Environment Complexity and Nash Equilibria in a Sequential Social Dilemma

Mustafa Yasir
The Alan Turing Institute
myasir@turing.ac.uk

Andrew Howes
University of Exeter
andrew.howes@exeter.ac.uk

Vasilios Mavroudis
The Alan Turing Institute
vmavroudis@turing.ac.uk

Chris Hicks
The Alan Turing Institute
c.hicks@turing.ac.uk

Abstract

Multi-agent reinforcement learning (MARL) methods, while effective in zero-sum or positive-sum games, often yield suboptimal outcomes in general-sum games where cooperation is essential for achieving globally optimal outcomes. Matrix game social dilemmas, which abstract key aspects of general-sum interactions, such as cooperation, risk, and trust, fail to model the temporal and spatial dynamics characteristic of real-world scenarios. In response, our study extends matrix game social dilemmas into more complex, higher-dimensional MARL environments. We adapt a gridworld implementation of the Stag Hunt dilemma to more closely match the decision-space of a one-shot matrix game while also introducing variable environment complexity. Our findings indicate that as complexity increases, MARL agents trained in these environments converge to suboptimal strategies, consistent with the risk-dominant Nash equilibria strategies found in matrix games. Our work highlights the impact of environment complexity on achieving optimal outcomes in higher-dimensional game-theoretic MARL environments.

1 Introduction

Multi-agent reinforcement learning (MARL) is concerned with training agents to maximise individual or shared rewards in environments with multiple concurrent learners. A critical aspect of MARL is the agent motive, which can be categorised as competitive (zero-sum), purely cooperative (positive-sum), or mixed (general-sum). MARL methods have shown considerable success both in competitive environments [9, 32] and in purely cooperative environments [27, 4]. In these settings, agent goals are either directly opposed or aligned, and learning paradigms like self-play, which train agents to compute best responses to rational opponents, empirically converge to stable strategies, i.e., Nash equilibria, that correspond to optimal outcomes. However, Nash equilibria in general-sum games often coincide with suboptimal outcomes, rendering the direct application of such methods ineffective for finding desirable solutions [8, 6, 14]. Instead, in many general-sum games, optimal outcomes are often only achievable through cooperation—by explicitly favouring collective gains over stable individual rewards.

Matrix game social dilemmas (MGSDs), a class of general-sum games, represent situations where individual rationality may lead to collective irrationality. These dilemmas are widely used to explore cooperation in various disciplines [23, 26], including MARL, where numerous methods have been proposed to induce cooperation among learning agents [7]. While MGSDs are valuable for abstracting key aspects of real-world multi-agent interactions—such as cooperation, risk, and trust—they lack the complexity of real-world temporal and spatial dynamics. Unlike real-world actions, matrix game

17th European Workshop on Reinforcement Learning (ewrl 2024).

actions are binary, i.e., limited to cooperate or defect choices, and do not model extended timeframes [12]. Consequently, recent works have looked at extending MGSDs to higher-dimensional MARL environments in order to more accurately model real-world scenarios (see Section 2).

In this study, we focus on an extension of the Stag Hunt MGSD to a gridworld MARL environment, and explore how unique environment dynamics—absent in the matrix game—affect the learnt strategies of independent MARL agents. Specifically, we introduce eight novel variants of the gridworld Stag Hunt, each incoporating a different degree of randomisation across various dynamics, which we model as environment complexity. This research direction is motivated by the inherent limitations of MGSDs, where modifications to a game are typically confined to changes in payoff structure. Extending these games into higher-dimensional environments offers a novel opportunity to examine how additional factors, such as environment dynamics, influence cooperative behaviours.

Furthermore, our inquiry is supported by studies on randomness in evolutionary social dilemmas [28], along with empirical research highlighting its impact on cooperation [1, 34]. A key finding is that multiple variants of the gridworld Stag Hunt can map to the same Stag Hunt MGSD; however, even minor variations in their complexity can promote convergence to vastly different strategies. Moreover, our work advances the development of game-theoretic MARL environments, more accurately reflecting real-world conditions, where environment dynamics may play a significant role in shaping general-sum multi-agent interaction. In summary, we present the following key contributions:

- We adapt a multi-agent gridworld implementation of the Stag Hunt MGSD to match the decision space of a one-shot matrix game and create eight unique environment variants, each with a specified level of complexity.
- We train independent MARL agents in each of these environment variants and observe distinct empirical convergence patterns to either the suboptimal Nash equilibrium strategy or a mixed strategy.
- We demonstrate that environment complexity promotes systematic convergence to the suboptimal Nash equilibrium strategy, despite the established viability of more optimal strategies.
- We conduct an empirical game-theoretic analysis on our trained policies, showing that certain gridworld Stag Hunt variants can be mapped to MGSDs with Stag Hunt payoffs, and can be modelled as Sequential Social Dilemmas.

2 Related Work

Addressing the viability of MGSDs in modelling key aspects of multi-agent interaction, numerous studies have proposed higher-dimensional game-theoretic MARL environments to explore cooperation under more realistic settings. The Coin Game, a widely used gridworld environment based on repeated MGSDs, has been used to examine cooperative MARL methods by Foerster et al. [8], and the Melting Pot suite combines various MGSDs in gridworld formats to benchmark MARL algorithms [13]. Recently, Khan et al. [11] highlight the challenges in adapting matrix games to gridworld environments by demonstrating the shortcomings of the Coin Game. Relevant to our focus on the Stag Hunt, is the Level-Based Foraging environment [5], a gridworld foraging game which targets multi-agent cooperation and coordination, and a study by Nica et al. [20], which successfully demonstrates enhanced learning in a Minecraft-based stag hunt through gridworld abstraction. Although existing works provide useful game-theoretic environments, primarily for proposing new MARL methods for repeated and partially observable settings, we simplify the learning problem to one-shot and fully observable games; to focus on modifying environment dynamics, whilst maintaining the relation between our environment and its matrix game formulation. To this end, we leverage an existing game-theoretic MARL environment [19, 22] , based on the Stag Hunt MGSD, chosen for its overlap with existing environments and accessibility of manipulating game dynamics, which was briefly explored by Peysakhovich and Lerer [22].

