SUMMARIZING SOCIETIES: AGENT ABSTRACTION IN MULTI-AGENT REINFORCEMENT LEARNING

Amin Memarian[*,†,‡], **Maximilian Puelma Touzel**[*,†,‡], **Matthew Riemer**[†,‡,§], **Rupali Bhati**[†,¶], **Irina Rish**[†,‡]
{memariaa, puelmatm, irina.rish}@mila.quebec, mdriemer@us.ibm.com, rupali.bhati.1@ulaval.ca

ABSTRACT

Agents cannot make sense of many-agent societies through direct consideration of small-scale, low-level agent identities, but instead must recognize emergent collective identities. Here, we take a first step towards a framework for recognizing this structure in large groups of low-level agents so that they can be modeled as a much smaller number of high-level agents—a process that we call *agent abstraction*. We illustrate this process by extending bisimulation metrics for state abstraction in reinforcement learning to the setting of multi-agent reinforcement learning and analyze a straightforward, if crude, abstraction based on experienced joint actions. It addresses non-stationarity due to other learning agents by improving minimax regret by a intuitive factor. To test if this compression factor provides signal for higher-level agency, we applied it to a large dataset of human play of the popular social dilemma game Diplomacy. We find that it correlates strongly with the degree of ground-truth abstraction of low-level units into the human players.

1 INTRODUCTION

Much of the complexity in life arises from the way that individuals organize into collective behaviours. This becomes evermore the case when we acknowledge that what we often think of as ’individuals’ are really abstract entities comprised of many smaller entities that can be separately viewed as agents (Levin, 2019). When tackling this complexity, it often becomes useful to exploit the coherence in that behaviour by abstracting the space of joint actions that those individuals can take. We provide the following working definition:

Definition 1.1. *Agent Abstraction.* An approximate clustering of part or all of the action space of two or more other agents in the environment, performed by either another interacting agent or an outside observer for the benefit of its own learning and/or planning.

What about the utility of such abstractions? For example, it would seem so obviously advantageous to abstract cells of a human into a whole, given how many there are, how well-separated they are from the outside-human environment, and how completely dependent they have evolved to become on the inside-human environment. However, there are collectives of simple and complex organisms for which the abstraction is more tenuous insofar as its utility is less certain. The uncertainty about the utility grows when considering abstractions for groups of agents that are not so obviously acting collectively and reminds us that the utility of agent abstraction arises from the strength of the collective behaviour and how a behaving agent can make use of that knowledge in maximizing the value of its actions. So, how can we measure the strength of collective behaviour and how can we tie abstracted representations of this behaviour to formal utility in multi-agent reinforcement learning (MARL)?

[*] Equal contribution.
[†] Mila - Quebec AI Institute
[‡] Université de Montréal
[§] IBM Research
[¶] Université Laval

Given the overhead inherent in identifying proper abstractions, is building this capability into artificial intelligence even advantageous? In this paper, we show formally that, yes, agent abstractions can help each agent navigate the learning and planning process in the face of the non-stationarity in the environment arising from the presence of other learning agents, a key challenge to efficient reinforcement learning in multi-agent settings.

This is perhaps not surprising, given that abstraction is a well-studied concept in reinforcement learning and there is a vast literature on state and temporal abstraction in single agent settings. So, in the MARL setting, where other agents can be viewed as part of the environment, there is a natural extension of these ideas to abstracting the actions of other agents. Here, we begin paving that extension, bringing us a step closer to a good agent abstraction metric that can be deployed by agents in MARL settings.

We make the following theoretical contributions:

- We formulate agent abstraction as a special case of well-established bisimulation metrics and present a simple, but limited strategy to obtain one based on unique joint actions (Section 2.1).
- We define a compression measure inspired by a connection that we reveal between this abstraction and an improvement factor in the standard minimax regret bound for a RL agent (Section 2.2).
- We reduce a two-level MARL system to a single, low-level version that serves to test a compression measure's ability to reveal higher-level agency from the joint actions of low-level agents (Section 3.1).

Finally in Section 3.2, we applied our reduction scheme to the game *Diplomacy* for which we obtained access to a large dataset of human-played games (Paquette et al., 2019). We show that despite its obvious limitations, the abstraction strategy we present gives a compression factor that correlates strongly with true player-groupings of unit agents controlled by individual human players. This suggests that more sophisticated metrics of the kind we outline that make better use of the action space structure could serve in forming useful agent abstractions.

2 Agent Abstraction For Behaving In Multi-Agent Environments

In MARL, the environment transition dynamics and reward function do not just depend on the environment state and actions from a single agent, but rather the joint space of actions of all agents acting in the environment. To be concrete, in an environment with N agents the environment transitions dynamics can be expressed by $T(\boldsymbol{s}'|\boldsymbol{s}, a^1, \ldots, a^N)$ with state $\boldsymbol{s} \in \mathcal{S}$, next state $\boldsymbol{s}' \in \mathcal{S}$, and an action for each agent $a^i \in \mathcal{A}^i \ \forall i \in \{1, \ldots, N\} \equiv \boldsymbol{N}$. Each agent i has their own reward function $R^i(\boldsymbol{s}, a^1, \ldots, a^N) \in \mathbb{R}$ and policy $\pi^i(a^i|\boldsymbol{s})$ for generating actions. Whereas in single agent RL a stationary model of the environment can be learned as only a function of an agent's own behavior, in MARL an attempt to do this has an implicit dependence on the potentially changing policies of other agents. Without loss of generality, we will conduct our analysis from the perspective from an arbitrary agent 1: the transitions for this agent are given by $T(\boldsymbol{s}'|\boldsymbol{s}, a^1) = \sum_{a^2 \in \mathcal{A}^2, \ldots, a^N \in \mathcal{A}^N} [\pi^2(a^2|\boldsymbol{s}) \times \cdots \times \pi^N(a^N|\boldsymbol{s}) \times T(\boldsymbol{s}'|\boldsymbol{s}, a^1, a^2, \ldots, a^N)]$. As such, even in decentralized and model-free settings it is necessary for agents to predict the actions of other agents in order to stabilize learning (Littman, 1994; Tesauro, 2003; Lowe et al., 2017). This stability is then achieved by approximating an action value function $Q^{\boldsymbol{\pi}}(\boldsymbol{s}, a^1, \ldots, a^N)$ over the joint policy space $\boldsymbol{\pi} = (\pi^1, \ldots, \pi^N)$ of all N agents.

