CENTER FOR CONNECTED AND AUTOMATED TRANSPORTATION

Final Report 135
November 2026



Toward Ubiquitous Trajectory-Based Traffic Network Diagnosis Systems

Peisen Li
Zhichen Liu
Zihao Wang
Henry X. Liu
Yafeng Yin

UNIVERSITY OF MICHIGAN

CENTER FOR CONNECTED AND AUTOMATED TRANSPORTATION

Report No. 135 **November 2026**

Project Start Date: 10/15/2025

Project End Date: 10/15/2026

Toward Ubiquitous Trajectory-Based Traffic Network Diagnosis Systems

Final Report

by

Peisen Li, Zhichen Liu, Zihao Wang,
Henry X. Liu, and Yafeng Yin

University of Michigan

CENTER FOR CONNECTED AND AUTOMATED TRANSPORTATION

DISCLAIMER

Funding for this research was provided by the Center for Connected and Automated Transportation under Grant No. 69A3552348305 of the U.S. Department of Transportation, Office of the Assistant Secretary for Research and Technology (OST-R), University Transportation Centers Program. The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the information presented herein. This document is disseminated under the sponsorship of the Department of Transportation, University Transportation Centers Program, in the interest of information exchange. The U.S. Government assumes no liability for the contents or use thereof.

Suggested citation:

Li, P., Liu, Z., Wang, Z., Liu, H. X., & Yin, Y. (2026). Toward ubiquitous trajectory-based traffic network diagnosis systems: Final report. Center for Connected and Automated Transportation, University of Michigan.
DOI: https://doi.org/10.7302/dspace/31069

Contacts

For more information:

Yafeng Yin
University of Michigan
Ann Arbor, MI 48109
Phone: (734) 647-4979
Email: yafeng@umich.edu

CCAT
University of Michigan Transportation Research Institute
2901 Baxter Road
Ann Arbor, MI 48109
Email: umtri-ccat@umich.edu
Phone: (734) 763-2498

CENTER FOR CONNECTED AND AUTOMATED TRANSPORTATION

Technical Report Documentation Page

1. Report No. Final Report 135	**2. Government Accession No.**	**3. Recipient's Catalog No.**
4. Title and Subtitle Toward Ubiquitous Trajectory-Based Traffic Network Diagnosis Systems Final Report DOI: https://doi.org/10.7302/dspace/31069		**5. Report Date** November 2026 **6. Performing Organization Code**
7. Author(s) Peisen Li (0009-0004-6823-9488), Zhichen Liu (0000-0001-6178-9883), Zihao Wang (0000-0002-5497-6100), Henry X. Liu (0000-0002-3685-9920), Yafeng Yin (0000-0003-3117-5463).		**8. Performing Organization Report No.**
9. Performing Organization Name and Address University of Michigan Ann Arbor, MI 48109		**10. Work Unit No.** **11. Contract or Grant No.** 69A3552348305
12. Sponsoring Agency Name and Address U.S. Department of Transportation Office of the Assistant Secretary for Research and Technology 1200 New Jersey Avenue, SE, Washington, DC 20590		**13. Type of Report and Period Covered** Final report; 10/15/2025–10/15/2026 **14. Sponsoring Agency Code** OST-R

15. Supplementary Notes
Conducted under the U.S. DOT Office of the Assistant Secretary for Research and Technology's University Transportation Centers program through CCAT.

16. Abstract
This report presents four complementary advances toward trajectory-based traffic-network diagnosis: a unified inverse-learning framework, scalable multiconvex equilibrium estimation, distributional equilibrium modeling, and count-calibrated demand generation. An Ann Arbor inverse-learning study achieved a best-setting link travel-time mean absolute percentage error of 34.3 percent versus 83.6 percent for an annual-average benchmark. A separate demand study obtained R-squared of 0.716 and normalized root-mean-square error of 34.1 percent on links excluded from origin–destination calibration. Controlled distributional experiments demonstrate risk-sensitive decisions, including a trade-off between lower tail loss and higher average total travel time in Chicago Sketch. The studies establish the fundamentals for constructing a trajectory-based diagnosis system for traffic networks.

17. Key Words traffic network diagnosis; trajectory data; inverse learning; user equilibrium; demand generation; uncertainty quantification; origin–destination calibration	**18. Distribution Statement** No restrictions.		
19. Security Classif. (of this report) Unclassified	**20. Security Classif. (of this page)** Unclassified	21. No. of Pages 61	**22. Price** Not used

Form DOT F 1700.7 (8-72) Reproduction of completed page authorized

EXECUTIVE SUMMARY

This project developed complementary methods for constructing traffic-network models from incomplete vehicle trajectories and sparse flow counts. The work combines an inverse-learning framework, scalable context-dependent equilibrium estimation, distributional equilibrium modeling, and calibrated demand generation. The studies establish methodological and empirical foundations for network diagnosis.

The contributions address different parts of the modeling problem. Demand generation supplies candidate trips and an origin–destination (OD) prior; count calibration adjusts demand under fixed route shares; inverse learning estimates equilibrium relationships; and distributional learning represents repeated-day variability for risk-sensitive decisions. These connections define a proposed architecture. The two Ann Arbor studies use different networks and observations, while the distributional experiments use controlled synthetic data.

For scalable inverse learning, a potential that is linear in nonnegative coefficients and convex in flows yields a multiconvex formulation based on equilibrium suboptimality gaps. Parallel block coordinate descent exploits its sample-wise structure. On the 810-node, 2,847-link Ann Arbor network, the best reported setting achieved link travel-time mean absolute percentage error (MAPE) of 34.3 percent, with a standard deviation of 1.9 percentage points across three runs, versus 83.6 percent for an annual-average benchmark. The reported runtime was 1.9 minutes. The comparison supports the learned model as a whole; it does not isolate the contribution of context alone.

The demand-generation study combines constrained trip generation with regularized OD estimation. On a separate Ann Arbor network, validation links excluded from OD calibration had R-squared of 0.716 and normalized root-mean-square error (%RMSE) of 34.1 percent, compared with 0.649 and 37.8 percent when OD matrix estimation (ODME) used the probe-trip pool. Generated routes broadened the support available for calibration. These results support demand construction, while population-level behavioral realism and final simulator performance require additional validation.

Distributional user equilibrium maps latent cost regimes into feasible equilibrium flows. Sioux Falls experiments showed coverage near the nominal level and useful risk-sensitive pricing results. Chicago Sketch yielded the lowest non-oracle point-prediction error, but nominal 95 percent intervals covered 99.9 percent of observations. Its pricing solution reduced conditional value-at-risk (CVaR) by about 1.4 percent relative to deterministic user equilibrium (UE) while increasing average total travel time by about 21.5 percent. The results demonstrate a trade-off between tail risk and mean performance under known synthetic conditions, rather than uniformly superior calibration or field readiness.

Taken together, the work advances the planning-level foundation for ubiquitous diagnosis: it makes model assumptions learnable from repeated trajectory observations; uses context to explain systematic variation; expands point predictions into uncertainty-aware distributions; and creates calibrated demand from incomplete probes.

Table of Contents

Executive Summary **1**

List of Figures **4**

List of Tables **5**

1. Introduction and Project Context **6**

1.1 Motivation 6

1.2 Research scope and contributions 6

1.3 Report organization 7

2. Project Objectives and Integration Framework **7**

2.1 Project objectives 7

2.2 Data-to-diagnosis logic 8

2.3 Distinct data regimes 9

2.4 Component interfaces and compatibility requirements 9

2.5 Levels of inference and diagnostic claims 10

2.6 Cross-component assumptions and failure propagation 11

3. Inverse Learning of Network Equilibrium **12**

3.1 Forward equilibrium as a structural layer 12

3.2 Unified inverse-learning problem 12

3.3 Equilibrium-gap choices 13

3.4 Observation, identifiability, and validation 14

3.5 Mathematical foundations and notation 14

3.6 Detailed inverse-learning formulations 18

4. Scalable Context-Dependent and Nonparametric Learning **21**

4.1 Context-dependent equilibrium 21

4.2 Convex-basis potential 22

4.3 Multiconvex formulation and parallel algorithm 22

4.4 Evidence from synthetic and empirical tests 23

4.5 Multiconvex algorithm and scalable computation 24

4.6 Implementation scope and reproducibility 26

5. Distributional User Equilibrium and Risk **26**

5.1 From one equilibrium to a distribution 26

5.2 Conditional variational learning and differentiable equilibrium 27

5.3 Uncertainty calibration and risk-sensitive pricing 27

5.4 Distributional equilibrium theory and risk-sensitive decisions 28
5.5 Further validation needs 31
6. City-Scale Trajectory Generation and OD Calibration 32
6.1 Problem setting and design rationale 32
6.2 Notation and available observations 33
6.3 Trip generation with a constrained, context-aware language model 35
6.4 Demand-prior extraction 39
6.5 OD demand calibration 41
6.6 Trip resampling 42
6.7 Scope, assumptions, and extensions 43
7. Case Studies and Validation Findings 43
7.1 Ann Arbor inverse-learning study 43
7.2 Demand-generation benchmarks and Ann Arbor count-validation study 45
7.3 Controlled distributional benchmarks 51
7.4 Priorities for further validation 51
7.5 Evidence synthesis across the component studies 52
8. Findings 53
8.1 Technical findings 53
8.2 Cross-cutting limitations 53
9. Recommendations 54
9.1 Model selection and evaluation 54
9.2 Guidance for agency use 54
9.3 Reporting and interpretation of results 55
10. Outputs, Outcomes, and Impacts 55
10.1 Research outputs 55
10.2 Outcomes 56
10.3 Impacts and beneficiaries 56
11. Challenges and Lessons Learned 56
11.1 Technical challenges 56
11.2 Lessons for project execution 57
12. Performance Indicator Synopsis 57
12.1 Documented research products 57
12.2 Indicators requiring administrative records 57
References 58

List of Figures

Figure 1. Proposed integration architecture linking trajectory observations, demand construction, equilibrium learning, uncertainty, and diagnosis. The component studies were evaluated separately. 8

Figure 2. Three principal modeling choices in inverse equilibrium learning: the priority between fit and equilibrium consistency, the performance metric or gap formulation, and the parameterization of the potential function. Source: Liu and Yin (2026). 13

Figure 3. Forward and inverse views of context-dependent equilibrium learning from multiday noisy link-flow observations. Source: Liu, Yin, Lin, and Wang (2026). 22

Figure 4. Multiconvex decomposition and parallel block coordinate descent for the scalable inverse-learning problem. Source: Liu, Yin, Lin, and Wang (2026). 23

Figure 5. General distributional inverse-learning architecture linking features, a generative latent layer, a cost function, an equilibrium solver, observed flows, and an output flow distribution. Source: Li et al. (2026). 27

Figure 6. Four-module workflow for calibrated demand generation: constrained trip generation, demand-prior extraction, OD calibration, and resampling. Source: Wang et al. (2026). 33

Figure 7. Architecture of the conditional language model (Module 1). Network context is encoded by a graph convolutional network and fused into the autoregressive backbone through cross-attention; logits processors LP1 to LP3 constrain decoding. Source: Wang et al. (2026). 37

Figure 8. Ann Arbor assignment network used in the context-dependent inverse-learning study. Source: Liu, Yin, Lin, and Wang (2026). 44

Figure 9. Observed differences in Ann Arbor trip patterns across snow conditions and day types. Source: Liu, Yin, Lin, and Wang (2026). 45

Figure 10. Generated versus training distributions of departure time (left, JSD = 0.0002) and route length (right, JSD = 0.0005) in the Ann Arbor demand study. Source: Wang et al. (2026). 48

Figure 11. TAZ-level route patterns for training probe trips and generated trips across entering, transiting, exiting, retail-destination, and medical-center-origin examples in Ann Arbor. Source: Wang et al. (2026). 49

Figure 12. Generated versus observed hourly link volumes for the four Ann Arbor demand methods on calibration and held-out validation links. Source: Wang et al. (2026). 50

List of Tables

Table 1. Research components and supporting evidence. 7

Table 2. Original project tasks and documented accomplishments. 8

Table 3. Required interfaces for future integration of the component methods. 10

Table 4. Inverse-learning design choices and their consequences for network diagnosis. 14

Table 5. Core mathematical objects used in the detailed report. 18

Table 6. Comparison of inverse-learning formulations and their practical consequences. 21

Table 7. Selected evidence for scalable context-dependent inverse learning. 23

Table 8. Selected Distributional UE proof-of-concept results. Values retain the units and scenario definitions of the source manuscript. 28

Table 9. Evaluation layers for Distributional User Equilibrium. 31

Table 10. Inputs, outputs, and roles of the four demand-generation modules. 33

Table 11. Notation used in Section 6. 34

Table 12. Datasets used in the demand-generation experiments. 46

Table 13. Distribution-level evaluation of individual trip generation on public benchmarks. Lower is better; all columns except Query Error are Jensen–Shannon divergences. Bold indicates the best value and underlining indicates the second-best value in each column. 47

Table 14. Diversity and feasibility of the six-million-trip Ann Arbor generated pool before OD calibration and resampling. 48

Table 15. Hourly link-volume reproduction in the Ann Arbor demand study. Calibration links were used by ODME; validation links were held out. Lower MAE, %RMSE, WAPE, and median GEH and higher R-squared and GEH<5 share are better. Bold marks the best value in each column of each block. 50

Table 16. Evidence boundaries across the integrated report. 52

Table 17. Research outputs and publication sources. 55

1. Introduction and Project Context

1.1 Motivation

Transportation agencies increasingly receive large volumes of passively collected location and trajectory data. These observations can reveal where vehicles travel and how network conditions vary, but they are not a direct observation of demand and they do not automatically identify the behavioral mechanisms that produced the observed flows. Probe penetration varies across travelers, locations, times, and vendors; many trips are never observed; sensor counts are sparse; and the same measured pattern may be consistent with several combinations of demand and network cost. A useful diagnosis system must therefore combine data with structural models, quantify the difference between what is observed and what is inferred, and make limitations visible to decision makers.

Classical traffic assignment provides an interpretable structural baseline through user equilibrium (UE; Wardrop, 1952): no traveler can unilaterally improve a travel choice when all travelers are assigned. In practice, however, link performance functions, inverse demand relationships, route sets, and context effects are often specified before calibration. That practice can be adequate for a bottom-up conventional planning workflow, but it becomes restrictive when the goal is to learn from repeated, high-resolution observations. The project reframes calibration as inverse learning: rather than selecting a few parameters around a fixed model, it asks which equilibrium model is most consistent with observed trajectories and flows under explicit assumptions about fit, equilibrium consistency, and functional form.

The research also accounts for uncertainty and estimates demand at city scale. A deterministic equilibrium fitted to an average day may hide important cross-day variation. Likewise, scaling a small probe sample into a citywide trip table can reproduce neither unobserved origin–destination (OD) pairs nor the diversity needed for simulation. The four research products synthesized in this report address these gaps at complementary levels. One organizes the inverse-learning design space; one develops scalable context-dependent estimation; one extends the equilibrium object from a point to a distribution; and one generates and calibrates city-scale demand from probe trajectories and sparse traffic counts.

1.2 Research scope and contributions

The project addresses foundational methods for trajectory-based traffic network diagnosis through inverse equilibrium learning, context-dependent estimation, distributional analysis, and city-scale demand generation. This report synthesizes the theoretical developments and empirical results documented in the four project manuscripts. Ann Arbor studies evaluate context-dependent learning and count-calibrated demand, while controlled Sioux Falls and Chicago Sketch experiments evaluate distributional equilibrium and risk-sensitive decisions.

Table 1. Research components and supporting evidence.

Research component	Supporting evidence	Contribution
Planning-level inverse learning	Review, multiconvex method, Ann Arbor empirical study	Learn equilibrium relationships from repeated observations under explicit structural assumptions
Trajectory-based demand generation	Public benchmarks and Ann Arbor count validation	Demand estimation at city scale
Uncertainty-aware equilibrium	Synthetic Sioux Falls and Chicago Sketch experiments	Capture variability in traffic network equilibrium

1.3 Report organization

Section 2 relates the original project objectives to the documented research and describes a proposed integration architecture. Sections 3–6 present the inverse-learning framework, multiconvex method, distributional equilibrium model, and demand-generation pipeline. Section 7 separates the empirical and controlled experiments and identifies remaining validation needs. Sections 8–12 summarize findings, recommendations, outputs, lessons, and performance indicators.

2. Project Objectives and Integration Framework

2.1 Project objectives

The original proposal targeted short-term, decision-relevant network analysis at both planning and operational levels. It proposed trajectory processing, equilibrium-model calibration, automatic mesoscopic simulation calibration, case studies in Ann Arbor and Oakland County, and a prototype diagnosis system. The research documented here addresses four linked questions:

1. How can equilibrium models be constructed from partial trajectory and flow observations without fixing all behavioral functions in advance?
2. How can the inverse problem remain computationally manageable on large networks and across many observed days?
3. How can cross-day variability and latent behavioral regimes be represented without collapsing them into a single average equilibrium?
4. How can biased probe trajectories support feasible trip generation and OD estimation that improve agreement with independent traffic counts?

The four studies develop complementary capabilities within this scope. Their individual results do not establish completion of an integrated prototype, operational simulation calibration, or every proposed case study. Table 2 maps the original commitments to the evidence presented in this

report. The table distinguishes demonstrated research from proposed activities whose completion is not established by the studies.

Table 2. Original project tasks and documented accomplishments.

Original commitment	Documented accomplishment	Status and remaining evidence
Methods for planning analysis	Inverse-learning framework and multiconvex equilibrium method; controlled risk-sensitive pricing	Component methods demonstrated; combined diagnosis workflow not validated
Trajectory processing and event detection	Processed trajectories, active-link construction, and constrained trip generation	Model inputs and generation demonstrated; event-detection evaluation not documented
Operational simulation calibration	Demand prior, OD calibration, and trip resampling	Demand-side component demonstrated; Simulation of Urban MObility (SUMO) supply calibration and joint simulation validation not documented
Ann Arbor and Oakland County cases	Two distinct Ann Arbor applications; synthetic and public benchmarks	Ann Arbor evidence presented; Oakland County case completion not documented
Prototype and software delivery	Mathematical formulations and computational methods in four studies	Integrated prototype, cloud deployment, and public release are not established by the studies

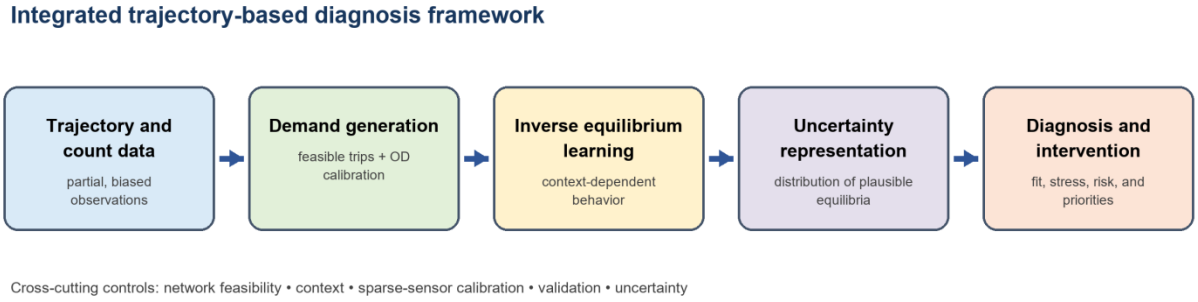


Figure 1. Proposed integration architecture linking trajectory observations, demand construction, equilibrium learning, uncertainty, and diagnosis. The component studies were evaluated separately.

