ARTIFICIAL MIRROR NEURONS: MODELLING OTHERS THROUGH LIKE-ME INFERENCE IN MULTI-AGENT REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Humans can understand others partly by using their own experience as a reference, a principle known as Like-Me inference (LIMI). We investigate whether this principle provides a computational mechanism for partner modelling in multi-agent reinforcement learning. We propose a simple LIMI architecture that represents another agent in the ego agent's own embedding space by combining an estimate of the partner's perspective with the ego's own representation mechanism. We study two variants: a fixed perspective transform that requires no behavioural-prediction objective, and a learnt transform that learns the self–other mapping from partner behaviour. We further show theoretically that the benefit of adapting to a partner is bounded by how much partner information is encoded in the agent's hidden state. Across Overcooked, multi-robot coordination, and embodied locomotion tasks, LIMI improves ad hoc teamwork in several settings, particularly when coordinating with unseen partners is difficult. Probing frozen representations further shows that LIMI increases information about partners' actions and internal states. Representational similarity analysis also reveals partial alignment between how self and other encoders in learnt-perspective-transform agents organise the same events, consistent with mirror-like representational structure. Together, our results suggest that reusing one's own representations provides a simple mechanism for implicit partner modelling and behavioural generalisation.

1 INTRODUCTION

Theory of Mind (ToM) – the ability to infer the mental states of others – is fundamental to social cognition. One way humans may understand others is by using their own experience as a reference: observing another individual's situation can recruit representations related to those used for one's own perception and action (Goldman, 2006; Gordon, 1986; Gallese & Goldman, 1998). Mirror-neuron responses provide striking biological evidence for such shared self–other representations, with overlapping neural populations responding during both action observation and execution (Rizzolatti & Craighero, 2004; Del Vecchio et al., 2020; Viaro et al., 2021). Relatedly, *Like-Me Inference* (LIMI) proposes that individuals use knowledge of themselves to interpret the experiences, intentions, and behaviour of others (Meltzoff, 2007).

Deep learning models have demonstrated ToM-like capabilities across domains ranging from belief inference to multi-agent interaction (Rabinowitz et al., 2018; Chen et al., 2025; Mon-Williams et al., 2025). A common approach is to learn models of others through behavioural prediction: given the current state and interaction history, the ego agent predicts what another agent will do next (Albrecht & Stone, 2018; He & Boyd-Graber, 2016; Papoudakis et al., 2021). However, computational approaches based on LIMI remain relatively underexplored. We ask whether reusing the ego agent's own representations can provide a complementary route to partner modelling, allowing information about another agent to emerge without directly training the acting policy to predict that agent's behaviour.

Like-Me inference has previously been explored by Raileanu et al. (2018) for goal inference. Their method reuses the ego policy to infer another agent's goal, but requires multiple inner-loop optimisation steps and backpropagation at inference time. More importantly, it assumes that the partner

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.

can be explained through an explicitly parameterised goal, limiting its applicability to settings where such goals can be specified in advance. We instead propose a representation-based formulation of LIMI that does not assume a predefined latent goal. At each timestep, the ego observation is processed along two pathways: one represents the ego itself, while the other estimates how the same situation would be represented from the partner's perspective. Both representations are expressed in the ego agent's embedding space and integrated by a recurrent policy.

We realise the partner pathway in two ways. In the fixed-transform variant, we explicitly transform the ego observation into the partner's perspective and pass it through a frozen copy of the ego encoder. In the learnt-transform variant, a second encoder learns this relation from behaviour by producing representations that, when passed through the ego agent's own frozen decision-making module, reproduce the partner's behaviour. The key contribution is therefore a more general computational formulation of LIMI in which self–other mapping is imposed directly at the representation level. Both variants represent the partner in the ego agent's own embedding space and integrate this representation with the ego stream through a recurrent policy, providing a general representation-level implementation of LIMI that can be trained within a standard PPO framework (Schulman et al., 2017; de Witt et al., 2020).

We use Multi-Agent Reinforcement Learning (MARL), and particularly Ad Hoc Teamwork (AHT), as a natural testbed for our computational account of Theory of Mind. In standard self-play, agents can succeed by co-adapting to shared conventions without necessarily modelling one another. AHT removes this shortcut by evaluating agents with partners they have never encountered during training (Stone et al., 2010; Hu et al., 2020). Successful coordination therefore requires the ego agent to infer partner-specific behaviour, policies, or intentions from interaction and adapt accordingly. This makes AHT particularly well suited for testing whether LIMI produces richer internal models of others and whether those representations support behavioural generalisation. We evaluate this hypothesis across 10 tasks in Overcooked (Carroll et al., 2019; Wu et al., 2021), JaxRobotarium (Jain et al., 2025), and MA-Brax (Rutherford et al., 2024) environments, spanning discrete coordination, multi-robot interaction, and continuous control.

Across these benchmarks we compare the LIMI-based approaches against standard self-play (SP), E3T (Yan et al., 2023), which robustifies self-play by mixing randomised partners into training, and against the population-based methods FCP (Strouse et al., 2021) and PBT (Carroll et al., 2019), which obtain partner diversity by training against pools of agents at considerable additional compute cost. The LIMI agents improve returns over baselines in several settings. Beyond task performance, we test whether LIMI improves implicit partner modelling. We probe frozen agent representations using two measures: *action prediction gain*, which quantifies how much the ego agent's hidden state improves prediction of the partner's actions relative to a partner-blind control, and *neural prediction gain*, which measures the corresponding improvement in predicting the partner's hidden state. These probes assess whether the ego representation contains information about the partner beyond what is predictable from the environment alone. We find that LIMI generally increases both measures across the evaluated domains, although improvements are not universal, and stronger partner modelling does not always translate into higher ad hoc returns. This suggests that self-simulation improves task performance and also induces internal representations that better capture the behaviour and internal state of unseen partners. In summary, our contributions are:

(1) We introduce a computational implementation of *Like-Me Inference* that represents partners through the ego agent's own representation space, using either a fixed or learnt perspective transform. The fixed variant requires no explicit partner-prediction objective, while the learnt variant learns the self–other mapping directly from behaviour.

(2) We show that LIMI strengthens implicit partner modelling and ad hoc generalisation across several domains, and provide a theoretical analysis showing that partner information is necessary for partner-dependent adaptation but does not guarantee high coordination performance.

(3) We use representational similarity analysis to show that the learnt self and other encoders develop partial alignment in how they organise the same events, consistent with mirror-like representation.

2 Related Work

Machine Theory of Mind and Agent Modelling. A broad literature studies how artificial agents can represent and reason about others. One line explicitly models behaviour or policies, either

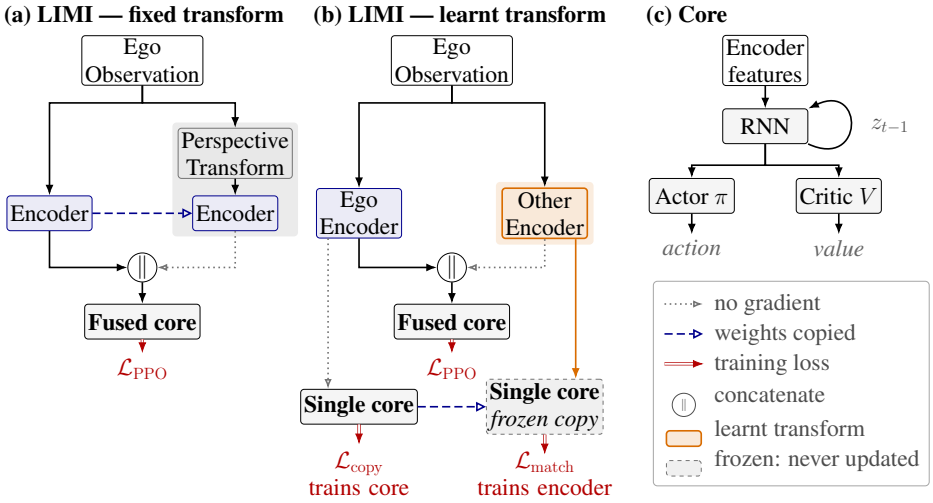

Figure 1: **Like-Me Inference (LIMI)** with a fixed (a) and a learnt (b) perspective transform, and the core both use (c). In (a) the partner stream is the ego observation put through the hardcoded perspective transform $\mathcal{T}_{i\to -i}$ and then through the ego's own encoder. In (b) it is a second encoder e_ψ reading the untransformed observation, trained to make the ego's own policy reproduce the partner behaviour.

jointly with the control policy or through learned policy features (He & Boyd-Graber, 2016; Hong et al., 2018). A second line treats agent modelling as representation learning, learning latent representations of policies, behaviours, or agent characteristics from interaction data (Grover et al., 2018; Rabinowitz et al., 2018; Papoudakis et al., 2021; Sclar et al., 2022; Gandhi et al., 2021; Shu et al., 2021; Nguyen et al., 2023). Related approaches introduce more structured forms of social reasoning, including reasoning about how other agents will respond or learn (Foerster et al., 2018; Wen et al., 2019) and explicitly inferring others' mental states or intentions to support communication and cooperation (Wang et al., 2022). More recently, partner representations have also been shown to emerge spontaneously in recurrent agents trained for cooperation, without an explicit partner-modelling objective (Mon-Williams et al., 2025). Our work differs in the inductive bias used to construct these representations. Rather than learning an unconstrained representation of another agent primarily from its behaviour, we impose a self–other correspondence by representing the partner through the ego agent's own representational machinery.

The closest to our work is Self-Other Modeling (SOM) (Raileanu et al., 2018), which reuses the ego policy to infer another agent's hidden goal by optimising over candidate goals at test time. We share the idea that a self model can serve as a basis for understanding others, but differ in how this is realised. Instead of inferring a goal through online optimisation, LIMI maps the partner's estimated perspective into the ego agent's own representation space, requiring no test-time gradients. We ask if self-based representations alone can induce implicit partner modelling and generalise to unseen partners, a setting not considered in Raileanu et al. (2018). We also compare distinct learning pressures, asking whether behavioural prediction and exposure to diverse partner populations similarly promote the emergence of partner mental-state representations.

