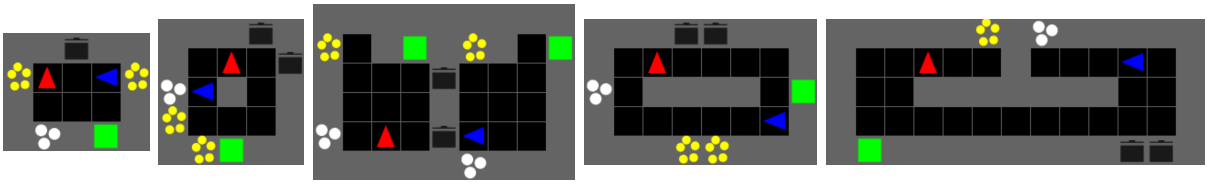
Intention modeling mediates cooperative incompatibility via character separability Supplementary Information

August 12, 2026

Supplementary Methods

Environment details



Supplementary Fig. S1. Layouts of the Overcooked environment. From left to right: Cramped Room, Coordination Ring, Asymmetric Coordination, Counter Circuit, and Blocked Corridor.

The pot starts cooking once three ingredients are placed in it. The agent receives a +20 reward for each delivery. We use a shaped reward signal for sub-goals with a linearly decaying weight during training, alongside the sparse delivery reward. A +3 shaped reward for placing an ingredient into the pot and for picking up a plate while the soup is cooking, and a +5 shaped reward for plating the cooked soup. The observation consists of 26 channels, each representing a distinct object or feature of the environment—agents, agent orientations, pots, ingredients, and the pot timer, among others; an urgency channel activates when fewer than 40 timesteps remain. Three layouts (Cramped Room, Coord. Ring, Counter Circuit) originate from the original Overcooked-AI [1], and two (Asymm. Coordination, Blocked Corridor) from ZSC-Eval [2].

Building test population

The test pool consists of 64 partners in total: 32 are trained with vanilla IPPO, differing only in their random seeds, and the remaining 32 are trained with the additional cross-population MEP objective—8 per training population size in $\{8, 16, 32, 64\}$, each against the corresponding training population. The cross-population MEP coefficient is annealed from 0 to its final value of 0.1.

Implementation details

all agents and classifiers share a CNN-LSTM backbone [3–5]. A pre-LSTM projection layer is inserted between the convolutional encoder and the LSTM to match the flattened dimension.

The intention classifier uses a single shared MLP whose final layer splits into action and cluster logits; we adopt this shared form for implementation simplicity. Architectural details are reported in Supplementary Table S1.

Training configurations are shared across layouts for both partners and ego agents, except for the number of iterations, the reward-shaping iterations, and the early-stopping iterations. Training stops early if the best mean return does not improve within the number of iterations listed under Early stopping iterations. We use Generalized Advantage Estimation (GAE) [6] for bootstrapped temporal-difference targets and Adam [7] as the optimizer for all. All experiments are implemented in Jax [8] and Flax [9]. Training details are reported in Supplementary Table S2.

Component	Value
Activation	ReLU
Input	26×H×W (H, W depending on layout)
Convolution out channels	[32, 32, 16]
Convolution strides	[1, 1, 1]
Convolution kernel size	[3×3, 3×3, 3×3]
Pre-LSTM projection dimension	(16×H×W)×128
LSTM hidden dimension	128
Actor head dimension (Implicit)	[128×128, 128×6]
Actor head dimension (Explicit)	[256×128, 128×6]
Critic head dimension (Implicit)	[128×128, 128×1]
Critic head dimension (Explicit)	[256×128, 128×1]
Intention classifier MLP (shared)	[128×128, 128×(#clusters+6)]
Probing dimension	[128×128, 128×(prediction target)]

Supplementary Table S1. Details on architectural configurations.

Component		Value
Partner clustering threshold		0.4
Discount rate (γ)		0.985
GAE λ		0.95
PPO clip (ϵ)		0.2
PPO epochs		8
PPO minibatches		2
Value loss coefficient		0.5
Entropy loss coefficient		0.05
Learning rate (intention, probing)		1e-5
Learning rate (others)		1e-4
Optimizer		Adam
Adam β_1		0.9
Adam β_2		0.999
Adam ϵ		1e-6
Weight decay		0.0
Gradient clipping		1.0
Batch size (ego agent, probing)		128
Batch size (population)		64
Episode length		400
Partner prioritized sampling coefficient		1.0
MEP coefficient		0.01
Cross-population MEP coefficient (test partners)		0.1
Reward shaping iteration ratio		0.75
number of iterations for probing		10000
Number of iterations (ego agent)	Cramped Room	3500
	Asymm. Coordination	25000
	Coord. Ring	40000
	Counter Circuit	40000
	Blocked Corridor	80000
Number of iterations (population)	Cramped Room	3000
	Asymm. Coordination	15000
	Coord. Ring	15000
	Counter Circuit	15000
	Blocked Corridor	25000
Early stopping iterations	Cramped Room	1000
	Asymm. Coordination	4000
	Coord. Ring	5000
	Counter Circuit	5000
	Blocked Corridor	5000

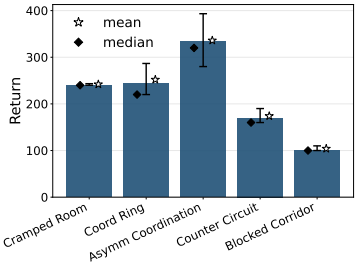
Supplementary Table S2. Details on training configurations.

Supplementary Note

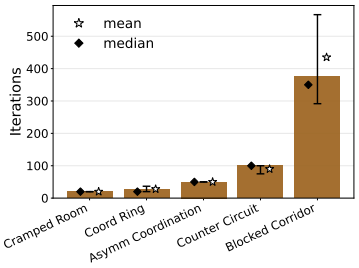
Details on layouts

Difficulty

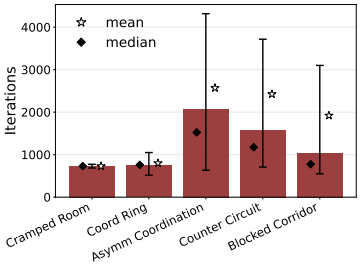
We characterize layout difficulty by three measures: the maximum self-play return, the number of iterations to obtain a first reward, and the number of iterations to reach 90% of the best return—indicating the trajectory length to reward, the sparsity of the reward signal, and the triviality of the coordination policy, respectively. As shown in Supplementary Fig. S2, Blocked Corridor and Counter Circuit are the most complex layouts, while Asymm. Coordination has a dense reward signal but a non-trivial coordination policy.



(a) Best mean return per layout



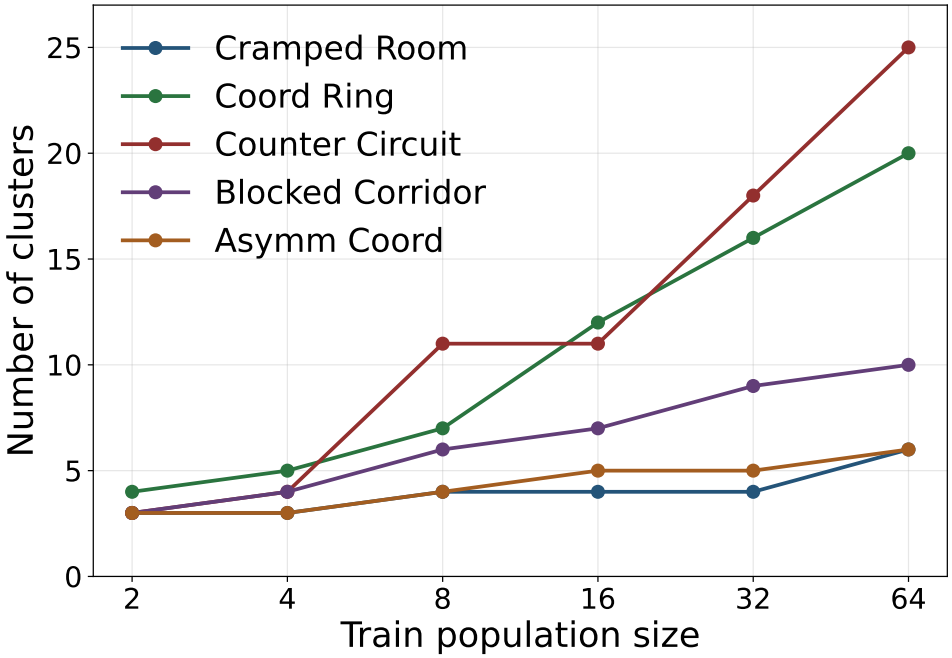
(b) Iterations to first reward



(c) Iterations to 90% of best return

Supplementary Fig. S2. Complexity of layouts. Bars indicate the IQM over 10 seeds per layout; error bars are 95% bootstrap CIs over seeds. Cramped Room and Coord. Ring are comparatively easy and trivial; Asymm. Coordination has a dense reward signal but a non-trivial coordination policy; Blocked Corridor and Counter Circuit are the most complex layouts.

