CGPO: COUNTERFACTUAL CREDIT FOR MULTI-AGENT DIFFUSION POLICY OPTIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusion policies offer expressive action representations for cooperative multi-agent control, but learning from shared team rewards requires distinguishing each agent's contribution to joint performance. We introduce Counterfactual Generative Policy Optimization (CGPO), an on-policy actor–critic framework that trains decentralized diffusion policies through counterfactual credit assignment. For each agent, a centralized critic estimates a counterfactual advantage by comparing the value of its executed action with the average value of alternatives independently sampled from its behavior policy, while holding teammates' actions fixed. This advantage weights PPO-style updates at each denoising step along the diffusion trajectory that generated the executed action. Our theoretical analysis shows that estimating the counterfactual baseline with M independent samples introduces an additional gradient covariance term that scales as $1/M$, and identifies conditions under which counterfactual credit reduces policy-gradient variance. Experiments on VMAS, MaMuJoCo, and MPE show that CGPO achieves the highest mean returns among the compared methods, with gains of 6.7–46.6% over the strongest compared baselines on MaMuJoCo.

1 INTRODUCTION

Cooperative multi-agent reinforcement learning studies how agents can learn to maximize a common team return. We consider online cooperative control with continuous actions and decentralized policies conditioned on local observations. In this setting, the value of an agent's action depends on its teammates' choices, while its decisions rely on local information. Centralized training with decentralized execution (CTDE) accommodates this asymmetry by using joint information to train actors that execute locally (Lowe et al., 2017). Advances in centralized actor–critic learning have made this paradigm effective across cooperative control tasks (Yu et al., 2022; Kuba et al., 2021; Foerster et al., 2018). Within this online setting, policy representation and credit assignment remain closely related: agents need both suitable action distributions and informative signals for improving them from shared feedback.

Research on policy representation has increasingly moved beyond simple parametric distributions toward expressive generative models. Diffusion policies can represent multimodal conditional action distributions, with applications spanning behavior learning and value-guided online control (Chi et al., 2023; Psenka et al., 2024). In multi-agent learning, joint trajectory modeling has also used generative models to address coordination in offline data (Zhu et al., 2024; Yuan et al., 2025). For online off-policy multi-agent learning, OMAD combines decentralized diffusion actors with a tractable entropy objective and centralized distributional value guidance (Li et al., 2026c). Together, these developments establish ways to learn expressive policies and coordinate their behavior. They also motivate a more specific question about the training signal: *how should a shared team evaluation be converted into an agent-specific comparison for updating a generative policy?*

The underlying difficulty is that team return evaluates a joint decision, whereas policy improvement is carried out by individual agents. A favorable outcome can arise from different combinations of actions and does not reveal whether each executed choice was better than the alternatives available to that agent. Value factorization provides one route by structuring the centralized value through agent utilities, including extensions to continuous control (Sunehag et al., 2018; Rashid et al., 2018; Peng et al., 2021). Counterfactual credit addresses this ambiguity by holding teammates' actions fixed and

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.

evaluating an agent's action relative to alternatives from that agent's policy (Foerster et al., 2018). Factorized-policy and multi-agent baseline analyses further characterize how conditioning on other action components shapes gradient variance (Wu et al., 2018; Kuba et al., 2021). For diffusion agents, this motivates a complementary direction to improving expressiveness: improving how team values guide the learning of an otherwise unchanged diffusion representation.

Building on recent advances in generative-policy optimization (Ren et al., 2025; McAllister et al., 2026; Ding et al., 2025), we introduce COUNTERFACTUAL GENERATIVE POLICY OPTIMIZATION (CGPO), an *on-policy* CTDE framework that trains decentralized diffusion policies through counterfactual credit assignment. For each agent, a centralized critic compares the value of its executed action with the expected value of alternatives drawn from its behavior policy, while holding teammates' actions fixed. We estimate the continuous-action counterfactual baseline via Monte Carlo sampling and use the resulting advantage to weight PPO-style updates at each diffusion denoising step. Under an additive team-value model, our analysis in Corollary 3 derives an explicit sampling-budget threshold above which counterfactual credit reduces policy-gradient variance relative to a global team advantage. Experiments on VMAS, MPE, and MaMuJoCo evaluate the method's performance, while paired ablations and frozen-policy diagnostics examine the contribution of counterfactual credit and its sampling noise.

Our contributions are threefold:

- We develop CGPO, a counterfactual training procedure for decentralized diffusion policies that anchors individual credit to behavior-policy comparisons and reuses it across factual-path updates.
- Under an additive team-value model, we derive an explicit sampling-budget threshold above which sampled counterfactual credit reduces policy-gradient variance relative to a global team advantage. Frozen-policy diagnostics further quantify the noise introduced by counterfactual sampling.
- We demonstrate strong performance on cooperative control benchmarks and higher mean returns from counterfactual credit in paired ablations with the diffusion actor held fixed.

2 PROBLEM FORMULATION AND PRELIMINARIES

2.1 COOPERATIVE MARL UNDER CTDE

We consider online cooperative control with N agents, continuous actions, and a shared team objective. At environment step t, the physical state is s_t. Agent i receives a local observation and chooses $a_{i,t} \in \mathcal{A}_i$ using its local history $\tau_{i,t}$, represented by actor input $o_{i,t} = \psi_i(\tau_{i,t})$. The joint action $\mathbf{a}_t = (a_{1,t}, \ldots, a_{N,t})$ produces team reward r_t and the next state according to $P(s_{t+1} \mid s_t, \mathbf{a}_t)$. When an environment supplies individual rewards, we use their mean as the team reward. The objective is

$$J(\theta) = \mathbb{E}_{\pi_\theta}\left[\sum_{t\geq 0} \gamma^t r_t\right], \qquad \gamma \in [0, 1). \tag{1}$$

Centralized training with decentralized execution (CTDE) uses joint information to evaluate actions during learning while preserving local decision-making at execution (Lowe et al., 2017; Yu et al., 2022). The critic receives centralized context c_t, such as joint observations, and the joint action $\mathbf{a}_t$. Each actor uses only $o_{i,t}$; neither the critic nor teammates' realized actions are required to execute its policy. The joint policy factorizes as

$$\pi_\theta(\mathbf{a} \mid \mathbf{o}) = \prod_{i=1}^{N} \pi_{\theta_i}(a_i \mid o_i). \tag{2}$$

Actor parameters may be shared, with agents sampling independent latent noise conditional on their local inputs. For analysis, let $H_t = (s_t, \boldsymbol{\tau}_t)$ denote the augmented Markov state, where $\boldsymbol{\tau}_t$ collects all agents' local histories.

The team action value $Q^\pi(H, \mathbf{a})$ is the expected discounted return after executing $\mathbf{a}$ at H and following π thereafter. In practice, we approximate this value with a learned critic $Q_\phi(c, \mathbf{a})$, where c

denotes the centralized information available during training (Lowe et al., 2017; Yu et al., 2022). We use the environment-provided global state as c_t for MPE and MaMuJoCo, and concatenated joint observations for VMAS.

2.2 Diffusion Policies and Path-Based Optimization

Diffusion policies represent an action distribution through iterative denoising of latent noise. This construction can model multimodal conditional distributions (Chi et al., 2023), while retaining tractable conditional densities at individual generation steps. Following the path-space view used by DPPO (Ren et al., 2025), one environment decision contains an internal Markov chain conditioned on the actor input. Agent i generates a_i through a path $\xi_i = (x_{i,0}, \ldots, x_{i,K})$:

$$p_{\theta_i}(\xi_i \mid o_i) = \rho(x_{i,0}) \prod_{k=0}^{K-1} p_{\theta_i,k}(x_{i,k+1} \mid x_{i,k}, o_i), \qquad a_i = h_i(x_{i,K}), \tag{3}$$

In our diffusion actor, $\rho = \mathcal{N}(0, \sigma_{\text{init}}^2 I)$ and each transition is Gaussian with a learned denoising mean and positive variance. Here k indexes generation steps within one environment decision, distinct from environment time t. The action policy π_{θ_i} is induced by sampling $\xi_i \sim p_{\theta_i}$ and executing $h_i(x_{i,K})$; ρ and h_i are parameter independent. The marginal action density may be unavailable, but transition densities give the path score

$$S_i = \nabla_{\theta_i} \log p_{\theta_i}(\xi_i \mid o_i) = \sum_{k=0}^{K-1} \nabla_{\theta_i} \log p_{\theta_i,k}(x_{i,k+1} \mid x_{i,k}, o_i). \tag{4}$$

For shared actor parameters θ, each S_i is instead defined as $\nabla_\theta \log p_\theta(\xi_i \mid o_i)$; the joint path score is $\sum_i S_i$.

DPPO uses these tractable transitions to optimize diffusion policies with PPO-style updates (Ren et al., 2025). This supplies the optimization structure for CGPO; the remaining multi-agent question is how to construct the credit that weights each agent's path.

2.3 Counterfactual Credit

Define $V^\pi(H) = \mathbb{E}_{\mathbf{a}\sim\pi} Q^\pi(H, \mathbf{a})$ and $A^\pi = Q^\pi - V^\pi$. The team advantage evaluates the joint outcome without isolating an individual agent's action. To compute credit for agent i, COMA-style comparison holds the other agents' actions fixed and varies only a_i (Foerster et al., 2018). The policy-conditional counterfactual baseline and credit are

$$b_i^\pi(H, \mathbf{a}_{-i}) = \mathbb{E}_{\widetilde{a}_i \sim \pi_i(\cdot \mid o_i)} Q^\pi(H, \mathbf{a}_{-i}, \widetilde{a}_i), \tag{5}$$

$$A_i^{\text{cf},\pi}(H, \mathbf{a}) = Q^\pi(H, \mathbf{a}) - b_i^\pi(H, \mathbf{a}_{-i}). \tag{6}$$

Only agent i's action is marginalized; H and teammates' actions remain fixed. Writing $C_i = (H, \mathbf{a}_{-i})$, the baseline is the conditional expectation $b_i^\pi = \mathbb{E}[Q^\pi \mid C_i]$. The resulting credit measures whether the executed action is better than the agent's policy-distributed alternatives in the same team context. We next connect this action-level comparison to the diffusion path gradient, then construct its finite-sample implementation.

3 Counterfactual Generative Policy Optimization

CGPO trains decentralized diffusion policies by evaluating each executed action against its behavior-policy alternatives. We first establish how this action-level credit enters the path policy gradient, then construct a finite-sample estimate and use it in transition-level PPO. Figure 1 illustrates the method, and Algorithm 1 summarizes training.

3.1 From Counterfactual Credit to Path Gradients

Applying the counterfactual baseline argument of Foerster et al. (2018) to the diffusion path score in equation 4 yields the following specialization of the policy-gradient theorem (Sutton et al., 1999). Let d^π denote normalized discounted occupancy. Formal assumptions and the proof appear in Appendix A.

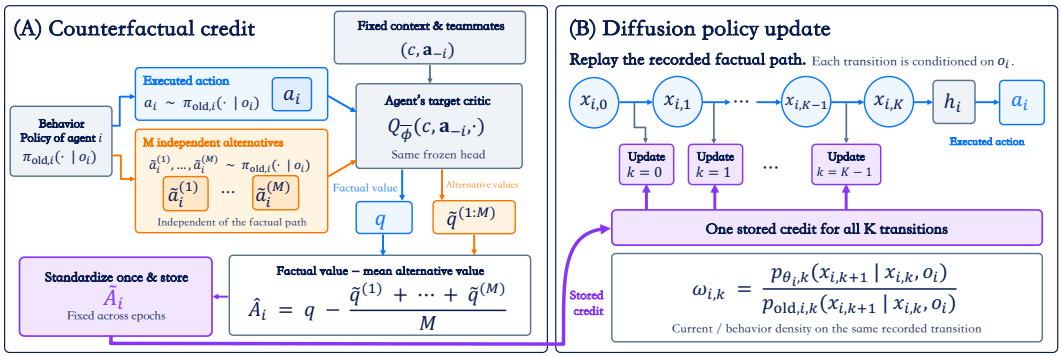


Figure 1: Overview of CGPO. (A) Holding context and teammates' actions fixed, the target critic compares factual value q with the mean value of M independent behavior-policy alternatives, $\tilde{q}^{(m)}$. The resulting credit is standardized once and stored. (B) The same stored credit weights updates along all K transitions of the recorded factual diffusion path and remains fixed across optimization epochs.

Proposition 1 (Counterfactual policy gradient in diffusion path space)**.** *For separately parameterized actors, the team action value Q^π and counterfactual advantage in equation 6 give*

$$\nabla_{\theta_i} J(\theta) = \frac{1}{1-\gamma} \mathbb{E}_{d^\pi, p_\theta} \left[A_i^{\text{cf},\pi} \sum_{k=0}^{K-1} \nabla_{\theta_i} \log p_{\theta_i,k}(x_{i,k+1} \mid x_{i,k}, o_i) \right]. \tag{7}$$

Proposition 1 expresses the counterfactual update through tractable diffusion transition scores: the same action-level advantage weights every denoising step along the path that generated the executed action. This avoids evaluating the marginal action likelihood.

For continuous-action diffusion policies, we estimate the counterfactual baseline by Monte Carlo sampling. Whether this yields a net variance benefit depends on the sampling noise introduced relative to the teammate variation removed. Section 4 characterizes this tradeoff and, under an additive team-value model, derives a sampling-budget threshold for lower policy-gradient variance than a global team advantage. We next describe the practical update using a learned critic and recorded behavior-policy paths.

3.2 Finite-Sample Credit Construction

We first construct the comparison in Figure 1A. We use a centralized critic $Q_\phi(c, \mathbf{a}) \in \mathbb{R}$ and its target counterpart $Q_{\bar{\phi}}$, where c denotes the centralized training information introduced in Section 2.1. For each agent's counterfactual comparison, the executed and alternative joint actions are evaluated using the same fixed target critic. In optimization, ϕ denotes the collection of all head parameters; scalar evaluations use the head for agent i. Both factual and alternative joint actions are evaluated by this same head. Each collection uses a behavior snapshot θ_{old}; credit is computed with this same actor and a fixed target-critic snapshot. Holding c and teammates' actions constant makes each comparison specific to agent i's executed action.

For each factual tuple and agent i, sample M independent alternative actions $\tilde{a}_i^{(m)} \sim \pi_{\text{old},i}(\cdot \mid o_i)$ using fresh generation noise. Replace only a_i by the resulting $\tilde{a}_i^{(m)}$ and compute the sampled credit

$$\widehat{A}_i = \text{stopgrad}\left[Q_{\bar{\phi}}(c, \mathbf{a}) - \frac{1}{M} \sum_{m=1}^{M} Q_{\bar{\phi}}(c, \mathbf{a}_{-i}, \tilde{a}_i^{(m)}) \right]. \tag{8}$$

Alternatives enter only the baseline; the actor update uses the factual path. Positive $\widehat{A}_i$ means that the critic values the factual action above the sampled baseline. Sampling requires no marginal action density.

The budget M controls the precision of this comparison. For a fixed critic, independent alternatives preserve the expected score update relative to its counterfactual expectation, while adding a covariance

term proportional to $1/M$ (Theorem 2, Section 4.1). Section 5.4 tests this prediction with frozen-policy diagnostics and examines how M affects learning returns and computational cost.

