QSIM: Mitigating Overestimation in Multi-Agent Reinforcement Learning via Action Similarity Weighted Q-Learning

Yuanjun Li[1], **Bin Zhang**[2], **Hao Chen**[2], **Zhouyang Jiang**[1], **Dapeng Li**[3,4], **Zhiwei Xu**[1*]

[1]Shandong University
[2]The Key Laboratory of Cognition and Decision Intelligence for Complex Systems, Institute of Automation, Chinese Academy of Sciences
[3]Li Auto Inc.
[4]Tsinghua University
liyuanjun@mail.sdu.edu.cn, zhiwei_xu@sdu.edu.cn

Abstract

Value decomposition (VD) methods have achieved remarkable success in cooperative multi-agent reinforcement learning (MARL). However, their reliance on the max operator for temporal-difference (TD) target calculation leads to systematic Q-value overestimation. This issue is particularly severe in MARL due to the combinatorial explosion of the joint action space, which often results in unstable learning and suboptimal policies. To address this problem, we propose **QSIM**, a similarity weighted Q-learning framework that reconstructs the TD target using action similarity. Instead of using the greedy joint action directly, QSIM forms a similarity weighted expectation over a structured near-greedy joint action space. This formulation allows the target to integrate Q-values from diverse yet behaviorally related actions while assigning greater influence to those that are more similar to the greedy choice. By smoothing the target with structurally relevant alternatives, QSIM effectively mitigates overestimation and improves learning stability. Extensive experiments demonstrate that QSIM can be seamlessly integrated with various VD methods, consistently yielding superior performance and stability compared to the original algorithms. Furthermore, empirical analysis confirms that QSIM significantly mitigates the systematic value overestimation in MARL.

Code — https://github.com/MaoMaoLYJ/pymarl-qsim

Introduction

Cooperative Multi-Agent Reinforcement Learning (MARL) has emerged as a powerful framework for addressing complex group decision-making problems across domains such as robotic swarm control (Tang et al. 2023; Shi et al. 2025), autonomous driving coordination (Zhang et al. 2024; Zheng and Gu 2024), and network resource management (Stepanov et al. 2024; Chafii et al. 2023). To handle the inherent non-stationarity in multi-agent learning, the Centralized Training with Decentralized Execution (CTDE) paradigm has become the standard approach (Lowe et al. 2017; Sunehag et al. 2017). Within this framework, Value Decomposition (VD) methods (Rashid et al. 2020b) dominate by factorizing the joint action-value function into individual utilities, thereby effectively resolving the credit assignment problem. Despite their success, these methods suffer from a fundamental limitation: their reliance on the *max operator* when computing the Temporal-Difference (TD) target. It is well-known in single-agent reinforcement learning that maximizing over noisy value estimates leads to overestimation (Hasselt 2010), and this effect becomes even more pronounced in multi-agent settings (Ding et al. 2024). As illustrated in Figure 1, theoretical analysis suggests that the upper bound of the overestimation bias continuously increases with the number of agents N, leading to severe learning instability and convergence to suboptimal policies in large-scale scenarios. Existing VD methods typically overlook this systemic overestimation error by constructing the TD target based on a single, biased greedy joint action. Therefore, addressing this intrinsic overestimation issue is critical.

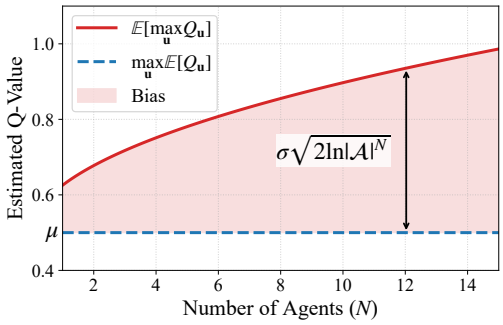

Figure 1: Theoretical curves showing how the upper bound of overestimation bias scales with the number of agents N. Further details are given in Theorem 1.

The standard practice in VD methods derives the TD target y directly from the Bellman optimality equation, namely $y = r + \gamma \max_{\boldsymbol{u}'} Q_{\text{tar}}(s', \boldsymbol{u}')$. These methods fundamentally rely on the Individual-Global-Max (IGM) principle (Son et al. 2019), which ensures that the greedy joint action $\boldsymbol{u}^*$ can be efficiently obtained through decentralized maximiza-

*Corresponding author

tion. While this design enables the method to comply with the CTDE paradigm, it remains limited by estimation noise in the learned action-value functions. Such noise induces the max operator to preferentially select actions with positively biased estimates (Thrun and Schwartz 2014). As a result, the calculated target $Q_{\text{tar}}(s', \boldsymbol{u}^*)$ becomes biased and systematically overestimates the true expected return. A common remedy is to replace the hard maximization with an expectation, following the intuition of Expected SARSA (Sutton, Barto et al. 1998). However, implementing such an expectation in MARL is highly impractical: the joint action space grows exponentially with the number of agents, making the expectation both computationally prohibitive and sensitive to errors from rarely sampled actions (Pan et al. 2021). This presents a critical dilemma in that the greedy target remains tractable yet biased, whereas computing a full expectation is theoretically sound but infeasible in practice.

To resolve this dilemma, we propose **QSIM**, a framework that mitigates overestimation via action similarity weighted Q-learning. QSIM reformulates the TD target by leveraging the local structure of the joint action space. Instead of relying on a single greedy point estimate or an intractable global expectation, QSIM operates within a constructed *near-greedy* joint action subspace. This subspace is composed of single-agent deviations from the greedy policy, characterizes the local topological neighborhood around the greedy action while maintaining linear computational complexity. Within this subspace, QSIM computes a weighted TD target, where the integration is governed by learned action similarity. We posit that actions leading to similar future transitions should share similar values. By assigning higher influence to actions that are semantically aligned with the greedy choice, QSIM effectively dampens the noise responsible for overestimation while preserving the optimality of the policy.

Our main contributions are summarized as follows:

- A self-supervised autoencoder is introduced, utilizing a feature encoder to effectively capture the semantic meaning of individual actions. The learned representation provides a reliable similarity metric that quantifies the relationship between joint actions.
- We propose a tractable near-greedy joint action space that scales linearly with the number of agents, upon which a similarity weighted TD target is computed. This design converts the high-variance greedy estimate into a more stable expectation, theoretically proven to constitute a lower bound on the standard greedy estimate, thereby guaranteeing the mitigation of overestimation bias.
- Extensive experiments on multi-agent benchmarks, including SMAC, MPE and Matrix Games, show that integrating QSIM into different value decomposition frameworks consistently improves both performance and training stability over the original methods, while substantially mitigating the overestimation inherent in Q-learning.

