

Network-Constrained Policy Optimization for Adaptive Multi-agent Vehicle Routing

FAZEL ARASTEH , York University, Canada

ARIAN HAGHPARAST , York University, Canada

MANOS PAPAGELIS , York University, Canada

Traffic congestion in urban road networks is marked by longer trip times and higher emissions, especially during peak periods. While the SHORTEST PATH FIRST (SPF) algorithm is optimal for a single vehicle in a static network, it performs poorly in dynamic, multi-vehicle settings, often worsening congestion by routing all vehicles along identical paths. We address dynamic vehicle routing through a **multi-agent reinforcement learning (MAREL)** framework for coordinated, network-aware fleet navigation. We first propose ADAPTIVE NAVIGATION (AN), a decentralized MAREL model where each intersection agent provides routing guidance based on (i) local traffic and (ii) neighborhood state modeled using **Graph Attention Networks (GAT)**. To improve scalability in large networks, we further propose HIERARCHICAL HUB-BASED ADAPTIVE NAVIGATION (HHAN), an extension of AN that assigns agents only to key intersections (*hubs*). Vehicles are routed hub-to-hub under agent control, while SPF handles micro-routing within each hub region. For hub coordination, HHAN adopts **centralized training with decentralized execution (CTDE)** under the **Attentive Q-Mixing (A-QMIX)** framework, which aggregates asynchronous vehicle decisions via attention. Hub agents use flow-aware state features that combine local congestion and predictive dynamics for proactive routing. Experiments on synthetic grids and real urban maps (Toronto, Manhattan) show that AN reduces average travel time versus SPF and learning baselines, maintaining 100% routing success. HHAN scales to networks with hundreds of intersections, achieving up to **15.9% improvement** under heavy traffic. These findings underscore the power of **network-constrained MAREL** for scalable, coordinated, congestion-aware routing in intelligent transportation systems.

CCS Concepts: • **Computing methodologies** → **Multi-agent reinforcement learning**; • **Applied computing** → *Transportation*.

Additional Key Words and Phrases: multi-agent reinforcement learning, adaptive vehicle navigation, intelligent transportation systems, Graph Attention Network (GAT)

ACM Reference Format:

Fazel Arasteh , Arian Haghighparast , and Manos Papagelis . 2025. Network-Constrained Policy Optimization for Adaptive Multi-agent Vehicle Routing. 1, 1 (October 2025), 29 pages. <https://doi.org/XXXXXXX.XXXXXXX>

*These authors contributed equally to this research.

The code for this research is publicly available at <https://github.com/Arianhgh/HHAN>

Authors' Contact Information: Fazel Arasteh , fazelara@eecs.yorku.ca, York University, Canada; Arian Haghighparast , arianhgh@my.yorku.ca, York University, Canada; Manos Papagelis , papaggel@eecs.yorku.ca, York University, Canada.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM.

Manuscript submitted to ACM

Manuscript submitted to ACM

1 Introduction

1.1 Motivation

Traffic congestion in urban road networks is a condition characterized by longer trip times, increased air pollution, and driver frustration. Different factors like rush hours, traffic incidents, road maintenance work and bad weather conditions can contribute to the traffic congestion. While construction of new road infrastructure is an expensive solution, the emergence of new technologies like widely available internet connection and GPS data can allow for more economical algorithmic traffic flow optimizations [30]. Currently, services like Google Maps¹ and Waze² help people with route planning mainly relying on a variant of the popular Shortest Path First (SPF) algorithm [10]. Mostly known as the Dijkstra algorithm, SPF is a routing algorithm in which a router computes the shortest path between a pair of nodes in a network.

1.2 Current approaches and limitations

In a static network and for a single vehicle, the SPF algorithm is optimal. However, road network conditions are not always static. In a dynamic road network, the SPF path between an origin and a destination is harder to compute due to variable traffic conditions. The main approach to address this issue is to recursively break down the problem and estimate the travel time for smaller road segments, where the traffic conditions remain unchanged. This is usually referred to as the *traffic prediction problem*. Several methods have been proposed to address the *traffic prediction problem* of a road segment, ranging from classical time series prediction methods, such as historical average and autoregressive integrated moving average (ARIMA) models, to machine learning methods, such as Support Vector Regression and Random Forest. More recently, deep learning methods have been proposed to address the traffic prediction problem [41, 50]. Still, the estimated travel times, specifically long-term predictions, may be inaccurate, rendering the global SPF algorithm to be sub-optimal at times. Another drawback of the SPF algorithm is that in multiple-vehicles scenarios, it will route every single vehicle through the currently available shortest path. As a result, due to the limited capacity of roads, the current shortest path gets quickly congested. In other words, SPF is short-sighted and causes congestion by naively sending every vehicle through the same shortest path. Other methods, such as probabilistic dynamic programming [46] and ant colony optimization [40] have been proposed to directly route the vehicles in the dynamic network. More recently, deep reinforcement learning has also been proposed for end-to-end routing without individual road segment travel time prediction [15, 22, 35]. Moreover, graph convolution networks have been proposed to embed the structure of the road network and exploit together with reinforcement learning for routing in large dynamic networks [49].

1.3 Problem formulation

We address the dynamic vehicle routing problem in urban networks, which seeks to minimize the overall travel time of a vehicle fleet while adapting to real-time traffic conditions. This problem, while extensively studied in transportation research, presents unique challenges when approached through multi-agent reinforcement learning due to the need for (i) real-time adaptation to dynamic traffic conditions and (ii) coordinated decision-making to avoid system-wide congestion cascades. Our formulation differs from classical shortest path routing in its emphasis on multi-agent coordination, and from fleet management problems [18, 27, 55] in its focus on individual vehicle routing rather than supply-demand balancing. The problem shares conceptual similarities with packet routing in IP networks, where discrete

¹<https://maps.google.com/>

²<https://www.waze.com/>

entities (vehicles/packets) are routed through intermediate nodes (intersections/routers) toward destinations. However, vehicular networks present distinct challenges including routing restrictions (one-way streets, turn prohibitions), physical capacity constraints, and the absence of hierarchical addressing schemes that characterize IP networks. These differences necessitate specialized modeling approaches that account for the unique spatiotemporal dynamics of urban traffic systems.

1.4 Our approach

To address the *vehicle navigation problem*, we propose a *network-aware multi-agent reinforcement learning (MARL)* approach with two distinct paradigms. Unlike existing MARL approaches in traffic management that focus primarily on signal control or fleet assignment, our method directly addresses individual vehicle routing through explicit coordination mechanisms. Our first contribution is the *ADAPTIVE NAVIGATION (AN)*, a fully decentralized system that assigns a reinforcement learning agent to every intersection. When a vehicle approaches an intersection, it submits a routing query to the agent, including its final destination. The agent then generates a routing response based on the vehicle’s destination and the current state of local traffic, leveraging Graph Attention Networks (GAT) for neighbor communication and emergent coordination.

However, assigning an agent to every intersection is not feasible for large, real-world city networks due to scalability constraints. To overcome this challenge, we introduce *HIERARCHICAL HUB-BASED ADAPTIVE NAVIGATION (HHAN)*, a hierarchical and scalable extension of our model. In this framework, agents are placed only at a strategically selected subset of critical intersections, referred to as *hubs*. A vehicle’s journey is thus decomposed into a sequence of long-range, hub-to-hub navigations. To enhance coordination among hub agents in HHAN, we employ the *Attentive Q-Mixing (A-QMIX)* framework following a centralized training with decentralized execution paradigm. Building upon the QMIX value decomposition approach [37], our method extends it to handle asynchronous agent decisions through attention-based aggregation. This approach allows agents to learn a shared, global value function during training, enabling them to discover collaborative routing policies that minimize system-wide congestion while maintaining decentralized execution. The feasibility of our approach relies on *vehicle-to-infrastructure (V2I)* communication, where vehicles query hub agents for their next routing directive. This is enabled by technologies such as DSRC [20], and can be applied to both conventional and autonomous vehicles [6, 31].

1.5 Contributions

Our work makes the following key technical contributions to multi-agent reinforcement learning for dynamic vehicle routing:

- **Adaptive Navigation (AN).** We develop *ADAPTIVE NAVIGATION (AN)*, a fully decentralized MARL approach for coordinated vehicle routing, incorporating Graph Attention Networks for intersection-level coordination and emergent multi-agent behavior.
- **Hierarchical Hub-based Adaptive Navigation (HHAN).** We introduce *HIERARCHICAL HUB-BASED ADAPTIVE NAVIGATION (HHAN)*, a scalable hierarchical extension of the *ADAPTIVE NAVIGATION* model using strategically-selected *hub agents*. HHAN employs the *Attentive Q-Mixing (A-QMIX)* framework for centralized training with decentralized execution to address large-scale networks. A-QMIX is a coordination mechanism that extends traditional QMIX to handle asynchronous multi-agent decisions through a novel *attention-based aggregation* method operating over *Global Collection Epochs (GCE)*.

- **Spatial locality preservation through Z-order curve encoding.** We adapt the *Z-order space-filling curve* for destination representation in traffic networks, providing a scalable method for preserving spatial locality while maintaining neural network separability.
- **Comprehensive Evaluation.** We conduct empirical evaluation on synthetic and realistic road networks, demonstrating performance improvements over established routing baselines and analyzing the learned coordination behaviors.
- **Open-source code.** We ensure reproducibility by making source code publicly available.

GitHub repository:

<https://github.com/Arianhgh/HHAN>

1.6 Organization

Section 2 provides preliminaries and formally defines the adaptive navigation problem. Section 3 presents the methodology for both the AN model and its hierarchical extension HHAN. Section 4 discusses our experimental evaluation. The related work is discussed in Section 5. We conclude in Section 6.

2 Problem Definition

We define the adaptive navigation problem based on an intersection-level formulation of the road network.

2.1 Network Model

We model the road network as a directed graph $W = (I, R)$, where $I = \{i_1, \dots, i_N\}$ is the set of vertices representing intersections, and $R = \{r_1, \dots, r_M\}$ is the set of directed edges representing roads. Each road $r \in R$ is a directed edge from $r.head \in I$ to $r.tail \in I$, meaning that every road connects exactly two intersections. Let $VCS = \{vc_1, \dots, vc_L\}$ denote the set of L vehicles, and $U = \{u_1, \dots, u_N\}$ the set of N router agents, each assigned to one intersection. When a vehicle approaches an intersection, it issues a routing query to the corresponding router agent.

DEFINITION 1 (q : ROUTING QUERY).

$$q = (t, vc, u, r_c, i_d, t_{\max})$$

A query generated at time t by vehicle vc (where vc denotes the vehicle identifier) currently traveling on road r_c , addressed to router agent u assigned to intersection $r_c.tail$.³ The query specifies the destination intersection i_d and the arrival deadline $t_{\max}$.

The router $q.u$, located at the tail of the vehicle's current road, responds to q with a routing decision. To define this, we first introduce the next-hop road set.

DEFINITION 2 ($NH(r)$: NEXT-HOP ROAD SET).

$$NH(r) = \{r_k \in R \mid r_k.head = r.tail, r_k \text{ is connected to } r\}$$

The set of outgoing roads from intersection $r.tail$ that can be reached directly from r . A road r_k is considered connected to r if the intersection's traffic rules allow travel from r to r_k (e.g., no U-turns or prohibited turns).

³We use dot notation to denote object attributes.

DEFINITION 3 ($resp(q)$: ROUTING RESPONSE).

$$resp(q) = \begin{cases} \langle success \rangle, & \text{if } q.r_c.tail = q.i_d, \\ \langle fail \rangle, & \text{if } q.t_{max} < q.t, \\ \langle r \in NH(q.r_c) \rangle, & \text{otherwise.} \end{cases}$$

A routing response may indicate success, failure, or specify the next road to take.

DEFINITION 4 ($trip$: TRIP OF VEHICLE vc).

$$trip = (vc, t, r, i, t_{max})$$

The trip of vehicle vc starting at time t from road r with destination intersection i and arrival deadline t_{max} .

