

# Partial Action Replacement: Tackling Distribution Shift in Offline MARL

Yue Jin<sup>1</sup>, Giovanni Montana<sup>1, 2, 3</sup>

<sup>1</sup> Warwick Manufacturing Group, University of Warwick, Coventry, CV4 7AL, UK

<sup>2</sup> Department of Statistics, University of Warwick, Coventry, CV4 7AL, UK

<sup>3</sup> The Alan Turing Institute, 96 Euston Road, London, NW1 2DB, UK  
{yue.jin.3, g.montana}@warwick.ac.uk

## Abstract

Offline multi-agent reinforcement learning (MARL) is severely hampered by the challenge of evaluating out-of-distribution (OOD) joint actions. Our core finding is that when the behavior policy is factorized—a common scenario where agents act fully or partially independently during data collection—a strategy of partial action replacement (PAR) can significantly mitigate this challenge. PAR updates a single or part of agents’ actions while the others remain fixed to the behavioral data, reducing distribution shift compared to full joint-action updates. Based on this insight, we develop Soft-Partial Conservative Q-Learning (SPaCQL), using PAR to mitigate OOD issue and dynamically weighting different PAR strategies based on the uncertainty of value estimation. We provide a rigorous theoretical foundation for this approach, proving that under factorized behavior policies, the induced distribution shift scales linearly with the number of deviating agents rather than exponentially with the joint-action space. This yields a provably tighter value error bound for this important class of offline MARL problems. Our theoretical results also indicate that SPaCQL adaptively addresses distribution shift using uncertainty-informed weights. Our empirical results demonstrate SPaCQL enables more effective policy learning, and manifest its remarkable superiority over baseline algorithms when the offline dataset exhibits the independence structure.

**Extended version** — <https://arxiv.org/pdf/2511.07629>

## Introduction

Multi-agent reinforcement learning (MARL) offers a powerful paradigm for solving complex coordination problems, yet its real-world application is often constrained by the need for extensive online environment interaction [Gronauer and Diepold 2022, Wong et al. 2022, Dinneweth et al. 2022]. Offline MARL overcomes this by enabling agents to learn from fixed, pre-collected datasets. This approach, however, confronts a fundamental obstacle rooted in the combinatorial nature of multi-agent systems: the curse of dimensionality in the joint-action space. Any finite dataset provides only sparse coverage of all possible action combinations. This forces the learning algorithm to reason about the value of countless out-of-distribution (OOD) joint actions. Standard

value-based methods, like Q-learning, are notoriously prone to failure in this regime [Kumar et al. 2020, 2019]. A function approximator, such as a neural network, can produce arbitrarily high Q-values for these unseen OOD actions. These erroneous estimates guide agents toward divergent policies that fail to generalize, making robust offline MARL a significant challenge [Yang et al. 2021, Shao et al. 2023].

This OOD problem can be understood from a geometric perspective. The offline dataset represents a sparse collection of known joint actions within the vast, high-dimensional space of all possible action combinations. A conventional Q-learning update requires evaluating a joint action where all agents adopt their learned policies. This is akin to querying a point far from any known data, forcing the Q-function to perform a large and unreliable extrapolation into uncharted regions of the joint-action space. The core insight of our work is we can avoid such risky extrapolations by instead querying points that remain anchored close to the known data.

We operationalize this insight through a principle we term *partial action replacement*: changing only a single or part of agents’ actions while keeping the others fixed to actions from the dataset. This simple change ensures the Q-value is queried for a joint action that differs minimally from the known data—requiring only a small, local extrapolation rather than a large leap into the unknown. The intuition is straightforward: if we change one coordinate of a known data point, we stay much closer to familiar territory than if we change all coordinates simultaneously.

Crucially, this approach is most effective when the offline dataset was collected by agents acting independently or whose behaviors are loosely coordinated—a common scenario in practice, including independent human demonstrations, independently trained agents, or decentralized systems [Chen et al. 2017, Tampuu et al. 2017, Leibo et al. 2017]. Under this factorized behavior policy assumption, we can prove that while the joint-action space remains exponentially large, our approach of partial replacement achieves linear scaling of distribution shift and value estimation error with respect to the number of deviating agents. This theoretical insight directly motivates our algorithmic design.

Based on this principle, we develop algorithms that navigate the critical balance between estimation stability and inter-agent coordination. We first present Individual CQL with Q-sharing (ICQL-QS), which directly implements par-

tial replacement by changing only one agent’s action at a time to maximize stability. While highly stable, this conservative approach may miss valuable coordinated behaviors. Our primary contribution, Soft Partial Conservative Q-Learning (SPaCQL), addresses this limitation by forming an adaptive algorithm that dynamically weights different coordination combinations based on uncertainty of target values, as measured by the standard deviation of a Q-function ensemble [An et al. 2021, Osband et al. 2016].

We provide a rigorous theoretical foundation for this approach. Our analysis formally proves that under factorized behavior policies, the distribution shift induced by partial action replacement scales linearly with the number of deviating agents, in stark contrast to the exponential complexity of arbitrary joint-action updates. We leverage this result to derive a novel, MARL-specific value-error bound for this important class of problems and then prove that SPaCQL adjusts distribution shift and value-error bound adaptively based on uncertainty-informed weights.

The main contributions of this paper are therefore:

- We introduce and formalize the principle of partial action replacement, showing that under factorized behavior policies, it reduces distribution shift in a multi-agent setting.
- We develop SPaCQL, an algorithm that dynamically weights partial action replacement strategies based on uncertainty estimates, adapting between conservative single-agent updates and coordinated multi-agent updates based on data characteristics.
- We provide theoretical analysis including a novel value-error bound that scales linearly with the number of deviating agents rather than exponentially with joint-action space, applicable when the behavior policy is factorized.
- Empirical results demonstrate SPaCQL’s efficacy, especially its consistent and remarkable superiority on all Random and Medium-Replay datasets where agent behaviors are less coordinated.

## Related Work

Offline reinforcement learning addresses learning effective policies from static datasets without environment interaction. Single-agent methods like CQL [Kumar et al. 2020], IQL [Kostrikov, Nair, and Levine 2022], and BEAR [Kumar et al. 2019] tackle distribution shift by constraining policies or regularizing value functions. The multi-agent setting amplifies these challenges due to exponential joint-action space growth and coordination requirements. Offline MARL research focuses on two main approaches: constraining learned policies or regularizing value functions. Our work builds on the latter with a novel adaptive mechanism.

