Learning to Negotiate via Voluntary Commitment

Shuhui Zhu[1,2,†], **Baoxiang Wang**[3], **Sriram Ganapathi Subramanian**[2], **Pascal Poupart**[1,2]

shuhui.zhu@uwaterloo.ca, bxiangwang@cuhk.edu.cn, sriram.subramanian@vectorinstitute.ai, ppoupart@uwaterloo.ca

[1]**David R. Cheriton School of Computer Science, University of Waterloo**
[2]**Vector Institute**
[3]**School of Data Science, The Chinese University of Hong Kong, Shenzhen**

[†] Corresponding author

Abstract

The partial alignment and conflict of autonomous agents lead to mixed-motive scenarios in many real-world applications. However, agents may fail to cooperate in practice even when cooperation yields a better outcome. One well known reason for this failure comes from non-credible commitments. To facilitate commitments among agents for better cooperation, we define Markov Commitment Games (MCGs), a variant of commitment games, where agents can voluntarily commit to their proposed future plans. Based on MCGs, we propose a learnable commitment protocol via policy gradients. We further propose incentive-compatible learning to accelerate convergence to equilibria with better social welfare. Experimental results in challenging mixed-motive tasks demonstrate faster empirical convergence and higher returns for our method compared with its counterparts. Our code is available at https://github.com/shuhui-zhu/DCL.

1 Introduction

In mixed-motive applications (Dafoe et al., 2020), agents often fail to cooperate even when cooperation leads to better outcomes. One key reason is the issue of non-credible commitments. For instance, in the Prisoner's Dilemma (Table 1), mutual cooperation would lead to higher payoffs for both players compared to mutual defection, but each player, driven by its self-interest, is incentivized to defect regardless of the other's choice. As a result, credible commitments to cooperate cannot be established.

To mitigate the commitment problem, a commitment device (Rogers et al., 2014; Sun et al., 2023) is often required to ensure that agents fulfill their commitments, either by binding their actions to fixed strategies (Schelling, 1980; Renou, 2009; Kalai et al., 2010; DiGiovanni & Clifton, 2023) or imposing penalties for noncompliance (Bryan et al., 2010). In particular, conditional commitment devices (Kalai et al., 2010; Dafoe et al., 2020) have been verified to enhance cooperation in the Prisoner's Dilemma. When one player conditionally commits to cooperate if and only if the other does the same, the other player is motivated to cooperate. However, these conditional commitment mechanisms, tailored to specific problems, typically rely on fixed, pre-specified rules, leaving no room for adaptation in more complex, dynamic environments. Additionally, such mechanisms are designed primarily for simple, repeated games such as the Prisoner's Dilemma, limiting their applicability to a broader range of strategic scenarios where the conditions for cooperation may evolve over time.

To address these limitations, we propose a learnable commitment mechanism, named differentiable commitment learning (DCL) based on the introduced Markov Commitment Games (MCGs, Fig-

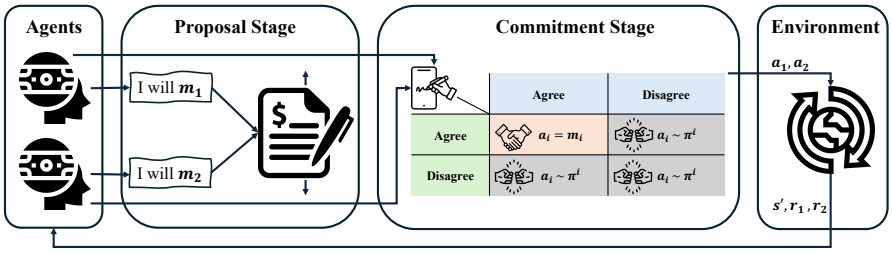


Figure 1: Markov Commitment Game: A Markov commitment game consists of three stages. In the first stage, agents announce their proposed future actions. In the second stage, agents observe others' proposals and decide whether to commit to the joint plan. In the final stage, agents choose their actions: if all agents commit, they follow their proposals; if any agent does not commit, all agents independently select actions based on the current state. Afterward, agents observe the resulting rewards and transit to the next state.

ure 1). MCGs are a variant of commitment games (Renou, 2009; Bryan et al., 2010; Forges, 2013; DiGiovanni & Clifton, 2023). In two-phase commitment games, each agent first announces a unilateral commitment to a subset of possible strategies, then selects an action based on strategies they have committed to. Different from commitment games, MCGs incorporate an additional proposal phase, where agents release a proposed future plan of their own actions in the current state without disclosing their strategies for other states. As a result, MCGs do not require mutual transparency of commitment strategies and avoid incompatibilities in commitment implementation. Furthermore, commitments in MCGs have linear size in the planning horizon and are therefore more tractable for agents to reason through, whereas in conditional commitment games (Bryan et al., 2010; Forges, 2013; DiGiovanni & Clifton, 2023), commitments are recursive and potentially infinite.

The core idea of DCL in MCGs is to learn a commitment protocol that enables agents to voluntarily align their actions based on the commitments of others. Under the assumption of self-interested agents, DCL adopts the scheme of reinforcement learning (Sutton, 2018), optimizes long-term individual returns via policy gradients. Different from common RL algorithms that treat other agents as part of the environment, DCL allows backpropagation through actual or estimated policies of other agents. The advantages of DCL are twofold. 1) The commitment mechanism is agnostic to environment dynamics so that it can generalize across various tasks. Whereas in commitment games (Renou, 2009; Bryan et al., 2010; Forges, 2013; DiGiovanni & Clifton, 2023), the commitment strategies are pre-defined for specific problems. 2) DCL provides more accurate value evaluation and policy gradient estimations through backpropagation across commitment channels. By explicitly leveraging the interdependence of agents' decisions, DCL enhances learning outcomes. Whereas other baseline RL algorithms (Schulman et al., 2017; Haupt et al., 2022; Ivanov et al., 2023) treat other agents as part of the environment, resulting in non-stationarity from each agent's perspective.

Extensive experiments in tabular, sequential and iterative social dilemmas verify the efficiency of our approach in promoting cooperation. DCL significantly outperforms several baseline methods, including independent RL, contract-based reward transfer RL, and mediated multi-agent RL, often by establishing mutually beneficial multilateral commitments.

2 Related Works

2.1 Binding Contracts Mechanism

Binding contracts are generally applied to establish commitments in multi-agent systems. The literature offers various approaches to contract design. Wang et al. (2024); Han et al. (2017); Sandholm & Lesser (1996) developed contracts that bind agents' future actions through side payments, rewarding agents for fulfilling commitments and penalizing them for noncompliance. Haupt et al. (2022);

Sodomka et al. (2013) also explored mechanisms where agents voluntarily agree to binding reward transfers. However, these methods directly alter agents' incentives, which may not be feasible in practice.

Instead, Kramár et al. (2022); De Jonge & Zhang (2020); Hughes et al. (2020) proposed adaptive binding actions without reward transfers, which are similar to MCGs but differ in specific details. De Jonge & Zhang (2020) focused on turn-taking games with unilateral commitments, while MCGs emphasize simultaneous moves and multilateral commitments. Hughes et al. (2020) required agents to propose a joint plan for all, with multilateral commitment only if they propose the same plan. In MCGs, however, each agent proposes an individual plan and uses a separate commitment model to decide whether to commit or not. Kramár et al. (2022) introduced pairwise negotiation through Nash Bargaining Solution (NBS) (Binmore et al., 1986), aiming to maximize the product of agents' utilities. In contrast, MCGs focus on selfish agents aiming to maximize their individual long-term returns.

2.2 Altruistic Third Party

Without manipulating agents' rewards, (Ivanov et al., 2023; McAleer et al., 2021; Greenwald et al., 2003) introduced pro-social third parties to mediate agents' actions and induce cooperative behaviors. These approaches optimize social welfare such as the sum of agents' returns while incorporating rationality constraints that define equilibria, ensuring that self-interested agents have no incentive to deviate from their strategies. Specifically, utilitarian correlated-Q learning (Greenwald et al., 2003) utilized a centralized model to optimize the joint action probability distribution of all agents, with an objective that maximizes the sum of the agents' rewards. In contrast, Ivanov et al. (2023); McAleer et al. (2021) trained agents to optimize their individual payoffs, allowing them to follow the recommendations of a prosocial mediator or take their actions independently if those recommendations do not align with their self-interests. However, these approaches still rely on a centralized altruistic third party, which may become ineffective in highly conflicting environments where collective interests significantly clash with individual self-interests.

3 Background on Commitment Games

A normal form game $G = (\mathcal{N}, (\mathcal{R}^i, \mathcal{A}^i)_{i\in\mathcal{N}})$ consists of a set $\mathcal{N}$ of agents, where each agent i chooses an action $a^i \in \mathcal{A}^i$ and earns a reward according to the function $\mathcal{R}^i : \prod_j \mathcal{A}^j \to \mathbb{R}$. A commitment game (Renou, 2009) extends a normal form game to two phases where each agent first makes a commitment and then plays an action. Formally, a commitment game $CG = (\mathcal{N}, (\mathcal{R}^i, \mathcal{A}^i, \mathcal{C}^i)_{i\in N})$ extends a normal form game with a commitment space $\mathcal{C}^i$ for each agent. Player i's strategy (c^i, σ^i) consists of a commitment $c^i \in \mathcal{C}^i$ and a response function $\sigma^i : \prod_j \mathcal{C}^j \to \mathcal{A}^i$. For example, Renou (2009) considered unconditional unilateral commitments where a commitment $c^i \subseteq \mathcal{A}^i$ is a subset of the action space, meaning that the agent commits to choose an action in that subset. Such unconditional unilateral commitments can yield better equilibria (i.e., Pareto optimal) when ruling out some threats will incite other agents to cooperate. However, in other games such as Prisoner's Dilemma, no unilateral commitment will induce convergence to mutual cooperation.

Kalai et al. (2010) proposed conditional unilateral commitments $\mathcal{C}^i : \prod_{j\neq i} \mathcal{C}^j \to \mathcal{A}^i$, where agents commit to some actions conditioned on the commitments of others. This space of commitments is recursive and potentially infinite, however it can turn mutual cooperation into a stable equilibrium in Prisoner's Dilemma when both agents commit to cooperating conditioned on the other one cooperating too. Kalai et al. (2010) further augmented conditional unilateral commitments with a voluntary commitment space. In this voluntary commitment space, agents are allowed to play the normal form game G without making any advanced commitment. Thus, agents will independently select their actions $a^i \in A^i$ if they voluntarily decide not to commit to any $c^i \in \mathcal{C}^i$. However, this conditional commitment mechanism requires agents to reveal their commitment strategies (Kalai et al., 2010; Forges, 2013) or source code of their models (DiGiovanni & Clifton, 2023), which may be imprac-

tical and lead to incompatibilities in commitment implementation. Two tables in the supplementary material summarize the differences and similarities between various types of games and associated algorithms to optimize strategies.

4 Markov Commitment Games

The ability to make binding commitments is a fundamental mechanism for promoting cooperation. To enable strategic commitment-making among intelligent agents in multi-agent systems, we formulate a Markov Commitment Game (MCG, Figure 1), formally defined by a tuple

$$MCG = (\mathcal{N}, \mathcal{S}, \mathcal{T}, (\mathcal{M}^i, \mathcal{C}^i, \mathcal{A}^i, \mathcal{R}^i)_{i\in\mathcal{N}}, \gamma). \tag{1}$$

MCGs include three stages. At each time step t, the agent $i \in \mathcal{N}$ observes a global state $s_t \in \mathcal{S}$ and announces a proposal $m^i \in \mathcal{M}^i = \mathcal{A}^i$ in the first stage. Then each agent i observes the joint proposal $\mathbf{m} = (m^i)_{i\in\mathcal{N}}$ and makes a commitment decision $c^i \in \mathcal{C}^i = \{0, 1\}$ in the second stage, where $c^i = 1$ indicates that agent i commits to the joint proposal, $c^i = 0$ indicates that agent i rejects the joint proposal. In the third stage, if all agents commit to the joint plan, they execute the actions in the proposal, i.e., $a^i = m^i, \forall i \in \mathcal{N}$; otherwise, each agent i independently selects an action $a^i \in \mathcal{A}^i$. Agent i receives the reward r^i, determined by the reward function $\mathcal{R}^i : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, where $\mathcal{A} = (\mathcal{A}^i)_{i\in\mathcal{N}}$ represents the joint action space. Meanwhile, the next state s_{t+1} is generated by the transition function $\mathcal{T} : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$, which satisfies the Markov property and the stationarity condition, i.e., $\mathcal{T}(s_{t+1} = s'|s_t = s, \mathbf{a}_t = \mathbf{a}) = \mathcal{T}(s_{t+1} = s'|s_t = s, \mathbf{a}_t = \mathbf{a}, s_{t-1}, \mathbf{a}_{t-1}..., s_0, \mathbf{a}_0) = \mathcal{T}(s'|s, \mathbf{a}), \forall t$. This process is repeated until the episode ends. It is important to note that the transition distribution conditions on the current state and joint actions only, not on the proposals or commitment decisions. This is because proposals and commitments indirectly influence the transition by affecting the actions executed.

In an MCG, each agent has three decisions to make at each time step: what to propose, whether to commit or not, and how to choose actions without joint commitment. Therefore, we decompose each agent's behavioral model into three strategic policies. The proposal policy, $\phi^i_{\eta^i} : \mathcal{S} \to \Delta(\mathcal{M}^i)$, maps the current state s_t to a distribution over agent i' space of proposals. The commitment policy, $\psi^i_{\zeta^i} : \mathcal{S} \times \mathcal{M} \to \Delta(\mathcal{C}^i)$, depends on the state s_t and the joint proposal $\mathbf{m}_t \in \mathcal{M} = (\mathcal{M}^i)_{i\in\mathcal{N}}$. The action policy, $\pi^i_{\theta^i} : \mathcal{S} \to \Delta(\mathcal{A}^i)$, samples action based on the current state s_t only.

MCGs adopt a strategic commitment mechanism in mixed-motive multi-agent systems. In this framework, the environment also serves as a commitment device, enforcing agents' voluntarily imposed restrictions on their future actions. Agents in MCGs have access to this device, which is effective only when all self-interested agents agree to commit to a public joint plan. If any agent declines, all agents will independently select actions without restrictions by commitment. Thus, the commitment device facilitates a conditional commitment: agents agree to execute their proposed actions only if every other agent also commits to the joint plan.

