Communications in Transportation Research 5 (2025) 100225

Contents lists available at ScienceDirect

Communications in Transportation Research

journal homepage: www.journals.elsevier.com/communications-in-transportation-research

Full Length Article

Scalable and reliable multi-agent reinforcement learning for traffic assignment

Leizhen Wang[a], Peibo Duan[a,*], Cheng Lyu[b], Zewen Wang[c], Zhiqiang He[d], Nan Zheng[e], Zhenliang Ma[f,**]

[a] Department of Data Science and Artificial Intelligence, Monash University, Clayton, VIC, 3800, Australia
[b] Chair of Transportation Systems Engineering, Technical University of Munich, Munich, 80333, Germany
[c] School of Transportation, Southeast University, Nanjing, 211189, China
[d] The Graduate School of Informatics and Engineering, The University of Electro-Communications, Tokyo, 1828585, Japan
[e] Department of Civil and Environmental Engineering, Monash University, Clayton, VIC, 3800, Australia
[f] Department of Civil and Architectural Engineering, KTH Royal Institute of Technology, Stockholm, 10044, Sweden

ARTICLE INFO

Keywords:
Multi-agent reinforcement learning
Traffic assignment
User equilibrium
Route choice

ABSTRACT

The evolution of metropolitan cities and increasing travel demand impose stringent requirements on traffic assignment methods. Multi-agent reinforcement learning (MARL) approaches outperform traditional methods in modeling adaptive routing behavior without requiring explicit system dynamics, making them attractive for real-world deployment. However, existing MARL frameworks face scalability and reliability challenges when managing large-scale networks with substantial and variable demand. This study proposes MARL-OD-DA, a novel framework that redefines agents as origin–destination (OD) pair routers and employs a continuous simplex-constrained action space. This reformulation reduces the agent population from $O(N)$ (number of travelers) to $O(|D|)$ (number of OD pairs), achieving at least two orders of magnitude fewer agents in practice while preserving convexity and enabling efficient adaptation to demand variation, thus significantly improving scalability. In contrast to prior MARL studies constrained to small-sized networks (up to 70 nodes, 2100 travelers) and fixed demand, MARL-OD-DA is validated on medium-sized networks (up to 416 nodes, 1406 OD pairs, and 360,600 travelers) under varying demand scenarios, demonstrating substantial improvements in scalability and applicability. To further enhance reliability, the framework integrates a Dirichlet-based policy, action pruning, and a relative gap-based reward. Theoretical analysis demonstrates that the Dirichlet-based policy reduces gradient bias, stabilizes variance, and enables sparse routing decisions, in contrast to the commonly used softmax-based policy. Experiments on three benchmark networks show that MARL-OD-DA significantly improves assignment quality and convergence speed. On the SiouxFalls network, the trained agents converge within 10 iterations during deployment, reducing the relative gap by 94.99% compared to conventional baselines.

1. Introduction

Traffic assignment is a pivotal element in traffic planning, concerned with the distribution of origin-destination (OD) demand within a transport network to achieve user equilibrium (UE) or system optimal (SO) objectives, as informed by the system goals (Boyles et al., 2020; de Dios Ortúzar and Willumsen, 2024).[1] The rise of metropolitan areas and the increase in travel demands require advanced online decision making tools to provide route guidance to human drivers, connected and automated vehicles, and shared mobility on demand. These developments place growing pressure on the efficiency, scalability, and real-time performance of traffic assignment methods (Guo et al., 2021; Liu et al., 2020; Lujak et al., 2015; Luo et al., 2023; Zhou and Roncoli, 2022).

Driven by recent advancements in artificial intelligence (AI) and data acquisition technologies, traditional traffic assignment methods have undergone substantial refinement. These approaches can be classified into two categories: supervised learning (SL)-based (Liu and Meidani,

* Corresponding author.
** Corresponding author.
E-mail addresses: peibo.duan@monash.edu (P. Duan), zhema@kth.se (Z. Ma).

[1] UE focuses on minimizing travel costs for individuals, whereas SO aims to minimize the total network-wide costs.

https://doi.org/10.1016/j.commtr.2025.100225
Received 2 August 2025; Received in revised form 17 September 2025; Accepted 17 September 2025
Available online xxxx

2024; Rahman and Hasan, 2023) and multi-agent reinforcement learning (MARL)-based (Ramos et al., 2018; Shou et al., 2022; Zhou et al., 2020). SL-based methods predict UE flow patterns but overlook critical travel behaviors, such as route choice, and fail to ensure link-path flow relationships. In contrast, MARL-based approaches bridge the gap in SL-based models by viewing travelers as independent agents who learn from continuous interactions with the traffic environment. This enables them to respond to dynamic traffic conditions, operate under partial information, handle complex and non-convex traffic scenarios, and integrate individual behavior modeling with system-level optimization. Furthermore, to improve the performance of MARL-based methods, most studies focus on action regulation like en-route and route decisions.

Despite advancements in MARL frameworks have led to impressive performance through enhanced modeling capabilities, such as action regulation, they still encounter substantial hurdles, notably in terms of scalability and reliability (S&R), which are mainly attributed to the intrinsic modeling approach. As existing MARL frameworks typically assign one agent per traveler, they often struggle to scale when the number of agents increases, given that real-world transport systems may involve huge numbers of travelers. For example, Shou et al. (2022) proposed a mean field Q-learning approach to address scalability issues, yet their trials were limited to 2100 travelers. Another study addressed the traffic assignment problem involving 360,600 travelers, but did not achieve the desired accuracy (Bazzan and Grunitzki, 2016). Furthermore, the largest traffic networks in these methods comprise no more than 69 nodes and 166 links (Ramos et al., 2018; Shou et al., 2022; Zhou et al., 2020), still representing small-sized networks insufficient for real-world application. In tandem with the scalability issue, flexibility has emerged as a critical concern. Due to the agent-per-traveler design, MARL frameworks rely on a fixed number of agents during training. However, in real-world scenarios, fluctuations in OD demand require adjustments in the agent configuration, necessitating model retraining or fine-tuning. This process incurs substantial computational overhead and introduces risks such as catastrophic forgetting, thereby limiting the practical deployment of MARL in dynamic traffic environments.

The reliability of MARL-based approaches is demonstrated through their ability to efficiently converge to a plausible solution applicable in real-world situations. However, many current investigations oversimplify real-world conditions by modeling travelers as making route choices solely based on location and time, neglecting the complexity of their decision-making processes, which compromises the quality of the solution. Furthermore, using average travel time to assess convergence to the UE works at cross purposes since it more likely aligns with the SO goal rather than minimizing individual costs. Braess' Paradox highlights that reaching UE can lead to increased average travel times compared to SO (Zhuang et al., 2022).

In order to tackle the previously mentioned issues of scalability and reliability, this study re-examines the MARL framework applied to traffic assignment, encompassing the definition of agents and the tailored state-action-reward formulation. The main contributions are as follows:

- **Scalable and reliable MARL formulation**: A novel MARL formulation is proposed, where each agent is redefined as an OD pair router and adopts a continuous simplex-constrained action space. This reformulation reduces agent dimensionality from $O(N)$ (number of travelers) to $O(|D|)$ (number of OD pairs), while maintaining a convex feasible region. It also allows adaptation to variable OD demand without retraining, significantly improving scalability and learning stability in larger networks.
- **Reliable policy design**: The proposed MARL-OD-DA framework integrates a Dirichlet-based policy, action pruning mechanism, and a relative gap-based reward to enhance reliability. Theoretical analysis demonstrates the advantages of the Dirichlet-based policy in bias reduction, variance control, and sparse action representability. This unified design improves convergence stability and routing consistency under variable OD demand.
- **Validation**: The proposed method is validated on three transportation networks, including medium-sized systems with over 400 nodes and 300,000 travelers under variable OD demand. Compared to prior MARL approaches limited to small networks and fixed demand, our results show substantial gains in scalability and reliability, extending MARL applicability to more realistic urban settings.

The remainder of this paper is organized as follows. Section 2 reviews conventional and learning-based traffic assignment methods. Section 3 outlines the problem and limitations of existing MARL frameworks. Section 4 details the proposed framework. Section 5 evaluates its performance on three transportation networks. Finally, Section 6 concludes the study and outlines future research directions.

2. Related work

2.1. Conventional traffic assignment methods

Traffic assignment problems are typically solved using mathematical programming methods. These methods can be categorized into three types: link-based (Frank et al., 1956; Fukushima, 1984; Mounce and Carey, 2015), path-based (Chen et al., 2020; Jayakrishnan et al., 1994), and origin-based approaches (Bar-Gera, 2002; Nie, 2010), all of them follow an iterative workflow: First, calculate travel times, then identify the shortest paths, adjust route choices toward equilibrium, and repeat the process. In parallel, several studies have explored traffic assignment from the perspective of traveler behavior and day-to-day dynamics (Siri et al., 2022; Wang et al., 2025a). In recent years, significant efforts have been made to improve the computational efficiency of these methods (Chen et al., 2020; Liu et al., 2024; Zhang et al., 2023). Partly inspired by the path-based method, this paper treats all OD-pairs as agents and performs optimization in parallel, offering a novel perspective to improve scalability and efficiency.

