Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Intention modeling mediates cooperative incompatibility via character separability

Joonsong Lee
Ulsan National Institute of Science and Technology
Jooyeon Kim
jooyeon.kim@unist.ac.kr

Ulsan National Institute of Science and Technology

Article

Keywords: Multi-Agent Reinforcement Learning, Zero-Shot Coordination, Ad Hoc Teamwork, Cooperative AI, Agent Modeling

Posted Date: August 26th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10677371/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Intention modeling mediates cooperative incompatibility via character separability

Joonsong Lee[1] and Jooyeon Kim[1*]

[1*]Graduate School of Artificial Intelligence, Ulsan National Institute of Science and Technology (UNIST), South Korea.

*Corresponding author(s). E-mail(s): jooyeon.kim@unist.ac.kr;
Contributing authors: ljsong7@unist.ac.kr;

Abstract

Humans continually infer what others intend. This inference matters equally for AI, where it is studied as partner modeling. Partner modeling arises in two forms: an implicit form, in which an agent dynamically encodes partner behavior in its hidden state as a byproduct of return maximization, and an explicit form, in which a dedicated module learns a separate representation of partner intent. While explicit methods report gains, it remains unclear when, and why, they improve over the implicit representation approach. We address this question by establishing a novel connection to the character and mental-state decomposition proposed in the nascent work, machine theory of mind. With an explicit module, a division of labor arises: the dedicated intention module maintains a stable, separable *character* of a partner, while the agent's internal latent specializes in its *mental state*, which is continually updated. Under certain settings, the explicit approach reduces partner incompatibility, i.e., the failure to coordinate with certain partners. Specifically, we find that, in settings that require distinct best responses per partner type, the reduction of partner incompatibility is associated with separable character modeling, which the explicit module supplies. In other settings that we examined, explicit and implicit agents produce comparable return distributions.

Keywords: Multi-Agent Reinforcement Learning, Zero-Shot Coordination, Ad Hoc Teamwork, Cooperative AI, Agent Modeling

1 Introduction

In our daily lives, we routinely cooperate with partners who we never met. Doing so requires more than simply reacting to the environment. We continually observe how our partners behave and infer what they intend, especially in tasks that demand tight coordination. For example, we may infer a stranger's willingness to cooperate from their gesture toward an open seat or a driver's behavior when determining whether they intend to let us merge into traffic.

Intention inference is a fundamental capability not only for humans but also for artificial intelligence (AI) agents expected to cooperate with novel partners. The problem of cooperating with unknown partners is studied primarily under two closely related paradigms: zero-shot coordination (ZSC) [1] and ad hoc teamwork (AHT) [2], which share the goal of coordinating without prior joint training. Recent work approaches this challenge from several angles, including enhancing partner diversity [3–8], increasing environmental diversity [9], and intention modeling [10–13], also referred to as partner modeling.

Intention modeling is most often *explicit*: a dedicated module learns a separate representation of partner intent that conditions the policy. Yet it can also emerge *implicitly*: even without an objective for it, agents learn to represent partner behavior in their hidden states. Recently, Mon-Williams et al. [14] show that such agents encode partner information whenever the task requires it. However, the work does not ask whether explicit modeling adds anything beyond this implicit representation, leaving open whether making it explicit helps at all, and if so, under what conditions.

We approach this by asking what these representations capture. A natural lens comes from the work: machine theory of mind (machine ToM) [15], which decomposes an agent's model of another into two parts: a *character*, a stable long-term representation of who the partner is and how it tends to behave, and a *mental state*, a continually updated representation of what the partner is currently doing. However, how the representation that emerges without such a designed decomposition is organized remains underexplored. Moreover, the performance gain from explicit modeling itself is unclear, i.e., when and why it matters, given that population-based methods report strong coordination performance in the absence of such explicit module [4, 7, 16].

We investigate these questions in the Overcooked benchmark [17] across varying layouts and training-population sizes. We first train a cooperative ego agent using the standard ZSC recipe: population-based training with respect to a diverse partner pool, where the sampled partner remains fixed throughout each training episode. During evaluation, however, the partner agent interacting with the ego agent is dynamically swapped multiple times over the course of a single episodic rollout. This intra-episode partner substitution allows us to assess a form of interpolative generalization. While the ego agent has already encountered each individual partner during training, the temporal mixture of their changing behaviors constitutes a novel, zero-shot collaborative challenge. Under this setup, our findings are fourfold:

First, a division of labor arises in an explicit intention modeling agent, establishing the novel connection with the character and mental-state decomposition of machine ToM. The intention module maintains the partner *character*, while the agent's internal

latent, carrying no such type of supervision, specializes in the *mental state*. An implicit agent must serve both roles with a single representation and lies in between on each.

Second, explicit modeling reduces cooperative incompatibility [7], i.e., the failure to coordinate with partners in the population even when the total return rises, and the reduction of partner incompatibility is associated with the separability of the character representation. The relationship is one-sided: In settings that demand distinct best responses per partner type, no agent reaches high worst-case return without separable character modeling in our experiments. The converse does not always hold: separable agents can nevertheless fail.

Third, in the other settings that we examined, implicit modeling performs comparably to explicit modeling: the return distributions closely match, for partners both similar to and distinct from the training population. A marginal edge in prediction accuracy of the explicit modeling over the implicit counterpart does not directly translate to the coordination gain.

Finally, we repeat the analyses under a transformer [18] backbone. A stronger architecture also mitigates cooperative incompatibility, by reducing failures that occur despite separable character modeling, yet the dependence on separable character modeling and the division of labor persist, and explicit modeling still adds a margin.

Overall, our findings bridge the gap between ZSC or AHT and machine ToM by mapping how agents internally represent their partners and their corresponding, dynamically updated intentions. We demonstrate that while explicit modeling is not universally required, it enforces a vital division of labor that prevents cooperative incompatibility when environments demand best responses per partner types.

2 Preliminaries

2.1 Problem formulation

We formulate our problem as a common-payoff Markov game with full state observability—equivalently, a multi-agent Markov decision process [19]. The zero-shot coordination setting leaves the game unchanged but makes the partner unknown at training time. The game is a tuple $\mathcal{G} = \langle \mathcal{N}, \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \gamma \rangle$.

Environment and dynamics. Let $\mathcal{N} = \{1, \ldots, N\}$ denote the set of N agents; throughout the experiment, we fix $N = 2$. At each step t, the global state $s_t \in \mathcal{S}$ is shared and fully observable to both agents. The joint action space is $\mathcal{A} = \mathcal{A}^1 \times \mathcal{A}^2$, where $\mathcal{A}^i$ is the discrete action space of agent i. The environment transitions according to $\mathcal{T} : \mathcal{S} \times \mathcal{A} \to \Delta\mathcal{S}$, written $P(s_{t+1} \mid s_t, a_t^1, a_t^2)$. The agents share a reward function $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, written $r(s_t, a_t^1, a_t^2)$.

Policy and objective. Let $\tau_t = (s_0, a_0^1, a_0^2, \ldots, s_t)$ denote the history up to time t, observable to both agents under full observability, and let $\mathcal{H}$ be the set of histories. Each agent i acts according to a history-dependent policy $\pi^i : \mathcal{H} \to \Delta\mathcal{A}^i$. Although the state is Markov, we allow history dependence because, in the zero-shot setting, the partner is unknown and itself history-dependent. An agent must infer its partner from the interaction so far, which is what motivates a recurrent representation. We denote the partner of agent i as $-i$ and its policy as π^{-i}. The goal is to maximize the discounted return $R_t = \sum_{k=0}^{T-t} \gamma^k r_{t+k}$, with discount factor $\gamma \in [0, 1)$.

