Knowledge transfer with weighted adversarial network for cold-start store site recommendation

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

Knowledge Transfer with Weighted Adversarial Network for Cold-Start Store Site Recommendation

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

Store site recommendation aims to predict the value of the store at candidate locations and then recommend the optimal location to the company for placing a new brick-and-mortar store. Most existing studies focus on learning machine learning or deep learning models based on large-scale training data of existing chain stores in the same city. However, the expansion of chain enterprises in new cities suffers from data scarcity issues, and these models do not work in the new city where no chain store has been placed (i.e., cold-start problem). In this article, we propose a unified approach for cold-start store site recommendation, Weighted Adversarial Network with Transferability weighting scheme (WANT), to transfer knowledge learned from a data-rich source city to a target city with no labeled data. In particular, to promote positive transfer, we develop a discriminator to diminish distribution discrepancy between source city and target city with different data distributions, which plays the minimax game with the feature extractor to learn transferable representations across cities by adversarial learning. In addition, to further reduce the risk of negative transfer, we design a transferability weighting scheme to quantify the transferability of examples in source city and reweight the contribution of relevant source examples to transfer useful knowledge. We validate WANT using a real-world dataset, and experimental results demonstrate the effectiveness of our proposed model over several state-of-the-art baseline models.

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

Additional Key Words and Phrases: Urban computing, cold-start problem, store site recommendation, transfer learning, neural networks

ACM Reference format:
Yan Liu, Bin Guo, Daqing Zhang, Djamal Zeghlache, Jingmin Chen, Ke Hu, Sizhe Zhang, Dan Zhou, and Zhiwen Yu. 2021. Knowledge Transfer with Weighted Adversarial Network for Cold-Start Store Site Recommendation. *ACM Trans. Knowl. Discov. Data* 15, 3, Article 47 (April 2021), 27 pages.
https://doi.org/10.1145/3442203

This work was supported by the National Science Fund for Distinguished Young Scholars (grant no. 62025205), the National Natural Science Foundation of China (grant nos. 61772428 and 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, K. Hu, S. Zhang, and D. Zhou, Alibaba Group, Hangzhou, China; emails: {jingmin.cjm, huke.huke, jincheng.zsz, modan.zd}@alibaba-inc.com; Z. Yu, Northwestern Polytechnical University, Xi'an, China; email: zhiwenyu@nwpu.edu.cn.

1556-4681/2021/04-ART47 $15.00
https://doi.org/10.1145/3442203

1 INTRODUCTION

Store site recommendation is one of the essential business services in smart cities for the company (e.g., chain enterprises) to evaluate candidate locations and select the optimal one for placing a new brick-and-mortar store. Traditional store placement methods rely on many professionals to make a detailed investigation of demographics and human flow statistics from all candidate places, which are time-consuming and do not scale up well. In recent years, with the development of internet technology and mobile devices, a large amount of user-generated data in cities has grown explosively, which provides new opportunities to the data-driven methods for store site recommendation. Specifically, data-driven store site recommendation aims to leverage the techniques of data analysis or machine learning to mine users' preferences and predict the popularity of candidate locations based on large-scale data for identifying the most promising one.

Traditionally, some basic regression models based on extracted features are used for store site recommendation. For example, Karamshuk et al. [16] mine useful features from check-in data considering of geographic and user mobility factors, and then adopt supervised machine learning to predict the popularity of retail stores at a set of candidate areas for a new store. However, these approaches depend on expertise feature engineering to analyze complex factors and extract available features from a single data source, which fail to characterize complicated influences from multiple factors and learn feature interactions from multi-source data. Recently, with the rapid development of deep learning techniques, many deep neural networks (DNNs) are used to learn deep representations from raw data. More and more works thus resort to the advanced deep learning methods to improve the performance of store site recommendation by learning deep feature interactions and modeling consumer behaviors from multi-source data. For example, Liu et al. [18] propose the unified interaction-aware model with attentional spatial embedding for store site recommendation, which aims to learn low- and high-order feature interactions based on latent feature representations.

However, most existing studies are conditioned on large-scale labeled data and learn a model based on the features of existing chain stores in the same city, these supervised methods, thus, suffer from the data scarcity and cold-start issues in many practical applications. For example, a chain enterprise usually lacks historical consumption data in some new cities where no chain store has been placed, and most of previous works fail to address the cold-start store site recommendation when a chain enterprise extends its business in a new city. Therefore, it motivates us to exploit enough data in other cities to solve this problem. The main challenge is that most supervised learning models are based on the assumption that the training and testing data are sampled from the same feature space with the same data distribution, but this assumption cannot hold because the data distributions vary from city to city due to different characteristics of multiple cities (e.g., point of interest (POI) distribution and road networks), thus a prediction model built for one city may not predict well in other cities because of different data distributions.

Transfer learning [26] has been proved to be an effective method to transfer knowledge across different domains in various applications. Recently, many works adopt transfer learning to deal with data scarcity by transferring available knowledge from source cities with rich training data to improve the performance in the data-scarce target city. Wang et al. [41] propose to transfer knowledge from a source city to a target city for spatio-temporal prediction tasks, by learning a matching function to match the region in target city to a similar source region. Guo et al. [13] propose a two-fold knowledge transfer framework to transfer inter-city and intra-city knowledge to solve the store placement recommendation in a new city for chain enterprises. However, these transfer learning methods mainly rely on building correspondence between two cities via the region matching function to transfer knowledge, which are hard to extend to solve our problem

because the naive function cannot build the correspondence of sophisticated consumer behavior. In addition, these models ignore the distribution shift and cannot reduce the effect of distribution discrepancy between source and target cities, which could lead to negative transfer when data distributions in source and target cities are significantly different.

Domain adaptation, a special scenario of transfer learning, is highly desirable to reduce the effect of domain discrepancy, which aims to minimize the domain gap and transfer knowledge across domains [8, 25]. Recently, most existing studies on domain adaptation learn domain-invariant representations by minimizing the distribution distance between two domains, or learning the feature representation that cannot be distinguished by adversarial learning. Although many domain adaptation methods show the superior performance in various computer vision and natural language processing tasks [7, 32, 34, 37], very few attempts have applied domain adaptation methods to transfer city knowledge from source city to target city in urban applications due to two key technical challenges: (1) *How to bridge different cities with various data distributions?* The complex data distributions of different cities in practical applications lead to the distribution discrepancy, and naively transferring knowledge between two cities may hurt the performance in the target city if the data distributions in source and target cities are significantly different. (2) *How to reduce negative transfer of useless source samples?* Existing domain adaptation methods mainly align the data distribution in the entire feature space to reduce distribution discrepancy between two domains, which will further trigger the negative effects of untransferable source samples on domain alignment. Intuitively, some examples in source city should not be transferred to the target city, such as low-quality examples or irrelevant examples, and directly aligning all examples in two cities could cause negative transfer.

To address the aforementioned challenges, we propose a unified framework for the cold-start problem, Weighted Adversarial Network with Transferability weighting scheme (WANT), to learn disentangled and transferable feature representations and transfer knowledge between two cities. In contrast to other methods, WANT is capable of reducing the risk of negative transfer by selecting transferable source examples. To solve the first challenge, we develop a domain discriminator to minimize the domain discrepancy by adversarial learning in order to bridge different cities with different data distributions. Specifically, the domain discriminator distinguishes the source data from the target data, and plays the minimax game with the feature extractor to guide it to learn transferable and domain-invariant feature representations across cities to promote positive transfer. To tackle the second challenge, we design a transferability weighting mechanism to highlight the contribution of useful source examples to the training of the transfer model in order to prevent negative transfer. Specifically, WANT first automatically quantifies the transferability of each source example with the weighting scheme based on the quality of source example and the similarity of source example to target data, which is then used to weigh its contributions to both the domain discriminator and the label predictor for transferring useful source examples.

In summary, the main contributions of this article are:

- To the best of our knowledge, this is the first work that studies how to transfer useful knowledge from a data-rich source city to a target city with no labeled data at both the feature level and the instance level on account of transfer tasks in some practical spatio-temporal applications.
- We propose a unified transfer framework, WANT, which effectively enables knowledge transfer across cities to solve cold-start store site recommendations. In contrast to existing transfer learning methods, WANT learns transferable and invariant feature representations itself across cities by adversarial learning, and transfers useful source examples to improve the performance of knowledge transfer in target city.

—We design a transferability weighting mechanism to quantify the transferability of examples in source city, and highlight their contributions to knowledge transfer to mitigate negative transfer. In particular, our proposed weighting scheme is capable of reducing negative effects of untransferable source examples to make the model more efficient and robust.
—Extensive experiments on the real-world dataset show the effectiveness and efficiency of the proposed method compared to several state-of-the-art models.

The remainder of this article is organized as follows. We begin by reviewing the related work in Section 2. We present an overview of our proposed framework in Section 3. Section 4 elaborates the detailed design of the proposed model. Empirical evaluation and discussion are reported in Section 5, while the conclusion is enclosed in Section 6.

2 RELATED WORK

In this section, we briefly review the works in two categories: store site recommendation and transfer learning.

2.1 Store Site Recommendation

In recent years, the proliferation of multi-source urban data has fostered unprecedented opportunities to the data-driven store placement approaches, which aim to analyze and mine users' preferences based on user-generated data to select the optimal location for placing a new brick-and-mortar store.

The earliest store placement methods are based on some basic regression models. For example, Karamshuk et al. [16] 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. [17] first extract some associative features from cross-space data sources, and then adopt supervised regression and classification to solve two scale-specific chain store placement problems. In [39], the authors consider three types of features: review-based attractiveness, review-based competitiveness, and geographic features of a location, which are used to predict the number of check-ins at a candidate location by the regression model. Zeng et al. [44] 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 expertise feature engineering to characterize sophisticated influences and extract features from a single data source, which hardly generalize to other applications and fail to learn complex deep feature interactions from multi-source data.

Recently, with the rapid development of DNNs [23, 24], more and more works adopt DNNs to improve the performance of store placement, which can characterize complex consumption behavior by learning deep feature representations based on multi-source data. Liu et al. [18] propose a model named DeepStore, which consists of the cross network, the deep network, and the linear component, thus, it can learn low- and high-order feature interactions explicitly and implicitly to model complex user behavior. Xu et al. [43] propose an attentive neural method to predict the business popularity of a given location. Specifically, it consists of three attention modules to learn deep feature interactions based on the discriminative features extracted from urban data and satellite data.