Even in the best case scenario where all actions are observed and all policies are known ahead of time, a single agent can naively view this as a single agent RL problem with a state space augmented by the action space of other agents, $\mathcal{S}_1^+ = \mathcal{S} \times \mathcal{A}^{-\mathbf{1}}$ where $\mathcal{A}^{-\mathbf{1}} = \mathcal{A}^2 \times \cdots \times \mathcal{A}^N$. Without exploiting the structure in the state and action spaces, a well-known result in the RL literature (Osband & Van Roy, 2016) is that an agent cannot achieve minimax regret (*i.e.* best in worst-case) better than

$$\Omega\left(\sqrt{HT|\mathcal{S}_1^+||\mathcal{A}^1|}\right) = \Omega\left(\sqrt{HT|\mathcal{S}||\mathcal{A}^{-\mathbf{1}}||\mathcal{A}^1|}\right) = \Omega\left(\sqrt{HT|\mathcal{S}|\left(\prod_{i=2}^{N}|\mathcal{A}^i|\right)|\mathcal{A}^1|}\right), \tag{1}$$

where H is the episode horizon length (or the minimum diameter for continuing problems), T is the number of steps in the environment, and Ω is standard notation for asymptotic lower-bound scaling behaviour. However, such structure often exists so that, *e.g.*, leveraging an abstract state space of reduced size can help significantly by reducing the $|\mathcal{S}|$ factor in Equation (1). There is already a vast literature on constructing such abstractions (Ferns et al., 2004; Li et al., 2006; Taylor et al., 2008; Ferns et al., 2011; Ferns & Precup, 2014; Castro, 2020; Zhang et al., 2020). In this work, we will focus instead on reducing the potentially much larger contribution for many agent settings from $\mathcal{A}^{-\mathbf{1}}$ by formulating abstractions on this action space of other agents in the environment.

2.1 Agent Abstraction As A Bisimulation Metric

Our view of agent abstraction can be seen as a special case of bisimulation-based state abstraction metrics following the results of (Ferns et al., 2004). The factored state view of agent abstraction presented previously can indeed be seen as a special case of a general MDP over the augmented state space $\mathcal{S}_1^+$ from the perspective of arbitrary agent 1. This leads to the following definition for state abstraction bisimulation metric $d(x,y) \quad \forall x, y \in \mathcal{S}_1^+$ (Lemma 4.1 of (Ferns et al., 2004)) leveraging the Wasserstein distance function between distributions $\mathcal{W}$:

$$d(x,y) = 0 \Leftrightarrow R^1(x,a^1) = R^1(y,a^1) \;\text{ and }\; \mathcal{W}\left(T(\cdot|x,a^1), T(\cdot|y,a^1)\right) = 0 \;\; \forall a^1 \in \mathcal{A}^1 \; . \tag{2}$$

For agent abstraction, we are interested in further decomposition of the augmented state space, $\mathcal{S}_1^+$. To illustrate, let us focus on whether an abstraction is valid between only a pair of agents, $i \neq 1$ and $j \neq 1$ $(i \neq j)$, for which we consider the decomposition $\mathcal{S}_1^+ = \mathcal{S} \times \mathcal{A}^i \times \mathcal{A}^j \times \mathcal{A}^{\text{rest}}$, where $\mathcal{A}^{\text{rest}}$ denotes the joint action space of all other agents not including agent 1. We can then narrow the scope of the metric onto $\mathcal{A}^i \times \mathcal{A}^j$.

Definition 2.1. A *bisimilar agent abstraction metric* for agent 1 on a pair of agents i and j with any pair of joint actions $a^{ij} = (a^i, a^j)$ and $a^{i'j'} = (a^{i'}, a^{j'})$ satisfies:

$$\begin{aligned} d(a^{ij}, a^{i'j'}) = 0 \Leftrightarrow R^1(s, a^1, a^i, a^j, a^{\text{rest}}) = R^1(s, a^1, a^{i'}, a^{j'}, a^{\text{rest}}) \;\text{ and} \\ \mathcal{W}\left(T(\cdot|s, a^1, a^i, a^j, a^{\text{rest}}), T(\cdot|s, a^1, a^{i'}, a^{j'}, a^{\text{rest}})\right) = 0 \;\; \forall a^1 \in \mathcal{A}^1 \; . \end{aligned} \tag{3}$$

For example, consider a partition, $\mathcal{C} = \{C_k\}$, on $\mathcal{A}^i \times \mathcal{A}^j$. Then, $a^{ij} \in C_k$ and $a^{i'j'} \in C_{k'}$ for some k and k', respectively. The semi-metric $d_{\mathcal{C}}(a^{ij}, a^{i'j'}) = 1 - \delta_{kk'}$ always satisfies Equation (3) when C_k and $C_{k'}$ contain only a^{ij} and $a^{i'j'}$, respectively. This is true when $\mathcal{C}$ is the set of all singletons for which $|\mathcal{C}| = |\mathcal{A}^i \times \mathcal{A}^j|$. We are interested instead in partitions that compress the joint action space, *i.e.* for which $|\mathcal{C}| < |\mathcal{A}^i \times \mathcal{A}^j|$. The exactness of partitions, however, limits their usefulness as a basis for constructing bisimulation metrics, especially in the typical case of stochastic joint action dependencies. We can thus broaden our notion of agent abstraction by specifying the following ϵ-approximate abstraction, following the general form proposed in (Ferns et al., 2004).

Definition 2.2. An *ϵ-bisimilar agent abstraction* by agent 1 for the joint action a^{ij} of agents i and j within a given neighborhood ϵ identifies any two joint actions a^{ij} and $a^{i'j'}$ if the metric

$$\begin{aligned} d(a^{ij}, a^{i'j'}) = \max_{a^1 \in \mathcal{A}^1} \Bigg[& c_R \big| R^1(s, a^1, a^i, a^j, a^{\text{rest}}) - R^1(s, a^1, a^{i'}, a^{j'}, a^{\text{rest}}) \big| \\ & + \; c_T \mathcal{W}\left(T(\cdot|s, a^1, a^i, a^j, a^{\text{rest}}), T(\cdot|s, a^1, a^{i'}, a^{j'}, a^{\text{rest}})\right) \Bigg] \leq \epsilon \; , \end{aligned} \tag{4}$$

where c_R and c_T are weighting constants such that $c_R, c_T \geq 0$, $c_R + c_T \leq 1$, and $c_T \geq \gamma$, where γ is the discount factor.

The central result of this formulation is that the optimal value function defined over an agent-abstracted state space (based on this kind of metric) is guaranteed to be within $\frac{2\epsilon}{c_R(1-\gamma)}$ of the optimal value function on $\mathcal{S}$ for agent 1 (following Theorem 5.2 of Ferns et al. (2004)). The primary goal of constructing an agent abstraction is then to maximize the compression of the joint action space such that $|\mathcal{A}^i \times \mathcal{A}^j|/|\mathcal{C}|$ is as large as possible (*i.e.* $|\mathcal{C}|$ as small as possible) while keeping ϵ in equation 4 as small as possible.

