A New Formalism, Method and Open Issues for Zero-Shot Coordination

Johannes Treutlein [1 2 *] **Michael Dennis** [3] **Caspar Oesterheld** [4] **Jakob Foerster** [1 2 5]

Abstract

In many coordination problems, independently reasoning humans are able to discover mutually compatible policies. In contrast, independently trained self-play policies are often mutually incompatible. *Zero-shot coordination* (ZSC) has recently been proposed as a new frontier in multi-agent reinforcement learning to address this fundamental issue. Prior work approaches the ZSC problem by assuming players can agree on a shared learning algorithm but not on labels for actions and observations, and proposes *other-play* as an optimal solution. However, until now, this "label-free" problem has only been informally defined. We formalize this setting as the *label-free coordination (LFC) problem* by defining the *label-free coordination game*. We show that other-play is not an optimal solution to the LFC problem as it fails to consistently break ties between incompatible maximizers of the other-play objective. We introduce an extension of the algorithm, *other-play with tie-breaking*, and prove that it is optimal in the LFC problem and an equilibrium in the LFC game. Since arbitrary tie-breaking is precisely what the ZSC setting aims to prevent, we conclude that the LFC problem does not reflect the aims of ZSC. To address this, we introduce an alternative informal operationalization of ZSC as a starting point for future work.

1. Introduction

In multi-agent reinforcement learning (MARL), variations of the *self-play* (SP) regime (Tesauro, 1994) have been successful in producing superhuman policies for two-player zero-sum games such as chess, go, and poker (Campbell et al., 2002; Silver et al., 2017; Brown & Sandholm, 2018). SP leads to policies that are highly adapted to each other and thus often appear artifical. This is not a problem in two-player zero-sum games as all optimal policies are interchangeable (Nash, 1951), at least when considering optimal opponents.

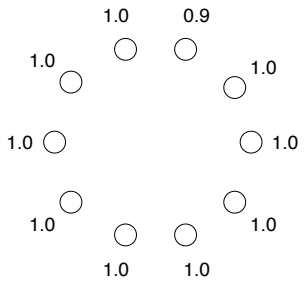


Figure 1. Rewards in the lever coordination game. Levers with equal rewards cannot be distinguished without labels.

In fully cooperative MARL, however, such arbitrary conventions can be undesirable, as they fail when paired with agents that were not present during SP training. For instance, consider a situation in which robots must avoid collisions, by either swerving right or left or slowing down to avoid the other robot. Here, robots trained via SP would randomly learn to swerve either left or right and thus crash half the time at test time when paired in *cross-play (XP)* with agents from independent training runs. Similarly, the arbitrary conventions learned by agents, e.g., in the card-game Hanabi, can prevent successful human-AI coordination (Foerster et al., 2019; Carroll et al., 2019).

This shortcoming of SP in fully cooperative problems motivates the study of the *zero-shot coordination (ZSC) problem*, which Hu et al. (2020) operationalize as finding a general-purpose learning algorithm that allows independently trained agents to coordinate successfully at test time. The independent training is a proxy for the independent decision making that humans have to undertake when solving coordination tasks, while the ability to agree on an algorithm corresponds to having a common high-level approach for solving these problems.

[1]Department of Computer Science, University of Toronto, Toronto, Canada [2]Vector Institute, Toronto, Canada [3]Department of Electrical Engineering and Computer Science, University of California, Berkeley, USA [4]Department of Computer Science, Duke University, Durham, USA [5]Facebook AI Research, USA [*]Part of the work was done as an intern at the Center for Human-Compatible AI, University of California, Berkeley. Correspondence to: Johannes Treutlein <treutlein@cs.toronto.edu>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

More specifically, Hu et al. (2020) assume that players only agree on a learning algorithm, but without sharing labels for observations, actions, and states in the environment. As an example, consider the lever coordination game in Figure 1. There are two agents, each having the choice between 10 different levers. If both agents choose the same lever, they receive rewards as specified in Figure 1. There is 1 lever with a reward of 0.9 and 9 levers with reward 1. If agents pull different levers, the reward is 0. Here, the SP algorithm will learn a joint policy wherein one lever of the nine with a reward of 1 is played by both agents, but such a policy cannot be coordinated on without labels for levers.

Hu et al. (2020) suggest the *other-play* (OP) algorithm as a solution and give an informal optimality proof. The idea behind the algorithm is that it learns policies that are robust to permutations by symmetries of a given problem. For example, in the lever coordination game, the actions leading to a reward of 1 are all symmetric, so a randomly permuted policy will pick them with equal probability. In contrast, the lever leading to a payoff of 0.9 does not have symmetric counterparts, so it is possible to consistently choose that lever. Hu et al. (2020) show that in Hanabi, OP improves performance over SP when playing with real humans, showing the benefits of ZSC to human-AI coordination. However, Hu et al. (2020) do not formalize the "no labels" assumption, but rely on an intuitive notion in their proof. Moreover, Hu et al. (2020)'s proof relies implicitly on the assumption that the OP algorithm's objective has a unique maximizer.

The first goal of this paper is a formalization of Hu et al. (2020)'s problem setting and a theoretical analysis of OP (Section 4). To do so, we introduce *label-free coordination (LFC) games* and define the *label-free coordination problem* as finding an optimal algorithm to recommend to players in a random LFC game (Section 4.2). The formalization provides a rigorous optimality criterion that can be used to compare OP to other algorithms theoretically and to discuss whether Hu et al. (2020)'s formulation is aligned with the goal of human-AI coordination. After introducing a generalized version of the OP algorithm (Section 4.3), we show that OP can be suboptimal in the LFC problem, as Hu et al. (2020)'s assumption of a unique maximizer is not always fulfilled (Section 4.4). Our findings suggest that the 8 point gap between the SP and XP scores for the vanilla version of OP in Hanabi (Hu et al., 2020) may be due to fundamental problem with the algorithm instead of an optimization issue.

Second, we fix this shortcoming by introducing an extension of the algorithm, *OP with tie-breaking*, in which players use a tie-breaking function to consistently break ties between different OP-optimal policies (Section 5). We prove that this extension is an optimal solution to the LFC problem and that all players using the algorithm is a Nash equilibrium of any LFC game.

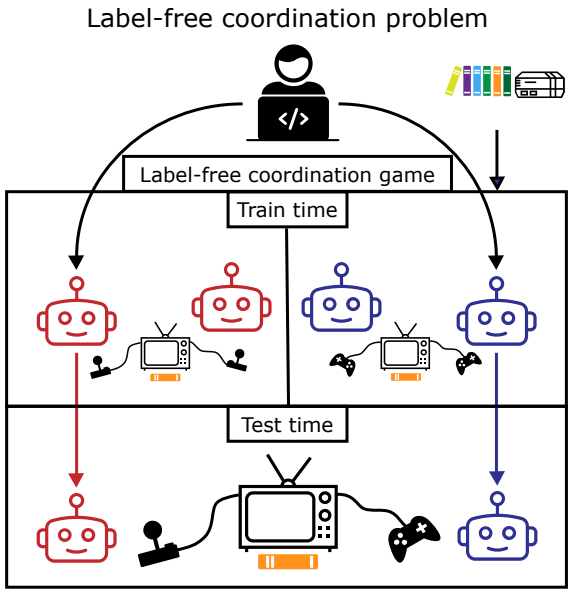


Figure 2. Illustration of the LFC problem. A learning algorithm trains agents independently in a randomly chosen LFC game. The use of different controllers by red and blue agents symbolizes that, while the agents can separately coordinate on policies during train time, they do not know the labels used by agents of the other color and cannot coordinate with them before test time.