Driven by self-interest, the objective of each agent i is to find the optimal strategy $(\phi^i_{\eta^{i*}}, \psi^i_{\zeta^{i*}}, \pi^i_{\theta^{i*}})$ that maximizes their future expected return, i.e. the expected cumulative discounted reward, defined by

$$\max_{\eta^i,\zeta^i,\theta^i} V^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s) = \mathbb{E}_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}[\sum_{k=t}^{\infty} \gamma^{k-t} r^i_{k+1} | s_t = s], \tag{2}$$

where γ is the discounted factor, $\boldsymbol{\phi} = (\phi^i_{\eta^i})_{i\in\mathcal{N}}$, $\boldsymbol{\psi} = (\psi^i_{\zeta^i})_{i\in\mathcal{N}}$, $\boldsymbol{\pi} = (\pi^i_{\theta^i})_{i\in\mathcal{N}}$. Note that agent i's value function $V^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s)$ is dependent on other agents' strategies, as the collective actions of all agents jointly decide the rewards and state transitions in multi-agent systems. Meanwhile, each agent's proposal and commitment decision also indirectly affect others' expectation of their future returns. Therefore, the impact of other players' policies on each agent's objective should be properly evaluated during learning.

4.1 Equilibrium Analysis in Prisoner's Dilemma

MCGs induce a conditional commitment mechanism, which can lead to different strategic behaviors and outcomes compared to a game without such commitments.

Proposition 4.1. *Mutual cooperation is a Pareto-dominant Nash equilibrium in the MCG of the Prisoner's Dilemma.*

Specifically, we demonstrate with Proposition 4.1 that with the ability to commit, both players have an incentive to strategically propose and commit to cooperation, given the other agent does the same, thereby transforming mutual cooperation into a Pareto-dominant Nash equilibrium. The formal proof of this proposition is provided in Appendix 12.

5 Differentiable Commitment Learning

Based on MCGs, we propose differentiable commitment learning (DCL) under the assumption of self-interested agents. Instead of treating other agents as part of the environment, DCL considers joint actions when evaluating individual returns. To formulate this idea, we define the state-action value function of agent i in MCGs as $Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s,\mathbf{a}) = \mathbb{E}_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}[\sum_{k=t}^{\infty}\gamma^{k-t}r^i_{k+1}|s_t = s, \mathbf{a}_t = \mathbf{a}]$, representing the expected future returns conditioned on the current state and the joint actions. Because the environment's transitions and reward function in MCGs depend only on the state and joint actions, the state-action value function does not condition on proposals or commitments either. Under the scheme of on-policy reinforcement learning (Sutton, 2018), DCL estimates this state-action value function by minimizing the mean square error between $Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s,\mathbf{a})$ and the Monte Carlo returns $\hat{G}^i_t = \sum_{k=t}^{T}\gamma^{k-t}r^i_{k+1}$ of the sampled trajectories. Similar to the policy gradient theorem (Sutton et al., 1999), we then derive unbiased policy gradients based on $Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s,\mathbf{a})$ in Equations (3), (4), and (5) respectively. The complete proof of Lemma 5.1 is provided in Appendix 10.

Lemma 5.1. *Given proposal policy $\phi^i_{\eta^i}$, commitment policy $\psi^i_{\zeta^i}$ and the action policy $\pi^i_{\theta^i}$ of each agent i in an MCG (1), the gradients of the value function $V^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s)$ w.r.t. θ^i, ζ^i, η^i are*

$$\nabla_{\theta^i}V^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s) \propto \mathbb{E}_{x\sim\rho_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}},\mathbf{m}\sim\boldsymbol{\phi},\mathbf{c}\sim\boldsymbol{\psi},\mathbf{a}\sim\boldsymbol{\pi}}\Big[\Big(1-\mathbb{1}(\mathbf{c}=\mathbf{1})\Big)Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(x,\mathbf{a})\nabla_{\theta^i}\log\pi^i(a^i|x)\Big], \tag{3}$$

$$\begin{aligned}\nabla_{\zeta^i}V^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s) \propto &\mathbb{E}_{x\sim\rho_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}},\mathbf{m}\sim\boldsymbol{\phi},\mathbf{c}\sim\boldsymbol{\psi},\mathbf{a}\sim\boldsymbol{\pi}}\Bigg[\Big[\mathbb{1}(\mathbf{c}=\mathbf{1})Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(x,\mathbf{m}) + \Big(1-\mathbb{1}(\mathbf{c}=\mathbf{1})\Big)Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(x,\mathbf{a})\Big]\\ &\nabla_{\zeta^i}\log\psi^i(c^i|x,\mathbf{m}) + \Big[Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(x,\mathbf{m}) - Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(x,\mathbf{a})\Big]\prod_{k\neq i}\mathbb{1}(c^k=1)\cdot\nabla_{\zeta^i}\mathbb{1}(c^i=1)\Bigg],\end{aligned} \tag{4}$$

$$\begin{aligned}\nabla_{\eta^i}V^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s) \propto &\mathbb{E}_{x\sim\rho_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}},\mathbf{m}\sim\boldsymbol{\phi},\mathbf{c}\sim\boldsymbol{\psi},\mathbf{a}\sim\boldsymbol{\pi}}\Bigg[\Big[\mathbb{1}(\mathbf{c}=\mathbf{1})Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(x,\mathbf{m}) + \Big(1-\mathbb{1}(\mathbf{c}=\mathbf{1})\Big)Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(x,\mathbf{a})\Big]\\ &\cdot\Big(\nabla_{\eta^i}\log\phi^i(m^i|x) + \sum_j\nabla_{\eta^i}\log\psi^j(c^j|x,\mathbf{m})\Big) + \sum_j\prod_{k\neq j}\mathbb{1}(c^k=1)\Big[Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(x,\mathbf{m})\\ &- Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(x,\mathbf{a})\Big]\cdot\nabla_{\eta^i}\mathbb{1}(c^j=1)\Bigg],\end{aligned} \tag{5}$$

where $\mathbb{1}(\cdot)$ denotes the indicator function, which equals 1 if the condition inside is true and 0 otherwise; $\rho_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(x)$ denotes a discounted probability of state x encountered, starting at s and then with all agents following $\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}$: $\rho_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(x) = \sum_{t=0}^{\infty}\gamma^t Pr\{s_t = x|s_0 = s\}$.

Through policy gradients in Lemma 5.1, DCL enables agents to optimize their strategies by considering both direct and indirect effects of their policies on their utilities. To capture the direct impact, DCL allows agents to differentiate through their own policies, updating in the direction that maximizes their individual returns. On the other hand, DCL allows agents to consider how their decisions

influence others' commitments and how these influences, in turn, affect their own utilities. This indirect influence is leveraged by differentiation through the commitment policies of other players when computing $\nabla_{\eta^i} V^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s)$. To backpropagate through discrete commitments, we apply the Gumbel-Softmax distribution (Jang et al., 2016) for differentiable sampling.

Instead of limiting DCL to centralized training (Appendix 11.1) with access to other agents' policies, we extend DCL to fully decentralized settings (Appendix 11.2). In decentralized DCL, each agent estimates others' policies and differentiates through these estimates to update their own policies.

Algorithm 1 Differentiable Commitment Learning

Input: initial parameters of action policy θ^i, commitment policy ζ^i, proposal policy η^i, action-value function w^i for $i \in \mathcal{N}$, learning rate β, Lagrange multiplier λ, number of iterations T.

for k=$0, 1, 2, ..., T-1$ **do**

Collect set of trajectories $\mathcal{D}_k = \{\tau_t\}$ by running latest policies $(\theta^i, \zeta^i, \eta^i), \forall i \in \mathcal{N}$.

Compute Monte-Carlo discounted accumulative rewards $\hat{G}^i_t, \forall i \in \mathcal{N}$.

Fit value function with gradient descent by minimizing the mean-squared error:

$$w^i_{k+1} = \arg\min_{w^i} \frac{1}{|\mathcal{D}_k|T} \sum_{\tau \in \mathcal{D}_k} \sum_{t=0}^{T} (Q^i_{w^i}(s_t, \mathbf{a}_t) - \hat{G}^i_t)^2.$$

Estimate action policy gradient $\hat{g}_{\theta^i_k}$ according to Equation (3).

Estimate commitment policy gradient $\hat{g}_{\zeta^i_k}$ according to Equation (4).

Estimate proposal policy gradient $\hat{g}_{\eta^i_k}$ w.r.t. expected return according to Equation (5).

Estimate proposal policy gradient $\hat{g}'_{\eta^i_k}$ w.r.t. the incentive-compatible constraints by

$$\frac{1}{|\mathcal{D}_k|} \sum_{\tau \in \mathcal{D}_k} \sum_{t=0}^{T} \sum_{j} \nabla_{\eta^i_k} \min\{0, Q^j_{w^j_{k+1}}(s_t, \mathbf{m}_t) - Q^j_{w^j_{k+1}}(s_t, \mathbf{a}_t)\}.$$

Update policy parameters for all agents with gradient ascent,

$$\theta^i_{k+1} = \theta^i_k + \beta \hat{g}_{\theta^i_k}, \zeta^i_{k+1} = \zeta^i_k + \beta \hat{g}_{\zeta^i_k}, \eta^i_{k+1} = \eta^i_k + \beta \hat{g}_{\eta^i_k} + \lambda \hat{g}'_{\eta^i_k}.$$

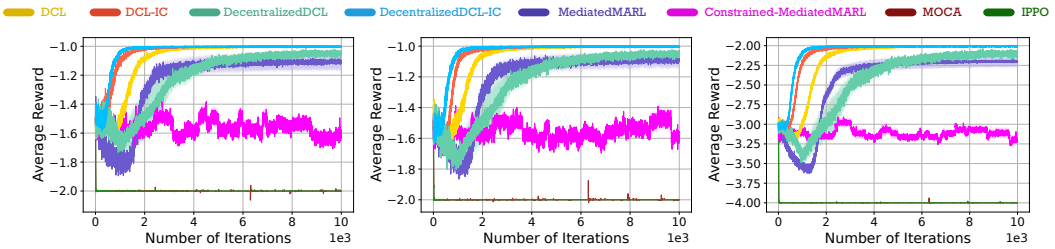
end for

5.1 Incentive-Compatible Constraints

Although mutual cooperation can be a Nash equilibrium in MCGs for some mixed-motive environments, agents may still have the equilibrium selection problem when multiple equilibria exist. For instance, mutual defection is another Nash equilibrium of the MCG in Prisoner's Dilemma, with less pay-offs of both agents compared to mutual cooperation equilibrium in Lemma 5.1. Even if agents are motivated by self-interest to select mutual cooperation equilibria over mutual defection equilibria with DCL, they may fail to find the equilibria with better outcomes because of inefficient exploration. To address this challenge, we introduce a set of incentive-compatible constraints on agents' proposal policies in Equation (6), which encourage agents to find mutually beneficial proposals.

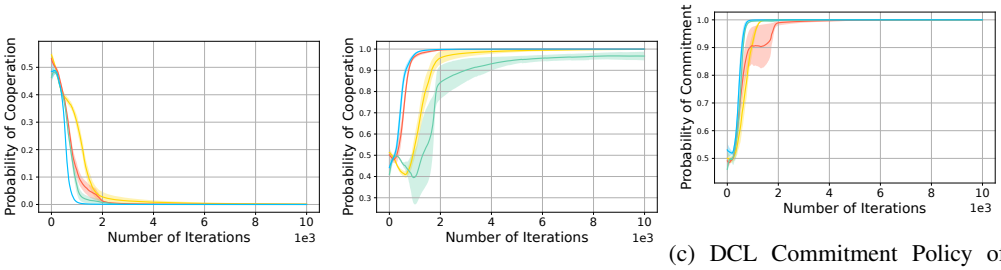
$$\mathbb{E}_{\mathbf{m} \sim \boldsymbol{\phi}}[Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s, \mathbf{m})] \geq \mathbb{E}_{\mathbf{a} \sim \boldsymbol{\pi}}[Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s, \mathbf{a})] \quad \forall\, i. \tag{6}$$

Combining these incentive-compatible constraints with the self-interested objective, agents are driven to maximize their expected returns and propose mutually beneficial agreements. If a joint proposal results in outcomes worse than actions induced by independent action policy for any player, agents are penalized during training through a regularization term induced by constraints in Equation (6). This regularization encourages agents to develop better agreements that benefit all players. Meanwhile, these constraints do not sacrifice agents' self-interests, as they retain the ability to reject



(a) Agent 1 Reward in PD (b) Agent 2 Reward in PD (c) Social Welfare in PD

Figure 2: Prisoner's Dilemma: DCL v.s. Other Baselines



(a) DCL Action Policy (b) DCL Proposal Policy (c) DCL Commitment Policy of (C, C)

Figure 3: DCL Policies in Prisoner's Dilemma

proposals that do not enhance their own utility. Thus, they will follow their unconstrained policies unless a mutually beneficial agreement emerges.

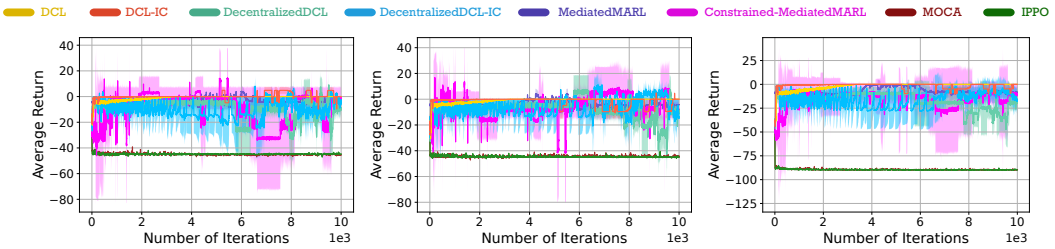
It is important to note that feasible solutions always exist for Equation (6), as agents can align their proposal policies with their action policies, i.e. $\phi^i(s) = \pi^i(s)$ for $\forall i \in \mathcal{N}$. We then integrate these constraints into the objective function of agent i with a Lagrange multiplier λ, to update the parameter η^i of the proposal policy:

$$\eta^i \leftarrow \eta^i + \nabla_{\eta^i} V^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s) + \lambda \nabla_{\eta^i} \sum_j \min\{0, \mathbb{E}_{\mathbf{m}\sim\boldsymbol{\phi}}[Q^j_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s, \mathbf{m})] - \mathbb{E}_{\mathbf{a}\sim\boldsymbol{\pi}}[Q^j_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s, \mathbf{a})]\}. \tag{7}$$

Note that when $\lambda = 0$, the proposal policies are not constrained by Equation (6). The abstract pseudocode of DCL is provided in Algorithm 1. Please refer to Appendix 11 for more details about DCL.

