder fails to capture the extremes (corners) of the behavior distribution. **(c)** Dirichlet (0.1) with parameter space autoencoder shows tighter parameter clusters. **(d)** Dirichlet (0.1) with behavior space autoencoder reveals broad behavioral coverage, particularly at the extremes of the strategy space. These results illustrate that parameter diversity does not necessarily translate to behavioral diversity and motivate the use of sparse Dirichlet priors to ensure exposure to qualitatively distinct strategic behaviors.

5 Experiments

5.1 Implementation details

The proposed population learning framework was evaluated across the four social dilemmas defined in Section 3.2. These games differ in their action spaces and payoff structures, and involve repeated interactions. After each round, agents updated their strategies using either exact or approximate policy gradients.

In the exact gradient setting, long-term expected rewards were computed precisely using full knowledge of the environment dynamics or exact value computation. This allowed for a higher learning rate of 0.1, as the precise gradient information reduced the risk of destabilizing updates. In contrast, approximate gradients were computed using vanilla policy gradient with rollouts of 250 time steps. Due to the increased variance in these estimates, a smaller learning rate of 0.05 was used to ensure stability. All updates were performed using stochastic gradient descent (SGD), which is better suited for non-stationary multi-agent environments. SGD provides more stable dynamics compared to adaptive optimizers like Adam, which may overfit to rapidly changing local gradients.

Both tabular and neural network (NN) strategies were implemented. In the tabular setting, strategies were stored as explicit state-action mappings, with each value updated independently. For neural strategies, each was represented by a feedforward neural network with a single hidden layer. The hidden layer had a width equal to four times the size of the state-space. Initial strategies were sampled independently for each agent using a Dirichlet distribution over the action simplex. This initialization method enabled controlled exploration of initial diversity.

Performance was assessed using the average total population reward. This metric serves as a meaningful proxy for social efficiency and cooperation: higher average rewards indicate not only successful individual behavior but also emergent alignment among agents. Importantly, this metric captures more than just full cooperation, especially in games where blindly maximizing cooperation can be suboptimal, making it a robust measure of overall population-level effectiveness. The learning framework was evaluated against a baseline of naïve self-play, in which agents interact with fixed opponents that do not change after each round. All results are reported as averages over 10 random seeds.

5.2 Exact gradients

Figure 3 summarizes the learning dynamics of populations of size $N = 50$, trained using exact gradient updates and tabular strategies, across several distinct Markov games. Each row in the figure corresponds to a different social dilemma, while the two columns compare two experimental conditions: (1) fixed pairings, where agents are matched with the same partner across rounds of the repeated game, and (2) random pairings, where partners are reshuffled after each round. The y-axis reports the average population reward over time (x-axis), serving as a proxy for the overall social welfare.

Across all games, a consistent and robust pattern emerges: random pairings systematically outperform fixed pairings in terms of long-run average population reward. This result is particularly striking because it holds across a diverse set of social dilemmas, each with its own strategic structure and payoff landscape. The presence of partner randomization significantly improves

learning outcomes, enabling agents to escape from local equilibria associated with mutual defection and to discover globally superior strategies.

In the standard repeated prisoner's dilemma, agents paired with the same partner learn to defect almost immediately. This behavior leads to low and stagnant population rewards. In contrast, when partners are randomized after each round, agents initially follow a similar trajectory toward defection, but then exhibit a strong and sustained reversal. Over time, the population converges toward strategies that support cooperation, leading to a markedly higher average reward. This rebound suggests that exposure to a broader variety of partner strategies encourages the evolution of more generalizable, cooperative strategies.

Similar trends are observed in the coin game and the nonlinear donation game. Under fixed pairings, agents again fall into early defection traps, never recovering to cooperative norms. However, random pairings lead to recovery and eventual convergence toward higher-reward outcomes. The nonlinear donation game is particularly illustrative, as it features a non-monotonic relationship between cooperation and global welfare. Unlike simpler dilemmas where maximum cooperation aligns with the global optimum, here it is an intermediate level of cooperation that yields the highest population reward. Remarkably, the population with randomized partners does not merely maximize cooperation blindly; rather, it converges to this more nuanced global optimum. This finding underscores the capacity of diverse interactions to enable agents to learn not just pro-social behavior, but strategically optimal cooperation.

The ecological variant of the prisoner's dilemma introduces a modified incentive structure in which cooperation is more naturally rewarded. In this case, both fixed and random pairings result in an increase in cooperative behavior. Agents in fixed pairs gradually learn to cooperate roughly two-thirds of the time, resulting in an average population reward just above –1, a significant improvement over the mutual defection outcome of –5. Nevertheless, even in this more favorable environment, fixed pairs plateau below the optimal outcome. In contrast, random pairings drive the population to full cooperation, achieving the global maximum. This again highlights the role of interaction diversity in promoting socially efficient outcomes, even in games where cooperation is already partially incentivized.

Overall, these findings demonstrate that the structure of partner interaction significantly shapes the emergent strategies in multi-agent learning. Fixed pairings promote myopic, partner-specific strategies that are prone to early convergence on suboptimal equilibria. In contrast, randomization introduces variability and strategic uncertainty, which serves as a form of regularization or exploration, driving the system toward more generalizable and socially optimal solutions.

These results connect between MARL and foundational insights from evolutionary game theory, where mechanisms that promote social mixing are known to support the emergence of cooperation. Our findings provide both algorithmic and empirical reinforcement of these theoretical principles in the context of modern learning agents. By showing that randomized partner interactions consistently lead to more favorable collective outcomes, this work bridges the gap between evolutionary models of cooperation and contemporary MARL frameworks. From a design standpoint, the results suggest that artificial systems composed of interacting agents should incorporate dynamic and heterogeneous partner interactions to avoid pathological convergence to selfish or suboptimal equilibria. Partner randomization and structural diversity act as an implicit curriculum, encouraging broader exploration and generalization.