We denote the set of all trips as $Trips = \{trip \mid trip.vc \in VCs\}$.

DEFINITION 5 ($path(trip)$: PATH OF A TRIP).

$$path(trip) = (resp(q_1), \dots, resp(q_z) \in \{\langle success \rangle, \langle fail \rangle\})$$

The sequence of routing responses for the queries generated by $trip.vc$. The length of a path is $|path| = z$ and the last element is $path_{|path|} = resp(q_z)$. The set of all paths is $Paths = \{path(trip) \mid trip \in Trips\}$.

DEFINITION 6 ($tt(p)$: TRAVEL TIME OF PATH p).

$$tt(p) = (q_{|p|}.t) - (q_1.t)$$

The difference between the timestamps of the last and first queries in path p .

DEFINITION 7 (RS : ROUTING SUCCESS).

$$RS = \{p \in Paths \mid p_{|p|} = \langle success \rangle\}$$

The set of paths that end with a $\langle success \rangle$ response.

DEFINITION 8 ($AVTT$: AVERAGE TRAVEL TIME).

$$AVTT = \frac{\sum_{p \in RS} tt(p)}{|RS|}$$

The average travel time of all successful paths.

DEFINITION 9 (LOCALITY OF ACCESS). Let $D(i, j)$ denote the Euclidean distance between intersections i and j , and $E(T(i, j))$ the expected travel time between them. A network satisfies locality of access if, for intersections $i_1, i_2, i_3 \in I$ with $D(i_1, i_2) > D(i_1, i_3)$, it holds that $E(T(i_1, i_2)) > E(T(i_1, i_3))$. This property is crucial for efficient destination representations, as preserved in our Z-order encoding.

2.2 Adaptive Navigation Problem

We now formally define the *adaptive navigation problem*.

DEFINITION 10 (ADAPTIVE NAVIGATION PROBLEM). Given a road network W and a set of routing queries Q , the objective is to generate a routing response $resp(q)$ for each $q \in Q$ so as to:

- (1) *maximize $|RS|$, the number of successful routes, and*
- (2) *minimize AVTT, the average travel time over all successful routes.*

3 Methodology

We present our methodology in two parts. The first part describes the ADAPTIVE NAVIGATION (AN), where an agent is assigned to each intersection in a fully decentralized multi-agent reinforcement learning (MARL) approach. The second part introduces a scalable extension that employs a hierarchical hub-based structure with centralized training and decentralized execution to address large-scale networks effectively.

Multi-Agent Paradigm Overview. Before detailing our methodology, we clarify the multi-agent paradigms employed. In **decentralized systems**, agents operate independently with local observations and decision-making, relying on emergent coordination through shared environment dynamics. Our AN model exemplifies this approach, with intersection agents making autonomous routing decisions based on local traffic states and limited neighborhood information via Graph Attention Networks. In contrast, **centralized training with decentralized execution (CTDE)** systems train agents using global information but deploy them with only local observations for scalability. Our HHAN model follows this paradigm, using centralized coordination during training through the A-QMIX framework while maintaining decentralized execution capabilities. The choice of paradigm reflects the inherent trade-offs between coordination effectiveness and computational scalability in multi-agent traffic systems.

3.1 Adaptive Navigation (AN)

In our ADAPTIVE NAVIGATION (AN) model, we formulate the traffic routing problem as a decentralized MARL task, assigning a unique agent to each intersection to handle routing decisions. This fully distributed approach leverages local information while achieving implicit coordination through shared network states and Q-learning updates. Below, we detail the formulation, state representations, actions, rewards, and learning process.

3.1.1 MARL Formulation. Router agent at intersection i , u_i . We assign a unique agent u_i to each intersection $i \in I$. The agent u_i responds only to queries $q \in Q$ where the tail of the current road segment $q.r_c.tail$ equals i . This ensures that each agent focuses on decisions relevant to its specific intersection.

State of query q , s_q . The state of a query q is defined as the unique representation of its destination intersection, denoted as $[q-i_d]$. We discuss efficient representations for destination IDs in Section 3.1.2:

$$s_q = [q-i_d]$$

State of intersection i at time t , s_i^t . The state of intersection i at time t captures the congestion status of its outgoing roads. A road $r \in R$ with $r.head == i$ is considered congested ($C(r) = True$) if its current speed is below a fixed proportion (defined by the hyperparameter `congestion-speed-factor`) of its free-flow speed. The state s_i^t is zero-extended to a fixed dimension $\mathbb{R}^F$, where F is the maximum number of outgoing roads among all intersections:

$$s_i^t = [[1 \text{ if } C(r) == True \text{ else } 0]] \quad \forall r \in R, r.head == i$$

State of road network W at time t , s_W^t . The network state at time t is the concatenation of all intersection states:

$$s_W^t = [s_1^t | \dots | s_N^t]$$

State of query q at step τ , s_q^τ . For a query q associated with vehicle $q\text{-vc}$ at step τ in its path, the state is a tuple combining the query's destination representation and the network state at the query's time:

$$s_q^\tau = (s_q, s_W^{q-t})$$

Action of agent u_i for s_q^τ , $a(s_q^\tau)$. The action is the selection of an outgoing road segment from the intersection $i = q\text{-}r_c\text{-tail}$:

$$a(s_q^\tau) = \text{resp}(q)$$

Next state of s_q^τ , $s_q^{\tau+1}$. For the next query q' in the vehicle's path (i.e., at step $\tau + 1$):

$$s_q^{\tau+1} = (s_{q'}, s_W^{q'-t'})$$

Reward, $r(a(s_q^\tau))$. The reward is defined as the negative time difference between consecutive queries in the vehicle's path:

$$\Delta T = (q'-t) - (q-t)$$

$$r(a(s_q^\tau)) = -\Delta T$$

The justification for this reward function, which incentivizes minimizing travel time, is discussed in Section 3.3.

Network State Aggregation with GAT. To provide agents with relevant traffic context, we aggregate the network state using a Graph Attention Network (GAT). The GAT takes the network state $s_W^t = \{s_1^t, \dots, s_N^t\}$, where $s_i^t \in \mathbb{R}^F$, as input and produces a local embedding $s'_i \in \mathbb{R}^{F'}$ for each agent u_i . A shared linear transformation, parameterized by a weight matrix $\omega \in \mathbb{R}^{F' \times F}$, transforms the input features into higher-level features:

$$H_i = \omega \cdot s_i^t$$

A self-attention mechanism computes attention coefficients $e_{ij} = a(H_i, H_j)$ for nodes $j \in N_i$ (the one-hop neighborhood of i , including i itself). These coefficients are normalized using a softmax function:

$$\alpha_{ij} = \text{softmax}(e_{ij}) = \frac{\exp(e_{ij})}{\sum_{k \in N_i} \exp(e_{ik})}$$

The GAT output for intersection i is:

$$(GAT(s_W^t))_i = s'_i = \sigma \left(\sum_{j \in N_i} \alpha_{ij} H_j \right)$$

Multiple attention heads stabilize the learning process, with concatenation for intermediate layers and averaging for the final layer. The number of GAT layers is a tunable hyperparameter, controlling the extent of neighborhood information aggregation.

State Representation for Learning. The destination s_q is processed through a linear layer with ReLU activation to produce embeddings:

$$[s_q] = \text{ReLU}(\text{Linear}_i(s_q))$$

The routing response is the action with the highest Q-value:

$$\text{resp}(q) = \underset{a}{\operatorname{argmax}} Q_i(s_q^\tau, a)$$

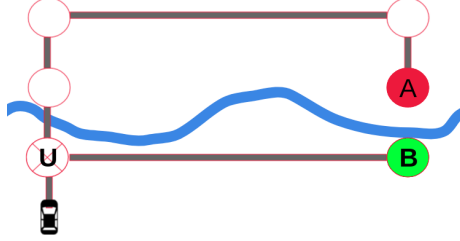


Fig. 1. Hard to separate destinations.

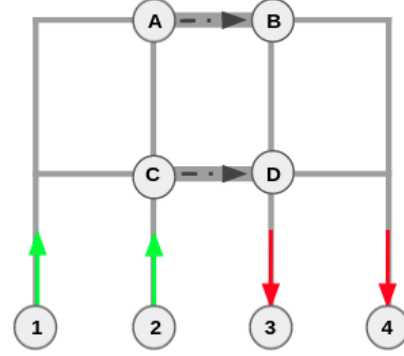


Fig. 2. Collaborative policies.

Each agent uses a Q-learning algorithm with the MSE loss:

$$L(s_q^t, a, r : \theta) = \mathbb{E} \left[\left(r + \gamma \max_{a'} Q_{i+1}(s_q^{t+1}, a') - Q_i(s_q^t, a) \right)^2 \right]$$

The intertwined Q-learning updates enable implicit coordination, as the value of a state at agent u_i depends on the next agent's Q-values.

3.1.2 Destination Representation. The destination representation must be unique, separable by the neural network, low-dimensional, and preserve locality of access. We compare two approaches and propose the Z-order curve for optimal performance.

Coordinates and One-Hot IDs. Normalized intersection coordinates are unique, low-dimensional (dimension = 2), and preserve locality but are hard for neural networks to separate for nearby intersections (see Figure 1). One-hot encodings are unique and separable but high-dimensional (dimension = N) and discard locality information.

Z-order ID. We propose using the Z-order curve [32] to create unique, linearly separable intersection IDs that partially preserve locality while maintaining a low dimension ($\log_2(N)$). The Z-order curve interleaves the binary representations of a point's coordinates to compute a Z-value, sorting points accordingly. This is equivalent to a depth-first traversal of a quad-tree, forming Z-shapes (see Figures 3a and 3b). For an intersection i_1 with Z-order index 2, its ID is $\text{binary}(2) = [0, 1, 0]$.

3.1.3 Algorithm Sketch. Algorithm 1 outlines the inference process at time step t , taking the network state s_W^t and queries Q as inputs to generate routing responses. Algorithm 2 describes the training process for the agents.

3.2 Scalable Extension: Hierarchical Hub-based Adaptive Navigation (HHAN)

To enhance scalability for large-scale road networks, we propose the HIERARCHICAL HUB-BASED ADAPTIVE NAVIGATION (HHAN) model. Instead of placing agents at every intersection, we strategically select a subset of key intersections (hubs) and assign agents to them. These agents coordinate through a centralized training scheme with the Attentive Q-Mixing (A-QMIX) framework, enabling efficient routing decisions across expansive networks.

3.2.1 Hierarchical Hub Abstraction. In large-scale, real-world road networks, assigning and coordinating agents at every intersection can be computationally expensive and operationally impractical. To address this, HHAN introduces a

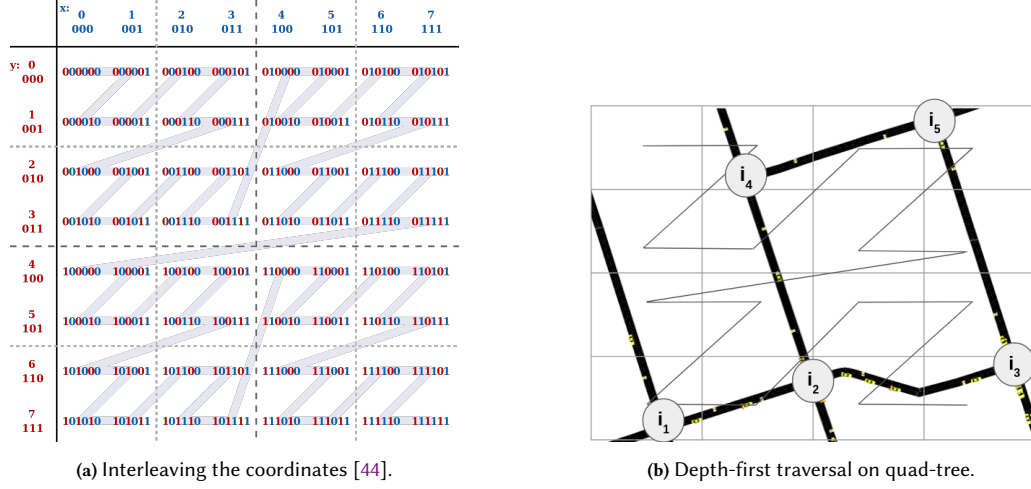


Fig. 3. Z-order curve (Morton Space Filtering).

