Adaptive Opponent Modeling for Adversarial Co-Training in MARL

A controlled predator–prey study: inferring a varying opponent's latent strategy and planning against it

Rajdeep Singh
REL Lab, University of Southern California

May 2026

Abstract

In adversarial co-training each team faces an opponent whose strategy varies and adapts; a best response that overfits to one opponent loses robustness to strategies it has not recently seen. The remedy we pursue is an *uncertainty-aware* model of the opponent's latent strategy (Part 1), *sampled inside a model-based planner* (Part 2). This note instantiates and validates both parts in a controlled predator–prey game. The opponent—the prey, team red—draws a hidden strategy each episode; the controlled team—the predators, team blue—cannot observe it and must infer it from red's behavior, then plan. We instantiate the latent strategy z as a discrete intent so inference can be scored against ground truth. We find: **(i)** the opponent's latent strategy is decisive—a blue policy that observes z does +51%; **(ii)** it is inferable from red's trajectory with calibrated uncertainty—an encoder $q_\phi(z \mid \tau^{\text{red}})$ recovers it with accuracy $0.37 \to 0.97$ over $3 \to 25$ observed steps as its posterior entropy collapses; **(iii)** a *point-estimate* opponent model is brittle—fed a wrong guess it loses 47%, exactly the failure uncertainty-aware modeling is meant to prevent; and **(iv)** sampling the uncertainty-aware opponent model inside a planner yields the most robust best response, +61% over an opponent-blind baseline and *above an oracle* blue policy told the true strategy. An ablation that flattens the belief shows the inference, not the lookahead, is the active ingredient (+40%). Finally, we replace the supervised encoder with a self-supervised *predictive* one (JEPA): it recovers the opponent's strategy far better than a generative VAE (0.89 vs. 0.53 probe; 0.65 vs. 0.14 ARI) and from about half the observation, and—driving the planner with a *label-free* belief—it matches the supervised pipeline (4.08 vs. 4.31 captures) with no opponent-strategy labels at all. The study validates the proposal's Part 1/Part 2 loop on its first target domain and shows a predictive, label-free encoder is the stronger Part 1.

1 Background: adaptive opponent modeling for co-training

Adversarial co-training—training both teams together rather than scripting one—is necessary for robust strategies in complex games, but it is hard: each team faces a *non-stationary* opponent, policies can fall into cyclic (rock–paper–scissors) dynamics, and a team readily *overfits to the current opponent*, losing to strategies that have not appeared recently. Existing approaches underserve exactly the opponent that varies at test time: league play is brute force and brittle to novel opponents; CTDE methods (e.g. MADDPG) assume centralized access to the opponent and model *no* uncertainty in its strategy; model-based opponent modeling assumes known goals or clones behavior to a moving target.

The approach. Model the opponent's strategy with *calibrated uncertainty* and *plan* against that model. Two coupled parts:

Part 1 (uncertainty-aware opponent model).
A variational encoder $q_\phi(z \mid \tau^{\text{red}})$ maps the opponent's (red's) trajectory to a latent strategy z with a posterior that expresses how sure we are; a latent-conditioned opponent policy $\pi^{\text{red}}(a \mid s, z)$ reconstructs red's behavior (by behavior cloning or contrastive preference learning). The model must stay robust to opponents that adapt and use varying strategies.

Part 2 (plan against it).
The controlled team (blue) plans with a model-based search that *samples* the opponent's moves from $\pi^{\text{red}}(\cdot \mid s, z)$ and conditions its value $v^{\text{blue}}(s, z)$ on the inferred strategy, so the plan is aware of which opponent it currently faces.

This study. We validate the loop in the proposal's first domain, predator–prey, in a form where every quantity can be measured. The opponent (prey, red) is assigned a hidden *intent* each episode—one of four corners it is rewarded for haunting—which the controlled predators (blue) cannot observe. The intent is the latent strategy z, made discrete so the encoder's belief can be checked against truth and so "the opponent uses varying strategies" has an exact meaning: a fresh z every episode. The environment is otherwise fixed, so z is *not* readable from a single observation and can only be had by modeling red's behavior over time.

2 Setup

Three blue predators P pursue one red prey q in a fixed 2D arena (JaxMARL MPE, five discrete actions, 25-step episodes, two fixed obstacles). At reset the prey draws $z \sim \text{Uniform}\{0, 1, 2, 3\}$ indexing a corner $c_z \in \{\pm 0.8\}^2$. With $k_i^t = \mathbf{1}[\|x_i^t - x_q^t\| < \rho_i + \rho_q]$ marking predator i in contact with the prey,

$$r^{\text{blue},t} = \lambda_{\text{tag}} \sum_{i \in P} k_i^t, \qquad r^{\text{red},t} = -\lambda_{\text{tag}} \sum_{i \in P} k_i^t + \lambda_{\text{int}} \, \mathbf{1}\big[\|x_q^t - c_z\| < r_{\text{int}}\big], \tag{1}$$

λ_{tag}=10, λ_{int}=4, r_{int}=0.35: the prey loiters at its assigned corner when safe while evading. The prey observes its own z; *the predators do not.* Each policy is a MAPPO co-training run (centralized critic, 3 seeds); we report captures per episode, cap $= \lambda_{\text{tag}}^{-1} \sum_t r^{\text{blue},t}$, over 300 episodes.

3 Method (the controlled instantiation)

Part 1 — an uncertainty-aware model of the opponent's strategy. From the prey's first k positions $\tau_{0:k}^{\text{red}}$ we fit $q_\phi(z \mid \tau_{0:k}^{\text{red}})$, here a classifier whose softmax is a posterior belief $\mathbf{b}_k$ over the four latent strategies. The belief, not a point estimate, is the object we carry: it sharpens online as more of the opponent is observed, which is the discrete instance of the proposal's variational posterior over z. The opponent policy model is $\pi^{\text{red}}(a \mid s, z)$, the prey's own behavior under strategy z (the target a latent-conditioned BC/CPL model is trained to reproduce).

