Route-Aware Dynamic Traffic Graphs for ETA: Dataset Generation and Interactive Model Testing

Guy Tordjman[1] and Nadav Voloch[2]

[1] Department of Mathematics and Computer Science, The Open University, University Road 1, Ra'anana, Israel
[2] Department of Computer Engineering, Ruppin Academic Center, Emek Hefer, Israel
nadavv@ruppin.ac.il

Abstract. Estimated time of arrival (ETA) predicts trip duration under evolving traffic and underpins navigation, dispatch, and logistics. Small errors degrade routing, coordination, and user trust.
Machine learning approaches to ETA require datasets that align road topology, vehicle motion, and labels in a form models can learn from. We present a toolchain that produces *rich, feature-heavy dynamic graph* data suitable for models that use explicit graph structure—for example, architectures with separate treatment of static network elements and time-varying traffic.
The system supports two complementary paths. First, *simulation-based* datasets: the user supplies a SUMO network and simulation configuration, and the tool materializes spatio-temporal graph snapshots with detailed per-node and per-edge features from microsimulation. Second, *trajectory-based* datasets: real vehicle trajectories are imported and aligned to the same graph representation via map matching and step conversion on a user-selected region. Porto taxi trajectories illustrate real-world conversion in our bundle, and other CSVs and regions are supported.
To demonstrate the value of this representation for learning and comparison, we also built a *web-based testing environment* that couples the simulator to interactive trips, model inference, and logged prediction error, so ETA models can be evaluated under controlled, repeatable conditions.

Keywords: Estimated time of arrival · Traffic simulation · SUMO · Dynamic graphs · Dataset generation · Graph neural networks · Web-based evaluation.

1 Introduction

Estimated time of arrival (ETA) is a central prediction task in transportation systems, informing navigation, dispatch, logistics, and traveler information services. In such settings, ETA is not merely a descriptive quantity: it shapes route choice, scheduling, service reliability, and user trust. For operational platforms, inaccurate or unstable ETA estimates propagate into missed coordination, inefficient routing, and degraded service quality [4].

The central difficulty addressed in this work is not the absence of traffic data in general, but the lack of open, reusable, route-aware data representations tailored to ETA-oriented tasks. Much of the traffic prediction literature relies on sensor-centric or aggregate spatio-temporal datasets, while trajectory-based data often captures movement without preserving the route-aware structure characteristic of navigation-style ETA settings [5]. This problem is compounded by the scarcity of openly reusable ETA datasets: many operationally valuable sources remain proprietary, making reproducible experimentation difficult. The gap is twofold: align road topology, traffic evolution, and route context in one coherent form, and support controlled downstream validation rather than stopping at offline export—ideally for both simulation-derived traffic and real trajectories in one framework.

This paper addresses that gap through a common ETA-oriented representation constructed from two fundamentally different data sources. The first path begins with controlled simulation, where the evolving traffic state and planned routes are directly observable. The second begins with real trajectories, where route inference and data repair are required before the data can be expressed in a coherent downstream form. Although these paths do not produce the same dataset, they produce distinct dataset instances expressed in the same graph-based representation. Roads are modeled as edges, while junctions and vehicles are modeled as nodes. The representation is explicitly route-aware: vehicles are tracked along known or inferred planned routes, making the data substantially closer to navigation-system usage than to generic traffic datasets. It is also dynamic in two senses: traffic evolution appears in time-varying node and edge features, and the graph structure itself changes as vehicles move and dynamic traffic relations are created and removed. While designed primarily for vehicle-level ETA-oriented tasks, the same representation can also support other downstream objectives, such as road-level load estimation or area-level traffic analysis.