Algorithm 1 Inference at time t

Require: state of the road network s_W^t , set of all the routing queries Q at time t
Ensure: optimal $resp(q)$ for $q \in Q$

```

1: for all  $q \in Q$  do
2:    $s_q \leftarrow$  state of query  $q$ 
3:    $u \leftarrow q-u$ , the router agent
4:    $u.memory.push(\text{previous experience tuple of agent } u)$ 
5:    $[s_q] \leftarrow \text{ReLU}(\text{Linear}_u(s_q))$ 
6:    $s'_i \leftarrow \text{GAT}((s_W^t))[i]$ 
7:    $s_{agg} \leftarrow s'_i$ 
8:    $r_q \leftarrow \text{argmax } Q-net_u([s_q] | s_{agg})$ 
9: end for
10: if in training mode then
11:   Train( $\{q-u | q \in Q\}$ )
12: end if

```

Algorithm 2 Train

Require: set of router agents U that need training
Ensure: training of RL agents

```

1: for all  $u \in U$  do
2:   if time-to-learn( $u$ ) then
3:     training-batch =  $u.memory.sample()$ 
4:     loss = MSE-loss(training-batch)
5:     loss.backward()
6:     UpdateGATParameters()
7:   end if
8: end for

```

hierarchical hub abstraction that reduces the complexity of decision-making by focusing only on a strategically selected subset of intersections. This abstraction is built directly on top of the underlying road network graph W .

DEFINITION 11 (HUB NETWORK). *The road network W is abstracted into a directed hub graph $W_H = (H, E_H)$, where $H = \{h_1, \dots, h_K\}$ is a set of K hubs, with $K \ll N$. Each hub $h_k \in H$ corresponds to a strategically chosen intersection from I based on criteria such as high traffic centrality, network connectivity, or bottleneck potential. An edge $(h_a, h_b) \in E_H$ exists if there is at least one viable route from hub h_a to hub h_b in the original road network W .*

In this abstraction, the vehicle navigation problem shifts from making local routing decisions at every intersection to making higher-level strategic decisions only at hubs. Each vehicle's journey is decomposed into a sequence of *hub-to-hub* segments. The micro-level routing between two hubs is delegated to a conventional Shortest Path First (SPF) algorithm, which ensures efficiency in low-level navigation while the hub-level agents focus on global coordination and congestion management.

DEFINITION 12 (HUB-LEVEL ROUTING QUERY). *When a vehicle arrives at a hub h_k , it submits a routing query to the corresponding hub agent u_k . The query contains the vehicle's current hub and its final destination. Instead of returning a single next road segment, the hub agent selects the next hub $h_{next} \in H$ to navigate towards, based on its learned policy and current traffic conditions.*

This hierarchical approach provides two major advantages. First, it significantly reduces the number of agents, making the problem tractable for metropolitan-scale networks. Second, by operating at a higher level of abstraction, hub agents can coordinate more effectively to prevent downstream congestion, rather than reacting only to immediate local traffic conditions. In HHAN, the hub network serves as a strategic decision layer, while standard SPF routing ensures fine-grained vehicle movement between hubs. This design balances scalability, adaptability, and coordination in a unified framework.

3.2.2 Hub Selection and Connectivity. The effectiveness of HHAN relies on the careful selection and connectivity of hubs, which serve as critical decision points in the network.

- (1) **Candidate Filtering:** We identify significant intersections by selecting nodes with an in-degree of at least three and an out-degree of at least three, ensuring that these nodes correspond to major junctions with sufficient directional connectivity to meaningfully affect routing decisions.
- (2) **Hub Selection:** From these candidates, we use the K-Medoids clustering algorithm with shortest-path distance as the metric to select hubs. This approach identifies the most central intersection (medoid) within each cluster, ensuring both centrality and spatial distribution across the network.
- (3) **Hub Connectivity:** To form the hub graph W_H , each hub is connected to at most $k = 3$ nearest neighboring hubs based on shortest-path travel time. A connection is established only if the Euclidean distance between hubs is below a threshold d_{max} , where d_{max} is chosen according to the scale of the map to prevent impractical long-distance routing.

This structured approach ensures that hubs are strategically placed to cover the network efficiently while maintaining feasible routing paths.

3.2.3 Hierarchical Agent Formulation. In this model, an agent u_k is assigned to each hub $h_k \in H$. Unlike the foundational model, where agents act at specific intersections, hub agents are triggered when a vehicle enters their operational vicinity, defined as a radius r_{vic} around the hub. The value of r_{vic} is chosen according to the scale of the map and is set as half of the minimum distance between two neighboring hubs. This vicinity-based approach provides navigational flexibility, allowing agents to make decisions based on broader traffic patterns. The agent's action is to select the next hub h_{next} for the vehicle to travel toward. If h_{next} is the vehicle's final destination hub, the vehicle is routed directly to its final road edge using a standard Shortest Path First (SPF) algorithm.

3.2.4 Flow-Aware State Representation. To enable agents to make informed decisions, we design a state representation that captures both local and predictive traffic flow dynamics, replacing the GAT with a fixed-size representation.

DEFINITION 13. *Local Observation τ_k*

The local observation for agent u_k is a concatenated vector:

$$\tau_k = \text{concat}(\text{Emb}(h_d), F_k, F_{N(k)})$$

- (1) *Destination Hub Embedding $\text{Emb}(h_d)$:* The Z-order embedding of the vehicle's destination hub h_d , as described in Section 3.1.2, ensuring a unique and locality-preserving representation.
- (2) *Current Hub Features F_k :* A feature vector for hub h_k , including:
 - **Vicinity Speed:** The average normalized speed of vehicles within a radius r_{vic} of the hub. This radius captures approaching and departing traffic, providing predictive insights into potential congestion.
 - **Outgoing Congestion Ratio:** The average ratio of current travel time to free-flow travel time on edges within a radius r_{vic} of the hub, indicating the ease of traffic dispersal from the hub.
- (3) *Padded Neighbor Features $F_{N(k)}$:* A fixed-size feature vector for up to $M_{neighbors}$ neighboring hubs, each containing:
 - **Estimated Travel Time:** A normalized estimate of travel time from h_k to neighbor h_j , reflecting real-time conditions.
 - **Neighbor Congestion Ratio:** The average congestion ratio on edges within a radius r_{vic} of the hub h_j , providing information about downstream conditions.
 - **Distance to Destination:** The normalized static network distance from h_j to the destination hub h_d , aiding in long-term routing decisions.

This flow-aware representation equips agents with an understanding of traffic dynamics, enabling proactive routing decisions.

3.2.5 Coordinated Training with Attentive Q-Mixing (A-QMIX). To achieve robust coordination in an asynchronous system, we adopt a Centralized Training with Decentralized Execution (CTDE) paradigm based on the QMIX framework [37], introducing the Global Collection Epoch (GCE) to bundle decisions over time.

Global Collection Epoch (GCE) A GCE aggregates all decisions made across the system within a time period or a fixed number of decisions into a transition tuple

$$(\mathbf{s}, \{\mathcal{D}_k\}_{k=1}^K, r, \mathbf{s}'),$$

where $\mathbf{s}$ is the global state, $\{\mathcal{D}_k\}_{k=1}^K$ is the set of decisions made by all agents $u_1, \dots, u_K$, r is the aggregated reward, and $\mathbf{s}'$ is the resulting global state after executing all the decisions in the GCE.

DEFINITION 14. *Global State $\mathbf{s}$*

The global state captures system-wide flow properties:

- (1) *All-Hub Flow Snapshot*: For each hub, the vicinity speed and edge congestion ratio within operational vicinity, providing a network-wide view of traffic bottlenecks.
- (2) *System-Wide Efficiency*: Metrics such as the total number of active vehicles, completion throughput ratio (completed vs. started trips), and average trip inefficiency (actual vs. shortest path travel time).
- (3) *System Imbalance*: The standard deviation of vicinity speeds across hubs, quantifying traffic flow imbalance.

Attentive Q-Mixing (A-QMIX). In this framework, each agent u_k maintains a local Q-network

$$Q_k(\tau_k, a_k),$$

which estimates the expected cumulative reward for taking action a_k given the local observation τ_k . Unlike standard MARL settings where agents act synchronously, in our traffic control environment, agents make multiple decisions asynchronously within a Global Collection Epoch (GCE). To handle this asynchrony, we introduce an *attention mechanism* that aggregates an agent's multiple local decisions into a single utility score.

For agent u_k with decision set $\mathcal{D}_k = \{d_1, d_2, \dots, d_{|\mathcal{D}_k|}\}$ in the current GCE, we compute attention weights $\alpha_{k,i}$ for each decision $d_i \in \mathcal{D}_k$ as:

$$\alpha_{k,i} = \frac{\exp(\mathbf{w}^\top \tanh(\mathbf{W}_1[\mathbf{s}; \tau_{k,d_i}; Q_k(\tau_{k,d_i}, a_{k,d_i})]))}{\sum_{j=1}^{|\mathcal{D}_k|} \exp(\mathbf{w}^\top \tanh(\mathbf{W}_1[\mathbf{s}; \tau_{k,d_j}; Q_k(\tau_{k,d_j}, a_{k,d_j})]))},$$

where $\mathbf{W}_1 \in \mathbb{R}^{h \times (|\mathbf{s}| + |\tau_k| + 1)}$ and $\mathbf{w} \in \mathbb{R}^h$ are learnable parameters, $[\cdot; \cdot]$ denotes concatenation, and h is the hidden dimension. The aggregated Q-value is then computed as:

$$Q_k^* = \sum_{i=1}^{|\mathcal{D}_k|} \alpha_{k,i} \cdot Q_k(\tau_{k,d_i}, a_{k,d_i}).$$

This attention mechanism dynamically weighs each decision according to its relevance to the global state $\mathbf{s}$ and local context τ_{k,d_i} . Critical decisions at congested hubs or along high-priority routes receive higher attention weights $\alpha_{k,i}$, amplifying their influence on the aggregated utility Q_k^* , while decisions in low-impact scenarios are down-weighted, reducing noise in the learning signal.

The aggregated utilities Q_k^* from all agents are then passed to a *mixing network* that produces the global value function:

$$Q_{tot}(\mathbf{s}, \mathbf{a}; \theta),$$

where $\mathbf{a} = \{a_1, a_2, \dots, a_K\}$ represents the joint actions of all agents. A key property of the mixing network is the *monotonicity constraint*:

$$\frac{\partial Q_{tot}}{\partial Q_k} \geq 0, \quad \forall k,$$

which guarantees that improving an individual agent's Q-value cannot decrease the global Q-value. This monotonicity enables decentralized execution: agents can greedily select actions based on their local Q-values while still optimizing the system-wide objective.

