GARLIC: GPT-Augmented Reinforcement Learning with Intelligent Control for Vehicle Dispatching

Xiao Han[1], Zijian Zhang[2], Xiangyu Zhao[1*], Yuanshao Zhu[1], Guojiang Shen[3], Xiangjie Kong[3], Xuetao Wei[4], Liqiang Nie[5], Jieping Ye[6]

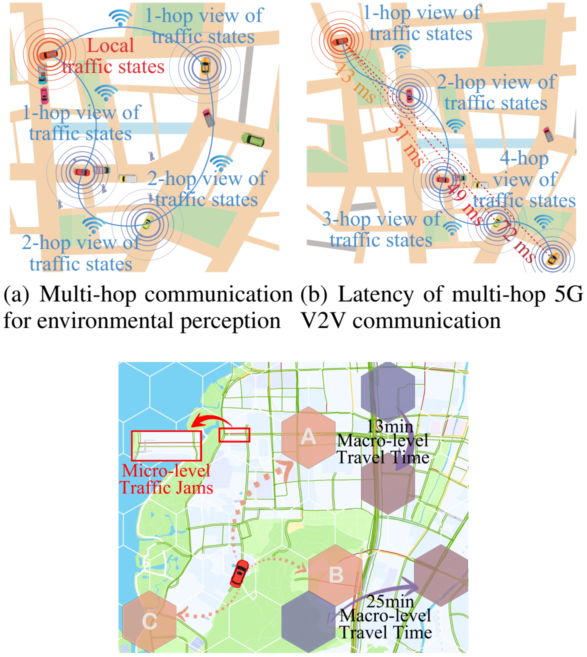
[1]City University of Hong Kong
[2]Jilin University
[3]Zhejiang University of Technology
[4]Southern University of Science and Technology
[5]Harbin Institute of Technology (Shenzhen)
[6]Alibaba Group

hahahenha@gmail.com, zhangzijian@jlu.edu.cn, xianzhao@cityu.edu.hk, yaso.zhu@my.cityu.edu.hk, gjshen1975@zjut.edu.cn, xjkong@ieee.org, weixt@sustech.edu.cn, nieliqiang@gmail.com, jieping@gmail.com

Abstract

As urban residents demand higher travel quality, vehicle dispatch has become a critical component of online ride-hailing services. However, current vehicle dispatch systems struggle to navigate the complexities of urban traffic dynamics, including unpredictable traffic conditions, diverse driver behaviors, and fluctuating supply and demand patterns. These challenges have resulted in travel difficulties for passengers in certain areas, while many drivers in other areas are unable to secure orders, leading to a decline in the overall quality of urban transportation services. To address these issues, this paper introduces GARLIC: a framework of GPT-Augmented Reinforcement Learning with Intelligent Control for vehicle dispatching. GARLIC utilizes multiview graphs to capture hierarchical traffic states, and learns a dynamic reward function that accounts for individual driving behaviors. The framework further integrates a GPT model trained with a custom loss function to enable high-precision predictions and optimize dispatching policies in real-world scenarios. Experiments conducted on two real-world datasets demonstrate that GARLIC effectively aligns with driver behaviors while reducing the empty load rate of vehicles.

Code — https://github.com/Applied-Machine-Learning-Lab/GARLIC


1-hop view of traffic states
Local traffic states
1-hop view of traffic states
2-hop view of traffic states
2-hop view of traffic states
1-hop view of traffic states
13 ms
2-hop view of traffic states
31 ms
4-hop view of traffic states
49 ms
3-hop view of traffic states
72 ms
13min Macro-level Travel Time
A
Micro-level Traffic Jams
B
25min Macro-level Travel Time
C

(a) Multi-hop communication for environmental perception (b) Latency of multi-hop 5G V2V communication

(c) A case of vehicle dispatching

Figure 1: A vehicle dispatching scenario.

1 Introduction

The past decade has witnessed explosive growth in online car-hailing services, fundamentally transforming urban transportation. Central to this transformation is the role of vehicle dispatching (Shi et al. 2024a), which serves as a pivotal component in reducing the waiting time of passengers, increasing the income of drivers, and facilitating daily transportation (Barrios, Hochberg, and Yi 2023; Rahman and Thill 2023; Sadrani, Tirachini, and Antoniou 2022). In recent years, reinforcement learning (RL) methods have emerged as outstanding performers in areas such as multi-agent control and sequential decision-making (Qiu et al. 2023; Ellis et al. 2024; Han et al. 2023a). Therefore, many studies have leveraged RL techniques to enhance vehicle dispatching, treating it as a multi-agent sequential decision-making task (Guo et al. 2024; Huang et al. 2023).

However, unlike traditional multi-agent reinforcement learning (MARL) approaches applied in other domains, vehicle dispatching presents a unique challenge due to the complex interplay between observable local traffic states and undetectable global spatiotemporal correlations. Each vehicle acts as an individual agent, with access limited to the environmental states in its immediate vicinity. This makes it

[*]Corresponding author.

difficult to obtain a comprehensive, global view of vehicle supply and demand. As illustrated in Figure 1(a), a vehicle must rely on multiple hops of vehicle-to-vehicle (V2V) communication to acquire more extensive traffic flow information. Furthermore, expanding a vehicle's receptive field exponentially increases communication latency among agents (Huang and Lin 2022; Wang 2023; Han et al. 2024), as depicted in Figure 1(b). According to the traffic flow theory (Gerlough and Huber 1976), traffic flows also behave differently at diverse granularities. For instance, macro-level traffic flow provides an overview of travel times, as shown by the arrows between purple and brown grids in Figure 1(c). In contrast, micro-level traffic states can pinpoint traffic jams directly, as illustrated by the different road segment colors (green, yellow, and red) in Figure 1(c). In summary, obtaining a comprehensive and accurate view of traffic states is a significant challenge in vehicle dispatching.

Accurate vehicle dispatching also necessitates nuanced driving behavior modeling, which accounts for the individual preferences of different vehicle agents regarding dispatching instructions. Driving behavior reflects the driver's personal inclination toward specific dispatching tasks, and plays a crucial yet often overlooked role in transportation (Wang et al. 2024a; Robbennolt and Levin 2023; Zhang et al. 2023d; Han et al. 2023b). For example, consider the taxi driver of the red car in Figure 1(c), who is more familiar with region A. This driver might prefer to pick up passengers in region A rather than in the unfamiliar regions B or C, even if those regions are closer. Consequently, a dispatching algorithm that ignores drivers' behavior patterns may disrupt the overall traffic system.