**Policy-Constrained Methods** Building on single-agent approaches like AWR [Peng et al. 2019] and AWAC [Nair et al. 2020], these methods ensure policies don’t deviate excessively from the behavior policy [Pan et al. 2022, Tseng et al. 2022]. [Pan et al. 2022] use evolution strategies for decentralized regularization, while [Tseng et al. 2022] employs a Teacher-Student paradigm where a centralized transformer

predicts joint actions and individual policies mimic both actions and structural relationships.

**Value-Constrained Methods** These methods constrain the value function, penalizing OOD actions [Yang et al. 2021, Shao et al. 2023, Barde et al. 2024, Ma and Wu 2023, Wang and Zhan 2023, Wang et al. 2023].

CFCQL [Shao et al. 2023] extends CQL [Kumar et al. 2020] to multi-agent settings, penalizing each agent’s OOD action individually while holding others constant—sharing motivation with our partial replacement. However, we use partial replacement for target value computation (not just regularization) and introduce adaptive weighting based on uncertainty.

Others adapt online MARL mechanisms like QMIX [Rashid et al. 2018] and QTRAN [Son et al. 2019] to offline settings. [Wang and Zhan 2023] and [Wang et al. 2023] decompose global Q-functions: the former uses IQL’s expectile regression, the latter formulates convex optimization. We differ by using centralized training with shared Q-functions, capturing coordination implicitly.

[Barde et al. 2024] use world models to generate trajectories with uncertainty-modified rewards. Recent works employ diffusion models: DoF [Li et al. 2025], MADIFF [Zhu et al. 2024], and INS [Fu et al. 2025].

Uncertainty estimation via ensembles [An et al. 2021, Bai et al. 2022, Zhang et al. 2020, Osband et al. 2016, Lakshminarayanan, Pritzel, and Blundell 2017] helps avoid overestimating OOD actions. We use ensemble variance to weight partial replacement strategies—higher uncertainty reduces contribution in Q-updates, adapting to dataset characteristics.

## Preliminaries

We study offline MARL in a Decentralized Markov Decision Process (Dec-MDP)  $\langle \mathcal{S}, \{\mathcal{A}_i\}_{i=1}^n, P, R, n, \gamma \rangle$ , where  $n$  agents interact in state space  $\mathcal{S}$ . At each timestep, agent  $i$  selects action  $a_{t,i} \in \mathcal{A}_i$ , forming joint action  $\mathbf{a}_t \in \mathcal{A} = \times_{i=1}^n \mathcal{A}_i$ . The environment transitions to  $s_{t+1} \sim P(\cdot | s_t, \mathbf{a}_t)$  and yields reward  $r_t = R(s_t, \mathbf{a}_t)$ .

In the offline setting, agents cannot interact with the environment. Learning occurs solely from a static dataset  $\mathcal{D} = \{(s, \mathbf{a}, r, s')_k\}$  collected by behavior policy  $\mu(\mathbf{a}|s)$ . The goal remains learning a joint policy  $\pi(\mathbf{a}|s)$  that maximizes expected return  $J(\pi) = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t r_t]$ , with Q-function defined as:

$$Q^\pi(s, \mathbf{a}) = \mathbb{E}_\pi \left[ \sum_{t=0}^{\infty} \gamma^t R(s_t, \mathbf{a}_t) \mid s_0 = s, \mathbf{a}_0 = \mathbf{a} \right].$$

The core challenge is that distribution shift between  $\pi$  and  $\mu$  causes Q-learning to overestimate OOD joint actions, leading to policy divergence.

## Methodology

Our approach is founded on a simple insight for mitigating the challenge of OOD actions in offline MARL. The core problem is that a Q-function trained on a static dataset can produce arbitrarily wrong values for novel joint actions not seen in the data. A conventional update, which evaluates a

Algorithm	Cooperative Navigation				World			
	Exp	Med	M-R	Rand	Exp	Med	M-R	Rand
CFCQL	<b>112</b>	<b>65</b>	52.2	62.2	<b>119.7</b>	<b>93.8</b>	73.4	68
ICQL-QS	97.2	62.5	<b>58.2</b>	<b>77.7</b>	106.5	88.6	<b>82.5</b>	<b>89.9</b>

Table 1: Performance comparison, CFCQL vs ICQL-QS.

joint action where all agents follow their new policies, is almost certain to be OOD. Our guiding principle is to avoid such large, unreliable extrapolations. Instead, we propose to evaluate a *partially new* joint action that differs from the dataset by only one or a few individual actions. This small, anchored step is inherently more stable and leads to more reliable Q-value estimates.

### ICQL-QS: A Stable but Myopic Baseline

Our first algorithm, Individual Conservative Q-Learning with Q-sharing (ICQL-QS), is the most direct implementation of our core idea. While standard CQL penalizes OOD actions, we recognize that in MARL, a fully new joint action is almost always OOD. ICQL-QS therefore takes a more targeted approach by constructing a Bellman target that only changes one agent’s action at a time.

This is formalized through an individual Bellman operator,  $\mathcal{T}_i^{\text{ind}}$ , for each agent  $i$ . This operator computes the expected future value by sampling agent  $i$ ’s next action,  $a'_i$ , from its learned policy  $\pi_i$ , while sampling the actions of all other agents,  $\mathbf{a}'_{-i}$ , from the dataset  $\mathcal{D}$ :

$$\mathcal{T}_i^{\text{ind}}Q(s, \mathbf{a}) := \mathbb{E}_{\substack{(s', \mathbf{a}'_{-i}) \sim \mathcal{D} \\ a'_i \sim \pi_i(\cdot | s')}} [r + \gamma Q(s', a'_i, \mathbf{a}'_{-i})]. \quad (1)$$

The full algorithm then learns a shared Q-function,  $Q_\theta$ , by minimizing the average of per-agent loss  $\mathcal{L}_i(\theta)$ :

$$\begin{aligned} \mathcal{L}_i(\theta) = & \mathbb{E}_{(s, \mathbf{a}) \sim \mathcal{D}} [(Q_\theta(s, \mathbf{a}) - \mathcal{T}_i^{\text{ind}}Q)^2] \\ & + \lambda (\mathbb{E}_{(s, \mathbf{a}_{-i}) \sim \mathcal{D}, a_i \sim \pi_i} [Q_\theta(s, a_i, \mathbf{a}_{-i})] - \mathbb{E}_{(s, \mathbf{a}) \sim \mathcal{D}} [Q_\theta(s, \mathbf{a})]). \end{aligned} \quad (2)$$

The first term is the squared Bellman error with respect to our individual operator, while the second is the conservative regularizer that penalizes OOD actions for agent  $i$ .