Training of A-QMIX is performed end-to-end using temporal-difference (TD) learning. The loss function is defined as:

$$L(\theta) = \mathbb{E}[(y^{tot} - Q_{tot}(\mathbf{s}, \mathbf{a}; \theta))^2],$$

where the TD target is

$$y^{tot} = r + \gamma \max_{\mathbf{a}'} Q_{tot}(\mathbf{s}', \mathbf{a}'; \theta^-).$$

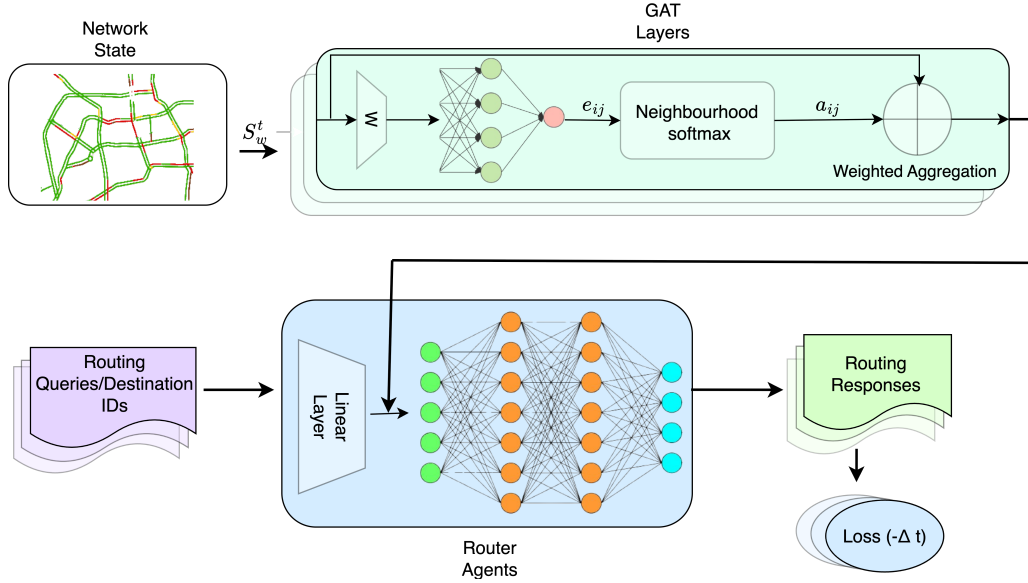


Fig. 4. ADAPTIVE NAVIGATION (AN) model architecture showing the decentralized MARL approach where each intersection has an agent that processes routing queries using destination embeddings and GAT-aggregated network states. The GAT layers enable neighborhood information sharing for implicit coordination between agents, while each agent makes independent routing decisions based on local Q-networks trained with intertwined Q-learning updates.

Here, r is the aggregated reward for the GCE, s' is the next global state, γ is the discount factor, and θ^- represents the parameters of a target network, which is periodically updated to stabilize training. The use of a global reward and state ensures that agents are incentivized to learn collaborative policies that improve overall traffic flow rather than only optimizing local metrics.

The attention-based aggregation in A-QMIX provides several benefits:

- **Handling Asynchrony:** Agents can make multiple decisions at different times, yet their contributions are combined meaningfully.
- **Prioritization of High-Impact Decisions:** Critical decisions affecting congestion or bottlenecks are weighted more heavily, improving learning efficiency.
- **Decentralized Execution with Global Coordination:** Monotonic mixing allows agents to act independently while still aligning with global objectives, which is crucial in real-time traffic systems.

3.2.6 Model Architecture. The architecture for both approaches is depicted in Figures 4 and 5. For the AN model, the destination s_q is processed through a linear layer with ReLU activation, while the network state s_w^t is aggregated via GAT or mean congestion methods. In HHAN, the local observation τ_k is used directly, with the centralized training framework handling coordination.

3.3 Reward Function Justification

The reward function $r(a(s_q^r)) = -\Delta T$ incentivizes minimizing travel time. Consider the Q-learning update rule:

$$Q_i(s_q^r, a) \leftarrow Q_i(s_q^r, a) + \alpha \left(r_\tau + \gamma \max_{a'} Q_{i+1}(s_q^{r+1}, a') - Q_i(s_q^r, a) \right)$$

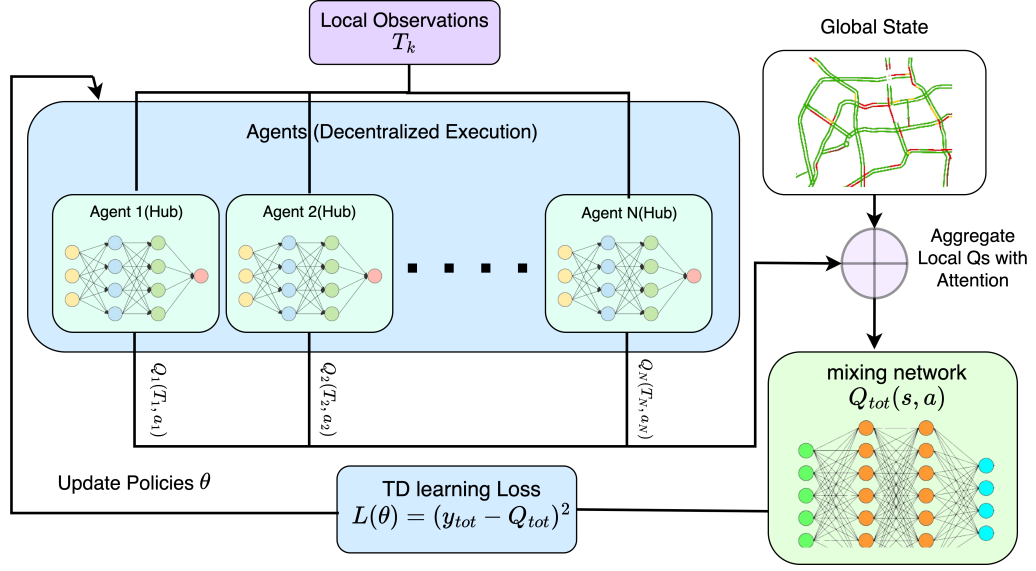


Fig. 5. HIERARCHICAL HUB-BASED ADAPTIVE NAVIGATION (HHAN) model architecture implementing centralized training with decentralized execution (CTDE) using the A-QMIX framework. Hub agents process local observations including destination embeddings and flow-aware features, making asynchronous routing decisions that are aggregated through an attention mechanism. The mixing network ensures monotonic value function combination while enabling coordinated learning across the hub-based network structure.

For an infinite horizon ($\gamma = 1$) and learning rate $\alpha = 1$, this simplifies to:

$$Q_i(s_q^\tau, a) = r_\tau + \max_{a'} Q_{i+1}(s_q^{\tau+1}, a')$$

Expanding for a terminal state $s_q^{\tau+Z}$:

$$Q_i(s_q^\tau, a) = r_\tau + r_{\tau+1} + \dots + r_{\tau+Z}$$

Substituting the reward function:

$$Q_i(s_q^\tau, a) = -\Delta T_1 - \Delta T_2 - \dots - \Delta T_Z$$

This shows that Q-values estimate the total travel time to the destination, prioritizing states closer to the destination over faster but less direct routes.

3.4 Justification of the MARL Architecture

An alternative single-agent RL approach introduces high variance, as identical actions at different intersections can lead to divergent outcomes (e.g., action 0 at i_1 goes north, but at i_2 goes south). This variance hinders learning the underlying routing logic. Following [2, 4, 52, 54], our MARL formulation reduces variance by assigning agents to specific intersections, ensuring consistent action-state mappings. The intertwined Q-learning updates (Equation 3.3) enable collaborative policies, as agents consider the Q-values of neighboring agents, fostering system-wide optimization.

For example, in Figure 2, the SPF algorithm may oscillate between bridges AB and CD under high load, causing congestion. The MARL approach explores collaborative policies, such as splitting traffic between bridges based on destinations, improving efficiency [4].

Table 1. AN Model Hyper-parameters.

Parameter	Value	Parameter	Value
Optimizer	Adam	Optimizer eps	1e-4
learning rate	0.01	batch-size	64
batch-norm	False	gradient-clipping-norm	5
buffer-size	10000	num-new-exp-to-learn	1
tau	0.01	discount rate	0.99
epsilon-decay-rate-denom	num episodes/100	stop-exploration-episode	num-eps-10
linear-hidden-units-size AN(0hop)	[8,6]	linear-hidden-units-size AN(1hop)	[10,6]
linear-hidden-units-size AN(2hop)	[12,9,6]		

Table 2. Graph Attention Network Hyper-parameters.

Parameter	Value	Parameter	Value
Optimizer	Adam	num-heads-per-layer	3
Optimizer eps	1e-4	learning rate	0.01
add-skip-connection	False	bias	True
dropout	0.6	layer-0 output dimension	7
intersection state dimension	4	layer-1 output dimension	10

Table 3. HHAN Model Hyper-parameters.

Parameter	Value	Parameter	Value
num_hubs	4	hub_agent_dim	64
max_waiting_vehicles	40	z_order_embedding_dim	8
num_episodes	500	max_steps_per_episode	3000
lr	0.0005	gamma	0.99
epsilon_start	1.0	epsilon_end	0.05
epsilon_decay	0.99	polyak	0.995
min_gce_buffer_size	200	gce_buffer_capacity	10000
qmix_batch_size	64	qmix_update_frequency_steps	32
mixing_hidden_dim	128	mixing_lr	0.0005
gce_size	10	gce_max_sim_time	100
clip_grad_norm	10.0		

3.5 Hyper-parameter Settings

Tables 1, 3 and 2 summarize the Hyper-parameter Settings.

4 Experimental Evaluation

This section provides an empirical evaluation of our proposed traffic routing models. We utilize the open-source microscopic traffic simulator, Simulation of Urban Mobility (SUMO) [28], to create reproducible testing environments.

Our evaluation examines two approaches: first, we assess the performance of the ADAPTIVE NAVIGATION (AN) model. Second, we evaluate the scalability of the HIERARCHICAL HUB-BASED ADAPTIVE NAVIGATION (HHAN) model on large-scale networks. Performance is benchmarked against established routing algorithms across synthetic and realistic road networks. Our primary evaluation metrics are **Average Vehicle Travel Time (AVTT)**, which quantifies system efficiency, and **Routing Success Rate (RSR)**, defined as the percentage of vehicles that successfully reach their destination within the simulated period, measuring the system’s reliability and ability to prevent gridlock [43].

4.1 Experimental Setup

To ensure reproducibility, we define a experimental protocol covering the simulation environment, network topologies, traffic demand profiles, baseline algorithms, and model configurations.

4.1.1 Simulation Environment and Metrics. All experiments were executed using SUMO, controlled via its Python API, TraCI [28]. The simulations were run on a server equipped with $2 \times$ Intel Xeon E5-2687W v4 3.0 GHz 12-Core Processors (30 MB L3 Cache), 512 GB of RAM ($8 \times$ 64 GB), and $8 \times$ NVIDIA MSI GeForce GTX 1080Ti 11 GB Aero OC GPUs for accelerating neural network training.

The core performance metrics are formally defined as:

- **AVTT:** $AVTT = \frac{1}{|V_{completed}|} \sum_{i \in V_{completed}} (t_{arrival,i} - t_{depart,i})$, where $V_{completed}$ is the set of vehicles that finished their trips, $t_{arrival,i}$ is the arrival time of vehicle i , and $t_{depart,i}$ is its departure time. Lower values are better.
- **RSR:** $RSR = \frac{|V_{completed}|}{|V_{total}|} \times 100\%$, where $|V_{total}|$ is the total number of vehicles introduced into the simulation. Higher values are better.

4.1.2 Road Networks. We employ three distinct road networks to evaluate our models under varying conditions of complexity and scale (Figures 6, 7, and 8):

- (1) **Synthetic 5x6 Grid:** A canonical Manhattan-style grid network consisting of 30 intersections and 98 edges (Figure 6). The 26 non-perimeter intersections are controlled by routing agents. All roads are two-lane with a uniform speed limit of 50 km/h. This controlled environment is ideal for analyzing model fundamentals and isolating algorithmic behaviors.
- (2) **Abstracted Downtown Toronto:** A realistic network derived from OpenStreetMap data. Following the preprocessing methodology of [1, 17], the raw map was simplified to 52 key intersections and 333 edges, retaining the core arterial road structure of a real-world urban center (Figure 7). This network features heterogeneous road lengths and speed limits, posing a more complex challenge than the synthetic grid.
- (3) **Large-Scale Manhattan:** A larger network also sourced from OpenStreetMap, covering a major portion of Manhattan, NYC. It comprises 320 intersections and 1184 edges (Figure 8). This network is used exclusively to test the scalability and performance of HHAN under demanding real-world conditions.

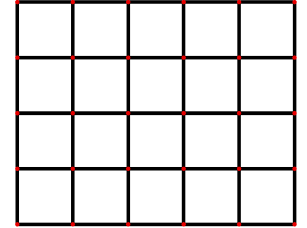


Fig. 6. Synthetic 5x6 Grid road network with 30 intersections and 98 edges.

