One Step is Enough: Multi-Agent Reinforcement Learning Based on One-Step Policy Optimization for Order Dispatch on Ride-Sharing Platforms

Zijian Zhao[1], **Sen Li**[1,2] *
[1]The Hong Kong University of Science and Technology
[2]The Hong Kong University of Science and Technology (Guangzhou)

Abstract

Order dispatch is a critical task in ride-sharing systems with Autonomous Vehicles (AVs), directly influencing operational efficiency and profitability. While Multi-Agent Reinforcement Learning (MARL) offers a scalable paradigm by decomposing the massive state-action space, existing methods are heavily reliant on accurate value function estimation. In large-scale, highly stochastic urban environments, such estimation is notoriously prone to bias and instability, severely limiting training efficiency and policy quality. To overcome this limitation, we propose two novel policy optimization methods that completely bypass the need for critic networks. First, we establish a formal connection between the homogeneity of AV fleets and the short-memory mixing property of the transportation network, proving that agent value functions converge to a common time-varying baseline up to a negligible residual. Leveraging this insight, we introduce Single-Trajectory Group Relative Policy Optimization (ST-GRPO), an adaptation of Large Language Models (LLMs) post-training techniques to multi-agent trajectories, which replaces the traditional value baseline with the fleet-wide average reward-to-go. Inspired by this reduction, we further derive One-Step Policy Optimization (OSPO), demonstrating that under the established structural priors, an optimal policy can be learned using only immediate, group-normalized rewards—rendering long-horizon bootstrapping unnecessary. Experiments on real-world ride-hailing datasets from Manhattan and Queens demonstrate that both ST-GRPO and OSPO achieve promising performance, particularly in reducing pickup times and increasing order service rates. Remarkably, both methods operate efficiently using simple Multilayer Perceptron (MLP) networks with minimal GPU utilization, underscoring the practical impact of exploiting domain structure for scalable MARL. Our code, trained models, and processed data are provided at `https://github.com/RS2002/OSPO`.

1 Introduction

The widespread adoption of on-demand ride-sharing platforms has fundamentally transformed urban transportation systems, offering scalable and sustainable mobility solutions for modern cities. By enabling multiple passengers with similar itineraries to share a single vehicle, ride-sharing platforms alleviate congestion, reduce vehicle kilometers traveled, and decrease urban fleet sizes, thereby addressing pressing challenges such as traffic, emissions, and inefficient resource utilization [1; 2]. Recently, with the development of Autonomous Vehicle (AV) technology, systems like Robotaxi have been introduced by various companies, including Baidu, Didi, and Zoox. These systems not only offer lower operating costs through the AV fleet but also enable better management, as the fleet is fully controlled rather than relying on human drivers.

*Corresponding Author: Sen Li

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

However, managing such platforms presents significant operational complexity. Passengers arrive randomly over time, each with unique origins and destinations, and the platform must dynamically determine not only how to bundle new orders together, but also how to assign each bundle to a suitable vehicle. These dispatching decisions must take into account the current routing of each vehicle, the destinations of onboard passengers, and the service quality requirements of all users. Critically, all decisions must be made in real time, under uncertainty regarding future demand and supply, making the order dispatch problem a central and challenging task for ride-sharing platforms with AVs.

Recently, Reinforcement Learning (RL) methods have shown great promise in addressing the order dispatch problem on ride-sharing platforms. Although order dispatch is fundamentally a centralized task, requiring the platform to aggregate information from all vehicles and orders to make globally optimal decisions, the massive state and action spaces encountered in realistic settings (which can reach millions or billions [3]) pose a serious Curse of Dimensionality (CoD), rendering Single-Agent Reinforcement Learning (SARL) approaches intractable. As a result, most recent works adopt the Multi-Agent Reinforcement Learning (MARL) framework, where each vehicle is treated as an individual agent, and a global controller coordinates the agents to optimize the overall system objectives. MARL methods are typically categorized into three types: Decentralized Training with Decentralized Execution (DTDE), Centralized Training with Decentralized Execution (CTDE), and Centralized Training with Centralized Execution (CTCE) [4]. Many existing ride-sharing approaches build on the DTDE paradigm, since centralized methods still face scalability limitations similar to SARL, and are thus difficult to apply in large-scale scenarios with hundreds or thousands of agents. These approaches rely on the independent assumption that each agent acts based only on its local observation and treats other agents as part of the environment. This often leads to unstable training and poor cooperation among agents, as well as large estimation errors in Q-values and V-values [5], ultimately resulting in suboptimal and inefficient learning. Due to page limitation, we left the detailed literature review in Appendix A.

To address this problem, we propose two novel multi-agent policy optimization methods that systematically exploit the homogeneous nature of AV fleets and the short-memory mixing properties of large-scale ride-sharing systems (formalized in Section 2). Crucially, these structural priors allow us to circumvent the notoriously difficult task of estimating Q-values or V-values in non-stationary multi-agent environments. First, we adapt Group Relative Policy Optimization (GRPO) [6] to the order dispatch task by replacing the V-value baseline in Proximal Policy Optimization (PPO) [7] with the group-average reward-to-go. In this adaptation, we replace group rewards computed over multiple trajectories with multiple agent rewards derived from a single trajectory, leading to our method, Single-Trajectory GRPO (ST-GRPO). This modification significantly reduces computational resource requirements and minimizes bias by eliminating the estimation errors associated with neural network-based methods. Furthermore, inspired by the baseline replacement strategy, we introduce a more efficient method called One-Step Policy Optimization (OSPO), which can be trained using only one-step group rewards. By analyzing the one-step Bellman expansion under the established homogeneity and short-memory conditions, we prove that the advantage function reduces to the immediate, group-normalized reward (up to a negligible residual). Both methods are validated through case studies using real-world ride-hailing data and demonstrate promising performance with high efficiency. The main contributions of this paper are summarized as follows:

- We introduce Single-Trajectory GRPO, a novel order dispatch method for AV-based ride-sharing systems built upon GRPO. By replacing the PPO baseline with the group-average reward-to-go, our method removes the necessity of estimating V-values, thereby eliminating the estimation bias introduced by neural network critics. Moreover, by substituting group rewards computed over multiple trajectories with multi-agent rewards drawn from a single trajectory, ST-GRPO further enhances training efficiency.
- We formally derive that, under the structural assumptions of fleet homogeneity and rapid mixing, the long-term advantage function collapses to the one-step group-relative reward. This result shows that the critic is unnecessary in this class of large-scale cooperative MARL problems, reducing long-horizon policy optimization to a one-step group-relative update.
- Extensive experiments on real-world datasets demonstrate that ST-GRPO and OSPO not only surpass existing DTDE, CTDE, and CTCE baselines in terms of service rate and pickup efficiency, but also achieve these gains using only simple Multilayer Perceptron (MLP) backbones with substantially lower GPU utilization. Notably, OSPO attains the best overall performance while maintaining the lowest computational footprint among all compared methods

2 Preliminary

2.1 Problem Setup

In this paper, we study the optimal order assignment problem for a ride-sharing platform with n homogeneous autonomous vehicles (AVs), which are treated as agents, each with capacity c. At each time step t, a set of w_t new ride requests arrives from passengers at different locations. The platform must make real-time dispatch decisions by assigning each order to an appropriate vehicle. These decisions should account for the current route of each vehicle, the destinations of onboard passengers, and the service quality requirements of all users. We formulate this problem as a Multi-Agent Markov Decision Process (MAMDP) [8], defined by $\langle n, S, U, \mathrm{P}, R, \gamma \rangle$, where n denotes the number of agents and γ is the discount factor. The global state is defined as $S = (s^o, s_1, s_2, \ldots, s_n)$, where s^o contains the information of unassigned orders and is observable to all agents, while s_i denotes the state of agent i. For notational convenience, in the remainder of this paper we use s_i to represent the concatenated state $[s_i, s^o]$. In our setting, s^o includes the origins, destinations, and arrival times of all unconfirmed orders. The agent state s_i includes the current location of vehicle i, its remaining capacity, its cumulative reward since the beginning of the episode, the average cumulative reward across all agents, and the information of all onboard orders, including their destinations, estimated total travel times, and estimated remaining travel times. The global action $U = (u_1, u_2, \ldots, u_n)$ is the joint action of all agents. At time step t, each agent action u_i is represented as a w_t-dimensional binary vector, where $u_{i,j} = 1$ indicates that order j is assigned to agent i, and w_t denotes the number of candidate orders at time t. The transition function $\mathrm{P}(\cdot)$ is not modeled explicitly, as we consider a model-free reinforcement learning setting. Following [5], we define the reward function as:

$$\mathrm{r}(s_i, u_i) = \begin{cases} \eta_1 + \eta_2 p_i^{in} - \eta_3 p_i^{out} - \eta_4 \chi_i - \eta_5 \rho_i, |u_i| = 1 \\ 0, |u_i| = 0 \end{cases} \tag{1}$$

where η_1 to η_5 are non-negative weights representing the platform's valuation of each term, p_i^{in} and p_i^{out} represent the income from passengers and the vehicle operational costs, respectively. The variables χ_i and ρ_i represent the number of en-route orders that will exceed their scheduled time and the additional travel time of all en-route orders when the assigned order is added to the scheduled route of agent i, respectively (*Note that these variables depend on the state s_i and the newly assigned order indicated by u_i*). This reward function is designed to comprehensively consider the interests of the platform and passengers, mimicking the operation of a real-world ride-sharing platform. It is important to emphasize that p_i^{in} and p_i^{out} are calculated based on the order distance and the additional travel distance for the agent, respectively. When calculating travel time, we will utilize the Traveling Salesman Problem (TSP) solution to optimize the agent's route. Since order dispatch is a fully cooperative task, the global reward can be expressed as the sum of rewards from each individual agent, written as $\mathrm{R}(S, U) = \sum_{i=1}^{n} \mathrm{r}(s_i, u_i)$.

2.2 Group Relative Policy Optimization (GRPO)

With the success and popularity of Reinforcement Learning from Human Feedback (RLHF), RL-based LLM post-training has garnered increasing attention. This has also led to the emergence of many new RL methods, such as Direct Preference Optimization (DPO) [9], Reinforce Leave-One-Out (RLOO) [10], and GRPO [6]. Among these, GRPO is noted for its simple format, high efficiency, and promising performance, which are core factors contributing to the success of DeepSeekMath [6] and DeepSeek-V3 [11]. Serving as a PG [12] based method, the optimization objective of GRPO has a similar format to PPO [7], expressed as:

$$\begin{aligned} &\mathrm{J}_{GRPO}(\theta) = -\iota \mathrm{KL}(\pi_\theta \| \pi_{ref}) + \\ &\mathbb{E}_{s \sim d^{\pi_{\theta^-}}, u \sim \pi_{\theta^-}(s)} \left[\min \left\{ \frac{\pi_\theta(u|s)}{\pi_{\theta^-}(u|s)} \hat{\mathrm{A}}_{\pi_\theta}(s,u), \mathrm{CLIP}\left(\frac{\pi_\theta(u|s)}{\pi_{\theta^-}(u|s)}, 1-\epsilon, 1+\epsilon' \right) \hat{\mathrm{A}}_{\pi_\theta}(s,u) \right\} \right], \end{aligned} \tag{2}$$

where π denotes the policy, d^π denotes the state distribution under policy π, θ and θ^- represent the current network parameters and the parameters used for experience collection, respectively, s and u denote the state and action, π_{ref} is the reference policy (i.e., the initial policy before post-training), serving as a normalization term to prevent the policy from diverging too much from the reference, $\hat{\mathrm{A}}_{\pi_\theta}$ represents the advantage function of policy π_θ, and $\iota, \epsilon, \epsilon'$ are hyper-parameters We follow the improvement in [13], which suggests setting a higher value for ϵ' than for ϵ to encourage exploration.

Aside from the KL divergence term, the core difference between GRPO and PPO lies in the definition of the advantage function. In PPO, the original advantage function is defined as:

$$\mathrm{A}_\pi(s_t, u_t) = \mathrm{Q}_\pi(s_t, u_t) - \mathrm{V}_\pi(s_t) \,, \tag{3}$$

where the Q-value can be estimated using real trajectories, and the V-value can be estimated through a critic network. In contrast, GRPO proposes a simpler and more efficient advantage function given by:

$$\hat{\mathrm{A}}_\pi(s_t, u_t) = \sum_{\tau \in K_t} \gamma^\tau \frac{\mathrm{r}(s_{t+\tau}, u_{t+\tau}) - \mu}{\sigma} \,, \tag{4}$$

where γ is discount factor[2], $K_t = \{0, 1, 2, \ldots, T-t\}$, T represents the time horizon, $\mathrm{r}(\cdot, \cdot)$ denotes the reward function, and μ and σ represent the mean and standard deviation of the rewards across multiple trajectories with the same initial state and policy.

