Heterogeneous Social Value Orientation Leads to Meaningful Diversity in Sequential Social Dilemmas

Udari Madhushani
Princeton University
udarim@princeton.edu

Kevin R. McKee
DeepMind
kevinrmckee@google.com

John P. Agapiou
DeepMind
jagapiou@google.com

Joel Z. Leibo
DeepMind
jzl@google.com

Richard Everett
DeepMind
reverett@google.com

Thomas Anthony
DeepMind
twa@google.com

Edward Hughes
DeepMind
edwardhughes@google.com

Karl Tuyls
DeepMind
karltuyls@google.com

Edgar A. Duéñez-Guzmán
DeepMind
duenez@google.com

ABSTRACT

In social psychology, Social Value Orientation (SVO) describes an individual's propensity to allocate resources between themself and others. In reinforcement learning, SVO has been instantiated as an intrinsic motivation that remaps an agent's rewards based on particular target distributions of group reward. Prior studies show that groups of agents endowed with heterogeneous SVO learn diverse policies in settings that resemble the incentive structure of Prisoner's dilemma. Our work extends this body of results and demonstrates that (1) heterogeneous SVO leads to meaningfully diverse policies across a range of incentive structures in sequential social dilemmas, as measured by task-specific diversity metrics; and (2) learning a best response to such policy diversity leads to better zero-shot generalization in some situations. We show that these best-response agents learn policies that are conditioned on their co-players, which we posit is the reason for improved zero-shot generalization results.

1 INTRODUCTION

In psychology research, Social Value Orientation (SVO) is a cognitive construct reflecting a person's preference for resource allocation between themselves and others [7, 15, 22]. While some individuals may solipsistically focus on maximizing their personal success, others demonstrate different motivations, including maximizing the difference between their own and others' outcomes (a competitive orientation), maximizing collective welfare (a prosocial orientation), or maximizing other peoples' benefit (an altruistic orientation). In artificial intelligence research, various algorithms draw inspiration from these insights in their design or implementation [19, 27]. In reinforcement learning, SVO is an intrinsic motivation that transforms an agent's reward based on a parameterized target distribution between its reward and the reward of others. Recently, studies have investigated the role of SVO in social dilemmas, situations where a group of agents or players interact in ways that involve trade-offs between their self-interest and the collective interest of the group. This research has generated insight into the impact of SVO on the emergence of diverse behaviors and cooperation [19, 20], and partner choice [18]. SVO research has focused primarily on social dilemmas with underlying incentive structures resembling the *Prisoner's dilemma* [26], wherein each player has an incentive to defect, even though they would be better off if they both cooperated.

Sequential social dilemmas are a class of social dilemmas in which the decision-making process of the interacting agents is temporally and spatially extended [13]. Performing well in a sequential social dilemma can be accomplished by considering of long-term consequences, interdependence, and cooperation among group members. Sequential social dilemmas have been widely studied in the context of emergence and maintenance of cooperation [14, 25], inequity aversion [11], partner choice [4, 18], and generalization [1, 20] wherein agents interact with novel co-players in test scenarios.

While environments provide an *extrinsic reward* that can be used to learn a policy, it is often useful to provide agents with an *intrinsic reward* to shape their behavior towards a policy with desirable properties. Intrinsic reward has be used analogously to social preferences in human decision making. In most research on sequential social dilemmas, all players either have no *intrinsic reward*, or they all have the same function (i.e. they have homogeneous social preferences) [14, 31]. However, it has been observed that having a population of agents who differ in their intrinsic reward function (i.e. they have heterogeneous social preferences) can lead to higher levels of cooperation [11]. In [18–20], the authors showed that heterogeneity can produce behavioral diversity in group dilemmas, and in games with incentive structures similar to the Prisoner's dilemma. Other incentive structures have not yet been explored. In addition, the precise interplay between diversity in social preferences and in agent policies, particularly on the mechanisms that enable generalization to novel social partners, remains under-explored.

Diversity in policies has been demonstrated to improve various aspects of agent performance, such as exploration [33], adaptation to environmental changes [3], positive group outcomes [19, 30], generalization to novel co-players [17], and collaboration with humans [29]. One way to quantify diversity is to examine the reward an agent obtains when interacting with different co-players (often

Proc. of the Adaptive and Learning Agents Workshop (ALA 2023), Cruz, Hayes, Wang, Yates (eds.), May 9-10, 2023, Online, https://ala2023.github.io/. 2023.

called *strategic diversity*) [2, 6]. To complement these methods, diversity can also be evaluated through state-action variation, which measures the distribution of state-action pairs that an agent traverses. State-action diversity can be assessed by measuring differences in the state visitation frequency [33], action selection frequency in a given state [20], or differences between state-action trajectories starting from a specific state [17]. Defining an environment agnostic metric based on state-action variation that captures *meaningful* diversity—that is, diversity that has a broader effect on group trajectories—can be challenging. An alternative is to instead use environment-specific measures of diversity, which the researcher can design using their knowledge of specific environment features.

Zero-shot generalization [9, 10, 12, 20, 29] seeks to develop general agents that are capable of successfully interacting with novel agents during test time (i.e., agents not seen during training). In such situations, the policies of the novel agents encountered at test time can be out-of-distribution for the agents, leading to poor coordination in purely cooperative settings [10, 17], and getting exploited in competitive settings [24]. In mixed-motive games, failure to generalize to novel agents can lead to deadweight loss by missing an opportunity to cooperate [12]. Learning a best response to partners/opponents with diverse policies has emerged as a promising approach to zero-shot generalization [29]. The intuition behind this approach is that training with a set of diverse policies decreases the likelihood of encountering out-of-distribution policies at test time. Despite this promise these best response techniques have not yet been applied in a wide range of incentive structures.

In this work, we assess heterogeneous SVO in a range of incentive structures in sequential social dilemmas. We include temporally and spatially extended environments with an underlying structure that resembles several different matrix games: *Prisoner's dilemma*; *Chicken*, where both players have an incentive to choose a risky behavior, but where the worst outcome is if both choose the high risk; and *Stag hunt* wherein players have a safe choice, and an incentive to coordinate on a high-reward strategy that carries a risk of costly miscoordination. Chicken and Stag hunt are equilibrium selection social dilemmas.