To address the all above challenges, we propose a **G**PT-**A**ugmented **R**einforcement **L**earning with **I**ntelligent **C**ontrol framework, **GARLIC**, which utilizes an improved MARL approach. Specifically, we design a hierarchical traffic state representation module to integrate traffic features at different granularities, providing a comprehensive representation of real-time traffic conditions. Additionally, we quantify driving behavior through dynamic rewards using a contrastive learning method, aligning dispatching instructions with the intents of drivers. Given the complex analytical and understanding capabilities required for learning vehicle dispatching policies, we employ a Generative Pretrained Transformer (GPT)-augmented model with a self-defined loss function to enhance the expression of the framework. To the best of our knowledge, our innovative framework offers a comprehensive solution to the core challenges in vehicle dispatching, setting a new benchmark in this field. Our main contributions can be summarized as follows:

- Our proposed framework, GARLIC, combines hierarchical traffic state representation, dynamic reward generation, and GPT-augmented dispatching policy learning. To the best of our knowledge, this novel approach builds a complete GPT-enhanced MARL vehicle dispatching framework that has not been explored previously;
- We utilize multiview graphs to depict the hierarchical traffic states in the road networks and establish a dynamic reward model for capturing driving behaviors, leading to better dispatching policy outcomes. These innovations contribute significantly to the improved performance of vehicle dispatching;
- Extensive experiments on two real-world road networks against advancing baselines demonstrate the effectiveness and efficiency of GARLIC.

2 Related Work

This section provides a concise overview of related research in vehicle dispatching. Unlike car-hailing order dispatching, vehicle dispatching focuses on relocating vehicles to ensure a future balance between supply and demand. Many previous studies have modeled this as a Markov decision process, which relies on explicitly fitted state transition probabilities (Zhang et al. 2024, 2023a; Sun et al. 2024). To efficiently model this Markov decision process, RL has been widely applied to vehicle repositioning tasks (Chen et al. 2024; Qin, Zhu, and Ye 2022), where the global traffic state and reward function are used to enhance the precision of repositioning. However, the global traffic-state perception in existing methods has high communication latency, hindering real-time dispatch (Shi et al. 2024b). To address this, we designed a multiview graph learning module with limited hops.

Furthermore, driver behavior plays a crucial role in transportation analysis (Cui et al. 2024; Zhang et al. 2023b; Ma et al. 2023; Han et al. 2023a). Recent studies have begun to incorporate driving behavior into driving applications. For example, Li *et al.* (Li et al. 2022) used IL method to replicate human driving behavior, effectively transferring these strategies to autonomous vehicle scenarios. Jackson *et al.* (Jackson, Jesus Saenz, and Ivanov 2024) uses the powerful analysis and processing capabilities of LLAMA-7B to characterize driving behavior and then uses it for autonomous driving simulation. However, there is a relative scarcity of research that quantifies vehicle driving behavior to directly evaluate the rationality of vehicle dispatching orders. Consequently, there is an urgent need to design a more efficient and accurate driving behavior-based vehicle dispatching system.

3 Preliminary

In this paper, we adopt a novel MARL method to optimize online car-hailing dispatching policies. This section outlines critical definitions for understanding our paper.

Vehicle Trajectory τ: This refers to a sequence of GPS points (x_t, y_t) recorded over a time interval $t \in [T]$, represented as $\tau = (x_1, y_1, t_1), \cdots, (x_T, y_T, t_T)$. A vehicle can generate multiple trajectories based on different statuses (such as empty or occupied). We focus solely on empty vehicle trajectories to better understand driving behavior when drivers don't have a specific destination.

Multiview Graph $\mathcal{G}^i$: We define the multiview graph as $\mathcal{G}^i = \{\boldsymbol{V}^i, \boldsymbol{E}^i\}$, where $i \in \{\text{micro,meso,macro}\}$ presents different views, the node set $\boldsymbol{V}^i$ represents various traffic zones, and the edge set $\boldsymbol{E}^i$ indicates the connections among these zones. The features of each traffic zone at time t are denoted by $\boldsymbol{X}_t^i \in \mathbb{R}^{|\boldsymbol{V}| \times m^i}$, capturing vehicle availability and order demand. For different views of graphs, we have different graph features: $m^{\text{micro}} \neq m^{\text{meso}} \neq m^{\text{macro}}$.

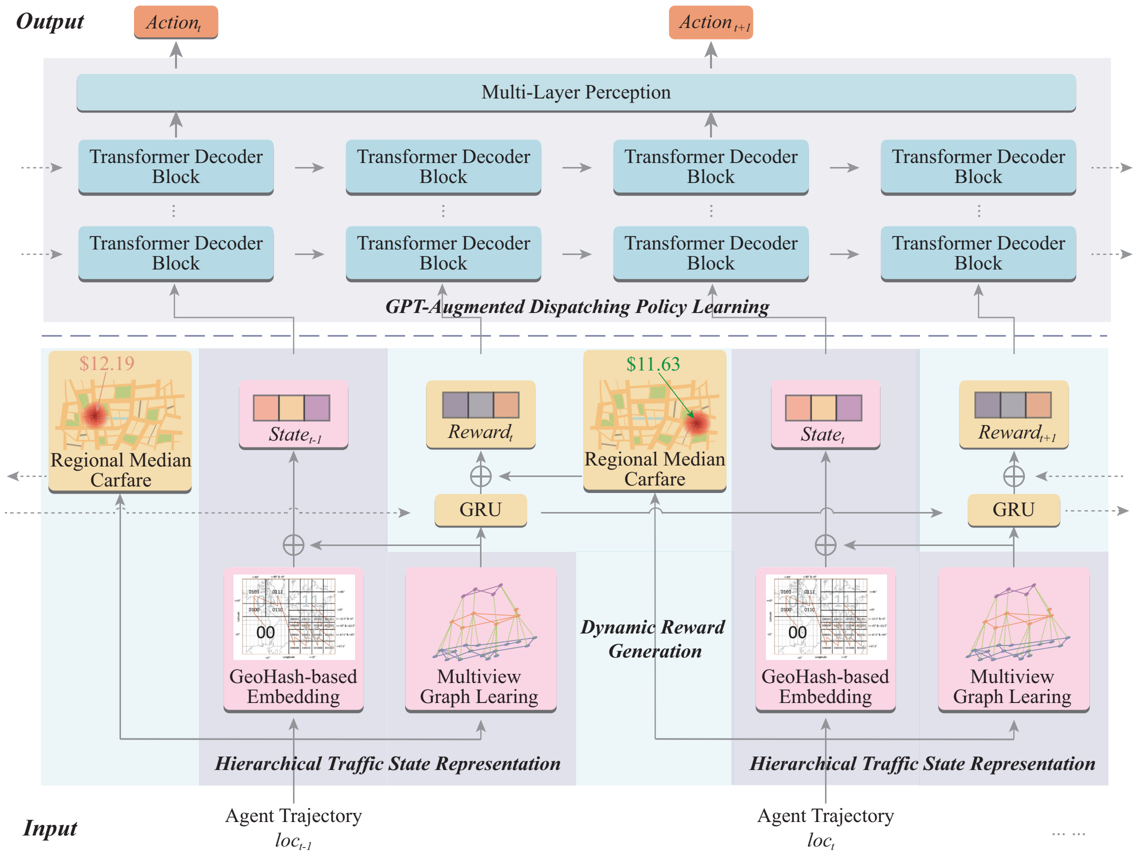


Figure 2: The framework overview of GARLIC.

