MetaStore: a task-adaptative meta-learning model for optimal store placement with multi-city knowledge transfer

Yan Liu, Bin Guo, Daqing Zhang, Djamal Zeghlache, Jingmin Chen, Sizhe Zhang, Dan Zhou, Xinlei Shi, Zhiwen Yu

MetaStore: A Task-adaptative Meta-learning Model for Optimal Store Placement with Multi-city Knowledge Transfer

YAN LIU and BIN GUO, Northwestern Polytechnical University, China
DAQING ZHANG and DJAMAL ZEGHLACHE, Télécom SudParis, France
JINGMIN CHEN, SIZHE ZHANG, DAN ZHOU, and XINLEI SHI, Alibaba Group
ZHIWEN YU, Northwestern Polytechnical University, China

Optimal store placement aims to identify the optimal location for a new brick-and-mortar store that can maximize its sale by analyzing and mining users' preferences from large-scale urban data. In recent years, the expansion of chain enterprises in new cities brings some challenges because of two aspects: (1) *data scarcity in new cities,* so most existing models tend to not work (i.e., overfitting), because the superior performance of these works is conditioned on large-scale training samples; (2) *data distribution discrepancy among different cities,* so knowledge learned from other cities cannot be utilized directly in new cities. In this article, we propose a task-adaptative model-agnostic meta-learning framework, namely, MetaStore, to tackle these two challenges and improve the prediction performance in new cities with insufficient data for optimal store placement, by transferring prior knowledge learned from multiple data-rich cities. Specifically, we develop a task-adaptative meta-learning algorithm to learn city-specific prior initializations from multiple cities, which is capable of handling the multimodal data distribution and accelerating the adaptation in new cities compared to other methods. In addition, we design an effective learning strategy for MetaStore to promote faster convergence and optimization by sampling high-quality data for each training batch in view of noisy data in practical applications. The extensive experimental results demonstrate that our proposed method leads to state-of-the-art performance compared with various baselines.

CCS Concepts: • **Information systems → Data mining**; • **Computing methodologies → Machine learning**;

Additional Key Words and Phrases: Urban computing, optimal store placement, machine learning, knowledge transfer, meta-learning

This work was supported by the National Key R&D Program of China (grant no. 2019YFB1703901), the National Science Fund for Distinguished Young Scholars (grant no. 62025205), the National Natural Science Foundation of China (grant nos. 61960206008, 61772428, 61725205), the China Scholarship Council, and the Innovation Foundation for Doctor Dissertation of Northwestern Polytechnical University (grant no. CX201958).

Authors' addresses: Y. Liu and B. Guo (corresponding author), Northwestern Polytechnical University, Xi'an, China; emails: yan_emily@outlook.com, guob@nwpu.edu.cn; D. Zhang and D. Zeghlache, Télécom SudParis, Évry, France; emails: {daqing.zhang, djamal.zeghlache}@telecom-sudparis.eu; J. Chen, S. Zhang, D. Zhou, and X. Shi, Alibaba Group, Hangzhou, China; emails: {jingmin.cjm, jincheng.zsz, modan.zd, sxl110655}@alibaba-inc.com; Z. Yu, Northwestern Polytechnical University, Xi'an, China; email: zhiwenyu@nwpu.edu.cn.

2157-6904/2021/04-ART28 $15.00
https://doi.org/10.1145/3447271

ACM Reference format:
Yan Liu, Bin Guo, Daqing Zhang, Djamal Zeghlache, Jingmin Chen, Sizhe Zhang, Dan Zhou, Xinlei Shi, and Zhiwen Yu. 2021. MetaStore: A Task-adaptative Meta-learning Model for Optimal Store Placement with Multi-city Knowledge Transfer. *ACM Trans. Intell. Syst. Technol.* 12, 3, Article 28 (April 2021), 23 pages.
https://doi.org/10.1145/3447271

1 INTRODUCTION

Recent advances in internet technology and mobile computing lead to a collection of large amounts of urban data from various sources in cities and significantly changes urban services and related applications [14, 40, 42, 44, 50], such as intelligent transportation systems, public safety, intelligent business, and so on. Intelligent business is a new business service in smart cities, which comprises the strategies and technologies used by internet companies [5, 9, 35]. Specifically, intelligent business technologies handle and analyze large amounts of business data to provide large-scale, real-time, and personalized services for mass consumers (e.g., recommendation in electronic commerce websites), as well as provide historical, current, and predictive views of business operations with a competitive market advantage and long-term stability. Optimal store placement is one of most fundamental services in intelligent business for the development of brick-and-mortar chain enterprises (e.g., Starbucks, Walmart) [4, 8, 46], as it can provide insights for the future success of the chain enterprise when placing a new store at the given candidate location. Thereby, an effective store placement approach becomes necessary to help the enterprise to improve the chain store's profit.

In recent years, the proliferation of multi-source data in cities has fostered unprecedented opportunities to the data-driven store placement [2, 47], and it aims to analyze and mine users' preferences based on user-generated data (e.g., check-in data, rating data) to select the optimal location for a new brick-and-mortar store that can maximize the sale of the store. The data-driven methods in optimal store placement can help the company (e.g., chain stores) to predict the popularity of the store at the given location accurately and intelligently via data mining and machine learning techniques.

Traditionally, some basic regression models are used for optimal store placement [18, 22, 39]. For example, in Reference [39], three types of features are incorporated into a regression model to predict the number of check-ins at a candidate location. Nevertheless, these methods rely on expertise feature engineering to characterize sophisticated influences and extract features from a single data source, which fail to learn complex feature interactions from multi-source data. Recently, with the rapid development of deep neural networks (DNNs), more and more works propose DNN-based approaches [23, 48] to improve the performance of site selection by characterizing consumption behavior based on multi-source data. The method proposed in Reference [23] learns low- and high-order feature interactions simultaneously to model complex user behaviors. However, the superior performance of existing models is conditioned on large-scale training data, and most of them cannot work in some new cities with a few historical consumption data where a chain enterprise just develops its market at the initial phase.

Recently, transfer learning [7, 29, 41] has been proved to be an effective approach to solve the data scarcity problem by transferring available knowledge from those cities with abundant data (i.e., source city) to improve the performance in the data-scarce city (i.e., target/new city). In Reference [43], the authors propose to transfer knowledge from a data-rich city to a data-scarce city by learning an inter-city region matching function to match each target city region to a similar source city region. Guo et al. [15] propose a two-fold knowledge transfer framework to transfer chain store knowledge for chain store site recommendation in a new city. However, the

major downside of these transfer models is that they focus on transferring knowledge from only a single source city, which limits the performance of knowledge transfer, because knowledge learned from multiple cities could be comprehensive and complementary. Furthermore, the knowledge transfer could hurt the performance due to the negative transfer if the data distribution between source city and target city are significantly different.

One of practical approaches to solve this issue is making sufficient use of samples in data-rich cities and transferring knowledge from multiple source cities. In contrast to transfer learning, meta-learning [12, 34] is a task-level learning method that has emerged recently aiming at learning to learn, or learning from small amounts of new data quickly. Specifically, meta-learning aims to accumulate knowledge/experience from learning multiple tasks and adapt to a new task rapidly based on few samples by leveraging prior experience. In view of the store placement scenario in new cities, we aim to develop a meta-learning approach to learn the prediction model in a new city based on a small number of historical consumption data by leveraging prior knowledge from multiple source cities.

In this article, we consider a state-of-the-art representative of meta-learning algorithms, namely, Model-Agnostic Meta-Learning (MAML) [12], which is widely used to solve few-shot learning problems because of its appealing performance. Specifically, it learns an initialization of a network by a meta-learner on a set of tasks, which is then optimized to adapt a new task via a small number of gradient updates. However, very few attempts have applied MAML to transfer city knowledge from source city to target city in urban applications, and we are still faced with two key technical challenges, which limit the effectiveness of this type of MAML-based approach for solving our problem: (1) ***How to learn available knowledge from multiple cities simultaneously with different data distributions?*** Intuitively, the data distribution may vary from city to city because of different characteristics of cities (e.g., POI distribution, road network). However, many MAML-based methods assume that all training and testing tasks are drawn from the same data distribution, and they aim to find a single meta-initialization, which could be difficult and inappropriate to search, since the data distribution in different cities is different and multimodal. Therefore, data distribution discrepancy among different cities poses a great challenge for transferring knowledge from multiple cities. (2) ***How to quickly adapt the knowledge to improve the performance in a new city with limited examples while avoiding overfitting to these new data?*** In real-world scenarios, many urban applications may suffer from data noisy issues. However, most conventional meta-learning methods randomly sample the training batch from each task, which makes it difficult to efficiently adapt to a new city based on a small number of training data, since the model tends to overfit on noisy data.

To tackle the aforementioned challenges, we propose MetaStore, a task-adaptative model-agnostic meta-learning framework for optimal store placement in new cities with insufficient data by transferring prior knowledge learned from multiple data-rich cities. Specifically, we propose a task-adaptative meta-learning algorithm to learn city-specific prior knowledge to solve the first challenge. Different from previous MAML-based methods that seek a common initialization shared across the entire task distribution, substantially limiting the diversity of the task distributions that they are able to learn from, our task-adaptative meta-learning algorithm learns a set of meta-learned initializations from a variety of source cities, which is capable of tackling complex multimodal data distributions and accelerating the adaptation in new cities. In particular, it first leverages the attention network to generate a set of city-specific parameters according to the characteristic of the city, which are fed in the prediction network to modulate meta-learned initialization. Then, the modulated parameters are further updated using a few steps of gradient descent to quickly adapt to the new city to improve its performance. Moreover, we design an effective learning strategy to solve the second challenge, which samples high-quality data in real-time

for each training batch instead of observing samples at random to promote faster convergence and stronger performance in view of noisy data in real-world situations. Note that although our focus in this article is on optimal store placement, our proposed method can also handle other few-shot learning problems with different models, such as spatial-temporal prediction, recommender systems, and so on.

