FINITE-TIME PARETO CONVERGENCE OF DECENTRALIZED MULTI-POLICY GRADIENT WITH PERIODIC REUSE FOR MULTI-AGENT MULTI-OBJECTIVE REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Decentralized multi-agent policy gradient methods enable collaborative decision optimization across multiple agents without a central coordinator. However, most existing theoretical studies on decentralized policy gradients are restricted to single-objective settings, while real-world complex tasks typically require each agent to simultaneously optimize multiple conflicting objectives. To the best of our knowledge, the theoretical foundations of decentralized multi-objective policy gradient algorithms remain largely underexplored. To tackle this problem, we propose D-PRMPG, a decentralized periodic-reuse multi-policy gradient algorithm. By exploiting the reusability of dynamic multi-objective preference weights, our method substantially reduces the excessive computational overhead caused by per-iteration dynamic weight computation. We further present a rigorous convergence analysis, demonstrating that D-PRMPG attains an $\mathcal{O}(1/\sqrt{T})$ convergence rate under both general convex and nonconvex objective functions, where T denotes the total number of iterations. Finally, empirical evaluations in multi-agent multi-objective reinforcement learning environments validate the superiority of the proposed algorithm.

1 INTRODUCTION

Multi-agent reinforcement learning (MARL) are widely deployed in real-world engineering scenarios, including drone coordination (Sun et al., 2025), intelligent transportation networks (Chu et al., 2019), and distributed power grids with time-varying topologies (Chen et al., 2022). Conventional MARL algorithms (Lowe et al., 2017; Rashid et al., 2018; Sun et al., 2021; Xu et al., 2023; Hou et al., 2026) are predominantly built upon the centralized-training decentralized-execution (CTDE) paradigm. However, this framework still relies on global information, and in practical scenarios this approach is limited, including wireless-network communication bottlenecsk (Feng et al., 2024), privacy constraints (Kasyap & Tripathy, 2021), and robustness challenges (Lee et al., 2026). To overcome these limitations, decentralized reinforcement learning (Sun et al., 2020; Wang et al., 2020; Zhu et al., 2024) employs fully decentralized training, in which agents update their policies based solely on local observations and communication within a limited neighborhood. In this framework, local value estimation biases in value-function-based methods accumulate over iterations, causing substantial errors in gradient estimates (Doan et al., 2021). In contrast, policy gradient (PG) (Sutton et al., 1999) methods directly parameterize the policy and iteratively optimize it along the gradient of the expected cumulative reward, eliminating the need for a globally consistent value function. Thanks to this advantage, decentralized policy gradient methods have become the mainstream paradigm in decentralized MARL. Recent research shows that decentralized natural policy gradient methods can achieve the optimal convergence rate within the class of decentralized policy gradient algorithms and exhibit linear speedup relative to centralized benchmarks Chen et al. (2024), while theoretical analyses have established both optimal convergence rates and linear acceleration properties under standard assumptions (Zhao et al., 2025). Based on these theoretical breakthroughs, the Decentralized policy gradient algorithm has gradually been validated in a wide range of practical applications (Li et al., 2025; Covone et al., 2025).

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

In contrast, the studies above largely focus on single-objective tasks, which limits their applicability to real-world multi-objective problems. For example, an autonomous vehicle (Lu et al., 2019; XinyuHuang et al., 2020) must simultaneously optimize traffic efficiency, passenger comfort, and driving safety. Because speed affects these objectives in opposite directions, no single policy dominates, different preference weightings thus yield distinct Pareto-optimal solutions (Navon et al., 2020). To address this issue, the policy gradient optimization method reduces conflicts between objectives by computing dynamic weights, thereby identifying a common descent direction (Chen et al., 2023; Xiao et al., 2023). Following this line of thinking, many policy gradient methods attempt to obtain a common descent direction through various approaches. For example, ARNPG Zhou et al. (2022) regularizes the natural policy gradient via anchor variation to dynamically adjust the gradient direction; LPPG-RL (Qiu et al., 2026) projects the policy gradient in lexicographical order of priority. However, these methods were originally designed for single-agent scenarios. In the field of multi-agent multi-objective reinforcement learning, existing research has primarily focused on the design and engineering applications (Wang et al., 2025) and networked algorithms (Hu et al., 2023) within the CTDE framework. There is a lack of theoretical analysis of decentralized multi-objective policy gradient algorithms. Specifically, three core challenges remain unaddressed in the decentralized multi-objective policy gradient methods:

(1) Conventional decentralized policy gradient methods only output a single policy under a given preference weighting, failing to fully cover the entire Pareto optimal front (Ambadkar et al., 2026).

(2) For handling large-scale high-dimensional optimization tasks, these algorithms require solving many additional subproblems to find a common descent direction across all objectives, which notably prolongs the training process (Kurin et al., 2022; Liu et al., 2023).

(3) The convergence properties of decentralized multi-objective policy gradient algorithms have not yet been rigorously established, especially under the coupled constraints of decentralized execution and multi-objective trade-offs. This naturally raises the following core questions:

Can we design a decentralized policy gradient algorithm for multi-agent multi-objective reinforcement learning that achieves theoretical convergence guarantees and computational efficiency ?

To answer the above questions, we have proposed D-PRMPG, a decentralized periodic reuse multi-policy gradient algorithm. The algorithm is based on the multi-policy gradient framework, it can effectively approximate the Pareto-optimal frontier, rather than providing a single solution under fixed preferences, as is the case with traditional single-policy methods. To alleviate the computational overhead in decentralized scenarios, we reduce this overhead by periodically calculating dynamic weights (Xu et al., 2025). We also provide a rigorous convergence analysis under both convex and nonconvex settings, filling a theoretical gap in the study of decentralized multi-objective policy gradients. In addition, a comprehensive review of related work is provided in Appendix A.1. Overall, the contributions of this paper are as follows:

- We propose D-PRMPG, a decentralized multi-objective multi-policy gradient algorithm with periodic reuse. By periodically updating dynamic weights and adopting batch gradient updates, it substantially reduces the frequency of weight optimization subproblem solving per iteration, effectively alleviating the computational overhead incurred by the decentralized multi-policy framework.
- We prove that under general convex conditions, D-PRMPG converges to the global Pareto-optimal solution at a rate of $\mathcal{O}(1/\sqrt{T})$, and under nonconvex conditions, it converges to Pareto-stationary point.
- We further conduct comparative evaluations on different multi-agent multi-objective benchmarks under different topologies and reuse periods. Results show that D-PRMPG reduces the number of λ computations while achieved better performance.

2 PRELIMINARIES

This section first introduces the background knowledge on Markov decision processes, policy gradients, and multi-objective optimization, laying the theoretical foundation for subsequent algorithm design and analysis.

