CORE ADVANTAGE DECOMPOSITION FOR POLICY GRADIENTS IN MULTI-AGENT REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

This work focuses on the credit assignment problem in cooperative multi-agent reinforcement learning (MARL). Sharing the global advantage among agents often leads to insufficient policy optimization, as it fails to capture the coalitional contributions of different agents. Existing methods mainly assign credits based on individual counterfactual contributions, while overlooking the influence of coalitional interactions. In this work, we revisit the policy update process from a coalitional perspective and propose an advantage decomposition method guided by the cooperative game-theoretic core solution. By evaluating marginal contributions of all possible coalitions, our method ensures that strategically valuable coalitions receive stronger incentives during policy gradient updates. To reduce computational overhead, we employ random coalition sampling to approximate the core solution efficiently. Experiments on matrix games, differential games, and multi-agent collaboration benchmarks demonstrate that our method outperforms baselines. These findings highlight the importance of coalition-level credit assignment and cooperative games for advancing multi-agent learning.

1 INTRODUCTION

Cooperative Multi-Agent Reinforcement Learning (MARL) aims to train a group of agents to jointly maximize a shared objective in a common environment (Panait & Luke, 2005). Such a paradigm has shown great potential in a wide range of applications (Hu et al., 2023), including autonomous driving platoons (Shalev-Shwartz et al., 2016), multi-robot systems (Busoniu et al., 2008), and large-scale network control (Ma et al., 2024). A key challenge in MARL is how to effectively coordinate decentralized agents so that they can learn global strategies that maximize the global return (Oliehoek et al., 2008; Lowe et al., 2017).

Recent advances in policy-gradient algorithms have significantly improved stability and scalability in multi-agent learning. Among these, MAPPO (Yu et al., 2022), a multi-agent extension of PPO (Schulman et al., 2017), has become a state-of-the-art baseline for cooperative MARL. Building upon this, HAPPO and HATRPO (Kuba et al., 2021; Zhong et al., 2023) introduced sequential agent-wise updates to further stabilize learning, achieving superior performance across various benchmarks.

However, these methods typically share the same global advantage value across agents (e.g., $A_i = A_{tot}(s, a)$), which can result in suboptimal updates. This is primarily due to the synchronous nature of policy updates, where shared credit fails to distinguish individual contributions and may hinder cooperation. Such issues are often attributed to the Relative Overgeneralization (RO) problem. To mitigate this, several approaches have explored more refined credit assignment techniques. Value-based methods like VDN (Sunehag et al., 2017) and QMIX (Rashid et al., 2018), QTRAN (Son et al., 2019), QPLEX (Wang et al., 2020b) and policy-gradient methods like LICA (Zhou et al., 2020), COMA (Foerster et al., 2018), VDAC (Su et al., 2021), and FACMAC (Peng et al., 2021) assign credit from an individual perspective and have improved coordination efficiency (Wang et al., 2022b; 2020c). Additionally, DOP (Wang et al., 2020d) has tackled the exploration challenge from a maximum entropy perspective.

Despite their success, these methods focus exclusively on either global or individual perspectives. Between these extremes lies an underexplored middle ground: coalitional granularity, where credits are evaluated and allocated at the level of agent subsets (i.e., coalitions $C \subseteq N$). To address this gap, recent works have introduced Shapley value-based credit assignment from cooperative game theory into policy gradient methods (Wang et al., 2020a; Li et al., 2021; Wang et al., 2022a). While these approaches provide theoretically grounded individual attributions, they often lack interpretability in the context of multi-agent policy updates and rely on rigid baselines (e.g., no-op or zero actions), which reduce flexibility. Furthermore, many other meaningful cooperative game solutions remain unexplored in MARL.

In this paper, we propose **Core Advantage Decomposition (CORA)**, a novel credit assignment framework for multi-agent policy gradient methods. CORA estimates *coalitional advantages* by evaluating the marginal contributions of coalitions to the global return and decomposes credit using the *core solution* from cooperative game theory. This ensures *coalitional rationality* and preserves beneficial exploratory behaviors. To improve scalability, CORA employs *random coalition sampling* for efficient approximation.

The main contributions of this paper are threefold:

- Coalition-level credit assignment. We propose a novel coalitional advantage formulation and allocate credits via the strong ϵ-Core, ensuring both global consistency and coalition rationality.
- Theoretical guarantees. We prove lower bounds on coalition policy improvement, showing that beneficial coalitions are reinforced even when global advantage is negative.
- Practical effectiveness. We develop an efficient sampling approximation and demonstrate consistent gains across diverse MARL benchmarks, including matrix games, VMAS, and cooperative MuJoCo tasks.

2 Related work

This section provides an overview of key research areas relevant to our work, including traditional value decomposition methods, policy gradient methods.

2.1 Value Decomposition Methods

Value decomposition methods aim to decompose the global value function in MARL into individual contributions from each agent, thereby facilitating decentralized learning. Value-Decomposition Networks (VDN) (Sunehag et al., 2017) is a pioneering approach that splits the joint action-value function into simpler, agent-specific value functions. This decomposition significantly reduces the complexity of multi-agent learning and allows for decentralized execution. QMIX (Rashid et al., 2018), an extension of VDN, introduces a monotonic mixing function that ensures the global Q-value is a monotonic combination of individual agent Q-values.

2.2 Multi-Agent Policy Gradient Methods

Policy gradient methods, particularly MAPPO (Multi-Agent Proximal Policy Optimization) (Yu et al., 2022), have become the dominant paradigm in MARL. MAPPO adapts the well-known PPO algorithm to multi-agent environments by using a centralized critic and decentralized actors. This allows for improved sample efficiency and stability in complex tasks, where agents must learn to cooperate while maintaining decentralized control. MAPPO has shown significant performance improvements over earlier methods, such as COMA (Counterfactual Multi-Agent Policy Gradients) (Foerster et al., 2018) and MADDPG (Multi-Agent Deep Deterministic Policy Gradient) (Lowe et al., 2017), across various benchmarks like SMAC (StarCraft Multi-Agent Challenge). The key challenge for MAPPO and similar methods is their difficulty in handling high-risk exploration in cooperative environments, where deviations from the global optimum can lead to significant losses. This interference problem impedes convergence to the global optimum and limits the exploration efficiency.

2.3 Credit Assignment based on Shapley value

Shapley-based methods in MARL, integrating cooperative game theory, address the credit assignment problem by fairly distributing rewards based on each agent's contribution. Early work, such as SQDDPG (Wang et al., 2020a), uses the Shapley value in Q-learning and DDPG for continuous action spaces to calculate each agent's marginal contribution.

A more recent advancement, Shapley Counterfactual Credit Assignment (SCCA) (Li et al., 2021), refines credit assignment by considering counterfactual scenarios, improving accuracy and stability. However, SCCA faces computational challenges in multi-agent settings. SHAQ-learning (Wang et al., 2022a) also integrates Shapley values into Q-learning, enhancing stability and fairness in cooperative tasks, but it struggles with scalability and efficiency.

2.4 Cooperative Game Theory and the Core

Cooperative game theory, traditionally used in economics (Driessen, 2013), is also applied in MARL for credit assignment. Recent works (Jia et al., 2019), (Ghorbani & Zou, 2019), and (Sim et al., 2020) have adopted the Shapley value for data valuation and reward allocation. In federated learning, (Chaudhury et al., 2022) and (Donahue & Kleinberg, 2021) applied cooperative game theory to fairness and stability.

The core (Driessen, 2013), another key concept in cooperative game theory, guarantees stability by ensuring no coalition of agents can improve their outcome by deviating from the allocation. While the Shapley value has been used for fair reward distribution in MARL, traditional methods often rely on a fixed baseline, limiting their applicability in dynamic environments. Additionally, they do not address interference from high-risk explorations in cooperative MARL.

3 Background

This section provides an overview of the foundational concepts and challenges in MARL, focusing on policy gradient methods and the credit assignment methods.

