Sequential Asynchronous Action Coordination in Multi-Agent Systems: A Stackelberg Decision Transformer Approach

Bin Zhang [1 2] **Hangyu Mao** [3] **Lijuan Li** [1 2] **Zhiwei Xu** [4] **Dapeng Li** [1 2] **Rui Zhao** [3] **Guoliang Fan** [1 2]

Abstract

Asynchronous action coordination presents a pervasive challenge in Multi-Agent Systems (MAS), which can be represented as a Stackelberg game (SG). However, the scalability of existing Multi-Agent Reinforcement Learning (MARL) methods based on SG is severely restricted by network architectures or environmental settings. To address this issue, we propose the Stackelberg Decision Transformer (STEER). It efficiently manages decision-making processes by incorporating the hierarchical decision structure of SG, the modeling capability of autoregressive sequence models, and the exploratory learning methodology of MARL. Our approach exhibits broad applicability across diverse task types and environmental configurations in MAS. Experimental results demonstrate both the convergence of our method towards Stackelberg equilibrium strategies and its superiority over strong baselines in complex scenarios.

1. Introduction

In multi-agent systems (MAS), agents must not only maximize their individual rewards by interacting with the environment, but also dynamically coordinate with other agents to achieve the optimal collective strategy (Lu & Yan, 2020). Multi-agent reinforcement learning (MARL) has emerged as a promising approach to tackle this task effectively, but it also poses significant challenges (Shen et al., 2022). Existing prevalent methods for MARL primarily focus on fully cooperative tasks and assume synchronous actions among all agents (Rashid et al., 2018; Yu et al., 2021). However, when considering mixed tasks, which are more generalized and widely applicable, self-interested agents with private rewards are involved in both cooperative and competitive dynamics. Moreover, in real-world scenarios, the decision-making process of agents is frequently influenced by the actions taken by other agents at the same time (Ruan et al., 2022). As a result, these methods exhibit limitations in effectively handling complex interactions among agents and encounter difficulties even in simple coordination scenarios (Xu et al., 2023).

Game theory provides an effective conceptual framework for addressing interactions among agents, thereby offers a promising avenue to address the challenges associated with mixed tasks and asynchronous action coordination (Hu & Wellman, 2003). Notably, Stackelberg game (SG) explicitly models the sequential asynchronous action coordination among agents. It entails agents making decisions in a prescribed sequence, with leaders committing to their actions and followers discovering the optimal response to leaders' decisions. AQL (Könönen, 2004), BiRL (Zhang et al., 2020), and STEP (Zhang et al., 2023b) are designed to acquire Stackelberg equilibrium (SE) strategies via MARL. However, they typically impose stringent requirements on the network structure and environment, thereby constraining their scalability:

(1) All methods are restricted to environments with shared states, allowing followers to infer the actions of leaders based on the same inputs. However, this significantly limits their applicability in scenarios where agents have private observations.

(2) As a heterogeneous policy learning approach, all methods update the policies of each agent in a sequential manner, resulting in significant learning cost.

Recent advances in autoregressive sequence models derived from natural language processing (NLP) (Radford et al., 2019) have facilitated the development of novel reinforcement learning (RL) applications (Chen et al., 2021; Janner et al., 2021). **In this paper, our central insight lies in the seamless alignment between the hierarchical decision-making structure of SG and the modeling approach of autoregressive sequence models**. Building upon this, we propose a novel approach that utilizes sequence models to address the aforementioned challenges.

[1]Institute of Automation, Chinese Academy of Sciences [2]School of Artificial Intelligence, University of Chinese Academy of Sciences [3]SenseTime Research [4]School of Computer Science and Technology, Shandong University. Correspondence to: Lijuan Li <lijuan.li@ia.ac.cn>, Zhiwei Xu <diligencexu@gmail.com>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

Specifically, we first introduce a heuristic Stackelberg decision mechanism. Through the construction of the decision form of SG and the utilization of RL techniques, the convergence of the SE strategy is achieved in a natural manner. Furthermore, as our core contribution, we formally introduce the Stackelberg Decision Transformer (STEER). It incorporates a dual Transformer architecture comprising an Inner Transformer Block (ITB) and an Outer Transformer Block (OTB). With respect to challenge (1), ITB enables us to effectively manage tasks in a variety of environment configurations. The OTB further facilitates autoregressive fitting of policy and value functions for each agent. As for challenge (2), the Transformer architecture enables parallel updates of all agents' policies during the training phase, thereby reducing the computational costs that are previously imposed by SG based RL methods. Finally, we also put forth a viable scheme for extending its application to decentralized execution systems.

We employ widely recognized benchmarks to evaluate the effectiveness of the proposed STEER. Experimental results demonstrate its SE policy learning capability in both single-step and multi-step matrix games. Moreover, it exhibits superior performance and applicability compared to baseline approaches in complex scenarios.

2. Related Work

MARL. MARL aims to learn an optimal joint strategy that maximizes collective return (Zhang et al., 2023a). In this context, learning heterogeneous strategies for agents aligns better with intuition compared to parameter sharing methods, particularly in scenarios with self-interested or heterogeneous agents. Asynchronous Actor-Critic (Xiao et al., 2022) framework enables agents to execute temporally-extended actions, making it applicable to scenarios where the execution time of actions may vary. Seraj et al. present a hierarchical coordination framework for addressing joint perception-action tasks in composite robot teams composed of perceptual agents and action agents, which aligns more closely with hierarchical RL methods. HAPPO (Kuba et al., 2022) and A2PO (Wang et al., 2023) optimize the policy of each agent through a sequence update scheme. However, these approaches suffer from higher learning costs and longer training time. Although MAT (Wen et al., 2022) can somewhat alleviate these issues, the advantage decomposition theorem on which all of these methods rely only applies to fully-cooperative scenarios, limiting their ability to handle diverse types of tasks. In situations where agents have private rewards, defining the joint advantage value becomes challenging, and evaluating the quality of the joint policy becomes difficult. In this paper, we aim to develop a universal approach for learning heterogeneous strategies in both fully-cooperative and diverse mixed scenarios.

SG Based MARL. Our research endeavors to address the prevalent challenge of asynchronous action coordination in MAS, with a particular emphasis on hierarchical coordination and SG structure among agents. Given the superiority of SE over Nash equilibrium (NE) in terms of existence, determinacy, and Pareto optimality (Başar & Olsder, 1998; Zhang et al., 2020), recent studies have delved into the application of SE in MARL. Gerstgrasser & Parkes propose the use of multi-task and meta-learning techniques to learn solutions for the SE in two-player games. He et al. employ a three-stage SG framework to achieve clustering federated learning for heterogeneous UAV swarms. Similar to Nash Q-learning (Hu & Wellman, 2003), AQL (Könönen, 2004) updates the Q-value function in an asymmetric setting by calculating the SE of the stage game at each iteration. BiRL (Zhang et al., 2020) proposes a two-player MARL method, utilizing a DQN-based (Mnih et al., 2013) learner for the leader and a DDPG-based (Lowe et al., 2017) learner for the follower. To enforce the SE policy, both the leader and follower need to store each other's model. STEP (Zhang et al., 2023b) leverages hypernetworks (von Oswald et al., 2020) to facilitate the execution of heterogeneous SE policies, with followers inferring the actions of leaders to determine their response policies. However, these methods utilize intricate network structures and follow the presupposition that all agents share global state, which narrows the scope of their applicability.

Transformer in RL. Recently, researchers have increased their focus on applying autoregressive sequence models (Vaswani et al., 2017) to MARL . UPdet (Hu et al., 2021) concentrates on representation learning, with Transformer processing relationships between various entities in observations and matching them with subsets of the action space. MAT (Wen et al., 2022) incorporates Transformer and employs the advantage decomposition theorem to solve fully cooperative tasks while anticipating convergence to NE policies. However, it adopts the standard encoder-decoder structure without further optimization for MARL. MADT (Meng et al., 2021) employs Transformer to introduce the MAS field to the offline pre-training and online fine-tuning paradigm. Our approach utilizes Transformer to enable agents to cognize environmental states and the decision factors involved in SG framework.