2.1 MARKOV DECISION PROCESS

A standard Markov Decision Process (MDP) can be fully characterized by a five-tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r, \gamma)$. Specifically, $\mathcal{S}$ denotes the state space of the environment, at any time step t, the state of the agent is denoted as $s_t \in \mathcal{S}$. $\mathcal{A}$ represents the action space, and $a \in \mathcal{A}$ denotes an executable action in state s. $P(s'|s, a)$ is the state transition probability function, representing the probability that the environment transitions to the next state s' after executing action a in state s. $r(s, a)$ is the reward function, denoting the immediate reward obtained after executing action a in state s. $\gamma \in (0, 1)$ is the discount factor, which is used to balance immediate rewards and long-term returns. In the MDP framework, the decision-making rule of the agent is represented by a policy π, where $\pi(a|s)$ denotes the probability of selecting action a in state s.

To quantitatively evaluate the long-term performance of a policy π, the MDP defines three core value functions from different dimensions to characterize the cumulative discounted return of the policy under an infinite horizon. Among them, the state value function $V^\pi(s)$ represents the expected cumulative discounted reward obtained by an agent starting from state s, consistently following policy π, and interacting with the environment, is denfined as

$$V^\pi(s) := \mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{\infty}\gamma^t r(s_t, a_t) \mid s_0 = s\right]. \tag{1}$$

This expectation is computed based on trajectories $\tau = \{s_0, a_0, s_1, a_1, \dots\}$ sampled by the policy. Assuming the reward function is bounded, $r(s, a) \in [0, 1]$, it can be derived from the properties of geometric series that the state value function is also bounded, satisfying $0 \le V^\pi(s) \le \frac{1}{1-\gamma}$. If the initial state follows a distribution ρ, the overall expected performance of the policy can be defined as $J(\pi) = \mathbb{E}_{s_0\sim\rho}[V^\pi(s_0)]$. Furthermore, to evaluate the long-term value of initially executing a specific action in a given state, defines the action value function $Q^\pi(s, a)$. It represents the expected cumulative discounted reward obtained after initially executing a specific action a in state s, and then following policy π to interact with the environment, defined as

$$Q^\pi(s, a) := \mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{\infty}\gamma^t r(s_t, a_t) \mid s_0 = s, a_0 = a\right]. \tag{2}$$

$Q^\pi(s, a)$ directly reflects the long-term value derived from a specific state or action. However, relying solely on the action value makes it difficult to intuitively measure the relative merits of an action compared to the average performance of the current policy. To this end, the advantage function $A^\pi(s, a)$ is introduced, which is defined as the difference between the action value function and the state value function, $A^\pi(s, a) := Q^\pi(s, a) - V^\pi(s)$. It characterizes the relative advantage of selecting action a in state s compared to the average performance of following the current policy π. Based on the above value function framework, the core objective of reinforcement learning boils down to finding the optimal policy π^* that maximizes the expected cumulative reward under the initial state distribution, expressed as

$$\pi^* = \arg\max_\pi \mathbb{E}_{s_0\sim\rho}\left[V^\pi(s_0)\right]. \tag{3}$$

2.2 POLICY GRADIENT METHODS

To solve the minimization problem in Eq. (3), the policy gradient (PG) method directly finds the optimal policy by optimizing the parameterized policy π_θ, where π_θ is the parameterized form of the policy in terms of policy parameters θ, $\theta \in \Theta$, and $\Theta \in \mathbb{R}^d$ represents the feasible set of parameters. This more direct approach is more advantageous in high-dimensional and continuous action spaces. The core objective of PG methods is to maximize the expected cumulative discounted reward $J(\theta) = \mathbb{E}_{s_0\sim\mu}\left[v^{\pi_\theta}(s_0)\right]$, where μ is the initial state distribution. The policy gradient can be calculated by the following fundamental relation (Sutton & Barto, 2018):

$$\nabla_\theta v^{\pi_\theta}(s_0) = \frac{1}{1-\gamma}\mathbb{E}_{s\sim d_{s_0}^{\pi_\theta}}\mathbb{E}_{a\sim\pi_\theta(a|s)}\left[\nabla_\theta \log \pi_\theta(a|s)\, q^{\pi_\theta}(s, a)\right], \tag{4}$$

where $d_{s_0}^{\pi_\theta}(s)$ is the discounted state-visitation distribution induced by π_θ, defined as

$$d_{s_0}^{\pi_\theta}(s) = (1-\gamma)\sum_{t=0}^{\infty}\gamma^t \Pr\left(s_t = s \mid s_0, \pi_\theta\right). \tag{5}$$

Here, $\Pr(s_t = s \mid s_0, \pi_\theta)$ denotes the state-visitation probability of a policy π_θ at time step t. The factor $(1-\gamma)$ acts is a normalization constant. Under the initial state distribution, the discounted state-visitation is given by $d_\mu^{\pi_\theta}(s) = \mathbb{E}_{s_0\sim\mu}\left[d_{s_0}^{\pi_\theta}(s)\right]$. Eq. (4) shows that the parameters are updated in a direction that increases the logarithmic probability of actions yielding returns above the average. Along the policy gradient, the policy parameters θ_t are updated at time t through the following iteration:

$$\pi_{\theta_{t+1}} = \mathcal{P}_\Theta\left[\pi_{\theta_t} - \alpha_t \nabla_\theta v^{\pi_{\theta_t}}(\mu)\right]. \tag{6}$$

Here, α_t is the step size, and $\mathcal{P}_\Theta[\cdot]$ is the projection operator on Θ, which ensures that the updated policy always remains within the feasible set. Therefore, the optimal policy can be found using the iterative Eq. (6).

2.3 Multi Objective Optimization

To solve a multi-objective decision-making problem, our goal is to find a policy π that maximizes m objective functions. This problem can be formulated as

$$\max_{\pi(s)} F\big(\pi(s)\big) := \max_{\pi(s)} \big[f_1\big(\pi(s)\big),\ f_2\big(\pi(s)\big), \ldots,\ f_m\big(\pi(s)\big)\big]. \tag{7}$$

Since objectives often conflict with each other, there is generally no single optimal policy that simultaneously maximizes all objectives. For this reason, the Pareto optimality criterion is typically introduced. If for all objectives $m \in \{1,\ldots,M\}$, $f_m(\pi_1(s)) \geq f_m(\pi_2(s))$ holds, and there exists at least one n such that $f_n(\pi_1(s)) \geq f_n(\pi_2(s))$, then policy π_1 is said to dominate π_2. Since finding a Pareto-optimal solution is typically an NP-hard problem, in practice, researchers often instead seek solutions that satisfy Pareto stationarity, which is a necessary condition for Pareto optimality. If for a solution θ, there is no common downward direction d such that all objectives $m \in M$ satisfy $\nabla f_s(\theta)^\top d < 0$, then θ is said to be a Pareto-stable point. If such a direction exists, it is possible to simultaneously minimize the values of all objective functions along that direction. To find a Pareto-stable point, the Multi-Gradient Descent Algorithm (MGDA) (Agarwal et al., 2021) constructs a common descent direction by finding the optimal weights for the gradients. Its objective function is

$$\lambda_*(\theta) = \arg\min_{\lambda} \left\|\lambda^\top \nabla F(\theta)\right\|^2. \tag{8}$$

Among them $\nabla F(\theta) := \left\{\nabla f_s(\theta),\ s \in [S]\right\}$. Based on Eq. (8) optimal weights, the common descent direction is $d = \lambda_*^\top \nabla F(\theta)$. The parameters are updated iteratively via $\theta \leftarrow \theta - \eta d$ until a Pareto-stable point is reached.

