Dynamic Route-Aware Graph Neural Networks for Accurate ETA Prediction

Guy Tordjman
Department of Mathematics and Computer Science
The Open University
University Road 1, Ra'anana, Israel
Email: guy.tordjman@openu.ac.il

Nadav Voloch
Department of Computer Engineering
Ruppin Academic Center
Emek Hefer, Israel
Email: nadavv@ruppin.ac.il

***Abstract*—Accurate Estimated Time of Arrival (ETA) prediction is a central challenge in intelligent transportation systems, navigation services, and mobility applications. Existing approaches based on static routing algorithms or trajectory-driven machine learning improve over simple baselines but often fail to account for drivers' intended routes and the dynamic evolution of traffic interactions. These limitations restrict their ability to deliver reliable estimates under congested and heterogeneous traffic conditions.This work presents a dynamic spatio-temporal graph neural network that integrates vehicle-level dynamics, junction states, and explicit route intent within a unified Mixture-of-Experts framework. The transportation network is represented as a multi-relational graph, combining static road edges with dynamic vehicle interactions and planned route segments. Temporal dependencies are modeled through sequential graph snapshots, enabling the system to capture both persistent infrastructure characteristics and evolving traffic behaviors. Evaluation on a large-scale simulated dataset shows that the proposed model reduces Mean Absolute Error (MAE) from a simple average-time baseline of 260 seconds to 46 seconds, corresponding to an 82% improvement. In contrast, representative trajectory- and graph-based baselines typically achieve only moderate gains in the range of 40–50%. These results underscore the importance of explicit route representation and dynamic spatio-temporal modeling for accurate and robust ETA prediction.**

***Index Terms*—ETA prediction, spatio-temporal graph neural networks, traffic forecasting, mixture of experts, intelligent transportation systems.**

I. Introduction

Estimated Time of Arrival (ETA) prediction is a fundamental component of modern navigation systems and intelligent transportation applications. Accurate ETA enables commuters to make informed travel decisions, supports fleet management and logistics operations, and reduces congestion by distributing demand across the road network. With the ubiquity of GPS-enabled devices and connected vehicles, navigation platforms such as Google Maps and Waze [1]–[3] have transformed how travelers plan and adapt their journeys. Despite this progress, ETA estimation remains challenging due to the dynamic and stochastic nature of urban traffic (see Fig. 1).

Traditional approaches rely on shortest-path algorithms such as Dijkstra's algorithm [4], which efficiently compute routes under static conditions but fail to capture evolving congestion or vehicle interactions. As a result, travel times obtained from such methods can diverge substantially from reality when conditions change during a trip.

With the growth of urban mobility datasets, machine-learning models have been applied to ETA prediction. Tree-based methods such as XGBoost [5] leverage handcrafted features and perform well on aggregated trip records, including NYC Taxi and Porto Taxi [6], [7]. Deep learning approaches improve accuracy by modeling temporal patterns and spatio-temporal context along a route. For example, DeepTTE learns ETA from raw GPS traces [8], TADNM incorporates transportation-mode awareness [9], MetaTTE applies meta-learning for cross-city generalization [10], and STAD corrects routing-engine outputs using spatio-temporal adjustments [11].

In parallel, graph-based spatio-temporal learning has advanced traffic prediction by representing road networks as graphs and capturing dynamics through diffusion or convolution operators (e.g., DCRNN, ST-GCN, Graph WaveNet) [12]–[14]. However, existing benchmarks such as NYC, Porto, Chengdu/DiDi, and Geolife [6], [7], [15], [16] lack explicit representation of pre-planned routes. Models must therefore infer likely paths between origin and destination, introducing ambiguity and reducing accuracy.

To address this limitation, we represent traffic as a dynamic, multi-relational graph in which static junction-to-junction road edges are complemented by time-varying vehicle and interaction edges (see Fig. 2). This formulation builds on our earlier research in traffic simulation and route-aware modeling [17], [18], which laid the methodological groundwork for graph-based analysis of traffic dynamics. In the present work, we advance this foundation by explicitly encoding each vehicle's remaining planned route (route intent) and by learning temporal context from stable road edges across recent history, while leveraging the complete dynamic graph at the final snapshot to predict per-vehicle ETA.

Results. On a four-week urban simulation, a simple average-time baseline attains 260.6 s MAE; our route-aware non-temporal variant achieves 58.4 s MAE, and the temporal route-aware model reaches 46.2 s MAE. These gains highlight the value of making route intent explicit and of learning temporal signals from stable road infrastructure rather than noisy, transient vehicle interactions.

Paper outline. We introduce the dynamic graph representation

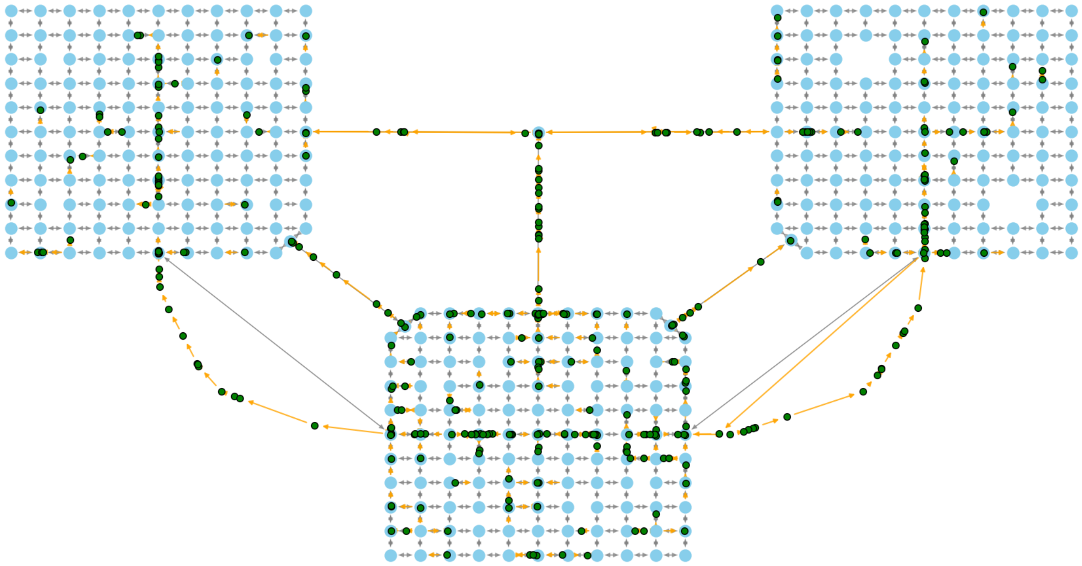


Fig. 1. Afternoon rush-hour snapshot spanning the network. Static junction nodes (light blue) and static road edges (gray) form the backbone; dynamic vehicle nodes (green) and traversal edges (orange arrows) reveal congestion corridors and high-flow funnels. Zones: A/B urban grids with dense signals and short links; C suburban ring with longer arterials and smoother flow; H connector hub that funnels inter-zone traffic and creates gateway bottlenecks.

and problem setup, summarize dataset characteristics and splits, detail the model architecture, and then present experiments and ablations followed by discussion and conclusions. To ground the setting early, Fig. 1 provides a system-level rush-hour snapshot.

