ConventionPlay: CAPABILITY-LIMITED TRAINING FOR ROBUST AD-HOC COLLABORATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Ad-hoc collaboration often requires agents to identify and adhere to some shared convention within a cooperative task. Existing work on reinforcement learning (RL) for ad-hoc collaboration focuses on training agents that adapt to the conventions established by their partners. These methods fail to consider the possibility that while some partners might follow only a single fixed convention, others may themselves be capable of adapting to multiple conventions. Here we present *ConventionPlay*, an RL-based approach that teaches agents to discover their partner's optimal convention by training against a learned population of partners that exhibit different degrees of adaptability across conventions. Some of these partners follow a single, fixed convention, while others are able to adapt to a subset of the possible conventions for the task in question. The existence of partners that support a limited subset of conventions forces agents trained against this population to actively probe their partner's capabilities, and steer their partner towards the most effective joint strategy that they are capable of following. Our experimental results demonstrate that agents trained via *ConventionPlay* achieve superior performance to existing ad-hoc collaboration methods against test populations of partners that are compatible with multiple conventions.

1 INTRODUCTION

Cooperative multi-agent tasks often admit a variety of different solutions, with success depending on the agents' ability to coordinate their strategies on a single solution. In such tasks, joint strategies can exhibit different levels of compatibility, with performance deteriorating when individual strategies are misaligned. We describe collections of mutually compatible strategies as "conventions". Within the broader challenge of designing agents to collaborate with unseen teammates (Mirsky et al., 2022), the ability to recognize and align with established conventions is crucial for effective ad-hoc collaboration (Stone et al., 2010; Shih et al., 2021; Lerer & Peysakhovich, 2019). In this work, we propose an approach to training collaborative agents for a novel setting – interacting with partners that have varying levels of proficiency over known conventions.

As a motivating example, consider an agent joining a cooperative setting such as a mechanic shop, a football pitch, or a professional kitchen. In each of these domains team members may possess a diverse and partially overlapping set of skills—e.g. in a kitchen, most chefs know how to both chop and sauté vegetables, but only some may be well versed in using a specialized tool. The skills of the individual agents determine the set of joint strategies available to the team, one of which they must actually commit to. A new chef joining the team then has two options. They could infer the status quo, and adapt to an established convention. Alternatively, they could act on an opportunity to produce a better outcome for the team by changing the joint strategy—for instance, reassigning tasks among the cooks—in the hope that the rest of the team would be able to adapt accordingly.

We consider how prior methods for ad-hoc collaboration would approach this scenario. A common approach (Strouse et al., 2021; Zhao et al., 2023; Carroll et al., 2019) trains collaborative agents through a two-step process: First, a diverse "population" of agents is trained for the target task—e.g. teams of cooks where each team assumes a certain task allocation. Then, a "generalist" agent is trained to coordinate with any member of that population. Each member of the initial population effectively acts as a "leader," committed to a fixed convention and expecting its partner to adapt to it. The generalist agent trained against these partners then becomes a "follower" that adapts to its

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

partner's convention. A shortcoming here is that the generalist defers unconditionally to its partner's convention, however sub-optimal, since it has never encountered a partner who could themselves adapt to a different convention. This does not reflect realistic populations, where partners vary in their capacity for adaptation—some follow only one convention, others several. Because existing methods do not model this variation, the resulting agents remain purely reactive, with no mechanism for recognizing when a partner could instead be steered toward a higher-value convention.

In this work, we introduce *ConventionPlay*, a reinforcement learning-based approach to training adaptive agents that are capable of coordinating with partners that may themselves be adaptive. The novelty of this algorithm lies in discovering how a population partitions into conventions, and using this partition to construct a structured training population of *generalist followers*—agents designed to adapt to specific, restricted subsets of conventions. Exposing our agent to these adaptive partners forces it to move beyond purely reactive behavior; it must actively probe its partner to determine whether to steer the team toward a higher-value convention, or defer to the partner's established one. Our experimental results demonstrate that *ConventionPlay* agents outperform state-of-the-art methods (Strouse et al., 2021; Cui et al., 2022; Sarkar et al., 2023) in identifying the most effective convention its partner can follow in a number of complex ad-hoc collaboration scenarios.

2 Related Work

Population Diversity The most common method for achieving robust ad-hoc collaboration is by anticipating the range of possible solutions to the task. A number of methods focus on computing a diverse set of solutions and training a generalist agent against this set of strategies. Fictitious Co-Play (FCP) (Strouse et al., 2021) provides behavioral diversity using intermediate checkpoints from a self-play training process, simulating partners of varying skill levels. Carroll et al. (2019) extract diversity from human interaction data in an offline dataset. TrajeDI (Lupu et al., 2021) incentivizes policies to generate distinct trajectories, while MEP (Zhao et al., 2023) uses an entropy bonus to encourage the population to cover the entire state-action space. LIPO (Charakorn et al., 2023), CoMeDi (Sarkar et al., 2023) and ADVERSITY (Cui et al., 2023) aim to learn policies that are incompatible with each other, yet individually optimal in self-play. Our work focuses not on discovering the task's solutions, but on leveraging these solutions to train a generalist agent that coordinates with partners of varying proficiency. To that end, our proposed method is compatible with any of the diversity-generation techniques discussed above.

We contrast our approach against two additional methods that structure diversity hierarchically. TBS (Ni et al., 2026) analyzes population diversity with a return-based similarity measure to train an ensemble of best-response agents and select among them online via a theory-of-mind classifier. SyKLRBR (Cui et al., 2022) implements a cognitive hierarchy, training agents at each level as best-responses to a uniform distribution over all lower levels. Neither method, however, considers the range of adaptive behaviors a real-world partner may exhibit.

Convention-Free Strategies Unlike methods that generate diverse solutions, another line of work focuses on zero-shot coordination (ZSC), aiming to avoid conventions that an independently-trained partner is unlikely to discover on its own. Other-Play (Hu et al., 2020) addresses the equilibrium selection problem by computing "focal" policies that are invariant to task symmetries. We argue this is unrealistic for human-AI coordination: humans do not always converge on such focal policies even when they exist, instead settling on conventions shaped by personal context and prior interactions. Off-Belief Learning (Hu et al., 2021) suppresses *arbitrary* conventions—e.g. secret handshakes that emerge to convey a partner's private state information in partially observed settings. Our emphasis is not on partial observability of the environment state, but on the ad-hoc agent's uncertainty about which equilibrium its partner prefers. Instead of avoiding conventions, our approach embraces them, aiming to construct agents capable of coordinating with any convention present in a population.

Partner Shaping We distinguish our method from explicit partner-shaping approaches. Jaques et al. (2019) employ intrinsic rewards to explicitly encourage policies that influence the actions of other agents, rewarding influence itself as an objective. With our method, influence arises naturally from a best-response objective against a population of partners with latent capabilities, requiring the agent to explore which capabilities a given partner supports. LOLA (Foerster et al., 2017) and M-FOS (Lu et al., 2022) similarly learn strategies for shaping a partner's adaptive behavior. LOLA

however requires a predefined model of the partner's learning process. While M-FOS does not require such a model, it must repeatedly interact with a specific partner to learn one, with the ability to "reset" the partner's learning process multiple times. Unlike these methods, *ConventionPlay* makes no assumptions about the specific learning algorithm the partner uses to adapt its strategy to different conventions, and learns a strategy that shapes a partner's behavior within a single episode. Instead, *ConventionPlay* uses the structure of the task itself to identify the behaviors a partner might be capable of implementing, allowing it to quickly identify the best-response to a new partner within a single episode by probing this partner with a limited set of strategies.

3 PRELIMINARIES

In this section we introduce the formalisms and notation needed to describe *ConventionPlay*. We define Dec-POMDPs, describe the process of training a best-response to a population of agents, and formalize the idea of a system of conventions.

3.1 DEC-POMDPS

We model each coordination task as a Decentralized Partially Observable Markov Decision Process (Dec-POMDP) (Oliehoek & Amato, 2016), defined by the tuple $\mathcal{M} = \langle n, \mathcal{S}, \mathcal{A}, P, R, \Omega, O \rangle$, where n is the number of agents. $\mathcal{S}$ is the set of global states, $\mathcal{A} = \times_{i=1}^{n} \mathcal{A}_i$ is the joint action space, and $P(s_{t+1}|s_t, a_t)$ defines the transition probabilities to a new state s_{t+1} given the current state s_t and joint action $a_t \in \mathcal{A}$. The task is fully cooperative, with all agents receiving the same scalar reward $R(s_t, a_t)$. The joint observation $o_t \in \Omega = \times_{i=1}^{n} \Omega_i$ is drawn from the distribution $O(o_t|s_t, a_{t-1})$, with each agent i only observing $o_{i,t}$. An individual policy $\pi_i : \mathcal{H}_i \to \Delta(\mathcal{A}_i)$ maps the agent i's local action-observation history, $h_{i,t} = (o_{i,0}, a_{i,0}, o_{i,1}, \ldots, a_{i,t-1}, o_{i,t}) \in \mathcal{H}_i$, to a probability distribution over its action space. A *joint policy* $\vec{\pi} = (\pi_1, \ldots, \pi_n)$ specifies the behavior of all agents. Here the goal for the agents is to maximize their expected *episodic* return:

$$J(\vec{\pi}) = \mathbb{E}_{s_0, a_t \sim \vec{\pi}, s_{t+1} \sim P} \left[\sum_{t=0}^{T} R(s_t, a_t) \right] \tag{1}$$

where T is the episode length. We develop *ConventionPlay* in symmetric, two-player Dec-POMDPs, with all of our experimental environments being symmetric. Symmetry means that $\mathcal{A}_1 = \mathcal{A}_2$ and $\Omega_1 = \Omega_2$, such that any individual policy can be executed by either agent, and that $J(\pi, \pi') = J(\pi', \pi)$ for all $\pi, \pi' \in \Pi$. The initial training populations for *ConventionPlay* will consist of policies $\pi \in \Pi$ trained in *self-play* (SP) to maximize the return $J_{SP}(\pi) = J(\pi, \pi)$ receives when paired with itself. We note that it is straightforward to transform any asymmetric two-agent Dec-POMDP into a symmetric one, by randomly assigning each agent to one of two possible roles in the environment, and allowing each individual policy to condition on the role to which it has been assigned. Furthermore, while we do not consider settings with more than two agents here, we see no fundamental barriers to extending *ConventionPlay* to tasks with larger numbers of agents.

3.2 BEST-RESPONSE TRAINING

An established paradigm for training agents capable of ad-hoc collaboration is to first train a finite *population* of partner policies $\mathcal{P} = \{\pi_1, \ldots, \pi_m\} \subset \Pi$, and then train a *best-response* policy against this population (Carroll et al., 2019; Strouse et al., 2021). A best-response to $\mathcal{P}$ is an individual policy that maximizes expected joint return over partners sampled uniformly from this population:

$$\pi^* \in \text{BR}(\mathcal{P}) = \arg\max_{\pi \in \Pi} \mathbb{E}_{\pi' \sim \mathcal{P}}[J(\pi, \pi')] \tag{2}$$

This objective ensures that π will perform well across a range of different partners. Here we train all best-response policies using PPO (Schulman et al., 2017) with recurrent (LSTM) policy networks that allow the agent to infer its partner's strategy from the interaction history. Note that such a best-response policy may achieve lower reward against a specific partner than a policy specialized for that single partner, reflecting the cost of adapting online to that partner.