3. Preliminaries

Markov Game. Markov game (MG) provides a powerful framework for modeling multi-agent decision-making problems in a stochastic environment. It is defined by the tuple $\Gamma \triangleq \langle \mathcal{I}, \mathcal{S}, \{\mathcal{A}^i\}_{i\in\mathcal{I}}, \mathcal{P}, \{r^i\}_{i\in\mathcal{I}}, \gamma \rangle$, where $\mathcal{I}$ represents the set of all agents with $|\mathcal{I}| = n$, and $s \in \mathcal{S}$ represents the environmental state. $a^i \in \mathcal{A}^i$ is the action of agent i and the joint action space is $\mathcal{A} = \prod_{i=1}^{n} \mathcal{A}^i$. $\mathcal{P} : \mathcal{S} \times \mathcal{A} \to \Omega(\mathcal{S})$

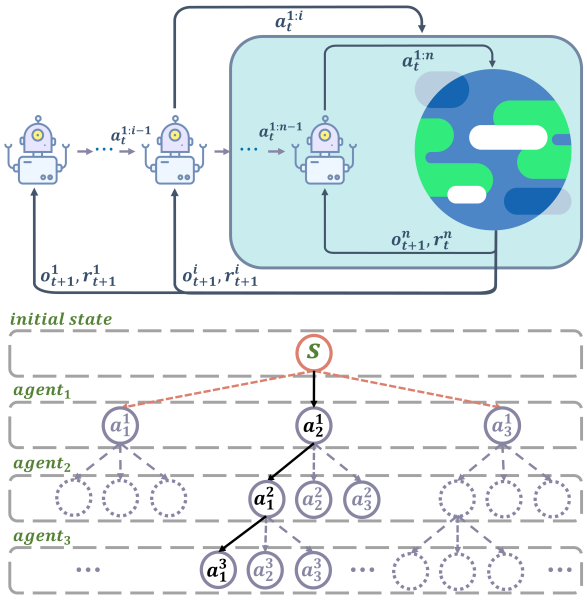


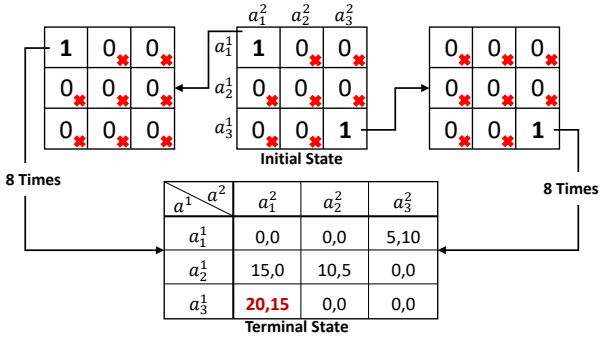
Figure 1. ***Top***: Illustration of the heuristic Stackelberg decision mechanism. Followers interact with the environment based on joint actions with leaders, while leaders instruct followers as constituents of the environment. ***Bottom***: Schematic representation of Stackelberg sequential decision-making.

represents the state transition function of the environment, where $\Omega(X)$ denotes the set of probability distributions over X. $r^i : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function of agent i and γ is the discount factor. At time step t, each agent i executes its policy $\pi^i : \mathcal{S} \to \Omega(\mathcal{A}^i)$ based on state s_t. The environment transitions to a new state $s_{t+1} \sim P(s_{t+1} \mid s_t, \boldsymbol{a_t})$ after receiving the joint action $\boldsymbol{a_t} = (a_t^1, ..., a_t^n)$ and assigns private rewards $r^i(s_t, \boldsymbol{a_t})$ for each agent. The joint policy is represented by $\boldsymbol{\pi}(s_t) = \prod_{i=1}^{n} \pi^i(s_t)$. The transition function and the joint strategy determine the state's marginal distribution $\rho_{\boldsymbol{\pi}}$ at each time step. Within this framework, each agent aims to maximize its own discounted cumulative return $R^i(\tau) = \sum_{t=0}^{T} \gamma^t r^i(s_t, \boldsymbol{a_t})$ over a trajectory τ of length T. According to Bellman Equation, the action value function of agent i in MG can be written as:

$$Q^i_{\boldsymbol{\pi}}(s, a^i) = \mathbb{E}_{s\sim\rho, \boldsymbol{a}\sim\boldsymbol{\pi}}\left[\sum_{t=0}^{\infty} \gamma^t \cdot r^i_t(s_t, \boldsymbol{a_t}) \mid \mathrm{s}_0 = s\right]. \quad (1)$$

The state value function is $V^i_{\boldsymbol{\pi}}(s) = \sum_{a^i \in \mathcal{A}^i} \pi^i(a^i|s) \cdot Q^i_{\boldsymbol{\pi}}(s, a^i)$. In certain environmental settings, agents may have access to localized observations $\{\mathcal{O}^i\}_{i\in\mathcal{I}}$ that are specific to each agent. Additionally, when $r^1 = \cdots = r^n$, the task is considered a fully-cooperative task, otherwise, it is referred to as a mixed task.

Stackelberg Game. The Stackelberg game (SG) (Von Stackelberg, 2010) is a well-established game-theoretic framework that models the hierarchical decision-making



a^1 \ a^2	a_1^2	a_2^2	a_3^2
a_1^1	0,0	0,0	5,10
a_2^1	15,0	10,5	0,0
a_3^1	20,15	0,0	0,0

Figure 2. Multi-step matrix game: Coordination. Only actions that yield non-zero rewards are permitted prior to the terminal state.

structure where some agents have priorities over others. Typically, such structure consists of leaders, who are superior agents capable of committing to their actions prior to other agents, and followers, who are inferior agents that must respond to the leaders' decisions. Leaders make decisions based on the assumption that followers will always react optimally to their actions. As an illustration, consider two agents whose leader and follower policies are denoted by $\boldsymbol{\pi} = (\pi^1, \pi^2)$. This can be formulated as a bi-level optimization problem:

$$\begin{aligned} &\max_{\pi^1 \in \Pi^1} \{\mathcal{J}^1(\pi^1, \pi^2) | \pi^2 \in \arg\max_{\pi^{2'} \in \Pi^2} \mathcal{J}^2(\pi^1, \pi^{2'})\}, \\ &\max_{\pi^2 \in \Pi^2} \mathcal{J}^2(\pi^1, \pi^2), \end{aligned} \quad (2)$$

where Π represents policy space and $\mathcal{J}^i(\pi^1, \pi^2) = E_{s\sim\rho, \boldsymbol{a}\sim\boldsymbol{\pi}}\left[\sum_{t=0} \gamma^t r^i_t(s, a^1_t, a^2_t)\right]$ is the objective function of agent i. SE strategy corresponds to the optimal solution of this bi-level optimization problem.

4. Heuristic Stackelberg Decision Mechanism

In the context of multi-agent SG, we allocate each agent i to an individual priority level h^i and $\mathcal{H} = \{h^1, ..., h^n\}$ is a prioritized permutation of agents. For simplicity, it is assumed that the priorities of the agents are assigned based on their agent *ID*, i.e., $h^i = i$. In this setting, we extend the bi-level optimization problem in Equation (2), leading to the derivation of an n-level optimization problem:

$$\max_{\pi^i \in \Pi^i} \{\mathcal{J}^i(\pi^{1:i-1}, \pi^i) | \pi^j \in \arg\max_{\pi^{j'} \in \Pi^j} \mathcal{J}^j(\pi^{1:j-1}, \pi^{j'})\}, \quad (3)$$

$$\max_{\pi^j \in \Pi^j} \mathcal{J}^j(\pi^{1:j-1}, \pi^j), \quad (4)$$

where $i \in [1:n]$ and $j \in [i+1, n]$. Drawing inspiration from the heuristic algorithm for bi-level optimization (Liu et al., 2021a; Sinha et al., 2017), we propose an RL-based heuristic Stackelberg decision mechanism (SDM) for this problem as shown in Figure 1. In SDM, each agent assumes the role of a follower to higher-level agents while

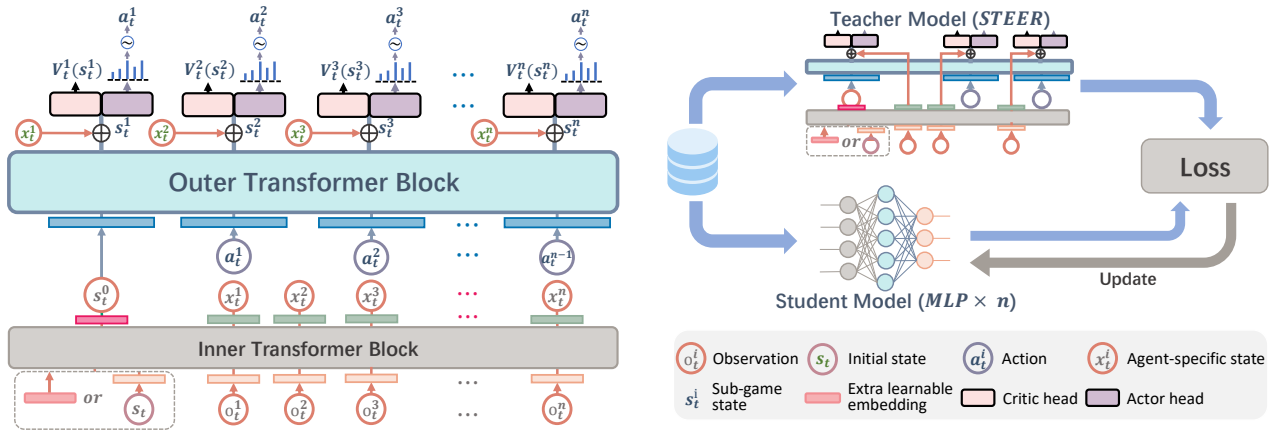


Figure 3. ***Left***: The overall architecture of STEER. At each time step, the state s_t and observation $\{o_t^i\}_{i\in\mathcal{I}}$ information are transformed into global state embedding s_t^0 and agent-specific state embeddings $\{x_t^i\}_{i\in\mathcal{I}}$ through the Inner Transformer Block. Subsequently, the perception of the leaders' actions by agents is achieved through the Outer Transformer Block. Each agent generates its action $\{a_t^i\}_{i\in\mathcal{I}}$ and sub-game state value function $\{V_t^i(s_t^i)\}_{i\in\mathcal{I}}$ in an autoregressive manner according to their priority level. ***Right***: Decentralized policy learning based on knowledge distillation.

simultaneously acting as a leader to lower-level agents. For followers, these inferior agents receive decision information from superior agents during both the execution and training procedures. The policy gradients of the agents are then updated in the direction of the optimal response to leaders, yielding an approximation of the solution to the inner optimization problem posed by Equation (4). On the other hand, for leaders, these superior agents interact with the environment and perceive the reaction of the inferior agents. When updating their policies, leaders consider followers as part of the surrounding environment and maximize their own private rewards, resulting in an approximate solution to the outer optimization problem in Equation (3).

