CENTER FOR CONNECTED AND AUTOMATED TRANSPORTATION

Final Report 88
November 2025



End-to-End Learning Framework for Transportation Network Equilibrium Modeling

Zhichen Liu
Yafeng Yin

UNIVERSITY OF MICHIGAN

CENTER FOR CONNECTED AND AUTOMATED TRANSPORTATION

Report No 88

November 2025

Project Start Date: 5/28/2024

Project End Date: 10/27/2025

End-to-End Learning Framework for Transportation Network Equilibrium Modeling

Zhichen Liu

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. 221183NI 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 APA Format Citation:
Liu, Z & Yin, Y. (2025). End-to-End Learning Framework for Transportation Network Equilibrium Modeling. Final Report.
DOI: https://doi.org/10.7302/27624

Contacts

For more information:

PI Name: Yafeng Yin
University: University of Michigan
Address: 2120 GG Brown, Ann Arbor, Michigan
Phone Number: (734) 764-8249
Email Address: yafeng@umich.edu
Web Address: https://limos.engin.umich.edu/

CCAT
University of Michigan Transportation Research Institute
2901 Baxter Road
Ann Arbor, MI 48152
uumtri-ccat@umich.edu
(734) 763-2498

CENTER FOR CONNECTED AND AUTOMATED TRANSPORTATION

Technical Report Documentation Page

1. Report No. CCAT Final Report 88	**2. Government Accession No.** Leave blank – not used	**3. Recipient's Catalog No.** Leave blank - not used
4. Title and Subtitle End-to-End Learning Framework for Transportation Network Equilibrium Modeling DOI: https://doi.org/10.7302/27624		**5. Report Date** September 1, 2025 **6. Performing Organization Code** Enter any/all unique numbers assigned to the performing organization, if applicable.
7. Author(s) Zhichen Liu, Ph.D.: https://orcid.org/0000-0001-6178-9883/ Yafeng Yin, Ph.D.: https://orcid.org/0000-0003-3117-5463/		**8. Performing Organization Report No.** Enter any/all unique alphanumeric report numbers assigned by the performing organization, if applicable.
12. Performing Organization Name and Address Center for Connected and Automated Transportation University of Michigan Transportation Research Institute 2901 Baxter Rd, Ann Arbor, MI 48109 University of Michigan Civil and Environmental Engineering 2350 Hayward Street, Ann Arbor, MI 48109		**10. Work Unit No.** **11. Contract or Grant No.** Contract No. 2022-0433 Z7 Project No. 221183NI
12. Sponsoring Agency Name and Address Michigan Department of Transportation (MDOT) Research Administration 8885 Ricks Road , P.O. Box 33049, Lansing, Michigan 48909 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 (May 2024 – October 2025) **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 (OST-R) University Transportation Centers (UTC) program.		

UMTRI · The University of Akron · Central State University · Illinois Center for Transportation · University of Minnesota · Northwestern · Purdue University · Washtenaw Community College · Wisconsin – University of Wisconsin-Madison

CENTER FOR CONNECTED AND AUTOMATED TRANSPORTATION

16. Abstract

This project introduces an end-to-end framework for constructing integrated traffic network equilibrium models, which can serve as lightweight travel demand models, directly from multi-day aggregate traffic state observations. Unknown components on both the supply and demand sides are parameterized with computational graphs and embedded in a variational inequality to enforce user equilibrium conditions. The approach flexibly incorporates model-based, model-free (e.g., neural networks), or hybrid components and calibrates unknown parameters by minimizing discrepancies between observed and estimated traffic states.

The framework was validated through numerical experiments with synthetic networks and empirical data from the Ann Arbor network. It demonstrated strong predictive accuracy for link flows under network changes and resilience to incomplete or noisy data. In the Ann Arbor case study, the framework reduced link travel time prediction error from 83.6% in the benchmark model to 34.3% and successfully captured behavioral patterns such as reduced travel on weekends and snow days. The results also show that the framework has strong potential for prescribing optimal improvement plans to reduce congestion, as it integrates learning and optimization into a single data-to-decision pipeline.

Overall, the proposed end-to-end framework enables automated construction and calibration of transportation network equilibrium models using cross-source data and supports the evaluation and prescription of strategies such as capacity expansion or congestion pricing. In practice, the framework can improve the efficiency of transportation planning, reduce both capital and operational costs, and guide more effective resource allocation to maximize public benefit.

17. Key Words	**18. Distribution Statement**		
Travel demand modeling, traffic network equilibrium models, user equilibrium, end-to-end learning and optimization	No restrictions.		
19. Security Classif. (of this report)	**20. Security Classif. (of this page)**	**21. No. of Pages**	**22. Price**
Unclassified	Unclassified	97	Leave blank – not used

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

UMTRI | The University of Akron | Central State University | Illinois Center for Transportation | University of Minnesota | Northwestern | Purdue University | Washtenaw Community College | Wisconsin – University of Wisconsin-Madison

Table of Contents

List of Tables 3

List of Figures 4

Executive Summary 6

1 Introduction 8

1.1 Background **8**

a. Objectives 10

b. Scope 11

1.2 Statement of Hypotheses **12**

2 Literature Review 14

2.1 Review of Previous Research **14**

2.1.1 Implicit Layer: Expressivity and Generalization 14

2.1.2 Auto-Differentiation-Based Algorithms for Bi-Level Optimization 15

2.2 Summary of the State-of-the-Art **17**

3 Methodology 19

3.1 End-to-End Learning Framework **19**

3.1.1 Neural-Network-Based User Equilibrium 19

3.1.2 Learning Formulation 23

3.1.3 Neural Network Architecture 24

3.1.4 Training 29

3.2 Feasibility Analysis **35**

3.3 End-to-End Optimization **39**

3.4 Equipment **40**

4 Validation: Synthetic Data 41

4.1 Example 1: Learn Demand Component on Braess **41**

4.2 Example 2: Learn Demand Component on Sioux Falls **44**

4.3 Example 3: Learn Behavior Component on Sioux Falls **47**

4.4 Example 4: Learn Demand and Supply component on Chicago Sketch **58**

5 Validation: Empirical Data from Ann Arbor 61

5.1 Data Processing **61**

5.2 Data Analysis **65**

Link-level 66

5.2.1 OD-level ... 68
5.2.2 Activity-chain-level ... 75
5.2.3 Network-level ... 75
5.3 Experiment Settings ... 78
5.4 Presentation of Results ... 80
6 Discussion ... 87
6.1 Validity of hypotheses ... 87
6.2 Factors affecting the results ... 88
6.3 Implications ... 89
6.4 Limitations and future work ... 90
7 Conclusions ... 91
8 Bibliography ... 95

List of Tables

Table 1 Summary of trade-offs. Results marked with an asterisk (*) are suggested but not explicitly proven due to the complexity of finding global optima in MPEC. 38
Table 2 WMAPE under different scenarios 43
Table 3 WMAPE under different training settings 48
Table 4 MAPE of different network equilibrium models. 51
Table 5 MAPE of proposed forward algorithms. 54
Table 6 WMAPE under different training settings. 60
Table 7 Observation statistics by number of OD pairs. 79
Table 8 Model performance across different scenarios and settings. Link time MAPEs are reported as percentages with standard deviations in parentheses. 81
Table 9 Parameters of link performance function under different settings. 82
Table 10 Non-zero parameters of the inverse demand function under different settings. 83

List of Figures

Figure 1 Illustration of traditional four-step travel demand modeling process 8
Figure 2 Illustration of "end-to-end" transportation network equilibrium modeling and optimization 10
Figure 3 Illustration of Attribute Net 26
Figure 4 Illustration of the end-to-end learning framework 32
Figure 5: Framework performances with different forward iterations under (a) ITD and (b) IMD. 44
Figure 6 Framework performances using different backward method with (a) N = 1, (b) N = 10, and (c) N = 50 45
Figure 7 (a) Testing optimality gap and (b) training optimality gap MSE with respect to epochs. 46
Figure 8 (a) Training link flow MSE and (b) testing link flow WMAPE with respect to epochs. 47
Figure 9 Training process of different forward algorithms. 52
Figure 10 Performances of different backpropagation methods under (a) base, (b) uncongsted, (c) congested demand. 55
Figure 11 Effects of spectral normalization under (a) base, (b) uncongested, and (c) congested demand. 55
Figure 12 Model performances with different sensor coverage rates under (a) base, (b) uncongested, and (c) congested demand. 56
Figure 13 Model performances with demand noises under (a) base, (b) uncongested, and (c) congested demand. 57
Figure 14 Effects of inaccurate feasible path sets under (a) base, (b) uncongested, and (c) congested demand. 58
Figure 15 Data processing pipeline. 62
Figure 16 (Left) raw trajectory, (middle) matched trajectory using Wang et al. (2023), and (right) filtered result. Color indicates time progression, with darker colors representing earlier timestamps. 63

Figure 17 (Left) Raw trip with a detour to residential roads (red circle); (middle) first subtrip; (right) second subtrip. Color indicates time progression, with darker colors representing earlier timestamps. 64
Figure 18 Ann Arbor network. Traffic analysis zones are shaded by area type; darker colors correspond to higher area type indices..... 65
Figure 19 (a) Number of observed links, (b) average stop delay per link, and (c) total system travel time on weekdays and weekends. 66
Figure 20 (a) Correlations between link-level features and observed travel behavior. (b) Comparison of snow-related reductions in link travel time for highway and non-highway segments..... 67
Figure 21 (a) Total number of observed trips, and (b) number of unique OD pairs. 68
Figure 22 Spatial distribution of trip attraction (top row) and production (bottom row) on weekdays (left) and weekends (right). Darker colors indicate higher aggregated trip counts. 69
Figure 23 Temporal distribution of (a) internal, and (b) external trips. 70
Figure 24 Mean number of OD observations for top internal and external trips by weekday 71
Figure 25 Mean number of observed dates for top internal and external trips by weekday 72
Figure 26 Correlation between OD demands and origin/destination TAZ characteristics. 73
Figure 27 Network-level equilibrium gap..... 74
Figure 28 Mean equilibrium gap for internal and external trips by day type: (top row) weekdays, (bottom row) weekends..... 76
Figure 29 Correlation between chain travel demand and TAZ-level features on weekdays and weekends. 77
Figure 30 Correlation between network travel patterns and context features on (a) weekday 78
Figure 31 Differences in observed trips between (a) snow and non-snow days, and (b) weekdays and weekends. 79
Figure 32 Candidate links for capacity expansion 85
Figure 33 System travel time before and after design implementation 86
Figure 34 Recommended investment levels across candidate links 86

Executive Summary

The current travel demand models, whether developed using the conventional trip-based four-step modeling approach or the state-of-the-art activity-based modeling approach, are time-consuming and costly to establish and update. They are typically revised only every five to ten years and often fail to accurately capture current travel behavior.

To address these challenges, this project introduces a unified, end-to-end learning framework for constructing integrated transportation network equilibrium models, which can serve as *lightweight* travel demand models, directly from empirical traffic data. The framework learns both supply- and demand-side model components from multi-day aggregate traffic state observations. Unknown components are parameterized with computational graphs and embedded in a variational inequality to enforce user equilibrium conditions. Each component can be model-based, model-free (e.g., neural networks), or hybrid. By minimizing the difference between estimated and observed traffic states, the framework simultaneously calibrates unknown supply- and demand-side parameters.

To validate robustness and effectiveness of the proposed framework, numerical experiments were conducted using both synthetic data from various networks and empirical data from the Ann Arbor network. The framework demonstrated strong predictive accuracy for link flows under changes in network topology and demonstrated resilience to incomplete data and noisy inputs. In the Ann Arbor case study, the framework reduced prediction error for link travel time from 83.6% in the benchmark model to 34.3% and successfully captured variations in traffic patterns, such as reduced travel on weekends and snow days.

A rigorous feasibility analysis of the end-to-end framework further identifies three potential sources of error. Expressivity risk arises when imperfect prior knowledge prevents accurate representation of real-world travel behavior. Generalization risk occurs when models trained on limited data perform poorly on unseen inputs. Optimization risk stems from the complexity of solving the inverse optimization

problem. The analysis shows that, with sufficiently large neural networks and adequate data, the end-to-end framework achieves improved performance and mitigates these prediction errors.

Another key advantage of the framework is its integration of learning and optimization into a single data-to-decision pipeline. Applied to the Ann Arbor network, the framework demonstrated potential for prescribing improvement plans and polices to alleviate congestion. It can serve as a decision-support tool for policymakers considering improvement strategies such as capacity expansion or congestion pricing. Automated implementations of the framework could enable transportation agencies to plan and operate traffic networks more efficiently, lowering both capital and operational costs. By guiding more informed resource allocation, the framework helps policymakers avoid unnecessary infrastructure investments and maximize public benefit.

1 Introduction

1.1 Background

Travel demand modeling is a fundamental tool in transportation planning, providing a structured approach to forecast whether, where, and how people travel within a region. At its core, it seeks to represent travelers' decisions, such as whether to travel, choice of destination, mode, departure time, and route based on land use, demographics, and transportation system characteristics. These forecasts form the basis for evaluating the impacts of policy interventions, infrastructure investments, and technological changes on mobility, accessibility, and equity.

Traditionally, travel demand has been analyzed through a four-step modeling process. The first step, trip generation, estimates the number of trips produced by and attracted to each traffic analysis zone (TAZ). The second step, trip distribution, applies a destination choice model to determine how these trips are spatially distributed, producing demand between origin-destination (OD) pairs. The third step, mode choice, allocates OD-level trips among available travel modes, yielding the number of vehicle trips. Finally, the fourth step, traffic assignment, describes travelers' route choices to distribute OD trips across the network, resulting in link-level flows and measures of network performance. These flow estimates provide benchmarks for assessing the effectiveness of proposed system improvements.

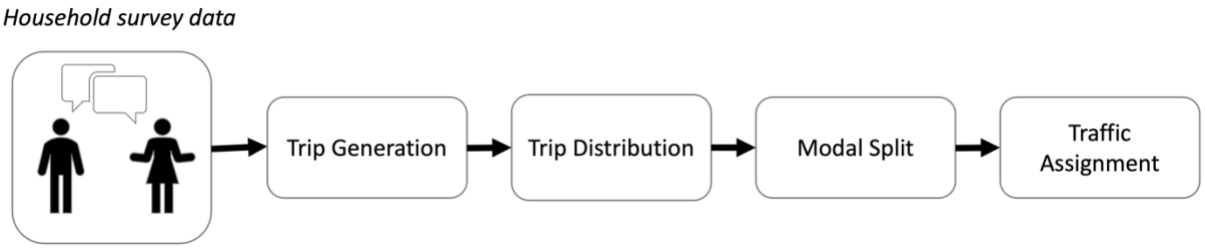


Figure 1 Illustration of traditional four-step travel demand modeling process

While the four-step model has long been the workhorse of transportation planning, its aggregate and sequential structure limits behavioral realism. To address these shortcomings, the field advanced to activity-based modeling (ABM). Unlike trip-based

approaches, ABMs view travel demand as derived from individuals' and households' need to participate in activities at different times and locations. By explicitly modeling daily activity patterns, scheduling, and interdependencies among trips, ABMs offer a richer behavioral foundation. This allows them to better capture heterogeneity across population groups, sensitivity to land use and policy changes, and temporal dynamics such as peak spreading.

Despite their differences, both the four-step and activity-based approaches share a common limitation: they rely heavily on extensive data collection efforts such as household travel surveys. Such efforts are costly, time-consuming, and resource-intensive, making it difficult to update established travel demand models frequently. As a result, metropolitan planning organizations (MPOs) often revise their regional travel demand models only once every five to ten years. This infrequent updating limits the ability of traditional models to reflect evolving travel behavior, rapid technological change, and emerging mobility options.

In recent years, however, advancements in connectivity and sensing technologies have made aggregate traffic state observations increasingly available. These states include traffic volumes, speeds, travel times, and even route choice probabilities. Data sources now extend beyond traditional loop detectors to include radar sensors, Bluetooth and Wi-Fi tracking, GPS-enabled mobile devices, and high-resolution vehicle trajectory data from connected and automated vehicles. The growing availability of these emerging datasets, combined with recent advances in artificial intelligence, creates new opportunities to rethink travel demand modeling.

Building on these opportunities, this project introduces a unified, end-to-end learning framework for constructing integrated transportation network equilibrium models that serve as lightweight yet behaviorally informed travel demand models, derived directly from empirical traffic observations. The framework simultaneously learns OD travel demand functions and route choice preferences while refining the link performance function, which relates traffic volume to travel time. Once calibrated with empirical data, it can not only reproduce observed traffic conditions with higher fidelity but also

prescribe optimal infrastructure improvements or policy interventions, offering decision-makers actionable, evidence-based insights.

Unlike conventional approaches that rely on costly and infrequent surveys, the proposed framework leverages passively collected traffic data to enhance behavioral realism and predictive accuracy at a fraction of the cost. By minimizing dependence on resource-intensive local data collection, it effectively democratizes access to advanced traffic network diagnostics, enabling even resource-constrained agencies to benefit from sophisticated modeling capabilities. The resulting system is a practical, deployable, and adaptive tool for data-driven analysis, supporting both short-term traffic management and long-term planning. More broadly, this framework represents a new generation of travel demand modeling that unifies empirical data, machine learning, and equilibrium theory into a single data-to-decision pipeline, laying the foundation for more responsive, cost-effective, and behaviorally grounded transportation planning.

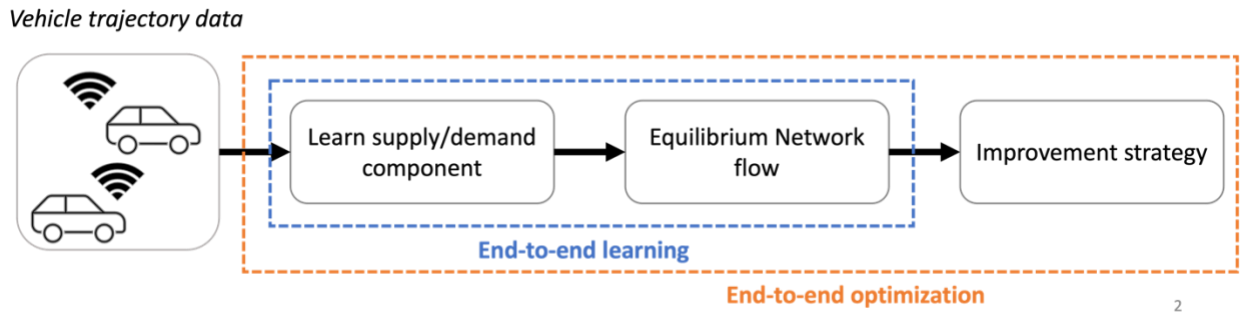


Figure 2 Illustration of "end-to-end" transportation network equilibrium modeling and optimization

a. Objectives

The primary objective of this project is to establish a unified end-to-end framework for transportation network equilibrium modeling. At its core, this framework leverages deep learning to jointly model supply and demand components while directly estimating equilibrium flow distributions from empirical data. A second objective of this project is to rigorously demonstrate and validate the framework through case studies. Building on this validation, the project identifies strategic enhancements that optimize system performance. Finally, the project seeks to establish future research directions and

provide practical recommendations for MDOT, SEMCOG, and Washtenaw Area Transportation Study (WATS) MPO, thereby laying the groundwork for continued development of advanced, data-driven modeling practices.

b. Scope

The scope of this project is structured around five major tasks.

Task 1: Developing a research management plan.

This task focuses on creating a comprehensive plan to guide technical and administrative activities. It includes coordination among the research team, project manager, and advisory panel, as well as scheduling regular meetings with MDOT staff to ensure timely progress, effective communication, and accountability.

Task 2: Developing a unified framework for end-to-end learning of network equilibrium.

This task aims to advance our prototype end-to-end learning framework to simultaneously model supply and demand components of network equilibrium. The framework employs computational graphs with learnable parameters to approximate unknown supply and demand relationships, embedding them within a variational inequality formulation to enforce user equilibrium conditions. By minimizing discrepancies between observed and estimated travel times, parameters on both the supply and demand sides are updated jointly. This approach represents the first integrated calibration of both components within a data-driven equilibrium framework, unifying model-based and model-free elements for greater flexibility. A theoretical analysis will also be conducted to assess the feasibility of this approach.

Task 3: Demonstrating the framework through a case study of Ann Arbor using GM trajectory data.

In this task, we will apply GM's vehicle trajectory data to enhance the behavioral realism and predictive accuracy of the planning model maintained by WATS. The research team

will design algorithms to address data sparsity and subsampling issues inherent in GM's data.

Task 4: Prescribing improvement strategies.

This task investigates efficient solution algorithms to prescribe system improvement strategies using the end-to-end framework. These algorithms will be demonstrated in the Ann Arbor case study to evaluate their effectiveness in guiding transportation planning decisions.

Task 5: Recommending next steps for future research and development.

The final task will outline future research directions and propose an operational framework for potential adoption by MDOT and MPOs. This step will help advance the long-term development of a practical modeling process.

1.2 Statement of Hypotheses

Before stating the specific hypotheses, we first outline the conceptual premises that underlie this project: travel demand forecasting, the notion of user equilibrium, and the idea of end-to-end learning.

Travel Demand Forecasting and Choice Preferences

Travel demand forecasting seeks to predict whether, where and how people will travel in the future. Travelers' decisions are governed by their choice preferences, which reflect underlying trade-offs among time, cost, convenience, and other attributes. While contextual factors evolve, the structural form of such preferences is generally stable over forecasting horizons. By analyzing observed behaviors across these choice dimensions, the proposed framework aims to infer the latent preferences that shape travel demand. Once identified, these preferences provide a behavioral foundation that is assumed to persist into the forecasting year, enabling the prediction of future travel patterns under new conditions (e.g., changes in land use, pricing, or infrastructure).

Equilibrium as Benchmark and Approximation

Another premise is the role of equilibrium states in transportation network modeling. Although exact equilibrium is rarely observed in practice, because of day-to-day variability, bounded rationality, or information asymmetries, empirical traffic states tend to cluster around equilibrium-like conditions. For planning, the notion of equilibrium remains indispensable: it provides a consistent, reproducible benchmark against which different policy or infrastructure scenarios can be fairly compared. Without such a benchmark, evaluations risk being inconsistent or misleading. Accordingly, this project adopts a data-driven approach to learn an equilibrium state that both reflects observed traffic conditions and preserves the consistency required for planning analyses.

End-to-End Learning Perspective

Finally, the project embraces an end-to-end learning paradigm. Rather than decomposing forecasting into separate, sequential modules (e.g., trip generation, distribution, mode split, assignment), the framework integrates demand and supply components within a unified learning architecture. Unknown model elements are parameterized and jointly estimated by fitting observed traffic states, while equilibrium constraints are embedded to ensure behavioral and network consistency. This end-to-end approach offers both flexibility, accommodating model-based, data-driven, or hybrid components, and rigor, as the learned equilibrium states serve as both predictive tools and planning benchmarks.

The project is guided by the following hypotheses:

1. **End-to-end learning improves predictive accuracy.** By jointly modeling supply and demand components from empirical trajectory and auxiliary data, the proposed framework will achieve more accurate predictions of network equilibrium flows compared with traditional four-step models.

2. **Data-driven integration enhances behavioral realism.** Incorporating vehicle trajectory data and other emerging data sources will allow the framework to better capture traveler behaviors, including route choice variability and responses to congestion.

3. **Unified calibration reduces error propagation.** Calibrating demand and supply simultaneously within one framework will reduce cumulative errors that typically arise from independent estimation in traditional models.

4. **Practical applicability to planning organizations.** The framework can be validated and adapted to support planning decisions within MDOT, SEMCOG, and WATS MPO, enabling data-driven policy and system improvement strategies.

These hypotheses frame the methodological innovations and expected contributions of the project, while providing the foundation for empirical validation through the case study.

2 Literature Review

2.1 Review of Previous Research

2.1.1 Implicit Layer: Expressivity and Generalization

The foundation of the proposed end-to-end framework lies in parameterizing unknown model components with neural networks and embedding user equilibrium conditions as an implicit layer. The term implicit is used because the output of this layer is defined implicitly: it cannot be computed through explicit rules as in conventional neural networks (Travacca et al., 2020). Instead, it is obtained by solving a fixed-point problem. The implicit layer was first proposed by Bai et al. (2019) and has since been applied to domains such as power flow prediction (Fioretto et al., 2020) and auction mechanism design (Feng et al., 2018). In this report, the term “neural network” refers broadly to functions representable by directed acyclic computational graphs, where vertices represent differentiable functions and edges represent function composition. Accordingly, the terms “neural networks” and “computational graphs” are used interchangeably.