2.3 Homogeneous and Short-Memory Properties in AV Ride-Sharing Systems

We apply GRPO to ride-sharing platforms with autonomous vehicles, where the method can be further tailored by exploiting two key properties of the problem setting: (i) homogeneity and (ii) short memory.

(i) Homogeneity: All AVs in the fleet are homogeneous, sharing the same vehicle model and characteristics, such as capacity and speed. Moreover, because all agents have identical state and action spaces, the optimal policy is expected to be the same across agents under the total cooperative game [14; 15; 16]. In particular, [14] showed that, for homogeneous agents, policy sharing during training can still converge to an optimal solution. Accordingly, in this paper, all agents share a common policy for order dispatch.

(ii) Short-Memory: We consider a short-memory ride-sharing system where the current state and action of each agent have limited influence on its decisions and rewards over long time horizons. Because travel demand exhibits strong temporal variation (e.g., rush hours vs. off-peak), the environment is inherently non-stationary. We capture the short-memory property in this non-stationary regime by requiring that the system mixes much faster than the timescale at which demand characteristics change appreciably. Formally, let $\mathcal{M}_t$ denote the exogenous demand parameters at time t (origin-destination flows, etc.). We assume there exists a time window length τ_{mix} such that over any interval of duration τ_{mix}, the variation of $\mathcal{M}_t$ is negligible, and within such an interval the state process of each agent satisfies a uniform β-mixing condition. To formalize this, let $\{s_{i,t}\}_{t \geq 0}$ be the state process of agent i under policy π, and let $\mathcal{F}_{-\infty}^t$ and $\mathcal{F}_{t+k}^\infty$ be the σ-algebras generated by the process up to time t and from time $t+k$ onward, respectively. The β-mixing coefficient at lag k is defined as:

$$\beta_\pi(k) = \sup_t \ \frac{1}{2} \sup \sum_{i=1}^{I} \sum_{j=1}^{J} \big|\mathrm{P}(A_i \cap B_j) - \mathrm{P}(A_i)\mathrm{P}(B_j)\big|, \tag{5}$$

where the inner supremum is taken over all finite partitions (A_i) of $\mathcal{F}_{-\infty}^t$ and (B_j) of $\mathcal{F}_{t+k}^\infty$. The process is said to be uniformly β-mixing if $\beta_\pi(k) \to 0$ as $k \to \infty$. This property aligns with real ride-sharing systems: regardless of an AV's current location or en-route orders, after a period ranging from minutes to hours, the system's stochastic order dynamics will drive the AV into an arbitrary random status, independent of its initial condition.

In our non-stationary setting, the supremum over t guarantees that the rate of memory decay is bounded independently of the specific time of day. Consequently, for any $\epsilon > 0$, there exists a finite lag $\kappa(\epsilon)$ such that $\beta_\pi(\kappa) \leq \epsilon$, uniformly in t.

For Markovian state processes (which holds under policy π), a standard consequence of uniform β-mixing is that the total variation distance between the κ-step state distributions starting from any two initial states s, s' is bounded by ϵ, irrespective of the starting time:

$$\forall \epsilon > 0, \ \exists \kappa > 0, \ \sup_t \sup_{s,s'} \big\| \mathrm{P}_\pi(s_{t+\kappa} \mid s_t = s) - \mathrm{P}_\pi(s_{t+\kappa} \mid s_t = s') \big\|_{\mathrm{TV}} \leq \epsilon. \tag{6}$$

This property relies on two operational conditions: (a) The operational region is sufficiently open, allowing orders to originate from and be destined to arbitrary locations without being constrained by

[2] γ is set as 1 in GRPO paper [6].

region-specific isolation. (b) The policy ensures that no vehicle is restricted to a particular operational region. Prior work [17; 18] demonstrates that a well-designed policy naturally satisfies this condition, with typical mixing times on the order of minutes, which is much shorter than the timescale over which demand profiles shift (tens of minutes to hours). Hence the uniform β-mixing assumption is empirically justified.

2.4 Value Homogeneity Property in AV Ride-Sharing Systems

Leveraging the homogeneity and short-memory properties, we identify a value homogeneity property among agents, formally defined as follows: there exists a time-dependent fleet-wide mean V-value $\bar{V}_t$ and small residual terms $\omega(\cdot)$ such that for every agent $i \in \mathcal{I} = \{1, \ldots, n\}$,

$$\mathrm{V}_\pi(s_{i,t}) = \bar{V}_t - \omega(s_{i,t}), \qquad \text{where } \omega(s_{i,t}) \ll \bar{V}_t. \tag{7}$$

The residual is defined as the deviation of the agent's value from the baseline $\bar{V}_t$:

$$\omega(s_{i,t}) := \bar{V}_t - \mathrm{V}_\pi(s_{i,t}), \quad \text{with} \quad \bar{V}_t := \mathbb{E}_{i\in\mathcal{I},\, s_{i,t}\sim d_t^\pi(i)}\big[\mathrm{V}_\pi(s_{i,t})\big], \tag{8}$$

where $d_t^\pi(i)$ denotes the state distribution of agent i under policy π at time step t. Note that ω and $\bar{V}$ implicitly depend on π, but we omit this dependence for notational simplicity. A detailed proof is provided in Appendix B.1, and the conditions under which our methods satisfy these properties are discussed in Appendix E.4.

3 Methodology

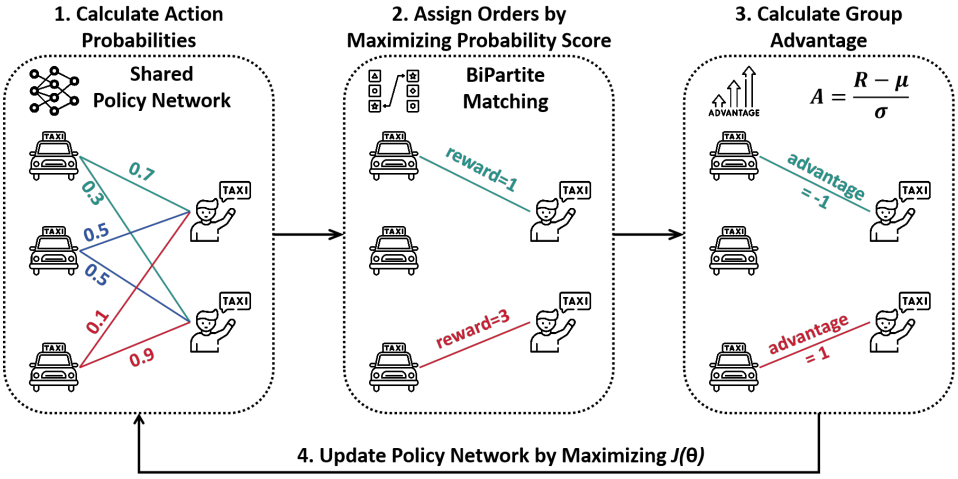

Figure 1: The training process consists of four steps: (1) Calculate the matching probability for each vehicle-order pair; (2) Assign orders by maximizing the probability score; (3) Calculate the group advantage; (4) Train the network using policy gradients.

3.1 Policy-Based Method for Order Dispatch in Ride-Sharing System

In policy-based deep RL, at each time step t, the agent directly generates the probabilities of each action based on the observation:

$$p_{i,t} = \pi_\theta(s_{i,t}) \ , \tag{9}$$

where $s_{i,t}$ represent the state of agent i at time t, and $p_{i,t} \in [0,1]^{w_t}$ represents the action probability (i.e., the probability of matching each order to agent i). Due to the homogeneity of the agents, we utilize a shared network among all agents, similar to approaches taken in previous works.

However, in the order dispatch task, we cannot assign the same order to multiple agents, making it infeasible to directly sample actions as $u_{i,t} \sim p_{i,t}$. To address this problem, we follow previous

policy-based ride-sharing methods [19; 20] by viewing the probabilities as scores and using bipartite matching to maximize the assignment score:

$$\max_{U_t} \sum_{i\in\mathcal{I}} u_{i,j,t} \cdot y_{i,j,t}, \tag{10a}$$

$$\text{s.t.} \quad \sum_{i\in\mathcal{I}} u_{i,j,t} \le 1, \quad \forall j \in \mathcal{J}_t, \tag{10b}$$

$$\sum_{j\in\mathcal{J}_t} u_{i,j,t} \le 1, \quad \forall i \in \mathcal{I}, \tag{10c}$$

$$u_{i,j,t} \in \{0,1\}, \quad \forall i \in \mathcal{I}, j \in \mathcal{J}_t, \tag{10d}$$

where $u_{i,j,t}$ indicates whether agent i is assigned to order j at time t (with 1 indicating assignment and 0 indicating no assignment), and $y_{i,j,t}$ denotes the score $p_{i,j,t}$ of agent i choosing order j at time t (with $y_{i,j,t} = -\infty$ for all unavailable workers at time t). The set $\mathcal{I} = \{1, 2, \ldots, n\}$ represents the agents, and the set $\mathcal{J}_t = \{1, 2, \ldots, w_t\}$ represents the orders to be assigned. Constraint (10b) ensures that an order can be assigned to at most one agent, while constraint (10c) guarantees that each agent is assigned at most one order. This process is illustrated in the first two steps of Fig. 1.

3.2 Single-Trajectory GRPO (ST-GRPO) Solution

In the original GRPO [6], the advantage function is defined as in Eq. (4), which eliminates the need for a critic network and improves training efficiency. Although GRPO has proven effective in LLM fine-tuning across numerous studies, adapting it directly to large-scale ride-sharing systems presents two notable limitations. (i) First, the advantage formulation in Eq. (4) lacks a clear theoretical connection to the standard advantage definition in Eq. (3), making its interpretation less transparent. (ii) Second, GRPO relies on repeatedly sampling from the same initial state (i.e., the same prompt in NLP). In a ride-sharing simulator with hundreds of thousands of AV agents, such repeated sampling is computationally prohibitive. Moreover, in practical deployments, resetting the entire system to an identical initial state of all agents and orders is infeasible. To overcome these limitations, we propose an adapted GRPO method that operates on a single trajectory collected from all agents. Our derivation starts from the standard advantage definition (Eq. (3)) and naturally leads to a form reminiscent of the original GRPO, as detailed below.

Under policy π, we collect a single system trajectory $\mathcal{T}^\pi = \{S_0, U_0, R_0, S_1, U_1, R_1, \ldots\}$ involving all agents. For agent i, let $\mathcal{T}^\pi_{i,t}$ denote the sub-trajectory from time step t onward, and define its reward-to-go as

$$G(\mathcal{T}^\pi_{i,t}) = \sum_{\tau=t}^{T} \gamma^{\tau-t} r_{i,\tau}, \tag{11}$$

where $G(\mathcal{T}^\pi_{i,t})$ can serve as a Monte Carlo estimate of the state-action value $Q_\pi(s_{i,t}, u_{i,t})$ (i.e. $\hat{Q}_{i,t} = G(\mathcal{T}^\pi_{i,t})$).

The V-value homogeneity property (Eq. (7)) states that all agents share a common time-dependent baseline $\bar{V}_t$ up to a small residual $\omega_{i,t}$. Expanding $V_\pi(s_{i,t})$ as an expectation over trajectories and then taking the population mean yields

$$\begin{aligned} \mathrm{V}_\pi(s_{i,t}) &= \mathbb{E}_{i\in\mathcal{I}}\big[\mathrm{V}_\pi(s_{i,t})\big] - \omega(s_{i,t}) = \mathbb{E}_{i\in\mathcal{I}}\Big[\mathbb{E}_{\mathcal{T}^\pi_{i,t}\sim\rho^\pi_t(i)}\big[G_{i,t}\big]\Big] - \omega(s_{i,t}) \\ &= \mathbb{E}_{i\in\mathcal{I},\, \mathcal{T}^\pi_{i,t}\sim\rho^\pi_t(i)}\big[G_{i,t}\big] - \omega(s_{i,t}) \approx \mathbb{E}_{i\in\mathcal{I},\, \mathcal{T}^\pi_{i,t}\sim\rho^\pi_t(i)}\big[G_{i,t}\big], \end{aligned}$$

where $\rho^\pi_t(i)$ represents the sub-trajectory distribution of agent i from step t under policy π.