One mean $\mu_{\mathcal{B}}$ and standard deviation $\sigma_{\mathcal{B}}$ over batch–time–agent entries produce the stored credit

$$\widetilde{A}_i = \frac{\widehat{A}_i - \mu_{\mathcal{B}}}{\max(\sigma_{\mathcal{B}}, \epsilon_A)}. \tag{9}$$

Batch centering can change the sign of $\widehat{A}_i$. The mean and population standard deviation are computed jointly over all entries and agents in the collection batch. The standardized credit is stored before optimization and remains fixed throughout the collection iteration's policy epochs, without minibatch recomputation or restandardization.

Algorithm 1 CGPO

Require: Actor parameters θ, critic parameters ϕ; M alternatives, E epochs, target rate τ.
Initialize target critic $\bar{\phi} \leftarrow \phi$.
while training **do**
  Snapshot $\theta_{\text{old}} \leftarrow \theta$; collect batch $\mathcal{B}$ with factual paths and old log densities.
  Compute target $\mathbf{y}$ using $Q_{\bar{\phi}}$ and behavior-policy next actions via equation 45.
  for each batch entry $(c, \mathbf{o}, \mathbf{a})$ and agent i **do**
    $\widetilde{a}_i^{(1:M)} \overset{\text{iid}}{\sim} \pi_{\text{old},i}(\cdot \mid o_i)$
    $\widehat{A}_i \leftarrow Q_{\bar{\phi}}(c, \mathbf{a}) - \frac{1}{M}\sum_{m=1}^{M} Q_{\bar{\phi}}(c, \mathbf{a}_{-i}, \widetilde{a}_i^{(m)})$ ▷ Fix teammates
  end for
  Standardize and freeze $\widetilde{\mathbf{A}}$ via equation 9.
  for E epochs, minibatch $\mathcal{D} \subset \mathcal{B}$ **do**
    Minimize $\mathcal{L}_Q(\phi; \mathcal{D}, \mathbf{y})$ over ϕ.
    Minimize $\mathcal{L}_{\text{actor}}(\theta; \mathcal{D}, \widetilde{\mathbf{A}})$ via equation 11. ▷ After critic warm-up
    $\bar{\phi} \leftarrow (1-\tau)\bar{\phi} + \tau\phi$.
  end for
end while

3.3 Policy Optimization and Training

Figure 1B connects the action-level credit to the path that produced it. Proposition 1 places the same action-level credit in every transition-score term. We implement this structure with stored credit and a transition-level clipped objective.

For a recorded path, define

$$\omega_{i,k}(\theta) = \frac{p_{\theta_i,k}(x_{i,k+1} \mid x_{i,k}, o_i)}{p_{\text{old},i,k}(x_{i,k+1} \mid x_{i,k}, o_i)}. \tag{10}$$

We adopt the likelihood-ratio clipping mechanism of PPO (Schulman et al., 2017), applied to individual generation transitions as in DPPO (Ren et al., 2025). With the stored counterfactual credit as the update weight, the actor objective is

$$\mathcal{L}_{\text{actor}}(\theta) = -\mathbb{E}_{\mathcal{B}}\left[\frac{1}{NK}\sum_{i,k}\min\Big(\omega_{i,k}\widetilde{A}_i,\ \text{clip}(\omega_{i,k}, 1-\epsilon, 1+\epsilon)\widetilde{A}_i\Big)\right], \tag{11}$$

where $\mathbb{E}_{\mathcal{B}}$ denotes the empirical average over collected factual entries.

Algorithm 1 combines credit construction with actor–critic learning. Before optimization, the target critic supplies TD(λ) regression targets $\mathbf{y}$; the critic minimizes $\mathcal{L}_Q$, the regression loss between each head's prediction and its corresponding target in $\mathbf{y}$, averaged over batch entries and heads. The actor reuses the factual paths and fixed credit across epochs, while the target critic tracks the online critic after each minibatch. Updated target parameters supply the next collection's credit and regression targets. Appendix B.1 specifies the diffusion transitions and action map, and Appendix B.2 gives the target construction and optimization details.

4 STATISTICAL PROPERTIES OF COUNTERFACTUAL CREDIT

Proposition 1 establishes the gradient structure underlying CGPO. We now analyze the covariance introduced by finite comparisons, the conditions for a benefit over a global advantage. We retain the path regularity and independent-sampling assumptions in Appendix A and analyze unstandardized updates at a fixed reference policy π. Within a collection, this is the behavior policy generating factual and alternative paths.

4.1 COST OF FINITE COUNTERFACTUAL SAMPLING

For a fixed agent i, write $S = S_i$ for the path score in equation 4 and $C = (H, \mathbf{a}_{-i})$ for the conditioning context. With the critic held fixed, let $q = Q_{\bar{\phi}}(c, \mathbf{a})$ and $\widetilde{q}^{(m)} = Q_{\bar{\phi}}(c, \mathbf{a}_{-i}, \widetilde{a}_i^{(m)})$ denote its evaluations of the executed joint action and an independently sampled alternative, respectively. To quantify the additional gradient variance introduced by counterfactual sampling, we compare the sampled update with its counterpart using the exact counterfactual expectation:

$$g_M = S\left(q - \frac{1}{M}\sum_{m=1}^{M} \widetilde{q}^{(m)}\right), \qquad g_\infty = S\big(q - \mathbb{E}[q \mid C]\big).$$

Theorem 2 (Finite-M sampling cost)**.** *For a fixed critic and independent counterfactual samples with finite conditional moments,* $\mathbb{E}[g_M \mid C] = \mathbb{E}[g_\infty \mid C]$ *and*

$$\operatorname{tr}\operatorname{Cov}(g_M \mid C) - \operatorname{tr}\operatorname{Cov}(g_\infty \mid C) = \frac{\operatorname{Var}(\widetilde{q} \mid C)}{M}\,\mathbb{E}[\|S\|^2 \mid C]. \tag{12}$$

Here $\widetilde{q}$ *denotes one counterfactual value draw.*

The sampling cost is the variance of alternative values, weighted by the path-score magnitude and divided by M. Doubling the budget halves this additional covariance, while its scale varies with the agent, context, and training stage. The full covariance-matrix identity appears in Appendix A.2. We next identify when removing teammate variation outweighs this sampling cost.

4.2 WHEN COUNTERFACTUAL CREDIT REDUCES VARIANCE

To identify when counterfactual credit reduces gradient variance, we introduce an additional simplifying assumption: the team value decomposes additively as $Q^\pi(H, \mathbf{a}) = u(H, a_i) + v(H, \mathbf{a}_{-i})$, where u captures agent i's action contribution and v captures the teammates' contribution. For this comparison, we use the exact team value, setting $q = Q^\pi$ in g_M, and define the global-advantage update as $g_{\text{global}} = S(Q^\pi - V^\pi)$.

Corollary 3 (Benefit versus finite-sample cost)**.** *Fix H and assume the additive value model, conditionally factorized policies, and independent counterfactual samples. If* $\operatorname{Var}(v \mid H) > 0$ *and* $\mathbb{E}[\|S\|^2 \mid H] > 0$*, then*

$$\operatorname{tr}\operatorname{Cov}(g_M \mid H) < \operatorname{tr}\operatorname{Cov}(g_{\text{global}} \mid H) \quad \Longleftrightarrow \quad M > \frac{\operatorname{Var}(u \mid H)}{\operatorname{Var}(v \mid H)}. \tag{13}$$

Holding teammates fixed cancels v in every factual–counterfactual difference. What remains is the cost of estimating the baseline for agent i, $\operatorname{Var}(u \mid H)/M$. The threshold therefore has a direct interpretation: counterfactual credit improves update precision when this sampling cost is smaller than the teammate variation it removes. Greater teammate variation lowers the required budget, whereas greater variation due to agent i's action raises it. Appendix A.3 gives the covariance decomposition.

5 EXPERIMENTS

Our experiments address three questions: whether CGPO performs consistently across cooperative-control domains, whether its gains arise from counterfactual credit when the diffusion-policy parameterization is held fixed, and how finite sampling and generation depth affect training.

5.1 Experimental Setup

Tasks and baselines. We evaluate CGPO on three MPE configurations (Lowe et al., 2017), seven MaMuJoCo partitions (Peng et al., 2021), and the VMAS tasks `balance`, `sampling`, and `multi_give_way` (Bettini et al., 2024a). On MPE and MaMuJoCo, we compare with MAPPO (Yu et al., 2022), HATD3 (Zhong et al., 2024), HASAC (Liu et al., 2024), the multi-agent diffusion baselines MADPMD and MASDAC (Ma et al., 2025; Li et al., 2026c), and OMAD (Li et al., 2026c). VMAS comparisons use the BenchMARL implementations of MAPPO, MASAC (Bettini et al., 2024b), and MADDPG (Lowe et al., 2017), together with counterfactual-baseline and global-advantage ablations.

Evaluation protocol and ablations. We report mean returns and standard deviations over five seeds for MPE/MaMuJoCo and four seeds for all VMAS experiments. Appendix B gives implementation and evaluation protocols, including training budgets in Appendix B.3 and hyperparameters in Table 3. Appendices C.2 and C.3 provide additional credit ablations and sensitivity studies for the sample budget M and diffusion generation depth K.

Table 1: MPE and MaMuJoCo final returns (mean $\pm$ SD). Bold indicates the highest mean. See Appendix B.3 for the evaluation protocol.

Task	HATD3	HASAC	MADPMD	MASDAC	OMAD	MAPPO	CGPO (ours)
MPE							
Coop. Navigation ($N = 3$)	$-26.2_{\pm 3.5}$	$-25.1_{\pm 1.3}$	$-29.4_{\pm 1.6}$	$-41.0_{\pm 1.1}$	$-23.6_{\pm 1.12}$	$-27.2_{\pm 1.1}$	$\mathbf{-21.3_{\pm 1.35}}$
Coop. Navigation ($N = 4$)	$-63.0_{\pm 2.2}$	$-65.6_{\pm 3.5}$	$-85.0_{\pm 3.8}$	$-84.6_{\pm 3.1}$	$-56.9_{\pm 0.81}$	$-60.2_{\pm 13.8}$	$\mathbf{-38.8_{\pm 0.7}}$
Physical Deception ($N = 2$)	$43.0_{\pm 3.6}$	$45.4_{\pm 3.1}$	$36.9_{\pm 4.5}$	$33.1_{\pm 5.1}$	$45.6_{\pm 4.18}$	$35.0_{\pm 12.6}$	$\mathbf{61.9_{\pm 13.4}}$
MaMuJoCo							
Ant 2×4	$6151.9_{\pm 408.9}$	$6980.4_{\pm 458.5}$	$7042.7_{\pm 379.8}$	$7266.2_{\pm 431.5}$	$7637.3_{\pm 282.8}$	$6433.6_{\pm 496.7}$	$\mathbf{10849.2_{\pm 132.0}}$
Ant 2×4d	$4992.1_{\pm 595.2}$	$4941.9_{\pm 320.0}$	$6672.5_{\pm 467.5}$	$6936.2_{\pm 309.7}$	$7546.1_{\pm 172.0}$	$5946.4_{\pm 660.1}$	$\mathbf{11059.6_{\pm 503.0}}$
Ant 4×2	$5911.2_{\pm 254.0}$	$6597.1_{\pm 418.7}$	$7139.9_{\pm 462.1}$	$6198.5_{\pm 1012.1}$	$7544.4_{\pm 42.2}$	$8731.2_{\pm 401.2}$	$\mathbf{9319.3_{\pm 341.76}}$
HalfCheetah 2×3	$8732.6_{\pm 413.6}$	$8835.7_{\pm 356.3}$	$10305.5_{\pm 735.9}$	$10540.7_{\pm 963.8}$	$14526.6_{\pm 1187.0}$	$4006.4_{\pm 1310.6}$	$\mathbf{16746.8_{\pm 1034.2}}$
HalfCheetah 6×1	$8453.7_{\pm 503.4}$	$8714.3_{\pm 424.6}$	$10402.8_{\pm 622.4}$	$10049.9_{\pm 751.4}$	$11232.1_{\pm 253.1}$	$4653.6_{\pm 721.2}$	$\mathbf{14242.6_{\pm 623.4}}$
Walker2d 2×3	$6248.6_{\pm 622.3}$	$6753.5_{\pm 436.2}$	$6112.8_{\pm 269.3}$	$8154.4_{\pm 322.5}$	$8303.1_{\pm 819.6}$	$5541.6_{\pm 4126.6}$	$\mathbf{9447.1_{\pm 869.2}}$
Swimmer 2×1	$98.5_{\pm 52.3}$	$134.2_{\pm 23.3}$	$136.8_{\pm 7.6}$	$149.3_{\pm 12.6}$	$165.2_{\pm 22.8}$	$133.6_{\pm 52.6}$	$\mathbf{188.9_{\pm 2.6}}$

5.2 Performance Across Cooperative Control Tasks

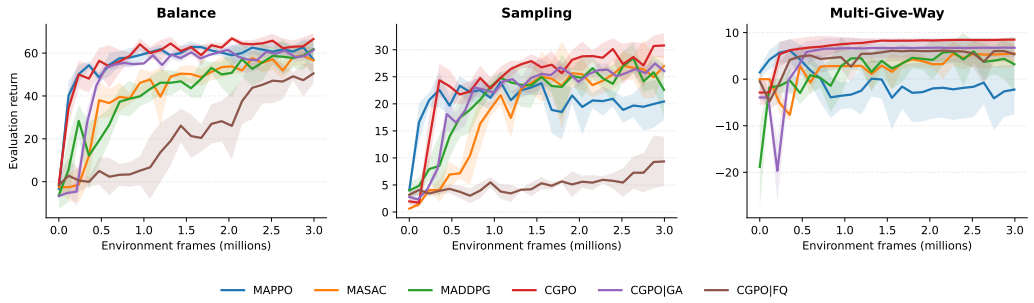
We first evaluate whether CGPO's performance persists across coordination structures in MPE, MaMuJoCo, and VMAS. Across all thirteen task configurations, it achieves the highest mean return among the compared methods (Tables 1 and 2). These comparisons span landmark coverage and coordinated deception, coupled locomotion with different actuator partitions, and team balancing, sampling, and collision avoidance. On MaMuJoCo, gains over the strongest baseline in each partition range from 6.7% to 46.6%; on VMAS, the comparisons use a common 3M-frame training budget and evaluation schedule. Learning curves appear in Figure 2 for VMAS and Appendix Figure 4 for MPE and MaMuJoCo. Figure 2 also reports normalized aggregate scores across the three VMAS tasks, and Appendix Figure 3 visualizes their final returns.

Table 2: VMAS final scheduled-evaluation returns after 3M frames (mean $\pm$ SD over four seeds). Bold indicates the highest mean in each task.

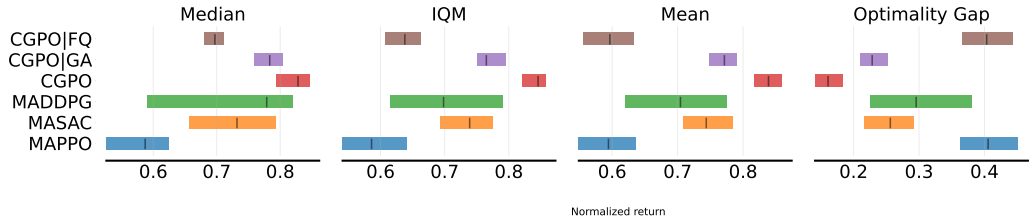
Task	MAPPO	MASAC	MADDPG	CGPO (ours)	Ablations	
					Factual-Q	Global advantage
`balance`	56.96 ± 9.42	56.56 ± 5.66	61.87 ± 3.64	$\mathbf{66.57 \pm 2.74}$	50.53 ± 4.78	61.49 ± 2.27
`sampling`	20.43 ± 3.42	26.99 ± 1.94	22.59 ± 4.83	$\mathbf{30.79 \pm 2.29}$	9.35 ± 4.58	26.06 ± 0.89
`multi_give_way`	-2.22 ± 5.37	5.53 ± 2.92	3.20 ± 6.00	$\mathbf{8.50 \pm 0.07}$	5.36 ± 2.07	6.74 ± 2.03

5.3 Counterfactual Credit Improves Diffusion-Policy Learning

Overall performance alone does not identify whether the gains come from the diffusion representation or from the learning signal. The matched *Factual-Q* ablation replaces counterfactual credit with



(a) Per-task learning curves.