We extend the observation that heterogeneous SVO leads to diverse policies to the Chicken- and Stag hunt-like incentive structures. We also show that this diversity, when leveraged via best response, can improve zero-shot generalization in equilibrium selection sequential social dilemmas. We found that best-response agents adapted to partners/opponents with diverse behaviors by learning a conditional policy. However, when the sequential social dilemma was not an equilibrium-selection problem, the learned best response collapsed to one unconditional policy, leading to poor zero-shot generalization

The paper is organized as follows. Section 2 outlines the methodology employed in the paper. In Subsection 2.1, we present the formulation of the N-agent partially observable Markov process used in the paper. Subsection 2.2 describes the Social Value Orientation (SVO) framework and its implementation. In Subsection 2.3, we discuss the various environments used in the study and their characteristics. Subsection 2.4 details the procedure for generating diverse policies in sequential social dilemmas. In Subsection 2.5, we present the process for training a best response agent with a population of agents and evaluating zero-shot generalization performance. Furthermore, we provide a description of the agent's architecture in Subsection 2.5. Section 3 presents the results of the work. In Subsection 3.1 and 3.2, we present the results obtained from generating diverse policies in environments with different incentive structures. In Subsection 3.3, we present the results of zero-shot generalization performance evaluation. Finally, in Section 4, we provide additional discussions and conclusions. The section summarizes the main contributions of the work and discuss potential societal impacts.

2 METHODS

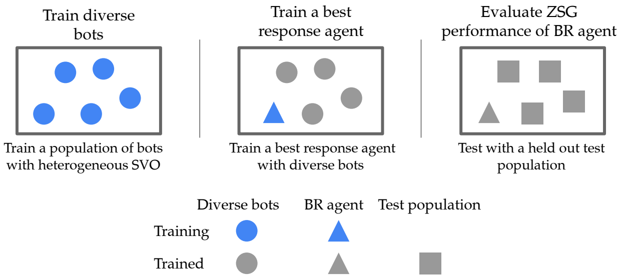

Train diverse bots
Train a best response agent
Evaluate ZSG performance of BR agent
Train a population of bots with heterogeneous SVO
Train a best response agent with diverse bots
Test with a held out test population
Diverse bots
BR agent
Test population
Training
Trained

Figure 1: Overview of the methodology. Blue shapes show agents that are actively being trained, whereas gray ones denote frozen agents (bots). Circles represent the agents trained with diverse SVO, triangles denote a best response agent, and squares denote a held-out set of co-players. Evaluation is zero-shot, meaning the best response agent is frozen (gray triangle) and is evaluated against the held-out bots.

2.1 N-agent POMDP

We consider a multi-agent partially observable Markov decision process defined by the tuple $\left\langle N, \mathcal{S}, \mathcal{A}, \mathcal{R}, P, \gamma \right\rangle$, where N is the number of agents, $\mathcal{S}$ is the joint state space, $\mathcal{A} = \times_{i=1}^{N} \mathcal{A}^i$ is the joint action space, P is the state transition probability distribution, $\mathcal{R}$ is the reward function and γ is the discount factor. This can also be referred to as a partially observable Markov game [16] or a partially observable stochastic game [28]. At each time step t, each agent $i \in 1, \ldots, N$ observes a private (local) observation o_t^i and takes an action a_t^i from a set of actions $\mathcal{A}^i$. The joint action of all agents at time step t is denoted as $a_t = (a_t^1, \ldots, a_t^N)$. The state s_t is not observed directly by the agents, instead the partial observation o_t^i depends on the current state of the environment s_t and the agent's observation function. The observation function for agent i is denoted as $O^i(o_t^i | s_t)$. Each agent i receives a reward r_t^i which is a function of the joint action a_t and the state s_t of the environment. The state of the environment transitions according to a probability distribution $P(s_{t+1} | s_t, a_t)$.

The objective of each agent i is to maximize their cumulative expected discounted reward, over a given finite time horizon, defined as $J^i = \mathbb{E}\left[\sum_{t=0}^{T} \gamma^t r_t^i\right]$, where $\gamma \in [0, 1]$ balances the importance of immediate and future rewards. The agents' policies are defined as the mapping from the agent's observation history to an action, i.e.,

$\pi^i(a_t^i|o_1^i, \cdots, o_t^i)$. The policies are updated using a multi-agent reinforcement learning algorithm that maximizes the agents' objective functions.

2.2 Social Value Orientation

Omitting the dependence on t, let r^i be the reward of agent i. Let $\bar{r}^{-i}$ be the average reward of all the agent except agent i. Then we have

$$\bar{r}^{-i} = \frac{1}{N-1} \sum_{j=1, j \neq i}^{N} r^j.$$

Let θ^i denote the SVO target angle of agent i. Following the definition given in [18], we define the effective reward $\hat{r}^i$ of agent i as

$$\hat{r}^i = r^i \cos(\theta^i) + \bar{r}^{-i} \sin(\theta^i).$$

While sometimes intrinsic rewards are temporally smoothed (e.g.[11]), in this work, effective reward does not include any temporal smoothing. Reintroducing the time step t from the previous section the objective function agent i optimizes for is

$$\hat{J}^i = \mathbb{E}\left[\sum_{t=0}^{T} \gamma^t \hat{r}_t^i\right].$$

2.3 Environments

We provide a brief description of the environments. For all experiments in this paper, we use environments from Melting Pot 2.0 without modifications [1].

