

SPO-VCS: An End-to-End Smart Predict-then-Optimize Framework with Alternating Differentiation Method for Relocation Problems in Large-Scale Vehicle Crowd Sensing

Xinyu Wang^a, Yiyang Peng^a, Wei Ma^{a,*}

^a*Civil and Environmental Engineering, The Hong Kong Polytechnic University
Hung Hom, Kowloon, Hong Kong SAR, China*

Abstract

Ubiquitous mobile devices have catalyzed the development of vehicle crowd sensing (VCS). In particular, vehicle sensing systems show great potential in the flexible acquisition of extensive spatio-temporal urban data through built-in smart sensors under diverse sensing scenarios. However, vehicle systems like taxis often exhibit biased coverage due to the heterogeneous nature of trip requests and varying routes. To achieve a high sensing coverage, a critical challenge lies in how to optimally relocate vehicles to minimize the divergence between the spatio-temporal distributions of vehicles and target sensing distributions. Conventional approaches typically employ a two-stage predict-then-optimize (PTO) process: first predicting real-time vehicle distributions and subsequently generating an optimal relocation strategy based on the predictions. However, this approach can lead to suboptimal decision-making due to the propagation of errors from upstream prediction. To this end, we develop an end-to-end Smart Predict-then-Optimize (SPO) framework by integrating optimization into prediction within the deep learning architecture, and the entire framework is trained by minimizing the task-specific matching divergence rather than the upstream prediction error. Methodologically, we formulate the vehicle relocation problem by quadratic programming (QP) and incorporate a novel unrolling approach based on the Alternating Direction Method of Multipliers (ADMM) within the SPO framework to compute gradients of the QP layer, facilitating backpropagation and gradient-based optimization for end-to-end learning. The effectiveness of the proposed framework is validated using two real-world taxi datasets ranging from mid-size to large-scale in Hong Kong and Chengdu, China. Utilizing the alternating differentiation method, the general SPO framework presents a novel concept of addressing decision-making problems with uncertainty, demonstrating significant potential for advancing applications in logistics and intelligent transportation systems.

Keywords: Vehicle Relocation, Vehicle Crowd Sensing (VCS), Smart Predict-then-Optimize (SPO), Alternating Direction Method of Multipliers (ADMM), Unrolling Approach

1. Introduction

In recent years, the development of ubiquitous mobile devices such as mobile phones has significantly advanced the field of mobile crowd sensing (MCS) (Ganti et al. 2011). Compared with the traditional fixed sensor networks, MCS offers more flexible and extensive coverage of sensing information while reducing the installation and collection cost (Li et al. 2018a). Individuals participating in MCS utilize the built-in smart sensors in mobile devices such as cameras and GPS, to sense the surroundings and gather city-wide information (Li et al. 2022b).

The MCS can be categorized into community sensing (CS) and vehicle crowd sensing (VCS) based on different types of sensor hosts (Ji et al. 2023a). CS utilizes sensors installed in mobile devices, while VCS leverages the sensing capabilities of vehicles to collect city-wide data (Xu et al. 2019). Host vehicles of VCS include taxis (Honicky et al. 2008), trams (Saukh et al. 2014), buses (Kang et al. 2016), and unmanned aerial vehicles (UAVs) (Li

*Corresponding author

Email addresses: xinyuu.wang@connect.polyu.hk (Xinyu Wang), yiyang.peng@connect.polyu.hk (Yiyang Peng), wei.w.ma@polyu.edu.hk (Wei Ma)

et al. 2022a). In general, the advantages of VCS in terms of temporal duration, geographical coverage, reliability, and consistency make it particularly suitable for scenarios requiring continuous, full coverage, and high-quality data (Ji et al. 2023a).

Consequently, VCS is instrumental in domains that require high-precision data, such as infrastructure monitoring (Song et al. 2020; Hasenfratz et al. 2015; Liu and Ma 2022), traffic state estimation (Guo et al. 2022; Zhu et al. 2014) and traffic management (Ji et al. 2023b; Dai and Han 2023).

Specifically, the *vehicles* in the VCS framework are classified into two types: dedicated vehicles (DVs) and non-dedicated vehicles (NDVs). DVs, such as vehicles equipped with air pollution sensors and delivery drones with cameras, are fully guided by the dispatching center to prioritize sensing tasks. In contrast, NDVs, such as e-hailing vehicles and taxis, are under the complete control of their drivers and do not follow the guidance of the dispatching center. Upon receiving a sensing task, DVs will proceed immediately to the specific sensing locations. In contrast, NDVs operate under the control of drivers and are confined to their preferred cruising areas and routes. Moreover, NDVs can be converted into DVs through incentives such as monetary compensation.

Figure 1 provides an overview of the sensing procedure in the VCS framework. The *data request platform* initially publishes sensing tasks. Upon receiving these tasks, the *dispatching center* develops relocation strategies for DVs. During the subsequent time intervals, all the *sensing vehicles*, including DVs and NDVs, move to the specified destinations to perform the assigned sensing tasks. After collecting the necessary data, the sensing vehicles will transmit sensing information back to the *data request platform*.

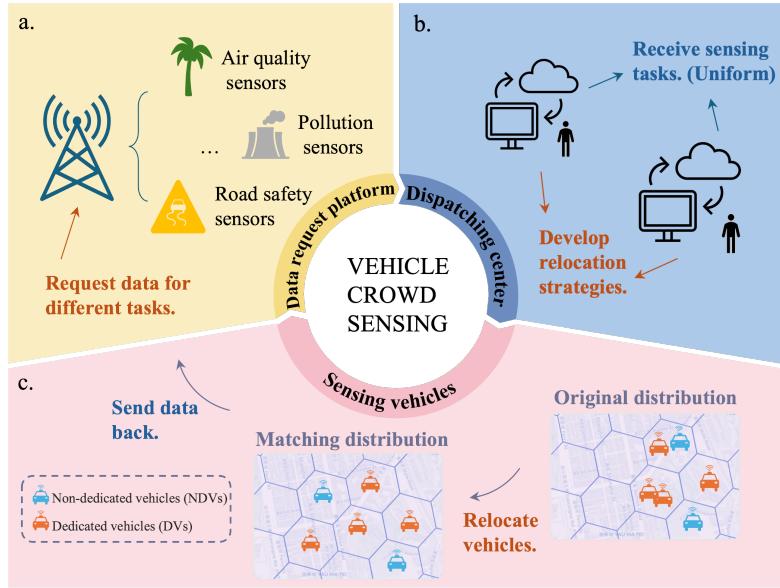


Figure 1: An illustration of the sensing procedure in the VCS framework.

In the domain of VCS, the target distribution of sensing tasks differs markedly from the spatio-temporal distribution of sensors. Target distributions typically vary based on specific sensing objectives and adapt over time and space. For instance, monitoring air quality may necessitate uniformly distributed data across a city (Bales et al. 2012; Chen et al. 2018), whereas tasks focusing on industrial pollution may require the spatial distribution in Gaussian (Khedo et al. 2010; Paulos et al. 2007). In contrast, the spatio-temporal distributions of sensors are heavily influenced by the movements of host sensing vehicles (Anjomshoaa et al. 2018).

Therefore, a critical challenge in VCS lies in the effective relocation of vehicles to align with the target sensing distribution. To be specific, the dispatching center needs to develop relocation strategies for DVs to minimize the divergence between the target sensing distributions and vehicle distributions. The state-of-the-art approach to addressing the vehicle relocation problem in VCS is a two-stage predict-then-optimize (PTO) method, where a prediction model is first established to forecast the movements of NDVs (Chang et al. 2023; Simchi-Levi 2014; Mišić and Perakis 2020). The prediction is then used as the input for the downstream optimization problem related to DVs. However, a key limitation of the PTO approach is that training the prediction model is mainly based on

empirical error minimization, and metrics such as Mean Squared Error (MSE) and Mean Absolute Error (MAE) are used. The trained prediction model may not necessarily yield optimal strategies in the subsequent optimization stage, because the focus of the optimization stage is task-specific, which may not align well with the metrics of MSE or MAE (Bengio 1997; Ford et al. 2015; Elmachtoub and Grigas 2022; Yan and Wang 2022). This discrepancy highlights the need for more integrated approaches that bridge prediction and optimization phases coherently in the context of VCS.

Recently, the end-to-end Smart Predict-then-Optimize (SPO) framework has been proposed and attracted wide attention (Elmachtoub and Grigas 2022). This framework integrates prediction and optimization within a unified deep-learning architecture. Specifically, the optimization module operates as a differentiation layer, mapping inputs directly to optimal solutions without requiring explicit closed-form solutions. This approach enables the entire neural network to be trained by directly minimizing task-specific errors. However, the application of the SPO framework to address challenges such as vehicle relocation has not been extensively explored.

The main challenge in applying the SPO framework is integrating the optimization layer seamlessly within the deep learning architecture. A widely used approach is the implicit differentiation, where the Karush-Kuhn-Tucker (KKT) conditions of the optimization problems are differentiated to obtain first-order derivatives of the optimal solution with respect to the parameters. However, this method can be computationally demanding in directly computing the Jacobian matrix of the KKT conditions for large-scale problems (Diamond and Boyd 2016; Agrawal et al. 2019; Tang and Khalil 2022).

In this paper, we advocate an alternative approach - explicit unrolling, which avoid the direct computation of the Jacobian matrix in the KKT conditions and compute the corresponding gradients iteratively, enhancing computational efficiency and suitability for large-scale problems (Sun et al. 2023; Li et al. 2020). Nevertheless, the majority of unrolling approaches are predominantly applicable to unconstrained problems. In the task of VCS, relocation challenges are usually formulated as constrained optimization problems. To this end, we incorporate the Alternating Direction Method of Multipliers (ADMM) to solve constrained optimization problems by decomposing the original problem into a series of unconstrained sub-problems (Boyd et al. 2011). Combining the ADMM with differentiation methods and unrolling approaches can potentially address the challenge of integrating large-scale constrained convex optimization problems into deep learning architectures.

To conclude, there are two major issues with the predict-and-optimize problem in VCS. First, the conventional PTO framework for vehicle relocation problems may not generate an optimal relocation strategy, resulting in a substantial divergence between the desired target distribution and the actual vehicle distribution. Second, embedding an optimization layer within neural networks is usually computationally intensive and impractical for large-scale networks, and existing unrolling approaches are often inadequate for handling constrained optimization problems effectively.

To address these challenges, we develop an innovative SPO framework coupled with an alternating differentiation method for vehicle relocation problems in VCS, aiming to achieve optimal sensing coverage. Specifically, we formulate the vehicle relocation problem based on constrained quadratic programming (QP) and embed the QP layer into neural networks. To our knowledge, this is the first time that the SPO framework coupled with an alternating differentiation method has been applied in vehicle relocation in mobile sensing. Our contributions are outlined as follows:

- *A novel end-to-end Smart Predict-then-Optimize framework for vehicle relocation problems.* We first time formulate the vehicle relocation problem using an SPO framework in deep learning. Specifically, the vehicle relocation problem is formulated as a constrained QP, where the parameters in its objectives are predicted using deep learning.
- *An alternating differentiation approach based on the ADMM for constrained optimization problems in large-scale networks.* A novel unrolling approach is proposed to obtain the gradients of the constrained QP iteratively for large-scale networks. Specifically, the forward pass is solved using the ADMM, and the backpropagation is achieved through alternating differentiation.
- *Two large-scale real-world experiments.* Numerical experiments on two real-world datasets across five network sizes from mid-size to large-scale are conducted. Superior matching performance and computational efficiency of the proposed SPO framework are demonstrated.

The remainder of the paper is organized as follows: Section 2 reviews the related work on the vehicle relocation problem in mobile sensing. Section 3 introduces the SPO framework in vehicle relocation, including the prediction module, the optimization module, the integrated model, and an unrolling approach based on the alternating differentiation method for the SPO framework. Section 4 presents the overall solution algorithm of the end-to-end SPO framework. Section 5 evaluates the proposed framework in two real-world scenarios and compares the method with the baseline approaches. Section 6 concludes the results and proposes future directions.

2. Literature Review

This section provides a comprehensive review of the literature related to the vehicle relocation problem in mobile sensing, the end-to-end SPO framework and its differentiation approach, and applications of the SPO framework in the field of transportation.

2.1. Vehicle relocation in mobile sensing

Pervasive crowd sensing plays an important role in intelligent transport systems, logistics and smart city applications (Ji et al. 2023a; Dai and Han 2023; Yang et al. 2025a). Given their relatively lower installation costs and extended operational duration, taxis are the most commonly utilized hosts in the VCS. However, due to heterogeneous trip patterns, the spatio-temporal distributions of these vehicles are often biased. To enhance sensing quality, numerous scholars have investigated intervention measures to effectively relocate vehicles through the development of incentivizing schemes (Fan et al. 2019; Xu et al. 2019; Chen et al. 2020) or proactive scheduling (Masutani 2015).

Existing research on solving the conventional vehicle relocation problem can be categorized into three types: the steady-state method, the model-free method, and the real-time optimization method (Lei et al. 2020; Qian et al. 2022). The steady-state method modeled the vehicle relocation system through a queuing theoretical framework and solved the optimal relocation strategy when the system reaches a steady state (Pavone et al. 2012; Sayarshad and Chow 2017; Chuah et al. 2018). A significant limitation of this approach is the inadequate capacity to manage the complexities of large-scale problems. In contrast, the model-free method, based on deep reinforcement learning (DRL), has emerged as a promising approach for its merits in addressing large-scale vehicle relocation challenges (Qian et al. 2022; Jiao et al. 2021; Shou and Di 2020; Qin et al. 2022).

Another classical approach in vehicle sensing is the real-time optimization method, which formulates the vehicle relocation problem from an optimization perspective and identifies optimal strategies through various solutions (Kek et al. 2009; Huang et al. 2018). Miao et al. (2016) and Zhang et al. (2016) formulated the vehicle scheduling and routing problem as a mixed integer program (MIP) and utilized the model predictive control (MPC) method to solve it. Recent advancements have also incorporated deep learning techniques to forecast future travel patterns, employing a two-stage PTO framework for the vehicle relocation problem (Weikl and Bogenberger 2013; Chang et al. 2022; 2023). This method first estimates the spatio-temporal vehicle patterns using deep learning, followed by solving the downstream optimization problem with the prediction results as inputs. However, a notable limitation of the PTO approach is that training the prediction model solely based on prediction error can result in suboptimal decision-making, as opposed to directly minimizing the decision error (Agrawal et al. 2019; Elmachtoub and Grigas 2022). Even with optimal predictions yielding minimal prediction errors, the resulting optimization may not be optimal. Our study extends the real-time PTO framework by integrating optimization directly within the prediction process through a deep learning architecture.

Table 1 summarizes related works of the vehicle relocation problem in mobile sensing and their fundamental settings. In a nutshell, our work differs from the previous research in three key aspects. First, unlike most existing literature that fully controls vehicles, we classify vehicles into dedicated and non-dedicated types, allowing the non-dedicated ones to operate freely. Second, to our knowledge, we are the first to apply the SPO framework to vehicle relocation in mobile sensing and validate it on large-scale networks. Third, we evaluate the framework across multiple target distributions, an aspect rarely explored in previous studies.

2.2. The end-to-end SPO framework and its differentiation methods

The SPO framework is an emerging paradigm in the data-driven optimization field that can leverage deep learning tools with potential applications in inventory management and electric grid scheduling etc. (Elmachtoub

Table 1: The related work of vehicle relocation in mobile sensing.

References	Fleet type	Control ratio	Target distribution	Solution approach
Miao et al. (2016)	Taxi	Fully	Reality	MPC
Xu et al. (2019)	Taxi	Partially	U, G, GM	MPC
Wang et al. (2019)	Not specific	Partially	Random	PTO
Chen et al. (2020)	Taxi	Partially	Balanced	DRL
Fan et al. (2021)	Taxi, DV	Fully	N,U	OPT
Jiao et al. (2021)	Ride-hailing	Fully	-	DRL
Xu et al. (2021)	UAV	Fully	R	OPT
Qian et al. (2022)	Taxi	Fully	U	DRL
Chang et al. (2023)	Bike	Fully	Predicted	PTO
Jiang et al. (2023)	Taxi	Fully	U	OPT
Ours	Taxi	Partially	U, G, GM	SPO

U: Uniform distribution; G: Gaussian distribution; GM: Gaussian mixture distribution; N: Normal distribution; OPT: optimization without prediction.

and Grigas 2022; Donti et al. 2017). Unlike traditional neural networks that utilize explicit formulations, the SPO framework integrates an optimization module as an implicit differentiation layer. This approach maps the inputs of the implicit functions within the optimization layer to optimal solutions. Consequently, training the optimization layer is not easy due to the absence of explicit closed-form derivatives.

To compute the gradients of the optimization layer, two principal methods are employed: the implicit differentiation method and the explicit unrolling approach. The implicit method involves calculating the first-order derivatives of the optimal solution concerning the parameters, typically by directly differentiating the KKT conditions (Geng et al. 2021a;b). Tools such as CVXPY (Diamond and Boyd 2016), Cvxpylayer (Agrawal et al. 2019), PyEPO (Tang and Khalil 2022), and OptNet (Amos and Kolter) facilitate this process for various problem types, including linear programming (LP), integer programming (IP) (Ferber et al. 2020), and QP. However, implicit differentiation methods necessitate extensive computation of the Jacobian matrix, rendering them unsuitable for large-scale network problems (Sun et al. 2023). An alternative approach is the alternating approach, which applies an iterative first-order gradient method. Existing research on alternating approaches is limited. A notable contribution is by Sun et al. (2023), which proposed an alternating differentiation method based on the ADMM to differentiate convex optimization problems with polyhedral constraints efficiently and recursively. The alternating differentiation method decouples the optimization layers in an alternative manner, thereby reducing the dimension of the Jacobian Matrix. Li et al. (2020) proposed both implicit and explicit unrolling methods for differentiating a Variational Inequalities (VIs) layer with a fixed point method. The implicit method leverages the sensitivity of the solutions to the VIs, while the explicit approach unrolls the projection method to solve the VIs.

Prior research has explored various approaches to enable differentiable optimization within the SPO framework for integer programming problems, which are broadly categorized into three main types, including linear programming relaxations and smoothing (Zhang et al. 2024; Mandi et al. 2020), incorporating the revised algorithms such as cutting-plane (Ferber et al. 2020), and utilizing surrogate loss functions such as Perturbed Fenchel-Young Loss (PFYL) and SPO+ loss (Berthet et al.; Elmachtoub and Grigas 2022). These approaches collectively provide effective differentiable substitutes that facilitate end-to-end learning for integer optimization problems and advance the integration of integer programming into the SPO framework.

In summary, implicit differentiation methods directly obtain derivatives but entail significant computational costs for large-scale problems, making them more suitable for simpler optimization problems, such as LP with few constraints or relaxed IP (Mandi et al. 2020; Elmachtoub and Grigas 2022). Conversely, explicit differentiation methods unroll the differentiation procedure, reducing the dimension of the inverse Jacobian matrix, and can be more suitable for large-scale and complex convex optimization problems, such as QP.

