For a new city that is committed to promoting Electric Vehicles (EVs), it is significant to plan the public charging infrastructure where charging demands are high. However, it is difficult to predict charging demands before the actual deployment of EV chargers for lack of operational data, resulting in a deadlock. A direct idea is to leverage the urban transfer learning paradigm to learn the knowledge from a source city, then exploit it to predict charging demands, and meanwhile determine locations and amounts of slow/fast chargers for charging stations in the target city. However, the demand prediction and charger planning depend on each other, and it is required to re-train the prediction model to eliminate the negative transfer between cities for each varied charger plan, leading to the unacceptable time complexity. To this end, we design an effective solution of Simultaneous Demand Prediction And Planning (SPAP): discriminative features are extracted from multi-source data, and fed into an Attention-based Spatial-Temporal City Domain Adaptation Network (AST-CDAN) for cross-city demand prediction; a novel Transfer Iterative Optimization (TIO) algorithm is designed for charger planning by iteratively utilizing AST-CDAN and a charger plan fine-tuning algorithm. Extensive experiments on real-world datasets collected from three cities in China validate the effectiveness and efficiency of SPAP. Specially, SPAP improves at most 72.5% revenue compared with the real-world charger deployment.

CCS Concepts:
• Mathematics of computing → Combinatorial optimization; • Theory of computation → Unsupervised learning and clustering;

Additional Key Words and Phrases: Urban transfer learning, demand prediction, infrastructure planning, electric vehicles

ACM Reference format:
Yizong Wang, Dong Zhao, Yajie Ren, Desheng Zhang, and Huadong Ma. 2023. SPAP: Simultaneous Demand Prediction and Planning for Electric Vehicle Chargers in a New City. ACM Trans. Knowl. Discov. Data. 17, 4, Article 60 (February 2023), 25 pages. https://doi.org/10.1145/3565577

This work was supported in part by the National Key Research and Development Program of China under Grant 2018AAA0101200; in part by the National Natural Science Foundation of China under Grant 61972044; in part by the Funds for International Cooperation and Exchange of NSFC under Grant 61720106007; in part by the 111 Project under Grant B18008; in part by the Fundamental Research Funds for the Central Universities under Grant 2020XD-A09-3; and in part by the BUPT Excellent Ph.D. Students Foundation under Grant CX2022217. Huadong Ma and Dong Zhao are corresponding authors.

Authors’ addresses: Y. Wang, D. Zhao (corresponding author), Y. Ren, and H. Ma (corresponding author), Beijing University of Posts and Telecommunications, Beijing, 100876, China; emails: wangyizong@bupt.edu.cn, dzhao@bupt.edu.cn, renyj@bupt.edu.cn, mhd@bupt.edu.cn; D. Zhang, Rutgers University, Piscataway, NJ, 08854; email: desheng.zhang@cs.rutgers.edu.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2023 Association for Computing Machinery.
1556-4681/2023/02-ART60 $15.00
https://doi.org/10.1145/3565577
1 INTRODUCTION

Electric Vehicles (EVs) are a key technology to reduce air pollution and greenhouse gas emissions. The global EV fleet has expanded significantly over the last decade, underpinned by supportive policies and technology advances. Only about 17,000 EVs were on the world’s roads in 2010, while the number had swelled to 10 million by 2020; meanwhile, the number of publicly accessible chargers increased by 60% and 46% in 2019 and 2020, respectively compared with the previous year [12, 13]. Publicly accessible chargers are indispensable where home and workplace charging is unavailable or insufficient to meet requirements (e.g., for long-distance travel) [13]. For a new city that is committed to promoting EVs, the primary task is to build a network of public EV charging stations from scratch for eliminating various concerns (e.g., charger unavailability, range anxiety) of potential EV buyers [39]. Moreover, given the high investment cost, charging station operators have concerns about the revenue and payback period. It is reported that the payback period would fall by 2 years for 1.9% increase in charger utilization ratio [1]. Accordingly, charging station operators would only want to deploy charging infrastructure where charging demands are high [9].

However, it is challenging to predict charging demands before the actual deployment of EV chargers for lack of operational data in a new city, resulting in a deadlock. To address this issue, a traditional way is to infer charging demands by leveraging implicit information such as parking demands [3] and population distribution [37]. Unfortunately, such indirect method is error-prone particularly when the EV market share is still small [3], as the implicit data have different distributions with charging demands in nature (detailed in Section 5.3). Recently, the advanced data acquisition technologies enable us to collect explicit data about charging events of EVs, which helps in charger planning [5, 9, 18, 23]. However, some popular data sources, such as taxi/bus trajectories [18, 33] and renting/returning records from electric car-sharing platforms [5, 23], are only limited to commercial EVs rather than private EVs. For the general charging stations except for those that are used exclusively for commercial EVs, the only available explicit data are their charger transaction records [9], whereas it is impossible in a new city.

To address the data scarcity issue, a direct thought is to leverage the emerging urban transfer learning paradigm, which has been successfully applied for various smart city cold-start problems [35] such as crowd flow prediction [34, 38], human mobility data generation [11], chain store site recommendation [10], POI recommendations [4], and parking hotspots detection [21]. More specifically, considering the similarity and difference of two cities via commonly available datasets, such as map, POI, traffic, and so on, can we learn the knowledge on charging demands from a charging station network that is already deployed in other cities, and further exploit it to predict charging demands, and meanwhile determine proper locations and amount of chargers for charging stations in a new city? However, it is still a non-trivial task, as the existing studies either still need a small amount of explicit data in the target city [4, 34, 38], or are very different from our problem settings [4, 10, 11, 21]. By contrast, this work does not rely on any explicit data in the new city. More specifically, we face a great challenge: the charger demand distribution varies with city-wide charger plans, and in turn, charger planning is dependent on the charging demand prediction, resulting in a deadlock. To effectively predict charging demands, it is necessary to capture complex spatial-temporal dependencies, affected by various profile factors (numbers of slow/fast chargers in a station and also its nearby stations) and context factors (POIs, road networks, and transportation). Furthermore, the data-driven prediction model trained on one city may not be well adapted to another city due to the dissimilar nature (e.g., city scale, development level, and strategy) of different cities, which is also known as the domain shift problem, resulting in the negative transfer [28]. Even if we have an effective model to predict the charging demands, it is still required to re-train the model to eliminate the negative transfer for each varied charging plan, leading to the unacceptable time complexity.
To this end, we design a novel algorithm named **Transfer Iterative Optimization (TIO)** for simultaneous demand prediction and planning for EV chargers in the target city, by iteratively utilizing an **Attention-based Spatial-Temporal City Domain Adaptation Network (AST-CDAN)** for charger demand prediction and a charger plan fine-tuning algorithm based on the dynamic programming. More specifically, we extract discriminative profile and context features from the multi-source data. The **AST-CDAN** is designed for transferring the knowledge on charging demands from the source city to the target city without EV charging stations, which consists of four components: a **ProfileNet** and a **ContextNet** that learn latent profile and context features from the raw extracted features, respectively, a **DemandNet** that predicts the charging demands over different time intervals of one day, and a **DomainNet** that promotes the features from **ProfileNet** and **ContextNet** to deeper domain-invariant representations. The collaboration of the four components effectively address the domain shift problem between cities. In summary, our main contributions are as follows:

- To the best of our knowledge, we are the first to present the concept and solution of **Simultaneous Demand Prediction And Planning (SPAP)** for EV chargers in a new city. It is different from the existing work, which conducts charger planning based on charging demands that are assumed to be known in advance or could be independently inferred. We prove the NP-hardness of the problem and the unacceptable time complexity of a straightforward approach (Section 2).

- We propose a novel **TIO** algorithm for simultaneous demand prediction and planning with a time complexity of $O(|C_{TC}|B^2)$, where $B$ is the budget and $|C_{TC}|$ is the number of charging stations in the target city (Section 4). A novel model **AST-CDAN** is proposed to accurately predict charger demands in the target city, based on the extracted profile and context features. (Section 3).

- Extensive experiments on real datasets collected from three cities in China demonstrate the advantages of **SPAP** over competitive baselines. Moreover, **SPAP** improves at most 72.5% revenue compared with the real-world charger deployment (Section 5). We have released the code and data for public use.\(^1\)

### 2 Overview

This section formally defines the **Simultaneous Demand Prediction and Planning** problem, and proves its NP-hardness and the unacceptable time complexity of a straightforward approach. Then, we outline our **SPAP** solution framework.

#### 2.1 Problem Formulation

We consider two cities: source and target cities with deployed charging station set $C_{SC}$ and candidate charging station set $C_{TC}$, respectively. There are two types of chargers, i.e., fast/slow charger, and each type gets different deployment costs and service prices. Therefore, we want to transfer the knowledge on charging demands from the source city to estimate the charging demand of each charger plan in the target city and decide the optimal charger plan. For convenience, we next define some basic concepts.