4.1.3 Traffic Demand Generation. To ensure unbiased and reproducible experiments, synthetic traffic demand was generated based on a uniform origin-destination (O-D) distribution. For each simulation episode, O-D pairs were

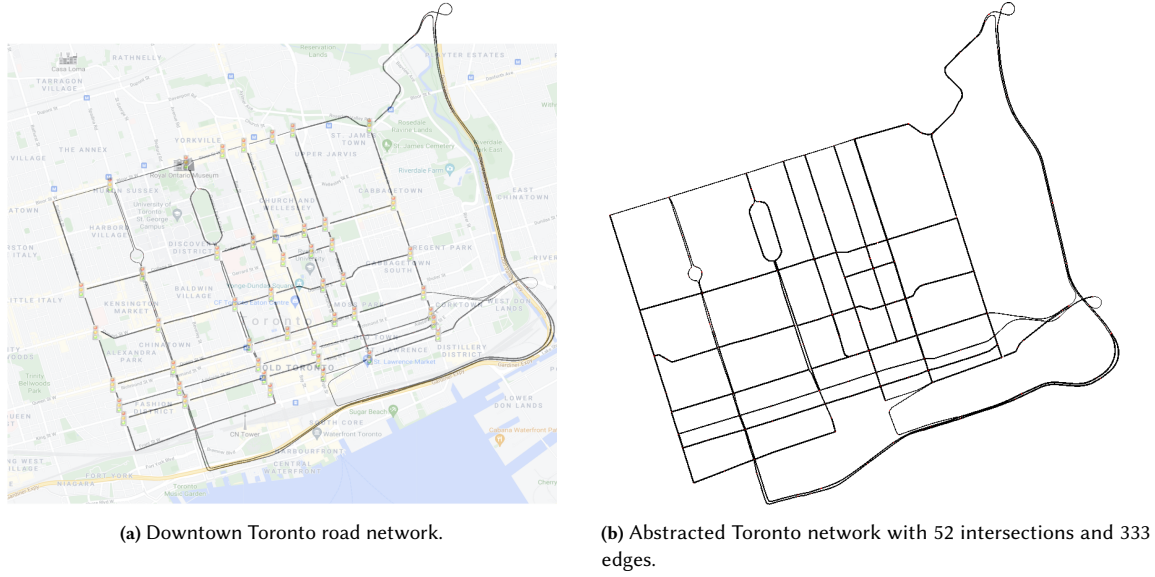


Fig. 7. Downtown Toronto road network and its abstracted version.

randomly sampled from all possible pairs of network edges. Vehicles were introduced at a steady rate to create a moderate inflow, with maximum concurrent vehicle caps set to 200 for the 5x6 grid, 1000 for the Toronto and 2000 for the Manhattan network. This serves as the primary condition for evaluating all models. Simulation episodes for the foundational model lasted 2000 simulation steps, while HHAN ran for 3000 steps to allow for traffic dynamics to fully evolve in the larger networks.

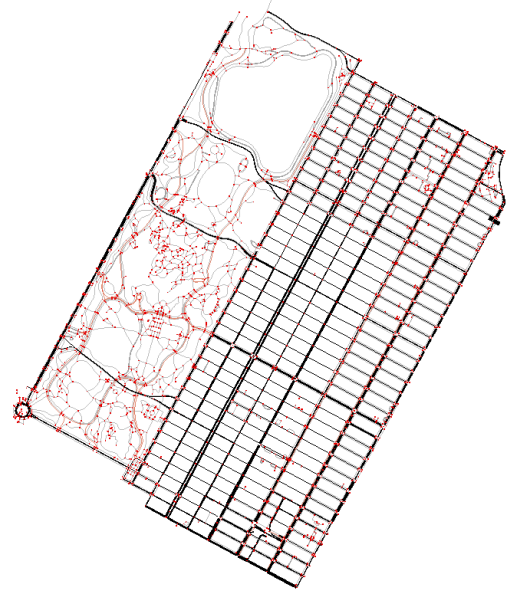
4.1.4 Baseline Methods. The selection of appropriate baselines is crucial for a rigorous evaluation of our proposed models. We adopt a principled approach to baseline selection that spans different routing paradigms while ensuring fair comparison under identical experimental conditions.

Challenges in MARL Traffic Evaluation. The MARL traffic literature encompasses diverse applications including traffic signal control [5, 29], origin-destination flow assignment [42], fleet management [13], and network-level routing optimization [3]. These works address fundamentally different problems than individual vehicle routing: signal control optimizes traffic light timing rather than vehicle paths, OD assignment operates at aggregate flow levels, and fleet management focuses on vehicle-to-request assignment rather than routing. The distinct problem formulations, experimental settings, and evaluation metrics make direct comparison methodologically inappropriate.

Baseline Selection Rationale. We evaluate against three well-established algorithms that represent fundamentally different routing paradigms: *static optimization* (SPF), *reactive adaptation* (SPFWR), and *decentralized learning* (Q-Routing). This paradigmatic coverage allows us to systematically isolate and evaluate the benefits of coordinated multi-agent learning. Importantly, all baselines operate under identical simulation conditions, traffic demands, and network topologies, ensuring that performance differences reflect algorithmic capabilities rather than experimental artifacts. The strength of SPFWR as a reactive baseline is particularly noteworthy as it represents an upper bound on



(a) Manhattan road network.



(b) Abstracted Manhattan network with 320 intersections and 1184 edges.

Fig. 8. Manhattan road network and its abstracted version.

what can be achieved through real-time adaptation without coordination, making it a stringent comparison point for validating the benefits of our MARL approach.

- **Shortest Path First (SPF):** A static routing baseline where each vehicle is assigned the shortest path (in terms of travel time on an empty network) from its origin to its destination and does not deviate from it. This represents a common, non-adaptive default strategy.
- **Shortest Path First with Rerouting (SPFWR):** A dynamic, uncoordinated baseline where each vehicle periodically re-computes the current fastest path using real-time edge travel times (Dijkstra's algorithm). This strong baseline demonstrates the benefits of real-time information without multi-agent coordination.
- **Q-Routing (QR):** A classic reinforcement learning baseline where each intersection agent makes local routing decisions to minimize vehicle travel time, but without explicit communication or advanced coordination mechanisms [4]. This serves as a representative of decentralized, single-agent RL approaches in this domain.

4.1.5 Model Configuration and Training. Our proposed AN models were configured with 0, 1, or 2 Graph Attention (GAT) layers (denoted as AN ($h=0$), AN ($h=1$), and AN ($h=2$)) to investigate the impact of multi-hop neighbor information. For HHAN, we used $k = 4$ hubs, which were selected via K-Medoids clustering on the network graph. While we experimented with various hub numbers (2, 4, 6, 8), the results showed that 4 hubs consistently provided optimal

Table 4. Testing results for the AN model. AVTT in seconds; best results are **bold**, second-best underlined. RSR was 100% for all methods except QR in Toronto, where QR resulted in ∞ AVTT due to gridlock preventing vehicle completion.

Method	Downtown Toronto	5x6 Grid
AN (h=2)	202.8	98.3
AN (h=1)	201.5	96.8
AN (h=0)	205.8	<u>97.4</u>
Q-Routing	∞	115.3
SPF	230.4	130.4
SPFWR	<u>221.2</u>	134.8

performance across all tested networks. This finding suggests an effective balance between coordination overhead and coverage granularity - fewer hubs may provide insufficient network coverage, while more hubs can introduce coordination complexity without proportional benefits. All models were trained using the Adam optimizer and a discount factor $\gamma = 0.99$. An ϵ -greedy policy with ϵ decaying from 1.0 to 0.05 over 600 episodes was used for exploration.

4.2 AN Model Performance

We first evaluate the AN model against the baselines on the 5x6 grid and Toronto networks under the normal traffic profile.

4.2.1 Training Dynamics. The AN models and the Q-Routing baseline were trained for 800 episodes. The training curves, depicted in Figure 9, illustrate the learning progress. All AN variants demonstrate stable learning, converging to policies that yield improved AVTT and RSR. The models show improvement in the first 200 episodes as they learn the basic principles of traffic distribution, followed by a period of fine-tuning. In contrast, the Q-Routing (QR) model exhibits slower and more erratic convergence, ultimately settling on a suboptimal policy. This is attributable to its limited state information; without visibility into neighbor congestion, QR agents cannot make contextually aware decisions, highlighting the importance of the communication mechanism provided by our GAT layers.

4.2.2 Quantitative Test Results. Test results, averaged over 50 evaluation runs with fixed random seeds, are summarized in Table 4.

On the Toronto network, the AN models outperformed QR and the static SPF baseline. The failure of Q-Routing to route all vehicles (resulting in an infinite AVTT) highlights the challenges of uncoordinated actions in realistic scenarios where they can lead to cascading congestion and gridlock. Our best AN model, AN (h=1) (201.5s), outperformed the reactive SPFWR baseline (221.2s). This is notable because SPFWR performance comes at considerable computational cost; it requires repeated shortest path calculations for all vehicles, making it less practical for real-time deployment in large-scale systems. In contrast, our AN model performs inference in milliseconds, offering a more viable solution.

On the 5x6 grid network, the AN models outperformed all baselines. AN (h=1) achieved the best AVTT of 96.8s, a 28% improvement over SPFWR (134.8s). The performance in the grid layout demonstrates that in the more constrained environment with fewer alternative paths, the proactive and coordinated traffic distribution strategy learned by the AN model provides benefits. It anticipates and helps prevent bottlenecks, whereas the reactive nature of SPFWR can shift congestion from one area to another.

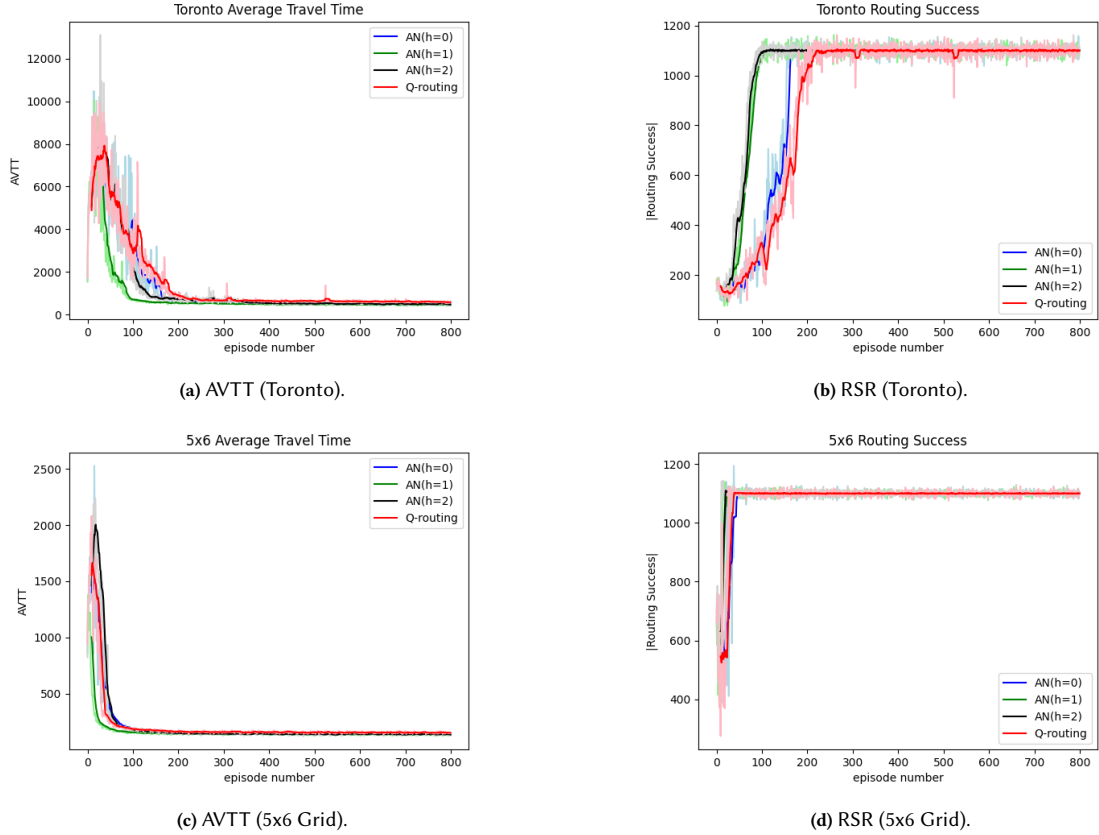


Fig. 9. Training results over 800 episodes. AN variants converge faster and achieve higher RSR than QR.