Multi-Agent Reinforcement Learning for Vehicle Dispatching: In our model, each vehicle in the road network acts as an independent agent, with distinct driving behaviors and the ability to generate continuous trajectories and monitor local traffic conditions. For each agent (vehicle) u, we consider the following five essential elements:

- **Decision Time** $[T]$: It is a set of all finite decision timesteps $[T] = \{1, \cdots, t, \cdots, T\}$. At each timestep t, the vehicle location and environment states are sampled.
- **Action** $\mathcal{A}^u$: $\mathcal{A}^u = \{a_0^u, \cdots, a_t^u, \cdots, a_T^u\}$ represent the set of actions to balance the vehicle supply and demand. $a_t^u := \{dis, deg\}$ is an action performed by the vehicle u at time t, where dis is the straight-line distance a vehicle needs to travel from time t to $t+1$, and deg means the azimuth angle between the target and current locations.
- **State** $\mathcal{S}^u$: $\mathcal{S}^u = \{\boldsymbol{S}_0^u, \cdots, \boldsymbol{S}_t^u, \cdots, \boldsymbol{S}_T^u\}$ represents the set of traffic states observed at each time t. Here $\boldsymbol{S}_t^u$ is the concatenation of the state embedding matrix $\boldsymbol{Emb}_{\mathcal{G},t}$ extracted from the traffic environment and the location embedding matrix $\boldsymbol{Emb}_{loc,t}^u$ of the vehicle at time t.
- **Reward** $\mathcal{R}^u$: $\mathcal{R}^u = \{r_0^u, \cdots, r_t^u, \cdots, r_T^u\}$ represents the set of rewards calculated by the reward function, and it is predefined according to the driver's driving behavior and the taxi fare. The total return is defined as $\sum_t \gamma \cdot r_t^u$, where γ is a discount factor, $\gamma \in [0, 1]$.
- **Policy** π_θ^u: $\pi_\theta^u = \pi_\theta^u(a|s)$ is a mapping from traffic states to dispatching actions of the u-th agent. The policy π determines the appropriate vehicle dispatch instructions a by analyzing the state s, which includes various features of the environment and the current status of the agent.

While agents in the same area and close to each other may share the same multiview graphs $\mathcal{G}^i$ of the road network and observe similar traffic features $\boldsymbol{X}_t$, they exhibit unique driving behaviors that significantly influence their vehicle trajectories. To account for these behavioral differences, our study departs from conventional MARL frameworks with fixed rewards by employing a dynamic reward model. Additionally, we propose a GPT-augmented MARL model to learn more effective dispatching policies.

4 The Proposed Framework

In this section, we first provide a framework overview of GARLIC. Then we introduce the hierarchical traffic state representation method to capture the real-time traffic states. Furthermore, we demonstrate a dynamic driving reward generation approach to score vehicle trajectories under different driving behaviors. Finally, a GPT-augmented dispatching policy learning model is applied to combine all of the components and learn the vehicle dispatching policy.

Overview

Figure 2 provides an illustration of the overall vehicle dispatching framework, which is composed of three key modules: the hierarchical traffic state representation module, the dynamic reward generation module, and the GPT-augmented dispatching policy learning module.

In the first module, we employ a multiview Graph Convolutional Network (GCN) to represent the hierarchical traffic state by integrating traffic information gathered by various vehicles at different levels of granularity. By combining this with GeoHash-based vehicle location embeddings, we can accurately calculate the real-time traffic state of the specific region where each vehicle is located.

The second module utilizes a Gated Recurrent Unit (GRU)-based Recurrent Neural Network (RNN) to model driving behaviors, generating dynamic rewards that are weighted by the regional median carfare. This approach ensures that the reward system reflects both the temporal and spatial nuances of driver behavior.

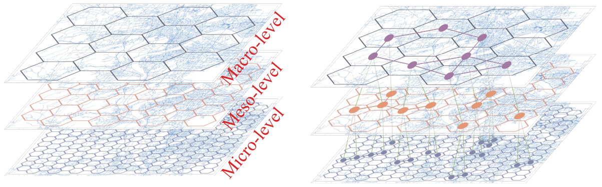
Finally, in the third module, we frame the training of the MARL-based vehicle dispatching task as a supervised learning process (Wang et al. 2024b; Yamagata, Khalil, and Santos-Rodriguez 2023). For each agent, the time-ordered states and rewards are utilized as inputs, and a GPT-augmented model is employed to produce high-precision actions for vehicle dispatching.

Hierarchical Traffic State Representation

Urban spatiotemporal data exhibits hierarchical characteristics (Ning et al. 2024; Zhang et al. 2023c; Han et al. 2020), which cannot be directly represented using a single structured data format. For instance, features such as turning movements at a crossroad can only be captured from a micro-level view of the traffic environment, whereas the average travel time is a feature observable only from a macro-level perspective of the same environment. These features differ in sampling frequencies, dimensions, and units, necessitating specialized approaches to represent and integrate them accurately.

To address this issue, we present the road network as multiview honeycomb graphs, as shown in Figure 3(a). In this representation, the road network is divided into grids comprising square hexagons of varying radii, each representing a distinct view. Here, the hexagon-based grids ensure uniform distance from all adjacent neighbors to the central grid, facilitating more precise modeling of different spatial regions than square grid-based methods. To construct the multiview graph, each grid is treated as a node, and the traffic information in a grid is considered to be the node feature, with edges connecting adjacent grids, as shown in Figure 3(b).

Unlike other road network modeling methods, we calculate distinct traffic indicators for different views of graphs. The micro-level graph primarily utilizes vehicle trajectory, road congestion status, and vehicle speed—data that can be directly obtained from the local environment ($radii \leq 1$km) of a vehicle. The meso-level graph considers factors such as traffic volume, average traffic speed, intersection performance, and parking availability, which require analysis of all vehicles passing through a set of certain traffic sections[1] (1km $< radii <$ 5km). Meanwhile, the macro-level graph includes features such as average travel time, road network connectivity, and overall traffic conditions, which necessitate a more comprehensive analysis of vehicles across a broader range ($radii \geq 5$km) of the road network.