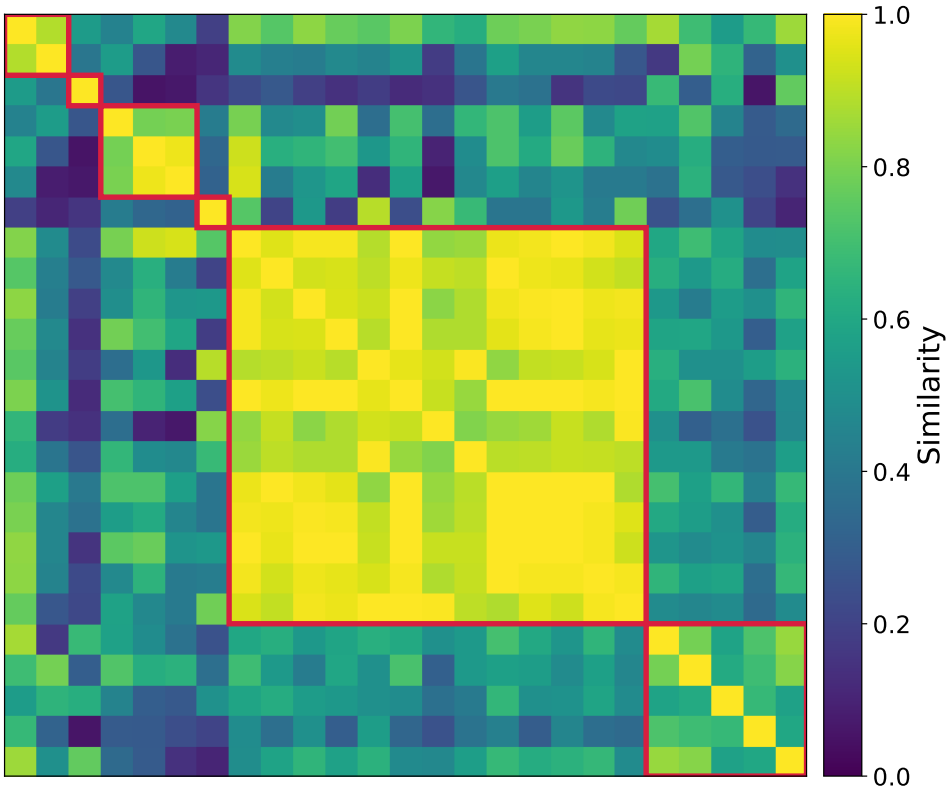
Population diversity



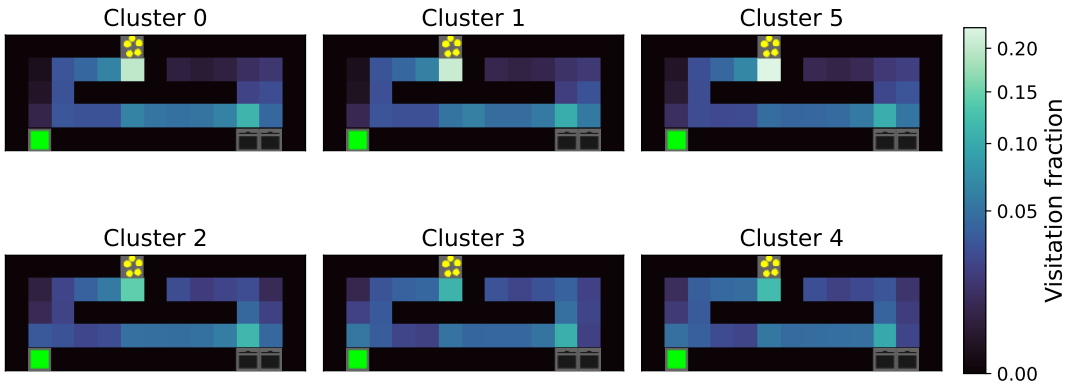
Supplementary Fig. S3. Number of behavioral clusters across population sizes and layouts. The x-axis is the number of expert partners; the y-axis is the number of behavioral clusters across all levels. The cluster count scales up in Counter Circuit and Coord. Ring.

We train partners with MEP [10] across population sizes $\{2, 4, 8, 16, 32, 64\}$. The number of behavioral clusters does not increase linearly with population size and varies across layouts, though it does grow consistently in some layouts (Counter Circuit and Coord. Ring); for a broader treatment of population diversity, see McKee et al. [11]. On Blocked Corridor, some expert partners fail to obtain any reward: 1, 7, and 12 partners at populations of 16, 32, and 64, respectively. We exclude these collapsed partners from both ego-agent training and evaluation; population labels refer to the nominal trained counts. The resulting cluster counts are shown in Supplementary Fig. S3.

Case study on Blocked Corridor with a population of 32



Supplementary Fig. S4. Cross-play similarity and behavioral clusters on Blocked Corridor with a population of 32. Red outlines mark cluster boundaries. Clusters are largely compatible with one another; one partner forms a singleton cluster.



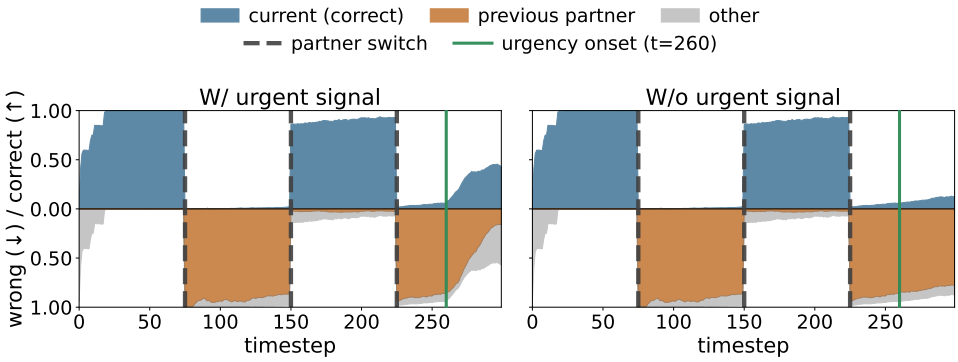
Supplementary Fig. S5. Cell-visitation frequencies of the ego agent against each expert cluster on Blocked Corridor with a population of 32. The responses collapse into two patterns that cover all expert clusters.

As reported in Subsection 4.2 in main manuscript, on Blocked Corridor with a population of 32, ego agents achieve well-coordinated performance even with weakly separable character representations (Fig. 5 of main manuscript). Two observations explain this. First, the behavioral clusters in this setting are largely compatible with one another despite their number (Supplementary Fig. S4); one partner forms a singleton cluster, behaviorally distinct in similarity space,

yet is still covered by one of the response policies below. Second, two response policies suffice to cover all expert clusters: the ego agent's cell-visitation frequencies against each expert cluster collapse into two patterns (Supplementary Fig. S5).