Under SDM, agents strive to maximize their individual returns based on known conditions, which alignes with the objective of RL. Through iterative interaction and trial-and-error with the environment, they ultimately converge to the SE policies. Formally, agent i executes policy based on sub-game state $s_t^i = (s_t, a_t^1, ..., a_t^n)$, and the corresponding action value function is denoted as:

$$Q_{\boldsymbol{\pi}}^i(s, \boldsymbol{a}^{1:i-1}, a^i) = \mathbb{E}_{s\sim\rho,\boldsymbol{a}\sim\boldsymbol{\pi}}\big[\sum_{t=0}^{\infty}\gamma^t\cdot r_t^{h^i}(s_t,\boldsymbol{a}_t)\mid \mathrm{s}_0 = s, \boldsymbol{a}_0^{1:i} = \boldsymbol{a}^{1:i}\big]. \quad (5)$$

We also have the state value function: $V_{\boldsymbol{\pi}}^i(s, \boldsymbol{a}^{1:i-1}) = \sum_{a^i\in\mathcal{A}^i}\pi^i(a^i|s,\boldsymbol{a}^{1:i-1})\cdot Q_{\boldsymbol{\pi}}^i(s,\boldsymbol{a}^{1:i-1},a^i)$. The advantage function is represented as:

$$A_{\pi}^i(s, \boldsymbol{a}^{1:i-1}, a^i) = Q_{\boldsymbol{\pi}}^i(s, \boldsymbol{a}^{1:i-1}, a^i) - V_{\boldsymbol{\pi}}^i(s, \boldsymbol{a}^{1:i-1}). \quad (6)$$

Introducing SDM provides several advantages for resolving coordination problems as opposed to assuming that agents act simultaneously. For better illustration, we consider a simple two-agent three-action multi-step matrix game as shown in Figure 2. In this game, agents must make a choice between actions (a_1^1, a_1^2) and (a_3^1, a_3^2) at the initial state and repeat the same joint action until they coordinate at the terminal state to receive the final reward. Incorrect choices lead to game termination, necessitating a restart from the beginning. Despite the game's simplicity, successful decision-making requires full cooperation and coordination between agents to ensure the maximum total return.

When making decisions using SDM, in the initial state, if agent 1 chooses action a_1^1 or a_3^1, the optimal strategy for agent 2 is uniquely determined as a_1^2 or a_3^2. As illustrated in Figure 1, the hierarchical decision-making process in SDM is similar to a depth-first search tree. The ideal space for the follower to act is narrowed down when the leaders commits to their actions. This constraint reduces the risk of both players pursuing disparate optimal strategies (a_1^1, a_3^2) or (a_3^1, a_1^2) in the initial state, which may result in game failure. Secondly, from a game theory perspective, all three joint-actions $(a_1^1, a_3^2), (a_2^1, a_2^2), (a_3^1, a_1^2)$ are NE points in the final state, wherein neither player can increase its payoff by changing its own strategy. However, only point (a_3^1, a_1^2) is the unique SE point, which results in the highest average payoff for both players. SDM facilitates the identification of the SE point naturally. The detailed process for finding the SE point can be found in Appendix D.

5. Stackelberg Decision Transformer

In this section, we develop a dual Transformer architecture called Stackelberg Decision Transformer (STEER), which

combines SDM and autoregressive sequence models. Specifically, as illustrated in Figure 3, STEER employs the Inner and Outer Transformer Blocks. The Inner Transformer Block (ITB) is tasked with processing state information across different environmental configurations, while the Outer Causal Transformer Block (OTB) handles decision information in an autoregressive manner for subsequent fitting of policy and value functions.

Inner Transformer Block. ITB consists of $n+1$ tokens, where agents' observation vectors $\{o_t^i\}_{i\in\mathcal{I}}$ are initially mapped to embeddings $\{e_t^i\}_{i\in\mathcal{I}}$ as the last n tokens. Regarding the first token:

- If additional global state information s_t is accessible, the state embedding e_t^0 is utilized as the first token.
- If only local observation information is available, an extra learnable embedding e_t^0 is applied as the first token, similar to the class token in ViT (Dosovitskiy et al., 2020).

Consequently, the input to ITB is represented as $\boldsymbol{e}_{l_0,t} = [e_t^0, e_t^1, ..., e_t^n] + \boldsymbol{E}_{pos}$, where $\boldsymbol{E}_{pos}$ represents the position embedding. Using multi-head self-attention (MHSA), multilayer perceptron (MLP) and layer normalization (LN), we can write the j-th block of ITB as:

$$\boldsymbol{e}'_{\ell_j,t} = \mathrm{MHSA}(\mathrm{LN}(\boldsymbol{e}_{\ell_{j-1},t})) + \boldsymbol{e}_{\ell_{j-1},t}, \tag{7}$$

$$\boldsymbol{e}_{\ell_j,t} = \mathrm{MLP}(\mathrm{LN}(\boldsymbol{e}'_{\ell_j,t})) + \boldsymbol{e}'_{\ell_j,t}. \tag{8}$$

Assuming a total of L blocks, the output of ITB can be written as:

$$\boldsymbol{Y}_t^{ITB} = \mathrm{MLP}(\boldsymbol{e}_{L,t}) = [s_t^0, x_t^1, ..., x_t^n]. \tag{9}$$

Here, s_t^0 is the global game state embedding, which is encoded by the output of the first token at the last block. $\{x_t^i\}_{i\in\mathcal{I}}$ represents agent-specific state embedding for all agents (Chen et al., 2022). ITB offers a flexible and adaptable methodology for handling various environmental state configurations. It facilitates the production of precise abstract representations of game scenarios.

Outer Transformer Block. In OTB, s_t^0 functions as the abstract representation of the global state. Together with the actions $\{a_t^i\}_{i\in\mathcal{I}}$ taken by each prioritized level agent, it constitutes the input sequence $\boldsymbol{z}_{0,t} = [s_t^0, a_t^1, ..., a_t^{n-1}]$ with the length of n. The input of the first block in OTB can be expressed as $\boldsymbol{z}_{l_0,t} = \mathrm{MLP}(\boldsymbol{z}_{0,t}) + \boldsymbol{E}_{pos}$. Subsequently, OTB utilizes masked multi-head self-attention (MMHSA) to generate decision information in an autoregressive manner. Similar to ITB, this process is summarized as:

$$\boldsymbol{z}'_{\ell_j,t} = \mathrm{MMHSA}(\mathrm{LN}(\boldsymbol{z}_{\ell_{j-1},t})) + \boldsymbol{z}_{\ell_{j-1},t}, \tag{10}$$

$$\boldsymbol{z}_{\ell_j,t} = \mathrm{MLP}(\mathrm{LN}(\boldsymbol{z}'_{\ell_j,t})) + \boldsymbol{z}'_{\ell_j,t}, \tag{11}$$

$$\boldsymbol{Y}_t^{OTB} = \mathrm{MLP}(\boldsymbol{z}_{L,t}). \tag{12}$$

OTB plays a critical role in SE strategy learning by effectively managing the process of sequential asynchronous action coordination through autoregressive generation of decision information for each agent.

Actor and Critic Heads. By combining the current state information of each agent with decision information from leaders, the current sub-game state embedding can be created and denoted as $\{s_t^i\}_{i\in\mathcal{I}} = \boldsymbol{Y}_t^{ITB}[1:n] + \boldsymbol{Y}_t^{OTB}[0:n-1]$. This embedding is then forwarded to the Critic head (CH) and Actor head (AH) to recursively approximate the value and policy functions of agents:

$$V_t^i(s_t^i) = V_t^i(s_t, a_t^{1:i-1}) = \mathrm{CH}(s_t^i), \tag{13}$$

$$a_t^i \sim \pi_t^i(s_t^i) = \mathrm{AH}(s_t, a_t^{1:i-1}). \tag{14}$$

Training Paradigm. Our approach is trained through end-to-end RL. Given the advantages of policy-based methods in addressing continuous control tasks and mixed tasks, it is appropriate to employ Proximal Policy Optimization (PPO) (Schulman et al., 2017) as the underlying algorithm. Assuming that the Transformer blocks (serve as both the policy and value networks), Actor head and Critic head are parameterized by ω, θ, ϕ, respectively, the policy network needs to maximize the clipping objective function:

$$\begin{aligned}\mathcal{L}(\theta,\omega) =& \mathbb{E}_{t,i}[\min(r_{\theta,\omega}^i \hat{A}_\pi^i, clip(r_{\theta,\omega}^i, 1\pm\epsilon)\hat{A}_\pi^i) + \\ & \eta S(\pi_{\theta,\omega}^i(s, a^{1:i-1}))],\end{aligned} \tag{15}$$

where $r_{\theta,\omega}^i = \frac{\pi_{\theta,\omega}^i(a^i|s,a^{1:i-1})}{\pi_{\theta_{old},\omega_{old}}^i(a^i|s,a^{1:i-1})}$, ϵ is the clipping ratio, $\hat{A}_\pi^i$ serves as an estimation of the advantage value in Equation (6), $S(\cdot)$ is the Shannon entropy and η is its coefficient. Furthermore, the value network is updated through the minimization of empirical Bellman TD-error:

$$\begin{aligned}\mathcal{L}(\phi,\omega) =& \mathbb{E}_{t,i}[\max((V_{\phi,\omega}^i(s^i) - R^i)^2, (\mathrm{clip}(V_{\phi,\omega}^i(s^i), \\ & V_{\phi_{old},\omega_{old}}^i(s^i) \pm \varepsilon) - R^i)^2)],\end{aligned} \tag{16}$$

where ε is the clipping ratio and R^i is the discounted cumulative return.