In summary, we make the following contributions:

- We present a task-adaptative model-agnostic meta-learning framework, namely, MetaStore, to improve the prediction performance in new cities with limited data for optimal store placement by transferring prior knowledge learned from multiple data-rich cities.
- We propose a task-adaptative meta-learning algorithm to learn city-specific prior knowledge from multiple cities with the multimodal data distribution. It first learns a set of meta-learned initializations from multiple source cites according to the characteristic of the city, and then quickly adapts the well-generalized initialization of the prediction model to obtain superior performance on a new city based on only a few numbers of historical data.
- We design an effective learning strategy to promote faster convergence and optimization by sampling high-quality data for each training batch in view of noisy data in practical applications.
- We validate the effectiveness of our proposed model on a real-world dataset. Extensive experiments are conducted from different perspectives, and the results demonstrate that our proposed approach outperforms baseline methods.

The rest of this article is organized as follows: We begin by reviewing the related work in Section 2. We present the preliminary in Section 3. Section 4 elaborates the detailed design of the proposed MetaStore architecture. Empirical evaluation and discussion are reported in Section 5, while the conclusion is enclosed in Section 6.

2 RELATED WORK

In this section, we review the related work, including optimal store placement and knowledge transfer.

2.1 Optimal Store Placement

In recent years, the proliferation of multi-source data in cities has fostered unprecedented opportunities to data-driven store placement, which aims to analyze and mine users' preferences based on user-generated data to select the optimal location for a new brick-and-mortar store [1, 17].

The earliest store placement methods are based on some basic regression models [18, 22, 51]. For example, Karamshuk et al. [18] mine geographic and user mobility features from check-in data and predict the best placement of retail stores based on extracted features. Li et al. [22] extract three types of features from cross-space data sources and then adopt supervised regression and classification to solve two scale-specific chain store placement problems. In Reference [39], three types of features are incorporated into a regression model to predict the number of check-ins at a candidate location. Zeng et al. [51] extract features from heterogeneous urban data and then predict the popularity of a new retail store in the candidate space using various machine learning models. Unfortunately, these methods rely on feature engineering to extract features from a single data source, which fails to learn complex feature interactions from multi-source data.

Recently, with the rapid development of DNNs [26, 37], more and more works propose DNN-based approaches to improve the performance of store placement by characterizing consumption behavior based on multi-source data [23, 48]. Liu et al. [23] propose a model named DeepStore,

including the cross network, the deep network, and the linear component, thus, it can learn low- and high-order feature interactions explicitly and implicitly from dense and sparse features simultaneously to model complex user behavior. Xu et al. [48] propose an attentive neural method to select promising business locations by fusing the discriminative features extracted from urban data and satellite data. Specifically, it consists of three attention modules to learn deep feature interactions according to business types and predict the business popularity of a given location.

However, these models rely on large-scale labeled data to train the prediction model, which could not be satisfied in some cases. For example, there may exist only a few historical data in some new cities when a chain enterprise just develops its market at the initial phase, and most of previous works fail to address this problem due to the data scarcity problem. Different from most of existing works based on enough training samples, we aim to tackle the data scarcity problem for optimal store placement by transferring prior knowledge learned from data-rich cities to improve the performance in a new city with a small number of data.

2.2 Knowledge Transfer

Recently, knowledge transfer has been studied as an effective solution to address the data scarcity problem by avoiding expensive data labeling efforts. Specifically, transfer learning and meta-learning are two major methods widely used to transfer knowledge and improve learning performance.

Transfer learning. The objective of transfer learning is to transfer knowledge from some source domains to the target domain when the latter does not have enough training data, including instance-based [7], feature-based [29], model-based [13], and relation-based [25] methods. There have been recently a few works that leverage transfer learning to deal with urban data scarcity [15, 24, 43]. Guo et al. [15] propose a two-fold knowledge transfer framework to solve the cold-start problem for chain store recommendation. Specifically, it builds correspondence between different regions to bridge the discrepancy between the source city and the target city to transfer chain store knowledge. Liu et al. [24] propose a domain adaption network for hotspots detection in a new city by transferring hotspots knowledge learned from one source city with shared bikes. In Reference [43], authors propose a cross-city transfer learning method for deep spatio-temporal prediction tasks, which aims to transfer knowledge from a source city to a target city by learning an inter-city region matching function to match each target city region to a similar source city region.

However, there are two downsides when using these transfer learning methods in our work. First, most of works transfer knowledge with the similarity function based on the correlation between source and target cities, which is hard to extend to solve our problem because of sophisticated consumption behavior. Moreover, existing works mainly focus on transferring knowledge from only a source city, which limits the performance of knowledge transfer, because knowledge learned from multiple cities could be comprehensive and complementary. The knowledge transfer could hurt the performance due to the negative transfer if the data distribution between source city and target city is significantly different.

Meta-learning. Meta-learning has emerged recently in machine learning aiming at learning knowledge/experience from a variety of learning tasks, and then transfer learned knowledge to a new task with a few examples for fast adaptation. Different from transfer learning, meta-learning can be capable of quickly adapting to new tasks that have never been encountered during training time, thus meta-learning is also known as learning-to-learn.

Generally, meta-learning methods can be divided into three types: metric-based, model-based, and optimization-based methods. Metric-based methods [36] learn a metric space in which learning is efficient, and they are mostly used for classification problems. Model-based methods [28] use an external network to store experience to facilitate the learning process, but they could suffer

from overfitting and show limited generalization ability, because they introduce additional parameters. Optimization-based methods [34] aim to adjust the optimization algorithm so the model can be good at learning with a few examples. Specifically, it uses the meta-learner to update the learner's parameters so the learner can adapt to the new task quickly.

MAML [12] is a state-of-the-art representative of the optimization-based meta-learning approaches. Specifically, it learns a good parameter initialization of a network by a meta-learner on a set of training tasks, which is then optimized to adapt a new task via a small number of gradient updates. Especially, MAML is agnostic and does not expand the number of learned parameters, so it is compatible with any models learned through gradient descent to solve a variety of problems [21, 49, 52]. For example, Lee et al. [21] propose a recommender system based on MAML for the cold-start problem, and it includes a meta-learned user preference estimator that can adapt to new users based on a small number of item-consumption history. Inspired by MAML, a meta-learning method is proposed by Yao et al. [49] to address spatial-temporal prediction in new cities with only a short period of data collection. Specifically, the method learns a good initialization of the spatial-temporal network, which can be quickly adapted to new cities. In Reference [52], Zhang et al. propose a meta-learning method for clinical risk prediction with limited patient electronic health records. Particularly, they adopt a model agnostic gradient descent framework, which trains a meta-learner on a variety of tasks where the target clinical risks are relevant.

However, MAML aims to find a single meta-initialization, which could be difficult to search for all tasks if the data distribution is different and multimodal; meanwhile, it could lead to bad performance. For example, the data distribution differs among multiple cities in our problem. In addition, conventional meta-learning methods randomly sample the training batch from each task, which could bring random difficulties, since most urban data may suffer from data noisy issues in the real-world scenario.

Inspired by above-mentioned works, we aim to leverage the model-agnostic meta-learning method to address the data scarcity problem for optimal store placement by transferring prior knowledge learned from other source cities to improve the performance in a new city with a small number of data. Different from MAML-based methods above, our model learns city-specific prior knowledge from multiple cities, which is capable of tackling complex multimodal data distributions and accelerating the adaptation in new cities. Furthermore, we present an effective learning strategy to sample high-quality data for each training batch instead of observing samples at random to promote faster convergence and optimization.

3 PRELIMINARY

In this section, we first describe the problem formulation. Next, we extract useful features from multi-source data. Finally, we present the data analysis results. For brevity, we present a table of notations used in our work in Table 1.

3.1 Problem Formulation

Definition 3.1 (User Consumption in a Store). In a city c, given a store s_j located at l_j, and a set of users $U = \{u_1, u_2 \ldots u_i \ldots\}$ who have the possibility to consume in the store, we use y_{u_i,s_j} to denote the amount of consumption per customer u_i in store s_j during a given period of time T (e.g., a month), which is to be predicted. Then, the overall sale of the store s_j can be represented as the total amount of consumption of all consumers in this store: $M(s_j) = \sum_{u_i \in U} y_{u_i,s_j}$.

Definition 3.2 (Optimal Store Placement in a City). Given a set of candidate locations $L = \{l_1, l_2 \ldots l_j \ldots\}$ to place a new store in a city, let $\hat{M}(l_j)$ be the predicted sale of the store located at

Table 1. Notations

Notation	Description
$C^s = \{c_1^s, c_2^s ... c_i^s ...\}$	Source cities
$C^t = \{c_1^t, c_2^t ... c_i^t ...\}$	New/target cities
$D_{T_{c_s}}^{train}$	Support set of source city c_s
$D_{T_{c_s}}^{test}$	Query set of source city c_s
$D_{T_{c_t}}^{train}$	Training data of target city c_t
$D_{T_{c_t}}^{eval}$	Evaluation data of target city c_t
θ_f	The parameters of feature extractor
θ_y	The parameters of label predictor
ω	The parameters of attention network
λ	The city-specific modulation vectors

the candidate place l_j. The optimal store placement in a city can be defined that the candidate location with the highest sale will be selected as the optimal location to place a new store.

