B3C: A Minimalist Approach to Offline Multi-Agent Reinforcement Learning

Woojun Kim
Carnegie Mellon University
Pittsburgh, United States
woojunk@andrew.cmu.edu

Katia P. Sycara
Carnegie Mellon University
Pittsburgh, United States
sycara@andrew.cmu.edu

ABSTRACT

Overestimation arising from selecting unseen actions during policy evaluation is a major challenge in offline reinforcement learning (RL). A minimalist approach in the single-agent setting—adding behavior cloning (BC) regularization to existing online RL algorithms—has been shown to be effective in terms of achieving competitive performance with minimal modification to online algorithms; however, this approach is understudied in multi-agent settings. In particular, overestimation becomes worse in multi-agent settings due to the presence of multiple actions, resulting in the BC regularization-based approach easily suffering from either over-regularization or critic divergence. To address this, we propose a simple yet effective method, Behavior Cloning regularization with Critic Clipping (B3C), which clips the target critic value in policy evaluation based on the maximum return in the dataset and pushes the limit of the weight on the RL objective over BC regularization, thereby demonstrating superior performance across benchmarks. Additionally, we leverage existing value factorization techniques, particularly non-linear factorization, which is understudied in offline settings. Integrated with non-linear value factorization, B3C outperforms state-of-the-art algorithms on various offline multi-agent benchmarks.

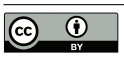
KEYWORDS

Offline Multi-Agent Reinforcement Learning; Value Factorization

ACM Reference Format:
Woojun Kim and Katia P. Sycara. 2026. B3C: A Minimalist Approach to Offline Multi-Agent Reinforcement Learning. In *Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), Paphos, Cyprus, May 25 – 29, 2026*, IFAAMAS, 9 pages. https://doi.org/10.65109/UOYL1486

1 INTRODUCTION

Reinforcement learning (RL) trains agents to maximize cumulative rewards through interaction with an environment in an online manner; however, such interaction can be time-consuming due to the sample inefficiency of RL and costly in safety-critical environments due to inherent risks [12, 19]. Furthermore, in many practical scenarios, only pre-collected datasets can be leveraged instead of interacting with the environment. These challenges have motivated a shift toward offline RL, which trains agents using pre-collected datasets without requiring interaction [2, 15, 25, 36].



Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), C. Amato, L. Dennis, V. Mascardi, J. Thangarajah (eds.), May 25 – 29, 2026, Paphos, Cyprus. © 2026 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). https://doi.org/10.65109/UOYL1486

A major challenge of offline RL is that bootstrapping from unseen actions in policy evaluation induces extrapolation error, leading to accumulated errors [5, 19], due to the inability to generate data. This issue consequently causes overestimation of value functions and adversely affects policy improvement. To address this, a variety of techniques, such as conservative value estimation [18] and modeling behavior policy [7], have been introduced. However, these approaches increase algorithmic complexity, hinder reproducibility, and amplify hyperparameter sensitivity. This motivates minimalist approaches, which introduce minimal changes to existing RL algorithms [5, 31]. One representative example is TD3+BC, which integrates behavior cloning (BC) regularization into an existing online RL algorithm called TD3 [5]. This simple approach does not require any additional components and introduces only one more hyperparameter for regularization, but it surprisingly achieves SOTA performance. This line of research, minimalist approaches, has been extensively investigated in single-agent settings [5, 31], but it has barely been explored in multi-agent settings, which entail more algorithmic complexity due to the presence of multiple agents.

In this paper, we aim to develop a minimalist approach that introduces minimal changes to existing multi-agent RL algorithms. One might ask: is adding BC regularization sufficient in multi-agent settings? We argue that BC regularization alone is not enough in multi-agent settings. This is because the overestimation becomes more severe in multi-agent settings. It arises from the use of a centralized critic, which conditions on multiple actions, increasing the likelihood of encountering unseen actions. Consequently, critic learning often becomes unstable, necessitating a greater reliance on BC regularization over the RL objective—resulting in over-regularization. This over-regularization inherently limits performance, as it becomes heavily dependent on the quality of the dataset. In addition, recent research on offline multi-agent RL overlooks existing value factorization techniques, such as non-linear decomposition—they either rely on linear factorization or disregard factorization entirely.

To address the aforementioned limitations, we propose a simple yet effective technique for offline multi-agent RL: **B**ehavior **C**loning regularization with **C**ritic **C**lipping (**B3C**). The proposed method alleviates overestimation by clipping the critic value during policy evaluation, using the maximum return from the given dataset, in addition to BC regularization. Critic clipping enables the use of a higher weight for the RL objective relative to BC regularization, resulting in improved performance. Furthermore, we integrate B3C with an existing online multi-agent RL algorithm called FACMAC [24], which utilizes factored critics and value factorization techniques, resulting in **FACMAC+B3C**. Notably, we provide a

design choice for value factorization in the offline setting, empirically demonstrating that non-monotonic factorization performs better than monotonic and linear factorization, which contrasts with findings in the online setting [24].

Despite its simplicity, the effectiveness of FACMAC+B3C is empirically demonstrated across a variety of offline multi-agent RL benchmarks, including multi-agent Mujoco and particle environments, with various types of dataset. We provide a thorough analysis demonstrating how the proposed method outperforms the baselines in terms of performance. Our main contributions are:

- We address the over-regularization problem in offline multi-agent RL and propose a minimalist framework that achieves SOTA performance with minimal modification to existing algorithms.
- We conduct an empirical study of value factorization in offline settings, an aspect that has been understudied.
- We conduct extensive experiments across seven environments and forty-two datasets, covering both fully and partially observable settings.

2 BACKGROUND AND RELATED WORK

2.1 Multi-Agent RL

We consider fully cooperative multi-agent tasks, which can be modeled as an N-agent decentralized partially observable MDP (Dec-POMDP) [3, 4, 22]. At each timestep t, each agent selects its own action, a_t^i, based on its local information such as the partial observation o_t^i. This forms a joint action $\boldsymbol{a}_t$ yielding the next state s_{t+1}, the joint observations $\boldsymbol{o}_{t+1} = (o_{t+1}^1, \cdots, o_{t+1}^N)$, and a shared reward r_t. The goal is to find the optimal joint policy that maximizes the team return, $\mathbb{E}[\sum_{l=0}^{\infty} \gamma^l r_l]$. For this, recent multi-agent RL algorithms adopt a framework of centralized training with decentralized execution (CTDE) [8, 11, 34, 39], where individual policies execute actions based on local information, e.g., Agent i's action-observation history τ^i, but are trained with global information such as the state.

One of the key challenges in Dec-POMDPs is to correctly assign credit to each agent for the team reward. For this, value factorization methods have been actively studied [21, 26, 29, 30, 32, 39]. Value factorization methods approximate the joint Q-function as a function of individual agent-wise value functions, thereby enabling effective credit assignment among agents. Linear factorization, such as Value Decomposition Networks (VDN) [30], assumes additive decomposition of individual Q^i values, while monotonic factorization in QMIX [26], enforces a monotonic mixing constraint to ensure consistency between local and global optima. Such methods have shown promising results in online multi-agent RL, but their applicability to offline settings remains unexplored.