2.2 Population-based methods

In the cooperative setting, agents may converge to mutually incompatible optima—for instance, through arbitrary conventions arising from environmental symmetries [1]. Independently trained agents therefore often fail to coordinate with one another. A major line of work trains the cooperative agent against a diverse population of partners. Formally, the partners π^{-i} are drawn from a designated distribution $\mathcal{D}$, and agent i maximizes the expected return over the distribution as follows:

$$\pi^{i,*} = \arg\max_{\pi^i} \mathbb{E}_{\pi^{-i}\sim\mathcal{D}} \left[\mathbb{E}_{(a_t^1, a_t^2)\sim(\pi^i, \pi^{-i})} \left[\sum_{t=0}^{T} \gamma^t r(s_t, a_t^1, a_t^2) \right] \right]. \tag{1}$$

Note that the joint action is written with agent i in the first slot for notational convenience; the roles of the two slots are interchangeable, and the ego agent may occupy either position. We let M denote the population size, the number of partners it contains. Notice that during the training phase, the sampled partner agent $-i$ from $\mathcal{D}$ remains consistent throughout the episode.

2.3 Theory of mind (ToM)

Theory of mind (ToM) originates in psychology [20]: the capacity to attribute unobservable states such as beliefs, desires, and intentions to others. Machine theory of mind [15] is a machine-learning approach to ToM that enables an artificial agent to understand and predict the behavior of others by inferring their hidden mental states purely from behavioral trajectories. An observer network parses a target's past and current state–action trajectories and encodes them into two embeddings that condition its predictions: a character embedding, capturing stable, long-term traits inferred from the target's past episodes, and a mental-state embedding, capturing episode-specific state inferred from the current trajectory. Formally,

$$e_j^{\text{char}} = f_\theta([\tau_j]_{0:T}), \qquad e^{\text{char}} = \sum_{j=1}^{J} e_j^{\text{char}}, \qquad e_t^{\text{mental}} = g_\phi\big([\tau^{\text{cur}}]_{0:t-1},\ e^{\text{char}}\big),$$

where τ_j is the observed agent's state–action trajectory in the j-th past episode and τ^{cur} is that of the current episode—the observer is assumed to observe the full environment state and the target's actions; f_θ and g_ϕ are the character and mental-state networks, respectively, implemented as recurrent networks to process trajectories; J is the number of past episodes; and T is the episode length. From both embeddings and the current state, a network is trained to predict the target's next-action distribution, which objects it will consume, and its successor representation, the discounted expected future occupancy over states. Conceptually, the two embeddings mirror a distinction from human social cognition: the character embedding encapsulates the target's inherent, enduring traits, while the mental-state embedding captures its transient, situation-dependent states [21].

3 Methods

3.1 Environment

Our experiments are conducted on the Overcooked benchmark [17], implemented in JaxMARL [22]. Overcooked is a well-established benchmark for cooperative reinforcement learning (RL) with 6 discrete actions: four directional moves, interact, and stay (wait). Agents must place onions into a pot, cook the soup, plate it, and deliver it to a goal station to obtain a reward. The layouts force agents to cooperate—avoiding mutual blocking, sharing access to limited workspaces and intermediate storage cells (e.g., counters used for handing items off), and taking on complementary roles. We conduct experiments on five layouts: *Cramped Room, Coord. Ring, Asymmetric Coord., Counter Circuit, and Blocked Corridor*. Behavioral patterns and difficulty levels vary substantially across layouts. All layouts are illustrated in Supplementary Fig. S1. See the Supplementary Methods–"Environment details" for additional details.

3.2 Training methods

Our training pipeline has three components: a diverse partner population of size M, from which behavioral types are derived by clustering; a cooperative ego agent trained against this population, in an implicit and an explicit variant; and probing heads for analyzing what each agent's representation encodes. We describe each in turn. Architectural and training configurations for reproducibility are reported in Supplementary Tables S1 and S2 online, respectively.

3.2.1 Intention modeling

We use a simple latent intention representation e_t derived from predicting the partner's next action a_t^p and role c_t^p, where the superscript p denotes the partner and subscript t denotes current timestep. The design is inspired by the work of Wang et al. [10] and by the prediction targets of machine ToM: the role is intended as a coarse counterpart of its longer-horizon targets (consumption and the successor representation), summarizing the partner's long-run behavior. A dedicated convolutional neural network—long short-term memory (CNN-LSTM) [23–25] encodes the observation history into an intention representation e_t, its internal latent state. From e_t, we model both targets with a single shared multi-layer perceptron (MLP) rather than separate heads, with its final layer split into action and cluster logits, which keeps the module lightweight:

$$[\hat{a}_t^p;\, \hat{c}_t^p] = \mathrm{MLP}_{\mathrm{clf}}(e_t), \qquad e_t = \mathrm{LSTM}_{\mathrm{clf}}(\mathrm{CNN}_{\mathrm{clf}}(o_t),\, e_{t-1})\,.$$

We define a role as a behavioral cluster within the partner pool by adapting Ni et al. [12]. To form clusters, we compute pairwise similarity from cross-play returns:

$$s(\pi^A, \pi^B) = \min\left(\frac{R(\pi^A, \pi^B) + R(\pi^B, \pi^A)}{R(\pi^A, \pi^A) + R(\pi^B, \pi^B)},\, 1\right),$$

where $R(\pi^A, \pi^B)$ denotes the expected return of the pair; it can be asymmetric in layouts where the two starting positions are not interchangeable. We then apply distance-threshold agglomerative hierarchical clustering [26] on the distance $1 - s$. As a warm start, the intention module is trained alone for the first thousand iterations, during which rollouts are collected with the ego agent's policy held at its initial state and no policy update is applied. Regular joint training then follows, throughout which the classifier continues to be updated only by its prediction losses.

3.2.2 Training a diverse population

We adopt maximum-entropy population (MEP) [4], a widely used off-the-shelf algorithm for promoting population diversity. MEP is analogous to maximum-entropy RL [27, 28], but maximizes the entropy of the population's average policy rather than per-step action entropy. Each population member thus maximizes its self-play return, regularized by a population-diversity bonus. The training objective for agent i is:

$$J(\pi^i) = \mathbb{E}_{(a_t^1, a_t^2) \sim (\pi^i,\, \pi^i)} \left[\sum_t \gamma^t \left(r(s_t, a_t^1, a_t^2) + \alpha\, \mathcal{H}(\bar{\pi}(\cdot \mid s_t)) \right) \right], \tag{2}$$

where $\bar{\pi}(a_t \mid s_t) = \frac{1}{M} \sum_{i=1}^{M} \pi^i(a_t \mid s_t)$ denotes the population's average policy. The population is built on top of independent proximal policy optimization (I-PPO) [29, 30] with a CNN-LSTM backbone network.

3.2.3 Training the cooperative agent

We train two variants of cooperative ego agents, with or without explicit intention modeling, referred to as the *explicit agent* and the *implicit agent* in the following sections. Both share the same algorithm—I-PPO—and a CNN-LSTM history encoder, and differ only in what the policy conditions on. The encoder summarizes the observation history for ego agent into its recurrent latent state h_t, on which both policies condition; the explicit agent additionally conditions on the intention representation e_t:

$$h_t = \mathrm{LSTM}_\pi(\mathrm{CNN}_\pi(o_t),\, h_{t-1}), \qquad a_t^{\mathrm{imp}} \sim \pi^{\mathrm{imp}}(\cdot \mid h_t), \qquad a_t^{\mathrm{exp}} \sim \pi^{\mathrm{exp}}(\cdot \mid h_t,\, \mathrm{sg}(e_t)),$$

where sg denotes stop-gradient—implementing the detachment of RL gradients from the classifier—and e_t is the latent state of the intention module defined above. We train all ego agents using the fictitious co-play (FCP) [3] regime, in which they are paired with partners from the population above at three skill levels: *expert* (the converged policy), *mid* (a checkpoint achieving roughly half of the expert's return), and *initial* (an untrained checkpoint). The pool the ego agent trains against therefore contains $3M$ partners, three checkpoints for each of the M population members. Clusters for role prediction used in intention module are formed separately at each FCP level, and all initial-level agents collapse into a single cluster. We adopt prioritized partner sampling as in MEP [4], which prioritizes partners on which the ego agent attains low cooperative return. We apply it at the cluster level rather than the agent level, sampling clusters by their average priority. Cooperative agents thus maximize the objective

in Equation 1, with $\mathcal{D}$ the partner distribution induced by cluster-level prioritized sampling. The sampled partner remains fixed throughout each training episode. For the experiments in subsection 4.1, however, we swap the partner within an episode to assess a form of interpolative generalization, so that the ego agent should model the changing behavior from the interaction itself. Unless noted otherwise, we train 3 ego agents from different random seeds per variant, and use 10 seeds for the analyses that require closer inspection, namely those of subsection 4.1 and subsection 4.2 and the probing comparisons, as noted in each figure.