Problem Statement. For a chain enterprise, assume that there are a set of source cities $C^s = \{c_1^s, c_2^s ... c_i^s ..., c_m^s\}$ with enough data and some new/target cities $C^t = \{c_1^t, c_2^t ... c_i^t ...\}$ with limited data. In this article, we aim to solve the optimal store placement for this chain enterprise in new cities based on only a small number of consumption records in stores, by leveraging and transferring knowledge from multiple source cites.

Specifically, for a target city $c_t \in C^t$, given a set of candidate locations $L^t = \{l_1^t, l_2^t ... l_j^t ...\}$, a set of users $U^t = \{u_1^t, u_2^t ... u_i^t ...\}$, and multi-source data (including consumption data, POI, etc.), the objective of this problem is to predict the consumption behavior of each user for each store located at candidate places in the target city, denoted as $\hat{y}_{u_i^t, l_j^t}$. It should be noted that known the predicted consumption behavior, we then compute the overall sale of the store $\hat{M}(l_j)$ located at each candidate place l_j, and hence the optimal store placement can be solved by selecting the optimal place with the highest sales from L^t to place a new store in the target city.

3.2 Feature Extraction

In this work, we choose a retail enterprise for a case study, which owns a lot of brick-and-mortar stores in some cities of China. The dataset is real-world urban data, which contains three types of data for this work, including retail enterprise data, user data, and POI data. The details of the dataset will be presented in the experiments. To predict the consumption behavior of each user in stores, we extract useful features. Specifically, we mainly consider the following features extracted from multi-source data, including user features, geographic features, commercial features, and time features.

User Features. Intuitively, identifying whether the candidate location is appropriate to place a new store in the long term mainly depends on the nearby users. Following previous work [23], we associate each user with a location-based community, which is a group of homes and other buildings built together. To characterize potential customers in different communities, we consider some demographic profiles, such as *gender*, *age*, *profession*, *income level*, and so on, and then make statistics on the number of people with different profiles in each community as user features.

Geographic Features. The spatial characteristics of the place where the store resides affect the possibility of users going to the store. We thus extract the following geographic features including: (1) *distance*: the Manhattan distance between the community where people live and the store,

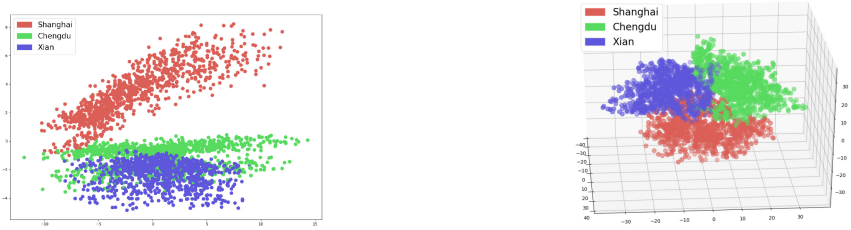

Fig. 1. T-SNE visualization on different cities.

(2) *traffic convenience*: the total number of transportation stations (including bus stations and subway stations) in the surrounding area, which is a disk centered at the store with radius r, (3) *POI set*: the number of POIs of each category (e.g., shopping, food, company) that could attract potential consumers in the surrounding area, (4) *neighbor's entropy*: POI entropy of different categories in the surrounding area.

Commercial Features. Inspired by Reference [18], to characterize commercial features around the area where the store resides, we extract three commercial features: (1) *density*: the total number of stores of different categories in the surrounding area, which lie in a disk of radius r around the store, (2) *competitiveness*: the proportion of neighboring places of the same type with respect to the total number of nearby places, (3) *complementarity*: Jensen Quality [17] of the store to assess the complementarity relationship of spatial interactions of places with respect to their ability to attract other places of certain types pairs.

Time Features. We extract time features to capture consumer behavior in the store at different stages: (1) the date of establishment to place the store (e.g., the year), (2) the number of existing stores, (3) whether the time is the holiday when users consume in the store (e.g., New Year's Day and National Day).

3.3 Data Analysis

Intuitively, the data distribution may vary from city to city because of different characteristics of cities, such as POI distribution, road network, and so on, thus a prediction model built for one city may not predict well in other cities because of different data distributions. Furthermore, data distribution discrepancy among different cities poses a great challenge for transferring knowledge from multiple cities.

To illustrate data distribution discrepancy among different cities, we visualize t-SNE results over the feature space. T-SNE [10] is an effective method to visualize the high-dimensional data distributions. Specifically, we randomly sample some examples from three cities in China, including Shanghai, Chengdu, and Xi'an, and then visualize the t-SNE embeddings of the feature representations extracted from multi-source data. Figure 1 shows the feature distribution using t-SNE embeddings under two kernel functions, respectively. As shown in Figure 1, when the number of feature dimension is reduced to low dimension (i.e., 2 and 3), we can see the obvious data distribution difference among cities, especially between the tier-1 city (Shanghai) and tier-2 city (Chengdu and Xi'an), which means that different tasks sampled from different cities with different distributions can require substantially different parameters of the prediction model in different cities. In general, it is necessary to consider the data distribution discrepancy for the optimal store placement in a new city when transferring prior knowledge learned from multiple source cities with different data distributions.

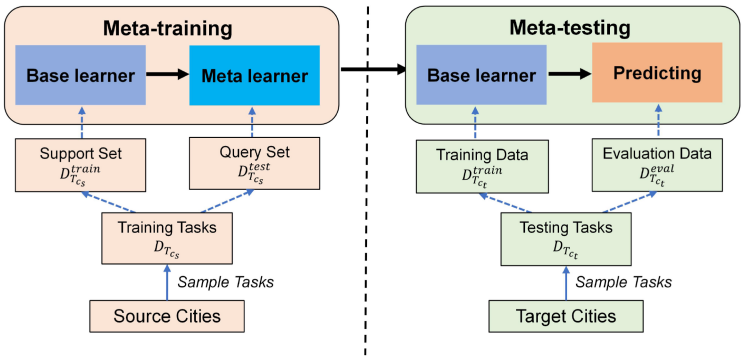

Fig. 2. The learning procedure of MetaStore.

4 THE METASTORE ARCHITECTURE

In this section, we describe the details of the MetaStore architecture. We first introduce the meta-learning setup for optimal store placement. Next, we design a prediction model to predict the consumption behavior of each user for each store. Finally, we propose a novel task-adaptative meta-learning algorithm to learn city-specific prior knowledge from a variety of source cities in view of multimodal data distribution. It should be noted that MetaStore is model-agnostic, thus it can be easily extended to more sophisticated neural networks to solve various problems, such as spatial-temporal prediction, recommender systems, and so on.

4.1 Meta-learning Setup

In this work, we aim to learn the prediction model for optimal store placement in new cities with limited data by transferring and leveraging knowledge from multiple source cites with sufficient data. However, traditional transfer learning would be constrained by the small number of training samples, and most of transfer learning works mainly focus on transferring the knowledge from only a single source city.

Different from transfer learning, meta-learning aims to train a model that can rapidly adapt to a new task. Consequently, we take advantage of meta-learning to learn knowledge from multiple source cities. The MAML provides us a parameter initialization strategy, and the parameters of the model can be viewed as the knowledge learned from multiple tasks that can be transferred to the new task. More formally, we regard each task as predicting consumer behavior in each city, and the details on the design of the prediction model will be introduced in the next section.

As shown in Figure 2, the learning procedure of MetaStore consists of two phases: meta-training and meta-testing. Specifically, meta-training aims to learn knowledge from a number of tasks sampled from a set of source cities. A new task in meta-testing will start from that knowledge and adapt to the new task quickly. In particular, in our work, a meta-training task is a regression task T_{c_s} sampled from each source city $c_s \in C^s$, and a meta-testing example T_{c_t} is the same prediction task in the target city $c_t \in C^t$.

Meta-training Phase. The objective of meta-training is to learn an initialization of the model by a meta-learner based on multiple source cities. In each source city, meta-training has a two-stage optimization using two sets, respectively, namely, the support set and query set. *During the local update*, the base-learner optimizes the parameters of the model by minimizing the training loss on each support set $D^{train}_{T_{c_s}}$ of source city c_s sampled from the set of source cities C^s. *During the global update*, for all source cities, the meta-learner trains the model parameters to minimize

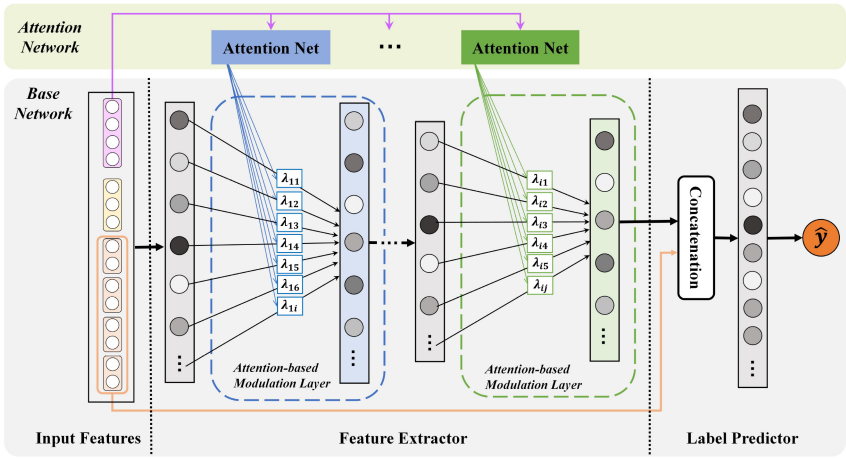

Fig. 3. The prediction model of MetaStore.

the testing losses using the locally adapted parameters on the query sets $D^{test}_{T_{c_s}}$ across cities, so the globally updated parameters could fit into various cities.