Third, we verify our results experimentally in two toy examples (Section 6). Our examples are stylized coordination problems that abstractly model real-world coordination problems.

Fourth, we argue that the operationalization of the ZSC problem by Hu et al. (2020) does not reflect ZSC's aims, and we suggest a new operationalization as a starting point for future work (Section 7). Despite the "no labels" assumption, an optimal algorithm for the problem implements arbitrary tie-breaks. While there may be some settings where it is feasible to pre-coordinate on a tie-breaking function, in general, arbitrary tie-breaking is precisely what the ZSC setting aims to prevent. This shows that algorithmic advances towards Hu et al. (2020) problem formulation are misaligned with the overarching goals of ZSC. We propose an improved informal operationalization in which players are allowed to coordinate only on high-level ideas for a learning algorithm but are prohibited from sharing implementation details such as random seeds, parameters, or code. We leave it to future work to refine and address this revised definition of ZSC.

To save space, we give informal statements and explanations of our theoretical results in the main text. A rigorous treatment of all results, including formal statements and proofs, can be found in the appendix. Our two main results are stated and proven in Appendices E and F.

2. Related work

Game theory A closely related problem to ZSC is the *equilibrium selection problem* in game theory (see Harsanyi & Selten, 1988), which arises when there are different equilibria in a game. The ZSC problem arises when there is an equilibrium selection problem between different optimal policies in a fully cooperative game. Equilibrium selection problems can introduce the additional difficulty that players can have different preferences over the equilibria.

Harsanyi & Selten (1988; see Harsanyi, 1975) introduce a general solution to the equilibrium selection problem in the framework of *standard-form games*, and Herings & Peeters (2003) have adapted it to stochastic games. One property of this solution is invariance to isomorphisms between games, which, like the LFC problem, is based on the idea that a solution should not depend on arbitrary labels (Harsanyi & Selten, 1988, ch. 3.4). A difference to our setting is that we are interested in practical algorithms that can be run on large-scale games, for which computing Harsanyi & Selten (1988)'s solution would be infeasible (cf. Goldberg et al., 2013; Herings & Van Den Elzen, 2002). Another difference is that standard-form games have less structure than Dec-POMDPs. We explicate this and compare our approaches in more detail in Appendix A.

Another game-theoretic approach to coordination problems is based on exogenous information about agents' options. For instance, consider the famous problem "you lost your friend in New York City, where are you going to meet?" (Schelling, 1980; Mehta et al., 1994). Here, additional meaning is attached to each option, independent of dynamics and rewards of the problem, which allows for picking a unique option. Agents might also be able to choose options based on social conventions and norms, such as which side of the street to drive on (Lewis, 2008). In this work, we instead restrict our attention to coordination-problem solutions based only on endogenous information present in the abstract structure of the problem. In many settings, conventions need to be introduced and adapted to within an episode at test time, rather than coordinating them beforehand, e.g., via joint training.

Coordination without joint training Some work looks at coordination problems in settings that do not assume agents are trained together. For instance, Boutilier (1999) introduces a dynamic programming algorithm for fully observable, fully cooperative stochastic games, where no prior coordination between agents is possible. Agents randomize between different optimal actions in a given state until they succeed on coordinating on an optimal joint action. Goldman et al. (2007) consider the Dec-POMDP setting with a cheap-talk channel in which agents cannot pre-coordinate on strategies. They introduce an algorithm in which agents learn to interpret each others' messages and use them to communicate observations and to suggest actions to coordinate on. The idea behind ZSC is to learn joint policies that implement similarly robust strategies as the above approaches, without having to explicitly specify such behavior.

Another related approach is ad-hoc teamwork, wherein the goal is to train an agent to perform well in expectation when subbed into a randomly chosen team of agents (Stone et al., 2010; Barrett et al., 2011; Barrett & Stone, 2015). Ultimately, this amounts to learning a best response to a distribution over team members. However, the team members' policies may themselves be ill-suited for coordination, e.g., if they are obtained via SP. ZSC instead assumes that all agents are optimized for being able to coordinate well, in the absence of pre-established conventions (even though, as mentioned above, optimal ZSC policies in many settings introduce conventions *within* an episode). It can thus find entirely different equilibria, ones that achieve good performance and can be consistently coordinated upon without SP training. As an example, consider the lever coordination game from the introduction. Optimal ad-hoc agents, trained as a best response to a population of SP agents would learn to uniformly randomize between the levers with a payoff of 1.0, while an optimal policy for ZSC always plays the unique lever with a payoff of 0.9.

Alternatively, an agent may be trained using data about other agents' behavior in order to learn a compatible strategy offline (Lerer & Peysakhovich, 2019; Tucker et al., 2020). This again differs from ZSC in that it is concerned with learning a best response instead of optimizing all agents to find non-arbitrary equilibria. Similarly, human-AI coordination can be improved by training agents as a best response to a human model (Carroll et al., 2019). Human-AI coordination is also an aim of ZSC, but we try to uncover general principles behind a human-like learning algorithm instead of learning problem-specific policies from human models.

3. Background

Dec-POMDPs We consider decentralized partially observable Markov decision problems (Dec-POMDPs) (Nair et al., 2003; Oliehoek et al., 2016). A (finite-horizon) Dec-POMDP D is a tuple of a set of agents $\mathcal{N} = \{1, \dots, N\}$ where $N \in \mathbb{N}$, a finite set of states $\mathcal{S}$, a set of joint actions $\mathcal{A} := \prod_{i \in \mathcal{N}} \mathcal{A}_i$, where $\mathcal{A}_i$ is a finite set of actions for agent $i \in \mathcal{N}$, a transition probability kernel $P \colon \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$, where $\Delta(\mathcal{S})$ denotes the set of probability distributions over $\mathcal{S}$, a reward function $\mathcal{R} \colon \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, a set of joint observations $\mathcal{O} := \prod_{i \in \mathcal{N}} \mathcal{O}_i$, where $\mathcal{O}_i$ is a finite set of observations for agent $i \in \mathcal{N}$, an observation probability kernel $O \colon \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{O})$, an initial state distribution $b_0 \in \Delta(\mathcal{S})$, and a horizon $T \in \mathbb{N}_0$. We write $\mathcal{A}^D, \mathcal{A}^E$, etc. to indicate which Dec-POMDP D, E, etc. a set or function belongs to.