Eq. (12) reveals that the desired value $\mathrm{V}_\pi(s_{i,t})$ equals the expectation of the reward-to-go over both the fleet distribution and the trajectory randomness, minus a small bias. Since we only have access to a single system trajectory, we approximate this double expectation by its sample mean across the fleet:

$$\hat{V}_t = \frac{1}{|\mathcal{I}|} \sum_{i\in\mathcal{I}} G_{i,t}. \tag{12}$$

This substitution is justified by viewing $\hat{V}_t$ as a Monte Carlo estimator of $\mathbb{E}_{i\in\mathcal{I},\, \mathcal{T}^\pi_{i,t}\sim\rho^\pi_t(i)}[G_{i,t}]$ with $|\mathcal{I}|$ samples, where each agent contributes one draw from its own trajectory distribution. In a large

fleet with fast mixing dynamics, the returns $G_{i,t}$ exhibit limited mutual dependence, and their sample mean $\hat{V}_t$ concentrates sharply around its expectation $\bar{V}_t$. Under such a circumstance, we can also get the estimation of V-value of each agent $\hat{V}_{i,t} \approx \hat{V}_t$.

Consequently we can write an estimate of the advantage function as:

$$\begin{aligned} \hat{\mathrm{A}}_\pi^{GRPO}(s_{i,t}, u_{i,t}) &= \frac{1}{\sigma_{t:}} \left(\hat{Q}_{i,t} - \hat{V}_{i,t} \right) \approx \frac{1}{\sigma_{t:}} \left(G_{i,t} - \mu_{t:} \right) , \\ \mu_{t:} = \hat{V}_t, \quad \sigma_{t:} &= \sqrt{\mathbb{E}_{i \in \mathcal{I}} \left[(G_{i,t} - \mu_{t:})^2 \right]} . \end{aligned} \tag{13}$$

where $\sigma_{t:}$ is utilized for normalization to improve training efficiency, utilized by many works [6; 21].

Additionally, when using the target function in Eq. (2) within the MARL scenario, we note that there is no reference policy π_{ref} as found in LLMs. We innovatively propose regularly tracking policy performance and retaining the best checkpoint. This best policy will serve as the reference policy, helping to ensure that the current policy does not deviate significantly from the optimal one, aided by the KL normalization term. We acknowledge this heuristic may affect theoretical convergence guarantees, but we found it effective in practice and it does not harm empirical performance.

3.3 One Step Policy Optimization (OSPO) Solution

Based on the homogeneous property in Eq. (7), we can recalculate the advantage function as follows:

$$\begin{aligned} &\mathrm{A}_\pi^{OSPO}(s_{i,t}, u_{i,t}) = \mathrm{Q}_\pi(s_{i,t}, u_{i,t}) - \mathrm{V}_\pi(s_{i,t}) \\ &= \underbrace{\mathrm{r}(s_{i,t}, u_{i,t}) - \mathbb{E}_{i \in \mathcal{I}, s_{i,t} \sim d_t^\pi(i), u_{i,t} \sim \pi(s_{i,t})} [\mathrm{r}(s_{i,t}, u_{i,t}))]}_{\text{Reward Term}} \underbrace{- \gamma \mathbb{E}_{s_{i,t+1} \sim \mathrm{P}(\cdot | s_{i,t}, u_{i,t})} [\omega(s_{i,t+1})] + \omega(s_{i,t})}_{\text{Residual Term}}, \end{aligned} \tag{14}$$

In Appendix B.2, we prove that when the $\overline{V}_t$ has an upper bound, which holds when the horizon is limited by T and the mixing time is far less than T, one can select an appropriate discount factor γ and mixing lag κ to make the residual term arbitrarily small and thus negligible.

Then, if we perform a similar operation to GRPO, using the average reward to replace the expectation, applying reward normalization [21], and ignoring the residual term, we arrive at the advantage function:

$$\begin{aligned} \hat{\mathrm{A}}_\pi^{OSPO}(s_{i,t}, u_{i,t}) &= \frac{\mathrm{r}(s_{i,t}, u_{i,t}) - \mu_t}{\sigma_t}, \\ \mu_t = \frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} \mathrm{r}(s_{i,t}, u_{i,t}), \quad \sigma_t &= \sqrt{\mathbb{E}_{i \in \mathcal{I}} \left[(\mathrm{r}(s_{i,t}, u_{i,t}) - \mu_t)^2 \right]} . \end{aligned} \tag{15}$$

3.4 A Heuristic Regulation Term

In this paper, our proposed ST-GRPO and OSPO are both based on the approximate V-value homogeneity property in Eq. (7). However, this property may not always hold during the training exploration phase, which could lead to incorrect update directions. To address this issue, we propose adding a heuristic regulation term to better align with this property. Intuitively, the most straightforward approach is to collect the full trajectory, compute the deviation of reward-to-go among agents, and use it as a penalty term for the reward or advantage function. However, collecting the full trajectory would break the efficiency of OSPO, which relies on only one-step samples. Therefore, we propose to replace it with the change in historical cumulative reward deviation. (Note that if agents have similar historical cumulative rewards at any time, they also have similar V-values at any time.) Consequently, the final advantage function is expressed as:

$$\begin{aligned} \hat{\mathrm{A}}_\pi^{'}(si, t, u_{i,t}) &= \hat{\mathrm{A}}_\pi(s_{i,t}, u_{i,t}) - \alpha(\zeta_t - \zeta_{t-1}), \\ \zeta_t &= \mathrm{Std} \left(\{ \sum_{\tau=0}^{t} \mathrm{r}(s_{i,\tau}, u_{i,\tau}) \mid i \in \mathcal{I} \} \right), \end{aligned} \tag{16}$$

where α is a hyper-parameter and $\hat{\mathrm{A}}$ is the advantage function of OSPO and ST-GRPO. A detailed algorithm description is provided in Appendix C. Note that when Eq. (7) holds, the regulation term is close to zero, thus having limited impact on the optimal policy.

4 Experiment

4.1 Experiment Setup

To validate our method, we conduct a series of experiments using a real-world ride-hailing dataset from Manhattan, New York City [22]. The data collected from 19:00 to 19:30 on July 17, 2024, is used as the training set, which includes 3,726 valid orders. We then evaluate the trained model using data from the entire day of July 18, 2024, selecting the first 30 minutes of data from each hour, resulting in a total of 24 independent tests. An expanded generalization experiment is provided in Appendix D.6, using the data from Queens, New York.

For the simulation, we set the total number of vehicles to 1,000, with each car having a capacity of 3 and a default speed of 60 km/h. The optimal routing is achieved using the OSRM simulator [23]. The experiments are conducted using the PyTorch framework [24] on a workstation running Windows 11, equipped with an Intel(R) Core(TM) i7-14700KF processor and an NVIDIA RTX 4080 graphics card. Following the setup of previous works, we set each episode length to 30 minutes, with each step representing 1 minute, resulting in a running time of approximately 40 to 120 seconds for different methods per episode. We maximize the training episodes at 1,000 and evaluate model performance every 10 episodes.

To further illustrate the efficiency of our method, we choose a simple four-layer MLP as the policy network, with 128 units in each hidden layer and LeakyReLU [25] as the activation function. Considering the variable action space (since the number of orders changes each time), we set the network input as a single vehicle-order pair, ignoring the relationships between orders. Finally, a softmax layer is used to normalize the output probabilities. For exploration, we add gradually decreasing random noise, employing the same Binary Symmetric Channel (BSC) noise as in [5; 20; 19; 26].

In the following, we first compare our ST-GRPO and OSPO ride-sharing methods with previously popular methods of different types, including DTDE, CTDE, and CTCE. In Appendix D.5, we conduct a series of ablation studies to detail the effects of different modules in ST-GRPO and OSPO, comparing them with similar policy gradient-based methods, including Independent PPO (IPPO) [27], Multi-Agent PPO (MAPPO) [28], and Independent PG (IPG) [12]. All comparisons are conducted under identical conditions to ensure fairness. For optimization, we use the Adam optimizer [29] with an initial learning rate of 10^{-4}, a decay rate of 0.99, and a batch size of 256. For MDP, we set the discount factor γ as 0.99.

Finally, we report the hyper-parameters used in our method. The weight for the KL term, denoted ι, is set to 0.5, and the weight for the heuristic term, denoted α, is set to 1. Following [13], the CLIP parameters ϵ and ϵ' are set to 0.2 and 0.28, respectively.

Table 1: Comparison of Different Ride Sharing Methods: **Bold** entries represent the best results.

Method	OSPO	ST-GRPO	DeepPool	BMG-Q	HIVES	Enders et al.	CEVD	Hoppe et al.
Reward Type	Local	Local	Local	Local	Global	Local	Global	Mixed
RL Algorithm	OSPO	ST-GRPO	IDDQN	IDDQN	QMIX	MASAC	VD	COMA
Network Backbone	MLP	MLP	MLP	GAT	GRU	MLP	MLP	MLP
Model Size	**20K**	**20K**	**20K**	117K	16M	118K	23K	167K
GPU Occupation (GB)	**3.82**	5.17	3.97	4.28	6.01	8.19	21.45	9.78
Time per Episode (s)	30.22	33.75	35.36	62.76	69.17	91.82	80.44	192.70
Reward ($\times 10^3$)	**11.00 ± 2.63**	10.90 ± 2.61	10.41 ± 2.34	10.88 ± 2.35	10.70 ± 2.32	9.77 ± 2.41	10.71 ± 2.36	9.93 ± 2.43
Service Rate	**1.00 ± 0.00**	**1.00 ± 0.00**	0.99 ± 0.02	1.00 ± 0.01	0.99 ± 0.02	0.99 ± 0.02	1.00 ± 0.01	0.99 ± 0.02
Delivery Time	12.32 ± 0.71	12.40 ± 0.73	12.20 ± 1.43	**10.61 ± 0.70**	10.81 ± 0.51	12.65 ± 0.52	11.95 ± 0.79	12.54 ± 0.61
Detour Time	1.26 ± 0.41	1.42 ± 0.52	1.24 ± 0.90	**0.59 ± 0.59**	0.84 ± 0.79	2.72 ± 0.33	1.12 ± 0.85	2.54 ± 0.43
Pickup Time	**6.71 ± 0.64**	7.33 ± 0.60	8.67 ± 1.23	7.62 ± 0.23	8.13 ± 0.31	8.05 ± 0.18	8.02 ± 0.88	8.10 ± 0.58
Confirmation Time	**0.00 ± 0.00**	**0.00 ± 0.00**	0.01 ± 0.03	0.00 ± 0.02	0.00 ± 0.01	0.06 ± 0.20	0.00 ± 0.01	0.02 ± 0.08

4.2 Experiment Results

The training process and testing results are presented in Fig. 2 and Table 1, respectively. We illustrate the cumulative reward alongside the number of orders served, as well as the average delivery time, detour time, pickup time, and confirmation time for each order. Detailed explanations of these metrics can be found in Appendix D.2. According to the results, we observe that ST-GRPO and OSPO outperform the others primarily by achieving lower pickup and confirmation times, which

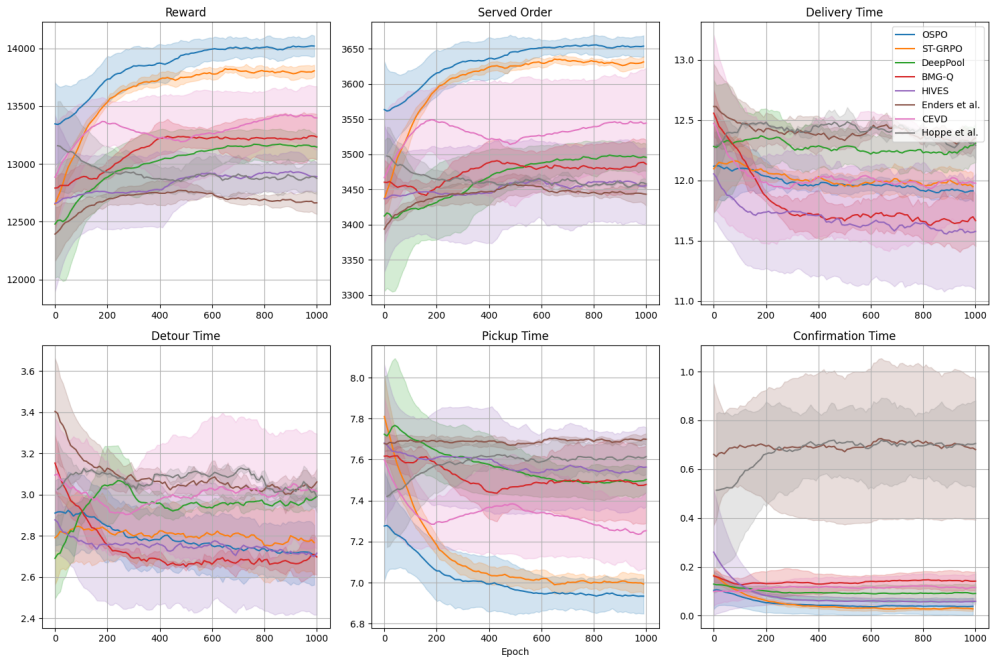

Figure 2: Training Process: Each method is trained five times. The shaded area represents the standard deviation of each method.

allows them to serve more orders. Notably, OSPO, with the smallest network size and lowest GPU utilization, achieves the best performance across most testing scenarios. Please note that the results in Table 1 represent the weighted average values from 24 different episodes, calculated according to order volume, along with the standard deviation, which thus reflects variation across the hourly test windows; the deviations in the remaining tables reflect variation across training seeds on a single time window, as detailed in Appendix D. More details, including a comprehensive ablation study and additional experiments in another scenario, can be found in Appendix D.