2.2. Learning-based traffic assignment methods

Recent studies have explored learning-based approaches for traffic assignment. For example, Wang et al. (2025a) proposed a large language model (LLM)-based agent to simulate human-like adaptive behavior in road networks, with the aim of achieving UE. However, their focus lies in behavior modeling rather than solving the assignment problem itself. In general, learning-based methods can be categorized into supervised learning (SL) and reinforcement learning (RL) approaches.

SL-based methods. Rahman and Hasan (2023) introduced graph neural networks (GNNs) for predicting UE assignment flows without making assumptions about travelers' routing behavior. Their experimental results demonstrated the ability to predict traffic flows with less than 2% mean absolute error. Liu and Meidani (2024, 2025) further improved this approach by adding auxiliary loss functions, ensuring that the predicted results partially satisfy the constraint of link-path flow relationship. Liu et al. (2023) proposed a novel end-to-end learning framework that directly learns travelers' route choice preferences and the UE flow from observational data. While this method shows potential, it is currently limited to proof-of-concept experiments, and thus was not pursued further in this study. Although SL-based models show promising accuracy in flow prediction, they typically require large volumes of labeled data and often lack interpretability in representing travelers' decision processes, limiting their deployment in dynamic or data-scarce scenarios.

RL-based methods. While SL-based methods primarily approximate UE flow patterns from observed data, RL-based methods instead aim to learn sequential decision-making policies through interaction with the environment. Wang et al. (2025b) proposed a single-agent RL framework that recommends routes for all travelers to achieve SO, with the

agent acting as a centralized decision-maker. In contrast, most existing studies adopt MARL, where each agent represents an individual traveler making decentralized routing decisions.

Based on the granularity of an agent's action, MARL-based methods can be categorized into two main types. The first is route-based methods, in which an agent selects a complete route from a set of pre-defined candidates and strictly follows it from origin to destination (Ramos et al., 2018; Wang et al., 2025b; Zhou et al., 2020). These route sets are commonly generated using k-shortest path algorithms or extracted from historical trajectory data. The second category is en-route methods, where an agent makes sequential decisions at each node by choosing one of the outbound links, continuing this process until the destination is reached or a predefined step limit is exceeded (Bazzan and Grunitzki, 2016; Shou et al., 2022). Compared to route-based approaches, en-route methods offer greater flexibility but often introduce higher complexity in policy learning and convergence.

Compared to conventional model-based traffic assignment techniques, MARL-based approaches offer several distinctive advantages. First, agents learn adaptive routing policies through repeated interactions, allowing them to respond to dynamic traffic conditions (Shou et al., 2022; Zhou et al., 2020). Second, RL-based methods are model-free and do not rely on explicit knowledge of system dynamics. This allows them to operate under partial or noisy observations and to solve complex, non-convex traffic scenarios that are often intractable for traditional optimization techniques. Finally, MARL naturally integrates individual behavior modeling with system-level optimization, eliminating the need for separately calibrated route choice models.

3. Problem statement and analysis

In traffic assignment, the urban transportation network is modeled as a directed graph $G(V, E)$, where V represents the intersections (nodes) and E represents the road segments (edges). Each road segment $e \in E$ is associated with a performance function $c_e(x_e)$, which maps the flow x_e on the segment to its corresponding travel time. The travel demand between OD pairs is represented by a demand matrix $D = \{d_{rs}\}$, where d_{rs} denotes the travel demand from origin node r to destination node s.

The traffic assignment problem seeks to allocate this demand across the available routes in the network, typically with UE or SO objective. For example, the UE objective function is formulated as

$$\min_x \sum_{e \in E} \int_0^{x_e} c_e(v)\, \mathrm{d}v \tag{1}$$

These objectives are subject to the following constraints:

- **Flow conservation.** The total flow distributed across all feasible paths between an origin r and a destination s must match the corresponding demand d_{rs}:

$$\sum_{p \in P_{rs}} f_p = d_{rs}, \quad \forall (r,s) \in D \tag{2}$$

where P_{rs} represents the set of all valid paths between r and s, and f_p denotes the flow assigned to path p.

- **Non-negativity.** All path flows must remain non-negative to ensure feasibility:

$$f_p \geq 0, \quad \forall p \in P_{rs},\ \forall (r,s) \in D \tag{3}$$

- **Link-path flow relationship.** The total flow on a road segment e is calculated as the sum of the flows on all paths that traverse e:

$$x_e = \sum_{(r,s) \in D} \sum_{p \in P_{rs}: e \in p} f_p, \quad \forall e \in E \tag{4}$$

To this end, the MARL framework typically formulates the traffic assignment problem as a decentralized partially observable Markov decision process (DEC-POMDP), which is well-suited for modeling complex multi-agent systems with local observations and decentralized objectives. Formally, a DEC-POMDP is defined by the tuple $\langle S, \mathcal{A}, \mathcal{O}, \mathcal{R}, P, n, \gamma \rangle$, where S represents the state. $\mathcal{A} = A_1 \times A_2 \times \ldots \times A_n$ is the joint action space, with A_i denoting the action for agent i. Similarly, $\mathcal{O} = O_1 \times O_2 \times \ldots \times O_n$ is the joint observation space, where O_i represents the local observation of agent i. $\mathcal{R} = \{R_i\}$ is the set of rewards, where each $R_i = R_i(S, \mathcal{A})$ defines the reward for agent i. The transition function P specifies the probability of transitioning from one state to another given the joint actions of all agents. The parameter n denotes the number of agents. Finally, $\gamma \in [0, 1]$ is the discount factor, which balances the importance of immediate rewards and long-term objectives.[2]

Existing MARL frameworks typically define each agent as an individual traveler with a discrete action space, where each action corresponds to a specific route choice. As a result, the agent count equals the total number of travelers (i.e., OD demand), leading to S&R issues discussed in Section 1. To address these limitations, we propose recharacterizing each agent as an OD-pair router. However, this redefinition introduces two core challenges:

1) According to the re-defined agent in our MARL framework, a straightforward formulation of the action space should reflect the potential number of travelers choosing a given route. However, if the discrete action space formulation persists, two issues arise: First, the action space dimensions become inconsistent across agents; second, a large number of travelers leads to an excessively large action space. These issues perpetuate the scalability challenges associated with OD-level demand. To address this, we introduce a continuous, simplex-constrained action space (detailed in Section 4.2) supported by a Dirichlet-based policy, which ensures both scalability and smooth policy learning.
2) The relative gap serves as a more accurate measure of convergence, as detailed in Eq. (18) within Section 4.3. However, its conventional computation relies on global traffic information, which is often inaccessible to individual agents in real-world settings. To address this, we propose a localized approximation of the relative gap that can be computed from partial observations (detailed in Section 4.3), which enables effective reward shaping while preserving convergence toward equilibrium.

4. Methodology

4.1. Overview of the reformulated MARL framework

Fig. 1 illustrates the proposed MARL framework for the traffic assignment problem, which is formulated as a DEC-POMDP. Each agent is defined as an OD-pair router, with n corresponding to the number of OD pairs in the network. Agents allocate their OD demand across feasible routes to optimize individual objectives by maximizing accumulated rewards based on local observations.

At each time step t, each agent i receives local observations O_i^t, which include both static and dynamic route features inspired by conventional route choice models:

- **Static features**: origin and destination coordinates, default OD-pair demand, route identifiers, free-flow travel time, number of links, and average link degree.
- **Dynamic features**: marginal travel time, recent changes, travel time history, average volume-to-capacity ratio, number of congested links, and the proportion of demand assigned to each route at the previous step.

[2] For a detailed review of DEC-POMDPs, refer to Oliehoek et al. (2016).

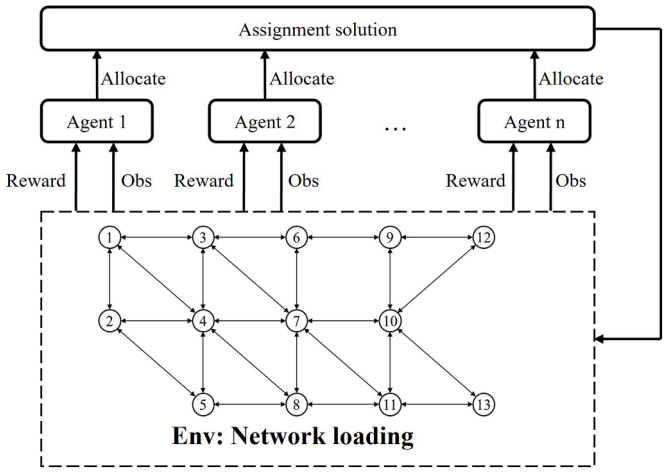


Fig. 1. DEC-POMDP framework for traffic assignment (Obs: local observation).

The joint action $\mathcal{A}_t = (A_1^t, \ldots, A_n^t)$ is then executed for network loading, which computes link costs and flow assignments. After network loading, the environment provides feedback via local rewards R_i^t. The objective of each agent is to maximize its discounted accumulated reward:

$$J_i(\theta_i) = E_{\mathcal{A}_t,\, S_t}\left[\sum_{t=0}^{\infty} \gamma^t\, R_i^t(S_t, \mathcal{A}_t)\right] \tag{5}$$

where θ_i parameterizes the local policy $\pi_{\theta_i}(A_i^t|O_i^t)$.