Part 2 — planning against the uncertainty-aware model. At state s_t with belief $\mathbf{b}_t$, blue scores each candidate joint action u by a short rollout in which red's moves are *sampled from the opponent model under a strategy drawn from the belief*:

$$Q(s_t, u) = \mathbb{E}_{z\sim\mathbf{b}_t}\left[\sum_{h=0}^{H-1}\gamma^h r^{\text{blue}}_{t+h} \;-\; w\,\min_{i\in P}\|x_i^{t+H} - x_q^{t+H}\| \;\middle|\; u_t{=}u,\; a^q_{t+h}\sim\pi^{\text{red}}(\cdot\mid s_{t+h}, z)\right], \qquad (2)$$

and acts $u_t = \arg\max_u Q(s_t, u)$ (H=5, γ=0.95, K=8 rollouts, leaf weight w=1, strategies shared across candidates). Marginalizing over $\mathbf{b}_t$ makes the plan *uncertainty-aware*: while the belief is flat blue hedges, and as it sharpens blue commits. The leaf term rewards ending near the imagined future prey, which under a sampled strategy sits at the believed corner—an interception. This is the simulator-based instance of the proposal's tree search that samples $\pi^{\text{red}}(\cdot \mid z)$ at each node and scores with a strategy-aware value $v^{\text{blue}}(s, z)$.

4 Results

(i) The opponent's latent strategy is inferable (Table 1, Figure 1). The prey's motion clusters by strategy (Figure 1, left), and the encoder recovers z with accuracy 0.37→0.97 over 3→25 steps as its posterior entropy collapses from 1.35 to 0.03 nats—a usable, increasingly-confident belief well before the episode ends.

Table 1: Recovery of the opponent's latent strategy from its first k steps (4-way; chance 0.25).

steps k	3	5	8	12	18	25
accuracy	0.37	0.70	0.74	0.84	0.93	0.97
posterior entropy (nats)	1.35	0.75	0.52	0.31	0.09	0.03

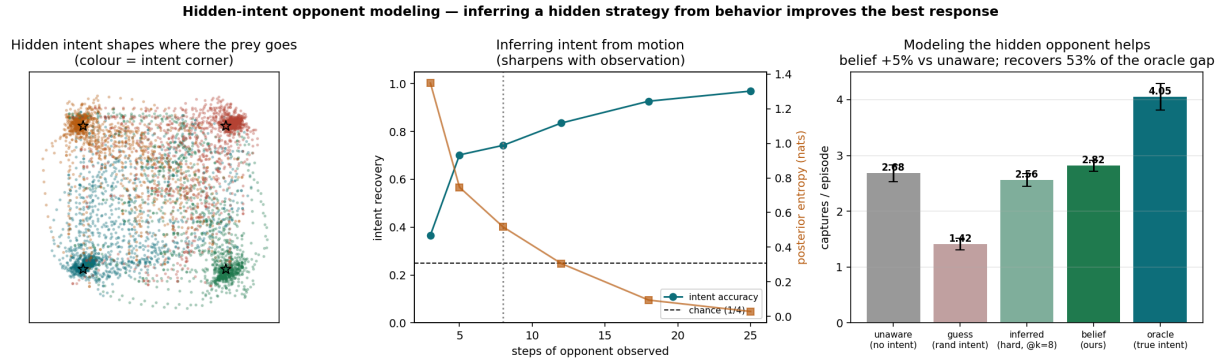

Figure 1: **Left:** red occupancy coloured by its hidden latent strategy—four clusters in one fixed arena. **Middle:** the encoder's accuracy rises and posterior entropy falls with steps of the opponent observed. **Right:** blue captures per episode across the opponent-model signals it is given.

(ii) The strategy is decisive, and a point estimate is brittle. A blue policy that observes the true z catches the prey 4.05 times/episode vs. 2.68 for an opponent-blind baseline (+51%): knowing which opponent one faces matters a great deal. But a blue policy trained on *perfect* z and

fed a *wrong* one collapses to 1.42 (−47%)—the precise brittleness of opponent models that ignore their own uncertainty, which the uncertainty-aware belief exists to prevent.

(iii) Planning against the uncertainty-aware model is the most robust response (Table 2, Figure 2). The planner of Eq. (2) reaches **4.31** captures: +61% over the opponent-blind baseline, +53% over a reactive policy given the same belief, and *above the oracle reactive policy* (4.05)—planning interceptions against the inferred opponent beats reacting even with the true strategy in hand. The control isolates the cause: a planner given a *flat* belief (lookahead, no opponent inference) reaches only 3.07, so the inferred opponent model supplies +40%. The opponent inference, not the lookahead, is what wins.

