# COORDINATED VALUE GRADIENTS FOR DECENTRALIZED TEST-TIME GUIDANCE IN OFFLINE MARL

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

In offline multi-agent reinforcement learning (MARL), diffusion and flow policies capture multimodal joint behavior, but existing methods couple the generator to critic training, which limits their scalability and adaptability. Offline RL avoids this coupling through test-time critic guidance, but extending it to cooperative MARL is nontrivial: centralized critics require global inputs, whereas locally queryable critics need not improve team value. We propose Coordinated Q-Gradient Flow (CQF), to our knowledge the first test-time guidance framework for offline MARL, whose guidance is both locally queryable and team-aligned: monotonically factorizing a team-level critic's objective yields local utilities whose simultaneous guidance steps form a first-order ascent direction of the learned team value at every joint action. The flow policy is fit by supervised matching alone, and at deployment each agent steers it using only its observation and utility gradient. A one-hot relaxation extends the same pipeline to discrete control. Across MA-MuJoCo, MPE, SMACv1, and goal-conditioned MangoBench, CQF attains the best average scores among compared Gaussian, diffusion, and flow methods.

## 1 INTRODUCTION

Multi-Agent Reinforcement Learning (MARL) coordinates decision makers in applications such as autonomous vehicles, collaborative robots, and multi-player games (Tan, 1993; Zhang et al., 2021; Osa & Harada, 2024; Wang et al., 2024a; Samvelyan et al., 2019). Offline MARL learns such policies from fixed datasets without risky or expensive online interaction (Levine et al., 2020; Formanek et al., 2023; Pan et al., 2022; Wang et al., 2023a; Shao et al., 2023). Gaussian policies, however, struggle with the sharply multimodal joint-behavior distribution and tend to produce out-of-distribution coordination; recent work therefore adopts diffusion and flow models to represent this distribution more faithfully (Zhu et al., 2024; Li et al., 2025; Lee et al., 2026).

Representing the multimodal behavior distribution is only half the problem. Existing methods inject value into the generative policy during training; while effective, this coupling of behavior modeling to value maximization creates a scalability bottleneck (Hansen-Estruch et al., 2023; Zhou et al., 2026). Multi-step trajectory planners (Zhu et al., 2024; Li et al., 2025) iteratively sample high-dimensional trajectories; critic-coupled policies (Li et al., 2026b;c) track evolving value estimates and become unstable and tuning-sensitive (Park et al., 2025); one-step alternatives (Park et al., 2025; Lee et al., 2026) require distillation. These computational, optimization, and design burdens limit the scalability and adaptability of generative offline MARL (Stiennon et al., 2020; Li et al., 2026a; Wang et al., 2023c; He et al., 2023; Ding & Jin, 2024; Zhang et al., 2024; Wang et al., 2024b; Espinosa-Dice et al., 2026; Hansen-Estruch et al., 2023; Park et al., 2025; Zhou et al., 2026).

These limitations follow from coupling the generator with the critic during training, not from using a generative policy. Single-agent offline RL already decouples behavior modeling from reward optimization: a behavior policy is fit by supervised learning alone, and a separately trained critic guides its actions only at inference, by ranking sampled candidates (Hansen-Estruch et al., 2023; Li et al., 2026a), by refining generated actions with value-gradient ascent (Mark et al., 2024; Xu et al., 2026), or by injecting critic gradients into the velocity field of a flow solver (Jang et al., 2025; Zhou et al., 2026). However, this decoupling does not transfer directly to cooperative MARL. Once value acts at test time, the critic is queried while actions are produced, so it must obey the same execution interface as the policy. In single-agent RL this is innocuous, because the critic that



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.

defines the improvement direction is exactly the critic the agent can evaluate. In a Dec-POMDP the two come apart: a centralized joint critic describes how the agents' actions interact but cannot be queried without the global state and the joint action, whereas independently trained local critics are deployable but may not point in team-improving directions. The question motivating this work is therefore: *under test-time guidance, how can offline MARL preserve decentralized execution while ensuring that a local critic still provides improvement directions aligned with the team objective?*

We propose Coordinated Q-Gradient Flow (CQF), to our knowledge the first test-time guidance framework for offline MARL, which answers this question by carrying coordination in the gradient of a value function that each agent can evaluate on its own: (i) local utilities are trained through a monotonic mixer and a team-level implicit Q-learning objective (Sunehag et al., 2017; Rashid et al., 2020; Kostrikov et al., 2022), so each utility is shaped by team credit assignment rather than by a private return; (ii) monotonic mixing makes the joint value's derivative with respect to an agent's action a nonnegative rescaling of that agent's own utility gradient, so all agents following their own utility gradients at once still take a first-order ascent step on the learned mixed value (*coordinated first-order ascent*, Proposition 1); and (iii) after training, the mixer and the global state are discarded, so an agent needs only its own observation and utility gradient, with no teammate observation, joint action, or communication. The improvement directions therefore remain computable from local information while still pointing in team-aligned directions, which is what makes test-time guidance instantiable in offline MARL.

The resulting pipeline keeps behavior modeling and reward optimization separate throughout: a local flow policy is fit by supervised flow matching, the factorized critic is trained independently, and at deployment both are frozen while every agent steers its own sampling process toward the team objective by adding its utility gradient to the learned velocity field. However, guidance differentiates the local utility with respect to its action argument and therefore presupposes a Euclidean action, which raises a challenge for discrete control. We therefore apply a simple one-hot relaxation: all networks are trained on hard one-hot encodings, while inference guides a differentiable relaxed vector and recovers an executable action by $\arg\max$, leaving the rest of CQF unchanged. Taken together, this design avoids costly and potentially unstable reward-driven updates to the generative policies during training, while retaining centralized coordination information for decentralized guidance at deployment, establishing a direction for scalable and adaptable generative offline MARL.

Our contributions are:

- We propose a monotonically factorized team-level critic to make test-time guidance both locally queryable and team-aligned: simultaneous local guidance steps form a first-order ascent direction of the learned team value at every joint action (Proposition 1), using only local observations.
- We propose CQF, to our knowledge the first test-time guidance framework for offline MARL, to keep behavior modeling and reward optimization separate, steer local flow toward the team objective with utility gradients, and extend the same pipeline to discrete actions by a one-hot relaxation.
- Across MA-MuJoCo, MPE, SMACv1, and goal-conditioned MangoBench, CQF attains the best average performance among compared Gaussian, diffusion, and flow methods, with the largest gains on mixed and suboptimal data.

## 2 Related Work

**Offline MARL.** Early offline MARL methods adapted behavior regularization and conservative value estimation. ICQ constrains value learning to observed actions (Yang et al., 2021); OMAR rectifies actors using zeroth-order joint-action search (Pan et al., 2022); and OMIGA uses global-to-local value regularization to support decentralized policies (Wang et al., 2023b). Related baselines extend CQL and TD3+BC to centralized training (Kumar et al., 2020; Fujimoto & Gu, 2021), while the Multi-Agent Decision Transformer replaces temporal-difference learning with autoregressive prediction (Kurenkov et al., 2022). Recent work curbs joint-action distribution shift inside the training objective, via uncertainty-weighted partial action replacement (Jin & Montana, 2026) or sequential updates conditioned on predecessors' actions (Formanek et al., 2025). This progres-

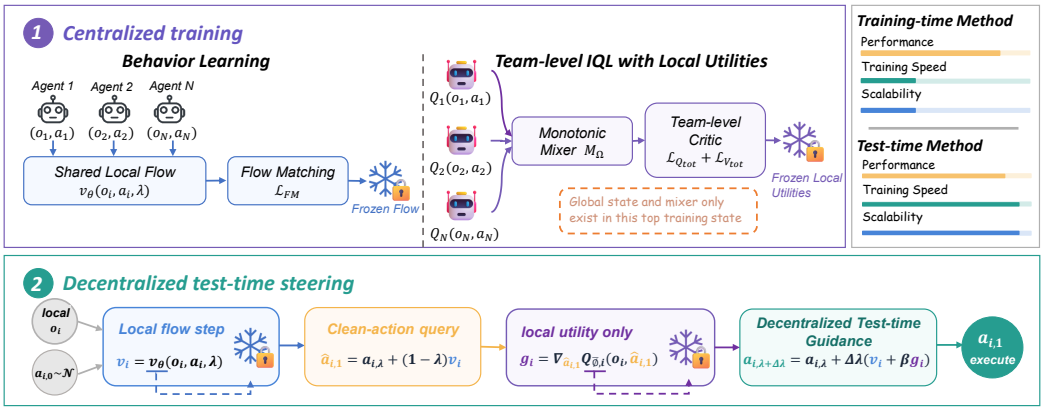


Figure 1: **Overview of CQF.** *Centralized training* fits two disjoint objectives, a local behavior flow trained by supervised matching alone and a monotonically factorized critic trained by a team-level IQL objective. *Decentralized execution*: each agent queries its own utility at the one-step clean-action estimate and adds that gradient to the learned velocity at every Euler step.

sion improved offline optimization, but Gaussian and autoregressive policies can be restrictive when demonstrations contain several incompatible coordination modes.

**Generative policies for offline MARL.** Diffusion and flow models address this limitation by representing multimodal action or trajectory distributions. Trajectory-level approaches model return-conditioned joint rollouts and recover actions through planning or inverse dynamics, requiring iterative joint sampling (Zhu et al., 2024; Li et al., 2025; Yuan et al., 2025; Zou et al., 2026). Action-level approaches instead incorporate value information directly into generative policy learning: DOM2 and OM2P add Q-maximization objectives, VGM2P uses critic-derived advantages as conditioning labels for classifier-free MeanFlow, MAC-Flow distills a Q-improved joint flow teacher into decentralized one-step policies, and OMSD decomposes the joint behavior score sequentially (Li et al., 2026b;c; Pang et al., 2026; Lee et al., 2026; Qiao et al., 2026). The last two are closest to CQF, since both also keep the generative model frozen, but both then fit a separate execution network in a second stage: a one-step student for MAC-Flow, and a score-regularized deterministic policy for OMSD. Such a stage is expensive and can cost expressivity (Zhou et al., 2026), and OMSD in particular retains only a single mode at execution, these design burdens also limit scalability of generative model (Zhou et al., 2026). In contrast, CQF learns the behavior flow independently of the critic and introduces value guidance only during action sampling, so the multimodal flow itself is executed and no second stage is trained, linking generative offline MARL to test-time policy improvement.

**Test-time policy improvement.** Single-agent test-time policy improvement methods can be categorized by how reward information modifies actions at inference. Sample-and-select methods repeatedly sample and rank actions with a critic (Hansen-Estruch et al., 2023; Li et al., 2026a); post-generation methods refine completed actions through value-gradient ascent (Mark et al., 2024; Xu et al., 2026). The former repeats full policy sampling and critic evaluation, while the latter adds iterative action-space optimization. A complementary conditioning-based family trains return- or optimality-conditioned policies and controls the degree of improvement through the conditioning variable or classifier-free guidance at inference (Kumar et al., 2019; Frans et al., 2025), thereby embedding reward information into generator training. Most closely, QFQL guides noisy flow states that may be outside the critic's training distribution (Jang et al., 2025). QGF instead queries a one-step approximation of the fully denoised action, avoiding both out-of-distribution intermediate queries and backpropagation through the complete solver (Zhou et al., 2026).

We adopt QGF's estimator but address its multi-agent coordination problem. Guidance from a centralized critic would require global information and joint actions, whereas independently learned local critics provide no team-alignment mechanism (Section 5.3). CQF resolves this tension through centralized training with monotonic value decomposition: local utilities are learned from a team-level IQL objective through a monotonic mixer, which makes each local utility gradient a positively scaled block of the learned joint-value gradient (Proposition 1). At deployment, the mixer and global state are discarded, so centralized team supervision yields decentralized guidance without teammate inputs, joint actions, or communication.

# 3 PRELIMINARIES

**Problem setting.** We consider a decentralized partially observable Markov decision process (Dec-POMDP) (Bernstein et al., 2002), $\mathcal{M} = \big(\mathcal{N}, \mathcal{S}, \{\mathcal{O}_i\}_{i=1}^N, \{\mathcal{A}_i\}_{i=1}^N, P, \{r_i\}_{i=1}^N, \gamma\big)$ with a fixed dataset $\mathcal{D}$. Here $\mathcal{N} = \{1, \dots, N\}$ indexes the agents, $\mathcal{S}$ is the global state space, $\mathcal{O}_i$ and $\mathcal{A}_i$ are the local observation and action spaces of agent $i$, and the joint action $\boldsymbol{a} = (a_1, \dots, a_N) \in \mathcal{A} := \prod_i \mathcal{A}_i$ drives the transition $P(s' \mid s, \boldsymbol{a})$. At time $t$, agent $i$ receives a local observation $o_{i,t} = O_i(s_t)$ and does not observe $s_t$ or $\{o_{j,t}\}_{j \neq i}$. The agents cooperate to maximize the team return $\mathbb{E}\big[\sum_{t \geq 0} \gamma^t \sum_{i=1}^N r_i(s_t, \boldsymbol{a}_t)\big]$. For goal-conditioned tasks (Wang et al., 2026), goals $g$ is partitioned into local goals $g_i$, and agent $i$ follows $\pi(a_i \mid o_i, g_i)$ under the sparse reward (Appendix H).

**Flow matching.** Flow matching (Lipman et al., 2023) learns a time-dependent velocity field and integrates associated ordinary differential equation, giving a simpler objective and a shorter, deterministic sampler than diffusion models (Ho et al., 2020; Song et al., 2021). Throughout, $t$ denotes environment time in the Dec-POMDP, whereas $\lambda \in [0, 1]$ is the interpolation time of the flow that transports $p_0 = \mathcal{N}(0, I)$ to the action distribution $p_1$. Let $a \sim p_{\mathcal{D}}(a \mid o)$, $\epsilon \sim \mathcal{N}(0, I)$, $\lambda \sim \mathcal{U}[0, 1]$, and set $a_\lambda = (1 - \lambda)\epsilon + \lambda a$ with target velocity $u = a - \epsilon$. A velocity field $\boldsymbol{v}_\theta(o, a_\lambda, \lambda)$ minimizes

$$\mathcal{L}_{\mathrm{FM}}(\theta) = \mathbb{E}_{(o,a)\sim\mathcal{D},\epsilon,\lambda}\left[\|\boldsymbol{v}_\theta(o, a_\lambda, \lambda) - u\|_2^2\right]. \tag{1}$$

At inference, Euler integration starts from Gaussian noise at $\lambda = 0$ and follows the velocity toward the data distribution at $\lambda = 1$ (Lipman et al., 2023).

**Implicit Q-learning and monotonic value factorization.** To mitigate the overestimation of out-of-distribution (OOD) actions caused by distributional shift in offline MARL (Levine et al., 2020; Formanek et al., 2023), we adopt Implicit Q-Learning (IQL) (Kostrikov et al., 2022), whose regression targets are built from dataset actions without explicit action maximization. Using single-agent notation, given an offline transition $(o, a, r, o', d) \sim \mathcal{D}$, IQL optimizes

$$\mathcal{L}_Q(\phi) = \mathbb{E}_{\mathcal{D}}\Big[\big(r + \gamma(1 - d)V_\nu(o') - Q_\phi(o, a)\big)^2\Big], \qquad \mathcal{L}_V(\nu) = \mathbb{E}_{\mathcal{D}}\big[L_2^\tau\big(Q_{\bar{\phi}}(o, a) - V_\nu(o)\big)\big], \tag{2}$$

where $L_2^\tau(u) = |\tau - \mathbf{1}(u < 0)|u^2$, $\tau \in (0.5, 1)$, and $Q_{\bar{\phi}}$ denotes a target critic.

Because per-agent IQL cannot capture how the joint action determines the team return, we represent the joint action-value function by a monotonic factorization,

$$Q_{\mathrm{tot}}(s, \boldsymbol{o}, \boldsymbol{a}) = M_\phi^Q\left(Q_{\phi,1}(o_1, a_1), \dots, Q_{\phi,N}(o_N, a_N); s\right), \qquad \frac{\partial Q_{\mathrm{tot}}}{\partial Q_{\phi,i}} \geq 0, \tag{3}$$

where $Q_{\phi,i}$ is the local utility of agent $i$ and $M_\phi^Q$ is a centralized mixer, parameterized together with the utilities and conditioned on the global state $s$. Monotonic factorization is commonly used to guarantee the individual-global-max (IGM) property (Son et al., 2019; Sunehag et al., 2017; Rashid et al., 2020); CQF instead exploits its pointwise gradient consequence, because guidance takes bounded ascent steps rather than computing per-agent maximizers (Proposition 1). We train the factorized critic with team-level IQL so that team supervision shapes every local utility.

# 4 COORDINATED Q-GRADIENT FLOW

We design test-time guidance for cooperative MARL around three properties: (i) the direction along which each agent improves its own action must serve the learned team value; (ii) that direction is computed at execution from the agent's own observation, without the global state, teammate inputs, or communication; and (iii) reward information must never enter the generative policy, so that behavior modeling remains supervised. We therefore learn two objects independently from the offline data (Section 4.1), which gives (iii), and a monotonically factorized critic trained by a team-level IQL objective (Section 4.2), whose local utility gradients give (i). At deployment, the mixer and the global state are discarded and each agent guides its own sampling process with its own utility gradient (Section 4.3), which gives (ii). The assembled local guidance directions form a first-order ascent of the learned team value (Proposition 1). Section 4.4 extends the same procedure to discrete actions by a one-hot relaxation; Appendix A summarizes the resulting training and execution procedure.

### 4.1 SEPARATED BEHAVIOR AND VALUE LEARNING

To keep behavior modeling supervised, the local flow $\boldsymbol{v}_\theta$ is fit by matching alone, independently of critic trained in Section 4.2. Let $\boldsymbol{a}_{i,\lambda} = (1-\lambda)\boldsymbol{\epsilon}_i + \lambda \boldsymbol{a}_{i,t}$ and $\boldsymbol{u}_i = \boldsymbol{a}_{i,t} - \boldsymbol{\epsilon}_i$ as in Eq. 1, and write $\boldsymbol{o}^{\mathrm{id}}_{i,t}$ for local observation of agent $i$ concatenated with a one-hot identifier. The local flow loss is

$$\mathcal{L}_{\mathrm{FM}}(\theta) = \mathbb{E}_{\mathcal{D},\,\boldsymbol{\epsilon}_i,\,\lambda}\Big[\tfrac{1}{N}\sum_{i=1}^{N}\big\|\boldsymbol{v}_\theta(\boldsymbol{o}^{\mathrm{id}}_{i,t}, \boldsymbol{a}_{i,\lambda}, \lambda) - \boldsymbol{u}_i\big\|_2^2\Big]. \tag{4}$$

Because $\mathcal{L}_{\mathrm{FM}}$ is disjoint from the critic objectives $\mathcal{L}_Q$ and $\mathcal{L}_V$, the flow parameters $\theta$ never receive a value gradient, so the generator remains a reusable behavior model and reward optimization is reserved for inference. Nevertheless, the flow loss is a sum of independent per-agent terms and does not encode how actions interact. That interaction is represented by the critic and applied at test time. Appendix G compares this local flow with a joint flow, where the critic does not enter the sampler.