However, most existing studies learn the prediction model based on the large-scale labeled data of existing chain stores in the same city, and these supervised methods suffer from the data scarcity and cold-start issues in many practical applications. For example, a chain enterprise usually lacks historical consumption data in some new cities where no chain store has been placed. Therefore, the expansion of chain enterprise in new cities is faced with the cold-start problem, and most of existing works fail to address this cold-start store site recommendation. Different from most of

previous works based on a large set of training samples, we aim to tackle the cold-start problem for store site recommendation, by transferring knowledge learned from rich labeled training data in source city to improve the performance in a new city.

2.2 Transfer Learning

In recent years, transfer learning has been studied as an effective solution to address the data scarcity problem by avoiding expensive data labeling efforts [35]. Different from traditional machine learning based on the assumption that training data and testing data are sampled from the same feature space with the same data distribution, which does not always hold in many practical applications. Transfer learning allows the domains, tasks, and distributions used in training data and testing data to be different. In general, the objective of transfer learning is to transfer the knowledge from some source tasks/domains to the target task/domain when the latter does not have enough training data [26].

2.2.1 Transfer Learning in Urban Computing. There have been recently a few works that leverage transfer learning to deal with the data scarcity in urban computing [42]. The main challenge in urban applications is that different cities usually have different distributions, which has posed a major bottleneck for adapting prediction models across cities. Intuitively, one typical solution is to choose similar source examples for transferring. For example, in [41], the authors present cross-city transfer learning method to solve spatio-temporal prediction problems, which transfers knowledge from a data-rich source city to a data-scarce target city by learning an inter-city region matching function in the label space to match each region in target city to a similar source city region. However, in cold-start store site recommendation, it is impossible to build the relationship between two cities in label space because the target city has no labeled information. Another possible way is to match two cities between data distributions in terms of the feature space. Guo et al. [13] propose a two-fold knowledge transfer framework called CityTransfer to solve the cold-start problem in chain store site recommendation. Specifically, CityTransfer first builds correspondence between different regions in source and target cities to bridge the distribution discrepancy, and then transfers chain store knowledge from a relevant source city with rich knowledge.

However, there are some difficulties to adopt previous transfer learning methods to solve our problem. First, some traditional transfer learning methods are not designed for deep learning model, which fail to address some complex problems in urban computing. Second, most of the previous methods mainly focus on transferring knowledge by building correspondence between the source and target regions, which are hard to extend to solve our problem, because naive matching function (e.g., Pearson correlation coefficient (PCC)) cannot build the relationship of sophisticated user consumption behavior. Third, knowledge transfer based on the matching function between source city and target city could lead to negative transfer if data distributions in two cities are significantly different, because they ignore the distribution shift and are not capable of reducing the distribution discrepancy.

Recently, some effective transfer learning methods have been proposed to solve the cold-start recommendation problems in social networks. Qi et al. [29] predict future links in a growing network with the use of the existing network structure to perform cross-network link inference by transfer learning methods, and a network re-sampling technique is proposed for calibrating the portions of the source network to be used in the transfer process. Wang et al. [40] propose to exploit media contents and link structures between users and groups to automatically recommend groups to users. However, it is not feasible to solve the cold-start store site recommendation problem via this type of cross-network transfer learning methods in social networks. First, the proposed methods are mainly based on the link structure or media contents in social networks, but the additional

linkage information in networks or enough content information is unavailable in most spatio-temporal scenarios (e.g., site recommendation), which cannot be further used to learn transferable and structural knowledge of different networks. In addition, most networks in existing cold-start recommendations between social networks are homogeneous, which only model the structural information in terms of one type of nodes. But in the store site recommendation problem, the linkage behavior between people and places should also be taken into consideration. Last but not least, the store site recommendation focuses on predicting the complex consumer behavior of people in each candidate place, not just inferring the links between different nodes.

2.2.2 Transfer Learning on Domain Adaptation. Domain adaptation, a special scenario of transfer learning under the domain shift between training and testing data distributions, which aims to reduce the effect of distribution discrepancy to transfer knowledge across domains [8, 21, 25]. In recent years, most studies on domain adaptation apply DNNs to learn domain-invariant feature representations because of its advantage of learning more transferable representations [4, 10, 11, 27].

Previous works focus on learning transferable representation by minimizing the distribution distance between two domains. Tzeng et al. [38] propose a convolutional neural network (CNN) architecture to solve both supervised and unsupervised adaptation, which adds an adaptation layer into the deep neural network along with an additional domain confusion loss based on the Maximum Mean Discrepancy (MMD) to learn domain-invariant representation. In [20], a Deep Adaptation Network (DAN) model is proposed, which extends the deep CNN for domain adaptation applications. Particularly, DAN learns transferable features by reducing the multi-kernel MMD of hidden representations of task-specific layers between source and target domains in reproducing kernel Hilbert space. Long et al. [21] propose joint adaptation networks to learn transferable features, which can reduce the effect of domain discrepancy based on a joint MMD criterion in joint distributions of the representations of multiple domain-specific layers across domains.

Recently, inspired by the idea of adversarial learning [22], adversarial domain adaptation methods have gained growing interest, which aim to learn the domain-invariant feature representation that cannot be distinguished by adversarial learning [31, 46]. Ganin et al. [8] propose the Domain-Adversarial Neural Networks (DANN) for domain adaptation in deep architectures, which can jointly learn discriminative and invariant features. Specifically, it adds a sub-network as the domain discriminator to distinguish source and target data, which makes the feature extractor to learn transferable features to confuse the domain discriminator by adversarial learning. Tzeng et al. [37] propose an Adversarial Discriminative Domain Adaptation (ADDA) model, and it allows independent source and target feature extractors with unshared weights, which is flexible to learn more domain-specific features. Cao et al. [2] present a Selective Adversarial Network (SAN) for partial transfer learning, which transfers knowledge from existing large-scale domains to small-scale domains. In particular, SAN includes multiple class-wise domain discriminators, and each domain discriminator is applied to align the source and target domain data associated with different labels to reduce negative transfer. Zhang et al. [45] propose an Importance Weighted Adversarial Nets (IWAN) model, which is capable of detecting outlier classes in source domain by an additional domain classifier to reduce the domain shift in partial domain adaptation. Shu et al. [33] propose a Transferable Curriculum Learning (TCL) method, which combines curriculum learning and adversarial learning to learn a robust model with transferable curriculum to deal with noisy data for weakly-supervised domain adaptation. Cao et al. [3] design an Example Transfer Network (ETN) for partial domain adaptation, and a weighting scheme is proposed for classification problems to promote positive transfer by identifying outlier classes in the source domain. Although these works have explored the transfer models in different fields, we also note that the

improvement of our work over these representative models. In this work, we aim to improve the DANN by designing a transferability weighting mechanism to transfer useful source samples, because DANN directly matches all examples in the source domain, which could cause negative transfer. Generally, the knowledge learned from some useless samples in the source city could hurt the performance of the model in the target city. In addition, we wish to utilize a shared feature extractor for both source and target cities, instead of including the independent source and target feature extractors with unshared weights in ADDA. Furthermore, our goal differs from most existing transfer models (e.g., IWAN, TCL, and ETN) for classification tasks, instead, we focus on establishing a unified transfer model for most regression tasks in practical scenarios, which is capable of reducing negative transfer of both dataset shift and noisy source examples to make the model more efficient and robust.

Most current works on transfer learning have explored how to avoid negative transfer. Qi et al. [30] present a joint Intermodal and Intramodal Label Transfer algorithm from texts to images for image classification tasks, which combines the advantages of both image labels and text labels in the context of a label transfer task to prevent the negative transfer. [28] proposes a cross-category label propagation approach that learns and leverages cross-category label correlations to transfer knowledge from different source categories to the target category. Nevertheless, these transfer learning methods are explored to reduce negative transfer by weighting the samples in the source domain based on label correlations in classification tasks, which are not effective to solve some complex regression tasks in most practical scenarios, because some extra information (e.g., label correlations) cannot be fully utilized and some crucial factors are not considered, which could lead to negative transfer. Different from most existing transfer learning methods that are proposed to reduce feature distribution shift or label distribution shift for classification tasks, we aim to propose a unified transfer learning approach for regression tasks to reduce negative transfer of both dataset shift and noisy source examples.

In recent years, most of domain adaptation methods are applied in the field of computer vision and natural language processing [7, 32, 34, 37], very few attempts have been made on transferring knowledge in urban applications due to complex data distributions among different cities. Liu et al. [19] propose a City Domain Adaptation Network named ConvCDAN for hotspots detection in a new city, consisting of a FeatureNet, a DensityNet, and a DomainNet, which can transfer hotspots knowledge learned from one source city with shared bikes. Nevertheless, ConvCDAN simply aligns the data distribution in the entire feature space between two cities, which could result in the negative transfer, because some irrelevant examples in source city should not be transferred to the target city.

Inspired by above-mentioned works, we aim to take full advantage of domain adaptation approaches to solve cold-start store site recommendation, by transferring knowledge learned from a source city with enough labeled data to improve the performance in a target city with no labeled data. Different from existing works, we design a unified framework for cold-start store site recommendation that can learn disentangled and transferable feature representations to transfer useful knowledge learned from the data-rich source city to a new city with unlabeled data. In addition, we present a transferability weighting mechanism, which is capable of reducing the risk of the negative transfer by reweighting the contribution of each source example.

3 OVERVIEW

In this section, we begin by introducing the definitions and the problem statement. Next, we extract useful features from multi-source data. Based on extracted features, we further present comprehensive analysis results. Finally, we describe the framework of the proposed model. For brevity, we present a table of notations used in our work in Table 1.