“in the matrix” repeated games: The “in the matrix” repeated games are a family of sequential social dilemmas in Melting Pot 2.0 where two-players interact. In the beginning of each episode the environment is initialized according to a given resource layout, and a set of fixed points where players can spawn. The map consists of two types of resources which can be distinguished by their colour; red corresponds to defection and green corresponds to cooperation (see Figure 3). Players can pick up resources by walking over them, and these resources go into a player inventory. Players spawn with one of each resource type in their inventory. After spawning, each player can move around the map, collect resources, and interact with the co-player by firing an interaction beam. When players interact (by one player hitting the other using their interaction beam), each player gets a reward equal to the expected payoff calculated from the inventory counts and environment-specific payoff matrix. The agent who zaps the other agent is considered as the row player. The inventory count of each player defines a mixed strategy where the probability of playing each pure strategy is equivalent to the percentage of the corresponding resource. Let N_r^i and N_g^i denote the inventory count, number of red resources and green resources respectively, for agent $i \in 1, 2$. For each agent i their mixed strategy is given as

$$p = \left[\frac{N_r^i}{N_r^i + N_g^i}, \frac{N_g^i}{N_r^i + N_g^i}\right]$$

Let A be the payoff matrix for the game. Let r_{row} and r_{col} be the reward of row player and column player respectively. Let p_{row} and p_{col} be the mixed strategy probability vector of row player and column player respectively. Then the rewards can be defined as

$$r_{row} = p_{row}^T A p_{col}, \quad r_{col} = p_{col}^T A^T p_{row}$$

These reward calculations correspond to those used in game theory for matrix games and iterated social dilemmas [32].

Stag hunt		Chicken		Prisoner′s dilemma	
4	**0**	**3**	**2**	**3**	**0**
2	**2**	**5**	**0**	**5**	**1**

Figure 2: Payoff matrices for Stag hunt, Chicken and Prisoner's dilemma. The values shown correspond to the payoff of the row player. The payoff of the column player is the transpose of the shown matrix (i.e. the games are symmetric games). Cooperation corresponds to the first row and column. Defection corresponds to the send row and column.

The payoff matrices A used are given in Figure 2. After interacting, players receive their reward from interaction, freeze for 16 steps, and have their inventory counts reset (to one of each resource type). And the end of the 16 steps players disappear and get respawned after 5 steps. Players can have multiple interactions within an episode. Once a resource is picked up, it begins to regenerate after a delay of 10 steps, with a 20% chance of regenerating on each subsequent step. As is standard in Melting Pot 2.0, in each game, there is a 10% chance that the episode will end after every 100 steps, with a minimum of 1000 steps per episode.

Externality mushrooms: Externality mushrooms is a sequential social dilemma where players are immediately affected from prosocial or antisocial behaviors of their co-players. This is a 5-player game where players eat mushrooms in order to receive rewards. Four types of mushrooms grow (in different amounts) on the map: red, green, blue, and orange. Eating a red (“fize”: full internality zero externality) mushroom gives a reward of 1 to the player who consumed the mushroom. Eating a green (“hihe”: half internality half externality) mushroom gives a total reward of 2/5 *to all players.* Eating a blue (“zife”: zero internality full externality) mushroom gives a total reward of 3/4 divided equally among all players *excluding the player who consumed it.* Eating an orange (“nize”: negative internality zero externality) mushroom causes red fize mushrooms to be destroyed, each with probability 0.25, and gives a reward of −0.1 to the player who consumed it. After eating a mushroom, the player who consumed it freezes for the mushroom's digestion time: 0 (red), 10 (green), 15 (blue), and 15 steps (orange). After spawning, a mushroom is removed from the map after its perishing time, i.e. the time it takes for the mushroom to disappear: 200 (red), 100 (green), and 75 steps (blue). Orange mushrooms never perish. Mushrooms respawn from spores depending on consumption of other mushrooms. Eating a red, green, or blue mushroom releases 3 spores for red mushrooms, each spore will spawn a mushroom with probability 0.25. Eating a green or blue mushrooms also releases

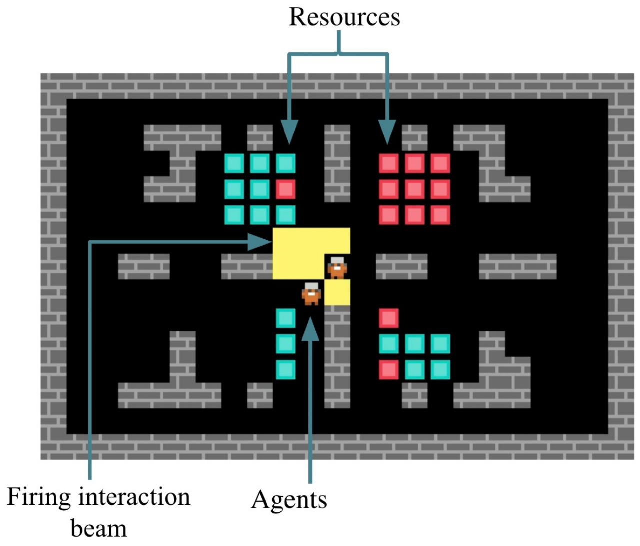


Figure 3: "In the matrix" repeated games. This is a 2-player game where agents can gather 2 types of resources (green corresponding to cooperation, red corresponding to defection). When agents interact (using an interaction beam) they get rewards according to their inventory counts and a game specific payoff matrix. The payoff matrix can be Stag hunt, Chicken or Prisoner's dilemma type payoff matrix

3 spores for green mushrooms which spawn with probability 0.4. Eating a blue mushroom also releases a blue spore which spawn with probability 0.6. Eating an orange mushroom releases a spore for a new orange mushroom which spawns with probability 1. Similar to "in the matrix" repeated games, in Externality mushrooms each episode runs for at least 1000 steps. Following that the episode terminates with probability 0.2 at every 100 steps.

Externality mushrooms has an incentive structure similar to Chicken, where reward is maximized selfishly by consuming red mushrooms while the others are consuming blue or green mushrooms. But if everyone else is eating red mushrooms, the selfish strategy is to eat green mushrooms, as otherwise all mushrooms would be eventually depleted.

2.4 Generating diverse policies in sequential social dilemmas

In the beginning of the training process we define distinct SVO angles for each agent. Each environment has a fixed number of players. We train the agents in a distributed asynchronous manner by initializing 'arenas' to train a population of agents. Arenas run in parallel and each arena is a copy of the environment with the number of players specified for that environment. This is a multi-agent version of A3C [21] that is commonly used for multi-agent reinforcement learning [1]. The Melting Pot evaluation protocol requires sampling of policies with replacement. Training in pure self-play introduces skewed reward incentives by playing with copies of oneself. To alleviate this issue, we set players in each arena to play the game for one episode either in self-play or in population-play (with equal probability). During population-play we sample agents without replacement. We train each agent for 10^9 learner frames.