Ad Hoc Teamwork and Zero-Shot Coordination. A related body of work studies coordination with unseen partners. Standard self-play can overfit to conventions that work well among co-trained agents but fail to generalise (Carroll et al., 2019). Existing approaches address this problem in several ways. Some seek conventions that are robust to environment symmetries and role permutations (Hu et al., 2020); others increase partner diversity during training, either through populations of policies, generated partners, or mixtures of behaviours (Strouse et al., 2021; Zhao et al., 2023; Li et al., 2023; 2024; Yan et al., 2023; Ruhdorfer et al., 2026). A complementary direction increases environment diversity to encourage more general cooperative strategies (Jha et al., 2025), while other methods explicitly model teammates' behaviour or latent characteristics and condition the policy on these estimates (Wang et al., 2022; 2024; Ni et al., 2026). In contrast, we ask a basic cognitive science question: *can partner modelling emerge from a self–other representational inductive bias, without requiring training against a population of partners?* Overcooked is a standard benchmark

for ad hoc and zero-shot coordination (Carroll et al., 2019; Wu et al., 2021; Ruhdorfer et al., 2025; Gessler et al., 2025). We use it as one testbed and evaluate the same LIMI mechanism in additional multi-agent control domains (Jain et al., 2025; Rutherford et al., 2024), allowing us to test whether the effect extends beyond a single environment family.

3 METHODS

3.1 PROBLEM FORMULATION

We study decentralised cooperation in a multi-agent partially observable Markov decision process (POMDP). A task with N agents is a tuple $\langle \mathcal{S}, \{\mathcal{A}^i\}_{i=1}^N, \{\Omega^i\}_{i=1}^N, P, \{O^i\}_{i=1}^N, R, \gamma \rangle$, where $\mathcal{S}$ is the state space, $\mathcal{A}^i$ and Ω^i are the action and observation spaces of agent i, $P(s_{t+1} \mid s_t, \boldsymbol{a}_t)$ is the transition function under the joint action $\boldsymbol{a}_t = (a_t^1, \ldots, a_t^N)$, $O^i(o_t^i \mid s_t)$ is agent i's observation function, $R(s_t, \boldsymbol{a}_t)$ is a shared team reward, and γ is the discount factor. No agent observes the state; agent i acts on its action–observation history $h_t^i = (o_1^i, a_1^i, \ldots, a_{t-1}^i, o_t^i)$, summarised in practice by the output z_t^i of a recurrent policy, $z_t^i = f(z_{t-1}^i, o_t^i)$ and $a_t^i \sim \pi^i(\cdot \mid z_t^i)$. All agents maximise the expected discounted team return $\mathbb{E}\big[\sum_t \gamma^t R(s_t, \boldsymbol{a}_t)\big]$. Throughout, we write $-i$ for the partner(s) of agent i. The difficulty this setting poses is that the partner is, from the ego agent's point of view, the most consequential unobserved part of the environment: the return depends on the joint action, yet h_t^{-i}, z_t^{-i}, and π^{-i} are all hidden from agent i. Coordination therefore rewards agents whose internal state z_t^i carries predictive information about the partner.

3.2 LIKE-ME INFERENCE

Simulation theory suggests that understanding others can rely on reusing the mechanisms used to represent one's own experience. We adopt this idea by representing the partner in the ego agent's embedding space. Let $x_t^i = e_\phi(o_t^i)$ denote the ego representation, where e_ϕ is the ego observation encoder. If the ego had access to the partner's observation o_t^{-i} and encoder $e_{\phi^{-i}}$, the corresponding partner representation would be $x_t^{-i} = e_{\phi^{-i}}(o_t^{-i})$. In practice, neither is available. LIMI therefore approximates the partner's encoder with e_ϕ and estimates its egocentric observation by perspective transformation, $\hat{o}_t^{-i} = \mathcal{T}_{i \to -i}(o_t^i)$. The resulting partner representation is

$$\hat{x}_t^{-i} = e_\phi(\hat{o}_t^{-i}) \simeq e_{\phi^{-i}}(o_t^{-i}) = x_t^{-i}. \tag{1}$$

Thus, LIMI approximates (1) the partner's viewpoint via perspective transformation and (2) the partner's encoding process using the ego's own encoder. Next, we present two approaches to operationalise LIMI.

Approach 1: Fixed transform. We apply rule-based transforms to the ego observation, to give the ego agent the other agent's perspective (Figure 1a). We suppose that we have access to a perspective transform $\mathcal{T}_{i \to -i} : \Omega^i \to \Omega^i$ and then simply apply a frozen copy of the ego-encoder to obtain:

$$\hat{x}_t^{-i} = e_\phi\big(\mathcal{T}_{i \to -i}(o_t^i)\big), \tag{2}$$

As the encoder is the same one, $\hat{x}_t^{-i}$ is in the ego's own embedding space. In many environments this is simple to compute analytically. In grid worlds, $\mathcal{T}$ changes the ego-centred observation into an approximation of what the partner would see: it places the partner at the centre of the view, rotates the scene to match the partner's heading, and swaps the self and partner feature channels. In the robotic and embodied tasks, $\mathcal{T}$ similarly reorders the observation vector so that positions and role-specific features are expressed from the partner's perspective. In case of partial observability, when the partner is outside the ego's field of view, $\hat{x}_t^{-i}$ is masked with zero. The other pathway carries no learned parameters and gradients flow only through the self pathway, and after every fixed number of updates its weights are overwritten with the current self-encoder weights.

Approach 2: Learnt transform. The fixed transform assumes that the mapping from the ego's viewpoint to the partner's viewpoint is known. When this correspondence is unavailable, we instead learn it. Here the composition $e_\phi \circ \mathcal{T}_{i \to -i}$ is replaced by a second encoder applied directly to the untransformed observation, $\hat{x}_t^{-i} = e_\psi(o_t^i)$. The goal is to train e_ψ so that its output predicts the partner's behaviour through the ego agent's own decision-making machinery. To do this, we use two recurrent modules: a *fused core* and a *single core*. The fused core receives both the ego and partner representations and produces the action and value estimates used to interact with the environment.

The single core receives only the ego representation and is trained to reproduce the ego agent's behaviour; it is auxiliary and does not control the environment. Once trained, the single core is frozen and reused as a readout for the partner representation. The other encoder e_ψ is then trained so that, when $\hat{x}_t^{-i}$ is passed through this frozen ego decision module, it reproduces the partner's behaviour. Intuitively, e_ψ asks: what representation would make the ego agent's own decision process behave as the partner does? The resulting representation can therefore be interpreted as an estimate of the partner's perspective expressed in the ego's own embedding space. This training scheme makes e_ψ play the same role as the shared encoder in Equation (2), but learns the self–other correspondence from behaviour rather than specifying it explicitly. The architecture is shown in Figure 1b.

The fused core receives both ego and partner streams, while a single core with parameters ζ receives only the ego stream and learns to reproduce the fused policy. Here g is a GRU, $\mathrm{sg}[\cdot]$ denotes stop-gradient, and θ and ζ include the recurrent core and actor–critic heads:

$$z_t^i = g\big(\theta, z_{t-1}^i, x_t^i, \mathrm{sg}[\hat{x}_t^{-i}]\big), \tag{3}$$

$$\hat{z}_t^i = g\big(\zeta, \hat{z}_{t-1}^i, \mathrm{sg}[x_t^i]\big), \tag{4}$$

$$\mathcal{L}_{\mathrm{copy}}(\zeta) = \mathrm{KL}\Big(\mathrm{sg}\big[\pi_\theta(\cdot \mid z_t^i)\big] \,\Big\|\, \pi_\zeta\big(\cdot \mid \hat{z}_t^i\big)\Big), \qquad \zeta - \text{trained} \tag{5}$$

The stop-gradients ensure that $\mathcal{L}_{\mathrm{copy}}$ updates only the single core, while the fused core and ego encoder remain shaped by the RL objective. We then freeze the single core and feed it the estimated partner representation $\hat{x}_t^{-i}$. Using the partner's rollout policy π_t^{-i} as the target, we train e_ψ so that the frozen ego decision module reproduces the partner's behaviour:

$$\hat{z}_t^{-i} = g\big(\mathrm{sg}[\zeta], \hat{z}_{t-1}^{-i}, \hat{x}_t^{-i}\big), \tag{6}$$

$$\mathcal{L}_{\mathrm{match}}(\psi) = \mathrm{KL}\Big(\pi_t^{-i} \,\Big\|\, \pi_{\mathrm{sg}[\zeta]}\big(\cdot \mid \hat{z}_t^{-i}\big)\Big), \qquad \hat{x}_t^{-i} - \text{trained} \tag{7}$$

The three parameter groups are therefore updated separately: $\mathcal{L}_{\mathrm{PPO}}$ updates the fused core and ego encoder, $\mathcal{L}_{\mathrm{copy}}$ updates the single core, and $\mathcal{L}_{\mathrm{match}}$ updates the other encoder. Value-network terms are omitted here for brevity and given in Appendix E.

Training. Agents are trained end-to-end with proximal policy optimisation in decentralised self-play: all agents in an episode are controlled by the same learning network, experience is collected from every seat, and the policy and value heads are updated from the shared team reward. We use PPO (Schulman et al., 2017; de Witt et al., 2020) for MARL training. LIMI and baseline agents use the same optimiser and recurrent memory capacity.

3.3 Partner Modelling

Good partner modelling is necessary for adapting to unfamiliar partners, but it does not guarantee high coordination performance. An agent can only adapt its behaviour to different partners if its internal state carries information about them; however, that information is useful only if the agent uses it to select actions that affect return. We formalise this relationship in Appendix C, where we show that the achievable benefit of partner adaptation is bounded jointly by the amount of partner information encoded by the agent and how much its actions matter.

Metrics In practice, a low partner-prediction error does not necessarily imply good partner modelling. In cooperative tasks, the environment itself can make the partner's behaviour predictable: for example, the current state may strongly constrain the optimal action, allowing the ego to anticipate what the partner will do without actually representing the partner. To separate this task-derived predictability from genuine partner modelling, we compare each agent against a *partner-blind control* with the same architecture and training regime but with the partner removed from its observation. We train two one-hidden-layer MLP probes on frozen representations: f_{act} predicts the partner's action distribution, while f_{neu} predicts the partner's hidden state.