Additional results on changing partner behavior

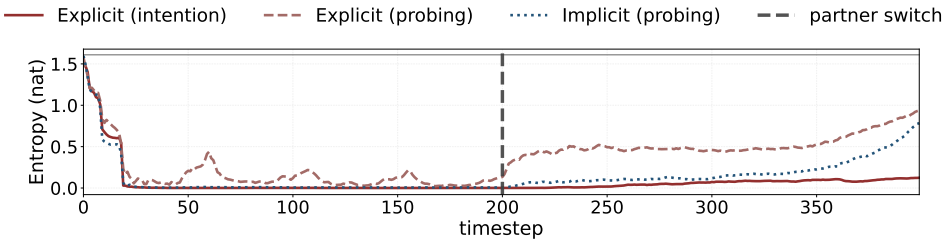
Implicit agents shift toward mental-state tracking under the urgency signal



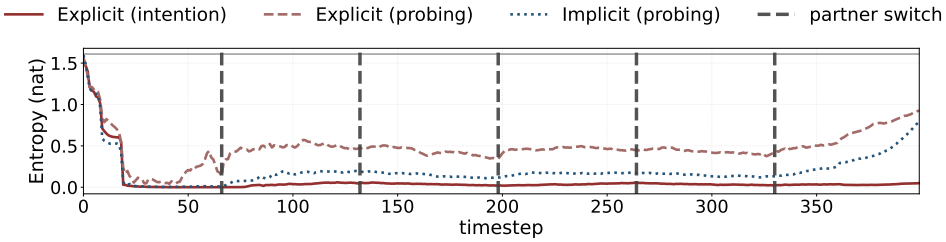
Supplementary Fig. S6. Prediction accuracy of the implicit probing classifier with and without the urgency signal. Episodes run for 300 timesteps with three evenly spaced partner switches; the urgency signal activates at timestep 260 or is kept off. With the signal, the implicit agent shifts toward mental-state modeling, following the current partner more readily.

We test whether the urgency signal drives the implicit agent's shift toward mental-state tracking. Episodes run for 300 timesteps with three partner switches evenly spaced across them. The urgency channel in the agent's observation is either activated from timestep 260 onward—40 steps remaining, matching the standard rule—or kept off throughout. As shown in Supplementary Fig. S6, without the signal the absorption of the switched partner remains low gradual, whereas with the signal the accuracy on the current partner rises steeply after onset—a shift toward mental-state modeling gated by the task signal.

Prediction confidence



(a) Single partner switch at timestep 200

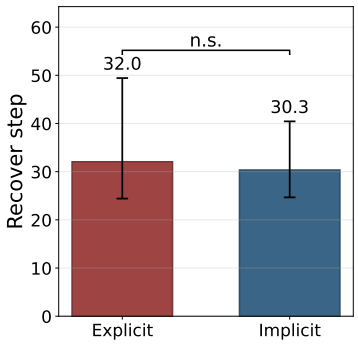


(b) Five partner switches per episode

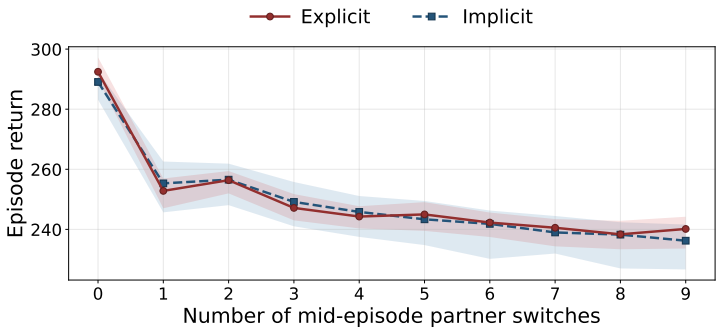
Supplementary Fig. S7. Prediction confidence of the implicit agent, the explicit agent's latent, and the intention classifier, measured as prediction entropy. The intention classifier is overconfident even when its prediction is wrong; the explicit agent's latent predicts with low confidence (high entropy); the implicit agent lies in between.

As reported in Subsection 4.1 of main manuscript, the intention classifier of the explicit agent, which maintains the character, remains highly confident even when its prediction is wrong; the probing head on the explicit agent's latent predicts with relatively low confidence; and the implicit agent lies in between. We measure confidence as the entropy of the prediction (Supplementary Fig. S7).

Performance comparison



(a) Steps to first delivery after a switch

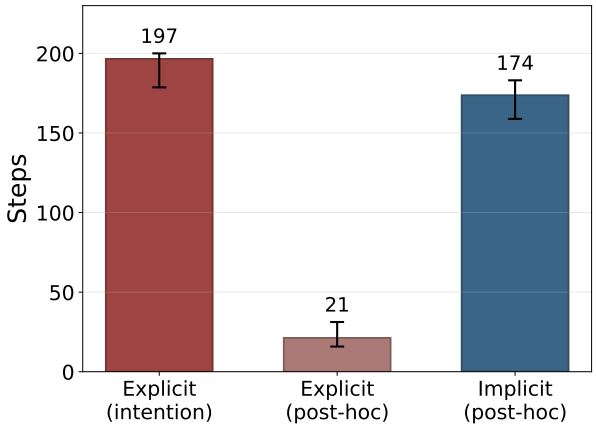


(b) Return across increasing numbers of switches

Supplementary Fig. S8. Performance under changing partner behavior. (a) Steps to the first delivery after a single switch, which occurs at the first delivery completion after timestep 200. (b) Return as the number of switches per episode increases, with per-partner durations sampled from a Dirichlet distribution ($\alpha = 2.0$). IQM with 95% bootstrap CIs over ego agents. Explicit modeling does not change performance under partner behavior change.

We measure performance under the behavior-change setting of Fig. 1 of main manuscript in two ways, each with a slight protocol difference. First, we measure the number of steps to the first delivery after a single switch; here the partner switches at the first delivery completion after timestep 200, rather than exactly at timestep 200, so that recovery is measured from a common reference point. Second, we measure total return while varying the number of switches per episode; this protocol is similar to the five-switch setting but injects stochasticity into switch timing, sampling each partner's duration from a Dirichlet distribution with uniform concentration $\alpha = 2.0$. As reported in Subsection 4.1 of main manuscript, explicit modeling yields no return benefit under behavior change: after a switch, the intention classifier locks onto a wrong prediction, even though the explicit agent's own latent tracks the change (Supplementary Fig. S8).

Mean adaptation steps



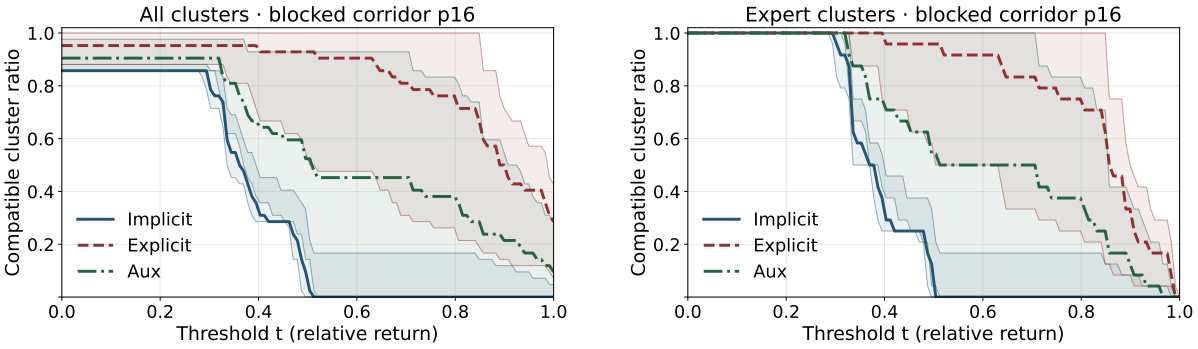
Supplementary Fig. S9. Mean adaptation steps after a single partner switch: the number of timesteps until the predicted cluster first matches the switched partner. IQM with 95% bootstrap CIs over ego agents. The explicit agent's latent re-identifies the partner substantially faster than the implicit agent and the intention classifier.

We measure the mean number of adaptation steps after a single switch: the number of timesteps until each predictor's cluster prediction first matches the switched partner. The probing head on the explicit agent's latent re-identifies the switched partner substantially faster than both the implicit agent and the intention classifier (Supplementary Fig. S9). Note the contrast with Supplementary Fig. S8(a): faster representational re-identification does not translate into faster return recovery.