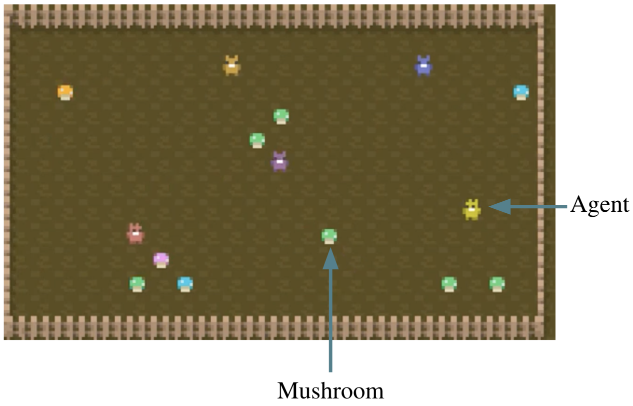


Figure 4: Externality Mushrooms. This is a 5-player sequential social dilemma game with immediate feedback. Agents instantaneously share rewards with others depending on the mushroom they are picking.

2.5 Training a best-response agent and zero-shot generalization performance evaluation

We train a selfish naive learner without intrinsic reward, to best respond against the policies generated using heterogeneous SVO. In order to avoid confusion we use the term *best-response agent* for the training agent, and *SVO bots* for the pre-trained diverse agents trained with heterogeneous SVO values. In each episode the best-response agent plays with a set of SVO bots sampled without replacement. We train the best-response agent for 10^9 learner frames.

Melting Pot 2.0 [1] provides a protocol for evaluating generalization to novel social partners, which are packaged with the suite as a held-out set of co-players in a suite of test scenarios. We measure the performance of the best-response agent using the Melting Pot test protocol.

We use the Melting Pot test scenarios for evaluation in Stag hunt, Chicken, Prisoners' dilemma "in the matrix " repeated games and Externality mushrooms. Test scenario details are provided below.

Test scenarios for "in the matrix" repeated.
Focal player (our best response agent) encounters:

S0: *(cooperator + defector)* either a cooperator or a defector with 0.5 probability each

S1: *(cooperator)* a cooperator

S2: *(defector)* a defector

S3: *(grim strike 1)* a player who starts by cooperating and defect for the rest the episode when focal player defects once

S4: *(grim strike 2)* a player who starts by cooperating and defect for the rest the episode when focal player defects twice

S5: *(tit-for-tat)* a player who plays tit-for-tat

S6: *(tit-for-tat tremble)* a player who a player who plays tit-for-tat and occasionally unconditionally defect. (noisy tit-for-tat)

S7: *(flipping)* a player who cooperate during the first 3 interactions and defect for the rest of the episode

S8: *(corrigible tit-for-tat)* a player who starts with defection and switch to tit-for-tat strategy when best-response agent defects

S9: *(corrigible tit-for-tat tremble)* a player who starts with defection and switch to noisy tit-for-tat strategy when best-response agent defects

Test scenarios for Externality mushrooms:
Focal player (our best response agent) encounters:

S0: *(visiting cooperators)* 4 cooperators

S1: *(visiting defectors)* 4 defectors

2 focal players (in our case 2 copies of best response agent) encounter:

S2: *(resident cooperators)* 3 cooperators

S3: *(resident cooperators)* 3 defectors

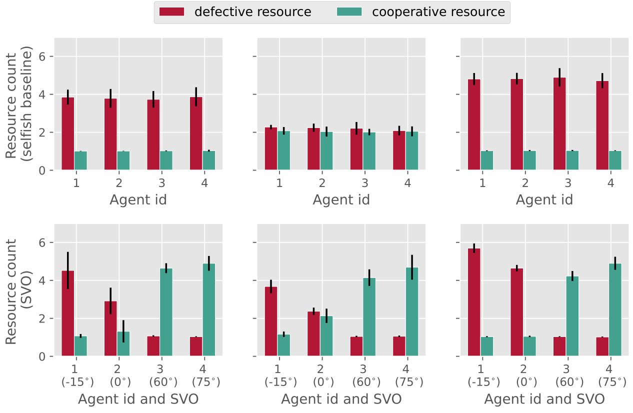
We provide an overview of the end to end methodological pipeline in Figure 1.

2.6 Agent architecture

We trained the agents using the well-established Actor-Critic baseline algorithm proposed in [5], building on the earlier work in [21] named Asynchronous Advantage Actor Critic or A3C.

The neural network of the agent consists of two convolutional layers, a two-layer perceptron, and an LSTM—all separated by ReLU activation functions. The convolutional layers have 16 and 32 output channels, kernel shapes of 8 and 4, and strides of 8 and 1. The perceptron layers are 64 neurons each, and the LSTM layer has 128 units. The policy and baseline for the critic are created by multilayer perceptrons (256 hidden units with ReLU activations) connected to the output of the LSTM.

Representation shaping is achieved through the use of an auxiliary loss and contrastive predictive coding [23], which is used to differentiate between nearby time points via LSTM state representations. PopArt [8] is used to adjust for the different reward scales of the different environments. The optimization method used is RMSProp with a learning rate of 4×10^{-4}, epsilon of 10^{-5}, zero momentum, decay of 0.99, and batch size of 256. The baseline cost for the critic is 0.5, and the entropy regularization cost for the policy is 0.003.



(a) Stag hunt (b) Chicken (c) Prisoner's dilemma

Figure 5: "In the matrix" repeated games. *Diversity of policies of selfish-baseline bots and SVO bots.* Each subfigure shows average inventory counts during evaluation for 4 agents, trained with 50% self-play and 50% population play. The bottom row corresponds to SVO bots with $\theta^i \in \{-15°, 0°, 60°, 75°\}$ and the top row corresponds to selfish-baseline bots. Green and red represents cooperative and defective resource counts respectively. Error bars show the standard deviation of results over 3 random seeds.