(a) The multi-level road network modeling (b) The multiview graph construction

Figure 3: The multiview graph of road networks.

To extract refined the traffic embeddings $\boldsymbol{Emb}^u_{\mathcal{G}^u_t}$, a GCN-based model is then deployed for multiview graph representation for a vehicle u:

$$\begin{aligned}\boldsymbol{Emb}^u_{\mathcal{G}^u_t} &= \text{Concat}\left(\boldsymbol{Emb}^u_{\mathcal{G}^{u,i}_t}\right),\\ \boldsymbol{Emb}^u_{\mathcal{G}^{u,i}_t} &= \text{GCN}(\boldsymbol{A}^{u,i}, \boldsymbol{X}^{u,i}_t) = \boldsymbol{A}^{u,i}\boldsymbol{X}^{u,i}_t\boldsymbol{W}^i,\end{aligned} \tag{1}$$

where $i \in \{\text{micro}, \text{meso}, \text{macro}\}$, $\boldsymbol{W}^i$ is the weight matrix that need to be trained, $\boldsymbol{A}^{u,i}$ is the adjacency matrix of the graph $\mathcal{G}^{u,i}$ under a specific view i, and $\boldsymbol{X}^{u,i}_t$ is the traffic features related to the graph $\mathcal{G}^{u,i}$ at time t.

Note that a high-accuracy location embedding of real-time trajectories is essential for this task. We first use GeoHash (Morton 1966) to encode each real-time GPS point, based on latitude and longitude, in the trajectory:

$$\boldsymbol{Emb}^u_{loc^u_t} = \text{GeoHash}\left(lat^u_t, lon^u_t\right), \tag{2}$$

where $t \in \{0, 1, \cdots, T\}$ presents a specific timestep, and $loc^u_t := (lat^u_t, lon^u_t)$ is the real-time GPS point of vehicle u.

By combining this location embedding with the traffic state surrounding vehicle u at time t, we obtain the overall state embeddings of vehicle u:

$$s^u_t = \text{Concat}(\boldsymbol{Emb}^u_{\mathcal{G}^u_t}, \boldsymbol{Emb}^u_{loc^u_t}). \tag{3}$$

Dynamic Reward Generation

The effective implementation of vehicle dispatching in real-world scenarios is largely influenced by driving behavior. However, early studies often ignored the quantification of driving behavior, focusing instead on minimizing vehicle imbalance or maximizing benefits in dispatching optimization (Wagenmaker and Pacchiano 2023). In this section, we propose a dynamic reward generation method that incorporates both driving behaviors and anticipated income. It quantifies the likelihood of drivers adhering to their driving habits by analyzing the vehicle trajectories in real-time.

[1] The "traffic section" refers to a specific segment of a road or highway between two points, often delineated by intersections, junctions, or other distinct markers.

To accurately capture the relationship between driving trajectories and corresponding driving behaviors in traffic embeddings, we deploy a GRU-based RNN model. This network calculates the probability p_t^u whether a trajectory belongs to a given vehicle u.

$$\begin{aligned} z_t^u =& \sigma\left(\boldsymbol{W}_{zx}\boldsymbol{Emb}^u_{loc_t^u} + \boldsymbol{W}_{zp}h^u_{t-1} + \boldsymbol{b}_z\right), \\ y_t^u =& \sigma\left(\boldsymbol{W}_{yx}\boldsymbol{Emb}^u_{loc_t^u} + \boldsymbol{W}_{yp}h^u_{t-1} + \boldsymbol{b}_y\right), \\ p_t^{u\prime} =& \tanh\left(\boldsymbol{W}'_x\boldsymbol{Emb}^u_{loc_t^u} + y_t^u \odot \boldsymbol{W}'_p p^u_{t-1} + \boldsymbol{b}'\right), \\ p_t^u =& z_t^u \odot p^u_{t-1} + (1 - z_t^u) \odot p_t^{u\prime}, \end{aligned} \tag{4}$$

where $t \in \{1, 2, \cdots, T\}$, $h_0 := \boldsymbol{Emb}^u_{loc_0}$ is the location embedding at initial timestep ($t = 0$), $\boldsymbol{W}_{\text{GRU}} = \left\{\boldsymbol{W}_{zx}, \boldsymbol{W}_{zp}, \boldsymbol{W}_{yx}, \boldsymbol{W}_{yp}, \boldsymbol{W}'_x, \boldsymbol{W}'_p\right\}$ is the set of weight matrices of GRU, and $\boldsymbol{b}_{\text{GRU}} = \left\{\boldsymbol{b}_z, \boldsymbol{b}_y, \boldsymbol{b}'\right\}$ is the set of bias.

We employ a contrastive learning method to optimize the model parameters. Since different drivers exhibit distinct driving behaviors, trajectories generated by other vehicles are used as negative samples when modeling a specific driver's behavior, as illustrated in Equation (5).

$$Loss_{\text{pre-training}} = \max \sum_{u\in[N]} \sum_{t\in[T]} q_t^u \log p_t^u, \tag{5}$$

where N is the total number of online car-hailing vehicles, $q_t^u \in \{0, 1\}$ is the ground truth of the GPS point generated by the vehicle u at time t.

Additionally, when a vehicle is carrying passengers, the regional median carfare earned by a driver is another factor influencing vehicle dispatching. Therefore, we introduce the dynamic reward function, which incorporates both factors by introducing a hyperparameter α to weigh them together.

$$r_t^u = \alpha \cdot p_t^u + (1 - \alpha) \cdot \sigma(\boldsymbol{W}_{\text{fare}}\hat{x}_{\text{fare},t}), \tag{6}$$

where $\sigma(\cdot)$ is the sigmoid activation function, $\hat{x}_{\text{fare},t} = \sum_{t'=t}^{T} x_{\text{fare},t'}$, and $\boldsymbol{W}_{\text{fare}}$ is the weight matrix.

GPT-Augmented Dispatching Policy Learning

As discussed in the framework overview, vehicle dispatching can be effectively modeled as a MARL problem, which can also be reformulated as a supervised learning task. In this context, states and rewards are treated as sequential input data, with the corresponding sequence of actions as the output data. However, the complexity of transportation systems, which requires the analysis of intricate traffic states and driving behaviors, demands advanced reasoning capabilities. Recently, the GPT model has demonstrated strong performance in handling long-sequence, context-dependent, and structured data. Therefore, we utilize a GPT-augmented model to address these challenges.

