BRIDGING THE DECENTRALIZED-EXECUTION GAP IN MULTI-AGENT REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Centralized training with decentralized execution (CTDE) is the standard paradigm for cooperative multi-agent reinforcement learning, but partial observability can lead to a substantial performance gap compared to fully centralized approaches. We identify two training-execution mismatches in standard CTDE: Actors are trained with signals informed by centralized information that is unavailable at decentralized execution, while policy updates ignore the fact that the centralized critic signal's reliability can vary across agent-and-time samples under partial observability and decentralized execution. We introduce Decentralized-Execution-Aligned Learning (DEAL) to address these mismatches. DEAL uses a *decentralized predictive representation* to ground actor-side learning by training each recurrent state to predict the agent's next observation using a conditional mixture model capturing the ambiguity induced by hidden context associated with decentralized operation. DEAL further uses *reliability-weighted critic signals* to predict the collection-time residual scale from critic features and adjust each sample's relative influence on the policy update. Empirically, across 37 tasks from four diverse environments, DEAL attains the best performance in every environment, outperforming strong CTDE and CTCE baselines under the same local policy model complexity budget.

1 INTRODUCTION AND MOTIVATION

In cooperative multi-agent reinforcement learning (MARL), learning to coordinate is the key problem to achieve good joint performance. Centralized training and centralized execution (CTCE) approaches this challenge by using global information and sequential action selection (Wen et al., 2022; Mahjoub et al., 2025), but such centralized execution is often impractical at deployment. Therefore, centralized training with decentralized execution (CTDE) was proposed as a practical compromise and has been the standard approach to cooperative MARL (Lowe et al., 2017; Sunehag et al., 2017; Rashid et al., 2020; Yu et al., 2022). However, CTDE shows inherent performance degradation due to decentralized execution compared to CTCE.

There have been prior works to bridge the gap caused by decentralized execution by transferring centralized information or behavior to decentralized actors, through privileged-information reconstruction (Song et al., 2023; Qifan et al., 2025; Kang et al., 2025) or distillation from centralized executors (Zhao et al., 2024; Li et al., 2026). However, these works have limitations and the performance improvement is limited. Reconstruction of privileged information such as global state or other agents' observations becomes increasingly ambiguous as partial observability, environmental stochasticity, and team size grow. In the case of distillation, a centralized teacher may rely on global context or action dependencies that decentralized actors cannot reproduce and may require centralized execution during training, a stronger assumption than standard CTDE. Basically, these approaches focus on how to estimate, acquire, store or transfer global knowledge at/to local actors. From a MARL design perspective, this effort concerns only the aspect at the actor side; actor shaping is also driven by the critic during centralized training. Consequently, we have two design dimensions, actor design and critic design; both can be exploited for better shaping the actor during centralized training to reduce the decentralized-execution gap.

In this paper we consider a holistic actor and critic design under the CTDE framework to narrow the gap caused by decentralized execution by answering the following questions: "How should one forge

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.

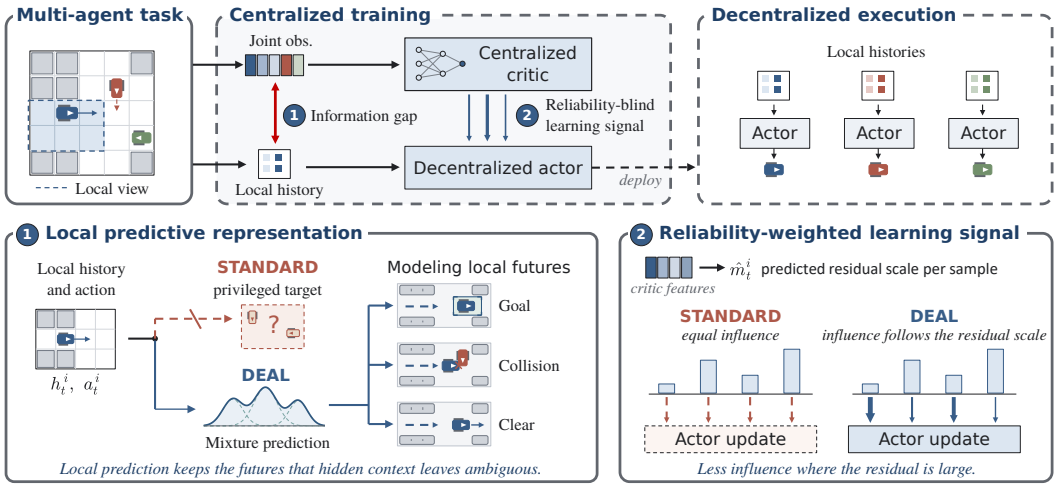


Figure 1: **Two training-execution mismatches and how DEAL aligns them.** Top: centralized training shapes a decentralized actor; at execution each actor sees only its local history. (1) Privileged information is not determined by the local history and action, so DEAL uses a mixture distribution over the agent's own next observations to capture this ambiguity. (2) Critic signal reliability is uneven across samples, yet standard updates weight them evenly; DEAL scales each sample's influence by its predicted residual scale.

the global information into a form grounded in local trajectories at the actor side, and how should one design critic signals for actor updates to better handle partial observability?"

Actor design reducing information loss. Under partial observability, teammate actions and stochastic transitions are not determined by local evidence, and several latent interaction configurations may remain compatible with the same local history and action. For example, in Figure 1 the same local action may lead to reaching the goal, either colliding with an unseen teammate or passing without collision. The occurrence of either event is not fully determined by the current local view window. This ambiguity can induce a *multimodal conditional distribution* over the agent's next observation. Reconstructing pointwise privileged-information target, e.g., other agents' observations as in previous approaches can hence be poorly matched to the actual information available to the actor at execution, as identical local inputs may be paired with conflicting targets. To address this problem, our approach to actor design learns the actor's representation by setting the prediction target as the agent's own next observation, which is locally observable in decentralized trajectories. The key point here is that we model the agent's next observation with a mixture distribution that can represent multiple feasible futures and maintain this mixture distribution during training. In this way, we reduce the information loss at the decentralized actor by modeling multiple plausible local futures.

Reliability-weighted critic signals. A centralized critic uses joint observations to predict return target, while constructing agent-wise representations to capture how global interactions affect each agent. During training, the critic's return predictability can vary across the agents and situations due to decentralized action sampling and stochastic transitions. Therefore, the reliability of learning signals from the critic can vary across agents and situations in decentralized operation. Standard policy optimization objectives do not consider this difference in reliability in critic signals. To address this issue, we predict the expected magnitude of the value residual from each agent's critic representation. We use the predicted residual scale as a proxy for signal reliability and assign weights to critic signals reflecting the reliability. In this way, centralized training based on reliability-weighted critic signals better trains decentralized actors.

By combining the two ideas mentioned above-a decentralized predictive representation with adaptive learning signal weighting, we propose **Decentralized-Execution-Aligned Learning (DEAL)**, which does not require either centralized action selection or communication at execution, but significantly reduces the decentralized-execution gap. We evaluated DEAL against strong CTDE and CTCE baselines on four cooperative benchmarks with ablation studies, and demonstrate that DEAL outperforms the baselines for MARL even including the methods developed under CTCE with the same local policy complexity budget.

2 PRELIMINARIES

We consider a cooperative Dec-POMDP $\mathcal{M} = (\mathcal{S}, \{\mathcal{A}_i\}_{i=1}^N, P, r, \{\Omega_i\}_{i=1}^N, O, \gamma, N)$. Agent i observes o_t^i and acts by $a_t^i \sim \pi_i(\cdot \mid h_t^i)$ from its local history $h_t^i = (o_{0:t}^i)$. All agents share the common reward r_t and maximize $\mathbb{E}[\sum_{t\geq 0} \gamma^t r_t]$. Under CTDE, training may use joint information, but each agent has access only to local information at execution. In our design, each agent summarizes its local history with a recurrent encoder, $b_t^i = f_\psi(h_t^i)$, as shown in the lower part of the actor block of Fig. 2. We call b_t^i the agent's *information feature*.

During centralized training, a critic conditioned on joint information outputs per-agent values V_t^i, $i = 1, \ldots, N$. We assume PPO (Schulman et al., 2017) with generalized advantage estimation (GAE) (Schulman et al., 2016) for actor updates, which uses the values V_t^i, $i = 1, \ldots, N$ to compute advantages $\hat{A}_t^i$, and the clipped surrogate with policy ratio $\rho_t^i = \pi_i^{\text{new}}/\pi_i^{\text{old}}$ for $i = 1, \cdots, N$.

Note that per-agent values V_t^i, $i = 1, \ldots, N$ provide agent-dependent baselines that reflect different roles and interaction contexts of different agents. Thus, centralized critics often learn agent-wise representations from joint information (Yu et al., 2022; Mahjoub et al., 2025). In this paper, we adopt cross-agent self-attention over observation tokens (Iqbal & Sha, 2019) by $x_t^i = \text{Attn}_\phi^i(o_t^{1:N})$, followed by a recurrent encoder f_ϕ^{v} that produces per-agent representations or features $\tilde{o}_t^i = f_\phi^{\text{v}}(x_{0:t}^i)$ from which value estimates V_ϕ^i are computed, as shown in the critic block of Fig. 2. Here, $\tilde{o}_t^i$ can be viewed as a modified observation of agent i incorporating the relationship to other agents' observations for better per-agent value acquisition.

3 METHOD

In this section, we present DEAL, which introduces two components to CTDE actor-critic. First, a locally grounded predictive objective shapes the decentralized actor representation. Second, critic-derived learning signals are reweighted according to their prediction residual scales. Fig. 2 provides an overview of the resulting architecture, with the two components detailed in the following subsections.

3.1 DECENTRALIZED PREDICTIVE REPRESENTATION

DEAL compresses each agent's local history into an information feature $b_t^i = f_\psi(h_t^i)$, where f_ψ is a recurrent encoder. From agent i's perspective, variables that affect its future observation but are not determined by its local history-and-action form the hidden context. This context can include unobserved state variables, teammates' behaviors, or stochastic environment factors. This hidden context is ambiguous from the local perspective. Different hidden contexts can be consistent with the same local history-and-action, while leading to different next observations. After marginalizing over the hidden context, the conditional distribution of the agent's next observation can hence be multimodal, as discussed in Appendix A.4.

Instead of requiring b_t^i to reconstruct such hidden context so-called privileged information as in previous methods, we use the agent's own next observation as the prediction target. This provides locally grounded supervision because the realized next observation is available from agent's decentralized trajectory. Although only one next observation is realized at each transition, multiple next observations can be feasible under the same local history and action when the hidden context is marginalized out. A single point prediction can collapse these different outcomes into one estimate. So, we model the conditional distribution of the next observation using a mixture model that can represent multiple modes. Thus in our method, the ambiguity of hidden context is captured in the multi-modal distribution of the agent's next observation.

Specifically, we model the symlog-transformed next observation z', defined as $z' := z_{t+1}^i = \text{symlog}(o_{t+1}^i)$, with the following conditional Gaussian mixture:

$$q_\theta(z' \mid b_t^i, a_t^i) = \sum_{k=1}^{K} \alpha_k(b_t^i, a_t^i)\, \mathcal{N}\big(z';\ \mu_k(b_t^i, a_t^i),\ \text{diag}\, v(b_t^i)\big)\,, \tag{1}$$

where $\alpha_k \geq 0$, $\sum_k \alpha_k = 1$, and $v(b) = \exp\big(\ell(b)\big)$, with ℓ denoting a log-variance readout of the information state. Here, $\text{symlog}(x) = \text{sign}(x)\log(1 + |x|)$ compresses the observation scale. The

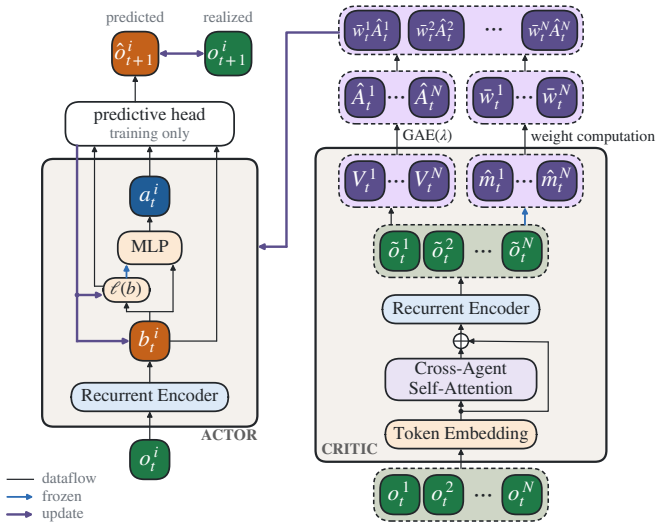


Figure 2: **DEAL architecture.** For the actor, a recurrent encoder compresses each agent's observation history into an information state b_t^i, which conditions the policy directly and through the detached predictive-scale readout $\zeta = \mathrm{sg}[\ell(b)]$. A training-only predictive head implements the decentralized predictive representation by predicting the agent's own next observation. For the critic, cross-agent attention followed by temporal recurrence produces per-agent features $\tilde{o}_t^1, ..., \tilde{o}_t^N$ for value and residual-scale estimation. The resulting GAE advantages are scaled inversely to the predicted residual scales, adaptively allocating each signal's influence on the actor update.

action-conditioned mixture weights and means in (1) allow the model to capture action-dependent, non-Gaussian structure, while the shared diagonal variance $v(b)$ captures residual prediction spread beyond the component means. We condition this scale only on b, making it available before action selection. Then, the prediction head is trained by negative log-likelihood on the next observation,

$$\mathcal{L}_{\text{pred}} = \mathbb{E}_{t,i}\left[-\log q_\theta\left(z_{t+1}^i \mid b_t^i, a_t^i\right)\right], \qquad z_{t+1}^i = \text{symlog}(o_{t+1}^i), \tag{2}$$

The loss is masked at episode boundaries, and its gradients update both the mixture model parameter θ and the recurrent encoder parameter ψ.

