Transportation Research Part C 158 (2024) 104409

Contents lists available at ScienceDirect

Transportation Research Part C

journal homepage: www.elsevier.com/locate/trc

Estimating network flow and travel behavior using day-to-day system-level data: A computational graph approach

Pablo Guarda [a], Matthew Battifarano [a], Sean Qian [a,b,*]

[a] Department of Civil and Environmental Engineering, Carnegie Mellon University, Pittsburgh, PA, USA
[b] Heinz College, Carnegie Mellon University, Pittsburgh, PA, USA

ARTICLE INFO

Keywords:
Computational graphs
Network modeling
Logit-based stochastic user equilibrium
Origin–destination demand estimation

ABSTRACT

To estimate network flow and travel behavior under recurrent traffic conditions, we leverage computational graphs and multi-source system-level data and solve a single-level optimization problem consistent with stochastic user equilibrium under logit assignment (SUELOGIT). Our model learns time-specific O-D matrices and utility functions and network flow parameters such as link flows, path flows, and travel times. To increase the model's representational capacity for reproducing observed link flows and travel times, the parameters of the link performance functions are assumed link-specific. More importantly, the utility function in the route choice model is enriched with (i) link-specific parameters to capture the effect of unobserved attributes on route choices and (ii) period-specific parameters weighting the observed features in the utility function to capture the heterogeneity of travelers preferences among periods of the day. Experiments on synthetic data show that the parameters of the models can be consistently recovered and that the solution of the model satisfies the SUELOGIT conditions with high accuracy. The estimation procedure is also robust to random noise in both the observed traffic flow and travel time, and it requires very few hyperparameter tuning. Subsequently, the algorithm is deployed at a large scale using real-world multi-source data in Fresno, CA, with hourly data collected during the morning and afternoon peak periods of October 2019. The utility function includes link-specific effects and attributes such as travel time, the standard deviation of travel time, the number of traffic incidents, and socio-demographic information obtained from the US Census. We obtain estimates for the total trips and travelers' utility function by hour of the day and for the average values of the link performance parameters that are reasonable and informative on the demand and supply characteristics of the transportation network.

1. Introduction

With assumptions on network flow routing behavior, e.g., logit-based stochastic user equilibrium models (SUELOGIT), traffic flow and travel time data have been used to estimate origin–destination (O-D) matrices (Cascetta and Postorino, 2001; Fisk, 1989; Krishnakumari et al., 2020; Ma and Qian, 2018a; Yang et al., 2001), the travelers' utility function of route choice models (Guarda and Qian, 2022; Yang et al., 2001) and link performance functions (Russo and Vitetta, 2011; Kurmankhojayev et al., 2022; García-Ródenas and Verastegui-Rayo, 2013; Suh et al., 1990; Wollenstein-Betech et al., 2022). Despite the advances in estimating the parameters of transportation network models from these sources of system-level data, existing modeling approaches

* Corresponding author at: Department of Civil and Environmental Engineering, Carnegie Mellon University, Pittsburgh, PA, USA.
E-mail address: seanqian@cmu.edu (S. Qian).

https://doi.org/10.1016/j.trc.2023.104409
Received 16 April 2023; Received in revised form 29 October 2023; Accepted 5 November 2023
Available online 5 December 2023

have some limitations. First, most prior studies do not leverage system-level data collected from multiple periods to learn the parameters (Guarda and Qian, 2022; Russo and Vitetta, 2011; Wang et al., 2016). Typically a daily average of hourly traffic data is used to calibrate static network models that do not adapt to within-day and day-to-day variation of traffic data. Estimating parameters with existing cross-sectional models and observations collected from different time points is feasible but inaccurate because the models cannot capture temporal dependencies among parameters. Other limitations concern the high computational costs of solving bilevel optimization formulations and using second-order methods to learn the model parameters (Wang et al., 2016; Yang et al., 2001). The latter limits the applications of existing methods to large-scale networks and makes harder to leverage the vast amounts of spatio-temporal system-level data that is nowadays available. Path flow estimation (PFE) models have been proposed to reproduce observations of traffic counts and travel times and circumvent the computational burden of bi-level optimization (Bell et al., 1997; Sherali et al., 1994). However, a main limitation of these models is related to their solution algorithms, which must be tailored to the problem structure and are sensitive to multiple hyperparameters that are used in iterative balancing schemes. For these reasons, extending these solution algorithm to estimate other parameters of the model different than the Origin-Destination matrix can become cumbersome.

Prior literature has formulated models to jointly estimate travelers' utility function and origin–destination (ODE) matrices from traffic count data, and which induce a route choice model consistent with SUELOGIT (Robillard, 1974; Fisk, 1977; Daganzo and Sheffi, 1977; Anas and Kim, 1990; Liu and Fricker, 1996; Yang et al., 2001; Lo and Chan, 2003; García-Ródenas and Marín, 2009; Russo and Vitetta, 2011; Wang et al., 2016). Models that solve this problem, which we refer to as ODLUE, have been extended to estimate additional parameters that depict demand and supply characteristics in transportation networks. Russo and Vitetta (2011) and Caggiani and Ottomanelli (2011) estimated the shape parameters of the link performance function using traffic counts and travel time measurements consistent with SUELOGIT. To leverage the travel time data, the outer level objective of their Non-linear least squares (NLLS) problem incorporated the squared difference between predicted and observed travel times. Although results on synthetic data suggested that all parameters could be consistently recovered, little attention was paid to the parameters' identifiability and overfitting due to the additional degrees of freedom introduced in the model. ODE is known to be an underdetermined problem and subject to identifiability issues even when a full set of traffic counts is available for a transportation network (Yang et al., 2018). Naturally, this issue worsens if additional parameters are introduced in the model. Besides the problem of identifiability (also known as observability), a high model complexity may lead to poor generalization on unseen data, even if excellent goodness of fit is obtained in the training set.

Another line of research has extended the application of the ODLUE beyond the trip distribution and traffic assignment steps of the classic *four-step model* (Ortúzar and Willumsen, 2011). Under the assumption of an exogenous demand matrix and exogenous travel times, Cascetta and Russo (1997) simultaneously estimated the parameters of a category index trips generation model, the parameters associated with the distribution step and the mode-specific constants, and the travel time and cost coefficients of the utility function of a multinomial logit model of mode choice. The authors tested their method with real data collected from two Italian cities and found that the magnitude and sign of the estimated parameters were consistent with their expectations. García-Ródenas and Marín (2009) estimated the utility function parameters of a route choice model and the mode choice parameters using a nested logit model via Nonlinear Generalized Least Squares (NGLS). A bilevel optimization program was formulated and solved via an alternating optimization of the upper and inner levels problems to account for the endogeneity of travel times due to traffic congestion.

More recent work has leveraged computational graphs to learn different sets of parameters in travel demand models (Li et al., 2020; Liu et al., 2023; Wu et al., 2018). A key advantage of computational graphs is the significant reduction of the computational burden of estimating parameters in high-dimensional models thanks to the use of forward and backward propagation algorithms. Once a mathematical model is cast in a computational graph, extending it with more flexible parameterization is easier. Furthermore, model training can be performed in computationally efficient packages that are open source and facilitate large-scale implementations.

Wu et al. (2018) leveraged a computational graph to estimate the parameters of a model that accounts for the trip generation, distribution, and assignment steps of the 4-step model. Data from household travel surveys, mobile phones, and traffic counts were used to estimate these parameters, and the model was deployed on a network with more than 2500 nodes and 5000 links. This research illustrated the potential of computational graphs to reduce computational burden and integrate heterogeneous data sources that are useful to estimate travel demand models. The computational graph approach learns the model parameters via gradient-based optimization, which circumvents the use of solution algorithms that must be tailored to the choice of decision variables in the optimization problem and which are sensitive to the choice of multiple hyperparameters (Bell et al., 1997). The main drawbacks of this work are that the authors assumed that travel time is observed in all links and that it is the only attribute that impacts travelers' decisions. In an extension of this work, Ma et al. (2020) reformulate the general dynamic O-D estimation (DODE) problem through computational graphs and with multiple vehicle classes. Through a forward–backward algorithm, the model can learn a dynamic O-D matrix that satisfies the Dynamic User Equilibrium conditions. In the context of static traffic assignment, Liu et al. (2023) generalizes work by Wu et al. (2018) to learn travel choice preferences and the equilibrium state from multi-day link flow observations through implicit differentiation. Travelers in the network are let to be of different classes according to the O-D pair of their trips, and a flexible neural network architecture is used to capture the impact of changes in road network features, which can be available at the link, node, or path levels.

2. Contributions of this research

Recent studies in the transportation field have leveraged advances in the ML community to address the limitations of existing methods to learn network flows and other parameters of transportation network models. These developments include the use of implicit differentiation (Liu et al., 2023; Li et al., 2020), computational graphs (Kim et al., 2021; Ma et al., 2020; Wu et al., 2018) and automatic differentiation. Despite these advances, there are still research gaps that must be addressed. To estimate network flow characteristics with spatio-temporal system-level data, this paper enhances traditional network models with the following contributions:

1. The model learns the parameters of network flow, including link flows, path flows, travel times at equilibrium, and O-D matrices and utility functions specific to the hour of the day, by solving a single-level optimization problem with a forward–backward propagation algorithm on a computational graph. This extends work by:
    - Wu et al. (2018) where travel times are assumed exogenous and observed.
    - Ma et al. (2020) where O-D matrices are not specific to the time of day
    - Guarda and Qian (2022) where only the utility function parameters are estimated
    - Liu et al. (2023) and Guarda and Qian (2022) where the O-D matrix is assumed given
2. The utility function is enriched with (i) link-specific parameters that capture the effect of unobserved attributes on route choices and (ii) parameters specific to the time of day that are weighting the observed features in the utility function to capture the heterogeneity of travelers' preferences among periods of the day. This extends work by:
    - Wu et al. (2018) and Ma et al. (2020) where utility depends on travel time only
    - Guarda and Qian (2022) where the utility does not depend on link-specific parameters
    - Wu et al. (2018), Ma et al. (2020), Guarda and Qian (2022), Liu et al. (2023) where the link costs are not specific to the time of the day
3. The link performance function parameters are estimated and assumed link-specific to increase the model's representational capacity to reproduce observed link flows and travel times. To our knowledge, this is the first work that estimates link-specific performance functions in stochastic user equilibrium models.

Our approach enhances existing models in several ways. First, the estimation of period-specific O-D matrices and utility functions allows a representation of traffic equilibrium on an hourly basis. It is reasonable to assume that destination and route choices vary on an hourly basis because the travelers' characteristics and activities are not the same for every hour. Second, the enrichment of the utility function allows us to understand the main drivers of route choices and capture the heterogeneity of travelers' preferences in each period of the day. Third, the estimation of link-specific performance functions allows to better capture of the heterogeneity in the road characteristics and to reproduce travel times more accurately. Assuming that the link performance parameters are exogenous can introduce bias in other parameters of the model, including those related to network flows and travel behavior. Fourth, the proposed single-level optimization problem allows us to simultaneously find solutions that approach traffic equilibrium and that better reproduce multi-source day-to-day system-level data. Thanks to the flexibility of the computational graph, our model can be easily adapted to incorporate alternative specifications of the utility function or the link performance functions. Furthermore, the solution of network equilibria via a fixed point formulation makes it easier to account for multiple classes of users and performance functions in our model.

The remainder of this paper is organized as follows. Section 3 presents the formulation details of our methodology. Section 4 describes the solution pipeline and discusses practical issues for our framework. A toy network is used to illustrate the computational graph resulting from our learning problem. Section 5 conducts experiments on a mid-size network to verify the effectiveness of the proposed framework. In Section 6, we demonstrate the scalability and computational efficiency of the proposed framework using real-world data of a large-scale network. Sections 7 and 8 present our main conclusions and discuss avenues for further research. All our analyses are replicable, open-sourced, and ready to be used by the transportation community (Section 9).

3. Formulation

This section explains each component of our modeling framework.

3.1. Notation

Tables 1 and 2 present some of the notation used throughout the paper.

Table 1
Parameters of network model.

Notations	Definitions
$\mathcal{A}$	The set of all links
$\mathcal{V}$	The set of all nodes
$\mathcal{W}$	The set of O-D pairs
$\mathcal{H}$	The set of all paths
$\mathcal{H}_w$	The set of paths connecting O-D pair $w \in \mathcal{W}$
$\mathbf{M} \in \mathbb{R}^{\lvert\mathcal{W}\rvert \times \lvert\mathcal{H}\rvert}$	The path-demand incidence matrix
$\mathbf{D} \in \mathbb{R}^{\lvert\mathcal{A}\rvert \times \lvert\mathcal{H}\rvert}$	The path-link incidence matrix
$\mathbf{Q} \in \mathbb{R}^{\lvert\mathcal{V}\rvert \times \lvert\mathcal{V}\rvert}$	The O-D matrix
$\mathbf{q} \in \mathbb{R}^{\lvert\mathcal{W}\rvert}$	The dense vector associated to the O-D matrix
q_w	The demand in O-D pair $w \in \mathcal{W}$
$\bar{\mathbf{x}}^{\max} \in \mathbb{R}^{\lvert\mathcal{A}\rvert}_{\geq 0}$	The vector of links' capacities
$\bar{x}_a^{\max} \in \mathbb{R}_{\geq 0}$	The capacity of link a
$\bar{\mathbf{t}}^{\min} \in \mathbb{R}^{\lvert\mathcal{A}\rvert}_{+}$	The vector of links' free flow travel times
$\bar{t}_a^{\min} \in \mathbb{R}_{+}$	The free flow travel time at link $a \in \mathcal{A}$
$\mathbf{f} \in \mathbb{R}^{\lvert\mathcal{H}\rvert}_{\geq 0}$	The vector of path flows
$f_h \in \mathbb{R}_{\geq 0}$	The path flow on path $h \in \mathcal{H}$
$\alpha_a, \beta_a \in \mathbb{R}^{\lvert\mathcal{A}\rvert}_{\geq 0}$	The BPR function parameters associated to link $a \in \mathcal{A}$

Table 2
Parameters of route choice model.

Notations	Definitions
$\mathcal{K}_{\mathbf{Z}}$	The set of exogenous attributes in the utility function
$\mathcal{K}$	The set of all attributes in the utility function
$\mathcal{L}$	The set of travelers in the network
$\mathbf{t} \in \mathbb{R}^{\lvert\mathcal{A}\rvert}_{\geq 0}$	The vector of link travel times
$\mathbf{Z} \in \mathbb{R}^{\lvert\mathcal{A}\rvert \times \lvert\mathcal{K}_{\mathbf{Z}}\rvert}$	The matrix of exogenous link attributes' values
Z_{ak}	The value of the exogenous attribute $k \in \mathcal{K}_{\mathbf{Z}}$ at link $a \in \mathcal{A}$
$\mathbf{z}_k \in \mathbb{R}^{\lvert\mathcal{A}\rvert}$	The vector of values of exogenous link attribute $k \in \mathcal{K}_{\mathbf{Z}}$
θ_k	The utility function parameter associated to attribute $k \in \mathcal{K}$
$\theta_t \in \mathbb{R}_{\leq 0}$	The travel time parameter
γ_a	The link-specific parameter associated to link $a \in \mathcal{A}$
$\mathbf{v} \in \mathbb{R}^{\lvert\mathcal{A}\rvert}$	The vector of link utilities
v_a	The link utility associated to link $a \in \mathcal{A}$
$\mathbf{p} \in \mathbb{R}^{\lvert\mathcal{H}\rvert}_{]0,1[}$	The vector of path choice probabilities
$p_h \in \mathbb{R}_{]0,1[}$	The choice probability of path $h \in \mathcal{H}$

3.2. Preliminaries

Consider a transportation network that is characterized with a graph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$, where $\mathcal{A}$ is the set of streets or links connecting the set $\mathcal{V}$ of nodes or locations. Individuals want to travel between the set $\mathcal{W}$ of origin–destination pairs. Each O-D pair $w \in \mathcal{W}$ is connected by the path set $\mathcal{H}_w$. A model is used to fit a set $\mathcal{S}$ of samples collected from the transportation network over multiple time periods. For simplicity, each period lasts one hour and may comprise multiple samples, e.g., two samples collected at 10 am on two subsequent Tuesdays. Formally, each sample $i \in \mathcal{S}$ is associated with vectors of hourly observations of traffic flows $\bar{\mathbf{x}}_i \in \mathbb{R}^{\lvert\mathcal{A}\rvert}$ and travel times $\bar{\mathbf{t}}_i \in \mathbb{R}^{\lvert\mathcal{A}\rvert}$. These vectors may contain missing entries for a subset of links in the network. In addition, a vector $\bar{\mathbf{q}}_i \in \mathbb{R}^{\lvert\mathcal{W}\rvert}$ with the historical number of trips among the set of O-D pairs $\mathcal{W}$ may be available for the period when the sample i is collected.

3.3. Parameters

The learnable parameters of the model are:

- link flows: $\hat{\mathbf{x}} \in \mathbb{R}^{\lvert\mathcal{S}\rvert \times \lvert\mathcal{A}\rvert}$:
- trips per origin destination pair: $\hat{\mathbf{q}} \in \mathbb{R}^{\lvert\mathcal{S}\rvert \times \lvert\mathcal{W}\rvert}$
- attribute-specific parameters in utility function: $\hat{\boldsymbol{\theta}} \in \mathbb{R}^{\lvert\mathcal{S}\rvert \times (1+\lvert\mathcal{K}_Z\rvert)}$
- link-specific parameters in utility function: $\hat{\boldsymbol{\gamma}} \in \mathbb{R}^{\lvert\mathcal{A}\rvert}$
- parameters of link performance function: $\hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\beta}} \in \mathbb{R}^{\lvert\mathcal{A}\rvert}$

Other parameters that are functions of the learnable parameters are:

- output link flows: $\mathbf{x} \in \mathbb{R}^{\lvert\mathcal{S}\rvert \times \lvert\mathcal{A}\rvert}$:
- link travel times: $\mathbf{t} \in \mathbb{R}^{\lvert\mathcal{S}\rvert \times \lvert\mathcal{A}\rvert}$:
- path flows: $\mathbf{f} \in \mathbb{R}^{\lvert\mathcal{S}\rvert \times \lvert\mathcal{H}\rvert}$:
- path utilities: $\mathbf{v} \in \mathbb{R}^{\lvert\mathcal{S}\rvert \times \lvert\mathcal{H}\rvert}$

- path choice probabilities: $\mathbf{p} \in \mathbb{R}^{|S|\times|\mathcal{H}|}$

3.4. Assumptions

Below are the main assumptions used to formulate our model:

Assumption 1 (*TIme-Varying Travel Behavior*)**.** The travelers' route choice preferences and the number of trips between O-D pairs vary on an hourly basis.

As a consequence of Assumption 1, the learnable parameters $\hat{\mathbf{x}}$, $\hat{\mathbf{q}}$ and $\hat{\boldsymbol{\theta}}$ can change on an hourly basis.