(b) Cross-task aggregate performance.

Figure 2: Four-seed VMAS performance. (a) Learning curves: mean ± one sample SD. (b) Normalized aggregate scores with 95% stratified bootstrap intervals; lower is better only for the optimality gap. See Appendix B.3 for the aggregation protocol.

$Q_{\bar{\phi}}(c, \mathbf{a})$ while preserving architectures, TD targets, target updates, hyperparameters, and weight processing. The *Global advantage* ablation instead subtracts a baseline estimated by averaging the same target-critic head over independently sampled behavior-policy joint actions, marginalizing all agents' actions rather than fixing teammates' factual actions (Appendix B.4; Table 4).

Against Factual-Q, CGPO improves mean returns by 16.0, 21.4, and 3.1 points on `balance`, `sampling`, and `multi_give_way`, respectively (Table 2). The gains are largest on the first two tasks, where Factual-Q falls below MAPPO while CGPO exceeds it. This advantage is not limited to the Factual-Q control: although Global advantage improves over Factual-Q on all three VMAS tasks, CGPO remains higher at 66.57 versus 61.49 on `balance`, 30.79 versus 26.06 on `sampling`, and 8.50 versus 6.74 on `multi_give_way`. The normalized aggregate scores show the same ordering (Figure 2(b)).

With the diffusion representation unchanged, CGPO has higher means than both ablations on all seven MaMuJoCo partitions and all three MPE configurations (Appendix Table 4), indicating that the benefit of policy-sampled comparison is not confined to one benchmark suite.

From action quality to team coordination. The return gains are accompanied by changes in both individual action quality and team behavior. Holding teammates' first actions and the continuation policy fixed, environment-branch evaluations give CGPO actions higher mean returns than Factual-Q actions (Appendix Figure 8). These evaluations measure action quality through environment returns rather than the training critic. Repeated stochastic executions reveal two forms of coordination: on `sampling`, CGPO combines more reliable reward-density coverage with varied successful region-to-agent assignments (Appendix Figure 9); on `multi_give_way`, it consistently delivers two agents to their goals with fewer collisions than Factual-Q and stable passing orders within each training seed (Appendix Figure 10). Together, these results connect the credit ablation to better actions and task-specific patterns of cooperation. Appendix C.4 provides the protocols and cross-seed statistics.

5.4 Sampling and Generation Tradeoffs

Having established the learning benefits of counterfactual credit, we next examine how the sampling budget and generation depth affect its estimation, learning returns, and computational cost.

Finite-sampling noise. Theorem 2 predicts a $1/M$ covariance contribution from finite counterfactual sampling. To isolate this contribution, we freeze the actor, critic, and factual paths and resample only the alternatives. Across three VMAS tasks and multiple checkpoints, measured covariance traces broadly follow the prediction, whose scale is determined by an independent sample pool without fitting the measured covariance (Appendix Figure 7; protocol in Appendix B.4).

Sample budget and cost. This predictable reduction in sampling noise does not imply monotonically higher learning returns: with generation depth fixed within each task, the sweep over $M \in \{4, 8, 16\}$ gives the highest mean at $M = 8$ for `balance` and $M = 16$ for `sampling` (Appendix Figure 5). Each additional comparison requires an alternative action and a critic evaluation, making sampling work linear in M at fixed generation depth. End-to-end A100 measurements on MPE Spread and MaMuJoCo Swimmer nevertheless show similar training times for CGPO, Global advantage, and Factual-Q, all below OMAD's (Appendix C.5).

Generation depth. Whereas M controls the precision of the sampled baseline, generation depth K changes the action-generation process itself. In the sweep over $K \in \{2, 5, 10\}$, `balance` improves sharply from two to five steps with no further mean-return gain at ten, whereas `sampling` favors $K = 2$ (Appendix Figure 6). Together, the budget and depth sweeps favor task-specific choices: more comparisons improve baseline precision, while neither more comparisons nor deeper generation consistently improves return.

6 Related Work

Generative multi-agent learning. OMAD trains decentralized diffusion actors with joint distributional value guidance, while DDPL constructs local values by averaging teammate actions (Li et al., 2026c; Zhang et al., 2026b). We address coordination through agent-specific credit: fixing teammates' realized actions, CGPO evaluates independent alternative actions for each agent and uses the resulting credit to update that agent's recorded generation path. Our experiments assess learning performance and the contribution of counterfactual credit.

Credit assignment for cooperative agents. For each agent, COMA marginalizes that agent's action while fixing teammates' actions, including a Monte Carlo extension to continuous control (Foerster et al., 2018). Factorized action-dependent baselines and multi-agent optimal baselines establish related variance-reduction principles (Wu et al., 2018; Kuba et al., 2021). Building on these foundations, we connect agent-specific action credit to the recorded stochastic generation path and evaluate its contribution to diffusion-policy learning. Our method and performance evaluation are complemented by a finite-budget analysis and quantitative tests of intervention-sampling covariance.

Diffusion and flow policy optimization. DPPO optimizes denoising-transition likelihoods; NCDPO backpropagates through noise-conditioned generation (Ren et al., 2025; Yang et al., 2025). FPO and PolicyFlow construct flow-based surrogate ratios, while GenPO and GenPO++ use reversible augmented representations (McAllister et al., 2026; Yang et al., 2026; Ding et al., 2025; Hu et al., 2026). CGPO uses explicit stochastic-transition ratios: one stored agent credit weights every transition of its factual path, connecting team-level evaluation to decentralized generative-policy updates. **Appendix D** discusses these optimization spaces, stochastic flow policies, and additional credit-assignment and generative MARL methods.

7 Conclusion

We introduced CGPO, an on-policy framework that trains decentralized diffusion policies using counterfactual credit conditioned on teammates' executed actions. Our analysis quantifies the additional gradient covariance from finite counterfactual sampling and, under an additive team-value model, identifies when removing teammate variation outweighs this sampling cost. Across MPE, MaMuJoCo, and VMAS, CGPO achieves the highest mean returns among the compared methods. With the diffusion architecture held fixed, it also outperforms Factual-Q and Global advantage in mean return across all thirteen task configurations, supporting the value of teammate-conditioned credit. Frozen-policy diagnostics support the predicted $1/M$ sampling law, while behavior evaluations show improved conditional action quality and coordination relative to Factual-Q. These results establish teammate-conditioned credit as an effective training signal for decentralized diffusion policies.

AI USE STATEMENT

We used AI tools to assist with language polishing and literature discovery during the preparation of this manuscript. For language polishing, these tools helped refine wording and improve the clarity and readability of the text. We also used AI-assisted search to help identify potentially relevant references for the literature review. The authors reviewed the AI-assisted edits before incorporating them into the manuscript. We take full responsibility for the final content of the paper, including its claims, results, and citations.

REPRODUCIBILITY STATEMENT

Section 3 and Algorithm 1 describe the proposed method and training procedure. Appendix A provides the assumptions and proofs of the theoretical results. Appendix B details the policy parameterization, critic targets, and optimization procedures. In particular, Appendix B.3 specifies the environments, training budgets, random seeds, evaluation protocols, and execution environment, with hyperparameters summarized in Table 3. Appendix B.4 describes the ablation and diagnostic implementations, and Appendix C.4 documents the action-quality and coordination evaluations. We will release the source code for CGPO upon acceptance.

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*, volume 34, 2021.

Haojie Bai, Aimin Li, Ruoyu Yao, Xiongwei Zhao, Tingting Zhang, Xing Zhang, Lin Gao, and Jun Ma. SCORP: Scene-consistent multi-agent diffusion planning with stable online reinforcement post-training for cooperative driving. *arXiv preprint arXiv:2604.11734*, 2026.

Matteo Bettini, Ryan Kortvelesy, Jan Blumenkamp, and Amanda Prorok. VMAS: A vectorized multi-agent simulator for collective robot learning. In *Distributed Autonomous Robotic Systems*, volume 28 of *Springer Proceedings in Advanced Robotics*, pp. 42–56. Springer, 2024a. doi: 10.1007/978-3-031-51497-5_4.

Matteo Bettini, Amanda Prorok, and Vincent Moens. BenchMARL: Benchmarking multi-agent reinforcement learning. *Journal of Machine Learning Research*, 25(217):1–10, 2024b.

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Robotics: Science and Systems*, 2023. doi: 10.15607/RSS.2023.XIX.026.

Shutong Ding, Ke Hu, Shan Zhong, Haoyang Luo, Weinan Zhang, Jingya Wang, Jun Wang, and Ye Shi. GenPO: Generative diffusion models meet on-policy reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Jakob Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. Counterfactual multi-agent policy gradients. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

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*, volume 35, pp. 5510–5521, 2022.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851, 2020.

Ke Hu, Shutong Ding, Panxin Tao, Jingya Wang, and Ye Shi. GenPO++: Generative policy optimization with jacobian-free likelihood ratios. *arXiv preprint arXiv:2606.06967*, 2026.

Shuangyao Huang. A self-evolving default action for cooperative tasks with continuous action space. *arXiv preprint arXiv:2607.18597*, 2026.

Jakub Grudzien Kuba, Muning Wen, Linghui Meng, Shangding Gu, Haifeng Zhang, David Henry Mguni, Jun Wang, and Yaodong Yang. Settling the variance of multi-agent policy gradients. In *Advances in Neural Information Processing Systems*, volume 34, pp. 13458–13470, 2021.

Dongsu Lee, Daehee Lee, and Amy Zhang. Multi-agent coordination via flow matching. In *International Conference on Learning Representations*, 2026.

Jeric Lew, Yuhong Cao, Derek Ming Siang Tan, and Guillaume Sartoretti. AID: Agent intent from diffusion for multi-agent informative path planning. In *Swarm Intelligence*, volume 16515 of *Lecture Notes in Computer Science*, pp. 41–54. Springer, 2026. doi: 10.1007/978-3-032-26123-6_4.

Chao Li, Ziwei Deng, Chenxing Lin, Wenqi Chen, Yongquan Fu, Weiquan Liu, Chenglu Wen, Cheng Wang, and Siqi Shen. DoF: A diffusion factorization framework for offline multi-agent reinforcement learning. In *International Conference on Learning Representations*, 2025.

Yueheng Li, Guangming Xie, and Zongqing Lu. Difference advantage estimation for multi-agent policy gradients. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 13066–13085. PMLR, 2022.

Zhuoran Li, Ling Pan, Jiatai Huang, and Longbo Huang. Improving generalization and data efficiency with diffusion in offline multi-agent RL. *Transactions on Machine Learning Research*, 2026a.

Zhuoran Li, Xun Wang, Hai Zhong, Qingxin Xia, Lihua Zhang, and Longbo Huang. OM2P: Offline multi-agent mean-flow policy. In *Proceedings of the 25th International Conference on Autonomous Agents and Multiagent Systems*, pp. 3767–3776, 2026b.

Zhuoran Li, Hai Zhong, Xun Wang, Qingxin Xia, Lihua Zhang, and Longbo Huang. Diffusing to coordinate: Efficient online multi-agent diffusion policies. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026c.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023.

Jiarong Liu, Yifan Zhong, Siyi Hu, Haobo Fu, Qiang Fu, Xiaojun Chang, and Yaodong Yang. Maximum entropy heterogeneous-agent reinforcement learning. In *The Twelfth International Conference on Learning Representations*, 2024.

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*, volume 30, 2017.

Haitong Ma, Tianyi Chen, Kai Wang, Na Li, and Bo Dai. Efficient online reinforcement learning for diffusion policy. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 41837–41853. PMLR, 2025.

David McAllister, Songwei Ge, Brent Yi, Chung Min Kim, Ethan Weber, Hongsuk Choi, Haiwen Feng, and Angjoo Kanazawa. Flow matching policy gradients. In *International Conference on Learning Representations*, 2026.

Bei Peng, Tabish Rashid, Christian Schroeder de Witt, Pierre-Alexandre Kamienny, Philip Torr, Wendelin Böhmer, and Shimon Whiteson. FACMAC: Factored multi-agent centralised policy gradients. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

Michael Psenka, Alejandro Escontrela, Pieter Abbeel, and Yi Ma. Learning a diffusion model policy from rewards via Q-score matching. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 41163–41182. PMLR, 2024.

Dan Qiao, Wenhao Li, Shanchao Yang, Hongyuan Zha, and Baoxiang Wang. Offline multi-agent reinforcement learning via sequential score decomposition. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026.

Tabish Rashid, Mikayel Samvelyan, Christian Schroeder de Witt, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. QMIX: Monotonic value function factorisation for deep multi-agent reinforcement learning. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 4295–4304. PMLR, 2018.

Allen Z. Ren, Justin Lidard, Lars L. Ankile, Anthony Simeonov, Pulkit Agrawal, Anirudha Majumdar, Benjamin Burchfiel, Hongkai Dai, and Max Simchowitz. Diffusion policy policy optimization. In *International Conference on Learning Representations*, 2025.

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

Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, Vinicius Zambaldi, Max Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z. Leibo, Karl Tuyls, and Thore Graepel. Value-decomposition networks for cooperative multi-agent learning based on team reward. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pp. 2085–2087, 2018.

Richard S. Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems*, volume 12, 1999.

George Tucker, Surya Bhupatiraju, Shixiang Gu, Richard Turner, Zoubin Ghahramani, and Sergey Levine. The mirage of action-dependent baselines in reinforcement learning. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 5015–5024, 2018.

Yihan Wang, Beining Han, Tonghan Wang, Heng Dong, and Chongjie Zhang. DOP: Off-policy multi-agent decomposed policy gradients. In *International Conference on Learning Representations*, 2021.

Cathy Wu, Aravind Rajeswaran, Yan Duan, Vikash Kumar, Alexandre M. Bayen, Sham Kakade, Igor Mordatch, and Pieter Abbeel. Variance reduction for policy gradient with action-dependent factorized baselines. In *International Conference on Learning Representations*, 2018.

Ningyuan Yang, Jiaxuan Gao, Feng Gao, Yi Wu, and Chao Yu. Fine-tuning diffusion policies with backpropagation through diffusion timesteps. *arXiv preprint arXiv:2505.10482*, 2025.

Shunpeng Yang, Ben Liu, and Hua Chen. PolicyFlow: Policy optimization with continuous normalizing flow in reinforcement learning. In *International Conference on Learning Representations*, 2026.

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*, volume 35, pp. 24611–24624, 2022.

Lei Yuan, Yuqi Bian, Lihe Li, Ziqian Zhang, Cong Guan, and Yang Yu. Efficient multi-agent offline coordination via diffusion-based trajectory stitching. In *International Conference on Learning Representations*, 2025.

Xianghua Zeng, Hang Su, Zhengyi Wang, and Zhiyuan Lin. Graph diffusion for robust multi-agent coordination. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 74157–74174. PMLR, 2025.

Yixian Zhang, Shu'ang Yu, Tonghe Zhang, Mo Guang, Haojia Hui, Kaiwen Long, Yu Wang, Chao Yu, and Wenbo Ding. SAC Flow: Sample-efficient reinforcement learning of flow-based policies via velocity-reparameterized sequential modeling. In *International Conference on Learning Representations*, 2026a.