2.3. Applications of the SPO framework

Recent studies have increasingly investigated the application of the end-to-end SPO framework to traffic operations and management (Yan et al. 2021; Lin et al. 2024; Yang et al. 2025c;b), in the fields of maritime transportation (Yan et al. 2020; Tian et al. 2023; Yan et al. 2023; Yang et al. 2024) and logistics (Liu et al. 2021; Qi et al. 2023). Yan et al. (2020) developed a semi-SPO framework for efficient ship inspections by utilizing a surrogate mean squared difference in overestimating the number of deficiencies, rather than minimizing the mean squared error. This approach involved first predicting deficiency numbers for each inspector per ship and then constructing an integer optimization model to allocate inspectors to deficient ships. It is noteworthy that in their semi-SPO framework, prediction and optimization remain distinct processes. Similarly, Yang et al. (2024) follows the SPO criterion and proposes a task-specific metric named cumulative detected deficiency number (CDDN) to evaluate the efficiency and effectiveness of port state control inspection. However, the above studies focus on proposing novel SPO metrics rather than embedding an optimization layer in the neural network.

Although the SPO framework has been applied to various fields (Yan et al. 2020; Liu et al. 2021; Yan et al. 2021; Tian et al. 2023; Yan et al. 2023; Qi et al. 2023; Yang et al. 2024; Zhang et al. 2024), its applications in vehicle relocation are overlooked. A closely related study of embedding optimization layer to ours is by Zhang et al. (2024), which applied the end-to-end SPO framework to express pickup and delivery systems. They constructed a Graph Convolutional Network (GCN)-based model to predict order quantities and formulated a K-means clustering model to determine the optimal Areas of Interest (AOI) assignment for couriers. They utilized implicit differentiation by directly differentiating the KKT conditions to enable backpropagation within the neural network.

Table 2: Comparison of the related work in the SPO framework in transportation within the deep learning architecture.

References	Optimization layer	Constraints	Differentiation method	Dimension	Scalability
Zhang et al. (2024)	LP	✓	Implicit by KKT	35, 100 2,025 (45 × 45), 4,624 (68 × 68),	Small
Ours	QP	✓	Alternating differentiation	3,600 (60 × 60) 6,400 (80 × 80) 10,000 (100 × 100)	Large

Table 2 presents a comparison of studies employing the SPO framework in transportation within the deep learning architecture. Notably, compared to the study by Zhang et al. (2024), our work demonstrates increased complexity in the formulation of the QP layer, characterized by a higher dimensionality of decision variables, and distinctively employs the alternating differentiation method to address large-scale problems.

3. The SPO framework

In this section, we first present a comprehensive overview of the SPO framework, detailing its core components: (1) the prediction module for demand estimation, (2) the optimization module for decision-making, and (3) the backpropagation mechanism that enables end-to-end learning. Second, we demonstrate the application of the proposed SPO framework to vehicle relocation problems in VCS, providing complete mathematical formulations for each sub-module and the integrated SPO framework. Finally, we develop the alternating differentiation method and derive specialized solution schemes to efficiently solve the relocation problem while maintaining gradient flow for backward propagation.

3.1. Overview of the SPO framework

Figure 2 presents the end-to-end SPO framework, which consists of two essential modules: a prediction module and an optimization module. The prediction module uses a machine learning model to generate parameter estimates, denoted as \hat{x} , which configure the optimization module by serving as input parameters θ . The optimization module, which is formulated as a quadratic program with objective function $\frac{1}{2}y^T P y + q(\theta)^T y$ and linear constraints $Gx \leq h$, yields the optimal solution y^* . Note that $x, \theta, y, P, q \in \mathbb{R}^n$, $G, h \in \mathbb{R}^m$. Following optimization, an optional

aggregation module performs post-processing computations on y (e.g., summation or averaging) to produce the final decision variable D , which is evaluated against the target D^* through a loss function \mathcal{L} . The SPO framework aims to minimize the discrepancy between decision D and target D^* , ensuring alignment with decision-making objectives.

Specifically, the optimization module incorporates an alternating differentiation algorithm designed to facilitate gradient flow through the module. The algorithm employs the ADMM to compute the optimal solution y^* while simultaneously updating slack variables s and dual variables μ . The alternating differentiation procedure iterates in a total of K layers, with each layer executing alternating updates until the solution converges to the optimal value. During the backward pass, gradients propagate through the optimal values involving primal variables y , slack variables s and dual variables μ , to update the model parameters θ . This complete gradient flow enables end-to-end learning of the entire SPO framework with alternating differentiation methods.

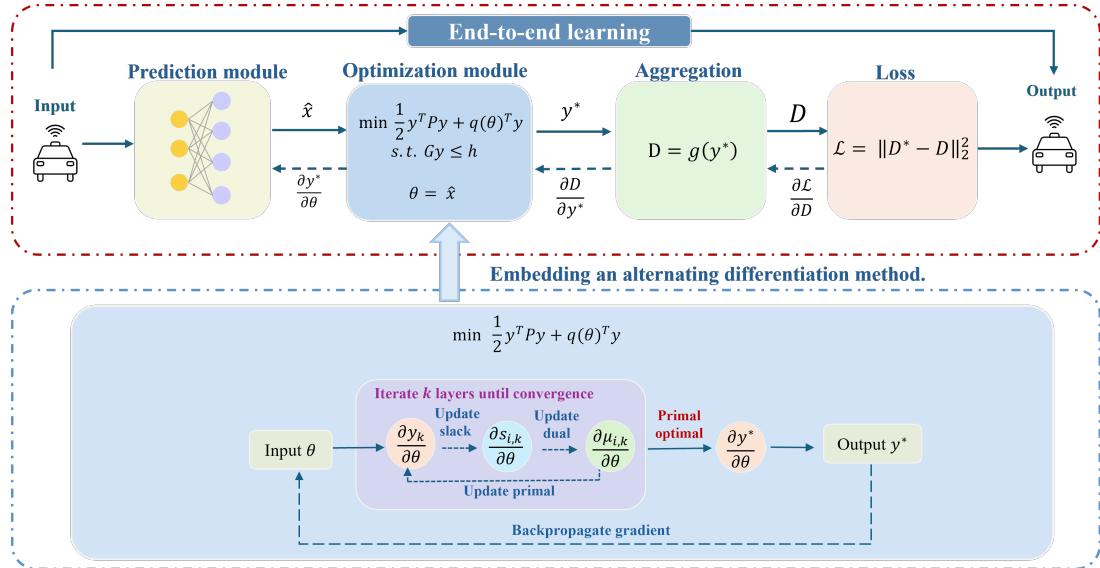


Figure 2: The end-to-end SPO framework of the vehicle sensing problem.

We then introduce the application of the SPO framework to the vehicle relocation problems in crowd sensing. As previously discussed, DVs in the VCS system adhere to designated routes under the direction of the dispatching center, whereas NDVs independently select their cruising areas and pathways. Consequently, when the dispatching center receives sensing requests from the upstream platform, it lacks prior knowledge regarding the future spatial distribution of NDVs, despite their considerable impact on the overall vehicle distribution. To devise an effective relocation strategy for DVs, it is crucial to predict the distribution of NDVs in advance. This predicted demand is then incorporated into the optimization module as a parameter.

3.2. Vehicle demand prediction model for NDVs

This section first introduces the spatio-temporal neural network, then presents the design of the prediction model for NDVs in the SPO framework.

3.2.1. Spatio-temporal vehicle movement prediction

The vehicle demand varies dynamically over time and space. A common approach to manage this variability is discretizing the study area and time length into small intervals for easy implementation of specialized spatio-temporal deep learning models (Ke et al. 2021).

In this paper, the vehicle prediction problem is modeled on a spatio-temporal scale. For the spatial dimension, the study area is partitioned into various regular hexagon grids based on geographical information (Ke et al. 2018). List of notations and dimensions for variables are provided in Appendix A (See Table A1 and Table A2). For the temporal dimension, each day is uniformly divided into equal time intervals. Consider a spatial network

$\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where \mathcal{V} denotes the set of vertices consisting of the set of origin grids I and the set of destination grids J , i.e., $\mathcal{V} = \mathcal{I} \cup \mathcal{J}$, such that $|\mathcal{I}| = |\mathcal{J}| = N$. The set of edges \mathcal{E} connects all adjacent grids.

Let V denote the vehicle class, with each class v defined as $v = c, f, a$, where c represents controllable DVs, f represents free NDVs, and a encompasses all vehicles. The set of time intervals is denoted as T , with each interval of length δ for each time interval $\tau \in T$. For each time interval τ , the spatial distribution of each vehicle type v is defined as $\mathbf{D}_v^\tau \in \mathbb{R}^N$. To be precise, $\mathbf{D}_v^\tau = [D_{v,1}^\tau, \dots, D_{v,i}^\tau, \dots, D_{v,N}^\tau]'$ is the transpose of a matrix, where $D_{v,i}^\tau$ represents the vehicle demand of vehicle type v in grid i at time τ , and $'$ is the transpose of a matrix. Then the matching distribution of all vehicles at time τ is represented by $\mathbf{D}_a^\tau = \mathbf{D}_f^\tau + \mathbf{D}_c^\tau$.

Based on the above notations, the short-term demand prediction model of NDVs is defined as follows: Given the historical observations of NDVs over the past m time intervals $[D_f^{\tau-m+1}, \dots, D_f^\tau]$, the goal is to predict the spatial demand $\mathbf{D}_f^{\tau+1}$ for the next time interval $\tau + 1$.

3.2.2. TGCN-based prediction model for NDVs

The Temporal Graph Convolutional Network (TGCN) (Zhao et al. 2019), which integrates the Graph Convolutional Network (GCN) and the Gated Recurrent Unit (GRU), is capable of concurrently capturing the inherent spatial dependencies of vehicle demand in adjacent regions and the temporal dynamics. Accordingly, we develop a two-layer prediction model based on the TGCN architecture to forecast the real-time demand of NDVs. The model incorporates a Rectified Linear Unit (ReLU) activation function and is defined as follows:

$$\hat{D}_{f,i}^{\tau+1} = \text{ReLU}(\text{TGCN}(\text{ReLU}(\text{TGCN}(D_{f,i}^{\leq \tau}, A_{f,i}^{\leq \tau}, \hat{H}_{f,i}^{\leq \tau})))), \quad (1)$$

where the $\hat{D}_{f,i}^{\tau+1}$ is the predicted demand of NDVs at time $\tau + 1$, which is also the output of the prediction module, $A_{f,i}^{\leq \tau}$ is the adjacency information, $\hat{H}_{f,i}^{\leq \tau}$ is the hidden information. The hat symbol, $\hat{\cdot}$, indicates the variable is an estimator for the true(unknown) variable. $\text{ReLU}(\cdot) = \max(0, \cdot)$.

We define the prediction loss in the prediction module in Eq. (2):

$$\mathcal{L}_1 = \|\mathbf{D}_f^{\tau+1} - \hat{\mathbf{D}}_f^{\tau+1}\|_2^2, \quad (2)$$

where $\mathbf{D}_f^{\tau+1}$ is the actual distribution of the NDVs at time $\tau + 1$.

Note that any spatio-temporal prediction model can be used in the SPO framework, here we employ the classical TGCN model as an example.

3.3. Vehicle relocation model for DVs

This section initially discusses the variable relationship between prediction and optimization modules, then presents the formulation of the vehicle relocation problem and its vectorized representation.

3.3.1. Connection between prediction and optimization variables

Before proposing the optimization model, we first illustrate the connection among the various variables between prediction and optimization modules, as shown in Figure 3. The output of the prediction layer $\hat{\mathbf{D}}_f^{\tau+1}$ is input in the optimization layer as a known parameter.

After predicting the movements of NDVs, we can easily obtain the required (estimated) distribution for DVs by a simple subtraction of the given target distribution $\mathbf{T}_a^{\tau+1}$ and the predicted distribution $\hat{\mathbf{D}}_f^{\tau+1}$ in Eq. (3):

$$\hat{\mathbf{D}}_c^{\tau+1} = \mathbf{T}_a^{\tau+1} - \hat{\mathbf{D}}_f^{\tau+1}, \quad (3)$$

where $\mathbf{T}_a^{\tau+1}$ is the target distribution for all vehicles at time $\tau + 1$.

3.3.2. Formulation of the vehicle relocation model

This section presents the QP formulation of the relocation model for DVs within the optimization module. We regard the original distribution of DVs \mathbf{D}_c^τ as *supply* and the required distribution of DVs $\hat{\mathbf{D}}_c^{\tau+1}$ as *demand* for simplicity. The goal in the subsequent optimization module is to develop an optimal vehicle relocation strategy that satisfies the future *demand* with the current *supply* by considering the total incentive cost budget, accessibility constraints, and supply limitations.

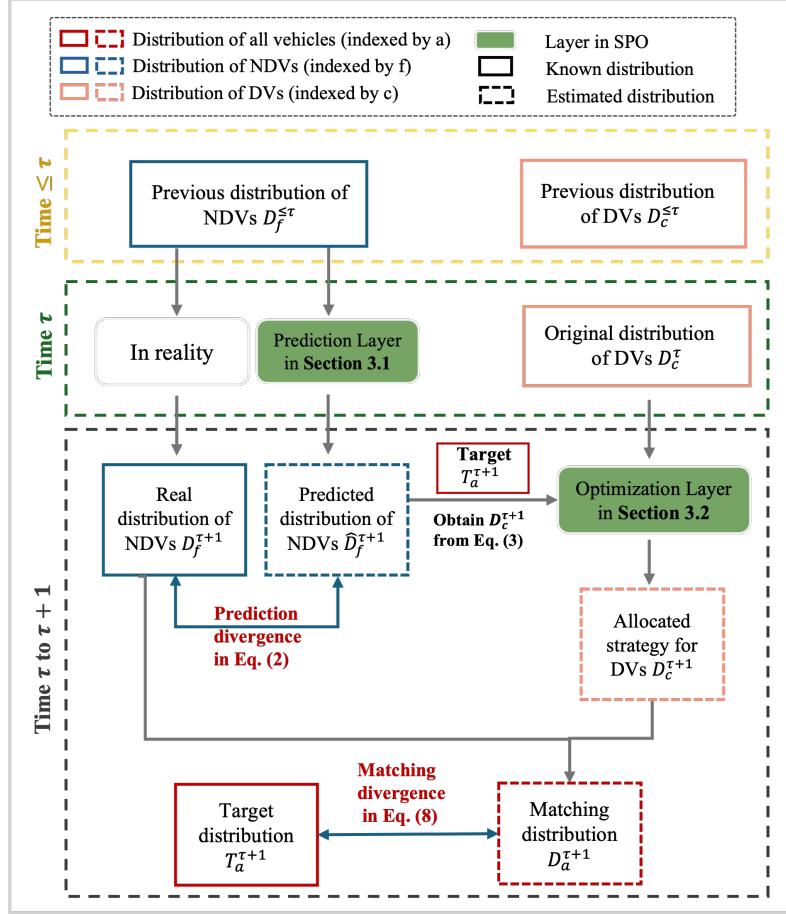


Figure 3: The variable relationship in different modules.

Denote $x_{c,ij}^\tau \in \mathbb{R}^{N \times N}$ as the dedicated vehicle flow from grid i to grid j during τ to $\tau + 1$. Then the vehicle relocation problem is formulated in Eq. (4):

$$\min_{x_{c,ij}^\tau} Z_0 = \frac{1}{2} \sum_j \left\| \sum_i x_{c,ij}^\tau - \hat{\mathbf{D}}_{c,j}^{\tau+1} \right\|_2^2, \quad (4a)$$

$$\text{s.t.} \quad \sum_j x_{c,ij}^\tau \leq \mathbf{D}_{c,i}^\tau, \forall i \in I, \tau \in T, \quad (4b)$$

$$(m_{c,ij}^\tau - \delta)x_{c,ij}^\tau \leq 0, \forall i \in I, j \in J, \tau \in T, \quad (4c)$$

$$\sum_i \sum_j w_{c,ij}^\tau x_{c,ij}^\tau \leq R^\tau, \forall \tau \in T, \quad (4d)$$

$$x_{c,ij}^\tau \geq 0, \forall i \in I, j \in J, \tau \in T, \quad (4e)$$

where $m_{c,ij}^\tau \in \mathbb{R}^{N \times N}$ is the travel time from grid i to j at time τ ; $w_{c,ij}^\tau \in \mathbb{R}^{N \times N}$ is the incentive cost from grid i to j at time τ ; and R is the maximum total incentive budget. The objective in Eq. (4a) aims to satisfy the *demand* in all regions. The constraints of the problem include supply constraint in Eq. (4b), time constraint in Eq. (4c), budget constraint in Eq. (4d) and non-negative constraint in Eq. (4e). The supply constraint in Eq. (4b) ensures that the total number of DVs at time $\tau + 1$ does not exceed the available supply. The time constraint in Eq. (4c) checks whether the actual travel time exceeds the desired relocation time δ . The budget constraint in Eq. (4d) ensures that the incentive cost of all DVs should not surpass the total budget R .