Table 2: Blue captures per episode against the varying red opponent (mean ± std, 3 seeds × 300 episodes).

blue predator	captures / ep	vs. opponent-blind
opponent-blind (no strategy info)	2.68 ± 0.15	—
reactive, hard-inferred intent (k=8)	2.56 ± 0.11	−5%
reactive, belief (uncertainty-aware)	2.82 ± 0.10	+5%
planner, flat belief (ablation)	3.07 ± 0.17	+15%
oracle, true strategy (reactive)	4.05 ± 0.24	+51%
planner, inferred belief (Part 1+2)	$\mathbf{4.31 \pm 0.36}$	**+61%**

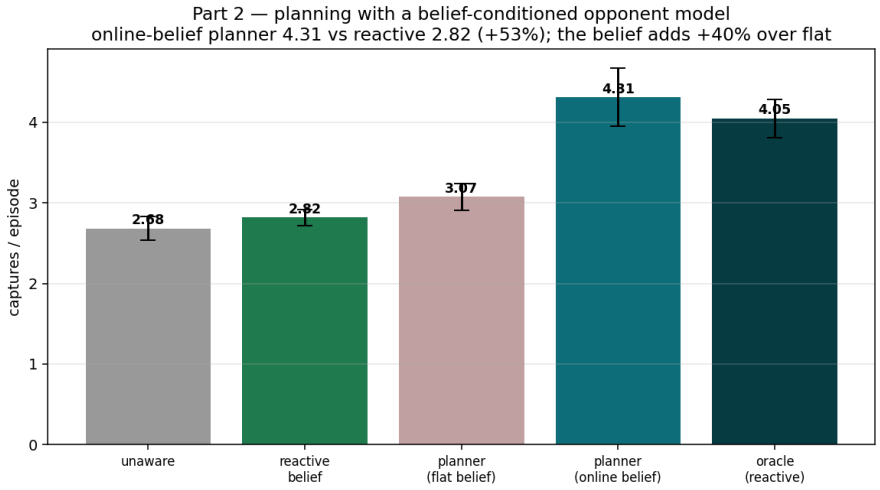

Figure 2: Planning against the uncertainty-aware opponent model (online belief) tops every reactive policy, including the oracle. Ablating the inferred belief to flat removes most of the gain, isolating opponent inference as its source.

5 Part 1 without labels: predict, don't reconstruct

The encoder above used intent labels. A Part 1 that scales should recover the opponent's strategy *self-supervised*, from trajectories alone—and the *objective* matters. We compare two self-supervised

encoders of the prey trajectory on identical data, both bottlenecked to a 2-D latent (so the latent *is* the figure) and scored only afterward against the four ground-truth intents:

- **VAE (generative):** encode the first 12 steps, then *reconstruct* them (ELBO). It must model every detail—the random start and the evasion noise included.
- **JEPA (predictive):** encode the first 12 steps, then *predict the representation* of the future window (steps 12–24, the corner approach) through an EMA target encoder—no reconstruction. It keeps only the predictable structure (where the prey is heading = its strategy) and discards the noise.

Table 3: Unsupervised recovery of the opponent's strategy from a 2-D latent (4-way; probe chance 0.25; mean ± std over 3 encoder seeds). The predictive encoder wins on both metrics and beats a supervised classifier trained on the raw window.

self-supervised encoder	probe acc.	GMM ARI
VAE (generative, reconstruct)	0.53 ± 0.00	0.14 ± 0.01
JEPA (predictive, predict repr.)	$\mathbf{0.89 \pm 0.01}$	$\mathbf{0.65 \pm 0.08}$
supervised classifier (raw window)	0.85	—

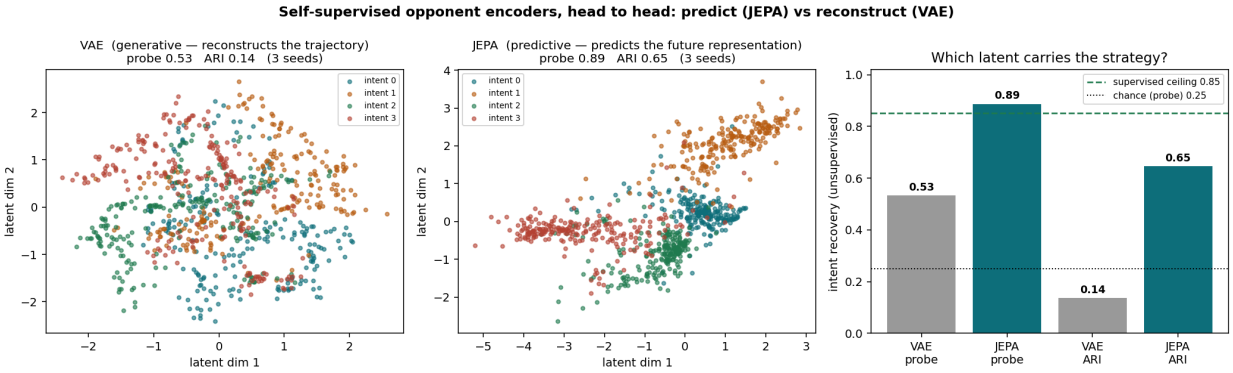

Figure 3: The generative VAE latent blends the four intents (probe 0.53); the predictive JEPA latent separates them (probe 0.89, ARI 0.65; mean of 3 encoder seeds), above the supervised ceiling. Same data, same capacity, same 2-D bottleneck—only the objective differs.

The predictive objective recovers the opponent's strategy where reconstruction cannot (Table 3, Figure 3)—and, notably, its smooth 2-D latent is *more* linearly separable than the raw window a supervised classifier sees. This is LeCun's JEPA principle (predict in representation space, not input space; cf. V-JEPA 2 [4]) applied to opponent modeling: the part of an opponent's behavior worth encoding is the part you can *predict*, and a generative encoder squanders capacity on the rest.

An anytime opponent encoder. Because it predicts where the prey is heading instead of waiting for it to arrive, the JEPA encoder also reads the strategy from *fewer* observed steps. Sweeping the observation budget k (observe the first k steps of the window, mask and predict the future), the predictive encoder dominates the generative one at every k (Figure 4): at k=11 steps it already recovers the intent at 0.80—what the VAE needs k=20 for. A planner can therefore commit roughly twice as soon.

