Unsupervised Knowledge Adaptation for Passenger Demand Forecasting

Can Li\textsuperscript{a}, Lei Bai\textsuperscript{b}, Wei Liu\textsuperscript{c,∗}, Lina Yao\textsuperscript{a}, S Travis Waller\textsuperscript{d}

\textsuperscript{a}School of Computer Science and Engineering, University of New South Wales, Sydney, NSW 2052, Australia
\textsuperscript{b}School of Electrical and Information Engineering, University of Sydney, Sydney, NSW 2008, Australia
\textsuperscript{c}Department of Aeronautical and Aviation Engineering, The Hong Kong Polytechnic University, Hong Kong, China
\textsuperscript{d}Research Centre for Integrated Transport Innovation, School of Civil and Environmental Engineering, University of New South Wales, Sydney, NSW 2052, Australia

Abstract

Considering the multimodal nature of transport systems (e.g., train, bus) and potential cross-modal correlations (e.g., similar temporal patterns), there is a growing trend of enhancing demand forecasting accuracy by learning from multimodal data, i.e., forecasting models that are optimized with data from multiple transport modes jointly. These multimodal forecasting models are able to improve accuracy, but they can be less practical when different parts of multimodal datasets are owned by different institutions who cannot directly share data among them. While various institutions may not be able to share their data with each other directly, they may share forecasting models trained by their data, where such models cannot be used to identify the exact information from their datasets. In this context, this study proposes an \textbf{Unsupervised Knowledge Adaptation Demand Forecasting (Un-Kadf)} framework, which forecasts the demand of one mode (i.e., the target mode) by utilizing a pre-trained model based on data of another mode, but does not require direct data sharing of another transport mode (i.e., the source mode). The proposed framework utilizes the potential shared patterns among multiple transport modes for improving forecasting performance while avoiding the direct sharing of data among different institutions. Specifically, a pre-trained forecasting model is first learned based on the data of a source mode, which can capture and memorize the source travel patterns (but not the exact source data). Then, the demand data of the target dataset is encoded into an individual knowledge part and a sharing knowledge part which will extract travel patterns by two networks, i.e., individual extraction network and sharing extraction network, respectively. The unsupervised knowledge adaptation strategy is utilized to form the sharing features for further forecasting by making the pre-trained network and the sharing extraction network analogous. Extensive experiments conducted on real-world datasets from the Greater Sydney area covering four public transit modes (i.e., bus, train, light rail, and ferry) demonstrate that the proposed approach outperforms a number of baseline methods and state-of-the-art models. Our findings also illustrate that unsupervised knowledge adaptation by sharing the pre-trained model to the target transport mode can improve the forecasting performance without the dependence on direct data sharing.

\textit{Keywords:} Demand Forecasting, Unsupervised Learning, Knowledge Adaptation

1. Introduction

Artificial intelligence and machine learning algorithms have now been widely adopted for demand forecasting given the increasing availability of multi-source datasets (Chen et al., 2020). Earlier studies often utilize statistical time-series methods and traditional machine learning methods to explore the temporal information for traffic prediction, such as Auto-Regressive Integrated Moving Average (ARIMA) (Lippi et al., 2013), Kalman Filter (Xue et al., 2015), and Support...
Vector Machine (SVM) (Feng et al., 2018). Given its strong capability to deal with non-linear relations and large-scale data, deep learning has attracted substantial attention for demand prediction in recent years. Specifically, Recurrent Neural Network (RNN) and its variants such as Long Short-Term Memory (LSTM) (Kim et al., 2020) and Gated Recurrent Unit (GRU) (Peng et al., 2021) are used for temporal relations extraction and analysis. To further explore the spatial correlations, Convolution Neural Network (CNN) (Ma et al., 2018; Liu et al., 2019; Zhang et al., 2019) and Graph Convolution Network (GCN) (Zhang et al., 2019; Cui et al., 2020; Guo et al., 2020; Tang et al., 2021) are utilized to characterize spatial features, which help improve the demand/traffic prediction accuracy.

While the aforementioned works have achieved competent prediction accuracy, they only focus on the exploration of a single transport mode and do not utilize the potential similarities/correlations among different transport modes for better prediction. Modern transport systems are in nature multimodal, including various modes such as the bus, train, and light rail. There are potential similar or correlated travel patterns among different modes. For instance, Fig. 1 presents the Pearson correlation coefficient for the demand at any ferry station and the demand at any light rail station in Sydney, where a lighter color means a more significant positive correlation. As can be observed, nearly 50% pairs of ferry stations and light rail stations have a relatively high Pearson correlation coefficient (i.e., larger than 0.8), indicating the strong correlations between these two public transit modes.\footnote{The detailed correlation information of the evaluated four transport modes collected from the Greater Sydney area (i.e., bus, train, light rail, and ferry) will be explained in Subsection 3.1.}