3.3.3. Vectorizing the relocation formulation

To seamlessly integrate the optimization module into the neural network, all two-dimensional variables and parameters will be flattened and vectored into one dimension. Specifically, $x_{c,ij}^\tau \in \mathbb{R}^{N \times N}$, $w_{c,ij}^\tau \in \mathbb{R}^{N \times N}$ are converted to $\mathbf{y}^\tau \in \mathbb{R}^{N^2}$, $\mathbf{C}^\tau \in \mathbb{R}^{N^2}$, respectively. For simplicity, we omit the superscript τ , therefore $\mathbf{y}^\tau \in \mathbb{R}^{N^2}$, $\mathbf{C}^\tau \in \mathbb{R}^{N^2}$ is replaced by \mathbf{y} , \mathbf{C} , as is shown below:

$$\mathbf{y} = (y_1, y_2, \dots, y_N | y_{N+1}, y_{N+2}, \dots, y_{2N} | y_{N^2-N+1}, y_{N^2-N+2}, \dots, y_{N^2})^T \quad (5)$$

$$= (x_{11}^\tau, x_{12}^\tau, \dots, x_{1N}^\tau | x_{21}^\tau, x_{22}^\tau, \dots, x_{2N}^\tau | x_{N1}^\tau, x_{N2}^\tau, \dots, x_{NN}^\tau)^T \in \mathbb{R}^{N^2}, \quad (6)$$

$$\mathbf{C} = (w_{11}^\tau, w_{12}^\tau, \dots, w_{1N}^\tau | w_{21}^\tau, w_{22}^\tau, \dots, w_{2N}^\tau | w_{N1}^\tau, w_{N2}^\tau, \dots, w_{NN}^\tau)^T. \quad (7)$$

The vehicle relocation problem in Eq. (4) is then reformulated into a general standardized quadratic form, as presented in Eq. (8).

$$\begin{aligned} \min_{\mathbf{y}} \quad & Z_1 = \frac{1}{2} \mathbf{y}' \mathbf{P} \mathbf{y} + \mathbf{q}' \mathbf{y}, \\ \text{s.t.} \quad & \mathbf{G}_1 \mathbf{y} \leq \mathbf{h}_1, \\ & \mathbf{G}_2 \mathbf{y} \leq \mathbf{h}_2, \\ & \mathbf{G}_3 \mathbf{y} \leq \mathbf{h}_3, \\ & \mathbf{G}_4 \mathbf{y} \leq \mathbf{h}_4, \end{aligned} \quad (8)$$

where $Z_1 = Z_0 - \frac{\hat{\mathbf{D}}_c^{\tau+1} \hat{\mathbf{D}}_c^{\tau+1}}{2}$, $\mathbf{P} = \mathbf{A}' \mathbf{A}$, $\mathbf{q} = -\mathbf{A}' \hat{\mathbf{D}}_c^{\tau+1}$, $\mathbf{G}_1 = \mathbf{B}$, $\mathbf{G}_2 = \text{diag}(m_{11}^\tau - \delta, m_{12}^\tau - \delta, \dots, m_{NN}^\tau - \delta) \in \mathbb{R}^{N^2}$, $\mathbf{G}_3 = \mathbf{C}'$, $\mathbf{G}_4 = -\mathbf{I}_{N^2}$, $\mathbf{h}_1 = \mathbf{D}_c^\tau$, $\mathbf{h}_2 = \mathbf{0}$, $\mathbf{h}_3 = \mathbf{R}$, $\mathbf{h}_4 = \mathbf{0}$. $\text{diag}(\cdot)$ is the diagonal matrix, \mathbf{I}_{N^2} is the Identity matrix in dimension N^2 . \mathbf{A} and \mathbf{B} are sparse matrices, and,

$$\mathbf{A} = \left(\begin{array}{ccc|ccc|c} 1 & 0 & \cdots & 0 & 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & 0 & 0 & \cdots & 1 \end{array} \right)_{N \times N^2}, \mathbf{B} = \left(\begin{array}{ccc|ccc|c} 1 & 1 & \cdots & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & 1 & 1 & \cdots & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \end{array} \right)_{N \times N^2} \quad .$$

The optimization problem in the vectorized form in Eq. (8), which includes four unequal constraints and a one-dimensional decision variable (tensor), is equivalent to Eq. (4). The three key properties of Eq. (8) are as follows:

- The parameter from the prediction layer $\hat{\mathbf{D}}_f^{\tau+1}$, is embedded exclusively in the objective function's term q and does not appear in any of the constraint terms.
- The vectorized form simplifies the practical implementation of embedding the optimization layer within the neural network, allowing for a seamless connection with the prediction layer and the construction of the entire framework.
- By proposing a generalized form with parameters represented as G_n and h_n , we create a more straightforward representation of the optimization problem, facilitating the derivation of the solution algorithm later on.

3.4. Aggregation

This section integrates the actual distribution of DVs $\mathbf{D}_c^{\tau+1}$ and NDVs $\mathbf{D}_f^{\tau+1}$ obtained from optimization module to derive the holistic matching distribution of all vehicles $\mathbf{D}_a^{\tau+1}$ in the subsequent time interval $\tau + 1$. For DVs, by solving the optimization problem in Eq. (8), we can generate the optimal strategy for DVs. In particular, the actual allocated distribution of DVs $\mathbf{D}_c^{\tau+1}$ is then determined by aggregating the number of DVs that arrive in grids at time $\tau + 1$:

$$\mathbf{D}_c^{\tau+1} = \mathbf{A} \mathbf{y}. \quad (9)$$

For NDVs, these vehicles will reach their destinations by time $\tau + 1$, enabling the direct determination of their actual distribution, denoted as $\mathbf{D}_f^{\tau+1}$. Note that within the optimization module, we employ a surrogate distribution—specifically, the predicted distribution of NDVs $\hat{\mathbf{D}}_f^{\tau+1}$ at time $\tau + 1$ to approximate the actual distribution $\mathbf{D}_f^{\tau+1}$. This predicted distribution serves as a proxy for $\mathbf{D}_f^{\tau+1}$ and plays a critical role in formulating the relocation plan.

Consequently, the actual holistic matching distribution for all vehicles can be straightforwardly represented by Eq. (10):

$$\mathbf{D}_c^{\tau+1} + \mathbf{D}_f^{\tau+1} = \mathbf{D}_a^{\tau+1}. \quad (10)$$

3.5. The integrated SPO model

In the end-to-end SPO framework for the vehicle relocation problem in VCS, the primary objective is to enhance the sensing accuracy of the sensors. Since the sensors are installed on vehicles, the sensing distribution is effectively represented by the matching distribution of the vehicles. Consequently, the ultimate goal is to minimize the matching divergence between the matching distribution $\mathbf{D}_a^{\tau+1}$ and the target distribution $\mathbf{T}_a^{\tau+1}$ across all vehicles. The matching loss is formally defined as follows:

$$\mathcal{L}_2 = \|\mathbf{T}_a^{\tau+1} - \mathbf{D}_a^{\tau+1}\|_2^2. \quad (11)$$

Compared to \mathcal{L}_1 in Eq. (2), which is a mid-term prediction loss of NDVs, \mathcal{L}_2 in Eq. (11) is the matching loss of all vehicles including DVs and NDVs.

From Eq. (10), it is evident that both the prediction error and relocation error contribute to the overall matching divergence. Therefore, the loss function for the SPO framework is formulated by combining the prediction loss in Eq. (2) and the matching loss in Eq. (11):

$$\mathcal{L}_{SPO} = w_1 \mathcal{L}_1 + w_2 \mathcal{L}_2, \quad (12)$$

where w_1 is the weight for prediction loss for the NDVs, and w_2 is the weight for the matching loss for all vehicles.

As illustrated in Figure 2, the vehicle relocation problem defined in Eq. (8) is embedded in the optimization module and connected with the preceding prediction module in Eq. (1) via gradients. This integration allows the prediction module to inform the optimization process through the backpropagation of gradients, facilitating end-to-end training. Consequently, the SPO framework can be formally presented as follows:

$$\begin{aligned} \min & \quad Eq. (12) \\ \text{s.t.} & \quad Eq. (1) - \text{Prediction}, \\ & \quad Eq. (8) - \text{Vehicle relocation problem}, \\ & \quad Eq. (9, 10) - \text{Aggregation}. \end{aligned} \quad (13)$$

Eq. (13) has two crucial properties as summarized below:

- This is a sequential decision-making problem where the solution process progresses from addressing the lower-level constraints to solving the upper-level minimization function. It begins with prediction in Eq. (1), followed by optimization in Eq. (8), and concludes with the calculation of divergence in Eq. (12).
- The overall nested optimization problem contains a sub-constrained optimization problem (8) within the constraint section, making it challenging to solve directly and efficiently.

Considering these properties, it is essential to develop novel solution methods to effectively address such nested-constrained optimization problems within the deep learning architecture.

3.6. Solution algorithm

In this section, we will first introduce the computational graph in the SPO framework, then propose the explicit unrolling approach of the SPO framework, including the forward pass by the ADMM, and the backward alternating differentiation method.

3.6.1. The computational graph

The computational graph illustrates how the gradients link and propagate in the neural network in the forward and backward pass. Therefore, we first propose the computational graph in the SPO framework, as is shown in Figure 4. The computation of gradients is essential to enable backpropagation (BP) within the neural network. illustrates the forward and backward pass process in the SPO framework.

To solve the SPO framework as formulated in Eq. (13), we derive the gradients of the loss function \mathcal{L}_{SPO} w.r.t. the weight \mathbf{w}_p in the prediction module, as is shown in Eq. (14):

$$\frac{\partial \mathcal{L}_{SPO}}{\partial \mathbf{w}_p} = \frac{\partial \mathcal{L}_{SPO}}{\partial \hat{\mathbf{D}}_f^{\tau+1}} \frac{\partial \hat{\mathbf{D}}_f^{\tau+1}}{\partial \mathbf{w}_p} + \frac{\partial \mathcal{L}_{SPO}}{\partial \mathbf{D}_a^{\tau+1}} \frac{\partial \mathcal{L}_2}{\partial \mathbf{D}_a^{\tau+1}} \frac{\partial \mathbf{y}}{\partial \mathbf{y}} \frac{\partial \hat{\mathbf{D}}_f^{\tau+1}}{\partial \mathbf{y}}. \quad (14)$$

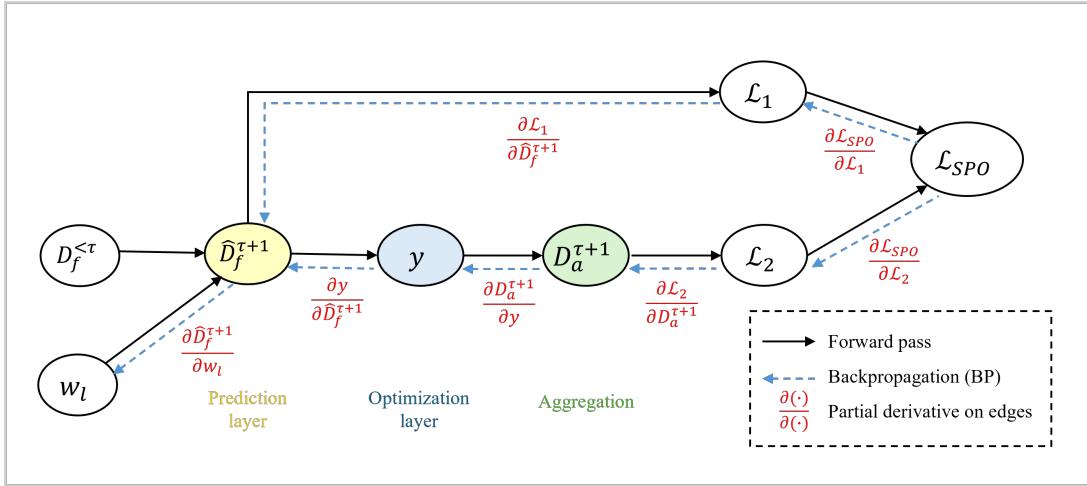


Figure 4: Illustration of the computational graph.

All the derivatives can be computed analytically and automatically by the Autograd package in PyTorch, except for the $\frac{\partial y}{\partial \hat{\mathbf{D}}_f^{\tau+1}}$ in the optimization layer. Hence, the primary challenge is to compute the gradient of the optimization layer to enable forward and backward propagation. To this end, the gradient of the optimization layers necessitates solving the internal optimization problem first and subsequently differentiating the solution with respect to the parameter provided from the preceding layer.

3.6.2. The unrolling K-layer neural network

Then we proceed to introduce how to derive the gradient of the optimal solution with respect to the predefined parameter from the prediction layer, $\frac{\partial y}{\partial \hat{\mathbf{D}}_f^{\tau+1}}$, using an unrolling approach through the alternating differentiation method. Figure 5 illustrates the unrolling alternating differentiation method and compares it with the implicit differentiation method.

In the implicit differentiation method by KKT condition in Figure 5 (a) and (c), the solution for forward is obtained directly, and the corresponding gradient for the backward pass is computed in a single layer. However, obtaining these gradients directly becomes challenging in large-scale networks due to the high computational cost associated with calculating the Jacobian matrix in the KKT condition.

To address this, we propose an alternating differentiation approach, as shown in Figure 5 (b) and (d). This approach unfolds the one-layer network into K iterations through an alternating updating procedure. In the forward pass, the problem is solved using the ADMM, where the primal, slack, and dual variables are computed iteratively over K iterations until convergence (or when the optimal solution is found). Concurrently, the gradients for these variables are computed simultaneously through analytical functions at each step. Each iteration in the forward pass is mapped to a single layer. Once convergence is achieved, these K layers are stacked together, forming a K -layer neural network. The final gradient in the optimizer for the alternating differentiation method is obtained by

intrinsically combining the gradients from all K layers. Then, in the backward pass, the final gradient $\frac{\partial \mathbf{y}}{\partial \hat{\mathbf{D}}_f^{\tau+1}}$ directly propagates at y^* in the final K^{th} layer, which inherently corresponds to passing through all iterating gradients. Essentially, the unrolled K -layer neural network is equivalent to a one-layer neural network used in the implicit differentiation method.

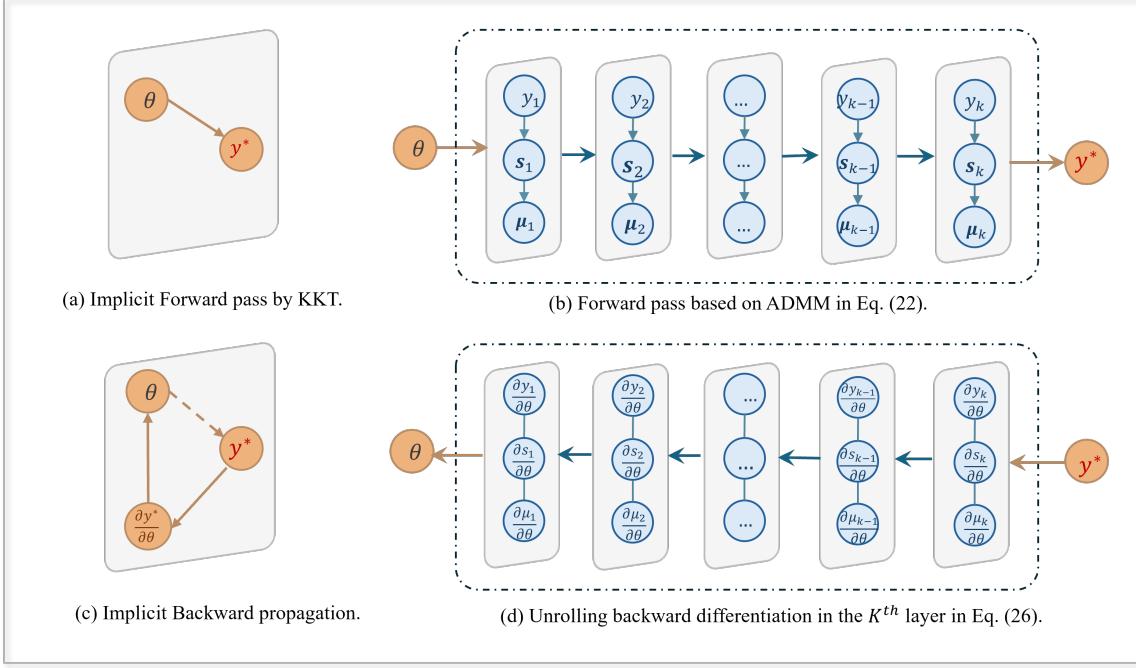


Figure 5: A comparison of the explicit and implicit differentiation method. (a) and (c) show the forward and backward pass through the implicit differentiation method by KKT, (b) and (d) depict the forward and backward pass through the alternating differentiation method.

Then we will introduce the solution algorithm for the forward pass and backward pass of the unrolled alternating differentiation method, respectively.

Forward pass based on the ADMM. The optimization problem in Eq. (8) is a constrained quadratic problem with four unequal constraints. It can be solved using the ADMM by transforming the original constrained problem into an unconstrained optimization problem. This section derives the forward pass and the optimal solution via the ADMM.

Note that the parameter from the prediction layer $\hat{\mathbf{D}}_f^{\tau+1}$ is embedded in the objective function's term $q(\hat{\mathbf{D}}_f^{\tau+1})$. Here we replace $\hat{\mathbf{D}}_f^{\tau+1}$ as θ as simplicity, and $q(\hat{\mathbf{D}}_f^{\tau+1})$ is replaced by $q(\theta)$.

Following the procedures of the ADMM, we first form the augmented Lagrange function of the problem in Eq. (8) with a quadratic penalty term as follows:

$$\begin{aligned} \max_{\mu} \min_{\mathbf{y} \geq 0, \mathbf{s} \geq 0} \mathcal{L}(\mathbf{y}, \mathbf{s}, \mu; \theta) = & \frac{1}{2} \mathbf{y}' \mathbf{P} \mathbf{y} + \mathbf{q}(\theta)' \mathbf{y} + \sum_{n=1}^4 \langle \mu_n, \mathbf{G}_n \mathbf{y} + \mathbf{s}_n - \mathbf{h}_n \rangle \\ & + \frac{\rho}{2} \left(\sum_{n=1}^4 \|\mathbf{G}_n \mathbf{y} + \mathbf{s}_n - \mathbf{h}_n\|^2 \right), \end{aligned} \quad (15)$$

where $\mathbf{s} = \{\mathbf{s}_n | n = 1, 2, 3, 4\} \geq 0$ is the set of non-negative slack variables corresponding to the four constraints in Eq. (8), $\mu = \{\mu_n | n = 1, 2, 3, 4\}$ is the set of dual variables of the four constraints, $\rho > 0$ is the penalty term, θ is the parameter from the prediction layer and $\theta = \hat{\mathbf{D}}_f^{\tau+1}$.

Then the primary, slack, and dual variables are updated alternatively based on the ADMM.

$$\begin{cases} \mathbf{y}_{k+1} = \underset{\mathbf{y}}{\operatorname{argmin}} \mathcal{L}(\mathbf{y}, \mathbf{s}_{n,k}, \mu_{n,k}; \theta) & \forall n \in \{1, 2, 3, 4\}, \end{cases} \quad (16a)$$

$$\begin{cases} \mathbf{s}_{n,k+1} = \underset{\mathbf{s}_n \geq 0}{\operatorname{argmin}} \mathcal{L}(\mathbf{y}_{k+1}, \mathbf{s}_{n,k}, \mu_{n,k}; \theta) & \forall n \in \{1, 2, 3, 4\}, \end{cases} \quad (16b)$$

$$\mu_{n,k+1} = \mu_{n,k} + \rho(\mathbf{G}_n \mathbf{y}_{k+1} + \mathbf{s}_{n,k+1} - \mathbf{h}_n) \quad \forall n \in \{1, 2, 3, 4\}. \quad (16c)$$

Then we derive the analytical updating solutions for the primal \mathbf{y}_{k+1} , slack $\mathbf{s}_{n,k+1}$, and dual variables $\mu_{n,k+1}$, respectively.