Related Work

Value-based MARL Following the CTDE paradigm, VDN (Sunehag et al. 2017) established the foundation of value decomposition by approximating the joint Q-value as a sum of individual utilities. QMIX (Rashid et al. 2020b) advanced this line of work by introducing a monotonic mixing network that guarantees consistency with the IGM principle. Subsequent research sought to alleviate the representational limitations of monotonic factorization. For example, WQMIX (Rashid et al. 2020a) applies a weighted objective that emphasizes higher-quality joint actions. QTRAN (Son et al. 2019) reformulates value factorization as a constrained optimization problem. In addition, QPLEX (Wang et al. 2020a) leverages a duplex dueling architecture to increase expressiveness within the IGM framework. Beyond discrete value-based methods, VMIX (Su, Adams, and Beling 2021) extends decomposition to actor-critic frameworks by applying monotonic mixing to a central critic. RIIT (Hu et al. 2023) further demonstrates that well-optimized monotonic baselines often remain competitive. Despite these advances, most prior works concentrate on improving the expressiveness of the value factorization itself. QSIM focuses on a complementary issue by enhancing the quality of the TD target and mitigating the systematic overestimation bias that emerges during training.

Mitigating Value Overestimation Applying the max operator to noisy estimates induces a positive bias (Thrun and Schwartz 2014), exacerbated in MARL by the combinatorial joint action space (Ding et al. 2024). Standard single-agent bias mitigation methods like Double Q-learning (Van Hasselt et al. 2016) or Softmax Bellman operators (Song, Parr, and Carin 2019) are often insufficient for multi-agent tasks due to the exponential complexity of the action space. Consequently, several MARL-specific methods have been developed. RES (Pan et al. 2021) computes the TD target via a regularized softmax average. λWD QMIX (Zhao et al. 2024) employs weighted double estimators to reduce bias. And Comix (Liu et al. 2025) constructs a Sandwich framework that constrains the value function between learnable upper and lower bounds. While QSIM adopts the similar near-greedy subspace as RES, it fundamentally differs in how the target is aggregated. Whereas RES relies on a softmax operator that can overemphasize overestimated Q-values, QSIM introduces an action similarity mechanism to guide the integration. This ensures that actions functionally aligned with the greedy policy contribute more to the target, effectively reducing noise from irrelevant actions.

Action Representation Learning A major challenge in MARL is the curse of dimensionality stemming from the exponentially growing joint action space. Action representation learning offers a promising solution by learning structured, low-dimensional embeddings that capture relationships among actions. In single-agent domains, prior work decompose policies via abstract embeddings (Chandak et al. 2019) or model dependencies using hypergraphs (Tavakoli, Fatemi, and Kormushev 2021). In multi-agent settings, ROMA (Wang et al. 2020b) uses latent representations to induce dynamic roles, and RODE (Wang et al. 2021) clusters actions to build a hierarchical decomposition of the joint space. Unlike these methods, which utilize representation learning primarily for policy decomposition or hierarchi-

cal control, QSIM employs it for calculating action similarity. This similarity then guides the value update, smoothing the TD target with semantically relevant neighboring actions and stabilizing learning.

In summary, QSIM integrates these insights to form a modular framework compatible with diverse VD methods, offering a novel perspective on robust value estimation in high-dimensional multi-agent systems.

Background

MARL Problem Definition

Cooperative MARL is typically formalized as a Decentralized Partially Observable Markov Decision Process (Dec-POMDP) (Oliehoek, Amato et al. 2016), defined by the tuple $\langle \mathcal{N}, \mathcal{S}, \mathcal{U}, \mathcal{T}, \mathcal{R}, \mathcal{O}, \gamma \rangle$. $\mathcal{N}$ is a finite set of N agents, and $\mathcal{S}$ is the global state space. Each agent $i \in \mathcal{N}$ draws actions from an individual action space $\mathcal{A}_i$. At each timestep t, the environment is in state $s^t \in \mathcal{S}$, and each agent receives a local observation $o_i^t \in \mathcal{O}_i$. Based on its action-observation history τ_i^t, each agent i selects an action $a_i^t \in \mathcal{A}_i$, forming a joint action $\boldsymbol{u}^t = (a_1^t, \ldots, a_N^t) \in \mathcal{U}$, where $\mathcal{U} = \prod_{i \in \mathcal{N}} \mathcal{A}_i$ represents the joint action space. The system then transitions to the next state s^{t+1} according to the state transition function $\mathcal{T}(s^{t+1}|s^t, \boldsymbol{u}^t)$ and yields a shared reward $r^t = \mathcal{R}(s^t, \boldsymbol{u}^t)$. The objective is to discover a joint policy $\boldsymbol{\pi}$ that maximizes the expected discounted return $J(\boldsymbol{\pi}) = \mathbb{E}_\pi[\sum_{t=0}^{\infty} \gamma^t r^t]$, where $\gamma \in (0, 1]$ is the discount factor.

This objective is approached by learning the optimal joint action-value function Q_{tot}^*. For a policy $\boldsymbol{\pi}$, the joint action-value function $Q_{\text{tot}}^{\boldsymbol{\pi}}$ is defined as the expected return following the joint action $\boldsymbol{u}^t$ conditioned on the joint history $\boldsymbol{\tau}^t$:

$$Q_{\text{tot}}^{\boldsymbol{\pi}}(\boldsymbol{\tau}^t, \boldsymbol{u}^t) = \mathbb{E}_{\boldsymbol{\pi}} \left[\sum_{k=t}^{\infty} \gamma^{k-t} r^k \mid \boldsymbol{\tau}^t, \boldsymbol{u}^t \right]. \tag{1}$$

The joint history $\boldsymbol{\tau}^t = (\tau_1^t, \ldots, \tau_N^t)$ comprises the individual histories $\tau_i^t = (o_i^1, a_i^1, \ldots, o_i^{t-1}, a_i^{t-1}, o_i^t)$, encapsulating all information available to agent i at timestep t.

Centralized Training with Decentralized Execution

The CTDE framework (Lowe et al. 2017) effectively blends the strengths of centralized and decentralized paradigms. This approach trains decentralized policies using global information during training. While CTDE algorithms have proven successful in many multi-agent problems (Mahajan et al. 2019; Li et al. 2025), they face scalability challenges as the joint action-observation space expands exponentially with the number of agents, potentially leading to inefficient learning in large-scale scenarios.