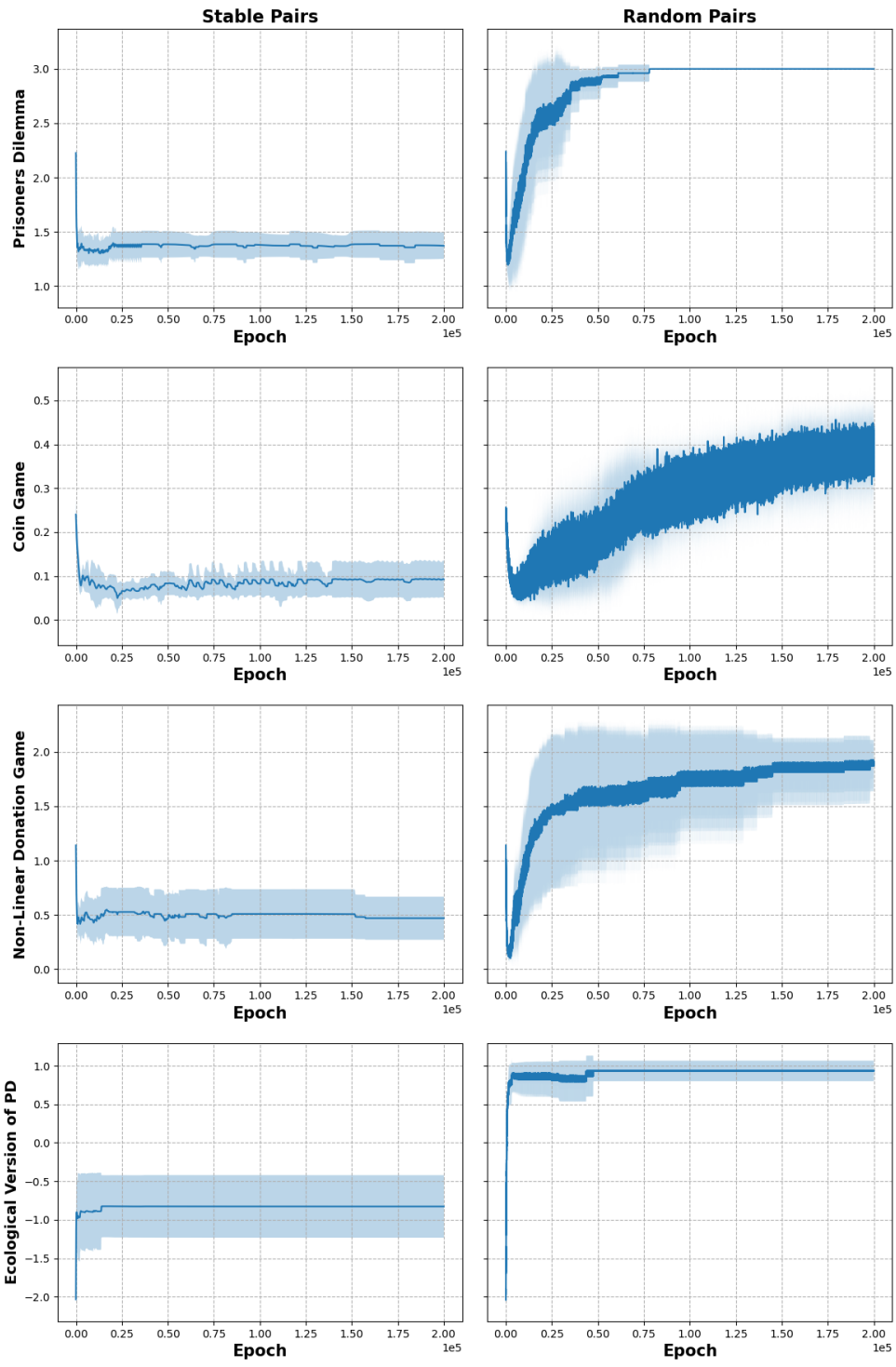


Figure 3: Impact of Interaction Stochasticity with Exact Gradients. A population of 50 agents with tabular, parameterized strategies was initialized using a Dirichlet $(0.75, 0.75)$ distribution. We compare the performance of stable pairings, where opponents remain fixed, with random pairings, where opponents are randomized each round. This figure demonstrates that fixed interactions often lead to a rapid convergence to a suboptimal equilibrium, while stochastic interactions, characterized by randomly changing opponents, can reverse these tendencies.

5.3 Approximate gradients

While the previous section examined learning dynamics under exact gradient updates, such methods are rarely practical due to their reliance on complete knowledge of the environment dynamics and reward functions. To address this limitation, we now consider agents that learn via approximate gradient methods, focusing on policy gradient algorithms [32].

Figure 4 presents the average population reward over time for a population of size $N = 50$ trained with policy gradient updates using tabular strategy representations. As in the exact gradient setting, each row corresponds to a different social dilemma and the columns compare learning under fixed and random pairings.

Despite the noise introduced by policy gradient estimation, we observe a surprisingly consistent replication of the dynamics observed under exact gradients, albeit over a longer learning horizon. In all games studied, populations with random pairings consistently achieve higher long-term rewards than those with fixed partners. Agents with fixed partners tend to converge toward low-reward equilibria, characterized by mutual defection or locally optimal but globally suboptimal cooperative strategies. In contrast, randomized interactions allow agents to escape these traps and discover globally optimal strategy profiles.

While policy gradients yield an unbiased estimate of the exact gradient, the replication of cooperative dynamics in this setting is non-trivial. In multi-agent environments, the learning landscape is inherently non-stationary: each agent's strategy update alters the environment observed by others. Under these conditions, local gradient estimation errors can accumulate and interact in complex ways, leading to drastically different strategy pairings and optimization trajectories. Such instabilities could, in principle, derail the previous findings entirely. The fact that partner randomization still leads to cooperative behavior, even under these noisy conditions, demonstrates that randomized interactions create a *powerful signal* encouraging independent selfish agents to cooperate. And this powerful signal survives even under noisy updates and more realistic environmental conditions.

5.4 NN parameterization

Tabular strategies offer a high degree of interpretability: each parameter directly corresponds to an action probability, allowing for transparent analysis of population dynamics and behavioral diversity. NNs, by contrast, introduce an abstract parameterization that enables scalability to high-dimensional and complex state spaces, which is critical for many real-world MARL problems, but at the cost of reduced interpretability and increased training complexity.

In the *tabular setting*, strategies are initialized by sampling directly from a Dirichlet distribution over the action simplex. This allows for fine control over initial diversity: lower concentration parameters yield more extreme (sparse) distributions, while higher values produce more uniform mixtures. In the *NN setting*, we pre-train each network to match a tabular policy sampled from the same Dirichlet distribution, ensuring that the initial output policy distribution (rather than the network weights) conforms to the same statistical structure.

Despite exhibiting similar behavior at initialization, neural networks (NNs) are substantially more sensitive to the concentration of the Dirichlet prior. Figure 5 illustrates this effect in the repeated prisoner's dilemma under random pairings. Populations initialized with low-concentration priors (e.g., Dirichlet $(0.5, 0.5)$) reliably recover cooperative behavior in populations of size 50. In contrast, populations initialized with more uniform priors (e.g., Dirichlet $(1.0, 1.0)$)

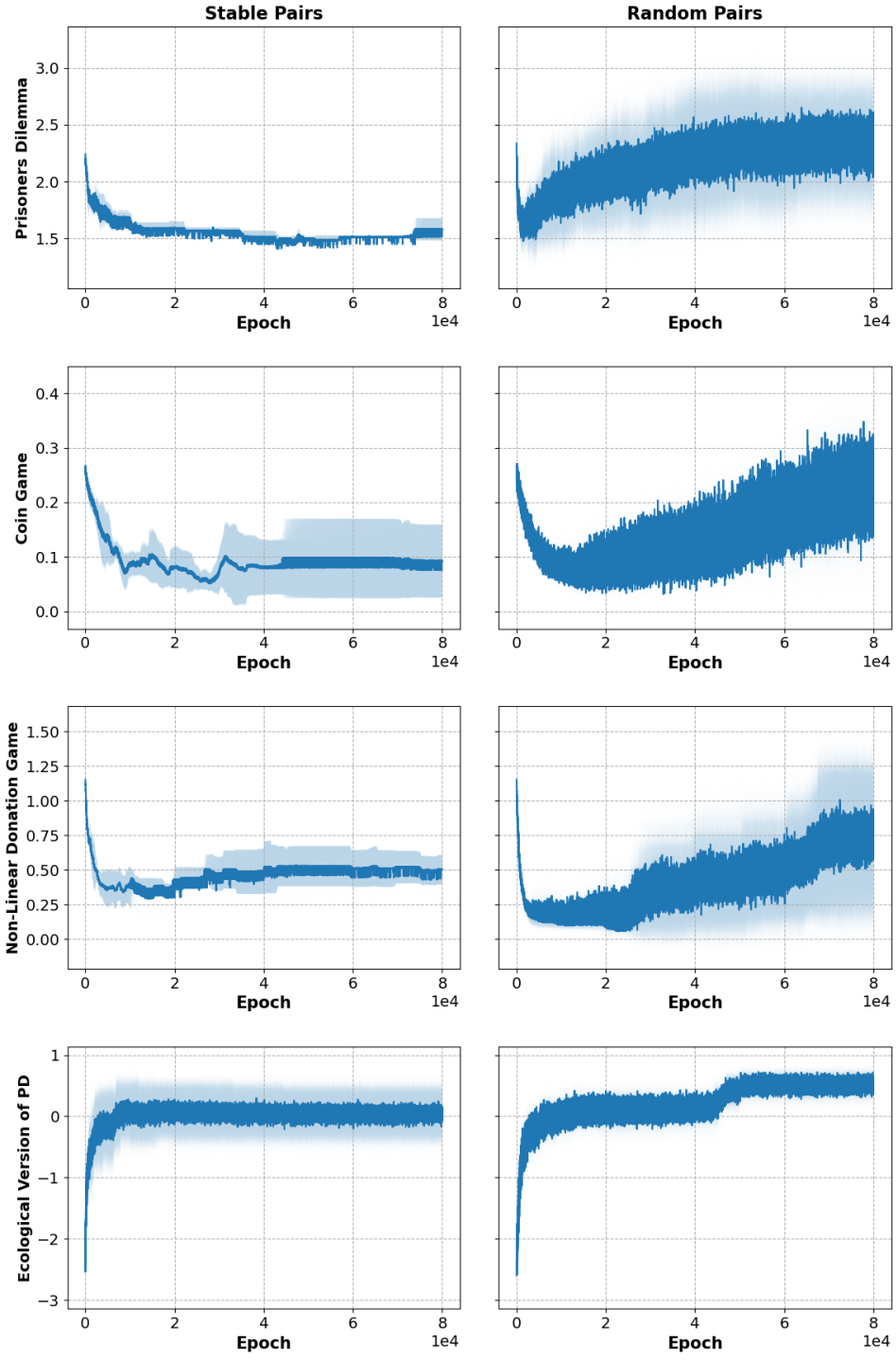


Figure 4: Impact of Interaction Stochasticity with Policy Gradients. A population of 50 agents with tabular, parameterized strategies was initialized using a Dirichlet $(0.75, 0.75)$ distribution. We compare the performance of fixed pairings, where opponents remain fixed, with random pairings, where opponents are randomized each round. This figure illustrates that, even with the additional noise introduced by approximate policy gradient updates, random pairings can reverse defective outcomes. Despite the stochastic nature of strategy updates, population dynamics effectively guide agents toward cooperative equilibria.

frequently fail to sustain cooperation. We focus here on the random pairings setting, as the fixed pairings condition typically leads to rapid defection regardless of initialization, offering limited insight. This highlights the critical role of *initial behavioral diversity* in stabilizing outcomes, even under abstract function approximators like NNs.