3.1 Problem Formulation

In cooperative multi-agent reinforcement learning, a group of agents works together to maximize a shared return within a common environment (Panait & Luke, 2005; Kuba et al., 2021). This setting can be formalized as a Markov game (Littman, 1994; Kuba et al., 2021; Zhao et al., 2024) defined by the tuple $\mathcal{G} = \langle N, S, A, \mathbb{P}, r, \gamma \rangle$, where $N = \{1, \ldots, n\}$ is the set of agents, S is the state space, $A = \prod_{i \in N} A_i$ is the joint action space, with A_i being the action space of agent i, $\mathbb{P} : S \times A \times S \to [0, 1]$ is the transition function, $r : S \times A \to \mathbb{R}$ is the reward function, and $\gamma \in [0, 1)$ is the discount factor. At each time $t \in \mathbb{N}$, each agent i observes the full state s^t, and selects an action $a_i^t \in A_i$ drawn from its policy $\pi_i(\cdot|s^t)$. The joint action $a^t = (a_1^t, \ldots, a_n^t)$ leads to the next state $s^{t+1} \sim \mathbb{P}(s^{t+1}|s^t, a^t)$ and generates a common reward $r^t = r(s^t, a^t)$ for all agents. The agents aim to updated their policies that maximize the shared expected cumulative reward:

$$\max_{\pi} J(\pi) = \mathbb{E}_{s,a \sim \pi, \mathbb{P}} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \right]. \tag{1}$$

Under the centralized training with decentralized execution (CTDE) paradigm Oliehoek et al. (2008); Lowe et al. (2017); Yu et al. (2022), each agent i is trained with global information and execute using only local observation $o_i = O_i(s) \in \mathcal{O}_i$. A central component in training process is the global state value function $V(s)$ (the global state-action value function $Q(s, a)$), estimating the expected return from state s (after taking joint action a). Denoting the advantage $A_{tot}(s^t, a^t) = Q(s^t, a^t) - V(s^t)$ with GAE estimator

$$A_{GAE}^t = \sum_{l=0}^{\infty} (\gamma\lambda)^l \delta_{t+l} \tag{2}$$

where δ_t denotes the TD error $\delta_t = r_t + \gamma V(s_{t+1}) - V(s_t)$, a standard multi-agent policy gradient for agent i is

$$\nabla_{\phi_i} J = \mathbb{E}\left[\nabla_{\phi_i} \log \pi_i(a_i|s) A_i(s,a)\right], \tag{3}$$

where individual advantage $A_i(s,a)$ is the per-agent credit signal. Sharing the same advantage $A_{tot}(s,a)$ across agents is simple and stable, but it fails to capture heterogeneous contributions of different agents, leading to inefficient credit assignment and slower convergence.

Throughout this paper, we focus on multi-agent credit assignment via advantage decomposition for policy-gradient methods, using it to drive policy updates that strengthen effective coalitional collaboration.

3.2 Sharing Advantage

Many credit assignment methods such as COMA Foerster et al. (2018), VDN Sunehag et al. (2017), QMIX Rashid et al. (2018), and LICA Zhou et al. (2020) assign advantage or value from individual or marginal perspective.

In this paper, besides the global advantage, we also consider the coalitional advantage for each coalition of agents. Let $N = \{1, \ldots, n\}$ denote the set of all agents. For a given sample (s, a), we evaluate the scenario where agents in coalition $C \subseteq N$ take actions a_C, while the remaining agents $N \setminus C$ execute baseline actions $\bar{a}_{N\setminus C}$. The baseline action $\bar{a}$ is defined as detailed in Remark 1.

Sharing the global advantage among agents often leads to suboptimal policy updates. This method incentivizes each agent to update its policy $\pi_i(a_i|s_i)$ to either approach action a_i with $A_{tot} > 0$ or avoid those with $A_{tot} < 0$. Specifically, when an action a with $Q(s,a) < V(s)$ is explored during training, all agents are penalized via $A_{tot}(s,a) < 0$, and the policy $\pi_i(a_i|s_i)$ for each agent is updated to reduce its probability. This occurs even if a coalition C could form a superior joint action $(a_C, \bar{a}_{N\setminus C})$ satisfying $Q(s, a_C, \bar{a}_{N\setminus C}) > V(s)$.

Moreover, consider the case where the executed action a^* is already optimal. If agents in coalition C explore a new action a_C while others act optimally, and $Q(s, a_C, a^*_{N\setminus C}) < V(s)$, then the probability $\pi_i(a_i^*|s_i)$ for each agent $i \notin C$ is reduced due to $A_{tot}(s,a) < 0$, destabilizing the probability distribution over the optimal action a^*.

In summary, the value of coalition actions can be further exploited. Given a baseline action $\bar{a}_i$, agents with greater potential (those belonging to a coalition C where $Q(s, a_C, \bar{a}_{N\setminus C}) \ll V(s)$) should receive larger advantage values to promote the action $(a_C, \bar{a}_{N\setminus C})$.

Remark 1. The baseline action $\bar{a}$ provides a reference for evaluating coalition values. We do not restrict its form, but in our experiments we mainly consider the most probable action as the baseline action: (i) Discrete actions: $\bar{a}_i = \arg\max_{a_i} \pi_{\theta_i}(a_i|s_i)$, while training samples are drawn from $\pi_i(\cdot|s_i)$; (ii) Continuous actions: $\bar{a}_i = \mu_{\theta_i}(s_i)$. For example, a Gaussian policy outputs (μ_i, σ_i), with training samples $a_i \sim \mathcal{N}(\mu_i, \sigma_i)$, while the baseline uses $\mu_{\theta_i}(s_i)$. Alternatively, a masking scheme can be applied by averaging over non-coalition actions: $Q(s, a_C) = \mathbb{E}_{a_{N\setminus C} \sim \pi_{N\setminus C}}[Q(s, a_C, a_{N\setminus C})]$, which corresponds to evaluating the joint case $(a_C, \pi_{N\setminus C})$.

4 Core Advantage Decomposition for Multi-Agent Policy Gradients

In this section, we evaluate the advantage of coalition actions and propose an advantage decomposition algorithm.

4.1 Coalitional Advantage

Consider a global value function $Q(s,a)$, which describes the return of the joint action a in state s. The advantage of coalition C, denoted as $A_{tot}(s, a_C, \bar{a}_{N\setminus C})$, is defined as:

$$A_{tot}(s, a_C, \bar{a}_{N\setminus C}) = Q(s, a_C, \bar{a}_{N\setminus C}) - V(s). \tag{4}$$

The first term in the above equation, $Q_{tot}(s, a_C, \bar{a}_{N\setminus C})$, represents the global value when coalition C executes sampled action a_C, and the other agents $N \setminus C$ execute the baseline action $\bar{a}_{N\setminus C}$. The

coalitional advantage $A_{tot}(s, a_C, \bar{a}_{N\setminus C})$ evaluates the marginal contribution to global value when coalition C tries new action a_C sampled from their policies while others continue to execute the baseline action $\bar{a}_{N\setminus C}$.

By defining the advantage in this way, we can clearly quantify the contribution of each coalition action a_C to the team.

4.2 Advantage Decomposition

The next problem we need to solve is how to allocate advantage $A_i(s, a)$ to each agent $i \in N$ based on 2^n advantage values $A_{tot}(s, a_C, \bar{a}_{N\setminus C})$ (for each $C \subseteq N$). Intuitively, if a coalition action a_C yields a high advantage value $A_{tot}(s, a_C, \bar{a}_{N\setminus C})$, the total advantage assigned to the agents in that coalition should not be too small. Formally, we require

$$\sum_{i\in C} A_i(s, a) \geq A_{tot}(s, a_C, \bar{a}_{N\setminus C}) - \epsilon. \tag{5}$$

This allocation principle aligns with coalitional rationality in cooperative game theory. If coalition actions $(a_C, \bar{a}_{N\setminus C})$ are promising, it is beneficial to incentivize each a_i $(i \in C)$ to adjust its policy distribution, thereby encouraging exploration of this action in the future.

Additionally, it is essential to ensure that $\sum_{i\in N} A_i(s, a) = A_{tot}(s, a)$, which is known as effectiveness in cooperative game theory and is also widely adopted as a guiding principle in value decomposition methods. For convenience, given current state s and action a, we denote the advantage value of agent i, $A_i(s, a)$, simply as A_i.

This form coincides with the classic solution Strong ϵ-Core of cooperative game theory Driessen (2013):

$$\begin{aligned}\mathrm{Core}_\epsilon(N, A_{tot}) = \Big\{ (A_1, \cdots, A_n) \in \mathbb{R}^n \;\Big|\; & \sum_{i\in N} A_i = A_{tot}(s, a), \\ & \sum_{i\in C} A_i \geq A_{tot}(s, a_C, \bar{a}_{N\setminus C}) - \epsilon, \text{for } \forall C \subseteq N \Big\},\end{aligned} \tag{6}$$

where $\epsilon \geq 0$ is a non-negative parameter that allows for a small deviation from the ideal condition.