**Definition 1 (Charging Station).** A charging station is represented by a tuple $c_i = (l_i, n_i^S, n_i^F, e_i^S, e_i^F, p_i^S, p_i^F, y_i^S, y_i^F)$, consisting of the following nine elements:

- $l_i$, the physical location of $c_i$;
- $n_i^S \in \mathbb{N}$ and $n_i^F \in \mathbb{N}$, # of slow/fast chargers deployed in $c_i$;

\(^1\)https://github.com/easysam/SPAP.
\(- e_i^s \text{ and } e_i^f, \) the unit costs for deploying any one slow/fast charger in \( c_i; \)
\(- p_i^s = [p_{i1}^s, p_{i2}^s, \ldots, p_{iT}^s] \text{ and } p_i^f = [p_{i1}^f, p_{i2}^f, \ldots, p_{iT}^f], \) the service price vectors of each slow/fast charger over \( T \) time intervals of one day, where \( p_{it}^s \text{ and } p_{it}^f \) are the service prices for using slow/fast chargers during the \( t \)-th time interval;
\(- y_i^s = [y_{i1}^s, y_{i2}^s, \ldots, y_{iT}^s] \text{ and } y_i^f = [y_{i1}^f, y_{i2}^f, \ldots, y_{iT}^f], \) the charging demand vectors of slow/fast chargers over \( T \) time intervals of one day, where \( y_{it}^s \text{ and } y_{it}^f \) are defined as the utilization rates, i.e., the average proportion of the occupied time to the interval time, of each slow/fast charger during the \( t \)-th time interval.

For each deployed charging station \( c_i \in C_{SC}, \) all of its elements are known, whereas for each candidate charging station \( c_i \in C_{TC}, \) only a part of its elements, \((l_i, e_i^s, e_i^f, p_i^s, p_i^f)\), are known.\(^2\) We require to make a plan for deploying proper numbers of slow/fast chargers in each candidate station of the target city, defined as follows:

**Definition 2 (EV Charger Plan).** Given a set of candidate stations \( C_{TC} \) in the target city, an EV charger plan is a set \( N_{TC} = \{(n_i^s, n_i^f) \mid c_i \in C_{TC}, n_i^s \in \mathbb{N}, n_i^f \in \mathbb{N}\}. \) Note that, it is possible that we do not deploy any charger for one candidate station \( c_j \in C_{TC}, \) i.e., \( n_j^s = n_j^f = 0. \) For convenience, let \( N_{SC} \) denote the charger plan that has been deployed in the source city.

**Definition 3 (Charging Demand Prediction in the Target City (CDPT)).** Given the deployed charger plan \( N_{SC} \) in the source city, a specific charger plan \( N_{TC} \) in the target city, the multi-source context data (POIs, transportation, road networks) \( D_{SC} \) and \( D_{TC} \) in both the source and target cites, and the historical charging demand data \( Y_{SC} = \{(y_i^s, y_i^f) \mid c_i \in C_{SC}\} \) in the source city, the CDPT problem is to learn a function \( f \) to predict the charging demands for all the stations in the target city \( Y_{TC} = \{(y_i^s, y_i^f) \mid c_i \in C_{TC}\} \):

\[
\begin{align*}
\min_{f} & \quad \text{error}(\hat{Y}_{TC}, Y_{TC}), \\
\text{s.t.} & \quad \hat{Y}_{TC} = f(N_{SC}, N_{TC}, D_{SC}, D_{TC}, Y_{SC}).
\end{align*}
\]  

The actual \( Y_{TC} \) is unavailable for learning the predictor \( f, \) which lets the CDPT problem fall into the unsupervised transfer learning paradigm. A common method of addressing this problem is to alleviate the negative impact of the feature distribution discrepancy of instances between the source and target cities. Note that the features of instances from \( C_{SC} \) and \( C_{TC} \) are extracted from \( (N_{SC}, D_{SC}) \) and \( (N_{TC}, D_{TC}) \), respectively. Different \( N_{TC} \) requires to train different models.

**Definition 4 (Charger Planning in the Target City (CPT)).** Given a set of candidate stations \( C_{TC} \) in the target city, the deployed charger plan \( N_{SC} \) and the historical charging demand data \( Y_{SC} \) in the source city, the multi-source data \( D_{SC} \) and \( D_{TC} \) in both the source and target cities, a charging demand predictor \( f \) and a budget constraint \( B, \) the CPT problem is to find an EV charger plan \( N_{TC} \) in the target city such that the total revenue \( R \) is maximized, while the total deployment cost of chargers does not exceed \( B: \)

\[
\begin{align*}
\max_{N_{TC}} & \quad R = \sum_{i=1}^{C_{TC}} \sum_{t=1}^{T} \left( y_{it}^s \cdot p_{it}^s \cdot n_i^s + y_{it}^f \cdot p_{it}^f \cdot n_i^f \right),
\end{align*}
\]  

\(^2\)The service price has little room for choice due to the operating cost and the business competition, so it is easy to be determined.
Table 1. Definition of Notations

| Notation | Description |
|----------|-------------|
| $c_i$    | $i^{th}$ charging station |
| $l_i$    | physical location of $c_i$ |
| $n_i^S, n_i^F$ | # of slow/fast chargers deployed in $c_i$ |
| $N_{SC}$ | charger plan that has been deployed in the source city |
| $N_{TC}$ | an EV charger plan in the target city |
| $u_i^S, u_i^F$ | limitation on slow/fast charger numbers in each charging station |
| $e_i^S, e_i^F$ | unit costs for deploying any one slow/fast charger in $c_i$ |
| $p_i^S, p_i^F$ | service price vectors of each slow/fast charger in $c_i$ over $T$ time intervals of one day |
| $y_i^S, y_i^F$ | charging demand vectors of slow/fast chargers in $c_i$ over $T$ time intervals of one day |

Note that the charger numbers in each station are constrained by $u_i^S$ and $u_i^F$ to avoid unrealistic charger allocation. For brevity, we summarize these notations in Table 1.

2.2 Problem Complexity Analysis

In this subsection, we prove the NP-hardness of the CPT problem and analyze the time complexity of a straightforward approach.

Theorem 1. The CPT problem is NP-hard.

Proof. We prove the NP-hardness of the CPT problem by reducing the unbounded knapsack problem (UKP) \footnote{The UKP problem is illustrated as follows: given a knapsack of capacity $c > 0$ and $n$ types of items, where each item of type $i$ has value $v_i > 0$ and weight $w_i > 0$, the objective is to find the number $x_i > 0$ of each type of item such that the total value $\sum_{i=1}^{n} x_i v_i$ is maximized, while the total weight does not exceed the capacity, $\sum_{i=1}^{n} x_i w_i \leq c$.}

The UKP problem is illustrated as follows: given a knapsack of capacity $c > 0$ and $n$ types of items, where each item of type $i$ has value $v_i > 0$ and weight $w_i > 0$, the objective is to find the number $x_i > 0$ of each type of item such that the total value $\sum_{i=1}^{n} x_i v_i$ is maximized, while the total weight does not exceed the capacity, $\sum_{i=1}^{n} x_i w_i \leq c$.

If $Y_{TC}$ is $N_{TC}$-independent, then the CPT problem is illustrated as a special case: given a budget $B$ and a set of charging stations $C_{TC}$, where each station $c_i \in C_{TC}$ is represented as a tuple $(l_i, n_i^S, n_i^F, e_i^S, e_i^F, p_i^S, p_i^F, y_i^S, y_i^F)$ (Definition 1), the objective is to determine a charger plan $N_{TC} = \{(n_i^S, n_i^F) | i = 1, \ldots, |C_{TC}|\}$ such that the total revenue is maximized, while the total cost of
deploying chargers does not exceed the budget:

\[
\begin{align*}
\max_{\mathcal{N}_{TC}} & \quad \sum_{i=1}^{\mid \mathcal{C}_{TC} \mid} \sum_{t=1}^{T} \left( y_{it}^S \cdot p_{it}^S \cdot n_{i}^S + y_{it}^F \cdot p_{it}^F \cdot n_{i}^F \right), \\
\text{s.t.} & \quad \sum_{i=1}^{\mid \mathcal{C}_{TC} \mid} e_{i}^S \cdot n_{i}^S + e_{i}^F \cdot n_{i}^F \leq B.
\end{align*}
\] (4a)

(4b)

Given \( W = \{ w_i \mid i = 1, \ldots, n \}, V = \{ v_i \mid i = 1, \ldots, n \}, \) and \( X = \{ x_i \mid i = 1, \ldots, n \}, \) we map an instance of the UKP problem, \( I = (W, V, X, n, c) \), with an even \( n \), to the instance of the CPT problem where \( Y_{TC} \) is \( \mathcal{N}_{TC} \)-independent, denoted by \( I' = (C_{TC}, B) \), as follows: \( c \) is mapped to \( B; n/2 \) is mapped to \( |C_{TC}| \); for any \( i = 1, 2, \ldots, n/2, w_{2i-1} \) is mapped to the slow charger cost \( e_i^S \) of \( c_i \in C_{TC} \); \( w_{2i} \) is mapped to the fast charger cost \( e_i^F \) of \( c_i \in C_{TC} \); \( v_{2i-1} \) is mapped to the daily revenue \( \sum_{t=1}^{T} (y_{it}^S \cdot p_{it}^S) \) of \( c_i \in C_{TC} \), and \( v_{2i} \) is mapped to the daily revenue \( \sum_{t=1}^{T} (y_{it}^F \cdot p_{it}^F) \) of \( c_i \in C_{TC} \).

On one hand, if there is a solution for the instance \( I, X = \{ x_1, x_2, \ldots, x_n \}, \) then \( \{(n_{i}^S, n_{i}^F) \mid n_{i}^S = x_{2i-1}, n_{i}^F = x_{2i}, i = 1, \ldots, |C_{TC}| \} \) is a solution for the instance \( I' \).