Yuyang Zhang, Haldun Balim, and Na Li. Decentralized diffusion policy learning for enhanced exploration in cooperative multi-agent reinforcement learning. *arXiv preprint arXiv:2605.07101*, 2026b.

Changyuan Zhao, Ruichen Zhang, Jiacheng Wang, Dusit Niyato, Geng Sun, Hongyang Du, Zan Li, Abbas Jamalipour, and Dong In Kim. Temporal spectrum cartography in low-altitude economy networks: A generative AI framework with multi-agent learning. *IEEE Transactions on Mobile Computing*, 25(6):8016–8033, 2026. doi: 10.1109/TMC.2025.3647029.

Xutong Zhao and Yaqi Xie. Multi-level advantage credit assignment for cooperative multi-agent reinforcement learning. In *Proceedings of the 28th International Conference on Artificial Intelligence and Statistics*, volume 258 of *Proceedings of Machine Learning Research*, pp. 2971–2979. PMLR, 2025.

Yifan Zhong, Jakub Grudzien Kuba, Xidong Feng, Siyi Hu, Jiaming Ji, and Yaodong Yang. Heterogeneous-agent reinforcement learning. *Journal of Machine Learning Research*, 25(32): 1–67, 2024.

Zhengbang Zhu, Minghuan Liu, Liyuan Mao, Bingyi Kang, Minkai Xu, Yong Yu, Stefano Ermon, and Weinan Zhang. MADiff: Offline multi-agent learning with diffusion models. In *Advances in Neural Information Processing Systems*, volume 37, pp. 4177–4206, 2024.

A Proofs of Main Results

This appendix gives the assumptions and proofs underlying the path-gradient construction in Section 3.1 and the estimator analysis in Section 4. The main results appear in main-text order, followed by a supplementary critic-error bound and the frozen-batch extension used by the covariance diagnostic.

Assumption 1 (Regular latent policy)**.** *For every agent, the path density $p_{\theta_i}(\xi_i \mid o_i)$ has parameter-independent support and is differentiable almost everywhere in the parameters. Differentiation can be exchanged with the path and discounted-return integrals. The score, value, and score-weighted value terms used below have the finite moments required for their expectations and covariances; for learned-critic results, this applies to the fixed critic output q as well as to Q^π.*

Given the augmented information state H, the joint path law factorizes as

$$p_\theta(\boldsymbol{\xi} \mid H) = \prod_i p_{\theta_i}(\xi_i \mid o_i).$$

Parameters may be shared, while generation noise is independent across agents conditional on H.

Assumption 2 (Valid counterfactual sampling)**.** *Condition on the reference actor and critic parameters, held fixed during credit construction. Given $C_i = (H, \mathbf{a}_{-i})$, the M counterfactual paths are i.i.d. from the same reference policy $p_{\theta_i}(\cdot \mid o_i)$ as agent i's factual path and are independent of that path. The resulting baseline is treated with stop-gradient in the actor update.*

A.1 Proof of Proposition 1

We apply the counterfactual-baseline argument of Foerster et al. (2018) to the diffusion path distribution. The joint generation path may be treated as an augmented joint action whose environmental consequence is the executed joint action $\mathbf{a}$. For separately parameterized actors, the policy-gradient theorem (Sutton et al., 1999) therefore gives

$$\nabla_{\theta_i} J = \frac{1}{1-\gamma} \mathbb{E}_{d^\pi, p_\theta} \left[S_i(\xi_i, o_i) Q^\pi(H, \mathbf{a}) \right], \tag{14}$$

where

$$S_i(\xi_i, o_i) = \nabla_{\theta_i} \log p_{\theta_i}(\xi_i \mid o_i).$$

Because the initial density ρ is parameter independent, the path score decomposes as

$$S_i = \sum_{k=0}^{K-1} \nabla_{\theta_i} \log p_{\theta_i,k}(x_{i,k+1} \mid x_{i,k}, o_i),$$

as in equation 4.

Let $C_i = (H, \mathbf{a}_{-i})$. Conditional policy factorization and Assumption 1 imply the conditional score identity

$$\mathbb{E}[S_i \mid C_i] = \int p_{\theta_i}(\xi_i \mid o_i) \nabla_{\theta_i} \log p_{\theta_i}(\xi_i \mid o_i)\, d\xi_i \tag{15}$$

$$= \nabla_{\theta_i} \int p_{\theta_i}(\xi_i \mid o_i)\, d\xi_i = 0. \tag{16}$$

The exact counterfactual baseline $b_i^\pi(C_i)$ depends on C_i but not on agent i's factual path once C_i is fixed. Hence

$$\mathbb{E}[S_i b_i^\pi(C_i) \mid C_i] = b_i^\pi(C_i)\mathbb{E}[S_i \mid C_i] = 0. \tag{17}$$

Subtracting this term from equation 14 gives

$$\nabla_{\theta_i} J = \frac{1}{1-\gamma} \mathbb{E}_{d^\pi, p_\theta} \left[S_i A_i^{\mathrm{cf},\pi} \right].$$

Substituting the path-score decomposition above yields equation 7.

Now consider the finite-sample exact-Q baseline

$$\widehat{b}_i = \frac{1}{M} \sum_{m=1}^{M} Q^\pi\big(H, \mathbf{a}_{-i}, \widetilde{a}_i^{(m)}\big), \qquad \widetilde{a}_i^{(m)} \overset{\text{iid}}{\sim} \pi_i(\cdot \mid o_i).$$

By Assumption 2, $\widehat{b}_i$ is conditionally independent of the factual path given C_i. Therefore

$$\mathbb{E}[S_i \widehat{b}_i \mid C_i] = \mathbb{E}[S_i \mid C_i]\, \mathbb{E}[\widehat{b}_i \mid C_i] = 0. \tag{18}$$

Thus replacing the exact counterfactual baseline by any finite $M \geq 1$ independent Monte Carlo estimate leaves the expected unstandardized score update unchanged.

If actor parameters are shared, differentiation of the joint path law produces the sum of the corresponding agent path scores, and the same baseline identity applies to each agent contribution. □

A.2 Proof of Theorem 2

Fix an agent and retain the main-text notation S, C, g_M, and g_∞. Let

$$q = Q_{\bar{\phi}}(c, \mathbf{a}), \qquad b^q = \mathbb{E}[q \mid C], \qquad \sigma_q^2(C) = \mathrm{Var}(q \mid C),$$

and

$$F(C) = \mathbb{E}[SS^\top \mid C].$$

The same fixed critic is used for the factual and all counterfactual evaluations. With

$$g_M = S\left(q - \frac{1}{M}\sum_{m=1}^{M} \widetilde{q}^{(m)}\right), \qquad g_\infty = S(q - b^q),$$

we prove the stronger matrix identity

$$\mathrm{Cov}(g_M \mid C) = \mathrm{Cov}(g_\infty \mid C) + \frac{\sigma_q^2(C)}{M} F(C). \tag{19}$$

Define the Monte Carlo baseline error

$$\eta_M = \frac{1}{M}\sum_{m=1}^{M} \widetilde{q}^{(m)} - b^q.$$

Conditional on C, Assumption 2 gives

$$\mathbb{E}[\eta_M \mid C] = 0, \qquad \mathrm{Var}(\eta_M \mid C) = \frac{\sigma_q^2(C)}{M}, \tag{20}$$

and η_M is independent of the factual pair (S, q). Since

$$g_M = g_\infty - S\eta_M,$$

the cross covariance vanishes:

$$\mathbb{E}\big[(g_\infty - \mathbb{E}[g_\infty \mid C])(S\eta_M)^\top \mid C\big] = 0. \tag{21}$$

Moreover,

$$\mathrm{Cov}(S\eta_M \mid C) = \mathbb{E}[SS^\top \mid C]\,\mathbb{E}[\eta_M^2 \mid C] \tag{22}$$

$$= F(C)\frac{\sigma_q^2(C)}{M}. \tag{23}$$

equation 20 also gives

$$\mathbb{E}[g_M \mid C] = \mathbb{E}[g_\infty \mid C]. \tag{24}$$

Combining these identities proves equation 19. Taking the trace gives equation 12. □

A.3 Proof of Corollary 3

Fix H and use the additive value model from the main text,

$$Q^\pi(H, \mathbf{a}) = u(H, a_i) + v(H, \mathbf{a}_{-i}),$$

where u is the component that depends on agent i's action and v is the component that depends only on teammates' actions. Write

$$S = S_i, \qquad F(H) = \mathbb{E}[SS^\top \mid H],$$

and set $q = Q^\pi$ in g_M and g_∞. Let

$$g_{\mathrm{global}} = S(Q^\pi - V^\pi).$$

We establish the matrix identity

$$\mathrm{Cov}(g_{\mathrm{global}} \mid H) - \mathrm{Cov}(g_M \mid H) = \left(\mathrm{Var}(v \mid H) - \frac{\mathrm{Var}(u \mid H)}{M}\right) F(H). \tag{25}$$

Let

$$\bar{u} = \mathbb{E}[u \mid H], \qquad \bar{v} = \mathbb{E}[v \mid H].$$

The exact counterfactual baseline is

$$\mathbb{E}[Q^\pi \mid H, \mathbf{a}_{-i}] = \bar{u} + v,$$

so the exact counterfactual update is

$$g_\infty = S(u - \bar{u}).$$

By contrast,

$$g_{\mathrm{global}} = S\big[(u - \bar{u}) + (v - \bar{v})\big].$$

Conditional policy factorization implies that (S, u) is independent of v given H. Since $\mathbb{E}[v - \bar{v} \mid H] = 0$, the cross covariance between $S(u - \bar{u})$ and $S(v - \bar{v})$ is zero. Therefore

$$\mathrm{Cov}(g_{\mathrm{global}} \mid H) = \mathrm{Cov}(g_\infty \mid H) + \mathrm{Var}(v \mid H)F(H). \tag{26}$$

For the sampled counterfactual baseline, define

$$\eta_M = \frac{1}{M}\sum_{m=1}^{M} u(H, \widetilde{a}_i^{(m)}) - \bar{u}.$$

Given H, η_M is independent of agent i's factual path and satisfies

$$\mathbb{E}[\eta_M \mid H] = 0, \qquad \mathrm{Var}(\eta_M \mid H) = \frac{\mathrm{Var}(u \mid H)}{M}.$$

The teammate-only component v cancels exactly in every factual–counterfactual difference, giving

$$g_M = g_\infty - S\eta_M.$$

The same calculation as in Theorem 2 yields

$$\mathrm{Cov}(g_M \mid H) = \mathrm{Cov}(g_\infty \mid H) + \frac{\mathrm{Var}(u \mid H)}{M} F(H). \tag{27}$$

Subtracting equation 27 from equation 26 proves equation 25.

The two estimators have the same conditional mean. Hence, when $\mathrm{Var}(v \mid H) > 0$ and $\mathrm{tr}\, F(H) = \mathbb{E}[\|S\|^2 \mid H] > 0$, taking traces gives

$$\mathrm{tr}\,\mathrm{Cov}(g_M \mid H) < \mathrm{tr}\,\mathrm{Cov}(g_{\mathrm{global}} \mid H)$$

if and only if

$$M > \frac{\mathrm{Var}(u \mid H)}{\mathrm{Var}(v \mid H)},$$

which is equation 13. □

Scaling with teammate variation. The same decomposition gives a simple scaling example. Suppose

$$v(H, \mathbf{a}_{-i}) = \sum_{j \neq i} v_j(H, a_j),$$

where the teammate-only terms are conditionally independent given H and their variances are uniformly bounded above and away from zero as N grows. Then

$$\mathrm{Var}(v \mid H) = \Theta(N).$$

If additionally $\mathrm{tr}\, F(H) = \Theta(1)$ and

$$\mathbb{E}[\|S\|^2 (u - \bar{u})^2 \mid H] = O(1),$$

then the covariance trace contributed by the teammate-only term in the global-advantage estimator is $\Theta(N)$, whereas the exact counterfactual term remains $O(1)$. This is only a scaling example for the additive special case; the finite-M tradeoff remains the one characterized by Corollary 3.

Remark 4 (Score-weighted optimal baseline). Following the optimal-baseline construction of Wu et al. (2018); Kuba et al. (2021), consider a scalar baseline depending on C_i. The counterfactual mean

$$b_i^q = \mathbb{E}[q \mid C_i]$$

minimizes the conditional squared residual $\mathbb{E}[(q - b)^2 \mid C_i]$. The scalar baseline that instead minimizes the trace of the path-score gradient covariance is

$$b_i^{q,*} = \frac{\mathbb{E}[\|S_i\|^2 q \mid C_i]}{\mathbb{E}[\|S_i\|^2 \mid C_i]}.$$

When $\mathbb{E}[\|S_i\|^2 \mid C_i] > 0$, the two coincide if $\|S_i\|^2$ and q are conditionally uncorrelated. Thus the counterfactual conditional mean provides the action comparison for agent i used by CGPO, while $b_i^{q,*}$ is the trace-optimal scalar control variate for the path score.

A.4 Critic Approximation Error

Proposition 5 (Critic error under sup-norm and L_2 control)**.** *For a fixed agent i, with the critic-head index suppressed, let*

$$e_i(H, \mathbf{a}) = Q_{\bar{\phi}}(c, \mathbf{a}) - Q^\pi(H, \mathbf{a}).$$

Let Δg_i be the difference between the expected unstandardized counterfactual score updates obtained using this critic output and using Q^π, with any $M \geq 1$ independent alternatives. Let ζ^π be the discounted factual law of $(H, \boldsymbol{\xi})$ and ν^π its $(H, \mathbf{a})$ marginal. With finite second moments,

$$\Delta g_i = \mathbb{E}_{\zeta^\pi}[S_i e_i], \qquad \|\Delta g_i\|_2 \leq \sqrt{\mathbb{E}_{\zeta^\pi}\|S_i\|_2^2}\,\sqrt{\mathbb{E}_{\nu^\pi} e_i^2}. \tag{28}$$

If $\|e_i\|_\infty \leq \varepsilon_{Q,i}$, then

$$\|\Delta g_i\|_2 \leq \varepsilon_{Q,i} \mathbb{E}_{\zeta^\pi}\|S_i\|_2. \tag{29}$$

If the critic is trained under a distribution μ satisfying $d\nu^\pi/d\mu \leq C$, then

$$\sqrt{\mathbb{E}_{\nu^\pi} e_i^2} \leq \sqrt{C\, \mathbb{E}_\mu e_i^2}.$$

For shared actor parameters, interpret S_i as the path score with respect to those parameters; the total error is the sum of the per-agent contributions.

Proof. Let $\widehat{b}_{e,i}$ denote the sample mean of the critic errors at agent i's independently sampled alternative actions. The difference between the two expected raw updates is

$$\Delta g_i = \mathbb{E}[S_i(e_i - \widehat{b}_{e,i})]. \tag{30}$$

Conditional on $C_i = (H, \mathbf{a}_{-i})$, $\widehat{b}_{e,i}$ is independent of agent i's factual path. Using the conditional score identity equation 16,

$$\mathbb{E}[S_i \widehat{b}_{e,i}] = 0.$$

Therefore

$$\Delta g_i = \mathbb{E}_{\zeta^\pi}[S_i e_i].$$

Cauchy–Schwarz gives

$$\|\Delta g_i\|_2 \le \sqrt{\mathbb{E}_{\zeta^\pi}\|S_i\|_2^2}\sqrt{\mathbb{E}_{\nu^\pi} e_i^2},$$

while the uniform bound gives

$$\|\Delta g_i\|_2 \le \mathbb{E}[\|S_i\|_2 |e_i|] \le \varepsilon_{Q,i}\mathbb{E}\|S_i\|_2.$$

Finally, $d\nu^\pi/d\mu \le C$ implies

$$\mathbb{E}_{\nu^\pi} e_i^2 = \mathbb{E}_\mu\left[\frac{d\nu^\pi}{d\mu} e_i^2\right] \le C\mathbb{E}_\mu e_i^2.$$

The shared-parameter statement follows by summing agent contributions. □

This result separates critic approximation from counterfactual-sampling noise: increasing M reduces the sampling covariance in Theorem 2, but does not remove the expected update error $\mathbb{E}[S_i e_i]$.