A central question in neural network research concerns their expressivity, or the ability

to approximate a wide class of functions. Standard neural networks, which stack linear and nonlinear activation layers and compute outputs explicitly, are known to have universal approximation capabilities. Foundational results (Kidger and Lyons, 2020; Pinkus, 1999) show that, with sufficient parameters, such networks can approximate any continuous function to arbitrary precision. The end-to-end framework, however, employs an implicit layer, and its expressivity is less well understood. Unlike explicit networks, the output of an implicit layer is defined by the solution to a fixed-point problem, raising questions about whether such layers can universally approximate equilibrium states, which may require infinitely deep standard networks to represent exactly. Recent results (Bai et al., 2019) indicate that implicit layers can replicate finite-depth standard networks, but the question of universal equilibrium approximation remains open.

Closely related to expressivity is generalization, the ability of trained networks to perform well on unseen data. Standard networks are known to generalize, though performance typically degrades with depth (Golowich et al., 2018). In the proposed framework, the implicit layer's effective depth is determined by the number of forward iterations needed to reach equilibrium. Existing theory suggests that because this depth may be infinite, implicit layers could generalize poorly. Yet, empirical studies demonstrate that implicit layers often generalize well, contradicting theoretical expectations. This discrepancy highlights the need for new theory. Gao and Gao (2022) show that implicit layers can generalize when their output dimension grows with the number of training samples. However, in the end-to-end framework, output dimensions are tied to network topology and cannot be freely adjusted. This project provides the first demonstration that implicit layers can accurately approximate "well-posed" equilibria and generalize effectively to unseen data, even without adjustable output dimensions.

2.1.2 Auto-Differentiation-Based Algorithms for Bi-Level Optimization

The end-to-end framework requires solving a bi-level optimization problem, or more generally a Mathematical Program with Equilibrium Constraints (MPEC), using gradient descent. This process necessitates differentiation of equilibrium states with

respect to parameters, a challenge known as implicit differentiation or equilibrium sensitivity analysis.

Several approaches to implicit differentiation have been proposed in the transportation literature. Tobin and Friesz (1988) selected non-degenerate extreme points of equilibrium path flow sets, while Yang and Huang (2005) identified linearly independent subsets of used paths. Both approaches rely on matrix inversion, which scales quadratically with problem dimension. Patriksson (2004) instead computed directional gradients by repeatedly solving auxiliary linear variational inequalities (VIs), but this method also suffers from scalability issues.

More recently, automatic differentiation (AD) has enabled new algorithms for implicit gradient approximation when equilibrium constraints are expressed as lower-level optimization problems, thereby reducing MPECs to bi-level formulations. Two notable approaches are iterative differentiation (ITD) and implicit differentiation (IMD). ITD tracks the optimization trajectory of the lower-level problem and backpropagates through it to approximate gradients. However, storing or unrolling long optimization trajectories is computationally burdensome. IMD avoids this by using the implicit function theorem, iteratively solving an auxiliary fixed-point problem to approximate gradients. Originally developed for hyperparameter optimization and meta-learning (Franceschi et al., 2018), both methods have demonstrated local convergence and efficiency in diverse applications (Ghadimi and Wang, 2018; Ji et al., 2021).

This project investigates the efficiency and convergence of AD-based algorithms for solving MPECs with constrained VIs, which present greater challenges than standard bilevel problems. Prior studies have handled equilibrium constraints in different ways: Liu et al. (2023) used decoupled projection for link-based constraints, while Li et al. (2022) employed mirror descent for path-based constraints, proving asymptotic convergence of a modified auto-differentiation-based method. Other heuristic approaches ignore equilibrium constraints altogether, replacing the equilibrating process with a single-step network loading (Guarda et al., 2023; Wu et al., 2018). This raises the open question of whether enforcing user equilibrium confers advantages in solving

MPECs. This project addresses this gap by demonstrating convergence rates of AD-based gradient descent under explicit equilibrium constraints, highlighting the role of the equilibrating process in achieving local convergence.

2.2 Summary of the State-of-the-Art

Research on travel pattern prediction from empirical observations has developed along two main disciplinary lines. The traditional equilibrium-based approach, rooted in travel demand modeling, views travel choices as outcomes of a congestion game and predicts future demand as an equilibrium state. In contrast, recent supervised learning-based approaches frame travel demand forecasting as a prediction task using machine learning. After years of parallel development, these two perspectives increasingly converge, now facing common challenges.

Equilibrium-Based Approach

In planning and "what-if" analyses, modelers require a benchmark state for comparing design or policy alternatives. Equilibrium has long been adopted for this purpose because of its independence from initial conditions, behavioral grounding, and mathematical tractability. Traditional parametric network modeling thus relies on equilibrium to calibrate and forecast flows. For example, Yang et al. (2001) used a logit-based stochastic user equilibrium model to estimate OD demands and dispersion parameters from link flow observations.

Significant efforts have focused on improving behavioral realism in route choice models, thereby enhancing forecast accuracy. Examples include dynamic dispersion parameters calibrated from empirical data (Wang et al., 2016) and advanced behavior models (Guarda and Qian, 2022). However, increased realism often comes at the cost of higher computational complexity. Recent studies have explored incorporating machine learning tools, such as computational graphs and AD, to improve calibration and efficiency.

Early efforts include encoding trip generation, distribution, and path-based logit loading with layered computational graphs for OD calibration (Ma et al., 2020; Wu et al., 2018).

Other work integrates neural networks with discrete choice models, demonstrating that neural networks can enhance both realism and interpretability in travel choice analysis (Sifringer et al., 2020). On the supply side, Lu et al. (2023) introduced physics-informed neural networks for calibrating queuing profiles from multi-source data. Nevertheless, most existing studies calibrate supply and demand components independently, neglecting their interactions.

More recent work extends computational-graph-based calibration to account for traveler interactions in routing games. Guarda et al. (2023) calibrated both supply and demand components jointly from real-world data by penalizing deviations from user equilibrium in the loss function. In contrast, Li et al. (2020) and Heaton et al. (2021) explicitly enforced equilibrium as fixed points, calibrating unknown components directly from flow and time observations. Despite these advances, most approaches remain parametric, with model forms pre-specified before calibration.

Supervised Learning-Based Approach

In parallel, a growing body of research applies deep learning to forecast traffic flows as a supervised learning problem. Models such as Long Short-Term Memory (LSTM) networks and Spatial–Temporal Graph Convolutional Networks (ST-GCNs) have demonstrated strong performance in predicting short-term flows by capturing complex spatiotemporal dependencies (Yao et al., 2019). However, these models rely on the assumption that training and test data are identically distributed. This assumption fails in long-term "what-if" scenarios, where changes in network topology alter traffic flow distributions (Shen et al.). Although such changes affect network-level flows, it is reasonable to assume that traveler preferences remain relatively stable, even under altered network conditions. Consequently, supervised approaches should focus on learning mappings from input features to stable behavioral parameters, rather than directly mapping inputs to flows. This shift could enhance their ability to generalize across different network topologies and planning scenarios.

To summarize, equilibrium-based and supervised learning-based approaches, once developed in parallel, are now converging toward shared goals and challenges. Both face

fundamental difficulties in accurately learning traveler behavior from limited observations. Travel choices often reflect bounded rationality, and internal decision-making processes remain unobservable. Researchers can only infer behavior from outcomes. This intrinsic limitation of the data complicates the task of behaviorally realistic prediction, underscoring the need for hybrid approaches that integrate behavioral models with modern data-driven methods.

3 Methodology

This section presents the end-to-end learning framework to simultaneously estimate the supply and demand components of a transportation network equilibrium model using empirical data. The approach employs computational graphs with learnable parameters to approximate the unknown supply and demand functions and embeds them in a variational inequality that enforces user equilibrium conditions. By minimizing discrepancies between estimated and observed travel times, the framework updates parameters on both sides concurrently. To our knowledge, this is the first framework that integrates the calibration of supply and demand components in a data-driven transportation network equilibrium model. It unifies model-based and model-free elements, enabling a flexible and adaptive approach to real-world transportation networks. We also provide a theoretical analysis of the feasibility of the proposed framework and end-to-end optimization framework that prescribes optimal network expansion designs based on the learned end-to-end framework.

3.1 End-to-End Learning Framework

3.1.1 Neural-Network-Based User Equilibrium

We consider a case where partial aggregate traffic measures, such as link flow and link time, at peak periods are observable for a long period. The general learning task is to learn OD demand functions, travelers' route choice preferences, and link performance functions from multi-day observations. If prior knowledge is available, some components can be pre-calibrated, and the end-to-end framework only focuses on the remaining components.

Path-based formulation

Mathematically, consider a network $\mathcal{G} = (\mathcal{N}, \mathcal{A})$, where $\mathcal{N}$ and $\mathcal{A}$ are the set of nodes and links. Let $\mathcal{R}$ denote the set of OD pairs. Each OD pair $r \in \mathcal{R}$ is connected by paths that form a finite and nonempty feasible path set $\mathcal{P}_r$. $\mathcal{P}$ represents the set of feasible paths for all OD pairs. Let $x^{[m]}$ be the input features observed on day (sample) m. The input features include traveler characteristics like income, road network attributes like free-flow time, and contextual features like weather and gas price. Input features can vary from day to day (or sample to sample). Throughout the report, the norm denotes the L2 norm, unless otherwise indicated. Superscript m associates sample-dependent variables with the m-th sample.

We propose three continuous functions to approximate the unknown supply or demand-side model components. The parameter of all components will be jointly learned and thus we say all components are parametrized by $\theta \in \Theta$. Each component can be model-based, model-free (e.g., neural networks), or hybrid (e.g., physics-informed neural networks). Therefore θ represents neural network parameters in a model-free or hybrid approach, or parameters of a given functional form in a model-based approach.

We will elaborate on the construction of each component, starting from the supply side. The link performance function τ_θ outputs the link travel time $t^{[m]} \in \mathcal{T}$ as a function of path flow $h^{[m]} \in \mathcal{H}$ and input features, defined as:

$$\tau_\theta : \mathcal{H} \times \mathcal{X} \to \mathcal{T} \tag{1}$$

where the input features $x^{[m]} \in \mathcal{X}$ include contextual features and road network attributes, such as link capacity and free-flow time; the feasible region $\mathcal{H} \subseteq R_+^{|\mathbb{P}|}$ requires path flow to be nonnegative and is the feasible region of link time.

On the demand side, travelers are free to switch paths to improve their utilities. Findings from travel behavior research suggest that travel choice behaviors are much more complicated than just choosing the shortest path. We use the cost function π_θ to describe

the perceived path cost given actual travel time. The cost function π_θ outputs the (perceived) path cost as a continuous function of link time and input features, defined as:

$$\pi_\theta: \mathcal{T} \times \mathcal{X} \to \mathcal{C} \tag{2}$$

where input features include traveler characteristics (e.g., income and travel purpose), route attributes (e.g., number of left turns), and contextual features. The feasible set $\mathcal{C} \subseteq R_+^{|\mathbb{P}|}$ requires path cost as nonnegative.

In addition to route choice, travelers have the freedom to choose travel or not and switch origin and/or destination to improve their utility. We assume the travel demand is upper bounded by a maximum possible demand $q \in R_+^{|\mathbb{R}|}$ and introduce the excess demand as $e^{[m]} = q - \Gamma^\top h^{[m]}$. Here, $\Gamma \in R^{|\mathbb{P}|\times|\mathbb{R}|}$ represents the path-OD incidence matrix and Γ_{pr} equals 1 if path p connects OD pair r and equals 0 otherwise. We use an inverse demand function λ_θ to depict the equilibrium path cost $u^{[m]} \in \mathcal{U}$ as a function of excess demand $e^{[m]} \in \mathcal{E}$ and input features, namely,

$$\lambda_\theta: \mathcal{E} \times \mathcal{X} \to \mathcal{U} \tag{3}$$

where the feasible region of excess demand is $\mathcal{E} = \{e \in R^{|\mathbb{R}|}: 0 \le e \le q\}$ and $\mathcal{U} \subseteq R_+^{|\mathbb{R}|}$ is the feasible region of equilibrium path cost.

Assuming rational travelers try to maximize their own travel utilities, the multi-class user equilibrium (UE) with elastic demand is formulated as a parametric VI, the solution to which is the equilibrium path flow $h^{*[m]}$ and equilibrium excess demand $e^{*[m]}$ for sample m. To simplify notation, we introduce the response variable as $y = (h, e)$ and the generalized cost as $z = (c, u)$. By defining the generalized cost function:

$$F_\theta: \mathcal{Y} \times \mathcal{X} \to \mathcal{Z} \tag{5}$$

where $F_\theta(y, x) = [\pi_\theta^\top(\tau_\theta(h, x), x), \lambda_\theta^\top(e, x)]^\top$.

Figure 3 illustrates the computational-graph-based generalized cost function for path-based elastic UE. Supply and demand-side components are shown in blue and green respectively. The dependence of variables on sample m is omitted to simplify the notation. Each parametrized component can be model-based, model-free, or hybrid. Then the parametric VI in Eq. (5) can be compactly reformulated as:

$$\langle F_\theta(y^{*[m]}, x^{[m]}), y - y^{*[m]} \rangle \geq 0, \qquad \forall y \in \mathcal{Y} \tag{6}$$

To compactly represent the feasible region of the response variable, the feasible region of the response variable becomes $\mathcal{Y} = \{y \geq 0, \bar{\Gamma}^T y = q\}$ where $\bar{\Gamma}$ is augmented path-OD incidence matrix

If we consider a special case where the OD demands are observable for each sample, the proposed framework can handle an inelastic demand setting. Let $q^{[m]}$ be the OD demands and link flows observed on sample m. Then the multi-class UE with inelastic demand for sample m is formulated as a parameterized VI in Eq. (6). In this case, the feasible path flow set becomes sample-dependent, i.e., $\mathcal{H}^{[m]} = \{h \in R^{|\mathbb{P}|}: h \geq 0, \Sigma^\top h = \overline{q}^{[m]}\}$ requires the feasible path flows to be nonnegative and satisfy flow conservation.

Link-based formulation

The parametric VI defined in Eq. (6) requires the knowledge of feasible path set. This is a common assumption for path-based UE formulation and methods for generating the feasible path set are well-developed in the literature (Frejinger et al., 2009). If the modelers believe the path cost is link-additive, the link-based elastic-UE formulation can be used instead.

We introduce OD-specific link flows for OD pair r as and the vectorized OD-specific link flows as $v = \{v_r\}_{r \in \mathcal{R}} \in \mathcal{V} \subseteq R_+^{|\mathbb{A}| \times |\mathbb{R}|}$. In this case, the link performance function becomes:

$$\tau_\theta: \mathcal{V} \times \mathcal{X} \to \mathcal{T}. \tag{7}$$

We slightly abuse the notation of path cost and define the OD-specific link cost $c_r \subseteq$

$R_+^{|\mathbb{A}|}$ with its vectorized form as $c = \{c_r\}_{r\in\mathcal{R}}$. The link-based equilibrium condition for sample m is formulated a following parametric VI, slightly adjust the notation for generalized cost and response variable to bring the link-based and path-based formulations under the same umbrella. For each OD pair *r*, we define the response variable as $y_r = (v_r, e_r)$ with its vectorized form given as $y = \{y_r\}_{r\in\mathcal{R}} \in \mathcal{Y}$. The generalized cost for OD pair r is represented as $z_r = (c_r, u_r)$, and its vectorized form is formulated as $z = \{z_r\}_{r\in\mathcal{R}} \in \mathcal{Z} \subseteq R_+^{(|\mathbb{A}|+\mathbb{1})\times|\mathbb{R}|}$.

To compactly formulate the feasible region for response variable *y*, we introduce the augmented link-node incidence matrix and vectorized demand constraint as follows. For the former, we add a number of $|\mathcal{R}|$ dummy links connecting the origin and destination of each OD pair, with a number of e_r travelers on each dummy link experiencing the equilibrium path cost u_r. Then we represent the augmented link-node incidence matrix including dummy link as $\Lambda \in R^{(|\mathbb{A}|+\mathbb{1})\times|\mathbb{N}|}$ where $\Lambda_{an} = 1$ if link a originates from a and $\Lambda_{an} = -1$ if link a terminates at node n. For each OD pair, we define a vectorized demand constraint, where $d_r \in R_+^{|\mathbb{A}|}$; $d_{rn} = q$ if OD pair r originates at node n and $d_{rn} = -q$ if OD pair r terminates at node n and $d_{rn} = 0$ otherwise. Then the feasible region of the response variable can be compactly formulated as $\mathcal{Y} = \{y \in R^{(|\mathbb{A}|+\mathbb{1})\times|\mathbb{R}|}: y \geq 0, \Lambda^\top y_r = d_r, \forall r \in \mathcal{R}\}$. It is straightforward to validate that both the path-based equilibrium condition in and the link-based equilibrium condition align with the same compact parametric VI in Eq. (6).

3.1.2 Learning Formulation

We consider a smooth loss function $l: \mathcal{Y} \times \mathcal{Y} \to R$ that measures the distance between the estimated equilibrium states and corresponding observations. We also consider a regularization function $r(\theta)$. The training of the end-to-end framework can be formulated as the following MPEC. Each training sample m corresponds to the pair Consider the dataset of M samples, where each data point is drawn i.i.d. from an unknown probability distribution over $\mathcal{X} \times \mathcal{Y}$.

$$\min_{\theta} \sum_{m=1}^{M} l\left(y^{*[m]}, y^{[m]}\right) \; s.t. \; \langle F_\theta(y^{*[m]}, x^{[m]}), y - y^{*[m]} \rangle \geq 0, \qquad \forall y \in \mathcal{Y} \tag{8}$$

The end-to-end framework unifies the parameters of supply and demand-side components, either model-based, model-free or hybrid, into a generalized cost function and jointly learns θ during training.

Remark 1 *If the cost function is independent of input feature x and equals the sum of link travel times and an entropy term, the learning problem will reduce to the logit dispersion parameter calibration problem investigated by Yang et al. (2001). If the equilibrium constraints are removed, the learning problem would directly learn a mapping from the context features x to link flows v. In this case, the problem reduces to neural-network-based short-term traffic flow prediction investigated in the literature (e.g., Yao et al. (2019)).*

The loss function is flexible to accommodate modelers' needs and available data sources. It can include partial aggregate traffic state observations like link flow and travel time, path choice probabilities from trajectory data, and benchmark OD demands from planning agencies. The framework integrates multi-source data into a single loss function and effectively handles inconsistencies among different data sources.

3.1.3 Neural Network Architecture

This section discusses the design of the neural network architecture in the proposed end-to-end learning framework. The architecture needs to accommodate the changes in the road network topology to facilitate "what-if" analysis. Moreover, it can be designed to ensure that the cost function possesses the desired properties to enable efficient training. We will illustrate it with the cost function with inelastic demand as an example. Hereinafter, we highlight that features/attributes are the **concatenation** of single features/attributes for all elements within one set. The design of the cost function requires

special consideration. We distinguish "feature" from "attribute" to avoid ambiguity: features refer to the input data of neural networks whereas attributes refer to the learned outputs of neural networks.

Attribute Net

We propose Attribute Net to learn the (path) attributes considered by travelers in their route choice decision process. As shown in, attributes $s^{[m]}$ depend on path flows $h^{[m]}$ and road network features $x^{\mathcal{G}}$. Attribute Net G_{θ} learns a continuous mapping from path flows and road network features to attributes, defined as:

$$G_{\theta}: \mathcal{H}^{[m]} \times \mathcal{X}^{\mathcal{G}} \to \mathcal{S}. \tag{9}$$

One may construct the Attribute Net with fully connected layers and learn a global mapping from link flows to link costs (e.g. Heaton et al. (2021)). In this case, the input and output dimensions of fully connected layers depend on the number of links in the road network. However, in "what-if" analysis, a planning agency may change the road network topology by adding or removing links. The fully connected layers—by definition with fixed size input and output—are incapable of accommodating the change in the number of links.

Inspired by the "kernel" concept in Convolution Neural Networks, we propose to learn the local attributes on the link, node, and path levels with three parallel, fully connected layers. As shown in Figure 3, the feature/attribute subscripts for enumerating the elements within a set and the superscripts for a sample m are omitted to facilitate presentation. The fully connected layers that learn link, node, and path attributes are called link, node, and path block respectively. The parameters of each block are shared among all elements of the same level to capture repeated patterns. Each block's input and output dimensions are independent of road network topology, allowing for changeable input sizes. To facilitate the presentation, the superscripts for a sample m are omitted for the rest of this section.

We use the superscript $\mathcal{A}, \mathcal{N}$, and $\mathcal{P}$ to distinguish the notations related to link, node, and path block.

The detailed constructions of link, node, and path block are similar. Hence, we take the link block as an example. As opposed to accepting multiple links as input, the link block takes the single link flow and single link features of one link $a \in A$ as input and outputs the corresponding link attributes, defined as:

$$g_\theta^{\mathcal{A}}: R_+ \times \mathcal{X}_a^{\mathcal{A}} \to \mathcal{S}_a^{\mathcal{A}}, \tag{10}$$

where $|\mathcal{I}^{\mathcal{A}}|$ is the number of features associated with one link; $|\mathcal{S}^{\mathcal{A}}|$ is the number of link attributes considered by travelers. Note the input and output dimensions of the link block are independent of link numbers.

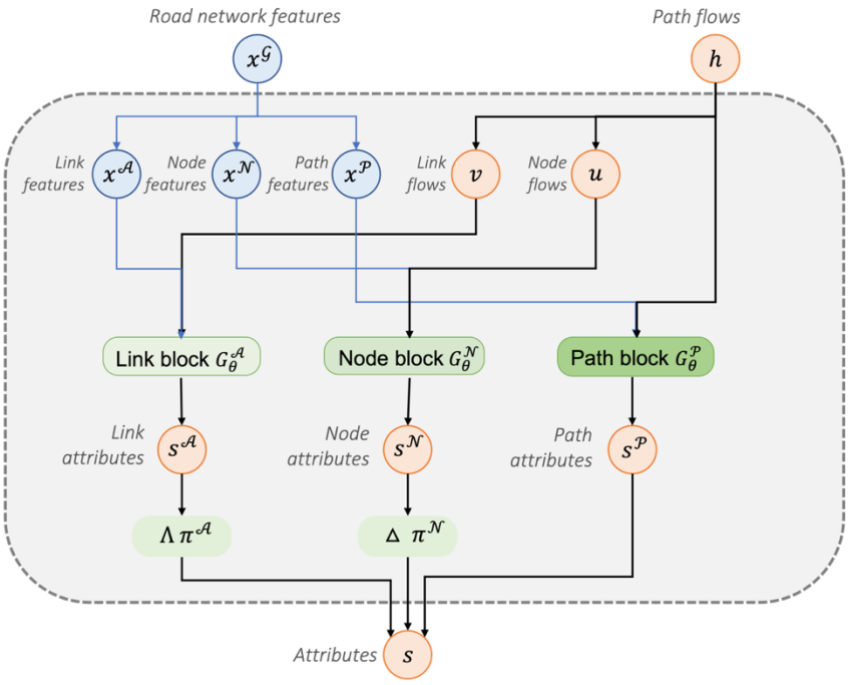


Figure 3 Illustration of Attribute Net

Similarly, let node flow be the sum of link flows from all approaches at node. To capture

the interactions among link flows, node block $g_\theta^{\mathcal{N}}: R_+ \times \mathcal{X}_n^{\mathcal{N}} \rightarrow s_n^{\mathcal{N}}$ maps the single node flow $\overline{u}_n \in R_+$ and single node features of one node to its local node attributes. The node attributes are the concatenation of single node attributes.

And the path block $g_\theta^{\mathcal{P}}: R_+ \times \mathcal{X}_p^{\mathcal{P}} \rightarrow s_p^{\mathcal{P}}$ maps the single path flows and single path features one path to its path attributes. Finally, the attributes π are the concatenation of link attributes, node attributes and path attributes, defined as:

$$s = \{\Lambda s^{\mathcal{A}}, \Sigma s^{\mathcal{N}}, s^{\mathcal{P}}\}, \tag{11}$$

where is Σ the path-node incidence matrix.

To facilitate training and enhance model performance, we can fully or partially replace each block with a pre-calibrated function, if available. For instance, we can replace the link block with the link performance functions calibrated by a planning agency. In addition, our future study will explore the use of convolution layers to accommodate changeable input sizes. The challenge will be to ensure the desired properties of the learned cost function.

Weight Net

Weight Net is proposed to capture traveler heterogeneities and learn the OD-specific preferences over learned attributes. We treat all travelers between the same OD pair as a single class that shares the same preferences. It is straightforward to further classify travelers between one OD pair to be multiple classes to reflect the preference heterogeneity among them. Weight Net L_θ learns a mapping from traveler characteristics to OD-specific weights $w \in W$, defined as $L_\theta: \mathcal{X}^{\mathcal{R}} \rightarrow \mathcal{W}$.

