Dynamic Graph Neural Networks for Accurate ETA prediction

Guy Tordjman Nadav Voloch

I. Abstract

Accurate Estimated Time of Arrival (ETA) prediction is crucial for modern navigation systems and intelligent transportation applications. While existing approaches rely on static routing algorithms or machine learning models that infer likely paths between origin and destination, they fail to leverage the user's actual intended route and the dynamic nature of traffic interactions. This paper introduces a novel Dynamic Graph Neural Network (DGNN) architecture that explicitly incorporates user route intent and temporal memory to achieve superior ETA prediction accuracy.

Our approach represents the transportation network as a dynamic multi-relational graph where junctions and vehicles are nodes, connected by typed edges capturing road segments, vehicle traversals, traffic interactions, and route intentions. The model employs a mixture-of-experts architecture with 6 specialized experts and a memory layer that records historical snapshots to capture temporal dependencies. Unlike existing benchmarks that lack explicit route representation, our dataset unifies junction states, vehicle dynamics, and pre-planned route information, enabling the model to condition predictions on the actual path intended by travelers.

Experimental results on simulated traffic data demonstrate the effectiveness of our approach. The model achieves a validation MAE of [RESULTS_PLACEHOLDER] seconds, representing a [IMPROVEMENT_PLACEHOLDER]% improvement over baseline methods. Performance analysis across different trip duration bins shows particularly strong results for [DETAILED_RESULTS_PLACEHOLDER].
The mixture-of-experts architecture with route awareness proves essential for handling heterogeneous traffic regimes, with the router entropy stabilizing at [ENTROPY_PLACEHOLDER] indicating effective expert specialization.

Key contributions: (1) First dynamic graph-based dataset for ETA prediction incorporating explicit user route intent, (2) Novel DGNN architecture with mixture-of-experts and temporal memory layers, (3) Comprehensive evaluation demonstrating significant accuracy improvements over existing approaches that lack route awareness and dynamic graph modeling.

II. 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], [2], [3] have transformed how travelers plan and adapt their journeys. However, despite significant progress, ETA estimation remains challenging due to the dynamic and stochastic nature of urban traffic.

Traditional approaches to ETA computation rely on shortest-path algorithms such as Dijkstra's algorithm [4], which provide efficient solutions under static conditions but do not account for evolving congestion or interaction effects between vehicles. As a result, travel times computed via such methods can diverge substantially from reality when traffic conditions fluctuate during the trip.

With the growth of urban mobility datasets, machine-learning approaches have been introduced. Tree-based models such as XGBoost [5] predict ETA from handcrafted features (e.g., distance, departure time, day-of-week) and perform well on aggregated trip records like NYC Taxi [6] and Porto Taxi [7]. Deep learning methods improve performance by modeling temporal patterns and spatio-temporal context along a route: 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 learning their dynamics with diffusion or convolution operators (e.g., DCRNN, ST-GCN, Graph WaveNet) [12], [13], [14]. Nevertheless, existing benchmarks (NYC, Porto, Chengdu/DiDi, Geolife) [6], [7], [15], [16] lack explicit representation of the user's **pre-planned route**. Models must therefore infer the likely path between origin and destination, introducing ambiguity and reducing prediction accuracy. Early work proposed incorporating future-traffic–aware route selection to improve ETA accuracy [17]. We build on this direction by explicitly modeling user intent and the pre-planned route within a dynamic spatio-temporal graph, and by learning ETA with a graph-based mixture-of-experts architecture.

This paper addresses these limitations by introducing a novel Dynamic Graph Neural Network (DGNN) that explicitly incorporates user route intent and temporal memory. Our approach represents the transportation network as a dynamic multi-relational graph where both junctions and vehicles are nodes, connected by typed edges that capture road segments, vehicle traversals, traffic interactions, and route intentions. The model employs a mixture-of-experts architecture with specialized experts and a memory layer that records historical snapshots to capture temporal dependencies.

Contributions.
1) We introduce the first dynamic graph-based dataset for ETA prediction that unifies junction states, vehicle dynamics, and explicit pre-planned route information.
2) We propose a novel DGNN architecture that combines graph attention, temporal encoding, and mixture-of-experts specialization for accurate ETA prediction.
3) We conduct comprehensive experiments on simulated traffic data, demonstrating significant accuracy improvements over existing approaches that lack route awareness and dynamic graph modeling.

III. Related Work

A. Classical pathfinding and routing

Modern navigation stacks historically separate **route computation** from **travel-time estimation**. On a static graph $G = (V, E)$ with non-negative edge costs, the canonical approach is to compute the minimum-cost path using Dijkstra's label-setting algorithm [4], then obtain an ETA by summing edge weights along the selected path. This paradigm assumes costs are fixed during the trip and ignores how congestion evolves while the vehicle is en route.