This heightened sensitivity in neural networks (NNs) arises from the structural complexity of their parameter space. Unlike tabular strategies, which directly specify action probabilities, NNs encode these behaviors indirectly through a non-linear and high-dimensional weight vector θ. This added abstraction introduces a disconnect between the parameters being optimized and the resulting behavioral strategies, complicating the learning dynamics.

To formalize and visualize the disconnect between parameter updates and strategic behavior, the following distance metrics are defined:

- **NN parameter distance**: the normalized L^2 distance between the weight vectors θ of two neural network strategies.
- **Strategy (tabular) parameter distance**: the normalized L^2 distance between the action probability distributions defined by two strategies.
- **Behavioral distance**:

$$D_{\text{behavior}}\left(\pi_i, \pi_k\right) = \left|V_{\text{avg}}\left(\pi_i, P\right) - V_{\text{avg}}\left(\pi_k, P\right)\right|, \tag{12}$$

  where V_{avg} denotes the average performance of a strategy π against a fixed population P (see Section 4).

Since neural networks ultimately define tabular policies through their outputs, both the distance between internal weight vectors (NN parameter distance) and the distance between resulting action distributions (strategy parameter distance) can be measured. This dual representation enables analysis of how differences in NN parameters translate into behavioral differences.

To investigate this relationship, 2,000 strategies were sampled randomly and the above distance metrics were computed for further analysis. **??**a shows the relationship between NN parameter distance and the corresponding strategy parameter distance, while **??**b compares strategy parameter distance with behavioral distance across both representations.

??a reveals a weak correlation between NN parameter distance and the corresponding tabular policy distance, highlighting the entangled nature of NN parameters. **??**b further shows that tabular strategies maintain a stronger correlation between parameter and behavioral distances, whereas NN strategies exhibit little to no such alignment. This indicates that even minor changes in NN weights can result in large and unpredictable shifts in behavior.

This instability presents a key challenge for preserving behavioral diversity during learning. In NN-based systems, even a single training iteration can be sufficient to eliminate initial variation. The challenge is amplified by the overparameterized and non-linear structure of the NN architecture. However, initializing neural strategies with low-concentration Dirichlet priors (e.g., Dirichlet$(0.5, 0.5)$) promotes greater dispersion in parameter space, increasing the chance that agents converge to distinct local optima. Although the behavioral space is not directly controlled, this increased parameter-space variability can indirectly support a broader range of behaviors, helping to sustain diversity that would otherwise collapse early in training.

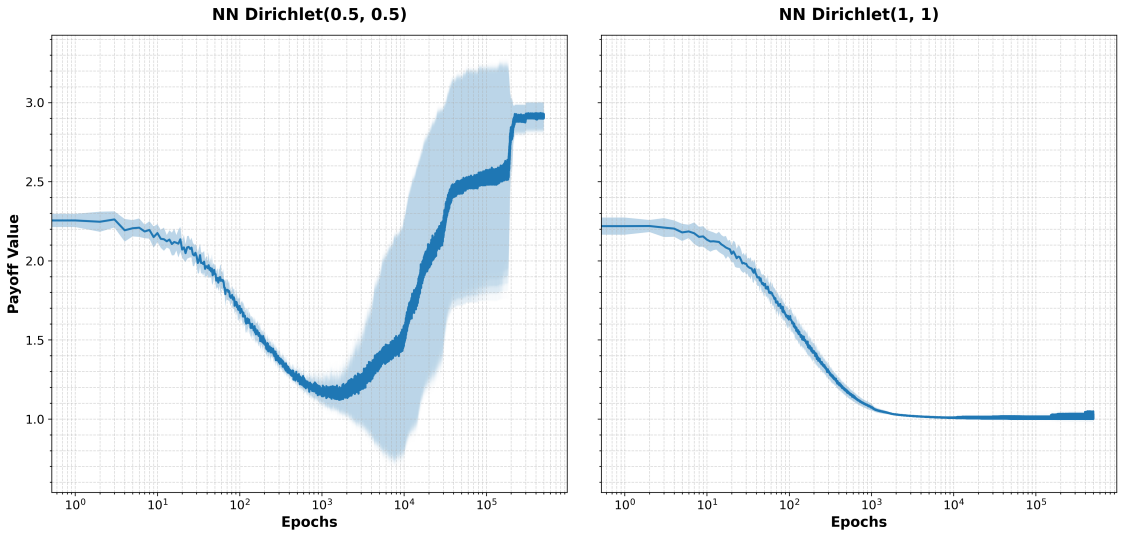


Figure 5: Parameterizing strategies with Neural Networks: A population of 50 agents is parameterized with neural network strategies using both Dirichlet $(0.5, 0.5)$ and Dirichlet $(1.0, 1.0)$. Exact gradients were used to update strategies. Cooperation is consistently achieved with a Dirichlet $(0.5, 0.5)$ initialization, while it is unlikely with a Dirichlet $(1.0, 1.0)$ initialization. Neural networks introduce complexities to the learning process, but these results indicate that cooperation remains attainable when parameters are appropriately tuned.

This sensitivity can be mitigated however, by using natural gradients. In continuous time, the natural gradient flow [33] uses the update $\frac{d\pi}{dt} = \nabla_\pi J(\pi)$, which corresponds to the learning dynamics used in the tabular setting. On the other hand, the naïve gradient flow is $\frac{d\theta}{dt} = \nabla_\theta J\left(\pi^\theta\right)$. Since $\frac{d\pi}{dt} = \frac{d\pi}{d\theta}\frac{d\theta}{dt}$ and $\nabla_\theta J\left(\pi^\theta\right) = \left(\frac{d\pi}{d\theta}\right)^\intercal \nabla_\pi J(\pi)$, the natural gradient gives the equation

$$\frac{d\theta}{dt} = \left(\left(\frac{d\pi}{d\theta}\right)^\intercal \frac{d\pi}{d\theta}\right)^{-1} \nabla_\theta J\left(\pi^\theta\right). \tag{13}$$

These learning dynamics account for the curvature of the parameterization via pulling back the metric tensor on the underlying Euclidean space.

5.5 Efficient optimization

Random pairings have been shown to mitigate defection, even under exact and approximate selfish gradient updates. While these results have been validated in relatively simple Markov games, a natural question arises as to whether such findings scale to more complex, high-dimensional environments. To address this challenge, *Latent Reinforcement Learning* (LRL) is proposed, a novel framework enabling strategy optimization within a learned, low-dimensional latent embedding space.