2.2 Unique Joint Actions Experienced As A Bisimilar Agent Abstraction

Note that an arbitrary fixed policy π^i need not leverage its full action space when used in the environment. We can denote by $|\mathcal{A}_w^i|$ the total number of unique actions in a realized action sequence

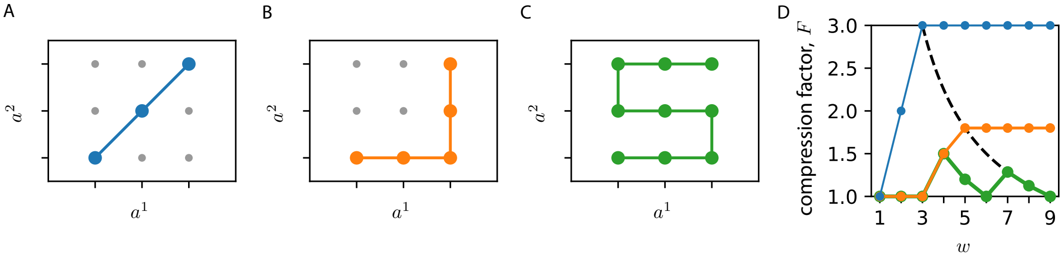

Figure 1: **Illustrative examples of abstracting the joint action space of two agents**. Three contrived cases of trajectories of in the action space (a^1, a^2) of two agents (the coverage of the respective trajectories is shown in color): **(a)** the copy case where the two agents behave identically; **(b)** the iterative case, where agents take turns sequencing through their actions; and **(c)** the space-filling case via snaking along coordinate directions. **(d)** The compression factor $F(\boldsymbol{a}_{0w}^{\{1,2\}})$ Equation (5) as a function of w for cases (a-c) (same colors). The maximum possible value $|\mathcal{A}|^{K-1}$ is attained by case (a; blue). Cases (a) and (b) grow to a fixed value $F(\boldsymbol{a}_{0\infty}^{\{1,2\}}) > 1$ with w because their trajectories do not fill the joint action space. The values for trajectories that do fill the space (e.g. case (c)) end up on $\max\{1, |\mathcal{A}|^K/w\}$ (black-dashed line). ($K = 2$, $|\mathcal{A}| = 3$.)

up to time w that agent i has taken in the environment. Thus, $|\mathcal{A}_w^i| \leq \min\{w, |\mathcal{A}^i|\}$ so that in the limit $w \to \infty$, $|\mathcal{A}_\infty^i| \leq |\mathcal{A}^i|$. Without considering any additional structure then, this result can be used to obtain an improved minimax regret bound, $\Omega\left(\sqrt{HT|\mathcal{S}|\left(\prod_{i=2}^N |\mathcal{A}_\infty^i|\right)|\mathcal{A}^1|}\right)$. One important structural constraint of MARL not exploited in this result is the fact that every policy π^i is a function of the current state $\boldsymbol{s}$. Thus, to the extent that the actions taken by each agent are correlated with this state, it is very possible that large regions of the joint action space will never be experienced at any single state (*e.g.* competitive actions when conditioned on states of plenty). Exploiting these correlations for a subset $\boldsymbol{K} \equiv \{i_1, \ldots, i_K\} \subset \boldsymbol{N}$, of $K = |\boldsymbol{K}|$ agents suggests an improvement factor of as much as $\sqrt{\left(\prod_{k=1}^K |\mathcal{A}_\infty^{i_k}|\right)/|\mathcal{A}_\infty^{\boldsymbol{K}}|}$ in the minimax regret, where $|\mathcal{A}_\infty^{\boldsymbol{K}}|$ denotes the number of unique joint actions experienced in $\mathcal{A}^{\boldsymbol{K}} \equiv \mathcal{A}^{i_1} \times \cdots \times \mathcal{A}^{i_K}$. For illustration, consider two arbitrary agents i and j and the set of unique joint actions taken in the environment over a window of time w, which we denote $\mathcal{A}_w^{i,j}$ ($|\mathcal{A}_w^{i,j}| \leq w$). Note that in the limit $w \to \infty$, $|\mathcal{A}_\infty^{i,j}| \leq |\mathcal{A}_\infty^i||\mathcal{A}_\infty^j| \leq |\mathcal{A}^i \times \mathcal{A}^j| = |\mathcal{A}^i||\mathcal{A}^j|$. Some examples of these sets are given in Figure 1(a-c). Importantly, the partition of $\mathcal{A}^i \times \mathcal{A}^j$ into singleton sets of the unique action pairs that are counted to obtain $|\mathcal{A}_\infty^{i,j}|$ (with the complement of their union added as an element) must by definition satisfy equation 3 and thus serves as a straightforward (if not optimal) bisimilar agent abstraction since it is easy to implement. We suggest some strategies for retrieving optimally compressed abstractions in the discussion, but leave their development to future work.

The form of the regret improvement factor and this metric that partitions the joint action space into visited joint actions suggests a definition for a crude measure of the utility of abstracting an action block, i.e. the joint action trajectories of a subset of $\boldsymbol{K}$ agents over an time interval from t to t', denoted $\boldsymbol{a}_{tt'}^{\boldsymbol{K}}$:

Definition 2.3. The *compression factor* achievable by abstracting an action block $\boldsymbol{a}_{tt'}^{\boldsymbol{K}}$ (formed from the subset $\boldsymbol{K} \subset \boldsymbol{N}$, of $K = |\boldsymbol{K}|$ agents over the interval from t to t') with unique joint actions is the multiplicative factor,

$$F(\boldsymbol{a}_{tt'}^{\boldsymbol{K}}) := \left(\prod_{k=1}^K n(\boldsymbol{a}_{tt'}^{\{i_k\}})\right) \Big/ n(\boldsymbol{a}_{tt'}^{\boldsymbol{K}}) \geq 1\,. \tag{5}$$

where $n(\boldsymbol{a}_{tt}^{\boldsymbol{K}})$ is the number of unique joint actions in the action block for the agent subset $\boldsymbol{K}$.

This factor is largest for the contrived case of $|\mathcal{A}|$-periodic joint action sequences with non-repeating single agent actions in the period (here we assumed for simplicity that all agents have the same

