Who Needs to Know? Minimal Knowledge for Optimal Coordination

Niklas Lauffer [1]   **Ameesh Shah** [1]   **Micah Carroll** [1]   **Michael Dennis** [1]   **Stuart Russell** [1]

Abstract

To optimally coordinate with others in cooperative games, it is often crucial to have information about one's collaborators: successful driving requires understanding which side of the road to drive on. However, not every feature of collaborators is *strategically relevant*: the fine-grained acceleration of drivers may be ignored while maintaining optimal coordination. We show that there is a well-defined dichotomy between strategically relevant and irrelevant information. Moreover, we show that, in dynamic games, this dichotomy has a compact representation that can be efficiently computed via a Bellman backup operator. We apply this algorithm to analyze the strategically relevant information for tasks in both a standard and a partially observable version of the Overcooked environment. Theoretical and empirical results show that our algorithms are significantly more efficient than baselines. Videos are available at https://minknowledge.github.io.

1. Introduction

When designing a policy for a cooperative multi-agent setting, it is often critical to have some idea of how one's co-players will behave. A policy for a self driving car must take into account various driving norms such as which side of the street other cars will drive on, and how to interpret stop light signals.

While there are many relevant features of co-player behavior that a policy designer must keep in mind, there are often many more irrelevant features that can be safely ignored. It is unnecessary for a self driving car to know the final destination of every other car, the current positions or trajectories of far away cars, or their idiosyncratic driving behaviors.

In such settings, it can be useful to separate the *strategically*

relevant information from the *strategically irrelevant* information. If much of the information is irrelevant, it's easy to imagine how this could lead to significant increases in efficiency for finding optimal policies. For example, this could allow a focused effort on few-shot or zero-shot adaptation to co-players (Zand et al., 2022; Albrecht & Stone, 2019; Stone et al., 2010; Hu et al., 2021) or more efficient DecPOMDP planning algorithms (Szer & Charpillet, 2006; Seuken & Zilberstein, 2007). In order to leverage these benefits, we build the theory, data structures, and algorithms required to distinguish between relevant and irrelevant information.

We formalize the idea of the strategically relevant information via the idea of *strategic ambiguity*, described in Section 4. We show that this leads to a uniquely well-defined dichotomy between the strategically relevant and irrelevant information, captured by the *strategic equivalence relation* (SER) which defines two co-player policies as being *strategically equivalent* if and only if they have the same set of best-response policies.

We show that this strategic equivalence relation can be efficiently computed and stored in Section 5. In fact, we find that our proposed algorithm for computing these strategic equivalence relations has better computational complexity than the state of the art for finding best-response policies in DecPOMDPs.

We summarize the contributions of this paper as follows:

1. We formalize the dichotomy between strategically relevant and irrelevant information by introducing the concept of a strategic equivalence relation (SER).

2. We demonstrate and prove that SERs have a recursive substructure in dynamic games and show how they can be stored compactly in the form of a directed acyclic graph.

3. We provide novel algorithms for computing SERs in both fully-observed and partially-observed common-payoff stochastic games. These algorithms also have notable applications in more efficiently solving DecPOMDPs.

4. We demonstrate what SERs look like in various problems and use them to analyze the levels of coordination required in *Overcooked*, a popular benchmark for testing coordination in the AI literature.

---

[1]Department of Electrical Engineering and Computer Science, University of California, Berkeley, CA, USA. Correspondence to: Niklas Lauffer <nlauffer@berkeley.edu>.

2. Related Work

There have been several lines of research towards getting AI agents to optimally coordinate in fully-cooperative multi-agent problems. Many works have explored designing multi-agent reinforcement learning methods to train agents in a centralized way to converge on Nash equilibrium in common-payoff games (Zhang et al., 2019; Foerster et al., 2016; Jin et al., 2022; Wang & Sandholm, 2002). Others have explored the setting of *ad-hoc teamwork*, in which agents aim to coordinate with teammates from a target population (Barrett & Stone, 2015; Suriadinata et al., 2021), and *zero-shot coordination*, where agents need aim to coordinate with arbitrary teammates without any prior coordination (Hu et al., 2021; Treutlein et al., 2021; Muglich et al., 2022). Our work aims to accelerate these lines of research by providing a framework that focuses on the strategically relevant aspects of the coordination problem.

A distinct line of research has been in the direction of analyzing and understanding *conventions* that agents use to overcome coordination problems (Lewis, 1975; Hadfield-Menell et al., 2018). We show that strategic equivalence relations naturally divide policies into the different conventions that agents could follow in a problem. Various measures (Fontaine et al., 2021; Fontaine & Nikolaidis, 2021) have been developed to analyze the levels of coordination required in multiagent problems. However, all of these approaches are based on simulating agents' behaviors rather than directly analyzing the problem.

Most close to our line of research are some works focused on understanding what the relevant pieces of information are in different decision making problems. Pynadath & Marsella (2007) discuss the idea of minimal models of belief that agents need to solve decision making problems. Oliehoek et al. (2021) instead focus on how agents locally influence each other in factored partially observable stochastic games. Several works explore settings in which games have preexisting *influence diagrams* that describe which decision variables are dependent on each other (Koller & Milch, 2003; Kearns et al., 2013; Mura, 2000). In contrast, our framework for computing strategically relevant information requires no preexisting assumptions or knowledge about how agents interact or influence each other within the decision making problem.

We find ourselves in the setting of cooperative (common-payoff) partially observable stochastic games, also known as decentralized partially observable Markov decision processes (DecPOMDPs). Our algorithms pull ideas from existing work on solving DecPOMDPs (Nair et al., 2003; Szer & Charpillet, 2006; Seuken & Zilberstein, 2007; Dibangoye et al., 2016). We also draw inspiration from Fagin et al. (2004) and model knowledge as *possible worlds*. That is, we represent the knowledge that agents have about co-policies through the subset of possible co-policies that they cannot tell apart.

3. Preliminaries

We consider the setting in which agents play in a common-payoff *stochastic game*, also known as a *decentralized partially observable Markov decision process* (DecPOMDP).

Definition 3.1. A *decentralized partially observable Markov decision process* is a tuple $(M, \mathcal{S}, \mathcal{A}, R, \gamma, P, \Omega, \mathcal{O})$ whose elements are defined as follows.

- $M = \{1, \ldots, m\}$ is a set of agents.
- $\mathcal{S}$ is a set of states.
- $\mathcal{A} = \mathcal{A}_1 \times \cdots \times \mathcal{A}_m$ is the space of joint actions. For ease of notation, we assume without loss of generality that the action set is identical across states.
- $R : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ is the common reward function.
- γ is the discount factor.
- $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ is the transition function that satisfies $\sum_{s' \in \mathcal{S}} P(s, a, s') = 1$.
- $\Omega = \Omega_1 \times \cdots \times \Omega_m$ is the joint observation space.
- $\mathcal{O} : \mathcal{S} \times \mathcal{A} \times \Omega \to [0, 1]$ is the observation function.

A DecPOMDP is *fully-observed* if $\Omega_i = \mathcal{S}$, i.e., agents always observe the current state. We only consider *finite-horizon* DecPOMDPs with episode length T.

A stationary policy for agent i in a DecPOMDP specifies a distribution over actions $\mathcal{A}_i$ at every state for every possible history $h_i^t = [(a_i^1, o_i^1), (a_i^2, o_i^2), \ldots, (a_i^t, o_i^t)]$ of play. We often interpret policies as *trees*, where the nodes are distributions over actions and the edges are observations. We say a policy is *pure* if its choice of distribution over actions is deterministic everywhere. A *Markov policy* is a policy where the action distribution depends only on the latest observation (or state). Let Π denote the joint policy space for all agents, Π_i denote the policy space of agent i, and Π_{-i} denote the *co-policy* space, the joint policy space of the co-players (all agents other than i).

Game theory. As an abuse of notation, we will sometimes write $R(\pi)$ or $R(\pi_i, \pi_{-i})$ to denote the expected return of joint policy $\pi = (\pi_i, \pi_{-i})$.