3 Background

3.1 Matrix Game Social Dilemmas

A matrix game is a formal representation of strategic interactions between two players, where each player's decision affects the other's outcome. The game is represented by a matrix in which each cell details the outcomes (or payoffs) for the players based on their combined choices. Players choose a strategy without knowledge of the other's simultaneous decision, and the combination of their choices leads to a specific payoff for each. In matrix game social dilemmas (MGSDs), the two actions, or strategies, available to each player are *'cooperate'* and *'defect'*, and the exact payoffs for each player are given by the values $\{R, P, S, T\} \in \mathbb{R}$, detailed in Table 1. Formally, a matrix game is a social dilemma when its four payoffs satisfy the following inequalities:

$$R > P, R > S, 2R > T + S, \text{ and either: } T > R, \text{or, } P > S. \tag{1}$$

Table 1: Payoff table for a matrix game social dilemma, where (X, Y) denotes the payoff for the given row and column player, respectively.

		Player 2	
		$Cooperate$	$Defect$
Player 1	$Cooperate$	(R, R)	(S, T)
	$Defect$	(T, S)	(P, P)

Central to our work, is the MGSD, Stag Hunt [31]; where the *'cooperate'* and *'defect'* actions correspond to *'hunt'* and *'forage'*, respectively. Conceptually, this represents a scenario where players are tasked with either hunting a stag or foraging a plant. A bilateral decision to hunt offers the highest payoff, h, to both players, but carries the risk of being mauled by the stag, yielding the lowest possible reward, m, to a player hunting unilaterally. In mauling scenarios, the opposing player may receive a unique foraging reward $f^* \geq f$. Formally, the Stag Hunt is characterised by Table 2 and corresponds to an MGSD where $R = h$, $P = f$, $S = m$ and $T = f^*$.

Table 2: Payoff table for the generalised 2-player stag hunt, where $h > f^* \geq f > m$.

		Player 2	
		$Hunt$	$Forage$
Player 1	$Hunt$	(h, h)	(m, f^*)
	$Forage$	(f^*, m)	(f, f)

3.2 Nash Equilibria

Nash Equilibria, a fundamental concept in game theory introduced by Nash [18], refers to a situation in which each player in a game makes an optimal choice considering the choices of the other players, and no player has anything to gain by changing only their own strategy unilaterally. Formally, a Nash Equilibrium in a game with n players is defined as a strategy profile $(s_1^*, s_2^*, \ldots, s_n^*)$ such that for each player i, the strategy s_i^* is a best response to the strategies s_{-i}^* chosen by the other players. This can be expressed as:

$$s_i^* \in \arg\max_{s_i} u_i(s_i, s_{-i}^*), \quad \forall i \in \{1, 2, \ldots, n\} \tag{2}$$

where s_i represents the strategy chosen by player i, s_{-i}^* represents the strategies chosen by all other players except i, and $u_i(s_i, s_{-i})$ is the payoff function for player i. In the Stag Hunt, players are faced with the decision between hunting a stag or foraging for plants, i.e., $s_i \in \{Hunt, Forage\}$, and the game is particularly valuable for containing two Nash Equilibria which reflect different resolutions of its inherent social dilemma:

Payoff-Dominant Equilibrium $(Hunt, Hunt)$**:** If both players choose to hunt stag, neither has an incentive to deviate unilaterally since doing so would result in a lower payoff. If one player switches to foraging while the other continues to hunt, the deviating player's payoff decreases from h (under successful cooperation) to f or f^*, both of which are less than h under the game's assumptions. Thus, $(Hunt, Hunt)$ is a Nash Equilibrium. This outcome is payoff dominant because it results in the highest rewards for both players, i.e., both receive h, which is greater than any other payoff in the game.

Risk-Dominant Equilibrium $(Forage, Forage)$**:** Similarly, if both players choose to forage, deviating to hunting is disadvantageous. A unilateral switch would result in a payoff of m, as the lone hunter would face the risk of mauling, which is the worst outcome. Hence, $(Forage, Forage)$ also constitutes a Nash Equilibrium. Although this equilibrium yields a lower payoff, f, than mutual stag hunting, it is considered risk dominant because it minimises the risk associated with the potential non-cooperation of the other player.

3.3 Markov Games

We model the MARL adaptation of Stag Hunt as a two-player Markov Game, also known as a Stochastic Game [25]; a natural extension of Markov Decision Processes to multi-agent settings [17]. A Markov game for two players is a tuple $\langle S, A, R, P, \gamma \rangle$, where S is the set of states describing the environment, A is the combined action space across all players, given by $A := A_1 \times A_2$, where the A_i is is the set of actions available to player i, R is the reward function, given by $R := R_1 \times R_2$, where $R_i : S \times A \times S \to \mathbb{R}$ is the reward function for player i, which maps a state and action tuple (including the resulting state) to a real number and defines the reward that player i receives after all players choose their actions. $P : S \times A \times S \to \Delta(S)$ is the state transition probability function, and γ is the discount factor, which is typically constrained to $0 \leq \gamma \leq 1$.

In a two-player Markov Game, each player chooses a policy that determines their action based on the current state of the game. The joint policy vector $\vec{\pi}$ combines the individual policies of each player, where $\vec{\pi} = (\pi_1, \pi_2)$. For player i , the value function under this joint policy, $V_i^{\vec{\pi}}$, is defined as the expected sum of discounted rewards when both players adhere to their respective parts of $\vec{\pi}$. The formal definition is:

$$V_i^{\vec{\pi}}(s) = \mathbb{E}_{\vec{a}_t \sim \vec{\pi}, s_{t+1} \sim P(s_t, \vec{a}_t)} \left[\sum_{t=0}^{\infty} \gamma^t R_i(s_t, \vec{a}_t, s_{t+1}) \middle| s_0 = s \right] \tag{3}$$

where s_t is the state at time t, $\vec{a}_t$ is the joint actions taken by Player 1 and Player 2 at time t, determined by their policies π_1 and π_2, respectively, s_{t+1} is the state resulting from the actions $\vec{a}_t$, $R_i(s_t, \vec{a}_t, s_{t+1})$ is the reward received by player i after the actions are taken and the state transitions, and γ is the discount factor.