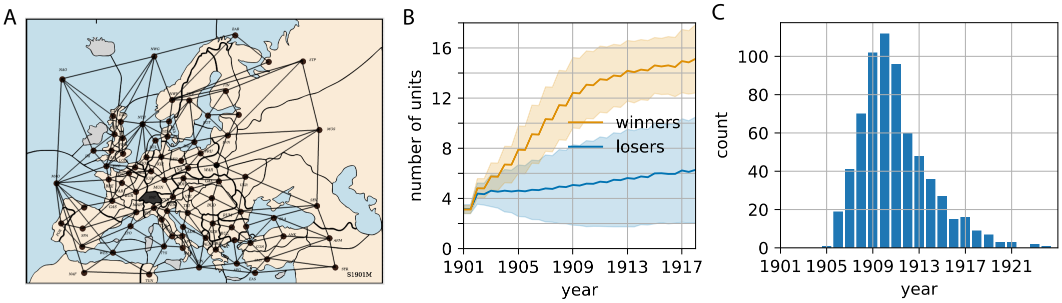

Figure 2: **The Diplomacy dataset**. **(a)** The graph of positions and movement lines along which units move (adapted from Paquette et al. (2019)). There are $|\mathcal{V}| = 81$ positions, $|\mathcal{V}^j| = \{3, 4\}$ number of unit-build locations/player, $|\mathcal{V}_{\text{supply}}| = 34$ supply centers, $n_{\text{players}} = 7$ players and $n_{\text{units}} = 18$ units/player. **(b)** Number of in-the-game units versus time in the game (2 time steps/year). Mean and standard deviation over $n_{\text{games}} = 10^3$ randomly selected games from the Paquette et al. (2019) no-press Diplomacy dataset of the $\sim 10^5$ human-player games are shown, grouped by the winning player (orange), and the rest (blue). Players aim to increase over the course of the game the size of the pool of units they control. **(c)** Histogram of game durations, $p(t_{\text{f}})$, over the same games as used in (b).

action space, $\mathcal{A}$). In this case, $F(\boldsymbol{a}_{tt'}^{\boldsymbol{K}}) = x^K/x = x^{K-1}$ for $x = \min\{t' - t, |\mathcal{A}|\}$ (Figure 1(a)). See Figure 1 for more examples.

3 Measuring player control in multi-unit, multi-player games

Here, we investigate the compression factor (Definition 2.3) as a measure of higher-level agency. In particular, when applied to a set of multi-agent trajectories, does it reflect the degree to which they can be said to be coordinating? To have access to a ground truth higher-level agency to test with, we focus on two-level, multi-agent settings, in which a set of higher-level controllers ('players', indexed by $j = 1, \dots, n_{\text{players}}$) mutually compete for resources using their control of a set of lower-level agents ('units', with each player allotted the same number of n_{units} units indexed by $i = 1, \dots, n_{\text{units}}$). We marginalize out the effects of the players that are not directly tied to unit actions, leaving a multi-agent Markov decision process (without reward) of player-labeled units indexed by (i, j). We then perform two analyses: (1) using compression factors to classify pairs units as belonging to the same versus two different players; and (2) the compression factor dependence on the number of a subset of units that belong to the same player. As an application, we focus on the board game *Diplomacy*, for which we analyzed 1000 games of a large dataset of human-played games (see Figure 2; Paquette et al. (2019)). In this section, we first give a precise formulation of a unit-level description of the game in 3.1 (to which we transformed the player action-structured data from Paquette et al. (2019)), then in 3.2 we present the statistics of the compression factor computed on multi-unit action sequences constructed from the data.

3.1 Multi-unit Markov games on graphs

Here we present a unit-level formulation of Markov games suited to describing even complicated games like Diplomacy. The environment is a graph, $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, with set of unit positions $\mathcal{V}$ and set of valid lines $\mathcal{E}$ along which units move between positions. We augment the graph with a set of $n_{\text{players}} \cdot n_{\text{units}}$ *out-of-game* positions, denoted $v_\emptyset$, one for each unit. The state of the ith unit of the jth player is then $s^{ij} \in \mathcal{S} = \{v_\emptyset\} \cup \mathcal{V}$. The state of the environment is then the tuple of positions occupied by all units, $\boldsymbol{s} = (s^{11}, \dots, s^{1n_{\text{units}}}, s^{21}, \dots, s^{n_{\text{players}}n_{\text{units}}}) \in \mathcal{S}^{\otimes(n_{\text{players}} \cdot n_{\text{units}})\,1}$. There are fixed, player-labelled unit-build locations, $\mathcal{V}^j \subset \mathcal{V}$, $\mathcal{V}^j \cap \mathcal{V}^{j'} = \emptyset$ to which units of that player transition from their *out-of-game* position when they are 'built'. Units transition to their *out-of-game* position

[1] Positions can not be occupied by more than one unit so $s^{ij} = s^{i'j'}$ only when $i' = i$ and $j' = j$.

when they are ‘disbanded’ from any in-game position as a result of an engagement (for details about engagement and other aspects of a Markov formulation of Diplomacy, see Appendix A.1).

Gameplay requires action selection for all the units, which we consider stochastic. At the player-level, action selection arises from a given set of player policies, $\{\pi^j = \pi(a^{1j}, \ldots, a^{n_{\text{units}}j}|\boldsymbol{s})\}_{j=1}^{n_{\text{players}}}$. Note that the conditioning on the state $\boldsymbol{s}$ means that each player could play the same, putative optimal policy, π^*, in which case their play is distinguished by the different starting positions of their respective units, encoded in $\boldsymbol{s}_0$. For any given set of player policies, the joint action given the state $\boldsymbol{s}$ is determined by the effective joint policy $\boldsymbol{\pi} = (\pi^1, \ldots, \pi^{n_{\text{players}}})$. Thus, for a given environment state distribution, $p(\boldsymbol{s})$, the game state distribution is $p^{\boldsymbol{\pi}}(\boldsymbol{s}, \boldsymbol{a}) = \boldsymbol{\pi}(\boldsymbol{a}|\boldsymbol{s})p(\boldsymbol{s})$.

The game dynamics are given by $T^{\boldsymbol{\pi}}(\boldsymbol{s}'|\boldsymbol{s}) = \sum_{\boldsymbol{a}} T(\boldsymbol{s}'|\boldsymbol{s}, \boldsymbol{a})\boldsymbol{\pi}(\boldsymbol{a}|\boldsymbol{s})$. Here, $T(\boldsymbol{s}'|\boldsymbol{s}, \boldsymbol{a})$ is a deterministic map and encodes the game rules, including resolutions of engagement for complicated spatial configurations. In contrast, the game evolution $T^{\boldsymbol{\pi}}(\boldsymbol{s}'|\boldsymbol{s})$ inherits stochasticity from $\boldsymbol{\pi}$, such that the variance of state distributions over games increases with time in the game from zero at their shared initial state, $\boldsymbol{s}_0$. In particular, the distribution of the environmental state evolves as the Markov chain given by $T^{\boldsymbol{\pi}}(\boldsymbol{s}'|\boldsymbol{s})$, $p^{\boldsymbol{\pi}}(\boldsymbol{s}') = T^{\boldsymbol{\pi}}(\boldsymbol{s}'|\boldsymbol{s})p^{\boldsymbol{\pi}}(\boldsymbol{s})$. For game time $t = 0, 1, \ldots$, we have $p^{\boldsymbol{\pi}}(\boldsymbol{s}_t) = (T^{\boldsymbol{\pi}})^t p(\boldsymbol{s}_0)$, with $\boldsymbol{s}_t = (s_t^{11}, \ldots, s_t^{n_{\text{players}} n_{\text{units}}})$. Note that for Diplomacy, the initial state distribution,$p(\boldsymbol{s}_0) = \mathbb{1}_{\{\boldsymbol{s}_0\}}$, is concentrated on $\boldsymbol{s}_0$, the deterministic starting state. Thus, $p^{\boldsymbol{\pi}}(\boldsymbol{s}_t, \boldsymbol{a}_t) = \boldsymbol{\pi}(\boldsymbol{a}_t|\boldsymbol{s} = \boldsymbol{s}_t)p^{\boldsymbol{\pi}}(\boldsymbol{s}_t)$ with $\boldsymbol{a}_t = (a_t^{11}, \ldots, a_t^{n_{\text{players}} n_{\text{units}}})$.