3.2.4 Training the probing heads

The explicit ego agent's intention representation e_t, by construction, models both the partner agent's following action a_t^p and its designated role c_t^p. However, its internal latent representation h_t does not model both. To investigate the functionality of the latent representation h_t of both the explicit and the implicit ego agents, we separately train probing heads, in a post-hoc fashion, that predict the same targets as the intention module, together with a regression target d_t, the number of steps remaining until the next delivery. The regression target examines whether the representations capture task-relevant quantities beyond partner identity, inspired by Richens et al. [31]. Each head is attached to the LSTM latent state of the agent—implicit and explicit alike—with the agent's parameters frozen:

$$\hat{a}_t^p = \mathrm{MLP}_a(\mathrm{sg}(h_t)), \qquad \hat{c}_t^p = \mathrm{MLP}_c(\mathrm{sg}(h_t)), \qquad \hat{d}_t = \mathrm{MLP}_d(\mathrm{sg}(h_t)).$$

All probing heads are trained on standard rollouts without partner switches, with training episodes sampled uniformly across behavioral clusters. The protocol is identical for the implicit and explicit variants. Since the explicit agent carries two classifiers, for the intention module and for the probing head, we use the terms *intention* and *probing* to distinguish them in the following sections.

3.3 Intention and machine ToM

The decomposition of machine ToM cannot be directly instantiated in our setting: in cooperative games where memory persists only within an episode, there are no past episodes from which to form the character embedding. We therefore recast the two components as roles within a single episode. The *character* refers to the stable component of the partner representation—who the partner is and how it tends to behave—hence separable across behaviorally distinct partners and constant within an episode. The *mental state* refers to the transient component—what the partner is currently doing—continuously inferred and updated within the episode. Both roles are crucial for a cooperative ego agent: it must commit to partner-specific responses and adapt to the partner's current behavior, corresponding to the character and the mental state, respectively. An implicit agent must encode both in a single latent, h_t. An explicit agent, in contrast, could realize this division across its two representations,

$$e_t^{\mathrm{exp}} \leftrightarrow e^{\mathrm{char}}, \qquad h_t^{\mathrm{exp}} \leftrightarrow e_t^{\mathrm{mental}}.$$

The first correspondence is partly by construction, since the intention module is supervised on the partner's type. Whether this correspondence in fact emerges, and how the implicit agent manages both roles within a single representation, is what we examine in subsection 4.1.

3.4 Statistical tools

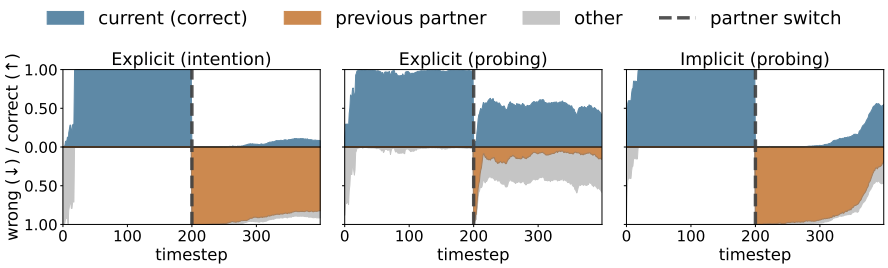
Unless noted otherwise, all aggregate results in the main text report the interquartile mean (IQM) over ego-agent seeds, with 95% bootstrap confidence intervals (CIs) from 2,000 resamples at the level of ego-agent seeds. When a result pools multiple layout–population combinations, the bootstrap is stratified by combination [32]; otherwise a plain bootstrap is used. We adopt this estimation-oriented protocol, rather than distributional tests, because the number of seeds per setting is small and the IQM is robust to outlier runs without discarding tail information. Group comparisons of silhouette scores (subsection 4.2) use a two-sided bootstrap test on the difference of group IQMs with the same resampling scheme. P values below the resolution of the bootstrap (1/2,000) are reported as $P < 5 \times 10^{-4}$.

4 Results

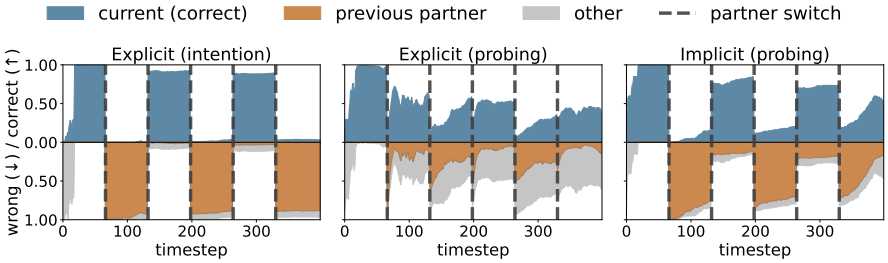
4.1 Explicit modeling divides the partner representation into the components of machine ToM

We examine how the correspondence hypothesized in subsection 3.3 emerges in the explicit agent, and how the implicit agent manages both roles within a single representation, using two protocols that change partner behavior within an episode. Both run on Coord Ring with a training population of four—a setting where fast adaptation is required and behavior is diverse, yet not so diverse that a single mixed policy covers all partners. We pick the most incompatible pair in the training population, the pair with the lowest cross-play return. We conduct either (i) a single switch at timestep 200, the midpoint of the 400-step episode, or (ii) five switches evenly spaced across the 400 steps. The agent receives no explicit signal of a switch. We train 10 ego agents from different random seeds for each of the implicit and explicit variants.

We find that a division of labor arises in the explicit agent, paralleling the character/mental-state decomposition of machine ToM. Figure 1 traces each module's cluster prediction through the switches, and Figure 2 shows the corresponding latent trajectories via uniform manifold approximation and projection (UMAP) [33] with kernel density estimation (KDE) [34]. The intention classifier of the explicit agent realizes the *character* role: it keeps its initial prediction essentially unchanged after the switch. The explicit agent's own latent specializes in the *mental state*: it is inferred from the partner's latest behavior. After a single switch it follows the changed partner or maps to regions belonging to *neither* cluster—a reasonable representation, since the trajectory mixes the behavior of multiple partners and matches no single one. The implicit agent must serve both roles with a single representation, and the two demands conflict: the character role requires the representation to stay unchanged within an



(a) Single partner switch at timestep 200



(b) Five partner switches per episode

Fig. 1: Cluster-prediction accuracy under behavior change, with respect to the current and previous partner. In each panel, the area above zero (blue) is the fraction of correct predictions of the current partner. The area below zero is the fraction of incorrect predictions, split into the previous partner (orange) and other clusters (gray). The intention classifier keeps its initial prediction. The probing head on the explicit agent's latent updates its prediction to the current partner or assigns the trajectory outside both clusters—the defining behaviors of *character* and *mental state*, respectively. The implicit agent lies in between: it stays committed to the initial cluster well past a switch, and its prediction shifts to the current partner only gradually.

episode, while mental-state tracking requires it to be continually updated. The compromise is visible in Figure 1: the implicit latent stays committed to the initial cluster well past the switch, and follows only gradually. Mean adaptation steps for switched partner under the single-switch protocol are reported in Supplementary Fig. S9.

