MULTI-AGENT CONTRASTIVE WORLD MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Cooperative multi-robot control is usually learned from a hand-designed reward and often relies on communication between robots at execution. We drop both assumptions and present the Multi-Agent Contrastive World Model (MACWM), which combines a contrastive goal-conditioned critic with an action-conditioned world model that predicts the critic's own latent representation. A critic alone gives the actor a weak signal, because at a high control rate one command barely changes which goals the team reaches. MACWM therefore lets the critic score short rollouts of the world model inside the actor's training objective, and each robot executes its policy on its own observation. We train it and reward-free, communication-free baselines under one protocol on a benchmark of cooperative tasks for pairs of quadruped robots in Isaac Lab. Across nine tasks MACWM is the only one of these methods whose every training run reaches the goal, and over a wide band of success thresholds it fails less often than either contrastive baseline. With three to five robots our method leads both contrastive baselines on every seed of cooperative box pushing, and its policies transfer to two real quadrupeds without further training. See our anonymous project webpage: MACWM.

1 INTRODUCTION

Learning to control a team of robots usually relies on a hand-designed task reward (Xiong et al., 2024; Zhao et al., 2026), which presumes a designer who already knows what good behavior looks like and can express it as a scalar. Many methods also add a communication channel at execution (Egorov & Shpilman, 2022; Wen et al., 2022; Zhao et al., 2026), which presumes bandwidth and synchronization that fielded robots often lack.

Either assumption can be dropped on its own. Reward-free control is addressed by goal-conditioned methods, which replace the designer's scalar with a target configuration and learn from hindsight (Andrychowicz et al., 2017). Communication-free control is addressed by centralized training with decentralized execution (CTDE), in which a privileged critic is discarded after training (Lowe et al., 2017; Rashid et al., 2018). Dropping both at once is harder than dropping either. Without reward, the learning signal must come from the agent's own experience of reaching goals, and that signal is weakest where coordination matters, when progress requires two robots to act together and neither is closer to success. Without communication, no agent can condition on what the other intends.

This paper works entirely inside that intersection. We build on contrastive goal-conditioned reinforcement learning, which learns a critic by classifying which future states follow which state-action pairs, and scores a pair by its distance to a goal in a learned embedding space (Eysenbach et al., 2022; van den Oord et al., 2018). That critic scores one action at a time, and at the control rate of legged robots one command barely changes which goals the team reaches. We therefore add an action-conditioned world model over the critic's latent space so that the critic can also score short sequences of commands. The model enters only the actor's training objective, where the critic scores each imagined step against the goal so that every sampled action receives a gradient. Because the model and the critic both read the whole team, that gradient accounts for the teammate. On a trained critic, the values of commands near the actor's own differ several times more when scored through a few imagined steps than when scored by the critic alone.

Our contributions are the following:

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.

1. MACWM, a reward-free and communication-free method in which a world model over a contrastive critic's latent space enters only the actor's training objective, so each robot executes its policy on its own observation.
2. A benchmark of cooperative Unitree Go1 tasks in Isaac Lab, specified by goal and success predicate, on which MACWM and the reward-free, communication-free baselines are trained and evaluated under one protocol.
3. An evaluation on nine tasks in which MACWM fails less often than the baselines across a wide band of success thresholds. Ablations cross the world model with its exploration noise, and a probe of the trained critic shows what the imagined actor loss adds. With three to five robots MACWM leads both contrastive baselines on every seed of cooperative box pushing, and its policies transfer to two Unitree Go2 robots without further training.

2 Related Work

Hindsight relabeling treats whatever a goal-conditioned agent reached as though it had been intended (Andrychowicz et al., 2017). Contrastive RL casts goal-conditioned value learning as InfoNCE classification (Eysenbach et al., 2022; van den Oord et al., 2018). Temporal-difference variants add bootstrapped stitching (Zheng et al., 2024), and deep residual encoders improve the family (Wang et al., 2025). We carry this line into cooperative multi-robot control.

Carrying a contrastive objective into cooperative control raises the question of how much of the critic to centralize. ICRL (Nimonkar et al., 2025) trains independent per-robot critics, and each robot treats its teammates as part of the environment. MACRL (Kich et al., 2026b) instead scores the whole team with one centralized critic, which can represent value that exists only in the joint action, and factors it by pooling per-robot embeddings instead of encoding the team jointly. Centralized critics with decentralized actors are the standard route to deployable multi-agent policies (Lowe et al., 2017; Rashid et al., 2018; Yu et al., 2022). Sequence models such as MAT decide agents in order, each conditioned on the actions chosen before it (Wen et al., 2022).

Reconstruction-based models predict observations (Hafner et al., 2025), while value-equivalent models predict only what affects decisions (Schrittwieser et al., 2020; Grimm et al., 2020; Hansen et al., 2024), and contrastive transition models learn structured latents without a decoder (Kipf et al., 2020). Ours predicts the critic's latent with action-conditioned contrastive predictive coding (Burchi & Timofte, 2025), reuses the critic's energy as its value term, and decodes observations only to re-encode imagined steps. In the multi-agent setting, MAMBA (Egorov & Shpilman, 2022) gives each agent its own model and communicates, MARIE (Zhang et al., 2025) learns aggregated models, and MACT (Kich et al., 2026a) keeps decentralized models, aggregates the team with a single Perceiver step during training and learns by action-conditioned contrastive coding. SeqWM (Zhao et al., 2026) plans on quadrupeds from the MQE benchmark (Xiong et al., 2024), and each agent passes its intended plan to the agents deciding after it. In all four a task reward drives the policy, and no goal-conditioned critic scores imagined rollouts.

3 Method

MACWM has three learned components, a contrastive critic, a world model over the critic's latents and a decentralized actor. Figure 1 shows how they compose during training and what survives at execution.

3.1 Problem setting

We model a cooperative task as a decentralized partially observable Markov decision process (Dec-POMDP) (Oliehoek & Amato, 2016) without a reward function. N robots act in an environment with state s_t. Robot j receives a local observation o_t^j of that state and chooses an action a_t^j, and the joint action $a_t = (a_t^1, \dots, a_t^N)$ drives the transition $s_{t+1} \sim P(\cdot \mid s_t, a_t)$.

Goals take the place of the reward. At the start of an episode each robot is assigned a desired goal g_*^j. A task-defined map gives the goal it has actually achieved, $g_t^j = \phi^j(s_t)$, such as its own planar position or the position of an object it moves. We write $g_t = (g_t^1, \dots, g_t^N)$ and $g_* = (g_*^1, \dots, g_*^N)$

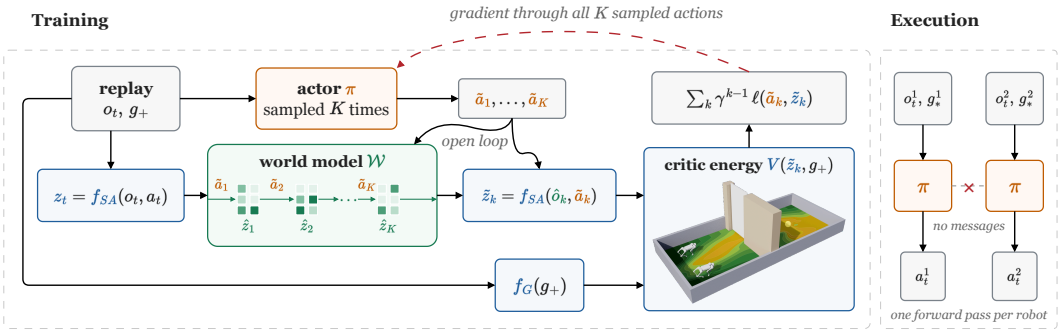


Figure 1: Training, left, where the critic's energy scores a K-step rollout of $\mathcal{W}$, and execution, right, where each robot runs π on its own observation. Symbols are colored by the component that computes them, critic blue, world model green and actor orange. The inset shows the critic's energy over the Door arena (Appendix G).

for the team's achieved and desired goals. A task-defined success predicate, for example the box resting at its target, decides whether an episode succeeds. It is used to evaluate every method and never to train ours. We call a method reward-free when it receives no reward designed for the task.

Without a reward, we measure a policy π by the goals its team reaches. With a discount $\gamma \in (0, 1)$, the discounted occupancy of team goals,

$$p_\gamma^\pi(g \mid s_t, a_t) \;=\; (1-\gamma) \sum_{\Delta=1}^{\infty} \gamma^{\Delta-1}\, p^\pi\big(g_{t+\Delta} = g \mid s_t, a_t\big), \tag{1}$$

is the density of reaching the team goal g at an offset Δ drawn from a geometric distribution with parameter $1-\gamma$, and the task is to place as much of it as possible on g_*. This density is the Q-function of a reward that pays the probability of reaching g at the next step (Eysenbach et al., 2022), so maximizing it is a reinforcement learning problem with no designed reward.

Execution is decentralized. Each robot acts on its own observation and goal,

$$\pi(a_t \mid o_t, g_*) \;=\; \prod_{j=1}^{N} \pi\big(a_t^j \mid o_t^j, g_*^j\big), \tag{2}$$

with parameters shared across robots. No message is exchanged. Training is centralized, and the critic and the world model may read every robot's observations and actions. The occupancy of Eq. 1 depends on the joint action, as when a box moves only if both robots push, while each factor of Eq. 2 sees a single robot's observation. MACWM uses its centralized components to give each factor a training signal that accounts for its teammates.

In the main comparison $N = 2$, and each robot is a Unitree Go1 quadruped (Section 4.4 extends Gate, Door and PushBox to larger teams). Its observation o_t^j contains its proprioception, the last command it issued, its goal error $g_*^j - g_t^j$, and egocentric offsets to its teammate and to the task's salient geometry, with no map and no exteroception. Its action $a_t^j \in [-1, 1]^3$ is a base-velocity command, which a frozen 50 Hz locomotion policy tracks. That policy is identical for every robot and never updated by our algorithm. On six of the nine tasks the observation also carries a one-hot agent identity, and every method uses the same observation and action spaces on every task. The full specifications, including the posture objectives, are in Appendix A.

3.2 Contrastive goal-conditioned critic

The critic estimates the occupancy of Eq. 1 from trajectories alone. A state-action encoder f_{SA}, shared by the robots, embeds each robot's observation and action, and a goal encoder f_G embeds a team goal. A factored critic (Kich et al., 2026b) compares the mean of the robots' embeddings with the goal embedding,

$$z_t^j = f_{SA}\big(o_t^j, a_t^j\big), \qquad V(z_t, g) \;=\; -\Big\| \frac{1}{N} \sum_{j=1}^{N} z_t^j - f_G(g) \Big\|_2, \tag{3}$$

where $z_t = (z_t^1, \dots, z_t^N)$. The critic reads the joint observation o_t in place of the state s_t of Eq. 1, treating it as sufficient to predict the team's achieved goals. Writing it on latents lets the same energy score the world model's predictions in Section 3.5.