Assumption 2 (*Time-Varying Equilibrium*)**.** Network flow follows stochastic user equilibrium with logit assignment on an hourly basis

Assumption 2 follows from Assumption 1. If the O-D demand and the travelers' utility function are allowed to change by the hour of the day, the equilibrium state should vary on an hourly basis. Consequently, the link flows $\hat{\mathbf{x}}$, path flows $\mathbf{f}$, and link travel times $\mathbf{t}$ can change hourly.

Assumption 3 (*BPR*)**.** The link performance functions are of BPR class, and their parameters are independent of the hour of the day.

It is reasonable to assume that the physical properties of the links are not dependent on the period of the day. Therefore, based on Assumption 3, the parameters of the link performance functions $\hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\beta}}$ are not hour-specific.

Assumption 4 (*Time Invariant Path Sets*)**.** The composition of the path set associated with each sample of observations is given and independent of the period of the day

As a result of Assumption 4, the incidence matrices $\mathbf{D} \in \mathbb{R}^{|\mathcal{A}|\times|\mathcal{H}|}, \mathbf{M} \in \mathbb{R}^{|\mathcal{W}|\times|\mathcal{H}|}$ are the same for all samples.

3.5. Constraints

The parameters estimated with the model must satisfy a series of constraints. This section derives the constraints associated with a sample $i \in S$. The parameters that are time-dependent must be indexed by i in the constraints because samples may be available for different hours of the day.

3.5.1. Conservation of network flows

As in any path-based static traffic assignment model, two sets of flow conservation constraints must be satisfied:

$$\mathbf{x}_i = \mathbf{D}\mathbf{f}_i \tag{1}$$

$$\mathbf{q}_i = \mathbf{M}\mathbf{f}_i \tag{2}$$

where $\mathbf{x}_i, \mathbf{f}_i, \mathbf{q}_i$ represents the flow in links, paths, and O-D pairs for sample i. As a consequence of Assumption 4, the incidence matrices $\mathbf{D} \in \mathbb{R}^{|\mathcal{A}|\times|\mathcal{H}|}$ and $\mathbf{M} \in \mathbb{R}^{|\mathcal{W}|\times|\mathcal{H}|}$ are not indexed by the sample i. As shown in the equations, $\mathbf{D}$ and $\mathbf{M}$ map path flows to link flows and O-D flows, respectively.

3.5.2. Route choice model

The logit model remains a gold standard to model route choices due to its good compromise between behavioral realism and mathematical tractability (McFadden, 1973, 2001). In line with the Multinomial Logit (`MNL`) model, the probability that a traveler $l \in \mathcal{L}$ chooses a path $h \in \mathcal{H}_w$ connecting the o-d pair $w \in \mathcal{W}$ is given by:

$$p_h = \text{Softmax}(\{v_j\}_{j\in H_w}) = \frac{\exp(\mu \cdot v_h)}{\sum_{j\in\mathcal{H}_w}\exp(\mu \cdot v_j)} \tag{3}$$

where v_h is the deterministic component of the travelers' utility associated with the alternative path h, and $\mu > 0$ is a scale parameter of the Extreme Value (EV) Type 1 distribution set to 1 for identification purposes. Eq. (3) can be further vectorized and written in terms of path utilities as follows:

$$\mathbf{p}_i = \frac{\exp\left(\mathbf{v}_i\right)}{\mathbf{M}^\top\mathbf{M}\exp(\mathbf{v}_i)} \tag{4}$$

where $\mathbf{v}_i, \mathbf{p}_i \in \mathbb{R}^{|\mathcal{H}|}$ are the vectors of path utilities and path choice probabilities for sample i, respectively. Because the numerator and denominator are vectors, the division is performed element-wise.

Thanks to the flexibility of the computational graph, our model works for any utility function specified by the modeler. However, a primary advantage of the `MNL` model is that path choice probabilities can be written in closed form. In addition, the path choice

probabilities are differentiable with respect to the parameters of the utility functions. This helps to increase stability during parameter optimization, and it avoids sampling from probability distributions to obtain path choice probabilities, e.g., the multinomial probit route choice model. Alternatively, our model could be extended to represent complex route choice decision rules in the network context, e.g., prospect theory (Connors and Sumalee, 2009), or to correct for the path overlapping problem, e.g., C-logit and Path Size logit. In practice, the main modification to the model would be in the definition of the utility function and the path choice probabilities presented in Eqs. (4) and (7), respectively. Bekhor and Toledo (2005) review multiple route choice models in the context of Stochastic User Equilibrium that could be implemented as alternatives to `MNL`.

3.5.3. Aggregation of individual route choices in path flows

Path flows are an aggregate of individual route choices. Therefore, path flows, and path choice probabilities can be coupled, in vectorized form, via the following equation:

$$\mathbf{f}_i = \left(\mathbf{M}^\top \hat{\mathbf{q}}_i\right) \circ \mathbf{p}_i \tag{5}$$

where $\mathbf{M}^\top \hat{\mathbf{q}}_i$ gives a vector with the total number of trips in the O-D pair associated with each path. If this vector is further multiplied by the vector of path choice probabilities $\mathbf{p}$, we obtain the path flows vector $\mathbf{f}_i$. With (5), the conservation constraint between O-D flows and path flows in (2) is directly satisfied.

3.5.4. Specification of utility function

Under the assumption of a linear-in-parameters utility function, the utility of a path $h \in \mathcal{H}$ can be decomposed in terms of the utilities of the links traversed by that path as follows:

$$v_{ih} = \sum_{h\in\mathcal{H}} \sum_{a\in\mathcal{A}} D_{ah}\left(\theta_{it} t_{ia} + \sum_{k\in\mathcal{K}_Z} \theta_{ik} Z_{iak} + \sum_{k\in\mathcal{K}_Y} \theta_{ik} Y_{iak}\right) = \sum_{h\in\mathcal{H}} \sum_{a\in\mathcal{A}} D_{ah}\left(\theta_{it} t_{ia} + \sum_{k\in\mathcal{K}_Z} \theta_k Z_{iak} + \gamma_a\right) \tag{6}$$

where θ_i is the vector of parameters of the utility function associated with sample i, Z_{iak} is the value of the exogenous attribute $k \in \mathcal{K}_Z$ in link $a \in \mathcal{A}$ and t_{ia} is the travel time of link a. Furthermore, Y_{iak} is the value of the unobserved exogenous attribute $k \in \mathcal{K}_Y$ in link a and whose joint effect is captured in a vector of link-specific parameters $\boldsymbol{\gamma} \in \mathbb{R}^{|\mathcal{A}|}$. D_{ah} is an element of the incidence matrix $\mathbf{D}$ taking the value 1 if path h traverses link a and 0 otherwise. Finally, the paths utilities associated with sample i can be written in a vectorized form as:

$$\mathbf{v}_i = \mathbf{D}^\top \left(\mathbf{t}_i \hat{\theta}_{it} + \mathbf{Z}_i \hat{\boldsymbol{\theta}}_{i,1:|\mathcal{K}_Z|} + \hat{\boldsymbol{\gamma}}\right) = \mathbf{D}^\top \left(\begin{bmatrix}\mathbf{t}_i & \mathbf{Z}_i\end{bmatrix} \begin{bmatrix}\hat{\theta}_{it} \\ \hat{\boldsymbol{\theta}}_{\mathcal{K}_Z}\end{bmatrix} + \hat{\boldsymbol{\gamma}}\right) = \mathbf{D}^\top \left(\begin{bmatrix}\mathbf{t}_i & \mathbf{Z}_i\end{bmatrix} \hat{\boldsymbol{\theta}} + \hat{\boldsymbol{\gamma}}\right) \tag{7}$$

where $\mathbf{Z}_i \in \mathbb{R}^{|\mathcal{A}|x|\mathcal{K}_Z|}$ is the matrix containing the values of the exogenous link attributes and $\hat{\boldsymbol{\theta}}_{i,1:|\mathcal{K}_Z|} \in \mathbb{R}^{|\mathcal{K}_Z|}$ is a column vector containing the parameters of the utility function weighting the exogenous attributes that are associated to sample i.

Remark 1. Note that the link-specific effects $\boldsymbol{\gamma}$ and the parameters weighting the set of exogenous attributes $\mathcal{K}_Z$ in the utility function are jointly identifiable only if data from multiple periods are available. With data from a single hour of the day, link-specific fixed effects can arbitrarily reproduce the values of the observable exogenous component of the utility function. The decision of excluding either the fixed effects or the set observable exogenous attributes from the utility function is left to the modeler.

3.5.5. Relationship between travel time and traffic flow

A standard assumption in static traffic assignment models is that the travel time in a link is a function of the traffic flow in that link only. For simplicity, we assume that the link performance function is of BPR class with parameters $\hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\beta}}$. Thus, the vector of link travel times $\mathbf{t}$ associated with sample i can be written in a vectorized form as:

$$\mathbf{t}_i = \mathrm{BPR}(\hat{\mathbf{x}}_i, \hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\beta}}) = \bar{\mathbf{t}}^{\min} \circ \left(1 + \hat{\boldsymbol{\alpha}} \circ \left(\frac{\hat{\mathbf{x}}_i}{\bar{\mathbf{x}}^{\max}}\right)^{\hat{\boldsymbol{\beta}}}\right) \tag{8}$$

where division, multiplication ($\circ$) and exponentiation operations are applied element-wise. The vectors $\bar{\mathbf{t}}^{\min}, \bar{\mathbf{x}}^{\max} \in \mathbb{R}_+^{|\mathcal{A}|}$ correspond to the free flow travel time and capacity of each link, which are assumed exogenous. To ensure the existence and uniqueness of the `SUELOGIT` solution, it is convenient that the link performance functions are strictly monotone increasing (see Theorem 2, Cantarella (1997)). Because we assume no link interactions, the previous condition can be satisfied by simply constraining the parameter space to be $\hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\beta}} > 0$.

Remark 2. Note that alternative representations of the link performance functions can also be accommodated in our model, which includes conical volume-delay functions (Spiess, 1990) or polynomial kernel functions (Wollenstein-Betech et al., 2022). The choice of link performance function will likely impact the parameter stability during model training, and it is a decision left to the modeler.

3.5.6. Network equilibrium model

In a congested network, travel times are endogenous and dependent on the traffic flows. At equilibria, traffic flows and travel times are expected to reach a stationary point where travelers have no incentives to switch to alternative paths. Different network equilibrium models can capture the interdependence among travelers' decisions. With a logit-based route choice model, Stochastic User Equilibrium under Logit assignment (SUELOGIT) is a natural choice (Fisk, 1980). At SUELOGIT, no traveler believes that utility can be improved by unilaterally changing routes (Daganzo and Sheffi, 1977), and the travelers' choice probabilities become consistent with the path flows assigned to every O-D pair (Fisk, 1980).

The SUELOGIT problem is typically formulated as a convex optimization program (Guarda and Qian, 2022). Alternatively, it can be cast as a fixed point formulation in link flow space (Gentile, 2018; Cantarella, 1997). In our case, the fixed-point problem equates to finding a solution for $\hat{\mathbf{x}}_i$ in the following equation:

$$\hat{\mathbf{x}}_i = \mathbf{D}_i \mathbf{f}_i(\mathbf{p}_i(\mathbf{v}_i(\mathbf{t}_i(\hat{\mathbf{x}}_i)))) \tag{9}$$

which is derived by substituting constraints (8), (7), (5), (4), into (1), by enforcing that $\mathbf{x} = \hat{\mathbf{x}}$ and by defining $\hat{\mathbf{x}}$ as the only learnable parameter.

3.6. Loss function

The loss function ℓ is a weighted sum of four components. The demand loss $\ell_q = d_q(\hat{\mathbf{q}}, \bar{\mathbf{q}})$, the link flow loss $\ell_x = d_x(\hat{\mathbf{x}}, \bar{\mathbf{x}})$, the travel time loss $\ell_t = d_{\bar{t}}(\mathbf{t}, \bar{\mathbf{t}})$ and the equilibrium loss $\ell_e = d_e(\hat{\mathbf{x}}, \mathbf{x})$, where d_q, d_x, d_t, d_e represent some distance metric between the inputs of the functions, e.g., the Euclidean norm. The set of hyperparameters $\lambda_q, \lambda_x, \lambda_t, \lambda_e$ weight the components $\ell_q, \ell_x, \ell_t, \ell_e$ in the loss function. Incorporating the equilibrium component in the loss function aims to satisfy the fixed point constraint in (9). Lower values of ℓ_t and ℓ_x are associated with a better capacity to reproduce observed travel times and link flows.

3.7. Optimization problem

Below is the resulting learning problem ($\mathcal{P}$). The constraints are included in vectorized form according to the derivations presented in Section 3.5. To allow the model to fit samples collected in multiple days and hours of the day, the flow conservation constraints are written for every sample $i \in \mathcal{S}$. Note that the conservation constraint (2) is excluded since it becomes redundant with the presence of constraint (5) (Section 3.5.3). Based on the assumptions discussed in Section 3.4, the only parameters that do not vary between samples are $\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}$. Note that we choose the Euclidean norm as the distance metric to compute the loss function, but other metrics are also allowable. Given that the coverage of observations may vary between samples and data sources, we normalize each loss component by $n_{\bar{x}_i}, n_{\bar{t}_i}, n_{\bar{q}_i}$, namely, the number of non-missing observations associated with the link flows, travel times and historical O-D trips for sample i, respectively.

$$\begin{aligned}
\underset{\substack{\{\hat{\mathbf{x}}_i, \hat{\mathbf{q}}_i, \hat{\boldsymbol{\theta}}_i\}_{\forall i \in \mathcal{S}}, \\ \hat{\boldsymbol{\gamma}}, \hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\beta}}}}{\text{minimize}} \quad & \lambda_x \frac{\sum_{i \in \mathcal{S}} \|\hat{\mathbf{x}}_i - \bar{\mathbf{x}}_i\|_2^2}{\sum_{i \in \mathcal{S}} n_{\bar{x}_i}} + \lambda_t \frac{\sum_{i \in \mathcal{S}} \|\hat{\mathbf{t}}_i - \bar{\mathbf{t}}_i\|_2^2}{\sum_{i \in \mathcal{S}} n_{\bar{t}_i}} + \lambda_q \frac{\sum_{i \in \mathcal{S}} \|\hat{\mathbf{q}}_i - \bar{\mathbf{q}}_i\|_2^2}{\sum_{i \in \mathcal{S}} n_{\bar{q}_i}} + \lambda_e \frac{\sum_{i \in \mathcal{S}} \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|_2^2}{|\mathcal{S}| \cdot |\mathcal{A}|} \\
\text{subject to} \quad & \mathbf{t}_i = \bar{\mathbf{t}}^{\min} \circ \left(1 + \hat{\boldsymbol{\alpha}} \circ \left(\frac{\hat{\mathbf{x}}_i}{\bar{\mathbf{x}}^{\max}}\right)^{\hat{\boldsymbol{\beta}}}\right) && \forall i \in \mathcal{S} \\
& \mathbf{v}_i = \mathbf{D}^\top \left(\begin{bmatrix} \mathbf{t}_i & \mathbf{Z}_i \end{bmatrix} \hat{\boldsymbol{\theta}}_i + \hat{\boldsymbol{\gamma}}\right) && \forall i \in \mathcal{S} \\
& \mathbf{p}_i = \frac{\exp(\mathbf{v}_i)}{\mathbf{M}^\top \mathbf{M} \exp(\mathbf{v}_i)} && \forall i \in \mathcal{S} \\
& \mathbf{f}_i = (\mathbf{M}_i^\top \hat{\mathbf{q}}_i) \circ \mathbf{p}_i && \forall i \in \mathcal{S} \\
& \mathbf{x}_i = \mathbf{D}_i \mathbf{f}_i && \forall i \in \mathcal{S} \\
& \hat{\mathbf{x}}_i, \hat{\mathbf{q}}_i \geq \mathbf{0} && \forall i \in \mathcal{S} \\
& \hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\beta}} > \mathbf{0}
\end{aligned}$$

A key difference with respect to previous work (Wu et al., 2018; Ma et al., 2020) is that link flows are defined as learnable parameters of the model. The minimization of the equilibrium component ℓ_e in the loss function induces a fixed point in link flow space, i.e., $\ell_e \to 0 \Rightarrow \mathbf{x} \approx \hat{\mathbf{x}}$, which encourages that both link flow and travel time satisfy the SUELOGIT equilibrium condition. The predicted link flows and travel times are given by $\hat{\mathbf{x}}$ and $\mathbf{t}$. Because travel times are a function of the input link flows, our model provides predictions of travel times and link flows that are internally consistent (Eq. (8)). Besides, in contrast to prior work (Wu et al., 2018), our model does not require that travel times are observed in every link. The minimization of the loss components $\ell_{\bar{x}}$ and $\ell_{\bar{t}}$ encourages that the solution matches the set of observed link flows and travel times. Because the predicted values $\ell_{\bar{x}}$ and $\ell_{\bar{t}}$ are allowed to vary within-day but not between-days, their prediction error can be interpreted as their day-to-day irreducible variance. Finally, the component $\ell_{\bar{q}}$ leverages historical O-D information to estimate the flow $\mathbf{q}$ between O-D pairs.

3.8. Illustrative example

Consider a network with five nodes and six links (Fig. 1). Assume that only pair 1–4 (q_1) and 1–5 (q_2) carry flow and that the travelers' consideration set includes the following paths: 1-2-4 (f_1), 1-3-4 (f_2), 1-2-5 (f_3), 1-3-5 (f_4). The cost functions associated with each link are of BPR class with parameters $\alpha, \beta > 0$. The travelers' utility function depends on the travel time t and on a set of exogenous attributes $\mathcal{K}_Z$ whose values vary among links but not between different hours of the day.

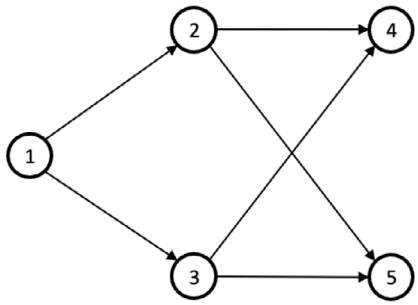


Fig. 1. Network in illustrative example.

Suppose a set $\mathcal{S}$ of samples of observations of travel times $\bar{\mathbf{t}}$ and link flows $\bar{\mathbf{x}}$ is available for some links in the network and over multiple time points. Assume that there is access to vector $\bar{\mathbf{q}}$ with historical information of the O-D flows and that the travel times and link flows are obtained under recurrent traffic conditions, i.e., a state where travelers' route choices are consistent with logit-based Stochastic User Equilibrium (SUELOGIT). Assume that the goal is jointly estimating the parameters of the utility function, the link performance functions, and the O-D flows using the available data $\bar{\mathbf{t}}$ and $\bar{\mathbf{x}}$.