Notably, the allocation between the two roles is not fixed but task-gated: the implicit agent shifts toward mental-state tracking under the *urgency signal*, an observation channel indicating that fewer than 40 timesteps remain. In Figure 1, its accuracy rises near the episode end, slightly exceeding that of the explicit agent's latent. Without urgency, the implicit latent leans toward character modeling. It shifts toward the mental state under urgency. See Supplementary Fig. S6 for the intervention analysis. We also find that the intention classifier is overconfident in its stale prediction, the explicit agent's latent predicts with low confidence, and the implicit agent lies in between. Prediction entropies are reported in Supplementary Fig. S7.

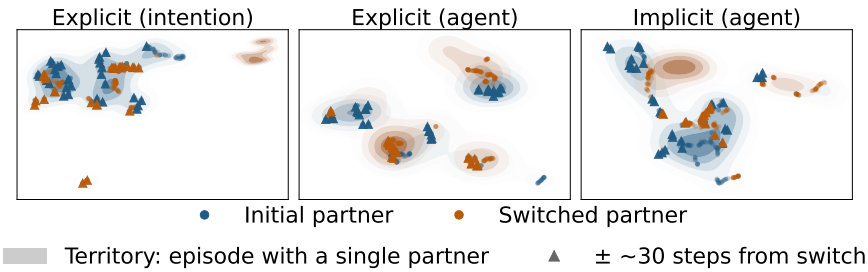


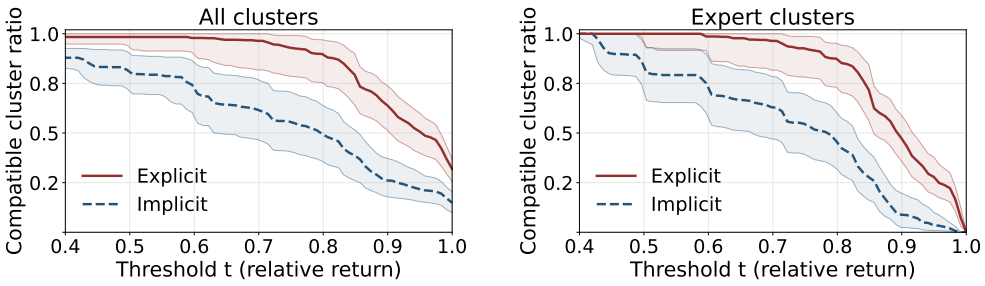
Fig. 2: Latent-space view of the single-switch protocol in Figure 1, showing how each representation follows the partner switch. Shaded territories are KDEs of latent trajectories from switch-free episodes with only the initial (blue) or only the switched (orange) partner, the reference regions where each partner's episodes lie in that space. Dots trace the switch episode, highlighted within ±30 steps of the switch. In the intention classifier's space, the trajectory stays in the initial partner's territory (*character*). In the explicit agent's own latent space, the trajectory swiftly moves into the switched partner's territory after the switch (*mental state*). In the implicit agent's space, the trajectory follows the change only gradually.

4.2 Separable character modeling mitigates cooperative incompatibility

Cooperative incompatibility is the phenomenon in which an ego agent fails to coordinate with some partners in the population even as its total return rises [7]. This section shows that explicit modeling reduces cooperative incompatibility, and that the reduction depends on the separability of the character representation, the character property defined in subsection 3.3.

We quantify incompatibility as follows. We measure coordination by relative return, the return of a pair as a fraction of the partner's self-play return. For a threshold t, a partner cluster is incompatible with an ego agent if any partner in the cluster attains a relative return below t. Incompatibility is thus a worst-case notion: a single partner below the threshold suffices. We report the compatible-cluster ratio—the fraction of partner clusters in which every partner clears the threshold—as a function of t. Explicit agents dominate implicit agents across nearly the entire range of t on the settings where incompatibility appears (Figure 3).

The settings shown are selected by a fixed failure criterion: we define a cooperation failure between an ego agent and a partner as an episode in which the pair delivers at most two dishes. A setting is included if either variant exhibits at least one such failure. Under this criterion, incompatibility appears in the two hardest layouts: Blocked Corridor at population sizes up to 32, and Counter Circuit—the layout with the most diverse behavioral clusters—at a population of 64. Blocked Corridor at 64, where no variant exhibits incompatibility, is included for completeness. Layout difficulty and population diversity are reported in Supplementary Figs. S2-S3. We train 10 ego agents from different random seeds per variant.



(a) All cluster levels

(b) Expert clusters only

Fig. 3: Cooperative incompatibility comparison between explicit and implicit agents, on the settings where incompatibility appears: Blocked Corridor, and Counter Circuit at a population of 64. The y-axis is the compatible-cluster ratio as a function of the threshold t on the x-axis. A partner cluster counts as compatible if every partner in it attains a relative return above t. IQM with 95% stratified bootstrap CIs over ego agents, stratified by layout–population combination. In panel (a), the self-play floor for initial-level clusters is set to 40. Explicit intention modeling reduces cooperative incompatibility.

Mitigating incompatibility requires the ego agent to commit to distinct best responses for distinct behavioral types, which in turn requires encoding the types separably—the character property. We therefore compare the separability of partner-cluster representations between failed and successful cases by silhouette score [35] (Figure 4). The intention classifier attains the highest silhouette score, and implicit agents that avoid failure form clearly more separable representations than those that fail. Figure 5 examines the relationship at the level of individual agents. Character separability and worst-case relative return are positively correlated. Moreover, the relationship is one-sided in these settings: no agent reaches high worst-case return without separable character modeling, whereas the converse failure—a separable representation with poor worst-case return—does occur. Separable character modeling is thus not sufficient on its own, but we find no case of coordination succeeding without it. Figure 6 visualizes the same relationship in latent space.

The exception, Blocked Corridor at a population of 32, is itself consistent with this account. There, two response policies suffice to cover all expert clusters, so the setting exerts no pressure to separate characters: agents achieve high worst-case return with low silhouette scores, precisely where cluster-specific best responses are not required (analysis in Supplementary Figs. S4-S5.). The one-sided relationship therefore appears where distinct types demand distinct best responses. This suggests a policy-level reading of the relationship: highly separable failure cases correspond to agents that develop cluster-specific responses yet fail to make them optimal, whereas weakly separable successful cases correspond to agents that cover the population with a common response. When such a common response fails against some partners, cooperative incompatibility appears. Notably, the case study shows a setting where a few response policies