OD pairs can be added or removed in "what-if" analysis thus Weight Net also needs to accommodate the change in the number of OD pairs. Weight Net learns a function that

maps the single traveler characteristics of one OD pair $x_r^{\mathcal{R}}$ to its OD-specific weights w_r.

The parameters of neural network are shared among all OD-pairs to capture the repeated patterns in weights. Recent developments in interpretable neural-network-based discrete choice modeling can be incorporated into the proposed framework and guide the design of neural network architectures, particularly when behavior interpretability is desired.

Cost Function and Regularization

Subsequently, we assume that travelers choose routes to minimize their perceived path costs, which are represented as a weighted sum of attributes. Equivalently, let context features $\mathbb{1} \in R^{|\mathbb{S}|}$ include traveler characteristics $x^{\mathcal{R}}$ and road network feature $x^{\mathcal{G}}$. The cost function maps path flows and context features to path costs. The continuity of cost function ensures the existence of equilibria. However, stronger properties of the cost function may be desired to ensure the uniqueness of equilibrium or enable an efficient solution algorithm. In this section, we seek to entail the cost function with monotonicity and Lipschitz continuity via neural network regularization techniques. Both monotonicity, which suggests the path cost is non-decreasing as more travelers use this path, and Lipschitz continuity, which suggests a finite change in path flows results in a finite change in path costs, are mild assumptions but will largely enhance computational traceability.

Theorem 1 shows sufficient conditions to entail the cost function with monotonicity and Lipschitz continuity. Note that only path flows are treated as variables in this case.

Theorem 1 (Monotonicity and Lipschitz continuity of cost function) The cost function is maximal monotone and Lipschitz continuous with respect to path flows if weight is positive and link block, node block and path block are column-wise monotone.

Recall that each block is composed of fully connected layers. Let $y^{(l-1)}$and $\sigma^{(l)}$ represent the input and activation function of the l-th layer respectively. We constrain the sign of weights as strict positive by using SoftPlus as the last layer of Weight Net. The column-

wise monotonicity and Lipschitz continuity of attribute blocks, however, are more challenging to obtain. Most activation layers, such as ReLU and SoftPlus, are monotone and Lipschitz (Bibi et al., 2019) and both monotonicity and Lipschitz continuity are preserved via operator composition. Therefore, we only need to regularize the linear layer to entail the block with desired properties. Without loss of generality, we design a monotonic and Lipschitz continuous architecture that explicitly constrains the weights of the linear layers. More specifically, the weight of each linear layer is constrained to be positive to maintain monotonicity. The linear layer can be parameterized specifically if strict monotonicity or strong monotonicity are desired. The spectral normalization as proposed by Miyato et al. (2018) is applied to constrain the spectral norm of each $W^{(l)}$ and maintain Lipschitz continuity. This explicit method is reliable, easy to implement, and shows satisfactory performances in our numerical experiments. Other regularization methods, such as adding heuristic penalty terms to the loss function or solving integral problems in forward propagation (Wehenkel and Louppe, 2019; Gouk et al., 2021) are open for exploration in our future study.

3.1.4 Training

We need to deal with two computational challenges to implementing implicit layers in the proposed framework. First, it requires efficiently solving a batch of VI problems in the forward propagation, as previous methods for solving VI may not necessarily be suitable for batch operations. Second, because solving VIs usually entails many iterations, explicit backpropagation through each iteration can be computationally expensive. Efficient differentiation through the implicit layer, i.e., the VI, is needed.

This section presents an auto-differentiation-based gradient descent algorithm to solve the MPEC in Eq. (8). For simplicity, we explicitly formulate the dependence of the parameters while omitting input features. The optimality condition of the parametric VI in Eq. (6) can be recast as a fixed-point problem $y = g(\theta, y)$ where $g(\theta, y)$ is the fixed-point operator. We define the total loss function $f(\theta, y) = \ell(y(\theta), y) + r(\theta)$.

We consider the generalized cost function is strongly monotone and Lipschitz continuous

so that the equilibrium state is unique and is a continuous function of parameter θ (Dafermos, 1988). In a model-free modeling approach, neural networks can be regularized to ensure these desired properties. In this case, the proposed algorithm updates the parameter with its hypergradient in each training epoch. The hypergradient requires differentiating through the equilibrium state $y^*(\theta)$ to calculate the implicit gradient. To formally define the implicit gradient, we assume the following assumption holds.

Assumption 1 *The fixed-point operator $g(\theta, y)$ is continuously differentiable with respect to θ and y and the corresponding matrix is invertible.*

Supposing Assumption 1 holds, one can differentiate through the optimality condition and calculate the implicit gradient. Here we proceed to discuss the differentiability assumption in Assumption 1. If we assume the travelers follow the logit model when choosing their paths, the fixed-point operator is a logit loading function and is indeed differentiable. In a more general setting, the solution to VI can always be formulated as the fixed point of a gradient-projection operator. The gradient-projection operator is non-differentiable at the boundary of the feasibility set. In this case, Assumption 1 implies that we are focusing on the differential region of the gradient-projection operator, thereby keeping it within the differential programming region for convergence analysis. This approach is also adopted by Li et al. (2022). How to tackle the non-differentiability at the boundary remains to be an open question.

We consider training the end-to-end framework with K epochs. Each epoch handles two sub-problems: forward propagation, which finds an approximate optimal response variable via N iterations, and backpropagation, which employs auto-differentiation to approximate the hypergradient and update parameters. We will then elaborate on each subproblem. Subscript script k associates a variable with the k-th epoch and superscript n and q associates a variable to the n-th forward and q-th backward iteration respectively.

Forward: N-step Closed-Form Updates

Batched operation is essential for efficiently handling large empirical data sets when training the end-to-end framework. Specifically, forward propagation requires solving a batch of VIs in parallel, rather than solving a single constrained VI. Previous methods for solving VIs require repeatedly calling external optimization libraries to project onto the polyhedron constraint set of feasible path flows, and thus may not necessarily be suitable for batch operations (Li et al., 2020). To manage batch operations, we require a closed-form method for updating response variables so that we can encode this iterative process with computational graphs. These closed-form update rules also facilitate efficient auto-differentiation through equilibrium states during backpropagation (see Figure 4). We will discuss two types of solution algorithms, decoupled gradient-projection and mirror descent method, to handle link-based and path-based formulation respectively.

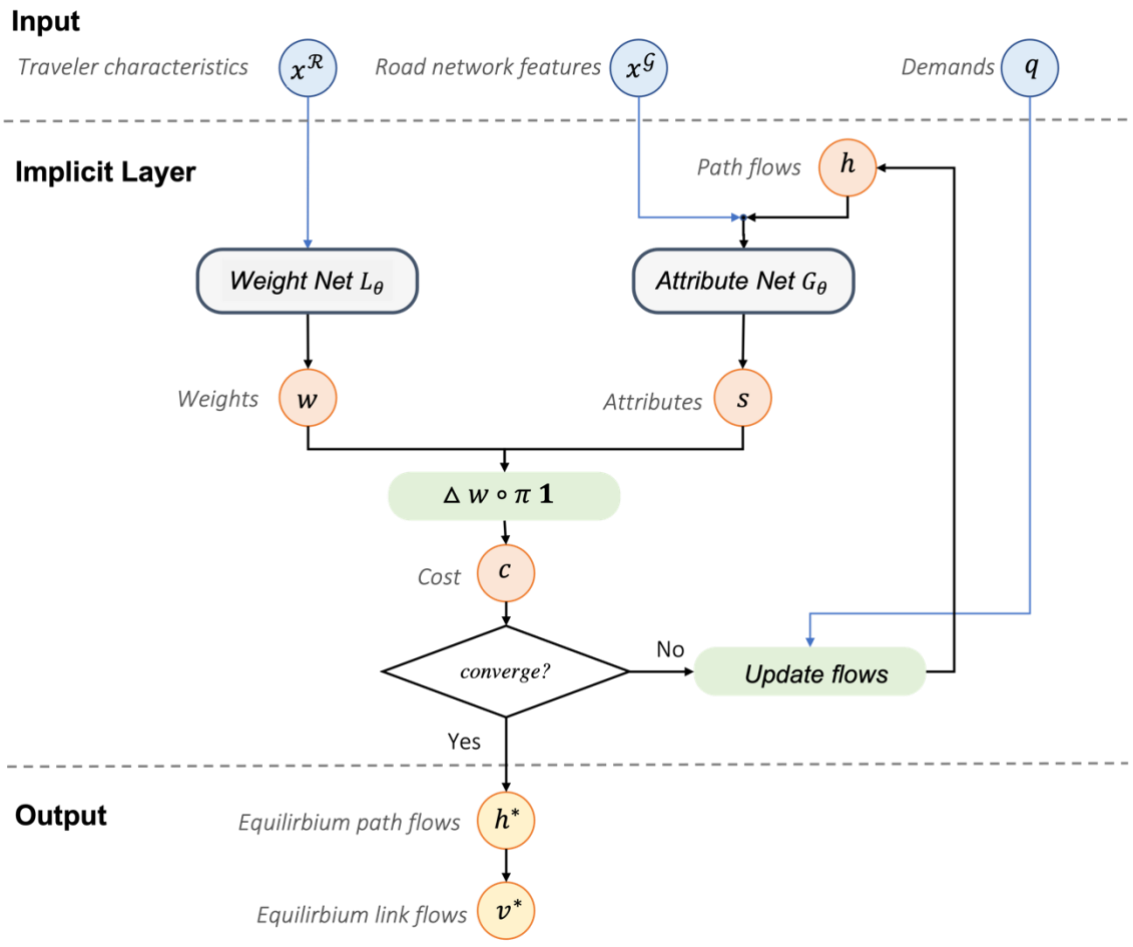


Figure 4 Illustration of the end-to-end learning framework

We apply the decoupled gradient-projection method to deal with link-based equilibrium constraints. The forward propagation updates the response variable via N-step gradient-projection operations. In a link-based formulation, the feasibility set is the Minkowski sum of the feasibility set for each OD pair, namely, $\mathcal{Y} = \sum_{r\in\mathcal{R}} \mathcal{Y}_r$ where $\mathcal{Y}_r = \{y_r : y_r \geq 0, \Lambda^\top y_r = d_r\}$. This allows us to break down the constraints by OD pairs and sequentially handle every pair on a large road network. Projecting directly onto the polyhedron constraint set, which requires repeatedly solving a batch of quadratic optimization problems. To tackle this efficiently, we leverage recent advancements in operator-splitting methods and decompose the polyhedron constraint set $\mathcal{Y}_r$ into two simpler sets: (i) one only involves inequality constraint $\mathcal{Y}_r^1 = \{y_r : y_r \geq 0\}$and (ii) another only involves equality $\mathcal{Y}_r^2 = \{y_r : \Lambda^\top y_r = d_r\}$. The projection onto

two simplier set $\mathcal{Y}_r^1$ and $\mathcal{Y}_r^2$ have closed-form solutions that can be encoded within computational graphs and then efficiently implemented in a batched manner. The convergence of this decoupled gradient-projection method has been demonstrated by Heaton et al. (2021).

Starting with an initial point, the decoupled gradient projection repeats $y^{n+1} = \overline{g}(\theta_k, y^n)$ for each step until the iteration step n exceeds the maximum number of iterations N. The initial point is not necessarily feasible and will be projected onto the feasible region during the iteration. The selection of step size is vital. If the step size is too large, the iteration may diverge; if too small, the convergence can be extremely slow. The optimal step size depends on an unknown Lipschitz constant, the exact computation of which is NP-hard (Virmaux and Scaman, 2018). We thus explore two variants of decoupled gradient-projection iteration to adjust the step sizes and speed up the convergence: Anderson mixing (Walker and Ni, 2011) and weighted ergodic iteration (Davis and Yin, 2017).

Anderson mixing updates y^{n+1} as an optimal linear combination of τ previous iterations. The optimal step size solves a quadratic program where the objective function is to minimize the sum of optimality gap over τ iterations. Here ϕ^{n-i+1} represents the optimality gap defined as follows.

Mirror descent method has shown good performance in dealing with path-based equilibrium constraints. We define the path choice probability $\sigma^n = \frac{y^n}{\overline{\Gamma}^\top \overline{\Gamma}_{y^n}}$ as auxiliary variables. The response variable is updated with a closed-form mirror descent operator. The constraint set of response variables becomes a probability simplex in path-based formulation and mirror descent has been shown efficient to deal with such a constraint set. This update rule can be viewed as a variant of the logit loading where the observed cost is scaled by the logarithm of route choice probability. The mirror descent iteration converges to the solution to the parametric VI in Eq.(2), as demonstrated in Li et al. (2022).

Additionally, solving for the auxiliary fixed point is equivalent to finding the root of $y^* - g_\theta(y^*, x) = 0$ via a root-finding method. The projection operator is non-differentiable at the boundary of a set and thus Newton s method may diverge. Therefore, we use Broyden's method, a quasi-Newton method that does not require differentiability. Broyden's method approximates Newton's direction and updates the point as $y^{n+1} = y^n - s^n$. Let the initial guess be $s^0 = -I$ and the direction is updated as $s^{n+1} = s^n + \frac{\Delta y^{n+1} - s^n \Delta \phi^{n+1}}{\Delta y^{(n+1)\top} s^n \Delta \phi^{n+1}} \Delta y^{(n+1)\top} s^n$, where $\Delta y^{n+1} = y^{n+1} - y^n$ and $\Delta \phi^{n+1} = \phi^{n+1} - \phi^n$.

Backward: Approximate Hypergradient

In forward propagation, we consider a practical setting where the parametric VI is solved with N steps and terminated before reaching perfect equilibrium. Consequently, in backpropagation, we need to approximate the hypergradient at a non-optimal response variable. To avoid the computationally expensive matrix inversion in approximating the implicit gradient, we present two auto-differentiation-based methods to approximate the implicit gradient.

Iterated Differentiation (ITD) memorizes the trajectory of N-step forward iterations and directly backpropagates through the equilibrating trajectory. In the N-th forward iteration, the response variable y_k^N depends on θ_k and y_k^{N-1}, namely $y_k^N = g(\theta_k, y_k^{N-1})$.

Here we use the fixed-point operator g as the "unified" operator that includes both gradient-projection operator and mirror descent operator. By applying the chain rule, we obtain a computational tractable approximation for the implicit gradient under ITD.

Inexact Implicit Differentiation (IMD) approximates the Hessian-inverse-vector product by solving an auxiliary fixed-point problem. Reformulating the definition of auxiliary variable suggests that it solves an auxiliary fixed-point problem. Then IMD recursively approximates the auxiliary variable using Q-step fixed-point iteration to approximate hypergradient under IMD. The auxiliary fixed-point iteration converges if

$I - \nabla_y g(\theta_k, y_k^N)$ is a stable matrix with a maximum eigenvalue that has a magnitude less than one. Previous studies show that these iterations typically are convergent in practice (Bai et al., 2019).

There are other methods in the literature to reduce the computational difficulty by approximating the matrix inversion. First, the Jacobian-free backpropagation replaces the matrix inverse with one identity matrix. This method can be viewed as a preconditioned gradient and only requires backpropagating through the final forward step (Fung et al., 2021). Second, an inverse matrix can be approximated with truncated Neumann series, reducing the computational cost from matrix inversion to matrix-matrix multiplications.

Remark 2 *Calculating the gradients of equilibrium flows with respect to demand or supply-side perturbations has been studied as equilibrium flow sensitivity analysis in the transportation literature. Tobin and Friesz (1988) showed that the Jacobian exists if the utilized path set remains the same with a small perturbation in parameters. Patriksson (2004) further suggested that the Jacobian exists if all unused paths remain unused with the perturbation. Li et al. (2020) pointed out the Jacobian exists if the cost function is strongly monotone in a neighborhood of h^*. These conditions may not hold in a general setting. However, the aforementioned numerical methods work well in our numerical experiments.*

To sum up, leveraging the hypergradient approximated under ITD and IMD, the parameter for epoch k is updated with learning rate $\beta > 0$ as with gradient descent under approximate implicit gradient. Here we adopt a warm-start strategy by setting the initialization as the output of the preceding training epoch rather than initiating it with random values.

3.2 Feasibility Analysis

The primary goal of the end-to-end framework is to support downstream planning by conducting "what-if" analyses and comparing different planning or design options. To

ensure consistent comparisons, a reference or benchmark is required, for which the notion of equilibrium has long been utilized due to its independence from initial conditions, behavioral foundation, and mathematical traceability. However, real-world network states may never reach equilibria, and equilibrium states are rarely directly observable from empirical data but must be inferred.

The above two points create a dilemma: on one hand, empirical observations do not reflect equilibrium while, on the other hand, the modeling paradigm requires equilibrium. To reconcile, we assume that empirical flow observations oscillate around an "ideal" equilibrium state or are perturbations from this state. Consequently, such an "ideal" equilibrium state is defined as the one that is the closest to all perturbed observations by a distance measure. The end-to-end learning framework seeks to learn this ideal state from finite observations. Conceptualized as a mathematical construct derived from empirical data, this "ideal" equilibrium state serves as a crucial benchmark, enabling consistent comparisons across different planning options. Our analysis in this chapter seeks to determine whether the proposed framework can learn this "ideal" equilibrium state from data.

We proceed to evaluate the feasibility of the proposed end-to-end network equilibrium model by comparing its population risk against that of the target model. As illustrated, the difference between the two population risks can be decomposed into three distinct components: expressivity, generalization, and optimization risk. Each represents a unique challenge in the end-to-end process.

We analyze each risk under the following two assumptions:

Assumption 2 (Lipschitz continuity of the loss function) The loss function $\ell(\ )$ is L-Lipschitz continuous in y.

Assumption 3 (Well-posed target cost function) The target cost function is continuous in x and μ-strongly monotone and L-Lipschitz continuous in y.

Following the compact parameterized VI formulation, this feasibility analysis applies to both path- and link-based formulations. While our primary focus is on the path-based formulation, both formulations are validated through numerical experiments.

First, expressivity risk arises when modelers, using imperfect prior knowledge, fail to select a model component that accurately represents real-world processes. The end-to-end framework is proven to be expressive: this framework, when parameterized with sufficiently large neural networks, can replicate any unique, differentiable equilibrium state that solves a "well-posed" VI. Theorem 2 establishes that the expressivity risk becomes negligible when the augmented cost function is parameterized by a sufficiently expressive neural network. In other words, the end-to-end framework, supported by neural networks, can accurately replicate any unique, differentiable equilibrium flow that solves the "well-posed" VI.

Theorem 2 (Expressivity risk) *Suppose Assumptions 1, 2 and 3 hold. If the augmented cost function is parameterized by an infinite-depth neural network with a continuous nonpolynomial activation function δ, the expressivity risk is bounded by arbitrarily small $\epsilon > 0$.*

Modelers may also pre-calibrate certain components using domain knowledge, leaving the remainder to the end-to-end framework. Such pre-calibration can reduce overall expressivity risk if it lowers the learning error in critical components. It is therefore essential to strategically choose which components to learn. Focusing on less important components will not reduce expressivity risk if significant errors persist in critical ones.

To continue the analysis of generalization risk, generalization risk emerges when the framework, trained on a limited dataset, struggles to adapt to new, unseen data. This end-to-end framework is demonstrated to be able to generalize to unobserved data when trained with sufficient observations.

Lemma 2 (Bounded approximate implicit gradient) Under Assumptions 3 and 4, the approximate implicit gradient is bounded above.

Building on Lemma 2, the following theorem provides an upper bound for the generalization risk.

Theorem 3 (Generalization risk) *Suppose there exists a parameterized model that yields zero loss on one sample, where the parameterized cost is L-Lipschitz continuous in θ and x, the generalization risk is bound above and decreases with the number of samples.*

As shown in Theorem 2, the generalization bound decreases at a rate of number of samples and scales with the feasible parameter set size. This suggests that larger training samples and simpler models help control generalization risk.

Finally, optimization risk arises from the complexity of finding the optimal parameters, which involves solving a batch of VIs and differentiating through equilibrium states. Convergence analysis shows that ITD requires an iterative equilibrating process to guarantee local convergence, while IMD can compensate for the absence of an equilibrating process with additional information from the implicit function theorem.

Table 1 Summary of trade-offs. Results marked with an asterisk (*) are suggested but not explicitly proven due to the complexity of finding global optima in MPEC.

	# parameters $\|\Theta\|\uparrow$	# training samples $\|M\|\uparrow$	Forward iterations $N\uparrow$
Model risk	↓	=	=
Generalization risk	↑	↓	=
Optimization risk*	↑	↑	↓

Table 1 summarizes the trade-offs in inversely learning an end-to-end network equilibrium model. Increasing model parameters reduces expressivity risk but raises the likelihood of overfitting and training complexity. Expanding the training dataset improves generalization but may complicate optimization. Finally, increasing the number of forward iterations enhances the approximation of user equilibrium but incurs higher computational costs and potential numerical instability when N becomes very large.

The proposed end-to-end framework allows each model component to be parametric, semi-parametric, or end-to-end. Modelers may select specific functional forms for certain components, encoding them with computational graphs while leaving other components learnable. Compared to a fully end-to-end approach, parametric and semi-parametric approaches offer greater interpretability but require stronger prior knowledge and correct model class selection, thereby heightening expressivity risk. Nonetheless, the analysis of generalization and optimization risk remains valid for both parametric and end-to-end settings under the stated assumptions.

3.3 End-to-End Optimization

We further investigate efficient solution algorithms to prescribe improvement strategies using the end-to-end learning framework, using a transportation design problem as an example. Transportation network design seeks to optimize road network expansion to accommodate anticipated travel demands in the future years. A central challenge is accounting for how travelers adapt their route choices in response to network expansion. These choices are inherently uncertain, influenced by supply-side factors such as weather-dependent link capacities and demand-side factors such as daily variations in travel demand. Traditional approaches to transportation network design under uncertainty typically begins by modeling these parameters—for example, link capacities or demands—as random variables following calibrated probability distributions. The resulting network design problem is formulated as a stochastic optimization (SO) problem with user equilibrium constraints. The objective is to minimize social cost, such as total system travel time or emissions, by selecting appropriate design variables, such as continuous capacity expansions. Initially, planners are often assumed to be risk-neutral and aim to minimize the expected total social cost (Patriksson, 2008). Later approaches extend this framework to incorporate more flexible risk attitudes, such as chance constraints (Lo and Tung, 2003).

We focus on context-related uncertainty, which is particularly challenging. Consider a planning agency that seeks to allocate a budget to optimize continuous link capacity

expansions for long-term demand growth. The design vector $z \in Z$ represents the proportion of capacity expansion for each link, subject to the budget constraint $Z = \{0 \leq z \leq 1, s^T z \leq \bar{s}\}$, where $\bar{s}$ is the total budget and s is the per-project budget. Network expansion affects travelers' route choices, as travelers can reoptimize their paths to minimize individual disutility in response to expanded capacity. Beyond route selection, network expansion represents long-term infrastructure investments that may influence broader travel decisions, such as whether and where to travel.

In an idealized setting where the uncertain context is fully known, a risk-neutral planner aims to minimize the expected social cost $\omega(z, y)$ under contextual uncertainty:

$$\min_{z} \sum_{m=1}^{M} \omega\left(y^{*[m]}, z\right) \; s.t. \; \langle F_\theta(y^{*[m]}, z, x^{[m]}), y - y^{*[m]} \rangle \geq 0, \qquad \forall y \in \mathcal{Y} \tag{12}$$

This bi-level formulation mirrors the structure of the end-to-end learning problem. Consequently, both can be addressed using the same class of solution algorithms.

3.4 Equipment

The Lab for Innovative Mobility Systems (LIMOS) at the University of Michigan utilizes the campus-wide Great Lakes High-Performance Computing (HPC) Cluster, a state-of-the-art resource supporting simulation, modeling, machine learning, data science, genomics, and other computationally intensive research. This Slurm-managed cluster comprises 380 nodes with a total of 13,000 processing cores, providing at least 5 GB of RAM per core. Optimized for large-scale batch processing, the cluster is well-suited for handling extensive, long-duration tasks requiring significant CPU, memory, or I/O resources. All experiments for this project were conducted on the Great Lakes cluster using a single NVIDIA A40 GPU (48 GB of memory), 8 CPU cores, and 64 GB of system memory.

4 Validation: Synthetic Data

In this section, we validate the proposed end-to-end framework using three synthesized datasets from three benchmark networks widely used in the literature, i.e., Braess, Sioux Falls, and Chicago Sketch. We use the Braess example to validate the approximation guarantee of the end-to-end framework. Through the Sioux Falls example, we examine the effect of enforcing equilibrium constraints and provide practical guidelines for training. The Chicago Sketch example demonstrates the simultaneous learning of supply and demand-side components. We evaluate the framework performance using two key metrics: the empirical optimality gap, which measures the convergence of the parametric VI, and the Weighted Mean Absolute Percentage Error (WMAPE), which quantifies percentage differences in flow predictions. We define the empirical optimality gap as the sample-averaged inner product between the generalized cost function and the changes in the response variable across two successive.