2.2 Data-to-diagnosis logic

The left side of Figure 1 emphasizes that trajectory data are observations, not demand truth. Probe samples must be map matched, associated with time and context, and interpreted together with network topology and independent counts. The demand-generation component uses the network topology to constrain trip generation, so every generated route is connected without repeated links and conserves flow at intermediate nodes. Sparse counts then anchor the scale of the trip set. This division of labor uses trajectories for rich path structure and counts for aggregate magnitude.

The middle of the framework separates the forward and inverse problems. The forward problem calculates an equilibrium for a given demand, context, and model. The inverse problem observes a subset of flows or travel times and adjusts the model so that the observed state is close to an equilibrium prediction. That separation is important for diagnosis: poor fit can result from inadequate functions, sparse data, incorrect demand, missing context, or genuine departures from equilibrium. The report therefore presents fitting accuracy and equilibrium consistency as related but distinct criteria.

The right side extends diagnosis beyond point prediction. A learned distribution of equilibria can show whether a high-congestion outcome is typical, rare, or outside the range represented by training data. Risk-sensitive interventions such as conditional value-at-risk (CVaR) tolling use the tail of that distribution rather than only its mean. In this project, distributional results are demonstrated in synthetic settings where the latent ground truth is known. Empirical deployment will require repeated-day observations, explicit treatment of measurement error, and stronger validation of identifiability.

2.3 Distinct data regimes

- Scalable inverse learning uses map-matched 2022 evening-peak trajectories and link travel times on an 810-node Ann Arbor assignment network; the target is a context-dependent equilibrium model evaluated through empirical cross-validation.
- Demand generation benchmarks individual trip generation on public Porto and Beijing taxi trajectories and evaluates the full pipeline on a different 5,615-node Ann Arbor network with probe trajectories and hourly camera counts from 9 a.m. to 7 p.m.; the target is simulation-ready trip demand evaluated on calibration and held-out count links.
- Distributional UE uses controlled repeated-flow observations on Sioux Falls and Chicago Sketch; the target is a latent equilibrium distribution evaluated against known synthetic ground truth.

2.4 Component interfaces and compatibility requirements

Future integration requires explicit interfaces between the independently evaluated components. Network identifiers, link directions, time intervals, OD zones, units, and data provenance must agree before outputs are transferred. Table 3 summarizes the required interface information. These interfaces are implementation requirements, not evidence that a combined system has been deployed.

The demand-to-equilibrium interface is especially important. Calibration holds generated route proportions fixed, whereas a subsequent equilibrium model can change those proportions when costs or demand change. An integrated implementation would need a consistent assignment model

or an explicitly evaluated alternating procedure. The current component results do not validate such an iteration.

Observation and context definitions must also match the intended application. Historical explanation may use realized conditions, whereas forecasting can use only information available at prediction time. Independent evaluation data must be excluded from all fitted or count-guided stages, not just from the final optimization objective.

Table 3. Required interfaces for future integration of the component methods.

Interface	Required content	Compatibility check	Failure if omitted
Trajectory to demand	Network hash, link sequence, time, OD zones, quality flags	Topology and time-zone match	Disconnected or temporally shifted trips
Demand to assignment	Time-indexed OD totals, route support, route-choice probabilities	Units, period, and feasible routes	Counts cannot be interpreted consistently
Assignment to inverse learning	Feasible state, observation operator, context, demand	Same network and aggregation	Behavior absorbs representation error
Inverse learning to distribution	Cost map, admissible parameter domain, solver tolerances	Stable forward map	Latent model generates ill-conditioned states
Model to diagnosis	Residuals, uncertainty, sensitivity, provenance	Evidence trace to raw observations	Alerts appear precise but are not auditable

2.5 Levels of inference and diagnostic claims

The framework separates four levels of inference. The first is descriptive reconstruction: estimate a feasible network state that agrees with available trajectories and counts. The second is structural learning: infer a cost, potential, demand, or latent-regime model that explains repeated states. The third is diagnostic comparison: identify where a new observation is inconsistent with the learned conditional distribution. The fourth is causal or policy interpretation: attribute the inconsistency to a mechanism or evaluate an intervention. Evidence adequate for one level is not automatically adequate for the next.

A reconstruction can be accurate on observed links while remaining underdetermined elsewhere. A structurally regularized estimator can select one complete state, but the unobserved part should be treated as model based. Likewise, a learned cost function may support reliable interpolation without uniquely identifying traveler preferences. The report therefore distinguishes identification of predictive quantities from behavioral identification. The former asks whether the available data determine the quantities needed for a forecast. Behavioral identification asks whether alternative behavioral mechanisms have been ruled out. The latter is a stronger claim.

A diagnostic score should be interpreted as a conditional discrepancy. If observed flow lies outside a calibrated prediction interval, the observation warrants investigation, but a single exceedance does not establish a change in demand, supply, context, or the measurement process. Attribution requires additional tests. Demand can be re-estimated while supply remains fixed; capacity or cost parameters can be perturbed while demand remains fixed; sensor health can be checked

independently; and spatial patterns can be compared with known bottleneck or incident signatures. A diagnosis becomes stronger as alternative explanations are rejected with independent evidence.

Policy evaluation adds another requirement: invariance under intervention. A model learned from historical equilibrium observations may reproduce the observed regime yet fail when tolls or capacities move outside the training support. Shape constraints, monotonicity, and structural network conservation improve extrapolation, but they do not prove invariance. Counterfactual use should report the range of interventions, the distance from observed conditions, and the sensitivity of conclusions to cost specification and demand response.

The studies establish methods and planning-level evidence for reconstructing, learning, and quantifying network states. Their diagnostic interpretation follows the distinction between descriptive reconstruction, structural learning, conditional discrepancy, and policy analysis developed in this section.

2.6 Cross-component assumptions and failure propagation

Errors propagate through the framework in structured ways. Map-matching ambiguity changes route frequencies and therefore the generated demand prior. Demand-prior error changes the feasible state used by inverse learning. A misspecified cost map changes both the deterministic equilibrium and the distribution obtained by pushing latent regimes through the solver. An uncertainty model trained on residuals from a biased deterministic layer can become well calibrated to the wrong center. End-to-end evaluation must therefore inspect intermediate artifacts instead of judging only the final count error.

One useful practice is perturbation analysis at each interface. Alternative map matches can be sampled for ambiguous traces. OD priors can be bootstrapped by day or probe. Assignment matrices can be recomputed under plausible route-choice parameters. Equilibrium parameters can be refit from multiple initializations and under several basis resolutions. Latent distributions can be evaluated with several random seeds and held-out contexts. The resulting variation does not automatically combine into a formal posterior, but it reveals whether conclusions depend on one fragile preprocessing choice.

Failure containment is equally important. A trip generator should make topology violations impossible, as the constrained decoding of Section 6.3 does, or reject such routes rather than pass them to the simulator. An equilibrium solver should record nonconvergence rather than return its last iterate as though it were a solution. A distributional sampler should identify regimes that produce infeasible or numerically unstable states. A diagnostic interface should suppress or qualify results when upstream coverage is insufficient. These controls turn model assumptions into observable system states that can be reviewed.

The integrated architecture is therefore modular but not independent. Each module has a local objective and a local validation test, while the full system requires compatibility and traceability across modules. The advantage of modularity is that a failure can be localized and a component

can be replaced. The risk is that apparently reasonable outputs are chained without checking that their assumptions align. The mathematical definitions and validation requirements below make these compatibility conditions explicit.

3. Inverse Learning of Network Equilibrium

3.1 Forward equilibrium as a structural layer

A network equilibrium model maps demand and context into flows. Let $V(q)$ denote the set of feasible network flows for demand q, and let $c(v, x; \theta)$ denote generalized costs at flow v, context x, and model parameter θ. A variational-inequality form of user equilibrium requires that moving away from the equilibrium in any feasible direction cannot reduce the aggregate first-order cost. This representation is broad enough to support fixed-demand route choice and, with virtual links or elastic-demand constructions, joint flow and demand relationships (Liu & Yin, 2026).

$$v^* \in V(q): c(v^*, x; \theta)^T (v - v^*) \geq 0, \ \forall v \in V(q) \tag{1}$$

At equilibrium, no feasible redistribution of flow provides a collective first-order improvement under the modeled generalized costs.

The equilibrium layer provides interpretability and physical feasibility that a direct black-box predictor may not preserve. Its costs can be decomposed into travel time, delay, comfort, toll, or other components, and conservation is enforced by the feasible set. At the same time, equilibrium should be treated as a modeling hypothesis rather than an unquestioned fact. Observed trajectories may include perception error, day-to-day adjustment, incidents, and sampling noise. The inverse formulation therefore allows controlled discrepancies between observations and equilibrium predictions.

3.2 Unified inverse-learning problem

Inverse learning estimates model structure or parameters from observed network states. In the terminology of Liu and Yin (2026), equilibrium-prioritizing formulations impose equilibrium consistency as a constraint and minimize fitting error, as in Equation 2. Data-prioritizing formulations instead constrain the data mismatch and minimize an equilibrium gap. Priority therefore refers to the requirement protected by a constraint. A weighted sum of fitting error and equilibrium gap is a compromise formulation whose trade-off depends on the penalty weight.

$$\min_{\theta} \sum_{m=1}^{M} L(P_\Omega v^*(x_m; \theta), y_m) + \lambda R(\theta) \tag{2}$$

Here m indexes M observations, x_m is context, y_m is the observed state, P_Ω selects observed components, L measures fitting error, and $\lambda R(\theta)$ regularizes model complexity. The equilibrium $v^*(x_m; \theta)$ is obtained from the forward model; demand is included among its inputs.

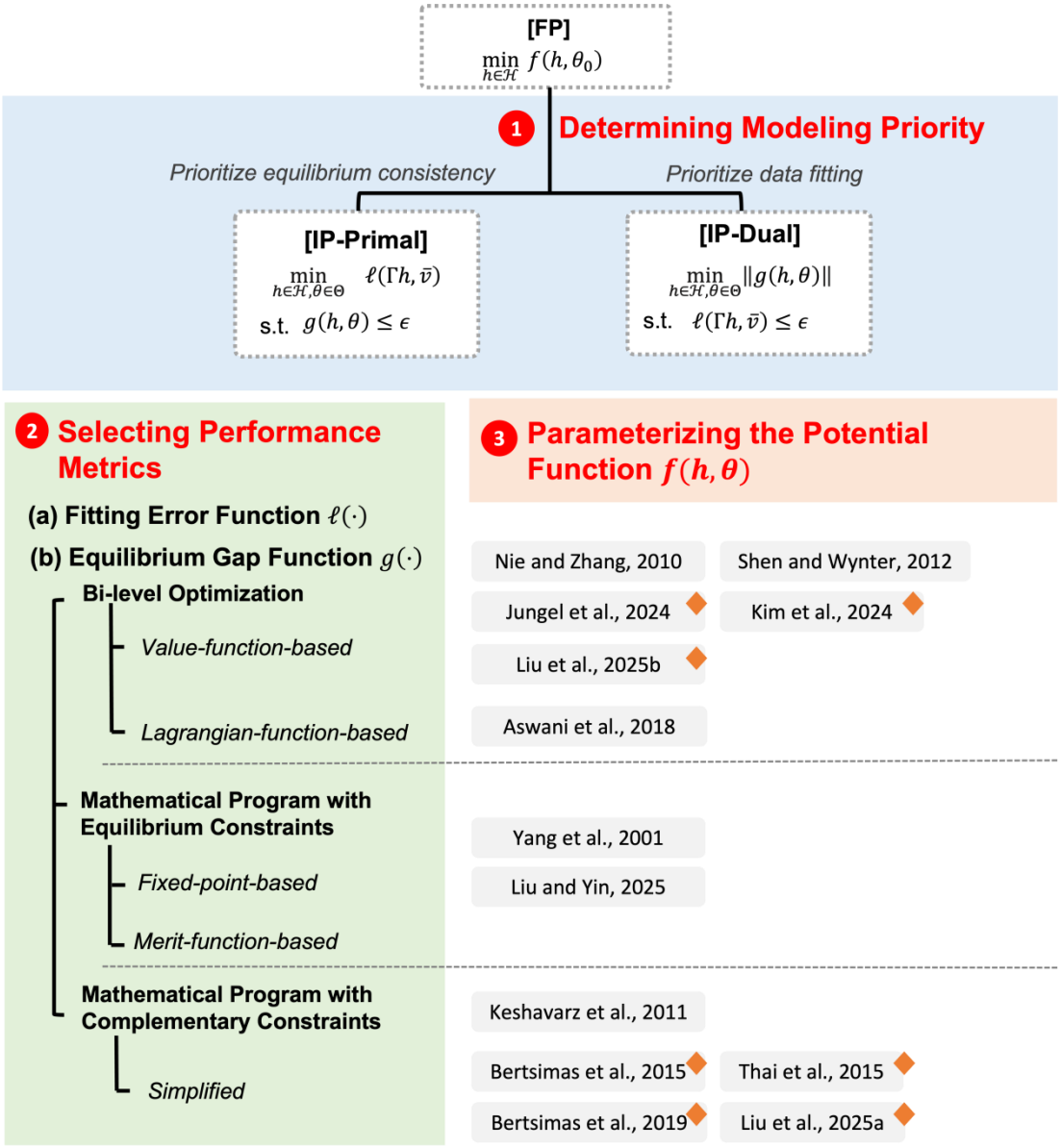


Figure 2. Three principal modeling choices in inverse equilibrium learning: the priority between fit and equilibrium consistency, the performance metric or gap formulation, and the parameterization of the potential function. Source: Liu and Yin (2026).

3.3 Equilibrium-gap choices

The review manuscript shows that several mathematical programs can encode equilibrium consistency. Bilevel models embed a forward equilibrium problem within the inverse problem. Mathematical programs with equilibrium constraints use fixed-point or merit-function conditions. Complementarity formulations express route-use and minimum-cost relationships directly. For models admitting a convex potential f, value-function formulations replace the lower-level problem with the suboptimality gap in Equation 3. The candidate v must belong to the same

feasible set V as the minimizing flow u. These forms are related, but their statistical interpretation and computational burden differ sharply under partial observations.

$$g(v, x; \theta) = f(v, x; \theta) - min_{u \in V} f(u, x; \theta) \geq 0 \quad (3)$$

The gap measures how far an observed or predicted flow is from minimizing the equilibrium potential; zero indicates exact optimality under the modeled potential.

Table 4. Inverse-learning design choices and their consequences for network diagnosis.

Design choice	Question	Implication for diagnosis
Fit versus equilibrium priority	Is mismatch treated as an objective or a constraint?	Changes whether unusual observations are absorbed by the model or flagged as disequilibrium
Observed quantity	Path flow, link flow, travel time, or demand?	Determines identifiability and which discrepancies can be localized
Gap function	Value, Lagrangian, fixed-point, merit, or complementarity?	Controls tractability and the meaning of an equilibrium-consistency score
Functional form	Parametric, convex basis, or neural representation?	Trades interpretability and guarantees against flexibility
Context	Which exogenous features shift cost or demand?	Separates systematic conditions from unexplained residuals
Regularization	What prior structure is defensible?	Prevents sparse observations from supporting unstable or nonphysical functions

3.4 Observation, identifiability, and validation

Path-flow observations can simplify inverse learning because they expose route choices more directly, while link-flow observations are typically more abundant but aggregate many OD and path combinations. Partial observations amplify non-identifiability: several cost functions can produce similar observed flows, and unobserved links may carry the differences. This is why the report emphasizes held-out sensors, repeated contexts, and explicit regularization rather than calibration fit alone.

Validation must match the intended use. If the model is intended to predict traffic under observed contexts, held-out days and links test generalization. If it is intended to recover behavioral parameters, synthetic experiments with known parameters are informative but do not establish empirical validity. If it will support interventions, policies should be evaluated under scenarios not used for fitting and, when uncertainty is material, against tail-risk measures as well as averages. These distinctions organize the evidence in later sections.

3.5 Mathematical foundations and notation

3.5.1 Network representation and feasibility

This section states the mathematical objects that are used implicitly throughout the report. A directed transportation network is represented by a node set and a link set. An origin-destination pair identifies a travel market, and each market has a set of admissible paths. Path flow is the most behaviorally direct decision variable because Wardrop's principle compares path costs. Link flow

is usually the observed and computationally convenient variable because travel time and count data are attached to links. The link-path incidence matrix maps path flow to link flow, while the OD-path incidence matrix enforces conservation of demand.

Let h collect nonnegative path flows, v collect link flows, q collect OD demands, Λ denote the link-path incidence matrix, and Δ denote the OD-path incidence matrix. The two equalities in Equation 4 perform different roles: $v = \Lambda h$ aggregates route choices into link volumes; $\Delta h = q$ ensures that the total flow assigned to the paths of each OD pair equals its demand. These relationships prevent a statistical learner from predicting internally inconsistent traffic states.

$$v = \Lambda h, \Delta h = q, h \geq 0 \tag{4}$$

Network feasibility couples every predicted link volume to a nonnegative set of routes that conserves each OD demand.

The feasible path-flow set in Equation 5 is a product of simplices when each OD demand is fixed and the path set is finite. It is closed and convex. Those properties are important because the variational-inequality existence results and projection algorithms used later require a nonempty closed convex feasible set. In practice, path enumeration is often avoided by using link-based algorithms or a restricted path set that is expanded during assignment. The mathematical definition nevertheless clarifies what must be preserved by any implementation.

$$H(q) = \{h: \Delta h = q, h \geq 0\} \tag{5}$$

A feasible traffic state assigns every unit of OD demand to a connected path and does not create or lose trips.

3.5.2 Wardrop equilibrium as a variational inequality

For a fixed demand vector, the generalized path-cost mapping $C(h)$ assigns a perceived cost to every path at a candidate flow. The mapping may include congested travel time, toll, distance, reliability, or other attributes. A Wardrop equilibrium is a feasible flow at which no infinitesimal traveler can reduce generalized cost by switching routes. Equation 6 expresses this condition without selecting a particular route as a reference. The inner product compares equilibrium cost with every feasible reallocation. If a direction lowered total first-order cost, the inequality would be violated.

The variational-inequality form is preferable to a purely verbal definition because it remains valid when costs are asymmetric and no scalar potential exists. It also exposes the assumptions needed for computation. Continuity of C on a compact feasible set is sufficient for existence. Monotonicity supports variational-inequality algorithms; convexity of a particular gap function depends on its construction. Strong monotonicity gives uniqueness of the equilibrium flow and quantitative sensitivity bounds. These are properties of the modeled cost map; empirical traffic need not satisfy them exactly.

$$\langle C(h^*), h - h^* \rangle \geq 0, \ \forall h \in H(q) \tag{6}$$

No feasible infinitesimal redistribution of path flow can lower generalized cost at the modeled equilibrium.