On the other hand, if there is a solution for the instance \( I', \{(n_{i}^S, n_{i}^F) \mid i = 1, \ldots, |C_{TC}| \} \), then the numbers

\[
x_i = \begin{cases} 
n_{i}^S, & \text{if } i \in \{1, 3, \ldots, n-1\} \\
n_{i}^F, & \text{if } i \in \{2, 4, \ldots, n\}
\end{cases}
\] (5)

are a solution for the instance \( I \).

Thus, as long as there is a solution for the UKP problem, there is a solution for the special case of the CPT problem where \( Y_{TC} \) is \( \mathcal{N}_{TC} \)-independent, and vice versa. Then, the UKP problem can be reduced to the simplified CPT problem. Since the UKP problem is NP-hard [2], the general CPT problem is NP-hard.

Note that if \( \hat{Y}_{TC} \) is \( \mathcal{N}_{TC} \)-independent, then the CPT problem can be reduced to an unbounded Knapsack problem [2], which can be solved by dynamic programming or approximation algorithms. Indeed, the existing studies on charger planning generally determine charging demands in advance by estimating from historical data [5] or leveraging a plan-independent demand prediction method [9], and thus the charger planning problem can be transformed into the well-known Knapsack and Set-Cover problems or their variants. However, these studies do not apply to a new city. Now let us return to our problem setting where \( \hat{Y}_{TC} \) is \( \mathcal{N}_{TC} \)-dependent. In essence, the charging demands \( \hat{Y}_{TC} \) are determined by a non-linear function of \( \mathcal{N}_{TC} \), which requires to be trained with a deep learning model (see Section 3). Thus, the existing solutions, whether dynamic programming or other approximation algorithms, are not directly applicable anymore. Alternatively, a straightforward approach could be used, which finds the optimal solution from all the possible charger plans (\( \mathcal{N}_{TC} \)) by the brute-force search. However, it has an unacceptable time complexity as follows.

**Theorem 2.** If \( e_{i}^S = e_{i}^F = 1, \forall c_i \in C_{TC} \), then the CPT problem has \( \binom{B+2|\mathcal{C}_{TC}|-1}{2|\mathcal{C}_{TC}|-1} \) possible charger plan solutions by the brute-force search.

**Proof.** If \( e_{i}^S = e_{i}^F = 1, \forall c_i \in C_{TC} \), then the budget \( B \) is equal to the total number of chargers that we can deploy. Under this case, the number of the possible charger plan solutions for the CPT problem can be proved in two steps.
First, we change the constraints \( n_i^S \geq 0 \) and \( n_i^F \geq 0 \) to \( n_i^S \geq 1 \) and \( n_i^F \geq 1 \). The number of possible charger plans will be \( \binom{B-1}{2|C_{TC}|-1} \) by splitting \( B \) to the \( 2|C_{TC}| \) parts (for 2 charger types in \( |C_{TC}| \) stations) according to the stars and bars method in the context of combinatorial mathematics.

Second, we add a “virtual” charger to each charger type of each station in advance, and accordingly the budget is increased by \( 2|C_{TC}| \). Similarly, the number of possible charger plans is \( \binom{B+2|C_{TC}|-1}{2|C_{TC}|-1} \). Note that the “virtual” chargers are placeholders to satisfy the changed constraints, which do not actually exist. After removing the “virtual” charger in each charger type of each station, the budget is still \( B \), but the original constraints \( n_i^S \geq 0 \) and \( n_i^F \geq 0 \) can be satisfied. As a result, the number of the possible charger plan solutions for the CPT problem is \( \binom{B+2|C_{TC}|-1}{2|C_{TC}|-1} \).

Now we consider a small-scale problem setting with \( |C_{TC}| = 5 \) and \( B = 100 \) for example: given the time of 1 millisecond for demand prediction with a candidate charger plan, the total time required to traverse through all the \( \binom{109}{9} \) plans will reach 137 years! Not only that, for each changed charger plan, it requires to re-train the demand prediction model; given the time of 1 hour for training a model with a candidate charger plan, the total time required to train all the possible models will grow to \( 4.87 \times 10^8 \) years! Thus, it is necessary to design an effective solution that is able to greatly reduce the required number of trainings and predictions.

2.3 Solution Framework of SPAP

Figure 1 gives the framework of SPAP, consisting of two components: charger demand prediction and charger planning, which coordinate to make the charger plan with the highest revenue.

**Charger Demand Prediction.** This component addresses the CDPT problem with the following two modules:

- **Feature Extraction.** It extracts discriminative profile and context features for charging stations from both source and target cities (Section 3.1).
- **Attention-based Spatial-Temporal City Domain Adaptation Network (AST-CDAN).** It leverages the features from both source and target cities and the demand data from source city to predict the charging demands in the target city (Section 3.2).

**Charger Planning.** This component addresses the CPT problem with the following two modules:
Table 2. Top Eight Features with Highest Pearson Coefficients

| Feature                  | Pearson | Feature                  | Pearson |
|--------------------------|---------|--------------------------|---------|
| # of fast chargers       | 0.6678  | # of spot POIs           | 0.3087  |
| # of community POIs      | 0.4244  | # of slow chargers       | −0.2945 |
| # of parking lots        | 0.4010  | # of school POIs         | −0.2927 |
| # of all chargers        | 0.4609  | Street density           | 0.2632  |

— Transfer Iterative Optimization. To greatly reduce the required number of trainings and predictions, the TIO algorithm is designed to iteratively utilize the AST-CDAN for charger demand prediction and the Charger Plan Fine-tuning module to update the charger plan (Section 4.1).

— Charger Plan Fine-tuning. It fine-tunes the current charger plan to maximize the total revenue constrained by the budget using a dynamic programming algorithm (Section 4.2).

3 CHARGER DEMAND PREDICTION

3.1 Feature Extraction

To predict the charging demands, we extract the context and profile features of each charging station, and then analyze their correlations and also the feature domain shift between two cities.

3.1.1 Context Features. Intuitively, the number and diversity of POIs reflect the prosperity, and the surrounding road network and transportation conditions of a charging station reflect its convenience, all of which have influences on charging demands. Thus, we extract useful context features in the surrounding region (within radius $r$) of each charging station:

— POI Features. We classify POIs into eight categories: company, school, hotel, fast food, spot, community, hospital, and life service. Then, a 17-D POI feature vector is extracted, including fraction of POIs in each category, number of POIs in each category and POI entropy.

— Road Network Features. They include the average street length, intersection density, street density, and normalized degree centrality of intersections,\(^3\) obtained from the nearby streets.

— Transportation Features. They include the number of subway stations, number of bus stops and number of parking lots.

The above features are concatenated as a single vector and fed into the prediction model.

3.1.2 Profile Features. Intuitively, the charging demand of a station $c_i$ is affected not only by the amount and type of chargers deployed in the station itself, but also by the nearby stations $NS(c_i)$ in its surrounding region (within radius $r$). Thus we extract the profile feature vector as $[|NS(c_i)|, \sum_{c_j \in NS(c_i)}(n_j^S + n_j^F), n_i^S, n_i^F, n_i^S + n_i^F]$.

3.1.3 Feature Correlation Analysis. Table 2 lists eight features that are most correlated with the charging demands. It shows that the absolute Pearson coefficients are all above 0.26, indicating the effectiveness of the selected features for charging demand prediction.

3.1.4 Domain Analysis between Cities. To analyze the domain shift problem, we use the Maximum Mean Discrepancy (MMD)\(^2\) to quantify the difference between feature domains from the source city $SC$ and the target city $TC$, which maps the features into the Reproducing Kernel Hilbert Space (RKHS) $\mathcal{H}$\(^3\) and calculates the square distance between the means of the

\(^3\)The normalized degree centrality is defined as the proportion of links incident upon a node (i.e., the proportion of intersections connected to the given intersection).
Simultaneous Demand Prediction and Planning for Electric Vehicle Chargers

Fig. 2. Domain analysis between cities.

### Embedded Features

\[
\text{MMD}(SC, TC) = \left\| \frac{1}{m_s} \sum_{i=1}^{m_s} \phi(s_i) - \frac{1}{m_t} \sum_{j=1}^{m_t} \phi(t_j) \right\|_H^2,
\]

where \(s_i\) and \(t_j\) are training samples from the source city and target city, \(m_s\) and \(m_t\) are the numbers of training samples, and \(\phi(\cdot)\) is the kernel function.

We estimate the MMD for three cities, Beijing (BJ), Guangzhou (GZ), and Tianjin (TJ) in China, as shown in Figure 2(a). The solid black lines in Figure 2(a) are the rejecting thresholds for the null hypothesis test with power \(\delta = 0.05\). For all the three city pairs, the MMD results are much larger than the threshold, confirming that there exists a domain shift problem. Furthermore, we use the TSNE visualization \([24]\) to show the feature distributions of three cities, which reduces the feature dimension to 2. As shown in Figure 2(b), Beijing and Guangzhou have more similar feature distribution, probably because they have closer city scale, EV development level and strategy (they develop EVs earlier and deploy more slow chargers, as shown later in Table 3). By contrast, there is a larger feature difference between Tianjin and the other two cities, and the corresponding MMD values are also larger, probably because Tianjin develops EVs later and has a more different deployment strategy (more fast chargers). In summary, both MMD and TSNE results demonstrate the necessity of designing a city domain adaptation approach to address the domain shift issue.