Note that the core part of a GPT model is the transformer structure. The input of the transformer is a sequence of temporal data, and we assign different positional embeddings to the data at different timesteps. At each timestep, we mainly use two deep transformer decoder blocks to extract the probability of the next action. We use the expected reward at the current time step as the input of the first transformer decoder block, and get the output embedding to guide the subsequent transformer decoder block to calculate the result:

$$\begin{aligned} \boldsymbol{Emb}_{r_t} &= \text{Decoder}_{\text{T}}(\boldsymbol{P}_{a_{t-1}}, r_t, t), \\ \boldsymbol{P}_{a_t} &= \text{Decoder}_{\text{T}}(\boldsymbol{Emb}_{r_t}, s_t, t), \end{aligned} \tag{7}$$

where t starts from 1, and $\boldsymbol{P}_{a_0} = \boldsymbol{0}$ is initialized as the zero tensor at $t = 0$. $\text{Decoder}_{\text{T}}(x, y, z) = \text{Decoder}_{\text{T}}^{k} \odot \text{Decoder}_{\text{T}}^{k-1} \odot \cdots \odot \text{Decoder}_{\text{T}}^{(1)}(x, y, z)$ is a k-layer deep neural network of the transformer decoders. For each layer $\text{Decoder}_{\text{T}}^{(l)}(x, y, t) = \text{Attention}(x + \boldsymbol{Emb}_{pos}(t), x + \boldsymbol{Emb}_{pos}(t), y + \boldsymbol{Emb}_{pos}(t))$, we add the same positional embedding $\boldsymbol{Emb}_{pos}(t)$ to each input x and y. Here $\text{Attention}(x, y, z) = \text{softmax}(\frac{\sigma(x\boldsymbol{W}_x)\sigma(y\boldsymbol{W}_y)^\top}{\sqrt{d_y}})\sigma(z\boldsymbol{W}_z)$, where $\sigma(\cdot)$ is the GeLU activation function.

Finally, we use a Multi-Layer Perceptron (MLP) mapping the action probability tensor $\boldsymbol{P}_{a_t}$ to a unique result a'_t in the closed action set $\mathcal{A}$ as the action that a vehicle needs to perform in the current step:

$$a'_t = [a_t^{(1)\prime}, a_t^{(2)\prime}] = \text{MLP}_{\boldsymbol{W}_a}(\boldsymbol{P}_{a_t}), \tag{8}$$

where $a_t^{(1)\prime}$ is the normalized distance from the current location, $a_t^{(2)\prime} \in [0°, 360°]$ is the direction that a vehicle headed to, and both of $a_t^{(1)\prime}$ and $a_t^{(2)\prime}$ make up the unique action that controls this vehicle, $\boldsymbol{W}_a$ is the training parameters.

Note that the difference between 359° and 1° is only 2 degrees when measuring angles. Most common loss functions (*e.g.,* MAE Loss and MSE Loss) cannot describe this phenomenon well. To train our framework GARLIC effectively, we proposed a novel loss function, named Geospatial Loss (GeoLoss), to minimize the geospatial difference between the predicted action and the ground truth for this task, and our training target is to minimize the GeoLoss that we defined below:

$$\begin{aligned} \min Loss(a'_t, a_t) = \min \Big(& |a_t^{(2)\prime} - a_t^{(2)}|^2, \\ & (360° - |a_t^{(2)\prime} - a_t^{(2)}|)^2 \Big) + |a_t^{(1)\prime} - a_t^{(1)}|^2. \end{aligned} \tag{9}$$

5 Experiments

This section conducts extensive experiments using 2 real-world datasets to evaluate the effectiveness of GARLIC. We first introduce the experimental settings. Next, we compare GARLIC with representative baselines. Finally, the ablation study and a case study are introduced.

Experimental Settings

Dataset. We use two datasets with different scales for experiments: one is located in lower and midtown Manhattan, New York City, USA[2], and the other larger dataset is the taxi

[2] https://data.cityofnewyork.us/Transportation/2018-Yellow-Taxi-Trip-Data/t29m-gskq/about_data

trajectory data from the core area of Hangzhou, Zhejiang Province, CHN[3]. More details can be found in Appendix A.

Metrics. We use the Euclidean distance metric, *Error*, to assess the discrepancy between predicted actions and the driver's actual driving intentions. Additionally, the *empty-loaded rate* metric is employed to measure the efficiency of the car-hailing service, which is another widely used metric in transportation systems (Cao, Wang, and Li 2021).

Baselines. We compare GARLIC with baselines from two different categories: (1) Online RL methods: MT (Robbennolt and Levin 2023) and FTPEDEL (Wagenmaker and Pacchiano 2023); (2) Offline RL methods: CQL (Kumar et al. 2020), TD3+BC (Fujimoto and Gu 2021), Decistion Transformer (DT) (Chen et al. 2021), RLPD (Ball et al. 2023), latent-ORL (Hong, Levine, and Dragan 2024), and SS-DT (Zheng et al. 2023); (3) traditional vehicle dispatching systems: DGS (Cheng, Jha, and Rajendram 2018) and A-RTRS (Riley, Hentenryck, and Yuan 2020). More details about these methods can be found in Appendix C.

Implementation Detail

To avoid network congestion, we only allow V2V communications between vehicles in adjacent regions. In addition, we limit the waiting time for vehicles to broadcast and receive V2V multi-hop messages across different regions to 1 second, ignoring any timed-out transmissions. The implementation details can be found in Appendix D.

Overall Performance

The performance of all the baselines in both two datasets is shown in Table 1, in terms of the two metrics we introduced before, *i.e., Error* and *empty-loaded rate*. We use **M** to present the Manhattan dataset and use **H** to stand for the Hangzhou dataset. The performance of all methods is the average result of the last 100 epochs in a total of 1000 runs.

We can see that GARLIC significantly reduces the error compared to other traditional vehicle dispatching systems, primarily due to our adoption of a more effective loss function for guiding the model during back-propagation and training. Unlike online reinforcement learning methods, nearly all offline RL approaches, including ours, outperform the online method by better utilizing offline data for effective training. Additionally, conventional offline RL methods (*e.g.,* TD3+BC, CQL, and RLPD) perform poorly on the larger Hangzhou dataset due to their inability to gather comprehensive traffic information within acceptable transmission delays. Although DT, latent-ORL, and SS-DT use similar stacked transformer decoder layers as our framework, they do not model driving behavior in scheduling tasks, limiting their accuracy.