The same condition can be written as path-level complementarity. For each OD pair r, π_r denotes the minimum generalized cost. A path with positive flow must attain that minimum; an unused path cannot be cheaper. Equation 7 is useful for interpretation and for mathematical programs with complementarity constraints, but it introduces one condition per path and can be large when route sets are extensive.

$$h_p^r > 0 \Rightarrow C_p^r(h^*) = \pi_r; h_p^r = 0 \Rightarrow C_p^r(h^*) \geq \pi_r \tag{7}$$

Travelers use only minimum-cost routes under the modeled generalized cost, although multiple equal-cost routes may carry flow.

3.5.3 Beckmann potential and optimization equivalence

When each link cost depends only on its own link flow and is integrable, the equilibrium can be represented by the Beckmann potential (Beckmann et al., 1956). Equation 8 integrates each link cost from zero to its current flow and sums across links. Differentiating the potential with respect to link flow returns the link-cost vector. The first-order condition of minimizing this potential over the feasible link-flow set is therefore the variational inequality in the main report.

This equivalence is central to the project's inverse-learning formulations. It turns a behavioral equilibrium condition into an optimization layer whose objective can be parameterized, regularized, and differentiated. It also explains why convexity matters: if every link cost is nondecreasing, the integral potential is convex. Strictly increasing link costs make the equilibrium link-flow vector unique, even though the path-flow decomposition may remain nonunique.

$$\Phi(v) = \sum\nolimits_{a \in \mathcal{A}} \int_0^{v_a} t_a(s) ds \tag{8}$$

The potential accumulates marginal link costs; its gradient is the vector of link costs.

$$v^* \in \underset{v \in V(q)}{arg\ min} \Phi(v) \tag{9}$$

Under separable monotone link costs, solving one convex program produces the Wardrop equilibrium.

3.5.4 Existence uniqueness and sensitivity

A learned model should be evaluated not only by prediction error but also by whether the forward equilibrium is well defined. Existence follows under mild continuity and compactness assumptions. Uniqueness is more demanding. If the cost map is merely monotone, multiple equilibria can exist along flat cost directions. If it is strongly monotone with modulus μ, Equation 10 provides curvature that rules out distinct solutions and, with suitable regularity, supports sensitivity bounds for parameter changes on a fixed feasible set. Sensitivity to demand also requires control of how that set changes.

Strong monotonicity is a modeling restriction rather than an empirical fact. In a nonparametric potential, it can be encouraged through basis selection or a small strongly convex regularizer. The benefit is numerical stability and a single-valued equilibrium map. The cost is reduced flexibility in representing flat or nonmonotone phenomena. The project therefore uses convex structure as a disciplined approximation for planning-level inference, not as a claim that every observed traffic process is globally monotone.

$$(c(u) - c(v))^T(u - v) \geq \mu\|u - v\|_2^2 \tag{10}$$

Moving flows apart must increase the aligned difference in cost by at least a quadratic amount, which stabilizes and ensures uniqueness of the equilibrium.

When the feasible set is fixed and the cost map varies smoothly with parameter θ, strong monotonicity yields a local bound of the form shown in Equation 11. The constant L grows when the equilibrium becomes ill conditioned. This relationship explains why small data or parameter perturbations can produce large flow changes near capacity transitions or when several routes have nearly equal costs. Sensitivity diagnostics should therefore accompany counterfactual analysis.

$$\|v^*(\theta_1) - v^*(\theta_2)\| \leq L\|\theta_1 - \theta_2\| \tag{11}$$

A stable equilibrium map does not amplify small parameter perturbations without bound; a large estimated L signals fragile predictions.

3.5.5 Observation operators and incomplete data

Empirical datasets rarely observe the full equilibrium state. A link-count system measures a subset of link volumes; probe trajectories sample paths with nonuniform penetration; travel-time feeds may aggregate over time; and map matching adds classification uncertainty. Equation 12 separates the latent equilibrium from the measurement process. H_m is an observation operator that selects, aggregates, or transforms the modeled state for sample m, and ϵ_m collects measurement error and short-run variation not represented by the model.

Writing the observation process explicitly prevents two common errors. First, a missing link is not a zero-flow link. Second, probe-path frequencies are not automatically population path shares. The observation operator may incorporate availability masks, aggregation intervals, or penetration weights, while the loss function can reflect heterogeneous reliability. This separation also supports held-out validation because sensors can be omitted from H during calibration and used only for evaluation.

$$y_m = H_m v_m^* + \varepsilon_m \tag{12}$$

The model predicts a complete feasible state, but the comparison with data is made only through the measurements actually available.

Table 5. Core mathematical objects used in the detailed report.

Symbol	Meaning	Diagnostic role
h, v, q	Path flow, link flow, and OD demand	Separate route choice, network loading, and trip generation
Λ, Δ	Link-path and OD-path incidence matrices	Enforce topology and conservation
C or c	Path- or link-cost mapping	Encodes behavior and network performance
H	Observation operator	Represents sparse sensors, aggregation, and probe sampling
Φ or f	Equilibrium potential	Supports convex optimization and inverse learning
g	Equilibrium gap	Measures structural inconsistency of a candidate flow
x, θ, γ	Observed context, deterministic parameters, and latent regime	Separate explained variation from unobserved cross-day variation

3.6 Detailed inverse-learning formulations

3.6.1 Joint fitting and equilibrium consistency

A joint fitting-and-gap formulation treats sample-specific network states as decision variables, penalizing both disagreement with measurements and departure from equilibrium. Equation 13 illustrates this compromise; it is not assigned either priority label. The gap weight sets the relative importance of structural consistency, while the regularizer controls model complexity. Throughout Equations 13–14, θ belongs to an admissible parameter set Θ and each v_m belongs to its network-feasible set V_m. These formulations describe the design space; not all were implemented in the project.

The joint-state formulation is useful when observations are incomplete because unobserved links can be inferred subject to conservation and equilibrium structure. It also allows the modeled state to deviate from an exact equilibrium when data are noisy. Its weakness is tuning: different values of ρ can yield similar measured-link fit but different counterfactual behavior. A transparent analysis should therefore report a regularization path rather than only the selected value.

$$\min_{\theta,\{v_m\}} \sum_m [\ell(H_m v_m, y_m) + \rho g(v_m, x_m; \theta)] + \lambda R(\theta) \tag{13}$$

The learned states must explain measurements while remaining close to equilibrium under a model that is not unnecessarily complex.

A data-prioritizing alternative constrains the fitting error and minimizes the equilibrium gap, as in Equation 14. The error budget should reflect measurement uncertainty and the intended fit requirement. This is the dual perspective in Liu and Yin (2026): among models meeting that data requirement, seek the most equilibrium-consistent explanation.

$$\min_{\theta,\{v_m\}} \sum_m g(v_m, x_m; \theta) + \lambda R(\theta),\ \sum_m \ell(H_m v_m, y_m) \le \varepsilon \tag{14}$$

Among explanations that fit the data adequately, prefer the one that is most consistent with equilibrium structure.

3.6.2 Complementarity and equilibrium constrained programs

Path optimality can be imposed directly with complementarity. Equation 15 says that a path flow and its excess cost cannot both be positive. This creates a mathematical program with equilibrium constraints when θ is also a decision variable. Complementarity is exact and interpretable, but standard constraint qualifications fail at many feasible points. Large route sets also produce many complementarity pairs, making generic nonlinear solvers difficult to scale.

Smoothing, penalty, Scholtes relaxation, and Fischer-Burmeister residuals are common computational devices. Each changes the finite-iteration meaning of equilibrium. A smoothed solution should be reported with the residual tolerance and continuation schedule, because two solvers can return similar objective values while satisfying complementarity to different degrees. The value-function gap used in the project avoids explicit path complementarity and better exposes convex subproblems.

$$0 \leq h_p^r \perp C_p^r(h) - \pi_r \geq 0 \tag{15}$$

A path carries flow only when its generalized cost equals the OD minimum; a more expensive path must be unused.

3.6.3 Local identifiability under partial observations

Good fit does not imply that the learned parameters are uniquely determined. Local identifiability depends on how equilibrium predictions change with θ at the observed links and contexts. Equation 16 stacks those sensitivities into a Jacobian. If distinct parameter perturbations produce indistinguishable observed responses, those perturbations cannot be resolved from the available measurements. A rank-deficient Jacobian indicates a lack of first-order identification, although higher-order identification may still be possible.

Context variation can improve identification because repeated days excite different combinations of demand, capacity, and behavior. Network coverage also matters: sensors concentrated on one corridor may leave alternative-route parameters weakly identified. Regularization can select one solution among many but does not create empirical information. Accordingly, parameter confidence should be interpreted as conditional on the chosen functional class and priors.

$$J(\theta) = \left[\frac{\partial(H_m v_m^*(\theta))}{\partial \theta}\right]_{m=1}^{M} \tag{16}$$

A parameter is learnable only if changing it creates a distinguishable pattern in the measurements and contexts.

For a continuously differentiable observation map at an interior parameter point, full column rank in Equation 17 is a sufficient local identification condition. In high-dimensional nonparametric models full rank may be unrealistic or numerically unstable. Singular values provide a graded diagnostic: very small values indicate directions that are technically identifiable but highly sensitive to noise. Profile objectives, bootstrap refits, and comparisons across initializations offer practical complements to a formal rank check.

$$rank(J(\theta_0)) = dim(\theta) \tag{17}$$

Full column rank excludes nonzero parameter perturbations that leave all observed predictions unchanged to first order.

3.6.4 Regularization and model selection

Regularization has both statistical and behavioral interpretations. A ridge term stabilizes correlated basis coefficients and favors gradual responses. An L_1 term can remove weak context interactions and produce a smaller diagnostic model. Equation 18 combines the two. Group penalties can select or discard whole families of basis functions, while shape constraints preserve monotonicity or convexity. The regularizer should be specified before evaluating the test set.

Validation should respect the deployment question. Randomly holding out link-time pairs tests interpolation when nearby days and links remain in training. Holding out entire days tests temporal generalization. Holding out corridors tests spatial transfer. Holding out a context class, such as severe weather, is a stronger out-of-distribution test but may have too few samples for stable tuning. Reporting more than one split helps distinguish statistical fit from structural transfer.

$$R(\theta) = \lambda_1 \|\theta\|_1 + \lambda_2 \|\theta\|_2^2 \tag{18}$$

The model can suppress unsupported effects while stabilizing the coefficients that remain.

3.6.5 Differentiating through equilibrium

Some inverse-learning formulations differentiate equilibrium outcomes with respect to parameters. If a smooth residual $F(v, \theta) = 0$ characterizes the equilibrium and its state Jacobian is nonsingular, the implicit-function theorem gives Equation 19. This is one computational option, not a requirement of inverse learning: the multiconvex method in Section 4 avoids implicit-gradient computation.

Unrolling instead differentiates a finite sequence of solver iterations and stores intermediate states. The distributional method in Section 5 uses unrolled mirror descent. Its memory and truncation costs differ from those of the multiconvex method. Both approaches require validation of solver accuracy appropriate to their formulation.

$$\frac{dv^*}{d\theta} = -\left(\frac{\partial F}{\partial v}\right)^{-1} \frac{\partial F}{\partial \theta} \tag{19}$$

Sensitivity is obtained by differentiating the equilibrium condition rather than perturbing and resolving the model one parameter at a time.

Table 6. Comparison of inverse-learning formulations and their practical consequences.

Formulation	Strength	Primary risk	Recommended diagnostic
Bilevel equilibrium layer	Direct predictive interpretation	Expensive repeated solves	Forward residual and held-out fit
Joint fit plus gap	Handles noisy partial states	Penalty-weight sensitivity	Regularization path and gap scale
Fit-constrained gap	Error budget is explicit	Requires defensible tolerance	Sensitivity to epsilon
Complementarity MPEC	Exact path-use logic	Nonconvexity and scale	Complementarity residual
Differentiable solver	End-to-end scalable learning	Gradient error or memory burden	Finite-difference or adjoint checks

4. Scalable Context-Dependent and Nonparametric Learning

4.1 Context-dependent equilibrium

The context-dependent formulation allows link costs and inverse demand to vary with observed features while retaining equilibrium structure. In the Ann Arbor application, road and OD attributes are combined with snow and weekday variables. Learned coefficients summarize conditional associations within this model; their signs are not independent causal estimates of weather or calendar effects (Liu, Yin, Lin, and Wang, 2026).

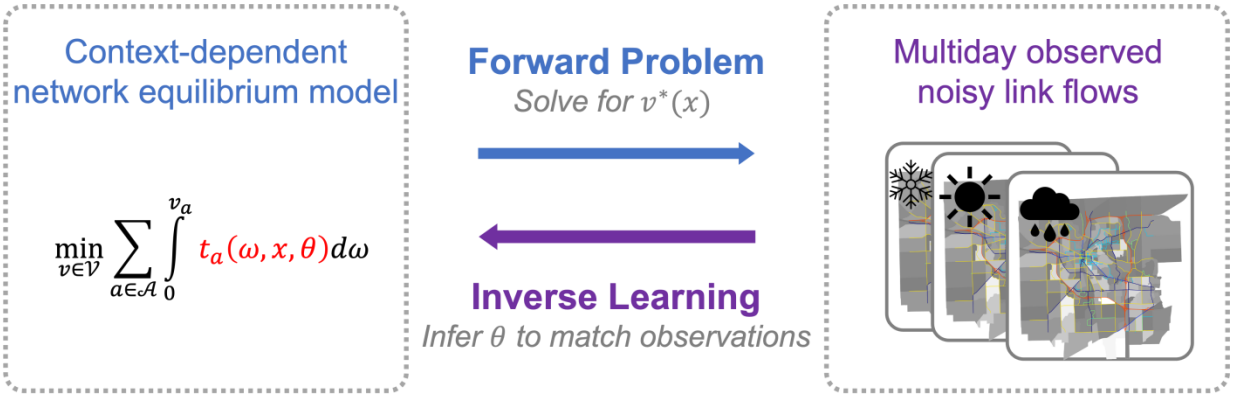


Figure 3. Forward and inverse views of context-dependent equilibrium learning from multiday noisy link-flow observations. Source: Liu, Yin, Lin, and Wang (2026).

4.2 Convex-basis potential

The potential is a nonnegative linear combination of differentiable convex basis functions. For each link, observed nonnegative features enter linearly through shared nonnegative coefficients.

Equation 20 gives the source formulation: the basis vector collects link-specific convex functions, and the feature matrix contains the link and context attributes. Convexity in flows and linearity in the estimated coefficients are both essential to the multiconvex inverse problem.

$$f(v, x; \theta) = \sum_{s=1}^{S} \Phi_s{}^T(v) x \theta_s\,,\ x \geq 0, \theta \geq 0 \tag{20}$$

The unknown equilibrium potential is learned as a context-dependent, nonnegative mixture of convex building blocks, preserving convexity while allowing functional flexibility.

Virtual links can be used to represent inverse demand within the same network potential. Physical-link terms describe congestion or generalized travel cost; virtual-link terms encode the willingness to travel for OD pairs. Joint learning then recovers a context-dependent model in which both route costs and demand responses can change with snow, day type, and OD features. The approach is nonparametric in the sense that it learns the combination of basis functions rather than only fitting coefficients in one preselected link formula.

4.3 Multiconvex formulation and parallel algorithm

The value-function gap is convex in flows with parameters fixed and convex in parameters with flows fixed under the source parameterization. The relaxed inverse objective adds an observation loss convex in the modeled flows. Alternating updates therefore use convex subproblems, although the complete problem is not jointly convex. Flow updates decompose across observations, and operator splitting handles nonnegativity and conservation constraints.

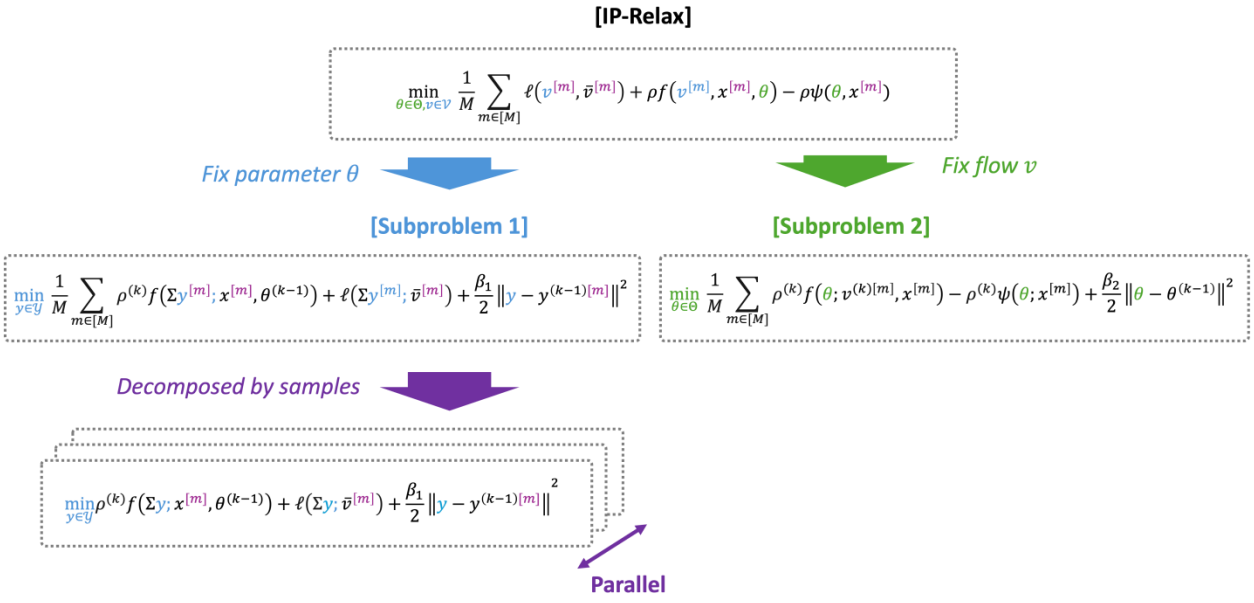


Figure 4. Multiconvex decomposition and parallel block coordinate descent for the scalable inverse-learning problem. Source: Liu, Yin, Lin, and Wang (2026).

The statistical consistency results concern nearly globally optimal solutions of the stated inverse formulation under compactness, regularity, observation, and identifiability assumptions.

Parameter recovery additionally requires normalization to remove scale ambiguity. These results do not imply that every stationary point produced by block coordinate descent is statistically consistent. Optimization convergence, predictive performance, and parameter identification must therefore be assessed separately (Liu, Yin, Lin, & Wang, 2026).

4.4 Evidence from synthetic and empirical tests

Table 7. Selected evidence for scalable context-dependent inverse learning.

Test	Condition	Reported result	Interpretation
Sioux Falls recovery	340 generated contexts; under 5% flow noise	Best settings produced link-flow MAPE near 4–6% when functions and observations varied	Supports controlled recovery under known data generation
Initialization	Both link performance and inverse demand unknown	Random/overestimate starts achieved about 5.7% link-flow MAPE; poor start was worse	Multiple starts and validation remain important
Ann Arbor benchmark	Context-independent annual-average model	Link-time MAPE 83.6%	A single average pattern poorly represents daily variation
Ann Arbor best setting	Linear link performance; quadratic inverse demand; Scenario 2	MAPE 34.3% (SD 1.9 percentage points); runtime 1.9 minutes	Improvement of the complete learned model; context contribution not isolated
Ann Arbor context coefficients	Snow, non-snow, weekday, OD features	Positive snow-related cost terms and positive non-snow/weekday demand effects	Conditional associations under parameter constraints and normalization

4.5 Multiconvex algorithm and scalable computation

4.5.1 Convex basis representation

For observation m, the potential can be written as in Equation 21, with a separate coefficient for each link and basis. The convex functions describe admissible flow-dependent shapes. Their nonnegative weights preserve forward convexity. Equation 22 then ties those weights to observed features through the same shared parameter vector used across days.

