STAR: Spatio-Temporal Attention Rebalancing for Eco-Friendly Autonomous Mobility-on-Demand Systems

Jungeun Lee[1] , **Seungjae Baek**[2] , **Seongjae Lee**[1] , **Sunhwi Kim**[1] **and Jeong hwan Jeon**[1,2]
[1] Department of Electrical Engineering, UNIST
[2] Artificial Intelligence Graduate School, UNIST
{jungeunlee, bsj970, lsj1999, shkim308, jhjeon}@unist.ac.kr

Abstract

Existing Autonomous Mobility-on-Demand (AMoD) systems and demand-responsive algorithms are designed to adaptively respond to traffic demands, enhancing traffic service quality and profitability. However, such approaches may exacerbate traffic congestion by concentrating vehicles in high-demand areas. In addition, repeated braking, acceleration, and low-speed driving in traffic jams lead to substantial increases in emissions. Moreover, existing Deep Reinforcement Learning (DRL)-based approaches fail to address the instability of cooperative learning in environments with highly variable rewards, such as carbon emissions. To address these issues, we propose the Spatio-Temporal Attention Rebalancing (STAR) framework, leveraging an encoder-decoder architecture. In the proposed framework, we maximize the number of requests served and minimize CO_2 emissions, while ensuring high service quality by limiting maximum travel delays and waiting times for passengers. To evaluate the proposed framework, we conduct simulations in realistic urban traffic scenarios. Experimental results demonstrate that our framework consistently outperforms existing baselines. In a large-scale scenario with a fleet of one hundred vehicles, our approach improves the service rate by up to 10.1% and reduces CO_2 emissions per passenger by up to 69.6% compared to the baseline methods. The proposed framework and baselines are publicly available at https://github.com/2jungeuni/eco-friendly-fleet-rebalancing.

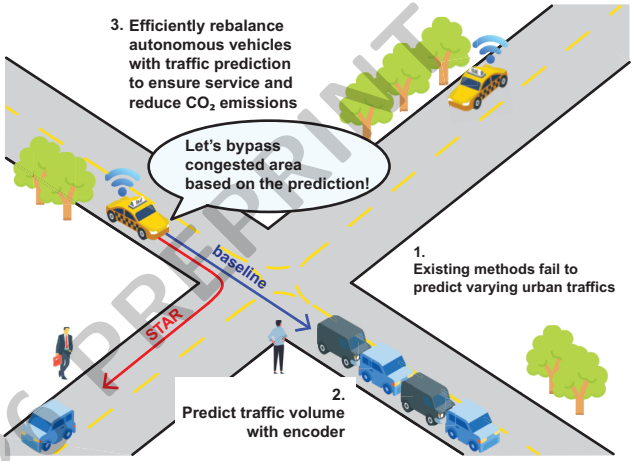


Figure 1: Overview of the proposed Spatio-Temporal Attention Rebalancing (STAR) framework. The proposed framework enables autonomous vehicles to proactively bypass predicted traffic bottlenecks while responding to nearby passenger requests. The policy serves a large number of passengers while effectively reducing CO_2 emissions.

1 Introduction

Traditional public transportation systems, which rely on fixed routes and schedules, face several limitations. Low ridership during off-peak hours not only reduces operational efficiency but also increases per-passenger carbon emissions. Furthermore, underserved areas remain poorly connected to public transit, reflecting the persistent First-Mile-Last-Mile (FMLM) problem. To address these issues, Autonomous Mobility-on-Demand (AMoD) systems have emerged as a next-generation solution that flexibly optimizes routes and schedules in real time. Today, advances in demand forecasting and large-scale optimization make citywide deployments possible, reducing waiting times and improving reliability [Alonso-Mora *et al.*, 2017]. However, capturing complex spatiotemporal demand dynamics over extended horizons remains computationally prohibitive in practice [Tsao *et al.*, 2019]. To overcome these limitations, optimization-based AMoD studies have focused on vehicle rebalancing strategies, relocating idle vehicles to predicted demand hotspots to reduce waiting times and travel delays [Jiao *et al.*, 2021; Haliem *et al.*, 2020; Qin *et al.*, 2021; Alonso-Mora *et al.*, 2017]. However, such approaches may worsen traffic congestion by concentrating AMoD fleets in high-demand regions, which in turn leads to higher CO_2 emissions [Fiedler *et al.*, 2017; Fagnant and Kockelman, 2015]. Frequent stops, prolonged idling, and low-speed driving in congested areas substantially increase fuel consumption and CO_2 emissions [Dhanorkar and Burtch, 2022].

As shown in Fig. 1, congestion-induced CO_2 emissions emerge from interactions among vehicles and background traffic. Moreover, these interactions evolve continuously over

time, making emissions neither fixed nor fully predictable within traditional optimization frameworks. These observations motivate the adoption of learning-based approaches for AMoD rebalancing. In particular, Multi-Agent Reinforcement Learning (MARL) provides a natural framework for modeling decentralized vehicle interactions. However, deploying MARL at the city scale introduces fundamental challenges. In urban AMoD systems, interactions among vehicles, nearby passenger requests, and predicted traffic congestion are spatially sparse and temporally dynamic. At any given time, only a small subset of vehicles meaningfully influences system behavior, and the structure of these interactions shifts rapidly as traffic conditions evolve. Under such circumstances, value-decomposition methods that rely on simple summation or averaging of individual value functions, such as Value Decomposition Networks (VDNs) [Sunehag *et al.*, 2017], struggle to assign credit effectively. More sophisticated methods such as QMIX [Rashid *et al.*, 2020] adopt monotonic mixing networks, but their weights depend only on the global state, failing to capture agent-specific context. Attention-based extensions such as QPLEX [Wang *et al.*, 2020] address agent-wise weighting but rely on dueling decomposition and do not explicitly handle dynamically varying agent sets. As a result, these methods struggle to distinguish which agents are truly responsible for system-level outcomes, including service profitability and CO_2 emissions. Moreover, they fail to adapt to dynamically varying agent sets, leading to severe credit assignment degradation in large-scale AMoD environments.

To tackle this challenge, we propose the Spatio-Temporal Attention Rebalancing (STAR) framework, a cooperative MARL approach for large-scale AMoD systems that jointly optimizes service profitability and CO_2 emissions. At the core of our framework is the STAR mixer, a state-dependent attention mechanism designed to address sparse and non-stationary spatiotemporal interactions by enabling structured credit assignment. In addition, the STAR framework adopts a feasibility-filtered control architecture. At each decision time step, a dispatching optimization module strictly enforces all hard service quality constraints, such as maximum waiting time and allowable travel delay. This design restricts policy learning to a feasible action space defined by these constraints. As a result, constraint satisfaction is guaranteed by construction, allowing the STAR agent to focus on optimizing long-term service performance and environmental sustainability.