Latent optimization techniques have been extensively applied in domains such as computer vision and natural language processing to facilitate more efficient learning by focusing on semantically meaningful, compressed representations. However, their application to multi-agent reinforcement learning, and specifically to strategy optimization in game-theoretic settings, is a significant advancement. The inherent non-stationarity and strategic interdependence of multi-

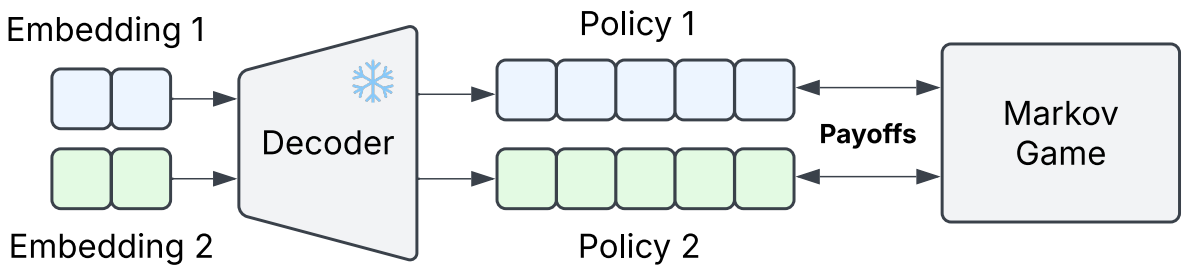


Figure 6: Latent Reinforcement Learning (LRL). LRL comprises two phases: (1) pre-training an autoencoder to learn compact strategy representations, using either parameter reconstruction or behavioral similarity, and (2) optimizing latent strategy embeddings directly. The encoder is discarded after initialization, while a frozen decoder maps latent embeddings back into full strategies deployed in the environment. Payoff gradients from interactions are backpropagated through the decoder, enabling stable and efficient optimization within the latent space.

agent environments make direct parameter optimization noisy and sample-inefficient. LRL provides a promising approach by leveraging a structured latent space.

LRL operates in two stages. First, an autoencoder is trained to learn a compact and informative representation of tabular strategies by minimizing reconstruction error. This yields an encoder E that maps strategies into a continuous latent space $\mathbb{R}^d$. Latent strategies are initialized by sampling from a Dirichlet distribution and encoding these samples:

$$z_0 \sim E\left(\text{Dirichlet}\left(\alpha\right)\right), \quad z_0 \in \mathbb{R}^d, \tag{14}$$

where z_0 denotes the initial latent embedding of the strategy. During the second stage, optimization proceeds directly in the latent space using gradient ascent updates:

$$z_{t+1} = z_t + \eta \nabla_{z_t} J\left(z_t\right), \tag{15}$$

where $J\left(z_t\right)$ represents the objective function evaluated on the decoded strategy corresponding to z_t. The decoder remains frozen during optimization, ensuring updates remain within the manifold of plausible strategies. An overview of the framework is illustrated in Figure 6.

A novel and compelling advantage emerges from integrating LRL with the BSAE, a combination not previously investigated. This integration combines the benefits of compact latent space optimization with the unique ability to optimize directly in behavior space, resulting in exceptional training efficiency. Beyond its immediate applications, this dual optimization framework has the potential to transform a wide range of RL problems. Existing approaches focus on parameter-space optimization, despite the fact that what truly matters is the resulting behavior. Since indirect optimization of behavior via parameters is often inefficient, this approach could open new pathways to faster, more interpretable, and more effective learning.

Results shown in Figure 7 demonstrate that latent optimization substantially improves sample efficiency, using the same parameters and x-axis scale as Figure 3 for a fair comparison. The autoencoder effectively filters out high-frequency strategy details; for example, strategies differing only in cooperation probability by small fractions such as 0.99 versus 0.9999 are nearly indistinguishable in latent space. By abstracting away such fine-grained details, LRL guides learning toward behaviorally meaningful changes, reducing noise in gradient estimates and accelerating convergence.

The latent space functions as a form of structural regularization, much like how population learning distributes learning pressure across diverse interactions. In LRL, this regularization arises from dimensionality reduction, which restricts updates to a subspace capturing only the most salient strategic features. Beyond improving sample efficiency, LRL provides a principled mechanism for retaining the expressive capacity of neural networks while mitigating the adverse effects of parameter sensitivity highlighted earlier. In high-dimensional NN weight spaces, small perturbations can trigger abrupt, often erratic behavioral shifts, posing significant challenges in non-stationary, multi-agent environments characterized by sharp local discontinuities. In contrast, latent optimization constrains learning to a smoother, low-dimensional manifold that filters out noisy or redundant weight-level variation. This yields more behaviorally coherent updates and stabilizes training dynamics.

6 Ablations and visualizations

6.1 Population size

To understand how population structure influences the emergence of cooperation, an ablation study was conducted varying the population size from 2 to 50 in increments of 2. For each population size, agent strategies were initialized by sampling from a Dirichlet distribution with a concentration parameter of 0.75. This distribution ensures a moderate level of initial diversity, preventing the population from starting in either highly uniform or overly chaotic configurations. Each simulation was run until convergence, and results were averaged over 25 independent trials. The average population reward is reported in Figure 8.

The results reveal that while small populations typically converge to defect-dominated outcomes, populations as small as 10 begin to exhibit consistent convergence toward cooperative equilibria. This trend becomes more robust as population size increases, with larger populations reliably achieving high final rewards. Interestingly, the timescale of convergence remains largely unaffected by population size. That is, while larger populations are more likely to discover cooperative strategies, they do not require more time to do so. These results have significant practical implications suggesting that random interactions within a modestly sized and diverse group are sufficient to reverse defection; unrealistically large populations are not required.

6.2 Diversity

There are two primary levers for increasing strategy coverage in a population: expanding population size and modifying the diversity of initial strategy distributions. While larger populations naturally span a broader region of the strategy space, a more direct and controllable method is to adjust the Dirichlet concentration parameter used for initialization. In the previous section, the effect of population size on cooperative convergence was explored. Here, the focus shifts to how varying the Dirichlet concentration influences learning outcomes. To this end, a population of 50 agents was initialized using concentration values in $\{0.25, 0.5, 0.75, 1.0, 1.25\}$, and learning trajectories were examined across repeated trials and multiple games until convergence.

As shown in Figure 9, cooperative outcomes emerge reliably across a wide range of concentration values. Interestingly, the most diverse setting (concentration = 0.25) produces the worst performance, with final rewards significantly lower than in less diverse settings.