3.3 CONVENTIONS

Intuitively, a "convention" is a feature of the joint behavior of two agents, the knowledge of which is sufficient for these agents to coordinate. For example, the convention that all cars should drive on the right allows vehicles to avoid collisions without knowing where each one is going. Here we formalize a *system of conventions* as a collection of subsets $\mathcal{C} = \{C_1, \ldots, C_q\}$ of the policy space Π (with $\Pi = \bigcup_{i=1}^{m} C_i$). Each convention is identified with a set $C_i \in \mathcal{C}$ of strategies that *implement* that convention. Note that the sets C_i need not be disjoint, as there may be adaptive policies that can implement multiple conventions depending on which partner they are paired with. A useful system should allow us to determine whether two agents are compatible based solely on the conventions that they implement. For example, in a task requiring natural language communication between agents, each convention in $\mathcal{C}$ could correspond to a specific language (e.g., English or French). Any two policies π and π' belonging to a single convention $C_i \in \mathcal{C}$ will be able to successfully cooperate because they speak a common language. In this work, we define compatibility in terms of return alone, with compatible policies (those that implement compatible conventions) achieving high joint return when paired together, while pairs of incompatible policies yield low returns. Crucially, all conventions may not be equally good, with some yielding substantially higher joint return than others. *ConventionPlay* leverages the *cross-play matrix* of its initial training population $\mathcal{P}$ to identify conventions. For a finite population of m policies, the cross-play matrix $\boldsymbol{X} \in \mathbb{R}^{m \times m}$ is defined such that $X_{ij} = J(\pi_i, \pi_j)$ for each $i, j \in \{1, \ldots, m\}$.

4 DISCOVERING CONVENTIONS

To simplify the task of discovering a system of conventions over the full policy space Π (Section 3.3), we focus on partitioning a known, finite population of policies into disjoint conventions. We implement two algorithms for convention discovery that compute partitionings based on the cross-play matrix induced by these populations.

4.1 PARTITIONING INTO CONVENTIONS

Given a finite population $\mathcal{P}$ of m policies, we first compute a cross-play matrix $\boldsymbol{X}$, which captures the varying degrees of compatibility between members of the population. We consider two approaches to identifying an underlying system of conventions $\mathcal{C}$, both of which attempt to find systems that serve as a compact representation of $\boldsymbol{X}$. The first approach clusters the rows (or equivalently the columns) of $\boldsymbol{X}$ into discrete sets which become the conventions. The second uses a matrix factorization technique to compute a system of conventions that can be used to reconstruct $\boldsymbol{X}$.

Spectral Clustering A useful system of conventions should allow us to determine appropriate responses to a policy based only on the knowledge of the convention that policy is following. A natural approach to convention discovery then is to cluster policies that have similar compatibility structures, and treat these clusters as conventions. We treat each row $\boldsymbol{X}_{i,:}$ as a feature vector representing policy π_i's interaction profile, and apply spectral clustering (Ng et al., 2001) to partition policies into groups with similar interaction signatures (Ni et al., 2026). Conventions emerge naturally as clusters of policies that achieve comparable returns with the same partners, such that a single policy should be able to achieve high return when paired with any member of a given cluster. The clustering approach requires specifying the number of conventions q as a hyperparameter, which can be determined through eigengap analysis, silhouette analysis, or domain knowledge about the task structure.

Matrix Factorization We also evaluate the use of low-rank matrix factorization to discover conventions based on $\boldsymbol{X}$. We approximately decompose $\boldsymbol{X}$ as $\boldsymbol{X} \approx \boldsymbol{Z}\boldsymbol{W}\boldsymbol{Z}^\top$, where $\boldsymbol{Z} \in [0, 1]^{m \times q}$ records soft membership weights indicating how strongly each policy belongs to each convention, and $\boldsymbol{W} = \text{diag}(\boldsymbol{w})$ captures the expected return for each convention when two policies that implement the same convention are paired together. The reconstructed cross-play matrix is then $\bar{\boldsymbol{X}}_{ij} = \sum_s Z_{is} W_s Z_{js}$. $\boldsymbol{Z}$ and $\boldsymbol{W}$ are found via gradient descent on the mean squared reconstruction error $\|\boldsymbol{X} - \bar{\boldsymbol{X}}_{ij}\|$. Finally, each policy is assigned to the convention for which it has the highest membership score. Like spectral clustering, this approach requires specifying the rank q (the number of conventions) as a hyperparameter. The requirement that $\boldsymbol{W}$ be a diagonal matrix encodes two

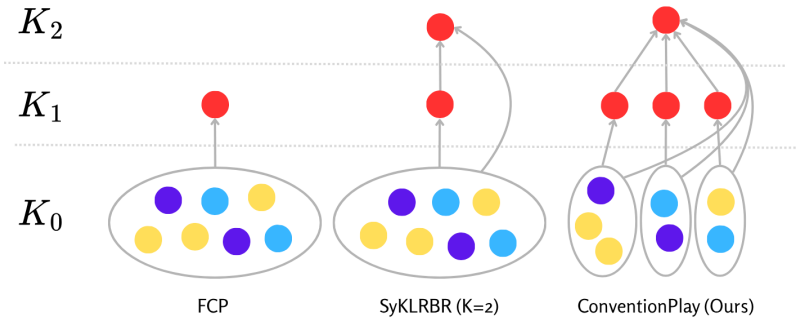


Figure 1: Comparison of the *ConventionPlay* pipeline with existing ad-hoc collaboration methods. Each circle represents an agent. Colors at the K_0 level indicate the convention to which each agent adheres. Adaptive agents are shown in red with an arrow from their training population. While baseline approaches diversify the K_0 population, we introduce diversity among K_1 agents to force the K_2 agent to infer its partner's limited capability set.

strong assumptions about the target task: 1) a policy's compatibility with one convention is independent of its compatibility with another, and 2) any two agents that follow the same convention will be compatible with one another. There are tasks in which these assumptions do not hold, for example, tasks where each player must take on a distinct role (and there is no environmental signal to coordinate these role assignments). Each of our experimental tasks, however, approximately satisfy these assumptions.

4.2 Convention-Rebalanced Training

We leverage the discovered convention system to improve the effectiveness of the generalist policy learned via best-response training, which depends on the distribution of policies within $\mathcal{P}$. The existing diversity methods discussed in Section 2 train against the empirical distribution of a learned policy population, which may be skewed across convention labels – common conventions over-represented while sub-optimal or hard-to-discover conventions under-represented. However, human biases over conventions may differ substantially from the biases that emerge in RL-discovered populations. We posit that a test-time partner is equally likely to follow any convention for the task, and use a discovered convention system to correct this potential imbalance in the training population. Given a population $\mathcal{P}$ and its convention partition $\{C_1, \ldots, C_q\}$, we replace sampling $\pi \sim \mathcal{P}$ with a rebalanced procedure: first sample a convention C_i uniformly from $\{C_1, \ldots, C_q\}$, then sample a policy uniformly at random, with replacement, from C_i. This gives every convention equal weight in the training distribution regardless of its size or frequency in $\mathcal{P}$. A best-response agent is then trained against the rebalanced population $\mathcal{P}'$ rather than $\mathcal{P}$. Experimental results on convention-rebalanced training are discussed in Section 7.

5 ConventionPlay

Different approaches to training policies for ad-hoc collaboration can be distinguished by the assumptions they make about the target population of potential partners that they will eventually need to interact with. Convention-rebalanced training (4.2) models our assumption that a partner may hold any single convention with equal likelihood, but still assumes each partner is limited to just one convention. We now consider that different partners will have varying degrees of ability in coordinating across conventions. The population model of *ConventionPlay* is described using a variant of the Cognitive Hierarchy (CH) (Camerer et al., 2003) model. Informally, policies at level $k \geq 1$ of the CH model are best-responses to a mixture over policies at the previous levels $0, \ldots, k-1$. *ConventionPlay* assumes that members of the target population will sit at levels 0 or 1 of the hierarchy, and that the final policy it outputs sits at level 2. We denote a level-k policy as $\pi^{(k)}$, and construct populations at each level iteratively. We describe the assumptions made at each level of the hierarchy below, and highlight how the assumptions made by *ConventionPlay* differ from those made by the FCP (Strouse et al., 2021) and SyKLRBR (Cui et al., 2022) baselines; this is illustrated in Figure 1.

Level-0 (K_0): Fixed Convention Agents. We would expect a shared convention to emerge among a single population of agents that routinely interact with one another. These agents are brittle: they expect a partner to match their specific mode of coordination. Multiple isolated populations can in turn converge on distinct conventions. The K_0 population, $\mathcal{P}^{(0)}$, represents this aggregate: diverse as a population, but composed of individually fixed-convention agents. Population diversity methods such as MEP, LIPO, and CoMeDi generate and train against a K_0 population alone. In our implementation of *ConventionPlay* the members of the K_0 population are trained using PPO in self-play.

Level-1 (K_1): Adaptive Agents with Limited Capabilities. Training only against a K_0 population ignores the possibility that some partners will be able to adapt to multiple conventions. A solution is to incorporate agents at the K_1 level of the hierarchy into the training population. SyKLRBR, for example, trains against a K_1 population that includes best-responses to the uniform distribution over the full K_0 population $\mathcal{P}^{(0)}$. Realistic partners, however, may have varying capabilities: they may adapt to multiple conventions, but not all. *ConventionPlay* captures this possibility by assuming that each K_1 agent only encounters a restricted subset of the possible conventions for the task. For each K_1 agent $\pi_i^{(1)} \in \mathcal{P}^{(1)}$, we define a *capability set* $\mathcal{K}_i \subset \mathcal{C}$ as the non-empty subset of conventions agent i supports. The training distribution for $\pi_i^{(1)}$ is constructed by sampling a training set of policies $\mathcal{P}_i$ from the conventions in its capability set $\mathcal{K}_i$:

$$\mathcal{P}_i = \bigcup_{C \in \mathcal{K}_i} \text{Sample}(C, m), \tag{3}$$

where $\text{Sample}(C, m)$ draws m policies from convention C. The agent is then defined as the best-response to its specific training population: $\pi_i^{(1)} \in \text{BR}(\mathcal{P}_i)$. We create a population of K_1 agents, $\mathcal{P}^{(1)}$, by randomly sampling capability sets of size $1 < |\mathcal{K}_i| < |\mathcal{C}|$, ensuring coverage of limited-capability partners while maintaining the constraint that no agent supports all conventions.

Level-2 (K_2): Robust Ad-hoc Collaborators. The K_2 agent, $\pi^{(2)}$, is trained to be an optimal collaborator by setting its training population $\mathcal{P}^{(2)}$ to be the union of the fixed-convention K_0 partners and adaptive K_1 partners: $\mathcal{P}^{(2)} = \mathcal{P}^{(0)} \cup \mathcal{P}^{(1)}$. This objective incentivizes $\pi^{(2)}$ to adopt a *probing-and-steering* strategy; it must interact with a partner to infer whether they are a restricted K_0 agent, requiring strict adherence to a fixed convention, or an adaptive K_1 agent, allowing the K_2 agent to steer the team toward the highest-value convention within this partner's specific capability set; the full training procedure is summarized in Algorithm 1 (Appendix C).

6 Experimental Setup

This section introduces the environments and evaluation protocol used to train and test our ad-hoc agents; further details and visualizations for all environments are provided in Appendix A.

6.1 Environments

We evaluate our approach on multiple versions of three multi-agent environments where coordination on shared conventions is critical. To test robustness across different payoff structures, we configure three variants of each environment: uniform payoffs where all conventions yield equal returns (*non-diff*), partial differentiation where some conventions dominate others (*partial*), and full differentiation where each convention has a unique value (*full*).