3 Algorithm Design

This section formulates the decentralized multi-objective decision-making problem and proposes a corresponding periodic multi-policy gradient algorithm under a decentralized framework.

3.1 Decentralized Multi Objective Markov Decision Processes

This study considers a multi-agent system consisting of N agents and M potential objectives, where $i \in \{1,\ldots,N\} \triangleq [N]$ and $m \in \{1,\ldots,M\} \triangleq [M]$. All agents collaboratively compute value functions in the same environment. The interaction relationships among agents are defined by a communication graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where the set of nodes is $\mathcal{V} = \{1, 2, \ldots, N\}$, and the set of edges is $\mathcal{E}$. At time t, agent i contributes information through its neighborhood set $\mathcal{N}_i$, where $\mathcal{N}_i(t) = \{j | (i,j) \in \mathcal{E}_t, i, j \in \mathcal{V}\}$. The above multi-agent system can be formalized as a 6-tuple $(\mathcal{S}, \{\mathcal{A}_i\}_{i=1}^N, P, \{r_{i,1:M}\}_{i=1}^N, \gamma, \mathcal{G})$. $\mathcal{S}$ and $\mathcal{A}$ denote, respectively, the finite state space and action space of agent i, where $\mathcal{S} = \{s_1, s_2, \ldots, s_{|\mathcal{S}|}\}$, $\mathcal{A} = \{a_1, a_2, \ldots, a_{|\mathcal{A}|}\}$. The joint action space of all agents is denoted as $\mathcal{A} = \mathcal{A}_1 \times \mathcal{A}_2 \times \cdots \times \mathcal{A}_n$, and each agent selects an action based on its local policy π. After agent i executes an action, the reward it receives for the m objectives is $r_{i,m}$, and the corresponding M-dimensional reward vector is denoted as $r_{i,1:M} = (r_{i,1}, \ldots, r_{i,M})$. We assume that states and actions are globally observable. As with the parameterized form of the policy defined in Section 2, $\pi_{\theta_i}(a_i|s) = \theta_i(s, a_i)$, where $\theta_i(s, a_i) \in \Delta(\mathcal{A}_i)^{|S|}$ is a policy parameter, and $\sum_{a_i \in A_i} \theta_i(s, a_i) = 1$ holds for all agents i. $|\mathcal{S}|$ denotes the cardinality of the state space. Therefore,

the joint policy $\pi_\theta(a|s)$ is obtained according to the expression $\pi_\theta(a|s) = \prod_{i\in[M]} \pi_{\theta_i}(a_i|s)$, where $\theta = (\theta_1^\top, \theta_2^\top, \dots, \theta_N^\top)^\top$. For each goal $m \in [M]$ of agent $i \in [N]$, its value function $v_{i,m}^\theta(s)$ is defined as

$$v_{i,m}^\theta(s) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_{i,m}(s_t, a_{i,t}) \mid \theta, s_0 = s\right]. \tag{9}$$

In particular, the expected value is obtained from the trajectory $\xi = \{s_0, a_{i,0}, \dots\}$ under policy θ. Given $r_{i,m} \in [0,1]$, $0 \le v_{i,m}^\theta \le \frac{1}{1-\gamma}$ can be derived. Similar to single-agent, single-objective reinforcement learning, the expected value function is defined as $v_{i,m}^\theta(\rho) = \mathbb{E}_{s_0\sim\rho}\left[v_{i,m}^\theta(s_0)\right]$. Furthermore, the action-value function of agent i is defined as

$$q_{i,m}^\theta(s, a_i) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_{i,m}(s_t, a_{i,t}) \mid \theta, s_0 = s, a_{i,0} = a_i\right]. \tag{10}$$

Similarly, the goal of this study is to find the optimal policy θ^* cooperatively by minimizing the following expression:

$$\theta^* = \arg\min_{\theta\in\Delta(\mathcal{A})^{|S|}} J_\rho(\theta) := \frac{1}{N}\sum_{i=1}^{N} V_{i,1:M}^\theta(\rho), \tag{11}$$

where $V_{i,1:M}^\theta(\rho) := \left(v_{i,1}^\theta(\rho), v_{i,2}^\theta(\rho), \dots, v_{i,M}^\theta(\rho)\right)^\top$. However, each agent may have multiple conflicting objectives, it is difficult to find a policy that minimizes all objective functions. In this paper, we investigate decentralized multi-agent multi-objective systems, with the goal of finding equilibrium policies with no Pareto improvement. Specifically, no feasible alternative policy exists that can simultaneously lift the objective values of all agents on every optimization target. To this end, we define a decentralized Pareto-optimal policy.

Definition 1. *(Decentralized Pareto Optimality and Stationarity). Let $v_{i,m}^\theta$ denote the expected reward of agent $i \in [N]$ on objective $m \in [M]$ under policy $\theta \in \Delta(A)^{|S|}$. For any two policies θ and θ', if $v_{i,m}^\theta \le v_{i,m}^{\theta'}, \forall i \in [N], m \in [M]$, and there exists at least one i, m such that the inequality holds strictly, then θ is said to Pareto-dominate θ'. A policy θ^* that is not dominated by any other policy is called a Pareto-optimal policy. If there is no policy θ such that $v_{i,m}^\theta > v_{i,m}^{\theta^*}, \forall i \in [N], m \in [M]$, then θ^* is called a weakly Pareto-optimal policy. In continuous parameter optimization, if there is no common descent direction $\mathbf{d} \in \mathbb{R}^{|\theta|}$ at policy θ such that $\nabla_\theta v_{i,m}^{\theta^\top}\mathbf{d} < 0, \forall i \in [N], m \in [M]$, then θ is said to be a Pareto-stable policy.*

Similar to solving single-objective reinforcement learning problems, finding a Pareto-optimal solution in multi-objective optimization problems is generally NP-hard (Xu et al., 2025). According to Definition 1, if θ is not a Pareto-stable point and no common descent direction $\mathbf{d}$ exists, we can determine the optimal gradient weights by solving for a Pareto-stable solution. Specifically, we solve the problem in Eq. (11) by designing the following multi-agent, multi-objective reinforcement learning algorithm.