The empirical study evaluates linear or quadratic physical-link performance functions and several inverse-demand specifications, including entropy terms. The general basis representation permits richer function classes, but arbitrary splines or neural context mappings were not evaluated in the reported Ann Arbor experiment.

$$f_m(v) = \sum_{a,s} \alpha_{mas} \phi_{sa}(v_a), \ \alpha_{mas} \geq 0 \tag{21}$$

Each day uses the same admissible shapes but combines them differently, allowing context-sensitive behavior while preserving convexity.

4.5.2 Mapping observed context to admissible coefficients

For each link a and basis s, the coefficient is the inner product of the nonnegative feature vector and the corresponding nonnegative parameter vector, as shown in Equation 22. This linear dependence is what makes the value function concave in the estimated parameters and the suboptimality gap convex in that block.

A softplus or neural transformation could enforce nonnegative coefficients, but it would generally lose the parameter-block convexity used here. Such mappings are possible extensions and cannot inherit the multiconvex guarantee without a separate argument. Feature normalization must retain the nonnegative domain assumed by the source formulation.

$$\alpha_{mas} = {x_{ma}}^{T}\theta_s,\ x_{ma} \geq 0, \theta_s \geq 0 \tag{22}$$

Nonnegative features and coefficients preserve forward convexity, and their linear parameterization supplies the additional structure needed by the inverse problem.

4.5.3 Multiconvex objective structure

Equation 23 states the relaxed objective schematically, using the gap defined in Equation 3 and feasible sample flows. The shared parameter block and each sample-flow block are convex when the remaining variables are fixed. The source formulation uses OD-specific link flows and an aggregation operator to impose conservation before forming aggregate link flows.

Linearity in the coefficients is crucial: the forward value function is the pointwise minimum of affine functions of those coefficients and is therefore concave. Subtracting that value function makes the gap convex in the parameter block. This explains the multiconvexity; nonnegativity alone would not establish it.

$$\min_{\theta \in \Theta, v_m \in V_m} \sum_m [\ell(H_m v_m, y_m) + \rho g(v_m, x_m; \theta)] \tag{23}$$

The common model learns across days, while each day retains its own feasible state and observation mask.

4.5.4 Parallel block coordinate updates

The source algorithm alternates two updates. With parameters fixed, it solves one proximal convex flow problem per observation. A decoupled projection method separates nonnegativity from affine conservation and parallelizes over observations and OD pairs. With updated flows fixed, it updates the shared parameters using projected subgradients of the proximal parameter objective.

A parameter subgradient compares basis values at the fitted flows with those at a forward equilibrium, as in Equation 24. Thus the algorithm solves forward equilibria during parameter updates, but does not differentiate through those solutions. Here v_m^* minimizes $f(\cdot, x_m; \theta)$ over V_m; d_{ms} is a valid subgradient even when this minimizer is nonunique. A proximal term stabilizes the parameter step. The outer loop increases the gap penalty up to a prescribed maximum; the source paper specifies step sizes and inner iteration counts.

$$d_{ms} = {x_m}^T[\Phi_s(v_m) - \Phi_s(v_m^*)],\ d_{ms} \in \partial_{\theta_s} g_m \tag{24}$$

The parameter update compares the fitted network states with equilibria under the current model, using basis-value differences instead of implicit derivatives.

4.5.5 Descent and convergence interpretation

The paper invokes local convergence results for proximal block coordinate methods under their technical assumptions. For a fixed gap penalty and sufficiently accurate block minimization, the objective F_ρ of Equation 23 decreases as in Equation 25. When the penalty changes between outer iterations, objective values belong to different functions and need not form one monotone sequence.

A local or block-stationary solution is not a global optimum. Finite inner iterations, changing penalties, and initialization can affect the result. Report conservation residuals, equilibrium gaps, parameter-update residuals, and held-out performance alongside runtime. The statistical consistency results in Section 4.3 require stronger solution and identification conditions than local algorithmic convergence alone.

$$F_\rho(\theta^{k+1}, v^{k+1}) \leq F_\rho(\theta^k, v^k) \tag{25}$$

Descent applies to a fixed objective under suitable update conditions; it does not certify a global solution or statistical identification.

4.6 Implementation scope and reproducibility

The reported scalability comes from convex block structure, parallel updates over observations and OD pairs, and restricted active-link sets. The experiments do not evaluate multi-device training, mixed precision, or checkpointing; these remain possible implementation extensions.

A reproducible run should record basis and feature definitions, parameter normalization, active-link construction, hardware, penalty schedule, inner tolerances, stopping rules, and data splits. Runtime comparisons should use comparable accuracy criteria. The empirical results and their evaluation conditions are summarized in Section 7.1.

5. Distributional User Equilibrium and Risk

5.1 From one equilibrium to a distribution

Repeated network observations vary for reasons that are not exhausted by measurement error. Demand, preference weights, perceived costs, incidents, and unrecorded conditions may change across days. A deterministic inverse model compresses that variation into one representative parameter vector and one predicted equilibrium. Standard predictive-uncertainty methods can place uncertainty around parameters or outputs, but they do not necessarily model the latent generative mechanism that maps day-level regimes into equilibria.

Distributional user equilibrium (Dist-UE) changes the inferential object. A latent variable z generates equilibrium-relevant cost coefficients γ conditional on features; an equilibrium solver maps each coefficient draw into a feasible flow. Repeating the draw produces a distribution of equilibrium flows. The approach is not a claim that travelers coordinate on a random equilibrium in real time. It is a modeling hypothesis for repeated observations: cross-day outcomes can be represented by a distribution over latent regimes, each of which induces an equilibrium under the same network structure (Li et al., 2026).

$$z \sim p(z), \gamma = G_\vartheta(z, x), v^* = S(q, x, \gamma) \tag{26}$$

A latent day-level regime generates behavioral or cost coefficients, and the equilibrium solver converts each generated regime into a feasible network-flow outcome.

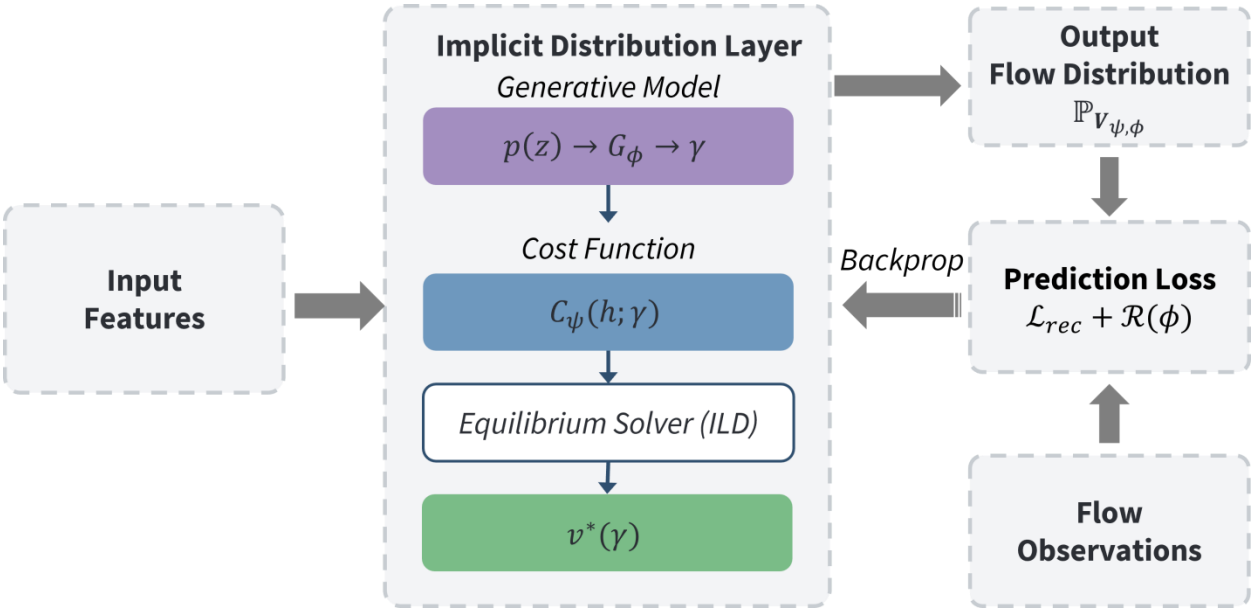


Figure 5. General distributional inverse-learning architecture linking features, a generative latent layer, a cost function, an equilibrium solver, observed flows, and an output flow distribution. Source: Li et al. (2026).

5.2 Conditional variational learning and differentiable equilibrium

The manuscript instantiates the framework with a conditional variational autoencoder (CVAE). An encoder maps observed flows and conditions into a latent posterior; a decoder generates cost-regime coefficients; and an unrolled mirror-descent solver enforces the equilibrium mapping. Training balances reconstruction of observed flow patterns with a divergence term that regularizes the latent posterior toward its prior. Differentiation through the unrolled solver allows the generative parameters to learn from equilibrium outcomes rather than from coefficients that are never observed directly.

$$\mathcal{L}_{\text{dist}} = \mathbb{E}_{q_\phi(z|v,x)}[-\log p_\psi(v \mid z, x)] + \beta \text{KL}(q_\phi \| p) \tag{27}$$

The first term rewards reproducing the observed flow distribution; the second regularizes the learned latent representation so it can generate plausible new regimes.

The equilibrium solver is an essential structural layer. Without it, a generative network could emit flow vectors that violate conservation or ignore network interactions. Unrolling, however, creates a substantial memory burden because intermediate iterations must be retained for backpropagation. This trade-off becomes visible on the Chicago Sketch network, where Monte Carlo scenario evaluation and unrolled optimization reduce the degree of parallelism available on a standard GPU.

5.3 Uncertainty calibration and risk-sensitive pricing

Uncertainty quality is evaluated through coverage and sharpness. Prediction interval coverage probability (PICP) measures the fraction of observations inside prediction intervals; prediction interval normalized average width (PINAW) measures their width relative to a reference range. A validation-set temperature factor rescales Monte Carlo covariance before testing. The benchmarks include Monte Carlo dropout (Gal & Ghahramani, 2016), deep ensembles (Lakshminarayanan et al., 2017), and Laplace approximations (Ritter et al., 2018; Daxberger et al., 2021). For decisions, conditional value-at-risk (CVaR) summarizes upper-tail loss.

$$min_{\tau \geq 0, \xi} \xi + \frac{1}{1-\alpha} \mathbb{E}_{\gamma}[(TT(v^*(\tau, \gamma)) - \xi)_+] \tag{28}$$

Here $\tau \geq 0$ is the toll vector, ξ is an auxiliary threshold, α is the CVaR confidence level, TT is total travel time, and $(\cdot)_+$ denotes the positive part. The objective minimizes upper-tail travel-time loss.

Table 8. Selected Distributional UE proof-of-concept results. Values retain the units and scenario definitions of the source manuscript.

Network/scenario	Mean-flow accuracy	Uncertainty result	Pricing result	Boundary
Sioux Falls, Gaussian σ=0.1	Dist-UE MAPE 1.22%; ensemble 1.20%	Dist-UE PICP 0.954	Dist-UE CVaR 1.530 vs. ensemble 1.710	Controlled synthetic data
Sioux Falls, Gaussian σ=0.5	Dist-UE reported lowest MAPE at 2.10%	Coverage remained near nominal	Lowest non-oracle tail risk across reported methods	Known latent distribution
Sioux Falls, discrete + observation noise	Dist-UE MAPE 1.92%	PICP 0.974	Dist-UE CVaR 1.237	Noise separated by construction
Chicago Sketch, Gaussian σ=0.1	Dist-UE MAPE 13.65%, lowest non-oracle	PICP 0.999; PINAW 2.484	CVaR 3.559; mean 1.689 (both $\times 10^8$ vehicle-hours)	Conservative intervals; tail improvement with higher mean total travel time

The Chicago experiment uses 2,950 links, 933 nodes, and 93,135 OD pairs. Dist-UE achieved MAPE of 13.65 percent, but its nominal 95 percent intervals had coverage of 0.999 and normalized width of 2.484. Coverage was conservative, and intervals were wider than the Laplace and ensemble intervals. Training took 15,042.4 seconds (about 4.18 hours), and robust pricing took 33,166.4 seconds (about 9.21 hours). This is a large-network research demonstration, not evidence of real-time operation. Section 7.3 reports the pricing trade-off.

5.4 Distributional equilibrium theory and risk-sensitive decisions

5.4.1 Distribution as a pushforward of latent regimes

Distributional user equilibrium retains a deterministic equilibrium conditional on a latent regime and makes the repeated-day outcome random through variation in that regime. Let Γ_x be a context-conditioned distribution of effective generalized-cost coefficients and let S denote the equilibrium solution map. Equation 29 defines the conditional flow distribution as the pushforward of Γ_x through S. This is a distribution over aggregate equilibrium states, not a classical stochastic user equilibrium in which individual perception errors average into one deterministic aggregate assignment.

The latent regime represents common day-level variation in effective generalized-cost weights. It is a reduced-form representation, not a separately identified distribution of capacity, incidents, demand, or individual traveler preferences. In the controlled experiments, demand and observation noise are specified by the data-generating design. Their empirical separation would require additional observations and assumptions.

$$P_{V^*|x} = S(q, x, \cdot)_{\#}\Gamma_x \tag{29}$$

Drawing a latent day type and solving its deterministic equilibrium generates one draw from the distribution of plausible network states.

5.4.2 Nonparametric latent generation

A simple reference variable z can be transformed by a learned generator into a flexible regime distribution. Equation 30 uses a standard Gaussian reference, but the resulting distribution of γ need not be Gaussian. Context can enter the generator so that observed weather or calendar conditions shift the distribution while residual uncertainty remains latent.

The decoder must generate coefficients in the admissible domain specified for the forward model. The resulting flow distribution is evaluated through its accuracy, coverage, width, and pricing consequences. Additional restrictions or alternative latent transformations would change the model and require their own evaluation.

$$\gamma = G_\vartheta(z, x), z \sim \mathcal{N}(0, I) \tag{30}$$

A simple source of randomness becomes a flexible distribution of day-level effective cost conditions.

5.4.3 Variational learning objective

The conditional variational objective in Equation 31 balances reconstruction and regularization. The first term rewards latent draws that reproduce an observed flow after passing through the decoder and equilibrium layer. The Kullback–Leibler term keeps the approximate posterior near the reference distribution and discourages the encoder from assigning an unrelated latent code to every observation. The coefficient β controls the strength of that trade-off.

In an equilibrium model, reconstruction is not merely a neural decoder output. A regime draw must produce a feasible equilibrium, so gradients pass through the forward solver. The reconstruction distribution should also distinguish observation noise from latent state variation. If all residual error is assigned to γ, prediction intervals can be overly wide and the latent distribution may become physically implausible. If all error is assigned to measurement noise, the inferred regime distribution can collapse toward a point.

$$\mathcal{L} = \mathbb{E}_q[-\log p(v \mid z, x)] + \beta \mathrm{KL}(q(z \mid v, x) \| p(z)) \tag{31}$$

The model must reconstruct repeated flows while keeping its latent representation regular enough to generate new plausible states.

5.4.4 Mirror descent equilibrium solver

The Dist-UE study uses unrolled mirror descent as a differentiable equilibrium solver. Multiplicative updates are natural for nonnegative path flows and can be followed by an OD-wise normalization or KL projection that restores demand conservation. Equation 32 presents the conceptual update. Here η_k is the step size, $\odot$ denotes elementwise multiplication, and Π^{KL} denotes Kullback–Leibler projection onto $H(q)$. Positive initial path flows retain support during multiplicative updates.

Step size controls both convergence and gradient behavior. A large step may oscillate among paths; a very small step requires many unrolled iterations and increases memory. Solver truncation means the learned model is trained against an approximate equilibrium. Training and evaluation should therefore report the final gap as well as predictive loss, and sensitivity tests should increase iteration count until conclusions stabilize.

$$h^{k+1} = \Pi^{KL}_{H(q)}\left(h^k \odot exp\left(-\eta_k C(h^k; \gamma)\right)\right) \tag{32}$$

Higher-cost paths lose relative mass, lower-cost paths gain mass, and projection preserves nonnegativity and OD totals.

5.4.5 Calibration and uncertainty decomposition

A distributional predictor must be evaluated by more than mean error. In Equation 33, $[l_i, u_i]$ is a nominal $(1 - \alpha)$ interval for observation v_i, N is the number of evaluated observations, and $1\{\cdot\}$ is an indicator. Here α denotes the miscoverage level, whereas in Equations 28 and 35–36 it denotes the CVaR confidence level. Coverage alone is insufficient because arbitrarily wide intervals cover everything. Average interval width, proper scores, quantile loss, and calibration curves should be reported together. Calibration should be assessed by link, flow regime, and context because aggregate coverage can hide systematic corridor-level failure.

$$\mathrm{PICP}_{1-\alpha} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}\{l_i \le v_i \le u_i\} \tag{33}$$

Coverage assesses calibration; interval width separately assesses sharpness. High coverage alone does not establish either nominal calibration or useful precision.

Equation 34 is the law of total variance, adapted to clarify uncertainty sources. The expectations and variances over γ are conditional on x. The first term represents variation conditional on a regime, which may include measurement or within-regime stochasticity. The second represents variation in the conditional mean across regimes. The decomposition is conceptual unless the observation model makes both terms explicit. It is nevertheless useful for preventing all variability from being labeled behavioral uncertainty.

$$Var(V \mid x) = \mathbb{E}[Var(V \mid \gamma, x)] + Var(\mathbb{E}[V \mid \gamma, x]) \tag{34}$$

Observed day-to-day spread may combine measurement variation within a regime and systematic shifts between regimes.

5.4.6 CVaR and sample-based optimization

Conditional value-at-risk measures the average upper-tail loss. The Rockafellar–Uryasev (2000, 2002) representation in Equation 35 introduces an auxiliary threshold and a positive-part expectation. It is convex in that threshold for a fixed loss distribution; congestion-pricing optimization need not be convex in the toll vector because network equilibrium depends on tolls.

A CVaR-oriented policy can trade higher average total travel time for lower tail loss. Both quantities must be reported, with the confidence level, evaluation distribution, and scenario count. The Chicago results in Section 7.3 make this trade-off visible.

$$CVaR_{\alpha}(Z) = min_{\xi}\left\{\xi + \frac{1}{1-\alpha}\mathbb{E}[(Z-\xi)_{+}]\right\} \tag{35}$$

The objective focuses on the average of adverse outcomes beyond an optimized tail threshold.

The expectation is approximated by Monte Carlo samples, giving the sample-average objective in Equation 36. The minimization includes the nonnegative toll vector τ and the auxiliary threshold ξ; $Z_s(\tau)$ is total travel time in scenario s, and S is the number of scenarios. For future policy evaluation, independent evaluation draws and common scenarios across competing policies would help assess the precision of small tail-risk differences.

$$min_{\tau,\xi}\xi + \frac{1}{(1-\alpha)S}\sum_{s=1}^{S}(Z_s(\tau)-\xi)_{+} \tag{36}$$

A finite set of learned equilibrium scenarios converts tail-risk planning into a sample optimization problem that can be differentiated through the equilibrium solver and treated with subgradients at positive-part kinks.