The critic learns to tell the goals a team reached from the goals it did not. For an anchor step t, a relabeled goal $g_+ = g_{t+\Delta}$ serves as the positive. It is read from the same trajectory at an offset Δ drawn with probability proportional to $\gamma^{\Delta-1}$ over the $\Delta_{\max}(t)$ steps left in the episode, which is hindsight relabeling (Andrychowicz et al., 2017) in the form Eq. 1 prescribes. In a batch of B anchors, row i has latents $z_{t,i}$ and relabeled goal $g_{+,i}$, and the other rows' goals serve as its negatives,

$$\mathcal{L}_V = -\frac{1}{B}\sum_{i=1}^{B} \log \frac{e^{V(z_{t,i},\, g_{+,i})}}{\sum_{i'=1}^{B} e^{V(z_{t,i},\, g_{+,i'})}} \;+\; \lambda_{\text{lse}}\, \frac{1}{B}\sum_{i=1}^{B} \Big(\log \sum_{i'=1}^{B} e^{V(z_{t,i},\, g_{+,i'})}\Big)^2. \tag{4}$$

At the optimum of the first term, V equals $\log p_\gamma^\pi(g \mid s, a) - \log p(g)$ plus a function of the state and action (van den Oord et al., 2018; Eysenbach et al., 2022). That function can change with the action, and the second term keeps it close to a constant, so raising V by changing the action raises the occupancy at g. The actor's loss in Section 3.4 relies on this. The identity also requires a critic that can represent the optimum, and averaging per-robot embeddings narrows that class relative to a monolithic encoder of the whole team. We accept the restriction because the mean is symmetric under permuting the robots and keeps the per-robot latents z_t^j that the world model predicts. A per-robot critic that scores each robot against its own goal, as in ICRL (Nimonkar et al., 2025), is the main alternative (Appendix F).

Appendix G maps this energy over the arena for one seed of each contrastive critic (Figure 12).

3.3 Effective credit horizon

Drawing offsets up to the end of the episode requires the sampler to reach whole episodes. Replay stores fixed slices of T steps, and drawing the offset inside the anchor's slice replaces $\Delta_{\max}(t)$ with the distance to the end of that slice. The horizon the critic represents then depends on where the anchor falls, and it falls short of the $1/(1-\gamma)$ steps that γ implies unless slices are much longer than that. Eq. 4 cannot reveal the mismatch, because it is computed on whatever positives the sampler returns. We therefore concatenate S consecutive slices, mask episode boundaries, and renormalize the offset distribution over the steps that remain reachable. On the four tasks where we measured it, with two seeds each, that change alone raises mean success by 0.37 (Appendix F).

3.4 Decentralized actor

The actor is trained on the relabeled goals of Section 3.2. Each robot samples $\tilde{a}^j \sim \pi(\cdot \mid o_t^j, g_+^j)$, where its observation o_t^j still carries the error to the desired goal g_*^j, and the one-step loss raises the critic's energy at replayed observations,

$$\ell(\tilde{a}, z) \;=\; \alpha \sum_{j=1}^{N} \log \pi\big(\tilde{a}^j \mid o_t^j, g_+^j\big) - V(z, g_+), \qquad \mathcal{L}_\pi^{\text{1-step}} \;=\; \mathbb{E}\big[\ell\big(\tilde{a}, f_{SA}(o_t, \tilde{a})\big)\big], \tag{5}$$

where $f_{SA}(o_t, \tilde{a})$ encodes each robot's observation with its sampled action and α is an entropy temperature tuned toward a target entropy (Haarnoja et al., 2018a;b).

This signal is weak at our control rate. The advantage of a single action scales with the control interval and vanishes as the interval shrinks (Tallec et al., 2019), and each velocity command lasts 20 ms. The world model adds signal the one-step loss lacks. It lets the critic score a short sequence of sampled commands at predicted future latents, so each command also receives a gradient through its effect on the next few steps. That sequence spans $K\sigma$ environment steps, a tenth of a second at the settings of Table 8, and rolling the model out eight times further did not improve success on average (Appendix F).

3.5 LATENT WORLD MODEL

The world model $\mathcal{W}$ predicts how the robots' latents evolve under a sequence of joint actions. From $\hat{z}_0 = z_t$ it applies one residual update per step,

$$\hat{z}_k \;=\; \hat{z}_{k-1} + \mathcal{W}\big(\hat{z}_{k-1},\, a_{t+k\sigma}\big), \qquad k = 1, \dots, K, \tag{6}$$

where $\hat{z}_k = (\hat{z}_k^1, \dots, \hat{z}_k^N)$ and step k predicts the latents $k\sigma$ environment steps ahead for a stride σ. Attention across robots makes the prediction for robot j depend on every robot's latent and action (Appendix B).

We train $\mathcal{W}$ with the critic's kind of energy, by action-conditioned contrastive predictive coding (AC-CPC) (Burchi & Timofte, 2025). Each predicted latent has to identify its true future latent among those of the other rows in the batch,

$$\mathcal{L}_{\text{CPC}} = -\sum_{k=1}^{K} \rho^{k-1} \frac{1}{NB} \sum_{j=1}^{N} \sum_{i=1}^{B} \log \frac{\exp\big(-\|\hat{z}_{k,i}^j - z_{t+k\sigma,i}^j\|_2\big)}{\sum_{i'=1}^{B} \exp\big(-\|\hat{z}_{k,i}^j - z_{t+k\sigma,i'}^j\|_2\big)}, \tag{7}$$

where $\rho < 1$ weights near steps more and the targets are held fixed, so training the model never changes the critic's representation.

Whole predicted latents are more than the actor needs, because it reads only the team energy against a goal. Following the value-equivalence principle (Grimm et al., 2020), we also ask the predicted latents to score like the true ones,

$$\mathcal{L}_{\text{VE}} \;=\; \sum_{k=1}^{K} \rho^{k-1}\, \mathbb{E}\Big[\big(V(\hat{z}_k, g_+) - V(z_{t+k\sigma}, g_+)\big)^2\Big]. \tag{8}$$

The term adds no parameters, because the value is the critic's energy of Eq. 3.

Nothing forces a predicted latent onto the set of latents the encoder produces, so the critic could score it where it was never fitted. An observation head therefore decodes each of $K_o \geq K$ model steps into observations $\hat{o}_k^j$,

$$\mathcal{L}_{\text{obs}} \;=\; \sum_{k=1}^{K_o} \rho^{k-1}\, \mathbb{E}\Big[\big\|\hat{o}_k - o_{t+k\sigma}\big\|_2^2\Big], \tag{9}$$

and the actor's loss scores imagined steps only after re-encoding them (Section 3.6). The model minimizes $\mathcal{L}_{\mathcal{W}} = \mathcal{L}_{\text{CPC}} + \lambda_{\text{VE}}\mathcal{L}_{\text{VE}} + \lambda_o\mathcal{L}_{\text{obs}}$, and none of these terms reaches the critic's encoders (Appendix B).

3.6 IMAGINED ACTOR LOSS

The imagined term scores a sequence of sampled actions through the model. It draws K joint actions $\tilde{a}_k$ with $\tilde{a}_k^j \sim \pi(\cdot \mid o_t^j, g_+^j)$ and rolls Eq. 6 forward from the replayed latents z_t with $\tilde{a}_k$ in place of $a_{t+k\sigma}$. Each step is scored at the re-encoded latents $\tilde{z}_k^j = f_{SA}(\hat{o}_k^j, \tilde{a}_k^j)$, which the critic's own encoder produces, and the actor minimizes

$$\mathcal{L}_\pi \;=\; \mathcal{L}_\pi^{\text{1-step}} + \frac{\eta}{K} \sum_{k=1}^{K} \gamma^{k-1}\, \mathbb{E}\big[\ell(\tilde{a}_k, \tilde{z}_k)\big]. \tag{10}$$

The imagined term discounts by γ once per model step. The chain from the sampled actions through $\mathcal{W}$, the observation head and f_{SA} is differentiable, so gradients reach the actor through every action of the sequence, and only the actor's parameters are updated by this loss. All K actions are drawn at the same replayed observation, which makes the rollout open-loop (Appendix F). At execution, each robot conditions π on its desired goal g_*^j.

During data collection, pink noise (Eberhard et al., 2023) replaces the independent sampling noise of the actor with temporally correlated noise of the same per-step marginal, so the log-probabilities in Eqs. 5 and 10 and the temperature tuning are unaffected. Appendix F compares it with Gaussian noise.

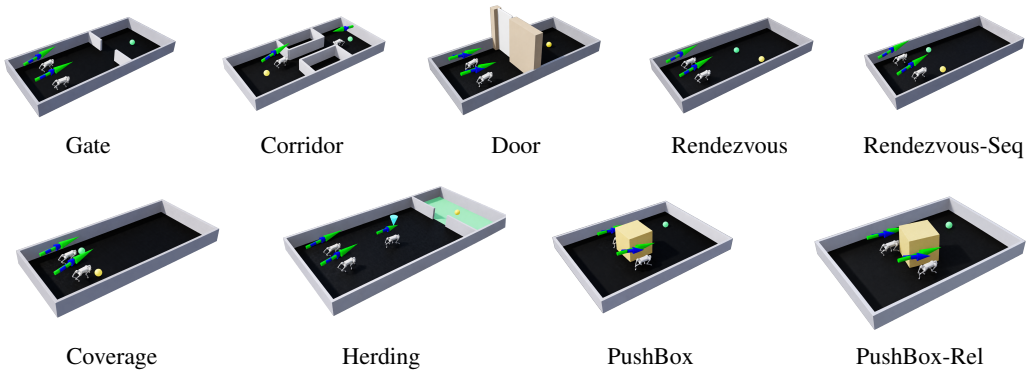


Figure 2: The nine benchmark tasks. Balls mark the goals, and arrows show each robot's commanded (green) and current (blue) velocity.

4 EXPERIMENTS

We evaluate on nine cooperative tasks in Isaac Lab, each with two Unitree Go1 quadrupeds (Figure 2). They cover aperture navigation (Gate, Door), swapping sides through a narrow corridor (Corridor), object transport with absolute and relational goals (PushBox, PushBox-Rel), meeting points that are crossed or ordered (Rendezvous, Rendezvous-Seq), interleaved waypoints (Coverage), and driving a fleeing quadruped through a barrier (Herding). Appendix A gives each task's goal, success predicate and episode length. Every reward-free method is trained on each task with eight seeds under a budget of 4M environment steps, and the reward-based reference methods use four. The ablations of Appendix F use the first four seeds of each task. We report final-window success, which consist in a mean of the last five periodic evaluations of a completed run.

4.1 SETTING AND BASELINES

The methods fall into two groups by what they assume (Table 1). Every method in our main comparison trains without a task reward, executes without messages and shares one policy network across the team. Methods that require a designed reward or pass messages serve only as a reference point.

MACRL (Kich et al., 2026b) and ICRL (Nimonkar et al., 2025) are contrastive goal-conditioned methods implemented in our codebase with Gaussian exploration noise and no world model. MACRL scores the team with the same factored critic as MACWM and trains its actor with the one-step loss of Eq. 5 alone, while ICRL uses independent per-robot critics. MA-SAC+HER and MA-GCIQL keep the codebase and hindsight relabeling (Andrychowicz et al., 2017) but replace the contrastive critic by a monolithic $Q(s, g, a)$, trained by TD on a self-labeled indicator of reaching the relabeled goal, and pair it with soft actor-critic (Haarnoja et al., 2018a;b) or implicit Q-

Table 1: What each method assumes, and whether one policy network serves the whole team. A mixed row groups methods that share a policy (MAPPO, IPPO) with methods that do not (HASAC, SeqWM).

	Requires			
Method	Task reward	Messages	Policy	Learning signal
MACWM (ours)	–	–	shared	contrastive critic + world model
MACRL	–	–	shared	contrastive critic, factored
ICRL	–	–	shared	contrastive critic, per robot
MA-SAC+HER	–	–	shared	$Q(s, g, a)$, self-labeled indicator
MA-GCIQL	–	–	shared	$Q(s, g, a)$, self-labeled indicator
MAPPO, IPPO, SeqWM (sparse)	● sparse	–	mixed	task success indicator
MAPPO, IPPO, HASAC, SeqWM (DEC)	● dense	–	mixed	hand-designed dense reward
SeqWM (SEQ)	● dense	● plans	per robot	hand-designed dense reward

Table 2: Reward-free and communication-free methods. Mean ± sample standard deviation at a 4M-step budget, over eight seeds for every method.