Although each update appears to be “myopic” by only considering a single agent’s deviation, the use of a single, shared Q-function means that every update provides a learning signal that implicitly couples the agents and fosters coordination. We formalize this connection to a centralized learning objective in our theoretical analysis.

### SPaCQL: An Adaptive Mixture of Partial Backups

While ICQL-QS achieves stability through single-agent updates, it cannot capture the value of coordinated multi-agent improvements. Yet fully joint updates risk severe extrapolation errors. The key insight motivating SPaCQL is that the optimal number of deviating agents depends on the data: random datasets favor conservative single-agent updates, while expert datasets may not, but may support coordinated deviations. Rather than committing to either extreme, SPaCQL adaptively interpolates between them. Our

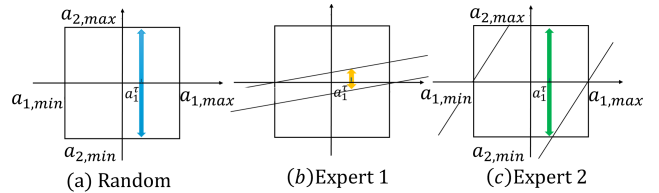


Figure 1: Action distributions in different dataset types. Blue, yellow and green lines show the range of agent actions, with  $a_1^i$  denoting agent 1’s action from the dataset. (a) **Random dataset**: Agents act independently. Agent 2’s actions may span the full range regardless of agent 1’s choice. Partial action replacement (fixing agent 1) allows reasonable coverage of agent 2’s action space. (b) **Expert dataset with tight correlation**: Agents’ actions are strongly correlated within a narrow band. Fixing agent 1 may still lead to severe OOD. (c) **Expert dataset with loose correlation**: While correlated, actions have sufficient spread. Fixing agent 1 still allows reasonable coverage of agent 2’s action space. This illustrates the potential advantage of partial action replacement for random datasets and some expert datasets, but it may struggle with highly coordinated behaviors.

initial experiments confirmed this intuition, revealing a stark performance trade-off dependent on dataset quality.

The performance comparison between CFCQL [Shao et al. 2023] and ICQL-QS showed in Table 1 indicates that on datasets composed of noisy, “random,” or exploratory behavior, ICQL-QS demonstrates superior performance. By staying close to the data manifold, they minimize distribution shift and avoid the OOD extrapolation errors that plague joint-action methods in sparse data regimes. Figure 1 illustrates the potential advantage of partial action replacement in terms of mitigating OOD especially for random datasets.

Conversely, on “expert” datasets containing highly coordinated trajectories, CFCQL using a full joint-action update can outperform ICQL-QS. On the one hand, the advantage of ICQL-QS in mitigating OOD issue may not be prominent on expert dataset, as illustrated in Figure 1 (b). On the other hand, evaluating fully novel joint actions is essential for learning the value of synergistic maneuvers, and a factorized approach is “coordination-blind” in this setting and can become overly pessimistic. However, as illustrated in Figure 1(c), on some expert datasets, partial action replacement may still have a considerable advantage in mitigating OOD.

Our experiments shown in Figure 2 confirm this advantage of mitigating OOD issue by measuring the uncertainty of Q-estimates. We empirically compare the estimation uncertainty of different target-value strategies, measured by the standard deviation across a Q-function ensemble. As shown, a standard joint-action update (CFCQL) exhibits higher uncertainty than ICQL-QS that uses partial action replacement.

This presents a clear challenge: no single, fixed backup strategy is optimal across all data regimes. This motivates our primary objective: to design a unified algorithm that can gracefully and automatically interpolate between these two extremes, letting the data itself decide how many coordinated deviations are safe to consider at any given moment.

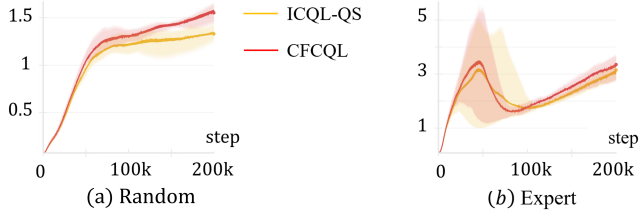


Figure 2: Comparison of the uncertainty of Q-value estimation (CFCQL vs ICQL-QS) on Cooperative Navigation benchmark. Uncertainty is measured as the standard deviation of Q-estimates from an ensemble of networks. ICQL-QS, which uses partial action replacement yields significantly lower estimation uncertainty on Random dataset. On Expert dataset, the two methods show similar estimation uncertainty, but finally ICQL-QS leads to lower uncertainty.

### The Soft-Partial Bellman Operator

Instead of committing to a fixed strategy, we introduce SPaCQL, which constructs its Bellman target as a mixture of  $n$  base operators,  $\{\mathcal{T}^{(k)}\}_{k=1}^n$ , where each operator corresponds to replacing exactly  $k$  agents' actions.

For any integer  $k \in \{1, \dots, n\}$ , we define a Bellman operator  $\mathcal{T}^{(k)}$  that bootstraps from a next-state action where exactly  $k$  agents (chosen uniformly at random) deviate from the behavior policy, while the other  $n - k$  agents' actions are taken from the logged data.

$$\mathcal{T}^{(k)}Q(s, \mathbf{a}) := \mathbb{E}_{s' \sim \mathcal{D}, \mathbf{a}'^{(k)}} \left[ r + \gamma Q(s', \mathbf{a}'^{(k)}) \right]. \quad (3)$$

Each of these operators is a  $\gamma$ -contraction in the  $\ell_\infty$  norm.

The final SPaCQL operator,  $\mathcal{T}^{\text{SP}}$ , is a convex combination of these base operators, with mixture weights  $w_k$ :

$$\mathcal{T}^{\text{SP}}Q(s, \mathbf{a}) := \sum_{k=1}^n w_k \mathcal{T}^{(k)}Q(s, \mathbf{a}). \quad (4)$$

As a convex combination of  $\gamma$ -contractions,  $\mathcal{T}^{\text{SP}}$  is itself guaranteed to be a  $\gamma$ -contraction.