FACMAC [24] is a representative multi-agent RL algorithm that trains a centralized but factored critic, which is decomposed into a nonlinear combination of individual critics, and uses it to train decentralized deterministic policies.The key idea is factorizing the centralized critic into individual critics via a non-linear function, where the centralized critic is decomposed as

$$Q_{jt}(s, \boldsymbol{\tau}, \mathbf{a}) = f_{mixer}(s, Q^1(\tau^1, a^1), \cdots, Q^N(\tau^N, a^N)), \quad (1)$$

where f_{mixer} is a mixer network that encodes state information and learns the weights of individual critics, and $Q^i(\tau^i, a^i)$ is the individual critic of Agent i. FACMAC considers two value factorization methods: (a) monotonic factorization (*mono*), which constrains the parameters of f_{mixer} to enforce $\partial Q_{jt}/\partial Q^i \geq 0$, as in QMIX [26]; and (b) non-monotonic factorization (*non-mono*), which removes the constraint from (a) to increase representational capacity. In addition to these two methods (c) linear factorization [30] (*vdn*) was also considered. It was shown that *non-mono* and *mono* approaches are complementary: *non-mono* performs better on tasks requiring greater representational capacity, while *mono* outperforms on others. However, *vdn* underperforms both and performs drastically worse, particularly in multi-agent Mujoco environment.

The centralized but factored critic, parameterized by $\boldsymbol{\theta_Q}$, is trained to minimize the temporal-difference (TD) error:

$$\mathcal{L}_{\text{FACMAC}}(\boldsymbol{\theta_Q}) = \mathbb{E}_{\mathcal{D}}\left[\left(y^{jt} - Q_{jt}(s, \boldsymbol{\tau}, \boldsymbol{a}; \boldsymbol{\theta_Q})\right)^2\right], \text{ where}$$
$$y^{jt} = r + \gamma Q_{jt}(s', \boldsymbol{\tau'}, \boldsymbol{\pi}(\boldsymbol{\tau'}); \boldsymbol{\theta_Q^-}), \quad (2)$$

$\mathcal{D}$ is the replay buffer, $\boldsymbol{\pi}$ is the deterministic joint policy, and $\boldsymbol{\theta_Q^-}$ is the parameter of the target critic networks. Based on this centralized critic, the decentralized policies, parameterized by $\boldsymbol{\theta_\pi} = \{\theta_\pi^i\}_{i=1}^N$, are trained using the deterministic policy gradient [28] where the loss function is written as $\mathcal{L}_{\text{FACMAC}}(\boldsymbol{\theta_\pi}) =$

$$\mathbb{E}_{\mathcal{D}}\left[-Q_{jt}(s, \boldsymbol{\tau}, \pi^1(\tau^1; \theta_\pi^1), \cdots, \pi^N(\tau^N; \theta_\pi^N))\right], \quad (3)$$

where θ_π^i is Agent i's actor parameter. A main benefit of FACMAC over prior works such as MADDPG [20] is leveraging non-linear factorization. However, in offline settings, value factorization techniques remain underexplored—recent research still relies on linear factorization [33, 37] or omits it by using a non-factored critic. In this paper, to investigate value factorization for offline settings, we adopt FACMAC to build upon the recent achievements.

2.2 Offline RL

Offline RL aims to learn a policy that maximizes the expected return from a fixed dataset $\mathcal{D}$, consisting of trajectories generated by arbitrary behavior policies, without additional environment interactions [9, 16, 18]. This inability to generate additional data introduces a major challenge in offline RL—*extrapolation error* in policy evaluation, i.e., the target value in the Bellman equation becomes inaccurate if actions from the learning policy are not included in the dataset. This extrapolation error often leads to overestimation and degrades performance. To address this challenge, several approaches have been proposed, such as behavior-constrained policy optimization, which regularizes the policy to stay close to the behavior policy [5, 35, 36, 38], and conservative value estimation [14, 18], which lower-bounds the value function by penalizing the values of unseen actions. These approaches have demonstrated effectiveness; however, they often require significant modifications or additional components to existing RL algorithms, such as generative models. These additions introduce more hyperparameters, which can affect performance and impact stability and reproducibility. This necessitates an offline RL algorithm with minimal changes from the current online RL algorithms. As an example, [5] proposed TD3 with Behavior Cloning (TD3+BC), which adds a BC loss function as a regularizer on top of the TD3 loss [6] to update the policy. This encourages the action generated by the learning policy to be the

same as the action in the dataset. The policy of TD3+BC is updated to minimize the following loss function.

$$\mathcal{L}_{\text{TD3+BC}}(\pi) = \mathbb{E}_{\mathcal{D}}\Big[-\boldsymbol{w}\underbrace{Q(s,\pi(s))}_{TD3} + \underbrace{(\pi(s)-a)^2}_{BC}\Big],$$
$$\text{where } \boldsymbol{w} = \frac{\alpha}{\frac{1}{N}\sum |Q(s,\pi(s))|} \tag{4}$$

where α is a hyperparameter that controls the balance between RL and BC. TD3+BC has been widely used due to its simplicity and performance comparable to SOTA algorithms [10].

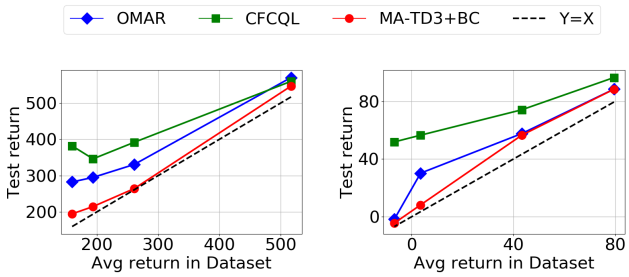
2.3 Offline Multi-Agent RL

A natural approach to offline multi-agent RL is to extend the previously mentioned single-agent offline RL to the multi-agent setting. However, naively extending these methods has been shown to be insufficient [23, 27]. For example, naive conservative value estimation for the joint policy can lead to excessively large penalties, significantly degrading performance as the number of agents increases [27]. Furthermore, the problem of extrapolation error worsens as the size of the action space grows with the number of agents [37], making it essential to balance conservatism and regularization with extrapolation error. To mitigate this problem, CFCQL [27] introduces counterfactual regularization to each agent individually to alleviate excessive conservatism. OMAR [23] introduces zeroth-order optimization to address policies getting stuck in poor local optima under conservative value function training. Note that both OMAR and CFCQL are based on conservative value estimation, which penalizes value functions for unseen action spaces, thereby forcing them to underestimate. OMIGA [33] introduces a framework that converts global regularization into implicit local-level regularization using linear value decomposition, while employing in-sampling techniques to avoid querying unseen actions. MADIFF [40] presents a diffusion-based generative approach for multi-agent systems, enabling decentralized policy development through effective teammate modeling.