### 4.2 TEAM-LEVEL IQL WITH LOCAL UTILITIES

The critic must therefore encode how actions interact, while remaining evaluable from local inputs at execution. A parameter-shared network maps agent $i$'s observation and action to a scalar utility $Q_{\phi,i} = Q_\phi(\boldsymbol{o}^{\mathrm{id}}_{i,t}, \boldsymbol{a}_{i,t})$, and therefore contains no global inputs. During centralized training only, a monotonic mixer $M^Q_\phi$ lifts these local utilities to a joint action value

$$Q_{\mathrm{tot},t} = M^Q_\phi\big(Q_{\phi,1}(\boldsymbol{o}^{\mathrm{id}}_{1,t}, \boldsymbol{a}_{1,t}), \ldots, Q_{\phi,N}(\boldsymbol{o}^{\mathrm{id}}_{N,t}, \boldsymbol{a}_{N,t}); \boldsymbol{s}_t\big). \tag{5}$$

A separate value mixer combines the local values analogously, $V_{\mathrm{tot},t} = M^V_\psi\big(V_{\psi,1}(\boldsymbol{o}^{\mathrm{id}}_{1,t}), \ldots, V_{\psi,N}(\boldsymbol{o}^{\mathrm{id}}_{N,t}); \boldsymbol{s}_t\big)$. The global state can shape how strongly each utility contributes to the team value, while monotonicity keeps every such contribution nonnegative.

To train the mixed critic without maximizing over potentially out-of-distribution joint actions, we apply IQL at the team level. With $d_t$ the termination flag and $r_t$ the scalar team reward $r_t = \frac{1}{N}\sum_{i=1}^{N} r_{i,t}$, the team TD target is $y_t = r_t + \gamma(1-d_t)V_{\mathrm{tot},t+1}$, and the mixed critic minimizes $\mathcal{L}_Q(\phi) = \mathbb{E}_{\mathcal{D}}\big[(Q_{\mathrm{tot},t} - y_t)^2\big]$. The mixer backpropagates this team TD error to each utility, $\partial\mathcal{L}_Q/\partial Q_{\phi,i} = 2(Q_{\mathrm{tot},t} - y_t)\,\partial Q_{\mathrm{tot},t}/\partial Q_{\phi,i}$. Unlike independently trained local critics, these utilities are shaped by team credit assignment rather than by a private return. Let $\bar{\phi}$ be a Polyak-averaged copy of $\phi$, and write $\bar{Q}_{\mathrm{tot},t}$ for the mixed joint value of this target critic. We fit the team value by expectile regression $\mathcal{L}_V(\psi) = \mathbb{E}_{\mathcal{D}}\big[L^\tau_2(\bar{Q}_{\mathrm{tot},t} - V_{\mathrm{tot},t})\big]$, where $L^\tau_2$ is the expectile loss of Eq. 2.

After this team-level IQL training, the mixer and the global state are discarded. Team credit assignment has already been injected into the local utilities, so test-time guidance uses only the local target utilities $Q_{\bar{\phi},i}$ (Section 4.3).

### 4.3 DECENTRALIZED TEST-TIME GUIDANCE

At execution, each agent steers its flow with local utility, without reconstructing centralized critic. At interpolation time $\lambda$, agent $i$ evaluates $\boldsymbol{v}_i = \boldsymbol{v}_\theta(\boldsymbol{o}^{\mathrm{id}}_i, \boldsymbol{a}_{i,\lambda}, \lambda)$ and forms a one-step clean-action estimate $\hat{\boldsymbol{a}}_{i,1} = \boldsymbol{a}_{i,\lambda} + (1-\lambda)\boldsymbol{v}_i$. Following first-order QGF, critic is queried at $\hat{\boldsymbol{a}}_{i,1}$ rather than at the noisy intermediate $\boldsymbol{a}_{i,\lambda}$, which keeps value queries closer to the training support, and the Jacobian from $\boldsymbol{a}_{i,\lambda}$ to $\hat{\boldsymbol{a}}_{i,1}$ is dropped (Zhou et al., 2026). The guidance direction is the gradient of target local utility, $\boldsymbol{g}_i = \nabla_{\hat{\boldsymbol{a}}_{i,1}} Q_{\bar{\phi},i}(\boldsymbol{o}^{\mathrm{id}}_i, \hat{\boldsymbol{a}}_{i,1})$, and is added to the learned velocity at every Euler step:

$$\boldsymbol{a}_{i,\lambda+\Delta\lambda} = \boldsymbol{a}_{i,\lambda} + \Delta\lambda\,(\boldsymbol{v}_i + \beta \boldsymbol{g}_i)\,, \tag{6}$$

where $\Delta\lambda$ is the solver step and $\beta \geq 0$ controls the strength of policy improvement. The special case $\beta = 0$ recovers the unguided behavior flow; other values can be selected after training because $\beta$ affects inference only (Section 5.4).

**Why local guidance directions serve the learned team value.** All $N$ agents compute $\boldsymbol{g}_i$ from their own utilities and add it in Eq. 6. The coordination question is whether these *guidance* directions remain aligned with the learned mixed value; monotonic factorization answers it at the level of gradients.

**Proposition 1** (Coordinated first-order ascent)**.** *Fix a global state $\boldsymbol{s}$, local observations $\boldsymbol{o}$, and a joint action $\boldsymbol{a} = (a_1, \ldots, a_N)$ at which the mixer branch of Eq. 3 is differentiable, and write $m_i := \partial Q_{\mathrm{tot}}/\partial Q_{\phi,i} \geq 0$ and $\boldsymbol{g}_i := \nabla_{a_i} Q_{\phi,i}(o_i, a_i)$. Then*

$$\nabla_{a_i} Q_{\mathrm{tot}} = m_i\, \boldsymbol{g}_i \quad \textit{for every } i, \qquad \textit{hence} \qquad \big\langle \nabla_{\boldsymbol{a}} Q_{\mathrm{tot}},\, \boldsymbol{g} \big\rangle = \sum_{i=1}^{N} m_i \|\boldsymbol{g}_i\|_2^2 \;\geq\; 0, \tag{7}$$

*where $\boldsymbol{g} = (\boldsymbol{g}_1, \ldots, \boldsymbol{g}_N)$ is the joint guidance direction formed by the local utilities (the added term in Eq. 6). Consequently $Q_{\mathrm{tot}}(\boldsymbol{s}, \boldsymbol{o}, \boldsymbol{a} + \beta \boldsymbol{g}) = Q_{\mathrm{tot}}(\boldsymbol{s}, \boldsymbol{o}, \boldsymbol{a}) + \beta \sum_i m_i \|\boldsymbol{g}_i\|_2^2 + O(\beta^2)$, and the inequality is strict whenever $m_i > 0$ and $\boldsymbol{g}_i \neq \mathbf{0}$ for some $i$.*

The identity follows from the chain rule: under Eq. 5, $a_i$ affects $Q_{\mathrm{tot}}$ only through $Q_{\phi,i}$, yielding $\nabla_{a_i} Q_{\mathrm{tot}} = m_i \boldsymbol{g}_i$, with $m_i \geq 0$ by monotonicity. At execution, the same argument applies to the target critic $\bar{\phi}$ at the joint clean-action estimate.

Together, these properties turn a centrally trained team critic into decentralized, team-aligned test-time guidance. First, whereas IGM concerns agreement between joint and per-agent maximizers, Eq. 7 establishes first-order alignment at every differentiable joint action, so each agent can apply its local utility gradient directly during test-time guidance without solving $\max_{a_i} Q_{\phi,i}$. Second, the simultaneously computed local directions assemble into a nonnegative first-order ascent direction of the learned mixed value, ensuring that the decentralized guidance remains team-aligned at first order. Third, teammate, state, and mixer information enters each joint-gradient block only through the nonnegative coefficient $m_i$; agents can therefore omit this centralized information at execution while preserving nonnegative first-order alignment with $\nabla_{\boldsymbol{a}} Q_{\mathrm{tot}}$. Independently trained local critics provide no analogous guarantee of alignment with the learned team value, and the corresponding inner product may be negative. Section 5.3 provides behavioral evidence consistent with this distinction under the same decentralized flow. Appendix C.1 verifies Eq. 7 on frozen checkpoints.

### 4.4 Discrete actions via one-hot relaxation

The guidance direction $\boldsymbol{g}_i$ requires a Euclidean derivative of local utility with respect to its action argument, which a categorical action $a_i \in \{1, \ldots, A_i\}$ does not admit. We address this with a change of representation that leaves the core of CQF intact: every discrete action is lifted to a one-hot embedding, so the losses of Sections 4.1–4.2 and the Euler update of Eq. 6 apply without change. During training, each dataset action is encoded as $\boldsymbol{e}(a_{i,t}) \in \{0, 1\}^{A_i}$. At execution, the Euler state $\boldsymbol{a}_{i,\lambda}$ is a relaxed vector in $\mathbb{R}^{A_i}$, $\hat{\boldsymbol{a}}_{i,1}$ is clipped to $[0, 1]^{A_i}$ to match the one-hot training support, and a legal-set $\arg\max$ recovers $a_i = \arg\max_{k \in \mathcal{K}_i} [\boldsymbol{a}_{i,1}]_k$, where $\mathcal{K}_i$ is agent $i$'s legal action set.

## 5 Experiments

We evaluate whether team-supervised local utility gradients yield coordinated improvement under strictly decentralized execution, and whether obtaining that improvement at test time remains competitive with prior offline MARL methods across diverse benchmarks. The following sections examine, in turn, benchmark performance (Section 5.2), coordination relative to independent and centralized guidance (Section 5.3), the effect of guidance on a matched frozen flow (Section 5.4), training and inference cost (Section 5.5), and goal-conditioned generalization (Section 5.6).

### 5.1 Benchmarks, datasets and baselines

**Benchmarks.** We evaluate CQF on continuous-control MA-MuJoCo (Peng et al., 2021) and MPE (Lowe et al., 2017), discrete-control SMACv1 (Samvelyan et al., 2019), and the goal-conditioned continuous-control benchmark MangoBench (Wang et al., 2026). We use the MA-MuJoCo datasets released by (Wang et al., 2023b) (Expert, Medium-Expert, Medium, and Medium-Replay), the MPE datasets from (Pan et al., 2022) (Expert, Medium, Medium-Replay, and Random), and the SMACv1 datasets from (Formanek et al., 2023) (Good, Medium, and Poor for each map). For MangoBench, we evaluate the $2 \times 4$ AntMaze partition on Medium-Navigate, Large-Navigate, and Medium-Stitch.

**Baselines.** Our baselines cover Gaussian, diffusion, and flow policies. On MA-MuJoCo and SMACv1, we include MA-TD3+BC, MA-CQL, ICQ, OMAR, OMIGA, MADiff, MAC-Flow, and

Table 1: **Performance evaluation for MA-MuJoCo.** Our results are averaged over 3 seeds, and we report the standard deviations after the ± sign. We highlight the best performance in **bold** and the second best in underlined. For readability, we use the acronyms M-E and M-R for Medium-Expert and Medium-Replay, respectively.

| | Scenarios | Dataset | Extension of offline SARL | | Offline MARL | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | MATD3BC | MACQL | ICQ | OMAR | OMIGA | MADiff | MAC-Flow | CQF |
| MA-MuJoCo | HalfCheetah | Expert | 4401.6 ±169.1 | 4589.5 ±98.5 | 2955.9 ±459.2 | −206.7 ±161.1 | 3383.6 ±552.7 | **4711.4** ±213.6 | 4650.0 ±271.6 | 4461.9 ±22.0 |
| | | Medium | 2620.8 ±69.9 | 3189.4 ±306.9 | 2549.3 ±96.3 | −265.7 ±147.0 | 3608.1 ±237.4 | 2650.0±365.4 | **4358.5** ±369.2 | 3721.2 ±26.2 |
| | | M-R | 3528.9 ±120.9 | 3500.7 ±293.9 | 1922.4 ±612.9 | −235.4 ±154.9 | 2504.7 ±83.5 | 2830.5±292.8 | 3030.2 ±436.8 | **3765.3** ±4.7 |
| | | M-E | 3518.1 ±381.0 | 4738.2 ±181.1 | 2834.0 ±420.3 | −253.8 ±63.9 | 2948.5 ±518.9 | 4410.9±836.8 | **5139.9** ±84.1 | 4589.5 ±28.3 |
| | Hopper | Expert | 3309.9 ±4.5 | 3359.1 ±513.8 | 754.7 ±806.3 | 2.4 ±1.5 | 859.6 ±709.5 | 2853.3±593.8 | **3592.1** ±8.9 | 3391.7 ±224.9 |
| | | Medium | 870.4 ±156.7 | 901.3 ±199.9 | 501.8 ±14.0 | 21.3 ±24.9 | 1189.3 ±544.3 | 1436.8±449.5 | 1023.5 ±253.0 | **3211.8** ±5.1 |
| | | M-R | 269.7 ±41.8 | 31.4 ±15.2 | 195.4 ±103.6 | 3.3 ±3.2 | 774.2 ±494.3 | 936.1 ±574.0 | 1166.3 ±451.9 | **2012.4** ±314.5 |
| | | M-E | 2904.3 ±477.4 | 2751.8 ±123.3 | 355.4 ±373.9 | 1.4 ±0.9 | 709.0 ±595.7 | 2810.4 ±723.2 | 2988.3 ±480.2 | **3001.9** ±241.5 |
| | Ant | Expert | 2046.9 ±17.1 | 2082.4 ±21.7 | 2050.0 ±11.9 | 312.5 ±297.5 | 2055.5 ±1.6 | 2060.0±10.3 | 2060.2 ±20.0 | **2123.2** ±10.1 |
| | | Medium | 1422.6 ±21.1 | 1033.9 ±66.4 | 1412.4 ±10.9 | −1710.0 ±1589.0 | 1418.4 ±5.4 | 1428.4 ±14.7 | 1432.4 ±17.8 | **1532.6** ±18.4 |
| | | M-R | 995.2 ±52.8 | 434.6 ±108.3 | 1016.7 ±53.5 | −2014.2 ±844.7 | 1105.1 ±88.9 | 1294.5 ±360.2 | 1498.4 ±20.3 | **1525.0** ±8.4 |
| | | M-E | 1636.1 ±96.0 | 1800.2 ±21.5 | 1590.2 ±85.6 | −2992.8 ±7.0 | 1720.3 ±110.6 | 1740.2±158.9 | 2053.3 ±20.4 | **2096.8** ±8.3 |
| | Average rewards | | 2293.7 | 2367.7 | 1511.5 | −611.5 | 1856.4 | 2430.21 | 2749.4 | **2952.8** |

Table 2: **Performance evaluation for MPE.** Entries are D4RL-normalized scores following OMAR and MADiff, $100 \times (S - 159.8)/(516.8 - 159.8)$. Results are reported following the conventions of Table 1.

| | Scenarios | Dataset | Extension of offline SARL | | Offline MARL | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | MATD3BC | MACQL | ICQ | OMAR | OMIGA | MADiff | CQF |
| MPE | Spread | Expert | 108.3 ±3.3 | 98.2 ±5.2 | **114.9** ±2.6 | 104.0 ±3.4 | 80.8 ±13.8 | 95.0 ±5.3 | 96.4 ±0.6 |
| | | Medium | 29.3 ±4.8 | 34.1 ±7.2 | 47.9 ±18.9 | 29.3 ±5.5 | 30.1 ±16.9 | **64.9** ±7.7 | 55.0 ±6.8 |
| | | M-R | 15.4 ±5.6 | 20.0 ±8.4 | **37.9** ±12.3 | 13.6 ±5.7 | 5.4 ±11.0 | 30.3 ±2.5 | 33.1 ±1.7 |
| | | Random | 9.8 ±4.9 | 24.0 ±9.8 | 34.4 ±5.3 | 6.3 ±3.5 | −3.8 ±12.3 | 6.9 ±3.1 | **58.3** ±9.4 |
| | Average scores | | 40.7 | 44.1 | 58.8 | 38.3 | 28.1 | 49.2 | **60.7** |

Table 3: **Performance evaluation for SMACv1**. Results are reported following the conventions of Table 1.