Repeated Matrix Game. A coordination task where agents must select matching actions across twenty trials. Agents observe a limited three step action history of their partners. Figure 2 shows payoff matrices for uniform and fully differentiated variants. Here, the expected convention is a single coordinated action.

Point Mass Rendezvous (PMR). A spatial coordination task (Charakorn et al., 2023) where agents must rendezvous at one of six pre-configured landmarks. Agents observe their position, velocity, and distance to landmarks, along with their partner's velocity. The reward signal is sparse, forcing agents to commit to rendezvous at a specific landmark before observing rewards for it. Here, the expected

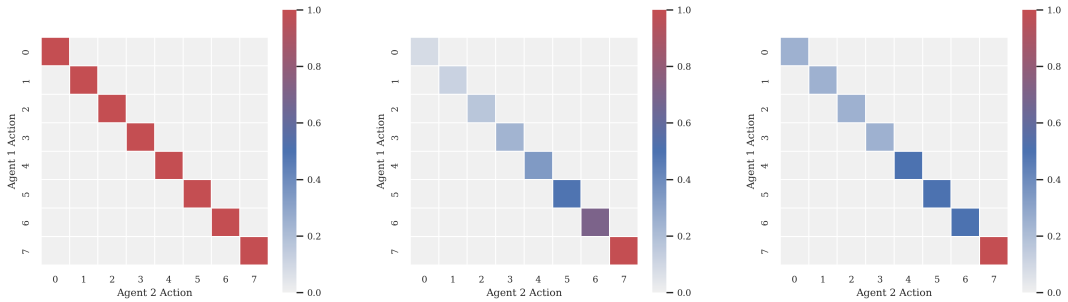


Figure 2: Payoff matrices for the Repeated Matrix Game. Left: Uniform payoffs across conventions. Middle: Partially differentiated payoffs across conventions. Right: Fully differentiated payoffs.

convention is the landmark agents rendezvous at. Like the matrix game, we vary rewards issued at each landmark to create uniform and differentiated convention structures.

Task-Relay. A novel two-player cooperation game modeling an industrial manufacturing task in which two agents build an object by switching to and applying the appropriate tools. Optimal policies partition labor between the agents so that idling time is minimized, effectively simulating a resource allocation problem. The object's evolution is defined by a task graph with five goal states, each representing a distinct way of manufacturing the object given a set of tools, and each yielding a unique cumulative reward. Here, the expected convention is a choice of which sequence of tool applications to pursue. Further details for the Task-Relay environment are provided in Appendix A.3.

6.2 Evaluation Protocol

Our evaluation protocol is designed to probe ad-hoc teamwork capabilities across every level of the cognitive hierarchy. For each environment, we construct three evaluation populations. First, $\mathcal{P}^{(0)}_{\text{test}}$ uniformly covers the task's known conventions leveraging domain knowledge. Second, $\mathcal{P}^{(1)}_{\text{sampled}}$ applies the capability-set sampling procedure of Section 5 directly to $\mathcal{P}^{(0)}_{\text{test}}$, using its known ground-truth convention labels in place of a discovered partition. Third, $\mathcal{P}^{(1)}_{\text{heur}}$ consists of hand-coded heuristic agents that implement limited-capability behavior directly – for example, a stubborn tit-for-tat agent in the repeated matrix game that matches its partner's action only when that action falls within a specific capability set. Environment-specific details for developing $\mathcal{P}^{(0)}_{\text{test}}$ and $\mathcal{P}^{(1)}_{\text{heur}}$ are provided in Appendix A. Because $\mathcal{P}^{(0)}_{\text{test}}$, $\mathcal{P}^{(1)}_{\text{sampled}}$, and $\mathcal{P}^{(1)}_{\text{heur}}$ are constructed independently of any training population, they may include behaviors or behavioral noise that no evaluated K_2 method encountered during training; performance against these populations reflects out-of-distribution generalization.

We compare against **FCP** (Strouse et al., 2021), which introduces behavioral diversity through intermediate checkpoints, and **SyKLRBR** (Cui et al., 2022), representing state-of-the-art cognitive hierarchy modeling for zero-shot coordination; for structural parity, we evaluate a two-level SyKLRBR hierarchy grounded in our learned K_0 population in addition to the canonical five-level hierarchy. We also include the population-diversity methods **MEP**, **LIPO**, and **CoMeDi**. For CoMeDi, we evaluate both the original implementation (CoMeDi (BC)) and a variant, CoMeDi (BR), that replaces its joint self-play and imitation objective with a single best-response agent trained on our recurrent architecture. Appendix B.1 details the motivation for all baselines.

7 Results and Discussion

We first validate convention discovery in isolation, through an experiment in convention-rebalancing. We then evaluate our generalist K_2 agent. For each experiment we test convention discovery with both spectral clustering (Spectral) and matrix factorization (MatFact). All results are reported in terms of normalized reward and coordination efficiency, defined respectively in Appendices D.2 and D.3. Hyperparameters and implementation details for all experimental runs are reported in Appendices B and C. We additionally study the effect of K_1 population size on *ConventionPlay* in Appendix D.5.

Convention Rebalancing We validate the convention identification and rebalancing of Section 4 using a variant of the matrix game in which unequal action-set sizes across conventions cause K_0 training to naturally yield a heavily imbalanced population. Figure 8 compares a best-response agent trained on this imbalanced population against ones trained on rebalanced populations $\mathcal{P}'$, evaluated on a K_0 test population that is uniform over conventions. Rebalancing improves normalized return by a margin of up to 40%, confirming that correcting for skewed convention representation during training improves generalization to an unbiased test-time partner distribution. Full experimental setup and additional figures are provided in Appendix D.1.

Table 1: Mean normalised episodic return on $\mathcal{P}^{(0)}_{\text{test}}$ partners. Each cell reports mean $\pm$ 1 s.d. (unbiased sample standard deviation, $n-1$ denominator) across $n=5$ seeds. Bold marks the row's best value.

Environment	Var	FCP	SyKLRBR (L5)	SyKLRBR (L2)	CoMeDi (BC)	CoMeDi (BR)	LIPO	MEP	ConvPlay (matfact)	ConvPlay (spectral)
Matrix Game	Non-diff	95.16 ± 0.30	95.47 ± 0.17	95.55 ± 0.14	67.64 ± 0.21	68.86 ± 4.86	**95.58 ± 0.09**	69.44 ± 1.31	94.64 ± 0.15	94.42 ± 0.37
	Partial	41.76 ± 0.14	41.76 ± 0.20	**41.82 ± 0.20**	31.45 ± 0.11	30.57 ± 2.75	38.70 ± 0.10	34.18 ± 3.36	40.69 ± 1.31	41.21 ± 0.16
	Full	**37.44 ± 0.06**	37.23 ± 0.76	37.42 ± 0.52	29.00 ± 0.09	29.95 ± 0.67	33.90 ± 0.74	30.25 ± 0.83	36.35 ± 0.44	36.56 ± 0.57
PMR	Non-diff	89.42 ± 7.94	**97.54 ± 0.24**	97.46 ± 0.14	66.48 ± 3.18	80.17 ± 11.29	90.64 ± 6.80	52.11 ± 7.26	78.21 ± 7.14	91.07 ± 9.01
	Partial	60.33 ± 6.97	**66.00 ± 0.15**	65.98 ± 0.19	35.49 ± 5.25	53.16 ± 6.44	43.60 ± 0.27	53.96 ± 10.76	64.85 ± 2.12	65.80 ± 0.26
	Full	56.49 ± 6.22	54.27 ± 6.05	32.68 ± 7.56	16.56 ± 7.40	37.81 ± 10.94	47.17 ± 7.54	38.17 ± 2.51	57.92 ± 3.62	**60.58 ± 0.11**
Task Relay	Non-diff	**90.00 ± 0.87**	87.56 ± 3.60	87.06 ± 4.79	62.93 ± 4.57	38.21 ± 9.79	74.11 ± 5.43	78.92 ± 4.52	79.36 ± 3.55	83.12 ± 6.32
	Partial	78.17 ± 1.20	80.56 ± 2.57	**81.31 ± 0.81**	46.77 ± 9.67	61.41 ± 7.56	57.39 ± 1.30	68.14 ± 3.52	70.83 ± 4.04	74.28 ± 6.15
	Full	38.88 ± 9.05	44.26 ± 4.43	42.68 ± 5.95	16.84 ± 1.35	19.86 ± 6.65	18.88 ± 4.90	3.10 ± 1.55	31.13 ± 12.45	**44.70 ± 4.44**

Table 2: Mean normalised episodic return on $\mathcal{P}^{(1)}_{\text{sampled}}$ partners. Each cell reports mean $\pm$ 1 s.d. (unbiased sample standard deviation, $n-1$ denominator) across $n=5$ seeds. Bold marks the row's best value.

Environment	Var	FCP	SyKLRBR (L5)	SyKLRBR (L2)	CoMeDi (BC)	CoMeDi (BR)	LIPO	MEP	ConvPlay (matfact)	ConvPlay (spectral)
Matrix Game	Non-diff	65.93 ± 2.69	63.78 ± 2.06	63.64 ± 2.56	66.48 ± 1.59	70.66 ± 3.31	60.41 ± 5.12	71.98 ± 4.02	**86.19 ± 1.76**	85.52 ± 1.91
	Partial	26.03 ± 3.90	25.65 ± 2.41	25.56 ± 2.50	47.05 ± 0.21	38.23 ± 1.58	30.09 ± 0.91	39.86 ± 2.31	46.50 ± 0.93	**50.47 ± 1.52**
	Full	24.03 ± 1.31	25.74 ± 4.37	23.97 ± 4.24	48.12 ± 0.33	33.33 ± 1.22	33.47 ± 2.68	46.27 ± 0.89	51.52 ± 1.37	**52.63 ± 0.41**
PMR	Non-diff	89.33 ± 5.75	94.39 ± 0.37	**94.43 ± 0.52**	63.70 ± 3.95	80.00 ± 8.47	90.10 ± 5.53	62.91 ± 8.20	86.30 ± 4.50	93.46 ± 2.14
	Partial	58.44 ± 5.35	62.25 ± 1.10	62.93 ± 1.93	40.57 ± 3.44	57.89 ± 7.25	47.82 ± 1.19	56.30 ± 8.30	**74.82 ± 1.16**	72.58 ± 1.24
	Full	53.73 ± 5.81	53.08 ± 4.95	36.07 ± 3.68	20.99 ± 10.27	39.29 ± 9.71	47.62 ± 5.66	43.74 ± 2.96	65.54 ± 1.41	**66.11 ± 1.28**
Task Relay	Non-diff	82.72 ± 2.43	8.56 ± 8.64	5.90 ± 6.51	53.84 ± 9.61	2.40 ± 4.65	7.74 ± 9.18	18.17 ± 20.86	75.09 ± 21.41	**83.22 ± 4.52**
	Partial	43.40 ± 36.42	2.67 ± 0.90	2.88 ± 1.08	41.34 ± 5.81	53.76 ± 27.75	1.62 ± 1.41	22.83 ± 14.01	62.81 ± 8.21	**75.12 ± 11.31**
	Full	14.45 ± 23.57	3.30 ± 3.66	2.29 ± 1.94	8.93 ± 1.14	4.43 ± 3.53	0.56 ± 0.43	1.18 ± 0.42	32.38 ± 20.70	**53.66 ± 8.03**

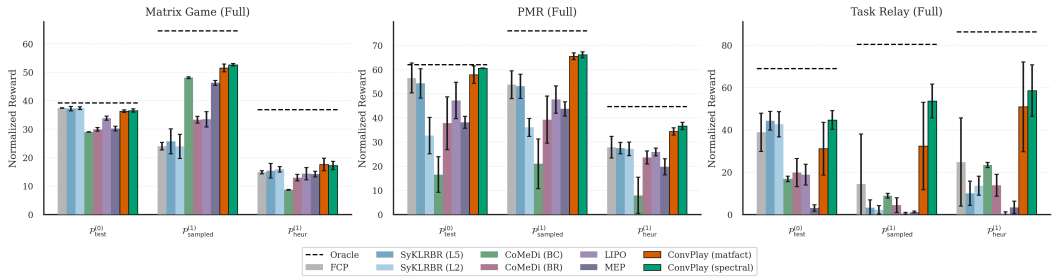


Figure 3: Normalized reward under full payoff differentiation across all three environments (Matrix Game, PMR, Task-Relay) and all three evaluation populations ($\mathcal{P}^{(0)}_{\text{test}}$, $\mathcal{P}^{(1)}_{\text{sampled}}$, $\mathcal{P}^{(1)}_{\text{heur}}$). The same pattern can be observed in all settings: *ConventionPlay* shows a modest cost against fixed-convention partners, but outperforms baselines against various types of adaptive partners.