When comparing the metric of empty-loaded rate in Table 1, our method ranks among the best ones. However, our model needs to weigh the driver's personal driving behavior habits. Therefore, an area with a slightly longer route that is more familiar to the driver has more chance of being selected for vehicle dispatching. This caused the empty-loaded rate of GARLIC to be slightly higher than the SS-DT method on the Manhattan dataset. However, when the scale of the offline dataset becomes larger (Hangzhou dataset), our method has a stronger ability to find the optimal dispatching strategy and achieve the best performance while satisfying the driver's driving behavior.

	Error (km)		Empty-loaded rate (%)	
	M	**H**	**M**	**H**
MT	0.3517	0.3929	38.23	47.52
FTPEDEL	0.3307	0.3451	36.03	42.74
TD3+BC	0.3371	0.3243	37.22	50.13
CQL	0.3125	0.2368	35.17	46.87
DT	0.3051	0.2573	33.49	41.45
RLPD	0.3213	0.3004	36.24	48.21
Latent-ORL	0.3117	0.2086	34.37	45.22
SS-DT	0.3048	0.1843	**32.25***	40.99
DGS	0.4125	0.1982	32.57	41.23
A-RTRS	0.3567	0.1957	32.39	41.04
GARLIC	**0.3044***	**0.1582***	32.38	**40.71***

"*" indicates the statistically significant improvements (*i.e.,* two-sided t-test with $p < 0.05$) over the best baseline. For all metrics: the lower, the better.

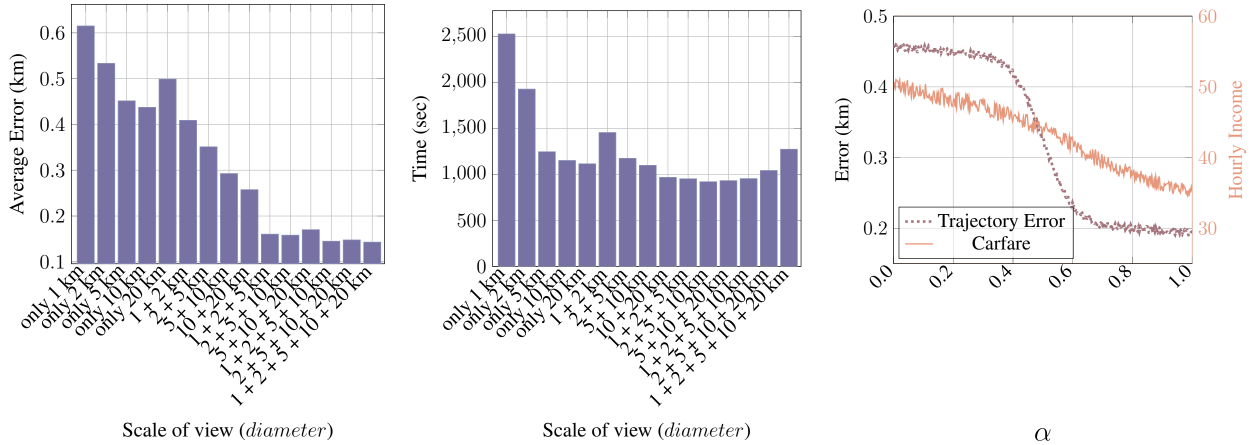
Table 1: Experimental results of different baselines.

Ablation Study

The effectiveness of multiview graph. To better understand the role of multiview graph learning in GARLIC, we divide the road network into regions with diameters of 1 km and 2 km (micro-level), 5 km (meso-level), and 10 km and 20 km (macro-level). We then extract 5 graphs with varying traffic features based on these granularities. Under the same 1-second V2V data transmission delay previously mentioned, we sequentially use various combinations of these graphs as inputs to conduct experiments. The results are presented in Figure 4(a). It indicates that the error in graph learning using a single view is significantly higher than that of multiview graph learning methods. Moreover, when comparing different scales, it is evident that model performance improves as the granularity of the scale decreases. To further explore the relationship between computational latency (including V2V communication and model training time) and multiview graphs, we compared the training time of each model when achieving a scheduling error of 0.7 km, as shown in Figure 4(b). It also verifies that the multiview graph-based learning method could be more efficient than the single-view graph-based learning method. In addition, by analyzing the average error in Figure 4(a) alongside the time cost in Figure 4(b), we selected three multiview graphs with diameters of 2 + 5 + 10 km to model the traffic of road networks efficiently.

The influence of driving behavior. To verify the influence of driving behavior, we conduct experiments on this method alone via setting different hyperparameter α defined in Equation (6). The experimental result is shown in Figure 4(c). When the weight of driving behavior increases, the error of the predicted vehicle trajectory decreases. At this time,

[3] Prviate dataset. To protect data copyright, we will share the full dataset through academic collaboration only.



(a) Error under 1-sec data transmission (b) Results under different settings of α (c) Results for achieving sub-0.7 km error

Figure 4: Different settings of hyperparamters.

the vehicle follows the path given by the offline data and cannot explore the path that can generate higher income in accordance with the vehicle's driving behavior. Conversely, when α is close to 0, many passenger loading locations seriously deviate from the roads and regions familiar to the driver although the vehicle can receive more orders. These potential issues are more likely to cause traffic accidents. From Figure 4(c) we can see that the trajectory error significantly drops when α is between 0.4-0.7. Therefore, in this paper, we set $\alpha = 0.67$ to let the dispatch strategy increase the driver's income as much as possible while satisfying each driver's driving behavior.

Case Study

We randomly choose an online car-hailing car in Hangzhou as our experimental object. We simulated the vehicle dispatching routes using different methods to compare with the ground truth (Origin), as shown in Figure 5. We selected a local area in Hangzhou for visualization. Different shades of red in honeycomb grids indicate the length of time the vehicle stayed in one week of history. When the area has no color, the vehicle has not been to this area that week. It represents the driver's personalized driving behavior. The vehicle is currently located in the bright blue grid, and we visualize 4 dark blue areas with ride-hailing demand in the next 15 minutes. We use arrows of different colors to indicate the calculation results of different models.

As can be seen from Figure 5, Order 1 is farther from the departure point of the vehicle compared to Orders 2, 3, and 4. Since Order 4 is in the city center, most methods select this area as the vehicle pick-up point. However, there is a direct arterial road between Order 2 and the vehicle's departure location, so some methods choose to dispatch the vehicle to where Order 2 is located. Our method analyzes the driver's driving behavior and finds that the most suitable place to pick up passengers is the area where order 1 is located. Meanwhile, only the results calculated by our method are consistent with the actual vehicle trajectory, which shows the effectiveness of our proposed method.