5 Conclusion

In this paper, we introduced ST-GRPO and OSPO, two novel MARL algorithms tailored for the large-scale order dispatch problem in AV ride-sharing platforms. By exploiting the inherent homogeneity and short-memory mixing properties of AV fleets, we formally demonstrated that the value functions across agents converge to a common time-varying baseline, enabling a radical simplification of the advantage estimation process. ST-GRPO successfully adapts the group-relative paradigm from language model fine-tuning to the multi-agent trajectory setting, while OSPO pushes this reduction to its logical extreme, demonstrating that one step is indeed enough: competitive policies can be learned using only immediate, group-normalized rewards without any critic network or long-horizon backpropagation. Our extensive experiments on real-world Manhattan and Queens datasets confirm that both methods not only match but often surpass the performance of competitive MARL baselines, achieving lower pickup times and higher service rates, all while utilizing simple MLP backbones with low GPU memory. More discussion could be found at Appendix E.

Acknowledgments

This research was supported by General Research Fund of Hong Kong Research Grants Council under project 16202922.

References

[1] H. Wang and H. Yang, "Ridesourcing systems: A framework and review," *Transportation Research Part B: Methodological*, vol. 129, pp. 122–155, 2019.

[2] P. Santi, G. Resta, M. Szell, S. Sobolevsky, S. H. Strogatz, and C. Ratti, "Quantifying the benefits of vehicle pooling with shareability networks," *Proceedings of the National Academy of Sciences*, vol. 111, no. 37, pp. 13290–13294, 2014.

[3] A. Sivagnanam, A. Pettet, H. Lee, A. Mukhopadhyay, A. Dubey, and A. Laszka, "Multi-agent reinforcement learning with hierarchical coordination for emergency responder stationing," *arXiv preprint arXiv:2405.13205*, 2024.

[4] W. Jin, H. Du, B. Zhao, X. Tian, B. Shi, and G. Yang, "A comprehensive survey on multi-agent cooperative decision-making: Scenarios, approaches, challenges and perspectives," *arXiv preprint arXiv:2503.13415*, 2025.

[5] Y. Hu, S. Feng, and S. Li, "Bmg-q: Localized bipartite match graph attention q-learning for ride-pooling order dispatch," *arXiv preprint arXiv:2501.13448*, 2025.

[6] Z. Shao, P. Wang, Q. Zhu, R. Xu, J. Song, X. Bi, H. Zhang, M. Zhang, Y. Li, Y. Wu, *et al.*, "Deepseekmath: Pushing the limits of mathematical reasoning in open language models," *arXiv preprint arXiv:2402.03300*, 2024.

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

[8] M. L. Littman, "Markov games as a framework for multi-agent reinforcement learning," in *Machine learning proceedings 1994*, pp. 157–163, Elsevier, 1994.

[9] R. Rafailov, A. Sharma, E. Mitchell, C. D. Manning, S. Ermon, and C. Finn, "Direct preference optimization: Your language model is secretly a reward model," *Advances in Neural Information Processing Systems*, vol. 36, pp. 53728–53741, 2023.

[10] A. Ahmadian, C. Cremer, M. Gallé, M. Fadaee, J. Kreutzer, O. Pietquin, A. Üstün, and S. Hooker, "Back to basics: Revisiting reinforce style optimization for learning from human feedback in llms," *arXiv preprint arXiv:2402.14740*, 2024.

[11] A. Liu, B. Feng, B. Xue, B. Wang, B. Wu, C. Lu, C. Zhao, C. Deng, C. Zhang, C. Ruan, *et al.*, "Deepseek-v3 technical report," *arXiv preprint arXiv:2412.19437*, 2024.

[12] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," *Advances in neural information processing systems*, vol. 12, 1999.

[13] Q. Yu, Z. Zhang, R. Zhu, Y. Yuan, X. Zuo, Y. Yue, T. Fan, G. Liu, L. Liu, X. Liu, *et al.*, "Dapo: An open-source llm reinforcement learning system at scale," *arXiv preprint arXiv:2503.14476*, 2025.

[14] N. Grammel, S. Son, B. Black, and A. Agrawal, "Revisiting parameter sharing in multi-agent deep reinforcement learning," *arXiv preprint arXiv:2005.13625*, 2020.

[15] S. Sanjari, N. Saldi, and S. Yüksel, "Optimality of independently randomized symmetric policies for exchangeable stochastic teams with infinitely many decision makers," *Mathematics of Operations Research*, vol. 48, no. 3, pp. 1254–1285, 2023.

[16] S. Sanjari, N. Saldi, and S. Yüksel, "Optimality of decentralized symmetric policies for stochastic teams with mean-field information sharing," in *2024 IEEE 63rd Conference on Decision and Control (CDC)*, pp. 3317–3322, IEEE, 2024.

[17] J. Sun, H. Jin, Z. Yang, L. Su, and X. Wang, "Optimizing long-term efficiency and fairness in ride-hailing via joint order dispatching and driver repositioning," in *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, pp. 3950–3960, 2022.

[18] J. Sun, H. Jin, Z. Yang, and L. Su, "Optimizing long-term efficiency and fairness in ride-hailing under budget constraint via joint order dispatching and driver repositioning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 36, no. 7, pp. 3348–3362, 2024.

[19] T. Enders, J. Harrison, M. Pavone, and M. Schiffer, "Hybrid multi-agent deep reinforcement learning for autonomous mobility on demand systems," in *Learning for Dynamics and Control Conference*, pp. 1284–1296, PMLR, 2023.

[20] H. Hoppe, T. Enders, Q. Cappart, and M. Schiffer, "Global rewards in multi-agent deep reinforcement learning for autonomous mobility on demand systems," in *6th Annual Learning for Dynamics & Control Conference*, pp. 260–272, PMLR, 2024.

[21] V. Kurin, A. De Palma, I. Kostrikov, S. Whiteson, and P. K. Mudigonda, "In defense of the unitary scalarization for deep multi-task learning," *Advances in Neural Information Processing Systems*, vol. 35, pp. 12169–12183, 2022.

[22] N. Y. C. Taxi and L. Commission, "Nyc taxi and limousine commission-trip record data nyc.," 2024.

[23] D. Luxen and C. Vetter, "Real-time routing with openstreetmap data," in *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, GIS '11, (New York, NY, USA), pp. 513–516, ACM, 2011.

[24] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[25] A. L. Maas, A. Y. Hannun, A. Y. Ng, *et al.*, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. icml*, vol. 30, p. 3, Atlanta, GA, 2013.

[26] A. O. Al-Abbasi, A. Ghosh, and V. Aggarwal, "Deeppool: Distributed model-free algorithm for ride-sharing using deep reinforcement learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 12, pp. 4714–4727, 2019.

[27] C. S. De Witt, T. Gupta, D. Makoviichuk, V. Makoviychuk, P. H. Torr, M. Sun, and S. Whiteson, "Is independent learning all you need in the starcraft multi-agent challenge?," *arXiv preprint arXiv:2011.09533*, 2020.

[28] C. Yu, A. Velu, E. Vinitsky, J. Gao, Y. Wang, A. Bayen, and Y. Wu, "The surprising effectiveness of ppo in cooperative multi-agent games," *Advances in neural information processing systems*, vol. 35, pp. 24611–24624, 2022.

[29] D. P. Kingma, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[30] M. Li, Z. Qin, Y. Jiao, Y. Yang, J. Wang, C. Wang, G. Wu, and J. Ye, "Efficient ridesharing order dispatching with mean field multi-agent reinforcement learning," in *The world wide web conference*, pp. 983–994, 2019.

[31] O. De Lima, H. Shah, T.-S. Chu, and B. Fogelson, "Efficient ridesharing dispatch using multi-agent reinforcement learning," *arXiv preprint arXiv:2006.10897*, 2020.

[32] J. Hao and P. Varakantham, "Hierarchical value decomposition for effective on-demand ride-pooling," in *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*, pp. 580–587, 2022.

[33] T. Rashid, M. Samvelyan, C. S. De Witt, G. Farquhar, J. Foerster, and S. Whiteson, "Monotonic value function factorisation for deep multi-agent reinforcement learning," *Journal of Machine Learning Research*, vol. 21, no. 178, pp. 1–51, 2020.

[34] A. Bose, H. Jiang, P. Varakantham, and Z. Ge, "On sustainable ride pooling through conditional expected value decomposition," in *ECAI 2023*, pp. 295–302, IOS Press, 2023.

[35] P. Sunehag, G. Lever, A. Gruslys, W. M. Czarnecki, V. Zambaldi, M. Jaderberg, M. Lanctot, N. Sonnerat, J. Z. Leibo, K. Tuyls, *et al.*, "Value-decomposition networks for cooperative multi-agent learning based on team reward," in *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pp. 2085–2087, 2018.

[36] Z. Zhang, L. Yang, J. Yao, C. Ma, and J. Wang, "Joint optimization of pricing, dispatching and repositioning in ride-hailing with multiple models interplayed reinforcement learning," *IEEE Transactions on Knowledge and Data Engineering*, 2024.

[37] H. Jiang, Y. Xu, and P. Varakantham, "Optimizing ride-pooling operations with extended pickup and drop-off flexibility," *arXiv preprint arXiv:2503.08472*, 2025.

[38] Y. Hu, T. Dong, and S. Li, "Coordinating ride-pooling with public transit using reward-guided conservative q-learning: An offline training and online fine-tuning reinforcement learning framework," *arXiv preprint arXiv:2501.14199*, 2025.

[39] M. Tan, "Multi-agent reinforcement learning: Independent vs. cooperative agents," in *Proceedings of the tenth International Conference on Machine Learning*, pp. 330–337, 1993.

[40] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double q-learning," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 30, 2016.

[41] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, Y. Bengio, *et al.*, "Graph attention networks," *stat*, vol. 1050, no. 20, pp. 10–48550, 2017.

[42] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *International conference on machine learning*, pp. 1861–1870, Pmlr, 2018.

[43] J. Foerster, G. Farquhar, T. Afouras, N. Nardelli, and S. Whiteson, "Counterfactual multi-agent policy gradients," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, 2018.

Appendix Contents

A Related Work **14**

B Proof Derivation **14**

B.1 Approximate Value Homogeneity 14

B.2 One-Step Advantage Function 16

C Algorithm **16**

D Experiment Details **16**

D.1 Simulator Pipeline and Reward Design 16

D.2 Evaluation Metrics 17

D.3 Introduction of Comparative Methods 17

D.4 Detailed Experiment Results 18

D.5 Ablation Study 18

D.6 Expanded Experiment 20

E Discussions **22**

E.1 On the Value Homogeneity Property 22

E.2 Probabilistic Assignment and the Matching Layer 22

E.3 Bias and Variance Analysis 23

E.4 Application Conditions of the Proposed Methods 23

E.5 Behavior When the Conditions Are Not Met 24

E.6 Comparison between GRPO, ST-GRPO, and OSPO 25

E.7 Limitations and Future Work 25

E.8 Broader Impact 25

A Related Work

In this section, we provide a brief overview of current ride-sharing methods. Starting from the Independent Double Deep Q-Learning (IDDQN)-based DeepPool [26], DTDE methods have become a mainstream paradigm, as the large number of agents, along with the extensive state and action spaces, severely hinders the development of centralized methods. Subsequent works have primarily aimed to enhance algorithms by fostering better cooperation among agents. For example, Graph Neural Networks (GNNs) have been widely explored to effectively utilize neighboring information [5]. [30] also propose using mean field approaches to capture more useful global information.