### 3.2 Attention-based Spatial-Temporal City Domain Adaptation Network

Figure 3 shows the architecture of AST-CDAN, consisting of four components: (1) ContextNet integrates convolution and spatial attention to model the influences from context features; (2) ProfileNet learns latent features from the raw profile features by fully connected layers; (3) DemandNet is fed into the concatenation of outputs from ContextNet and ProfileNet, and integrates the temporal information to predict charging demands over different time intervals of one day; (4) DomainNet guides the network to promote the features from ProfileNet and ContextNet to deeper domain-invariant representations for domain adaptation. For convenience, let \(S_{SC}\) and \(S_{TC}\) denote the sets of training instances from source city and target city, respectively.

#### 3.2.1 ContextNet \(G_c\)

It takes a feature map \(F_c \in \mathbb{R}^{\lambda \times d}\) as input, which contains context features from \(\lambda\) stations (itself and \(\lambda - 1\) nearest neighbor stations); \(d\) is the dimension of context features. We employ convolutional blocks to model the effects of context features. Each convolutional block contains one convolution layer, one batch normalization layer, and one ReLU activation function:

\[
F^\text{out}_c = \text{ReLU}(BN(W_c \ast F^\text{in}_c + b_c)),
\]

ACM Transactions on Knowledge Discovery from Data, Vol. 17, No. 4, Article 60. Publication date: February 2023.
where $W_c$ and $b_c$ are learnable parameters, $\ast$ represents the convolutional operation, and $BN$ means batch normalization [14]. To prevent overfitting, dropout [31] is applied after the first convolutional block. Note that, different context features have inconsistent importance to the charging demands. Therefore, we further use the spatial-wise attention model (SAM) [8] to encode the spatial dependencies, the details of which are illustrated in Figure 4.

The spatial attention block takes $E$ as input to three $1 \times 1$ convolutional layers. After the reshape (and transpose) operations, we get three vectors $M_1 \in \mathbb{R}^{HW \times 1}$, $M_2 \in \mathbb{R}^{1 \times HW}$ and $M_3 \in \mathbb{R}^{HW \times 1}$. $M_1$ and $M_2$ go through the matrix multiplication and softmax operations to get the spatial attention map $M_a \in \mathbb{R}^{HW \times HW}$. Then, we apply a matrix multiplication for $M_a$ and $M_3$, and reshape the output back to the size of $H \times W$. After one convolutional layer, we sum $E$ and the output to get $E_a$, which captures the effects of the contextual information on the original feature map. This process can be formulated as

$$M_{ai} = \frac{\exp(M_1^i \cdot M_2^j)}{\sum_{i=1}^{HW} \exp(M_1^i \cdot M_2^j)}, \quad (8)$$

$$E_a = W_a \ast \text{vec}_{H,W}^{-1}(M_a \times M_3) + b_a + E, \quad (9)$$

where $W_a$ and $b_a$ are learnable parameters, $\ast$ represents the convolution operation, $\text{vec}_{H,W}^{-1}$ means reshaping vector to matrix in shape of $H \times W$, and $M_{ai}^j$ means the influence of the value in the $i$th position on the value in the $j$th position.

The output of SAM is fed into the second convolutional block to enhance the performance. Finally, we apply the global average pooling operation on the output to get the final context feature $f_c$.

ACM Transactions on Knowledge Discovery from Data, Vol. 17, No. 4, Article 60. Publication date: February 2023.
3.2.2 ProfileNet $G_p$. It takes the raw profile features $F_p$ as input, and utilizes two fully connected layers, each with a ReLU activation function. After that, we get the station profile feature $f_p$. The context feature $f_c$ and the profile feature $f_p$ are concatenated to obtain the final station feature $f$, which will be fed into the DemandNet and the DomainNet simultaneously.

3.2.3 DemandNet $G_y$. The DemandNet aims at predicting the charging demand in each time interval. We use an embedding layer to transform the time into a vector $q$. Meanwhile, the feature $f$ is fed into two fully connected layers, and the output is concatenated with $q$ to get the hidden feature $H_y$. Finally, we use one fully-connected layer to get the predicted demand $\hat{y}$.

A direct method to optimize the DemandNet is to minimize the regression loss over $S_{SC}$. Inspired by [21, 29], we find that the ranking loss is beneficial to enhance the regression prediction accuracy (Section 5.2). Thus, we combine the regression loss and the ranking loss for the DemandNet, using a hyperparameter $\alpha$:

$$L_{demand} = (1 - \alpha)L_{reg} + \alpha L_{rank},$$

where $L_{reg}$ is the mean square error between the predicted value $\hat{y}$ and the ground truth $y$ in $S_{SC}$:

$$L_{reg} = \frac{1}{|S_{SC}|} \sum_{X \in S_{SC}} (\hat{y} - y)^2.$$  

We define $o_{ij} = y_i - y_j$ for the instance $i$ and $j$, which satisfy $y_i > y_j$. Thus, the probability that instance $i$ is listed higher than $j$ can be defined as $P_{ij} = \frac{e^{o_{ij}}}{1 + e^{o_{ij}}}$. Likewise, the predicted probability is $\hat{P}_{ij}$. Thus, we can use the cross entropy function to define the $L_{rank}$

$$L_{rank} = \sum_{i, j: i \neq j} -P_{ij} \log \hat{P}_{ij} - (1 - P_{ij}) \log (1 - \hat{P}_{ij})$$

3.2.4 DomainNet $G_d$. One way to solve the domain shift problem is to map the feature spaces of the source and target cities to the same space. Inspired by the previous study [21], we introduce the domain adaptation network to AST-CDAN. The DomainNet takes $f$ as input and outputs a domain label that indicates which domain the feature belongs to. It contains two fully connected layers

$$H_d^1 = \text{ReLU} \left(W_d^1 f + b_d^1 \right),$$

$$\hat{d} = \text{softmax} \left( W_d^2 H_d^1 + b_d^2 \right),$$

where $W_d^1$ and $W_d^2$ are the weight parameters, $b_d^1$ and $b_d^2$ are the bias parameters, and $\hat{d}$ is the predicted domain label.

We use the binary cross-entropy loss $L_{domain}$ to optimize the domain discrimination component

$$L_{domain} = \frac{1}{|S|} \sum_{X \in S} -d \log \hat{d} - (1 - d) \log (1 - \hat{d}),$$

where $d$ is the domain label and $S = S_{SC} \cup S_{TC}$.

3.2.5 Optimization. Based on the above components, we design the joint loss function composed by $L_{reg}$, $L_{rank}$ and $L_{domain}$. The DemandNet needs to minimize $L_{reg}$ and $L_{rank}$ to improve the demand prediction performance. The DomainNet needs to minimize $L_{domain}$ for the domain classification. However, the ContextNet and ProfileNet aim at minimizing $L_{reg}$ and $L_{rank}$ while maximizing $L_{domain}$, because their goal is to produce domain-invariant feature representation that is indistinguishable across domains. The optimization of the above components can be done with
the following gradient updates:

$$\theta_s = \theta_s - \gamma \left( \alpha \frac{\partial L_{reg}}{\partial \theta_s} + (1-\alpha) \frac{\partial L_{rank}}{\partial \theta_s} - \beta \frac{\partial L_{domain}}{\partial \theta_s} \right), \quad (16)$$

$$\theta_y = \theta_y - \gamma \left( \alpha \frac{\partial L_{reg}}{\partial \theta_y} + (1-\alpha) \frac{\partial L_{rank}}{\partial \theta_y} \right), \quad (17)$$

$$\theta_d = \theta_d - \gamma \frac{\partial L_{domain}}{\partial \theta_d}, \quad (18)$$

where $\theta_s$ are parameters of ContextNet and ProfileNet; $\theta_y$ are parameters of DemandNet; and $\theta_d$ are parameters of DomainNet.

In Equation (16), the gradients of $L_{reg}$, $L_{rank}$, and $L_{domain}$ are subtracted, which is different with summation in normal stochastic gradient descent (SGD) updates. Accordingly, we add the gradient reversal layer [7] before the DomainNet, which multiples the gradient from the DomainNet by $-\beta$ during backward propagation. As a result, the joint loss function is defined as

$$L = (1-\alpha)L_{reg} + \alpha L_{rank} - \beta L_{domain}. \quad (19)$$

## 4 CHARGER PLANNING

In this section, we first present the TIO algorithm and then elaborate how to fine-tune the charger plan, followed by the algorithm complexity analysis.

### 4.1 Transfer Iterative Optimization

Hindered by the unacceptable complexity $\binom{B+2|\mathcal{C}_c|-1}{2|\mathcal{C}_c|-1}$ of the straightforward approach in Section 2.2, we adopt a heuristic strategy. Generally, TIO starts from a naive charger plan and iteratively fine-tunes the current charger plan toward a higher revenue with a local search mechanism. In each iteration, we scale down the complexity by (1) decomposing the whole searching space into a small-scale collection of 5-element fine-tuned charger plan sets, and (2) only training the AST-CDAN once. In this way, the time complexity is proportional to the required number of iterations with a constant upper bound (Theorem 3).