Task	MACWM (ours)	MACRL	ICRL	MA-SAC+HER	MA-GCIQL
Gate	0.959 ± 0.077	$\mathbf{0.961} \pm 0.040$	0.946 ± 0.066	0.291 ± 0.204	0.368 ± 0.380
Corridor	0.460 ± 0.426	$\mathbf{0.549} \pm 0.384$	0.537 ± 0.280	0.000 ± 0.000	0.044 ± 0.074
Door	$\mathbf{0.589} \pm 0.241$	0.360 ± 0.258	0.318 ± 0.271	0.021 ± 0.031	0.005 ± 0.013
Coverage	$\mathbf{0.365} \pm 0.186$	0.284 ± 0.132	0.290 ± 0.181	0.032 ± 0.027	0.034 ± 0.020
Herding	$\mathbf{0.723} \pm 0.138$	0.395 ± 0.242	0.513 ± 0.236	0.054 ± 0.080	0.271 ± 0.195
PushBox	$\mathbf{0.460} \pm 0.218$	0.350 ± 0.301	0.086 ± 0.143	0.009 ± 0.011	0.024 ± 0.048
PushBox-Rel	0.716 ± 0.177	0.769 ± 0.093	$\mathbf{0.789} \pm 0.126$	0.089 ± 0.039	0.100 ± 0.183
Rendezvous	0.697 ± 0.073	0.732 ± 0.075	$\mathbf{0.752} \pm 0.064$	0.118 ± 0.021	0.284 ± 0.126
Rendezvous-Seq	0.698 ± 0.179	$\mathbf{0.833} \pm 0.127$	0.734 ± 0.152	0.186 ± 0.098	0.225 ± 0.226
Mean	**0.630**	0.581	0.552	0.089	0.150
Reach the goal	**72/72**	67/72	67/72	54/72	43/72
Ending below 0.05	**3/72**	6/72	7/72	36/72	38/72

learning (Kostrikov et al., 2022; Peng et al., 2019). Hyperparameters follow the original papers (Table 8). MAPPO (Yu et al., 2022), IPPO (de Witt et al., 2020), HASAC and SeqWM receive either the success indicator or a dense reward that applies the structure of MQE's reward wrappers (Xiong et al., 2024) to every task (Appendix C). With that dense reward, which our setting assumes is unavailable, HASAC (Liu et al., 2024) and SeqWM (Zhao et al., 2026) average about 0.9 over the nine tasks, a reference point for what this suite allows when a designer supplies the reward (Appendix C).

4.2 Main comparison

MACWM has the highest mean success in Table 2 and exceeds one half on six of the nine tasks. Our method leads on Door, Herding, PushBox and Coverage, the four tasks on which the best baseline scores lowest, and its lead is largest on Door and Herding, at 0.23 and 0.21 over the best baseline. A baseline leads on the other five, MACRL on Gate, Corridor and Rendezvous-Seq and ICRL on PushBox-Rel and Rendezvous. On Corridor each of the three contrastive methods has at least one seed that ends near zero and at least four that end above one half, so the spread lies within each method.

MACWM also has the fewest failures. All 72 of its runs reach the goal, with some evaluation above 0.05, against 67 for each contrastive baseline, and only three end below 0.05, against six and seven. Those three are all Corridor, and their best single evaluations lie between 0.06 and 0.11. Over all 72 runs the fraction clearing a given threshold is higher for MACWM at every threshold from 0.1 to 0.65, by 0.08 on average, which is five to six runs and close to the binomial standard error at this sample size. Its seed spread is no smaller than the baselines', at a mean per-task standard deviation of 0.19 against 0.18 and 0.17, and Figure 3 shows that the spread within each method is wider than the gap between them. Every seed is listed in Appendix E. Additional results on the two monolithic-Q methods can be also seen in Appendix E.

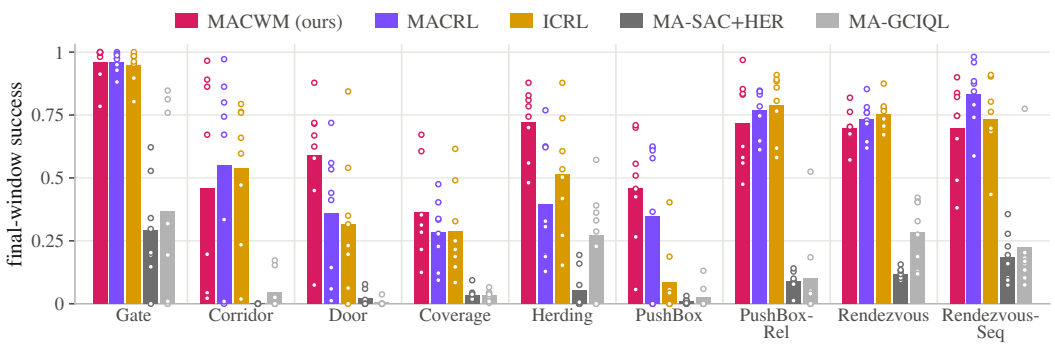


Figure 3: Per-task mean final-window success, with each method's eight seeds drawn over its bar.

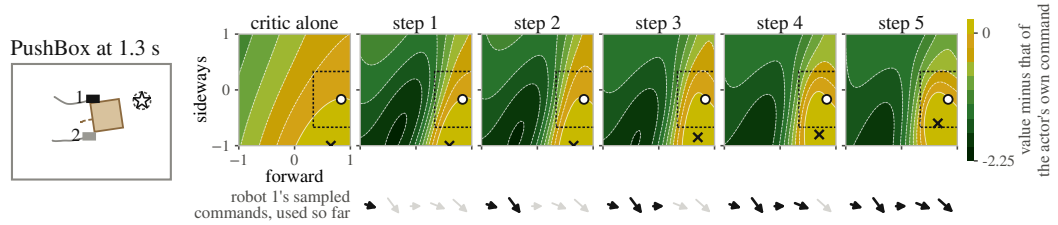


Figure 4: MACWM's PushBox critic scoring robot 1's command at one state (inset), relative to the actor's own command (white dot), by the critic alone and after k imagined steps. The dashed square holds the commands within 0.5 of the actor's own, and a cross marks the best.

4.3 What the imagined term adds

Appendix F ablates MACWM on the first four seeds of each task. Disconnecting the imagined term from the actor, $\eta = 0$ in Eq. 10, changes the nine-task mean by $+0.03$. On PushBox the term adds 0.36 and raises every seed, while on the other eight tasks the change lies between -0.08 and $+0.06$, within the seed spread of each. Pink noise is not specific to our method (Eberhard et al., 2023). Giving it to MACRL changes that baseline by 0.01 while it raises MACWM by 0.20. It makes MACWM's world model more accurate in 34 of 36 paired runs, so the imagined term trains the actor through a better model (Appendix F). Removing the logsumexp penalty of Eq. 4 or the multi-slice hindsight sampler of Section 3.3 makes every one of eight paired runs worse.

Figure 4 shows what the imagined term adds to the actor's signal on PushBox. It takes MACWM's PushBox policy during an episode that succeeds, sweeps robot 1's forward and sideways command, and scores each command as the imagined term does, held for k steps of the world model. Among the commands within 0.5 of the actor's own, roughly where the imagined term samples, the critic alone spans 0.38 in value. Scored through the model, the same commands span 1.24 to 1.51 over one to five steps, three to four times as much. On Door, where the term is within seed variation, the probe gives four to seven times (Appendix G.2).

4.4 Larger teams

The imagined term matters most on PushBox, where the box moves only if the robots push it together. We trained MACWM, MACRL and ICRL with three, four and five robots on PushBox, Gate and Door, the three tasks whose goals and success tests extend to any team size, with four seeds per setting and the two-robot hyperparameters (Appendix H). With three to five robots MACWM and MACRL are indistinguishable on Gate and Door (Figure 5), pooling to -0.04 and $+0.02$ with paired standard errors of 0.05, and from four robots on, every method fails most Door episodes. On PushBox MACWM leads MACRL by 0.38, 0.29 and 0.24 with three, four and five robots, and all 12 paired runs are ahead.

4.5 Real robots

We also took the two-robot PushBox and Gate policies of MACWM, MACRL and ICRL to two Unitree Go2 robots in a motion-capture room, without further training (Appendix I). One computer

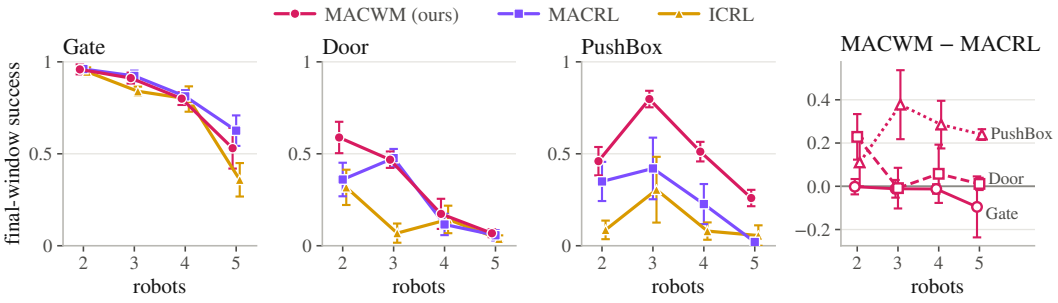


Figure 5: Final-window success by team size, as mean and standard error over eight seeds with two robots and four above. The right panel gives MACWM minus MACRL, paired by seed.

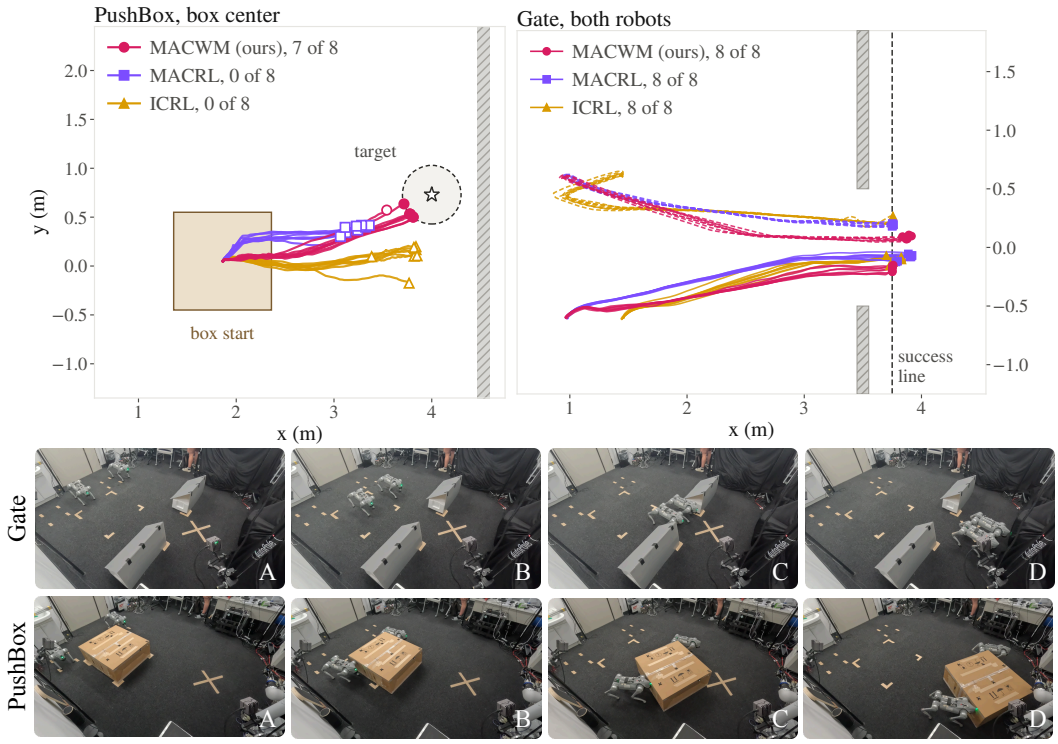


Figure 6: Top, all hardware trials by method, with successes out of eight. PushBox traces the box center and Gate both robots (robot 2 dashed), and a filled end marker is a success. Bottom, one MACWM trial per task from start (A) to end (D).

runs the shared policy for each robot on that robot's observation alone, assembled from motion capture in the simulated layout, and the manufacturer's walking controller tracks the velocity commands. On Gate every method succeeds in eight of eight trials. On PushBox MACWM succeeds in seven of eight trials and neither baseline succeeds (Figure 6). The PushBox room was fitted during MACWM's own trials and its start suits MACWM's policy, while from the simulator's randomized starts under the same walls and command limits ICRL succeeds most often. These trials therefore show transfer to hardware and do not rank the methods. Appendix I adds a simulated twin of every trial.