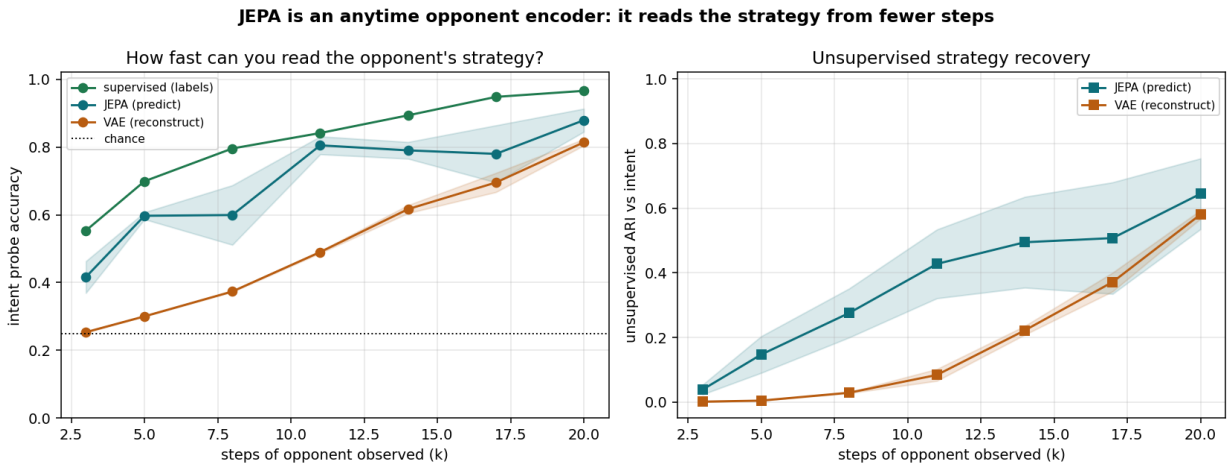

Figure 4: Intent recovery vs. steps of the opponent observed (3-seed mean ± std). The predictive (JEPA) encoder reads the strategy from about half the observation of the generative (VAE) one, tracking just under the supervised ceiling.

It closes the loop, without labels. Finally we drive the Part 2 planner with a *fully self-supervised* belief: a JEPA encoder, a readout $z \mapsto$ predicted arrival position (trained on the prey's own future positions—no intent labels), and a posterior over the four *known* corners, $b(c) \propto \exp(-\|\text{readout}(z) - c_g\|^2/\sigma)$. This drops into the planner's belief slot unchanged. With no opponent-strategy labels anywhere, it reaches $\mathbf{4.08 \pm 0.57}$ captures/episode (Figure 5)—matching the supervised-belief planner (4.31, overlapping error bars) and above the oracle reactive predator (4.05). The labels the rest of the pipeline used are not, in fact, necessary: a predictive self-supervised opponent model recovers the same best response.

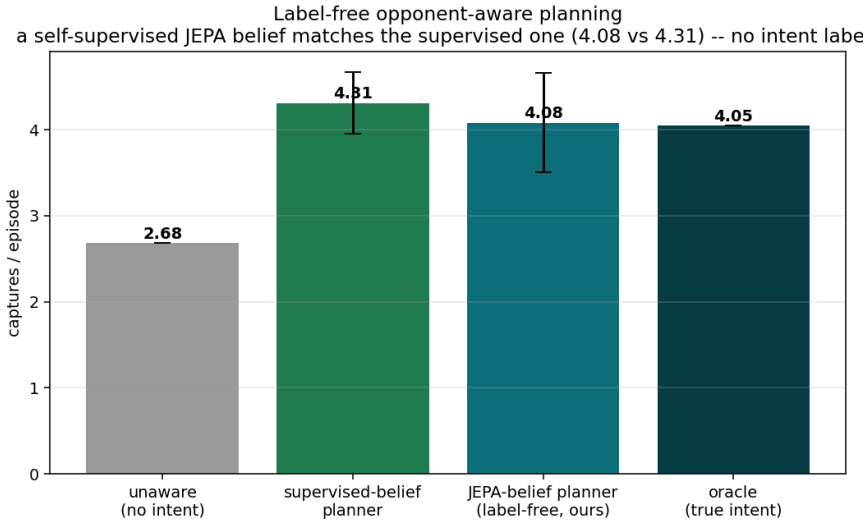

Figure 5: Label-free opponent-aware planning. A self-supervised JEPA belief drives the planner to the supervised result without any intent labels.

This is the cleaner, more general Part 1 for the proposal: no labels, a latent that recovers strategy

faster and drops straight into the belief and the planner, and a downstream best response that matches the supervised pipeline.

6 The second strategy axis: circle vs. corners

The hidden-intent task has a four-way *discrete* strategy. The lab's other axis is induced by the environment: a prey trained with resources placed on a circle (radius 0.6) routes differently from one trained with resources in the corners (∼1.13). We ask the same Part-1 question there: can an unsupervised encoder recover *which specialist* it is watching? (Specialists are rolled out in their native environments against the weakened predators they were trained with; 1,200 episodes, 3 encoder seeds throughout.)

The signal is in occupancy, not in a trajectory window. A supervised probe on a raw absolute-coordinate window of the first episode is at *chance* (0.54 at k=12–25; chance 0.5): no encoder, of any kind, could separate the placements from that input, because a single episode is dominated by evasion. The same probe on a normalized 8×8 *occupancy histogram* — *where* the prey spends its time — reaches 0.71/0.88/0.93 over 25/50/100 observed steps. The featurization, not the encoder objective, gates this task.