Intention modeling as an auxiliary task

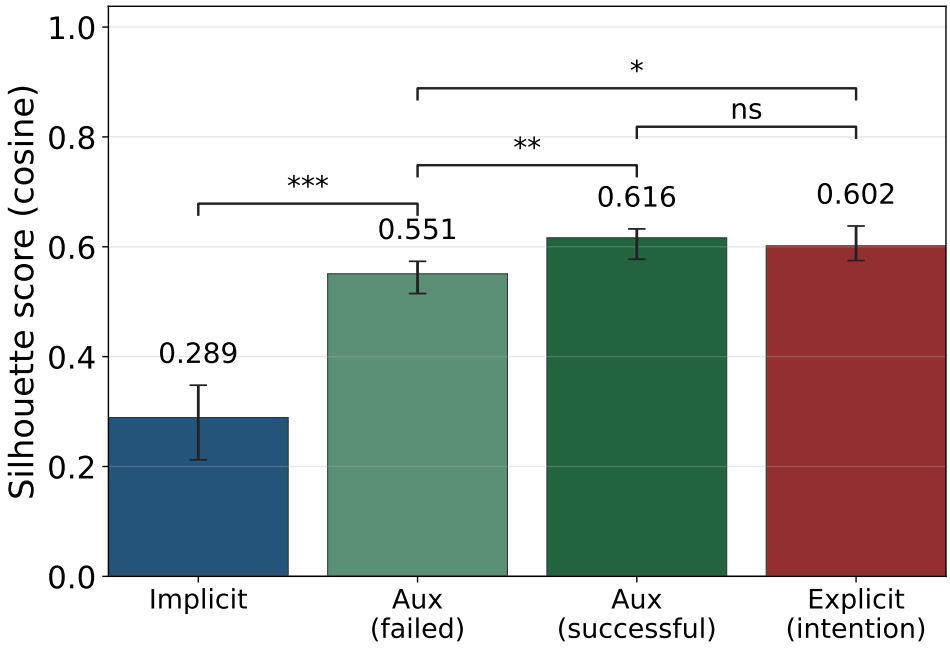
As mentioned in Subsection 4.2 of main manuscript, we train auxiliary agents whose objective additionally predicts the partner's behavioral cluster and next action from the ego agent's own latent, with gradients flowing into the CNN-LSTM encoder; apart from this, training is identical to the implicit agent, and unlike the intention module, the auxiliary heads receive no warm-start phase. Throughout this section, cluster predictions are read from the auxiliary head itself rather than from a separately trained probe.

Character separability and cooperative incompatibility



(a) All cluster levels **(b)** Expert clusters only

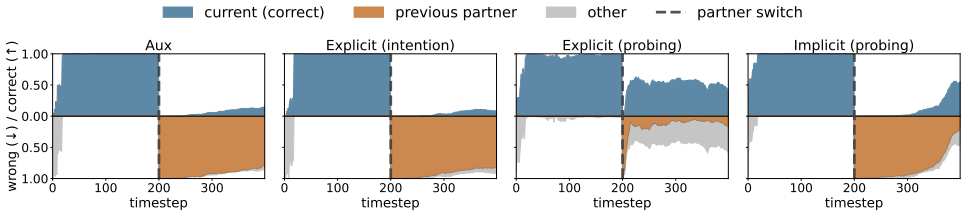
Supplementary Fig. S10. Counterpart of Fig. 3 of main manuscript with auxiliary task. Auxiliary task tighten the gap between explicit and implicit agent. Separable character modeling is important factor to mitigate cooperative incompatibility.



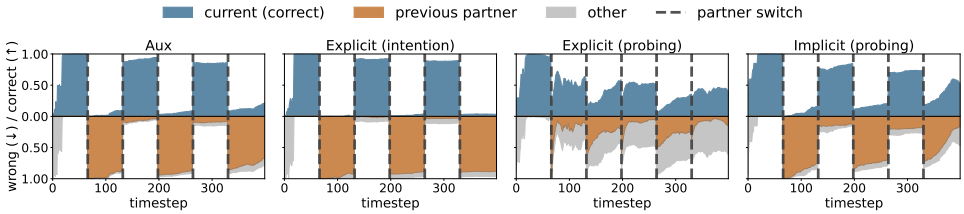
Supplementary Fig. S11. Counterpart of Fig. 4 of main manuscript with auxiliary task. Bars, left to right: Implicit n = 10, Aux (failed) n = 5, Aux (successful) n = 5, Explicit n = 10. Brackets: Implicit vs Aux (failed) p ¡ 0.0005; Aux (failed) vs Aux (successful) p = 0.006; Aux (successful) vs Explicit p = 0.691; Aux (failed) vs Explicit p = 0.01. The character is substantially separable than implicit agents but, there are still gap between failed and not failed agents.

We train 10 ego agents at Blocked Corridor with a population of 16 only for simplicity, since it is the most distinct setting of incompatibility between explicit and implicit agent. We find that it tighten the gap between explicit and implicit agents. Also, there are still gap of silhouette score between failed and not failed agents.

The division of labor with auxiliary task

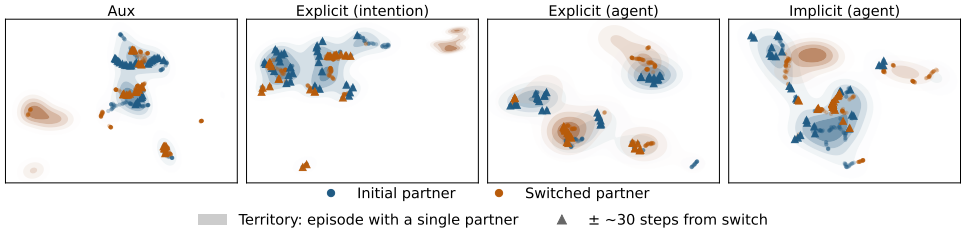


(a) Single partner switch at timestep 200



(b) Five partner switches per episode

Supplementary Fig. S12. Counterpart of Fig. 1 of main manuscript with a auxiliary task. Auxiliary task agent does not follow mental state during changing behaviors.



Supplementary Fig. S13. Counterpart of Fig. 2 of main manuscript with auxiliary task. Latent does not follow changed partner.

The following question could be the behavior about division of labor like Subsection 4.1 of main manuscript. We find that mental state tracking does not emerge in auxiliary agent.

Transformer configuration and results

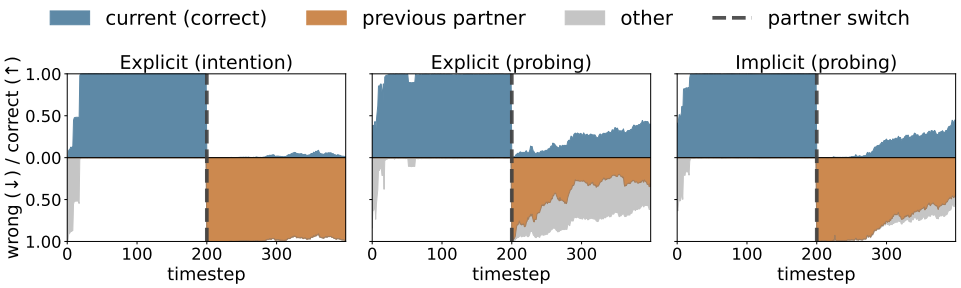
Architecture configuration

Component	Value
Activation	GeLU
Normalization	Pre-layer norm
Positional embedding	Learnable
Context length	400 (full episode)
# of layers	2
# of heads	4
Embedding dimension	128
MLP hidden dimension	4× embedding

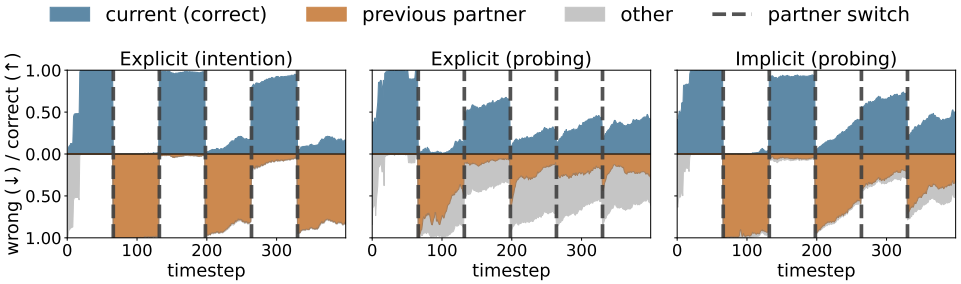
Supplementary Table S3. Architecture configuration for the Transformer backbone.