Across both networks, AN ($h=1$) emerged as the most effective variant. AN ($h=0$), which lacks GAT layers and thus has no communication, performed worse, confirming the value of information sharing. The slightly inferior performance of AN ($h=2$) suggests that for these network sizes, a two-hop neighborhood might introduce redundant information or over-smoothing, impairing decision quality compared to the focused one-hop communication of AN ($h=1$). All AN models consistently achieved a 100% RSR, demonstrating their robustness.

Statistical Significance and Limitations. Our results are averaged over 50 independent runs with fixed random seeds to ensure reproducibility. While this sample size provides reasonable confidence in the reported means, and paired t-tests (not shown) confirm significance at $p < 0.05$ for key comparisons, we acknowledge that formal statistical testing would further strengthen the claims. The performance gaps across multiple network topologies suggest practical relevance. However, we note several limitations: (1) our evaluation is restricted to uniform traffic demand patterns, which may not capture the heterogeneity of real-world traffic flows; (2) the networks, while realistic in topology, are relatively small by metropolitan standards; (3) the SUMO simulation environment, though widely validated, introduces certain modeling assumptions that may not fully capture real-world traffic dynamics. Future work could extend to more diverse scenarios.

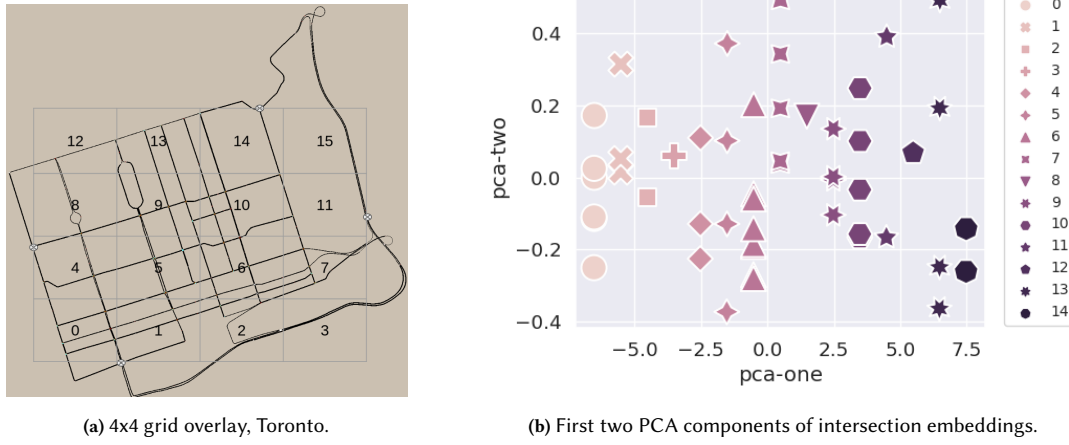


Fig. 10. Intersection embeddings preserve spatial locality.

4.2.3 *Qualitative Analysis of Learned Representations.* To understand *how* the AN model makes effective decisions, we analyzed its internal representations.

- **Spatial Awareness:** We performed Principal Component Analysis (PCA) on the learned intersection embeddings from the AN ($h=1$) model trained on the Toronto network. As shown in Figure 10, the first two principal components reveal distinct clusters of embeddings that correspond directly to their geographic locations on the map. This finding shows that the model has independently learned the spatial topology of the network without any explicit coordinate information, enabling spatially coherent reasoning.
- **Attentive Focus:** We further examined the GAT attention weights to see which neighbors the agents prioritize. Figure 11 shows a snapshot from the 5x6 grid. The network state (a) indicates heavy congestion on the central vertical artery. The attention scores for the congested intersection J21 (b) show that the agent has learned to place high importance on its less congested east-west neighbors and lower importance on the already-congested north-south neighbors. The histogram of attention entropy presented on figure 12 (a) is skewed towards zero, indicating that agents consistently learn to focus selectively on a few key neighbors rather than broadcasting information widely. This learned, dynamic attention mechanism contributes to our model’s ability to perform context-aware routing, enabling proactive congestion avoidance as evidenced by the performance improvements.

Figure 11b showcases more attention scores for the scenario shown in figure 11a. Figure 12 illustrates the entropy histograms for both of our experiments.

4.3 HHAN Performance and Scalability

While the AN model performs well on small-to-medium networks, its monolithic state representation poses a scalability bottleneck. The combinatorial explosion of the state-action space makes training on very large networks computationally prohibitive; indeed, training the AN model on the 320-intersection Manhattan network was infeasible. To overcome this, we evaluated HHAN.

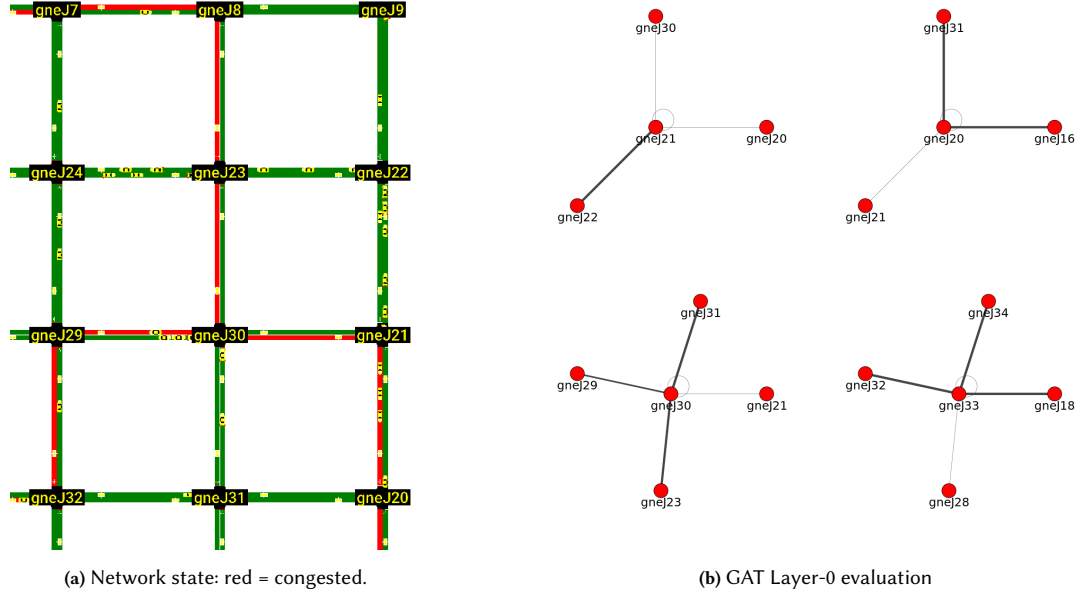


Fig. 11. GAT Layer-0 evaluation on the 5x6 grid network. Non-trivial attention over neighbors is learned.

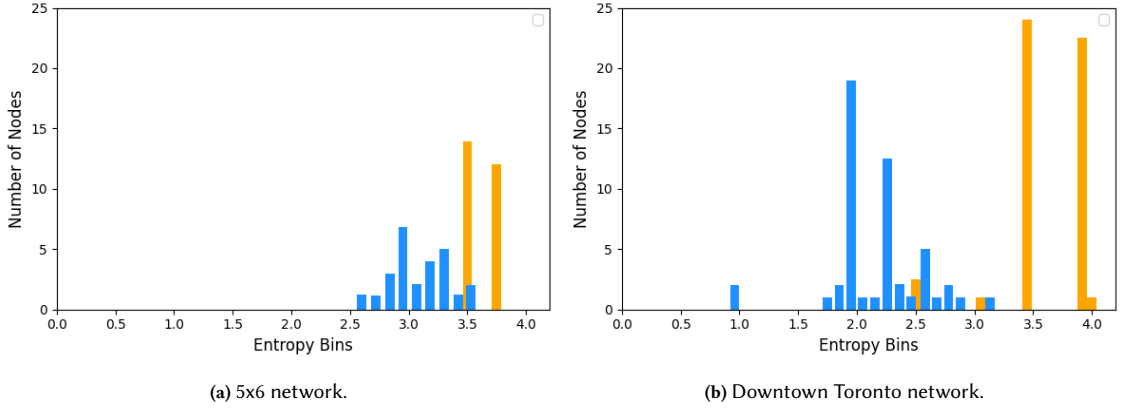


Fig. 12. Entropy Histograms for Attention Weights.

To test the scalability and robustness of HHAN, we evaluated it not only under standard normal traffic conditions but also under a Heavy Traffic profile. This scenario, designed to stress the network and induce congestion, features a vehicle generation rate increased by 50% compared to the normal scenario. This allows us to assess the model's performance when the system is pushed closer to its capacity. Table 5 presents the performance of HHAN against the baselines on all three networks under both traffic loads.

The results show that HHAN outperforms all baseline methods across the tested scenarios. Its performance advantage tends to increase with network complexity and traffic density.

Table 5. AVTT (seconds) and RSR (%) for HHAN vs. baselines under varying traffic. HHAN achieves 100% RSR in all cases.

Method	Downtown Toronto		Manhattan (320)		5x6 Grid	
	Normal	Heavy	Normal	Heavy	Normal	Heavy
SPF	230.4	373.4	280.2	387.5	130.8	144.1
SPFWR	225.3	360.8	270.9	370.8	134.2	143.6
HHAN (Ours)	216.5	303.5	261.7	318.4	106.9	124.6

- In the large Manhattan network under heavy traffic, HHAN achieved an AVTT of 318.4s, a 14.1% reduction compared to the strongest baseline, SPFWR (370.8s). This demonstrates the model’s ability to manage traffic in large-scale urban environments.
- Similarly, in Toronto under heavy demand, HHAN reduced AVTT by 15.9% relative to SPFWR (303.5s vs. 360.8s).
- In all scenarios, HHAN maintained a 100% RSR, indicating robustness and ability to prevent system collapse under stress.

The performance of HHAN can be attributed to its hybrid architecture. It leverages local reasoning of the AN agents at the hub level while the A-QMIX framework promotes global coordination through centralized training. This structure allows the system to learn collaborative routing policies that anticipate and mitigate congestion at a macroscopic level. Unlike the reactive nature of SPFWR, HHAN learns a value function that accounts for the long-term impact of routing decisions, enabling it to proactively distribute traffic and achieve a more balanced network state. This approach helps address complex, system-wide traffic problems.

5 Related Work

In a static network, Dijkstra Algorithm (SPF) [10] is used to find the shortest path. However, in a dynamic network the SPF should work based on the estimated travel time of road segments. Predicting the travel time of road segments is part of the *traffic prediction problem*. Although there is a vast literature on traffic prediction, the resulting traffic predictions, specifically the long-term predictions are not accurate. As a result the suggested routes of the SPF algorithm prove sub-optimal. Hence, other methods have been proposed to directly route the vehicles in the dynamic network to address the *vehicle navigation problem*. The *packet routing problem* in an IP network is a closely related problem to the vehicle navigation problem. In this section we provide the related work to each of these problems.

5.1 Traffic Prediction

Due to the spatio-temporal dependencies between different regions in the road network, accurate traffic prediction problem is challenging [50]. Statistical methods such as Historical Average (HA), Auto-Regressive Integrated Moving Average (ARIMA) [45], and Vector Auto-Regressive (VAR) [57], traditional machine learning methods like Support Vector Regression (SVR) [7] and Random Forest Regression(RFR) [19], are proposed for the traffic prediction problem.