4.1 Example 1: Learn Demand Component on Braess

The Braess network has five links, four nodes, and a single OD pair from node 1 to node 4 with three feasible paths. With a maximum possible demand of $q = 5$, the ground-truth demand function for OD pair r follows and exponential function using x_r represents OD-specific features; $x^{[m]}$ is a one-dimensional sample-dependent contextual feature; $\alpha_u = 2$ and $\beta_u = 4$ are functional parameters. We use the standard BPR function as link performance functions and assume travelers only consider travel time when selecting their paths. The dataset includes 1024 training, 258 validation, and 258 testing samples.

We will focus on learning the inverse demand function in this example and assume both link performance and cost functions are given. We consider that multi-day link flows are observable, and the loss function measures the Mean Square Error (MSE) between predicted and observed link flow distributions. The framework is trained using the Adam optimizer over $K = 500$ epochs with early stopping implemented if there

is no improvement in the training MSE over twenty consecutive epochs. The forward propagation uses mirror descent with $N = 100$ iterations, while backpropagation employs the ITD method.

We evaluate both model-free and model-based end-to-end frameworks under the following scenarios, fine-tuning the learning rate and step sizes via grid search for each setting.

- *Benchmark:* We use a grid search to identify a constant demand that best matches all testing samples, which is 2.4 in this case.

- *Functional:* Assuming the functional form is known and encoded with computational graphs, the framework learns two parameters: α_u and β_u.

- *Constant:* The framework learns a context-independent fixed demand, with neural networks using only excess demand and OD-specific features as input.

- *Linear:* The neural network includes a single linear layer.

- *Nonlinear:* The neural network combines a linear part (as in the Linear scenario) and a nonlinear part, comprising three layers with eight neurons each.

- *Residual:* The neural network includes three layers with eight neurons each and employs a residual strategy between layers.

Each neural network is designed to accommodate potential changes in the number of OD pairs during "what-if" analyses. The input dimension of these neural networks only depends on the number of input features, which in this case, is three. In the Nonlinear and Residual scenarios, neural networks are regularized to be monotone and Lipschitz continuous.

Table 2 presents the WMAPE under different learning scenarios. WMAPE is shown in

percentage, and parentheses display the relative reduction in WMAPE. The optimal scenario is highlighted with a star. Same for the following tables. In Benchmark scenario, the link flow WMAPE is remarkably high at 72.1%. This error drops to 4.2% when we embed the ground-truth functional form in the framework and adjust the α_u and β_u. The non-zero error can be attributed to the nonconvexity of the MPEC, which can trap the training process at a local minimum. The model-free Constant scenario learns context-independent demands and yields an error of 72.3%, comparable to Benchmark. The Residual scenario knows contextual information but has no information about the functional form of the inverse demand function. By exploring the representation power of neural networks, the model-free framework still yields a WMAPE of 4.7%, comparable to the Functional scenario. This result confirms that the end-to-end framework can generate reliable flow distributions without knowing each component's functional form. The Residual scenario provides the best performance since the residual strategy helps avoid the gradient vanishing when *N* becomes large.

Table 2 WMAPE under different scenarios

Model	# Parameters	Link flow	Link time	Demand
Benchmark	/	72.1	31.2	71.9
Functional	2	4.2 (-94.1%)	1.6 (-94.9%)	3.9 (-94.6%)
Constant	109	72.3 (+ 2.8 %)	31.28 (+2.5%)	72.4 (+ 7.0%)
Linear	4	15.5 (-78.5%)	5.6 (-82.1%)	12.9 (-82.0%)
Nonlinear	117	6.3 (-91.2%)	4.3 (-86.2%)	6.2 (-91.3%)
Residual *	112	4.7 (-93.5%)	3.2 (-89.7%)	4.6 (-93.6%)

4.2 Example 2: Learn Demand Component on Sioux Falls

Sioux Falls network has 76 links, 28 nodes, and 528 OD pairs. We scaled the default demand in Stabler (2023) by a factor of three to serve as the maximum possible OD demand q. The rest of the ground-truth setting follows the Braess example. The dataset is divided into 1,024 training, 258 validation, and 258 testing samples. We consider a link-based formulation using the decoupled gradient-projection method in forward propagation. With known link performance and cost functions, our focus is on learning the inverse demand function.

We first investigate the framework performance with different forward steps. Figure 5 shows that increasing N from 1 to 50 enables faster and better training under both IMD and ITD. A larger N requires more iterations for both forward and backward propagation and notably increases computation time under ITD. By contrast, IMD avoids the differentiation along the equilibrating trajectory and the computation time changes relatively minimally when N varies.

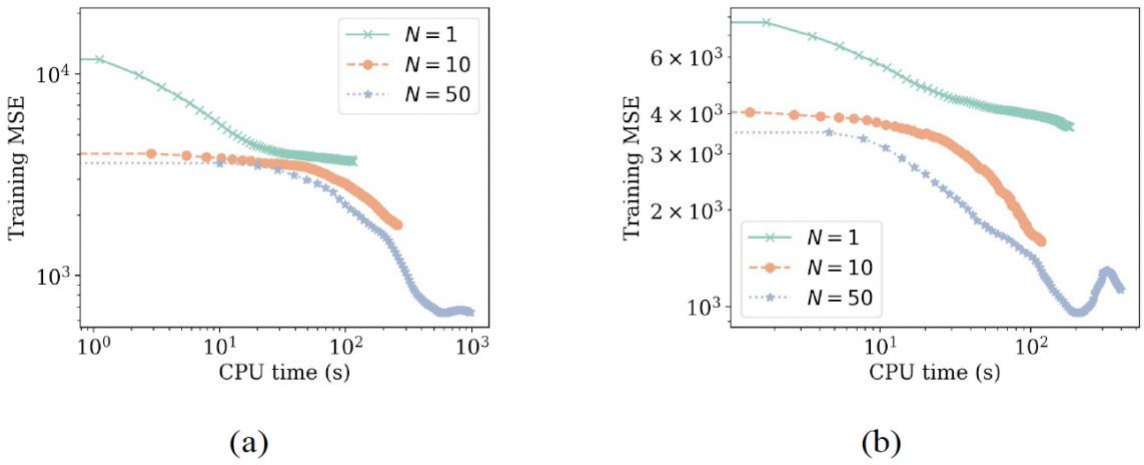
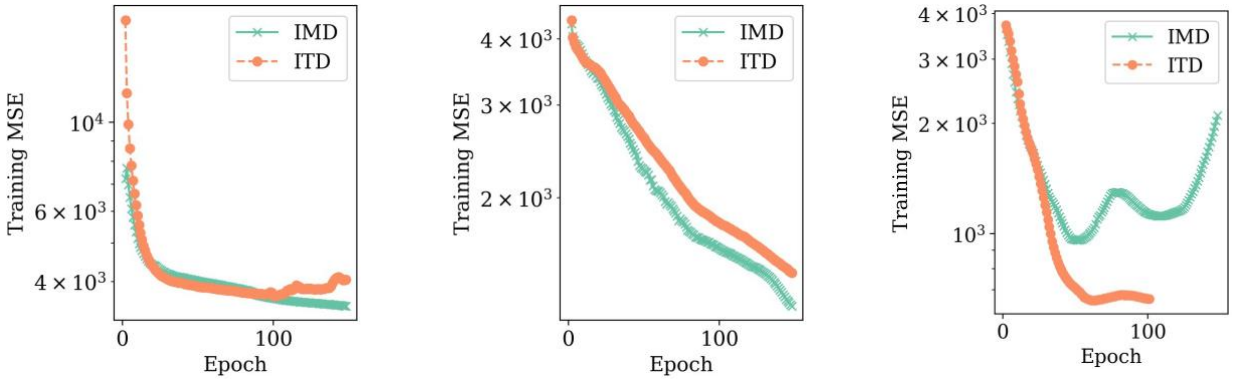


Figure 5: Framework performances with different forward iterations under (a) ITD and (b) IMD.

Moreover, Figure 5 shows that the training process under ITD stops prematurely with $N = 1$, resulting in a high training MSE of 4e3. This highlights an iterative equilibrium process is necessary to ensure local convergence under ITD. By contrast, IMD keeps reducing the training MSE with $N = 1$ because it uses extra information from the implicit function theorem to correct auto-differentiation. Both ITD and IMD manage to avoid getting stuck when N increases to 10. ITD outperforms IMD in finding better local optima when N increases to 50.

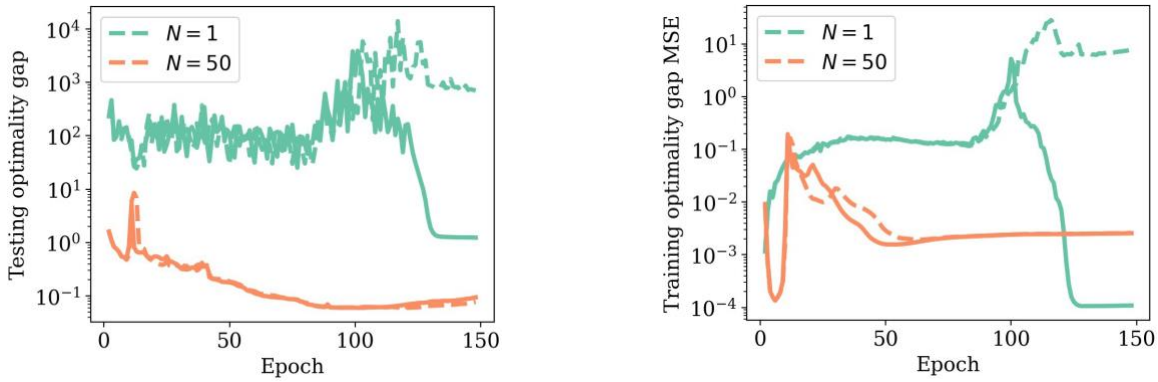


(a) (b) (c)

Figure 6 Framework performances using different backward method with (a) N = 1, (b) N = 10, and (c) N = 50

Next, we examine whether penalizing the empirical optimality gap in the loss function can replace the need for enforcing equilibrium conditions during training. As illustrated in Figure 7, Scenarios with optimality gap regularization are represented by solid lines, while those without are denoted by dotted lines. When the equilibrium constraints are poorly approximated with $N = 1$, the optimality gap regularization indeed steers the parametric VI towards a smaller empirical optimality gap. As the training proceeds, the optimality gap MSE converges towards zero. Similar findings have been found in (Guarda et al., 2023). By contrast, when the equilibrium constraints are well-approximated with $N = 50$, the optimality gap regularization has little impact on

framework performance.



(a) (b)

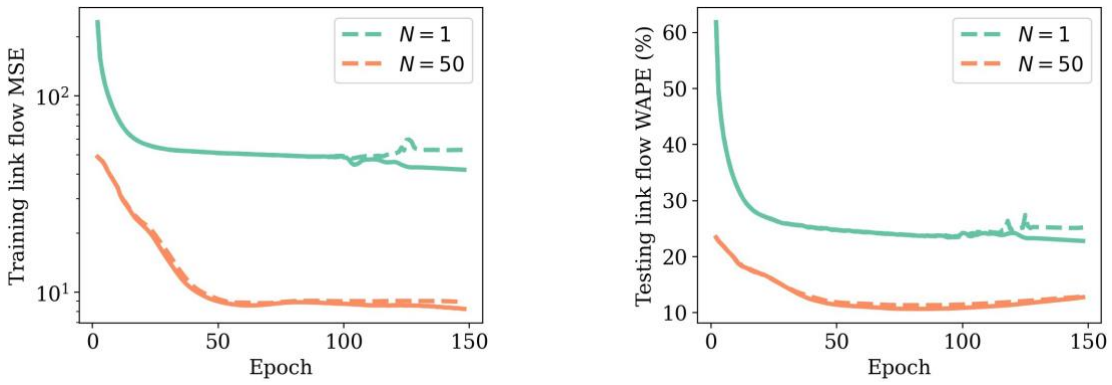
Figure 7 (a) Testing optimality gap and (b) training optimality gap MSE with respect to epochs.

Despite guiding the training process towards an equilibrium state, the optimality gap regularization fails to lead the framework to find suitable parameters. As shown in Figure 7(b), the training MSE with optimality regularization and $N = 1$ remains noticeably higher than that with $N = 50$. The link flow prediction error with $N = 1$ is also significantly larger. This suggests that "softly" penalizing the optimality gap in the loss function is not a viable alternative to the "hard" enforcement of equilibrium conditions. Therefore, it is essential to at least roughly approximate the equilibrium conditions to facilitate effective end-to-end learning.

Finally, we experiment with two enhanced training strategies:

- **Adaptive N (denoted as A):** Increases the number of forward iterations linearly during training, from 50 to 150 in our case.

- **Two-stages training (denoted as T):** Initially, the linear part of the neural network is trained while keeping the nonlinear part fixed. Once the linear part converges, both parts are trained jointly in the second stage.



(a) (b)

Figure 8 (a) Training link flow MSE and (b) testing link flow WMAPE with respect to epochs.

In the Benchmark scenario, we proportionally scaled the maximum possible OD demands q and use grid search to determine the optimal scale that best matches all observations, which is 1.2 in this case. The remaining Functional, Nonlinear, and Residual scenarios follow the Braess example with $N = 50$ and ITD as the backpropagation method. Table 3 indicates that the adaptive N strategy improves the model's performance because a rough estimation of equilibria is sufficient when the parameters are considerably off-target during the initial training epochs. The two-stage training strategy also enhanced performance because it trains a shallow linear network in the first stage. On one hand, a linear approximation of the monotone generalized cost function is relatively good. On the other hand, shallow neural networks mitigate vanishing or exploding gradients during training. Thus, incorporating both strategies, our end-to-end framework achieves the best performance of 4.3%, comparable to the Functional scenario (i.e., 1.3%).

In this example, we calculate WMAPE only for flows over 0.001 to avoid infinite WMAPE due to zero ground-truth flows in training samples. Thus, despite Sioux Falls' larger size, its WMAPE is numerically smaller than Braess. Since our main concern is the relative WMAPE reduction, rendering this should be insignificant to our conclusions.

4.3 Example 3: Learn Behavior Component on Sioux Falls

In this case study, each OD pair r is assumed to have one continuous feature denoting income and one binary feature denoting travel purpose, which equals 1 if the destination of OD pair r is a commercial area and equals 0 otherwise. We assume the path travel time includes two parts: link travel times and node delays.

The link travel time on link a follows the BPR function. The node delay on node follows an exponential form as proposed by Jeihani et al. (2006), Moreover, pavement surface conditions, such as roughness, are the main feature that decides user comfort (Hawas, 2004; Yin et al., 2008). We classify the links as good and bad pavement conditions and assume travelers experience a non-link-additive discomfort on bad-condition links. Let x_p denote the proportion of bad-condition link length to the total path length. The discomfort follows the exponential form and increases with the bad-condition link proportion. We set the discomfort is zero if path p only includes good condition links.

Table 3 WMAPE under different training settings

Scenario	# Parameters	Link flow	Link time	Demand
Benchmark	/	50.9	97.6	59.3
Functional	2	1.3 (-97.4%)	3.7 (-96.2%)	1.3 (-97.8%)
Linear	4	14.1 (-72.3%)	40.9 (-58.1%)	6.4 (-89.2%)
Nonlinear	117	10.2 (-80.0%)	12.1 (-87.6%)	6.8 (-88.5%)
Nonlinear (+ T)	117	9.0 (-82.3%)	24.4 (-75.0%)	5.6 (-90.6%)
Nonlinear (+ A)	117	7.9 (-84.5%)	22.1 (-77.4%)	4.9 (-91.7%)
Nonlinear (+ T + A) *	177	4.3 (-91.5%)	9.2 (-90.5%)	2.7 (-95.4%)

Residual	112	12.8 (-74.9%)	37.8 (-61.4%)	7.2 (-87.9%)
Residual (+ A)	112	8.3 (-83.7%)	24.4 (-75.0%)	5.2 (-91.2%)

The "ground-truth" cost for travelers of OD pair r to use path p is a weighted sum of link travel times, node delays, and a discomfort constant. This suggests that travelers with higher incomes have higher weights on both node delays and discomfort. Travelers traveling to commercial areas have higher weights on discomfort yet lower weights on node delays.

The feasible path set includes the top three paths with the shortest free-flow time. If one OD pair has fewer than three feasible paths, its path flows are padded to a dimension of three and the padded path flows are nullified with the mask trick during training. Three demand levels are considered: (i) base scenario, (ii) uncongested scenario with base demand reduced by 50%, and (iii) congested scenario with base demand increased by 50%. For each scenario, we randomly sample travel demands from a uniform distribution between 0.5 and 1.5 base demand. The equilibrium flow is solved for each sampled demand given the ground-truth cost. The training and test sets include 1, 536 and 512 samples respectively. So far, all links are assumed to be observable.

The link block is replaced with pre-calibrated BPR functions. Weight Net, node block, and path block are composed of three fully connected layers with four neurons and with LeaklyReLu as the activation function. Normalization layers are added to enhance training stability. The input of the node block includes node flows and intersection parameters. The proportion of bad-condition links is the input of the path block. The input and output dimensions are as follows: Weighted ergodic iteration and IMD are used as the default forward and backward methods respectively. The model is trained with Adam optimizer with Mean Square Error as the loss function under the learning rate of 0.1. Early stop is enabled if no loss descent is observed in five consecutive epochs. To illustrate the feasibility and importance of learning route choice preferences, we benchmark our model with three well-established network equilibrium models.

First, the cost function is assumed to be link travel time and travelers choose the paths with minimum travel time, yielding conventional Deterministic User Equilibrium (denoted as DUE). The second behavior model assumes travelers' path choices follow a logit model and thus results in a Stochastic User Equilibrium (denoted as SUE). In this case, the dispersion parameter is calibrated, similar to Yang et al. (2001). The third model keeps the same path choice model but assumes the cost function is a linear combination of link travel time and the proportion of bad-condition links (denoted as SUE-2). Two linear coefficients are calibrated in this case, similar to Guarda and Qian (2022).

We compare the efficiency and robustness of different forward algorithms. The first type includes decoupled gradient-projection iteration (F) and its accelerated variant: Anderson mixing (FA) and weighted ergodic iteration (FW). The second type is Broyden's method (R). We also explore the combinations of two types (denoted as F-R, FA-R, FW-R), which use decoupled gradient-projection iterations initially and switch to the root-finding when the relative residential is sufficiently small. We consider two types of tests: in-distribution and out-of-distribution. In in-distribution tests, the model is trained on observations from the Sioux Falls network and tested on the same road network. By contrast, in out-of-distribution tests, the trained model is tested on a partially changed road network. In our experiments, four links are added to the original Sioux Falls network and 25% links are randomly selected to increase or decrease their capacities by 50%. Decreasing the capacities under congested demand generates unreasonable training sets and is excluded in later analysis.

Performance comparisons

Table 4 compares the MAPE of different network equilibrium models. The proposed end-to-end learning framework is denoted as "Implicit". We use DUE as the baseline and denote its MAPE as η_0. The change in MAPE of other models is denoted as $\Delta\eta = \eta - \eta_0$. Note that the behavioral assumptions of SUE are different from the ground truth. Although SUE can reduce the in-distribution MAPE by 18.2%, it shows inferior performance in out-of-distribution tests, increasing the MAPE by 9.2%. This suggests

inaccurately assuming an SUE behavior model can cause bias in parameter estimation, misleading the flow prediction in subsequent "what-if" analysis. Similar results have been shown in Torres et al. (2011) and Van Der Pol et al. (2014). In comparison, SUE-2 performs better, because it happens to capture the impact of discomfort from the bad-condition links. The performance of SUE-2 is still less satisfactory compared with the end-to-end framework because the former learns linear combinations by assumption whereas the latter can deal with nonlinear patterns. Since neural networks include more parameters than baseline models and offer greater flexibility to recover the complicated ground truth cost function, the proposed framework has the best performance in both in-distribution and out-of-distribution tests as expected, reducing the benchmark MAPE by 61.5% and 55.1% respectively.

Table 4 MAPE of different network equilibrium models.

In-distribution test

Demand	Capacity	DUE η_0	SUE $\Delta\eta$	SUE-2 $\Delta\eta$	Implicit $\Delta\eta$
Base	Default	20.6	-4.7	-11.8	-15.0
Uncongested	Default	12.5	-3.1	-0.2	-3.4
Congested	Default	13.41	-0.6	-4.4	-10.2
Mean		15.5	-2.8 (-18.2%)	-5.4 (-35.1%)	-9.5 (-61.5%)

Out-of-distribution test

Demand	Capacity	DUE η_0	SUE $\Delta\eta$	SUE-2 $\Delta\eta$	Implicit $\Delta\eta$
Base	Default	22.3	-7.3	-14.4	-16.6
	-50%	11.3	+13.4	-1.6	-7.9
	+50%	8.1	+4.8	-1.0	-1.3
	Default	23.4	-8.5	-15.6	-14.9

Uncongested	-50%	12.1	+12.6	-2.4	-4.1
	+50%	10.4	+2.5	-3.3	-1.1
Congested	Default	13.8	-3.5	-6.3	-10.1
	+50%	11.9	-3.5	-5.2	-6.4
Mean		14.2	+1.3 (+9.2%)	-6.2 (-44.0%)	-7.8 (-55.1%)

As shown in Table 5, FW and FW-R achieve the smallest MAPE of 5.7% in in-distribution tests whereas FW-R slightly outperforms FW by 1% in out-of-distribution tests. Forward algorithms involving Anderson mixing, such as FA and FA-R, can be the most unstable. By contrast, forward algorithms involving weighted ergodic iteration, such as FW and FW-R, are more stable as they consistently shrink the step size during iterations.

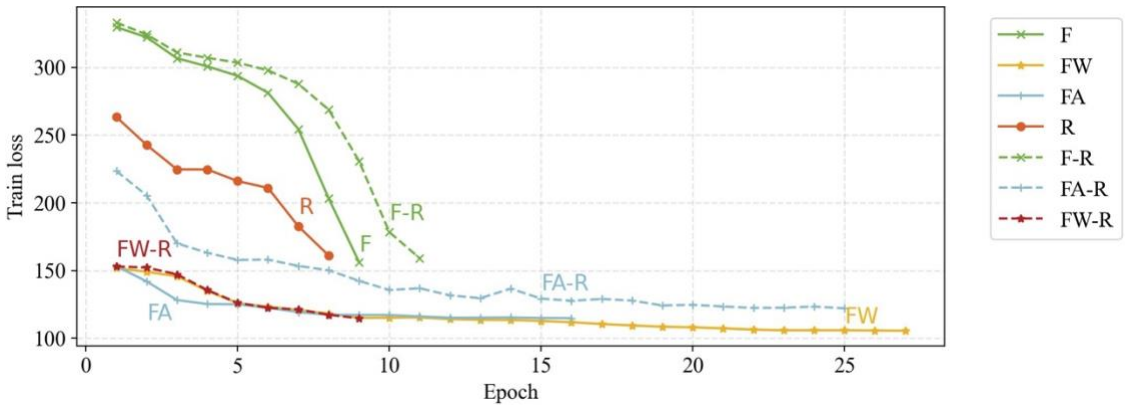


Figure 9 Training process of different forward algorithms.

Figure 10 compares the performance of three backpropagation methods: Jacobian-Free (JF) approximation, Newman Approximation (NA), and Inexact Implicit Differentiation (FA) under different demand levels. FA has the best performance among the three proposed backward methods. JF significantly hurts the learning process. Similar results have been found by Huang et al. (2021).

The effects of spectral normalization are shown in Figure 11. “w” suggests ”with spectral normalization” and ”w/o” suggests ”without spectral normalization”. Although requiring additional computation, the spectral normalization constrains the Lipshitz constant of the cost function within a reasonable range and speeds up the convergence by three to four times under all demand levels.

Table 5 MAPE of proposed forward algorithms.