3 RESULTS

3.1 Experiment 1: Generating diverse policies in "in the matrix" repeated games

Experimental setup: We consider Stag hunt, Chicken and Prisoners' dilemma "in the matrix" repeated games. For each game we average the results over 3 random seeds. We train four agents with SVO values of $-15°$, $0°$, $60°$, and $75°$, respectively. These values were chosen to cluster around the incentives of competition ($-15°$), selfishness ($0°$) and pro-sociality ($60°$, $75°$). The "in the matrix" repeated games are 2-player games. In addition to SVO bots we also train and freeze a set of selfish-baseline bots (i.e., no intrinsic reward) using the same procedure for comparison.
Finding 1: Heterogeneous SVO bots learn meaningfully diverse policies

We use the inventory count of the bots at the time of interaction as an environment-specific diversity measure. Since the inventory counts define the mixed strategy probability vectors, sufficiently distinct ratios of inventory counts indicate distinct mixed strategies. During evaluation agents play in population-play.

Figure 5 shows the inventory counts for the 4 bots averaged over 500 interactions during evaluation after the completion of training. Top and bottom rows correspond to resource counts of selfish-baseline bots and SVO bots respectively. Figures 5(a), 5(b) and 5(c) correspond to Stag hunt, Chicken and Prisoners' dilemma respectively. The error bars presented in the figure correspond to the average results of 3 independent runs. The results demonstrate that in each game, all 4 selfish-baseline bots have comparable inventory count ratios, suggesting that their policies lack diversity.

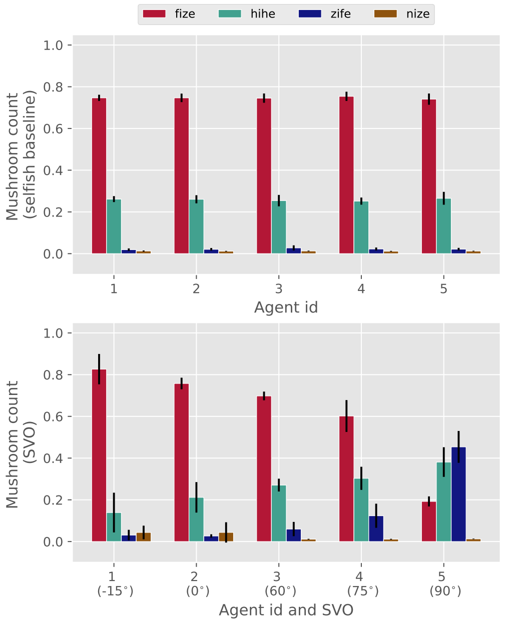


Figure 6: Externality mushrooms. *Diversity of policies of selfish-baseline bots and SVO bots.* Each plot shows average fraction of mushrooms consumed by 5 agents during evaluation, trained with 50% self-play and 50% population play in Externality mushrooms dense game. The bottom row corresponds to SVO agents with $\theta^i \in \{-15°, 0°, 60°, 75°, 90°\}$ and the top row corresponds to selfish-baseline agents. Error bars show the standard deviation of results over 3 random seeds.

Conversely, the 4 SVO bots exhibit varied inventory count ratios, indicating diverse behaviors. For each "in the matrix" repeated game, resource counts correspond to SVO bots with $\theta = [-15°, 0°, 60°, 75°]$, where $\theta^i = \theta[i]$, for $i \in \{1, 2, 3, 4\}$. We denote the cooperative resource counts and defective resource counts using green and red respectively. As the SVO angles increase from $-15°$ to $75°$, the ratio between the red and green resource counts increases, indicating more altruistic behavior.

3.2 Experiment 2: Generating diverse policies in Externality Mushrooms

Experimental setup: Similar to the training process in "in the matrix" repeated game we average the results from 3 random seeds. For each seed we train 5 agents with SVO values of $-15°$, $0°$, $60°$, $75°$, and $90°$, respectively in 50% self-play and 50% population-play. In addition to SVO bots we also train a set of selfish-baseline bots, using the same procedure for comparison.

Finding 2: The results extends to multi-player games with more than 2 players

We show that our method scales to games with more than 2 players. Figure 6 shows that in Externality Mushrooms, agents trained using heterogeneous SVO learn diverse policies. We use the count of mushrooms consumed of each type as the environment-specific diversity metric. The selfish-baseline bots tend to consume mushrooms at similar ratios across different types, whereas the SVO bots consume varying ratios of different mushroom types exhibiting meaningfully diverse behaviors. Agents with low (or negative) SVO consume the selfish mushroom (red), and even the spiteful mushroom (orange), whereas those with high SVO, tend to consume more of the prosocial mushrooms (green and blue).

3.3 Experiment 3: Zero-shot generalization evaluation

We evaluate the zero-shot generalization performance of a learned best response to the SVO bots trained using heterogeneous SVO. **Baselines:** We compare the performance of a learned best response policy for SVO bots with a best response to selfish-baseline bots, Fictitious co-play (FCP, a type of best response that includes also earlier checkpoints of the agents to best respond to) [29], and exploiters (i.e., a best response agent trained on the test scenario directly) [1]. We train one exploiter for each test scenario. To train FCP agents we train a naive learning agent with 3 checkpoints for each bot from a bot population. Here we use the first checkpoint, mid checkpoint and last checkpoint. The mid checkpoint is the time during training where the agent first obtains half of its final reward, of the policies of the bots. We report results for FCP applied to the heterogeneous SVO bots FCP(SVO), as well as to selfish baselines FCP(selfish-baseline). To evaluate zero-shot generalization, we also compare the performance of best response agents with the performance of selfish-baseline agents and random agents.

Experimental setup: We train best-response agents for the selfish-baseline bots and SVO bots. Recall that we trained each type of bots, i.e., selfish-baseline or SVO, for 3 random seeds in this setup. We train a best-response agent for bots from each seed. For each type of test bots we show the average performance evaluation runs correspond to these 3 training runs.