Although the aforementioned offline multi-agent RL algorithms perform reasonably, they still require significant modifications from existing online multi-agent RL algorithms, as described in single-agent settings in Sec. 2.2, with even more adjustments needed due to the presence of multiple agents. Additionally, they have not fully leveraged the currently existing value factorization techniques, which are central to the recent success of multi-agent RL. Therefore, inspired by [5], we aim to develop a minimalist approach that minimally modifies an existing multi-agent RL algorithm, focusing on simplicity and practicality while incorporating existing multi-agent RL techniques.

3 METHODOLOGY

To achieve the aforementioned goal, we focus on a BC regularization-based approach. In multi-agent settings, the presence of multiple agents amplifies the issue of overestimation, as mentioned in Sec.2.3. Consequently, the BC regularization-based approach is susceptible to either over-regularization or unstable critic learning. This occurs because the joint action space grows exponentially with the number of agents, increasing the likelihood of encountering unseen actions during policy evaluation. As a result, critic updates become unstable and prone to divergence, which in turn encourages excessive reliance on the BC term to stabilize training, leading to over-regularization and performance degradation when the dataset quality is limited.



(a) CN (b) World

Figure 1: Test return as a function of dataset quality (average dataset return) in CN and World, based on [27]. Proximity to $y = x$ reflects strong data dependence. MA-TD3+BC remains close to dataset-level performance, indicating over-regularization by behavior cloning.

To address these challenges, we propose a simple yet effective regularization method called **B**ehavior **C**loning with **C**ritic **C**lipping (**B3C**), which prevents over-regularization while ensuring stability, allowing a focus on the RL objective and thereby significantly improving performance. Additionally, we explore the application of value factorization techniques with B3C, which remains understudied.

3.1 Revisiting Behavior Cloning: Over-regularization

We begin by revisiting BC regularization in offline multi-agent RL. Building on the success of BC regularization in the single-agent setting, recent studies on offline multi-agent RL [23, 27] have introduced an extension of TD3+BC, called MA-TD3+BC, as a baseline by adapting Eq. 4 to the multi-agent setting. Although MA-TD3+BC achieves reasonable performance, it is less effective than the proposed methods, particularly when applied to low-quality datasets. Fig. 1 illustrates the test returns of OMAR [23], CFCQL [27], and MA-TD3+BC with respect to the average return of the dataset, reported in [27]. MA-TD3+BC achieves performance close to dataset-level quality, but with a small margin. In contrast, OMAR and CFCQL show significant improvements over dataset quality, particularly for low-quality datasets. This observation, that MA-TD3+BC heavily depends on data quality, indicates that it is over-regularized by behavior cloning.

One might ask: what if we place more weight on the RL objective in Eq. 4, which can be achieved by increasing α in Eq. 4, to reduce dependence on data quality? We observed that increasing α sometimes leads to better performance compared to prior work; however, it often fails to converge. The critic is updated unstably and even frequently diverges due to extrapolation error, as we will discuss in Sec. 4.3.1. Therefore, to achieve stable performance, α must be kept low, which results in over-regularization and limits performance based on dataset quality.

3.2 Critic Clipping: Alleviate Overestimation

In order to simultaneously avoid unstable critic training and over-regularization, we propose a simple yet effective technique called Critic Clipping (CC) for training the centralized critic, which integrates with BC regularization for training policies.

CC clips the critic target value when calculating the target value during policy evaluation, which can lead to overestimation based on the maximum return in the dataset. Unlike conservative value estimation, which requires additional sampling or modeling of the behavior policy and introduces underestimation bias, CC is easy to implement and proves effective by limiting overestimation to below a certain threshold. Intuitively, this prevents the critic from propagating excessively optimistic targets that would otherwise destabilize value learning across agents. Here, the reason for using the maximum return rather than the average return as the clipping standard is as follows: The learned RL policy is expected to outperform the behavior policy, and therefore, the expected return, which the critic approximates, is expected to be larger than the average return in the dataset. For simplicity, we use the maximum return, representing the best episode in the dataset. To allow further flexibility, we can use a scaled value of the maximum return for the clipping value. Note that we observe using the maximum return is sufficient empirically, as we will discuss in Sec.4.3.2. The corresponding loss function of the critic is given by

$$\mathcal{L}_{CC}(\boldsymbol{\phi}, \psi) = \mathbb{E}_{(s,\boldsymbol{\tau},\boldsymbol{a},r,s',\boldsymbol{\tau}')\sim\mathcal{D}}\left[\left(y^{jt} - Q_{jt}(s,\boldsymbol{\tau},\boldsymbol{a};\boldsymbol{\phi})\right)^2\right], \text{ where}$$
$$y^{jt} = r + \gamma \underbrace{\text{Min}\left[Q_{jt}(s',\boldsymbol{\tau}',\boldsymbol{\pi}(\boldsymbol{\tau}';\boldsymbol{\theta}^-);\boldsymbol{\phi}^-),\ R^*\right]}_{\text{Critic Clipping}} \text{ and} \tag{5}$$

$R^* = M \times \max_d \sum_{t=1}^{T} r_{d,t}$. Here, M is a scalar value, and $\max_d \sum_{t=1}^{T} r_{d,t}$ represents the maximum return in the dataset where $r_{d,t}$ is the reward at time-step t of d-th episode in the dataset. For simplicity, we use $M = 1$ in most cases during the experiments. We observed that this straightforward choice is sufficient in most cases, aligning with our minimalist design philosophy and reducing the burden of hyperparameter tuning. Although M can be fine-tuned (e.g., smaller M performed slightly better in one environment; see Appendix A), we did not observe a clear or consistent improvement from adjusting it. Additionally, this clipping does not constrain the learned policy to achieve returns below the maximum return in the dataset. The critic value provides an approximation of policy performance but does not constitute a hard upper bound, as its estimates are used for optimization guidance rather than to determine the final achievable return. We provide an ablation study on this in Sec. 4.3.2.