Predict vs. reconstruct, revisited. On occupancy features with a 2-D latent, the JEPA encoder again leads the VAE (0.63 vs. 0.48 probe at k=50; 0.67 vs. 0.66 at k=100), but with a 4-D latent the ordering *reverses* (0.81 VAE vs. 0.67 JEPA at k=100). This delineates the principle from Part 1: JEPA's advantage comes from *discarding unpredictable detail*, which is decisive when the input mixes evasion noise with signal (raw trajectories, Table 3) — but an occupancy histogram has *already* aggregated the noise away, so reconstruction stops paying a noise tax and capacity favors the generative objective. Neither objective, at any tested configuration, produces unsupervised *clusters* (GMM ARI ≈ 0, matching the earlier occupancy-VAE study): the placement is a linearly decodable *direction* in latent space, not a pair of separated modes. Closing that gap is exactly where return-relevant intent objectives [32] or equivariant/contrastive shaping enter. The predators' own occupancy is far less informative (supervised probe 0.64 at k=100; latents near chance): the *opponent* is the signal carrier.

Vanilla behaviour cloning is a faithful clone. Before asking whether a latent helps, we check the policy class. We clone the MAPPO predator from a hand-built state feature (all agent positions + a one-step velocity proxy + predator id) — strictly less than the policy network sees. Deployed against the same prey, the cloned predators recover 86% of the expert's captures-edge over random (1.22 vs. MAPPO's 1.35 captures/episode, random 0.40; 3 checkpoint seeds, two placements), at a held-out action match of 0.80 (episode-level split; Fig. 6). The BC policy class is not the bottleneck; the remaining 14% is the headroom a strategy signal could close.

Does the latent help a predator model? We predict a predator's next action from all agent positions, $\pi(a \mid s)$, versus the same input plus the unsupervised prey latent, $\pi(a \mid s, z)$ — with z computed from the prey's *first* episode and BC evaluated on the *second* (episode-level train/val split, 3 seeds; Table 4). Conditioning on the *true* placement helps clearly (0.800 ± 0.007 vs. $0.776 \pm$

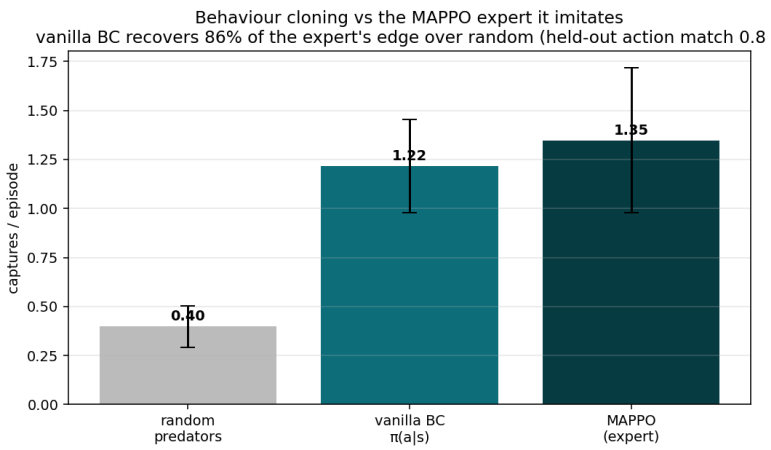

Figure 6: Deployed captures/episode for the random floor, vanilla behaviour cloning, and the MAPPO expert it imitates (3 checkpoint seeds, two placements). From a hand-built state feature, BC recovers 86% of the expert's edge over random — the policy class is faithful, so the residual gap is the headroom an opponent-strategy signal could close.

0.004 held-out action accuracy), so the hypothesized effect is real; conditioning on a *single-episode* unsupervised latent does not (0.772 for both VAE and JEPA) — exactly as the probe curve predicts, since at one episode of observation those latents carry almost no placement signal (≈ 0.51 probe). The placement is constant across a specialist's episodes, so giving the encoder *more* observation strengthens the latent and, with it, the BC gain. Reading the VAE latent from L steps before a held-out episode (steps 125–150 of a six-episode rollout), its placement probe climbs from 0.60 at L=25 to 0.78 at L=125, and latent-conditioned BC tracks it: from no gain at L=25 (0.764 vs. vanilla 0.766) it overtakes vanilla by L=50 and reaches the oracle (true-placement) ceiling by L=125 (0.795 vs. oracle 0.792; Fig. 7). The JEPA latent, whose probe stays flat ($\approx$0.6) at this 4-D latent size, gives no gain at any L — the downstream BC benefit appears exactly to the degree the latent recovers the strategy. Encoder quality translates directly into control benefit.

Table 4: Predator behaviour cloning with an opponent-strategy latent: held-out action accuracy, episode-level train/val split, watch episode 1 $\rightarrow$ predict episode 2 (mean $\pm$ std, 3 BC seeds). The oracle ceiling shows the conditioning effect is real; the unsupervised episode-1 latents do not yet carry enough signal to realize it.

predator BC variant	held-out action acc.
$\pi(a \mid s)$ (naive BC)	0.776 ± 0.004
$\pi(a \mid s, z_{\text{VAE}})$	0.772 ± 0.006
$\pi(a \mid s, z_{\text{JEPA}})$	0.772 ± 0.004
$\pi(a \mid s,$ **placement**$)$ **(oracle)**	$\mathbf{0.800 \pm 0.007}$

7 Related work

Opponent modeling and theory of mind. Modeling other agents is a long-standing problem [1]. Deep approaches range from encoding opponent observations into a value network [12] to