Specifically, given a charger plan $\mathcal{N}_tc$ in the target city, where station $c_i$’s charger plan is $(n_i^s, n_i^f)$, the fine-tuned charger plans are obtained as follows: (1) extending $c_i$’s charger plan to a fine-tuned charger plan set $N_i = \{(n_i^s, n_i^f), (n_i^s + 1, n_i^f), (n_i^s - 1, n_i^f), (n_i^5, n_i^f + 1), (n_i^5, n_i^f - 1)\}$, and (2) obtaining the collection of fine-tuned charger plan sets in the target city as $\mathcal{N}_{tc}' = \{N_i | i = 1, \ldots, |\mathcal{C}_c|\}$. In this way, $5^{|\mathcal{C}_c|}$ new plans could be constructed from $\mathcal{N}_{tc}'$. If we re-train the AST-CDAN and predict the charging demand for each plan, it will require to respectively conduct model training and prediction operations for $5^{|\mathcal{C}_c|}$ times, which is still impractical. To address this issue, we further adopt two strategies to reduce the time complexity:

(1) The AST-CDAN is trained only once in each iteration, taking the current plan $\mathcal{N}_{tc}$ as the input. The features for fine-tuned plans only have slight difference on number of chargers, compared with the current plan, implying the versatility of the AST-CDAN trained with $\mathcal{N}_{tc}$. In each iteration, this strategy reduces the number of trainings from $5^{|\mathcal{C}_c|}$ to 1.

(2) For each fine-tuned plan of any station $c_i$, we fix the features of the nearby stations the same as those extracted from the current plan $\mathcal{N}_{tc}$, and only use the new features of station $c_i$ to feed into the model trained with $\mathcal{N}_{tc}$, outputting the demand prediction results. The features of nearby stations only have slight difference for those fine-tuned plans. This feature maintenance affects prediction results by 0.82%, 0.22%, and 1.46% in transfer cases of BJ→GZ, BJ→TJ, and GZ→TJ.
ALGORITHM 1: Transfer Iterative Optimization (TIO)

input: $N_{SC}, D_{SC}, D_{TC}, Y_{SC}, B$
output: $N_{TC}$

1. Initialize the revenue: $R \leftarrow 0$;
2. Construct an initial plan $N_{TC}$ by evenly allocating budget $B$ to each candidate station and charger type;
3. while True do
   1. $\tilde{Y}_{TC} \leftarrow f(N_{SC}, N_{TC}, D_{SC}, D_{TC}, Y_{SC})$;
   2. $R_{TC} \leftarrow \sum_{i=1}^{[TC]} \sum_{t=1}^{T} (\tilde{y}_{i}^{S} p_{i}^{S} n_{i}^{S} + \tilde{y}_{i}^{F} p_{i}^{F} n_{i}^{F})$;
   3. if $R_{TC} - R \leq \theta$ then
      1. return $N_{TC}$
   4. $R \leftarrow R_{TC};$
   5. Build the fine-tuned plan sets $N'_{TC}$ from $N_{TC}$;
   6. $\tilde{Y}'_{TC} \leftarrow f'(f(N_{SC}, \cdot, D_{SC}, D_{TC}, Y_{SC}), N_{TC}, N'_{TC})$;
   7. $N_{TC} \leftarrow DP-MK(N'_{TC}, \tilde{Y}'_{TC}, B)$;

respectively. Thus, in each iteration, this strategy reduces the number of prediction operations from $5^{[TC]}$ to 5, while ensuring almost the same prediction accuracy.

For convenience, the predicted demands for any fine-tuned plan set $N_{i}$ are denoted by $\tilde{Y}_{i} = \{(\tilde{y}_{ijt}^{S}, \tilde{y}_{ijt}^{F})|i = 1, \ldots, 5, j = 1, \ldots, 6, t = 1, \ldots, T\}$ and the predicted demands for all the fine-tuned plan sets in all the stations are denoted by $\tilde{Y}'_{TC} = \{\tilde{Y}_{i}|i = 1, \ldots, |TC_{i}|\}$. The simplified prediction operation $f'$ is defined as

$$\tilde{Y}'_{TC} = f'(f(N_{SC}, \cdot, D_{SC}, D_{TC}, Y_{SC}), N_{TC}, N'_{TC}),$$

where $f(N_{SC}, \cdot, D_{SC}, D_{TC}, Y_{SC})$ is a predictor trained with $N_{TC}$ and used for outputting the prediction results for any fine-tuned plan in $N'_{TC}$. By now, we have obtained $N'_{TC}$ and $\tilde{Y}'_{TC}$, so the remaining problem is how to update $N_{TC}$ by selecting a charger plan from $N_{i}$ for each station $c_{i}$, so that the total revenue is maximized under the budget constraint $B$, as we will elaborate in Section 4.2. Note that, there are biases in predicted demands, caused by the drift of data between training and prediction. Therefore, the updated plan will be confirmed by retraining AST-CDAN and prediction again, which will further determine whether to stop the TIO algorithm.

Algorithm 1 shows the pseudocode of the TIO algorithm, which operates with five main steps:

1. Initialize the revenue and construct a naive charger plan by evenly allocating budget to each charger type of each candidate station, as illustrated in Figure 5(a) (lines 1–2).
2. Train the AST-CDAN model with the current charger plan $N_{TC}$ to predict the demands $\tilde{Y}_{TC}$, and compute the revenue $R_{TC}$; if the increased revenue is not greater than a threshold $\theta$, then return the current plan (lines 4–8).
3. Extend the current plan $N_{TC}$ to the collection of fine-tuned plan sets $N'_{TC}$ (line 9), as illustrated in Figure 5(b).
4. Predict the charging demands $\tilde{Y}'_{TC}$ for the fine-tuned plan sets $N'_{TC}$ (line 10).
5. Invoke the DP-MK algorithm (Algorithm 2) to update the current plan $N_{TC}$ (line 11), as illustrated in Figure 5(b); then go to step 2.)
4.2 Charger Plan Fine-tuning

For convenience, let \( N_i = \{(n^S_{ij}, n^F_{ij}) | j = 1, \ldots, 5\} \) denote the fine-tuned charger plan set of station \( c_i \). To optimize the plan \( N_{TC} \) toward higher revenue, it is required to solve the charger plan fine-tuning problem: given the collection of fine-tuned charger plan sets \( N_i'_{TC} \), the predicted demands \( \hat{Y}^i_{TC} \) and the budget constraint \( B \), the objective is to select one plan \((n^S_{ij}, n^F_{ij})\) from \( N_i \) for each station \( c_i \), so that the total revenue is maximized while the total deployment cost of chargers does not exceed \( B \). In essence, the problem is an instance of the Multiple-choice Knapsack (MK) problem [26], formulated as follows:

\[
\max_{v_{ij}} \sum_{i=1}^{\left| N_{TC} \right|} \sum_{j=1}^{5} \sum_{t=1}^{T} (y^S_{ijt} \cdot p^S_{it} \cdot n^S_{ij} \cdot v_{ij} + y^F_{ijt} \cdot p^F_{it} \cdot n^F_{ij} \cdot v_{ij}),
\]

(21a)

\[
s.t. \sum_{i=1}^{\left| N_{TC} \right|} \sum_{j=1}^{5} c^S_{ij} \cdot n^S_{ij} \cdot v_{ij} + c^F_{ij} \cdot n^F_{ij} \cdot v_{ij} \leq B,
\]

(21b)

\[
\sum_{j=1}^{5} v_{ij} = 1, \quad i = 1, \ldots, \left| N_{TC} \right|,
\]

(21c)

\[
v_{ij} \in \{0, 1\}, \quad i = 1, \ldots, \left| N_{TC} \right|, j = 1, \ldots, 5,
\]

(21d)

where \( v_{ij} \) is a binary decision variable, representing whether to choose the \( j \)th fine-tuned plan \((n^S_{ij}, n^F_{ij})\) for station \( c_i \). The MK problem has been proven to be NP-complete, and it was pointed out that the dynamic programming approach performs well for a relatively small-scale problem [26]. Moreover, branch and bound algorithms with different relaxations could be used for providing approximate solutions while greatly reducing the time complexity [26]. In this work, we use a dynamic programming algorithm \( DP-MK \) to obtain the optimal solution with the time complexity of \( O(|N_{TC}|B) \).

Algorithm 2 shows the pseudocode of \( DP-MK \) algorithm, where

- \( W[i][j] \) is the cost of the \( j \)th fine-tuned plan of the \( i \)th station;
- \( V[i][j] \) is the daily revenue of the \( j \)th fine-tuned plan of the \( i \)th station;
- \( R[i][k] \) is the maximum revenue under the budget of \( k \), considering only the first \( i \) stations;
ALGORITHM 2: DP-MK

input: \( N'_{TC}, Y'_{TC}, B \)
output: \( N_{TC} \)

1. for \( i = 1, \ldots, |C_{TC}| \) do
   2. for \( j = 1, \ldots, 5 \) do
      3. \( W[i][j] \leftarrow e_i^S n_i^S + e_i^F n_i^F \);
      4. \( V[i][j] \leftarrow \sum_{t=1}^{T} (y_{ijt}^S p_{ijt}^S n_{ijt}^S + y_{ijt}^F p_{ijt}^F n_{ijt}^F) \);

5. for \( i = 0, 1, \ldots, |C_{TC}| \) do
   6. for \( B = 0, 1, \ldots, B \) do
      7. \( R[i][k] \leftarrow 0; \)
      8. \( S[i][k] \leftarrow \) an empty list;
   9. for \( i = 1, 2, \ldots, |C_{TC}| \) do
      10. for \( j = 1, 2, \ldots, 5 \) do
          11. for \( k = W[i][j], W[i][j] + 1, \ldots, B \) do
              12. if \( R[i][k] < R[i-1][k-W[i][j]] + V[i][j] \) then
                  13. \( R[i][k] \leftarrow R[i-1][k-W[i][j]] + V[i][j]; \)
                  14. \( S[i][k] \leftarrow S[i-1][k-W[i][j]]; \)
                  15. Append \( j \) to the tail of \( S[i][k]; \)

16. return \( S[|C_{TC}|][\text{argmax}_k R[|C_{TC}|][k]] \);

\(- S[i][k] \) records the optimal selection for the maximum revenue under the budget of \( k \), considering only the first \( i \) stations.