Generally, the ϵ-core may admit infinitely many solutions, but not all of them are desirable. In particular, some allocations satisfying coalition rationality may place all credit on a single agent, leaving others without effective incentives. To avoid such imbalanced solutions, we introduce an additional objective that penalizes large deviations from the uniform allocation. Specifically, we minimize the variance of credits among agents, leading to the quadratic program:

$$\begin{aligned}\underset{\epsilon\geq 0, A_1, \ldots, A_n}{\text{minimize}} \quad & \epsilon + \sum_{i\in N} \left(A_i - \frac{A_{tot}(s, a)}{|N|} \right)^2, \\ \text{subject to:} \quad & \sum_{i\in N} A_i = A_{tot}(s, a), \\ & \sum_{i\in C} A_i \geq A_{tot}(s, a_C, \bar{a}_{N\setminus C}) - \epsilon, \forall C \subseteq N.\end{aligned} \tag{7}$$

This formulation ensures a more balanced allocation while respecting coalition rationality. The diagram and pseudocode are shown as Figure 5 and Algorithm 1. In summary, our framework requires two critics, $Q(s, a)$ and $V(s)$, both updated using temporal-difference (TD) errors. The value critic V is employed for generalized advantage estimation (GAE), which stabilizes policy updates; in addition, $A_{tot}(s, a)$ is also estimated based on GAE. The state-action value critic Q, on the other hand, is responsible for computing the advantage A_i.

5 Theoretical Analysis

We consider n agents $N = \{1, \ldots, n\}$ with a factored joint policy $\pi_\phi(a \mid s) = \prod_{i=1}^n \pi_i(a_i \mid s; \phi_i)$.

Theorem 1. *Under compatible approximation and a natural policy gradient (NPG) step, for small step size $\alpha > 0$,*

$$\Delta \log \pi_i(a_i \mid s) \approx \alpha\, A_i(s_i, a_i),$$
$$\Delta \log \pi(a \mid s) = \sum_{i=1}^{n} \Delta \log \pi_i(a_i \mid s) \;\approx\; \alpha\, A_{tot}(s, a) = \alpha A_N,$$
$$\Delta \log \pi_C(a_C \mid s) = \sum_{i \in C} \Delta \log \pi_i(a_i \mid s) \;\approx\; \alpha \sum_{i \in C} A_i.$$

Theorem 2. *Consider one NPG step $\phi_i' = \phi_i + \alpha\, F_i^{-1} g_i$ with $g_i := \mathbb{E}[\psi_i A_i]$, $\psi_i := \nabla_{\phi_i} \log \pi_i(a_i \mid s)$, $F_i := \mathbb{E}[\psi_i \psi_i^\top]$, and step size $\alpha > 0$. Assume for each agent i that $\log \pi_i(\cdot \mid s; \phi_i)$ is twice continuously differentiable and its Hessian is uniformly bounded on the line segment between ϕ_i and ϕ_i':*

$$\left\| \nabla^2_{\phi_i} \log \pi_i(a_i \mid s; \xi_i) \right\|_{\mathrm{op}} \;\le\; L_i \quad \textit{for all } \xi_i \in [\phi_i, \phi_i'].$$

Then for any coalition $C \subseteq N$ and any sampled (s, a),

$$\Delta \log \pi_C(a_C \mid s) \;\ge\; \alpha \sum_{i \in C} A_i(s_i, a_i) \;-\; \frac{\alpha^2}{2} \sum_{i \in C} L_i \left\| F_i^{-1} g_i \right\|_2^2. \tag{8}$$

If, in addition, the strong ϵ-Core constraints hold, $\sum_{i \in C} A_i \ge A_{tot}(s, a_C, \bar{a}_{N \setminus C}) - \epsilon$, then

$$\Delta \log \pi_C(a_C \mid s) \;\ge\; \alpha\Big(A_{tot}(s, a_C, \bar{a}_{N \setminus C}) - \epsilon \Big) \;-\; \frac{\alpha^2}{2} \sum_{i \in C} L_i \left\| F_i^{-1} g_i \right\|_2^2. \tag{9}$$

Theorem 3. *Let $C^\star \in \arg\max_{C \subseteq N} A_{tot}(s, a_C, \bar{a}_{N \setminus C})$. Under the strong ϵ-Core, $\sum_{i \notin C^\star} A_i \le \epsilon$ and thus $\Delta \log \pi_{N \setminus C^\star}(a_{N \setminus C^\star} \mid s) \lesssim \alpha\, \epsilon$, while $\Delta \log \pi_{C^\star}(a_{C^\star} \mid s) \;\gtrsim\; \alpha\big(A_{C^\star} - \epsilon\big)$. Moreover, $\Delta \log \pi(a \mid s) \approx \alpha A_N$.*

Solving the quadratic programming problem (7) requires $2^{|N|}$ inferences of the value network to obtain $Q_{tot}(s, a_C, \bar{a}_{N \setminus C})$ for each coalition $C \subseteq N$. This may results in significant computational overhead for large-scale problems. To address this issue, our method randomly samples a relatively small number of coalitions $\mathcal{C} = \{C_1, C_2, \cdots, C_m\}$ and computes the desired solution satisfing the constraints of these coalitions, resulting in the quadratic programming 18. Theorem 4 shows that its approximation error can be controlled by the sample size m.

Theorem 4. *Given a distribution $\mathcal{P}$ over 2^N, and δ, $\Delta > 0$, solving the programming (18) over $O((n + 2 + \log(1/\Delta))/\delta^2)$ coalitions sampled from $\mathcal{P}$ gives an allocation vector in the δ-probable core with probability $1 - \Delta$.*

6 Experiments

We evaluate the CORA method across several multi-agent environments, including matrix games, differential games, the VMAS simulator Bettini et al. (2022), and the Multi-Agent MuJoCo environment de Lazcano et al. (2024); Kuba et al. (2021).

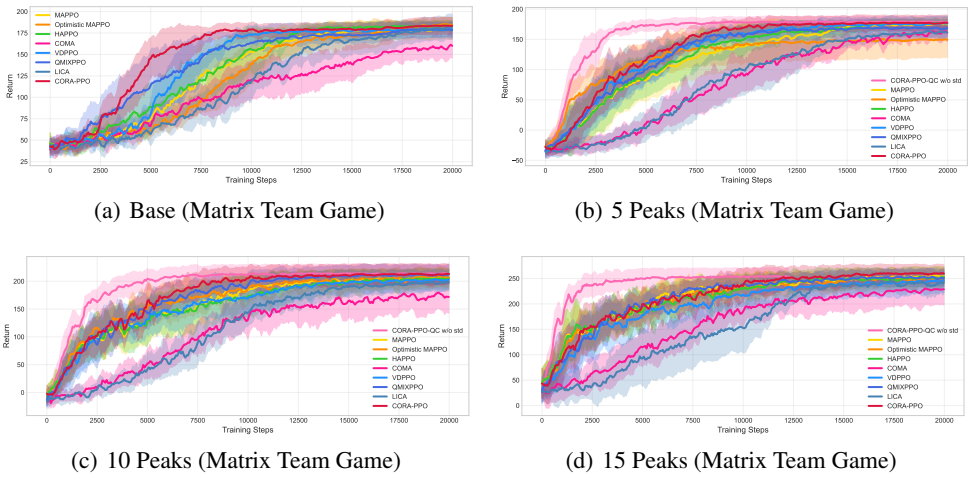
6.1 Matrix Games

In this section, we construct two types of matrix-style cooperative game environments to evaluate the fundamental performance of different algorithms.

Matrix Team Game (MTG): In this environment, agents receive a shared reward at each time step based on their joint action, determined by a randomly generated reward matrix. Each element of the matrix is uniformly sampled from the interval $[-10, 20]$. The game proceeds for 10 steps per episode. Each agent observes a global one-hot encoded state indicating the current step number, allowing them to learn time-dependent coordination strategies.

Multi-Peak Matrix Team Game: To further evaluate each algorithm's ability to optimize cooperative strategies in environments with multiple local optima, we extend MTG to design a more

challenging setting. The matrix is filled with background noise in the range $[-10, 0]$, overlaid with multiple reward peaks. Among them, one peak is the global optimum (highest value), while the rest are local optima. Actions deviating from peak combinations incur heavy penalties due to the negative background. This setting is designed to test whether algorithms can escape suboptimal solutions and discover globally coordinated strategies.