3.2 Decentralized Periodically Sampled Multi-objective Policy Gradient Algorithm

In the Section 1, we discussed the problem of conflicts among agents in multi-objective scenarios. Given the limitations of traditional decentralized projection-based gradient methods in handling gradient conflicts, the search direction for agent $i \in [N]$ is designed to solve a max-min problem within the policy space $\Delta(\mathcal{A})^{|S|}$. This minimizes the maximum projected value of all objective gradients and is supplemented by L2 regularization to ensure stability. It can be expressed as

$$\boldsymbol{d}_i^\rho(\theta_t) = \arg\min_{\boldsymbol{d}\in\Delta(\mathcal{A})^{|S|}} \left\{ \max_{m\in[M]} \left(\nabla_\theta v_{i,m}^{\theta_t}(\rho)\right)^\top \boldsymbol{d} + \frac{1}{2}\|\boldsymbol{d}\|_2^2 \right\}. \tag{12}$$

In particular, $\max_{m\in[M]} \nabla_\theta v_{i,m}^{\theta_t}(\rho)^\top$ computes the worst-case gradient projection of the update direction $\boldsymbol{d}$ over all objectives. It prevents excessive deviation from any individual objective and

balances competing objectives. For tractability, this problem can be equivalently recast as a constrained quadratic program:

$$(d_i^{\rho}(\theta_t), \beta_{i,t}) \in \arg\min_{d \in \Delta(\mathcal{A})^{|\mathcal{S}|}, \beta_i \in \mathbb{R}} \beta_i + \frac{1}{2}|\boldsymbol{d}|^2, \tag{13}$$
$$\text{subject to} \quad \nabla_\theta v_{i,m}^{\theta_t}(\rho)^\top \boldsymbol{d} \leq \beta_i, \quad i \in [N] \text{ and } m \in [M].$$

Eq. (13) introduces an auxiliary variable β_i to substitute the maximization term of the original problem, converting the non-smooth minimax problem into an efficiently solvable convex quadratic program. As Eq. (13) is a standard convex quadratic program subject to linear constraints, its optimal solution obeys the Karush–Kuhn–Tucker (KKT) first-order necessary conditions. Accordingly, we formulate its Lagrangian function as follows:

$$\mathcal{L}(\boldsymbol{d}, \beta_i, \lambda_{i,m}) = \beta_i + \frac{1}{2}\|\boldsymbol{d}\|^2 + \sum_{m=1}^{M} \lambda_{i,m} \left(\nabla_\theta v_{i,m}^{\theta_i}(\rho)^\top \boldsymbol{d} - \beta_i \right). \tag{14}$$

Here, $\lambda_{i,m}$ denotes the Lagrange multiplier corresponding to the m-th objective constraint. According to the KKT conditions, we take the partial derivative of the Lagrangian $\mathcal{L}$ with respect to $\boldsymbol{d}$ and set it to 0. After simplification, we directly obtain the solution for the policy search direction:

$$\boldsymbol{d}_i^{p}(\theta_t) = -\sum_{m=1}^{M} \lambda_{i,m}(t) \nabla_\theta v_{i,m}^{\theta_t}(\rho). \tag{15}$$

Furthermore, dual feasibility from the KKT conditions imposes constraints on the weight vector $\lambda_{i,m}$, i.e, $\sum_{m=1}^{M} \lambda_{i,m}(t) = 1$. Meanwhile, we introduce the normalization constraint $\lambda_{i,m}$ to interpret Lagrange multipliers as the weighting proportions of each objective in the joint optimization. The feasible region satisfying the above property is expressed as

$$\Delta(M) \coloneqq \left\{ \lambda \in \mathbb{R}^M \;\middle|\; \sum_{m=1}^{M} \lambda_{i,m} = 1,\ \lambda_{i,m} \geq 0,\ \forall i \in [N] \right\}. \tag{16}$$

Substituting Eq. (15) into the objective function of the original problem, the weight optimization problem can be transformed into a minimum-norm problem subject to the constraints in Eq. (16).

$$\lambda_i(t) \in \arg\min_{\boldsymbol{\lambda} \in \Delta(M)} \left\| \sum_{m=1}^{M} \lambda_{i,m}(t) \nabla_\theta v_{i,m}^{\theta_{i,t}}(\rho) \right\|_2^2. \tag{17}$$

Since the gradient of the value function with respect to policy parameters Eq. (4) depends on expected future trajectories, exact direct computation is intractable. Inspired by the decomposition in (Xiao, 2022), we approximate this gradient with a form that can be estimated via sampling:

$$\nabla_\theta v_{i,m}^{\theta}(\mu) \coloneqq \frac{\partial v_{i,m}^{\theta}(\mu)}{\partial \theta_{s,a}} = \frac{1}{1-\gamma} d_\mu^{\theta}(s) q_{i,m}^{\theta_{i,t}}(s, a_i). \tag{18}$$

Note that the sampling distribution μ used by the algorithm deviates from the initial state distribution ρ of the original problem. For direct gradient estimation via trajectory sampling, μ is commonly taken to be an empirical sampling distribution or a uniform distribution. Combining Eq. (15) and Eq. (18) yields the local multi-policy gradient for agent i as

$$g_i^{\mu}(\theta_t) = -\boldsymbol{d}_i^{\mu}(\theta_t) = \frac{1}{1-\gamma} \boldsymbol{d}_\mu^{\theta_t}(s) \sum_{m=1}^{M} \lambda_{i,m}(t) q_{i,m}^{\theta_{i,t}}(s, a_i). \tag{19}$$

After obtaining the local gradient $g_i^{\mu}(\theta_t)$, the local parameters $\theta_{i,t}$ of agent i are updated using projected gradient descent:

$$z_{i,t} = \mathcal{P}_{\Delta(\mathcal{A})^{|\mathcal{S}|}} \left[\theta_{i,t} - \eta_t g_i^{\mu}(\theta_{i,t}) \right]. \tag{20}$$

Here, $\mathcal{P}_{\Delta(\mathcal{A})^{|\mathcal{S}|}}[\cdot]$ is the projection operator that projects the policy onto the probability simplex $\Delta(\mathcal{A})^{|\mathcal{S}|}$ and guarantees the policy feasibility throughout the update. In the parameter-update stage,

each agent carries out consensus coordination, where it updates its parameters by aggregating information from neighboring agents:

$$\theta_{i,t+1} = \sum_{j \in \mathcal{N}_i(t)} w_{ij}(t) z_{j,t}. \tag{21}$$

Here, $w_{ij}(t)$ denotes the weight at time t. Furthermore, if $(i,j) \in \mathcal{E}_t$, then $w_{ij}(t) \geq \eta$, otherwise, $w_{ij}(t) = 0$, where $\eta \in (0,1)$.

However, this rule is computationally expensive for multi-agent multi-objective problems in decentralized settings. Over T time steps, each of the N agents must independently solve the optimization problem in Eq. (17) at every iteration, resulting in an overall complexity of $O(NT)$. In reinforcement learning, the time horizon T is typically on the order of tens of thousands. Combined with the sampling cost for weight evaluation, this degrades training efficiency and limits the algorithm's scalability to agent populations.