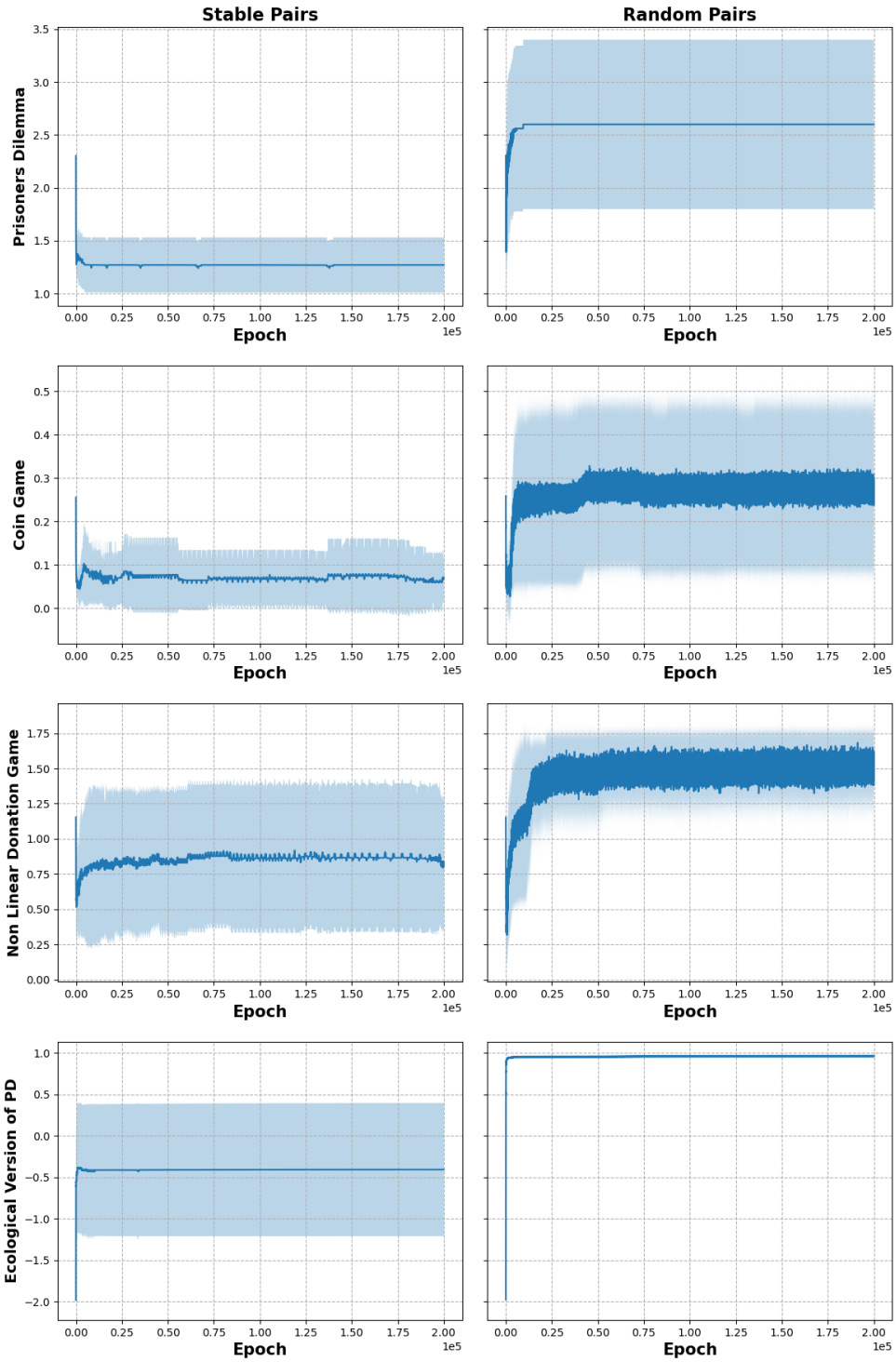


Figure 7: Impact of Latent Optimization on Sample Efficiency. Using the same parameters as in Figure 3, optimization is performed in the reduced latent space to enable a fair comparison. Latent optimization greatly enhances sample efficiency, with random pairings leading to rapid convergence toward cooperative outcomes, while fixed pairings remain prone to defection.

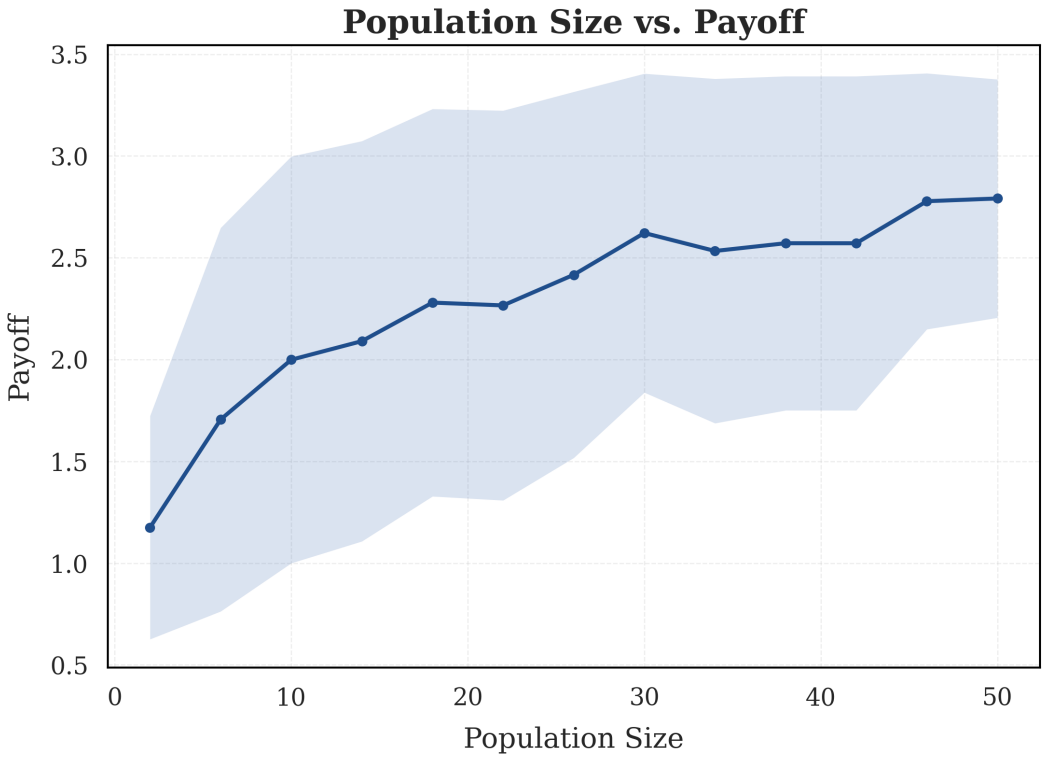


Figure 8: Influence of Population Size on Convergence Rewards: We investigate the impact of varying population sizes on the dynamics of cooperation. Each player in the population was initialized with strategies drawn from a Dirichlet $(0.75, 0.75)$ distribution, and the population learning framework was run for up to 500,000 interactions or until convergence. Exact gradients were used to update strategies. The results show that larger populations support more robust convergence to cooperative equilibria.

To understand why excessive diversity can hinder learning in the IPD, consider how π_i would update their strategy under each of the following conditions:

1. $\pi_i \approx 0, \pi_j \approx 0$ (Mutual Cooperation):
   Mutual cooperation is stable and beneficial, leaving no incentive to change; gradients are near zero.

2. $\pi_i \approx 0, \pi_j \approx 1$ (Sucker's Payoff):
   The cooperator receives the lowest possible payoff ($S = 0$), but the gradient is also near zero, providing no learning signal.

3. $\pi_i \approx 1, \pi_j \approx 0$ (Temptation):
   The defector is highly rewarded and thus sees no reason to change, reinforcing selfish behavior.

4. $\pi_i \approx 1, \pi_j \approx 1$ (Mutual Defection):
   Although suboptimal, mutual defection is stable and offers no gradient incentive to cooperate.

In each of these cases, gradients either reinforce the same behavior or vanish. High initial diversity increases the likelihood that many agent pairs begin in one of these unproductive regions of the strategy space. As a result, learning either stalls or exhibits unstable, oscillatory behavior. This limitation is directly tied to the reward structure of the IPD, particularly the zero-valued sucker's payoff, which fails to provide a gradient for improvement in crucial scenarios. If the payoff matrix were altered (e.g., $S > 0$), these effects might be mitigated.

Another observation from Figure 9 is that as the Dirichlet concentration parameter increases, reducing diversity, so too does the slope in the optimization trajectory. This indicates that with higher initial concentrations, strategy behavior changes more readily. This pattern is consistent with the nature of the initialized strategies: lower concentrations generate more extreme "confident" agents (near 0 or 1) that are difficult to shift through learning. In contrast, higher concentrations produce more moderate "undecided" agents with strategies closer to the midpoint. These undecided agents are more sensitive to learning signals and require fewer reinforcing updates to shift toward a confident strategy, resulting in faster adaptation and a steeper optimization slope.

While diversity is generally beneficial, its effectiveness is highly dependent on the structure of the environment. In poorly defined or weakly informative environments, such as those where certain payoffs fail to produce meaningful learning signals, excessive initial diversity can lead agents into unproductive regions of the strategy space. This, in turn, can hinder effective learning and destabilize collective dynamics, ultimately impeding convergence toward optimal outcomes.

6.3 Visualization

To gain deeper insight into how population dynamics naturally foster cooperation in social dilemmas, the IPD is simulated over 100,000 iterations with a population of 50 agents. Agent strategies are recorded every 100 epochs for subsequent analysis. In the IPD, memory-one strategies are represented as five-dimensional vectors $(p_0, p_{CC}, p_{CD}, p_{DC}, p_{DD})$, where each component corresponds to the probability of cooperating given the outcome of the previous round. To facilitate interpretation, these high-dimensional vectors are projected into a two-dimensional space.