For value bootstrapping at the terminal step, the input at position 400 (with positions indexed from 0) reuses the positional embedding of position 399 (Supplementary Table S3).

Results on changing behavior



(a) Single partner switch at timestep 200

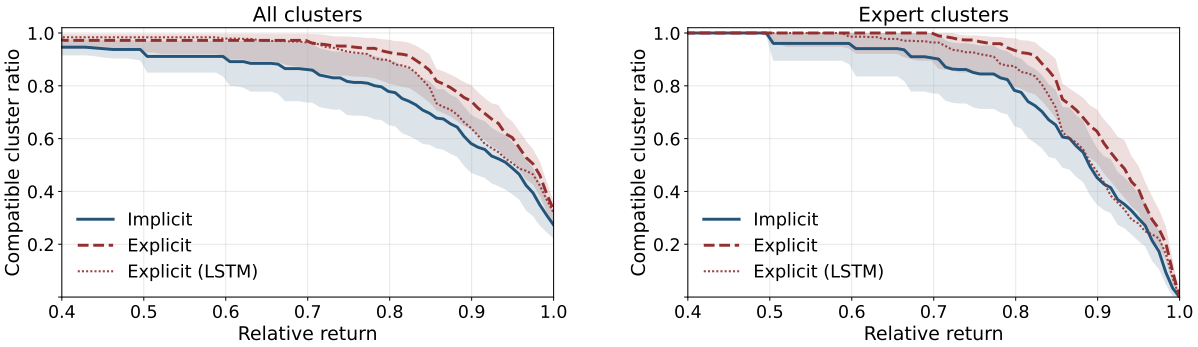


(b) Five partner switches per episode

Supplementary Fig. S14. Counterpart of Fig. 1 of main manuscript with a Transformer backbone. The Machine ToM structure persists, with mental-state tracking redistributed: the classifier still maintains the character, the Transformer-explicit agent tracks the mental state more slowly than its LSTM counterpart, and the Transformer-implicit agent tracks faster than the LSTM-implicit one.

The division of labor persists under the Transformer backbone, with mental-state tracking redistributed: the Transformer-explicit agent tracks a switch more slowly than its LSTM counterpart, while the Transformer-implicit agent tracks faster than the LSTM-implicit one. The results are shown in Supplementary Fig. S14.

Results on character modeling and cooperative incompatibility



(a) All cluster levels

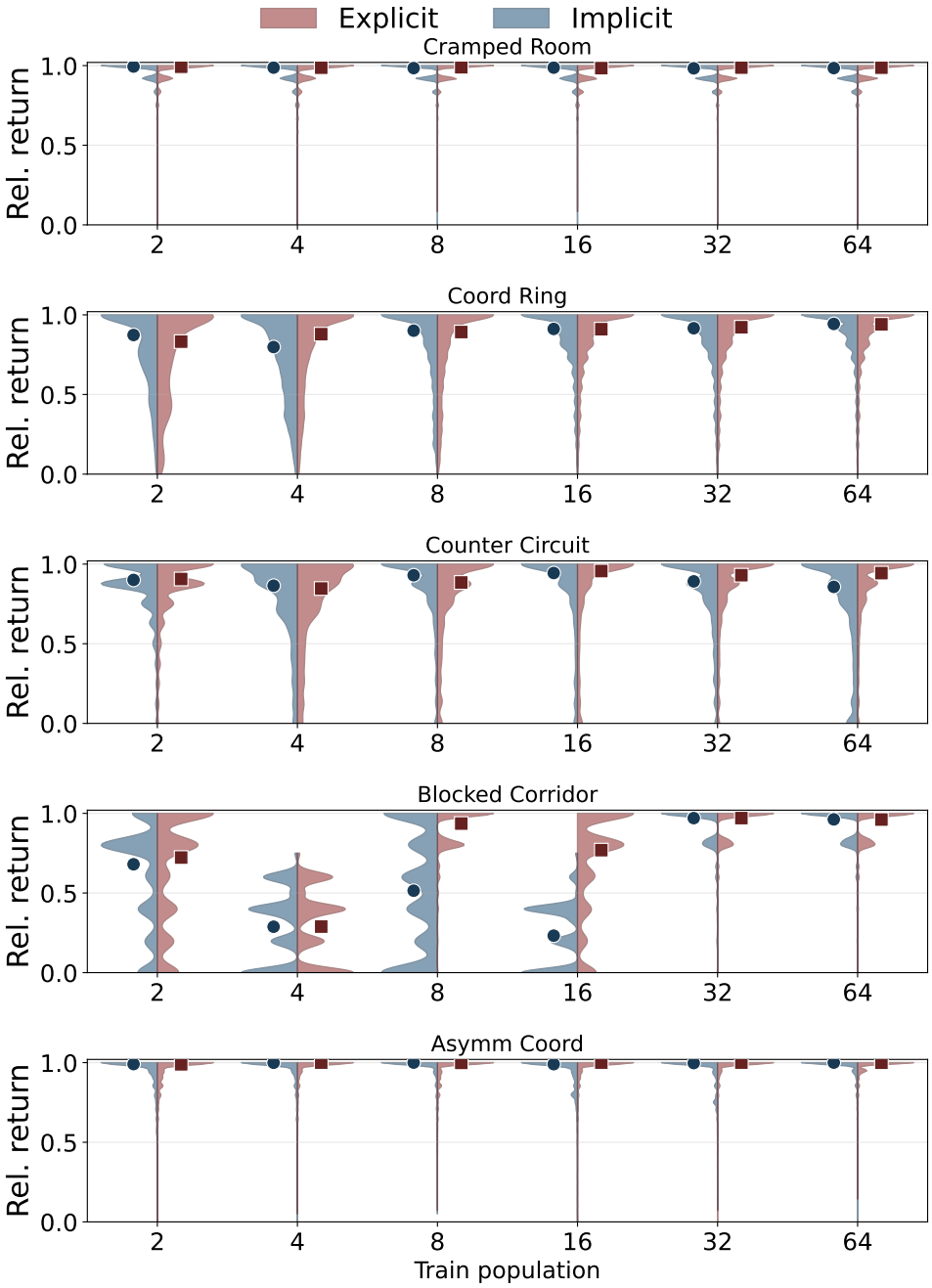
(b) Expert clusters only

Supplementary Fig. S15. Counterpart of Fig. 3 of main manuscript with a Transformer backbone. The backbone narrows, but does not close, the explicit–implicit gap: the Transformer-implicit agent nearly reaches the LSTM-explicit agent, while the Transformer-explicit agent remains the most robust.