4.3 Algorithm Complexity Analysis

As previously mentioned, the time complexity of the TIO algorithm is proportional to the required number of iterations, with a constant upper bound as follows:

**Theorem 3.** The required number of iterations for the TIO algorithm has an upper bound

\[
\frac{B}{\theta} \times \max_i \left( \frac{\sum_{t=1}^{T} p_{it}^S}{e_i^S}, \frac{\sum_{t=1}^{T} p_{it}^F}{e_i^F} \right).
\]

**Proof.** Let \( u = \max_i \left( \frac{\sum_{t=1}^{T} p_{it}^S}{e_i^S} \right) \) and \( v = \max_i \left( \frac{\sum_{t=1}^{T} p_{it}^F}{e_i^F} \right) \) denote the maximum revenues that per unit cost can produce by slow chargers and fast chargers among all the stations \( C_{TC} \). Further, let \( w = \max(u, v) \) denote the maximum revenue that per unit cost can produce among any charger and any \( c_i \in C_{TC} \). There is an upper bound of revenue \( R: \frac{B}{w} \). Since the increased revenue is at least \( \theta \) in each iteration, there is an upper bound of the number of iterations: \[
\frac{B}{w} = \frac{B}{\theta} \times \max_i \left( \frac{\sum_{t=1}^{T} p_{it}^S}{e_i^S}, \frac{\sum_{t=1}^{T} p_{it}^F}{e_i^F} \right). \]

Then, we analyze the time complexity of the DP-MK algorithm. The number of loops in line 9 is \(|C_{TC}|\); the number of loops in line 10 is 5; the number of loops in line 11 is \( B \) (at most). As a result, the time complexity of DP-MK algorithm is \( O(|C_{TC}|B) \).

Finally, we analyze the time complexities of the TIO algorithm where four steps are processed in each iteration: the time complexity for model training in step (2) is a constant, about 1 hour; that in steps (3) and (4) is \( O(|C_{TC}|) \); and that in step (5) is \( O(|C_{TC}|B) \). There are at most \( O(B) \) iterations. Thus, the total time complexity of the TIO algorithm is \( O(|C_{TC}|B^2) \).
Table 3. Details of the Datasets

| City          | Beijing | Guangzhou | Tianjin |
|---------------|---------|-----------|---------|
| # of charging stations | 138     | 123       | 101     |
| # of slow chargers      | 1473    | 1434      | 273     |
| # of fast chargers       | 733     | 608       | 551     |
| # of POIs               | 576,726 | 503,920   | 362,160 |
| # of transportation facilities | 251,758 | 211,756   | 155,996 |
| # of roads              | 841     | 726       | 651     |

5 EXPERIMENTAL EVALUATION

5.1 Experimental Settings

Datasets. We collected the charging station data, including the locations, number of slow/fast chargers, service prices, and historical charging demands, from a public EV charging platform Star Charge,\(^4\) which has the highest monthly usage in Chinese public EV charging market. Meanwhile, we collected the POI and transportation data from AutoNavi,\(^5\) and collected the road network data from OpenStreetMap.\(^6\) All the data are from three cities, Beijing, Guangzhou, and Tianjin in China, and the charging demands are recorded during 8:00–21:00 every day from 05/12/2019 to 15/01/2020. Table 3 shows the dataset details in each city. In addition, according to China’s charging pile industry report \(^1\), we set \(e_s^i\) and \(e_f^i\) of each station as 33,000 and 54,000 in RMB. The radius \(r\) used for feature extraction is set to 1 km.

We mainly consider three cross-city prediction/planning tasks, BJ → GZ, BJ → TJ, and GZ → TJ, which is in line with the development order and level of EV charging stations in three cities.

All the experiments are run in a Linux server (CPU: E5-2620 v4 @ 2.10GHz, GPU: NVIDIA Tesla P100). For the AST-CDAN, we use Pytorch to build it, and set \(\alpha \in \{0, 0.3, 0.5, 0.8, 1.0\}\); \(\beta = 0.1\); the batch size \(bs = 64\); the learning rate \(lr \in \{0.01, 0.005, 0.001, 0.0005, 0.0001\}\). For the TIO, we set \(\theta = 0.1\), \(u^S = 40\), and \(u^F = 20\).

5.2 Evaluation on Charger Demand Prediction

Baselines. We compare our AST-CDAN with three baselines:

- LASSO (Least Absolute Shrinkage and Selection Operator), a well-known linear regression method that performs both variable selection and regularization to enhance the prediction accuracy;
- GBRT (Gradient Boost Regression Tree), a boosting method based on decision tree that can deal with heterogeneous data and has been widely used in many data mining tasks;
- MLP (Multi-layer Perceptron), a feedforward deep neural network with four full-connected layers and one ReLU activation function.

Variants. We also compare our AST-CDAN with three variants:

- AST-CDAN/AP, which removes both the spatial attention and the ProfileNet from AST-CDAN;
- AST-CDAN/P, which removes the ProfileNet from AST-CDAN;
- AST-CDAN/D, which removes the DomainNet from AST-CDAN.

\(^4\)https://en.starcharge.com.
\(^5\)https://amap.com.
\(^6\)https://www.openstreetmap.org.
Table 4. Comparison Results for Charger Demand Prediction

| Metric | BJ → GZ | BJ → TJ | GZ → TJ |
|--------|---------|---------|---------|
|        | Slow    | Fast    | Slow    | Fast    | Slow    | Fast    |
| LASSO  | 0.3052  | 0.2935  | 0.2999  | 0.3527  | 0.2995  | 0.3666  |
| GBRM   | 0.3077  | 0.2919  | 0.3378  | 0.3321  | 0.3032  | 0.3429  |
| MLP    | 0.3010  | 0.2914  | 0.3435  | 0.2939  | 0.3560  |
| AST-CDAN | **0.2834** | **0.2860** | **0.2584** | **0.3234** | **0.2401** | **0.3264** |

The performance gain is achieved by our AST-CDAN compared with the best baseline (underlined). The bold values are the best results among all methods. The underlined values are the best results among all methods except ours.

**Metric.** One widely used metric, **Root Mean Square Error (RMSE)**, is adopted to evaluate the prediction performance.

**Performance Comparisons.** Table 4 compares our AST-CDAN with the three baselines for slow and fast charger demands in three city pairs. We observe that: (1) the deep learning methods (MLP and AST-CDAN) are superior to traditional regression methods (LASSO and GBRM), demonstrating the advantages of capturing non-linear correlations between features and charging demands; (2) our AST-CDAN performs best due to the added ability of domain adaptation; (3) our AST-CDAN has more gains in BJ → TJ and GZ → TJ, indicating its bigger superiority when the target city (Tianjin) has a more different feature distribution (as analyzed in Section 3.1).

**Effect of Each Component.** Figure 6(a) compares our AST-CDAN with its three variants for the overall prediction results in three city pairs. We observe that: (1) AST-CDAN/P outperforms AST-CDAN/AP, indicating the important role of the spatial attention on capturing contextual information; (2) AST-CDAN outperforms AST-CDAN/P, indicating the necessity of modeling profile features; (3) The RMSE of AST-CDAN/D increases significantly compared with AST-CDAN, implying a great negative influence of the domain shift problem, while AST-CDAN has an obvious advantage on addressing the domain shift problem. To further illustrate whether DomainNet can learn domain-invariant feature representations, we show the TSNE visualization results of feature distributions for AST-CDAN with and without DomainNet respectively in Figure 7. It is obvious to see that the feature distribution becomes more consistent between source city (BJ) and target city (TJ) with DomainNet. We also get a lower MMD value with DomainNet (0.0575) than that without DomainNet (0.9612). In addition, Figure 6(b) shows the effect of ranking loss weight $\alpha$. The optimal
results are always achieved when $\alpha$ is equal to some intermediate value, implying that the ranking loss can help to enhance the prediction accuracy.

5.3 Evaluation on Charger Planning

Baselines. We compare our TIO algorithm with four baselines:

- **Parking as Proxy (Park)**, which follows the work [3] to take parking sessions as the proxy of charging demands. We build the Voronoi diagram by taking charging stations as seeds and aggregate parking sessions\(^7\) in each divided region as the charging demand. Then the budget is allocated to each charging station proportionally to the charging demand.

- **Population as Proxy (Pop)**, which follows the work [37] to estimate the charging demand in proportion to the population of the region to which the charging station belongs. Population as Proxy has the same process as Parking as Proxy except that the population\(^8\) is used for estimation.

- **Even**, which is a naive solution by evenly allocating the budget to each charger type of each charging station.

- **Charger-based Greedy (CG) algorithm** [5], which assumes that the charging demands of all the stations are already known, and greedily places the charger in the candidate station with the maximum increased demand reward. In our experiments, we use the historical charging demands in the real world as inputs, although it is impractical in a new city.