### The SPaCQL Algorithm and Learning Objective

The learning objective for SPaCQL,  $\mathcal{L}(\theta)$ , which combines the TD error for our adaptive Bellman operator, is:

$$\mathcal{L}(\theta) = \mathbb{E}_{\mathcal{D}} \left[ (Q_\theta(s, \mathbf{a}) - Y_{\text{SP}})^2 \right] + \xi_c, \quad (5)$$

where  $\xi_c = \alpha \sum_{i=1}^n \lambda_i (\mathbb{E}_{(s, \mathbf{a}_{-i}) \sim \mathcal{D}, a_i \sim \pi_i} [Q_\theta(s, a_i, \mathbf{a}_{-i})] - \mathbb{E}_{(s, \mathbf{a}) \sim \mathcal{D}} [Q_\theta(s, \mathbf{a})])$  is a conservative penalty same as that used in CFCQL.

The target  $Y_{\text{SP}}$  is constructed using the soft-partial operator, with an ensemble for conservatism:

$$Y_{\text{SP}} := r + \gamma \sum_{k=1}^n w_k \min_j Q_j^{\text{tar}}(s', \mathbf{a}'^{(k)}). \quad (6)$$

The weights are determined by uncertainty: high ensemble disagreement signals poor data coverage, so we down-weight the corresponding risky deviations. We measure uncertainty via Q-ensemble variance:

$$u_k = \sqrt{\text{Var}_j [Q_{\theta_j}(s', \mathbf{a}'^{(k)})]}. \quad (7)$$

### Algorithm 1: SPaCQL

---

```

1: Initialize: Q-ensemble  $\{Q_{\theta_j}\}$ , target ensemble  $\{\bar{Q}_{\bar{\theta}_j}\}$ , policies  $\{\pi_{\psi_i}\}_{i=1}^n$ , target policies  $\{\bar{\pi}_{\bar{\psi}_i}\}_{i=1}^n$ , replay buffer  $\mathcal{D}$ 
2: for each iteration do
3:   Sample batch  $\mathcal{B} = \{(s, \mathbf{a}, r, s', \mathbf{a}')\}$  from  $\mathcal{D}$ 
4:   Set  $\mathcal{L}(\theta) \leftarrow 0$ 
5:   for each transition  $(s, \mathbf{a}, r, s', \mathbf{a}')$  in  $\mathcal{B}$  do
6:     for  $k = 1$  to  $n$  do
7:       Sample  $k$  agent indices  $\{\sigma_\rho\}_{\rho=1}^k$ 
8:       Sample  $\{a_{\sigma_\rho}^\pi \sim \pi_{\sigma_\rho}(\cdot | s')\}_{\rho=1}^k$ 
9:       Construct  $\mathbf{a}'^{(k)} \leftarrow \mathbf{a}'$ ;  $\forall \sigma_\rho \in \{\sigma_\rho\}_{\rho=1}^k$ , replace the  $\sigma_\rho$ -th component of  $\mathbf{a}'^{(k)}$  with  $a_{\sigma_\rho}^\pi$ 
10:       $u_k \leftarrow \sqrt{\text{Var}_j [Q_{\theta_j}(s', \mathbf{a}'^{(k)})]}$ 
11:      Set  $y_k = \frac{1}{u_k} \min_j \bar{Q}_{\theta_j}(s', \mathbf{a}'^{(k)})$ 
12:    end for
13:     $Y_{\text{SP}} = r + \gamma \sum_k y_k / \sum_k \frac{1}{u_k}$ 
14:     $\mathcal{L}(\theta) += \sum_j (Q_{\theta_j}(s, \mathbf{a}) - Y_{\text{SP}})^2 + \xi_c$ 
15:  end for
16:   $\theta \leftarrow \theta - \eta_\theta \nabla_\theta \mathcal{L}(\theta)$ 
17:   $\bar{\theta} \leftarrow (1 - \tau)\bar{\theta} + \tau\theta$ 
18:  — Agent policy update —
19:  for each agent  $i$  do
20:     $\psi_i \leftarrow \psi_i + \eta_\pi \nabla_{\pi_i} \mathbb{E}_{s, \mathbf{a}_{-i} \sim \mathcal{D}, a_i \sim \pi_i} Q_{\theta_1}(s, \mathbf{a})$ 
21:     $\bar{\psi}_i \leftarrow (1 - \tau)\bar{\psi}_i + \tau\psi_i$ 
22:  end for
23: end for

```

---

We then map this uncertainty signal to the weight vector using the normalized standard deviation of the Q-values at the next state:  $w_k = \frac{1/u_k}{\sum_k 1/u_k}$ , such that higher variance pushes weight to more conservative, smaller- $k$  backups. A complete algorithm is shown in Algorithm 1.

### Theoretical Analysis

We provide a theoretical foundation explaining why *partial action replacement* alleviates distribution shift in offline MARL. Our key insight is that while the joint-action space grows exponentially in the number of agents, the distribution shift induced by partial replacement grows only linearly. All proofs are in Appendix.

### Setting and Key Assumptions

We consider a finite Dec-MDP  $\langle \mathcal{S}, \{\mathcal{A}_i\}_{i=1}^n, P, R, \gamma \rangle$  and a fixed dataset  $\mathcal{D} = \{(s, \mathbf{a}, r, s')\}$  of *i.i.d.* transitions. Rewards are bounded:  $|R(s, \mathbf{a})| \leq 1$ .

**Assumption 1** (Factorized behavior policy). *The behavior policy is factorized:*  $\mu(s, \mathbf{a}) = \prod_{i=1}^n \mu_i(a_i | s)$ .

The finite state-action space assumption is standard in theoretical MARL analysis and allows us to use discrete probability measures and avoid measure-theoretic complications. The *i.i.d.* assumption on dataset transitions simplifies the analysis by avoiding temporal correlations, though our practical algorithm works with standard trajectory data. Assumption 1 is crucial—it means the data collection process involved independent agent policies, which is realistic for

many multi-agent scenarios where agents were trained separately or act without explicit coordination. This assumption enables us to decompose distribution shifts across individual agents, which is fundamental to our analysis. The bounded rewards assumption is standard and can always be achieved through reward clipping; it ensures value functions remain bounded and Lipschitz continuous.

**Metric and total variation.** On the finite state-action space we use the 0-1 metric. We define total variation distance as  $\text{TV}(\nu, \nu') := \frac{1}{2} \|\nu - \nu'\|_1$ . Under the 0-1 metric, the *Wasserstein-1 distance equals total variation*:

$$W_1(\nu, \nu') = \text{TV}(\nu, \nu') = \frac{1}{2} \|\nu - \nu'\|_1 \quad (8)$$

The choice of 0-1 metric simplifies the analysis significantly—in discrete spaces, it makes Wasserstein distance equivalent to total variation distance, eliminating the need for optimal transport calculations while still capturing meaningful notions of distribution similarity.