Definition 3.2. The *best-response function* for agent i is the set-valued function $\text{BR}_i : \Pi_{-i} \to \mathcal{P}(\Pi_i)$ such that $\text{BR}_i(\pi_{-i}) = \arg\max_{\pi_i}\{R(\pi_i, \pi_{-i})\}$, denoting the set of policies agent i could play to maximize payoff in response to the co-policy π_{-i}. The individual best-response functions for each agent can be combined to form the joint best-response function BR for all agents, where $\text{BR}(\pi) = (\text{BR}_1(\pi_{-1}), \text{BR}_2(\pi_{-2}), \ldots, \text{BR}_m(\pi_{-m}))$.
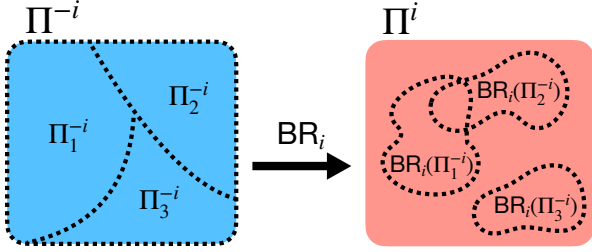
Figure 1: An abstract visualization of how the best-response function partitions the co-policy space into strategic equivalence classes. The best-response function maps the strategic equivalence classes from the co-policy space Π^{-i} (left) to their best-response subset of the policy space Π^i (right).

A Markov policy π is a Nash equilibrium starting in state s if $\forall i \in [m]$,

$$Q^{\pi-i}(s, \pi(s)) \geq Q^{\pi-i}(s, a_i, \pi_{-i}(s)), \quad \forall a \in \mathcal{A}. \quad (1)$$

where $Q^{\pi-i}$ is the Q-function induced by the co-policy π_{-i}. In our context, a joint policy π is a *subgame perfect equilibrium* if the one-stage payoff game induced by the Q-function $Q^{\pi-i}$ is a Nash equilibrium for all $s \in S$. We call the policies $\bigcup_{\pi_{-i} \in \Pi_{-i}} \mathrm{BR}(\pi_{-i})$, the set of *best-response policies* for player i.

4. Strategic relevance

4.1. The minimal knowledge required to coordinate

Consider an agent, Alice, driving on the road. Alice feels safe driving on the road because she knows that the drivers around her will follow certain conventions and rules. She knows that other drivers will stop at red lights and go at green lights, that everyone will drive on the right side of the road, and that everyone will alternate who gets to go at stop signs. All of these conventions give Alice knowledge about other driver's policies that is relevant to how she can successfully coordinate with them on the open road. Despite this knowledge that Alice has about other drivers' behaviors, many aspects, if not most, are unknown to Alice. She does not know what the other drivers' destinations are, whether the other cars are manual or automatic, or the fine-grained motor controls of drivers. Luckily, from Alice's perspective, these other details about other drivers' behavior are irrelevant to her task of safely driving on the road.

In the rest of this section, we formalize the question: how can we distinguish between the knowledge of other agents' policies that is relevant and irrelevant to the success of the task?

4.2. Strategic ambiguity

Let $\Pi'_{-i} \subset \Pi_{-i}$ be the subset of policies that agent i thinks their co-policy could possibly be, given their current knowledge. If agent i can narrow down Π'_{-i} to a small enough subset, then agent i has enough knowledge to exactly characterize what their optimal response should be. In such settings, we call Π'_{-i} *strategically unambiguous*.

Definition 4.1. A nonempty subset $\tilde{\Pi}_{-i} \subset \Pi_{-i}$ of policies is *strategically unambiguous* if the best-response set to all elements of $\tilde{\Pi}_{-i}$ are identical: $\mathrm{BR}_i(\pi_{-i}) = \mathrm{BR}_i(\pi'_{-i})$ for all $\pi_{-i}, \pi'_{-i} \in \tilde{\Pi}_{-i}$. We call sets of policies that don't satisfy this property *strategically ambiguous*.

If Π'_{-i} is strategically ambiguous, it means that agent i does not have enough information to unambiguously know what their best response is; they are lacking some strategically relevant information. Let's consider the two extremes. If agent i knows agents $-i$'s policies exactly, then the policies Π'_{-i} that agent i thinks their co-players are following is a singleton and is always trivially strategically unambiguous. On the other hand, if agent i knows nothing, then $\Pi'_{-i} = \Pi_{-i}$ is the full policy space. However, in a fully decentralized problems where agent i does not not need to know anything about their co-policy, the full policy space Π_{-i} is still strategically unambiguous.

A natural next question to ask is: what is the *minimal* amount of knowledge an agent needs to have about their co-policy to characterize what policies are optimal? Or dually, what is the *maximal* subset of policies that is still strategically unambiguous?

Definition 4.2. A strategically unambiguous subset $\Pi'_{-i} \subset \Pi_{-i}$ of policies *contains no irrelevant information* if any strict superset of Π'_{-i} is strategically ambiguous.

A strict superset of Π'_{-i} corresponds to a scenario in which agent i has less knowledge. So a subset that contains no irrelevant information corresponds with a scenario in which *any* less information would lead to strategic ambiguity, In other words, all of the strategically irrelevant information has already been discarded.
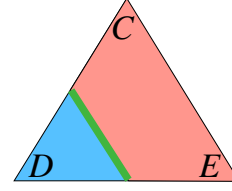
4.3. Strategic equivalence relations

Now we turn our attention to partitioning an agent's co-policy space in terms of best responses, and show how this relates to strategic ambiguity.

All of the strategically unambiguous subsets of Π_{-i} that contain no irrelevant information are given by taking the preimage of the best response function for agent i.

Theorem 1. The strategically unambiguous subsets of Π_{-i} that contain no irrelevant information are given by the preimage of the best-response map $\mathrm{BR}_i : \Pi_{-i} \to \mathcal{P}(\Pi_i)$.

Player Y

		C	D	E
Player X	A	1	0	1
	B	0	1	0

(a) A common-payoff game between two players.

(b) The strategic equivalence relation over player Y's policy space. Vertices represent pure strategies.

Figure 2: Figure 2a shows a common-payoff game between two players with multiple Nash equilibria. In Figure 2b, the red region (upper right) represents the policies for which A is a best response, the blue region (lower left) represents the policies for which B is a best response, and the green region (separating line, which technically has zero width) represents the policies for which both A and B are best responses.

Proof. We will show that a strategically unambiguous subsets of Π_{-i} contain no irrelevant information if and only if it is in the preimage of the best-response map. Consider $\mathrm{BR}_i^{-1} : \mathcal{P}(\Pi_i) \to \Pi_{-i}$, the inverse of BR_i.

($\impliedby$) Let $B = \mathrm{BR}_i^{-1}(A)$ be the preimage of A, an arbitrary set in the domain of BR_i. By definition, for any co-policy $\pi_{-i} \in \Pi_{-i}$, we have $\mathrm{BR}_i(\pi_{-i}) = A$ if and only if $\pi_{-i} \in B$. Therefore, any strict superset of B must contain a policy that has a best-response set different from A, which would introduce strategic ambiguity. Therefore, B contains no irrelevant information.

($\implies$) Now let B be a strategically unambiguous subset of Π_{-i} that contains no irrelevant information. Since B is strategically unambiguous, by definition, all policies in B must have the same best-response set; call this set A. Since B contains no irrelevant information, it must contain *all* policies in Π_{-i} that induce best response A. Therefore, $B = \mathrm{BR}_i^{-1}(A)$ and is in the preimage of BR_i. $\square$

In this sense, the preimage of BR_i gives us the coarsest (fewest element) partitioning of Π_{-i} into subsets that are strategically unambiguous. We call the relation induced by this partitioning the *strategic equivalence relation*.

Definition 4.3. The *strategic equivalence relation* (SER) $\sim_i$ for player i is the equivalence relation over the co-policy space Π_{-i} such that $\pi_{-i} \sim_i \pi'_{-i}$ if and only if $\mathrm{BR}_i(\pi_{-i}) = \mathrm{BR}_i(\pi'_{-i})$. We refer to the equivalence classes of the SER as *strategic equivalence classes* (SECs), which partition the co-policy space. We write $[[\pi_{-i}]]$ to denote the SEC that contains π_{-i}.

Figure 1 shows how the best-response function partitions the co-policy space into SECs. The SECs are exactly the strategically unambiguous subsets of Π_{-i} that contain no irrelevant information. If two policies fall in the SEC, then they must induce the same best response from agent i, by definition. Knowledge that allows agent i to differentiate between policies within the same class is extraneous, since it has no affect on agent i's choice of optimal policies. In this sense, which equivalence class contains π_{-i} is the minimum

knowledge that agent i needs to compute their set of best responses.