Taking the first and second-order derivative of Eq. (15) w.r.t. \mathbf{y} and we have $\nabla_{\mathbf{y}} \mathcal{L}$ and $\nabla_{\mathbf{y}}^2 \mathcal{L}$ as follows:

$$\nabla_{\mathbf{y}} \mathcal{L} = \mathbf{P}' \mathbf{y} + \mathbf{q}(\theta) + \mathbf{G}'_n \mu_n + \sum_{n=1}^4 \rho (\mathbf{G}'_n (\mathbf{G}_n \mathbf{y} + \mathbf{s}_n - \mathbf{h}_n)), \quad (17)$$

$$\nabla_{\mathbf{y}}^2 \mathcal{L}(\mathbf{y}_{k+1}) = \mathbf{P} + \sum_{n=1}^4 \rho \mathbf{G}'_n \mathbf{G}_n. \quad (18)$$

Let Eq. (17) equals 0 and \mathbf{y}_{k+1} in Eq. (16a) is therefore obtained as follows:

$$-(\mathbf{P} + \sum_{n=1}^4 \rho \mathbf{G}'_n \mathbf{G}_n) \mathbf{y}_{k+1} = (\mathbf{q}(\theta) + \sum_{n=1}^4 \rho \mathbf{G}'_n (\mathbf{s}_{n,k} - \mathbf{h}_n) + \sum_{n=1}^4 \mathbf{G}'_n \mu_{n,k}). \quad (19)$$

Let $U = -(\mathbf{P} + \sum_{n=1}^4 \rho \mathbf{G}'_n \mathbf{G}_n)'$ and U is constant. Then \mathbf{y}_{k+1} can be updated by Eq. (20):

$$\mathbf{y}_{k+1} = U(\mathbf{q}(\theta) + \sum_{n=1}^4 \rho (\mathbf{G}'_n (\mathbf{s}_{n,k} - \mathbf{h}_n) + \mathbf{G}'_n \mu_{n,k})). \quad (20)$$

Taking the first-order derivative of Eq. (15) w.r.t. $\mathbf{s}_{n,k}$, $\forall n \in \{1, 2, 3, 4\}$, the slack variables $\mathbf{s}_{n,k+1}$ in Eq. (16b) are substituted by Eq. (21) with a ReLU function:

$$\mathbf{s}_{n,k+1} = \text{ReLU}\left(-\frac{1}{\rho} \mu_{n,k} - (\mathbf{G}_n \mathbf{y}_{k+1} - \mathbf{h}_n)\right). \quad (21)$$

We finally obtain the forward solution of primal, slack, and dual variables in Eq. (22):

$$\begin{cases} \mathbf{y}_{k+1} = U(\mathbf{q}(\theta) + \sum_{n=1}^4 \rho \mathbf{G}'_n (\mathbf{s}_{n,k} - \mathbf{h}_n) + \sum_{n=1}^4 \mathbf{G}'_n \mu_{n,k}), \end{cases} \quad (22a)$$

$$\begin{cases} \mathbf{s}_{n,k+1} = \text{ReLU}\left(-\frac{1}{\rho} \mu_{n,k} - (\mathbf{G}_n \mathbf{y}_{k+1} - \mathbf{h}_n)\right) & \forall n \in \{1, 2, 3, 4\}, \end{cases} \quad (22b)$$

$$\mu_{n,k+1} = \mu_{n,k} + \rho(\mathbf{G}_n \mathbf{y}_{k+1} + \mathbf{s}_{n,k+1} - \mathbf{h}_n) \quad \forall n \in \{1, 2, 3, 4\}. \quad (22c)$$

Backward pass by the alternating differentiation method. Based on the forward solving procedures in Eq. (22), we can then present the backward differentiation algorithm, which is to compute the derivatives of the primal, slack, and dual variables with respect to predefined parameters alternatively to enable backpropagation.

In the end-to-end SPO framework, we address a real-time vehicle relocation problem in Eq. (8), where the pre-defined parameter (output) from the prediction layer is denoted as $\hat{\mathbf{D}}_f^{\tau+1}$. In this section, we regard $\hat{\mathbf{D}}_f^{\tau+1}$ as θ for simplicity, and θ is in $\mathbf{q}(\theta)$.

Applying the Implicit Function Theorem (Krantz and Parks 2002) to Eq. (17), then the derivative of the solution \mathbf{y}^{k+1} w.r.t. the parameter θ can be formulated as:

$$\frac{\partial \mathbf{y}^{k+1}}{\partial \theta} = -\nabla_{\mathbf{y}}^2 \mathcal{L}(\mathbf{y}_{k+1})' \nabla_{\mathbf{y}, \theta} \mathcal{L}(\mathbf{y}_{k+1}) = U\left(\frac{\partial \mathbf{q}(\theta)}{\partial \theta} + \sum_{n=1}^4 \rho \mathbf{G}'_n \frac{\partial \mathbf{s}_{n,k}}{\partial \theta} + \sum_{n=1}^4 \mathbf{G}'_n \frac{\partial \mu_{n,k}}{\partial \theta}\right), \quad (23)$$

where the second-order derivative $\nabla_y^2 \mathcal{L}$ is obtained in Eq. (18).

The derivative of $\mathbf{s}_{n,k+1}$ w.r.t. θ can be obtained as follows:

$$\frac{\partial \mathbf{s}_{n,k+1}}{\partial \theta} = -\frac{1}{\rho} \mathbf{sgn}(\mathbf{s}_{n,k+1}) \cdot \mathbf{1}' \odot \left(\frac{\partial \mu_{n,k}}{\partial \theta} + \rho \frac{\partial(G\mathbf{y}_{n,k+1} - \mathbf{h}_n)}{\partial \theta} \right). \quad (24)$$

The derivative of $\mu_{n,k+1}$ w.r.t. θ can be easily obtained as follows:

$$\frac{\partial \mu_{n,k+1}}{\partial \theta} = \frac{\partial \mu_{n,k}}{\partial \theta} + \rho \frac{\partial(G\mathbf{y}_{n,k+1} + \mathbf{s}_{n,k+1} - \mathbf{h}_n)}{\partial \theta}. \quad (25)$$

We finally obtain the explicit differentiation function of primal, slack and dual variables as summarized in Eq. (26):

$$\begin{cases} \frac{\partial \mathbf{y}_{k+1}}{\partial \theta} = U \left(\frac{\partial \mathbf{q}(\theta)}{\partial \theta} + \sum_{n=1}^4 \rho \mathbf{G}'_n \frac{\partial \mathbf{s}_{n,k}}{\partial \theta} + \sum_{n=1}^4 \mathbf{G}'_n \frac{\partial \mu_{n,k}}{\partial \theta} \right), \end{cases} \quad (26a)$$

$$\begin{cases} \frac{\partial \mathbf{s}_{n,k+1}}{\partial \theta} = -\frac{1}{\rho} \mathbf{sgn}(\mathbf{s}_{n,k+1}) \cdot \mathbf{1}' \odot \left(\frac{\partial \mu_{n,k}}{\partial \theta} + \rho \frac{\partial(G_n \mathbf{y}_{k+1} - \mathbf{h}_n)}{\partial \theta} \right), \end{cases} \quad (26b)$$

$$\begin{cases} \frac{\partial \mu_{n,k+1}}{\partial \theta} = \frac{\partial \mu_{n,k}}{\partial \theta} + \rho \frac{\partial(G_n \mathbf{y}_{k+1} + \mathbf{s}_{n,k+1} - \mathbf{h}_n)}{\partial \theta}, \end{cases} \quad (26c)$$

where $\frac{\partial q}{\partial \theta} = \mathbf{A}'$ and \odot is the Hadamard product. In particular, Eq. (26) is utilized to perform the backward propagation on the computational graph. Note the above forward pass by the ADMM in Eq. (22) and the backward differentiation in Eq. (26) is suitable for any quadratic problems with unequal constraints in the form in Eq. (8).

4. Solution algorithms

The overall solution algorithm of the proposed SPO framework is presented in Algorithm 1, which can be performed in batch form.

5. Numerical experiments

In this section, we evaluate the proposed end-to-end SPO framework embedded with an unrolling approach using real-world taxi data from two distinct locations: the Kowloon district in Hong Kong SAR and Chengdu City, China. First, we outline the experiment settings of the two cases and then present the experimental results to verify the effectiveness and efficiency of the proposed framework in each case. For both cases, we compare the proposed SPO-A across mid-size and large-scale datasets to assess scalability against various baseline models, including the SPO-C framework and the conventional two-stage PTO framework. In Case A, we perform ablation studies and sensitivity analyses to examine the influence of key parameters on the proposed SPO-A framework. In Case B, we extend our evaluation to larger-scale parallel experiments, further validating the robustness and applicability of the proposed framework in a larger-scale urban setting. All the experiments are conducted on a desktop with Intel Core i7-13700K CPU 3.40 GHz \times 32G RAM, 500 GB SSD, GeForce RTX 3090 Ti GPU.

5.1. General settings

In this section, we introduce the characteristics of the research areas, hyper-parameter configuration, baseline models, target sensing distributions, and evaluation metrics for the two cases.

5.1.1. Experimental setup

Table 3 summarizes the detailed experimental setups for the two cases. Both experiments are conducted using real-world road networks and taxi datasets. Compared to Case A, Case B encompasses a broader geographical region and a larger network scale. Additionally, the average taxi demand per grid per time interval in Case B is roughly three times higher than in Case A. In both cases, the ratio of DVs to NDVs varies from 20% to 80% with an increment of 10%. While the prediction module in Case A only incorporates historical taxi demand as input features, Case B employs a more comprehensive feature set, including historical demand, weather conditions, and the number of Points of Interest (POIs).

Algorithm 1: Solving procedure for the end-to-end SPO framework.

Data: Input the historical demand $\mathbf{D}_f^{\leq \tau}$, the target distribution $\mathbf{T}_a^{\tau+1}$, the control ratio γ , the total incentive cost budget R , the actual travel time matrix C , the penalty term ρ , the convergence threshold ξ .

Result: Output the relocation strategy \mathbf{y}^* , the matching distribution \mathbf{D}_a .

```

1 Initialize Max epoch  $N_e$ , prediction weight  $\mathbf{w}_1$ , matching weight  $\mathbf{w}_2$  ;
2 for Epoch  $e = 1, \dots, N_e$  do
3   Input the  $\mathbf{D}_f^{\leq \tau}$  into prediction module ;
4   Predict the distribution of NDVs  $\hat{\mathbf{D}}_f^{\tau+1}$  in Eq. (1) ;
5   Compute prediction loss  $\mathcal{L}_1$  in Eq. (2) Input  $\hat{\mathbf{D}}_f^{\tau+1}$  and  $\mathbf{D}_c^\tau$  into optimization module in Eq. (8) ;
6   Solve Optimization() ;
7   Function Optimization():
8     Input the parameter  $\theta$  ;
9     Initialize  $k \leftarrow 0$  ;
10    Initialize variables and derivatives  $\mathbf{y}_k, s_{i,k}, \mu_{i,k}, \frac{\partial \mathbf{y}_k}{\partial \theta}, \frac{\partial s_{i,k}}{\partial \theta}, \frac{\partial \mu_{i,k}}{\partial \theta}$  ;
11    Initialize parameters  $\mathbf{G}_1, \mathbf{G}_2, \mathbf{G}_3, \mathbf{G}_4, \mathbf{h}_1, \mathbf{h}_2, \mathbf{h}_3, \mathbf{h}_4, \mathbf{A}, \mathbf{B}$ , convergence threshold  $\xi$  ;
12    while  $|\hat{Z}_{k+1} - Z_k| \geq \xi$  do
13      Primal update  $\mathbf{y}_k$  by Eq. (22a),  $\frac{\partial \mathbf{y}_k}{\partial \theta}$  by Eq. (26a) ;
14      Slack update  $s_{n,k}$  by Eq. (22b),  $\frac{\partial s_{n,k}}{\partial \theta}$  by Eq. (26b) ;
15      Dual update  $\mu_{n,k}$  by Eq. (22c),  $\frac{\partial \mu_{n,k}}{\partial \theta}$  by Eq. (26c) ;
16      Compute  $\hat{Z}_k$  ;
17       $k \leftarrow k + 1$  ;
18    end
19    return the relocation strategy  $\mathbf{y}^*$  ;
20  Aggregate  $\mathbf{y}^*$  and obtain distribution for DVs  $\hat{\mathbf{D}}_c^{\tau+1}$  in Eq. (9) ;
21  Obtain matching distribution  $\hat{\mathbf{D}}_a^{\tau+1}$  in Eq. (10) ;
22  Compute the SPO loss  $\mathcal{L}_2$  in Eq. (12);
23  Back propagate and update weights  $w_p$  in prediction model;
24 end

```

5.1.2. Hyper-parameter setting

The hyper-parameters in the SPO framework of Case A and Case B are summarized in Table 4. We utilize the same Adagrad optimizer, weight decay, look-back window, and time interval for both cases. However, since Case B involves a larger-scale dataset than Case A and requires significantly longer training times, we adopt a higher learning rate, a more relaxed convergence threshold, and a stronger penalty term to accelerate convergence while balancing accuracy and computational efficiency. Additionally, we use a small batch size of 16 for Case B to avoid memory overflow during the computation of the large Jacobian. All hyper-parameters in both the prediction and optimization modules of the two cases are carefully fine-tuned and selected through cross-validation to ensure optimal performance. The setting of prediction modules is provided in Appendix B (See Table B1). The cross-validation procedure of the penalty term is supplemented in Section 5.2.4.4.

5.1.3. Dynamic weighting strategies

This section summarizes the tuning techniques applied in the proposed SPO framework to prevent overfitting. While standard methods such as dropout and learning rate scheduling are employed, we additionally adopt two innovative techniques specific to the SPO framework: the dynamic weighting mechanism applicable to both SPO-A and SPO-C frameworks, and optimization layer regularization via the penalty term to the SPO-A.

- *Dynamic weight adjustment and warm initialization.* The final SPO loss function provided in Equation 12 combines both the prediction MSE and optimization MSE. To harmonize their convergence, we implement a phased weighting strategy: 1) Initial Phase (Epochs 1–100): Prediction loss weight is amplified 20–50

Table 3: Experimental setup in Case A and Case B.

	Case A	Case B
Area size	50.11 km^2	73.73 km^2
Total data sample	2,075,864	9,985,238
Average demand in each grid	35	126
Network scale	$45 \times 45, 68 \times 68$	$60 \times 60, 80 \times 80, 100 \times 100$
Prediction features	Historical demand	Historical demand, weather, POI

Table 4: Configuration of the SPO framework in Case A and Case B.

Hyper-parameters	Case A	Case B
Optimizer	Adagrad	Adagrad
Batch size	64	16
Learning rate	5e-3	1e-2
Weight decay	1e-4	1e-4
Look-back window	12	12
Time interval	15 min	15 min
Convergence threshold	5e-2	1
Penalty term in optimization	2.0	5.0
Total travel budget for optimization	8,000/10,000/12,000/15,000	30,000
Target distributions	U, G, GM ^a	U, G, GM ^b

^{a,b} The parameters in U, G, and GM for Case A and Case B are different.

times over optimization loss to stabilize early training via warm initialization. 2) Transition Phase (Next 100 Epochs): Prediction weight is linearly reduced to achieve a 1:1 balance, ensuring equitable contributions from both prediction and optimization. 3) Final Phase (Last few epochs): The ratio is either maintained at 1:1 or slightly tilted towards the optimization loss to refine decision quality. The weight setting procedure is provided in Table 5.

- *Optimization layer regularization via ADMM penalty tuning.* In the SPO-A framework, the penalty term ρ serves a dual role: an adaptive learning rate for dual variable updates while simultaneously transforming the original hard constraints into a more flexible soft solution space. By carefully tuning ρ through cross-validation, we balance the trade-off between constraint satisfaction and model flexibility, enhancing generalization without compromising feasibility. The value of the penalty term is selected through cross-validation, which is provided in Section 5.2.4.4.

Table 5: Dynamic weight setting of prediction to matching MSE in Case A and Case B.

Epoch	Case A	Case B
0 ~ 100	50 : 1	100 : 1
100 ~ 200	50 : 1 → 1 : 1 ^a	100 : 1 → 1 : 1 ^b
> 200	1 : 1	1 : 1.05

^{a,b} The ratio is scheduled to decrease linearly over each epoch.

These two techniques offer complementary advantages for managing the prediction-optimization balance and ensuring solution feasibility for the SPO framework. When integrated with conventional regularization methods, they collectively form a comprehensive approach that simultaneously prevents overfitting while preserving high-quality decision quality across diverse operational scenarios.

5.1.4. Baselines

To compare the importance of both the SPO framework and the embedded alternating differentiation method (SPO-A), we compare the proposed model with three baseline methods: SPO with CVXPY (SPO-C), 2-stage predict-then-Optimize (PTO), and do-nothing method (DON).

- SPO-C: The SPO-C is an end-to-end SPO framework embedded with the implicit differentiation algorithm CVXPY by differentiating the KKT condition.
- PTO: The PTO method is the two-stage predict-then-optimize framework, in which the matching strategy is generated completely based on the prediction result.
- DON: The DON method does not change the original routes of the DVs and lets the drivers drive by themselves.

5.1.5. Target sensing distributions

The target sensing distribution can vary significantly depending on specific sensing tasks in VCS, necessitating that the proposed SPO-A framework exhibit robustness across a wide range of target sensing distributions. For instance, air quality monitoring typically requires uniformly distributed data collection across an entire urban area (Bales et al. 2012; Chen et al. 2018). In contrast, specialized monitoring tasks, such as detecting factory pollution or forest fires, often demand more frequent and granular data collection in densely populated regions or specific locations during particular dates or seasons (Khedo et al. 2010; Paulos et al. 2007).

Therefore, we create three different target distributions: Uniform distribution, Gaussian distribution, and Gaussian Mixture distribution (Xu et al. 2019) based on the one-week data from March 7 to March 13 to represent various potential requirements. Each distribution is generated using unique random seeds to ensure reproducibility while maintaining statistical independence. The specific generation procedures are as follows:

- **Uniform distribution (U).** We first compute the mean distribution of the one-week data in all grids for each time interval. We then generate uniformly distributed samples within the range [0.90, 1.10] and [0.85, 1.15] of the mean value for each grid for Case A and Case B, respectively. Negative values are filtered.
- **Gaussian distribution (G).** Based on the mean distribution for each time interval, we generate samples from a Gaussian distribution across the spatial domain. The distribution is centered at the mean with a fixed variance (15 for Case A and 25 for Case B). Negative values are filtered.
- **Gaussian Mixture distribution (GM).** We sample a Gaussian Mixture distribution over the spatial domain based on the mean distribution with two components of two different fixed variances for each time interval (10 and 20 for Case A, 40 and 50 for Case B, respectively). Negative values are filtered.

5.1.6. Evaluation metrics

To evaluate the matching performance of the SPO framework, we adopt two metrics to compare the divergence between the matching distribution $\mathbf{D}_a^{\tau+1}$ and the target distribution $\mathbf{T}_a^{\tau+1}$. Since the sensors are installed on vehicles, enhanced matching performance reflects better sensing accuracy for a given sensing task.

- **Root Mean Squared Error (RMSE):**

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (\mathbf{D}_{a,i}^{\tau+1} - \mathbf{T}_{a,i}^{\tau+1})^2}$$

- **Symmetric Mean Squared Percentage Error (SMAPE):**

$$\text{SMAPE} = \frac{100\%}{N} \sum_{i=1}^N \frac{|\mathbf{D}_{a,i}^{\tau+1} - \mathbf{T}_{a,i}^{\tau+1}|}{(\mathbf{D}_{a,i}^{\tau+1} + |\mathbf{T}_{a,i}^{\tau+1}|)/2}$$

5.2. Case A: Mid-size hailing dataset

In this section, we conduct the first experiment in the real-world taxi dataset in Hong Kong. The study area in the Kowloon District, Hong Kong SAR, is first discretized into hexagon grids by Uber's Hexagonal Hierarchical Spatial Index (H3) ([Uber H3 2023](#)) at H8 resolution. The average edge length of each hexagon is 531.41 m. For the mid-size network, we select 45 grids, and for the large-scale network, 68 grids are chosen, as illustrated in Figure 6. Real-time taxi information is collected from the HKTaxi application programming interface (API) from March 14 to March 24, 2023, and the taxi demand in each grid is aggregated every 15 minutes. Following preprocessing, the dataset contains a total of 2,075,864 data samples. The dataset is partitioned into training, validation, and testing sets in a ratio of 8 : 1 : 1.