We also compare the algorithms with the real-world EV charger plans (named as “Real”) that have been deployed in the three cities. Specifically, we compute the total cost that is required to deploy the real plan, and use it as the budget to determine charger plans with different algorithms for performance comparisons.

Metrics. We compare all the algorithms in terms of daytime revenue (during 8:00–21:00 of one day). Besides, we evaluate the time complexity in terms of \# of trainings.

Performance Comparisons with Real Plans. As shown in Figure 8(a), TIO outperforms other baselines and achieves 72.5%, 7.9%, and 6.7% revenue increment comparing with the real plans in BJ→GZ, BJ→TJ, and GZ→TJ cases, respectively. The increment is smaller in Tianjin than that in Guangzhou, because (1) the lately deployed plan in Tianjin has a higher average utilization rate (45%) than the early deployed plan in Guangzhou (31%), and (2) the deployment scale and the used

\(^7\)The parking session data are provided by www.soargift.com.

\(^8\)The population map data are provided by www.worldpop.org.
Fig. 8. Comparison results for charger planning.

Analysis on Charging Demand Proxies. To inspect the representativeness of alternative proxies for charging demands, we compare the distributions of charging stations and two proxies as shown in Figure 9. We observe that: (1) Parking lots have different spatial distribution with charging stations. In fact, a city has a large number of parking plots belonging to different operators, so we could only obtain parking session data in a biased manner (e.g., mainly distributed in the urban centers in Figure 9). Even if we could collect the comprehensive parking session data, they may still have very different spatial-temporal patterns because chargers are not so ubiquitous as parking lots particularly when the EV market share is still small. (2) The population distribution is wider than that of charging stations, which will bring errors to the estimation method where population is allocated to the nearest charging station. Compared with the general population distribution, early EV adopters are disproportionately younger, male, more educated, and more
sensitive to environmental concerns [3]. In summary, such implicit data have so different distributions with charging demands in nature that they are inappropriate to represent EV charging demands for charger planning.

**Performance Comparisons with Varied Budgets.** From Figures 8(b)–(d), we observe (1) the revenues achieved by all the algorithms increase with the budget; (2) our TIO achieves the highest revenue under all the cases, and its advantage is more obvious as the budget increases, indicating that TIO is able to utilize the budget more efficiently on those chargers with higher demands; (3) CG performs better than Even in Tianjin, but the results are just the opposite in Guangzhou when there is a big budget (> ¥20 million), due to the same reasons as explained before. In addition, we want to emphasize that, our TIO applies to various city-pair cases, while CG is unpractical in a new city due to lack of historical demand data before the actual deployment.

**Comparison with the Optimal Solution.** Since TIO is a heuristic solution, we are interested to know its effectiveness and efficiency compared with the optimal solution. Nevertheless, with the large search space analyzed in Section 2.2, the optimal algorithm is unpractical. Thus, we select at most six candidate stations in the central area of Tianjin and small budgets (≤ 15, meanwhile we proportionally set $e^B_i = 2$ and $e^F_i = 3$) for experiments. Figure 10(a) compares the results with varied $B$ when $|CTC| = 4$, and Figure 10(b) compares the results with varied $|CTC|$ when $B = 12$. We observe that the revenue achieved by TIO is very close to the optimal solution. However, the
Simultaneous Demand Prediction and Planning for Electric Vehicle Chargers

Fig. 10. Comparison with the optimal solution (BJ→TJ).

Fig. 11. Scalability of TIO.

required number of trainings by the optimal solution dramatically increases with $B$ and $|C_{TC}|$, up to 2,796 when $B = 15$ and $|C_{TC}| = 4$. By contrast, our TIO only needs at most four trainings.

**Time Efficiency.** We demonstrate the scalability of TIO in Figure 11. Recall that the budget and number of candidate stations affect the time complexity of the TIO algorithm. The experimental results show that the time consumption grows with the budget and number of candidate stations. The underlying reason is that with a larger $B$ the TIO algorithm will need more iteration times for convergence. With a larger $|C_{TC}|$, the TIO will consume more time to train the prediction model. We also evaluate the time cost with the real budgets and numbers of candidate stations. The TIO at most consumes 2.7 hours with nine iterations in BJ→GZ; 2.14 hours with eight iterations in BJ→TJ; 3.34 hours with 13 iterations in GZ→TJ, which is completely acceptable in reality.

6 DISCUSSION

In spite of many merits for our SPAP, some possible limitations are still worthy of discussions or further research in the future, summarized as follows:

**Cross-city Prediction.** Demand prediction is a challenging task in a new city where no explicit historical data are available. Although we have designed the AST-CDAN model for addressing the domain shift problem, the performance may be degraded specially when source city and target city have quite different characteristics (e.g., city scale, development level, and strategy), or source city has low demand diversity. We plan to tackle this challenge by learning from more source cities to enhance the generalization ability of the transfer learning model.
Cross-city Planning. The TIO algorithm adopts a heuristic idea without strict guarantee on the optimality. Nevertheless, it is still promising because (1) the TIO algorithm is at least better than any naive method (e.g., “even”), by taking the naive method as the initial plan and iteratively optimizing it; (2) it consistently outperforms various existing charger planning methods; and (3) the achieved performance is very close to the optimal solution, which has been verified by extensive experiments. In the future, other solutions with a solid theoretical guarantee are worth investigating, while this work can provide important insights as a starting point. The TIO algorithm has the potential to apply to the unidirectional Vehicle-to-Grid (V2G) context, where the charging rate of each EV is throttled to prevent grid overloading, system instability, and voltage drop issues, resulting in a different charging service price. The TIO algorithm can help to optimize the charger deployment plan in the V2G context by considering the dynamic service prices in the problem definition.

Long-term planning. Given that the EV market is still young, one would need much more data before coming to conclusion on how to construct the whole charging station network. It could be wise to place chargers in phases, which is also consistent with the gradual development mode commonly adopted by charging station operators in reality. As one collects data and learns more, the chargers could be placed in other locations in multiple phases or use dynamic pricing as a complement. Guided by that, this work is committed to solving the cold-start problem in the first phase. Dynamic urban macro factors, e.g., newly built infrastructure in the future, will influence charging demands of the related regions, which should be considered in long-term construction. Nevertheless, static urban factors used in this work are sufficient for planning in the first phase, whose target is to find a subset of candidate locations with the highest utility in the current phase.

7 RELATED WORK

7.1 Charger Demand Modeling and Prediction

The related work on charger demand modeling and prediction can be classified into two categories based on the used data type.

Implicit Data. A traditional way is to infer charging demands by leveraging relevant implicit information [3, 37]. Chen et al. [3] use the parking demand as a proxy to estimate the charging demand. Xiong et al. [37] use the population distribution to estimate the charging demand. Liu et al. [20] assume that the charging demand is proportional to the traffic flow. Liu et al. [19] leverage the refueling demand to define the charging demand. Unfortunately, such indirect method is error-prone due to the dissimilar nature of different spatio-temporal mobility patterns. In other words, the implicit data have intrinsic defects for charging demand prediction.

Explicit Data. Recently, the advanced data acquisition technologies enable us to collect explicit data about charging events of EVs, which helps in charger planning [5, 9, 18, 23, 33]. Li et al. [18] extract charging demands from the seeking sub-trajectories of EV taxis. Du et al. [5] use the return records of an EV sharing platform as the charging demand. These data sources are only limited to commercial EVs rather than private EVs. For the general charging stations except for those that are used exclusively for commercial EVs, the only available explicit data are their charger transaction records [9], whereas it is impossible in a new city.

7.2 Charger Planning

Existing work on charger planning mainly falls into two categories. In the first category, all charger demands are required to be fulfilled to maximize the social welfare [15, 18, 22, 37]. For example, Li et al. [18] minimize the average seeking and waiting time of all charging demands based on taxi trajectory data. The second category takes charging demand as objectives [5, 6, 17]. For example,
Du et al. [5] use both coverage and charging demand as the optimization objective. Our work takes charging demands as part of the objective. However, charging demands are affected by both the station profile and nearby stations, which is ignored by the existing work. Moreover, we are the first to conduct simultaneous demand prediction and planning in a new city.

7.3 Urban Transfer Learning

Recently, urban transfer learning [4, 10, 11, 16, 21, 34–36] has emerged to be an effective paradigm for solving urban computing problems [40] by applying transfer learning approaches [28]. Wei et al. [36] tackle the label scarcity and data insufficiency problems. Katranji et al. [16] predict the Home-to-Work time for families in a new city using survey data of families in both source and target cities. Guo et al. [10] propose an SVD-based transfer method for chain store site recommendation in a new city. Wang et al. [34] propose a cross-city transfer learning method for deep spatio-temporal prediction tasks. Ding et al. [4] solve the problem of cross-city POI recommendation for the travelers by learning from users’ visiting behaviors in both their hometown and current city. However, these works need homogeneous data in the target domain, which is not satisfied in our problem, because there is not any historical charging data in the new city. On the other hand, the domain generalization technique [25] is leveraged to address the problem of label unavailability in the target domain [11, 21]. Liu et al. [21] detect the parking hotspots of the dockless shared bikes in a new city. He et al. [11] generate mobility data for a new target city. However, they have different problem settings from us, as we consider both cross-city demand prediction and station planning simultaneously.