5 Discussion and Conclusion

We presented MACWM, which trains decentralized goal-conditioned policies without a task reward by scoring world-model rollouts with a contrastive critic. On nine cooperative quadruped tasks it is the only reward-free, communication-free method whose every run reaches the goal, and across a wide band of success thresholds it fails less often than the contrastive baselines. Its imagined actor loss more than doubles success on PushBox, where MACWM also leads both contrastive baselines on every seed with three to five robots, and its policies transfer to two Go2 robots without further training.

These results have limits. The imagined term is within seed variation on several tasks, including PushBox-Rel, the other transport task. Our training budget is far below the regime in which deep contrastive critics were shown to help (Wang et al., 2025), and on three tasks at least one method is still improving when it ends, so the ranking there may change with a larger budget. Few of our tasks require moving away from the goal before approaching it. Larger teams do not separate MACWM from MACRL on Gate and Door, and the hardware trials use one room fitted during MACWM's own trials. All our networks have one width and one depth, and contrastive critics in this family improve sharply with depth (Wang et al., 2025; Kich et al., 2026b), so what MACWM does at scale remains open. The question we would ask next is whether the world model's gain grows with deeper critics and longer training.

REFERENCES

M. Andrychowicz, F. Wolski, A. Ray, J. Schneider, R. Fong, P. Welinder, B. McGrew, J. Tobin, P. Abbeel, and W. Zaremba. Hindsight experience replay. In *Advances in Neural Information Processing Systems*, 2017.

M. Burchi and R. Timofte. Learning transformer-based world models with contrastive predictive coding. In *International Conference on Learning Representations*, 2025.

C. Schroeder de Witt, T. Gupta, D. Makoviichuk, V. Makoviychuk, P. H. S. Torr, M. Sun, and S. Whiteson. Is independent learning all you need in the StarCraft multi-agent challenge? *arXiv preprint arXiv:2011.09533*, 2020.

O. Eberhard, J. Hollenstein, C. Pinneri, and G. Martius. Pink noise is all you need: Colored noise exploration in deep reinforcement learning. In *International Conference on Learning Representations*, 2023.

V. Egorov and A. Shpilman. Scalable multi-agent model-based reinforcement learning. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2022.

B. Eysenbach, T. Zhang, S. Levine, and R. Salakhutdinov. Contrastive learning as goal-conditioned reinforcement learning. In *Advances in Neural Information Processing Systems*, 2022.

C. Grimm, A. Barreto, S. Singh, and D. Silver. The value equivalence principle for model-based reinforcement learning. In *Advances in Neural Information Processing Systems*, 2020.

T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning*, 2018.

T. Haarnoja, A. Zhou, K. Hartikainen, G. Tucker, S. Ha, J. Tan, V. Kumar, H. Zhu, A. Gupta, P. Abbeel, and S. Levine. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*, 2018.

D. Hafner, J. Pasukonis, J. Ba, and T. Lillicrap. Mastering diverse control tasks through world models. *Nature*, 640(8059):647–653, 2025.

N. Hansen, H. Su, and X. Wang. TD-MPC2: Scalable, robust world models for continuous control. In *International Conference on Learning Representations*, 2024.

A. Jaegle, F. Gimeno, A. Brock, O. Vinyals, A. Zisserman, and J. Carreira. Perceiver: General perception with iterative attention. In *International Conference on Machine Learning*, 2021.

V. A. Kich, S. Yamamori, J. C. de Jesus, and J. Morimoto. Action-conditioned transformers for decentralized multi-agent world models. *Transactions on Machine Learning Research*, 2026.

V. A. Kich, S. Yamamori, and J. Morimoto. Scalable multi-agent contrastive reinforcement learning. In *Advances in Neural Information Processing Systems*, 2026.

T. Kipf, E. van der Pol, and M. Welling. Contrastive learning of structured world models. In *International Conference on Learning Representations*, 2020.

I. Kostrikov, A. Nair, and S. Levine. Offline reinforcement learning with implicit Q-learning. In *International Conference on Learning Representations*, 2022.

J. Liu, Y. Zhong, S. Hu, H. Fu, Q. Fu, X. Chang, and Y. Yang. Maximum entropy heterogeneous-agent reinforcement learning. In *International Conference on Learning Representations*, 2024.

R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. In *Advances in Neural Information Processing Systems*, 2017.

C. Nimonkar, S. Shah, C. Ji, and B. Eysenbach. Self-supervised goal-reaching results in multi-agent cooperation and exploration. *arXiv preprint arXiv:2509.10656*, 2025.

F. A. Oliehoek and C. Amato. *A Concise Introduction to Decentralized POMDPs*. SpringerBriefs in Intelligent Systems. Springer, 2016.

A. van den Oord, Y. Li, and O. Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

X. B. Peng, A. Kumar, G. Zhang, and S. Levine. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *arXiv preprint arXiv:1910.00177*, 2019.

T. Rashid, M. Samvelyan, C. Schroeder de Witt, G. Farquhar, J. Foerster, and S. Whiteson. QMIX: Monotonic value function factorisation for deep multi-agent reinforcement learning. In *International Conference on Machine Learning*, 2018.

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(7839):604–609, 2020.

C. Tallec, L. Blier, and Y. Ollivier. Making deep Q-learning methods robust to time discretization. In *International Conference on Machine Learning*, 2019.

K. Wang, I. Javali, M. Bortkiewicz, T. Trzciński, and B. Eysenbach. 1000 layer networks for self-supervised RL: Scaling depth can enable new goal-reaching capabilities. In *Advances in Neural Information Processing Systems*, 2025.

M. Wen, J. Kuba, R. Lin, W. Zhang, Y. Wen, J. Wang, and Y. Yang. Multi-agent reinforcement learning is a sequence modeling problem. In *Advances in Neural Information Processing Systems*, 2022.

Z. Xiong, B. Chen, S. Huang, W.-W. Tu, Z. He, and Y. Gao. MQE: Unleashing the power of interaction with multi-agent quadruped environment. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2024.

G. Yang, E. J. Hu, I. Babuschkin, S. Sidor, X. Liu, D. Farhi, N. Ryder, J. Pachocki, W. Chen, and J. Gao. Tuning large neural networks via zero-shot hyperparameter transfer. In *Advances in Neural Information Processing Systems*, 2021.

C. Yu, A. Velu, E. Vinitsky, J. Gao, Y. Wang, A. Bayen, and Y. Wu. The surprising effectiveness of PPO in cooperative multi-agent games. In *Advances in Neural Information Processing Systems Datasets and Benchmarks Track*, 2022.

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*, 2025. arXiv:2406.15836.

Z. Zhao, H. Guo, S. Chen, K. Xu, B. Jiang, Y. Zhu, and D. Zhao. Empowering multi-robot cooperation via sequential world models. In *International Conference on Learning Representations*, 2026. arXiv:2509.13095.

C. Zheng, R. Salakhutdinov, and B. Eysenbach. Contrastive difference predictive coding. In *International Conference on Learning Representations*, 2024.

Y. Zhong, J. G. Kuba, X. Feng, S. Hu, J. Ji, and Y. Yang. Heterogeneous-agent reinforcement learning. *Journal of Machine Learning Research*, 25, 2024.

APPENDICES

A Tasks and agent interface **13**
A.1 Tasks . . . 13
A.2 Observations and actions . . . 13
A.3 Locomotion policy . . . 13

B Training and evaluation protocol **14**
B.1 Training and evaluation . . . 14
B.2 Networks and world-model losses . . . 14

C Baselines and reference methods **14**
C.1 Monolithic-Q baselines . . . 14
C.2 The dense reward . . . 14
C.3 Reference methods . . . 15

D Hyperparameters and compute **16**
D.1 Hyperparameters . . . 16
D.2 Compute . . . 17

E Per-seed results and training curves **18**

F Further ablations **19**
F.1 The imagined term . . . 19
F.2 Exploration noise . . . 20
F.3 Logsumexp penalty . . . 20
F.4 Hindsight slices . . . 20
F.5 Imagined rollout . . . 21

G What the critic represents **21**
G.1 Value maps . . . 21
G.2 The imagined term on one state . . . 22

H Teams of three to five robots **22**

I Real-robot experiments **23**
I.1 Setup . . . 23
I.2 Results . . . 23
I.3 A simulated twin of every trial . . . 25

AI use statement **25**

A TASKS AND AGENT INTERFACE

A.1 TASKS

Table 3 defines the nine tasks. Distances are in meters, and x points along the direction of travel. Goals marked per episode are resampled at every reset.

Table 3: The nine tasks, their goals and their success predicates.

Task	Desired goal of each robot	Success	Episode
Gate	past the gate, x=5.0	both robots at $x > 3.75$	15 s
Corridor	the other robot's start, x=0.6 or x=5.4	both within 0.5 of their own goal	15 s
Door	past the door, x=4.5	both robots at $x > 4.0$	25 s
Coverage	its current of three waypoints, per episode	both at their third, within 0.35	40 s
Herding	the evader past the barrier, x=7.2	the evader at $x > 6.0$	15 s
PushBox	the box at its target, x=4.0, y=0.73	box within 0.3 of its target	15 s
PushBox-Rel	0.3 behind the box and 0.3 to one side, a different side for each robot, and the box at its target	box within 0.3 of its target	15 s
Rendezvous	its own target across the other robot's path, per episode	both within 0.35 of their own goal	15 s
Rendezvous-Seq	its current of two waypoints, per episode	both at their second, within 0.35	15 s

Corridor forces the robots to take turns, because a robot in the gap blocks the other. Coverage interleaves the two waypoint sequences so that the paths cross three times. In Herding the achieved goal is the evader's position. The evader is a third quadruped on the same gait that turns away from the team's centroid and walks off whenever the team comes near, so the direction it flees in depends on where both robots stand.

A robot's goal concatenates its own position with the positions of the task's objects, so a navigation goal contains only the robot's position. In PushBox these terms are absolute, and a robot that first walks behind the box moves away from its goal. PushBox-Rel instead uses the robot-to-box and box-to-target vectors $[\, b - p^j,\ b^* - b \,]$ as the achieved goal and $[\, \delta(\hat{e}_x + s_j \hat{e}_y),\ 0 \,]$ as the desired goal, for robot position p^j, box position b and box target b^*, with $\delta = 0.3$ and a sign $s_j = \pm 1$ that differs between the two robots. The same maneuver then reduces the goal error throughout.