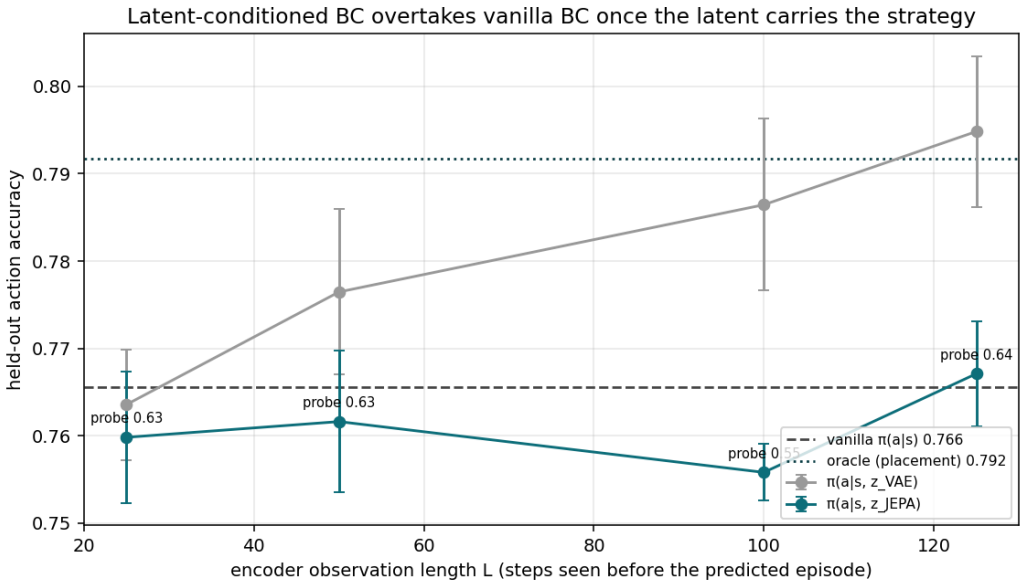

Figure 7: Latent-conditioned predator BC vs. encoder observation length L (the steps the encoder sees before the held-out episode, steps 125–150 of a six-episode specialist rollout; 4-D latents, 3 BC seeds). As L grows the VAE latent's placement probe rises (annotated) and its BC accuracy climbs from below vanilla at L=25 to the oracle (true-placement) ceiling by L=125; the JEPA latent's probe stays flat and it yields no gain. The downstream BC benefit tracks the latent's probe — encoder quality is the lever.

meta-learning a prior that infers an agent's goals from behavior alone [28], with recursive variants that reason about what the opponent believes about us [36] and recent attention-based instantiations that infer other agents' goal-attention states [16]. Closest to our Part 1 is the unsupervised opponent encoder of Papoudakis and Albrecht [25], which models opponents with a variational autoencoder from local information. Our predictive encoder targets exactly that setting and improves on it: a VAE spends capacity reconstructing the prey's evasion noise, whereas a joint-embedding predictor keeps only what is predictable—the strategy (probe accuracy 0.89 vs 0.53, matching a supervised classifier). Reasoning about an *uncertain* opponent type is studied for imperfect-information games [34]; our belief-sharpness ablation sharpens the point—acting on a confident but wrong type (−47%) is worse than ignoring type altogether.

Non-stationarity and adaptive opponents. When opponents adapt, the learning target moves [13]. Meta-learning yields few-shot adaptation against changing opponents [2], online schemes detect and switch responses as the opponent type changes mid-interaction [9], and recent work adds *active* context-aware exploration that gathers information to identify the peer before exploiting it [17]. Our prey draws a *fixed* hidden intent per episode, so we study calibrated inference of a static type rather than tracking an adaptive one; making the prey switch or best-respond, and tracking it online, is the natural escalation and the regime in which recursive reasoning [36] and meta-adaptation [2] become the right comparisons.

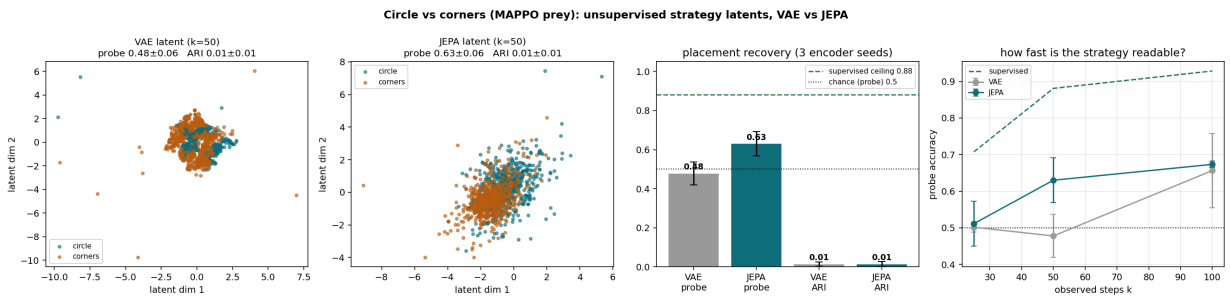

Figure 8: Circle vs. corners specialists, occupancy features (2-D latents, 3 encoder seeds). Left to right: VAE and JEPA latent scatters, placement recovery vs. the supervised ceiling, and the observation-length sweep. The signal is present (supervised 0.93 at k=100) and JEPA leads the VAE at every length at this latent size, but no unsupervised clustering emerges (ARI ≈ 0) — the honest open problem on this axis.