Consider the payoffs in Table 2a. Suppose player X knows that player Y's policy takes the form $w_1 C + w_2 E$. Notice that although different weights w_1, w_2 give distinct policies, they all fall into the same SEC because they induce the same best response, A, from player X. All policies that are more likely to choose D than C or E (combined) fall into a separate SEC and policies that choose D and C or E (combined) with equal probability fall in a third. Figure 2b shows player Y's mixed policy space and its three SECs. In order for player X to play optimally, they only need to know which of the three SECs player Y's policy falls in.

SERs partition policies into classes depending on the best response that they induce, regardless of the payoff that those policies induce. It is possible for two policies to be strategically equivalent $\pi_{-i} \sim_i \hat{\pi}_{-i}$ (i.e., they induce the same best response from agent i), while inducing different payoffs. A *valued strategic equivalence relation* is a refinement of an SER that differentiates such policies.

Definition 4.4. A *valued strategic equivalence relation* (VSER) $\overset{v}{\sim}_i$ for player i is an equivalence relation over the co-policy space Π_{-i} such that $\pi_{-i} \overset{v}{\sim}_i \hat{\pi}_{-i}$ if and only if,

(i) $\mathrm{BR}_i(\pi_{-i}) = \mathrm{BR}_i(\hat{\pi}_{-i})$, and

(ii) $R(\pi_i, \pi_{-i}) = R(\pi_i, \hat{\pi}_{-i})$ for all $\pi_i \in \mathrm{BR}_i(\pi_{-i})$.

5. Computing strategic equivalence relations

In the most general setting, our theory applies to strategic equivalence relation (SER) over the space of *mixed* policies. Even though the number of SECs is still guaranteed to be finite in the case of mixed policies (see Lemma 2 in Appendix A for more details), we have a stronger bound in the case of pure policies, where the number of strategic equivalence classes is upper-bounded by the number of pure policies (see Appendix B.1 for more details). Moreover, it is computationally easier to only compute the set of best-response policies over pure strategies in DecPOMDPs (Szer
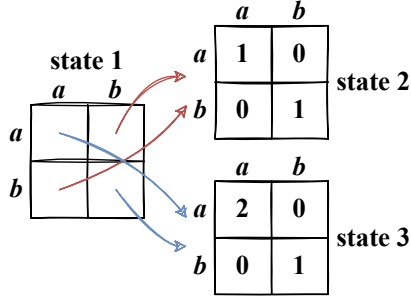
(a) A two-step dynamic game. The agents first take a joint action that transitions them into one of two final states, and then they receive a joint payoff according to their second joint action.
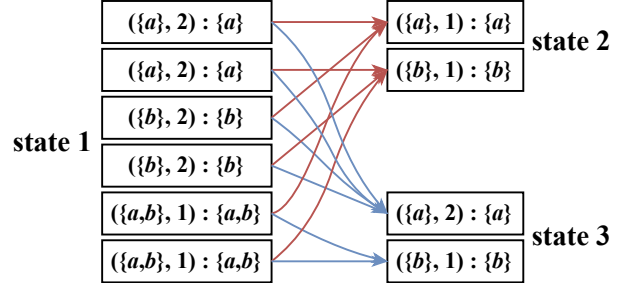
(b) The VSER DAG $\mathcal{G}_i$ for both players. Each box represents a node in $\mathcal{G}_i$ with the format $[(A_i^*, V^*) : \hat{\Pi}_{-i}]$ where A_i^* is the best-response set, V^* is the best-response value, and $\hat{\Pi}_{-i}$ are the one-step co-policies, the actions the co-policies in the equivalence class prescribe at state s.

Figure 3: A dynamic game along with the VSER DAG $\mathcal{G}_i$ for both players (since the game is symmetric).

& Charpillet, 2006). For these reasons, we limit ourselves to computing SERs over pure strategies. For similar computational reasons, we restrict ourselves to the domain of common-payoff games, although all of our theory extends to general-sum games as well.

In Section 5.3 we compute the VSER over the set of best-response policies in DecPOMDPs. In Appendix B.6, we give a simplified (and faster) algorithm for computing VSERs over Nash equilibrium in fully-observed DecPOMDPs. For more details and rationale behind computing strategic equivalence relations over subsets of the full policy space, see Appendix B.4.

5.1. Normal-form games

As a warmup, we quickly explain how to compute the strategic equivalence relation (Definition 4.3) of a normal-form game (i.e., a DecPOMDP with a single state). In order to compute the strategic equivalence relation for agent i, we just need to enumerate the pure policies $\pi_{-i} \in \Pi_{-i}$, compute the best responses $\mathrm{BR}_i(\pi_{-i}) = \arg\max_{\pi_i \in \Pi_i} R(\pi_i, \pi_{-i})$ for each one, and group the policies that induce the same best responses.

5.2. Strategic equivalence graphs

In DecPOMDPs with multiple states, instead of explicitly computing the policy within every valued strategic equivalence class (VSEC) (see Definition 4.4), we compute a more compact structure that takes advantage of the inherently recursive nature of optimal policies. This will entail computing a directed acyclic graph $\mathcal{G}_i$ as follows for each agent i.

For simplicity, let's first restrict ourselves to the case of a *fully-observed* DecPOMDP. For each subgame starting in state s there exist multiple nodes in $\mathcal{G}_i$. Each node conceptually stores a single class of the *intermediate VSER*, the

VSER over policies restricted to the subgame starting at state s. Each node explicitly contains the

1. The state s that it's associated with;
2. The *one-step co-policies*: the actions that the co-policies in the SEC prescribe at state s;
3. The set of best-responses A_i^* for agent i in state s;
4. The value V^* that they induce from agent i in state s;
5. The edges to children nodes associated with the successor states of s that continue the VSEC in the future.

Item (5) is required because the value of agent i's actions at state s are dependent on what the policies of all agents prescribe to future states. Each of these children nodes contains the information on how the VSEC continues in the future.

Each VSEC over the complete policy space is given by a root node of $\mathcal{G}_i$ (which are each associated with the starting state of the DecPOMDP) along with all of its children. For simplicity and to give computational speedup, nodes that share the same history, best-response actions, and best-response value can be merged, turning $\mathcal{G}_i$ into a multigraph (i.e. a graph that is allowed to have multiple edges between the same two nodes).

Figure 3b shows $\mathcal{G}_i$ for the simple two-step game depicted in Figure 3a. Each box represents a node in $\mathcal{G}_i$ along with it's information in the format $[(A_i^*, V^*) : \hat{\Pi}_{-i}]$, where $\hat{\Pi}_{-i}$ are the one-step co-policies. See Appendix B.2 for a detailed explanation of $\mathcal{G}_i$ in this example.

The primary change to $\mathcal{G}_i$ in partially-observed DecPOMDPs is that nodes must be associated with histories, not states, since optimal policies are computed over history-dependent beliefs. Let $\Pi_{-i}^{h_i} \subset \Pi_{-i}$ denote the subset of co-policies that are *consistent* with history h_i, i.e., there exists some random outcomes that make h_i a valid history. The nodes associated with history h_i in $\mathcal{G}_i$ store

intermediate VSERs $\overset{v}{\sim}_{h_i^t}$ over the subsets $\Pi_i^{h_i}$ based on the best response they induce from agent i, given that they've already observed h_i. Formally, for $\pi_{-i}, \pi'_{-i} \in \Pi_{-i}^{h_i}$, we have $\pi_{-i} \overset{v}{\sim}_{h_i^t} \pi'_{-i}$ if

$$\begin{aligned} &\mathrm{BR}_{h_i}(\pi_{-i}) = \mathrm{BR}_{h_i}(\pi'_{-i}), \text{ and} \\ &R(\pi_i, \pi_{-i}) = R(\pi_i, \pi'_{-i}), \forall \pi_i \in \mathrm{BR}_{h_i}(\pi_{-i}) \end{aligned} \quad (2)$$

where $\mathrm{BR}_{h_i} : \Pi_{-i}^{h_i} \to \Pi_i^t$ maps to the set of best-response policies for agent i from timepoint t onward, given history h_i.