3.4 Sequential Social Dilemmas

We also use the Sequential Social Dilemma (SSD) model, introduced by Leibo et al. [12], which characterises when a Markov Game contains embedded MGSDs, thus extending the MGSD model to temporally and spatially extended settings. Formally, an SSD is a tuple $\langle \mathcal{M}, \Pi^C, \Pi^D \rangle$, where $\mathcal{M}$ is a Markov Game with state space $\mathcal{S}$, and, Π^C and Π^D, are disjoint sets of policies that implement cooperative and defective strategies, respectively. These policy sets are defined by selecting a metric, $\alpha : \Pi \to \mathbb{R}$, and threshold values α_c and α_d such that $\alpha(\pi) < \alpha_c \iff \pi \in \Pi_C$ and $\alpha(\pi) > \alpha_d \iff \pi \in \Pi_D$. For a given Markov Game, $\mathcal{M}$, and state, $s \in \mathcal{M}$, we can construct an empirical payoff matrix, $(R(s), P(s), S(s), T(s))$, induced by the outcomes of policies, $(\pi^C \in \Pi^C, \pi^D \in \Pi^D)$, via the following equations: $R(s) := V_1^{\pi^C, \pi^C}(s) = V_2^{\pi^C, \pi^C}(s)$, $P(s) := V_1^{\pi^D, \pi^D}(s) = V_2^{\pi^D, \pi^D}(s)$, $S(s) := V_1^{\pi^C, \pi^D}(s) = V_2^{\pi^D, \pi^C}(s)$, and $T(s) := V_1^{\pi^D, \pi^C}(s) = V_2^{\pi^C, \pi^D}(s)$. Then, $\mathcal{M}$ is an SSD when there exist states $s \in S$ and policies $(\pi^C \in \Pi^C, \pi^D \in \Pi^D)$ such that the induced empirical payoff matrix satisfies the five MGSD inequalities in Equation 1. Hence, a Markov Game is an SSD when it can be mapped by empirical game-theoretic analysis [33] to an MGSD.

3.5 Proximal Policy Optimisation

Proximal Policy Optimisation (PPO) is a widely used policy gradient method for single-agent reinforcement learning, introduced by Schulman et al. [24]. The core idea behind PPO is the use of a clipped surrogate objective function to prevent the policy from deviating too far from the current policy during updates. Through its extension to the multi-agent paradigm, PPO has proven effective in numerous MARL environments, including those that target cooperation [35].

4 Methods

4.1 Gridworld Stag Hunt

We adapt a gridworld implementation of the Stag Hunt matrix game, initially developed by Nesterov-Rappoport [19] and Peysakhovich and Lerer [22]. The selected environment, built on the PettingZoo [30] and Gym [3] frameworks, simulates a 5x5 grid game featuring two agents—red and blue, visualised in Appendix A. Each episode begins with placing both agents, along with non-agent entities: one stag and two plants, on distinct cells of the grid. The agents and the stag can move in any of the four cardinal directions or remain stationary at every timestep. The state of the environment is fully observable for each agent, with integer arrays representing the coordinates of all agents and entities on the grid. Similarly, the action space is represented by five unique integers, corresponding to the four cardinal directions and the option to remain stationary. Plants remain stationary within an episode and are randomly re-spawned at every episode start. Individual agent rewards are dependent on the actions of both agents and are determined at the end of every timestep as follows:

- Agents receive an individual hunting reward, h, by jointly occupying the stag's grid cell.
- An agent receives a foraging reward, f, by independently, or jointly, occupying a plant's grid cell.
- An agent receives a mauling reward, m, by independently occupying the stag's grid cell.

The rewards values assigned satisfy the generalised two-player stag hunt inequality in Table 2, with values set at $h = 25, f = f^* = 2$ and $m = -1$. This configuration significantly favours hunting strategies over foraging strategies, whilst inducing an element of risk to hunting with a negative mauling reward. Furthermore, to align the gridworld stag hunt with a one-shot matrix game, we implemented an infinite-horizon setup. In this format, episodes conclude only when both agents have received a reward: through hunting, foraging, or being mauled. In asynchronous episode terminations, an agent that has received a reward is frozen on the grid, while the other agent continues to act.

4.1.1 Variable Complexity

In Table 3 we introduce randomised and deterministic configurations for three environment parameters in the gridworld Stag Hunt: the spawn locations of agents, the spawn locations of the stag, and movement of the stag, where agent and stag spawn parameters apply to the start of every episode, and the stag movement applies to every timestep. Enumerating all combinations of values for these parameters provides eight unique environment variants, each with a specified degree of randomisation. These environment variants are outlined in Table 4 and we use the corresponding environment label to denote the degree of randomisation in a given environment, i.e., the label **FFR** denotes an environment with: deterministic agent spawn (**F**), deterministic stag spawn (**F**) and randomised stag movement (**R**). We characterise the spectrum of randomisation across these eight variants as environment complexity in our study, where environments with randomisation on multiple parameters, such as environment **RRR**, are considered more complex than those with deterministic dynamics, such as environment **FFF**.

4.2 Experimental Setup

Our experiments use RLLib's [16] Proximal Policy Optimisation (PPO) [24] implementation to train pairs of agents in the gridworld Stag Hunt environment. We consider the simplest multi-agent variant of PPO, Independent PPO (IPPO), in which each agent uses an independent instance of PPO to optimise its own policy, through maximising individual rewards [29]. In particular, no parameters are shared between the multiple policies and each agent treats other agents as part of their environment

Table 3: Additional configurations for three environment parameters introduced to control complexity in the gridworld Stag Hunt.

Environment Parameter	Value	Description
Agent Spawn	Fixed	Blue and red agents spawn in the upper left and right corners of the grid, respectively.
	Random	Blue and red agents spawn at random, distinct cells on the grid.
Stag Spawn	Fixed	Stag spawns at the center cell of the grid.
	Random	Stag spawns at a random cell location, distinct from either agent.
Stag Movement	Follows	Stag moves to a neighbouring cell closest to the nearest agent, measured by Euclidean distance.
	Random	Stag moves to a random neighbouring cell.

Table 4: The eight environment variants we create with our modifications to the gridworld Stag Hunt. Each row represents an environment with a specified degree of complexity.