A.5 Frozen-Batch Covariance

Corollary 6 (Frozen-batch covariance law)**.** *Fix L factual context–path entries. All scores are taken with respect to the full actor parameter vector θ; for separately parameterized actors, each score may equivalently be embedded in the full parameter space with zeros outside its own block.*

Write the realized path scores as s_j, the corresponding factual critic values as q_j, their exact stopped counterfactual baselines as b_j^q, and their conditional counterfactual-value variances as $\sigma_{q,j}^2$. For each entry, independently draw M counterfactual values, with the sets also independent across entries, and define

$$\widehat{g}_M = \frac{1}{L}\sum_{j=1}^{L} s_j\left(q_j - \frac{1}{M}\sum_{m=1}^{M}\widetilde{q}_j^{(m)}\right), \qquad g_\infty = \frac{1}{L}\sum_{j=1}^{L} s_j(q_j - b_j^q). \tag{31}$$

Conditional on the fixed factual entries,

$$\mathbb{E}[\widehat{g}_M] = g_\infty, \tag{32}$$

$$\mathrm{Cov}(\widehat{g}_M) = \frac{1}{ML^2}\sum_{j=1}^{L}\sigma_{q,j}^2 s_j s_j^\top, \tag{33}$$

$$\mathrm{tr}\,\mathrm{Cov}(\widehat{g}_M) = \frac{1}{ML^2}\sum_{j=1}^{L}\sigma_{q,j}^2\|s_j\|^2. \tag{34}$$

Proof. For entry j, define

$$\eta_{j,M} = \frac{1}{M}\sum_{m=1}^{M}\widetilde{q}_j^{(m)} - b_j^q.$$

Conditional on the fixed factual entries, the $\eta_{j,M}$ are mutually independent and satisfy

$$\mathbb{E}[\eta_{j,M}] = 0, \qquad \mathrm{Var}(\eta_{j,M}) = \frac{\sigma_{q,j}^2}{M}.$$

Moreover,

$$\widehat{g}_M - g_\infty = -\frac{1}{L}\sum_{j=1}^{L} s_j \eta_{j,M}.$$

The conditional mean is therefore zero, proving equation 32. Independence across entries eliminates the cross terms, so

$$\mathrm{Cov}(\widehat{g}_M) = \frac{1}{L^2}\sum_{j=1}^{L} s_j s_j^\top \mathrm{Var}(\eta_{j,M}) \tag{35}$$

$$= \frac{1}{ML^2}\sum_{j=1}^{L} \sigma_{q,j}^2 s_j s_j^\top, \tag{36}$$

which proves equation 33. Taking the trace gives equation 34. □

B Implementation and Evaluation Protocols

This appendix specifies the diffusion actor and critic updates used in Section 3.3, followed by the task, evaluation, and diagnostic protocols for the experiments.

B.1 Actor Transitions and Likelihoods

Strided diffusion actor. All agents within a policy group share actor parameters θ. Let a base schedule have T_{D} noise levels β_t and cumulative products $\bar{\alpha}_t = \prod_{s=0}^{t}(1-\beta_s)$. The actor uses descending selected noise indices $t_0 > \cdots > t_{K-1}$; the reported five-step configuration uses $(99, 74, 49, 24, 0)$. Define $A_k = \bar{\alpha}_{t_k}$, $B_k = \bar{\alpha}_{t_{k+1}}$ for $k < K-1$, and $B_{K-1} = 1$. For the current latent x and predicted noise $\epsilon_\theta(x, t_k, o_i)$,

$$\widehat{x}_{\mathrm{clean}} = \mathrm{clip}\left(\frac{x - \sqrt{1-A_k}\,\epsilon_\theta(x, t_k, o_i)}{\sqrt{A_k}}, -c_x, c_x\right), \tag{37}$$

$$\alpha_k = A_k/B_k, \qquad \beta_k^{\mathrm{skip}} = 1-\alpha_k, \tag{38}$$

$$\mu_{\theta,k}(x, o_i) = \frac{\sqrt{B_k}\,\beta_k^{\mathrm{skip}}}{1-A_k}\widehat{x}_{\mathrm{clean}} + \frac{\sqrt{\alpha_k}(1-B_k)}{1-A_k}x, \tag{39}$$

$$\sigma_k = \max\left\{\eta\sqrt{\beta_k^{\mathrm{skip}}},\, \sigma_{\min}\right\}. \tag{40}$$

The resulting transition is

$$p_{\theta,k}(x_{i,k+1} \mid x_{i,k}, o_i) = \mathcal{N}\big(\mu_{\theta,k}(x_{i,k}, o_i), \sigma_k^2 I\big). \tag{41}$$

All transitions, including the final one, have positive standard deviation. The mean uses the DDPM posterior-mean form with a clipped clean prediction; the chosen variance defines a Gaussian transition kernel rather than the DDPM posterior variance. The same mean and standard deviation are used for sampling and for recomputing old/new likelihood ratios.

Action map and reference configuration. The diffusion actor uses $x_{i,0} \sim \mathcal{N}(0, \sigma_{\mathrm{init}}^2 I)$ and scalar time $t_k/100$ concatenated with the observation and noisy action, without a separate time-embedding network, and the parameter-independent action map

$$h_i(x) = \frac{a_i^{\max} + a_i^{\min}}{2} + \frac{a_i^{\max} - a_i^{\min}}{2} \odot \tanh(x), \tag{42}$$

where $a_i^{\min}$ and $a_i^{\max}$ are the environment's componentwise action bounds and $\odot$ denotes elementwise multiplication. For latent dimension d_i, each stored transition log density is

$$\log p_{\theta,k}(x' \mid x, o_i) = -\frac{d_i}{2}\log(2\pi\sigma_k^2) - \frac{\|x' - \mu_{\theta,k}(x, o_i)\|^2}{2\sigma_k^2}. \tag{43}$$

No marginal action density or action-transform Jacobian is required for the latent-path ratios. Before exponentiation, the implementation clamps the log ratio to $[-20, 20]$ for numerical stability; this clamp is inactive near the behavior parameters and preserves the local first-order identity. The actors use $T_{\mathrm{D}} = 100$ and a linear β schedule from 10^{-4} to 0.02. The default $K = 5$ configuration selects indices $(99, 74, 49, 24, 0)$; the CGPO result on VMAS `sampling` in Table 2 instead uses $K = 2$. MPE and MaMuJoCo use $(\eta, \sigma_{\min}, \sigma_{\mathrm{init}}, c_x) = (0, 0.3, 0.1, 2)$, so every transition has standard deviation 0.3 despite $\eta = 0$. VMAS uses $(\eta, \sigma_{\min}, \sigma_{\mathrm{init}}, c_x) = (1, 10^{-6}, 1, 5)$, giving transition standard deviations for $K = 5$ of approximately $(0.599, 0.521, 0.415, 0.247, 0.010)$. The clipping threshold c_x applies to the clean prediction at each step; the final latent is mapped to the action bounds by equation 42. Deterministic evaluation sets the initial latent and all transition noise to zero; stochastic generation retains noise at every step.

B.2 Critic Targets and Optimization Details

We use $\lambda = 0.9$ for all reported CGPO configurations. Fix an agent i and let $Q_{\bar{\phi}}$ denote its target head, following Section 3.2. We suppress the agent index in the following scalar recursion. Write $\bar{Q}_t = Q_{\bar{\phi}}(c_t, \mathbf{a}_t)$ and $\bar{Q}'_{t+1} = Q_{\bar{\phi}}(c_{t+1}, \mathbf{a}'_{t+1})$, where $\mathbf{a}'_{t+1}$ is sampled from the behavior policy before optimization. For a rollout fragment of length L_{roll}, initialize $e_{L_{\mathrm{roll}}} = 0$ and recurse backward:

$$\delta_t = r_t + \gamma(1 - \mathrm{terminated}_t)\bar{Q}'_{t+1} - \bar{Q}_t, \tag{44}$$

$$e_t = \delta_t + \gamma\lambda(1 - \mathrm{done}_t)e_{t+1}, \qquad y_t = \mathrm{stopgrad}[\bar{Q}_t + e_t]. \tag{45}$$

True termination suppresses bootstrapping, while time-limit truncation retains it. The `done` mask stops recursion at either episode boundary. At the fragment boundary the recursion ends, but the last TD residual still contains its next-state bootstrap. Applying this recursion to each agent forms the target vector $\mathbf{y}_t = (y_{1,t}, \ldots, y_{N,t})$. The collection of these vectors is stored as $\mathbf{y}$ before optimization and remains fixed throughout the policy epochs. The critic regresses factual values to these stopped targets y_t using squared error, except for VMAS balance, which uses smooth-L_1 loss. The critic-loss coefficient is one.

The reward r_t and boundary masks are shared across heads; their bootstrap values, and hence their finite-training targets, need not coincide.

The variance used in equation 9 is the mean squared deviation over all batch–time–agent entries, with standard-deviation floor $\epsilon_A = 10^{-8}$. A critic-only warm-up precedes actor updates. The MaMuJoCo clipping radius is $\epsilon = 0.1$, except for Walker2d, which uses 0.2. The configured target updater is called once after all optimizer updates for an optimization minibatch. We use a soft update $\bar{\phi} \leftarrow (1 - \tau)\bar{\phi} + \tau\phi$ with $\tau = 0.005$, including minibatches in which the actor update is skipped. For U soft-update calls within a collection iteration, the coefficient on its initial target parameters is $(1 - \tau)^U$. Holding this coefficient fixed across cadences matches cumulative decay, but the intermediate online parameters can still differ. At the behavior parameters, all transition ratios equal one and clipping is locally inactive; the negative gradient of equation 11 is therefore the empirical average of the stored-credit-weighted path scores, scaled by $1/K$. It does not assert unbiasedness after same-batch standardization.

The policy trace contains $K + 1$ latents and K old transition log probabilities for each agent. During loss evaluation, observations, actions, traces, old log probabilities, value targets, and counterfactual credits are detached. Gradients flow only through recomputed transition means and the factual online critic. The actor and critic have disjoint optimizers.

B.3 Experimental Tasks and Evaluation Details

MPE and MaMuJoCo. We consider the Multi-Agent Particle Environments (MPE) (Lowe et al., 2017), using the task selection reported by OMAD (Li et al., 2026c). Cooperative Navigation requires three or four agents to cover landmarks while avoiding collisions. For Physical Deception, we follow OMAD's two-agent cooperative setting: agents reach a target landmark while concealing its identity from an adversary. We train only the two cooperative agents; during both training and evaluation, the adversary follows a fixed, non-learning policy that samples each action uniformly from its action bounds. Its stochastic behavior is treated as part of the environment dynamics. Table 1 reports CGPO results on Physical Deception and both Cooperative Navigation configurations, together with the comparison methods; Appendix Table 4 reports the CGPO-family variants.

We evaluate seven continuous MaMuJoCo control partitions (Peng et al., 2021): Ant 2×4, Ant 2×4d, Ant 4×2, HalfCheetah 2×3, HalfCheetah 6×1, Walker2d 2×3, and Swimmer 2×1. Each partition assigns disjoint actuator groups to decentralized agents while retaining the robot's coupled dynamics. The MaMuJoCo baseline and CGPO results report mean returns and standard deviations over five seeds per task; Appendix Table 4 reports the corresponding no-CF results.

MPE/MaMuJoCo comparison protocol. For Table 1, CGPO and OMAD use 3M environment steps, while MAPPO uses 10M. HATD3, HASAC, MADPMD, and MASDAC use 10M environment steps, following Li et al. (2026c). Table 1 summarizes training seeds $\{0, 1, 2, 3, 4\}$. We report final scheduled-evaluation returns at the end of each method's stated training budget, with ten evaluation episodes per checkpoint and mean and standard deviation over five training seeds; we do not select the best checkpoint. The learning curves in Figure 4 retain task-specific display horizons; their endpoints need not equal these table statistics.

VMAS. We use the BenchMARL implementations (Bettini et al., 2024b) of three vectorized continuous-control tasks from VMAS (Bettini et al., 2024a): `balance`, `sampling`, and `multi_give_way`. `balance` gives shared progress and failure rewards while agents at different support locations have unequal influence on a package; `sampling` requires spatially coordinated coverage; and `multi_give_way` requires four agents to cross a shared central region while avoiding collisions. Each VMAS run uses 3M environment frames, evaluation every 120k frames, and ten evaluation episodes per checkpoint. All VMAS experiments, including the diffusion-depth sweep and diagnostics, use seeds $\{0, 1, 2, 3\}$. All entries in Table 2 use four training seeds. We use the final scheduled evaluation rather than selecting the best checkpoint, and plot all scheduled evaluation checkpoints without smoothing.

Environment and input configuration. MPE uses PettingZoo's `simple_spread_v3` and `simple_adversary_v3` with a 25-step episode limit. MaMuJoCo uses a 1,000-step limit, the listed actuator partitions, and `agent_obsk=1`. Its actor observations are homogenized and normalized, with a one-hot agent identity appended. VMAS uses episode limits of 100 steps for balance and sampling and 200 for multi_give_way. MPE and VMAS actors use local observations without an appended agent ID. For MPE and MaMuJoCo, the critic receives the environment-provided global state concatenated with the flattened joint action. For VMAS, it receives all agents' observation–action pairs concatenated into a centralized input. Continuous actions are concatenated directly, without discretization.

Rollout and reward processing. We use ten parallel environments per worker. Each collection contains 300 transitions per environment for MaMuJoCo, 250 for MPE, and 600 for VMAS. Episode boundaries and collector boundaries are handled separately in the return recursion. Training rewards are not scaled or normalized; MaMuJoCo observation normalization does not change its reward scale. Cooperative Navigation uses `local_ratio=0.5`. VMAS balance uses shared rewards, whereas sampling uses individual rewards (`shared_rew=false`). The evaluation field `episode_reward_mean` averages raw episodic rewards over agents; for MaMuJoCo, identical rewards across agents make this equal to the underlying environment return.

Models and optimization. In the VMAS study, actors and critics use two-layer MLPs with 256 hidden units. CGPO uses five latent transitions by default and two for the `sampling` result in Table 2, together with PPO clipping, independent target joint-Q heads predicting the same team return, and truncated TD(λ) with $\lambda = 0.9$. The main CGPO results use $M = 8$ on `balance`, $M = 16$ on `sampling`, and $M = 4$ on `multi_give_way`. The corresponding `sampling` no-CF ablation also uses $K = 2$. The sample-budget sweep uses the same task-specific generation depth for all $M \in \{4, 8, 16\}$: $K = 5$ for `balance` and $K = 2$ for `sampling`. The diffusion-step sweep fixes $M = 4$ for `balance` and $M = 16$ for `sampling`. The diffusion actor selects its generation steps from a 100-step linear schedule and uses a positive reverse standard-deviation floor. Training, counterfactual sampling, and TD bootstrapping are stochastic; scheduled evaluation sets the initial latent and transition noise to zero. This evaluation decode differs from the stochastic training policy analyzed in Section 4. The stated settings are shared across tasks except for the explicitly listed generation depths, sample counts, and environment-specific collection settings.