Value decomposition has emerged as a key approach within the CTDE framework. It employs a mixing network to compose individual utility functions $Q_i(\tau_i, a_i)$ into a factored joint action-value function $Q_{\text{tot}}(\boldsymbol{\tau}, \boldsymbol{u})$. Most value decomposition methods are built upon the IGM assumption, which states that the global optimum of the factored value function coincides with the set of local optima for the individual utility functions:

$$\arg\max_{\boldsymbol{u}} Q_{\text{tot}}(\boldsymbol{\tau}, \boldsymbol{u}) = \begin{pmatrix} \arg\max_{a_1} Q_1(\tau_1, a_1) \\ \arg\max_{a_2} Q_2(\tau_2, a_2) \\ \vdots \\ \arg\max_{a_N} Q_N(\tau_N, a_N) \end{pmatrix}. \tag{2}$$

This property is essential for enabling efficient decentralized execution. The networks are trained by minimizing the TD-error loss:

$$\mathcal{L}(\theta) = \mathbb{E}\left[\left(y - Q_{\text{tot}}(\boldsymbol{\tau}, \boldsymbol{u}; \theta) \right)^2 \right], \tag{3}$$

where the TD target y is defined as:

$$y = r + \gamma \max_{\boldsymbol{u'}} Q_{\text{tar}}(\boldsymbol{\tau'}, \boldsymbol{u'}; \theta^-). \tag{4}$$

$Q_{\text{tot}}(\boldsymbol{\tau}, \boldsymbol{u}; \theta)$ denotes the value produced by the main network, and the target value $Q_{\text{tar}}(\boldsymbol{\tau'}, \boldsymbol{u'}; \theta^-)$ is generated by a target network with frozen parameters θ^-. The target parameters are periodically updated from the main network to improve training stability (Mnih et al. 2015).

Overestimation in Q-learning

Maximization bias arises from applying the max operator to noisy value estimates. Let $\hat{Q}(s, a) = Q^*(s, a) + \epsilon_a$ be the learned approximation corrupted by zero-mean noise ϵ_a. The standard TD target $y = r + \gamma \max_{a'} \hat{Q}(s', a')$ introduces a positive bias due to Jensen's inequality:

$$\begin{aligned} \mathbb{E}[\max_{a'} \hat{Q}(s', a')] &= \mathbb{E}[\max_{a'} (Q^*(s', a') + \epsilon_{a'})] \\ &\geq \max_{a'} \mathbb{E}[Q^*(s', a') + \epsilon_{a'}] \\ &= \max_{a'} Q^*(s', a'). \end{aligned} \tag{5}$$

This inequality demonstrates that the target systematically overestimates the true maximum action value.

This bias becomes substantially more severe in MARL due to the combinatorial explosion of the joint action space. Consider a system with N agents, each possessing an action space of size $|\mathcal{A}|$, resulting in a joint action space size of $|\mathcal{U}| = |\mathcal{A}|^N$. To quantify the impact of this exponential expansion on value estimation, we analyze the bias under the standard assumption that estimation errors follow a Gaussian distribution (Smith and Winkler 2006).

Theorem 1. *Let the estimated joint Q-value be modeled as $\hat{Q}(\boldsymbol{u}) = Q^*(\boldsymbol{u}) + \epsilon_{\boldsymbol{u}}$ for any joint action $\boldsymbol{u} \in \mathcal{U}$, where $\epsilon_{\boldsymbol{u}} \sim \mathcal{N}(0, \sigma^2)$ denotes independent zero-mean Gaussian noise. The maximization bias is upper bounded as follows:*

$$\left(\mathbb{E}\left[\max_{\boldsymbol{u}} \hat{Q}(\boldsymbol{u}) \right] - \max_{\boldsymbol{u}} Q^*(\boldsymbol{u}) \right) \leq \sigma \sqrt{2 \ln |\mathcal{A}|^N}.$$

The theorem establishes that the upper bound of the overestimation bias grows proportionally to $\sqrt{N}$ (since $\sqrt{\ln |\mathcal{A}|^N} = \sqrt{N \ln |\mathcal{A}|}$). Figure 1 illustrates this theoretical trend. As the number of agents increases, the potential gap between the expected greedy estimate and the true value expands accordingly. This confirms that the exponential growth of the multi-agent action space significantly amplifies overestimation, ultimately weakening the decentralized policies trained under these biased central estimates.

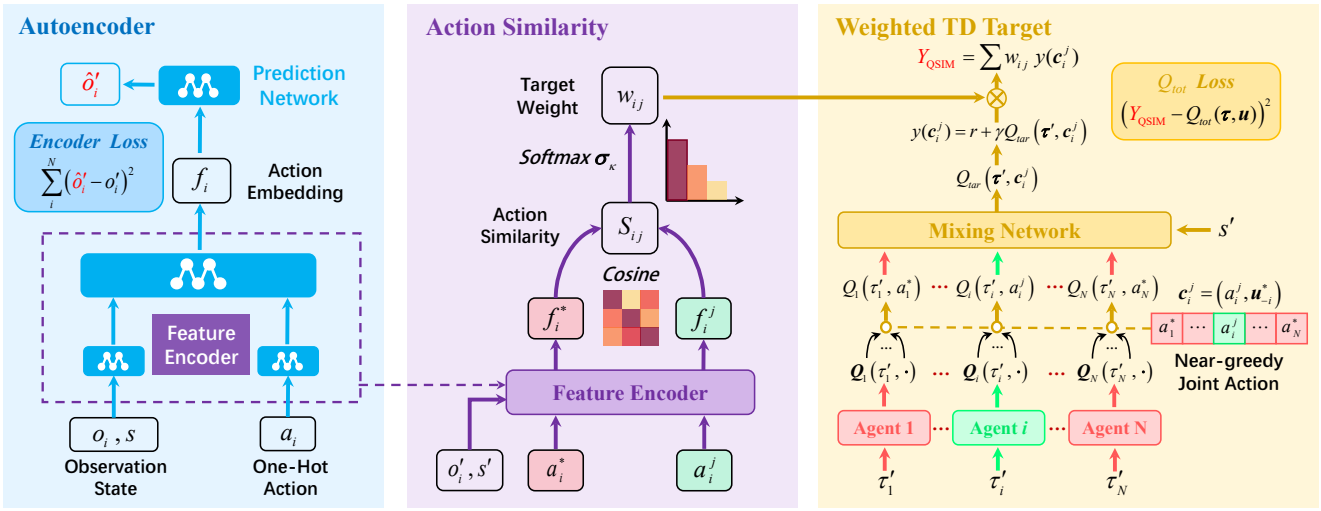

Figure 2: QSIM framework. **(a) Autoencoder:** Self-supervised learning of action representations. **(b) Action Similarity:** Computing cosine similarity between deviating action a_i^j and greedy action a_i^* to derive softmax-normalized weights. **(c) Weighted TD Target:** Constructing near-greedy joint actions $\boldsymbol{c}_i^j$ and aggregating their TD target into the final weighted TD target Y_{QSIM}.

QSIM Framework

To mitigate the maximization bias in VD methods, we introduce the **QSIM** framework. It replaces the high-variance greedy TD target with a robust, similarity weighted expectation over a tractable near-greedy joint action space. By transitioning from a biased point estimate to a smoothed expectation, QSIM dampens the noise responsible for overestimation, promoting stable and accurate value updates. The framework consists of three components: (1) a self-supervised autoencoder learning functional action representations; (2) an action similarity mechanism for deriving integration weights from learned action embeddings; (3) a similarity weighted aggregation scheme for the final TD target. The overall architecture is illustrated in Figure 2.