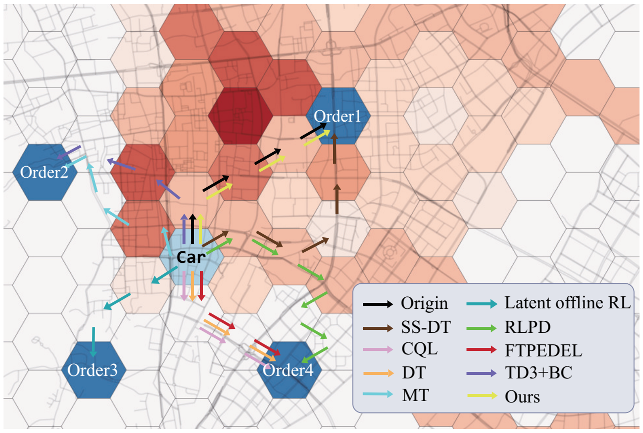


Figure 5: An example of vehicle dispatching.

6 Conclusion

In this paper, a novel framework called GARLIC is proposed to address the problem of vehicle dispatching while considering the driving behavior of drivers at the same time. Specifically, it can be divided into three modules, *i.e.,* the hierarchical traffic state representation module for traffic state extraction, the dynamic reward generation module for driving behavior as well as carfare analysis, and the GPT-augmented dispatching policy learning module for balancing vehicle supply and passenger demand. The model achieves a response in seconds under multiple real datasets and has excellent performance. In the future, we hope to combine the Kafuka engine and cloud-edge collaboration technologies to further optimize the information transmission of each node in vehicle dispatching, achieve a quick response of hundreds of milliseconds, and improve the driver's order acceptance and user's riding experience.

Acknowledgments

This research was partially supported by Research Impact Fund (No.R1015-23), APRC - CityU New Research Initiatives (No.9610565, Start-up Grant for New Faculty of CityU), CityU - HKIDS Early Career Research Grant (No.9360163), Hong Kong ITC Innovation and Technology Fund Midstream Research Programme for Universities Project (No.ITS/034/22MS), Hong Kong Environmental and Conservation Fund (No. 88/2022), and SIRG - CityU Strategic Interdisciplinary Research Grant (No.7020046), Huawei (Huawei Innovation Research Program), Tencent (CCF-Tencent Open Fund, Tencent Rhino-Bird Focused Research Program), Ant Group (CCF-Ant Research Fund, Ant Group Research Fund), Alibaba (CCF-Alimama Tech Kangaroo Fund No. 2024002), Collaborative Research Fund No.C1043-24GF, CCF-BaiChuan-Ebtech Foundation Model Fund, and Kuaishou.

References

Ball, P. J.; Smith, L.; Kostrikov, I.; and Levine, S. 2023. Efficient online reinforcement learning with offline data. In *International Conference on Machine Learning*, 1577–1594. PMLR.

Barrios, J. M.; Hochberg, Y. V.; and Yi, H. 2023. The cost of convenience: Ridehailing and traffic fatalities. *Journal of Operations Management*, 69(5): 823–855.

Cao, Y.; Wang, S.; and Li, J. 2021. The optimization model of ride-sharing route for ride hailing considering both system optimization and user fairness. *Sustainability*, 13(2): 902.

Chen, H.; Sun, P.; Song, Q.; Wang, W.; Wu, W.; Zhang, W.; Gao, G.; and Lyu, Y. 2024. i-Rebalance: Personalized Vehicle Repositioning for Supply Demand Balance. In Wooldridge, M. J.; Dy, J. G.; and Natarajan, S., eds., *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, 46–54. AAAI Press.

Chen, L.; Lu, K.; Rajeswaran, A.; Lee, K.; Grover, A.; Laskin, M.; Abbeel, P.; Srinivas, A.; and Mordatch, I. 2021. Decision transformer: Reinforcement learning via sequence modeling. *Advances in neural information processing systems*, 34: 15084–15097.

Cheng, S.; Jha, S. S.; and Rajendram, R. 2018. Taxis Strike Back: A Field Trial of the Driver Guidance System. In André, E.; Koenig, S.; Dastani, M.; and Sukthankar, G., eds., *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS 2018, Stockholm, Sweden, July 10-15, 2018*, 577–584. International Foundation for Autonomous Agents and Multiagent Systems Richland, SC, USA / ACM.

Cui, C.; Ma, Y.; Cao, X.; Ye, W.; Zhou, Y.; Liang, K.; Chen, J.; Lu, J.; Yang, Z.; Liao, K.-D.; et al. 2024. A survey on multimodal large language models for autonomous driving. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 958–979.

Ellis, B.; Cook, J.; Moalla, S.; Samvelyan, M.; Sun, M.; Mahajan, A.; Foerster, J.; and Whiteson, S. 2024. Smacv2: An improved benchmark for cooperative multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 36.

Fujimoto, S.; and Gu, S. S. 2021. A minimalist approach to offline reinforcement learning. *Advances in neural information processing systems*, 34: 20132–20145.

Gerlough, D. L.; and Huber, M. J. 1976. Traffic flow theory. Technical report.

Guo, Y.; Li, W.; Xiao, L.; Choudhary, A.; and Allaoui, H. 2024. Enhancing efficiency and interpretability: A multi-objective dispatching strategy for autonomous service vehicles in ride-hailing. *Computers & Industrial Engineering*, 110385.

Han, X.; Shen, G.; Yang, X.; and Kong, X. 2020. Congestion recognition for hybrid urban road systems via digraph convolutional network. *Transportation Research Part C: Emerging Technologies*, 121: 102877.

Han, X.; Zhao, X.; Zhang, L.; and Wang, W. 2023a. Mitigating action hysteresis in traffic signal control with traffic predictive reinforcement learning. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 673–684.

Han, X.; Zhao, X.; Zhang, L.; and Wang, W. 2023b. Mitigating Action Hysteresis in Traffic Signal Control with Traffic Predictive Reinforcement Learning. In Singh, A. K.; Sun, Y.; Akoglu, L.; Gunopulos, D.; Yan, X.; Kumar, R.; Ozcan, F.; and Ye, J., eds., *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2023, Long Beach, CA, USA, August 6-10, 2023*, 673–684. ACM.

Han, X.; Zhou, D.; Shen, G.; Kong, X.; and Zhao, Y. 2024. Deep Trajectory Recovery Approach of Offline Vehicles in the Internet of Vehicles. *IEEE Trans. Veh. Technol.*, 73(11): 16051–16062.

Hong, J.; Levine, S.; and Dragan, A. 2024. Learning to influence human behavior with offline reinforcement learning. *Advances in Neural Information Processing Systems*, 36.