The state distribution depends on time throughout the game even for fixed $\boldsymbol{\pi}$, since the dynamics approaches the termination condition linearly in n_{units} (*i.e.* one captured supply center allows for transitioning one unit into the game), while the mixing time of $T^{\boldsymbol{\pi}}(\boldsymbol{s}'|\boldsymbol{s})$ that sets the characteristic time until the stationary distribution is reached scales exponentially with n_{units} (keeping $n_{\text{units}}/|\mathcal{V}|$ fixed).

A realization of a game initialized at $\boldsymbol{s}_0$ is produced by sampling joint actions according to $\boldsymbol{a}_t \sim \boldsymbol{\pi}(\cdot|\boldsymbol{s} = \boldsymbol{s}_t)$ and successive states from $\boldsymbol{s}_{t+1} \sim T(\cdot|\boldsymbol{s} = \boldsymbol{s}_t, \boldsymbol{a} = \boldsymbol{a}_t)$. A complete realization of a game is the corresponding sequence of ‘environment state’-‘joint action’ pairs $\tau_t = (\boldsymbol{s}_t, \boldsymbol{a}_t)$, $\boldsymbol{\tau} = (\tau_1, \ldots, \tau_{t_{\text{f}}})$, where $t_{\text{f}} = \min\{t|\mathrm{T}(\boldsymbol{s}_t)\}$ is the (stochastic) time at which the termination condition is first satisfied and the game ends. The distribution over games for this $\boldsymbol{\pi}$ is denoted $p^{\boldsymbol{\pi}}(\boldsymbol{\tau}) = \sum_{t_{\text{f}}=1}^{\infty} p^{\boldsymbol{\pi}}(\boldsymbol{\tau}|t_{\text{f}})p^{\boldsymbol{\pi}}(t_{\text{f}})$ with $p^{\boldsymbol{\pi}}(\boldsymbol{\tau}|t_{\text{f}}) = \left(\prod_{t=0}^{t_{\text{f}}-1} T^{\boldsymbol{\pi}}(\boldsymbol{s}_{t+1}|\boldsymbol{s} = \boldsymbol{s}_t, \boldsymbol{a} = \boldsymbol{a}_t)\boldsymbol{\pi}(\boldsymbol{a}_t|\boldsymbol{s} = \boldsymbol{s}_t)\right) \mathbb{1}_{\{\boldsymbol{s}_0\}}$ and $p^{\boldsymbol{\pi}}(t_{\text{f}})$ the distribution of game durations. For a subset of units, $\boldsymbol{K} = \{i_k\}_{k=1}^{K}$ for $i_k = (i, j)$ for the ith unit of player j, the action block is denoted $\boldsymbol{a}_{tt'}^{\boldsymbol{K}} = (\boldsymbol{a}_t^{\boldsymbol{K}}, \ldots, \boldsymbol{a}_{t'}^{\boldsymbol{K}})$. Over the full set of agents, we use $\boldsymbol{a}_{tt'} \equiv \boldsymbol{a}_{tt'}^{\boldsymbol{N}}$ for simplicity. Action blocks are realizations from the distribution, $p^{\boldsymbol{\pi}}(\boldsymbol{a}_{tt'}^{\boldsymbol{K}}) = \sum_{t_{\text{f}}} \sum_{\boldsymbol{s}_{0t_{\text{f}}}, \boldsymbol{a}_{t'+1,t_{\text{f}}}^{\boldsymbol{K}}} p^{\boldsymbol{\pi}}(\boldsymbol{\tau}|t_{\text{f}})p^{\boldsymbol{\pi}}(t_{\text{f}}|\{t_{\text{f}} > t'\})$.

3.2 COMPRESSION FACTOR FOR MULTI-UNIT ABSTRACTION

Within a realization $\boldsymbol{\tau}$ of the game, the number of unique joint actions of the $\boldsymbol{a}_{tt'}^{\boldsymbol{K}}$ block, i.e. the joint actions of the $\boldsymbol{K}$ subset of agents over the interval from t to t', can be written

$$n(\boldsymbol{a}_{tt'}^{\boldsymbol{K}}) = \sum_{\tilde{\boldsymbol{a}}^{\boldsymbol{K}} \in \mathcal{A}^K} \Theta\left(\sum_{\tilde{t}=t}^{t'} \delta_{\boldsymbol{a}_{\tilde{t}}^{\boldsymbol{K}}, \tilde{\boldsymbol{a}}^{\boldsymbol{K}}}\right) , \tag{6}$$

where $\Theta(x) = 1$ for $x > 0$ and 0 otherwise. Note that $n(\boldsymbol{a}_{tt'}^{ij}), n(\boldsymbol{a}_{tt'}^{\boldsymbol{K}}) \leq t' - t + 1$. Using this in the definition of the compression factor Definition 2.3 $F(\boldsymbol{a}_{tt'}^{\boldsymbol{K}})$ makes it a random variable depending on the game realization $\boldsymbol{\tau}$ sampled from $p^{\boldsymbol{\pi}}(\boldsymbol{\tau})$. For example, the expected compression factor over the game ensemble for the joint policy $\boldsymbol{\pi}$ is then

$$\overline{F(\boldsymbol{a}_{tt'}^{\boldsymbol{K}})} = \sum_{\boldsymbol{a}_{tt'}^{\boldsymbol{K}}} p^{\boldsymbol{\pi}}(\boldsymbol{a}_{tt'}^{\boldsymbol{K}})F(\boldsymbol{a}_{tt'}^{\boldsymbol{K}}) . \tag{7}$$

We can increase the signal in this factor by conditioning on subsequences for which $\left\{s_{\tilde{t}}^{\boldsymbol{K}} \in \mathcal{V}^K \; \forall \, \tilde{t} \in \{t, t+1, \ldots, t+t'\}\right\}$, *i.e.* the event that all the K agents are in the game between t and t'. The probability of this event vanishes quickly with increasing K and $t' - t$.