Contributions. We introduce a dynamic, multi-relational traffic graph that unifies static junction nodes and road edges with dynamic vehicle nodes and interaction edges, and we provide explicit remaining-route annotations per vehicle to expose route intent. Building on this representation, we design a route-aware spatio-temporal GNN that learns temporal context from stable road edges over the first $H-1$ snapshots and fuses it with full-graph features at prediction time using a sparse Mixture-of-Experts head for specialization.

II. Related Work

A. Classical pathfinding and routing

On static graphs $G = (V, E)$ with non-negative edge costs, shortest paths are classically computed via Dijkstra's algorithm [4]. Alternatives such as Bellman–Ford [19] and bidirectional search [20] trade efficiency for generality. More advanced techniques such as A*/ALT landmarks [21], Contraction Hierarchies [22], and Customizable Route Planning [23] scale to web applications but rely on static edge-time models. As a result, they fail to capture evolving congestion or vehicle interactions. Our approach differs fundamentally by learning ETA directly from dynamic traffic graphs rather than static cost assumptions.

B. Learning-based ETA and traffic forecasting

The proliferation of urban mobility datasets motivated machine learning methods for ETA. Gradient boosting models such as XGBoost [5] leverage handcrafted features (trip distance, time-of-day) and perform well on aggregated taxi records (e.g., NYC, Porto [6], [7]), but they cannot capture fine-grained spatio-temporal interactions.

Trajectory- and trip-based neural models address this gap. DeepTTE [8] learns ETA from raw GPS traces and reports results on city-scale taxi/ride-hailing datasets (e.g., NYC TLC [6], Porto Taxi [7], DiDi 2016 [15]) with per-trip horizons typically within tens of minutes; MetaTTE [10] and related methods exploit cross-domain adaptation; STAD [11] adjusts routing outputs with spatio-temporal corrections. Production systems such as STANN [24] evaluate on large ride-hailing ETA datasets with short-to-medium trip durations (e.g., sub-30-minute trips). Complementary to these, the Smart Simulative Route framework [18] anticipates future congestion via heuristic simulation to improve ETA, but it remains simulation-driven and does not leverage graph-structured, data-driven temporal learning. Our work instead integrates explicit route intent and dynamic graph learning in a unified model.

C. Spatio-temporal graph neural networks

Graph neural networks advanced traffic forecasting by modeling sensor networks as spatio-temporal graphs. DCRNN [12] combined diffusion graph convolutions with recurrent units and became a common backbone for traffic speed/flow prediction on sensor benchmarks (METR-LA, PEMS-BAY; 5–60 min horizons in speed units). Notably, DCRNN-style encoders have also been adapted for ETA prediction in production settings, e.g., within DuETA [25]. In contrast to sensor-grid forecasting, we target per-vehicle ETA on dynamic multi-relational graphs with explicit route intent, while ConSTGAT [26] targets sensor-grid speed forecasting on METR-LA/PEMS-BAY at fixed horizons (not per-trip ETA).

Importantly, reported MAEs across these works are not directly comparable without dataset and time-horizon context (city, data source, and maximum trip duration). We therefore reference methods together with their evaluation scope rather than quoting unscoped headline numbers.

D. State-of-the-art production models

Most recently, DuETA [25] introduced duration-aware ETA modeling at Baidu Maps, categorizing trips into short (0–3 km), medium (3–10 km), and long (>10 km). DuETA achieved MAEs of 27s, 46s, and 98s respectively across these categories, significantly improving upon prior baselines. Unlike DuETA, which relies on trip segmentation, our method unifies all trip types under a single dynamic spatio-temporal framework. By incorporating route-left path features directly into the graph, our model can generalize across short and long trips without predefined bins. Moreover, we learn temporal context from stable road edges across history and fuse it at the prediction step, complementing DuETA's duration-aware design with explicit route intent and road-based temporal aggregation.

E. Summary

Prior work highlights the value of sequential modeling (DeepTTE), spatio-temporal attention (STANN, ConSTGAT), graph diffusion (DCRNN), and duration-aware embeddings (DuETA). However, none fully integrate explicit pre-planned routes with vehicle–junction dynamics within a unified dynamic graph. Our contribution is to combine explicit route intent, road-based temporal aggregation, and a sparse MoE head in a single spatio-temporal GNN for ETA.

III. Methodology

A. Dynamic Graph Representation

We represent the transportation system at snapshot time t as a directed multi-relational graph

$$G_t = \left(V_t, E_t, \{A_t^{(\rho)}, W_t^{(\rho)}\}_{\rho \in \mathcal{R}}\right), \tag{1}$$

where:

- $V_t = V^j \cup V_t^v$ is the set of nodes at time t, consisting of
  - V^j: static junction nodes, representing intersections in the road network,
  - V_t^v: dynamic vehicle nodes, representing vehicles present at time t.
- E_t is the set of directed edges at time t, partitioned by relation type ρ.
- $\mathcal{R}$ is the set of relation types (road, traversal, interaction).
- $A_t^{(\rho)} \in \{0,1\}^{|V_t| \times |V_t|}$ is the binary adjacency matrix for relation ρ at time t.
- $W_t^{(\rho)} \in \mathbb{R}_{\geq 0}^{|V_t| \times |V_t|}$ is the optional weighted adjacency for relation ρ (e.g., distance-based interaction weights).