In a Dec-POMDP, at time step $1 \leq t \leq T$, the environment is in a state S_t, agent $i \in \mathcal{N}$ receives observations $O_{i,t}$ via $(O_{j,t})_{j\in\mathcal{N}} \sim O(\cdot \mid S_t, A_{t-1})$ and chooses an action $A_{i,t} \sim \pi_i(\cdot \mid \overline{AO}_{i,t})$ according to a *local policy* $\pi_i \in \Pi_i$, where $\overline{AO}_{i,t} := (A_{i,0}, O_{i,1}, \ldots, A_{i,t-1}, O_{i,t})$ is a random variable for agent i's action-observation history at step t, with values $\tau_{i,t} \in \overline{\mathcal{AO}}_{i,t} := (\mathcal{A}_i \times \mathcal{O}_i)^t$.[1] Agents receive a joint reward $R_t := \mathcal{R}(S_t, A_t)$ and the environment transitions into a state $S_{t+1} \sim P(\cdot \mid S_t, A_t)$. The initial state is $S_0 \sim b_0$. We define the set of entire histories, containing tuples of all states, actions, rewards and observations until step T as $\mathcal{H}$ and denote H as a random variable for the entire history.

Denote $\mathbb{P}_\pi$ for a probability measure on a space with the random variables defined above, where agents follow the *(joint) policy* $\pi \in \Pi := \prod_{i\in\mathcal{N}} \Pi_i$, and let $\mathbb{E}_\pi$ be the expectation with respect to that measure. Given a Dec-POMDP D, the *self-play (SP) objective* $J^D \colon \Pi^D \to \mathbb{R}$ of D is defined via $J^D(\pi) := \mathbb{E}_\pi\left[\sum_{t=0}^T R_t\right]$ for $\pi \in \Pi^D$. Here, $J^D(\pi)$ is called the expected return of the joint policy π.

Zero-shot coordination and other-play As explicated in the lever coordination problem, there can be different, incompatible SP-optimal joint policies. A SP algorithm tries to maximize the SP objective and will in general randomly learn any one of these policies. When two such independently trained joint policies $\pi^{(1)}, \pi^{(2)}$ are matched, this can yield bad *XP values* $J(\pi_1^{(1)}, \pi_2^{(2)})$.

To address this shortcoming, Hu et al. (2020) introduce the ZSC problem. In spirit, the problem is to find a general-purpose learning algorithm for fully cooperative environments to train agents that are able to robustly coordinate with their teammates. It is assumed that teammates have also been optimized for ZSC, using a common high-level approach. However, arbitrarily co-adapting agents' policies, e.g., through joint training, is disallowed. Hu et al. (2020) operationalize this as the problem of recommending one learning algorithm to players in a fully cooperative game. Each player trains a joint policy using the algorithm and discards all but one agent. The resulting agents from all players are then evaluated in XP over one episode. Hu et al. (2020) assume players are able to coordinate on a common learning algorithm, but that they are unable to coordinate the learned policies based on common labels for the Dec-POMDP.

Hu et al. (2020) propose the OP algorithm as a method for this setting. The algorithm's main idea is to train a joint policy to achieve high expected return when each local policy is randomly permuted to break symmetries in different ways. The hope is that this results in a unique joint policy, at the cost of a potentially suboptimal expected return. Informally, one can consider *equivalence mappings* $\phi \in \Phi$, which are maps that can be applied to actions, observations, and states, such that applying the map leaves the problem dynamics unchanged. Equivalence mappings can also be applied to a local policy π_i to get a new policy $\phi(\pi_i)$. The OP objective J_{OP} can then be defined via $J_{\text{OP}}(\pi) := \mathbb{E}_{\phi\sim\mathcal{U}(\Phi)}[J(\phi(\pi_1), \phi(\pi_2))]$ for a joint policy $\pi \in \Pi$, where $\mathcal{U}(\Phi)$ is a uniform distribution over Φ.[2] Hu et al. (2020)'s definitions do not apply to Dec-POMDPs in which agents have different action or observation sets, and they do not account for symmetries between agents. We will formally define a more general version of OP in Section 4.3.

Hu et al. (2020) provide an informal proof that both players using OP is an optimal equilibrium in the fully cooperative game described above. Using an appropriate formalization, we show in the next section that this is in general not correct. Contrary to Hu et al. (2020)'s implicit assumption, there can be multiple, incompatible maximizers of the OP objective.

4. Formalism and analysis of OP

4.1. Dec-POMDP isomorphisms

To formalize the no-labels assumption, we introduce isomorphisms between Dec-POMDPs, which formalize the intuition that two Dec-POMDPs may represent the same problem using different labels. Our definition is a trivial generalization of Kang & Kim (2012)'s *automorphisms* over partially observable stochastic games (POSGs) to the concept of an *isomorphism*, but restricted to fully cooperative problems. Analogous definitions of isomorphisms between games have been introduced before in different frameworks (e.g., Harsanyi & Selten, 1988, ch. 3.4; Peleg et al., 1999).

Let D, E be two Dec-POMDPs. Consider a tuple of bijective maps $f := (f_N, f_S, (f_{A_i})_{i\in\mathcal{N}}, (f_{O_i})_{i\in\mathcal{N}})$, where

$$f_N \colon \mathcal{N}^D \to \mathcal{N}^E \tag{1}$$
$$f_S \colon \mathcal{S}^D \to \mathcal{S}^E \tag{2}$$
$$\forall i \in \mathcal{N} \colon \quad f_{A_i} \colon \mathcal{A}_i^D \to \mathcal{A}_{f_N(i)}^E \tag{3}$$
$$\forall i \in \mathcal{N} \colon \quad f_{O_i} \colon \mathcal{O}_i^D \to \mathcal{O}_{f_N(i)}^E. \tag{4}$$

Define a map $f_A \colon \mathcal{A}^D \to \mathcal{A}^E$ via

$$f_A(a) := \left(f_{A_{f_N^{-1}(i)}} a_{f_N^{-1}(i)}\right)_{i\in\mathcal{N}^E}, \tag{5}$$

for $a \in \mathcal{A}^D$, and f_O analogously for observations $o \in \mathcal{O}^D$. That is, in the joint action $f_A(a) \in \mathcal{A}^E$, agent $j = f_N(i) \in \mathcal{N}^E$ (where $i \in \mathcal{N}^D$) plays action $f_{A_i}(a_i) \in \mathcal{A}_j^E$.

[1] In a slight abuse of notation, we use O for both observation probabilities and observation random variable.

[2] This is not Hu et al. (2020)'s original definition, but it is equivalent, by Hu et al. (2020)'s Proposition 2. The version presented here is the one we will generalize later.

Definition 1 (Dec-POMDP isomorphism)**.** Let D, E be Dec-POMDPs such that both have the same horizon $T^D = T^E$, and let f be a tuple of bijective maps as defined in Equations (1)–(4). Then f is an isomorphism from D to E if for any $a \in \mathcal{A}^D$, $s, s' \in \mathcal{S}^D$, and $o \in \mathcal{O}^D$,

$$P^D(s' \mid s, a) = P^E(f_S(s') \mid f_S(s), f_A(a)) \quad (6)$$

$$O^D(o \mid s, a) = O^E(f_O(o) \mid f_S(s), f_A(a)) \quad (7)$$

$$\mathcal{R}^D(s, a) = \mathcal{R}^E(f_S(s), f_A(a)) \quad (8)$$

$$b_0^D(s) = b_0^E(f_S(s)). \quad (9)$$

We denote $\mathrm{Iso}(D, E)$ for the set of isomorphisms from D to E. If that set is non-empty, D and E are called isomorphic.