We evaluate the STAR framework in a realistic urban simulation of New York City (NYC), focusing on For-Hire Vehicle (FHV) services such as Uber and Lyft. Real-world FHV trip data [Taxi and Commission, 2024] are integrated into the Simulation of Urban MObility (SUMO) environment, together with large-scale background traffic consisting of over $100,000$ Human-Driven Vehicles (HDVs). The high-fidelity setting enables a rigorous assessment of cooperative rebalancing under realistic congestion patterns and their environmental impacts.

The following is a summary of key contributions:

- **Eco-friendly cooperative AMoD framework.** We propose the STAR framework, which jointly optimizes service profitability and CO_2 emissions. By adopting a feasibility-filtered control architecture, STAR strictly enforces hard service-quality constraints while learning long-term sustainable rebalancing policies.
- **Spatio-temporal attention for robust credit assignment.** We devise the STAR mixer, a state-dependent attention mechanism that addresses sparse and non-stationary spatiotemporal interactions in urban traffic. Unlike existing value-decomposition methods, the STAR mixer leverages agent-specific observations for credit assignment. As each attention weight is computed independently per agent, the design is inherently fleet-size-agnostic, enabling stable cooperative learning across varying fleet scales.
- **Comprehensive evaluation under realistic traffic conditions.** We validate STAR in a high-fidelity SUMO environment with real-world NYC traffic data. Experimental results show that STAR consistently outperforms baselines.

	DeepPool '19	Guo '23	Chang '24	Sahebdel '25	STAR (Ours)
RL-based	✓	✓	×	×	✓
Ridesharing	✓	×	✓	✓	✓
Emission-aware	×	✓	✓	✓	✓
Congestion modeled	×	✓	✓	✓	✓
Large-scale	△	✓	△	△	✓

Table 1: Comparison of our framework to recent works. Except for DeepPool, methods are referred to by the last name of the first author. The symbol △ denotes evaluation in a large-scale environment without publicly available source code.

2 Related Works

Several recent studies [Al-Abbasi *et al.*, 2019; Guo *et al.*, 2023; Chang *et al.*, 2024; Sahebdel *et al.*, 2025] have incorporated traffic congestion or emissions-related objectives into transportation systems. However, as summarized in Table 1, these works differ substantially from our setting in both domain and problem formulation. Among existing methods, Guo *et al.* [2023] focus on traffic signal control, an infrastructure-level problem distinct from vehicle fleet control. Chang *et al.* [2024] and Sahebdel *et al.* [2025] estimate emissions through offline analysis, lacking the capability for real-time and adaptive fleet control. Therefore, these methods [Guo *et al.*, 2023; Chang *et al.*, 2024; Sahebdel *et al.*, 2025] are not directly applicable as baselines for online AMoD rebalancing.

Although DeepPool [Al-Abbasi *et al.*, 2019] does not explicitly model background traffic congestion or directly optimize CO_2 emissions, it remains the most relevant baseline for our problem setting. This is because it focuses on system-level operational efficiency and service profitability, which are closely related to vehicle utilization and, indirectly, to emission outcomes. To ensure a fair comparison under realistic traffic conditions, we reimplemented DeepPool within our SUMO environment. This allows us to isolate the effects of the cooperative coordination enabled by our framework and its congestion-aware credit-assignment mechanism.

t	Discrete decision time step, $t \in \mathbb{Z}_{\geq 0}$.
$\mathcal{V}_t$	Set of CAVs present at time t.
$\mathcal{R}_t$	Set of passenger requests active at time t.
r_i	Request time of passenger i.
$\mathcal{N}_t$	Set of nodes at time t, which includes the pickup and drop-off locations of passenger requests and the current location of CAVs.
$\mathcal{N}_t^p$	Pickup locations of active requests at time t, where $\mathcal{N}_t^p \subset \mathcal{N}_t$.
$\mathcal{N}_t^d$	Drop-off locations of active requests at time t, where $\mathcal{N}_t^d \subset \mathcal{N}_t$.
ℓ_i^p	Pickup location of request i, where $\ell_i^p \in \mathcal{N}_t^p$.
ℓ_i^d	Drop-off location of request i, where $\ell_i^d \in \mathcal{N}_t^d$.
ℓ_v^t	Current location of CAV v at time t, where $\ell_v^t \in \mathcal{N}_t$.
x_{nv}^t	Binary variable indicating whether node n is included in the planned route of CAV v at time t.
y_{nmv}^t	Binary decision variable indicating whether the path from node n to m is included in the overall scheduled route of CAV v at time t.
z_{iv}^t	Binary decision variable indicating whether passenger request i is assigned to CAV v at time t.
a_{nv}^t	Positive continuous decision variable representing the expected arrival time of CAV v at node n, as estimated at time t.
c_{nm}	Expected CO_2 emissions of CAV traveling from node n to m.
τ_v^t	Idle time of vehicle v at time t.
b_v^t	On-board passenger count of vehicle v at time t.
$\hat{\mathbf{h}}_t$	Predicted regional traffic volumes at time t.
$\mathbf{h}_t$	Observed regional traffic volumes at time t.
$\mathcal{S}_t$	Set of completed requests at time t.

Table 2: Notations. The notations are used to describe our system. Since all Cooperative Autonomous Vehicles (CAVs) are homogeneous, expected travel time and CO_2 emissions between any nodes are defined identically for all vehicles. Here, node denotes a vertex in the road network.

3 Problem Formulation

In this study, we consider a cooperative fleet rebalancing problem in which multiple vehicles make coordinated decisions to achieve shared system-level objectives. These vehicles are referred to as Cooperative Autonomous Vehicles (CAVs). They operate alongside Human-Driven Vehicles (HDVs), which generate background traffic and induce dynamic congestion. This section formalizes the system objectives and hard service-quality constraints. These constraints are not directly optimized by the learning agents but are instead enforced by a dispatching optimization module within the control pipeline. Under this formulation, the proposed STAR framework learns cooperative rebalancing policies within the resulting feasible decision space. All notations used throughout this section are summarized in Table 2.

3.1 Objectives

Our primary objectives are twofold: minimizing CO_2 emissions

$$\min_{y_{nmv}^t} \sum_{n\in\mathcal{N}_t} \sum_{m\in\mathcal{N}_t} \sum_{v\in\mathcal{V}_t} c_{nm} y_{nmv}^t \tag{1}$$

and serving the maximum number of passengers

$$\max_{z_{iv}^t} \sum_{i\in\mathcal{R}_t} \sum_{v\in\mathcal{V}_t} z_{iv}^t \tag{2}$$

at every decision time step t.