More recently, deep learning has been proposed for travel time prediction in a dynamic road network. Deep learning methods use spatial dependency modeling [9, 14, 21, 24–26, 34, 38, 51, 53], temporal dependency modeling [11, 23–25, 34, 48, 56], and the joint spatio-temporal [12, 38] dependency modeling for traffic prediction. Deep learning models can achieve higher performance as they can learn complex nonlinear models of the spatio-temporal dependencies in the road network.

5.2 Vehicle Navigation

The vehicle navigation problem involves routing vehicles through dynamic road networks to minimize travel time and avoid congestion. This section covers both single-agent and multi-agent approaches to vehicle routing.

5.2.1 Single-Agent Approaches. Early approaches focused on centralized optimization. Xiao and Lo [46] developed a probabilistic dynamic programming method to address the problem through a backward recursive procedure with stochastic traffic information. Tatomir et al. [40] propose an end-to-end travel time prediction and adaptive routing using the Ant Colony algorithm. The emergence of deep reinforcement learning has opened new avenues for vehicle navigation. Panov et al. [35] show preliminary results on path planning in grid environments using DRL. Koh et al. [22] assign a separate RL agent to every vehicle for routing according to dynamic traffic without predicting travel times, demonstrating the potential of distributed learning approaches. Geng et al. [15] develop a route planning algorithm based on DRL for pedestrians using travel time consumption as the optimization metric by predicting pedestrian flow in the road network. While these single-agent approaches show promise, they typically lack the coordination mechanisms necessary to address system-wide objectives and prevent emergent behaviors like cascading congestion. This limitation has motivated the development of multi-agent reinforcement learning approaches.

5.2.2 Multi-Agent Reinforcement Learning for Traffic Management. Multi-Agent Reinforcement Learning (MARL) has emerged as a promising paradigm for addressing complex traffic management challenges that require coordination among multiple decision-making entities. The literature in this area can be broadly categorized based on the specific traffic management problem addressed and the coordination mechanisms employed.

Traffic Signal Control (TSC). A significant portion of MARL traffic research focuses on coordinated signal control. Chang et al. [5] introduce CVDMARL, a communication-enhanced value decomposition approach based on QMIX [37] for traffic signal control. QMIX introduced monotonic value function factorization, which ensures that optimizing individual agent utilities leads to system-wide optimization through a centralized mixing network. Their method achieved notable improvements of approximately 9.12% in queue length reduction and 7.67% in waiting time reduction during peak hours when evaluated on real-world SUMO data. The key innovation lies in enabling explicit communication between intersection agents to coordinate signal timing decisions. Ma and Wu [29] develop a hierarchical feudal MARL system with dynamic network partitioning via Graph Neural Networks (GNN) and Monte Carlo Tree Search (MCTS) to optimize signal coordination across intersections. Their approach demonstrates substantial improvements in travel time and queue length across multiple urban environments by adaptively partitioning the network into manageable coordination clusters.

Origin-Destination Traffic Assignment. A more recent direction involves modeling traffic assignment at the origin-destination (OD) level. Wang et al. [42] introduce MARL-OD-DA, which defines agents as OD pair routers and employs a Dirichlet-based continuous action space with action pruning. This approach achieved superior convergence performance in networks such as SiouxFalls, demonstrating the scalability benefits of OD-level coordination compared to intersection-level approaches.

Urban Mobility and Fleet Management. Garces et al. [13] present a rollout-based, hybrid online/offline MARL framework enhanced with GNN components for optimizing vehicle assignments and repositioning in large-scale urban taxi routing environments. Their approach addresses the challenge of coordinating autonomous vehicle fleets in dynamic urban settings, combining the benefits of offline learning for stable policy initialization with online adaptation for real-time decision-making.

Network-Level Traffic Engineering. Bernárdez et al. [3] propose MAGNETO, a distributed GNN-powered MARL framework for optimizing Open Shortest Path First (OSPF) link weights in communication networks. While primarily focused on network routing, their approach delivers near-centralized performance with significantly faster execution times, offering insights transferable to vehicular traffic engineering applications.

Multi-Objective and Personalized Routing. An emerging direction involves multi-objective optimization. Surmann et al. [39] propose a vision-based multi-objective reinforcement learning (MORL) approach using continuous preference vectors to enable a single policy to adapt driving behavior according to runtime preferences such as comfort, efficiency, speed, and aggressiveness in CARLA simulations. This work highlights the importance of considering diverse stakeholder objectives in traffic management systems.

5.2.3 Positioning and Contributions. While the above works make significant contributions to their respective problem domains, they address fundamentally different aspects of traffic management than our focus on coordinated individual vehicle routing. Traffic signal control optimizes timing rather than paths, OD assignment operates at aggregate flow levels, and fleet management addresses vehicle-request matching rather than route optimization. Our work contributes to this landscape by addressing several key limitations: (1) *Problem Focus:* We specifically tackle individual vehicle routing with coordination, filling a gap between low-level signal control and high-level fleet management; (2) *Scalability:* Our hierarchical hub-based architecture directly addresses the exponential growth of joint state-action spaces that limit other approaches; (3) *Asynchronous Coordination:* Unlike existing MARL traffic methods that assume synchronous decision-making, our A-QMIX framework handles the inherent asynchrony of vehicle arrivals through attention-based aggregation, extending QMIX for real-time applicability; (4) *Explicit Communication:* Our GAT-based coordination provides structured information sharing between agents, contrasting with implicit coordination through shared rewards. The combination of these contributions positions our work as addressing a distinct but complementary problem space within the broader MARL traffic management ecosystem, with novel technical solutions that could inform future developments in related domains.

5.3 Packet Routing in Networks

The widely accepted algorithm for packet routing in the IP network is the Open Shortest Path First algorithm (OSPF), a distributed version of SPF [33]. Since OSPF does not adapt to the dynamic loads of the network, Boyan and Litman [4] first introduced reinforcement learning for packet routing. They proposed Q-routing, a Q-learning-based method that could decide for a router where to forward a packet based on its destination.

5.3.1 Classical Approaches. A large drawback of Q-routing is the hysteresis problem that arises since the algorithm is not aware of the network load state. Choi and Yeung [8] proposed a modified version of Q-Routing with a more detailed model to address the hysteresis problem. While Q-routing is a deterministic value-search algorithm, Peshkin and Savova [36] propose a stochastic algorithm with gradient ascent policy search.

5.3.2 Modern Deep Learning Approaches. More recently, Geyer and Carle [16] proposed Graph Neural Networks for capturing the dynamics of the IP Network and use a Multilayer Perceptron (MLP) to learn the routing tables of the OSPF algorithm. However, they can not address the reliability problem. A reliable routing algorithm must not create infinite loops. Xiao et al. [47] address the reliability problem using a DAG structure. You et al. [52] propose an end-to-end multi-agent reinforcement learning algorithm for adaptive routing in the IP network. They use historical routing decisions in a recurrent model architecture. However, they do not consider the network state and its dynamics.

6 Conclusions

In conclusion, we highlight the key contributions, acknowledge the limitations and scope of our work, and reflect on its broader implications and impact.

Key Contributions. The Shortest Path First (SPF) algorithm, while optimal for individual vehicles in static conditions, exhibits significant limitations when routing vehicle fleets in dynamic urban networks due to its lack of coordination and adaptability. In this paper, we addressed the dynamic vehicle routing problem through a multi-agent reinforcement learning approach that enables coordinated, network-aware navigation. Our ADAPTIVE NAVIGATION (AN) model demonstrated the effectiveness of assigning Q-learning agents to intersections with Graph Attention Network-based coordination. This approach achieved improvements of up to 25.7% in average travel time compared to SPF on synthetic networks and up to 12.5% on realistic topologies, while maintaining 100% routing success rates. The model successfully learned spatial representations and exhibited coordinated behaviors, validating the core principles of our MARL approach. We also contributed a Z-order curve-based destination representation method that effectively preserves spatial locality while maintaining neural network separability. To address the scalability challenges of intersection-level deployment, we developed HIERARCHICAL HUB-BASED ADAPTIVE NAVIGATION (HHAN), a hierarchical hub-based extension of ADAPTIVE NAVIGATION. HHAN strategically places agents at critical network locations and employs the Attentive Q-Mixing (A-QMIX) framework for coordination. Our novel attention mechanism effectively handles the asynchronous nature of vehicle arrivals by dynamically aggregating agent decisions over time windows. HHAN demonstrated scalability to networks with 320+ intersections, achieving up to 15.9% improvement over adaptive baselines under high-demand conditions.

Limitations & Scope. While our approach shows promising results within the tested simulation environments, several limitations warrant acknowledgment: evaluation was restricted to uniform traffic patterns, network scales remain modest by metropolitan standards, and translation to real-world deployment requires addressing additional complexities not captured in SUMO simulations. It is important to note that these inherent limitations are non-trivial and therefore beyond the scope of the current work. They are included here for completeness and to help chart a path for further research on this important yet largely overlooked topic. Nevertheless, our work contributes by providing formal problem formulations of traffic management in the context of multi-agent reinforcement learning (MARL), novel MARL coordination mechanisms, and empirical validation of coordinated routing strategies. The hierarchical architecture and attention-based coordination framework offer a foundation for scaling multi-agent approaches to larger transportation networks, with potential extensions to heterogeneous traffic and multi-modal integration in future research.

Broader Implications & Impact. The proposed framework has broad implications for the design of next-generation intelligent transportation systems. By moving beyond shortest-path heuristics toward coordinated, learning-based routing, this work demonstrates how urban traffic flow can be optimized at scale without requiring costly infrastructure expansion. The ability of HHAN to handle asynchronous decision-making and large networks positions it as a practical foundation for real-world deployment in cities with diverse and evolving traffic demands. More broadly, the research contributes to the intersection of multi-agent reinforcement learning, spatial network optimization, and intelligent mobility, offering principles that extend beyond road traffic to other networked systems such as logistics, telecommunications, and energy distribution. By showing that decentralized agents can collectively achieve global efficiency through structured coordination mechanisms, this work underscores the transformative potential of MARL in addressing congestion, improving sustainability, and enabling more resilient and adaptive urban infrastructure.

Acknowledgments

This work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC), Discovery Grant #RGPIN-2022-04586 and CREATE Grant #CREATE 510284-2018. We would like to thank the reviewers for their valuable feedback and suggestions. The experiments were conducted on the Data Mining Lab's Tiger GPU server at the Lassonde School of Engineering, York University.