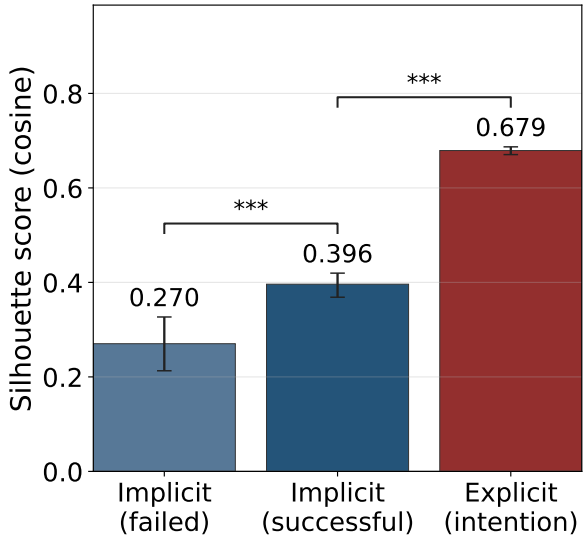


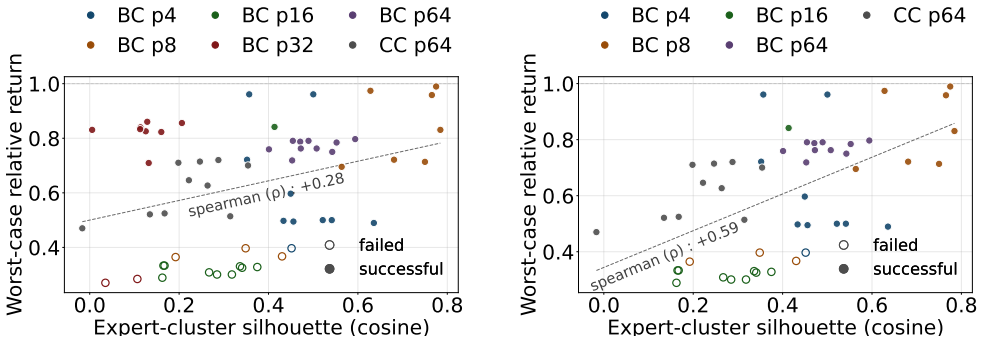
Fig. 4: Separability of character modeling for implicit agents, split into failed and successful cases, and the explicit agent's intention classifier. The explicit agent's intention classifier is not split into groups, since its supervision on partner type separates the representation by construction, regardless of coordination outcome. Cosine silhouette scores of partner-cluster representations, measured on expert clusters. Implicit agents are split by whether they exhibit a cooperation failure with at least one expert-cluster partner. IQM with 95% stratified bootstrap CIs over agents, stratified by layout–population combination, with two-sided bootstrap test brackets. Successful implicit agents form more separable representations than failure cases (two-sided bootstrap test, $P < 5 \times 10^{-4}$; $n = 15$ and 45 agents).

suffice to cover the entire population. Whether responses can be specialized further beyond such a point remains open, and we leave it for future work.

A natural follow-up question is whether the separate conditioning channel is necessary at all. An intention objective attached directly to the agent's own encoder—as an auxiliary task with gradients flowing into the CNN-LSTM—also shapes character separability by construction. We find that this auxiliary variant tightens, though does not fully close, the gap to the explicit agent, suggesting that much of what matters is achieving separable character modeling rather than the conditioning channel itself. Moreover, the division of labor of subsection 4.1 does not emerge without the separate module (Supplementary Figs. S10-S13.).

4.3 Implicit modeling matches the explicit counterpart in most coordination settings

In the other settings that we examined, except for those analyzed in subsection 4.2, explicit modeling does not systematically change cooperation performance, regardless of partner novelty (Figure 7). To probe partner novelty, we merge the training and test



(a) All settings (b) Settings requiring specific best responses

Fig. 5: Character separability versus worst-case coordination for implicit agents. Each point is one ego agent: the x-axis is the mean silhouette of its partner-cluster representation on expert clusters, computed over bootstrap resamples of within-cluster mean latents to balance cluster sizes. The y-axis is its worst-case relative return over expert partners. BC and CC denote the Blocked Corridor and Counter Circuit layouts, respectively, and the number following p is the training-population size. Panel (b) excludes Blocked Corridor at a population of 32, where two policy patterns cover all clusters. No agent reaches high worst-case return without separable character modeling, while the converse does occur. Separable character modeling is thus empirically necessary, but not sufficient, in these settings.

populations, re-run the clustering of subsubsection 3.2.3, and split the *test* partners into a *merged* group, assigned to a cluster that also contains training partners, and a *novel* group, assigned to a cluster with no training partner. Explicit and implicit agents perform comparably for both groups. An equivalence analysis supports this in the merged group, while novel-group differences exceed the margin more often but in both directions with no consistent sign (Supplementary Figs. S16-S17). Each ego agent is evaluated against a held-out pool of 64 test partners per layout, with performance measured as the return relative to each partner's self-play return; the construction of the test pool is detailed in Supplementary Methods–"Building test population". The performance gaps on Blocked Corridor and on Counter Circuit at a population of 64 are the cooperative incompatibility analyzed in subsection 4.2.

We further examine what the implicit latent encodes, using the probing modules described in subsubsection 3.2.4. The probing classifier on the implicit latent nearly matches the intention classifier of explicit agents on both cluster and action prediction, with a slight edge to the explicit objective. As reported by Mon-Williams et al. [14], agents indeed encode partner information whenever the task requires it. An additional probing regression head—predicting the number of steps remaining until the next delivery—performs equally well for both agent types, showing that implicit agents encode task-relevant information as well as explicit agents do, even while devoting capacity to modeling their partner (Figure 8). These trends largely persist across the remaining layout–population combinations. Implicit accuracy drops only in

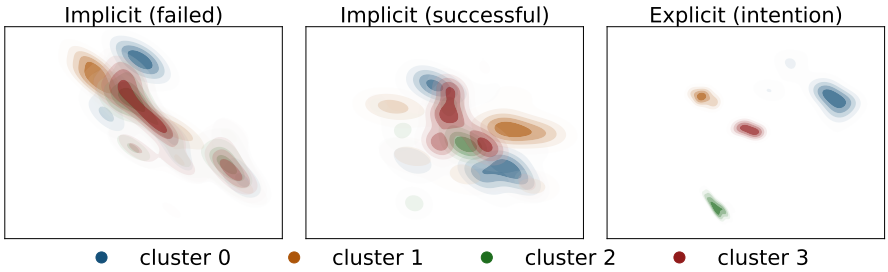


Fig. 6: Latent representation of character separability for implicit agents, split into failed and successful cases, and the explicit agent's intention classifier. KDE of latent trajectories per behavioral cluster over a 2-D UMAP projection on Blocked Corridor at a population of 8. The intention classifier of the explicit agent is the most separable. Implicit agents that avoid failure construct separable latents, whereas the latent spaces of failed agents collapse.

layouts where partner information is less required, consistent with the encode-when-needed account. Moreover, explicit modeling yields only marginal gains under partial observability. Detailed results are reported in Supplementary Figs. S19-S20.

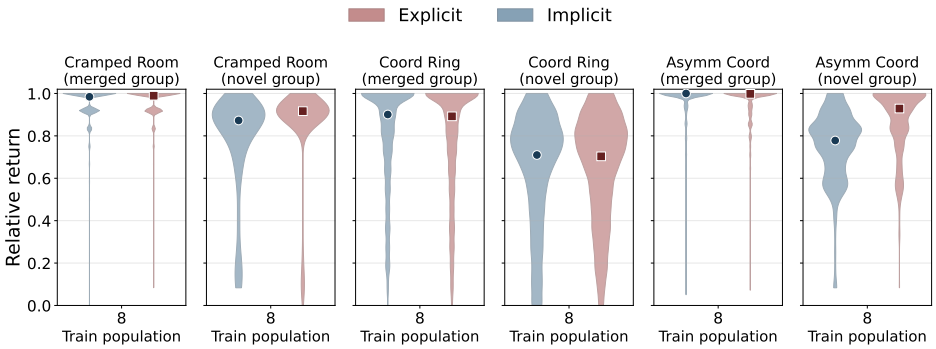
Read together with subsection 4.2, these results sharpen the picture. The partner-type information is present in the implicit latent even where its geometry is weakly separable. What limits worst-case coordination is thus not whether the partner type is encoded, but whether it is encoded separably enough for the policy to commit to distinct responses. In sum, most coordination settings exert no pressure toward explicit intention modeling: its systematic performance benefit is confined to the incompatibility-prone settings of subsection 4.2.