The concrete contribution is an integrated artifact ecosystem composed of a desktop dataset-construction tool and a web-based testing environment. The trajectory path is illustrated on the Porto taxi dataset [12] as a bundled conversion *demonstrator*, and other trajectories and geographies are supported. The desktop tool covers both paths: simulation scenarios with SUMO control, zone and fleet validation, and export of route-aware snapshots at chosen Δt, plus trajectories with network preparation, repair, route alignment, and export into the same representation. The web client attaches a compatible ETA predictor for interactive trips, logs outcomes to a database, and supports analysis, plots, and PDF reports. The open-source dataset generator, web evaluation client, and public example datasets are available online [13,14,15,16].

A bundled route-aware predictor exercises the toolchain end-to-end. Section 6 uses it only to demonstrate that the representation is learnable and usable in the web loop, not to advance predictive-model novelty.

The remainder of the paper positions the work (Section 2), presents the system overview (Section 3), the dataset-construction paths and shared representation (Section 4), the web-based evaluation loop (Section 5), and an evaluation snapshot with dataset statistics, metrics, and figures (Section 6), followed by discussion and conclusion (Section 7).

2 Related Work and Positioning

2.1 Traffic data, graph models, and ETA

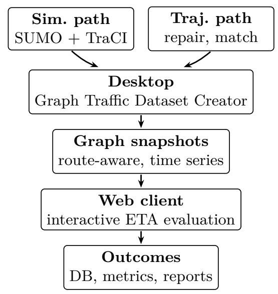
The broader traffic-prediction literature has been shaped by sensor-centric and aggregate spatio-temporal datasets, as reflected in surveys of traffic prediction with big data [5]. Frameworks such as LibCity have made this landscape more reproducible by standardizing datasets, tasks, and model interfaces for urban spatial-temporal prediction [9]. The dominant benchmark orientation remains broad traffic forecasting rather than route-aware ETA estimation in which the planned route of each vehicle is explicit. ETA tasks are inherently path-dependent, whereas many widely used resources primarily capture network state, trajectories, or aggregate flows without preserving a navigation-style route context.

Graph-based learning has become a standard approach to traffic forecasting because it aligns naturally with road-network structure. Models such as DCRNN and ST-GCN illustrate explicit graph formulations for spatial and temporal dependencies [7,6]. In the ETA domain, production-scale work on ETA prediction with graph neural networks in Google Maps underscores structured, route-aware modeling for travel-time estimation [8]. Related routing-oriented work explores future-traffic-aware route choice and simulative ETA estimation [10,11]. Our emphasis is reusable route-aware data and integrated tooling to produce and test it.

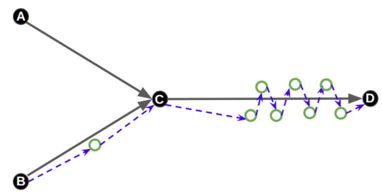
2.2 Simulation, trajectories, and integrated evaluation

Microscopic traffic simulation, with SUMO and TraCI, provides repeatable scenarios and external programmatic control [1,2], and is attractive for ETA-oriented dataset construction because vehicle state, route context, and traffic evolution are highly observable. Real trajectory data requires repair, alignment, and route inference before coherent downstream ETA analysis. Simulation and trajectory pipelines are often optimized separately, while this work supports both in one common representation.

Existing libraries and evaluation frameworks improve reproducibility for traffic prediction but typically start from pre-existing datasets rather than from integrated dataset construction. Here, a web evaluation layer is coupled to generation: it attaches a compatible predictor, registers outcomes, and exposes analysis under controlled conditions.



(**a**) End-to-end workflow



(**b**) Static vs. dynamic graph layers

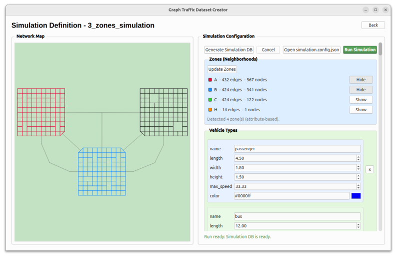
Fig. 1. (**a**) End-to-end workflow: simulation and trajectory paths merge in the desktop tool, export route-aware graph time series, and feed the web layer for testing and logged analysis. (**b**) Static layer: black nodes A–D are junctions, and grey edges are roads. Dynamic layer: green nodes are vehicles, and dashed blue edges connect vehicles to other vehicles and to junctions.