Additionally, some works attempt to transfer the successes of CTDE and CTCE to ride-sharing, where centralized critics can facilitate better cooperation among agents. However, these approaches face various shortcomings and challenges. [31] first introduce QMIX to ride-sharing, but only consider tasks involving a limited number of vehicles in simulated grid scenarios, which presents a significant gap to real-world applications. [32] explore hierarchical structures to address the curse of dimensionality in QMIX [33], but the input dimensions of the mixture network still increase with the number of agents. [34] propose a Value Decomposition (VD) [35] based method; however, the credit assignment challenge becomes more complex in ride-sharing scenarios with hundreds of agents.

Lastly, there exists a series of works aimed at improving system efficiency by jointly considering relocation [36], order bundling [37], and multi-modal transportation [38]. Since these topics extend beyond the scope of this paper, we do not discuss them further here.

Nevertheless, most methods rely on precise value-function estimation (e.g., Q/V-networks), which introduces bias and instability due to approximation errors [5]. Our work addresses this gap by proposing ST-GRPO/OSPO—eliminating value estimation entirely through group rewards—for stable and scalable coordination in homogeneous MARL systems.

B Proof Derivation

B.1 Approximate Value Homogeneity

Under the shared policy π and the uniform short-memory property, we now show that the value functions of all agents can be made arbitrarily close in a relative sense, provided the mixing is sufficiently fast and the discount factor is chosen appropriately. This result justifies the use of a common time-varying baseline in our subsequent decentralized coordination.

Because the environment is non-stationary, the value function of an agent starting from state s at time t is defined as

$$\mathrm{V}_\pi(s;t) = \mathbb{E}_\pi\Big[\sum_{\tau=0}^{\infty} \gamma^\tau r(s_{t+\tau}, u_{t+\tau}) \mid s_t = s\Big], \tag{17}$$

where $\mathrm{r}(s,u) \in [0, R_{\max}]$ is the bounded one-step reward (The agent can always choose to take no order and receive zero reward, so non-negativity is natural. This can be also achieved in practice by adding a constant shift.). The discount factor $\gamma \in (0,1)$ is fixed. Although $\mathrm{V}_\pi(\cdot;t)$ depends on time, the uniform mixing condition (6) ensures that the difference between values of any two states can be uniformly controlled.

Fix an arbitrary $\epsilon > 0$. By the uniform short-memory property, choose κ such that the total variation bound (6) holds for all t. For any two states s, s' and any fixed time t, we decompose the value functions using the κ-step Bellman expansion:

$$\mathrm{V}_\pi(s;t) = \mathbb{E}_\pi\Big[\sum_{\tau=0}^{\kappa-1} \gamma^\tau r_{t+\tau} \mid s_t = s\Big] + \gamma^\kappa \mathbb{E}_{s_{t+\kappa}\sim \mathrm{P}_\pi^\kappa(\cdot|s,t)}\big[V_\pi(s_{t+\kappa}; t+\kappa)\big], \tag{18}$$

where $r_{t+\tau} = \mathrm{r}(s_{t+\tau}, u_{t+\tau})$ and $\mathrm{P}_\pi^\kappa(\cdot \mid s, t)$ denotes the κ-step state distribution starting from s at time t. Their difference satisfies:

$$\big|\mathrm{V}_\pi(s;t) - \mathrm{V}_\pi(s';t)\big| \le \underbrace{\Big|\mathbb{E}_\pi\big[\sum_{\tau=0}^{\kappa-1}\gamma^\tau r_{t+\tau} \mid s_t = s\big] - \mathbb{E}_\pi\big[\sum_{\tau=0}^{\kappa-1}\gamma^\tau r_{t+\tau} \mid s_t = s'\big]\Big|}_{\text{Term I}}$$
$$+ \gamma^\kappa \underbrace{\Big|\mathbb{E}_{\mathrm{P}_\pi^\kappa(\cdot|s,t)}[\mathrm{V}_\pi(\cdot; t+\kappa)] - \mathbb{E}_{\mathrm{P}_\pi^\kappa(\cdot|s',t)}[\mathrm{V}_\pi(\cdot; t+\kappa)]\Big|}_{\text{Term II}}. \tag{19}$$

Because $0 \le r \le R_{\max}$, Term I is bounded by $R_{\max}\frac{1-\gamma^\kappa}{1-\gamma}$. For Term II, note that $\mathrm{V}_\pi(\cdot; t+\kappa)$ is bounded by $\frac{R_{\max}}{1-\gamma}$. Using the total variation bound (6) (which holds uniformly in t),

$$\big|\mathbb{E}_{\mathrm{P}_\pi^\kappa(\cdot|s,t)}[V_\pi] - \mathbb{E}_{\mathrm{P}_\pi^\kappa(\cdot|s',t)}[V_\pi]\big| \le \|V_\pi\|_\infty \cdot \big\|\mathrm{P}_\pi^\kappa(\cdot \mid s,t) - \mathrm{P}_\pi^\kappa(\cdot \mid s',t)\big\|_{\mathrm{TV}} \le \frac{R_{\max}}{1-\gamma}\cdot\epsilon. \tag{20}$$

Thus we obtain a uniform bound:

$$\big|\mathrm{V}_\pi(s;t) - \mathrm{V}_\pi(s';t)\big| \le R_{\max}\frac{1-\gamma^\kappa}{1-\gamma} + \gamma^\kappa\frac{R_{\max}}{1-\gamma}\epsilon =: \Delta(\gamma,\kappa,\epsilon). \tag{21}$$

Lemma 1 (Time-Varying Relative Value Homogeneity)**.** *Assume the uniform short-memory property ($\beta_\pi(k) \to 0$ as $k \to \infty$ uniformly in t) and that from every state s at any time t, the agent has a probability $p_{\min} > 0$ of obtaining a strictly positive expected one-step reward (i.e., there exists $r_{\min} > 0$ such that $\mathbb{E}[r_t \mid s_t = s] \ge p_{\min} r_{\min}$ for all s, t). Define the fleet-average value at time t as $\bar{V}_t = \frac{1}{N}\sum_{i=1}^N V_\pi(s_{i,t}; t)$. Then for any desired relative tolerance $\delta > 0$, there exist a discount factor $\gamma \in (0,1)$ and an integer κ such that for all agents i and all times t,*

$$\frac{|V_\pi(s_{i,t};t) - \bar{V}_t|}{\bar{V}_t} \le \delta. \tag{22}$$

Proof. From (21), the maximum deviation between any two state-value functions at the same time t is at most $\Delta(\gamma,\kappa,\epsilon)$. A lower bound on $\bar{V}_t$ can be obtained by noting that from any state, the agent can expect at least $p_{\min} r_{\min}$ immediate reward, and thereafter the discounted sum is non-negative. Hence

$$\bar{V}_t \ge \frac{p_{\min} r_{\min}}{1-\gamma}. \tag{23}$$

Consequently,

$$\frac{|\mathrm{V}_\pi(s_{i,t};t) - \bar{V}_t|}{\bar{V}_t} \le \frac{\Delta(\gamma,\kappa,\epsilon)}{p_{\min} r_{\min}/(1-\gamma)} = \frac{R_{\max}}{p_{\min} r_{\min}}\big(1 - \gamma^\kappa + \gamma^\kappa\epsilon\big). \tag{24}$$

Choose κ so large that $\epsilon = \beta_\pi(\kappa) \le \frac{p_{\min} r_{\min}}{2R_{\max}}\delta$ and $\frac{1}{\kappa} \le \frac{p_{\min} r_{\min}}{2R_{\max}}\delta$ (possible because $\beta_\pi(\kappa) \to 0$ and $\frac{1}{\kappa} \to 0$). Set $\gamma = 1 - \frac{1}{\kappa^2}$. For $\kappa \ge 2$, we have $\gamma^\kappa \ge 1 - \frac{1}{\kappa}$, thus $1 - \gamma^\kappa \le \frac{1}{\kappa}$. Substituting gives

$$\frac{R_{\max}}{p_{\min} r_{\min}}\left(\frac{1}{\kappa} + \epsilon\right) \le \frac{R_{\max}}{p_{\min} r_{\min}}\left(\frac{p_{\min} r_{\min}}{2R_{\max}}\delta + \frac{p_{\min} r_{\min}}{2R_{\max}}\delta\right) = \delta. \tag{25}$$

Therefore, for every agent i, $|\mathrm{V}_\pi(s_{i,t};t) - \bar{V}_t| \le \delta\bar{V}_t$, completing the proof. □

The lemma establishes that, under realistic mixing conditions (uniform over time) and a mild probabilistic lower bound on obtaining positive rewards, the relative spread among agents' value functions can be made arbitrarily small by choosing the discount factor sufficiently close to 1. In practice, our system operates in exactly this regime: mixing occurs within minutes, whereas demand evolves over tens of minutes; moreover, any idle vehicle can reach a busy area with positive probability, guaranteeing $p_{\min} > 0$. Therefore, for the purpose of algorithm design and analysis, we adopt the following approximation: There exists a time-dependent common baseline $\bar{V}_t$ (shared by all agents but possibly varying slowly with t) and a small constant $\delta \ge 0$ such that for every agent i and every time step t, Eq. (22) holds. Equivalently, $\mathrm{V}_\pi(s_{i,t};t) \approx \bar{V}_t$ up to a small relative error. This property will be used to justify mean-field approximations and to simplify value estimation in our decentralized coordination scheme. In the remainder of the paper, we simplify the notation by incorporating the time index into the state variable $s_{i,t}$. Thus far, we have established the validity of Eq. (7).

B.2 One-Step Advantage Function

Here, we provide a detailed derivation of Eq. 14.

$$
\begin{aligned}
&\mathrm{A}_{\pi}^{OSPO}(s_{i,t},u_{i,t}) = \mathrm{Q}_{\pi}(s_{i,t},u_{i,t}) - \mathrm{V}_{\pi}(s_{i,t}) \\
=&\mathrm{r}(s_{i,t},u_{i,t}) + \gamma\mathbb{E}_{s_{i,t+1}\sim \mathrm{P}(\cdot|s_{i,t},u_{i,t})}[\mathrm{V}_{\pi}(s_{i,t+1})] - \mathrm{V}_{\pi}(s_{i,t}) \\
=&\mathrm{r}(s_{i,t},u_{i,t}) + \gamma\mathbb{E}_{s_{i,t+1}\sim \mathrm{P}(\cdot|s_{i,t},u_{i,t})}[\bar{V}_{t+1} - \omega(s_{i,t+1})] - \bar{V}_t + \omega(s_{i,t}) \\
=&\mathrm{r}(s_{i,t},u_{i,t}) + \gamma\bar{V}_{t+1} - \gamma\mathbb{E}_{s_{i,t+1}\sim \mathrm{P}(\cdot|s_{i,t},u_{i,t})}[\omega(s_{i,t+1})] - \mathbb{E}_{i\in\mathcal{I},s_{i,t}\sim d_t^{\pi}(i)}[\mathrm{V}_{\pi}(s_{i,t})] + \omega(s_{i,t}) \\
=&\mathrm{r}(s_{i,t},u_{i,t}) + \gamma\bar{V}_{t+1} - \gamma\mathbb{E}_{s_{i,t+1}\sim \mathrm{P}(\cdot|s_{i,t},u_{i,t})}[\omega(s_{i,t+1})] \\
&- \mathbb{E}_{i\in\mathcal{I},s_{i,t}\sim d_t^{\pi}(i),u_{i,t}\sim\pi(s_{i,t}),s_{i,t+1}\sim \mathrm{P}(\cdot|s_{i,t},u_{i,t})}[\mathrm{r}(s_{i,t},u_{i,t}) + \gamma\mathrm{V}_{\pi}(s_{i,t+1})] + \omega(s_{i,t}) \\
=&\mathrm{r}(s_{i,t},u_{i,t}) + \gamma\bar{V}_{t+1} - \gamma\mathbb{E}_{s_{i,t+1}\sim \mathrm{P}(\cdot|s_{i,t},u_{i,t})}[\omega(s_{i,t+1})] \\
&- \mathbb{E}_{i\in\mathcal{I},s_{i,t}\sim d_t^{\pi}(i),u_{i,t}\sim\pi(s_{i,t})}[\mathrm{r}(s_{i,t},u_{i,t}))] - \gamma\mathbb{E}_{i\in\mathcal{I},s_{i,t+1}\sim d_{t+1}^{\pi}(i)}[\mathrm{V}_{\pi}(s_{i,t+1})] + \omega(s_{i,t}) \\
=&\mathrm{r}(s_{i,t},u_{i,t}) + \gamma\bar{V}_{t+1} - \gamma\mathbb{E}_{s_{i,t+1}\sim \mathrm{P}(\cdot|s_{i,t},u_{i,t})}[\omega(s_{i,t+1})] \\
&- \mathbb{E}_{i\in\mathcal{I},s_{i,t}\sim d_t^{\pi}(i),u_{i,t}\sim\pi(s_{i,t})}[\mathrm{r}(s_{i,t},u_{i,t}))] - \gamma\bar{V}_{t+1} + \omega(s_{i,t}) \\
=&\underbrace{\mathrm{r}(s_{i,t},u_{i,t}) - \mathbb{E}_{i\in\mathcal{I},s_{i,t}\sim d_t^{\pi}(i),u_{i,t}\sim\pi(s_{i,t})}[\mathrm{r}(s_{i,t},u_{i,t}))]}_{\text{Reward Term}} \underbrace{-\gamma\mathbb{E}_{s_{i,t+1}\sim \mathrm{P}(\cdot|s_{i,t},u_{i,t})}[\omega(s_{i,t+1})] + \omega(s_{i,t})}_{\text{Residual Term } \epsilon_{i,t}}
\end{aligned}
\tag{26}
$$