This section will elaborate on the core design and is organized as follows. Section 4.2 describes the action space design, including a comparison between softmax- and Dirichlet-based policies, and their theoretical implications for scalability and reliability. Then, Section 4.3 introduces the reward function design that aligns local incentives with UE or SO conditions. Finally, the policy learning procedure and implementation details are summarized in Section 4.4.

4.2. Action space design

This section will explain how the proposed action space formulation enhances both scalability and reliability within the MARL-based traffic assignment framework. In particular, the design incorporates two complementary improvements:

- **OD-pair router agents** reduce the agent population from $O(N)$ (number of travelers) to $O(|D|)$ (number of OD pairs), and replace the discrete, combinatorial joint action space with continuous probability simplices. This design not only improves scalability and learning stability, but also enables efficient adaptation to varying OD demand without retraining.
- **Dirichlet-based policy** ensures unbiased sampling over the simplex, allows sparse solutions, and mitigates bias and variance issues common in softmax-based methods.

We will first present the action definition and implementation strategies, and then provide the theoretical justification.

4.2.1. Action definition

To address challenge 1) in Section 3, the action A_i for each agent i is defined as the set of proportions assigned to each available route, denoted as $A_i = \{a_1, a_2, \ldots, a_k\}$, where k is the number of feasible routes for the OD pair. For example, if agent i corresponds to the OD pair $3 \to 11$ with two feasible routes, its action A_i can be $\{0.2, 0.8\}$, denoting the proportion of OD demand allocate to each route.

The new formulation of action space should satisfy the following probability simplex:

$$\sum_{i=1}^{k} a_i = 1, \quad a_i \geq 0, \quad \forall i = 1, 2, \ldots, k \tag{6}$$

4.2.2. Softmax-based policy

To achieve this goal, softmax-based policy is a commonly used alternative. Specifically, the policy network predicts the mean and standard deviation of a Gaussian distribution, from which logits $z = [z_1, z_2, \ldots, z_k]$ are sampled to encourage exploration. These logits are then transformed using the softmax function, as shown in Fig. 2a:

$$a_i = \frac{\exp(z_i)}{\sum_{j=1}^{k} \exp(z_j)}, \quad i = 1, 2, \ldots, k \tag{7}$$

This formulation guarantees $\sum_{i=1}^{k} a_i = 1$ because the softmax function normalizes the exponentials and ensures that each $a_i > 0$ since the exponential function is strictly positive. However, despite its simplicity, the softmax transformation suffers from the translation-invariance problem $\sigma(\mathbf{z} + c\mathbf{1}) = \sigma(\mathbf{z})$, which may introduce estimation bias. Moreover, it cannot represent sparse actions (e.g., $a_i = 0$) because its output is always strictly positive. These limitations reduce the overall reliability and flexibility of the strategy during training, which motivates the Dirichlet-based approach below.

4.2.3. Dirichlet-based policy

The Dirichlet-based policy addresses the limitations above as it inherently satisfies the simplex constraint while enabling sparse and flexible action sampling. Fig. 2b shows the network predicts concentration parameters $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \ldots, \alpha_k]$ using a softplus activation to ensure positivity. Actions are then sampled from a Dirichlet distribution:

$$p(\mathbf{a}|\boldsymbol{\alpha}) = \frac{\Gamma(\alpha_0)}{\prod_{i=1}^{k}\Gamma(\alpha_i)} \prod_{i=1}^{k} a_i^{\alpha_i - 1}, \quad \text{where} \quad \alpha_0 = \sum_{i=1}^{k} \alpha_i \tag{8}$$

where $\mathbf{a} = [a_1, a_2, \ldots, a_k]$ is the action vector satisfying $\sum_{i=1}^{k} a_i = 1$ and $a_i \geq 0$, and $\Gamma(\cdot)$ is the gamma function.

4.2.4. Scalability and reliability analysis of the OD-pair agent design

The following analysis clarifies why combining the OD-pair agent granularity with the proposed action space design jointly improves both scalability and reliability. In existing MARL-based traffic assignment frameworks, each traveler is modeled as an agent that selects a discrete route A_i from its feasible set. This leads to a joint action space $\mathcal{A} = A_1 \times A_2 \times \cdots \times A_N$ with size $|\mathcal{A}| = k^N$ for N travelers and k feasible routes. The worst-case growth rate is therefore exponential: $O(k^N)$. While this defines a finite potential game that guarantees the existence of at least one pure strategy Nash equilibrium (Monderer and Shapley, 1996), the combinatorial nature severely limits scalability as N increases.

Scalability w.r.t. N. This combinatorial explosion not only increases computational burden but also causes high non-stationarity in decentralized learning, as each agent's payoff is influenced by the joint actions of all others. The exponential growth in joint action space renders learning unstable and inefficient in large-scale settings.

By contrast, the OD-pair router design reduces the agent population from $O(N)$ to $O(|D|)$, where $|D|$ is the number of OD pairs in the network. Its action space for each agent is a continuous probability simplex of fixed dimension $(k-1)$, independent of the total demand. Since $|D|$ is typically orders of magnitude smaller than N in urban networks, this reformulation substantially improves scalability by reducing both the number of learning agents and the dimensionality of their action space.

For example, in the SiouxFalls and Anaheim networks used in our experiments (see Table 2 and Fig. 3), the total traveler demand reaches over 360,000 in SiouxFalls and more than 100,000 in Anaheim, while the number of OD pairs remains under 1500. This demonstrates a clear $O(|D|)$ vs. $O(N)$ difference in practice. Reducing the agent count by two

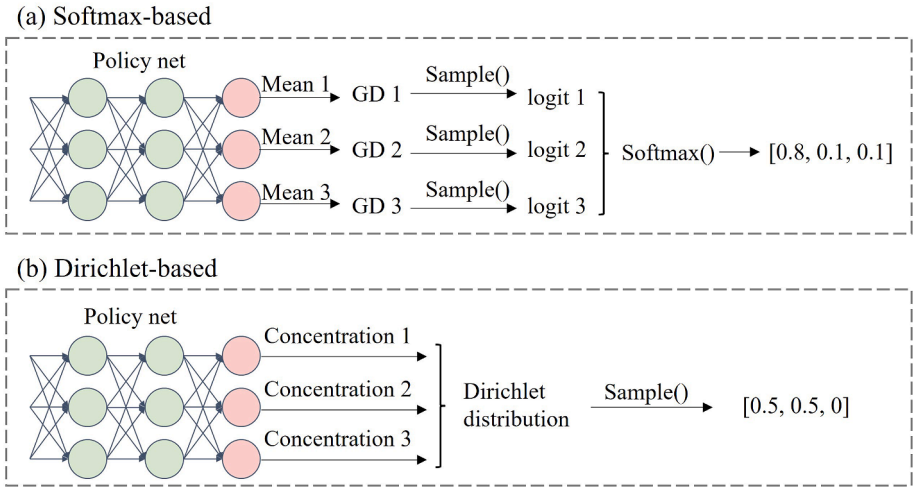


Fig. 2. Softmax-based and Dirichlet-based policy. (GD: Gaussian distribution).

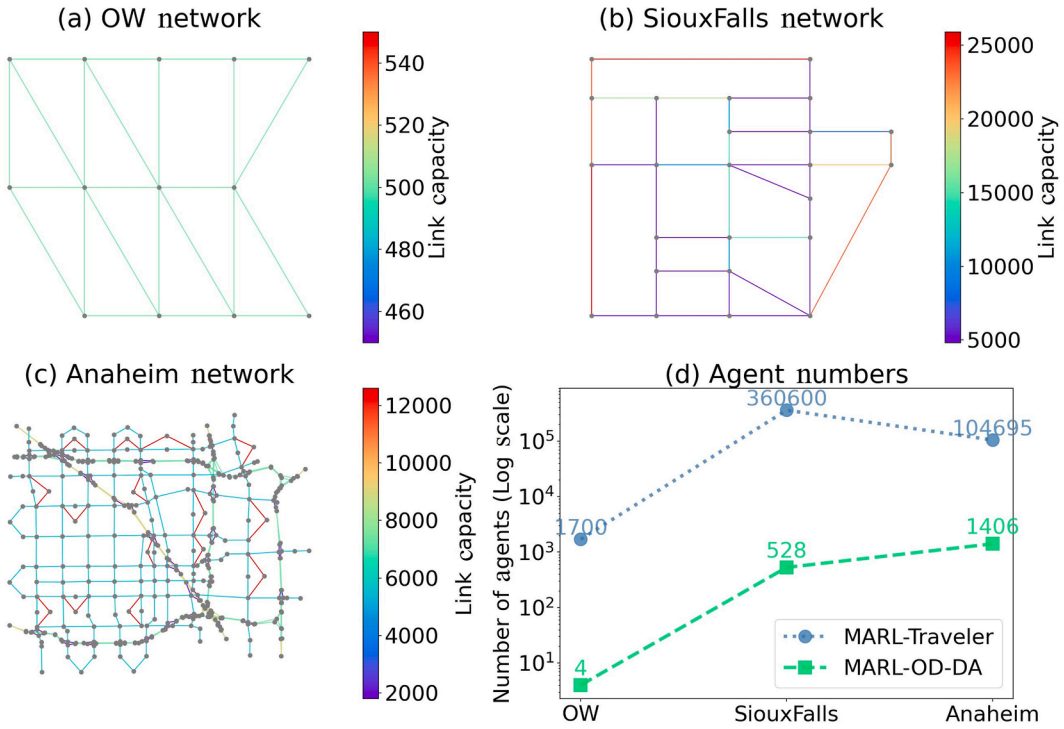


Fig. 3. Urban transportation networks and corresponding agent numbers.

orders of magnitude not only simplifies policy learning but also drastically lowers the memory usage of neural networks and replay buffers, making the proposed framework feasible for real-world city-scale scenarios on standard hardware.

