BEYOND MIMICRY: A VALUE-BASED APPROACH TO TASK-EFFICIENT OFFLINE MARL

Anonymous authors
Paper under double-blind review

ABSTRACT

Offline Multi-Agent Reinforcement Learning (MARL) offers a vital path to complex coordination without the risks of real-world interaction. However, most existing methods are task-specific, requiring retraining for new tasks, leading to redundancy and inefficiency. To address this issue, we propose a task-efficient value-based multi-task offline MARL algorithm, Task-efficient Skill-based Q-learning (TasQ). Unlike existing methods that rely on behavior cloning for skill execution, TasQ adopts a value-centric framework. It discovers skills in a latent space by reconstructing the next observation, encouraging the extraction of task-agnostic features that transfer across tasks. It then evaluates fixed and variable actions separately, and uses conservative Q-learning with local value calibration to select the optimal action for each skill, enabling strong multi-task generalization from limited source tasks. Substantial experiments spanning homogeneous and heterogeneous agents, single and multiple source tasks, and discrete and continuous actions demonstrate the superior generalization and task-efficiency of TasQ. It achieves the best performance on **17** out of 18 task sets, with up to **68.9%** improvement on individual task sets.

1 INTRODUCTION

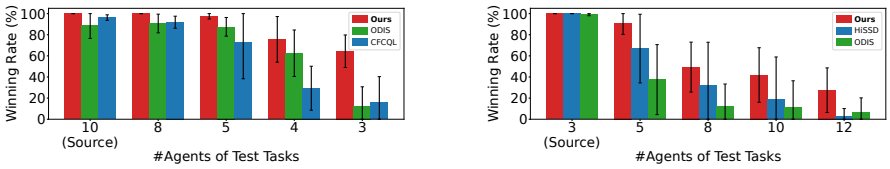
Multi-agent reinforcement learning (MARL) provides a principled framework for learning coordinated behaviors among multiple agents (Shoham & Leyton-Brown, 2008), with successful applications in games (Deng et al., 2026), autonomous systems (Yang et al., 2025), and finance (Li et al., 2025b). However, online interaction can be prohibitively expensive, risky, or infeasible in complex real-world systems. This motivates offline MARL, where collaborative policies are learned solely from previously collected data. Yet most existing offline MARL methods remain task-specific (Li et al., 2023; Shao et al., 2023; Liu et al., 2024b): they optimize policies for the source tasks represented in the dataset and often require retraining when task configurations change.

Therefore, a more generalizable paradigm, namely multi-task offline MARL, is highly desirable. Under this paradigm, agents learn from offline datasets collected on only a few source tasks and directly generalize to unseen but related tasks without access to additional interaction or target-task data. Yet, in many practical scenarios, collecting offline data from a large number of source tasks is itself costly or infeasible. This further motivates ***task-efficiency***, which we define as achieving reasonable generalization across diverse unseen tasks using as few source tasks as possible.

However, achieving task-efficient multi-task offline MARL entails two key challenges: (i) mitigating uncertainty and extrapolation errors caused by limited offline data coverage within each source task; and (ii) transferring effective decision-making structures to unseen tasks without access to their data or dynamics. Although conservatism in offline MARL can mitigate the former, these task-specific methods typically overfit to source tasks and are therefore limited in addressing the latter.

Figure 1a shows this challenge: CFCQL (Shao et al., 2023), a representative conservative offline MARL method, is trained on a source task with 10 agents but degrades shapely when evaluated on tasks with fewer agents. This suggests that low-level policies learned for a specific task are difficult to transfer directly, motivating the extraction of reusable high-level decision patterns. Skill discovery methods, e.g., ODIS (Zhang et al., 2023a) and HiSSD (Liu et al., 2025), therefore provide a natural

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.



(a) Trained on *10m*, tested from *10m* to *3m*.

(b) Trained on *3m*, tested from *3m* to *12m*.

Figure 1: Skill discovery enhances cross-task generalization (1a), but existing methods struggle with limited source tasks (1b). Error bars: ± 1 std.

mechanism for facilitating such transfer: agents identify distinguishable behavioral patterns, referred to as "skills,"[1] from source tasks and reuse them to guide action selection in unseen tasks.

It also shows that existing offline skill-discovery methods, such as ODIS, transfer more effectively than conventional offline MARL methods, demonstrating the benefit of reusable skills. However, Figure 1b reveals that they remain inadequate when only a few small-scale source tasks are available.

We identify the use of behavior cloning (BC) for action decoding as a key cause of this limited generalization. Even when an appropriate skill is identified, BC merely reproduces the action distribution in the offline dataset rather than selecting actions that maximize returns. We term this discrepancy the ***Skill-Action Optimality Mismatch***. Value-based optimization offers a promising remedy, but whether it can support scalable and generalizable learning in task-efficient multi-task offline MARL remains unclear, as value-estimation errors may accumulate across agents and be further amplified when learning jointly from heterogeneous source tasks. This motivates us to ask:

Can we combine the strengths of skill discovery and value-based methods to enable strong cross-task performance with high task-efficiency?

We answer affirmatively by proposing **Ta**sk-efficient **S**kill-based **Q**-learning (**TasQ**). Unlike existing methods that rely on global information or predefine the number of skills, TasQ learns continuous skill representations from only local information through next-observation reconstruction, encouraging task-agnostic features that transfer across tasks. It then learns separate skill-conditioned Q-networks for fixed and variable action sets via conservative Q-learning, together with local value calibration to mitigate value-estimation errors arising from variations in agent scale. Extensive experiments across homogeneous and heterogeneous agents, single- and multi-source-task settings, and discrete and continuous action spaces show that value-based skill execution is both viable and superior to existing skill-based approaches.

In summary, the key contributions of this work are:

- We study the underexplored yet vital problem of task-efficient multi-task offline MARL, and identify the skill-action optimality mismatch, where BC-based execution still pairs a correct skill with suboptimal actions, limiting existing methods' cross-task generalization.
- We propose TasQ, which discovers skills by reconstructing the next observation, evaluates fixed and variable actions separately, and executes optimal actions via conservative Q-learning with local value calibration for strong generalization from limited source tasks.
- Comprehensive experiments across diverse settings show that TasQ performs best in **17** out of 18 task sets, with up to **68.9%** improvement on individual task sets.

2 Background

2.1 Multi-task Offline MARL

A cooperative MARL task, indexed by m, can be formulated as a decentralized partially observable Markov decision process (Dec-POMDP) (Oliehoek et al., 2016). The Dec-POMDP for task m is defined by a tuple $(\mathcal{N}_m, \mathcal{S}_m, \mathcal{O}_m, \mathcal{A}_m, P_m, R_m, \gamma)$, where $\mathcal{N}_m$ is the set of agents, $\mathcal{S}_m$ is the set of states, $\mathcal{O}_m$ is the joint observation space, $\mathcal{A}_m$ is the joint action space, P_m is the transition

[1]This terminology follows prior work (Zhang et al., 2023a; Liu et al., 2025).

probability (i.e., the probability of transitioning to the next state given the current state and joint action), R_m is the immediate reward shared by all agents, and γ is the discount factor.

The multi-task MARL problem is a collection of such MARL tasks, which can be represented by the tuple $(\mathcal{N}, \mathcal{S}, \mathcal{O}, \mathcal{A}, P, R, \gamma, \mathbb{T})$. Here, $\mathbb{T}$ is the set of tasks, γ is the discount factor shared by all tasks, and the remaining elements are the union of corresponding elements across all tasks. For example, $\mathcal{N} = \bigcup_{m\in\mathbb{T}} \mathcal{N}_m$ represents the union of the sets of agents across all tasks.

For the task m, each agent maintains its observation-action history $\tau^i \in \mathcal{T}_m^i$, and the corresponding joint history is denoted by $\tau \in \mathcal{T}_m$. $\mathcal{T} = \bigcup_{m\in\mathbb{T}} \mathcal{T}_m$ represents the collection of observation-action histories across all tasks, and the goal is to find a joint policy $\pi : \mathcal{T} \to \mathcal{A}$ that maximizes the expected discounted return average over all tasks: $\mathcal{J} = \mathbb{E}_{\pi,\mathbb{T}}\left[\sum_{t=0}^{T} \gamma^t \cdot r_m(s_t^m, \boldsymbol{a}_t^m)\right]$, where T is the time horizon and $r_m(s_t^m, \boldsymbol{a}_t^m)$ is the reward for taking joint action $\boldsymbol{a}_t^m$ at state s_t^m in task m.

In offline setting, agents can only access the dataset $\mathcal{D} = \{\mathcal{D}_m\}_{m\in\mathbb{T}_s}$, where $\mathbb{T}_s \subset \mathbb{T}$ is *source tasks* (the complement referred to as *unseen*), without any interaction with the environment. The goal is still to maximize $\mathcal{J}$. Appendix I.5 compares multi-task learning in MARL and conventional RL.

2.2 CTDE Framework, QMIX and CQL

A common framework for cooperative MARL is Centralized Training for Decentralized Execution (CTDE). In this framework, agents can leverage centralized information during training but must rely solely on their local observations during execution. In this work, we adopt QMIX (Rashid et al., 2020) as our backbone algorithm, which is one of the most popular discrete-action CTDE algorithms. In principle, our method can be applied to any value-based algorithm.

In QMIX, each agent maintains an individual Q-function $Q_i(\tau^i, a^i)$, which is conditioned on its own observation-action history τ^i and action a^i. Then, it calculates a joint Q-function $Q_{tot}(\tau, \boldsymbol{a})$ from individual Q-functions through a mixing network f_s, such that:

$$Q_{\text{tot}}(\tau, \boldsymbol{a}) = f_s(Q_1(\tau^1, a^1), \ldots, Q_n(\tau^n, a^n)). \tag{1}$$

The mixing network is designed to satisfy the monotonicity constraint, ensuring that the partial derivative of Q_{tot} with respect to each Q_i is non-negative. To train the Q-function, QMIX minimizes the temporal-difference error on Q_{tot}.

In offline MARL, the lack of interaction may cause Q-value overestimation on out-of-distribution (OOD) samples. Thus, Conservative Q-learning (CQL) (Kumar et al., 2020) uses a regularization term to enhance the Q-value of in-dataset samples while suppressing that of OOD ones:

$$\mathcal{L}_{\text{CQL}} = \mathbb{E}_{\tau\sim\mathcal{D},\boldsymbol{a}\sim\mu}\left[Q_{tot}(\tau, \boldsymbol{a})\right] - \mathbb{E}_{(\tau,\boldsymbol{a})\sim\mathcal{D}}\left[Q_{tot}(\tau, \boldsymbol{a})\right] \tag{2}$$

where μ denotes a sampling distribution.

3 Related Work

In contrast, TasQ discovers skills entirely from local observations and directly optimizes skill-conditioned Q-values, achieving strong generalization and task-efficiency with a lightweight architecture. A comprehensive discussion is provided in Appendices A and I.

4 Theoretical Motivation

To isolate the key mechanism underlying TasQ from multi-agent complexities, we analyze it in the single-agent MDP setting with deterministic policies. This simplified setting clarifies the core difference between BC-based and value-based skill execution, motivating the design of TasQ for multi-agent learning. Proofs are provided in Appendix B.

Consider an infinite-horizon MDP $(\mathcal{S}, \mathcal{A}, P, r, \gamma, \mu_0)$ with finite discrete actions, the initial distribution μ_0, and the discount factor $\gamma \in (0, 1)$. For any deterministic policy $\pi : \mathcal{S} \to \mathcal{A}$, define its discounted state-occupancy measure $d^\pi(s) := (1-\gamma)\sum_{t=0}^{\infty} \gamma^t \Pr(s_t = s|\pi, \mu_0)$ and its expected return $\mathcal{J}(\pi) = \frac{1}{1-\gamma}\mathbb{E}_{s\sim d^\pi}[r(s, \pi(s))]$.

Let $\mathcal{Z}$ be the set of latent skill representation (e.g., retreat or attack), and $\mathring{\pi} : \mathcal{S} \times \mathcal{Z} \to \mathcal{A}$ be a deterministic skill-based policy. Then, for each skill $z \in \mathcal{Z}$, we define the feasible action set $\mathcal{A}_z$ containing all the actions belonging to the skill z, and assume that $\forall z \in \mathcal{Z},\ \mathcal{A}_z \neq \emptyset$. In a skill-based method, we first train a deterministic skill selector $\phi : \mathcal{S} \to \mathcal{Z}$, and then learn a ϕ-constrained skill-based policy $\mathring{\pi}(s, \phi(s))$, which only executes actions in $\mathcal{A}_{\phi(s)}$. Then, the set of all ϕ-constrained policies is defined as $\Pi_\phi := \{\pi | \exists \mathring{\pi},\ \forall s \in \mathcal{S},\ \pi(s) = \mathring{\pi}(s, \phi(s))\}$, and the optimal ϕ-constrained policy is $\pi^*_\phi = \arg\max_{\pi \in \Pi_\phi} \mathcal{J}(\pi)$.

In the offline setting, the dataset $\mathcal{D} = \{(s, a, s', r)\}$ can be viewed as being collected from an unknown behavior policy $\beta : \mathcal{S} \to \Delta(\mathcal{A})$. To isolate the impact of different policy learning schemes, we assume access to an optimal skill selector ϕ^* such that for any other $\widetilde{\phi}$, $\sup_{\pi \in \Pi_{\phi^*}} \mathcal{J}(\pi) \geq \sup_{\pi \in \Pi_{\widetilde{\phi}}} \mathcal{J}(\pi)$. Based on ϕ^*, the behavior action is defined as $a_{\mathrm{m}}(s) = \arg\max_{a \in \mathcal{A}_{\phi^*(s)}} \beta(a|s)$.

Now consider two ϕ^*-constrained policies. The first is BC-style: $\hat{\pi}_{\mathrm{BC}}(s) := \arg\max_{a \in \mathcal{A}_{\phi^*(s)}} \beta_{\mathrm{BC}}(a|s)$, where $\beta_{\mathrm{BC}}(a|s) := \arg\max_{\hat{\beta}} \mathbb{E}_{(s,a) \in \mathcal{D}}[\log \hat{\beta}(a|s)]$. The second is value-based, which first learns a skill-conditioned Q-value $\hat{Q}(s, a, \phi^*(s))$ via CQL, and executes the action with maximal Q-value: $\hat{\pi}_{\mathrm{CQL}}(s) = \arg\max_{a \in \mathcal{A}_{\phi^*(s)}} \hat{Q}(s, a, \phi^*(s))$.

As is standard, we assume the reward function is bounded (Kumar et al., 2020; Yu et al., 2020). To characterize the accuracy of the Q-values, we further assume its estimation error is also bounded.

Assumption 4.1 (Bounded Reward)**.** There exists $R_{\max} \geq 0$ such that for all $(s, a) \in \mathcal{S} \times \mathcal{A}$, $|r(s, a)| \leq R_{\max}$. Then, for any policy π, $\|Q^\pi\|_\infty \leq Q_{\max} := \frac{R_{\max}}{1-\gamma}$.

Assumption 4.2 (Bounded Q-value Error)**.** There exists $\varepsilon_Q \geq 0$ such that for all $s \in \mathcal{S}$ and all $a \in \mathcal{A}_{\phi^*(s)}$, $\left|\hat{Q}(s, a, \phi^*(s)) - Q^{\pi^*_{\phi^*}}(s, a)\right| \leq \varepsilon_Q$.

Let $\pi^* = \arg\max_\pi \mathcal{J}(\pi)$ be the unconstrained optimal policy. Define $\varepsilon_{\mathrm{rep}} := \mathcal{J}(\pi^*) - \mathcal{J}(\pi^*_{\phi^*}) \geq 0$, which captures the unavoidable loss induced by restricting to Π_{ϕ^*}. Then, the following theorem presents the performance gaps of $\hat{\pi}_{\mathrm{BC}}$ and $\hat{\pi}_{\mathrm{CQL}}$ relative to π^*.

Theorem 4.3. *Under Assumptions 4.1–4.2, we have* $\mathcal{J}(\pi^*) - \mathcal{J}(\hat{\pi}_{\mathrm{BC}}) \leq \varepsilon_{\mathrm{rep}} + \frac{\varepsilon_{BC}}{1-\gamma} + \frac{2Q_{\max}\kappa}{1-\gamma}$, *and* $\mathcal{J}(\pi^*) - \mathcal{J}(\hat{\pi}_{\mathrm{CQL}}) \leq \varepsilon_{\mathrm{rep}} + \frac{2\varepsilon_Q}{1-\gamma}$, *where* $\varepsilon_{BC} = \sup_{s \in \mathcal{S}} |Q^{\pi^*_{\phi^*}}(s, \pi^*_{\phi^*}(s)) - Q^{\pi^*_{\phi^*}}(s, a_{\mathrm{m}}(s))|$ *and* $\kappa = \mathbb{E}_{s \sim d^{\hat{\pi}_{\mathrm{BC}}}}[\mathbb{1}\{\hat{\pi}_{\mathrm{BC}}(s) \neq a_{\mathrm{m}}(s)\}]$.

Remark: *In the bound of* $\hat{\pi}_{BC}$, ε_{BC} *captures the loss incurred by the suboptimality of the behavior policy, whereas* κ *characterizes the estimation error of the behavior policy due to finite sampling.*

Although Theorem 4.3 provides upper bounds, it offers meaningful insights. When the offline dataset has broad coverage, κ can approach zero through sufficient learning. However, if the behavior policy β is suboptimal, $\varepsilon_{BC} > 0$ remains, making the actual performance gap also larger than $\varepsilon_{\mathrm{rep}}$, termed the *Skill–Action Optimality Mismatch*. In contrast, value-based methods with proper importance weighting and conservative regularization can reduce ε_Q to zero, yielding policies that could surpass BC-style baselines.

Therefore, value-based methods can be more effective for offline learning even when behavior policies are imperfect. Motivated by this, we aim to design a new value-based skill-discovery method tailored to multi-task offline MARL.

5 Task-efficient Skill-based Q-learning

To develop scalable policies that can generalize to varying unseen tasks through data from only a few source tasks, we propose Task-efficient Skill-based Q-learning (TasQ), a task-efficient method for multi-task offline MARL. As illustrated in Figure 2, TasQ discovers cross-task generalizable skills in the latent space through observation reconstruction, and then trains scalable policies via multi-task CQL and local value calibration. This approach mitigates distributional shifts and error accumulation in offline MARL, as well as the generalization challenges inherent in multi-task learning. We detail each component below and compare architectures with existing methods in Appendix I.2. Pseudocode is provided in Algorithm 1 in Appendix C.

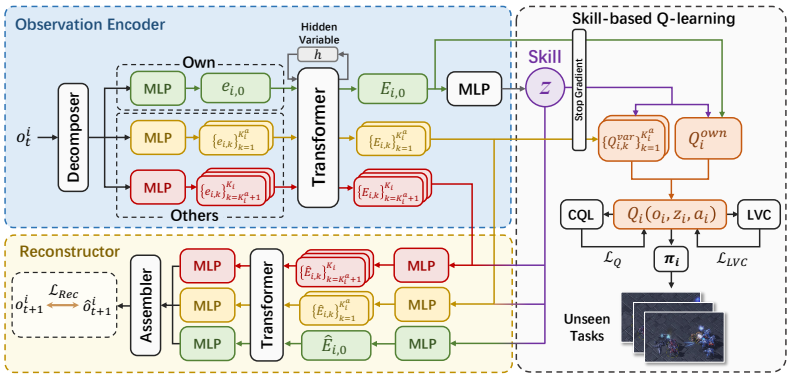


Figure 2: TasQ framework. For the i-th agent, the encoder splits the current o_t into $K_i + 1$ entities, embedding them into same dimensions, with the 0-th representing the agent itself. The skill z is extracted from $E_{i,0}$, the embedding associated with the agent itself. Finally, TasQ injects temporal behavior into z by reconstructing o_{t+1}. Meanwhile, it computes the z-conditioned Q-values for fixed and variable actions separately, and optimizes scalable polices via CQL and local value calibration.

5.1 Observation Reconstruction

To improve cross-task generalization, a key idea is to enable agents to learn decision features shared across tasks, i.e., "skills", from offline data. Prior methods extract skills using global information (e.g., states) (Zhang et al., 2023a; Liu et al., 2025), which entangles skills with task-specific configurations. Instead, TasQ discovers skills by reconstructing the next *local* observation alone, yielding task-agnostic features that transfer across scales. More details are in Appendix D.3.

5.1.1 Observation Encoder

As the number of agents varies across tasks, the dimensionality of each agent's observation may change accordingly. To enable unified optimization across tasks, following standard practice (Liu et al., 2020; Zhang et al., 2023a; Wu et al., 2024), we decompose the i-th agent's observation o_t^i into $o_t^{i,0}$ (the agent itself) and $\{o_t^{i,k}\}_{k=1}^{K_i}$ (K_i observable entities, of which K_i^a are interactable). These are encoded into embeddings $\{e_{i,k}\}_{k=0}^{K_i}$ with consistent dimensions, and processed by a single-layer transformer (Vaswani, 2017) to produce the final encoding $\{E_{i,k}\}_{k=0}^{K_i}$.

We add a "pseudo-entity" $h_t = e_{i,K_i+1}$ as a hidden variable. It is represented in the same embedding space as other entities and participates in the encoder transformer, but is excluded from the decoder. At each time step, h_t is carried over from the previous step's transformer output, allowing it to aggregate temporal information across the trajectory and enabling the skill vector to capture extended behavioral patterns beyond single-step transitions. We also incorporate the previous actions a_{t-1} in o_t following standard practice, but exclude them from the reconstruction target o_{t+1}.