Minimizing $\mathcal{L}_{\text{pred}}$ encourages b to retain information predictive of the agent's local future. This information can capture aspects of the global state and other agents' actions that are reflected in the agent's next observation. Under an expressive predictive family, this connection is formalized by the predictive-risk decomposition in Appendix A. Note that a single Gaussian would incur an irreducible misspecification gap when the conditional future is multimodal.

Pre-action readout. Because the shared log-variance depends only on b_t^i, it is available before action selection. The policy receives $\zeta_t^i = \mathrm{sg}[\ell(b_t^i)]$ together with b_t^i. The stop-gradient prevents the policy objective from directly modifying the predictive-scale readout through this input, while policy gradients still update the shared information state b. Thus, ζ provides an explicit predictive feature trained by $\mathcal{L}_{\text{pred}}$, rather than a feature that must be discovered solely through the policy gradient.

3.2 RELIABILITY-WEIGHTED LEARNING SIGNALS

Our centralized critic serves two roles during training: its per-agent features $\tilde{o}_t^i$ (Section 2) are fed *i)* to the value heads to generate values used for GAE and *ii)* to a residual-scale head to generate the adaptive reliability weights. The overall architecture is shown in Fig. 2.

Under full observability of the state, the value function can be defined as $V(s_t) = \mathbb{E}[G_t^\lambda | s_t]$, where G_t^λ is the λ-return, and this function is common to all agents in cooperative tasks with shared rewards. However, such design ignores possibly different credits of agents. Thus, the current approach to the centralized critic design is to generate a separate critic value for each agent. For this, each agent-wise feature $\tilde{o}_t^i$ is first extracted from the joint observation history, incorporating the influence of other agents, and then the individual value function is constructed as a function of $\tilde{o}_t^i$, with

$V^i_\phi(\tilde{o}^i_t) = \mathbb{E}[G^\lambda_t|\tilde{o}^i_t]$, as shown in Fig. 2, where the influence of other agents on agent i is modeled by the cross-agent attention block.

Under partial observability, the critic forms $x^i_t = \mathrm{Attn}^i_\phi(o^{1:N}_t)$ from the joint observation and obtains the recurrent feature $\tilde{o}^i_t = f^{\mathrm{v}}_\phi(x^i_{0:t})$, which is the input of the value head V^i_ϕ. Here, the conditional expectation of λ-return $\mathbb{E}[G^\lambda_t|\tilde{o}^i_t]$ can be different from $\mathbb{E}[G^\lambda_t|s_t]$ depending on how each $\tilde{o}^i_t$ captures the return-relevant state information. Consequently, the reliability of value estimate $V^i_\phi(\tilde{o}^i_t)$ varies across agents and situations. For example, two situations may be well separated in the feature space of one agent but mapped to similar representations in another agent's feature space. If the two situations lead to different returns, the second agent's value estimate has a wider range of value for the similar feature, even though the realized return is the same for all agents in each trajectory. Thus, conditioning the same return on different agent-wise features can yield different levels of value predictability under partial observability. We estimate the reliability of value estimation of each agent and use this reliability level to modulate each sample's influence on the policy update.

Residual-scale head. For each rollout sample, we define

$$d^i_t = \mathrm{sg}[\,|G^{\lambda,i}_t - V^i_\phi(\tilde{o}^i_t)|\,], \tag{3}$$

where $G^{\lambda,i}_t$ is the λ-return evaluated at agent i. The residual d^i_t between $G^{\lambda,i}_t$ and its estimate $V^i_\phi(\tilde{o}^i_t)$ is basically the instantaneous estimation error. Instead of using the instantaneous d^i_t directly, we train a network g_η to predict the average estimation error:

$$\hat{m}^i_t = \mathrm{softplus}\left(g_\eta(\mathrm{sg}[\tilde{o}^i_t])\right) \qquad \mathcal{L}_{\mathrm{res}} = \mathbb{E}_{t,i}\left[\left(\hat{m}^i_t - d^i_t\right)^2\right]. \tag{4}$$

For fixed critic features, the population squared-error objective is minimized by

$$m^\star(\tilde{o}) = \mathbb{E}[d \mid \tilde{o}] = \mathbb{E}[|G^\lambda - V| \mid \tilde{o}].$$

The residual head is trained to approximate this feature-conditioned mean absolute residual.

Reliability adjustment. Let $\mathcal{B}$ denote the set of all valid agent-time samples in the rollout. At the start of each PPO epoch, DEAL converts the current residual-scale predictions into relative reliability scores:

$$u^i_t = \frac{\bar{m}}{\bar{m} + \hat{m}^i_t}, \qquad \bar{m} = \mathbb{E}_{\mathcal{B}}[\hat{m}], \qquad \bar{w}^i_t = \mathrm{sg}\left[\frac{u^i_t}{\mathbb{E}_{\mathcal{B}}[u]}\right], \tag{5}$$

where u^i_t is the raw reliability score and $\bar{w}^i_t$ is its mean-normalized weight. Because u^i_t decreases monotonically with $\hat{m}^i_t$, samples with larger prediction error have less influence on the actor update. Mean normalization keeps the average coefficient equal to one.

The coefficients are recomputed from the current residual-scale prediction at the start of each PPO epoch and held fixed within that epoch, whereas the d^i_t remain fixed at their collection-time values. Importantly, $\bar{w}^i_t$ is not a direct function of the realized magnitude $|A^i_t|$. A large realized advantage therefore does not by itself produce a small coefficient; the coefficient decreases only through the predicted feature-conditioned residual scale $\hat{m}^i_t$.

Scope of signal adjustment. The reliability coefficients modify only the PPO policy surrogate; the value, predictive, and residual objectives remain unchanged. Every rollout sample therefore continues to contribute to representation and value learning, even when its relative influence on the actor update is reduced.

Objective. With $\hat{A}^i_t$ denoting the GAE advantage standardized within each PPO minibatch, DEAL modifies the PPO surrogate as

$$\mathcal{L}^{\mathrm{DEAL}}_\pi = \mathbb{E}_{t,i}\Big[\min\Big(\rho^i_t\,\bar{w}^i_t\hat{A}^i_t,\ \mathrm{clip}(\rho^i_t, 1-\epsilon, 1+\epsilon)\,\bar{w}^i_t\hat{A}^i_t\Big)\Big], \tag{6}$$

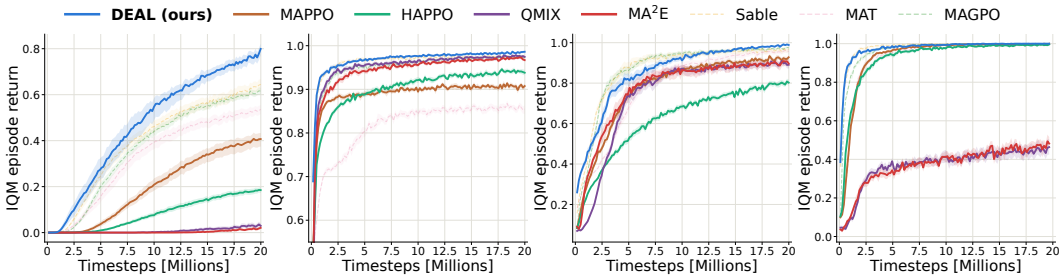
so $\bar{w}^i_t$ acts as a weight controlling each standardized advantage's relative influence on the actor update. Weights are normalized over the rollout, while advantages are standardized within PPO minibatches. The total loss is the standard PPO loss, with $\bar{w}^i_t\hat{A}^i_t$ in its actor term, plus $c_{\mathrm{pred}}\mathcal{L}_{\mathrm{pred}} + c_{\mathrm{res}}\mathcal{L}_{\mathrm{res}}$ (Eq. 19). The stop-gradient rules and the training loop are provided in Appendix B.1 and Algorithm 1.

4 EXPERIMENTS

We set up our experiments to answer three questions: **1.** Does *execution-aligned training* improve performance over strong baselines and against centralized execution? **2.** Does each component address the mismatch it targets and do they interact? **3.** Do the two training signals behave as designed and do they provide benefits under severe observability constraints? The three questions are answered in Sections 4.1, 4.2 and 4.3, respectively.

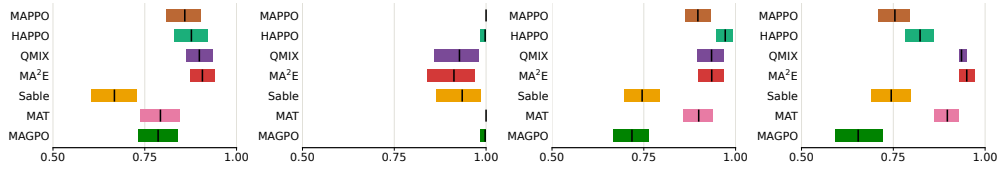
Environments and baselines. We evaluate 37 tasks from RWARE, Connector, SMAX, and Level-Based Foraging (LBF), covering sparse rewards, spatial interference, decentralized combat, and coordinated foraging under partial observability (Papoudakis et al., 2021; Rutherford et al., 2024; Christianos et al., 2020). Appendix C provides environment details and configurations. SMAX includes three SMACv2 maps (Ellis et al., 2023) with randomized unit compositions, introducing more stochastic hidden interaction contexts. We compare DEAL against MAPPO, HAPPO, QMIX, and MA^{2}E as CTDE baselines (Yu et al., 2022; Kuba et al., 2022; Rashid et al., 2020; Kang et al., 2025), and MAT, Sable, and MAGPO as CTCE baselines (Wen et al., 2022; Mahjoub et al., 2025; Li et al., 2026). MAGPO follows a centralized teacher with decentralized student (CTDS) framework and uses centralized execution only for data collection. MA^{2}E represents privileged-information reconstruction, and we additionally test its privileged-information reconstruction objective on our backbone in Table 2.

Evaluation protocol. We follow the evaluation protocol of Gorsane et al. (2022) as used in Sable evaluation (Mahjoub et al., 2025). All reported results use a training budget of 2×10^7 environment timesteps per task and 10 independent seeds. Performance is evaluated at 122 evenly spaced checkpoints, with the mean return computed over 32 episodes. Following the absolute metric of the protocol, we select the best checkpoint and re-evaluate it over 320 episodes. For each environment, we min-max normalize absolute episode returns within each task and aggregate them across tasks and seeds using the interquartile mean (IQM), with 95% stratified bootstrap confidence intervals. All methods use the same task registrations and training budget, while the tuning procedures are detailed in Appendix C.4.



(a) RWARE (b) Connector (c) SMAX (d) LBF

Figure 3: Learning curves. Performance is summarized across each environment's tasks by the min-max normalized interquartile mean (IQM) over 10 seeds; shading is a 95% stratified bootstrap confidence interval. CTDE methods are drawn with solid lines and CTCE methods with dashed lines. All methods share the evaluation protocol.



(a) RWARE (b) Connector (c) SMAX (d) LBF

Figure 4: Aggregated probability of improvement of DEAL over each baseline, per environment (Agarwal et al., 2021). A score above 0.5 whose confidence interval also lies entirely above 0.5 indicates a statistically significant improvement over that baseline in the given environment.

4.1 Main Results

Table 1 and Figure 3 summarize the comparison, and Figure 4 reports per-environment probabilities of improvement over each baseline. DEAL attains the highest normalized IQM in all four environments, with environment-level confidence intervals disjoint from every baseline's, including the centralized-execution baselines. It also achieves the highest absolute return on 30 of the 37 tasks. Over the strongest CTDE baseline of each environment the margin averages 11.1 normalized points, 25% in relative terms. Margins track the room each suite leaves: largest on RWARE, where sparse rewards and severe partial observability keep every method far from ceiling, and smallest on LBF and Connector, where the strongest baselines approach the maximum return.

Table 1: Main results over 10 seeds. For each environment, results are aggregated over all tasks and reported as the min-max normalized interquartile mean (IQM) of the absolute episode returns, multiplied by 100, with 95% stratified bootstrap confidence intervals.

Environment	CTDE methods					CTCE methods		
	DEAL	**MAPPO**	**HAPPO**	**QMIX**	**MA^{2}E**	**Sable**	**MAT**	**MAGPO**†
RWARE	**77.5**$_{(75.0,79.5)}$	40.8$_{(37.5,43.3)}$	18.5$_{(17.2,19.6)}$	5.9$_{(4.3,7.7)}$	3.1$_{(2.1,4.3)}$	63.9$_{(61.5,66.1)}$	53.5$_{(51.5,55.3)}$	60.5$_{(59.5,61.7)}$
Connector	**97.9**$_{(97.7,98.0)}$	95.3$_{(95.1,95.5)}$	94.0$_{(93.8,94.2)}$	97.2$_{(97.1,97.3)}$	97.0$_{(96.8,97.2)}$	97.0$_{(96.8,97.2)}$	93.3$_{(92.9,93.8)}$	96.6$_{(96.3,96.8)}$
SMAX	**98.6**$_{(98.5,98.7)}$	90.9$_{(89.9,91.8)}$	78.9$_{(77.9,79.9)}$	91.4$_{(90.1,92.6)}$	91.8$_{(90.0,93.3)}$	96.0$_{(94.7,97.0)}$	95.2$_{(94.1,96.1)}$	96.4$_{(96.0,96.8)}$
LBF	**99.9**$_{(99.9,100.0)}$	99.7$_{(99.6,99.7)}$	99.1$_{(98.9,99.2)}$	49.2$_{(46.0,52.2)}$	51.7$_{(49.5,54.1)}$	99.7$_{(99.6,99.8)}$	98.6$_{(98.4,98.8)}$	99.7$_{(99.7,99.8)}$

† Denotes a CTDS framework that uses centralized executors to gather samples during training but only decentralized actors at evaluation.