Specifically, the residual term satisfies:

$$
|\epsilon_{i,t}| \le |\omega(s_{i,t})| + \gamma\big|\mathbb{E}[\omega(s_{i,t+1})]\big| \le 2\delta\,\overline{V}_t. \tag{27}
$$

Consider a scenario in which the fleet-average value $\overline{V}_t$ admits a finite upper bound (a condition automatically satisfied in any finite-horizon setting with episode length T). Assume further that the system exhibits sufficiently fast mixing, such that the required mixing lag κ is only mildly affected by the horizon length. As shown previously, the relative error δ can be made arbitrarily small by choosing a discount factor γ sufficiently close to 1 and an appropriate mixing lag κ. Consequently, one can select these parameters such that the residual term $|\epsilon_{i,t}|$ becomes arbitrarily small and can therefore be safely neglected in the advantage approximation.

C Algorithm

The detailed algorithm of our ST-GRPO/OSPO for ride-sharing system is illustrated in Algorithm 1.

D Experiment Details

D.1 Simulator Pipeline and Reward Design

The dispatch pipeline couples a learned scoring network with a routing oracle. At each step the policy network produces a score for every feasible vehicle-order pair, the matching problem in Eq. (10) is solved to obtain the assignment, and the simulator then updates the corresponding vehicle route, recomputing the travel-time consequences for all en-route and newly assigned passengers. Routing and travel times are obtained from OSRM [23], which we query with a sequence of waypoints and which returns the shortest route together with its travel time; OSRM therefore serves as a travel-time oracle for evaluating candidate decisions rather than as an object of optimization. In the current setup an assigned order is not later reassigned to another vehicle, which follows the protocol of the compared methods.

The reward in Eq. (1) encodes the platform objective through passenger income, operational cost, an overtime penalty for en-route orders, and an additional travel-time penalty, and it follows the formulation used in prior ride-sharing studies [30; 5]. All compared methods are trained and evaluated under this same reward signal, so the comparison does not favor any method through the reward specification. Because the weights η_1 to η_5 express the platform's relative valuation of revenue, cost, and passenger delay, different weight settings would induce different trade-offs among pickup time,

Algorithm 1 ST-GRPO/OSPO for Ride-Sharing System

Require: Policy network π_θ with shared parameters θ
Require: Best policy checkpoint π_ϕ (initialized as π_θ)
Require: Environment simulator, Reward function (Eq. (1))
for episode $= 1$ to M **do**
  Initialize experience buffer $\mathcal{D} \leftarrow \emptyset$
  Reset environment, get initial states $\{s_{i,0}\}_{i=1}^{n}$, s_0^o
  for time step $t = 0$ to T **do**
    for each agent $i \in \mathcal{I}$ **do**
      Compute action probabilities (Eq. (9))
    end for
    Solve assignment problem via BiPartite matching (Eq. (10)), getting action U_t
    Execute assignment U_t, observe rewards $\{r_{i,t+1}\}_{i=1}^{n}$
    Store transition $(\{s_{i,t}\}, s_t^o, U_t, \{r_{i,t+1}\}, \{s_{i,t+1}\}, s_{t+1}^o)$ in $\mathcal{D}$
  end for
  for epoch $= 1$ to H **do**
    Sample batch $\mathcal{B} \sim \mathcal{D}$
    Compute group advantage $\mathcal{A}_t$
    Update policy parameters θ by maximizing Eq. (2)
  end for
  if current policy π_θ outperforms π_ϕ **then**
    Update best checkpoint: $\phi \leftarrow \theta$
  end if
end for

detour time, and service rate; we report our settings in Section 4.1 and release the configuration with the code.

Joint routing and dispatch optimization, and reassigning orders across vehicles within an episode, are useful extensions that we leave to future work. They enlarge the joint action space and couple several decisions within a single step, which places them alongside recent work on repositioning and order bundling [36; 37] rather than within the scope of the present dispatch formulation.

D.2 Evaluation Metrics

The evaluation metrics used in this paper include:

- **Served Rate:** The proportion of confirmed trips relative to the total number of trip requests initiated by customers.
- **Delivery Time:** The total time taken to serve a trip from origin to destination.
- **Detour Time:** The additional time spent on delivery beyond the minimum delivery time, which is defined as the time required if the vehicle serves only this trip without bundling others.
- **Pickup Time:** The waiting time for customers between trip confirmation and the arrival of the vehicle at the trip origin.
- **Confirmation Time:** The waiting time for customers from the moment they initiate a trip request to when the platform assigns the trip to a vehicle.

Specifically, for delivery time and detour time, only completed orders are counted, as these metrics are uncertain for unfinished orders.

D.3 Introduction of Comparative Methods

To illustrate the efficiency of our methods, we compare them with several representative ride-sharing approaches from various categories:

- **DeepPool [26]:** DeepPool is one of the earliest MARL-based ride-sharing works, based on IDDQN [39; 40], in which each agent trains its Q-network separately. To align it with our experimental

scenario, we replace the CNN used in original paper with an MLP similar to our methods (without the softmax).

- **BMG-Q [5]:** To achieve better cooperation among agents, BMG-Q is also based on IDDQN but utilizes Graph Attention Network (GAT) [41] to capture the relationships among neighboring agents.
- **HIVES [32]:** Based on QMIX [33], HIVES tries addressing the curse of dimensionality problem by proposing a novel hierarchical mixing structure.
- **Enders et al. [19]:** Based on Multi-Agent Soft Actor Critic (MASAC) [42], Enders et al. propose allowing agents to choose whether to accept assigned orders, which helps the system serve more valuable orders instead of treating all orders equally.
- **CEVD [34]:** CEVD modifies the Value Decomposition (VD) paradigm [35] from CTDE to CTCE. Specifically, it combines the Q-values of each agent with those of their neighbors to create a new type of Q-value, similar to the motivation behind BMG-Q.
- **Hoppe et al. [20]:** Building on the work of [19], a method based on COMA [43] is developed that utilizes a mixture of global and local rewards. Specifically, the global reward encourages cooperation, while the local reward enhances training efficiency.

For fairness, we modify the state space and reward function to align closely with those used in this paper.

D.4 Detailed Experiment Results

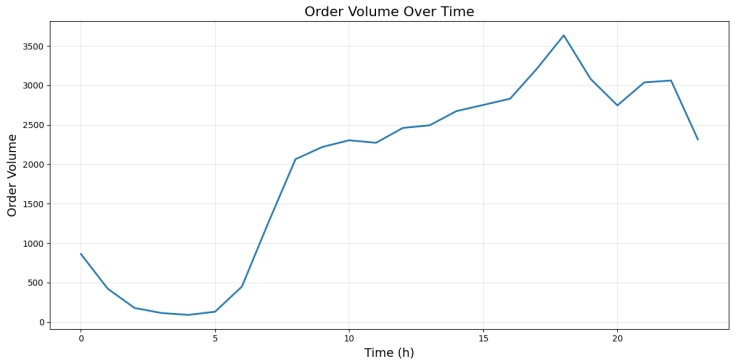

Figure 3: Order Temporal Distribution

In Fig. 3 and 4, we illustrated the order amount and the metrics of different methods under each hour in the whole day. First, it's intuitive that the trend of reward is similar to the order volume, since more orders can bring more profits to platform. Then we notice before 8 a.m., when the order volume is very low, the overall performance of different methods is similar, since the idle vehicles are sufficient. However, after that, the BMG-Q and HIVES first illustrate slightly advantage during 8 a.m. to 1 p.m. and then are outperformed by our ST-GRPO and OSPO methods as the order volume further increases after 1 p.m.. Overall, our ST-GSPO and OSPO mainly outperform others by optimizing the pickup time and confirmation time, leading to the highest order service rate in the on-peak hour 6 p.m..

Additionally, to further prove our homogeneity assumption, we illustrate the relationship among initial state (location) and metrics regarding cumulative reward (without discount factor γ) and reward-to-go (with discount factor γ) of our method. The results suggest that in our method, the state does not significantly influence the long-term normalized cumulative reward and reward-to-go. As shown in Fig. 5, this result supports our homogeneity assumption from a localization perspective. Specifically, the Coefficient of Variation (CV) of the reward-to-go (V-value estimation) is only 13.45%, indicating relatively low divergence.

D.5 Ablation Study

In the ablation study, we primarily investigate the effects of:

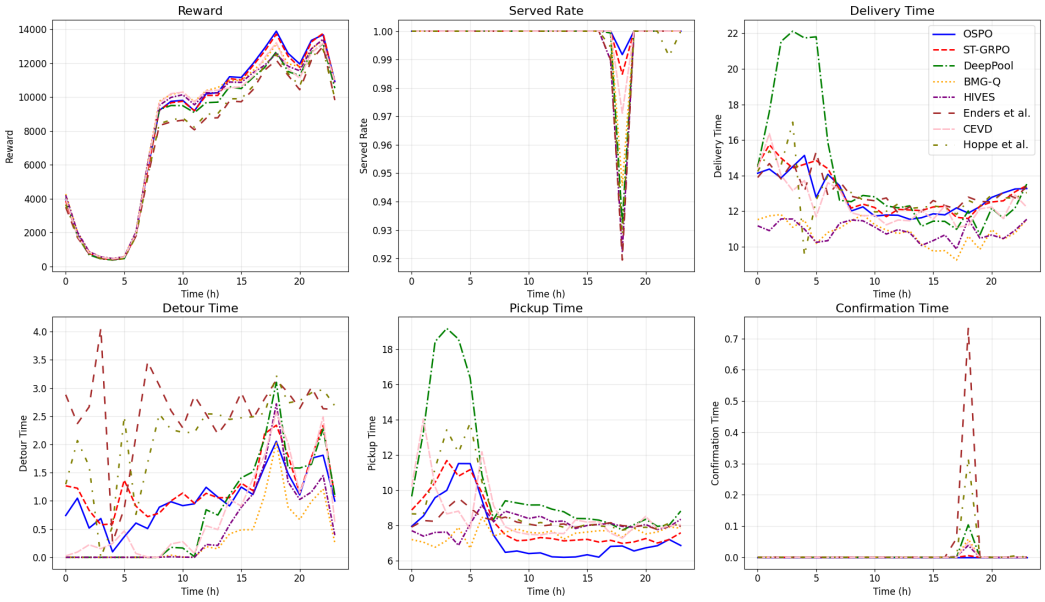

Figure 4: Experiment Results in Testing Set

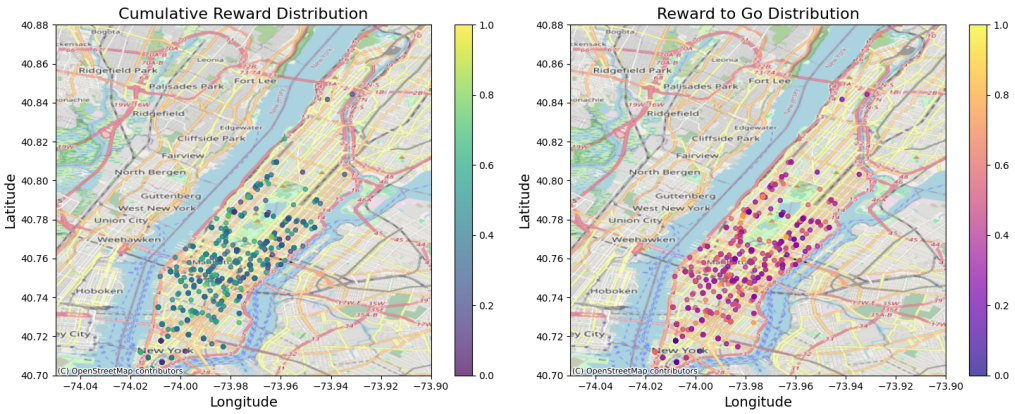

Figure 5: The relationship between initial location and metrics about cumulative reward and reward-to-go.