Fig. 2 describes the structure of the computational graph that solves the learning problem for this small network. Consider a single sample $i \in \mathcal{S}$ of travel times and link flow observations. The first layer of the computational graph receives the vector $\hat{\mathbf{x}}_i$ of link flow parameters as input. Then, link performance functions of BPR class and with the parameters $\hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\beta}}$ are applied to $\hat{\mathbf{x}}_i$ to obtain the vector of input link travel times $\mathbf{t}_i$ (Eq. (10)). Subsequently, the vector $\mathbf{t}_i$ is appended with the matrix of exogenous attributes $\mathbf{Z}_i$, if available, and then this new matrix is weighted by a vector of parameters $\boldsymbol{\theta}_i$ to compute the observable component of the link utilities. The effect of any unobserved exogenous link attribute is captured with a set of link-specific parameters $\boldsymbol{\gamma}$. The link utility is then weighted by path-link incidence matrix $\mathbf{D}$ to map the link utilities to the vector $\mathbf{v}_i$ of path utilities (Eq. (11)). Then, the vector $\mathbf{p}_i$ of path choice probabilities is obtained by applying Softmax functions to the utilities of each set of paths connecting every O-D pair (Eq. (12)). Subsequently, the transpose of the path-demand incidence matrix $\mathbf{M}$ is weighted by the vector $\mathbf{q}_i$ of O-D parameters to obtain a vector that matches the dimension of $\mathbf{p}_i$. The resulting vector is element-wised multiplied by $\mathbf{p}_i$ to obtain the path flow vector $\mathbf{f}_i$ (Eq. (13)). In Eq. (14), $\mathbf{D}^\top$ is multiplied with $\mathbf{f}_i$ to obtain the output link flow vector $\mathbf{x}_i$. Finally, the loss function in Eq. (15) is evaluated using the current estimates in $\hat{\mathbf{x}}_i, \mathbf{x}_i, \mathbf{t}_i, \hat{\mathbf{q}}_i$ and the observed values in $\bar{\mathbf{x}}_i, \bar{\mathbf{t}}_i, \bar{\mathbf{q}}_i$. The same process can be repeated to aggregate the losses of all samples in $\mathcal{S}$.

The elements in red in Fig. 2 synthesize our main contributions to existing literature. A key difference with respect to prior work (Wu et al., 2018; Ma et al., 2020; Liu et al., 2023) is the incorporation of link flow variables and the minimization of a loss component to satisfy the equilibrium condition. In addition, the simultaneous incorporation of attribute-specific parameters $\hat{\boldsymbol{\theta}}$ and link-specific parameters $\hat{\boldsymbol{\gamma}}$ in the utility function is able to increase representational capacity without compromising model interpretability. Furthermore, our model includes link-specific parameters $\hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\beta}}$ for the link performance functions, which are both interpretable and capture heterogeneity in the traffic flow dynamics across the network. Thus, our computational graph provides a framework to learn a set of fully interpretable parameters depicting the demand and supply characteristics of transportation networks and to find solutions that satisfy logit-based stochastic user equilibrium models (Section 3.7).

4. Solution algorithm

Our learning algorithm search for parameters $\hat{\mathbf{x}}_i, \hat{\mathbf{q}}_i, \hat{\boldsymbol{\theta}}_i, \hat{\boldsymbol{\gamma}}, \hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\beta}}$ that minimizes the loss function ℓ by solving a single-level optimization problem. The parameters of the model are learned with gradient-based optimization methods. The gradients of the parameters with respect to the loss function are obtained using a computational graph and an automatic differentiation package. With the inclusion of the equilibrium component in the loss function, our algorithm gradually learns a solution that satisfies the equilibrium condition and that reproduces the observed data. Our strategy differs from conventional approaches which alternate between a *fixed-point stage* that finds an equilibrium solution and a *learning stage* that fits the model parameters. Some examples in previous literature include the use of bilevel level optimization (Guarda and Qian, 2022), sensitivity analysis (Ma and Qian, 2018a) and deep implicit layers (Liu et al., 2023).

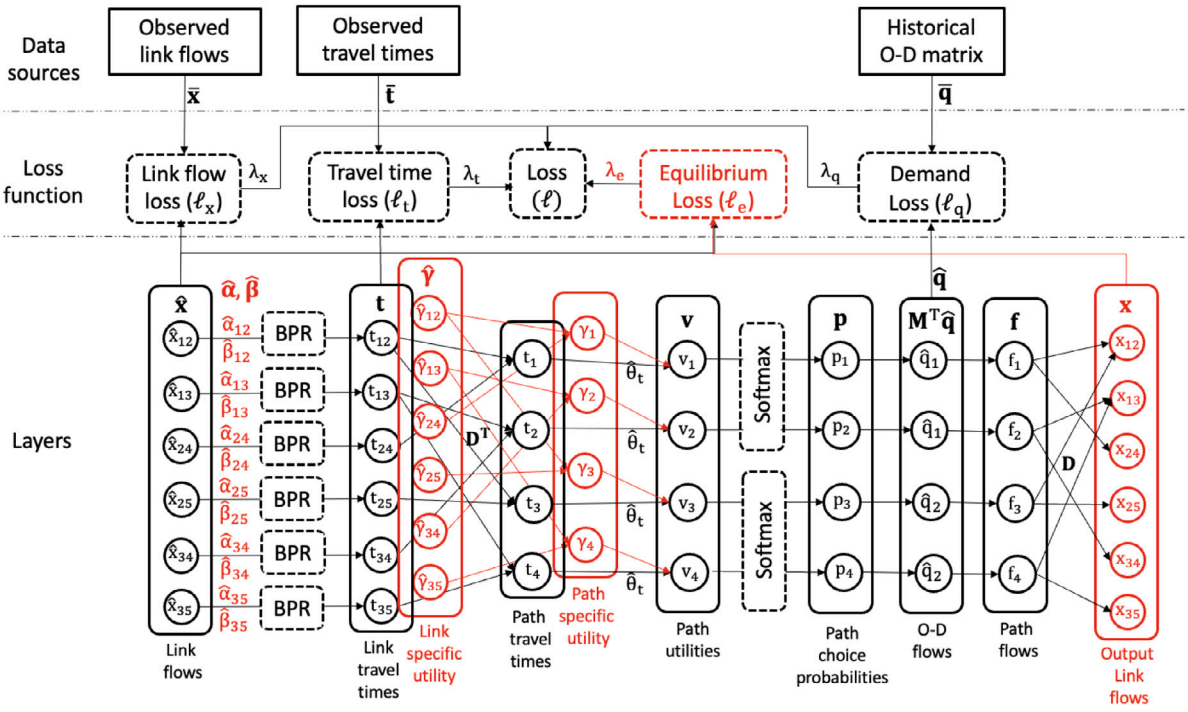


Fig. 2. An illustration of the forward–backward algorithm. The red elements illustrate the components of the model that extend previous work.

This section describes the main steps of the learning algorithm, including the forward and backward operations performed in the computational graph, the parameter updating scheme, and the check of convergence. In addition, it discusses practical issues related to hyperparameter tuning and analyzes the properties of the solutions. Finally, it presents an example that illustrates the use of the computational graph to learn the model parameters on a small transportation network.

4.1. Forward

A forward pass consists of a chain of functions or layers applied to an input that generates output values. In our problem, suppose that all learnable parameters $\hat{\mathbf{x}}_i, \hat{\mathbf{q}}_i, \hat{\theta}_i, \hat{\gamma}, \hat{\alpha}, \hat{\beta}$ (Section 3.3) are initialized to a set of feasible values, e.g., all parameters are set equal to 1. Then, let's pick a single sample $i \in S$ from the dataset. Denote $\mathcal{C}$ as the computational graph that solves problem $\mathcal{P}$ (Section 3.7). A forward pass on $\mathcal{C}$ outputs the value of the loss function in $\mathcal{P}$ by composing the equality constraints in $\mathcal{P}$ in the following order:

$$\mathbf{t}_i = \bar{\mathbf{t}}^{\min} \circ \left(1 + \hat{\boldsymbol{\alpha}} \circ \left(\frac{\hat{\mathbf{x}}_i}{\bar{\mathbf{x}}^{\max}}\right)^{\hat{\beta}}\right) \tag{10}$$

$$\mathbf{v}_i = \mathbf{D}^{\top}\left(\begin{bmatrix}\mathbf{t}_i & \mathbf{Z}_i\end{bmatrix}\hat{\theta}_i + \boldsymbol{\gamma}\right) \tag{11}$$

$$\mathbf{p}_i = \frac{\exp\left(\mathbf{v}_i\right)}{\left(\mathbf{M}^{\top}\mathbf{M}\exp(\mathbf{v}_i)\right)} \tag{12}$$

$$\mathbf{f}_i = \left(\mathbf{M}^{\top}\hat{\mathbf{q}}_i\right) \circ \mathbf{p}_i \tag{13}$$

$$\mathbf{x}_i = \mathbf{D}\mathbf{f}_i \tag{14}$$

$$\ell_i = \lambda_x \|\hat{\mathbf{x}}_i - \bar{\mathbf{x}}_i\|_2^2 + \lambda_t \|\mathbf{t}_i - \bar{\mathbf{t}}_i\|_2^2 + \lambda_q \|\hat{\mathbf{q}}_i - \bar{\mathbf{q}}_i\|_2^2 + \lambda_e \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|_2^2 \tag{15}$$

4.2. Backward

Suppose the algorithm has not converged after the forward pass. In that case, the gradients of the parameters with respect to the loss function can be backpropagated through the layers of the computational graph using the derivative chain rule:

$$
\begin{aligned}
&\frac{\partial \ell_i}{\partial \ell_x} = \lambda_x \\
&\frac{\partial \ell_x}{\partial \hat{\mathbf{x}}_i} = 2(\hat{\mathbf{x}}_i - \bar{\mathbf{x}}_i)
\end{aligned}
\qquad
\begin{aligned}
&\frac{\partial \ell_i}{\partial \ell_t} = \lambda_t \\
&\frac{\partial \ell_t}{\partial \hat{\mathbf{t}}_i} = 2(\hat{\mathbf{t}}_i - \bar{\mathbf{t}}_i) \\
&\frac{\partial \ell_t}{\partial \hat{\mathbf{x}}_i} = \left(\frac{\partial \hat{\mathbf{t}}_i}{\partial \hat{\mathbf{x}}_i}\right)^\top \frac{\partial \ell_t}{\partial \hat{\mathbf{t}}_i} \\
&\frac{\partial \ell_t}{\partial \hat{\boldsymbol{\alpha}}} = \left(\frac{\partial \hat{\mathbf{t}}_i}{\partial \hat{\boldsymbol{\alpha}}}\right)^\top \frac{\partial \ell_t}{\partial \hat{\mathbf{t}}_i} \\
&\frac{\partial \ell_t}{\partial \hat{\boldsymbol{\beta}}} = \left(\frac{\partial \hat{\mathbf{t}}_i}{\partial \hat{\boldsymbol{\beta}}}\right)^\top \frac{\partial \ell_t}{\partial \hat{\mathbf{t}}_i}
\end{aligned}
\qquad
\begin{aligned}
&\frac{\partial \ell_i}{\partial \ell_q} = \lambda_q \\
&\frac{\partial \ell_q}{\partial \hat{\mathbf{x}}_i} = 2(\hat{\mathbf{q}}_i - \bar{\mathbf{q}}_i)
\end{aligned}
\qquad
\begin{aligned}
&\frac{\partial \ell_i}{\partial \ell_e} = \lambda_e \\
&\frac{\partial \ell_e}{\partial \hat{\mathbf{x}}_i} = 2(\hat{\mathbf{x}}_i - \bar{\mathbf{x}}_i) \\
&\frac{\partial \ell_e}{\partial \hat{\mathbf{f}}_i} = \left(\frac{\partial \hat{\mathbf{x}}_i}{\partial \hat{\mathbf{f}}_i}\right)^\top \frac{\partial \ell_e}{\partial \hat{\mathbf{x}}_i} \\
&\frac{\partial \ell_e}{\partial \hat{\mathbf{q}}_i} = \left(\frac{\partial \hat{\mathbf{f}}_i}{\partial \hat{\mathbf{q}}_i}\right)^\top \frac{\partial \ell_e}{\partial \hat{\mathbf{f}}_i} \\
&\frac{\partial \ell_e}{\partial \hat{\mathbf{p}}_i} = \left(\frac{\partial \hat{\mathbf{f}}_i}{\partial \hat{\mathbf{p}}_i}\right)^\top \frac{\partial \ell_e}{\partial \hat{\mathbf{f}}_i} \\
&\frac{\partial \ell_e}{\partial \hat{\mathbf{v}}_i} = \left(\frac{\partial \hat{\mathbf{p}}_i}{\partial \hat{\mathbf{v}}_i}\right)^\top \frac{\partial \ell_e}{\partial \hat{\mathbf{p}}_i} \\
&\frac{\partial \ell_e}{\partial \hat{\boldsymbol{\theta}}_i} = \left(\frac{\partial \hat{\mathbf{v}}_i}{\partial \hat{\boldsymbol{\theta}}_i}\right)^\top \frac{\partial \ell_e}{\partial \hat{\mathbf{v}}_i} \\
&\frac{\partial \ell_e}{\partial \hat{\boldsymbol{\gamma}}} = \left(\frac{\partial \hat{\mathbf{v}}_i}{\partial \hat{\boldsymbol{\gamma}}_i}\right)^\top \frac{\partial \ell_e}{\partial \hat{\mathbf{v}}_i} \\
&\frac{\partial \ell_e}{\partial \hat{\mathbf{t}}_i} = \left(\frac{\partial \hat{\mathbf{v}}_i}{\partial \hat{\mathbf{t}}_i}\right)^\top \frac{\partial \ell_e}{\partial \hat{\mathbf{v}}_i} \\
&\frac{\partial \ell_e}{\partial \hat{\boldsymbol{\alpha}}} = \left(\frac{\partial \hat{\mathbf{t}}_i}{\partial \hat{\boldsymbol{\alpha}}}\right)^\top \frac{\partial \ell_e}{\partial \hat{\mathbf{t}}_i} \\
&\frac{\partial \ell_e}{\partial \hat{\boldsymbol{\beta}}} = \left(\frac{\partial \hat{\mathbf{t}}_i}{\partial \hat{\boldsymbol{\beta}}}\right)^\top \frac{\partial \ell_e}{\partial \hat{\mathbf{t}}_i} \\
&\frac{\partial \ell_e}{\partial \hat{\mathbf{x}}_i} = \left(\frac{\partial \hat{\mathbf{t}}_i}{\partial \hat{\mathbf{x}}_i}\right)^\top \frac{\partial \ell_e}{\partial \hat{\mathbf{t}}_i}
\end{aligned}
\tag{16}
$$

where

$$
\begin{aligned}
&\frac{\partial \hat{\mathbf{t}}_i}{\partial \hat{\mathbf{x}}_i} = \mathbf{1}_{|\mathcal{A}|\times|\mathcal{A}|}\ \bar{\mathbf{t}}^{\min} \circ \left(1 + \hat{\boldsymbol{\alpha}} \circ \hat{\boldsymbol{\beta}} \circ \left(\frac{\hat{\mathbf{x}}_i}{\bar{\mathbf{x}}^{\max}}\right)^{(\hat{\boldsymbol{\beta}}-1)}\right) \\
&\frac{\partial \hat{\mathbf{t}}_i}{\partial \hat{\boldsymbol{\alpha}}} = \mathbf{1}_{|\mathcal{A}|\times|\mathcal{A}|}\ \bar{\mathbf{t}}^{\min} \circ \left(1 + \left(\frac{\hat{\mathbf{x}}_i}{\bar{\mathbf{x}}_{\max}}\right)^{\hat{\boldsymbol{\beta}}}\right) \\
&\frac{\partial \hat{\mathbf{t}}_i}{\partial \hat{\boldsymbol{\beta}}} = \mathbf{1}_{|\mathcal{A}|\times|\mathcal{A}|}\ \bar{\mathbf{t}}^{\min} \circ \left(1 + \hat{\boldsymbol{\alpha}} \circ \left(\frac{\hat{\mathbf{x}}_i}{\bar{\mathbf{x}}_{\max}}\right)^{\hat{\boldsymbol{\beta}}}\right) \log\left(\frac{\hat{\mathbf{x}}_i}{\bar{\mathbf{x}}_{\max}}\right) \\
&\frac{\partial \hat{\mathbf{v}}_i}{\partial \hat{\mathbf{t}}_i} = \theta_i \mathbf{D}^\top \\
&\frac{\partial \hat{\mathbf{v}}_i}{\partial \hat{\boldsymbol{\theta}}_i} = \mathbf{D}^\top \left(\begin{bmatrix}\mathbf{t}_i & \mathbf{Z}_i\end{bmatrix}\right) \\
&\frac{\partial \hat{\mathbf{v}}_i}{\partial \hat{\boldsymbol{\gamma}}} = \mathbf{D}^\top \\
&\frac{\partial \hat{\mathbf{p}}_i}{\partial \hat{\mathbf{v}}_i} = \mathbf{M}^\top \mathbf{M} \circ \mathbf{p}_i \mathbf{p}_i^\top + \mathbf{1}_{|\mathcal{H}|\times|\mathcal{H}|}\left(\mathbf{p}_i \circ \left(1 - 2\mathbf{p}_i\right)\right) \\
&\frac{\partial \hat{\mathbf{f}}_i}{\partial \hat{\mathbf{q}}_i} = \mathbf{1}_{|\mathcal{H}|\times|\mathcal{H}|}\ \mathbf{p}_i\ \mathbf{M}^\top \\
&\frac{\partial \hat{\mathbf{f}}_i}{\partial \hat{\mathbf{p}}_i} = \mathbf{1}_{|\mathcal{H}|\times|\mathcal{H}|} \mathbf{M}^\top \hat{\mathbf{q}}_i \\
&\frac{\partial \hat{\mathbf{x}}_i}{\partial \hat{\mathbf{f}}_i} = \mathbf{D}
\end{aligned}
\tag{17}
$$

and division, multiplication ($\circ$) and exponentiation are applied element-wise, $\mathbf{1}_{|\mathcal{A}|\times|\mathcal{A}|}$ and $\mathbf{1}_{|\mathcal{H}|\times|\mathcal{H}|}$ are identity matrices with dimensions $|\mathcal{A}|\times|\mathcal{A}|$ and $|\mathcal{H}|\times|\mathcal{H}|$, respectively, and $\mathbf{1}_{|\mathcal{A}|}$ is a column vector with ones and dimension $|\mathcal{A}|$.