Scalability w.r.t. demand variation. Additionally, traveler-level MARL assumes that the number of agents equals the total demand, which makes it rigid when real-world demand fluctuates. If the aggregate demand changes, the fixed agent set must be redefined, retrained, or manually scaled. The OD-pair router design overcomes this by modeling each agent's action A_i as a continuous probability vector allocating its entire OD demand d_{rs} over k feasible paths, satisfying the simplex constraint in Eq. (6). This means that only the demand magnitude changes, while the agent structure remains stable, making the framework inherently adaptive to varying flow conditions without retraining the agent set.

Reliability. The traveler-level design creates a non-convex feasible region composed of discrete combinations of pure strategies. For example, with four travelers and two available routes, both combinations (A, A, B, B) and (A, B, A, B) result in the same aggregate flow pattern [2, 2]. While the system-level flow is identical, the agent-level solutions differ, introducing redundant equilibria. This redundancy amplifies the risk of oscillations in policy learning, as multiple agent-level combinations can produce the same aggregate outcome but require the learning dynamics to stabilize on just one. As N grows, the likelihood of non-stationarity and fragile convergence will increase.

By contrast, each OD-pair router selects a continuous probability vector that lies on a probability simplex (Eq. (6)). This makes the agent-level action set continuous and convex, which in turn yields a smoother optimization landscape for policy learning. When link cost functions $c_e(\cdot)$ are strictly increasing, the Beckmann Eq. (1) ensures the uniqueness of the Wardrop UE, contributing to more reliable policy learning under the DEC-POMDP setting.

Both traveler-level and OD-pair router formulations satisfy the fundamental constraints in Eqs. (2)–(4). However, only the OD-pair design retains convexity at the agent level, making the theoretical guarantees of uniqueness and stable convergence more practically relevant.

In summary, redefining agents as OD-pair routers transforms the system from a discrete, combinatorial joint action space of size $O(k^N)$ to a continuous one composed of $|D|$ probability simplices. This reformulation, along with convex feasibility and adaptability to varying demand, effectively addresses the core scalability and reliability challenges faced by traveler-level MARL frameworks, as summarized in Table 1.

4.2.5. Theoretical reliability analysis of the dirichlet-based policy

Here we will formalize the theoretical advantages of the Dirichlet-based policy over the widely used softmax-based policy. Unlike the softmax-based policy, which introduces estimation bias and suffers from unstable gradient variance, the Dirichlet-based policy directly samples from the probability simplex (Eq. (6)), thus providing unbiased policy gradient estimates under the standard likelihood-ratio (REINFORCE) estimator (Williams, 1992), and naturally allowing sparse routing decisions. These properties contribute to more stable convergence and higher learning reliability for each OD-pair router agent.

The following propositions will justify these claims, with full proofs in the Electronic Supplementary Material. Throughout this subsection, "bias" and "variance" refer to those of the score-function policy gradient estimator:

$$\widehat{g}(\theta) = \mathbb{E}_{s\sim d^{\pi},\ a\sim\pi_\theta}[\nabla_\theta \log \pi_\theta(a|s)\, A^{\pi}(s,a)] \tag{9}$$

relative to the true gradient $\nabla_\theta J(\theta)$.

Proposition 1. **(*Bias of softmax-based policy*)** *Consider a softmax-based policy that first samples latent logits* $\mathbf{z} \sim \mathcal{N}(\boldsymbol{\mu}, \sigma^2 I)$ *and then maps them to the simplex via* $a_i = \exp(z_i)/\sum_j \exp(z_j)$. *Because of the translation-invariance*:

$$\sigma(\mathbf{z} + c\mathbf{1}) = \sigma(\mathbf{z}), \quad \forall c \in \mathbb{R} \tag{10}$$

the mapping $\mathbf{z} \leftrightarrow \mathbf{a}$ *is non-injective*: *each action* $\mathbf{a}$ *corresponds to an infinite ray* $\{\mathbf{z} + c\mathbf{1}\}$. *The true policy density therefore marginalizes over an unobserved shift variable* c, *but the standard estimator ignores this redundancy*. *Consequently*,

$$\mathbb{E}_{\mathbf{z}}[\nabla_{\boldsymbol{\mu}} \log\, \pi(\mathbf{a}|\mathbf{z})\, A(s,\mathbf{a})] \neq \nabla_{\boldsymbol{\mu}} J(\boldsymbol{\mu}) \tag{11}$$

i.e., the estimator is biased when the redundancy is not explicitly accounted for.

Proposition 2. **(*Variance blow-up in softmax-based policy*)** *For the softmax-based policy above, the Fisher information matrix with respect to the latent Gaussian mean satisfies* $F(\boldsymbol{\mu}) \propto I/\sigma^2$. *As* $\sigma \to 0$ (*the policy becomes nearly deterministic*), *the variance of the policy gradient estimator diverges unless the step size is scaled accordingly, leading to unstable training.*

Table 1
Comparison of traveler-level and OD-pair router MARL designs.

Aspect	Traveler-level	OD-pair router
Agent granularity	$O(N)$	$O(\|D\|)$
Agent action A_i	Discrete choice among k routes	$(k-1)$-dimensional continuous vector
Joint action space $\mathcal{A}$	k^N discrete combinations	Product of $\|D\|$ simplices
Feasibility region	Non-convex	Convex
Scalability w.r.t. N	Limited for large N	Scales with OD-pair count
Scalability w.r.t. demand	Fixed agents, inflexible	Adapts to varying demand
Reliability	High non-stationarity	Unique equilibrium, stable learning

Table 2
Detailed information of the urban transportation networks.

Network	Zone	Link	Node	OD-pair	OD-demand
OW	13	48	13	4	1700
SiouxFalls	24	76	24	528	360,600
Anaheim	38	914	416	1406	104,694.4

Proposition 3. **(*Unbiasedness of dirichlet-based policy*)** *A Dirichlet-based policy samples actions directly from* $\mathrm{Dir}(\boldsymbol{\alpha})$, *which is fully supported on the simplex. Since no redundant latent shift is introduced, the likelihood-ratio estimator coincides with the true gradient*:

$$\mathbb{E}_{\mathbf{a}\sim\mathrm{Dir}(\boldsymbol{\alpha})}[\nabla_{\boldsymbol{\alpha}} \log \pi(\mathbf{a}|\boldsymbol{\alpha})\, A(s,\mathbf{a})] = \nabla_{\boldsymbol{\alpha}} J(\boldsymbol{\alpha}) \tag{12}$$

Thus the estimator is unbiased.

Proposition 4. **(*Variance control in dirichlet-based policy*)** *For* $\mathbf{a} \sim \mathrm{Dir}(\boldsymbol{\alpha})$, *the variance of each component is*

$$\mathrm{Var}[a_j] = \frac{\alpha_j(\alpha_0 - \alpha_j)}{\alpha_0^2(\alpha_0 + 1)}, \quad \alpha_0 = \sum_{j=1}^{k} \alpha_j \tag{13}$$

As $\alpha_0 \to \infty$, *the variance decreases smoothly to zero, and the corresponding Fisher information remains bounded, ensuring stable gradient behavior near determinism.* Remark (Sparse expressiveness). Unlike the softmax-based policy, which always outputs strictly positive probabilities, the Dirichlet policy can concentrate mass arbitrarily close to simplex boundaries by letting certain $\alpha_j \to 0^+$. This enables near-sparse routing decisions when optimal behavior excludes suboptimal paths, without sacrificing the validity of gradient estimates.

4.2.6. Action pruning for sparse outputs

Challenges also arise when employing the Dirichlet-based policy in policy-based reinforcement learning methods. The policy is optimized by computing the gradient of the expected return $J(\theta)$, defined as

$$\nabla_\theta J(\theta) = E_{\pi_\theta}[\nabla_\theta \log \pi_\theta(a|s) \cdot \widehat{A}(s,a)] \tag{14}$$

where $\log\, \pi_\theta(a|s)$ represents the log-probability of the policy, s denotes the state, and $\widehat{A}(s,a)$ is the advantage function. For the Dirichlet-based policy, the log-probability of a sampled action a is given by

$$\log\, p(a|\boldsymbol{\alpha}) = \sum_{i=1}^{k} (\alpha_i - 1) \log a_i + \log\, \Gamma(\alpha_0) - \sum_{i=1}^{k} \log\, \Gamma(\alpha_i) \tag{15}$$

where $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \ldots, \alpha_k]$ are the concentration parameters and $\alpha_0 = \sum_{i=1}^{k} \alpha_i$.

If any sub-action $a_i = 0$, the term $\log\, a_i \to -\infty$, rendering the gradient computation undefined and disrupting the policy optimization process. Consequently, even the Dirichlet-based policy cannot truly achieve sub-actions with exact zero values. Instead, small sub-actions below a predefined threshold (e.g., 10^{-7}) are clipped to the threshold to prevent this issue.