The value-based methods are the exception: QMIX and MA^{2}E collapse on RWARE and LBF, consistent with the documented failure mode of value decomposition under sparse reward (Papoudakis et al., 2021). On SMAX, MA^{2}E holds an early advantage over QMIX up to roughly 2.5M steps, consistent with its privileged pretraining, but the difference vanishes with further training. Learning curves show stable gains throughout training; per-task scores and curves are in Appendix C.3.

4.2 Which Component Contributes What?

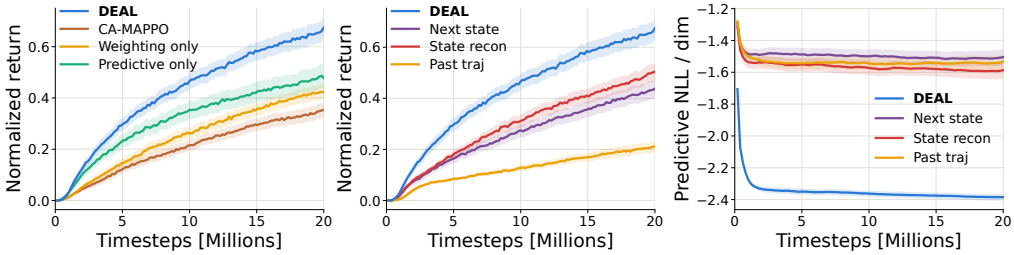
We use a recurrent MAPPO backbone whose centralized critic first mixes agent representations through cross-agent self-attention and then aggregates them temporally with a recurrent encoder. We refer to this backbone as CA-MAPPO (Cross-Agent-Attention MAPPO). CA-MAPPO itself performs similarly to MAPPO (34.2 versus 34.8), indicating that learning context-dependent per-agent critic representations alone does not account for the gains of DEAL.

Table 2: *Component evaluation*: min-max normalized mean absolute episode returns, multiplied by 100, over the 15 RWARE tasks and 10 seeds with 95% stratified bootstrap CIs.

Variant	Predictive target	Weight.	Return	Δ vs. base	Train cost
CA-MAPPO (base)	–	–	34.2$_{(31.4,37.0)}$	–	1.00×
Learning signal weighting only	–	✓	41.6$_{(38.7,44.3)}$	+22%	1.07×
Predictive representation only	next obs. o^i_{t+1}	–	47.4$_{(43.0,51.9)}$	+39%	1.03×
DEAL with next-state prediction	next state s_{t+1}	✓	42.5$_{(39.0,45.9)}$	+24%	1.15×
DEAL with state reconstruction	current state s_t	✓	49.0$_{(45.5,52.4)}$	+43%	1.16×
DEAL with past-trajectory recon.	joint obs. $o^{1:N}_{t-2:t}$	✓	20.7$_{(18.6,22.8)}$	-39%	1.32×
DEAL with a single Gaussian	next obs. o^i_{t+1}	✓	46.8$_{(42.4,51.2)}$	+37%	1.09×
DEAL (predictive + weighting)	next obs. o^i_{t+1}	✓	**65.4**$_{(61.3,69.4)}$	**+91%**	1.10×

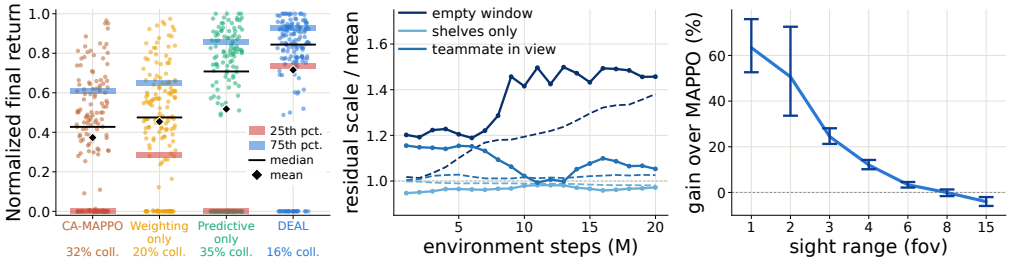
Table 2 evaluates the components over RWARE: it switches the two training modules on and off on top of the CA-MAPPO base and, holding the rest of DEAL fixed, swaps the predictive target for three alternatives. Figure 5 shows the corresponding learning curves and prediction loss.

Predictive representation. The predictive representation yields the largest performance improvement. It produces a clear gain over the base method early in training, improving both final performance



(a) Component arms (b) Return by prediction target (c) Prediction loss by target

Figure 5: *Component evaluation*, pooled over the 15 RWARE tasks of Table 2 (95% CIs). **(a)** Learning curves for the component arms. **(b)** Learning curves by prediction target with the rest of DEAL fixed. **(c)** Per-dimension prediction NLL of the same head.



(a) Tail stabilization (b) Residual scale by local context (c) Observability scope

Figure 6: *Weighting and observability diagnostics.* **(a)** Normalized performances over the 15 RWARE tasks (150 runs per arm) with the fraction of collapsed runs. **(b)** Realized residual $|G^\lambda - V|$ (solid) and predicted $\hat{m}$ (dashed), relative to the rollout mean, by the content of the agent's own observation window on RWARE. **(c)** DEAL's relative gain over MAPPO on LBF as sight shrinks (95% CIs).

and sample efficiency. Its effectiveness also depends strongly on the choice of prediction target. Replacing the target with the next global state, the current global state, or a past joint trajectory consistently degrades performance. Figure 5c provides complementary evidence: the alternative targets exhibit consistently higher predictive losses, consistent with being harder to predict from each agent's local history. In particular, reconstructing past team trajectories yields substantially lower returns, suggesting that this temporally extended reconstruction objective is poorly matched to the local predictive representation under strong partial observability.

Learning signal weighting. Learning signal weighting improves performance on its own, and its combination with the predictive representation outperforms either component alone. Its main effect is on the lower tail of the performance distribution. Figure 6a shows the final normalized return of each run for the four variants, with the 25th and 75th percentiles and the proportion of collapsed runs (normalized return below 0.05). The predictive representation substantially improves performance but retains a pronounced lower tail of failed runs. Adding the weighting improves the lower tail and substantially reduces the proportion of collapsed runs in both the base and predictive settings.

4.3 Effect of Observability

Residual-scale heterogeneity under partial observability. Section 3.2 argues that the collection-time residual scale can differ across agents and situations. We examine this on RWARE-tiny-4ag using the agent's observation window (Figure 6b). As training progresses, the residual for samples with an empty window increases to about $1.5\times$ the rollout mean, while the other two remain near the mean. In the empty-window case, the agent carries no shelf and cannot deliver at the current step, yet it can still receive shared rewards generated by teammates. The shared return is less directly tied to the agent's current action, leaving greater ambiguity in agent-wise value estimation. This pattern reveals persistent, structured heterogeneity in residual scale under partial observability. The residual-scale head tracks this structure from the critic features ($\hat{m}$, dashed), and the predicted scale adjusts each sample's influence on the policy update.

Performance gains under reduced observability. The LBF sight range sweep in Figure 6c varies the level of partial observability, with DEAL trained using MAPPO's tuned hyperparameters in this sweep. The relative gain increases steadily as sight decreases, from a small deficit under full observability to over 60% at the narrowest window. This trend is consistent with the intended role of DEAL: as sight decreases, locally predicting future observations becomes more challenging and the gains of DEAL increase. It can also make value estimates less reliable for some agents and situations, giving the reliability weighting more to exploit. As more of the environment becomes observable, these effects weaken, although predictive learning can still provide useful representation and dynamics information under full observability. Further analyses of the interaction between the two components, mixture size, hyperparameter sensitivity, and limitations are provided in Appendices D and E.

5 RELATED WORK

Narrowing decentralized-execution gap of CTDE. To overcome the decentralized-execution limitation of CTDE, CTCE approaches condition the actors on joint information or agent ordering and outperform decentralized policies when coordination hinges on locally unavailable information (Wen et al., 2022; Daniel et al., 2025; Mahjoub et al., 2025). Distillation, privileged teachers, and centralized guidance aim to transfer such centralized capability to decentralized actors (Zhao et al., 2024; Chen et al., 2024; Li et al., 2026). However, the transferred information or behavior may depend on global context that a decentralized actor cannot reliably infer from its local history or reproduce at execution (Warrington et al., 2021; Cai et al., 2024). Inter-agent communication (Das et al., 2019; Guan et al., 2022) relaxes the constraint itself, a stronger execution-time assumption than CTDE. The centralized critic is itself hard to fit here: state-conditioned critics are biased for history-dependent policies, and centralized inputs need not lower estimation variance (Baisero & Amato, 2022; Lyu et al., 2023). Rather than a better-specified critic, DEAL predicts each sample's residual scale and weights its policy-gradient influence accordingly.

Predictive representations under partial observability. A common remedy for partial observability is to learn richer representations from local histories, e.g., through recurrent belief learning and predictive-state representations (Littman et al., 2001; Igl et al., 2018; Gregor et al., 2019), auxiliary objectives that reconstruct global features or latent states (Song et al., 2023; Li et al., 2024; Xu et al., 2024; Kharrat et al., 2025; Kang et al., 2025), or explicit belief models (Kontogiannis et al., 2025; Pritz & Leung, 2025). Global or latent-state reconstruction can supervise the actor with information that may not be recoverable from its local history. In contrast, DEAL predicts each agent's own next observation, which is directly observed along its decentralized trajectory, and provides a predictive-scale readout at execution. Next-observation prediction has also been widely used in single-agent RL, including world models (Ha & Schmidhuber, 2018), auxiliary prediction (Jaderberg et al., 2017), and latent-dynamics or self-predictive representations (Hafner et al., 2019; Schwarzer et al., 2021). DEAL adopts the prediction objective without imagination or planning, and uses the next observation because of its locality: each agent predicts what it will observe rather than what the center observes.

Uncertainty-aware optimization. Uncertainty has been used to attenuate unreliable targets (Wu et al., 2021; Queeney et al., 2021; Mai et al., 2022), with heteroscedastic likelihoods a scalable but delicate way to estimate input-dependent residual scale (Nix & Weigend, 1994; Kendall & Gal, 2017; Seitzer et al., 2022); in MARL, baselines or teammate-action marginalization reduce policy-gradient variance (Kuba et al., 2021; Jafferjee et al., 2025). DEAL instead predicts each sample's collection-time return residual from critic features, separate from the observation-side predictive scale used at action selection.

6 CONCLUSION

We introduced DEAL, which aligns CTDE training with decentralized execution through a locally grounded predictive representation and adaptive learning signal weighting. Across 37 tasks in four suites, DEAL achieves the best performance in every suite, outperforming strong CTDE and CTCE baselines. Ablations and targeted diagnostics show that predictive representation drives the main gains while critic signal weighting improves stability, supporting the benefits of execution-aligned training under partial observability.

AI USE STATEMENT

In this work, we used generative AI tools for polishing the text and limited sentence revisions, and for reviewing the authors' implementation and theoretical analysis for potential issues. We also used AI tools to organize numerical results and prepare figures from the experimental results. The proposed method, experimental design, implementation, and experiments were conducted by the authors, and all AI outputs were reviewed and verified by the authors.

ETHICS STATEMENT

This work studies cooperative multi-agent reinforcement learning in simulated benchmark environments. No human subjects or personal data are used, and no new datasets are introduced beyond standard public benchmarks. We are not aware of harmful applications specific to the proposed method beyond the broader risks associated with reinforcement learning.

REPRODUCIBILITY STATEMENT

The full training loop and implementation details are stated in Algorithm 1 and Appendix B.1. The experimental protocol, environments, and hyperparameter settings appear in Appendix C. We report all task results in Appendix C.3. The full implementation is provided in the supplementary material.

REFERENCES

Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron Courville, and Marc G. Bellemare. Deep reinforcement learning at the edge of the statistical precipice. In *Advances in Neural Information Processing Systems*, 2021.

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2623–2631, 2019.

Andrea Baisero and Christopher Amato. Unbiased asymmetric reinforcement learning under partial observability. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pp. 44–52, 2022.

Yang Cai, Xiangyu Liu, Argyris Oikonomou, and Kaiqing Zhang. Provable partially observable reinforcement learning with privileged information. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Yiqun Chen, Hangyu Mao, Jiaxin Mao, Shiguang Wu, Tianle Zhang, Bin Zhang, Wei Yang, and Hongxing Chang. PTDE: Personalized training with distilled execution for multi-agent reinforcement learning. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, pp. 31–39, 2024.

Filippos Christianos, Lukas Schäfer, and Stefano V. Albrecht. Shared experience actor-critic for multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems*, 2020.

Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley-Interscience, 2nd edition, 2006.

Jemma Daniel, Ruan John de Kock, Louay Ben Nessir, Sasha Abramowitz, Omayma Mahjoub, Wiem Khlifi, Juan Claude Formanek, and Arnu Pretorius. Multi-agent reinforcement learning with selective state-space models. In *Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pp. 2481–2483, 2025. Extended Abstract.

Abhishek Das, Théophile Gervet, Joshua Romoff, Dhruv Batra, Devi Parikh, Mike Rabbat, and Joelle Pineau. TarMAC: Targeted multi-agent communication. In *International Conference on Machine Learning*, 2019.