Table 1. Notations

Notation	Description
c_s, c_t	Source city, target city
$\mathcal{D}_s = \{(\mathbf{x}_k^s, y_k^s)\}_{k=1}^{n_s}$	Source dataset including n_s labeled examples in source city c_s
$\mathcal{D}_t = \{\mathbf{x}_k^t\}_{k=1}^{n_t}$	Target dataset including n_t unlabeled examples in target city c_t
y_{m_i,l_j}	The amount of consumption per community m_i in the store located at l_j
G_f	The function of feature extractor with parameters θ_f
G_y	The function of label predictor with parameters θ_y
G_d	The function of adversarial domain discriminator with parameters θ_d
G_t	The function of non-adversarial domain discriminator with parameters θ_t
w	The transferable weight of the source example

3.1 Problem Formulation

Definition 3.1 (Location-based Community in a City). Considering that geographical factor is one of the main impacts of consumer behavior in brick-and-mortar stores, thus, we spatially divide a city into a set of location-based communities. Specifically, each user in the city is associate with a community (i.e., a housing estate), which is a group of homes and other buildings built together.

Definition 3.2 (User Consumption in a Store). In a city c, given a store s_j located at l_j and the set of nearby communities $\mathcal{M} = \{m_1, m_2 \dots m_i \dots\}$ that people in each community have the possibility to consume in the store. The user consumption in store s_j is denoted as $\mathcal{Y}_{s_j} = \{y_{m_i,s_j} | m_i \in \mathcal{M}\}$, where y_{m_i,s_j} is the amount of consumption per community m_i in store s_j during a given period of time T, which is to be predicted in view of the information of users and stores. Then, the overall sale of the store s_j can be represented as the total amount of consumption of all potential consumers in this store $\mathcal{S}(s_j) = \sum_{m_i \in \mathcal{M}} y_{m_i,s_j}$.

Problem Definition: ***Store site recommendation in a new city.*** For a chain enterprise, suppose that we have a source city c_s with enough consumption data and a target city c_t where no chain store has been placed yet, our goal in this article is to solve the cold-start store site recommendation for this chain enterprise in a new city by leveraging and transferring available knowledge learned from the source city.

Specifically, given the set of candidate places $\mathcal{L}^t = \{l_1^t, l_2^t \dots l_j^t \dots\}$, the set of communities $\mathcal{M}^t = \{m_1^t, m_2^t \dots m_i^t \dots\}$, and $\mathcal{D}_t = \{\mathbf{x}_k^t\}_{k=1}^{n_t}$ with n_t unlabeled examples in target city c_t, as well as $\mathcal{D}_s = \{(\mathbf{x}_k^s, y_k^s)\}_{k=1}^{n_s}$ with n_s labeled examples in source city c_s. The objective of our problem is to predict user consumption in each store located at candidate places in the target city, denoted as $\hat{y}_{m_i^t,l_j^t}$.

Known the predicted consumption behavior, we then compute the overall sale of the store s_j located at each candidate place l_j, $\hat{\mathcal{S}}(l_j) = \sum_{m_i^t \in \mathcal{M}^t} \hat{y}_{m_i^t,l_j^t}$. Finally, the store site recommendation can be solved by selecting the optimal location with the highest sale $\hat{\mathcal{S}}$ from $\mathcal{L}^t$ to place a new store in the target city.

3.2 Feature Extraction

In this work, we choose a chain retail enterprise for a case study, which owns many brick-and-mortar stores in some cities of China. Specifically, multi-source urban data is collected in view of three major stakeholders, including chain retail enterprise, potential consumers and POIs. *Chain retail enterprise data* contains profile information (e.g., name and location) and historical sale information (e.g., consumers and their expenditure). *User data* includes user location information

and profile information (e.g., gender, age, and profession). It should be noted that we associate each user with a community in order to protect user privacy, and only obtain the statistical information about users in the community. *POI data* contains the characteristics (e.g., name, location, and category) of different places related to people's consumption behavior, such as shop, food, and transport facilities.

For each store in a city, we aim to predict the total amount of consumption that people in each community consume in the store. Therefore, for each <*store*, *community*> pair, we treat it as an entity and extract useful features. Specifically, we mainly consider the following features extracted from multi-source data, which are classified into four categories: user features, geographic features, commercial features, and time features.

User Features. The possibility of people consuming in chain stores depends on the user attributes to a great extent. Furthermore, the amount of consumption in the store is closely related to the user's income level. Therefore, to characterize potential consumers in different communities around the store, we take some demographic profiles into consideration, and further obtain the statistical information about the number of people with different profiles in each community, such as the number of men or women in the community and the number of people on different income levels.

Geographic Features. In urban areas, the popularity of a store is related to spatial characteristics of the place where the store resides. Inspired by [13], we consider the following features to represent the geographic characteristics of a store and its surrounding area, which is a disc centered at the store with radius r. *Distance*: we consider the Manhattan distance between the store and the community where people live. *Traffic convenience*: we use the total number of public transportation stations (including bus stations and subway stations) in the surrounding area to denote the traffic convenience. *POI set*: although our prediction objects are retail shops, we consider all types of POIs (e.g., shop, food, and company) that could attract potential consumers, and compute the number of POIs of each category in the surrounding area. *Neighborhood Entropy*: it refers to an entropy measure [5] of the frequency of place categories near the store, assessing the spatial heterogeneity of the area around the store. A high entropy value indicates more diversity towards a lot of POI categories.

Commercial Features. The commercial environment around the store impacts the performance of the store, we thus extract the following three commercial features. *Density*: it refers to the total number of neighbors (i.e., market and restaurant) around the store, which could reflect what extent the popularity of a place. *Competitiveness*: known the type of the chain store, the competitive relationship of the store is defined as the proportion of neighboring places belonging to the same category with respect to the total number of places located in the surrounding area. *Jensen Quality*: we consider the complementarity relationship between different categories in the same area, which is measured by the Jensen Quality [14] to access the spatial interactions of the places with respect to their ability to attract other places of certain types.

Time Features. We consider time features to capture the temporal profile of the chain store and the temporal pattern of consumer behavior. Intuitively, we use the date of establishment (e.g., the year) and the number of existing stores to evaluate the popularity of the brand in users. In addition, we consider the date when users consume in the store to reflect the consumption habits, such as seasonal characteristics and the holiday (e.g., New Year's Day and National Day).

3.3 Data Analysis

3.3.1 Feature Correlation Analysis. To understand the impact of different features on chain store site selection, we analyze the feature correlation between the amount of consumption and different types of features in multiple cities, including two tier-1 cities (Beijing and Shanghai) and two tier-2

Table 2. The Impact of User Features with PCC in Different Cities

User features		Beijing	Shanghai	Chengdu	Xi'an
Gender	0	0.20	0.09	0.15	0.16
	1	−0.17	−0.11	−0.14	−0.13
Age	0	−0.10	−0.09	−0.14	−0.13
	1	0.03	−0.05	0.01	−0.06
	2	0.13	0.14	0.15	0.14
	3	0.07	0.01	0.11	0.19
Consumption level	0	−0.18	−0.08	−0.19	−0.22
	1	−0.19	−0.18	−0.20	0.24
	2	0.23	0.17	0.27	0.38
Car	0	−0.17	−0.09	−0.17	−0.21
	1	0.17	0.09	0.18	0.21

cities (Chengdu and Xi'an). Specifically, to guarantee the comprehensive analysis, we first compute the PCC, and then present the feature importance in GBDT model for different cities.

Table 2 lists the PCC between the amount of consumption and user features. More specifically, to protect user privacy, we just present the impact of different types in each user attribute, including gender, age, consumption level, and car. For example, different kinds of age attribute refer to different age groups, such as kid, youth, and old age. In the car attribute, 0 represents that people have the private car or not, and 1 represents the opposite. It should be noted that the main reason that the coefficients are all below 0.5 could be that the complex feature interactions from multi-source data play the important role in optimal site selection instead of a single feature, and PCC measures the linear correlation between the amount of consumption and one of the features. Intuitively, from the statistic results, we can observe that one gender has a positive linear correlation, and the other has a negative linear correlation with the amount of consumption for all four cities. Similarly, in all four cities, different consumption levels also have various correlations, such as level 2 has a significant positive linear correlation with the amount of consumption. *The results indicate that the potential consumer groups of the same chain enterprise are similar in different cities, which inspires us to transfer enterprise bias knowledge learned from the source city to the target city for chain store site selection.*

Furthermore, to obtain a better insight into the impact of different features in chain store site selection, we adopt the gradient boosting decision tree (GBDT) model, one of the effective boosted tree models, which is capable of modeling feature interactions and obtaining non-linear correlation, to compute the feature importance in different cities. Specifically, we first train the unique GBDT model under the same setting based on extracted features for each city, and then compute the importance value of feature X_i by variable importance measures [36].

Specifically, given n trees in GBDT, for a single tree T_j, the measure of variable importance X_i is defined as follows:

$$VIM(X_i, T_j) = \sum_{m \in T_j} \Delta I(X_i, m), \tag{1}$$

where $\Delta I(X_i, m)$ is the decrease in impurity as a result of an actual split on variable X_i at a node m of the optimally pruned tree T_j. Node impurity for our regression problem is defined as:

$$I(m) = \sum_{i \in m} \frac{(y_i - \overline{y})^2}{N(m)}, \tag{2}$$

Table 3. Feature Importance in GBDT for Different Cities

Features		Beijing	Shanghai	Chengdu	Xi'an
Distance		0.168	0.108	0.119	0.185
User attributes	Gender	0.072	0.028	0.037	0.015
	Age	0.082	0.082	0.193	0.154
	Consumption level	0.156	0.127	0.162	0.164
	Car	0.037	0.015	0.026	0.031
POI attributes	Transport	0.004	0.004	0.003	0.003
	Company	**0.009**	0.003	**0.006**	**0.010**
	Shopping	0.004	**0.006**	0.002	0.004
	Food	0.003	0.004	0.003	0.004
	Hotel	**0.016**	**0.007**	0.003	**0.006**
	Sport	0.007	0.002	**0.004**	0.003

where the sum and mean are taken over all observations i in node m, and $N(m)$ is the number of observations in node m.