It is worth noting that the process of action generation in the execution phase differs from that in the training phase. Specifically, during the execution phase, actions are generated autoregressively. In contrast, during the training phase, the joint action sequence of the agents is captured and stored in the replay buffer. This enables parallel computation and updating, leading to significantly increased training speed compared to other SG based learning techniques.

Scalability for Decentralized Execution Systems. The inherent properties of the Transformer architecture constrain its applicability of decentralized execution. Accordingly, STEER is employed as a centralized approach. To further

Table 1. Comparison of environment configurations in different scenarios.

Environment	1-step Matrix Game		n-step Matrix Game		Multi-Agent MuJoCo (MA-MuJoCo)	Google Research Football (GRF)	Highway On-Ramp Merging (HORM)
	Penalty	Mixing	Cooperation	Coordination			
Complete Collaboration	✔	✘	✔	✘	✔	✔	✘
Incomplete Collaboration	✘	✔	✘	✔	✘	✘	✔
Continuous Control	✘	✘	✘	✘	✔	✘	✘
Discrete Control	✔	✔	✔	✔	✘	✔	✔
Partial Observation	✘	✘	✘	✘	✔	✘	✘
Global State Sharing	✘	✘	✘	✘	✔	✘	✔
Individual Global State	✘	✘	✔	✔	✘	✔	✘
Constant State	✔	✔	✘	✘	✘	✘	✘

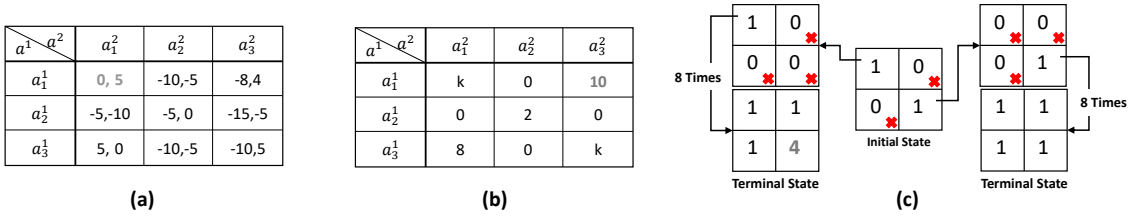


Figure 4. Matrix game scenarios. (a) Mixing. (b) Penalty ($k <= 0$). (c) Multi-step matrix game: Cooperation. Additionally, Multi-step matrix game: Coordination is also evaluated, which can be found in Figure 2.

broaden its applicability, we introduce an additional knowledge distillation module (Gou et al., 2021) and devise a student network comprised of MLP that receives local observations and outputs corresponding actions (see Figure 3). After the convergence of the STEER, we align the outputs of each student network with those of STEER, facilitating the learning of decentralized SE strategies by the student network. Specifically, we choose Logarithmic Root Mean Squared Error and add Shannon entropy loss:

$$\mathcal{L}_{\text{stu}} = \sqrt{\frac{1}{m}\sum_{k=1}^{m}\left(\log(\pi_{\text{stu}}(\bar{a}_k \mid o_k)) - \log(\pi_{\text{STEER}}(\bar{a}_k \mid s_k)\right)^2} - \eta S\left(\pi_{\text{stu}}(a \mid o_k)\right), \tag{17}$$

where $\bar{a}_k = argmax_a\, \pi_{\text{STEER}}(a \mid s_k)$ and m denotes the size of the replay buffer. It ensures that the student selects the STEER action with the same probability. The learning difficulty is reduced when focusing on the learning of the optimal action probabilities instead of fitting the complete policy distribution. The increase of Shannon entropy, on the other hand, prevents the student policy distribution from exhibiting another peak outside of the optimal action. It disperses the excess probability mass across sub-optimal actions, thereby ensuring the maximum probability is allocated to the action endorsed by the teacher network.

6. Evaluation

In this section, a comprehensive evaluation and analysis of the proposed STEER method is presented to validate several key aspects. These aspects encompass: (1) the method's capacity to identify SE solutions, (2) its adaptability to diverse environmental configurations, (3) the computational overhead of the algorithm, (4) the effectiveness of the decentralized execution scheme, (5) the functionality of each module and (6) the significance of agent priority allocation.

6.1. Experimental Settings

Evaluation Environments. We assess STEER's ability to converge to SE solutions in both single-step and multi-step matrix game scenarios, as shown in Figure 4. Moreover, we investigate the performance of STEER in complex scenarios encompassing Multi-Agent MuJoCo (MA-MuJoCo) (Peng et al., 2021), Google Research Football (GRF) (Kurach et al., 2020) and Highway On-Ramp Merging (HORM) (Zhang et al., 2023b). As depicted in Table 1, these benchmarks encompass nearly all types of task scenarios, including fully cooperative and mixed tasks, continuous and discrete control tasks, as well as tasks involving shared state and individual observations. Appendix B contains more thorough descriptions of these environments.

Baseline Algorithms. We compare STEER to various advanced and comparable policy-based MARL methods, including MAPPO (Yu et al., 2021), which is one of the most famous baselines in MARL, HAPPO (Kuba et al., 2022), which is specifically designed for heterogeneous policy learning, MAT (Wen et al., 2022), which is built upon Transformer architecture, and STEP (Zhang et al., 2023b), which is based on SG.

6.2. Finding SE Solutions

Main Results. The results in matrix game scenarios are able to offer an intuitive demonstration of whether algo-

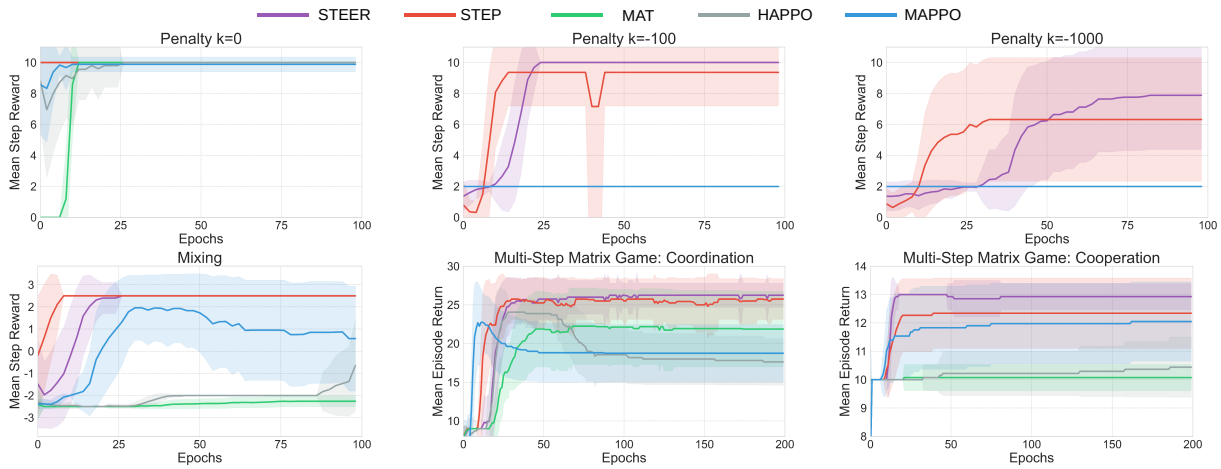


Figure 5. The average return of the two agents. A single standard deviation over trials is shaded.

Table 2. The percentage that converges to the optimal strategy in matrix game scenarios over 100 trials. STEER demonstrates the most stable convergence outcomes.

	Penalty			Mixing	Coordination	Cooperation
	k=0	k=-100	k=-1000			
STEER	**100%**	**100%**	**72%**	**100%**	**95%**	**96%**
STEP	**100%**	93%	44%	**100%**	94%	90%
MAT	**100%**	0%	0%	0%	46%	5%
HAPPO	**100%**	0%	0%	28%	6%	19%
MAPPO	95%	0%	0%	63%	14%	65%

rithms converge to their corresponding equilibrium strategies. As shown in Figure 5, STEER outperforms other methods in all scenarios. Table 2 also demonstrates that STEER consistently converges to SE solutions with the highest probability across all scenes. For instance, in the Penalty scenario, any deviation from the optimal strategy by an agent results in severe punishment for the other agent who has made the correct decision. As the penalty term k increases, it becomes increasingly difficult for agents to learn the optimal strategy. Therefore, only action a_2 remains uninfluenced by penalty terms, and all methods, converge to the sub-optimal NE (a_2^1, a_2^2) with a 100% probability when $k < 0$ except for STEP and STEER. In addition, these methods fail to converge to stable results in other scenarios.

Method Comparison. While STEP is designed to learn SE strategies and demonstrates excellent performance in matrix game scenarios, STEER exhibits superiority over it in terms of network expressive capacity. For example, STEER maintains its effectiveness even when subjected to extreme values such as $k = -1000$ in the Penalty scenario. This distinction becomes particularly evident in the results showed in the subsequent analysis of complex scenarios. Moreover, despite MAT employing a similar sequential decision structure, it fails to generate optimal outcomes. This can be attributed to MAT solely relying on agents' local observation data instead of considering the sub-game state when approximating the value function. Consequently, this approach lead to erroneous guidance for actor updates, ultimately resulting in poor outcomes. In addition, all methods except STEER and STEP are ineffectual when agents possess private rewards and must coordinate their actions.