- **Reward Normalization:** We evaluate the effect of reward normalization in both ST-GRPO and OSPO by testing model performance after removing this module.
- **Deviation Punishment Regulation:** We examine the impact of including deviation punishment ($\zeta_t - \zeta_{t-1}$ in Eq. (16)) to determine whether it helps to maintain homogeneous property (Eq. (7)).
- **KL Regulation Term:** We assess the effect of using a KL regulation term ($\mathrm{KL}(\pi_\theta \| \pi_{ref})$ in Eq. (2)) to illustrate whether the historical best policy can prevent model divergence.

Additionally, we compare our methods with conventional policy optimization methods, including IPPO, MAPPO, and IPG. As shown in Fig. 6, we notice that the proposed ST-GRPO and OSPO outperform all variant versions, illustrating the efficiency of the proposed modules. For the policy optimization methods, we observed that only IPPO demonstrates relatively good performance (even when compared to our proposed methods), while IPG struggles with high derivation and low sample efficiency. Additionally, MAPPO suffers from the curse of dimensionality in the centralized critic network, resulting in failure to converge.

Additionally, we observe that OSPO demonstrates overall better performance than ST-GRPO. This advantage arises from the nature of our ride-sharing scenario, where the number of served orders

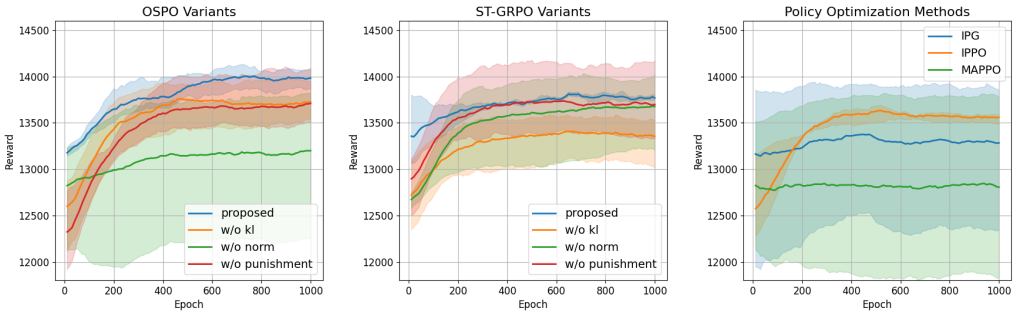

Figure 6: Training Process of Ablation Study

Table 2: Performance of Different Methods in Queens, New York City [22]

Method	Reward	Service Rate	Delivery Time	Detour Time	Pickup Time	Confirmation Time
DeepPool	5222.85	0.64	**11.24**	1.84	12.30	**0.21**
BMG-Q	5362.00	0.66	9.63	1.08	12.98	0.27
HIVES	3560.80	0.60	8.30	**0.36**	14.67	0.41
Enders et al.	4543.68	0.61	10.41	0.85	13.39	2.25
CEVD	4388.83	0.62	11.61	1.33	13.74	0.29
Hoppe et al.	4728.93	0.63	9.78	0.53	13.45	2.18
ST-GRPO	5365.89	0.64	10.56	1.86	12.21	**0.12**
OSPO	**5521.74**	**0.67**	11.37	2.21	**11.88**	0.21

may vary slightly among agents and episodes. For instance, an agent may receive a newly assigned order at the last step of the simulation, resulting in a relatively higher reward for that episode. While this variation is negligible over the long term, it becomes more pronounced within the confines of our 30-minute simulation when calculating the advantage based on the average reward in Eq. (13). In contrast, OSPO exhibits greater robustness, as it relies solely on one-step rewards and does not encounter this issue.

D.6 Expanded Experiment

To further demonstrate the generalization of our method, we conduct experiments using High Volume For-Hire Vehicle (FHV) trip data from Queens, New York City [22]. Unlike the Manhattan data used in our previous experiment, Queens is a larger area with a lower order density. This difference in order distribution can complicate the process of determining which orders to bundle together for vehicle sharing. In this scenario, we select the data from 19:00 to 19:30 on July 17, 2024, which includes 2,024 valid orders and 500 vehicles. The detailed experimental results are presented in Table 2.

The results demonstrate that our proposed ST-GRPO and OSPO methods continue to exhibit superior performance, primarily by optimizing pickup and confirmation times. Specifically, OSPO significantly outperforms the other methods, including ST-GRPO. This advantage arises because, in a more challenging and dynamic scenario (due to the enlarged area), the ST-GRPO method is more severely impacted by errors associated with the bounded time horizon, whereas OSPO leverages a one-step reward to mitigate this shortcoming.

The experiments above keep the fleet size, vehicle capacity, and travel speed at their training values, and evaluate on a different day or region. Here we report three further evaluations that each perturb one of these factors directly. The results are summarized in Tables 3, 4, and 5. In all three, we evaluate the trained models without any retraining. Throughout these three experiments, the reported deviations reflect variation across independent training seeds evaluated on a single 30-minute time window; this differs from the main comparison in Table 1, where the deviations reflect variation across the 24 hourly test windows of a single day.

Table 3: Performance under randomly sampled travel speeds, $\mathcal{N}(60, 36)$ km/h.

Method	Reward	Service Rate	Delivery Time	Detour Time	Pickup Time	Confirmation Time
DeepPool	13,393.20	0.92	11.25	3.36	7.31	0.14
BMG-Q	13,780.69	0.94	**10.66**	2.04	7.07	0.15
HIVES	12,986.26	0.90	11.02	2.43	7.56	**0.07**
Enders et al.	12,844.43	0.91	12.93	2.41	7.55	1.09
CEVD	13,959.38	0.96	10.79	2.12	6.88	0.15
Hoppe et al.	13,063.01	0.91	12.80	2.35	7.59	0.32
ST-GRPO	13,867.12	0.95	10.78	**1.92**	6.95	0.10
OSPO	**13,986.23**	**0.97**	11.42	2.09	**6.82**	**0.07**

Table 4: Performance under a supply-constrained setting with 500 vehicles.

Method	Reward	Service Rate	Delivery Time	Detour Time	Pickup Time	Confirmation Time
DeepPool	7,676.40	0.50	12.30	3.36	6.61	0.31
BMG-Q	7,784.56	0.52	12.23	3.77	6.82	0.48
HIVES	5,273.77	0.31	15.46	**1.52**	9.04	**0.07**
Enders et al.	6,525.59	0.48	12.98	3.32	7.88	2.72
CEVD	3,100.06	0.17	14.89	1.70	8.40	0.69
Hoppe et al.	6,683.32	0.47	13.21	3.30	7.65	1.17
ST-GRPO	7,833.37	0.53	12.23	3.29	6.34	0.12
OSPO	**8,086.97**	**0.54**	**12.20**	3.32	**6.05**	0.22

Random travel speeds. The simulator uses a fixed speed of 60 km/h during training, which is a simplification of real traffic conditions. We therefore resample each vehicle's speed at every step from $\mathcal{N}(60, 36)$ km/h, which perturbs the transition dynamics and the resulting travel-time consequences for all en-route passengers. As shown in Table 3, OSPO attains the highest reward, service rate, and pickup time, while ST-GRPO achieves the best detour time. The ranking among the proposed methods and the strongest baselines is preserved relative to the fixed-speed setting, which indicates that the learned score functions do not depend on a particular travel-time realization.

Supply-constrained operation. The main comparison deliberately uses an abundant fleet of 1,000 vehicles. We halve the fleet to 500 vehicles while keeping the same demand, which yields a supply-constrained regime where vehicles are rarely idle and the dispatch decision has a much larger effect on which orders are ultimately served. Table 4 shows that the differences between methods widen considerably under this stress: HIVES and CEVD degrade sharply, with service rates falling to 0.31 and 0.17, whereas ST-GRPO and OSPO preserve both the highest reward and the highest service rate. The sensitivity of the centralized-critic baselines is consistent with our motivation in Section 2: their value estimates must track the joint interaction pattern, which changes substantially when the supply-demand ratio shifts, while the group-relative statistics used by our methods adapt to the new reward scale directly. This experiment also explains why the margins in Table 1 are modest. That comparison averages over the full day, including several low-demand windows in which idle capacity is sufficient and any reasonable matching rule performs well; the differences between methods are concentrated in the on-peak windows, which is also visible in Fig. 4.

Reduced vehicle capacity. Finally, we reduce the per-vehicle capacity from 3 to 2 and evaluate the trained policies without retraining, to test whether the learned scores transfer when vehicles can carry fewer simultaneous orders. Table 5 shows that the two methods behave similarly as capacity is reduced: pickup and confirmation times are essentially unchanged, while reward and service rate decline because bundling fewer orders per vehicle necessarily lowers the total throughput. The decline is an effect of the smaller achievable throughput rather than a loss of policy quality, and it indicates that a policy trained at capacity 3 can be deployed directly when fewer orders fit in each vehicle.

Enlarging the capacity beyond the training value is not supported by the current architecture, because the policy network consumes a fixed-size vehicle-order pair representation and the input cannot accommodate a longer list of en-route orders. This limitation is shared by the compared baselines. Capacity-aware architectures, together with heterogeneous fleets, are discussed as future directions in Appendix E.7.

Table 5: Performance under reduced vehicle capacity, evaluated without retraining.

Method	Capacity	Reward	Service Rate	Delivery Time	Detour Time	Pickup Time
ST-GRPO	2	13,263.23	0.87	**11.55**	**1.66**	7.24
ST-GRPO	3 (train)	13,727.33	0.96	12.32	2.87	7.33
OSPO	2	13,697.24	0.90	11.78	1.67	**6.78**
OSPO	3 (train)	**14,003.78**	**0.97**	11.80	2.81	7.12

E Discussions

E.1 On the Value Homogeneity Property

The value homogeneity property in Eq. (7) underpins both ST-GRPO and OSPO, and Lemma 1 in Appendix B.1 establishes it in a relative sense. Here we relate that result to what is actually observed during training.

Fig. 5 plots the cumulative reward and the reward-to-go against a vehicle's initial location, and shows no pronounced spatial structure in either quantity. The coefficient of variation of the reward-to-go is 13.45%, so the typical spread of the finite-horizon value estimate across initial states is well below its mean. We emphasize that this is an indirect localization check rather than a verification of Lemma 1: the lemma requires a uniform total-variation bound on the κ-step transition kernels and a positive lower bound on the expected one-step reward, and neither can be certified by measuring the dispersion of realized returns.

It is nonetheless informative to compare magnitudes. Taking the fleet-average value scale as $\bar{V}_t \approx 11$ and using the coefficient of variation as a rough proxy for the relative tolerance δ, the worst-case residual bound in Eq. (27) evaluates to approximately $2\delta\bar{V}_t \approx 3.0$. This is a deliberately loose figure: it is the tightest consequence of Lemma 1 alone, applied with a δ that was estimated rather than selected. Two observations place it in context.

First, the lemma attains arbitrarily small δ only in an asymptotic regime, in which γ is driven sufficiently close to 1 and the required mixing lag κ grows accordingly. The training configuration used here does not operate in that regime: we use a finite-horizon truncation of the ride-sharing episode and a conventional discount factor $\gamma = 0.99$. The policy optimization therefore proceeds under conditions that are weaker than those the bound requires, and the resulting policies still improve on the value-function-based baselines across every setting we test (Tables 1, 2, 3, 4).

Second, the bound in Eq. (27) is a worst-case quantity, attained only if the deviation $\omega(s_{i,t})$ reaches its maximum at every step and remains perfectly aligned with the bootstrapped next-step deviation. Because $|\epsilon_{i,t}| \leq |\omega(s_{i,t})| + \gamma|\mathbb{E}[\omega(s_{i,t+1})]|$ involves a sum of two terms of a zero-mean residual, their typical magnitude is substantially smaller than the sum of their worst-case magnitudes. The residual that OSPO discards is therefore bounded and structured rather than arbitrary, in contrast to the approximation bias of a learned critic, which is neither bounded nor signed. This distinction is what the design trades on, and it is consistent with the observation that our methods remain competitive with IPPO while MAPPO does not converge at all in this setting (Fig. 6).