In the following, we adopt the convention to write fa instead of $f_A(a)$ and fa_i instead of $f_{A_i}a_i$, and we do the same for observations and states. We can also write $f\tau_{i,t}$ for $\tau_{i,t} \in \overline{\mathcal{AO}}_{i,t}$, which is defined as the element-wise application of f. Letting $fr := r$ for rewards, we can also define $f\tau$ for entire histories $\tau \in \mathcal{H}$. One can show that this action of isomorphisms can be inverted by f^{-1} (see Appendix C.2).

A policy $\pi \in \Pi^D$ can be transformed by an isomorphism $f \in \mathrm{Iso}(D, E)$ into a policy for E. We call this operation the *pushforward*, analogously, for instance, to the construction of pushforward measures, as precomposition of π with f^{-1} "pushes" the policy from one Dec-POMDP to another. The definition is analogous to that of applications of symmetries to policies in Hu et al. (2020).

Definition 2. Let D, E be isomorphic Dec-POMDPs, let $f \in \mathrm{Iso}(D, E)$, and let $\pi \in \Pi^D$. Then we define the *pushforward* $f^*\pi \in \Pi^E$ of π by f via

$$(f^*\pi)_j(a_j \mid \tau_{j,t}) := \pi_{f^{-1}j}(f^{-1}a_j \mid f^{-1}\tau_{j,t})$$

for all $j \in \mathcal{N}^E, a_j \in \mathcal{A}_j^E, t \in \{0, \dots, T\}$, and $\tau_{j,t} \in \overline{\mathcal{AO}}_{j,t}^E$. That is, in the joint policy $f^*\pi$, agent $j \in \mathcal{N}^E$ gets assigned the local policy π_i of agent $i := f^{-1}j \in \mathcal{N}^D$, precomposed with f^{-1}.

We show in Appendix C.3 that π and $f^*\pi$ lead to the same expected return in their respective Dec-POMDPs.

For an example of an isomorphism and a pushforward policy, consider the *two-stage lever game*, a stylized coordination problem like the lever coordination game, but with two rounds instead of one. We will use the example in Section 4.4 to show that OP is not optimal in the LFC problem.

Example 3 (Two-stage lever game)**.** Consider the following variant of the lever coordination game, denoted by D. The problem has two agents, $\mathcal{N} = \{1, 2\}$, and rounds ($T = 1$). Each round, each agent pulls a lever, $\mathcal{A}_1 = \mathcal{A}_2 = \{1, 2\}$. If both agents choose the same lever, the reward is 1, otherwise -1. There are two observations, $\mathcal{O}_1 = \mathcal{O}_2 = \{1, 2\}$, and one state. In the second round ($t = 1$), agents observe each other's previous action, so $O_{i,1} = A_{-i,0}$ for $i = 1, 2$.

a	$\mathcal{R}^D(s,a)$
(1,1)	1
(1,2)	-1
(2,1)	-1
(2,2)	1

$\xrightarrow{f^{-1}}$

a	$\mathcal{R}^E(s,a)$
(1,1)	-1
(1,2)	1
(2,1)	1
(2,2)	-1

Figure 3. Reward function $\mathcal{R}^D$ in the two-stage lever game and reward function $\mathcal{R}^E = \mathcal{R}^D \circ f^{-1}$ in an isomorphic problem.

Now consider an isomorphic problem E in which the labels for the actions of the second agent have been switched. A possible isomorphism $f \in \mathrm{Iso}(D, E)$ is one consisting of identity maps, except for f_{A_2}, which switches the two actions of player 2. We give tables of rewards for both the original and the isomorphic problem in Figure 3. Applying f^* to any policy $\pi \in \Pi^D$ creates an equivalent policy $f^*\pi$ for the Dec-POMDP E, in which the actions of agent 2 in D are replaced by the corresponding actions in E.

4.2. The LFC game and problem

We begin by defining LFC games, which we then use to define the LFC problem. We formalize an LFC game as a fully cooperative game between *principals* whose strategies are *learning algorithms*, as defined below. The LFC game is defined for a specific "ground truth" problem D. The game's players, called principals, are the same as the agents in D. Each principal observes a randomly *relabeled* but isomorphic Dec-POMDP and trains a joint policy on that problem using a learning algorithm. The policies are then pushed back to D and evaluated in XP.

For a set of Dec-POMDPs $\mathcal{C}$, let $\Delta(\Pi^D)$ be a set of probability measures over Π^D for $D \in \mathcal{C}$. A learning algorithm for $\mathcal{C}$ is then defined as a map σ that takes in Dec-POMDPs $E \in \mathcal{C}$ and outputs distributions $\sigma(D) \in \Delta(\Pi^D)$, and $\Sigma^{\mathcal{C}}$ is defined as the set of learning algorithms for $\mathcal{C}$. For $\nu \in \Delta(\Pi^D)$ and an isomorphism $f \in \mathrm{Iso}(D, E)$, define the pushforward distribution $f^*\nu := \nu \circ (f^*)^{-1} \in \Delta(\Pi^E)$.

Now fix a Dec-POMDP D. For a given Dec-POMDP, we can create infinitely many different isomorphic problems, as we can use any set of labels, such as natural or real numbers, to define the problem. To describe the process of randomly sampling an isomorphic version of D, then, we restrict ourselves to a specific subset $\mathcal{D}$ of isomorphic Dec-POMDPs in which the sets of states, actions, etc. are of the form $\{1, 2, \dots, k-1, k\} \subseteq \mathbb{N}$. $\mathcal{D}$ is defined as the set of all relabeled Dec-POMDPs, i.e., all problems that are isomorphic to D and have this canonical form (for a rigorous definition, see Appendix C.5). One can interpret sampling from this set as principals coordinating on a canonical way to rep-

resent Dec-POMDPs, but each implementing the problem independently.

Definition 4 (Label-free coordination game)**.** The *Label-free coordination (LFC) game* for D is defined as a game Γ^D where the set of players (here called principals) is $\mathcal{N}^D$, the set of strategies is $\Sigma^{\mathcal{D}}$, and the common payoff for the strategy profile $\boldsymbol{\sigma}_1, \dots, \boldsymbol{\sigma}_N \in \Sigma^D$ is

$$U^D(\boldsymbol{\sigma}) := \mathbb{E}_{D_i \sim \mathcal{U}(\mathcal{D}),\, f_i \sim \mathcal{U}(\mathrm{Iso}(D_i, D)),\, i \in \mathcal{N}} \Big[\mathbb{E}_{\pi^{(j)} \sim f_j^* \boldsymbol{\sigma}_j(D_j),\, j \in \mathcal{N}} \Big[J^D((\pi_k^{(k)})_{k \in \mathcal{N}}) \Big] \Big], \quad (10)$$

where $\mathcal{U}(\mathcal{D})$ is a uniform distribution over $\mathcal{D}$.

We show in Appendix C.6 that the LFC games for two isomorphic Dec-POMDPs are equivalent, up to a potential permutation of the principals.

Turning to the LFC problem, the goal is to find a general learning algorithm to recommend to principals in any LFC game (see Figure 2). We hence formalize the problem here for a distribution over LFC games; however, since this does not change our theoretical analysis, we will only consider LFC problems for single LFC games afterwards. Let a set $\mathcal{C}$ of Dec-POMDPs be given, and denote $\overline{\mathcal{C}} := \bigcup_{E \in \mathcal{C}} \mathcal{D}^E$ where $\mathcal{D}^E$ is the set of all relabeled problems of E. The LFC problem for $\mathcal{C}$ is then defined as the problem of finding one learning algorithm $\sigma \in \Sigma^{\overline{\mathcal{C}}}$ to be used by principals in a randomly drawn game Γ^E for $E \sim \mathcal{U}(\mathcal{C})$.