6.3. Performance in Complex Scenarios

Figure 6 illustrates the experimental results of all methods across 3 tasks and 9 scenarios. It can be seen that STEER demonstrates advantages over existing state-of-the-art methods across all scenarios, achieving higher sample efficiency (faster convergence speed) and improved final performance. These results highlight the superiority and adaptability of STEER in confronting complex scenarios.

Adaptability to Diverse Environmental Configurations. The three testing tasks are characterized by distinct state and reward configurations. From the perspective of the state, most methods are typically designed for specific environmental setups. For instance, MAT utilizes a standard encoder-decoder structure, yielding promising outcomes in independent observable environments (GRF & MA-Mujoco). Nevertheless, it faces challenges in shared-state environments (HORM), where the encoder's outputs may exhibit similarities and be employed as query values in the decoder. This has a significant negative impact on the self-attention mechanism. Conversely, STEP and HAPPO are better suited for global state sharing environments. In contrast, STEER effectively handles all these types of tasks.

From the perspective of the reward, STEER exhibits optimal performance in both fully cooperative tasks and mixed tasks. In HORM, for example, one must first observe whether the vehicles on the main road are slowing down before deciding whether to merge into the lane. Each agent wishes

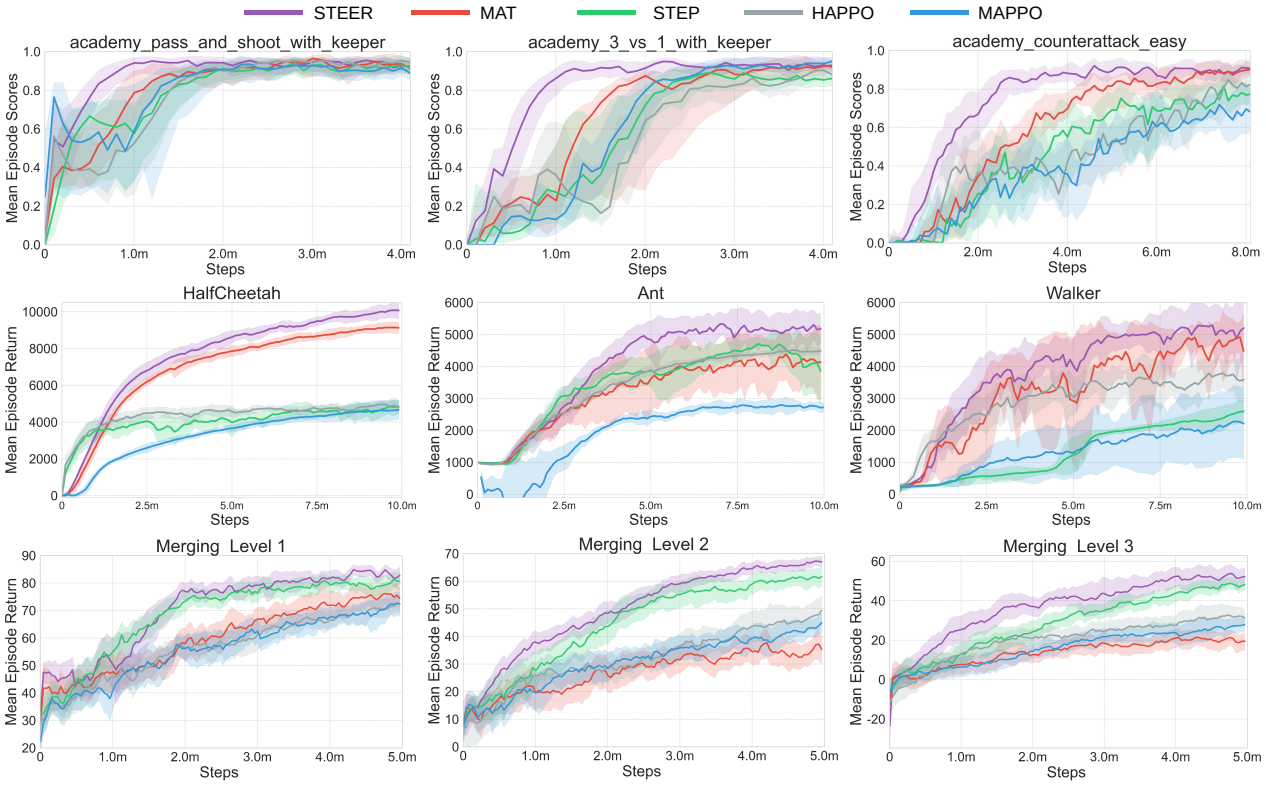


Figure 6. The evaluation performance on GRF (top), MA-MuJoCo (middle) and HORM (bottom). Error bars are a 95% confidence interval across 5 runs.

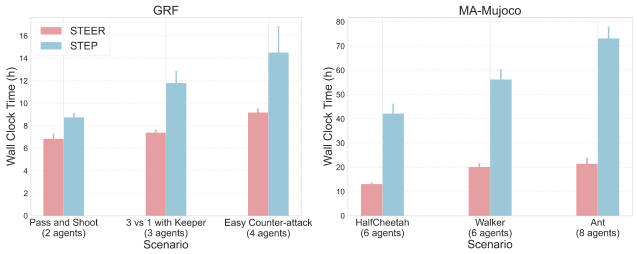


Figure 7. The wall clock time required for 10 million training steps.

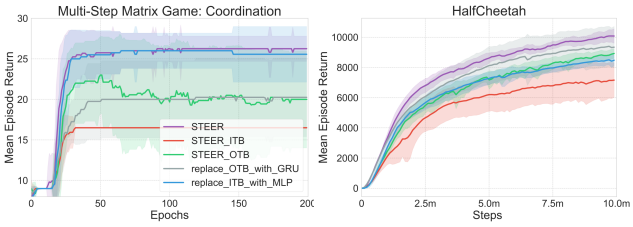


Figure 8. Performance comparison for different model architectures to explore the functionality of each component.

to pass through the intersection as quickly as possible while avoiding collisions, and they receive individual rewards from the environment. In such mixed tasks, the introduction of SDM is imperative, as agents must possess the ability to perceive other agents in order to make optimal decisions. Therefore, STEER achieves the best performance.

Computational Overhead. As a heterogeneous strategy learning approach, SG-based methods require sequentially updating the policy of each agent, often resulting in extended training time, which becomes unacceptable as the number of agents increases. However, the Transformer architecture in STEER enables parallel training, thereby achieving superior performance within significantly reduced training time. As illustrated in Figure 7, while the actual training time is influenced by various factors, it is evident that STEER exhibits a significant advantage, particularly when the number of agents is larger.

Performance of Decentralized Execution. We verify the effectiveness of our decentralized execution scheme on GRF experiments and conduct 5 runs in each of three scenarios. Table 3 clearly demonstrates that the student networks have achieved a performance level that closely approximates that of STEER. Interestingly, in some cases, the performance of the student networks can even marginally surpass that of STEER, which is acceptable considering the inconsistency between their policy distributions and the uncertainty

Table 3. Performance of decentralized execution in GRF. The values in parentheses correspond to a single standard deviation over trials.

	academy_pass_and_shoot_with_keeper	academy_3_vs_1_with_keeper	academy_counterattack_easy
STEER	0.9339(0.0358)	0.9636(0.0375)	0.9176(0.0815)
Decentralized Student Network	0.9426(0.0143)	0.9417(0.0203)	0.9025(0.0190)

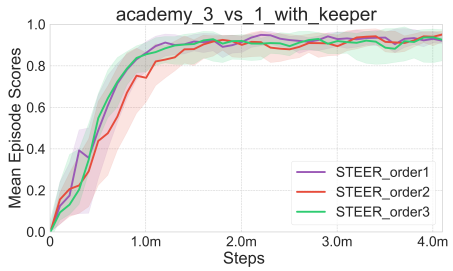


Figure 9. The evaluation performance of STEER in GRF with three priority order.

of neural networks. In conclusion, decentralized execution realized via knowledge distillation ensures maintained algorithmic performance.

6.4. Ablation Studies

In order to further investigate the functionality of each module, we perform ablation experiments. This involves replacing the ITB with a simple MLP (*replace_ITB_with_MLP*), using a Recurrent Neural Network to replace the OTB (*replace_OTB_with_GRU*), and fitting the value and policy functions directly with the output from either the ITB (*STEER_ITB*) or the OTB (*STEER_OTB*).

Functionality of Each Module. The ITB and OTB modules are respectively charged with processing state information and leaders' decision information, thereby jointly comprising an abstract representation of the current subgame state. The experimental results, as shown in Figure 8, indicate that each module plays an indispensable role in facilitating the performance of the algorithm. In matrix game scenarios, the state configuration is simple, but it demands high perception of decision information. As a result, *replace_ITB_with_MLP* seems to result in marginal performance differences and *replace_OTB_with_GRU* yields poor performance. In contrast, when dealing with MA-MuJoCo, where agents have partial observability and require heightened perception of state information, *replace_ITB_with_MLP* leads to a significant decline in performance compared to *replace_OTB_with_GRU*. Moreover, relying solely on ITB (*STEER_ITB*) for decision-making in matrix game scenarios is equivalent to neglecting the SG structure, while exclusively relying on OTB (*STEER_OTB*) for decision-making in complex scenarios implies agents lacking adequate perception of the current environment. Both approaches lead to poor performance.