4.4 Machine ToM still matters with a stronger backbone

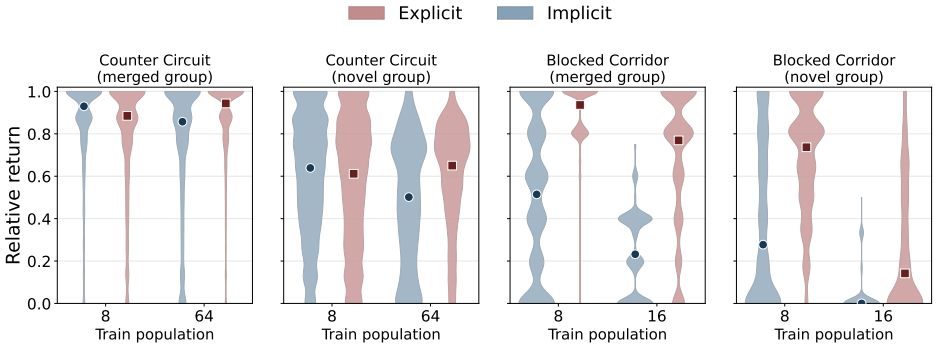
The findings of subsection 4.1 and subsection 4.2 raise the question of whether they depend on the recurrent architecture or hold more generally. We test this by replacing the backbone with a transformer [18]—specifically a decoder-only model [36]—and repeating the machine ToM and cooperative incompatibility analyses.

The division of labor persists, with a shift in who tracks what. The intention classifier still maintains the character, keeping its initial prediction through unseen switches. Mental-state tracking, however, redistributes: the transformer-explicit agent follows a switch more slowly than its LSTM counterpart, while the transformer-implicit agent tracks faster than the LSTM-implicit agent—though still behind the explicit one.

On cooperative incompatibility, the transformer backbone narrows the explicit–implicit gap: the transformer-implicit agent nearly reaches the LSTM-explicit agent, while the transformer-explicit agent remains the most robust overall. A stronger architecture thus mitigates cooperative incompatibility on its own, and explicit modeling adds a further margin on top. The relationship between character modeling and incompatibility is unchanged: the correlation persists, and we again observe no agent reaches



(a) Layouts where explicit and implicit modeling perform comparably



(b) Layouts where they diverge (Counter Circuit, Blocked Corridor)

Fig. 7: General cooperation performance of explicit and implicit agents. Each plot shows the distribution of per-episode relative return, over 160 episodes per test partner for each of three ego seeds per variant; markers denote the IQM. The *merged* group shares behavioral patterns with the training population, whereas the *novel* group does not. Explicit modeling does not systematically change performance in most settings (a). In the novel group, the two variants sometimes differ in their return distributions, most visibly on Asymmetric Coordination. Which variant leads, however, alternates across settings, with no consistent advantage for either. In (b), implicit and explicit agents remain comparable on Counter Circuit at a population of 8 and diverge on Blocked Corridor and on Counter Circuit at a population of 64—the cooperative incompatibility analyzed in subsection 4.2.

high worst-case return without separable character modeling (Figure 9). The backbone's contribution, however, is not to make character representations more separable; rather, it reduces the failures that occur despite separable character modeling—the insufficiency cases noted in subsection 4.2. The one-sided relationship is thus backbone-general. What a stronger backbone improves is how reliably a separable character

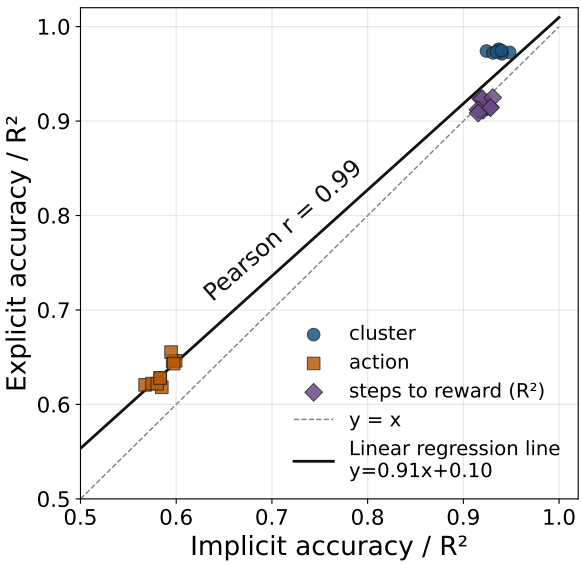


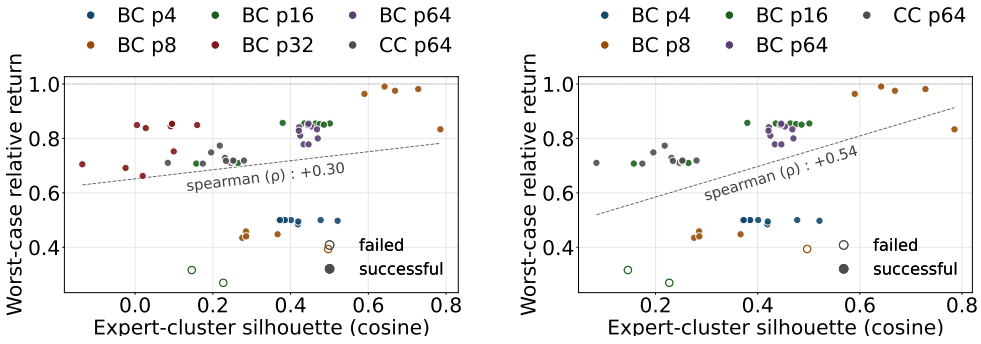
Fig. 8: Partner-prediction comparison between the implicit probing classifier and the intention classifier of explicit agents (Counter Circuit, population of eight). *Cluster* and *Action* report prediction accuracy. R^2 score reports how well an identical probing head predicts the number of steps remaining until the next delivery, for both agent types. Implicit modeling recovers information comparably to the explicit objective, with a slight edge to the explicit one.

translates into successful coordination. See Supplementary Figs. S14-S15 and the corresponding Supplementary Results section.

5 Related work

The central topic of this paper is cooperative multi-agent reinforcement learning, studied under two closely related settings: AHT [2] and ZSC [1]. Both focus on adapting to unseen partners that were not encountered during training. Prior work has primarily explored two fronts: constructing a diverse partner population, and explicitly modeling the partner's intent—in contrast to the implicit representation that this paper compares against.

Population-based methods build a diverse partner population in various ways: from independently trained self-play agents and their intermediate checkpoints spanning skill levels (FCP) [3], through a cognitive hierarchy [37], with explicit diversity objectives [4, 6], or biased toward human preferences modeled as hidden, event-based reward functions [5]. Li et al. [7] propose an open-ended training framework to address cooperative incompatibility. Incompatible or adversarial partners can be generated by maximizing self-play return while minimizing cross-play return [8, 16, 38], and generative models offer another route, sampling novel partners from a learned latent strategy space [16, 39].



(a) All settings (b) Settings requiring specific best responses

Fig. 9: Counterpart of Figure 5 with a transformer backbone. The positive relationship between character separability and worst-case return persists; the transformer's gain appears as fewer failures at high silhouette rather than as more separable representations. The one-sided character–incompatibility relationship is backbone-general.

Explicit intention modeling equips an agent with conditional information about its partner's intent, and the modeled quantity takes many forms, from pre-defined roles [40] and auxiliary prediction of other agents' actions [41] to learned latent representations, the form adopted in this paper; Albrecht & Stone [42] survey the broader space. Prior work represents a partner's strategy class or type in several ways: theory of mind reasoning [12, 15], as a per-type action bias [13], as a Bayesian belief factorized into permanent and temporal latents [43], or as a latent context tracked under within-episode policy switches [44]. Other approaches predict richer quantities, including teammate-aware returns-to-go and sub-goals that condition the ego policy [11], other agents' attention weights over goals inferred from their behavior [45], and the teammates' observations and actions reconstructed from the ego agent's local trajectory [10, 46]. The type-inference component is long studied in ad hoc teamwork, where the agent identifies the teammate's type and reuses a matched policy [47]; see Mirsky et al. [48] for a survey. Our work predicts the partner's next action together with its cluster-level role, and our aim is diagnostic—characterizing what such conditioning induces and when it matters—rather than proposing a new prediction target.