To tackle this issue, we introduce a reuse round $\mathcal{D}$. The overall algorithm flow is shown in Algorithm 1. The Eq. (17) is solved and dynamic weights $\lambda_i(t)$ are updated only at time $t\%\mathcal{D} == 0$, while the weights $\lambda_i(t)$ from the prior iteration are retained otherwise. This method has demonstrated promising results in multi-objective optimization (Xu et al., 2025). Specifically, in $t \in [T]$ iteration, update rules are formulated as follows:

$$\lambda_i(t) = \begin{cases} \alpha_t \lambda_i(t-\mathcal{D}) + (1-\alpha_t)\lambda_i^*(t), & t \bmod \mathcal{D} = 0, \\ \lambda_i(t-1), & \text{otherwise.} \end{cases} \tag{22}$$

Algorithm 1: Decentralized Periodic Reuse Multi-Policy Gradient Algorithm with (D-PRMPG)

Initialization: Step size α_t, weight matrix W_{ij}, number of agents N, total iterations T, number of objectives M, reuse period $\mathcal{D}$

```
for t = 1, 2, 3, ..., T do
    for i = 1, 2, 3, ..., N do
        for m = 1, 2, 3, ..., M do
            Compute local gradient of agent via Eq. (15) and Eq. (18);
            if t mod D = 0 then
                Solve dynamic weights via Eq. (17);
                Update weights: λ_i(t) = α_t λ_i(t − D) + (1 − α_t) λ_i^*(t);
            else
                Reuse previous weights: λ_i(t) = λ_i(t − 1);
        Solve local gradient via Eq. (19);
        Update local parameters via Eq. (20);
        Perform parameter consensus via Eq. (21);
```

4 Theoretical Analysis

This section presents the convergence analysis of the D-PMGD algorithm together with its necessary assumptions. Convergence bounds are derived for both convex and non-convex objective functions. Due to space limitations, the detailed proofs are deferred to the Appendix A.2. Finally, we provide the associated theoretical analysis to verify the feasibility of the proposed algorithm.

Assumption 1. *Let $\mathcal{G}_t$ denote a directed graph. At any time t, the time-varying graph $\left(\mathcal{V}, \bigcup_{\ell=1,\ldots,L} \mathcal{E}_{\ell+\tau}\right)$ is strongly connected and obeys the positive scale condition $\mathcal{H}$. Moreover, the weight sequence $\{w_{ij}(t)\}_{i,j\in[N]}$ satisfies the double stochasticity condition:*

$$\sum_{i=1}^{N} w_{ij}(t) = \sum_{j=1}^{N} w_{ij}(t) = 1. \tag{23}$$

Assumption 2. *For each agent $i, i \in [N]$, there exists a constant $D > 0$ such that the distance from the iterative parameter sequence $\{\theta_{i,t}\}$ to the Pareto-optimal is uniformly bounded:*

$$\|\theta_{i,t} - \theta_i^*\| \leq D \tag{24}$$

Assumption 3. *For each agent $i, i \in [N]$, the gradient of its value function satisfies the Lipschitz continuity condition. That is, there exists a constant $L > 0$ such that, for any $\theta, \theta' \in \mathbb{R}$, the following holds:*

$$\|\nabla v^i(\theta) - \nabla v^i(\theta')\| \leq L\|\theta - \theta'\|, \quad \forall \theta, \theta' \in \mathbb{R}^d. \tag{25}$$

To maintain notational consistency and clarity for the following theoretical proofs, we present rigorous definitions of the gradient estimates used hereafter. Let $\nabla G\left(\theta_t^i, \lambda_t\right)$ represent the true gradient and $\widehat{\nabla} G\left(\boldsymbol{\theta}_t^i, \lambda_t\right)$ the randomly sampled gradient estimate:

$$\nabla G(\theta_t^i, \lambda_t) = \sum_{s\in[S]} \lambda_{t,s} \nabla v_s(\theta_t^i), \quad \widehat{\nabla} G(\theta_t^i, \lambda_t) = \sum_{s\in[S]} \lambda_{t,s} \nabla v_s(\theta_t^i, \xi_t). \tag{26}$$

Among them, λ_t is obtained from the Eq. (41). After defining the auxiliary variables above, we perform a rigorous theoretical convergence analysis for the proposed algorithm. As the objective function can be either convex or nonconvex, we consider two separate cases. Specifically, our convergence analysis builds upon the assumptions presented above. First, for the ideal case where the objective function is convex, by combining the auxiliary variables and update rules discussed earlier, we can derive the convergence bounds for Algorithm 1 as stated in Theorem 1.

Theorem 1. *(Convergence of General Convex Functions): For all agents $i \in [N]$ and all objectives $m \in [M]$, if the objective function G is General convex, then the iterative sequence generated by Algorithm 1 satisfies:*

$$\frac{1}{T}\sum_{t\in[T]} \mathbb{E}\big[G(\theta_t^i, \lambda_t^i) - G(\theta^*, \lambda_t^i)\big] \leq \frac{\|\theta_1^i - \theta^*\|^2}{T\eta_1} + \frac{1}{T}\left[\frac{2\gamma^2 A^2 B^2 M^2 N^2 \nu^2 \alpha^2}{(1-\gamma)^{10}(1-\zeta)^2} + \frac{M^2 S B^2}{(1-\gamma)^4}\right]\sum_{t\in T}\eta_t \tag{27}$$

$$+ \frac{S^2\sqrt{M}\, BD}{T(1-\gamma)^2} \sum_{t\%\mathcal{D}==0} (1-\alpha_t).$$

Here, we set $\eta_t = 1/\sqrt{t}$ and $\alpha_t = 1 - 1/\sqrt{t}$. The Eq. (27) characterizes the expected suboptimality gap between the objective values generated by the algorithm and the global Pareto optimal frontier. As the number of iterations T increases, the gap converges to zero at a rate of $\mathcal{O}(1/\sqrt{T})$, indicating that the algorithm converges to the global Pareto optimal solution.