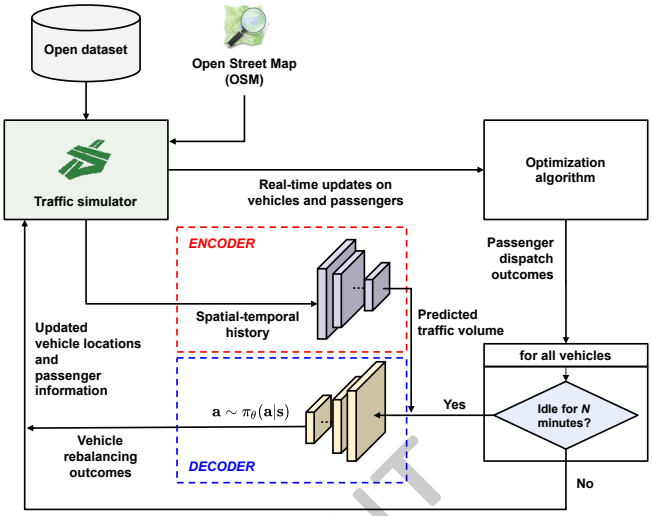


Figure 2: The Spatio-Temporal Attention Rebalancing (STAR) framework operating within a realistic urban driving environment. The flows of inputs and outputs among modules are presented.

3.2 Constraints

This section describes the key constraints considered critical within our system. First, passengers assigned at the previous decision time must remain unchanged until service completion:

$$z_{iv}^t = z_{iv}^{t-1}, \tag{3}$$

for all passengers not yet served and all vehicles. Here, a passenger is considered not yet served if the corresponding drop-off has not been completed. In addition, every vehicle has a maximum passenger capacity C:

$$\sum_{i\in\mathcal{R}_t} z_{iv}^t \leq C \quad ,\forall\, v \in \mathcal{V}_t. \tag{4}$$

All assigned passengers must be served within the maximum allowable waiting time W and travel delay T:

$$a_{l_i^p v}^t - r_i \leq W + M\left(1 - x_{l_i^p v}^t\right) \quad ,\forall\, i \in \mathcal{R}_t,\ v \in \mathcal{V}_t, \tag{5}$$

$$a_{l_i^d v}^t - a_{l_i^p v}^t \leq T + M\left(1 - x_{l_i^p v}^t\right) \quad ,\forall\, i \in \mathcal{R}_t,\ v \in \mathcal{V}_t, \tag{6}$$

where M is a sufficiently large constant. If a passenger request i is assigned to vehicle v, both its pickup and drop-off locations must be included in the vehicle schedule:

$$x_{l_i^p v}^t = z_{iv}^t, \quad x_{l_i^d v}^t = z_{iv}^t \quad ,\forall\, i \in \mathcal{R}_t,\ v \in \mathcal{V}_t. \tag{7}$$

In addition, pickup locations should be visited before drop-off locations:

$$a_{l_i^p v}^t \leq a_{l_i^d v}^t \quad ,\forall\, i \in \mathcal{R}_t,\ v \in \mathcal{V}_t. \tag{8}$$

4 Methodology

Building on the objectives and hard service-quality constraints defined in Section 3, we propose the Spatio-Temporal Attention Rebalancing (STAR) framework. An overview is illustrated in Fig. 2. The dispatching optimization module explicitly enforces all feasibility constraints, including maximum

Algorithm 1 STAR framework.

Initialize decision step $t \leftarrow 0$
while $t < E$ **do**
  $\mathcal{S}_t \leftarrow \text{DispatchingOptimization}(\mathcal{V}_t, \mathcal{R}_t, W, T)$
  if $t \geq N_h$ **then**
    $\hat{\mathbf{h}}_t \leftarrow \text{Encoder}\left(\mathbf{h}_{t-N_h:t-1}\right)$
    for each vehicle $v_i \in \mathcal{V}_t$ **do**
      if v_i is unmatched in $\mathcal{S}_t$ **and** $\tau_i^t > N$ **then**
        $\mathbf{a}_t^{(i)} \leftarrow \text{Decoder}\left(\hat{\mathbf{h}}_t, \mathbf{s}_t, \mathbf{o}_{v_i}^t\right)$
      end if
    end for
    Observe reward $r_t = w_1 r_{\text{idle}}(t) + w_2 r_{\text{sr}}(t)$ resulting from the joint action $\mathbf{a}_{t-1}$
    Store transition $(\mathbf{s}_{t-1}, \mathbf{a}_{t-1}, r_t, \mathbf{s}_t)$, where $\mathbf{a}_{t-1}$ denotes the rebalancing actions taken at decision step $t-1$
    Update encoder parameters by minimizing MAE between $\hat{\mathbf{h}}_t$ and $\mathbf{h}_t$
    Update decoder parameters using sampled transitions from the replay buffer
  end if
  $t \leftarrow t + 1$
end while

waiting time, maximum travel delay, and vehicle capacity, thereby defining the admissible decision space of the system. Within this feasible space, the proposed encoder-decoder Deep Reinforcement Learning (DRL) model learns cooperative fleet rebalancing policies aimed at improving service profitability and reducing CO_2 emissions.

4.1 Overview

To simulate a more realistic urban driving environment, we integrate an actual road network from Open Street Map (OSM) into the Simulation of Urban MObility (SUMO) environment. Additionally, we use an open dataset [Donovan *et al.*, 2016] providing real-time traffic volumes to simulate the movements of HDVs. In this simulated urban scenario, a fleet of CAVs operates in traffic congestion caused by HDVs and responds to passenger requests. Vehicle-to-passenger matching is performed at each decision time step using an optimization algorithm to maximize the number of served passengers while maintaining acceptable service quality. The matching algorithm is inspired by the greedy algorithm proposed by Alonso-Mora et al. [Alonso-Mora *et al.*, 2017] and complies with all constraints specified in Section 3.2. To further improve service rates and reduce environmental pollution, CAVs that remain idle beyond a specified threshold (N minutes) are identified for rebalancing. Observations of idle vehicles are provided to an encoder-decoder architecture within the proposed framework, which determines optimal repositioning locations by balancing objectives of increasing service rates and reducing CO_2 emissions. Algorithm 1 summarizes the overall procedure of the proposed method under the Centralized Training and Decentralized Execution (CTDE) paradigm. During training, the encoder and decoder are optimized jointly, as shown in Fig. 3. The encoder is trained using both the supervised prediction loss and gradients propagated from the reinforcement learning objective, whereas the decoder is optimized as the reinforcement learning policy. The global state and the

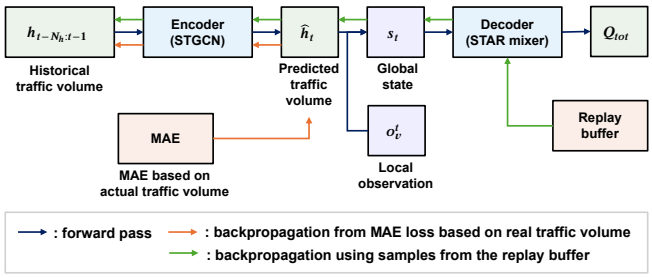


Figure 3: Training flow of the encoder-decoder architecture. Forward and backward computational flows are illustrated across the encoder, mixer, and decoder modules during training.