3 System Overview

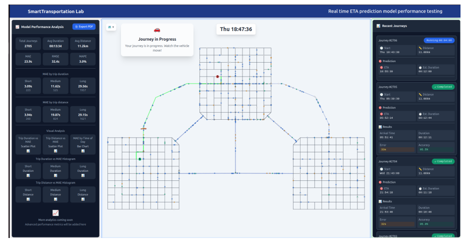
The proposed workflow links heterogeneous traffic inputs to downstream ETA-oriented evaluation through a single artifact ecosystem. At a high level, the system begins from one of two data-construction paths: a simulation-derived path built on controlled SUMO scenarios, and a real-trajectory path built on imported GPS-like traces aligned to a road network. Although these paths differ substantially in observability, preprocessing, and error sources, both terminate in a common graph-based representation designed for route-aware ETA tasks. This shared representation can then be exercised through a web-based evaluation environment in which a compatible predictor is attached, trip-level outcomes are registered, and statistical analyses are produced. Figure 1(**a**) summarizes this pipeline, and panel (**b**) sketches the static and dynamic parts of the graph representation.

Dataset construction is operationalized through the Graph Traffic Dataset Creator, an interactive desktop environment rather than a collection of disconnected scripts. On the simulation side, it supports scenario definition, control of simulation behavior, visual validation of zones, landmarks, and vehicle classes, and export of route-aware graph snapshots with adjustable temporal granularity, including configurable noise such as random untracked trips. On the trajectory side, it supports map download, network preparation, trajectory repair, route alignment, and batch export into the common representation. Exports are written in a single schema so training code and the web evaluation client consume the same node/edge feature layout, which reduces mismatch bugs between offline dataset generation and interactive evaluation.

The conceptual center of the system is the common ETA-oriented representation: roads as edges, junctions and vehicles as nodes, with each vehicle associated with a known or inferred route. Traffic state evolves in node and edge features, and graph relations change with vehicle movement. The representation is aimed



(**a**) Simulation path: zones and scenario setup (desktop)



(**b**) Web client: evaluation on simulation-backed trips

Fig. 2. Simulation construction path and its web evaluation loop. (**a**) Simulation Definition: SUMO network map with attribute-based zones (A/B/C/H), zone statistics, and configuration before database generation and runs (example project). (**b**) Real-time ETA testing: model performance summaries, interactive map with active route, and journey log.

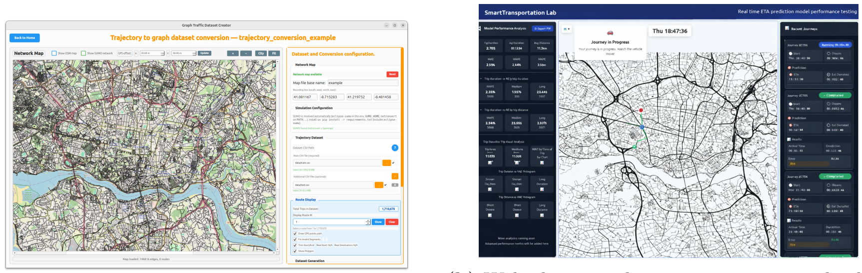
at vehicle-level ETA but supports other views such as road-level load or area-level aggregates. The web evaluation client provides the downstream evaluation layer: interactive testing with an attached ETA predictor, outcomes stored in a database, and statistical summaries, plots, and report generation, bridging representation design and measurable use. Compatible predictors can be attached, replaced, and tested in the same environment, including by external users of the open-source release. The cloud-deployed instance discussed later is one demonstrator configuration of the broader workflow.

4 Dataset Construction and Shared Representation

4.1 Goals and construction paths

The representation is designed first for route-aware ETA learning while remaining useful beyond a single supervised target: it places road topology, vehicles, route context, and time-varying traffic state in one coherent form for controlled experimentation and downstream testing. Because it preserves graph semantics rather than only a final ETA label, it can also support related tasks such as road-level load estimation and area-level analysis.