Both metrics have the intuitive form $1-\text{error}/\text{blind error}$: they measure how much prediction error is removed when the ego can observe and model the partner, relative to what can already be predicted without seeing the partner. The *action prediction gain* uses cross-entropy error when predicting the partner's current action ($\mathrm{d}t = 0$), while the *neural prediction gain* uses squared error when predicting the partner's hidden state:

$$g_{\mathrm{act}} = 1 - \frac{\mathrm{CE}\big(\pi^{-i}(\cdot \mid z_t^{-i}) \,\|\, f_{\mathrm{act}}(z_t^i)\big)}{\mathrm{CE}\big(\pi^{-i}(\cdot \mid z_t^{-i}) \,\|\, f_{\mathrm{act}}(z_t^{i,\mathrm{blind}})\big)}, \qquad g_{\mathrm{neu}} = 1 - \frac{\mathbb{E}\big\|z_t^{-i} - f_{\mathrm{neu}}(z_t^i)\big\|^2}{\mathbb{E}\big\|z_t^{-i} - f_{\mathrm{neu}}(z_t^{i,\mathrm{blind}})\big\|^2}. \tag{8}$$

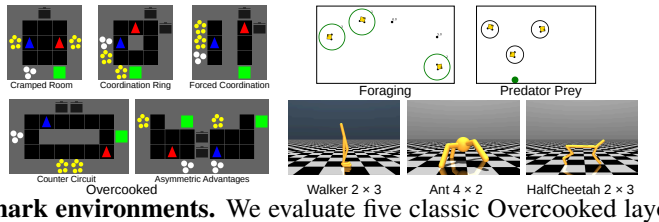

Figure 2: **Benchmark environments.** We evaluate five classic Overcooked layouts, two JaxRobotarium multi-robot coordination tasks (Foraging and Predator–Prey), and three MA-Brax multi-agent continuous-control tasks (Walker, Ant, and HalfCheetah).

For MA-Brax's continuous actions, the action probe predicts the mean of the partner's Gaussian policy before the $\tanh$ transformation. The decoded distribution retains the partner's recorded diagonal variance. We measure prediction error using the KL divergence from the partner distribution to the decoded distribution, averaged over held-out examples. Action prediction gain is then $1 - D_{\text{normal}}/D_{\text{blind}}$, where D denotes this average KL divergence.

Thus, a gain of 0 means that seeing the partner provides no predictive advantage over the blind baseline, while larger positive values indicate that the ego's representation contains additional information about the partner beyond what is already implied by the task state. This normalization factors out partner behaviour that is predictable from environmental regularities alone.

4 Experiments

4.1 Setup

Baselines. **SP** is a recurrent PPO self-play baseline. **E3T** increases partner diversity by mixing self-policy and random-policy partners during training. It also predicts the partner's action and concatenates this prediction with the recurrent features before passing them to the actor head. **SP+rand** uses the same partner mixture as E3T but without an action prediction component. **FCP** and **PBT** obtain partner diversity by training against pools of agents. **SP+LIMI** adds the LIMI pathway of Section 3.2 to SP. **E3T+LIMI** combines this pathway with E3T's random-partner curriculum, without its action-prediction module. We denote the learnt-transform variants by **SP+LIMI (learnt)** and **E3T+LIMI (learnt)**. All methods use a GRU-based recurrent backbone (Chung et al., 2014), with matched input and hidden-state widths in the acting policy. Fixed-transform LIMI preserves the GRU input dimensionality by concatenating two half-width encoder outputs, using shared or copied encoder weights. Learnt-transform variants instead use a separately trained other encoder. We evaluate five trained policies per method.

Environments. Figure 2 summarises the three benchmark domains. **Overcooked:** We use the five classic layouts under the standard full-observation setting adopted in prior work (Carroll et al., 2019; Strouse et al., 2021; Yan et al., 2023; Jha et al., 2025; Ruhdorfer et al., 2026). Both agents observe the entire kitchen, with their observations differing only in the assignment of self and partner feature channels. The perspective transform therefore simply exchanges these channels, expressing the same state from the partner's perspective without adding information. Any benefit of LIMI must consequently arise from its representation rather than access to additional observations. **JaxRobotarium:** We use two multi-robot coordination tasks (Jain et al., 2025). Foraging adapts Level-Based Foraging (Christianos et al., 2020): three robots have heterogeneous capabilities and must combine sufficient capability to collect resources of varying difficulty. Predator–Prey adapts the classic multi-agent predator–prey task (Lowe et al., 2017): three predators must coordinate to tag a faster heuristic prey that moves away from its nearest predator. **MA-Brax:** We use three continuous-control tasks in which the actuators of a single body are partitioned among agents sharing a team reward and full physical observations (Peng et al., 2021; Rutherford et al., 2024). Walker 2×3 assigns one three-joint leg to each of two agents, Ant 4×2 assigns the hip and ankle of each leg to one of four agents, and HalfCheetah 2×3 divides the three-joint rear and front chains between two agents. Successful locomotion therefore requires agents to coordinate control of the same body. Details of the observations in these environments are provided in Appendix F.

Evaluation Protocol. For ad-hoc evaluation, each trained ego seed is paired with a *fixed pool of held-out SP partners* per task or layout, trained under the same regime but from disjoint seeds. These partners are never encountered during ego training. In Overcooked, actions are sampled from the policy following the standard evaluation convention, and returns are averaged over both

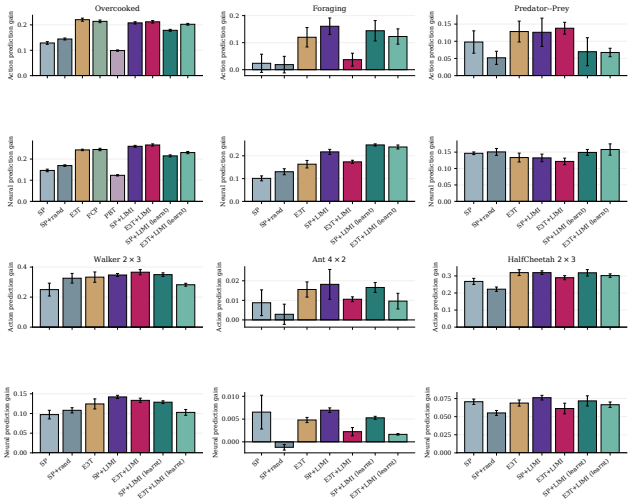

Figure 3: **Partner-Modelling Gains** across Overcooked, Foraging, Predator–Prey, Walker 2×3, Ant 4×2, and HalfCheetah 2×3. Each environment has an action-gain panel above a neural-gain panel. Overcooked averages its five classic layouts. Gains are defined in Section 3.3. Bars show partner medians within ego seeds, then mean $\pm$ SE across seeds.

seat assignments. Robotarium and MA-Brax follow the same ad-hoc protocol. On Robotarium, actions are selected greedily; on MA-Brax, deterministic evaluation uses the mean action of the tanh-squashed Gaussian policy. Partner-modelling probes are one-hidden-layer MLPs trained on training episodes and evaluated on held-out episodes. Action prediction uses the partner's policy distribution as a soft target, while both action and neural prediction gains are computed relative to the partner-blind control. We report instantaneous action prediction and neural prediction gains.

4.2 Results

Partner modelling: LIMI strengthens partner representations Figure 3 shows that partner-predictive information can emerge without an explicit partner-prediction objective. Adding the fixed-transform LIMI pathway raises SP's action prediction gain from 0.129 to 0.208 and its neural prediction gain from 0.146 to 0.260. E3T+LIMI attains a similar neural gain (0.266). These two agents have the highest mean neural gains among the compared methods, exceeding both E3T and the population-based baselines. Their action gains (0.208 and 0.212) approach those of FCP (0.214) and E3T (0.220), although neither fixed-transform LIMI agent is trained to predict its partner. Thus, direct prediction supervision is not necessary for the acting representation to become informative about partner behaviour and internal state. This pattern extends beyond Overcooked. On Foraging, SP+LIMI raises action gain from 0.024 to 0.161 and neural gain from 0.101 to 0.217; on Walker, the corresponding changes are $0.250 \rightarrow 0.347$ and $0.097 \rightarrow 0.142$. HalfCheetah shows smaller increases, particularly for neural prediction. Predator–Prey is an exception to the joint improvement: SP+LIMI's mean action gain is higher than SP's, but its neural gain is slightly lower. On Ant, both gains remain close to zero across methods. Robotarium comparisons use the available finite probe estimates, so incomplete coverage limits their interpretation.

The learnt transform does not uniformly improve on the fixed correspondence. It yields the highest mean neural gain on Foraging (0.248 for SP+LIMI (learnt)), but both learnt-transform agents have lower neural gains than their fixed-transform counterparts on Overcooked. Unlike the fixed transform, the learnt transform receives action and value supervision during training; claims of emergence without partner supervision therefore apply only to the fixed-transform agents. The architecture-matched ablations in Appendix B isolate the effect of the fixed perspective transform.

LIMI can improve Ad-hoc Teamwork. On Overcooked, Table 1 shows that LIMI improves average return over SP, without producing a uniform advantage across layouts or training regimes. SP+LIMI raises the five-layout average from 104 ± 6 to 128 ± 4 ($+23.6\%$), while its learnt-transform counterpart reaches 131 ± 3. E3T+LIMI (learnt) has the highest average, 149 ± 3 ($+43.9\%$ over SP), followed by SP+rand at 143 ± 4 and fixed-transform E3T+LIMI at 142 ± 3. Layout-specific rankings differ: PBT leads on Counter Circuit, FCP on Forced Coordination, and SP+rand on Co-

Table 1: **Overcooked ad-hoc returns against 20 held-out SP partners.** Means ± SE over five ego seeds. The average weights the five raw layout scores equally. Parentheses () show improvement over the SP average. Bold denotes the highest unrounded mean and * the second-highest unrounded mean in each column; ranks do not imply statistical significance.

Method	Cramped Room	Counter Circuit	Coord. Ring	Asymm. Adv.	Forced Coord.	Average
SP	153 ± 7	42 ± 10	143 ± 22	107 ± 8	74 ± 12	104 ± 6 (+0.0%)
SP+rand	166 ± 15	37 ± 3	$\mathbf{198 \pm 6}$	211 ± 13	104 ± 3	143 ± 4 (+38.3%)*
E3T	166 ± 7	51 ± 7	167 ± 4	180 ± 17	104 ± 4	134 ± 4 (+29.0%)
FCP	150 ± 8	42 ± 1	142 ± 11	147 ± 8	$\mathbf{110 \pm 3}$	118 ± 3 (+14.1%)
PBT	77 ± 5	$\mathbf{69 \pm 2}$	104 ± 3	200 ± 8	55 ± 4	101 ± 2 (-2.7%)
SP+LIMI	170 ± 3	42 ± 8	153 ± 15	182 ± 8	93 ± 3	128 ± 4 (+23.6%)
E3T+LIMI	$\mathbf{181 \pm 4}$	47 ± 8	172 ± 7	221 ± 10*	90 ± 5	142 ± 3 (+37.4%)
SP+LIMI (learnt)	173 ± 2*	56 ± 12*	148 ± 8	199 ± 6	80 ± 7	131 ± 3 (+26.5%)
E3T+LIMI (learnt)	165 ± 6	56 ± 4	181 ± 11*	$\mathbf{239 \pm 5}$	105 ± 3*	$\mathbf{149 \pm 3}$ **(+43.9%)**

Table 2: **Other environments' ad-hoc returns against 20 held-out SP partners.** Each cell gives mean ± SE over five ego seeds and percentage improvement over SP in the same environment. Bold denotes the highest mean and * the second-highest mean in each column.