| | Scenarios | Dataset | Gaussian policies | | | Diffusion policies | | | Flow policies | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | BC | MABCQ | MACQL | Diffusion BC | MADiff | DoF | Flow BC | MAC-Flow | CoFlow | CQF |
| SMACv1 | 3m | Good | 16.0 ±1.0 | 3.7 ±1.1 | 19.1 ±0.1 | 19.5 ±0.5 | 19.3 ±0.5 | 19.8 ±0.2 | **20.0** ±0.0 | 19.8 ±0.2 | 17.3 ±5.5 | **20.0** ±1.9 |
| | | Medium | 8.2 ±0.8 | 4.0 ±1.0 | 13.7 ±0.3 | 13.3 ±0.7 | 16.4 ±2.6 | 18.6 ±1.2 | 14.7 ±1.5 | 18.0 ±3.2 | 12.8 ±5.9 | **18.8** ±1.2 |
| | | Poor | 4.4 ±0.1 | 3.4 ±1.0 | 4.2 ±0.1 | 4.2 ±0.2 | 10.3 ±6.1 | 10.9 ±1.1 | 4.5 ±0.1 | 10.6 ±2.2 | 10.4 ±6.3 | **12.2** ±2.7 |
| | 8m | Good | 16.7 ±0.4 | 4.8 ±0.6 | 18.9 ±0.9 | 19.4 ±0.5 | 18.9 ±1.1 | 19.6 ±0.3 | 19.5 ±0.2 | 19.7 ±0.3 | **20.0** ±0.0 | 19.8 ±0.4 |
| | | Medium | 10.7 ±0.5 | 5.6 ±0.6 | 15.5 ±1.5 | 18.6 ±0.6 | 16.8 ±1.6 | 18.6 ±0.8 | 18.2 ±0.8 | 19.4 ±0.6 | 17.2 ±4.3 | **20.0** ±0.0 |
| | | Poor | 5.3 ±0.1 | 3.6 ±0.8 | 7.5 ±1.0 | 4.8 ±0.2 | 9.8 ±0.9 | **12.0** ±1.2 | 4.9 ±0.1 | 11.5 ±0.8 | 6.4 ±0.7 | 8.0 ±0.4 |
| | 2s3z | Good | 18.2 ±0.4 | 7.7 ±0.9 | 17.4 ±0.3 | 18.0 ±1.0 | 15.9 ±1.2 | 18.5 ±0.8 | 19.5 ±0.1 | 19.5 ±0.5 | 19.2 ±2.5 | **19.9** ±0.3 |
| | | Medium | 12.3 ±0.7 | 7.6 ±0.7 | 15.6 ±0.4 | 13.4 ±1.4 | 15.6 ±0.3 | 18.1 ±0.9 | 15.1 ±2.0 | 17.6 ±0.6 | **18.8** ±2.6 | 18.2 ±0.0 |
| | | Poor | 6.7 ±0.3 | 6.6 ±0.2 | 8.4 ±0.8 | 6.2 ±1.2 | 8.5 ±1.3 | 10.0 ±1.1 | 6.9 ±0.8 | 8.5 ±0.6 | 10.1 ±1.9 | **10.7** ±1.3 |
| | 5m_vs_6m | Good | 15.8 ±3.6 | 2.4 ±0.4 | 16.2 ±1.6 | 16.8 ±2.3 | 16.5 ±2.8 | 17.7 ±1.1 | 14.7 ±2.1 | 18.6 ±3.5 | 14.0 ±5.0 | **18.8** ±0.4 |
| | | Medium | 12.4 ±0.9 | 3.8 ±0.5 | 15.1 ±2.9 | 12.5 ±2.1 | 15.2 ±2.6 | 16.2 ±0.9 | 12.8 ±0.8 | 15.6 ±1.3 | 15.2 ±4.9 | **16.7** ±1.2 |
| | | Poor | 7.5 ±0.2 | 3.3 ±0.5 | 10.5 ±3.1 | 8.0 ±1.0 | 8.9 ±1.3 | 10.8 ±0.3 | 7.7 ±0.8 | 9.8 ±2.1 | **10.9** ±3.1 | 10.4 ±0.7 |
| | Average rewards | | 11.2 | 4.7 | 13.5 | 12.9 | 14.3 | 15.9 | 13.2 | 15.7 | 14.4 | **16.1** |

CoFlow (Yang et al., 2021; Pan et al., 2022; Wang et al., 2023b; Zhu et al., 2024; Lee et al., 2026; Zou et al., 2026). On MPE, we include MA-TD3+BC, MA-CQL, ICQ, OMAR, OMIGA, and MADiff, but omit MAC-Flow and CoFlow from the main table because their published Spread scores are not comparable under a single evaluation protocol . For more difficult goal-conditioned tasks, we compare our goal-conditioned CQF with the goal-conditioned variants of MADiff (Zhu et al., 2024) and MAC-Flow (Lee et al., 2026). The details of these variants and the comparison settings are provided in the Appendix H.

### 5.2 Comparison with prior offline MARL methods

Tables 1, 2, and 3 summarize the comparison of CQF with prior offline MARL methods. On MA-MuJoCo, CQF attains the highest average reward (2952.8 vs. 2749.4 for MAC-Flow), ranking first on 8 of 12 datasets and second on another 2 datasets. The same holds on MPE (60.7 vs. 58.8 for ICQ) and SMACv1 (16.1 vs. 15.9 for DoF; best or tied-best on 8 of 12). CQF remains competitive on Expert data and obtains its largest gains on mixed and suboptimal splits (Medium, Medium-Replay, and Random; e.g., Hopper Medium-Replay and MPE Random), which Section 5.4 identifies as the test-time improvement regime. Thus, the standard benchmark results support the effectiveness of CQF as a general offline MARL solution. Compared with prior methods, CQF delivers state-of-the-art or near state-of-the-art performance while keeping the policy and the critic decoupled during training, and retaining centralized coordination information for decentralized guidance at deployment. Section 5.3 empirically verifies this coordination.

### 5.3 COORDINATION VIA TEAM-ALIGNED LOCAL GRADIENTS

Proposition 1 claims that team-supervised local gradients are simultaneously locally queryable and first-order aligned with the learned team critic. We empirically verify this claim in two steps. First, independently trained local critics remain equally queryable under the same decentralized flow, yet they lose team return, so local queryability without team alignment is not sufficient. Second, we measure whether team-aligned local gradients remain competitive with a centralized critic and flow. Whether the return gain over independent local guidance comes from better coordination, rather than from more aggressive local improvement, is diagnosed on MPE in Appendix E.3.

Table 4: **Independent, team-aligned and centralized guidance on MPE.**

| Dataset | MA-QGF | CQF | Cent.† |
|---|---|---|---|
| Expert | 88.6 ±16.0 | **96.4** ±0.6 | 112.6 ±4.0 |
| Medium | 38.7 ±2.3 | **55.0** ±6.8 | 45.9 ±2.3 |
| M-Replay | 13.8 ±0.4 | **33.1** ±1.7 | 33.5 ±1.9 |
| Random | 48.5 ±1.2 | **58.3** ±9.4 | 33.6 ±3.2 |
| Avg. | 47.4 | **60.7** | 56.4 |

Table 4 compares three ablations of CQF on MPE Simple Spread. MA-QGF replaces the factorized team critic by independently trained per-agent IQL critics, which is a fully decentralized variant of CQF. The centralized ablation replaces both by a joint flow on the global state and a centralized critic, which is a fully centralized variant of CQF. It requires the global state and the joint action at execution, is not executable under a Dec-POMDP, and is reported only as a reference. Appendix E.1 repeats the comparisons on MA-MuJoCo. MA-QGF is as locally queryable as CQF, so the gap in Table 4 is the cost of missing team alignment: CQF is the best deployable ablation on every split and raises the average from 47.4 to 60.7. The centralized variant is team-aligned but not locally queryable. It exceeds CQF only on Expert and is otherwise matched or worse, including on average (56.4 versus 60.7). Independent local guidance is therefore not sufficient, and team-aligned local gradients remain competitive with a centralized critic and flow. Proposition 1 ensures that assembled local guidance cannot first-order oppose the learned $Q_{\text{tot}}$; Table 4 shows that this decentralized interface does not incur a systematic gap to a non-deployable centralized variant.

### 5.4 TEST-TIME GUIDANCE OF A FROZEN BEHAVIOR FLOW

This section evaluates whether utility-gradient guidance improves a matched frozen behavior flow, and whether that improvement is concentrated on mixed and suboptimal data. Because the guidance strength $\beta$ enters only Eq. 6, the same frozen flow and critic can be re-evaluated on a shared grid without retraining. Figure 2 shows how performance varies with the guidance strength $\beta$ at 200k updates. On every split the return first rises above the unguided flow ($\beta = 0$) and then falls as $\beta$ becomes large. The gain is small on Expert (2044.7 to 2123.2), where imitation already yields a strong policy, and largest on Medium-Replay (1023.0 to 1525.0). Appendix F shows the same pattern on all datasets: the benchmark average rises from 1718.3 to 2952.8 on MA-MuJoCo, from 28.2 to 60.7 on MPE, and from 13.0 to 16.1 on SMACv1.

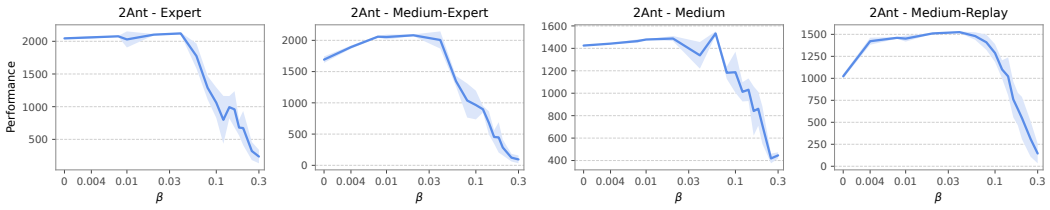


Figure 2: **Guidance-weight sweep on 2-Ant.** Mean episode return versus $\beta$ at 200k updates, with the same frozen flow and critic. Shaded regions are the standard deviation across 3 training seeds.

### 5.5 TRAINING AND INFERENCE EFFICIENCY

We next measure whether separating behavior modeling from reward optimization also reduces computational cost. Figure 3(a) shows BC-like training time: 0.28 hours on discrete and 0.23 hours on continuous tasks, 7.4× faster than MAC-Flow and 200–237× faster than diffusion planners on SMACv1, and 4.4–21× faster than MAC-Flow, OMAR, and MACQL on continuous control. These savings follow from the separated objectives in Section 4.1: flow matching and team-level IQL require neither teacher-student distillation nor Q-maximization through the generator, and avoid

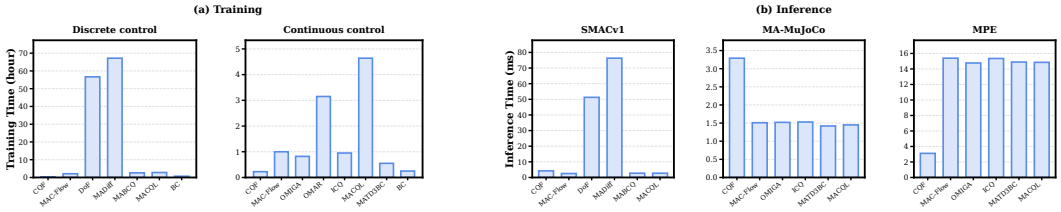


Figure 3: **Training and inference efficiency.** (a) Wall-clock training time averaged over discrete (SMACv1) and continuous (MA-MuJoCo, MPE) control. (b) Per-step inference time on SMACv1, MA-MuJoCo, and MPE. Lower is better in all panels.

backpropagation through long denoising or trajectory models. Because $\beta$ is applied only at deployment, it can be swept without retraining (Figure 2). The same training-time decoupling also reduces hyperparameter tuning. If the flow and critic have $M$ and $N$ candidate configurations, decoupled tuning requires $\mathcal{O}(M+N)$ rather than $\mathcal{O}(MN)$ coupled runs. Figure 3(b) shows that our inference stays in the few-millisecond regime (4.21 ms on SMACv1, 3.29 ms on MA-MuJoCo, 3.11 ms on MPE), 12–18$\times$ faster than diffusion planners.

### 5.6 Goal-conditioned multi-goal generalization

To move beyond the fixed-task evaluation in Section 5.2, we test whether CQF extends to goal-conditioned offline MARL, where agents must coordinate toward goals specified only at evaluation time. We evaluate the $2 \times 4$ AntMaze partition on Medium-Navigate, Large-Navigate, and Medium-Stitch. We compare GCCQF with GCMadiff, the goal-conditioned variant of MADiff (Zhu et al., 2024), and GCMAC-Flow, the goal-conditioned variant of MAC-Flow (Lee et al., 2026).

As shown in Table 5, GCCQF is consistently the best-performing method compared with prior diffusion-based and flow-based methods. On Large-Navigate, the gap over GCMADiff and GCMAC-Flow remains larger as the maze scale grows, indicating that our method scales well on long-horizon tasks. The lead on Medium-Stitch indicates that CQF can stitch short-horizon fragments toward a requested goal.

Table 5: **MangoBench success rates (%).** Mean $\pm$ std over 3 seeds and five test-time goals. Best in **bold**.

| Method | Medium Navigate | Large Navigate | Medium Stitch |
|---|---|---|---|
| GCMadiff | 63.3 $\pm$6.4 | 24.0 $\pm$4.0 | 22.0 $\pm$7.2 |
| GCMAC-Flow | 60.3 $\pm$1.4 | 30.3 $\pm$2.2 | 37.5 $\pm$2.4 |
| GCCQF | **95.3** $\pm$0.6 | **89.0** $\pm$2.0 | **84.0** $\pm$5.6 |

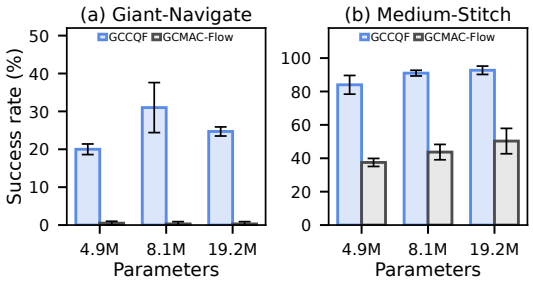


Figure 4: **Scaling model size of GCCQF.**

We further evaluate the scalability of GCCQF with model size. We retrain GCCQF and GCMAC-Flow at 8.1M and 19.2M on Giant-Navigate and Medium-Stitch. As shown in Figure 4, on Giant-Navigate, scaling GCCQF from 4.9M to 8.1M raises the success rate by 55% relative, while GCMAC-Flow remains near zero at every size. On Medium-Stitch, GCCQF improves monotonically with capacity, remaining well ahead of GCMAC-Flow.

## 6 Conclusion

We present CQF, to our knowledge the first test-time guidance framework for offline MARL, where guidance must come from a critic that is locally queryable yet team-aligned. Monotonic factorization of a team-level critic objective delivers both: simultaneous local guidance steps form a first-order ascent direction of the learned team value, and a one-hot relaxation covers discrete control. Across MA-MuJoCo, MPE, SMACv1, and MangoBench, CQF attains the best average scores among Gaussian, diffusion, and flow baselines, with the largest gains on mixed and suboptimal data, and team-supervised utilities outperform independently trained local critics under the same frozen flow. Coordination can therefore be injected through local value gradients while keeping execution decentralized and behavior modeling separate from reward optimization, which makes test-time guidance practical for offline MARL.

**AI use statement.** In this work, we used generative AI tools for assisting with translation. We have not used generative AI tools to generate synthetic datasets, develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proofs, assist in writing proofs, propose or refine hypotheses, design or provide feedback on research methodology or experiments, implement methods, clean or reformat datasets, support qualitative or thematic data analysis, or interpret results, and the remaining required disclosure tasks are not applicable to this work. Additionally, we used generative AI tools to edit the research paper for readability, draft or revise small portions of manuscript text, suggest wording and structure, format references, and assist with software code editing and debugging. We have reviewed all AI-assisted work; all revised text was checked by the authors for accuracy, consistency with the experiments, and potential misrepresentation, and any AI-assisted code was manually inspected and tested by the authors. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

## REFERENCES

Daniel S Bernstein, Robert Givan, Neil Immerman, and Shlomo Zilberstein. The complexity of decentralized control of Markov decision processes. *Mathematics of operations research*, 27(4): 819–840, 2002.

Zihan Ding and Chi Jin. Consistency models as a rich and efficient policy class for reinforcement learning. In *International Conference on Learning Representations (ICLR)*, volume 2024, pp. 53047–53066, 2024.

Nicolas Espinosa-Dice, Yiyi Zhang, Yiding Chen, Bradley Guo, Owen Oertell, Gokul Swamy, Kiante Brantley, and Wen Sun. Scaling offline rl via efficient and expressive shortcut models. *Advances in Neural Information Processing Systems (NeurIPS)*, 38:2376–2414, 2026.

Claude Formanek, Asad Jeewa, Jonathan Shock, and Arnu Pretorius. Off-the-grid MARL: Datasets with baselines for offline multi-agent reinforcement learning. In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2023.

Claude Formanek, Omayma Mahjoub, Louay Ben Nessir, Sasha Abramowitz, Ruan de Kock, Wiem Khlifi, Simon Du Toit, Felix Chalumeau, Daniel Rajaonarivonivelomanantsoa, Arnol Fokam, et al. Oryx: a scalable sequence model for many-agent coordination in offline MARL. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Kevin Frans, Seohong Park, Pieter Abbeel, and Sergey Levine. Diffusion guidance is a controllable policy improvement operator. *arXiv preprint arXiv:2505.23458*, 2025.

Scott Fujimoto and Shixiang Shane Gu. A minimalist approach to offline reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, pp. 20132–20145, 2021.

Philippe Hansen-Estruch, Ilya Kostrikov, Michael Janner, Jakub Grudzien Kuba, and Sergey Levine. IDQL: Implicit Q-learning as an actor-critic method with diffusion policies. *arXiv preprint arXiv:2304.10573*, 2023.

Longxiang He, Li Shen, Linrui Zhang, Junbo Tan, and Xueqian Wang. Diffcps: Diffusion model based constrained policy search for offline reinforcement learning. *arXiv preprint arXiv:2310.05333*, 2023.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems (NeurIPS)*, 33:6840–6851, 2020.

Yejun Jang, Hong Chul Nam, Jeong Min Park, Gimin Bae, and Hyun Kwon. Q-guided flow q-learning. In *CoRL 2025 Workshop RemembeRL*, 2025.

Michael Janner, Yilun Du, Joshua B Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. *arXiv preprint arXiv:2205.09991*, 2022.

Yue Jin and Giovanni Montana. Partial action replacement: Tackling distribution shift in offline MARL. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2026.

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit Q-learning. In *International Conference on Learning Representations (ICLR)*, 2022.

Aviral Kumar, Xue Bin Peng, and Sergey Levine. Reward-conditioned policies. *arXiv preprint arXiv:1912.13465*, 2019.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative Q-learning for offline reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pp. 1179–1191, 2020.

Andrei Kurenkov, Ajay Mandlekar, Roberto Martin-Martin, Silvio Savarese, and Animesh Garg. Multi-agent decision transformer. *arXiv preprint arXiv:2203.13691*, 2022.

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

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

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 *The Thirteenth International Conference on Learning Representations (ICLR)*, 2025.

Qiyang Li, Zhiyuan Paul Zhou, and Sergey Levine. Reinforcement learning with action chunking. *Advances in Neural Information Processing Systems (NeurIPS)*, 38:55518–55553, 2026a.

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 (TMLR)*, 2026b. ISSN 2835-8856.

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 (AAMAS)*, pp. 3767–3776, 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 (ICLR)*, 2023.

Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. *Advances in Neural Information Processing Systems (NeurIPS)*, 30, 2017.

Max Sobol Mark, Tian Gao, Georgia Gabriela Sampaio, Mohan Kumar Srirama, Archit Sharma, Chelsea Finn, and Aviral Kumar. Policy agnostic rl: Offline rl and online rl fine-tuning of any class and backbone. *arXiv preprint arXiv:2412.06685*, 2024.

Takayuki Osa and Tatsuya Harada. Robustifying a policy in multi-agent rl with diverse cooperative behaviors and adversarial style sampling for assistive tasks. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 15158–15164, 2024.