STAR mixer are used only during centralized training to enable coordinated credit assignment. During execution, each CAV selects its action based solely on its local observation.

4.2 Encoder

In the encoder-decoder architecture, the encoder generates predicted traffic congestion states $\hat{\mathbf{h}}_t$ and passes them to the decoder within the STAR framework. Effective traffic-volume forecasting is essential for eco-friendly AMoD systems to respond to dynamic congestion conditions involving spatial and temporal relationships. To address this, our framework adopts Spatio-Temporal Graph Convolutional Networks (STGCN) [Yu *et al.*, 2018], which leverage Graph Convolutional Networks (GCNs) to capture these spatial and temporal dependencies. The encoder receives traffic volumes from the SUMO environment for the last N_h decision time steps, grouped into spatial regions via K-means clustering based on spatial proximity, and predicts traffic conditions for the next N_p decision time steps. Although originally developed to estimate average vehicle speeds, STGCN is adapted in our framework to predict regional traffic volumes for the AMoD system, achieving a Mean Absolute Error (MAE) of approximately 0.74 at the 100th epoch, which we find sufficient to capture short-term congestion trends relevant for real-time rebalancing decisions.

4.3 Decoder

The decoder generates cooperative rebalancing actions for idle CAVs based on the predicted traffic states $\hat{\mathbf{h}}_t$ produced by the encoder. In the proposed STAR framework, the decoder is formulated as a cooperative Multi-Agent Reinforcement Learning (MARL) policy operating under a value-decomposition paradigm. Urban AMoD systems depend on sparse and non-stationary spatiotemporal interactions, where only a subset of vehicles meaningfully influences system-level outcomes such as service profitability and CO_2 emissions at any given time. To address the resulting credit assignment challenges, the STAR framework introduces a state-dependent attention-based mixing mechanism, referred to as the STAR mixer. Although centralized signals are used during training, each idle CAV v selects its rebalancing action a_v^t based on its local observation $\mathbf{o}_v^t$ at execution time. During centralized training, the individual agent action-values $Q_v\left(\mathbf{o}_v^t, a_v^t\right)$ are aggregated

into a global action-value function through the STAR mixer:

$$Q_{\text{tot}}\left(\mathbf{s}_t, \mathbf{a}_t\right) = \sum_{v=1}^{I} \alpha_v \left(\mathbf{s}_t, \mathbf{o}_v^t\right) Q_v \left(\mathbf{o}_v^t, a_v^t\right), \tag{9}$$

where $I = |\mathcal{V}_t^{\text{idle}}|$ denotes the number of idle vehicles at time t. For each agent v, an attention logit is computed from the concatenation of the global state $\mathbf{s}_t$ and its local observation $\mathbf{o}_v^t$:

$$e_v = f_{\text{attn}}\left(\left[\mathbf{s}_t, \mathbf{o}_v^t\right]\right), \tag{10}$$

where f_{attn} is a two-layer Multi-Layer Perceptron (MLP) with ReLU hidden activations and $[\cdot, \cdot]$ denotes concatenation. The attention weights are then obtained via softmax across agents:

$$\alpha_v = \frac{\exp(e_v)}{\sum_j \exp(e_j)}, \tag{11}$$

which ensures $\alpha_v > 0$ and $\sum_v \alpha_v = 1$. Since each e_v is computed independently per agent, the mixer accommodates a variable number of active vehicles without architectural changes. Moreover, the aggregation in Eq. (9) is invariant to the ordering of agents. Furthermore, the non-negativity of α_v satisfies the sufficient condition for the Individual-Global-Max (IGM) principle [Rashid *et al.*, 2020], as it guarantees $\partial Q_{\text{tot}} / \partial Q_v \geq 0$ for all v regardless of the state or the number of agents. This ensures that the local greedy actions derived from Q_v are consistent with the global optimal joint action.

4.4 Markov Decision Process (MDP)

We model the fleet rebalancing problem as a Decentralized Partially Observable Markov Decision Process (Dec-POMDP).

State and observation

The global state $\mathbf{s}_t \in \mathcal{S}$ represents the overall environment at time step t, including all vehicle states, passenger requests, and traffic conditions:

$$\mathbf{s}_t = \left(\hat{\mathbf{h}}_t, \left(\mathbf{o}_v^t\right)_{v \in \mathcal{V}_t}\right), \tag{12}$$

where $\mathbf{o}_v^t$ is each vehicle's local observation, and $\hat{\mathbf{h}}_t$ is the predicted regional traffic volumes. In the decentralized execution setting, each CAV v makes decisions based exclusively on its local observation $\mathbf{o}_v^t$. To capture the local context while maintaining a fixed input dimension for the policy network, we define the observation as:

$$\mathbf{o}_v^t = \left(\xi_v^t, \left(\rho_i^t\right)_{i \in \mathcal{P}_k(v)}\right), \tag{13}$$

where $\xi_v^t = (\ell_v^t, \tau_v^t, b_v^t)$ is the per-vehicle state, and $\rho_i^t = \left(r_i, \ell_i^p, \ell_i^d\right)$ is the state of a request. Crucially, $\mathcal{P}_k(v)$ denotes the set of the k nearest unassigned passenger requests to vehicle v, ensuring a fixed-sized input vector.

Action

At each decision time step t, every idle vehicle $v \in \mathcal{V}_t^{\text{idle}}$ selects a rebalancing location a_v^t from a discrete action space $\mathcal{A}$. Each agent selects its action greedily based on its individual value function, $\mathbf{a}_v^t = \arg\max_{a \in \mathcal{A}} Q_v\left(\mathbf{o}_v^t, a\right)$. The individual value functions are aggregated through a monotonic structure that guarantees that increasing any individual utility does not decrease the global state-action value, as formalized in Eq. (9).