Method	Foraging	Predator–Prey	Walker 2×3	Ant 4×2	HalfCheetah 2×3
SP	7.42 ± 0.94 (+0.0%)*	4.35 ± 1.06 (+0.0%)	294 ± 21 (+0%)	159 ± 209 (+0%)	1450 ± 334 (+0%)
SP+rand	5.24 ± 1.08 (-29.4%)	11.10 ± 3.82 (+155.0%)*	297 ± 17 (+1%)	401 ± 67 (+152%)	1563 ± 80 (+8%)*
E3T	5.09 ± 1.06 (-31.5%)	9.32 ± 1.23 (+114.0%)	336 ± 26 (+14%)	576 ± 70 (+263%)	1480 ± 173 (+2%)
SP+LIMI	$\mathbf{8.46 \pm 0.10}$ **(+14.0%)**	6.06 ± 1.91 (+39.2%)	424 ± 28 (+44%)*	611 ± 87 (+285%)	1442 ± 241 (-1%)
E3T+LIMI	4.72 ± 0.78 (-36.5%)	10.52 ± 0.65 (+141.6%)	406 ± 47 (+38%)	537 ± 65 (+238%)	1343 ± 202 (-7%)
SP+LIMI (learnt)	7.08 ± 0.96 (-4.7%)	6.79 ± 2.29 (+55.9%)	384 ± 26 (+31%)	$\mathbf{836 \pm 118}$ **(+427%)**	$\mathbf{1597 \pm 347}$ **(+10%)**
E3T+LIMI (learnt)	6.45 ± 1.15 (-13.1%)	$\mathbf{12.93 \pm 2.24}$ **(+197.1%)**	$\mathbf{458 \pm 59}$ **(+55%)**	689 ± 33 (+334%)*	1321 ± 155 (-9%)

ordination Ring. The result of E3T vs SP-rand suggests that partner randomisation is therefore a strong baseline, and adding a partner-modelling pathway does not always improve on it. On Foraging, fixed-transform SP+LIMI has the highest mean return, 8.46 ± 0.10, compared with SP's 7.42 ± 0.94; all random-partner methods have lower means than SP. On Predator–Prey, random-partner training is more successful: SP+rand reaches 11.10 ± 3.82, compared with SP's 4.35 ± 1.06, and E3T+LIMI (learnt) has the highest mean, 12.93 ± 2.24. The learnt transform therefore leads on one task and the fixed transform on the other. Several differences are modest relative to their seed-level uncertainty, so these rankings should not be read as established performance separations.

On Walker, SP+LIMI reaches 424 ± 28 versus SP's 294 ± 21, while E3T+LIMI (learnt) has the highest mean at 458 ± 59. On Ant, the learnt-transform agents, SP+LIMI (learnt) and E3T+LIMI (learnt), reach 836 ± 118 and 689 ± 33 under SP and random-partner training, respectively. Their large SP-relative percentages partly reflect the low and uncertain SP baseline (159 ± 209). HalfCheetah provides no comparably clear advantage: SP+LIMI (learnt) has the highest mean (1597 ± 347), but SP (1450 ± 334) and SP+rand (1563 ± 80) are close relative to the reported uncertainty. Overall, stronger partner-prediction gains and higher coordination returns are distinct outcomes. For example, E3T exceeds SP+rand on both gains in Overcooked and HalfCheetah, yet has lower mean returns. This dissociation is consistent with the one-sided theoretical bound in Appendix C.

Self–other representational alignment in learnt-transform agents. Inspired by mirror-neurons literature (Gallese & Goldman, 1998), which shows that observing another individual perform an action can elicit neural responses similar to those produced when performing the same action oneself, we ask whether the learnt other encoder represents cooking events in a similar way to the self encoder. Particularly, we study SP+LIMI (learnt) and E3T+LIMI (learnt), using representational similarity analysis (Kriegeskorte et al., 2008) on Overcooked's cooking events: *pick up onion, pick up plate, place onion in pot, pick up soup, and deliver soup*. We restrict the study to three of the Overcooked layouts because both chefs have symmetric capabilities, avoiding role-specific asymmetries that could confound comparisons between self and other representations. We find partial self–other alignment: across all three layouts and all 30 trained-model–layout comparisons, same events are represented more similarly across the two encoders than different events. The matched-event advantage Δ_{event} ranges from 0.12 to 0.39, while agreement in event geometry ρ is strongest in Counter Circuit (0.80 ± 0.03 for SP+LIMI and 0.61 ± 0.08 for E3T+LIMI) and weakest in Cramped Room. The alignment is therefore consistent but incomplete, and its strength depends on the layout. Full methodology, controls, and results are given in Appendix A.

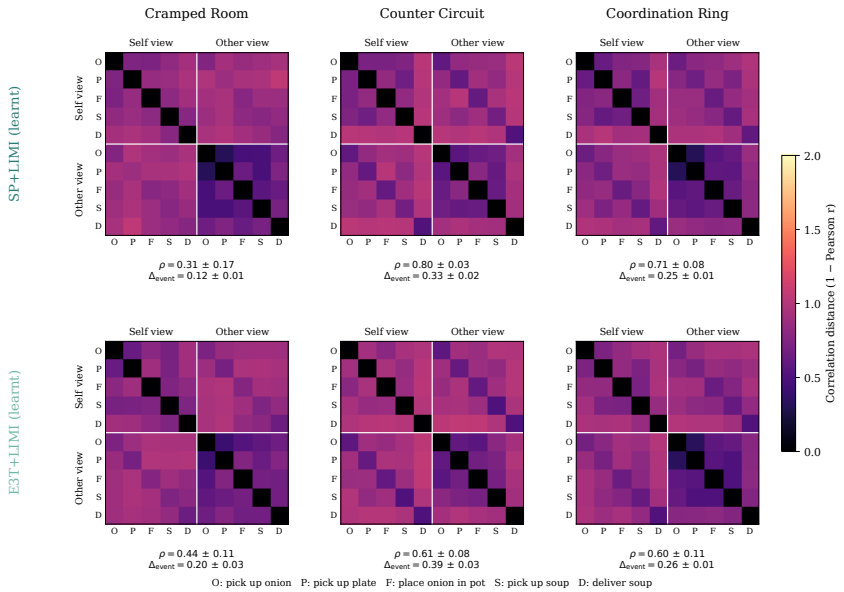

Figure 4: **Self–other representational alignment in learnt-transform agents.** Rows show the two learnt-transform methods and columns three Overcooked layouts. Each panel is an Representational Dissimilarity Matrix (RDM) over five self-view and five other-view cooking events; darker cells indicate more similar representations. The cross-stream diagonal compares matching events. Annotations report event-structure agreement (ρ) and the matched-event advantage (Δ_{event}). See Appendix A for full methodology and controls.

5 Discussion

Modelling others is a fundamental capability in social interaction. Previous computational-cognitive-science work shows that predicting others' behaviour can induce representations of higher-level mental states such as goals, intentions, and beliefs (Rabinowitz et al., 2018; Gandhi et al., 2021). Our results show that a different inductive bias can also support partner modelling: representing others through the ego agent's own representational machinery. We also find that exposure to diverse partners can strengthen partner modelling. This is consistent with our theoretical analysis in Appendix C: successful adaptation to different partners requires the agent to encode partner information, but strong partner modelling alone does not guarantee high coordination performance.

E3T illustrates the behavioural-prediction route, achieving strong action and neural prediction gains through its explicit partner-action prediction module. LIMI instead tests whether Like-Me inference can support partner modelling and behavioural generalisation, while the learnt-transform variant shows that self-based representations can also benefit from behavioural supervision.

The two LIMI variants support different interpretations. The fixed transform uses no behavioural-prediction objective, so partner information emerges from the LIMI architecture and coordination objective alone. The learnt transform instead uses partner behaviour to train e_ψ through Equation (7), while keeping the readout frozen and the acting policy free from direct partner-prediction supervision. The fixed transform therefore provides the cleaner test of emergent partner modelling, while the learnt transform tests whether the self–other mapping itself can be learnt from behaviour.

Limitations. Our formulation makes several simplifying assumptions. First, ego agents and partners remain mutually observable; limited fields of view would additionally require memory or belief over unobserved partners. Second, the fixed transform assumes a known perspective transformation. This is natural for structured vector observations, but more complex sensory inputs such as 3D vision may require a learnt view-synthesis model. Third, the learnt transform requires access to the partner's actions and value estimates during training. These are available in self-play, but may not be available for external or human partners. Finally, we study only one to a few partners; how to represent many partners simultaneously in the ego's embedding space remains an open question.

AI USE STATEMENT

We used generative AI tools to assist with code implementation, visualisation, the editing or paraphrasing of text, and the formalisation of mathematical proofs that were conceptualised by the authors. We did not use generative AI to develop the core methods or to implement the underlying reinforcement-learning training pipeline from scratch; the RL training code was adapted from the JaxMARL codebase. We also did not use generative AI to identify or cite related work. All AI-assisted code, figures, proofs, and revised text were reviewed and verified by two authors. We take full responsibility for the final manuscript and all associated code and artefacts.

REPRODUCIBILITY STATEMENT

Our reinforcement-learning implementation is built on the JaxMARL codebase. We will release the code, model checkpoints, training configurations, and evaluation scripts upon publication.

REFERENCES

Stefano V Albrecht and Peter Stone. Autonomous agents modelling other agents: A comprehensive survey and open problems. *Artificial Intelligence*, 258:66–95, 2018.

Micah Carroll, Rohin Shah, Mark K. Ho, Tom Griffiths, Sanjit A. Seshia, Pieter Abbeel, and Anca D. Dragan. On the utility of learning about humans for human-ai coordination. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 5175–5186, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/f5b1b89d98b7286673128a5fb112cb9a-Abstract.html.