In the simulation-derived path, construction begins from a controlled microscopic environment built with SUMO and managed through the desktop tool and TraCI [1,2]. Planned routes, vehicle identities, network structure, and evolving traffic states are directly observable during execution. The desktop workflow is configurable: simulation behavior, validation of zones, landmarks, and vehicle classes, temporal sampling of snapshots, and optional irregularity such as random untracked trips for robustness experiments. Figure 2(**a**) shows the simulation-definition stage, and Figure 2(**b**) shows the web client used to evaluate predictors on simulation-backed journeys.



(**a**) Trajectory path: conversion (desktop)

(**b**) Web client: evaluation on trajectory-backed trips

Fig. 3. Trajectory construction path and its web evaluation loop. (**a**) Trajectory dataset conversion: OSM/SUMO map, network overlay, CSV inputs, and batch conversion controls (example Porto-area project). (**b**) Same web client as in Figure 2(**b**), here exercised on exports aligned with real trajectories: map with active route, aggregate metrics, and recent journeys.

The real-trajectory path starts from noisier traces: GPS inaccuracies, temporal inconsistencies, or jumps may break a coherent path interpretation. Conversion is a staged repair-and-alignment workflow: network preparation, import, repair, route inference or matching, slicing of trajectories with major jumps, and filtering of degenerate origin–destination pairs. We use the Porto taxi trajectory dataset [12] as a representative real-world conversion example in our open-source bundle. The toolchain applies to other CSVs and regions. Figure 3(**a**) shows the desktop conversion interface, and Figure 3(**b**) shows web-side evaluation on trajectory-backed runs. Figure 4 visualizes route-matching outputs on the map: search scope, candidates, raw GPS, the chosen SUMO route, and projections onto it.

4.2 Route matching on the network (abstract procedure)

Matching is not run on the full regional road graph: first, an axis-aligned bounding box B is built around the GPS polyline P (optionally padded), and $N_{\text{bb}} = (V_{\text{bb}}, E_{\text{bb}})$ is the subnetwork of N formed from all edges whose geometry intersects B and their incident nodes. GPS polylines are then aligned to N_{bb} in two phases: a *scoring pass* assigns every edge a coarse tier (1, 10, or 1000) from successive segment–edge comparisons, then a *routing pass* searches for a low-cost path that respects those tiers. For each polyline segment, edges whose heading differs from the segment by more than $\pi/9$ are discarded. Among survivors, edges are ranked by $r_e = \lambda\delta_e + \frac{1}{2}\big(d(p_i, e) + d(p_{i+1}, e)\big)$, where δ_e is the segment–edge heading gap in radians, $d(p, e)$ is the distance from point p to the centreline polyline of edge e, and λ fixes the radian-to-metre (map-unit) trade-off. The first k edges become *primary* candidates (tier 1), and the next j become *secondary* (tier 10). Edges that never qualify as a potential match in any segment keep tier 1000. Whenever an edge appears among the top-k candidates for a segment, its tier is set to 1 outright. Secondary (10) updates use min with the

(**a**) Bounding box and search scope (**b**) Split trajectory after a GPS outlier

Fig. 4. Route matching on the imported SUMO network (conversion path). Panels (**a**) and (**b**) both show the *bounding box*, *burgundy* roads and junctions that limit the matcher's search entities, and the thick *dark* final route on the SUMO network. Panel (**a**) additionally shows the raw GPS trace as a *light blue* polyline (numbered blue samples) and *green* segments as the top-three candidate edges at sampled locations. Panel (**b**) additionally shows *light blue* and *orange* polylines for the *split* trajectory—two segment polylines after cutting at an outlying GPS point (e.g. $\sim$5 km from adjacent samples), with each segment matched separately.

current tier. This prevents an edge already assigned tier 1 from being raised to 10 by a later segment. The routing pass runs A* exactly k times: each run starts from one edge in Top_1 (the k primaries for the first segment) and *terminates* as soon as the search reaches any of the *last* primaries—any edge in Top_{m-1} for segment (p_{m-1}, p_m)—which form the goal set. Edge costs are tier-weighted, and the heuristic is Euclidean distance to p_m. The cheapest route among the k runs is returned. Algorithms 1–2 state this procedure formally.