5.1.2 Observation Decoder

The decoder extracts a d-dimensional continuous latent skill vector $z \in \mathbb{R}^d$ from the agent's own embedding $E_{i,0}$, and reconstructs the next observation from z. Unlike discrete skill indices in prior work (Zhang et al., 2023a), a continuous latent space allows richer representations. Since the reconstruction target is action-agnostic, z captures higher-level spatiotemporal patterns rather than low-level details. Concretely, the decoder reconstructs embeddings $\{\widehat{E}_{i,k}\}_{k=0}^{K_i}$ from z, passes them through a single-layer transformer, restores entity vectors $\{\hat{o}_{i,k}\}_{k=0}^{K_i}$, and reassembles the next observation $\hat{o}_{t+1}^i$. The reconstruction loss is:

$$\mathcal{L}_{\text{Rec}} = \frac{1}{N}\sum_{i=1}^{N} \text{MSE}(\hat{o}_{t+1}^i, o_{t+1}^i) \tag{3}$$

where N is the number of controllable agents in the task, and $\text{MSE}(\cdot,\cdot)$ is the Mean Square Error.

Remark: *The encoder-only hidden state h_t, together with next-observation prediction, turns z into a temporal information bottleneck that summarizes behavioral patterns beyond purely observational features. In addition, inter-entity relations within local observations provide z with short-horizon cooperative cues. Team-level coordination is further enforced by the skill-conditioned Q_{tot} under CTDE. Section 6.3 shows that removing skill discovery severely hurts generalization.*

5.2 Skill-conditioned Policy Optimization

After identifying and selecting the appropriate skill, existing multi-task offline MARL algorithms primarily execute the corresponding actions through behavior cloning (BC). However, since a single skill often encompasses multiple specific actions, selecting the optimal skill does not necessarily ensure that all associated actions are optimal.

Therefore, we use conservative Q-learning to optimize the skill-conditioned policy $\pi(a|s, z)$, enabling value-based execution of the selected skill. To accommodate task-varying action spaces, TasQ separately evaluates task-shared fixed actions and task-dependent variable actions. It further introduces local value calibration to mitigate accumulated value-estimation errors and distributional shift, thereby improving training stability on larger-scale tasks.

5.2.1 Conservative Skill-conditioned Q-value

To handle the variation in the number of actions across different tasks, we utilize two separate Q-networks. One network, called Q^{own}, is responsible for a fixed set of actions related to the agent itself, while the other, called Q^{var}, handles a variable set of actions associated with other entities in its observation. Both networks receive the corresponding entity embeddings $E_{i,k}$ and skill vectors z as inputs and output the Q-values for their respective actions. The parameters of Q-value networks are shared by all the agents. Therefore, the individual Q-values for the i-th agent with skill z_i are:

$$Q_i(o_i, z_i, a) = \begin{cases} Q^{own}(E_{i,0},\ z_i, a) & \text{if } a \in \mathcal{A}_i^{own} \\ Q^{var}(E_{i,k},\ z_i, a) & \text{if } a \in \mathcal{A}_i^{k} \end{cases} \tag{4}$$

where $\mathcal{A}_i^{own}$ is the set of actions related only to the i-th agent itself, and $\mathcal{A}_i^k$ is the set of actions related to the k-th interactable entity for agent i. Since Q^{var} evaluates each entity independently, the network naturally scales to any number of entities across tasks. While similar decompositions have been adopted in previous work (Lin & Lee, 2024), our use is primarily motivated by accommodating the varying action dimensions inherent in multi-task offline MARL.

We apply stop-gradient to the embeddings and skill vectors before feeding them into the Q-network. This decouples skill discovery from policy optimization, preventing Q-learning gradients from distorting the skill representations.

Finally, a mixing network aggregates the individual Q-values into the global Q-value Q_{tot} via Equation equation 1. Then, we can optimize Q_{tot} via the temporal difference loss:

$$\mathcal{L}_{\text{TD}} = \left[Q_{\text{tot}}(\tau, \boldsymbol{z}, \boldsymbol{a}) - \left(r + \gamma \max_{\boldsymbol{a}'} \bar{Q}_{\text{tot}}(\tau', \boldsymbol{z}', \boldsymbol{a}') \right) \right]^2 \tag{5}$$

where τ', $\boldsymbol{a}'$ and $\boldsymbol{z}'$ denote the next observation-action history, joint action, and joint skill, respectively, and $\bar{Q}_{\text{tot}}$ represents the target joint action-value function.

In the offline setting, we apply the CQL regularizer in Equation equation 2 to $Q_{\text{tot}}(\tau, \boldsymbol{z}, \boldsymbol{a})$. The Q-learning objective is:

$$\mathcal{L}_{\text{Q}} = \mathcal{L}_{\text{TD}} + \alpha \mathcal{L}_{\text{CQL}} \tag{6}$$

where $\alpha > 0$ is a hyperparameter to control conservatism.

5.2.2 Local Value Calibration

Beyond distributional shifts, value-estimation errors can accumulate across agents (Pan et al., 2021) and could be amplified when training jointly on heterogeneous tasks with varying numbers of agents. Unlike existing task-specific methods (Fujimoto & Gu, 2021; Pan et al., 2022), we propose *Local*

Value Calibration (LVC) to calibrate each agent's local action-value estimates by increasing the relative value of the logged action over alternative available actions under different agent scales:

$$\mathcal{L}_{\text{LVC}} = \frac{1}{N}\sum_{i=1}^{N} \text{CE}\left(\text{Softmax}(Q_i),\ \beta_i\right) \tag{7}$$

where $\text{CE}(\cdot,\cdot)$ is the cross-entropy loss, N is the number of agents in the current task instance, Q_i is the individual Q-values of all the actions available for the i-th agent, and β_i is the corresponding one-hot action in the offline datasets. Then, the total loss function for TasQ is:

$$\mathcal{L} = (1-\eta)\cdot\mathcal{L}_{\text{Q}} + \eta\cdot\mathcal{L}_{\text{LVC}} + \mathcal{L}_{\text{Rec}} \tag{8}$$

where $\eta \in [0,1]$ controls the strength of value calibration.

6 Experiments

We evaluate TasQ on multiple task sets based on SMAC (Samvelyan et al., 2019), SMACv2 (Ellis et al., 2023), and MPE (Lowe et al., 2017), to address: (i) Does TasQ outperform existing methods across diverse multi-task offline MARL settings (Sec. 6.2)? (ii) Are TasQ's components critical to its performance (Sec. 6.3)? (iii) Do skills capture meaningful decision-making features (Sec. 6.4)?

6.1 Setup

Datasets. Following the definition by D4RL (Fu et al., 2020), our experiments utilize datasets of four quality: *Expert*, *Medium*, *Medium-Replay*, and *Medium-Expert*. For fair comparisons, we use the datasets provided by ODIS (Zhang et al., 2023a). Details are provided in Appendix D.1.

Task Sets. We construct 18 task sets across two groups: (i) Multi-to-Multi (training on multiple tasks, testing on multiple tasks), following ODIS, covering *Marine-Easy*, *Marine-Hard*, and *Stalker-Zealot* (heterogeneous agents) scenarios with four dataset qualities each; and (ii) One-to-Multi (training on one task, testing on multiple tasks), newly designed to evaluate task-efficiency, covering *Marine-Single*, *Marine-Single-Inv*, *Marine-Single-Mid*, and *Marine-Single-Large* scenarios. We further include SMACv2 (Ellis et al., 2023) and continuous-action MPE (Lowe et al., 2017) task sets into One-to-Multi group, to evaluate generalization under randomized configurations and continuous action spaces. More details are included in Appendices D.2, G, and H.

Baselines. We compare against BC-t (BC with a multi-task transformer), BC-r (BC-t augmented with return-to-go input), CQL (Kumar et al., 2020) with a multi-task transformer, ODIS (Zhang et al., 2023a), and HiSSD (Liu et al., 2025). Implementation details are provided in Appendix D.3.

Experiment Setup. All algorithms are built on PYMARL2 (Hu et al., 2023) and the ODIS codebase, and run $35{,}000$ steps of multi-task offline training (ODIS additionally requires $15{,}000$ pre-training steps). We report the average performance of the final policy across five random seeds. Hyperparameters and learning curves are in Appendices D.4 and K.

6.2 Evaluation

6.2.1 Multi-to-Multi

These task sets include *Marine-Easy*, *Marine-Hard*, and *Stalker-Zealot*, each comprising 3 source tasks of varying scales and several unseen tasks. Agents are trained offline using datasets from the source tasks and are tested across all tasks, with zero-shot testing conducted on the unseen tasks. For each task set, we set up four datasets of different qualities. We report the average performance of the algorithms across all tasks for each task set and dataset in Table 1, where BC-best denotes the better result between BC-t and BC-r. Detailed results are in Appendix J.1.

The results show that TasQ achieves the best performance for 11 out of 12 task sets, with the remaining one being very close to the best baseline (within 1.4%). The detailed results in Appendix J further show that TasQ stays competitive on source tasks while achieving stronger zero-shot generalization on unseen tasks. In particular, on heterogeneous task sets such as *Stalker-Zealot-M* and *Stalker-Zealot-ME*, TasQ significantly outperforms other methods, with improvements of 68.9% and 66.2%, respectively, showing its superior generalization across various dataset qualities.

Table 1: Win rates (%) of Multi-to-Multi task sets. The results are the average performance over all source and unseen tasks (10 tasks for *marine-easy*, 12 for *marine-hard*, and 13 for *stalker-zealot*), and are averaged over 5 random seeds. Bold indicates the best, and ± denotes one standard deviation.

Task Set	BC-best*	CQL	ODIS	HiSSD	TasQ (Ours)
marine-easy-e	**99.38** ± 0.40	31.13 ± 6.36	70.50 ± 30.14	97.94 ± 1.23	98.06 ± 1.00
marine-hard-e	64.84 ± 3.24	24.84 ± 1.53	21.09 ± 14.72	68.02 ± 4.47	**70.68** ± 4.80
stalker-zealot-e	68.61 ± 3.93	10.62 ± 5.88	55.05 ± 9.11	66.59 ± 7.77	**75.72** ± 3.73
marine-easy-m	71.81 ± 3.74	17.38 ± 7.31	68.50 ± 5.74	71.44 ± 2.77	**75.81** ± 1.73
marine-hard-m	42.08 ± 2.63	9.17 ± 1.78	32.55 ± 4.31	47.60 ± 2.66	**48.12** ± 5.66
stalker-zealot-m	23.94 ± 2.72	15.29 ± 4.89	11.97 ± 7.53	23.51 ± 3.76	**40.43** ± 7.06
marine-easy-mr	49.62 ± 9.17	29.56 ± 11.73	11.06 ± 8.94	72.88 ± 3.16	**75.81** ± 6.83
marine-hard-mr	46.98 ± 1.95	21.67 ± 2.83	37.29 ± 5.48	47.55 ± 3.56	**49.84** ± 3.36
stalker-zealot-mr	17.64 ± 4.40	14.57 ± 2.02	8.17 ± 6.08	15.00 ± 4.02	**23.08** ± 3.38
marine-easy-me	75.25 ± 7.41	33.56 ± 6.47	66.00 ± 9.21	78.31 ± 4.99	**86.88** ± 5.68
marine-hard-me	51.93 ± 7.21	20.31 ± 1.97	19.58 ± 17.63	53.28 ± 1.88	**56.20** ± 3.21
stalker-zealot-me	39.62 ± 2.98	13.41 ± 2.53	30.72 ± 14.65	40.29 ± 4.64	**66.97** ± 7.01

-e, *-m*, *-mr*, and *-me* represent *-expert*, *-medium*, *-medium-replay*, and *-medium-expert*, respectively.
* BC-best represents the better average performance between BC-r and BC-t.

Table 2: Results of One-to-Multi task sets. Win rates (%) are for discrete tasks and average returns for continuous tasks. The results are the average performance over all source and unseen tasks, and are averaged over 5 random seeds. Bold indicates the best, and ± denotes one standard deviation.

Task Set	BC-best*	CQL	ODIS	HiSSD	TasQ (Ours)
marine-single	50.12 ± 9.65	56.00 ± 11.07	33.25 ± 16.50	44.12 ± 18.89	**62.00** ± 13.80
marine-single-inv	48.50 ± 17.09	2.75 ± 5.19	68.38 ± 11.47	86.88 ± 4.49	**87.50** ± 6.54
marine-single-mid	62.88 ± 22.12	75.12 ± 16.84	57.75 ± 8.54	46.25 ± 11.77	**85.12** ± 9.23
marine-single-large	68.38 ± 6.60	0.00 ± 0.00	60.13 ± 3.07	OOM**	**70.38** ± 4.57
SMACv2	30.00 ± 3.07	34.06 ± 9.09	20.16 ± 8.23	30.00 ± 4.70	**34.38** ± 3.46
Continuous-MPE	-327.00 ± 25.77	-895.04 ± 117.72	-309.41 ± 9.99	-326.31 ± 16.15	**-299.22** ± 14.37

* BC-best represents the better average performance between BC-r and BC-t.
** OOM indicates that the method runs out of memory on our GPUs.

6.2.2 One-to-Multi

All One-to-Multi task sets train on a single source task and test across a range of scales. *Marine-Single* trains on *3m* and tests up to *12m*; *Marine-Single-Inv* trains on *10m* and tests down to *3m*; *Marine-Single-Mid* trains on *5m* (a mid-scale source) and tests up to *12m*; and *Marine-Single-Large* trains on *25m* and tests on tasks ranging from *3m* to *30m*. *SMACv2* trains with 3 agents per side and tests up to 10; *Continuous-MPE* trains with 2 agents and tests up to 5. We use expert datasets for the 4 Marine task sets, and medium datasets for *SMACv2* and *Continuous-MPE*. We report the average performance across all test tasks in Table 2. Per-task results are in Appendices G, H, and J.2.

As shown in Table 2, TasQ achieves the best average performance across all six task sets. On the four SMAC task sets, TasQ consistently ranks first regardless of whether the source task is small (*Single*, *Single-Mid*), large (*Single-Large*), or whether agents generalize to larger or smaller scales (*Single-Inv*). In contrast, CQL collapses when the search space grows large (*Single-Inv* and *Single-Large*), and BC-based methods remain limited in overall performance due to imitating frequent actions rather than learning transferable skills. ODIS's skill mechanism partially alleviates this, but its BC-based action decoder still constrains generalization. HiSSD is competitive on some task sets but runs out of memory on the *Single-Large* due to its large parameter overhead.

Results on SMACv2 and continuous-action MPE further highlight the instability of existing methods: CQL is competitive on SMACv2 but collapses on MPE, while BC-based methods show the opposite trend. Neither paradigm alone generalizes reliably across environments. TasQ bridges both gaps by combining skill discovery for task-agnostic representations with LVC-regularized Q-



(a) Training TasQ in *3m*. (b) Testing TasQ in *12m*. (c) Clusters of skills. (d) Skill self-similarity.

Figure 4: (a-b) TasQ learns two skills, *retreat* and *attack*, from the source *3m* and transfers them to the unseen *12m*. (c) TasQ's skill cluster by skill type across tasks. (d) TasQ's skill self-similarity cos-sim(z_t, z_{t+1}) for one agent over a *12m* episode, high within a phase and dropping at transitions.

learning for robust value estimation, achieving consistent superiority across discrete and continuous action spaces, diverse source-task scales, and varying environment dynamics.

6.3 Ablation Study

We ablate TasQ's key components on *Marine-Single* and *Marine-Single-Inv* with 4 variants: (i) w/o LVC: removing LVC and allowing gradient flow between skill discovery and Q-learning; (ii) w/o TD: removing the Q-learning objective (i.e., η=1); (iii) w/o SD: removing skill discovery; and (iv) w/o SD & LVC: i.e., vanilla CQL. In Figure 3, skill discovery (SD) is the primary contributor: removing it causes the largest drop on *Marine-Single* and severely degrades generalization on both task sets. Meanwhile, TD learning and LVC play complementary roles: w/o LVC performs comparably on forward generalization (*Marine-Single*) but collapses on inverse generalization (*Marine-Single-Inv*), consistent with vanilla CQL's failure in this setting. Conversely, w/o TD retains reasonable inverse performance but is still significantly below TasQ, and degrades notably on forward generalization. We further validate more design choices and analyze hyperparameter sensitivity in Appendix E. These confirm that every component is indispensable, and TasQ's superiority arises from the architectural coordination of all modules rather than any single one.

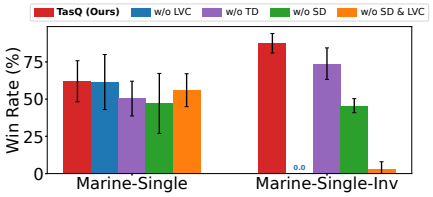


Figure 3: Component ablation. All components matter, and TasQ's gains come from their synergy.

6.4 Skill-Discovery Visualization

As shown in Figure 4, TasQ acquires two skills, *retreat* and *attack*, from the *3m* task (Figure 4a). When deployed in the unseen *12m* task (Figure 4b), TasQ agents *retreat* when health is low and *attack* when health is sufficient, with *attack* encompassing two specific actions: firing and advancing. The skill representations of TasQ (Figure 4c) cluster by skill type regardless of the task, confirming that the learned skills transfer across scales. Figure 4d further tracks the skill self-similarity cos-sim(z_t, z_{t+1}) of a single agent over a *12m* episode: it stays high within a behavioral phase, even as concrete actions vary, and drops sharply at phase transitions, confirming that z encodes high-level behavioral modes rather than observation representations, and remains meaningful on the unseen task. More results are in Appendix F, including a mutual information analysis in Appendix F.4.

7 Conclusion

In this paper, we study a rarely explored yet important field, multi-task offline MARL, and propose a novel task-efficient algorithm, TasQ. It discovers skills offline via next-observation reconstruction and trains scalable policies via conservative Q-learning with local value calibration. We conduct substantial experiments across diverse settings to present the superior generalization and task-efficiency of TasQ: It achieves the best performance on **17** out of 18 task sets, with up to **68.9%** improvement on individual task sets, demonstrating value-based methods are not only viable but superior in this setting. Future work includes alternative skill-discovery methods and online fine-tuning.

AI USE STATEMENT

In this work, we used generative AI tools for providing feedback on research methodology and experiments, formulating mathematical claims, assisting in the writing of proofs, and assisting with translation. We have not used generative AI tools for helping develop theoretical models or conceptual frameworks, providing critical ingredients for proving mathematical claims, proposing or refining hypotheses, designing research methodology or experiments, and supporting qualitative and thematic data analysis, interpreting results, and generating synthetic data sets, and cleaning and reformatting dataset are not applicable to this work. Additionally, we used generative AI tools for creating or editing software code, drafting parts of a research paper, summarizing existing literature, and identifying relevant literature. We have reviewed all AI-assisted work. LLM-generated code was verified and tested for correctness by 2 authors. LLM-generated text and LLM-suggested references were also checked manually. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Johannes Ackermann, Volker Gabler, Takayuki Osa, and Masashi Sugiyama. Reducing overestimation bias in multi-agent domains using double centralized critics, 2019.

Christopher Amato. An Introduction to Centralized Training for Decentralized Execution in Cooperative Multi-Agent Reinforcement Learning, 2024.

Jiayu Chen, Tian Lan, and Vaneet Aggarwal. Variational offline multi-agent skill discovery. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI-25*, pp. 4833–4841, 8 2025. doi: 10.24963/ijcai.2025/538. Main Track.

Christian Schroeder de Witt, Tarun Gupta, Denys Makoviichuk, Viktor Makoviychuk, Philip HS Torr, Mingfei Sun, and Shimon Whiteson. Is independent learning all you need in the starcraft multi-agent challenge? *arXiv preprint arXiv:2011.09533*, 2020.

Yue Deng, Weiyu Ma, Yuxin Fan, Ruyi Song, Yin Zhang, Haifeng Zhang, and Jian Zhao. Llm-smac: Solving multi-agent decision-making tasks via llm decision tree code generation. In *Proceedings of the 25th International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '26, pp. 3335–3337, Richland, SC, 2026. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 9798400723179. doi: 10.65109/RVAB7186. URL https://doi.org/10.65109/RVAB7186.

Benjamin Ellis, Jonathan Cook, Skander Moalla, Mikayel Samvelyan, Mingfei Sun, Anuj Mahajan, Jakob Nicolaus Foerster, and Shimon Whiteson. SMACv2: An improved benchmark for cooperative multi-agent reinforcement learning. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023.

Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4rl: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219*, 2020.

Scott Fujimoto and Shixiang (Shane) Gu. A Minimalist Approach to Offline Reinforcement Learning. In *Advances in Neural Information Processing Systems*, volume 34, pp. 20132–20145, 2021.

Jian Hu, Siying Wang, Siyang Jiang, and Weixun Wang. Rethinking the Implementation Tricks and Monotonicity Constraint in Cooperative Multi-agent Reinforcement Learning. In *ICLR Blogposts 2023*, 2023. https://iclr-blogposts.github.io/2023/blog/2023/riit/.

Siyi Hu, Fengda Zhu, Xiaojun Chang, and Xiaodan Liang. UPDeT: Universal Multi-agent RL via Policy Decoupling with Transformers. In *International Conference on Learning Representations*, 2021.

Jiwon Jeon, Myungsik Cho, and Youngchul Sung. STAIRS-former: Spatio-temporal attention with interleaved recursive structure transformer for offline mulit-task multi-agent reinforcement learning. In *The Fourteenth International Conference on Learning Representations*, 2026.

Jiechuan Jiang and Zongqing Lu. Offline decentralized multi-agent reinforcement learning. In *European Conference on Artificial Intelligence*, 2023.