Learning Action Representations

To quantify semantic similarity between actions, we learn embeddings that describe how each action influences future observations. The key idea is that the meaning of an action is reflected in the state changes it causes. We employ a self-supervised autoencoder parameterized by ϕ, utilizing the next observation as a training signal. This module is visualized in Figure 2(a).

Formally, the architecture consists of a feature encoder E_ϕ and a predictive network P_ϕ. For each agent i, the encoder E_ϕ processes the local observation o_i, the global state s, and the chosen action a_i. As implemented, o_i and s are encoded in parallel with a_i before being merged to produce a low-dimensional state-action embedding f_i formulated as:

$$f_i = E_\phi(o_i, s, a_i). \tag{6}$$

To ensure these embeddings capture global transition dynamics, the embeddings from all agents $\boldsymbol{f} = (f_1, \ldots, f_N)$ are concatenated and passed to the prediction network P_ϕ. This network reconstructs the next joint local observation $\hat{\boldsymbol{o}}' = (\hat{o}'_1, \ldots, \hat{o}'_N)$:

$$\hat{\boldsymbol{o}}' = P_\phi(\boldsymbol{f}). \tag{7}$$

The autoencoder is optimized by minimizing the Mean Squared Error (MSE) between the predicted and actual next observations:

$$\mathcal{L}_{\text{AE}}(\phi) = \mathbb{E}\left[\sum_{i=1}^{N} (\hat{o}'_i - o'_i)^2\right]. \tag{8}$$

Minimizing this loss encourages each f_i to encode predictive information about the transition, yielding a reliable metric for action similarity. Empirically, this objective converges rapidly, enabling QSIM to utilize stable and informative similarity metrics from the early stages of training.

Near-greedy Joint Action Space

Traditional VD methods rely on a single greedy TD target. While computing an expectation over the full space $\mathcal{U}$ is theoretically appealing, it is computationally intractable and highly sensitive to estimation noise due to sparse visitation. To address these issues, we construct a tractable near-greedy joint action space $\mathcal{C}$ anchored on the greedy joint action, focusing the expectation on the most plausible region. Following the Double Q-learning, we first identify the greedy action $\boldsymbol{u}^*$ using the main network, which is efficiently computed via local greedy selections under the IGM principle. Given $\boldsymbol{\tau}'$, the greedy joint action $\boldsymbol{u}^*$ is defined as:

$$\boldsymbol{u}^* = \arg\max_{\boldsymbol{u} \in \mathcal{U}} Q_{\text{tot}}(\boldsymbol{\tau}', \boldsymbol{u}; \theta). \tag{9}$$

The near-greedy joint action space $\mathcal{C}$ is then formed by considering all possible single-agent deviating actions from

$\boldsymbol{u}^*$. Specifically, it is the union of joint actions where exactly one agent i deviates to an available action $a_i \in \mathcal{A}_i$, while all other agents $\mathcal{N}_{-i}$ maintain their greedy actions $\boldsymbol{u}^*_{-i}$:

$$\mathcal{C} = \bigcup_{i=1}^{N} \{(a_i, \boldsymbol{u}^*_{-i}) \mid a_i \in \mathcal{A}_i\}. \tag{10}$$

The size of the near-greedy action space grows linearly with the number of agents, $|\mathcal{C}| = N \times |\mathcal{A}|$, in sharp contrast to the exponential growth of the full joint action space $|\mathcal{U}| = |\mathcal{A}|^N$.

Finally, each near-greedy joint action $\boldsymbol{c} \in \mathcal{C}$ is evaluated to generate a TD target candidate $y(\boldsymbol{c})$:

$$y(\boldsymbol{c}) = r + \gamma Q_{\text{tar}}(\boldsymbol{\tau}', \boldsymbol{c}; \theta^-), \tag{11}$$

where $Q_{\text{tar}}(\boldsymbol{\tau}', \boldsymbol{c}; \theta^-)$ denotes the global joint action-value predicted by the target network. This process yields a diverse set of target candidates, which are subsequently aggregated as detailed in the next section.

Weighted Q-learning with Action Similarity

Upon constructing the near-greedy joint action space $\mathcal{C}$ and generating the set of TD target candidates, the final phase of QSIM is to synthesize these candidates into a single robust learning signal. This is achieved by computing a weighted expectation in which each weight reflects the functional similarity between a near-greedy joint action and the greedy joint action. This weighting scheme ensures that the value update is regularized by diverse actions, while giving higher importance to actions that are most functionally aligned with the greedy choice.

Similarity Calculation First, we leverage the feature encoder E_ϕ to quantify the functional semantics of actions in the next timestep. For each agent i, let a_i^* denote its component in the greedy joint action $\boldsymbol{u}^*$, and $a_i^j \in \mathcal{A}_i$ denote any available action. Conditioned on the next global state s' and local observation o_i', we compute the greedy action embedding f_i^* and the deviating action embedding f_i^j:

$$\begin{aligned} f_i^* &= E_\phi(o_i', s', a_i^*), \\ f_i^j &= E_\phi(o_i', s', a_i^j). \end{aligned} \tag{12}$$

The similarity score S_{ij} is then computed using cosine similarity between these embeddings:

$$S_{ij} = \text{Cosine}(f_i^*, f_i^j) = \frac{f_i^* \cdot f_i^j}{\|f_i^*\| \|f_i^j\|}. \tag{13}$$

To extend the learned individual action similarity metrics to the joint action space, the specific composition of the near-greedy space $\mathcal{C}$ is first formalized. A near-greedy joint action $\boldsymbol{c}_i^j \in \mathcal{C}$ denotes the configuration where agent i selects an deviating action $a_i^j \in \mathcal{A}_i$, while all other agents maintain their greedy actions $\boldsymbol{u}^*_{-i}$:

$$\boldsymbol{c}_i^j = (a_i^j, \boldsymbol{u}^*_{-i}). \tag{14}$$

Given this, the relationship between joint action similarity and individual action similarity is established as follows:

Definition 1 (Action Similarity). *The functional similarity between a near-greedy joint action $\boldsymbol{c}_i^j$ and the greedy joint action $\boldsymbol{u}^*$ is defined as the local action similarity of the deviating agent:*

$$\text{Sim}(\boldsymbol{c}_i^j, \boldsymbol{u}^*) \triangleq \text{Sim}(a_i^j, a_i^*) = S_{ij}. \tag{15}$$

This definition ensures that the similarity score is fully determined by the deviating action of the specific agent modifying its policy, as the behaviors of all other agents remain constant.