Performance on Test Populations The results across all three evaluation populations and environments—Tables 1, 2, 7—follow a similar shape: parity with baselines when conventions are non-differentiated, and both a 10–50% gain against adaptive partners and a modest 0–5% reduction against fixed-convention partners when conventions are differentiated (Figure 3). Non-differentiated settings pose little challenge, since convergence to any shared convention is equally good. The tradeoff in the differentiated setting follows directly from the probing-and-steering behavior induced by our training population: baseline agents lack any mechanism for distinguishing a partner capable of a higher-value convention from one that is not, and so commit to the first shared convention they discover, even when a superior alternative lies within the partner's capability set. *ConventionPlay*, by contrast, first probes for a higher-value convention before committing—a strategy that pays off against adaptive partners, and costs only a little against rigid ones, since the agent cannot immediately

tell the two apart. This performance gap is most pronounced in Task-Relay, where baselines perform particularly poorly against $\mathcal{P}^{(1)}_{\text{sampled}}$ and $\mathcal{P}^{(1)}_{\text{heur}}$ partners. This stems from Task-Relay's susceptibility to a co-adaptation deadlock: each agent repeats non-committal actions, waiting for the other to commit first—precisely the dynamic *ConventionPlay* is designed to break.

Analysis of Probing We examine the emergent probing behavior by analyzing trajectories from interactions in the Matrix Game under fully differentiated payoffs. The left panel of Figure 4 shows each method's interaction with a fixed-convention partner committed to the lowest-value convention – a scenario in which the ad-hoc agent is most likely to "probe": deviate from its partner's current action in an attempt to induce the partner to adapt toward a higher-value convention. Whereas most baseline methods converge to the partner's convention within 1–3 time steps, *ConventionPlay* probes for 3–6 time steps before converging, explaining its modest reward cost against $\mathcal{P}^{(0)}_{\text{test}}$. The right panel evaluates ad-hoc methods coordinating with a modified version of the learned policies in $\mathcal{P}^{(1)}_{\text{sampled}}$ in which partner policies are run for a fixed number of steps before committing to their last action, allowing us to trace *ConventionPlay*'s performance across a spectrum of partner adaptivity. At low adaptivity, the agent has little time to gather information about its partner and must coordinate based on what it has already observed; this setting effectively reduces the partner to a K_0 agent, and *ConventionPlay* matches the expected 0–5% performance drop compared to leading baselines. As adaptivity increases, however, *ConventionPlay* increasingly outperforms all baselines by effectively inferring the partner's hidden capabilities. A similar probing behavior can be visualized with individual PMR trajectories and is shown in Appendix D.4.

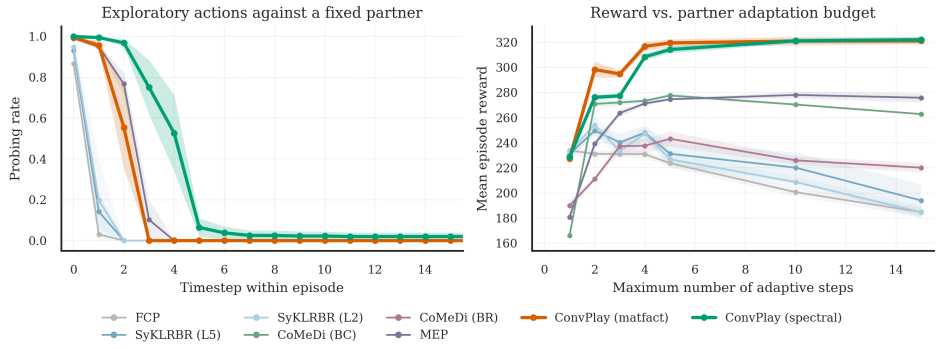


Figure 4: Left: fraction of various methods applying a probing (i.e., uncoordinated) action to a fixed-convention partner; *ConventionPlay* can be seen to probe for significantly longer, explaining its reduced performance with the $\mathcal{P}^{(0)}$ population. Right: performance on a modified $\mathcal{P}^{(1)}_{\text{sampled}}$ in which partners adapt for only a fixed number of steps before committing, tracing *ConventionPlay*'s advantage across a spectrum of partner adaptivity.

8 Conclusion and Future Work

In this work, we have introduced *ConventionPlay*, a reinforcement learning-based approach for ad-hoc collaboration that moves beyond purely reactive partner adaptation. By introducing strategic diversity into the K_1 layer of a cognitive hierarchy, agents learn to dynamically balance leading and following. Our experimental results demonstrate that when partners themselves are capable of adapting across multiple conventions, *ConventionPlay* outperforms existing methods by actively probing a partner's capabilities and steering the team toward high-value conventions.

Limitations A primary limitation in our approach is the reliance on cross-play returns for convention identification. Incorporating other behavioral signals—state-action trajectories or policy similarity (Lupu et al., 2021)—could provide more nuanced representations of agent capabilities, which would in turn improve the construction of capability-limited K_1 training sets. Additionally, our evaluation focuses on relatively small populations of conventions; scaling to environments with many overlapping or complex convention structures remains an open question.

AI USE STATEMENT

We used AI in a number of targeted ways, and all AI-assisted work was heavily reviewed and revised by the authors.

For software development, we used AI to build a lightweight harness for parallelizing large experimental runs with a declared dependency graph, to vectorize the Task-Relay environment, and to write utilities converting experiment artifacts into formatted data and visualizations (e.g. Table 1, Figures 5 and 10). Prompts were narrowly scoped, producing small diffs that we carefully reviewed.

When developing experiments, we used AI to help debug results and suggest hyperparameters, primarily to maximize throughput on available compute (e.g., data collections rate and batch sizes).

We used AI for grammar and language refinement while writing this document.

REFERENCES

Matteo Bettini, Ryan Kortvelesy, Jan Blumenkamp, and Amanda Prorok. Vmas: A vectorized multi-agent simulator for collective robot learning. *The 16th International Symposium on Distributed Autonomous Robotic Systems*, 2022.

Colin F. Camerer, Teck Ho, and Juin-Kuan Chong. A cognitive hierarchy theory of one-shot games and experimental analysis. *SSRN Electronic Journal*, 2003. ISSN 1556-5068. doi: 10.2139/ssrn.411061. URL http://dx.doi.org/10.2139/ssrn.411061.

Micah Carroll, Rohin Shah, Mark Ho, Tom Griffiths, Sanjit Seshia, Pieter Abbeel, and Anca Dragan. On the utility of learning about humans for human-ai coordination. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/f5b1b89d98b7286673128a5fb112cb9a-Paper.pdf.

Rujikorn Charakorn, Poramate Manoonpong, and Nat Dilokthanakul. Generating diverse cooperative agents by learning incompatible policies. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=UkU05GOH7_6.

Brandon Cui, Hengyuan Hu, Luis Pineda, and Jakob Nicolaus Foerster. K-level reasoning for zero-shot coordination in hanabi. *ArXiv*, abs/2207.07166, 2022. URL https://api.semanticscholar.org/CorpusID:246996266.

Brandon Cui, Andrei Lupu, Samuel Sokota, Hengyuan Hu, David J Wu, and Jakob N Foerster. ADVERSARIAL DIVERSITY IN HANABI. 2023.

Jakob N. Foerster, Richard Y. Chen, Maruan Al-Shedivat, Shimon Whiteson, P. Abbeel, and Igor Mordatch. Learning with opponent-learning awareness. In *Adaptive Agents and Multi-Agent Systems*, 2017. URL https://api.semanticscholar.org/CorpusID:8708073.

Hengyuan Hu, Adam Lerer, Alex Peysakhovich, and Jakob Foerster. "Other-play" for zero-shot coordination. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 4399–4410. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/hu20a.html.

Hengyuan Hu, Adam Lerer, Brandon Cui, Luis Pineda, Noam Brown, and Jakob Foerster. Off-belief learning. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 4369–4379. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/hu21c.html.

Natasha Jaques, Angeliki Lazaridou, Edward Hughes, Caglar Gulcehre, Pedro Ortega, Dj Strouse, Joel Z. Leibo, and Nando De Freitas. Social influence as intrinsic motivation for multi-agent deep reinforcement learning. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine*

Learning Research, pp. 3040–3049. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/jaques19a.html.

Adam Lerer and Alexander Peysakhovich. Learning existing social conventions via observationally augmented self-play. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, AIES '19, pp. 107–114, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450363242. doi: 10.1145/3306618.3314268. URL https://doi.org/10.1145/3306618.3314268.

Chris Lu, Timon Willi, Christian Schroeder de Witt, and Jakob Foerster. Model-Free Opponent Shaping, November 2022.

Andrei Lupu, Brandon Cui, Hengyuan Hu, and Jakob Foerster. Trajectory Diversity for Zero-Shot Coordination. In *Proceedings of the 38th International Conference on Machine Learning*, pp. 7204–7213. PMLR, July 2021.

Reuth Mirsky, Ignacio Carlucho, Arrasy Rahman, Elliot Fosong, William Macke, Mohan Sridharan, Peter Stone, and Stefano V. Albrecht. A survey of ad hoc teamwork research, 2022. URL https://arxiv.org/abs/2202.10450.

Andrew Y. Ng, Michael I. Jordan, and Yair Weiss. On spectral clustering: analysis and an algorithm. In *Proceedings of the 15th International Conference on Neural Information Processing Systems: Natural and Synthetic*, NIPS'01, pp. 849–856, Cambridge, MA, USA, 2001. MIT Press.

Andrew Ni, Simon Stepputtis, Stefanos Nikolaidis, Michael Lewis, Katia P. Sycara, and Woojun Kim. Theory of mind guided strategy adaptation for zero-shot coordination. In *Proceedings of the 25th International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '26, pp. 1810–1819, Richland, SC, 2026. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 9798400723179. doi: 10.65109/UMCV8852. URL https://doi.org/10.65109/UMCV8852.

Frans A. Oliehoek and Christopher Amato. *A Concise Introduction to Decentralized POMDPs*. SpringerBriefs in Intelligent Systems. Springer International Publishing, Cham, 2016. ISBN 978-3-319-28927-4 978-3-319-28929-8. doi: 10.1007/978-3-319-28929-8.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Bidipta Sarkar, Andy Shih, and Dorsa Sadigh. Comedi – Diverse Conventions for Human-AI Collaboration, October 2023.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Andy Shih, Arjun Sawhney, Jovana Kondic, Stefano Ermon, and Dorsa Sadigh. On the critical role of conventions in adaptive human-ai collaboration. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=8Ln-Bq0mZcy.