Huang, C.-M.; and Lin, J.-J. 2022. The k-hop V2V data offloading using the predicted utility-centric path switching (PUPS) method based on the SDN-controller inside the multi-access edge computing (MEC) architecture. *Vehicular Communications*, 36: 100496.

Huang, X.; Zhang, X.; Ling, J.; and Cheng, X. 2023. Effective credit assignment deep policy gradient multi-agent reinforcement learning for vehicle dispatch. *Applied Intelligence*, 53(20): 23457–23469.

Jackson, I.; Jesus Saenz, M.; and Ivanov, D. 2024. From natural language to simulations: applying AI to automate simulation modelling of logistics systems. *International Journal of Production Research*, 62(4): 1434–1457.

Kumar, A.; Zhou, A.; Tucker, G.; and Levine, S. 2020. Conservative q-learning for offline reinforcement learning.

Advances in Neural Information Processing Systems, 33: 1179–1191.

Li, G.; Ji, Z.; Li, S.; Luo, X.; and Qu, X. 2022. Driver behavioral cloning for route following in autonomous vehicles using task knowledge distillation. *IEEE Transactions on Intelligent Vehicles*, 8(2): 1025–1033.

Ma, Q.; Zhang, Z.; Zhao, X.; Li, H.; Zhao, H.; Wang, Y.; Liu, Z.; and Wang, W. 2023. Rethinking sensors modeling: Hierarchical information enhanced traffic forecasting. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, 1756–1765.

Morton, G. M. 1966. A computer oriented geodetic data base and a new technique in file sequencing.

Ning, Y.; Liu, H.; Wang, H.; Zeng, Z.; and Xiong, H. 2024. UUKG: unified urban knowledge graph dataset for urban spatiotemporal prediction. *Advances in Neural Information Processing Systems*, 36.

Qin, Z. T.; Zhu, H.; and Ye, J. 2022. Reinforcement learning for ridesharing: An extended survey. *Transportation Research Part C: Emerging Technologies*, 144: 103852.

Qiu, D.; Wang, Y.; Zhang, T.; Sun, M.; and Strbac, G. 2023. Hierarchical multi-agent reinforcement learning for repair crews dispatch control towards multi-energy microgrid resilience. *Applied Energy*, 336: 120826.

Rahman, M. M.; and Thill, J.-C. 2023. Impacts of connected and autonomous vehicles on urban transportation and environment: A comprehensive review. *Sustainable Cities and Society*, 96: 104649.

Riley, C.; Hentenryck, P. V.; and Yuan, E. 2020. Real-Time Dispatching of Large-Scale Ride-Sharing Systems: Integrating Optimization, Machine Learning, and Model Predictive Control. In Bessiere, C., ed., *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, 4417–4423. ijcai.org.

Robbennolt, J.; and Levin, M. W. 2023. Maximum Throughput Dispatch for Shared Autonomous Vehicles Including Vehicle Rebalancing. *IEEE Transactions on Intelligent Transportation Systems*.

Sadrani, M.; Tirachini, A.; and Antoniou, C. 2022. Vehicle dispatching plan for minimizing passenger waiting time in a corridor with buses of different sizes: Model formulation and solution approaches. *European Journal of Operational Research*, 299(1): 263–282.

Shi, B.; Xia, Y.; Xu, S.; and Luo, Y. 2024a. A vehicle value based ride-hailing order matching and dispatching algorithm. *Engineering Applications of Artificial Intelligence*, 132: 107954.

Shi, W.; Jiang, H.; Xiong, B.; Chen, X.; Zhang, H.; Chen, Z.; and Wu, Q. 2024b. RIS-Empowered V2V Communications: Three-Dimensional Beam Domain Channel Modeling and Analysis. *IEEE Trans. Wirel. Commun.*, 23(11): 15844–15857.

Sun, G.; Boateng, G. O.; Liu, K.; Ayepah-Mensah, D.; and Liu, G. 2024. Dynamic Pricing for Vehicle Dispatching in Mobility-as-a-Service Market via Multi-Agent Deep Reinforcement Learning. *IEEE Transactions on Vehicular Technology*.

Wagenmaker, A.; and Pacchiano, A. 2023. Leveraging offline data in online reinforcement learning. In *International Conference on Machine Learning*, 35300–35338. PMLR.

Wang, Y. 2023. Optimizing V2V Unicast Communication Transmission with Reinforcement Learning and Vehicle Clustering. *arXiv preprint arXiv:2309.12052*.

Wang, Y.; Sun, H.; Lv, Y.; Chang, X.; and Wu, J. 2024a. Reinforcement learning-based order-dispatching optimization in the ride-sourcing service. *Computers & Industrial Engineering*, 192: 110221.

Wang, Y.; Yang, C.; Wen, Y.; Liu, Y.; and Qiao, Y. 2024b. Critic-guided decision transformer for offline reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 15706–15714.

Yamagata, T.; Khalil, A.; and Santos-Rodriguez, R. 2023. Q-learning decision transformer: Leveraging dynamic programming for conditional sequence modelling in offline rl. In *International Conference on Machine Learning*, 38989–39007. PMLR.

Zhang, L.; Yang, C.; Yan, Y.; Cai, Z.; and Hu, Y. 2024. Automated guided vehicle dispatching and routing integration via digital twin with deep reinforcement learning. *Journal of Manufacturing Systems*, 72: 492–503.

Zhang, X.; Xiong, G.; Ai, Y.; Liu, K.; and Chen, L. 2023a. Vehicle dynamic dispatching using curriculum-driven reinforcement learning. *Mechanical Systems and Signal Processing*, 204: 110698.

Zhang, Z.; Huang, Z.; Hu, Z.; Zhao, X.; Wang, W.; Liu, Z.; Zhang, J.; Qin, S. J.; and Zhao, H. 2023b. MLPST: MLP is All You Need for Spatio-Temporal Prediction. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, 3381–3390.

Zhang, Z.; Zhao, X.; Liu, Q.; Zhang, C.; Ma, Q.; Wang, W.; Zhao, H.; Wang, Y.; and Liu, Z. 2023c. Promptst: Prompt-enhanced spatio-temporal multi-attribute prediction. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, 3195–3205.

Zhang, Z.; Zhao, X.; Miao, H.; Zhang, C.; Zhao, H.; and Zhang, J. 2023d. Autostl: Automated spatio-temporal multitask learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 4902–4910.

Zheng, Q.; Henaff, M.; Amos, B.; and Grover, A. 2023. Semi-supervised offline reinforcement learning with action-free trajectories. In *International conference on machine learning*, 42339–42362. PMLR.