3.3 B3C with Value Factorization

We integrate CC with BC regularization, introducing the proposed method, B3C. The main advantage of CC is its flexibility, which reduces sensitivity to BC regularization and allows for maximizing the weight of the RL objective. This ultimately leads to improved performance. Additionally, we adopt FACMAC, which uses a centralized but factored critic with non-linear value factorization. Accordingly, the policy loss function of FACMAC+B3C is given by

$$\mathcal{L}_{\text{FACMAC+B3C}}(\boldsymbol{\theta}_{\boldsymbol{\pi}}) =$$
$$\mathbb{E}_{\mathcal{D}}\Big[-\boldsymbol{w} \underbrace{Q_{jt}(s,\boldsymbol{\tau},\pi^1(\tau^1),\cdots,\pi^N(\tau^N))}_{FACMAC} + \beta \sum_{i=1}^{N} \underbrace{(\pi^i(\tau^i) - a^i)^2}_{BC} \Big]$$
$$\text{where } \boldsymbol{w} = \frac{\alpha}{\frac{1}{N}\sum |Q_{jt}(s,\boldsymbol{\tau},\boldsymbol{a})|}, \tag{6}$$

$\boldsymbol{\pi} = (\pi^1, \cdots, \pi^N)$ and $\boldsymbol{a} = (a^1, \cdots, a^N)$ represent the joint policy and the joint action, respectively. Here, $\boldsymbol{\theta}_{\boldsymbol{\pi}}$ denotes the parameters of the joint policy, and $\mathcal{D}$ represents the given offline dataset. We normalize the Q-value based on the sample batch to ensure robustness to scale, following [5]. Note that there are two important hyperparameters, α and β, which balance the trade-off between RL and BC. We will refer to α and β as the RL coefficient and the BC coefficient, respectively. A key difference from [5] is the introduction of the BC coefficient, β, alongside the RL coefficient, α. This decouples the relative RL-BC weighting and the overall objective scale, enabling independent control of regularization strength, which can affect optimization in deep networks. For hyperparameter tuning, we recommend practitioners start by fixing $\beta = 1$ and tuning α, following [5]. If over-regularization persists, β can then be adjusted, which has been shown to be effective in multi-agent particle environments.

Empirical Observation regarding value factorization: We consider three value factorization methods discussed in Sec.2.1: *non-mono*, *mono*, and *vdn*. We empirically observe that *non-mono* outperforms *mono* in most cases, which differs from the findings in the online setting discussed in Sec. 2.1. We argue that limiting representational capacity, which is inherent to monotonic factorization, leads to reduced performance in offline settings. We provide an ablation study for this in Sec. 4.3.3.

As summarized above, we update the joint policy to maximize the critic with BC regularization by minimizing Eq. 6, and we update the centralized critic to minimize the TD error with clipping, as shown in Eq. 5. We refer to the proposed algorithm—FACMAC integrated with Behavior Cloning regularization and Critic Clipping—as FACMAC+B3C. Additionally, we also apply B3C to MA-TD3, referred to as MA-TD3+B3C.

4 EXPERIMENTS

4.1 Experimental Setup

Environments and Offline Dataset. We consider various multi-agent environments categorized by their types, the number of agents, and levels of partial observability. Additionally, we use offline datasets generated by different MARL algorithms, covering various levels.

(1) Three multi-agent particle environments (provided by [23] and generated using MA-TD3 [1]): (a) Cooperative Navigation (CN), where three agents cover three landmarks without colliding, requiring coordination and teamwork; (b) Predator-Prey (PP), where three slower predators cooperate to capture a faster, pre-trained prey; and (c) World, where four predators attempt to catch two prey that aim to eat food while strategically using forest hiding spots for protection. For each task, evaluation is performed using four datasets of varying quality: expert, medium, medium-replay, and random.

Table 1: Performance of MA-TD3+B3C and baselines on multi-agent particle environments, averaged over five seeds. Tasks include Cooperative Navigation (CN), Predator-Prey (PP), and World, each evaluated on expert (e), medium (m), medium-replay (mr), and random (r) datasets. "Avg R" and "Max R" denote the dataset average and maximum return. Boldface indicates the best or statistically comparable performance.

Task-Dataset	Avg R	Max R	OMAR	CFCQL	MADIFF	TD3+BC	TD3+BC (ours)	TD3+B3C (ours)
CN-e	100.0	163.1	**114.9±2.6**	112.0±4.0	95.0±5.3	108.3±3.3	100.6±8.3	102.3±6.7
CN-mr	9.5	123.2	37.9±12.3	**52.2±9.6**	30.3±2.5	15.4±5.6	**53.8±4.8**	**53.8±4.8**
CN-m	27.8	145.0	47.9±18.9	**65.0±10.2**	64.9±7.7	29.3±4.8	55.9±25.8	**63.5±9.7**
CN-r	0.0	103.7	6.9±3.1	62.2±8.1	6.9±3.1	9.8±4.9	**72.6±7.0**	**73.3±6.6**
PP-e	100.0	275.7	116.2±19.8	118.2±13.1	120.9±14.6	115.2±12.5	108.1±22.9	**124.6±21.2**
PP-mr	9.6	128.0	47.1±15.3	71.1±6.0	62.3±9.2	28.7±20.9	**75.2±6.8**	**76.5±8.6**
PP-m	48.9	196.2	66.7±23.2	68.5±21.8	77.2±10.4	65.1±29.5	91.3±35.7	**104.3±11.9**
PP-r	0.0	61.4	11.1±2.8	78.5±15.6	3.2±4.0	5.7±3.5	**105.0±16.7**	**105.5±10.0**
World-e	100.0	270.7	110.4±25.7	119.7±26.4	**122.6±14.4**	110.3±21.3	120.7±17.9	**124.3±5.1**
World-mr	12.0	138.7	42.9±19.5	**73.4±23.2**	57.1±10.7	17.4±8.1	**75.2±23.1**	**75.8±10.9**
World-m	58.1	206.8	74.6±11.5	93.8±31.8	**123.5±4.5**	73.4±9.3	119.3±18.2	119.7±16.7
World-r	0.0	63.1	5.9±5.2	68.0±20.8	2.0±3.0	2.8±5.5	98.3±49.7	**101.0±16.8**

(2) Three fully observable multi-agent MuJoCo environments (provided in [33] and generated using HAPPO [17]): We consider three tasks including 3-Agent Hopper (HC), 2-Agent Ant (Ant), and 6-Agent HalfCheetah (HC). For each task, evaluation is performed using four datasets of varying quality: expert, medium, medium-replay, and medium-expert.

(3) Two partially observable multi-agent MuJoCo environments: 6-Agent HalfCheetah and 5-Agent Swimmer, each with differing degrees of partial observability and performance levels. We generated the offline datasets using ADER [13] with varying levels of performance: expert, medium-1, medium-2, and their combinations. In both environments, each agent can observe its K nearest neighbors. Here, K determines the degree of partial observability. We consider $K = 0, 1$ for HC, which is denoted as HC-kK, and $K = 0$ for Swimmer (SW).

Baselines. We compare the proposed method against various baselines for each environment. For the multi-agent particle environments, we use MA-TD3+BC [23], OMAR [23], MADIFF [40], and CFCQL [27] as baselines. For multi-agent Mujoco environments, we use OMAR [23], CFCQL [27], and OMIGA [33]. In addition, we include TD3+BC, FACMAC+BC, and TD3+B3C to analyze the effect of value factorization and CC.

Implementation and Hyperparameters. To ensure a fair comparison, we implement the proposed method using the released CFCQL [27] code for the multi-agent particle environments and the released OMIGA [33] code for the multi-agent Mujoco environments. The RL and BC coefficients in Eq. 6 are critical hyperparameters in our approach. As mentioned in Sec. 3.3, we initially set $\beta = 1$ and tune α for each task, as done in [5]. We observe that $\beta = 1$ is sufficient for multi-agent Mujoco, but tuning β improves performance in the multi-agent particle environments. For another hyperparameter, the scalar clipping value M in Eq. 5, we use $M = 1$ for simplicity, except for the Ant task with the medium-replay dataset, where a lower M significantly improves performance. An ablation study on this value is provided in Sec. 4.3.2. The detailed hyperparameters used are listed in Appendix A. Appendix is available at https://arxiv.org/pdf/2501.18138.