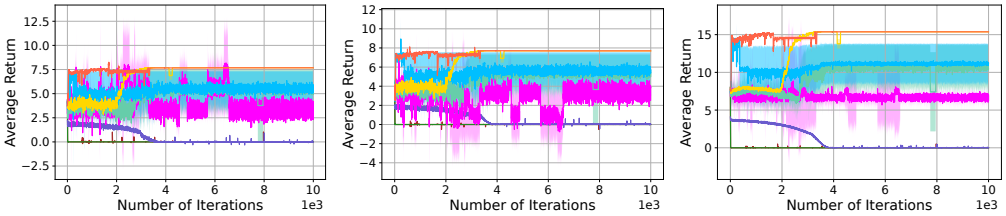
6 Experiments

We evaluated the performance of DCL focusing on two objectives. First, we investigated DCL's ability to foster cooperative behaviors among agents in challenging mixed-motive tasks. To validate this, we analyzed the behaviors of agents with mutual commitment and without commitment. Second, we compared DCL's efficiency against other multi-agent reinforcement learning algorithms in tabular, repeated, and sequential social dilemmas. We demonstrated improvements in both agents' self-interest optimization and social welfare. Additionally, we compared centralized (Algorithm 2, Appendix 11.1) and decentralized (Algorithm 3, Appendix 11.2) versions of DCL. Each algorithm was executed with and without incentive-compatible constraints (denoted as DCL-IC and DCL respectively), to further explore the impact of the constraints introduced in Equation (6).



(a) Agent 1 Return in Grid Game (b) Agent 2 Return in Grid Game (c) Social Welfare in Grid Game

Figure 4: Grid Game (Horizon=16): DCL v.s. Other Baselines.



(a) Agent 1 Return in RPC (b) Agent 2 Return in RPC (c) Social Welfare in RPC

Figure 5: Repeated Purely Conflicting Game (Horizon=16): DCL v.s. Other Baselines.

6.1 Baselines

We compared DCL with the following baselines. Each curve was averaged over 10 seeds with shaded regions indicating standard errors. Hyperparameters and more implementation details can be found in Appendix 13.

Independent PPO (IPPO) In this baseline, each agent was trained independently with the proximal policy optimization (PPO) (Schulman et al., 2017). The objective of each agent is maximizing individual expected returns. We implemented multi-agent independent PPO with RLlib (Liang et al., 2018).

Mediated MARL To compare with an altruistic third party mechanism, we implemented mediated multi-agent reinforcement learning using the code released by Ivanov et al. (2023). The mediator, whether constrained or unconstrained, was trained to maximize the utilitarian social welfare, i.e., the expected sum of all agents' returns, while other agents were trained independently to maximize their self-interests. Both agents and the mediator were optimized via actor-critic algorithms (Mnih et al., 2016).

Multi-Objective Contract Augmentation Learning (MOCA) To compare with a contract mechanism with reward transfer, we implemented multi-objective contract augmentation learning with the code released by Christoffersen (2024); Haupt et al. (2022). Each agent was trained to maximize self-interest, with a learnable transfer payment that directly modifies agents' rewards.

6.2 Results

6.2.1 Prisoner's Dilemma

Prisoner's Dilemma (Rapoport, 1965) is a normal form mixed-motive game, with payoff matrix in Table 1.

In accord with Proposition 4.1, Figure 2 shows that the DCL agents converge to mutual cooperation in the MCG with utilitarian social welfare -2. The fully decentralized DCL also converges to mutual cooperation, while having a larger oscillation before convergence (Figure 2). This behavior is expected since decentralized DCL estimates policies of other agents rather than directly accessing the true policies, which introduces biases, particularly in the early stages of training. These biases are gradually reduced as the estimated policies approach the actual policies over time. Figure 3 shows the policies of proposals, commitments and actions. Without mutual commitment, the probability of cooperation converges to 0. Whereas under the conditional commitment mechanism, the probabilities of proposing and committing to mutual cooperation converge to 1. This result aligns with our theoretical analysis in Proposition 12 and demonstrates the capability of commitment mechanism to achieve cooperation.

Table 1: Prisoner's Dilemma

	C	D
C	(-1,-1)	(-3,0)
D	(0,-3)	(-2,-2)

Mediated MARL with an unconstrained mediator shows the second-best performance, while constrained mediated MARL performs worse, failing to converge to either mutual cooperation or defection. This failure may arise from inaccurate value estimation in mediated MARL, which constrains the mediator's policy during training. Specifically, mediated MARL trains each agent with independent actor critic (Mnih et al., 2016), considering other agents as part of the environment, leading to nonstationarity from each agent's perspective. In contrast, DCL agents consider joint actions when evaluating future expected returns, avoiding conflicts with the stationary environment assumption in MCGs. Furthermore, the constrained mediated MARL dynamically updates the Lagrange multiplier, shifting the optimization objective at each timestep, which may lead to divergence.

The other baselines, MOCA and IPPO, converge to the mutual defection equilibrium after only a few iterations. Without mechanism design, mutual defection is the only Nash equilibrium in Prisoner's Dilemma, so it is expected that IPPO fails to achieve cooperation. Without a specific choice of contract space and hand-crafted rules, MOCA also fails to find a contract acceptable to all agents.

6.2.2 Grid Game

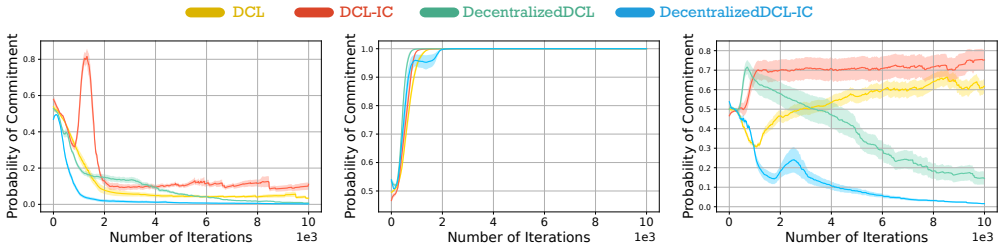
The above results show that DCL works well on a tabular social dilemma with a single state, we next extend the evaluation to sequential social dilemmas. We created a 2-player, T-step, N-grid game, where agent 1 starts at grid position $p_0^1 = 0$, and agent 2 starts at $p_0^2 = N - 1$. At each timestep, each player observes both agents' locations, $s_t = (p_t^1, p_t^2)$, and chooses between moving forward, $p_{t+1}^i = \min\{p_t^i + 1, N - 1\}$, or moving backward, $p_{t+1}^i = \max\{p_t^i - 1, 0\}$. Rewards are defined based on agents' positions: for agent 1, $r^1 = p^1 - 2(N - 1 - p^2)$; for agent 2, $r^2 = N - 1 - p^2 - 2p^1$. This grid game presents a social dilemma at every state. Agents benefit from cooperation by moving away from the other player's initial position, while the dominant strategy is to move towards the other's starting point. Figure 4 demonstrates that DCL agents gradually learn to cooperate, with zero accumulated discounted rewards. In contrast, other baselines fail to converge to such cooperative strategies.

6.2.3 Repeated Purely Conflicting Game

To investigate whether DCL can adapt effectively to scenarios with significant competition, we then introduced a purely conflicting game presented in Table 2. In this game, an increase in one agent's payoff always results in a decrease in the payoff of others. The dominant strategy of each agent is to play A_2 regardless of the opponent's action, which also holds true in finitely repeated versions (denoted as RPC). Under such conditions, agents have no opportunity to establish 1-step mutually beneficial agreements. As a result, all players receive zero payoff throughout episodes.

Table 2: Purely Conflicting Game

	A_1	A_2
A_1	(0,0)	(-1,2)
A_2	(2,-1)	(0,0)



(a) Commitment Policy of (C, D) (b) Commitment Policy of (D, C) (c) Commitment Policy of (D, D)

Figure 6: DCL Commitment Policies in Prisoner's Dilemma

However, if agents can commit to actions over multiple steps, both can achieve positive long-term returns by committing to a tit-for-tat agreement. To explore this, we extended DCL with mega-step commitments, enabling agents to commit to multi-step, mutually beneficial proposals. Our experiments show that DCL agents successfully converge to cooperative strategies $[(A_1, A_2), (A_2, A_1), ...]$ by alternating between A_1 and A_2 in multiple steps. While DCL agents make sacrifices at certain steps, they achieve significantly higher cumulative payoffs over the long run compared to other baselines (Figure 5), demonstrating DCL's adaptability to highly competitive environments.

7 Discussion on Experiments

7.1 Many-player Scenarios

In MCGs, the joint proposal space grows exponentially with the number of agents, which would inevitably increase the computational complexity. To investigate how DCL handles scalability with many players, we conducted additional experiments on an N-player public goods game (Marwell & Ames, 1981) with benefit factor 1.5, where the dominant strategy for each agent is to free-ride by not contributing to the public pool. The results demonstrate that DCL with incentive-compatible constraints performs effectively across scenarios with 2, 3, 5, and 10 agents, achieving high social welfare. Most agents converge to propose contributions and commit to joint proposals that result in positive individual welfare. These findings indicate that DCL scales well to many-player games, with the agreement rate of joint proposals remaining stable (> 0.99) as the number of agents increases. We report runtime, average joint proposal agreement rate and average social welfare per batch (batch size =256) across 5 random seeds in Table 8, Appendix 14.

7.2 Robustness to Maliciously Irrational Agents

As shown in Figure 3c and Figure 6, DCL agents converge to commitment policies that accept proposals for mutual cooperation and self-defection when the co-player cooperates, while rejecting cooperation when the co-player proposes defection in the Prisoner's Dilemma. Consequently, when interacting with irrational agents—such as those who always propose defection—DCL agents will reject such proposals and choose to defect following their action policies (Figure 3a). This demonstrates the robustness of DCL agents against malicious agents, as they effectively reject disadvantageous agreements and act in their own best interests.

8 Conclusion

We introduced the Markov Commitment Games, a framework that allows self-interested agents to negotiate future plans through voluntary commitments. It responds to the open problem in cooperative AI (Dafoe et al., 2020) on commitment capabilities without relying on altruism. We derived unbiased proposal, commitment, and action policy gradients (Lemma 5.1), which facilitates the design of policy updates while preserving the stationarity assumption of the multi-agent environment.

Under the framework of MCGs, we proposed differentiable commitment learning (DCL), which maximizes agents' expected self-interests while incorporating incentive-compatible constraints on their proposal policies to encourage mutually beneficial agreements. DCL also mitigates limitations of non-stationary training of existing methods. Rather than treating other agents as part of a stationary environment—a simplification that does not hold in multi-agent settings—DCL explicitly leverages other agents' actions when estimating future expected values. This approach enhances the accuracy of value estimations and promotes stability during training. We empirically showed that our method outperforms the baseline methods in multiple tasks, often by successfully facilitating cooperation among agents. We also demonstrated the efficacy of DCL in its fully decentralized implementation.

9 Limitations and Future Work

Sample Efficiency Both centralized and decentralized versions of DCL employ on-policy updates for agents' actors and critics, which explores by sampling actions according to the current policy models. This is less sample efficient compared to off-policy methods, which use past trajectories from a replay buffer for model updates. However, off-policy methods may bring biases due to discrepancies between the behavior policy and the target policy. While importance sampling can mitigate this issue by re-weighting experiences, it may also introduce high variance, especially when policies diverge significantly. Furthermore, in fully decentralized DCL, agents do not have access to other agents' policies, and importance sampling based on estimations of other agents' policies may introduce additional biases. Therefore, the trade-off between the sample efficiency, bias and variance can be further explored in our future work.

Complex Proposal Domain In DCL and MCGs, the proposal domain is formulated as a set of future actions. This reflects real-world scenarios, where agreements often specify future actions conditioned on the behavior of other parties. Nevertheless, human commitments can take various forms, such as stochastic policies of future plan. Extending our framework to accommodate more complex proposal domains presents a promising direction for future research.

Acknowledgements

We acknowledge funding from the Canada CIFAR AI Chair program, a discovery grant from the Natural Sciences and Engineering Research Council of Canada and a grant from IITP & MSIT of Korea (No. RS-2024-00457882, AI Research Hub Project). Computational resources used in preparing this research were provided, in part, by the Province of Ontario, the Government of Canada through CIFAR, and companies sponsoring the Vector Institute https://vectorinstitute.ai/partners/. Baoxiang Wang was sponsored by the Vector Institute's visiting researcher program.

References

Ken Binmore, Ariel Rubinstein, and Asher Wolinsky. The nash bargaining solution in economic modelling. *The RAND Journal of Economics*, pp. 176–188, 1986.

Gharad Bryan, Dean Karlan, and Scott Nelson. Commitment devices. *Annu. Rev. Econ.*, 2(1): 671–698, 2010.

Yair Censor. Pareto optimality in multiobjective problems. *Applied Mathematics and Optimization*, 4(1):41–59, 1977.

Phillip Johannes Kerr Christoffersen. *Mitigating Social Dilemmas in Multi-Agent Reinforcement Learning with Formal Contracting*. PhD thesis, Massachusetts Institute of Technology, 2024.

Allan Dafoe, Edward Hughes, Yoram Bachrach, Tantum Collins, Kevin R McKee, Joel Z Leibo, Kate Larson, and Thore Graepel. Open problems in cooperative ai. *arXiv preprint arXiv:2012.08630*, 2020.

Dave De Jonge and Dongmo Zhang. Strategic negotiations for extensive-form games. *Autonomous Agents and Multi-Agent Systems*, 34:1–41, 2020.

Anthony DiGiovanni and Jesse Clifton. Commitment games with conditional information disclosure. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 5616–5623, 2023.

Shaddin Dughmi. Algorithmic information structure design: A survey. *ACM SIGecom Exchanges*, 15(2):2–24, 2017.

James D Fearon. Rationalist explanations for war. *International organization*, 49(3):379–414, 1995.

Jakob Foerster, Ioannis Alexandros Assael, Nando De Freitas, and Shimon Whiteson. Learning to communicate with deep multi-agent reinforcement learning. *Advances in neural information processing systems*, 29, 2016.

Francoise Forges. A folk theorem for bayesian games with commitment. *Games and Economic Behavior*, 78:64–71, 2013.

Drew Fudenberg. *Game theory*. MIT press, 1991.

Amy Greenwald, Keith Hall, Roberto Serrano, et al. Correlated q-learning. In *ICML*, volume 3, pp. 242–249, 2003.

The Anh Han, Luis Moniz Pereira, and Tom Lenaerts. Evolution of commitment and level of participation in public goods games. *Autonomous Agents and Multi-Agent Systems*, 31(3):561–583, 2017.

Andreas A Haupt, Phillip JK Christoffersen, Mehul Damani, and Dylan Hadfield-Menell. Formal contracts mitigate social dilemmas in multi-agent rl. *arXiv e-prints*, pp. arXiv–2208, 2022.

Junling Hu and Michael P Wellman. Nash q-learning for general-sum stochastic games. *Journal of machine learning research*, 4(Nov):1039–1069, 2003.

Edward Hughes, Thomas W Anthony, Tom Eccles, Joel Z Leibo, David Balduzzi, and Yoram Bachrach. Learning to resolve alliance dilemmas in many-player zero-sum games. *arXiv preprint arXiv:2003.00799*, 2020.