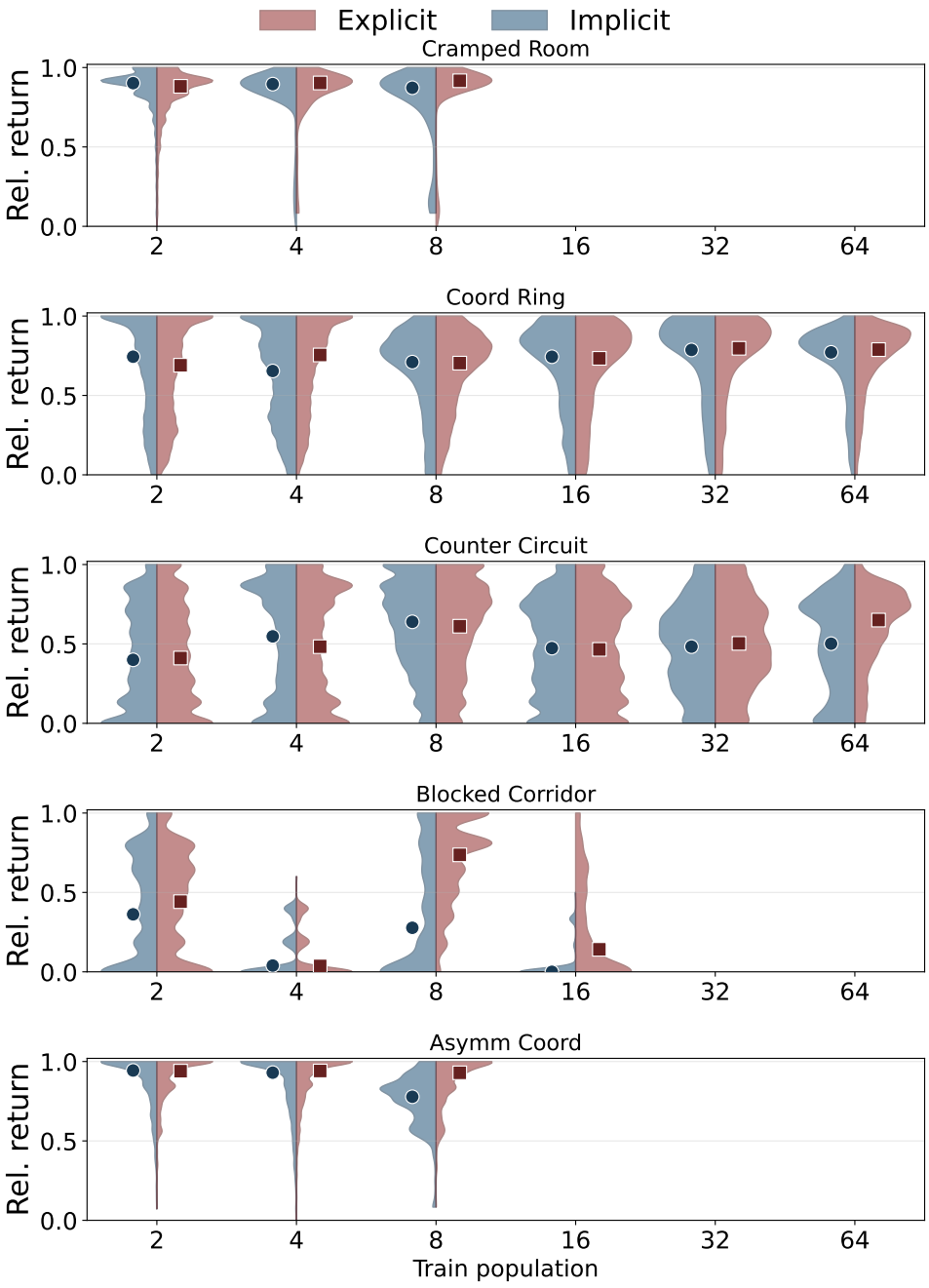
A stronger backbone mitigates cooperative incompatibility on its own, and explicit modeling adds a further margin on top: the two are complementary. Consistent with Subsection 4.4 of main manuscript, the backbone's contribution is not sharper character separability but a reduction of the failures that occur despite separable character modeling.

Additional results on explicit and implicit agents

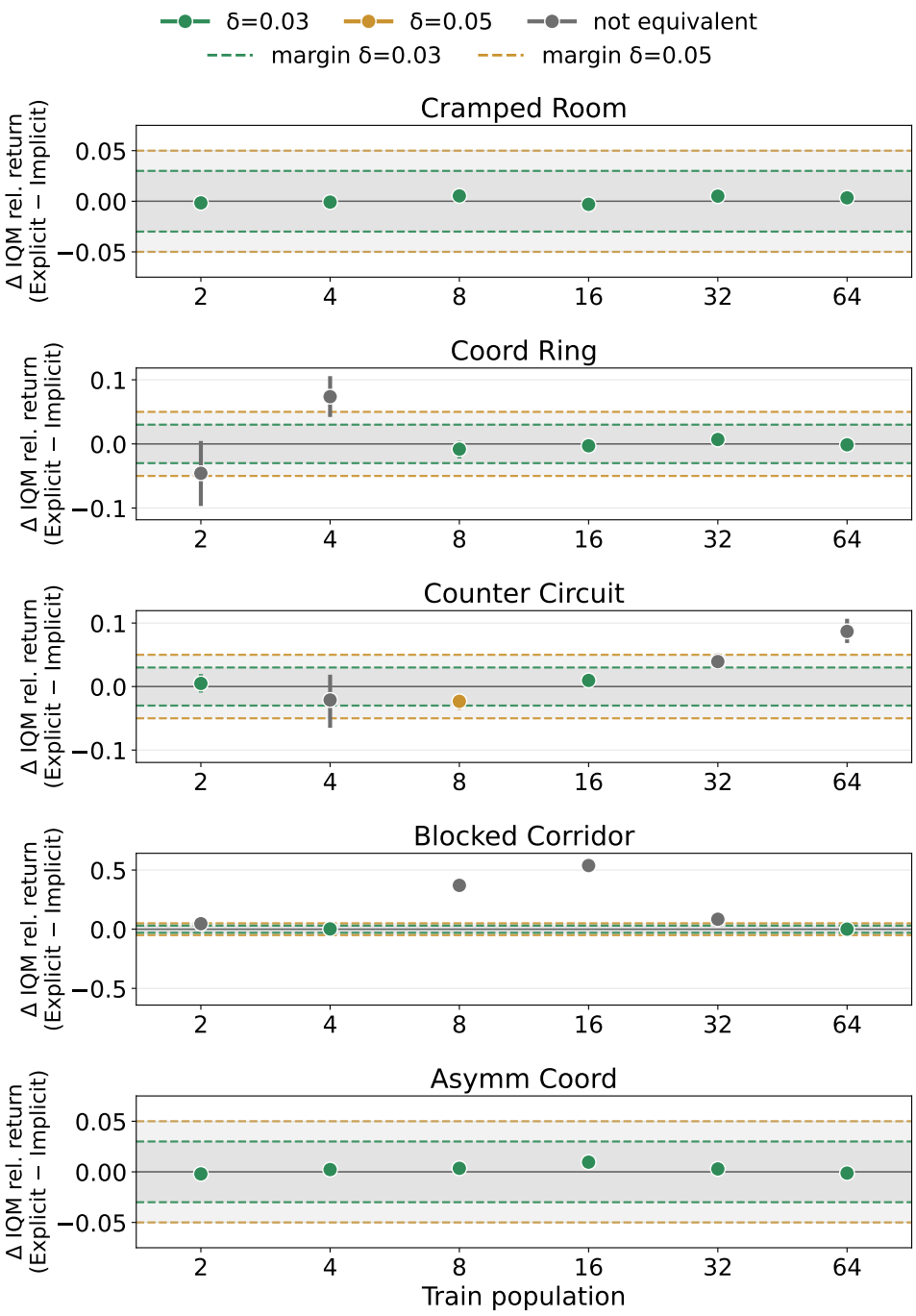
Full layout–population combination results