Priority Allocation. We assess the performance of STEER in scenarios using three distinct priority arrangements. The results shown in Figure 9 indicate that the predefined ordering has minimal impact on the experimental outcomes. However, it is conceivable that the optimal priority arrangement may vary across different time steps. Consequently, we assert that adaptive priority determination in the decision-making process will be a promising direction.

7. Conclusion

Our core insight is that the hierarchical decision-making structure of SG aligns perfectly with the modeling approach of autoregressive sequence models. Building upon this, we introduce the Stackelberg Decision Transformer method to solve the SE strategies for coordination tasks in MARL. Compared to previous work, our approach offers a more systematic and scholarly foundation for investigating the intricacies of MARL, as well as a more comprehensive training paradigm. Additionally, our method is more flexible in handling different environmental configurations, making it more applicable and scalable across different scenarios. To the best of our knowledge, we are the first to propose using an autoregressive sequence model to solve SE. We firmly believe that our method has broad potential for applications in the MARL community.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

Acknowledgments

This project is supported by the Strategic Priority Research Program of the Chinese Academy of Sciences, Grant No. XDA27050102

References

Başar, T. and Olsder, G. J. *Dynamic noncooperative game theory*. SIAM, 1998.

Chen, L., Lu, K., Rajeswaran, A., Lee, K., Grover, A., Laskin, M., Abbeel, P., Srinivas, A., and Mordatch, I. Decision transformer: Reinforcement learning via sequence

modeling. *Advances in neural information processing systems*, 34:15084–15097, 2021.

Chen, Y., Mao, H., Zhang, T., Wu, S., Zhang, B., Hao, J., Li, D., Wang, B., and Chang, H. Ptde: Personalized training with distillated execution for multi-agent reinforcement learning. *arXiv preprint arXiv:2210.08872*, 2022.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Ding, Z., Su, K., Hong, W., Zhu, L., Huang, T., and Lu, Z. Multi-agent sequential decision-making via communication. *arXiv preprint arXiv:2209.12713*, 2022.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Gerstgrasser, M. and Parkes, D. C. Oracles & followers: Stackelberg equilibria in deep multi-agent reinforcement learning. In *International Conference on Machine Learning*, pp. 11213–11236. PMLR, 2023.

Gou, J., Yu, B., Maybank, S. J., and Tao, D. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129:1789–1819, 2021.

He, W., Yao, H., Mai, T., Wang, F., and Guizani, M. Three-stage stackelberg game enabled clustered federated learning in heterogeneous uav swarms. *IEEE Transactions on Vehicular Technology*, 2023.

Hu, J. and Wellman, M. P. Nash q-learning for general-sum stochastic games. *J. Mach. Learn. Res.*, 4:1039–1069, 2003.

Hu, S., Zhu, F., Chang, X., and Liang, X. Updet: Universal multi-agent reinforcement learning via policy decoupling with transformers. *arXiv preprint arXiv:2101.08001*, 2021.

Janner, M., Li, Q., and Levine, S. Offline reinforcement learning as one big sequence modeling problem. *Advances in neural information processing systems*, 34: 1273–1286, 2021.

Könönen, V. Asymmetric multiagent reinforcement learning. *Web Intell. Agent Syst.*, 2(2):105–121, 2004.

Kuba, J. G., Chen, R., Wen, M., Wen, Y., Sun, F., Wang, J., and Yang, Y. Trust region policy optimisation in multi-agent reinforcement learning. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022.

Kurach, K., Raichuk, A., Stańczyk, P., Zając, M., Bachem, O., Espeholt, L., Riquelme, C., Vincent, D., Michalski, M., Bousquet, O., et al. Google research football: A novel reinforcement learning environment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 4501–4510, 2020.

Liu, R., Gao, J., Zhang, J., Meng, D., and Lin, Z. Investigating bi-level optimization for learning and vision from a unified perspective: A survey and beyond. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44 (12):10045–10067, 2021a.

Liu, Z., Wan, L., Sun, K., Lan, X., et al. Multi-agent intention sharing via leader-follower forest. *arXiv preprint arXiv:2112.01078*, 2021b.

Lowe, R., Wu, Y., Tamar, A., Harb, J., Abbeel, P., and Mordatch, I. Multi-agent actor-critic for mixed cooperative-competitive environments. In Guyon, I., von Luxburg, U., Bengio, S., Wallach, H. M., Fergus, R., Vishwanathan, S. V. N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 6379–6390, 2017.

Lu, Y. and Yan, K. Algorithms in multi-agent systems: a holistic perspective from reinforcement learning and game theory. *arXiv preprint arXiv:2001.06487*, 2020.

Mao, H., Zhao, R., Li, Z., Xu, Z., Chen, H., Chen, Y., Zhang, B., Xiao, Z., Zhang, J., and Yin, J. Pdit: Interleaving perception and decision-making transformers for deep reinforcement learning. *arXiv preprint arXiv:2312.15863*, 2023.

Meng, L., Wen, M., Yang, Y., Le, C., Li, X., Zhang, W., Wen, Y., Zhang, H., Wang, J., and Xu, B. Offline pre-trained multi-agent decision transformer: One big sequence model conquers all starcraftii tasks. *arXiv preprint arXiv:2112.02845*, 2021.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

Peng, B., Rashid, T., de Witt, C. S., Kamienny, P., Torr, P. H. S., Boehmer, W., and Whiteson, S. FACMAC: factored multi-agent centralised policy gradients. In Ranzato, M., Beygelzimer, A., Dauphin, Y. N., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 12208–12221, 2021.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Rashid, T., Samvelyan, M., de Witt, C. S., Farquhar, G., Foerster, J. N., and Whiteson, S. QMIX: monotonic value function factorisation for deep multi-agent reinforcement learning. In Dy, J. G. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pp. 4292–4301. PMLR, 2018.

Ruan, J., Meng, L., Xiong, X., Xing, D., and Xu, B. Learning multi-agent action coordination via electing first-move agent. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 32, pp. 624–628, 2022.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Seraj, E., Chen, L., and Gombolay, M. C. A hierarchical coordination framework for joint perception-action tasks in composite robot teams. *IEEE Transactions on Robotics*, 38(1):139–158, 2021.

Shen, R., Zhong, S., Wen, X., An, Q., Zheng, R., Li, Y., and Zhao, J. Multi-agent deep reinforcement learning optimization framework for building energy system with renewable energy. *Applied Energy*, 312:118724, 2022.

Sinha, A., Malo, P., and Deb, K. A review on bilevel optimization: From classical to evolutionary approaches and applications. *IEEE Transactions on Evolutionary Computation*, 22(2):276–295, 2017.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

von Oswald, J., Henning, C., Sacramento, J., and Grewe, B. F. Continual learning with hypernetworks. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

Von Stackelberg, H. *Market structure and equilibrium*. Springer Science & Business Media, 2010.

Wang, X., Tian, Z., Wan, Z., Wen, Y., Wang, J., and Zhang, W. Order matters: Agent-by-agent policy optimization. In *The Eleventh International Conference on Learning Representations*, 2023.

Wen, M., Kuba, J., Lin, R., Zhang, W., Wen, Y., Wang, J., and Yang, Y. Multi-agent reinforcement learning is a sequence modeling problem. *Advances in Neural Information Processing Systems*, 35:16509–16521, 2022.

Xiao, Y., Tan, W., and Amato, C. Asynchronous actor-critic for multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 35:4385–4400, 2022.

Xu, Z., Zhang, B., Li, D., Zhou, G., Zhang, Z., and Fan, G. Dual self-awareness value decomposition framework without individual global max for cooperative multi-agent reinforcement learning. *arXiv preprint arXiv:2302.02180*, 2023.

Yu, C., Velu, A., Vinitsky, E., Wang, Y., Bayen, A., and Wu, Y. The surprising effectiveness of ppo in cooperative, multi-agent games. *arXiv preprint arXiv:2103.01955*, 2021.

Zhang, B., Bai, Y., Xu, Z., Li, D., and Fan, G. Efficient policy generation in multi-agent systems via hypergraph neural network. In *Neural Information Processing: 29th International Conference, ICONIP 2022, Virtual Event, November 22–26, 2022, Proceedings, Part II*, pp. 219–230. Springer, 2023a.

Zhang, B., Li, L., Xu, Z., Li, D., and Fan, G. Inducing stackelberg equilibrium through spatio-temporal sequential decision-making in multi-agent reinforcement learning. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, pp. 353–361, 2023b.

Zhang, B., Mao, H., Ruan, J., Wen, Y., Li, Y., Zhang, S., Xu, Z., Li, D., Li, Z., Zhao, R., et al. Controlling large language model-based agents for large-scale decision-making: An actor-critic approach. *arXiv preprint arXiv:2311.13884*, 2023c.

Zhang, H., Chen, W., Huang, Z., Li, M., Yang, Y., Zhang, W., and Wang, J. Bi-level actor-critic for multi-agent coordination. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pp. 7325–7332. AAAI Press, 2020.