Similarity Weighted Aggregation We transform the raw similarity scores into a normalized probability distribution to derive the integration weights. This is achieved using a softmax function with an inverse temperature parameter $\kappa \geq 0$:

$$w_{ij} = \frac{\exp(\kappa \cdot S_{ij})}{\sum_{k=1}^{|\mathcal{A}_i|} \exp(\kappa \cdot S_{ik})}. \tag{16}$$

The hyperparameter κ controls the sharpness of the weight distribution. As $\kappa \to \infty$, the distribution becomes increasingly peaked, approximating a hard max operator in which the action most similar to the greedy choice dominates. Conversely, as $\kappa \to 0$, the distribution becomes uniform, yielding a uniform weight expectation over the near-greedy joint action space.

Finally, the QSIM weighted TD target Y_{QSIM} is computed as the weighted value of the target candidates over the near-greedy space $\mathcal{C}$. Let w_{ij} denote the weight associated with the near-greedy joint action $\boldsymbol{c}_i^j$. The final target is given by:

$$Y_{\text{QSIM}} = \sum_{\boldsymbol{c}_i^j \in \mathcal{C}} w_{ij} \cdot y(\boldsymbol{c}_i^j), \tag{17}$$

where $y(\boldsymbol{c}_i^j)$ is the candidate target defined in Eq. (11). This aggregated target Y_{QSIM} replaces the standard greedy target in the TD loss function provided in Eq. (3). By incorporating value estimates from the neighborhood of the greedy policy, the resulting update provides a principled mechanism for mitigating overestimation bias.

Theorem 2. *The value estimation component of the QSIM TD target, denoted as $V_{QSIM}(s')$, constitutes a lower bound on the standard greedy value estimate $V_{Greedy}(s')$. Formally, since $\sum_{\boldsymbol{c} \in \mathcal{C}} w(\boldsymbol{c}) = 1$ and $w(\boldsymbol{c}) \geq 0$, it holds that:*

$$V_{\text{QSIM}}(s') \leq V_{\text{Greedy}}(s').$$

This theorem demonstrates that the QSIM operator systematically reduces the value estimate compared to the greedy max operator.

By leveraging this similarity weighted expectation, QSIM effectively exploits the local structure of the value function around the greedy policy. Instead of relying on a single potentially unstable point estimate, it synthesizes a consensus target from functionally similar actions. This approach not only provides a strictly lower-variance learning signal to mitigate overestimation but also ensures that the value update remains grounded in the agents' semantic behavior.

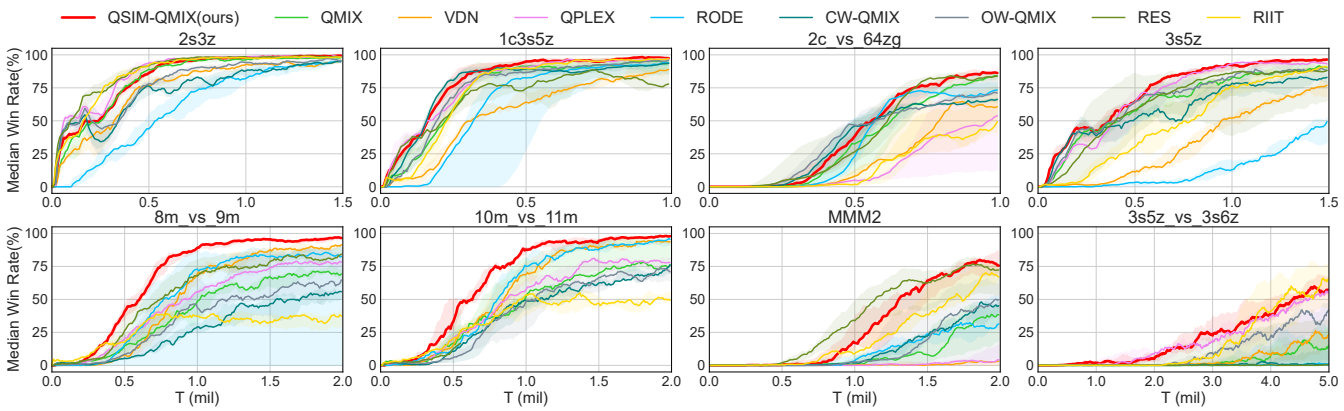

Figure 3: Performance comparison on SMAC maps.

Task	QSIM-QMIX	QMIX	VDN	QPLEX	RODE	CW-QMIX	OW-QMIX	RES	RIIT	Steps
2s3z	**99.4**(0.2)	97.9(0.7)	95.8(1.4)	98.8(0.6)	95.6(1.6)	94.8(2.6)	95.4(1.3)	98.3(0.5)	98.1(0.9)	1.5e6
1c3s5z	**97.5**(0.8)	93.8(4.1)	89.0(2.8)	96.5(1.9)	93.8(13.9)	93.5(3.2)	95.8(1.4)	77.9(28.6)	97.3(3.0)	1.0e6
2c_vs_64zg	**86.3**(2.0)	84.0(17.8)	60.6(19.0)	53.8(36.0)	73.3(27.5)	66.3(2.2)	71.0(15.6)	84.0(3.4)	49.4(15.4)	1.0e6
3s5z	**96.5**(1.1)	90.2(2.6)	76.9(7.6)	93.1(2.2)	49.4(15.2)	82.9(5.0)	88.3(2.4)	88.1(28.0)	90.8(3.6)	1.5e6
8m_vs_9m	**96.6**(2.2)	69.4(8.4)	91.5(**2.0**)	78.3(3.3)	82.5(43.4)	55.8(8.8)	64.8(17.0)	84.4(6.2)	37.3(9.3)	2.0e6
10m_vs_11m	**97.9**(1.4)	76.0(9.9)	93.8(2.9)	77.9(9.3)	95.6(2.3)	76.3(8.6)	70.8(7.6)	16.5(24.7)	49.4(7.8)	2.0e6
MMM2	**75.6**(8.4)	38.3(27.9)	3.1(19.8)	4.2(20.8)	31.5(24.2)	45.2(20.5)	49.8(31.3)	72.4(33.4)	67.1(10.1)	2.0e6
3s5z_vs_3s6z	57.1(**8.8**)	15.0(13.9)	23.1(21.8)	56.5(23.8)	1.5(20.7)	0.4(14.9)	41.9(24.0)	0.4(32.1)	**65.6**(22.9)	5.0e6

Table 1: Performance comparison on the SMAC benchmark, where results are reported as the final median test win rate (%) with standard deviation over 5 random seeds.

Experiments

We evaluate the proposed QSIM framework primarily using the QMIX backbone, denoted as **QSIM-QMIX**. Our empirical analysis addresses five key research questions:

1. Performance & Stability: Does QSIM-QMIX outperform existing VD baselines in both win rate and stability?
2. Generality: Does integrating QSIM with other VD backbones (VDN, QPLEX) consistently yield gains across diverse environments?
3. Ablation: How critical are the near-greedy action space and the similarity weighted aggregation scheme?
4. Overestimation: Does QSIM effectively mitigate the overestimation bias? We analyze the error between estimated $\hat{Q}_{\text{tot}}$ and actual discounted return.
5. Action Representation Visualization: Do the learned action embeddings capture meaningful action semantics?