Table 9. Evaluation layers for Distributional User Equilibrium.

Layer	Question	Evaluation measure or further diagnostic
Forward equilibrium	Do sampled regimes produce feasible, converged states?	Conservation residual, equilibrium gap, solver stability
Point prediction	Is the conditional center accurate?	MAPE, RMSE, bias, link-level residuals
Distribution	Are spread, dependence, and tails reproduced?	Coverage, width, quantile loss, correlation, proper scores
Latent model	Are regimes stable and plausible?	Coefficient ranges, sensitivity, posterior checks
Decision	Does uncertainty improve an intervention?	Out-of-sample mean, CVaR, regret, policy stability

5.5 Further validation needs

The reported experiments evaluate point prediction, interval coverage and width, and risk-sensitive pricing under controlled conditions. Before empirical deployment, evaluation should also test conditional calibration across day types, joint dependence among links, and sensitivity to observation noise and distribution shift. Proper scoring rules can complement coverage, but these additional analyses are recommendations rather than results of this project.

Policy comparisons should report mean and tail outcomes on common independent evaluation scenarios, together with uncertainty in their differences. A small numerical CVaR advantage is not evidence of a statistically distinguishable improvement without repeated-run or sampling-error analysis. Repeated real-world network states are needed to determine whether the learned regime distribution captures variability beyond measurement and sampling error.

6. City-Scale Trajectory Generation and OD Calibration

6.1 Problem setting and design rationale

City-scale traffic simulation requires OD flows, routes, and departure times. Conventional demand models combine surveys and other observations with behavioral assumptions; obtaining and calibrating their inputs can be costly. This project investigates a complementary source of demand inputs: a generative trip model trained on probe trajectories and calibrated using sparse roadside counts (Wang et al., 2026).

Probe trajectories reveal departure times and routes but cover a spatially and temporally nonuniform subset of traffic. Fixed sensors supply an independent count reference at selected locations, subject to their own detection and processing errors. Scaling or resampling the probe corpus cannot introduce routes absent from that corpus. Generation can broaden candidate support, while count calibration constrains its aggregate scale; neither step automatically removes population sampling bias.

OD matrix estimation adjusts a prior to improve assigned-flow agreement with counts, typically while penalizing changes from that prior. Generative trajectory models supply candidate trips but do not, by themselves, establish count consistency. The proposed method connects these tasks: a constrained generator supplies the OD prior and empirical route shares, and a regularized convex OD matrix estimation (ODME) problem adjusts demand. Generated trips can include routes absent from the observations, but they need not be duplicate-free.

The generation task has four objectives: valid network paths, fidelity to observed trip characteristics, agreement with aggregate counts, and diverse route and OD support. Hard decoding constraints enforce format and permitted network transitions. Behavioral realism,

diversity, and count accuracy are empirical objectives, not automatic mathematical guarantees. Figure 6 shows the four-module workflow and Table 10 summarizes its components.

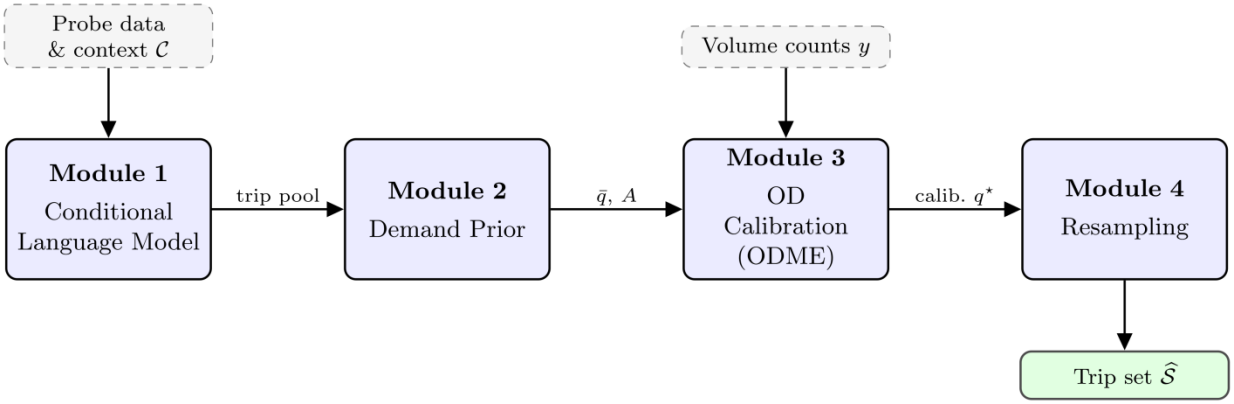


Figure 6. Four-module workflow for calibrated demand generation: constrained trip generation, demand-prior extraction, OD calibration, and resampling. Source: Wang et al. (2026).

Table 10. Inputs, outputs, and roles of the four demand-generation modules.

Module	Input	Output	Role and evidence boundary
1. Conditional language model	Probe trips $\mathcal{S}_{\text{probe}}$, context $\mathcal{C}$, counts y (soft use)	Generated trip pool $\mathcal{G}$	Hard format and transition constraints; realism and diversity require evaluation
2. Demand prior	Pool $\mathcal{G}$, scale factor ϕ	Prior OD demand $\bar{q}$, route shares $P(p \mid r)$, assignment matrix A	Data-driven OD structure and route choice without a survey seed
3. OD calibration	$\bar{q}$, A, counts y on Ω	Calibrated OD demand $q^\star$	Regularized count fit; unique optimizer conditional on prior and assignment
4. Resampling	$q^\star$, pool $\mathcal{G}$	Trip set $\hat{\mathcal{S}}$	Feasible trips with rounded OD totals; final count fit must be checked

6.2 Notation and available observations

This section follows the report-wide notation of Section 3.5 wherever an object has a counterpart there: q denotes OD demand, v link volume, y measured counts, Λ the link-path incidence matrix, and Δ the OD-path incidence matrix. Symbols specific to the language model are defined where they first appear and are local to this section; in particular, θ denotes language-model weights here rather than equilibrium parameters. Table 11 collects the symbols. The source manuscript uses a different lettering, with x for OD demand and q for link counts, so the equations and values quoted from it have been translated to the report notation.

The road network is a directed graph $G = (\mathcal{N}, \mathcal{A})$ whose nodes are intersections and whose links $a \in \mathcal{A}$ are directed road segments. The study region is partitioned into a set $\mathcal{Z}$ of traffic analysis zones (TAZs), and an OD pair is $r = (o, d) \in \mathcal{Z} \times \mathcal{Z}$. Two temporal resolutions serve two purposes. At the microscopic level, an individual departure time $\tau \in \mathcal{D}$ is recorded at a fine resolution, such as one minute. At the macroscopic level used for volume aggregation and calibration, the study period is partitioned into K equal time slices $T = \{t_1, \dots, t_K\}$, such as hours. The map $\kappa: \mathcal{D} \to \mathcal{T}$ returns the unique slice $\kappa(\tau)$ that contains departure time τ.

A trip is $\omega = (\tau, p)$, where $p = (a_1, a_2, \dots, a_L)$ is its route, an ordered sequence of links in which consecutive links are connected in G. The origin and destination zones of a trip are those of its first and last links, which defines its OD pair $r(\omega)$. The time-dependent OD demand $q = \{q_{r,t}\}$ counts the trips of OD pair r whose departure time falls in slice t, and $v_{a,t}$ is the volume on link a in slice t. Each link carries a context feature vector $x_{a,t}$, such as topological attributes and speed relative to the speed limit. The network-context tensor $C = \{x_{a,t}\}$ is the link- and slice-resolved counterpart of the context x used in Sections 4 and 5.

Probe trajectories provide sampled departure times and matched routes; fixed sensors provide counts at selected link–time pairs. The observation operator selects those measured pairs, as in Equation 12. Counts are used as an independent reference, but detection, aggregation, and map-matching errors can remain. Ratios of probe and total counts help estimate penetration where both are available; elsewhere, additional estimation assumptions are needed. The resulting total-demand estimate sets the scale of the generated prior.

Table 11. Notation used in Section 6.

Symbol	Meaning
$G = (\mathcal{N}, \mathcal{A}), a \in \mathcal{A}$	Road network; directed link (road segment)
$\mathcal{Z}, r = (o, d)$	Traffic analysis zones; OD pair
$\tau \in \mathcal{D}, t \in \mathcal{T}, \kappa$	Fine departure time; macroscopic time slice; map from departure time to its slice
$\omega = (\tau, p), p = (a_1, \dots, a_L)$	Trip and its route, a connected sequence of links
$C = \{x_{a,t}\}$	Time-varying network context (link features by slice)
$\mathcal{S}_{\text{probe}}$	Probe trajectories; training corpus of the language model
$y_{a,t}, \Omega$	Observed link count; set of instrumented link-slice pairs
$\mathcal{V}, w^{\tau}, w^{a}$	Token vocabulary; departure-time token; link token
$\mathbf{w}(\omega), w_k, n$	Serialized trip sentence; token at position k; sentence length
$P_\theta, \mathbf{z}_k, \mathbf{b}_k$	Language model with weights θ; logits and logit bias at position k
$\mathbf{C}_T, \mathbf{C}_L(\tau)$	Time context; link context at the departure slice of a trip
$\mathcal{G}, M, \widehat{N}, \phi$	Generated trip pool; pool size; estimated total demand; demand-scale factor $\widehat{N}/M$
$\overline{q}_{r,t}, q^{\star}_{r,t}$	Prior and calibrated OD demand
$\mathcal{P}_r, P(p \mid r)$	Generated route set and empirical route-choice probability of OD pair r
$\Lambda_{a,p}, A_{a,r}, v_{a,t}$	Link-path incidence; OD-to-link assignment matrix; modeled link volume
$\lambda, \overline{y}$	Prior-regularization weight; mean observed count per calibrated link-slice

Symbol	Meaning
$\hat{\mathcal{S}}$	Calibrated, resampled trip set

6.3 Trip generation with a constrained, context-aware language model

Module 1 treats trip generation as language generation. A language model assigns a probability to a sequence of discrete tokens by predicting each token from the tokens before it; generative pretrained transformers trained in this way on text learn grammar and long-range dependencies without hand-written rules. A trip has the same structure. It is a sequence of discrete decisions, a departure time followed by one link after another, and each decision depends on the decisions already made. Module 1 therefore defines a purpose-built language of traffic whose words are departure times and network links and whose sentences are trips, trains a language model on the probe corpus, and generates new sentences under network constraints. Route choice and departure-time choice are learned implicitly from data, without a hand-crafted utility function or a user-equilibrium solution. Figure 7 shows the architecture.

6.3.1 Trips as sentences in a traffic language

The token vocabulary is $\mathcal{V} = \mathcal{V}_{\text{time}} \cup \mathcal{V}_{\text{link}} \cup \mathcal{V}_{\text{ctrl}}$. The set $\mathcal{V}_{\text{time}}$ contains departure-time tokens at a fixed temporal resolution, written T_HHMM; $\mathcal{V}_{\text{link}}$ contains one token per directed link $a \in \mathcal{A}$; and $\mathcal{V}_{\text{ctrl}} = \{[\text{BOS}], [\text{SEP}], [\text{PAD}], [\text{UNK}], [\text{EOS}]\}$ contains control tokens that mark sentence structure and support training and inference. Let $w^{\tau} \in \mathcal{V}_{\text{time}}$ encode departure time τ and $w^{a} \in \mathcal{V}_{\text{link}}$ encode link a. Because a language model operates on one flat token sequence rather than on the structured pair $\omega = (\tau, p)$, each trip is serialized as in Equation 37.

$$\mathbf{w}(\omega) = ([\text{BOS}],\, w^{\tau},\, [\text{SEP}],\, w^{a_1},\, w^{a_2},\, \ldots,\, w^{a_L},\, [\text{EOS}]) \tag{37}$$

A trip becomes a sentence: a start marker, a departure-time word, a separator, the links of the route in travel order, and an end marker.

A superscript on w names the trip component that a token encodes, whereas a subscript, as in w_k, indexes a position in the sentence; the sentence of a route with L links has $n = L + 4$ tokens. The serialized probe trips form the training corpus of the traffic language. Encoding a route directly as link tokens, rather than as GPS coordinates or grid cells, keeps generation on the road network and lets one token stand for a whole road segment, which yields short, information-dense sentences.

6.3.2 Backbone and training objective

The backbone is a compact decoder-only transformer with the Qwen3 architecture, trained from scratch. Because the vocabulary is a purpose-built traffic language rather than natural-language text, this implementation does not use pretrained language weights. Performance therefore rests on learning from trip data and on the architecture and decoding constraints described below. Let θ denote the model weights and $\mathbf{z}_k \in \mathbb{R}^{|\mathcal{V}|}$ the logits, one unnormalized score per vocabulary token, produced at position k. Conditioned on the network context $\mathcal{C}$, the model factorizes the sentence probability autoregressively and converts logits into probabilities with a softmax:

$$P_\theta(\mathbf{w} \mid \mathcal{C}) = \prod_{k=2}^{n} P_\theta(w_k \mid w_{<k}, \mathcal{C}), \qquad P_\theta(w_k = i \mid w_{<k}, \mathcal{C}) = \frac{\exp(z_{k,i})}{\sum_{j\in\mathcal{V}} \exp(z_{k,j})} \tag{38}$$

The probability of a trip is the product of next-token probabilities, each obtained by normalizing the model's scores over the whole vocabulary.

The product starts at $k = 2$ because $w_1 = [\mathrm{BOS}]$ is always the prompt. Inside the transformer, causal self-attention lets the prediction at position k read the entire prefix $w_{<k}$, so the next link can depend on the departure time, the origin, and every link already traversed, not only on the current link as in a first-order Markov chain. This is what allows the model to represent dependencies that span a whole route. Training minimizes the next-token cross-entropy over the corpus:

$$\mathcal{L}(\theta) = -\sum_{\omega\in\mathcal{S}_{\mathrm{probe}}} \sum_{k=2}^{n(\omega)} \log P_\theta(w_k \mid w_{<k}, \mathcal{C}) \tag{39}$$

Training adjusts the model so that the observed probe trips, token by token, become as probable as possible.

6.3.3 Traffic-context extraction and fusion

Travel behavior depends on the state of the network: travelers respond to congestion as well as to topology. To represent this state, the raw features $x_{a,t}$ attached to each link and time slice are passed through a graph convolutional network (GCN) over G, which produces traffic-state embeddings for all links and slices. Two encoders then summarize these embeddings into contexts for the two decisions that make up a trip. A global (time) encoder aggregates them across the network into a time context $\mathbf{C}_T$ for departure-time choice. A local (link) encoder produces a link context $\mathbf{C}_L(\tau)$ for route choice. Because route choice is time dependent, the local encoder is conditioned on the trip's departure-time token, so that $\mathbf{C}_L(\tau)$ reflects the network state in slice $\kappa(\tau)$.

The contexts enter the autoregressive token stream through cross-attention, in which a token's hidden state acts as the query and the context supplies the keys and values. Each context is routed to the positions that make the corresponding decision: the [BOS] state attends to $\mathbf{C}_T$ before w^τ is predicted, while the [SEP] state and every link-token state attend to $\mathbf{C}_L(\tau)$ before the next link is predicted. The resulting conditional distributions are

$$P_\theta(w^\tau \mid [\text{BOS}], \mathbf{C}_T) \quad \text{and} \quad P_\theta\big(w^{a_\ell} \mid [\text{BOS}], w^\tau, [\text{SEP}], w^{a_1}, \ldots, w^{a_{\ell-1}}, \mathbf{C}_L(\tau)\big) \tag{40}$$

The departure time is chosen given the network-wide traffic state, and each next link is chosen given the route so far and the link-level traffic state at the departure time.

The model is therefore context aware: the distribution of departure times and routes can vary with network conditions. Section 7.2 isolates the effect of this conditioning by comparing the full model with a base variant that omits it.

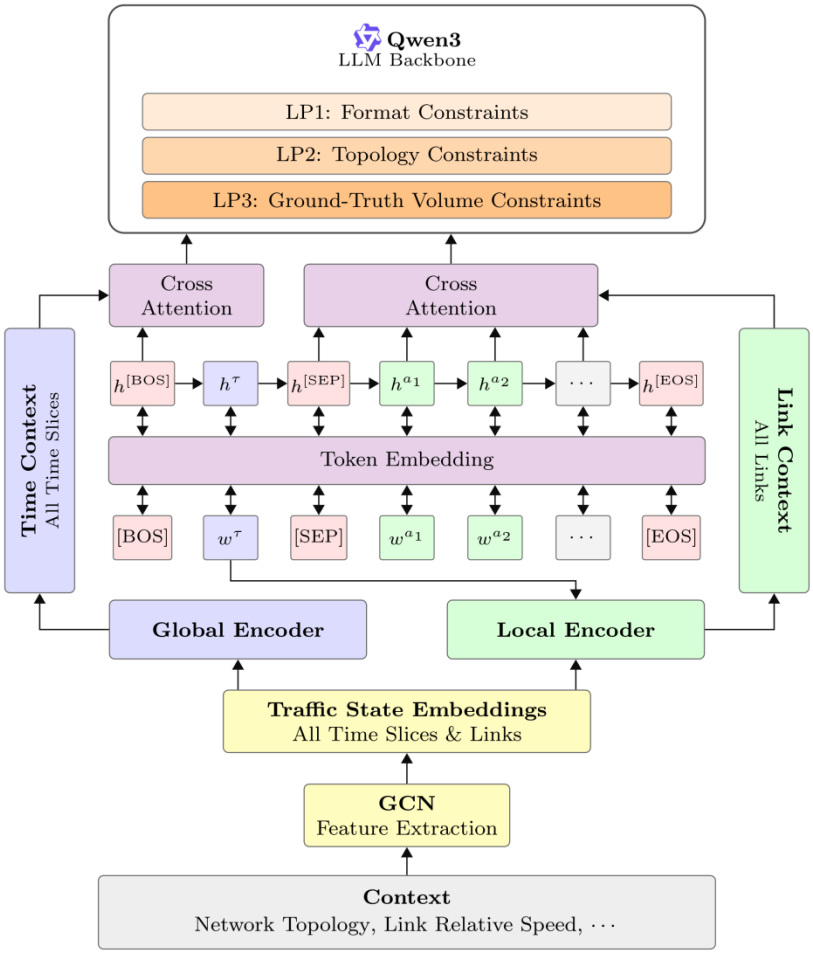


Figure 7. Architecture of the conditional language model (Module 1). Network context is encoded by a graph convolutional network and fused into the autoregressive backbone through cross-attention; logits processors LP1 to LP3 constrain decoding. Source: Wang et al. (2026).

6.3.4 Constrained decoding with logits processors

A freely sampled token sequence need not be a valid trip: it may break the sentence format, jump between disconnected links, or revisit a link. Decoding is therefore steered by logits processors, which add a bias vector $\mathbf{b}_k$ to the logits before the softmax:

$$\tilde{P}(w_k = i \mid w_{<k}, C) = \frac{\exp(z_{k,i}+b_{k,i})}{\sum_{j\in\mathcal{V}} \exp(z_{k,j}+b_{k,j})} \tag{41}$$

Constraints act on the model's scores before normalization: a bias of minus infinity makes a token impossible, and a finite bias only makes it more or less likely.