Model-based MARL and planning with a learned model. Planning with a learned model attains strong sample efficiency in single-agent RL [30, 37, 10], and recent work lifts this to multiple agents—Dreamer-style world models [8], multi-agent tree search over a learned model [22], and opponent-aware model-based methods that imagine improving opponent policies [38] or build per-opponent rollout models with sample-complexity guarantees [39]; transformer world models now lead this line on sample efficiency [40, 7]. Closest to our full pipeline, HOP [15] couples a Bayesian belief over other agents' goals—updated both within and across episodes—with Monte-Carlo tree search and reports few-shot adaptation to unseen opponents; we differ in the predictive, label-free opponent encoder and the calibrated-uncertainty ablations. Our planner (simulator-based, and its learned-model variant) is in this family but deliberately minimal: rather than learn an opponent *policy* or *dynamics* model, we infer a low-dimensional *intent* belief and plan against it. We have not yet benchmarked against [22, 8, 38, 15, 7]; doing so—measuring episodes-to-recover against a novel prey under limited interaction—is the experiment that would directly test whether an explicit intent belief is more sample-efficient than a learned opponent model, the central hypothesis of the broader project.

Self-predictive representations and JEPA. Predicting one's own future latent state via an EMA target encoder and a learned transition model is a powerful auxiliary for sample-efficient RL [31], unified theoretically as self-predictive abstraction [24]; the joint-embedding predictive architecture applies the same non-generative principle to images [3], to trajectories [19], and—action-conditioned—to video world models that support planning directly in representation space [4]. Our contribution is to transplant this machinery from *self*-modeling to *opponent* modeling: the JEPA encoder predicts the representation of the opponent's future motion, which we show is a better strategy code than a reconstructive VAE and yields a label-free belief that matches the supervised pipeline.

Symmetry and the strategy-versus-position gap. A representation that organizes by *where* an agent goes rather than by its *strategy* is exactly what equivariant networks address: an MDP homomorphic network is equivariant to symmetries of the state-action space [26], and its multi-agent form notes that "rotating the state globally results in a permutation of the optimal joint policy" [27]. Euclidean-equivariant actor-critics [5] and equivariant learned models [6] extend this to

cooperative control and planning, and equivariant graph networks bring 2–5× generalization gains to MARL on this very benchmark family—with the documented caveat that naive equivariance can hurt early exploration [18]. Our arena's four corners carry a dihedral (D_4) symmetry, so a D_4-equivariant JEPA encoder—whose strategy latent is invariant to which corner is targeted—is the principled route to a representation that captures strategy independently of position; we flag this as the highest-leverage open direction. A complementary, *learned* route is to pick the intent representation by its value to the ego agent: mixing multiple intent representations and maximizing mutual information with the ego agent's future returns [32]—a natural baseline for the equivariant encoder. When latent abstraction discards control-relevant detail [35], planning through it can fail, which accounts for our negative result with a JEPA world model.

Belief-space planning. Planning over a belief about partially observed agents is classical in interactive POMDPs, including differentiable instantiations [11] and scalable type-based reasoning that maintains a belief over discrete opponent types and plans with Monte-Carlo tree search [33]; modern systems online-update a latent intent belief to feed a tree-search planner [14]. Our pipeline has the same shape—infer a belief, then plan—at small scale, in the JaxMARL [29] reimplementation of the MPE predator–prey task [23], with the distinguishing features of a predictive, label-free opponent encoder.

8 Relation to the proposal and next steps

This study validates the proposal's loop end to end on its first domain. Part 1: the opponent's latent strategy is recoverable from its trajectory into a calibrated belief, and—crucially for co-training—a model that ignores that uncertainty is brittle to a strategy it did not expect (−47%), the overfitting failure adaptive opponent modeling targets. Part 2: sampling the uncertainty-aware opponent model inside a planner produces the most robust best response against an opponent that varies its strategy every episode, beating even an oracle that is handed the answer. The natural next steps are exactly the full proposal: replace the discrete classifier with the variational encoder $q_\phi(z \mid \tau^{\text{red}})$ and a latent-conditioned π_ψ^{red} (behavior cloning, then contrastive preference learning with planner-generated counterfactuals), and the simulator rollout with a MuZero-style learned model and a strategy-conditioned value $v^{\text{blue}}(s, z)$, then close the co-training loop so red and blue adapt to each other.

References

[1] S. V. Albrecht and P. Stone. Autonomous agents modelling other agents: A comprehensive survey and open problems. *Artificial Intelligence*, 258:66–95, 2018.

[2] M. Al-Shedivat, T. Bansal, Y. Burda, I. Sutskever, I. Mordatch, and P. Abbeel. Continuous adaptation via meta-learning in nonstationary and competitive environments. *ICLR*, 2018.

[3] M. Assran, Q. Duval, I. Misra, P. Bojanowski, P. Vincent, M. Rabbat, Y. LeCun, and N. Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. *CVPR*, 2023.

[4] M. Assran, A. Bardes, D. Fan, Q. Garrido, R. Howes, et al. V-JEPA 2: Self-supervised video models enable understanding, prediction and planning. *arXiv:2506.09985*, 2025.

[5] D. Chen and Q. Zhang. E(3)-equivariant actor-critic methods for cooperative multi-agent reinforcement learning. *ICML*, 2023.

[6] A. Deac, T. Weber, and G. Papamakarios. Equivariant MuZero. *Transactions on Machine Learning Research*, 2023.

[7] A. Deihim, E. Alonso, and D. Apostolopoulou. Transformer world model for sample efficient multi-agent reinforcement learning. *arXiv:2506.18537*, 2025.

[8] V. Egorov and A. Shpilman. Scalable multi-agent model-based reinforcement learning. *AAMAS*, 2022.

[9] R. Everett and S. Roberts. Learning against non-stationary agents with opponent modelling and deep reinforcement learning. *AAAI Spring Symposia*, 2018.

[10] D. Hafner, T. Lillicrap, J. Ba, and M. Norouzi. Dream to control: Learning behaviors by latent imagination. *ICLR*, 2020.