Peter Stone, Gal Kaminka, Sarit Kraus, and Jeffrey Rosenschein. Ad hoc autonomous agent teams: Collaboration without pre-coordination. *Proceedings of the AAAI Conference on Artificial Intelligence*, 24(1):1504–1509, Jul. 2010. doi: 10.1609/aaai.v24i1.7529. URL https://ojs.aaai.org/index.php/AAAI/article/view/7529.

DJ Strouse, Kevin McKee, Matt Botvinick, Edward Hughes, and Richard Everett. Collaborating with humans without human data. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 14502–14515. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/797134c3e42371bb4979a462eb2f042a-Paper.pdf.

Ardi Tampuu, Tambet Matiisen, Dorian Kodelja, Ilya Kuzovkin, Kristjan Korjus, Juhan Aru, Jaan Aru, and Raul Vicente. Multiagent cooperation and competition with deep reinforcement learning. *PloS one*, 12(4):e0172395, 2017.

Rui Zhao, Jinming Song, Yufeng Yuan, Haifeng Hu, Yang Gao, Yi Wu, Zhongqian Sun, and Wei Yang. Maximum entropy Population-Based training for zero-shot human-AI coordination. *Proc. Conf. AAAI Artif. Intell.*, 37(5):6145–6153, June 2023.

A Environments

This section describes environments used for evaluation and specifically details how domain knowledge is leveraged to construct $\mathcal{P}^{(0)}_{\text{test}}$ and $\mathcal{P}^{(1)}_{\text{heur}}$ populations.

A.1 Repeated Matrix Game

The repeated matrix game requires agents to coordinate on matching actions across 20 trials. Figure 2 shows payoff matrices for the uniform and fully differentiated variants used in our experiments.

Observation space. Each agent observes its partner's three most recent actions.

Action space. Each agent selects one of N discrete actions per trial; in our configured variant, $N = 8$.

Evaluation agents. For the $\mathcal{P}^{(0)}_{\text{test}}$ set, we create hard-coded policies that each play a single fixed action throughout the episode. For the $\mathcal{P}^{(1)}_{\text{heur}}$ set, we implement a stubborn tit-for-tat agent that plays a fixed default action until it observes its partner playing an action within its own capability set, at which point it switches to mirroring that action for the remainder of the episode. We instantiate many such agents, parameterized by a pair (a_0, a_1) of default and candidate actions from $\{1, \ldots, N\}$ – e.g., $(0, 1)$, $(0, 2)$, $(1, 3)$, and so on – covering a range of default/candidate combinations.

A.2 Point Mass Rendezvous (PMR)

Point Mass Rendezvous (PMR) is a spatial coordination task (Charakorn et al., 2023) implemented using VMAS (Bettini et al., 2022), in which agents must rendezvous at one of 6 pre-configured landmarks.

Observation space. Each agent observes its own position and velocity, its distance to each landmark, and its partner's velocity.

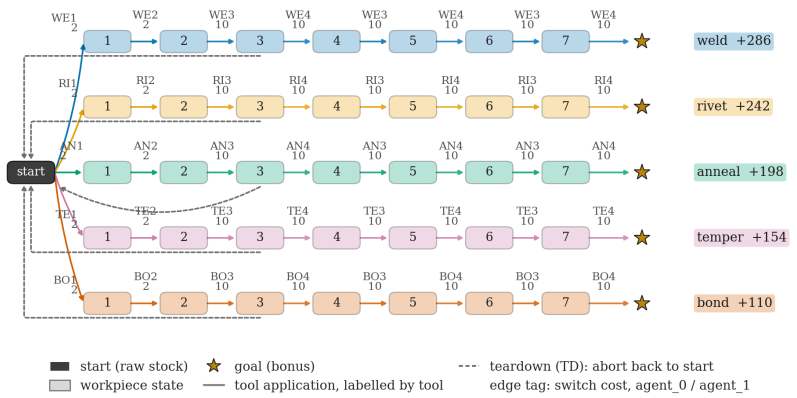
Action space. Each agent outputs a continuous 2D action controlling its planar acceleration.

Evaluation agents. For the $\mathcal{P}^{(0)}_{\text{test}}$ set, we create policies that deterministically drive to each one of the pre-configured landmarks. For the $\mathcal{P}^{(1)}_{\text{heur}}$ set, we implement a stubborn tit-for-tat agent that drives toward a default landmark for a fixed initial period before potentially switching to a landmark in its capability set if the partner is closest to it. We instantiate many such agents, parameterized by a pair (l_0, l_1) of default and candidate landmarks – e.g., $(0, 1)$, $(0, 2)$, $(1, 3)$, and so on – covering a broad range of default/candidate combinations.

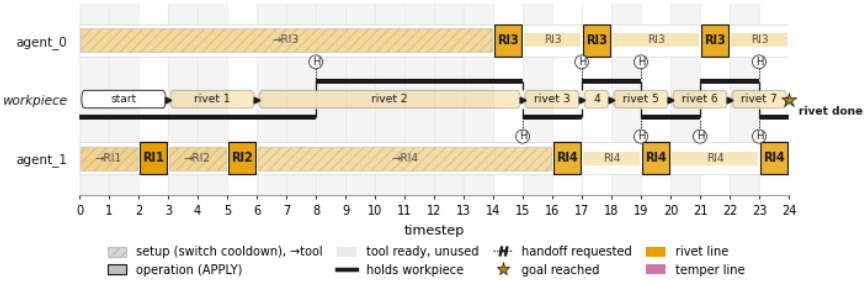
A.3 Task-Relay

Task-Relay requires two agents to jointly build a single object by applying tools that advance the object along a directed graph of object states; only one agent may hold the object at a time, and only the holder may apply a tool. Figure 5a shows the task graph used in our experiments; the graph has five manufacturing paths (weld, rivet, anneal, temper, bond) branching from a shared start state. Switching tools is not instantaneous, so the two agents must interleave staging the next tool and holding the object to avoid wasted time.

Observation space. Each agent observes its own and its partner's equipped tool, in-progress switch target, and remaining switch cooldown; whether it holds the object; a one-hot encoding of the object's



(a) The task graph for constructing an object underlying Task-Relay. The dark node is the shared start (raw stock) state; each colored row is a distinct strategy, advancing through a chain of intermediate workpiece states via applications of its associated tool, and terminating in a goal state (star) labeled with its completion bonus and tool switch cost. Dashed arrows denote a teardown action that aborts a strategy back to the start state.



(b) Example resolved trajectory for the *rivet* strategy, showing each agent's tool timeline and the workpiece's progression over the episode.

Figure 5: Task-Relay task structure and an example resolved trajectory.

current node; which tools have a live edge at the current node and one hop ahead; and the fraction of the episode elapsed.

Action space. Each agent selects one of NOOP, APPLY, HANDOFF, or SWITCH to one of K tools per step; in our configured variant, $K = 21$.

Evaluation agents. For the $\mathcal{P}^{(0)}_{\text{test}}$ set, we train self-play populations under environment variants, where each variant biases self-play toward specializing on a possible manufacturing path. For the $\mathcal{P}^{(1)}_{\text{heur}}$ set, we implement a hard-coded policy that never initiates or contests the line choice made by its partner, but will be a competent partner if the chosen path is within the agent's repertoire. We instantiate both singleton (e.g. $\{\text{weld}\}$) and pair (e.g. $\{\text{weld}, \text{rivet}\}$) repertoires, so scoring optimally requires identifying which line(s) are supported by the partner.

B Baselines and Hyperparameters

B.1 Choice of Baselines

We compare *ConventionPlay* against five baselines: FCP, SyKLRBR, MEP, LIPO, and CoMeDi.

FCP (Strouse et al., 2021) is a standard and widely used baseline for generalist ad-hoc coordination.

SyKLRBR (Cui et al., 2022) is the closest existing baseline to our approach, as it is grounded in cognitive hierarchy reasoning; comparing against it isolates the specific contribution of *ConventionPlay*'s

capability-limited K_1 population relative to a hierarchy that does not model partner capability limits explicitly.

MEP (Zhao et al., 2023), **LIPO** (Charakorn et al., 2023), and **CoMeDi** (Sarkar et al., 2023) each contribute a distinct method for generating diversity within the K_0 population and represent more recent points of comparison. For CoMeDi specifically, we evaluate two versions. The published version, **CoMeDi (BC)**, trains the ad-hoc agent with a self-play plus behavior-cloning objective that clones the CoMeDi population; this design targets human-AI collaboration settings, where the partner is assumed to abruptly switch strategies mid-episode, and behavior cloning helps the agent track that shift. We additionally construct **CoMeDi (BR)**: a variant that uses the CoMeDi population purely as a source of K_0 diversity, and trains a single best-response agent against it with the same recurrent architecture used for our other baselines and *ConventionPlay* itself; this allows us to disentangle CoMeDi's contribution to population diversity from its approach to leveraging that diversity for ad-hoc collaboration.

B.2 Self-Play and Best-Response Hyperparameters

Self-play policies are trained with DQN (Tampuu et al., 2017) and PPO (Schulman et al., 2017); training uses feedforward networks. All best-response policies are trained with PPO and use recurrent (LSTM) policy networks to handle the partial observability of inferring partner conventions. Additionally for all best-response training, the value networks include an auxiliary head that predicts the partner identity to aid partner identification. All policies were trained using 5 random seeds. For best-response training,

Table 3 reports hyperparameters for self-play (K_0) training, and Table 4 reports hyperparameters for best-response (K_1/K_2) training, by environment.

Table 3: Self-play hyperparameters by environment family.

Hyperparameter	Matrix Game	PMR	Task Relay
Training Loop			
Total Env Frames	12,500	1×10^7	1.5×10^7
Training Iterations	25	50	100
Frames per Iteration	500	2×10^5	1.5×10^5
Minibatch Size	500	4,000	1,500
Minibatches per Iteration	10	50	100
Epochs per Iteration	1.00	1.00	1.00
Buffer Size	5,000	2×10^5	1.5×10^5
Sequence Length	–	–	–
Max Grad Norm	10	2	10
Learner			
Algorithm	IQL	PPO	PPO
Policy Network	MLP [32, 32]	MLP [256, 256]	MLP [256, 256]
Critic Network	–	MLP [256, 256]	MLP [256, 256]
Learning Rate (LR)	0.001	0.0009	0.0009
Adam Epsilon	0.001	1e-05	1e-05
LR Annealed	Yes	Yes	Yes
Discount Factor (γ)	0.99	0.99	0.99
GAE λ	–	0.95	0.95
Clip Epsilon	–	0.2	0.2
Entropy Coefficient	–	0.01	0.01
Critic Loss Coefficient	–	0.5	0.5
Epsilon-Greedy Init	0.8	–	–
Epsilon-Greedy End	0.05	–	–
Exploration Fraction	0.5	–	–
Target Update τ	0.05	–	–
Delayed Target Network	Yes	–	–
TD(λ)	0	–	–

Table 4: Best-response hyperparameters by environment family.