In traffic assignment, achieving better performance often requires certain sub-actions to be exactly zero. However, this constraint can lead the policy network to excessively reduce logits or concentration values for these sub-actions, potentially resulting in overfitting, a problem similar to the challenges observed in "Label Smoothing" (Szegedy et al., 2016).

To address this issue, we propose action pruning, inspired by the principles of "Label Smoothing". This technique modifies the action vector in two steps. First, any sub-actions smaller than a predefined threshold τ (e.g., 10^{-4}) are explicitly set to zero:

$$a_i'' = \begin{cases} 0, & \text{if } a_i < \tau \\ a_i, & \text{otherwise} \end{cases} \tag{16}$$

Next, the remaining sub-actions are normalized to ensure they satisfy the probability simplex constraint:

$$a_i^{'} = \frac{a_i^{''}}{\sum_{j=1}^{K} a_j^{''}} \tag{17}$$

This approach enforces sparsity by explicitly eliminating negligible sub-actions while keeping the action vector valid on the probability simplex. Since the adjustment is applied externally after action sampling and does not alter the policy network or its outputs, it does not interfere with gradient-based optimization. Although this introduces a mild approximation during action execution, it serves as a practical heuristic that improves training stability with negligible impact on learning performance.

4.3. Relative-gap-based reward function

To address challenge 2) in Section 3, the reward is derived from the concept of the relative gap, a widely used metric for evaluating convergence in traffic assignment (Boyles et al., 2020). The relative gap quantifies the deviation of the current solution from an optimal state, reaching zero when the flow conditions satisfy either SO or UE. In the UE, it is specified as

$$\omega = \frac{\sum_{e\in E} c_e(x_e)\cdot x_e}{\sum_{(r,s)\in D} k^{rs} d_{rs}} - 1 \tag{18}$$

where ω is the relative gap, E is the road segments (edges), x_e is the flow on road segment e, d_{rs} is the OD demand between the OD pair (r, s), $c_e(x_e)$ is the performance function which maps the flow x_e on the segment to its corresponding travel time, and k^{rs} is the travel cost on the shortest path for OD pair (r, s).

This metric, here referred to as the global relative gap, measures the assignment performance across the entire network. Unless otherwise specified, the term relative gap in this paper refers to the global relative gap. For individual agent i, the metric is adapted to a local relative gap based on the flow and demand associated with the agent's OD pair (r_i, s_i):

$$\omega_i = \frac{\sum_{p\in P_{r_i s_i}} c_p \cdot f_p}{k^{r_i s_i} d_{r_i s_i}} - 1 \tag{19}$$

where ω_i is the local relative gap for agent i, $P_{r_i s_i}$ is the set of all available paths for OD pair (r_i, s_i), c_p is the travel cost for path p, f_p is the flow assigned to path p, $k^{r_i s_i}$ is the travel cost on the shortest path, and $d_{r_i s_i}$ is the demand for (r_i, s_i).

The reward for each agent is based on the change in its local relative gap between consecutive time steps. At step t, the reward for agent i is defined as

$$R_i^t = \begin{cases} -\omega_i^1 & \text{if } t = 1 \\ \omega_i^{t-1} - \omega_i^t & \text{if } t > 1 \end{cases} \tag{20}$$

where R_i^t is the reward for agent i at time t, and ω_i^t is the local relative gap for agent i at time t.

By aligning each agent's reward directly with the reduction in its local relative gap, this design provides each agent with an intuitive and consistent signal for improving its routing decisions. This local feedback helps ensure that decentralized updates remain coordinated, reducing the risk of conflicting policies and supporting more stable convergence within the MARL framework.

4.4. Policy learning via PPO

To optimize each OD-pair router's local policy $\pi_{\theta_i}(A_i^t|O_i^t)$ within the DEC-POMDP framework, we adopt proximal policy optimization (PPO) (Schulman et al., 2017), which is widely used for its robust stability and simplicity in continuous action spaces. In our setting, each agent operates independently with partial observations; therefore, we employ the decentralized variant, independent PPO (IPPO) (Yu et al., 2022), which aligns naturally with the proposed OD-pair router design and scales efficiently with the number of OD pairs.

In practice, the policy network uses a shared encoder to extract both static and dynamic route features, followed by route-level output heads for either softmax logits or Dirichlet concentration parameters. Entropy regularization is included to encourage exploration and prevent premature policy collapse.

The clipped surrogate objective for each agent i is defined as

$$\begin{aligned} \mathcal{L}_i^{\text{PPO}}(\theta_i) &= \mathbb{E}_t[\min(r_t(\theta_i)\widehat{A}_t,\ \text{clip}(r_t(\theta_i), 1-\epsilon, 1+\epsilon)\widehat{A}_t)], \quad r_t(\theta_i) \\ &= \frac{\pi_{\theta_i}(A_i^t|O_i^t)}{\pi_{\theta_i^{\text{old}}}(A_i^t|O_i^t)} \end{aligned} \tag{21}$$

where ϵ is the clipping threshold and $\widehat{A}_t$ denotes the estimated advantage function.[3]

Algorithm 1 summarizes the training procedure for reproducibility.

Algorithm 1. IPPO-based training with parameter sharing.

```
Initialize shared policy parameters θ.
for each episode do
  Generate OD demand for the episode.
  for t = 0 to T do
    for each agent i do
      Observe O_i^t and sample action A_i^t ~ π_θ(A_i^t|O_i^t).
    end for
    Perform network loading with joint action A_t and obtain local rewards {R_i^t}.
    Store (O_i^t, A_i^t, R_i^t) for all i in replay buffer.
  end for
  Compute advantage estimates Â_t from stored trajectories.
  Update shared parameters θ by maximizing the clipped PPO objective L^PPO(θ).
end for
```

5. Experiment

5.1. Experiment settings

5.1.1. Urban transportation network

The proposed method is evaluated on three widely used urban transportation networks: the Ortuzar and Willumsen (OW) network, the SiouxFalls network, and the Anaheim network (de Dios Ortúzar and Willumsen, 2024; Liu and Meidani, 2024, 2025; Transportation Networks for Research Core Team, 2024). Their topologies, link capacities, and corresponding agent numbers are shown in Fig. 3, with detailed statistics provided in Table 2. For all networks, each agent's available route set is derived using the k-shortest path algorithm (Yen, 1971) or traditional traffic assignment methods, with a maximum of six routes (Boyles et al., 2020).

The OW network, classified as small-sized, is used to compare the proposed algorithm with an existing MARL-based method. In contrast, SiouxFalls and Anaheim are medium-sized networks, with SiouxFalls representing a city-level primary road network due to its high total OD demand (360,600 travelers), and Anaheim reflecting more complex urban systems with its larger number of nodes and links.

A widely adopted non-linear performance function, the Bureau of Public Roads (BPR) function, is used to calculate the travel time on a given road segment (Boyles et al., 2020):

[3] In practice, we compute $\widehat{A}_t$ using Generalized Advantage Estimation (GAE) (Schulman et al., 2015), which measures how much better the chosen action is compared to a baseline value estimate.

$$c_e(x_e) = t_0^e\left(1 + 0.15\left(\frac{x_e}{w_e}\right)^4\right) \tag{22}$$

where t_0^e denotes the free-flow travel time on e and w_e represents the capacity of road segment e.

To validate the model's stability, variable OD demand is generated by scaling the default OD demand for each OD pair using a random scaling factor, following approaches commonly adopted in previous studies (Liu and Meidani, 2024, 2025; Rahman and Hasan, 2023). The scaled demand is expressed as

$$d_{rs} = \beta_{rs} \cdot d_{rs}^{\text{default}} \tag{23}$$

where d_{rs}^{default} is the default OD demand between origin r and destination s, and β_{rs} is a uniformly distributed random scaling factor. For the OW and SiouxFalls networks, $\beta_{rs} \sim U(0.5, 1)$, as their default OD demands represent peak flow under congested conditions. In contrast, the Anaheim network uses $\beta_{rs} \sim U(0.5, 1.5)$, as its default OD demand reflects less congested conditions, allowing for better representation of traffic fluctuations between moderate and peak volumes, aligning with more realistic dynamics. This variability reflects day-to-day fluctuations across different simulation runs, with demand being time-aggregated rather than time-dependent within each assignment instance.

5.1.2. Hyperparameter settings

The hyperparameters for the MARL methods were optimized using grid search. The core settings are summarized in Table 3.

5.1.3. Evaluation metrics

The primary metric is the relative gap, which quantifies the deviation from the optimal solution—a smaller value indicates better performance. For instance, a relative gap of 0.001 roughly represents a 0.1% deviation. Additionally, the episode average reward is used to assess training performance.

5.1.4. Baselines

Conventional methods. In this study, the proposed MARL framework is compared with both conventional and widely used MARL-based methods. Specifically, the conventional methods encompass:

- **Method of successive averages (MSA).** An iterative algorithm that updates link flows using a weighted average of previous iterations and current shortest path flows (Mounce and Carey, 2015).
- **Frank-wolfe algorithm (FW).** An enhanced version of MSA that employs an adaptive step size and has been widely adopted in practice (Boyles et al., 2020).