6 Discussion & Limitations

In this work, we showed that a division of labor arises with intention modeling in cooperative tasks, paralleling the character/mental-state decomposition of machine ToM. With an explicit module, the intention classifier maintains a stable, separable character while the agent's own latent, free of the type supervision, specializes in a continually updated mental state. An implicit agent must serve both roles with a single representation. We found that the mitigation of cooperative incompatibility depends on separable character modeling, the channel that the explicit agent's intention classifier supplies. The relationship is one-sided: in the settings that demand cluster-specific

best responses, no agent reaches high worst-case return without separable character modeling, while separable agents can still fail. In the other settings that we examined, explicit modeling does not change coordination performance: the implicit latent already encodes partner and task information comparably to the explicit objective. Finally, we examined the role of the backbone. A stronger sequence model narrows the explicit–implicit gap—not by making character representations more separable, but by reducing the failures that occur despite separable character modeling. The one-sided relationship itself persists, and an explicit module still adds a margin on top.

We note three limitations that suggest directions for future research. First, while we showed that the explicit agent's latent specializes in mental-state tracking, its functional role remains unexamined: this work focused on the relationship between cooperative incompatibility and character modeling. Investigating when and how mental-state tracking contributes to coordination is a valuable future direction. Second, we did not examine settings in which intention is not well defined, such as ensemble partner policies [49] or random-mixed self-play [50, 51]. Our analysis ties the decision making of population-based agents to a machine ToM structure. What structure arises in such settings is an interesting open question. Lastly, our experiments are confined to two-player games on the Overcooked benchmark [17]; extending the analysis to more than two players, where multiple partners must be modeled at once, is left to future work. Its layouts—which the JaxMARL implementation [22] makes easy to construct and vary—span diverse coordination structures and difficulty levels within purely cooperative settings; whether our findings extend to mixed-motive settings such as Melting Pot [52], where cooperation and competition coexist, remains to be validated and would be a compelling direction for studying social intelligence.

Data availability

All data generated or analyzed during this study are included in submitted files.

Code availability

The source code, configuration files, and scripts required to reproduce the experiments are provided with this manuscript as submitted materials.

References

[1] Hu, H., Lerer, A., Peysakhovich, A. & Foerster, J. *"Other-play" for zero-shot coordination. International Conference on Machine Learning (ICML)* (2020). URL http://proceedings.mlr.press/v119/hu20a/hu20a.pdf.

[2] Stone, P., Kaminka, G., Kraus, S. & Rosenschein, J. *Ad hoc autonomous agent teams: Collaboration without pre-coordination. The Association for the Advancement of Artificial Intelligence (AAAI)* (2010). URL https://ojs.aaai.org/index.php/TheAssociationfortheAdvancementofArtificialIntelligence(AAAI)/article/view/7529.

[3] Strouse, D., McKee, K., Botvinick, M., Hughes, E. & Everett, R. *Collaborating with humans without human data. Advances in Neural Information Processing Systems (NeurIPS)* (2021). URL https://openreview.net/pdf?id=79zWncwO2p.

[4] Zhao, R. *et al. Maximum entropy population-based training for zero-shot human-AI coordination. The Association for the Advancement of Artificial Intelligence (AAAI)* (2023). URL https://openreview.net/forum?id=v-f7ifhKYps.

[5] Yu, C. *et al. Learning zero-shot cooperation with humans, assuming humans are biased. International Conference on Learning Representations (ICLR)* (2023). URL https://openreview.net/forum?id=TrwE8l9aJzs.

[6] Lupu, A., Cui, B., Hu, H. & Foerster, J. *Trajectory diversity for zero-shot coordination. International Conference on Machine Learning (ICML)* (2021). URL https://proceedings.mlr.press/v139/lupu21a.html.

[7] Li, Y. *et al. Cooperative open-ended learning framework for zero-shot coordination. International Conference on Machine Learning (ICML)* (2023). URL https://proceedings.mlr.press/v202/li23au/li23au.pdf.

[8] Sarkar, B., Shih, A. & Sadigh, D. *Diverse conventions for human-AI collaboration. Advances in Neural Information Processing Systems (NeurIPS)* (2023). URL https://openreview.net/forum?id=MljeRycu9s.

[9] Jha, K. *et al. Cross-environment cooperation enables zero-shot multi-agent coordination. International Conference on Machine Learning (ICML)* (2025). URL https://openreview.net/forum?id=zBBYsVGKuB.

[10] Wang, C., Rahman, A., Durugkar, I., Liebman, E. & Stone, P. *N-agent ad hoc teamwork. Advances in Neural Information Processing Systems (NeurIPS)* (2024). URL https://proceedings.neurips.cc/paper_files/paper/2024/file/cabf611498431ad89a85ace75f790d93-Paper-Conference.pdf.

[11] Zhang, X., Chan, H., Ye, D., Cai, Y. & Zhao, M. *Ad hoc teamwork via offline goal-based decision transformers. International Conference on Machine Learning (ICML)* (2025). URL https://proceedings.mlr.press/v267/zhang25h.html.

[12] Ni, A. *et al. Theory of mind guided strategy adaptation for zero-shot coordination. International Conference on Autonomous Agents and Multiagent Systems (AAMAS)* (2026). URL https://doi.org/10.65109/UMCV8852.

[13] Li, B. *et al. Adaptively coordinating with novel partners via learned latent strategies. Advances in Neural Information Processing Systems (NeurIPS)* (2025). URL https://proceedings.neurips.cc/paper_files/paper/2025/file/020973f2093a6053261da93cac30ab71-Paper-Conference.pdf.

[14] Mon-Williams, R. *et al. Partner modelling emerges in recurrent agents (but only when it matters). Advances in Neural Information Processing Systems (NeurIPS)* (2026). URL https://openreview.net/pdf?id=WydxWM2xNb.

[15] Rabinowitz, N. *et al. Machine theory of mind. International Conference on Machine Learning (ICML)* (2018). URL https://proceedings.mlr.press/v80/rabinowitz18a.html.

[16] Chaudhary, P., Liang, Y., Chen, D., Du, S. & Jaques, N. *Improving human-AI coordination through online adversarial training and generative models. International Conference on Learning Representations (ICLR)* (2026). URL https://openreview.net/forum?id=AeehNfbHqD.

[17] Carroll, M. *et al. On the utility of learning about humans for human-AI coordination. Advances in Neural Information Processing Systems (NeurIPS)* (2019). URL https://proceedings.neurips.cc/paper/2019/file/f5b1b89d98b7286673128a5fb112cb9a-Paper.pdf.

[18] Vaswani, A. *et al. Attention is all you need. Advances in Neural Information Processing Systems (NeurIPS)* (2017). URL https://dl.acm.org/doi/10.5555/3295222.3295349.

[19] Boutilier, C. *Planning, learning and coordination in multiagent decision processes. Conference on Theoretical Aspects of Rationality and Knowledge (TARK)* (1996). URL https://dl.acm.org/doi/abs/10.5555/1029693.1029710.

[20] Premack, D. & Woodruff, G. Does the chimpanzee have a theory of mind? *Behavioral and brain sciences* **1**, 515–526 (1978). URL https://doi.org/10.1017/S0140525X00076512.

[21] Tamir, D. & Thornton, M. Modeling the predictive social mind. *Trends in Cognitive Sciences* **22**, 201–212 (2018). URL https://www.sciencedirect.com/science/article/pii/S136466131730267X.

[22] Rutherford, A. *et al. JaxMARL: Multi-agent RL environments and algorithms in jax. Advances in Neural Information Processing Systems (NeurIPS)* (2024). URL https://openreview.net/forum?id=X90tyXDe8z.