Definition 5 (Label-free coordination problem)**.** We define the *Label-free coordination (LFC) problem* for $\mathcal{C}$ as the optimization problem

$$\max_{\sigma \in \Sigma^{\overline{\mathcal{C}}}} U^{\mathcal{C}}(\sigma) \quad (11)$$

where $U^{\mathcal{C}}(\sigma) := \mathbb{E}_{E \sim \mathcal{U}(\mathcal{C})} \left[U^E(\sigma, \dots, \sigma) \right]$ for $\sigma \in \Sigma^{\overline{\mathcal{C}}}$. If $\mathcal{C} = \{E\}$, we refer to this as the LFC problem for E and write $U^E(\sigma) := U^{\{E\}}(\sigma) = U^E(\sigma, \dots, \sigma)$.

4.3. Generalization of OP

Here, we introduce a generalized version of the OP algorithm by Hu et al. (2020). It is based on *Dec-POMDP automorphisms*, which are isomorphisms $g \in \mathrm{Aut}(D) := \mathrm{Iso}(D, D)$ and can be thought of as representing symmetries of the problem. Our definition of an automorphism is equivalent to that of Kang & Kim (2012) in the POSG framework[3] and is a generalization of Hu et al. (2020)'s equivalence mappings. Unlike the latter, automorphisms are defined for agents with potentially different action and observation sets, they can consist of different permutations for different agents, and they can incorporate permutations of the agents themselves.

As an example, consider an automorphism $g \in \mathrm{Aut}(D)$ in the two-stage lever game. g is an automorphism if precomposition with g^{-1} does not change the reward function, observation probability kernel, etc. For instance, note that players are symmetric, so g_N can be either the identity or it can switch both agents. In the latter case, applying g^{-1} to joint actions switches the two players' actions, which does not have an effect on the relevant functions.

To define the OP objective, we use the pushforward by automorphisms, which is by definition a self-map on the set of joint policies Π^D. Similarly to equivalence mappings, we can randomly permute agents' local policies by different automorphisms, but we have to take into account potential permutations of agents. To that end, for a profile of automorphisms $\mathbf{g} \in \mathrm{Aut}(D)^{\mathcal{N}}$, we define the joint policy $\mathbf{g}^* \pi := \hat{\pi}$, where the local policy $\hat{\pi}_i$ of agent $i \in \mathcal{N}$ is given by $\hat{\pi}_i := (\mathbf{g}_i^* \pi)_i = \pi_{\mathbf{g}_i^{-1} i}(\mathbf{g}_i^{-1} \cdot \mid \mathbf{g}_i^{-1} \cdot)$.

Definition 6 (Other-play)**.** For a Dec-POMDP D and joint policy $\pi \in \Pi^D$, define

$$J^D_{\mathrm{OP}} \colon \pi \mapsto \mathbb{E}_{\mathbf{g} \sim \mathcal{U}(\mathrm{Aut}(D)^{\mathcal{N}})} \left[J^D(\mathbf{g}^* \pi) \right]. \quad (12)$$

We say that J^D_{OP} is the *other-play (OP) objective* of D, and $J^D_{\mathrm{OP}}(\pi)$ is the *OP value* of $\pi \in \Pi^D$. Given a set of Dec-POMDPs $\mathcal{C}$, we define an OP learning algorithm as any learning algorithm $\sigma^{\mathrm{OP}} \in \Sigma^{\mathcal{C}}$ such that $\mathbb{E}_{\pi \sim \sigma^{\mathrm{OP}}(D)}[J^D_{\mathrm{OP}}(\pi)] = \max_{\pi \in \Pi^D} J^D_{\mathrm{OP}}(\pi)$ for all $D \in \mathcal{C}$.

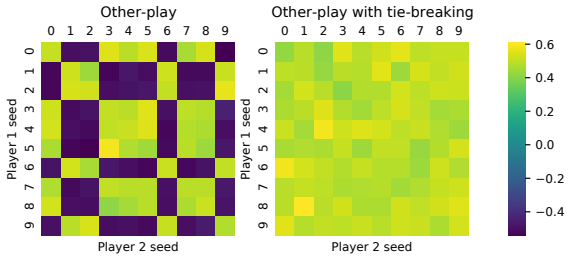
Hu et al. (2020) show that their objective can be maximized in practice by considering a modified Dec-POMDP and applying any learning algorithm for Dec-POMDPs (e.g., Sunehag et al., 2018) to that problem. We show in Appendix D.5 that this does not work for our objective, as optimal policies may need to be stochastic, while in Dec-POMDPs there always exist optimal deterministic policies (Oliehoek et al., 2008, sec. 2.4.4). However, we can still apply a vanilla policy gradient method (see Appendix B.1).

4.4. OP is not optimal in the LFC problem

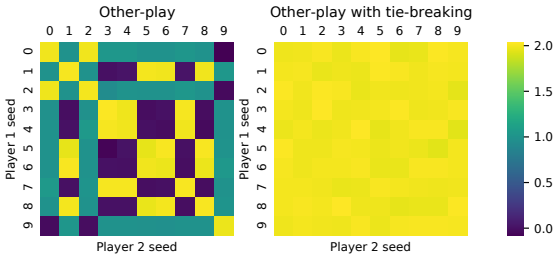
An OP learning algorithm may learn different, potentially incompatible OP-optimal policies in independent training runs. Hence, if the algorithm does not only learn compatible policies, it can be suboptimal in the LFC problem.

To see this, consider the two-stage lever game. In a simple game with one round, such as the lever coordination game, applying symmetries helps avoid arbitrary coordination on one lever. However, this changes in a game with two rounds. Since symmetries are not applied independently to the rounds, but they always apply to the whole episode, agents are able to coordinate in the second round if they coordinated by chance in the first round. This is advantageous

[3] It is a straightforward consequence of Kang & Kim (2012)'s results that the problem of finding Dec-POMDP automorphisms is graph isomorphism–complete.



(a) Two-stage lever game



(b) Asymmetric lever game

Figure 4. Heatmaps indicating XP values between policies from independent training runs. OP is on the left and OP with tie-breaking on the right, in which ties were broken between 32 different seeds. Each value has been averaged over 2048 episodes.

for getting a higher return, but unfortunately, there are two different ways to coordinate in the second round.

Consider the two policies π^R and π^S. In both policies, agents randomize uniformly between both levers in the first round. They also both randomize in the second round if coordination was unsuccessful in the first one. If coordination in the first round was successful, there are two different strategies: in π^R, both agents repeat their respective actions from round one. In π^S, both agents switch to the action they did not play in round one, which is unique, given there are only two levers. We show in Appendix E.1 that these policies are both optimal under OP.