Ruan de Kock, Omayma Mahjoub, Sasha Abramowitz, Wiem Khlifi, Callum Rhys Tilbury, Claude Formanek, Andries P. Smit, and Arnu Pretorius. Mava: A research library for distributed multi-agent reinforcement learning in jax. *arXiv preprint arXiv:2107.01460*, 2023.

Benjamin Ellis, Jonathan Cook, Skander Moalla, Mikayel Samvelyan, Mingfei Sun, Anuj Mahajan, Jakob Foerster, and Shimon Whiteson. SMACv2: An improved benchmark for cooperative multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems*, 2023.

Rihab Gorsane, Omayma Mahjoub, Ruan John de Kock, Roland Dubb, Siddarth Singh, and Arnu Pretorius. Towards a standardised performance evaluation protocol for cooperative MARL. In *Advances in Neural Information Processing Systems*, 2022.

Karol Gregor, Danilo Jimenez Rezende, Frederic Besse, Yan Wu, Hamza Merzic, and Aäron van den Oord. Shaping belief states with generative environment models for RL. In *Advances in Neural Information Processing Systems*, 2019.

Cong Guan, Feng Chen, Lei Yuan, Chenghe Wang, Hao Yin, Zongzhang Zhang, and Yang Yu. Efficient multi-agent communication via self-supervised information aggregation. In *Advances in Neural Information Processing Systems*, 2022.

David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. In *Advances in Neural Information Processing Systems*, 2018.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pp. 2555–2565. PMLR, 2019.

Maximilian Igl, Luisa Zintgraf, Tuan Anh Le, Frank Wood, and Shimon Whiteson. Deep variational reinforcement learning for POMDPs. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pp. 2117–2126. PMLR, 2018.

Shariq Iqbal and Fei Sha. Actor-attention-critic for multi-agent reinforcement learning. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2961–2970, 2019.

Max Jaderberg, Volodymyr Mnih, Wojciech Marian Czarnecki, Tom Schaul, Joel Z. Leibo, David Silver, and Koray Kavukcuoglu. Reinforcement learning with unsupervised auxiliary tasks. In *International Conference on Learning Representations*, 2017.

Taher Jafferjee, Juliusz Ziomek, Tianpei Yang, Zipeng Dai, Jianhong Wang, Matthew E. Taylor, Kun Shao, Jun Wang, and David Mguni. Taming multi-agent reinforcement learning with estimator variance reduction. In *Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pp. 1042–1050, 2025.

Sehyeok Kang, Yongsik Lee, Gahee Kim, Song Chong, and Se-Young Yun. MA^{2}E: Addressing partial observability in multi-agent reinforcement learning with masked auto-encoder. In *International Conference on Learning Representations*, 2025.

Alex Kendall and Yarin Gal. What uncertainties do we need in Bayesian deep learning for computer vision? In *Advances in Neural Information Processing Systems*, 2017.

Salma Kharrat, Fares Fourati, Marco Canini, Mohamed-Slim Alouini, and Vaneet Aggarwal. Latent inference for effective multi-agent reinforcement learning under partial observability. In *Eighteenth European Workshop on Reinforcement Learning*, 2025.

Andreas Kontogiannis, Konstantinos Papathanasiou, Yi Shen, Giorgos Stamou, Michael M. Zavlanos, and George Vouros. Enhancing cooperative multi-agent reinforcement learning with state modelling and adversarial exploration. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pp. 31437–31466. PMLR, 2025.

Jakub Grudzien Kuba, Muning Wen, Linghui Meng, Shangding Gu, Haifeng Zhang, David Mguni, Jun Wang, and Yaodong Yang. Settling the variance of multi-agent policy gradients. In *Advances in Neural Information Processing Systems*, 2021.

Jakub Grudzien Kuba, Ruiqing Chen, Muning Wen, Ying Wen, Fanglei Sun, Jun Wang, and Yaodong Yang. Trust region policy optimisation in multi-agent reinforcement learning. In *International Conference on Learning Representations*, 2022.

Xiaoyang Li, Guohua Yang, Dawei Zhang, and Jianhua Tao. APC: Predict global representation from local observation in multi-agent reinforcement learning. In *International Joint Conference on Neural Networks*, pp. 1–8, 2024.

Yueheng Li, Guangming Xie, and Zongqing Lu. Multi-agent guided policy optimization. In *International Conference on Learning Representations*, 2026.

Michael L. Littman, Richard S. Sutton, and Satinder Singh. Predictive representations of state. In *Advances in Neural Information Processing Systems*, 2001.

Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. In *Advances in Neural Information Processing Systems*, 2017.

Xueguang Lyu, Andrea Baisero, Yuchen Xiao, Brett Daley, and Christopher Amato. On centralized critics in multi-agent reinforcement learning. *Journal of Artificial Intelligence Research*, 77: 295–354, 2023.

Omayma Mahjoub, Sasha Abramowitz, Ruan John De Kock, Wiem Khlifi, Simon Verster Du Toit, Jemma Daniel, Louay Ben Nessir, Louise Beyers, Juan Claude Formanek, Liam Clark, and Arnu Pretorius. Sable: A performant, efficient and scalable sequence model for MARL. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 42579–42614, 2025.

Vincent Mai, Kaustubh Mani, and Liam Paull. Sample efficient deep reinforcement learning via uncertainty estimation. In *International Conference on Learning Representations*, 2022.

David A. Nix and Andreas S. Weigend. Estimating the mean and variance of the target probability distribution. In *Proceedings of 1994 IEEE International Conference on Neural Networks (ICNN'94)*, volume 1, pp. 55–60, 1994.

Georgios Papoudakis, Filippos Christianos, Lukas Schäfer, and Stefano V. Albrecht. Benchmarking multi-agent deep reinforcement learning algorithms in cooperative tasks. In *Advances in Neural Information Processing Systems*, 2021.

Bo Peng, Eric Alcaide, Quentin Anthony, Alon Albalak, Samuel Arcadinho, Stella Biderman, Huanqi Cao, Xin Cheng, Michael Chung, Leon Derczynski, Xingjian Du, Matteo Grella, Kranthi Gv, Xuzheng He, Haowen Hou, Przemyslaw Kazienko, Jan Kocon, Jiaming Kong, Bartłomiej Koptyra, Hayden Lau, Jiaju Lin, Krishna Sri Ipsit Mantri, Ferdinand Mom, Atsushi Saito, Guangyu Song, Xiangru Tang, Johan Wind, Stanisław Woźniak, Zhenyuan Zhang, Qinghua Zhou, Jian Zhu, and Rui-Jie Zhu. RWKV: Reinventing RNNs for the transformer era. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 14048–14077, 2023.

Paul J. Pritz and Kin K. Leung. Belief states for cooperative multi-agent reinforcement learning under partial observability. *arXiv preprint arXiv:2504.08417*, 2025.

Liang Qifan, Yixiang Shan, Haipeng Liu, Zhengbang Zhu, Ting Long, Weinan Zhang, and Yuan Tian. Reconstruction-guided policy: Enhancing decision-making through agent-wise state consistency. In *The Thirteenth International Conference on Learning Representations*, 2025.

James Queeney, Ioannis Ch. Paschalidis, and Christos G. Cassandras. Uncertainty-aware policy optimization: A robust, adaptive trust region approach. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 9377–9385, 2021.

Tabish Rashid, Mikayel Samvelyan, Christian Schroeder de Witt, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. Monotonic value function factorisation for deep multi-agent reinforcement learning. *Journal of Machine Learning Research*, 21(178):1–51, 2020.

Alexander Rutherford, Benjamin Ellis, Matteo Gallici, Jonathan Cook, Andrei Lupu, Garðar Ingvarsson, Timon Willi, Ravi Hammond, Akbir Khan, Christian Schroeder de Witt, Alexandra Souly, Saptarashmi Bandyopadhyay, Mikayel Samvelyan, Minqi Jiang, Robert Lange, Shimon Whiteson, Bruno Lacerda, Nick Hawes, Tim Rocktäschel, Chris Lu, and Jakob Foerster. Jaxmarl: Multi-agent rl environments and algorithms in jax. In *Advances in Neural Information Processing Systems*, 2024.

Mikayel Samvelyan, Tabish Rashid, Christian Schroeder de Witt, Gregory Farquhar, Nantas Nardelli, Tim G. J. Rudner, Chia-Man Hung, Philip H. S. Torr, Jakob Foerster, and Shimon Whiteson. The StarCraft multi-agent challenge. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, pp. 2186–2188, 2019. Extended Abstract.

John Schulman, Philipp Moritz, Sergey Levine, Michael I. Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. In *International Conference on Learning Representations*, 2016.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Max Schwarzer, Ankesh Anand, Rishab Goel, R Devon Hjelm, Aaron Courville, and Philip Bachman. Data-efficient reinforcement learning with self-predictive representations. In *International Conference on Learning Representations*, 2021.

Maximilian Seitzer, Arash Tavakoli, Dimitrije Antic, and Georg Martius. On the pitfalls of heteroscedastic uncertainty estimation with probabilistic neural networks. In *International Conference on Learning Representations*, 2022.

Haolin Song, Mingxiao Feng, Wengang Zhou, and Houqiang Li. MA2CL: Masked attentive contrastive learning for multi-agent reinforcement learning. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, pp. 4226–4234, 2023.

Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, Vinicius Zambaldi, Max Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z Leibo, Karl Tuyls, et al. Value-decomposition networks for cooperative multi-agent learning. *arXiv preprint arXiv:1706.05296*, 2017.

Andrew Warrington, Jonathan W Lavington, Adam Scibior, Mark Schmidt, and Frank Wood. Robust asymmetric learning in POMDPs. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pp. 11013–11023. PMLR, 2021.

Muning Wen, Jakub Grudzien Kuba, Runji Lin, Weinan Zhang, Ying Wen, Jun Wang, and Yaodong Yang. Multi-agent reinforcement learning is a sequence modeling problem. In *Advances in Neural Information Processing Systems*, 2022.

Yue Wu, Shuangfei Zhai, Nitish Srivastava, Joshua M. Susskind, Jian Zhang, Ruslan Salakhutdinov, and Hanlin Goh. Uncertainty weighted actor-critic for offline reinforcement learning. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pp. 11319–11328. PMLR, 2021.

Zhiwei Xu, Hangyu Mao, Nianmin Zhang, Xin Xin, Pengjie Ren, Dapeng Li, Bin Zhang, Guoliang Fan, Zhumin Chen, Changwei Wang, and Jiangjin Yin. Beyond local views: Global state inference with diffusion models for cooperative multi-agent reinforcement learning. *arXiv preprint arXiv:2408.09501*, 2024.

Chao Yu, Akash Velu, Eugene Vinitsky, Jiaxuan Gao, Yu Wang, Alexandre Bayen, and Yi Wu. The surprising effectiveness of PPO in cooperative multi-agent games. In *Advances in Neural Information Processing Systems*, 2022.

Jian Zhao, Xunhan Hu, Mingyu Yang, Wengang Zhou, Jiangcheng Zhu, and Houqiang Li. CTDS: Centralized teacher with decentralized student for multiagent reinforcement learning. *IEEE Transactions on Games*, 16(1):140–150, 2024.

Organization. Appendix A develops the theory underlying the two components and clarifies its scope; Appendix B describes the training loop, implementation details, and execution cost; Appendix C details the environments, experimental protocol, full per-task results, and hyperparameters; Appendix D presents the extended ablations; and Appendix E discusses limitations and future work.

A Theory and Scope

This appendix explains the predictive objective, the weighting rule, and the two scale readouts used by DEAL. We use the fixed target $Z' = \mathrm{symlog}(O^i_{t+1})$ and suppress agent and time indices when unambiguous. The density-based results assume conditional densities and finite entropies and cross-entropies. For discrete observation coordinates, the Gaussian NLL serves as a continuous surrogate, as described in Appendix B.1.

A.1 Predictive Risk Decomposition

Fact 1 (Predictive risk decomposition)**.** Let $b = f_\psi(h)$ and let $p^\star(\cdot \mid h, a)$ be the conditional distribution of Z'. Then

$$\mathcal{R}_{\mathrm{NLL}} := \mathbb{E}[-\log q_\theta(Z' \mid b, a)] = \mathbb{E}[H(p^\star(\cdot \mid h, a))] + \underbrace{\mathbb{E}[D_{\mathrm{KL}}(p^\star(\cdot \mid h, a) \,\|\, q_\theta(\cdot \mid b, a))]}_{\Delta_{\mathrm{pred}} \geq 0}. \quad (7)$$

Equality with the entropy term holds exactly when the two conditional distributions agree almost surely.

Proof. Write $-\log q_\theta = -\log p^\star + \log(p^\star/q_\theta)$ and take expectations, first over Z' given (h, a) and then over (h, a). □

The first term is the irreducible NLL for the chosen target representation. The second measures the predictive error due to both compression into b and the predictive family. Reaching zero requires b to retain the history information needed to determine $p^\star(\cdot \mid h, a)$ and the model to represent that distribution. The identity applies on valid transitions; per-dimension normalization divides both terms by the target dimension.

A.2 Single-Gaussian Misspecification Gap

Proposition 2 (Single-Gaussian misspecification gap)**.** Fix (h, a) and write $p^\star = p^\star(\cdot \mid h, a)$. Suppose Z' has finite mean $\mu^\star$ and finite, positive-definite covariance $\Sigma^\star$. The full-covariance Gaussian minimizing expected NLL is $q^\star = \mathcal{N}(\mu^\star, \Sigma^\star)$, with gap

$$\mathcal{G}_{\mathrm{uni}} = H(q^\star) - H(p^\star) = D_{\mathrm{KL}}(p^\star \| q^\star) \geq 0. \quad (8)$$

The gap is zero exactly when $p^\star$ is Gaussian.