**Lipschitz constant of  $Q$ .** With bounded rewards, the optimal value function obeys  $\|Q^*\|_\infty \leq 1/(1-\gamma)$  and is  $2/(1-\gamma)$ -Lipschitz under the 0-1 metric. To see this, note that for any two state-action pairs  $(s, \mathbf{a})$  and  $(s', \mathbf{a}')$ :

$$|Q^*(s, \mathbf{a}) - Q^*(s', \mathbf{a}')| \leq \sum_{t=0}^{\infty} \gamma^t \cdot 2 = \frac{2}{1-\gamma} \quad (9)$$

since rewards are bounded by 1 and the 0-1 metric ensures the maximum difference is 2.

This Lipschitz property is automatic given bounded rewards and provides the key tool for translating distribution shifts into value function estimation errors. The specific constant  $2/(1-\gamma)$  is tight and captures how value function smoothness degrades as the discount factor approaches 1.

**Mixed policies.** For any subset  $S \subseteq \{1, \dots, n\}$ , define:

$$\pi^{(S)} := \left( \prod_{i \in S} \pi_i \right) \left( \prod_{j \notin S} \mu_j \right) \quad (10)$$

and let  $d^{(S)}$  denote its discounted occupancy measure.

These mixed policies are the key technical tool for our analysis.  $\pi^{(S)}$  represents a policy where agents in set  $S$  follow their learned policies  $\pi_i$  while agents outside  $S$  follow the behavior policies  $\mu_j$ . This construction allows us to decompose the transition from behavior policy  $\mu = \pi^{(\emptyset)}$  to learned policy  $\pi = \pi^{\{1, \dots, n\}}$  through a sequence of intermediate policies, enabling us to track how distribution shift accumulates as we add more deviating agents.

## Distribution-Shift Control via Partial Replacement

**Total-variation convention.** For two stochastic kernels  $\kappa, \kappa'$  on  $\mathcal{S}$  we write

$$\text{TV}(\kappa, \kappa') := \sup_{s \in \mathcal{S}} \frac{1}{2} \|\kappa(s, \cdot) - \kappa'(s, \cdot)\|_1.$$

Under the 0-1 ground metric,  $W_1(\kappa, \kappa') = \text{TV}(\kappa, \kappa')$ .

**Lemma 1 (Linear Divergence Bound).** For every  $S \subseteq \{1, \dots, n\}$ :

$$W_1(d^{(S)}, d^{(\emptyset)}) \leq \frac{\gamma}{1-\gamma} \sum_{i \in S} \text{TV}(\pi_i, \mu_i) \quad (11)$$

The proof shows that despite the exponential joint-action space, the occupancy shift grows *additively* with the number of deviating agents.

This result formalizes our core intuition: while there are  $|\mathcal{A}|^n$  possible joint actions, the distribution shift from changing  $k$  agents' policies is only  $k$  times larger than changing a single agent's policy. This linear scaling is the theoretical foundation for why partial action replacement is fundamentally more stable than full joint-action updates.

## Value-Estimation Error

**Theorem 1 (Tight Value-Error Bound).** Let  $\hat{Q}$  be the  $Q$ -function learned offline and  $\hat{V}^\pi = \mathbb{E}_{d^\pi, \pi} \hat{Q}$ . We assume that the learning process ensures  $\hat{Q}$  is  $\frac{2}{1-\gamma}$ -Lipschitz under the 0-1 metric. While this property is not automatically guaranteed for neural network approximators, it can be encouraged with techniques like spectral normalization or gradient clipping and serves as a key theoretical assumption. Then:

$$|V^\pi - \hat{V}^\pi| \leq \varepsilon_{\text{Subopt}} + \varepsilon_{\text{FQI}} + \frac{4\gamma}{(1-\gamma)^2} \sum_{i=1}^n \text{TV}(\pi_i, \mu_i) \quad (12)$$

where  $\varepsilon_{\text{Subopt}} = \|Q^\pi - Q^*\|_\infty$  and  $\varepsilon_{\text{FQI}} = \|Q^* - \hat{Q}\|_\infty$ .

The proof is provided in Appendix.

**Remark.** If only a single agent  $k$  deviates from  $\mu$ , the last term reduces to  $\frac{4\gamma}{(1-\gamma)^2} \text{TV}(\pi_k, \mu_k)$ , strictly improving on the usual joint-TV bound.

## Generalization to Correlated Behavior Policies

The guarantee in Theorem 1 relies on the assumption that the behavior policy  $\mu$  is fully factorized. We now relax this assumption and show that the linear scaling of distribution shift is robust, degrading gracefully with the degree of correlation in the dataset.

First, we define a measure of the maximal correlation present in behavior policy. Let  $\mu^\otimes(\mathbf{a} | s) := \prod_{i=1}^n \mu_i(a_i | s)$  be the product of the marginal behavior policies.

**Definition 1 (Maximal Excess Correlation).** The maximal excess correlation,  $\kappa$ , is the largest Total Variation distance between the true joint behavior policy and its factorized approximation, over all states:

$$\kappa := \sup_{s \in \mathcal{S}} \frac{1}{2} \|\mu(\cdot | s) - \mu^\otimes(\cdot | s)\|_1$$

where  $\kappa \in [0, 1]$ . If  $\mu$  is perfectly factorized, then  $\kappa = 0$ .

With this, we can state a more general version of our distribution-shift bound.

**Lemma 2** (Linear Divergence with Correlations). *For any subset of agents  $S \subseteq \{1, \dots, n\}$ , the distribution shift is bounded by:*

$$W_1(d^{(S)}, d^\mu) \leq \frac{\gamma}{1-\gamma} \left( \sum_{i \in S} \text{TV}(\pi_i, \mu_i) + \kappa \right) \quad (13)$$

This generalized lemma allows us to bound the value-estimation error without the factorization assumption.