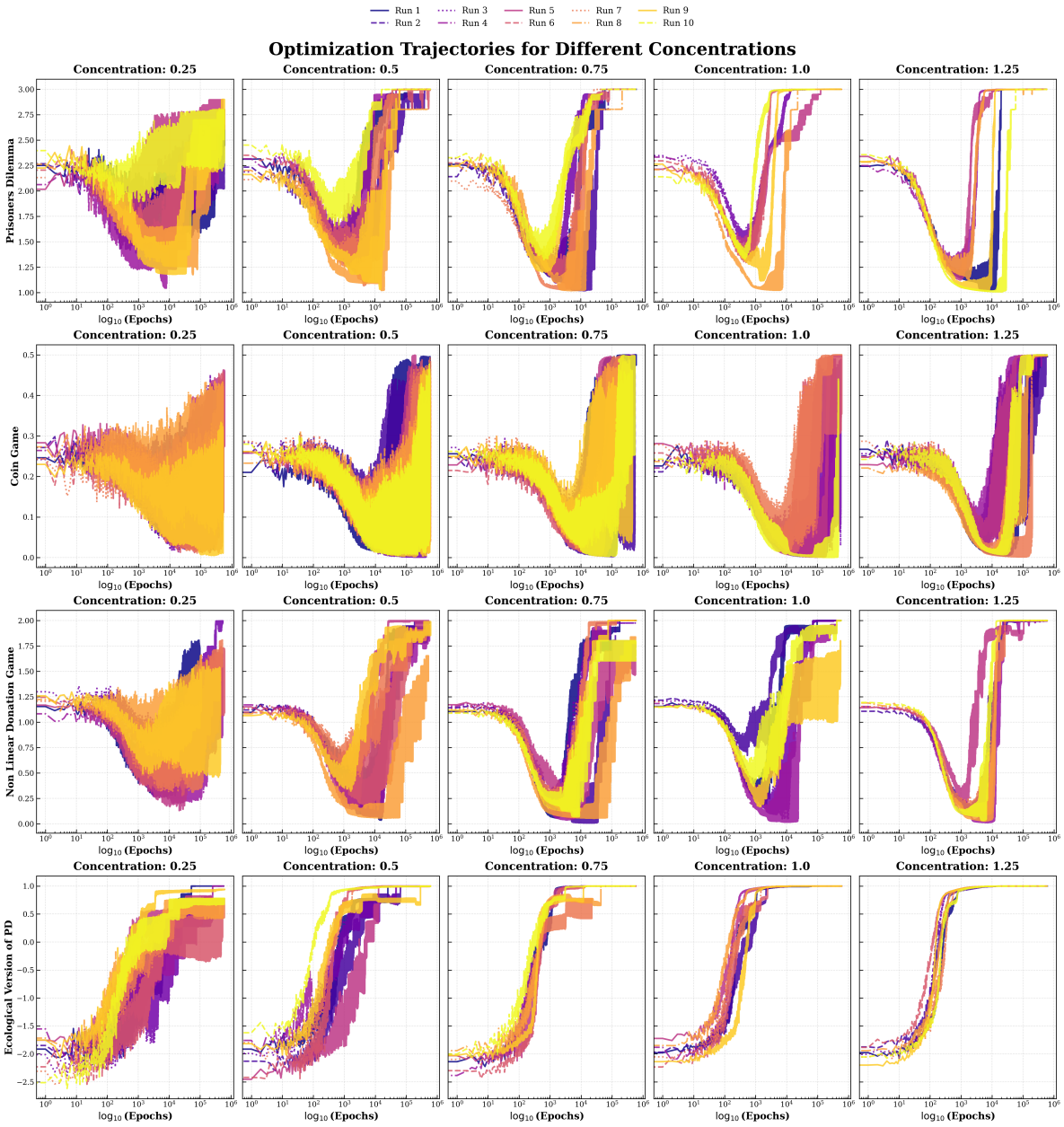


Figure 9: Effect of Initial Population Diversity on Equilibrium Rewards: We initialize a population of 50 agents with tabular, parameterized strategies, updating them using exact gradients. This figure demonstrates how the initial diversity of the population, controlled through the concentration parameters of the Dirichlet distribution, influences convergence rewards. Initial population diversity can have a big impact on the optimization trajectory.

While standard dimensionality reduction techniques such as Principal Component Analysis (PCA) and t-distributed Stochastic Neighbor Embedding (t-SNE) are commonly used for such tasks, they have notable limitations in this context. These methods typically rely on fixed, predefined bases derived from the data distribution at a single point in time, such as the initial population. However, as strategies adapt and evolve, these projections may lose interpretability. Furthermore, these methods operate in parameter space and are agnostic to how strategies behave in practice; they do not capture how differences in parameters translate into differences in outcomes.

To address these challenges, the behavior space autoencoder, introduced in previous sections, is used to project policies into a behaviorally meaningful two-dimensional space. Recall that this model is trained using pairs of strategies sampled from a Dirichlet $(0.5, 0.5)$ distribution, ensuring diverse and representative coverage of the strategy space. Each strategy is encoded into a low-dimensional latent vector, and the decoder predicts the expected payoffs when the two strategies interact in the IPD. The model is optimized to minimize the discrepancy between these predicted rewards and the true game-theoretic payoffs. This training objective ensures that the latent space reflects behavioral equivalence: strategies that perform similarly, regardless of their parameterization, are embedded near each other. Crucially, these embeddings remain meaningful under any distribution of population strategies.

This approach constitutes a novel contribution to the analysis of evolving strategic behavior. By grounding the projection space in observed behavioral consequences rather than structural similarity, the behavior space autoencoder provides a consistent and interpretable framework for understanding complex dynamics in policy evolution. Unlike static techniques, it retains interpretability across time, even as the population distribution shifts dramatically. Beyond the specific context of the IPD, this methodology offers a general and flexible tool for interpretability in dynamic multi-agent systems. The same principles can be applied to a wide variety of domains, including reinforcement learning, evolutionary games, policy space exploration, and real-world multi-agent coordination problems. By providing a way to visualize and analyze agent behavior, the behavior space autoencoder opens new pathways for understanding and guiding emergent behavior in complex adaptive systems.

The trained encoder is leveraged to visualize the evolving landscape of strategies throughout the simulation. A complete visualization of the full evolutionary trajectory is provided in the supplementary material, while a selection of representative snapshots is presented in **??**. In these figures, each agent's strategy is depicted as a blue point embedded in the learned two-dimensional behavior space. To provide behavioral context and aid interpretation, canonical reference strategies, such as Tit-for-Tat (TFT), Always Cooperate (ALLC), Always Defect (ALLD), and Generous Tit-for-Tat (GTFT), are also plotted. These strategies serve as well-known behavioral archetypes: TFT initiates with cooperation and then reciprocates the opponent's previous action, promoting mutual cooperation; ALLC cooperates unconditionally, leaving it vulnerable to exploitation; ALLD defects unconditionally, representing a purely selfish approach; and GTFT modifies TFT by occasionally forgiving defections, thus helping sustain cooperation even in noisy or error-prone environments.

The visualization reveals that the initial population is highly diverse, exploring a broad region of the strategy space. However, within the first few hundred epochs, the population dynamics lead the distribution to drift toward ALLD and cluster within a region characterized by extortionate strategies. These extortionate strategies consistently punish defectors but also defect with

some probability against cooperators, exploiting them for gain. Crucially, these strategies exhibit negative reinforcement: when two extortionate strategies interact, the selfish-learning gradients push both toward increasingly defecting behaviors, ultimately converging on ALLD. A similar pattern emerges when extortionate strategies are paired with TFT; despite TFT's reciprocity, both agents tend to evolve toward defection in response to exploitation.

Escaping these defectionary feedback loops requires sufficient population diversity. In particular, forgiving strategies, characterized by high probabilities of cooperating following an opponent's defection (high p_{CD}), play a pivotal role in redirecting the learning dynamics of extortionate strategies toward more cooperative outcomes. While these forgiving strategies are vulnerable to exploitation in the short term, they are rewarded over a longer horizon as they serve to shift the population toward cooperative dynamics. Importantly, with enough diversity, these forgiving strategies naturally emerge through random interactions.

When player pairings are fixed, agents learn exclusively from repeated interactions with the same partner, limiting their exposure to diverse behaviors. If both agents fall into a mutual defection pattern, there are no external influences to break the cycle, making full defection (ALLD) a likely outcome. While cooperation is not impossible under these conditions, it tends to be rare and fragile. In contrast, when agents are randomly paired across the population, they are continually exposed to new behaviors. This variety introduces opportunities to escape local defection traps and promotes the discovery and reinforcement of cooperative strategies.

This study reveals that forgiving strategies, those that offer opportunities for recovery after defection, can fundamentally reshape learning dynamics in multi-agent systems. In MARL settings, such strategies provide a stabilizing force that prevents convergence to degenerate outcomes. In systems with many agents, such as swarm robotics, decentralized energy grids, or financial trading platforms, missteps and exploitation are inevitable. Forgiving strategies allow agents to absorb occasional adversarial behavior without collapsing into permanent mistrust or retaliation. This creates space for cooperation to recover and persist. From a systems design perspective, incorporating mechanisms that promote or incentivize forgiveness, such as reward shaping, memory-based policies, or structured exploration, can help unlock more resilient cooperative equilibria. This insight also bridges to broader societal systems, where forgiveness underlies everything from diplomatic treaties to community conflict resolution, reinforcing its importance as a universal lever for long-term coordination.