Benchmarks and Baselines Our empirical evaluation spans four diverse benchmarks: SMAC (Samvelyan et al. 2019), MPE (Lowe et al. 2017) and Matrix Games (Papoudakis et al. 2020). Comparative analysis is conducted against a comprehensive suite of baselines, including VDN (Sunehag et al. 2017), QMIX (Rashid et al. 2020b), WQMIX (Rashid et al. 2020a), QPLEX (Wang et al. 2020a), RES (Pan et al. 2021), RODE (Wang et al. 2021), and RIIT (Hu et al. 2023).

Experimental Setup Experiments are conducted over 5 random seeds. Plots report median performance (solid line) with 25th–75th percentile interquartile ranges (shaded area), where smaller shaded areas indicate higher stability.

Performance on SMAC

We evaluate on SMAC, a benchmark requiring fine-grained micromanagement under partial observability. The combinatorial explosion in SMAC poses a severe challenge for value estimation. To ensure a comprehensive evaluation, we select eight maps spanning a spectrum of difficulty: two easy (*2s3z*, *1c3s5z*), four hard (*2c_vs_64zg*, *3s5z*, *8m_vs_9m*, *10m_vs_11m*), and two super-hard (*MMM2*, *3s5z_vs_3s6z*).

Figure 3 and Table 1 collectively illustrate the superior performance and stability of QSIM-QMIX. In high-dimensional scenarios where standard VD methods are prone to instability driven by maximization bias, QSIM-QMIX achieves higher mean win rates with significantly lower standard deviation.

Generality of QSIM

A key strength of QSIM lies in its modular architecture, which enables seamless integration into various VD methods. To empirically verify this generality, we extend our evaluation beyond QMIX, integrating QSIM with two other prominent VD backbones: VDN and QPLEX. We then compare the resulting variants (QSIM-VDN, QSIM-QMIX,

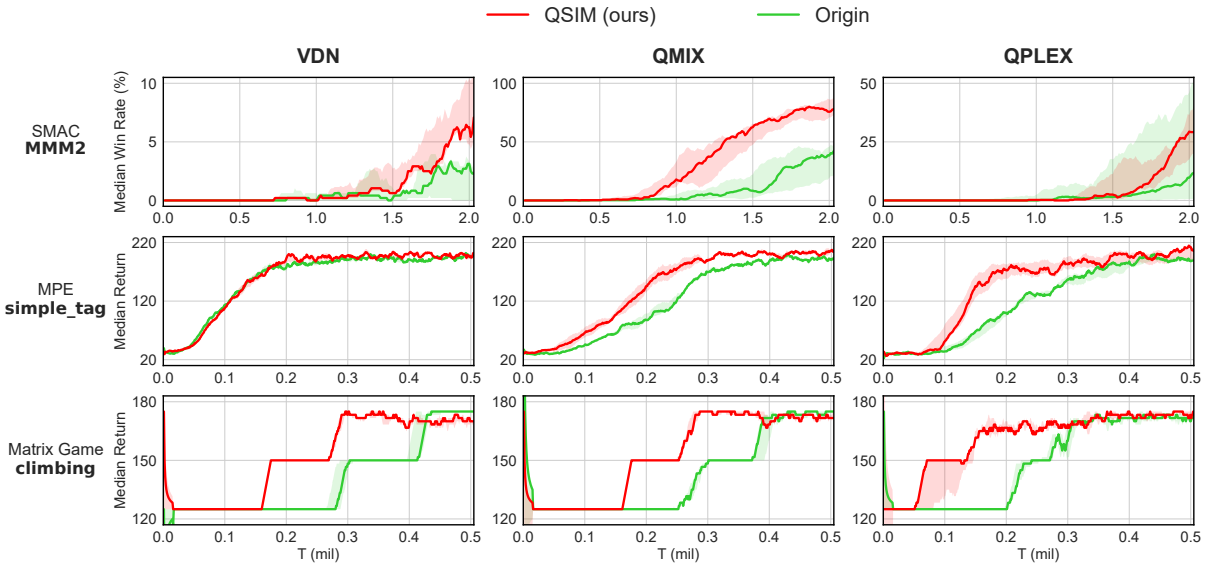

Figure 4: Comparison of QSIM-enhanced variants with their original baselines across different benchmarks.

QSIM-QPLEX) with their original counterparts across multiple environments.

Figure 4 summarizes the comparative results on three representative scenarios covering different domains. Across all tested environments, the QSIM-enhanced variants consistently outperform or match the original baselines. These findings demonstrate that QSIM is broadly compatible with existing value decomposition frameworks and reliably boosts their performance. By providing a more robust learning signal, QSIM improves the sample efficiency and stability of arbitrary value decomposition architectures, regardless of the task complexity.

Ablation Study

We conduct an ablation study to isolate the contributions of two core mechanisms, namely the construction of the near-greedy joint action space and the similarity weighted scheme. To this end, we introduce QSIM-Mean, a variant that retains the near-greedy space structure but replaces the similarity weight with a uniform weight $w(\boldsymbol{c}) = 1/|\mathcal{C}|$. This is equivalent to setting the inverse temperature $\kappa = 0$ in Eq. (16), thereby assigning the same weight to each near-greedy joint action $\boldsymbol{c}$.

As shown in Figure 5, the results across SMAC tasks reveal a clear performance ordering in which QSIM-QMIX outperforms QSIM-Mean, and both variants outperform QMIX. This observation provides two important insights.

First, the improvement of QSIM-Mean over QMIX highlights the structural benefit brought by the near-greedy action space. Since $\mathcal{C}$ consists of single-agent deviations, it captures the local neighborhood of the greedy policy. Aggregating over these near-greedy actions provides a more diverse and better-regularized learning signal. Compared with the single-point greedy estimate used by QMIX, this smoothed expectation is more robust and helps avoid premature convergence to suboptimal solutions.

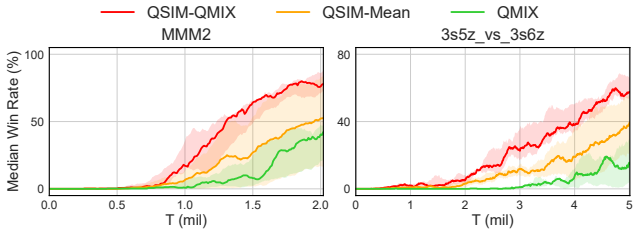

Figure 5: Ablation study comparing the full QSIM-QMIX model against the unweighted QSIM-Mean variant and the original QMIX baseline.

Second, the further gains achieved by QSIM-QMIX over QSIM-Mean underscore the importance of similarity weighting. While simple averaging reduces computational overhead, it naively aggregates all actions, inevitably including irrelevant or counter-productive deviating actions. By assigning higher weights to actions functionally similar to the greedy policy, QSIM effectively filters out noisy or implausible deviations. This selective aggregation yields a more informative target, enabling faster and more stable convergence.