MARL-based methods. All evaluated MARL-based methods use the IPPO algorithm for training agents' policies. The methods under consideration comprise:

- **MARL-traveler.** An altered adaptation of the frameworks as proposed in Shou et al. (2022) and Zhou et al. (2020) has been implemented, originating from their state-reward design, given the lack of open-source availability. Travelers are modeled as agents with OD locations as observations, route choices as discrete actions, and negative travel costs as rewards. This implementation is developed for comparison due to the lack of open-source code in this domain. This serves as a representative baseline for comparison.
- **MARL-OD-S.** The proposed MARL formulation using a softmax-based policy. This variant is used as an ablation baseline to assess the impact of replacing the Dirichlet-based policy.
- **MARL-OD-SA.** The proposed MARL formulation using a softmax-based policy with action pruning. This variant is designed to evaluate the effectiveness of action pruning method in softmax-based policy.
- **MARL-OD-D.** The proposed MARL formulation using a Dirichlet-based policy. This variant is used to isolate the effect of the Dirichlet-based compared to softmax-based policy.
- **MARL-OD-DA (Ours).** The proposed full model combining Dirichlet-based policy with action pruning.

Table 3
Hyperparameter settings for MARL training.

Hyperparameter	Value
Hidden size	512
Number of layers	3
Learning rate	4×10^{-5}
Clipping threshold, ϵ	0.2
Discount factor, γ	0.75
GAE, λ	0.95
Mini-batch size	512
Steps per episode, T	50
Rollout workers	5
Pruning threshold, τ	10^{-4}

5.2. Experiment results

The proposed MARL framework is validated from three perspectives, namely effectiveness, scalability, and reliability. Effectiveness refers to the framework's ability to consistently achieve high-quality solutions under controlled settings, as evaluated on the OW network in Section 5.2.1. Scalability assesses how well the method generalizes to larger and more complex networks with variable OD demand, and is examined in Section 5.2.2. Reliability focuses on convergence behavior and robustness under training and practical deployment conditions. It is analyzed from multiple aspects: convergence trends under increasing network complexity (Section 5.2.3), the stability and impact of action pruning (Section 5.2.4), and comparative performance against conventional optimization methods (Section 5.2.5).

5.2.1. Effectiveness validation based on small-sized network (OW network)

Fig. 4a illustrates the reduction in the relative gap over training steps for the OW network under two demand scenarios: Fixed OD demand, where the demand remains constant as default, and variable OD demand, which evolves as defined in Eq. (23). The existing method, MARL-Traveler, is limited to learning under fixed demand scenarios, whereas the proposed method demonstrates significant advantages. Specifically, the proposed method, MARL-OD-DA, achieves a significantly smaller relative gap in both scenarios. Under fixed demand conditions, it not only converges to the lowest relative gap but also maintains strong training stability. Notably, under variable demand conditions, the relative gap achieved by MARL-OD-DA even surpasses the best performance of MARL-Traveler under fixed demand. Note that this comparison is conducted under the UE objective. Since the two methods adopt different reward designs, the episode relative gap to UE is used as a consistent performance metric.

5.2.2. Scalability and reliability validation based on medium-sized networks (SiouxFalls and Anaheim networks)

This experiment validates the framework's scalability by testing on larger networks with variable OD demand, while the stable convergence also demonstrates its reliability in handling non-stationary, large-scale scenarios. Fig. 3d compares the agent numbers for MARL-Traveler and MARL-OD-DA across the three networks, showing that the proposed framework reduces agent numbers by at least two orders of magnitude. MARL-Traveler is impractical for SiouxFalls and Anaheim networks due to its excessive memory requirements, stemming from the large number of agent networks and replay buffers, which exceed the GPU memory capacity on the experimental hardware (NVIDIA GeForce RTX 3090 Ti). In contrast, the proposed MARL framework demonstrates better

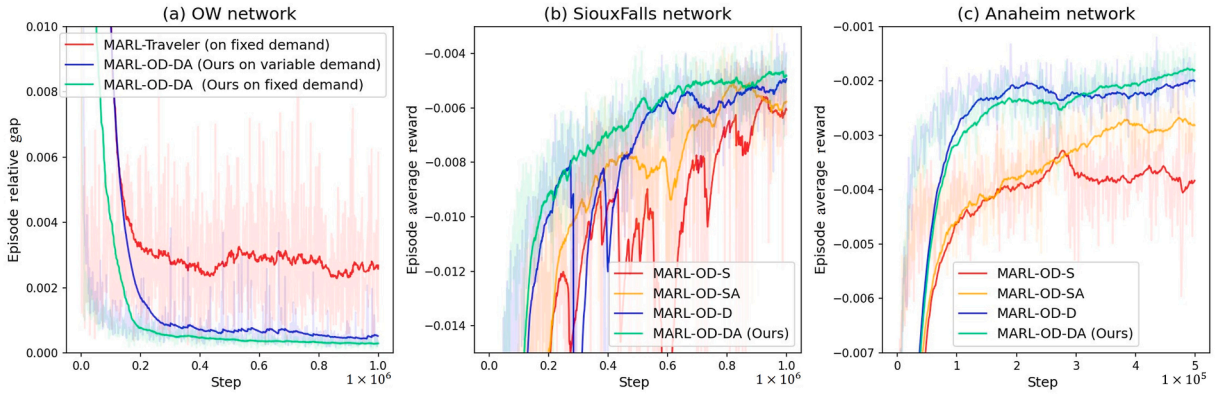


Fig. 4. Training curves across three transportation networks.

scalability and can be applied to these networks.

Figs. 4b and 4c compare the variants of the proposed MARL method (Dirichlet-based vs. softmax-based policies) on medium-sized networks under a consistent reward structure, indicating that the proposed Dirichlet-based policy (MARL-OD-D) outperforms the softmax-based policy (MARL-OD-S). Table 4 further validates the effectiveness of the trained agents in addressing variable OD demand scenarios. The proposed method, MARL-OD-DA, achieves minimum relative gaps of 0.001702 and 0.000869 on the SiouxFalls and Anaheim networks after 50 steps (averaged over five experiments), corresponding to deviations of 0.1702% and 0.0869% from the optimal solution. These results demonstrate the framework's scalability to medium-sized networks, its stability in handling variable OD demand, and its reliability in producing high-quality solutions. Note that the experiments on SiouxFalls and Anaheim are conducted under the SO objective, confirming the flexibility of the proposed framework beyond the UE setting.

5.2.3. *Convergence behavior across networks*

Fig. 4 shows that the proposed MARL-OD-DA framework converges faster than all MARL baselines across tested networks, in terms of both episode average reward and relative gap. While this indicates superior learning efficiency, further investigation is conducted to examine how convergence behavior varies across networks of different scales and structural properties.

As summarized in Table 5, the total convergence time is computed as the product of the training time per 10^5 steps and the estimated convergence steps, where the latter is approximated from Fig. 4 based on when the episode average reward or relative gap stabilizes. In general, training time increases with network size and the number of OD-pairs. However, SiouxFalls requires more steps to converge than Anaheim despite being smaller, indicating that convergence efficiency is not solely determined by network scale or agent count.

To explain this behavior, the average number of OD-pairs per link is analyzed as a proxy for route-level competition and inter-agent coupling. This metric is defined as the ratio of total OD-pairs to total links in the network. A higher value suggests that more agents are likely to share common road segments, intensifying competition and hindering stable policy learning. As shown in Table 5, the SiouxFalls network exhibits the highest average value, indicating denser agent interactions and more complex coordination. Such routing overlap increases the likelihood of policy interference, leading to degraded learning stability and slower convergence. In contrast, the Anaheim network, despite its larger size, offers greater route diversity and link capacity, reducing route contention. The smallest and simplest OW network demonstrates the fastest and most stable convergence.

These results suggest that while training cost generally increases with agent count and network scale, the convergence efficiency and final solution quality are more strongly associated with routing density, as captured by the average number of OD-pairs per link. As illustrated in Fig. 5, this metric shows a strong positive correlation with the minimum relative gap achieved across networks (refer to Table 4).

Table 4
Minimum relative gap in 50 steps (OW-fixed: fixed OD demand; others: variable OD demand).

Method	OW-fixed	OW	SiouxFalls	Anaheim
MARL-Traveler	0.001587	—	—	—
MARL-OD-S	0.000418	0.001082	0.005758	0.001864
MARL-OD-SA	0.000370	0.000839	0.003266	0.001586
MARL-OD-D	0.000356	0.000651	0.003903	0.001067
MARL-OD-DA (Ours)	0.000218	0.000407	0.001702	0.000869

Table 5
Training convergence statistics across different networks.

Metric	OW	SiouxFalls	Anaheim
Time per 10^5 steps (h)	0.3	8.9	19.6
Estimated steps to convergence (10^5)	3	8	4
Total convergence time (h)	0.9	71.2	78.4
OD-pairs per link	0.0833	6.9474	1.5382

5.2.4. *Reliability validation of action pruning*

Fig. 4b and c also demonstrate that both Dirichlet-based and softmax-based policies benefit from the incorporation of action pruning. With action pruning, these variants generally exhibit improved training stability and faster convergence compared to their non-pruned counterparts, with the exception of one case on the Anaheim network where the performance remains comparable.