Transitional dynamics

The transition function $P\left(\mathbf{s}_{t+1} | \mathbf{s}_t, \mathbf{a}_t\right)$ is implicitly defined by the SUMO traffic simulator. After agents take a joint action $\mathbf{a}_t$ at decision time step t, the system evolves for an interval Δt. During this time, vehicles move, new requests arrive, and traffic conditions change, leading to the next global state $\mathbf{s}_{t+1}$.

Reward

For cooperative behavior, all agents receive a shared global reward R_t at the end of each decision interval. The reward function is designed to learn a policy that improves the passenger service rate while simultaneously minimizing vehicle idle time, a proxy for reducing CO_2 emissions. This proxy is used because direct carbon emission signals suffer from high variance and a wide dynamic range. Normalizing such high-variance signals often leads to reward sparsity, which destabilizes the learning process. Instead, we adopt vehicle idle time as a surrogate reward signal that produces smooth gradients and directly captures prolonged low-speed driving and stop-and-go congestion patterns known to increase fuel consumption and emissions [Dhanorkar and Burtch, 2022]. We empirically verify that cumulative idle time strongly correlates with per-service CO_2 emissions in our simulation environment, with a Pearson correlation coefficient of 0.708. Thus, the reward function consists of two components.

- **Passenger service reward (r_{sr}).** It encourages serving as many passengers as possible. It is defined as the cumulative service rate, the ratio of all serviced requests to all received requests up to time step t:

$$r_{\text{sr}}(t) = \frac{|\mathcal{S}_{1:t}|}{|\mathcal{R}_{1:t}|}, \tag{14}$$

  where $|\cdot|$ denotes the cardinality of a set, and $\mathcal{X}_{1:t}$ is defined as $\cup_{i=1}^{t} \mathcal{X}_i$ for an arbitrary set $\mathcal{X}$.

- **Idle time penalty (r_{idle}).** It penalizes vehicles for being idle longer than a predefined threshold. The strategy aims to prevent significant carbon emissions from idling and helps vehicles avoid traffic jams. Specifically, the penalty is formulated using the hyperbolic tangent function to be smooth and saturated:

$$r_{\text{idle}}(t) = -\sum_{v \in \mathcal{V}_t^{\text{idle}}} \tanh\left(\gamma\left(\tau_v^t - \delta\right)\right), \tag{15}$$

  where τ_v^t is the duration vehicle v has been idle, δ is the maximum allowable idle duration, and γ is a scaling constant.

To balance these two competing objectives, the final reward function is a linear combination of these components:

$$r(t) = w_1 r_{\text{idle}}(t) + w_2 r_{\text{sr}}(t), \tag{16}$$

where w_1 and w_2 are positive weighting factors that determine the relative importance of service rate and idle time reduction.

5 Experiments

We evaluate the proposed framework in a realistic urban traffic simulation built on Lower Manhattan, New York City,

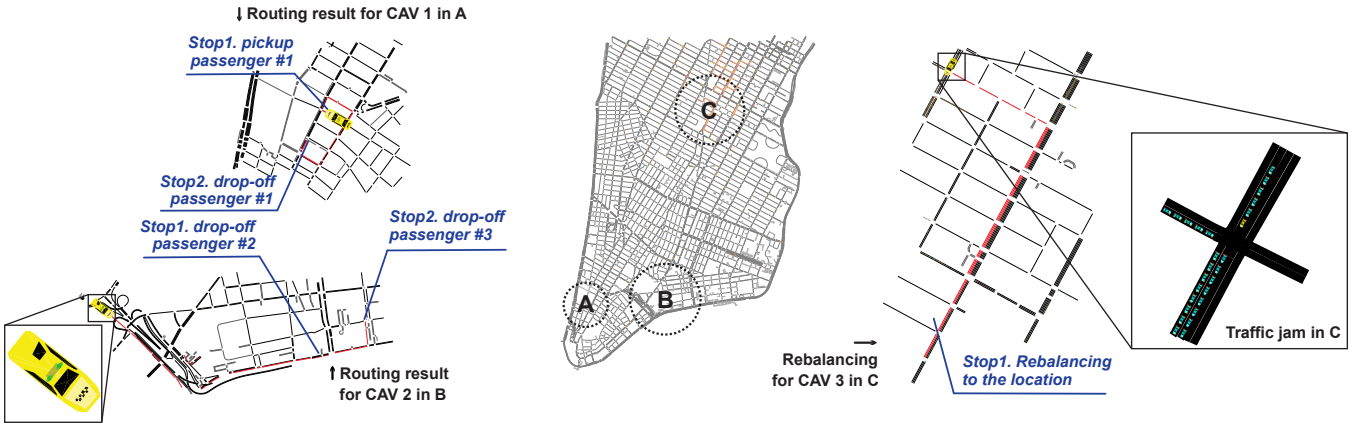


Figure 4: A moment from the SUMO simulation. It visualizes CAVs in yellow, HDVs in cyan, and passengers in green. CAV 1, currently empty, is planning a route to pick up and drop off passenger 1. CAV 2, carrying two passengers, is planning a route to drop off passenger 2 and then passenger 3. CAV 3, idle for N minutes, anticipates demand and moves towards a region with lower expected CO_2 emissions. Currently, CAV 3 is located in a heavily congested area, surrounded by many HDVs.

selected for its heterogeneous travel demand, and availability for high-quality public datasets [Donovan *et al.*, 2016; Taxi and Commission, 2024]. Our study focuses on For-Hire Vehicle (FHV) services regulated by the New York City Taxi and Limousine Commission (TLC). The simulation environment is implemented in SUMO and integrates two real-world datasets. The first dataset consists of FHV passenger requests recorded in January 2024 [Taxi and Commission, 2024], which are injected into the simulator according to their original request times and pickup locations. The second dataset provides hourly average traffic volumes for road segments by day of the week [Donovan *et al.*, 2016]. Based on these volumes, Human-Driven Vehicles (HDVs) are generated by randomly sampling departure times within each hourly interval, creating realistic background congestion. Moreover, vehicle-level CO_2 emissions are estimated online using the Handbook Emission Factors for Road Transport (HBEFA), which models emissions as a function of speed, acceleration, deceleration, and road type. Following European standards, we adopt the HBEFA3/PC_G_EU4 category for gasoline passenger vehicles.