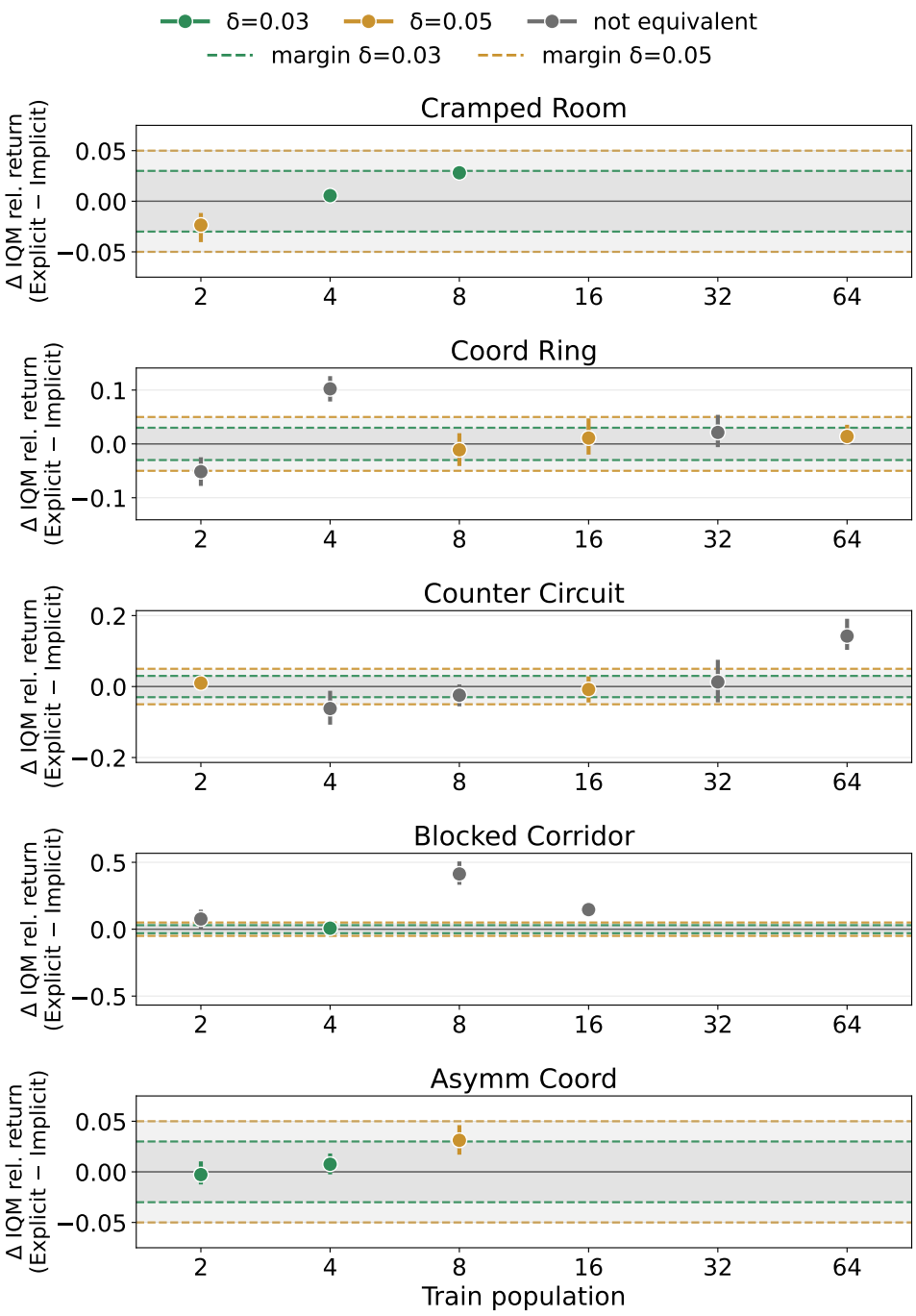
(a) Merged group



Supplementary Fig. S16. General cooperation performance of explicit and implicit agents across all layout–population combinations. Absent bars in the novel group indicate that all test partners fall into the merged group for that setting. All data are measured over 160 independent episodes per test partner, for each of three ego-agent seeds per variant. The results mirror the findings of Fig. 7 of main manuscript.

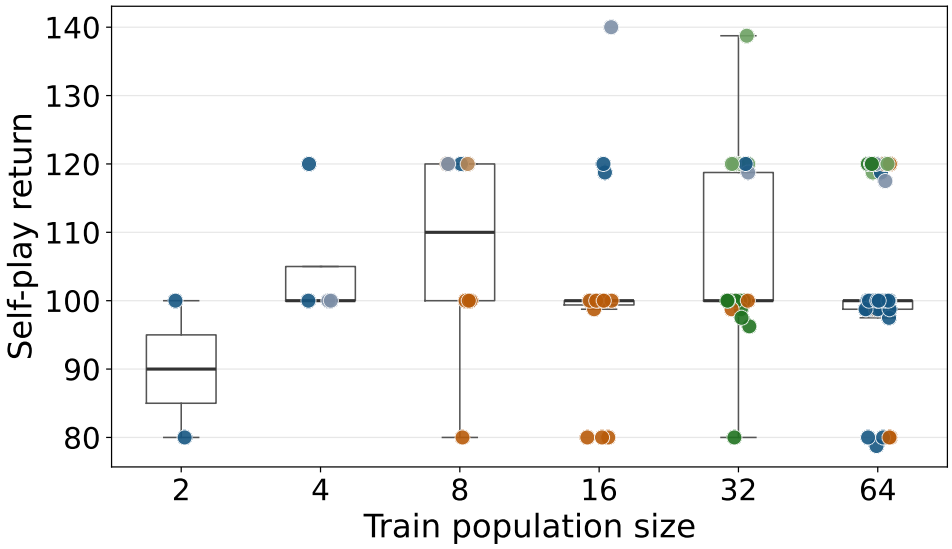


(a) Merged group



(b) Novel group

Supplementary Fig. S17. Equivalence analysis (TOST) for Supplementary Fig. S16. Each point is the explicit−implicit difference in IQM relative return, with 90% stratified bootstrap CIs (resampling test partners, with ego seeds as strata); the inference is thus over the partner distribution given the trained agents. a setting is equivalent at margin δ if the CI lies entirely within $\pm\delta$. The primary margin is $\delta = 0.05$, well below the effect sizes observed in the incompatibility settings; the stricter $\delta = 0.03$ is shown for reference. In the merged group, most settings outside the cooperative-incompatibility regimes are equivalent, many at the strict margin, with a few exceptions at the smallest populations. In the novel group, deviations are more common; across both groups, the exceptions go in both directions with no consistent sign.

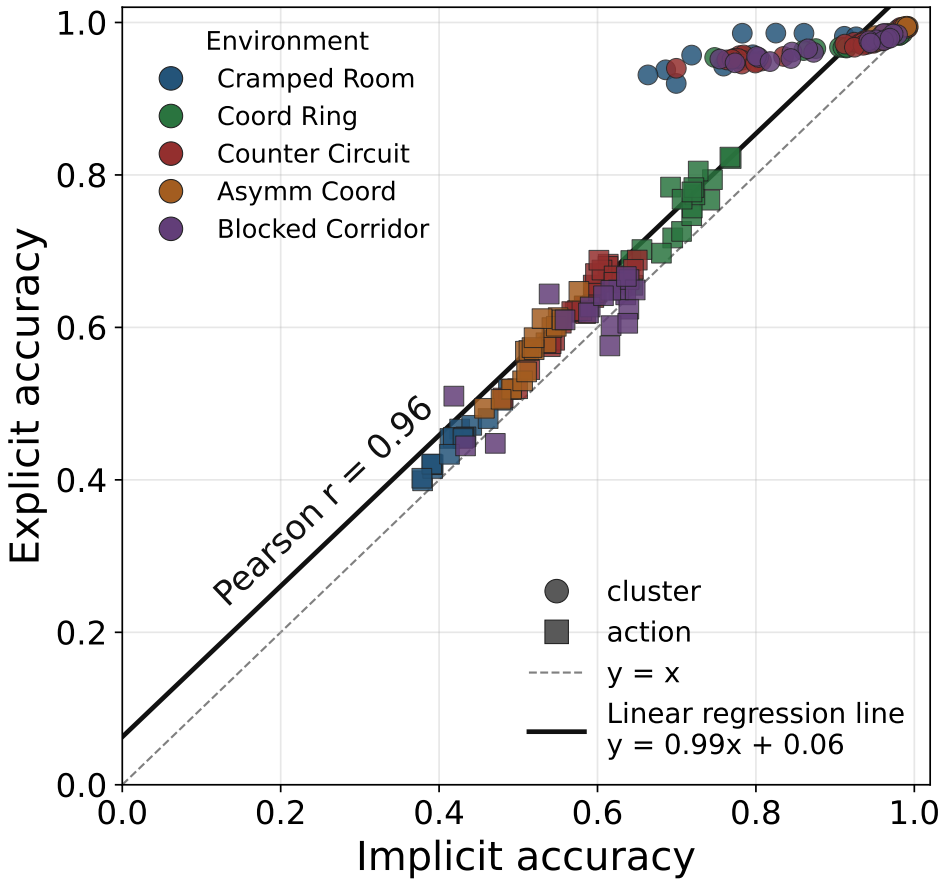


Supplementary Fig. S18. Self-play returns of the training populations for Blocked Corridor. Colors indicate the behavioral clusters within each population. The populations of 4 and 16 each contain a single upper outlier.

The pattern that explicit intention modeling does not change performance outside the incompatibility settings persists in the full results (Supplementary Fig. S16). The equivalence analysis in Supplementary Fig. S17 quantifies this with a caveat: merged-group differences fall within the equivalence margin outside the incompatibility regimes, whereas novel-group differences exceed it more often—yet in both directions and with no consistent sign, indicating seed- and partner-level variability rather than a systematic advantage of either variant.