Task-specific training settings. Table 3 collects the shared settings and task-specific overrides. Each environment transition contains the joint transition of all agents. Nominal epochs specify the minibatch-update budget; random sampling from the current rollout does not constitute exhaustive shuffled passes. The MaMuJoCo KL gate can reduce actor steps without reducing critic or target updates. All these CGPO configurations use the frozen-credit procedure in Algorithm 1: factual and alternative values come from the same target-critic snapshot, alternatives are sampled from the behavior actor, and standardized credit is stored before the optimization minibatches and reused without recomputation.

Cross-task VMAS aggregation. Figure 2(b) uses the default BenchMARL `marl-eval/rliable` aggregation pipeline (Bettini et al., 2024b; Gorsane et al., 2022; Agarwal et al., 2021) for methods with a complete three-task matrix. For each task t, evaluation returns are min–max normalized as $z = (R - L_t)/(U_t - L_t + 10^{-6})$, where L_t and U_t are the minimum and maximum returns across the methods, runs, and evaluation entries supplied for that task. These are empirical reference bounds, not random-policy and optimal-policy scores. Episode returns are normalized before averaging within each evaluation; the final scheduled evaluation supplies one score for each of four training seeds and three tasks.

Each method therefore supplies a 4×3 matrix to the aggregate estimators. The median is the median of the three task-wise seed means; the mean averages those task means. The interquartile mean (IQM) is the 25% trimmed mean over the pooled seed–task scores. The optimality gap is $1 - \mathrm{mean}(\min(z, 1))$, using normalized score 1 as its reference rather than a known task optimum. The default 50,000 bootstrap replicates give 95% percentile intervals by resampling training runs with replacement independently within each task, keeping the three tasks and normalization bounds fixed. Evaluation episodes are not separate bootstrap units, and tasks are not resampled.

All actors and critics use two 256-unit hidden layers with Tanh activations and no normalization layers. Adam uses betas $(0.9, 0.999)$, epsilon 10^{-6}, zero weight decay, and constant learning rates; gradients are clipped to norm 5 per loss. Warm-up is measured in collection iterations.

Table 3: Core CGPO hyperparameters. Asterisks mark default values with task-specific overrides listed below. The no-CF ablation uses the same settings.

Parameter	**MaMuJoCo**	**MPE**	**VMAS**
Training and optimization			
Rollout batch size	3,000	2,500	6,000
Minibatch size	300	250	400
Epochs	4	4	45
Actor learning rate	3×10^{-5}	3×10^{-5}	5×10^{-5}
Critic learning rate	5×10^{-4}	5×10^{-4}	5×10^{-5}
Discount γ	0.99	0.99^*	0.99
Ratio clip ϵ	0.1^*	0.1	0.2
CF samples M	4^*	4	4^*
Critic warm-up	5	5	5^*
KL threshold	0.03	Disabled	Disabled
Diffusion policy			
Diffusion steps K	5	5	5^*
Noise multiplier η	0	0	1
Noise floor $\sigma_{\min}$	0.3	0.3	10^{-6}
Initial latent std. σ_{init}	0.1	0.1	1
Clean prediction clip c_x	2	2	5

**Task-specific overrides.* MaMuJoCo: Walker2d uses $\epsilon = 0.2$ and $M = 16$.
MPE: Cooperative Navigation with $N = 3$ uses $\gamma = 0.95$.
VMAS: `balance` uses $M = 8$; `sampling` uses $M = 16$ and $K = 2$; `multi_give_way` uses a critic warm-up of 20 collection iterations.

Actor update gating. For MaMuJoCo, the minibatch approximate KL statistic is the mean of $(\omega - 1) - \log \omega$ over transition ratios. If it exceeds 0.03, that minibatch's actor update is skipped before backpropagation. Critic and target updates continue, and subsequent minibatches are still processed. This gate is disabled for VMAS and MPE.

Execution environment. Our implementation is built on BenchMARL and uses Python 3.10. We ran the experiments on a shared server equipped with eight NVIDIA A100 GPUs, each with 80 GB of memory.

Intervention computation. For B factual entries, interventions require BNM additional single-agent generation paths of K transitions and BNM joint-Q evaluations. Vectorization improves utilization; chunking trades memory for batching without removing this work. Appendix C.5 reports measured end-to-end training times on MPE and MaMuJoCo.

B.4 Internal Ablations and Diagnostic Implementations

Factual-value credit (Factual-Q). The ablation without a counterfactual baseline (Factual-Q in Table 4) retains the CGPO actor, independently parameterized critic heads, TD targets, target updater, and task-specific training hyperparameters. It removes only the counterfactual baseline and applies the same collection-batch standardization and storage procedure to the factual critic values:

$$\widetilde{A}_i^{\mathrm{noCF}} = \mathrm{stopgrad}\left[\frac{Q_{\bar{\phi}}(c, \mathbf{a}) - \mu_{\mathcal{B}}^{Q}}{\max(\sigma_{\mathcal{B}}^{Q}, \epsilon_A)}\right], \qquad i = 1, \ldots, N. \tag{46}$$

Here $\mu_{\mathcal{B}}^{Q}$ and $\sigma_{\mathcal{B}}^{Q}$ are the mean and population standard deviation of factual target-Q values over batch–time–agent entries. Agent i uses its own head's factual value. These factual-Q weights are standardized and stored before policy epochs. Both variants compute population statistics over the entire collection batch and freeze the standardized weights before optimization. Neither recomputes nor restandardizes them within minibatches or subsequent epochs. Their normalization rule is identical, although its input values and hence its resulting statistics differ.

Joint-action-marginalized baseline (Global advantage). The Global advantage ablation retains the diffusion actor, joint-action critic heads, TD targets, target updater, and task-specific training settings of CGPO, while replacing its teammate-conditioned baseline with a baseline that marginalizes all agents' actions. For each factual entry, we hold the centralized context c and actor inputs $\mathbf{o}$ fixed and draw M independent joint alternatives from the frozen behavior policy,

$$\widetilde{\mathbf{a}}^{(m)} \overset{\mathrm{iid}}{\sim} \prod_{j=1}^{N} \pi_{\mathrm{old},j}(\cdot \mid o_j), \qquad m = 1, \ldots, M,$$

independently of the factual joint generation path. With the head index explicit, the baseline and raw credit are

$$\widehat{V}_i^{\mathrm{GB}}(c, \mathbf{o}) = \frac{1}{M} \sum_{m=1}^{M} Q_{\bar{\phi},i}(c, \widetilde{\mathbf{a}}^{(m)}), \tag{47}$$

$$\widehat{A}_i^{\mathrm{GB}} = \mathrm{stopgrad}\Big[Q_{\bar{\phi},i}(c, \mathbf{a}) - \widehat{V}_i^{\mathrm{GB}}(c, \mathbf{o})\Big]. \tag{48}$$

The factual value and all alternative values use the same frozen target-critic head for agent i; no additional state-value network is introduced. The joint alternative pool is shared across heads, but each head computes its own baseline, so agents' credits need not coincide. Unlike CGPO, this baseline does not condition on teammates' factual actions. It estimates the full behavior-policy action expectation of the learned critic, rather than the exact V^{π} used in the theoretical comparison, and finite M introduces Monte Carlo sampling noise.

As in CGPO and Factual-Q, raw credits are standardized jointly over batch–time–agent entries, detached, stored before policy epochs, and reused without recomputation. Generation depth is held fixed at the corresponding CGPO setting, including $K = 2$ for VMAS sampling. This comparison isolates the choice between marginalizing the full joint action and marginalizing only the focal agent's action while retaining factual teammate actions.

Gaussian-policy variants. For Table 5, we replace the diffusion actor with a stochastic diagonal Gaussian actor. Given local input o_i, the network outputs a mean $\mu_\theta(o_i)$ and scale logits $z_\theta(o_i)$, with componentwise standard deviation

$$\sigma_\theta(o_i) = \mathrm{softplus}(z_\theta(o_i) + b) + 0.01, \qquad \mathrm{softplus}(b) + 0.01 = 1.$$

A Gaussian sample is mapped through $\tanh$ and scaled to the environment action bounds, yielding a TanhNormal policy. Its action log density sums over the agent's action dimensions and includes the Jacobian of the action transformation. The actor uses the standard per-agent PPO ratio

$$\omega_i(\theta) = \frac{\pi_\theta(a_i \mid o_i)}{\pi_{\mathrm{old}}(a_i \mid o_i)}$$

in place of the diffusion-transition ratios in equation 11, with one clipped-surrogate term per agent and no generation-step average. This is the density ratio for one agent's full action vector, not the product over agents. The log ratio is clamped to $[-20, 20]$ before exponentiation.

The joint-action critic heads, TD targets, target-update schedule, critic warm-up, and applicable task-specific hyperparameters follow the CGPO configuration. Gaussian + CF uses M independent alternatives from its behavior Gaussian policy, with the task-specific budgets in Table 3. Factual and alternative values use the same target-critic snapshot. Gaussian + Factual-Q removes only this baseline and retains the same Gaussian actor and training settings. Both variants standardize their raw credits over the collection batch before policy epochs and freeze them throughout optimization. Thus the Gaussian pair differs only in its baseline; diffusion-specific generation parameters are inapplicable. Training and counterfactual sampling are stochastic, while scheduled evaluation uses the transformed Gaussian mean. Training budgets, seed counts, and final-return evaluation follow Appendix B.3.

Offline covariance diagnostics. Figure 7 evaluates four training seeds per VMAS task at collection iterations 83, 250, and 499 (0.498M, 1.500M, and 2.994M frames). At each frozen checkpoint, four contexts are selected at evenly spaced positions from a batch collected over eight episodes with environment seeds 6000–6007. Four diagnostic replicates use isolated random-number streams with seeds 91000–91003. Each replicate generates fresh factual actions and latent paths and recomputes their scores; these paths remain fixed during its counterfactual resampling.

The entries in Corollary 6 are context–agent path pairs, so $L = 4N$ when all N agents are included, rather than the number of contexts alone. Each replicate draws a 512×16 counterfactual-value pool per entry. For $M \in \{1, 2, 4, 8, 16\}$, the first M alternatives in each repetition form its baseline. Gradients are first averaged across the L entries; their covariance trace is then estimated over the 512 repetitions using denominator 511. Consequently, different budgets reuse nested samples. A separate, independent pool of 8192 alternatives per entry estimates the value variance for the $1/M$ prediction, including the $1/L^2$ factor for the average gradient.

Empirical traces and predictions are computed separately for each diagnostic replicate, then arithmetically averaged over the four replicates within each training seed. The repetitions are not pooled before covariance estimation. Across the four training seeds, the figure reports geometric means and bands $\exp(\mu_{\log} \pm \sigma_{\log})$. These bands describe variation across training seeds; the four diagnostic replicates are repeated measurements of the same trained checkpoint.

C Additional Experimental Results

We first provide the return plots accompanying the main comparisons, then organize the additional evidence around credit ablation, generation depth, finite-sample diagnostics, and qualitative policy behavior.

C.1 Supplementary Performance Comparisons

Figures 3 and 4 complement the performance comparisons in Section 5.2 with final-return plots and learning curves.

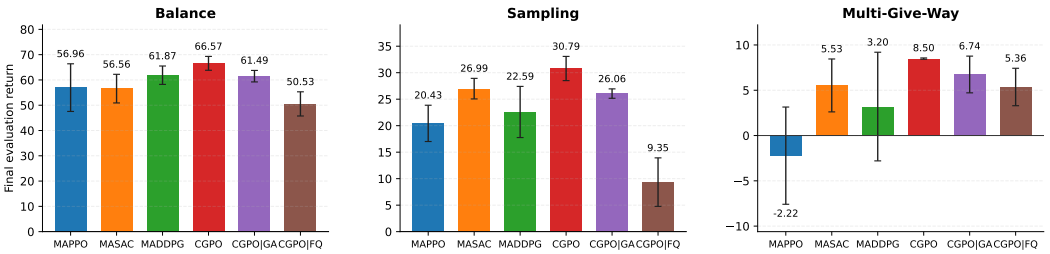


Figure 3: Four-seed VMAS final scheduled-evaluation returns. Bars show means with error bars of one sample standard deviation.

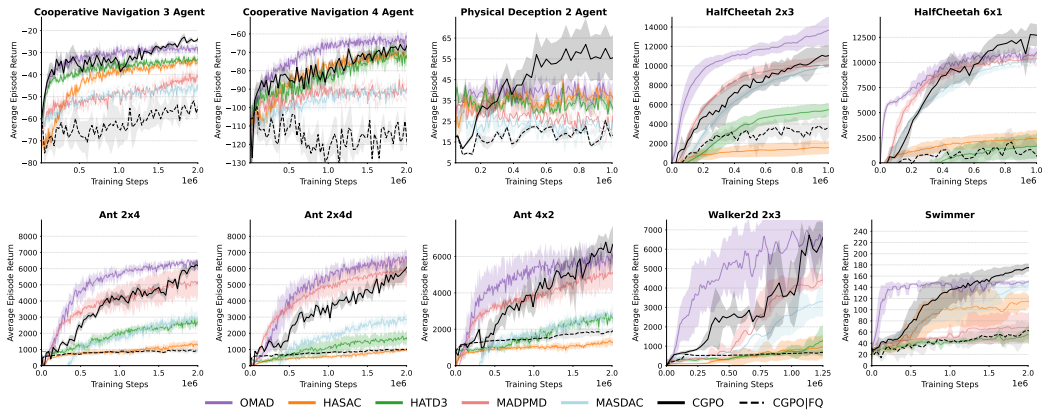


Figure 4: Learning curves on MPE and MaMuJoCo. Black solid curves show CGPO; black dash-dotted curves show its Factual-Q ablation (CGPO|FQ in the legend). Colored curves show the comparison methods identified in the legend. The panels use task-specific display horizons; Table 1 reports final scheduled-evaluation returns at each method's stated training budget.

C.2 Credit-Signal Ablation across Benchmarks

Table 4 compares three training signals for diffusion policies: counterfactual credit (CGPO), Factual-Q, and Global advantage. CGPO has higher reported means than both ablations on all thirteen listed tasks, spanning MPE, MaMuJoCo, and VMAS.

Supplementary Gaussian-policy comparison. Table 5 reports the available CF and Factual-Q results for Gaussian actors implemented as described in Appendix B.4. Among tasks with both results, CF has a higher mean on all seven MaMuJoCo partitions, Physical Deception, and all three VMAS tasks, but a lower mean on both Cooperative Navigation configurations. This supplementary comparison illustrates task-dependent effects of credit construction; it does not constitute a complete comparison of policy representations and credit signals.

C.3 Generation Depth, Comparison Budgets, and Covariance Diagnostics

These plots accompany Section 5.4. Figures 5 and 6 examine the comparison budget M and generation depth K, respectively; Figure 7 examines the finite-sampling covariance prediction with the actor and critic held fixed.

C.4 Action Quality and Team Coordination

We evaluate final checkpoints from 3M-frame training runs to connect the credit ablation to individual action quality and team behavior. The comparisons use CGPO, the Factual-Q ablation (labeled CGPO-FQ in the figures), and MAPPO, with four training seeds and shared evaluation contexts or initial worlds. These diagnostics use stochastic action sampling and execution; Tables 2 and 4 report

Table 4: Diffusion-policy returns with different credit signals (mean ± SD). CGPO results reproduce Tables 1 and 2. Bold indicates the highest mean in each task.