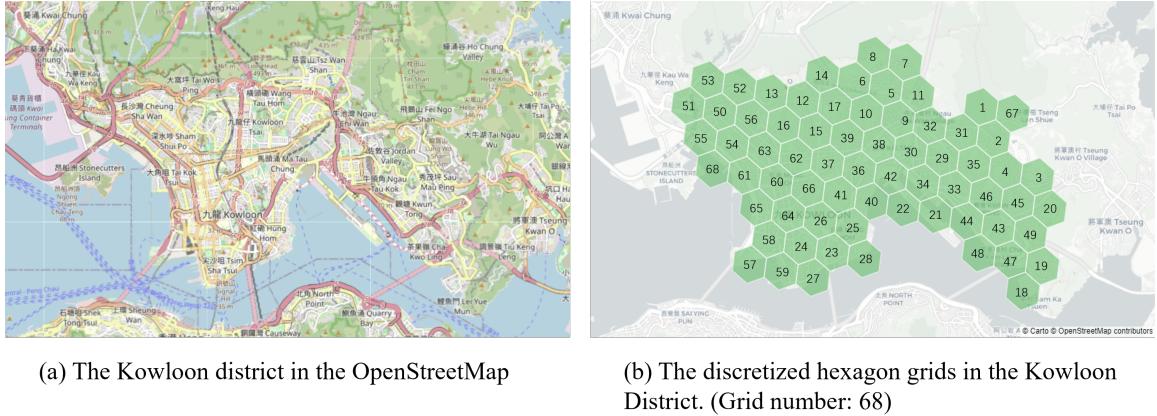


Figure 6: Overview of the research area in the Kowloon District, Hong Kong SAR.

Figure 7 visualizes the spatio-temporal taxi demand pattern in a week in the research area. From Figure 7 (a), we observe that high-demand areas are concentrated in the west-central and southern regions, particularly around Mong Kok and Hung Hom areas. Grid 62, located in the Mong Kok area, experiences the highest number of taxi requests. Figure 7 (b) illustrates the daily demand pattern in a week (Moving average with sliding window = 4) for three selected grids and on average. One can see that the temporal patterns vary across different grids with different demands. The average demand ranges from 100 to 200 vehicles per hour. Grid 62 exhibits the highest demand, ranging from 300 to 600 vehicles per hour, with three distinct demand peaks in the morning, noon, and late night. Grid 47 shows diverse peak hours in the morning, noon, and afternoon, occurring at around 8:00, 12:00, and 19:00. Meanwhile, the demand in grid 39 fluctuates smoothly at around 50 vehicles/hour.

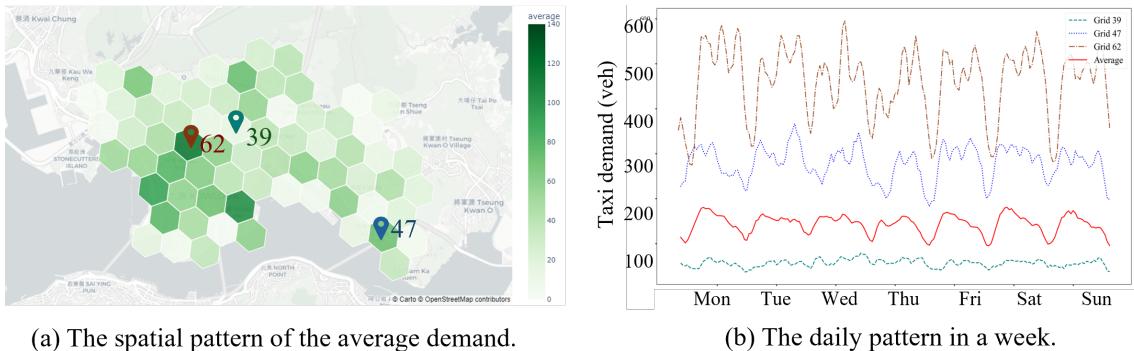


Figure 7: The spatio-temporal demand pattern in the research area.

We then compare the accuracy and computational efficiency performance of the proposed SPO framework with the baseline methods in mid and large-scale networks in dimensions 2,025 (45×45) and 4,624 (68×68),

respectively. The total travel budget is set to 10,000. The TGCN module is utilized as the prediction module in both the SPO framework and baseline methods. The ratio of DVs to NDVs is set to 6: 4.

5.2.1. Training curves

Figure 8 illustrates the training and testing curves of the proposed SPO-A framework under Uniform distribution in mid-sized and large-scale networks in case A. Both the training and testing curves for the large-scale (68×68) dataset exhibit greater fluctuations compared to those of the mid-sized (45×45) dataset, with both configurations converging around Epoch 250.

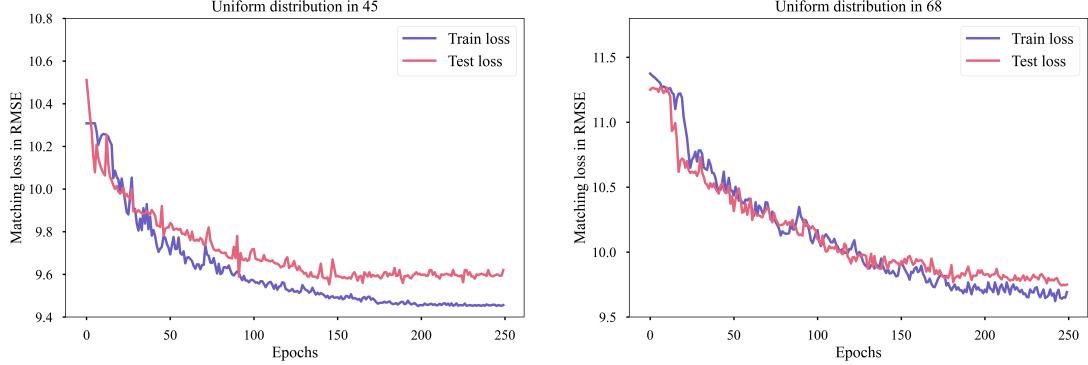


Figure 8: Training curves and test curves of SPO-A under the Uniform distribution in the mid-sized (45×45) and large-scale (68×68) network (Unit for RMSE: Vehicles/hour).

5.2.2. Discussions

This section analyzes the matching accuracy, computational efficiency, and detailed matching patterns comparing the PTO and proposed SPO frameworks.

5.2.2.1 Discussion on the matching accuracy

The accuracy performance of the proposed SPO-A framework and other baseline methods is presented in Table 6.

Table 6: Performance of SPO-A and other baselines in HK dataset under Uniform target distribution (Unit for RMSE: Vehicles/hour).

Grid size	Dimension	Total budget	SPO-A(Ours)		SPO-C		PTO		DON	
			RMSE	SMAPE (%)	RMSE	SMAPE (%)	RMSE	SMAPE (%)	RMSE	SMAPE (%)
45 × 45	2,025	8000	9.072	31.623	9.510	31.551	9.563	34.081	14.517	52.643
		10,000	9.647	31.552	9.541	31.457	9.551	33.987	14.517	52.643
		12,000	9.589	31.587	9.593	31.558	9.554	31.523	14.517	52.643
		15,000	9.059	31.426	9.507	31.431	9.563	34.087	14.517	52.643
68 × 68	4,624	8,000	9.834	33.479	9.962	33.514	10.701	38.166	15.449	56.384
		10,000	9.689	32.966	9.820	33.643	10.519	36.955	15.449	56.384
		12,000	9.291	30.504	9.678	33.282	10.098	33.162	15.449	56.384
		15,000	9.752	33.277	10.096	32.478	10.547	37.495	15.449	56.384

Under the mid-size network with 45×45 grids, the two end-to-end methods, SPO-A and SPO-C show comparable matching performance to the two-stage PTO method. Notably, the SPO-A method outperforms the PTO method under budget constraints of 8,000 and 15,000, showing an improvement of 5.20%. However, in scenarios with a budget of 10,000 and 12,000, SPO-A slightly underperforms compared to SPO-C and PTO by less than 1.15%. Under the large-scale network, one can see that the proposed method consistently outperforms PTO and SPO-C across all budget settings, achieving the highest average matching accuracy. On average, the SPO-A framework improved matching performance by 7.90% compared to PTO and by 2.51% compared to SPO-C across all budget settings.

The results from different network scales indicate that the end-to-end framework presents an obvious enhancement over the two-stage methods under the predict-then-optimize paradigm. Moreover, the proposed SPO with alternating differentiation methods (SPO-A) method shows better and more robust performance in the large-scale network than in the mid-size network, validating the effectiveness of the alternating differentiation method in optimization.

5.2.2.2 Discussion on the computational efficiency

In this section, we evaluate the computational efficiency of the SPO-based frameworks, SPO-A and SPO-C, across both mid-size (45×45) and large-scale (68×68) networks, with a convergence threshold of $\xi = 5 \times 10^{-2}$. All experiments are conducted with a total budget of 10,000 under a Uniform distribution, and the other hyper-parameters follow the configuration specified in Table 4. Each experiment is repeated 5 times. The average total running time and the basic information of variables and constraints are reported in Table 7.

The results in Table 7 indicate that the running time of both SPO-A and SPO-C increases when the dimension of variables increases from mid-size (2,025) to large-scale (4,624) networks. Notably, SPO-A exhibits a 4.01% advantage over SPO-C in the large-scale setting, although SPO-A slightly under performs SPO-C for the mid-size network.

Table 7: Comparison of the total running time of SPO-A and SPO-C with convergence threshold $\xi = 0.05$ (Unit for time: $\times 10^3$ sec) in Case A.

	Mid-size	Large-scale
Number of grids	45	68
Number of variables	2,025	4,624
Number of constraints	6,120	13,940
Running time of SPO-A	26.81 (+9.79%)	35.28 (-4.01%)
Running time of SPO-C	24.42	36.75

Figure 9 presents a comparative evaluation of the matching accuracy (measured in RMSE) and computational efficiency (total running time) for SPO-A and SPO-C under both mid-size and large-scale networks, with a convergence threshold $\xi = 5 \times 10^{-2}$. The results demonstrate that SPO-A achieves superior performance in the large-scale network, outperforming SPO-C in both accuracy and computational speed. This suggests that SPO-A is more scalable for high-dimensional problems, delivering competitive accuracy while reducing runtime. In the mid-size network, SPO-C exhibits marginally better performance, with SPO-A showing a modest 1.11% decrease in RMSE matching accuracy and an approximate 9.79% increase in running time. However, this difference in accuracy is relatively small, indicating that SPO-A remains a viable alternative even for mid-size applications.

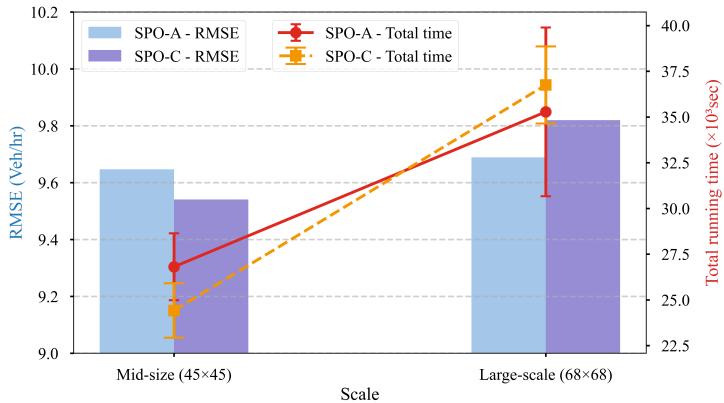


Figure 9: Comparison of matching accuracy and computational efficiency of SPO-A and SPO-C under mid-size and large-scale networks (Unit for accuracy: Vehicles/hour, unit for time: $\times 10^3$ sec).

5.2.2.3 Discussion on the prediction and matching patterns on the PTO and SPO frameworks

As introduced in Section 1, while the PTO framework may achieve better prediction accuracy in some cases, it can underperform in final decision accuracy. Table 8 provides a simple example of the relocation problems in VCS to illustrate how PTO fails in matching accuracy despite its predictive performance. Consider a scenario where the required number of vehicles in a grid is 100, but only 40 NDVs are available before relocating.

- If the NDVs are overestimated to 44 in the prediction stage, the system will then plan to relocate 56 DVs to this area in optimization. In practice, the final allocation of DVs might reach 56. Then the total number of final vehicles after relocation would be 96.
- If the NDVs are underestimated to 35 in the upstream prediction, the system will plan to relocate 65 DVs in optimization. However, achieving this higher requirement is more challenging, and the final allocation of DVs might only reach 62. The total number of final vehicles after relocation would be 102.

Overall, although overestimation of NDVs yields a smaller prediction deviation (overestimation: 4 vs. underestimation: 5), underestimation, despite its higher prediction error, can result in better final matching performance, as reflected in the lower matching deviation for DVs (overestimation: 4 vs. underestimation: 2). During peak hours, the tendency of PTO to overestimate demand, especially in high-demand areas (grids), often results in larger matching deviations. Because overestimation reduces the required number of relocation demands in the optimization stage, which will be easily fulfilled, inadvertently worsening the final matching discrepancy.

Table 8: Matching and prediction deviation comparison of the PTO in performing relocation tasks, with deviations measured in absolute terms. Unit for deviation: Vehicles/hour.

Case	Total required NDVs	Actual NDVs	Predicted NDVs	Required DVs	Achieved DVs	Final vehicles	Predicted NDV deviation	Matching DV deviation
Overestimation	100	40	44	56	56	96	4	4
Underestimation	100	40	35	65	62	102	5	2

We further investigate this phenomenon in the experiments for Case A, where we compare the prediction and matching deviations of the PTO and SPO-A frameworks across six selected grids at 10:00 AM under a mid-size network with a Uniform distribution, as shown in Figure 10. While both frameworks exhibit similar prediction accuracy across all 45 grids (PTO RMSE: 8.593 vs. SPO-A RMSE: 8.625), the SPO-A framework demonstrates superior overall matching performance (PTO: 9.551 vs. SPO-A: 9.647). However, in the high-demand six grids illustrated in Figure 10, PTO exhibits substantial prediction overestimation and larger matching deviation compared to SPO-A. This suggests that while the PTO framework demonstrates slightly better overall prediction performance in terms of RMSE (weighted across all grids), its tendency to overestimate in critical high-demand areas results in compounded errors in matching deviation, ultimately degrading the final matching accuracy. In contrast, the SPO-A framework, which is trained by directly minimizing task-specific matching errors with the assistance of prediction errors, corrects both intermediate predictions and final matching deviations more effectively than the PTO framework and therefore shows better overall matching accuracy.

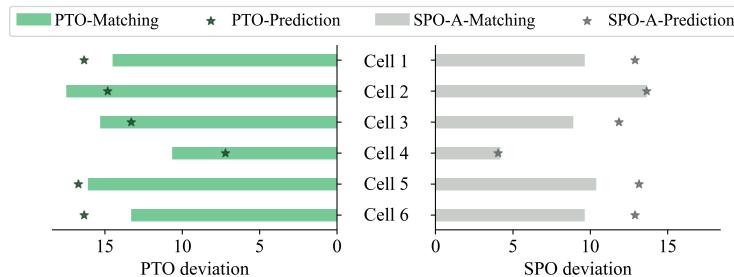


Figure 10: Matching and prediction deviation comparison of PTO and SPO-A frameworks under mid-size network in Case A. Unit for deviation: Vehicles/hour.

We then analyze the spatio-temporal patterns of demand prediction errors for the PTO and proposed SPO-A frameworks on the large-scale (68×68) network during the morning (9:00) and afternoon (18:00) peak hours on Thursday for Case A, as illustrated in Figure 11. Generally, the afternoon peak exhibits higher total demand and greater regional variation than the morning peak. SPO-A achieves more balanced predictions with lower bias during the lower-demand morning peak (Prediction RMSE: SPO-A: 8.782 vs. PTO: 9.175). However, during the high-variation afternoon peak, SPO-A shows a tendency toward underestimation, resulting in a higher RMSE than PTO (SPO-A: 9.583 vs. PTO: 9.272) but shows higher matching accuracy (SPO-A: 9.792 vs. PTO: 10.012 at 18:00). For a detailed spatial analysis, we select six representative grids in each time interval. In high-demand downtown areas, both frameworks tend to overestimate demand (e.g., cells 25 and 62 at 9:00; cell 66 at 18:00). But the PTO framework tends to show larger over-estimation than SPO-A. In transition areas surrounding high-demand downtown regions (e.g., cells 5, 47), prediction errors for both frameworks are generally low, fluctuating near zero. The degree of this overestimation is consistently lower for SPO-A. In peripheral areas, demand is generally underestimated, and here, SPO-A exhibits a relatively larger underestimation bias than PTO (e.g., cells 1, 20, and 29). To conclude, although the relatively large under-estimation of the SPO-A leads to final larger prediction errors, the reduced over-estimation in high-demand areas compensates and primarily contributes to the final overall superior matching performance. In terms of PTO, the pronounced overestimation in high-demand and surrounding transition areas during peak hours adversely impacts the final matching accuracy. This finding is consistent with analysis in mid-size networks.

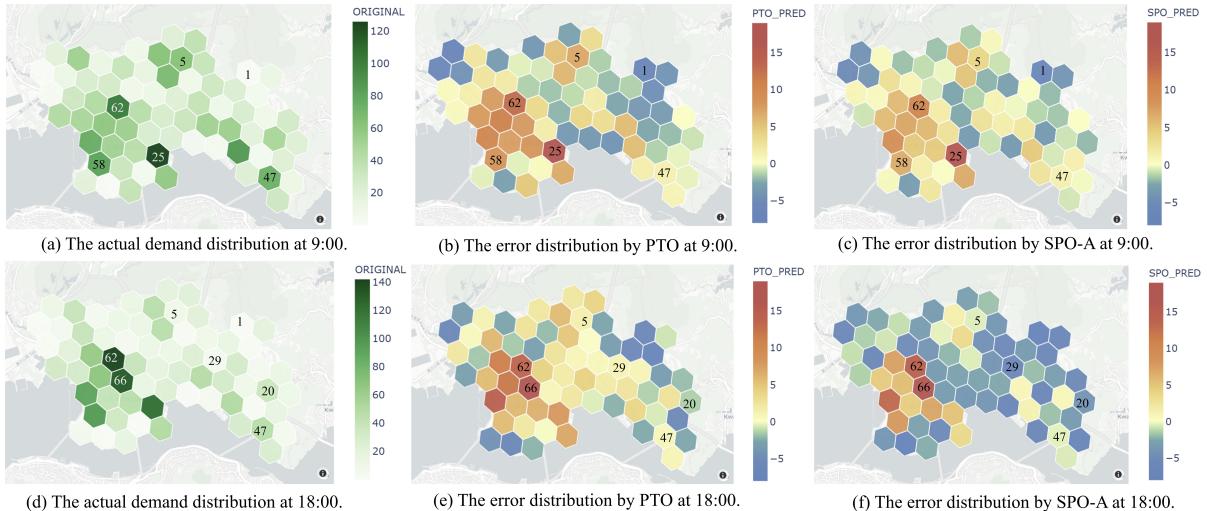


Figure 11: The spatial error distribution of predicted demand at 9:00 and 18:00 by the PTO and the proposed SPO-A framework under large-scale (68×68) networks. In (a) and (d), the underlying green represents the actual demand, with darker shades indicating higher values. In (b-c) and (e-f), the prediction error is defined as the deviation between predicted and actual demand, which is overlaid in color: red for overestimation, blue for underestimation, and yellow for near-zero errors.