Algorithm 1 Edge-weight scoring pass for a GPS polyline

Require: Polyline $P = (p_1, \dots, p_m)$, bounding-box subnetwork $N_{\mathrm{bb}} = (V_{\mathrm{bb}}, E_{\mathrm{bb}})$ with edge headings ψ_e, integers $k, j \geq 1$, angle threshold $\theta_{\max} = \pi/9$, and trade-off $\lambda > 0$.
Ensure: Tier $w(e) \in \{1, 10, 1000\}$ for every $e \in E_{\mathrm{bb}}$ (lower is better).
$w(e) \leftarrow 1000$ for all $e \in E_{\mathrm{bb}}$
for each segment $s_i = (p_i, p_{i+1})$, $i = 1, \dots, m-1$ **do**
  $\phi_i \leftarrow$ heading of s_i, and $\mathcal{C} \leftarrow \emptyset$
  for each edge $e \in E_{\mathrm{bb}}$ with heading gap $\delta_e = |\phi_i - \psi_e| \leq \theta_{\max}$ **do**
    $r_e \leftarrow \lambda \delta_e + \frac{1}{2}\big(d(p_i, e) + d(p_{i+1}, e)\big)$, and add e to $\mathcal{C}$
  end for
  Sort $\mathcal{C}$ by increasing r_e. Let Top_i be the first k edges, Sec_i the next j
  $w(e) \leftarrow 1$ for $e \in \mathrm{Top}_i$, and $w(e) \leftarrow \min(w(e), 10)$ for $e \in \mathrm{Sec}_i$
end for

Finally, GPS samples are projected onto the inferred SUMO route so vehicle position can be aggregated as a time series in the shared representation.

Algorithm 2 Route extraction by multi-start A*

Require: N_{bb} with tiers $w(e)$, primary sets $\mathrm{Top}_1, \mathrm{Top}_{m-1}$ from Algorithm 1, and trip endpoint p_m.

Ensure: Lowest-cost route among k multi-start runs.

1: Goal set $G \leftarrow \mathrm{Top}_{m-1}$
2: **for** each start edge $e \in \mathrm{Top}_1$ **do**
3: Run A* from e with cost $c(e') = w(e') \cdot \mathrm{len}(e')$ and heuristic h = Euclidean distance to p_m, stopping as soon as the search reaches G
4: **end for**
5: **return** the cheapest of the k resulting routes

4.3 Shared representation, dynamics, and use

Both paths emit dataset instances in the same conceptual form: roads as edges, junctions and vehicles as nodes, with planned or inferred routes so the graph encodes route-conditioned traffic context, not only a traffic snapshot. Exports are feature-rich at node and edge level. Figure 1(**b**) illustrates static junctions and roads together with vehicle nodes and dynamic edges (vehicle–vehicle and vehicle–junction). The current implementation is therefore road-vehicle oriented. A multimodal extension would fit the same graph principle only after adding typed transport nodes and edges, mode-specific features, and transfer or interchange semantics between modes.

Each snapshot is tied to a simulation step or to a trajectory update instant and bundles the static road/junction substrate with the active vehicle set, its routes, and the current traffic descriptors on touched edges and nodes. That alignment matters for learning: models trained on SUMO exports and models trained on converted trajectories ingest the same logical tensors even though the emission times are regular in one branch and event-driven in the other.