Task	CGPO (ours)	Global advantage	Factual-Q
MPE			
Coop. Navigation ($N = 3$)	$\mathbf{-21.3}_{\pm 1.35}$	$-22.73_{\pm 5.99}$	$-40.65_{\pm 3.75}$
Coop. Navigation ($N = 4$)	$\mathbf{-38.8}_{\pm 0.7}$	$-55.90_{\pm 14.72}$	$-66.2_{\pm 1.8}$
Physical Deception ($N = 2$)	$\mathbf{61.9}_{\pm 13.4}$	$13.82_{\pm 11.32}$	$33.4_{\pm 3.6}$
MaMuJoCo			
Ant 2×4	$\mathbf{10849.2}_{\pm 132.0}$	$8924.7_{\pm 529.2}$	$2928.0_{\pm 317.6}$
Ant 2×4d	$\mathbf{11059.6}_{\pm 503.0}$	$8485.8_{\pm 606.9}$	$2926.8_{\pm 281.6}$
Ant 4×2	$\mathbf{9319.3}_{\pm 341.76}$	$7754.0_{\pm 523.6}$	$5452.8_{\pm 676.4}$
HalfCheetah 2×3	$\mathbf{16746.8}_{\pm 1034.2}$	$10110.0_{\pm 1002.4}$	$5779.2_{\pm 607.4}$
HalfCheetah 6×1	$\mathbf{14242.6}_{\pm 623.4}$	$11083.5_{\pm 105.0}$	$7612.2_{\pm 850.8}$
Walker2d 2×3	$\mathbf{9447.1}_{\pm 869.2}$	$7875.9_{\pm 3024.9}$	$5882.6_{\pm 744.8}$
Swimmer 2×1	$\mathbf{188.9}_{\pm 2.6}$	$123.5_{\pm 3.0}$	$49.2_{\pm 8.3}$
VMAS			
Balance	$\mathbf{66.57}_{\pm 2.74}$	$61.49_{\pm 2.27}$	$50.53_{\pm 4.78}$
Sampling	$\mathbf{30.79}_{\pm 2.29}$	$26.06_{\pm 0.89}$	$9.35_{\pm 4.58}$
Multi-Give-Way	$\mathbf{8.50}_{\pm 0.07}$	$6.74_{\pm 2.03}$	$5.36_{\pm 2.07}$

Table 5: Supplementary credit ablation with Gaussian policies (mean ± SD). CF uses counterfactual credit; Factual-Q uses factual critic values.

Task	Gaussian + CF	Gaussian + Factual-Q
MPE		
Coop. Navigation ($N = 3$)	$-36.15_{\pm 2.85}$	$-32.79_{\pm 2.19}$
Coop. Navigation ($N = 4$)	$-53.8_{\pm 3.6}$	$-46.22_{\pm 0.36}$
Physical Deception ($N = 2$)	$13.74_{\pm 2.94}$	$10.88_{\pm 3.48}$
MaMuJoCo		
Ant 2×4	$7898.8_{\pm 757.2}$	$3204.8_{\pm 237.3}$
Ant 2×4d	$8787.4_{\pm 692.0}$	$3010.4_{\pm 416.5}$
Ant 4×2	$5311.4_{\pm 2232.4}$	$4345.6_{\pm 428.5}$
HalfCheetah 2×3	$4487.8_{\pm 838.9}$	$2697.2_{\pm 229.0}$
HalfCheetah 6×1	$5219.7_{\pm 424.5}$	$2369.4_{\pm 118.0}$
Walker2d 2×3	$2179.0_{\pm 226.9}$	$1126.5_{\pm 211.4}$
Swimmer 2×1	$73.53_{\pm 17.63}$	$34.7_{\pm 25.9}$
VMAS		
Balance	$47.11_{\pm 7.24}$	$21.22_{\pm 3.56}$
Sampling	$18.23_{\pm 0.62}$	$3.94_{\pm 0.60}$
Multi-Give-Way	$5.37_{\pm 2.13}$	$-4.11_{\pm 1.19}$

scheduled returns with deterministic decoding. Training seeds, rather than repeated executions, are the units of replication. The behavior comparisons assess CGPO relative to Factual-Q and MAPPO; the Global-advantage comparison is reported separately in the return tables.

Conditional action quality. Figure 8 evaluates actions on `sampling` while fixing the context and teammates' first actions. The upper panels show 128 action draws per method and training seed at four shared contexts, one from each task phase. The branch evaluation uses 32 phase-stratified shared contexts and 16 candidate actions per method and seed. Teammate actions come from a common method-balanced candidate pool; only the evaluated agent's first action changes. Each branch then uses the same frozen continuation policy. This controls the setting in which candidate actions are compared and evaluates their consequences through environment returns rather than the learned critic.

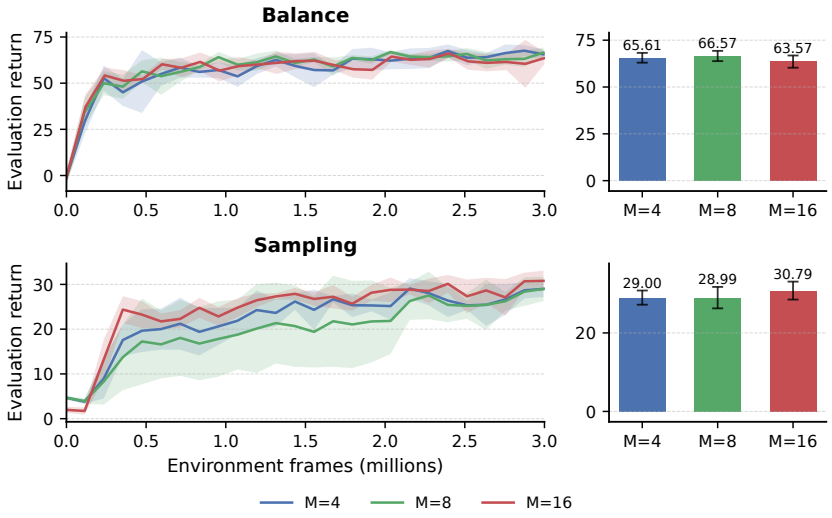


Figure 5: Counterfactual sample-count ablation for CGPO on VMAS, fixing the task-specific best-performing generation depth across all budgets ($K = 5$ for balance and $K = 2$ for sampling). Left: evaluation curves over 3M frames. Right: final scheduled-evaluation returns. Curves and bars show means; shaded bands and error bars show one sample standard deviation over four training seeds. The highest mean occurs at $M = 8$ for balance and $M = 16$ for sampling.

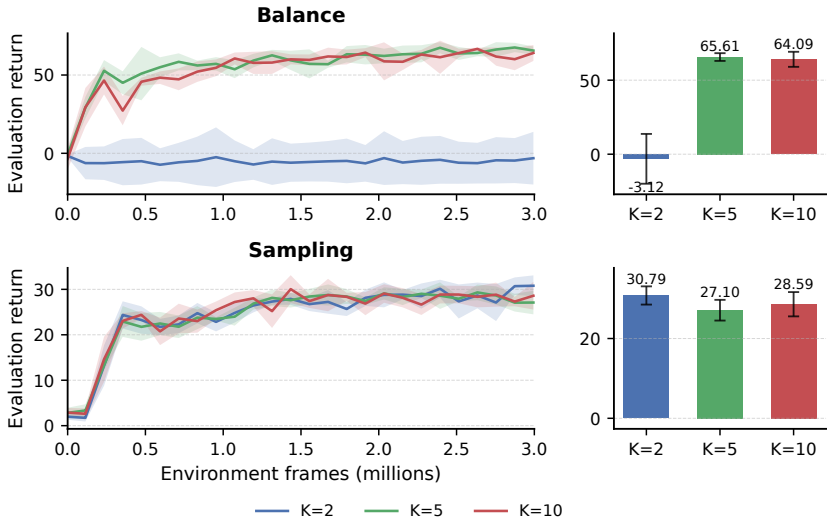


Figure 6: Diffusion-step ablation for CGPO with $K \in \{2, 5, 10\}$, holding all other hyperparameters fixed within each task. Left: evaluation curves over 3M frames. Right: final scheduled-evaluation returns. Curves and bars show means over four seeds; shaded bands and error bars show ± 1 SD.

Let Q_{env} denote the branch return. The displayed A_{env} subtracts the mean return of an independent uniform reference-action batch. The favorable-action probability is $P(Q_{\text{env}} > q^{\text{ref}}_{0.9})$, where $q^{\text{ref}}_{0.9}$ is that reference distribution's 90th percentile. This reference-centered return differs from the behavior-policy counterfactual advantage used in training: it depends on the chosen continuation and is not each method's Q^{π}.

CGPO has higher mean reference-centered branch returns and favorable-action probabilities than Factual-Q. The paired branch-return difference is positive for all four training seeds in the phase-stratified evaluation and the reported robustness comparisons, including a MAPPO continuation. Boundary concentration alone does not explain action quality: MAPPO also samples boundary

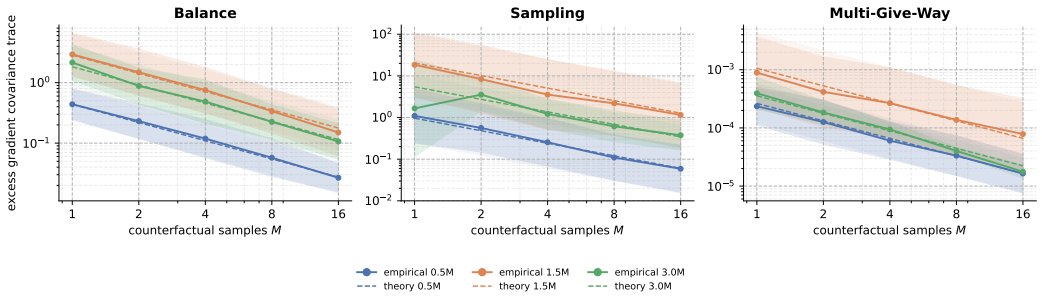


Figure 7: Finite-sample covariance diagnostics for CGPO on three VMAS tasks over four training seeds. Colors denote checkpoints near 0.5M, 1.5M, and 3.0M frames. Each checkpoint uses four diagnostic replicates at fixed actor, critic, and contexts. Within each replicate, factual paths are fixed: solid lines show covariance traces of gradients averaged over context–agent entries, estimated from 512 counterfactual repetitions; dashed lines show $1/M$ predictions from an independent pool of 8192 alternatives per entry. Replicate estimates are averaged within each training seed. Lines and bands show geometric means and ± 1 standard deviation in log space across training seeds.

actions but has lower mean branch values than CGPO. These observations support the learned policy's conditional action quality; they do not directly measure policy-gradient variance.

Coverage and successful team allocations. Figure 9 connects action quality to team behavior on `sampling`. We evaluate 32 shared initial worlds with 32 stochastic repeats per method and training seed. Weighted coverage measures the fraction of total reward-density mass collected by the team. We define a high-quality execution by weighted coverage of at least 0.2 and report both its probability and the entropy of region-to-agent assignment labels conditional on high-quality executions. Conditioning on quality distinguishes alternative successful allocations from variability due to unsuccessful behavior.

CGPO exceeds the coverage threshold in approximately 96–99% of executions across training seeds, compared with 0.2–7% for Factual-Q and 23–25% for MAPPO. Its successful executions retain variation in region-to-agent assignments. The representative seed-0 rollouts illustrate the difference: CGPO agents collect reward across the density regions, while the Factual-Q and MAPPO examples concentrate reward collection on one agent. The shared-world aggregate statistics, rather than the selected trajectories alone, support the comparison. Conditional entropy for Factual-Q is based on relatively few successful executions and should be interpreted together with its success probability.

Reliable passage with fewer collisions. Figure 10 examines `multi_give_way` in its fixed corridor layout over 16 stochastic repeats for each of four training seeds, with a 200-step episode budget. We record the number of agents reaching their goals, collision steps per episode, and passing-order conventions. For each seed, the modal passing order defines its convention; within-seed consistency is the fraction of repeats following that order. The cross-seed statistic is the largest fraction of seeds sharing a modal order.

CGPO achieves two arrivals in every evaluated repeat, with mean collision counts ranging from approximately 0.4 to 5 steps across seeds, compared with 12–37 for Factual-Q. Factual-Q achieves two arrivals only on its best training seed, while MAPPO has no arrivals on three of four seeds. Two is the largest arrival count observed in these evaluations; this does not establish a task-wide performance ceiling. In the illustrated seed-0 execution, both CGPO and Factual-Q deliver two agents, but their collision counts are 2 and 98, respectively, demonstrating why arrival counts alone do not capture coordination quality.

Three CGPO training seeds share the same modal passing order, and each CGPO seed follows its own modal order in every repeat. This is evidence of consistent conventions, while successful passage and low collision counts provide the measures of task quality. Together with Figure 9, the results show task-dependent coordination: varied successful allocations in coverage and repeatable passage in a constrained corridor. These behavioral outcomes complement the return comparisons and the finite-sampling covariance diagnostic in Figure 7.

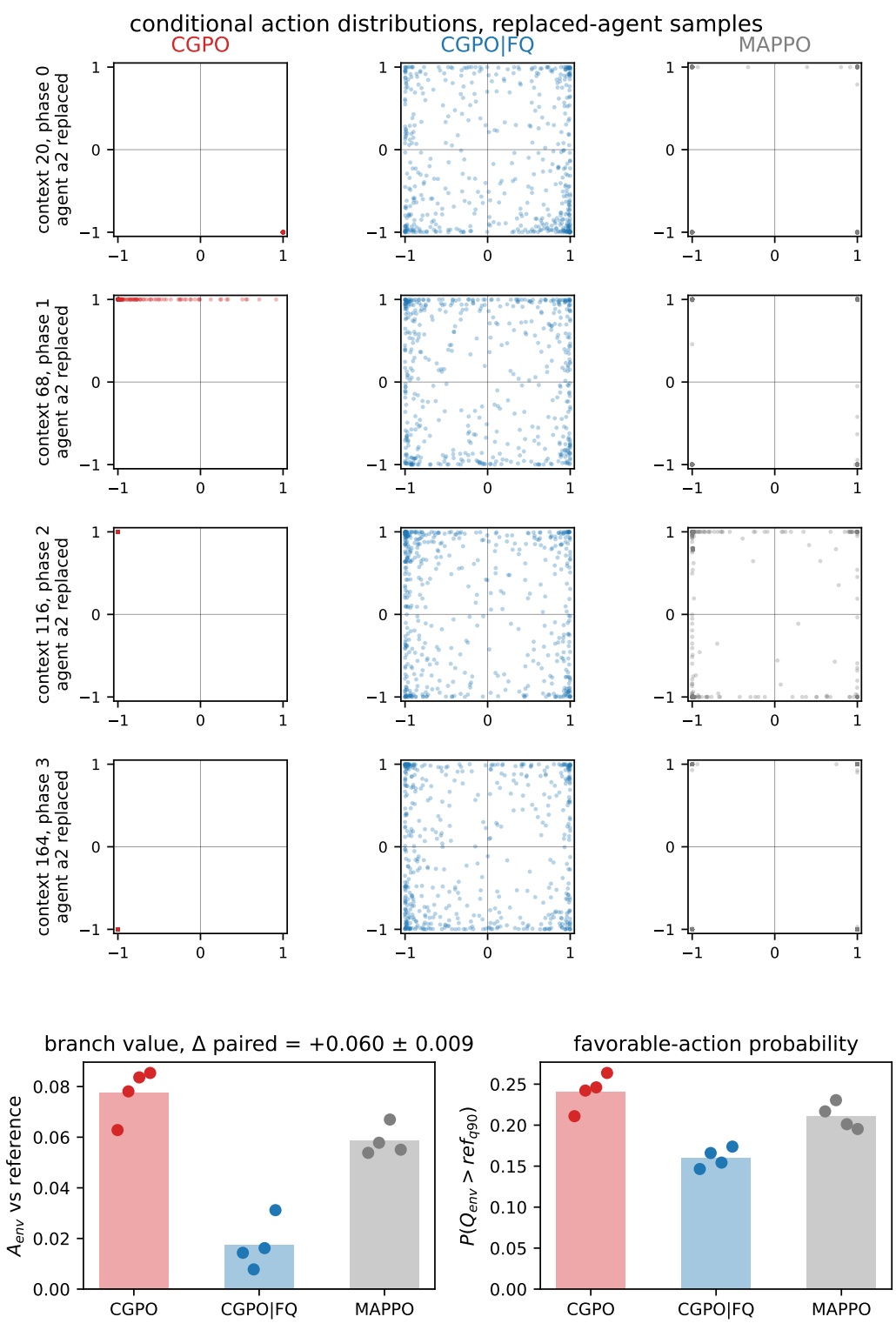


Figure 8: Conditional action quality on `sampling`. Top: action samples at four shared contexts. Bottom: reference-centered environment-branch returns and favorable-action probabilities under fixed teammates' first actions and a common continuation. Bars show means and points show training seeds. CGPO-FQ denotes the Factual-Q ablation.