Practical RL objectives are often highly nonconvex, so we also analyze convergence in the nonconvex setting. There, convergence is characterized not by proximity to the global optimum but by reaching a Pareto-stable point. Given Assumption 3, as follows:

Theorem 2. *(Convergence of Nonconvex Functions): For all agents $i \in [N]$ and all objectives $m \in [M]$, if the objective function G is Nonconvex, then the iterative sequence generated by Algorithm 1 satisfies:*

$$\frac{1}{T}\sum_{t\in[T]} \mathbb{E}\big\|\nabla G(\theta_t^i, \lambda_t^i)\big\|^2 \tag{28}$$

$$\leq \frac{4SB(1-\gamma)/r_{\max}}{T\eta_T} + \frac{2(1-\gamma)/r_{\max}}{T} \sum_{t\%\mathcal{D}==0, t\neq 0} \frac{\mathbb{E}\left[\sum_{s\in[S]}(1-\alpha_t)\big|\hat{\lambda}_t^s - \lambda_{t-\mathcal{D}}^s\big|\right]}{\eta_t}$$

$$+ 2LS^2B^2\partial^2 \frac{1}{T}\sum_{t\in[T]}\eta_t + \frac{S^2B^2(SB+\mathcal{S})}{T(1-\gamma)^8} \sum_{t\%\mathcal{D}==0}(1-\alpha_t).$$

Here, we set $\eta_t = 1/\sqrt{t}$ and $\alpha_t = 1 - 1/\sqrt{t}$. The Eq. (28) provides an analytical upper bound for the expected gradient norm under nonconvex conditions. As the iteration count T increases, the expected gradient norm vanishes at the rate $\mathcal{O}(1/\sqrt{T})$, establishing convergence of the algorithm to a Pareto-stable point.

Remark 1. Compared with existing policy gradient methods, D-PRMPG matches the convergence rate of classic single-agent single-objective algorithms (Agarwal et al., 2021) under both general convex and nonconvex settings. While its convergence rate lags behind some state-of-the-art centralized multi-objective policy gradient algorithms Zhou et al. (2022), it is among the first decentralized multi-objective methods that provide formal convergence guarantees while substantially lowering computational overhead.

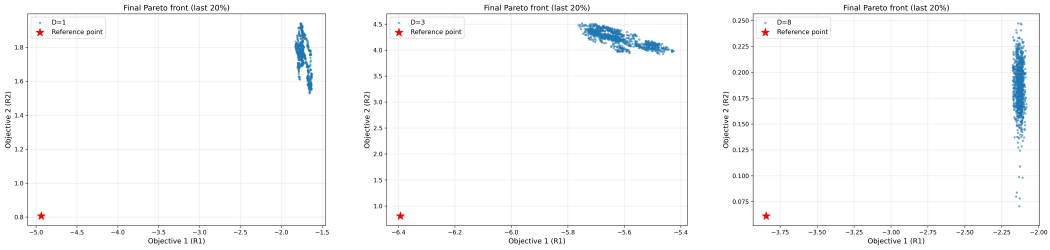
Table 1: Hypervolume comparison under different reuse periods $\mathcal{D}$. The number of $\lambda_i(t)$ computations is measured as $T/\mathcal{D} \times N$, where $T = 5000$ is the number of training episodes and $N = 4$ is the number of agents.

$\mathcal{D}$	**Hypervolume**	**Relative Improvement (%)**	**Number of λ Computations**
1	25.091	—	20000
3	**57.101**	**+127.6**	6664
5	32.594	+29.9	4000
8	32.935	+31.3	2500
9	38.175	+52.2	2220
10	42.366	+68.9	2000

5 Experimental Setup and Results Analysis

In this section, we evaluate the performance of D-PRMPG on multi-agent multi-objective environments from the MOMALand benchmark (Felten et al., 2024). Owing to space constraints, further details and parameter settings are deferred to the Appendix A.4.

To verify the effectiveness of the proposed reuse mechanism in balancing Pareto front quality and computational efficiency, we summarize in Table 1 the hypervolume (HV) of the final Pareto front under varying reuse periods $\mathcal{D}$, together with the relative improvement over the $\mathcal{D} = 1$ baseline and the total number of $\lambda_i(t)$ computations. The baseline setting yields an HV of 25.091, while $\mathcal{D} = 3$ achieves the highest HV of 57.101, marking a 127.6% relative improvement. This performance gain is coupled with a notable reduction in computational overhead: with $\mathcal{D} = 3$, only 6664 λ computations are required, compared with 20000 for $\mathcal{D} = 1$. As $\mathcal{D}$ further increases to 10, the HV drops to 42.366 and the count of λ computations falls to 2000. This indicates that an overly large reuse period impairs the algorithm's ability to adapt to the evolving Pareto front. These findings confirm that $\mathcal{D} = 3$ achieves the optimal trade-off between solution quality and computational efficiency. As shown in Figure 1, we plot the (R_1, R_2) from the last 20% of training episodes. The (R_1, R_2) are agents' two different objective reward pairs, $\mathcal{D} = 3$ produces the widest and best-distributed front. In contrast, $\mathcal{D} = 1$ collapses into a narrow region, while $\mathcal{D} = 10$ sacrifices almost all of Objective 2 or only a marginal gain in Objective 1. For brevity, all detailed results and evaluations are deferred to Appendix A.4.



(a) Reuse Periods $\mathcal{D} = 1$ (b) Reuse Periods $\mathcal{D} = 3$ (c) Reuse Periods $\mathcal{D} = 10$

Figure 1: Final Pareto fronts under different reuse periods $\mathcal{D}$. The red star marks the reference point.

6 Conclusion

This paper proposes D-PRMGD, a decentralized multi-objective policy gradient algorithm filling the gap in multi-objective policy optimization for multi-agent reinforcement learning. It uses a periodic reuse mechanism to cut computational cost while retaining convergence. Theoretical analysis shows it converges to the global Pareto optimum at $\mathcal{O}(1/\sqrt{T})$ under general convex conditions, and to a Pareto-stationary point under nonconvex conditions. Experiments on different MOMALand multi-objective environments verify its significant advantages in hypervolume and convergence stability.

AI USE STATEMENT

We did not use generative AI tools to generate research ideas or draft the core methodological content. Generative AI tools were used solely for grammatical correction, language polishing, and improving the readability of the manuscript. All research ideas, methods, results, and conclusions were developed by the authors. The authors have reviewed and approved all AI-assisted revisions and take full responsibility for the final content.

ETHICS STATEMENT

This work is purely simulation-based and does not involve human subjects, personal data, or any real-world deployment. All experiments are conducted in the MOMALand benchmark, which are open-source and publicly available. We do not collect, process, or release any sensitive or personally identifiable information. To the best of our knowledge, our research does not raise discrimination, bias, fairness, privacy, or security concerns. The proposed algorithm is a general-purpose multi-objective reinforcement learning method; while it could in principle be applied to autonomous drone coordination, we do not foresee any direct harmful applications beyond those already present in the existing literature. We have no conflicts of interest to declare. All code used in this work are provided as supplementary material to support reproducibility and research integrity.

REPRODUCIBILITY STATEMENT

To ensure reproducibility, we provide the source code and all experimental data as supplementary material. The hyperparameter settings used in our experiments are summarized in Appendix A. Theoretical assumptions, proofs, and convergence analysis are provided in Appendix A. We also describe the environment configurations, communication topologies, and evaluation metrics in the A. We hope these resources facilitate the reproduction and verification of our results.

AUTHOR CONTRIBUTIONS

If you'd like to, you may include a section for author contributions as is done in many journals. This is optional and at the discretion of the authors.