The graph is dynamic in two senses: node and edge attributes evolve with traffic conditions, and structure changes as vehicle nodes move and dynamic relations are created or removed. In each emitted snapshot, the stable road graph is retained, while transient vehicle relations are rebuilt only on occupied road edges by ordering active vehicles along the edge and linking neighboring vehicle nodes. This keeps the update local and bounds the per-edge cost by the number of vehicles that can physically occupy the segment, while total construction cost still increases under rush-hour or high-volume conditions. Downstream users may take the full dynamic graph or simplified views, including vehicles, roads, junctions, and optional filtering of dynamic relations.

5 Web-Based Evaluation and Validation Loop

5.1 Workflow, interface, and evidence

The web application closes the loop between dataset construction and measurable use. It accepts representation-compatible inputs, runs prediction sessions under controlled conditions, and persists evidence for later inspection, so the representation is exercised end-to-end rather than left as a static export. The environment supports interactive ETA-oriented testing, cumulative storage, and repeatable analysis. It is not a production navigation system, but it shows navigation-style interaction is feasible on the representation. Section 4 places the simulation path first: Figure 2(**b**) shows the main evaluation dashboard for simulation-backed trips (map, cohort metrics, journey log). The trajectory path uses the same client after export, as in Figure 3(**b**), following desktop conversion in Figure 3(**a**).

A compatible ETA predictor is attached as a pluggable component that consumes the shared representation and returns trip-level ETA outputs. Users can trigger predictions for selected trips, inspect outputs, and run controlled sessions. Other predictors can be substituted if they conform to the representation interface. Predictions and metadata are stored in a database so results can be revisited and aggregated. The client computes measures such as MAE, RMSE, and MAPE over selected sets and supports cohort grouping. It also generates plots from registered data and can assemble PDF reports combining statistics and figures. In typical use, an operator loads an exported bundle, selects trips or cohort filters on the map, runs a batch of evaluations, and inspects per-journey logs alongside aggregate error panels. The same session state can be exported for archival or for inclusion in supplementary material without re-running inference.

5.2 Demonstrator scope and validation role

The deployed application is one configuration of a broader framework: the same logic can be reused with alternative predictors and datasets from supported environments. The contribution is the coupling of construction and evaluation through a reusable interface, not a single prototype deployment.

The web loop validates representation coherence, downstream trainability, and a workflow that does not stop at raw export, with continuity from construction to evaluation.

6 Evaluation

We evaluate the same route-aware predictor attached to the web environment on two representative bundles from the open-source workflow: one from the simulation path and one from repaired real trajectories. For each source, performance is measured on held-out journeys registered through the evaluation client, using MAE, RMSE, and MAPE. As a sanity-check comparator, we also report a simple mean-duration baseline that predicts the training-set average trip duration for every test journey. Test journeys are disjoint from the trips used to fit the route-aware model for that bundle, and the mean baseline uses only training-trip durations, so both scores refer to the same held-out set. The web client persists each prediction with its realized duration, which makes the reported aggregates reproducible from the logged database and ties the numbers to the interactive evaluation protocol rather than to an ad-hoc offline script.

Table 1. Representative open-source bundles. Junction/road-edge counts use non-internal SUMO `net.xml` edges. Coverage uses the `location` bbox. Trajectory snapshots are actual trace-driven emissions.

Quantity	Simulation	Trajectory conversion
Temporal span	4 weeks	1 year
Project name	3 zones city	Porto taxi
Trips after repair	403 848	1 312 637
Junction nodes	1 031	75 264
Road edges (non-internal)	1 294	146 816
Approx. coverage area (km^2)	$\approx$ 203	$\approx$ 385
Graph snapshots (at Δt)	80 640	925 644
Snapshot period Δt (s)	30	30
Node feat. dim. / edge feat. dim.	28 / 6	28 / 6