Table 4 further validates the advantages of action pruning during deployment. For example, MARL-OD-DA achieves a relative gap of

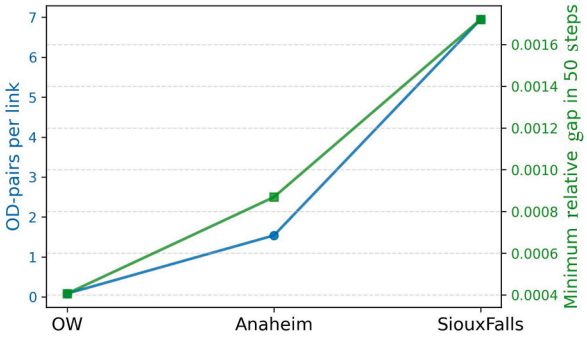


Fig. 5. Relationship between average OD-pairs per link and MARL convergence.

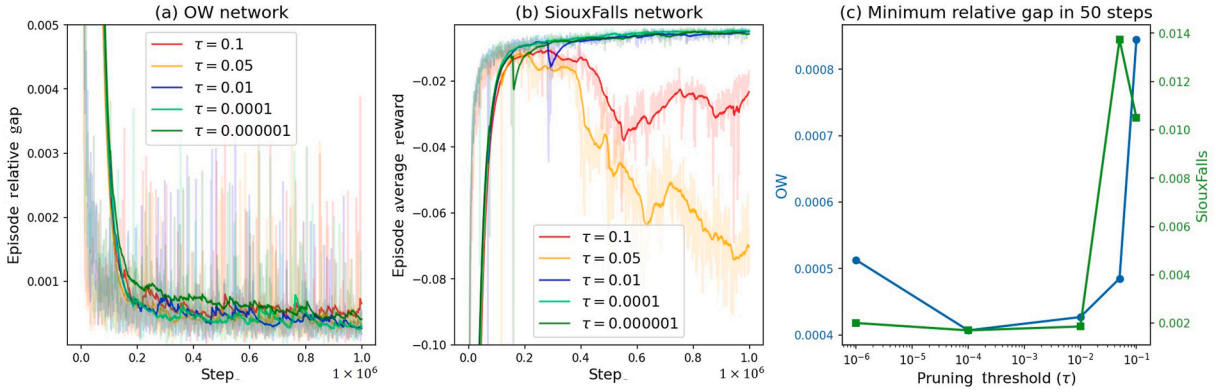


Fig. 6. Sensitivity analysis of the pruning threshold, τ.

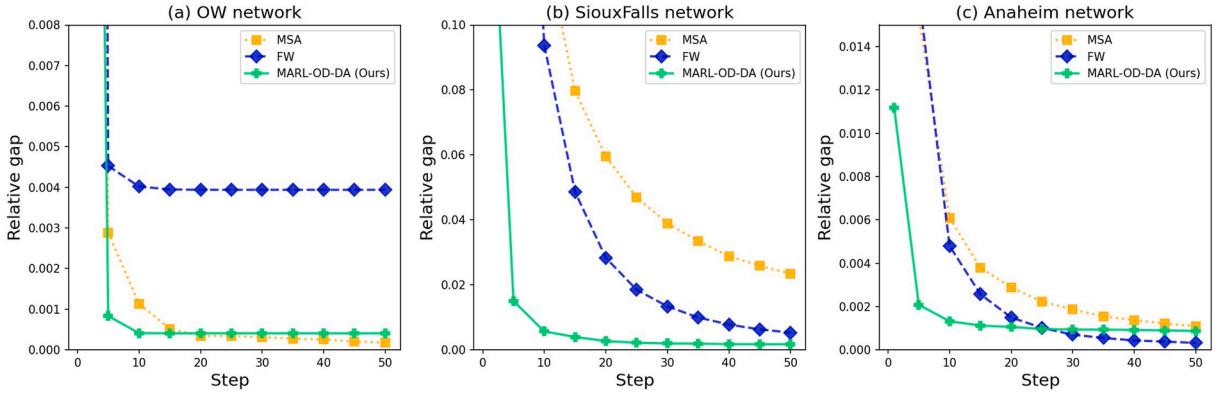


Fig. 7. Performance comparison of trained agents and conventional methods across three transportation networks.

0.001702 on the SiouxFalls network, significantly lower than the 0.003903 reported by MARL-OD-D, corresponding to a reduction of approximately 55.9%. These results highlight the role of action pruning in improving model reliability and convergence quality.

To further assess the robustness of the proposed action pruning mechanism, a sensitivity analysis is conducted on the pruning threshold parameter τ, based on the MARL-OD-DA framework. Specifically, five different threshold values $\tau \in \{10^{-1}, 5 \times 10^{-2}, 10^{-2}, 10^{-4}, 10^{-6}\}$ are evaluated on both the OW and SiouxFalls networks. The corresponding training curves are shown in Fig. 6a and b, while Fig. 6c reports the minimum relative gap observed during deployment within a step budget of 50.

The results indicate that large thresholds (e.g., $\tau = 10^{-1}, 5 \times 10^{-2}$) significantly hinder convergence, especially in the SiouxFalls network, likely due to premature removal of viable actions and insufficient policy flexibility. On the other hand, overly small thresholds (e.g., $\tau = 10^{-6}$) reduce the effect of pruning, leading to a larger effective action space that may increase training difficulty. Optimal performance is achieved with moderate thresholds in the range $\tau \in [10^{-4}, 10^{-2}]$, where a favorable balance is observed between training stability and solution quality.

5.2.5. *Reliability validation via comparison with conventional methods*

Fig. 7 compares the proposed MARL method, MARL-OD-DA, with conventional traffic assignment methods, which assume global state knowledge, in terms of solution quality across steps during deployment. For MARL, a step consists of two sub-steps: (1) Agents make decisions based on their local observations at step t, and (2) the environment updates by simulating or calculating travel times to produce the state at step $t+1$. In contrast, a step in conventional methods involves three sub-steps: (1) computing shortest paths for all OD pairs, (2) optimizing the current assignment along those paths, and (3) simulating or calculating travel times, which aligns with MARL's second sub-step. Among these, shortest path computation in conventional methods becomes increasingly computationally intensive as network size grows, while optimization is relatively fast. Both methods share the same final sub-step of simulating or calculating travel times, which can be highly time-consuming in large networks or when complex traffic flow models or simulation tools are used. This makes the step budget critical for near-real-time applications, where fewer steps are preferred to achieve high-quality solutions.

Fig. 7 shows MARL-OD-DA achieves higher-quality solutions in fewer steps compared to conventional methods across all three networks. For example, on the SiouxFalls network, MARL-OD-DA achieves a relative gap of approximately 0.004626 after 10 steps, while the best-performing conventional method, FW, obtains a relative gap of 0.092301, representing an improvement of approximately 94.99%. This demonstrates MARL-OD-DA's efficiency in providing superior solutions in fewer steps, a key advantage for real-time applications. Moreover, unlike conventional methods, MARL does not require global state knowledge, making it more suitable for real-world deployment.

While our method achieves higher-quality solutions with fewer steps, conventional approaches such as FW can eventually catch up given more steps. This is because conventional methods are deterministic numerical optimization algorithms, where each step guarantees a decrease in the objective value. In contrast, MARL relies on neural networks, and fitting errors can cause performance to plateau once a certain accuracy threshold is reached.

6. Conclusions

This study proposes a novel MARL paradigm for the traffic assignment problem by redefining agents as OD-pair routers. This reformulation reduces the number of learning agents from $O(N)$ to $O(|D|)$, resulting in a reduction by at least two orders of magnitude in practical scenarios. It preserves convexity and enables efficient adaptation to demand variation, thereby significantly improving scalability. To further enhance the reliability, the framework incorporates a Dirichlet-based policy, an action pruning mechanism, and a reward structure based on the relative gap. These components collectively improve policy flexibility, stabilize training, and encourage efficient convergence toward optimal assignments. Experiments on three transportation networks demonstrate that MARL-OD-DA consistently outperforms existing methods in convergence speed and assignment quality. Unlike existing MARL-based methods limited to small networks (with up to 70 nodes, 2100 travelers) and fixed demand, the proposed framework is validated on networks with up to 416 nodes and over 300,000 travelers under varying demand scenarios, demonstrating its applicability to medium-scale urban systems. On the SiouxFalls network, the trained agents achieve convergence within 10 iterations during deployment, reducing the relative gap by 94.99% compared to conventional baselines. These results suggest that the framework is well-suited for time-sensitive applications such as adaptive routing, real-time demand management, and policy evaluation in intelligent transportation systems.

A key limitation of this work lies in the use of a trackable, analytical traffic assignment model, which facilitates theoretical analysis but does not fully capture the complexity of real-world traffic dynamics. Future research will extend the framework to high-fidelity simulation environments and larger-scale urban applications.

CRediT authorship contribution statement

Leizhen Wang: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Formal analysis, Data curation, Conceptualization. **Peibo Duan:** Writing – review & editing, Writing – original draft, Supervision, Resources, Methodology, Funding acquisition, Conceptualization. **Cheng Lyu:** Writing – review & editing, Writing – original draft, Visualization, Methodology, Conceptualization. **Zewen Wang:** Writing – review & editing, Writing – original draft, Validation, Methodology. **Zhiqiang He:** Writing – review & editing, Software, Methodology. **Nan Zheng:** Writing – review & editing, Methodology, Conceptualization. **Zhenliang Ma:** Writing – review & editing, Writing – original draft, Supervision, Resources, Methodology, Funding acquisition, Formal analysis, Conceptualization.