Sham Kakade and John Langford. Approximately optimal approximate reinforcement learning. In *Proceedings of the nineteenth international conference on machine learning*, pp. 267–274, 2002.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Ilya Kostrikov, Rob Fergus, Jonathan Tompson, and Ofir Nachum. Offline reinforcement learning with fisher divergence critic regularization. In *International Conference on Machine Learning*, pp. 5774–5783. PMLR, 2021.

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline Reinforcement Learning with Implicit Q-Learning. In *International Conference on Learning Representations*, 2022.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative Q-Learning for Offline Reinforcement Learning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 1179–1191, 2020.

Tong Li, Chenjia Bai, Kang Xu, Chen Chu, Peican Zhu, and Zhen Wang. Skill matters: Dynamic skill learning for multi-agent cooperative reinforcement learning. *Neural Networks*, 181:106852, 2025a. ISSN 0893-6080. doi: https://doi.org/10.1016/j.neunet.2024.106852.

Xiangyu Li, Yawen Zeng, Xiaofen Xing, Jin Xu, and Xiangmin Xu. Hedgeagents: A balanced-aware multi-agent financial trading system. In *Companion Proceedings of the ACM on Web Conference 2025*, WWW '25, pp. 296–305, New York, NY, USA, 2025b. Association for Computing Machinery. ISBN 9798400713316. doi: 10.1145/3701716.3715232. URL https://doi.org/10.1145/3701716.3715232.

Zhuoran Li, Ling Pan, and Longbo Huang. Beyond conservatism: Diffusion policies in offline multi-agent reinforcement learning. *arXiv preprint arXiv:2307.01472*, 2023.

Bor-Jiun Lin and Chun-Yi Lee. HGAP: Boosting permutation invariant and permutation equivariant in multi-agent reinforcement learning via graph attention network. In *Forty-first International Conference on Machine Learning*, 2024.

Iou-Jen Liu, Raymond A. Yeh, and Alexander G. Schwing. Pic: Permutation invariant critic for multi-agent deep reinforcement learning. In *Proceedings of the Conference on Robot Learning*, volume 100 of *Proceedings of Machine Learning Research*, pp. 590–602. PMLR, 30 Oct–01 Nov 2020.

Jiarong Liu, Yifan Zhong, Siyi Hu, Haobo Fu, QIANG FU, Xiaojun Chang, and Yaodong Yang. Maximum Entropy Heterogeneous-Agent Reinforcement Learning. In *The Twelfth International Conference on Learning Representations*, 2024a.

Sicong Liu, Yang Shu, Chenjuan Guo, and Bin Yang. Learning Generalizable Skills from Offline Multi-Task Data for Multi-Agent Cooperation. In *The Thirteenth International Conference on Learning Representations*, 2025.

Yuntao Liu, Yuan Li, Xinhai Xu, Yong Dou, and Donghong Liu. Heterogeneous Skill Learning for Multi-agent Tasks. In *Advances in Neural Information Processing Systems*, volume 35, pp. 37011–37023, 2022.

Zongkai Liu, Qian Lin, Chao Yu, Xiawei Wu, Yile Liang, Donghui Li, and Xuetao Ding. Offline Multi-Agent Reinforcement Learning via In-Sample Sequential Policy Optimization, 2024b.

Ryan Lowe, YI WU, Aviv Tamar, Jean Harb, OpenAI Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Frans A Oliehoek, Matthijs TJ Spaan, and Nikos Vlassis. Optimal and approximate Q-value functions for decentralized POMDPs. *Journal of Artificial Intelligence Research*, 32:289–353, 2008.

Frans A Oliehoek, Christopher Amato, et al. *A concise introduction to decentralized POMDPs*, volume 1. Springer, 2016.

Shayegan Omidshafiei, Jason Pazis, Christopher Amato, Jonathan P. How, and John Vian. Deep decentralized multi-task multi-agent reinforcement learning under partial observability. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 2681–2690. PMLR, 06–11 Aug 2017.

Ling Pan, Tabish Rashid, Bei Peng, Longbo Huang, and Shimon Whiteson. Regularized Softmax Deep Multi-Agent Q-Learning. In *Advances in Neural Information Processing Systems*, volume 34, pp. 1365–1377, 2021.

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 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 17221–17237. PMLR, 17–23 Jul 2022.

Bei Peng, Tabish Rashid, Christian Schroeder de Witt, Pierre-Alexandre Kamienny, Philip Torr, Wendelin Boehmer, and Shimon Whiteson. Facmac: Factored multi-agent centralised policy gradients. In *Advances in Neural Information Processing Systems*, volume 34, pp. 12208–12221, 2021.

Tabish Rashid, Mikayel Samvelyan, Christian Schroeder de Witt, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning. *Journal of Machine Learning Research*, 21(178):1–51, 2020.

Mikayel Samvelyan, Tabish Rashid, Christian Schroeder de Witt, Gregory Farquhar, Nantas Nardelli, Tim G. J. Rudner, Chia-Man Hung, Philip H. S. Torr, Jakob Foerster, and Shimon Whiteson. The StarCraft Multi-Agent Challenge. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '19, pp. 2186–2188, 2019. ISBN 9781450363099.

Jianzhun Shao, Yun Qu, Chen Chen, Hongchang Zhang, and Xiangyang Ji. Counterfactual Conservative Q Learning for Offline Multi-agent Reinforcement Learning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Yoav Shoham and Kevin Leyton-Brown. *Multiagent systems: Algorithmic, game-theoretic, and logical foundations*. Cambridge University Press, 2008.

Peter Stone, Gal Kaminka, Sarit Kraus, and Jeffrey Rosenschein. Ad hoc autonomous agent teams: Collaboration without pre-coordination. *Proceedings of the AAAI Conference on Artificial Intelligence*, 24(1):1504–1509, Jul. 2010. doi: 10.1609/aaai.v24i1.7529.

Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, Vinicius Zambaldi, Max Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z. Leibo, Karl Tuyls, and Thore Graepel. Value-Decomposition Networks For Cooperative Multi-Agent Learning Based On Team Reward. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '18, pp. 2085–2087, 2018.

Ardi Tampuu, Tambet Matiisen, Dorian Kodelja, Ilya Kuzovkin, Kristjan Korjus, Juhan Aru, Jaan Aru, and Raul Vicente. Multiagent cooperation and competition with deep reinforcement learning. *PLOS ONE*, 12(4):1–15, 04 2017. doi: 10.1371/journal.pone.0172395.

Zikang Tian, Ruizhi Chen, Xing Hu, Ling Li, Rui Zhang, Fan Wu, Shaohui Peng, Jiaming Guo, Zidong Du, Qi Guo, and Yunji Chen. Decompose a Task into Generalizable Subtasks in Multi-Agent Reinforcement Learning. In *Advances in Neural Information Processing Systems*, volume 36, pp. 78514–78532, 2023.

Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of machine learning research*, 9(11), 2008.

A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

Tonghan Wang, Tarun Gupta, Anuj Mahajan, Bei Peng, Shimon Whiteson, and Chongjie Zhang. RODE: Learning Roles to Decompose Multi-Agent Tasks. In *International Conference on Learning Representations*, 2021.

Weixun Wang, Tianpei Yang, Yong Liu, Jianye Hao, Xiaotian Hao, Yujing Hu, Yingfeng Chen, Changjie Fan, and Yang Gao. From few to more: Large-scale dynamic multiagent curriculum learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):7293–7300, Apr. 2020. doi: 10.1609/aaai.v34i05.6221.

Jizhou Wu, Jianye Hao, Tianpei Yang, Xiaotian Hao, Yan Zheng, Weixun Wang, and Matthew E. Taylor. PORTAL: Automatic Curricula Generation for Multiagent Reinforcement Learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(14):15934–15942, Mar. 2024. doi: 10.1609/aaai.v38i14.29524.

Yingxuan Yang, Huacan Chai, Shuai Shao, Yuanyi Song, Siyuan Qi, Renting Rui, and Weinan Zhang. Agentnet: Decentralized evolutionary coordination for llm-based multi-agent systems. In *Advances in Neural Information Processing Systems*, volume 38, pp. 107309–107336, 2025.

Chao Yu, Akash Velu, Eugene Vinitsky, Jiaxuan Gao, Yu Wang, Alexandre Bayen, and YI WU. The Surprising Effectiveness of PPO in Cooperative Multi-Agent Games. In *Advances in Neural Information Processing Systems*, volume 35, pp. 24611–24624, 2022.

Tianhe Yu, Garrett Thomas, Lantao Yu, Stefano Ermon, James Y Zou, Sergey Levine, Chelsea Finn, and Tengyu Ma. Mopo: Model-based offline policy optimization. *Advances in Neural Information Processing Systems*, 33:14129–14142, 2020.

Fuxiang Zhang, Chengxing Jia, Yi-Chen Li, Lei Yuan, Yang Yu, and Zongzhang Zhang. Discovering Generalizable Multi-agent Coordination Skills from Multi-task Offline Data. In *The Eleventh International Conference on Learning Representations*, 2023a.

Junkai Zhang, Jinmin He, Yifan Zhang, Yifan Zang, Ning Xu, and Jian Cheng. Bi-level knowledge transfer for multi-task multi-agent reinforcement learning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Ziqian Zhang, Lei Yuan, Lihe Li, Ke Xue, Chengxing Jia, Cong Guan, Chao Qian, and Yang Yu. Fast Teammate Adaptation in the Presence of Sudden Policy Change. In *Proceedings of the Thirty-Ninth Conference on Uncertainty in Artificial Intelligence*, volume 216 of *Proceedings of Machine Learning Research*, pp. 2465–2476. PMLR, 31 Jul–04 Aug 2023b.

Technical Appendices

The organization of the appendix is as follows:

- Appendix A: We provide related work in detail.
- Appendix B: We provide the proofs of our theorems presented in Section 4.
- Appendix C: We provide the pseudocode of TasQ.
- Appendix D: We provide details of the experiment, including the composition of the datasets (D.1), the primary information of the tasks and task sets (D.2), implementation specifics of the algorithms (D.3), key hyperparameters (D.4), and the computational resources (D.5).
- Appendix E: We provide ablation experiments and results regarding the main components (E.1.1), design choices (E.1.2), and hyperparameters (E.2) of TasQ.
- Appendix F: We provide detailed visualization process and results, together with a mutual information analysis of the discovered skills.
- Appendix G: We provide additional results for SMACv2 tasks.
- Appendix H: We provide additional results for continuous action tasks.
- Appendix I: We provide discussions about the assumption and scope of this paper (I.1), detailed comparison with related works (I.2–I.4), and the term usage (I.5).
- Appendix J: We report the detailed results of each task set.
- Appendix K: We display the learning curves of TasQ and all baselines.

A Related Work

MARL Multi-Agent Reinforcement Learning (MARL) has seen substantial progress in recent years, with numerous approaches developed under different paradigms. The centralized training with decentralized execution (CTDE) paradigm (Oliehoek et al., 2008; Amato, 2024) has been particularly influential, with methods such as HASAC (Liu et al., 2024a), MAPPO (Yu et al., 2022), QMIX (Rashid et al., 2020), VDN (Sunehag et al., 2018), and MADDPG (Lowe et al., 2017). These approaches use centralized training for better coordination and decentralized execution for real-time decision-making. On the other hand, fully decentralized training and execution schemes have also been explored (Tampuu et al., 2017; de Witt et al., 2020). However, the performance of such methods is often constrained by the absence of information sharing. In this paper, we mainly focus on the CTDE paradigm with QMIX as the backbone.

Offline MARL Due to the absence of online interaction with the environment, offline training faces a fundamental challenge—distributional shift. To address this issue, several techniques for single-agent RL have been proposed, many of which leverage conservatism to regularize either the policy (Fujimoto & Gu, 2021; Kostrikov et al., 2021) or the Q-value function (Kumar et al., 2020; Kostrikov et al., 2022). These methods mitigate the risks of overestimating the value of unseen state-action pairs. However, specific challenges caused by multiple agents, such as the exponential explosion of complexity, hinder these techniques from directly extending to multi-agent scenarios. Therefore, several tailored approaches (Jiang & Lu, 2023; Pan et al., 2022; Li et al., 2023; Shao et al., 2023; Liu et al., 2024b) have been proposed to address offline MARL. However, these methods often focus excessively on source tasks, inevitably compromising their multi-task generalization ability.

Multi-task MARL Current research on multi-task MARL can be broadly categorized into two types. The first, often referred to as Ad-Hoc Teamwork (Stone et al., 2010; Zhang et al., 2023b), focuses on exploring how to effectively collaborate with unknown and uncontrollable teammates within a given task. The second type, which is the primary focus of this paper, involves scenarios where the algorithm controls the entire team, but the agents are trained on some tasks and tested on unseen tasks with similar structures (Omidshafiei et al., 2017; Wang et al., 2020; Wu et al., 2024). This requires agents to learn and utilize generalizable decision-making structures across tasks from a limited set of source tasks, positioning skill discovery as a promising solution. However, most existing approaches discover skills typically through sample reconstruction (Wang et al., 2021; Liu et al., 2022), clustering (Li et al., 2025a), or sub-task decomposition (Tian et al., 2023), primarily relying on online interactions, with limited attention to offline settings. VO-MASD (Chen et al., 2025) explores offline skill discovery with a codebook. However, its emphasis lies in training high-level policies online using the discovered skills, rather than in a fully offline setting. Recently, ODIS (Zhang et al., 2023a) proposed a method for multi-task offline MARL, but it suffers from the limitations of behavior cloning when the scale of source tasks is small. Inspired by ODIS, HiSSD (Liu et al., 2025) leverages value functions to separately extract common and task-specific skills, improving generalization performance. However, its task efficiency remains constrained by the BC action decoder, which hinders the discovery of effective skills from a limited number of source tasks. BiKT (Zhang et al., 2025) leverages a decision transformer to extract skills at both the individual and team levels, enabling the generation of more fine-grained, skill-conditioned actions. STAIRS-Former (Jeon et al., 2026) introduces an interleaved spatio-temporal transformer to capture long interaction histories. However, these dedicated architectures pay little attention to task efficiency and often run at substantially higher computational cost. A more detailed comparison with these related works is provided in Appendices I.2–I.4.

B Proofs for Theorem in Section 4

Here we provide detailed proofs for Theorem 4.3 presented in Section 4.

First, we introduce the Performance Difference Lemma by Kakade & Langford (2002):

Lemma B.1 (Lemma 6.1 from Kakade & Langford (2002)). *For any two policies π, π',*

$$\mathcal{J}(\pi') - \mathcal{J}(\pi) = \frac{1}{1-\gamma}\,\mathbb{E}_{s\sim d^{\pi'}}\Big[\,\mathbb{E}_{a\sim\pi'(\cdot|s)}\big[A^{\pi}(s,a)\big]\Big],$$

where $d^{\pi'}$ is the discounted state-occupancy measure of π', $A^{\pi}(s,a) := Q^{\pi}(s,a) - V^{\pi}(s)$ and $V^{\pi}(s) := \mathbb{E}_{a\sim\pi(\cdot|s)}[Q^{\pi}(s,a)]$.

Based on this lemma, we derive the following corollary for our specific setting:

Corollary B.2. *Let $\hat{\pi} \in \Pi_{\phi^*}$ be any ϕ^*-constrained policy. Define $\delta_{\hat{\pi}}(s) := Q^{\pi^*_{\phi^*}}(s, \pi^*_{\phi^*}(s)) - Q^{\pi^*_{\phi^*}}(s, \hat{\pi}(s))$, we have*

$$\mathcal{J}(\pi^*) - \mathcal{J}(\hat{\pi}) = \varepsilon_{\text{rep}} + \frac{1}{1-\gamma}\,\mathbb{E}_{s\sim d^{\hat{\pi}}}\big[\delta_{\hat{\pi}}(s)\big].$$

Proof. Apply Lemma B.1 with $\pi = \pi^*_{\phi^*}$ and $\pi' = \hat{\pi}$, we obtain:

$$\mathcal{J}(\pi^*_{\phi^*}) - \mathcal{J}(\hat{\pi}) = -\frac{1}{1-\gamma}\,\mathbb{E}_{s\sim d^{\hat{\pi}}}\Big[\,\mathbb{E}_{a\sim\hat{\pi}(\cdot|s)}\big[A^{\pi^*_{\phi^*}}(s,a)\big]\Big].$$

Since $\hat{\pi}$ is deterministic, we have $\mathbb{E}_{a\sim\hat{\pi}(\cdot|s)}[A^{\pi^*_{\phi^*}}(s,a)] = A^{\pi^*_{\phi^*}}(s,\hat{\pi}(s))$. Furthermore, by definition, $A^{\pi^*_{\phi^*}}(s,a) = Q^{\pi^*_{\phi^*}}(s,a) - V^{\pi^*_{\phi^*}}(s) = Q^{\pi^*_{\phi^*}}(s,a) - Q^{\pi^*_{\phi^*}}(s,\pi^*_{\phi^*}(s))$. Thus,

$$\begin{aligned}\mathcal{J}(\pi^*_{\phi^*}) - \mathcal{J}(\hat{\pi}) &= \frac{1}{1-\gamma}\mathbb{E}_{s\sim d^{\hat{\pi}}}\Big[Q^{\pi^*_{\phi^*}}(s,\pi^*_{\phi^*}(s)) - Q^{\pi^*_{\phi^*}}(s,\hat{\pi}(s))\Big]\\ &= \frac{1}{1-\gamma}\mathbb{E}_{s\sim d^{\hat{\pi}}}\big[\delta_{\hat{\pi}}(s)\big].\end{aligned}$$

Recall that $\varepsilon_{\text{rep}} := \mathcal{J}(\pi^*) - \mathcal{J}(\pi^*_{\phi^*})$. By decomposing the total gap, we have:

$$\mathcal{J}(\pi^*) - \mathcal{J}(\hat{\pi}) = \big(\mathcal{J}(\pi^*) - \mathcal{J}(\pi^*_{\phi^*})\big) + \big(\mathcal{J}(\pi^*_{\phi^*}) - \mathcal{J}(\hat{\pi})\big) = \varepsilon_{\text{rep}} + \frac{1}{1-\gamma}\mathbb{E}_{s\sim d^{\hat{\pi}}}[\delta_{\hat{\pi}}(s)].$$

□

Now, we present the proof for Theorem 4.3.

Theorem 4.3. *Under Assumptions 4.1–4.2, we have*

$$\begin{aligned}\mathcal{J}(\pi^*) - \mathcal{J}(\hat{\pi}_{\text{BC}}) &\le \varepsilon_{\text{rep}} + \frac{\varepsilon_{BC}}{1-\gamma} + \frac{2Q_{\max}\kappa}{1-\gamma},\\ \mathcal{J}(\pi^*) - \mathcal{J}(\hat{\pi}_{\text{CQL}}) &\le \varepsilon_{\text{rep}} + \frac{2\varepsilon_{Q}}{1-\gamma},\end{aligned}$$

where $\varepsilon_{BC} = \sup_{s\in\mathcal{S}} |Q^{\pi^*_{\phi^*}}(s,\pi^*_{\phi^*}(s)) - Q^{\pi^*_{\phi^*}}(s,a_{\text{m}}(s))|$ and $\kappa = \mathbb{E}_{s\sim d^{\hat{\pi}_{\text{BC}}}}\left[\mathbb{1}\left\{\hat{\pi}_{\text{BC}}(s) \neq a_{\text{m}}(s)\right\}\right]$.

Proof. Our proof contains two steps: (i) bounding the $\hat{\pi}_{\text{BC}}$ performance and (ii) Bounding the $\hat{\pi}_{\text{CQL}}$ performance.

1. Bounding the $\hat{\pi}_{\text{BC}}$ performance.

By the definition of $\hat{\pi}_{\text{BC}}$, we examine the local gap $\delta_{\hat{\pi}_{\text{BC}}}(s)$. Since $\pi^*_{\phi^*}$ is optimal within Π_{ϕ^*}, $\delta_{\hat{\pi}_{\text{BC}}}(s) \ge 0$. Thus, we have:

$$\begin{aligned}\delta_{\hat{\pi}_{\text{BC}}}(s) =& Q^{\pi^*_{\phi^*}}(s,\pi^*_{\phi^*}(s)) - Q^{\pi^*_{\phi^*}}(s,\hat{\pi}_{\text{BC}}(s))\\ \le& \big|Q^{\pi^*_{\phi^*}}(s,\pi^*_{\phi^*}(s)) - Q^{\pi^*_{\phi^*}}(s,a_{\text{m}}(s))\big| + \big|Q^{\pi^*_{\phi^*}}(s,a_{\text{m}}(s)) - Q^{\pi^*_{\phi^*}}(s,\hat{\pi}_{\text{BC}}(s))\big|\\ \le& \varepsilon_{BC} + \big|Q^{\pi^*_{\phi^*}}(s,a_{\text{m}}(s)) - Q^{\pi^*_{\phi^*}}(s,\hat{\pi}_{\text{BC}}(s))\big|\mathbb{1}\{\hat{\pi}_{\text{BC}}(s) \neq a_{\text{m}}(s)\}\\ \le& \varepsilon_{BC} + 2Q_{\max}\mathbb{1}\{\hat{\pi}_{\text{BC}}(s) \neq a_{\text{m}}(s)\}\end{aligned}$$

Plugging this into Corollary B.2:

$$\mathcal{J}(\pi^*) - \mathcal{J}(\hat{\pi}_{\text{BC}}) \le \varepsilon_{\text{rep}} + \frac{\varepsilon_{BC}}{1-\gamma} + \frac{2Q_{\max}}{1-\gamma}\mathbb{E}_{s\sim d^{\hat{\pi}_{\text{BC}}}}\left[\mathbb{1}\left\{\hat{\pi}_{\text{BC}}(s) \neq a_{\text{m}}(s)\right\}\right].$$

Substituting κ completes the first part of the proof.