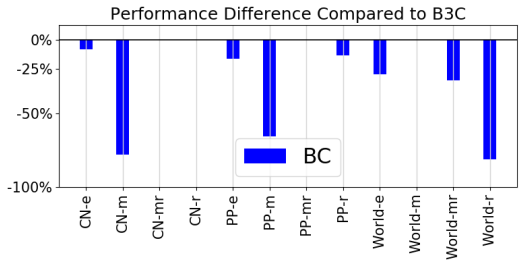


Figure 2: Worst-seed performance gap between MA-TD3+BC and MA-TD3+B3C. Negative values indicate that MA-TD3+BC performs worse (i.e., a gap of -5% means its worst-seed performance is 5% lower than that of MA-TD3+B3C).

4.2 Numerical Results

Multi-agent particle environments. The results are shown in Table 1. As discussed in Sec. 3.1, MA-TD3+BC from prior works suffers from over-regularization, which is evident from its performance on medium-replay and random datasets, where it significantly underperforms compared to the baselines. Thus, we first further tune the coefficients of RL and BC, which is referred to as MA-TD3+BC (ours) in Table 1. This simple tuned version substantially improves performance and even outperforms the baselines in some cases. However, this simple hyperparameter tuning is insufficient: *BC regularization alone is often unstable*. We claim that the proposed CC can address this instability. Fig. 2 illustrates the performance difference between the worst-performing seed among 5 seeds of MA-TD3+BC and that of MA-TD3+B3C, where -5% indicates that the worst-performing seed of MA-TD3+BC performs 5% worse than MA-TD3+B3C. As shown in Fig. 2, using BC alone results in a significant performance drop in the worst-performing seed compared to B3C due to its instability, demonstrating that B3C, which incorporates critic clipping, stabilizes learning. Additionally, in terms of average performance, TD3+B3C outperforms the baselines in most cases. In particular, on the medium-replay and random datasets, which are generated by multiple behavior policies, TD3+B3C achieves state-of-the-art performance with large margins.

Multi-agent Mujoco. In this environment, we evaluate FACMAC+B3C, which combines non-monotonic value factorization

Table 2: Results of FACMAC+B3C and baselines on six Mujoco tasks across multiple datasets. Both maximum and average returns are reported. Dataset types are abbreviated as: expert (e), medium1 (m1), and medium2 (m2). Combined datasets are denoted as e-m1 and m1-m2. Bold indicates the best or comparable performance.

Task-Dataset	Avg R	Max R	OMAR	CFCQL	OMIGA	TD3+BC	FACMAC+BC	TD3+B3C	FACMAC+B3C
Fully observable multi-agent MuJoCo provided in OMIGA [33] and generated by HAPPO [17]									
Hop-e	2452.0	3762.7	2.4	718.5	859.6	3598	3621	1549	**3619.7±1.6**
Hop-m	723.6	2776.5	21.3	674.2	1189.3	1487	2880	1549	**3242.7±129.1**
Hop-mr	746.4	2801.2	3.3	**1380.2**	774.2	330	143	263	736.8±469.4
Hop-me	1190.6	3762.7	1.4	383.0	709.0	3212	3502	3356	**3328.0±369.8**
Ant-e	2055.1	2124.2	313.5	1756.1	2055.5	2201	2153	2200	**2162.8±46.0**
Ant-m	1418.7	1473.9	-1710.0	1159.6	1418.4	1048	1042	1061	**1516.5±14.8**
Ant-mr	1029.5	1517.1	-2014.2	1052.9	1105.1	959	968	1226	**1259.8±302.4**
Ant-me	1736.9	2124.2	-2992.8	613.2	1720.3	2060	931	2174	**2077.6±194.2**
HC-e	2785.1	3866.1	-206.7	4999.2	3383.6	4777.2	1043.2	4724.3	**5403.5±169.7**
HC-m	1415.7	2113.5	-265.7	4345.0	3608.1	2984.1	4667.7	2995.2	**4756.6±56.3**
HC-mr	655.8	2132.6	-235.4	3655.3	2504.7	3652.7	4538.6	3635.9	**4602.6±150.2**
HC-me	2105.4	3866.1	-253.8	5030.9	2948.5	4826.3	5395.2	4860.6	**5413.7±99.4**
Partial observable multi-agent MuJoCo generated by ADER [13]									
HC-k0-e	1394.3	1520.8	197.2	750.7	1390.1	-145	1381	1316	**1396.8±4.5**
HC-k0-m1	1103.3	1332.9	189.6	443.7	1106.3	1079	1158	1078	**1141.6±18.9**
HC-k0-m2	840.8	1157.4	839.1	766.5	847.6	967	1197	963	**1195.0±51.9**
HC-k0-e-m1	1252.3	1520.8	136.4	542.6	1199.5	590	784	1209	**1307.1±27.3**
HC-k0-e-m2	1121.7	1520.8	299.2	398.4	1097.9	98	1211	1080	**1230.0±40.5**
HC-k0-m1-m2	976.2	1332.9	709.1	1186.1	1027.3	1007	1107	995	**1210.1±22.8**
HC-k1-e	3766.0	3863.3	3232.6	3390.2	3089	3731	3748	**3760.8**	**3760.5±24.2**
HC-k1-m1	1976.2	2350.0	2312.4	2356.0	2017.3	2182	2413	2200	**2508.1±55.7**
HC-k1-m2	1223.9	1758.4	1108.9	1440.6	1196.5	1244	1284	1387	**2187.8±66.7**
HC-k1-e-m1	2873.2	3863.3	2296.4	1949.6	2112.3	1892	2611	2380	**2682.0±175.4**
HC-k1-e-m2	2486.0	3863.3	524.8	864.7	**1088.2**	992	1106	1211	**1102.8±349.8**
HC-k1-m1-m2	1591.6	2237.2	1410.6	1862.2	1633.5	1926	1186	2021	**2222.6±84.1**
Sw-e	430.9	438.9	395.3	403.3	**430.7**	371.0	424.1	428.2	**430.3±2.6**
Sw-m1	290.4	306.3	268.4	277.1	288.3	295.6	288.8	**297.0**	289.5±1.1
Sw-m2	142.7	158.5	97.9	130.0	142.6	140.1	76.6	147.1	**155.3±1.7**
Sw-e-m1	360.6	438.9	216.5	292.1	261.4	68.6	99.9	288.4	**297.1±53.8**
Sw-e-m2	286.8	438.9	113.8	138.6	148.2	160.8	213	**236.1**	211.9±8.4
Sw-m1-m2	216.3	306.3	226.4	222.7	205.8	185.4	200.7	**228.4**	189.9±7.7

with B3C. For fully observable cases, we use the reported results of OMAR and OMIGA [33], and generate results for TD3+BC, FACMAC+BC, and TD3+B3C for comparison. As shown in Table 2, FACMAC+B3C achieves the best overall performance across environments, often exceeding the dataset's maximum return, particularly on HalfCheetah and several Hopper and Ant tasks. Moreover, unlike the baselines, FACMAC+B3C exhibits no performance collapse across any dataset, indicating stable and consistent learning.