A.2 OBSERVATIONS AND ACTIONS

Each robot observes its planar position, body quaternion, yaw-frame linear velocity, angular velocity, projected gravity, last action and goal error $g_*^j - g_t^j$, and each teammate's position relative to its own in the task frame. For each landmark it also receives the offset in its own heading frame and the offset's magnitude. Landmarks are aperture centers and edges, object centers with their corners or endpoints, and target positions. On PushBox and PushBox-Rel the box's orientation enters as the sine and cosine of twice its yaw relative to the robot's heading, because the box is symmetric under a half turn. The observation contains no map, occupancy grid or exteroception. Gate, Corridor and Door omit the one-hot agent identity for every method, while the other six tasks include it.

An action in $[-1, 1]^3$ is scaled to a base-velocity command with $v_x \in [-2, 2]$ m/s and $v_y \in [-0.5, 0.5]$ m/s. On Herding, PushBox and PushBox-Rel the yaw rate is limited to ± 0.5 rad/s. On the other tasks yaw is disabled and a stabilizer holds the heading within $\pm 1^\circ$ of its initial value.

A.3 LOCOMOTION POLICY

Velocity tracking alone admits a gait that drags the trunk at about a third of the stance height while still following the command, and position-based success predicates cannot tell it from walking. The frozen gait is therefore trained with a base-height target, contact penalties on every non-foot body, a foot air-time term and termination below a minimum height, and we check its stance height before using it.

B Training and evaluation protocol

B.1 Training and evaluation

MACWM, MACRL, ICRL, MA-SAC+HER and MA-GCIQL train for 651 iterations of 96-step rollouts in 64 parallel environments, 4M environment steps in total, and are evaluated every 25 iterations. An evaluation reports the fraction of environments in which the success predicate holds at any step, and the reported score averages the last five evaluations. The code, including the nine benchmark tasks, will be released upon acceptance, together with the logs of the real-robot trials of Appendix I.

B.2 Networks and world-model losses

The state-action and goal encoders are residual MLPs with LayerNorm whose blocks are scaled by $1/\sqrt{n_b}$ for n_b blocks, following μP (Yang et al., 2021). We use that scaling at one width and do not transfer hyperparameters across widths, which is what μP is for. The robots share one tanh-Gaussian actor. Each world-model step adds the output of a Perceiver-style aggregator (Jaegle et al., 2021), with one learned query per robot, to the embedded latents and actions. A transformer across robots then computes the residual update of Eq. 6. We divide $\mathcal{L}_{\text{CPC}}$ and $\mathcal{L}_{\text{obs}}$ by $\sum_k \rho^{k-1}$ and $\mathcal{L}_{\text{VE}}$ by K. Figure 7 shows how the three losses are wired.

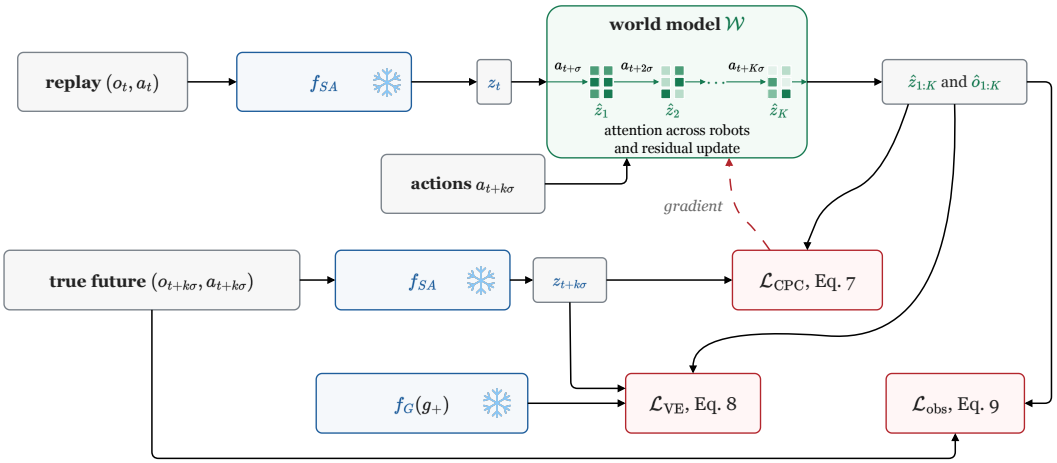


Figure 7: World-model losses. No gradient reaches the critic's encoders.

C Baselines and reference methods

C.1 Monolithic-Q baselines

MA-GCIQL is the stronger of the two monolithic-Q comparators at 0.15 against 0.09, but it is better on only 31 of the 72 runs and ends below 0.05 on 38 of them, against 36 for MA-SAC+HER. It follows implicit Q-learning, which was designed for a fixed dataset and runs online here. Its value head uses expectile regression at $\tau = 0.7$, its Q bootstraps from that value, and its advantage-weighted actor never backpropagates through Q.

C.2 The dense reward

The dense-reward variants of MAPPO, IPPO, HASAC and SeqWM receive one team reward that follows the structure of MQE's reward wrappers (Xiong et al., 2024)[1]. MQE provides wrappers for

[1] github.com/ziyanx02/multiagent-quadruped-environment

Table 4: The dense team reward, one term per line. The first three apply to every task and the last two only to the tasks named.

Term	ϕ_k	w_k	Applies to
Progress	$\sum_j \mathrm{clip}\big(d_{j,t-1} - d_{j,t},\, -0.15,\, 0.15\big)$, and zero on the last step	2	every task
Distance	$\sum_j d_{j,t}$	-0.01	every task
Success	1 once the team solves the task, else 0	2	every task
Crowding	$\sum_j 1/\max(m_{j,t},\, 0.15)$, over robots with $m_{j,t} < 0.5$	-0.1	Gate, Door
Approach	$\sum_j \mathrm{clip}\big(\lVert p_t^j - b_t \rVert - 0.8,\, 0,\, 2\big)$	-0.05	PushBox

only three tasks, so we apply their structure to all nine. The reward is a weighted sum of five terms,

$$r_t \;=\; \sum_k w_k\, \phi_k(s_t),$$

listed one per line in Table 4. There $d_{j,t} = \lVert g_t^j - g_*^j \rVert$ is robot j's distance to its pursued goal, $m_{j,t}$ is the squared distance from robot j to its nearest teammate, p_t^j is the robot's position and b_t the box position. On Gate and Door the progress target is the aperture until a robot has passed it and the goal afterwards, following MQE's gate wrapper.

C.3 Reference methods

HASAC and SeqWM log the outcome of every training episode instead of running periodic evaluations, so we apply the success predicate to these episodes, group them by environment step into 60 to 85 bins and average the last five bins. These scores include exploration noise and are therefore conservative.

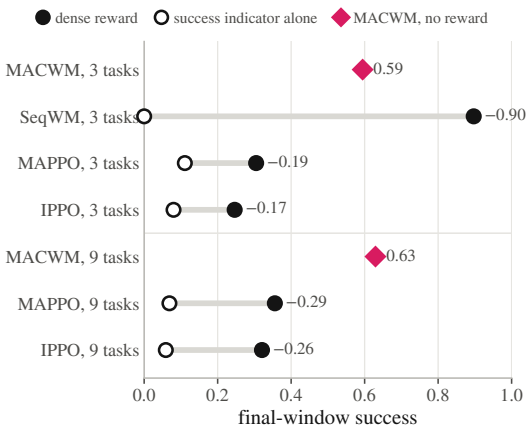


Figure 8: Final-window success of each method with the dense reward and with the success indicator alone, and of MACWM, which receives no reward.

Table 5 lists the reference methods over four seeds. Figure 8 trains each method twice, with the dense reward and with the success indicator alone. SeqWM's sparse variant ran on Gate, PushBox and Coverage, so MAPPO and IPPO appear both on those three tasks and on all nine, and both SeqWM variants use its decentralized mode so that the pair differs only in the reward. MACWM, which receives no reward, scores 0.63 over the nine tasks, above MAPPO and IPPO even with the dense reward. Removing the dense reward costs MAPPO 0.29 and IPPO 0.26 over the nine tasks, and SeqWM never succeeds without it: none of its twelve sparse runs, each with more than five thousand training episodes, met the success predicate.

Table 5: Methods that receive a hand-designed dense reward, reported as a reference point.

Task	HASAC	SeqWM (SEQ)	SeqWM (DEC)	MAPPO	IPPO
Gate	0.970	0.973	0.976	0.281	0.282
Corridor	0.997	0.986	0.988	0.377	0.398
Door	0.472	0.971	0.724	0.190	0.147
Coverage	0.893	0.740	0.791	0.140	0.080
Herding	0.904	0.807	0.827	0.252	0.173
PushBox	0.961	0.933	0.923	0.493	0.377
PushBox-Rel	0.966	0.946	0.929	0.439	0.337
Rendezvous	0.981	0.914	0.913	0.506	0.507
Rendezvous-Seq	0.984	0.944	0.940	0.522	0.584
Mean	0.903	0.913	0.890	0.356	0.321

D HYPERPARAMETERS AND COMPUTE

D.1 HYPERPARAMETERS

Tables 6 to 9 list the settings of every method in Section 4.

Table 6: Settings shared by every method.

Setting	Value
Parallel environments	64
Environment-step budget	4M per run
Seeds	1000 to 8000 in steps of 1000 (four for the reference methods)
Control frequency	50 Hz, 3-D velocity command to a frozen gait
Agent-identity one-hot	six tasks, off on Gate, Corridor and Door

Table 7: Settings shared by MACWM, MACRL, ICRL, MA-SAC+HER and MA-GCIQL.

Setting	Value
Iterations $\times$ rollout length	651×96 steps
Gradient updates per iteration	155
Batch size	256
Replay buffer	last 128 rollouts, first in first out
Hindsight slices concatenated (S)	8, giving a 768-step reachable offset
Discount γ	0.99
Optimizer, learning rate	Adam, 3×10^{-4} for every network and the temperature
Encoder width and depth	256, 4 residual blocks
Actor width and depth	256, 4 residual blocks
Entropy temperature, target entropy	automatic, -0.5 per action dimension
Evaluation interval	every 25 iterations
Reported statistic	mean of the final five evaluations

Table 8: Settings that differ between MACWM, MACRL, ICRL, MA-SAC+HER and MA-GCIQL. A dash marks an absent component, and a blue method name links to its official code.

Setting	MACWM	MACRL	ICRL	MA-SAC+HER	MA-GCIQL
Critic	factored	factored	per-robot	$Q(s,g,a)$, TD(0)	$Q(s,g,a)$, TD(0)
Latent dimension	64	64	64	–	–
Logsumexp penalty λ_{lse}	0.1	0.1	0.1	–	–
Target network, Polyak τ	–	–	–	0.005	0.005
World model: layers, heads, width	4, 4, 128	–	–	–	–
Observation head, horizon K_o, weight λ_o	linear, 15, 1.0	–	–	–	–
Imagined-term weight η	0.5	–	–	–	–
Imagination horizon K, model stride σ	5, 1	–	–	–	–
AC-CPC step weight ρ	0.75	–	–	–	–
Value-equivalence weight λ_{VE}	0.5	–	–	–	–
Exploration noise	pink, β=1	Gaussian	Gaussian	Gaussian	Gaussian
Training reward	none	none	none	self-labeled	self-labeled
Relabeled-goal radius	–	–	–	0.5	0.5
Expectile τ	–	–	–	–	0.7
AWR temperature β, weight clip	–	–	–	–	3.0, 100

Table 9: Settings of the reward-based reference methods. A blue method name links to its official code.