A. Implementation Details

A.1. Pseudo Code of STEER

Algorithm 1 Stackelberg Decision Transformer

Hyperparameters: Learning rate α, batch size B, number of episodes K, length of episode L, number of agents n.
Initialize: Inner and outer Transformer blocks parameters ω_0, actor head parameters θ_0, critic head parameters ϕ_0, replay buffer $\mathcal{B}$.

for episode $k = 1$ **to** K **do**
  for time step $t = 1$ **to** T **do**
    Rank agents by priority ID
    Collect a sequence of environmental state information and generate representation sequence $[s_t^0, x_t^1, ..., x_t^n]$ by feeding it to the inner transformer block (ITB).
    for agent $i = 1$ **to** N **do**
      Input $s_t^0, a_t^1, ..., a_t^{i-1}$ to the outer transformer block (OTB)
      Add the outputs of ITB and OTB to generate sub-game state s_t^i.
      Sample action a_t^i according to the actor head $a_t^i \sim \pi_t^i(s_t^i; \theta_t, \omega_t)$.
      Calculate state value $v_t^i = V_t^i(s_t^i; \phi_t, \omega_t)$.
    end for
    Execution the joint action $\mathbf{a}_t = \{a_t^1, a_t^2, \ldots, a_t^n\}$, obtain reward $\boldsymbol{r}_t$ and state $\boldsymbol{s}_t$.
  end for
  Push transitions $\{(\boldsymbol{s}_t, \boldsymbol{a}_t, \boldsymbol{r}_t, \boldsymbol{s}_{t+1})\}_{t \in \mathcal{T}}$ into $\mathcal{B}$.
  Sample a random minibatch of M transitions from $\mathcal{B}$.
  Compute advantage estimate $\hat{A}$ via GAE.
  Parallel generation policy $\boldsymbol{\pi}_{\theta,\omega}$.
  Update actor by maximizing objective function in Eq 15.
  Update critic by minimizing the loss in Eq 16.
end for

B. Environment details

The adaptability to a wide range of environmental configurations constitutes a significant contribution of our approach. The meticulously constructed ITB structure employed in our methodology facilitates the handling of a diverse set of state information, encompassing tasks with partially observable or globally shared states. Furthermore, the utilization of a policy-based approach empowers us to effectively address both continuous and discrete control tasks. Simultaneously, the employment of the SG-based modeling form enables the processing of both complete and incomplete collaboration tasks.

B.1. Google Research Football

Google Research Football (GRF) aims to establish a benchmark for artificial intelligence (AI) in football. The objective is to develop a trustworthy and standardized evaluation framework for football game. This environment, which is based on a football game and a physics engine, requires agents to act swiftly and implement cutting-edge cooperative techniques. In a multi-agent environment, each agent controls one team member who learns fundamental skills and team coordination in order to overcome the opposition's defense and score goals. In our experiments, each agent independently observes the global environment and shares a global reward.

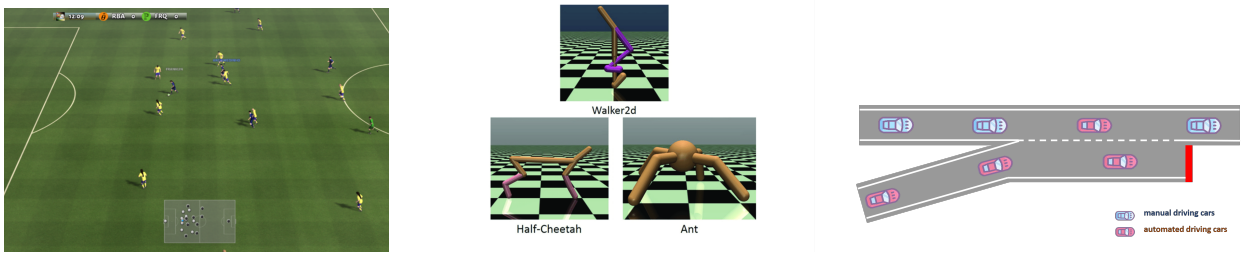
B.2. Multi-Agent MuJoCo

Multi-Agent MuJoCo (MA-MuJoCo) is a software platform designed for simulating and controlling multiple interacting physical agents. This platform allows multiple agents to control separate joints to operate a single robot. As an advanced version of MuJoCo, it is capable of examining the behavior and interactions of multiple agents in a complex environment. Furthermore, it can be used to test and create multi-agent control algorithms that can handle the increased scale and complexity by simulating a variety of physical phenomena. In this study, we utilized Ant 8x1, HalfCheetah 6x1, and Walker 6x1 scenarios to compare STEER with other baselines. The objective is to actuate each joint of the robot to move

forward steadily. The state is composed of the locations, velocities, and accelerations of each joint, while the action is the intended torque to be applied to each joint. Additionally, all agents receive shared rewards as a fully cooperative task. We employed two commonly used environmental configurations: all agents sharing a global state and each agent possessing an independent local observation and providing a centralized global state interface. Figure 6 illustrates the optimal performance of all methods under both state configurations.

B.3. Highway On-Ramp Merging

Highway on-ramp merging (HORM) is a challenging task that involves merging on-ramps while accommodating both manual and automated driving conditions. This is considered one of the most difficult tasks of automatic driving. As shown in Figure, on-ramp cars must merge into the through lane without accidents. Ideally, vehicles on the main road should adjust their speed to provide sufficient space for the vehicles on the entrance ramp to merge safely. Similarly, vehicles on the entrance ramp should enter the main road quickly while adjusting their speed and ensuring safety. The vehicles can perform optional actions such as left and right turns, constant-speed cruising, acceleration, and deceleration. Each vehicle has the ability to monitor the horizontal and vertical coordinates and speed of all other vehicles. HORM is an environment where each intelligent agent possesses a private reward and shares a global state. In this context, the performance of MAT is unsatisfactory.



(a) Google Research Footbal (b) Multi-Agent MuJoCo (c) Highway On-Ramp Merging

Figure 10. Examples of the three experimental platforms.

C. Additional Experimental Results

C.1. Finding SE Solutions

Table 4. The average return of different methods in matrix games. The values in parentheses correspond to a single standard deviation over 100 trials. STEER demonstrates superior performance to other methods.

	Penalty			Mixing	Coordination	Cooperation
	k=0	k=-100	k=-1000			
STEER	**10.0(0)**	**10.0(0)**	**8.0(3.39)**	**2.5(0)**	**26.0(2.42)**	**12.88(0.58)**
STEP	**10.0(0)**	9.44(2.04)	5.52(3.97)	**2.5(0)**	25.9(2.37)	12.69(0.89)
MAT	**10.0(0)**	2.0(0)	2.0(0)	-2.68(0.25)	21.32(4.94)	10.15(0.65)
HAPPO	**10.0(0)**	2.0(0)	2.0(0)	-0.74(2.33)	17.62(2.94)	10.57(1.18)
MAPPO	9.90(0.43)	2.0(0)	2.0(0)	0.72(2.33)	19.17(4.05)	11.95(1.43)

C.2. Ablation Study

We show more ablation experiments in Figure 11. The primary function of ITB resides in its capacity for game state abstraction, employing an encoder structure that enables each agent to selectively attend to the state information of other agents, thereby facilitating more effective learning of state representations. In matrix game scenarios, agents receive a consistent state, whereas in GRF, agents receive individual global state observations. We posit that MLP's capabilities in these scenarios are adequate for achieving game abstraction functionality. Consequently, *replace_ITB_with_MLP* appears to yield marginal disparities in performance when compared to STEER. However, in MA-MuJoCo, where agents exhibit partial

observability, *replace_ITB_with_MLP* leads to a notable decline in performance. Correspondingly, replacing or deleting the OTB module can also cause performance degradation.

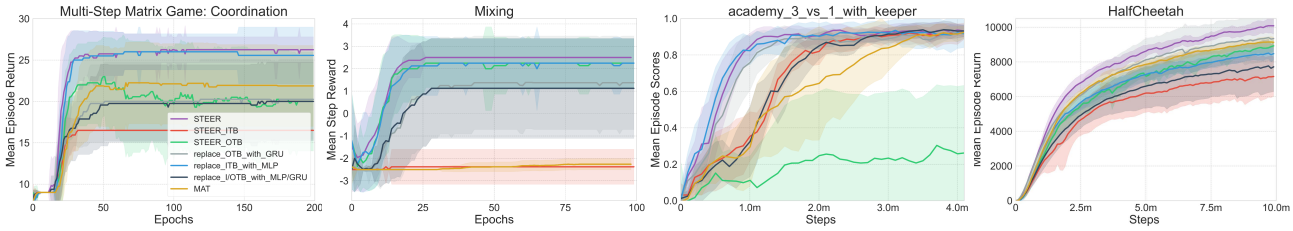


Figure 11. Performance comparison for different model architectures to explore the indispensability of each component.

C.3. Additional Experiments on Knowledge Distillation

As shown in Section 5, although our method is centralized, we provide a Knowledge Distillation solution for extending it to decentralized execution systems. We verify the effectiveness of this method on matrix game experiments and conducted 100 tests for each scenario. As shown in Table 5, the student network replicates 100% of STEER's performance in all matrix game scenarios.

Table 5. Performance comparison between STEER and student networks in Matrix games. The values in parentheses correspond to a single standard deviation over trials.

	Penalty			Mixing	Coordination	Cooperation
	k=0	k=-100	k=-1000			
STEER	10.0(0.00)	10.0(0.00)	7.84(3.55)	2.5(0.00)	12.97(0.30)	25.65(3.23)
Decentralized Sudent Network	10.0(0.00)	10.0(0.00)	7.84(3.55)	2.5(0.00)	12.97(0.30)	25.65(3.23)

C.4. Performance in Different Environment Configurations

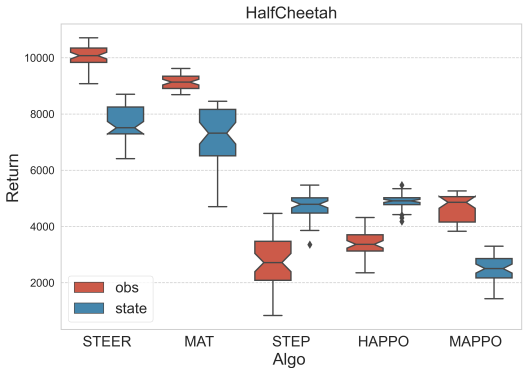
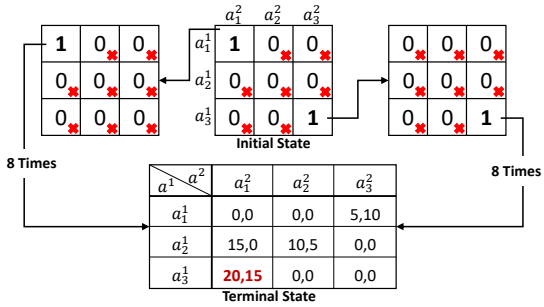


Figure 12. Comparing the final performance with different environment configurations.