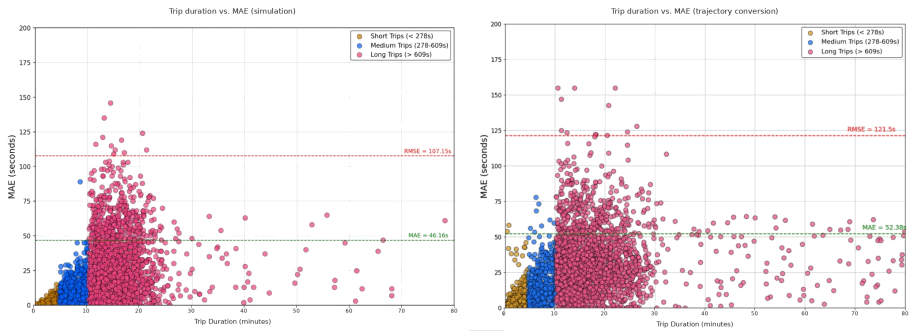
Table 2. ETA error on held-out journeys: route-aware model vs. train-set mean-duration baseline. MAE/RMSE are seconds, MAPE is percent.

Data source	n	Route-aware model			Mean baseline (avg)		
		MAE (s)	RMSE (s)	MAPE (%)	MAE (s)	RMSE (s)	MAPE (%)
Simulation	2 707	**46.16**	**107.15**	**17.99**	260.63	325.13	127.75
Trajectory conversion	3 055	**52.38**	**121.50**	**20.41**	282.66	352.41	138.50

Table 1 summarizes the two dataset instances used in this snapshot. The simulation-backed example spans four weeks with dense 30 s sampling and 80 640 snapshots. The trajectory example spans one year of Porto taxi data, with 925 644 emitted graph snapshots driven by repaired trajectory observations rather than continuous sampling over the full calendar period. Both branches therefore differ in scale and observability but share the same node/edge feature interface.

Table 2 shows that the representation supports substantially better ETA estimation than the naive average-duration baseline in both settings. On simulation-backed data, the route-aware predictor reduces MAE from 260.63 s to 46.16 s (about 82% relative improvement), while RMSE drops from 325.13 s to 107.15 s. On the trajectory-conversion data, MAE falls from 282.66 s to 52.38 s (about 81% relative improvement), while RMSE falls from 352.41 s to 121.50 s. MAPE follows the same trend, dropping from well above 100% under the baseline to 17.99% and 20.41%. These numbers indicate that the shared graph representation carries enough route and traffic context for the downstream predictor to learn nontrivial travel-time structure across both branches.

The scatter plots in Figure 5 show how absolute error varies with trip duration: most journeys fall in short-to-medium durations, with a heavy tail of longer trips. Trajectory conversion is slightly harder, consistent with repair, map matching, and sparser observability, yet errors stay far below the baseline in Table 2, supporting end-to-end usefulness of the workflow.



(**a**) Simulation path (**b**) Trajectory conversion

Fig. 5. Trip duration vs. absolute ETA error (MAE in seconds) on held-out evaluation journeys. Horizontal lines show aggregate model MAE (green) and RMSE (red) for the route-aware predictor. (**a**) Microsimulation-backed graph data. (**b**) Real trajectories after map matching and conversion.

7 Discussion and Conclusion

The main contribution of this paper is an integrated workflow for constructing route-aware ETA-oriented graph datasets and validating their downstream usability through a connected web evaluation environment. The desktop tool supports construction from both controlled simulation and repaired real trajectories. The web environment turns those outputs into persistent, analyzable evaluation evidence, narrowing the gap between dataset preparation and downstream experimentation. By keeping schema, export, training, and interactive testing in one lineage, the artifacts are easier to share and to critique: reviewers and readers can relate dataset statistics, model checkpoints, and logged web sessions without reconciling incompatible file formats or hidden preprocessing steps.

At the same time, the present demonstrator has clear limitations. The deployed web instance represents one practical configuration of the broader framework, not the full space of possible deployments, networks, or predictors. The real-trajectory branch also remains inherently constrained by source-data noise, route-inference uncertainty, and data-availability limits that do not arise in the same way under controlled simulation. More broadly, the current paper emphasizes workflow coherence and proof of usability over exhaustive comparative experimentation.