Recognizing the correlations/similarities among various transport modes, some studies (Ye et al., 2019; Toman et al., 2020; Li et al., 2021a; Ke et al., 2021) try to co-predict demand of multiple modes by utilizing the multimodal datasets. The experimental results of these studies illustrate that the knowledge adaptation via data sharing among various transport modes is able to improve the forecasting accuracy. However, in practice, different parts (e.g., data of different modes operated by different public/private operators) of the multimodal dataset might be owned by different institutions. There might be constraints on directly sharing data among these different institutions. For example, in Hong Kong, five different companies provide bus services\footnote{https://en.wikipedia.org/wiki/Bus_services_in_Hong_Kong}, and Mass Transit Railway (MTR) is in charge of the subway operation\footnote{https://www.mtr.com.hk/en/customer/main/}. These institutions may not be able to directly share their detailed data with each other due to either privacy concerns or other practical constraints. This motivates the current study to explore new directions for enhancing passenger demand prediction that only requires indirect/limited input from other sources and does not require direct access to details of other datasets.

In particular, this study aims to improve the demand forecasting performance of the target transport mode with the mechanism of unsupervised knowledge adaptation by model sharing instead of direct data sharing in the context of the multimodal transport system in a city. To make full use of the (indirect/limited) information from the source transport mode to enhance the forecasting performance of the target transport mode, this study proposes an **Unsupervised**...
**Knowledge Adaptation Demand Forecasting (Un-Kadf)** framework for station-level (or location-based) demand forecasting. To illustrate the mechanism of model sharing between two transit modes, Fig. 2 displays the framework of the model sharing. Given the potential correlations among two datasets collected from two transport modes operated by different institutions, the two datasets can be used as the source dataset for each other, i.e., Dataset A is regarded as the source dataset for target Dataset B and vice versa. First, a pre-trained demand forecasting Model A is optimized by Dataset A. The mechanism of such a pre-trained model has been used in natural language process (NLP) (Devlin et al., 2019) and computer vision (CV) (Tan and Le, 2019), which can help dramatically improve the model performance. Then, Model A can be employed in the optimization process of Model B on Dataset B via a model sharing strategy to get Model B’, which could improve the prediction accuracy (i.e., Performance B’ is better than Performance B where Performance B is obtained only based on its own data). The learning process of Model B’ does not need direct access to any data from Dataset A. Instead, the model sharing strategy adapts the knowledge from A to B in an unsupervised manner (i.e., unsupervised knowledge adaption) (Pan and Yang, 2009).

To achieve performance improvement by unsupervised knowledge adaptation via model sharing, several challenges have to be addressed. First, different public transit modes have different numbers of stations so that the temporal information of them cannot be directly extracted by recurrent networks with the same structure. An adaptable recurrent network has to be developed that can handle such differences between the source and target datasets without destroying travel patterns. Second, unnecessary task-specific features from the source dataset may influence the forecasting results of the target dataset, which should be avoided. In particular, this study designs an encoder-decoder structure with LSTM to extract valuable knowledge from the source dataset and ensure that the adaptable pre-trained LSTM structure can be applied to the target dataset. Based on the encoder-decoder framework, the target dataset is encoded into an individual knowledge part and a sharing knowledge part. The individual part is analyzed by the individual extraction recurrent network, while the sharing part is explored by the sharing extraction recurrent network. Then, we adopt the transfer learning strategy for unsupervised knowledge adaptation to optimize the sharing extraction network based on the sharing knowledge part of the target dataset, which can utilize the information of the source dataset memorized in the adaptable LSTM structure. The aim of transfer learning is to develop a new model for the target distribution by transferring relative knowledge based on the source distribution (Pan and Yang, 2009).

The main contributions of this paper are summarized in the following. (i) This study proposes a new approach to enhance the passenger demand forecasting performance for multimodal transport systems by unsupervised knowledge adaptation from the source dataset to the target dataset without the dependence of direct data sharing. (ii) The proposed **Un-Kadf** framework helps address the unsupervised knowledge adaptation problem via designing the pre-trained recurrent network optimized by the source dataset and sharing the pre-trained network to the target dataset for further optimization. (iii) This study conducts comprehensive experiments on large-scale real-world public transport datasets with four modes. The results show that the proposed model significantly outperforms existing methods and demonstrate the potential of boosting multimodal demand forecasting accuracy by unsupervised knowledge adaptation via model sharing.

The rest of this paper is organized as follows. We introduce the related literature in Section 2. We then discuss the case study dataset and define the demand prediction problems in Section 3. Section 4 presents the proposed **Un-Kadf** model. The evaluation of the proposed method and comparison with other existing methods are presented in Section 5. Finally, Section 6 concludes this paper.

**2. Related Work**

This section first reviews relevant data-driven studies on demand forecasting (both single-mode and multi-mode). Then, transport forecasting problems dealing with knowledge adaptation methods are discussed.
2.1. Demand Forecasting under Single Travel Mode

Given the increasing availability of transport datasets, data-driven demand forecasting has been received much attention in recent years (Ma and Qian, 2018). A large number of studies examine the estimation of demand with the data collected from the one transit mode in concern via the usage of statistical time-series methods or machine learning strategies, especially deep-learning-based models.