Each training epoch simulates a three-day horizon, starting on January 4, 2024. All experiments are performed on a machine equipped with 32GB RAM, an NVIDIA RTX 4090 GPU, and an AMD Ryzen 7 7800X3D CPU. Each epoch takes approximately 39.5 minutes, resulting in a total training time of about 39 hours and 40 minutes for 100 epochs. Using the best-performing checkpoint selected during training, we report all metrics as the mean and standard deviation over ten evaluation runs.

To provide an intuitive illustration of the operational challenges captured in this simulation, Fig. 4 shows a snapshot of the simulation environment. CAVs operate under realistic congestion induced by HDVs. In this example, one CAV is actively serving passengers, another is completing sequential drop-offs, and an idle CAV anticipates future demand by repositioning. Notably, the idle CAV is trapped in a heavily congested region, emitting high levels of CO_2 while being unable to efficiently approach nearby passengers.

5.1 Baselines

To evaluate the STAR framework proposed in Section 4 for reducing CO_2 emissions and improving the service rate of our AMoD system, three baselines are established.

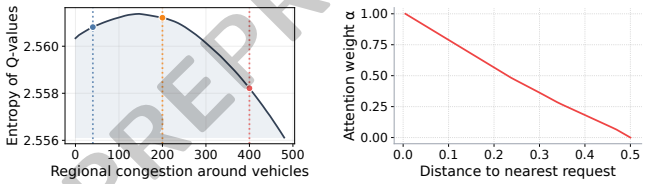
- **Greedy Rebalancing (GR) [Alonso-Mora *et al.*, 2017].** The method operates in two sequential stages: greedy assignment and rebalancing. First, vehicles are greedily matched to passenger requests to maximize the number of served passengers using a demand-responsive assignment strategy. After this initial matching, the remaining idle vehicles are rebalanced to serve unassigned requests. This rebalancing stage is formulated as a linear assignment problem that minimizes the total travel time from idle vehicles to waiting passengers. The procedure is repeated until either all idle vehicles are dispatched or no unserved requests remain. Since no official open-source implementation is available, we implemented the algorithm based on its original description.
- **Historical Average (HA).** The method rebalances idle vehicles toward areas with the lowest traffic congestion. The target locations are selected based on average traffic volumes observed over the previous N_h decision steps. Passenger assignment is then performed using the matching procedure proposed by Alonso et al. [Alonso-Mora *et al.*, 2017].
- **DeepPool [Al-Abbasi *et al.*, 2019].** It is a distributed, model-free ridesharing dispatch algorithm based on DQN, where each vehicle independently learns a dispatch policy through its own agent. The learned policy jointly minimizes supply-demand mismatch, passenger waiting time, idle driving distance, and additional delays from shared rides, enabling vehicles to both serve current requests and proactively reposition toward high-demand regions. Since the official implementation of DeepPool is not publicly available and was not designed for realistic urban simulators such as SUMO, we reimplemented its core mechanisms within our SUMO environment.

# of CAVs	Scenario	Operational metrics			User experience metrics	
		CO_2 per passenger (kg) ↓	Service rate ↑	Average travel distance (km)	Average travel delay (s)	Average waiting time (s)
5	GR	0.519 ± 0.008	0.506 ± 0.014	23.116 ± 1.166	62.235 ± 6.338	132.712 ± 1.385
	HA	0.594 ± 0.022	0.485 ± 0.026	33.812 ± 5.249	61.627 ± 13.672	138.163 ± 6.855
	DeepPool	1.321 ± 0.022	0.530 ± 0.111	31.811 ± 6.022	68.394 ± 12.227	127.519 ± 19.354
	STAR (Ours)	$\mathbf{0.397 \pm 0.129}$	$\mathbf{0.582 \pm 0.105}$	27.267 ± 12.102	101.371 ± 78.235	133.310 ± 6.151
10	GR	0.848 ± 0.030	0.788 ± 0.016	16.663 ± 5.319	70.131 ± 12.740	81.481 ± 2.189
	HA	1.029 ± 0.142	0.784 ± 0.032	10.984 ± 8.180	57.807 ± 6.625	86.327 ± 4.296
	DeepPool	1.329 ± 0.065	0.780 ± 0.025	29.344 ± 9.947	54.719 ± 7.935	87.365 ± 4.183
	STAR (Ours)	$\mathbf{0.466 \pm 0.035}$	$\mathbf{0.830 \pm 0.020}$	23.140 ± 6.396	60.311 ± 17.294	76.311 ± 1.096
100	GR	3.438 ± 0.082	0.890 ± 0.012	23.634 ± 5.342	44.504 ± 7.140	46.923 ± 0.017
	HA	3.753 ± 0.097	0.916 ± 0.023	16.328 ± 4.767	59.683 ± 5.443	57.942 ± 4.037
	DeepPool	7.965 ± 0.257	0.976 ± 0.000	17.217 ± 5.155	72.577 ± 14.002	47.558 ± 1.385
	STAR (Ours)	$\mathbf{2.418 \pm 0.268}$	$\mathbf{0.980 \pm 0.000}$	18.445 ± 10.080	66.504 ± 12.110	47.272 ± 0.599

Table 3: Comparison of operational and user experience metrics under varying fleet sizes. Lower CO_2 emissions per passenger and higher service rates are desirable outcomes. Other metrics, while not primary objectives of this study, were measured to assess service quality.

Symbol	Description	Value
C	Maximum passenger capacity per vehicle	3
W	Maximum allowable waiting time	3 minutes
T	Maximum allowable travel delay	5 minutes
N	Idle time threshold for rebalancing	3 minutes
Δt	Decision time interval	90 seconds
N_h	Historical horizon for traffic prediction	12
N_p	Prediction horizon	1
K	Number of spatial regions for traffic aggregation	13
k	Number of nearest unserved requests in observation	3
w_1	Weight of idle time penalty	0.5
w_2	Weight of service rate reward	0.5

Table 4: Experimental parameters. These are key parameters used in the main experiments.



(a) Entropy of action values with respect to regional congestion.

(b) Attention weight α_v by distance to the nearest request.

Figure 5: Interpretability of attention. The attention weights reflect both traffic congestion and proximity to passenger demand, indicating that the mechanism captures meaningful coordination signals.

5.2 Simulation results

We evaluate the STAR framework in terms of quantitative comparison against baselines, qualitative analysis of the attention mechanism, and computational complexity.