The gradients of the loss function with respect to the parameters of interest are given in Eqs. (18). They can be efficiently computed by substituting Eqs. (16) and (17). From the RHS of the equations, we note that the choice of hyperparameters affects the magnitude of the gradients. However, in practice very few hyperparameter tuning for model training is required.

$$\begin{aligned}
\frac{\partial \ell_i}{\partial \hat{\boldsymbol{\alpha}}} &= \frac{\partial \ell_i}{\partial \ell_t}\frac{\partial \ell_t}{\partial \hat{\boldsymbol{\alpha}}} + \frac{\partial \ell_i}{\partial \ell_e}\frac{\partial \ell_e}{\partial \hat{\boldsymbol{\alpha}}} = \lambda_t \frac{\partial \ell_t}{\partial \hat{\boldsymbol{\alpha}}} + \lambda_e \frac{\partial \ell_e}{\partial \hat{\boldsymbol{\alpha}}} \\
\frac{\partial \ell_i}{\partial \hat{\boldsymbol{\beta}}} &= \frac{\partial \ell_i}{\partial \ell_t}\frac{\partial \ell_t}{\partial \hat{\boldsymbol{\beta}}} + \frac{\partial \ell_i}{\partial \ell_e}\frac{\partial \ell_e}{\partial \hat{\boldsymbol{\beta}}} = \lambda_t \frac{\partial \ell_t}{\partial \hat{\boldsymbol{\beta}}} + \lambda_e \frac{\partial \ell_e}{\partial \hat{\boldsymbol{\beta}}} \\
\frac{\partial \ell_i}{\partial \hat{\mathbf{q}}_i} &= \frac{\partial \ell_i}{\partial \ell_q}\frac{\partial \ell_q}{\partial \hat{\mathbf{q}}_i} + \frac{\partial \ell_i}{\partial \ell_e}\frac{\partial \ell_e}{\partial \hat{\mathbf{q}}_i} = \lambda_q \frac{\partial \ell_q}{\partial \hat{\mathbf{q}}_i} + \lambda_e \frac{\partial \ell_e}{\partial \hat{\mathbf{q}}_i} \\
\frac{\partial \ell_i}{\partial \hat{\boldsymbol{\theta}}_i} &= \frac{\partial \ell_i}{\partial \ell_e}\frac{\partial \ell_e}{\partial \hat{\boldsymbol{\theta}}_i} = \lambda_e \frac{\partial \ell_e}{\partial \hat{\boldsymbol{\theta}}_i} \\
\frac{\partial \ell_i}{\partial \hat{\mathbf{x}}_i} &= \frac{\partial \ell_i}{\partial \ell_x}\frac{\partial \ell_x}{\partial \hat{\mathbf{x}}_i} + \frac{\partial \ell_i}{\partial \ell_e}\frac{\partial \ell_e}{\partial \hat{\mathbf{x}}_i} + \frac{\partial \ell_i}{\partial \ell_t}\frac{\partial \ell_t}{\partial \hat{\mathbf{x}}_i} = \lambda_x \frac{\partial \ell_x}{\partial \hat{\mathbf{x}}_i} + \lambda_e \frac{\partial \ell_e}{\partial \hat{\mathbf{x}}_i} + \lambda_t \frac{\partial \ell_t}{\partial \hat{\mathbf{x}}_i}
\end{aligned} \tag{18}$$

4.3. Convergence check

Our algorithm should converge when it finds a solution that approximately satisfies the SUELOGIT equilibrium conditions, that is, a fixed point in link flow space. A standard strategy to check convergence in equilibrium algorithms is to compare network flows between successive iterations. Because every forward pass in C provides input link flows ($\hat{\mathbf{x}}$) and the output link flows ($\mathbf{x}$), a natural choice is to compare the relative difference between these quantities. Formally, the relative gap in epoch j is defined as:

$$\rho_j = \frac{\left\|\hat{\mathbf{x}}^{(j)} - \mathbf{x}^{(j)}\right\|_1}{\left\|\mathbf{x}^{(j)}\right\|_1} \tag{19}$$

Because the estimated and output link flows vectors will not be exactly equal within a finite number of epochs, the algorithm is assumed to converge at epoch $j > 1$ if $\rho_j < \rho_\star$ where $\rho_\star$ is a critical threshold set for the modeler beforehand, e.g., 10^{-5}. This condition is not checked in epoch 0 because otherwise, the algorithm would converge immediately.

4.4. Parameter updating

The primary input of a first-order optimization method is the gradient of the loss function with respect to the model parameters. To compute the gradient at every epoch j is necessary to add the contribution to the gradient of each sample in the batch. The parameter updates are a function of the magnitudes of the gradients and can be projected to output parameters that are within the feasible space. For the standard project gradient descent method, the parameter updates are given by:

$$\begin{aligned}
\hat{\boldsymbol{\alpha}}^{(j+1)} &= \mathbb{P}_{\hat{\boldsymbol{\alpha}}>0}\left(\hat{\boldsymbol{\alpha}}^{(j)} + \eta \sum_{i\in S} \frac{\partial \ell_i^{(j)}}{\partial \hat{\boldsymbol{\alpha}}}\right) \\
\hat{\boldsymbol{\beta}}^{(j+1)} &= \mathbb{P}_{\hat{\boldsymbol{\beta}}>0}\left(\hat{\boldsymbol{\beta}}^{(j)} + \eta \sum_{i\in S} \frac{\partial \ell_i^{(j)}}{\partial \hat{\boldsymbol{\beta}}}\right) \\
\hat{\boldsymbol{\theta}}^{(j+1)} &= \mathbb{P}_{\theta_t<0}\left(\hat{\boldsymbol{\theta}}^{(j)} + \eta \sum_{i\in S} \frac{\partial \ell_i^{(j)}}{\partial \hat{\boldsymbol{\theta}}_i}\right) \\
\hat{\mathbf{q}}^{(j+1)} &= \mathbb{P}_{\hat{\mathbf{q}}\geq 0}\left(\hat{\mathbf{q}}^{(j)} + \eta \sum_{i\in S} \frac{\partial \ell_i^{(j)}}{\partial \hat{\mathbf{q}}_i}\right) \\
\hat{\mathbf{x}}^{(j+1)} &= \mathbb{P}_{\hat{\mathbf{x}}\geq 0}\left(\hat{\mathbf{x}}^{(j)} + \eta \sum_{i\in S} \frac{\partial \ell_i^{(j)}}{\partial \hat{\mathbf{x}}_i}\right)
\end{aligned} \tag{20}$$

where $\eta > 0$ is the learning rate and S is the set of available samples. The learning rate is a hyperparameter that moderates the speed of the updates. Because the problem is a minimization, the parameter updates are in the negative direction of the gradient. Note that the projector operator $\mathbb{P}_{\theta_t<0}$ for $\hat{\boldsymbol{\theta}}$ may include additional sign constraints for the parameter depending on the prior knowledge of the modeler, e.g., a parameter weighting the price of a toll could be assumed negative. Using a refined scheme for parameter updating as the one implemented in the Adam optimizer (Kingma and Ba, 2015) tends to improve the performance of our algorithm.

4.5. Initialization

The initialization of the model's parameters can significantly impact the convergence to a solution. A convenient strategy is to leverage prior domain knowledge. The link performance parameters are initialized with the standard values of the BPR function ($\alpha = 0.15, \beta = 4$). The utility function parameters can be initialized to zero when no prior values are available. The parameters $\mathbf{q}$ can

be initialized with the historical O-D matrix if available. To initialize the input link flows $\hat{\mathbf{x}}$, we use the output link flows of a single pass of logit-based stochastic traffic assignment, assuming that travel times are equal to the free flow travel times. This heuristic allows initializing the full set of link flow parameters and equates to a single forward pass (Section 4.1) in the computational graph, where the input link flows are equal to zero. The values of the remaining parameters of the model do not need to be explicitly initialized because they are a function of the set of learnable parameters (Section 3.3).

4.6. Framework

Our solution algorithm is summarized in Algorithm 1.

Algorithm 1 PESUELOGIT: Parameter Estimation of Stochastic User Equilibrium under LOGIT assignment models

Step 0: Initialization.
a: Initial epoch: $j = 0$.
b: Initialize parameters for every sample $i \in S$: $\hat{\mathbf{x}}_i, \hat{\mathbf{q}}_i, \hat{\theta}_i, \hat{\boldsymbol{\gamma}}_i, \hat{\boldsymbol{\alpha}}_i, \hat{\boldsymbol{\beta}}_i \leftarrow \tilde{\mathbf{x}}_i, \tilde{\mathbf{q}}_i, \tilde{\theta}_i, \tilde{\boldsymbol{\gamma}}, \tilde{\boldsymbol{\alpha}}, \tilde{\boldsymbol{\beta}}$

Learning stage

Step 1: Forward Iteration.
a: Compute $\mathbf{t}_i, \mathbf{v}_i, \mathbf{p}_i, \mathbf{f}_i, \mathbf{x}_i$ for each sample $i \in S$ using Eqs. (10)-(14), Section 4.1.
b: Compute the loss function ℓ_j in epoch j with Eq. (15), Section 4.1.

Step 2: Convergence check
a: Compute relative gap ρ_j using current values of $\mathbf{x}_i$ and $\hat{\mathbf{x}}_i, \forall i \in S$ (Section 4.3)
b: Go to Step 3 if $\rho_j, \forall j \geq 1$[a] is higher than the critical threshold $\rho_\star$. Otherwise, terminate the algorithm.

Step 3: Backward iteration. Compute gradient of the loss function ℓ with respect to the parameters $\hat{\mathbf{x}}_i, \hat{\mathbf{q}}_i, \hat{\theta}_i, \hat{\boldsymbol{\gamma}}, \hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\beta}}$ (Section 4.2)

Step 4: Parameters updating. Parameters $\hat{\mathbf{x}}_i, \hat{\mathbf{q}}_i, \hat{\theta}_i, \hat{\boldsymbol{\gamma}}, \hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\beta}}$ are updated for every sample $i \in S$ using a gradient-based projection method (Section 4.4).

Step 5: Termination criterion. If the number of epochs is lower than the maximum epochs for the *learning stage*, set $j \leftarrow j + 1$ and go back to Step 1. If not, continue to the *equilibrating stage*.

Equilibrating stage

Step 6: Forward Iteration.
a: Compute $\mathbf{t}_i, \mathbf{v}_i, \mathbf{p}_i, \mathbf{f}_i, \mathbf{x}_i$ for each sample $i \in S$ using Eqs. (10)-(14), Section 4.1.
b: Compute the loss function $\ell_j = \sum_{i \in S} \|\hat{\mathbf{x}}_i - \mathbf{x}_i\|_2^2$ in epoch j with Eq. (15), Section 4.1.

Step 7: Convergence check
a: Compute relative gap ρ_j using current values of $\mathbf{x}_i$ and $\hat{\mathbf{x}}_i, \forall i \in S$ (Section 4.3)
b: Go to Step 8 if ρ_j is higher than the critical threshold $\rho_\star$. Otherwise, terminate the algorithm.

Step 8: Backward iteration. Compute gradient of the loss function ℓ with respect to $\hat{\mathbf{x}}$ (Section 4.2)

Step 9: Link flow updating. Update input link flow $\hat{\mathbf{x}}_i, \forall i \in S$ using a gradient-based projection method (Section 4.4).

Step 10: Termination criterion. If the number of epochs is lower than the maximum epochs for the *equilibrating stage*, set $j \leftarrow j + 1$ and go back to Step 6. Otherwise, terminate the algorithm.

[a] With the initialization strategy discussed in Section 4.5, $\mathbf{x} = \hat{\mathbf{x}}$ at epoch 0, which, by construction, results into $\rho = 0 < \rho_\star$. Thus, the convergence check in Step 2b is evaluated from $j \geq 1$.

4.7. Other design aspects

The optimization is divided into two stages. In the *learning stage*, all parameters of the model are estimated. In the *equilibrating stage*, only the link flow variables are estimated and all hyperparameters except for λ_e are set to 1. The *learning stage* ends when the relative gap between values of the SUELOGIT objective function of two consecutive epochs becomes lower than a certain threshold or when the maximum number of epochs for this stage is reached. If the second condition is met, the *equilibrating stage* starts, and it continues iterating until the threshold for the relative gap is reached or when the maximum number of epochs for this stage is achieved. This strategy encourages finding a solution that satisfies the equilibrium condition at the end of the optimization. Finally, to avoid exploding gradients, the hyperparameters weighting the components of the loss function are normalized at every epoch such that their sum always equals one.

4.8. Properties

The following propositions identify some properties of the solutions obtained by our learning algorithm. The remarks of the propositions provide some recommendations to set some hyperparameters of the model and to analyze the final solutions.

Proposition 1 (*Equivalence Between Computational Graph and Optimization Problem*)**.** *The problem solved by the computational graph $\mathcal{C}$ is $\mathcal{P}$*

Proof. Suppose forward and backward passes are performed sequentially for every sample in $\mathcal{S}$. Without any loss of generality, consider the forward pass of a sample $j \in \mathcal{S}$. A forward in the computational graph must satisfy Eqs. (10)–(14), and thus, the outputs of the layers in $\mathcal{C}$ satisfy the five equality constraints in $\mathcal{P}$. In the backward pass, the parameters updates of $\hat{\mathbf{x}}_j, \hat{\mathbf{q}}_j, \hat{\boldsymbol{\theta}}_j, \hat{\boldsymbol{\gamma}}, \hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\beta}}$ are projected to their feasible space, which guarantees that inequality constraints in $\mathcal{P}$ are also satisfied. Then, the values of the parameters in $\mathcal{C}$ obtained after performing a backward pass in any sample are a feasible solution in $\mathcal{P}$. Lastly, the value of the loss function obtained for each sample associated with $\mathcal{C}$ can be summed (Eq. (15)), $\sum_{i\in\mathcal{S}} \ell_i = \lambda_{\bar{x}}\|\hat{\mathbf{x}} - \bar{\mathbf{x}}\|_F^2 + \lambda_{\bar{t}}\|\hat{\mathbf{t}} - \bar{\mathbf{t}}\|_F^2 + \lambda_{\bar{q}}\|\hat{\mathbf{q}} - \bar{\mathbf{q}}\|_F^2 + \lambda_e\|\mathbf{x} - \hat{\mathbf{x}}\|_F^2$. Given that the loss functions of $\mathcal{C}$ and $\mathcal{P}$ are identical, a solution of $\mathcal{C}$ is also a solution in $\mathcal{P}$, and vice-versa, which proves that the problem solved by $\mathcal{C}$ is $\mathcal{P}$. □

Remark 3. The forward and backward passes can be performed sequentially with the full set of samples, batches of samples, or single samples, i.e., a batch of size 1. The choice of batch size may affect the algorithm's convergence speed but not the proposition result.

Definition 1. At Stochastic User Equilibrium with logit assignment (SUELOGIT), path flows are distributed among the paths connecting each O-D pair according to a logit distribution, where path costs are determined as functions of the assigned path flows (Fisk, 1980).

Proposition 2 (*Equilibrium Solution*)**.** *If the computational graph $\mathcal{C}$ achieves a fixed point, the network flow is at Stochastic User Equilibrium with logit assignment (*SUELOGIT*).*

Proof. Without loss of generality, consider a single sample $i \in \mathcal{S}$. Because a fixed point in $\mathcal{C}$ has been achieved, $\hat{\mathbf{x}}_i = \mathbf{x}_i = \mathbf{x}_i^\star$, where $\mathbf{x}_i^\star$ satisfy (9). From (14), $\mathbf{x}_i = \mathbf{D}\mathbf{f}_i^\star$, where $\mathbf{f}_i^\star$ is the path flow vector at the fixed-point. Suppose the link flow $\hat{\mathbf{x}}_i$ is induced by a path flow vector $\hat{\mathbf{f}}_i$, such that $\hat{\mathbf{x}}_i = \mathbf{D}\hat{\mathbf{f}}_i$. Given that problem $\mathcal{P}$ imposes no constraint on $\hat{\mathbf{f}}_i$ and that $\hat{\mathbf{x}}_i = \mathbf{x}_i$, we can set $\hat{\mathbf{f}}_i = \mathbf{f}_i^\star$. The solution $\mathbf{f}_i = \hat{\mathbf{f}}_i = \mathbf{f}_i^\star$ complies with Definition 1, which proves that the network flow is at SUELOGIT. □

Remark 4. To obtain solutions that satisfy SUELOGIT, the equilibrium loss component must approach zero during model training. For this purpose, it can help to finetune the value of the hyperparameter λ_e that weights the equilibrium component. Similar to classical network equilibrium algorithms, the termination criterion for model training is based on the relative gap between the input link flows $\hat{\mathbf{x}}$ and the output link flows $\mathbf{x}$. It is essential to check that this criterion effectively leads to lower values of the equilibrium component because the latter is the quantity that informs how close the solution is to the fixed point. If the relative gap condition is not satisfied during the *learning stage*, it follows the *equilibrating stage*. The *equilibrating stage* directly solves SUELOGIT, and thus, it encourages that the final solution of Algorithm 1 satisfies the equilibrium condition for a given relative gap threshold.

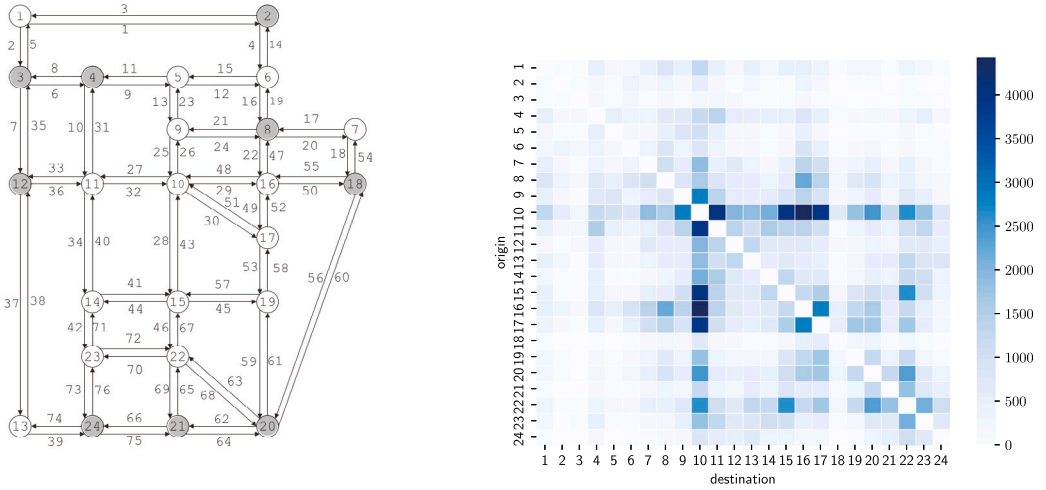
Proposition 3 (*Non-Convexity*)**.** *Problem $\mathcal{P}$ is non-convex.*

Proof. Suppose there is only a single sample of observations available. Let us initialize all parameters of $\mathcal{P}$ to values that satisfy the constraints in (10)–(14) and set $\lambda_q = 0, \lambda_t = 0, \lambda_e = 0$. Let us freeze all parameters except the attribute-specific parameters $\boldsymbol{\theta}$ of the utility function. This problem is equivalent to the optimization program solved by Guarda and Qian (2022), which was proved to be non-convex. Therefore, $\mathcal{P}$ is non-convex. □

Remark 5. Due to the non-convexity of $\mathcal{P}$, if the gradients of the parameters with respect to the loss function vanish at a point, we can only guarantee that solution is a stationary point. To verify if that solution is a local minimum, it is advisable to check that the Hessian evaluated at that point is positive semi-definite. Furthermore, the termination criterion of the algorithm is based on the relative gap between the input and output link flows (Section 4.3). Thus, the final solution is not guaranteed to generate gradients close to zero with respect to every learnable parameter.