To distinguish player information, however, we need only compare pairs of units from the same and different players, $\boldsymbol{P}_{\text{same}} = \{i_1, i_3\}$ and $\boldsymbol{P}_{\text{different}} = \{i_1, i_2\}$, respectively, with $\boldsymbol{K} = \{i_1 =$

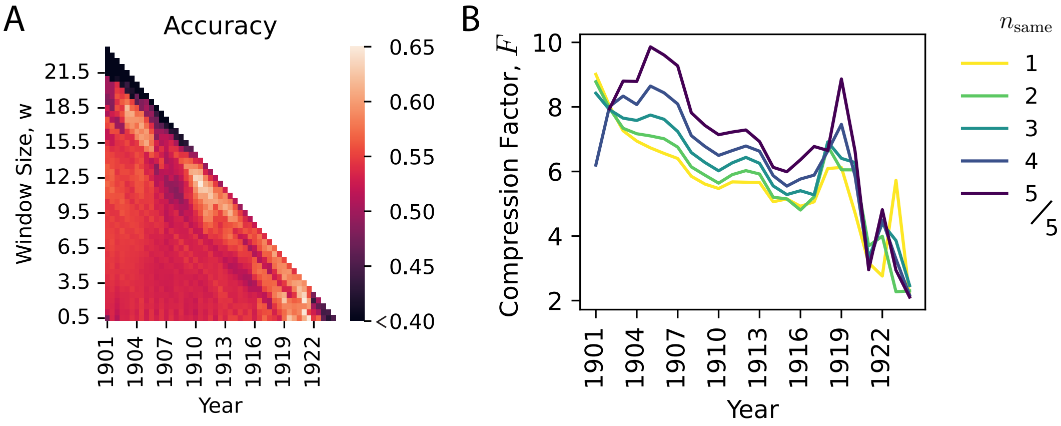

Figure 3: **Compression factor for subsets of units from Diplomacy**. **(a)** The average same-different classification accuracy based on Equation (9) as a function of block window size, w, and block start time, t, in years ($n_{\text{games}} = 10^3$). **(b)** The average compression factor, $\bar{F}$ Equation (7), as a function of block start time for $n_{\text{same}} = 1, 2, 3, 4, 5$ out of $K = 5$ agents in a subset that belong to the same player (average over window size, w; $n_{\text{games}} = 10^3$).

$(i, j), i_2 = (i', j'), i_3 = (i'', j))\}$ with $j' \neq j$. We thus track the actions of a given unit triple $\boldsymbol{K}$ in the game over the interval from t to t'. The corresponding compression factor abstracting the pair of agents over the window from t to t' is

$$F(\boldsymbol{a}^{\boldsymbol{P}}_{tt'}) = n(\boldsymbol{a}^{ij}_{tt'})n(\boldsymbol{a}^{i'j'}_{tt'})/n(\boldsymbol{a}^{\boldsymbol{P}}_{tt'})\ , \tag{8}$$

for unit pair, $\boldsymbol{P} = \{(i, j), (i', j')\}$. Applying this to the pair of same and pair of different player units in the triple blocks of length w that begin at time t in the game gives us

$$\chi^{\boldsymbol{K}}_{t,w} = (F(\boldsymbol{a}^{\boldsymbol{P}_{\text{same}}}_{tt+w}), F(\boldsymbol{a}^{\boldsymbol{P}_{\text{different}}}_{tt+w}))\ . \tag{9}$$

In obtaining the following results, we coarse-grained the single agent action space into $\mathcal{A} \in \{\text{H}, \text{M}, \text{S}\}$. We calculated $\chi^{\boldsymbol{K}}_{t,w}$ over the set of all in-game unit triples $\mathcal{K} = \{\boldsymbol{K}\}$ ordered by t and w. To assess the discriminability of this variable, we compute the classification accuracy based on the fraction of mass below the diagonal to the sum of off-diagonal mass. This accuracy is plotted as a function of t and w in Figure 3(a). The diagonal structure shows that there are periods in the game that are more informative than others. The highest discriminability occurs at an intermediate time in game.

Beyond discriminability, it remains to show that the compression factor correlates with how many units, $n_{\text{same}} = 1, \ldots, K$, in $\boldsymbol{K}$ belong to the same player (with the remaining $K - n_{\text{same}}$ units randomly sampled from the remaining players). We conditioned on blocks whose duration is set by the full duration during which all K units are in the game. This means that a unit is built at the beginning of the block and another is disbanded at its end. For computational limitations, we limited our analysis to $K = 5$. We plot the corresponding average compression factors as a function of block starting time for different n_{same} in Figure 3(b). We find that the curves are well-ordered by n_{same} and even reveal periods of enhanced compression when most units belong to the same player.

4 Related Work

Facilitating learning in MARL settings is a focus of much current research, only some of which are related to our work. For example, formation of effective teams and the emergence of player roles in teams (Wang et al., 2020a), relates to partitioning the policy space to pull specialized agents from. This is in contrast to partitioning the joint action space as considered here. Learning prototypical/archetypal agents also falls into this interesting, but unrelated research area.

Abstraction, on the other hand, is a well-studied concept in reinforcement learning from which our work based on bisimulation metrics is a direct extension (Ferns et al., 2004). With these metrics now more accessible computationally via modern methods, *e.g.* function approximation (Castro, 2020; Zhang et al., 2020), they are poised to make an impact on RL and MARL in particular. Another related area of work is factorized MDPs (e.g. VAST, CAMPs), which aim to lift the curse of dimensionality by modelling the joint space through some factorized version (Phan et al., 2021; Chitnis et al., 2020). While this approach builds in high-level agent structure, our work focusses on how an agent might learn this latent structure. However, unlike inverse RL that extracts reward function, e.g. from observed communication (Yuan* et al., 2022)– our agent abstraction metric does not depend on reward. This makes it more generally useful to cases when rewards are not observed or modelled.

A secondary challenge dealt with in Wang et al. (2020b) is coming up with a set of roles (associated with subsets of possibly overlapping action spaces) to decompose the task in a way that serves a bi-level learning hierarchy that assigns roles to different agents and learns the corresponding policy of each role. To achieve this, the effect-based representation of the joint action space is clustered using the K-means algorithm.

Our work differs from Wang et al. (2020b) in the following aspects: the goal is not to deal with scalability issues by role assignment but to find partitions that maximally compress the joint action space from the perspective of an arbitrary agent, and our focus is on a compression factor that reveals the utility of abstraction. Furthermore, our metric is neither effect-based nor requires a learned action representation.

5 Discussion

In this paper we have presented the concept of agent abstraction and grounded it in existing formulations of abstraction. We also presented a crude metric based on unique joint actions to measure the degree of abstraction and showed it provides some signal in the complex multi-agent setting of Diplomacy. Nevertheless, this metric has some obvious limitations. First, it is realization-dependent at least for non-stationary settings. Second, it does not capture the correlations among experienced joint actions. A more useful version would be based on running estimates of frequencies of joint actions. Given this distribution, optimal compression schemes could be designed that cluster joint actions with equal probability such that the entropy of the distribution over the abstracted space is maximized. This optimization must be regularized by adding the constraint that an agent learning a value function using this space does so with bounded error in the way that bisimulation metrics for RL have been designed to accomplish. There may be an interesting connection to be made here with the deterministic information bottleneck (Strouse & Schwab, 2017). See Appendix A.2 for a formulation and estimation procedure for the conditional mutual information on the action block distribution of a pair of units, $MI(A^{ij}_{tt'}; A^{i'j'}_{tt'} | \boldsymbol{S}_{tt'})$.