Ruirui Chen, Weifeng Jiang, Chengwei Qin, and Cheston Tan. Theory of mind in large language models: Assessment and enhancement. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pp. 31539–31558. Association for Computational Linguistics, 2025. doi: 10.18653/V1/2025.ACL-LONG.1522. URL https://doi.org/10.18653/v1/2025.acl-long.1522.

Filippos Christianos, Lukas Schäfer, and Stefano V. Albrecht. Shared experience actor-critic for multi-agent reinforcement learning. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/7967cc8e3ab559e68cc944c44b1cf3e8-Abstract.html.

Junyoung Chung, Çaglar Gülçehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR*, abs/1412.3555, 2014. URL http://arxiv.org/abs/1412.3555.

Christian Schröder de Witt, Tarun Gupta, Denys Makoviichuk, Viktor Makoviychuk, Philip H. S. Torr, Mingfei Sun, and Shimon Whiteson. Is independent learning all you need in the starcraft multi-agent challenge? *CoRR*, abs/2011.09533, 2020. URL https://arxiv.org/abs/2011.09533.

Maria Del Vecchio, Fausto Caruana, Ivana Sartori, Veronica Pelliccia, Flavia Maria Zauli, Giorgio Lo Russo, Giacomo Rizzolatti, and Pietro Avanzini. Action execution and action observation elicit mirror responses with the same temporal profile in human sii. *Communications biology*, 3 (1):80, 2020.

Jakob N. Foerster, Richard Y. Chen, Maruan Al-Shedivat, Shimon Whiteson, Pieter Abbeel, and Igor Mordatch. Learning with opponent-learning awareness. In Elisabeth André, Sven Koenig, Mehdi Dastani, and Gita Sukthankar (eds.), *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS 2018, Stockholm, Sweden, July 10-15, 2018*,

pp. 122–130. International Foundation for Autonomous Agents and Multiagent Systems Richland, SC, USA / ACM, 2018. URL http://dl.acm.org/citation.cfm?id=3237408.

Vittorio Gallese and Alvin Goldman. Mirror neurons and the simulation theory of mind-reading. *Trends in cognitive sciences*, 2(12):493–501, 1998.

Kanishk Gandhi, Gala Stojnic, Brenden M. Lake, and Moira R. Dillon. Baby intuitions benchmark (BIB): discerning the goals, preferences, and actions of others. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 9963–9976, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/525b8410cc8612283c9ecaf9a319f8ed-Abstract.html.

Tobias Gessler, Tin Dizdarevic, Ani Calinescu, Benjamin Ellis, Andrei Lupu, and Jakob Nicolaus Foerster. Overcookedv2: Rethinking overcooked for zero-shot coordination. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=hlvLM3GX8R.

Alvin I Goldman. *Simulating minds: The philosophy, psychology, and neuroscience of mindreading*. Oxford University Press, 2006.

Robert M Gordon. Folk psychology as simulation. *Mind & language*, 1(2):158–171, 1986.

Aditya Grover, Maruan Al-Shedivat, Jayesh K. Gupta, Yuri Burda, and Harrison Edwards. Learning policy representations in multiagent systems. In Jennifer G. Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1797–1806. PMLR, 2018. URL http://proceedings.mlr.press/v80/grover18a.html.

He He and Jordan L. Boyd-Graber. Opponent modeling in deep reinforcement learning. In Maria-Florina Balcan and Kilian Q. Weinberger (eds.), *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pp. 1804–1813. JMLR.org, 2016. URL http://proceedings.mlr.press/v48/he16.html.

Zhang-Wei Hong, Shih-Yang Su, Tzu-Yun Shann, Yi-Hsiang Chang, and Chun-Yi Lee. A deep policy inference q-network for multi-agent systems. In Elisabeth André, Sven Koenig, Mehdi Dastani, and Gita Sukthankar (eds.), *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS 2018, Stockholm, Sweden, July 10-15, 2018*, pp. 1388–1396. International Foundation for Autonomous Agents and Multiagent Systems Richland, SC, USA / ACM, 2018. URL http://dl.acm.org/citation.cfm?id=3237907.

Hengyuan Hu, Adam Lerer, Alex Peysakhovich, and Jakob N. Foerster. "other-play" for zero-shot coordination. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 4399–4410. PMLR, 2020. URL http://proceedings.mlr.press/v119/hu20a.html.

Shalin Jain, Jiazhen Liu, Siva Kailas, and Harish Ravichandar. Jaxrobotarium: Training and deploying multi-robot policies in 10 minutes. In *Proceedings of the 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 1975–1996. PMLR, 2025.

Kunal Jha, Wilka Carvalho, Yancheng Liang, Simon Shaolei Du, Max Kleiman-Weiner, and Natasha Jaques. Cross-environment cooperation enables zero-shot multi-agent coordination. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, volume 267 of *Proceedings of Machine Learning Research*. PMLR / OpenReview.net, 2025. URL https://proceedings.mlr.press/v267/jha25b.html.

Nikolaus Kriegeskorte, Marieke Mur, and Peter A Bandettini. Representational similarity analysis-connecting the branches of systems neuroscience. *Frontiers in systems neuroscience*, 2:249, 2008.

Yang Li, Shao Zhang, Jichen Sun, Yali Du, Ying Wen, Xinbing Wang, and Wei Pan. Cooperative open-ended learning framework for zero-shot coordination. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 20470–20484. PMLR, 2023. URL https://proceedings.mlr.press/v202/li23au.html.

Yang Li, Shao Zhang, Jichen Sun, Wenhao Zhang, Yali Du, Ying Wen, Xinbing Wang, and Wei Pan. Tackling cooperative incompatibility for zero-shot human-ai coordination. *J. Artif. Intell. Res.*, 80: 1139–1185, 2024. doi: 10.1613/JAIR.1.15884. URL https://doi.org/10.1613/jair.1.15884.

Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 6379–6390, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/68a9750337a418a86fe06c1991a1d64c-Abstract.html.

Andrew N Meltzoff. 'like me': a foundation for social cognition. *Developmental science*, 10(1): 126–134, 2007.

Ruaridh Mon-Williams, Max Taylor-Davies, Elizabeth Mieczkowski, Natalia Vélez, Neil Bramley, Yanwei Wang, Tom Griffiths, and Christopher G. Lucas. Partner modelling emerges in recurrent agents (but only when it matters). In Danielle Belgrave, Cheng Zhang, Laura N. Montoya, Hsuan-Tien Lin, Razvan Pascanu, Piotr Koniusz, Marzyeh Ghassemi, Nancy Chen, Iván Vladimir Meza Ruíz, and Arturo Loaiza-Bonilla (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025. URL http://papers.nips.cc/paper_files/paper/2025/hash/0b8e8bfc40184226888e821620b216c9-Abstract-Conference.html.

Dung Nguyen, Phuoc Nguyen, Hung Le, Kien Do, Svetha Venkatesh, and Truyen Tran. Memory-augmented theory of mind network. In Brian Williams, Yiling Chen, and Jennifer Neville (eds.), *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*, pp. 11630–11637. AAAI Press, 2023. doi: 10.1609/AAAI.V37I10.26374. URL https://doi.org/10.1609/aaai.v37i10.26374.

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 Viviana Mascardi, John Thangarajah, Chris Amato, and Louise A. Dennis (eds.), *Proceedings of the 25th International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2026, Paphos, Cyprus, May 25-29, 2026*, pp. 1810–1819. ACM, 2026. doi: 10.65109/UMCV8852. URL https://dl.acm.org/doi/10.65109/UMCV8852.

Georgios Papoudakis, Filippos Christianos, and Stefano V. Albrecht. Agent modelling under partial observability for deep reinforcement learning. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 19210–19222, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/a03caec56cd82478bf197475b48c05f9-Abstract.html.

Bei Peng, Tabish Rashid, Christian Schröder de Witt, Pierre-Alexandre Kamienny, Philip H. S. Torr, Wendelin Boehmer, and Shimon Whiteson. FACMAC: factored multi-agent centralised policy gradients. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang,

and Jennifer Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 12208–12221, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/65b9eea6e1cc6bb9f0cd2a47751a186f-Abstract.html.

Neil C. Rabinowitz, Frank Perbet, H. Francis Song, Chiyuan Zhang, S. M. Ali Eslami, and Matthew M. Botvinick. Machine theory of mind. In Jennifer G. Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pp. 4215–4224. PMLR, 2018. URL http://proceedings.mlr.press/v80/rabinowitz18a.html.

Roberta Raileanu, Emily Denton, Arthur Szlam, and Rob Fergus. Modeling others using oneself in multi-agent reinforcement learning. In Jennifer G. Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pp. 4254–4263. PMLR, 2018. URL http://proceedings.mlr.press/v80/raileanu18a.html.

Giacomo Rizzolatti and Laila Craighero. The mirror-neuron system. *Annu. Rev. Neurosci.*, 27(1): 169–192, 2004.

Constantin Ruhdorfer, Matteo Bortoletto, Anna Penzkofer, and Andreas Bulling. The overcooked generalisation challenge: Evaluating cooperation with novel partners in unknown environments using unsupervised environment design. *Trans. Mach. Learn. Res.*, 2025, 2025. URL https://openreview.net/forum?id=K2KtcMlW6j.

Constantin Ruhdorfer, Matteo Bortoletto, Victor Oei, Anna Penzkofer, and Andreas Bulling. Unsupervised partner design enables robust ad-hoc teamwork. In *International Conference on Machine Learning*, 2026.

Alexander Rutherford, Benjamin Ellis, Matteo Gallici, Jonathan Cook, Andrei Lupu, Garðar Ingvarsson, Timon Willi, Ravi Hammond, Akbir Khan, Christian Schröder de Witt, Alexandra Souly, Saptarashmi Bandyopadhyay, Mikayel Samvelyan, Minqi Jiang, Robert T. Lange, Shimon Whiteson, Bruno Lacerda, Nick Hawes, Tim Rocktäschel, Chris Lu, and Jakob N. Foerster. Jaxmarl: Multi-agent RL environments and algorithms in JAX. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/5aee125f052c90e326dcf6f380df94f6-Abstract-Datasets_and_Benchmarks_Track.html.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017. URL http://arxiv.org/abs/1707.06347.

Melanie Sclar, Graham Neubig, and Yonatan Bisk. Symmetric machine theory of mind. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato (eds.), *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pp. 19450–19466. PMLR, 2022. URL https://proceedings.mlr.press/v162/sclar22a.html.