5. Numerical experiments

To study the performance of our algorithm to recover the model parameters, we conduct experiments with synthetic data generated from the Sioux Falls network. The Sioux Falls network is a standard testing bed used by transportation researchers in network modeling studies. The network comprises 24 nodes and 76 links (Fig. 3(a)). The O-D matrix is obtained from TNTP (2021), and a heatmap with its values is presented in Fig. 3(b). We generate a set of 1584 paths corresponding to the three shortest paths between each of the 528 O-D pairs.



(a) Network topology (b) O-D matrix

Fig. 3. Topology and O-D matrix of Sioux Falls network.

For the experiments, we employ a validation framework where the ground truth values of the parameters for the number of trips between O-D pairs ($\tilde{\mathbf{q}}$), the utility function ($\tilde{\theta}, \tilde{\boldsymbol{\gamma}}$) and the link performance functions ($\tilde{\alpha}, \tilde{\beta}$) are assumed known and used to generate synthetic traffic flow and travel times measurements consistent with logit-based stochastic user equilibrium (Section 5.1). All the experiments in this section are conducted on an Apple M2 Pro with a 10-core CPU, 16 GB of unified memory, and 1 TB SSD. The runtime of the experiments presented in this section is approximately 2 h, with every epoch taking between 0.1–0.2 [s]. All models are trained in TensorFlow (Abadi et al., 2016) and using the Adam optimizer (Kingma and Ba, 2015).

5.1. Data generating process

The data generating process (DGP) in the numerical experiments assumes that the modeler has perfect information about the path sets, link performance functions, the network topology, the ground truth parameters of the model, and the exogenous attributes of the utility function. The link cost performance functions are assumed equal among links and of BPR class with parameters $\tilde{\alpha} = 0.15$ and $\tilde{\beta} = 4$. The utility function is assumed linear in parameters and dependent on travel time [minutes] and on two exogenous attributes of each link; the standard deviation of travel time υ [minutes] and the density of streets intersections s [intersections/mile]. The attribute values are generated using uniform random variables bounded between 0 and 1. It is assumed that the parameters are common among travelers and are linearly weighting the attributes of the utility function. The parameters in the utility function weighting the attributes t, υ and s are set to $\tilde{\theta}_t = -1$, $\tilde{\theta}_\upsilon = -1.3$, $\tilde{\theta}_s = -3$, such that the reliability ratio (RR) becomes equal to 1.3. In line with Carrion and Levinson (2012), the RR is defined as the ratio of the marginal utility associated with the travel time and the standard deviation of travel time, namely, $\mathrm{RR} = \tilde{\theta}_\upsilon / \tilde{\theta}_t$. The RR can be interpreted as the travelers' valuation of travel time variability relative to the expected travel time.

To generate synthetic data, we solve logit-based stochastic user equilibrium (SUELOGIT) with the optimization formulation presented by Guarda and Qian (2022) for multi-attribute utility functions. The optimization problem is then solved by the Frank-Wolfe algorithm to obtain traffic flow and travel time observations at every link. To introduce randomness in the DGP, we add a Gaussian error term with zero mean and standard deviation equal to 10% of the true mean of the observed measurements. The process is repeated 100 times to represent a scenario where traffic flow and travel time measurements under recurrent traffic conditions are available for 100 days and at the same time of the day and the day of the week. Finally, we assume that only 75% of the links have sensors that can record travel times and traffic flows. The vector of predicted travel times and link flows are masked to account for the 25% of links in the network that do not report observed measurements.

5.2. Models specifications

Table 3 lists the parameters of the models trained in the numerical experiments. The SUELOGIT model performs logit-based stochastic user equilibrium and only has the input link flows ($\hat{\mathbf{x}}$) as learnable parameters. The input link flows are included as learnable parameters in all models, given that they are key to satisfying the equilibrium condition. The O-D estimation (ODE) model has one parameter for each origin–destination pair that carries flow. Besides the parameters included in ODE, the ODLPE model also estimates the parameters α, β of the link performance functions. Both the LUE and ODLUE models estimate parameters for $\theta_t, \theta_\upsilon, \theta_s$, but the latter also estimates the O-D matrix. Finally, the ODLULPE model estimates the link performance parameters on top of the parameters of the ODLUE model. Given that the numerical experiments assume perfect knowledge about the specification of the utility function and the models are trained with data from a single period, no link-specific parameters $\boldsymbol{\gamma}$ are estimated (Section 3.3).

Table 3
Models and parameters estimated using synthetic data from Sioux Falls network.

Model	Number of parameters				
	Link flows	Utility function	O-D matrix	Link performance function	Total
SUELOGIT	76	–	–	–	–
ODE	76	–	528	–	604
ODLPE	76	–	528	2	606
LUE	76	3	–	–	79
ODLUE	76	3	528	–	609
ODLULPE	76	3	528	2	611

Table 4
Hyperparameters of the components of the loss function used for training the models with synthetic data from Sioux Falls network.

Model	Loss component			
	Equilibrium	Traffic flows	Travel times	O-D matrix
SUELOGIT	1	0	0	0
ODE	1	1	1	1
ODLPE	1	1	1	1
LUE	1	1	1	0
ODLUE	1	1	1	1
ODLULPE	1	1	1	1

Note: During the optimization, the hyperparameters are normalized such that their sum is 1.

5.3. Parameters initialization

The starting points for optimization of all the learnable parameters, except for the input link flow, are set to their ground truth values. This initialization strategy accelerates models' convergence during the numerical experiments and ensures that the initial loss used to compute the relative MSEs in all model specifications (Table 3) is the same. The main reason to choose this initialization strategy is to facilitate model comparisons, not a recommendation to ensure convergence to the ground truth values. The initial values for the input link flows are obtained with the output of a single pass of logit-based stochastic traffic assignment (Section 4.5). The travel times used to initialize the input link flows in the forward are assumed equal to the free flow travel times.

5.4. Hyperparameters

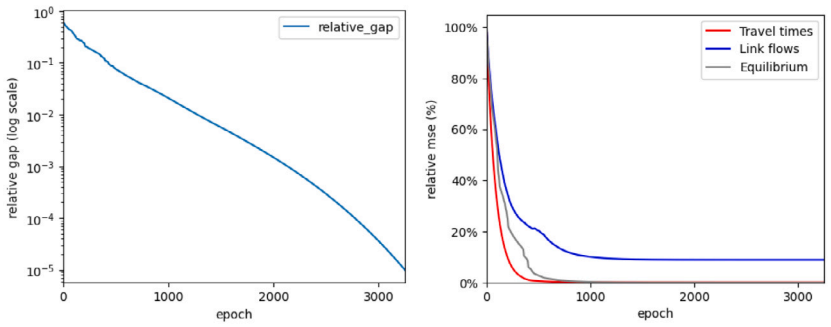
The learning rate is set at 0.1 and a full batch of observations is used at every epoch. The maximum number of epochs for the learning and equilibrating stages (Section 4.7) are 5000 and 100, respectively. Similar to Liu et al. (2023), the threshold for the relative gap is set to 10^{-5} (Section 4.3). Table 4 shows the values of the hyperparameters weighting the loss components (Section 3.6). Because the SUELOGIT model only seeks to find link flows that result into a fixed point, i.e., where the equilibrium component approaches zero, the values of all hyperparameters weighting the loss components except for the equilibrium component are set to zero. Note that the hyperparameter weighting the O-D loss component in the SUELOGIT and the LUE model is set to zero. However, this can be set to an arbitrary value because the O-D parameters in both models are fixed to their true values. For the remaining models, the hyperparameters of all components have values different than zero (Table 3) because these models can leverage measurements of traffic flows and travel times and information of a prior historic O-D matrix to learn parameters.

5.5. Results

This section analyses the parameter recovery and convergence of model training during the numerical experiments. To study convergence, we compute the change of the relative value of the loss components over epochs, namely, the percentage difference of the value of a loss component at a given epoch with respect to epoch 0. For the equilibrium component, the relative MSEs are computed with respect to the loss reported in epoch 1 because the equilibrium component is by construction equal to zero in epoch 0 (Section 4.5). The relative MSE associated with the O-D matrix is not computed because this curve is not expected to decrease monotonically over epochs. Instead, the quality of O-D estimation is studied by comparing heatmaps of the estimated, historical, and true O-D matrices.

5.5.1. Logit-based stochastic user equilibrium (SUELOGIT)

The SUELOGIT problem has been proven to be strictly convex when utility functions are linear in parameters, and link performance functions are monotonic (see Theorem 2, Cantarella (1997)). Thus, estimating the SUELOGIT model should find link flows that globally minimize the loss function of $\mathcal{P}$. Fig. 4(a) shows the change in the relative MSE of the travel times, link flows, and equilibrium components of the loss function over epochs. We observe that the relative MSE associated with the equilibrium component (gray line) decreases monotonically over epochs and that the relative gap becomes lower than the threshold set at 10^{-5}



(a) Change in relative gap (b) Change in relative MSE

Fig. 4. Estimation of logit-based stochastic user equilibrium (SUELOGIT) in the Sioux Falls network.

before 4000 epochs (Fig. 4(b)). This result suggests that our algorithm can find a link flow solution that approximately satisfies the equilibrium conditions. Note also that the loss components associated with the travel times and link flows steadily decrease over epochs, despite that the hyperparameters λ_t, λ_x are set to zero in the loss function. This behavior is expected given that the average values of the observed measurements in the synthetic data satisfy the equilibrium condition and that all parameters except for the input link flows were fixed at their ground truth values (Section 5.3).

Remark 6. The portions of the loss component associated with link flows and travel times that are not minimized correspond to the random component introduced in the observed measurements during the data-generating process (Section 5.1). It can be interpreted as the irreducible day-to-day variance of link flow and travel time. While the prediction error could be minimized entirely if the O-D matrices are allowed to differ for each observation, this model will try to fit the irreducible variance of the data-generating process, and thus, it will overfit the training data.

5.5.2. Origin–destination matrix (ODE)

The O-D estimation (ODE) problem is typically formulated as a bilevel optimization program that seeks to find link flows that satisfy the equilibrium condition and match a set of observed link flow measurements. Due to its bilevel nature, ODE is a non-convex optimization problem with a non-unique global optimum. To narrow the searching domain for the estimated O-D matrix $\tilde{\mathbf{Q}}$, the upper-level objective function can include a term that accounts for the deviation between $\tilde{\mathbf{Q}}$ and a historical O-D matrix $\bar{\mathbf{Q}}$ (Ma and Qian, 2018b). In our formulation, this term corresponds to the component ℓ_q in the loss function (Fig. 2). To represent a more realistic setting of ODE, we add a Gaussian error to each cell of the historic O-D matrix with a standard deviation equal to 10% of the average number of trips per cell in the true O-D matrix.

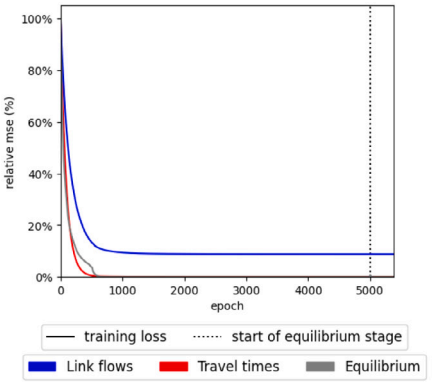
Fig. 5(a) shows the change in the relative MSE over epochs of the ODE model. The optimization converges after 5383 epochs, meaning that the critical threshold for the relative gap is achieved with 383 additional iterations in the *equilibrating stage*. The number of additional iterations is small relative to the number of iterations that are necessary to achieve equilibrium with the SUELOGIT model (Fig. 4(a)). This suggests that the *learning stage* can find solutions that not only reproduce the observed measurements but that also approximately satisfy equilibrium. Possibly due to the incorporation of historical O-D information, we also observe that the estimated and true O-D matrix are close (Fig. 5(b)). Furthermore, the relative MSEs of travel times and link flows reach values similar to those observed in the estimation of the SUELOGIT (Fig. 4(b)). Because the estimated and true O-D matrices are similar, the input link flows in the ODE and SUELOGIT models are close and thus, both models exhibit similar relative MSE curves.

5.5.3. Link performance parameters (LPE)

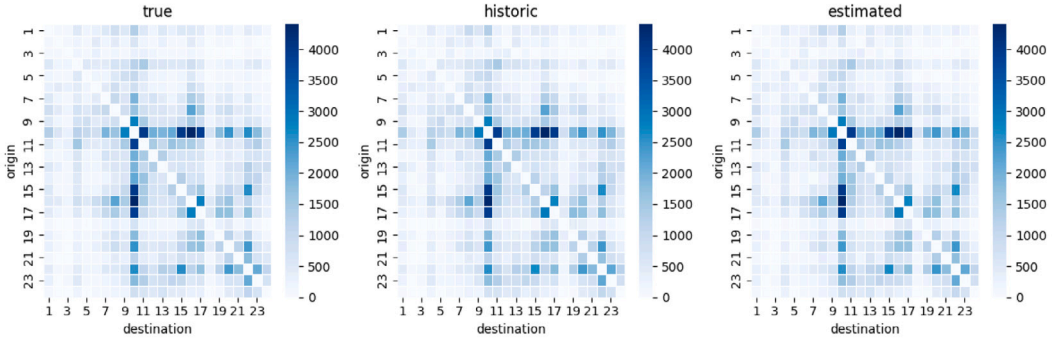
The model for link performance estimation (LPE) includes parameters that are linear ($\hat{\alpha}$) and non-linear ($\hat{\beta}$) with respect to link flows. Our results show that both parameters can be recovered accurately, $\hat{\alpha} = 0.14, \hat{\beta} = 4.06$. Possibly due to the nonlinearity of $\hat{\beta}$, the value of this parameter has a more considerable fluctuation over epochs (Fig. 6(b)). Similar to the ODE model, the critical threshold for the relative gap is reached during the *equilibrating stage* (Fig. 6(a)). Compared to the LPE model, the ODLULPE model recovers the value of the link performance parameters with less accuracy ($\hat{\alpha} = 0.18,\ \hat{\beta} = 3.77$). This suggests that the joint estimation of $\hat{\alpha}$, $\hat{\beta}$ and the travel time coefficient $\hat{\theta}_t$ of the utility function could make it harder to identify the true values of the link performance parameters.

5.5.4. Utility function parameters

The utility function parameters are estimated in three model specifications: (a) estimation of logit utility (LUE) parameters only, (b) joint estimation of O-D matrix (ODLUE) and (c) joint estimation of O-D matrix and link performance parameters (ODLULPE). To compute the relative MSE curves with the same initial loss, the ODLUE and ODLULPE models are trained with the same historic O-D matrix generated to train the ODE model (Section 5.5.2). Fig. 7 suggests that the relative MSE curves over epochs converge to similar values in all models. In all cases, the solutions reach the critical threshold for the relative gap during the *equilibrating stage*. The estimated reliability ratios (RR) of the LUE, ODLUE, and ODLULPE models are 1.63, 1.17 and 2.00, respectively, and they are close to the ground truth RR of 1.3 (Fig. 8) used for the numerical experiments (see Fig. 8).

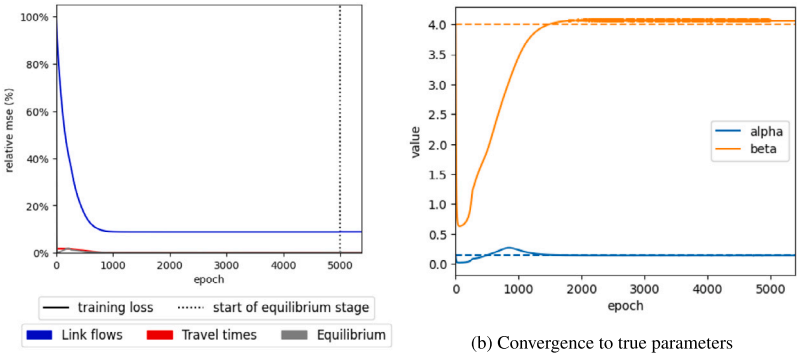


(a) Change in relative MSE



(b) Heatmaps of historical and estimated origin-destination matrices

Fig. 5. Estimation of Origin-Destination matrix (ODE) in the Sioux Falls network.



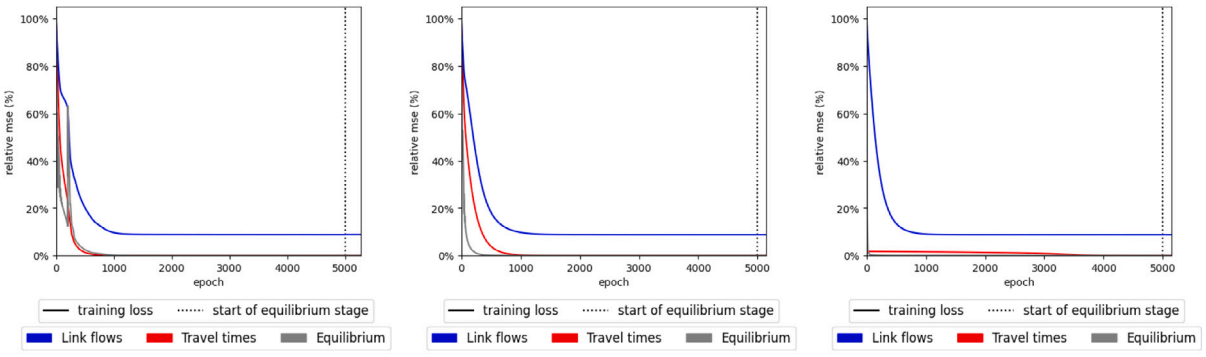
(a) Change in relative MSE

(b) Convergence to true parameters

Fig. 6. Estimation of link performance parameters (LPE) in the Sioux Falls network.

5.5.5. Link flows and travel times

To study the impact of accounting for the equilibrium condition in model training, we compare the performance of ODLULPE against a modified model that excludes the equilibrium loss component from the loss function, i.e., $\lambda_e = 0$. We denominate this model as ODLULPTTE, with the last TTE part referring to travel time estimation. Similar to Wu et al. (2018), the ODLULPTTE model produces solutions that do not satisfy the equilibrium condition. However, estimating travel times relaxes the assumption that travel times must be observed at every link. Furthermore, ODLULPTTE accounts for the relationship between traffic flows and travel times by encouraging that both quantities are consistent with the BPR function. Note that when $\lambda_e = 0$, the gradient of the loss function with respect to some of the parameters becomes zero by construction. To overcome this issue, we modify the original model as follows. First, the predicted link flow associated with the component ℓ_x of the loss function is redefined as the output link flows $\mathbf{x}$. As discussed in Section 3.6, the predicted link flow in our models is defined as the input link flows $\hat{\mathbf{x}}$. Second, we add an additional term in the loss function equal to the difference between travel times $\mathbf{t}$ and the output of the BPR function applied to



(a) LUE (b) ODLUE (c) ODLULPE

Fig. 7. Change in relative MSE of the LUE, ODLUE and ODLULPE models trained in the Sioux Falls network.