Proof. For $q = \mathcal{N}(\mu, \Sigma)$ in d dimensions,

$$\mathbb{E}_{p^\star}[-\log q(Z')] = \tfrac{1}{2}\log((2\pi)^d|\Sigma|) + \tfrac{1}{2}\,\mathrm{tr}(\Sigma^{-1}\Sigma^\star) + \tfrac{1}{2}(\mu^\star - \mu)^\top \Sigma^{-1}(\mu^\star - \mu). \quad (9)$$

Minimizing first over μ gives $\mu = \mu^\star$; minimizing over the precision matrix Σ^{-1} then gives $\Sigma = \Sigma^\star$. Substitution yields $H(q^\star)$, and the cross-entropy identity gives the gap (Cover & Thomas, 2006). □

Remark 1 (Relation to the deployed mixture)**.** The deployed $K = 1$ head has diagonal covariance $\mathrm{diag}\, v(b)$, so its best achievable gap is at least $\mathcal{G}_{\mathrm{uni}}$. Additional components can model non-Gaussian structure and dependence between coordinates through their means. The $K = 1$ ablation therefore tests the value of this additional mixture capacity.

A.3 The Reliability-Weighting Objective

The weighting rule reduces the influence of samples with larger predicted residual scales while keeping the average coefficient at one. To express this design, index the B valid agent-time samples by j, let $M_j \geq 0$ be their fixed scores, and consider

$$\min_{w_1,\dots,w_B} \sum_{j=1}^{B} M_j w_j^2 + c_B \sum_{j=1}^{B} (w_j - 1)^2, \qquad \frac{1}{B}\sum_{j=1}^{B} w_j = 1, \tag{10}$$

where $c_B > 0$. The first term penalizes large weights on high-score samples; the second keeps weights near uniform.

Lemma 1 (Minimizer of the weighting objective)**.** The unique solution of Eq. (10) is

$$w_j^\star = \frac{(c_B + M_j)^{-1}}{\frac{1}{B}\sum_{k=1}^{B}(c_B + M_k)^{-1}}. \tag{11}$$

Proof. The objective is strictly convex. The Lagrange multiplier condition makes $(M_j + c_B)w_j$ constant across samples. Enforcing the mean-one constraint gives Eq. (11). □

Setting $M_j = \hat{m}_j$ and $c_B = \bar{m} = \mathbb{E}_{\mathcal{B}}[\hat{m}]$ recovers Eq. (5): $u_j = \bar{m}/(\bar{m} + \hat{m}_j)$ and $\bar{w}_j = u_j/\mathbb{E}_{\mathcal{B}}[u]$. Without the numerical floor, multiplying all scores by the same positive constant leaves the weights unchanged. With the floor in Algorithm 1, take $c_B = \mathbb{E}_{\mathcal{B}}[\hat{m}] + \epsilon_0$ instead; the same solution applies.

Lemma 2 (Bounded weights and effective sample size)**.** For positive scores $\hat{m}_j$ and $c_B \geq \mathbb{E}_{\mathcal{B}}[\hat{m}]$, the normalized weights satisfy

$$0 < \bar{w}_j < 2, \qquad \mathbb{E}_{\mathcal{B}}[\bar{w}] = 1, \qquad \mathrm{ESS} := \frac{(\sum_j \bar{w}_j)^2}{\sum_j \bar{w}_j^2} > \frac{B}{2}. \tag{12}$$

Proof. Let $u_j = c_B/(c_B + \hat{m}_j) \in (0, 1)$. Convexity and Jensen's inequality give

$$\mathbb{E}_{\mathcal{B}}[u] \geq \frac{c_B}{c_B + \mathbb{E}_{\mathcal{B}}[\hat{m}]} \geq \frac{1}{2}.$$

Thus $\bar{w}_j = u_j/\mathbb{E}_{\mathcal{B}}[u] < 2$. Since $\bar{w}_j^2 < 2\bar{w}_j$ and the weights have mean one, $\mathrm{ESS} = B/\mathbb{E}_{\mathcal{B}}[\bar{w}^2] > B/2$. □

These results characterize the allocation of coefficients, including the numerical floor. Mean normalization controls their average, not the direction or magnitude of the resulting policy gradient.

A.4 Interpretation of Predictive and Residual Scales

Hidden context and predictive scale. Let c collect the hidden variables affecting an agent's next observation. Marginalizing them gives

$$p^\star(z' \mid h, a) = \int p(c \mid h, a)\, p(z' \mid h, a, c)\, dc. \tag{13}$$

The integral includes summation for discrete variables. Different contexts compatible with the same (h, a) can yield a multimodal predictive distribution. The learned mixture represents this distribution without requiring its components to identify individual contexts. Its covariance is

$$\mathrm{Cov}_{q_\theta}(Z' \mid b, a) = \mathrm{diag}\, v(b) + \sum_k \alpha_k (\mu_k - \bar{\mu})(\mu_k - \bar{\mu})^\top, \qquad \bar{\mu} = \sum_k \alpha_k \mu_k. \tag{14}$$

Here α_k and μ_k depend on (b, a). The policy readout $\zeta = \mathrm{sg}[\ell(b)]$ describes the shared within-component scale, while the second term captures spread between component means.

The collection-time residual target. Fix the collection policy and critic, and write V_b^i for the collection-time snapshot of V_ϕ^i. The return target $G_t^{\lambda,i}$, denoted $R_t^{\lambda,i}$ in Algorithm 1, satisfies

$$G_t^{\lambda,i} = V_{b,t}^i + A_t^i, \qquad d_t^i = |G_t^{\lambda,i} - V_{b,t}^i| = |A_t^i|, \tag{15}$$

where A_t^i is the unstandardized GAE estimate. These targets remain fixed throughout the PPO update. For the following analysis, let $Y = G_t^{\lambda,i}$, let $\tilde{o}$ be the collection-time critic feature, and abbreviate its value by $V_b(\tilde{o})$. All conditional quantities refer to this rollout distribution and its fixed λ-return construction.

For the collection-time feature, the population regression target is

$$m^\star(\tilde{o}) = \mathbb{E}[d \mid \tilde{o}] = \mathbb{E}[|Y - V_b(\tilde{o})| \mid \tilde{o}].$$

For a fixed feature representation, the conditional squared-error risk equals $(\hat{m} - m^\star)^2 + \mathrm{Var}(d \mid \tilde{o})$ up to an additive constant, so its minimizer is $m^\star$. During PPO, the stored residual targets remain fixed while the critic features are refreshed, and the head fits these same targets under the current representation. Weighting uses the resulting prediction rather than the realized $|A_t^i|$ directly.

What the residual scale captures. Let $\mu(\tilde{o}) = \mathbb{E}[Y \mid \tilde{o}]$. Target variation can differ across features: for context c containing the latent state and joint histories, with $\tilde{o}$ determined by c,

$$\mathrm{Var}(Y \mid \tilde{o}) = \mathbb{E}[\mathrm{Var}(Y \mid c) \mid \tilde{o}] + \mathrm{Var}(\mathbb{E}[Y \mid c] \mid \tilde{o}). \tag{16}$$

The first term reflects variation in future actions and transitions; the second reflects differences in expected targets among contexts sharing a feature. The latter vanishes when $\mathbb{E}[Y \mid c]$ is determined by $\tilde{o}$.

Proposition 3 (Residual-scale bounds)**.** Define the conditional target spread $s_1(\tilde{o}) = \mathbb{E}[|Y - \mu(\tilde{o})| \mid \tilde{o}]$ and the conditional-mean mismatch $e_V(\tilde{o}) = |\mu(\tilde{o}) - V_b(\tilde{o})|$. Then

$$e_V \le m^\star, \qquad |m^\star - s_1| \le e_V. \tag{17}$$

Proof. Jensen's inequality gives $\mathbb{E}[|Y - V_b| \mid \tilde{o}] \ge |\mathbb{E}[Y \mid \tilde{o}] - V_b|$. The reverse triangle inequality gives $\big||Y - V_b| - |Y - \mu|\big| \le |\mu - V_b|$; taking conditional expectations proves the second bound. □

Thus, the residual scale reflects both conditional spread in the training target and mismatch of the collection-time value from its conditional mean. In particular, $m^\star = s_1$ when $V_b = \mu$, and $s_1 \le m^\star + e_V \le 2m^\star$ in general. These statements concern the collected λ-return target, including its bootstrap values. We use *reliability* in this feature-conditioned sense: a critic-derived signal is less reliable where the return target has greater conditional spread given the critic feature or where the collection-time value has a larger mismatch from its conditional mean.

Action-dependent target differences. To distinguish action effects from the remaining residual, define

$$\mu_a(\tilde{o}, a) = \mathbb{E}[Y \mid \tilde{o}, a], \qquad \Delta(\tilde{o}, a) = \mu_a(\tilde{o}, a) - \mu(\tilde{o}), \qquad \xi = Y - \mu_a(\tilde{o}, a).$$

Here a is the agent's action, and Δ is its associated difference in the expected λ-return target. Since $Y - V_b = \xi + \Delta + (\mu - V_b)$, set

$$m_0(\tilde{o}) = \mathbb{E}[|\xi + \mu(\tilde{o}) - V_b(\tilde{o})| \mid \tilde{o}], \qquad a_1(\tilde{o}) = \mathbb{E}[|\Delta(\tilde{o}, a)| \mid \tilde{o}].$$

The reverse triangle inequality yields

$$|m^\star(\tilde{o}) - m_0(\tilde{o})| \le a_1(\tilde{o}). \tag{18}$$

Consequently, for two feature values,

$$m_0(\tilde{o}_1) - m_0(\tilde{o}_2) > a_1(\tilde{o}_1) + a_1(\tilde{o}_2) \quad \Longrightarrow \quad m^\star(\tilde{o}_1) > m^\star(\tilde{o}_2).$$

This gives a sufficient condition under which residual-scale ordering preserves the ordering of the non-action residual scale m_0: the separation in m_0 must exceed the corresponding action-dependent contributions. Under this condition, larger predicted residual scales indicate lower feature-conditioned reliability of the critic-derived learning signal.

B ALGORITHM AND IMPLEMENTATION

Algorithm 1 summarizes the full DEAL training loop. Relative to recurrent MAPPO, DEAL introduces two components: a predictive objective that shapes the actor information feature and produces the pre-action readout ζ, and a reliability weight $\bar{w}$ that modulates each sample's contribution to the policy update. The residual target d is computed once using collection-time values and remains fixed throughout the PPO update, whereas $\hat{m}$ and $\bar{w}$ are recomputed at the start of each PPO epoch. Removing the predictive pathway and setting $\bar{w} \equiv 1$ recovers the CA-MAPPO base in Table 2.

Algorithm 1 DEAL

Require: actor $(f_\psi, \pi_\omega, q_\theta, \ell)$; critic (V_ϕ, g_η); coefficients $c_{\text{pred}}, c_{\text{res}}$

while training budget not exhausted **do**
  Decentralized rollout
  for $t = 0, \dots, T-1$ **do**
    for each agent i **in parallel do**
      $b_t^i \leftarrow f_\psi(h_t^i)$
      $\zeta_t^i \leftarrow \text{sg}[\ell(b_t^i)]$
      $a_t^i \sim \pi_\omega(\cdot \mid b_t^i, \zeta_t^i)$
    end for
    execute joint action and observe $(r_t, o_{t+1}^{1:N})$
  end for
  Collection-time targets
  $\tilde{o}_{b,t}^i \leftarrow \text{CriticFeat}_\phi(o_t^{1:N})$; $V_{b,t}^i \leftarrow V_\phi(\tilde{o}_{b,t}^i)$
  $A_t^i \leftarrow \text{GAE}(r, V_b)$; $R_t^{\lambda,i} \leftarrow A_t^i + V_{b,t}^i$
  $d_t^i \leftarrow \text{sg}\left[|R_t^{\lambda,i} - V_{b,t}^i|\right]$ ▷ fixed during the PPO update
  for each PPO epoch **do**
  Recompute reliability weights
    $\tilde{o}_t^i \leftarrow \text{CriticFeat}_\phi(o_t^{1:N})$ ▷ current critic
    $\hat{m}_t^i \leftarrow \text{softplus}\big(g_\eta(\text{sg}[\tilde{o}_t^i])\big) + \epsilon_0$
    $\bar{m} \leftarrow \mathbb{E}_{\mathcal{B}}[\hat{m}] + \epsilon_0$
    $u_t^i \leftarrow \bar{m}/(\bar{m} + \hat{m}_t^i)$; $\bar{w}_t^i \leftarrow \text{sg}\big[u_t^i / \mathbb{E}_{\mathcal{B}}[u]\big]$ ▷ $\mathbb{E}_{\mathcal{B}}[\bar{w}] = 1$
    for each minibatch $\mathcal{M}$ **do**
      $\widetilde{A}_t^i \leftarrow \text{Standardize}_{\mathcal{M}}(A_t^i)$
      $\mathcal{L}_{\text{PPO}} \leftarrow \text{PPO}\Big(\bar{w}_t^i \widetilde{A}_t^i\Big)$ ▷ weights only in actor term
      $\mathcal{L}_{\text{pred}} \leftarrow \mathbb{E}_{\mathcal{M}}\big[-\log q_\theta(z_{t+1}^i \mid b_t^i, a_t^i)\big]$
      $\mathcal{L}_{\text{res}} \leftarrow \mathbb{E}_{\mathcal{M}}\big[(\hat{m}_t^i - d_t^i)^2\big]$
      update parameters using $\mathcal{L}_{\text{PPO}} + c_{\text{pred}}\mathcal{L}_{\text{pred}} + c_{\text{res}}\mathcal{L}_{\text{res}}$
    end for
  end for
end while

B.1 IMPLEMENTATION DETAILS

Architecture and inputs. Figure 2 summarizes the overall data flow. Actor parameters are shared across agents, and each actor input includes an agent identifier in addition to the local observation; previous actions are not provided. The centralized critic operates on the joint observation rather than the environment state. At execution, all centralized modules are discarded and each agent acts solely from its local history.