5.2.3. Ablation study

In this section, we present four groups of ablation models with three different prediction models (TGCN, DCRNN (Li et al. 2018b), and ASTGCN (Guo et al. 2019)) to justify the effect of different prediction and optimization modules: SPO-A, SPO-C, PTO, NOP (no prediction but with the optimization), and DON (no prediction and no optimization). The four grouped models are examined in both mid and large-scale networks under Uniform and Gaussian target sensing distributions. The ratio of DVs to NDVs is 6: 4, and the total budget is 10,000.

5.2.3.1 Matching performance

We first discuss the matching accuracy of different prediction models.

The matching results are summarized in Table 9. Overall, the SPO-A methods with different prediction modules achieve robust and satisfactory performance for all scale network sizes and all target distributions in matching accuracy.

Table 9: The comparison of the matching performance of prediction modules in different frameworks under different networks and target distributions (Unit for RMSE: Vehicles/hour).

Prediction module	Framework	45-Uniform		45-Gaussian		68-Uniform		68-Gaussian	
		RMSE	SMAPE (%)	RMSE	SMAPE (%)	RMSE	SMAPE (%)	RMSE	SMAPE (%)
TGCN	SPO-A (Ours)	9.647	31.552	9.692	36.673	9.689	32.966	10.312	44.287
	SPO-C	9.541	31.457	9.858	36.680	9.820	33.643	10.642	46.029
	PTO	9.551	33.987	9.704	36.742	10.519	36.955	11.527	48.112
	NOP	13.465	47.988	13.581	53.551	14.569	54.450	15.722	52.675
	DON	14.517	52.643	19.352	75.164	15.449	56.384	20.583	76.103
DCRNN	SPO-A (Ours)	8.779	30.447	8.126	36.368	9.223	35.507	10.186	42.956
	SPO-C	8.747	26.317	8.327	36.602	9.710	36.583	10.480	43.236
	PTO	8.781	28.895	8.804	40.760	9.710	36.853	11.957	51.171
	NOP	13.465	47.988	13.581	53.551	14.569	54.450	15.722	52.675
	DON	14.517	52.643	19.352	75.164	15.449	56.384	20.583	76.103
ASTGCN	SPO-A (Ours)	8.476	29.591	9.430	46.292	9.225	30.316	10.667	44.606
	SPO-C	8.486	29.980	9.496	36.479	9.644	36.360	10.669	44.702
	PTO	9.694	30.301	9.986	47.338	9.721	32.610	11.071	45.366
	NOP	13.465	47.988	13.581	53.551	14.569	54.450	15.722	52.675
	DON	14.517	52.643	19.352	75.164	15.449	56.384	20.583	76.103

- (1) **Justifying the matching performance of different prediction models:** When comparing the SPO-A methods with two-stage methods incorporating three similar prediction modules—TGCN, DCRNN, and ASTGCN, we observe that methods utilizing the DCRNN prediction module consistently demonstrate superior matching performance relative to the other two prediction modules. Furthermore, the SPO-A method outperforms other benchmark models in all scenarios except for the Uniform distribution with 45 grids. This indicates the robustness of the SPO framework across various network sizes and target distribution scenarios.
- (2) **Justifying the use of the prediction module:** Comparing NOP with SPO-A, we observe a significant decrease in matching performance by 34.01% on average when prediction is removed while retaining optimization. This indicates the crucial role of the prediction module in achieving higher matching performance.
- (3) **Justifying the use of the alternating differentiation method in optimization:** Comparing SPO-A and SPO-C, it is evident that substituting the alternating differentiation method with the implicit differentiation method results in a significant decline in matching accuracy in large-scale networks, despite only minor differences between the two frameworks in mid-size networks. This highlights the importance of the alternating differentiation method in large-scale networks.

5.2.3.2 Prediction performance

We then proceed to compare the prediction accuracy of different prediction modules, including TGCN, DCRNN, and ASTGCN in SPO-C, SPO-A, and PTO framework, and compare the prediction accuracy with matching accuracy under Uniform and Gaussian distribution in mid-size and large-scale networks, as is illustrated in Figure 12. The implementation details of the hyperparameters of each prediction module are provided in Appendix B1.

- (1) **Justifying the prediction accuracy across different frameworks.** For the SPO-A, SPO-C, and PTO frameworks with various prediction models, it is noteworthy that the prediction accuracy of the SPO framework (including both SPO-A and SPO-C) is slightly lower (by 0.2–0.6 in RMSE) than that of the two-step PTO framework in some cases (in off-peak hours), despite the SPO framework achieving better final matching accuracy on average. This difference can be attributed to the SPO framework’s training process, which involves

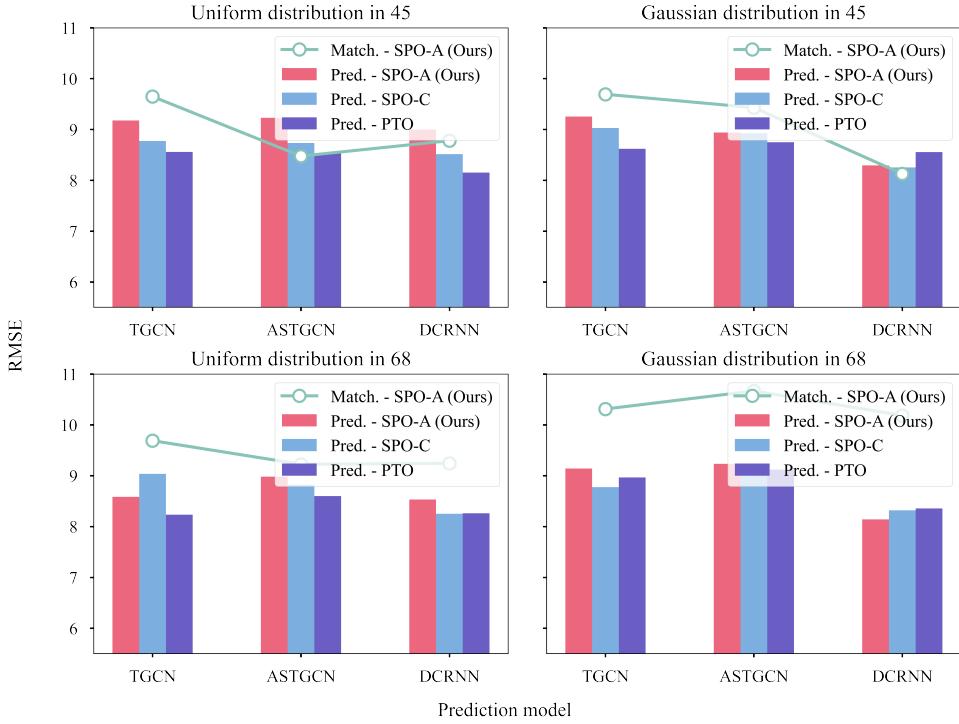


Figure 12: The prediction accuracy of different prediction models in SPO-A, SPO-C, and PTO framework and the matching accuracy of SPO-A framework under Uniform and Gaussian distribution in mid-size and large-scale networks (Unit for RMSE: Vehicles/hour; Pred., prediction; Match., matching).

a combined prediction and matching divergence, whereas the PTO framework is trained solely based on prediction divergence. This suggests that incorporating matching divergence can reduce the final matching error but may slightly increase the mid-term prediction error.

- (2) **Justifying the prediction accuracy of different prediction models.** DCRNN demonstrates the highest prediction accuracy, outperforming TGCN and ASTGCN, with ASTGCN being the least accurate. This trend is also consistent with the matching performance of the three prediction models discussed in Section 5.2.3.1.
- (3) **Comparing prediction accuracy with matching accuracy.** In most cases, the matching RMSE is higher than the prediction RMSE. However, in the Uniform and Gaussian distributions within mid-size networks with 45 grids, we observed that the final matching RMSE is lower than the mid-term prediction RMSE. This is because matching accuracy depends on both prediction accuracy and relocation accuracy, and the relocation RMSE can be relatively lower based on these specific prediction results.

5.2.4. Sensitivity analysis

This section presents four sensitivity analyses conducted on the proposed SPO methods and baselines, focusing on the convergence threshold, target sensing distributions, control ratios, and the penalty term. The first sensitivity analysis assesses how runtime scales with the convergence threshold under varying precision settings. The second sensitivity analysis on potential target distributions examines the robustness and applicability of the proposed SPO framework across diverse sensing tasks in VCS by evaluating its performance under varying target distributions. The third analysis investigates the influence of the control ratio, specifically the impact of NDVs and DVs in different proportions, on the execution of sensing tasks. In the fourth sensitivity analysis, cross-validation is used to select the appropriate value for the penalty term hyper-parameter. These analyses together provide insights into the robustness, adaptability, and effectiveness of the proposed framework under different realistic operational conditions in VCS.

5.2.4.1 Sensitivity analysis on the convergence threshold

This section systematically evaluates the relationship between computational efficiency and accuracy performance within the SPO-based framework, SPO-A and SPO-C. we conduct controlled experiments comparing SPO-A and SPO-C across mid-size (45×45) and large-scale (68×68) networks, with the threshold $\xi \in \{10^{-1}, 5 \times 10^{-2}, 10^{-2}\}$. All experiments are conducted with a total budget of 10,000 under a Uniform distribution while maintaining consistent hyperparameters as specified in Table 4. TGCN serves as our prediction module. Each configuration is replicated 5 times. Table 10 evaluates the computational efficiency and accuracy performance of the SPO-A framework across mid-size and large-scale networks with varying convergence thresholds, and Figure 13 compares the computational efficiency of both SPO-A and SPO-C frameworks across mid-size and large-scale networks with different convergence thresholds.

Table 10: Comparison of matching accuracy and computational efficiency of the SPO-A framework under mid-size and large-scale networks with varying convergence thresholds.

Threshold ξ	Mid-size (45×45)		Large-scale (68×68)	
	Accuracy (RMSE)	Efficiency (Time)	Accuracy (RMSE)	Efficiency (Time)
10^{-1}	9.995	23.52	10.137	32.54
5×10^{-2}	9.647	26.81	9.689	35.28
10^{-2}	9.501	31.20	9.465	41.25

Note: Accuracy measured in RMSE (Vehicles/hour), efficiency in total running time ($\times 10^3$ sec).

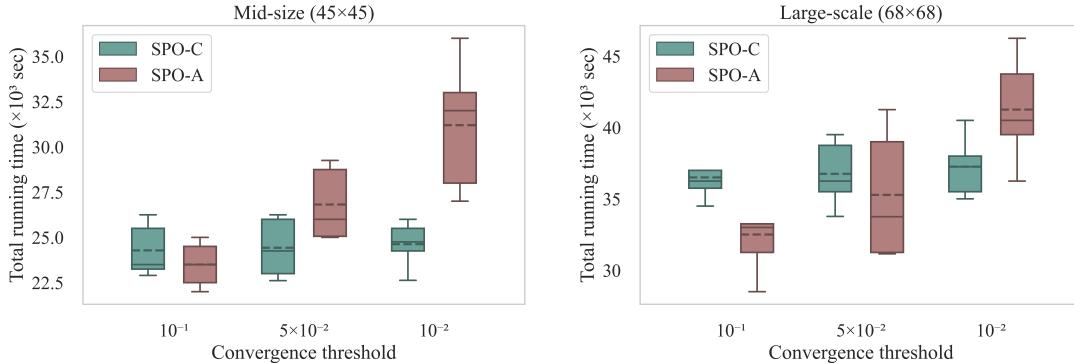


Figure 13: Comparison of the computational efficiency of SPO-A and SPO-C framework on the mid-size and large-scale networks on varying convergence thresholds. (Unit for running time: $\times 10^3$ sec).

Combined the results in Table 10 and Figure 13, we summarize four key findings:

- *Accuracy improvement:* A consistent improvement in matching accuracy (measured in RMSE) of SPO-A is observed with smaller threshold values across both mid-size and large-scale networks. But this accuracy gain comes at a substantial computational cost, with the total running time increasing approximately 21.11% to 24.62% when the threshold ξ tightens from 10^{-1} to 10^{-2} .
- *Scalability advantage:* SPO-A exhibits superior computational efficiency in large-scale networks compared to SPO-C, especially at $\xi = 10^{-1}$ and 5×10^{-2} . This advantage diminishes in mid-size networks, where SPO-C shows marginally better performance.
- *Threshold sensitivity:* SPO-A displays greater sensitivity to threshold precision in computational efficiency, evidenced by larger runtime standard deviations (maximum 16.17) compared to SPO-C (maximum 9.42) across all thresholds and all network scales.

- *Accuracy-Efficiency trade-off*: In large-scale networks, SPO-A achieves higher matching accuracy by 6.62% as ξ tightens from 10^{-1} to 5×10^{-2} , although requires approximately 26.76% additional computation time.

Overall, SPO-A demonstrates better scalability than SPO-C in high-dimensional settings, achieving comparable accuracy with relatively lower running time. Besides, selecting an appropriate convergence threshold for SPO-A is critical to maintaining both acceptable accuracy and stable computational efficiency across varying operational conditions. Our experiment results identify 5×10^{-2} as the optimal convergence threshold for the SPO-A framework in large-scale networks, achieving a balanced accuracy gain and comparable running time. These findings also highlight the context-dependent nature of threshold selection, where strict thresholds (smaller ξ) favor matching accuracy while moderate thresholds (larger ξ) prioritize computational efficiency.

5.2.4.2 Sensitivity analysis on the uncertainty of target sensing distributions

To evaluate the robustness of the framework under varying target distributions, we conduct a sensitivity analysis using Uniform, Gaussian, and Gaussian Mixture distributions. This analysis is performed at 15-minute intervals over a 24-hour period to comprehensively assess the adaptability of the proposed framework to different sensing scenarios. Note that the Gaussian Mixture distribution diverges most from the original distribution. The ratio of DVs to NDVs is set to 6: 4, and the total budget is set to 10,000. Figure 14 shows the absolute divergence in each grid from 8:00 to 16:00. Each pixel in each sub-figure represents the absolute value of the divergence between the target distribution and the matching distribution. The lighter the color is, the larger the divergence is.

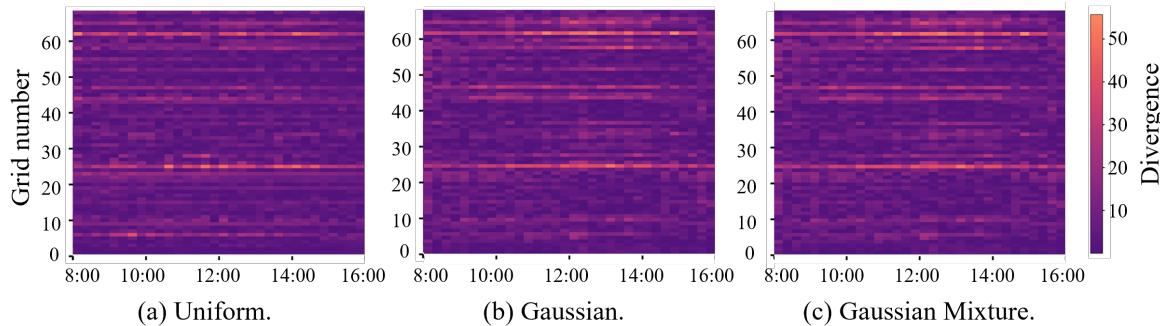


Figure 14: Sensitivity analysis of the SPO framework on different time intervals under Uniform, Gaussian, and Gaussian Mixture distributions in large-scale networks.

From Figure 14, it is apparent that Grid 62 and Grid 25 have larger divergence, especially from 10:00 to 12:00 in all three target distributions. This discrepancy may be attributed to the anticipation of the prediction from free vehicle movements based on weekly patterns. Grid 62 in the Mong Kok area and Grid 25 in the Hung Hom area experience a surge in demand during noon. When the distribution of NDVs deviates from the target distribution, the remaining DVs struggle to compensate for this divergence. Conversely, in grids with smoother demand fluctuations, the matching divergence is less. Overall, significant fluctuations in actual taxi demand may challenge the ability of the SPO-A framework to effectively satisfy matching distributions. However, if the target distribution diverges substantially from the original distribution, SPO-A is likely to outperform other baseline methods.

5.2.4.3 Sensitivity analysis on the uncertainty of control ratios

In this section, we analyze how matching performance varies among the SPO-A, SPO-C, and the PTO framework across different control ratios in real traffic networks, as illustrated in Figure 15. As we assume that the DVs will fully obey the assignment from the dispatching center, we implement a straightforward classification approach: any DV rejecting a task is classified as an NDV. Therefore, to account for varying compliance levels, we perform sensitivity analysis across a range of DV-to-NDV control ratios (20%-80%), with increments of 10%, in both mid and large-scale networks under Uniform, Gaussian, and Gaussian Mixture distributions to simulate different operational scenarios. We compare the improvement in matching accuracy of the proposed SPO-A framework against the SPO-C and PTO frameworks, using the RMSE of the PTO framework as a reference.

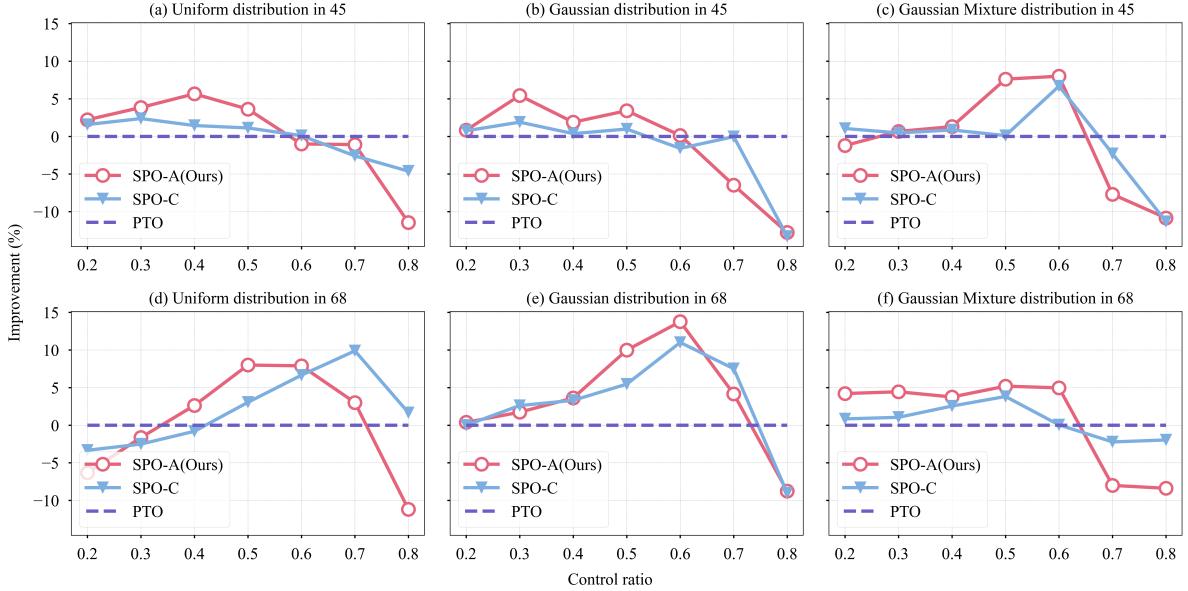


Figure 15: The improvement in the matching performance of SPO-A with SPO-C and PTO frameworks under Uniform, Gaussian, and Gaussian Mixture distributions in mid-size and large-scale networks, using the matching RMSE of the PTO framework as a reference.