Each node in $\mathcal{G}_i$ conceptually stores a single class of the intermediate VSER $\overset{v}{\sim}_{h_i^t}$. However, each node does not specify the full policies in that class. Each node contains the one-step co-policies, the partial policies for agents $-i$ that prescribe actions to the roots of the policy that has already been played. The nodes also store the corresponding one-step best response actions A_i^* and value V^* that they induce from agent i given history h_i. Like before, the value of these one-step actions is dependent on the actions taken by the policies at future timesteps. Therefore, nodes have one edge for every action-observation pair $(a_i, o_i) \in \mathcal{A}_i \times \Omega_i$ to other nodes associated with histories $h_i + (a_i, o_i)$. Each of these nodes contains information on how to continue the SEC in the future.

5.3. DecPOMDPs

In this section we present an algorithm for simultaneously computing the set of best-response policies in a DecPOMDP along with the VSER over these policies. The algorithm is based on point-based, dynamic programming approaches for computing the set of best-response policies in DecPOMDPs (Szer & Charpillet, 2006).

At a high-level, Algorithm 1 is based on a dynamic programming operator (the function **BackupSER**) that computes best-response policies starting at timestep t, along with the intermediate VSECs over them, given the VSECs for policies starting at timestep $t + 1$. Using this dynamic programming operator, we can then efficiently compute the full set of best-response policies along with their strategic equivalence classes through backwards induction over a finite-horizon DecPOMDP. Repeated application of this operator corresponds with iteratively computing the nodes of the VSER DAG $\mathcal{G}_i$ from the last timestep to the first.

As hinted by the structure of $\mathcal{G}_i$ in Section 5.2, intermediate VSECs at history h_i^t can be recursively computed in terms of intermediate VSECs from histories that are one-step extensions of h_i^t. The following theorem captures this idea.

Theorem 2 (Bellman backup for strategic equivalence). For all histories h_i^t and policies $\pi_{-i}, \pi'_{-i} \in \Pi_{-i}^{h_i^t}$, we have

---

Algorithm 1 Computes the valued strategic equivalence relation over best-response policies in the form of $\mathcal{G}_i$.

input DecPOMDP $\mathcal{M}$.
1: **function Main**
2:     **for all** $t \in \{T, T-1, \ldots, 0\}$ **do BackupSER**(t)
3: **end function**
4: **function BackupSER**(t)
5:     **for** all previous joint policies π^t **do**
6:         **for** each agent i and history h_i^t **do**
7:             $B(A_i^*, V^*) \leftarrow \emptyset$
8:             **for** $[[\pi_{-i}]] \in \mathcal{O}(\mathcal{G}, h_i^t, \pi^t)$ **do**
9:                 **Bellman**(Q_i^*, π_{-i}, h_i^t)
10:               $A_i^* \leftarrow \arg\max_{a_i} Q_i^*(\pi_{-i}, h_i^t, a_i)$
11:               $V^* \leftarrow \max_{a_i} Q_i^*(\pi_{-i}, h_i^t, a_i)$
12:               $B(A_i^*, V^*)$.add($[[\pi_{-i}]]$)
13:             **end for**
14:             **for all** (A_i^*, V^*) **do** $\mathcal{G}_i$.addNode($B(A_i^*, V^*)$)
15:         **end for**
16:     **end for**
17: **end function**
output $\{\mathcal{G}_i\}_{i \in [m]}$

$\pi_{-i} \overset{v}{\sim}_{h_i^t} \pi'_{-i}$ if and only if,

(i) $\arg\max\limits_{a_i} Q_i^*(\pi_{-i}, h_i^t, a_i) = \arg\max\limits_{a_i} Q_i^*(\pi_{-i}, h_i^t, a_i)$, and $\max\limits_{a_i} Q_i^*(\pi_{-i}, h_i^t, a_i) = \max\limits_{a_i} Q_i^*(\pi_{-i}, h_i^t, a_i)$;

(ii) $\pi_{-i} \overset{v}{\sim}_{h_i^t + (a_i, o_i)} \pi'_{-i}$ for all actions $a_i \in \arg\max_{a_i} Q_i^*(\pi_{-i}, h_i^t, a_i)$ and corresponding possible observations o_i.

Proof. See Appendix B.3   □

As we see below, if we back up the optimal Q-functions, $Q_{\pi_{-i}}^*$ based on future timesteps, we can check condition (i). Since we have already computed VSECs for all future histories, we can check condition (ii).

The optimal Q-functions Q_i^* can be backed up using a recursive computation analogous to the classic Bellman backup operator (see Lemma 3 in Appendix B.3). This operation is defined as the function **Bellman** on line 9 of Algorithm 1. Now we give a detailed explanation of the remaining components of Algorithm 1.

Lines 5 and 6 iterate over each past joint policy π^t, agent i, and possible history h_i^t under π^t for agent i.

The operator $\mathcal{O}(\mathcal{G}_i, h_i^t, \pi^t)$ on line 8 conceptually iterates over the class $[[\pi_{-i}]]$ of policies that are equivalent under condition (ii). It is implemented by (1) enumerating the one-step co-policies, i.e., assignments of actions to the leaves of π_{-i}^t and (2) enumerating over choices of nodes in $\mathcal{G}_i$ associated with all possible one-step extensions $h_i^t + (a_i, o_i)$ of h_i^t.
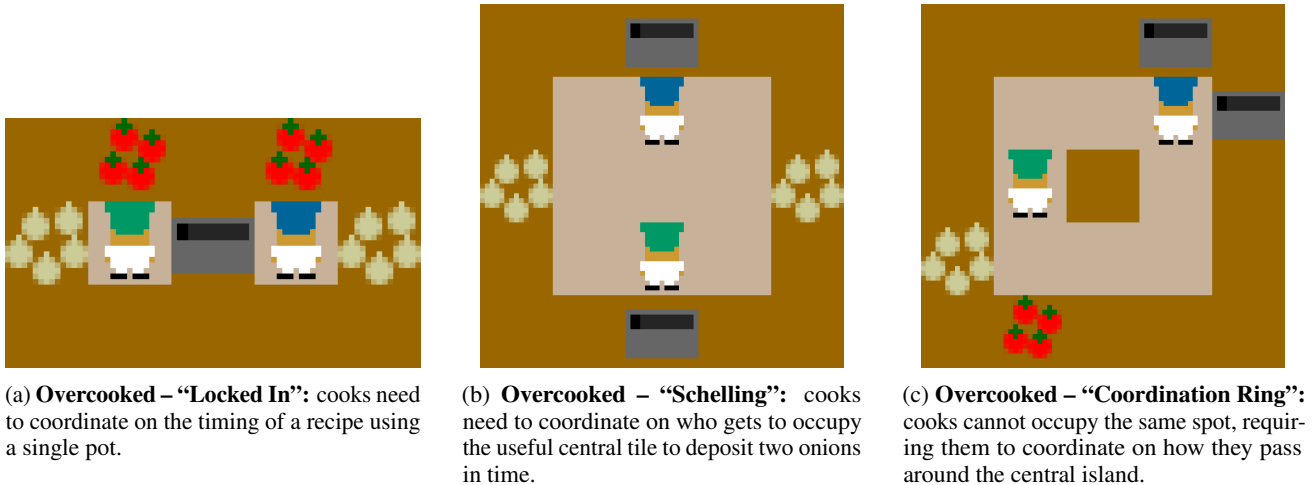
(a) **Overcooked – "Locked In":** cooks need to coordinate on the timing of a recipe using a single pot.

(b) **Overcooked – "Schelling":** cooks need to coordinate on who gets to occupy the useful central tile to deposit two onions in time.

(c) **Overcooked – "Coordination Ring":** cooks cannot occupy the same spot, requiring them to coordinate on how they pass around the central island.

Figure 4: The three Overcooked environments analyzed in Section 6, each requiring different types of coordination.

Consider computing the VSECs (Figure 3b) in state 1 of the game in Figure 3a given the intermediate VSECs of states 2 and 3. The operator $\mathcal{O}$ enumerates all choices of nodes for the two future states, and all assignments to the co-policy at state 1. For example, choosing the node $\boxed{(\{a\}, 1) : \{a\}}$ in state 2 and node $\boxed{(\{b\}, 1) : \{b\}}$ in state 3 and the one-step co-policy a produces best response $A_i^* = \{a, b\}$ in state 1.