Our results on the game Diplomacy deserve some discussion. Why do subsets with few units from a single player give lower compression factors at early times? We speculate this is because the unconditioned case actually has more than 1 agent on average from the same player. More generally, the players in the dataset are played by different humans across samples. Thus, it is unclear to what degree the player label, i.e. the country, constrains this play variability across individual humans. While good games such as Diplomacy sculpt player agency into elaborate roles, individual differences for players of the same country are bound to impact our results since we have likely not averaged over enough games to cover the space of possibilities. Nonetheless, our paper has taken critical first steps for the community to build on towards developing scalable methods that address agent abstraction. Our work has the promise to open up an important area of future research particularly for combatting the inherent combinatorial complexity of large scale multi-agent RL applications.

ACKNOWLEDGMENTS

The authors would like to thank the team of authors of Paquette et al. (2019), and Philip Paquette in particular, for collecting and providing us with the dataset.

REFERENCES

Pablo Samuel Castro. Scalable methods for computing state similarity in deterministic markov decision processes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 10069–10076, 2020.

Rohan Chitnis, Tom Silver, Beomjoon Kim, Leslie Pack Kaelbling, and Tomas Lozano-Perez. CAMPs: Learning context-specific abstractions for efficient planning in factored mdps. In *CoRL*, 2020.

Norm Ferns and Doina Precup. Bisimulation metrics are optimal value functions. In *UAI*, 2014.

Norm Ferns, Prakash Panangaden, and Doina Precup. Metrics for finite markov decision processes. In *UAI*, volume 4, pp. 162–169, 2004.

Norm Ferns, Prakash Panangaden, and Doina Precup. Bisimulation metrics for continuous markov decision processes. *SIAM Journal on Computing*, 40(6):1662–1714, 2011.

Natasha Jaques, Angeliki Lazaridou, Edward Hughes, Caglar Gulcehre, Pedro A Ortega, DJ Strouse, Joel Z Leibo, and Nando de Freitas. Intrinsic social motivation via causal influence in multi-agent rl. 2018.

Michael Levin. The computational boundary of a "self": developmental bioelectricity drives multicellularity and scale-free cognition. *Frontiers in psychology*, 10:2688, 2019.

Lihong Li, Thomas J Walsh, and Michael L Littman. Towards a unified theory of state abstraction for mdps. *ISAIM*, 4(5):9, 2006.

Michael L. Littman. Markov games as a framework for multi-agent reinforcement learning. pp. 157–163, jan 1994. doi: 10.1016/b978-1-55860-335-6.50027-1.

Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, OpenAI Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. In *Advances in Neural Information Processing Systems*, pp. 6382–6393, 2017.

Ian Osband and Benjamin Van Roy. On lower bounds for regret in reinforcement learning. *arXiv preprint arXiv:1608.02732*, 2016.

Philip Paquette, Yuchen Lu, Seton Steven Bocco, Max Smith, Satya O-G, Jonathan K Kummerfeld, Joelle Pineau, Satinder Singh, and Aaron C Courville. No-press diplomacy: Modeling multi-agent gameplay. *Advances in Neural Information Processing Systems*, 32, 2019.

Thomy Phan, Fabian Ritz, Lenz Belzner, Philipp Altmann, Thomas Gabor, and Claudia Linnhoff-Popien. VAST: Value function factorization with variable agent sub-teams. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021. URL `https://openreview.net/forum?id=hyJKKIhfxxT`.

D J Strouse, Max Kleiman-Weiner, Josh Tenenbaum, Matt Botvinick, and David J Schwab. Learning to share and hide intentions using information regularization. In *Neural Information Processing Systems (NeurIPS)*, 2018.

DJ Strouse and David J. Schwab. The Deterministic Information Bottleneck. *Neural Computation*, 29(6):1611–1630, 06 2017. ISSN 0899-7667. doi: 10.1162/NECO_a_00961. URL `https://doi.org/10.1162/NECO_a_00961`.

Jonathan Taylor, Doina Precup, and Prakash Panagaden. Bounding performance loss in approximate mdp homomorphisms. *Advances in Neural Information Processing Systems*, 21, 2008.

Gerald Tesauro. Extending q-learning to general adaptive multi-agent systems. pp. 871–878, 2003.

Tonghan Wang, Heng Dong, Victor R. Lesser, and Chongjie Zhang. ROMA: Multi-agent reinforcement learning with emergent roles. *ArXiv*, abs/2003.08039, 2020a.

Tonghan Wang, Tarun Gupta, Anuj Mahajan, Bei Peng, Shimon Whiteson, and Chongjie Zhang. Rode: Learning roles to decompose multi-agent tasks. *arXiv preprint arXiv:2010.01523*, 2020b.

Lei Yuan*, Jianhao Wang*, Fuxiang Zhang, Chenghe Wang, Zongzhang Zhang, Yang Yu, and Chongjie Zhang. Multi-agent incentive communication via decentralized teammate modeling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022.

Amy Zhang, Rowan Thomas McAllister, Roberto Calandra, Yarin Gal, and Sergey Levine. Learning invariant representations for reinforcement learning without reconstruction. In *International Conference on Learning Representations*, 2020.

A Appendix

A.1 Markov formulation for the game of Diplomacy

The game always begins from the same state, $\boldsymbol{s}_0$, having units positioned at all respective unit-build locations. The remaining units are initialized at their *out-of-game* position. Occupation of each of a target subset of positions, $\mathcal{V}_{\text{supply}} \subset \mathcal{V}$, called *supply centers*, confers the ability to sustain an additional unit. Thus, the recurrent goal of the game is to capture supply centers in order to build more units to capture more supply centers. The termination condition that ends the game is the event $\mathrm{T}(\boldsymbol{s}) = \{s^{ij} \in \mathcal{V} \; \forall \; i, \; \text{for any } j\}$, *i.e.* when some player j has managed to occupy n_{units} supply centers such that all its units are in the game.