Ego agents fail especially often on Blocked Corridor at populations of 4 and 16. This is consistent with ego agents specializing toward a single partner: both populations contain a single upper outlier in self-play return (Supplementary Fig. S18), and no ego agent exhibits a failure case against the outlier partner.

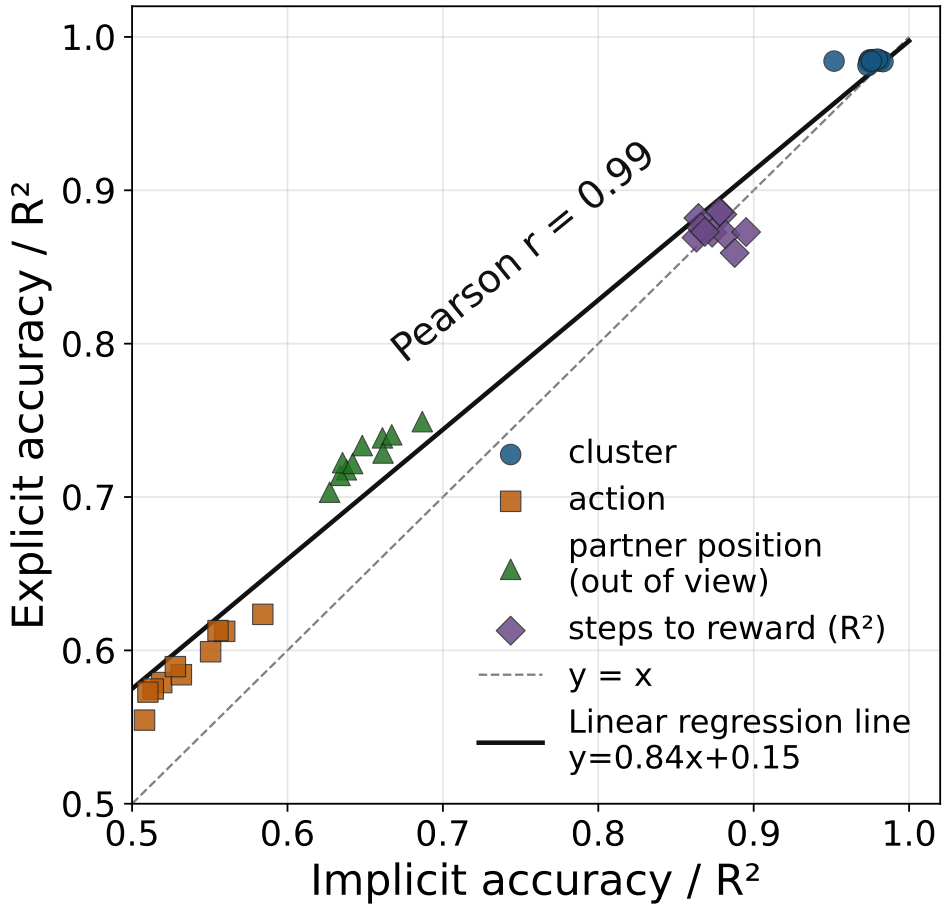
Additional results on prediction accuracy



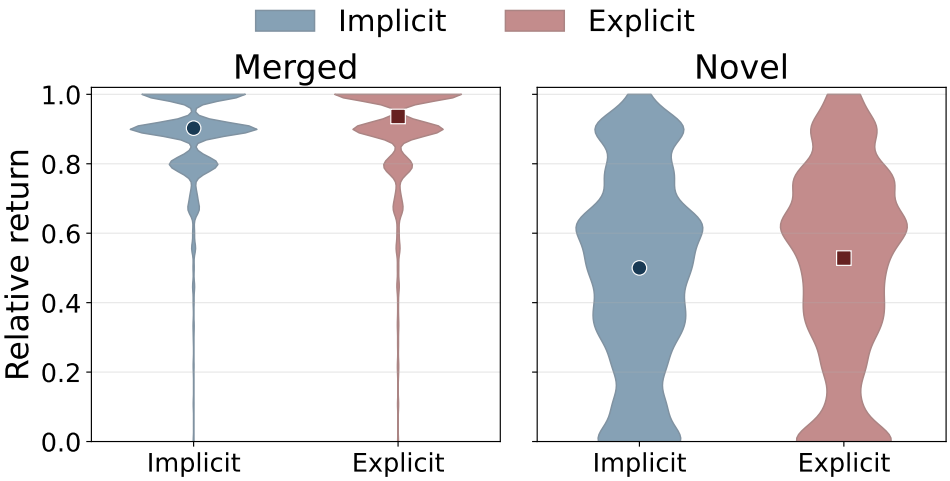
Supplementary Fig. S19. Comparison between the intention classifier of explicit agents and the probing classifier of implicit agents across all layout–population combinations. Implicit accuracy nearly matches the explicit one wherever partner information is required for coordination, and drops where it is not (e.g., Cramped Room).

We measure prediction accuracy as in Fig. 8 of main manuscript for all layout–population combinations. Implicit accuracy nearly matches the intention classifier wherever coordination requires partner information; it drops in layouts that do not require it, such as Cramped Room—consistent with Mon-Williams et al. [12], who argue that implicit partner modeling emerges only when it matters. Full results are shown in Supplementary Fig. S19.

Results on the partially observable setting



(a) Prediction accuracy comparison



(b) Coordination performance comparison

Supplementary Fig. S20. Prediction accuracy and coordination performance of implicit and explicit agents under partial observability. (a) is the counterpart of Fig. 8 of main manuscript with partner-position prediction added; (b) is the counterpart of Fig. 7 of main manuscript. Explicit modeling yields small accuracy gains (a) on partner position, and a marginal performance gain in (b).

We additionally conduct an experiment under partial observability on Counter Circuit with a population of eight, using the Overcooked V2 environment [13]. In this setting, we add the partner's current position as an additional prediction target for intention modeling, inspired by Wang et al. [14], who use the teammate's observation as a reconstruction target. We train 10 ego agents per variant from different random seeds. As the test pool, we use a separate set of 32 partners trained with the standard MEP objective alone, without the cross-population term used in Supplementary Method–"Building test population". Explicit modeling adds small accuracy gains on partner position (when the partner is outside the view radius). A coordination-performance gain also exists, but it is marginal (Supplementary Fig. S20). In sum, intention modeling helps when information is restricted, yet the effect is marginal; how this effect scales with the amount of restricted information is an interesting direction for future work.

References

[1] 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.

[2] Wang, X. *et al. ZSC-Eval: An evaluation toolkit and benchmark for multi-agent zero-shot coordination. Advances in Neural Information Processing Systems (NeurIPS)* (2024). URL https://proceedings.neurips.cc/paper_files/paper/2024/file/54a7139c548c88e288aa0fcd2bcbeceb-Paper-Datasets_and_Benchmarks_Track.pdf.

[3] 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.

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

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

[6] Schulman, J., Moritz, P., Levine, S., Jordan, M. & Abbeel, P. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438* (2015). URL https://arxiv.org/abs/1506.02438.

[7] Kingma, D. & Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014). URL https://arxiv.org/pdf/1412.6980.

[8] Bradbury, J. *et al.* JAX: composable transformations of Python+NumPy programs, 0.3.13 (2018). URL http://github.com/jax-ml/jax.

[9] Heek, J. *et al.* Flax: A neural network library and ecosystem for JAX, 0.10.4 (2024). URL http://github.com/google/flax.

[10] 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.

[11] McKee, K., Leibo, J., Beattie, C. & Everett, R. *Quantifying the effects of environment and population diversity in multi-agent reinforcement learning. International Conference on Autonomous Agents and Multiagent Systems (AAMAS)* (2022). URL https://link.springer.com/article/10.1007/s10458-022-09548-8.

[12] 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.

[13] Gessler, T. *et al. OvercookedV2: Rethinking overcooked for zero-shot coordination. International Conference on Learning Representations (ICLR)* (2025). URL https://openreview.net/forum?id=hlvLM3GX8R.

[14] 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.