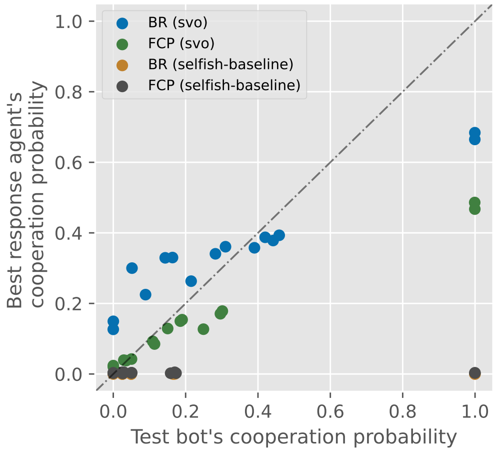


Figure 7: *Comparing how well best-response agents learn conditional policies in Stag hunt in the matrix.*

Finding 3: Best-response agents learn a conditional behaviour

In order to get a better understanding about the learned policies

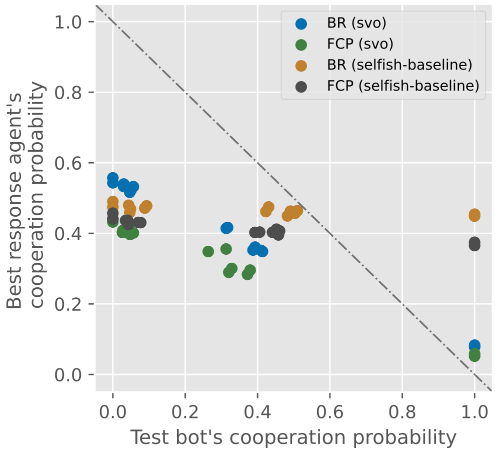


Figure 8: ***Comparing how well best-response agents learn conditional policies in Chicken in the matrix.***

of the best-response agents we analyze the behaviour of the best-response agents during test time. For each test bot, Figures 7 and 8 show the fraction of interactions where the best-response agent cooperated with a bot with respect to the fraction of interactions where the bot cooperated with the best-response agent. Figure 7 corresponds to Stag hunt "in the matrix" repeated and 8 corresponds to Chicken "in the matrix" repeated.

In this analysis we define the best-response agent's interaction as a cooperation when they have higher number of cooperative resources than defective resources in their inventory at the time of interaction. In Stag hunt in the matrix, both agents cooperating, i.e., both agents playing Stag, yields a higher reward, but it is a riskier strategy. Defecting, yields a secure payoff. Both agents cooperating or both defecting are Nash equilibria, that is, there is no incentive to unilaterally deviate from that strategy. An agent who cooperates with a defector gets 0 reward. When trained in Stag hunt in the matrix, selfish-baseline bots learn to defect. The best response to unconditional defectors is defecting. Hence the best-response agents trained with selfish-baseline bots learn to unconditionally defect. In contrast the heterogeneous SVO bot population consists of both defectors and cooperators with different levels of cooperation and defection. Best-response agents training with SVO bots encounter both cooperators and defectors and subsequently learn a conditional policy that tends to cooperate with cooperators and defect with defectors.

In Chicken in the matrix, the two Nash equilibria are for one agent to cooperate (swerve) and the other agent to defect (straight). In this case selfish-baseline agents learn to do both defection and cooperation. Hence the best-response agents trained with selfish-baseline bots also learn to defect and cooperate. However in Figure 8 we see that this behaviour is not conditional. In contrast best-response agents training with SVO bots encounter mostly cooperative and mostly defective bots, leading to best-response agents learning a conditional behavior where they tend to cooperate with defectors and defect against cooperators.

Finding 4: Failure case with Prisoner's dilemma In Prisoner's dilemma in the matrix, the Nash equilibrium is both agents defecting, as a result selfish-baseline agents learn to unconditionally defect. Thus, the best response agents that are trained with selfish agents also learn to defect. Moreover, defection is also a best response to unconditional cooperation. Because SVO bots learned only unconditional strategies (either cooperate or defect), the best response to SVO bots is also to unconditionally defect. Figure 9 illustrates this showing that all the best-response agents are learning to defect regardless of the level of cooperation of their partners.

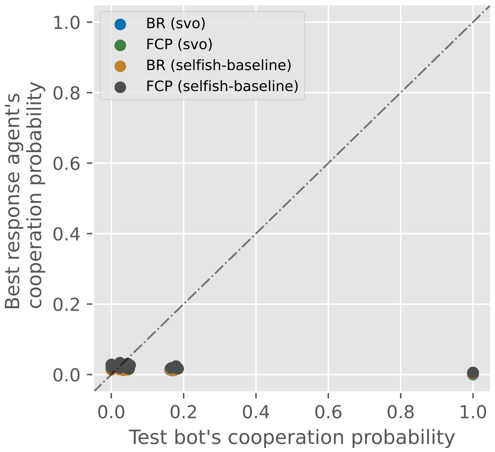


Figure 9: ***Comparing how well best-response agents learn conditional policies in Prisoner's dilemma in the matrix.***

Finding 5: Best response agents perform better in zero-shot generalization Zero-shot generalization performance of the best response agents, selfish-baseline agent, random agent, and exploiters are given in Table 1. Each agent type is run on each test scenario and their average returns are calculated. The score is normalized across agents for each scenario where the best agent receives a score of 1, and the worst a score of 0. The final score of an agent is their average over all scenarios. This is the same method used in Melting Pot [1]. The exploiters and random agent are intended to provide approximate upper and lower bounds for performance across all environments. As expected the table shows that the exploiters achieve the best performance, while the random agent performs the worst. Across all environments at least one best response agent performs better than the selfish-baseline agent indicating that learning a best response improves zero-shot generalization.