Recurrent encoder. We use a single RWKV block (Peng et al., 2023) as the temporal encoder for both the decentralized actor and the temporal component of the centralized critic. Each block consists of a time-mixing module followed by a gated channel-mixing module. The time-mixing recurrence maintains exponentially decayed per-channel key-value statistics and combines them with the current input through a receptance gate. Our implementation uses RMS normalization, and a GELU channel-mixing network. The recurrent carry consists of the WKV numerator, denominator,

running normalization statistic, and two token-shift buffers, all of fixed width; it is reset at episode boundaries. Consequently, the recurrent state size is independent of the processed history length.

Predictive-loss implementation. All observation coordinates are optimized with the Gaussian mixture NLL, with discrete or categorical coordinates treated as continuous surrogates. The mixture likelihood is evaluated with log-sum-exp, masked at episode boundaries, and normalized by observation dimension. For numerical stability, the log-variance used in the mixture likelihood is clamped, while the policy receives the unclamped detached readout $\zeta_t^i = \mathrm{sg}[\ell(b_t^i)]$, and a small floor ϵ_0 is added to $\bar{m}$ in Eq. (5).

Optimization details. The full training objective is

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{PPO}} + c_{\text{pred}}\mathcal{L}_{\text{pred}} + c_{\text{res}}\mathcal{L}_{\text{res}}, \qquad \mathcal{L}_{\text{PPO}} = -\mathcal{L}_{\pi}^{\text{DEAL}} + c_V\mathcal{L}_V^- c_H\mathcal{H}(\pi), \tag{19}$$

where $\mathcal{L}_V$ is the PPO value loss. The prediction loss updates both the prediction head and the shared actor encoder, whereas the residual loss receives detached critic features and targets and updates only g_η. The weight is also detached before entering the actor surrogate. Architectural widths, optimizer settings, and hyperparameters are reported in Appendix C.4.

B.2 Complexity and Execution Cost

At execution, each DEAL agent maintains a fixed-size recurrent state and processes only its local observation history. For the single RWKV block with hidden width d, the recurrent state requires $O(d)$ memory per agent, independent of episode length, and the dense projections require $O(d^2)$ compute per step. The deployed predictive-scale pathway $\zeta_t^i = \mathrm{sg}[\ell(b_t^i)]$, with $d_z = \dim(o^i)$, adds $O(d\,d_z)$ compute. Thus, across N agents, DEAL requires $O(N(d^2 + d\,d_z))$ total work, while all N policy evaluations are independent and can be executed in parallel. The action-conditioned mixture components, predictive likelihood, centralized critic, and residual-scale head are used only during training. DEAL therefore requires neither cross-agent attention, communication, nor autoregressive action selection at deployment.

For comparison, a recurrent MAPPO actor requires $O(Nd^2)$. For Sable, letting B denote the number of retention blocks, recurrent retention gives $O(BNd^2)$ total inference work for fixed width, but its centralized encoder processes agents jointly and its autoregressive decoder introduces $O(BN)$ sequential depth across agents (Mahjoub et al., 2025). MAT with L Transformer blocks incurs $O(L(N^2d+Nd^2))$ computation from joint self-attention and dense projections, together with $O(LN)$ autoregressive action-generation depth (Wen et al., 2022). MAGPO differs from these centralized-execution policies at deployment. Its centralized autoregressive guider is used for coordinated data collection and training, whereas the deployed learner is factorized and decentralized, giving $O(Nd^2)$ like MAPPO (Li et al., 2026).

Overall, except for MAT, all compared methods have inference complexity that scales linearly with the number of agents N for fixed model width and depth. However, the centralized sequence-model approaches additionally introduce inter-agent sequential dependencies through autoregressive action generation, making their execution latency less favorable than fully parallel decentralized policies as N grows. MAGPO avoids this dependency at deployment through its decentralized learner, but requires an additional centralized autoregressive guider during training, resulting in a larger training-time computational budget. Accordingly, Table 3 shows that DEAL retains near-backbone execution cost while using a substantially smaller deployed actor than Sable and MAT; MAGPO is lighter at deployment, but incurs substantially greater training-time model and computation cost.

Table 3: Training and execution cost. Measurements are averaged over RWARE `tiny-4ag` runs on a single RTX 4090 with an i9-14900K CPU.

	CA-MAPPO	DEAL	Sable	MAT	MAGPO
Actor parameters	0.24M	0.26M	1.08M	0.84M	0.13M
Training parameters	0.61M	0.67M	1.08M	0.84M	1.20M
Training speed (steps/s)	71.4k	63.6k	45.1k	81.1k	34.6k
Execution time, CPU / GPU (ms)	0.035 / 0.105	0.038 / 0.107	0.375 / 0.932	2.287 / 0.863	0.020 / 0.044

C EXPERIMENTAL PROTOCOL

C.1 ENVIRONMENTS

We evaluate on 37 tasks from four cooperative suites. Exact task registrations follow the Sable evaluation protocol (Mahjoub et al., 2025), and the full task lists appear in the per-task tables of Appendix C.3.

RWARE (15 tasks). Robots deliver requested shelves across a shared warehouse (Christianos et al., 2020; Papoudakis et al., 2021). Each agent observes only a small window around itself and reward is given only on successful delivery, so observability is severe and rewards are sparse, while narrow corridors couple the agents' trajectories. Tasks vary grid size (`tiny` through `xlarge`), team size (2-8 agents), and request difficulty (`-hard`).

Connector (4 tasks). Agents navigate to assigned targets on a shared grid while every move leaves behind an impassable trail, so early path choices permanently constrain teammates and greedy routes block the team (Mahjoub et al., 2025). Each agent observes a local view of the grid; grid and team size grow jointly from 5×5 with 3 agents to 15×15 with 23 agents.

SMAX (11 tasks). StarCraft micromanagement in JaxMARL (Rutherford et al., 2024; Samvelyan et al., 2019): each agent controls one allied unit against a scripted opponent and observes only units within its sight range. The set contains eight classic SMAC maps and three SMACv2 maps (Ellis et al., 2023) whose unit compositions and start positions are randomized per episode, making the hidden context explicitly stochastic.

LBF (7 tasks). In Level-Based Foraging (Christianos et al., 2020; Papoudakis et al., 2021), agents and foods carry levels, and a food is collected only when the levels of the agents loading it simultaneously sum to at least its level, so higher-level foods require coordinated joint actions under sparse reward. Tasks vary grid size, team size, food count, and forced cooperation (`-coop`); the `2s-` variants restrict each agent's sight to a two-cell radius.

C.2 PROTOCOL

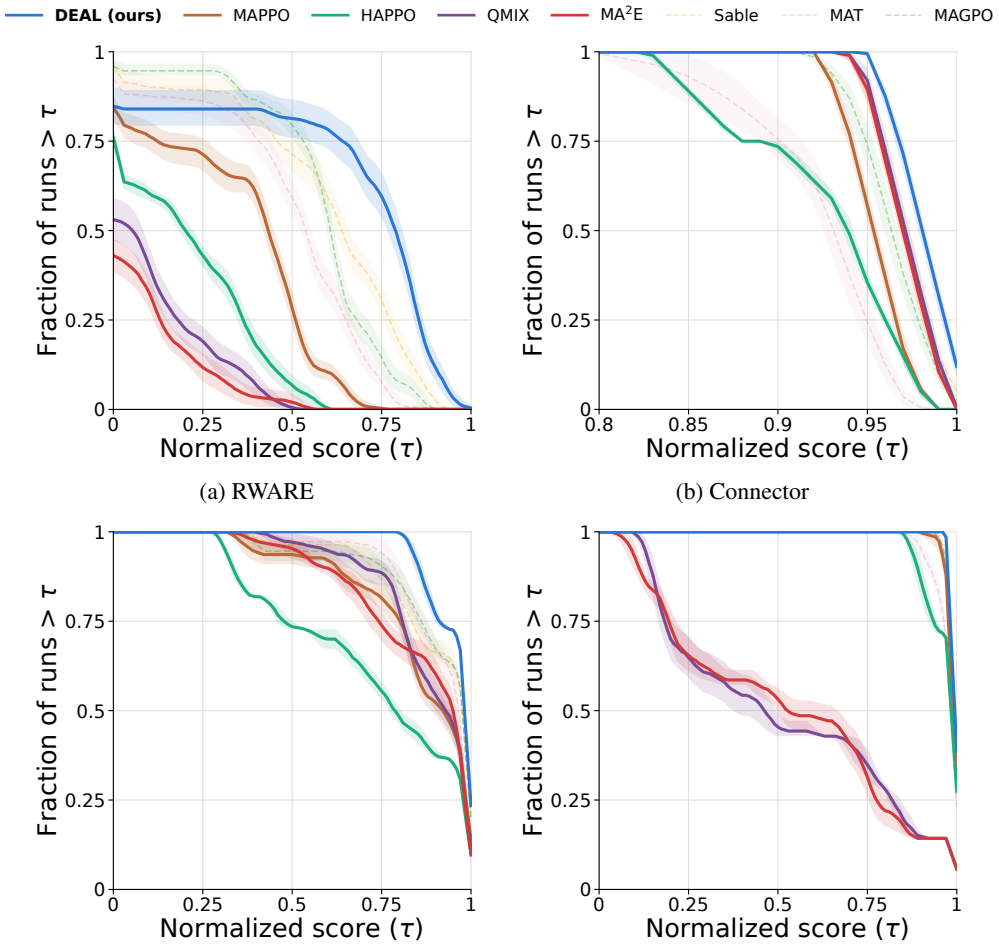
For each method we report raw task scores, per-environment IQM, and performance profiles over all 37 tasks (Figure 7). All main comparisons use the same evaluation frequency, smoothing convention, and final-score aggregation, with episode return as the metric on every suite.

C.3 FULL RESULTS

Table 4 reports per-task absolute returns for every task of the four suites (mean over 10 seeds with 95% confidence intervals), and Figure 8 shows the corresponding per-task learning curves; aggregating over tasks yields Table 1 and Figure 3.

C.4 HYPERPARAMETERS

All implementations build on the JAX-based MARL library Mava (de Kock et al., 2023), and the tuning protocol follows the Sable evaluation protocol (Mahjoub et al., 2025), also used by MAGPO (Li et al., 2026). Algorithms tuned in this study use a per-task budget of 40 trials with the Tree-structured Parzen Estimator from Optuna (Akiba et al., 2019), over the discrete space of Table 6; each trial is scored by mean evaluation return, and tuning seeds are disjoint from reported-result seeds. Where a baseline's per-task values are already published under this protocol (MAPPO, HAPPO, QMIX, MAGPO, MAT, and Sable on RWARE, LBF, and Connector (Mahjoub et al., 2025; Li et al., 2026)), we use them. we tune DEAL and MA^{2}E on all tasks, the latter over a value-based space with the same budget. The DEAL-specific parameters (c_{pred}, c_{res}, and K) are tuned jointly with the shared space, and the same procedure and budget apply to all ablations and sensitivity analyses. All arms of the component evaluation are tuned per task under this procedure and budget. The mixture-size and coefficient sweeps vary one factor at a time around the selected configuration. Table 5 lists the settings shared by all released configurations and Table 6 the tuned space.



(a) RWARE (b) Connector

(c) SMAX (d) LBF

Figure 7: Per-environment performance profiles (Agarwal et al., 2021): the fraction of task-seed runs whose min-max normalized absolute score exceeds a threshold τ, as a function of τ, with 95% stratified bootstrap confidence bands; curves and bands are smoothed for display with a moving average over a 0.05-wide τ window, which preserves monotonicity and the endpoints. Curves that dominate pointwise indicate uniformly better score distributions. The Connector axis starts at $\tau = 0.8$: every method's normalized scores concentrate near the maximum there, and a full $[0, 1]$ axis is an empty panel.

Table 4: Per-task absolute returns across all environments: mean over 10 seeds with 95% confidence intervals. Aggregating the task-level results yields the corresponding environment rows of Table 1.