In-distribution test								
Demand	Capacity	F	FA	FW	R	F-R	FA-R	FW-R
Base	Default	8.4	5.6∗	5.7	8.0	8.7	6.2	6.0
Uncongested	Default	9.5	9.1	8.1	9.5	8.3	8.5	8.0∗
Congested	Default	6.1	3.2	3.2	6.2	11.0	4.5	3.1∗
Mean		8.0	6.0	5.7∗	7.9	9.3	6.4	5.7∗
Std		1.8	3.0	2.4	1.7	1.4	2.0	2.5
Out-of-distribution test								
Scenario	Capacity	F	FA	FW	R	F-R	FA-R	FW-R
Base	Default	7.5	5.7∗	5.8	7.2	7.7	6.4	6.0
	-50%	4.5	3.4∗	3.4∗	5.0	4.8	3.6	3.4∗
	+50%	9.1	6.9∗	7.0	10.2	9.3	8.8	7.4
Uncongested	Default	8.3	8.5	7.6	8.1	8.0	8.2	7.5∗
	-50%	9.5	8.0	7.3	9.9	7.6	14.4	6.9∗
	+50%	8.4	9.4	7.9∗	8.4	8.2	7.9∗	7.9∗
Congested	Default	5.7	3.6∗	3.8	5.1	10.2	4.3	3.6∗
	+50%	8.8	5.5∗	5.7	6.2	11.9	6.1	5.5∗
Mean		7.7	6.4	6.1	7.5	8.5	7.5	6.0∗
Std		1.8	2.2	1.7	2.0	2.1	3.4	1.7

Robustness analysis

In this section, we examine the robustness of the proposed framework by relaxing model assumptions. FW, R, and FW-R have the best performance and are thus selected. Since in-distribution and out-of-distribution performances have similar trends, all the following analyses are based on in-distribution tests.

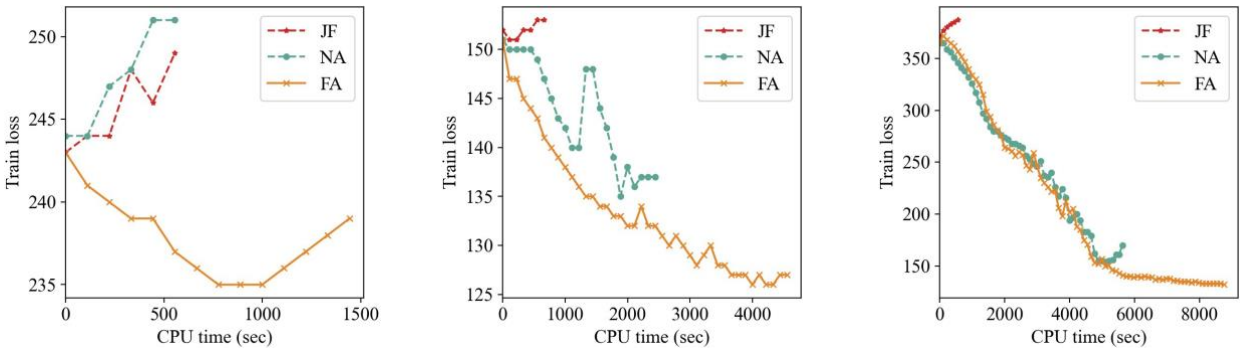


Figure 10 Performances of different backpropagation methods under (a) base, (b) uncongsted, (c) congested demand.

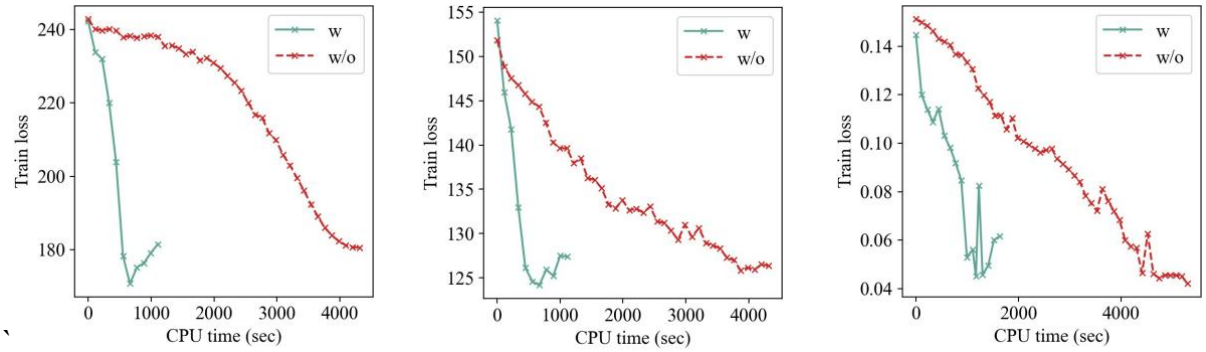


Figure 11 Effects of spectral normalization under (a) base, (b) uncongested, and (c) congested demand.

All links are assumed to be observable in previous analyses. We relax this assumption by randomly observing a proportion of links. FW-R is the most stable when only a

proportion of links are equipped with sensors. For example, Figure 12 shows the MAPE of FW-R slightly increases from 8.0% to 11.5% when the proportion of observable links decreases from 100% to 20% under uncongested demand. Since approximation errors can accumulate in both forward propagation, where iterations terminate with residuals, and backward propagation, where the gradients are approximated, the training of the proposed framework can stop at local optimums. Previous studies have shown the training process and final performances of models involving implicit layers can be relatively noisy and require more hyperparameter tuning (Huang et al., 2021; Li et al., 2020).

Usually, there are no direct observations of OD demands in urban road networks. OD demands need to be estimated and thus prone to estimation errors. We examine the model performances when the input OD demands are different from the ground truth. More specifically, random observation noises, which are proportional to the ground

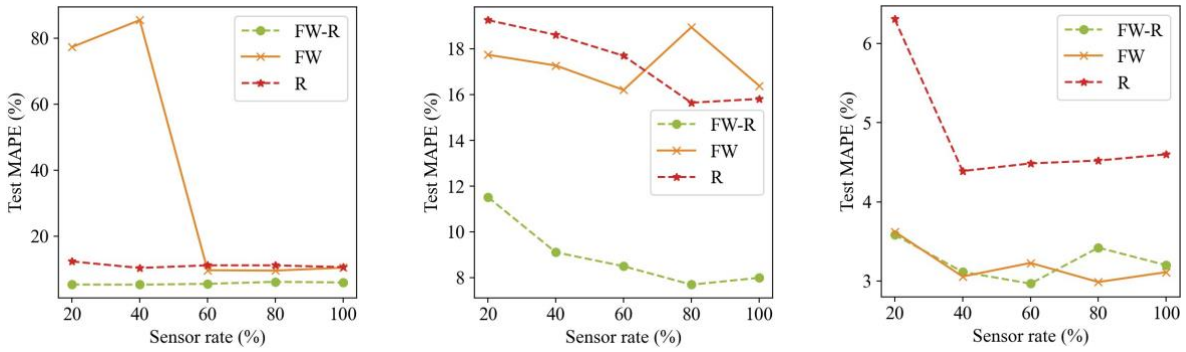


Figure 12 Model performances with different sensor coverage rates under (a) base, (b) uncongested, and (c) congested demand.

truth, are added to all demands. As shown in Figure 13, FW is the most stable in the case of demand noises. Given a noise scale of 100%, the increase in its MAPE ranges from 12.5% to 22.2% under different demand levels. Note that if we consider an elastic demand user equilibrium, the travel demand function can also be approximated with another neural network and learned with the proposed framework. The simultaneous

learning of route choice preferences and demand functions will be explored in our future study.

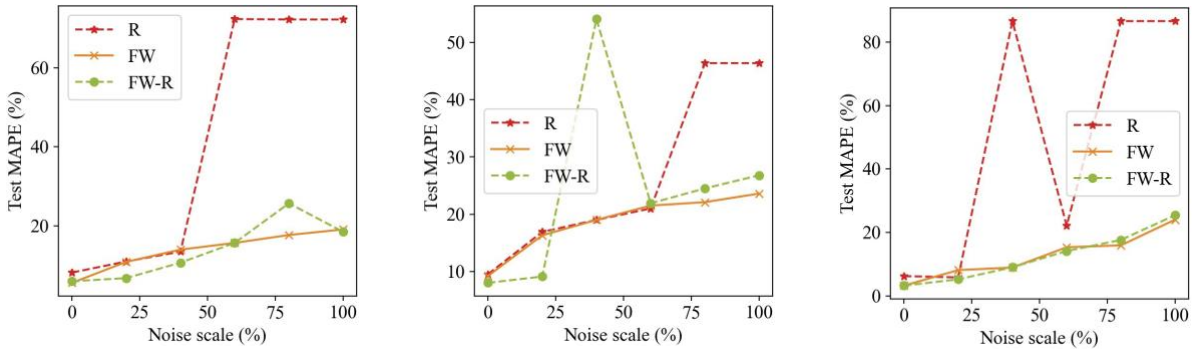


Figure 13 Model performances with demand noises under (a) base, (b) uncongested, and (c) congested demand.

The selection of feasible path sets can be tricky when no information about path choices is available. We examine the model performances when the selection of feasible paths is different from travelers' actual path choices. There are 1,587 paths in the ground-truth path set and we consider two scenarios: one with an incomplete path set of 1,058 paths and the other with a redundant path set of 2,645 paths. FW-R has the best performance when the selection of feasible paths is inaccurate. As shown in Figure 14, an incomplete path set increases the MAPE by 8.0% under base demand, compared with an increase of 2.9% induced by a redundant path set. Since an incomplete path set yields more negative effects, one can start with a large feasible set with sufficient feasible paths and gradually reduce it during training.

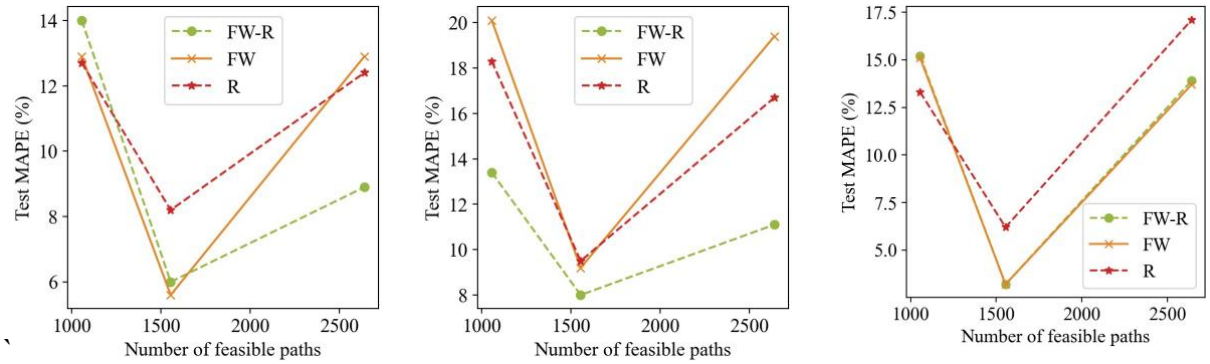


Figure 14 Effects of inaccurate feasible path sets under (a) base, (b) uncongested, and (c) congested demand.

To sum up, the proposed framework is robust to incomplete observations and input noises. More specifically, the combined method (i.e., FW-R) is more robust when only a proportion of links are equipped with sensors or no information about path choice is available. The fixed-point iteration method (i.e., FW) is preferred when the input OD demands are poorly estimated.

4.4 Example 4: Learn Demand and Supply component on Chicago Sketch

We consider a path-based formulation on the Chicago Sketch with 2,950 links, 933 nodes, and 2,493 OD pairs. Each OD pair has three feasible paths, and the feasible path set is assumed as prior information. We scale the default demand in Stabler (2023) by a factor of five and use it as the maximum possible OD demand. The following inverse demand function is used and the ground-truth BPR function is assumed with a context-dependent capacity for each link $a \in A$:

$$cap_a(x) = cap_a^0 \cdot (\alpha_c \cdot e^x + \beta_c)$$

where cap_a^0 is the default capacity; α_c = 1.5 and β_c = 1.4. The dataset contains 258 training, 64 validation, and 64 testing samples. We assume the cost function is known and focus on learning the inverse demand function and link performance function. Mirror descent with a forward step of $N = 10$ and ITD are used in training.

The end-to-end framework is set to learn the inverse demand function, the link performance function, or both, using either a model-based or a model-free approach. In the model-free setting, the inverse demand function is approximated using the residual neural networks specified in the Sioux Falls example. We employ a physics-informed neural network to learn the link-performance function. We retain the functional form of the BPR function and approximate the context-dependent capacities using neural networks with three layers and eight neurons each. Additionally, both link time and flows are assumed observable, enabling modelers to include either or both of these observations in the loss function We consider two benchmarks with fixed capacities in the standard BPR function. Benchmark-1 scales the default demand with a factor of 3.56 and achieves the best match to observed flows (29.5%) with a high time error of 160.5%. Benchmark-2 scales the default demand with a factor of 1.4 and achieves the best match to observed time (5.1%) with a high flow error of 68.9%.

Table 6 shows the performance of the end-to-end framework with different learnable components and loss functions. Scenarios yielding the lowest errors are marked: a single star denotes the best model-free scenario, while double stars indicate the best model-based one. The joint calibration of supply and demand-side components proves important. Both Functional and Residual scenarios, when adjusting both sides, yield the lowest time and demand errors. The Functional scenario has the lowest flow error of 18.1% and time error of 2.6%, while the Residual scenario generates comparable results of 23.3% and 8.2%. Incorporating flow observations into the loss function in general outperforms the use of link time. Nevertheless, using link time observations can help avoid overfitting when the link performance function can be adjusted. Overall, the complexity of training escalates with the size of the road network. The Chicago sketch example has higher errors than Sioux Falls and Braess, regardless of the approach used.

Table 6 WMAPE under different training settings.

Component	Approach	Loss function	Link flow	Link time	Demand
Benchmark-1	/	/	29.5	160.5	21.5
Benchmark-2	/	/	68.9	5.1	68.3
λ_θ	Functional	Flow	22.1	27.1	14.2
		Time	28.3	9.2	21.9
		Flow + time	22.1	27.1	14.1
	Residual	Flow	23.5	64.8	15.3
		Time	23.5	54.2	16.2
		Flow + time	23.3 *	65.8	15.1
τ_θ	Functional	Flow	32.9	1751.8	19.6
		Time	39.3	8.6	19.6
		Flow + time	32.8	1734.3	19.6
	Residual	Flow	34.1	15.8	19.6
		Time	36.5	16.0	19.6
		Flow+ time	32.8	15.9	19.6
λ_θ and τ_θ	Functional	Flow	18.1 **	2.6 **	7.7 **
		Time	22.3	5.1	9.5
		Flow + time	18.3	2.6	7.9
	Residual	Flow	26.9	8.2 *	13.6 *
		Time	37.6	10.4	19.0
		Flow + time	25.0	193.9	13.9

5 Validation: Empirical Data from Ann Arbor

5.1 Data Processing

This section outlines the crowdsourced data processing steps. We first construct a hypothetical graph using the trip-based travel demand model from the Southeast Michigan Council of Governments (SEMCOG) 2050 Regional Forecast, using year of 2025 as base year. We then extract traveler patterns from vehicle telemetric data provided by General Motors (GM) during evening peak hours.

Constructing the Hypothetical Graph

The Ann Arbor road network is derived from the trip-based travel demand forecast model in the SEMCOG 2050 Regional Forecast, using 2025 as the base year. The SEMCOG network includes highways and primary roads to capture major travel patterns. For this study, the network topology was provided in shapefile format, and the Ann Arbor subnetwork was selected. Two categories of link features were extracted: (i) free-flow speed and capacity, and (ii) traffic volume and travel time from the trip-based assignment outputs.

A total of 259 TAZs within Ann Arbor were initially included, along with boundary TAZs to capture cross-boundary trips (see Figure 15). Geometric and socio-economic attributes of the TAZs were extracted from the SEMCOG model. After merging TAZs with no connected links, 178 TAZs were retained for analysis, each connected with artificial connectors. The resulting graph consists of 810 nodes (178 centroids and 632 intersections) and 2,847 links (1,583 road segments and 1,264 connectors). Based on aggregated OD demand from the trip-based model, the Ann Arbor road network includes 31,668 OD pairs, of which 7,952 involve at least one boundary TAZ.

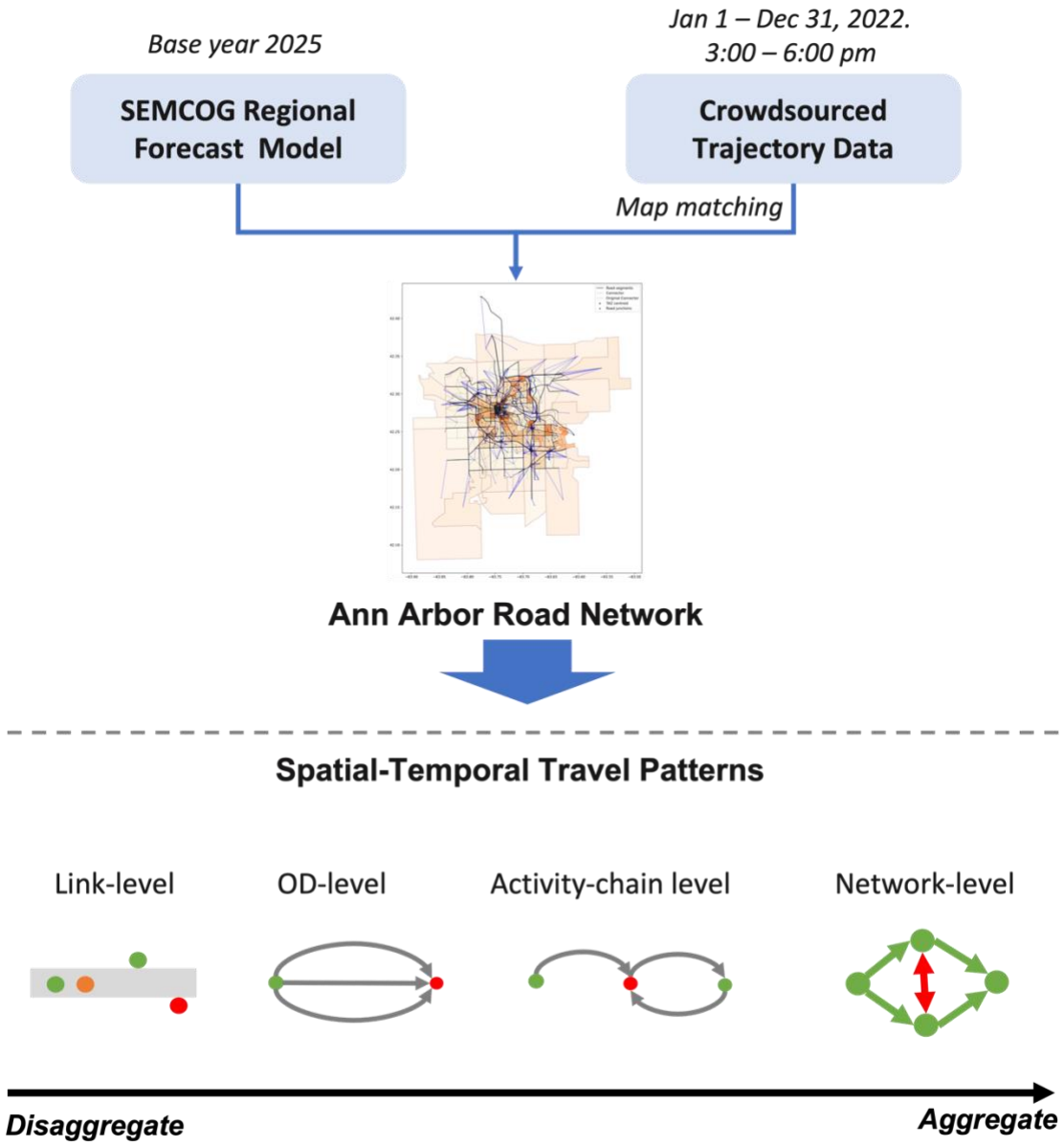


Figure 15 Data processing pipeline.

Trajectory Data and Map Matching

Crowdsourced vehicle telemetric data were provided by GM, covering evening peak hours (3:00–6:00 PM) between January 1 and December 31, 2022. To match raw trajectories with the Ann Arbor road network, we applied the map-matching algorithm from Wang et al. (2023). This process converts raw sequences of (latitude, longitude, timestamp) into link-level trajectories, recording travel times, speeds, and stop delays.

Since residential roads are excluded from the network, trajectories sometimes only partially cover links, which can lead to underestimation of travel times. To address this, link travel time was estimated using link length divided by average speed rather than raw observations.

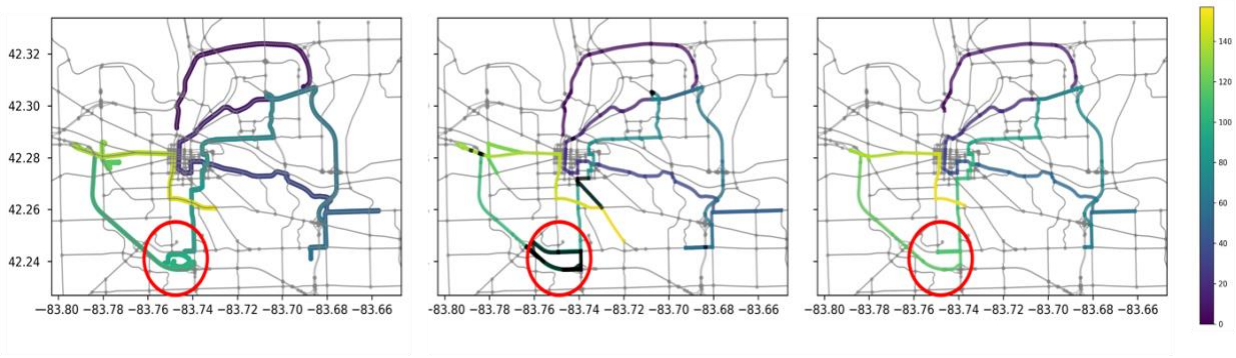


Figure 16 (Left) raw trajectory, (middle) matched trajectory using Wang et al. (2023), and (right) filtered result. Color indicates time progression, with darker colors representing earlier timestamps.

The algorithm in Wang et al. (2023) is optimized for lane-level and signal-control analysis but is less effective for capturing trip-level behavior. A trip is defined as the full sequence of links traversed between an origin and destination. As illustrated in Figure 16, raw trajectories often contain detours through residential roads (e.g., a shopping mall), which manifest as loops. To better capture trip-level behavior, we implemented a filtering process. We retained only link-level trajectories with sufficiently low matching error and removed those where the time difference from the preceding link was shorter than the free-flow travel time. When filtering disconnected a trip, the fastest path was appended to reconnect the sequence. This procedure removed approximately 9.8% of the matched link-level trajectories, producing cleaner trip-level representations.

Each record in the crowdsourced dataset corresponds to an engine start-to-stop interval, which may include multiple trips. Detours into residential roads—interpreted as access to activities—were used as splitting points. In practice, such detours often create short self-loops of length two (A–B–A). These loops were treated as access trips and used to segment longer records into subtrips. For example, Figure 17 shows a residential-to-

school pickup round trip segmented into two trips. To ensure data quality, we also removed any path whose travel time exceeded three times the shortest-path travel time for the corresponding OD pair and time slot. After processing, we identified a total of 9,382 subtrips per day during the three-hour evening peak.

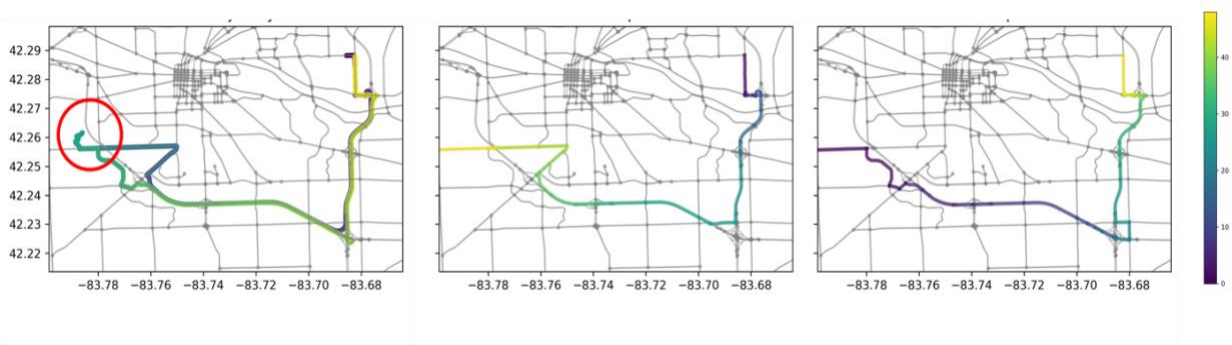


Figure 17 (Left) Raw trip with a detour to residential roads (red circle); (middle) first subtrip; (right) second subtrip. Color indicates time progression, with darker colors representing earlier timestamps.