The full game has five distinct seasons of dynamics each year. However, marginalizing over the players allows us to reduce this to only the two seasons when units act. Each season, every unit must either *hold* (H), i.e. do nothing, *move* (M) to an adjacent position, or *support* (S) an adjacent position. We denote the action of the ith unit of the jth player located at position k, $a_k^{ij} \in \mathcal{A}_k = \{\mathrm{H}, \mathrm{M}_1, \dots, \mathrm{M}_{|E_k|}, \mathrm{S}_1, \dots, \mathrm{S}_{|E_k|}\}$, where E_k is the set of lines connected to positions k. Thus, the action space for each unit depends on its location (except for *out-of-game* positions from which actions have no effect). We can remove this dependence on location by combining all position-dependent action spaces, such that the action of the ith unit of the jth player $a^{ij} \in \mathcal{A} = \cup_{k=1}^{|\mathcal{V}|} \mathcal{A}_k$, where state-conditioning narrows the accessible actions to $\mathcal{A}_k$ when $s^{ij} = k$. Thus, similar to the joint state $\boldsymbol{s}$, the joint action is denoted $\boldsymbol{a} = (a^{11}, \dots, a^{1 n_{\text{units}}}, a^{21}, \dots, a^{n_{\text{players}} n_{\text{units}}}) \in \mathcal{A}^{\otimes (n_{\text{players}} \cdot n_{\text{units}})}$.

When a pair of agents *engage*, *i.e.* when at least one acts such that they would occupy the same position, the unit having the larger support wins and can reside in that location, while the loser must retreat or be disbanded. A unit's support is the number of units supporting it, as well as itself. When engagement results in a draw (matching support), the effect of the movement actions precipitating the engagement are nullified.

A.2 Mutual information analysis

The respective pair action distribution at a single time t is

$$
\begin{aligned}
p^{\boldsymbol{\pi}}(a_t^{ij}, a_t^{i'j'} | \boldsymbol{s}_t) &= \sum_{\boldsymbol{a}_t / \{a_t^{ij}, a_t^{i'j'}\}} p^{\boldsymbol{\pi}}(\boldsymbol{s}_t, \boldsymbol{a}_t) \Big/ \sum_{\boldsymbol{a}_t} p^{\boldsymbol{\pi}}(\boldsymbol{s}_t, \boldsymbol{a}_t) \\
&= \begin{cases} \pi^{\{i,i'\}j}(a_t^{ij}, a_t^{i'j} | \boldsymbol{s}_t), & \text{if } j' = j \\ \pi^{\{i\}j}(a_t^{ij} | \boldsymbol{s}_t) \pi^{\{i'\}j'}(a_t^{i'j'} | \boldsymbol{s}_t), & \text{if } j' \neq j \; , \end{cases}
\end{aligned}
\tag{10}
$$

where $\pi^{Ij}(\cdot | \boldsymbol{s}) \equiv \sum_{\{a^{ij}\}_{i \notin I}} \pi^j(a^{1j}, \dots, a^{n_{\text{units}} j} | \boldsymbol{s})$ is the marginal policy for the subset I of units of the jth player.

The mutual information between this pair of state sequence-conditioned unit action sequences, $a^{ij}_{tt'}$ and $a^{i'j'}_{tt'}$, is

$$MI(A^{ij}_{tt'};A^{i'j'}_{tt'}|\boldsymbol{s}_{tt'}) = \sum_{a^{ij}_{tt'},a^{i'j'}_{tt'}} p^{\boldsymbol{\pi}}(a^{ij}_{tt'},a^{i'j'}_{tt'}|\boldsymbol{s}_{tt'}) \log\left[\frac{p^{\boldsymbol{\pi}}(a^{ij}_{tt'},a^{i'j'}_{tt'}|\boldsymbol{s}_{tt'})}{p^{\boldsymbol{\pi}}(a^{ij}_{tt'}|\boldsymbol{s}_{tt'})p^{\boldsymbol{\pi}}(a^{i'j'}_{tt'}|\boldsymbol{s}_{tt'})}\right] \tag{11}$$
$$\geq 0 \;\text{ with }\; 0 \text{ if } \; j' \neq j \; (c.f. \text{ Equation (10)}) \; ,$$

and where we have denoted the unit action sequence marginals $p^{\boldsymbol{\pi}}(a^{ij}_{tt'}|\boldsymbol{s}_{tt'}) = \sum_{a^{i'j'}_{tt'}} p^{\boldsymbol{\pi}}(a^{ij}_{tt'},a^{i'j'}_{tt'}|\boldsymbol{s}_{tt'})$. This demonstrates that action sequences arising from coordinated units ($j' = j$) can be informative of each other. Can this mutual information serve to measure coordination more broadly and be used as a way to define an effective higher-level agency between observed units, even in the absence of prior information (j)?

Equation (11) can be rewritten in a more tractable form as

$$MI(A^{ij}_{tt'};A^{i'j'}_{tt'}|\boldsymbol{s}_{tt'}) = \mathbb{E}_{p^{\boldsymbol{\pi}}(a^{i'j'}_{tt'}|\boldsymbol{s}_{tt'})}\left[D_{\text{KL}}[p^{\boldsymbol{\pi}}(A^{ij}_{tt'}|a^{i'j'}_{tt'},\boldsymbol{s}_{tt'})||p^{\boldsymbol{\pi}}(A^{ij}_{tt'}|\boldsymbol{s}_{tt'})]\right] \; , \tag{12}$$

with $p^{\boldsymbol{\pi}}(a^{ij}_{tt'}|a^{i'j'}_{tt'},\boldsymbol{s}_{tt'}) = p^{\boldsymbol{\pi}}(a^{ij}_{tt'},a^{i'j'}_{tt'}|\boldsymbol{s}_{tt'})/p^{\boldsymbol{\pi}}(a^{i'j'}_{tt'}|\boldsymbol{s}_{tt'})$. The conditional mutual information, $MI(A^{ij}_{tt'};A^{i'j'}_{tt'}|\boldsymbol{S}_{tt'})$, averages Equation (12) also over $\boldsymbol{s}_{tt'}$. In this form it can be computed directly from a measured set of game trajectories, $\mathcal{D} = \{\boldsymbol{\tau}_g\}^{n_{\text{games}}}_{g=1}$, by approximating the expectation with respect to the game trajectory marginal over the window $p^{\boldsymbol{\pi}}(\boldsymbol{s}_{tt'},a^{i'j'}_{tt'})$ using Monte Carlo estimation (Strouse et al., 2018; Jaques et al., 2018):

$$MI(A^{ij}_{tt'};A^{i'j'}_{tt'}|\boldsymbol{S}_{tt'}) \approx \frac{1}{n_{\text{games}}}\sum_{g=1}^{n_{\text{games}}} D_{\text{KL}}[p^{\boldsymbol{\pi}}(A^{ij}_{tt'}|a^{i'j'}_{tt',g},\boldsymbol{s}_{tt',g})||p^{\boldsymbol{\pi}}(A^{ij}_{tt'}|\boldsymbol{s}_{tt',g})] \; . \tag{13}$$

Here, the D_{KL} must still be computed via integration using the unit policies. This estimation still suffers from the curse of dimensionality because of the high dimensions of the distributions of game-long trajectories and so $t' = t + w$ with small w.