Dmitry Ivanov, Ilya Zisman, and Kirill Chernyshev. Mediated multi-agent reinforcement learning. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*, pp. 49–57, 2023.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

Natasha Jaques, Angeliki Lazaridou, Edward Hughes, Caglar Gulcehre, Pedro Ortega, DJ Strouse, Joel Z Leibo, and Nando De Freitas. Social influence as intrinsic motivation for multi-agent deep reinforcement learning. In *International conference on machine learning*, pp. 3040–3049. PMLR, 2019.

Adam Tauman Kalai, Ehud Kalai, Ehud Lehrer, and Dov Samet. A commitment folk theorem. *Games and Economic Behavior*, 69(1):127–137, 2010.

Emir Kamenica. Bayesian persuasion and information design. *Annual Review of Economics*, 11: 249–272, 2019.

Woojun Kim, Jongeui Park, and Youngchul Sung. Communication in multi-agent reinforcement learning: Intention sharing. In *International Conference on Learning Representations*, 2020.

Sachin Konan, Esmaeil Seraj, and Matthew Gombolay. Iterated reasoning with mutual information in cooperative and byzantine decentralized teaming. *arXiv preprint arXiv:2201.08484*, 2022.

János Kramár, Tom Eccles, Ian Gemp, Andrea Tacchetti, Kevin R McKee, Mateusz Malinowski, Thore Graepel, and Yoram Bachrach. Negotiation and honesty in artificial intelligence methods for the board game of diplomacy. *Nature Communications*, 13(1):7214, 2022.

Eric Liang, Richard Liaw, Robert Nishihara, Philipp Moritz, Roy Fox, Ken Goldberg, Joseph Gonzalez, Michael Jordan, and Ion Stoica. Rllib: Abstractions for distributed reinforcement learning. In *International conference on machine learning*, pp. 3053–3062. PMLR, 2018.

Yue Lin, Wenhao Li, Hongyuan Zha, and Baoxiang Wang. Information design in multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 36, 2024.

Gerald Marwell and Ruth E Ames. Economists free ride, does anyone else?: Experiments on the provision of public goods, iv. *Journal of public economics*, 15(3):295–310, 1981.

Stephen McAleer, John Lanier, Michael Dennis, Pierre Baldi, and Roy Fox. Improving social welfare while preserving autonomy via a pareto mediator. *arXiv preprint arXiv:2106.03927*, 2021.

Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pp. 1928–1937. PMLR, 2016.

Robert Powell. War as a commitment problem. *International organization*, 60(1):169–203, 2006.

Arotol Rapoport. Prisoner's dilemma: a study in conflict and cooperation, 1965.

Ludovic Renou. Commitment games. *Games and Economic Behavior*, 66(1):488–505, 2009.

Todd Rogers, Katherine L. Milkman, and Kevin G. Volpp. Commitment devices: Using initiatives to change behavior. *JAMA*, 311(20):2065–2066, 2014.

Tuomas W Sandholm and Victor R Lesser. Advantages of a leveled commitment contracting protocol. In *AAAI/IAAI, Vol. 1*, pp. 126–133. Citeseer, 1996.

Thomas C Schelling. *The Strategy of Conflict: with a new Preface by the Author*. Harvard university press, 1980.

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

Eric Sodomka, Elizabeth Hilliard, Michael Littman, and Amy Greenwald. Coco-q: Learning in stochastic games with side payments. In *International Conference on Machine Learning*, pp. 1471–1479. PMLR, 2013.

Sainbayar Sukhbaatar, Rob Fergus, et al. Learning multiagent communication with backpropagation. *Advances in neural information processing systems*, 29, 2016.

Xinyuan Sun, Davide Crapis, Matt Stephenson, and Jonathan Passerat-Palmbach. Cooperative ai via decentralized commitment devices. In *Multi-Agent Security Workshop@ NeurIPS'23*, 2023.

Richard S Sutton. Reinforcement learning: An introduction. *A Bradford Book*, 2018.

Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 12, 1999.

Ina Taneva. Information design. *American Economic Journal: Microeconomics*, 11(4):151–85, 2019.

Tonghan Wang, Paul Duetting, Dmitry Ivanov, Inbal Talgam-Cohen, and David C Parkes. Deep contract design via discontinuous networks. *Advances in Neural Information Processing Systems*, 36, 2024.

Ying Wen, Yaodong Yang, Rui Luo, Jun Wang, and Wei Pan. Probabilistic recursive reasoning for multi-agent reinforcement learning. *arXiv preprint arXiv:1901.09207*, 2019.

Supplementary Materials

The following content was not necessarily subject to peer review.

10 Proof of Lemma 5.1

The proof of Lemma 5.1 derives the action, commitment, and proposal policy gradients in DCL. Recall that the state value function (the objective function of self-interested agents) in MCGs is:

$$V^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s) = \mathbb{E}_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}[\sum_{k=t}^{\infty} \gamma^{k-t} r^i_{k+1} | s_t = s]. \tag{8}$$

The state-action value function is:

$$Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s, \mathbf{a}) = \mathbb{E}_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}[\sum_{k=t}^{\infty} \gamma^{k-t} r^i_{k+1} | s_t = s, \mathbf{a}_t = \mathbf{a}]. \tag{9}$$

Therefore we can expand the state value function by:

$$\begin{aligned} V^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s) = \sum_{\mathbf{m}\sim\boldsymbol{\phi}} \boldsymbol{\phi}(\mathbf{m}|s) \sum_{\mathbf{c}\sim\boldsymbol{\psi}} \boldsymbol{\psi}(\mathbf{c}|s,\mathbf{m}) \sum_{\mathbf{a}\sim\boldsymbol{\pi}} \boldsymbol{\pi}(\mathbf{a}|s) \Bigg[& \mathbb{1}(\mathbf{c}=\mathbf{1}) Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s,\mathbf{m}) \\ & + \Big(1 - \mathbb{1}(\mathbf{c}=\mathbf{1})\Big) Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s,\mathbf{a}) \Bigg]. \end{aligned} \tag{10}$$

We then derive policy gradients based on the state-action value function and policy functions.

10.1 Unconstrained Policy Gradient

Proof. First, we consider the action policy gradient $\nabla_{\theta^i} V^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s)$ for each agent $i \in \mathcal{N}$:

$$\begin{aligned} & \nabla_{\theta^i} V^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s) \\ = & \sum_{\mathbf{m}\sim\boldsymbol{\phi}} \boldsymbol{\phi}(\mathbf{m}|s) \sum_{\mathbf{c}\sim\boldsymbol{\psi}} \boldsymbol{\psi}(\mathbf{c}|s,\mathbf{m}) \Bigg[\mathbb{1}(\mathbf{c}=\mathbf{1}) \nabla_{\theta^i} Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s,\mathbf{m}) + \Big(1 - \mathbb{1}(\mathbf{c}=\mathbf{1})\Big) \\ & \sum_{\mathbf{a}\sim\boldsymbol{\pi}} Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s,\mathbf{a}) \nabla_{\theta^i} \boldsymbol{\pi}(\mathbf{a}|s) + \boldsymbol{\pi}(\mathbf{a}|s) \nabla_{\theta^i} Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s,\mathbf{a}) \Bigg], \\ = & \sum_{\mathbf{m}\sim\boldsymbol{\phi}} \boldsymbol{\phi}(\mathbf{m}|s) \sum_{\mathbf{c}\sim\boldsymbol{\psi}} \boldsymbol{\psi}(\mathbf{c}|s,\mathbf{m}) \Bigg[\Big(1 - \mathbb{1}(\mathbf{c}=\mathbf{1})\Big) \sum_{\mathbf{a}\sim\boldsymbol{\pi}} Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s,\mathbf{a}) \nabla_{\theta^i} \boldsymbol{\pi}(\mathbf{a}|s) \Bigg] \\ & + \sum_{\mathbf{m}\sim\boldsymbol{\phi}} \boldsymbol{\phi}(\mathbf{m}|s) \sum_{\mathbf{c}\sim\boldsymbol{\psi}} \boldsymbol{\psi}(\mathbf{c}|s,\mathbf{m}) \cdot \Bigg[\mathbb{1}(\mathbf{c}=\mathbf{1}) \nabla_{\theta^i} Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s,\mathbf{m}) \\ & + \Big(1 - \mathbb{1}(\mathbf{c}=\mathbf{1})\Big) \sum_{\mathbf{a}\sim\boldsymbol{\pi}} \boldsymbol{\pi}(\mathbf{a}|s) \nabla_{\theta^i} Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s,\mathbf{a}) \Bigg]. \end{aligned} \tag{11}$$

Let

$$f_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s) = \sum_{\mathbf{m}\sim\boldsymbol{\phi}} \boldsymbol{\phi}(\mathbf{m}|s) \sum_{\mathbf{c}\sim\boldsymbol{\psi}} \boldsymbol{\psi}(\mathbf{c}|s,\mathbf{m}) \Bigg[\Big(1 - \mathbb{1}(\mathbf{c}=\mathbf{1})\Big) \sum_{\mathbf{a}\sim\boldsymbol{\pi}} Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s,\mathbf{a}) \nabla_{\theta^i} \boldsymbol{\pi}(\mathbf{a}|s) \Bigg].$$

We have

$$\begin{aligned}\nabla_{\theta^i} V^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s) =& f_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s) + \sum_{\mathbf{m}\sim\boldsymbol{\phi}} \boldsymbol{\phi}(\mathbf{m}|s) \sum_{\mathbf{c}\sim\boldsymbol{\psi}} \boldsymbol{\psi}(\mathbf{c}|s,\mathbf{m}) \Bigg[\mathbb{1}(\mathbf{c}=\mathbf{1}) \nabla_{\theta^i} Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s,\mathbf{m}) \\ &+ \Big(1-\mathbb{1}(\mathbf{c}=\mathbf{1})\Big) \cdot \sum_{\mathbf{a}\sim\boldsymbol{\pi}} \boldsymbol{\pi}(\mathbf{a}|s) \nabla_{\theta^i} Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s,\mathbf{a}) \Bigg].\end{aligned} \tag{12}$$

Since $Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s,\mathbf{a}) = R^i(s,\mathbf{a}) + \gamma \sum_{s'} p(s'|s,\mathbf{a}) V^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s')$, we obtain

$$\nabla_{\theta^i} Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s,\mathbf{a}) = \nabla_{\theta^i} \Big(R^i(s,\mathbf{a}) + \gamma \sum_{s'} p(s'|s,\mathbf{a}) V^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s') \Big) = \gamma \sum_{s'} p(s'|s,\mathbf{a}) \nabla_{\theta^i} V^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s'). \tag{13}$$

Therefore,

$$\begin{aligned}&\nabla_{\theta^i} V^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s) \\ =& f_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s) + \gamma \sum_{\mathbf{m}\sim\boldsymbol{\phi}} \boldsymbol{\phi}(\mathbf{m}|s) \sum_{\mathbf{c}\sim\boldsymbol{\psi}} \boldsymbol{\psi}(\mathbf{c}|s,\mathbf{m}) \Bigg[\mathbb{1}(\mathbf{c}=\mathbf{1}) \sum_{s'} p(s'|s,\mathbf{m}) \nabla_{\theta^i} V^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s') + \Big(1-\mathbb{1}(\mathbf{c}=\mathbf{1})\Big) \\ &\cdot \sum_{\mathbf{a}\sim\boldsymbol{\pi}} \boldsymbol{\pi}(\mathbf{a}|s) \sum_{s'} p(s'|s,\mathbf{a}) \nabla_{\theta^i} V^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s') \Bigg].\end{aligned} \tag{14}$$

Define $d_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s,s',k)$ as the probability of transitioning from state s to state s' in k steps under $\boldsymbol{\phi}, \boldsymbol{\psi}, \boldsymbol{\pi}$, then we have

$$d_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s,s',1) = \sum_{\mathbf{m}\sim\boldsymbol{\phi}} \boldsymbol{\phi}(\mathbf{m}|s) \sum_{\mathbf{c}\sim\boldsymbol{\psi}} \boldsymbol{\psi}(\mathbf{c}|s,\mathbf{m}) \Bigg[\mathbb{1}(\mathbf{c}=\mathbf{1}) p(s'|s,\mathbf{m}) + \Big(1-\mathbb{1}(\mathbf{c}=\mathbf{1})\Big) \sum_{\mathbf{a}\sim\boldsymbol{\pi}} \boldsymbol{\pi}(\mathbf{a}|s) p(s'|s,\mathbf{a}) \Bigg], \tag{15}$$

and

$$d_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s,s',k+1) = \sum_{x} d_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s,x,k) d_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(x,s',1). \tag{16}$$

Note

$$d_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s,s,0) = \sum_{x} d_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s,x,0) = 1. \tag{17}$$

Then,

$$\begin{aligned}&\nabla_{\theta^i} V^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s) \\ =& f_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s) + \gamma \sum_{s'} \sum_{\mathbf{m}\sim\boldsymbol{\phi}} \boldsymbol{\phi}(\mathbf{m}|s) \sum_{\mathbf{c}\sim\boldsymbol{\psi}} \boldsymbol{\psi}(\mathbf{c}|s,\mathbf{m}) \Bigg[\mathbb{1}(\mathbf{c}=\mathbf{1}) p(s'|s,\mathbf{m}) + \Big(1-\mathbb{1}(\mathbf{c}=\mathbf{1})\Big) \sum_{\mathbf{a}\sim\boldsymbol{\pi}} \boldsymbol{\pi}(\mathbf{a}|s) p(s'|s,\mathbf{a}) \Bigg] \\ &\cdot \nabla_{\theta^i} V^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s'), \\ =& f_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s) + \gamma \sum_{s'} d_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s,s',1) \nabla_{\theta^i} V^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s').\end{aligned} \tag{18}$$

By induction,

$$\begin{aligned}&\nabla_{\theta^i} V^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s) \\ =& f_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s) + \gamma \sum_{s'} d_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s,s',1) \Big(f_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s') + \gamma \sum_{s''} d_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s',s'',1) \nabla_{\theta^i} V^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s'') \Big), \\ =& f_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s) + \gamma \sum_{s'} d_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s,s',1) f_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s') + \gamma^2 \sum_{s''} d_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s,s'',2) \nabla_{\theta^i} V^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s''), \\ =& \sum_{x\in\mathcal{S}} \sum_{k=0}^{\infty} \gamma^k d_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s,x,k) f_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(x).\end{aligned} \tag{19}$$