Line 10 and 11 record the optimal actions and their value. Line 12 partitions the sets of co-policies based on their valued best response, i.e. checking condition (i). Finally, line 14 takes the computed classes and adds the corresponding node and edges to $\mathcal{G}_i$.

6. Experiments

The purpose of our experiments is twofold: (1) to characterize and gain intuition about what the strategic equivalence relation (and its equivalence classes) look like in practice, and (2) to empirically evaluate the scaling performance of our algorithms. We compute the strategic equivalence relation (SER) for different environments and provide several videos made available on our website that include 1-4 sampled representative policies from each class.

Our primary evaluation environment is the Overcooked environment (Carroll et al., 2019), in which players control chefs that cook meals in a kitchen. Agents need to coordinate on the high-level strategy for collecting the different ingredients in a dish and the low-level motion controls to avoid getting in each other's way. The onion and tomato tiles hold an infinite supply of the ingredients and agents get positive reward for placing various combinations of vegetables in the pots, depending on the specific environment. Agents have access to different movement actions (depending on the environment) and an interact action which picks up or places ingredients depending on what tile the agent is

facing. We provide analysis and visualization of additional Overcooked environments in Appendix C.

6.1. Fully-observed settings

Overcooked – Locked-in. The first Overcooked environment we investigate is visualized in Figure 4a. We let the agents' action space be composed of the following: rotate left, rotate right, and interact. The agents need to collaborate to make a soup containing an onion and a tomato within 7 timesteps. However, the recipe requires placing the onion into the pot precisely one timestep after the tomato to get +1 reward, or the soup will be ruined.

This environment has two strategic equivalence classes: one class has the left agent collecting an onion and the right agent collecting a tomato, while the other class has the roles switched. Because the environment is fully-observed, both classes have both agents delaying their commitment to either vegetable at the beginning of the episode to see which vegetable their co-player will choose, and then at some point breaking the symmetry by committing to a vegetable. See the two videos on our website for visualizations.

In the rest of this section, we investigate computing valued strategic equivalence relations (VSECs) over trembling-hand subgame perfect Nash equilibrium using the optimized algorithm described in Appendix B.6.

Overcooked – Schelling Point. This Overcooked environment is depicted in Figure 4b. Agents get a reward of +1 for each onion that is put into either of the pots. We let agents also have a larger action space: move left, right, up, down, stay still, and interact. We use a horizon of $H = 8$. In this environment agents need to coordinate on who gets to occupy the useful central tile without running into each other (which results in a no-op). Optimal policies achieve a total reward of +2.
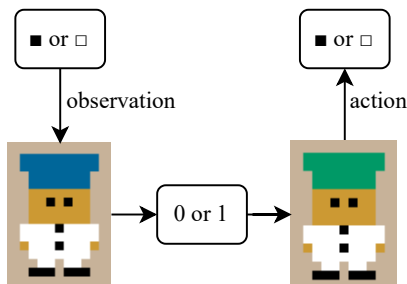
Figure 5: **Referential game:** Blue (left) first randomly observes either ■ or □. Green (right) observes a 0/1 signal from Blue and chooses ■ or □, trying to match Blue's observation.

This environment has only two equivalence classes, depending on which agent gets to occupy the central square first. After one agent occupies the central square, the symmetry in the problem is broken and the second agent's best responses become fixed. See the two videos on our website for visualizations.

Overcooked – Coordination Ring. This Overcooked environment is depicted in Figure 4c. Agents get a reward of +1 as soon as there is an onion and tomato in the same pot. Again we let agents also have a larger action space: move left, right, up, down, stay still, and interact. We use a horizon of $H = 11$, although different values of H produce similar results.

At first examination, one might predict that there should be multiple VSEC in this problem, perhaps resulting from which agent gets which vegetable, which pot to use, or the various options for how the agents could walk around the central island to stay out of each other's way. However, all of these variations are found within the same strategic equivalence class!

Surprisingly, this environment has only a *single* VSEC for either agent. Conceptually, this means that both agents *do not need to know anything* about their co-policy in order to play optimally; agents require no preexisting agreements on how to successfully coordinate in the environment. Since the environment is fully-observed, the blue agent is able to watch the green agent's actions and best respond no matter what strategy green chooses. See the video on our website for a visualization of this VSEC.

6.2. Partially-observed environments

Referential Game. The first partially-observed environment we investigate is a referential game between two agents described in Figure 5. The tricky aspect of this game is that both agents need to agree on the meaning of the signal that is communicated across the channel.

We compute three natural VSECs in this game: one class where the agents agree that $■ \mapsto 0$ and $□ \mapsto 1$ (which achieves optimal reward of 1), another class where the meaning is swapped, and a third class where the green agent (right) ignores the meaning of the signal and tries to guess the blue agent's (left) observation with expected value $\frac{1}{2}$.

Overcooked – Locked-in. The second partially-observed environment we investigate is a modified version of the Overcooked environment depicted in Figure 4a. In this version of the environment, the central pot blocks the agents' vision, preventing them from observing the position or actions of their co-player. This modification greatly increases the degree of coordination required between the agents because they can not rely on observing and adapting to their co-player's behavior. The reward is the same as before: agents get +1 reward whenever an onion is put into the pot precisely one timestep after a tomato is put into the pot.

This greater degree of coordination is clearly reflected in the size of the strategic equivalence relation: there are six classes for either agent that vary along two axis: (1) who takes charge of which vegetable and (2) whether the tomato should enter the pot on timestep 4, 5, or, 6 (and onion one timestep after). See the six videos on our website for visualization.

6.3. Computational efficiency

Since strategic equivalence relations are a novel concept, there are no pre-existing algorithms for computing them. We compare our algorithms against an enumerative baseline. The enumerative baseline computes strategic equivalence relations by first flattening the (partially observed) DecPOMDP down to a normal-form game where actions in the normal-form game represent policies in the DecPOMDP. Then, strategic equivalence classes are computed by enumerating co-policies and grouping by best response.

We experiment on increasing the horizon of policies in the Overcooked environment depicted in Figure 4a and testing how long it takes to compute the strategic equivalence relation using different algorithms. From Figure 6a (which is on a log scale), we can see the clear benefit of exploiting recursive structure to compute strategic equivalence classes, with results amplified using the optimized algorithm for a fully-observed version of the same setting. Figure 8 in the Appendix shows the variant of this plot on an absolute scale.

Since our algorithm also computes the set of best-response policies, it provides notable computational complexity improvements over existing algorithms (Szer & Charpillet, 2006; Seuken & Zilberstein, 2007) for computing the best-response policies in DecPOMDPs. By grouping co-policies into VSECs, the backup step of our algorithm avoids redundant computation from computing the best-responses to
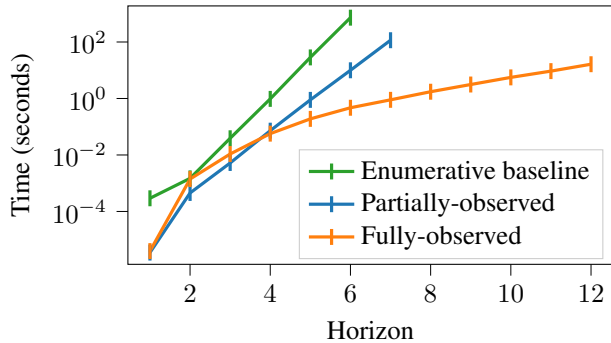
(a) *Fully-observed* (Algorithm in Appendix B.6) uses the fully-observed version of the environment. *Partially-observed* (Algorithm 1) and *Enumerative baseline* (described in Section 6.3) use the partially-observed version of the environment.
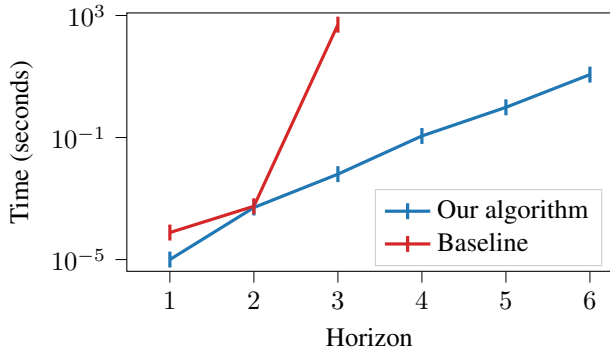
(b) The time it take to compute the set of best-response policies using our algorithm (Algorithm 1) and the baseline from Szer & Charpillet (2006). Beyond a horizon of two, the baseline timed-out at 30 minutes.