Tianmin Shu, Abhishek Bhandwaldar, Chuang Gan, Kevin A. Smith, Shari Liu, Dan Gutfreund, Elizabeth S. Spelke, Joshua B. Tenenbaum, and Tomer D. Ullman. AGENT: A benchmark for core psychological reasoning. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 9614–9625. PMLR, 2021. URL http://proceedings.mlr.press/v139/shu21a.html.

Peter Stone, Gal A. Kaminka, Sarit Kraus, and Jeffrey S. Rosenschein. Ad hoc autonomous agent teams: Collaboration without pre-coordination. In Maria Fox and David Poole (eds.), *Proceedings*

of the Twenty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2010, Atlanta, Georgia, USA, July 11-15, 2010, pp. 1504–1509. AAAI Press, 2010. doi: 10.1609/AAAI.V24I1.7529. URL https://doi.org/10.1609/aaai.v24i1.7529.

DJ Strouse, Kevin R. McKee, Matt M. Botvinick, Edward Hughes, and Richard Everett. Collaborating with humans without human data. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 14502–14515, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/797134c3e42371bb4979a462eb2f042a-Abstract.html.

Riccardo Viaro, Emma Maggiolini, Emanuele Farina, Rosario Canto, Atsushi Iriki, Alessandro D'Ausilio, and Luciano Fadiga. Neurons of rat motor cortex become active during both grasping execution and grasping observation. *Current Biology*, 31(19):4405–4412, 2021.

Caroline Wang, Arrasy Rahman, Ishan Durugkar, Elad Liebman, and Peter Stone. N-agent ad hoc teamwork. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/cabf611498431ad89a85ace75f790d93-Abstract-Conference.html.

Yuanfei Wang, Fangwei Zhong, Jing Xu, and Yizhou Wang. Tom2c: Target-oriented multi-agent communication and cooperation with theory of mind. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=2t7CkQXNpuq.

Ying Wen, Yaodong Yang, Rui Luo, Jun Wang, and Wei Pan. Probabilistic recursive reasoning for multi-agent reinforcement learning. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=rkl6As0cF7.

Sarah A Wu, Rose E Wang, James A Evans, Joshua B Tenenbaum, David C Parkes, and Max Kleiman-Weiner. Too many cooks: Bayesian inference for coordinating multi-agent collaboration. *Topics in Cognitive Science*, 13(2):414–432, 2021.

Xue Yan, Jiaxian Guo, Xingzhou Lou, Jun Wang, Haifeng Zhang, and Yali Du. An efficient end-to-end training approach for zero-shot human-ai coordination. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/07a363fd2263091c2063998e0034999c-Abstract-Conference.html.

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. In Brian Williams, Yiling Chen, and Jennifer Neville (eds.), *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*, pp. 6145–6153. AAAI Press, 2023. doi: 10.1609/AAAI.V37I5.25758. URL https://doi.org/10.1609/aaai.v37i5.25758.

Appendix Contents

A Self–other representational alignment in Learnt Transform Variants **15**

B Ablation: perspective transformation strengthens partner modelling **17**

C Theoretical Analysis: Partner adaptation and partner modelling **17**

D Proof of Proposition 1 **18**

E The learnt transform in full **20**

F Environment observations **21**

A Self–other representational alignment in Learnt Transform Variants

To compare the methods fairly, we show every agent the same recorded cooking situations. Therefore, we collect these recordings using 20 SP agent checkpoints, instead of SP+LIMI (learnt) or E3T+LIMI (learnt), each checkpoint controlling both chefs for 32 episodes of 400 steps. Using a shared dataset prevents differences in the situations encountered by each method from affecting the comparison. We record five types of successful event: picking up an onion, picking up a plate, placing an onion in a pot, picking up soup, and delivering soup. An event counts as successful only when the interaction produces the corresponding change in the environment. Immediately before each event, we save both chefs' observations, giving us two views of the same situation: the view of the chef performing the event and the view of its teammate. We retain at most one example of each event type per chef per episode. For example, if chef 0 picks up an onion five times in one episode, we randomly select one of those pickups. We apply the same rule separately to chef 1. This prevents frequently repeated actions within an episode from dominating the dataset.

We construct a shared collection of cooking situations from SP self-play episodes. For each layout, the RDM analysis uses 128 recorded examples of each of five successful events, separately for each acting chef, giving 1,280 situations in total. Each situation contains both chefs' observations immediately before the event. We replay this identical collection through every evaluated checkpoint and extract its self-view and other-view encoder features. Thus, differences between methods reflect their representations of the same situations, rather than differences in the situations they encountered. These examples come from SP source seeds 10–19; examples from seeds 0–9 are reserved for a separate individual-feature analysis.

Extracting self and other representations. For each recorded cooking event, we ask how the evaluated model represents the situation from two viewpoints. We feed the acting chef's observation into the model's self encoder, and its teammate's observation into the model's learnt other encoder:

$$x_t^{\text{self}} = e_\phi(o_t^j), \qquad x_t^{\text{other}} = e_\psi(o_t^i). \tag{9}$$

Here, chef j performs the event and chef i observes it. Both encoders belong to the same evaluated LIMI model. The SP agents only provide the recorded situations; we do not use their internal representations.

Each encoder produces a list of 256 numbers describing its input. We extract these features after LayerNorm, before the two streams are combined and before recurrent processing. The other encoder receives the teammate's original observation, without a hand-written perspective transform. We use the saved model weights without modifying them.

Importantly, the evaluated models are viewing recorded situations, not performing the recorded actions themselves. This analysis therefore compares their immediate responses to the two viewpoints, rather than their memory or behaviour over time.

Constructing the representational dissimilarity matrix. We use a representational dissimilarity matrix (RDM) to show which cooking events produce similar internal responses (Kriegeskorte et al., 2008). For each event, we average the features from its 128 validation examples. We do this separately for the self view and other view, and separately for each acting chef and trained model:

$$\mu^{v}_{a,c} = \frac{1}{128} \sum_{n=1}^{128} x^{v}_{a,c,n}. \tag{10}$$

Here, c identifies the cooking event, a identifies the acting chef, and v identifies the self or other view. This gives ten average response patterns: five events represented by the self encoder and the same five events represented by the other encoder. Each row and column of the 10×10 matrix corresponds to one event–view pair. For each acting chef a, we compute

$$D^{(a)}_{(c,v),(c',v')} = 1 - \mathrm{corr}\left(\mu^{v}_{a,c},\, \mu^{v'}_{a,c'}\right), \tag{11}$$

where c and c' identify cooking events, and $v, v' \in \{\mathrm{self}, \mathrm{other}\}$ identify the encoder views.

A distance near 0 means that the two patterns are highly similar; a distance of 1 means that they are uncorrelated; and a distance near 2 means that they vary in opposite directions. Similarity concerns the pattern of feature responses, not their absolute size: two patterns can have zero distance without containing exactly the same numbers.

Each matrix has four blocks. The upper-left block compares cooking events within the self encoder, and the lower-right block compares them within the other encoder. The upper-right block compares events across the two encoders. Its diagonal contains the comparisons of greatest interest: self-view onion pickup versus other-view onion pickup, self-view plate pickup versus other-view plate pickup, and so on. Darker colours indicate smaller distances. A darker diagonal within this cross-encoder block therefore suggests that the two encoders respond more similarly to the same event than to different events. This is distinct from the main diagonal of the entire matrix, which is always zero because it compares each pattern with itself. The lower-left block contains the same cross-encoder comparisons in the reverse order.

We first compute an RDM separately for each acting chef and trained model. We then average the two chef-specific matrices, followed by the matrices from the five training seeds. We do not average feature values across independently trained models before calculating their distances.

Agreement in event structure (ρ). We first ask whether the self and other encoders agree about which cooking events resemble one another. For example, if the self encoder represents onion pickup as more similar to pot filling than to soup delivery, does the other encoder make the same distinction? We measure this using the Spearman correlation ρ between the ten unique event-pair distances within each encoder. A high value indicates agreement in how the encoders organise the five events, but does not require their feature vectors to be similar.

Event-matching metric (Δ_{event}). We separately ask whether matching events are closer across encoders than different events. Let $C_{cc'}$ be the correlation distance between the self encoder's average response to event c and the other encoder's average response to event c'. We compute

$$\Delta_{\mathrm{event}} = \underbrace{\frac{1}{20} \sum_{c \neq c'} C_{cc'}}_{\text{different events}} - \underbrace{\frac{1}{5} \sum_{c} C_{cc}}_{\text{matching events}}. \tag{12}$$

A positive value means that pairing the same event across encoders gives a closer match, on average, than pairing different events. This measures event-specific correspondence rather than overall similarity: matching events may still be far apart, provided that mismatched events are even farther apart. For both metrics, we first compute the score separately for each acting chef and trained model. We then average over the two acting chefs within each model and report the mean $\pm$ standard error across five training seeds. These errors describe variation between trained models on the fixed recorded dataset; individual event examples and matrix cells are not treated as independent training runs.

Table 3: Perspective-transform (PT) ablation for fixed-transform methods: raw returns (mean ± SE over five ego seeds) using the same partner pools as Tables 1 and 2. Overcooked is the equally weighted mean of its five raw layout scores. Removing PT retains the second stream but supplies the untransformed observation. Within each SP+LIMI or E3T+LIMI pair, bold denotes the higher mean.

Method	Overcooked	Foraging	Predator–Prey	Walker 2×3	Ant 4×2	HalfCheetah 2×3
SP+LIMI	$\mathbf{128 \pm 4}$	$\mathbf{8.46 \pm 0.10}$	$\mathbf{6.06 \pm 1.91}$	$\mathbf{424 \pm 28}$	$\mathbf{611 \pm 87}$	$\mathbf{1442 \pm 241}$
SP+LIMI w/o PT	122 ± 4	4.34 ± 0.85	5.38 ± 1.37	329 ± 10	545 ± 177	1440 ± 303
E3T+LIMI	$\mathbf{142 \pm 3}$	$\mathbf{4.72 \pm 0.78}$	$\mathbf{10.52 \pm 0.65}$	406 ± 47	537 ± 65	$\mathbf{1343 \pm 202}$
E3T+LIMI w/o PT	140 ± 3	3.70 ± 0.15	5.00 ± 1.50	$\mathbf{434 \pm 27}$	$\mathbf{559 \pm 64}$	1326 ± 140

Controls and interpretation. Randomly shuffling the other encoder's event labels produces average scores close to zero, providing a sanity check on the correspondence measures. However, the two views already contain the same full-grid scene information, with the chefs' features reordered. RDMs computed directly from these observations have a self/other correlation of 1. The encoder results therefore show how much event structure is preserved across the learnt pathways, not that the networks discover correspondence absent from their inputs. In particular, this analysis cannot distinguish representations of actions from representations of the objects, inventories, and locations associated with those actions.