ACKNOWLEDGMENTS

Use unnumbered third level headings for the acknowledgments. All acknowledgments, including those to funding agencies, go at the end of the paper.

REFERENCES

Axel Abels, Diederik M. Roijers, Tom Lenaerts, Ann Nowé, and Denis Steckelmacher. Dynamic weights in multi-objective deep reinforcement learning. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pp. 11–20. PMLR, 2019.

Alekh Agarwal, Sham M Kakade, Jason D Lee, and Gaurav Mahajan. On the theory of policy gradient methods: Optimality, approximation, and distribution shift. *Journal of Machine Learning Research*, 22(98):1–76, 2021.

Tanmay Ambadkar, Sourav Panda, Shreyash Kale, Jonathan Dodge, and Abhinav Verma. Preference conditioned multi-objective reinforcement learning: Decomposed, diversity-driven policy optimization. *CoRR*, abs/2602.07764, 2026.

Aadyot Bhatnagar, Peter Mørch Groth, and Ali Madani. Pareto-optimal offline reinforcement learning via smooth tchebysheff scalarization. *arXiv preprint arXiv:2604.13175*, 2026.

Adam Callaghan, Karl Mason, and Patrick Mannion. Moma-ac: A preference-driven actor-critic framework for continuous multi-objective multi-agent reinforcement learning. *Neurocomputing*, pp. 132032, 2025.

Jinchi Chen, Jie Feng, Weiguo Gao, and Ke Wei. Decentralized natural policy gradient with variance reduction for collaborative multi-agent reinforcement learning. *J. Mach. Learn. Res.*, 25:172:1–172:49, 2024.

Lisha Chen, Heshan Fernando, Yiming Ying, and Tianyi Chen. Three-way trade-off in multi-objective learning: Optimization, generalization and conflict-avoidance. *Proceedings of the 37th Neural Information Processing Systems, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 36:70045–70093, 2023.

Xin Chen, Guannan Qu, Yujie Tang, Steven Low, and Na Li. Reinforcement learning for selective key applications in power systems: Recent advances and future challenges. *IEEE Transactions on Smart Grid*, 13(4):2935–2958, 2022.

Tianshu Chu, Jie Wang, Lara Codecà, and Zhaojian Li. Multi-agent deep reinforcement learning for large-scale traffic signal control. *IEEE transactions on intelligent transportation systems*, 21(3): 1086–1095, 2019.

Stefano Covone, Italo Napolitano, Francesco De Lellis, and Mario di Bernardo. Hierarchical policy-gradient reinforcement learning for multi-agent shepherding control of non-cohesive targets. In *64th IEEE Conference on Decision and Control, CDC 2025, Rio de Janeiro, Brazil, December 9-12, 2025*, pp. 3121–3126. IEEE, 2025.

Thinh T. Doan, Siva Theja Maguluri, and Justin Romberg. Finite-time performance of distributed temporal-difference learning with linear function approximation. *SIAM Journal on Mathematics of Data Science*, 3(1):298–320, 2021.

Florian Felten, Umut Ucak, Hicham Azmani, Gao Peng, Willem Röpke, Hendrik Baier, Patrick Mannion, Diederik M. Roijers, Jordan K. Terry, El-Ghazali Talbi, Grégoire Danoy, Ann Nowé, and Roxana Radulescu. Momaland: A set of benchmarks for multi-objective multi-agent reinforcement learning. *CoRR*, abs/2407.16312, 2024.

Pu Feng, Junkang Liang, Size Wang, Xin Yu, Xin Ji, Yiting Chen, Kui Zhang, Rongye Shi, and Wenjun Wu. Hierarchical consensus-based multi-agent reinforcement learning for multi-robot cooperation tasks. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2024, Abu Dhabi, United Arab Emirates, October 14-18, 2024*, pp. 642–649. IEEE, 2024.

Jie Hou, Haowen Dou, Lujuan Dang, Liangjun Chen, and Chenyang Ge. Gradient-protected value decomposition for cooperative multi-agent reinforcement learning. In *Proceeding of the 40th AAAI Conference on Artificial Intelligence, 38th Conference on Innovative Applications of Artificial Intelligence, 16th Symposium on Educational Advances in Artificial Intelligence, AAAI 2026, Singapore, January 20-27, 2026*, pp. 21779–21787. AAAI Press, 2026.

Tianmeng Hu, Biao Luo, Chunhua Yang, and Tingwen Huang. Mo-mix: Multi-objective multi-agent cooperative decision-making with deep reinforcement learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(10):12098–12112, 2023.

Harsh Kasyap and Somanath Tripathy. Privacy-preserving decentralized learning framework for healthcare system. *ACM Trans. Multim. Comput. Commun. Appl.*, 17(2s):68:1–68:24, 2021.

Vitaly Kurin, Alessandro De Palma, Ilya Kostrikov, Shimon Whiteson, and Pawan Kumar Mudigonda. In defense of the unitary scalarization for deep multi-task learning. In *Proceedings of the 36th Conference on Neural Information Processing Systems (NeurIPS 2022), New Orleans, LA, USA, November 28–December 9, 2022*, volume 35, 2022.

Sunwoo Lee, Mingu Kang, Yonghyeon Jo, and Seungyul Han. Interaction-breaking adversarial learning framework for robust multi-agent reinforcement learning. *CoRR*, abs/2605.18024, 2026.

Chao Li, Bingkun Bao, and Yang Gao. Fully decentralized cooperative multi-agent reinforcement learning is A context modeling problem. *CoRR*, abs/2509.15519, 2025.

Junyi Li, Xiaowei Qian, Yingyi Zhang, Wenlin Zhang, Guojing Li, Sheng Zhang, Xiao Han, Yichao Wang, and Xiangyu Zhao. Towards pareto-optimal tool-integrated agents with pareto ranking policy optimization. *arXiv preprint arXiv:2606.16111*, 2026.

Bo Liu, Yihao Feng, Peter Stone, and Qiang Liu. FAMO: fast adaptive multitask optimization. In *Proceedings of the 37th Conference on Neural Information Processing Systems (NeurIPS 2023), New Orleans, LA, USA, December 10–16, 2023*, 2023.

Erlong Liu, Yu-Chang Wu, Xiaobin Huang, Chengrui Gao, Ren-Jian Wang, Ke Xue, and Chao Qian. Pareto set learning for multi-objective reinforcement learning. In *Proceedings of the 39th AAAI Conference on Artificial Intelligence (AAAI 2025), Philadelphia, PA, USA, February 25–March 4, 2025*, volume 39, pp. 18789–18797, 2025.

Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. In *Proceedings of the 30st Neural Information Processing Systems, NIPS 2017*, volume 30, pp. 6379–6390, 2017.