Environment Label	Agent spawn	Stag spawn	Stag movement
FFF	Fixed	Fixed	Follows
RFF	Random	Fixed	Follows
FRF	Fixed	Random	Follows
FFR	Fixed	Fixed	Random
RRF	Random	Random	Follows
FRR	Fixed	Random	Random
RFR	Random	Fixed	Random
RRR	Random	Random	Random

observation. Note that, given the global observation space provided to agents in the gridworld Stag Hunt, our IPPO approach aligns closely with the widely used multi-agent shared-critic variant of PPO, MAPPO [35]. We also consider a single-agent approach to the multi-agent problem, using fully-centralised PPO to serve as baseline to compare the performance of IPPO. In this approach, the observations of both agents are combined into a single observation, to train a fully centralised joint-action policy that controls both agents and maximises combined rewards.

5 Results

5.1 Environment Complexity and Strategy Convergence

We conduct five training runs of IPPO and PPO in each of the eight environments listed in Table 4, training for 1,000 iterations, each consisting of 4,000 environment timesteps. In each trial we use a distinct random seed for determining environment randomisation dynamics where applicable. We defer the results of our baseline, PPO, and the hyperparameters used, to Appendices B and C.1.

In Figure 1, which shows the training performance of IPPO agents, measured by mean combined episode reward per iteration, averaged across 5 trials; we group environments based on observed reward patterns. Environments yielding high rewards (i.e., **FFF, RFF, FRF, RRF, FRR**) are denoted as Group A, while those with lower rewards (**FFR, RFR, RRR**) constitute Group B. These groupings reveal two insights into the learnt strategies of IPPO agents:

Observation 1: Agents trained in Group B converge to globally suboptimal strategies, consisting of bilateral foraging. By analysing the mean episode rewards at the end of 1,000 training iterations, we infer agent strategies based on the previously established payoff structure. Specifically, exclusive bilateral hunting yields a combined reward of 50, significantly higher than the 4 from bilateral foraging. Consequently, stable convergence to a mean episode reward over 4 indicates that agents have adopted strategies that include bilateral hunting. Thus, in Group B, where the mean combined rewards rarely

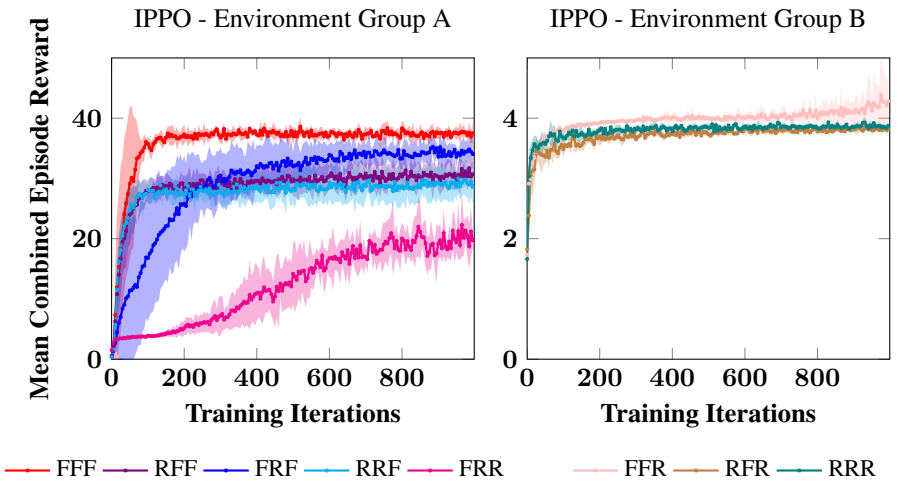


Figure 1: Training performance of IPPO in eight environments with varying degrees of complexity, detailed in Table 4. Environments are grouped by reward patterns: Group A (left) includes **FFF, RFF, FRF, RRF, FRR**, and Group B (right) includes **FFR, RFR, RRR**. Bold lines represent the average performance over five trials, and the shading represents ±1 standard deviation from each point.

exceed 4 (second column of Figure 1), we can conclude that agents have almost-exclusively converged to bilateral foraging strategies.

Observation 2: Agents trained in Group A converge to mixed-strategies, consisting of both bilateral foraging and hunting. In Group A, where mean combined rewards consistently exceed the foraging threshold (1st column of Figure 1), there is evident near-convergence to bilateral hunting strategies. However, we observe that agents in these environments do not achieve a consistent reward of 50, which is required to indicate convergence to exclusive bilateral hunting strategies. Specifically, in the 1,000th training iteration

5.2 Escaping Suboptimal Strategies

We consider only Group B environments from the previous subsection, in which we observed IPPO agents converging to suboptimal strategies (Observation 1). To assess the feasibility of higher-reward strategies in these environments, we bias agents toward bilateral hunting strategies through a simple curriculum. We introduce a new environment variant, **cXXX**, designed to enforce a purely cooperative Nash equilibrium by eliminating the reward for foraging, thus exclusively incentivising bilateral hunting. This corresponds to a Stag Hunt game, as formalised in Table 2, where $h = 25$, $f = f^* = 0$ and $m = -1$.

For each environment in Group B, a corresponding **cXXX** variant is created: **cFFR, cRFR** and **cRRR**. These variants maintain their respective randomisation dynamics described in Table 4. Initially, agents are trained in their respective **cXXX** environments for 500 iterations to learn hunting strategies (Stage 1). To discourage passive strategies and further promote hunting, a timestep penalty of -0.5 is introduced in these environments. Following this step, the pre-trained policies are then trained in their original environments (**FFR, RFR, RRR**) for another 500 iterations, during which the timestep penalty is removed (Stage 2). We use only IPPO for these experiments.

Observation 3: Curriculum learning allows agents to escape suboptimal strategies in Group B. Our results in Figure 2 demonstrate that our curriculum enables agents to find strategies in Group B environments that yield significantly higher rewards than those observed previously. In particular, during the second training stage, performance begins to converge toward average rewards that substantially exceed those attainable through exclusive foraging. Consistent learning of higher-reward strategies demonstrates that agents have not only discovered, but also converged to strategies that include bilateral hunting—effectively escaping the suboptimal strategies that were previously observed.