Figure 6: The time (on a log scale) for computing strategic equivalence relations over policies of varying horizon using varying algorithms in the Overcooked environment depicted in Figure 4a.

co-policies that are in the same SEC. Therefore, the number of multiagent beliefs that we need to consider for each history no longer scales in the exponential space of our co-policies, but rather in the number of intermediate VSECs over those co-policies. Figure 6b shows the time it take to compute the set of best-response policies using our algorithm (Algorithm 1 and *Partially-observed* in Figure 6a) and the baseline from Szer & Charpillet (2006). Both algorithms are implemented in Python using the same data structures.

7. Discussion

We have shown that the distinction between strategically relevant and irrelevant information can be formalized via the novel concept of strategic equivalence relation (SER). By providing a compact representation of this relation, and an efficient Bellman backup to compute it, we have provided an approach to efficiently understand what is strategically relevant to any given task. This allows us to shed new light on existing coordination benchmarks. For instance, we show in Section 6 that there is non-trivial strategically relevant information in Overcooked only in situations in which simultaneous decisions between incompatible optimal joint plans must be made. Given that such situations are rare, this provides a theoretical explanation as to why relatively good coordination with humans is achievable in this domain without any human data (Strouse et al., 2021). Our framework also explains why the introduction of simultaneous decisions (most easily through partial observability) can lead to benchmarks that are specifically more challenging for coordination, such as Hanabi (Bard et al., 2020).

The SER could serve as a critical component of both centralized and decentralized approaches for creating cooperative policies. In a decentralized settings, the SER tells us exactly what is needed in order to optimally coordinate with

co-players. This could allow a focused effort on few-shot or zero-shot adaptation to co-players (Zand et al., 2022; Albrecht & Stone, 2019; Stone et al., 2010; Hu et al., 2021), or tell us how to prioritize strategically relevant information if only a limited communication bandwidth is available (Wang et al., 2020; Mao et al., 2020; Berna-Koes et al., 2004).

In a centralized setting, the SER tells us what must be agreed upon before the problem may be reduced to independent single-agent problems. As such, it can provide critical information for finding useful decompositions of centralized value functions (Jin et al., 2022; Cassano et al., 2021; Wang* et al., 2020), or more efficient DecPOMDP planning algorithms (Szer & Charpillet, 2006; Seuken & Zilberstein, 2007). In fact, in Section 6.3, we showed that our algorithm for computing the SER already represents a complexity improvement in the state-of-the-art for computing the set of best-response strategies.

While we are conscious of the computational challenges to scale our approach to complex domains, we are optimistic about future work to approximate our method. In this work, we aim to provide the theoretical foundations for this effort. Ultimately, using SERs to both better understand the challenges of cooperative tasks and to accelerate our algorithms for solving cooperative problems are both exciting directions for future research.

8. Acknowledgement

References

Albrecht, S. V. and Stone, P. Reasoning about hypothetical agent behaviours and their parameters. *CoRR*, abs/1906.11064, 2019. URL http://arxiv.org/abs/1906.11064.

Bard, N., Foerster, J. N., Chandar, S., Burch, N., Lanctot, M., Song, H. F., Parisotto, E., Dumoulin, V., Moitra, S., Hughes, E., Dunning, I., Mourad, S., Larochelle, H., Bellemare, M. G., and Bowling, M. The hanabi challenge: A new frontier for ai research. *Artificial Intelligence*, 280:103216, 2020. ISSN 0004-3702. doi: https://doi.org/10.1016/j.artint.2019.103216. URL https://www.sciencedirect.com/science/article/pii/S0004370219300116.

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 Twenty-Ninth AAAI Conference on Artificial Intelligence*, January 2015.

Berna-Koes, M., Nourbakhsh, I., and Sycara, K. Communication efficiency in multi-agent systems. In *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004*, volume 3, pp. 2129–2134 Vol.3, 2004. doi: 10.1109/ROBOT.2004.1307377.

Carroll, M., Shah, R., Ho, M. K., Griffiths, T. L., Seshia, S. A., Abbeel, P., and Dragan, A. On the utility of learning about humans for human-ai coordination, 2019. URL https://arxiv.org/abs/1910.05789.

Cassano, L., Yuan, K., and Sayed, A. H. Multiagent fully decentralized value function learning with linear convergence rates. *IEEE Transactions on Automatic Control*, 66(4):1497–1512, 2021. doi: 10.1109/TAC.2020.2995814.

Dibangoye, J. S., Amato, C., Buffet, O., and Charpillet, F. Optimally solving dec-pomdps as continuous-state mdps. *Journal of Artificial Intelligence Research*, 55:443–497, 2016.

Fagin, R., Halpern, J. Y., Moses, Y., and Vardi, M. *Reasoning About Knowledge*. The MIT Press, 01 2004. ISBN 9780262256094. doi: 10.7551/mitpress/5803.001.0001. URL https://doi.org/10.7551/mitpress/5803.001.0001.

Foerster, J., Assael, I. A., de Freitas, N., and Whiteson, S. Learning to communicate with deep multi-agent reinforcement learning. In Lee, D., Sugiyama, M., Luxburg, U., Guyon, I., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper/2016/file/c7635bfd99248a2cdef8249ef7bfbef4-Paper.pdf.

Fontaine, M. and Nikolaidis, S. Differentiable quality diversity. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 10040–10052. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/file/532923f11ac97d3e7cb0130315b067dc-Paper.pdf.

Fontaine, M., Hsu, Y.-C., Zhang, Y., Tjanaka, B., and Nikolaidis, S. On the Importance of Environments in Human-Robot Coordination. In *Proceedings of Robotics: Science and Systems*, Virtual, July 2021. doi: 10.15607/RSS.2021.XVII.038.

Hadfield-Menell, D., Andrus, M., and Hadfield, G. K. Legible normativity for AI alignment: The value of silly rules. *CoRR*, abs/1811.01267, 2018. URL http://arxiv.org/abs/1811.01267.

Hu, H., Lerer, A., Peysakhovich, A., and Foerster, J. "other-play" for zero-shot coordination, 2021.

Jin, C., Liu, Q., Wang, Y., and Yu, T. V-learning–a simple, efficient, decentralized algorithm for multiagent rl. In *ICLR 2022 Workshop on Gamification and Multiagent Solutions*, 2022.

Kearns, M. J., Littman, M. L., and Singh, S. Graphical models for game theory. *CoRR*, abs/1301.2281, 2013. URL http://arxiv.org/abs/1301.2281.

Koller, D. and Milch, B. Multi-agent influence diagrams for representing and solving games. *Games and Economic Behavior*, 45(1):181–221, 2003. ISSN 0899-8256. doi: https://doi.org/10.1016/S0899-8256(02)00544-4. URL https://www.sciencedirect.com/science/article/pii/S0899825602005444. First World Congress of the Game Theory Society.

Lewis, D. K. Languages and language. In Gunderson, K. (ed.), *Minnesota Studies in the Philosophy of Science*, pp. 3–35. University of Minnesota Press, 1975.

Mao, H., Zhang, Z., Xiao, Z., Gong, Z., and Ni, Y. Learning agent communication under limited bandwidth by message pruning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04): 5142–5149, Apr. 2020. doi: 10.1609/aaai.v34i04.5957. URL https://ojs.aaai.org/index.php/AAAI/article/view/5957.

Muglich, D., Schroeder de Witt, C., van der Pol, E., Whiteson, S., and Foerster, J. Equivariant networks for zero-shot coordination. *Advances in Neural Information Processing Systems*, 35:6410–6423, 2022.

Mura, P. L. Game networks. In *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence*, pp. 335–342, 2000.

Nair, R., Tambe, M., Yokoo, M., Pynadath, D. V., and Marsella, S. Taming decentralized pomdps: Towards efficient policy computation for multiagent settings. In *International Joint conference on Artificial Intelligence (IJCAI)*, 2003.

Oliehoek, F., Witwicki, S., and Kaelbling, L. Influence-based abstraction for multiagent systems. *Proceedings of the AAAI Conference on Artificial Intelligence*, 26(1):1422–1428, Sep. 2021. doi: 10.1609/aaai.v26i1.8253. URL https://ojs.aaai.org/index.php/AAAI/article/view/8253.