Ling Pan, Longbo Huang, Tengyu Ma, and Huazhe Xu. Plan better amid conservatism: Offline multi-agent reinforcement learning with actor rectification. In *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 17221–17237, 2022.

Teng Pang, Zhiqiang Dong, Yan Zhang, Rongjian Xu, Guoqiang Wu, and Yilong Yin. Value-guidance MeanFlow for offline multi-agent reinforcement learning. *arXiv preprint arXiv:2604.08174*, 2026.

Seohong Park, Qiyang Li, and Sergey Levine. Flow Q-learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2025.

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. *Advances in Neural Information Processing Systems (NeurIPS)*, 34:12208–12221, 2021.

Dan Qiao, Wenhao Li, Shanchao Yang, Hongyuan Zha, and Baoxiang Wang. Offline multi-agent reinforcement learning via sequential score decomposition. In *Forty-third International Conference on Machine Learning (ICML)*, 2026.

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

Mikayel Samvelyan, Tabish Rashid, Christian Schroeder De Witt, Gregory Farquhar, Nantas Nardelli, Tim GJ Rudner, Chia-Man Hung, Philip HS Torr, Jakob Foerster, and Shimon Whiteson. The starcraft multi-agent challenge. *arXiv preprint arXiv:1902.04043*, 2019.

Jianzhun Shao, Yun Qu, Chen Chen, Hongchang Zhang, and Xiangyang Ji. Counterfactual conservative q learning for offline multi-agent reinforcement learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 36:77290–77312, 2023.

Kyunghwan Son, Daewoo Kim, Wan Ju Kang, David Earl Hostallero, and Yung Yi. QTRAN: Learning to factorize with transformation for cooperative multi-agent reinforcement learning. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, pp. 5887–5896, 2019.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations (ICLR)*, 2021.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems (NeurIPS)*, 33:3008–3021, 2020.

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. *arXiv preprint arXiv:1706.05296*, 2017.

Ming Tan. Multi-agent reinforcement learning: Independent vs. cooperative agents. In *Proceedings of the tenth International Conference on Machine Learning (ICML)*, pp. 330–337, 1993.

Weizheng Wang, Le Mao, Ruiqi Wang, and Byung-Cheol Min. Multi-robot cooperative socially-aware navigation using multi-agent reinforcement learning. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 12353–12360, 2024a.

Xiangsen Wang, Haoran Xu, Yinan Zheng, and Xianyuan Zhan. Offline multi-agent reinforcement learning with implicit global-to-local value regularization. *Advances in Neural Information Processing Systems (NeurIPS)*, 36:52413–52429, 2023a.

Xiangsen Wang, Haoran Xu, Yinan Zheng, and Xianyuan Zhan. Offline multi-agent reinforcement learning with implicit global-to-local value regularization. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, pp. 52413–52429, 2023b.

Yi Wang, Ningze Zhong, Zhiheng Fu, Longguang Wang, Ye Zhang, and Yulan Guo. MangoBench: A benchmark for multi-agent goal-conditioned offline reinforcement learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6219–6228, 2026.

Yinuo Wang, Likun Wang, Yuxuan Jiang, Wenjun Zou, Tong Liu, Xujie Song, Wenxuan Wang, Liming Xiao, Jiang Wu, Jingliang Duan, et al. Diffusion actor-critic with entropy regulator. *Advances in Neural Information Processing Systems (NeurIPS)*, 37:54183–54204, 2024b.

Zhendong Wang, Jonathan J Hunt, and Mingyuan Zhou. Diffusion policies as an expressive policy class for offline reinforcement learning. In *The Eleventh International Conference on Learning Representations (ICLR)*, 2023c.

Haoran Xu, Kaiwen Hu, Somayeh Sojoudi, and Amy Zhang. Reinforcement learning via value gradient flow. In *The Fourteenth International Conference on Learning Representations (ICLR)*, 2026.

Yiqin Yang, Xiaoteng Ma, Chenghao Li, Zewu Zheng, Qiyuan Zhang, Gao Huang, Jun Yang, and Qianchuan Zhao. Believe what you see: Implicit constraint approach for offline multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, pp. 10299–10312, 2021.

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

Kaiqing Zhang, Zhuoran Yang, and Tamer Başar. Multi-agent reinforcement learning: A selective overview of theories and algorithms. *Handbook of Reinforcement Learning and Control*, pp. 321–384, 2021.

Ruoqi Zhang, Ziwei Luo, Jens Sjölund, Thomas B Schön, and Per Mattsson. Entropy-regularized diffusion policy with q-ensembles for offline reinforcement learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 37:98871–98897, 2024.

Zhiyuan Zhou, Andy Peng, Charles Xu, Qiyang Li, Tobias Springenberg, Kevin Frans, and Sergey Levine. Test-time gradient guidance of flow policies in reinforcement learning. *arXiv preprint arXiv:2606.11087*, 2026.

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 (NeurIPS)*, volume 37, pp. 4177–4206, 2024.

Guowei Zou, Haitao Wang, Beiwen Zhang, Boning Zhang, and Hejun Wu. CoFlow: Coordinated few-step flow for offline multi-agent decision making. *arXiv preprint arXiv:2605.01457*, 2026.

# APPENDIX

**A Pseudocode of CQF** **15**

**B Benchmark Details** **15**

B.1 MA-MuJoCo . . . . . 15

B.2 MPE . . . . . 15

B.3 SMACv1 . . . . . 16

B.4 MangoBench . . . . . 16

**C First-Order Alignment of Guidance and the Composite** **16**

C.1 Coordinated first-order ascent (Proposition 1) . . . . . 16

C.2 The team-aligned guidance corrects the composite direction . . . . . 17

**D Ablations on Discrete Relaxation** **18**

**E Additional Results for Coordination Diagnostics** **19**

E.1 Coordination Comparison on MA-MuJoCo . . . . . 20

E.2 Learning Curves . . . . . 20

E.3 Coordination Analysis on MPE . . . . . 22

**F Test-Time Guidance and Guidance-Weight Sweeps** **23**

F.1 Test-Time Guidance Comparisons . . . . . 24

F.2 Guidance Weights Sweep of MA-MuJoCo . . . . . 25

**G Ablations on Local Flow and Centralized Flow** **26**

**H Goal-Conditioned Offline MARL** **27**

H.1 GCCQF . . . . . 27

H.2 GCMAC-Flow . . . . . 28

H.3 GCMADiff . . . . . 29

**I Implementation Details** **29**

**J Limitations** **33**

# A PSEUDOCODE OF CQF

Algorithm 1 records the two independent training objectives of Sections 4.1–4.2 and the decentralized Euler update of Section 4.3. Discrete control uses the one-hot relaxation of Section 4.4: $\hat{\boldsymbol{a}}_{i,1}$ is clipped to $[0,1]^{A_i}$ before the utility gradient is taken, and a legal-set $\arg\max$ recovers the executed action.

**Algorithm 1** Coordinated Q-Gradient Flow (CQF)

**Require:** Offline dataset $\mathcal{D}$; Euler steps $L$; guidance weight $\beta$

Initialize $\boldsymbol{v}_\theta$, $Q_{\phi,i}$, $V_{\psi,i}$, $M_\phi^Q$, $M_\psi^V$

**Centralized training** ▷ flow and critic updated independently

**for** each minibatch from $\mathcal{D}$ **do**

  Update $\theta$ by Eq. 4; update $\phi, \psi$ by $\mathcal{L}_Q$, $\mathcal{L}_V$; Polyak-update $\bar{\phi}$

**end for**

**Decentralized execution** ▷ no mixer, global state, or communication

**for all** agent $i$ in parallel **do**

  $\boldsymbol{a}_{i,0} \leftarrow \boldsymbol{\epsilon}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

  **for** $\ell = 0, \dots, L-1$ **do**

    $\lambda \leftarrow \ell/L$, $\Delta\lambda \leftarrow 1/L$, $\boldsymbol{v}_i \leftarrow \boldsymbol{v}_\theta(\boldsymbol{o}_i^{\mathrm{id}}, \boldsymbol{a}_{i,\lambda}, \lambda)$

    $\hat{\boldsymbol{a}}_{i,1} \leftarrow \boldsymbol{a}_{i,\lambda} + (1-\lambda)\boldsymbol{v}_i$, $\boldsymbol{g}_i \leftarrow \nabla_{\hat{\boldsymbol{a}}_{i,1}} Q_{\bar{\phi},i}(\boldsymbol{o}_i^{\mathrm{id}}, \hat{\boldsymbol{a}}_{i,1})$ ▷ discrete: clip $\hat{\boldsymbol{a}}_{i,1}$ to $[0,1]^{A_i}$

    $\boldsymbol{a}_{i,\lambda+\Delta\lambda} \leftarrow \boldsymbol{a}_{i,\lambda} + \Delta\lambda(\boldsymbol{v}_i + \beta \boldsymbol{g}_i)$ ▷ Eq. 6

  **end for**

  Execute $\boldsymbol{a}_i \leftarrow \boldsymbol{a}_{i,1}$ ▷ discrete: legal $\arg\max$; continuous: clip to $[-1,1]$

**end for**

# B BENCHMARK DETAILS

This section describes the environments and offline data used in our evaluation. All experiments train only from the corresponding fixed data and evaluate decentralized policies in the original simulator. Dataset-quality names refer to the behavior policy that generated the trajectories.

## B.1 MA-MUJOCO

MA-MuJoCo is the multi-agent extension of the MuJoCo locomotion suite used in FACMAC and subsequent offline-MARL work (Peng et al., 2021). A single physical robot (Ant, HalfCheetah, or Hopper in our experiments) is divided into agents by assigning disjoint groups of joints to each agent. The agents therefore have separate continuous action vectors, while the body dynamics, termination condition, and locomotion reward remain coupled. This construction tests coordination through a shared return without giving the execution policy a centralized action or critic.

We use the public MA-MuJoCo data released with OMIGA (Wang et al., 2023b). Each transition contains per-agent observations and actions together with reward, termination, and episode-boundary information. The four quality levels are *Expert*, high-performing trajectories; *Medium-Expert*, a mixture of expert and medium-quality experience; *Medium*, data from a partially trained controller; and *Medium-Replay*, the replay stream accumulated while training that controller. Medium-Replay and Medium-Expert consequently expose broader and more heterogeneous behavior support than Expert. Our evaluation uses the 6-agent HalfCheetah, 3-agent Hopper, and 2-agent Ant partitions shown in the tables, with the released scaling and episode-return convention.

## B.2 MPE

The Multi-Agent Particle Environment (MPE) is a two-dimensional particle simulator for mixed cooperative–competitive tasks (Lowe et al., 2017). We use cooperative SIMPLE SPREAD, where three controllable particles cover a set of landmarks. The reward combines landmark coverage with a collision penalty, so agents must select distinct targets while accounting for one another. Actions are continuous two-dimensional movement controls and execution uses each particle's local observation.

Following the OMAR data protocol (Pan et al., 2022), we evaluate Expert, Medium, Medium-Replay, and Random trajectories. Expert and Medium are collected at different training stages; Medium-Replay is the replay stream collected during medium-policy training; and Random is collected from randomly initialized behavior. These latter splits test recovery from poor coordination and limited coverage of successful landmark assignments. We use the released observations, actions, rewards, next observations, and terminal indicators without online demonstrations.

### B.3 SMACv1

SMACv1 is a StarCraft II micromanagement benchmark in which allied units fight scripted enemies on a fixed map (Samvelyan et al., 2019). At each decision point an agent selects a discrete legal action, such as movement, stop, or attacking an available enemy. Agents observe local unit features and nearby entities, while the global state is available only to centralized training. The benchmark thus combines partial observability, action masking, heterogeneous roles, and delayed team success.

We use the SMACOffline/Off-the-Grid datasets released by Formanek et al. (2023). For each map, trajectories are partitioned by behavior quality into *Good*, *Medium*, and *Poor*; these labels describe the competence of the data-collecting team, not per-agent rewards. Our experiments cover homogeneous 3m and 8m and heterogeneous 2s3z. The data preserve legal-action masks, local observations, global states used by critics, rewards, and episode termination, so every method is evaluated under identical availability constraints.

### B.4 MangoBench

MangoBench is a fully cooperative, multi-goal benchmark for goal-conditioned offline MARL (Wang et al., 2026). It extends AntMaze-style locomotion to multiple controllers: one ant is partitioned among agents, each agent receives a local observation and goal component, and the team must reach a requested goal under sparse success feedback. The benchmark therefore combines long-horizon continuous control, partial observability, goal conditioning, and evaluation-goal shift.

We use the $2 \times 4$ AntMaze partition, in which two agents control four disjoint joint dimensions each. *Medium-Navigate* and *Large-Navigate* evaluate navigation to held-out goals at increasing maze scale, whereas *Medium-Stitch* tests composing trajectory fragments toward a requested goal. We report the benchmark success rate (the percentage of episodes reaching the goal), rather than dense locomotion return, using the released goal-conditioned transitions and their original goal, observation, action, and termination fields. The evaluation goals are disjoint from the goals used for actor or value hindsight relabeling and are never supplied to training.

## C First-Order Alignment of Guidance and the Composite

This appendix checks two facts on the same rollouts. Appendix C.1 tests the guidance claim of Proposition 1 on the added term $\boldsymbol{g}$. Appendix C.2 then measures how that term changes the composite $\boldsymbol{v} + \beta\boldsymbol{g}$ that Eq. 6 actually applies, at the clean-action estimate where the critic is queried. Both analyses reuse the frozen checkpoints of Tables 4 and 10 without retraining, each checkpoint's domain-specific $\beta$, and the default 10-step Euler sampler, storing quantities at every recorded solver step ($\lambda = k/10$; MA-MuJoCo subsampled every 10 planning calls, MPE every call).

### C.1 Coordinated first-order ascent (Proposition 1)

The guidance that CQF adds at execution is the local term $\boldsymbol{g}$, not the Euler displacement $\boldsymbol{v} + \beta\boldsymbol{g}$. Proposition 1 is the corresponding guarantee: after the mixer is discarded, this $\boldsymbol{g}$ is a first-order ascent direction of the learned mixed value $Q_{\mathrm{tot}}$, which is the quantity each agent can query from $(o_i, a_i)$. Team-level IQL determines *which* mixed value is being ascended; the proposition determines that simultaneous local steps cannot first-order oppose it. Those two facts are the whole guidance claim of local queryability and team alignment, so no further condition on $\boldsymbol{v}$ or on the true return is required for it. The object is the same as in VDN, QMIX, and IGM: local utilities are related to $Q_{\mathrm{tot}}$, not to the true return (Sunehag et al., 2017; Rashid et al., 2020; Son et al., 2019). How return changes with $\beta$ is reported in Section 5.4 and Appendix F.

We test this claim on the executed $\boldsymbol{g}$. $B = \langle \nabla_{\boldsymbol{a}} Q_{\text{tot}}, \boldsymbol{g} \rangle$ is the first-order directional derivative of the learned mixed value along local utilities, and CFA predicts $B \geq 0$. $\cos(\nabla_{\boldsymbol{a}} Q_{\text{tot}}, \boldsymbol{g})$ measures how parallel the executed $\boldsymbol{g}$ remains to $\nabla_{\boldsymbol{a}} Q_{\text{tot}}$ after mixer weights are dropped (exactly one under VDN; not required under QMIX). Independent (matched) reuses the frozen CQF flow, state, action, $\lambda$, and $\beta$, replacing only $\boldsymbol{g}$; the inner product of that replacement against CQF's $Q_{\text{tot}}$ is the negative control that local queryability does not imply team alignment.

Table 6 confirms the identity for this $\boldsymbol{g}$, not for the composite Euler displacement $\boldsymbol{v} + \beta \boldsymbol{g}$. For CQF, $B \geq 0$ on **every** recorded step of every VDN and QMIX checkpoint. Mixer monotonicity is likewise strict: $m_i = \partial Q_{\text{tot}} / \partial Q_{\phi,i} > 0$ throughout. On VDN, $m_i$ is constant across agents, so $\nabla_{\boldsymbol{a}} Q_{\text{tot}}$ is parallel to $\boldsymbol{g}$ and $\cos(\nabla Q_{\text{tot}}, g) = 1.00$; the empirical inner product matches $\sum_i m_i \|\boldsymbol{g}_i\|_2^2$ to relative error $\sim 10^{-4}$. On QMIX, $m_i$ vary, so the cosine need not be one: we observe mean cosine $0.84$ while $B \geq 0$ remains exact. These QMIX runs apply per-agent gradient normalization at execution; the reported $B$ is the inner product with the *executed* $\boldsymbol{g}$, which is still nonnegative by the same chain rule. Independently trained local critics remain equally queryable, but they supply no analogous sign constraint against the learned team value. On the *same* CQF trajectory (identical $(\boldsymbol{s}, \boldsymbol{a}, \lambda, \beta)$ and frozen flow $\boldsymbol{v}$), replacing only $\boldsymbol{g}$ with independent-critic gradients yields a $B \geq 0$ rate far below the corresponding CQF setting (VDN: $B \geq 0$ $76.6\%$ against $100\%$, mean cosine $0.32$ against $1.00$; QMIX: $B \geq 0$ $60.3\%$ against $100\%$, mean cosine $0.15$) against $0.84$. Local queryability without team supervision is therefore not a substitute for Proposition 1: the independent direction can oppose $Q_{\text{tot}}$ at first order, whereas CQF cannot.

### C.2 The team-aligned guidance corrects the composite direction

Proposition 1 fixes the sign of the added term $\boldsymbol{g}$. Nevertheless, the action reached at the next denoising step is produced by the composite displacement $\boldsymbol{d}_\beta = \boldsymbol{v} + \beta \boldsymbol{g}$ (Eq. 6). This subsection therefore tests that composite at the critic's query point, and asks whether the guidance $\beta \boldsymbol{g}$ shaped by the team-level objective moves it toward ascent of the learned $Q_{\text{tot}}$.