Setting	Value
MAPPO and IPPO, our PPO implementation. IPPO uses per-robot critics.	
Iterations × rollout length	977×64 steps
Epochs per iteration, minibatches	5, 4
Learning rate	3×10^{-4}
Clip ratio	0.2
Discount γ, GAE λ	0.99, 0.95
Entropy coefficient	0.01
Hidden width	256
Evaluation interval	every 40 iterations
HASAC, HARL (Zhong et al., 2024) with its continuous-control settings.	
Hidden sizes	$[256, 256]$
Actor and critic learning rate	5×10^{-4}
Discount γ	0.99
Replay buffer, batch size	10^6, 1000
Polyak τ	0.005
n-step returns	20
Entropy temperature	automatic
Huber loss, value normalization	off, off
Parallel environments	64
SeqWM, authors' released configuration in both SEQ and DEC modes.	
Planner settings	authors' released values
Learning rate, batch size	5×10^{-4}, 1000
Hidden sizes	$[256, 256]$
n-step returns	20
Updates per training step	16
Warm-up steps	50k

D.2 COMPUTE

Table 10 reports the wall-clock time of one training run on one GPU, an NVIDIA RTX PRO 6000 Blackwell, from the simulator's elapsed-time counter. MACWM is the most expensive reward-free method, at a median of 8.2 hours against about 3 for the others, because it trains the world model and rolls it out in every actor update. Its throughput varies by a factor of 1.1 across hosts while every other method's varies by three to four, so contention between runs narrows the gap in these times. HASAC and SeqWM run in their own frameworks, whose overhead their times include.

Table 10: Wall-clock time of one training run of 4M environment steps on one GPU, as the median run.

Method	Runs	Median run (h)
MACWM	72	8.23
MA-GCIQL	72	3.31
MACRL	72	3.05
MA-SAC+HER	72	3.03
ICRL	72	2.97
SeqWM (DEC)	36	26.84
SeqWM (SEQ)	36	26.44
HASAC	36	2.45
MAPPO and IPPO	144	0.83

E PER-SEED RESULTS AND TRAINING CURVES

Table 11 lists the final-window success of every seed behind Table 2, and Figure 9 shows success over training. MA-SAC+HER's best single evaluation reaches 0.97 on Gate, while its mean final-window success stays below 0.3 on every task.

Table 11: Per-seed final-window success of every reward-free and communication-free method, over eight seeds. A value is the mean of the last five periodic evaluations of that run.

Task	seed 1	seed 2	seed 3	seed 4	seed 5	seed 6	seed 7	seed 8
MACWM								
Gate	0.997	0.981	0.784	0.912	0.997	1.000	1.000	1.000
Corridor	0.044	0.966	0.197	0.862	0.025	0.022	0.891	0.672
Door	0.716	0.450	0.669	0.578	0.625	0.075	0.878	0.719
Coverage	0.350	0.672	0.216	0.606	0.284	0.353	0.125	0.313
Herding	0.878	0.700	0.744	0.559	0.481	0.809	0.785	0.828
PushBox	0.556	0.700	0.709	0.509	0.425	0.456	0.057	0.266
PushBox-Rel	0.969	0.475	0.831	0.834	0.853	0.581	0.559	0.625
Rendezvous	0.703	0.572	0.672	0.703	0.763	0.669	0.675	0.819
Rendezvous-Seq	0.747	0.491	0.381	0.837	0.747	0.822	0.656	0.900
MACRL								
Gate	0.950	0.928	1.000	0.975	0.994	0.972	0.881	0.987
Corridor	0.744	0.000	0.862	0.800	0.672	0.972	0.334	0.009
Door	0.144	0.719	0.559	0.412	0.012	0.059	0.441	0.534
Coverage	0.122	0.094	0.278	0.403	0.334	0.228	0.475	0.338
Herding	0.306	0.194	0.769	0.625	0.622	0.328	0.128	0.188
PushBox	0.000	0.578	0.366	0.619	0.000	0.625	0.000	0.609
PushBox-Rel	0.612	0.747	0.647	0.847	0.784	0.841	0.831	0.844
Rendezvous	0.759	0.781	0.853	0.644	0.728	0.619	0.716	0.759
Rendezvous-Seq	0.741	0.587	0.959	0.844	0.875	0.981	0.791	0.884
ICRL								
Gate	0.897	0.803	0.991	0.953	0.981	1.000	0.962	0.984
Corridor	0.597	0.019	0.472	0.234	0.762	0.759	0.794	0.659
Door	0.231	0.063	0.350	0.316	0.197	0.844	0.540	0.000
Coverage	0.188	0.491	0.616	0.250	0.147	0.084	0.216	0.328
Herding	0.418	0.535	0.153	0.606	0.503	0.878	0.272	0.737
PushBox	0.056	0.000	0.000	0.403	0.044	0.000	0.000	0.187
PushBox-Rel	0.909	0.884	0.619	0.859	0.766	0.891	0.581	0.806
Rendezvous	0.778	0.700	0.672	0.706	0.784	0.765	0.735	0.875
Rendezvous-Seq	0.763	0.803	0.903	0.435	0.681	0.685	0.697	0.909
MA-SAC+HER								
Gate	0.341	0.622	0.000	0.194	0.528	0.297	0.147	0.203
Corridor	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Door	0.012	0.000	0.016	0.000	0.000	0.059	0.078	0.000
Coverage	0.022	0.013	0.044	0.013	0.019	0.094	0.019	0.038
Herding	0.000	0.000	0.000	0.075	0.000	0.159	0.194	0.006
PushBox	0.000	0.013	0.016	0.031	0.000	0.012	0.000	0.003
PushBox-Rel	0.128	0.141	0.091	0.081	0.013	0.100	0.078	0.078
Rendezvous	0.097	0.112	0.094	0.103	0.118	0.156	0.131	0.134
Rendezvous-Seq	0.356	0.228	0.075	0.103	0.159	0.094	0.194	0.278
MA-GCIQL								
Gate	0.319	0.000	0.847	0.000	0.194	0.010	0.813	0.759
Corridor	0.000	0.172	0.153	0.025	0.000	0.000	0.000	0.000
Door	0.000	0.000	0.000	0.000	0.037	0.000	0.000	0.000
Coverage	0.066	0.041	0.028	0.044	0.003	0.044	0.010	0.037
Herding	0.287	0.000	0.331	0.228	0.362	0.000	0.391	0.572
PushBox	0.003	0.000	0.000	0.000	0.059	0.000	0.131	0.000
PushBox-Rel	0.000	0.000	0.525	0.050	0.000	0.041	0.185	0.000
Rendezvous	0.275	0.188	0.328	0.128	0.422	0.119	0.406	0.403
Rendezvous-Seq	0.159	0.181	0.169	0.075	0.775	0.134	0.103	0.203

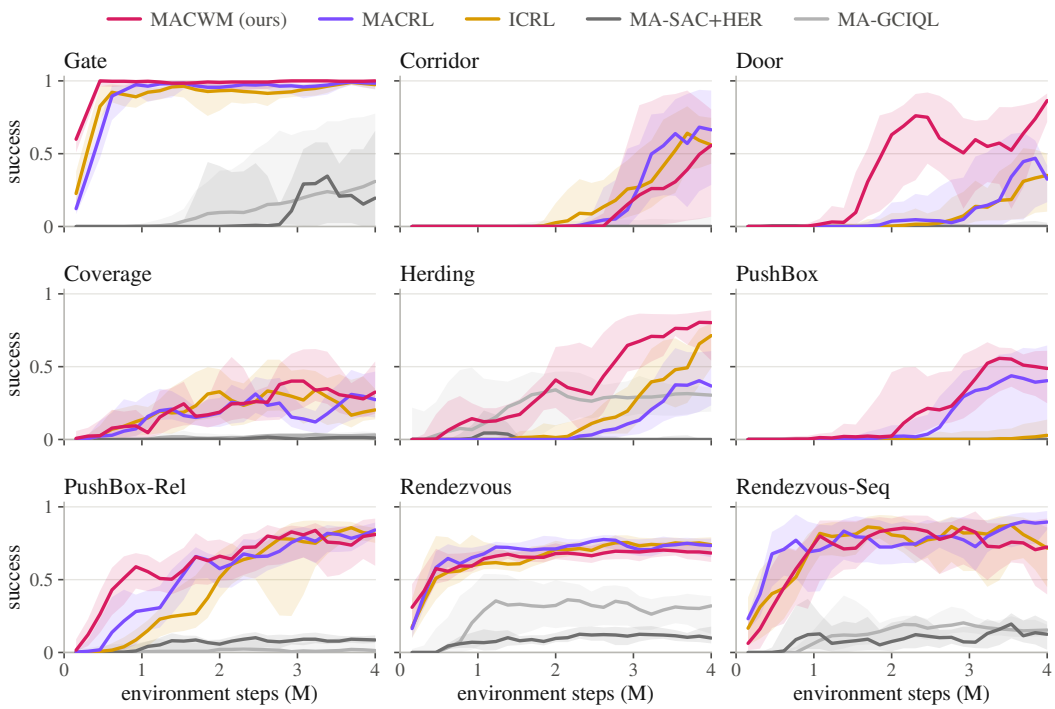


Figure 9: Success over training on each task, as the median over eight seeds with the interquartile range shaded, smoothed over three evaluations.

F Further ablations

Every ablation changes one setting of MACWM and is paired with it by task and seed, where MACWM's mean is 0.65. MACRL and ICRL differ only in whether the critic is factored or per-robot, so Table 2 already compares the two critics. A joint critic over the whole team cannot serve MACWM, because it produces no per-robot latents for the world model to predict.

F.1 The imagined term

Setting $\eta = 0$ in Eq. 10 leaves the world model trained but disconnects it from the actor, and 20 of the 36 paired runs improve with the term. On PushBox, halving it to $\eta = 0.25$ still costs 0.28. Figure 10a pools the three tasks run at all three weights.

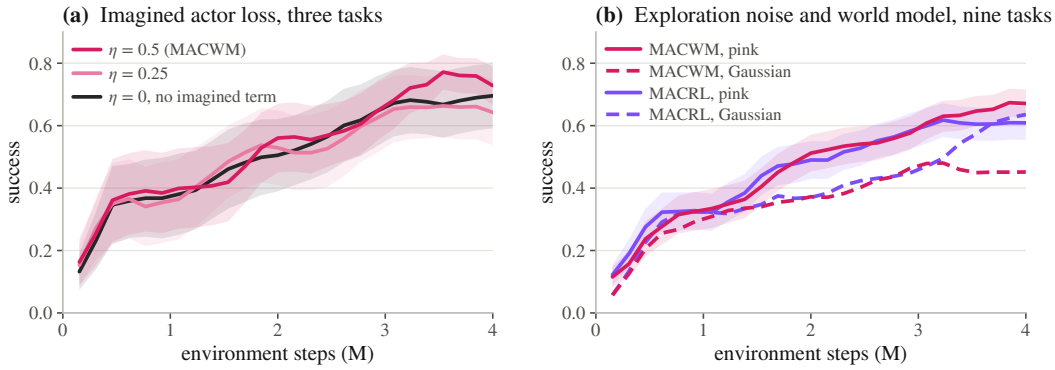


Figure 10: (a) The imagined weight η on Gate, Herding and PushBox, as mean and standard error over four seeds. (b) Exploration noise crossed with the world model, over nine tasks and four seeds.

F.2 Exploration noise

Giving pink noise to MACRL crosses the noise with the world model (Figure 10b). Both gains of Section 4.3 have a paired standard error of 0.06, and their interaction, 0.19 (0.07), is positive on eight of the nine tasks and on 27 of the 36 paired runs. Without Corridor, where MACRL loses the task under the noise, it is 0.11 with a bootstrap interval over tasks of $[0.04, 0.18]$. With Gaussian noise MACWM falls below MACRL, at 0.45 against 0.60, and its gain from the noise is largest on Door (0.51) and Herding (0.35).