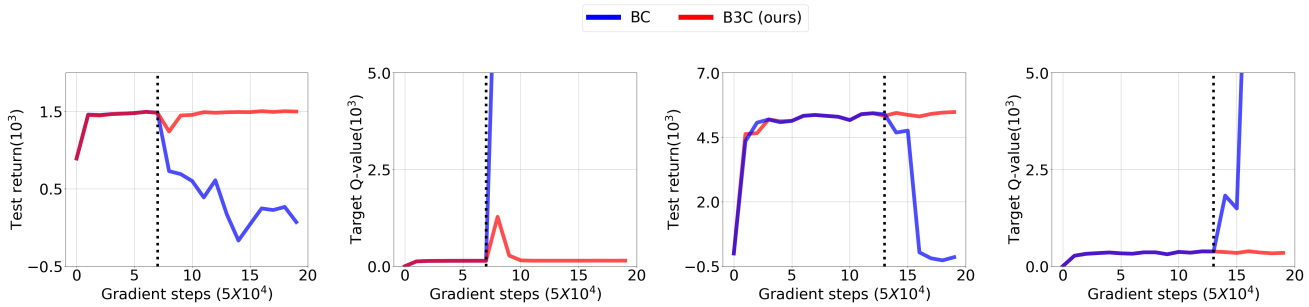
The consistent improvement of FACMAC+B3C over FACMAC+BC, as well as TD3+B3C over TD3+BC, demonstrates the effectiveness of `Critic Clipping` in stabilizing value estimation and improving overall policy quality. This result highlights that the benefit of B3C is not limited to a specific architecture but generalizes across both factored and non-factored critics.

Furthermore, FACMAC+BC generally outperforms TD3+BC except for a few fully observable tasks such as Ant-me and HC-e, which aligns with findings from online MARL literature where value factorization improves coordination and representation efficiency. These results collectively confirm that integrating value factorization with B3C provides both high and stable performance in complex multi-agent continuous control settings.

4.3 Analysis

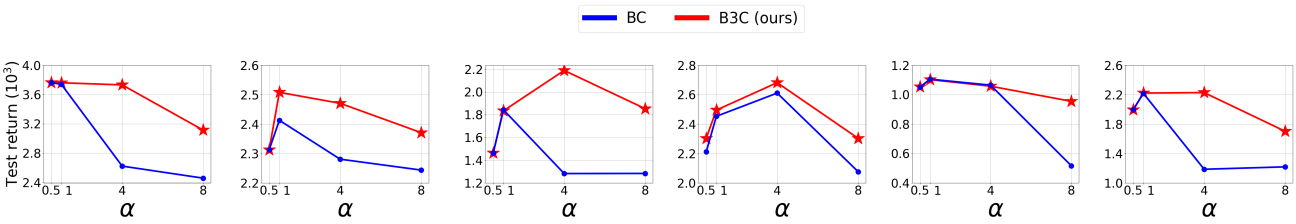
4.3.1 Critic Clipping: B3C versus BC. **Critic clipping alleviates overestimation** by directly constraining the target value in policy evaluation. As mentioned earlier, BC regularization alone often makes learning unstable, and even the critic diverges. We have already shown instability problem in Sec. 4.2 through the worst performance of MA-TD3+BC in the multi-agent particle environments. Here, we provide further analysis of how critic clipping addresses this issue for better understanding in multi-agent Mujoco environments.

Fig. 3 presents the test return and the target value, y^{jt} from Eq. 5, during the training of FACMAC+BC and FACMAC+B3C. The results indicate that FACMAC+BC, which relies solely on BC regularization, results in diverging value estimates. When this divergence occurs, marked by the dotted line in Fig. 3, the performance correspondingly degrades. However, B3C effectively mitigates this divergence—it occasionally tends to overestimate but either stabilizes quickly or avoids divergence entirely, ensuring more consistent performance. Note that this phenomenon does not always occur but is more likely when the RL objective is weighted more heavily



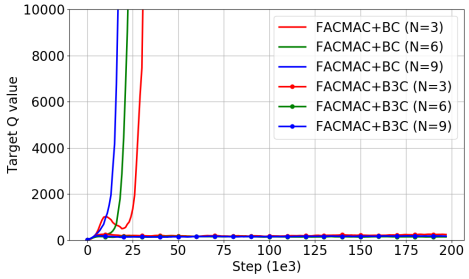
(a) Ant2x4-return (b) Ant2x4-target (c) HC6x1-return (d) HC6x1-target

Figure 3: Test return and target value during policy evaluation in the training of BC (blue) and B3C (red). The black dotted line indicates the moment when the target value starts to diverge.



(a) HC-k1-e (b) HC-k1-m1 (c) HC-k1-m2 (d) HC-k1-e-m1 (e) HC-k1-e-m2 (f) HC-k1-m1-m2

Figure 4: Performance with respect to the RL coefficient α in the partially observable Halfcheetah environment. Larger α increases the weight on the RL objective relative to BC regularization.



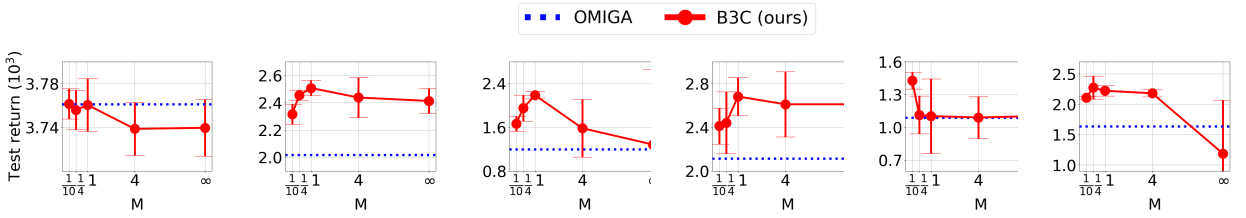
Method	PP (N=3)	PP (N=6)	PP (N=9)
FACMAC+BC	71.1 ± 3.6	74.8 ± 20.1	85.6 ± 47.8
FACMAC+B3C	97.1 ± 6.0	90.8 ± 12.4	104.5 ± 16.8

Figure 5: Performance and target Q-values of FACMAC with BC and B3C for varying N.

than BC regularization; however, critic clipping can prevent it and enable stable learning.