We evaluate those effects at the one-step clean-action estimate for the following reasons. Team IQL fits $Q_{\text{tot}}$ on clean dataset actions (Section 4.2). Following QGF, execution queries the target critic at the one-step clean-action estimate $\hat{\boldsymbol{a}} = \boldsymbol{a}_\lambda + (1-\lambda)\boldsymbol{v}$, so $\boldsymbol{g} = \nabla_{\hat{\boldsymbol{a}}} Q$ is computed in the clean-action coordinates on which $Q_{\text{tot}}$ is trained (Zhou et al., 2026). The network that produces $\boldsymbol{v}$ is evaluated at the noisy interpolant $\boldsymbol{a}_\lambda$, but the linear path fixes the vector it returns: $\boldsymbol{v} = (\hat{\boldsymbol{a}} - \boldsymbol{a}_\lambda)/(1-\lambda)$ for $\lambda < 1$. Thus $\boldsymbol{v}$ is the action-space displacement from the noisy action to the clean estimate, up to a positive scale, and lies in the same Euclidean coordinates as $\hat{\boldsymbol{a}}$ and $\boldsymbol{g}$. The solver uses that shared space when it drops the Jacobian from $\boldsymbol{a}_\lambda$ to $\hat{\boldsymbol{a}}$ and adds $\boldsymbol{g}$ to $\boldsymbol{v}$. We therefore propose Proposition 2 at this query. It measures whether adding the team-aligned guidance to the imitation velocity moves the composite toward ascent of the learned $Q_{\text{tot}}$ at $\hat{\boldsymbol{a}}$. The probe it defines, $\langle \nabla_{\hat{\boldsymbol{a}}} Q_{\text{tot}}, \boldsymbol{d}_\beta \rangle$, is the directional derivative of $Q_{\text{tot}}$ at $\hat{\boldsymbol{a}}$ along the action-space vector the Euler step applies. The probe stays at the current query: the next clean estimate satisfies $\hat{\boldsymbol{a}}_{k+1} \neq \hat{\boldsymbol{a}}_k + \Delta\lambda \boldsymbol{d}_\beta$.

**Proposition 2** (Composite expansion at the query point)**.** *Evaluate $\nabla_{\boldsymbol{a}} Q_{\text{tot}}$ at a joint clean-action estimate $\hat{\boldsymbol{a}}$. Let $\boldsymbol{d}_\beta = \boldsymbol{v} + \beta \boldsymbol{g}$, $A = \langle \nabla_{\boldsymbol{a}} Q_{\text{tot}}, \boldsymbol{v} \rangle$, $B = \langle \nabla_{\boldsymbol{a}} Q_{\text{tot}}, \boldsymbol{g} \rangle$, and $C = A + \beta B$. Proposition 1 gives $B \geq 0$. Linearity yields $C = \langle \nabla_{\boldsymbol{a}} Q_{\text{tot}}, \boldsymbol{d}_\beta \rangle$. If $Q_{\text{tot}}(\boldsymbol{s}, \boldsymbol{o}, \cdot)$ is $L_Q$-smooth near $\hat{\boldsymbol{a}}$, the finite probe at the solver step $\Delta\lambda > 0$ obeys*

$$Q_{\text{tot}}(\hat{\boldsymbol{a}} + \Delta\lambda \boldsymbol{d}_\beta) \geq Q_{\text{tot}}(\hat{\boldsymbol{a}}) + \Delta\lambda\, C - \frac{L_Q (\Delta\lambda)^2}{2} \|\boldsymbol{d}_\beta\|_2^2. \tag{8}$$

*The finite probe is the difference of the two values of $Q_{\text{tot}}$,*

$$\Delta Q_{\text{tot}} = Q_{\text{tot}}(\hat{\boldsymbol{a}} + \Delta\lambda \boldsymbol{d}_\beta) - Q_{\text{tot}}(\hat{\boldsymbol{a}}), \tag{9}$$

*and the same bound rearranges to*

$$\Delta Q_{\text{tot}} \geq \Delta\lambda\, C - \frac{L_Q (\Delta\lambda)^2}{2} \|\boldsymbol{d}_\beta\|_2^2. \tag{10}$$

*On the right, $\Delta\lambda\, C$ is linear in the solver step and has the sign of the directional derivative $C$; the second term is a quadratic curvature term. As the step vanishes the linear piece dominates, so $C > 0$ is the first-order sign of the probe. At the step the solver uses, $\Delta\lambda = 1/10$, the curvature term can still make the finite difference negative.*

Table 6: Pooled diagnostics on frozen CQF checkpoints, split by mixer. $B \geq 0$ and $\cos(\nabla Q, g)$ test Proposition 1: the added team term is a first-order ascent of $Q_{\text{tot}}$. $A > 0$ is the imitation velocity alone. $C > 0$ and $\Delta Q > 0$ measure the composite $\boldsymbol{d}_\beta = \boldsymbol{v} + \beta \boldsymbol{g}$ at the clean-action estimate. The gap from $A > 0$ to $C > 0$, and from Independent (matched) to CQF, is the contribution of the team term. Percentages are over recorded Euler steps. Independent (matched) shares CQF's flow and $Q_{\text{tot}}$ and only replaces $\boldsymbol{g}$.

| Mixer | Method | $B \geq 0$ | $\cos(\nabla Q, g)$ | $A > 0$ | $C > 0$ | $\Delta Q > 0$ |
|---|---|---|---|---|---|---|
| VDN | CQF | 100.0 | 1.00 | 38.4 | 63.8 | 53.5 |
| VDN | Ind. (matched) | 76.6 | 0.32 | 38.4 | 40.1 | 33.7 |
| QMIX | CQF | 100.0 | 0.84 | 46.7 | 56.6 | 54.0 |
| QMIX | Ind. (matched) | 60.3 | 0.15 | 46.7 | 48.4 | 46.6 |

CQF therefore does not guarantee $C > 0$ or $\Delta Q_{\text{tot}} > 0$ on every step. The test is whether team-aligned guidance raises the fraction of steps with $C > 0$, and with a positive finite difference, relative to independent local gradients. We measure, for CQF and for Independent (matched), the fraction of solver steps with $A > 0$, $C > 0$, and $\Delta Q_{\text{tot}} > 0$. $A > 0$ is how often the imitation velocity $\boldsymbol{v}$ is itself an ascent direction of $Q_{\text{tot}}$ when no guidance is added. $C > 0$ is how often, at the operating $\beta$, the composite $\boldsymbol{d}_\beta$ is a first-order ascent direction of $Q_{\text{tot}}$. $\Delta Q_{\text{tot}} > 0$ is how often $Q_{\text{tot}}$ increases after one solver step $\Delta\lambda = 1/10$ along that same direction. The rise from $A > 0$ to $C > 0$, and the gap relative to Independent (matched), is the contribution of the team term $\beta B$.

We show the effect of the team-aligned guidance to the composite direction in Table 6. The frozen flow on its own is a poor ascent direction of the learned team value: $A > 0$ on $38.4\%$ of VDN steps and $46.7\%$ of QMIX steps in CQF. That gap is what test-time guidance is for. The behavior field copies the dataset, including actions the team critic ranks below other in-support actions, and the added term is the correction. Because $B \geq 0$ on every recorded step in CQF guaranteed by Proposition 1, the correction cannot point against $Q_{\text{tot}}$. So CQF change the composite: $C > 0$ rises to $63.8\%$ (VDN) and $56.6\%$ (QMIX) in CQF. The independent-matched control keeps the same flow, the same $Q_{\text{tot}}$, and the same $\beta$, and replaces only $\boldsymbol{g}$, so $A$ is unchanged. Its composite stays near the unguided rate, $C > 0$ on $40.1\%$ of VDN steps and $48.4\%$ of QMIX steps. The additional $23.7$ and $8.2$ percentage points are the team factorization. The finite probe has the same order: $\Delta Q_{\text{tot}} > 0$ on $53.5\%$ (VDN) and $54.0\%$ (QMIX) of steps, against $33.7\%$ and $46.6\%$ when the team gradient is replaced.

## D Ablations on Discrete Relaxation

A discrete action is a vertex of the probability simplex $\Delta^{A-1}$, and SMAC further restricts that vertex to a state-dependent legal set. Our method does not operate on this face. After each Euler step it clips the continuous estimate coordinatewise into the box $[0, 1]^A$, queries the critic there, and only at the end of denoising maps the vector to a legal action by masked $\arg\max$. This design is cheap, but it leaves two gaps relative to the discrete geometry: the critic query need not sum to one, and it may place mass on illegal coordinates. A natural alternative is to project the query onto $\Delta^{A-1}$, or onto the legal face of that simplex, so that $Q$-guidance stays on the set that the discrete policy actually occupies. The ablation below asks whether those projections change closed-loop return, and whether the residuals of box clipping are large enough, at the operating point we actually use, to justify the extra geometry.

We keep the trained parameters fixed and change only the inference geometry. **BoxClip** is the method used throughout the paper: coordinatewise clipping to $[0, 1]^A$, no legal mask on the gradient, and a legal $\arg\max$ at the end of denoising. **SimplexProj** replaces the box clip with a Duchi Euclidean projection onto the full simplex $\Delta^{A-1}$ and restricts the guidance gradient to the zero-sum tangent space; illegal coordinates may still carry mass until the final $\arg\max$. **MaskedSimplex** projects onto the legal face $\{p \geq 0, \sum_i p_i = 1, p_{\text{illegal}} = 0\}$ and zeros the gradient on illegal coordinates before the tangent projection. The flow state itself is never projected; only the critic query and the terminal output are. We evaluate the checkpoints in Table 3 ($4$ maps $\times$ $3$ dataset

Table 7: Return performance for three ablations. All the results are averaged over 3 seeds, and we report the standard deviations after the $\pm$ sign. All variants use the same $\beta$ as shown in Appendix I.

| Map | Dataset | BoxClip | SimplexProj | MaskedSimplex |
|---|---|---|---|---|
| 3m | Good | $\mathbf{20.0 \pm 1.9}$ | $17.1 \pm 2.5$ | $17.1 \pm 2.6$ |
| | Medium | $\mathbf{18.8 \pm 1.2}$ | $12.0 \pm 0.7$ | $12.1 \pm 0.7$ |
| | Poor | $\mathbf{12.2 \pm 2.7}$ | $6.8 \pm 0.8$ | $7.3 \pm 0.4$ |
| 2s3z | Good | $\mathbf{19.9 \pm 0.3}$ | $18.0 \pm 0.8$ | $17.4 \pm 0.9$ |
| | Medium | $\mathbf{18.2 \pm 0.0}$ | $14.0 \pm 1.2$ | $13.3 \pm 0.8$ |
| | Poor | $\mathbf{10.7 \pm 1.3}$ | $9.1 \pm 2.6$ | $9.3 \pm 1.7$ |
| 8m | Good | $\mathbf{19.8 \pm 0.4}$ | $17.8 \pm 0.7$ | $18.7 \pm 0.4$ |
| | Medium | $\mathbf{20.0 \pm 0.0}$ | $15.1 \pm 0.6$ | $13.1 \pm 1.0$ |
| | Poor | $\mathbf{8.0 \pm 0.4}$ | $6.6 \pm 0.8$ | $7.5 \pm 0.1$ |
| 5m_vs_6m | Good | $\mathbf{18.8 \pm 0.4}$ | $16.1 \pm 1.8$ | $16.2 \pm 1.7$ |
| | Medium | $\mathbf{16.7 \pm 1.2}$ | $14.5 \pm 0.3$ | $13.4 \pm 1.4$ |
| | Poor | $\mathbf{10.4 \pm 0.7}$ | $8.7 \pm 0.5$ | $8.6 \pm 0.3$ |

qualities $\times$ 3 seeds) with 10 Euler steps. The guidance weight $\beta$ is domain-specific as shown in Appendix I, and we evaluate the two projected variants at the same $\beta$.

Table 7 reports the task metric. BoxClip attains the highest return on every map and dataset. The gaps are largest on `3m` Medium (18.8 versus 12.0 and 12.1), `8m` Medium (20.0 versus 15.1 and 13.1), `3m` Poor (12.2 versus 6.8 and 7.3), and `2s3z` Medium (18.2 versus 14.0 and 13.3), each well outside the BoxClip seed interval. On `2s3z` Good, `8m` Good, and both Medium and Poor `5m_vs_6m`, the mean gap remains larger than one BoxClip standard deviation. Two cells are closer: on `3m` Good the BoxClip and SimplexProj intervals overlap ($20.0 \pm 1.9$ versus $17.1 \pm 2.5$), and on `2s3z` Poor SimplexProj is within noise of BoxClip ($10.7 \pm 1.3$ versus $9.1 \pm 2.6$). Averaging the twelve cells, BoxClip returns 16.1, SimplexProj 13.0, and MaskedSimplex 12.8. In all three variants the executed action is always legal: the terminal masked $\arg\max$ maps every query onto the legal set, so the environment never receives an illegal index. The geometry of the query is therefore a property of the critic input, not of the action that is executed.

Tables 8–9 confirm that the projections do what they are designed to do. The simplex residual $\|a - \Pi_\Delta(a)\|_1$ is the $\ell_1$ distance from the critic query to the probability simplex: it is zero iff $a$ is already a distribution over actions. Illegal mass is the total positive weight that $a$ places on coordinates forbidden by the current legal mask, $\sum_{i \notin \mathcal{L}(s)} \max(a_i, 0)$. It is zero if the query puts no mass on actions the environment would reject. The BoxClip query sits a small distance off the simplex: $\|a - \Pi_\Delta(a)\|_1$ and $|\sum_i a_i - 1|$ coincide at 0.016–0.035 on most cells, and they rise only when the selected $\beta$ is large (`2s3z` Poor at $\beta = 4$ gives 0.135). Both projected variants bring these residuals below $10^{-7}$. Illegal mass on the critic query is intermediate: BoxClip places 0.010–0.028 on forbidden coordinates on Good and Medium cells, SimplexProj about half of that, and MaskedSimplex exactly zero. The same ordering holds on Poor cells, where BoxClip illegal mass reaches 0.103 on `2s3z`. These diagnostics show that box clipping is not a simplex method. They also show that, at the $\beta$ chosen for return, the violation is small except at the two largest weights.

Taken together, the three tables separate geometry from control. Projecting the query onto $\Delta^{A-1}$ or onto the legal face removes residual mass that BoxClip leaves on the table, but that mass is not what the environment acts on, and removing it does not raise return. Box clipping plus a terminal legal $\arg\max$ is therefore a sufficient discrete interface: it produces legal executed actions, it matches or exceeds the closed-loop return of the two projected inference procedures on the $\beta$ we deploy, and it avoids a sorting projection and a legal-face gradient at every Euler step.

## E Additional Results for Coordination Diagnostics

In this section, we provide additional results for coordination diagnostics. We first compare the performance of CQF, MA-QGF and centralized variant on MA-MuJoCo in section E.1, plot the

Table 8: Simplex residual $\|a - \Pi_\Delta(a)\|_1$ of the critic query. Both simplex variants are numerically zero ($< 10^{-7}$) and are not bolded against each other.

| Map | Dataset | BoxClip | SimplexProj | MaskedSimplex |
|---|---|---|---|---|
| 3m | Good | 0.021 | **0.000** | **0.000** |
| | Medium | 0.018 | **0.000** | **0.000** |
| | Poor | 0.016 | **0.000** | **0.000** |
| 2s3z | Good | 0.028 | **0.000** | **0.000** |
| | Medium | 0.035 | **0.000** | **0.000** |
| | Poor | 0.135 | **0.000** | **0.000** |
| 8m | Good | 0.028 | **0.000** | **0.000** |
| | Medium | 0.024 | **0.000** | **0.000** |
| | Poor | 0.061 | **0.000** | **0.000** |
| 5m_vs_6m | Good | 0.025 | **0.000** | **0.000** |
| | Medium | 0.023 | **0.000** | **0.000** |
| | Poor | 0.027 | **0.000** | **0.000** |

Table 9: Illegal mass of the critic query. Lower is closer to the legal face. MaskedSimplex is identically zero.

| Map | Dataset | BoxClip | SimplexProj | MaskedSimplex |
|---|---|---|---|---|
| 3m | Good | 0.010 | 0.005 | **0.000** |
| | Medium | 0.010 | 0.006 | **0.000** |
| | Poor | 0.010 | 0.006 | **0.000** |
| 2s3z | Good | 0.019 | 0.011 | **0.000** |
| | Medium | 0.028 | 0.016 | **0.000** |
| | Poor | 0.103 | 0.057 | **0.000** |
| 8m | Good | 0.022 | 0.010 | **0.000** |
| | Medium | 0.022 | 0.011 | **0.000** |
| | Poor | 0.058 | 0.031 | **0.000** |
| 5m_vs_6m | Good | 0.014 | 0.005 | **0.000** |
| | Medium | 0.015 | 0.007 | **0.000** |
| | Poor | 0.019 | 0.010 | **0.000** |

learning curves of all ablations in section E.2, and then dive deeper into the coordination mechanism of CQF and MA-QGF on MPE in section E.3.

### E.1 Coordination Comparison on MA-MuJoCo

Table 10 repeats the comparison of Table 4 on MA-MuJoCo, where agents control disjoint joints of one body and are therefore coupled through shared dynamics and a shared reward. The protocol is the same as Section 5.3. As on MPE, CQF is the best deployable ablation on every split and raises the average from 2576.0 for MA-QGF to 2952.8. The centralized reference remains below CQF on average (2686.8 vs. 2952.8) and exceeds it only on 2-Ant Expert, Medium, and Medium-Expert. Figure 5 shows the same three ablations throughout training.

### E.2 Learning Curves

Figure 5 plots the corresponding learning curves: at each evaluation step we record the mean episode return averaged across 3 training seeds. Legend labels match Table 10: Independent is MA-QGF and Centralized is the fully centralized variant.