2. Bounding the $\hat{\pi}_{\mathrm{CQL}}$ performance.

By the definition of $\hat{\pi}_{\mathrm{CQL}}$, for any s, $\hat{Q}(s, \pi^*_{\phi^*}(s), \phi^*(s)) \leq \hat{Q}(s, \hat{\pi}_{\mathrm{CQL}}(s), \phi^*(s))$. Using this property and Assumption 4.2:

$$\begin{aligned}
\delta_{\hat{\pi}_{\mathrm{CQL}}}(s) &= Q^{\pi^*_{\phi^*}}(s, \pi^*_{\phi^*}(s)) - Q^{\pi^*_{\phi^*}}(s, \hat{\pi}_{\mathrm{CQL}}(s)) \\
&= Q^{\pi^*_{\phi^*}}(s, \pi^*_{\phi^*}(s)) - \hat{Q}(s, \pi^*_{\phi^*}(s), \phi^*(s)) + \hat{Q}(s, \pi^*_{\phi^*}(s), \phi^*(s)) - Q^{\pi^*_{\phi^*}}(s, \hat{\pi}_{\mathrm{CQL}}(s)) \\
&\leq Q^{\pi^*_{\phi^*}}(s, \pi^*_{\phi^*}(s)) - \hat{Q}(s, \pi^*_{\phi^*}(s), \phi^*(s)) + \hat{Q}(s, \hat{\pi}_{\mathrm{CQL}}(s), \phi^*(s)) - Q^{\pi^*_{\phi^*}}(s, \hat{\pi}_{\mathrm{CQL}}(s)) \\
&\leq \left|Q^{\pi^*_{\phi^*}}(s, \pi^*_{\phi^*}(s)) - \hat{Q}(s, \pi^*_{\phi^*}(s), \phi^*(s))\right| + \left|Q^{\pi^*_{\phi^*}}(s, \hat{\pi}_{\mathrm{CQL}}(s)) - \hat{Q}(s, \hat{\pi}_{\mathrm{CQL}}(s), \phi^*(s))\right| \\
&\leq \varepsilon_Q + \varepsilon_Q = 2\varepsilon_Q.
\end{aligned}$$

Plugging this constant bound into Corollary B.2 yields:

$$\mathcal{J}(\pi^*) - \mathcal{J}(\hat{\pi}_{\mathrm{CQL}}) \leq \varepsilon_{\mathrm{rep}} + \frac{2\varepsilon_Q}{1-\gamma}.$$

Thus, we finish the proof. □

C Algorithm Pseudocode

Algorithm 1 Task-efficient Skill-based Q-learning (TasQ)

Input: Datasets of source tasks $\{\mathcal{D}_m\}_{m \in \mathbb{T}_s}$.
Initialize the parameters ϕ for the encoder and decoder, θ for Q_{tot} and $\bar{\theta}$ for target Q_{tot}.
for $i = 1$**to** T_{max} **do**
  for $j = 1$ **to** $|\mathbb{T}_s|$ **do**
    Sample trajectories $\hat{\tau} = \{(s_t, o_t, a_t, r_t, o'_t)\}$ from $\mathcal{D}_j$.
    Calculate the total loss $\mathcal{L}_j$ by $\hat{\tau}$ for task j, according to equation 8.
  end for
  Calculate the multi-task loss $\mathcal{L} = \sum_{j=1}^{|\mathbb{T}_s|} \mathcal{L}_j$.
  Update ϕ with $\nabla_\phi \mathcal{L}$ and update θ with $\nabla_\theta \mathcal{L}$
  if $i \equiv 0 \pmod{T_{\text{update}}}$ **then**
    $\bar{\theta} \leftarrow \theta$
  end if
end for

D Experiment Details

D.1 Datasets

Following the definition by Fu et al. (2020), we conduct experiments primarily on datasets of four qualities. The collection procedure for each quality is as follows:

- Expert: Trajectories collected from policies trained to an expert level using QMIX.
- Medium: Trajectories collected from policies trained to a medium level using QMIX.
- Medium-Replay: The replay buffer generated while training QMIX policies to a medium level.
- Medium-Expert: A 50-50 mixture of Medium and Expert trajectories.

To ensure fair comparisons, we use the datasets provided by ODIS (Zhang et al., 2023a), where only up to $2,000$ trajectories are sampled for each dataset. The key information of the datasets used in our experiments is summarized in Table 3.

Table 3: The primary information of the offline datasets used in our experiments.

Task	#Trajectories	Average Return
3m-Expert	2,000	19.87
3m-Medium	2,000	14.00
3m-Medium-Replay	2,000	10.71
3m-Medium-Expert	2,000	16.94
5m-Expert	2,000	19.93
5m-Medium	2,000	17.35
5m-Medium-Replay	1,266	3.21
5m-Medium-Expert	2,000	18.66
10m-Expert	2,000	19.89
10m-Medium	2,000	16.63
10m-Medium-Replay	331	2.30
10m-Medium-Expert	2,000	18.27
25m-Expert	2,000	19.90
5m_vs_6m-Expert	2,000	17.34
5m_vs_6m-Medium	2,000	12.63
5m_vs_6m-Medium-Replay	2,000	9.41
5m_vs_6m-Medium-Expert	2,000	14.98
9m_vs_10m-Expert	2,000	19.59
9m_vs_10m-Medium	2,000	15.52
9m_vs_10m-Medium-Replay	2,000	11.76
9m_vs_10m-Medium-Expert	2,000	17.49
2s3z-Expert	2,000	19.78
2s3z-Medium	2,000	16.61
2s3z-Medium-Replay	2,000	14.25
2s3z-Medium-Expert	2,000	18.26
2s4z-Expert	2,000	19.73
2s4z-Medium	2,000	16.85
2s4z-Medium-Replay	2,000	14.38
2s4z-Medium-Expert	2,000	18.26
3s5z-Expert	2,000	19.78
3s5z-Medium	2,000	16.31
3s5z-Medium-Replay	2,000	15.29
3s5z-Medium-Expert	2,000	18.05

D.2 TASKS CONFIGURATION

In this section, we present the task composition of each SMAC task set in Table 4 and the key information of all SMAC tasks involved in our experiments in Table 5. The details about SMACv2 and Continuous-MPE tasks are provided in Appendices G and H, respectively.

Table 4: The task composition of each SMAC scenario used in our experiments.

Scenarios	Source Tasks	Unseen Tasks
Marine-Easy	3m, 5m, 10m	4m, 6m, 7m, 8m, 9m, 11m, 12m
Marine-Hard	3m, 5m_vs_6m, 9m_vs_10m	4m, 5m, 10m, 12m, 7m_vs_8m, 8m_vs_9m, 10m_vs_11m, 10m_vs_12m, 13m_vs_15m
Stalker-Zealot	2s3z, 2s4z, 3s5z	1s3z, 1s4z, 1s5z, 2s5z, 3s3z, 3s4z, 4s3z, 4s4z, 4s5z, 4s6z
Marine-Single	3m	5m, 8m, 10m, 12m
Marine-Single-Inv	10m	3m, 4m, 5m, 8m
Marine-Single-Mid	5m	3m, 8m, 10m, 12m
Marine-Single-Large	25m	3m, 10m, 28m, 30m

Table 5: The key information about the SMAC tasks used in our experiments.

Task Name	Allied Units	Enemy Units
3m	3 Marines	3 Marines
4m	4 Marines	4 Marines
5m	5 Marines	5 Marines
6m	6 Marines	6 Marines
7m	7 Marines	7 Marines
8m	8 Marines	8 Marines
9m	9 Marines	9 Marines
10m	10 Marines	10 Marines
11m	11 Marines	11 Marines
12m	12 Marines	12 Marines
25m	25 Marines	25 Marines
28m	28 Marines	28 Marines
30m	30 Marines	30 Marines
5m_vs_6m	5 Marines	6 Marines
7m_vs_8m	7 Marines	8 Marines
8m_vs_9m	8 Marines	9 Marines
9m_vs_10m	9 Marines	10 Marines
10m_vs_11m	10 Marines	11 Marines
10m_vs_12m	10 Marines	12 Marines
13m_vs_15m	13 Marines	15 Marines
1s3z	1 Stalkers, 3 Zealots	1 Stalkers, 3 Zealots
1s4z	1 Stalkers, 4 Zealots	1 Stalkers, 4 Zealots
1s5z	1 Stalkers, 5 Zealots	1 Stalkers, 5 Zealots
2s3z	2 Stalkers, 3 Zealots	2 Stalkers, 3 Zealots
2s4z	2 Stalkers, 4 Zealots	2 Stalkers, 4 Zealots
2s5z	2 Stalkers, 5 Zealots	2 Stalkers, 5 Zealots
3s3z	3 Stalkers, 3 Zealots	3 Stalkers, 3 Zealots
3s4z	3 Stalkers, 4 Zealots	3 Stalkers, 4 Zealots
3s5z	3 Stalkers, 5 Zealots	3 Stalkers, 5 Zealots
4s3z	4 Stalkers, 3 Zealots	4 Stalkers, 3 Zealots
4s4z	4 Stalkers, 4 Zealots	4 Stalkers, 4 Zealots
4s5z	4 Stalkers, 5 Zealots	4 Stalkers, 5 Zealots
4s6z	4 Stalkers, 6 Zealots	4 Stalkers, 6 Zealots

D.3 Implementation Details

D.3.1 Decomposer

To handle variable-sized inputs across different tasks, we follow prior works (Hu et al., 2021; Tian et al., 2023; Zhang et al., 2023a; Liu et al., 2025) and adopt a rule-based decomposer to split the input observations or states into individual entity information units. This is feasible because SMAC provides a well-defined and consistent structure for states and observations across tasks.

Specifically, taking observations as an example, each agent's observation is a concatenation of its own features, the features of all allies, and the features of all enemies. While the feature schema for each entity is fixed and standardized, the number of allies and enemies depends on the task configuration. For entities that are not observable (e.g., outside the agent's field of view), their corresponding feature fields are zero-masked. This allows a deterministic rule-based decomposer to segment the observation into individual entity features based on the known observation layout and the number of agents involved in the current task.

Importantly, the decomposer operates solely based on environment-level specifications rather than task-level ones. And, as discussed in the main text, our study focuses on a series of structurally similar tasks that differ in scale, difficulty, and objectives. Therefore, although SMAC explicitly provides the number of interactable entities and the total number of other entities, similar information can be obtained in other environments using techniques such as entity detection or masking mechanisms. Hence, this component does not limit the generality or contribution of TasQ.

D.3.2 TasQ

Skill Discovery. In the skill discovery part, we use single-layer Transformers for both the encoder and the decoder, and fully connected layers for embedding and skill extraction. To make the skill vector extract more information, the decoder reconstructs the embedding associated with the agent itself directly with z:

$$\widehat{E}_{i,0} = W_0 \cdot \mathrm{ReLU}(z) + b_0.$$

While for embeddings of other entities, it first masks parts of the information by a ReLU function, and then concatenates them with z for reconstruction:

$$\widehat{E}_{i,k} = W_k \cdot [\mathrm{ReLU}(E_{i,k}),\ z] + b_k, \quad k = 1, 2, \ldots, K_i \tag{9}$$

where $[\cdot,\ \cdot]$ represents the concatenation operation.

Furthermore, regarding the implementation of the hidden variable h, it is designed as a tensor with the same shape as other embedding units and serves as the final token of the input/output sequence in the transformer (denoted as e_{i,K_i+1} or E_{i,K_i+1}). We initialize h to zero at the start of each trajectory (during both training and inference) and update it recurrently through the transformer at each timestep until the trajectory terminates.

Skill-conditioned Policy Optimization. In the skill-conditioned policy optimization part, we use MLPs as the individual Q network, and similar to ODIS, we employ an attention-based mixing network to handle variable inputs in multi-task training. The entity decomposition method and its dimensions are the same as those used in the observation reconstruction. The specific parameters of the network structure are shown in Table 6. For Conservative Q-Learning, the sampling strategy μ we use is a uniform distribution over the action space, with 100 samples drawn to estimate the CQL regularization term. Additionally, consistent with the QMIX implementation provided by PYMARL2 (Hu et al., 2023), we use TD(λ) to enhance stability when calculating the TD loss, and the λ parameter is set to the default value from PYMARL2.

D.3.3 Baselines

For the baselines except for HiSSD, we utilize the implementations provided by ODIS. Specifically, BC-t employs a transformer capable of decomposing observations for multi-task training while optimizing the policy through behavior cloning. BC-r is similar to BC-t but incorporates return-to-go

Table 6: The main network structure involved in TasQ.

Network	Architecture	Activation Function
Transformer	Head=1, Depth=1, Embedding Dim=64	——
Fixed Individual Q Network	$[96, 128, 6]$	ReLU
Variable Individual Q Network	$[96, 128, 1]$	ReLU
HyperNet in Mixing Network	$[128, 64, 32]$	ReLU
Attention in Mixing Network	Attention Dim=8	——

information as an additional input alongside observations. The vanilla CQL is almost the same as TasQ, but it removes the skill-discovery module and the local value calibration term. It is only equipped with the observation encoder, and feeds the encoding vectors $\{E_{i,k}\}_{k=0}^{K_i}$ to separated Q-networks directly. Additionally, in vanilla CQL, the gradient can propagate from the encoder to the Q-networks. The implementations of ODIS and HiSSD align with their official version.

Reproduction Validation. Although all baselines, when applicable, are implemented *using the official codebases and the hyperparameters recommended in the original papers*, we provide a side-by-side comparison on Multi-to-Multi task sets between our reproduced results and the numbers reported in the corresponding works in Tables 7, 8, and 9 for validation. Due to factors such as randomness and instability of the baselines, we observe certain discrepancies. For example, the BC-best scores in our reproduction are noticeably higher than those reported by Zhang et al. (2023a). In addition, both our results and those in the ODIS paper (Zhang et al., 2023a) are based on the final policy, whereas the HiSSD paper (Liu et al., 2025) reports the best-performing checkpoint (denoted by "Best"), which may further contribute to the differences. Table 9 also includes our results for HiSSD and TasQ under the "Best" protocol. The data indicates that our approach maintains its superiority over HiSSD in this setting.

Remark: *Regarding the results, several points should be emphasized:*

- *Since the original papers did not release raw data for individual seeds, it is mathematically impossible to derive the standard deviation for the aggregated mean performance across the tasks. To maintain empirical integrity, we report only the mean values for the original results.*
- *Consequently, directly comparing the original results with our reproduced results is statistically unsound. Instead, the validity of our method is substantiated through a side-by-side comparison with the results in Table 1. Notably, on the* ***heterogeneous scenario stalker-zealot****, TasQ outperforms the baseline by a significant margin, marking a substantial contribution.*
- *The inclusion of original results is intended solely for reference. To ensure a fair comparison, we strictly adhere to the "Final" protocol (i.e., using the evaluation values from the last iteration) of our reproduced results as the standard.*

Table 7: Comparison between the original results and our reproduced results for BC-best.

Task	BC-best (Ours), Final	BC-best (Zhang et al., 2023a), Final
marine-easy-e	99.38	82.42
marine-easy-m	71.81	70.57
marine-easy-mr	49.62	16.07
marine-easy-me	75.25	58.52
marine-hard-e	64.84	55.40
marine-hard-m	42.08	31.81
marine-hard-mr	46.98	42.55
marine-hard-me	51.93	39.51
stalker-zealot-e	68.61	67.15
stalker-zealot-m	23.94	21.84
stalker-zealot-mr	17.64	13.54
stalker-zealot-me	39.62	40.98

Table 8: Comparison between the original results and our reproduced results for ODIS.

Task	ODIS (Ours), Final	ODIS (Zhang et al., 2023a), Final
marine-easy-e	70.50	85.83
marine-easy-m	68.50	75.30
marine-easy-mr	11.06	11.25
marine-easy-me	66.00	71.60
marine-hard-e	21.09	55.02
marine-hard-m	32.55	41.43
marine-hard-mr	37.29	43.07
marine-hard-me	19.58	42.33
stalker-zealot-e	55.05	67.27
stalker-zealot-m	11.97	27.52
stalker-zealot-mr	8.17	11.72
stalker-zealot-me	30.72	40.27

Table 9: Comparison between the original results of HiSSD, our reproduced results for HiSSD, and the results of TasQ under the "Best" protocol.

Task	HiSSD (Ours), Final	HiSSD (Liu et al., 2025), Best	HiSSD (Ours), Best	TasQ (Ours), Best
marine-easy-e	97.94	98.64	100.00	100.00
marine-easy-m	71.44	78.74	87.50	93.31
marine-easy-mr	72.88	79.97	90.94	93.25
marine-easy-me	78.31	74.34	97.75	99.56
marine-hard-e	68.02	68.15	78.80	83.39
marine-hard-m	47.60	49.02	62.34	63.23
marine-hard-mr	47.55	48.59	62.24	64.74
marine-hard-me	53.28	59.53	74.27	72.71
stalker-zealot-e	66.59	73.84	94.76	98.75
stalker-zealot-m	23.51	18.42	48.12	83.27
stalker-zealot-mr	15.00	12.94	47.84	67.55
stalker-zealot-me	40.29	43.91	79.76	96.63

D.4 HYPERPARAMETERS

For TasQ, the main hyperparameters shared across all tasks are listed in Table 10. We primarily adjust the local value calibration coefficient η, with the specific settings provided in Table 11. All other hyperparameters are kept consistent with the default configuration of QMIX in PYMARL2.

Remark: *η is an important hyperparameter for TasQ to balance in-task performance and cross-task generalization. A practical tuning guideline is to start from a moderate value (e.g., $\eta = 0.5$). When we need to reduce Q-value estimation error, for example, with lower-quality datasets or larger agent populations, a larger η tends to work better. Conversely, when Q-value can be estimated well, such as with higher-quality data or fewer agents, a smaller η is generally more favorable for generalization. Although exceptions exist for simpler tasks where direct imitation could be more effective, these guidelines remain applicable to most non-trivial scenarios. For instance, $\eta = 0$ works best on SMACv2, whose per-episode randomization makes the logged actions a less stable calibration reference.*

Table 10: The hyperparameters shared by all tasks for TasQ.

Hyperparameter	Value
Entity Embedding Dim	64
Skill Dimension of z	32
λ for $TD(\lambda)$	0.6
Conservative Weight α	1.0
T_{max}	35,000
T_{update}	80
Batch Size	32
Learning Rate	0.005
Optimizer	Adam (Kingma & Ba, 2015)

Table 11: Local value calibration coefficient η for each task set.

Task set	Value
Marine-Easy-Expert	0.9
Marine-Easy-Medium	0.2
Marine-Easy-Medium-Replay	0.5
Marine-Easy-Medium-Expert	0.5
Marine-Hard-Expert	0.5
Marine-Hard-Medium	0.8
Marine-Hard-Medium-Replay	0.9
Marine-Hard-Medium-Expert	0.7
Stalker-Zealot-Expert	0.5
Stalker-Zealot-Medium	0.1
Stalker-Zealot-Medium-Replay	0.5
Stalker-Zealot-Medium-Expert	0.3
Marine-Single	0.5
Marine-Single-Inv	0.3
Marine-Single-Mid	0.2
Marine-Single-Large	0.2
SMACv2	0.0
Continuous-MPE	0.5

D.5 Computational Resources

We run TasQ on a single NVIDIA RTX A6000 GPU. Each run takes approximately 5 to 13 hours, depending on the task set (about 5 hours for the Marine-Easy-Expert task set and about 13 hours for the Marine-Single-Large task set).

E Ablation Study

To investigate the contribution of each component of TasQ to its superior performance, as well as its sensitivity to different hyperparameter values, we perform an ablation study on the One-to-Multi task sets.

E.1 Component Ablation

E.1.1 Key Components of TasQ

The component ablation is presented in Figure 3 (Section 6.3). Here we provide per-task results in Table 12 for a detailed analysis. We evaluate four variants: (i) **w/o LVC**, which removes the LVC term and instead allows the gradient to flow freely between skill discovery and Q-learning; (ii) **w/o TD**, which removes the Q-learning objective and relies solely on LVC (i.e., $\eta = 1$); (iii) **w/o SD**, which removes the skill-discovery module; and (iv) **w/o SD & LVC**, which reduces to vanilla CQL. Note that the variants without skill discovery (w/o SD and w/o SD & LVC) have no stop-gradient operator to apply. Therefore, we likewise remove the operator in w/o LVC so that this variant is not disadvantaged by restricting the parameters its value objective can optimize. The effect of the stop-gradient operator itself is examined separately in Table 13.

Skill discovery is the primary contributor: Removing skill discovery (w/o SD) causes a substantial drop on both task sets ($62.00 \rightarrow 47.12$ on *Marine-Single* and $87.50 \rightarrow 45.62$ on *Marine-Single-Inv*). Moreover, it degrades most sharply on the test tasks whose scale is farthest from the source (e.g., 12m on *Marine-Single* and 3m–5m on *Marine-Single-Inv*), confirming that the discovered skills are the main driver of cross-task generalization.

LVC is essential under large-scale training and mitigates Q-value overestimation: The behavior of w/o LVC differs markedly between the two settings. On *Marine-Single* (trained on the small-scale

3m), w/o LVC performs comparably to full TasQ (61.5 vs. 62.00), as the small source task induces limited estimation error and pure value-based optimization already works well. In stark contrast, on *Marine-Single-Inv* (trained on the large-scale *10m*), w/o LVC collapses entirely to 0.00 across all test tasks, essentially identical to vanilla CQL (w/o SD & LVC, 2.75). This indicates that, when training on tasks with many agents, purely value-based optimization suffers from severe Q-value overestimation over the enlarged joint action space, and skill discovery alone cannot remedy it. Introducing LVC calibrates each agent's local action-values against the logged actions, effectively suppressing this overestimation and restoring the performance to 87.50.