8 CONCLUSIONS

In this article, we investigate an important problem of planning the charging station network in a new city. The concept of simultaneous demand prediction and planning is first proposed to address the deadlock between charger demand prediction and charger planning. We prove the NP-hardness of the problem and point out the unacceptable time complexity of a straightforward approach. We propose the SPAP solution by combining discriminative features extracted from multi-source data, an AST-CDAN model for knowledge transfer between cities, and a novel TIO algorithm for charger planning. Extensive experiments on real datasets from three cities validate the effectiveness and efficiency of SPAP. Moreover, SPAP improves at most 72.5% revenue compared with the real-world charger deployment. Our work also has potential implications for other infrastructure planning problems in a new city.

REFERENCES

[1] 2020. China’s public charging pile industry research report. (2020). Retrieved October 2022 from http://report.iresearch.cn/report/202006/3583.shtml.
[2] Rumen Andonov, Vincent Poirriez, and Sanjay Rajopadhye. 2000. Unbounded knapsack problem: Dynamic programming revisited. European Journal of Operational Research 123, 2 (2000), 394–407.
[3] T. Donna Chen, Kara M. Kockelman, and Moby Khan. 2013. Locating electric vehicle charging stations: Parking-based assignment method for Seattle, Washington. Transportation Research Record 2385, 1 (2013), 28–36.
[4] Jingtao Ding, Guanghui Yu, Yong Li, Depeng Jin, and Hui Gao. 2019. Learning from hometown and current city: Cross-city POI recommendation via interest drift and transfer learning. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies 3, 4 (2019), 1–28.
[5] Bowen Du, Yongxin Tong, Zimu Zhou, Qian Tao, and Wenjun Zhou. 2018. Demand-aware charger planning for electric vehicle sharing. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 1330–1338.
[6] Inês Frade, Anabela Ribeiro, Gonçalo Gonçalves, and António Pais Antunes. 2011. Optimal location of charging stations for electric vehicles in a neighborhood in Lisbon, Portugal. Transportation Research Record 2252, 1 (2011), 91–98.
[7] Yaroslav Ganin and Victor Lempitsky. 2015. Unsupervised domain adaptation by backpropagation. In Proceedings of the International Conference on Machine Learning. 1180–1189.

[8] Junyu Gao, Qi Wang, and Yuan Yuan. 2019. SCAR: Spatial-/channel-wise attention regression networks for crowd counting. Neurocomputing 363, C (2019), 1–8.

[9] Ragavendran Gopalakrishnan, Arpita Biswas, Alefiya Lightwala, Skanda Vasudevan, Partha Dutta, and Abhishek Tripathi. 2016. Demand prediction and placement optimization for electric vehicle charging stations. In Proceedings of the 25th International Joint Conference on Artificial Intelligence. 3117–3123.

[10] Bin Guo, Jing Li, Vincent W. Zheng, Zhuo Wang, and Zhiwen Yu. 2018. Citytransfer: Transferring inter-and intra-city knowledge for chain store site recommendation based on multi-source urban data. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies 1, 4 (2018), 1–23.

[11] Tianfu He, Jie Bao, Ruiyuan Li, Sijie Ruan, Yanhua Li, Li Song, Hui He, and Yu Zheng. 2020. What is the human mobility in a new city: Transfer mobility knowledge across cities. In Proceedings of The Web Conference 2020. 1355–1365.

[12] IEA. 2020. Global EV Outlook 2020. Retrieved October 2022 from https://www.iea.org/reports/global-ev-outlook-2020.

[13] IEA. 2021. Global EV Outlook 2021. Retrieved October 2022 from https://www.iea.org/reports/global-ev-outlook-2021.

[14] Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv:1502.03167. Retrieved from https://arxiv.org/abs/1502.03167.

[15] Long Jia, Zechun Hu, Yonghua Song, and Zhuowei Luo. 2012. Optimal siting and sizing of electric vehicle charging stations. In Proceedings of the 2012 IEEE International Electric Vehicle Conference. 1–6.

[16] Mehdi Katranji, Etienne Thuillier, Sami Kraiem, Laurent Molaic, and Fouad Hadji Selem. 2016. Mobility data disaggregation: A transfer learning approach. In Proceedings of the 2016 IEEE 19th International Conference on Intelligent Transportation Systems. 1672–1677.

[17] Albert Y. S. Lam, Yu-Wing Leung, and Xiaowen Chu. 2014. Electric vehicle charging station placement: Formulation, complexity, and solutions. IEEE Transactions on Smart Grid 5, 6 (2014), 2846–2856.

[18] Yanhua Li, Jun Luo, Chi-Yin Chow, Kam-Lam Chan, Ye Ding, and Fan Zhang. 2015. Growing the charging station network for electric vehicles with trajectory data analytics. In Proceedings of the 2015 IEEE 31st International Conference on Data Engineering. 1376–1387.

[19] Chen Liu, Ke Deng, Chaojie Li, Jianxin Li, Yanhua Li, and Jun Luo. 2016. The optimal distribution of electric-vehicle chargers across a city. In Proceedings of the 2016 IEEE 16th International Conference on Data Mining. 261–270.

[20] Qiuyu Liu, Yuxiang Zeng, Lei Chen, and Xiuren Zheng. 2019. Social-aware optimal electric vehicle charger deployment on road network. In Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems. 398–407.

[21] Zhaoyang Liu, Yanan Shen, and Yanmin Zhu. 2018. Where will dockless shared bikes be stacked? —Parking hotspots detection in a new city. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 566–575.

[22] Zi-fa Liu, Wei Zhang, Xing Ji, and Ke Li. 2012. Optimal planning of charging station for electric vehicle based on particle swarm optimization. In Proceedings of the IEEE PES Innovative Smart Grid Technologies. 1–5.

[23] Man Luo, Bowen Du, Konstantin Klemmer, Hongming Zhu, Hakan Ferhatosmanoglu, and Hongkai Wen. 2020. D3P: Data-driven demand prediction for fast expanding electric vehicle sharing systems. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies 4, 1 (2020), 1–21.

[24] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. Journal of Machine Learning Research 9 (2008), 2579–2605.

[25] Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. 2013. Domain generalization via invariant feature representation. In Proceedings of the International Conference on Machine Learning. 10–18.

[26] Robert M. Nauss. 1978. The 0–1 knapsack problem with multiple choice constraints. European Journal of Operational Research 2, 2 (1978), 125–131.

[27] Sinno Jialin Pan, Ivor W. Tsang, James T. Kwok, and Qiang Yang. 2010. Domain adaptation via transfer component analysis. IEEE Transactions on Neural Networks 22, 2 (2010), 199–210.

[28] Sinno Jialin Pan and Qiang Yang. 2009. A survey on transfer learning. IEEE Transactions on Knowledge and Data Engineering 22, 10 (2009), 1345–1359.

[29] David Sculley. 2010. Combined regression and ranking. In Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 979–988.

[30] Alex Smola, Arthur Gretton, Le Song, and Bernhard Schölkopf. 2007. A Hilbert space embedding for distributions. In Proceedings of the International Conference on Algorithmic Learning Theory. Springer, 13–31.

[31] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research 15, 56 (2014), 1929–1958.
[32] Kang Miao Tan, Vigna K. Ramachandaramurthy, and Jia Ying Yong. 2016. Integration of electric vehicles in smart grid: A review on vehicle to grid technologies and optimization techniques. Renewable and Sustainable Energy Reviews 53 (2016), 720–732. DOI: https://doi.org/10.1016/j.rser.2015.09.012

[33] Guang Wang, Xiaoyang Xie, Fan Zhang, Yunhuai Liu, and Desheng Zhang. 2018. bCharge: Data-driven real-time charging scheduling for large-scale electric bus fleets. In Proceedings of the 2018 IEEE Real-Time Systems Symposium. 45–55.

[34] Leye Wang, Xu Geng, Xiaojuan Ma, Feng Liu, and Qiang Yang. 2019. Cross-city transfer learning for deep spatio-temporal prediction. In Proceedings of the 28th International Joint Conference on Artificial Intelligence. 1893–1899.

[35] Leye Wang, Bin Guo, and Qiang Yang. 2018. Smart city development with urban transfer learning. IEEE Computer 51, 12 (2018), 32–41.

[36] Ying Wei, Yu Zheng, and Qiang Yang. 2016. Transfer knowledge between cities. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 1905–1914.

[37] Yanhai Xiong, Jiarui Gan, Bo An, Chunyan Miao, and Ana LC Bazzan. 2017. Optimal electric vehicle fast charging station placement based on game theoretical framework. IEEE Transactions on Intelligent Transportation Systems 19, 8 (2017), 2493–2504.

[38] Huaxiu Yao, Yiding Liu, Ying Wei, Xianfeng Tang, and Zhenhui Li. 2019. Learning from multiple cities: A meta-learning approach for spatial-temporal prediction. In Proceedings of the World Wide Web Conference (WWW’19). Association for Computing Machinery, New York, NY, 2181–2191.

[39] Weijia Zhang, Hao Liu, Fan Wang, Tong Xu, Haoran Xin, Dejing Dou, and Hui Xiong. 2021. Intelligent electric vehicle charging recommendation based on multi-agent reinforcement learning. In Proceedings of the Web Conference 2021. Association for Computing Machinery, New York, NY, 1856–1867.

[40] Yu Zheng, Licia Capra, Ouri Wolfson, and Hai Yang. 2014. Urban computing: Concepts, methodologies, and applications. ACM Transactions on Intelligent Systems and Technology 5, 3 (2014), 1–55.

Received 22 January 2022; revised 21 June 2022; accepted 17 September 2022