We define three relation types:

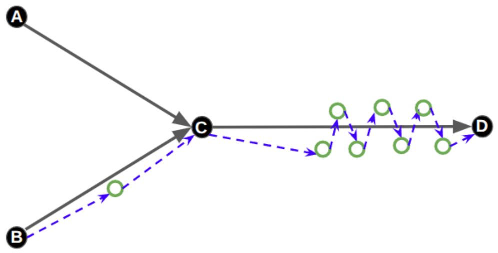
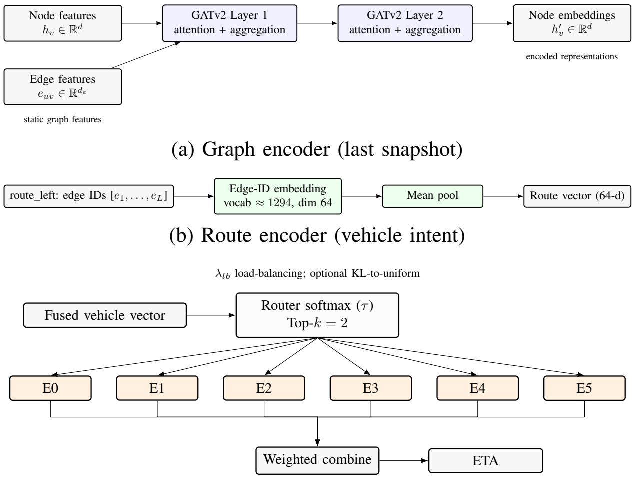


Fig. 2. Dynamic, multi-relational traffic graph: static junction nodes (labeled A–D) and road edges (solid gray) with dynamic vehicle nodes (hollow green circles) and interaction edges (blue dashed).



(a) Graph encoder (last snapshot)

route_left: edge IDs $[e_1, \ldots, e_L]$
Edge-ID embedding
vocab ≈ 1294, dim 64
Mean pool
Route vector (64-d)

(b) Route encoder (vehicle intent)

(c) Router and Mixture-of-Experts

Fig. 3. Key components: (a) two-layer GATv2 with edge features; (b) route intent via edge-ID embeddings and mean pooling; (c) sparse Top-$k = 2$ routing over six experts with load balancing.

1) **Road edges ($\rho =$ road):** For adjacent junctions (u, v) with legal direction $u \to v$, we add $(u, v) \in E_t^{(\text{road})}$, encoding static road topology.
2) **Traversal edges ($\rho =$ trav):** If vehicle v_i^t occupies segment (a, b), we add (a, v_i^t) and (v_i^t, b), linking the vehicle to its upstream and downstream junctions.
3) **Interaction edges ($\rho =$ inter):** For vehicles v_i^t, v_j^t on the same segment with spacing $d_{ij}(t) \leq \varepsilon$ and aligned headings, we add (v_i^t, v_j^t) weighted by $\omega(d_{ij}(t)) = e^{-d_{ij}(t)/\lambda}$.

a) Dynamic edge construction.: In implementation, vehicles on each road segment are ordered by their normalized position along the edge. We then create a short chain per segment: a junction→first-vehicle edge (type 1), vehicle→vehicle links along the ordered list (type 3), and last-vehicle→junction (type 2). These dynamic edges sit on top of the static road graph (type 0).

Each node and edge carries feature vectors. Nodes use 28 features per snapshot (junction/vehicle type, kinematics, temporal encodings, zone one-hot, coordinates, current-edge attributes, demand/occupancy surrogates, and junction type). Edges use 7 features (average speed, lanes one-hot, length, demand, occupancy). Average speed, demand, and occupancy are updated per snapshot from data; vehicle features at indices for current-edge demand/occupancy are filled from the corresponding edge attributes.

b) Route intent.: In addition to graph-based relations, each vehicle node includes a feature called `vehicle_route_left`, which encodes the sequence of upcoming edges along its pre-computed source–destination path. Construction: for each trip we compute a static shortest path over E^{road} and record its ordered edge-ID list. At time t, we locate the vehicle's current edge within this list and take the suffix from the next edge to the destination. Indices are the canonical 0-based edge IDs shared across the dataset (vocabulary size 1,294). This sequence is summarized by a Route Encoder (edge-ID embedding + mean pooling) and used in fusion.

B. Temporal Windowing

ETA prediction requires reasoning over temporal dynamics. We construct a window of H consecutive snapshots:

$$\mathcal{G}_{t-H+1:t} = \{G_\tau\}_{\tau=t-H+1}^{t}, \tag{2}$$

where H is configurable (default H=30 in our experiments, with 30-second spacing). The prediction target is the ETA of vehicles present in the final snapshot G_t.

C. Problem Statement

Given a temporal window of dynamic graphs $\mathcal{G}_{t-H+1:t} = \{G_\tau\}_{\tau=t-H+1}^{t}$, where each snapshot G_τ contains static junction nodes V^j, dynamic vehicle nodes V^v_τ, static road edges, and time-varying interaction edges, the goal is to predict per-vehicle ETAs at the final snapshot t^*=t. Let $\mathcal{V}_t = V^v_t$ denote vehicles present at time t. We learn a function f_θ that maps

$$f_\theta\big(\mathcal{G}_{t-H+1:t},\ \texttt{route\_left}\big) \ \to\ \hat{\mathbf{y}}_t \in \mathbb{R}^{|\mathcal{V}_t|}, \tag{3}$$

where $\hat{\mathbf{y}}_t[i]$ is the ETA for vehicle $i \in \mathcal{V}_t$. Training minimizes mean absolute error (MAE) over vehicles present at t (additional terms are described in Section III-F):

$$\mathcal{L}_{\text{MAE}} = \frac{1}{|\mathcal{V}_t|} \sum_{i \in \mathcal{V}_t} \big|\hat{y}_t[i] - y_t[i]\big|. \tag{4}$$

D. Model Architecture

Our DSTRA-GNN (Dynamic Spatio-Temporal Route-Aware GNN) integrates spatial, temporal, and route information. See the complete architecture in Fig. 4. The main components are:

- **Graph Encoder:** Each snapshot G_t is processed by a multi-layer GATv2-based encoder with residual connections and edge features, producing embeddings for junction and vehicle nodes (see the encoder panel in Fig. 3).
- **Temporal Aggregator:** For each pre-final snapshot $\tau \in \{t-H+1, \ldots, t-1\}$ we take the static road-edge features (edge_type= 0) and build edge-wise sequences across time. A temporal module (Transformer with sinusoidal positional encodings; GRU optional) processes these sequences per edge; the temporally aggregated edge features are then projected and pooled over all static edges to a graph-level context vector (detail in Fig. 5). This context is broadcast to all vehicles at t^* for prediction. For *all* temporal variants, temporal context is computed from static road edges only for $t-H+1 \ldots t-1$, while the final snapshot G_t is encoded with the appropriate spatial topology (static or full dynamic) per variant.
- **Vehicle Selection:** From the final snapshot G_t, we retain only vehicle node embeddings, as these are the prediction targets for ETA.
- **Route Encoder:** Each vehicle's remaining route (`vehicle_route_left`) is embedded via an edge-ID lookup (vocabulary $|E^{\text{road}}|$=1,294, embedding dim 64) followed by mean pooling over the variable-length sequence (see the route panel in Fig. 3). The pooled route vector is concatenated with the vehicle embedding and temporal context when route awareness is enabled.
- **Fusion and MoE Head:** Vehicle embeddings, enriched with route and temporal context, are fused through a feed-forward network and routed to a sparse Top-k Mixture-of-Experts (MoE) head (router panel in Fig. 3), where specialized experts capture heterogeneous traffic regimes. The output is the ETA prediction for each vehicle.

E. Model Configuration

Unless stated otherwise, experiments use the following settings:

- **Node/edge features**: 28 node features; 7 edge features. Temporal window H=30 consecutive snapshots.
- **Graph encoder**: 2-layer GATv2 with edge features, hidden size 128, 4 heads per layer, GELU activations, LayerNorm residuals, dropout 0.1.
- **Temporal module**: static-road edge sequences over the first $H-1$ snapshots. Default is a Transformer Encoder with n_{head}=4, n_{layers}=2, GELU, FFN size $2d$, dropout 0.1, and sinusoidal positional encodings; an optional single-layer GRU (hidden 128) is also supported. Aggregated edge context is projected, mean-pooled over static edges to a graph vector, then broadcast to vehicles.
- **Route encoder**: edge-ID embedding of size 64 with mean pooling over `vehicle_route_left` (vocabulary size 1294).
- **Fusion**: MLP (256 hidden with GELU, LayerNorm, dropout 0.1) to a 192-dim fused representation.
- **MoE head**: Top-k=2 sparse routing over 6 experts (dropout 0.1). Predictions are made for vehicles in the last snapshot only.
- **Target normalization**: we train in the $y_{log,z}$ space:

$$y_{log} = \log\big(1 + y_{sec}\big) \tag{5}$$

$$y_{log,z} = \frac{y_{log} - \mu}{\sigma} \tag{6}$$

where y_{sec} is the raw ETA in seconds, and (μ, σ) are computed on the training set; metrics are reported after inverting to seconds.
- **Training**: AdamW (lr 10^{-3}, weight decay 10^{-2}), linear warmup then cosine annealing ($T_{\max}$=200), batch size 2,

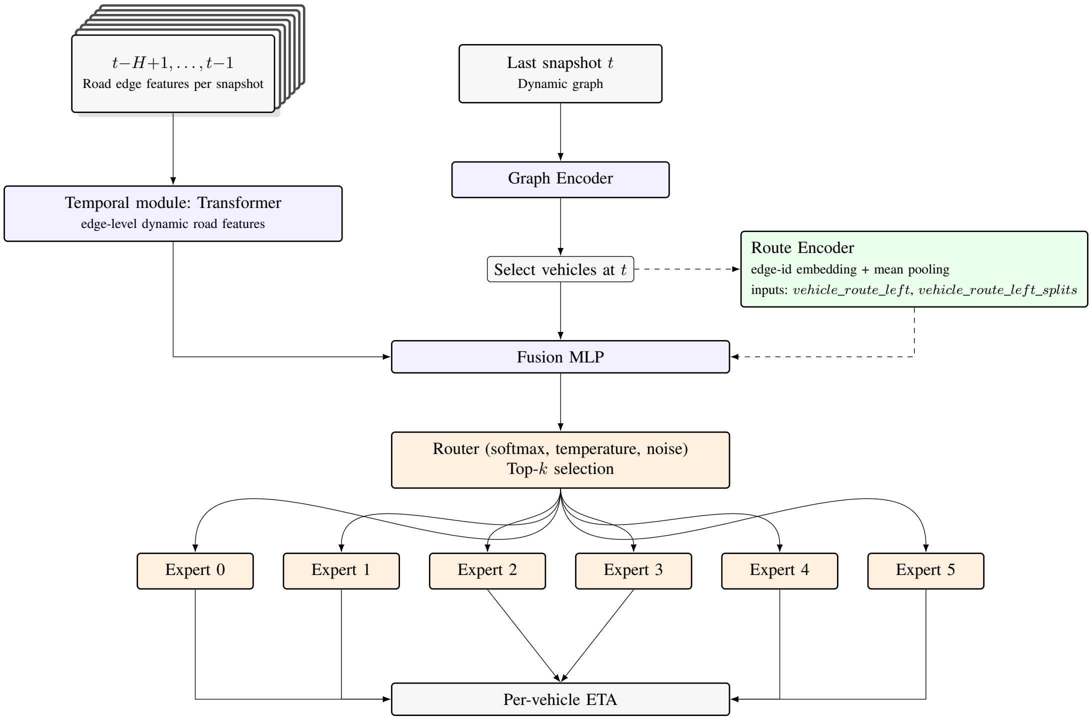

$t-H+1,\ldots,t-1$
Road edge features per snapshot
Last snapshot t
Dynamic graph
Graph Encoder
Temporal module: Transformer
edge-level dynamic road features
Select vehicles at t
Route Encoder
edge-id embedding + mean pooling
inputs: $vehicle\_route\_left$, $vehicle\_route\_left\_splits$
Fusion MLP
Router (softmax, temperature, noise)
Top-k selection
Expert 0
Expert 1
Expert 2
Expert 3
Expert 4
Expert 5
Per-vehicle ETA

Fig. 4. DSTRA-GNN architecture. A window of T=30 dynamic graph snapshots (shown as an overlapped deck) is processed by a *shared* Graph Encoder; prediction is made at the last snapshot t^*. In the Full variant, a Route Encoder summarizes each vehicle's remaining path before Fusion and a Top-k MoE head produces per-vehicle ETA.