On average in the Stag hunt in the matrix scenarios, BR(SVO) outperforms other agents. From Figure 7 we see that BR(SVO) and FCP(SVO) cooperate with unconditional defectors with a small probability. However, in Stag hunt in the matrix, an agent cooperating with a defector or defecting with a defector receives the same reward. Thus when encountering defectors and test bots that are more likely to defect BR(SVO), FCP(SVO) receives comparable rewards to BR(selfish-baseline) and FCP(selfish-baseline). When encountering more cooperative test bots, best response agents that

	BR(SVO)	FCP(SVO)	BR(selfish-baseline)	FCP(selfish-baseline)	selfish-baseline	random	exploiter
Stag hunt ITMR	**0.876**	0.830	0.856	0.847	0.850	0.000	**0.988**
Chicken ITMR	0.696	0.668	**0.745**	0.723	0.723	0.000	**0.958**
Prisoner's dilemma ITMR	0.738	0.702	0.777	**0.783**	0.754	0.000	**1.000**
Externality mushrooms	0.619	0.764	0.612	**0.846**	0.660	0.000	**0.900**

Table 1: Zero-shot generalization performance of best response agents, selfish-baseline agent, random agent and exploiter. The score is calculated by first re scaling the rewards received by each agent such that in each scenario the agent with highest(lowest) reward gets score 1(0) and then averaging over all scenarios for each environment.

are able to adapt to partner behaviours and cooperate with cooperators receive a higher reward. This leads to the higher score of the BR(SVO) agent in Stag hunt in the matrix.

Table 1 shows that in Chicken in the matrix scenarios, BR(selfish-baseline) outperforms other agents. Note that in Chicken in the matrix, an agent cooperating with a defector receives a higher reward than an agent defecting against a defector. From results in Figure 8 we see that when test bots defect with a probability close to 1 all the best response agents cooperate with similar probabilities. Thus in scenarios where test bots are unconditionally defecting all the best response agents obtain comparable performance. However, when test bots are cooperating with about 0.4 probability, BR(selfish-baseline) and FCP(selfish-baseline) cooperate with a higher matching probability compared to BR(SVO) and FCP(SVO) thus leading to better performance for BR(selfish-baseline) and FCP(selfish-baseline). In scenarios where best response agents encounter unconditional cooperators BR(SVO) and FCP(SVO) defect with a probability close to 1 obtaining better performance compared to BR(selfish-baseline) and FCP(selfish-baseline). Since most of the test scenarios consist of defectors or test bots that are more likely to defect, this leads to BR(selfish-baseline) outperforming BR(SVO) and BR(FCP) agents.

Recall that Figure 9 illustrates that all the best-response agents are defecting against all test bots. Thus we expect the performance score of best response agents for Prisoner's dilemma in the matrix given in Table 1 to be similar. However, surprisingly BR(selfish-baseline) and FCP(selfish-baseline) perform better than BR(SVO) and FCP(SVO). We leave investigating this as future work.

In Externality mushrooms, FCP type best response agents perform better than best response agents trained with only final policies of the co-players. This indicates that best response agents that encounter less proficient agents as well as more proficient agents perform better than the best response agents that only encounter proficient agents during training time.

4 DISCUSSION

In this paper we investigated the impact of heterogeneous Social Value Orientation on different incentive structures in sequential social dilemmas. We tested whether the presence of heterogeneous SVO leads to diverse policies and if learning a best response to these policies improves zero-shot generalization. The study found that the presence of heterogeneous SVO does indeed lead to measurable diversity in policies, and this diversity sometimes results in better zero-shot generalization for agents that best respond to them.

The best-response agents achieve better performance by learning a conditional policy that adapts to novel agents during test time. The study also revealed that when the sequential social dilemma is not an equilibrium-selection problem, this method still generates meaningful diversity in policies, but it fails to achieve better zero-shot generalization performance. This occurs because the best response to a diverse set of policies collapses to one unconditional policy that performs poorly when encountering conditional policies during test time.

Our findings have implications for understanding how heterogeneous SVO impacts incentive structures and policy diversity, and how agents can learn to adapt to diverse policies during test time to achieve better zero-shot generalization performance. Our findings provide new insights into the behavior of agents in sequential social dilemmas and highlights the importance of considering the role of heterogeneity in SVO in the design of incentive structures.

We observed that SVO agents were able to learn cooperative policies in all of the environments we tested. This hints at the potential value of using SVO to capture at least some of the aspects necessary to align agents with human values.

ACKNOWLEDGEMENT

Authors thank Daniel Hennes, DJ Strouse, Marc Lanctot, Alexander (Sasha) Vezhnevets, Julien Perolat, Andrea Tacchetti and Ian Gemp for helpful discussion.

REFERENCES

[1] John P Agapiou, Alexander Sasha Vezhnevets, Edgar A Duéñez-Guzmán, Jayd Matyas, Yiran Mao, Peter Sunehag, Raphael Köster, Udari Madhushani, Kavya Kopparapu, Ramona Comanescu, et al. 2022. Melting Pot 2.0. *arXiv preprint arXiv:2211.13746* (2022).

[2] David Balduzzi, Marta Garnelo, Yoram Bachrach, Wojciech Czarnecki, Julien Perolat, Max Jaderberg, and Thore Graepel. 2019. Open-ended learning in symmetric zero-sum games. In *International Conference on Machine Learning*. PMLR, 434–443.

[3] Kenneth Derek and Phillip Isola. 2021. Adaptable agent populations via a generative model of policies. *Advances in Neural Information Processing Systems* 34 (2021), 3902–3913.

[4] Edgar A Duéñez-Guzmán, Kevin R McKee, Yiran Mao, Ben Coppin, Silvia Chiappa, Alexander Sasha Vezhnevets, Michiel A Bakker, Yoram Bachrach, Suzanne Sadedin, William Isaac, et al. 2021. Statistical discrimination in learning agents. *arXiv preprint arXiv:2110.11404* (2021).

[5] Lasse Espeholt, Hubert Soyer, Remi Munos, Karen Simonyan, Vlad Mnih, Tom Ward, Yotam Doron, Vlad Firoiu, Tim Harley, Iain Dunning, et al. 2018. Impala: Scalable distributed deep-rl with importance weighted actor-learner architectures. In *International conference on machine learning*. PMLR, 1407–1416.