In the mid-size network, as shown in Figure 15 (a), (b), and (c), with control ratios ranging from 20% to 60%, the SPO-A framework consistently outperforms the SPO-C and PTO frameworks in most cases. The highest improvement of the SPO-A, 7.52%, is observed at a 60% control ratio under the Gaussian Mixture distribution. Additionally, both the SPO-A and SPO-C frameworks demonstrate better matching performance than the PTO framework within this control ratio range. The SPO-C framework performs comparably to the PTO framework, showing an improvement of around 2% over the PTO framework.

In the large-scale network, as shown in Figure 15 (d), (e), and (f), the SPO-A framework achieves even better matching performance compared to the mid-size network, with a peak improvement of 13.75% under the Gaussian distribution compared with the PTO framework. The SPO-A method also significantly outperforms both the SPO-C and PTO frameworks, showing more substantial improvements over these two baselines.

However, with control ratios exceeding 60% in mid-size and 70% in large-scale networks, the matching performance of the SPO-A framework declines, falling below that of the SPO-C and PTO methods. This decline is attributed to precision errors occurring during the convergence process in the alternating differentiation iterations.

Overall, the SPO-A framework demonstrates significantly better performance than the SPO-C and PTO frameworks within the control ratio range of 0.3 to 0.7. As the control ratio increases, the precision errors in the alternating differentiation method can enlarge, negatively impacting the final matching performance.

5.2.4.4 Sensitivity analysis on the penalty term in SPO-A

The penalty term ρ plays a crucial role in balancing computational efficiency and accuracy within the SPO-A framework. It acts as an adaptive learning rate for dual variable updates while simultaneously transforming the original hard constraints into a more flexible soft solution space. A large ρ can accelerate convergence speed, show faster initial convergence. However, a large ρ may lead to overfitting, cause oscillations near the optimum, and even end in a local optimum. Conversely, a small ρ slows down convergence, provides smoother updates, but increases the total training time, especially in large-scale problems. In this section, we perform cross-validation on the penalty term ρ in the SPO-A framework in the HK dataset under size 45×45 to identify the optimal value that balances accuracy and computational efficiency. We evaluate ρ over the set $\{0.02, 0.2, 1, 2, 3, 4, 5, 10, 20\}$ with the result presented in Figure 16. It is observed that as ρ increases, the RMSE rises significantly beyond $\rho = 10$, while the training time decreases marginally. The efficiency and accuracy curves intersect at $\rho = 2$, indicating

that $\rho = 2$ strikes a reasonable balance between accuracy and efficiency for the HK dataset. Although the RMSE continues to decrease with a minimum value of $\rho = 0.02$, the training time increases by up to four times compared to $\rho = 2$. Within the range of $\rho = 1$ to $\rho = 5$, the RMSE and efficiency fluctuate slightly. These results suggest that, in selecting an optimal penalty term ρ , it is advisable to first evaluate across different orders of magnitude, followed by fine-tuning within the chosen range to achieve the best trade-off between accuracy and computational efficiency.

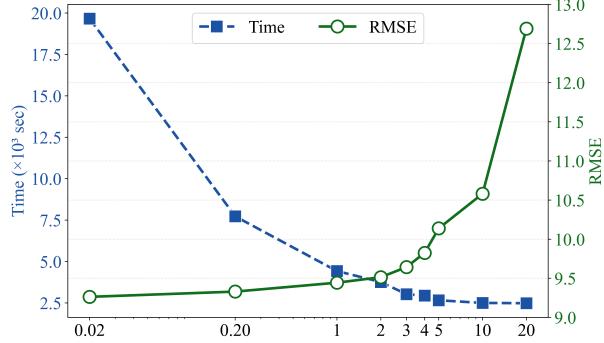


Figure 16: Sensitivity analysis on the penalty term ρ in the SPO-A framework (in log scale).

5.3. Case B: Large-scale ride-hailing dataset

To further validate the adaptability of the proposed SPO framework in real-world scenarios, we conduct an additional experiment using a larger dataset from central Chengdu, China.

5.3.1. Research area

The dataset comprises two weeks of taxi trajectory data between August 3 and August 16, 2014, sourced from the Open Dataset, with demand aggregated at 15-minute intervals. After preprocessing, the dataset contains 9,985,238 samples, approximately 5 times the total volume of the Hong Kong dataset (Case A). Figure 17(a) presents the study area mapped on OpenStreetMap. Figures 17(b) and (c) illustrate the spatial distribution of travel demand across 100 grids (H8 resolution) at 12:00 and 18:00, respectively. The visualization reveals that high-demand grids are predominantly concentrated in central areas, with demand intensity decreasing radially outward. The comparison reveals significantly higher demand during the evening peak (18:00) compared to noon (12:00), as clearly shown by the increased density of darker-shaded grids in Figures 17(c).

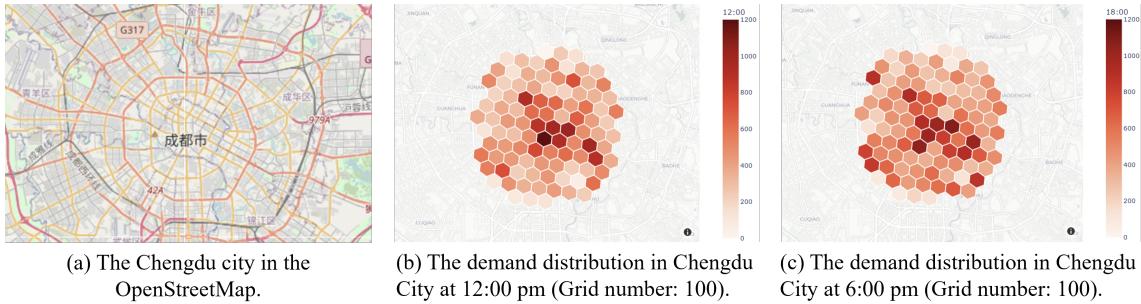


Figure 17: Overview of the research area in Chengdu City, China. Grids with darker shading represent higher taxi demand.

5.3.2. Experimental configuration

The prediction module in Case B extends beyond Case A by incorporating three distinct feature categories: 1) historical taxi demand per grid, 2) weather conditions, and 3) POI distributions. For weather data, we utilize hourly

temperature (continuous) and precipitation (categorical) measurements obtained from Weather Spark ([Weather Spark 2025](#)). The POI feature encompasses 13 facility categories within each grid, including transportation hubs, tourist attractions, public amenities, dining establishments, and shopping centers.

To ensure comprehensive evaluation, we examine three network scales (60×60 , 80×80 , and 100×100 grids) representing progressively larger urban areas. Vehicle compliance is assessed through systematic variation of the DV-to-NDV control ratio from 20% to 80% in 10% increments across all scenarios. Furthermore, we employ three distinct target sensing distributions (Uniform, Gaussian, and Gaussian Mixture) to prevent model overfitting to any single demand pattern. Complete hyperparameter configurations are detailed in Table 4, while the dynamic weighting strategy specifications are provided in Table 5.

5.3.3. Accuracy and efficiency performance

We evaluate the matching accuracy of the proposed SPO-A framework against SPO-C and PTO across three large-scale network configurations in Case B, considering Uniform, Gaussian, and Gaussian Mixture distributions. As shown in Figure 18, the results demonstrate three key findings consistent with Case A: 1) Both SPO-A and SPO-C consistently outperform the two-stage PTO framework for control ratios between 20% and 70%, achieving a maximum accuracy improvement of 9.22% in the Uniform distribution scenario for the 100×100 network. However, this performance advantage reverses when the control ratio reaches 80%, where SPO-A underperforms PTO in nearly all scenarios. We attribute this to precision errors in the alternating differentiation method, as the optimization module's influence becomes dominant at higher control ratios. 2) The SPO-A and SPO-C frameworks maintain comparable accuracy levels across the 20%-70% control ratio range, with observed deviations remaining minimal (peak deviation: 5.23%). 3) Overall, the SPO-A framework demonstrates robust consistency in matching accuracy across all tested network sizes and distribution scenarios, replicating the reliability observed in Case A.

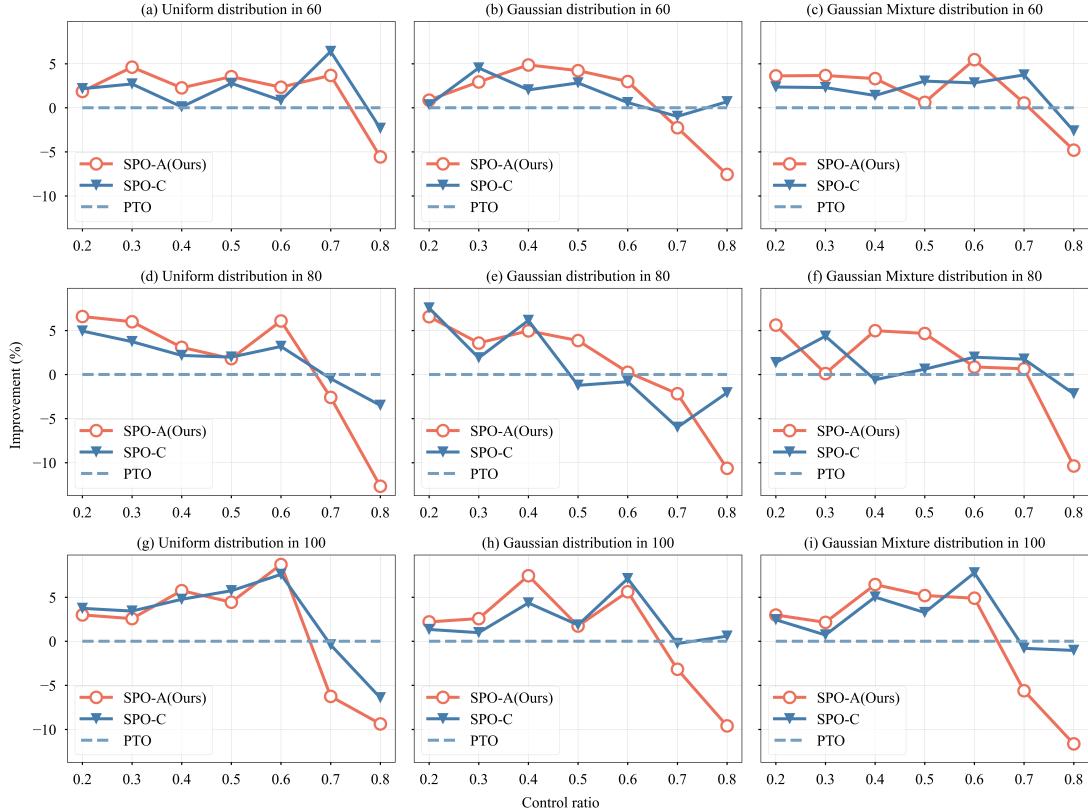


Figure 18: The improvement in matching performance of SPO-A with SPO-C and PTO frameworks under Uniform, Gaussian, and Gaussian Mixture distributions in three large-scale networks, using the matching RMSE of the PTO framework as a reference.

Table 11 summarizes the computational efficiency across the three network sizes (60×60 , 80×80 , and 100×100) and target distributions (Uniform, Gaussian, and Gaussian Mixture). One can observe minimal variation in total running time for SPO-A under different distributions within the same network size (0.91, 1.09, and 1.49 for 60×60 , 80×80 , and 100×100 networks, respectively). However, both SPO-A and SPO-C exhibit substantial running time expansion with the increment of network sizes, escalating from approximately 32,000 to 97,000. Notably, SPO-A maintains consistent superiority over SPO-C, demonstrating improvements of 2.2%, 5.7%, and 2.4% in total running time for the respective network sizes. The findings align with the results in Case A.

Overall, the consistent matching and efficiency performance of SPO-A across both Case A and Case B not only demonstrates the robustness and effectiveness of the SPO-A framework in terms of both matching accuracy and computational efficiency, but also establishes the generalization of the proposed SPO-A framework across different scales of problems formulated in quadratic programming.

Table 11: Computational efficiency comparison of different target sensing distributions in three large-scale networks in Case B. (Efficiency is reported in total running time ($\times 10^3$ sec.).)

Distribution	60×60		80×80		100×100	
	SPO-A	SPO-C	SPO-A	SPO-C	SPO-A	SPO-C
Uniform	32.39	33.12	54.82	56.38	95.85	98.24
Gaussian	33.53	33.98	56.97	58.43	97.43	95.33
Gaussian Mixture	34.18	34.23	56.23	59.66	97.69	97.95
Mean	33.37 ± 0.91	33.77 ± 0.58	56.01 ± 1.09	58.15 ± 1.49	96.09 ± 1.49	97.17 ± 1.60

6. Conclusions

This paper presents an end-to-end SPO framework coupled with an alternating differentiation method (SPO-A) for vehicle relocation problems in mobile sensing, aimed at enhancing sensing efficiency with limited budgets. The proposed SPO-A framework integrates a constrained QP optimization layer in the neural network and derives an explicit alternating differentiation method based on the ADMM. The matrix-based formulation in optimization enables seamless integration of the optimization layer into deep learning frameworks, facilitating efficient training and deployment for vehicle relocation problems in VCS. The unrolling alternating differentiation approach within the optimization layer enables effective backpropagation for large-scale networks, essential for training deep learning models.

The effectiveness of the proposed SPO-A framework is validated through two real-world experiments conducted in both mid-size and large-scale networks in Hong Kong and Chengdu, China. Results demonstrate that the SPO-A framework surpasses most benchmarks set by the SPO with implicit differentiation techniques (SPO-C) and PTO in overall matching accuracy, particularly in large-scale networks. Specifically, the SPO-A framework also outperforms the SPO-C in both matching accuracy and computational efficiency in most scenarios, highlighting the superior scalability of SPO-A for high-dimensional problems. The SPO-A framework outperforms PTO by providing more stable predictions during low demand variation and, crucially, by reducing overestimation in high-demand downtown areas during peak hours, which compensates for its peripheral underestimation and leads to superior overall matching performance. Finally, sensitivity analyses validate the robustness of the SPO-A framework against uncertainties in target distribution and varying control ratios, underscoring its applicability across relocation problems in diverse VCS scenarios.

Future directions of this research could focus on exploring further explicit differentiation methods specifically tailored for optimization layers, aimed at significantly enhancing computational efficiency, particularly in large-scale networks. Additionally, we aim to extend the application of the SPO framework to encompass other critical traffic management scenarios, such as vehicle routing, which presents an opportunity to tackle more intricate real-world challenges effectively. These advancements would not only expand the applicability of the SPO framework but also contribute to advancing the state-of-the-art in logistics and intelligent transportation systems.

Acknowledgments

The work described in this paper is supported by grants from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. PolyU/25209221 and PolyU/15206322) and a grant from the Otto Poon Charitable Foundation Smart Cities Research Institute (SCRI) at the Hong Kong Polytechnic University (Project No. P0043552). The contents of this article reflect the views of the authors, who are responsible for the facts and accuracy of the information presented herein.

Appendix A. Notations

Table A1 presents the notations in the paper, and Table A2 summarizes the dimensions of the vectorized variables for the SPO framework.

Table A1: List of notations.

<i>I</i>	The set of origin grids
<i>J</i>	The set of destination grids, $ I = J = N$.
<i>T</i>	The set of all time intervals.
<i>V</i>	The set of vehicle class.
Indices	
<i>v</i>	The index of vehicle type, $v = \{a, c, f\}$. a represents all vehicles, c represents controllable DVs, f represents non-dedicated free vehicles.
τ	The index of the time interval.
<i>k</i>	The index of iterations.
<i>n</i>	The index of constraints.
Variables as scalars	
$D_{v,i}^\tau$	The demand of vehicle type v in grid i at time τ .
$A_{f,i}^\tau$	The adjacent information for NDVs in grid i at time τ .
$H_{f,i}^\tau$	The hidden information for NDVs in grid i at time τ .
$x_{v,ij}^\tau$	The vehicle flow for vehicle type v from origin i to destination j at time τ .
$w_{c,ij}^\tau$	The incentive cost from origin i to destination j for DVs at time τ .
$m_{c,ij}^\tau$	The travel time from origin i to destination j for DVs at time τ .
Variables/Parameters as tensors	
\mathbf{y}_k	The one-dimension vector of vehicle flow in the k^{th} iteration.
\mathbf{D}_v^τ	The spatial distribution for vehicle type v at time τ .
\mathbf{T}_a^τ	The target distribution of all vehicles a at time τ .
\mathbf{s}_k	The slack variables in k^{th} iteration.
μ_k	The dual variables in k^{th} iteration.
Parameters	
\mathcal{L}	The loss function. \mathcal{L}_1 is the prediction loss, \mathcal{L}_2 is the matching loss, and \mathcal{L}_{SPO} is the SPO loss.
w	The weight. w_1 is the weight of the prediction loss in the SPO function, and w_2 is the weight of the matching loss in the SPO framework.
\mathbf{w}_p	The weight in the prediction model.
δ	The time length for each time interval.
α	The Look back window in the prediction module.
\mathbf{C}	The incentive cost vector.
R	The total budget of the incentive cost.
K	The total number of alternating layers.
ξ	The convergence threshold.
γ	The control ratio.
ρ	The penalty term.

Table A2: Dimension of the vectors in the SPO framework.