		CTDE methods					CTCE methods		
Env	Task	DEAL (Ours)	MAPPO	HAPPO	QMIX	MA^{2}E	Sable	MAT	MAGPO
RWARE	tiny-2ag	**24.13**$_{(23.34,24.92)}$	12.28$_{(8.18,15.48)}$	10.93$_{(9.48,12.32)}$	7.71$_{(4.43,10.83)}$	1.84$_{(0.43,3.80)}$	21.17$_{(20.42,21.95)}$	17.06$_{(16.11,18.09)}$	19.70$_{(18.95,20.38)}$
	tiny-2ag-hard	**18.70**$_{(18.31,19.02)}$	13.60$_{(13.18,14.08)}$	9.61$_{(7.47,11.06)}$	1.56$_{(0.00,3.45)}$	3.66$_{(1.11,6.37)}$	15.93$_{(15.50,16.41)}$	13.44$_{(11.74,14.58)}$	16.61$_{(15.99,17.20)}$
	tiny-4ag	41.51$_{(40.01,42.74)}$	26.29$_{(24.38,27.91)}$	17.84$_{(16.42,19.11)}$	16.24$_{(13.98,18.54)}$	14.22$_{(12.05,16.40)}$	**43.56**$_{(41.76,45.10)}$	28.19$_{(27.56,28.83)}$	31.02$_{(29.83,32.23)}$
	tiny-4ag-hard	**31.76**$_{(30.11,33.31)}$	19.01$_{(18.23,19.86)}$	16.66$_{(14.97,18.06)}$	12.93$_{(10.30,15.19)}$	12.81$_{(10.07,15.82)}$	30.97$_{(29.91,31.96)}$	20.54$_{(12.75,27.41)}$	20.49$_{(18.87,22.09)}$
	small-4ag	**18.88**$_{(16.61,20.60)}$	10.52$_{(10.10,11.03)}$	9.64$_{(8.27,11.26)}$	1.94$_{(1.09,2.70)}$	0.53$_{(0.00,1.28)}$	16.47$_{(10.43,20.80)}$	18.27$_{(17.92,18.57)}$	15.09$_{(14.65,15.52)}$
	small-4ag-hard	**13.84**$_{(10.43,16.10)}$	9.44$_{(9.22,9.66)}$	6.50$_{(5.85,7.19)}$	3.09$_{(2.28,3.76)}$	1.73$_{(1.11,2.31)}$	12.02$_{(11.29,12.78)}$	9.68$_{(7.46,10.97)}$	11.48$_{(11.22,11.72)}$
	medium-4ag	**14.06**$_{(13.01,15.07)}$	7.82$_{(5.60,9.49)}$	4.04$_{(3.66,4.44)}$	0.35$_{(0.00,0.85)}$	0.00$_{(0.00,0.00)}$	12.74$_{(11.72,13.41)}$	7.62$_{(5.18,9.90)}$	11.46$_{(10.80,12.21)}$
	medium-4ag-hard	**8.84**$_{(5.61,11.21)}$	2.80$_{(1.13,4.55)}$	3.75$_{(3.36,4.14)}$	0.76$_{(0.19,1.35)}$	0.19$_{(0.00,0.57)}$	6.79$_{(5.90,7.55)}$	4.64$_{(2.96,6.08)}$	8.49$_{(8.20,8.86)}$
	medium-6ag	**20.47**$_{(19.00,22.04)}$	12.13$_{(11.83,12.48)}$	4.99$_{(4.47,5.48)}$	1.48$_{(0.93,1.99)}$	3.21$_{(2.32,4.00)}$	12.97$_{(12.27,13.53)}$	13.32$_{(12.93,13.70)}$	14.78$_{(12.91,16.94)}$
	large-4ag	**7.85**$_{(5.06,10.07)}$	3.02$_{(1.58,4.38)}$	0.61$_{(0.28,0.95)}$	0.40$_{(0.00,0.93)}$	0.31$_{(0.00,0.81)}$	6.22$_{(5.03,7.14)}$	4.61$_{(4.46,4.78)}$	7.63$_{(7.04,8.24)}$
	large-4ag-hard	2.96$_{(1.41,4.48)}$	0.00$_{(0.00,0.01)}$	0.00$_{(0.00,0.01)}$	0.21$_{(0.00,0.63)}$	0.21$_{(0.00,0.64)}$	3.46$_{(2.22,4.46)}$	2.28$_{(1.09,3.42)}$	**4.56**$_{(4.18,4.96)}$
	large-8ag	10.78$_{(4.94,16.63)}$	8.35$_{(7.95,8.77)}$	0.00$_{(0.00,0.01)}$	2.59$_{(1.32,3.84)}$	2.90$_{(2.12,3.50)}$	11.01$_{(10.69,11.33)}$	**14.72**$_{(14.27,15.24)}$	10.40$_{(10.06,10.71)}$
	large-8ag-hard	7.64$_{(4.30,10.52)}$	3.38$_{(1.52,5.33)}$	0.00$_{(0.00,0.00)}$	0.44$_{(0.00,0.88)}$	0.00$_{(0.00,0.00)}$	**9.22**$_{(8.93,9.52)}$	9.07$_{(8.61,9.48)}$	8.66$_{(8.25,9.00)}$
	xlarge-4ag	4.22$_{(2.37,5.84)}$	3.73$_{(2.93,4.40)}$	0.00$_{(0.00,0.00)}$	0.90$_{(0.15,1.72)}$	0.00$_{(0.00,0.00)}$	3.76$_{(2.28,5.08)}$	4.71$_{(4.42,4.96)}$	**5.74**$_{(5.29,6.16)}$
	xlarge-4ag-hard	**1.62**$_{(0.62,2.63)}$	0.00$_{(0.00,0.00)}$	0.00$_{(0.00,0.00)}$	0.00$_{(0.00,0.00)}$	0.00$_{(0.00,0.00)}$	0.70$_{(0.01,1.74)}$	0.39$_{(0.01,1.07)}$	0.31$_{(0.00,0.74)}$
Connector	con-5x5x3a	**0.87**$_{(0.86,0.87)}$	0.83$_{(0.82,0.84)}$	0.84$_{(0.84,0.84)}$	0.85$_{(0.85,0.86)}$	0.86$_{(0.85,0.86)}$	0.85$_{(0.85,0.86)}$	0.81$_{(0.80,0.82)}$	0.82$_{(0.81,0.82)}$
	con-7x7x5a	**0.81**$_{(0.81,0.81)}$	0.75$_{(0.74,0.76)}$	0.77$_{(0.77,0.78)}$	0.80$_{(0.79,0.80)}$	0.80$_{(0.79,0.80)}$	0.79$_{(0.79,0.80)}$	0.75$_{(0.74,0.76)}$	0.79$_{(0.78,0.79)}$
	con-10x10x10a	**0.77**$_{(0.76,0.77)}$	0.70$_{(0.70,0.70)}$	0.64$_{(0.64,0.65)}$	0.75$_{(0.75,0.76)}$	0.74$_{(0.73,0.74)}$	0.74$_{(0.74,0.74)}$	0.65$_{(0.63,0.67)}$	0.74$_{(0.73,0.75)}$
	con-15x15x23a	**0.73**$_{(0.73,0.74)}$	0.63$_{(0.62,0.64)}$	0.20$_{(0.20,0.21)}$	0.68$_{(0.67,0.68)}$	0.66$_{(0.65,0.67)}$	0.70$_{(0.70,0.71)}$	0.25$_{(0.18,0.31)}$	0.69$_{(0.68,0.69)}$
SMAX	2s3z	**2.00**$_{(2.00,2.00)}$	2.00$_{(1.99,2.00)}$	1.99$_{(1.99,2.00)}$	2.00$_{(1.99,2.00)}$	1.97$_{(1.95,1.98)}$	2.00$_{(2.00,2.00)}$	1.99$_{(1.99,2.00)}$	**2.00**$_{(2.00,2.00)}$
	3s5z	**2.00**$_{(2.00,2.00)}$	2.00$_{(2.00,2.00)}$	1.98$_{(1.97,1.98)}$	2.00$_{(2.00,2.00)}$	1.99$_{(1.98,1.99)}$	2.00$_{(1.99,2.00)}$	2.00$_{(2.00,2.00)}$	1.99$_{(1.98,2.00)}$
	3s5z_vs_3s6z	**1.99**$_{(1.99,1.99)}$	1.90$_{(1.86,1.94)}$	1.44$_{(1.36,1.53)}$	1.79$_{(1.75,1.82)}$	1.76$_{(1.73,1.79)}$	1.94$_{(1.93,1.95)}$	1.93$_{(1.91,1.95)}$	1.95$_{(1.93,1.96)}$
	3s_vs_5z	**2.00**$_{(2.00,2.00)}$	1.98$_{(1.98,1.99)}$	1.94$_{(1.93,1.95)}$	1.95$_{(1.93,1.97)}$	1.95$_{(1.93,1.96)}$	1.98$_{(1.98,1.99)}$	1.96$_{(1.95,1.97)}$	1.98$_{(1.98,1.99)}$
	6h_vs_8z	**2.00**$_{(2.00,2.00)}$	2.00$_{(1.99,2.00)}$	1.99$_{(1.99,2.00)}$	1.85$_{(1.64,1.96)}$	1.94$_{(1.92,1.96)}$	2.00$_{(2.00,2.00)}$	1.99$_{(1.98,1.99)}$	2.00$_{(2.00,2.00)}$
	5m_vs_6m	**1.96**$_{(1.96,1.97)}$	1.05$_{(0.75,1.38)}$	0.71$_{(0.69,0.73)}$	1.75$_{(1.62,1.87)}$	1.81$_{(1.63,1.95)}$	1.47$_{(1.13,1.80)}$	1.60$_{(1.29,1.88)}$	1.15$_{(0.89,1.45)}$
	10m_vs_11m	**1.99**$_{(1.98,1.99)}$	1.28$_{(1.22,1.33)}$	0.94$_{(0.92,0.98)}$	1.96$_{(1.93,1.98)}$	1.86$_{(1.70,1.99)}$	1.80$_{(1.62,1.97)}$	1.86$_{(1.70,1.97)}$	1.98$_{(1.96,1.99)}$
	27m_vs_30m	1.96$_{(1.95,1.97)}$	1.81$_{(1.74,1.87)}$	1.74$_{(1.70,1.78)}$	1.70$_{(1.63,1.78)}$	1.83$_{(1.69,1.91)}$	**2.00**$_{(1.99,2.00)}$	1.87$_{(1.80,1.92)}$	1.99$_{(1.98,1.99)}$
	smacv2_5_units	**1.80**$_{(1.78,1.83)}$	1.70$_{(1.68,1.71)}$	1.60$_{(1.58,1.62)}$	1.68$_{(1.65,1.70)}$	1.42$_{(1.38,1.46)}$	1.72$_{(1.70,1.74)}$	1.77$_{(1.75,1.78)}$	1.78$_{(1.76,1.80)}$
	smacv2_10_units	**1.75**$_{(1.73,1.77)}$	1.70$_{(1.68,1.71)}$	1.33$_{(1.27,1.37)}$	1.60$_{(1.56,1.63)}$	1.51$_{(1.47,1.55)}$	1.52$_{(1.48,1.56)}$	1.62$_{(1.59,1.65)}$	1.68$_{(1.65,1.73)}$
	smacv2_20_units	**1.55**$_{(1.53,1.57)}$	1.42$_{(1.38,1.45)}$	0.60$_{(0.59,0.61)}$	1.11$_{(0.99,1.23)}$	0.89$_{(0.80,0.97)}$	1.47$_{(1.40,1.52)}$	1.49$_{(1.46,1.53)}$	1.44$_{(1.39,1.49)}$
LBF	10x10-3p-3f	**1.00**$_{(1.00,1.00)}$	1.00$_{(1.00,1.00)}$	1.00$_{(1.00,1.00)}$	0.30$_{(0.20,0.40)}$	0.57$_{(0.51,0.64)}$	1.00$_{(1.00,1.00)}$	0.99$_{(0.99,1.00)}$	1.00$_{(1.00,1.00)}$
	15x15-3p-5f	0.99$_{(0.98,0.99)}$	0.97$_{(0.95,0.97)}$	0.91$_{(0.90,0.93)}$	0.15$_{(0.13,0.16)}$	0.10$_{(0.09,0.11)}$	0.96$_{(0.96,0.97)}$	0.91$_{(0.90,0.92)}$	**0.99**$_{(0.99,0.99)}$
	15x15-4p-3f	**1.00**$_{(1.00,1.00)}$	1.00$_{(0.99,1.00)}$	1.00$_{(0.99,1.00)}$	0.39$_{(0.34,0.44)}$	0.27$_{(0.23,0.32)}$	1.00$_{(1.00,1.00)}$	0.99$_{(0.99,1.00)}$	1.00$_{(1.00,1.00)}$
	15x15-4p-5f	**1.00**$_{(1.00,1.00)}$	0.98$_{(0.97,0.98)}$	0.89$_{(0.88,0.90)}$	0.20$_{(0.17,0.23)}$	0.19$_{(0.17,0.21)}$	0.99$_{(0.99,0.99)}$	0.97$_{(0.96,0.97)}$	0.99$_{(0.99,1.00)}$
	2s-10x10-3p-3f	**1.00**$_{(1.00,1.00)}$	1.00$_{(1.00,1.00)}$	0.99$_{(0.99,0.99)}$	0.74$_{(0.72,0.76)}$	0.74$_{(0.71,0.76)}$	0.99$_{(0.99,1.00)}$	0.97$_{(0.96,0.97)}$	0.97$_{(0.97,0.98)}$
	2s-8x8-2p-2f-coop	**1.00**$_{(1.00,1.00)}$	**1.00**$_{(1.00,1.00)}$	1.00$_{(1.00,1.00)}$	0.84$_{(0.83,0.86)}$	0.80$_{(0.76,0.83)}$	**1.00**$_{(1.00,1.00)}$	1.00$_{(0.99,1.00)}$	1.00$_{(1.00,1.00)}$
	8x8-2p-2f-coop	**1.00**$_{(1.00,1.00)}$	1.00$_{(1.00,1.00)}$	**1.00**$_{(1.00,1.00)}$	1.00$_{(1.00,1.00)}$	1.00$_{(1.00,1.00)}$	1.00$_{(1.00,1.00)}$	1.00$_{(1.00,1.00)}$	**1.00**$_{(1.00,1.00)}$