This case study focuses on the Ann Arbor, Michigan road network. As shown in Figure 18, the network consists of 810 nodes (178 TAZ centroids and 632 intersections) and 2,847 links (1,583 road segments and 1,264 connectors). The network topology, zone attributes, and OD matrix are derived from the SEMCOG 2050 Regional Forecast.

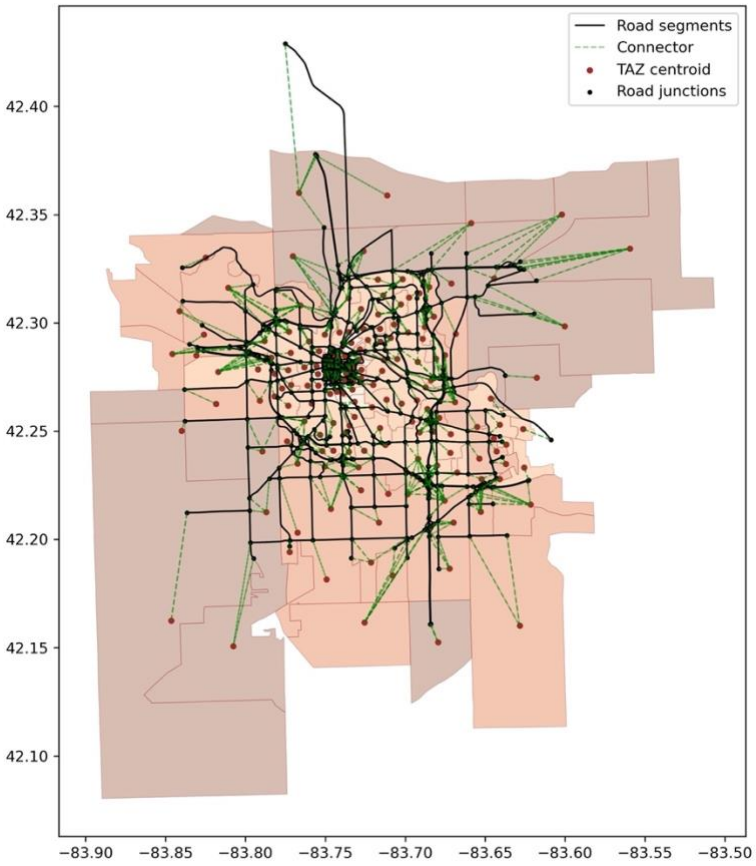


Figure 18 Ann Arbor network. Traffic analysis zones are shaded by area type; darker colors correspond to higher area type indices

GM crowdsourced trajectory data were collected during the evening peak hour (4:30–5:30 PM) throughout 2022. After applying the map-matching algorithm Wang et al. (2023), the dataset yielded an average of 3,127 observed trips per day. Each trip reports with starting/ending time, staring/ending location, list of traversed link segments and per link arrival time. These trajectories reveal variability in travel patterns, which allows us to demonstrate the proposed end-to-end framework using Ann Arbor as a case study with GM's vehicle telemetric data.

5.2 Data Analysis

The processed crowdsourced trajectory data offers insights into travel patterns in two novel ways. First, compared with traditional fixed traffic sensors such as loop detectors

or cameras, trajectory data provides more detailed information about individual travel behavior. This enables the analysis of spatial-temporal travel patterns at multiple levels of spatial aggregation, including the link, OD, activity-chain, and network levels. Second, crowdsourced trajectory data is passively collected, allowing for continuous observation over an extended period. The one-year duration enables us to capture and analyze daily variations in travel patterns and to relate these variations to external context features. We consider three categories of contextual features: weather-related, weekday-related, and accident-related. Specifically, we collect daily snow and precipitation data to represent average weather conditions and retrieve extreme weather and hazard event reports from the RITIS platform. Below, we examine spatial-temporal travel patterns revealed by the one-year crowdsourced trajectory data at the link, OD, activity-chain, and network levels.

Link-level

On the link-level, Figure 19 shows that approximately 70% of physical road segments included in the graph are observed on weekdays, compared to about 60% on weekends. The average link stop delay weighted by the number of observations and averaged across the network, and the total system travel time are both significantly higher on weekdays. Total system travel time peaks around 17:00 on weekdays and around 15:30 on weekends.

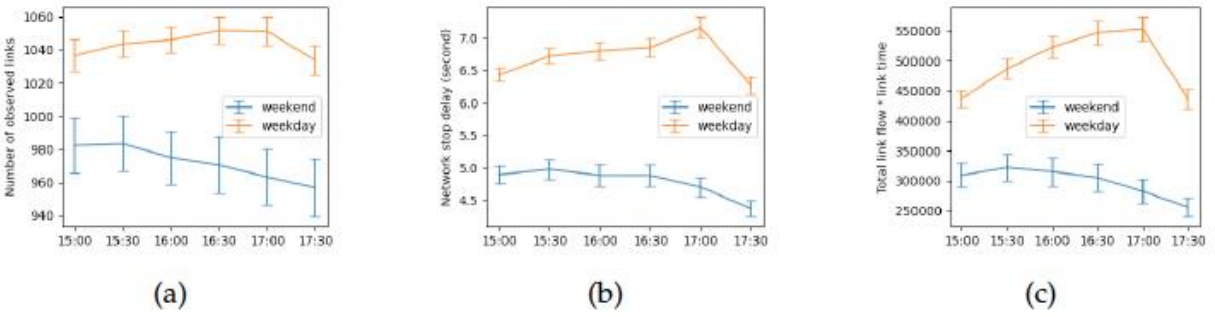


Figure 19 (a) Number of observed links, (b) average stop delay per link, and (c) total system travel time on weekdays and weekends.

Figure 20(a) shows the correlations among four link features, including curvature, highway classification, length, and maximum speed, and observed link travel patterns. Roads with higher speed limits (and thus more likely to be highways) are observed more frequently. As illustrated in Figure 20(b), highways experience an average reduction of 7.68 seconds in link travel time under snow conditions, while non-highways show a reduction of only 3.82 seconds.

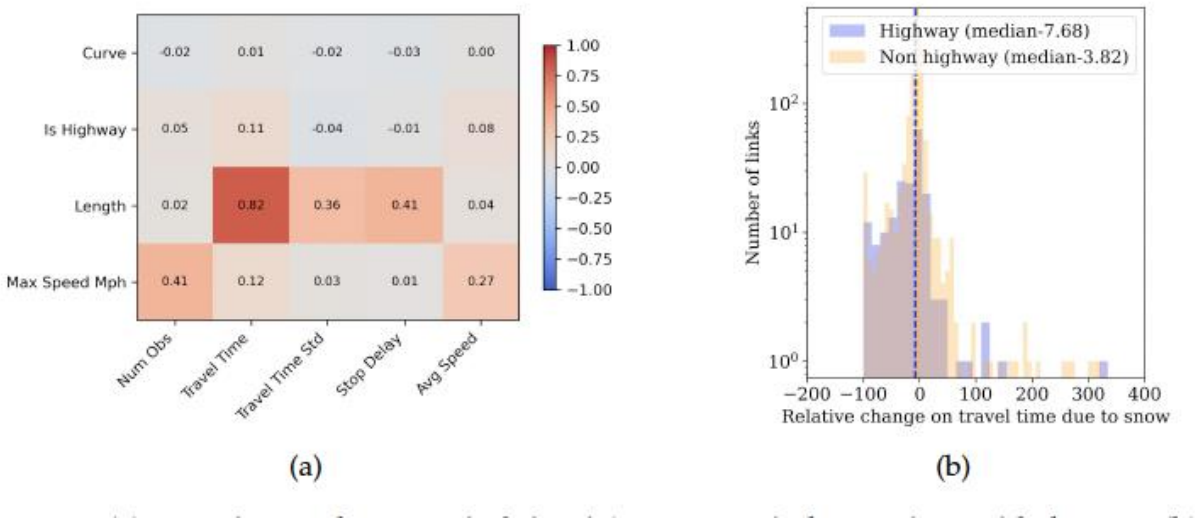


Figure 20 (a) Correlations between link-level features and observed travel behavior. (b) Comparison of snow-related reductions in link travel time for highway and non-highway segments

5.2.1 OD-level

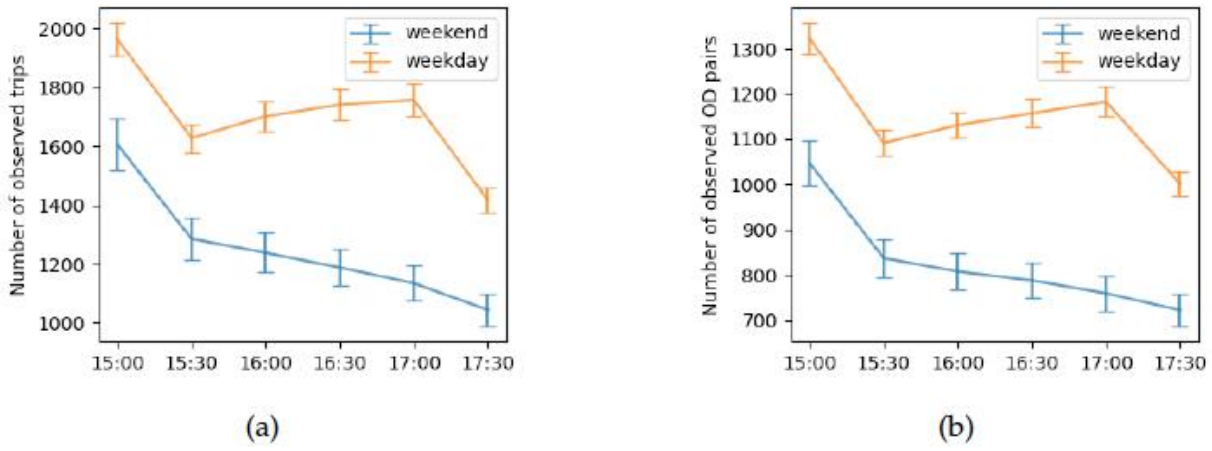


Figure 21 (a) Total number of observed trips, and (b) number of unique OD pairs.

At the OD level, Figure 21 shows that travel activity is significantly higher on weekdays than on weekends, with approximately 10,000 trips observed over three hours on weekdays, compared to around 5,000 on weekends. On weekdays, travel demand peaks first at 15:00 and again at 17:00, while on weekends, the peak occurs around 15:00. Across both cases, between 4,000 and 6,000 OD pairs are observed in the trajectory data over the three-hour period.

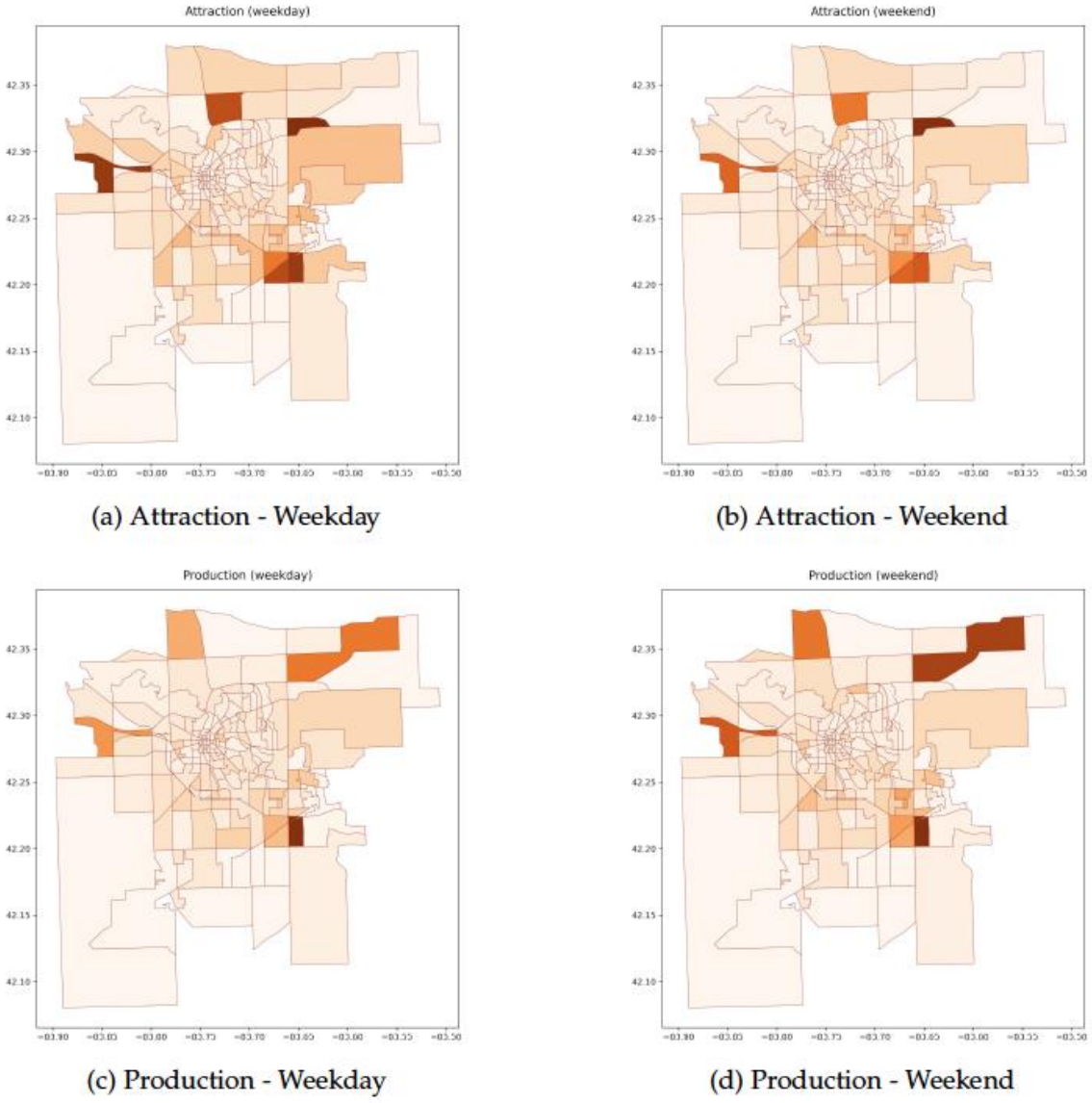


Figure 22 Spatial distribution of trip attraction (top row) and production (bottom row) on weekdays (left) and weekends (right). Darker colors indicate higher aggregated trip counts.

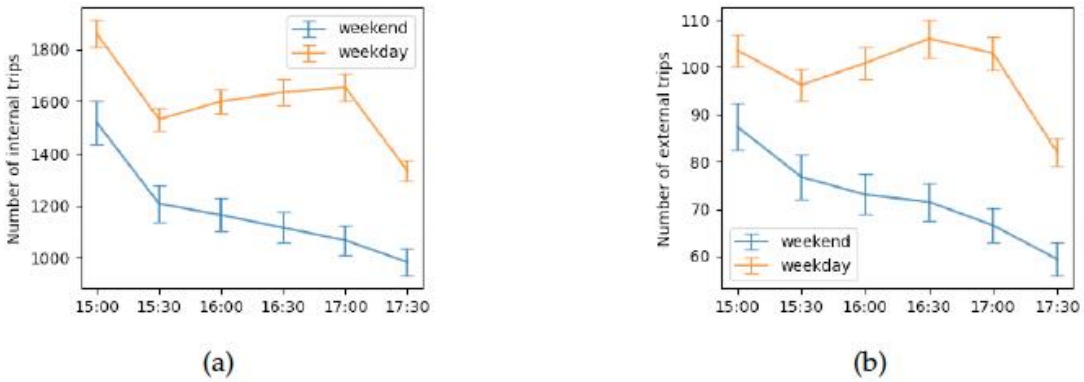


Figure 23 Temporal distribution of (a) internal and (b) external trips.

Figure 26 explores how characteristics of origin and destination TAZs influence internal and external travel demand. Internal trips are more sensitive to area type, especially that of the destination, while external trips are more strongly associated with specific land uses such as retail, trade, healthcare, and social services. On weekdays, management and entertainment-related activities show a strong association with external travel. Notably, retail-related features at the origin have a greater impact on internal travel during weekends. These features are used to select the OD pairs included in the Ann Arbor case study.

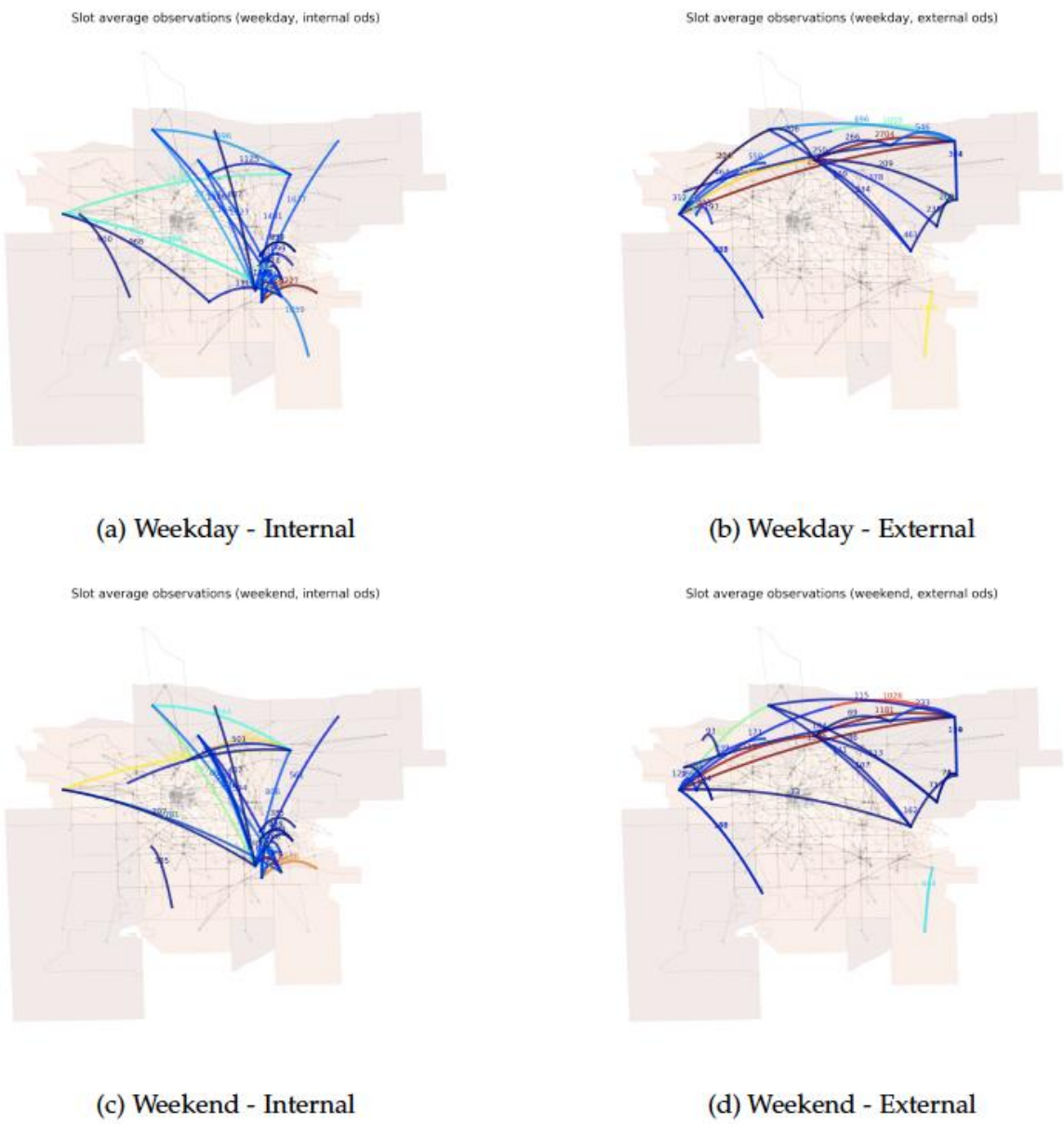
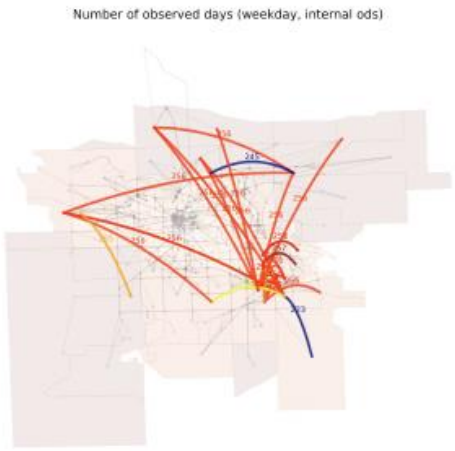
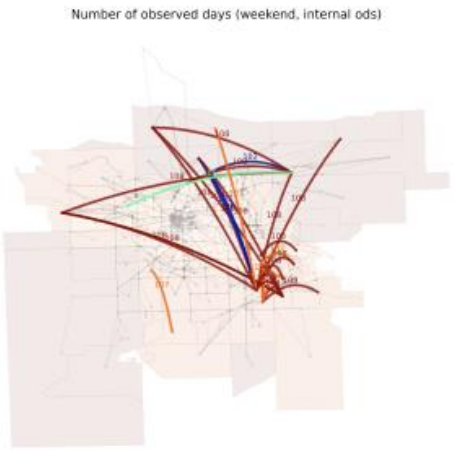


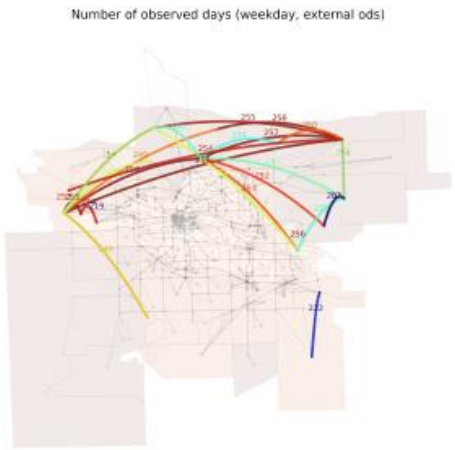
Figure 24 Mean number of OD observations for top internal and external trips by weekday



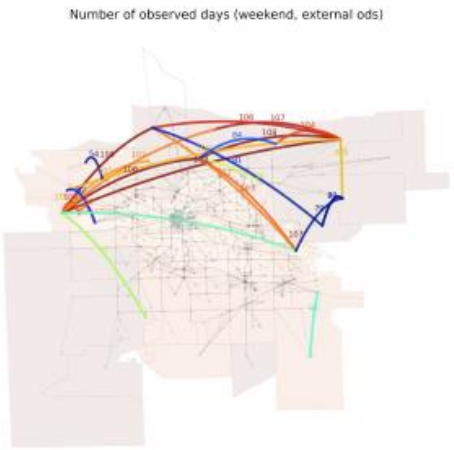
(a) Internal Trips – Weekday



(b) Internal Trips – Weekend



(c) External Trips – Weekday



(d) External Trips – Weekend

Figure 25 Mean number of observed dates for top internal and external trips by weekday

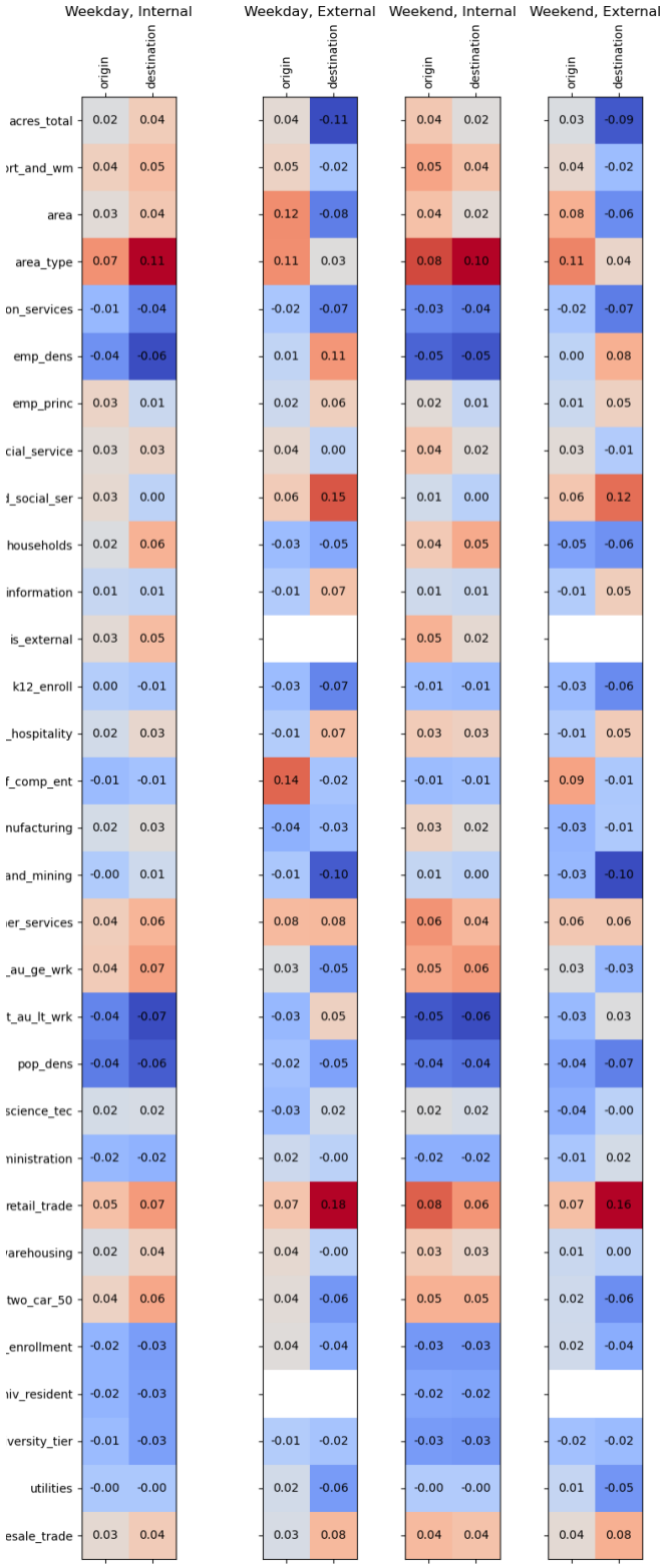


Figure 26 Correlation between OD demands and origin/destination TAZ characteristics.