Therefore, the importance value of feature X_i in GBDT is simply averaged over n trees:

$$VIM(X_i) = \frac{1}{n}\sum_{j=1}^{n} VIM(X_i, T_j). \tag{3}$$

Table 3 shows the feature importance in GBDT for four different cities. In particular, we compute the total importance value of each user attribute, instead of presenting values of all types in each user attribute. For example, the importance value of the age attribute is the sum of different age groups. We can find that the importance values of age and consumption level are higher than other user attributes for all four cities, which is further evidence that there is chain enterprise bias on consumer groups in different cities. However, we also observed that there are certain differences in terms of the feature importance of POI attributes among different cities. For example, the most important POI categories in Beijing are company and hotel, which are different from Chengdu. One possible reason is that *different cities have different urban geographical structures, which could affect consumer behavior in brick-and-mortar stores to some extent. Therefore, it is necessary to consider the diversity of different cites when transferring knowledge.*

3.3.2 Feature Distribution between Different Cities. Generally, most cities have many differences in POI distributions, road networks, and so on, which could lead to feature distribution differences among multiple cities. Furthermore, the feature shift between two cities poses a major bottleneck for transferring knowledge and adapting the prediction model from source city to target city.

To quantitatively measure the discrepancy of feature distributions in different cities, we adopt MMD [12] to compute the distance between two feature distributions. MMD is a nonparametric statistic that measures the distribution difference in terms of the distance between the mean embedding representations of the source and target data in the reproducing kernel Hilbert space $\mathcal{H}$. Formally, given feature distributions p_{c_s} and q_{c_t} in two cities, respectively, the MMD distance between source city c_s and target city c_t is defined as:

$$MMD(p_{c_s}, q_{c_t}) \triangleq \|E_{p_{c_s}}[\phi(\mathbf{x}_{c_s})] - E_{q_{c_t}}[\phi(\mathbf{x}_{c_t})]\|^2_{\mathcal{H}}, \tag{4}$$

Table 4. MMD Distance for Different Cities

City 1	City 2	MMD-Linear	MMD-Gaussian
Beijing	Shanghai	15.696	0.165
Beijing	Chengdu	41.001	0.339
Shanghai	Chengdu	33.293	0.326
Beijing	Xi'an	42.712	0.407
Chengdu	Xi'an	12.514	0.154

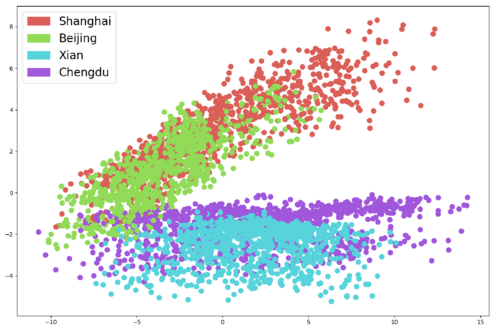


Fig. 1. T-SNE of features on different cites.

where $\phi(.)$ is the kernel function that maps the raw features into $\mathcal{H}$, $E_{p_{c_s}}[\phi(\mathbf{x}_{c_s})]$ and $E_{q_{c_t}}[\phi(\mathbf{x}_{c_t})]$ are the kernel mean embeddings of c_s and c_t, respectively.

Table 4 lists the MMD distance of feature distributions in four cities under the linear kernel and Gaussian kernel settings respectively. We can find that distance between city pairs (Beijing, Chengdu) and (Beijing, Xi'an) are significantly larger than the distance between city pairs (Beijing, Shanghai) and (Chengdu, Xi'an), which indicates the obvious feature discrepancy between tier-1 cities (Beijing, Shanghai) and two tier-2 cities (Chengdu, Xi'an). The reason is that tier-1 cities have more diversified users and richer POI structures than tier-2 cities. In addition, the MMD distance between Beijing and Shanghai is larger than the threshold for rejecting the hypothesis although they are both tier-1 cities.

To further show the distribution difference among four cities, we visualize the feature distribution in Figure 1 using t-SNE embeddings [6]. Similarly, we can observe that feature distributions on tier-1 cities and tier-2 cities have obvious difference. *In general, all the above observations indicate the feature distribution shift among different cities. Therefore, we need to reduce the feature distribution discrepancy between different cities so that the knowledge learned from source city can be effectively transferred to target city to improve its performance.*

3.4 The WANT Framework

In this article, we focus on the cold-start store site recommendation. We assume to have source city dataset $\mathcal{D}_s = \{(\mathbf{x}_k^s, y_k^s)\}_{k=1}^{n_s}$ consisting of n_s labeled examples and target city dataset $\mathcal{D}_t = \{\mathbf{x}_k^t\}_{k=1}^{n_t}$ with n_t unlabeled examples. From the data analysis in the last section, we observe that the source city and target city follow different distributions p and q, respectively. The goal of our proposed method is to build a prediction model, which can minimize the target city error, by transferring knowledge learned from the source city to help a target city.

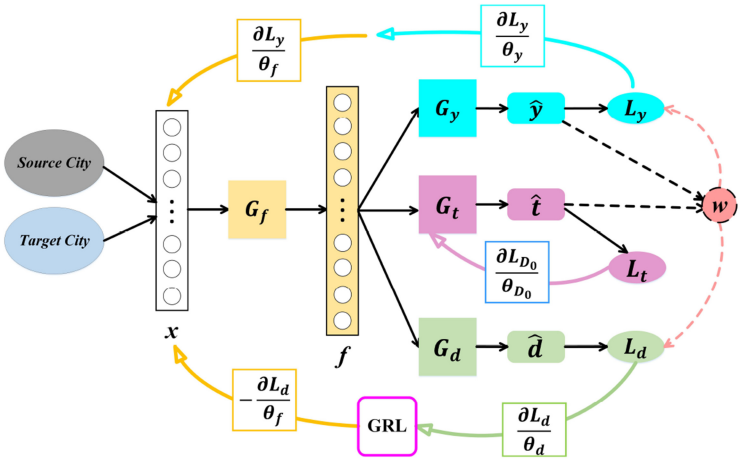


Fig. 2. The framework of WANT.

Nevertheless, the domain shift between source and target cities poses two key challenges to enable effective knowledge transfer: *(1) How to bridge different cities with different data distributions?* Intuitively, naively transferring knowledge from the source city to the target city could hurt the performance in the target city due to data distribution discrepancy. Therefore, we need to bridge different cities by reducing the data distribution discrepancy to promote positive transfer between two cities. One possible way to enforce distribution consistency is learning the feature representations that are invariant to the domain shift, which could reduce negative transfer at the feature level. *(2) How to reduce negative transfer of useless source samples?* Existing domain adaptation methods mainly align the data distribution in the entire feature space to reduce distribution discrepancy between two domains, which will further trigger the negative effects of untransferable source samples on domain alignment. Generally, urban data could suffer from data noisy issues in many practical applications. If we directly align the entire data spaces in source city and target city, noisy data in source city will result in negative transfer. In addition, forcefully matching all examples in source city and target city if some source examples are significant irrelevant will also lead to weak transferability. Therefore, it is necessary to automatically choose useful examples in source city that are transferable to target city, such that the distribution alignment based on transferable examples can be done to mitigate the negative effects of untransferable source samples at the instance level.

To address these two challenges, we propose WANT, to transfer the knowledge learned from a data-rich source city to a target city with no labeled data. The framework of WANT is illustrated in Figure 2, which mainly consists of four major components: feature extractor, domain discriminator, label predictor, and transferability weighting quantification.

Feature extractor G_f learns deep feature interactions $\boldsymbol{f}$ based on raw extracted features $\boldsymbol{x}$, which is used for both source city and target city.

Domain discriminator G_d aims to align the feature distributions of the source data and target data to bridge different cities to solve the first challenge. Specifically, it distinguishes the source data from the target data, and plays the minimax game with the feature extractor G_f to guide it to learn transferable and domain-invariant feature representations by adversarial learning to promote positive transfer. Note that we utilize the Gradient Reversal Layer (GRL) to reverse the gradient between the feature extractor G_f and adversarial domain discriminator G_d in backward propagation for domain adversarial training.

Label predictor G_y is trained based on the transferable and domain-invariant feature representations $\mathbf{f}$ of labeled source samples, which could be applied to predict the label of target samples, i.e., the amount of consumption per community in each store in the target city.

Transferability weighting quantification computes the transferability weight w of the source sample, and further highlights the contribution of transferable source examples to tackle the second challenge. Specifically, we reweigh the source examples in the loss of adversarial domain discriminator L_d and source predictor L_y for transferring useful source examples to prevent negative transfer. The dotted lines in Figure 2 indicate the process of transferability weighting scheme w, which is based on both the non-adversarial domain discriminator G_t to obtain the similarity of the source example to target data and the label predictor G_y to obtain the quality of the source example. It should be noted that non-adversarial domain discriminator G_t is only applied for computing the similarity of source example to target data, thus, the gradient of G_t will not be back-propagated to update the feature extractor during the training procedure.

4 METHODOLOGY

In this section, we describe the details of the WANT to solve the cold-start problem in chain store site recommendation. WANT consists of four major components: feature extractor, domain discriminator, label predictor, and transferability weighting quantification. We first describe each of the four components, and then present the minimax optimization problem with joint optimization.

4.1 Feature Extractor

Generally, consumer behavior is usually affected by various complicated factors simultaneously. Therefore, besides some valuable features extracted from multi-source data, as presented in Section 3.2, we further adopt a feature extractor to learn latent factors and deep feature interactions based on the raw extracted features.

Formally, the feature extractor G_f is a feed-forward neural network including N-layer fully-connected layers for mapping the input from source and target cities into a common feature space. Given the raw feature vectors $\mathbf{x}$ as the input, feature extractor is defined as follows:

$$\begin{aligned} \mathbf{f}_1 &= \sigma(\mathbf{W}_{f1}\mathbf{x} + \mathbf{b}_{f1}), \\ \mathbf{f} &= \sigma(\mathbf{W}_{fN}\mathbf{f}_{N-1} + \mathbf{b}_{fN}), \end{aligned} \tag{5}$$

where $\mathbf{x} = [\mathbf{x}_g, \mathbf{x}_c, \mathbf{x}_t, \mathbf{x}_u]$ is the input vector containing geographic features $\mathbf{x}_g$, commercial features $\mathbf{x}_c$, time features $\mathbf{x}_t$, and user features $\mathbf{x}_u$. $\mathbf{W}_f$ denotes the weight matrices and $\mathbf{b}_f$ represents the bias vectors. σ is the *ReLU* function. The output of feature extractor is the final deep feature representation $\mathbf{f} = G_f(\mathbf{x})$, which will be applied in label predictor and domain discriminator.