Vector	Dimension	Type	Vector	Dimension	Type
Variables					
\mathbf{y}	\mathbb{R}^{N^2}	Vector			
Parameters					
\mathbf{A}	$\mathbb{R}^{N \times N^2}$	Sparse Matrix	\mathbf{B}	$\mathbb{R}^{N \times N^2}$	Sparse Matrix
\mathbf{C}	\mathbb{R}^{N^2}	Vector	\mathbf{D}_v^τ	\mathbb{R}^N	Vector
\mathbf{P}	$\mathbb{R}^{N^2 \times N^2}$	Sparse Matrix	\mathbf{q}	\mathbb{R}^{N^2}	Vector
\mathbf{G}_1	$\mathbb{R}^{N \times N^2}$	Sparse Matrix	\mathbf{s}_1	\mathbb{R}^N	Vector
\mathbf{G}_2	$\mathbb{R}^{N^2 \times N^2}$	Sparse Matrix	\mathbf{s}_2	\mathbb{R}^{N^2}	Vector
\mathbf{G}_3	$\mathbb{R}^{1 \times N^2}$	Vector	\mathbf{s}_3	\mathbb{R}^1	Constant
\mathbf{G}_4	$\mathbb{R}^{N^2 \times N^2}$	Sparse Matrix	\mathbf{s}_4	\mathbb{R}^{N^2}	Vector
\mathbf{h}_1	\mathbb{R}^N	Vector	μ_1	\mathbb{R}^N	Vector
\mathbf{h}_2	\mathbb{R}^{N^2}	Vector	μ_2	\mathbb{R}^{N^2}	Vector
\mathbf{h}_3	\mathbb{R}^1	Vector	μ_3	\mathbb{R}^1	Constant
\mathbf{h}_4	\mathbb{R}^{N^2}	Vector	μ_4	\mathbb{R}^{N^2}	Vector

Appendix B. Hyper-parameters in different prediction modules.

Table B1 lists the hyper-parameters used in various prediction models.

Table B1: Hyper-parameters of different prediction modules.

Model	Hyper-parameter	Value
TGCN	Number of hidden layers	2
	Number of channels in each layer	64, 32
	Kernel size	2
	Dropout	0.2
	Activation function	ReLU
DCRNN	Number of hidden layers	2
	Number of channels in each layer	64, 32
	Dropout	0.2
	Activation function	ReLU
ASTGCN	Number of hidden layers	2
	Number of channels in each layer	32, 16
	Number of heads	2
	Dropout	0.2
	Activation function	ReLU

References

- Agrawal, A., Amos, B., Barratt, S., Boyd, S., Diamond, S., Kolter, J.Z., 2019. Differentiable convex optimization layers. *Advances in Neural Information Processing Systems* 32.
- Amos, B., Kolter, J.Z., . Optnet: Differentiable optimization as a layer in neural networks, in: *International Conference on Machine Learning*.
- Anjomshoaa, A., Duarte, F., Rennings, D., Matarazzo, T.J., deSouza, P., Ratti, C., 2018. City scanner: Building and scheduling a mobile sensing platform for smart city services. *IEEE Internet of things Journal* 5, 4567–4579.
- Bales, E., Nikzad, N., Quick, N., Ziftci, C., Patrick, K., Griswold, W., 2012. Citisense: Mobile air quality sensing for individuals and communities design and deployment of the citisense mobile air-quality system, in: *2012 6th International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth) and Workshops*, IEEE. pp. 155–158.
- Bengio, Y., 1997. Using a financial training criterion rather than a prediction criterion. *International Journal of Neural Systems* 8, 433–443.
- Berthet, Q., Blondel, M., Teboul, O., Cuturi, M., Vert, J., Bach, F., . Learning with differentiable perturbed optimizers. arxiv 2020. arXiv preprint arXiv:2002.08676 .
- Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J., et al., 2011. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning* 3, 1–122.
- Chang, X., Wu, J., de Almeida Correia, G.H., Sun, H., Feng, Z., 2022. A cooperative strategy for optimizing vehicle relocations and staff movements in cities where several carsharing companies operate simultaneously. *Transportation Research Part E: Logistics and Transportation Review* 161, 102711.
- Chang, X., Wu, J., Sun, H., Yan, X., 2023. A smart predict-then-optimize method for dynamic green bike relocation in the free-floating system. *Transportation Research Part C: Emerging Technologies* 153, 104220.
- Chen, X., Xu, S., Han, J., Fu, H., Pi, X., Joe-Wong, C., Li, Y., Zhang, L., Noh, H.Y., Zhang, P., 2020. Pas: Prediction-based actuation system for city-scale ridesharing vehicular mobile crowdsensing. *IEEE Internet of Things Journal* 7, 3719–3734.
- Chen, X., Xu, X., Liu, X., Pan, S., He, J., Noh, H.Y., Zhang, L., Zhang, P., 2018. Pga: Physics guided and adaptive approach for mobile fine-grained air pollution estimation, in: *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers*, pp. 1321–1330.
- Chuah, S.P., Xiang, S., Wu, H., 2018. Optimal rebalancing with waiting time constraints for a fleet of connected autonomous taxi, in: *2018 IEEE 4th World Forum on Internet of Things (WF-IoT)*, IEEE. pp. 629–634.
- Dai, Z., Han, K., 2023. Exploring the drive-by sensing power of bus fleet through active scheduling. *Transportation Research Part E: Logistics and Transportation Review* 171, 103029.
- Diamond, S., Boyd, S., 2016. Cvpxy: A python-embedded modeling language for convex optimization. *Journal of Machine Learning Research* 17, 1–5.
- Donti, P., Amos, B., Kolter, J.Z., 2017. Task-based end-to-end model learning in stochastic optimization. *Advances in Neural Information Processing Systems* 30.
- Elmachtoub, A.N., Grigas, P., 2022. Smart “predict, then optimize”. *Management Science* 68, 9–26.
- Fan, G., Jin, H., Liu, Q., Qin, W., Gan, X., Long, H., Fu, L., Wang, X., 2019. Joint scheduling and incentive mechanism for spatio-temporal vehicular crowd sensing. *IEEE Transactions on Mobile Computing* 20, 1449–1464.

- Fan, G., Zhao, Y., Guo, Z., Jin, H., Gan, X., Wang, X., 2021. Towards fine-grained spatio-temporal coverage for vehicular urban sensing systems, in: IEEE INFOCOM 2021-IEEE conference on computer communications, IEEE. pp. 1–10.
- Ferber, A., Wilder, B., Dilkina, B., Tambe, M., 2020. Mipaal: Mixed integer program as a layer, in: Proceedings of the AAAI conference on artificial intelligence, pp. 1504–1511.
- Ford, B., Nguyen, T., Tambe, M., Sintov, N., Fave, F.D., 2015. Beware the soothsayer: From attack prediction accuracy to predictive reliability in security games, in: Decision and Game Theory for Security: 6th International Conference, GameSec 2015, London, UK, November 4-5, 2015, Proceedings 6, Springer. pp. 35–56.
- Ganti, R.K., Ye, F., Lei, H., 2011. Mobile crowdsensing: current state and future challenges. *IEEE Communications Magazine* 49, 32–39.
- Geng, Z., Guo, M.H., Chen, H., Li, X., Wei, K., Lin, Z., 2021a. Is attention better than matrix decomposition?, in: International Conference on Learning Representations.
- Geng, Z., Zhang, X.Y., Bai, S., Wang, Y., Lin, Z., 2021b. On training implicit models. *Advances in Neural Information Processing Systems* 34, 24247–24260.
- Guo, S., Lin, Y., Feng, N., Song, C., Wan, H., 2019. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting, in: Proceedings of the AAAI conference on Artificial Intelligence, pp. 922–929.
- Guo, S., Qian, X., Dasgupta, S., Rahman, M., Jones, S., 2022. Sensing and monitoring of urban roadway traffic state with large-scale ride-sourcing vehicles. *The Rise of Smart Cities* , 551–582.
- Hasenfratz, D., Saukh, O., Walser, C., Hueglin, C., Fierz, M., Arn, T., Beutel, J., Thiele, L., 2015. Deriving high-resolution urban air pollution maps using mobile sensor nodes. *Pervasive and Mobile Computing* 16, 268–285.
- Honicky, R., Brewer, E.A., Paulos, E., White, R., 2008. N-smarts: networked suite of mobile atmospheric real-time sensors, in: Proceedings of the Second ACM SIGCOMM Workshop on Networked Systems for Developing Regions, pp. 25–30.
- Huang, K., de Almeida Correia, G.H., An, K., 2018. Solving the station-based one-way carsharing network planning problem with relocations and non-linear demand. *Transportation Research Part C: Emerging Technologies* 90, 1–17.
- Ji, W., Han, K., Liu, T., 2023a. A survey of urban drive-by sensing: An optimization perspective. *Sustainable Cities and Society* , 104874.
- Ji, W., Han, K., Liu, T., 2023b. Trip-based mobile sensor deployment for drive-by sensing with bus fleets. *Transportation Research Part C: Emerging Technologies* 157, 104404.
- Jiang, H., Ren, Y., Fang, J., Yang, Y., Xu, L., Yu, H., 2023. Ship: A state-aware hybrid incentive program for urban crowd sensing with for-hire vehicles. *IEEE Transactions on Intelligent Transportation Systems* .
- Jiao, Y., Tang, X., Qin, Z.T., Li, S., Zhang, F., Zhu, H., Ye, J., 2021. Real-world ride-hailing vehicle repositioning using deep reinforcement learning. *Transportation Research Part C: Emerging Technologies* 130, 103289.
- Kang, L., Poslad, S., Wang, W., Li, X., Zhang, Y., Wang, C., 2016. A public transport bus as a flexible mobile smart environment sensing platform for iot, in: 2016 12th International Conference on Intelligent Environments (IE), IEEE. pp. 1–8.
- Ke, J., Feng, S., Zhu, Z., Yang, H., Ye, J., 2021. Joint predictions of multi-modal ride-hailing demands: A deep multi-task multi-graph learning-based approach. *Transportation Research Part C: Emerging Technologies* 127, 103063.
- Ke, J., Yang, H., Zheng, H., Chen, X., Jia, Y., Gong, P., Ye, J., 2018. Hexagon-based convolutional neural network for supply-demand forecasting of ride-sourcing services. *IEEE Transactions on Intelligent Transportation Systems* 20, 4160–4173.

- Kek, A.G., Cheu, R.L., Meng, Q., Fung, C.H., 2009. A decision support system for vehicle relocation operations in carsharing systems. *Transportation Research Part E: Logistics and Transportation Review* 45, 149–158.
- Khedo, K.K., Perseedoss, R., Mungur, A., 2010. A wireless sensor network air pollution monitoring system. *International Journal of Wireless & Mobile Networks* 2, 31–45.
- Krantz, S.G., Parks, H.R., 2002. The implicit function theorem: history, theory, and applications. Springer Science & Business Media.
- Lei, Z., Qian, X., Ukkusuri, S.V., 2020. Efficient proactive vehicle relocation for on-demand mobility service with recurrent neural networks. *Transportation Research Part C: Emerging Technologies* 117, 102678.
- Li, C., Liu, M., Hu, Y., Wang, H., Xiong, Z., Wu, W., Liu, C., Zhang, C., Du, Y., 2022a. Investigating the vertical distribution patterns of urban air pollution based on unmanned aerial vehicle gradient monitoring. *Sustainable Cities and Society* 86, 104144.
- Li, J., Yu, J., Nie, Y., Wang, Z., 2020. End-to-end learning and intervention in games. *Advances in Neural Information Processing Systems* 33, 16653–16665.
- Li, S., Zhang, J., Xie, D., Yu, S., Dou, W., 2018a. High quality participant recruitment of mobile crowdsourcing over big data, in: 2018 IEEE Global Communications Conference (GLOBECOM), IEEE. pp. 1–6.
- Li, X., Feng, G., Sun, Y., Qin, S., Liu, Y., 2022b. A unified framework for joint sensing and communication in resource constrained mobile edge networks. *IEEE Transactions on Mobile Computing* .
- Li, Y., Yu, R., Shahabi, C., Liu, Y., 2018b. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting, in: International Conference on Learning Representations (ICLR '18).
- Lin, Y.H., Yin, X.F., Tian, Q., 2024. Unlocking efficiency: End-to-end optimization learning for recurrent facility operational planning. *Transportation Research Part E: Logistics and Transportation Review* 189, 103683.
- Liu, S., He, L., Max Shen, Z.J., 2021. On-time last-mile delivery: Order assignment with travel-time predictors. *Management Science* 67, 4095–4119.
- Liu, X., Ma, W., 2022. Ubiquitous sensing for smart cities with autonomous vehicles, in: The Rise of Smart Cities. Elsevier, pp. 523–549.
- Mandi, J., Stuckey, P.J., Guns, T., et al., 2020. Smart predict-and-optimize for hard combinatorial optimization problems, in: Proceedings of the AAAI conference on artificial intelligence, pp. 1603–1610.
- Masutani, O., 2015. A sensing coverage analysis of a route control method for vehicular crowd sensing, in: 2015 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops), IEEE. pp. 396–401.
- Miao, F., Han, S., Lin, S., Stankovic, J.A., Zhang, D., Munir, S., Huang, H., He, T., Pappas, G.J., 2016. Taxi dispatch with real-time sensing data in metropolitan areas: A receding horizon control approach. *IEEE Transactions on Automation Science and Engineering* 13, 463–478.
- Mišić, V.V., Perakis, G., 2020. Data analytics in operations management: A review. *Manufacturing & Service Operations Management* 22, 158–169.
- Paulos, E., Honicky, R.J., Goodman, E., 2007. Sensing atmosphere .
- Pavone, M., Smith, S.L., Frazzoli, E., Rus, D., 2012. Robotic load balancing for mobility-on-demand systems. *The International Journal of Robotics Research* 31, 839–854.
- Qi, M., Shi, Y., Qi, Y., Ma, C., Yuan, R., Wu, D., Shen, Z.J., 2023. A practical end-to-end inventory management model with deep learning. *Management Science* 69, 759–773.

- Qian, X., Guo, S., Aggarwal, V., 2022. Drop: Deep relocating option policy for optimal ride-hailing vehicle repositioning. *Transportation Research Part C: Emerging Technologies* 145, 103923.
- Qin, Z.T., Zhu, H., Ye, J., 2022. Reinforcement learning for ridesharing: An extended survey. *Transportation Research Part C: Emerging Technologies* 144, 103852.
- Saukh, O., Hasenfratz, D., Thiele, L., 2014. Route selection for mobile sensor nodes on public transport networks. *Journal of Ambient Intelligence and Humanized Computing* 5, 307–321.
- Sayarshad, H.R., Chow, J.Y., 2017. Non-myopic relocation of idle mobility-on-demand vehicles as a dynamic location-allocation-queueing problem. *Transportation Research Part E: Logistics and Transportation Review* 106, 60–77.
- Shou, Z., Di, X., 2020. Reward design for driver repositioning using multi-agent reinforcement learning. *Transportation Research Part C: Emerging Technologies* 119, 102738.
- Simchi-Levi, D., 2014. Om forum—om research: From problem-driven to data-driven research. *Manufacturing & Service Operations Management* 16, 2–10.
- Song, J., Han, K., Stettler, M.E., 2020. Deep-maps: Machine-learning-based mobile air pollution sensing. *IEEE Internet of Things Journal* 8, 7649–7660.
- Sun, H., Shi, Y., Wang, J., Tuan, H.D., Poor, H.V., Tao, D., 2023. Alternating differentiation for optimization layers, in: The Eleventh International Conference on Learning Representations. URL: <https://openreview.net/forum?id=KKBMz-EL4td>.
- Tang, B., Khalil, E.B., 2022. Pyepo: A pytorch-based end-to-end predict-then-optimize library with linear objective function, in: OPT 2022: Optimization for Machine Learning (NeurIPS 2022 Workshop).
- Tian, X., Yan, R., Liu, Y., Wang, S., 2023. A smart predict-then-optimize method for targeted and cost-effective maritime transportation. *Transportation Research Part B: Methodological* 172, 32–52.
- Uber H3, 2023. H3: A Hexagonal Hierarchical Geospatial Indexing System. URL: <https://github.com/uber/h3>.
- Wang, J., Wang, F., Wang, Y., Wang, L., Qiu, Z., Zhang, D., Guo, B., Lv, Q., 2019. Hytasker: Hybrid task allocation in mobile crowd sensing. *IEEE Transactions on Mobile Computing* 19, 598–611.
- Weather Spark, 2025. The Weather Year Round Anywhere on Earth. URL: <https://tw.weatherspark.com/>.
- Weikl, S., Bogenberger, K., 2013. Relocation strategies and algorithms for free-floating car sharing systems. *IEEE Intelligent Transportation Systems Magazine* 5, 100–111.
- Xu, S., Chen, X., Pi, X., Joe-Wong, C., Zhang, P., Noh, H.Y., 2019. ilocus: Incentivizing vehicle mobility to optimize sensing distribution in crowd sensing. *IEEE Transactions on Mobile Computing* 19, 1831–1847.
- Xu, S., Zhang, J., Meng, S., Xu, J., 2021. Task allocation for unmanned aerial vehicles in mobile crowdsensing. *Wireless Networks* , 1–13.
- Yan, R., Wang, S., 2022. Integrating prediction with optimization: Models and applications in transportation management.
- Yan, R., Wang, S., Fagerholt, K., 2020. A semi-“smart predict then optimize”(semi-spo) method for efficient ship inspection. *Transportation Research Part B: Methodological* 142, 100–125.
- Yan, R., Wang, S., Zhen, L., 2023. An extended smart “predict, and optimize”(spo) framework based on similar sets for ship inspection planning. *Transportation Research Part E: Logistics and Transportation Review* 173, 103109.

- Yan, R., Wang, S., Zhen, L., Laporte, G., 2021. Emerging approaches applied to maritime transport research: Past and future. *Communications in Transportation Research* 1, 100011.
- Yang, B., Han, K., Liu, S., Li, R., 2025a. Joint matching and pricing for taxi drive-by sensing. *Transportation Research Part E: Logistics and Transportation Review* 197, 104071.
- Yang, J., Chen, L., Su, Z., Ma, W., Zou, Z., An, K., 2025b. Decision-focused learning for optimal subsidy allocation in ride-hailing services. *Transportation Research Part C: Emerging Technologies* 180, 105301.
- Yang, J., Liang, E., Su, Z., Zou, Z., Zhen, P., Guo, J., Ma, W., An, K., 2025c. Dff: Decision-focused fine-tuning for smarter predict-then-optimize with limited data, in: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 26868–26876.
- Yang, Y., Yan, R., Wang, S., 2024. An efficient ranking-based data-driven model for ship inspection optimization. *Transportation Research Part C: Emerging Technologies* 165, 104731.
- Zhang, J., Shan, E., Wu, L., Yin, J., Yang, L., Gao, Z., 2024. An end-to-end predict-then-optimize clustering method for stochastic assignment problems. *IEEE Transactions on Intelligent Transportation Systems* .
- Zhang, R., Rossi, F., Pavone, M., 2016. Model predictive control of autonomous mobility-on-demand systems, in: 2016 IEEE International Conference on Robotics and Automation (ICRA), IEEE. pp. 1382–1389.
- Zhao, L., Song, Y., Zhang, C., Liu, Y., Wang, P., Lin, T., Deng, M., Li, H., 2019. T-gcn: A temporal graph convolutional network for traffic prediction. *IEEE Transactions on Intelligent Transportation Systems* 21, 3848–3858.
- Zhu, N., Liu, Y., Ma, S., He, Z., 2014. Mobile traffic sensor routing in dynamic transportation systems. *IEEE Transactions on Intelligent Transportation Systems* 15, 2273–2285.