Value-based optimization is indispensable: The w/o TD variant, which discards the Q-learning objective and retains only the imitation-style LVC term, is consistently and significantly below full TasQ (50.38 vs. 62.00 on *Marine-Single* and 73.87 vs. 87.50 on *Marine-Single-Inv*). Notably, on *Marine-Single* it even underperforms vanilla CQL (50.38 vs. 56.00), showing that imitation alone cannot select the return-maximizing action and thus fails to fully exploit the discovered skills.

Overall, these results confirm that every component is indispensable and complementary: skill discovery provides the transferable decision structures, TD-based value optimization selects the optimal skill-conditioned actions, and LVC stabilizes value estimation under large agent scales. TasQ's superiority thus arises from the coordination of all three modules rather than from any single one.

Table 12: Win rates of different TasQ variants. The reported results are averaged over 5 random seeds. The bold number denotes the best performance, and ± denotes one standard deviation.

Task Set		TasQ	w/o LVC	w/o TD	w/o SD	w/o SD & LVC
Marine Single	3m ◇	**100.00** ± 0.00	96.88 ± 2.21	99.38 ± 1.40	99.38 ± 1.40	**100.00** ± 0.00
	5m	**91.25** ± 10.92	80.62 ± 20.66	83.75 ± 15.69	60.62 ± 41.79	88.12 ± 7.46
	8m	**49.38** ± 23.63	48.12 ± 29.28	30.00 ± 23.24	33.75 ± 30.49	40.00 ± 26.00
	10m	41.88 ± 25.83	**46.25** ± 35.04	27.50 ± 27.19	31.87 ± 36.54	34.38 ± 25.39
	12m	27.50 ± 21.13	**35.62** ± 29.03	11.25 ± 9.53	10.00 ± 9.73	17.50 ± 10.27
Average		**62.00** ± 13.80	61.50 ± 18.48	50.38 ± 11.68	47.12 ± 20.16	56.00 ± 11.07
Marine Single-Inv	10m ◇	**100.00** ± 0.00	0.00 ± 0.00	26.25 ± 21.72	**100.00** ± 0.00	0.62 ± 1.40
	8m	**100.00** ± 0.00	0.00 ± 0.00	50.62 ± 33.25	98.75 ± 2.80	3.75 ± 8.39
	5m	**97.50** ± 2.61	0.00 ± 0.00	92.50 ± 9.53	26.25 ± 20.32	6.25 ± 13.98
	4m	75.62 ± 21.58	0.00 ± 0.00	**100.00** ± 0.00	3.12 ± 6.99	0.62 ± 1.40
	3m	64.38 ± 15.40	0.00 ± 0.00	**100.00** ± 0.00	0.00 ± 0.00	2.50 ± 4.07
Average		**87.50** ± 6.54	0.00 ± 0.00	73.87 ± 10.56	45.62 ± 4.73	2.75 ± 5.19

◇ denotes the source task.

E.1.2 Design Choices in Reconstructor

TasQ extracts effective skills by reconstructing the next local observation. To investigate the role of several design choices in the reconstructor, such as the reconstruction target o_{t+1}, the temporal latent variable h, the skill vector z, and the stop-gradient operation, we conduct additional ablation studies. Specifically, we evaluate five variants on the marine-single and marine-single-inv task sets: (i) Reconstructing the current observation instead of the next (Rec o_t); (ii) Removing the temporal latent variable (w/o h); (iii) Not feeding the skill vector into the Q-network (No z to Q); (iv) Performing reconstruction only without extracting any skill vector (Only Rec); and (v) Enabling gradient flow between the Q-network and the skill-discovery module (w/o stop-gradient).

Apart from the modifications specific to each variant, all other components (e.g., the skill-discovery module and the LVC term) and hyperparameters remain the same as in TasQ. We report the average performance of these variants across both task sets in Table 13.

It can be observed that the complete TasQ achieves the highest overall performance, and skill extraction is critical to its superior performance: all variants that incorporate the skill vector into the Q-function outperform those that do not. Even when the skill vector is excluded from the Q-function, incorporating it into the reconstruction process still helps extract more informative individual features. At the same time, predicting the next local observation and incorporating temporal latent

variables indeed help TasQ better capture generalizable decision-making patterns. Moreover, truncating the gradients between the Q-network and the skill-discovery module further helps prevent mutual interference between them. These results further indicate that TasQ's improvements stem from its overall architectural design rather than from any single component.

In addition, we validate our decoder design choice of excluding h on more task sets. As shown in Table 14, excluding h from the decoder tends to be more beneficial on the more complex tasks, while having only a minor impact on simpler ones, and it does not change our main performance trends.

Table 13: The average win rates of different architectural variants of TasQ. The reported results are averaged over 5 random seeds. The bold number denotes the best performance, and $\pm$ denotes one standard deviation.

Task Set	TasQ	Rec o_t	w/o h	No z to Q	Only Rec	w/o stop-gradient
Marine-Single	62.00 $\pm$ 13.80	54.88 $\pm$ 15.27	**62.25** $\pm$ 16.70	62.13 $\pm$ 15.17	40.88 $\pm$ 15.17	59.13 $\pm$ 14.42
Marine-Single-Inv	**87.50** $\pm$ 6.54	87.38 $\pm$ 4.06	81.38 $\pm$ 8.69	76.25 $\pm$ 9.60	64.63 $\pm$ 23.69	84.38 $\pm$ 8.24
Average	**74.75** $\pm$ 6.57	71.13 $\pm$ 8.87	71.94 $\pm$ 4.89	69.19 $\pm$ 9.77	52.75 $\pm$ 14.88	71.75$\pm$11.35

Table 14: The average win rates of different decoder variants on 4 task sets of TasQ. The reported results are averaged over 5 random seeds. The bold number denotes the best performance, and $\pm$ denotes one standard deviation.

Task Set	*Marine-Single*	*Marine-Single-Inv*	*Marine-Hard-Medium*	*Stalker-Zealot-Medium*
TasQ	62.00 $\pm$ 13.80	**87.50** $\pm$ 6.54	**48.12** $\pm$ 5.66	**40.43** $\pm$ 7.06
Decoder with h	**71.38** $\pm$ 10.52	84.25 $\pm$ 8.68	46.20 $\pm$ 4.56	38.03 $\pm$ 10.12

E.2 Hyperparameter Sensitivity

In Table 15, we separately adjusted one of the weights of the CQL regularization α and the LVC η while keeping the other consistent with that in the evaluation. It can be observed that the performance of TasQ remains relatively stable and insensitive to the values of these two hyperparameters on the informative *Marine-Single-Inv* task set. In contrast, on the more limited *Marine-Single* task set, hyperparameter tuning requires slightly more care. However, it is not particularly challenging, as it is reasonable to adopt a fairly conservative α on expert datasets and apply less LVC regularization η when dealing with a small number of agents, given the lower level of accumulated error compared to *Marine-Single-Inv*.

In addition, combining these results with the component ablation in Table 12 characterizes η over its full range: performance peaks at a small weight and degrades toward both $\eta = 0$ (i.e., w/o LVC) and $\eta = 1$ (i.e., w/o TD). LVC is thus necessary, but works best as a lightweight calibrator, with value optimization remaining the main learning driver.

Table 15: Win rates of TasQ with different hyperparameters. The reported results are averaged over 5 random seeds. The $\pm$ denotes one standard deviation.

Hyperparameters		$\eta = 0.7$	$\eta = 0.5$	$\eta = 0.3$
Average	*Marine-Single*	45.50 $\pm$ 9.32	62.00 $\pm$ 13.80	62.50 $\pm$ 16.28
	Marine-Single-Inv	83.50 $\pm$ 8.67	84.00 $\pm$ 9.64	87.50 $\pm$ 6.54
Hyperparameters		$\alpha = 0.5$	$\alpha = 1.0$	$\alpha = 2.5$
Average	*Marine-Single*	51.88 $\pm$ 12.52	62.00 $\pm$ 13.80	56.12 $\pm$ 13.45
	Marine-Single-Inv	81.12 $\pm$ 4.43	87.50 $\pm$ 6.54	78.88 $\pm$ 10.69



Current Timestep **Next Timestep**

Figure 5: In the *12m* task, the agents first approach the enemy through the *advancing* action and then execute the *firing* action in the next timestep.

F SKILL-DISCOVERY VISUALIZATION

F.1 SKILL VISUALIZATION FOR TASQ

We visualize the discovered skills as follows: First, we deploy the TasQ agent, which has been trained on the *Marine-Single* task set, in the *3m* and *12m* tasks. Then, we run one episode in each task and record the actions and skills of each agent at each time step, while saving replay videos. Next, we collect the skill z of every agent at every time step from both trajectories as samples. Finally, we apply t-SNE (Van der Maaten & Hinton, 2008) to project them into a 2D plane and normalize the two dimensions to align their scales.

In Figure 4c, the markers represent the skills of agents at corresponding time steps and tasks (Figure 4a and Figure 4b) after dimensionality reduction. Red markers represent skills from the *12m* task, green markers represent skills from the *3m* task, triangles indicate the ***retreat***, circles indicate the ***attack*** skill, and crosses indicate dead agents.

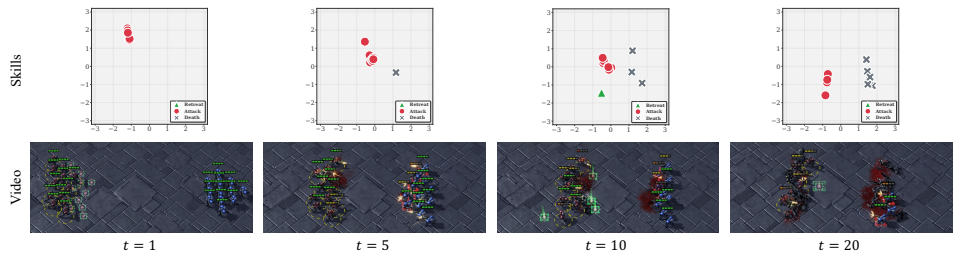
It can be observed that TasQ successfully learns two skills, ***retreat*** and ***attack***, in the *3m* task and applies them effectively in the unseen *12m* task. When the healthy level is low, the agents choose to ***retreat***, while when health is sufficient, they opt to ***attack***, which is consistent with the visualization in Figure 4c. This demonstrates that TasQ indeed discovers effective skills. Notably, in *12m*, agents adopting the ***attack*** skill exhibit two specific actions: *firing* and *advancing*, further supporting that TasQ extracts cross-task generalizable decision structures instead of mimicking specific actions.

Additionally, to verify that *firing* and *advancing* belong to the ***attack*** skill, we present in Figure 5 the next-step actions of the marine units that have just executed the *advancing* action. It can be observed that after moving forward to approach the enemies, they immediately perform a *firing* action. This supports the interpretation in Figure 4b that the *advancing* action is also part of the ***attack*** skill. It also indicates that TasQ is capable of extracting a certain general skill (e.g., ***attack***) and flexibly executing different specific actions (e.g., *firing* and *advancing*) based on the situation.

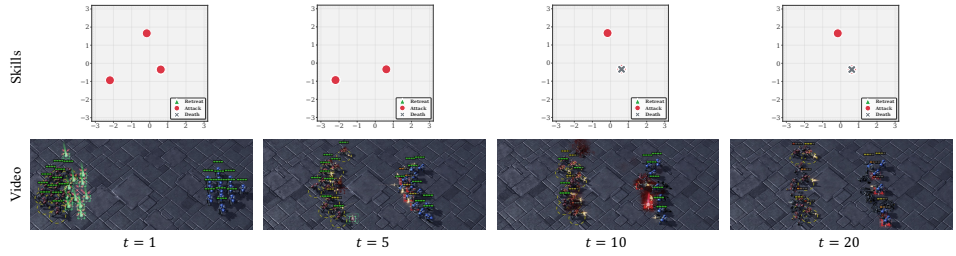
F.2 MULTI-TIMESTEP VISUALIZATION

To illustrate the consistency of TasQ's skill-discovery mechanism across multiple timesteps, we present in Figure 6a the game replays at different moments of the *12m* task, alongside the 2D representations of the skills selected by each agent. Since HiSSD employs both common and task-specific skills, making a direct comparison with TasQ's skills less straightforward, we instead visualize the skills and corresponding game replays of ODIS in Figure 6b using a method similar to that described in Appendix F.1.

By comparing the visualization results of TasQ and ODIS in Figure 6, we observe that, due to the limited expressiveness of predefined discrete skills, ODIS fails to distinguish between dead and surviving agents. Moreover, we do not observe any agents performing retreat-like actions, which undermines the performance of ODIS. In contrast, the continuous skill vectors discovered by TasQ in the latent space offer much higher expressiveness, successfully differentiating agents in different states. Furthermore, by directly optimizing the skill-conditioned Q-values for specific actions, TasQ

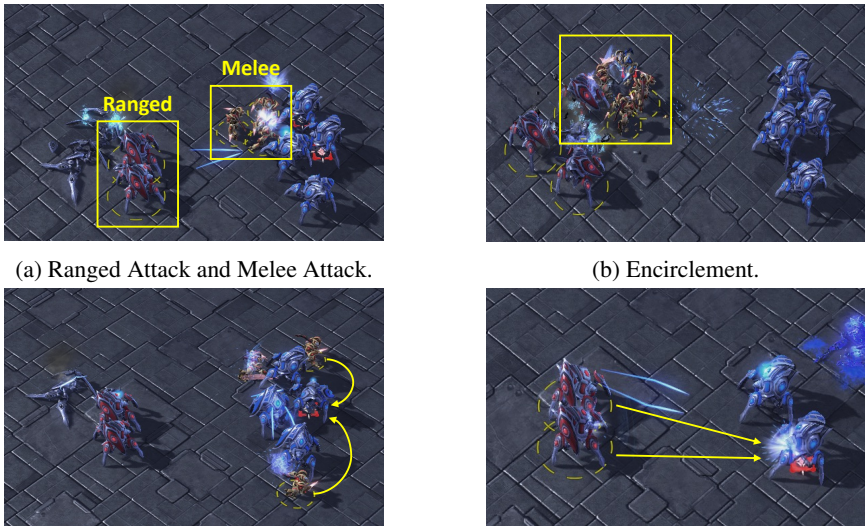


(a) Visualization of TasQ skill representations and game replays across multiple timesteps.



(b) Visualization of ODIS skill representations and game replays across multiple timesteps.

Figure 6: Visualization comparison of TasQ and ODIS across multiple timesteps in the *12m* task.



(a) Ranged Attack and Melee Attack.

(b) Encirclement.

(c) Flanking.

(d) Focus Fire.

Figure 7: Visualization of more complex skills used by TasQ in the *4s3z* task.

can flexibly execute different actions based on the situation after extracting certain skills, further enhancing its generalization performance and task efficiency.

F.3 Visualization for More Complex Skills

We deploy the TasQ policy trained on the heterogeneous *Stalker–Zealot–Expert* task set to the unseen *4s3z* scenario to illustrate the emergence of more complex behaviors. As shown in Figure 7, TasQ effectively extracts rich and diverse behavioral patterns from the source-task datasets, including ranged attack, melee attack, focus fire, flanking, and encirclement, and successfully transfers these skills to the unseen task, demonstrating the expressivity of TasQ's skill discovery module.

F.4 Mutual Information Analysis

We further quantitatively evaluate the quality of the learned skill representations by computing the mutual information (MI) between the latent skill z and the observations $I(O; Z)$, as well as between the latent skill and the actions $I(A; Z)$, on the *Stalker-Zealot-Expert* task set. As shown in Table 16, TasQ yields substantially higher $I(O; Z)$ across all tasks, indicating that z captures distinct and diverse behavioral patterns from observations. Furthermore, TasQ also achieves significantly higher $I(A; Z)$, providing strong statistical evidence that the learned skills are fundamentally coupled with actual behavioral execution rather than merely serving as latent embeddings.

Table 16: Mutual information $I(O; Z)$ and $I(A; Z)$ for TasQ and ODIS on *Stalker-Zealot-Expert* tasks. Higher values indicate that z captures more informative and action-relevant skill representations.

Task	$I(O; Z)$		$I(A; Z)$	
	ODIS	**TasQ**	**ODIS**	**TasQ**
1s3z	0.60	**4.58**	0.54	**1.52**
1s4z	0.00	**4.48**	0.00	**1.73**
1s5z	0.00	**4.86**	0.00	**1.88**
2s3z $^\diamond$	0.32	**4.88**	0.26	**1.73**
2s4z $^\diamond$	0.00	**4.67**	0.00	**1.87**
2s5z	0.00	**3.94**	0.00	**1.80**
3s3z	0.00	**4.17**	0.00	**1.81**
3s4z	0.00	**3.80**	0.00	**1.69**
3s5z $^\diamond$	0.00	**3.90**	0.00	**1.54**
4s3z	0.00	**3.80**	0.00	**1.79**
4s4z	0.00	**3.64**	0.00	**1.78**
4s5z	0.00	**3.84**	0.00	**1.88**
4s6z	0.00	**5.32**	0.00	**2.30**

$\diamond$ denotes the source tasks.

G Additional Results for SMACv2

Table 17: Win rates of SMACv2 tasks. The results are averaged over 5 random seeds. Bold numbers indicate the best performance, and $\pm$ denotes one standard deviation.

Task	**BC-t**	**BC-r**	**CQL**	**ODIS**	**HiSSD**	**TasQ (Ours)**
Zerg-3v3 $^\diamond$	32.50 $\pm$ 11.18	41.88 $\pm$ 6.09	**51.25** $\pm$ 3.56	34.38 $\pm$ 13.62	46.25 $\pm$ 4.64	44.38 $\pm$ 6.77
Zerg-5v5	25.00 $\pm$ 8.84	34.38 $\pm$ 10.60	37.50 $\pm$ 14.49	20.62 $\pm$ 8.73	33.75 $\pm$ 14.56	**38.75** $\pm$ 3.56
Zerg-8v8	12.50 $\pm$ 7.97	21.88 $\pm$ 8.56	26.25 $\pm$ 14.42	16.25 $\pm$ 15.05	18.12 $\pm$ 4.07	**31.25** $\pm$ 9.63
Zerg-10v10	8.12 $\pm$ 10.03	21.88 $\pm$ 4.42	21.25 $\pm$ 13.87	9.38 $\pm$ 8.84	21.88 $\pm$ 14.15	**23.12** $\pm$ 11.18
Average	19.53 $\pm$ 5.50	30.00 $\pm$ 3.07	34.06 $\pm$ 9.09	20.16 $\pm$ 8.23	30.00 $\pm$ 4.70	**34.38** $\pm$ 3.46

$\diamond$ denotes the source task.

We evaluate TasQ on the more challenging SMACv2 (Ellis et al., 2023) benchmark, where in each episode the unit types and spawn locations for both teams are randomly generated. We collect $2,000$ medium-quality trajectories in the SMACv2 *Zerg-3v3* scenario using a policy trained with QMIX, and then use this dataset to train TasQ and other baseline methods in an offline manner. We finally test on *Zerg-3v3* as well as unseen tasks, including *Zerg-5v5*, *Zerg-8v8*, and *Zerg-10v10*. As shown in Table 17, in this more complex setting, value-based methods outperform BC-style approaches, and TasQ further achieves strong performance on unseen tasks by leveraging its skill-discovery mechanism, demonstrating superior average performance and generalization ability.

H Additional Results for Continuous-action Environments

Although TasQ is primarily designed for discrete-action settings, in this section we further investigate whether its generalization capability remains effective when extended to continuous-action environments.

H.1 TasQ for Continuous Action

One intuitive approach to extending TasQ to the continuous-action setting is to retain the skill discovery module and replace the QMIX-based policy optimization with an actor–critic architecture similar to FACMAC (Peng et al., 2021). Specifically, both the actors and the local critics additionally take the latent skill vector z as input, while the global critic is computed by passing all local critic outputs through a QMIX-style mixing network. The LVC regularization term can be adapted by adding a mean-squared error behavior regularizer between the actor's output and the actions in the offline dataset.

H.2 Experiment

Tasks and Datasets We build our experiments on the *simple-spread* environment from the Multi-Agent Particle Environment (MPE) (Lowe et al., 2017). Specifically, we consider four tasks: *SS-2*, *SS-3*, *SS-4*, and *SS-5*, where *SS-N* denotes an environment with N agents and N landmarks, and each agent must navigate to a unique landmark while avoiding collisions. We treat *SS-2* as the source task and use MATD3 (Ackermann et al., 2019) to train a medium-level policy, collecting $2,000$ trajectories as the offline dataset. The remaining three tasks serve as unseen tasks for zero-shot generalization evaluation.

Experiment Setup We use the same baselines as in Section 6, but re-implement their continuous-action versions. For example, for BC-based methods we replace the cross-entropy loss with a mean-squared error objective, and for CQL we adopt the same actor–critic architecture used in the continuous version of TasQ. All algorithms are trained offline for $35,000$ steps on the source-task dataset, with ODIS receiving an additional $15,000$ pretraining steps. During evaluation, each test environment is run for 32 episodes, and we report the average return. We report the average performance of the final policy across five random seeds.

Results As shown in Table 18, even under continuous-action environment and with an intuitive extension, TasQ still achieves the best average performance across tasks. Meanwhile, although BC performs well on the source task, its generalization degrades as the number of agents increases, which matches our observations in discrete-action settings. Furthermore, we would also like to emphasize that achieving effective multi-task offline MARL in continuous-control environments remains an open and challenging future direction.