Meta-testing Phase. After the meta-training phase, we could obtain the desirable initialization of the prediction model, which can adapt quickly to various cities. Therefore, the parameters of the model can be viewed as the knowledge learned from multiple source cities, which can be transferred to the new city for fast adaptation. The meta-testing phase aims to test the generalization performance of the initialization learned by meta-learner to a new city. Specifically, given a new city c_t sampled from the target cities C^t, to improve the performance in the new city, we first transfer the initialization of the prediction model, and then adjust the parameters to adapt to the new city c_t on training data $D^{train}_{T_{c_t}}$ (i.e., fast adaptation on each target city). Finally, we predict the value of the target city c_t using the adapted parameters and evaluate the prediction model on evaluation data $D^{eval}_{T_{c_t}}$.

4.2 Prediction Model

The main idea of MetaStore model is to combine the base network and the attention network to quickly adapt to a new city in view of multimodal data distribution. To acquire city-specific prior knowledge learned by the meta-learner, we first leverage the attention network to generate a set of city-specific parameters considering of the unique characteristic of the city, and then the output vector of the attention network is fed in the base network to modulate parameters of the base network through the attention-based modulation layer. Finally, the parameters of the modulated base network are further updated to adapt to the new city by the base-learner. The prediction model of MetaStore is shown in Figure 3.

4.2.1 Input Features. For each store, we first identify its neighbor area where most of potential consumers live, such as the surrounding area that is a disk centered at the store with radius r. Instead of considering each potential consumer as an entity in the prediction task, we associate each user with a location-based community. Hence, we extract geographic features $\mathbf{x}_{fg}$, commercial features $\mathbf{x}_{fc}$, and time features $\mathbf{x}_{ft}$ of each store, and user features $\mathbf{x}_{fu}$ of nearby communities from multi-source data, as presented in Section 3.2. In addition, to further reinforce the consumer behavior learning, we take the embedding vectors of city $\mathbf{x}_{ec}$ and the embedding vectors of

community $\mathbf{x}_{em}$ into consideration, which can be learned based on other related consumption data (e.g., a large number of online consumption data).

4.2.2 Attention Network. The attention network aims to produce a set of city-specific parameters for the base network based on the characteristic of the city. Instead of modulating all parameters of the base network, we just focus on modulating parameters in feature extractor (the details about feature extractor will be introduced in Section 4.2.3), because the feature extractor is capable of obtaining the internal representation, which could be more transferrable. More specifically, to module the parameters in the base network as good initialization that can be updated to adapt to the new city efficiently, we propose to employ the attention mechanism on the output of each layer in feature extractor in the base network. Moreover, the feature extractor could consist of multiple layers, such as the convolutional layer, fully connected layer. Therefore, we apply multiple attention sub-networks (e.g., MLPs) to generate various modulation vectors for each layer of feature extractor in the base network, which can be formulated as:

$$\lambda_i = g_i\left(\mathbf{x}_{ec}; \omega\right), where\ i = 1, \ldots, N, \tag{1}$$

where N is the number of layers in feature extractor, and function g_i with parameters ω represents the attention network. It should be noted that $\mathbf{x}_{ec}$ is the embedding vector of each city that encodes the characteristic of a city. In this work, we learn the city embedding vector by unsupervised learning approach in advance based on other related consumption data, and it can also be learned by other methods in terms of different problems.

4.2.3 Base Network. We present the base model to predict the consumption behavior of each user for each store located at candidate areas, which consists of the following two components: feature extractor and label predictor.

Feature Extractor. Besides some valuable features extracted from multi-source data, we adopt the feature extractor to learn their deep interactions based on the raw extracted features, since consumer behavior is usually affected by various complicated factors simultaneously. Specifically, the feature extractor contains N fully connected layers for deep feature representation learning:

$$\begin{aligned} \mathbf{x}_1 &= \sigma\left(\mathbf{W}_{f1}\mathbf{x} + \mathbf{b}_{f1}\right), \\ \mathbf{f} &= \sigma\left(\mathbf{W}_{fN}\mathbf{x}_{N-1} + \mathbf{b}_{fN}\right), \end{aligned} \tag{2}$$

where $\mathbf{x} = [\mathbf{x}_{fg}, \mathbf{x}_{fc}, \mathbf{x}_{ft}, \mathbf{x}_{fu}, \mathbf{x}_{ec}, \mathbf{x}_{em}]$ is the input vector containing different types of features, the $\mathbf{W}$ and $\mathbf{b}$ are weight and bias of the feature extractor, and σ is the non-linear activation function for which we use *ReLU*. The output of the feature extractor is the final feature representation $\mathbf{f}$, which will be fed in the label predictor.

In recent years, modulation operators have been widely used in modern deep learning models to modulate neural networks for achieving the conditioning effects of data from different modalities. There are some representative modulation operations, such as attention-based modulation [27, 38] and feature-wise linear modulation (FiLM) [16, 30, 32]. For example, FiLM modulation has been used in a variety of tasks and shows its effectiveness [11, 19, 31, 45]. Inspired by previous works, we employ the modulation operation in our proposed meta-learning framework. To acquire city-specific prior knowledge learned by meta-learner, we add attention-based modulation layers in the feature extractor. The attention-based modulation layer aims to modulate parameters $\theta_{fi} = \{\mathbf{W}_{fi}, \mathbf{b}_{fi}\}$ of each layer of feature extractor using city-specific parameters λ_i, which is the output of the attention network. Specifically, modulation vectors are used to scale the pre-activation of

each feature extractor layer, which can be defined as:

$$\begin{aligned} \mathbf{x}_i &= \mathbf{W}_{fi}\mathbf{x}_{i-1} + \mathbf{b}_{fi}, \\ \mathbf{x}_i &= \sigma\left(\mathbf{x}_i \odot \lambda_i\right). \end{aligned} \tag{3}$$

Label Predictor. Label predictor aims to predict the amount of consumption of users in the store located at the candidate place. According to the insights of the Wide&Deep model [6], we consider low- and high-order feature interactions simultaneously for label prediction. Formally, given the deep feature representation $\mathbf{f}$ from feature extractor and raw extracted features, we first concatenate them to obtain the input vector of the label predictor, and then predict the amount of consumption of potential customers based on the following operations. The layers can be formulated as:

$$\begin{aligned} \mathbf{a} &= \sigma\left(\mathbf{W}_y\left[\mathbf{f}, \mathbf{x}_{fg}, \mathbf{x}_{fc}, \mathbf{x}_{ft}, \mathbf{x}_{fu}\right] + \mathbf{b}_y\right), \\ \hat{y} &= \mathbf{h}_y^T\mathbf{a}, \end{aligned} \tag{4}$$

where the $\mathbf{W}_y$ and $\mathbf{b}_y$ terms are weight matrix and bias vector, respectively, σ is the *ReLU* activation function, and $\mathbf{h}_y$ denotes the neuron weights in the output layer.

In general, we use a parameterized function f with parameters $\Theta = \{\theta_f, \theta_y, \lambda$ to represent the prediction model for optimal store placement, where λ denotes modulation vector, θ_f and θ_y denote parameters of feature extractor and label predictor, respectively. The prediction model can be optimized by minimizing the loss function:

$$\mathcal{L}\left(\Theta\right) = \frac{1}{m}\sum_{i=1}^{m}\left(\hat{y}_i - y_i\right)^2, \tag{5}$$

where $\hat{y}$ is the output of the prediction model $\hat{y} = f(\mathbf{x}; \theta_f, \theta_y, \lambda)$, and y the ground-truth.

4.3 Task-adaptative Meta-learning Algorithm

To learn various knowledge and enable transferring knowledge across different modes of the data distribution sampled from multiple cities, we propose a task-adaptative meta-learning algorithm. Specifically, we first present the base-learner with feature reuse to facilitate the learning for fast adaptation. Next, a novel task-adaptative meta-learner is proposed to learn city-specific prior knowledge from a variety of cities in view of multimodal data distributions. Furthermore, we introduce an effective learning strategy to sample high-quality data for each training batch to promote faster convergence. Finally, Algorithm 1 and Algorithm 2 outline the meta-training and meta-testing process, respectively.

4.3.1 Feature-reused Base-learner. Base-learner aims to locally update the model's parameters using one or more gradient descent steps by minimizing the training loss on the support set when adapting to a city c_s. Note that the attention network is used to generate a set of city-specific parameters for the base network according to the characteristic of the city, thus, parameters ω of the attention network are kept fixed during the local update.