Hyperparameter	Matrix Game	PMR	Task Relay
Training Loop			
Total Env Frames	1.6×10^6	4×10^6	7.5×10^6
Training Iterations	20	5	50
Frames per Iteration	80,000	8×10^5	1.5×10^5
Minibatch Size	5,000	8,000	2,000
Minibatches per Iteration	20	100	75
Epochs per Iteration	1.25	1.00	1.00
Buffer Size	80,000	8×10^5	1.5×10^5
Sequence Length	5	10	10
Max Grad Norm	10	2	10
Learner			
Algorithm	PPO	PPO	PPO
Policy Network	MLP [64, 64] $\rightarrow$ 64 $\rightarrow$ LSTM(128) $\rightarrow$ MLP [64, 64]	MLP [256, 256] $\rightarrow$ 128 $\rightarrow$ LSTM(128) $\rightarrow$ MLP [256, 256]	MLP [64, 64] $\rightarrow$ 64 $\rightarrow$ LSTM(128) $\rightarrow$ MLP [64, 64]
Critic Network	MLP [256, 256]	MLP [256, 256]	MLP [256, 256]
Learning Rate (LR)	0.005	0.001	0.005
Adam Epsilon	1e-05	1e-05	1e-05
LR Annealed	Yes	Yes	Yes
Discount Factor (γ)	0.99	0.99	0.99
GAE λ	0.95	0.95	0.95
Clip Epsilon	0.2	0.3	0.2
Entropy Coefficient	0.1	0.25	0.1
Critic Loss Coefficient	0.5	0.5	0.5

B.3 Baseline Population Parameters

Table 5 reports the K_0 population size used for each method, by environment. Table 6 reports the size of the K_1 test (evaluation) population, shared across all methods, and the number of conventions q used as a hyperparameter for both convention partitioning approaches of Section C.2.

Table 5: K_0 population size for each method, by environment.

Method	Matrix Game	PMR	Task-Relay
MEP	20	12	10
LIPO	20	12	5
CoMeDi (BC) / CoMeDi (BR)	20	12	10
FCP	40	30	30
SyKLRBR	40	30	30
ConventionPlay	40	30	30

Table 6: K_1 test population size and number of conventions q, by environment.

	Matrix Game	PMR	Task-Relay
K_1 test population size	50	12	10
q (number of conventions)	8	6	5

C Experiment Implementation Details

Algorithm 1 summarizes the full *ConventionPlay* training procedure; the following subsections detail each stage.

C.1 K0 Population Construction

We generate our K_0 population by training independent PPO (Schulman et al., 2017) across multiple random seeds. While specialized methods like TrajeDI or LIPO could also be used, we found that random initialization provided sufficient behavioral diversity. This approach does not guarantee that the population will partition into the well-structured system of incompatible policies described in our

Algorithm 1 ConventionPlay

Input: Number of K_1 agents n
$\mathcal{P}^{(0)} \leftarrow$ `TrainK0Population()` ▷ Stage 1: Generate and partition K_0 agents
$\{C_1, \ldots, C_q\} \leftarrow$ `PartitionIntoConventions`$(\mathcal{P}^{(0)})$
$\mathcal{P}^{(1)} \leftarrow \emptyset$
for $j = 1$ **to** n **do** ▷ Stage 2: Train diverse K_1 agents
    Sample capability set $\mathcal{K}_j \subseteq \{C_1, \ldots, C_q\}$ with $1 < |\mathcal{K}_j| < q$
    $\mathcal{P}_j \leftarrow \bigcup_{C \in \mathcal{K}_j}$ `Sample`(C, m)
    $\pi_j^{(1)} \leftarrow \text{BR}(\mathcal{P}_j)$
    $\mathcal{P}^{(1)} \leftarrow \mathcal{P}^{(1)} \cup \{\pi_j^{(1)}\}$
end for
$\pi^{(2)} \leftarrow \text{BR}(\mathcal{P}^{(0)} \cup \mathcal{P}^{(1)})$ ▷ Stage 3: Train robust K_2 agent
return $\pi^{(2)}$

formalism; in some tasks, we may encounter complex, non-transitive compatibility structures. We therefore verify this inter-convention incompatibility empirically to ensure our agents provide a clear foundation of distinct strategies for subsequent training levels.

C.2 Convention Partitioning

For spectral clustering, we first normalize the cross-play matrix $\boldsymbol{X}$ based on the expected reward range for the environment, yielding $\tilde{\boldsymbol{X}}$. We compute an affinity matrix $\boldsymbol{A} \in \mathbb{R}^{m \times m}$ using a standard RBF kernel over each pair of agents' normalized feature vectors, $A_{ij} = \exp(-\gamma \|\tilde{\boldsymbol{X}}_{i,:} - \tilde{\boldsymbol{X}}_{j,:}\|^2)$ with $\gamma = 1.0$. Clustering is then computed via the off-the-shelf implementation from scikit-learn (Pedregosa et al., 2011).

For matrix factorization, we parameterize $\boldsymbol{Z}$ via a learned logit matrix $\boldsymbol{E} \in \mathbb{R}^{m \times q}$, with entries passed through a temperature-scaled softmax: $\boldsymbol{Z} = \text{softmax}(\boldsymbol{E}/\tau)$. We optimize $\boldsymbol{E}$ and $\boldsymbol{W}$ via gradient descent (Adam) to minimize a mean-squared reconstruction loss between $\boldsymbol{X}$ and $\bar{\boldsymbol{X}}$. In this work, temperature τ is annealed from $1.0 \rightarrow 0.01$ during training, implementing a relaxation from soft membership assignments (high τ) to near one-hot convention assignments (low τ). This need not be the case, however; an adjusted temperature or alternate activation function could allow downstream applications to take advantage of overlapping or probabilistic convention structures.

For both convention partitioning approaches (spectral clustering, and matrix factorization), we set the expected number of conventions q based on domain knowledge (detailed in Table 6, Appendix B.3). However, this can be extended to a parameter-free approach via silhouette analysis or eigen decomposition of the cross-play matrix.

C.3 K1 Population Construction

When constructing the K_1 training population, we sample capability sets $\mathcal{K}_j$ for each of the n agents such that $1 < |\mathcal{K}_j| < q$. To ensure diversity in partner flexibility while maintaining computational efficiency, we prioritize smaller capability sets during sampling. Specifically, we sample capability sets in order of increasing size: we first sample agents with 2-convention capability sets, then 3-convention sets, and so on, until we have constructed the full K_1 population of size n. This stratified sampling strategy ensures that the final K_2 agent encounters partners with varying degrees of adaptability, from those capable of adapting across just two conventions to those capable of adapting across nearly all conventions.

For each sampled capability set $\mathcal{K}_j$, we then construct the training population $\mathcal{P}_j$ by uniformly sampling m policies from each convention in $\mathcal{K}_j$, where m is the number of policies per convention. This ensures that each K_1 agent receives balanced exposure to the conventions in its capability set. Since we uniformly sample capability subsets but different subsets may contain different numbers of K_0 policies, individual agents may be non-uniformly represented across the full K_1 population.

Figure 6 visualizes the composition of generated K_1 subsets across multiple panels, illustrating the cross-play structure of the K_0 population, stratified sampling across conventions, capability variation among K_1 agents, and the distribution of K_0 policies across the full K_1 population.

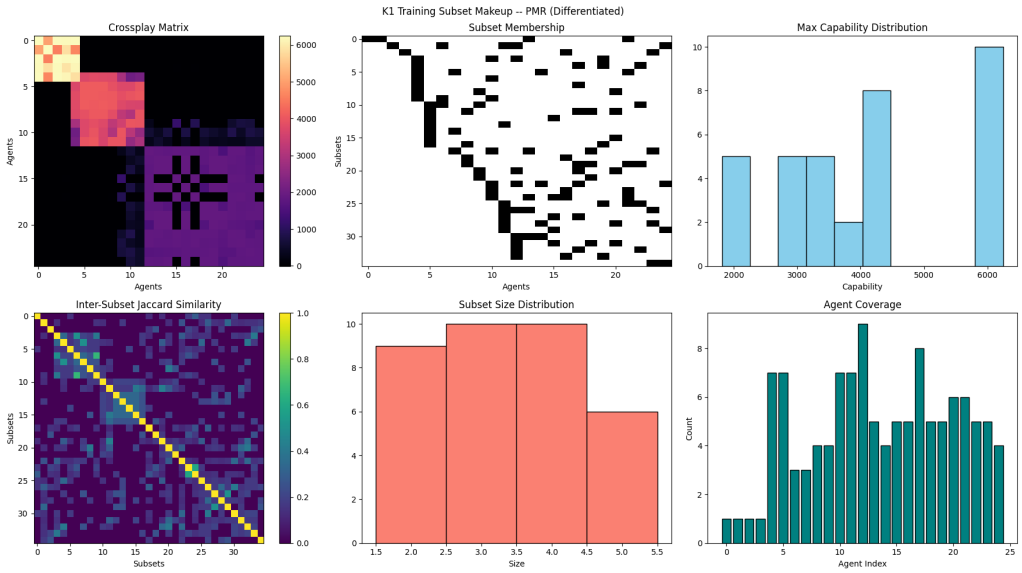


Figure 6: Makeup of the generated K_1 subsets. Top left: cross-play matrix of the K_0 population, clustered into convention sets. Top middle: sampling of individual policies from each convention to form training distributions for individual K_1 agents. Top right: distribution of maximum achievable returns across K_1 subsets, reflecting varying capability levels. Bottom left: Jaccard similarity between generated K_1 subsets. Bottom right: K_1 subset sizes and frequency with which each K_0 policy appears across all K_1 training distributions.

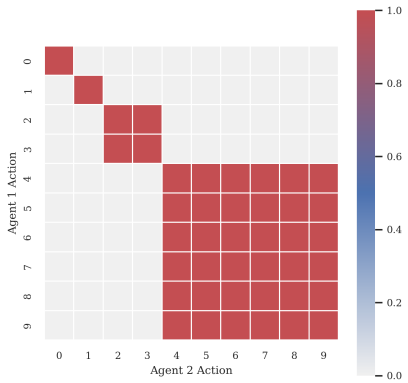
D Supplementary Experimental Results

D.1 Convention-Rebalanced Training

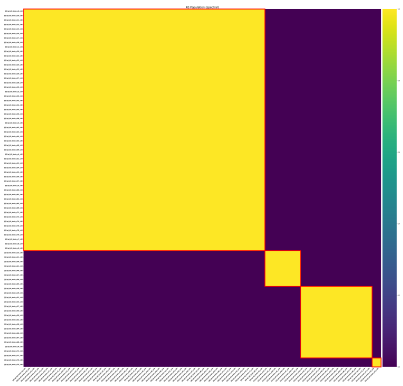
This section reports the population-rebalancing experiment introduced in Section 7, isolating the effect of rebalancing a training population across conventions independent of the full *ConventionPlay* hierarchy.

Setup We construct a variant of the matrix game whose payoff structure is designed to make some conventions harder to discover than others: every convention here yields the same reward, but differs in how many compatible actions it spans (1, 1, 2, and 6 actions respectively—shown in Figure 7a). This mirrors the hard-to-find CMG-H setup of Charakorn et al. (2023), in which smaller compatible-action sets are less likely to be discovered by self-play under random exploration. As expected, training a K_0 population under self-play skews heavily toward easy-to-find conventions (Figure 7b), with the hardest-to-find conventions represented by only a small number of policies. Because our ad-hoc setting assumes a test-time partner is equally likely to follow any convention, we construct a $\mathcal{P}^{(0)}_{\text{test}}$ population that is uniform over convention labels, using a hard-coded, single-fixed-action policy for each of the four conventions. We train best-response agents against the imbalanced population $\mathcal{P}$ as well as convention-balanced populations $\mathcal{P}'$ computed via matrix factorization and spectral clustering; for this experiment we additionally evaluate k-means clustering, applied directly to the rows of the cross-play matrix $\boldsymbol{X}$, as a simpler alternative to the two.