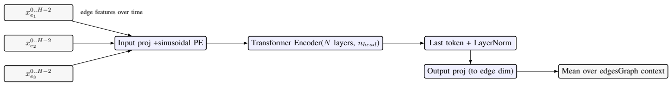

$x_{e_1}^{0..H-2}$
edge features over time
$x_{e_2}^{0..H-2}$
$x_{e_3}^{0..H-2}$
Input proj +sinusoidal PE
Transformer Encoder(N layers, n_{head})
Last token + LayerNorm
Output proj (to edge dim)
Mean over edgesGraph context

Fig. 5. Temporal Transformer detail: per-edge sequences across $H-1$ snapshots are projected, positionally encoded, encoded by a Transformer stack, reduced via last-token + norm, projected back to edge dimension, then pooled over edges.

200 epochs. Primary loss in target space with metrics reported after inversion to seconds.

- **Router regularization**: load-balancing weight λ_{lb}=0.05 plus optional KL-to-uniform router regularization with weight 0.002.

F. Loss Functions and Training Protocol

We train the model with supervised regression on ETA targets using mean absolute error (MAE) in seconds:

$$\mathcal{L}_{\text{MAE}} = \frac{1}{N}\sum_{i=1}^{N} \left|\hat{y}_i - y_i\right|, \tag{7}$$

where $\hat{y}_i$ and y_i are the predicted and true ETAs. Additional evaluation metrics include RMSE and mean absolute percentage error (MAPE). The optimization objective includes MoE load balancing and optional router regularization: For completeness, we report the following definitions used in evaluation:

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2} \tag{8}$$

$$\text{MAPE}(\%) = 100 \cdot \frac{1}{N}\sum_{i=1}^{N}\left|\frac{\hat{y}_i - y_i}{y_i}\right| \tag{9}$$

All metrics are computed in seconds (after inverting any target-space transforms) and averaged over vehicles in the test split.

$$\mathcal{L} = \mathcal{L}_{\text{reg}} + \lambda_{\text{lb}}\,\mathcal{L}_{\text{lb}} \ \ \text{(and optionally } \lambda_{\text{kl}}\,\text{KL}(p\,\|\,\text{Uniform})\text{)}, \tag{10}$$

where $\mathcal{L}_{\text{lb}}$ encourages uniform expert usage via importance/load terms from the router, and p are routing probabilities. Training occurs in a chosen target space (e.g., y_{log} or $y_{\text{log\_z}}$), while metrics are reported after consistent inversion to seconds.

a) Complexity and inference.: Per-snapshot spatial encoding is $\mathcal{O}(|E_t|)$; temporal aggregation is $\mathcal{O}(|E_{\text{road}}| \cdot (H-1))$ over static road edges only. Inference processes a window and emits per-vehicle ETAs at t^* with negligible overhead beyond a single forward pass of the encoder plus temporal aggregator.

The dataset spans four simulated weeks of traffic. We partition this chronologically into two weeks for training, one week for validation, and one week for testing. This split ensures that models are evaluated on non-overlapping time intervals that include different traffic patterns (rush hours, weekends, and long trips).

IV. Experimental Evaluation

A. Dataset

We evaluate on a simulation-derived dynamic traffic dataset built with the Eclipse SUMO traffic simulator [27], exported as time-ordered snapshots at 30-second intervals. Each snapshot $G_t = (V^j, V_t^v, E^{\text{road}}, E_t^{\text{dyn}})$ contains:

- **Static structure**: junction nodes V^j (types: traffic_light/priority) and directed road edges E^{road} (attributes: length, speed limit, num_lanes, zone).
- **Dynamic state**: vehicle nodes V_t^v with kinematics (speed, acceleration), positions, zone and current_edge; edge-time signals (avg_speed, vehicles_on_road_count, density).
- **Route-related features**: per-vehicle remaining route sequence (`vehicle_route_left`). For each trip, we compute the source–destination path using SUMO's Dijkstra implementation [4], [27] on the static road graph, with edge weights given by a travel-time cost derived from edge length and average speed. We then store the ordered list of edge IDs. At snapshot t, `vehicle_route_left` is the suffix of this list from the vehicle's current edge (exclusive) to destination; indices are 0-based edge IDs consistent with $|E^{\text{road}}|$=1,294. This sequence feeds the route encoder (edge-ID embedding + mean pooling) used by route-aware variants. We also include per-edge demand/occupancy proxies: `vehicles_on_road_count` and its log, and the number of routes that traverse each edge (`edge_route_count`, log).

Fig. 1 shows an afternoon rush-hour snapshot illustrating realistic congestion buildup across three zones: (i) **A/B urban grids** with dense signalized junctions and short links, driving stop-and-go dynamics and rich vehicle interactions; (ii) **C suburban ring** with longer arterials and fewer signals, yielding smoother flows; and (iii) **H hub/connector** that funnels inter-zone traffic, creating recurrent bottlenecks at gateways. The zone label is available at both node and edge level and conditions many dynamic features (e.g., average speed, occupancy), which the model can leverage.

Scope and scale:

- Duration and coverage: ∼4 weeks; snapshots every 30 s; total snapshots: 80,730.
- Topology: junctions: $|V^j|$=362; static road edges: $|E^{\text{road}}|$=1,294.
- Traffic: unique vehicles: 208,763; trips: 22,767,090; avg active vehicles per snapshot: 282.02.

Chronological splits are used throughout (no overlap): train, validation, and test. Unless otherwise stated, we use a temporal window of H=30 snapshots (15 minutes) and a stride of 10 between windows.

B. Dataset Statistics

Key feature statistics:

- **Vehicles**: mean speed 12.11 m/s (std 10.21), acceleration mean -0.15 m/s^2 (std 1.12); route length mean 13.07 km (median 13.98 km); remaining route (`route_length_left`) mean 7.60 km (median 7.90 km); vehicle types: passenger/truck/bus.
- **Junctions**: 362 nodes; types split between traffic_light and priority; zones A/B/C/H.
- **Edges**: speed limit median 13.89 m/s; length median 485.6 m; lanes in {1,2,3}; avg vehicles on road count is sparse (median 0), heavy-tailed; `edge_route_count` is highly skewed (max 785), use log scaling.
- **Labels/ETA**: ETA mean 504 s (std 416), spans 0–∼5000 s; total travel time mean 1011 s.