Table 12 crosses the noise with the imagined term alone. The term lowers success by 0.07 under Gaussian noise and raises it by 0.03 under pink noise, an interaction of $+0.10$ with a standard error of 0.06, not separable from zero at four seeds. Over the last fifth of training the world model's contrastive accuracy is 0.04 higher under pink noise (standard error 0.007), while the critic's does not change (-0.01 ± 0.01), so under Gaussian noise the imagined term follows a less accurate model. Without the imagined term, MACWM under Gaussian noise scores 0.52, still 0.08 below MACRL (paired standard error 0.04).

Table 12: Pink noise crossed with the imagined actor loss, as mean final-window success over nine tasks and four seeds.

	Gaussian noise	Pink noise
No imagined term ($\eta = 0$)	0.521	**0.629**
Imagined term ($\eta = 0.5$)	0.451	**0.655**

F.3 Logsumexp penalty

Without the logsumexp penalty, Gate loses 0.01 and 0.03, and the six runs on the other three tasks end at or below 0.01. The InfoNCE loss is unchanged and contrastive accuracy falls only from 0.31 to 0.27, so the penalty preserves the value scale that the actor climbs (Section 3.2). We set $\lambda_{\text{lse}} = 0.1$ from a search with two seeds per task.

F.4 Hindsight slices

With hindsight offsets restricted to one 96-step slice instead of $S = 8$ (Section 3.3), the loss follows episode length: Gate loses 0.05, Door 0.58, PushBox 0.51 and Coverage 0.33. Three of those runs never reach the goal at all, where the same seed with eight slices first succeeds between 1.4M and 2.3M environment steps.

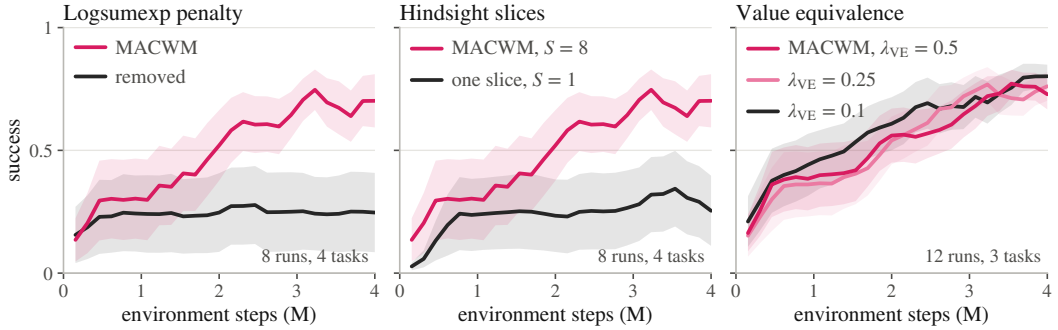


Figure 11: Three further design choices over training, each comparing MACWM with the same configuration with one setting replaced, over the task and seed pairs that have both runs. Shading is one standard error.

F.5 IMAGINED ROLLOUT

Rolling the world model out for $K = 10$ steps at stride $\sigma = 4$, eight times the default span, changes mean success by $+0.01$ over eight tasks (paired standard error 0.06), although PushBox falls from 0.62 to 0.14. Re-querying the actor at each imagined observation, as Dreamer does (Hafner et al., 2025), lowers Door from 0.41 to 0.00 and PushBox from 0.63 to 0.04 on an earlier configuration and costs 1.3 times the wall-clock time, so the rollout stays open-loop. On Gate, Herding and PushBox, lowering λ_{VE} to 0.25 or 0.1 changes mean success by less than its standard error, while halving η costs more. We keep both weights at 0.5.

G WHAT THE CRITIC REPRESENTS

G.1 VALUE MAPS

We sweep the critic's energy, the quantity the actor climbs, over the arena,

$$V(z, g) \;=\; -\left\| \tfrac{1}{N} \textstyle\sum_j z^j - f_G(g) \right\|_2, \qquad z^j = f_{SA}(o^j, \bar{a}^j), \tag{11}$$

with the goal fixed, $\bar{a}^j$ the trained actor's action at that observation, and the robots pooled as in Eq. 3.

On Gate and Door the whole team is placed at each cell, and on Herding and PushBox the evader or the box, with the robots staged behind it. Observations are read from the simulator as in training, and cells inside a wall are left out. The inset of Figure 1 draws MACWM's Door map on the rendered floor, with each cell left out given the nearest value. Each panel is scaled to its own second and ninety-eighth percentile, because the InfoNCE temperature and the logsumexp penalty set the energy's magnitude, so only the shape is comparable across panels. The critics do not represent distance to the goal (Figure 12). On the three tasks with a barrier, value rises through the opening

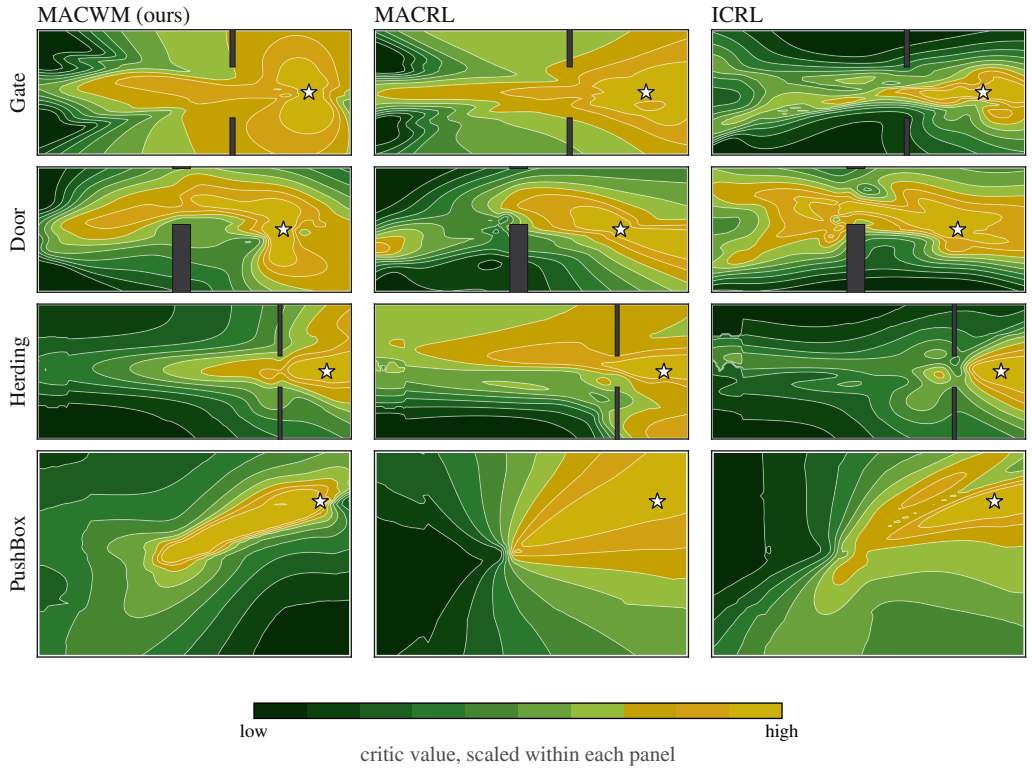


Figure 12: Critic value from Eq. 11 with the goal fixed, in ten bands scaled within each panel. The team is swept on Gate and Door, the evader on Herding and the box on PushBox, and the star marks the goal.

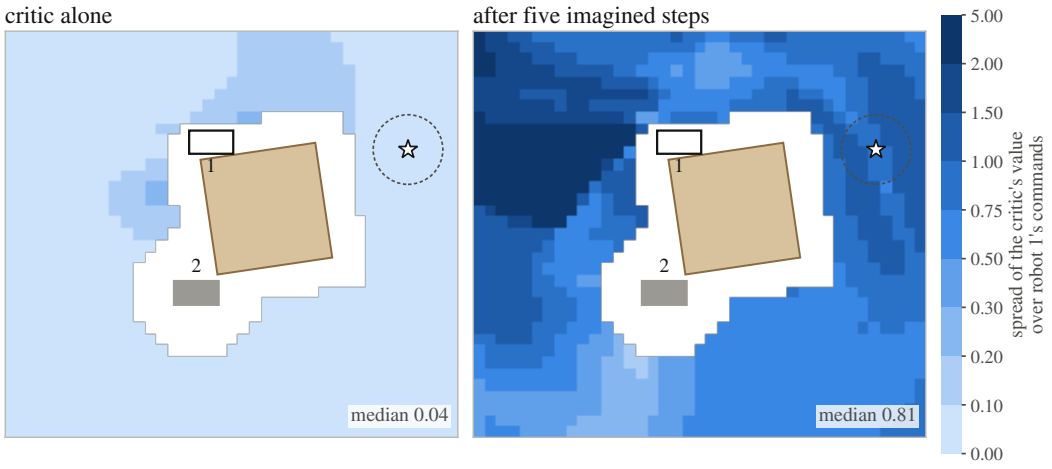


Figure 13: Spread of the critic's value over robot 1's commands within 0.5 of the actor's own, with robot 1 placed on every 10 cm cell and the rest held at the state of Figure 4. The outlined cell is robot 1's true position and the dashed circle the target tolerance.

instead of forming level sets around the goal, and on PushBox it is organized along the line from the box to its target. The maps come from one seed per method and show qualitative structure only.

G.2 The imagined term on one state

Figure 4 uses MACWM's PushBox seed 3 (Table 11) at 1.28 s, with robot 1 pushing the box. Each imagined step lasts 0.02 s, every other command stays at the actor's mean, and commands are scored at re-encoded latents. Within the dashed square the span is 1.51, 1.37, 1.28, 1.24 and 1.26 after one to five imagined steps. Commands far outside the square lie beyond what the world model was trained on.

Figure 13 repeats the probe over space, with robot 1 placed on every 10 cm cell around the box. The median spread of the value over robot 1's commands within 0.5 of the actor's own rises from 0.04 for the critic alone to 0.81 after five imagined steps, and imagination at least doubles the spread in every cell.

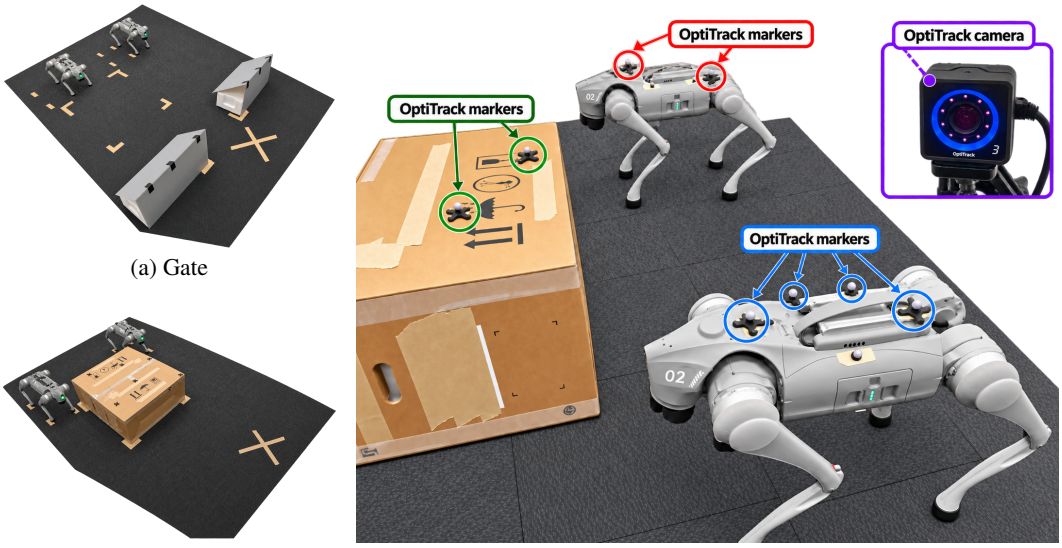
On Door, with seed 7 at 2.5 s into an episode that succeeds, the same probes give a span of 0.11 for the critic alone and 0.41 to 0.77 after one to five steps, and median spreads of 0.09 and 0.51. Each probe uses one state of one seed and illustrates the mechanism without measuring its size.