(a) Base (Matrix Team Game)

(b) 5 Peaks (Matrix Team Game)

(c) 10 Peaks (Matrix Team Game)

(d) 15 Peaks (Matrix Team Game)

Figure 1: Training performance on Matrix Team Game and its Multi-Peak variants with 5, 10, and 15 reward peaks.

As shown in Figure 1, CORA exhibits faster convergence and higher returns compared to the baseline algorithms, demonstrating superior coordination and learning efficiency in this simple and general environment. As a comparison, we also implemented a quadratic critic that directly parameterizes the joint action value as a quadratic form:

$$Q(s, a) \;=\; b(s) \;+\; \sum_{i} \langle u_i(s), a_i \rangle \;+\; \sum_{i<j} a_i^\top W_{ij}(s) a_j,$$

where a_i is the one-hot or probabilistic action vector of agent i. This representation allows us to evaluate coalition values in closed form with baseline $a_i \sim \pi_i(a_i|s)$ for $i \notin C$, e.g.,

$$Q_C(s, a_C) \;=\; \mathbb{E}_{a_{N\setminus C} \sim \pi_{N\setminus C}} \big[Q(s, a_C, a_{N\setminus C}) \big],$$

by simply replacing the action inputs of non-coalition agents with their policy distributions. The results, shown as CORA-PPO-QC, confirm this gap and highlight the stability advantage of CORA.

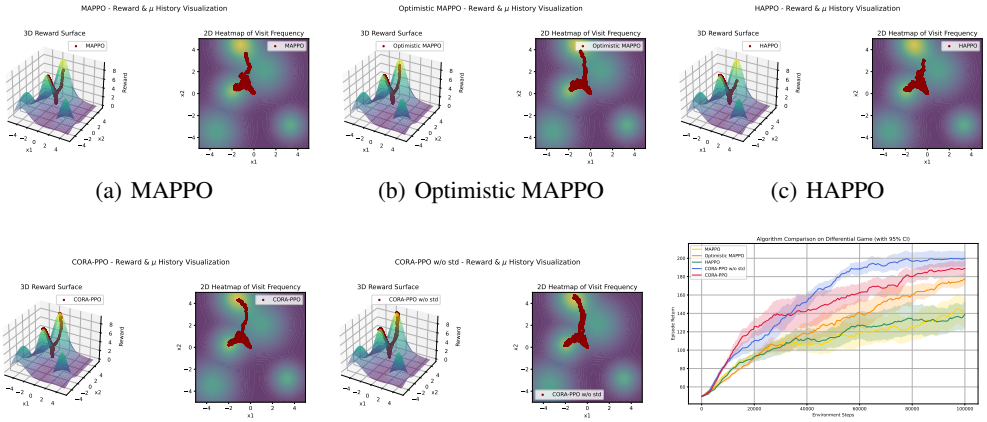
6.2 Differential Games

To demonstrate the learning process, we designed a 2D differential game environment (similar to (Wei & Luke, 2016)). Each agent selects an action $x_1, x_2 \in [-5, 5]$ at every step. The reward function $R(x_1, x_2)$ is composed of a sum of several two-dimensional Gaussian potential fields, defined as:

$$R(x_1, x_2) = \sum_{i=1}^{n} h_i \cdot \exp\left(-\frac{(x_1 - c_{x_i})^2 + (x_2 - c_{y_i})^2}{\sigma_i^2} \right) \tag{10}$$

Here, n is the number of fields, (c_{x_i}, c_{y_i}) is the center of the i-th potential field, $h_i \in [5, 10]$ indicates the peak height of the potential field, and $\sigma_i \in [1, 2]$ controls its spread. This setup results in an environment with multiple local optima, presenting significant strategy exploration and learning challenges for MARL algorithms. The environment state itself does not evolve and can be regarded as a repeated single-step game. Key parameters like location, height, width of potential fields are set by a random seed.

Figure 2(f) shows the performance of MAPPO, HAPPO, CORA-PPO, CORA-PPO without std, and Optimistic MAPPO in this environment. CORA-PPO demonstrates the best learning speed and



(a) MAPPO (b) Optimistic MAPPO (c) HAPPO

(d) CORA-PPO (e) CORA-PPO w/o std (f) All Methods

Figure 2: The reward and learning trajectories of various algorithms in the differential game scenario (μ in Gaussian strategy).

performance, and CORA-PPO without std outperforms other algorithms. We believe this is because the std term somewhat suppresses agent exploration. Since the differential game has multiple local optima, the std term constrains exploration across these optima. Furthermore, thanks to the theory of Optimistic Q-learning, Optimistic MAPPO also outperforms both HAPPO and MAPPO in this environment.

The detailed learning trajectories are visualized in Figure 2, which illustrate the learning trajectories of various methods during training (through the mean μ_i in the Gaussian policy $N(\mu_i, \sigma_i)$). It is clearly visible that the CORA-PPO series effectively promotes agents to learn optimal cooperative strategies (reaching the peak in the 3D Reward Surface; reaching the brightest point in the 2D Heatmap).

6.3 VMAS

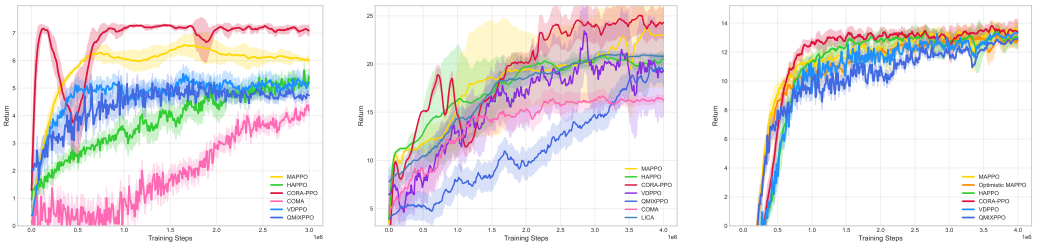
VMAS (Vectorized Multi-Agent Simulator) is a PyTorch-based vectorized multi-agent simulator designed for efficient multi-agent reinforcement learning benchmarking Bettini et al. (2022; 2024); Bou et al. (2023). It provides a range of challenging multi-agent scenarios, and utilizes GPU acceleration, making it suitable for large-scale MARL training. We selected the following scenarios for testing:

- **Multi-Give-Way**: Four agents must coordinate to cross a shared corridor by giving way to each other to reach their respective goals. This task requires strong coordination and implicit role assignment.
- **Give-Way**: Two agents are placed in a narrow corridor with goals on opposite ends. Success requires one agent to yield and allow the other to pass first, reflecting asymmetric cooperative behavior.
- **Navigation**: Agents are randomly initialized and must navigate to their own goals while avoiding collisions. The environment supports partial observability, testing decentralized policy learning and generalization.

As shown in Figure 3, CORA achieves higher returns and more stable performance compared to the other algorithms.

6.4 Multi-agent MuJoCo

To demonstrate the effectiveness of CORA in continuous control tasks, we conducted experiments on the popular benchmark Multi-Agent MuJoCo (MA-MuJoCo) Kuba et al. (2021), using its latest version, MaMuJoCo-v5 de Lazcano et al. (2024).



(a) Give-Way (b) Multi-Give-Way (c) Navigation

Figure 3: Training performance on the VMAS scenarios.

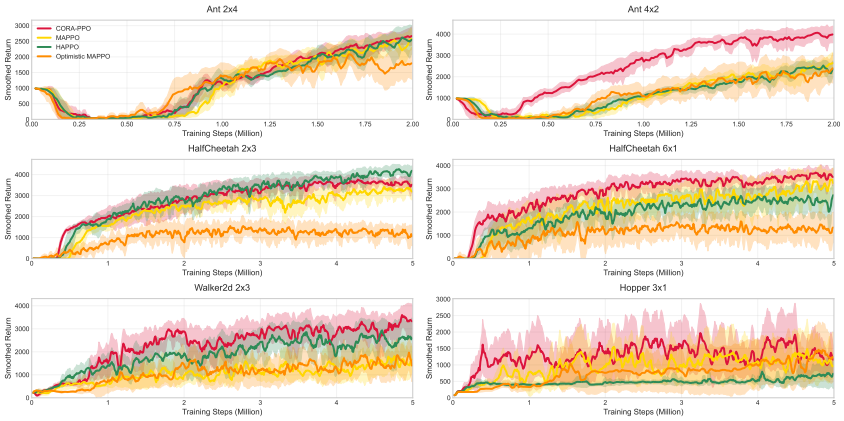


Figure 4: Training performance in the Multi-Agent MuJoCo (MaMuJoCo-v5) scenario.