Table 18: Returns of continuous-action environments. The results are averaged over 5 random seeds. Bold numbers indicate the best performance, and $\pm$ denotes one standard deviation.

Task	BC-t	BC-r	CQL	ODIS	HiSSD	TasQ (Ours)
SS-2 $^\diamond$	**-114.89** $\pm$ 2.25	-168.94 $\pm$ 20.75	-322.61 $\pm$ 36.93	-122.19 $\pm$ 5.20	-126.29 $\pm$ 5.90	-121.07 $\pm$ 2.72
SS-3	-232.62 $\pm$ 5.72	-356.87 $\pm$ 88.11	-641.45 $\pm$ 89.99	-224.62 $\pm$ 7.64	-247.59 $\pm$ 16.98	**-224.45** $\pm$ 12.03
SS-4	-378.52 $\pm$ 38.54	-569.22 $\pm$ 199.68	-1063.86 $\pm$ 157.31	**-354.97** $\pm$ 16.05	-408.62 $\pm$ 60.12	-357.15 $\pm$ 24.98
SS-5	-581.95 $\pm$ 81.39	-838.31 $\pm$ 283.33	-1552.23 $\pm$ 248.58	-535.84 $\pm$ 34.75	-522.73 $\pm$ 36.07	**-494.20** $\pm$ 27.62
Average	-327.00 $\pm$ 25.77	-483.34 $\pm$ 131.68	-895.04 $\pm$ 117.72	-309.41 $\pm$ 9.99	-326.31 $\pm$ 16.15	**-299.22** $\pm$ 14.37

$\diamond$ denotes the source task.

I Discussions

I.1 Assumptions and Scope

We consider the standard multi-task MARL setting, in line with prior work (Hu et al., 2021; Zhang et al., 2023a; Chen et al., 2025; Liu et al., 2025), where tasks, whether in online or offline regimes,

share a similar and well-defined observation structure that enables cross-task generalization of policy models. When this assumption is violated, such as in less structured, vision-based settings, ambiguous or incorrect entity decomposition may hinder performance for existing approaches, including TasQ. This challenge is shared across current multi-task MARL methods. Nevertheless, techniques such as entity segmentation and representation learning can transform noisy or unstructured inputs into structured representations that TasQ can exploit to learn generalizable policies. We leave this as an important direction for future work.

Even under well-defined observations, multi-task offline MARL remains highly challenging and largely open. Our experiments show that existing methods often fall short in this regime, while TasQ delivers substantial gains and establishes a new state of the art.

I.2 Detailed Architecture Comparison with ODIS, HiSSD, and VO-MASD

I.2.1 Comparison with ODIS

ODIS requires a pre-specified number of skills and learns a skill classifier and action decoder by reconstructing actions from global states. During policy training, it freezes the classifier, retrains a local encoder, and uses CQL to choose skills, decoding actions through the decoder.

In contrast, TasQ directly discovers skills in the latent space via a local reconstructor. It then chooses actions using skill-conditioned CQL with local value calibration. This eliminates the need for predefined skill numbers or retraining, and achieves better generalization and task efficiency.

I.2.2 Comparison with HiSSD

HiSSD comprises a high-level planner and a low-level controller. The high-level planner predicts the distribution of the next global state from the current observation and trains an IQL-style value function to weight the likelihood loss, thereby extracting common skills. The low-level controller learns task-specific skills via contrastive learning and generates actions through a BC decoder conditioned on the observation and both types of skills.

Although HiSSD also operates in a latent skill space, it exhibits several limitations compared to TasQ. First, like ODIS, it relies on BC imitation for action generation, limiting its ability to exceed the behavior policy. Second, its contrastive learning mechanism cannot be applied when only a single source task is available. Finally, its complex architecture inflates the parameter count to over four times that of ODIS and TasQ, leading to longer training times and lower efficiency.

In contrast, TasQ employs a simpler yet more effective design: it discovers skills through observation reconstruction and directly optimizes skill-conditioned Q-values with local value calibration, achieving better generalization and task efficiency with significantly fewer parameters and shorter training durations.

I.2.3 Comparison with VO-MASD

VO-MASD targets multi-task MARL by discovering offline skills via an improved VQ-VAE and then training target-task high-level policies online with a hierarchical PPO. This is fundamentally different from our multi-task offline setting, where learning is completed without any online interaction and policies are deployed zero-shot.

Beyond the training protocol, the core technical designs also diverge. VO-MASD reconstructs actions from a predefined number of codebooks, implicitly assuming a fixed discrete skill inventory.

In contrast, TasQ reconstructs the next-step observation in a continuous latent space, enabling flexible and scalable skill discovery without fixing the number of skills, and then optimizes skill-conditioned policies entirely offline. Our results show that, when coupled with next-observation–based offline skill discovery and regularized by local value calibration, value-based methods are not only viable but superior in multi-task offline MARL.

Nevertheless, exploring alternative skill-discovery modules within TasQ, as well as hierarchical offline-to-online extensions, is a promising direction for future work.

I.3 COMPARISON WITH BIKT

Table 19: Win rates of TasQ and BiKT. The reported results are the average performance over all source tasks and unseen tasks, and are averaged over 5 random seeds. Bold numbers indicate the best performance, and ± denotes one standard deviation.

Task Set	Baseline-Best*	BiKT	TasQ (Ours)
marine-easy-expert	**99.4**	99.3	98.2 ± 1.0
marine-easy-medium	71.8	**79.5**	75.0 ± 4.1
marine-easy-medium-replay	72.9	76.9	**78.7** ± 5.7
marine-easy-medium-expert	78.3	77.1	**86.8** ± 5.6
Average	80.6	83.2	**84.7**±2.4
marine-hard-expert	68.0	**75.4**	73.6 ± 1.3
marine-hard-medium	47.6	**51.8**	51.5 ± 4.5
*marine-hard-medium-replay***	51.5	**55.5**	54.4 ± 2.6
*marine-hard-medium-expert***	58.1	64.0	**67.1** ± 3.7
Average	56.3	**61.7**	**61.7**±0.8
stalker-zealot-expert	68.6	75.3	**76.2** ± 3.3
stalker-zealot-medium	23.9	26.1	**44.3** ± 3.5
stalker-zealot-medium-replay	17.6	18.5	**22.9** ± 3.0
stalker-zealot-medium-expert	40.3	46.4	**62.1** ± 6.7
Average	37.6	41.6	**51.4** ± 1.7

∗ The best performance of the baselines involved in Section 6.
∗∗ The original paper does not provide the results on the 13m_vs_15m task.

BiKT (Zhang et al., 2025) leverages the decision transformer (DT) to discover skills at both the individual- and team-level, enabling finer-grained skill-conditioned decision making. While it achieves strong performance, the DT backbone also incurs substantial computational overhead. In contrast, TasQ provides an alternative, lightweight value-based solution.

Moreover, although the comparison is not entirely fair, we evaluate TasQ against BiKT on the Multi-to-Multi tasksets, with results reported in Table 19. Since the source code of BiKT has not been released publicly, we refer to the numbers reported in the original paper. The comparison is therefore restricted to the Multi-to-Multi task sets, as our One-to-Multi task sets are newly designed in this work and no BiKT results are available on them. Meanwhile, we rerun TasQ with the hidden size increased from 128 to 256 to reduce the impact of the parameter-count gap as much as possible.

As shown in the results, despite having fewer parameters, TasQ achieves performance comparable to BiKT and performs better on 7 task sets. Notably, on the more challenging ***heterogeneous*** scenario *stalker-zealot*, TasQ substantially outperforms BiKT.

Remark: *We further compare BiKT and TasQ on source tasks and unseen tasks separately. We find that, on unseen tasks, TasQ outperforms BiKT on* ***10*** *out of 12 task sets, whereas the DT backbone provides BiKT with strong performance on source tasks. Therefore, we believe that combining DT's strong in-task decision-making capability with TasQ's superior cross-task generalization is a promising direction for future work.*

I.4 COMPARISON WITH STAIRS-FORMER

STAIRS-Former (Jeon et al., 2026) carefully designs interleaved spatio-temporal blocks for its transformer backbone and optimizes actions through a value head, but does not employ skill-discovery mechanism. Benefiting from this more elaborate architecture, it attains an average win rate of 67.4% over the twelve Multi-to-Multi task sets, as reported in its original paper, about 5% higher than TasQ's 64.0% in Table 1.

Since its official implementation is publicly available, we further run it on our One-to-Multi Marine task sets under the same protocol as the other baselines, and report the results in Table 20. These results reveal the cost of forgoing skill discovery. When only a single source task is available, STAIRS-Former generalizes markedly worse than TasQ, trailing on every task set it can run, with

Table 20: Win rates of TasQ and STAIRS-Former on One-to-Multi Marine task sets. The reported results are the average performance over all source tasks and unseen tasks, and are averaged over 5 random seeds. Bold numbers indicate the best performance, and $\pm$ denotes one standard deviation.

Task Set	STAIRS-Former	TasQ (Ours)
marine-single	60.00 $\pm$ 10.68	**62.00** $\pm$ 13.80
marine-single-inv	75.38 $\pm$ 3.04	**87.50** $\pm$ 6.54
marine-single-mid	75.05 $\pm$ 9.55	**85.12** $\pm$ 9.23
marine-single-large	OOM*	**70.38** $\pm$ 4.57
Average	52.61	**76.25**

* Counted as 0 when computing the average.

TasQ achieving a 45% higher average performance. Moreover, like HiSSD, its large parameter count causes it to run out of memory on *marine-single-large*, whereas TasQ completes this task set within the same budget. Even excluding the OOM case, TasQ still leads by 11.5% on average across the remaining three task sets.

These results reinforce our central message: rather than pursuing ever more sophisticated architectures, TasQ's lightweight value-based skill-discovery design yields better overall performance and substantially higher task efficiency when source tasks are scarce.

I.5 "Task label" in (Offline) Multi-Task MARL and Conventional RL

The "task label" is an essential concept in multi-task RL, as it explicitly indicates the task an agent is solving and guides policy learning across diverse objectives. Although multi-task MARL (MT-MARL) typically lacks such explicit task labels, the underlying tasks can still be related to those in conventional multi-task RL by abstracting them at two levels. The first level focuses on decision patterns based on the current team state and executing corresponding actions within a specific task, and the second level corresponds to team-level objectives that differ across tasks. The core objective of MT-MARL is to discover and leverage the decision patterns that generalize across tasks to enable multi-task generalization and task efficiency.

Therefore, the concept of "task label" in MT-MARL may share some relationship with these first-level decision patterns, i.e., "skills". As illustrated in Figure 4, the agents' behaviors naturally cluster into distinct skills, and they must infer the appropriate skill from the current observation and accurately execute the corresponding actions, which is fundamentally similar to conventional single-agent multi-task RL, where different task objectives require different action policies.

However, note that the "skills" are not exactly equivalent to "task labels" in the conventional sense. Since task labels in conventional multi-task RL settings are typically discrete, pre-defined, and well-specified, whereas in MT-MARL, the skill boundaries are usually not explicitly available. As shown in Figures 1b and 6, prior methods that predefine the number of skills and discover them through classification struggle to accurately capture the characteristics of each skill, thereby hindering generalization. In contrast, leveraging continuous representations in the latent space enables more effective identification of these decision patterns and improves generalization. This also highlights that the design of TasQ is tailored to address the unique challenges of MT-MARL.

J Detailed Results of Each Task Sets

J.1 Multi-to-Multi Task Sets

J.1.1 Separated Average Performance for Source and Unseen Tasks

In Table 1, we report averages over all tasks to assess each algorithm's overall capability across both source and unseen tasks. For completeness, Table 21 lists the averages on the unseen tasks only. As shown, TasQ again achieves the best performance on 11 out of 12 task sets in this zero-shot setting. We also list the averages on the source tasks only in Table 22, which shows that TasQ achieves

Table 21: Win rates of Multi-to-Multi task sets. The reported results are the average performance over unseen tasks, and are averaged over 5 random seeds. Bold numbers indicate the best performance, and ± denotes one standard deviation.

Task Set	BC-t	BC-r	CQL	ODIS	HiSSD	TasQ (Ours)
marine-easy-e	**99.29** ± 0.46	98.48 ± 0.92	18.84 ± 7.54	64.38 ± 34.86	97.41 ± 1.86	97.86 ± 1.07
marine-easy-m	71.70 ± 6.07	72.50 ± 4.17	8.48 ± 4.12	67.77 ± 5.32	70.54 ± 3.43	**73.66** ± 4.35
marine-easy-mr	48.04 ± 10.78	40.98 ± 10.39	22.59 ± 13.56	6.34 ± 9.86	70.27 ± 3.82	**74.02** ± 5.31
marine-easy-me	71.88 ± 8.00	70.45 ± 2.73	24.11 ± 4.87	63.75 ± 11.41	74.29 ± 4.16	**84.02** ± 7.24
marine-hard-e	57.85 ± 3.91	54.93 ± 2.30	22.08 ± 2.11	14.44 ± 12.06	60.00 ± 5.73	**65.97** ± 5.90
marine-hard-m	38.06 ± 3.57	38.61 ± 3.51	2.99 ± 2.11	28.06 ± 4.23	44.65 ± 3.39	**45.90** ± 6.99
marine-hard-mr	45.35 ± 2.32	46.53 ± 3.02	17.64 ± 3.97	38.75 ± 6.23	45.14 ± 3.59	**50.62** ± 3.59
marine-hard-me	42.71 ± 6.81	49.24 ± 5.97	15.35 ± 2.90	16.53 ± 17.28	52.22 ± 1.02	**56.18** ± 4.48
stalker-zealot-e	56.38 ± 3.97	62.12 ± 4.39	10.00 ± 5.36	49.81 ± 7.58	60.25 ± 9.57	**74.19** ± 4.52
stalker-zealot-m	17.06 ± 2.74	20.81 ± 3.53	11.31 ± 4.72	9.94 ± 8.09	20.31 ± 4.74	**40.94** ± 6.74
stalker-zealot-mr	18.19 ± 5.49	11.50 ± 1.43	11.37 ± 2.21	7.56 ± 5.85	16.31 ± 5.60	**24.56** ± 5.06
stalker-zealot-me	32.06 ± 5.98	31.06 ± 6.38	12.50 ± 3.10	26.06 ± 12.29	34.81 ± 3.39	**63.44** ± 7.50

-e, *-m*, *-mr*, and *-me* represent *-expert*, *-medium*, *-medium-replay*, and *-medium-expert*, respectively.

Table 22: Win rates of Multi-to-Multi task sets. The reported results are the average performance over source tasks, and are averaged over 5 random seeds. Bold numbers indicate the best performance, and ± denotes one standard deviation.

Task Set	BC-t	BC-r	CQL	ODIS	HiSSD	TasQ (Ours)
marine-easy-e	99.58 ± 0.51	**99.79** ± 0.42	59.79 ± 3.40	84.79 ± 19.98	99.17 ± 1.21	98.54 ± 1.06
marine-easy-m	71.67 ± 2.41	70.21 ± 6.64	38.12 ± 13.89	70.21 ± 7.29	73.54 ± 2.76	**80.83** ± 5.57
marine-easy-mr	53.33 ± 9.58	51.46 ± 4.69	45.83 ± 3.49	22.08 ± 11.21	78.96 ± 5.25	**80.00** ± 11.30
marine-easy-me	83.12 ± 6.76	82.71 ± 10.43	55.62 ± 9.52	71.25 ± 12.27	87.71 ± 12.38	**93.54** ± 4.29
marine-hard-e	85.83 ± 5.80	87.08 ± 3.06	33.12 ± 0.78	41.04 ± 25.36	**92.08** ± 1.56	84.79 ± 3.13
marine-hard-m	51.88 ± 6.57	52.50 ± 3.52	27.71 ± 2.68	46.04 ± 4.86	**56.46** ± 0.78	54.79 ± 3.33
marine-hard-mr	51.88 ± 5.72	47.29 ± 6.10	33.75 ± 4.40	32.92 ± 4.96	**54.79** ± 6.51	47.50 ± 5.21
marine-hard-me	**68.96** ± 9.89	60.00 ± 15.29	35.21 ± 1.67	28.75 ± 20.88	56.46 ± 6.50	56.25 ± 3.09
stalker-zealot-e	87.29 ± 5.03	**90.21** ± 2.84	12.71 ± 7.89	72.50 ± 14.86	87.71 ± 2.02	80.83 ± 4.69
stalker-zealot-m	33.33 ± 8.62	34.38 ± 5.74	28.54 ± 7.78	18.75 ± 6.81	34.17 ± 4.90	**38.75** ± 9.80
stalker-zealot-mr	15.83 ± 1.38	15.83 ± 3.32	**25.21** ± 6.70	10.21 ± 9.02	10.62 ± 3.39	18.12 ± 3.52
stalker-zealot-me	64.79 ± 8.77	61.88 ± 13.35	16.46 ± 3.39	46.25 ± 26.19	58.54 ± 10.06	**78.75** ± 7.81

-e, *-m*, *-mr*, and *-me* represent *-expert*, *-medium*, *-medium-replay*, and *-medium-expert*, respectively.

comparable or even superior performance on most task sets. These results further substantiate its generalization ability.

J.1.2 Task-wise Results

We present the detailed results of our evaluation experiments on the Multi-to-Multi task sets. The results for *Marine-Easy* are presented in Table 23, the results for *Marine-Hard* are shown in Table 24, and the results for *Stalker-Zealot* are provided in Table 25. In each table, we present the multi-task evaluation results on datasets of varying quality, with the source tasks used for training marked by "◇". The results show that, despite being trained on a limited number of source tasks, TasQ exhibits strong multi-task generalization, as reflected in its exceptional average performance across all test tasks.

Table 23: Win rates of *Marine-Easy* scenario. The reported results are averaged over 5 random seeds. The bold number denotes the best performance, and ± denotes one standard deviation.