Different from the base-learner in most existing works, in our proposed framework, we do not update the parameters of the feature extractor in the base network to ensure the stability of the learning process and reduce the learning time. The main reason is that the meta-initialization already provides high-quality feature representations that could be more transferrable. In Reference [33], the authors have proved that the effectiveness of MAML is due to feature reuse with meta-initialization via ablation studies. The experiments show that the meta-initialization has already learned good enough features that can be reused without needing to perform any rapid

learning for each test task. Inspired by this work, we propose the feature-reused base-learner instead of the standard base-learner in MAML. The feature-reused base-learner in our framework has two advantages. On the one hand, the feature-reused base-learner significantly speeds up training, as it does not update the parameters of the feature extractor. On the other hand, it could avoid overfitting on noisy data to some extent, because high-quality feature representations have been learned on a lot of training data during the global update, and they are fixed during the local update although there are a small amount of training data in each task. Specifically, for city c_s, the base-learner only updates the parameters θ_y in label predictor for fast adaptation:

$$\theta'_{y,c_s} \leftarrow \theta_y - \alpha \nabla_{\theta_y} \mathcal{L}_{c_s} \left(f\left(\mathbf{x}; \theta_f, \theta_y, \lambda_{c_s}\right); D^{train}_{T_{c_s}} \right), \tag{6}$$

where α is a hyperparameter controlling the update rate, and λ_{c_s} is the city-specific parameters generated by the attention network.

4.3.2 Task-adaptative Meta-learner. The limitation of most MAML-based frameworks is that they seek a common initialization shared across the entire task distribution, substantially limiting the diversity of the task distributions that they are able to learn from. In our problem, the data distribution may vary from city to city because of different characteristics of cities (e.g., POI distribution, road network). Therefore, we propose a task-adaptative meta-learning algorithm, which is able to modulate its meta-learned prior parameters according to the characteristic of the city. The objective of task-adaptative meta-learner is to acquire the task-specific prior knowledge (i.e., city-specific initialization of the prediction model), which can be transferred to adapt to various tasks (i.e., cities) with different data distributions that achieve good performance after a few local updates. As mentioned above, city-specific prior initialization is computed by modulating the parameters of the base network using a set of city-specific parameters generated by the attention network. Therefore, all parameters of the base network and attention network are globally updated by meta-learner to minimize the testing losses using the locally adapted parameters θ'_{y,c_s} on the query sets $D^{test}_{T_{c_s}}$. The meta-optimization is performed using stochastic gradient descent as follows, where β is the meta-learning rate:

$$\begin{aligned}
\theta_f &\leftarrow \theta_f - \beta \nabla_{\theta_f} \sum_{c_s \in C^s} \mathcal{L}_{c_s} \left(f\left(\mathbf{x}; \theta_f, \theta'_{y,c_s}, \lambda_{c_s}\right); D^{test}_{T_{c_s}} \right), \\
\theta_y &\leftarrow \theta_y - \beta \nabla_{\theta_y} \sum_{c_s \in C^s} \mathcal{L}_{c_s} \left(f\left(\mathbf{x}; \theta_f, \theta'_{y,c_s}, \lambda_{c_s}\right); D^{test}_{T_{c_s}} \right), \\
\omega &\leftarrow \omega - \beta \nabla_{\omega} \sum_{c_s \in C^s} \mathcal{L}_{c_s} \left(f\left(\mathbf{x}; \theta_f, \theta'_{y,c_s}, \lambda_{c_s}\right); D^{test}_{T_{c_s}} \right).
\end{aligned} \tag{7}$$

4.3.3 Learning Strategy. The conventional meta-learning methods randomly sample the training batch from each task, which could bring random difficulties. For example, in real-world scenarios, most urban data may suffer from noisy data issues, including feature and label noises. In this case, it is difficult to efficiently adapt to a new city based on a small number of training data, since the prediction model tends to overfit on noisy data. Inspired by curriculum learning [3], which organizes examples by a better arrangement to promote faster convergence and stronger performance, we present a learning strategy to sample high-quality data in a meaningful way for each training batch instead of observing samples at random.

Given the prediction model represented by the parametrized function f with parameters Θ, our proposed learning strategy is to update parameters by minimizing the following objective:

$$\mathcal{L}(\Theta) = \frac{1}{m} \sum_{i=1}^{m} w_i (f(\mathbf{x}_i; \Theta) - y_i)^2, \tag{8}$$

ALGORITHM 1: MetaStore Training

Input: the set of source cities C^s, and step size hyperparameters α and β
Output: Feature extractor θ_f, label predictor θ_y, attention network ω

1: Randomly initialize θ_f, θ_y
2: Randomly initialize ω
3: **while** not done **do**
4: Sample batch of cities from C^s
5: **for** each source city c_s **do**
6: Sample high-quality datapoints $D_{T_{c_s}}^{train}$, $D_{T_{c_s}}^{test}$ from $D_{T_{c_s}}$
7: Compute city-specific parameters:

$$\lambda_{c_s} = \{g_i\ (\mathbf{x}_{ec,c_s}; \omega)\,|i = 1, \ldots, N\}$$

8: Evaluate $\nabla_{\theta_y} \mathcal{L}_{c_s}(f(\mathbf{x}; \theta_f, \theta_y, \lambda_{c_s}); D_{T_{c_s}}^{train})$
9: Local update:

$$\theta_{y,\,c_s}^{'} \leftarrow \theta_y - \alpha \nabla_{\theta_y} \mathcal{L}_{c_s}(f(\mathbf{x}; \theta_f, \theta_y, \lambda_{c_s}); D_{T_{c_s}}^{train})$$

10: **end for**
11: Global update:

$$\theta_f \leftarrow \theta_f - \beta \nabla_{\theta_f} \sum_{c_s \in C^s} \mathcal{L}_{c_s}(f(\mathbf{x}; \theta_f, \theta_{y,\,c_s}^{'}, \lambda_{c_s}); D_{T_{c_s}}^{test})$$

$$\theta_y \leftarrow \theta_y - \beta \nabla_{\theta_y} \sum_{c_s \in C^s} \mathcal{L}_{c_s}(f(\mathbf{x}; \theta_f, \theta_{y,\,c_s}^{'}, \lambda_{c_s}); D_{T_{c_s}}^{test})$$

$$\omega \leftarrow \omega - \beta \nabla_{\omega} \sum_{c_s \in C^s} \mathcal{L}_{c_s}(f(\mathbf{x}; \theta_f, \theta_{y,\,c_s}^{'}, \lambda_{c_s}); D_{T_{c_s}}^{test})$$

12: **end while**

where $w_i \in [0,1]$ is a weight to quantify whether the ith example is a high-quality instance. Specifically, the loss l_i between predicted value and label is the metric to choose high-quality samples, $l_i = (f\ (\mathbf{x}_i; \Theta) - y_i)^2$. Furthermore, we can choose easy samples based on fixing a threshold ahead of time. For example, we adopt a classic curriculum in Reference [20] to identify the optimal weight w as follows:

$$w_i = \mathbb{I}(l_i \leq \varphi),\ i = 1, \ldots, n, \tag{9}$$

where $\mathbb{I}$ is the indicator function, and the hyper-parameter φ controls the learning pace.

In our proposed framework, we build the task for each city with different distributions, and multiple meta-training tasks are trained simultaneously by the task-adaptative meta-learner. Although we add the weights for each sample in the loss function to choose high-quality samples, the model only trained on high-quality samples with good performance could not lead to lower predictability. There are two reasons: On the one hand, we just filter out the noisy data, which not only makes almost no contribution to the model but also could hurt the performance of the model. On the other hand, multiple meta-training tasks in our framework enhance the number of training data and the diversity of training data. Therefore, our proposed learning strategy could promote faster convergence and optimization.

4.3.4 Algorithm of Optimization. Algorithm 1 shows the detailed procedure of the meta-training process, consisting of two main stages: local update and global update, to learn the prior knowledge from multiple source cities, i.e., meta-initialization of the model. Note that the data sampling shown in line 6 means that we update parameters based on high-quality examples using

ALGORITHM 2: MetaStore Testing

Input: the set of target cities C^t, learned parameter θ_f, θ_y, ω, and step size hyperparameters α
Output: predicted value $\hat{y}$

1: **for** each target city $c_t \in C^t$ **do**
2: Sample high-quality datapoints $D^{train}_{T_{c_t}}$ from $D_{T_{c_t}} \backslash D^{eval}_{T_{c_t}}$
3: Compute city-specific parameters:

$$\lambda_{c_t} = \{g_i\ (\mathbf{x}_{ec,c_t}; \omega)\ | i = 1, \ldots, N\}$$

4: Evaluate $\nabla_{\theta_y} \mathcal{L}_{c_t}(f(\mathbf{x}; \theta_f, \theta_y, \lambda_{c_t}); D^{train}_{T_{c_t}})$
5: Parameter fast adaptation with gradient descent:

$$\theta^{'}_{y,\,c_t} \leftarrow \theta_y - \alpha \nabla_{\theta_y} \mathcal{L}_{c_t}(f(\mathbf{x}; \theta_f, \theta_y, \lambda_{c_t}); D^{train}_{T_{c_t}})$$

6: Predict the result:

$$\hat{y} = f(\mathbf{x}; \theta_f, \theta^{'}_{y,\,c_t}, \lambda_{c_t}) \quad \mathbf{x} \in D^{eval}_{T_{c_t}}$$

7: **end for**

our proposed learning strategy, instead of selecting high-quality data before the learning process. During the local update (lines 5–10), the algorithm first computes the city-specific parameters, and then updates parameters in label predictor based on modulated meta-initialization. During the global update (line 11), all parameters are updated to minimize the testing losses of a batch of cities using the locally adapted parameters until the stopping criteria is met.