Pynadath, D. V. and Marsella, S. Minimal mental models. In *AAAI*, pp. 1038–1044, 2007.

Seuken, S. and Zilberstein, S. Memory-bounded dynamic programming for dec-pomdps. In *IJCAI*, pp. 2009–2015, 2007.

Stone, P., Kaminka, G. A., Kraus, S., and Rosenschein, J. S. Ad hoc autonomous agent teams: Collaboration without pre-coordination. In *Proceedings of the Twenty-Fourth Conference on Artificial Intelligence*, July 2010.

Strouse, D., McKee, K., Botvinick, M., Hughes, E., and Everett, R. Collaborating with humans without human data. *Advances in Neural Information Processing Systems*, 34: 14502–14515, 2021.

Suriadinata, J., Macke, W., Mirsky, R., and Stone, P. Reasoning about human behavior in ad hoc teamwork. In *Adaptive and learning Agents Workshop at AAMAS 2021*, May 2021.

Szer, D. and Charpillet, F. Point-based dynamic programming for dec-pomdps. In *Proceedings of the 21st National Conference on Artificial Intelligence - Volume 2*, AAAI'06, pp. 1233–1238. AAAI Press, 2006. ISBN 9781577352815.

Treutlein, J., Dennis, M., Oesterheld, C., and Foerster, J. A new formalism, method and open issues for zero-shot coordination. In *International Conference on Machine Learning*, pp. 10413–10423. PMLR, 2021.

Wang, R., He, X., Yu, R., Qiu, W., An, B., and Rabinovich, Z. Learning efficient multi-agent communication: An information bottleneck approach. In III, H. D. and Singh, A. (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 9908–9918. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/wang20i.html.

Wang*, T., Wang*, J., Zheng, C., and Zhang, C. Learning nearly decomposable value functions via communication minimization. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=HJx-3grYDB.

Wang, X. and Sandholm, T. Reinforcement learning to play an optimal nash equilibrium in team markov games. *Advances in neural information processing systems*, 15, 2002.

Zand, J., Parker-Holder, J., and Roberts, S. J. On-the-fly strategy adaptation for ad-hoc agent coordination. In *Adaptive Agents and Multi-Agent Systems*, 2022.

Zhang, K., Yang, Z., and Başar, T. Multi-agent reinforcement learning: A selective overview of theories and algorithms, 2019. URL https://arxiv.org/abs/1911.10635.

A. Strategic relevance

Remark 1. The definition related to strategic ambiguity and strategic equivalence relations hold for any response function $\Pi_{-i} \to \Pi_i$, not just the best response function defined in the main text, which assumes the agents' policies to be perfectly rational. Some appealing alternatives might include models of irrationality, such as "Boltzmann" best response or ϵ-best response. Exploring strategic equivalence relations in these contexts is left for future work.

A.1. Additional Lemmas

We can extend the notion of strategic equivalence to capture all perspectives of a team of agents. The preimage of the joint best-response function BR : $\Pi \to \mathcal{P}(\Pi)$ forms an equivalence relation $\sim$ over Π. In turn, $\sim$ partitions Π into equivalence classes that capture the minimum sufficient knowledge required to characterize the best response set for *every* agent. A natural question to ask is what (if any) the connection between $\sim$ and $\sim_i$ is. The following Lemma answers this question.

Lemma 1. $\pi \sim \pi'$ if and only if $\pi_{-i} \sim_i \hat{\pi}_{-i}$.

Proof. $\pi \sim \pi'$ if and only if $\mathrm{BR}(\pi) = \mathrm{BR}(\pi)$ if and only if $\mathrm{BR}_i(\pi_{-i}) = \mathrm{BR}_i(\pi'_{-i}), \forall i \in [m]$ if and only if $\pi \sim_i \pi', \forall i \in [m]$. $\qquad\square$

Another result that follows from our definitions is that the number of strategic equivalence classes (even in the case of mixed policies) is always finite whereas the space of policies or Nash equilibria can be infinite.

Lemma 2. The number of strategic equivalence classes is finite, upper bounded by $\prod_i 2^{|A_i|}$.

Proof. Each best response set is the convex combination of a subset of pure strategies. There are $\prod_i 2^{|A_i|}$ such unique subsets of pure strategies. $\qquad\square$

B. Computing strategic equivalence relations

B.1. Mixed policies can have an exponential number of strategic equivalence classes

Consider two agents in a DecPOMDP with a single state and identical finite strategy sets $\mathcal{A}$. The agents receive a payoff of 1 if they play the same strategy and 0 if they differ. In this game, depending on what the co-player does, it is possible for every single subset of $\mathcal{A}$ to be a strategic equivalence class. Namely, for each subset $A \subset \mathcal{A}$, the simplex over A is the best response to the uniform policy over A. Moreover, since π is a uniform policy, the simplex over S defines the set of best responses to this policy. Therefore, there are an exponential (in the size of A) number of distinct strategic equivalence classes, one for every subset of A.

B.2. Example strategic equivalence graph

Figure 3b shows the DAG $\mathcal{G}_i$ for the simple two-step game depicted in Figure 3a. Each box represents a node in $\mathcal{G}_i$ along with it's information in the format $(A_i^*, V^*) : \hat{\Pi}_{-i}$ where $\hat{\Pi}_{-i}$ are the one-step co-policies. The two boxes in the top-right of Figure 3b represent the two nodes in $\mathcal{G}_i$ associated with the subgame beginning in the top-right state in Figure 3a. The first (second) node show that action a (b) is the best-response set to their co-player choosing action a (b) and achieves value 1. The six boxes on the left of Figure 3b represent the root nodes of the six SECs of the full game. Each box shows the best-response set, the value of the best-response, and the actions of the policies in that class. The red and blue arrows point to future states in the game and how that class is continued at those states. Notice the pairs of nodes in the first layer that can be merged, because they share the same valued best response.

B.3. Miscellaneous Lemmas

Lemma 3 (Corollary of Bellman's principle of optimality)**.** Let b_i^t be the belief over the state space derived from history h_i^t and co-policy π_{-i}. Then,

$$Q^*_{\pi_{-i}}(h_i^t, a_i) = \mathop{\mathbb{E}}_{s^t \sim b_i^t}[R(s^t, a^t) + \mathop{\mathbb{E}}_{o_i \in P(\cdot, \cdot | s^t, a^t)}[\gamma \max_a (Q^*_{\pi_{-i}}(h_i^t + (a_i, o_i), a)))]] \tag{3}$$

where a^t is the joint action specified by a_i and the roots of π_{-i}^t.

Theorem 2 (Bellman backup for strategic equivalence)**.** For all histories h_i^t and policies $\pi_{-i}, \pi'_{-i} \in \Pi_{-i}^{h_i^t}$, we have $\pi_{-i} \overset{v}{\sim}_{h_i^t} \pi'_{-i}$ if and only if,

(i) $\underset{a_i}{\arg\max} Q_i^*(\pi_{-i}, h_i^t, a_i) = \underset{a_i}{\arg\max} Q_i^*(\pi'_{-i}, h_i^t, a_i)$, and $\underset{a_i}{\max} Q_i^*(\pi_{-i}, h_i^t, a_i) = \underset{a_i}{\max} Q_i^*(\pi'_{-i}, h_i^t, a_i)$;

(ii) $\pi_{-i} \overset{v}{\sim}_{h_i^t + (a_i, o_i)} \pi'_{-i}$ for all actions $a_i \in \arg\max_{a_i} Q_i^*(\pi_{-i}, h_i^t, a_i)$ and corresponding possible observations o_i.