Then we define a stationary distribution $\rho_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(x) = \frac{\sum_{k=0}^{\infty}\gamma^k d_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s,x,k)}{\sum_{x\in\mathcal{S}}\sum_{k=0}^{\infty}\gamma^k d_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s,x,k)}$, also known as an occupancy measure of $\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}$. Thus,

$$
\begin{aligned}
&\nabla_{\theta^i} V^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s)\\
\propto&\sum_{x\in\mathcal{S}}\rho_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(x) f_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(x),\\
=&\sum_{x\in\mathcal{S}}\rho_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(x)\sum_{\mathbf{m}\sim\boldsymbol{\phi}}\boldsymbol{\phi}(\mathbf{m}|x)\sum_{\mathbf{c}\sim\boldsymbol{\psi}}\boldsymbol{\psi}(\mathbf{c}|x,\mathbf{m})\left[\Big(1-\mathbb{1}(\mathbf{c}=\mathbf{1})\Big)\sum_{\mathbf{a}\sim\boldsymbol{\pi}}Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(x,\mathbf{a})\nabla_{\theta^i}\boldsymbol{\pi}(\mathbf{a}|x)\right],\\
=&\sum_{x\in\mathcal{S}}\rho_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(x)\sum_{\mathbf{m}\sim\boldsymbol{\phi}}\boldsymbol{\phi}(\mathbf{m}|x)\sum_{\mathbf{c}\sim\boldsymbol{\psi}}\boldsymbol{\psi}(\mathbf{c}|x,\mathbf{m})\sum_{\mathbf{a}\sim\boldsymbol{\pi}}\boldsymbol{\pi}(\mathbf{a}|x)\left[\Big(1-\mathbb{1}(\mathbf{c}=\mathbf{1})\Big)Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(x,\mathbf{a})\nabla_{\theta^i}\log\boldsymbol{\pi}(\mathbf{a}|x)\right],\\
=&\mathbb{E}_{x\sim\rho_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}},\mathbf{m}\sim\boldsymbol{\phi},\mathbf{c}\sim\boldsymbol{\psi},\mathbf{a}\sim\boldsymbol{\pi}}\left[\Big(1-\mathbb{1}(\mathbf{c}=\mathbf{1})\Big)Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(x,\mathbf{a})\nabla_{\theta^i}\log\boldsymbol{\pi}(\mathbf{a}|x)\right],\\
=&\mathbb{E}_{x\sim\rho_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}},\mathbf{m}\sim\boldsymbol{\phi},\mathbf{c}\sim\boldsymbol{\psi},\mathbf{a}\sim\boldsymbol{\pi}}\left[\Big(1-\mathbb{1}(\mathbf{c}=\mathbf{1})\Big)Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(x,\mathbf{a})\nabla_{\theta^i}\log\pi^i(a^i|x)\right].
\end{aligned}
\tag{20}
$$

Therefore, we have

$$
\nabla_{\theta^i} V^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s)\propto\mathbb{E}_{x\sim\rho_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}},\mathbf{m}\sim\boldsymbol{\phi},\mathbf{c}\sim\boldsymbol{\psi},\mathbf{a}\sim\boldsymbol{\pi}}\left[\Big(1-\mathbb{1}(\mathbf{c}=\mathbf{1})\Big)Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(x,\mathbf{a})\nabla_{\theta^i}\log\pi^i(a^i|x)\right].\qquad\square
$$

10.2 Commitment Network Gradient

Proof. Next, we consider commitment policy gradient $\nabla_{\zeta^i} V^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s)$:

$$
\begin{aligned}
&\nabla_{\zeta^i} V^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s)\\
=&\nabla_{\zeta^i}\sum_{\mathbf{m}\sim\boldsymbol{\phi}}\boldsymbol{\phi}(\mathbf{m}|s)\sum_{\mathbf{c}\sim\boldsymbol{\psi}}\boldsymbol{\psi}(\mathbf{c}|s,\mathbf{m})\left[\mathbb{1}(\mathbf{c}=\mathbf{1})Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s,\mathbf{m})+\Big(1-\mathbb{1}(\mathbf{c}=\mathbf{1})\Big)\sum_{\mathbf{a}\sim\boldsymbol{\pi}}\boldsymbol{\pi}(\mathbf{a}|s)Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s,\mathbf{a})\right],\\
=&\sum_{\mathbf{m}\sim\boldsymbol{\phi}}\boldsymbol{\phi}(\mathbf{m}|s)\sum_{\mathbf{c}\sim\boldsymbol{\psi}}\left[\mathbb{1}(\mathbf{c}=\mathbf{1})Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s,\mathbf{m})+\Big(1-\mathbb{1}(\mathbf{c}=\mathbf{1})\Big)\sum_{\mathbf{a}\sim\boldsymbol{\pi}}\boldsymbol{\pi}(\mathbf{a}|s)Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s,\mathbf{a})\right]\nabla_{\zeta^i}\boldsymbol{\psi}(\mathbf{c}|s,\mathbf{m})\\
&+\sum_{\mathbf{m}\sim\boldsymbol{\phi}}\boldsymbol{\phi}(\mathbf{m}|s)\sum_{\mathbf{c}\sim\boldsymbol{\psi}}\boldsymbol{\psi}(\mathbf{c}|s,\mathbf{m})\nabla_{\zeta^i}\left[\mathbb{1}(\mathbf{c}=\mathbf{1})Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s,\mathbf{m})+\Big(1-\mathbb{1}(\mathbf{c}=\mathbf{1})\Big)\sum_{\mathbf{a}\sim\boldsymbol{\pi}}\boldsymbol{\pi}(\mathbf{a}|s)Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s,\mathbf{a})\right],\\
=&\sum_{\mathbf{m}\sim\boldsymbol{\phi}}\boldsymbol{\phi}(\mathbf{m}|s)\sum_{\mathbf{c}\sim\boldsymbol{\psi}}\left[\mathbb{1}(\mathbf{c}=\mathbf{1})Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s,\mathbf{m})+\Big(1-\mathbb{1}(\mathbf{c}=\mathbf{1})\Big)\sum_{\mathbf{a}\sim\boldsymbol{\pi}}\boldsymbol{\pi}(\mathbf{a}|s)Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s,\mathbf{a})\right]\nabla_{\zeta^i}\boldsymbol{\psi}(\mathbf{c}|s,\mathbf{m})\\
&+\sum_{\mathbf{m}\sim\boldsymbol{\phi}}\boldsymbol{\phi}(\mathbf{m}|s)\sum_{\mathbf{c}\sim\boldsymbol{\psi}}\boldsymbol{\psi}(\mathbf{c}|s,\mathbf{m})\left[Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s,\mathbf{m})-\sum_{\mathbf{a}\sim\boldsymbol{\pi}}\boldsymbol{\pi}(\mathbf{a}|s)Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s,\mathbf{a})\right]\nabla_{\zeta^i}\mathbb{1}(\mathbf{c}=\mathbf{1})\\
&+\sum_{\mathbf{m}\sim\boldsymbol{\phi}}\boldsymbol{\phi}(\mathbf{m}|s)\sum_{\mathbf{c}\sim\boldsymbol{\psi}}\boldsymbol{\psi}(\mathbf{c}|s,\mathbf{m})\left[\mathbb{1}(\mathbf{c}=\mathbf{1})\nabla_{\zeta^i}Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s,\mathbf{m})+\Big(1-\mathbb{1}(\mathbf{c}=\mathbf{1})\Big)\sum_{\mathbf{a}\sim\boldsymbol{\pi}}\boldsymbol{\pi}(\mathbf{a}|s)\nabla_{\zeta^i}Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s,\mathbf{a})\right].
\end{aligned}
\tag{21}
$$

Let

$$
\begin{aligned}
g_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s)=&\sum_{\mathbf{m}\sim\boldsymbol{\phi}}\boldsymbol{\phi}(\mathbf{m}|s)\sum_{\mathbf{c}\sim\boldsymbol{\psi}}\left[\mathbb{1}(\mathbf{c}=\mathbf{1})Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s,\mathbf{m})+\Big(1-\mathbb{1}(\mathbf{c}=\mathbf{1})\Big)\sum_{\mathbf{a}\sim\boldsymbol{\pi}}\boldsymbol{\pi}(\mathbf{a}|s)Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s,\mathbf{a})\right]\nabla_{\zeta^i}\boldsymbol{\psi}(\mathbf{c}|s,\mathbf{m})\\
&+\sum_{\mathbf{m}\sim\boldsymbol{\phi}}\boldsymbol{\phi}(\mathbf{m}|s)\sum_{\mathbf{c}\sim\boldsymbol{\psi}}\boldsymbol{\psi}(\mathbf{c}|s,\mathbf{m})\left[Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s,\mathbf{m})-\sum_{\mathbf{a}\sim\boldsymbol{\pi}}\boldsymbol{\pi}(\mathbf{a}|s)Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s,\mathbf{a})\right]\nabla_{\zeta^i}\mathbb{1}(\mathbf{c}=\mathbf{1}).
\end{aligned}
$$

Then,

$$
\begin{aligned}
&\nabla_{\zeta^i} V^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s) \\
=&g_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s) + \sum_{\mathbf{m}\sim\boldsymbol{\phi}} \boldsymbol{\phi}(\mathbf{m}|s) \sum_{\mathbf{c}\sim\boldsymbol{\psi}} \boldsymbol{\psi}(\mathbf{c}|s,\mathbf{m}) \Bigg[\mathbb{1}(\mathbf{c}=\mathbf{1}) \nabla_{\zeta^i} Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s,\mathbf{m}) + \Big(1-\mathbb{1}(\mathbf{c}=\mathbf{1})\Big) \sum_{\mathbf{a}\sim\boldsymbol{\pi}} \boldsymbol{\pi}(\mathbf{a}|s) \\
&\cdot \nabla_{\zeta^i} Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s,\mathbf{a}) \Bigg], \\
=&g_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s) + \gamma \sum_{\mathbf{m}\sim\boldsymbol{\phi}} \boldsymbol{\phi}(\mathbf{m}|s) \sum_{\mathbf{c}\sim\boldsymbol{\psi}} \boldsymbol{\psi}(\mathbf{c}|s,\mathbf{m}) \Bigg[\mathbb{1}(\mathbf{c}=\mathbf{1}) \sum_{s'} p(s'|s,\mathbf{m}) \nabla_{\zeta^i} V^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s') + \Big(1-\mathbb{1}(\mathbf{c}=\mathbf{1})\Big) \\
&\cdot \sum_{\mathbf{a}\sim\boldsymbol{\pi}} \boldsymbol{\pi}(\mathbf{a}|s) \sum_{s'} p(s'|s,\mathbf{a}) \nabla_{\zeta^i} V^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s') \Bigg].
\end{aligned} \tag{22}
$$

According to (15),

$$
\nabla_{\zeta^i} V^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s) = g_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s) + \gamma \sum_{s'} d_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s,s',1) \nabla_{\zeta^i} V^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s'). \tag{23}
$$

Similarly by induction,

$$
\begin{aligned}
&\nabla_{\zeta^i} V^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s) \\
=&\sum_{x\in\mathcal{S}} \sum_{k=0}^{\infty} \gamma^k d_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s,x,k) g_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(x), \\
\propto&\sum_{x\in\mathcal{S}} \rho_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(x) g_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(x), \\
=&\sum_{x\in\mathcal{S}} \rho_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(x) \Bigg[\sum_{\mathbf{m}\sim\boldsymbol{\phi}} \boldsymbol{\phi}(\mathbf{m}|x) \sum_{\mathbf{c}\sim\boldsymbol{\psi}} \Big[\mathbb{1}(\mathbf{c}=\mathbf{1}) Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(x,\mathbf{m}) + \Big(1-\mathbb{1}(\mathbf{c}=\mathbf{1})\Big) \sum_{\mathbf{a}\sim\boldsymbol{\pi}} \boldsymbol{\pi}(\mathbf{a}|x) Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(x,\mathbf{a}) \Big] \\
&\cdot \nabla_{\zeta^i} \boldsymbol{\psi}(\mathbf{c}|x,\mathbf{m}) + \sum_{\mathbf{m}\sim\boldsymbol{\phi}} \boldsymbol{\phi}(\mathbf{m}|x) \sum_{\mathbf{c}\sim\boldsymbol{\psi}} \boldsymbol{\psi}(\mathbf{c}|x,\mathbf{m}) \Big[Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(x,\mathbf{m}) - \sum_{\mathbf{a}\sim\boldsymbol{\pi}} \boldsymbol{\pi}(\mathbf{a}|x) Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(x,\mathbf{a}) \Big] \nabla_{\zeta^i} \mathbb{1}(\mathbf{c}=\mathbf{1}) \Bigg], \\
=&\mathbb{E}_{x\sim\rho_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}},\mathbf{m}\sim\boldsymbol{\phi},\mathbf{c}\sim\boldsymbol{\psi},\mathbf{a}\sim\boldsymbol{\pi}} \Bigg[\Big[\mathbb{1}(\mathbf{c}=\mathbf{1}) Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(x,\mathbf{m}) + \Big(1-\mathbb{1}(\mathbf{c}=\mathbf{1})\Big) Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(x,\mathbf{a}) \Big] \nabla_{\zeta^i} \log \boldsymbol{\psi}(\mathbf{c}|x,\mathbf{m}) \\
&+ \Big[Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(x,\mathbf{m}) - Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(x,\mathbf{a}) \Big] \nabla_{\zeta^i} \mathbb{1}(\mathbf{c}=\mathbf{1}) \Bigg], \\
=&\mathbb{E}_{x\sim\rho_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}},\mathbf{m}\sim\boldsymbol{\phi},\mathbf{c}\sim\boldsymbol{\psi},\mathbf{a}\sim\boldsymbol{\pi}} \Bigg[\Big[\mathbb{1}(\mathbf{c}=\mathbf{1}) Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(x,\mathbf{m}) + \Big(1-\mathbb{1}(\mathbf{c}=\mathbf{1})\Big) Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(x,\mathbf{a}) \Big] \nabla_{\zeta^i} \log \psi^i(c^i|x,\mathbf{m}) \\
&+ \Big[Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(x,\mathbf{m}) - Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(x,\mathbf{a}) \Big] \prod_{k\neq i} \mathbb{1}(c^k=1) \nabla_{\zeta^i} \mathbb{1}(c^i=1) \Bigg].
\end{aligned} \tag{24}
$$

Therefore,

$$
\begin{aligned}
\nabla_{\zeta^i} V^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s) \propto&\mathbb{E}_{x\sim\rho_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}},\mathbf{m}\sim\boldsymbol{\phi},\mathbf{c}\sim\boldsymbol{\psi},\mathbf{a}\sim\boldsymbol{\pi}} \Bigg[\Big[\mathbb{1}(\mathbf{c}=\mathbf{1}) Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(x,\mathbf{m}) + \Big(1-\mathbb{1}(\mathbf{c}=\mathbf{1})\Big) Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(x,\mathbf{a}) \Big] \\
&\cdot \nabla_{\zeta^i} \log \psi^i(c^i|x,\mathbf{m}) + \Big[Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(x,\mathbf{m}) - Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(x,\mathbf{a}) \Big] \prod_{k\neq i} \mathbb{1}(c^k=1) \nabla_{\zeta^i} \mathbb{1}(c^i=1) \Bigg].
\end{aligned}
$$

□

Note that $\nabla_{\zeta^i}\mathbb{1}(c^i = 1) = \frac{d\mathbb{1}(c^i=1)}{dc^i}\frac{\partial c^i}{\partial \zeta^i}$. To compute $\frac{\partial c^i}{\partial \zeta^i}$, we apply the Gumbel-Softmax distribution (Jang et al., 2016) for differentiable sampling. This allows backpropagation through the differentiable commitment sample c^i for $\forall i \in \mathcal{N}$.