Now suppose one agent chooses a local policy from π^R and the other chooses a local policy from π^S. It is clear that this will yield a suboptimal expected return compared with π^R or π^S as agents will always fail to coordinate in the second round if they coordinated in the first round. Thus, in the LFC problem for the two-stage lever game, if a learning algorithm is not concentrated on only one of π^S or π^R, but instead learns both policies (or potentially equivalent policies in relabeled problems), then that algorithm is suboptimal. We hence have the following result:

Theorem 7 (Informal)**.** *Any learning algorithm that learns both π^R and π^S in the two-stage lever game is an OP learning algorithm, but it is not optimal in the LFC problem for that game.*

5. OP with tie-breaking

To fix OP's shortcoming outlined above, we introduce *OP with tie-breaking*, which is based on the notion of a tie-breaking function that uniquely ranks the different OP-optimal policies in a given problem and thus allows for consistently choosing among them. A tie-breaking function could, for instance, compare the two incompatible policies, π^R and π^S, in the two-stage lever game and choose the one under which actions are more highly correlated, which is π^R. A tie-breaking function has to satisfy certain conditions, e.g., it always must have a unique maximizer, and it must choose equivalent policies in isomorphic Dec-POMDPs. We define OP with tie-breaking as an algorithm that chooses an OP-optimal policy that maximizes a tie-breaking function (see Appendix F.1). We then have the following result:

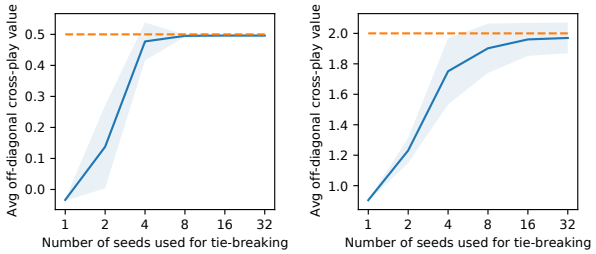
Theorem 8 (Informal)**.** *OP with tie-breaking is optimal in the LFC problem, and all principals using the algorithm is a Nash equilibrium of any LFC game. In particular, the optimal value in the LFC problem for any Dec-POMDP D is equal to the OP value of any OP-optimal policy, i.e.,* $\max_{\sigma \in \Sigma^{\mathcal{D}}} U^D(\sigma) = \max_\pi J^D_{\mathrm{OP}}(\pi)$.

In practice, we can implement OP with tie-breaking by sampling, for a given Dec-POMDP D, $K \in \mathbb{N}$ policies using an OP algorithm σ^{OP} and choosing the policy with the highest tie-breaking value. To compute tie-breaking values, we use a neural network, randomly initialized using a fixed random seed, to map histories deterministically to real numbers. We call these numbers "hash values" in analogy to the hash functions used in many areas of computer science to assign unique keys. The joint policy's tie-breaking value is then calculated as the expected hash value of histories under that policy. A few additional operations (randomly permuting policies, computing normal forms of histories, and summing over agent permutations) are required to ensure this works independently of labels.

Concretely, the tie-breaking function is computed as a Monte Carlo estimate of

$$\frac{1}{N!} \sum_{f_{\mathcal{N}} \in \mathrm{Bij}(\mathcal{N})} \mathbb{E}_{\mathbf{g} \sim \mathcal{U}(\mathrm{Aut}(D)^{\mathcal{N}})} \left[\mathbb{E}_{\mathbf{g}^* \pi} \left[\#(f_{\mathcal{N}}(\iota(H))) \right] \right], \tag{13}$$

where $\#$ is the neural network, $\mathrm{Bij}(\mathcal{N})$ is the set of permutations of $\mathcal{N}$, $\iota(\tau)$ is a *normal form* of the history τ, and for $\iota(\tau) := (s_0, (a_{i,0})_{i \in \mathcal{N}}, r_0, \dots)$, we define $f_{\mathcal{N}}(\iota(\tau)) := (s_0, (a_{f_{\mathcal{N}}^{-1} i, 0})_{i \in \mathcal{N}}, r_0, \dots)$. The normal form $\iota(\tau)$ is computed by replacing the first occurrence of each state, action, or observation in τ by a 0, the second occurrence by a 1, and so on, and repeating the number if an element repeats



(a) Two-stage lever game (b) Asymmetric lever game

Figure 5. Plot of average off-diagonal XP value of the tie-breaking method, using different numbers of seeds for tie-breaking (using one seed is equivalent to OP and represents the baseline). The dashed orange line indicates the theoretical optimum. The shaded area indicates standard deviations across 20 different seeds used for the hash function.

itself in the history. Together with the summation over permutations of agents, this achieves that $\iota(\tau)$ does not depend on particular labels for agents, states, etc. Moreover, consistent tie-breaking is only possible between policies that are randomly permuted by applications of $\mathbf{g}$. We prove in Appendix F.3 that, using a suitable random function $\#$, a modification of the tie-breaking function described here satisfies our formal requirements almost surely.

6. Experiments

MARL training and XP evaluation We use a vanilla policy gradient algorithm (Nguyen et al., 2017; Williams, 1992) to train recurrent neural network policies on the OP objective. We use a randomly initialized feedforward neural network as a hash function and implement a tie-breaking function as described in Section 5. To implement OP with tie-breaking and study the dependency of its performance on the number of policies used for tie-breaking, we apply the tie-breaking function to K learned policies and choose the one with maximal value, for $K \in \{2, 4, 8, 16, 32\}$.

To evaluate a given learning algorithm σ, we simplify the objective of the LFC problem. Instead of using relabeled Dec-POMDPs, we evaluate policies from independent training runs on the same Dec-POMDP, permuted by random automorphisms. That is, we estimate

$$\mathbb{E}_{\pi^{(i)}\sim\sigma(D),\, i=1,2}\Big[\mathbb{E}_{\mathbf{g}_i\sim\mathcal{U}(\mathrm{Aut}(D)),\, i=1,2}\Big[J^D((\mathbf{g}_1^*\pi^{(1)})_1, (\mathbf{g}_2^*\pi^{(2)})_2)\Big]\Big] \quad (14)$$

by computing average off-diagonal XP values between independently trained, permuted policies. We believe that this is a realistic estimate of the objective, as the distributions over policies produced by our learning algorithms should not depend on the used labels (see Appendix C.8).

Table 1. Rewards in the asymmetric lever game, for $t = 0, 1$ on the left, and for $t = 2$ on the right.

	$a_{2,1}$	$a_{2,2}$	$a_{2,3}$
$a_{1,1}$	1	−1	−1
$a_{1,2}$	−1	1	−1
$a_{1,3}$	−1	−1	−1

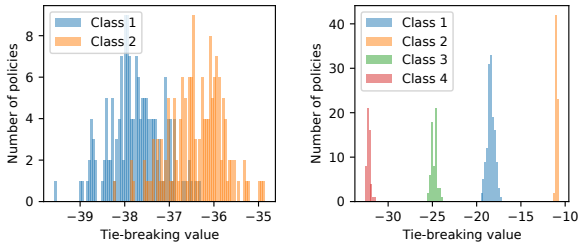
	$a_{2,1}$	$a_{2,2}$	$a_{2,3}$
$a_{1,1}$	0	0	0
$a_{1,2}$	0	0	0
$a_{1,3}$	1	1	1

In both examples, we train 320 policies using different random seeds, and partition them into 10 sets of 32 policies, where the 32 policies can be used for tie-breaking and each set corresponds to one run to be used for computing XP values. We apply OP with tie-breaking using 20 different random seeds for the hash function to explore to what degree the quality of the tie-breaking function depends on the random initialization of the hash network. Additional experimental details and results are described in Appendix B. Code for our experiments can be found at `https://github.com/johannestreutlein/op-tie-breaking`.