Future work includes alternative predictors, additional networks and trajectory sources, and larger-scale experiments. Integration with navigation clients for live or near-real-time testing could further stress the evaluation loop. We also see value in packaging additional reference scenarios (network scale, demand patterns, sensor noise) so teams can compare models under shared construction assumptions rather than only sharing opaque CSV extracts. Overall, the toolchain is a reusable foundation for route-aware dataset construction and controlled ETA-oriented research.

References

1. Behrisch, M., Bieker, L., Erdmann, J., Krajzewicz, D.: SUMO—Simulation of Urban MObility: An Overview. In: Proceedings of SIMUL (2011)
2. López, P.A., et al.: Microscopic Traffic Simulation using SUMO. In: The SUMO traffic simulation in 21 days (2018)
3. OpenStreetMap contributors: `https://www.openstreetmap.org`. Accessed 2026
4. Modica, T., Tappia, E., Colicchia, C., Melacini, M.: Integrating arrival time estimation in truck scheduling: an explorative study in grocery retailing. Production & Manufacturing Research 12(1), Article 2425678 (2024). `https://doi.org/10.1080/21693277.2024.2425678`
5. Yu, J.J.Q., Gu, Y., Li, H., Li, S.: A survey on traffic prediction using big data. IEEE Transactions on Intelligent Transportation Systems 18(10), 2570–2586 (2017)
6. Yu, B., Yin, H., Zhu, Z.: Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting. In: Proceedings of IJCAI (2018)
7. Li, Y., Yu, R., Shahabi, C., Liu, Y.: Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting. In: Proceedings of ICLR (2018)
8. Derrow-Pinion, A., She, J., Wong, D., Lange, O., Hester, T., Perez, L., Nunkesser, M., Lee, S., Guo, X., Wiltshire, B., Battaglia, P.W., Gupta, V., Li, A., Xu, Z., Sanchez-Gonzalez, A., Li, Y., Veličković, P.: ETA prediction with graph neural networks in Google Maps. In: Proceedings of the 30th ACM International Conference on Information and Knowledge Management, pp. 3767–3776 (2021)
9. Wang, J., Jiang, J., Jiang, W., Li, C., Zhao, W.X.: LibCity: An open library for traffic prediction. In: Proceedings of the 29th International Conference on Advances in Geographic Information Systems, pp. 145–148 (2021)
10. Voloch, N., Voloch-Bloch, N.: Finding the fastest navigation route by real-time future traffic estimations. In: Proceedings of the IEEE International Conference on Microwaves, Communications, Antennas and Electronic Systems (COMCAS) (2021)
11. Voloch, N., Penkar, N., Tordjman, G.: 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), Limassol, Cyprus (2025)
12. Moreira-Matias, L., Gama, J., Ferreira, M., Mendes-Moreira, J., Damas, L.: ECML/PKDD 2015 competition: Predict taxi service trajectory (I)—Porto, Portugal, GPS records (July 2013–June 2014). Competition dataset documentation and archive (2015). `https://www.kaggle.com/c/pkdd-15-predict-taxi-service-trajectory-i`. Accessed 2026
13. Ruppin-SmartTransportation: SUMO Data Generator: A tool to generate simulation-derived dynamic traffic datasets. `https://github.com/Ruppin-SmartTransportation/sumo_data_generator`. GitHub repository (2024)
14. Ruppin-SmartTransportation: TrafficLab: Test environment for traffic model validation. `https://github.com/Ruppin-SmartTransportation/TrafficLab`. GitHub repository (2024)
15. Tordjman, G.: 3-Zone City Traffic Simulation Dataset. `https://www.kaggle.com/datasets/guytordjman/3-zones-city-traffic-simulation`. Kaggle dataset (2026)
16. Tordjman, G.: Porto Taxi Converted to Graph Snapshots. `https://www.kaggle.com/datasets/guytordjman/porto-taxi-converted-to-graph-snapshots`. Kaggle dataset (2026)