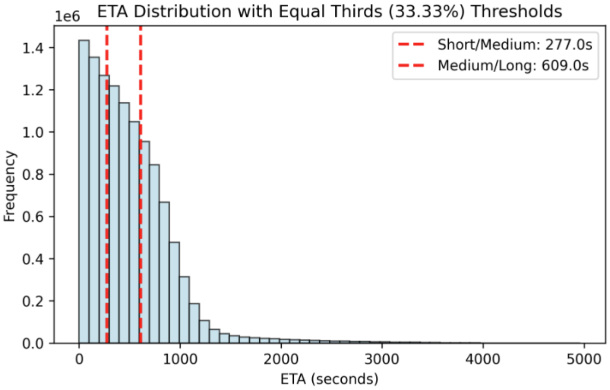


Fig. 6. ETA distribution (seconds) with equal-thirds thresholds. Vertical dashed lines indicate short/medium (≈277 s) and medium/long (≈609 s) boundaries.

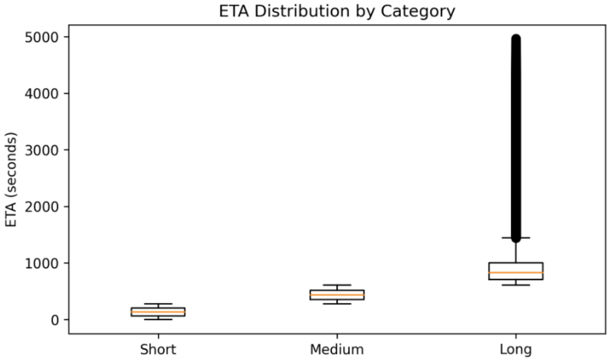


Fig. 7. ETA distribution by route-length category (short/medium/long). Boxes summarize medians and IQRs; whiskers capture spread and outliers.

ETA statistics and bin thresholds are shown in Figs. 6–8. We use equal-thirds thresholds (Fig. 6) at ≈277 s and ≈609 s to define short/medium/long duration bins; validation counts per bin are listed in Table III.

C. Experimental Setup

Models are trained using AdamW with learning rate 10^{-3} and weight decay 10^{-2}, batch size 2, for 200 epochs. The schedule uses a short linear warmup followed by cosine annealing ($T_{\max}$=200). Targets are learned in a normalized space (default $y_{\text{log\_z}}$), while metrics are computed after consistent inversion to seconds. The router uses load-balancing (weight 0.05) and KL-to-uniform regularization (weight 0.002). Unless stated otherwise, reported numbers are the mean over seeds 42, 43, and 44; model selection uses the lowest validation MAE per seed.

D. Baselines and Ablations

We compare six code-defined variants (shared encoder and training protocol). Table I summarizes the variants and which

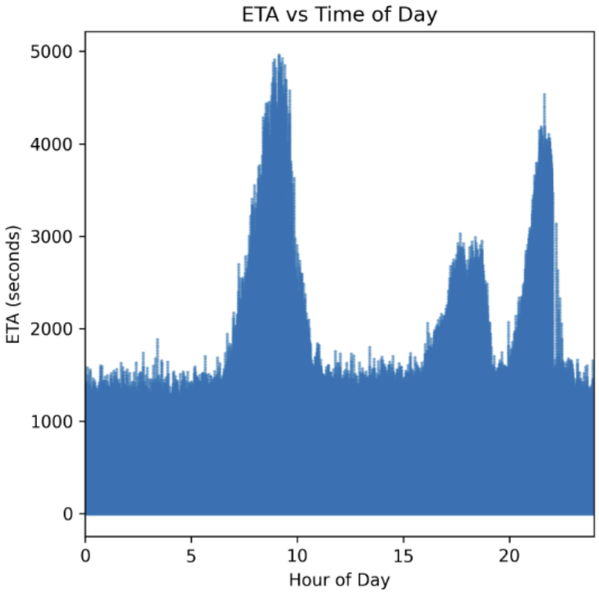


Fig. 8. ETA versus hour-of-day showing strong diurnal effects (morning and evening peaks), with occasional evening events/spikes.

TABLE I
ABLATION VARIANTS AND ENABLED COMPONENTS. TEMPORAL USES A TRANSFORMER WITH STATIC-ROAD CONTEXT FOR THE FIRST $H-1$ SNAPSHOTS.

Variant	**Dyn. Edges**	**Route Feat.**	**Temporal**
base_graph	—	—	—
dynamic_graph	✓	—	—
route_aware_graph	✓	✓	—
temporal_base	—	—	✓
temporal_dynamic	✓	—	✓
temporal_route_aware	✓	✓	✓

components are enabled:

- `base_graph`: static road graph only; no dynamic edges; no temporal; no route features.
- `dynamic_graph`: adds dynamic edges (junction→vehicle, vehicle→vehicle, vehicle→junction); no temporal; no route features.
- `route_aware_graph`: adds route features/encoder and dynamic edges; no temporal.
- `temporal_base`: temporal context from static-road edges only for the first $H-1$ snapshots; no route features; spatial encoder uses only static edges.
- `temporal_dynamic`: temporal context from the first $H-1$ snapshots is aggregated over static-road edges; spatial encoder uses dynamic edges; no route features.
- `temporal_route_aware`: full model with dynamic edges, route features/encoder, and temporal context aggregated from static-road edges for the first $H-1$ snapshots.

E. Evaluation Metrics

We report Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) in seconds, and Mean Absolute Percentage Error (MAPE). We also provide stratified MAE by route-length and duration bins.

TABLE II
OVERALL AND PER-BIN VALIDATION MAE (SECONDS) AVERAGED OVER SEEDS 42–44 (BEST EPOCH PER SEED). IMPROVEMENT IS RELATIVE TO THE AVG BASELINE.

Variant	**MAE**	**Short**	**Medium**	**Long**	**Improve vs AVG**
avg	260.63	282.20	83.56	430.67	0.0%
base_graph	86.46	39.20	81.79	140.27	66.8%
dynamic_graph	104.51	42.65	95.99	177.54	59.9%
route_aware_graph	58.42	24.19	54.84	97.61	77.6%
temporal_base	92.62	61.35	83.54	135.00	64.5%
temporal_dynamic	78.79	32.04	67.58	142.14	69.8%
temporal_route_aware	46.16	24.05	38.31	76.90	82.3%

V. RESULTS AND ANALYSIS

Validation results. Tables II and IV report the performance of all ablation variants. The naive average-time baseline (`avg`) yields 260.6 s MAE and 325.1 s RMSE, with a MAPE of 127.8%, confirming the task's difficulty.