H Teams of three to five robots

Only each method's input layer grows with the team. On Gate and Door the opening and the episode length stay fixed, the arena grows by 1.5 m of width and 1 m of run-out per robot beyond two, as in MQE (Xiong et al., 2024), and the team succeeds once every robot is past the opening. On PushBox the box mass grows by 3 kg per robot, so the load per robot is constant, and each added robot starts behind the box.

Pooled over three to five robots, MACWM leads MACRL on PushBox by 0.30 (paired standard error 0.06). Over the nine settings MACWM reaches the goal in 35 of 36 runs, MACRL in 30 and ICRL in 27, and MACWM's pooled lead over ICRL is 0.20 (paired standard error 0.04, ahead in 32 of 36).

Widening the world model's Perceiver-style aggregator (Appendix B) fourfold, from width 128 with four heads to 512 with eight, tests whether a larger team outgrows it. Paired against MACWM, its effect ranges from -0.16 to $+0.08$ over the six Gate and Door settings, so a wider aggregator does not help at these team sizes.



(a) Gate

(b) PushBox

(c) Motion capture

Figure 14: The real-robot setup. (a) Gate and (b) PushBox at their start poses. (c) OptiTrack markers on both robots and on the box, and one of the cameras that track them.

I REAL-ROBOT EXPERIMENTS

Each method ran eight trials per task in the room of Figure 14. It deploys its seed with the highest final-window success in simulation, which reaches 0.71 (MACWM), 0.63 (MACRL) and 0.40 (ICRL) on PushBox and 1.00 each on Gate. The two monolithic-Q baselines were not deployed, since their best PushBox seeds succeed in at most 0.13 of simulated episodes.

I.1 SETUP

The policy runs at 50 Hz, and OptiTrack cameras track markers on both robots and the box at 120 Hz (Figure 14c). Replaying the logged observations through the deployed networks reproduces every logged action to within 10^{-5}. For safety the forward speed is clamped to 0.6 m/s, against 2 m/s in simulation, the lateral speed to 0.2 m/s on Gate, and command changes to 1 m/s^2. A trial stops when a robot would leave the tracked area or the robots come too close, and on Gate a robot is parked once it has crossed, because the room is too short for it to walk on. The box measures 1 by 1 by 0.4 m and weighs about 6 kg.

The PushBox room was fitted before any baseline ran. From the simulator's randomized starts, with the room's walls and command limits, the three methods succeed in 0.30, 0.18 and 0.53 of episodes, against 0.72, 0.35 and 0.42 from the hardware start.

I.2 RESULTS

Table 13 lists every outcome. On PushBox MACWM's robots stay at the two rear corners of the box and turn it toward the target (Figure 15). MACRL's box follows the same path at first, but one robot walks past the box's front face and a fence stops every trial within 6 s. ICRL pushes the box almost straight, so it never turns toward the target. On Gate MACWM and MACRL take the robots through the opening almost side by side, 0.28 to 0.31 m apart, and they never touched. ICRL backs one robot up by about half a meter and sends the robots through in turns, so its trials started 0.5 m further forward, without the fences that would have stopped them. A parked Go2 creeps back, so a robot counts once it has crossed the line, and in three of ICRL's trials the first robot had crept just behind it when its partner crossed. Under the simulator's predicate, which needs both robots past the line at once, ICRL therefore succeeds in five of eight.

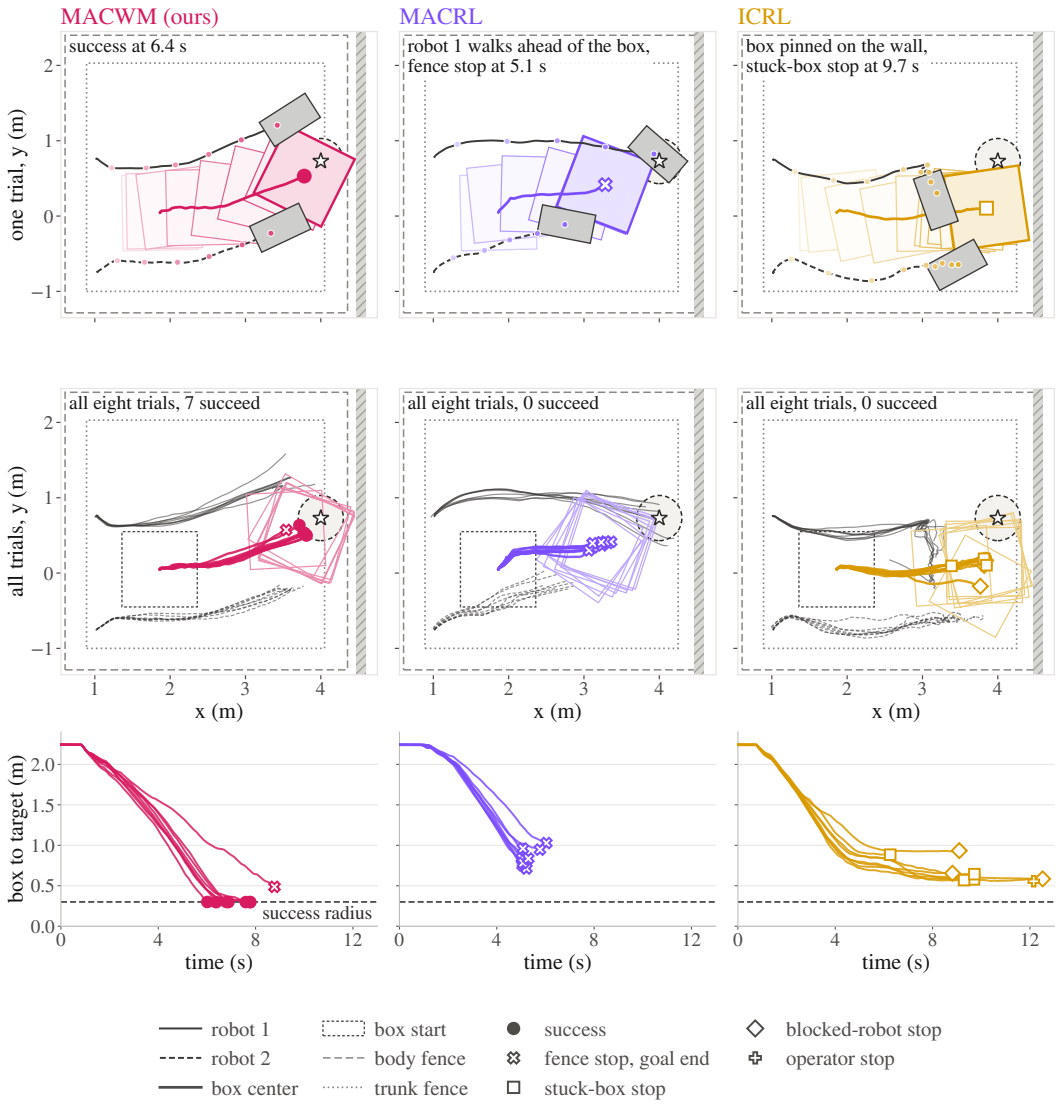


Figure 15: PushBox on hardware, with one trial per method and its box drawn every second (top), all eight trials (middle) and the box's distance to its target (bottom).

Table 13: Wilson 95% intervals and time to success as mean and standard deviation.

Task	Method	Success	Interval	Time (s)
PushBox	MACWM	7/8	0.53 to 0.98	7.0 ± 0.7
	MACRL	0/8	0 to 0.32	–
	ICRL	0/8	0 to 0.32	–
Gate	MACWM	8/8	0.68 to 1	5.8 ± 0.1
	MACRL	8/8	0.68 to 1	5.7 ± 0.1
	ICRL	8/8	0.68 to 1	7.2 ± 0.5

I.3 A SIMULATED TWIN OF EVERY TRIAL

To separate transfer from the room, we started the simulator from each trial's logged start and ran the same policy under the hardware's command limits inside the room's walls, 15 times per start on PushBox and 10 on Gate, with the room's stop rules evaluated but not acting. Stops at the rear fence in the first 0.5 s are ignored, because the simulated robot settles after its pose is set. On PushBox the twin succeeds in 0.67 of episodes from MACWM's starts, 0.03 from MACRL's and 0.25 from ICRL's (Figure 16). For MACRL a fence at the goal end is the first stop in 117 of 120 episodes, as on the hardware. At ICRL's twin rate, eight failures in eight trials have a probability of about 0.10, so its hardware result is weak evidence that it transferred worse than MACWM. On Gate every twin episode succeeds for all three methods.

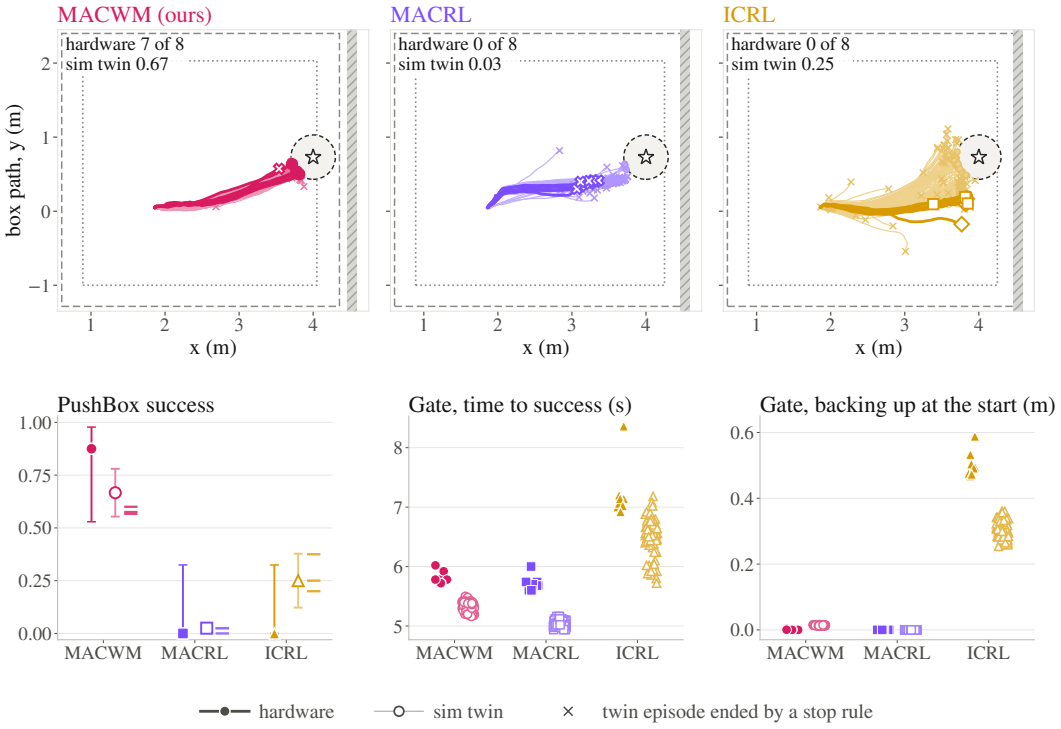


Figure 16: Hardware trials against their simulated twins, as PushBox box paths from each method's own eight starts (top) and three comparisons with 95% intervals (bottom). Short bars mark three evaluations of the twin that count every stop at the rear fence.

AI USE STATEMENT

Generative AI is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigor, or originality of the research. We take responsibility for the final content of this work, including text, claims and artifacts produced with the aid of generative AI.