Algorithm 2 presents the meta-testing process for fast adaptation on target cities. For each target city $c_t \in C^t$, the parameters of the attention network ω and parameters in feature extractor θ_f are fixed, and parameters θ_y in label predictor are trained to adapt to city c_t (line 5). Finally, we could predict the result and evaluate the model in each target city.

5 EXPERIMENTS

In this section, we first introduce the experimental setup. Next, we compare our proposed algorithm with baseline methods empirically. Finally, we discuss the deep insights and limitations of our work.

5.1 Experimental Setup

5.1.1 Dataset. We choose a chain retail enterprise for a case study, which owns a lot of brick-and-mortar stores in some cities of China. The datasets used in our experiments are real-world urban data provided by commercial companies, including chain retail enterprise data, user data, and POI data.

For the retail enterprise data, we collect data of the chain retail enterprise in 134 brick-and-mortar stores from 22 cities in China during 15/01/2016 and 15/11/2019, including the profile information and historical sales data of the store. The profile information contains the shop name, city, location (e.g., longitude and latitude), and opening date. For the historical sales data, each record contains the customer ID, shop name, and customer behavior. The amount of sales is one of the important factors used to evaluate the location of the store, mainly including daily sales, monthly sales, and annual sales. Generally speaking, daily sales has bigger uncertainty and annual sales is too long-term, thus, we select monthly sales as the evaluation indicator. Therefore, the units in our sales data represent the amount of money in RMB consumed in each store per community per month.

For the user data, we obtain data of users in 66,792 communities in 22 cities, including the location information (i.e., the location of the community where users live) and profile information (e.g., gender, age, profession, income level). To protect user privacy, in this work, we just collect the information of a group of people as the units (i.e., the location-based community), which shares a sense of place that is situated in a given geographical area (e.g., a neighborhood). More specifically, a community in our dataset means a housing estate, which is a group of homes and other buildings built together. Then, we make statistics on user information in the community. For the location information, we obtain the number of people in a community and the location of the community. For the user profile, we have the following information: the number of men or women in the community, the number of people in different age groups, and so on.

For the POI data, it contains geographic information (e.g., name, location, and category) of multiple categories of POI related to the retail business. All the brick-and-mortar stores of the chain retail enterprise are one type of stores, and they are similar to the supermarket (e.g., Walmart, Carrefour), which mainly sell food (e.g., vegetables, fruits, seafood), commodities, and so on. However, different from the supermarket, these stores are also similar to the restaurant that people could also eat in the store after buying the seafood. Therefore, in this work, we consider 10 categories of POI related to these stores, including shopping, food, transport, company, education, sport, service, medical, hotel, and scene.

5.1.2 Compared Baselines. We compare the proposed method with two categories of methods: transfer learning method and meta-learning method, since they can transfer learned knowledge from source cities to improve the prediction accuracy in the target city. Note that we do not consider traditional methods (e.g., GBDT, DNN) for comparison, because they cannot learn the prediction model with limited examples in new cities. The structures of transfer learning method and meta-learning method are the same as that of our proposed model.

Fine-tuned Method. For transfer learning, we could train the model on training data of source cities to obtain general knowledge, and then fine-tune the pretrained model based on a few training data of target cities. Specifically, given multiple source cities, we first mix all examples from all source cities, and then train the general model based on source cities for the adaptation in target cities. It should be noted that we did not consider other transfer learning methods in our experiment, such as RegionTrans [43], because we aim to study the consumption behavior in a fine-grained manner, and it is improper to define the grid region in the city for optimal store placement, which is used to transfer knowledge based on the similarity of different regions.

MAML-based Method (MAML). MAML is a state-of-the-art meta-learning method to learn a better initialization of a network from a set of source cities, which is then optimized to adapt to a new city via a small number of gradient updates. We present two variants of MAML according to parameter optimization by base-learner. *MAML-base* locally updates all parameters of the model by base-learner using one or more gradient descent steps when adapting to a city. Different with *MAML-base*, *MAML-fr* only updates the parameters in the label predictor, and the parameters in feature extractor will be not updated to facilitate the learning for fast adaptation.

5.1.3 Evaluation Metrics. We measure the prediction performance of our model and baselines using the two metrics:

Mean Square Error. For each target city, our model predicts the total amount of consumption that all users in each community consume in the store located at the candidate place. Thus, we use *Mean Square Error* (MSE) for result comparison:

$$MSE = \frac{1}{m}\sum_{i=1}^{m}(\hat{y}_i - y_i)^2, \tag{10}$$

where $\hat{y}_i$ and y_i are predicted and actual values, respectively, and m is the number of all samples in the evaluation set in the target city.

Mean Absolute Error. For the optimal store placement, we assume that there could be several candidate places to be chosen in the target city, and the objective of our problem is to select the optimal location with the highest sale from a set of candidate places. In this way, we adopt *Mean Absolute Error* (MAE) to evaluate the performance of the model for each store s_j that will be located at the candidate place l_j, as follows:

$$MAE = \frac{1}{n}\sum_{j=1}^{n}\left|\frac{\hat{M}_j - M_j}{M_j}\right|, \tag{11}$$

where n is the number of candidate locations to place the store in the evaluation set in the target city, and M_j is the actual sale of store s_j. $\hat{M}_j$ is the overall sale of store s_j predicted by the model, which could be represented as the total amount of consumption of all potential consumers in this store.

5.1.4 Implementation Details. In this work, for each store, we predict the consumption behavior of users in one month. Based on the empirical knowledge and our previous studies [23], we identify users who live in the communities within 5 *kilometers* of the store as all potential consumers in the experiments. In addition, we extract the geographic features of surrounding areas, which lie in a disk of radius 3 *kilometers* around the store and the community.

In the experiment, we consider 22 tasks sampled from 22 cities, and we select 6 typical cities as new test cities (i.e., meta-testing tasks), including 2 tier-1 cities (Shanghai and Guangzhou) and 4 tier-2 cities (Nanjing, Hangzhou, Chengdu and Xi'an), respectively, to evaluate the performance of the proposed model. Specifically, we learn a set of meta-learned initializations of a network by a task-adaptative meta-learner on 16 source cities (i.e., meta-training tasks), and then, respectively, optimize the prior initialization to adapt new cities (i.e., meta-testing tasks) via a small number of gradient updates. More specifically, in the meta-testing phase, for each new test city, we further split the data into training data and evaluation data. Instead of randomly selecting training data, we select the store with the earlier opening date as training instances, because the prediction model needs to predict the optimal location of a new store in practical applications. In general, there is a total of 30 stores in the evaluation data of 6 new cities to measure the predictive ability of the proposed model.

All experiments were implemented with Tensorflow using the following structure: For the base network, two layers were used for the feature extractor, and one layer is used for the label predictor. For the attention network, we use one fully connected layer to produce modulation vectors. The dimensions of city embedding vectors and community embedding vectors are 17 and 128, respectively. Moreover, Batch normalization is employed to avoid over-fitting. We apply mini-batch stochastic optimization with Adam optimizer, and set step sizes α and β to 0.001 and 0.01, respectively. The training batch size for each meta-iteration is set as 16, and the maximum of iteration of meta-learning is set as 200,000.

5.2 Experimental Results

Having depicted the experimental setups and baselines, we present the experimental results to evaluate our proposed method.

5.2.1 Performance Comparison of Different Models. The performance of different methods is presented in Table 2. Note that MSE is the major metric to evaluate the performance of the proposed method, and MAE is the additional result of computation based on predicted value. We find

Table 2. Performance Comparison of Different Models

City		Shanghai		Nanjing		Hangzhou		Guangzhou		Chengdu		Xi'an	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Fine-Tune		115.39	0.376	74.21	0.155	86.67	0.364	137.07	0.395	100.29	0.343	82.63	0.387
MAML	MAML-base	106.81	0.330	70.80	**0.057**	69.65	0.328	118.50	0.319	99.81	0.315	83.33	0.351
	MAML-fr	140.58	0.401	82.66	0.204	71.99	0.317	117.78	0.292	100.4	0.330	**71.46**	0.342
MetaStore	MetaStore-base	103.27	0.319	67.28	0.089	66.70	**0.168**	111.93	**0.125**	**66.72**	**0.163**	83.48	0.213
	MetaStore-fr	**98.91**	**0.308**	**62.13**	0.081	**63.76**	0.184	**101.00**	0.275	67.04	0.195	78.49	**0.187**

that MAE values vary a lot from city to city and have no obvious patterns compared with MSE. This is because we choose users who live in the communities within 5 *kilometers* of the store as all potential consumers in the experiments, and some users could be missed. In addition, the computation of MAE also could lead error to some extent.