It is noteworthy to specify that a feature extractor is used for both source city and target city, because we find it is beneficial to consider shared feature space between two cities. In addition, the feature extractor is capable of learning transferable and invariant feature representations across cities by adversarial learning, and the details will be introduced in the next section.

4.2 Domain Discriminator

To transfer knowledge across cities with different data distributions, it is necessary to bridge the gap between the source city and the target city in the presence of domain shift by reducing the distribution discrepancy. Therefore, it is essential to measure the difference between the source and target domains in the probability distribution space, then the transferable feature representations across two cities could be learned to minimize that distance. Inspired by DANN [9], we adopt a domain discriminator G_d to distinguish the source from the target.

Intuitively, if the feature representation is transferable across cities, it should be invariant so that a classifier cannot discriminate which city the sample is from. Formally, we model the domain discriminator as a binary domain classifier to predict the domain labels, where label 1 represents the example belong to the source city, and label 0 from the target city. Given the input $\mathbf{x}$ from source data or target data, the domain discriminator G_d takes the output of feature extractor $\mathbf{f}$ as input, and then calculates the probability $\hat{d} = G_d(\mathbf{f})$ that $\mathbf{x}$ comes from the source data by the following process:

$$\begin{aligned} \mathbf{d}_1 &= \sigma(\mathbf{W}_{d1}\mathbf{f} + \mathbf{b}_{d1}), \\ \mathbf{d}_2 &= \sigma(\mathbf{W}_{d2}\mathbf{d}_1 + \mathbf{b}_{d2}), \\ \hat{d} &= sigmoid\left(\mathbf{h}_d^T \mathbf{d}_2\right), \end{aligned} \tag{6}$$

where the $\mathbf{W}_d$ and $\mathbf{b}_d$ are weight and bias terms of domain discriminator, respectively, and $\mathbf{h}_d$ represents the weights of the output layer.

Given the domain label d_i of instance $\mathbf{x}_i$ from source city or target city, the objective function of domain discriminator G_d is formulated as:

$$\begin{aligned} E_{G_d} &= \frac{1}{n_s + n_t} \sum_{\mathbf{x}_i \in \mathcal{D}_s \cup \mathcal{D}_t} L_d(G_d(G_f(\mathbf{x}_i)), d_i) \\ &= -\frac{1}{n_s} \sum_{\mathbf{x}_i \in \mathcal{D}_s} \log(G_d(G_f(\mathbf{x}_i))) - \frac{1}{n_t} \sum_{\mathbf{x}_i \in \mathcal{D}_t} \log(1 - G_d(G_f(\mathbf{x}_i))). \end{aligned} \tag{7}$$

However, the goal of the general training process for the classifier is to minimize the classification error, i.e., to distinguish the two domains as accurately as possible. Nevertheless, the objective of our approach is to reduce the effect of distribution discrepancy and learn invariant features which the domain classifier cannot discriminate between domains, it means that a classifier should have very low accuracy.

To solve this problem, we take advantage of adversarial learning. Specifically, to align the distribution of feature representations from different cities, we apply adversarial learning to force our model to learn transferrable representations that can confuse the discriminator trained to distinguish which domain a representation vector is from.

More formally, a two-player minimax game is constructed, in which the first player is the feature extractor G_f, and the second player is the domain discriminator G_d. One the one hand, the domain discriminator is trained to distinguish the source from the target by minimizing the classification loss E_{G_d}. On the other hand, the feature extractor aims to learn domain-invariant representations to confuse the domain discriminator, i.e., parameters of the feature extractor are learned simultaneously by maximizing the loss E_{G_d} of domain discriminator.

$$\max_{G_f} \min_{G_d} E_{G_d}. \tag{8}$$

4.3 Label Predictor

After obtaining transferable feature representations, we then use them to predict the amount of consumption of consumers in the store. Therefore, we adopt a label predictor G_y to produce predicted value $\hat{y}$, and the layers can be formulated as:

$$\begin{aligned} \mathbf{y}_1 &= \sigma(\mathbf{W}_{y1}\mathbf{f} + \mathbf{b}_{y1}), \\ \mathbf{y}_2 &= \sigma(\mathbf{W}_{y2}\mathbf{y}_1 + \mathbf{b}_{y2}), \\ \hat{y} &= \mathbf{h}_y^T \mathbf{y}_2, \end{aligned} \tag{9}$$

where $\mathbf{f}$ is deep feature representation from the feature extractor G_f, the $\mathbf{W}_y$ and $\mathbf{b}_y$ are weight and bias of the label predictor, respectively, σ is the non-linear activation function for which we use *ReLU*, and $\mathbf{h}_y$ denotes the neuron weights in the output layer.

Given the labeled data from source city, label predictor G_y is learned by minimizing the following objective function:

$$E_{G_y} = \frac{1}{n_s} \sum_{\mathbf{x}_i, y_i \in \mathcal{D}_s} L_y(G_y(G_f(\mathbf{x}_i)), y_i), \tag{10}$$

where L_y is the loss function of regression problem. The label predictor G_y trained only using examples in source city is capable of predicting the consumer behavior in target city, because the feature representations from feature extractor are transferable and invariant across cities.

4.4 Transferability Weighting Quantification

Intuitively, not all examples in source city are equally transferable, and some examples could be more transferable than others. On the one hand, some source examples that are significantly dissimilar to target data will lead to weak transferability. On the other hand, noisy data in source city will also result in negative transfer. Therefore, we design the transferability weighting mechanism to quantify the transferability of each example in source city based on the similarity of source example to target data and the quality of source example.

4.4.1 The Similarity of Source Example. It is obvious that examples in source city which are similar to the target city should be more transferable. Intuitively, we could adopt the domain discriminator to generate the similarity value for each source example, and the examples that are more similar to target data have larger values. However, the adversarial domain discriminator presented in the previous section aims to match source examples and target examples, if we apply the output of adversarial domain discriminator as weights to select transferable examples, the theoretical results of the minimax game will not be reducing the distribution divergence [45].

Therefore, we adopt an additional non-adversarial domain discriminator G_t consisting of two fully connected feed-forward layers, and use the activations of the non-adversarial domain discriminator as an indicator of the similarity, which is defined as:

$$\hat{t} = G_t(G_f(\mathbf{x}_i)), \tag{11}$$

where $\hat{t}$ is the output of non-adversarial domain discriminator that indicates the possibility of instance $\mathbf{x}_i$ belonging to the source city, and smaller $\hat{t}$ means that it is more similar to the target data. Thus, we obtain the similarity value of each example $s(\mathbf{x}_i)$ that denotes the probability of classifying the instance $\mathbf{x}_i$ coming from the target city:

$$s(\mathbf{x}_i) = 1 - \hat{t} = 1 - G_t(G_f(\mathbf{x}_i)). \tag{12}$$

It is noteworthy to specify that the gradient of non-adversarial domain discriminator G_t will not be back-propagated to update the feature extractor G_f since G_f is not learned to confuse G_t, and G_t is only trained to distinguish the source and target examples by minimizing the following objective function:

$$E_{G_t} = \frac{1}{n_s + n_t} \sum_{\mathbf{x}_i \in \mathcal{D}_s \cup \mathcal{D}_t} L_t(G_t(G_f(\mathbf{x}_i)), d_i), \tag{13}$$

where d_i is the domain label of instance $\mathbf{x}_i$, and L_t is the domain classification loss (i.e., cross-entropy loss).

4.4.2 The Quality of Source Example. To improve the robustness of the model, we also expect transferable examples in source city are high-quality. However, urban data could suffer from data noisy issues in many practical applications. If we forcefully match all source examples with target examples, noisy data in source city will result in negative transfer.

Inspired by curriculum learning [1], which organizes examples by a better arrangement to promote stronger optimization, we aim to select high-quality source data and transfer them to reduce negative transfer. Our intuition is that smaller loss means more confident prediction, thus, we construct the self-paced curriculum by assigning higher weights to easier examples that have smaller loss, and selecting them into training iteratively [15]. Specifically, the loss $l(\mathbf{x}_i)$ between the predicted value and label can be view as the criterion to choose easy examples, and we select high-quality examples based on predefined curriculum:

$$l(\mathbf{x}_i) = L_y(G_y(G_f(\mathbf{x}_i)), y_i), \tag{14}$$

$$q(\mathbf{x}_i) = \mathbb{I}(l(\mathbf{x}_i) \leq \gamma) \tag{15}$$

where $\mathbb{I}$ is the indicator function, the hyper-parameter γ controls the learning pace, and $q(\mathbf{x}_i) \in [0, 1]$ is a weight to quantify whether $\mathbf{x}_i$ is a high-quality instance.

4.4.3 Transferability of Source Example. After getting the similarity of source example to target data and the quality of source example, we compute the transferable weights of each example in source city to target city, which is represented as:

$$w(\mathbf{x}_i) = s(\mathbf{x}_i)q(\mathbf{x}_i), \qquad q(\mathbf{x}_i) \in [0, 1]. \tag{16}$$

In general, the low-quality examples in source city are filtered out first, and then the high-quality examples whose representations are more similar to target data will be weighted by larger weight values. In this way, the transferability weighting mechanism is capable of selecting transferable examples from high-quality source data. Furthermore, we could highlight their contributions to the transfer model by reweighing the source examples to improve the process of distribution alignment to prevent negative transfer.

4.5 Jointly Optimization

As explored in the last section, to enhance positive transfer and reduce negative transfer, we develop a transferability weighting mechanism to quantify the transferability of source examples. Therefore, we construct new objectives for adversarial domain discriminator G_d and label predictor G_y based on the transferable weight of each source example $w(\mathbf{x}_i)$. Our intuition is that a source example with a larger weight should contribute more to the transfer model to promote positive transfer.