10.3 Proposing Network Gradient

Proof. Finally, we consider the proposal policy gradient $\nabla_{\eta^i} V^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s)$:

$$
\begin{aligned}
&\nabla_{\eta^i} V^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s)\\
=&\nabla_{\eta^i}\sum_{\mathbf{m}\sim\boldsymbol{\phi}}\boldsymbol{\phi}(\mathbf{m}|s)\sum_{\mathbf{c}\sim\boldsymbol{\psi}}\boldsymbol{\psi}(\mathbf{c}|s,\mathbf{m})\Bigg[\mathbb{1}(\mathbf{c}=\mathbf{1})Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s,\mathbf{m}) + \Big(1-\mathbb{1}(\mathbf{c}=\mathbf{1})\Big)\sum_{\mathbf{a}\sim\boldsymbol{\pi}}\boldsymbol{\pi}(\mathbf{a}|s)Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s,\mathbf{a})\Bigg],\\
=&\sum_{\mathbf{m}\sim\boldsymbol{\phi}}\sum_{\mathbf{c}\sim\boldsymbol{\psi}}\boldsymbol{\psi}(\mathbf{c}|s,\mathbf{m})\Bigg[\mathbb{1}(\mathbf{c}=\mathbf{1})Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s,\mathbf{m}) + \Big(1-\mathbb{1}(\mathbf{c}=\mathbf{1})\Big)\sum_{\mathbf{a}\sim\boldsymbol{\pi}}\boldsymbol{\pi}(\mathbf{a}|s)Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s,\mathbf{a})\Bigg]\nabla_{\eta^i}\boldsymbol{\phi}(\mathbf{m}|s)\\
&+\sum_{\mathbf{m}\sim\boldsymbol{\phi}}\boldsymbol{\phi}(\mathbf{m}|s)\sum_{\mathbf{c}\sim\boldsymbol{\psi}}\Bigg[\mathbb{1}(\mathbf{c}=\mathbf{1})Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s,\mathbf{m}) + \Big(1-\mathbb{1}(\mathbf{c}=\mathbf{1})\Big)\sum_{\mathbf{a}\sim\boldsymbol{\pi}}\boldsymbol{\pi}(\mathbf{a}|s)Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s,\mathbf{a})\Bigg]\nabla_{\eta^i}\boldsymbol{\psi}(\mathbf{c}|s,\mathbf{m})\\
&+\sum_{\mathbf{m}\sim\boldsymbol{\phi}}\boldsymbol{\phi}(\mathbf{m}|s)\sum_{\mathbf{c}\sim\boldsymbol{\psi}}\boldsymbol{\psi}(\mathbf{c}|s,\mathbf{m})\Bigg[Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s,\mathbf{m}) - \sum_{\mathbf{a}\sim\boldsymbol{\pi}}\boldsymbol{\pi}(\mathbf{a}|s)Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s,\mathbf{a})\Bigg]\nabla_{\eta^i}\mathbb{1}(\mathbf{c}=\mathbf{1})\\
&+\sum_{\mathbf{m}\sim\boldsymbol{\phi}}\boldsymbol{\phi}(\mathbf{m}|s)\sum_{\mathbf{c}\sim\boldsymbol{\psi}}\boldsymbol{\psi}(\mathbf{c}|s,\mathbf{m})\Bigg[\mathbb{1}(\mathbf{c}=\mathbf{1})\nabla_{\eta^i}Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s,\mathbf{m})\\
&+\Big(1-\mathbb{1}(\mathbf{c}=\mathbf{1})\Big)\sum_{\mathbf{a}\sim\boldsymbol{\pi}}\boldsymbol{\pi}(\mathbf{a}|s)\nabla_{\eta^i}Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s,\mathbf{a})\Bigg].
\end{aligned}
\tag{25}
$$

Let

$$
\begin{aligned}
&h_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s)\\
=&\sum_{\mathbf{m}\sim\boldsymbol{\phi}}\sum_{\mathbf{c}\sim\boldsymbol{\psi}}\boldsymbol{\psi}(\mathbf{c}|s,\mathbf{m})\Bigg[\mathbb{1}(\mathbf{c}=\mathbf{1})Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s,\mathbf{m}) + \Big(1-\mathbb{1}(\mathbf{c}=\mathbf{1})\Big)\sum_{\mathbf{a}\sim\boldsymbol{\pi}}\boldsymbol{\pi}(\mathbf{a}|s)Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s,\mathbf{a})\Bigg]\nabla_{\eta^i}\boldsymbol{\phi}(\mathbf{m}|s)\\
&+\sum_{\mathbf{m}\sim\boldsymbol{\phi}}\boldsymbol{\phi}(\mathbf{m}|s)\sum_{\mathbf{c}\sim\boldsymbol{\psi}}\Bigg[\mathbb{1}(\mathbf{c}=\mathbf{1})Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s,\mathbf{m}) + \Big(1-\mathbb{1}(\mathbf{c}=\mathbf{1})\Big)\sum_{\mathbf{a}\sim\boldsymbol{\pi}}\boldsymbol{\pi}(\mathbf{a}|s)Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s,\mathbf{a})\Bigg]\nabla_{\eta^i}\boldsymbol{\psi}(\mathbf{c}|s,\mathbf{m})\\
&+\sum_{\mathbf{m}\sim\boldsymbol{\phi}}\boldsymbol{\phi}(\mathbf{m}|s)\sum_{\mathbf{c}\sim\boldsymbol{\psi}}\boldsymbol{\psi}(\mathbf{c}|s,\mathbf{m})\Bigg[Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s,\mathbf{m}) - \sum_{\mathbf{a}\sim\boldsymbol{\pi}}\boldsymbol{\pi}(\mathbf{a}|s)Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s,\mathbf{a})\Bigg]\nabla_{\eta^i}\mathbb{1}(\mathbf{c}=\mathbf{1}).
\end{aligned}
\tag{26}
$$

Similarly we have

$$
\begin{aligned}
&\nabla_{\eta^i} V^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s)\\
\propto&\sum_{x\in\mathcal{S}}\rho_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(x)h_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(x),\\
=&\sum_{x\in\mathcal{S}}\rho_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(x)\Bigg[\sum_{\mathbf{m}\sim\boldsymbol{\phi}}\sum_{\mathbf{c}\sim\boldsymbol{\psi}}\boldsymbol{\psi}(\mathbf{c}|x,\mathbf{m})\Big[\mathbb{1}(\mathbf{c}=\mathbf{1})Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(x,\mathbf{m}) + \Big(1-\mathbb{1}(\mathbf{c}=\mathbf{1})\Big)\sum_{\mathbf{a}\sim\boldsymbol{\pi}}\boldsymbol{\pi}(\mathbf{a}|x)Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(x,\mathbf{a})\Big]\\
&\cdot\nabla_{\eta^i}\boldsymbol{\phi}(\mathbf{m}|x) + \sum_{\mathbf{m}\sim\boldsymbol{\phi}}\boldsymbol{\phi}(\mathbf{m}|x)\sum_{\mathbf{c}\sim\boldsymbol{\psi}}\Big[\mathbb{1}(\mathbf{c}=\mathbf{1})Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(x,\mathbf{m}) + \Big(1-\mathbb{1}(\mathbf{c}=\mathbf{1})\Big)\sum_{\mathbf{a}\sim\boldsymbol{\pi}}\boldsymbol{\pi}(\mathbf{a}|x)Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(x,\mathbf{a})\Big]\\
&\cdot\nabla_{\eta^i}\boldsymbol{\psi}(\mathbf{c}|x,\mathbf{m}) + \sum_{\mathbf{m}\sim\boldsymbol{\phi}}\boldsymbol{\phi}(\mathbf{m}|x)\sum_{\mathbf{c}\sim\boldsymbol{\psi}}\boldsymbol{\psi}(\mathbf{c}|x,\mathbf{m})\Big[Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(x,\mathbf{m}) - \sum_{\mathbf{a}\sim\boldsymbol{\pi}}\boldsymbol{\pi}(\mathbf{a}|x)Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(x,\mathbf{a})\Big]\nabla_{\eta^i}\mathbb{1}(\mathbf{c}=\mathbf{1})\Bigg],
\end{aligned}
$$

$$
\begin{aligned}
&=\mathbb{E}_{x\sim\rho_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}},\mathbf{m}\sim\boldsymbol{\phi},\mathbf{c}\sim\boldsymbol{\psi},\mathbf{a}\sim\boldsymbol{\pi}}\Bigg[\Big[\mathbb{1}(\mathbf{c}=\mathbf{1})Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(x,\mathbf{m})+\Big(1-\mathbb{1}(\mathbf{c}=\mathbf{1})\Big)Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(x,\mathbf{a})\Big]\Big(\nabla_{\eta^i}\log\boldsymbol{\phi}(\mathbf{m}|x)\\
&\quad+\nabla_{\eta^i}\log\boldsymbol{\psi}(\mathbf{c}|x,\mathbf{m})\Big)+\Big[Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(x,\mathbf{m})-Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(x,\mathbf{a})\Big]\nabla_{\eta^i}\mathbb{1}(\mathbf{c}=\mathbf{1})\Bigg],\\
&=\mathbb{E}_{x\sim\rho_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}},\mathbf{m}\sim\boldsymbol{\phi},\mathbf{c}\sim\boldsymbol{\psi},\mathbf{a}\sim\boldsymbol{\pi}}\Bigg[\Big[\mathbb{1}(\mathbf{c}=\mathbf{1})Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(x,\mathbf{m})+\Big(1-\mathbb{1}(\mathbf{c}=\mathbf{1})\Big)Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(x,\mathbf{a})\Big]\Big(\nabla_{\eta^i}\log\phi^i(m^i|x)\\
&\quad+\sum_j\nabla_{\eta^i}\log\psi^j(c^j|x,\mathbf{m})\Big)+\Big[Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(x,\mathbf{m})-Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(x,\mathbf{a})\Big]\Big(\mathbb{1}(c^{-i}=1)\nabla_{\eta^i}\mathbb{1}(c^i=1)+\mathbb{1}(c^i=1)\\
&\quad\cdot\nabla_{\eta^i}\mathbb{1}(c^{-i}=1)\Big)\Bigg],\\
&=\mathbb{E}_{x\sim\rho_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}},\mathbf{m}\sim\boldsymbol{\phi},\mathbf{c}\sim\boldsymbol{\psi},\mathbf{a}\sim\boldsymbol{\pi}}\Bigg[\Big[\mathbb{1}(\mathbf{c}=\mathbf{1})Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(x,\mathbf{m})+\Big(1-\mathbb{1}(\mathbf{c}=\mathbf{1})\Big)Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(x,\mathbf{a})\Big]\Big(\nabla_{\eta^i}\log\phi^i(m^i|x)\\
&\quad+\sum_j\nabla_{\eta^i}\log\psi^j(c^j|x,\mathbf{m})\Big)+\sum_j\prod_{k\neq j}\mathbb{1}(c^k=1)\Big[Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(x,\mathbf{m})-Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(x,\mathbf{a})\Big]\nabla_{\eta^i}\mathbb{1}(c^j=1)\Bigg].
\end{aligned}
\tag{27}
$$

Therefore,

$$
\begin{aligned}
\nabla_{\eta^i}V^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s)\propto&\mathbb{E}_{x\sim\rho_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}},\mathbf{m}\sim\boldsymbol{\phi},\mathbf{c}\sim\boldsymbol{\psi},\mathbf{a}\sim\boldsymbol{\pi}}\Bigg[\Big[\mathbb{1}(\mathbf{c}=\mathbf{1})Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(x,\mathbf{m})+\Big(1-\mathbb{1}(\mathbf{c}=\mathbf{1})\Big)Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(x,\mathbf{a})\Big]\cdot\\
&\Big(\nabla_{\eta^i}\log\phi^i(m^i|x)+\sum_j\nabla_{\eta^i}\log\psi^j(c^j|x,\mathbf{m})\Big)+\sum_j\prod_{k\neq j}\mathbb{1}(c^k=1)\Big[Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(x,\mathbf{m})-\\
&Q^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(x,\mathbf{a})\Big]\nabla_{\eta^i}\mathbb{1}(c^j=1)\Bigg].
\end{aligned}
$$

□

Note that $\nabla_{\eta^i}\mathbb{1}(c^i=1)=\frac{d\mathbb{1}(c^i=1)}{dc^i}(\frac{\partial\psi^i}{\partial c^i})^{-1}\frac{\partial\psi^i}{\partial m^i}\frac{\partial m^i}{\partial\eta^i}$, $\nabla_{\eta^i}\mathbb{1}(c^j=1)|_{j\neq i}=\frac{d\mathbb{1}(c^j=1)}{dc^j}(\frac{\partial\psi^j}{\partial c^j})^{-1}\frac{\partial\psi^j}{\partial m^i}\frac{\partial m^i}{\partial\eta^i}$. We apply Gumbel-Softmax distribution (Jang et al., 2016) again, which allows autodifferentiation through $m^i,\forall i$.

11 DCL Details

11.1 Centralized DCL

DCL updates policies with respect to policy gradients in Lemma 5.1. Because calculating $\nabla_{\eta^i}V^i_{\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\pi}}(s)$ requires differentiation through commitment policies of other agents $j\in\mathcal{N}\setminus i$, we present centralized DCL in Algorithm 2 that allows agents to backpropagate through exact policies of others.

11.2 Decentralized DCL

Centralized training is not always feasible in mixed-motive environments. To address this limitation, we further present decentralized DCL in Algorithm 3. In decentralized DCL, each agent estimates others' policies and value functions with DCL. Then, agents can differentiate through these estimates to update their own policies.

12 Proof of Proposition 4.1

Recall the definition of Nash equilibrium and Pareto-dominant outcome:

Definition 12.1. *(Hu & Wellman, 2003) In stochastic game Γ, a Nash equilibrium point is a tuple of n strategies $(\pi_*^1, ..., \pi_*^n)$ such that for all $s \in \mathcal{S}$ and $i = 1, ..., n$,*

$$V^i(s, \pi_*^1, ..., \pi_*^n) \geq V^i(s, \pi_*^1, ..., \pi_*^{i-1}, \pi^i, \pi_*^{i+1}, ..., \pi_*^n), \quad \forall \pi^i \in \Pi^i, \tag{28}$$

where Π^i is the set of strategies available to agent i.

At a Nash equilibrium, no player can improve their payoff by changing their strategy, assuming that the other players stick to their current strategies.