Proof. We will show that $\pi_{-i} \overset{v}{\sim}_{h_i^t} \pi'_{-i}$, i.e., $\mathrm{BR}_{h_i^t}(\pi_{-i}) = \mathrm{BR}_{h_i^t}(\pi'_{-i})$ if and only if (i) and (ii) hold. Condition (i) ensures that the best response actions to π and π' are the same at timepoint t, given history h_i^t. Condition (ii) recursively checks that all other parts of the best response policies are the same: $\pi_{-i} \overset{v}{\sim}_{h_i + (a_i, o_i)} \pi'_{-i}$ means that $\mathrm{BR}_{h_i + (a_i, o_i)}(\pi_{-i}) = \mathrm{BR}_{h_i + (a_i, o_i)}(\pi'_{-i})$ for all actions $a_i \in \arg\max_{a_i} Q_i^*(\pi_{-i}, h_i^t, a_i)$ and possible observations o_i, ensuring that no matter which observation agent i observes, the set of best-response subpolicies are the same. $\qquad\square$

B.4. Computing strategic equivalence relations over subsets of the policy space

In some settings it is known that agents will follow some subset of the full policy space. In our examples, we assume that agents have some degree of rationality, so they play Nash-equilibrium or best-response policies. In these cases, it makes sense to restrict your strategic equivalence relations to the relevant subset of policies, potentially eliminating extraneous equivalence classes and thereby simplifying the strategy space.

One way of computing the SER over a subset of the full policy space is by computing the SER over the full policy space and then discarding unwanted policies. This is usually undesirable, because it can introduce lots of unnecessary computation, especially in dynamic games. Instead, we simultaneously compute the relevant subset of policies along with the SER over them. In order to compute the SER over a subset of the policies in this way, a key characteristic is that the subset of policies have a recursive substructure that can be exploited during backwards recursion. Best-response policies and subgame perfect Nash equilibrium have the property that policies at time t can be computed in terms of the policies at time $t + 1$.

B.5. Remarks on Algorithm 1

Along with providing an efficient method for computing strategic equivalence classes, Algorithm 1 implements two strict improvements over traditional point-based, dynamic programming approaches for DecPOMDPs (Szer & Charpillet, 2006).

(i) The $\arg\max$ on line 6 of the algorithm proposed in (Szer & Charpillet, 2006) only needs to be computed over the current timestep's possible actions (rather than the full space of t-step policy trees) since we record best responses and their associated value functions for future histories.

(ii) By computing strategic equivalence classes associated with future histories, we can reduce the total number of co-policies that need to be considered during the computation. Specifically, we only need to consider a single policy for each element of each of the strategic equivalence class, since all other policies will necessary induce the same best response.

Computing subgame perfect Nash Equilibrium. In order to compute only the subgame perfect Nash equilibrium (rather than all of the best-response policies), we can do iterated elimination of strictly dominated strategies (IESDS) at each stage of the backwards induction. This will leave only the subgame perfect Nash equilibrium since IESDS eliminates all of the non-Nash equilibrium in common-payoff games.

B.6. Computing strategic equivalence relations in fully-observed settings

Here, we provide additional context to computing the valued strategic equivalence relation (VSER) in fully-observed DecPOMDPs.

In fully-observed DecPOMDPs, optimal policies and Q-functions can be Markovian. Therefore, we consider the set of intermediate VSERs $\overset{v\,s}{\sim}_i$ over the subgames starting at states s. Let $T(s) = \{s' \mid \exists a \in \mathcal{A} \text{ s.t. } P(s, a, s') > 0\}$ denote the successors of state s. Then, $\pi_{-i} \overset{v}{\sim}_s \pi'_{-i}$ if and only if,
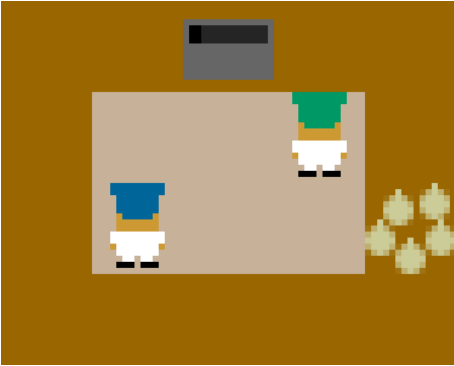
(a) **Overcooked – "Cramped":** cooks need to coordinate on low-level movements to avoid getting in each other's way.
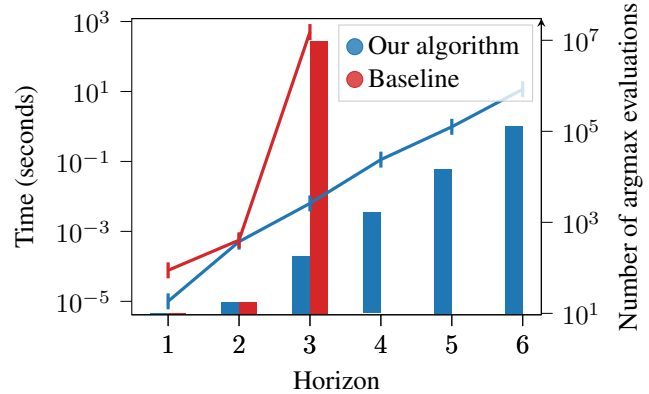
(b) The time (lines) and number of argmax evaluations (bars) it take to compute the set of best-response policies using our algorithm (Algorithm 1) and the baseline from (Szer & Charpillet, 2006). Beyond a horizon of two, the baseline timed-out at 30 minutes.

Figure 7: Additional experiments.

(i) $\pi_{-i} \overset{v\ s'}{\sim_i} \pi'_{-i}, \forall s' \in T(s)$, and

(ii) $\arg\max_{a^i}\{Q^{\pi_{-i}}(s, a^i)\} = \arg\max_{a^i}\{Q^{\pi'_{-i}}(s, a_i)\}.$

As $Q^{\pi}(s, a) = \mathbb{E}_{s'}[R(s, a) + \gamma V^{\pi}(s')]$, the value function is also recursively defined in terms of successor states $T(s)$.

Given this recursive construction of the VSER of the subgames of G, we can compute the VSER over the subgame perfect Nash equilibrium of the full game G using backwards induction similar to Algorithm 1. The two key differences are (1) we only need to do backwards induction over the state space, not the history, and (2) Nash equilibrium can be computed directly, since the actions of co-policies are determined by the current state, rather than having to resort to iterated elimination.

C. Additional Experiments

In this section, we provide two additional experimental results. In the first, we analyze the valued strategic equivalence relations (VSERs) over subgame perfect Nash equilibrium of an additional fully-observed Overcooked environment (depicted in 7a). Videos of each of the SECs in this environment is provided in the supplemental material. As in the environment depicted in Figure 4b, agents get +1 reward anytime an onion is placed into any pot and have access to six actions: move left, right, up, down, stay still, and interact. Optimal policies achieve a total payoff of +2. Agents need to coordinate on low-level movements to avoid getting in each other's way to get as many onions in the pots within a horizon of 9. This environment gives rise to only two equivalence classes, depending on which agent collects the onions first. See the two videos on our website for visualizations.

We also report an expanded version of Figure 6b in the form of Figure 7b that also reports the number of *argmax evaluations* performed by our algorithm and the baseline from (Szer & Charpillet, 2006). Within the dynamic programming step of either algorithm, an argmax is performed to find the optimal action (line 10 of Algorithm 1) or subpolicy (step 2.a.iii of Figure 2 in (Szer & Charpillet, 2006)) for the current subgame. This measure provides a proxy for the number of iterations the inner-most loop of each algorithm needs to take and is recorded in the form of bars in Figure 7b, following a trend similar to wall-clock time.
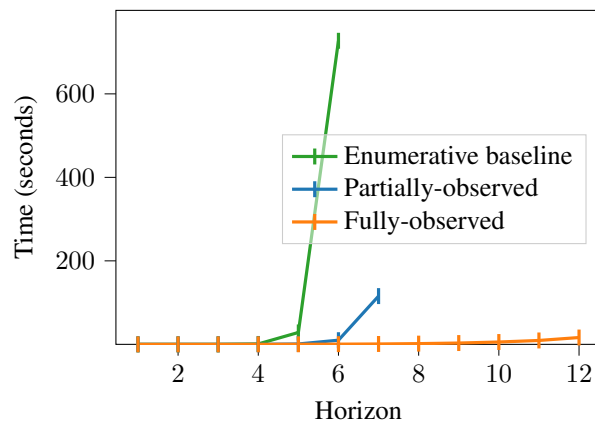
Figure 8: The time (on an absolute scale) for computing strategic equivalence relations over policies of varying horizon in the Overcooked environment depicted in Figure 4a. *Fully-observed* (Algorithm in Appendix B.6) uses the fully-observed version of the environment. *Partially-observed* (Algorithm 1) and *Enumerative baseline* (described in Section 6.3) use the partially-observed version of the environment.