[23] SHI, X. *et al. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. Advances in Neural Information Processing Systems (NeurIPS)* (2015). URL https://proceedings.neurips.cc/paper_files/paper/2015/file/07563a3fe3bbe7e3ba84431ad9d055af-Paper.pdf.

[24] LeCun, Y. & Bengio, Y. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks* **3361**, 1995 (1995).

[25] Hochreiter, S. & Schmidhuber, J. Long short-term memory. *Neural computation* **9**, 1735–1780 (1997).

[26] Defays, D. An efficient algorithm for a complete link method. *The Computer Journal* **20**, 364–366 (1977). URL https://doi.org/10.1093/comjnl/20.4.364.

[27] Ziebart, B., Maas, A., Bagnell, A. & Dey, A. *Maximum entropy inverse reinforcement learning. The Association for the Advancement of Artificial Intelligence (AAAI)* (2008). URL https://dl.acm.org/doi/abs/10.5555/1620270.1620297.

[28] Haarnoja, T., Zhou, A., Abbeel, P. & Levine, S. *Soft Actor-Critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. International Conference on Machine Learning (ICML)* (2018). URL https://proceedings.mlr.press/v80/haarnoja18b.html.

[29] Schulman, J., Wolski, F., Dhariwal, P., Radford, A. & Klimov, O. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347* (2017). URL https://arxiv.org/abs/1707.06347.

[30] Witt, C. *et al.* Is independent learning all you need in the starcraft multi-agent challenge? *arXiv preprint arXiv:2011.09533* (2020). URL https://arxiv.org/abs/2011.09533.

[31] Richens, J., Everitt, T. & Abel, D. *General agents need world models. International Conference on Machine Learning (ICML)* (2025). URL https://openreview.net/forum?id=dlIoumNiXt.

[32] Agarwal, R., Schwarzer, M., Castro, P. S., Courville, A. & Bellemare, M. *Deep reinforcement learning at the edge of the statistical precipice. Advances in Neural Information Processing Systems (NeurIPS)* (2021). URL https://proceedings.neurips.cc/paper_files/paper/2021/file/f514cec81cb148559cf475e7426eed5e-Paper.pdf.

[33] McInnes, L., Healy, J., Saul, N. & Großberger, L. UMAP: Uniform manifold approximation and projection. *Journal of Open Source Software* **3**, 861 (2018). URL https://doi.org/10.21105/joss.00861.

[34] Parzen, E. On estimation of a probability density function and mode. *The Annals of Mathematical Statistics* **33**, 1065–1076 (1962). URL http://www.jstor.org/stable/2237880.

[35] Rousseeuw, P. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics* **20**, 53–65 (1987). URL https://www.sciencedirect.com/science/article/pii/0377042787901257.

[36] Brown, T. *et al. Language models are few-shot learners. Advances in Neural Information Processing Systems (NeurIPS)* (2020). URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

[37] Cui, B., Hu, H., Pineda, L. & Foerster, J. K-level reasoning for zero-shot coordination in Hanabi. *Advances in Neural Information Processing Systems (NeurIPS)* (2021). URL https://proceedings.neurips.cc/paper/2021/file/4547dff5fd7604f18c8ee32cf3da41d7-Paper.pdf.

[38] Charakorn, R., Manoonpong, P. & Dilokthanakul, N. *Generating diverse cooperative agents by learning incompatible policies. International Conference on Learning Representations (ICLR)* (2023). URL https://openreview.net/forum?id=UkU05GOH7_6.

[39] Liang, Y., Chen, D., Gupta, A., Du, S. & Jaques, N. *Learning to cooperate with humans using generative agents. Advances in Neural Information Processing Systems (NeurIPS)* (2024). URL https://proceedings.neurips.cc/paper_files/paper/2024/file/6e9f99809b6df59e8fe5c9ffe269bf40-Paper-Conference.pdf.

[40] Genter, K., Agmon, N. & Stone, P. *Role-based ad hoc teamwork. The Association for the Advancement of Artificial Intelligence (AAAI)* (2011). URL https://ojs.aaai.org/index.php/AAAI/article/view/8057.

[41] Hernandez-Leal, P., Kartal, B. & Taylor, M. *Agent modeling as auxiliary task for deep reinforcement learning. The Association for the Advancement of Artificial Intelligence (AAAI)* (2019). URL https://ojs.aaai.org/index.php/AIIDE/article/view/5221.

[42] Albrecht, S. & Stone, P. Autonomous agents modelling other agents: A comprehensive survey and open problems. *Artificial Intelligence* **258**, 66–95 (2018). URL https://www.sciencedirect.com/science/article/pii/S0004370218300249.

[43] Zintgraf, L., Devlin, S., Ciosek, K., Whiteson, S. & Hofmann, K. *Deep interactive bayesian reinforcement learning via meta-learning. Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)* (2021). URL https://dl.acm.org/doi/10.5555/3463952.3464210.

[44] Zhang, Z. *et al. Fast teammate adaptation in the presence of sudden policy change. Conference on Uncertainty in Artificial Intelligence (UAI)* (2023). URL https://proceedings.mlr.press/v216/zhang23a.html.

[45] Long, Q., Li, R., Zhao, M., Gao, T. & Terzopoulos, D. *Inverse attention agents for multi-agent systems. International Conference on Learning Representations (ICLR)* (2025). URL https://proceedings.iclr.cc/paper_files/paper/2025/file/07c7ea6b89aa8381a8bc927e70d67fe0-Paper-Conference.pdf.

[46] Papoudakis, G., Christianos, F. & Albrecht, S. *Agent modelling under partial observability for deep reinforcement learning. Advances in Neural Information Processing Systems (NeurIPS)* (2021). URL https://proceedings.neurips.cc/paper_files/paper/2021/file/a03caec56cd82478bf197475b48c05f9-Paper.pdf.

[47] Barrett, S., Rosenfeld, A., Kraus, S. & Stone, P. Making friends on the fly: Cooperating with new teammates. *Artificial Intelligence* **242**, 132–171 (2017). URL https://www.sciencedirect.com/science/article/pii/S0004370216301266.

[48] Mirsky, R. *et al. A survey of ad hoc teamwork research. Multi-Agent Systems* (2022).

[49] Lou, X. *et al. PECAN: Leveraging policy ensemble for context-aware zero-shot human-AI coordination. International Conference on Autonomous Agents and Multiagent Systems (AAMAS)* (2023). URL https://dl.acm.org/doi/abs/10.5555/3545946.3598700.

[50] Yan, X. *et al. An efficient end-to-end training approach for zero-shot human-AI coordination. Advances in Neural Information Processing Systems (NeurIPS)* (2023). URL https://openreview.net/forum?id=6ePsuwXUwf&noteId=SIiM2yZFMP.

[51] Ruhdorfer, C., Bortoletto, M., Oei, V., Penzkofer, A. & Bulling, A. *Unsupervised partner design enables robust ad-hoc teamwork. International Conference on Machine Learning (ICML)* (2026). URL https://openreview.net/forum?id=0xtMUL0eiF.

[52] Leibo, J. *et al. Scalable evaluation of multi-agent reinforcement learning with Melting Pot. International Conference on Machine Learning (ICML)* (2021). URL https://proceedings.mlr.press/v139/leibo21a.html.

Acknowledgements

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.RS-2020-II201336, Artificial Intelligence Graduate School Program(UNIST)) and the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. RS-2025-25441313, Professional AI Talent Development Program for Multimodal AI Agents).

Author contributions

J.L. conceived the study, developed the methodology, implemented the algorithms, conducted the experiments, analyzed the results, prepared the figures, and wrote the original manuscript. J.K. supervised the research, contributed to the study design and interpretation of the results, and revised the manuscript. Both authors reviewed and approved the final manuscript.

Additional information

Competing interests

The authors declare no competing interests.

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- SupplementaryInformation.pdf