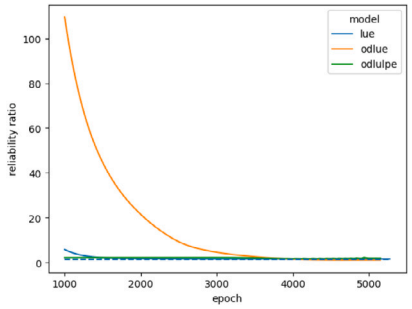


Fig. 8. Convergence to the true reliability ratio in the Sioux Falls network.

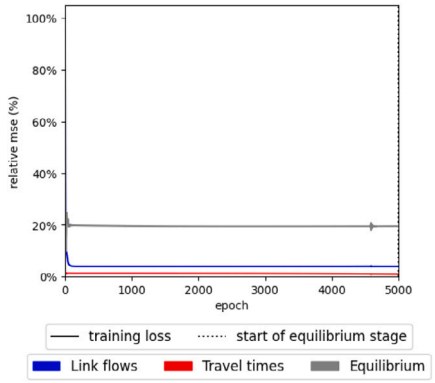


Fig. 9. Change in relative MSE of the ODLULPTTE model trained in the Sioux Falls network.

the output link flow **x**. This term encourages that the BPR relationship between the predicted travel time **t** and the predicted link flow **x** holds.

Fig. 9 shows the change in the relative MSE over epochs of the ODLULPTTE model. Results show that the equilibrium loss decreases during model training, but the reduction is not close to 100% as in the ODLULPE model (Fig. 7(c)). Due to the exclusion of the equilibrium component in the loss function, the equilibrium loss will not necessarily approach zero. The partial reduction of the equilibrium loss occurs because the algorithm tries to minimize the loss component associated with the observed link flows, whose average values satisfy the equilibrium condition (Section 5.1). The loss associated with the observed travel time decreases as expected.

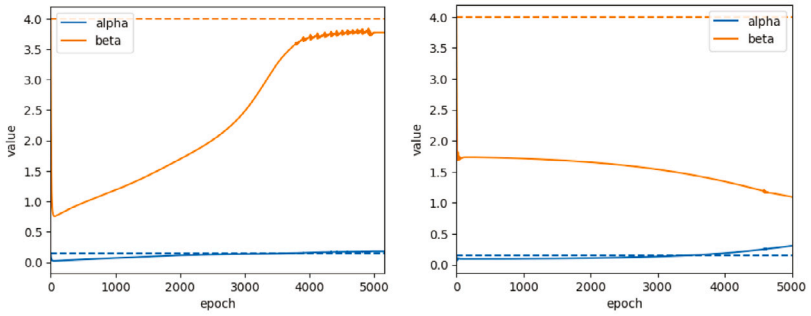
Tables 5 and 6 show the mean squared error (MSE) and mean absolute percentage error (MAPE) associated with link flows and travel times in the ODLULPE and ODLULPTTE models. The in-sample and out-of-sample errors are computed for the sets of links that do and do not report observed measurements, respectively. As discussed in Section 5.1, 25% of links are assumed to not report observed measurements of travel time and link flow. In line with our expectation, the training MSE is larger than

Table 5
In-sample and out-of-sample generalization mean squared error (MSE) for the models trained in the Sioux Falls network.

Model	Training MSE		In-sample generalization MSE		Out-of-sample generalization MSE	
	Traffic flow	Travel time	Traffic flow	Travel time	Traffic flow	Travel time
ODLULPE	1.3×10^6	1.2	1.1×10^4	0.2	1.3×10^5	0.9
ODLULPTTE	1.3×10^6	33	1.2×10^4	32.3	2.8×10^6	27.4

Table 6
In-sample and out-of-sample generalization mean absolute percentage error (MAPE) for the models trained in the Sioux Falls network.

Model	Training MAPE		In-sample generalization MAPE		Out-of-sample generalization MAPE	
	Traffic flow	Travel time	Traffic flow	Travel time	Traffic flow	Travel time
ODLULPE	9.2	17.8	0.8	3.0	2.6	5.3
ODLULPTTE	9.2	41.5	0.8	31.7	10.74	30.7



(a) ODLULPE (b) ODLULPTTE

Fig. 10. Convergence to the true link performance parameters in the Sioux Falls network when training the ODLULPE and ODLULPTTE models.

the in-sample generalization MSE. This is because the in-sample generalization MSE is computed using the ground truth observed measurements, whose values are not subject to random noise. Furthermore, the ODLULPE model achieves a lower training MSE than the ODLULPTTE model for the travel time measurements, which is in line with the better recovery of ODLULPE of the link performance parameters (Fig. 10). Finally, although both models achieve the same training and in-sample generalization MSEs for link flows, the ODLULPE model achieves a significantly lower out-of-sample generalization MSE in both link flow and travel time. Similar trends are observed in terms of MAPE (Table 6) where the ODLULPE model consistently outperforms the ODLULPTTE model. Overall, these results suggest that accounting for the equilibrium condition improves parameter recovery and generalization performance. Naturally, these results hold under the assumption that the ground truth measurements follow SUELOGIT.

6. Large scale implementation

The proposed methodology is applied to a large-scale transportation network in the City of Fresno, California (Fig. 11). This network primarily covers major roads and highways around the SR-41 corridor and it comprises 1789 nodes and 2413 links (Guarda and Qian, 2022). A historic O-D matrix with 6970 O-D pairs representative of a typical weekday at 4pm is also available for model training. To reduce the computational burden of path generation and path selection through the column generation method proposed by Guarda and Qian (2022), we use the augmented path set of size 18,289 generated by the authors for the same network and historic O-D matrix. The path set is assumed to be the same for every sample, meaning that the incidence matrices do not change between samples.

6.1. Data processing

The models are fitted and evaluated using measurements of travel time and link flow collected every 15 min between Tuesdays and Thursdays of October 2019 during the morning (6 am to 9 am) and afternoon (3 pm to 6 pm) peak hours. Traffic flow data at 15-min resolution is obtained from Caltrans Performance Measurement System (PeMS, 2021). Traffic speed with a 5 min resolution is obtained from INRIX (2021). Both data sources are used with a 1-hour resolution, meaning that the observed link flows are aggregated by hour, and the observed speeds are averaged by hour. A total of 141 and 2099 links of the Fresno network are matched to PeMS stations and INRIX street segments, respectively. The latter equates to coverages of 5.8% and 87% associated with the link flow and travel time measurements, respectively. INRIX data is used to compute at a link level the free flow travel time of the performance functions and the standard deviation of travel time by hour of the day and day of the week.

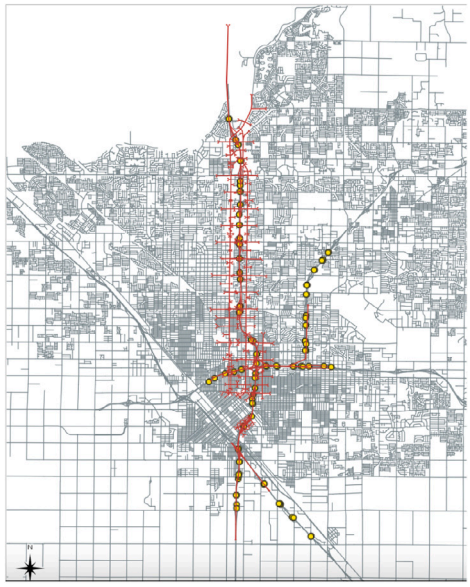


Fig. 11. SR-R1 corridor. The yellow circles represent the location of the traffic counters (PeMS, 2021).

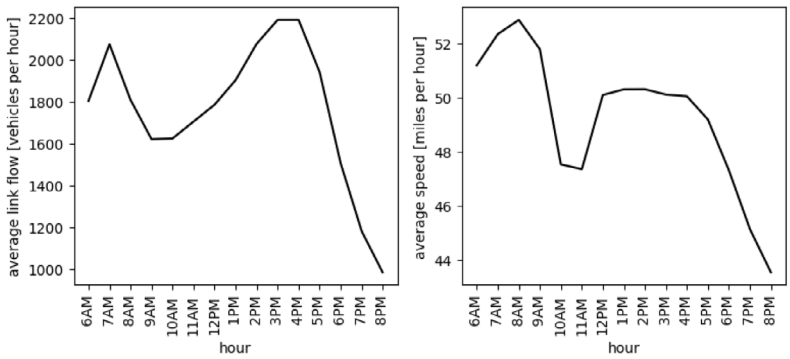


Fig. 12. Change of average link flow and average speed by hour of the day in October 2019.

The value of the rest of the exogenous attributes of the utility function, such as the number of bus stops, the number of street intersections, the number of yearly incidents, and the average monthly household income of the census block, are obtained from https://github.com/pabloguarda/isuelogit. For scaling purposes, the average travel time and the standard deviation of travel time are left in minute units.

6.2. Exploratory data analysis

Fig. 12 shows the hourly change in the average speed and link flow in October 2019. Fig. 13 shows the variation of speed [miles/hour], link flow [cars/hour], and the exogenous attributes of the travelers' utility function in October 2019. Some exogenous attributes, such as the median household income and the number of intersections and bus stops, are constant because their values vary among links but not between dates. The standard deviation of speed varies day to day because this attribute is computed using historical travel time information, which changes according to the hour of the day and day of the week.

6.3. Model specifications

Table 7 describes the specifications of the models estimated using data from the Fresno network. Similar to Guarda and Qian (2022), the travelers' utility function is linear-in-parameters and computed at the link level. The utility function is also assumed to be dependent on travel time and five exogenous attributes; the standard deviation of travel time, the number of bus stops, the number of street intersections, the number of yearly incidents, and the average monthly household income [1000 USD per month] of the census block.

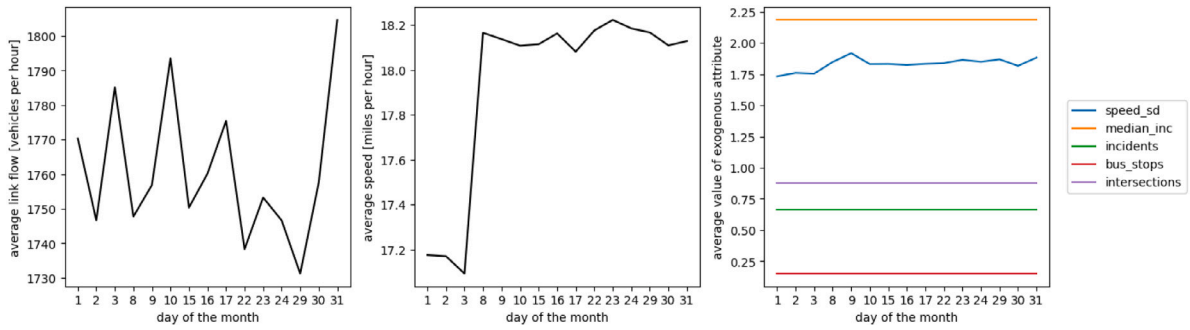


Fig. 13. Variation of link flow, speed, and the exogenous attributes of the utility function during Tuesdays, Wednesdays, and Thursdays of October 2019 between 6 am and 8 pm in Fresno, CA.

Table 7
Models specifications and parameters estimated with data from Fresno, CA.

Model	Number of parameters					
	Link flows	Attribute specific utility	Link specific utility	O-D matrix	Link performance function	Total
LUE	2,413	6	–	–	–	2,419
ODLUE	2,413	6	–	6,970	–	9,389
ODLULPE	2,413	6	–	6,970	$2 \times 2,413$	14,215
TVODLULPE	$2{,}413T$	$6T$	2,413	$6{,}970T$	$2 \times 2,413$	$7{,}239 + 9{,}389T$

T: number of distinct hourly periods in the TVODLULPE model.

To train the LUE and ODLUE models, the parameters of the link performance functions are assumed exogenous and equal to $\alpha = 0.15$ and $\beta = 4$. In contrast, the ODLULPE and TVODLULPE models estimate vectors of link-specific parameters $\hat{\boldsymbol{\alpha}}$ and $\hat{\boldsymbol{\beta}}$ that capture the heterogeneity of the link performance functions. To leverage data collected from multiple hours of the day, the TVODLULPE model estimates O-D matrices and utility functions by hour of the day. Furthermore, the utility function of the TVODLULPE model includes a link-specific parameter γ_a that takes the value 1 for the link a and 0 otherwise. These parameters act as fixed effects or intercepts that capture the impact of unobserved attributes in the utility function. These parameters are not included in the other models because, otherwise, the non-link specific parameters of the utility function would become not identifiable (Section 3.5.4).

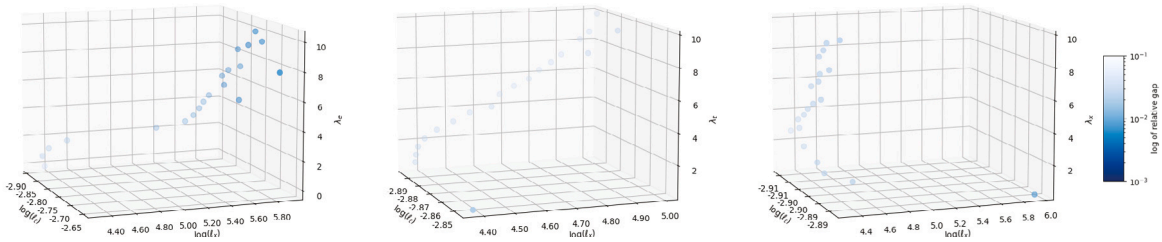
Remark 7. To reduce the number of parameters of the model, we assume that the O-D matrix and the travelers' utility function can change between hours but not between days of the week. This assumption is reasonable given that the observed measurements collected during peak hours of Tuesdays, Wednesdays, and Thursdays are expected to exhibit similar hourly patterns. Although fitting O-D matrices and utility function parameters by day and hour can reduce prediction error, it could also increase the risk of overfitting.

6.4. Loss function

Using multi-source data to estimate the model requires solving a multi-objective optimization problem (Wu et al., 2018). Differences in the range of values of the sources of observed measurements can produce an imbalance among the components of the objective function. Searching for weights of the loss components that correct this imbalance is possible. However, this can be computationally costly because it requires estimating models on a grid of hyperparameters whose dimensionality exponentially increases with the number of components in the loss function. To avoid hyperparameter tuning and additional computational burden, we apply various scaling and normalization strategies to balance the contribution of the loss components. For the Fresno network, the best results are obtained when the loss components associated with link flow and travel time are expressed as a root mean squared error (RMSE) divided by the mean of their observed values. As a consequence, because the observed travel times have a significantly lower mean than the observed link flows in our data, our normalization strategy scales down the values of the link flow component. Because the input link flows should be on a similar scale to the observed link flows, the equilibrium loss component is divided by the mean of the observed link flows. Normalizing the equilibrium loss component by the input link flows tends to increase numerical stability because the mean of the input link flows is not constant over epochs. We also test the normalization strategy suggested by Wu et al. (2018) but the results are not satisfactory in our application.

6.5. Estimation procedure

The models presented in this section are trained on an Apple M2 Pro with a 10-core CPU, 16 GB of unified memory, and 1 TB SSD. We use the Adam optimizer (Kingma and Ba, 2015), with a learning rate of 0.5 and a full batch of samples at every



(a) Equilibrium hyperparameter λ_e (b) Travel time hyperparameter λ_t (c) Link flow hyperparameter λ_x

Fig. 14. Value of the loss components for different combinations of hyperparameters.

epoch. The automatic differentiation routines of TensorFlow (Abadi et al., 2016) are crucial to reducing the computational burden of training the models' specifications (Table 7). Each sample in a batch gives a snapshot of the transportation network, namely, a set of observations of travel time and traffic flow collected at a given hour and day of the week. The LUE, ODLUE, and ODLULPE models are trained using hourly samples collected from Tuesdays, Wednesdays, and Thursdays between 4 pm and 5 pm in October 2019. The TVODLULPE model is trained with samples collected in the morning (6am to 9am) and afternoon (3 pm to 6 pm) peak hours during the same days in October 2019. The maximum number of epochs in the *learning stage* and the *equilibrating stage* are set to 100 and 50, respectively.

All models are trained with the same parameter initialization. The initial values of the utility function parameters are set to zero, and the initial values of the parameters $\hat{\boldsymbol{\alpha}}$ and $\hat{\boldsymbol{\beta}}$ of the link performance functions are set to 0.15 and 4, respectively. The values of the O-D matrix are initialized with the values of the available historical O-D matrix. Furthermore, to increase the stability of the model training and to comply with the monotone property of the link performance functions (Section 3.5.5), the values of the link-specific parameters $\hat{\boldsymbol{\alpha}}$ and $\hat{\boldsymbol{\beta}}$ are constrained to be between 0 and 8. Finally, similar to Guarda and Qian (2022), the parameters' domain during the optimization is constrained such that estimated parameters have signs consistent with our prior expectation.

6.6. Hyperparameter search

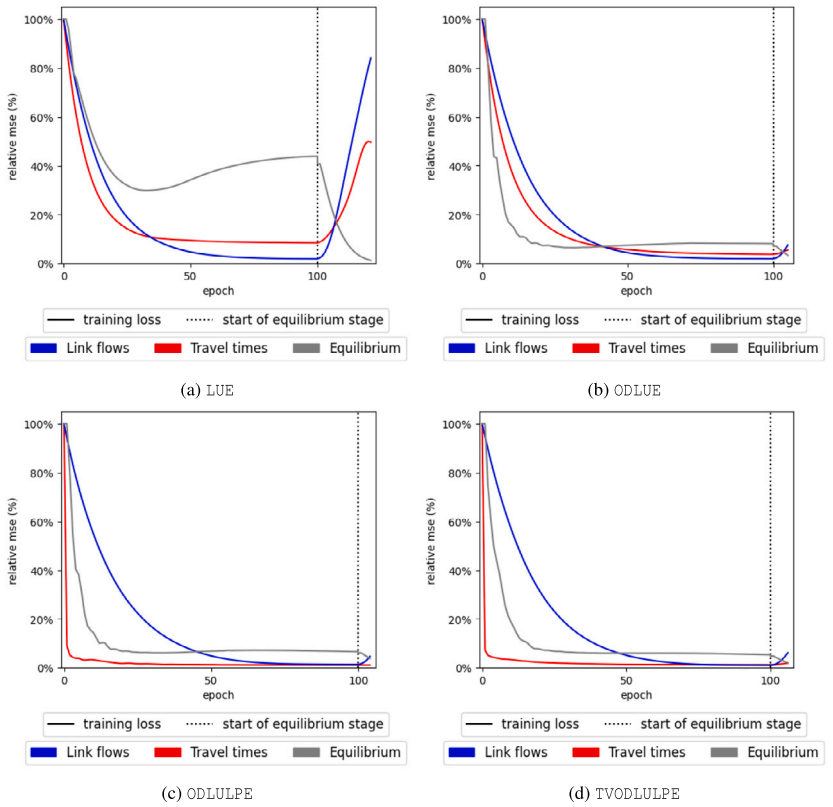
To understand the impact of the choice of hyperparameters of the loss function on model training, we conduct a grid search. Given that we had low confidence in the available historical O-D matrix, we set λ_q to zero. To set the rest of the hyperparameters, we conduct a grid search on 20 equally spaced values between 0.1 and 10. To analyze the impact of each hyperparameter in the final training loss, we conduct the grid search on one hyperparameter and fix the other two hyperparameters to 1. For each value in the grid, we train the ODLULPE model during the *learning stage* only and without imposing a threshold on the relative gap (Algorithm 1). The total runtime of the grid searches is approximately 10 h. Fig. 14 shows the variation of the logarithm of the travel time and link flow losses with respect to different combinations of the hyperparameters λ_t, λ_x and λ_e (z-axis). The color bar on the right represents the values of the relative gap at the end of model training on a logarithmic scale.