Results Figure 8 reports both the partner-sampling marginals and the resulting returns for all best-response agents. Naive best-response sampling reproduces the population imbalance seen in Figure 7b, while spectral-clustering and k-means rebalancing both track the uniform target closely,



(a) Payoff matrix for the Matrix Game variant: four equal-reward conventions with unequal numbers of compatible actions (1, 1, 2, and 6), following the payoff formulation of Charakorn et al. (2023).



(b) Cross-play matrix of a K_0 population trained via self-play on this variant, clustered into its four discovered conventions. The population is heavily skewed toward the largest, most discoverable convention.

Figure 7: Setup for the convention-rebalancing experiment: a matrix-game variant designed so that some conventions are harder to discover than others, and the resulting imbalance in a self-play K_0 population.

indicating accurate discovery of the underlying convention system. Downstream performance mirrors this ordering: evaluated against the $\mathcal{P}^{(0)}_{\text{test}}$ population, policies trained under spectral-clustering and k-means rebalancing achieve a significant 40% improvement in return over the naive baseline.

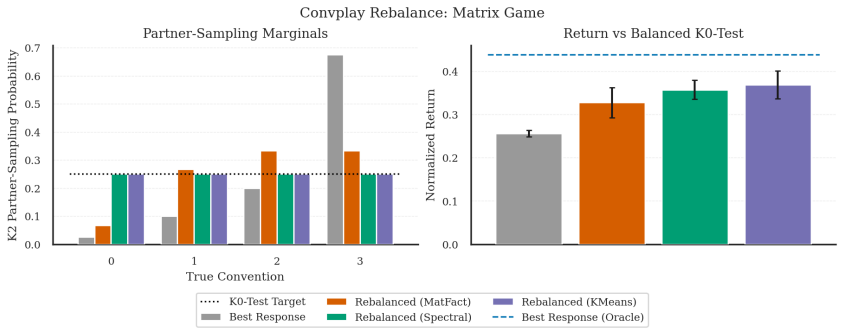


Figure 8: Left: K_2 partner-sampling marginal probability by true convention for the Matrix Game variant, comparing naive best-response sampling against matrix-factorization, spectral-clustering, and k-means rebalancing, relative to the uniform target (dotted line). Right: normalized episodic return for a best-response policy trained under each rebalancing scheme, evaluated against the uniform $\mathcal{P}^{(0)}_{\text{test}}$ population, with an oracle upper bound (dashed line) given by the balanced test population's own self-crossplay.

We test the sensitivity of each partitioning method to its expected number of conventions, q. Figure 9 reports normalized return under each method across a range of q values spanning the ground-truth count. All three methods improve over the unrebalanced baseline across the full range tested, with performance maximized at the correct q. All methods degrade when q is underspecified, but MatFact in particular remains resilient to overspecification. This sensitivity may vary across task domains, since the partitioning methods differ in how they handle non-transitive compatibility structure and noise in the cross-play matrix.

D.2 Normalized Return

Reward values are normalized to each environment's known range, so results can be compared across experimental domains. Under the partial and full differentiation variants, many capability-limited

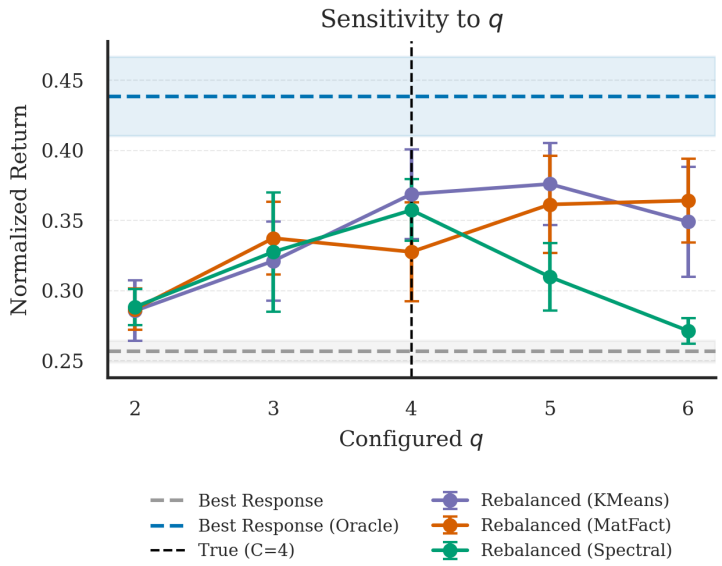


Figure 9: Normalized episodic return for a best-response policy trained under each rebalancing scheme (MatFact, Spectral, k-means), as a function of the specified number of conventions q, evaluated against the uniform $\mathcal{P}^{(0)}_{\text{test}}$ population for the Matrix Game variant of Appendix D.1. The dashed vertical line marks the ground-truth number of conventions. All methods improve over the unrebalanced baseline even when q is mis-specified, with the largest gains at the correct value; MatFact remains comparatively resilient to overspecification of q.

agents cannot score the environment maximum, so the optimal score is not 100%, but an average over each test agent's own achievable maximum. We compute this oracle empirically as the best response achievable against each test agent – something we can compute directly since we know each agent's capabilities – and plot the resulting average in Figure 3 as a reference.

Table 7: Mean normalised episodic return on $\mathcal{P}^{(1)}_{\text{heur}}$ partners. Each cell reports mean $\pm$ 1 s.d. (unbiased sample standard deviation, $n-1$ denominator) across $n = 5$ seeds. Bold marks the row's best value.

Environment	Var	FCP	SyKLRBR (L5)	SyKLRBR (L2)	CoMeDi (BC)	CoMeDi (BR)	LIPO	MEP	ConvPlay (matfact)	ConvPlay (spectral)
Matrix Game	Non-diff	93.04 ± 0.83	93.47 ± 0.43	**94.06 ± 0.50**	44.25 ± 1.88	87.25 ± 2.39	94.02 ± 0.38	81.26 ± 18.89	93.62 ± 0.49	93.01 ± 0.68
	Partial	27.07 ± 0.98	27.77 ± 1.07	27.75 ± 0.92	9.11 ± 0.32	13.29 ± 1.66	27.60 ± 0.54	22.85 ± 5.02	28.75 ± 4.37	**29.54 ± 0.60**
	Full	14.87 ± 0.58	15.38 ± 2.55	15.91 ± 0.96	8.69 ± 0.17	12.97 ± 1.12	14.36 ± 2.18	14.25 ± 1.00	**17.60 ± 2.15**	17.24 ± 1.46
PMR	Non-diff	62.91 ± 6.71	68.17 ± 1.21	68.30 ± 0.93	51.17 ± 3.28	63.83 ± 7.42	**68.52 ± 1.44**	47.63 ± 2.14	60.78 ± 4.33	64.94 ± 7.64
	Partial	32.49 ± 3.39	32.18 ± 1.03	32.73 ± 1.06	21.83 ± 1.91	28.78 ± 3.08	26.10 ± 1.30	30.84 ± 4.90	39.18 ± 2.49	**39.26 ± 1.33**
	Full	27.88 ± 4.42	27.52 ± 2.34	27.20 ± 2.77	7.96 ± 7.46	23.64 ± 2.72	25.84 ± 1.61	19.74 ± 3.37	34.42 ± 1.48	**36.64 ± 1.50**
Task Relay	Non-diff	**76.91 ± 9.41**	3.54 ± 5.71	1.70 ± 2.68	43.17 ± 13.17	12.10 ± 11.38	5.83 ± 7.65	13.51 ± 9.66	68.91 ± 26.33	70.79 ± 6.58
	Partial	16.21 ± 19.55	0.00 ± 0.00	1.13 ± 1.28	40.23 ± 9.31	39.47 ± 21.22	0.01 ± 0.02	23.28 ± 13.24	45.46 ± 31.67	**68.73 ± 1.91**
	Full	24.80 ± 20.81	10.06 ± 5.76	13.62 ± 4.50	23.41 ± 1.24	13.86 ± 5.19	0.44 ± 0.79	3.37 ± 2.91	50.93 ± 21.21	**58.60 ± 12.17**

D.3 Coordination Efficiency

To quantify how effectively an agent adapts to its partner's specific capability limitations, we introduce the *Coordination Efficiency* $\eta(\pi, \pi')$ for a policy π against a partner π', which we define as:

$$\eta(\pi, \pi') = \frac{J(\pi, \pi')}{J^*(\pi')} \tag{4}$$

where $J^*(\pi') = \max_{\hat{\pi} \in \Pi} J(\hat{\pi}, \pi')$ represents the maximum possible return achievable with partner π'. We approximate $J^*(\pi')$ based on the level k of the partner strategy π'. For a level $k = 0$ partner, $J^*_{K_0}(\pi^{(0)}_i) = J_{SP}(\pi^{(0)}_i)$. For a level $k = 1$ partner, we define J^* as the maximum joint return possible with any policy belonging to a convention within that partner's repertoire:

$$J^*_{K_1}(\pi^{(1)}_i) = \max_{C \in \mathcal{K}_i} \left[\max_{\hat{\pi} \in C} J(\pi^{(1)}_i, \hat{\pi}) \right] \tag{5}$$

This metric normalizes by each partner's achievable maximum return, providing a fair assessment of coordination efficiency even when conventions have substantially different expected returns.

Table 8: Coordination efficiency η (%) on $\mathcal{P}^{(0)}_{\text{test}}$ partners. Each cell reports mean $\pm$ 1 s.d. (unbiased sample standard deviation, $n-1$ denominator) across $n=5$ seeds. Bold marks the row's best value.

Environment	Var	FCP	SyKLRBR (L5)	SyKLRBR (L2)	CoMeDi (BC)	CoMeDi (BR)	LIPO	MEP	ConvPlay (matfact)	ConvPlay (spectral)
Matrix Game	Non-diff	95.16 ± 0.30	95.47 ± 0.17	95.55 ± 0.14	67.64 ± 0.21	68.86 ± 4.86	**95.58 ± 0.09**	69.44 ± 1.31	94.64 ± 0.15	94.42 ± 0.37
	Partial	95.35 ± 0.55	95.25 ± 0.35	**95.41 ± 0.40**	55.77 ± 0.24	53.83 ± 6.64	83.15 ± 0.30	68.33 ± 10.25	90.19 ± 5.04	92.12 ± 0.37
	Full	**95.26 ± 0.26**	92.53 ± 5.88	93.07 ± 5.35	44.80 ± 0.16	48.01 ± 1.97	64.11 ± 4.55	56.18 ± 0.47	87.36 ± 4.78	86.02 ± 5.34
PMR	Non-diff	90.05 ± 7.96	**98.20 ± 0.24**	98.11 ± 0.14	66.95 ± 3.24	80.55 ± 11.43	91.27 ± 6.79	52.50 ± 7.26	78.71 ± 7.25	91.70 ± 9.04
	Partial	90.14 ± 8.64	**98.20 ± 0.13**	98.16 ± 0.19	51.02 ± 7.01	73.20 ± 8.43	63.66 ± 0.32	83.68 ± 13.72	94.31 ± 7.10	97.53 ± 0.24
	Full	87.40 ± 13.70	88.43 ± 8.62	78.20 ± 10.79	30.13 ± 9.03	63.62 ± 14.91	70.07 ± 8.67	55.67 ± 9.16	90.22 ± 8.88	**96.69 ± 0.21**
Task Relay	Non-diff	**91.49 ± 0.90**	89.00 ± 3.68	88.49 ± 4.86	64.06 ± 4.67	38.93 ± 9.97	75.44 ± 5.45	80.13 ± 4.60	80.65 ± 3.57	84.48 ± 6.45
	Partial	**95.65 ± 7.99**	92.18 ± 3.17	94.28 ± 2.23	55.15 ± 11.91	72.27 ± 9.88	65.41 ± 1.47	79.46 ± 4.58	81.92 ± 4.61	89.24 ± 7.00
	Full	53.37 ± 13.16	60.69 ± 5.83	59.61 ± 7.35	27.14 ± 2.35	32.49 ± 9.79	29.78 ± 6.19	8.05 ± 1.94	42.80 ± 18.10	**62.90 ± 4.85**

Table 9: Coordination efficiency η (%) on $\mathcal{P}^{(1)}_{\text{sampled}}$ partners. Each cell reports mean $\pm$ 1 s.d. (unbiased sample standard deviation, $n-1$ denominator) across $n=5$ seeds. Bold marks the row's best value.