B Ablation: perspective transformation strengthens partner modelling

We isolate the contribution of perspective transformation (PT) by comparing each fixed-transform LIMI agent with an architecture-matched ablation that retains the second encoder pathway but receives the untransformed observation. Tables 3 and 4 report returns and partner-prediction gains, respectively, using the same evaluation pools as the main comparisons. Overcooked results are averaged over its five layouts.

Across our ten-task benchmark, five Overcooked layouts, two Robotarium tasks, and three MABrax tasks, the overall pattern is that PT increases both action and neural prediction gains. These improvements indicate that the correspondence imposed by PT, rather than the additional pathway alone, strengthens partner modelling. The pattern is not universal: on Predator–Prey, SP+LIMI has a slightly lower neural prediction gain than its no-PT counterpart.

Stronger partner modelling does not, however, consistently translate into higher returns. For example, E3T+LIMI improves both prediction gains on Walker and Ant, yet its mean return is lower than that of its no-PT ablation. This dissociation is not specific to PT. The result is consistent with the well-established baselines. In particular, E3T also has lower mean returns than SP+rand on several tasks, including the Overcooked layout average and HalfCheetah, even though SP+rand retains E3T's partner-randomisation curriculum while removing its explicit action-prediction module.

Improvements in partner modelling do not necessarily translate into better generalisation performance, consistent with our theoretical analysis (Appendix C). Partner information constrains the possible benefit of adaptation, but the bound is one-sided: possessing more information does not guarantee that a policy uses it to select actions that improve coordination. Prediction gains therefore diagnose partner information in the representation, whereas returns assess whether the resulting policy coordinates successfully.

C Theoretical Analysis: Partner adaptation and partner modelling

The goal of this subsection is to give a theoretical link between ad hoc performance and partner modelling. Coordinating better with an unfamiliar partner means behaving differently depending on which partner it is, and an agent can only do that if its hidden state carries information about the partner. To make this precise, let W be the partner's behavioural class and let Σ_t be what the ego agent perceives at time t without inferring anything about the partner, namely its current

Table 4: Perspective-transform ablation in action and neural prediction gains for fixed-transform methods. Within each layout or task, we take the median over partner checkpoints for each ego seed, then report the mean ± SE across valid ego seeds. Overcooked averages its five layout means and combines their SEs in quadrature. Within each SP+LIMI or E3T+LIMI pair, bold denotes the higher mean.

Method	Overcooked	Foraging	Predator–Prey	Walker 2 × 3	Ant 4 × 2	HalfCheetah 2 × 3
Action prediction gain						
SP+LIMI	$\mathbf{0.208 \pm 0.005}$	$\mathbf{0.161 \pm 0.031}$	$\mathbf{0.126 \pm 0.041}$	$\mathbf{0.347 \pm 0.010}$	$\mathbf{0.018 \pm 0.008}$	$\mathbf{0.319 \pm 0.011}$
SP+LIMI w/o PT	0.132 ± 0.004	0.030 ± 0.008	0.070 ± 0.038	0.308 ± 0.014	0.004 ± 0.004	0.265 ± 0.014
E3T+LIMI	$\mathbf{0.212 \pm 0.005}$	$\mathbf{0.037 \pm 0.024}$	$\mathbf{0.138 \pm 0.017}$	$\mathbf{0.366 \pm 0.017}$	$\mathbf{0.011 \pm 0.001}$	$\mathbf{0.290 \pm 0.012}$
E3T+LIMI w/o PT	0.157 ± 0.004	0.014 ± 0.007	-0.005 ± 0.009	0.290 ± 0.013	-0.011 ± 0.008	0.228 ± 0.009
Neural prediction gain						
SP+LIMI	$\mathbf{0.260 \pm 0.004}$	$\mathbf{0.217 \pm 0.011}$	0.132 ± 0.012	$\mathbf{0.142 \pm 0.004}$	$\mathbf{0.007 \pm 0.001}$	$\mathbf{0.076 \pm 0.003}$
SP+LIMI w/o PT	0.176 ± 0.009	0.071 ± 0.005	$\mathbf{0.140 \pm 0.010}$	0.108 ± 0.004	0.000 ± 0.002	0.061 ± 0.007
E3T+LIMI	$\mathbf{0.266 \pm 0.005}$	$\mathbf{0.173 \pm 0.007}$	$\mathbf{0.122 \pm 0.010}$	$\mathbf{0.133 \pm 0.006}$	$\mathbf{0.002 \pm 0.001}$	$\mathbf{0.061 \pm 0.007}$
E3T+LIMI w/o PT	0.193 ± 0.005	0.079 ± 0.012	0.085 ± 0.043	0.127 ± 0.004	-0.002 ± 0.001	0.056 ± 0.003

observation o_t^i, which includes whatever of the partner is instantaneously visible. Write $\bar{\pi}_t(a \mid \sigma) = \Pr_\pi(a_t^i = a \mid \Sigma_t = \sigma)$ for the Σ-marginal of an ego policy π: the same policy with its action–partner correlation removed but its per-observation action statistics preserved. The value of adaptation is $\delta(\pi) = J_\rho(\pi) - J_\rho(\bar{\pi})$, with J_ρ the expected team return against a partner drawn from the pool ρ.

Proposition 1 (Adaptation is capped by stakes times partner information)**.** *Let $X_t = (\Sigma_t, W)$, let $Q_t^{\bar{\pi}}, V_t^{\bar{\pi}}$ be the value functions of $\bar{\pi}$ on this augmented state, and write $\mathrm{span}_t = \max_a Q_t^{\bar{\pi}}(X_t, a) - \min_a Q_t^{\bar{\pi}}(X_t, a)$ for the range of action-values at X_t and $I_t(\pi) = I(z_t^i; W \mid \Sigma_t)$ for the partner information in the ego's hidden state, in bits. Then*

$$\begin{aligned} \delta(\pi) \;&=\; \sum_{t=1}^{T} \mathbb{E}_\pi\big[\, Q_t^{\bar{\pi}}(X_t, a_t^i) - V_t^{\bar{\pi}}(X_t) \,\big] \\ &\le\; \sum_{t=1}^{T} \sqrt{\mathbb{E}\big[\mathrm{span}_t^2\big]} \;\cdot\; \sqrt{\tfrac{\ln 2}{2}\, I_t(\pi)} \end{aligned} \tag{13}$$

where the first line is an exact identity and the second bounds it term by term.

Interpretation Low partner information forces low adaptation, while high partner information allows but does not guarantee adaptation, so an agent that models well without coordinating better is consistent with Equation (13).

We do not estimate δ directly, which would require rolling out each agent's Σ-marginal; we measure ad hoc return $J_\rho(\pi)$ against a pool of held-out partners (Section 4.2). Raw return is not itself a measure of adaptation, since a competent agent that keeps out of the way can score well without conditioning on its partner.

D Proof of Proposition 1

Assumptions. We work in the setting of Appendix C: horizon T, team reward $R \in [0, 1]$, and a partner drawn once per episode from the pool ρ with behavioural class W, fixed within an episode and never revealed to the ego. We assume

(A1) (Σ_t, W) evolves as a controlled Markov chain under the ego's actions, and the expected team reward at time t is a function of (Σ_t, W, a_t^i);

(A2) Σ_t is recoverable from z_t^i, and the ego's action depends on the partner only through its own hidden state, so that conditional on Σ_t,

$$W \;\longrightarrow\; z_t^i \;\longrightarrow\; a_t^i \tag{14}$$

is a Markov chain.

(A2) holds by construction of general RL environments, the ego agent does not observe W, so every dependence of a_t^i on the partner is routed through z_t^i. (A1) is a modelling assumption. (A1) does not require the partner to be stationary, since a partner with its own evolving internal state still yields a controlled Markov chain once that state is absorbed into W. All mutual informations are measured under the trajectory distribution induced by π.

Augmented process. Let $X_t = (\Sigma_t, W)$ and define the one-step reward and transition of an augmented Markov decision process by marginalising the partner's action out of the original game,

$$r\big((\sigma, w), a\big) = \mathbb{E}\big[R \mid \Sigma_t = \sigma,\, W = w,\, a_t^i = a\big], \tag{15}$$

with W held constant by the transition. By (A1) this is well defined, and by $R \in [0,1]$ also $r \in [0,1]$. The cross-play return of any ego policy in the original game equals its value in this augmented process, so we may work in the latter throughout.

Because the Σ-marginal $\bar{\pi}_t$ of Appendix C is a function of σ alone, it is in particular a (time-dependent) Markov policy on X_t. Let $V_t^{\bar{\pi}}$ and $Q_t^{\bar{\pi}}$ denote its finite-horizon value and action-value functions, with $V_{T+1}^{\bar{\pi}} \equiv 0$.

Step 1: the exact identity. Write $J_\rho(\pi) = \mathbb{E}_\pi\big[\sum_{t=1}^T r(X_t, a_t^i)\big]$ and $J_\rho(\bar{\pi}) = \mathbb{E}\big[V_1^{\bar{\pi}}(X_1)\big]$. Adding and subtracting the telescoping sum $\sum_{t=1}^T \big(V_{t+1}^{\bar{\pi}}(X_{t+1}) - V_t^{\bar{\pi}}(X_t)\big) = -V_1^{\bar{\pi}}(X_1)$ under π's trajectory law,

$$J_\rho(\pi) - J_\rho(\bar{\pi}) = \sum_{t=1}^T \mathbb{E}_\pi\Big[\, r(X_t, a_t^i) + V_{t+1}^{\bar{\pi}}(X_{t+1}) - V_t^{\bar{\pi}}(X_t)\Big]. \tag{16}$$

By the Markov property of the augmented process, $\mathbb{E}\big[r(X_t, a_t^i) + V_{t+1}^{\bar{\pi}}(X_{t+1}) \mid X_t, a_t^i\big] = Q_t^{\bar{\pi}}(X_t, a_t^i)$,

$$\delta(\pi) \;=\; \sum_{t=1}^T \mathbb{E}_\pi\Big[\, Q_t^{\bar{\pi}}(X_t, a_t^i) - V_t^{\bar{\pi}}(X_t)\Big]. \tag{17}$$

This is the finite-horizon performance-difference identity; note it is an equality, it requires no relation between the occupancies of π and $\bar{\pi}$, and it holds for history-dependent π. Intuitively, adaptation here is defined as the accumulated advantage of the actions π actually took over what its own marginal would have done, each valued under the marginal's future.