As shown in Figure 5, independently trained local critics remain as queryable as CQF, yet their team return typically plateaus early and can decline: on 2-Ant Medium and Medium-Replay the Independent curve peaks near 10k and then falls, and on 3-Hopper Medium it never leaves a band around 900–1180 while CQF climbs from 737 at 10k to 3211.8 at 200k. The largest and most

Table 10: **Team-level supervision, independent local guidance, and full centralization on MA-MuJoCo.** Entries are mean episode returns averaged over 3 seeds, and we report standard deviations after the $\pm$ sign. We highlight the best deployable arm in **bold** and the second best in underlined. M-E and M-R denote Medium-Expert and Medium-Replay. All arms share the same frozen behavior flow, critic architecture, $H$, expectile, and discount, and select $\beta$ from the same grid under the same protocol. †The centralized arm requires the global state and the joint action at execution, is not deployable under a Dec-POMDP, and is reported only as a reference.

| Scenario | Dataset | MA-QGF | CQF | Centralized† |
|---|---|---|---|---|
| 6-HalfCheetah | Expert | 4241.6 ±70.9 | **4461.9** ±22.0 | 4226.9 ±15.1 |
| | Medium | 3353.6 ±11.9 | **3721.2** ±26.2 | 3290.1 ±58.8 |
| | M-R | 3310.6 ±52.8 | **3765.3** ±4.7 | 3440.4 ±99.7 |
| | M-E | 4454.8 ±38.1 | **4589.5** ±28.3 | 4444.7 ±3.2 |
| 3-Hopper | Expert | 2837.4 ±388.1 | **3391.7** ±224.9 | 2832.6 ±155.4 |
| | Medium | 1097.9 ±77.8 | **3211.8** ±5.1 | 2961.7 ±490.6 |
| | M-R | 1845.6 ±300.2 | **2012.4** ±314.5 | 1304.7 ±178.7 |
| | M-E | 2735.8 ±275.1 | **3001.9** ±241.5 | 2302.4 ±160.5 |
| 2-Ant | Expert | 2067.6 ±5.9 | **2123.2** ±10.1 | 2168.2 ±28.3 |
| | Medium | 1442.0 ±6.7 | **1532.6** ±18.4 | 1620.4 ±9.7 |
| | M-R | 1459.1 ±8.2 | **1525.0** ±8.4 | 1509.6 ±2.2 |
| | M-E | 2066.5 ±2.9 | **2096.8** ±8.3 | 2139.9 ±16.3 |
| Average | | 2576.0 | **2952.8** | 2686.8 |

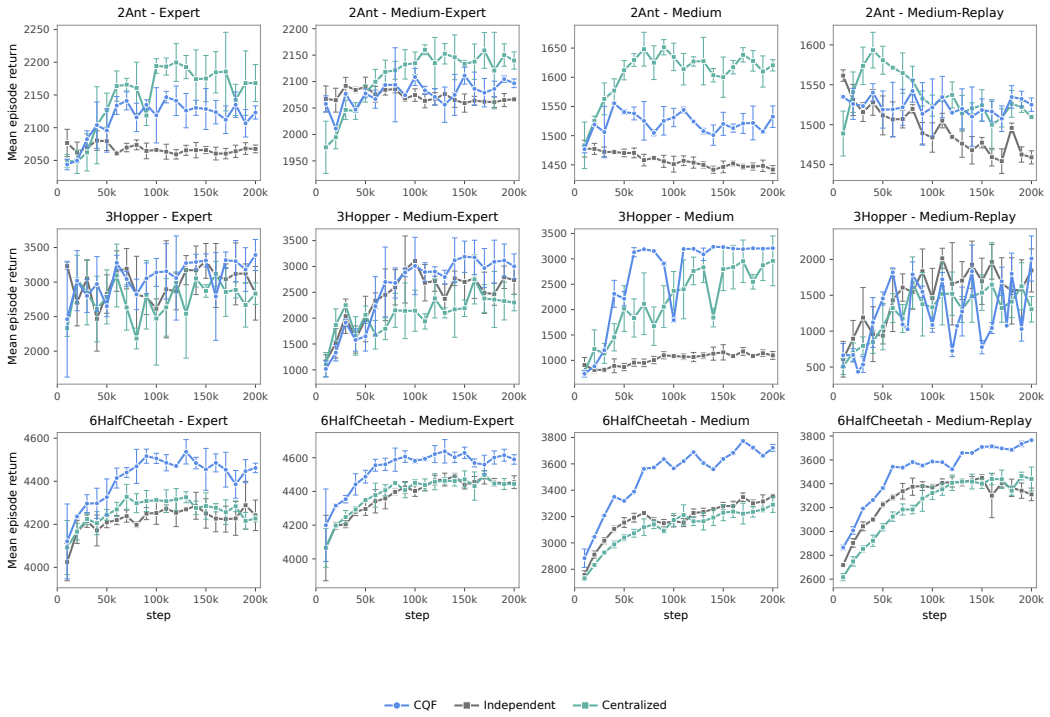


Figure 5: **Independent, team-aligned, and centralized guidance throughout training on MA-MuJoCo.** Each marker is the mean episode return at that training step. Error bars are the sample standard deviation across 3 training seeds. Independent denotes MA-QGF; Centralized denotes the joint-flow reference of Table 10, which is not executable under a Dec-POMDP.

persistent gaps between Independent and CQF appear on mixed and suboptimal data and on the higher-agent partitions (3-Hopper, 6-HalfCheetah), consistent with local IQL critics failing to assign team credit when the dataset does not already contain coordinated joint actions.

The centralized variant is team-aligned but not locally queryable, and it is not a uniform upper bound. It exceeds CQF only on 2-Ant Expert, Medium, and Medium-Expert, where a joint generator

can still exploit relatively complete team tuples; on 3-Hopper and 6-HalfCheetah, CQF lies above the centralized curve for most of training, and the centralized 3-Hopper Medium run is unstable (standard deviation $490.6$ at 200k). Taken together with Table 10, independently trained local critics are not sufficient, while team-supervised local utilities remain competitive with, and often more stable than, a centralized critic and joint flow that cannot be executed under a Dec-POMDP.

### E.3 Coordination Analysis on MPE

Section 5.3 shows that team-supervised local gradients raise team return over independently trained local critics under the same frozen flow. Return alone cannot determine whether that gap reflects better coordination or merely more aggressive local improvement. We therefore use MPE Simple Spread to ask two ordered questions: whether team supervision improves the task-defining behaviors of one-to-one landmark allocation and coverage, and whether any such gains extend to other coordination dimensions, particularly collision avoidance.

**Protocol.** We evaluate MA-QGF and CQF on all four MPE Simple Spread dataset qualities, using the same frozen behavior-flow initialization and the same hyperparameters, including the action chunk length $H$, expectile, discount, and guidance-weight grid. For the available training seed, each method is evaluated on the same 100 environment seeds, forming paired rollouts. The main coordination window is fixed to the final 25% of post-step states, and coverage uses $\epsilon = 0.1$.

For agent locations $p_i(t)$ and landmarks $\ell_j$, we define

$$D_{\text{cover}}(t) = \frac{1}{n}\sum_{j=1}^{n} \min_i \|p_i(t) - \ell_j\|_2, \tag{11}$$

$$D_{\text{assign}}(t) = \frac{1}{n}\min_{\sigma\in S_n}\sum_{i=1}^{n} \|p_i(t) - \ell_{\sigma(i)}\|_2, \tag{12}$$

$$C_\epsilon(t) = \frac{1}{n}\sum_{j=1}^{n} \mathbf{1}\left[\min_i \|p_i(t) - \ell_j\|_2 < \epsilon\right], \tag{13}$$

$$R_{\text{dup}}(t) = 1 - \frac{|\{\arg\min_j \|p_i(t) - \ell_j\|_2 : i = 1, \ldots, n\}|}{n}. \tag{14}$$

Here $D_{\text{cover}}$ measures whether every landmark is approached, $D_{\text{assign}}$ uses Hungarian matching to measure one-to-one task allocation, $C_\epsilon$ is the covered-landmark fraction, and $R_{\text{dup}}$ measures whether agents select the same nearest landmark. Joint completion indicates whether all landmarks are simultaneously covered; we report both whether this ever occurs and its frequency in the final-quarter window.

Table 11: **Effects of team supervision on MPE.** Rel. return is the percentage improvement of CQF over MA-QGF; all other entries are absolute differences over 100 paired episodes. $D_{\text{assign}}$ is the Hungarian one-to-one assignment distance, $R_{\text{dup}}$ is the duplicate nearest-landmark targeting rate, $D_{\text{cover}}$ is the mean landmark-to-nearest-agent distance, $C_{0.1}$ is the fraction of landmarks within $0.1$ of an agent, and $J_{\text{tail}}$ is the frequency with which all landmarks are simultaneously covered. Lower is better for distances and duplicate targeting; higher is better otherwise.

| Dataset | Rel. score ↑ | $\Delta D_{\text{assign}}$ ↓ | $\Delta R_{\text{dup}}$ ↓ | $\Delta D_{\text{cover}}$ ↓ | $\Delta C_{0.1}$ ↑ | $\Delta J_{\text{tail}}$ ↑ |
|---|---|---|---|---|---|---|
| Expert | $+8.8\%$ | $-0.043$ | $-0.031$ | $-0.039$ | $+0.090$ | $+0.109$ |
| Medium | $+42.1\%$ | $-0.060$ | $+0.063$ | $-0.063$ | $+0.127$ | $+0.011$ |
| M-Replay | $+139.9\%$ | $-0.312$ | $-0.006$ | $-0.192$ | $+0.054$ | $0.000$ |
| Random | $+20.2\%$ | $-0.048$ | $+0.066$ | $-0.002$ | $+0.082$ | $+0.007$ |

**Results analysis.** Table 11 shows that the normalized score of CQF exceeds that of MA-QGF by $8.8\%$–$139.9\%$ across the four data qualities. In the paired diagnostic rollouts, these performance gains consistently coincide with lower assignment distance ($0.043$–$0.312$), lower coverage distance ($0.002$–$0.192$), and higher coverage rate ($0.054$–$0.127$). Duplicate targeting decreases on Expert

and Medium-Replay but increases on Medium and Random. The stricter joint-completion measure is sparse but never decreases: its final-quarter frequency rises by $0.109$ on Expert, $0.011$ on Medium, and $0.007$ on Random, while remaining zero on Medium-Replay. Thus, the most stable behavioral pattern is that team-supervised guidance assigns agents to landmarks more effectively and leaves landmarks closer to an agent.

**Paired-rollout visualization.** To visualize this effect, we plot the agent trajectories, landmarks, and the Hungarian agent-to-landmark assignment at the end of the episode in Figures 6–9. For each dataset split, we uniformly sample eight episodes without replacement from the same pool of 100 paired evaluation seeds using a fixed random seed. No episode is selected according to its return or coordination metrics. For every sampled seed, CQF and MA-QGF start from exactly the same agent and landmark configuration, use the same frozen behavior-flow checkpoint, and use the same action-sampling random key. Each column therefore constitutes a paired comparison that differs only in the value guidance used during action generation. Colored curves show the trajectories of the three agents, open circles mark their initial positions, filled circles mark their final positions, and black crosses denote landmarks. Dashed lines show the optimal one-to-one Hungarian assignment between agents and landmarks at the final timestep. We report $\Delta R = R_{\mathrm{CQF}} - R_{\mathrm{MA\text{-}QGF}}$ for each paired episode. As shown in Figures 6–9, across the unfiltered random episodes, CQF more frequently disperses the agents toward different landmarks and substantially shortens the final one-to-one assignments, whereas MA-QGF often leaves multiple agents on the same side of the scene and requires long assignments to uncovered landmarks.

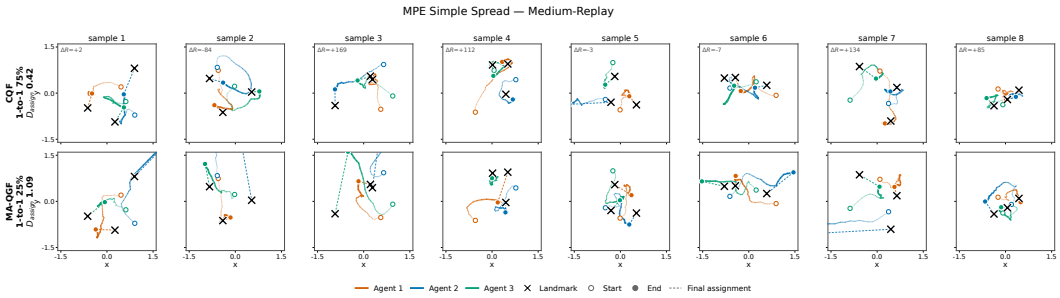


Figure 6: **Landmark coverage on MPE Simple Spread (Medium-Replay).** Paired rollouts of MA-QGF and CQF from identical initial states under the same frozen behavior flow, with agent trajectories, landmarks, and the Hungarian agent-to-landmark assignment at the end of the episode. Eight episodes are sampled uniformly from the same pool of 100 paired evaluation seeds.

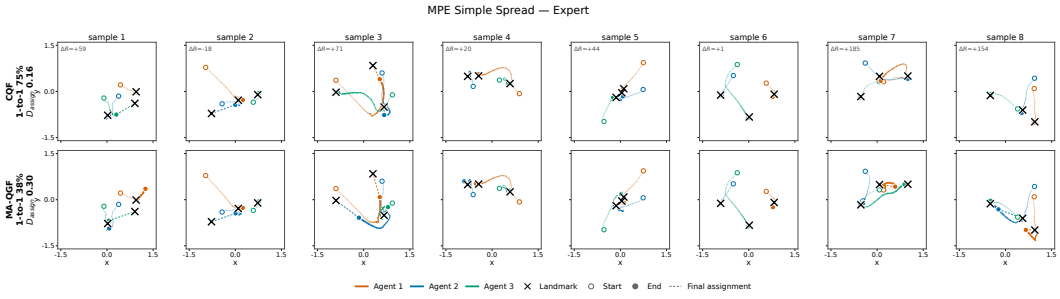


Figure 7: **Landmark coverage on MPE Simple Spread (Expert).** Same protocol as Figure 6.

## F Test-Time Guidance and Guidance-Weight Sweeps

Throughout this appendix the flow and critic are frozen and only $\beta$ varies, so every curve is produced from a single trained checkpoint per seed. The grid of Table 21 is evaluated on all 3 training seeds. We report the complete grids below, so that the width of the useful $\beta$ region, and not only its maximum, is visible.

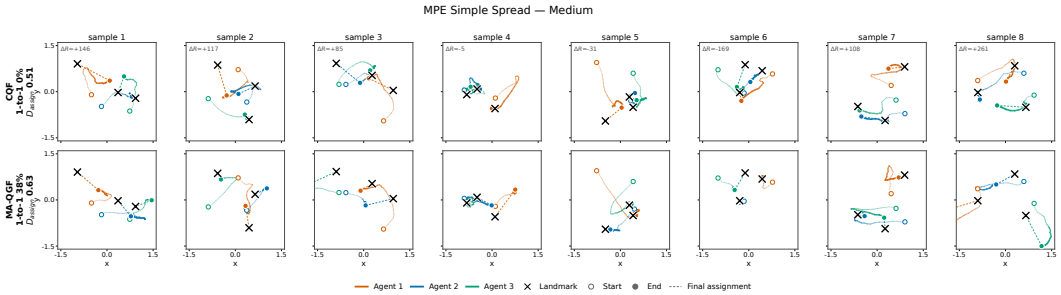


Figure 8: **Landmark coverage on MPE Simple Spread (Medium).** Same protocol as Figure 6.

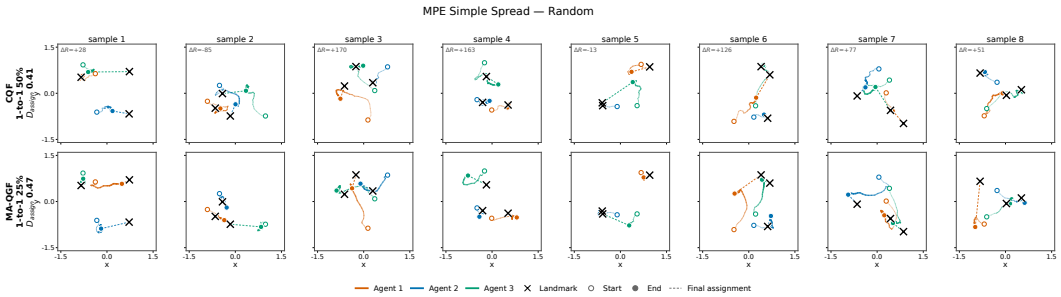


Figure 9: **Landmark coverage on MPE Simple Spread (Random).** Same paired-rollout protocol as Figure 6.

### F.1 Test-Time Guidance Comparisons

Tables 12–14 compare the matched $\beta = 0$ policy with the complete method. Test-time guidance improves all 28 settings, raising the benchmark average from 1718.3 to 2952.8 on MA-MuJoCo, from 28.2 to 60.7 on MPE, and from 13.0 to 16.1 on SMACv1.

On MA-MuJoCo, the largest absolute improvements occur on mixed and replay data, including Hopper Medium (672.8 to 3211.8) and HalfCheetah Medium-Replay (1241.3 to 3765.3). The gain is smaller on Ant Expert (2044.7 to 2123.2), where the unguided behavior flow already closely matches the complete method. On MPE, guidance yields its largest gain on Random (3.1 to 58.3) and more than doubles the average score. On SMACv1, the improvement is consistent across all four maps and three data qualities, with the largest gap on 3m Poor (4.2 to 12.2). Together, these comparisons show that the gains do not require updating the behavior model and are strongest when pure behavior cloning inherits weak or heterogeneous coordination from the dataset.

Table 12: **Effect of test-time guidance on MA-MuJoCo.** CQF ($\beta = 0$) is the frozen local behavior flow evaluated without guidance, while CQF uses the complete test-time guidance procedure. Entries are means with standard deviations. M-E and M-R denote Medium-Expert and Medium-Replay.

| Scenario | Method | Expert | Medium | M-R | M-E |
|---|---|---|---|---|---|
| HalfCheetah | CQF ($\beta = 0$) | 3624.0 ±467.5 | 1606.7 ±186.4 | 1241.3 ±121.2 | 3065.3 ±109.7 |
| | CQF | **4461.9** ±22.0 | **3721.2** ±26.2 | **3765.3** ±4.7 | **4589.5** ±28.3 |
| Hopper | CQF ($\beta = 0$) | 2471.4 ±524.0 | 672.8 ±8.4 | 506.9 ±24.2 | 1247.4 ±235.1 |
| | CQF | **3391.7** ±224.9 | **3211.8** ±5.1 | **2012.4** ±314.5 | **3001.9** ±241.5 |
| Ant | CQF ($\beta = 0$) | 2044.7 ±11.5 | 1424.9 ±2.0 | 1023.0 ±23.8 | 1691.5 ±50.9 |
| | CQF | **2123.2** ±10.1 | **1532.6** ±18.4 | **1525.0** ±8.4 | **2096.8** ±8.3 |

Table 13: **Effect of test-time guidance on MPE Simple Spread.** CQF ($\beta = 0$) evaluates the frozen local behavior flow without guidance, while CQF uses the complete test-time guidance procedure. Entries are D4RL-normalized means with standard deviations, $100 \times (S - 159.8)/(516.8 - 159.8)$. M-R denotes Medium-Replay.

| Method | Expert | Medium | M-R | Random |
|---|---|---|---|---|
| CQF ($\beta = 0$) | 65.5 ±3.6 | 28.0 ±4.1 | 16.0 ±6.4 | 3.1 ±4.9 |
| CQF | **96.4** ±0.6 | **55.0** ±6.8 | **33.1** ±1.7 | **58.3** ±9.4 |

Table 14: **Effect of test-time guidance on SMACv1.** CQF ($\beta = 0$) evaluates the frozen local behavior flow without guidance, while CQF uses the complete test-time guidance procedure. Entries are raw-return means with standard deviations.

| Scenario | Method | Good | Medium | Poor |
|---|---|---|---|---|
| 3m | CQF ($\beta = 0$) | 16.1 ±2.4 | 14.7 ±2.1 | 4.2 ±0.4 |
| | CQF | **20.0** ±1.9 | **18.8** ±1.2 | **12.2** ±2.7 |
| 8m | CQF ($\beta = 0$) | 17.5 ±0.6 | 18.1 ±1.2 | 4.9 ±0.2 |
| | CQF | **19.8** ±0.4 | **20.0** ±0.0 | **8.0** ±0.4 |
| 2s3z | CQF ($\beta = 0$) | 18.2 ±1.1 | 14.9 ±1.2 | 6.9 ±0.4 |
| | CQF | **19.9** ±0.3 | **18.2** ±0.0 | **10.7** ±1.3 |
| 5m_vs_6m | CQF ($\beta = 0$) | 17.4 ±0.7 | 13.6 ±0.3 | 9.2 ±0.8 |
| | CQF | **18.8** ±0.4 | **16.7** ±1.2 | **10.4** ±0.7 |

### F.2 Guidance Weights Sweep of MA-MuJoCo

Section 5.4 reports the 2-Ant sweep in Figure 2. This appendix collects the remaining MA-MuJoCo grids under the same protocol: $\theta$ and the critic are frozen, $\beta$ varies only at inference, and each point is the mean episode return at that grid value. The curves use the 200k-update checkpoint and the grid $\{0, 0.004, 0.008, 0.01, 0.02, 0.04, 0.06, 0.08, 0.1, 0.12, 0.14, 0.16, 0.18, 0.2, 0.25, 0.3\}$; $\beta = 0$ is CQF without guidance, evaluated on the same checkpoint.