One commonly used assumption in travel demand forecast is that travelers seek to minimize their travel time, resulting in route choices that follow the shortest or quickest path. This principle has long been used in travel demand modeling and planning. With trajectory data, we can directly observe the paths chosen by travelers and their experienced travel times, enabling us to empirically examine whether travelers follow shortest paths or conform to Wardrop's equilibrium behavior.

Specifically, we evaluate the equilibrium gap for each OD pair $r \in R$. The observed variables include: the set of observed paths P_r; the sampled path flow h_p for each path $p \in P_r$, where the total sampled OD demand is $q_r = \sum_{p \in P_r} h_p$; the travel time t_p for each path $p \in P_r$. The shortest observed path time is defined as t_r^*. The OD-level equilibrium gap is then calculated as the percentage absolute different between the used path and shortest path. At the network level, the equilibrium gap is computed as a demand-weighted average.

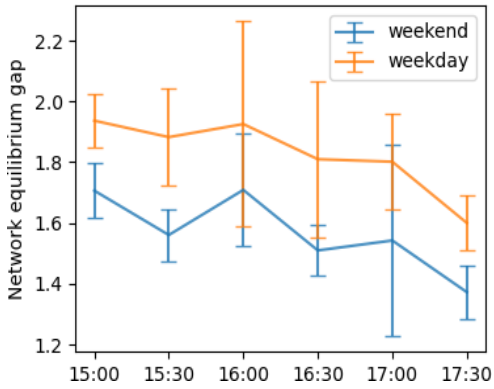


Figure 27 Network-level equilibrium gap.

Figure 27 shows that the average network-level equilibrium gap on Ann Arbor ranges from 1.5 to 2.0, indicating that travelers often select routes that are 50% to 100% longer than the shortest available path. This suggests that travelers may not be switching to alternative paths even when current paths are congested, possibly due to limited knowledge of the network or a preference for familiar paths. Interestingly, the equilibrium gap is larger on weekdays, despite the fact that more trips are likely to be routine commutes. Previous studies using taxi data from 2009 and 2014, collected over a one-

month period in the large cities of Wuhan and Shenzhen, report lower equilibrium gaps (1.2–1.4) (Chen et al., 2024). This difference is likely attributable to the use of taxi data, as taxi drivers are generally more familiar with the road network than the average commuter. Additionally, Ann Arbor is much smaller than Shenzhen and Wuhan. Therefore, an increase of 100% over the shortest path in Ann Arbor may correspond to only a 5-minute difference, whereas in larger cities, even a 40% increase could result in a substantially longer delay.

Figure 28 compares the equilibrium gaps for the top 30 internal and external OD pairs on weekdays and weekends. Longer trips tend to exhibit smaller equilibrium gaps, often below 1.2. Our analysis also finds a positive correlation between OD demand and the equilibrium gap, suggesting that higher sampled demand may be associated with greater deviation from the shortest path.

5.2.2 Activity-chain-level

One advantage of trajectory data is its ability to reveal the activity chains that travelers follow, offering deeper insight into how multiple trips are conducted. Figure 29 shows the correlation between TAZ features and activity chains. On weekdays, higher university enrollment and a higher ratio of autos to workers are positively associated with chain travel generation. On weekends, TAZs where more than 50% of households own two or more vehicles are more likely to originate chain trips. On weekdays, such TAZs are more likely to be destinations for chain travel. Additionally, weekend chains often include trips that enter and exit the Ann Arbor region, as indicated by the external flow variable. Figure 29 shows that the area types of the origin and destination are the most important factors correlated with activity chains.

5.2.3 Network-level

To conclude, Figure 30 illustrates how contextual features, including the number of accidents, hazardous weather events, precipitation, and snowfall, impact network-level travel patterns. On the demand side, snowfall significantly reduces the number of

observed internal and external trips, particularly on weekends. Accidents and hazardous weather events also contribute to reduced weekday demand. Meanwhile, accidents significantly increase network-level stop delays and total system travel time on weekdays. This analysis informs the selection of contextual features used in the Ann Arbor case study.

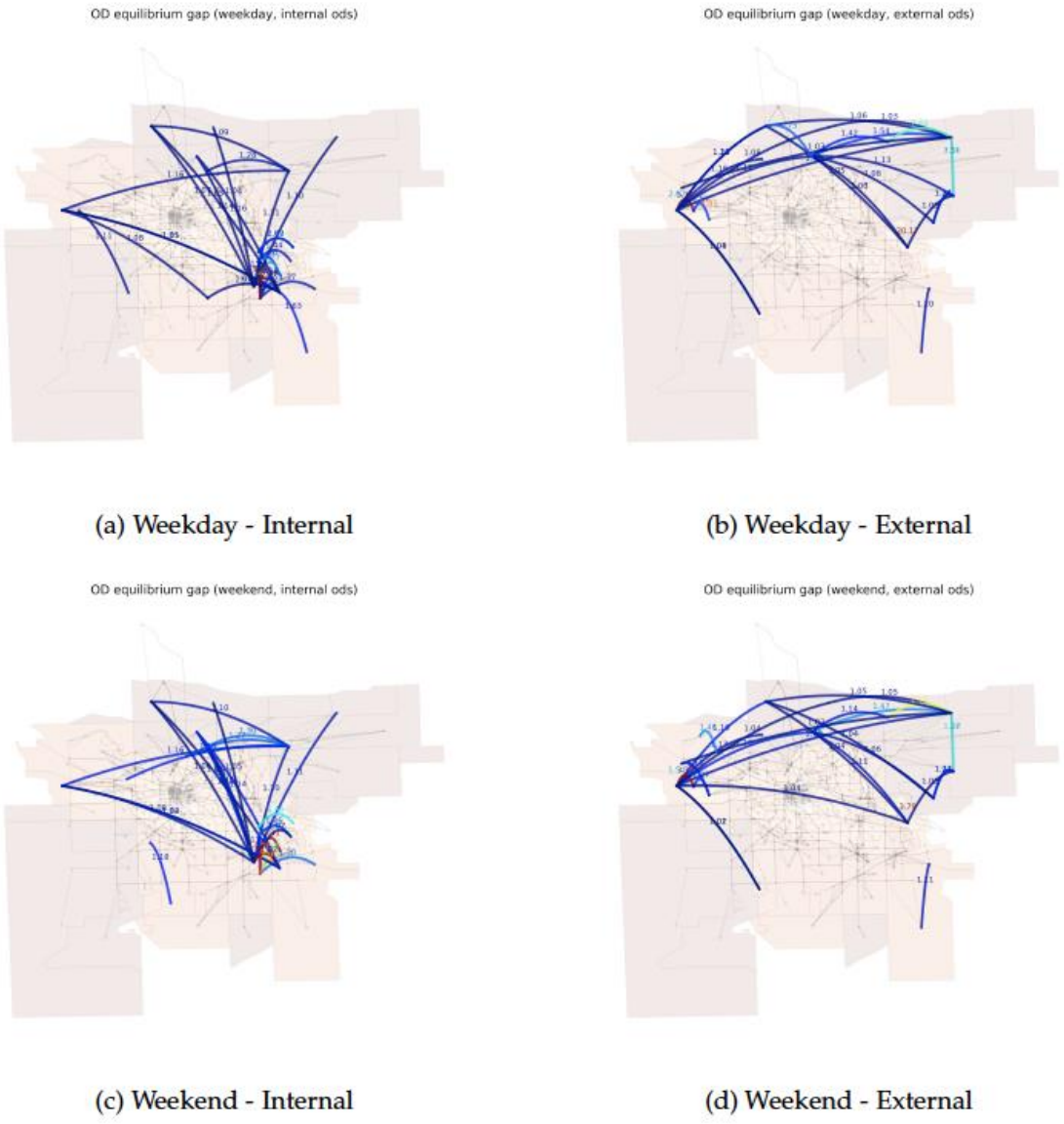


Figure 28 Mean equilibrium gap for internal and external trips by day type: (top row) weekdays, (bottom row) weekends.

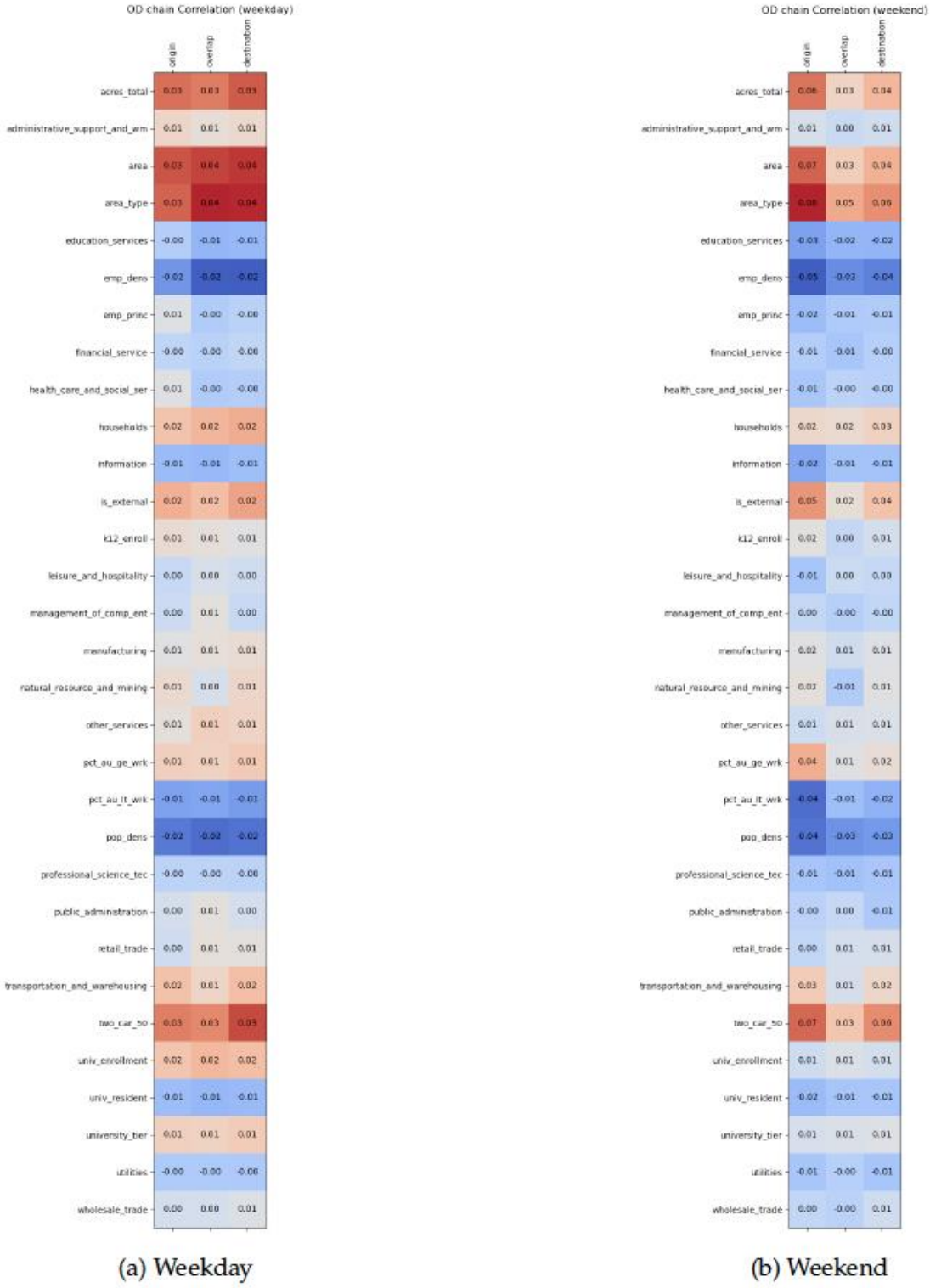


Figure 29 Correlation between chain travel demand and TAZ-level features on weekdays and weekends.

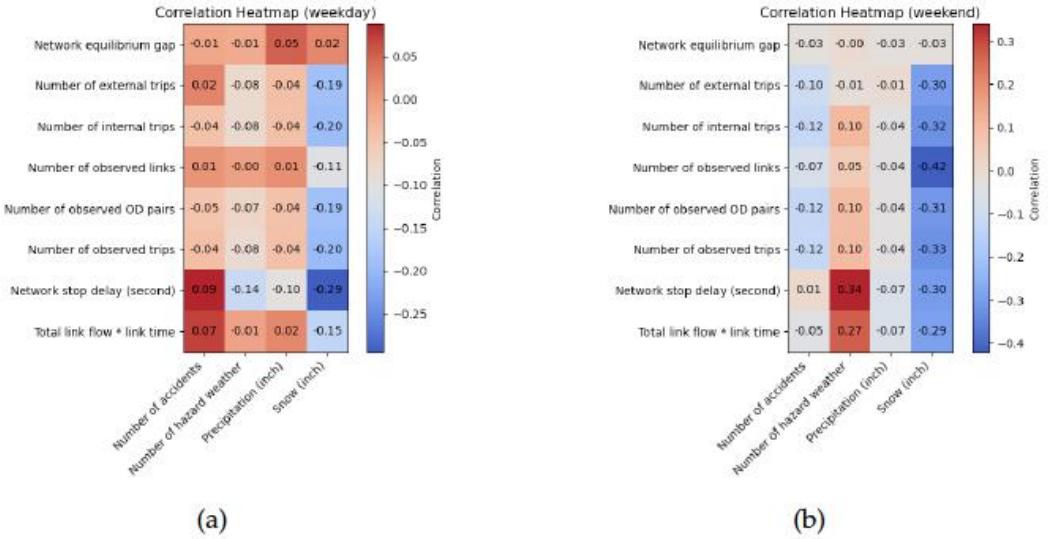


Figure 30 Correlation between network travel patterns and context features on (a) weekday

5.3 Experiment Settings

In this section, we specify the basic setting of the Ann Arbor case study. Figure 31(a) compares the numbers of observed trips on snow and non-snow days, where the x-axis represents the number of average trips between an OD pair and the y-axis indicates the number of observed OD pairs. Fewer trips are observed on snow days, and similarly on weekends, as shown in Figure 31(b). To capture these effects, context features include indicators for non-snow days and weekdays.

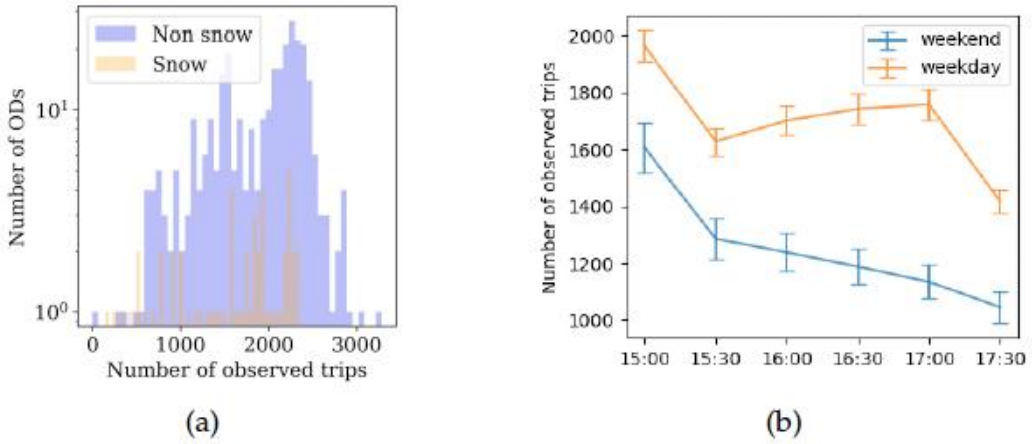


Figure 31 Differences in observed trips between (a) snow and non-snow days, and (b) weekdays and weekends.

We consider two settings covering the top 20% and 65% of total OD demands, respectively. To improve scalability and computational efficiency, we further reduce the number of links considered per OD pair by leveraging observed trajectories. Although the Ann Arbor network contains 2,847 links, travelers typically use only a subset of them. Empirical data indicate that travelers may deviate from the shortest path. Therefore, for each OD pair, we select the top six fastest paths to construct a subset of active links considered by travelers when choosing routes. This subset can be further refined by calibrating a perturbed utility model (Fosgerau et al., 2022), though estimating active link sets is beyond the scope of this project and is not discussed in detail. As shown in Table 7, using a subset of active links significantly reduces the number of variables.

Table 7 Observation statistics by number of OD pairs.

Scenario	# ODs	Max #active links	Avg # active links	Avg # ODs observed
20%	1,817	73	25.5	761.6
65%	10,418	86	27.8	1,525.2

We consider the following functional forms for the cost functions. For a physical link, the link performance function can be either linear or quadratic function of the link flow and capacity ratio plus free flow time. The free-flow travel time is computed as the link length divided by the maximum speed. The maximum speed is defined as the posted speed limit

plus 5 mph, reflecting typical driving behavior in Michigan. The context denotes the normalized snow depth to account for winter weather effects, and cap_a is the link capacity. Higher-order polynomial functions, such as the 4th-degree polynomial used in the standard BPR function, were also considered but are not listed here due to their poor empirical performance. One possible explanation is that the observed flows do not fall within the region where the flow-to-capacity ratio approaches 1.

For virtual links, the cost of the inverse demand function is selected from different forms plus the free-flow travel time of the shortest path for each OD pair. The other context vector represents normalized context features and OD-specific features. The dataset is divided into 237 training samples, 64 validation samples, and 64 testing samples. We use three-fold cross-validation and report the average performance and standard deviation across the three runs unless stated otherwise.

5.4 Presentation of Results

Because trajectory data are sampled, link travel times are more reliably observed than link flows. Therefore, Table 8 reports the link time MAPE across different numbers of OD pairs (referred to as "scenarios") and different choices of basis functions (referred to as "settings"). The benchmark context-independent flow prediction from the benchmark trip-based demand forecast model yields a link time MAPE of 83.6%. Incorporating context features and estimating context-dependent user equilibrium substantially reduces the MAPE, and prediction accuracy improves as the number of OD pairs increases. The best performance is achieved in Setting 2, which uses a linear link performance function and a quadratic inverse demand function, resulting in a MAPE of 34.3%. The reported MAPE of 34.3% also includes error due to the observed flows not strictly adhering to user equilibrium conditions. In addition, the parallel block coordinate descent algorithm scales efficiently and completes within 10 minutes.

In addition to prediction accuracy, we also examine the link and demand parameters learned under different settings. For the remainder of the discussion, we focus on the

second scenario, which covers 65% of the total demand, due to its higher accuracy. Table 9 presents the three parameters of the link performance function under different settings. The strictly positive parameters across all settings suggest that snow reduces link capacity, with a stronger impact on links with higher maximum speeds.

Table 8 Model performance across different scenarios and settings. Link time MAPEs are reported as percentages with standard deviations in parentheses.

Scenario	Setting	Link performance	Inverse demand	Link time MAPE	Time (min)
			Benchmark	83.6	/
20%	1	Linear	Linear	36.6 (1.8)	0.6
	2	Linear	Quadratic	36.1 (2.0)	0.3
	3	Linear	Linear + entropy	35.3 (1.9)	0.1
	4	Linear	Quadratic + entropy	37.2 (1.2)	0.7
	5	Quadratic	Linear	35.0 (1.0)	0.5
	6	Quadratic	Quadratic	35.6 (2.1)	0.6
	7	Quadratic	Linear + entropy	35.6 (1.4)	0.3
	8	Quadratic	Quadratic + entropy	36.3 (1.1)	0.7
65%	1	Linear	Linear	35.5 (3.4)	2.4
	2	Linear	Quadratic	34.3 (1.9)	1.9
	3	Linear	Linear + entropy	41.4 (3.6)	1.1
	4	Linear	Quadratic + entropy	36.4 (3.0)	1.4
	5	Quadratic	Linear	37.6 (2.5)	3.9
	6	Quadratic	Quadratic	36.1 (2.2)	4.2
	7	Quadratic	Linear + entropy	45.0 (4.5)	1.1
	8	Quadratic	Quadratic + entropy	36.1 (1.6)	8.2

Table 10 shows the non-zero parameters of the inverse demand function under different settings. In Setting 2, which achieves the highest predictive accuracy, the positive coefficient of 0.22 for the “Destination # veh” feature suggests that an increase in the

number of vehicles at the destination raises the inverse demand cost, resulting in higher travel demand. Similarly, the positive coefficient of 0.22 for the "Non-snow" feature indicates that demand is generally higher on non-snow days.

Setting 3, while yielding a higher link time MAPE (increasing from 34.3% to 41.4%), better captures the influence of context features. The positive coefficient of 0.11 for the "Weekday" feature in Setting 3 indicates that the end-to-end framework, in addition to recognizing reduced demand on snow days, also learns that travel demand is lower on weekends, echoing empirical trends shown. Furthermore, the greater number of non-zero parameters in Setting 3 suggests an enhanced ability to capture variations in travel patterns associated with different OD features.

Table 9 Parameters of link performance function under different settings.

Setting	Link performance	Inverse demand	Link basis	θ_1 (max speed)	θ_2 (snow)	θ_3 (max speed × snow)
1	Linear	Linear	Quadratic	1.00	0.28	0.03
2	Linear	Quadratic	Quadratic	1.00	0.37	0.09
3	Linear	Linear + entropy	Quadratic	1.00	0.14	0.03
4	Linear	Quadratic + entropy	Quadratic	1.00	0.06	0.00
5	Quadratic	Linear	Polynomial	1.00	0.21	0.03
6	Quadratic	Quadratic	Polynomial	1.00	0.17	0.03
7	Quadratic	Linear + entropy	Polynomial	1.00	0.27	0.08
8	Quadratic	Quadratic + entropy	Polynomial	1.00	0.14	0.03

Table 10 Non-zero parameters of the inverse demand function under different settings.

Setting	Link perform	Inverse demand	Demand basis	OD feature	Parameter
1	Linear	Linear	Quadratic	External OD	1.00
				Origin # retail	0.10
2	Linear	Quadratic	Polynomial	Destination # veh	0.03
				External	1.00
				Non -snow	0.22
3	Linear	Linear + entropy	Linear	Destination # veh	0.79
				Destination area type	0.81
				Destination # household	0.34
				Destination # retail	0.34
				External OD	1.00
				Origin area type	0.48
				Origin # retail	0.39
				Non-snow	0.66
				Weekday	0.11
4	Linear	Quadratic + entropy	Quadratic	Destination # veh	0.26
				Non-snow	0.01
			Entropy	Destination # household	0.52
				Destination # retail	1.00
				External OD	0.05
				Origin # retail	0.76
5	Quadratic	Linear	Quadratic	External OD	1.00
6	Quadratic	Quadratic	Polynomial	External OD	1.00
				Destination # veh	0.79
				Destination area type	0.81
				Destination # household	0.34

7	Quadratic	Linear + entropy	Linear	Destination # retail	0.34
				External OD	1.00
				Origin area type	0.48
				Origin # retail	0.39
				Non-snow	0.66
				Weekday	0.11
8	Quadratic	Quadratic + entropy	Quadratic	Destination # veh	0.01
				Non-snow	0.01
			Entropy	Destination # household	0.19
				Destination # retail	1.00
				External OD	0.24
				Origin # retail	0.65

To identify potential infrastructure improvement projects, we begin by selecting the top 20 most congested links in the network. Each selected link is assumed to allow a maximum of 50% capacity expansion. The construction cost for each link is assumed to be proportional to its maximum speed, scaled by a factor of 0.1. These candidate links are highlighted in blue in Figure 32. The total estimated cost for upgrading all 20 links is 44.93 (in units of $10,000). For this experiment, the budget constraint is set to 20, introducing a hard limit for investment planning. To simulate realistic variability in travel conditions, context features are randomly sampled from a normal distribution truncated to the interval [0, 1].