Mitigation of Q-Value Overestimation

To evaluate QSIM's ability to mitigate the systematic Q-value overestimation in VD methods, we analyze the estimation error δ_q, defined as the error between the estimated joint Q-value $\hat{Q}_{\text{tot}}$ and the actual discounted return:

$$\delta_q = \hat{Q}_{\text{tot}}(\boldsymbol{\tau}_t, \boldsymbol{u}_t) - \sum_{k=t}^{\infty} \gamma^{k-t} r_k, \tag{18}$$

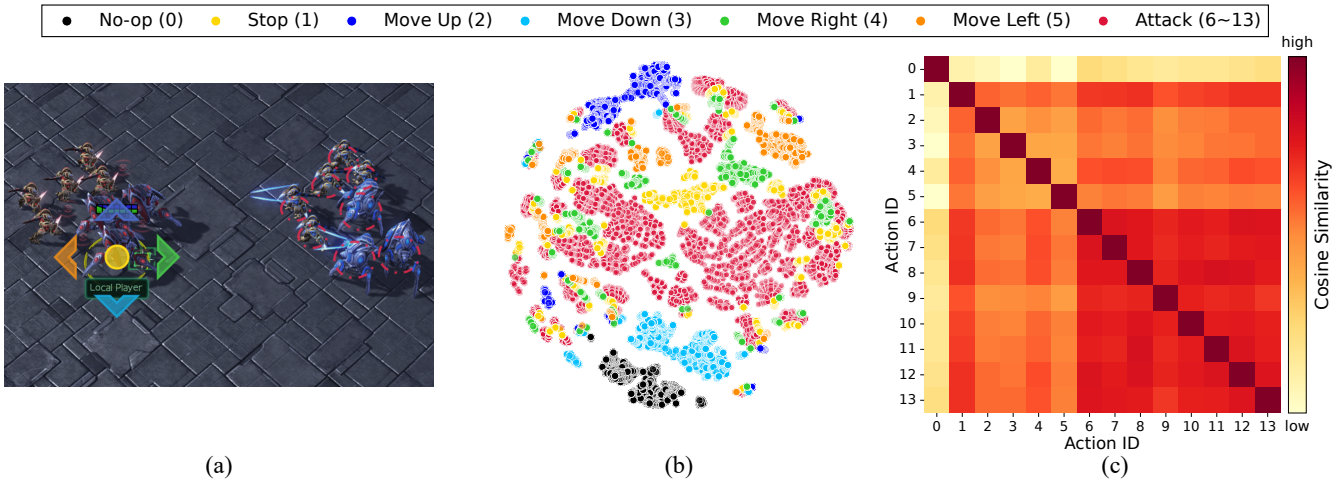

Figure 6: Visualization of learned action embeddings for a Stalker agent in the SMAC *3s5z* scenario. (a) The action space of a Stalker agent in the *3s5z* scenario. (b) A projection of the learned action embeddings produced via t-SNE. (c) Similarity matrix between all actions of the Stalker agent.

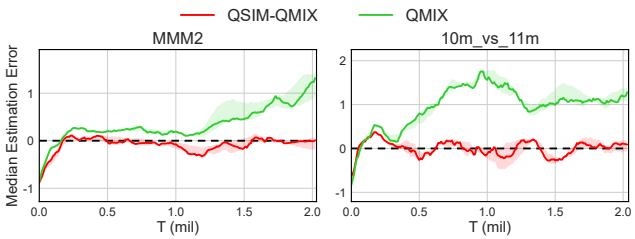

Figure 7: Comparison of Q-value estimation error δ_q on SMAC maps.

where the summation term approximates the actual joint Q-value Q_{tot}^{π} in Eq. (1). We compare the resulting estimation error δ_q of QSIM-QMIX with that of QMIX on several challenging SMAC maps by examining how the error evolves over training steps.

As illustrated in Figure 7, QSIM consistently exhibits lower estimation error δ_q compared with QMIX. The baseline QMIX suffers from a substantial accumulation of positive bias, stemming directly from its reliance on the standard max operator. In contrast, QSIM mitigates this bias by substituting the greedy target with a similarity weighted target. Consequently, the learned value estimates align more closely with the true returns, demonstrating that QSIM effectively mitigates overestimation and provides more accurate value estimates and a stable critic for VD methods.

Visualization of Learned Action Representations

To assess whether the QSIM's feature encoder E_ϕ learns action embeddings that capture meaningful action semantics, we visualize the representation space for a Stalker agent in SMAC *3s5z* scenario, as shown in Figure 6. The t-SNE projection (Maaten and Hinton 2008) in Figure 6(b) reveals that actions with similar functional effects form coherent clusters. For instance, "Attack" actions are grouped tightly, distinct from "Move" and "No-op". The similarity heatmap in Figure 6(c) further supports this structure, showing high similarity scores among functionally related actions, with "Attack" actions displaying strong pairwise affinity highlighted in red. These results confirm that the learned action embeddings provide a reliable and interpretable metric for our similarity-based weighting scheme.

Conclusion

In this paper, we proposed QSIM to mitigate the systematic Q-value overestimation in VD methods. By replacing the greedy TD target in Bellman optimality equation with a similarity weighted TD target aggregated over a constructed near-greedy joint action space, QSIM effectively mitigates the maximization bias inherent in Q-learning. Extensive empirical evaluations on SMAC, MPE, and Matrix Games demonstrate that QSIM serves as a generalized module compatible with various VD methods. The results confirm that QSIM consistently improves the performance and stability of existing algorithms while significantly reducing the Q-value estimation error.

For future work, we plan to extend the paradigm of similarity weighted expectation to a broader range of multi-agent domains. Promising directions include integrating the QSIM mechanism into multi-agent actor-critic algorithms or adapting it for offline reinforcement learning settings, where robust value estimation is equally critical for learning stable policies from fixed datasets.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant No. 62506210).

References

Chafii, M.; Naoumi, S.; Alami, R.; Almazrouei, E.; Bennis, M.; and Debbah, M. 2023. Emergent communication in multi-agent reinforcement learning for future wireless networks. *IEEE Internet of Things Magazine*, 6(4): 18–24.

Chandak, Y.; Theocharous, G.; Kostas, J.; Jordan, S.; and Thomas, P. 2019. Learning action representations for reinforcement learning. In *International conference on machine learning*, 941–950. PMLR.

Ding, L.; Du, W.; Zhang, J.; Guo, L.; Zhang, C.; Jin, D.; and Ding, S. 2024. Better value estimation in Q-learning-based multi-agent reinforcement learning. *Soft Computing*, 28(6): 5625–5638.

Hasselt, H. 2010. Double Q-learning. *Advances in neural information processing systems*, 23.

Hu, J.; Wang, S.; Jiang, S.; and Wang, W. 2023. Rethinking the Implementation Tricks and Monotonicity Constraint in Cooperative Multi-agent Reinforcement Learning. In *The Second Blogpost Track at ICLR 2023*.