[11] Y. Han and P. Gmytrasiewicz. IPOMDP-Net: A deep neural network for partially observable multi-agent planning using interactive POMDPs. *AAAI*, 2019.

[12] H. He and J. Boyd-Graber. Opponent modeling in deep reinforcement learning. *ICML*, 2016.

[13] P. Hernandez-Leal, M. Kaisers, T. Baarslag, and E. Munoz de Cote. A survey of learning in multiagent environments: Dealing with non-stationarity. *arXiv:1707.09183*, 2017.

[14] Z. Huang, C. Tang, C. Lv, M. Tomizuka, and W. Zhan. Learning online belief prediction for efficient POMDP planning in autonomous driving. *IEEE Robotics and Automation Letters*, 2024.

[15] Y. Huang, A. Liu, F. Kong, Y. Yang, S.-C. Zhu, and X. Feng. Efficient adaptation in mixed-motive environments via hierarchical opponent modeling and planning. *ICML*, 2024.

[16] Q. Long, R. Li, M. Zhao, T. Gao, and D. Terzopoulos. Inverse attention agents for multi-agent systems. *ICLR*, 2025.

[17] L. Ma, Y. Wang, F. Zhong, S.-C. Zhu, and Y. Wang. Fast peer adaptation with context-aware exploration. *ICML*, 2024.

[18] J. McClellan, N. Haghani, J. Winder, F. Huang, and P. Tokekar. Boosting sample efficiency and generalization in multi-agent reinforcement learning via equivariance. *NeurIPS*, 2024.

[19] L. Li, H. Xue, Y. Song, and F. D. Salim. T-JEPA: A joint-embedding predictive architecture for trajectory similarity computation. *ACM SIGSPATIAL*, 2024.

[20] T. Li, K. Zhu, J. Li, and Y. Zhang. Learning distinguishable trajectory representation with contrastive loss. *NeurIPS*, 2024.

[21] S. Lei, K. Lee, L. Li, and J. Park. Learning strategy representation for imitation learning in multi-agent games. *AAAI*, 2025.

[22] Q. Liu, J. Ye, X. Ma, J. Yang, B. Liang, and C. Zhang. Efficient multi-agent reinforcement learning by planning. *ICLR*, 2024.

[23] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. *NeurIPS*, 2017.

[24] T. Ni, B. Eysenbach, E. Seyedsalehi, M. Ma, C. Gehring, A. Mahajan, and P.-L. Bacon. Bridging state and history representations: Understanding self-predictive RL. *ICLR*, 2024.

[25] G. Papoudakis and S. V. Albrecht. Variational autoencoders for opponent modeling in multi-agent systems. *arXiv:2001.10829*, 2020.

[26] E. van der Pol, D. E. Worrall, H. van Hoof, F. Oliehoek, and M. Welling. MDP homomorphic networks: Group symmetries in reinforcement learning. *NeurIPS*, 2020.

[27] E. van der Pol, H. van Hoof, F. Oliehoek, and M. Welling. Multi-agent MDP homomorphic networks. *ICLR*, 2021.

[28] N. C. Rabinowitz, F. Perbet, H. F. Song, C. Zhang, S. M. A. Eslami, and M. Botvinick. Machine theory of mind. *ICML*, 2018.

[29] A. Rutherford, B. Ellis, M. Gallici, et al. JaxMARL: Multi-agent RL environments and algorithms in JAX. *arXiv:2311.10090*, 2023.

[30] J. Schrittwieser, I. Antonoglou, T. Hubert, K. Simonyan, L. Sifre, S. Schmitt, A. Guez, E. Lockhart, D. Hassabis, T. Graepel, T. Lillicrap, and D. Silver. Mastering Atari, Go, chess and shogi by planning with a learned model. *Nature*, 588:604–609, 2020.

[31] M. Schwarzer, A. Anand, R. Goel, R. D. Hjelm, A. Courville, and P. Bachman. Data-efficient reinforcement learning with self-predictive representations. *ICLR*, 2021.

[32] M. Odrowaz-Sypniewski, J. Bayrooti, A. Shankar, and A. Prorok. Generalized intention modeling (MIX) in multi-agent reinforcement learning. *arXiv:2605.31318*, 2026.

[33] J. Schwartz, H. Kurniawati, and M. Hutter. Combining a meta-policy and Monte-Carlo planning for scalable type-based reasoning in partially observable environments. *arXiv:2306.06067*, 2023.

[34] M. Shen and J. P. How. Robust opponent modeling via adversarial ensemble reinforcement learning in asymmetric imperfect-information games. *arXiv:1909.08735*, 2019.

[35] B. Terver, T.-Y. Yang, J. Ponce, A. Bardes, and Y. LeCun. What drives success in physical planning with joint-embedding predictive world models? *arXiv:2512.24497*, 2025.

[36] Y. Wen, Y. Yang, R. Luo, J. Wang, and W. Pan. Probabilistic recursive reasoning for multi-agent reinforcement learning. *ICLR*, 2019.

[37] W. Ye, S. Liu, T. Kurutach, P. Abbeel, and Y. Gao. Mastering Atari games with limited data. *NeurIPS*, 2021.

[38] X. Yu, J. Jiang, W. Zhang, H. Jiang, and Z. Lu. Model-based opponent modeling. *NeurIPS*, 2022.

[39] W. Zhang, X. Wang, J. Shen, and M. Zhou. Model-based multi-agent policy optimization with adaptive opponent-wise rollouts. *IJCAI*, 2021.

[40] Y. Zhang, C. Bai, B. Zhao, J. Yan, X. Li, and X. Li. Decentralized transformers with centralized aggregation are sample-efficient multi-agent world models. *Transactions on Machine Learning Research*, 2024.