Additionally, we evaluate FACMAC+BC and FACMAC+B3C in a partially observable Predator-Prey environment [24] with 3, 6, and 9 agents. This experiment aims to examine how the degree of overestimation changes as the number of agents N increases. For a fair comparison, we normalize each dataset return by its average to compare target Q-values across the 3-, 6-, and 9-agent settings. As shown in Fig. 5, the target Q-values of FACMAC+BC diverge more rapidly as N increases, indicating that overestimation becomes more severe with a larger number of agents due to the exponentially growing joint action space. In contrast, critic clipping in FACMAC+B3C effectively suppresses this divergence, keeping target values bounded even under increased coordination complexity. This stability corresponds to higher and more consistent performance, as shown in the table below, indicating that B3C maintains robustness as the number of agents increases and alleviates the instability induced by large-scale coordination in multi-agent offline RL.

Critic clipping improves performance and robustness by enabling a higher weight on the RL objective, overcoming over-regularization while ensuring stability. As discussed in Sec. 3.1, prior works employing BC regularization-based approaches suffered from over-regularization, which constrained performance to the quality of the data. This issue is inevitable without the proposed critic clipping, as increasing the weight on the RL objective induces instability, making divergence more likely, as observed in the worst-performing comparison between BC and B3C shown in Fig. 2. However, CC allows for reduced BC regularization, i.e., a higher weight on the RL objective, resulting in improved performance. We investigate this by comparing FACMAC+BC and FACMAC+B3C with respect to the RL objective weight α, while fixing $\beta = 1$. The corresponding results are shown in Fig. 4. In most cases, FACMAC+BC performs worse when $\alpha > 1$, whereas FACMAC+B3C performs better or equal when $\alpha = 4$ in most cases. Additionally, although FACMAC+B3C experiences a performance drop when



(a) HC-k1-e (b) HC-k1-m1 (c) HC-k1-m2 (d) HC-k1-e-m1 (e) HC-k1-e-m2 (f) HC-k1-m1-m2

Figure 6: Ablation study on the clipping value, M: Performance of FACMAC+B3C with respect to M. $M = \infty$ (no clip) corresponds to FACMAC+BC. The performance of OMIGA is also included as a blue dotted line. Error bars represent one standard deviation.

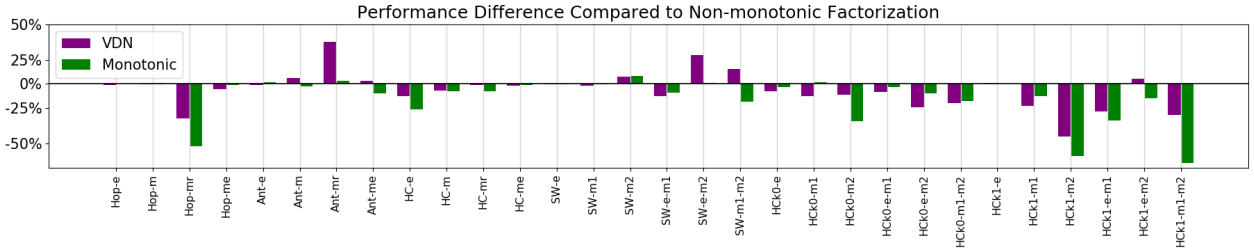


Figure 7: Ablation study on value factorization: Performance differences of VDN and monotonic factorization compared to non-monotonic factorization.

$\alpha = 8$, it remains more stable than FACMAC+BC, demonstrating the robustness of B3C across different values of α. Additionally, we provide the averaged performance difference between BC and B3C in multi-agent Mujoco in the Appendix B.

4.3.2 Critic Clipping: Clipping Value. `CC` clips the Q-function in the target value by the scaled maximum return in the dataset. This introduces a hyperparameter—the clipping value, M in Eq. 5. As discussed in Sec. 3.2, $M = 1$ is sufficient since the maximum return in the dataset is already high enough to handle overestimation. We conduct an ablation study regarding the clipping value. Fig. 6 shows the performance of FACMAC+B3C by varying M across 1/10, 1/4, 1, and 4. Additionally, we include the performance of FACMAC+BC, which does not use critic clipping, as $M = \infty$, and OMIGA [33] as a blue dotted line. It is seen that $M = 1$ achieves the best performance in all cases except for the HC-k1-e-m2 environment. While extremely small or large values of M perform worse than $M = 1$, FACMAC+B3C with any of the tested M values still outperforms both FACMAC+BC and OMIGA. In the HC-k1-e-m2 environment, where FACMAC+B3C performs suboptimally compared to the dataset quality, reducing M improves performance. Overall, $M = 1$ is a reasonable choice, providing the best performance while minimizing the burden of hyperparameter tuning.

4.3.3 Value factorization. We here provide an ablation study of the empirical observation described in Sec. 3.3 that non-monotonic factorization performs better than monotonic factorization in offline settings. Fig. 7 shows the performance differences of VDN and monotonic factorization compared to non-monotonic factorization across all environments. Overall, `non-mono` consistently outperforms both `mono` and `vdn`, though the linear factorization occasionally achieves comparable or slightly higher scores in certain easy datasets.

This result contrasts with prior findings in online settings [24], where monotonic factorization often yielded stable and competitive performance. We attribute this difference to the distributional shift inherent in offline learning: enforcing monotonicity restricts the representational capacity of the critic, making it difficult to approximate complex joint action dependencies when unseen samples dominate the dataset. In contrast, non-monotonic factorization allows the critic to flexibly capture nonlinear inter-agent correlations, leading to more accurate value estimation and better learning.

5 CONCLUSION

In this paper, we have proposed a simple regularization technique named behavior cloning with critic clipping for offline multi-agent RL that addresses overestimation and over-regularization while ensuring stability. In addition, we have investigated existing value factorization techniques in offline settings, providing the observation that non-monotonic factorization performs better than monotonic factorization. Numerical results show that B3C with non-monotonic factorization yields outstanding performance across the considered environments with diverse dataset levels. We have also provided several analyses and ablation studies to illustrate how the proposed method operates and affects learning.

Limitation Our work lacks theoretical analysis and is empirically driven. We consider this adequate for our minimalist objective and leave theoretical extensions for future work.

ACKNOWLEDGMENTS

This work was supported by the DARPA EMHAT Program under Agreement No. HR00112490409 and by the ONR Award No. N00014-23-1-2840.

REFERENCES

[1] Johannes Ackermann, Volker Gabler, Takayuki Osa, and Masashi Sugiyama. 2019. Reducing overestimation bias in multi-agent domains using double centralized critics. *arXiv preprint arXiv:1910.01465* (2019).

[2] Rishabh Agarwal, Dale Schuurmans, and Mohammad Norouzi. 2020. An optimistic perspective on offline reinforcement learning. In *International conference on machine learning*. PMLR, 104–114.

[3] Christopher Amato, Girish Chowdhary, Alborz Geramifard, N Kemal Üre, and Mykel J Kochenderfer. 2013. Decentralized control of partially observable Markov decision processes. In *52nd IEEE Conference on Decision and Control*. IEEE, 2398–2405.