Environments As environments, we implement the two-stage lever game introduced in Example 3, as well as the *asymmetric lever game*. In the asymmetric lever game, there are two agents $i = 1, 2$, which can pull one of three levers $\{a_{i,1}, a_{i,2}, a_{i,3}\}$. There are three states $\{0, 1, 2\}$ representing three rounds of this game (i.e., $P(s + 1 \mid s, a) = 1$ for $s \in \{0, 1\}$ and $T = 2$). As before, both agents observe the previous action of the other agent, so $O_{-i,t} = A_{i,t-1}$ for $i = 1, 2$, $t = 1, 2$. The reward function is given in Table 1. The first agent has an extra task in the third round, which makes the agents asymmetric. We choose this example since, unlike the two-stage lever game, it is one where no OP-optimal policy is an intuitively sensible solution to ZSC.

Results All learned policies are close to optimal under the OP objective. In the two-stage lever game, both OP-optimal policies are learned equally well, while there is one dominant policy in the asymmetric lever game. In XP evaluation, OP with tie-breaking outperforms OP, achieving close to optimal XP values in both games.

We display XP matrices, which indicate XP values for any matching of two agents from 10 independent runs, in Figure 4. OP learns incompatible policies in different training runs, whereas policies chosen by OP with tie-breaking appear to be compatible. Average off-diagonal XP values for different numbers of policies used for tie-breaking are plotted in Figure 5. Here, using only one policy for tie-breaking is equivalent to OP, as there is only one policy to choose from. We divide the policies into classes of policies with a high mutual XP score and show a histogram of the tie-breaking values for each class in Figure 6.



(a) Two-stage lever game (b) Asymmetric lever game

Figure 6. Histogram of the tie-breaking values of the learned policies, categorized into classes with mutually high XP values.

7. Open issues

Tie-breaking may be a feasible solution in some coordination problems. For instance, if a central authority can recommend an arbitrary tie-breaking function to all principals, using OP with tie-breaking may provide an easy way to coordinate over a range of different problems, even if joint training is impossible and common labels for problems are unavailable. Moreover, there could be tie-breaking functions based on natural biases that are shared between all principals, such as a simplicity bias. In the two-stage lever game, such a bias could be used to justify the strategy in which agents both repeat their action if they succeeded in coordinating in the first round.

However, our work shows that the current operationalization of ZSC needs to be revised. This is because OP with tie-breaking, an optimal solution to the LFC problem, is ultimately unsatisfactory as a solution to ZSC. First, OP with tie-breaking allows for arbitrary tie-breaking functions. This goes against the spirit of the ZSC problem, which prohibits arbitrary co-adaptation of policies and allows only tie-breaking functions based on plausible, non-arbitrary meta conventions, such as repeating actions that have previously been coordinated on.

Second, sometimes there may be no plausible tie-breaking function; instead, an entirely different policy should be learned. For instance, in the asymmetric lever game, OP-optimal policies choose one agent to switch to a different lever if coordination failed in the first round. But this choice of agent appears arbitrary. Hence, it would be preferable to learn a policy that randomizes if the players could not coordinate in the first round, similar to the policies in the two-stage lever game. Since such a policy is not learned by OP, no appropriate policy can be chosen by OP with tie-breaking. In fact, no optimal solution to the LFC problem would be to able learn this policy, as doing so would lead to lower performance under the LFC problem's objective.

Our results imply that an operationalization of ZSC should preclude principals from sharing not only labels but also any other implementation details. This suggests the following improved setting: principals coordinate on high-level ideas for learning algorithms, but they cannot coordinate on specific implementation details, such as random seeds, parameters, or code. Each principal then implements their algorithm independently and trains an agent on a given randomly sampled environment. As prior coordination between principals is restricted and the algorithm must work in a range of environments, it can only rely on general high-level principles for coordination, not on arbitrary tie-breaking.

Unfortunately, the question of what counts as an implementation detail versus a high-level idea for an algorithm is vague, and thus, unlike the LFC problem, this operationalization does not have a straightforward formalization. Nevertheless, it better suits ZSC's spirit and thus serves as an improved problem setting. We leave it to future work to address and refine this new operationalization.

8. Conclusion and future work

We formalized Hu et al. (2020)'s operationalization of ZSC as the LFC problem, showed that OP is not optimal in the problem, and introduced an extension, OP with tie-breaking, that is optimal. We supported our theoretical results experimentally in two toy examples. Lastly, based on our findings, we concluded that the LFC problem is misaligned with ZSC's aims and suggested a revised intuitive operationalization of ZSC.

More work is required to devise formalisms and algorithms that suit this revised operationalization. One avenue may be different symmetry concepts (e.g., Harsanyi & Selten, 1988; von Neumann & Morgenstern, 1947; Nash, 1951; Peleg et al., 1999; Casajus, 2001) using weaker notions of equivalence. Unfortunately, as we show in Appendix A, considering Dec-POMDPs as standard-form games and applying symmetries in that formalism leads to too little possible coordination between agents. This raises the question whether there is a "Goldilocks" concept obviating the need for tie-breaks while allowing for maximal coordination.

Acknowledgements

We are grateful to Leon Lang and Lennart Stern for help with proofs and writing, and Christian Schroeder de Witt for assistance with the PyMARL code repository. Johannes Treutlein would like to thank his readers Prof. Gitta Kutyniok and Prof. Reinhold Schneider at Technical University of Berlin, where he submitted a version of this paper as a BSc thesis. During his work on this paper, Johannes Treutlein was supported by Open Philanthropy, the Berkeley Existential Risk Initiative, and the Center on Long-Term Risk. Caspar Oesterheld is thankful for support by the National Science Foundation under Award IIS-1814056.

References

Alon, N. and Spencer, J. H. *The probabilistic method*. John Wiley & Sons, Hoboken, NJ, 4th edition, 2016.

Barrett, S. and Stone, P. Cooperating with unknown teammates in complex domains: A robot soccer case study of ad hoc teamwork. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, 2015.

Barrett, S., Stone, P., and Kraus, S. Empirical evaluation of ad hoc teamwork in the pursuit domain. In *AAMAS*, pp. 567–574, 2011.

Boutilier, C. Sequential optimality and coordination in multiagent systems. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence - Volume 1*, pp. 478–485, 1999.

Brown, N. and Sandholm, T. Superhuman AI for heads-up no-limit poker: Libratus beats top professionals. *Science*, 359(6374):418–424, 2018.

Campbell, M., Hoane Jr, A. J., and Hsu, F.-h. Deep Blue. *Artificial Intelligence*, 134(1-2):57–83, 2002.

Carroll, M., Shah, R., Ho, M. K., Griffiths, T., Seshia, S., Abbeel, P., and Dragan, A. On the utility of learning about humans for human-AI coordination. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Casajus, A. Weak isomorphisms of extensive games. In *Focal Points in Framed Games: Breaking the Symmetry*, pp. 55–90. Springer Berlin Heidelberg, Berlin, Heidelberg, 2001.