As shown in Figure 4, CORA-PPO achieves state-of-the-art performance across multiple scenarios. Except for the *HalfCheetah 2x3* task where HAPPO slightly outperforms, CORA-PPO demonstrates superior results in the *Ant 4x2*, *HalfCheetah 6x1*, *Walker2d 2x3*, and *Hopper 3x1* tasks. These results highlight the effectiveness of CORA in handling diverse and challenging multi-agent continuous control environments.

7 Limitation and Conclusion

CORA (Core Advantage Decomposition) demonstrates strong performance in multiple tasks by adjusting credit assignment based on coalitional advantages. However, there are several limitations to CORA that need to be addressed in future work. The primary challenge lies in the computational overhead, as evaluating coalition actions requires additional calculations for each possible coalition, which can be costly in large-scale settings. Additionally, future work could explore alternative solution concepts from cooperative game theory and investigate their practical implications in multi-agent reinforcement learning.

Ethics Statement

We have read and will adhere to the ICLR Code of Ethics. This work does not involve human subjects, personally identifiable information, or sensitive attributes. No new datasets with personal data are collected. Experiments are conducted in standard public benchmarks under their respective licenses.

Potential negative societal impacts: our method could be used to optimize multi-agent coordination in safety-critical or competitive scenarios. To mitigate risks, we (i) avoid claims beyond measured settings; (ii) release only research artifacts necessary to reproduce results; and (iii) encourage

deployment-time safeguards (e.g., monitoring, intervention policies). We are unaware of legal compliance issues specific to the presented experiments.

Conflicts of interest: none declared.

REPRODUCIBILITY STATEMENT

We take the following steps to support reproducibility. (1) **Algorithm details.** CORA's objective and constraints are specified in Sec. 4.2 (Eq. 7, 18); the policy-gradient estimator and baselines are defined in Sec. 3. (2) **Implementation.** Pseudocode is provided in 1. (3) **Hyperparameters.** Complete training hyperparameters per environment are listed in Table 1 (actor/critic learning rates, γ, GAE λ, PPO clip, parallel envs, epochs). (4) **Environments & seeds.** We describe matrix/differential games, VMAS, and Multi-Agent MuJoCo settings in Sec. 6 and Appendix, including action/state spaces, reward definitions, and episode lengths. We run 5 random seeds for both environment and algorithm initializations and report mean with 95% confidence intervals. (5) **Code & artifacts.** Anonymized code and configuration files (including environment wrappers and plotting scripts) will be provided in the supplementary materials; instructions include exact package versions, and commands to reproduce all figures. (6) **Ablations.** We report the effect of coalition sample size and the variance regularizer in Appendix (Fig. 6, 7).

REFERENCES

Matteo Bettini, Ryan Kortvelesy, Jan Blumenkamp, and Amanda Prorok. Vmas: A vectorized multi-agent simulator for collective robot learning. *The 16th International Symposium on Distributed Autonomous Robotic Systems*, 2022.

Matteo Bettini, Ryan Kortvelesy, and Amanda Prorok. Controlling behavioral diversity in multi-agent reinforcement learning. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=qQjUgItPq4.

Albert Bou, Matteo Bettini, Sebastian Dittert, Vikash Kumar, Shagun Sodhani, Xiaomeng Yang, Gianni De Fabritiis, and Vincent Moens. Torchrl: A data-driven decision-making library for pytorch, 2023.

Lucian Busoniu, Robert Babuska, and Bart De Schutter. A comprehensive survey of multiagent reinforcement learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 38(2):156–172, 2008.

Bhaskar Ray Chaudhury, Linyi Li, Mintong Kang, Bo Li, and Ruta Mehta. Fairness in federated learning via core-stability. In *Proceedings of Advances in Neural Information Processing Systems*, pp. 5738–5750, 2022.

Rodrigo de Lazcano, Kallinteris Andreas, Jun Jet Tai, Seungjae Ryan Lee, and Jordan Terry. Gymnasium robotics, 2024. URL http://github.com/Farama-Foundation/Gymnasium-Robotics.

Kate Donahue and Jon M. Kleinberg. Optimality and stability in federated learning: A game-theoretic approach. In *Proceedings of Advances in Neural Information Processing Systems*, pp. 1287–1298, 2021.

Theo SH Driessen. *Cooperative games, solutions and applications*. Springer Science and Business Media, 2013.

Jakob Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. Counterfactual multi-agent policy gradients. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Amirata Ghorbani and James Zou. Data shapley: Equitable valuation of data for machine learning. In *Proceedings of International Conference on Machine Learning*, pp. 2242–2251, 2019.

Siyi Hu, Yifan Zhong, Minquan Gao, Weixun Wang, Hao Dong, Xiaodan Liang, Zhihui Li, Xiaojun Chang, and Yaodong Yang. Marllib: A scalable and efficient multi-agent reinforcement learning library. *Journal of Machine Learning Research*, 2023.

Ruoxi Jia, David Dao, Boxin Wang, Frances Ann Hubis, Nick Hynes, Nezihe Merve Gürel, Bo Li, Ce Zhang, Dawn Song, and Costas J Spanos. Towards efficient data valuation based on the shapley value. In *Proceedings of International Conference on Artificial Intelligence and Statistics*, pp. 1167–1176, 2019.

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

Jiahui Li, Kun Kuang, Baoxiang Wang, Furui Liu, Long Chen, Fei Wu, and Jun Xiao. Shapley counterfactual credits for multi-agent reinforcement learning. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 934–942, 2021.

Michael L. Littman. Markov games as a framework for multi-agent reinforcement learning. In William W. Cohen and Haym Hirsh (eds.), *Machine Learning Proceedings 1994*, pp. 157–163. Morgan Kaufmann, 1994. doi: 10.1016/B978-1-55860-335-6.50027-1.

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

Chengdong Ma, Aming Li, Yali Du, Hao Dong, and Yaodong Yang. Efficient and scalable reinforcement learning for large - scale network control. *Nature Machine Intelligence*, 6(9):1006–1020, 09 2024. ISSN 2522-5839. doi: 10.1038/s42256-024-00879-7.

Frans A. Oliehoek, Matthijs T. J. Spaan, and Nikos Vlassis. Optimal and approximate q-value functions for decentralized pomdps. *J. Artif. Int. Res.*, 32(1):289–353, May 2008.

Liviu Panait and Sean Luke. Cooperative multi-agent learning: The state of the art. *Auton Agent Multi-Agent Syst*, 11:387–434, 2005. doi: 10.1007/s10458-005-2631-2.

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

Tabish Rashid, Mikayel Samvelyan, Christian Schroeder, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning. In *International conference on machine learning*, pp. 4295–4304. PMLR, 2018.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Shai Shalev-Shwartz, Shaked Shammah, and Shai Amnon. Safe, multi-agent, reinforcement learning for autonomous driving, 2016.

Rachael Hwee Ling Sim, Yehong Zhang, Mun Choon Chan, and Bryan Kian Hsiang Low. Collaborative machine learning with incentive-aware model rewards. In *Proceedings of International Conference on Machine Learning*, pp. 8927–8936, 2020.

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

Jianyu Su, Stephen Adams, and Peter Beling. Value-decomposition multi-agent actor-critics. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 11352–11360, 2021.

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

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, second edition edition, 2018. See http://incompleteideas.net/book/first/the-book.html for the first edition.

J. Wang, Y. Zhang, T.-K. Kim, and Y. Gu. Shapley q-value: A local reward approach to solve global reward games. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):7285–7292, 2020a. doi: 10.1609/aaai.v34i05.6220.

Jianhao Wang, Zhizhou Ren, Terry Liu, Yang Yu, and Chongjie Zhang. Qplex: Duplex dueling multi-agent q-learning. *arXiv preprint arXiv:2008.01062*, 2020b.

Jianhong Wang, Yuan Zhang, Yunjie Gu, and Tae-Kyun Kim. Shaq: Incorporating shapley value theory into multi-agent q-learning. *Advances in Neural Information Processing Systems*, 35: 5941–5954, 2022a.

Li Wang, Yupeng Zhang, Yujing Hu, Weixun Wang, Chongjie Zhang, Yang Gao, Jianye Hao, Tangjie Lv, and Changjie Fan. Individual reward assisted multi-agent reinforcement learning. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato (eds.), *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pp. 23417–23432. PMLR, 2022b. URL https://proceedings.mlr.press/v162/wang22ao.html.