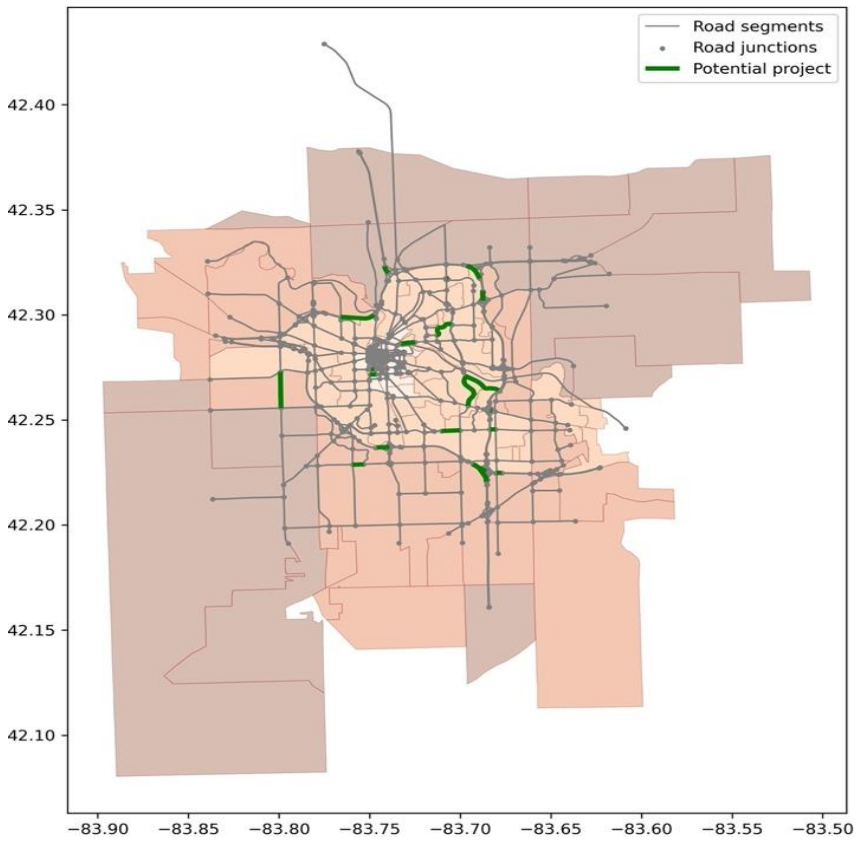


Figure 32 Candidate links for capacity expansion.

Figure 33 presents the reduction in total system travel time achieved through the proposed link upgrades. The design based on the learned model results in measurable improvement in overall network efficiency.

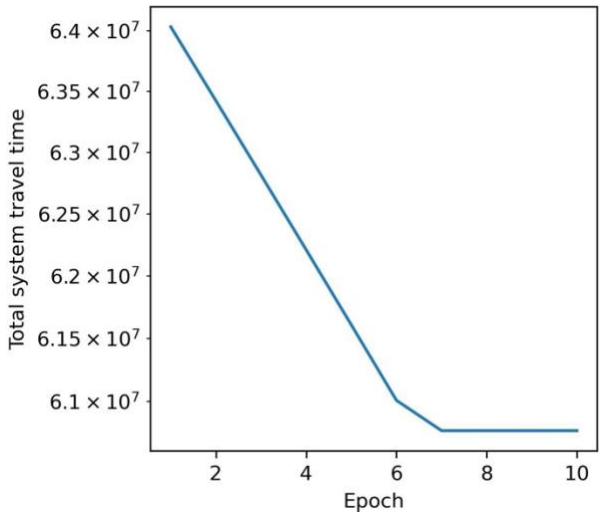


Figure 33 System travel time before and after design implementation

Using the learned inverse demand function and subject to the budget constraint, the recommended infrastructure design is illustrated in Figure 34. The color map visualizes the investment allocation across candidate links, with intensity representing relative importance under the optimization.

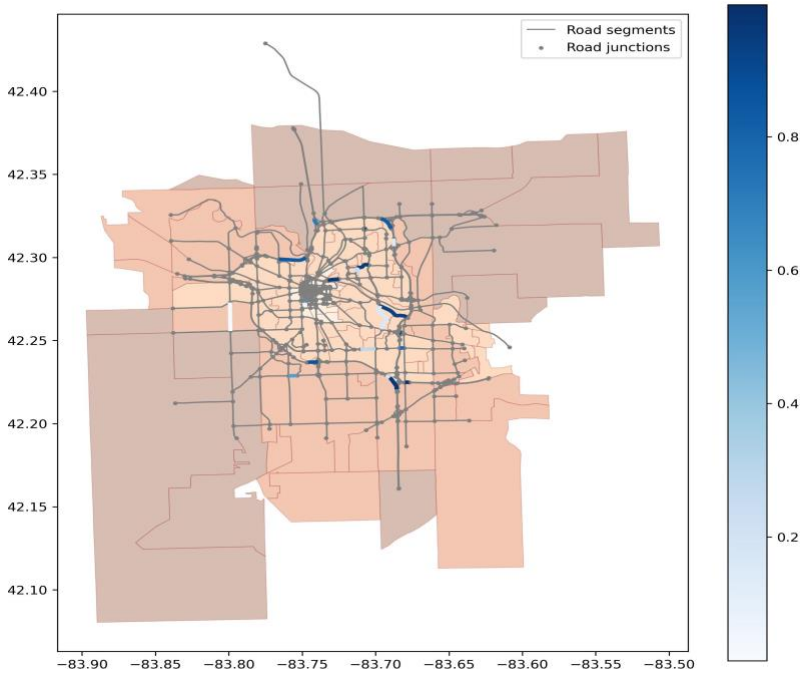


Figure 34 Recommended investment levels across candidate links

6 Discussion

6.1 Validity of hypotheses

The results of this project provide strong evidence supporting the proposed hypotheses.

Hypothesis 1: End-to-end learning improves predictive accuracy.

The findings confirm that the end-to-end framework significantly improves predictive performance relative to traditional four-step models. In the Ann Arbor case study, the framework reduced link travel time prediction errors from 83.6% in the benchmark model to 34.3%. The ability to simultaneously estimate supply- and demand-side parameters led to consistent accuracy across varying network conditions, including changes in topology, incomplete data, and noisy inputs.

Hypothesis 2: Data-driven integration enhances behavioral realism.

Empirical analyses on Ann Arbor dataset demonstrate that the framework effectively captures traveler behavior, particularly variations in route choice and responses to congestion. By incorporating trajectory and multi-source data, the model successfully identified travel reductions on weekends and during snow events, patterns that conventional models often fail to capture. This validates the role of emerging data sources in enriching behavioral representation.

Hypothesis 3: Unified calibration reduces error propagation.

The simultaneous calibration of supply and demand components minimized the cascading errors common in traditional sequential models. By embedding both components within a single optimization structure, the framework produced more stable and consistent equilibrium estimates. This reduction in error propagation strengthens the reliability of the predictions.

Hypothesis 4: Practical applicability to planning organizations.

The end-to-end framework demonstrates direct applicability to planning agencies such as MDOT, SEMCOG, and WATS MPO. The Ann Arbor case study illustrates its potential as a decision-support tool, capable of evaluating strategies including capacity expansion. The ability to integrate multi-source data and automate calibration enhances its practicality for operational use, offering planning organizations a means to make more informed, data-driven policy decisions.

Overall, the empirical findings validate the core hypotheses, showing that the proposed end-to-end framework not only advances methodological innovation but also provides practical value for transportation planning and management.

6.2 Factors affecting the results

Several factors influence the performance of the proposed learning and design framework. First, random noise in link flow observations poses challenges for parameter estimation. Because multiple parameter configurations may approximate observed data equally well, stochastic variability can lead to deviations in the estimated parameters and, consequently, in model predictions.

Second, the availability of multi-day observations strongly impacts generalizability. In principle, the framework can generalize effectively if it observes diverse context features and travel patterns across multiple cities and over sufficiently long periods. In practice, however, assembling such comprehensive datasets may require years of data collection. Given current crowdsourced data availability, one strategy can enhance generalization for "what if" analyses: leveraging existing travel demand forecast models, like the SEMCOG benchmark model used in the Ann Arbor case study.

Third, the richness and diversity of context features directly affect identifiability. Networks with limited variability in contextual factors (e.g., weather, demand fluctuations, special events) may restrict the framework's ability to distinguish between

competing behavioral responses, thereby reducing the discriminatory power of the learning process.

6.3 Implications

The results have important implications for both methodological development and policy design.

Automated Model Construction with Crowdsourced Data.

The proposed framework demonstrates that readily available crowdsourced data can be systematically leveraged to automate the calibration of supply- and demand-side components. This reduces the reliance on costly, time-consuming data collection efforts and enables agencies to build and update models more efficiently.

Vehicle connectivity and automation will make trajectory data more readily available. Leveraging this dataset, the proposed modeling paradigm, if successful, can potentially help metropolitan planning organizations and traffic authorities in the US better plan and manage their traffic networks to reduce traffic congestion and vehicle emissions, without requiring new investment in expanding the existing infrastructure. With more and more connected vehicles, we believe that the solution would transform the existing paradigm of transportation systems planning and management and has a great potential for widespread market adoption.

Enhanced Predictive Accuracy in Travel Demand Forecasting.

By integrating multi-source data and jointly estimating supply and demand, the framework substantially improves the accuracy of network equilibrium predictions. Improved forecasts of travel demand and network performance allow planners to better anticipate congestion patterns and evaluate the impacts of alternative policy or infrastructure scenarios.

Support for More Robust Transportation Network Planning

The framework functions as a decision-support tool for planning organizations such as MDOT, SEMCOG, and WATS MPO. Its ability to combine data-driven learning with equilibrium modeling provides a rigorous basis for evaluating strategies such as capacity expansion, congestion pricing, and demand management, ultimately enabling more resilient and cost-effective transportation planning.

6.4 Limitations and future work

While the proposed framework shows promise, several limitations remain and present opportunities for future research.

Time of Day and Travel Mode Modeling.

The current framework produces only a single average travel pattern and does not capture time-of-day variations in traffic congestion. This limitation reduces its effectiveness for applications that require a more detailed understanding of travel dynamics, and the formation and dissipation of traffic congestion. Capturing these dynamics is essential for improving the realism and policy relevance of the end-to-end learning framework. Another valuable direction for future research would be to integrate this framework as an input into dynamic traffic assignment (DTA) or microsimulation models.

While the proposed framework is primarily developed for vehicle assignment, future work could explore its applicability to transit and non-motorized travel data, as well as its extension to multimodal metropolitan transportation planning by encoding Logit models as softmax layer.

Computational Efficiency and Software Development.

Although the framework achieves strong predictive accuracy, computational efficiency

remains a challenge, particularly for large-scale networks and scenario analyses. Future work will focus on improving algorithmic efficiency and developing a robust, user-friendly software package that can be readily applied by planning organizations. Such tools would facilitate adoption in practice and reduce the technical burden for agencies.

Data Limitations and Robust Policy Making.

The present study is constrained by the availability and quality of empirical data. As more comprehensive multi-day and multi-source datasets become available, the framework can be refined to deliver more reliable insights. A promising direction is to extend the methodology toward **robust policy making**, where the framework not only estimates current conditions but also prescribes improvement schemes—such as capacity expansion or congestion pricing—under data uncertainty. This raises additional challenges in solving end-to-end optimization problems with discrete decision variables (e.g., selecting projects to build or upgrade), which will require methodological innovations in optimization and machine learning. Moreover, an additional limitation is representativeness of data. Traditional household travel survey data can be compared to Census data to understand representativeness. Relying on a single vehicle manufacturer may miss many different types of users and can present bias that is hard to identify and correct. There is also value in evaluating how effectively the proposed framework performs in forecasting and scenario analysis.

There would be value in testing how well this tool performs in forecasting and scenario analysis. Beyond adjustments to the network, future work could also explore how planners can best utilize this tool to evaluate different future scenarios.

7 Conclusions

This project advances travel demand forecast model by introducing an end-to-end framework that directly constructs lightweight travel demand models or integrated network equilibrium models from aggregate traffic observations.

The outputs, outcomes, and impacts of this study are summarized as follows:

Research Outputs

- Publication*:* Liu, Zhichen, and Yafeng Yin. "End-to-end learning of user equilibrium: Expressivity, generalization, and optimization." Transportation Science (2025).

- *Poster:* End-to-end learning of user equilibrium: Expressivity, generalization, and optimization. Transportation Research Board Annual Meeting. Washington, D.C. 2024.

Research Outcomes

The framework encodes unknown supply- and demand-side components as parameterized computational graphs and embeds them within a VI to enforce user equilibrium. During forward propagation, the traffic state is iteratively updated until equilibrium is reached; during backpropagation, discrepancies between estimated and observed states are used to simultaneously calibrate all parameters through auto-differentiation.

A key strength of the framework is its ability to integrate the four-step travel demand forecast within a single automated pipeline by combining domain knowledge with the representational power of neural networks. From a methodological perspective, the study addresses the main challenges of training such a unified framework. An auto-differentiation-based gradient descent algorithm was developed, leveraging computational graphs for efficiency. Forward propagation employs operator-splitting methods and differential optimization to solve batches of VI problems, while backpropagation applies iterated and inexact implicit differentiation to differentiate

through equilibrium states. Together, these advances ensure that the framework is both computationally feasible and adaptable to real-world planning contexts.

Validation on synthetic networks (Braess, Sioux Falls, and Chicago Sketch) and empirical data from the Ann Arbor network confirms the robustness and accuracy of the framework. The model achieved strong predictive performance under changes in network topology and maintained resilience in the presence of incomplete data and noisy inputs. In the Ann Arbor case study, the framework substantially reduced prediction errors in link travel time and successfully captured behavioral patterns, such as reduced travel on weekends and snow days.

A feasibility analysis further highlighted three sources of potential error: expressivity risk (misrepresentation of real-world behavior due to imperfect prior knowledge), generalization risk (poor performance on unseen data), and optimization risk (challenges in solving inverse optimization problems). Results show that, with sufficiently large neural networks and adequate data, these risks can be mitigated, leading to improved predictive performance.

Moreover, the framework represents a methodological and practical step forward in transportation planning. By unifying learning and optimization in a single data-to-decision pipeline, it offers a powerful tool for policymakers. Applied to the Ann Arbor case, the framework demonstrated the potential to reduce congestion, evaluate strategies such as capacity expansion or congestion pricing, and support more efficient resource allocation. With automated implementation, the approach can improve operational efficiency within transportation agencies, reduce costs, and guide investment decisions that maximize public benefit.

Research Impacts

From an implementation perspective, the proposed framework can be incorporated into existing transportation planning workflows to complement, rather than replace, conventional travel demand models. Whereas conventional models are resource-

intensive to construct and infrequently updated, the proposed framework produces lightweight demand models directly from passively collected data. These models can be refreshed more frequently, enabling planners to conduct light-duty analyses such as diagnosing network performance, identifying emerging congestion patterns, and prescribing short- to medium-term improvement plans and policy interventions. In this way, the framework adds agility to planning practice while preserving the value of comprehensive models for more in-depth analyses.

Agencies may consider the following phased implementation plan:

- ***Data Integration.*** Collect and organize traffic data along with relevant contextual features (e.g., weather conditions, day-of-week patterns, and socioeconomic indicators). These datasets provide the foundation for calibrating the framework and enhancing its predictive power. Another promising direction is the integration of multisource data to leverage existing survey datasets, thereby enhancing interpretability and mitigating bias.

- ***Model Training and Enhancement.*** Apply the end-to-end framework to refine and augment existing travel demand forecasting models. For example, benchmark models can be enhanced with physics-informed neural network components, thereby improving predictive accuracy while maintaining interpretability and consistency with established practices. Alternatively, specific elements such as travel time functions or route choice preferences can be calibrated directly using techniques from the proposed framework, yielding incremental improvements without the need to rebuild full models.

- ***Decision Support and Scenario Evaluation.*** Leverage the trained framework to construct lightweight travel demand models capable of rapidly prescribing and comparing candidate projects or policy interventions. Promising candidates identified through this process can then be subjected to deeper investigation using full-scale demand models. Outputs from the lightweight framework can also be incorporated into multi-criteria decision-making processes to evaluate

trade-offs among efficiency, equity, and resilience in project selection.

In practice, the framework can serve as a decision-support tool at multiple scales. Its lightweight models allow agencies to rapidly evaluate emerging conditions and update forecasts, while comprehensive, full-scale models remain valuable for in-depth, long-range analysis. Together, this dual approach offers transportation agencies a more flexible, data-driven planning toolkit to prioritize cost-effective investments, anticipate the impacts of context-specific conditions, and ultimately improve the performance and sustainability of transportation systems.

Future research may extend the proposed framework along several promising directions. First, incorporating time-of-day modeling would enable the analysis of temporal variations in travel demand and congestion, thereby advancing beyond the prediction of a single average traffic pattern to capture the dynamic evolution of traffic states. Second, extending the framework to a multimodal setting would allow for the integration of mode choice behavior and the interactions among diverse transportation modes, thereby broadening its scope beyond traffic networks to multimodal transportation networks where public transit, walking, cycling, and emerging mobility services play critical roles. Third, incorporating richer behavioral heterogeneity, such as differences in value of time, risk attitudes, and route choice preferences, would enhance the behavioral realism of the demand model and improve its predictive accuracy. Fourth, expanding empirical validation to larger and more diverse metropolitan areas would strengthen both the robustness and the generalizability of the framework across heterogeneous urban contexts. Finally, exploring more realistic and nonlinear cost structures for infrastructure investments would generate network design recommendations that better align with the complexities, trade-offs, and implementation challenges inherent in real-world decision-making.

8 Bibliography

Bai, S., Kolter, J.Z., Koltun, V., 2019. Deep equilibrium models. Advances in Neural Information Processing Systems 32.

Chen, B.Y., Chen, X.Y., Chen, H.P., Huang, Y.B., Jia, T., Lam, W.H., 2024. Understanding user equilibrium states of road networks: Evidence from two chinese mega-cities using taxi trajectory mining. Transportation research part A: policy and practice 180, 103976.

Dafermos, S., 1988. Sensitivity analysis in variational inequalities. Mathematics of Operations Research 13, 421–434.

Feng, Z., Narasimhan, H., Parkes, D.C., 2018. Deep learning for revenue-optimal auctions with budgets, in: Proceedings of the 17th International Conference on Autonomous Agents and Multiagent Systems, pp. 354–362.

Fioretto, F., Mak, T.W., Van Hentenryck, P., 2020. Predicting ac optimal power flows: Combining deep learning and lagrangian dual methods, in: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 630–637.

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

Franceschi, L., Frasconi, P., Salzo, S., Grazzi, R., Pontil, M., 2018. Bilevel programming for hyperparameter optimization and meta-learning, in: International conference on machine learning, PMLR. pp. 1568–1577.

Gao, T., Gao, H., 2022. On the optimization and generalization of overparameterized implicit neural networks. arXiv preprint arXiv:2209.15562 .

Ghadimi, S., Wang, M., 2018. Approximation methods for bilevel programming. arXiv preprint arXiv:1802.02246.

Golowich, N., Rakhlin, A., Shamir, O., 2018. Size-independent sample complexity of neural networks, in: Conference On Learning Theory, PMLR. pp. 297–299.

Grohs, P., Kutyniok, G., 2022. Mathematical aspects of deep learning. Cambridge University Press.

Guarda, P., Battifarano, M., Qian, S., 2023. Estimating network flow and travel behavior using day-to-day system-level data: A computational graph approach. Available at SSRN 4490930 .

Guarda, P., Qian, S., 2022. Statistical inference of travelers' route choice preferences with system-level data. arXiv preprint arXiv:2204.10964 .

Heaton, H., McKenzie, D., Li, Q., Fung, S.W., Osher, S., Yin, W., 2021. Learn to predict equilibria via fixed point networks. arXiv preprint arXiv:2106.00906 .

Ji, K., Yang, J., Liang, Y., 2021. Bilevel optimization: Convergence analysis and enhanced design, in: International conference on machine learning, PMLR. pp. 4882–4892.

Kidger, P., Lyons, T., 2020. Universal approximation with deep narrow networks, in: Conference on learning theory, PMLR. pp. 2306–2327.

Krueger, D., Caballero, E., Jacobsen, J.H., Zhang, A., Binas, J., Zhang, D., Le Priol, R., Courville, A., 2021. Out-of-distribution generalization via risk extrapolation (rex), in: International Conference on Machine Learning, PMLR. pp. 5815–5826.

Lawphongpanich, S., Hearn, D.W., 2004. An mpec approach to second-best toll pricing. Mathematical programming 101, 33–55.

Li, J., Yu, J., Nie, Y., Wang, Z., 2020. End-to-end learning and intervention in games. Advances in Neural Information Processing Systems 33, 16653–16665.

Li, J., Yu, J., Wang, Q., Liu, B., Wang, Z., Nie, Y.M., 2022. Differentiable bilevel programming for stackelberg congestion games. arXiv preprint arXiv:2209.07618 .

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.

Lo, H.K., Tung, Y.K., 2003. Network with degradable links: capacity analysis and design. Transportation Research Part B: Methodological 37, 345–363.

Lu, J., Li, C., Wu, X.B., Zhou, X.S., 2023. Physics-informed neural networks for integrated traffic state and queue profile estimation: A differentiable programming approach on layered computational graphs. Transportation Research Part C: Emerging Technologies 153, 104224.

Ma, W., Pi, X., Qian, S., 2020. Estimating multi-class dynamic origin-destination demand through a forward-backward algorithm on computational graphs. Transportation Research Part C: Emerging Technologies 119, 102747.

Nakkiran, P., Kaplun, G., Bansal, Y., Yang, T., Barak, B., Sutskever, I., 2021. Deep double descent: Where bigger models and more data hurt. Journal of Statistical Mechanics: Theory and Experiment 2021, 124003.

Patriksson, M., 2004. Sensitivity analysis of traffic equilibria. Transportation Science 38, 258–281.

Patriksson, M., 2008. On the applicability and solution of bilevel optimization models in transportation science: A study on the existence, stability and computation of optimal solutions to stochastic mathematical programs with equilibrium constraints. Transportation Research Part B: Methodological 42, 843–860.

Pinkus, A., 1999. Approximation theory of the mlp model in neural networks. Acta numerica 8, 143–195.

Shen, Z., Liu, J., He, Y., Zhang, X., Xu, R., Yu, H., Cui, P., . Towards out-of-distribution generalization: A survey. arxiv 2021. arXiv preprint arXiv:2108.13624 .

Sifringer, B., Lurkin, V., Alahi, A., 2020. Enhancing discrete choice models with representation learning. Transportation Research Part B: Methodological 140, 236–261.

Stabler, B. 2023. Transportation networks. https://github.com/bstabler/. Transportation Networks

Tobin, R.L., Friesz, T.L., 1988. Sensitivity analysis for equilibrium network flow. Transportation Science 22, 242–250.

Travacca, B., El Ghaoui, L., Moura, S., 2020. Implicit optimization: Models and methods, in: 2020 59th IEEE Conference on Decision and Control (CDC), IEEE. pp. 408–415.

Wang, X., Jerome, Z., Zhang, C., Shen, S., Kumar, V.V., Liu, H.X., 2023. Trajectory data processing and mobility performance evaluation for urban traffic networks. Transportation Research Record 2677, 355–370.

Wang, Y., Ma, X., Liu, Y., Gong, K., Henricakson, K.C., Xu, M., Wang, Y., 2016. A two-stage algorithm for origin-destination matrices estimation considering dynamic dispersion parameter for route choice. PloS one 11, e0146850.

Wu, X., Guo, J., Xian, K., Zhou, X., 2018. Hierarchical travel demand estimation using multiple data sources: A forward and backward propagation algorithmic framework on a layered computational graph. Transportation Research Part C: Emerging Technologies 96, 321–346.

Yang, H., Huang, H.J., 2005. Mathematical and economic theory of road pricing. Emerald Group Publishing Limited.

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

Yao, H., Tang, X., Wei, H., Zheng, G., Li, Z., 2019. Revisiting spatial-temporal similarity: A deep learning framework for traffic prediction, in: Proceedings of the AAAI conference on artificial intelligence, pp. 5668–5675.