For adversarial domain discriminator G_d, we reweigh the source examples in the loss, which means that we focus on aligning the feature distributions of transferable source examples and target examples to promote positive transfer. In our work, we aim to obtain the relative transferability of source samples, thus, we normalize the weight of each source example $w(\mathbf{x}_i)$ in each mini-batch of batch size B as: $w(\mathbf{x}_i) = \frac{w(\mathbf{x}_i)}{\frac{1}{B}\sum_{j=1}^{B} w(\mathbf{x}_j)}$.

$$E_{G_d} = -\frac{1}{n_s} \sum_{\mathbf{x}_i \in \mathcal{D}_s} w(\mathbf{x}_i) \log(G_d(G_f(\mathbf{x}_i))) - \frac{1}{n_t} \sum_{\mathbf{x}_i \in \mathcal{D}_t} \log(1 - G_d(G_f(\mathbf{x}_i))). \tag{17}$$

In addition, we also reweigh the source loss of label predictor G_y, which significantly reduces the risk of negative transfer by diminishing the contribution of irrelevant source examples.

$$E_{G_y} = \frac{1}{n_s} \sum_{\mathbf{x}_i, y_i \in \mathcal{D}_s} w(\mathbf{x}_i) L_y(G_y(G_f(\mathbf{x}_i)), y_i). \tag{18}$$

With the weighted loss of adversarial domain discriminator and label predictor based on the transferability of source examples, we now present the final optimization procedure in detail. Our proposed model, WANT, jointly optimizes over label classification error E_{G_y}, domain classification error E_{G_t}, and domain adversarial error E_{G_d}. Thus, the total objective function can be written as follows:

$$E(\theta_f, \theta_y, \theta_d, \theta_t) = E_{G_y} + \alpha E_{G_t} - \beta E_{G_d}, \tag{19}$$

where $\theta_f, \theta_y, \theta_d, \theta_t$ are parameters of G_f, G_y, G_d, G_t, respectively, and the hyper-parameters α and β control the trade-off between the objectives of non-adversarial domain discriminator and adversarial domain discriminator in the unified optimization problem.

The objective of the minimax optimization problem is to find the network parameters $\hat{\theta}_f, \hat{\theta}_y, \hat{\theta}_d$ and $\hat{\theta}_t$ by the following operations:

$$\begin{aligned} (\hat{\theta}_f, \hat{\theta}_y) &= \arg\min_{\theta_f, \theta_y} E_{G_y} - \beta E_{G_d}, \\ \hat{\theta}_d &= \arg\max_{\theta_d} E_{G_y} - \beta E_{G_d}, \\ \hat{\theta}_t &= \arg\min_{\theta_t} \alpha E_{G_t}. \end{aligned} \tag{20}$$

To be noted, adversarial domain discriminator G_d plays the minimax game with the feature extractor G_f for updating G_f, but non-adversarial domain discriminator G_t is only used for obtaining the similarity of source examples to target city. In summary, the feature extractor is trained to minimize the label classification loss and maximize domain adversarial loss simultaneously, thus, the WANT is the capability to simultaneously learn transferable and discriminative features. In addition, WANT reduces the risk of negative transfer by learning to transfer useful examples in source city based on a transferability weighting mechanism.

In this work, we utilize the GRL [9] to reverse the gradient between the feature extractor and adversarial domain discriminator in order to jointly learn all parameters in an end-to-end framework, as shown in Figure 2. The proposed optimization procedure is summarized as a pseudocode in Algorithm 1.

5 EXPERIMENTS

In this section, we systematically evaluate our proposed model on real-world datasets. After giving the detailed experimental settings, we investigate the effectiveness of our approach and compare it with several state-of-the-art methods for cold-start store site recommendation. Finally, we discuss the limitations of our work and future work.

5.1 Experimental Settings

5.1.1 Data Description. The datasets we used to evaluate the performance of our model and baselines are real-word datasets from a chain retail enterprise, which owns a lot of brick-and-mortar stores in multiple cities of China. In our experiments, we obtain 6 datasets that contain over 100 stores with more than 50,000 communities from 6 different cities in China, including 3 tier-1 cities (i.e., Beijing, Shanghai, and Guangzhou) and 3 tier-2 cities (i.e., Chengdu, Hangzhou, and Xi'an), respectively. For each city, we collect three types of data, including store data, user data, and POI data. Store data contain the basic information of different stores (e.g., name and location), as well as the set of consumption records of each user in each store (e.g., the amount of consumption). User data provide basic user information in each community, including profile information (e.g., gender, age, profession, and income level), and location information (i.e., the location of the community where the user lives). POI data include the information (e.g., name,

ALGORITHM 1: Learning Algorithm for WANT

Input: Labeled dataset in source city $\mathcal{D}_s = \{(\mathbf{x}_i^s, y_i^s)\}_{i=1}^{n_s}$, unlabeled dataset in target city $\mathcal{D}_t = \{\mathbf{x}_j^t\}_{j=1}^{n_t}$.

Output: Learned parameters $(\hat{\theta}_f, \hat{\theta}_y, \hat{\theta}_d, \hat{\theta}_t)$.

Randomly initialize $\theta_f, \theta_y, \theta_d, \theta_t$;

while *not done* **do**

Sample k labeled examples $\{(\mathbf{x}_i^s, y_i^s)\}_{i=1}^k$ and k unlabeled examples $\{\mathbf{x}_j^t\}_{j=1}^k$ uniformly from $\mathcal{D}_s$ and $\mathcal{D}_t$, respectively;

Compute transferable weights for k source examples: $w(\mathbf{x}_i^s)$ in Equation (16);

Update θ_d by minimizing the following objective E_{G_d}:

$$E_{G_d} = -\frac{1}{k}\sum_{i=1}^k w(\mathbf{x}_i^s)\log(G_d(G_f(\mathbf{x}_i^s))) - \frac{1}{k}\sum_{j=1}^k \log(1 - G_d(G_f(\mathbf{x}_j^t)));$$

Update θ_y by minimizing the following objective E_{G_y}:

$$E_{G_y} = \frac{1}{k}\sum_{i=1}^k w(\mathbf{x}_i^s)((G_y(G_f(\mathbf{x}_i^s)) - y_i^s))^2;$$

Update θ_f by simultaneously minimizing the objective E_{G_y} and maximizing the objective E_{G_d} based on the reversed gradient between G_d and G_f;

Update θ_t by minimizing the following objective:

$$E_{G_t} = -\frac{1}{k}\sum_{i=1}^k \log(G_d(G_f(\mathbf{x}_i^s))) - \frac{1}{k}\sum_{j=1}^k \log(1 - G_d(G_f(\mathbf{x}_j^t)));$$

end

location, and category) of 12 different categories of POI related to the retail business, such as shop, food, transport, company, and education.

5.1.2 Evaluation Metrics. In our experiments, we adopt Mean Square Error (MSE) and Mean Absolute Error (MAE) defined as follows as the evaluation metrics.

—**MSE.** Our model predicts the amount of consumption per community in each store located at the candidate places in target city. Therefore, we adopt MSE for result comparison, as shown in follows:

$$MSE = \frac{1}{n}\sum_{i=1}^n (\hat{y}_i - y_i)^2, \tag{21}$$

where n is the number of instances in the target city, and $\hat{y}_i$ and y_i are predicted result and ground truth, respectively.

—**MAE.** The objective of the store site recommendation is to assess the value of the store at candidate locations and then select the optimal one from a set of candidate locations for the company to place a new brick-and-mortar store. In this way, we use MAE to evaluate the performance of our model for each store s_j, which will be placed at the candidate location l_j,

$$MAE = \frac{1}{m}\sum_{j=1}^m \left|\frac{\hat{\mathcal{S}}_j - \mathcal{S}_j}{\mathcal{S}_j}\right|, \tag{22}$$

where m is the number of candidate locations to place the new store in target city c_t, and $\mathcal{S}_j$ is the actual sale of store s_j. It should be noted that $\hat{\mathcal{S}}_j$ is the predicted sale of store s_j, which is represented as the total amount of consumption of all potential consumers in this store; $\hat{\mathcal{S}}_j = \sum_{m_i \in \mathcal{M}_j} \hat{y}_{m_i,l_j}$, and $\hat{y}_{m_i,l_j}$ is the amount of consumption per community m_i in store s_j located at l_j. Note that we identify the set of nearby communities $\mathcal{M}_j$ around store s_j as most of the potential consumers (i.e., lying in a disk of radius r around the store), instead

of considering all people in target city as potential consumers on account of users' mobility patterns in the physical world.

5.1.3 Compared Methods. We compare our approach with several state-of-the-art transfer learning methods. Note that the structures of baseline methods are the same as that of our proposed model. We also note that we don't consider other transfer models for store site recommendation in our experiment, such as Citytransfer [13], because it cannot recommend the optimal location from any given candidate locations (e.g., the street or the shopping mall) for a new brick-and-mortar store to solve our proposed problem.

- **DNN(Source-only).** This method simply uses labeled data in source city to train the prediction model for target city without any adaptation.
- **DANN.** DANN [9] is a representative method based on adversarial learning for domain adaptation, which includes a feature extractor, a label predictor, and a domain classifier to learn discriminative and domain-invariant features for cross-domain transfer.
- **ADDA.** ADDA [37] is an asymmetric domain adaptation framework. It first pre-trains a source feature extractor using labeled source examples, and then performs adversarial adaptation to learn a separate feature extractor in target domain that maps the target examples to the same space by a domain discriminator.
- **IWAN.** IWAN [45] extends ADDA by considering an additional domain classifier to detect the source samples that are potentially from the outlier classes and identify the importance weights of source samples.
- **TCL.** TCL [33] transfers relevant and clean source data by learning a transferable curriculum to address weakly-supervised domain adaptation, which consists of a feature extractor, a domain discriminator and a label classifier.

5.1.4 Implementation Details. In this work, we aim to predict the amount of consumption per community in the store, and then obtain the total amount of consumption of all potential consumers in each store for recommending the optimal location to the company to place a new brick-and-mortar store. Based on the empirical knowledge and previous studies [18], we choose people 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.

The models including the proposed WANT and baseline methods in our experiments are implemented with Tensorflow. In detail, the structure of WANT contains four components: feature extractor consists of 2 fully connected layers with 256 nodes each; two layers are used for the label predictor layers, and the dimension of the hidden layer is set as 256; adversarial domain discriminator consists of 2 fully connected layers with 128 nodes for the hidden layer, which plays the minimax game with the feature extractor to learn transferable and domain-invariant feature representations. Similarly, we employ the same architecture with the hidden dimension of 128 as the non-adversarial domain discriminator, to identify the similarity of source examples for target data.