1) Static vs. time-dependent routing: In practice, edge costs are **time-dependent**: the travel time of a road segment $e \in E$ is a function $c_e(t)$ of the departure time t. The **time-dependent shortest path (TDSP)** problem generalizes classical routing by optimizing arrival time given these functions. Under the FIFO (first-in–first-out) property—i.e., departing later cannot lead to an earlier arrival—label-setting methods can be adapted to TD networks; otherwise, label-correcting techniques are required (with weaker optimality guarantees). Despite this, most production systems still approximate TDSP by (i) computing a route under "current" costs and (ii) aggregating ETAs along that route, or by using coarse predictive layers on top of a static route—both approaches remain **route-first** rather than **ETA-first** and can drift as traffic evolves.

2) Speed-up techniques for web-scale routing: Because city- and nation-scale graphs contain millions of nodes, raw Dijkstra is too slow for interactive use. A large body of work speeds up queries via preprocessing:

- **A*** with **ALT landmarks** adds an admissible heuristic via triangle inequalities to reduce the search space [18].
- **Contraction Hierarchies (CH)** contract low-importance vertices to build shortcuts, enabling millisecond-level exact queries on road networks [19].
- **Highway-dimension theory** explains why such hierarchies are effective on real networks and yields provable bounds [20].
- **Arc-flags/SHARC** and **Multi-Level/Customizable Route Planning (CRP)** partition the graph; CRP separates metric-independent preprocessing from a fast "customization" step for new cost profiles [21].

These techniques deliver fast **static** (or piecewise-static) queries. Extensions exist for **time-dependent** costs, but accuracy hinges on the fidelity of $c_e(t)$, and updates must be frequent to track live congestion.

3) Multi-criteria and k-shortest paths: In urban routing, users often optimize multiple criteria (time, tolls, reliability). **k-shortest paths** (e.g., Yen's algorithm) enumerate alternatives for robustness or post-processing [22], yet the ETA still depends on exogenous edge-time models rather than endogenous traffic interactions.

4) Limitations for ETA prediction: Classical routing **treats travel time as input**, not as a learned outcome of interacting flows. Even TDSP with high-quality $c_e(t)$ remains **myopic** if those functions are derived from current/historical averages: it does not anticipate **how vehicles currently en route will reshape future speeds** on the chosen path. Consequently, route-first stacks can systematically under- or over-estimate arrival times when congestion is building or dissipating. This motivates **learning-**

based, graph-temporal approaches that (i) reason over interactions between vehicles and junctions, and (ii) condition ETA on **the intended route** (when available) to remove path ambiguity—directions pursued by our method.

IV. Methodology

A. Graph representation and construction (formal)

Let $t \in \mathbb{Z}$ index uniformly spaced snapshots (interval Δt). The transportation state at time t is a **directed, multi-relational graph**

$$G_t = \big(V_t,\ E_t,\ \{A_t^{(\rho)}\}_{\rho\in\mathcal{R}}\big),$$

with

$$V_t = V^j \ \cup\ V_t^v, \qquad V^j \cap V_t^v = \varnothing,$$

where V^j are **junction nodes** (static across time) and V_t^v are **vehicle nodes** present at time t. Each vehicle i maintains a persistent identity across time; we denote its node at time t by $v_i^t \in V_t^v$.

We consider a finite set of **edge relations** (types)

$$\mathcal{R} = \{\text{road},\ \text{trav},\ \text{inter},\ \text{intent}\}.$$

For each relation $\rho \in \mathcal{R}$ we define a binary adjacency $A_t^{(\rho)} \in \{0,1\}^{|V_t|\times|V_t|}$ and an optional weight matrix $W_t^{(\rho)} \in \mathbb{R}_{\ge 0}^{|V_t|\times|V_t|}$. The (typed) edge set is

$$E_t^{(\rho)} = \{(u,v) : A_t^{(\rho)}[u,v]=1\}, \qquad E_t = \biguplus_{\rho\in\mathcal{R}} E_t^{(\rho)}.$$

Typed edges. At time t,

1) **Road segment edges** ($\rho = \text{road}$): for adjacent junctions $u, v \in V^j$ with legal driving direction $u \to v$,

$$(u,v) \in E_t^{(\text{road})}, \quad A_t^{(\text{road})}[u,v] = 1.$$

2) **Traversal edges** ($\rho = \text{trav}$): if vehicle v_i^t occupies directed segment (a,b) with $a, b \in V^j$,

$$(a, v_i^t) \in E_t^{(\text{trav})}, \qquad (v_i^t, b) \in E_t^{(\text{trav})}.$$

3) **Interaction edges** ($\rho = \text{inter}$): for vehicles $v_i^t, v_j^t \in V_t^v$ on the same segment (or lane group) with longitudinal spacing $d_{ij}(t) \le \varepsilon$ and aligned headings,

$$(v_i^t, v_j^t) \in E_t^{(\text{inter})}, \quad W_t^{(\text{inter})}[v_i^t, v_j^t] = \omega\big(d_{ij}(t)\big),$$

where $\omega : \mathbb{R}_{\ge 0} \to \mathbb{R}_{\ge 0}$ is a monotone kernel (e.g., $\omega(d) = \exp(-d/\lambda)$).