E.2 Probabilistic Assignment and the Matching Layer

The policy network outputs a score $p_{i,j,t}$ for each vehicle-order pair, and the executed assignment is obtained by solving the bipartite matching problem in Eq. (10). This two-stage structure raises a natural question about what the policy gradient in Eq. (2) actually optimizes, since the matching layer is a deterministic argmax over a combinatorial set and the induced distribution over joint assignments is not available in closed form.

Probabilistic assignment enters our framework at the exploration stage. Following prior policy-based ride-sharing methods [19; 20], we perturb the score matrix before matching: a randomly corrupted copy of the scores is passed to the matching solver, so that the executed assignment is stochastic and vehicle-order pairs with higher scores appear more often, though not with certainty, in the resulting matching. The perturbation follows the Binary Symmetric Channel noise used in [26; 5], and its magnitude decays over training. Because the perturbation is applied to the scores rather than to the matching outcome, the preference encoded by the network directly shapes which assignments are

sampled. During exploitation no perturbation is added, and the matching layer simply returns the highest-utility feasible assignment under the learned scores.

We describe the resulting update as a structured surrogate for the policy gradient rather than an exact likelihood-ratio estimator. An exact estimator would require the normalized probability of each feasible joint matching, which is intractable at the scale considered here, where the number of candidate assignments grows combinatorially with the fleet size. The surrogate instead preserves the ordering induced by the scores and supplies the exploration needed for the group-relative advantage to be informative: at each step, the fleet-wide spread of one-step rewards reflects the quality of the pairs that the perturbed matching selected. This is the same approximation that underlies the compared methods [19; 20], and our results indicate that it is sufficient to learn effective dispatch policies. More principled formulations, such as differentiable matching or explicit sampling over feasible assignments, would tighten the connection to the policy gradient and are a natural direction for future work.

E.3 Bias and Variance Analysis

The methods compared in Section D.5 occupy different positions in the bias-variance trade-off, and the design of ST-GRPO and OSPO is best understood in those terms.

Textbook policy gradient methods, represented by IPG in Fig. 6, use the Monte Carlo reward-to-go directly and are therefore unbiased, at the cost of variance that grows with the horizon and with the stochasticity of the environment. The PPO family replaces the reward-to-go with an advantage estimate built on a learned critic. This reduces variance substantially, but it substitutes sampling noise for approximation bias: the critic's error depends on how well a neural network can fit the value function of a non-stationary, strongly interacting multi-agent system. The failure modes of the two members of this family differ in a way that is instructive. IPPO conditions its critic on a single agent's observation and converges, whereas MAPPO conditions its critic on the joint state and does not converge here at all (Fig. 6); the only difference between the two is the scope of the critic's input, so the divergence cannot be attributed to the policy update itself. The centralized critic must track a value function whose input dimension scales with the fleet size, which is the curse of dimensionality reappearing inside the critic rather than in the policy.

ST-GRPO and OSPO start from the same advantage decomposition as IPPO and therefore inherit its low-variance benefit, but replace the network-estimated value with fleet-relative reward statistics: the average reward-to-go in the case of ST-GRPO, and the average one-step reward in the case of OSPO. This removes the critic and with it the approximation bias, at the cost of the residual term analyzed in Appendix B.2, whose magnitude is governed by the homogeneity property established in Appendix B.1. The trade is favorable in this setting precisely because the residual is bounded and the critic error is not, and the two do not have to be weighed against each other in a regime where the homogeneity property holds. Improving on both the bias of the PPO family and the variance of IPG is possible here because the structure of the task, rather than a better estimator, makes the substitution valid. Where that structure is absent, for instance in a heterogeneous fleet or a weakly connected operating region, the residual term would no longer be controlled and the substitution would no longer be justified; Appendix E.4 states these conditions explicitly.

One component of our method is not covered by this analysis. The reference policy π_{ref} in Eq. (2) is instantiated as the best checkpoint observed so far, rather than as a fixed pre-training policy. This choice keeps the KL term meaningful in a setting where no separate reference policy exists, and it performed well in the ablation of Section D.5 (the KL regulation entry), but it makes the optimization objective depend on the training history and falls outside the assumptions under which the residual bound is derived. Convergence of the full training procedure with this heuristic in place is therefore not claimed here, and characterizing it is left for future work.

E.4 Application Conditions of the Proposed Methods

In this section, we summarize the conditions under which our methods are applicable and explain their practical implications in real-world AV ride-sharing systems. First, both ST-GRPO and OSPO require the following two conditions:

Condition I: Homogeneity (Identical AV Fleet): All agents must share the same internal attributes, state space, action space, and reward function, and the task must be fully cooperative. In the context of ride-sharing systems, this means that all AVs adopt the same vehicle model, the system policy is non-discriminatory, and each agent's objective is to maximize the global cumulative reward.

Condition II: Short Memory (Rapid Mixing): The system must be insensitive to initial conditions after a moderate time horizon. Concretely, this requires that the operational region be sufficiently connected and that the order distribution be diverse enough to relocate vehicles across different areas within a short period. It means that orders have a roughly equal probability of driving each AV to arbitrary locations.

Moreover, our methods rely on a problem setup condition: the discount factor γ should be sufficiently large. This condition is used to establish the V-value homogeneity property, which depends on the objective construction rather than the system dynamics themselves. Furthermore, OSPO introduces an additional application condition:

Condition III: Bounded V-Value (e.g. Finite Horizon): In OSPO, the condition for replacing the reward-to-go with a single-step reward is that the V-value is upper bounded. This condition is satisfied when the horizon is finite and the system mixes quickly, which is practical in ride-sharing systems where the running time is finite and newly assigned orders can rapidly relocate AVs to new positions.

Remark on the role of the discount factor and practical interpretation: The discount factor γ appears only in the theoretical analysis that connects OSPO to long-horizon objectives; OSPO itself does not use γ in its updates. The key practical requirement is that the system mixes much faster than the effective horizon of the objective, i.e., $1/(1-\gamma) \gg \kappa_{\text{mix}}$. Under this condition, meaning that the planning horizon extends well beyond the time it takes for a vehicle's state to "forget" its initial conditions, the residual terms in the advantage approximation become negligible relative to the one-step group-normalized reward signal. The proofs assume non-negative rewards ($r \in [0, R_{\max}]$) to establish a lower bound on $\bar{V}_t$; this can be satisfied in practice by shifting or clipping the reward function. A constant shift cancels exactly in both ST-GRPO and OSPO: for OSPO, the numerator $r - \mu_t$ and denominator σ_t are shift-invariant; for ST-GRPO, the reward-to-go $G_{i,t}$ receives the same additive shift for all agents at time t, which also cancels in $G_{i,t} - \mu_{t:}$. Thus, when the fleet is homogeneous and the ride-sharing network mixes rapidly (Condition II), both ST-GRPO and OSPO are theoretically justified and empirically effective, with OSPO requiring no explicit discount factor and only one-step rewards.

E.5 Behavior When the Conditions Are Not Met

Conditions I, II, and III in Appendix E.4 delimit the settings in which the residual bound of Appendix B.2 applies, and it is worth stating what follows when they fail.

A heterogeneous fleet breaks Condition I directly. Once vehicles differ in capacity, speed, or operating constraints, the shared-policy argument no longer holds and the residual $\omega(s_{i,t})$ can no longer be bounded by the mixing argument alone, since agents with different capabilities have genuinely different long-run values. A partial remedy is to restrict the advantage computation to neighboring vehicles with similar characteristics, so that the fleet average becomes a local average over a more homogeneous subset; this preserves the group-relative structure while narrowing the population over which homogeneity is asserted.

A weakly connected operating region breaks Condition II. If orders rarely relocate vehicles between areas, the system retains a memory of initial conditions and the mixing lag κ becomes comparable to or larger than the effective horizon $1/(1-\gamma)$, which invalidates the total-variation bound in Eq. (6). We do not have a guarantee in this regime, and we note that the spatial heterogeneity literature [17; 18] documents exactly the kind of persistent disparities that would be expected there.

Condition III concerns the boundedness of the value function and is the mildest of the three, since any finite episode length suffices. Across all three cases, the empirical results in Appendix D.6 indicate a degree of robustness to perturbations that do not change the fleet composition or the connectivity of the region, such as randomized speeds, a reduced fleet, or a smaller capacity.

Table 6: Comparison of GRPO [6], ST-GRPO, and OSPO

	GRPO [6]	**ST-GRPO**	**OSPO**
Application Scenario	LLM post-training	Cooperative MARL with Conditions I and II	Cooperative MARL with Conditions I, II, and III
Theoretical Guarantee	$\times$	✓	✓
Training Samples	Multiple trajectories from the same initial state	Single trajectory of multiple agents	Single-step samples from multiple agents
Normalization Term	Reward-to-go over multiple trajectories	Reward-to-go over multiple agents	Single-step reward over multiple agents

The guarantees for ST-GRPO and OSPO refer to the residual bound on the group-relative advantage established in Appendix B.2 under Conditions I, II, and III. The reference-policy heuristic we introduce in place of π_{ref} is not covered by this analysis, so convergence of the full training procedure is not claimed; we discuss this in Appendix E.

E.6 Comparison between GRPO, ST-GRPO, and OSPO

To illustrate the differences between our proposed methods (ST-GRPO and OSPO) and the original GRPO [6], we summarize them in Table 6.

E.7 Limitations and Future Work

Our method is specifically tailored to homogeneous AV ride-sharing systems with fast mixing speed. While this design choice trades off generality for training efficiency and strong empirical performance, the underlying conditions are readily met in many practical deployments. For instance, many ride-sharing systems operate with identical AV fleets within well-connected metropolitan areas (e.g., differently colored taxis in Hong Kong and New York), which naturally leads to fast mixing. Experimental results also demonstrate the robustness of our method across different scenarios under similar experimental settings as previous work. As a next step, it is worthwhile to explore the applicability of our method to tasks that share similar conditions with our AV ride-sharing setting. It is also worth investigating how to extend our method to more general ride-sharing scenarios, such as those with heterogeneous fleets.

E.8 Broader Impact

On the positive side, more efficient dispatch improves vehicle utilization and reduces passenger waiting times. Because the reward penalizes detours and overtime, the learned policies are also incentivized to avoid unnecessary travel, which lowers energy consumption and emissions relative to dispatching rules that ignore travel-time consequences; the service-rate improvements reported in Section 4 mean that the same demand can be met with fewer vehicle-kilometers.

The same efficiency objective carries risks. Optimizing platform-level throughput can concentrate service in high-demand areas and leave low-demand regions with longer waits, and it can favor order bundles that are profitable for the platform over those that best serve individual passengers. In deployments that mix autonomous and human-driven vehicles, dispatch policies of this kind also affect the distribution of work and income across drivers. These effects are properties of the objective rather than of the optimizer, and the homogeneity and connectivity conditions under which our method applies are themselves unevenly distributed across cities: peripheral and less connected areas are precisely where Condition II is most likely to fail. Evaluating dispatch policies on distributional metrics alongside aggregate efficiency, and extending the approach to heterogeneous fleets, would make these trade-offs measurable rather than implicit.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes], [No], or [N/A].
- [N/A] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes]" is generally preferable to "[No]", it is perfectly acceptable to answer "[No]" provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No]" or "[N/A]" is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction accurately reflect the paper's contributions by introducing two highly efficient MARL solutions for the AV ride-sharing task, which eliminate value estimation errors from neural networks through a group-relative advantage calculation method, achieving superior performance with low GPU utilization.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Please refer to Appendix E.7.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: Please refer to Appendix E.4.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: All configurations are provided at Section 4.1

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.

- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The code for this paper is available in an anonymous repository at `https://anonymous.4open.science/r/OSPO-2105`.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: The experiment setup is provided at Section 4.1 and Appendix D.2.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: All experiments were repeated with five random seeds, and the shaded regions in the training curves represent the standard deviation.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: The compute resources are described in Table 1.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

Answer: [Yes]

Justification: [N/A]

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: Please refer to Appendix E.8, where we discuss improved vehicle utilization and reduced emissions on the positive side, and the risk of uneven service quality across passengers and regions on the negative side.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: [N/A]

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.

- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We have cited the experiment dataset utilized in this paper [22].

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: [N/A]

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: This paper uses LLMs solely to check grammar and spelling.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.