7 Discussion

The central finding is that randomized interactions among selfish agents reverse defectionary outcomes, a result that contradicts prior literature, which points to the fact that stochastic pairings invariably degrade cooperation. In a diverse set of simple Markov games designed to capture a wide range of realistic social dilemmas, agents matched with randomly drawn opponents not only learn to cooperate but also reliably converge to the optimal strategy. Rather than serving as a barrier, population-level stochasticity produces forgiving strategies that act as attractors in the strategic landscape, drawing defectors back into cooperative clusters and guiding the entire population toward mutually beneficial, optimal outcomes, achieved without any engineered interventions such as reward shaping, partner selection, or centralized coordination. Randomized interactions serve as a natural mechanism for robust optimization, enabling selfish agents to develop strategies that perform reliably in the context of conflicting goals.

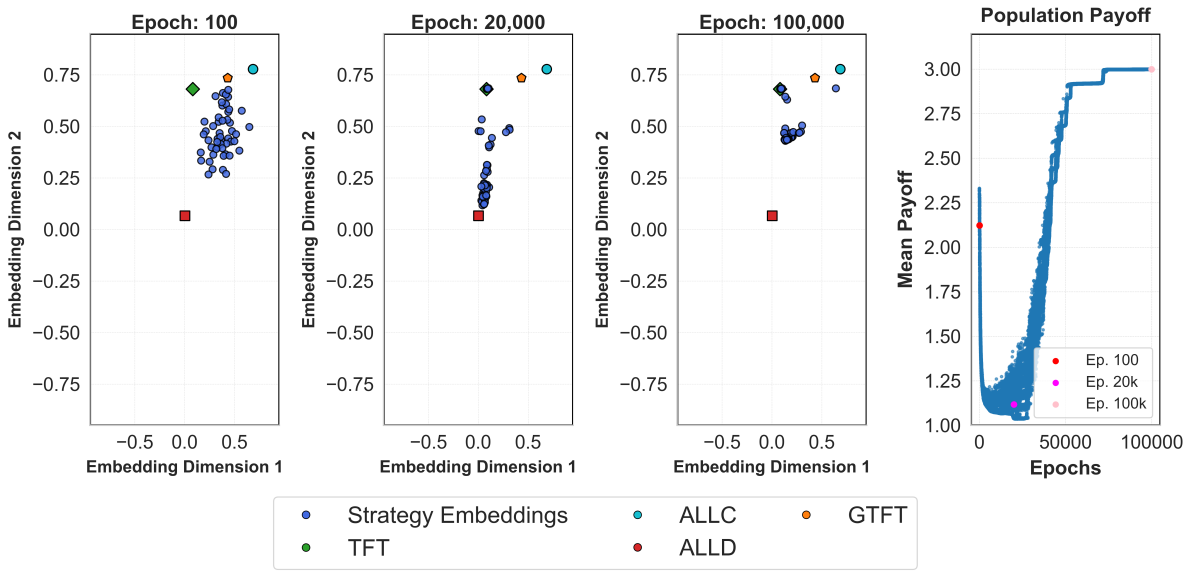


Figure 10: Evolving Strategy Landscape. A sequence of snapshots illustrates the trajectory of agent embeddings over training time. Initially, the population is highly diverse, exploring a wide range of strategies. By epoch 20,000, extortionist strategies dominate, leading to lower overall payoffs. However, diversity persists, and the presence of forgiving strategies creates gradients that shift the population toward cooperation. By epoch 100,000, most agents have moved away from ALLD, converging toward more forgiving, prosocial behaviors. This evolution underscores the role of transient interactions in shaping long-term dynamics and highlights how self-interested learning can lead to cooperative outcomes.

The behavior space autoencoder is a truly novel advance in strategy representation. Unlike conventional dimensionality-reduction methods, such as principal component analysis, that rely on a fixed basis and can quickly become obsolete as strategy distributions evolve, the BSAE learns latent embeddings directly from empirical payoff data. This behavior-centric approach ensures that the representation remains meaningful even as new strategies emerge, clustering them by functional payoff outcomes rather than superficial parameter similarities. As a result, the BSAE not only makes it possible to visualize the strategic landscape but also provides a robust foundation for subsequent optimization techniques that exploit these learned latent spaces to accelerate convergence and improve sample efficiency.

Latent optimization techniques have become widespread in machine learning, yet their application to game-theoretic multi-agent reinforcement learning remains novel. This work shows that even optimizing simple parameter-space embeddings with Latent Reinforcement Learning (LRL) yields strong gains in sample efficiency. However, when combined with the behavior space autoencoder (BSAE), the benefits are amplified: first uncovering the geometry of strategy space and then performing optimization directly within that manifold, this integrated framework shifts multi-agent reinforcement learning away from blind parameter search toward behavior-aligned learning, dramatically increasing sample efficiency.

Beyond overturning longstanding assumptions in multi-agent reinforcement learning, these findings reveal a broader design principle for decentralized, heterogeneous systems: randomness, when combined with behavior-aware learning, can cultivate resilient cooperation. This insight carries significant implications for robotics and autonomous systems. Multi-robot teams and autonomous vehicle fleets frequently depend on carefully engineered coordination proto-

cols that often falter when faced with real-world variability. Introducing randomized encounters among agents provides a more flexible and adaptive approach, enabling systems to organically discover robust, cooperative behaviors without explicit programming. For instance, in autonomous traffic systems, vehicles interacting with a diverse and unpredictable set of partners may naturally develop forgiving strategies that help maintain smooth flow and safety, even when individual agents occasionally make errors. Similarly, in distributed computing or peer-to-peer networks, random peer selection can promote tolerant behaviors, such as accepting intermittent packet loss, that sustain overall system integrity amid decentralization and noise. In these contexts, randomness does not breed chaos but instead guides systems toward globally beneficial outcomes that rigid, pre-scripted solutions frequently fail to achieve.

This dynamic echoes across numerous disciplines. In ecology, random disturbances prevent any single species from dominating resources, thereby preserving biodiversity and maintaining ecosystem resilience. Similarly, random interactions within agent populations break up uniform defection, enabling cooperative strategies to persist, propagate, and ultimately prevail. In economics, decentralized markets often depend on random buyer-seller pairings, where trust is cultivated not through heavy regulation but through repeated, diverse interactions, fostering forgiving norms such as leniency following minor defaults. Sociology reveals comparable patterns, where cooperation and forgiveness arise organically from informal, stochastic exchanges, like gossip, reputation building, and interpersonal negotiation, rather than from top-down mandates. Political science further reinforces this insight: truth and reconciliation processes deliberately introduce unpredictable pairings to disrupt cycles of conflict and rebuild social cohesion. Across these varied fields, randomness acts as a catalyst for flexibility, diversity, and the emergence of stable cooperative norms. These same principles hold true in artificial systems, where forgiving strategies spontaneously emerge through diverse encounters, serving as powerful stabilizers within complex, decentralized environments. Despite these promising insights, several important limitations should be acknowledged. The results are derived from relatively simple stochastic environments, games defined by memory-one strategies and clear payoff structures. While such settings effectively isolate core social dilemmas, they abstract away many complexities inherent to real-world multi-agent systems. Domains with continuous action spaces, partial observability, asynchronous decision-making, or large, heterogeneous populations present open challenges not addressed in this work. Scalability, in particular, remains a central concern, as the computational burden of modeling and optimizing over increasingly rich strategy spaces grows rapidly. Nonetheless, the integration of the behavior space autoencoder (BSAE) with Latent Reinforcement Learning (LRL) offers a potential path forward. By enabling direct optimization within a compressed, behaviorally meaningful latent space, this framework can dramatically reduce sample complexity and improve learning efficiency, making it a promising foundation for scaling cooperation dynamics to more complex and realistic environments.