According to these results, we draw the following conclusions:

- We can find that most of meta-learning methods outperform the fine-tuned method, especially the MetaStore method, which demonstrates that a better initialization could improve the performance, and the meta-learned initialization has the advantage in adapting to a new city rapidly based on few samples.
- In some cities, such as Hangzhou, MAML outperforms the fine-tuned method. However, in other cities, such as Nanjing, the fine-tuned method achieves competitive performance compared with MAML. One possible reason is that Nanjing is more similar to source cities than Hangzhou. Different from MAML, which learns the initialization of the model that a few gradient descent steps will lead to superior performance on a new city, the fine-tuned method focuses on training the model that achieves superior performance on existing source cities. Therefore, the fine-tuned method could obtain a good result only if a new city is similar to source cities, and meta-learning methods have the ability to adapt to various new cities quickly.
- MetaStore methods achieve better performance than MAML methods in most cities. This is because MetaStore methods can acquire city-specific prior knowledge from a variety of source cities in view of multimodal data distribution. However, in Xi'an, meta-learning methods have almost no improvement compared to the fine-tuned method, and the best result of MAML methods is better than the results of MetaStore methods. One possible reason is that Xi'an is similar to source cities, so the fine-tuned method shows good performance. Another possible reason is that MetaStore methods do not learn high-quality city-specific knowledge due to embedding vectors of the city. The results implicitly indicate that we should pay more attention to the embedding vector of the city that contains the characteristic of the city, and then use them to obtain high-quality city-specific knowledge.
- Compared with MetaStore-base, MetaStore-fr improves performance in most cities. The results indicate that MetaStore learns the meta-initialization already providing high-quality feature representations that could be more transferrable. Besides, MetaStore-fr only updates parameters in label predictor instead of all parameters, and it significantly speeds up training time.
- As we can see, MetaStore-fr achieves the best performance compared to baselines in most cities. Since MetaStore-fr learns city-specific prior knowledge from multiple cities with multimodal data distribution. Moreover, an effective learning strategy is adopted to sample

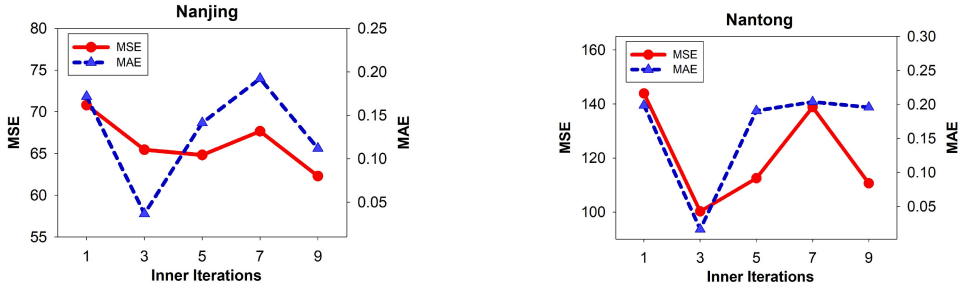

Fig. 4. Impact of different number of inner iterations.

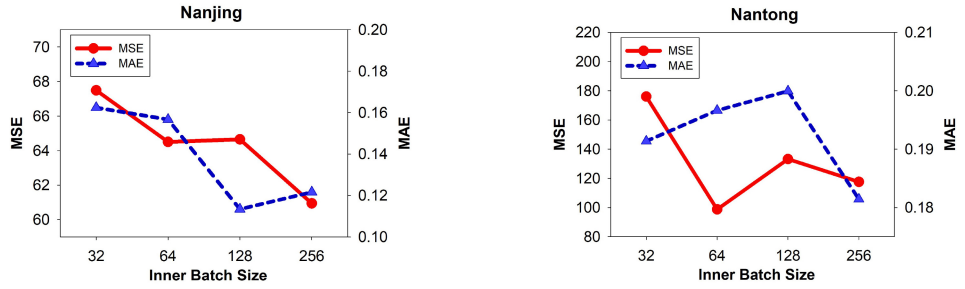

Fig. 5. Impact of different number of inner batch size.

high-quality data to promote convergence. In a nutshell, MetaStore shows superior performance compared with MAML in some cities, which have unique characteristic with multimodal data distribution. For example, the improvement of MetaStore in Chengdu is 30% compared with MAML. For some cities that are similar to source cities, MetaStore-fr still achieves good performance. Thus, learning a better and city-specific initialization is one of effective methods to improve the performance in a new city with limited data.

5.2.2 Hyper-parameter Investigation. We evaluate how different selections of hyper-parameters impact our model. Specifically, we study impacts of two key parameters of MetaStore-fr, i.e., the number of inner iterations and batch size during the local update.

Inner Iterations. Figure 4 shows the performance of our method for varying the number of inner iterations in two cities. We find that the performance dramatically increases at the beginning for two cities. This is because the first local updates might bring the model drops into a local optimum, and then further updates might simply escape from this local optimum. However, the performance then decreases and increases later. One potential reason could be that we sample the final batch (i.e., query set) from the same set of data as the earlier batches (i.e., support set), that is to say, some samples in the final batch could have appeared in previous batches leading to the model close to a local optimum. As shown in Figure 4, the results demonstrate that our proposed method can adapt quickly after a few local updates, such as five local updates and three local updates in Nanjing and Nantong, respectively.

Inner Batch Size. We fixed the number of inner iterations to five and instead varied batch size during the local update. Figure 5 shows the similar change of the performance: The performance increases at the beginning, but decreases a little later, and increases at the end. Specifically, for

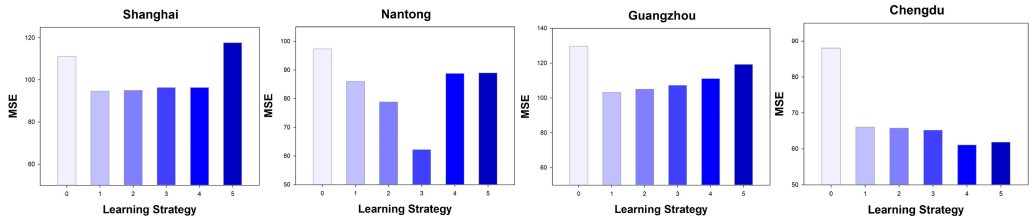

Fig. 6. Results with respect to different learning strategies.

batch sizes greater than 64, the final inner-loop batch for our method necessarily contains samples from the previous batches. In general, as shown in Figures 4 and 5, the proposed method is sensitive to the inner-loop hyperparameters, and the performance significantly drops if parameters are selected in the wrong way.

5.2.3 Impact of the Learning Strategy. We evaluate the effects of different learning strategies for MetaStore-fr by varying the hyper-parameter φ, which controls the learning pace to sample high-quality data instead of observing samples at random. In this experiment, we compare the performance of five learning strategies, as shown in Figure 6. Specifically, 0 means that φ is infinite (i.e., sampling data at random), and φ decreases gradually from levels 1 to 5. It is easy to see that better prediction results can always be achieved by adopting the learning strategy to sample high-quality data for each training batch to improve the performance. We observe that the best setting for φ varies on different cities, and the performance decreases when φ increases in some cities. The reasons may be the total number of training examples decreases when we increase φ to filter out noisy data and there is not enough training data for some cities to train the model.

5.3 Discussions

We next discuss the research findings from this work and potential future directions to improve this work.

End-to-end Learning. The current work pre-trains the embedding vectors of cities to encode the characteristics of different cities, which are learned based on other related consumption data. Then, they are further used to modulate parameters of the prediction model by the attention network to acquire city-specific prior knowledge. In the experiment, however, we observe that it is difficult to guarantee that the embedding vectors of cities learned in advance could improve the prediction model. In our future work, we intend to build an end-to-end learning model, such as adopting the self-attention mechanism or external memory network to learn embedding vectors of cities simultaneously.

Time Series Modeling. In this article, we focus on predicting the consumption behavior of users in a short period of time for store placement. Future studies should explore the dynamic consumer behavior based on the sequence model to further improve the results, such as recurrent neural networks.

Unsupervised Adaptation. In MetaStore, a few examples in a new city are used to fine-tune the model learned by meta-learner for fast adaptation. However, for the cold-start problem that we cannot access any training samples in a new city, it is impossible to quickly adjust the prediction model to obtain a good performance in this city. Therefore, we are planning to study the unsupervised problem by combining meta-learning and unsupervised adaptation.

Interpretable Model. Although we could obtain the weights of the output layer in MetaStore, they cannot disclose the impact of different categories of data, because the deep

feature representations learned by the feature extractors contain low- and high-order feature interactions, which cannot be separated into different categories of data, such as users, location, and communities. In our future work, we intend to improve our model by taking into account interpretability, and we aim to provide not only the predicted value but also the reason why we obtain the value for the commercial company to enhance the trust of the model.

6 CONCLUSION

In this article, we study the optimal store placement problem in new cities with a small number of data. We propose a task-adaptative model-agnostic meta-learning framework, namely, MetaStore, which aims to address the data scarcity problem by transferring prior knowledge learned from multiple source cities. Specifically, we develop a novel task-adaptative meta-learning algorithm to learn city-specific prior knowledge from multiple cities with the multimodal data distribution, which is then transferred to adapt to new cities quickly to obtain superior performance. Finally, we conduct comprehensive experiments on the real-world dataset to compare the performance of MetaStore and baseline methods, and the results demonstrate the superior performance of our proposed approach.

REFERENCES

[1] Adee Athiyaman. 2011. Location decision making: The case of retail service development in a closed population. *Acad. Market. Stud. J.* 15, 1 (2011), 87.

[2] Seth A. Bata, Jonathan Beard, Erica Egri, and David Morris. 2011. Retail revenue management: Applying data-driven analytics to the merchandise line of business. *J. Bus. Retail Manag. Rese.* 5, 2 (2011).

[3] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th International Conference on Machine Learning*. 41–48.

[4] Barry Berman and Joel R. Evans. 1995. *Retail Management: A Strategic Approach*. Ph.D. Dissertation. Univerza v Mariboru, Ekonomsko-poslovna fakulteta.

[5] Erik Brynjolfsson and Andrew Mcafee. 2017. The business of artificial intelligence. *Harv. Bus. Rev.* 7 (2017), 3–11.