Weixin Lu, Yao Zhou, Guowei Wan, Shenhua Hou, and Shiyu Song. L3-net: Towards learning based lidar localization for autonomous driving. In *Proceedings of the 30th IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16–20, 2019*, pp. 6382–6391. IEEE, 2019.

Kristof Van Moffaert, Madalina M. Drugan, and Ann Nowé. Scalarized multi-objective reinforcement learning: Novel design techniques. In *Proceedings of the 2013 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning, ADPRL 2013, IEEE Symposium Series on Computational Intelligence (SSCI), 16-19 April 2013, Singapore*, pp. 191–199. IEEE, 2013.

Aviv Navon, Aviv Shamsian, Gal Chechik, and Ethan Fetaya. Learning the pareto front with hypernetworks. *arXiv preprint arXiv:2010.04104*, 2020.

Nianli Peng, Muhang Tian, and Brandon Fain. Multi-objective reinforcement learning with nonlinear preferences: Provable approximation for maximizing expected scalarized return. In *Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2025, Detroit, MI, USA, May 19-23, 2025*, pp. 1632–1640. International Foundation for Autonomous Agents and Multiagent Systems / ACM, 2025.

Ruiyu Qiu, Rui Wang, Guanghui Yang, Xiang Li, and Zhijiang Shao. LPPG-RL: lexicographically projected policy gradient reinforcement learning with subproblem exploration. In Sven Koenig, Chad Jenkins, and Matthew E. Taylor (eds.), *Fortieth AAAI Conference on Artificial Intelligence, Thirty-Eighth Conference on Innovative Applications of Artificial Intelligence, Sixteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2026, Singapore, January 20-27, 2026*, pp. 25009–25017. AAAI Press, 2026.

Tabish Rashid, Mikayel Samvelyan, Christian Schröder de Witt, Gregory Farquhar, Jakob N. Foerster, and Shimon Whiteson. QMIX: monotonic value function factorisation for deep multi-agent reinforcement learning. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pp. 4292–4301. PMLR, 2018.

Diederik M. Roijers, Peter Vamplew, Shimon Whiteson, and Richard Dazeley. A survey of multi-objective sequential decision-making. *J. Artif. Intell. Res.*, 48:67–113, 2013. doi: 10.1613/JAIR.3987. URL https://doi.org/10.1613/jair.3987.

Changyin Sun, Wenzhang Liu, and Lu Dong. Reinforcement learning with task decomposition for cooperative multiagent systems. *IEEE Trans. Neural Networks Learn. Syst.*, 32(5):2054–2065, 2021.

Jun Sun, Gang Wang, Georgios B. Giannakis, Qinmin Yang, and Zaiyue Yang. Finite-time analysis of decentralized temporal-difference learning with linear function approximation. In *Proceeding of the 23rd International Conference on Artificial Intelligence and Statistics, AISTATS 2020, 26-28 August 2020, Online [Palermo, Sicily, Italy]*, volume 108 of *Proceedings of Machine Learning Research*, pp. 4485–4495. PMLR, 2020.

Tianjiao Sun, Ningyan Guo, Haozhe Gu, Yanyan Peng, and Zhiyong Feng. Integrated communication and control for energy-efficient UAV swarms: A multi-agent reinforcement learning approach. *CoRR*, abs/2509.23905, 2025.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

Richard S. Sutton, David A. McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In Sara A. Solla, Todd K. Leen, and Klaus-Robert Müller (eds.), *Advances in Neural Information Processing Systems 12, [NIPS Conference, Denver, Colorado, USA, November 29 - December 4, 1999]*, pp. 1057–1063. The MIT Press, 1999.

Gang Wang, Songtao Lu, Georgios B. Giannakis, Gerald Tesauro, and Jian Sun. Decentralized TD tracking with linear function approximation and its finite-time analysis. In *Proceedings of the Neural Information Processing Systems, NIPS 2020, December 6-12, 2020, virtual*, 2020.

Xiaowen Wang, Shuai Liu, Qianwen Xu, and Xinquan Shao. Distributed multi-agent reinforcement learning for multi-objective optimal dispatch of microgrids. *ISA Transactions*, 158:130–140, 2025. ISSN 0019-0578.

Lin Xiao. On the convergence rates of policy gradient methods. *J. Mach. Learn. Res.*, 23:282:1–282:36, 2022.

Peiyao Xiao, Hao Ban, and Kaiyi Ji. Direction-oriented multi-objective learning: Simple and provable stochastic algorithms. *Proceedings of the 37th Neural Information Processing Systems, NeurIPS 2023, Sydney, Australia, December 6–12, 2026*, 36:4509–4533, 2023.

P XinyuHuang, D XinjingCheng, Qichuan Geng, and Ruigang Yang. The apolloscape open dataset for autonomous driving and its application. *IEEE Trans Pattern Anal Machine Intell*, 42:10, 2020.

Mingjing Xu, Peizhong Ju, Jia Liu, and Haibo Yang. PSMGD: periodic stochastic multi-gradient descent for fast multi-objective optimization. In *Proceedings of the 39th AAAI Conference on Artificial Intelligence (AAAI 2025), 37th Conference on Innovative Applications of Artificial Intelligence (IAAI 2025), 15th Symposium on Educational Advances in Artificial Intelligence (EAAI 2025), Philadelphia, PA, USA, February 25–March 4, 2025*, pp. 21770–21778. AAAI Press, 2025.

Xing Xu, Rongpeng Li, Zhifeng Zhao, and Honggang Zhang. The gradient convergence bound of federated multi-agent reinforcement learning with efficient communication. *IEEE transactions on wireless communications*, 23(1):507–528, 2023.

Chao Yu, Akash Velu, Eugene Vinitsky, Yu Wang, Alexandre Bayen, and Yi Wu. The surprising effectiveness of ppo in cooperative, multi-agent games. In *Proceedings of the 36th Neural Information Processing Systems, NIPS 2022*, volume 35, pp. 24611–24624, 2022.

Xin Zhang, Jia Liu, Zhengyuan Zhu, and Elizabeth S Bentley. Compressed distributed gradient descent: Communication-efficient consensus over networks. In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*, pp. 2431–2439. IEEE, 2019.

Xiaoxiao Zhao, Jinlong Lei, Li Li, Lucian Busoniu, and Jia Xu. Distributed policy gradient with variance reduction in multi-agent reinforcement learning. *J. Syst. Sci. Complex.*, 38(5):1853–1886, 2025.

Ruida Zhou, Tao Liu, Dileep Kalathil, P. R. Kumar, and Chao Tian. Anchor-changing regularized natural policy gradient for multi-objective reinforcement learning. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022.

Junlong Zhu, Tao Mao, Mingchuan Zhang, Quanbo Ge, Qingtao Wu, and Keqin Li. Decentralized adaptive td(λ) learning with linear function approximation: Nonasymptotic analysis. *IEEE Trans. Syst. Man Cybern. Syst.*, 54(8):4630–4641, 2024.