Tonghan Wang, Tarun Gupta, Anuj Mahajan, Bei Peng, Shimon Whiteson, and Chongjie Zhang. Rode: Learning roles to decompose multi-agent tasks. *arXiv preprint arXiv:2010.01523*, 2020c.

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

Ermo Wei and Sean Luke. Lenient learning in independent-learner stochastic cooperative games. *Journal of Machine Learning Research*, 17(84):1–42, 2016. URL http://jmlr.org/papers/v17/15-417.html.

Tom Yan and Ariel D Procaccia. If you like shapley then you'll love the core. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 5751–5759, 2021.

Chao Yu, Akash Velu, Eugene Vinitsky, Jiaxuan Gao, Yu Wang, Alexandre Bayen, and Yi Wu. The surprising effectiveness of ppo in cooperative multi-agent games. In *Advances in Neural Information Processing Systems*, volume 35, pp. 24611–24624. Curran Associates, Inc., 2022.

Wenshuai Zhao, Yi Zhao, Zhiyuan Li, Juho Kannala, and Joni Pajarinen. Optimistic multi-agent policy gradient. In *Proceedings of the International Conference on Machine Learning*, 2024.

Yifan Zhong, Jakub Grudzien Kuba, Siyi Hu, Jiaming Ji, and Yaodong Yang. Heterogeneous-agent reinforcement learning, 2023.

Meng Zhou, Ziyu Liu, Pengwei Sui, Yixuan Li, and Yuk Ying Chung. Learning implicit credit assignment for cooperative multi-agent reinforcement learning. *arXiv preprint arXiv:2007.02529*, 2020.

A Appendix

In this appendix, we provide detailed statements of the theorems, lemmas, and their corresponding proofs presented in the main text.

A.1 Preliminaries and Notation

We consider the on-policy occupancy measure for (s, a) unless stated otherwise. The actor parameters are $\phi = (\phi_1, \ldots, \phi_n)$ and the factored joint policy $\pi_\phi(a \mid s) = \prod_{i=1}^{n} \pi_i(a_i \mid s; \phi_i)$. Define score features and per-agent Fisher matrices

$$\psi_i(s, a_i) := \nabla_{\phi_i} \log \pi_i(a_i \mid s), \qquad F_i := \mathbb{E}[\psi_i \psi_i^\top], \qquad F = \mathrm{diag}(F_1, \ldots, F_n) \succeq 0.$$

The NPG step is

$$\phi_i' = \phi_i + \alpha\, F_i^{-1} g_i, \qquad g_i := \mathbb{E}[\psi_i\, A_i], \tag{11}$$

for step size $\alpha > 0$. Global advantage $A_{tot}(s,a) := Q_{tot}(s,a) - V(s)$ and coalitional advantage $A_{tot}(s, a_C, \bar{a}_{N\setminus C}) := Q_{tot}(s, a_C, \bar{a}_{N\setminus C}) - V(s)$. A credit allocation $\{A_i\}_{i\in N}$ satisfies the strong ϵ-Core if

$$\sum_{i\in N} A_i = A_{tot}(s,a) =: A_N, \qquad \sum_{i\in C} A_i \geq A_{tot}(s, a_C, \bar{a}_{N\setminus C}) - \epsilon,\ \forall C \subseteq N. \tag{12}$$

A.2 Compatible Function Approximation

Definition 5 (Compatible approximation)**.** *For agent i, consider $\mathcal{S}_i = \{w_i^\top \psi_i : w_i \in \mathbb{R}^{d_i}\}$. We say A_i is compatibly representable if*

$$w_i^\star = \arg\min_w\ \mathbb{E}\big[(A_i - w^\top \psi_i)^2\big]$$

exists and satisfies the normal equation $\mathbb{E}[\psi_i A_i] = \mathbb{E}[\psi_i \psi_i^\top] w_i^\star = F_i w_i^\star$.

Lemma 6. *With $g_i = \mathbb{E}[\psi_i A_i]$, the NPG step equation 11 gives $\phi_i' - \phi_i = \alpha\, F_i^{-1} g_i = \alpha\, w_i^\star$.*

Proof. From $F_i w_i^\star = g_i$, left-multiply by F_i^{-1} to obtain $w_i^\star = F_i^{-1} g_i$. Substitute into equation 11. □

Lemma 7. *For small α, the Taylor expansion yields*

$$\Delta \log \pi_i(a_i \mid s) := \log \pi_i'(a_i \mid s) - \log \pi_i(a_i \mid s) \approx \psi_i(s,a_i)^\top (\phi_i' - \phi_i).$$

Proof. Differentiate $\log \pi_i(a_i \mid s; \phi_i)$ at ϕ_i in direction $(\phi_i' - \phi_i)$. □

A.3 First-Order Changes

Theorem 1′. *Under compatible approximation and equation 11,*

$$\Delta \log \pi_i(a_i \mid s) \approx \alpha\, A_i(s_i, a_i), \tag{13}$$

$$\Delta \log \pi(a \mid s) = \sum_{i=1}^n \Delta \log \pi_i(a_i \mid s)\ \approx\ \alpha\, A_{tot}(s,a) = \alpha A_N, \tag{14}$$

$$\Delta \log \pi_C(a_C \mid s) = \sum_{i\in C} \Delta \log \pi_i(a_i \mid s)\ \approx\ \alpha \sum_{i\in C} A_i. \tag{15}$$

Proof. By Lemma 7 and Lemma 6, $\Delta \log \pi_i \approx \psi_i^\top(\alpha w_i^\star) = \alpha w_i^{\star\top} \psi_i = \alpha A_i$, which proves equation 13. Because $\log \pi = \sum_i \log \pi_i$, summing equation 13 over i and using $\sum_i A_i = A_{tot}(s,a)$ yields equation 14. Similarly, $\log \pi_C = \sum_{i\in C} \log \pi_i$ gives equation 15. □

Corollary 8. *Using $\Delta\pi(\cdot) \approx \pi(\cdot)\Delta \log \pi(\cdot)$, $\Delta\pi(a \mid s) \approx \alpha\, \pi(a \mid s) A_N$ and $\Delta\pi_C(a_C \mid s) \approx \alpha\, \pi_C(a_C \mid s) \sum_{i\in C} A_i$.*

Remark 2 (If $A_i \notin \mathcal{S}_i$). All first-order relations remain valid with A_i replaced by its L^2 projection onto $\mathcal{S}_i$. Operationally, NPG realizes this via $w_i^\star = F_i^{-1}\mathbb{E}[\psi_i A_i]$.

A.4 Coalitional Lower Bounds from the Strong ϵ-Core

Theorem 2′. *Consider one NPG step $\phi_i' = \phi_i + \alpha\, F_i^{-1} g_i$ with $g_i := \mathbb{E}[\psi_i A_i]$, $\psi_i := \nabla_{\phi_i} \log \pi_i(a_i \mid s)$, $F_i := \mathbb{E}[\psi_i \psi_i^\top]$, and step size $\alpha > 0$. Assume for each agent i that $\log \pi_i(\cdot \mid s; \phi_i)$ is twice continuously differentiable and its Hessian is uniformly bounded on the line segment between ϕ_i and ϕ_i':*

$$\big\|\nabla_{\phi_i}^2 \log \pi_i(a_i \mid s; \xi_i)\big\|_{\mathrm{op}}\ \leq\ L_i \quad \textit{for all } \xi_i \in [\phi_i, \phi_i'].$$

Then for any coalition $C \subseteq N$ and any sampled (s, a),

$$\Delta \log \pi_C(a_C \mid s) \;\geq\; \alpha \sum_{i \in C} A_i(s_i, a_i) \;-\; \frac{\alpha^2}{2} \sum_{i \in C} L_i \left\| F_i^{-1} g_i \right\|_2^2. \tag{16}$$

If, in addition, the strong ϵ-Core constraints hold, $\sum_{i \in C} A_i \geq A_{tot}(s, a_C, \bar{a}_{N \setminus C}) - \epsilon$, then

$$\Delta \log \pi_C(a_C \mid s) \;\geq\; \alpha \Big(A_{tot}(s, a_C, \bar{a}_{N \setminus C}) - \epsilon \Big) \;-\; \frac{\alpha^2}{2} \sum_{i \in C} L_i \left\| F_i^{-1} g_i \right\|_2^2. \tag{17}$$

Proof. For each i, apply the second-order Taylor expansion of $\log \pi_i(a_i \mid s; \phi_i)$ along the direction $\Delta\phi_i := \phi_i' - \phi_i$:

$$\Delta \log \pi_i(a_i \mid s) = \psi_i(s, a_i)^\top \Delta\phi_i + \frac{1}{2} \Delta\phi_i^\top \left(\nabla_{\phi_i}^2 \log \pi_i(a_i \mid s; \xi_i) \right) \Delta\phi_i,$$

for some ξ_i on the line segment between ϕ_i and ϕ_i'. With $\Delta\phi_i = \alpha F_i^{-1} g_i$ and the operator-norm bound on the Hessian,

$$\Delta \log \pi_i(a_i \mid s) \;\geq\; \alpha\, \psi_i^\top F_i^{-1} g_i \;-\; \frac{\alpha^2}{2} L_i \left\| F_i^{-1} g_i \right\|_2^2.$$

By compatible approximation, $\psi_i^\top F_i^{-1} g_i = A_i(s_i, a_i)$. Summing over $i \in C$ yields equation 16. Combining with the strong ϵ-Core inequality gives equation 17. □

A.5 Advantage Concentration on a Maximizing Coalition

Let $C^\star \in \arg\max_{C \subseteq N} A_C(s, a_C)$ with $A_C(s, a_C) := A_{tot}(s, a_C, \bar{a}_{N \setminus C})$. Since $C = N$ is admissible and $A_N(s, a_N) = A_{tot}(s, a)$, we have $A_{C^\star} \geq A_N$.