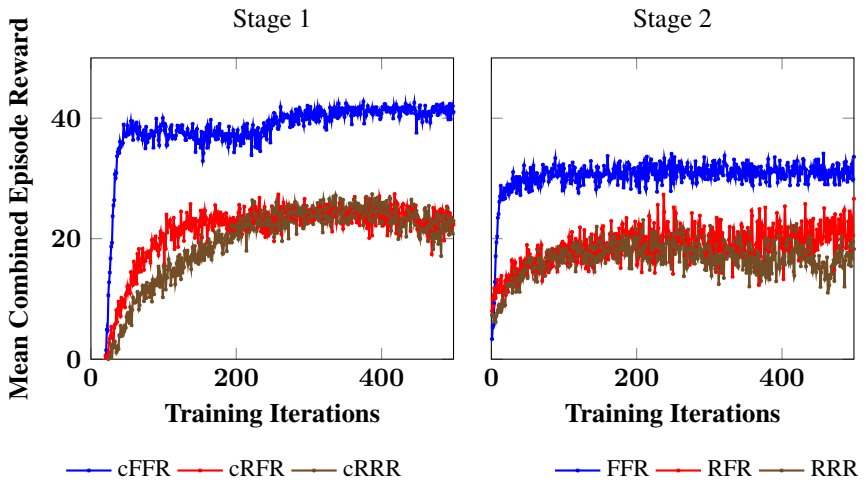


Figure 2: Training performance of IPPO in a 2-stage curriculum in Group B environments from Section 5.1. Agents are initially trained in a cooperation-inducing environment (**cXXX**), before being trained in their target environments.

5.3 Empirical Game-Theoretic Analysis

Here we further characterise the Group B environments with an empirical game-theoretic analysis following the methodology of Leibo et al. [12]. Recall from Section 5.1 that agents trained in these environments converged to suboptimal foraging strategies (Observation 1). In the context of the Stag Hunt, foraging corresponds to the defect action within a matrix game social dilemma (MGSD). Consequently, we categorise the strategies that these policies implement as defective. Conversely, in Section 5.2, we present a learning curriculum that shifts agents from these suboptimal strategies towards cooperative hunting-focused strategies. Thus we can train two distinct types of policy for each environment in Group B: cooperative policies, Π^C, and defective policies, Π^D. This allows us to construct a meta-game with cooperate and defect actions that involve sampling a policy from either category and determining payoffs based on the resulting pairwise average rewards. This meta-game represents an MGSD, induced by the policy sets, Π^C and Π^D, embedded within the underlying Markov Game, if such a mapping exists. We compute the empirical payoffs of this meta-game as follows:

For each environment, two pairs of policies, (π_1^C, π_1^D) and (π_2^C, π_2^D), are sampled, where $\pi_i^C \in \Pi^C$ and $\pi_j^D \in \Pi^D$. These policy pairs represent the actions of a row player and a column player in a two-player matrix game with cooperate or defect actions (i.e., an MGSD), respectively. Specifically, π_i^C corresponds to the cooperate action for player i, and π_j^D corresponds to the defect action for player j. The outcomes (payoffs) for each combination of actions are determined by playing the respective policies against one another for an episode, in the respective Stag Hunt environment. The rewards obtained in each episode are then averaged in the corresponding cell in the payoff matrix. This procedure is repeated for each environment until convergence of all cell values, yielding the set of induced empirical payoff matrices shown in Table 5.

Table 5: Empirical payoff matrices for the meta-game in Group B environments. Each cell in the table contains a reward tuple: (x, y), where x is the reward for player/policy 1, and y is the reward for player/policy 2, averaged over 5,000 episodes (2 d.p.)

	RFR		RRR		FFR	
	π_2^C	π_2^D	π_2^C	π_2^D	π_2^C	π_2^D
π_1^C	(10.76, 10.81)	(0.45, 1.89)	(8.38, 8.37)	(0.42, 1.87)	(15.29, 15.26)	(0.49, 1.88)
π_1^D	(1.95, 0.59)	**(1.93, 1.89)**	(1.96, 0.33)	**(1.95, 1.87)**	(2.02, 0.24)	**(1.98, 1.99)**

Observation 4: Group B environments map to MGSDs with Stag Hunt payoffs, and consequently, can be modelled as SSDs. Section 5.3 maps a Markov Game to an MGSD, induced by a set of policies. To verify if this mapping is valid, here we examine the empirical payoffs of the constructed MGSD. It is important to note that it is not necessary for this process to generate empirical payoffs that constitute a valid MGSD, as the space of Markov Games is much larger than that of MGSDs. This is particularly relevant in the gridworld Stag Hunt we consider in this work, alongside our modifications, which has not been formally characterised in relation to the MGSD Stag Hunt. Nonetheless, the empirical payoff matrices in Table 5.3 satisfy the five the MGSD inequalities in Equation 1, as well as the Stag Hunt inequality in Table 2. The Markov Games that model Group B environments can be mapped to MGSD Stag Hunt games using the method described above. Consequently, they can indeed be modelled as Sequential Social Dilemmas (SSDs) (Section 3.3).

Observation 5: Agents trained in Group B empirically converge to the suboptimal, risk-dominant Nash equilibrium strategy from the MGSD Stag Hunt. Although formally verifying Nash equilibria in general-sum Markov Game policies can often be intractable [21, 10], our analysis provides empirical evidence that agents in Section 5.1 have converged not only to suboptimal strategies (Insight 1), but also to Nash equilibria strategies. Consider the outcomes of two defecting policies, (π_1^D, π_2^D), in Table 5. If agents had not converged to the suboptimal Nash equilibrium, we would see empirical payoffs for these outcomes exceeding a reward of 2 for each defecting policy. Instead, we observe convergence to a reward less than 2, indicating that when these policies attempt to deviate from the suboptimal strategy, they only reduce their own payoffs. This behaviour aligns precisely with the definition of Nash equilibria. Moreover, the specific equilibrium they converge to is reflective of the risk-dominant Nash equilibrium in the MGSD Stag Hunt (Section 3.1).