Step 2: one step, in total variation. Fix $x = (\sigma, w)$ and let

$$q_t(a \mid \sigma, w) = \Pr_\pi\big(a_t^i = a \mid \Sigma_t = \sigma,\, W = w\big) \tag{18}$$

be π's action distribution given both the perceived state and the partner class. Since $V_t^{\bar{\pi}}(x) = \sum_a \bar{\pi}_t(a \mid \sigma)\, Q_t^{\bar{\pi}}(x, a)$,

$$\mathbb{E}_\pi\big[Q_t^{\bar{\pi}} - V_t^{\bar{\pi}} \,\big|\, X_t = x\big] = \sum_a \big(q_t(a \mid \sigma, w) - \bar{\pi}_t(a \mid \sigma)\big)\, Q_t^{\bar{\pi}}(x, a). \tag{19}$$

The coefficients sum to zero, so any constant may be subtracted from $Q_t^{\bar{\pi}}(x, \cdot)$ without changing the value. Subtracting the midpoint $c_t(x) = \frac{1}{2}\big(\max_a Q_t^{\bar{\pi}}(x, a) + \min_a Q_t^{\bar{\pi}}(x, a)\big)$ leaves every term bounded by $\mathrm{span}_t(x)/2$ in absolute value, and $\sum_a |q_t - \bar{\pi}_t| = 2\,\mathrm{TV}(q_t, \bar{\pi}_t)$, so

$$\mathbb{E}_\pi\big[Q_t^{\bar{\pi}} - V_t^{\bar{\pi}} \,\big|\, X_t = x\big] \;\le\; \mathrm{span}_t(x) \;\cdot\; \mathrm{TV}\big(q_t(\cdot \mid \sigma, w),\, \bar{\pi}_t(\cdot \mid \sigma)\big). \tag{20}$$

Subtracting the midpoint rather than bounding $Q_t^{\bar{\pi}}$ by the remaining horizon is what replaces the crude factor $T - t + 1$ by the span: what limits the payoff of deviating at time t is not how much return is left, but how much the action taken at t changes it.

Step 3: from total variation to bits. Taking expectations over X_t in Equation (20) and applying Cauchy–Schwarz,

$$\mathbb{E}_\pi\big[Q_t^{\bar{\pi}} - V_t^{\bar{\pi}}\big] \;\le\; \sqrt{\mathbb{E}_\pi\big[\mathrm{span}_t^2\big]} \;\cdot\; \sqrt{\mathbb{E}_\pi\big[\mathrm{TV}^2\big]}. \tag{21}$$

Pinsker's inequality in squared form gives $\mathrm{TV}^2 \le \frac{1}{2} D_{\mathrm{KL}}\big(q_t(\cdot \mid \sigma, w) \,\|\, \bar{\pi}_t(\cdot \mid \sigma)\big)$ in nats. Because $\bar{\pi}_t(\cdot \mid \sigma)$ is exactly the W-marginal of $q_t(\cdot \mid \sigma, W)$, averaging that divergence over (Σ_t, W) is by definition a conditional mutual information,

$$\mathbb{E}_\pi\big[\mathrm{TV}^2\big] \;\le\; \tfrac{1}{2}\, I_{\mathrm{nats}}\big(a_t^i;\, W \mid \Sigma_t\big) \;=\; \tfrac{\ln 2}{2}\, I_{\mathrm{bits}}\big(a_t^i;\, W \mid \Sigma_t\big). \tag{22}$$

Finally, the Markov chain in Equation (14) and the conditional data-processing inequality give

$$I_{\mathrm{bits}}\big(a_t^i;\, W \mid \Sigma_t\big) \;\le\; I_{\mathrm{bits}}\big(z_t^i;\, W \mid \Sigma_t\big) \;=\; I_t(\pi), \tag{23}$$

which substituted above bounds the t-th term of Equation (17); summing over t gives Equation (13). □

E THE LEARNT TRANSFORM IN FULL

This appendix gives the learnt-transform agent's forward pass and both auxiliary losses in full, including the value terms omitted from Section 3.2 for brevity.

Correspondence with the notation of Section 3.2. The listings name tensors as the implementation does:

paper	code	what it is
x_t^i	`x_ego`	ego representation, $e_\phi(o_t^i)$
$\hat{x}_t^{-i}$	`x_other`	estimated partner representation, $e_\psi(o_t^i)$
z_t^i	`z`	fused-core state; drives the rollout policy
$\hat{z}_t^i$	`z_solo`	single-core state on the ego's own representation
$\hat{z}_t^{-i}$	`z_hat`	single-core state on the partner representation
ϕ	`phi_ego`	ego encoder parameters
ψ	`phi_other`	other encoder parameters
θ	`theta_fused`	fused core: recurrence and actor/critic heads
ζ	`zeta`	single core: recurrence and actor/critic heads

Forward pass. One timestep, with `sg` the stop-gradient. Both encoders read the same ego observation; with P partners the other encoder is read once per partner k, with a slot one-hot that lets one shared encoder tell them apart, and the results are mean-pooled so the fused input stays two streams wide however many agents there are. Each partner keeps its own single-core carry, so partner k is tracked individually over time using the ego's own dynamics.

Representations

```
x_ego      = layer_norm(mlp(obs, phi_ego))                          x_t^i
x_other[k] = layer_norm(mlp([obs, slot_onehot(k)], phi_other))      x̂_t^{-i}
```

Fused core: the policy that is actually rolled out

```
fused   = mean_k(sg(x_other[k]))
z, cz   = gru_fused([x_ego, fused], cz, theta_fused)                 z_t^i
pi      = categorical(actor_fused(z, theta_fused))                   π_θ(· | z_t^i)
v       =             critic_fused(z, theta_fused)
```

Single core, ego branch: input frozen, weights train

```
z_solo, cs = gru_single(sg(x_ego), cs, zeta)                         ẑ_t^i
pi_solo    = categorical(actor_single(z_solo, zeta))                 π_ζ
v_solo     =             critic_single(z_solo, zeta)
```

Single core, partner branch: weights frozen, input trains

```
z_hat[k], ck[k] = gru_single(x_other[k], ck[k], sg(zeta))           ẑ_t^{-i}
pi_hat[k]       = categorical(actor_single(z_hat[k], sg(zeta)))
v_hat[k]        =             critic_single(z_hat[k], sg(zeta))
```

Note that on the partner branch the frozen pathway includes the recurrence, not only the output layer: what is transplanted onto the partner is the ego's own dynamics.

Losses. The partner targets are arrays stored at rollout time rather than live tensors, so no gradient reaches the partner through them; combined with `sg(zeta)` above, the whole partner gradient lands on `phi_other` and nowhere else. In $\mathcal{L}_{\text{copy}}$ both the teacher and the input are stop-gradiented, so distillation can move only ζ – it can neither reshape ϕ nor compete with the RL objective. The partner value term uses a Huber loss rather than a squared error: that gradient lands on ψ alone, and an unbounded one drove $\hat{x}_t^{-i}$, and with it the whole agent, to NaN.

```
L_ppo = ppo_clip(pi, v, traj)                          rollout policy only; unchanged from SP

L_copy   = kl(sg(pi), pi_solo)                                                    Eq. 5
L_copy_v = 0.5 * (v_solo - sg(v))**2

L_match   = mean_k(kl(traj.partner_pi[k], pi_hat[k]))                             Eq. 7
L_match_v = mean_k(huber(v_hat[k], traj.partner_v[k], delta=1.0))

loss = L_ppo + 1.0 * L_match + 0.5 * L_match_v + 1.0 * L_copy + 0.5 *
    L_copy_v
```

`sg(x_other)` in the fused core keeps the RL objective off the other encoder, so $\hat{x}_t^{-i}$ stays an estimate of the partner rather than whatever happens to help the ego's return. `sg(x_ego)` in the ego branch makes the single core a pure readout, so distillation cannot reshape the ego's own representation. `sg(zeta)` in the partner branch is the one that makes this like-me inference rather than a partner model: with the readout free to adapt, the single core would become a dedicated partner predictor and nothing would identify what it represents.

Hyperparameters. Those below are the learnt transform's; every other setting – optimiser, network widths, rollout length, PPO constants – is shared with the baselines and listed with them.

setting	value	effect
`PARTNER_ACTION_COEF`	1.0	weight on $\mathcal{L}_{\text{match}}$
`PARTNER_VALUE_COEF`	0.5	weight on its value term
`DISTILL_COEF`	1.0	weight on $\mathcal{L}_{\text{copy}}$
`VF_DISTILL_COEF`	0.5	weight on its value term
`SINGLE_RNN_HIDDEN_DIM`	128	single-core width, half the fused width
`FUSE_PARTNER_ENCODER`	true	one shared e_ψ, read once per partner
`PARTNER_SLOT_ONEHOT`	true	lets the shared encoder tell partners apart
`PERSPECTIVE_TRANSFORM`	false	the learnt transform; true recovers the fixed one
`RL_GRAD_TO_OTHER_ENCODER`	false	the first stop-gradient above

F Environment observations

Overcooked The five classic kitchens use spatial feature maps with 30 channels encoding the two chefs' positions, orientations and held objects, kitchen stations, loose objects, pot contents, recipe information and cooking timers. The observation covers the entire kitchen layout, with shape $H \times W \times 30$. Both chefs see the same scene; their observations differ only in which chef occupies the self and partner channel blocks. The fixed perspective transform swaps these blocks without adding information.

Robotarium. Policies receive global numerical features. In both tasks, the three robots' planar poses (x, y, θ) are listed in cyclic order, starting with the observing robot. Foraging appends the positions and resource levels of two resources and the three robots' capabilities, giving an 18-dimensional vector. Predator–Prey appends the prey's pose to the three predator poses, giving 12 dimensions. The fixed perspective transform reorders robot poses and, in Foraging, capabilities; resource or prey features remain unchanged.

MA-Brax. We use full-observation Walker 2×3, Ant 4×2 and HalfCheetah 2×3. Each agent receives the complete Brax observation vector: its own joint positions and velocities, the other agents' joint blocks in cyclic order, and shared body features. These contain 17, 27 and 18 physical

values, respectively. All methods additionally receive one visibility bit per value and a one-hot agent-role identifier, yielding policy inputs of 36, 58 and 38 dimensions. Visibility bits are all one during ordinary full-observation execution. Physical values are normalised, using saved training statistics. The fixed transform reorders features and visibility masks and updates the role identifier without adding body information.