[4] Jilles Dibangoye and Olivier Buffet. 2018. Learning to act in decentralized partially observable MDPs. In *International Conference on Machine Learning*. PMLR, 1233–1242.

[5] Scott Fujimoto and Shixiang Shane Gu. 2021. A minimalist approach to offline reinforcement learning. *Advances in neural information processing systems* 34 (2021), 20132–20145.

[6] Scott Fujimoto, Herke Hoof, and David Meger. 2018. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*. PMLR, 1587–1596.

[7] Scott Fujimoto, David Meger, and Doina Precup. 2019. Off-policy deep reinforcement learning without exploration. In *International conference on machine learning*. PMLR, 2052–2062.

[8] Jeewon Jeon, Woojun Kim, Whiyoung Jung, and Youngchul Sung. 2022. Maser: Multi-agent reinforcement learning with subgoals generated from experience replay buffer. In *International Conference on Machine Learning*. PMLR, 10041–10052.

[9] Jeonghye Kim, Suyoung Lee, Woojun Kim, and Youngchul Sung. 2024. Adaptive Q-Aid for Conditional Supervised Learning in Offline Reinforcement Learning. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

[10] Jeonghye Kim, Suyoung Lee, Woojun Kim, and Youngchul Sung. 2024. Decision ConvFormer: Local Filtering in MetaFormer is Sufficient for Decision Making. In *The Twelfth International Conference on Learning Representations*.

[11] Woojun Kim, Whiyoung Jung, Myungsik Cho, and Youngchul Sung. 2023. A Variational Approach to Mutual Information-Based Coordination for Multi-Agent Reinforcement Learning. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*. 40–48.

[12] Woojun Kim, Yongjae Shin, Jongeui Park, and Youngchul Sung. 2023. Sample-Efficient and Safe Deep Reinforcement Learning via Reset Deep Ensemble Agents. In *Advances in Neural Information Processing Systems*, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (Eds.), Vol. 36. Curran Associates, Inc., 53239–53260. https://proceedings.neurips.cc/paper_files/paper/2023/file/a6f6a5c517b2b92f3d309786af64086c-Paper-Conference.pdf

[13] Woojun Kim and Youngchul Sung. 2023. An adaptive entropy-regularization framework for multi-agent reinforcement learning. In *International Conference on Machine Learning*. PMLR, 16829–16852.

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

[15] Ilya Kostrikov, Ashvin Nair, and Sergey Levine. [n.d.]. Offline Reinforcement Learning with Implicit Q-Learning. In *International Conference on Learning Representations*.

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

[17] Jakub Grudzien Kuba, Ruiqing Chen, Muning Wen, Ying Wen, Fanglei Sun, Jun Wang, and Yaodong Yang. 2021. Trust region policy optimisation in multi-agent reinforcement learning. *arXiv preprint arXiv:2109.11251* (2021).

[18] Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. 2020. Conservative q-learning for offline reinforcement learning. *Advances in Neural Information Processing Systems* 33 (2020), 1179–1191.

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

[20] Ryan Lowe, Yi I Wu, Aviv Tamar, Jean Harb, OpenAI Pieter Abbeel, and Igor Mordatch. 2017. Multi-agent actor-critic for mixed cooperative-competitive environments. *Advances in neural information processing systems* 30 (2017).

[21] Enrico Marchesini, Andrea Baisero, Rupali Bhati, and Christopher Amato. 2025. On Stateful Value Factorization in Multi-Agent Reinforcement Learning. In *Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems*. 1445–1453.

[22] Frans A Oliehoek. 2012. Decentralized pomdps. In *Reinforcement learning: state-of-the-art*. Springer, 471–503.

[23] Ling Pan, Longbo Huang, Tengyu Ma, and Huazhe Xu. 2022. Plan better amid conservatism: Offline multi-agent reinforcement learning with actor rectification. In *International conference on machine learning*. PMLR, 17221–17237.

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

[25] Zhiyong Peng, Changlin Han, Yadong Liu, and Zongtan Zhou. 2023. Weighted policy constraints for offline reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37. 9435–9443.

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

[27] Jianzhun Shao, Yun Qu, Chen Chen, Hongchang Zhang, and Xiangyang Ji. 2024. Counterfactual Conservative Q Learning for Offline Multi-agent Reinforcement Learning. *Advances in Neural Information Processing Systems* 36 (2024).

[28] David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. 2014. Deterministic policy gradient algorithms. In *International conference on machine learning*. Pmlr, 387–395.

[29] Kyunghwan Son, Daewoo Kim, Wan Ju Kang, David Earl Hostallero, and Yung Yi. 2019. Qtran: Learning to factorize with transformation for cooperative multi-agent reinforcement learning. In *International conference on machine learning*. PMLR, 5887–5896.

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

[31] Denis Tarasov, Vladislav Kurenkov, Alexander Nikulin, and Sergey Kolesnikov. 2024. Revisiting the minimalist approach to offline reinforcement learning. *Advances in Neural Information Processing Systems* 36 (2024).

[32] Jianhao Wang, Zhizhou Ren, Beining Han, Jianing Ye, and Chongjie Zhang. 2021. Towards understanding cooperative multi-agent q-learning with value factorization. *Advances in Neural Information Processing Systems* 34 (2021), 29142–29155.

[33] Xiangsen Wang, Haoran Xu, Yinan Zheng, and Xianyuan Zhan. 2024. Offline multi-agent reinforcement learning with implicit global-to-local value regularization. *Advances in Neural Information Processing Systems* 36 (2024).

[34] Yihan Wang, Beining Han, Tonghan Wang, Heng Dong, and Chongjie Zhang. 2020. Dop: Off-policy multi-agent decomposed policy gradients. In *International conference on learning representations*.

[35] Jialong Wu, Haixu Wu, Zihan Qiu, Jianmin Wang, and Mingsheng Long. 2022. Supported policy optimization for offline reinforcement learning. *Advances in Neural Information Processing Systems* 35 (2022), 31278–31291.

[36] Yifan Wu, George Tucker, and Ofir Nachum. 2019. Behavior regularized offline reinforcement learning. *arXiv preprint arXiv:1911.11361* (2019).

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

[38] Jing Zhang, Chi Zhang, Wenjia Wang, and Bingyi Jing. 2023. Constrained policy optimization with explicit behavior density for offline reinforcement learning. *Advances in Neural Information Processing Systems* 36 (2023), 5616–5630.

[39] Tianhao Zhang, Yueheng Li, Chen Wang, Guangming Xie, and Zongqing Lu. 2021. Fop: Factorizing optimal joint policy of maximum-entropy multi-agent reinforcement learning. In *International conference on machine learning*. PMLR, 12491–12500.

[40] Zhengbang Zhu, Minghuan Liu, Liyuan Mao, Bingyi Kang, Minkai Xu, Yong Yu, Stefano Ermon, and Weinan Zhang. 2023. Madiff: Offline multi-agent learning with diffusion models. *arXiv preprint arXiv:2305.17330* (2023).