Li, D.; Lou, N.; Xu, Z.; Zhang, B.; and Fan, G. 2025. Efficient Communication in Multi-Agent Reinforcement Learning with Implicit Consensus Generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 23240–23248.

Liu, K.; Zhang, T.; Xu, X.; and Zhao, Y. 2025. Counterfactual value decomposition for cooperative multi-agent reinforcement learning. *Neural Networks*, 190: 107692.

Lowe, R.; Wu, Y. I.; Tamar, A.; Harb, J.; Pieter Abbeel, O.; and Mordatch, I. 2017. Multi-agent actor-critic for mixed cooperative-competitive environments. *Advances in neural information processing systems*, 30.

Maaten, L. v. d.; and Hinton, G. 2008. Visualizing data using t-SNE. *Journal of machine learning research*, 9(Nov): 2579–2605.

Mahajan, A.; Rashid, T.; Samvelyan, M.; and Whiteson, S. 2019. Maven: Multi-agent variational exploration. *Advances in neural information processing systems*, 32.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, M. G.; Graves, A.; Riedmiller, M.; Fidjeland, A. K.; Ostrovski, G.; et al. 2015. Human-level control through deep reinforcement learning. *nature*, 518(7540): 529–533.

Oliehoek, F. A.; Amato, C.; et al. 2016. *A concise introduction to decentralized POMDPs*, volume 1. Springer.

Pan, L.; Rashid, T.; Peng, B.; Huang, L.; and Whiteson, S. 2021. Regularized softmax deep multi-agent q-learning. *Advances in Neural Information Processing Systems*, 34: 1365–1377.

Papoudakis, G.; Christianos, F.; Schäfer, L.; and Albrecht, S. V. 2020. Benchmarking multi-agent deep reinforcement learning algorithms in cooperative tasks. *arXiv preprint arXiv:2006.07869*.

Rashid, T.; Farquhar, G.; Peng, B.; and Whiteson, S. 2020a. Weighted qmix: Expanding monotonic value function factorisation for deep multi-agent reinforcement learning. *Advances in neural information processing systems*, 33: 10199–10210.

Rashid, T.; Samvelyan, M.; De Witt, C. S.; Farquhar, G.; Foerster, J.; and Whiteson, S. 2020b. Monotonic value function factorisation for deep multi-agent reinforcement learning. *Journal of Machine Learning Research*, 21(178): 1–51.

Samvelyan, M.; Rashid, T.; De Witt, C. S.; Farquhar, G.; Nardelli, N.; Rudner, T. G.; Hung, C.-M.; Torr, P. H.; Foerster, J.; and Whiteson, S. 2019. The starcraft multi-agent challenge. *arXiv preprint arXiv:1902.04043*.

Shi, R.; Yu, X.; Wang, Y.; Tian, Y.; Liu, Z.; Wu, W.; Zhang, X.-P.; and Veloso, M. M. 2025. Symmetry-Informed MARL: A Decentralized and Cooperative UAV Swarm Control Approach for Communication Coverage. *IEEE Transactions on Mobile Computing*.

Smith, J. E.; and Winkler, R. L. 2006. The optimizer's curse: Skepticism and postdecision surprise in decision analysis. *Management Science*, 52(3): 311–322.

Son, K.; Kim, D.; Kang, W. J.; Hostallero, D. E.; and Yi, Y. 2019. Qtran: Learning to factorize with transformation for cooperative multi-agent reinforcement learning. In *International conference on machine learning*, 5887–5896. PMLR.

Song, Z.; Parr, R.; and Carin, L. 2019. Revisiting the softmax bellman operator: New benefits and new perspective. In *International conference on machine learning*, 5916–5925. PMLR.

Stepanov, E.; Smeliansky, R.; Plakunov, A.; Borisov, A.; Zhu, X.; Pei, J.; and Yao, Z. 2024. On fair traffic allocation and efficient utilization of network resources based on MARL. *Computer Networks*, 250: 110540.

Su, J.; Adams, S.; and Beling, P. 2021. Value-decomposition multi-agent actor-critics. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, 11352–11360.

Sunehag, P.; Lever, G.; Gruslys, A.; Czarnecki, W. M.; Zambaldi, V.; Jaderberg, M.; Lanctot, M.; Sonnerat, N.; Leibo, J. Z.; Tuyls, K.; et al. 2017. Value-decomposition networks for cooperative multi-agent learning. *arXiv preprint arXiv:1706.05296*.

Sutton, R. S.; Barto, A. G.; et al. 1998. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge.

Tang, H.; Zhang, H.; Shi, Z.; Chen, X.; Ding, W.; and Zhang, X.-P. 2023. Autonomous swarm robot coordination via mean-field control embedding multi-agent reinforcement learning. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 8820–8826. IEEE.

Tavakoli, A.; Fatemi, M.; and Kormushev, P. 2021. Learning to Represent Action Values as a Hypergraph on the Action Vertices. In *International Conference on Learning Representations (ICLR 2021)*.

Thrun, S.; and Schwartz, A. 2014. Issues in using function approximation for reinforcement learning. In *Proceedings of the 1993 connectionist models summer school*, 255–263. Psychology Press.

Van Hasselt, H.; Guez, A.; Silver, D.; et al. 2016. Deep reinforcement learning with double q-learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30.

Wang, J.; Ren, Z.; Liu, T.; Yu, Y.; and Zhang, C. 2020a. Qplex: Duplex dueling multi-agent q-learning. *arXiv preprint arXiv:2008.01062*.

Wang, T.; Dong, H.; Lesser, V.; and Zhang, C. 2020b. Roma: Multi-agent reinforcement learning with emergent roles. *arXiv preprint arXiv:2003.08039*.

Wang, T.; Gupta, T.; Mahajan, A.; Peng, B.; Whiteson, S.; and Zhang, C. 2021. RODE: LEARNING ROLES TO DECOMPOSE MULTI-AGENT TASKS. In *9th International Conference on Learning Representations, ICLR 2021*.

Zhang, R.; Hou, J.; Walter, F.; Gu, S.; Guan, J.; Röhrbein, F.; Du, Y.; Cai, P.; Chen, G.; and Knoll, A. 2024. Multi-agent reinforcement learning for autonomous driving: A survey. *arXiv preprint arXiv:2408.09675*.

Zhao, L.-y.; Chang, T.-q.; Guo, L.-b.; Zhang, J.; Zhang, L.; and Ma, J.-d. 2024. An overestimation reduction method based on the multi-step weighted double estimation using value-decomposition multi-agent reinforcement learning. *Neural Processing Letters*, 56(3): 152.

Zheng, Z.; and Gu, S. 2024. Safe multi-agent reinforcement learning with bilevel optimization in autonomous driving. *IEEE Transactions on Artificial Intelligence*.