The bias is the sum of three processors, $b_{k,i} = b_{k,i}^{\mathrm{fmt}} + b_{k,i}^{\mathrm{topo}} + b_{k,i}^{\mathrm{vol}}$, and setting $b_{k,i} = -\infty$ removes token i from consideration at position k. The first two processors impose hard feasibility; the third is a soft prior.

LP1, format. The format processor enforces the template of Equation 37: after [BOS] only a time token is allowed; after the time token only [SEP]; after [SEP] only a link token; and after a link token only another link token or [EOS]. Every grammar-violating token receives $b^{\mathrm{fmt}} = -\infty$.

LP2, topology. When the most recent link token is w^a, the topology processor admits as the next link only links a' that are connected to a in G, and it forbids any link already used in the current route. Every generated route is therefore a connected path without repeated links. This constraint excludes repeated links but does not, by itself, exclude revisiting a node. Because each trip is connected, the flow it contributes enters and leaves every intermediate node, so the generated demand conserves flow by construction rather than through post-processing.

LP3, volume. The volume processor softly steers the aggregate generated flow toward the observed counts. Consider a step that extends a trip whose departure time τ has already been generated, with slice $t = \kappa(\tau)$, and a candidate token i that encodes link a'. Let $v_{a',t}^{\mathrm{gen}}$ be the running generated volume on a' in slice t, aggregated over the trips generated so far and assigned to slices by departure time. Because the pool is deliberately larger than the demand (Section 6.3.5), the pool-scale target is $y_{a',t}/\phi$, where $\phi = \widehat{N}/M \leq 1$ is the demand-scale factor. The processor adds

$$b_{k,i}^{\mathrm{vol}} = \beta\left(y_{a',t}/\phi,\ v_{a',t}^{\mathrm{gen}}\right) \tag{42}$$

Links whose generated volume is still below their pool-scale count target are made more attractive, and links that have reached the target are discouraged.

The function β is positive while the generated volume is below the target and decreases toward a strongly negative value once the target is reached; links without observations receive $b^{\mathrm{vol}} = 0$. To avoid forcing every trip to start on an under-filled link, LP3 acts only from the second link of a route onward, and v^{gen} is refreshed after each batch of generated trips. Unlike LP1 and LP2, LP3 does not guarantee aggregate count matching: because trips are generated one at a time and LP3 is a local, per-trip bias, it cannot by itself make aggregate volumes consistent with the counts. Aggregate count fit is improved by the regularized calibration of Module 3, and the experiments in Section 7.2 confirm that LP3 alone improves count fit only modestly.

6.3.5 Trip-pool generation

At inference, each trip is generated autoregressively from a single [BOS] prompt under the constraints above: first the departure-time token, then the route link by link until [EOS]. Repeating this produces a pool $G = \{\omega_1, \dots, \omega_M\}$ of M generated trips. The pool is deliberately larger than the demand of the study period so that it covers diverse routes; M therefore typically far exceeds the estimated demand $\widehat{N}$, and the two scales are related by $\phi = \widehat{N}/M$. Module 2 uses the same factor to return pool counts to demand scale. Because trips are sampled from the learned distribution rather than retrieved from the corpus, the pool can contain routes and OD pairs that no probe recorded, while LP1 and LP2 guarantee that each of them is a feasible trip.

6.4 Demand-prior extraction

Module 2 turns the generated pool into the two macroscopic objects that calibration requires: a prior OD demand matrix and an OD-to-link assignment matrix derived from empirical route choice. For a trip $\omega = (\tau, p) \in \mathcal{G}$, let $r(\omega)$ be its OD pair, determined by the TAZs of its first and last links, and let $\kappa(\tau)$ be its departure slice.

Prior OD demand. Counting pool trips by OD pair and departure slice and rescaling them to demand scale gives

$$\bar{q}_{r,t} = \phi \cdot |\{\omega = (\tau, p) \in \mathcal{G}\colon\ r(\omega) = r,\ \kappa(\tau) = t\}| \tag{43}$$

The prior demand of an OD pair in a time slice is the number of generated trips in that cell, scaled from pool size down to the estimated demand.

The generated prior reflects the patterns learned from the probe sample and the constraints used during generation. It can retain sampling bias. Its support bounds the calibrated solution: an OD–time cell with zero prior remains zero after calibration. Broader generated support therefore provides additional candidate cells, but independent validation is needed to assess whether those added cells represent actual demand.

Empirical route choice. For each OD pair r, let $\mathcal{P}_r$ be the set of distinct routes taken by pool trips from o to d, and let n_p be the number of pool trips that follow route p. The empirical route-choice probability is

$$P(p \mid r) = \frac{n_p}{\sum_{p' \in \mathcal{P}_r} n_{p'}}, \qquad p \in \mathcal{P}_r \tag{44}$$

Route-choice probabilities are the route shares observed among the generated trips of each OD pair.

The choice set $\mathcal{P}_r$ is data driven: it contains exactly the routes that the model generates between o and d, rather than a fixed or enumerated set of shortest paths. Route shares are pooled across time slices, which yields one route distribution per OD pair and keeps the estimates stable for OD pairs with few trips in any single slice.

Assignment matrix. The assignment matrix combines the route-choice probabilities with the links that each route traverses:

$$A_{a,r} = \sum_{p \in \mathcal{P}_r} P(p \mid r)\, \Lambda_{a,p}\,, \qquad v_{a,t} = \sum_r A_{a,r}\, q_{r,t} \tag{45}$$

Each trip of an OD pair is spread over its generated routes in proportion to their shares, so the modeled volume of a link is the demand-weighted share of routes that use it.

Here $\Lambda_{a,p} = 1$ if link a lies on route p and 0 otherwise, which is the link-path incidence of Equation 4. Because generated routes do not repeat links, $A_{a,r}$ is the expected number of times that a trip of OD pair r uses link a. In matrix form, $A = \Lambda\, \mathrm{diag}(\pi)\, \Delta^{\top}$, where π stacks the route probabilities $P(p \mid r)$: the path flows $h = \mathrm{diag}(\pi)\, \Delta^{\top} q$ satisfy $\Delta h = q$ because the route shares of each OD pair sum to one, and $v = \Lambda h$. Module 2 therefore produces a feasible network loading in the sense of Equation 4, with route proportions fixed by the generator.

Two approximations make the calibration linear in demand. Route shares are fixed and pooled across time, and every traversed link is assigned to the trip's departure slice. These choices do not represent congestion-responsive rerouting or downstream passage times. Their adequacy depends on the application; temporal aggregation alone does not establish that the resulting count error is negligible.

6.5 OD demand calibration

The prior $\bar{q}$ is only as accurate as the generated pool, and Module 1 anchors generation to the counts only softly. Module 3 therefore calibrates demand against the counts by solving a prior-regularized ODME problem:

$$q^{\star} = \arg\min_{q \geq 0} \; \sum_{(a,t) \in \Omega} \Big(\sum_r A_{a,r}\; q_{r,t} - y_{a,t} \Big)^2 + \lambda \sum_{(r,t):\, \bar{q}_{r,t} > 0} \frac{\left(q_{r,t} - \bar{q}_{r,t}\right)^2}{\bar{q}_{r,t}} \tag{46}$$

Choose the nonnegative OD demand whose assigned link volumes best match the counts while changing each OD cell as little as possible relative to its prior size.

The first term fits modeled link volumes to observed counts on the instrumented subset Ω. The second term penalizes squared deviations from the prior, weighted by the inverse prior and summed over OD cells with a positive prior; cells outside the support of the prior are fixed at zero. The weight $\lambda > 0$ trades off the two terms, and $q \geq 0$ enforces nonnegative demand. No other scaling constants enter the objective: the relative weight of the two terms is carried entirely by λ, whose value is fixed by the rule in Equation 48.

The inverse-prior penalty weights a squared change by the size of its prior cell. It is consistent with a generalized-least-squares model when prior-error variance is proportional to prior demand, and with a local quadratic approximation to an entropy-based penalty (Cascetta, 1984; Van Zuylen & Willumsen, 1980). These are modeling motivations rather than a verified variance model for the generated pool. The penalty discourages large absolute changes to weakly supported cells; it does not force a particular multiplicative update.

Matrix form and uniqueness. Stack the supported OD cells into q, let $\mathbf{A}_\Omega$ be the matrix that maps q to modeled volumes at the instrumented link-slices (row (a,t) holds $A_{a,r}$ in the columns of cells (r,t) of the same slice), let y_Ω stack the counts, and let $\overline{D} = \mathrm{diag}(\bar{q})$. Equation 46 then reads

$$q^\star = \arg\min_{q \geq 0} \|\mathbf{A}_\Omega\, q - y_\Omega\|_2^2 + \lambda \|q - \bar{q}\|_{\overline{D}^{-1}}^2 \tag{47}$$

Calibration is a nonnegative least-squares fit to the counts with a weighted penalty that pulls each OD cell toward its prior.

The data term is convex, and the penalty is strictly convex because every supported cell carries a positive weight $1/\bar{q}_{r,t}$; the Hessian of the objective is proportional to $\mathbf{A}_\Omega^\top \mathbf{A}_\Omega + \lambda \overline{D}^{-1}$, which is positive definite. The minimizer $q^\star$ is therefore unique, and Equation 47 is a convex quadratic program with nonnegativity bounds that standard solvers handle at city scale. The matrix form also shows how the two information sources divide the work. Sparse counts leave many directions of q unobserved: OD cells whose routes never cross an instrumented link correspond to zero columns of $\mathbf{A}_\Omega$, and cells whose routes cross the same sensors in the same proportions correspond to nearly collinear columns. In those directions the data term is flat or nearly flat, and the solution is determined by the prior. Counts set the scale where they are informative, and the generated prior resolves what the sensor layout cannot distinguish.

Strict convexity gives a unique optimizer conditional on the assignment matrix, prior, and regularization weight. It does not identify the true OD demand uniquely from sparse counts. Because count mismatch is penalized rather than constrained to zero, residuals generally remain at calibration locations.

The study sets the regularization weight to the mean count over calibration link–time pairs, without using validation counts for this choice. A Poisson-like count variance and prior variance

proportional to prior size motivate the rule. Equation 48 expresses the corresponding rescaling; it should not be interpreted as an empirically verified equality of the two data sources' uncertainty.

$$F(q) = \bar{y}\left[\sum_{(a,t)\in\Omega} \frac{\left((\mathbf{A}_\Omega q)_{a,t} - y_{a,t}\right)^2}{\bar{y}} + \sum_{(r,t):\,\bar{q}_{r,t}>0} \frac{\left(q_{r,t} - \bar{q}_{r,t}\right)^2}{\bar{q}_{r,t}}\right] \tag{48}$$

The mean-count rule supplies a reproducible relative weight under approximate variance assumptions; sensitivity to that weight remains an empirical question.

The common positive factor does not affect the minimizer. The Ann Arbor application uses a mean calibration count of approximately 495. Future sensitivity analysis should vary this weight using training or validation data reserved for tuning, without using the final evaluation counts.

6.6 Trip resampling

Module 3 returns an aggregate quantity, the calibrated OD demand $q^\star$. Module 4 turns it back into individual trips and retains the individual trip patterns represented in the pool. For each OD pair r and slice t, let $\mathcal{G}_{r,t} = \{\omega = (\tau, p) \in \mathcal{G}:\ r(\omega) = r,\ \kappa(\tau) = t\}$ be the pool trips in that cell, and let $n_{r,t} = \lfloor q^\star_{r,t} \rceil$ be the calibrated trip count, where $\lfloor\cdot\rceil$ rounds to the nearest integer. Drawing $n_{r,t}$ trips uniformly, with replacement, from each cell gives the final trip set

$$\hat{\mathcal{S}} = \biguplus_{r,t} \left\{\omega^{(1)}, \ldots, \omega^{(n_{r,t})}\right\}, \qquad \omega^{(m)} \sim \mathrm{Unif}\left(\mathcal{G}_{r,t}\right) \tag{49}$$

Each calibrated OD–time total is filled by sampling trips from the same cell. The multiset union retains repeated draws, as required for sampling with replacement.

Uniform sampling within an OD–time cell reproduces that cell's empirical route and departure-time distribution in expectation. It need not reproduce the time-pooled route shares used in the assignment matrix. Unless cell-specific and pooled route shares coincide, expected link volumes after resampling can differ from the calibrated matrix prediction. Integer rounding and finite sampling introduce additional discrepancies. Resampling with replacement can also duplicate trips.

The final trip set inherits network feasibility from the pool and has rounded calibrated OD–time totals. Its count fit, diversity, and trip distributions must be evaluated after resampling. To ensure calibration and sampling use the same route model, an implementation can use time-specific assignment matrices or a sampling rule consistent with the pooled shares. The choice depends on the implemented assignment and sampling procedures; count-fit claims must be evaluated on the final delivered trips.

6.7 Scope, assumptions, and extensions

The demand prior can retain probe bias, and its aggregate scale depends on the penetration-rate estimate. Calibration improves count agreement under fixed route shares and departure-slice aggregation; it neither guarantees exact counts nor validates dynamic traffic behavior. Independent evaluation should distinguish assigned volumes, the final resampled trip set, and simulator outputs. The Ann Arbor study reports links withheld from OD calibration. A complete validation audit

must also document whether those counts were excluded from volume-guided decoding, penetration-rate estimation, and all other fitted stages.

Further work should align calibration and resampling route shares, evaluate the delivered trip set over repeated sampling seeds, and test sensitivity to penetration rates, prior support, and count errors. Loading trips into a simulator and comparing speeds, travel times, and queues would evaluate demand and supply jointly. Link-level passage times would address the departure-slice approximation. Transfer to other networks or travel modes requires separate validation.

7. Case Studies and Validation Findings

7.1 Ann Arbor inverse-learning study

The equilibrium-learning case study used an assignment network with 810 nodes (178 TAZ centroids and 632 road junctions) and 2,847 links (1,583 road segments and 1,264 centroid connectors). Network topology, zone features, and a reference OD matrix came from the Southeast Michigan Council of Governments (SEMCOG) 2050 regional forecast model. Probe trajectories were collected during the 4:30–5:30 p.m. evening peak throughout 2022, map matched to the network, and averaged 3,127 observed trips per day.

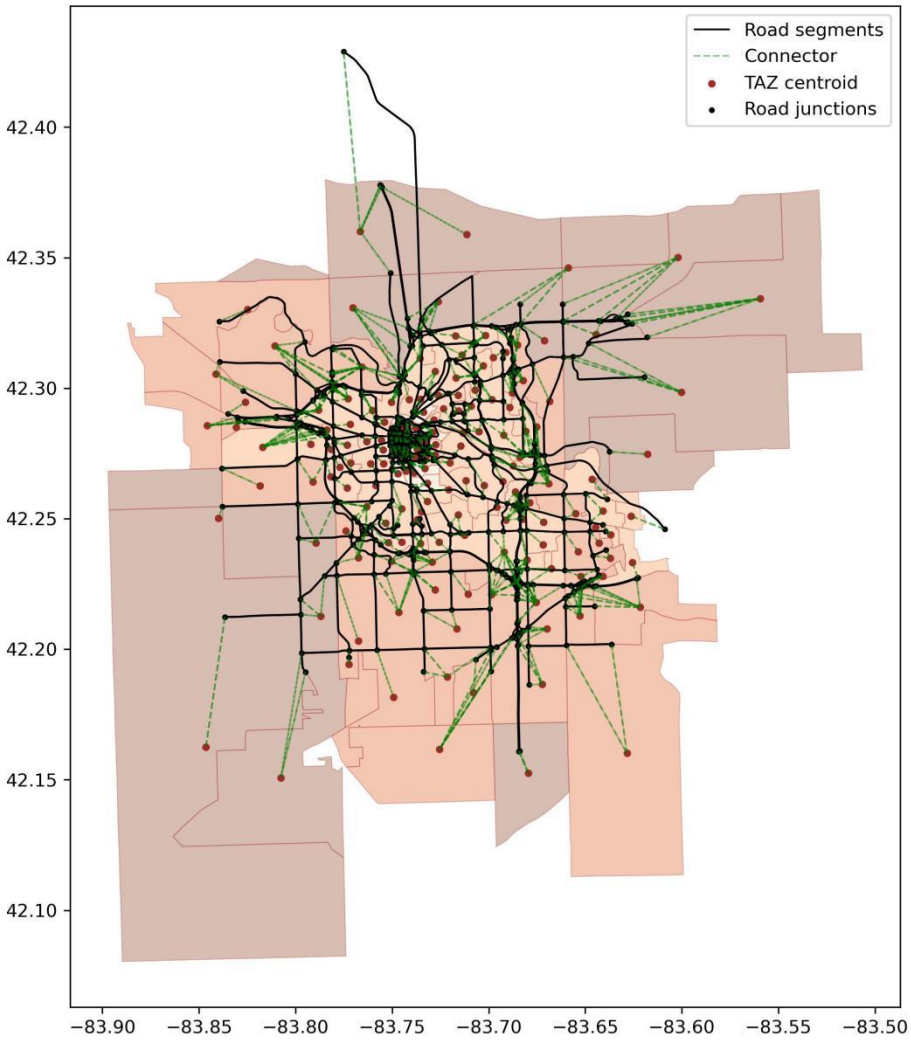


Figure 8. Ann Arbor assignment network used in the context-dependent inverse-learning study. Source: Liu, Yin, Lin, and Wang (2026).

Observed trip counts were lower on snow days and weekends. The model therefore included snow and weekday indicators together with road and OD features. To manage the large OD space, each OD pair used an active-link subset derived from the six fastest paths. Scenario 1 contained 1,817 OD pairs and covered the top 20 percent of reference demand; Scenario 2 contained 10,418 OD pairs and covered 65 percent. The corresponding active sets averaged 25.5 and 27.8 links per OD pair, far smaller than the full 2,847-link network.

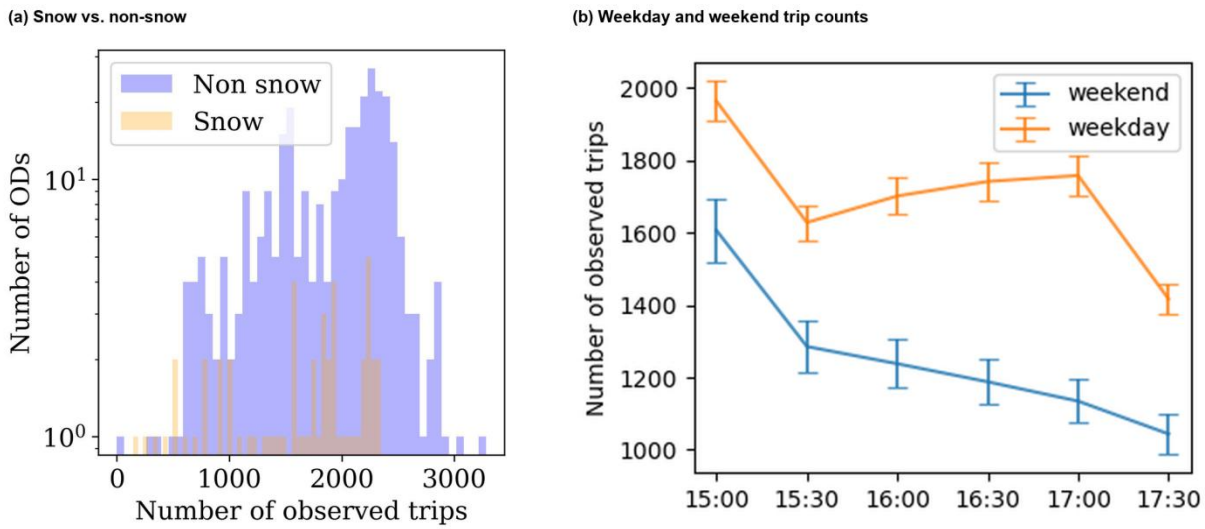


Figure 9. Observed differences in Ann Arbor trip patterns across snow conditions and day types. Source: Liu, Yin, Lin, and Wang (2026).