For training, we apply mini-batch stochastic gradient descent (SGD) with momentum of 0.9 to update all the parameters, and the learning rate is adjusted during SGD using the following schedule implemented in DANN [9]: $lr = \frac{\lambda}{(1+\alpha p)^{\beta}}$, where p is the training progress linearly changing from 0 to 1, α and β are set as 10 and 0.75, respectively, and λ is the upper bound of learning rate set as 0.0001 in our experiments. The batch size and the maximum number of epochs are set as 32 and 10,000, respectively. In addition, to avoid over-fitting, we adopt the dropout strategy with dropout rate set as 0.5. It should be noted that under the setting of the cold-start recommendation problem, no labeled data are available for the target city. Therefore, we only use the labeled

Table 5. Performance Comparison of Different Models

Transfer tasks	Beijing→Xi'an		Chengdu→Xi'an		Guangzhou→Xi'an		Shanghai→Hangzhou	
Metrics	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
DNN(Source-only)	145.211	0.4634	206.263	1.3185	206.260	1.3185	356.353	1.1748
DANN	128.518	0.7015	68.919	0.1662	94.658	0.4764	202.285	0.4401
ADDA	184.243	0.3437	148.881	0.4754	189.305	0.2252	259.866	0.4785
IWAN	151.895	**0.2764**	137.520	0.2484	177.059	0.2372	296.701	0.7270
TCL	92.096	0.6522	67.812	0.1595	94.243	0.5391	248.408	0.5516
WANT	**79.463**	0.3451	**66.708**	**0.1473**	**82.428**	**0.2250**	**146.376**	**0.4394**

training data from source city and unlabeled training data from target city to train the prediction model.

5.2 Experimental Results

We conduct the following experiments on the cold-start scenario, and present the experimental results to evaluate our proposed method.

5.2.1 Performance Comparison of Different Models. In this experiment, we consider four unsupervised transfer tasks, including Beijing→Xi'an, Chengdu→Xi'an, Guangzhou→Xi'an, and Shanghai→Hangzhou. The comparison results with various baselines are shown in Table 5. It should be noted that MSE is the major metric to evaluate the performance of the prediction model, and MAE is the additional result of computation based on predicted value. We find that the performances of all methods are consistent over two metrics in most cases, but MSE and MAE results sometimes conflict. There are two possible reasons. First, we choose people 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 to error to some extent, for example, given the MSE value, MAE calculated by adding up predicted values could be large because of error accumulation, or small because of error elimination. Therefore, we mainly consider MSE values for results comparison.

According to these results, we draw the following observations:

- —DANN significantly outperforms DNN(Source-only) in terms of MSE on all transfer tasks, because DANN considers the difference of data distribution between source city and target city, and adopts a domain discriminator to facilitate the prediction model to learn invariant features. The results verify that feature adaptation is effective for knowledge transfer between two cities with different data distributions. In addition, DANN is a simplified model of our proposed WANT without the transferability weighting mechanism, which also indicates that adversarial learning in our framework is effective to learn the domain-invariant feature representation for knowledge transfer between two cities with different data distributions.
- —We observe that ADDA and IWAN perform worse than DANN. One possible reason is that ADDA and IWAN first pre-train a feature extractor in source city, and then learn a separate feature extractor in target city, so that deep features in source city and target city cannot be matched adequately, and label predictor trained based on some irrelevant source examples may hurt the performance.
- —WANT achieves more superior results than DANN, especially on tasks Beijing→Xi'an and Shanghai→Hangzhou. It indicates that general transfer learning methods are prone to

Table 6. Performance Comparison of WANT and Its Variants

Transfer tasks	Beijing→Xi'an		Chengdu→Xi'an		Guangzhou→Xi'an		Shanghai→Hangzhou	
Metrics	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
WANT w/o quality	145.058	0.7839	**64.847**	0.2088	96.861	0.2861	223.850	0.3717
WANT w/o predictor	108.275	0.5381	68.790	0.2271	88.672	0.2274	223.523	**0.3584**
WANT	**79.463**	**0.3451**	66.708	**0.1473**	**82.428**	**0.2250**	**146.376**	0.4394

negative transfer if data distributions between source city and target city are significantly different, because they ignore some source examples irrelevant to target city that could lead to negative transfer. Thus, it is not an effective way to transfer all examples in source city.

- We find that WANT achieves better performance than DANN and TCL on most transfer tasks, which proves the effectiveness of transferable weighting mechanism. However, DANN and TCL achieve competitive performance compared with WANT in Chengdu→Xi'an task. One possible reason is that most examples in source city are useful and relevant to target city, thus directly matching all source examples to target examples could also obtain good results.
- As we can see, our proposed WANT is superior to all the state-of-the-art methods on most tasks, showing its power to transfer useful source examples to target city. Specifically, compared with the non-transfer learning method (i.e., DNN(Source-only)), WANT achieves 45% improvements in terms of MSE on transfer task Beijing→Xi'an, similar improvements can be found in the other three transfer task. The results demonstrate the advantages of WANT in learning transferable weights and filtering outlier noisy data from source city, which diminishes the negative impact of irrelevant and noisy source examples and promotes positive transfer.
- It can be observed that the performances of knowledge transfer from different source cities are a little different. Specifically, for the target city Xi'an, WANT achieves the best performance in Chengdu→Xi'an task, followed by Beijing→Xi'an task, and finally Guangzhou→Xi'an task. This is intuitive because Chengdu and Xi'an are both tier-2 cities, which could have similar feature distributions. This implicitly indicates that choosing an appropriate source city can improve the performance to some extent.

5.2.2 Impact of the Transferability Weighting Mechanism. To reduce the risk of negative transfer, we propose a transferability weighting scheme to quantify the transferability of examples in source city, and highlight their contributions to knowledge transfer. In order to evaluate the effectiveness of our proposed transferability weighting mechanism, we investigate two variants of WANT:

(1) **WANT w/o quality** is the variant by removing the quality of source example term $q(\mathbf{x}_i)$ from the transferable weights $w(\mathbf{x}_i)$ in Equation (16) on the domain discriminator and label predictor.
(2) **WANT w/o predictor** is the variant by removing the transferable weights $w(\mathbf{x}_i)$ for the source examples on the label predictor.
(3) **WANT** considers the transferable weights for the source examples on both the domain discriminator and label predictor.

The experimental results of different variants of WANT are shown in Table 6, we can observe that:

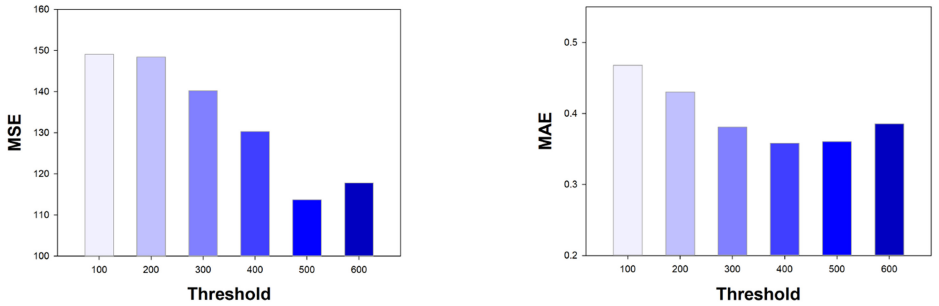


Fig. 3. Results with respect to different thresholds.

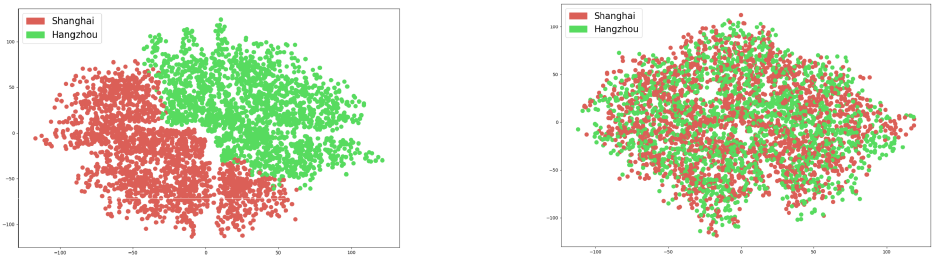
—WANT outperforms WANT w/o quality, which indicates that noisy data in source city could lead to negative transfer. Thus, it is necessary to consider the quality of source example in the weighting mechanism to filter out low-quality examples, so that the noisy data in source city have little influence on knowledge transfer to the target city.
—WANT outperforms WANT w/o predictor, which demonstrates that the weighting mechanism on the label predictor can reduce the negative influence of irrelevant source examples and focus on the transferable examples in source city.
—In general, WANT can successfully select high-quality and relevant source examples, and highlight their contributions to knowledge transfer to promote positive transfer and circumvent negative transfer.

5.2.3 Hyper-parameter Investigation. To reduce negative transfer in view of noisy data in practical applications, we construct the self-paced curriculum to transfer high-quality source data, where a threshold γ is defined to control the learning pace, and the source example that its loss is smaller than γ will be selected into the curriculum learning procedure. In this experiment, we evaluate how different selections of the hyper-parameters γ in the transferability weighting mechanism impact our proposed model's performance.

Figure 3 shows the performance of WANT for varying different thresholds γ on transfer task Shanghai→Hangzhou. We find that the performance improves with the decreasing thresholds at first, because smaller threshold could filter out more noisy data, which proves that the transferability weighting mechanism is capable of selecting high-quality examples in source city. However, the performance drops later with the decrease of thresholds. One possible reason is that the total number of training examples decreases when we decrease γ, and there is no enough training data for source city to train the model.