In MA-MuJoCo, we investigate two environment configurations: one in which agents receive shared global state (*state*), and the other in which agents receive both independent local observations and global state (*obs*). The evaluation results of all methods across both configurations are presented in Figure 12. All baselines result in significant performance degradation after departing from their original settings. MAPPO and MAT, for instance, are designed for environments that entail local observation, with MAPPO's performance being severely impacted when all agents receive shared global states due to its parameter sharing. MAT employs an encoder to process state information and outputs it as the decoder's query value. The operation of the attention mechanism is also impacted by the inputs of the shared state. In contrast, STEER outperforms all

other methods in both configurations of the environment by processing environmental information flexibly based on the specific setting and maximizing the efficacy of state representation learning - a capability absent in other methods.

D. Details of Toy Example



a^1 \ a^2	a_1^2	a_2^2	a_3^2
a_1^1	0,0	0,0	5,10
a_2^1	15,0	10,5	0,0
a_3^1	20,15	0,0	0,0

Figure 13. Multi-step matrix game: Coordination. Only actions with non-zero rewards are permissible before reaching the terminal state.

Incorporating social norms or game structures into MARL is an effective approach to enhance collective intelligence. Under the Stackelberg decision mechanism, we assume that agent a^1 is the leader and agent a^2 is the follower. At the initial state, there are two optimal strategies, where the follower's unique optimal response is determined as a_1^2 or a_3^2 when the leader chooses a^1 or a^3, respectively. Similarly, at the final state, when a^1 commits to action a_1^1, the optimal response for a^2 is a_3^2; when a^1 commits to action a_2^1, the optimal response for a^2 is a_2^2; when a^1 commits to action a_3^1, the optimal response for a^2 is a_1^2. In these three cases, a^1 receives a reward of 5, 10, or 20, respectively, leading to the optimal strategy for a^1 to be action a_3^1. Therefore, the Stackelberg equilibrium solution is (a_3^1, a_1^2).

E. Additional discussion

E.1. Sequential Decision Making in Communication Mechanisms

Communication is an essential method to facilitate coordination. In MARL, research efforts pertaining to communication mechanisms have also explored similar sequence-based decision-making methods. SeqComm (Ding et al., 2022) and IS-LFF (Liu et al., 2021b) specifically concentrate on devising communication mechanism within the Leader-Follower framework. In contrast, our approach centers on acquiring SE strategies by incorporating the Stackelberg Decision Mechanism and autoregressive sequence models, resulting in the development of a centralized STEER method. Therefore, we conducted further investigations into the amalgamation of SG or Transformer with MARL.

E.2. Asynchronous Action Coordination and Simultaneous Decision-Making

In this paper, we employ an asynchronous action coordination approach based on the Stackelberg Game framework. We believe that asynchronous action coordination is an improvement over methods that presume agents make decisions simultaneously. The mainstream approach for modeling multi-agent decision-making tasks is to use Markov Game. However, this assumption of agents making decisions simultaneously appears counterintuitive. It results in agents relying only on trained tacit actions and not modifying their strategies based on information from their teammates or opponents. This approach resembles a prisoner's dilemma. By comparison, we propose using Stackelberg Game for modeling. Unless it is explicitly required that intelligent agents in MAS cannot obtain each other's actions (such as prisoner's dilemma settings), in almost all scenarios that allow centralized training, we can learn asynchronous action coordination strategies. Actually, two distinct categories of decision-making methods correspond to normal-form game and extensive-form game in game theory. In the HORM scenario, for example, agents can develop better response strategies by perceiving other agents' actions besides observing the current state.

In our proposed method, the follower is engaged in a conditional optimization problem wherein it acquires its own response strategy by leveraging the present state and decision information from leaders, with the aim of maximizing individual gains (Equation 2). This mechanism is effectively implemented through an autoregressive OTB framework. Following the fulfillment of the specified conditions, the optimization problem undergoes training via RL to acquire an approximate solution. That's to say, we successfully accomplish a integration of the optimization problem, the autoregressive sequence

model, as well as the RL method.

E.3. Stackelberg Decision Demonstration

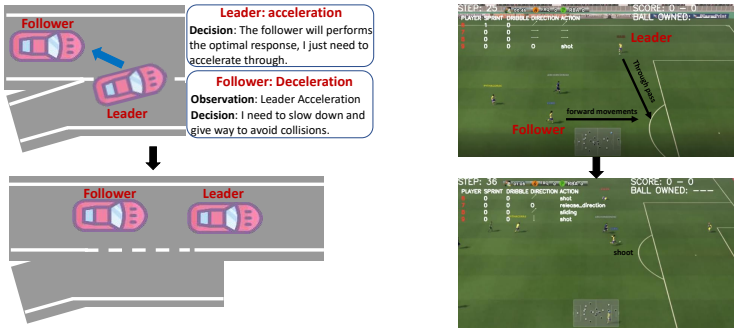


Figure 14. Stackelberg hierarchical coordination behavior demonstration.

In Figure 1, we present the decision form of the Stackelberg decision mechanism, which employs a hierarchical decision structure resembling a depth-first search tree. This structure simplifies the search space of child nodes considerably. In real-world scenarios, this form of decision-making is prevalent and intuitive. Specifically, in HORM, before deciding to merge into a lane, one must first assess whether the vehicles on the main road are decelerating. For a clearer presentation, as shown in Figure 14, we graphically represent the coordinated behavior of agents through the utilization of the Stackelberg decision mechanism in our experimental scenarios.

F. Experiments Details

F.1. Computing Infrastructure and Overhead

Our experiments are carried out on Nvidia GeForce RTX 3090 graphicscards and Intel(R) Xeon(R) Platinum 8280 CPU. Although we employ ITB and OTB, we limit the number of blocks per Transformer structure to one or two. Furthermore, the attention weight is shared by all agents, resulting in a not significant overhead. When compared to prior researches, the models of STEER and MAT exhibit nearly identical sizes. Moreover, leveraging the benefits of Transformer, we are able to train all agents' strategies in parallel, leading to significantly faster training for STEER relative to STEP and HPPO. Notably, the training time for STEP and HPPO becomes prohibitively long when the number of agents becomes excessive.

F.2. Hyper-parameter Settings for Experiments

The implementations of baseline methods adhere to their official repositories, wherein default parameters are maintained during training. The detailed hyperparameter settings for STEER can be found in the Table 6- 9.

Table 6. Hyperparameter settings for Google Reasearch Football.

hyperparameter	value	hyperparameter	value	hyperparameter	value	hyperparameter	value
actor lr	7e-4	num blocks	1	batch size	4000	gamma	0.99
critic lr	5e-4	num head	1	rollout threads	10	gain	0.01
ppo epochs	15	stacked frames	1	episode length	400	training threads	16
ppo clip	0.2	hidden layer dim	64	num mini-batch	1	max grad norm	0.5
entropy coef	0.01	optim eps	1e-5	training steps	5e6	optimizer	Adam

Table 7. Hyperparameter settings for Multi-Agent MuJoCo.

hyperparameter	value	hyperparameter	value	hyperparameter	value	hyperparameter	value
actor lr	5e-5	num blocks	2	batch size	4000	gamma	0.99
critic lr	5e-5	num head	1	rollout threads	40	gain	0.01
ppo epochs	10	stacked frames	1	episode length	100	training threads	16
ppo clip	0.05	hidden layer dim	64	num mini-batch	40	max grad norm	0.5
entropy coef	1e-3	optim eps	1e-5	training steps	1e7	optimizer	Adam

Table 8. Hyperparameter settings for Highway On-Ramp Merging.

hyperparameter	value	hyperparameter	value	hyperparameter	value	hyperparameter	value
actor lr	5e-4	num blocks	1	batch size	4000	gamma	0.99
critic lr	5e-4	num head	1	rollout threads	20	gain	0.01
ppo epochs	5	stacked frames	1	episode length	200	training threads	16
ppo clip	0.05	hidden layer dim	64	num mini-batch	1	max grad norm	0.5
entropy coef	0.01	optim eps	1e-5	training steps	5e5	optimizer	Adam

Table 9. Hyperparameter settings for single-step/multi-step matrix games.

hyperparameter	value	hyperparameter	value	hyperparameter	value	hyperparameter	value
actor lr	5e-4	num blocks	1	batch size	100/1000	gamma	0.99
critic lr	5e-3	num head	1	rollout threads	4/10	gain	0.01
ppo epochs	5	stacked frames	1	episode length	25/100	training threads	8
ppo clip	0.05	hidden layer dim	64	num mini-batch	1	max grad norm	0.5
entropy coef	0.05	optim eps	1e-5	training steps	1e4/2e5	optimizer	Adam