6 Discussion

Our results indicate that agents trained in Group B environments converge to suboptimal strategies (Observation 1) and that these policies empirically represent Nash equilibria strategies (Observation 5). However, we also discover that agents are in-fact capable of learning more-optimal strategies within these environments (Observation 3). Despite the viability of higher-reward strategies, we observe that agents *systematically* converge to the suboptimal Nash equilibrium strategy (i.e., mutual foraging). We suggest that this behaviour is a result of the increased complexity introduced by our modifications to the gridworld Stag Hunt. To support this, we explore three alternative explanations for our empirical findings:

First, although the observation space for each agent includes the positions of all entities within the grid, converging on higher-reward strategies in more complex environments, such as those in Group B, may require previous observations, actions, or rewards to be included in the agent's observation. However, this information is not available in a one-shot version of the game and thus cannot be considered a viable solution. Second, it could be that performing higher-reward strategies in Group B environments is significantly improbable due to highly randomised environment dynamics, and hence agents are behaving as expected, converging to the most optimal strategy (i.e., maximising long-term reward). We mitigate this proposition (and also the first explanation) by demonstrating empirically that through curriculum learning the same agents are able to learn significantly more cooperative strategies (Observation 3). Third, the dynamics of Group B environments might simply require more suitable hyperparameter values or additional training time, e.g., to increase exploration behaviour or allow more time to discover better strategies. Thus, in Appendix C.2, we run further training experiments in Group B, including multiple hyperparameter searches, for 12,000 iterations. The results show that no further improvements are made under any circumstances, indicating that the observed behaviour is likely not an artifact of hyperparameter selection or insufficient training time, but a systematic defection toward lower-reward strategies due to increased environment complexity.

Thus, our findings suggest that the observed suboptimal convergence behaviour is a direct result of the increased complexity found in the Group B environments. Examining the environment dynamics within Group B indicates that the complexity introduced by a randomly moving stag—environments labelled by **xxR**—induces significant uncertainty which constrains learned strategies to suboptimal equilibria. Indeed, in three of the four environments featuring a randomly moving stag, our agents

are consistently unable to learn cooperative strategies. Furthermore, environments with intermediate levels of complexity— i.e., those in-between **FFF** and **RRR**—maintain average rewards above the suboptimal strategy threshold (Observation 2) and show rewards decreasing proportionally to increased environment complexity.

Finally, our results show that even within the framework of SSDs, multiple variants of the gridworld Stag Hunt can map to the same MGSD (Observation 4) and yet slight variations in environment dynamics, i.e., increased complexity, can promote convergence to vastly different strategies. Future work will seek to formalise Nash equilibria strategies in SSDs, and the notion of environment complexity, in the context of these empirical findings.

Acknowledgements

Research was sponsored by the Army and was accomplished under Cooperative Agreement Number W911NF-22-2-0162. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

References

[1] Forest Baker and H. Rachlin. Teaching and learning in a probabilistic prisoner's dilemma. *Behavioural Processes*, 2002.

[2] James Bergstra, Daniel Yamins, and David Cox. Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. In *International Conference on Machine Learning*, pages 115–123. PMLR, 2013.

[3] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym, 2016.

[4] Micah Carroll, Rohin Shah, Mark K. Ho, Thomas L. Griffiths, Sanjit A. Seshia, Pieter Abbeel, and Anca Dragan. *On the utility of learning about humans for human-AI coordination*. Curran Associates Inc., Red Hook, NY, USA, 2019.

[5] Filippos Christianos, Lukas Schäfer, and Stefano Albrecht. Shared experience actor-critic for multi-agent reinforcement learning. *Advances in neural information processing systems*, 33: 10707–10717, 2020.

[6] Allan Dafoe, Edward Hughes, Yoram Bachrach, Tantum Collins, Kevin R. McKee, Joel Z. Leibo, Kate Larson, and Thore Graepel. Open problems in cooperative ai, 2020.

[7] Yali Du, Joel Z. Leibo, Usman Islam, Richard Willis, and Peter Sunehag. A review of cooperation in multi-agent learning. (arXiv:2312.05162), December 2023. doi: 10.48550/arXiv.2312.05162. arXiv:2312.05162 [cs].

[8] Jakob Foerster, Richard Y. Chen, Maruan Al-Shedivat, Shimon Whiteson, Pieter Abbeel, and Igor Mordatch. Learning with opponent-learning awareness. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '18, page 122–130, Richland, SC, 2018. International Foundation for Autonomous Agents and Multiagent Systems.

[9] Max Jaderberg, Wojciech M. Czarnecki, Iain Dunning, Luke Marris, Guy Lever, Antonio Garcia Castañeda, Charles Beattie, Neil C. Rabinowitz, Ari S. Morcos, Avraham Ruderman, Nicolas Sonnerat, Tim Green, Louise Deason, Joel Z. Leibo, David Silver, Demis Hassabis, Koray Kavukcuoglu, and Thore Graepel. Human-level performance in 3d multiplayer games with population-based reinforcement learning. *Science*, 364(6443):859–865, May 2019. ISSN 1095-9203. doi: 10.1126/science.aau6249.

[10] Fivos Kalogiannis, Ioannis Anagnostides, Ioannis Panageas, Emmanouil-Vasileios Vlatakis-Gkaragkounis, Vaggos Chatziafratis, and Stelios Stavroulakis. Efficiently Computing Nash Equilibria in Adversarial Team Markov Games. *arXiv e-prints*, art. arXiv:2208.02204, August 2022. doi: 10.48550/arXiv.2208.02204.

[11] Akbir Khan, Timon Willi, Newton Kwan, Andrea Tacchetti, Chris Lu, Edward Grefenstette, Tim Rocktäschel, and Jakob Foerster. Scaling opponent shaping to high dimensional games. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '24, page 1001–1010, Richland, SC, 2024. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 9798400704864.

[12] Joel Z. Leibo, Vinícius Flores Zambaldi, Marc Lanctot, Janusz Marecki, and Thore Graepel. Multi-agent reinforcement learning in sequential social dilemmas. In *Adaptive Agents and Multi-Agent Systems*, 2017.

[13] Joel Z Leibo, Edgar A Dueñez-Guzman, Alexander Vezhnevets, John P Agapiou, Peter Sunehag, Raphael Koster, Jayd Matyas, Charlie Beattie, Igor Mordatch, and Thore Graepel. Scalable evaluation of multi-agent reinforcement learning with melting pot. In *International conference on machine learning*, pages 6187–6199. PMLR, 2021.

[14] Adam Lerer and Alexander Peysakhovich. Maintaining cooperation in complex social dilemmas using deep reinforcement learning, 2018.

[15] Liam Li, Kevin Jamieson, Afshin Rostamizadeh, Ekaterina Gonina, Jonathan Ben-Tzur, Moritz Hardt, Benjamin Recht, and Ameet Talwalkar. A system for massively parallel hyperparameter tuning. *Proceedings of Machine Learning and Systems*, 2:230–246, 2020.