Quantitative results

Table 3 presents a comprehensive quantitative comparison of the STAR framework against several baselines across fleet sizes of 5, 10, and 100 CAVs to evaluate scalability. The key parameters used in these experiments are summarized in Table 4, while additional training settings, such as network layer sizes and learning rates, are provided in the released open-source code. Performance is evaluated using five metrics. Total CO_2 emissions per passenger (kg) measures the average carbon emissions generated by the CAV fleet for each successfully served passenger. The service rate refers to the percentage of requests that are successfully serviced. The average travel distance (km) measures the mean distance traveled by CAVs, while the average travel delay (s) captures the additional time experienced by passengers in shared rides compared to solo trips. Lastly, the average waiting time (s) represents the average duration passengers wait from the moment of request until pickup. Among these metrics, we primarily focus on CO_2 emissions and service rate, as they directly reflect the objective of our framework. The remaining metrics are reported for service quality evaluation. In particular, the user experience metrics verify that the STAR framework consistently satisfies the imposed constraints on maximum allowable waiting time and travel delay.

As shown in Table 3, although the greedy rebalancing method [Alonso-Mora *et al.*, 2017] focuses on maximizing service rate, the STAR framework consistently achieves higher service rates across all fleet sizes while substantially reducing CO_2 emissions. Across all experimental scales, STAR reduces per-passenger CO_2 emissions while improving service rates relative to all baseline methods. Specifically, for fleets of 5, 10, 100 CAVs, STAR reduces per-passenger CO_2 emissions by up to 69.9%, 64.9%, and 69.6%, respectively, while increasing service rates by up to 20.0%, 6.4%, and 10.1%.

Qualitative results

To understand what the attention learns, we analyze the learned weights across the STAR framework scenarios reported in Table 3. Fig. 5 shows that attention does not depend on congestion alone. As shown in Fig. 5a, higher congestion leads to lower Q-value entropy, so the optimal rebalancing action is more clearly defined. In Fig. 5b, attention decreases with

Number of CAVs	Number of parameters	Inference time (μs)	Training time (min / epoch)
5	33,755	145.2	32.0
10	66,470	261.3	32.4
100	655,340	2,390.1	39.5

Table 5: Computational scalability of the STAR framework. The number of parameters, inference time, and training time per epoch are reported across varying fleet sizes.

# of CAVs	Scenario	CO_2 per passenger (kg) ↓	Service rate ↑
5	DQN	1.443 ± 0.038	0.388 ± 0.013
	TD3	0.875 ± 0.020	0.462 ± 0.020
	QMIX	0.760 ± 0.204	0.574 ± 0.115
	STAR (Ours)	$\mathbf{0.397 \pm 0.129}$	$\mathbf{0.582 \pm 0.105}$
10	DQN	1.152 ± 0.197	0.612 ± 0.088
	TD3	1.007 ± 0.076	0.538 ± 0.032
	QMIX	0.501 ± 0.102	0.794 ± 0.073
	STAR (Ours)	$\mathbf{0.466 \pm 0.035}$	$\mathbf{0.830 \pm 0.020}$
100	DQN	3.892 ± 0.406	0.976 ± 0.000
	TD3	2.423 ± 0.003	0.966 ± 0.027
	QMIX	2.645 ± 0.144	$\mathbf{0.980 \pm 0.000}$
	STAR (Ours)	$\mathbf{2.418 \pm 0.268}$	$\mathbf{0.980 \pm 0.000}$

Table 6: Ablation study comparing different decoders under varying fleet sizes. For clarity, all baseline methods are referred to by their corresponding decoder algorithms. Lower CO_2 emissions per passenger and higher service rates are desirable outcomes.

the distance to the nearest request, so agents closer to demand receive higher attention. Overall, these results indicate that the attention mechanism captures meaningful coordination rather than increasing model capacity.

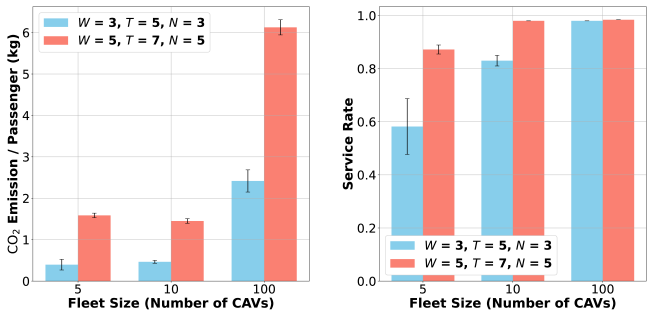
Computational complexity

As shown in Table 5, the number of parameters and inference time scale approximately linearly with the number of CAVs, while training time per epoch remains nearly constant, demonstrating that the STAR framework scales efficiently to large fleets.

5.3 Ablation study

In addition, we conduct an ablation study by replacing the decoder of our framework with three representative Multi-Agent Reinforcement Learning (MARL) algorithms [Mnih *et al.*, 2015; Fujimoto *et al.*, 2018; Rashid *et al.*, 2020] under the Centralized Training with Decentralized Execution (CTDE) paradigm. It allows us to isolate the effect of the proposed decoder design. Table 6 shows that STAR reduces per-passenger CO_2 emissions by up to 48-72% compared to alternative decoders, while achieving superior service rates across all fleet sizes.

Furthermore, we perform sensitivity analyses by systematically varying key system parameters to evaluate the robustness of the proposed framework. In particular, to examine STAR's behavior under relaxed service quality constraints



(a) Per-passenger CO_2 emissions under different W, T, and N.

(b) Service rate under different W, T, and N.

Figure 6: Trade-off under relaxed service quality constraints. Per-passenger CO_2 emissions and service rate achieved by STAR when the waiting time, travel delay, and idle time constraints are relaxed from $(W, T, N) = (3, 5, 3)$ to $(5, 7, 5)$ minutes.

compared to the default setting in Table 4, we increase the maximum waiting time, travel delay, and idle time threshold from $(W, T, N) = (3, 5, 3)$ to $(5, 7, 5)$ minutes. This setting allows greater flexibility in vehicle dispatch and rebalancing decisions. Under these looser constraints, the STAR framework is able to serve a large number of passengers even with a smaller fleet size, as shown in Fig. 6. However, this improvement in service coverage comes at the cost of increased per-passenger CO_2 emissions, as vehicles are permitted to tolerate longer delays and idle periods. These results show the inherent trade-off between service efficiency and environmental sustainability. At the same time, the performance trends confirm that STAR responds robustly to variations in constraints.

6 Conclusion

This paper presented the STAR framework, a cooperative MARL framework for eco-friendly fleet rebalancing in large-scale AMoD systems. By combining spatio-temporal traffic prediction with attention-based value decomposition, STAR enables robust credit assignment under sparse and non-stationary urban interactions while strictly enforcing service-quality constraints.