**Theorem 2** (Value-Error Bound with Correlations). *Under the same conditions as Theorem 1, but without the assumption of a factorized behavior policy, the value estimation error is bounded by:*

$$|V^\pi - \hat{V}^\pi| \leq \varepsilon_{\text{Subopt}} + \varepsilon_{\text{FQI}} + \frac{4\gamma}{(1-\gamma)^2} \left( \sum_{i=1}^n \text{TV}(\pi_i, \mu_i) + \kappa \right) \quad (14)$$

*Proof.* The proof is identical to that of Theorem 1, substituting the bound from Lemma 2.  $\square$

This result shows that correlations in the behavior policy introduce a simple additive penalty  $\kappa$  to the error bound. Crucially, this penalty is independent of the number of agents,  $n$ . This means the ‘‘curse of dimensionality’’ is still avoided; the bound remains linear in  $n$  and is simply shifted upwards by a constant reflecting the data’s inherent correlation. The theory therefore remains informative even when the factorization assumption is violated.

### Theoretical Guarantee of SPaCQL

SPaCQL retains the strong theoretical guarantees of our framework without requiring any new structural assumptions on the Q-function. The expected distribution shift under the SPaCQL operator is linear in the *effective* number of deviating agents,  $k_{\text{eff}} = \sum_k w_k \cdot k$ . This leads directly to our final value-error bound.

**Theorem 3** (SPaCQL Value-Error Bound). *Let  $\hat{Q}$  be the function learned by SPaCQL and define the average single-agent policy deviation as  $\overline{\text{TV}}_{\pi, \mu} = \frac{1}{n} \sum_{i=1}^n \text{TV}(\pi_i, \mu_i)$ . The value estimation error is bounded by:*

$$|V^\pi - \hat{V}^\pi| \leq \varepsilon_{\text{Subopt}} + \varepsilon_{\text{FQI}} + \frac{4\gamma}{(1-\gamma)^2} \mathbb{E}_{s' \sim d^\pi} [k_{\text{eff}}(s')] \overline{\text{TV}}_{\pi, \mu}.$$

*Proof Sketch.* The proof follows the same structure as that of Theorem 1. The key difference is that the distribution shift term  $W_1(d^\pi, d^\mu)$  is replaced by the expected shift under the SPaCQL operator. This expectation is taken over the mixture of policies  $\sum_k w_k(s') \pi^{(k)}$ , leading to a dependency on the state-dependent effective number of deviations,  $k_{\text{eff}}(s')$ .  $\square$

The error scales with the effective number of deviations,  $k_{\text{eff}}$ , which the algorithm adapts on a state-by-state basis. When the weights concentrate on  $k = 1$ , we recover the tight ICQL-QS bound; when they shift towards  $k = n$ , we approach the looser, full-joint bound, thus formalizing the algorithm’s adaptive nature.

### Gradient Interpretation of ICQL-QS

While Lemma 1 and Theorem 1 establish the stability benefits of partial action replacement, a natural question arises: can partial replacement achieve effective coordination? One might worry that updating Q-function separately with loss (2), even with a shared Q-function, may lead to miscoordinated policies that fail to capture beneficial joint behaviors.

The following result addresses this concern by revealing the implicit coordination mechanism in ICQL-QS:

**Proposition 1** (Gradient equivalence of ICQL-QS). *Let  $\mathcal{T}_i^{\text{ind}}$  be the individual Bellman operator and define the averaged-individual operator:*

$$\mathcal{T}^{\text{ai}} := \frac{1}{n} \sum_{i=1}^n \mathcal{T}_i^{\text{ind}} \quad (15)$$

*Then the parameter update for the TD component in ICQL-QS is equivalent to stochastic gradient descent on centralized TD loss, under the semi-gradient assumption where the gradient is not backpropagated through the target network (i.e., treating  $\mathcal{T}^{\text{ai}} Q_\theta$  as a fixed target with respect to  $\theta$ ):*

$$\frac{1}{2} \mathbb{E}_{\mathcal{D}} \left[ (Q_\theta - \mathcal{T}^{\text{ai}} Q_\theta)^2 \right] \quad (16)$$

This result shows that ICQL-QS, despite appearing to perform individual agent updates, is mathematically equivalent to centralized training on an averaged objective. The shared Q-function acts as an implicit coordination mechanism: each agent’s update influences the value estimates for all possible joint actions, enabling coordination without explicit joint reasoning. This provides theoretical justification for why partial action replacement can maintain coordination benefits while achieving the stability advantages established in our previous results.

## Experimental Settings and Results

**Experimental setup** We conduct a comprehensive evaluation of SPaCQL on two widely used offline MARL benchmarks: Multi-Agent Particle Environments (MPE) and Multi-Agent MuJoCo (MaMujoco). We use the same dataset as in recent works [Shao et al. 2023, Pan et al. 2022, Kostrikov, Nair, and Levine 2022]. For MPE, there are three distinct tasks: Cooperative Navigation (CN), Predator-Prey (PP), and World. For MaMujoco, there is Half-Cheetah (Half-C). To assess performance under varying data quality, each task utilizes four dataset types: Expert (Exp), Medium (Med), Medium-Replay (Med-R), and Random (Rand).

Our implementation uses an ensemble of 10 Q-networks to estimate value functions and their uncertainty. To ensure robust results, we run all experiments with 5 different random seeds and report the mean and standard deviation of the normalized scores. All hyperparameters for our method and the CFCQL baseline are identical to those specified in the original CFCQL paper. All experiments are implemented with pytorch and run on NVIDIA Tesla V100 GPUs.

**Baselines** The performance of SPaCQL is compared against a suite of state-of-the-art offline MARL algorithms. These baselines are chosen to represent the primary avenues of research in the field:

	Dataset	OMAR	MACQL	IQL	MA-TD3+BC	DoF	CFCQL	SPaCQL	
MPE	CN	Exp	114.9 ± 2.6	12.2±31	103.7±2.5	108.3±3.3	<b>136.4±3.9</b>	112±4	111.9±4.5
		Med	47.9±18.9	14.3±20.2	28.2±3.9	29.3±4.8	75.6±8.7	65.0±10.2	<b>78.6±6.4</b>
		Med-R	37.9±12.3	25.5±5.9	10.8±4.5	15.4±5.6	57.4±6.8	52.2±9.6	<b>71.9±13.2</b>
		Rand	34.4±5.3	45.6±8.7	5.5±1.1	9.8±4.9	35.9±6.8	62.2±8.1	<b>78.2±14</b>
	PP	Exp	116.2±19.8	108.4±21.5	109.3±10.1	115.2±12.5	<b>125.6±8.6</b>	118.2±13.1	111.2±16.4
		Med	66.7±23.2	55±43.2	53.6±19.9	65.1±29.5	<b>86.3±10.6</b>	68.5±21.8	61.9±20
		Med-R	47.1±15.3	11.9±9.2	23.2±12	28.7±20.9	65.4±12.5	71.1±6	<b>75.0±12.7</b>
		Rand	11.1±2.8	25.2±11.5	1.3±1.6	5.7±3.5	16.5±6.3	78.5±15.6	<b>89.4±13.7</b>
	World	Exp	110.4±25.7	99.7±31	107.8±17.7	110.3±21.3	<b>135.2±19.1</b>	119.7±26.4	112.3±7.8
		Med	74.6±11.5	67.4±48.4	70.5±15.3	73.4±9.3	85.2±11.2	93.8±31.8	<b>98.1±17.7</b>
		Med-R	42.9±19.5	13.2±16.2	41.5±9.5	17.4±8.1	58.6±10.4	73.4±23.2	<b>105.2±11.1</b>
		Rand	5.9±5.2	11.7±11	2.9±4.0	2.8±5.5	13.1±2.1	68±20.8	<b>94.3±7.4</b>
MaMujoco	Half-C	Exp	113.5±4.3	50.1±20.1	115.6±4.2	114.4±3.8	-	<b>118.5±4.9</b>	110.5±5.9
		Med	80.4±10.2	51.5±26.7	<b>81.3±3.7</b>	75.5±3.7	-	80.5±9.6	70.3±7.8
		Med-R	57.7±5.1	37.0±7.1	58.8±6.8	27.1±5.5	-	59.5±8.2	<b>66.1±3.4</b>
		Rand	13.5±7.0	5.3±0.5	7.4±0.0	7.4±0.0	-	39.7±4.0	<b>43.8±4.9</b>

Table 2: The average normalized score on offline MARL tasks. The best performance is highlighted in bold.

- Policy-constrained methods: OMAR [Pan et al. 2022], IQL [Kostrikov, Nair, and Levine 2022], and MA-TD3+BC [Fujimoto and Gu 2021, Pan et al. 2022].
- Value-constrained methods: MACQL [Shao et al. 2023] and CFCQL [Shao et al. 2023].
- Diffusion model-based methods: DoF [Li et al. 2025], a recent approach using diffusion models.

**Main quantitative results** The average normalized scores across all tasks and datasets are presented in Table 2, where all baseline algorithm scores are reported as in the corresponding papers. Overall, SPaCQL outperforms all baseline algorithms on 10 of the 16 tasks. The most significant performance gains are observed on datasets with low-quality or uncoordinated data. SPaCQL consistently demonstrates remarkable superiority over all baselines on every “Random” and “Medium-Replay” dataset across all four tasks (CN, PP, World, and Half-C). For instance, on the World task with the “Random” dataset, SPaCQL achieves a score of  $94.3 \pm 7.4$ , whereas the next-best baseline, CFCQL, scores only  $68 \pm 20.8$ .

On high-quality “Expert” datasets, the performance is comparable across several algorithms. Here, methods like DoF, which is the top performer on all three MPE Expert datasets, show strong results. This highlights that when the dataset already contains highly coordinated trajectories, keeping close to the underlying policies is enough.

**Analysis of adaptive weights** To verify that SPaCQL’s adaptive mechanism functions as intended, we visualize the learned weights in Figure 3. The results show the algorithm dynamically adjusts its strategy based on dataset quality. On unstructured datasets like “Random” and “Medium-Replay”, the weight for single-agent deviations ( $w_1$ ) is dominant, prioritizing stability by staying close to the data manifold. Conversely, on “Expert” datasets, the weights for coordinated deviations ( $w_2$  and  $w_3$ ) increase, showing that SPaCQL shifts its focus to find better coordinated policies when the data suggests coordination is reliable.

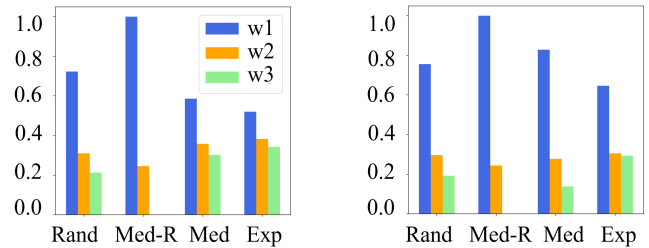


Figure 3: Values of weights with min-max normalization.

## Discussion and Conclusions

Our theoretical analysis reveals a key insight: distribution shift in offline MARL scales linearly with the number of deviating agents, not exponentially with the joint-action space. This striking result suggests the “curse of dimensionality” in this setting may be overstated, as the distribution shift penalty accumulates additively across agents.

Our algorithm, SPaCQL, is built on this principle. It explicitly manages the stability-coordination trade-off by using partial action replacement, dynamically weighting different coordination combinations based on uncertainty to achieve a provably tighter value-error bound.

This work suggests a broader principle for offline MARL: algorithms should minimize simultaneous deviations from the behavior policy. While complete theoretical safety guarantees are pending analysis, the empirical motivation for SPaCQL is strong, as it adapts to the dataset by dynamically selecting the most reliable update strategy. Important future directions therefore include developing more sophisticated weighting schemes and uncertainty estimation techniques.

By formalizing partial action replacement and demonstrating its theoretical advantages, we provide a more optimistic perspective on the challenges of offline MARL. The linear scaling of distribution shift suggests that principled, localized approaches can overcome seemingly intractable combinatorial barriers in multi-agent learning.

## Acknowledgments

Giovanni Montana acknowledges support from a UKRI AI Turing Acceleration Fellowship (EPSRC EP/V024868/1).