[16] Eric Liang, Richard Liaw, Robert Nishihara, Philipp Moritz, Roy Fox, Ken Goldberg, Joseph Gonzalez, Michael Jordan, and Ion Stoica. Rllib: Abstractions for distributed reinforcement learning. In *International conference on machine learning*, pages 3053–3062. PMLR, 2018.

[17] Michael L. Littman. Markov games as a framework for multi-agent reinforcement learning. In *Proceedings of the Eleventh International Conference on International Conference on Machine Learning*, ICML'94, page 157–163, San Francisco, CA, USA, 1994. Morgan Kaufmann Publishers Inc. ISBN 1558603352.

[18] John F. Nash. Equilibrium points in n-person games. *Proceedings of the National Academy of Sciences*, 36(1):48–49, 1950. doi: 10.1073/pnas.36.1.48.

[19] David Lvovich Nesterov-Rappoport. The evolution of trust: Understanding prosocial behavior in multi-agent reinforcement learning systems. Technical report, 2022.

[20] Andrei Cristian Nica, Tudor Berariu, Florin Gogianu, and Adina Magda Florea. Learning to maximize return in a stag hunt collaborative scenario through deep reinforcement learning. In *2017 19th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC)*, pages 188–195, 2017. doi: 10.1109/SYNASC.2017.00039.

[21] Julien Perolat, Florian Strub, Bilal Piot, and Olivier Pietquin. Learning Nash Equilibrium for General-Sum Markov Games from Batch Data. In Aarti Singh and Jerry Zhu, editors, *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pages 232–241. PMLR, 20–22 Apr 2017.

[22] Alexander Peysakhovich and Adam Lerer. Prosocial learning agents solve generalized stag hunts better than selfish ones. In *Adaptive Agents and Multi-Agent Systems*, 2017.

[23] T.C. Schelling. *The Strategy of Conflict*. Harvard University Press, 1960.

[24] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017.

[25] L. S. Shapley. Stochastic games*. *Proceedings of the National Academy of Sciences*, 39(10): 1095–1100, 1953. doi: 10.1073/pnas.39.10.1095.

[26] John Maynard Smith. Evolution and the theory of games. 1976.

[27] Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, Vinicius Zambaldi, Max Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z. Leibo, Karl Tuyls, and Thore Graepel. Value-decomposition networks for cooperative multi-agent learning based on team reward. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '18, page 2085–2087. International Foundation for Autonomous Agents and Multiagent Systems, 2018.

[28] Attila Szolnoki and Matjaž Perc. Resolving social dilemmas on evolving random networks. *EPL (Europhysics Letters)*, 2009.

[29] Ming Tan. *Multi-agent reinforcement learning: independent vs. cooperative agents*, page 487–494. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1997. ISBN 1558604952.

[30] Jordan Terry, Benjamin Black, Nathaniel Grammel, Mario Jayakumar, Ananth Hari, Ryan Sullivan, Luis S Santos, Clemens Dieffendahl, Caroline Horsch, Rodrigo Perez-Vicente, et al. Pettingzoo: Gym for multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 34:15032–15043, 2021.

[31] Robert van Rooij. The stag hunt and the evolution of social structure. *Studia Logica*, 85: 133–138, 2007.

[32] Oriol Vinyals, Igor Babuschkin, Wojciech M. Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David Choi, Richard Powell, Timo Ewalds, Petko Georgiev, Junhyuk Oh, Dan Horgan, Manuel Kroiss, Ivo Danihelka, Aja Huang, L. Sifre, Trevor Cai, John P. Agapiou, Max Jaderberg, Alexander Sasha Vezhnevets, Rémi Leblond, Tobias Pohlen, Valentin Dalibard, David Budden, Yury Sulsky, James Molloy, Tom Le Paine, Caglar Gulcehre, Ziyun Wang, Tobias Pfaff, Yuhuai Wu, Roman Ring, Dani Yogatama, Dario Wünsch, Katrina McKinney, Oliver Smith, Tom Schaul, Timothy P. Lillicrap, Koray Kavukcuoglu, Demis Hassabis, Chris Apps, and David Silver. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 575:350 – 354, 2019.

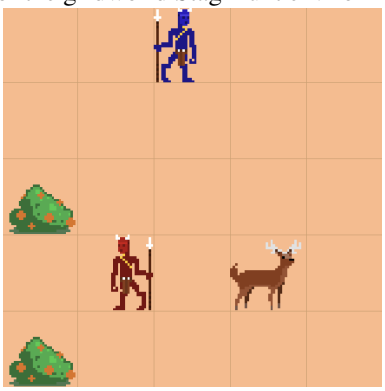
[33] Michael P. Wellman. Methods for empirical game-theoretic analysis. In *AAAI Conference on Artificial Intelligence*, 2006.

[34] Chun-Lei Yang, Ching-Syang Jack Yue, and I. Yu. The rise of cooperation in correlated matching prisoners dilemma: An experiment. *Experimental Economics*, 2007.

[35] Chao Yu, Akash Velu, Eugene Vinitsky, Yu Wang, Alexandre M. Bayen, and Yi Wu. The surprising effectiveness of ppo in cooperative multi-agent games. In *Neural Information Processing Systems*, 2021.

A Gridworld Stag Hunt

Figure 3: Rendering of the gridworld Stag Hunt environment used in this study.



B Hyperparameters

In Table 6, we list the relevant hyperparameters for RLlib's PPO implementation which was employed throughout this study, using the stated hyperparameter values. All parameters not stated were set to None.

Table 6: Hyperparameter values used for training PPO policies throughout this study. The second column includes the range of values used for hyperparameter tuning in Section C.2, where values given in square brackets [x,y,z] represent a discrete choice over values x, y and z, and values in round brackets (x,y) represent a continuous range, sampled according to the stated distribution.