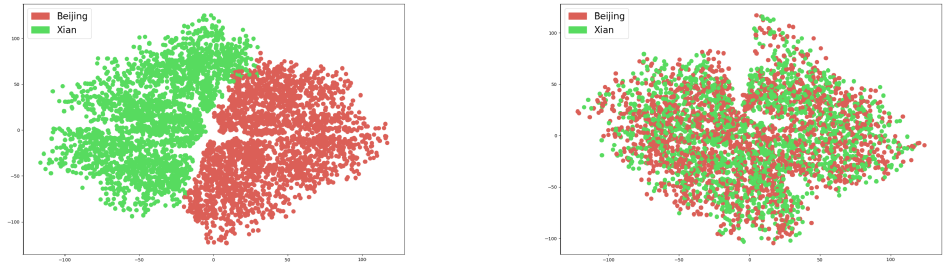
5.2.4 Feature Visualization. To transfer knowledge between different cities, we present a domain discriminator to align the feature distributions of source city and target city, which further guides the prediction model to learn transferable feature representations. In order to understand the transferable representation learned by WANT, we visualize the feature representation generated by feature extractor in two transfer tasks, including Shanghai→Hangzhou and Beijing→Xi'an.

In practice, we randomly sample some samples from source city and target city, respectively, and then use t-SNE [6] to reduce the dimensionality of feature vectors to 2. In this experiment, we plot the feature representations learned by DNN and WANT, and the results are shown in Figures 4 and 5. Note that features learned by DNN represent the features without adaptation, and features leaned by WANT represent the features after adaptation.



(a) No adapted (b) Adapted

Fig. 4. The T-SNE visualization of Shanghai→Hangzhou adaptation.



(a) No adapted (b) Adapted

Fig. 5. The T-SNE visualization of Beijing→Xi'an adaptation.

Table 7. Performance Comparison of DNN and WANT in Terms of MSE

Transfer tasks	Shanghai→Hangzhou	Beijing→Xi'an
No adapted (DNN)	356.353	145.211
Adapted (WANT)	**146.376**	**79.463**

As shown in Figures 4 and 5, we can observe that the distributions of source and target features learned by WANT with adaptation are more closer and indistinguishable compared to features without adaptation, which proves that WANT is capable of aligning the feature space and learning transferable features to reduce the feature distribution discrepancy between source city and target city. Especially, the results in Table 7 intuitively show that the improvement of WANT in Beijing→Xi'an task is 45%, and the improvement in Shanghai→Hangzhou is over 50% compared with DNN in terms of MSE. In general, the results indicate that WANT can learn both disentangled and transferable feature representations for knowledge transfer.

5.3 Discussion

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

—**Multi-city Knowledge Transfer.** WANT focuses on transferring knowledge from a single source city to the target city, and the experiments indicate that the performances of

knowledge transfer from different source cities vary a little because of distinct city characteristics. Intuitively, the knowledge learned from multiple cities could be comprehensive and complementary. Therefore, we plan to transfer knowledge learned from multiple source cities to improve the performance of knowledge transfer and increase the stability of transfer.

— **Time Series Modeling.** The current work aims to predict consumer behavior in a short period of time for store placement. Future studies should explore the dynamic consumer behavior based on the sequence model (e.g., Recurrent Neural Network or Long Short-Term Memory), and combine it with our proposed model to further improve the results.

— **Experiments for other Chain Enterprises.** In this article, we use a chain retail enterprise for a case study to evaluate our proposed model in the experiments, because the real-world commercial dataset is not easy to be obtained. In our future work, we intend to have more collaboration with other commercial companies which own a larger number of stores to obtain enough data, and further validate the effectiveness and robustness of our framework.

— **The Extension and Usage of WANT to other Applications.** Although our focus in this article is on the cold-start store site recommendation, our proposed WANT can also handle the data scarcity problem in other applications by transferring knowledge learned from source domains, such as spatial-temporal prediction and image classification. For different problems, there might be various types of feature representations learned by the feature extractor that can be transferred and integrated. Thus, we plan to extend our model by combining it with other network structures (e.g., CNN or Recurrent Neural Network), and apply them to different applications.

6 CONCLUSION

In this article, we present WANT for cold-start store site recommendation. Unlike previous transfer learning methods, our proposed approach focuses on transferring useful examples in source city by considering the transferability of different examples to reduce negative transfer. In particular, we propose a transferability weighting mechanism, which quantifies the transferability of source examples according to both the similarity of source examples to target data and the quality of source examples. Finally, we demonstrate that our model achieves the best performance on the real-world dataset among several state-of-the-art transfer learning approaches.

REFERENCES

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

[2] Zhangjie Cao, Mingsheng Long, Jianmin Wang, and Michael I. Jordan. 2018. Partial transfer learning with selective adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2724–2732.

[3] Zhangjie Cao, Kaichao You, Mingsheng Long, Jianmin Wang, and Qiang Yang. 2019. Learning to transfer examples for partial domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2985–2994.

[4] Ziliang Chen, Jingyu Zhuang, Xiaodan Liang, and Liang Lin. 2019. Blending-target domain adaptation by adversarial meta-adaptation networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2248–2257.

[5] Thomas M. Cover and Joy A. Thomas. 2012. *Elements of Information Theory*. John Wiley & Sons.

[6] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. 2014. A deep convolutional activation feature for generic visual recognition. In *International Conference on Machine Learning*. 647–655.

[7] Lisheng Fu, Thien Huu Nguyen, Bonan Min, and Ralph Grishman. 2017. Domain adaptation for relation extraction with domain adversarial neural network. In *Proceedings of the 8th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. 425–429.

[8] Yaroslav Ganin and Victor Lempitsky. 2014. Unsupervised domain adaptation by backpropagation. In *International Conference on Machine Learning*. 1180–1189.

[9] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2016. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research* 17, 1 (2016), 2096–2030.

[10] Behnam Gholami, Pritish Sahu, Ognjen Rudovic, Konstantinos Bousmalis, and Vladimir Pavlovic. 2020. Unsupervised multi-target domain adaptation: An information theoretic approach. *IEEE Transactions on Image Processing* 29 (2020), 3993–4002.

[11] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the 28th International Conference on Machine Learning*. 513–520.

[12] Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. 2012. A kernel two-sample test. *Journal of Machine Learning Research* 13, 1 (2012), 723–773.

[13] 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. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1, 4 (2018), 1–23.

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

[15] Lu Jiang, Deyu Meng, Qian Zhao, Shiguang Shan, and Alexander G. Hauptmann. 2015. Self-paced curriculum learning. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence*.

[16] 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.

[17] 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 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct*. 149–152.

[18] 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 of Things Journal* 6, 4 (2019), 7319–7333.

[19] 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.

[20] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I. Jordan. 2015. Learning transferable features with deep adaptation networks. In *International Conference on Machine Learning*. 97–105.

[21] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I. Jordan. 2017. Deep transfer learning with joint adaptation networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR.org, 2208–2217.

[22] Daniel Lowd and Christopher Meek. 2005. Adversarial learning. In *Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*. 641–647.

[23] Liqiang Nie, Yongqi Li, Fuli Feng, Xuemeng Song, Meng Wang, and Yinglong Wang. 2020. Large-scale question tagging via joint question-topic embedding learning. *ACM Transactions on Information Systems* 38, 2 (2020), 1–23.

[24] Liqiang Nie, Xiang Wang, Jianglong Zhang, Xiangnan He, Hanwang Zhang, Richang Hong, and Qi Tian. 2017. Enhancing micro-video understanding by harnessing external sounds. In *Proceedings of the 25th ACM International Conference on Multimedia*. 1192–1200.

[25] 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.

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

[27] Xingchao Peng, Zijun Huang, Ximeng Sun, and Kate Saenko. 2019. Domain agnostic learning with disentangled representations. In *International Conference on Machine Learning*. 5102–5112.

[28] Guo-Jun Qi, Charu Aggarwal, Yong Rui, Qi Tian, Shiyu Chang, and Thomas Huang. 2011. Towards cross-category knowledge propagation for learning visual concepts. In *Proceedings of the 2011 Conference on Computer Vision and Pattern Recognition*. IEEE, 897–904.

[29] Guo-Jun Qi, Charu C. Aggarwal, and Thomas Huang. 2013. Link prediction across networks by biased cross-network sampling. In *Proceedings of the 2013 IEEE 29th International Conference on Data Engineering (ICDE'13)*. IEEE, 793–804.

[30] Guo-Jun Qi, Wei Liu, Charu Aggarwal, and Thomas Huang. 2016. Joint intermodal and intramodal label transfers for extremely rare or unseen classes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39, 7 (2016), 1360–1373.

[31] Guo-Jun Qi, Liheng Zhang, Hao Hu, Marzieh Edraki, Jingdong Wang, and Xian-Sheng Hua. 2018. Global versus localized generative adversarial nets. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1517–1525.

[32] Darsh J. Shah, Tao Lei, Alessandro Moschitti, Salvatore Romeo, and Preslav Nakov. 2018. Adversarial domain adaptation for duplicate question detection. *arXiv preprint arXiv:1809.02255* (2018).

[33] Yang Shu, Zhangjie Cao, Mingsheng Long, and Jianmin Wang. 2019. Transferable curriculum for weakly-supervised domain adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. 4951–4958.

[34] Lingxiao Song, Man Zhang, Xiang Wu, and Ran He. 2018. Adversarial discriminative heterogeneous face recognition. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*.

[35] Lisa Torrey and Jude Shavlik. 2010. Transfer learning. In *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques*. IGI Global, 242–264.

[36] Eugene Tuv, Alexander Borisov, George Runger, and Kari Torkkola. 2009. Feature selection with ensembles, artificial variables, and redundancy elimination. *Journal of Machine Learning Research* 10 (2009), 1341–1366.

[37] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. 2017. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 7167–7176.

[38] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. 2014. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474* (2014).

[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] Jingdong Wang, Zhe Zhao, Jiazhen Zhou, Hao Wang, Bin Cui, and Guojun Qi. 2012. Recommending Flickr groups with social topic model. *Information Retrieval* 15, 3–4 (2012), 278–295.

[41] 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).

[42] 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.

[43] 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 Transactions on Knowledge Discovery from Data* 14, 2 (2020), 1–28.

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

[45] Jing Zhang, Zewei Ding, Wanqing Li, and Philip Ogunbona. 2018. Importance weighted adversarial nets for partial domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 8156–8164.

[46] Yiru Zhao, Zhongming Jin, Guo-jun Qi, Hongtao Lu, and Xian-sheng Hua. 2018. An adversarial approach to hard triplet generation. In *Proceedings of the European Conference on Computer Vision (ECCV'18)*. 501–517.

Received June 2020; revised October 2020; accepted December 2020