## References

- An, G.; Moon, S.; Kim, J.-h.; and Song, H. O. 2021. Uncertainty-Based Offline Reinforcement Learning with Diversified Q-Ensemble. In *35th Conference on Neural Information Processing Systems (NeurIPS 2021)*.
- Bai, C.; Wang, L.; Yang, Z.; Deng, Z.; Garg, A.; Liu, P.; and Wang, Z. 2022. Pessimistic Bootstrapping for Uncertainty-Driven Offline Reinforcement Learning. In *International Conference on Learning Representations*.
- Barde, P.; Foerster, J.; Nowrouzezahrai, D.; and Zhang, A. 2024. A Model-Based Solution to the Offline Multi-Agent Reinforcement Learning Coordination Problem. *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems*, 141–150.
- Chen, Y. F.; Liu, M.; Everett, M.; and How, J. P. 2017. Decentralized non-communicating multiagent collision avoidance with deep reinforcement learning. *Proceedings - IEEE International Conference on Robotics and Automation*, 285–292.
- Dinneweth, J.; Boubezoul, A.; Mandiau, R.; and Espié, S. 2022. Multi-agent reinforcement learning for autonomous vehicles: a survey. *Autonomous Intelligent Systems*, 2(1).
- Fu, Y.; Zhu, Y.; Zhao, J.; Chai, J.; and Zhao, D. 2025. INS: Interaction-aware Synthesis to Enhance Offline Multi-agent Reinforcement Learning. In *The Thirteenth International Conference on Learning Representations*.
- Fujimoto, S.; and Gu, S. S. 2021. A Minimalist Approach to Offline Reinforcement Learning. In *35th Conference on Neural Information Processing Systems (NeurIPS 2021)*.
- Gronauer, S.; and Diepold, K. 2022. Multi - agent deep reinforcement learning : a survey. *Artificial Intelligence Review*, 55(2): 895–943.
- Kostrikov, I.; Nair, A.; and Levine, S. 2022. Offline Reinforcement Learning with Implicit Q-Learning. In *International Conference on Learning Representations*.
- Kumar, A.; Fu, J.; Tucker, G.; and Levine, S. 2019. Stabilizing off-policy Q-learning via bootstrapping error reduction. In *33rd Conference on Neural Information Processing Systems (NeurIPS 2019)*.
- Kumar, A.; Zhou, A.; Tucker, G.; and Levine, S. 2020. Conservative Q-Learning for Offline Reinforcement Learning. In *34th Conference on Neural Information Processing Systems (NeurIPS 2020)*.
- Lakshminarayanan, B.; Pritzel, A.; and Blundell, C. 2017. Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles. *31st Conference on Neural Information Processing Systems*.
- Leibo, J. Z.; Zambaldi, V.; Lanctot, M.; Marecki, J.; and Graepel, T. 2017. Multi-agent Reinforcement Learning in Sequential Social Dilemmas. In *Proceedings of the 16th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2017)*, 464–473.
- Li, C.; Deng, Z.; Lin, C.; Chen, W.; Fu, Y.; Liu, W.; Ab, W.; Wang, C.; and Shen, S. 2025. Dof: A diffusion factorization framework for offline multi-agent reinforcement learning. In *The Thirteenth International Conference on Learning Representations*.
- Ma, J.; and Wu, F. 2023. Learning to Coordinate from Offline Datasets with Uncoordinated Behavior Policies. In *Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023)*, Aamas, 1258–1266.
- Nair, A.; Gupta, A.; Dalal, M.; and Levine, S. 2020. AWAC: Accelerating Online Reinforcement Learning with Offline Datasets. *arXiv preprint arXiv:2006.09359*.
- Osband, I.; Blundell, C.; Pritzel, A.; and Van Roy, B. 2016. Deep exploration via bootstrapped DQN. *Advances in Neural Information Processing Systems*, 4033–4041.
- Pan, L.; Huang, L.; Ma, T.; and Xu, H. 2022. Plan Better Amid Conservatism: Offline Multi-Agent Reinforcement Learning with Actor Rectification. *Proceedings of Machine Learning Research*, 162: 17221–17237.
- Peng, X. B.; Kumar, A.; Zhang, G.; and Levine, S. 2019. Advantage-Weighted Regression: Simple and Scalable Off-Policy Reinforcement Learning. *arXiv preprint arXiv:1910.00177*.
- Rashid, T.; Samvelyan, M.; De Witt, C. S.; Farquhar, G.; Foerster, J.; and Whiteson, S. 2018. QMIX: Monotonic value function factorisation for deep multi-agent reinforcement Learning. *35th International Conference on Machine Learning, ICML 2018*, 10: 6846–6859.
- Shao, J.; Qu, Y.; Chen, C.; Zhang, H.; and Ji, X. 2023. Counterfactual Conservative Q Learning for Offline Multi-agent Reinforcement Learning. *37th Conference on Neural Information Processing Systems (NeurIPS 2023)*.
- Son, K.; Kim, D.; Kang, W. J.; Hostallero, D.; and Yi, Y. 2019. QTRAN: Learning to factorize with transformation for cooperative multi-agent reinforcement learning. *36th International Conference on Machine Learning, ICML 2019*, 2019-June: 10329–10346.
- Tampuu, A.; Matiisen, T.; Kodelja, D.; Kuzovkin, I.; Korjus, K.; Aru, J.; Aru, J.; and Vicente, R. 2017. Multiagent cooperation and competition with deep reinforcement learning. *PLoS ONE*, 12(4).
- Tseng, W. C.; Wang, T. H.; Yen-Chen, L.; and Isola, P. 2022. Offline Multi-Agent Reinforcement Learning with Knowledge Distillation. In *Advances in Neural Information Processing Systems*, volume 35.
- Wang, X.; Xu, H.; Zheng, Y.; and Zhan, X. 2023. Offline Multi-Agent Reinforcement Learning with Implicit Global-to-Local Value Regularization. In *37th Conference on Neural Information Processing Systems (NeurIPS 2023)*.
- Wang, X.; and Zhan, X. 2023. Offline Multi-Agent Reinforcement Learning with Coupled Value Factorization. In *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS*, volume 2023-May, 2781–2783.

Wong, A.; Bäck, T.; Kononova, A. V.; and Plaat, A. 2022. Deep multiagent reinforcement learning: challenges and directions. *Artificial Intelligence Review*.

Yang, Y.; Ma, X.; Li, C.; Zheng, Z.; Zhang, Q.; Huang, G.; Yang, J.; and Zhao, Q. 2021. Believe What You See: Implicit Constraint Approach for Offline Multi-Agent Reinforcement Learning. In *35th Conference on Neural Information Processing Systems (NeurIPS 2021)*.

Zhang, K.; Sun, T.; Tao, Y.; Genc, S.; Mallya, S.; and Basar, T. 2020. Robust multi-agent reinforcement learning with model uncertainty. In *34th Conference on Neural Information Processing Systems (NeurIPS 2020)*.

Zhu, Z.; Liu, M.; Mao, L.; Kang, B.; Xu, M.; Yu, Y.; Ermon, S.; Zhang, W.; and Jiao Tong University, S. 2024. MAD-IFF: Offline Multi-agent Learning with Diffusion Models. In *38th Conference on Neural Information Processing Systems (NeurIPS 2024)*.