Definition 12.2. *(Censor, 1977; Fudenberg, 1991) An outcome of a game is Pareto-dominant, also known as Pareto-optimal and Pareto-efficient, if it's impossible to make one player better-off, without making some other players worse-off.*

To prove Proposition 4.1, we need to find a tuple of strategies $((\phi_*^i, \psi_*^i, \pi_*^i), (\phi_*^{-i}, \psi_*^{-i}, \pi_*^{-i}))$ in MCGs that satisfies the conditions of Nash equilibrium and Pareto-optimality.

Proof. In the MCG of the Prisoner's Dilemma, we define a tuple of deterministic strategies $\forall i \in \mathcal{N}$, $\phi^i(s) = C$, $\pi_*^i(s) = D$ and

$$\begin{aligned} \psi_*^i(s, \mathbf{m} = \{C, C\}) =& 1, \\ \psi_*^i(s, \mathbf{m} = \{D, C\}) =& 1, \\ \psi_*^i(s, \mathbf{m} = \{C, D\}) =& 0, \\ \psi_*^i(s, \mathbf{m} = \{D, D\}) =& 0 \text{ or } 1. \end{aligned} \tag{29}$$

So the value function of this tuple is:

$$V^i(s, (\phi_*^i, \psi_*^i, \pi_*^i)|(\phi_*^{-i}, \psi_*^{-i}, \pi_*^{-i})) = -1. \tag{30}$$

Then we show that no player can increase their payoff by unilaterally changing to other deterministic strategies, assuming all other players keep their strategies fixed.

1. $\forall \phi^i \neq \phi_*^i$, i.e. $\phi^i(s) = D$, and $\forall \psi^i$:
   (a) if $\pi^i(s) = C$,
   $$V^i(s, (\phi^i, \psi^i, \pi^i)|(\phi_*^{-i}, \psi_*^{-i}, \pi_*^{-i})) = -3 < -1 = V^i(s, (\phi_*^i, \psi_*^i, \pi_*^i)|(\phi_*^{-i}, \psi_*^{-i}, \pi_*^{-i})), \tag{31}$$
   (b) otherwise $\pi^i(s) = D$,
   $$V^i(s, (\phi^i, \psi^i, \pi^i)|(\phi_*^{-i}, \psi_*^{-i}, \pi_*^{-i})) = -2 < -1 = V^i(s, (\phi_*^i, \psi_*^i, \pi_*^i)|(\phi_*^{-i}, \psi_*^{-i}, \pi_*^{-i})). \tag{32}$$
2. $\phi^i = \phi_*^i$ and $\psi^i \neq \psi_*^i$:
   (a) $\forall \psi^i \neq \psi_*^i$ s.t. $\psi^i(s, \mathbf{m} = \{C, C\}) = 1$ and $\forall \pi^i$,
   $$V^i(s, (\phi^i, \psi^i, \pi^i)|(\phi_*^{-i}, \psi_*^{-i}, \pi_*^{-i})) = -1 = V^i(s, (\phi_*^i, \psi_*^i, \pi_*^i)|(\phi_*^{-i}, \psi_*^{-i}, \pi_*^{-i})). \tag{33}$$
   (b) $\forall \psi^i \neq \psi_*^i$ s.t. $\psi^i(s, \mathbf{m} = \{C, C\}) = 0$,
      i. if $\pi^i(s) = C$,
      $$V^i(s, (\phi^i, \psi^i, \pi^i)|(\phi_*^{-i}, \psi_*^{-i}, \pi_*^{-i})) = -3 < -1 = V^i(s, (\phi_*^i, \psi_*^i, \pi_*^i)|(\phi_*^{-i}, \psi_*^{-i}, \pi_*^{-i})), \tag{34}$$

ii. otherwise $\pi^i(s) = D$,

$$V^i(s, (\phi^i, \psi^i, \pi^i)|(\phi_*^{-i}, \psi_*^{-i}, \pi_*^{-i})) = -2 < -1 = V^i(s, (\phi_*^i, \psi_*^i, \pi_*^i)|(\phi_*^{-i}, \psi_*^{-i}, \pi_*^{-i})). \tag{35}$$

3. $\phi^i = \phi_*^i, \psi^i = \psi_*^i, \forall \pi^i$:

$$V^i(s, (\phi^i, \psi^i, \pi^i)|(\phi_*^{-i}, \psi_*^{-i}, \pi_*^{-i})) = -1 = V^i(s, (\phi_*^i, \psi_*^i, \pi_*^i)|(\phi_*^{-i}, \psi_*^{-i}, \pi_*^{-i})). \tag{36}$$

Thus,

$$V^i(s, (\phi^i, \psi^i, \pi^i)|(\phi_*^{-i}, \psi_*^{-i}, \pi_*^{-i})) \leq V^i(s, (\phi_*^i, \psi_*^i, \pi_*^i)|(\phi_*^{-i}, \psi_*^{-i}, \pi_*^{-i})), \quad \forall \phi^i, \psi^i, \pi^i. \tag{37}$$

Therefore, $((\phi_*^i, \psi_*^i, \pi_*^i), (\phi_*^{-i}, \psi_*^{-i}, \pi_*^{-i}))$ is a pure strategy Nash equilibrium in the MCG of Prisoner's Dilemma. Meanwhile, $((\phi_*^i, \psi_*^i, \pi_*^i), (\phi_*^{-i}, \psi_*^{-i}, \pi_*^{-i}))$ is also a Pareto-optimal equilibrium. Given the payoff matrix in Table 1, other possible outcomes are $(-2, -2)$, $(0, -3)$ and $(-3, 0)$. Therefore, no further improvement can be made to one player's outcome without reducing the payoff of another player compared to $(-1, -1)$ achieved by $((\phi_*^i, \psi_*^i, \pi_*^i), (\phi_*^{-i}, \psi_*^{-i}, \pi_*^{-i}))$. □

13 Hyperparameters

For all algorithms, we utilized 2-layer MLP networks with ReLU activation in the hidden layers. All policy networks apply a softmax function as the output activation, whereas the value network uses a linear output without any activation function. Other hyperparameters are reported in Table 5, 6 and 7.

Table 3: Comparison with Related Frameworks

Name	Commitment	Share Policies	Altruistic Third Party	Reward Transfer	Proposal of Actions
Commitment Games (Renou, 2009)	Unconditional	Yes	No	No	No
Conditional Commitment Games (Bryan et al., 2010)	Conditional	Yes	No	No	No
Contract Mechanism (Hughes et al., 2020)	Conditional	No	No	No	Joint Action
Formal Contracting (Haupt et al., 2022)	Conditional	No	No	Yes	No
Mediated-MARL (Ivanov et al., 2023)	N/A	No	Yes	No	No
MCGs (Ours)	Conditional	No	No	No	Self Action

Table 4: Comparison with MARL Baselines

Name	Objective	Reward Transfer	Independent Learning	Decentralized Learning
IPPO (Schulman et al., 2017)	Individual Returns	No	Yes	Yes
MOCA (Haupt et al., 2022)	Individual Returns	Yes	Yes	Yes
Mediated-MARL (Ivanov et al., 2023)	Social Welfare + Individual Returns	No	Yes	Yes
Centralized DCL (Ours)	Individual Returns	No	No	No
Decentralized DCL (Ours)	Individual Returns	No	No	Yes

Algorithm 2 Differentiable Commitment Learning (Centralized Version)

Input: initial action policy parameters θ^i, initial commitment policy parameters ζ^i, initial proposal policy parameters η^i, initial action-value function parameters w^i for all $i \in \mathcal{N}$.

for k=$0, 1, 2, ...$ **do**

Collect set of trajectories $\mathcal{D}_k = \{\tau_t\}$ by running latest policies $(\theta^i, \zeta^i, \eta^i), \forall i \in \mathcal{N}$.

Compute Monte-Carlo discounted accumulative rewards $\hat{G}_t^i, \forall i \in \mathcal{N}$.

Fit value function for all $i \in \mathcal{N}$ with gradient descent by minimizing the mean-squared error:

$$w_{k+1}^i = \arg\min_{w^i} \frac{1}{|\mathcal{D}_k|T} \sum_{\tau \in \mathcal{D}_k} \sum_{t=0}^{T} (Q_{w^i}^i(s_t, \mathbf{a}_t) - \hat{G}_t^i)^2.$$

Estimate action policy gradient for all $i \in \mathcal{N}$ as

$$\hat{g}_{\theta_k^i} = \frac{1}{|\mathcal{D}_k|} \sum_{\tau \in \mathcal{D}_k} \sum_{t=0}^{T} \Big(1 - \mathbb{1}(\mathbf{c}_t = \mathbf{1})\Big) Q_{w_{k+1}^i}^i(s_t, \mathbf{a}_t) \nabla_{\theta_k^i} \log \pi_{\theta_k^i}^i(a_t^i | s_t).$$

Estimate commitment policy gradient for all $i \in \mathcal{N}$ as

$$\begin{aligned}\hat{g}_{\zeta_k^i} =& \frac{1}{|\mathcal{D}_k|} \sum_{\tau \in \mathcal{D}_k} \sum_{t=0}^{T} \Big[\mathbb{1}(\mathbf{c}_t = \mathbf{1}) Q_{w_{k+1}^i}^i(s_t, \mathbf{m}_t) + \Big(1 - \mathbb{1}(\mathbf{c}_t = \mathbf{1})\Big) Q_{w_{k+1}^i}^i(s_t, \mathbf{a}_t)\Big] \\ & \cdot \nabla_{\zeta_k^i} \log \psi_{\zeta_k^i}^i(c_t^i | s_t, \mathbf{m}_t) + \Big[Q_{w_{k+1}^i}^i(s_t, \mathbf{m}_t) - Q_{w_{k+1}^i}^i(s_t, \mathbf{a}_t)\Big] \prod_{j \neq i} \mathbb{1}(c_t^j = 1) \nabla_{\zeta^i} \mathbb{1}(c_t^i = 1).\end{aligned}$$

Estimate proposal policy gradient w.r.t. the expected return for all $i \in \mathcal{N}$ by

$$\begin{aligned}\hat{g}_{\eta_k^i} =& \frac{1}{|\mathcal{D}_k|} \sum_{\tau \in \mathcal{D}_k} \sum_{t=0}^{T} \Big[\mathbb{1}(\mathbf{c}_t = \mathbf{1}) Q_{w_{k+1}^i}^i(s_t, \mathbf{m}_t) + \Big(1 - \mathbb{1}(\mathbf{c}_t = \mathbf{1})\Big) Q_{w_{k+1}^i}^i(s_t, \mathbf{a}_t)\Big] \\ & \cdot \Big(\nabla_{\eta_k^i} \log \phi_{\eta_k^i}^i(m_t^i | s_t) + \sum_j \nabla_{\eta_k^i} \log \psi_{\zeta_k^j}^j(c_t^j | s_t, \mathbf{m}_t)\Big) \\ & + \sum_j \prod_{l \neq j} \mathbb{1}(c_t^l = 1) \Big[Q_{w_{k+1}^i}^i(s_t, \mathbf{m}_t) - Q_{w_{k+1}^i}^i(s_t, \mathbf{a}_t)\Big] \nabla_{\eta_k^i} \mathbb{1}(c_t^j = 1).\end{aligned}$$

Estimate proposal policy gradient w.r.t. incentive-compatible constraints for all $i \in \mathcal{N}$ by

$$\hat{g}_{\eta_k^i}^{'} = \frac{1}{|\mathcal{D}_k|} \sum_{\tau \in \mathcal{D}_k} \sum_{t=0}^{T} \sum_j \nabla_{\eta_k^i} \min\{0, Q_{w_{k+1}^j}^j(s_t, \mathbf{m}_t) - Q_{w_{k+1}^j}^j(s_t, \mathbf{a}_t)\}.$$

Update policy parameters for all $i \in \mathcal{N}$ with gradient ascent,

$$\theta_{k+1}^i = \theta_k^i + \beta \hat{g}_{\theta_k^i},$$

$$\zeta_{k+1}^i = \zeta_k^i + \beta \hat{g}_{\zeta_k^i},$$

$$\eta_{k+1}^i = \eta_k^i + \beta \hat{g}_{\eta_k^i} + \lambda \hat{g}_{\eta_k^i}^{'}.$$

end for

Algorithm 3 Differentiable Commitment Learning (Decentralized Version)

Input: initial action policy parameters: θ^i, initial estimated action policy parameters of $b \in \mathcal{N} \setminus i$: $\tilde{\theta}^{ib}$, initial commitment policy parameters: ζ^i, initial estimated commitment policy parameters of $b \in \mathcal{N} \setminus i$: $\tilde{\zeta}^{ib}$, initial proposal policy parameters: η^i, initial estimated proposal policy parameters of $b \in \mathcal{N} \setminus i$: $\tilde{\eta}^{ib}$, initial action-value function parameters: w^i for $i \in \mathcal{N}$, initial estimated action-value function parameters of $b \in \mathcal{N} \setminus i$: $\tilde{w}^{ib}$ for all $i \in \mathcal{N}$.

for k=$0, 1, 2, ...$ **do**

Collect set of trajectories $\mathcal{D}_k = \{\tau_t\}$ by running latest policies $(\theta^i, \zeta^i, \eta^i), \forall i \in \mathcal{N}$.

Compute Monte-Carlo discounted accumulative rewards $\hat{G}_t^i, \forall i \in \mathcal{N}$.