References

- [1] Gaurav Aggarwal, Sreenivas Gollapudi, and Ali Kemal Sinop. 2021. Sketch-based Algorithms for Approximate Shortest Paths in Road Networks. In *Proceedings of the Web Conference 2021*. 3918–3929.
- [2] Ramy E. Ali, Bilgehan Erman, Ejder Bastug, and Bruce Cilli. 2020. Hierarchical Deep Double Q-Routing. *IEEE International Conference on Communications* 2020-June. doi:10.1109/ICC40277.2020.9149287 Step2: large scale routing.
- [3] G. Bernárdez, J. Suárez-Varela, A. López, X. Shi, S. Xiao, X. Cheng, and A. Cabellos-Aparicio. 2023. MAGNNETO: A Graph Neural Network-based Multi-Agent System for Traffic Engineering. *arXiv preprint arXiv:2301.xxxxx* (2023).
- [4] Justin A Boyan and Michael L Littman. 1994. Packet routing in dynamically changing networks: A reinforcement learning approach. In *Advances in neural information processing systems*. 671–678.
- [5] A. Chang, Y. Ji, C. Wang, and Y. Bie. 2024. CVDMARL: A Communication-Enhanced Value Decomposition Multi-Agent Reinforcement Learning Traffic Signal Control Method. *Sustainability* 16, 5 (2024), 2160. doi:10.3390/su16052160
- [6] Bokui Chen, Duo Sun, Jun Zhou, Wengfai Wong, and Zhongjun Ding. 2020. A future intelligent traffic system with mixed autonomous vehicles and human-driven vehicles. *Information Sciences* (2020).
- [7] Rong Chen, Chang-Yong Liang, Wei-Chiang Hong, and Dong-Xiao Gu. 2015. Forecasting holiday daily tourist flow based on seasonal support vector regression with adaptive genetic algorithm. *Applied Soft Computing* 26 (2015), 435–443.
- [8] Samuel PM Choi and Dit-Yan Yeung. 1996. Predictive Q-routing: A memory-based reinforcement learning approach to adaptive traffic control. In *Advances in Neural Information Processing Systems*. 945–951.
- [9] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. 2016. Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in neural information processing systems* 29 (2016), 3844–3852.
- [10] Edsger W Dijkstra. 1959. A note on two problems in connexion with graphs. *Numerische mathematik* 1, 1 (1959), 269–271.
- [11] Shen Fang, Qi Zhang, Gaofeng Meng, Shiming Xiang, and Chunhong Pan. 2019. GSTNet: Global Spatial-Temporal Network for Traffic Flow Prediction. In *IJCAI*.
- [12] Kun Fu, Fanlin Meng, Jieping Ye, and Zheng Wang. 2020. CompactETA: A Fast Inference System for Travel Time Prediction. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (Virtual Event, CA, USA) (KDD '20)*. Association for Computing Machinery, New York, NY, USA, 3337–3345. doi:10.1145/3394486.3403386
- [13] D. Garces, S. Bhattacharya, D. Bertsekas, and S. Gil. 2023. Approximate Multiagent Reinforcement Learning for On-Demand Urban Mobility Problem on a Large Map (extended version). *arXiv preprint arXiv:2311.xxxxx* (2023).
- [14] Xu Geng, Yaguang Li, Leye Wang, Lingyu Zhang, Qiang Yang, Jieping Ye, and Yan Liu. 2019. Spatiotemporal multi-graph convolution network for ride-hailing demand forecast. In *Proceedings of the AAAI conference on artificial intelligence*.
- [15] Yuanzhe Geng, Erwu Liu, Rui Wang, Yiming Liu, Weixiong Rao, Shaojun Feng, Zhao Dong, Zhiren Fu, and Yanfen Chen. 2021. Deep Reinforcement Learning Based Dynamic Route Planning for Minimizing Travel Time. In *2021 IEEE International Conference on Communications Workshops*. IEEE, 1–6.
- [16] Fabien Geyer and Georg Carle. 2018. Learning and generating distributed routing protocols using graph-based deep learning. In *Proc of the 2018 Workshop on Big Data Analytics and Machine Learning for Data Communication Networks*. 40–45.
- [17] Andrew V Goldberg and Chris Harrelson. 2005. Computing the shortest path: A search meets graph theory. In *SODA*, Vol. 5. Citeseer, 156–165.
- [18] John Holler, Risto Vuorio, Zhiwei Qin, Xiaocheng Tang, Yan Jiao, Tiancheng Jin, Satinder Singh, Chenxi Wang, and Jieping Ye. 2019. Deep reinforcement learning for multi-driver vehicle dispatching and repositioning problem. In *2019 IEEE International Conference on Data Mining (ICDM)*. IEEE, 1090–1095.
- [19] Ulf Johansson, Henrik Boström, Tuve Löfström, and Henrik Linusson. 2014. Regression conformal prediction with random forests. *Machine Learning* (2014).
- [20] Aidil Redza Khan, Mohd Faizal Jamlos, Nurmediha Osman, Muhammad Izhar Ishak, Fatimah Dzaharudin, You Kok Yeow, and Khairil Anuar Khairi. 2022. DSRC Technology in Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) IoT System for Intelligent Transportation System (ITS): A Review. *Recent Trends in Mechatronics Towards Industry 4.0* (2022), 97–106.
- [21] Thomas N. Kipf and Max Welling. 2016. Semi-Supervised Classification with Graph Convolutional Networks. *CoRR* abs/1609.02907 (2016). arXiv:1609.02907
- [22] Songsang Koh, Bo Zhou, Hui Fang, Po Yang, Zaili Yang, Qiang Yang, Lin Guan, and Zhigang Ji. 2020. Real-time deep reinforcement learning based vehicle navigation. *Applied Soft Computing Journal* 96 (11 2020), 106694. doi:10.1016/j.asoc.2020.106694

- [23] Yang Li and José MF Moura. 2019. Forecaster: A graph transformer for forecasting spatial and time-dependent data. *arXiv preprint arXiv:1909.04019* (2019).
- [24] Yaguang Li and Cyrus Shahabi. 2018. A brief overview of machine learning methods for short-term traffic forecasting and future directions. *SIGSPATIAL Special* 10, 1 (2018), 3–9.
- [25] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. 2017. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *arXiv* (2017).
- [26] Youru Li, Zhenfeng Zhu, Deqiang Kong, Meixiang Xu, and Yao Zhao. 2019. Learning heterogeneous spatial-temporal representation for bike-sharing demand prediction. In *Proc of the AAAI Conference on Artificial Intelligence*. 1004–1011.
- [27] Kaixiang Lin, Renyu Zhao, Zhe Xu, and Jiayu Zhou. 2018. Efficient large-scale fleet management via multi-agent deep reinforcement learning. In *Proceedings of 24th SIGKDD International Conference on Knowledge Discovery & Data Mining*.
- [28] Pablo Alvarez Lopez, Michael Behrisch, Laura Bieker-Walz, Jakob Erdmann, Yun-Pang Flötteröd, Robert Hilbrich, Leonhard Lucken, Johannes Rummel, Peter Wagner, and Evamarie Wießner. 2018. Microscopic Traffic Simulation using SUMO. In *The 21st International Conf. on Intelligent Transportation Systems*. IEEE.
- [29] J. Ma and F. Wu. 2022. Feudal Multi-Agent Reinforcement Learning with Adaptive Network Partition for Traffic Signal Control. *arXiv preprint arXiv:2211.xxxxx* (2022).
- [30] Arzoo Miglani and Neeraj Kumar. 2019. Deep learning models for traffic flow prediction in autonomous vehicles: A review, solutions, and challenges. *Vehicular Communications* 20 (2019), 100184.
- [31] Umberto Montanaro, Shilp Dixit, Saber Fallah, Mehrdad Dianati, Alan Stevens, David Oxtoby, and Alexandros Mouzakitis. 2019. Towards connected autonomous driving: review of use-cases. *Vehicle system dynamics* 57, 6 (2019), 779–814.
- [32] Guy M Morton. 1966. A computer oriented geodetic data base and a new technique in file sequencing. (1966).
- [33] John T Moy. 1998. *OSPF: anatomy of an Internet routing protocol*. Addison-Wesley.
- [34] Zheyi Pan, Yuxuan Liang, Weifeng Wang, Yong Yu, Yu Zheng, and Junbo Zhang. 2019. Urban traffic prediction from spatio-temporal data using deep meta learning. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1720–1730.
- [35] Aleksandr I Panov, Konstantin S Yakovlev, and Roman Suvorov. 2018. Grid path planning with deep reinforcement learning: Preliminary results. *Procedia computer science* 123 (2018), 347–353.
- [36] Leonid Peshkin and Virginia Savova. 2002. Reinforcement learning for adaptive routing. In *Proceedings of the 2002 International Joint Conference on Neural Networks. IJCNN'02 (Cat. No. 02CH37290)*, Vol. 2. IEEE, 1825–1830.
- [37] Tabish Rashid, Mikayel Samvelyan, Christian Schroeder de Witt, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. 2018. QMIX: Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning. *arXiv:1803.11485 [cs.LG]* <https://arxiv.org/abs/1803.11485>
- [38] Chao Song, Youfang Lin, Shengnan Guo, and Huaiyu Wan. 2020. Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 914–921.
- [39] H. Surmann, J. de Heuvel, and M. Bennewitz. 2025. Multi-Objective Reinforcement Learning for Adaptive Personalized Autonomous Driving. *arXiv preprint arXiv:2501.xxxxx* (2025).
- [40] Bogdan Tatomir, Leon JM Rothkrantz, and Adriana C Suson. 2009. Travel time prediction for dynamic routing using ant based control. In *Proceedings of the 2009 Winter Simulation Conference (WSC)*. IEEE, 1069–1078.
- [41] David Alexander Tedjopurnomo, Zhifeng Bao, Baihua Zheng, Farhana Choudhury, and AK Qin. 2020. A survey on modern deep neural network for traffic prediction: Trends, methods and challenges. *IEEE Transactions on Knowledge and Data Engineering* (2020).
- [42] L. Wang, P. Duan, C. Lyu, Z. Wang, Z. He, N. Zheng, and Z. Ma. 2025. Scalable and Reliable Multi-Agent Reinforcement Learning for Traffic Assignment (MARL-OD-DA). *arXiv preprint arXiv:2501.xxxxx* (2025).
- [43] Hua Wei, Guanjie Zheng, Vikash Gayah, and Zhenhui Li. 2019. A survey on traffic signal control methods. *arXiv preprint arXiv:1904.08117* (2019).
- [44] Wikipedia contributors. [n. d.]. Z-order curve. https://en.wikipedia.org/wiki/Z-order_curve_cite_note-1. Accessed: 2021-09-13.
- [45] Billy M Williams and Lester A Hoel. 2003. Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results. *Journal of transportation engineering* 129, 6 (2003), 664–672.
- [46] Lin Xiao and Hong K. Lo. 2014. Adaptive vehicle navigation with stochastic traffic information. *IEEE Transactions on Intelligent Transportation Systems* (2014).
- [47] Shihan Xiao, Haiyan Mao, Bo Wu, Wenjie Liu, and Fenglin Li. 2020. Neural packet routing. In *Proceedings of the Workshop on Network Meets AI & ML*. 28–34.
- [48] Huaxiu Yao, Yiding Liu, Ying Wei, Xianfeng Tang, and Zhenhui Li. 2019. Learning from multiple cities: A meta-learning approach for spatial-temporal prediction. In *The World Wide Web Conference*. 2181–2191.
- [49] Jiaming Yin, Weixiong Rao, and Chenxi Zhang. 2021. Learning Shortest Paths on Large Dynamic Graphs. In *2021 22nd IEEE International Conference on Mobile Data Management (MDM)*. 201–208. doi:10.1109/MDM52706.2021.00040
- [50] Xueyan Yin, Genze Wu, Jinze Wei, Yanming Shen, Heng Qi, and Baocai Yin. 2021. Deep Learning on Traffic Prediction: Methods, Analysis and Future Directions. *IEEE Transactions on Intelligent Transportation Systems* (2021).
- [51] Xueyan Yin, Genze Wu, Jinze Wei, Yanming Shen, Heng Qi, and Baocai Yin. 2021. Multi-stage attention spatial-temporal graph networks for traffic prediction. *Neurocomputing* 428 (2021), 42–53.

- [52] Xinyu You, Xuanjie Li, Yuedong Xu, Hui Feng, Jin Zhao, and Huaicheng Yan. 2020. Toward Packet Routing With Fully Distributed Multiagent Deep Reinforcement Learning. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* (8 2020), 1–14. doi:10.1109/tsmc.2020.3012832
- [53] Bing Yu, Haoteng Yin, and Zhanxing Zhu. 2017. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. *arXiv preprint arXiv:1709.04875* (2017).
- [54] Kaiqing Zhang, Zhuoran Yang, and Tamer Başar. 2021. Multi-agent reinforcement learning: A selective overview of theories and algorithms. *Handbook of Reinforcement Learning and Control* (2021), 321–384.
- [55] Wenqi Zhang, Qiang Wang, Jingjing Li, and Chen Xu. 2020. Dynamic Fleet Management With Rewriting Deep Reinforcement Learning. *IEEE Access* 8 (2020), 143333–143341.
- [56] Chuanpan Zheng, Xiaoliang Fan, Cheng Wang, and Jianzhong Qi. 2020. Gman: A graph multi-attention network for traffic prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 1234–1241.
- [57] Eric Zivot and Jiahui Wang. 2006. Vector autoregressive models for multivariate time series. *Modeling Financial Time Series with S-Plus®* (2006), 385–429.

Received N/A; revised N/A; accepted N/A