Experiments in a high-fidelity SUMO environment with real-world demand and background traffic demonstrate that STAR improves service rates while substantially reducing per-passenger CO_2 emissions across different fleet sizes, outperforming existing baselines. Future work will extend the STAR framework to heterogeneous fleets.

Ethical Statement

There are no ethical issues.

Acknowledgments

This work was supported by the InnoCORE program of the Ministry of Science and ICT (#N10250155), by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (No.RS-2022-NR070854), by

the Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korean government (MSIT) (No.RS-2020-II201336, Artificial Intelligence Graduate School Program (UNIST)), by the Green Venture R&D Program (#S3236472) funded by the Ministry of SMEs and Startups (MSS, Korea), and by the Electronics and Telecommunications Research Institute (ETRI) grant funded by the Korean government [26ZD1160, Regional Industry ICT Convergence Technology Advancement and Support Project in Daegu-GyeongBuk].

Contribution Statement

Jungeun Lee†: Conceptualization, Methodology, Software, Formal analysis, Investigation, Data curation, Writing – original draft, Writing – review & editing, Visualization. **Seungjae Baek**†: Formal analysis, Investigation, Writing – original draft, Writing – review & editing, Visualization. **Seongjae Lee**: Formal analysis, Investigation. **Sunhwi Kim**: Conceptualization, Software. **Jeong hwan Jeon**∗: Conceptualization, Writing – review & editing, Supervision, Funding acquisition.

References

[Al-Abbasi *et al.*, 2019] Abubakr Al-Abbasi, Arnob Ghosh, and Vaneet Aggarwal. DeepPool: Distributed model-free algorithm for ride-sharing using deep reinforcement learning. *IEEE Transactions on Intelligent Transportation Systems*, 20:4714–4727, 2019.

[Alonso-Mora *et al.*, 2017] Javier Alonso-Mora, Samitha Samaranayake, Alex Wallar, Emilio Frazzoli, and Daniela Rus. On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment. *Proceedings of the National Academy of Sciences*, 114(3):462–467, 2017.

[Chang *et al.*, 2024] Xinyu Chang, Jing Wu, Zhaoyang Kang, et al. Estimating emissions reductions with carpooling and vehicle dispatching in ridesourcing mobility. *npj Sustainable Mobility and Transport*, 1(16), 2024.

[Dhanorkar and Burtch, 2022] Suvrat Dhanorkar and Gordon Burtch. The heterogeneous effects of p2p ride-hailing on traffic: Evidence from uber's entry in california. *Transportation Science*, 56(3):750–774, 2022.

[Donovan *et al.*, 2016] Brian Donovan, Alec Mori, Nimit Agrawal, Yalan Meng, Jong Lee, and Daniel Work. New York City hourly traffic estimates (2010-2013). University of Illinois at Urbana-Champaign, 2016.

[Fagnant and Kockelman, 2015] Daniel J. Fagnant and Kara Kockelman. Preparing a nation for autonomous vehicles: opportunities, barriers and policy recommendations. *Transportation Research Part A: Policy and Practice*, 77:167–181, 2015.

[Fiedler *et al.*, 2017] David Fiedler, Michal Čáp, and Michal Čertický. Impact of mobility-on-demand on traffic congestion: Simulation-based study. In *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–6, 2017.

† These authors contributed equally to this work.
∗ Corresponding author.

[Fujimoto *et al.*, 2018] Scott Fujimoto, Herke van Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International Conference on Machine Learning*, 2018.

[Guo *et al.*, 2023] Jiaying Guo, Long Cheng, and Shen Wang. CoTV: Cooperative control for traffic light signals and connected autonomous vehicles using deep reinforcement learning. *IEEE Transactions on Intelligent Transportation Systems*, 24(10):10501–10512, 2023.

[Haliem *et al.*, 2020] Marina Haliem, Ganapathy Mani, Vaneet Aggarwal, and Bharat Bhargava. A distributed model-free ride-sharing algorithm with pricing using deep reinforcement learning. In *Proceedings of the 4th ACM Computer Science in Cars Symposium*. Association for Computing Machinery, 2020.

[Jiao *et al.*, 2021] Yan Jiao, Xiaocheng Tang, Zhiwei (Tony) Qin, Shuaiji Li, Fan Zhang, Hongtu Zhu, and Jieping Ye. Real-world ride-hailing vehicle repositioning using deep reinforcement learning. *Transportation Research Part C: Emerging Technologies*, 130:103289, 2021.

[Mnih *et al.*, 2015] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

[Qin *et al.*, 2021] Guoyang Qin, Qi Luo, Yafeng Yin, Jian Sun, and Jieping Ye. Optimizing matching time intervals for ride-hailing services using reinforcement learning. *Transportation Research Part C: Emerging Technologies*, 129:103239, 2021.

[Rashid *et al.*, 2020] Tabish Rashid, Mikayel Samvelyan, Christian Schroeder De Witt, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. Monotonic value function factorization for deep multi-agent reinforcement learning. *Journal of Machine Learning Research*, 21(178):1–51, 2020.

[Sahebdel *et al.*, 2025] Mahsa Sahebdel, Ali Zeynali, Noman Bashir, Prashant Shenoy, and Mohammad Hajiesmaili. LEAD: Towards learning-based equity-aware decarbonization in ridesharing platforms. In *Proceedings of the 2025 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '25, page 817–827, New York, NY, USA, 2025. Association for Computing Machinery.

[Sunehag *et al.*, 2017] Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech M. Czarnecki, Vinícius Flores Zambaldi, Max Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z. Leibo, Karl Tuyls, and Thore Graepel. Value-decomposition networks for cooperative multi-agent learning. *ArXiv*, abs/1706.05296, 2017.

[Taxi and Commission, 2024] New York City Taxi and Limousine Commission. TLC trip record data. https://www.nyc.gov/site/tlc/about/tlc-trip-record-data.page, 2024. Accessed: 23-Apr-2025.

[Tsao *et al.*, 2019] Matthew Tsao, Dejan Milojevic, Claudio Ruch, Mauro Salazar, Emilio Frazzoli, and Marco Pavone. Model predictive control of ride-sharing autonomous mobility-on-demand systems. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 6665–6671. IEEE, 2019.

[Wang *et al.*, 2020] Jianhao Wang, Zhizhou Ren, Terry Liu, Yang Yu, and Chongjie Zhang. QPLEX: Duplex dueling multi-agent q-learning. *arXiv preprint arXiv:2008.01062*, 2020.

[Yu *et al.*, 2018] Bing Yu, Haoteng Yin, and Zhanxing Zhu. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 3634–3640. International Joint Conferences on Artificial Intelligence Organization, 7 2018.