Figures 10 and 11 show that a nonzero guidance weight improves every split over the unguided flow, but the response to $\beta$ is strongly dataset dependent. The largest gains occur where the frozen flow is weakest: on 6-HalfCheetah, Medium rises from 1606.7 to 3721.2 and Medium-Replay from 1241.3 to 3765.3; on 3-Hopper, the corresponding gains are 672.8 to 3211.8 and 506.9 to 2012.4. The 6-HalfCheetah curves generally improve across a relatively broad range of intermediate weights, whereas 3-Hopper Medium-Replay has a narrow optimum near $\beta = 0.004$, and 3-Hopper Expert remains comparatively variable across the grid. At overly large weights, performance can fall sharply, especially on 6-HalfCheetah Expert and 3-Hopper Medium and Medium-Replay, indicating that value guidance must remain balanced against the frozen behavior flow.

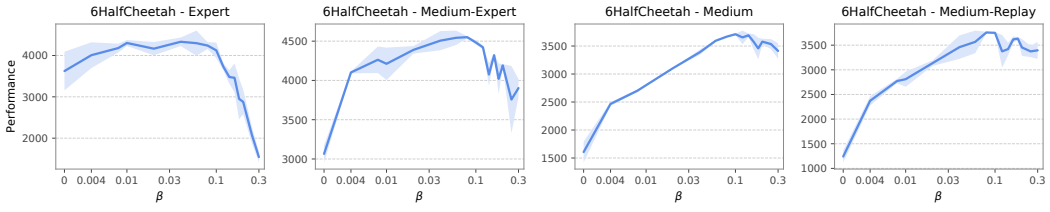


Figure 10: **Guidance-weight sweep on 6-HalfCheetah.** Same protocol as Figure 2. Shaded bands are the sample standard deviation across training seeds.

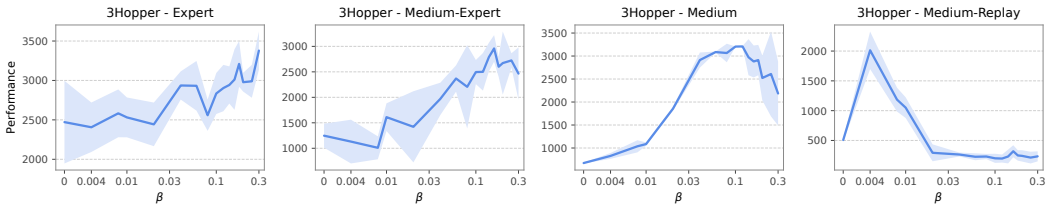


Figure 11: **Guidance-weight sweep on 3-Hopper.** Same protocol as Figure 2. Shaded bands are the sample standard deviation across training seeds.

## G Ablations on Local Flow and Centralized Flow

Eq. 4 matches each agent's velocity on its own action, so the generator does not score a joint action. This section compares that local flow with a flow trained on the joint action, both at $\beta = 0$. With the guidance weight removed, the critic does not enter the sampler, and the comparison is between the two behavior models.

Both flows train for $2 \times 10^5$ steps. On every split they share the network width, the number of Euler steps, the expectile, the discount, and the action-chunk length $H$ of that split in Table 19. The local column is CQF at $\beta = 0$, transcribed from Tables 12 and 13: each agent samples from its own flow given its local observation. The centralized column is one flow over the joint action, conditioned on the global state, also at $\beta = 0$. Sampling from it requires the global state and the joint action, so the centralized flow is not locally queryable under a Dec-POMDP and is reported only as a reference. MA-MuJoCo entries are episode returns. MPE entries use the normalization of Table 13, $100 \times (S - 159.8)/(516.8 - 159.8)$. The centralized MPE raw returns $553.5 \pm 20.4$, $259.0 \pm 38.0$, $221.7 \pm 22.5$, and $172.6 \pm 18.5$ are converted by that map.

Table 15: **Local and centralized behavior flows at $\beta = 0$.** Means $\pm$ sample standard deviation over 3 seeds after $2 \times 10^5$ steps. Bold marks the higher mean on each split. CQF ($\beta = 0$) is transcribed from Tables 12 and 13. MA-MuJoCo entries are episode returns. MPE entries are D4RL-normalized as in Table 13. $^\dagger$The centralized flow requires the global state and the joint action at execution and is a reference.

| Environment | Dataset | CQF ($\beta = 0$) | Centralized$^\dagger$ ($\beta = 0$) |
|---|---|---|---|
| 2-Ant | Expert | **2044.7** $\pm$11.5 | 2034.5 $\pm$2.2 |
| | Medium-Expert | 1691.5 $\pm$50.9 | **1717.1** $\pm$52.7 |
| | Medium | **1424.9** $\pm$2.0 | 1406.5 $\pm$11.8 |
| | Medium-Replay | 1023.0 $\pm$23.8 | **1071.6** $\pm$76.8 |
| 6-HalfCheetah | Expert | **3624.0** $\pm$467.5 | 3451.7 $\pm$387.3 |
| | Medium-Expert | **3065.3** $\pm$109.7 | 2721.6 $\pm$677.5 |
| | Medium | 1606.7 $\pm$186.4 | **1882.2** $\pm$113.9 |
| | Medium-Replay | **1241.3** $\pm$121.2 | 1208.1 $\pm$33.3 |
| 3-Hopper | Expert | 2471.4 $\pm$524.0 | **2611.5** $\pm$292.5 |
| | Medium-Expert | **1247.4** $\pm$235.1 | 1141.0 $\pm$361.7 |
| | Medium | 672.8 $\pm$8.4 | **780.0** $\pm$158.6 |
| | Medium-Replay | **506.9** $\pm$24.2 | 496.9 $\pm$60.3 |
| MPE | Expert | 65.5 $\pm$3.6 | **110.3** $\pm$5.7 |
| | Medium | **28.0** $\pm$4.1 | 27.8 $\pm$10.6 |
| | Medium-Replay | 16.0 $\pm$6.4 | **17.3** $\pm$6.3 |
| | Random | 3.1 $\pm$4.9 | **3.6** $\pm$5.2 |

On MA-MuJoCo the twelve-split average is 1718.3 for the local flow and 1710.2 for the joint flow. On ten of the twelve splits the gap lies inside the larger of the two seed standard deviations. The joint flow is higher on 6-HalfCheetah Medium ($1882.2 \pm 113.9$ versus $1606.7 \pm 186.4$). The local flow is higher on 2-Ant Medium ($1424.9 \pm 2.0$ versus $1406.5 \pm 11.8$). On MPE the joint flow is higher on Expert ($110.3 \pm 5.7$ versus $65.5 \pm 3.6$) and matches the local flow on Medium, Medium-Replay, and Random.

An explicit joint behavior model therefore raises unguided return on Expert MPE, where the logged actions are already coordinated, and does not do so across the remaining splits. Reading that model

at execution still requires the global state and the joint action, so the extra joint dependence is not available to a locally queryable policy. On the local flow, the same dependence is carried by the monotonically factorized team critic and added at test time as a local utility gradient (Proposition 1). Section 5.4 evaluates these local checkpoints with utility gradients and records the rise over the $\beta = 0$ column, including on the mixed and suboptimal splits where the joint flow at $\beta = 0$ matches the local flow. Guiding the joint flow instead uses the gradient of a monolithic centralized critic, evaluated on the global state and the joint action. Tables 4 and 10 compare that guided centralized policy with CQF. CQF attains the higher average, $60.7$ versus $56.4$ on MPE and $2952.8$ versus $2686.8$ on MA-MuJoCo, and it is the best deployable method on every split. The centralized arm is higher only on MPE Expert and on 2-Ant Expert, Medium, and Medium-Expert. Table 15 shows that the unguided flows do not produce this average gap, so the advantage appears when the centralized critic gradient is replaced by the factorized local utilities. The centralized arm stays a reference, because that gradient cannot be evaluated under a Dec-POMDP.

# H GOAL-CONDITIONED OFFLINE MARL

The single-task datasets of Sections 5.2–5.5 fix the reward at collection time, so a trained policy is never asked to pursue a new objective. Goal-conditioned offline MARL instead commands the team, only at evaluation, generalize to multiple goals that need not have been the destination of any logged trajectory, which is more difficult than the single-goal setting of Sections 5.2–5.5. We compare GCCQF, GCMadiff, and GCMAC-Flow on one shared MangoBench protocol (Wang et al., 2026); the problem below is common, and the subsections record only method-specific adaptations.

**Problem.** Each agent $i \in \{1, \dots, N\}$ receives a local observation $o_i$ and a local slice $g_i$ of a commanded configuration $g$, and outputs a local action $a_i$. The offline dataset $\mathcal{D}$ consists of unlabeled trajectories. Following MangoBench (Wang et al., 2026), training uses hindsight relabeling: a transition is assigned a goal sampled from a future state on the same trajectory or from a random dataset state, so every snippet becomes an example of reaching that relabeled goal. The learning signal is a sparse goal-reaching reward $r$,

$$r(s, g) = \begin{cases} 0, & s = g, \\ -1, & s \neq g, \end{cases} \tag{15}$$

where $s = g$ denotes reaching the goal within the environment's success radius. The reward $r$ is $0$ upon success and $-1$ otherwise, so the return equals minus the remaining discounted time-to-goal and the action value satisfies $Q \in [-1/(1-\gamma), 0]$. Evaluation reports success rate and the standard deviation.

**Comparison protocol.** The three methods share the dataset, the hindsight-relabeling distributions, the agent partition, the discount, the number of gradient steps, the evaluation goals, and the number of evaluation episodes, all listed in Table 18; they are also matched in the width and depth of the networks they share, so that no method is given a larger backbone. Each method retains the components that define it, namely the joint trajectory diffusion and inverse dynamics of MADiff, the distilled one-step actor of MAC-Flow, and the factorized critic and test-time guidance of CQF, and each is tuned over its own method-specific controls, namely the planning window and diffusion steps for GCMADiff, the distillation coefficient and Q-aggregation for GCMAC-Flow, and $\beta$ for GCCQF, as shown in Table 20 and Table 21.

## H.1 GCCQF

Relative to CQF, every network concatenates the local goal $g_i$:

$$Q_\phi(\boldsymbol{o}_i^{\mathrm{id}}, g_i, \boldsymbol{a}_i) = \mathrm{MLP}_\phi\big([\boldsymbol{o}_i^{\mathrm{id}};\, g_i;\, \boldsymbol{a}_i]\big), \tag{16}$$

$$V_\psi(\boldsymbol{o}_i^{\mathrm{id}}, g_i) = \mathrm{MLP}_\psi\big([\boldsymbol{o}_i^{\mathrm{id}};\, g_i]\big), \tag{17}$$

$$\boldsymbol{v}_\theta(\boldsymbol{o}_i^{\mathrm{id}}, g_i, \boldsymbol{a}_{i,\lambda}, \lambda) = \mathrm{MLP}_\theta\big([\boldsymbol{o}_i^{\mathrm{id}};\, g_i;\, \boldsymbol{a}_{i,\lambda};\, \lambda]\big). \tag{18}$$

The flow, critic, and guidance objectives are then the goal-conditioned counterparts of Eqs. 4–6. The local flow loss is

$$\mathcal{L}_{\mathrm{FM}}(\theta) = \mathbb{E}_{\mathcal{D},\,\boldsymbol{\epsilon}_i,\,\lambda}\Big[\tfrac{1}{N}\sum_{i=1}^{N}\big\|\boldsymbol{v}_\theta(\boldsymbol{o}^{\mathrm{id}}_{i,t}, g_i, \boldsymbol{a}_{i,\lambda}, \lambda) - \boldsymbol{u}_i\big\|_2^2\Big], \tag{19}$$

with $\boldsymbol{a}_{i,\lambda}$ and $\boldsymbol{u}_i$ as in Eq. 4. Local utilities are mixed as in Eq. 5, now conditioned on $g$,

$$Q_{\mathrm{tot},t} = M_\phi^Q\big(Q_{\phi,1}(\boldsymbol{o}^{\mathrm{id}}_{1,t}, g_1, \boldsymbol{a}_{1,t}), \ldots, Q_{\phi,N}(\boldsymbol{o}^{\mathrm{id}}_{N,t}, g_N, \boldsymbol{a}_{N,t}); \boldsymbol{s}_t\big), \tag{20}$$

and likewise $V_{\mathrm{tot},t} = M_\psi^V\big(V_{\psi,1}(\boldsymbol{o}^{\mathrm{id}}_{1,t}, g_1), \ldots, V_{\psi,N}(\boldsymbol{o}^{\mathrm{id}}_{N,t}, g_N); \boldsymbol{s}_t\big)$. With team reward $r_t = \frac{1}{N}\sum_{i=1}^{N} r(s_{i,t}, g_i)$ from Eq. 15 and termination $d_t$ when the commanded goal is reached, the IQL losses are

$$\mathcal{L}_Q(\phi) = \mathbb{E}_{\mathcal{D}}\big[(Q_{\mathrm{tot},t} - y_t)^2\big], \qquad y_t = r_t + \gamma(1-d_t)V_{\mathrm{tot},t+1}, \tag{21}$$

$$\mathcal{L}_V(\psi) = \mathbb{E}_{\mathcal{D}}\big[L_2^\tau(\bar{Q}_{\mathrm{tot},t} - V_{\mathrm{tot},t})\big]. \tag{22}$$

At execution the mixer is discarded. Agent $i$ evaluates $\boldsymbol{v}_i = \boldsymbol{v}_\theta(\boldsymbol{o}^{\mathrm{id}}_i, g_i, \boldsymbol{a}_{i,\lambda}, \lambda)$, forms the same one-step estimate $\hat{\boldsymbol{a}}_{i,1}$ as in Section 4.3, and guides with the local target utility $\boldsymbol{g}_i = \nabla_{\hat{\boldsymbol{a}}_{i,1}} Q_{\bar{\phi},i}(\boldsymbol{o}^{\mathrm{id}}_i, g_i, \hat{\boldsymbol{a}}_{i,1})$:

$$\boldsymbol{a}_{i,\lambda+\Delta\lambda} = \boldsymbol{a}_{i,\lambda} + \Delta\lambda\,(\boldsymbol{v}_i + \beta\boldsymbol{g}_i)\,. \tag{23}$$

### H.2 GCMAC-Flow

GCMAC-Flow is the goal-conditioned adaptation of MAC-Flow (Lee et al., 2026), which instantiates FQL (Park et al., 2025) with parameter-shared networks and a team-mean critic. Every network concatenates $g_i$:

$$Q_\psi(\boldsymbol{o}^{\mathrm{id}}_i, g_i, \boldsymbol{a}_i) = \mathrm{MLP}_\psi\big([\boldsymbol{o}^{\mathrm{id}}_i;\, g_i;\, \boldsymbol{a}_i]\big), \tag{24}$$

$$\boldsymbol{v}_\theta(\boldsymbol{o}^{\mathrm{id}}_i, g_i, \boldsymbol{a}_{i,\lambda}, \lambda) = \mathrm{MLP}_\theta\big([\boldsymbol{o}^{\mathrm{id}}_i;\, g_i;\, \boldsymbol{a}_{i,\lambda};\, \lambda]\big), \tag{25}$$

$$\mu_\phi(\boldsymbol{o}^{\mathrm{id}}_i, g_i, \boldsymbol{z}_i) = \mathrm{MLP}_\phi\big([\boldsymbol{o}^{\mathrm{id}}_i;\, g_i;\, \boldsymbol{z}_i]\big). \tag{26}$$

The critic is trained by one-step TD on the team-mean value, with next actions from the one-step actor. With team reward $r_t = \frac{1}{N}\sum_{i=1}^{N} r(s_{i,t}, g_i)$ from Eq. 15 and termination $d_t$ when the commanded goal is reached,

$$y_{i,t} = r_t + \gamma(1-d_t)\,Q_{\bar{\psi}}(\boldsymbol{o}^{\mathrm{id}}_{i,t+1}, g_i, \mu_\phi(\boldsymbol{o}^{\mathrm{id}}_{i,t+1}, g_i, \boldsymbol{z}'_i)), \tag{27}$$

$$Q_t^{\mathrm{mix}} = \tfrac{1}{N}\sum_{i=1}^{N} Q_\psi(\boldsymbol{o}^{\mathrm{id}}_{i,t}, g_i, \boldsymbol{a}_{i,t}), \qquad \mathcal{L}_Q(\psi) = \mathbb{E}_{\mathcal{D}}\big[(Q_t^{\mathrm{mix}} - y_t^{\mathrm{mix}})^2\big], \tag{28}$$

where $y_t^{\mathrm{mix}} = \frac{1}{N}\sum_i y_{i,t}$ and $\boldsymbol{z}'_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. The actor keeps MAC-Flow's three terms, now conditioned on $g_i$: rectified-flow behavioral cloning, distillation of an Euler-integrated flow into $\mu_\phi$, and team-mean $Q$-maximization,

$$\mathcal{L}_{\mathrm{BC}}(\theta) = \mathbb{E}\big[\big\|\boldsymbol{v}_\theta(\boldsymbol{o}^{\mathrm{id}}_i, g_i, \boldsymbol{a}_{i,\lambda}, \lambda) - \boldsymbol{u}_i\big\|_2^2\big], \tag{29}$$

$$\mathcal{L}_{\mathrm{distill}}(\phi) = \mathbb{E}\big[\big\|\mu_\phi(\boldsymbol{o}^{\mathrm{id}}_i, g_i, \boldsymbol{z}_i) - \hat{\boldsymbol{a}}_i^{\mathrm{flow}}\big\|_2^2\big], \tag{30}$$

$$\mathcal{L}_Q^\pi(\phi) = -\mathbb{E}\big[\tfrac{1}{N}\sum_{i=1}^{N} Q_\psi(\boldsymbol{o}^{\mathrm{id}}_i, g_i, \mu_\phi(\boldsymbol{o}^{\mathrm{id}}_i, g_i, \boldsymbol{z}_i))\big], \tag{31}$$

$$\mathcal{L}_\pi = \mathcal{L}_{\mathrm{BC}} + \alpha\,\mathcal{L}_{\mathrm{distill}} + \mathcal{L}_Q^\pi. \tag{32}$$

Here $\hat{\boldsymbol{a}}_i^{\mathrm{flow}}$ is obtained by $K$ Euler steps of $\boldsymbol{v}_\theta$ from $\boldsymbol{z}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. At execution, agent $i$ samples

$$\boldsymbol{a}_i = \mu_\phi(\boldsymbol{o}^{\mathrm{id}}_i, g_i, \boldsymbol{z}_i), \qquad \boldsymbol{z}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{33}$$

### H.3 GCMADIFF

GCMADiff is the goal-conditioned adaptation of MADiff (Zhu et al., 2024). MADiff (Zhu et al., 2024) models joint observation trajectories with a shared attention-based diffusion U-Net and recovers actions with inverse dynamics; targeted generation is originally realized by classifier-free guidance on a scalar return $\boldsymbol{y}(\boldsymbol{\tau})$. Unlike GCCQF and GCMAC-Flow, which concatenates $g_i$ into every MLP and retrains a goal-conditioned critic, GCMADiff leaves the U-Net, the attention backbone, and the inverse-dynamics head unchanged. It instead treats the commanded goal as a trajectory-level *hard* constraint and implements it by inpainting (Janner et al., 2022): the last observation of the planning window is overwritten with the goal and held fixed throughout denoising.