Fit value function for all $i \in \mathcal{N}$ with gradient descent by minimizing the mean-squared error:

$$w_{k+1}^i = \arg\min_{w^i} \frac{1}{|\mathcal{D}_k|T} \sum_{\tau \in \mathcal{D}_k} \sum_{t=0}^{T} (Q_{w^i}^i(s_t, \mathbf{a}_t) - \hat{G}_t^i)^2.$$

Fit estimated value function of b for $\forall b \in \mathcal{N} \setminus i$ and $\forall i \in \mathcal{N}$ with gradient descent by minimizing the mean-squared error:

$$\tilde{w}_{k+1}^{ib} = \arg\min_{w^b} \frac{1}{|\mathcal{D}_k|T} \sum_{\tau \in \mathcal{D}_k} \sum_{t=0}^{T} (Q_{w^b}^b(s_t, \mathbf{a}_t) - \hat{G}_t^b)^2.$$

Estimate action policy gradient for all $i \in \mathcal{N}$ as

$$\hat{g}_{\theta_k^i} = \frac{1}{|\mathcal{D}_k|} \sum_{\tau \in \mathcal{D}_k} \sum_{t=0}^{T} \Big(1 - \mathbb{1}(\mathbf{c}_t = \mathbf{1})\Big) Q_{w_{k+1}^i}^i(s_t, \mathbf{a}_t) \nabla_{\theta_k^i} \log \pi_{\theta_k^i}^i(a_t^i | s_t).$$

Estimate action policy of b for $\forall b \in \mathcal{N} \setminus i$ and $\forall i \in \mathcal{N}$ by

$$\hat{g}_{\tilde{\theta}_k^{ib}} = \frac{1}{|\mathcal{D}_k|} \sum_{\tau \in \mathcal{D}_k} \sum_{t=0}^{T} \Big(1 - \mathbb{1}(\mathbf{c}_t = \mathbf{1})\Big) Q_{\tilde{w}_{k+1}^{ib}}^b(s_t, \mathbf{a}_t) \nabla_{\tilde{\theta}_k^{ib}} \log \pi_{\tilde{\theta}_k^{ib}}^b(a_t^b | s_t).$$

Estimate commitment policy gradient for all $i \in \mathcal{N}$ as

$$\begin{aligned}\hat{g}_{\zeta_k^i} =& \frac{1}{|\mathcal{D}_k|} \sum_{\tau \in \mathcal{D}_k} \sum_{t=0}^{T} \Big[\mathbb{1}(\mathbf{c}_t = \mathbf{1}) Q_{w_{k+1}^i}^i(s_t, \mathbf{m}_t) + \Big(1 - \mathbb{1}(\mathbf{c}_t = \mathbf{1})\Big) Q_{w_{k+1}^i}^i(s_t, \mathbf{a}_t)\Big] \\ & \cdot \nabla_{\zeta_k^i} \log \psi_{\zeta_k^i}^i(c_t^i | s_t, \mathbf{m}_t) + \Big[Q_{w_{k+1}^i}^i(s_t, \mathbf{m}_t) - Q_{w_{k+1}^i}^i(s_t, \mathbf{a}_t)\Big] \prod_{j \neq i} \mathbb{1}(c_t^j = 1) \nabla_{\zeta^i} \mathbb{1}(c_t^i = 1).\end{aligned}$$

Estimate commitment policy gradient of b for $\forall b \in \mathcal{N} \setminus i$ and $\forall i \in \mathcal{N}$ by

$$\begin{aligned}\hat{g}_{\tilde{\zeta}_k^{ib}} =& \frac{1}{|\mathcal{D}_k|} \sum_{\tau \in \mathcal{D}_k} \sum_{t=0}^{T} \Big[\mathbb{1}(\mathbf{c}_t = \mathbf{1}) Q_{\tilde{w}_{k+1}^{ib}}^b(s_t, \mathbf{m}_t) + \Big(1 - \mathbb{1}(\mathbf{c}_t = \mathbf{1})\Big) Q_{\tilde{w}_{k+1}^{ib}}^b(s_t, \mathbf{a}_t)\Big] \\ & \cdot \nabla_{\tilde{\zeta}_k^{ib}} \log \psi_{\tilde{\zeta}_k^{ib}}^i(c_t^b | s_t, \mathbf{m}_t) + \Big[Q_{\tilde{w}_{k+1}^{ib}}^b(s_t, \mathbf{m}_t) - Q_{\tilde{w}_{k+1}^{ib}}^b(s_t, \mathbf{a}_t)\Big] \prod_{j \neq b} \mathbb{1}(c_t^j = 1) \nabla_{\tilde{\zeta}_k^{ib}} \mathbb{1}(c_t^b = 1).\end{aligned}$$

Estimate proposal policy gradient w.r.t. the expected return for all $i \in \mathcal{N}$ by

$$\begin{aligned}\hat{g}_{\eta_k^i} =& \frac{1}{|\mathcal{D}_k|} \sum_{\tau \in \mathcal{D}_k} \sum_{t=0}^{T} \Big[\mathbb{1}(\mathbf{c}_t = \mathbf{1}) Q_{w_{k+1}^i}^i(s_t, \mathbf{m}_t) + \Big(1 - \mathbb{1}(\mathbf{c}_t = \mathbf{1})\Big) Q_{w_{k+1}^i}^i(s_t, \mathbf{a}_t)\Big] \cdot \Big(\nabla_{\eta_k^i} \log \phi_{\eta_k^i}^i(m_t^i | s_t) \\ & + \sum_j \nabla_{\eta_k^i} \log \psi_{\zeta_k^j}^j(c_t^j | s_t, \mathbf{m}_t)\Big) + \sum_j \prod_{l \neq j} \mathbb{1}(c_t^l = 1) \Big[Q_{w_{k+1}^i}^i(s_t, \mathbf{m}_t) - Q_{w_{k+1}^i}^i(s_t, \mathbf{a}_t)\Big] \nabla_{\eta_k^i} \mathbb{1}(c_t^j = 1).\end{aligned}$$

Estimate proposal policy gradient w.r.t. incentive-compatible constraints for all $i \in \mathcal{N}$ by

$$\hat{g}'_{\eta^i_k} = \frac{1}{|\mathcal{D}_k|} \sum_{\tau \in \mathcal{D}_k} \sum_{t=0}^{T} \nabla_{\eta^i_k} \min\{0, Q^i_{w^i_{k+1}}(s_t, \mathbf{m}_t) - Q^i_{w^i_{k+1}}(s, \mathbf{a})\} + \sum_{b \neq i} \nabla_{\eta^i_k} \min\{0, Q^b_{\tilde{w}^{ib}_k}(s, \mathbf{m}) - Q^b_{\tilde{w}^{ib}_k}(s, \mathbf{a})\}.$$

Estimate proposal policy gradient w.r.t. the expected return of b for $\forall b \in \mathcal{N} \setminus i$ and $\forall i \in \mathcal{N}$ by

$$\begin{aligned}\hat{g}_{\tilde{\eta}^{ib}_k} =& \frac{1}{|\mathcal{D}_k|} \sum_{\tau \in \mathcal{D}_k} \sum_{t=0}^{T} \Big[\mathbb{1}(\mathbf{c}_t = \mathbf{1}) Q^b_{\tilde{w}^{ib}_{k+1}}(s_t, \mathbf{m}_t) + \Big(1 - \mathbb{1}(\mathbf{c}_t = \mathbf{1})\Big) Q^b_{\tilde{w}^{ib}_{k+1}}(s_t, \mathbf{a}_t)\Big] \cdot \Big(\nabla_{\tilde{\eta}^{ib}_k} \log \phi^b_{\tilde{\eta}^{ib}_k}(m^b_t | s_t) \\ &+ \sum_j \nabla_{\tilde{\eta}^{ib}_k} \log \psi^j_{\tilde{\zeta}^{ij}_k}(c^j_t | s_t, \mathbf{m}_t)\Big) + \sum_j \prod_{l \neq j} \mathbb{1}(c^l_t = 1) \Big[Q^b_{\tilde{w}^{ib}_{k+1}}(s_t, \mathbf{m}_t) - Q^b_{\tilde{w}^{ib}_{k+1}}(s_t, \mathbf{a}_t)\Big] \nabla_{\tilde{\eta}^{ib}_k} \mathbb{1}(c^j_t = 1).\end{aligned}$$

Estimate proposal policy gradient w.r.t. incentive-compatible constraints of b for $\forall b \in \mathcal{N} \setminus i$ and $\forall i \in \mathcal{N}$ by

$$\hat{g}'_{\tilde{\eta}^{ib}_k} = \frac{1}{|\mathcal{D}_k|} \sum_{\tau \in \mathcal{D}_k} \sum_{t=0}^{T} \nabla_{\tilde{\eta}^{ib}_k} \min\{0, Q^i_{w^i_{k+1}}(s_t, \mathbf{m}_t) - Q^i_{w^i_{k+1}}(s, \mathbf{a})\} + \sum_{b \neq i} \nabla_{\tilde{\eta}^{ib}_k} \min\{0, Q^b_{\tilde{w}^{ib}_{k+1}}(s, \mathbf{m}) - Q^b_{\tilde{w}^{ib}_{k+1}}(s, \mathbf{a})\}.$$

Update policy parameters for all $i \in \mathcal{N}$ with gradient ascent,

$$\theta^i_{k+1} = \theta^i_k + \beta \hat{g}_{\theta^i_k}, \zeta^i_{k+1} = \zeta^i_k + \beta \hat{g}_{\zeta^i_k}, \eta^i_{k+1} = \eta^i_k + \beta \hat{g}_{\eta^i_k} + \lambda \hat{g}'_{\eta^i_k}.$$

Update policy parameters for all $b \in \mathcal{N} \setminus i$ and $i \in \mathcal{N}$ with gradient ascent,

$$\tilde{\theta}^{ib}_{k+1} = \tilde{\theta}^{ib}_k + \beta \hat{g}_{\tilde{\theta}^{ib}_k}, \tilde{\zeta}^{ib}_{k+1} = \tilde{\zeta}^{ib}_k + \beta \hat{g}_{\tilde{\zeta}^{ib}_k}, \tilde{\eta}^{ib}_{k+1} = \tilde{\eta}^{ib}_k + \beta \hat{g}_{\tilde{\eta}^{ib}_k} + \lambda \hat{g}'_{\tilde{\eta}^{ib}_k}.$$

end for

Table 5: Hyperparameters of Prisoner's Dilemma

Hyperparameters	**DCL**	**Mediated-MARL**	**IPPO**	**MOCA**
Num of Iterations	10,000	10,000	10,000	10,000
Batch size	128	128	128	128
Entropy Coef. Start	1.0	1.0	N/A	N/A
Entropy Decay	0.0005	0.0005	N/A	N/A
Min. Entropy Coef.	0	0	N/A	N/A
LR of Value Function	8e-4	8e-4	8e-4	8e-4
LR of Policies	4e-4	4e-4	4e-4	4e-4
Hidden Layer size	8	8	8	8
Num of Layers	2	2	2	2
KL-coefficient	N/A	N/A	0.2	0.2
KL-target	N/A	N/A	0.01	0.01
Clip Parameter in PPO	N/A	N/A	0.3	0.3
Temperature	10.0	N/A	N/A	N/A
Temperature Decay	0.05	N/A	N/A	N/A
Min. Temperature	1.0	N/A	N/A	N/A
Num of Update Per Iteration	1	1	1	1

Table 6: Hyperparameters of Grid Game

Hyperparameters	**DCL**	**Mediated-MARL**	**IPPO**	**MOCA**
Horizon	16	16	16	16
Grid Size	4	4	4	4
Num of Iterations	10,000	10,000	10,000	10,000
Discount Factor	0.99	0.99	0.99	0.99
Batch size	512	512	512	512
Entropy Coef. Start	2.0	2.0	N/A	N/A
Entropy Decay	0.0005	0.0005	N/A	N/A
Min. Entropy Coef.	0.001	0.001	N/A	N/A
LR of Value Function	8e-4	8e-4	8e-4	8e-4
LR of Policies	4e-4	4e-4	4e-4	4e-4
Hidden Layer size	32	32	32	32
Num of Layers	2	2	2	2
KL-coefficient	N/A	N/A	0.2	0.2
KL-target	N/A	N/A	0.01	0.01
Clip Parameter in PPO	N/A	N/A	0.3	0.3
Temperature	1.0	N/A	N/A	N/A
Temperature Decay	0	N/A	N/A	N/A
Min. Temperature	1.0	N/A	N/A	N/A
Num of Update Per Iteration	30	30	30	30

Table 7: Hyperparameters of Repeated Pure Conflicting Game

Hyperparameters	**DCL**	**Mediated-MARL**	**IPPO**	**MOCA**
Num of Iterations	10,000	10,000	10,000	10,000
Discount Factor	0.99	0.99	0.99	0.99
Batch size	512	512	512	512
Entropy Coef. Start	2.0	2.0	N/A	N/A
Entropy Decay	0.0005	0.0005	N/A	N/A
Min. Entropy Coef.	0.001	0.001	N/A	N/A
LR of Value Function	8e-4	8e-4	8e-4	8e-4
LR of Policies	4e-4	4e-4	4e-4	4e-4
Hidden Layer size	32	32	32	32
Num of Layers	2	2	2	2
KL-coefficient	N/A	N/A	0.2	0.2
KL-target	N/A	N/A	0.01	0.01
Clip Parameter in PPO	N/A	N/A	0.3	0.3
Temperature	1.0	N/A	N/A	N/A
Temperature Decay	0	N/A	N/A	N/A
Min. Temperature	1.0	N/A	N/A	N/A
Num of Update Per Iteration	30	30	30	30

14 Many-player Experiments

To investigate how DCL handles scalability with many players, we conducted experiments on an N-player public goods game. For each agent i, the reward is calculated as $R^i = \sum_j C^j * \beta - C^i$, where C^i denotes the contribution of agent i, β denotes the benefit factor with a range between $(1, N)$. In our experiments, we set $\beta = 1.5$ for all scenarios. Results in Table 8 indicate that DCL with incentive-compatible constraints scales effectively with large numbers of agents. While the runtime of DCL increases with the number of agents, the agreement rate of joint proposals remains stable (> 0.99), achieving high social welfare.

Table 8: DCL-IC on Many-player Public Goods Game

Number of Agents	Run Time (Hours)	Agreement Rate	Social Welfare
2	4	0.996 ± 0.002	0.997 ± 0.002
3	7	0.994 ± 0.001	1.491 ± 0.004
5	12	0.996 ± 0.001	1.989 ± 0.002
10	32	0.991 ± 0.001	3.659 ± 0.143

15 Other Related Works

15.1 Cooperation Problems in Mixed-Motive Environments

The causes of cooperation failures between self-interested agents in mixed-motive environments have been primarily categorized into two types: information problems and commitment problems (Dafoe et al., 2020; Powell, 2006; Fearon, 1995). Information problems refer to cooperation failures caused by incorrect or insufficient information, which frequently occur in partially observable environments. Existing works have demonstrated that information problems can be alleviated by communication (Kim et al., 2020; Sukhbaatar et al., 2016; Foerster et al., 2016) and opponent reasoning (Konan et al., 2022; Jaques et al., 2019; Wen et al., 2019). However, in mixed-motive environments, agents driven by conflicting self-interests may deceive others regarding their

private observations (Lin et al., 2024; Kamenica, 2019; Taneva, 2019; Dughmi, 2017). Cooperation may also fail due to agents' inability to make credible commitments, known as commitment problems, even in the absence of information asymmetries. For instance, cooperation can not be achieved through cheap talk communication or non-binding promises of cooperation in the Prisoner's Dilemma (Rapoport, 1965), as agents achieve higher payoffs by defecting regardless of the opponent's actions. To address commitment problems, a commitment device is often required to ensure that agents fulfill their commitments, either by restricting their actions or imposing penalties for noncompliance (Sun et al., 2023; Rogers et al., 2014). Static conditional commitments facilitated by such devices have been shown to enhance cooperation in the prisoner's dilemma (Kalai et al., 2010; Renou, 2009; Schelling, 1980). However, these fixed strategies are difficult to generalize across various games and environments.

15.2 Comparison with Related Works

Table 3 and 4 summarize the differences and similarities between various types of games and associated algorithms to optimize strategies.