[6] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. 2016. Wide & deep learning for recommender systems. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*. 7–10.

[7] Wenyuan Dai, Gui-Rong Xue, Qiang Yang, and Yong Yu. 2007. Transferring naive Bayes classifiers for text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 540–545.

[8] H. Damavandi, N. Abdolvand, and F. Karimipour. 2018. The computational techniques for optimal store placement: A review. In *Proceedings of the International Conference on Computational Science and Its Applications*. Springer, 447–460.

[9] Nedim Dedić and Clare Stanier. 2016. Measuring the success of changes to existing business intelligence solutions to improve business intelligence reporting. In *Proceedings of the International Conference on Research and Practical Issues of Enterprise Information Systems*. Springer, 225–236.

[10] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. 2013. A deep convolutional activation feature for generic visual recognition. *arXiv preprint arXiv:1310.1531* 1 (2013).

[11] Vincent Dumoulin, Jonathon Shlens, and Manjunath Kudlur. 2016. A learned representation for artistic style. *arXiv preprint arXiv:1610.07629* (2016).

[12] Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning*. JMLR.org, 1126–1135.

[13] Jing Gao, Wei Fan, Jing Jiang, and Jiawei Han. 2008. Knowledge transfer via multiple model local structure mapping. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 283–291.

[14] Bin Guo, Huihui Chen, Zhiwen Yu, Xing Xie, Shenlong Huangfu, and Daqing Zhang. 2014. FlierMeet: A mobile crowdsensing system for cross-space public information reposting, tagging, and sharing. *IEEE Trans. Mob. Comput.* 14, 10 (2014), 2020–2033.

[15] Bin Guo, Jing Li, Vincent W. Zheng, Zhu Wang, and Zhiwen Yu. 2018. CityTransfer: Transferring inter-and intra-city knowledge for chain store site recommendation based on multi-source urban data. *Proc. ACM Interact., Mob., Wear. Ubiq. Technol.* 1, 4 (2018), 1–23.

[16] Minyoung Huh, Shao-Hua Sun, and Ning Zhang. 2019. Feedback adversarial learning: Spatial feedback for improving generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1476–1485.

[17] Pablo Jensen. 2006. Network-based predictions of retail store commercial categories and optimal locations. *Phys. Rev. E* 74, 3 (2006), 035101.

[18] Dmytro Karamshuk, Anastasios Noulas, Salvatore Scellato, Vincenzo Nicosia, and Cecilia Mascolo. 2013. Geo-spotting: Mining online location-based services for optimal retail store placement. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 793–801.

[19] Tero Karras, Samuli Laine, and Timo Aila. 2019. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4401–4410.

[20] M. Pawan Kumar, Benjamin Packer, and Daphne Koller. 2010. Self-paced learning for latent variable models. In *Proceedings of the International Conference on Advances in Neural Information Processing Systems*. 1189–1197.

[21] Hoyeop Lee, Jinbae Im, Seongwon Jang, Hyunsouk Cho, and Sehee Chung. 2019. MeLU: Meta-learned user preference estimator for cold-start recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1073–1082.

[22] Jing Li, Bin Guo, Zhu Wang, Mingyang Li, and Zhiwen Yu. 2016. Where to place the next outlet? Harnessing cross-space urban data for multi-scale chain store recommendation. In *Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing*. 149–152.

[23] Yan Liu, Bin Guo, Nuo Li, Jing Zhang, Jingmin Chen, Daqing Zhang, Yinxiao Liu, Zhiwen Yu, Sizhe Zhang, and Lina Yao. 2019. DeepStore: An interaction-aware wide&deep model for store site recommendation with attentional spatial embeddings. *IEEE Internet Things J.* 6, 4 (2019), 7319–7333.

[24] Zhaoyang Liu, Yanyan 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.

[25] Lilyana Mihalkova, Tuyen Huynh, and Raymond J. Mooney. 2007. Mapping and revising Markov logic networks for transfer learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 608–614.

[26] Risto Miikkulainen, Jason Liang, Elliot Meyerson, Aditya Rawal, Daniel Fink, Olivier Francon, Bala Raju, Hormoz Shahrzad, Arshak Navruzyan, Nigel Duffy et al. 2019. Evolving deep neural networks. In *Artificial Intelligence in the Age of Neural Networks and Brain Computing*. Elsevier, 293–312.

[27] Volodymyr Mnih, Nicolas Heess, Alex Graves, et al. 2014. Recurrent models of visual attention. In *Proceedings of the International Conference on Advances in Neural Information Processing Systems*. 2204–2212.

[28] Tsendsuren Munkhdalai and Hong Yu. 2017. Meta networks. In *Proceedings of the 34th International Conference on Machine Learning*. JMLR. org, 2554–2563.

[29] Jialin Pan. 2010. *Feature-based Transfer Learning with Real-world Applications*. Ph.D. Dissertation. Hong Kong University of Science and Technology.

[30] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. 2019. Semantic image synthesis with spatially adaptive normalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2337–2346.

[31] Ethan Perez, Harm De Vries, Florian Strub, Vincent Dumoulin, and Aaron Courville. 2017. Learning visual reasoning without strong priors. *arXiv preprint arXiv:1707.03017* (2017).

[32] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. 2017. Film: Visual reasoning with a general conditioning layer. *arXiv preprint arXiv:1709.07871* (2017).

[33] Aniruddh Raghu, Maithra Raghu, Samy Bengio, and Oriol Vinyals. 2019. Rapid learning or feature reuse? Towards understanding the effectiveness of MAML. *arXiv preprint arXiv:1909.09157* (2019).

[34] Sachin Ravi and Hugo Larochelle. 2016. Optimization as a model for few-shot learning. (2016).

[35] Olivia Parr Rud. 2009. *Business Intelligence Success Factors: Tools for Aligning Your Business in the Global Economy*. Vol. 18. John Wiley & Sons.

[36] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip H. S. Torr, and Timothy M. Hospedales. 2018. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1199–1208.

[37] Vivienne Sze, Yu-Hsin Chen, Tien-Ju Yang, and Joel S. Emer. 2017. Efficient processing of deep neural networks: A tutorial and survey. *Proc. IEEE* 105, 12 (2017), 2295–2329.

[38] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the International Conference on Advances in Neural Information Processing Systems*. 5998–6008.

[39] Feng Wang, Li Chen, and Weike Pan. 2016. Where to place your next restaurant? Optimal restaurant placement via leveraging user-generated reviews. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. 2371–2376.

[40] Hongjian Wang, Xianfeng Tang, Yu-Hsuan Kuo, Daniel Kifer, and Zhenhui Li. 2019. A simple baseline for travel time estimation using large-scale trip data. *ACM Trans. Intell. Syst. Technol.* 10, 2 (2019), 1–22.

[41] Jindong Wang, Yiqiang Chen, Wenjie Feng, Han Yu, Meiyu Huang, and Qiang Yang. 2020. Transfer learning with dynamic distribution adaptation. *ACM Trans. Intell. Syst. Technol.* 11, 1 (2020), 1–25.

[42] Jiangtao Wang, Yasha Wang, Daqing Zhang, Qin Lv, and Chao Chen. 2019. Crowd-powered sensing and actuation in smart cities: Current issues and future directions. *IEEE Wirel. Commun.* 26, 2 (2019), 86–92.

[43] Leye Wang, Xu Geng, Xiaojuan Ma, Feng Liu, and Qiang Yang. 2018. Cross-city transfer learning for deep spatio-temporal prediction. *arXiv preprint arXiv:1802.00386* (2018).

[44] Hua Wei, Guanjie Zheng, Huaxiu Yao, and Zhenhui Li. 2018. Intellilight: A reinforcement learning approach for intelligent traffic light control. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2496–2505.

[45] Saining Xie, Sainan Liu, Zeyu Chen, and Zhuowen Tu. 2018. Attentional shapecontextnet for point cloud recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4606–4615.

[46] Mengwen Xu, Tianyi Wang, Zhengwei Wu, Jingbo Zhou, Jian Li, and Haishan Wu. 2016. Demand driven store site selection via multiple spatial-temporal data. In *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. 1–10.

[47] Mengwen Xu, Tianyi Wang, Zhengwei Wu, Jingbo Zhou, Jian Li, and Haishan Wu. 2016. Store location selection via mining search query logs of Baidu maps. *arXiv preprint arXiv:1606.03662* (2016).

[48] Yanan Xu, Yanyan Shen, Yanmin Zhu, and Jiadi Yu. 2020. AR2Net: An attentive neural approach for business location selection with satellite data and urban data. *ACM Trans. Knowl. Discov. Data* 14, 2 (2020), 1–28.

[49] 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*. 2181–2191.

[50] Fei Yi, Zhiwen Yu, Fuzhen Zhuang, and Bin Guo. 2019. Neural network based continuous conditional random field for fine-grained crime prediction. In *Proceedings of the International Joint Conferences on Artificial Intelligence*. 4157–4163.

[51] Jian Zeng and Bo Tang. 2019. Mining heterogeneous urban data for retail store placement. In *Proceedings of the ACM Turing Celebration Conference*. 1–5.

[52] Xi Sheryl Zhang, Fengyi Tang, Hiroko H. Dodge, Jiayu Zhou, and Fei Wang. 2019. MetaPred: Meta-learning for clinical risk prediction with limited patient electronic health records. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2487–2495.

Received August 2020; revised December 2020; accepted January 2021