Emmons, S., Oesterheld, C., Critch, A., Conitzer, V., and Russell, S. Symmetry, equilibria, and robustness in common-payoff games. In *3rd Games, Agents, and Incentives Workshop (GAIW 2021). Held as part of the Workshops at the 20th International Conference on Autonomous Agents and Multiagent Systems.*, May 2021.

Foerster, J., Farquhar, G., Afouras, T., Nardelli, N., and Whiteson, S. Counterfactual multi-agent policy gradients. In *AAAI Conference on Artificial Intelligence*, 2018.

Foerster, J., Song, F., Hughes, E., Burch, N., Dunning, I., Whiteson, S., Botvinick, M., and Bowling, M. Bayesian action decoder for deep multi-agent reinforcement learning. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 1942–1951, 2019.

Gibbons, R. S. *Game theory for applied economists*. Princeton University Press, Princeton, NJ, 1992.

Glicksberg, I. L. A further generalization of the Kakutani fixed point theorem, with application to Nash equilibrium points. *Proceedings of the American Mathematical Society*, 3(1):170–174, 1952.

Goldberg, P. W., Papadimitriou, C. H., and Savani, R. The complexity of the homotopy method, equilibrium selection, and Lemke-Howson solutions. *ACM Transactions on Economics and Computation*, 1(2):1–25, 2013.

Goldman, C. V., Allen, M., and Zilberstein, S. Learning to communicate in a decentralized environment. *Autonomous Agents and Multi-Agent Systems*, 15(1):47–90, 2007.

Harsanyi, J. C. The tracing procedure: a Bayesian approach to defining a solution for n-person noncooperative games. *International Journal of Game Theory*, 4(2):61–94, 1975.

Harsanyi, J. C. and Selten, R. *A general theory of equilibrium selection in games*. The MIT Press, Cambridge, MA, 1988.

Herings, P. J.-J. and Peeters, R. J. Equilibrium selection in stochastic games. *International Game Theory Review*, 5 (4):307–326, 2003.

Herings, P. J.-J. and Van Den Elzen, A. Computation of the Nash equilibrium selected by the tracing procedure in n-person games. *Games and Economic Behavior*, 38(1): 89–117, 2002.

Hu, H., Lerer, A., Peysakhovich, A., and Foerster, J. "Other-play" for zero-shot coordination. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 4399–4410, 2020.

Kang, B. K. and Kim, K.-E. Exploiting symmetries for single-and multi-agent partially observable stochastic domains. *Artificial Intelligence*, 182:32–57, 2012.

Kuhn, H. W. Extensive games and the problem of information. In *Contributions to the Theory of Games: Volume II*, volume 28 of *Annals of Mathematics Studies*, pp. 193–216. Princeton University Press, Princeton, NJ, 1953.

Lerer, A. and Peysakhovich, A. Learning existing social conventions via observationally augmented self-play. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pp. 107–114, 2019.

Lewis, D. *Convention: A philosophical study*. John Wiley & Sons, New York, NY, 2008.

Maschler, M., Solan, E., and Zamir, S. Behavior strategies and Kuhn's theorem. In *Game Theory*, pp. 219–250. Cambridge University Press, Cambridge, UK, 2013.

Mehta, J., Starmer, C., and Sugden, R. The nature of salience: An experimental investigation of pure coordination games. *The American Economic Review*, 84(3): 658–673, 1994.

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D., and Kavukcuoglu, K. Asynchronous methods for deep reinforcement learning. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 1928–1937, 2016.

Nair, R., Tambe, M., Yokoo, M., Pynadath, D., and Marsella, S. Taming decentralized POMDPs: Towards efficient policy computation for multiagent settings. In *IJCAI'03: Proceedings of the 18th International Joint Conference on Artificial intelligence*, pp. 705–711, 2003.

Nash, J. F. Non-cooperative games. *Annals of Mathematics*, 54(2):286–295, 1951.

Nguyen, D. T., Kumar, A., and Lau, H. C. Policy gradient with value function approximation for collective multiagent planning. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Oliehoek, F., Vlassis, N., et al. Dec-POMDPs and extensive form games: equivalence of models and algorithms. IAS technical report IAS-UVA-06-02, University of Amsterdam, Intelligent Systems Lab, Amsterdam, The Netherlands, 2006.

Oliehoek, F. A., Spaan, M. T., and Vlassis, N. Optimal and approximate Q-value functions for decentralized POMDPs. *Journal of Artificial Intelligence Research*, 32:289–353, 2008.

Oliehoek, F. A., Amato, C., et al. *A concise introduction to decentralized POMDPs*. Springer, 2016.

Osborne, M. J. and Rubinstein, A. *A course in game theory*. MIT Press, Cambridge, MA, 1994.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*. 2019.

Pearl, J. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, New York, NY, 2nd edition, 2009.

Peleg, B., Rosenmüller, J., and Sudhölter, P. The canonical extensive form of a game form: Symmetries. In Alkan, A., Aliprantis, C. D., and Yannelis, N. C. (eds.), *Current Trends in Economics*, pp. 367–387. Springer, 1999.

Rotman, J. J. *An Introduction to the Theory of Groups*. Springer, New York, NY, 2012.

Samvelyan, M., Rashid, T., de Witt, C. S., Farquhar, G., Nardelli, N., Rudner, T. G. J., Hung, C.-M., Torr, P. H. S., Foerster, J., and Whiteson, S. The StarCraft Multi-Agent Challenge. arXiv preprint arXiv:1902.04043, 2019.

Schelling, T. C. *The Strategy of Conflict*. Harvard University Press, Cambridge, MA, 1980.

Schulman, J., Abbeel, P., and Chen, X. Equivalence between policy gradients and soft Q-learning. arXiv preprint arXiv:1704.06440, 2017.

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., Chen, Y., Lillicrap, T., Hui, F., Sifre, L., van den Driessche, G., Graepel, T., and Hassabis, D. Mastering the game of go without human knowledge. *Nature*, 550(7676):354–359, 2017.

Stone, P., Kaminka, G., Kraus, S., and Rosenschein, J. Ad hoc autonomous agent teams: Collaboration without precoordination. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 24, 2010.

Sunehag, P., Lever, G., Gruslys, A., Czarnecki, W. M., Zambaldi, V., Jaderberg, M., Lanctot, M., Sonnerat, N., Leibo, J. Z., Tuyls, K., and Graepel, T. 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*, pp. 2085–2087, 2018.

Sutton, R. S. and Barto, A. G. *Reinforcement learning: An introduction*. MIT Press, Cambridge, MA, 2018.

Tesauro, G. TD-Gammon, a self-teaching backgammon program, achieves master-level play. *Neural Computation*, 6 (2):215–219, 1994.

Tucker, M., Zhou, Y., and Shah, J. Adversarially guided self-play for adopting social conventions. arXiv preprint arXiv:2001.05994, 2020.

von Neumann, J. and Morgenstern, O. *Theory of Games and Economic Behavior*. Princeton University Press, Princeton, NJ, 1947.

Williams, D. *Probability with Martingales*. Cambridge University Press, Cambridge, UK, 1991.

Williams, R. J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3-4):229–256, 1992.