	Task	BC-t	BC-r	CQL	ODIS	HiSSD	TasQ (Ours)
Expert	3m ◇	99.38 ± 1.40	**100.00** ± 0.00	98.13 ± 2.80	97.50 ± 1.40	98.75 ± 2.80	98.75 ± 2.80
	4m	90.62 ± 7.33	96.88 ± 3.83	80.63 ± 13.69	56.25 ± 37.69	92.50 ± 8.73	**97.50** ± 2.61
	5m ◇	**100.00** ± 0.00	**100.00** ± 0.00	81.25 ± 11.48	78.12 ± 31.48	**100.00** ± 0.00	98.75 ± 1.71
	6m	**100.00** ± 0.00	**100.00** ± 0.00	40.00 ± 30.89	55.62 ± 43.49	98.75 ± 1.71	96.88 ± 4.42
	7m	**100.00** ± 0.00	98.75 ± 1.71	10.63 ± 23.76	58.12 ± 44.28	98.75 ± 2.80	97.50 ± 4.07
	8m	99.38 ± 1.40	**100.00** ± 0.00	0.63 ± 1.40	63.75 ± 41.14	**100.00** ± 0.00	98.12 ± 1.71
	9m	**100.00** ± 0.00	**100.00** ± 0.00	0.00 ± 0.00	70.62 ± 42.65	99.38 ± 1.40	**100.00** ± 0.00
	10m ◇	**100.00** ± 0.00	98.75 ± 1.71	0.00 ± 0.00	78.75 ± 35.24	98.75 ± 1.71	98.12 ± 2.80
	11m	99.38 ± 1.40	**100.00** ± 0.00	0.00 ± 0.00	80.00 ± 28.78	98.75 ± 1.71	98.12 ± 4.19
	12m	**100.00** ± 0.00	99.38 ± 1.40	0.00 ± 0.00	66.25 ± 42.01	93.75 ± 9.11	96.88 ± 2.21
	Average	98.87 ± 0.58	**99.38** ± 0.40	31.13 ± 6.36	70.50 ± 30.14	97.94 ± 1.23	98.06 ± 1.00
Medium	3m◇	61.25 ± 10.03	65.62 ± 8.84	**70.63** ± 17.48	64.38 ± 10.96	61.88 ± 5.59	70.62 ± 12.62
	4m	66.25 ± 8.09	64.38 ± 34.42	27.50 ± 30.09	55.00 ± 30.98	**71.88** ± 10.13	52.50 ± 33.18
	5m ◇	78.75 ± 8.09	80.00 ± 4.19	43.75 ± 34.59	76.25 ± 7.19	83.12 ± 9.78	**84.38** ± 4.94
	6m	90.00 ± 6.01	84.38 ± 5.85	24.38 ± 23.43	**93.12** ± 8.39	76.88 ± 12.02	89.38 ± 9.27
	7m	99.38 ± 1.40	90.62 ± 11.69	7.50 ± 9.00	90.62 ± 12.50	95.00 ± 6.48	**100.00** ± 0.00
	8m	86.88 ± 10.69	**95.62** ± 6.48	0.00 ± 0.00	89.38 ± 5.23	93.12 ± 4.07	90.00 ± 6.40
	9m	**83.12** ± 8.73	78.75 ± 5.13	0.00 ± 0.00	72.50 ± 18.67	76.88 ± 7.84	82.50 ± 13.00
	10m ◇	70.62 ± 17.34	69.38 ± 6.77	0.00 ± 0.00	70.00 ± 15.08	75.62 ± 6.01	**87.50** ± 6.25
	11m	43.75 ± 13.07	43.75 ± 5.85	0.00 ± 0.00	43.12 ± 9.73	46.25 ± 12.18	**55.00** ± 10.27
	12m	38.12 ± 7.78	44.38 ± 5.59	0.00 ± 0.00	30.62 ± 12.38	33.75 ± 5.59	**46.25** ± 10.69
	Average	71.81 ± 3.74	71.69 ± 4.33	17.38 ± 7.31	68.50 ± 5.74	71.44 ± 2.77	**75.81** ± 1.73
Medium-Replay	3m ◇	80.00 ± 6.48	70.62 ± 22.81	71.88 ± 11.05	60.62 ± 35.05	**82.50** ± 9.53	78.75 ± 16.45
	4m	75.00 ± 15.15	70.00 ± 9.53	74.38 ± 9.48	18.75 ± 25.77	69.38 ± 21.13	**79.38** ± 7.53
	5m ◇	77.50 ± 31.51	88.75 ± 15.56	62.50 ± 14.82	73.75 ± 26.01	86.25 ± 7.19	**91.25** ± 9.73
	6m	88.75 ± 23.45	98.12 ± 2.80	40.63 ± 41.93	12.50 ± 27.95	**100.00** ± 0.00	99.38 ± 1.40
	7m	86.88 ± 24.15	90.62 ± 12.88	26.25 ± 36.01	8.12 ± 18.17	**100.00** ± 0.00	**100.00** ± 0.00
	8m	19.38 ± 29.43	43.75 ± 35.84	9.38 ± 20.96	0.00 ± 0.00	**96.25** ± 5.13	89.38 ± 10.27
	9m	15.62 ± 18.88	25.00 ± 24.51	3.75 ± 8.39	5.00 ± 11.18	76.25 ± 4.74	**88.75** ± 4.19
	10m ◇	**80.62** ± 12.18	76.88 ± 17.62	3.13 ± 6.99	76.88 ± 11.82	68.12 ± 10.92	70.00 ± 39.26
	11m	1.25 ± 2.80	5.00 ± 8.15	3.13 ± 6.99	0.00 ± 0.00	24.38 ± 13.15	**35.62** ± 19.96
	12m	0.00 ± 0.00	3.75 ± 6.77	0.63 ± 1.40	0.00 ± 0.00	**25.62** ± 23.74	**25.62** ± 15.05
	Average	44.12 ± 8.19	49.62 ± 9.17	29.56 ± 11.73	11.06 ± 8.94	72.88 ± 3.16	**75.81** ± 6.83
Medium-Expert	3m ◇	90.00 ± 18.93	83.75 ± 23.94	**96.88** ± 3.13	63.12 ± 27.90	90.62 ± 20.96	**96.88** ± 3.12
	4m	**87.50** ± 9.11	78.12 ± 14.82	55.63 ± 24.55	49.38 ± 37.72	76.25 ± 12.62	72.50 ± 19.19
	5m ◇	69.38 ± 9.22	77.50 ± 10.69	70.00 ± 29.03	5.00 ± 11.18	88.12 ± 14.89	**96.25** ± 6.77
	6m	63.75 ± 25.92	68.75 ± 20.73	80.00 ± 15.08	50.00 ± 9.88	55.00 ± 15.08	**93.75** ± 6.25
	7m	85.62 ± 6.85	86.25 ± 16.33	31.25 ± 17.26	65.62 ± 22.53	86.25 ± 7.84	**90.62** ± 11.05
	8m	56.88 ± 11.14	66.25 ± 24.65	1.88 ± 4.19	66.25 ± 23.84	76.25 ± 8.15	**90.00** ± 9.73
	9m	70.62 ± 12.62	75.00 ± 22.43	0.00 ± 0.00	73.12 ± 18.43	**90.62** ± 5.85	87.50 ± 10.60
	10m ◇	5.00 ± 6.85	11.88 ± 12.96	0.00 ± 0.00	0.62 ± 1.40	84.38 ± 16.83	**87.50** ± 11.90
	11m	68.12 ± 14.22	71.88 ± 8.56	0.00 ± 0.00	76.88 ± 18.83	75.00 ± 19.52	**81.88** ± 17.73
	12m	60.62 ± 12.81	56.88 ± 4.64	0.00 ± 0.00	65.00 ± 16.00	60.62 ± 12.81	**71.88** ± 18.09
	Average	74.12 ± 3.21	75.25 ± 7.41	33.56 ± 6.47	66.00 ± 9.21	78.31 ± 4.99	**86.88** ± 5.68

◇ denotes the source tasks.

Table 24: Win rates of *Marine-Hard* scenario. The reported results are averaged over 5 random seeds. The bold number denotes the best performance, and ± denotes one standard deviation.

	Task	BC-t	BC-r	CQL	ODIS	HiSSD	TasQ (Ours)
Expert	3m ◇	**100.00** ± 0.00	**100.00** ± 0.00	98.12 ± 2.80	74.38 ± 34.47	98.12 ± 1.71	99.38 ± 1.40
	4m	92.50 ± 8.44	95.00 ± 7.19	92.50 ± 7.19	37.50 ± 32.40	98.75 ± 1.71	**99.38** ± 1.40
	5m	97.50 ± 2.61	83.75 ± 12.38	96.25 ± 5.13	30.62 ± 32.73	**100.00** ± 0.00	99.38 ± 1.40
	10m	95.62 ± 3.56	91.25 ± 13.15	7.50 ± 13.55	18.12 ± 12.96	**99.38** ± 1.40	95.00 ± 9.53
	12m	68.75 ± 31.25	**88.75** ± 10.73	1.88 ± 4.19	5.00 ± 11.18	53.75 ± 50.70	74.38 ± 38.04
	5m6m ◇	62.50 ± 8.84	67.50 ± 8.44	0.62 ± 1.40	25.00 ± 29.06	**79.38** ± 6.48	60.00 ± 9.73
	7m8m	30.62 ± 17.73	23.12 ± 19.84	0.62 ± 1.40	8.75 ± 11.14	**37.50** ± 9.38	**37.50** ± 16.24
	8m9m	41.88 ± 6.85	43.75 ± 19.01	0.00 ± 0.00	15.00 ± 15.84	53.12 ± 16.09	**87.50** ± 14.32
	9m10m ◇	**98.75** ± 2.80	90.00 ± 12.38	0.62 ± 1.40	23.75 ± 36.55	**98.75** ± 1.71	95.00 ± 1.71
	10m11m	63.75 ± 7.84	**86.25** ± 8.44	0.00 ± 0.00	12.50 ± 14.66	83.75 ± 16.89	75.62 ± 9.73
	10m12m	3.75 ± 2.61	8.12 ± 5.68	0.00 ± 0.00	1.88 ± 4.19	10.00 ± 4.64	**23.12** ± 16.77
	13m15m	0.00 ± 0.00	0.62 ± 1.40	0.00 ± 0.00	0.62 ± 1.40	**3.75** ± 5.13	1.88 ± 2.80
	Average	62.97 ± 2.18	64.84 ± 3.24	24.84 ± 1.53	21.09 ± 14.72	68.02 ± 4.47	**70.68** ± 4.80
Medium	3m ◇	62.50 ± 9.38	52.50 ± 15.21	**81.25** ± 6.63	59.38 ± 7.97	64.38 ± 5.23	75.62 ± 8.95
	4m	41.25 ± 23.32	41.88 ± 25.92	8.12 ± 4.74	58.12 ± 24.37	**76.88** ± 7.53	71.88 ± 7.97
	5m	86.88 ± 6.01	85.62 ± 11.40	18.75 ± 17.26	68.12 ± 30.49	90.62 ± 12.69	**96.25** ± 5.59
	10m	90.00 ± 12.77	90.00 ± 12.38	0.00 ± 0.00	70.00 ± 32.67	93.75 ± 3.12	**96.25** ± 8.39
	12m	75.00 ± 21.31	64.38 ± 30.51	1.88 ± 2.80	18.12 ± 23.74	**79.38** ± 9.53	72.50 ± 21.81
	5m6m ◇	30.62 ± 2.61	**38.75** ± 6.48	0.00 ± 0.00	10.00 ± 11.98	30.63 ± 11.35	30.00 ± 10.50
	7m8m	3.12 ± 3.12	**10.00** ± 8.67	0.00 ± 0.00	4.38 ± 3.56	8.75 ± 10.92	**10.00** ± 13.51
	8m9m	6.88 ± 7.78	6.25 ± 7.65	0.00 ± 0.00	11.25 ± 13.18	13.12 ± 6.01	**15.00** ± 18.01
	9m10m ◇	64.38 ± 14.76	64.38 ± 12.62	0.00 ± 0.00	68.75 ± 15.93	**74.38** ± 10.22	58.75 ± 16.15
	10m11m	41.88 ± 11.18	43.12 ± 16.30	0.00 ± 0.00	21.25 ± 9.73	38.12 ± 20.66	**50.62** ± 23.11
	10m12m	0.62 ± 1.40	0.62 ± 1.40	0.00 ± 0.00	0.00 ± 0.00	**1.25** ± 2.80	0.00 ± 0.00
	13m15m	**1.88** ± 1.71	0.62 ± 1.40	0.00 ± 0.00	1.25 ± 2.80	0.00 ± 0.00	0.62 ± 1.40
	Average	42.08 ± 2.63	41.51 ± 3.53	9.17 ± 1.78	32.55 ± 4.31	47.60 ± 2.66	**48.12** ± 5.66
Medium-Replay	3m ◇	75.62 ± 7.13	80.00 ± 17.06	81.88 ± 6.77	75.00 ± 13.62	84.38 ± 8.56	**85.00** ± 10.22
	4m	83.12 ± 10.03	83.12 ± 8.44	57.50 ± 25.64	55.62 ± 34.40	75.00 ± 8.27	**91.88** ± 9.78
	5m	95.62 ± 5.23	95.62 ± 9.78	89.38 ± 11.61	93.12 ± 8.39	83.75 ± 13.51	**100.00** ± 0.00
	10m	87.50 ± 11.27	90.62 ± 11.05	3.12 ± 6.99	89.38 ± 8.44	**93.12** ± 10.22	86.88 ± 7.78
	12m	85.62 ± 10.96	88.12 ± 4.07	1.25 ± 2.80	70.62 ± 16.33	88.12 ± 10.22	**94.38** ± 4.07
	5m6m ◇	30.00 ± 13.55	**33.12** ± 6.48	18.75 ± 9.63	9.38 ± 6.25	29.38 ± 13.37	13.75 ± 5.23
	7m8m	**14.38** ± 2.80	3.75 ± 2.61	6.25 ± 4.94	5.00 ± 7.84	9.38 ± 7.65	8.12 ± 7.19
	8m9m	8.12 ± 3.56	11.88 ± 5.59	1.25 ± 1.71	1.88 ± 1.71	12.50 ± 4.94	**26.88** ± 10.96
	9m10m ◇	36.25 ± 14.08	42.50 ± 14.76	0.62 ± 1.40	14.38 ± 13.18	**50.62** ± 13.51	43.75 ± 12.30
	10m11m	36.25 ± 8.15	29.38 ± 18.57	0.00 ± 0.00	32.50 ± 22.27	**40.00** ± 13.69	38.12 ± 17.17
	10m12m	**2.50** ± 2.61	0.62 ± 1.40	0.00 ± 0.00	0.00 ± 0.00	0.62 ± 1.40	1.25 ± 2.80
	13m15m	5.62 ± 2.61	5.00 ± 6.48	0.00 ± 0.00	0.62 ± 1.40	3.75 ± 4.07	**8.12** ± 2.80
	Average	46.72 ± 2.35	46.98 ± 1.95	21.67 ± 2.83	37.29 ± 5.48	47.55 ± 3.56	**49.84** ± 3.36
Medium-Expert	3m ◇	74.38 ± 14.72	85.00 ± 19.44	**100.00** ± 0.00	53.75 ± 36.94	90.62 ± 19.26	98.75 ± 1.71
	4m	**95.00** ± 3.56	85.00 ± 23.11	57.50 ± 29.86	34.38 ± 45.93	94.38 ± 4.07	92.50 ± 2.80
	5m	91.88 ± 9.78	58.12 ± 28.00	80.62 ± 13.15	31.25 ± 38.84	**98.75** ± 2.80	93.12 ± 10.92
	10m	93.75 ± 10.83	80.00 ± 19.71	0.00 ± 0.00	26.25 ± 34.13	**95.62** ± 1.71	95.00 ± 5.68
	12m	59.38 ± 43.24	73.12 ± 26.85	0.00 ± 0.00	3.75 ± 5.59	86.25 ± 13.73	**90.00** ± 6.77
	5m6m ◇	**43.12** ± 22.14	37.50 ± 18.62	5.62 ± 5.59	8.75 ± 8.09	42.50 ± 23.03	29.38 ± 17.06
	7m8m	18.12 ± 4.64	16.25 ± 14.72	0.00 ± 0.00	4.38 ± 6.85	**21.25** ± 8.67	19.38 ± 21.01
	8m9m	31.25 ± 19.39	26.25 ± 18.96	0.00 ± 0.00	15.00 ± 21.58	28.12 ± 11.05	**39.38** ± 26.57
	9m10m ◇	62.50 ± 24.90	**84.38** ± 9.38	0.00 ± 0.00	23.75 ± 35.81	36.25 ± 20.20	40.62 ± 21.42
	10m11m	51.88 ± 17.34	41.88 ± 19.21	0.00 ± 0.00	33.75 ± 38.87	42.50 ± 4.74	**61.88** ± 20.18
	10m12m	1.88 ± 4.19	2.50 ± 4.07	0.00 ± 0.00	0.00 ± 0.00	2.50 ± 5.59	**4.38** ± 9.78
	13m15m	0.00 ± 0.00	1.25 ± 1.71	0.00 ± 0.00	0.00 ± 0.00	0.62 ± 1.40	**10.00** ± 10.69
	Average	51.93 ± 7.21	49.27 ± 5.16	20.31 ± 1.97	19.58 ± 17.63	53.28 ± 1.88	**56.20** ± 3.21

◇ denotes the source tasks.
"XmYm" represents "Xm_vs_Ym".

Table 25: Win rates of *Stalker-Zealot* scenario. The reported results are averaged over 5 random seeds. The bold number denotes the best performance, and ± denotes one standard deviation.

	Task	BC-t	BC-r	CQL	ODIS	HiSSD	TasQ (Ours)
Expert	1s3z	51.88 ± 28.26	65.00 ± 7.46	70.00 ± 39.57	41.25 ± 38.55	65.62 ± 19.76	**74.38** ± 31.59
	1s4z	34.38 ± 10.83	**55.00** ± 31.14	25.62 ± 20.30	19.38 ± 29.18	52.50 ± 37.33	53.75 ± 28.07
	1s5z	24.38 ± 28.42	19.38 ± 25.81	1.25 ± 1.71	30.62 ± 40.41	18.75 ± 22.10	**60.00** ± 24.35
	2s3z ◇	**95.62** ± 2.80	95.00 ± 4.74	36.88 ± 25.33	77.50 ± 25.14	94.38 ± 7.78	85.62 ± 10.50
	2s4z ◇	**81.25** ± 7.33	76.88 ± 8.44	1.25 ± 1.71	53.12 ± 24.61	73.75 ± 5.68	71.88 ± 13.80
	2s5z	**66.25** ± 23.43	65.00 ± 18.54	0.00 ± 0.00	48.12 ± 20.44	61.88 ± 17.73	63.75 ± 10.50
	3s3z	**91.25** ± 5.59	76.88 ± 19.47	3.12 ± 5.41	81.88 ± 15.53	72.50 ± 23.53	90.00 ± 8.39
	3s4z	93.12 ± 4.64	78.75 ± 6.01	0.00 ± 0.00	81.88 ± 8.95	93.12 ± 4.07	**95.62** ± 5.23
	3s5z ◇	93.75 ± 5.85	90.00 ± 8.39	0.00 ± 0.00	86.88 ± 5.13	**95.00** ± 4.74	85.00 ± 11.14
	4s3z	77.50 ± 17.31	78.12 ± 16.97	0.00 ± 0.00	71.25 ± 19.06	83.75 ± 15.05	**90.00** ± 9.48
	4s4z	66.25 ± 16.89	52.50 ± 8.39	0.00 ± 0.00	52.50 ± 20.89	60.62 ± 34.28	**74.38** ± 13.51
	4s5z	54.38 ± 12.81	38.75 ± 3.56	0.00 ± 0.00	35.00 ± 27.99	52.50 ± 22.90	**71.88** ± 17.95
	4s6z	61.88 ± 10.92	34.38 ± 13.62	0.00 ± 0.00	36.25 ± 21.83	41.25 ± 21.36	**68.12** ± 14.89
	Average	68.61 ± 3.93	63.51 ± 2.64	10.62 ± 5.88	55.05 ± 9.11	66.59 ± 7.77	**75.72** ± 3.73
Medium	1s3z	3.12 ± 3.83	14.38 ± 25.73	36.25 ± 25.25	10.62 ± 22.05	35.00 ± 22.69	**64.38** ± 29.70
	1s4z	18.75 ± 25.86	11.88 ± 24.84	43.75 ± 29.89	7.50 ± 8.15	13.75 ± 5.23	**48.75** ± 37.80
	1s5z	5.62 ± 6.77	3.75 ± 5.13	11.88 ± 13.33	0.62 ± 1.40	9.38 ± 6.25	**51.25** ± 38.76
	2s3z ◇	49.38 ± 13.51	48.12 ± 4.74	**62.50** ± 14.32	38.12 ± 11.57	50.62 ± 10.22	30.63 ± 23.94
	2s4z ◇	11.88 ± 8.09	8.75 ± 13.15	23.12 ± 16.03	8.75 ± 7.78	9.38 ± 7.65	**29.38** ± 16.92
	2s5z	13.75 ± 10.50	16.25 ± 8.67	1.25 ± 2.80	18.75 ± 27.15	14.37 ± 6.48	**42.50** ± 23.24
	3s3z	39.38 ± 15.56	26.25 ± 15.56	16.25 ± 22.25	21.88 ± 20.01	31.87 ± 16.30	**45.00** ± 12.62
	3s4z	46.25 ± 17.59	23.75 ± 12.22	3.75 ± 8.39	18.12 ± 17.59	45.62 ± 14.59	**64.38** ± 28.35
	3s5z ◇	41.88 ± 9.78	43.12 ± 18.14	0.00 ± 0.00	9.38 ± 9.38	42.50 ± 16.62	**56.25** ± 9.11
	4s3z	37.50 ± 26.42	42.50 ± 18.57	0.00 ± 0.00	11.88 ± 8.67	33.12 ± 13.91	**48.12** ± 16.92
	4s4z	**23.75** ± 5.23	16.25 ± 14.56	0.00 ± 0.00	8.12 ± 10.03	7.50 ± 3.56	18.12 ± 10.69
	4s5z	**10.62** ± 5.68	10.00 ± 9.73	0.00 ± 0.00	0.62 ± 1.40	4.38 ± 4.19	**10.62** ± 7.84
	4s6z	9.38 ± 4.94	5.62 ± 4.64	0.00 ± 0.00	1.25 ± 2.80	8.12 ± 4.74	**16.25** ± 11.35
	Average	23.94 ± 2.72	20.82 ± 3.78	15.29 ± 4.89	11.97 ± 7.53	23.51 ± 3.76	**40.43** ± 7.06
Medium-Replay	1s3z	17.50 ± 11.40	33.12 ± 20.32	18.75 ± 14.66	3.12 ± 5.41	**55.62** ± 26.46	21.25 ± 14.39
	1s4z	7.50 ± 6.09	13.75 ± 13.00	**32.50** ± 16.18	10.62 ± 10.73	15.00 ± 11.98	13.75 ± 24.06
	1s5z	3.75 ± 4.07	3.75 ± 8.39	**10.00** ± 9.48	5.00 ± 7.19	**10.00** ± 13.87	8.75 ± 8.39
	2s3z ◇	12.50 ± 6.63	6.88 ± 5.59	**53.75** ± 13.51	12.50 ± 16.39	12.50 ± 6.99	7.50 ± 2.80
	2s4z ◇	8.12 ± 9.53	10.62 ± 5.68	**18.12** ± 11.98	12.50 ± 13.44	5.62 ± 4.07	6.25 ± 4.42
	2s5z	11.25 ± 9.27	**22.50** ± 24.25	9.38 ± 8.27	10.62 ± 14.59	9.38 ± 6.25	17.50 ± 15.08
	3s3z	13.75 ± 12.62	12.50 ± 4.94	22.50 ± 7.13	12.50 ± 13.44	18.75 ± 24.11	**57.50** ± 15.87
	3s4z	27.50 ± 17.46	45.00 ± 19.09	13.12 ± 11.57	8.12 ± 9.00	20.62 ± 10.50	**47.50** ± 32.43
	3s5z ◇	26.88 ± 13.91	30.00 ± 11.18	3.75 ± 5.13	5.62 ± 3.42	13.75 ± 7.84	**40.62** ± 8.27
	4s3z	11.25 ± 18.57	**23.75** ± 19.59	4.38 ± 5.23	13.75 ± 16.18	15.00 ± 18.67	**23.75** ± 21.49
	4s4z	12.50 ± 9.11	18.75 ± 15.15	2.50 ± 3.42	3.75 ± 3.42	11.88 ± 8.09	**28.12** ± 15.93
	4s5z	6.25 ± 3.83	5.00 ± 6.48	0.00 ± 0.00	5.00 ± 8.15	1.88 ± 2.80	**16.88** ± 9.00
	4s6z	3.75 ± 5.13	3.75 ± 4.07	0.62 ± 1.40	3.12 ± 3.83	5.00 ± 1.71	**10.62** ± 6.09
	Average	12.50 ± 1.75	17.64 ± 4.40	14.57 ± 2.02	8.17 ± 6.08	15.00 ± 4.02	**23.08** ± 3.38
Medium-Expert	1s3z	47.50 ± 39.12	25.00 ± 30.22	**80.00** ± 20.20	49.38 ± 36.74	58.75 ± 35.93	73.75 ± 40.42
	1s4z	5.00 ± 3.56	13.12 ± 9.73	43.12 ± 18.67	6.88 ± 15.37	10.00 ± 6.01	**72.50** ± 27.72
	1s5z	11.88 ± 23.22	10.62 ± 20.32	1.88 ± 1.71	0.62 ± 1.40	6.25 ± 12.30	**72.50** ± 17.73
	2s3z ◇	80.62 ± 21.47	71.25 ± 10.22	49.38 ± 11.35	59.38 ± 35.63	68.75 ± 14.99	**86.25** ± 10.50
	2s4z ◇	37.50 ± 32.10	61.25 ± 15.87	0.00 ± 0.00	47.50 ± 31.44	56.25 ± 24.71	**62.50** ± 19.64
	2s5z	25.62 ± 19.94	21.88 ± 11.48	0.00 ± 0.00	27.50 ± 18.14	23.12 ± 9.53	**40.62** ± 13.80
	3s3z	47.50 ± 37.85	56.88 ± 11.57	0.00 ± 0.00	55.00 ± 29.20	63.75 ± 29.37	**71.25** ± 16.00
	3s4z	60.00 ± 17.87	74.38 ± 12.77	0.00 ± 0.00	50.00 ± 30.78	72.50 ± 16.15	**93.12** ± 3.42
	3s5z ◇	67.50 ± 17.76	61.88 ± 12.18	0.00 ± 0.00	31.88 ± 24.45	50.62 ± 18.01	**87.50** ± 6.99
	4s3z	62.50 ± 18.62	**70.62** ± 7.84	0.00 ± 0.00	39.38 ± 27.12	55.00 ± 24.27	55.00 ± 28.86
	4s4z	30.62 ± 5.59	35.00 ± 23.11	0.00 ± 0.00	19.38 ± 12.77	41.25 ± 21.36	**61.88** ± 20.54
	4s5z	10.00 ± 5.13	6.88 ± 10.22	0.00 ± 0.00	6.88 ± 2.61	7.50 ± 6.48	**49.38** ± 26.28
	4s6z	10.00 ± 7.46	6.25 ± 5.85	0.00 ± 0.00	5.62 ± 7.78	10.00 ± 5.13	**44.38** ± 19.94
	Average	38.17 ± 5.38	39.62 ± 2.98	13.41 ± 2.53	30.72 ± 14.65	40.29 ± 4.64	**66.97** ± 7.01

◇ denotes the source tasks.