The `base_graph` model, which uses only static road topology and basic vehicle kinematics, reduces MAE to 86.5 s (66.8% improvement). Short trips are predicted with 39.2 s MAE, while medium and long trips remain harder (81.8 s and 140.3 s).

Adding dynamic interaction edges in `dynamic_graph` achieves 104.5 s MAE (59.9%), weaker than the static-only model. Errors for medium/long trips (95.9 s and 177.5 s) indicate that dynamic edges without route features add noise rather than signal.

The `route_aware_graph` variant delivers strong gains: 58.4 s MAE overall, with 24.2 s (short), 54.8 s (medium), and 97.6 s (long). Explicitly encoding planned routes and edge demand/occupancy lets the model anticipate congestion along a vehicle's actual path.

Temporal variants further refine performance. `temporal_base` achieves 92.6 s MAE (64.5%), mainly improving medium trips but struggling on short trips (61.4 s). `temporal_dymanic` improves to 78.8 s MAE (69.8%), showing that temporal memory of interaction edges helps even without route features.

The best performance is `temporal_route_aware`, which combines temporal aggregation with explicit route features: 46.2 s MAE and 107.2 s RMSE (82.3% improvement), with balanced per-bin accuracy (24.1 s short, 38.3 s medium, 76.9 s long) and the lowest MAPE (17.9%).

Equally important is the underlying dataset representation. Unlike prior work that models only road segments or aggregated trajectories, our dataset is structured as a dynamic graph where both junctions and vehicles are nodes. Static road edges encode the persistent infrastructure, while dynamic edges evolve at every snapshot to reflect vehicle–junction and vehicle–vehicle relationships. This design allows the model to jointly learn from the stable backbone of the road network and the transient dynamics of traffic flow, enabling the strong gains observed in route-aware and temporal variants.

TABLE III
Duration binning used throughout (validation split). Thresholds from Fig. 6.

Bin	ETA range (s)	τ (s)	Count
Short	[0, 277]	30	33,227
Medium	(277, 609]	60	34,996
Long	$>$ 609	120	32,228

TABLE IV
Overall validation MAE and RMSE (seconds) and MAPE (%). Best epoch per variant on validation; averaged over seeds 42–44 where available. MAPE values are estimated where missing in logs.

Variant	MAE (s)	RMSE (s)	MAPE (%)
avg	260.63	325.13	127.75
base_graph	86.46	145.08	33.70
dynamic_graph	104.51	157.65	40.73
route_aware_graph	58.42	106.99	22.77
temporal_base	92.62	155.48	26.51
temporal_dynamic	78.79	127.88	30.70
temporal_route_aware	46.16	107.15	17.99

VI. Discussion

The results highlight three clear trends. First, static road graphs already provide a strong inductive bias, with the `base_graph` achieving a two-thirds reduction in error relative to the average baseline. This demonstrates the value of graph-structured representations in capturing spatial constraints.

Second, route intent emerges as the most decisive feature. The `route_aware_graph` achieves 58.4 s MAE overall, more than 40% lower than either the `base_graph` or `dynamic_graph`. Notably, short trips drop from 39–43 s error to just 24 s, indicating that route features resolve much of the ambiguity about vehicle destination and path choice.

Third, temporal context provides an additional layer of refinement. The `temporal_route_aware` variant achieves 46.2 s MAE and 17.9% MAPE, corresponding to an 82.3% improvement over the average baseline and outperforming all other variants across every trip-length category. The temporal aggregation of $T-1$ snapshots enables the network to recognize evolving congestion patterns, improving robustness for medium and long trips where static and route-only models tend to accumulate error.

These gains are substantially higher than those reported by state-of-the-art methods on real-world datasets. For example, DuETA reports MAE reductions of 40–50% relative to average baselines across Beijing, Shanghai, and Tianjin [25], while earlier models such as DCRNN, DeepTTE, and ConSTGAT report similar ranges [8], [12], [26]. In contrast, our model delivers over 80% improvement, underscoring the benefit of combining explicit route intent with dynamic spatio-temporal graph modeling.

Together, these findings validate the dataset design, which provides per-vehicle route annotations, dynamic interaction edges, and temporally aligned graph snapshots. Each of these components is directly reflected in the ablation gains. At the same time, long trips remain the most challenging case (76.9 s MAE even in the best model), indicating that compounding local errors across extended horizons remains an open problem.

A further contribution of this work lies in the dataset formulation itself. By explicitly representing both junctions and vehicles as nodes, and layering dynamic edges on top of static road topology, the traffic network is captured in a richer and more flexible way than traditional road-segment–based graphs. This representation provides the inductive bias needed to integrate infrastructure constraints with real-time vehicle interactions, and the ablation results demonstrate that it is precisely this combination—static roads, dynamic edges, and route-aware features—that yields the largest performance gains.

The scale and temporal coverage of the dataset further reinforce these results. With over 80,000 graph snapshots, 200,000+ vehicles, and approximately 927,000 distinct routes across a continuous four-week horizon, the evaluation provides dense temporal coverage and rich route diversity. This extended duration ensures that models are exposed to recurring traffic cycles such as rush hours, weekday–weekend variations, and long-term congestion patterns, while the large number of routes captures heterogeneous travel intents and path choices. Compared to widely used benchmarks such as METR-LA (342 loop sensors, aggregated speeds over a few months) or trajectory datasets like Q-Traffic and DiDi, which provide trip-level GPS traces without explicit route intent or graph structure, our dataset offers far richer temporal, spatial, and behavioral signals. Importantly, this level of detail reflects information readily available in modern navigation systems through GPS traces, map matching, and planned-route logging, making the evaluation both realistic and robust while reducing the risk of overfitting to short-term artifacts.

VII. Conclusion

This work introduced a dynamic spatio-temporal graph neural network for ETA prediction that unifies static infrastructure, vehicle-level dynamics, and explicit route intent within a Mixture-of-Experts framework. A central innovation of our approach is the formulation of the traffic environment as a dynamic graph in which both junctions and vehicles are nodes, connected by persistent road edges and evolving interaction edges. This representation enables the model to leverage the stability of the road network while simultaneously adapting to transient traffic patterns.

By systematically evaluating ablation variants, we demonstrated that each component contributes to predictive accuracy: graph structure captures spatial constraints, route features provide the largest single boost, and temporal aggregation refines predictions by modeling evolving congestion. The resulting temporal route-aware model reduced validation MAE from 260.6 s in the average baseline to 46.2 s, an 82.3% improvement, with balanced accuracy across short, medium, and long trips. Compared with improvements typically reported in the range of 40–50% by state-of-the-art baselines, these results highlight the benefit of representing the traffic system as a dynamic vehicle–junction graph.