[6] Marta Garnelo, Wojciech Marian Czarnecki, Siqi Liu, Dhruva Tirumala, Junhyuk Oh, Gauthier Gidel, Hado van Hasselt, and David Balduzzi. 2021. Pick Your Battles: Interaction Graphs as Population-Level Objectives for Strategic Diversity. In *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems*. 1501–1503.

[7] Donald W Griesinger and James W Livingston Jr. 1973. Toward a model of interpersonal motivation in experimental games. *Behavioral science* 18, 3 (1973), 173–188.

[8] Matteo Hessel, Hubert Soyer, Lasse Espeholt, Wojciech Czarnecki, Simon Schmitt, and Hado van Hasselt. 2019. Multi-task deep reinforcement learning with popart. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 3796–3803.

[9] Hengyuan Hu, Adam Lerer, Brandon Cui, Luis Pineda, Noam Brown, and Jakob Foerster. 2021. Off-belief learning. In *International Conference on Machine Learning*. PMLR, 4369–4379.

[10] Hengyuan Hu, Adam Lerer, Alex Peysakhovich, and Jakob Foerster. 2020. "Other-Play" for Zero-Shot Coordination. In *International Conference on Machine Learning*. PMLR, 4399–4410.

[11] Edward Hughes, Joel Z Leibo, Matthew Phillips, Karl Tuyls, Edgar Dueñez-Guzman, Antonio García Castañeda, Iain Dunning, Tina Zhu, Kevin McKee, Raphael Koster, et al. 2018. Inequity aversion improves cooperation in intertemporal social dilemmas. *Advances in neural information processing systems* 31 (2018).

[12] 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. 2021. Scalable evaluation of multi-agent reinforcement learning with melting pot. In *International Conference on Machine Learning*. PMLR, 6187–6199.

[13] Joel Z Leibo, Vinicius Zambaldi, Marc Lanctot, Janusz Marecki, and Thore Graepel. 2017. Multi-agent Reinforcement Learning in Sequential Social Dilemmas. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*. 464–473.

[14] Adam Lerer and Alexander Peysakhovich. 2017. Maintaining cooperation in complex social dilemmas using deep reinforcement learning. *arXiv preprint arXiv:1707.01068* (2017).

[15] Wim BG Liebrand and Charles G McClintock. 1988. The ring measure of social values: A computerized procedure for assessing individual differences in information processing and social value orientation. *European journal of personality* 2, 3 (1988), 217–230.

[16] Michael L Littman. 1994. Markov games as a framework for multi-agent reinforcement learning. In *Machine learning proceedings 1994*. Elsevier, 157–163.

[17] Andrei Lupu, Brandon Cui, Hengyuan Hu, and Jakob Foerster. 2021. Trajectory diversity for zero-shot coordination. In *International Conference on Machine Learning*. PMLR, 7204–7213.

[18] Kevin R McKee, Xuechunzi Bai, and Susan T Fiske. 2022. Warmth and Competence in Human-Agent Cooperation. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*. 898–907.

[19] Kevin R McKee, Ian Gemp, Brian McWilliams, Edgar A Duèñez-Guzmán, Edward Hughes, and Joel Z Leibo. 2020. Social Diversity and Social Preferences in Mixed-Motive Reinforcement Learning. In *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems*. 869–877.

[20] Kevin R McKee, Joel Z Leibo, Charlie Beattie, and Richard Everett. 2022. Quantifying the effects of environment and population diversity in multi-agent reinforcement learning. *Autonomous Agents and Multi-Agent Systems* 36, 1 (2022), 1–16.

[21] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. 2016. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*. PMLR, 1928–1937.

[22] Ryan O Murphy, Kurt A Ackermann, and Michel JJ Handgraaf. 2011. Measuring social value orientation. *Judgment and Decision making* 6, 8 (2011), 771–781.

[23] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748* (2018).

[24] Nicolas Perez-Nieves, Yaodong Yang, Oliver Slumbers, David H Mguni, Ying Wen, and Jun Wang. 2021. Modelling behavioural diversity for learning in open-ended games. In *International Conference on Machine Learning*. PMLR, 8514–8524.

[25] Alexander Peysakhovich and Adam Lerer. 2018. Consequentialist conditional cooperation in social dilemmas with imperfect information (short workshop version). In *Workshops at the Thirty-Second AAAI Conference on Artificial Intelligence*.

[26] Anatol Rapoport. 1974. Prisoner's dilemma—recollections and observations. In *Game Theory as a Theory of a Conflict Resolution*. Springer, 17–34.

[27] Wilko Schwarting, Alyssa Pierson, Javier Alonso-Mora, Sertac Karaman, and Daniela Rus. 2019. Social behavior for autonomous vehicles. *Proceedings of the National Academy of Sciences* 116, 50 (2019), 24972–24978.

[28] Lloyd S Shapley. 1953. Stochastic games. *Proceedings of the national academy of sciences* 39, 10 (1953), 1095–1100.

[29] DJ Strouse, Kevin McKee, Matt Botvinick, Edward Hughes, and Richard Everett. 2021. Collaborating with humans without human data. *Advances in Neural Information Processing Systems* 34 (2021), 14502–14515.

[30] Zhenggang Tang, Chao Yu, Boyuan Chen, Huazhe Xu, Xiaolong Wang, Fei Fang, Simon Shaolei Du, Yu Wang, and Yi Wu. 2021. Discovering Diverse Multi-Agent Strategic Behavior via Reward Randomization. In *International Conference on Learning Representations*.

[31] Jane X Wang, Edward Hughes, Chrisantha Fernando, Wojciech M Czarnecki, Edgar A Duéñez-Guzmán, and Joel Z Leibo. 2018. Evolving intrinsic motivations for altruistic behavior. *arXiv preprint arXiv:1811.05931* (2018).

[32] Jörgen W Weibull. 1997. *Evolutionary game theory*. MIT press.

[33] Tom Zahavy, Yannick Schroecker, Feryal Behbahani, Kate Baumli, Sebastian Flennerhag, Shaobo Hou, and Satinder Singh. 2022. Discovering policies with domino: Diversity optimization maintaining near optimality. *arXiv preprint arXiv:2205.13521* (2022).