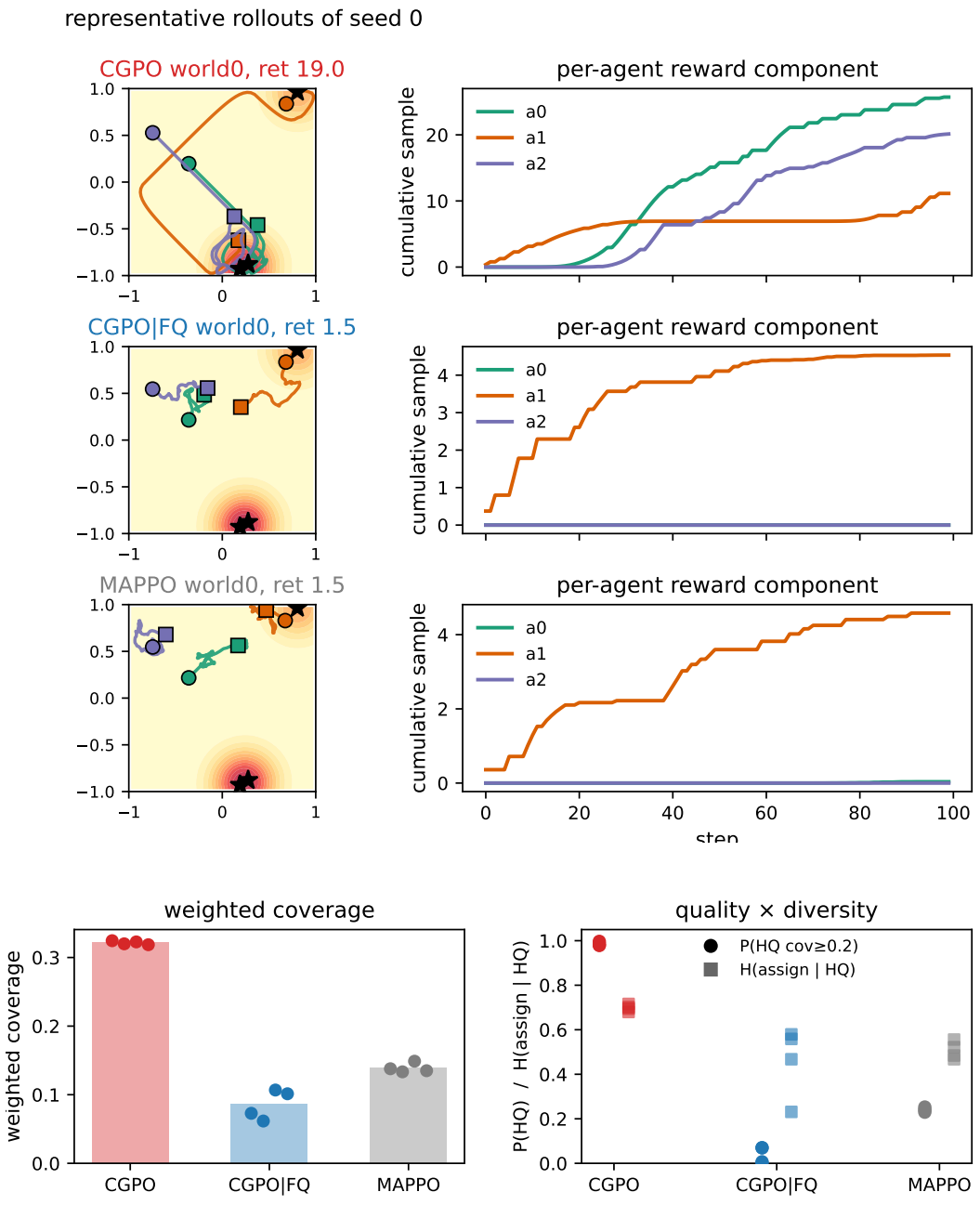


Figure 9: Team coverage and allocation on `sampling`. Top: representative trajectories and per-agent cumulative reward components in one shared world. Bottom left: weighted coverage, with mean bars and training-seed points. Bottom right: high-quality execution probability (circles) and assignment entropy conditional on high quality (squares). CGPO combines more reliable coverage with multiple successful allocations.

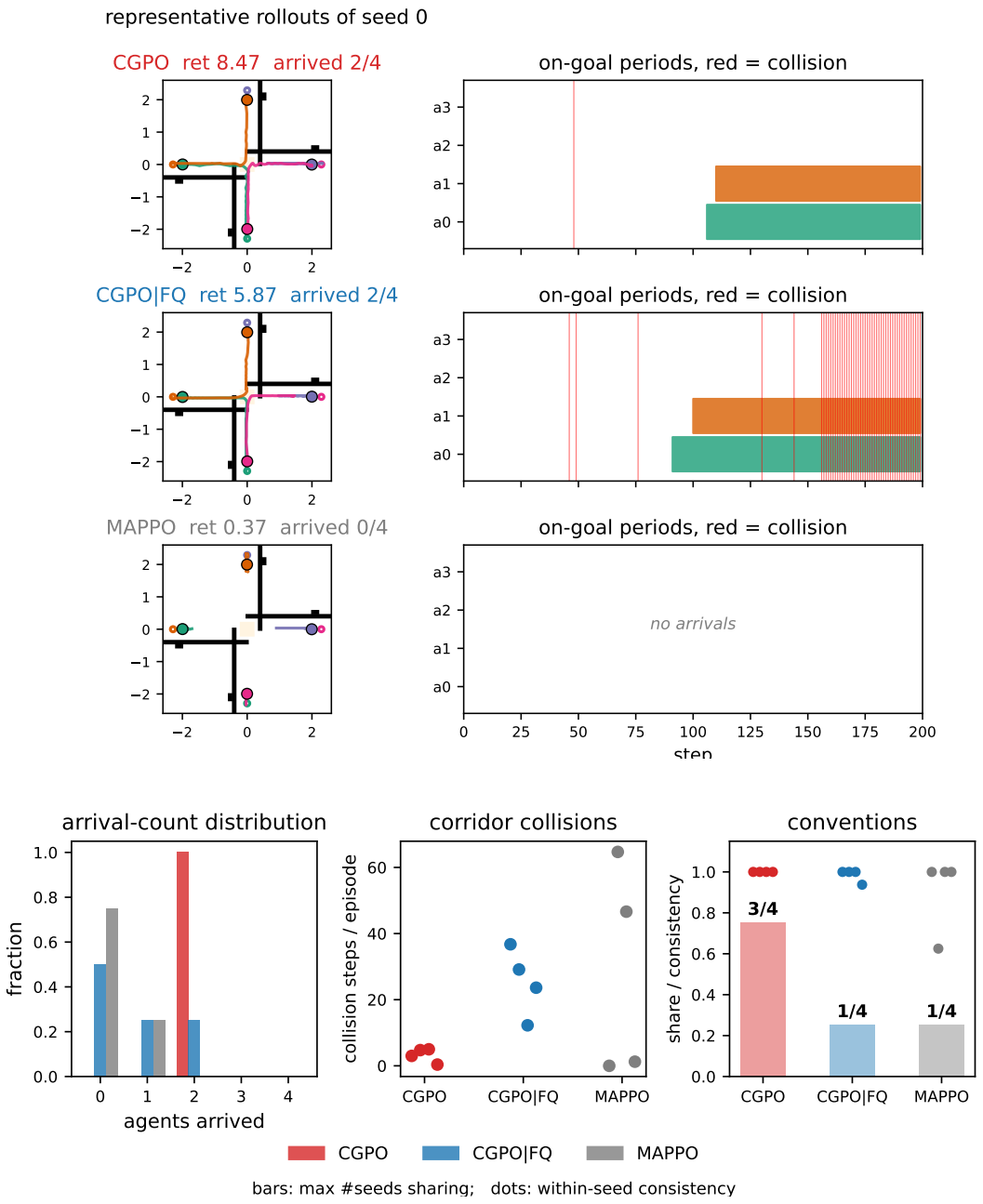


Figure 10: Coordination reliability on `multi_give_way`. Top: representative trajectories and on-goal periods, with collision steps in red. Bottom: arrival-count distribution, per-seed mean collision counts, and passing-order conventions. Convention bars show the largest fraction of seeds sharing a modal order; points show within-seed consistency. CGPO attains two arrivals reliably with fewer collisions than Factual-Q.

C.5 Computational Cost

Figure 11 compares measured training times for MAPPO, CGPO, Global advantage (CGPO—GA), Factual-Q (CGPO—FQ), and OMAD on MPE `simple_spread_v3` with $N = 3$ and MaMuJoCo Swimmer 2×1. All measurements use NVIDIA A100 GPUs and five seeds per method and task. For each run, we divide the cumulative training time recorded by TorchRL by the actual number of environment frames and scale the result to 1M frames.

CGPO takes 1.13 and 2.30 hours per 1M frames on Spread and Swimmer, respectively, compared with 1.11 and 2.22 hours for Global advantage and 1.16 and 2.16 hours for Factual-Q. The three credit variants have similar end-to-end training costs. OMAD takes 3.92 and 34.36 hours, approximately $3.5\times$ and $14.9\times$ the corresponding CGPO times, while MAPPO takes 0.82 and 1.80 hours.

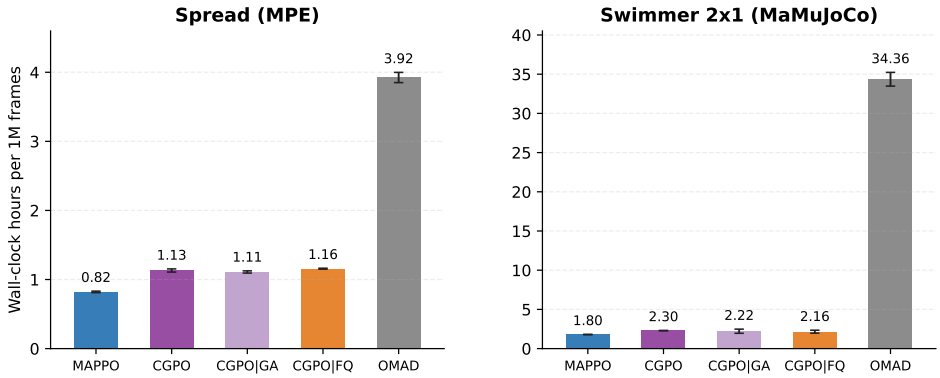


Figure 11: Training time per 1M environment frames on MPE Spread ($N = 3$) and MaMuJoCo Swimmer 2×1, measured on NVIDIA A100 GPUs. Bars show means over five seeds; error bars show ± 1 sample SD. Each run's total training time is normalized by its actual frame count.

D Additional Related Work

Value decomposition and credit assignment. DOP uses a decomposed critic, SAFE learns a default-action comparison, and Difference Advantage Estimation examines credit assignment and policy bias (Wang et al., 2021; Huang, 2026; Li et al., 2022). VDN and QMIX restrict the joint value through individual utilities (Sunehag et al., 2018; Rashid et al., 2018), whereas a centralized joint-action critic can retain arbitrary interactions. MACA considers counterfactuals at multiple levels of interaction (Zhao & Xie, 2025). These approaches provide alternative structures for credit estimation; CGPO replaces one agent's action at a time without imposing a value-decomposition constraint.

Generative policies and planning. Diffusion Policy builds on denoising diffusion models (Chi et al., 2023; Ho et al., 2020), while flow matching trains continuous transport vector fields (Lipman et al., 2023). Offline multi-agent work uses diffusion, factorization, graph structure, and sequential scores to model trajectories or policies (Zhu et al., 2024; Li et al., 2025; Zeng et al., 2025; Li et al., 2026a; Qiao et al., 2026). MADiTS uses attribution to select and stitch offline trajectory fragments (Yuan et al., 2025). Online work also includes MADP's MADDPG-style diffusion actors (Zhao et al., 2026). AID behavior-clones planner trajectories before DPPO fine-tuning for decentralized informative path planning; SCORP post-trains a joint cooperative-driving planner (Lew et al., 2026; Bai et al., 2026). MAC-Flow and OM2P study offline multi-agent flow policies (Lee et al., 2026; Li et al., 2026b). These settings differ from the on-policy comparisons of individual agents' actions and stored latent-path credit studied here.

Credit estimators and practical optimization. Wu et al. (2018) study factorized action-dependent baselines, including Monte Carlo marginalization of a joint critic. Kuba et al. (2021) analyze multi-agent gradient variance and develop score-weighted optimal baselines, including finite-sample continuous-action implementations and integration with PPO. Our policy-conditional mean provides

action-centered credit; the score-weighted baseline in Appendix A is a distinct variance-minimizing choice. We study the independent sampling cost for a generative-path score and its additive-case comparison with global advantage. Tucker et al. (2018) further show why empirical improvements must be distinguished from idealized variance arguments, analyzing value approximation, normalization, and sample reuse. We accordingly evaluate the full training procedure separately from frozen-estimator diagnostics.

Optimization spaces for generative actors. Diffusion Policy learns action distributions from demonstrations (Chi et al., 2023); Q-score matching and DPMD/SDAC provide value-guided score objectives (Psenka et al., 2024; Ma et al., 2025). Among on-policy methods, NCDPO stores noise and backpropagates through generation to optimize the executed action's noise-conditioned distribution; its football experiments use centralized joint-action generation (Yang et al., 2025). FPO substitutes a conditional-flow-matching loss ratio, whereas PolicyFlow approximates a conditional ratio through velocity changes along an interpolation path (McAllister et al., 2026; Yang et al., 2026). GenPO uses a doubled dummy-action representation; GenPO++ evaluates a joint ratio over paired solver states (Ding et al., 2025; Hu et al., 2026). These augmented ratios should be distinguished from executed-action marginal ratios. SAC Flow uses noisy flow paths with an entropy-regularized off-policy objective (Zhang et al., 2026a). CGPO trains its diffusion actor with explicit transition ratios and stored counterfactual credit.