4) **Intent edges** ($\rho = \text{intent}$): let the **planned route** for vehicle i at departure time $t_0 \le t$ be the ordered edge sequence

$$\mathcal{R}_i = \big(e_1, e_2, \ldots, e_K\big), \qquad e_k = (u_k, u_{k+1}) \in E_{t_0}^{(\text{road})}.$$

We connect the vehicle to upcoming junctions along its route,

$$(v_i^t,\ u_{t,i}^{(k)}) \in E_t^{(\text{intent})} \quad \text{for } k = 1, \ldots, K_0,$$

with **decay weights** $W_t^{(\text{intent})}[v_i^t, u_{t,i}^{(k)}] = \alpha_k$ where $\alpha_1 \ge \cdots \ge \alpha_{K_0} > 0$ (e.g., $\alpha_k \in \{1.0, 0.8, 0.6, 0.4, 0.2\}$).

The learning context is a length-H **temporal window**

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

used to predict vehicle ETAs at time t.

V. Results

Run the actual system, evaluate results, and compare to baselines.

VI. Discussion

Reflect on the results. Where did your model perform well? Where did it struggle? Why?

VII. Conclusion

Summarize what was learned and propose next steps.

VIII. Acknowledgments

This should be a simple paragraph before the References to thank those individuals and institutions who have supported your work on this article.

References

[1] A. Derrow-Pinion *et al.*, "ETA prediction with graph neural networks in google maps," in *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, 2021, pp. e.g., 4034–4042. doi: 10.1145/3459637.3481916.

[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, doi: 10.1016/j.compenvurbsys.2020.101518.

[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, doi: 10.1177/0361198118790619.

[4] E. W. Dijkstra, “A note on two problems in connexion with graphs,” *Numerische Mathematik*, vol. 1, no. 1, pp. 269–271, 1959, doi: 10.1007/BF01386390.

[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. doi: 10.1145/2939672.2939785.

[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--.

[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. doi: 10.1109/ITSC.2013.6728228.

[8] J. Wang, Y. Fu, and Z. Zhang, “When will you arrive? Estimating travel time based on deep neural networks,” in *Proceedings of the AAAI conference on artificial intelligence*, 2018, pp. 2500–2507. doi: 10.1609/aaai.v32i1.11772.

[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, doi: 10.3390/app10217599.

[10] X. Wang, J. Li, and N. J. Yuan, “MetaTTE: A meta-learning framework for travel time estimation,” in *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, ACM, 2022, pp. 4080–4088. doi: 10.1145/3534678.3539066.

[11] S. Abbar, A. Anagnostopoulos, S. Bhagat, P. Cudre-Mauroux, and A. Kumar, “STAD: Spatio-temporal adjustment for improving travel-time estimation,” in *Proceedings of the web conference 2020*, ACM, 2020, pp. 2839–2845. doi: 10.1145/3366423.3380127.

[12] Y. Li, R. Yu, C. Shahabi, and Y. Liu, “Diffusion convolutional recurrent neural network: Data-driven traffic forecasting,” in *International conference on learning representations*, 2018.

[13] B. Yu, H. Yin, and Z. Zhu, “Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting,” in *Proceedings of the 27th international joint conference on artificial intelligence*, 2018, pp. 3634–3640. doi: 10.24963/ijcai.2018/505.

[14] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, “Graph WaveNet for deep spatial-temporal graph modeling,” in *Proceedings of the 28th international joint conference on artificial intelligence*, 2019, pp. 1907–1913. doi: 10.24963/ijcai.2019/264.

[15] DiDi Chuxing Research, “Di-tech challenge 2016.” https://outreach.didichuxing.com/research/opendata/en/, 2016.

[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. doi: 10.1145/1526709.1526816.

[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, doi: 10.1109/COMCAS52884.2021.9628700.

[18] A. V. Goldberg and C. Harrelson, “Computing the shortest path: A* search meets graph theory,” in *Proceedings of the sixteenth annual ACM-SIAM symposium on discrete algorithms (SODA)*, 2005, pp. 156–165.

[19] R. Geisberger, P. Sanders, D. Schultes, and D. Delling, “Contraction hierarchies: Faster and simpler hierarchical routing in road networks,” in *International workshop on experimental and efficient algorithms (WEA)*, Springer, 2008, pp. 319–333.

[20] I. Abraham, A. Fiat, A. V. Goldberg, and R. F. Werneck, “Highway dimension, shortest paths, and provably efficient algorithms,” in *Proceedings of the 21st annual ACM-SIAM symposium on discrete algorithms (SODA)*, 2010, pp. 782–793. doi: 10.1137/1.9781611973075.64.

[21] D. Delling, A. V. Goldberg, T. Pajor, and R. F. Werneck, “Customizable route planning,” in *International symposium on experimental algorithms (SEA)*, Springer, 2011, pp. 376–387.

[22] J. Y. Yen, “Finding the k shortest loopless paths in a network,” in *Management science*, 1971, pp. 712–716. doi: 10.1287/mnsc.17.11.712.