Several promising avenues for future research arise from this work. One key direction is to explore how forgiving strategies can be systematically introduced into existing populations to shift their dynamics toward more cooperative and socially beneficial outcomes. Understanding how to guide populations toward forgiveness could have broad implications for designing resilient multi-agent systems in economics, robotics, and distributed computing. Another important area is investigating the impact of different social network structures on cooperation dynamics. Since real-world interactions are rarely fully random, studying how network topology influences the emergence and stability of cooperative behaviors can inform the design of more effective decen-

tralized systems and social platforms. Finally, a particularly timely direction is examining how populations of self-interested large language models (LLMs) behave when faced with conflicts of interest. Understanding whether and how population-level dynamics encourage cooperation among LLMs is crucial as these models are increasingly deployed in settings requiring negotiation, collaboration, or conflict resolution. Insights here could help ensure that AI systems align better with human values and collective goals.

Code availability

Implementation details may be found at https://github.com/smerrillunc/population_learning.

References

[1] L. Chen, P. Deng, L. Li, and X. Hu, "Mixed motivation driven social multi-agent reinforcement learning for autonomous driving", IEEE/CAA Journal of Automatica Sinica **12**, 1272–1282 (2025).

[2] R. Zhang, J. Hou, F. Walter, S. Gu, J. Guan, F. Röhrbein, Y. Du, P. Cai, G. Chen, and A. Knoll, *Multi-Agent Reinforcement Learning for Autonomous Driving: A Survey*, 2024.

[3] S. Sarin, S. K. Singh, S. Kumar, S. Goyal, B. B. Gupta, W. Alhalabi, and V. Arya, "Unleashing the power of multi-agent reinforcement learning for algorithmic trading in the digital financial frontier and enterprise information systems", Computers, Materials and Continua **80**, 3123–3138 (2024).

[4] I. Maeda, D. deGraw, M. Kitano, H. Matsushima, H. Sakaji, K. Izumi, and A. Kato, "Deep reinforcement learning in agent based financial market simulation", Journal of Risk and Financial Management **13**, 71 (2020).

[5] Z. Xiong, B. Luo, B.-C. Wang, X. Xu, X. Liu, and T. Huang, "Decentralized multiagent reinforcement learning based state-of-charge balancing strategy for distributed energy storage system", IEEE Transactions on Industrial Informatics **20**, 12450–12460 (2024).

[6] X. Fang, J. Wang, G. Song, Y. Han, Q. Zhao, and Z. Cao, "Multi-agent reinforcement learning approach for residential microgrid energy scheduling", Energies **13**, 123 (2019).

[7] T. Roughgarden, *Selfish routing and the price of anarchy* (The MIT Press, 2005).

[8] J. Hu and M. P. Wellman, "Nash q-learning for general-sum stochastic games", Journal of Machine Learning Research **4**, 1039–1069 (2003).

[9] J. P. Agapiou, A. S. Vezhnevets, E. A. Duéñez-Guzmán, J. Matyas, Y. Mao, P. Sunehag, R. Köster, U. Madhushani, K. Kopparapu, R. Comanescu, et al., *Melting pot 2.0*, 2022.

[10] T. W. Sandholm and R. H. Crites, "Multiagent reinforcement learning in the iterated prisoner's dilemma", Biosystems **37**, 147–166 (1996).

[11] P. Barnett and J. Burden, "Oases of cooperation: an empirical evaluation of reinforcement learning in the iterated prisoner's dilemma", in Proceedings of the workshop on artificial intelligence safety 2022 (safeai 2022) co-located with the thirty-sixth aaai conference on artificial intelligence (aaai2022), Vol. 3087, edited by G. Pedroza, J. Hernández-Orallo, X. C. Chen, X. Huang, H. Espinoza, M. Castillo-Effen, J. McDermid, R. Mallah, and S. Ó hÉigeartaigh, CEUR Workshop Proceedings (2022).

[12] E. M. de Cote, A. Lazaric, and M. Restelli, "Learning to cooperate in multi-agent social dilemmas", in Proceedings of the fifth international joint conference on autonomous agents and multiagent systems, AAMAS '06 (2006), pp. 783–785.

[13] Y. Shoham, R. Powers, and T. Grenager, "If multi-agent learning is the answer, what is the question?", Artificial Intelligence **171**, 365–377 (2007).

[14] J. Z. Leibo, V. Zambaldi, M. Lanctot, J. Marecki, and T. Graepel, *Multi-agent reinforcement learning in sequential social dilemmas*, 2017.

[15] E. Hughes, J. Z. Leibo, M. Phillips, K. Tuyls, E. Dueñez-Guzman, A. García Castañeda, I. Dunning, T. Zhu, K. McKee, R. Koster, et al., "Inequity aversion improves cooperation in intertemporal social dilemmas", in Advances in neural information processing systems, Vol. 31 (2018).

[16] U. Madhushani, K. R. McKee, J. P. Agapiou, J. Z. Leibo, R. Everett, T. Anthony, E. Hughes, K. Tuyls, and E. A. Duéñez-Guzmán, *Heterogeneous social value orientation leads to meaningful diversity in sequential social dilemmas*, 2023.

[17] E. A. Duéñez-Guzmán, R. Comanescu, Y. Mao, K. R. McKee, B. Coppin, S. Sadedin, S. Chiappa, A. S. Vezhnevets, M. A. Bakker, Y. Bachrach, W. Isaac, K. Tuyls, and J. Z. Leibo, "Perceptual interventions ameliorate statistical discrimination in learning agents", Proceedings of the National Academy of Sciences **122**, e2319933121 (2025).

[18] K. R. McKee, X. Bai, and S. T. Fiske, "Warmth and competence in human-agent cooperation", Autonomous Agents and Multi-Agent Systems **38**, 23 (2024).

[19] E. Bahel, S. Ball, and S. Sarangi, "Communication and cooperation in prisoner's dilemma games", Games and Economic Behavior **133**, 126–137 (2022).

[20] E. M. de Cote, A. Lazaric, and M. Restelli, "Learning to cooperate in multi-agent social dilemmas", in Proceedings of the fifth international joint conference on autonomous agents and multiagent systems, AAMAS '06 (2006), pp. 783–785.

[21] W. Barfuss and J. M. Meylahn, "Intrinsic fluctuations of reinforcement learning promote cooperation", Scientific Reports **13**, 10.1038/s41598-023-27672-7 (2023).

[22] J. Foerster, R. Y. Chen, M. Al-Shedivat, S. Whiteson, P. Abbeel, and I. Mordatch, "Learning with Opponent-Learning Awareness", in Proceedings of the 17th international conference on autonomous agents and multiagent systems, AAMAS '18 (2018), pp. 122–130.

[23] M. Babes, E. M. de Cote, and M. L. Littman, "Social reward shaping in the prisoner's dilemma", in Proceedings of the 7th international joint conference on autonomous agents and multiagent systems - volume 3, AAMAS '08 (2008), pp. 1389–1392.

[24] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch, *Multi-agent actor-critic for mixed cooperative-competitive environments*, 2017.

[25] W. Barfuss and J. M. Meylahn, "Intrinsic fluctuations of reinforcement learning promote cooperation", Scientific Reports **13**, 10.1038/s41598-023-27672-7 (2023).

[26] M. L. Littman, "Friend-or-Foe Q-learning in General-Sum Games", in Proceedings of the eighteenth international conference on machine learning (2001), pp. 322–328.

[27] A. McAvoy, Y. Mori, and J. B. Plotkin, "Selfish optimization and collective learning in populations", Physica D: Nonlinear Phenomena **439**, 133426 (2022).

[28] N. Anastassacos, S. Hailes, and M. Musolesi, "Partner selection for the emergence of cooperation in multi-agent systems using reinforcement learning", Proceedings of the AAAI Conference on Artificial Intelligence **34**, 7047–7054 (2020).

[29] C.-w. Leung and P. Turrini, "Learning partner selection rules that sustain cooperation in social dilemmas with the option of opting out", in Proceedings of the 23rd international conference on autonomous agents and multiagent systems, AAMAS '24 (2024), pp. 1110–1118.

[30] L. S. Shapley, "Stochastic games", Proceedings of the National Academy of Sciences **39**, 1095–1100 (1953).

[31] A. Lerer and A. Peysakhovich, *Maintaining cooperation in complex social dilemmas using deep reinforcement learning*, 2017.

[32] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation", in Advances in neural information processing systems, Vol. 12, edited by S. Solla, T. Leen, and K. Müller (1999).

[33] W. Li and G. Montúfar, "Natural gradient via optimal transport", Information Geometry **1**, 181–214 (2018).