The study divides observations into 237 training, 64 validation, and 64 test samples and reports averages and standard deviations across three cross-validation runs. Across eight specifications, Scenario 2 link travel-time MAPE ranges from 34.3 to 45.0 percent. The best setting uses linear link performance and quadratic inverse demand: MAPE is 34.3 percent (standard deviation 1.9 percentage points), versus 83.6 percent for the annual-average trip-based benchmark; reported runtime is 1.9 minutes. This comparison evaluates the learned model as a whole and does not isolate context from basis choice, demand specification, and estimation. Snow-related coefficient signs are conditional on the nonnegative parameterization, normalization, and active-link approximation, rather than independent causal estimates.

7.2 Demand-generation benchmarks and Ann Arbor count-validation study

This study evaluates the method of Section 6 at two levels: individual trip generation on two public taxi-trajectory benchmarks, and the complete four-module pipeline on Ann Arbor, where roadside camera counts permit macroscopic validation. The Ann Arbor demand study used a different network and time window from Section 7.1 and should not be merged numerically with the equilibrium-learning application.

7.2.1 Data, baselines, and metrics

Datasets. Table 12 summarizes the datasets. Porto and Beijing are public taxi-trajectory benchmarks. The Ann Arbor study used a 5,615-node, 13,446-link network, 955,623 probe trajectories, and the 302 traffic analysis zones of the 2015 base-year Washtenaw Area Transportation Study (WATS) regional model that cover the Ann Arbor area, a subset of the 809 zones of that model.

Table 12. Datasets used in the demand-generation experiments.

Dataset	Nodes	Links	Trajectories	Average trip length	Average links/trip
Porto taxi	5,330	11,491	1,426,312	5.27 km	51.07
Beijing taxi	31,199	72,156	1,382,948	7.39 km	36.08
Ann Arbor demand study	5,615	13,446	955,623	6.48 km	36.43

Baselines. Individual trip generation was compared with seven generative trajectory models: a random walk, a Markov mobility chain (MMC), SeqGAN, LSTM-TrajGAN, TS-TrajGen, DiffTraj, and MobilityGPT (Haydari et al., 2026). City-scale volume reproduction used a two-by-two design over trip source and calibration. The Baseline method aggregates 15 days of probe trajectories to approximate one synthetic day of volume, since roughly 7 percent penetration over 15 days approximates full traffic. LM-only uses Module 1 generation without the calibration of Module 3. Baseline+ODME applies Modules 2 to 4 to the probe trips instead of the generated pool and provides a comparison with the same calibration method applied to observed trips. Proposed combines the generated pool with calibration.

Metrics. Benchmark evaluation followed the MobilityGPT protocol. Query Error is the normalized absolute difference between the numbers of real and generated trips that pass through 500 uniformly sampled links. Jensen–Shannon divergences (JSD) compare the generated and real distributions of OD pairs, trip length, radius of gyration, and gravity-model flows. For Ann Arbor, microscopic realism was measured by the JSDs of departure time, link frequency, and route length, together with link coverage and the shares of novel routes and OD pairs. Volume reproduction was measured on hourly link volumes by the mean absolute error (MAE); the percent root-mean-square error (%RMSE), which divides the RMSE by the mean observed volume and expresses the result as a percentage; the weighted absolute percentage error (WAPE); R-squared; and the GEH statistic, summarized by the share of link-hours with GEH below five and the median GEH. WAPE is total absolute error divided by total observed volume, expressed as a percentage. GEH is the square root of the squared count difference divided by the mean of modeled and observed counts.

Implementation. The language model used a tailored Qwen3 backbone with about 0.1 billion parameters and a customized tokenizer with a 14,891-token vocabulary, trained from scratch. The implementation used a single graphics processing unit (GPU) and generated on the order of 100,000 trips per hour on one NVIDIA A100.

7.2.2 Individual trip generation on public benchmarks

Table 13 compares two variants of Module 1 with the generative baselines. Proposed (conditional) is the full model; Proposed (base) removes the traffic-context conditioning but keeps constrained decoding. The Porto taxi sample is too sparse to compute meaningful daily traffic states, so only the base variant was trained there. MobilityGPT* is the project team's replication of MobilityGPT without its fine-tuning stage based on reinforcement learning from trajectory feedback (RLTF),

which isolates its pretrained generator; all other baseline rows are quoted from Haydari et al. (2026).

Table 13. Distribution-level evaluation of individual trip generation on public benchmarks. Lower is better; all columns except Query Error are Jensen–Shannon divergences. Bold indicates the best value and underlining indicates the second-best value in each column.

Method	Porto–Query E.	Porto–OD	Porto–Trip L.	Porto–Radius	Porto–Gravity	Beijing–Query E.	Beijing–OD	Beijing–Trip L.	Beijing–Radius	Beijing–Gravity
Random Walk	1.729	0.375	0.739	0.393	0.307	0.105	0.167	0.473	0.187	0.133
MMC	1.409	0.210	0.742	0.393	0.307	0.212	0.167	0.550	0.465	0.291
SeqGAN	0.138	0.116	0.220	0.191	0.326	0.063	0.296	0.162	0.275	0.120
LSTM-TrajGAN	0.234	0.254	0.435	0.425	0.277	0.078	0.182	0.398	0.365	0.219
TS-TrajGen	0.309	0.120	0.161	0.115	0.248	0.277	0.105	0.570	0.485	0.336
DiffTraj	0.267	0.177	0.126	0.123	0.252	0.074	0.146	0.158	0.123	0.114
MobilityGPT*	0.148	0.117	0.195	0.203	0.379	0.079	0.113	0.226	0.172	0.141
MobilityGPT	0.144	0.114	0.124	**0.107**	**0.225**	0.066	0.099	0.123	**0.102**	**0.105**
Proposed (base)	**0.078**	**0.083**	**0.119**	0.118	0.365	0.053	**0.084**	0.138	0.119	0.129
Proposed (conditional)	-	-	-	-	-	**0.050**	0.085	**0.118**	0.105	0.115

Note: MobilityGPT* denotes MobilityGPT without its RLTF fine-tuning stage, replicated by the project team; all other baseline rows are quoted from Haydari et al. (2026). Dashes indicate that the conditional variant was not trained on Porto.

The proposed models attained the best Query Error, OD divergence, and trip-length divergence in both cities. The first two directly assess aggregate demand patterns: Query Error measures link-level through-flow, the quantity that macroscopic calibration later adjusts, and OD divergence measures the demand structure itself. MobilityGPT retained the best radius-of-gyration and gravity divergences in both cities. These individual-mobility-shape metrics are what its gravity-based sampling optimizes, and they provide complementary measures of trajectory realism.

The ablation comparisons clarify the effects of training stages and context conditioning. MobilityGPT relies on post-training: removing its RLTF fine-tuning degrades every metric. The proposed models are pretraining-only, trained by next-token prediction on the traffic language without any fine-tuning stage. Even so, the base model without context conditioning outperformed the pretrained-only MobilityGPT* on all ten columns and surpassed the fully post-trained MobilityGPT on Query Error and OD divergence in both cities. Context conditioning improved four of five metrics on Beijing, where traffic states were available. The metrics that MobilityGPT still leads are the ones its feedback fine-tuning targets, so a comparable post-training stage is a possible extension whose benefit remains to be tested.

7.2.3 Ann Arbor microscopic realism and diversity

Generated trips closely matched the probe sample's marginal distributions: JSD was approximately 0.0002 for departure time, 0.0007 for link frequency, and 0.0005 for route length. Of the distinct generated routes, 91.9 percent were absent from training, demonstrating generation beyond exact route replay. Training-link coverage was 99.3 percent, and TAZ-level OD coverage was 87.6 percent versus 61.4 percent in the observed sample. The top 1,000 routes accounted for 15.8 percent of generated trips and 15.1 percent of training trips. These statistics describe the six-million-trip pool before calibration and resampling. Novelty and marginal agreement do not, by

themselves, establish population realism or absence of memorization. The unequal pool sizes also affect coverage; a matched-size comparison would better isolate the generator's contribution.

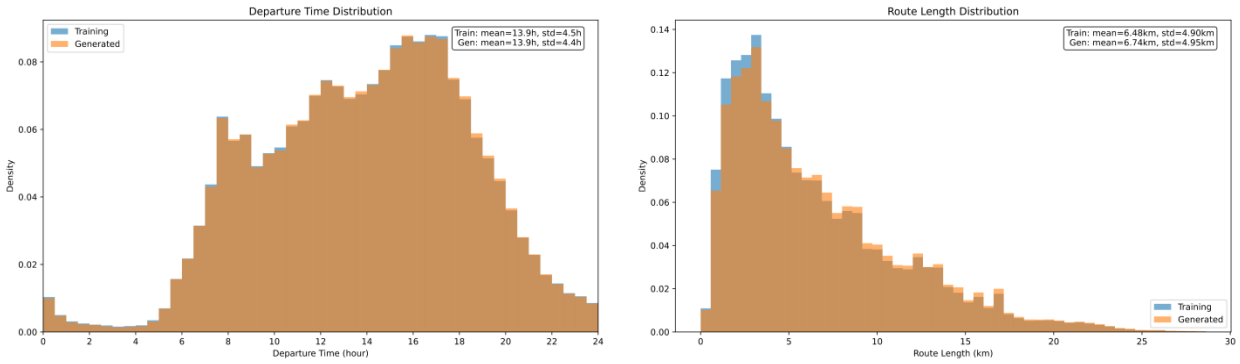


Figure 10. Generated versus training distributions of departure time (left, JSD = 0.0002) and route length (right, JSD = 0.0005) in the Ann Arbor demand study. Source: Wang et al. (2026).

Table 14. Diversity and feasibility of the six-million-trip Ann Arbor generated pool before OD calibration and resampling.

Quantity	Training	Generated	Interpretation
Unique-route share	52.1%	40.7%	Generated traffic preserves recurring route concentration
Top-1,000 route share	15.1%	15.8%	Common-route repetition remains similar
TAZ OD-pair coverage	61.4%	87.6%	Broader candidate support; comparison uses unequal sample sizes
Connected routes without repeated links	-	100.0%	Network constraints prevent infeasible routes
Distinct routes absent from training	-	91.9%	Evidence beyond exact replay; not a complete memorization test
Training-link coverage, (train and gen.) / train	-	99.3%	Most observed network support is retained
Inverse link coverage, (train and gen.) / gen.	-	98.6%	Generated trips rarely use links absent from the probe data

Figure 11 provides illustrative spatial comparisons for entering, transiting, and exiting trips and two activity hotspots. The expanded OD support matters because ODME cannot create demand outside the prior's support. However, broader support creates candidate explanations rather than independently verified new OD movements. Quantitative validation of those additional movements would require observations beyond the training probe sample.

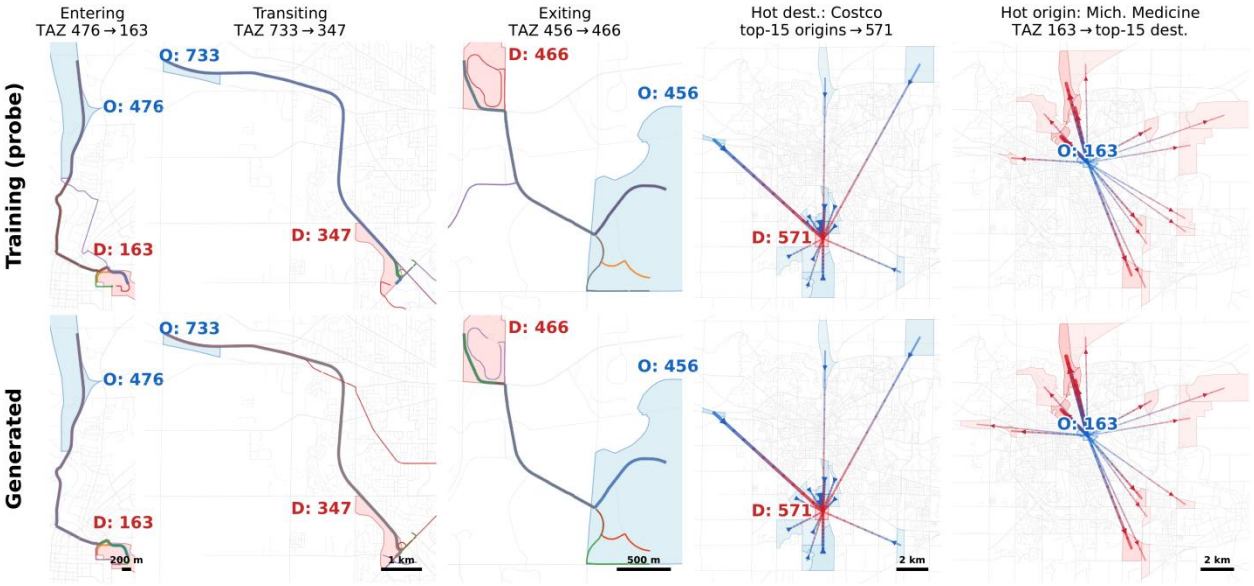


Figure 11. TAZ-level route patterns for training probe trips and generated trips across entering, transiting, exiting, retail-destination, and medical-center-origin examples in Ann Arbor. Source: Wang et al. (2026).

7.2.4 Macroscopic calibration and held-out validation

Reference traffic volumes came from intersection cameras between 9 a.m. and 7 p.m., recorded at half-hourly resolution and aggregated to hourly volumes, which gives 10 time slices for calibration and evaluation. The 24 instrumented intersections contributed 153 links: 129 links at 20 intersections formed the calibration set Ω, and 24 links at four intersections were held out for validation. Because not every calibration link had a valid count in every hour, Ω contained 1,274 of the 1,290 possible link-hours; the validation links yielded all 240. Following Section 6.5, the regularization weight was set a priori to the mean observed volume per calibrated link-hour, $\lambda = \bar{y} \approx 495$; no validation data entered this choice.

Table 15 and Figure 12 compare the four methods. On calibration links, the two calibrated methods far exceeded the uncalibrated ones, with R-squared of 0.73 to 0.74 versus 0.27. Without ODME, LM-only improved MAE from the Baseline value of 192.2 to 169.3. This comparison changes both trip generation and count steering, so it does not isolate LP3. Adding ODME to the generated pool reduced MAE further to 101.8. The calibrated methods were close on these links, with Proposed ahead on every metric (MAE 101.8 and 63.3 percent of link-hours with GEH below five): once ODME is applied, the counts used in fitting can be matched almost equally well from either pool.

On links excluded from OD calibration, Proposed achieved the lowest MAE, percent RMSE, and WAPE and the highest R-squared among the four methods. Relative to Baseline+ODME, MAE fell from 129.1 to 113.5 (12.1 percent), percent RMSE fell from 37.8 to 34.1 percent, and R-squared rose from 0.649 to 0.716. LM-only performed best on the GEH measures, with 50.8 percent below five and median 4.61. These results support the combined approach under the reported split, without establishing superiority on every metric or across repeated sensor selections.

Table 15. Hourly link-volume reproduction in the Ann Arbor demand study. Calibration links were used by ODME; validation links were held out. Lower MAE, %RMSE, WAPE, and median GEH and higher R-squared and GEH<5 share are better. Bold marks the best value in each column of each block.

Panel A. Calibration links (n = 1,274 link-hours)

Method	MAE	%RMSE	WAPE	R-squared	GEH<5	GEH median
Baseline	192.2	49.5%	38.8%	0.265	26.1%	8.35
Baseline+ODME	109.5	30.2%	22.1%	0.728	56.8%	4.47
LM-only	169.3	49.4%	34.2%	0.269	42.0%	6.90
Proposed	**101.8**	**29.5%**	**20.5%**	**0.739**	**63.3%**	**3.67**

Panel B. Held-out validation links (n = 240 link-hours)

Method	MAE	%RMSE	WAPE	R-squared	GEH<5	GEH median
Baseline	143.5	44.9%	35.1%	0.506	35.4%	6.46
Baseline+ODME	129.1	37.8%	31.5%	0.649	35.8%	6.13
LM-only	117.7	40.8%	28.8%	0.592	**50.8%**	**4.61**
Proposed	**113.5**	**34.1%**	**27.7%**	**0.716**	45.4%	5.40

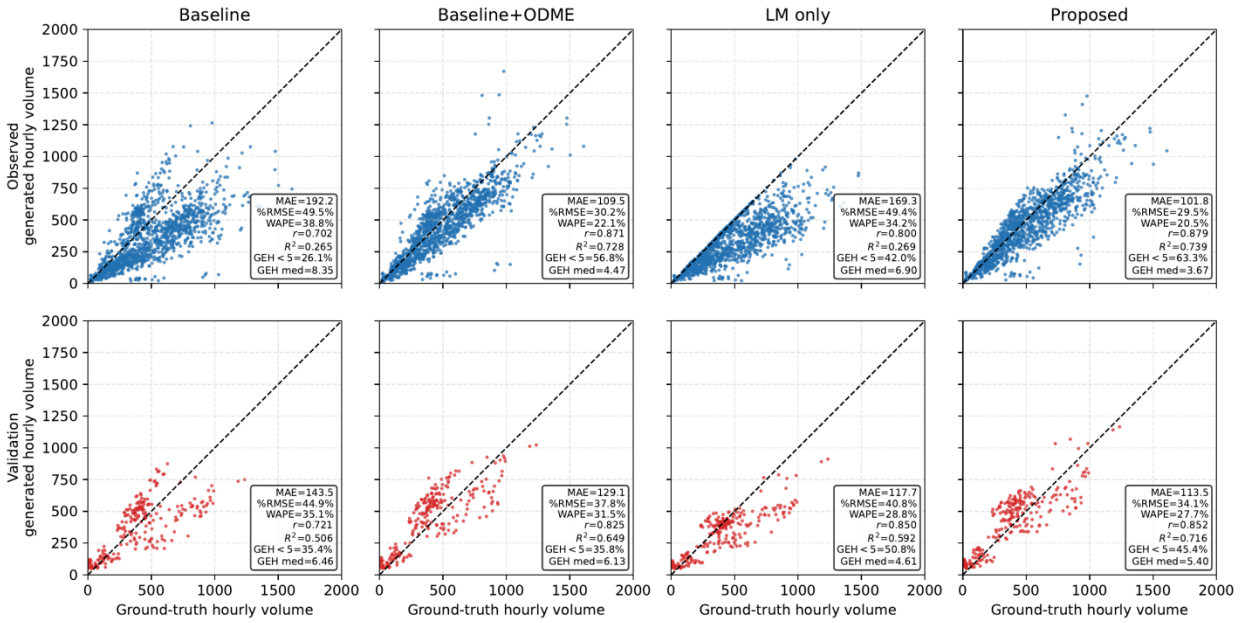


Figure 12. Generated versus observed hourly link volumes for the four Ann Arbor demand methods on calibration and held-out validation links. Source: Wang et al. (2026).