Results show that the link flow loss is more sensitive than the travel time loss to changes in the values of the hyperparameters. As shown in Figs. 14(a) and 14(b), the link flow loss increases in one order of magnitude if λ_e or λ_t are increased from 0.1 to 10. In contrast, the choice of hyperparameters has a negligible impact on the travel time loss. Fig. 14(c) suggests that setting the hyperparameter λ_x to zero is suboptimal since the resulting increase in the link flow error induces small reductions of the travel time loss and the relative gap. Another key finding is the trade-off between the equilibrium loss and the link flow prediction error. Therefore, reducing the relative gap threshold can highly compromise the link flow prediction error. As shown in Fig. 14(a), the link flow loss must increase in one order of magnitude to achieve relative gaps close to 10^{-3}. In this context, a threshold of 2×10^{-2} provides a proper trade-off between link flow and equilibrium losses. For training stability, it is also advisable that the choice of hyperparameters induces a monotonic decrease of the equilibrium loss over epochs. We find that setting all hyperparameters of the loss function to 1 generates this desired behavior during the optimization. Note that the proposed grid search strategy can help to find the best hyperparameter configurations for applications of our methodology in other transportation networks.

6.7. Convergence

Fig. 15 shows the change of relative MSE over epochs associated with the travel time, link flow, and the equilibrium components of the loss function. We observe that the reduction in the equilibrium loss is generally monotonic, which suggests that the choice of the hyperparameters of the loss function is appropriate. All models, except for LUE, require only a few iterations during the *equilibrating stage* to achieve the critical relative gap threshold of 2×10^{-2}. Due to the lower representational capacity of the LUE model, the reduction of equilibrium loss during the *equilibrating stage* must be compensated with a large increase in MSE of the link flow and travel time components.

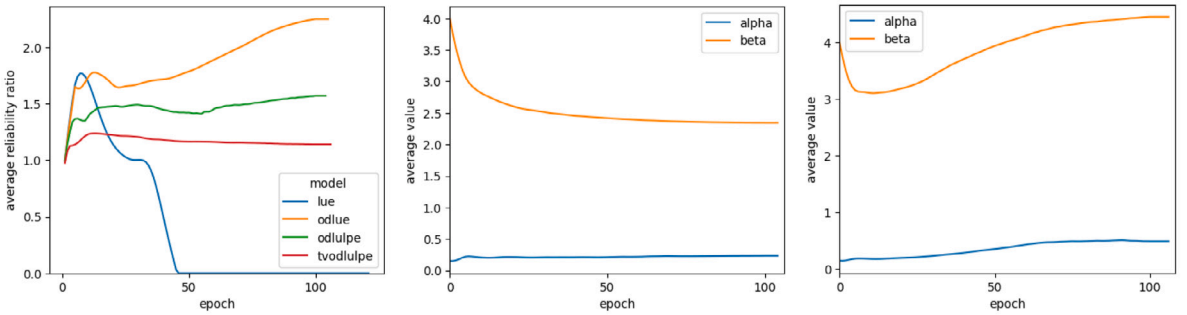
Fig. 16 shows the convergence of the average value of the link performance parameters and of the reliability ratio (RR) during model training. All models, except for LUE, converge to an average reliability ratio different than zero and ranging between 1 and 4. The value of the link performance parameters is only reported for ODLULPE and TVODLULPE as they are the only models where these parameters are estimated.



(a) LUE (b) ODLUE

(c) ODLULPE (d) TVODLULPE

Fig. 15. Convergence of relative MSE in models trained with data from Fresno, CA.



(a) All models (b) ODLULPE (c) TVODLULPE

Fig. 16. Convergence of the reliability ratio and the link performance parameters estimated with data from Fresno, CA.

6.8. Goodness of fit

Table 8 shows the runtimes, the relative gap, and the mean squared error (MSE) and mean absolute percentage error (MAPE) by loss component obtained with the four models. In parentheses, it is shown the percentage reduction in MSE of the models relative to the MSE obtained in the initial epoch.[1] Note that the LUE model has a larger runtime than the rest of the models because it requires more epochs to achieve the relative gap threshold of 2%. For the models estimated with data from a single hour, the ODLULPE model reports the highest MSE reduction for the travel time and link flow components. Due to its lower representational capacity, the LUE model is vastly outperformed by the ODLUE and ODLULPE models. Therefore, estimating the O-D matrix reduces the MSE of the link

[1] Because the parameters of the utility function are initialized to zero, the baseline MSE equates to a model where travelers are uniformly distributed among paths.

Table 8
Goodness of fit of the models.

Model	Metric							
	Runtime [s]	Total epochs	Relative gap	MSE equilibrium[a] [veh^2/hr^2]	MSE traffic flow [veh^2/hr^2]	MSE travel time [s^2]	MAPE traffic flow [%]	MAPE travel time [%]
LUE	665.6	121	1.8×10^{-2}	3.4×10^{3} (−98.8%)	1.5×10^{6} (−16.0%)	5.6×10^{-2} (−50.4%)	99.0	27.8
ODLUE	611.6	105	1.9×10^{-2}	1.0×10^{4} (−96.7%)	1.3×10^{5} (−92.5%)	6.2×10^{-3} (−94.6%)	35.3	12.8
ODLULPE	597.6	104	1.6×10^{-2}	1.1×10^{4} (−96.3%)	8.3×10^{4} (−95.4%)	1.2×10^{-3} (−98.9%)	30.6	5.0
TVODLULPE[b]	1455.7	106	1.7×10^{-2}	6.6×10^{3} (−98.0%)	1.3×10^{5} (−93.9%)	2.1×10^{-3} (−98.2%)	29.6	5.9

Note: the table reports the Mean Squared Error (MSE) and Mean Absolute Percentage Error (MAPE) at the end of model training. The sum of squared and percentage errors are averaged over the total number of observations across all samples. Thus, the MSE and MAPE associated with the predicted traffic flow and travel times can be interpreted as the squared and percentage errors that are obtained on average in a given link. The numbers shown in parentheses are the percentage reduction in MSEs with respect to the initial epoch.

[a] In the initial epoch, the MSE of the equilibrium component is zero because the utility parameters are initialized at zero. Thus, the percentage difference of MSEs shown in parentheses is computed relative to the MSE obtained in the following epoch.

[b] The TVODLULPE model uses the same parameter initialization as the other models, but its initial MSE is different because it is trained with a larger number of samples (Section 6.5).

flow and travel time loss components by at least one order of magnitude. Furthermore, estimating the link performance parameters in the ODLULPE model improves the capacity of the ODLUE model to reproduce the travel times and link flows observations in the training set. Notably, the performance of the TVODLULPE is similar to ODLULPE, even though the TVODLULPE must fit data collected in six different hours of the day.

Under the current model specification, the prediction errors of link flow and travel time can be interpreted as the irreducible day-to-day variance of the data-generating process. The prediction error could be minimized entirely if the O-D matrices are allowed to be different for each sample. However, this increases the risk of overfitting the training data, and thus, it may be detrimental to generalization performance.

6.9. Parameter estimation

Table 9 shows the parameters estimated with each model. All models find that the average travel time and the number of street intersections have a negative impact on the travelers' utility. In line with the findings reported by Guarda and Qian (2022), LUE is the only model that cannot properly capture the negative effect of the standard deviation of travel time. Notably, the reliability ratio obtained with the remaining models falls within the range of 0 to 3.5 that Carrion and Levinson (2012) report in an extensive review of travel time reliability studies. This result suggests that the estimation of the O-D matrix in the ODLUE, ODLULPE, and TVODLULPE models may contribute to reducing bias in estimating the utility function parameters. Furthermore, the estimation of the TVODLULPE model seems to reduce an overestimation of the O-D flows. Therefore, with respect to the historical O-D matrix, the average hourly trips estimated with the ODLUE and ODLULPE are almost two times larger, whereas they are only 15%–20% larger with the TVODLULPE model. Surprisingly, the running time per epoch of the TVODLULPE model is about three times larger than the rest of the models, even though this model fits six times more samples and four times more parameters than the latter. This is evidence that our algorithm can be scaled up to handle samples that are large and spread over multiple hours of the day.

Figs. 17(a) and 17(b) show the distribution of the fixed effects and of the link performance parameters $\hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\beta}}$ in the TVODLULPE model, respectively. We observe that the modes of the link performance parameters are close to the standard values of $\alpha = 0.15, \beta = 4$ assumed in transportation studies. Furthermore, the average value of the fixed effects is negative, which is consistent with the idea that the attributes in the utility function usually decrease travelers' utility. Nevertheless, the interpretation of the sign and value of the fixed effects should be looked at carefully because the model predictions would not change if an arbitrary positive constant were added in the utility function of every path[2].

6.10. Analysis by period

Fig. 18(a) shows the reliability ratio by hour of the day obtained with each model. The reliability ratio can be interpreted as the travelers' valuation of travel time variability relative to the expected travel time. Between 4 pm and 5 pm, all models except for LUE report a reliability ratio higher than 1, which suggests that travelers tend to penalize more a unit increase in the standard deviation than in the average travel time. Because the TVODLULPE estimates utility functions and O-D matrices specific to the time of day, we can also analyze the dynamics of travel behavior and demand among hours of the day. During the morning and afternoon peak

[2] Path choice probabilities are computed based on differences in utility and thus, any constant cancels out.

Table 9
Point estimates of models fitted with data collected between 4 pm and 5 pm in Tuesdays, Wednesdays and Thursdays of October 2019 in Fresno, CA.

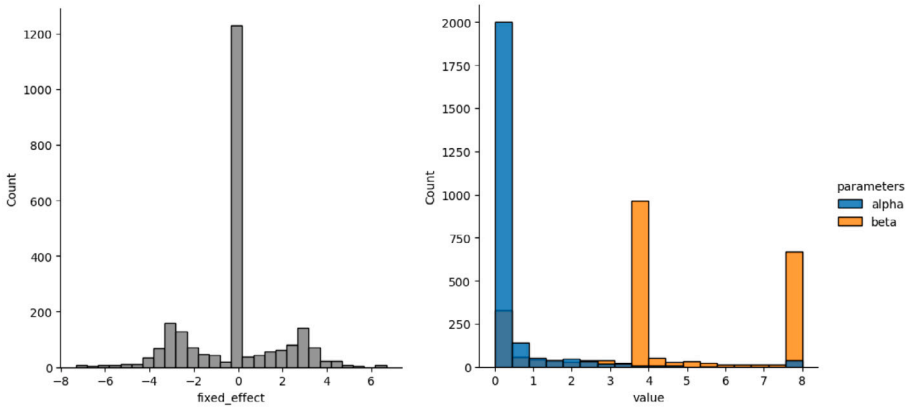
Parameter	Model			
	LUE	ODLUE	ODLULPE	TVODLULPE
Average travel time	−1.6238	−3.0614	−2.7104	−3.0588[a]
Std of travel time	0.0000	−6.8925	−4.2593	−3.4901[a]
Neighborhood income	0.0000	0.0000	0.0000	0.0000[a]
Incidents	0.0000	−0.5686	−0.8182	−4.7593[a]
Bus stops	0.0000	0.0000	0.0000	0.0000[a]
Streets Intersections	−1.4750	−1.5151	−1.8625	−3.804[a]
Reliability ratio	0.0000	2.2514	1.5715	1.1413[a]
Average fixed effects	0 (fixed)	0 (fixed)	0 (fixed)	−0.1401
Std. of fixed effects	0 (fixed)	0 (fixed)	0 (fixed)	2.0685
Average α	0.15 (fixed)	0.15 (fixed)	0.23	0.49
Std. α	0.00 (fixed)	0.00 (fixed)	0.50	1.20
Average β	4.00 (fixed)	4.00 (fixed)	2.34	4.45
Std. β	0.00 (fixed)	0.00 (fixed)	2.23	2.65
Average trips per O-D pair	9.51 (fixed)	20.59	19.17	13.85[a]
Std of trips per O-D pair	41.51 (fixed)	56.73	47.36	40.13[a]
Average hourly trips	66,266.3 (fixed)	143,525.4	133,646.4	96,535.2
Total parameters	2,419	9,389	14,215	63,573
Link flow observations	2,105	2,105	2,105	12,647
Travel time observations	31,314	31,314	31,314	186,612
Total observations	33,419	33,419	33,419	199,259
Total samples[b]	15	90	90	90
Runtime per epoch[c] [s]	6.1	6.0	5.9	14.3

Std.: standard deviation.

[a] Average value over periods.

[b] Each sample represents a snapshot of the transportation network at a given hour and day and it compresses a set of link flow and travel time observations for the subset of links where observations are available.

[c] Runtime does not include the computing time before model training, which includes tasks such as reading paths and data, and building the incidence matrices.

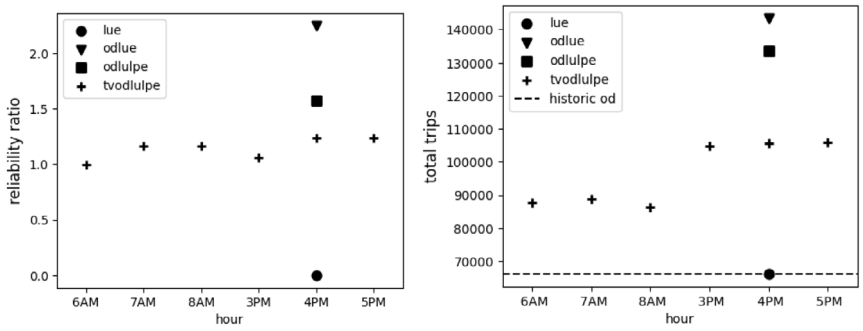


(a) Fixed effects in utility function (b) Link performance parameters

Fig. 17. Distribution of the link-specific parameters in the TVODLULPE model trained with data from the Fresno, CA network.

periods, the highest reliability ratio is reported at 8 am and 5 pm, respectively, which suggests that travelers are more averse to travel time reliability in the hours close to the end of the peak periods. Counter to our expectation, the slightly higher reliability ratios obtained during the afternoon period suggest that travelers are more averse to travel time reliability during the afternoon period. Therefore, given that most work starting times are during the morning, we had hypothesized that travelers should be more averse to travel time reliability during the morning period.

Fig. 18(b) shows the total trips by hour of the day. Because the LUE model does not estimate an O-D matrix, its value in the plot represents the total trips associated with the historical O-D matrix. Note that for the 4 pm−5 pm period, the total trips estimated by the other models surpass the total trips of the historical O-D matrix, which is plausible considering that the historical O-D matrix is outdated. Furthermore, the total trips of 579,211.3 estimated with the TVODLULPE model during the six hourly periods is reasonable. Fresno has 175,000 households (Census, 2022) and a typical household in the United States makes five vehicle trips per



(a) Reliability ratios (b) Total number of trips by hour

Fig. 18. Estimates of reliability ratios and total trips by hour of the day.

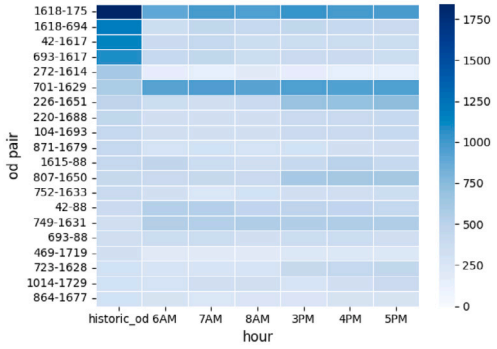


Fig. 19. Hourly trips obtained with the TVODLULPE model for the O-D pairs that reports the highest number of trips in the historical O-D matrix.

day (NHTS, 2018, p. 12), which equates to 875,000 daily trips approximately. The change in total trips between hours obtained with the TVODLULPE model is also consistent with the variation of link flows over hours. For instance, the total O-D flows estimated during the afternoon are higher than in the morning, which matches the pattern of hourly variation of the observed link flows (Fig. 12).

Finally, Fig. 19 shows a heatmap of the hourly trips within the O-D pairs that reports the highest number of trips in the historical O-D matrix. As expected, the hourly trip pattern tends to be more similar within the morning or afternoon periods, e.g., the numbers of trips in the O-D pairs 226-1651 and 807-1650 are similar within 6 am–9 am and 3 pm–6 pm but they are different between these 3-hour windows. Possibly due to the exclusion of the O-D loss component for model training, i.e., $\lambda_q = 0$, the number of trips estimated at 4 pm and those reported in the historical O-D matrix at the same hour do not look similar.

6.11. Cross-validation

To validate that the TVODLULPE model is not overfitting to the training set, we measure the model's predictive performance in a testing set. Similar to standard cross-validation procedures used in machine learning, we split the dataset into a training set and a testing set. To leave enough samples for model training and to comply with the temporal ordering of the data, the samples collected during the first three weeks and the last week of October are used as training and testing sets respectively.

Table 10 shows the MSE and MAPE associated with the traffic flow and travel time predictions in both the training and testing sets. Results show that the prediction error measured in terms of MSE is similar in the training and testing sets. Note that the MSE of travel time is slightly higher in the training set, which is possibly due to the lower variability of the travel time observations in the testing set. Furthermore, the MAPEs are lower in the test set, which may be due to an increase in the average value of the link flow and travel times observations during the last week of October. Overall, this evidence does not support that the TVODLULPE model overfits the training data.

7. Conclusions

Recent studies in the transportation field have leveraged advances in the ML community to address the limitations of existing methods to learn network flows and other parameters of transportation network models. These advances include using computational

Table 10
Cross-validation of TVODLULPE model.