J.2 ONE-TO-MULTI TASK SETS

We present the detailed results of our evaluation experiments on the One-to-Multi Marine task sets in Table 26. The source tasks used for training are marked by "◇". The results show that, despite being trained on only one source task, TasQ exhibits better multi-task generalization. For *Marine-Single-Large*, TasQ is competitive with the baselines on the scales close to the source (*25m–30m*), while its advantage becomes evident on the more distant *10m*, where it reaches 58.13 and all baselines stay below 46. The detailed results of the *SMACv2* and *Continuous-MPE* task sets are reported in Appendices G and H, respectively.

Table 26: Win rates of One-to-Multi Marine task sets. The reported results are averaged over 5 random seeds. The bold number denotes the best performance, and ± denotes one standard deviation.

Task Set		BC-t	BC-r	CQL	ODIS	HiSSD	TasQ (Ours)
Marine Single	3m ◇	**100.00** ± 0.00	96.88 ± 4.42	**100.00** ± 0.00	99.38 ± 1.40	**100.00** ± 0.00	**100.00** ± 0.00
	5m	81.88 ± 9.22	67.50 ± 27.30	88.12 ± 7.46	37.50 ± 33.15	66.88 ± 32.52	**91.25** ± 10.92
	8m	38.75 ± 22.38	35.62 ± 29.37	40.00 ± 26.00	11.88 ± 21.47	31.87 ± 41.01	**49.38** ± 23.63
	10m	20.62 ± 22.38	16.25 ± 20.54	34.38 ± 25.39	11.25 ± 25.16	18.75 ± 40.20	**41.88** ± 25.83
	12m	9.38 ± 13.98	2.50 ± 5.59	17.50 ± 10.27	6.25 ± 13.98	3.12 ± 6.99	**27.50** ± 21.13
Average		50.12 ± 9.65	43.75 ± 14.42	56.00 ± 11.07	33.25 ± 16.50	44.12 ± 18.89	**62.00** ± 13.80
Marine Single-Im	10m ◇	**100.00** ± 0.00	98.75 ± 2.80	0.62 ± 1.40	88.75 ± 12.22	**100.00** ± 0.00	**100.00** ± 0.00
	8m	98.75 ± 1.71	83.12 ± 29.28	3.75 ± 8.39	90.62 ± 8.84	**100.00** ± 0.00	**100.00** ± 0.00
	5m	3.75 ± 6.77	33.12 ± 33.63	6.25 ± 13.98	87.50 ± 8.84	91.88 ± 6.09	**97.50** ± 2.61
	4m	1.25 ± 2.80	20.62 ± 33.04	0.62 ± 1.40	62.50 ± 21.99	73.75 ± 15.72	**75.62** ± 21.58
	3m	0.00 ± 0.00	6.88 ± 13.69	2.50 ± 4.07	12.50 ± 18.22	**68.75** ± 17.40	64.38 ± 15.40
Average		40.75 ± 1.83	48.50 ± 17.09	2.75 ± 5.19	68.38 ± 11.47	86.88 ± 4.49	**87.50** ± 6.54
Marine Single-Mid	3m	51.88 ± 17.34	45.62 ± 33.70	81.25 ± 16.09	77.50 ± 17.73	83.12 ± 21.94	**91.25** ± 6.40
	5m ◇	**100.00** ± 0.00	99.38 ± 1.40	91.25 ± 2.61	**100.00** ± 0.00	**100.00** ± 0.00	99.38 ± 1.40
	8m	75.62 ± 36.60	45.62 ± 36.68	74.38 ± 32.28	71.25 ± 15.53	40.00 ± 39.12	**97.50** ± 3.42
	10m	46.25 ± 42.35	31.87 ± 42.76	71.25 ± 27.55	27.50 ± 28.42	7.50 ± 13.55	**73.12** ± 23.55
	12m	40.62 ± 42.33	12.50 ± 21.65	57.50 ± 40.42	12.50 ± 20.37	0.62 ± 1.40	**64.38** ± 30.59
Average		62.88 ± 22.12	47.00 ± 22.46	75.12 ± 16.84	57.75 ± 8.54	46.25 ± 11.77	**85.12** ± 9.23
Marine Single-Large	3m	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	OOM*	**1.25** ± 2.80
	10m	45.62 ± 40.30	10.00 ± 20.66	0.00 ± 0.00	16.25 ± 6.01	OOM*	**58.13** ± 23.55
	25m ◇	**100.00** ± 0.00	**100.00** ± 0.00	0.00 ± 0.00	97.50 ± 4.07	OOM*	99.38 ± 1.40
	28m	99.38 ± 1.40	**100.00** ± 0.00	0.00 ± 0.00	98.12 ± 2.80	OOM*	97.50 ± 4.07
	30m	96.88 ± 5.41	**99.38** ± 1.40	0.00 ± 0.00	88.75 ± 13.00	OOM*	95.62 ± 5.23
Average		68.38 ± 6.60	61.88 ± 3.45	0.00 ± 0.00	60.13 ± 3.07	OOM*	**70.38** ± 4.57

◇ denotes the source task.
* OOM indicates that the method runs out of memory on our GPUs.

K LEARNING CURVES

In Figure 8, we plot the learning curves of the average performance across different task sets for Multi-to-Multi. To make the figures clearer, we use abbreviations to represent the dataset quality. Specifically: -E, -M, -MR, and -ME stand for expert, medium, medium-replay, and medium-expert, respectively. We report results over 5 random seeds, where the solid line represents the mean and the shaded area represents one standard deviation. It is evident that TasQ sustains the highest average performance across most task sets.

In Figures 9, 10, and 11, we plot the learning curves for the average performance (marked as "AVG") and task-specific performance (marked with the respective task name) for One-to-Multi task sets. The results are reported over 5 random seeds, with the solid line representing the mean and the shaded area indicating one standard deviation. It can be observed that even when trained on a single source task, TasQ exhibits the best multi-task generalization performance. Notably, in the *Marine-Single* task set, TasQ's peak performance on unseen tasks significantly surpasses that of the other baselines.

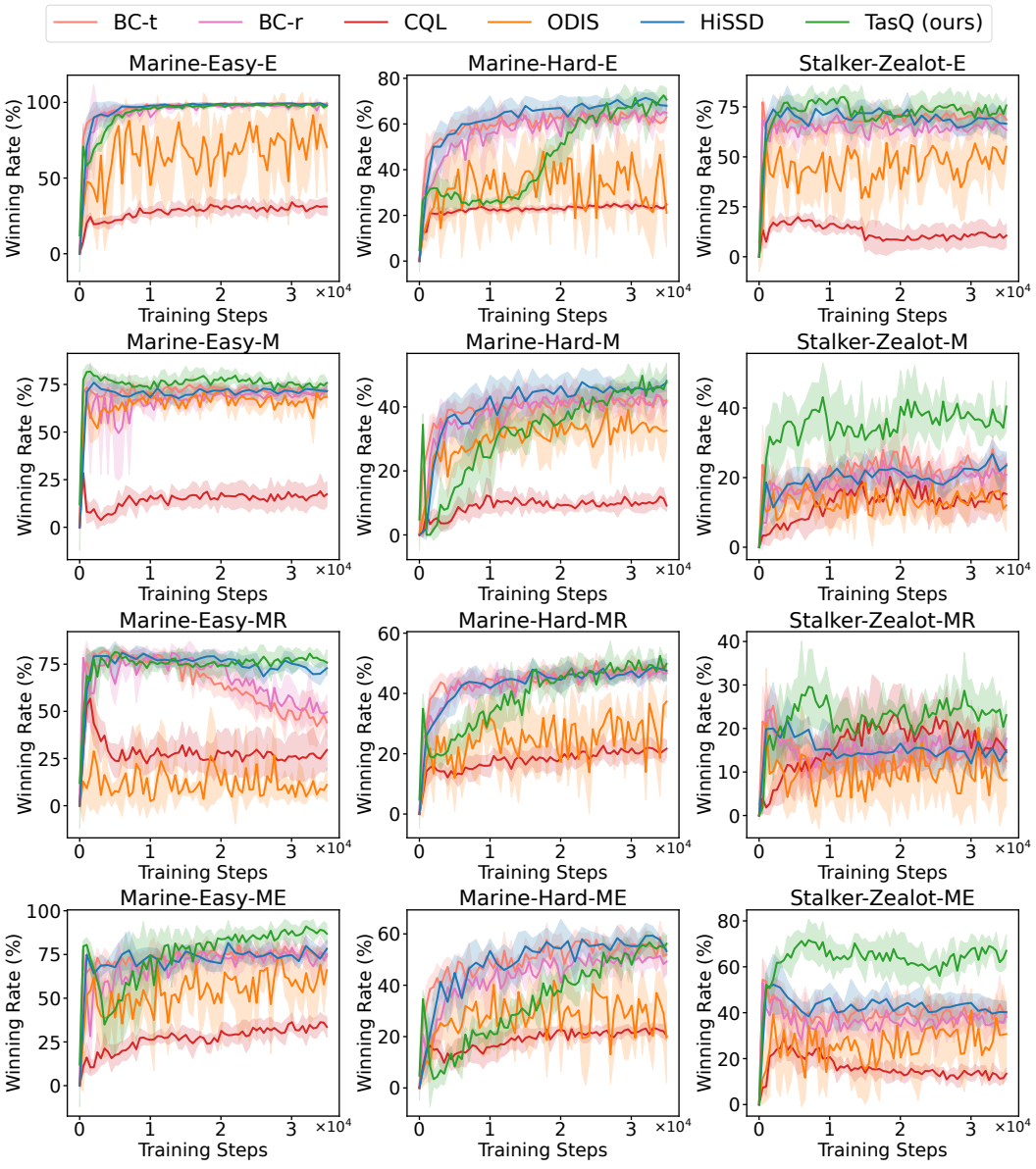


Figure 8: Average winning rates on Multi-to-Multi task sets. We report results over 5 random seeds, where the solid line represents the mean and the shaded area represents one standard deviation.

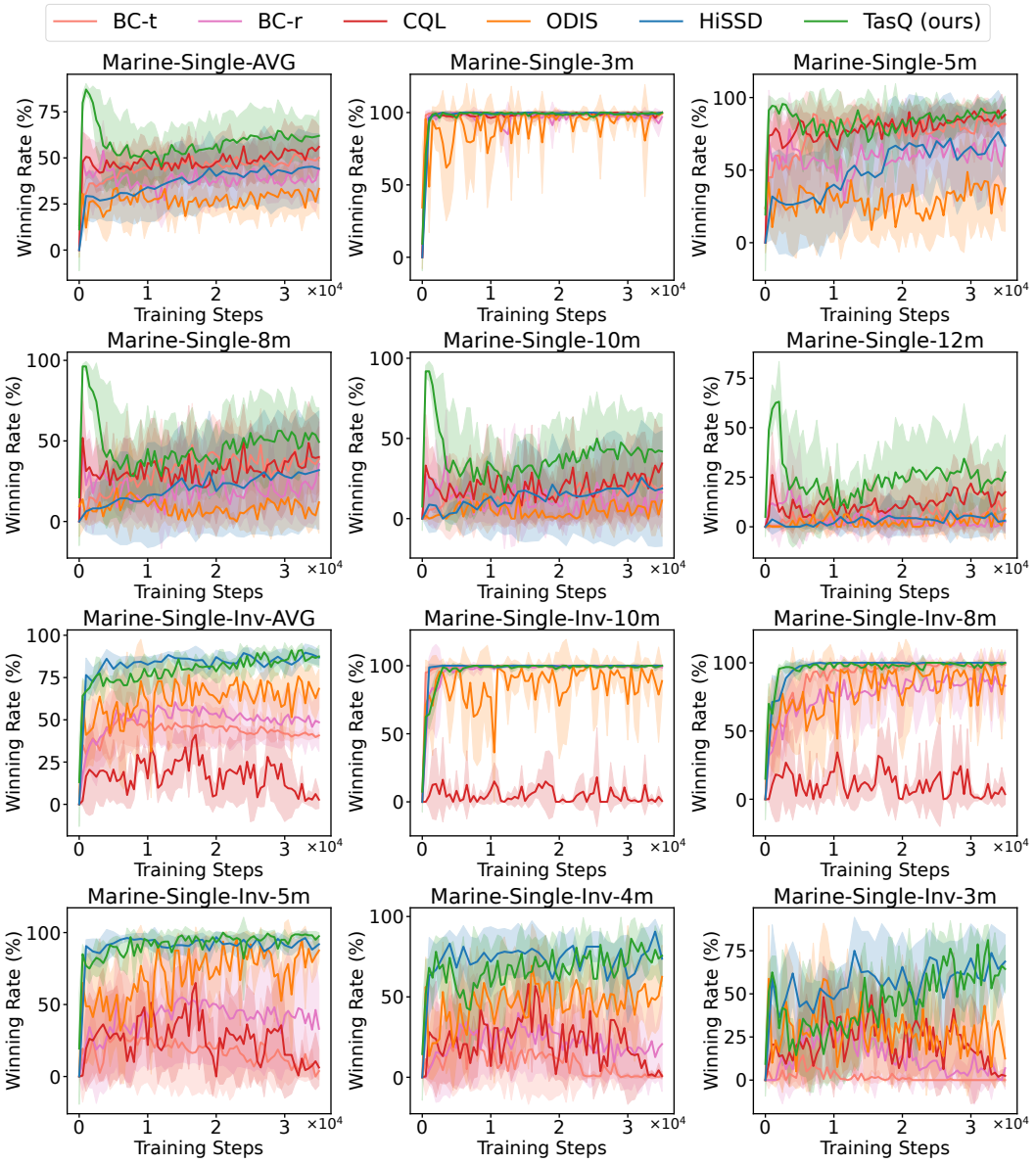


Figure 9: Average and task-specific winning rates on the *Marine-Single* and *Marine-Single-Inv* task sets. We report results over 5 random seeds, where the solid line represents the mean and the shaded area represents one standard deviation.

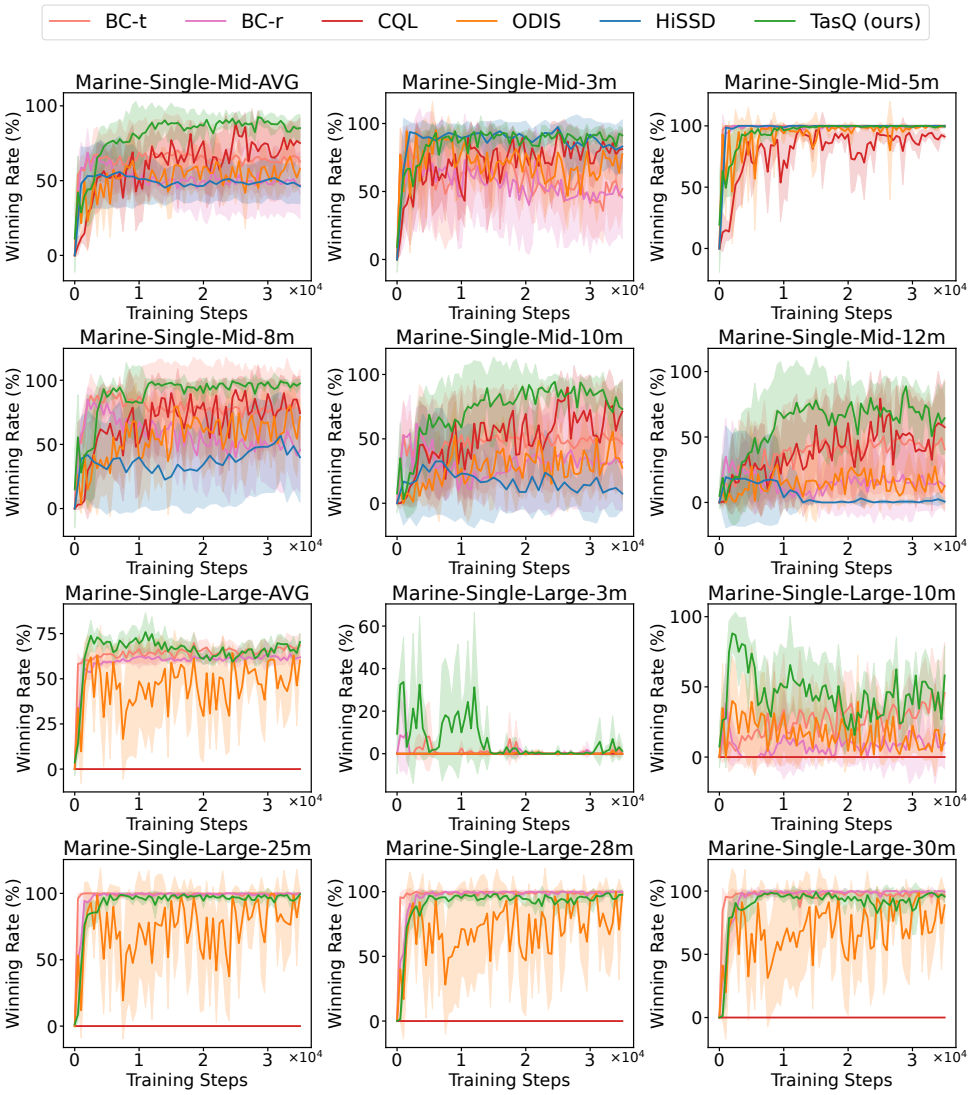


Figure 10: Average and task-specific winning rates on the *Marine-Single-Mid* and *Marine-Single-Large* task sets. HiSSD is absent from the *Marine-Single-Large* panels because it runs out of memory at this scale. We report results over 5 random seeds, where the solid line represents the mean and the shaded area represents one standard deviation.

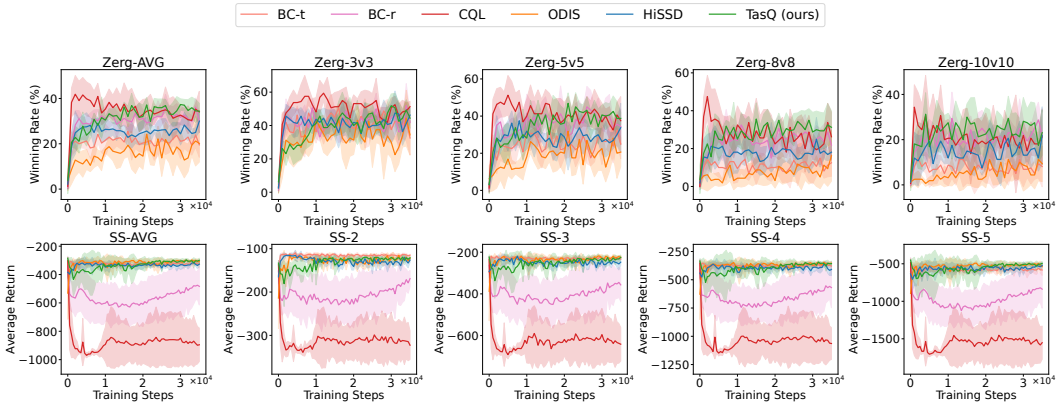


Figure 11: Average and task-specific results on the *SMACv2* (winning rates, top) and *Continuous-MPE* (average returns, bottom) task sets. We report results over 5 random seeds, where the solid line represents the mean and the shaded area represents one standard deviation.