The source paper compares its errors with travel-model and microsimulation guidance (Wegmann & Everett, 2016; Dowling et al., 2004). Such comparisons provide context, but their applicability depends on volume aggregation, facility mix, and evaluation purpose. The Proposed method has 63.3 percent of fitted link-hours below GEH = 5, below the cited 85 percent target. The trip set is therefore a starting point for simulation-based refinement, not a completed microsimulation calibration. Additional validation should identify whether the reported count metrics describe

matrix-assigned volumes or final sampled trips and verify the full upstream exclusion of validation counts.

7.3 Controlled distributional benchmarks

The Distributional UE experiments answer a different question. Sioux Falls provides a small network with 76 links, 24 nodes, and 528 OD pairs, while Chicago Sketch provides 2,950 links, 933 nodes, and 93,135 OD pairs. Demand and latent preference coefficients are generated under known Gaussian, uniform, and discrete regimes. Each demand scenario includes multiple coefficient realizations, and observation noise can be added separately. This design makes it possible to measure whether the latent flow distribution is recovered and whether risk-sensitive pricing performs under the known data-generating distribution.

Sioux Falls results generally show competitive mean-flow prediction, coverage near nominal levels, and favorable CVaR-oriented pricing under the tested synthetic distributions. Chicago Sketch gives a more qualified result. Dist-UE has the lowest non-oracle MAPE (13.65 percent), but nominal 95 percent intervals cover 99.9 percent of outcomes. Its normalized interval width is 2.484, versus 0.599 for Laplace and 0.952 for the ensemble. Thus high coverage does not establish nominal calibration or the sharpest intervals.

For Chicago pricing, Dist-UE's CVaR is 3.559 versus 3.608 for deterministic UE, a reduction of about 1.4 percent. Mean total travel time is 1.689 versus 1.390, an increase of about 21.5 percent. Both use the source table's scale of 10^8 vehicle-hours. CVaR is nearly tied with Monte Carlo dropout's 3.560; the table alone does not establish a statistically distinguishable difference. These findings demonstrate a trade-off between tail risk and mean performance under controlled conditions. They do not establish empirical separation of latent variability, measurement error, and disequilibrium.

7.4 Priorities for further validation

Validation must follow the intended use. Hold out entire days for temporal prediction, locations for spatial transfer, and policy scenarios for intervention evaluation. A final test set must remain outside all training, preprocessing calibration, count-guided generation, and tuning. The project's studies use different splits; their numerical results cannot be pooled as one experiment.

For generated demand, report calibration residuals and independent count errors separately for matrix-assigned volumes and final resampled trips. Repeat resampling, compare diversity at matched sample sizes, and test sensitivity to penetration rates, prior support, and regularization. A simulator-based evaluation should then compare link passage times, speeds, and queues.

For equilibrium learning, evaluate fitting error together with conservation and equilibrium residuals. Compare matched models with and without context before attributing gains to context alone. For distributional learning, report interval width with coverage, test joint network variability, and evaluate mean and tail policy outcomes with uncertainty in their differences.

A diagnosis demonstration should start from a documented anomalous observation, compare demand, capacity, context, and measurement explanations, and evaluate a proposed response against independent evidence. This is a proposed evaluation sequence, not a completed causal diagnosis. Common network versions, data periods, software versions, and split definitions would make future system-level testing reproducible.

7.5 Evidence synthesis across the component studies

The four research products answer complementary questions but do not constitute one end-to-end experimental run. The review and unified inverse-learning work organize formulations and modeling choices. The multiconvex study demonstrates scalable context-dependent learning on its Ann Arbor network and observation design. The distributional study uses controlled networks and repeated-flow settings to evaluate latent uncertainty and risk-sensitive pricing. The demand-generation study uses a separate Ann Arbor network representation, probe sample, camera calibration set, and held-out validation set.

A synthesis can align constructs without pooling observations. Feasibility has a consistent meaning across components, but it is measured differently for a route sequence, an OD matrix, and an equilibrium state. Held-out performance is important in every study, but the held-out unit differs. Context appears in both deterministic and distributional learning, but the variables and sample sizes differ. These relationships support an integrated architecture, not a claim that one fitted model simultaneously achieved all reported numbers.

Table 16 identifies the object evaluated by each study and the evidence that supports it. The demand study's camera-count results do not validate the multiconvex cost model, and controlled distributional results do not establish field calibration on the Ann Arbor study days.

Component-level validation links the scientific contributions to their corresponding assumptions, datasets, and evaluation metrics. This structure supports comparison across the studies while preserving the meaning of each reported result.

Table 16. Evidence boundaries across the integrated report.

Component	Validated object	Evidence and boundary
Unified inverse learning	Formulation and equilibrium-consistent learning logic	Conceptual synthesis and mathematical analysis
Multiconvex learning	Context-dependent potential and scalable optimization	Ann Arbor held-out prediction design
Distributional UE	Distribution of equilibrium states and risk optimization	Synthetic benchmarks; Chicago intervals conservative
Demand generation	Feasible trip pool and calibrated OD demand	Camera links excluded from ODME; full upstream exclusion requires documentation
Integrated diagnosis	Proposed component architecture	No single end-to-end validation experiment

8. Findings

8.1 Technical findings

- Inverse learning provides a common language for constructing equilibrium models from partial observations. Data fit, equilibrium consistency, and functional parameterization are separate modeling choices and should be reported separately.
- The complete learned context-dependent model improves prediction relative to the annual-average benchmark in the Ann Arbor experiment. This comparison does not isolate the incremental effect of context. Learned coefficient directions are conditional associations under the imposed model and constraints.
- Convex-basis potentials and value-function gaps make large inverse problems more tractable. The resulting multiconvex structure supports sample-wise decomposition and GPU parallelism without discarding network feasibility.
- A learned distribution of equilibria supports explicit risk-sensitive decisions in controlled experiments. Chicago results show a modest CVaR reduction accompanied by higher mean total travel time and conservative intervals; uncertainty quality and decision value must be assessed separately.
- Constrained trip generation broadens candidate route and OD support, while regularized ODME improves count fit under fixed route shares. Their combination performed best on several held-out volume metrics in the Ann Arbor study, but LM-only performed best on its GEH measures.
- Held-out validation changes the strength of the evidence. The demand study's R-squared of 0.716 and percent RMSE of 34.1 percent were obtained on links excluded from OD calibration, which is more informative than calibration fit alone.

8.2 Cross-cutting limitations

- Probe trajectories remain a biased sample. Generative and inverse models can reduce but cannot automatically eliminate coverage bias without independent data and validation.
- Equilibrium models can be non-identifiable under partial observations. Good prediction does not guarantee unique recovery of behavioral parameters or causal effects.
- The two Ann Arbor applications use different networks, time windows, and observations. Their numerical results should not be combined as one experiment.
- Distributional UE results come from controlled synthetic experiments, with known data-generating conditions and explicit evaluation of uncertainty calibration and risk-sensitive pricing.

The demonstrated contributions are component methods for trajectory-based modeling and decision analysis. Integrated diagnosis, operational simulation calibration, and transfer to other locations remain distinct validation tasks.

9. Recommendations

9.1 Model selection and evaluation

- Account for uncertainty in data and modeling. Distinguish trajectory sampling uncertainty, count error, map-matching ambiguity, OD prior uncertainty, route-set truncation, equilibrium approximation, and model-form error when interpreting model results.
- Use staged model complexity. Begin with interpretable convex bases and context variables, expand only when held-out performance warrants it, and retain simple benchmarks. Multiple initializations and validation-based selection should be standard for nonconvex inverse problems.
- Treat diagnosis as a comparison among explanations. For a detected mismatch, evaluate demand shift, capacity or cost shift, missing context, and disequilibrium alternatives rather than labeling all residuals as incidents.
- Establish reproducible releases. Record network version, coordinate system, observation window, probe penetration assumptions, code commit, environment, random seeds, and data-use restrictions. Provide public metadata even when trajectories cannot be shared.

9.2 Guidance for agency use

Agencies should first use these methods for planning and diagnostic support where analysts can inspect assumptions and compare outputs with independent evidence. Suitable near-term applications include OD prior development, scenario screening, context-aware performance baselines, sensor-placement studies, and uncertainty-informed comparison of candidate policies.

Performance reports should include both fitted and held-out locations, multiple day types, and metrics with clear units. R-squared should not be used alone because it can remain high with systematic bias; normalized error and GEH-type measures provide complementary views. For uncertainty outputs, coverage and interval width should be reported together. For interventions, mean, tail, and worst observed outcomes should be separated.

Generated trajectories require governance even when they are synthetic. Analysts should test whether rare training routes can be memorized, whether generated OD support reflects population or probe bias, and whether releasing route-level data creates re-identification or security concerns. Aggregated products and controlled access may be appropriate where source trajectories are restricted.

9.3 Reporting and interpretation of results

Report each result together with its network, observation period, calibration data, evaluation split, and metric definition. Keep the Ann Arbor inverse-learning and demand-generation studies distinct, and identify the controlled conditions used for Distributional UE experiments. Interpret predictive accuracy, conservation, equilibrium consistency, and uncertainty calibration as complementary measures. These practices make the reported findings useful for comparing models and assessing their suitability for a specific planning question.

10. Outputs, Outcomes, and Impacts

10.1 Research outputs

Four research studies support this report. Two are published in Transportation Research Part B: Methodological. The distributional-equilibrium and demand-generation studies are cited using their available SSRN manuscripts. Table 17 identifies the sources of the formulations and experimental results summarized here.

Table 17. Research outputs and publication sources.

Output	Primary contribution	Evidence used in this report	Publication record
Constructing Transportation Network Equilibrium Models from Empirical Data: An Inverse Learning Perspective	Unified design framework and research agenda	Sections 3 and 8	Published in Transportation Research Part B, 214, 103597 (2026). DOI: 10.1016/j.trb.2026.103597
Large-Scale Inverse Learning of User Equilibrium via Multiconvex Optimization	Context-dependent nonparametric learning and GPU algorithm	Sections 4 and 7.1	Published in Transportation Research Part B, 209, 103463 (2026). DOI: 10.1016/j.trb.2026.103463
Inverse Learning of Distributional User Equilibrium with Uncertainty Quantification	Latent equilibrium distribution and CVaR decisions	Sections 5 and 7.3	SSRN working paper (2026). DOI: 10.2139/ssrn.6878997
Calibrated City-Scale Traffic Demand via Constrained, Context-Aware Language Generation	Feasible trip generation and count-calibrated OD demand	Sections 6 and 7.2	SSRN manuscript (2026). DOI: 10.2139/ssrn.7288370

Technical outputs documented in the studies include the convex-basis formulation, parallel block-coordinate algorithm, distributional-equilibrium pipeline, constrained traffic-language representation, regularized ODME procedure, and experimental demand and evaluation products. Publication of a method does not establish public release of its software, model weights, or data. Release status and access restrictions require separate project records.

10.2 Outcomes

The demonstrated outcomes are methodological and analytical: scalable equilibrium learning with empirical predictive improvement, broader candidate demand support and improved count fit, and explicit mean–tail policy trade-offs under synthetic uncertainty. Agency deployment, realized congestion reduction, and completion of an integrated diagnostic prototype are not established by these experiments.

10.3 Impacts and beneficiaries

The principal near-term beneficiaries are transportation modelers and researchers who need interpretable ways to use large trajectory datasets without assuming that sampled probes are complete demand. The methods can help regional planning agencies and consultants build scenario demand, estimate context-sensitive relationships, and identify where data or model structure is insufficient. The broader impact is a pathway from passively collected trajectories to structured, uncertainty-aware planning models.

11. Challenges and Lessons Learned

11.1 Technical challenges

- Partial observation and identifiability: link counts and sampled trajectories do not uniquely determine OD demand or cost functions. Regularization, context variation, and held-out tests are essential.
- Scale: city networks create large feasible sets and many context-specific constraints. Multiconvex decomposition and active-link sets reduce the problem, but performance depends on GPU memory and network sparsity.
- Nonconvexity: alternating inverse problems and neural latent models can depend on initialization. Multiple starts, validation, and transparent stopping rules are needed.
- Distribution shift: probe penetration, sensor availability, weather, and travel behavior may change between training and application. Context features help only for shifts that are measured and represented.
- Generated-demand evaluation: matching marginal trip distributions does not guarantee correct network dynamics. Independent counts and simulator-based temporal validation address different parts of the problem.
- Uncertainty decomposition: empirical variability combines behavior, demand, incidents, sampling, and error. A flexible latent distribution can represent variation without uniquely explaining its cause.

11.2 Lessons for project execution

A useful lesson is to define validation products at the start of each work package. The inverse-learning and demand studies report distinct networks and metrics, which is appropriate for their research goals but complicates system-level integration. A shared evaluation protocol covering common days, sensors, network version, and data splits would make it easier to assess whether improvements compose across modules.

A second lesson is to distinguish a component improvement from an operational outcome. Better count fit does not establish accurate queues, and better tail-risk performance does not guarantee lower average total travel time. A staged evaluation should establish each link from observations to model, from model to policy, and from policy to field outcomes.

12. Performance Indicator Synopsis

12.1 Documented research products

This report documents four research studies: two published journal articles and two working papers cited using SSRN manuscripts. The studies include two distinct Ann Arbor applications, public trajectory-generation benchmarks, and controlled equilibrium experiments. These are descriptions of research products, rather than counts of separately funded deliverables.

12.2 Indicators requiring administrative records

The workbook records one innovation or technology beneficial to U.S. DOT priorities and two students or professionals educated on these innovations or technologies. The research participation indicator records six graduate students. The financial-support indicator separately records two doctoral students supported by the grant, and the degree-completion indicator records one doctoral degree awarded to a supported student. Part I also records one graduate transportation course and one doctoral degree program utilizing grant funds.

Part II records two published journal articles, consistent with the two Transportation Research Part B articles listed in Table 17. Dissemination activities include four keynote speeches and eight conference presentations. Engagement activities comprise seven engagements involving seven organizations and 250 individuals. The workbook also records one related research project funded outside UTC and matching-fund sources, with a value of $82,117, together with one national and one international leadership position. These entries summarize reported activities and research capacity; they do not measure changes in traffic conditions.

For technology transfer, Part II records one technology, procedure, policy, standard, or design practice influenced by the research and applied or commercialized by stakeholders.

References

Aswani, A., Shen, Z.-J. M., & Siddiq, A. (2018). Inverse optimization with noisy data. Operations Research, 66(3), 870–892.

Beckmann, M., McGuire, C. B., & Winsten, C. B. (1956). Studies in the economics of transportation. Yale University Press.

Cambridge Systematics, Inc. (2010). Travel model validation and reasonableness checking manual, second edition (FHWA-HEP-10–042). Federal Highway Administration.

Cascetta, E. (1984). Estimation of trip matrices from traffic counts and survey data: A generalized least squares estimator. Transportation Research Part B: Methodological, 18(4–5), 289–299.

Daxberger, E., Kristiadi, A., Immer, A., Eschenhagen, R., Bauer, M., & Hennig, P. (2021). Laplace redux - Effortless Bayesian deep learning. Advances in Neural Information Processing Systems, 34, 20089–20103.

Dowling, R., Skabardonis, A., & Alexiadis, V. (2004). Traffic analysis toolbox, Volume III: Guidelines for applying traffic microsimulation modeling software (FHWA-HRT-04–040). Federal Highway Administration.

Fosgerau, M., Paulsen, M., & Rasmussen, T. K. (2022). A perturbed utility route choice model. Transportation Research Part C: Emerging Technologies, 136, 103514.

Gal, Y., & Ghahramani, Z. (2016). Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. Proceedings of the International Conference on Machine Learning, 1050–1059.

Haydari, A., Chen, D., Lai, Z., Zhang, M., & Chuah, C.-N. (2026). MobilityGPT: Enhanced human mobility modeling with a GPT model. IEEE Transactions on Intelligent Transportation Systems, 27(1), 1681–1694.

Keshavarz, A., Wang, Y., & Boyd, S. (2011). Imputing a convex objective function. 2011 IEEE International Symposium on Intelligent Control, 613–619.

Lakshminarayanan, B., Pritzel, A., & Blundell, C. (2017). Simple and scalable predictive uncertainty estimation using deep ensembles. Advances in Neural Information Processing Systems, 30.

Li, P., Liu, Z., & Yin, Y. (2026). Inverse learning of distributional user equilibrium with uncertainty quantification. *SSRN*. https://doi.org/10.2139/ssrn.6878997

Liu, Z., & Yin, Y. (2026). Constructing transportation network equilibrium models from empirical data: An inverse learning perspective. *Transportation Research Part B: Methodological, 214*, 103597. https://doi.org/10.1016/j.trb.2026.103597

Liu, Z., Yin, Y., Bai, F., & Grimm, D. K. (2023). End-to-end learning of user equilibrium with implicit neural networks. Transportation Research Part C: Emerging Technologies, 150, 104085.

Liu, Z., Yin, Y., Lin, X., & Wang, Z. (2026). Large-scale inverse learning of user equilibrium via multiconvex optimization. *Transportation Research Part B: Methodological, 209*, 103463. https://doi.org/10.1016/j.trb.2026.103463

Lopez, P. A., Behrisch, M., Bieker-Walz, L., Erdmann, J., Flötteröd, Y.-P., Hilbrich, R., Lücken, L., Rummel, J., Wagner, P., & Wießner, E. (2018). Microscopic traffic simulation using SUMO. Proceedings of the 21st IEEE International Conference on Intelligent Transportation Systems.

Ritter, H., Botev, A., & Barber, D. (2018). A scalable Laplace approximation for neural networks. International Conference on Learning Representations.

Rockafellar, R. T., & Uryasev, S. (2000). Optimization of conditional value-at-risk. Journal of Risk, 2(3), 21–41.

Rockafellar, R. T., & Uryasev, S. (2002). Conditional value-at-risk for general loss distributions. Journal of Banking & Finance, 26(7), 1443–1471.

Van Zuylen, H. J., & Willumsen, L. G. (1980). The most likely trip matrix estimated from traffic counts. Transportation Research Part B: Methodological, 14(3), 281–293.

Wang, Z., Song, X., Han, H., Wang, X., Jerome, Z., & Liu, H. X. (2026). Calibrated city-scale traffic demand via constrained, context-aware language generation. SSRN. https://doi.org/10.2139/ssrn.7288370

Wardrop, J. G. (1952). Road paper: Some theoretical aspects of road traffic research. Proceedings of the Institution of Civil Engineers, 1(3), 325–362.

Wegmann, F., & Everett, J. (2016). Minimum travel demand model calibration and validation guidelines for the State of Tennessee.

Yang, H., Meng, Q., & Bell, M. G. H. (2001). Simultaneous estimation of the origin-destination matrices and travel-cost coefficient for congested networks in a stochastic user equilibrium. Transportation Science, 35(2), 107–123.