Dataset	Number of samples	Metric			
		MSE flow	MSE travel time	MAPE flow	MAPE travel time
Training	72	1.5×10^5	2.5×10^{-3}	27.2	6.9
Test	18	1.7×10^5	2.2×10^{-3}	18.8	6.3

MSE: Mean Squared Error. MAPE: Mean Absolute Percentage Error.

graphs, neural networks, automatic differentiation tools, and first-order optimization algorithms. Despite these advances, there are still research gaps that must be addressed. First, the travelers' utility function usually depends on travel time only. As such, it does not capture the effect of other general attributes that travelers consider in their route choice decisions. Second, in static traffic assignment settings, the parameters of the link performance functions are assumed exogenous or, at best, endogenous but homogeneous among links. Third, most existing models rely on bilevel optimization methods to generate solutions that reproduce a set of observed measurements and satisfy the network equilibrium conditions. These methods often require significant hyperparameter tuning to achieve convergence.

To estimate network flow under recurrent traffic conditions, this research leverages computational graphs and spatio-temporal multi-source system-level data to solve a single-level optimization problem via gradient-based methods. Because this formulation is consistent with the stochastic user equilibrium under logit assignment (SUELOGIT), the model parameters are easy to interpret, and the solution satisfies equilibrium conditions. Our model learns O-D matrices, utility functions, link flows, path flows, and travel times that are specific to the hour of day. To increase the model's representational capacity to reproduce observed link flows and travel times, the parameters of the link performance functions are assumed link-specific. More importantly, the utility function in the route choice model is enriched with (i) link-specific parameters to capture the effect of unobserved attributes on route choices and (ii) parameters specific to the time of day that are weighting the observed features in the utility function and that capture the heterogeneity of travelers preferences among periods of the day.

Numerical experiments in the Sioux Falls network helped study parameter recovery under various model specifications. We started computing SUELOGIT, which only defines the link flows as learnable parameters. We observed that a solution that satisfies the network equilibrium conditions could be obtained with a small computational burden. Subsequently, we let additional parameters of the models to be learnable, such as the parameters of the utility function, the link performance function, and the O-D matrix. Generally, the parameter recovery is worst when a larger number of parameters is estimated, which may be explained by increasing the non-linearity of the optimization landscape. Regardless, all model specifications show that the parameter recovery is robust to noise in the observed measurements and that the equilibrium conditions can be satisfied with high accuracy. One of the experiments also suggests that if the ground truth measurements follow SUELOGIT, accounting for the equilibrium condition helps to improve parameter recovery and goodness of fit.

Implementing our algorithm in the SR-41 corridor of the Fresno network is challenging because the models are high dimensional, and the amount of spatio-temporal data is large. Here, using computational graphs and automatic differentiation techniques is crucial to reducing the computational burden of model training. Our results show that the algorithm could scale well with data from multiple periods and models that estimate O-D matrices and utility function parameters that are specific to the time of day. In addition, the algorithm can find solutions that satisfy the equilibrium conditions and reproduce observations collected from multiple sources with reasonable accuracy. Furthermore, the estimates of the parameters of the link performance functions and the utility function obtained with real-world data are reasonable in terms of sign and magnitude. Notably, the estimates of the *reliability ratio*, namely, the ratio between the parameters of the standard deviation and the average travel time in the utility function, fall within the range reported in prior literature. Regarding O-D estimation, our results show that the change in the total trips between periods is informative of the transportation network dynamics during the morning and afternoon peak hours. Besides, the model specifications that estimate an O-D matrix seem to correct bias in the estimation of the attribute-specific coefficients of the utility function. For example, the sign of the standard deviation of travel time becomes negative when the O-D matrix is estimated, and this result is robust across time periods and model specifications.

8. Further research

Further research should focus on the predictive performance of the models. For this purpose, regularization techniques with cross-validation procedures should be used to prevent overfitting. It would also be interesting to incorporate some ideas developed in the context of physics-informed neural networks (Raissi et al., 2019) to have a more flexible representation of the mapping of traffic flow to travel times. Therefore, the use of a surrogate model rooted in deep neural networks to model this mapping could increase model representational capacity, and it avoids prespecifying the class of performance functions. An important challenge when implementing these extensions is to balance model interpretability and generalization performance.

Our work also has some limitations that could be overcome in future research. First, the performance function assumes symmetric cost functions where there are no interactions between flows of adjacent links. This assumption could be relaxed by enriching the layer in the computational graph that maps traffic flow into travel times. Under this modification, the minimization of the fixed point component in the loss function will remain a valid tool for the computation of network equilibrium. Second, due to data limitation, our model did not test the effect of non-link-additive attributes in the utility function, e.g., a binary variable taking the value one if

the path is habitual or 0, otherwise (Florian and Hearn, 1999). Thanks to the flexibility of our computational graph and the fact that our model is path-based, the incorporation of path-specific attributes should be straightforward. Third, our model only considers the existence of one mode of transportation and a single class of users. The mathematical models developed by Abrahamsson and Lundqvist (1999), Florian et al. (2002), Yang and Chen (2009) and Ryu et al. (2017) could be helpful to solve a combined modal split and traffic assignment problem in the presence of multiple classes of users in our computational graph framework.

The estimation of the O-D matrix could be improved in two ways. First, the loss function may be augmented to ensure that the estimated O-D matrix satisfies some constraints imposed by the modeler, e.g., the sum of trips should not surpass a certain threshold. We observed that some hyperparameter settings induced an overestimation of the total trips in the time-varying model. Second, a better initialization of the O-D matrices that are estimated during the morning periods could also contribute to better O-D estimation. Currently, the O-D matrices of all periods are initialized using the values of a historic O-D matrix that is representative of the afternoon peak period. Furthermore, GPS data could be leveraged to have a better prior of the path sets among O-D pairs and to improve the estimation of the utility function coefficients. Paths with high utility are associated with high choice probabilities, and thus, their omission from the path sets may have an impact in the estimation results. Lastly, the utility function coefficients could be allowed to vary by O-D pair to capture the heterogeneity of travelers among different locations in the transportation network.

9. Model implementation and data

The Python code that implements our methodology and the data collected from the Fresno, CA network can be found at https://github.com/pabloguarda/pesuelogit. The folder `notebooks` contains Jupyter notebooks that reproduce all the results presented in this paper.

CRediT authorship contribution statement

Pablo Guarda: Study conception and design, Data collection, Programming and experiments, Analysis and interpretation of results, Writing – original draft. **Matthew Battifarano:** Study conception and design, Programming and experiments, Analysis and interpretation of results, Writing – original draft. **Sean Qian:** Study conception and design, Data collection, Analysis and interpretation of results, Writing – original draft.

Acknowledgments

This research is supported by a National Science Foundation grant CMMI-1751448. All authors reviewed the results and approved the final version of the manuscript.

References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., 2016. TensorFlow: A system for large-scale machine learning. pp. 265–283.

Abrahamsson, T., Lundqvist, L., 1999. Formulation and estimation of combined network equilibrium models with applications to Stockholm. Transp. Sci. 33 (1), 80–100. http://dx.doi.org/10.1287/trsc.33.1.80, URL: https://pubsonline.informs.org/doi/10.1287/trsc.33.1.80.

Anas, A., Kim, I., 1990. Network loading versus equilibrium estimation of the stochastic route choice model maximum likelihood and least squares revisited. J. Reg. Sci. 30 (1), 89–103. http://dx.doi.org/10.1111/j.1467-9787.1990.tb00082.x.

Bekhor, S., Toledo, T., 2005. Investigating path-based solution algorithms to the stochastic user equilibrium problem. Transp. Res. B 39 (3), 279–295. http://dx.doi.org/10.1016/j.trb.2004.04.006.

Bell, M.G., Shield, C.M., Busch, F., Kruse, G., 1997. A stochastic user equilibrium path flow estimator. Transp. Res. C 5 (3–4), 197–210. http://dx.doi.org/10.1016/S0968-090X(97)00009-0, URL: https://linkinghub.elsevier.com/retrieve/pii/S0968090X97000090.

Caggiani, L., Ottomanelli, M., 2011. Calibration of equilibrium traffic assignment models and O-D matrix by network aggregate data. Adv. Intell. Soft Comput. 96 AISC, 359–367. http://dx.doi.org/10.1007/978-3-642-20505-7_32, ISBN: 9783642205040.

Cantarella, G.E., 1997. A general fixed-point approach to multimode multi-user equilibrium assignment with elastic demand. Transp. Sci. 31 (2), 107–128. http://dx.doi.org/10.1287/trsc.31.2.107, URL: http://pubsonline.informs.org/doi/10.1287/trsc.31.2.107.

Carrion, C., Levinson, D., 2012. Value of travel time reliability: A review of current evidence. Transp. Res. A 46 (4), 720–741. http://dx.doi.org/10.1016/j.tra.2012.01.003, URL: https://linkinghub.elsevier.com/retrieve/pii/S0965856412000043.

Cascetta, E., Postorino, M.N., 2001. Fixed point approaches to the estimation of O/D matrices using traffic counts on congested networks. Transp. Sci. 35 (2), 134–147. http://dx.doi.org/10.1287/trsc.35.2.134.10138.

Cascetta, E., Russo, F., 1997. Calibrating aggregate travel demand models with traffic counts: Estimators and statistical performance. Transportation 24, 271–293, Publisher: Springer.

Census, U., 2022. QuickFacts. Fresno city, California. URL: https://www.census.gov/quickfacts/fresnocitycalifornia.

Connors, R.D., Sumalee, A., 2009. A network equilibrium model with travellers' perception of stochastic travel times. Transp. Res. B 43 (6), 614–624. http://dx.doi.org/10.1016/j.trb.2008.12.002, Publisher: Elsevier Ltd.

Daganzo, C.F., Sheffi, Y., 1977. On stochastic models of traffic assignment. Transp. Sci. 11 (3), 253–274. http://dx.doi.org/10.1287/trsc.11.3.253, URL: http://pubsonline.informs.org/doi/abs/10.1287/trsc.11.3.253.

Fisk, C., 1977. Note on the maximum likelihood calibration on Dial's assignment method. Transp. Res. 11 (1), 67–68. http://dx.doi.org/10.1016/0041-1647(77)90069-7, URL: https://linkinghub.elsevier.com/retrieve/pii/0041164777900697.

Fisk, C., 1980. Some developments in equilibrium traffic assignment. Transp. Res. B 14 (3), 243–255. http://dx.doi.org/10.1016/0191-2615(80)90004-1.

Fisk, C.S., 1989. Trip matrix estimation from link traffic counts: The congested network case. Transp. Res. B 23 (5), 331–336. http://dx.doi.org/10.1016/0191-2615(89)90009-X.

Florian, M., Hearn, D., 1999. Network equilibrium and pricing. In: Handbook of Transportation Science. pp. 361–393, Publisher: Springer.

Florian, M., Wu, J.H., He, S., 2002. A multi-class multi-mode variable demand network equilibrium model with hierarchical logit structures. In: Pardalos, P.M., Hearn, D., Gendreau, M., Marcotte, P. (Eds.), Transportation and Network Analysis: Current Trends, Vol. 63. Springer US, Boston, MA, pp. 119–133. http://dx.doi.org/10.1007/978-1-4757-6871-8_8, URL: http://link.springer.com/10.1007/978-1-4757-6871-8_8, Series Title, Applied Optimization.

García-Ródenas, R., Marín, A., 2009. Simultaneous estimation of the origin-destination matrices and the parameters of a nested logit model in a combined network equilibrium model. European J. Oper. Res. 197 (1), 320–331. http://dx.doi.org/10.1016/j.ejor.2008.05.032, Publisher: Elsevier B.V.

García-Ródenas, R., Verastegui-Rayo, D., 2013. Adjustment of the link travel-time functions in traffic equilibrium assignment models. Transportmetrica A 9 (9), 798–824. http://dx.doi.org/10.1080/18128602.2012.669415.

Gentile, G., 2018. New formulations of the stochastic user equilibrium with logit route choice as an extension of the deterministic model. Transp. Sci. 52 (6), 1531–1547. http://dx.doi.org/10.1287/trsc.2018.0839, URL: http://pubsonline.informs.org/doi/10.1287/trsc.2018.0839.

Guarda, P., Qian, S., 2022. Statistical inference of travelers' route choice preferences with system-level data. URL: http://arxiv.org/abs/2204.10964, arXiv: 2204.10964 [physics, stat].

INRIX, 2021. Inrix Analytics: Speed. URL: https://inrix.com/products/speed/.

Kim, T., Zhou, X., Pendyala, R.M., 2021. Computational graph-based framework for integrating econometric models and machine learning algorithms in emerging data-driven analytical environments. Transportmetrica A http://dx.doi.org/10.1080/23249935.2021.1938744.

Kingma, D.P., Ba, J.L., 2015. Adam: A method for stochastic optimization. In: 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings. pp. 1–15, arXiv:1412.6980.

Krishnakumari, P., van Lint, H., Djukic, T., Cats, O., 2020. A data driven method for OD matrix estimation. Transp. Res. C 113 (December 2018), 38–56. http://dx.doi.org/10.1016/j.trc.2019.05.014, Publisher: Elsevier.

Kurmankhojayev, D., Hu, Z., Chen, A., Ma, W., 2022. Simultaneous calibration of traffic demand and transport network using real-world multi-source data. J. Eastern Asia Soc. Transp. Stud. 14, 601–615, Publisher: Eastern Asia Society for Transportation Studies.

Li, J., Yu, J., Nie, Y., Wang, Z., 2020. End-to-end learning and intervention in games. Adv. Neural Inf. Process. Syst. 33, 16653–16665.

Liu, S., Fricker, J.D., 1996. Estimation of a trip table and the theta parameter in a stochastic network. Transp. Res. A 30 (4 PART A), 287–305. http://dx.doi.org/10.1016/0965-8564(95)00031-3.

Liu, Z., Yin, Y., Bai, F., Grimm, D.K., 2023. End-to-end learning of user equilibrium with implicit neural networks. Transp. Res. C 150, 104085. http://dx.doi.org/10.1016/j.trc.2023.104085, URL: https://linkinghub.elsevier.com/retrieve/pii/S0968090X23000748.

Lo, H.P., Chan, C.P., 2003. Simultaneous estimation of an origin-destination matrix and link choice proportions using traffic counts. Transp. Res. A 37 (9), 771–788. http://dx.doi.org/10.1016/S0965-8564(03)00048-X.

Ma, W., Pi, X., Qian, S., 2020. Estimating multi-class dynamic origin-destination demand through a forward-backward algorithm on computational graphs. Transp. Res. C 119 (August), 102747. http://dx.doi.org/10.1016/j.trc.2020.102747.

Ma, W., Qian, S., 2018a. A generalized single-level formulation for origin–destination estimation under stochastic user equilibrium. Transp. Res. Rec. 2672 (48), 58–68. http://dx.doi.org/10.1177/0361198118782041.

Ma, W., Qian, S., 2018b. Statistical inference of probabilistic origin-destination demand using day-to-day traffic data. Transp. Res. C 88 (August 2017), 227–256. http://dx.doi.org/10.1016/j.trc.2017.12.015, Publisher: Elsevier.

McFadden, D., 1973. Conditional logit analysis of qualitative choice behavior. Front. Econ. 105–142. http://dx.doi.org/10.1108/eb028592, ISBN: 0127761500.

McFadden, D., 2001. Nobel Prize Lecture: Economic Choices. Amer. Econ. Rev. 91 (3), 351–378.

NHTS, 2018. Summary of Travel Trends 2017. Technical Report ORNL/TM-2004/297, 885762, National Household Travel Survey, http://dx.doi.org/10.2172/885762, URL: https://nhts.ornl.gov/assets/2017_nhts_summary_travel_trends.pdf.

Ortúzar, J.d.D., Willumsen, L.G., 2011. Modelling Transport. John Wiley & Sons, Chichester.

PeMS, 2021. Performance measurement system. URL: https://pems.dot.ca.gov/.

Raissi, M., Perdikaris, P., Karniadakis, G., 2019. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. J. Comput. Phys. 378, 686–707. http://dx.doi.org/10.1016/j.jcp.2018.10.045, URL: https://linkinghub.elsevier.com/retrieve/pii/S0021999118307125.

Robillard, P., 1974. Calibration of dial's assignment method. Transp. Sci. 8 (2), 117–125. http://dx.doi.org/10.1287/trsc.8.2.117.

Russo, F., Vitetta, A., 2011. Reverse assignment: Calibrating link cost functions and updating demand from traffic counts and time measurements. Inverse Probl. Sci. Eng. 19 (7), 921–950. http://dx.doi.org/10.1080/17415977.2011.565339.

Ryu, S., Chen, A., Choi, K., 2017. Solving the combined modal split and traffic assignment problem with two types of transit impedance function. European J. Oper. Res. 257 (3), 870–880. http://dx.doi.org/10.1016/j.ejor.2016.08.019, URL: https://linkinghub.elsevier.com/retrieve/pii/S0377221716306348.

Sherali, H.D., Sivanandan, R., Hobeika, A.G., 1994. A linear programming approach for synthesizing origin-destination trip tables from link traffic volumes. Transp. Res. B 28 (3), 213–233. http://dx.doi.org/10.1016/0191-2615(94)90008-6, URL: https://linkinghub.elsevier.com/retrieve/pii/0191261594900086.

Spiess, H., 1990. Technical note—Conical volume-delay functions. Transp. Sci. 24 (2), 153–158. http://dx.doi.org/10.1287/trsc.24.2.153, URL: http://pubsonline.informs.org/doi/10.1287/trsc.24.2.153.

Suh, S., Park, C.-H., Kim, T.J., 1990. A highway capacity function in Korea: Measurement and calibration. Transp. Res. A 24 (3), 177–186. http://dx.doi.org/10.1016/0191-2607(90)90055-B, URL: https://linkinghub.elsevier.com/retrieve/pii/019126079090055B.

TNTP, 2021. Transportation networks for research core team. URL: https://github.com/bstabler/TransportationNetworks.

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 (1), 1–24. http://dx.doi.org/10.1371/journal.pone.0146850.

Wollenstein-Betech, S., Sun, C., Zhang, J., Cassandras, C.G., Paschalidis, I.C., 2022. Joint data-driven estimation of origin–destination demand and travel latency functions in multiclass transportation networks. IEEE Trans. Control Netw. Syst. 9 (4), 1576–1588. http://dx.doi.org/10.1109/TCNS.2022.3161200, URL: https://ieeexplore.ieee.org/document/9739921/.

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. Transp. Res. C 96 (September), 321–346. http://dx.doi.org/10.1016/j.trc.2018.09.021.

Yang, C., Chen, A., 2009. Sensitivity analysis of the combined travel demand model with applications. European J. Oper. Res. 198 (3), 909–921. http://dx.doi.org/10.1016/j.ejor.2008.09.044, URL: https://linkinghub.elsevier.com/retrieve/pii/S0377221708008710.

Yang, Y., Fan, Y., Wets, R.J., 2018. Stochastic travel demand estimation: Improving network identifiability using multi-day observation sets. Transp. Res. B 107, 192–211. http://dx.doi.org/10.1016/j.trb.2017.10.007, Publisher: Elsevier Ltd.

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. Transp. Sci. 35 (2), 107–123. http://dx.doi.org/10.1287/trsc.35.2.107.10133.