Replication and data sharing

The code and Electronic Supplementary Material to a public GitHub repository are available at https://github.com/georgewanglz2019/MARL4TA.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The work was supported by start-up funds from Monash University (Nos. MSRI8001004 and MSRI9002005), as well as by the TRENoP center and Digital Futures at KTH Royal Institute of Technology, Sweden.

Electronic Supplementary Material

Supplementary data to this article can be found online at https://doi.org/10.1016/j.commtr.2025.100225.

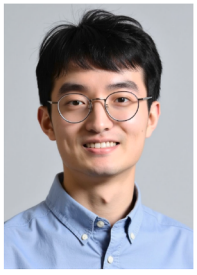
References

Bar-Gera, H., 2002. Origin-based algorithm for the traffic assignment problem. Transp. Sci. 36, 398–417.
Bazzan, A.L., Grunitzki, R., 2016. A multiagent reinforcement learning approach to en-route trip building. In: 2016 International Joint Conference on Neural Networks (IJCNN), pp. 5288–5295.
Boyles, S.D., Lownes, N.E., Unnikrishnan, A., 2020. Transportation Network Analysis. John Wiley & Sons, Hoboken, USA.
Chen, X., Liu, Z., Zhang, K., Wang, Z., 2020. A parallel computing approach to solve traffic assignment using path-based gradient projection algorithm. Transport. Res. C Emerg. Technol. 120, 102809.
de Dios Ortúzar, J., Willumsen, L.G., 2024. Modelling Transport. John Wiley & Sons, Hoboken, USA.
Frank, M., Wolfe, P., et al., 1956. An algorithm for quadratic programming. Nav. Res. Logist. 3, 95–110.
Fukushima, M., 1984. A modified frank-Wolfe algorithm for solving the traffic assignment problem. Transp. Res. Part B Methodol. 18, 169–177.
Guo, Q., Ban, X.J., Aziz, H.A., 2021. Mixed traffic flow of human driven vehicles and automated vehicles on dynamic transportation networks. Transport. Res. C Emerg. Technol. 128, 103159.
Jayakrishnan, R., Tsai, W.T., Prashker, J.N., Rajadhyaksha, S., 1994. A faster path-based algorithm for traffic assignment. Transp. Res. Rec. 1443, 75–83.
Liu, J., Mirchandani, P., Zhou, X., 2020. Integrated vehicle assignment and routing for system-optimal shared mobility planning with endogenous road congestion. Transport. Res. C Emerg. Technol. 117, 102675.
Liu, T., Meidani, H., 2024. End-to-end heterogeneous graph neural networks for traffic assignment. Transport. Res. C Emerg. Technol. 165, 104695.
Liu, T., Meidani, H., 2025. Multi-class traffic assignment using multi-view heterogeneous graph attention networks. Expert Syst. Appl. 286, 128072.
Liu, Z., Dong, Y., Zhang, H., Zheng, N., Huang, K., 2024. A novel parallel computing framework for traffic assignment problem: integrating alternating direction method of multipliers with Jacobi over relaxation method. Transp. Res. Part E Logist Transp Rev 189, 103687.
Liu, Z., Yin, Y., Bai, F., Grimm, D.K., 2023. End-to-end learning of user equilibrium with implicit neural networks. Transport. Res. C Emerg. Technol. 150, 104085.
Lujak, M., Giordani, S., Ossowski, S., 2015. Route guidance: bridging system and user optimization in traffic assignment. Neurocomputing 151, 449–460.
Luo, G., Wang, Y., Zhang, H., Yuan, Q., Li, J., 2023. AlphaRoute: large-scale coordinated route planning via Monte Carlo tree search. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 12058–12067.
Monderer, D., Shapley, L.S., 1996. Potential games. Game. Econ. Behav. 14, 124–143.
Mounce, R., Carey, M., 2015. On the convergence of the method of successive averages for calculating equilibrium in traffic networks. Transp. Sci. 49, 535–542.
Nie, Y.M., 2010. A class of bush-based algorithms for the traffic assignment problem. Transp. Res. Part B Methodol. 44, 73–89.
Oliehoek, F.A., Amato, C., et al., 2016. A Concise Introduction to Decentralized Pomdps. Springer International Publishing, Cham, Switzerland.
Rahman, R., Hasan, S., 2023. Data-driven traffic assignment: a novel approach for learning traffic flow patterns using graph convolutional neural network. Data Sci. Transp. 5, 11.
Ramos, G.d.O., Bazzan, A.L., da Silva, B.C., 2018. Analysing the impact of travel information for minimising the regret of route choice. Transport. Res. C Emerg. Technol. 88, 257–271.
Schulman, J., Moritz, P., Levine, S., Jordan, M., Abbeel, P., 2015. High-Dimensional Continuous Control Using Generalized Advantage Estimation arXiv:1506.02438.
Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O., 2017. Proximal Policy Optimization Algorithms arXiv:1707.06347.
Shou, Z., Chen, X., Fu, Y., Di, X., 2022. Multi-agent reinforcement learning for Markov routing games: a new modeling paradigm for dynamic traffic assignment. Transport. Res. C Emerg. Technol. 137, 103560.
Siri, E., Siri, S., Sacone, S., 2022. A topology-based bounded rationality day-to-day traffic assignment model. Commun. Transp. Res. 2, 100076.
Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z., 2016. Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2818–2826.
Transportation Networks for Research Core Team, 2024. Transportation networks for research. https://github.com/bstabler/TransportationNetworks.
Wang, L., Duan, P., He, Z., Lyu, C., Chen, X., Zheng, N., et al., 2025a. Agentic large language models for day-to-day route choices. Transport. Res. C Emerg. Technol. 180, 105307.
Wang, L., Duan, P., Lyu, C., Ma, Z., 2025b. Reinforcement Learning-based Sequential Route Recommendation for system-optimal Traffic Assignment arXiv:2505.20889.
Williams, R.J., 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. Mach. Learn. 8, 229–256.
Yen, J.Y., 1971. Finding the k shortest loopless paths in a network. Manag. Sci. 17, 712–716.

Yu, C., Velu, A., Vinitsky, E., Gao, J., Wang, Y., Bayen, A., et al., 2022. The surprising effectiveness of ppo in cooperative multi-agent games. In: Advances in Neural Information Processing Systems, pp. 24611–24624.
Zhang, K., Zhang, H., Dong, Y., Wu, Y., Chen, X., 2023. An ADMM-based parallel algorithm for solving traffic assignment problem with elastic demand. Commun. Transp. Res. 3, 100108.
Zhou, B., Song, Q., Zhao, Z., Liu, T., 2020. A reinforcement learning scheme for the equilibrium of the in-vehicle route choice problem based on congestion game. Appl. Math. Comput. 371, 124895.
Zhou, Z., Roncoli, C., 2022. A scalable vehicle assignment and routing strategy for real-time on-demand ridesharing considering endogenous congestion. Transport. Res. C Emerg. Technol. 139, 103658.
Zhuang, D., Huang, Y., Jayawardana, V., Zhao, J., Suo, D., Wu, C., 2022. The Braess's Paradox in dynamic traffic. In: 2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC), pp. 1018–1023.

Leizhen Wang received the M.S. degree in transportation engineering from Southeast University, China, in 2021. He is currently working as a Ph.D. student with the Department of Data Science and Artificial Intelligence, Monash University, Australia. His research interests include large language models, reinforcement learning, and intelligent transport systems.

Peibo Duan is currently a Senior Lecturer in the Department of Data Science and Artificial Intelligence, Monash University, Australia. He received the Ph.D. degree from the University of Technology Sydney, Australia, in 2020. His current research interests include machine learning and intelligent transportation systems.



Cheng Lyu received the B.S. degree and the M.S. degree from the School of Transportation, Southeast University, China, in 2018 and 2021, respectively. He is currently pursuing the Ph.D. degree with Technical University of Munich (TUM), Germany. His research interests include transportation big data analysis and modeling, machine learning, data mining, and intelligent transportation systems.

Zewen Wang received the M.S. degree in transportation engineering from Southeast University, China, in 2021. He is currently working as an Algorithm Engineer with Meituan, China. His research interests include machine learning and intelligent transportation systems.

Zhiqiang He is currently pursuing the Ph.D. degree in the University of Electric Communications in Japan. He received the M.S. degree in control science and engineering from Northeastern University, China. His research interests focus on deep reinforcement learning and its applications.

Nan Zheng is currently an Associate Professor and Deputy Director (International) at the Department of Civil and Environmental Engineering, Monash University. He received the B. S., M.S., and Ph.D. degrees in transportation engineering, from the Southeast University, Delft University of Technology, École polytechnique fédérale de Lausanne (EPFL), in 2007, 2009, and 2014, respectively. He has developed research progressively on autonomous driving safety utilising human-centric approaches. His work has also focused on autonomous vehicle applications and commercialisation, for example, unmanned transportation solutions in controlled environments such as industrial parks and open-pit mining sites.

Zhenliang Ma is currently an Associate Professor of transportation systems with the KTH Royal Institute of Technology. His research interests include statistics, machine learning, computer science-based modeling, simulation, optimization, and control within the framework of selected mobility-related complex systems, which are intelligent transport systems and multimodal mobility systems.