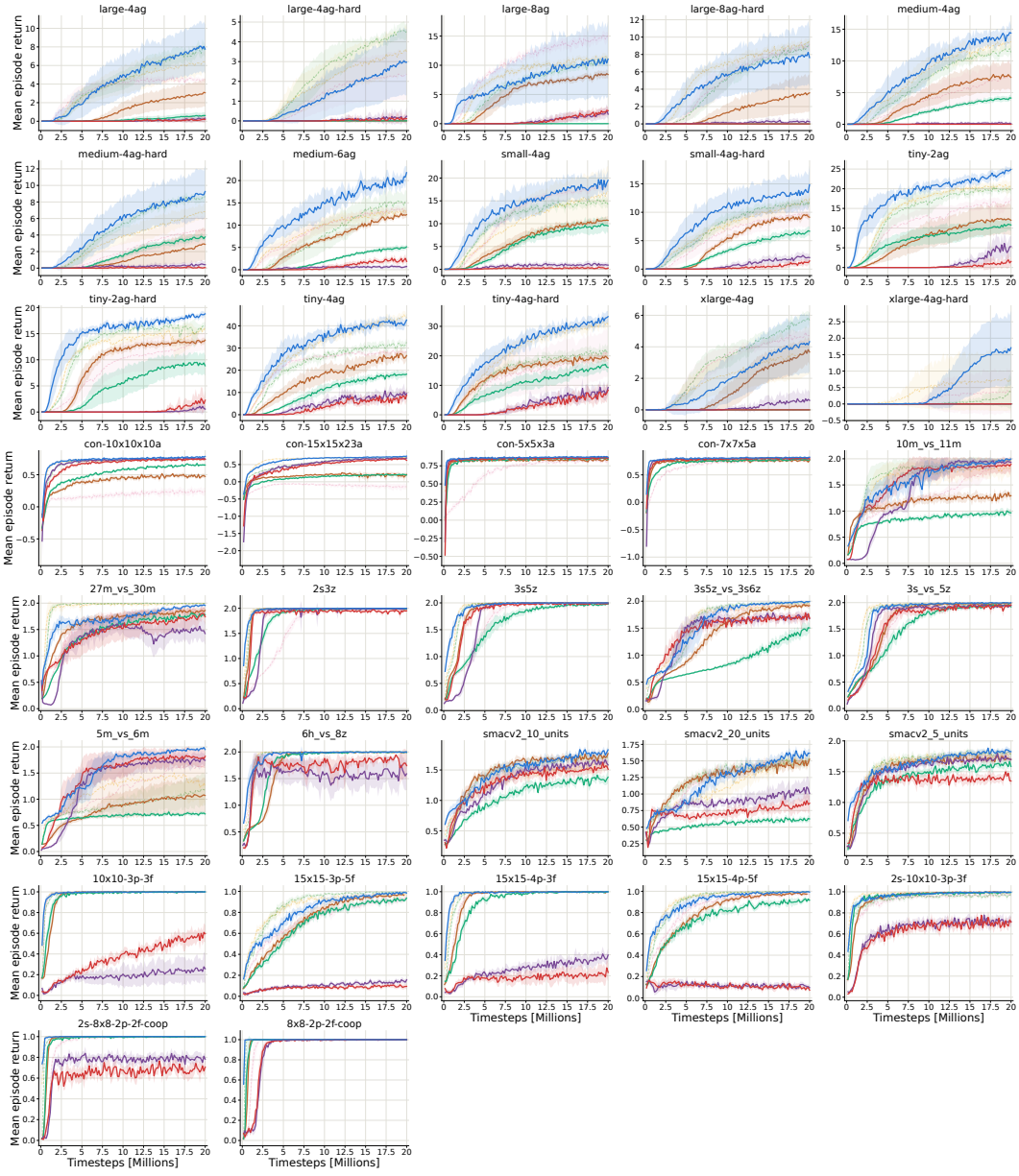


Figure 8: Per-task learning curves. Mean return over 10 seeds; shading is a 95% confidence interval.

Table 5: Default settings shared by all released DEAL configurations.

Setting	Value
Discount γ	0.99
Hidden size (all networks)	128
Cross-agent attention heads	2
BPTT chunk length	8
Value coefficient c_V	0.5
Activation function	GeLU
Optimizer	AdamW
Linear LR decay	True

Table 6: Hyperparameter search spaces used for DEAL

Parameter	Values
PPO epochs	{4, 8, 16}
Number of minibatches	{1,2, 4, 8}
Entropy coefficient c_H	{0, 1e-3, 3e-3, 1e-2}
PPO clip ϵ	{0.05, 0.1, 0.2}
Maximum gradient norm	{0.5, 5, 10}
Actor / critic learning rate	{5e-4, 1e-3, 2e-3}
Predictive coefficient c_{pred}	{0.05, 0.1}
Residual-scale coefficient c_{res}	{1, 5, 10}
Mixture size K	{2, 4}

C.5 PARTIAL-OBSERVABILITY STRESS TEST

The LBF sight-range sweep (Figures 6c and 9a) uses $\text{fov} \in \{1, 2, 3, 4, 6, 8, 15\}$ on `15x15-3p-5f`. DEAL's relative gain over MAPPO falls monotonically with sight, from 63% at $\text{fov} = 1$ through parity at $\text{fov} = 8$ to -4% at full sight under the matched configuration; absolute gains peak at $\text{fov} = 2$ rather than $\text{fov} = 1$, because at the lowest sight every method's absolute return collapses and ratios inflate.

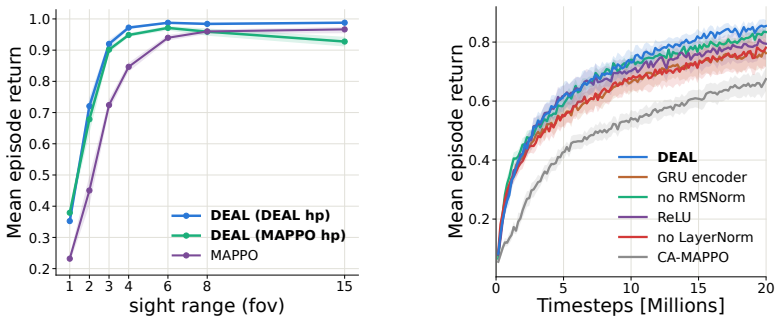
DEAL is run under two configurations, its own selected full-sight configuration and MAPPO's tuned PPO configuration, with MAPPO under that same published configuration. The relative gains of Figure 6c use the MAPPO-configuration DEAL at every sight value, so the pair shares every PPO hyperparameter throughout the sweep. Figure 9a plots the absolute episode return of the three arms against sight: the two DEAL curves nearly coincide, both degrade far more slowly than MAPPO as sight shrinks, and they separate mainly at full sight, where DEAL under its own tuned configuration surpasses MAPPO. The sight-sweep conclusion therefore does not depend on DEAL-specific tuning, and the matched-configuration deficit at full sight is consistent with sensitivity to the borrowed configuration and the additional predictive overhead.

D ABLATIONS AND DIAGNOSTICS

D.1 EXTENDED ABLATIONS

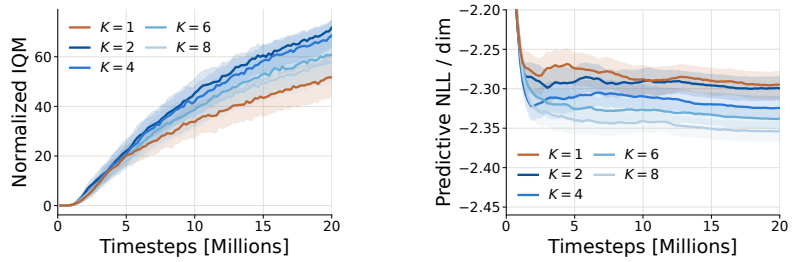
Figure 9b varies the recurrent architecture, normalization, and activation choices of the DEAL on the four-task ablation subset (`tiny-4ag`, `medium-4ag`, `3s5z`, `smacv2_5_units`), with min-max normalized mean episode return over training. Some alternatives modestly reduce performance but all variants remain substantially above the CA-MAPPO baseline. Thus, these implementation choices appear to provide mild stabilization rather than account for the overall performance gains.

Figure 10 sweeps the mixture size K on all 15 RWARE tasks around each task's selected configuration. Small mixtures ($K = 2$ and $K = 4$) perform best, while both single Gaussian and the larger mixtures perform worse. We hypothesize that a small mixture is enough to capture the main future outcomes that matter for coordination. With larger K, the model may spend more of its fixed capacity on



(a) *Partial-observability stress test* on LBF `15x15-3p-5f` (10 seeds, 95% CIs).

(b) *Implementation details* on the four-task ablation subset (10 seeds, 95% CIs).



(a) Return

(b) Predictive NLL

Figure 10: *Mixture-size sweep* on the 15 RWARE tasks. **(a)** Normalized return over training (IQM over tasks and seeds on the normalization of Table 2, 95% stratified bootstrap CI). **(b)** Per-dimension predictive NLL over training of the runs that did not collapse (mean, 95% CI).

fine-grained predictive details that are less useful for control. This is consistent with the NLL results: predictive NLL keeps improving as K increases, even though return decreases.

D.2 Prediction Target Details

The target-swap variants in Table 2 retain the same actor trunk and mixture-head architecture, varying only the predictive target and its output dimension. *Next state* predicts the next global state from the information feature and joint action, *state reconstruction* predicts the current global state, and *past trajectory* reconstructs all agents' observations from the preceding three steps.

Diagnostic protocol for Figure 6b. Five checkpointed `tiny-4ag` runs (20 evenly spaced checkpoints each) are replayed on fresh rollouts, and samples are grouped by the content of the agent's own 3×3 window: empty, shelves only, or a teammate in view. Each curve is the group's mean realized residual $|G^{\lambda} - V|$ or predicted $\hat{m}$ relative to the rollout mean, averaged over seeds and smoothed over three checkpoints.

Lower tail across tasks. We define a near-collapse run as one with a normalized return below 0.05, using the per-task min-max normalization from Table 2. The same trend holds for thresholds ranging from 0.02 to 0.10. Table 7 gives the per-task counts over 10 seeds.

D.3 Hyperparameter Sensitivity

We assess hyperparameter sensitivity by varying one coefficient at a time while keeping all other hyperparameters fixed. The analysis uses the seven tasks for which $c_{\text{pred}} = 0.1$ and $c_{\text{res}} = 5$ were selected; values beyond the search space of Table 6 are included for sensitivity only. Figure 11 summarizes the results from 10 seeds per setting. Each score averages the per-task IQM of normalized return across the seven tasks, with 95% stratified bootstrap CIs. Square markers at zero correspond to the respective single-component arms in Table 2.

Table 7: *Near-collapse runs per RWARE task* (out of 10 seeds): runs whose absolute return is below 0.05 after the per-task min-max normalization of Table 2.

Task	CA-MAPPO	+ weighting	+ predictive	DEAL
tiny-2ag	0	1	0	0
tiny-2ag-hard	3	1	1	0
tiny-4ag	0	0	0	0
tiny-4ag-hard	0	0	0	0
small-4ag	1	0	1	0
small-4ag-hard	2	1	3	1
medium-4ag	2	0	2	0
medium-4ag-hard	4	1	5	2
medium-6ag	1	0	0	0
large-4ag	4	3	7	2
large-4ag-hard	5	5	7	4
large-8ag	3	2	6	4
large-8ag-hard	4	4	6	3
xlarge-4ag	9	3	6	3
xlarge-4ag-hard	10	9	9	5
Total (of 150)	48 (32%)	30 (20%)	53 (35%)	24 (16%)

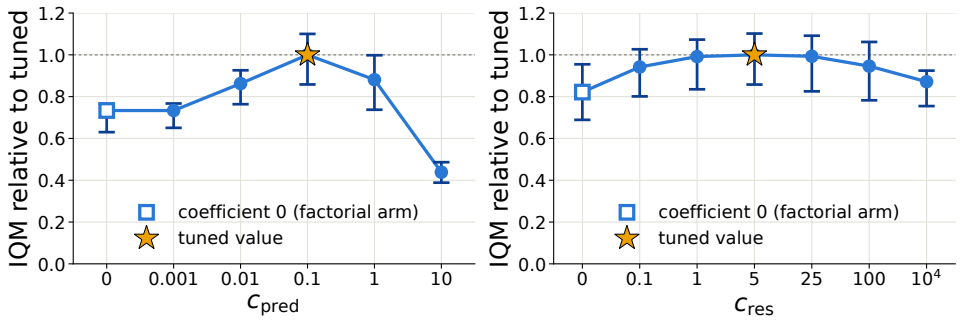


Figure 11: Sensitivity to the predictive and residual loss coefficients.

Performance varies more strongly with c_{pred} and is highest near the selected value. In contrast, c_{res} is robust across a broad range of positive values, with noticeable degradation only at the extremes.

E LIMITATIONS AND FUTURE WORK

Appendix A discusses the theoretical scope of DEAL. Empirically, our evaluation is limited to four simulated cooperative benchmark suites, and therefore does not establish that the same gains will hold in other MARL regimes or real-world settings. In addition, the predictive model uses a continuous Gaussian-mixture likelihood even for observation dimensions that are discrete or categorical, serving as a surrogate rather than a distribution matched to their native structure.

Beyond these limitations, the current residual-scale signal summarizes collection-time return residuals with a single scalar prediction. This signal can reflect multiple sources of variation, including return stochasticity, value predictability, and action-dependent differences in the expected λ-return target. Disentangling these factors may enable more selective weighting criteria while retaining the same execution-aligned training principle.

The current policy also uses only a compact predictive-scale readout ζ from the decentralized predictive model. Richer summaries of the predictive distribution, including component- or action-dependent statistics, as well as likelihoods better matched to discrete observation structure, are promising directions for extending the method without relaxing its decentralized execution constraints.

These extensions would broaden how predictive information and learning signals are used without changing the central principle of aligning training with decentralized execution.