Theorem 3′. *Under equation 12:*

1. $\sum_{i \notin C^\star} A_i = A_N - \sum_{i \in C^\star} A_i \leq A_N - (A_{C^\star} - \epsilon) \leq \epsilon$, *hence* $\Delta \log \pi_{N \setminus C^\star}(a_{N \setminus C^\star} \mid s) \lesssim \alpha\, \epsilon$.
2. $\Delta \log \pi_{C^\star}(a_{C^\star} \mid s) \;\gtrsim\; \alpha \big(A_{C^\star} - \epsilon \big)$.
3. $\Delta \log \pi(a \mid s) \;\approx\; \alpha\, A_N$ *(independent of the split of $\{A_i\}$).*

Proof. (1) From $\sum_{i \in C^\star} A_i \geq A_{C^\star} - \epsilon$ and $A_{C^\star} \geq A_N$, $\sum_{i \notin C^\star} A_i \leq \epsilon$; then apply equation 15 to the complement. (2) This is equation 14. □

A.6 Theorem 4: Approximation with Sampled Coalitions

The approximate quadratic programming problem mentioned in the main text is as follows.

$$\begin{aligned} \underset{\epsilon \geq 0, A_1, \ldots, A_n}{\text{minimize}} \quad & \epsilon + \sum_{i \in N} \left(A_i - \frac{A_{tot}(s, a)}{|N|} \right)^2, \\ \text{subject to:} \quad & \sum_{i \in N} A_i = A_{tot}(s, a), \\ & \sum_{i \in C_k} A_i \geq A_{tot}(s, a_{C_k}, \bar{a}_{N \setminus C_k}) - \epsilon, \forall C_k \in \mathcal{C}. \end{aligned} \tag{18}$$

The proof of Theorem 4 an approach inspired by Yan & Procaccia (2021), where the core allocation is approximated using sampled coalitions. The key idea is to leverage the properties of the VC-dimension of a function class to bound the probability of deviating from the true allocation in the core. To establish this result, we introduce the following two known lemmas, which play a crucial role in the proof.

Before proving the theorem, we first introduce a lemma regarding the VC-dimension of a function class, as this concept is essential to understanding the behavior of the classifier we employ in the proof.

Lemma 9. *Let $\mathcal{F}$ be a function class from $\mathcal{X}$ to $\{-1, 1\}$, and let $\mathcal{G}$ have VC-dimension d. Then, with $m = O\Big(\frac{d+\log\left(\frac{1}{\Delta}\right)}{\delta^2}\Big)$ i.i.d. samples $\{x^1, \ldots, x^m\} \sim \mathcal{P}$, we have:*

$$\Big| \Pr_{x\sim\mathcal{P}}[f(x) \neq y(x)] - \frac{1}{m}\sum_{i=1}^{m} \mathbb{1}_{f(x^i)\neq y(x^i)} \Big| \leq \delta,$$

for all $f \in \mathcal{F}$ and with probability $1 - \Delta$.

This lemma essentially states that if the VC-dimension of a function class is d, then by taking a sufficient number of samples m, the empirical error rate of a classifier f on those samples is close to the true error rate with high probability (i.e., $1 - \Delta$).

In the context of the theorem, we use linear classifiers to represent the core allocation constraints. The following lemma establishes the VC-dimension of the class of linear classifiers we use.

Lemma 10. *The function class $\mathcal{F}^n = \{x \mapsto sign(w \cdot x) : w \in \mathbb{R}^n\}$ has VC-dimension n.*

This lemma states that the VC-dimension of the class of linear classifiers is equal to the dimension n of the input space, which is important for bounding the number of samples required to approximate the core allocation effectively.

Now, we combine the insights from the previous lemmas to prove the theorem.

Proof. Consider a coalition C sampled from the distribution $\mathcal{P}$. We represent the coalition as a vector $\mathrm{z}^C = (z^C, -A_{tot}(s, a_C, \bar{a}_{N\setminus C}), 1)$, where $z^C \in \{0, 1\}^n$ is the indicator vector for the coalition and $A_{tot}(s, a_C, \bar{a}_{N\setminus C})$ is the total allocation for the agents not in C.

We define a linear classifier f based on parameters $\mathrm{w}^f = (A, 1, \epsilon)$, where $\mathrm{w}^f \in \mathbb{R}^{n+2}$. The classifier $f(\mathrm{z}^C) = \mathrm{sign}(\mathrm{w}^f \cdot \mathrm{z}^C)$ is designed to capture the core allocation for each coalition C.

To ensure coalition rationality, we want the classifier f to satisfy $f(\mathrm{z}^C) = 1$ for all coalitions $C \subseteq N$. This ensures that the allocation is in the core for all coalitions. The class of such classifiers is:

$$\mathcal{F} = \{\mathrm{z} \mapsto \mathrm{sign}(\mathrm{w} \cdot \mathrm{z}) : \mathrm{w} = (A, 1, \epsilon), A \in \mathbb{R}^n\}.$$

This class of functions $\mathcal{F}$ has VC-dimension at most $n + 2$ by Lemma 10.

Now, solving the quadratic programming problem on m samples of coalitions $\{C_1, \cdots, C_m\}$ provides a solution $(\hat{A}, \hat{\epsilon})$, and the corresponding classifier $\hat{f}$. For each sample coalition C_k, we have $\hat{f}(\mathrm{z}^{C_k}) = 1$.

By applying Lemma 9, with probability $1 - \Delta$, we obtain the following inequality:

$$\Pr_{C\sim\mathcal{P}}\Big[\sum_{i\in C} \hat{A}_i - A_{tot}(s, a_C, \bar{a}_{N\setminus C}) + \hat{\epsilon} \geq 0\Big] \geq 1 - \delta.$$

This shows that the allocation vector generated by solving the quadratic programming problem over the sampled coalitions is within the δ-probable core with high probability (i.e., with probability at least $1 - \Delta$).

Thus, Theorem 4 is proved. □

A.7 Algorithm Pseudocode and Diagram

Algorithm 1 outline the implementation of CORA within a standard actor–critic training loop. At each update, a set of coalitions is sampled, and the corresponding coalitional advantages are estimated. A constrained quadratic program is then solved to assign individual credits, which are

used to guide policy updates. This procedure ensures that policy gradients reflect coalition-level contributions, encouraging coalitional coordination. For the specific setting of baseline actions, see Remark 1.