Hyperparameter	IPPO/PPO (Section 5)	IPPO Extended (Section C.2)
clip	0.3	$[0.2, 0.3, 0.4, 0.45]$
discount	0.99	$[0.98, 0.99, 0.995, 0.999]$
training batch size	4000	4000
sgd batch size	128	$[128, 256, 512]$
num_sgd_iter	30	30
kl coeffient	0.2	0.2
kl target	0.2	0.2
entropy coefficient	0	$log_uniform(5 \times 10^{-2}, 3 \times 10^{-1})$
vf clip	10.0	10.0
vf loss coefficient	1.0	$uniform(0.5, 1.0)$
λ (GAE)	1.0	$[0.92, 0.95, 0.98]$
optimizer	Adam	Adam
learning rate	5×10^{-5}	$log_uniform(1 \times 10^{-6}, 5 \times 10^{-2})$

C Environment Complexity and Strategy Convergence

In this section, we provide results of the supplementary experiments we conducted to substantiate our claims in Section 5.1.

C.1 Centralised PPO

To establish a baseline for our earlier results in Section 5.1, which used the IPPO training methodology, we trained a single joint-action policy that controls both agents, labelled PPO. The distinction between IPPO and PPO is that while IPPO agents treat other learners as part of their own environment, PPO controls both agents under a single policy. This distinction allows us to assess the extent to which independent multi-agent interaction is necessary to achieve globally optimal outcomes in the gridworld Stag Hunt.

Our results, presented in Figure 4, further suggest that environment complexity promotes convergence to suboptimal strategies. We find that the same grouping of environments from IPPO (Figure 1) applies to PPO agents. Although the performance of PPO in Group B resembles that of IPPO in Group B, we observe that performance in Group A exhibits significantly larger variance between trials compared to IPPO. Specifically, multiple trials in the same environment, including the fully deterministic environment, **FFF**, show convergence to vastly different strategies. This observation highlights the necessity of independent multi-agent interaction in navigating the social dilemma aspects of our environment, where PPO may be overly influenced by early biases in training runs. Despite this, in Figure 4 we continue to observe that an increase in environment complexity correlates with decreased pay-offs.

C.2 Extended Training in Group B Environments

In Section 5.1, we found that IPPO agents converged to suboptimal strategies in Group B, whilst converging to more optimal strategies in Group A. To investigate whether the observed suboptimal convergence behaviour could be explained through either: inappropriate hyperparameters for the

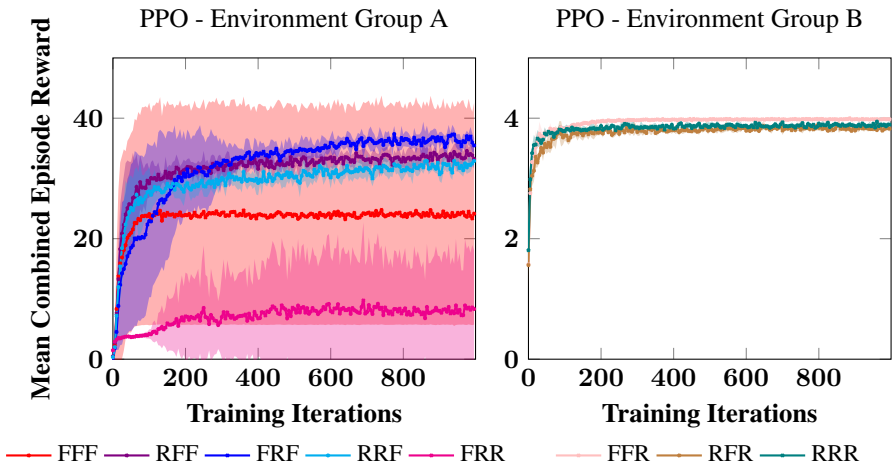


Figure 4: Training performance of PPO across eight environments of varying complexity, with labels detailed Table 4. Environments are grouped by reward patterns: Group A (left) includes **FFF, RFF, FRF, RRF, FRR**, and Group B (right) includes **FFR, RFR, RRR**. Bold lines represent the average performance across five trials and the shading represents ±1 standard deviation from each point.

given environment variants, or insufficient training time to discover more optimal strategies, we conducted an extended training experiment with hyperparameter tuning.

Initially, we trained IPPO agents in each of the group B environments (**FFR, RFR, RRR**) for five trials, each consisting of 1,000 iterations. Each of these trials used HyperOpt's Tree Parzen Estimator (TSE) search algorithm [2] and the Asynchronous Hyperband (ASHA) scheduler [15] to estimate the optimal hyperparameter values for each trial, where the hyperparameter ranges used are listed in the second column of Table 6. Subsequently, we selected the best performing trial in each environment, measured by maximum mean combined episode reward, to extract the optimal set of hyperparameter values for each environment. Finally, we used the set of optimal hyperparameter values to run an additional training experiment for each environment, consisting of 12,000 iterations, i.e., 4.8×10^7 sampled timesteps, representing a tenfold increase over the training duration used in Section 5.1.

Our results in Figure 5, show that IPPO agents continue to converge to the suboptimal strategy previously observed, despite the hyperparameter tuning and extended training duration. This indicates that the cause of this behaviour is likely neither insufficient training time nor unsuitable hyperparameter values. Rather, our results suggest that the convergence behaviour displayed is a direct result of the increased complexity of environments in Group B.

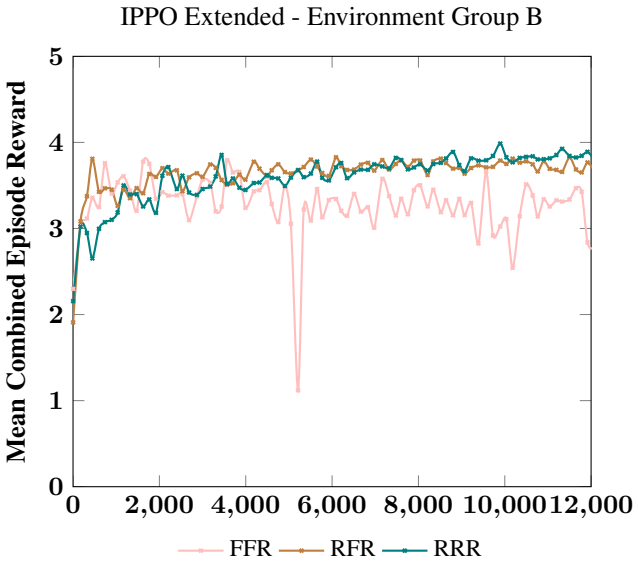


Figure 5: Training performance of IPPO in Group B environments from Section 5.1, using the best hyperparameters found from five trials consisting of 1,000 iterations each.