Future work will focus on scaling to real-world datasets, addressing error accumulation in long trips, and ensuring robustness under unexpected disruptions, thereby advancing the practical deployment of intent-aware dynamic graph models in intelligent transportation systems.

Acknowledgments

The authors would like to thank Ruppin Academic center, Israel, and the Israeli Ministry of Innovation, Science and Technology (Proposal no. 0007846), for the continuing support in this research, implementation, and presentation. This study was funded by grant number 34836.

References

[1] A. Derrow-Pinion, J. She, D. Wong, O. Lange, T. Hester, L. Perez, M. Nunkesser, S. Lee, X. Guo, B. Wiltshire, P. W. Battaglia, V. Gupta, A. Li, Z. Xu, A. Sanchez-Gonzalez, Y. Li, and P. Veličković, "Eta prediction with graph neural networks in google maps," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, 2021, pp. 4034–4042. [Online]. Available: https://arxiv.org/abs/2108.11482

[2] N. Hoseinzadeh, Y. Liu, L. D. Han, C. Brakewood, and A. Mohammadnazar, "Quality of location-based crowdsourced speed data on surface streets: A case study of waze and bluetooth speed data in sevierville, tn," *Computers, Environment and Urban Systems*, vol. 83, p. 101518, 2020.

[3] M. Amin-Naseri, P. Chakraborty, A. Sharma, S. B. Gilbert, and M. Hong, "Evaluating the reliability, coverage, and added value of crowdsourced traffic incident reports from waze," *Transportation Research Record*, vol. 2672, no. 43, pp. 34–43, 2018.

[4] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, no. 1, pp. 269–271, 1959.

[5] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2016, pp. 785–794.

[6] NYC TLC, "New york city taxi and limousine commission trip record data," https://www.nyc.gov/site/tlc/about/tlc-trip-record-data.page, 2013–, accessed: Aug. 26, 2025.

[7] L. Moreira-Matias, J. Gama, M. Ferreira, J. Mendes-Moreira, and L. Damas, "Predicting taxi–passenger demand using streaming data," in *2013 IEEE 16th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2013, pp. 140–145.

[8] Y. Wang, F. Yin, Y. Chen, J. Zhao, C. Xu, and J. Ye, "Deeptte: End-to-end trajectory-based travel time estimation," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 861–870.

[9] W. Xu, Z. Lin, Y. Zhao, T. Zhang, and B. Yang, "Tadnm: A transportation-mode aware deep neural model for travel time estimation," *Applied Sciences*, vol. 10, no. 21, p. 7599, 2020.

[10] X. Wang, J. Li, and N. J. Yuan, "Metatte: A meta-learning framework for travel time estimation," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, 2022, pp. 4080–4088.

[11] S. Abbar, A. Anagnostopoulos, S. Bhagat, P. Cudre-Mauroux, and A. Kumar, "Stad: Spatio-temporal adjustment for improving travel-time estimation," in *Proc. Web Conf.*, 2020, pp. 2839–2845.

[12] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," in *Int. Conf. Learn. Represent. (ICLR)*, 2018. [Online]. Available: https://arxiv.org/abs/1707.01926

[13] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," in *Proc. Int. Joint Conf. Artif. Intell. (IJCAI)*, 2018, pp. 3634–3640.

[14] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, "Graph wavenet for deep spatial-temporal graph modeling," in *Proc. Int. Joint Conf. Artif. Intell. (IJCAI)*, 2019, pp. 1907–1913.

[15] DiDi Chuxing Research, "Di-tech challenge 2016," https://outreach.didichuxing.com/research/opendata/en/, 2016, accessed: Aug. 26, 2025.

[16] Y. Zheng, L. Zhang, Z. Ma, X. Xie, and W.-Y. Ma, "Mining interesting locations and travel sequences from gps trajectories," in *Proceedings of the 18th international conference on World wide web*. ACM, 2009, pp. 791–800.

[17] N. Voloch and N. Voloch-Bloch, "Finding the fastest navigation route by real-time future traffic estimations," *2021 IEEE International Conference on Microwaves, Communications, Antennas and Electronic Systems (COMCAS)*, 2021.

[18] N. Voloch, N. Penkar, and G. Tordjman, "Estimating accurate traffic time by smart simulative route predictions," in *Proceedings of the 24th IFIP Conference on e-Business, e-Services and e-Society (I3E 2025)*. Limassol, Cyprus: IFIP WG 6.11, 2025, session: Smart Public Services and Technologies – 1. LNCS (Springer), to appear.

[19] R. Bellman, "On a routing problem," *Quarterly of Applied Mathematics*, vol. 16, pp. 87–90, 1958.

[20] I. Pohl, "Bi-directional search," *Machine Intelligence*, vol. 6, pp. 127–140, 1971.

[21] A. V. Goldberg and C. Harrelson, "Computing the shortest path: A* search meets graph theory," in *Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. SIAM, 2005, pp. 156–165. [Online]. Available: https://dl.acm.org/doi/10.5555/1070432.1070459

[22] R. Geisberger, P. Sanders, D. Schultes, and C. Vetter, "Contraction hierarchies: Faster and simpler hierarchical routing in road networks," in *Proceedings of the 7th International Conference on Experimental Algorithms (WEA)*. Springer, 2008, pp. 319–333.

[23] D. Delling, A. V. Goldberg, T. Pajor, and R. F. Werneck, "Customizable route planning," in *Proceedings of the 10th International Symposium on Experimental Algorithms (SEA)*. Springer, 2011, pp. 376–387.

[24]

[25] J. Zhou, X. Jin, X. Huang, and H. Tong, "Dueta: Duration-aware travel time estimation with graph neural networks," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2023. [Online]. Available: https://arxiv.org/abs/2306.02833

[26] C. Song, Y. Lin, S. Guo, H. Wan, G. Yu, Y. Liang, and K. Xie, "Contextualized spatial-temporal graph attention network for traffic forecasting," in *AAAI Conference on Artificial Intelligence*, 2020, pp. 10 049–10 056. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/6503

[27] DLR and contributors, "SUMO - simulation of urban mobility," https://www.eclipse.org/sumo/, 2018, accessed: Aug. 26, 2025.