Environment	Var	FCP	SyKLRBR (L5)	SyKLRBR (L2)	CoMeDi (BC)	CoMeDi (BR)	LIPO	MEP	ConvPlay (matfact)	ConvPlay (spectral)
Matrix Game	Non-diff	66.88 ± 2.76	64.68 ± 2.10	64.54 ± 2.61	67.45 ± 1.62	71.73 ± 3.36	61.28 ± 5.21	73.10 ± 4.09	**87.47 ± 1.79**	86.81 ± 1.93
	Partial	46.69 ± 6.71	46.24 ± 3.65	46.04 ± 4.17	67.40 ± 0.44	59.71 ± 3.32	53.29 ± 1.11	63.24 ± 5.07	76.38 ± 1.84	**84.23 ± 1.82**
	Full	39.32 ± 1.84	42.34 ± 6.18	39.32 ± 6.25	59.74 ± 0.52	50.83 ± 1.96	51.35 ± 4.97	64.18 ± 2.11	73.99 ± 1.15	**81.11 ± 0.92**
PMR	Non-diff	93.71 ± 6.04	99.02 ± 0.39	**99.07 ± 0.54**	66.82 ± 4.12	83.90 ± 8.91	94.51 ± 5.83	65.86 ± 8.60	90.53 ± 4.72	98.03 ± 2.26
	Partial	73.24 ± 5.58	77.58 ± 1.22	78.43 ± 2.32	47.97 ± 5.34	71.37 ± 7.19	60.96 ± 1.51	70.84 ± 9.72	**92.03 ± 1.65**	90.17 ± 1.54
	Full	74.78 ± 7.65	73.55 ± 6.69	52.59 ± 5.85	31.71 ± 8.82	55.86 ± 11.29	67.04 ± 7.07	59.92 ± 3.88	88.94 ± 3.00	**91.27 ± 1.73**
Task Relay	Non-diff	85.43 ± 2.63	8.78 ± 8.82	6.08 ± 6.61	55.49 ± 10.09	2.47 ± 4.76	7.91 ± 9.35	18.91 ± 21.60	77.75 ± 22.09	**86.07 ± 4.80**
	Partial	47.00 ± 39.51	2.83 ± 0.99	3.23 ± 1.07	43.84 ± 6.61	58.08 ± 30.14	1.63 ± 1.41	24.55 ± 15.87	68.19 ± 8.59	**81.16 ± 11.96**
	Full	18.17 ± 29.50	4.31 ± 4.32	2.83 ± 2.35	10.78 ± 1.76	5.10 ± 3.91	0.76 ± 0.57	1.55 ± 0.55	40.14 ± 26.49	**67.35 ± 10.91**

Table 10: Coordination efficiency η (%) on $\mathcal{P}^{(1)}_{\text{heur}}$ partners. Each cell reports mean $\pm$ 1 s.d. (unbiased sample standard deviation, $n-1$ denominator) across $n=5$ seeds. Bold marks the row's best value.

Environment	Var	FCP	SyKLRBR (L5)	SyKLRBR (L2)	CoMeDi (BC)	CoMeDi (BR)	LIPO	MEP	ConvPlay (matfact)	ConvPlay (spectral)
Matrix Game	Non-diff	93.77 ± 0.83	94.22 ± 0.43	**94.80 ± 0.50**	44.62 ± 1.89	87.93 ± 2.41	94.76 ± 0.39	81.90 ± 19.03	94.36 ± 0.49	93.74 ± 0.68
	Partial	73.68 ± 1.64	74.50 ± 1.32	74.66 ± 1.25	10.26 ± 0.67	20.37 ± 4.23	74.51 ± 0.63	56.05 ± 15.59	68.33 ± 13.38	**75.19 ± 1.00**
	Full	49.06 ± 1.02	46.20 ± 9.34	48.91 ± 4.48	9.38 ± 0.31	26.12 ± 2.95	27.99 ± 3.60	32.67 ± 3.80	42.12 ± 8.41	**51.01 ± 3.26**
PMR	Non-diff	86.32 ± 9.15	93.47 ± 1.66	93.68 ± 1.30	70.08 ± 4.49	87.45 ± 10.39	**93.94 ± 1.96**	64.77 ± 2.91	83.32 ± 5.95	88.98 ± 10.65
	Partial	65.43 ± 7.36	65.73 ± 1.84	66.81 ± 1.76	36.71 ± 4.63	55.79 ± 6.80	52.83 ± 2.26	63.85 ± 8.86	75.41 ± 6.62	**75.83 ± 2.72**
	Full	62.86 ± 9.40	62.71 ± 4.36	61.84 ± 5.23	19.93 ± 18.25	53.51 ± 6.56	57.06 ± 3.58	44.10 ± 7.84	73.11 ± 3.92	**77.68 ± 3.22**
Task Relay	Non-diff	**81.55 ± 10.06**	3.73 ± 6.03	1.79 ± 2.82	45.73 ± 13.94	12.75 ± 11.99	6.23 ± 8.15	14.33 ± 10.32	73.03 ± 27.95	75.18 ± 6.93
	Partial	17.89 ± 21.18	0.00 ± 0.00	1.26 ± 1.47	43.66 ± 9.86	42.64 ± 23.63	0.01 ± 0.02	25.38 ± 15.08	50.41 ± 35.06	**76.05 ± 2.01**
	Full	28.44 ± 24.40	11.19 ± 6.64	15.47 ± 5.14	28.50 ± 1.55	16.69 ± 6.46	0.48 ± 0.82	3.91 ± 3.33	59.92 ± 25.07	**68.05 ± 14.54**

D.4 Analysis of Probing in PMR

Figure 10 shows *ConventionPlay* against both a fixed-convention and adaptive partners in PMR. As in the matrix game, the ad-hoc agent "probes" with movement away from the partner's implied landmark and toward a higher-value one, in an attempt to induce the partner to adapt toward that convention.

D.5 Effect of K_1 Population Size

A critical design parameter is the size of the K_1 training population. With a low number of K_1 agents, *ConventionPlay* should approximate naive best-response training. As population size increases, diversity of adaptive partners should enhance probing capability. Figure 11 reveals an inflection point in this relationship: performance gains are substantial up to a moderate K_1 size, after which additional diversity yields diminishing returns. The exact inflection point varies by environment and convention discovery method, suggesting complex interactions between convention decomposition approach and K_1 population size. A thorough analysis of these interactions remains an important direction for future work.

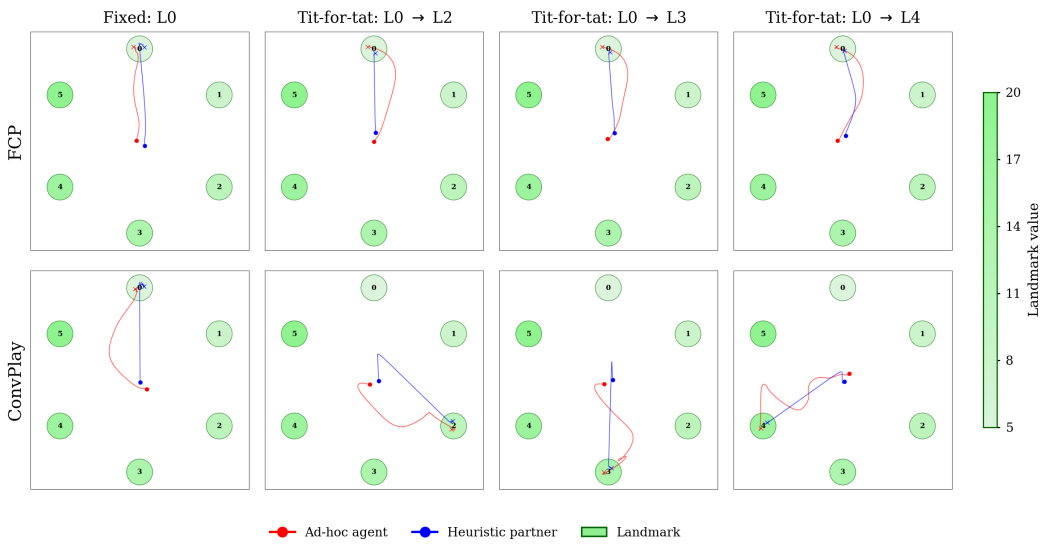


Figure 10: Trajectories in PMR for the ad-hoc agent (red) paired with a heuristic partner (blue), comparing FCP (top row) and *ConventionPlay* (bottom row). Left column: a fixed K_0 partner committed to landmark 0. Remaining columns: tit-for-tat partners that default to landmark 0 before switching to landmark 2, 3, or 4 respectively, conditioned on the position of the partner agent. FCP agents infer their partner's initially preferred convention and coordinate accordingly. *ConventionPlay*'s trajectories are visibly more winding, showing its ability to both lead a capability-limited adaptive partner or follow a fixed-convention one.

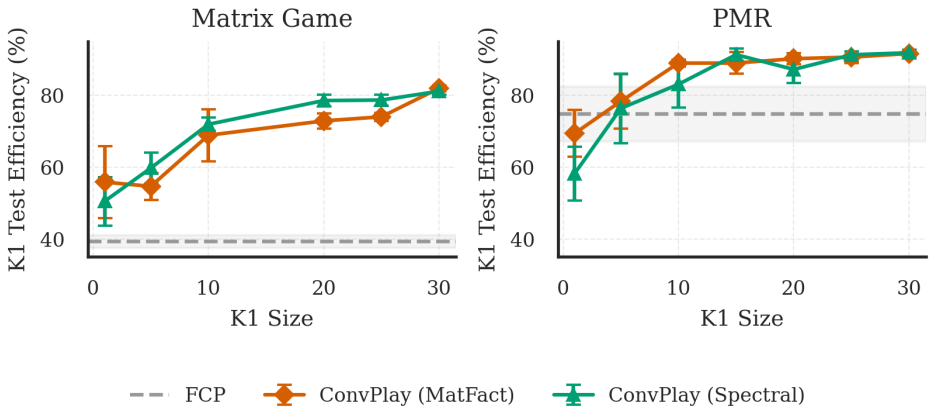


Figure 11: Effect of K_1 population size on coordination efficiency with $\mathcal{P}^{(1)}_{\text{sampled}}$ partners. Coordination efficiency $\eta \in [0, 1]$ is the fraction of a partner's achievable return obtained (defined in Appendix D.3). *ConventionPlay* exhibits substantial gains up to moderate diversity, followed by diminishing returns. FCP—invariant to K_1 size—is shown as a horizontal line for reference. Error bars show ± 1 sample standard deviation, where each seed's value is mean efficiency averaged over all test partners.