Algorithm 1 Core Advantage Decomposition (CORA)

Initialize: Central critic network θ_V, θ_Q; actor network ϕ_i for each agent i
for each training episode $e = 1, \ldots, E$ **do**
  Initialize state s^0 and experience buffer
  for each step t **do**
    Sample action a_i^t from $\pi_i(a_i|s^t;\phi_i)$ for each agent
    Execute the joint action $(a_1^t, \ldots, a_n^t)$
    Get reward r^{t+1} and next state s^{t+1}
    Add data to experience buffer
  end for
  Collate episodes in buffer into a single batch
  Compute the target value: $y^t = r(s^t, a^t) + \gamma Q_{tot}(s^{t+1};\theta_V)$
  for $t = 1, \ldots, T$ **do**
    Sample m coalitions $\mathcal{C} = \{C_1, \ldots, C_m\} \subseteq 2^N$
    for each coalition $C \in \mathcal{C}$ **do**
      Estimate coalitional advantage:

$$A_{tot}(s^t, a_C^t, \bar{a}_{N\setminus C}^t) = Q_{tot}(s^t, a_C^t, \bar{a}_{N\setminus C}^t;\theta_Q) - V(s^t;\theta_V)$$

      for each coalition $C \in \mathcal{C}$
      where $\bar{a}_i = \arg\max_{a_i} \pi_i(a_i|s_i^t;\phi_i)$
    end for
    Estimate grand coalition N's advantage $A_{tot}(s^t, a^t)$ with GAE estimator
    Solve the programming problem to obtain credit allocation $\hat{A}_i^t$
  end for
  Update actor networks ϕ_i using PPO-clipped policy gradient:

$$\nabla_{\phi_i} \log \pi_{\phi_i}(a_i^t|s_i^t) \cdot \text{clip}\left(\frac{\pi_{\phi_i}(a_i^t|s_i^t)}{\pi_{\phi_i}^{\text{old}}(a_i^t|s_i^t)}, 1-\epsilon, 1+\epsilon\right) \cdot \hat{A}_i^t$$

  Update critic θ_V using TD error: $\sum_t \left(V(s^t;\theta_V) - y^t\right)^2$
  Update critic θ_Q using error: $\sum_t \left(Q_{tot}(s^t, a^t;\theta_Q) - y^t\right)^2$
end for

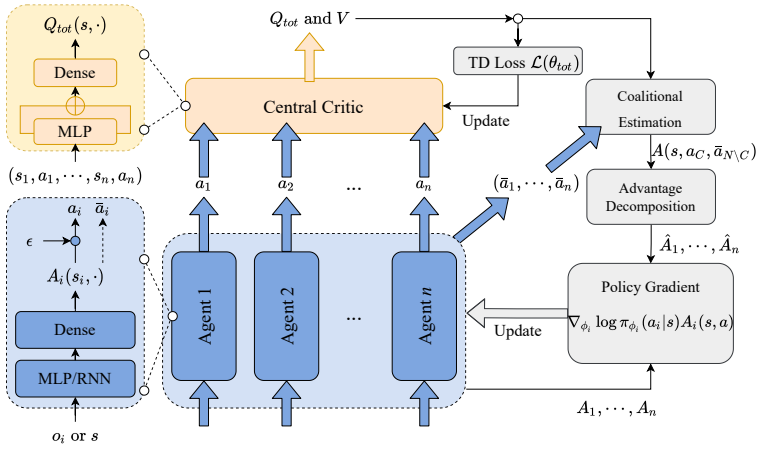


Figure 5: The framework of CORA in Multi-Agent Reinforcement Learning.

As illustrated in Figure 5, the framework demonstrates the process of Coalitional Advantage Estimation and subsequent Credit Allocation in policy gradient methods.

B EXPERIMENTAL DETAILS

Table 1: Training Hyperparameters for Each Environment

Environment	Actor LR	Critic LR	γ	GAE λ	Clip ϵ	Parallel Envs
Matrix Games	5×10^{-4}	5×10^{-3}	0.99	0.95	0.3	4
Differential Games	5×10^{-5}	5×10^{-4}	0.99	0.95	0.2	4
Multi-Agent MuJoCo	5×10^{-4}	5×10^{-3}	0.99	0.95	0.2	4
VMAS (Navigation)	5×10^{-4}	5×10^{-3}	0.99	0.95	0.2	64
VMAS (Others)	5×10^{-4}	5×10^{-3}	0.99	0.95	0.2	16

All experiments were conducted on platforms with AMD 7970X 32-Core CPU, 128GB RAM, and RTX 4090 GPU (24GB). Each algorithm was trained with a two-layer multilayer perceptron (MLP) with a hidden width of 64, except for the *Give-Way* scenario in VMAS, which used a custom network structure. Unless otherwise specified, each configuration was run five times with different random seeds for both the algorithm and the environment. We used the full coalition set (2^n coalitions) for credit assignment across all tasks. For efficiency, 64 parallel environments were used in the *Navigation* task of VMAS, while others used 16 or 4 as listed.

B.1 ABLATION STUDY OF COALITION SAMPLE SIZE, STD TERM

To evaluate the impact of coalition sampling size on performance, we conduct an ablation experiment in a differential game environment with 5 agents. Due to the high computational cost in large-scale multi-agent tasks, this experiment focuses on the 5D differential game setting, which balances complexity and tractability.

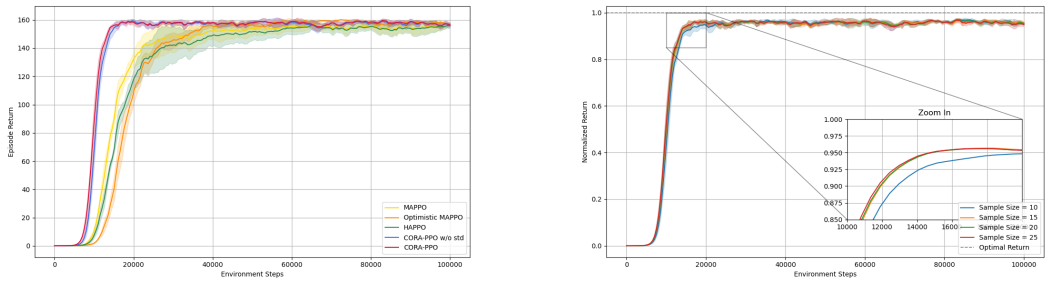


Figure 6: Training performance in the 5D differential game scenario. **Left**: Comparison among baseline methods. **Right**: Effect of coalition sampling size (sample sizes = 10, 15, 20, 25; full coalition size is $2^n - 2 = 30$). All algorithms are repeated 5 times to obtain a 95% confidence interval. Key hyperparameters: Actor learning rate 5×10^{-5}, Critic learning rate 5×10^{-4}, $\gamma = 0.99$, GAE $\lambda = 0.95$, 10 epochs per update, clip $\epsilon = 0.2$, and 4 parallel environments.

As shown in Figure 6, increasing the coalition sample size generally improves performance, particularly in the early stages of training, as highlighted in the zoomed-in window. However, even with smaller sampling sizes (e.g., 10 or 15), the CORA algorithm still achieves competitive results. This indicates that CORA is robust to sample efficiency and remains effective under reduced computation, making it applicable to environments with a moderate number of agents. In addition, CORA with variance often improves performance.

The original credit allocation formulation is a constrained quadratic program, which we relax by linearizing the variance term, resulting in a more efficient linear programming form.

Figure 7 shows that using only a small number of sampled coalitions yields an accurate and computationally efficient approximation. While constraint satisfaction may degrade slightly with fewer samples, the overall objective gap remains low, and compute time is significantly reduced. This supports the use of approximate credit assignment methods in large-scale scenarios, where full enumeration of 2^n coalitions is infeasible.

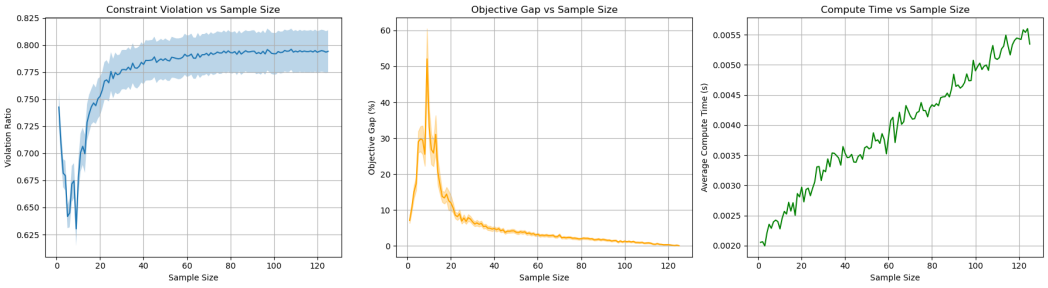


Figure 7: Error and Time Cost of Approximate Credit Assignment. Violation Ratio: proportion of coalition rationality constraints that are violated; Objective Gap: percentage difference in optimization objective compared to the full solution; Compute Time: average runtime across trials. (Number of agents = 7; Advantage functions are randomly generated across 20 trials.)

C LLM USAGE

We used a large language model (LLM) as an assistive tool for: (i) language editing (grammar and clarity), (ii) consistency checks on LaTeX labels and formatting. The LLM did not generate research ideas, proofs and experimental results. No proprietary or non-anonymized data were provided to the LLM.