Following MADiff, we diffuse over joint observations with a $C$-step history and a planning window of $W$ steps (distinct from the action chunk length $H$ of Section 4.1, which GCMADiff does not use),

$$\boldsymbol{\tau} = [\boldsymbol{o}_{t-C}, \ldots, \boldsymbol{o}_t, \boldsymbol{o}_{t+1}, \ldots, \boldsymbol{o}_{t+W-1}], \qquad \boldsymbol{h}_t = [\boldsymbol{o}_{t-C}, \ldots, \boldsymbol{o}_t], \tag{34}$$

and disable return-conditioning ($\boldsymbol{y}(\boldsymbol{\tau}) = \emptyset$), since rewards are defined only after a goal is specified in MangoBench (Wang et al., 2026). Let $\boldsymbol{g} = (g_1, \ldots, g_N)$ be the per-agent partition of the goal observation. Define the inpainting mask and the conditioned array by

$$\mathcal{M}_\ell = \begin{cases} \mathbf{1}, & \ell \in \{t-C, \ldots, t\} \text{ or } \ell = t+W-1, \\ \mathbf{0}, & \text{otherwise}, \end{cases} \qquad \boldsymbol{\tau}_\ell^{\text{cond}} = \begin{cases} \boldsymbol{o}_\ell, & \ell \le t, \\ \boldsymbol{g}, & \ell = t+W-1. \end{cases} \tag{35}$$

Every reverse-diffusion step overwrites the masked entries,

$$\boldsymbol{\tau}_k \leftarrow (1 - \mathcal{M}) \odot \boldsymbol{\tau}_k + \mathcal{M} \odot \boldsymbol{\tau}^{\text{cond}}, \tag{36}$$

so sampling is equivalent to drawing from the constrained distribution

$$p_\theta(\boldsymbol{\tau} \mid \boldsymbol{h}_t, \boldsymbol{g}) \propto p_\theta(\boldsymbol{\tau})\, \delta_{\boldsymbol{h}_t}(\boldsymbol{\tau}_{t-C:t})\, \delta_{\boldsymbol{g}}(\boldsymbol{\tau}_{t+W-1}). \tag{37}$$

The interior $[\boldsymbol{o}_{t+1:t+W-2}]$ is the only free region and is filled so as to be consistent with both the history and the terminal goal.

Training uses the same overwrite equation 36 with a hindsight goal: for a sampled window, the last observation of that window plays the role of $\boldsymbol{g}$. No extra goal channel is concatenated. Dropping the return token from MADiff's objective, the losses reduce to reverse diffusion and inverse dynamics,

$$\mathcal{L}_{\text{diff}}(\theta) = \mathbb{E}_{k, \boldsymbol{\tau}_0, \epsilon}\big[\big\|\epsilon - \epsilon_\theta(\hat{\boldsymbol{\tau}}_k, k)\big\|_2^2\big], \tag{38}$$

$$\mathcal{L}_{\text{inv}}(\phi) = \sum_{i=1}^{N} \mathbb{E}_{(o^i, a^i, o'^i) \sim \mathcal{D}}\big[\big\|a^i - I_\phi^i(o^i, o'^i)\big\|_2^2\big], \tag{39}$$

$$\mathcal{L} = \tfrac{1}{2}\big(\mathcal{L}_{\text{diff}} + \mathcal{L}_{\text{inv}}\big), \tag{40}$$

where $\hat{\boldsymbol{\tau}}_k$ is the noised window after inpainting. Centralized execution starts from low-temperature noise and applies the same constraint at every denoising step,

$$\underbrace{[\boldsymbol{h}_t,\, \tilde{\boldsymbol{x}}_{K,t+1:t+W-2},\, \boldsymbol{g}]}_{\hat{\boldsymbol{\tau}}_K} \xrightarrow{K \text{ steps}} \underbrace{[\boldsymbol{h}_t,\, \hat{\boldsymbol{o}}_{t+1:t+W-2},\, \boldsymbol{g}]}_{\hat{\boldsymbol{\tau}}_0}, \tag{41}$$

after which actions are obtained exactly as in MADiff (Zhu et al., 2024),

$$\hat{a}_t^i = I_\phi^i(\hat{o}_t^i, \hat{o}_{t+1}^i). \tag{42}$$

Thus GCMADiff conditions on $g_i$ by constraining the *last* column of the joint trajectory array, whereas GCCQF and GCMAC-Flow conditions by concatenating $g_i$ into the *input* of every network.

## I IMPLEMENTATION DETAILS

**System specification.** Table 16 lists the system specification for the experiments.

Table 16: System specification for the experiments.

| | |
|---|---|
| CPU | 2× Intel Xeon Platinum 8468 |
| GPU | NVIDIA H100 80GB HBM3 |
| Software | CUDA: 12.9, cudnn: 9.23.0, python: 3.10, JAX: 0.4.30 |

**Training and evaluation protocols.** We train CQF with $2 \times 10^5$ gradient steps on the standard tasks and $1 \times 10^6$ gradient steps on MangoBench (Tables 17 and 18). Training uses three seeds on the hardware of Table 16. Evaluation uses the episode counts in those tables, 10 episodes per seed on MA-MuJoCo, MPE, and SMACv1 following Lee et al. (2026) and 20 episodes per task on MangoBench following Zhou et al. (2026), and reports the mean and standard deviation across the three seeds. All methods we train read the same released datasets: the OMIGA splits for MA-MuJoCo (Wang et al., 2023b), the OMAR splits for MPE (Pan et al., 2022), the Off-the-Grid splits for SMACv1 (Formanek et al., 2023), and the $2 \times 4$ AntMaze partition of MangoBench (Wang et al., 2026). The goal-conditioned results use one shared MangoBench protocol for GCMAD-iff, GCMAC-Flow, and GCCQF (Appendix H): the same dataset, relabeling distributions, network width and depth, discount, gradient steps, evaluation goals, and episode count, with each method tuned only on the control that defines it.

**Protocols for guidance weights.** The weight $\beta$ in Eq. 6 scales the critic gradient added to a frozen behavior flow. A new value is chosen after training and leaves the flow and critic unchanged, as in single-agent test-time guidance (Zhou et al., 2026; Xu et al., 2026; Jang et al., 2025). For each dataset we evaluate a fixed grid on the online returns of all three training seeds, retain the grid value that maximizes their mean, and apply that one value to every seed of the dataset. Table 21 lists the grid, Table 19 lists the selected value, and Appendix F reports the return at every grid point on the same checkpoints, including $\beta = 0$. On MA-MuJoCo the grid is $\{0, 0.004, 0.008, 0.01, 0.02, 0.04, 0.06, 0.08, 0.1, 0.12, 0.14, 0.16, 0.18, 0.2, 0.25, 0.3\}$, which contains the grid QGF publishes and covers the rise-then-fall region of Section 5.4.

The table entry is the maximizer of this grid on the online episodes that the tables report. The grid is fixed before the comparison, and one weight is shared by all seeds, so no seed receives its own maximizer. The selection criterion is the mean online return of the three training seeds. Existing coupled methods likewise select a behavior-regularization weight by online return, and each candidate of that weight is a separate training run because the coefficient enters the training loss. Each candidate of $\beta$ reuses one frozen flow and critic, so the grid is an inference sweep on a checkpoint that has already been trained, improving tuning efficiency (Section 5.5). Each quality split receives its own weight, because the peak moves across splits: Expert data prefers a small weight, while mixed and suboptimal splits prefer a larger one (Section 5.4, Appendix F). The gain over the unguided flow is read from the whole curve. Return rises above $\beta = 0$ across a range of weights and falls once guidance is large. Ablations that reuse a frozen flow, including MA-QGF and the centralized reference, select $\beta$ by the same rule.

**Hyperparameters.** Table 17 lists the common configurations for the standard offline MARL experiments, and Table 18 lists those for the goal-conditioned experiments.

The monotonic mixer $M_\phi^Q$ in Eq. 5 is instantiated by VDN or QMIX (Table 19), both of which satisfy $\partial Q_{\text{tot}}/\partial Q_{\phi,i} \geq 0$ in Eq. 3 and are discarded at execution (Section 4.3). VDN (Sunehag et al., 2017) additively decomposes the team value, $Q_{\text{tot}} = \sum_i Q_{\phi,i}$. We replace the sum by a mean reduction so that the mixed value matches the team reward $r_t = \frac{1}{N}\sum_i r_{i,t}$ of Section 4.2; then $m_i$ in Proposition 1 is a constant, and each local utility gradient is exactly the corresponding block of $\nabla_{\boldsymbol{a}} Q_{\text{tot}}$ up to that constant. QMIX (Rashid et al., 2020) instead uses a mixing network whose weights are produced by state-conditioned hypernetworks and constrained to be nonnegative, typically by an absolute-value activation. This yields a larger class of monotonic mixers: $m_i$ may depend on $s$ and differ across agents, so discarding the mixer preserves the sign of the inner product in Eq. 7 but rescales each agent's guidance step. We therefore additionally normalize the local action gradient under QMIX, so that the per-agent steps remain on a comparable scale. Parameters are shared across agents, with one-hot identity appended to local observations as $\boldsymbol{o}_{i,t}^{\text{id}}$.

Table 19 lists the method-specific hyperparameters for standard offline MARL tasks, and Table 21 lists the tuning range of guidance weights for all tasks. The selected $\beta$ on each row is the dataset-level choice of the protocol above.

Table 17: **Common hyperparameters** used for standard offline MARL experiments on MA-MuJoCo, MPE and SMACv1.

| Parameter | Value |
|---|---|
| Batch size | 64 |
| Flow steps | 10 |
| Polyak averaging coefficient | 0.005 |
| Offline training steps | $2 \times 10^5$ |
| Evaluation episodes | 10 |
| Critic Network Size | [512, 512, 512, 512] |
| Actor Network Size | [512, 512, 512, 512] |
| Learning rate | 3e-4 |
| Critic ensemble size | 2 |
| Critic aggregation method | min |
| Optimizer | Adam |

Table 18: **Common hyperparameters** used for offline goal-conditioned MARL experiments on MangoBench.

| Parameter | Value |
|---|---|
| Batch size | 1024 |
| Discount factor | 0.99 for medium and large mazes, 0.995 for giant-maze |
| Optimizer | Adam |
| Learning rate | 3e-4 |
| Polyak averaging coefficient | 0.005 |
| Offline training steps | $1 \times 10^6$ |
| Evaluation episodes | 20 per task |
| Critic Network Size | [512, 512, 512, 512] |
| Actor Network Size | [512, 512, 512, 512] |
| Critic ensemble size | 2 |
| Critic aggregation method | min |
| IQL expectile | 0.9 |
| Flow steps | 10 |
| Horizon ($h$) | 5 |
| Action chunking | False |
| Agent configuration | 2×4 |
| Actor goal relabeling $(p_{\mathrm{cur}}, p_{\mathrm{traj}}, p_{\mathrm{rand}})$ | navigate $(0.0, 1.0, 0.0)$, stitch $(0.0, 0.5, 0.5)$, explore $(0.0, 0.0, 1.0)$ |
| Value goal relabeling $(p_{\mathrm{cur}}, p_{\mathrm{traj}}, p_{\mathrm{rand}})$ | $(0.2, 0.5, 0.3)$ |

Table 19: Domain-specific hyperparameters for MA-MuJoCo, MPE, and SMACv1. $H$ is the action chunk length of Section 4.1. $\beta$ is the guidance weight.

| Task | Dataset | Mixer | $H$ | Expectile | Discount | $\beta$ |
|---|---|---|---|---|---|---|
| 6-HalfCheetah | Expert | VDN | 1 | 0.8 | 0.99 | 0.04 |
| 6-HalfCheetah | M-Expert | VDN | 1 | 0.9 | 0.99 | 0.1 |
| 6-HalfCheetah | Medium | VDN | 1 | 0.9 | 0.99 | 0.1 |
| 6-HalfCheetah | M-Replay | VDN | 1 | 0.9 | 0.99 | 0.08 |
| 3-Hopper | Expert | QMIX | 5 | 0.9 | 0.995 | 0.3 |
| 3-Hopper | M-Expert | QMIX | 1 | 0.8 | 0.995 | 0.16 |
| 3-Hopper | Medium | VDN | 5 | 0.9 | 0.99 | 0.12 |
| 3-Hopper | M-Replay | VDN | 1 | 0.9 | 0.99 | 0.004 |
| 2-Ant | Expert | VDN | 5 | 0.9 | 0.99 | 0.04 |
| 2-Ant | M-Expert | VDN | 5 | 0.9 | 0.99 | 0.02 |
| 2-Ant | Medium | VDN | 5 | 0.9 | 0.99 | 0.06 |
| 2-Ant | M-Replay | VDN | 1 | 0.9 | 0.99 | 0.04 |
| MPE | Expert | VDN | 1 | 0.9 | 0.995 | 0.02 |
| MPE | Medium | VDN | 5 | 0.9 | 0.995 | 0.02 |
| MPE | M-Replay | VDN | 5 | 0.9 | 0.995 | 0.04 |
| MPE | Random | VDN | 5 | 0.9 | 0.995 | 0.02 |
| 2s3z | Good | VDN | 1 | 0.9 | 0.99 | 0.02 |
| 2s3z | Medium | VDN | 1 | 0.8 | 0.995 | 0.5 |
| 2s3z | Poor | VDN | 1 | 0.9 | 0.99 | 4.0 |
| 3m | Good | VDN | 1 | 0.9 | 0.99 | 0.08 |
| 3m | Medium | VDN | 1 | 0.9 | 0.99 | 0.08 |
| 3m | Poor | VDN | 1 | 0.9 | 0.99 | 2.0 |
| 5m_vs_6m | Good | VDN | 1 | 0.9 | 0.99 | 0.008 |
| 5m_vs_6m | Medium | VDN | 1 | 0.9 | 0.99 | 0.5 |
| 5m_vs_6m | Poor | VDN | 1 | 0.9 | 0.99 | 0.08 |
| 8m | Good | VDN | 1 | 0.9 | 0.99 | 0.08 |
| 8m | Medium | VDN | 1 | 0.9 | 0.99 | 0.02 |
| 8m | Poor | VDN | 1 | 0.8 | 0.99 | 2.0 |

Table 20: Domain-specific hyperparameters for goal-conditioned MARL.

| Method | Hyperparameters | Value |
|---|---|---|
| GCMADiff | Planning window | 256 for medium maze and 384 for large maze |
| GCMADiff | Diffusion steps | 200 |
| GCMADiff | DDIM steps | 50 |
| GCMAC-Flow | Distillation coefficient | 0.3 |
| GCMAC-Flow | Q-aggregation | mean |

Table 21: Guidance-weight ($\beta$) Tuning Range used at evaluation.

| Domain | Range |
|---|---|
| MA-MuJoCo / MPE | $[0, 0.3]$ |
| SMACv1 | $[0, 4.0]$ |
| AntMaze | $[0, 3.0]$ |

## J LIMITATIONS

CQF is, to our knowledge, the first decentralized test-time gradient-guidance framework for cooperative offline MARL: it keeps behavior modeling supervised and injects locally queryable utility gradients shaped by a team-level critic. While these properties are effective, two limitations remain. First, Proposition 1 is a first-order statement about the learned factorized critic rather than a guarantee that simultaneous local guidance increases the true team return. Second, to preserve decentralized deployment the behavior flow is trained independently, which may represent the joint data distribution less faithfully as the number of agents grows. Because CQF only improves and injects coordination on top of this frozen reference, a weakly fit flow leaves less headroom for guidance, as on 6-HalfCheetah Expert and Medium-Expert and SMACv1 8m Poor. A centralized joint flow is not necessarily a remedy: Table 10 shows that it underperforms CQF as the number of agents grows (6-HalfCheetah, 3-Hopper). Learning a stronger locally executable reference, without giving up this training-time decoupling, is a natural direction for future work.