In particular, at earlier stages, studies for demand prediction are mainly based on statistical methods and traditional time-series regression models. For instance, Auto-Regressive Integrated Moving Average (ARIMA) (Lippi et al., 2013), Kalman Filter (Xue et al., 2015), Support Vector Machine (SVM) (Feng et al., 2018), Vector Auto-regression (Cheng et al., 2022) and their variants (Moreira-Matias et al., 2013) have been widely utilized. However, these strategies have limited capability to deal with non-linear temporal correlations and large-scale datasets for precise demand forecasting.

To extract non-linear temporal information for large-scale datasets, deep-learning-based models are adopted for prediction problems under single target transport mode, including Fully Connected Layer (FCL) (Lv et al., 2014) and RNN-based networks (e.g., LSTM and GRU) (Xu et al., 2017; Kim et al., 2020). In detail, a series of AutoEncoder is stacked in Lv et al. (2014) to learn generic traffic flow features where AutoEncoder consists of multiple FCLs. FCLs can hardly capture the long-term knowledge, which motivates the utilization of RNN-based models for demand forecasting. For instance, both Xu et al. (2017) and Kim et al. (2020) designed the predictive framework based on LSTM for taxi demand prediction.

Considering the relevance of spatial information for demand estimation, CNN is introduced for transport problems and has been combined with RNN-based models to obtain a more comprehensive understanding of spatial-temporal correlations (Ke et al., 2018; Liu et al., 2019; Guo et al., 2019). Given that CNN can only handle relations among adjacent areas, GCN is further applied for non-Euclidean spatial information extraction. Similarly, a series of studies (Li et al., 2019; Geng et al., 2019; Jin et al., 2020; Bai et al., 2020) combine GCN and recurrent networks to capture spatial-temporal knowledge for better demand forecasting. Moreover, in order to reduce the error accumulation caused by the iteration of RNN, Bai et al. (2019) and Li et al. (2020a) stack a set of gated graph convolution layers to model spatial and temporal information simultaneously for further single-mode demand prediction. In addition, to avoid over simplistic integration of heterogeneous data, Zhou et al. (2021) takes advantage of neural ordinary differential equations (ODE) to capture the continuous-time dynamics of the latent states for further forecasting.

2.2. Demand Forecasting under Multiple Travel Modes

Correlations and similarities among multiple different transport modes provide new opportunities and dimensions to enhance the demand forecasting performance, which has been receiving growing attention recently.

Ye et al. (2019) designs a co-prediction model to predict the pick-up and drop-off demand for taxis and bikes with the incorporation of heterogeneous LSTM. Ke et al. (2021) focuses on the demand of ride-hailing systems to predict solo and shared service rides jointly by constructing multi-graph convolutional networks. Furthermore, Li et al. (2020b) designs a recurrent network for demand prediction of the station-intensive travel mode and the station-sparse travel mode simultaneously to improve the forecasting accuracy for the station-sparse mode. A larger range of different transport modes are explored in Toman et al. (2020), including taxis, bikes, subways, and vehicles operated by transit network companies (TNCs). The vector autoregressive model with exogenous predictors is fit to predict the demand. Training multiple modes jointly for demand prediction enhancement has been verified in the aforementioned studies. However, different parts of multimodal datasets might be owned by different institutions who cannot directly share data among them, which is often the case in large cities with many different public and private operators. Existing multimodal demand prediction methods might not be applicable when such data sharing issues exist. Thus, this study aims to propose a demand prediction method that does not need direct sharing of data, but instead takes advantage of a pre-trained network optimized by the source dataset to boost the prediction performance of the target dataset.
2.3. Transport Forecasting with Knowledge Adaptation

Knowledge adaptation strategies have been adopted to improve the prediction performance of the target dataset for transport systems. As a popular knowledge adaptation method, transfer learning is to utilize previously developed and trained learning models to adapt the knowledge learned from the source dataset to help learn the information of the target dataset by hunting for similarities among them (Pan and Yang, 2009).

Several works have explored transfer learning in the transport prediction area. However, they mainly target the data scarcity problem caused by the unbalanced development levels of different cities and depend on the direct access to all the relevant datasets Wang et al. (2019); Yao et al. (2019). In detail, Wang et al. (2019) transfers the knowledge from a data-rich (e.g., with 90 days data) source city to a data-scarce (e.g., with 10 days data) target city by learning an inter-city region matching function to match two similar regions for crowd flow prediction. Meta-learning (Finn and Levine, 2018) is adopted in Yao et al. (2019) to apply the knowledge learned from multiple cities to increase the transferring stability for spatial-temporal forecasting of the target city.

Similarly, Li et al. (2021c,b) improve the performance of the traffic flow prediction by analyzing similarities among multiple links in the highway. Li et al. (2021a) develops the memory-augmented recurrent network to adapt knowledge from the station-intensive mode to the station-sparse mode to improve the forecasting accuracy for the multimodal public transport system where the network is optimized by two datasets simultaneously. These methods have shown their capability to improve the forecasting performance for target datasets, which rely on full direct access to the source dataset and cannot be applied to dealing with the scenario with no direct access to the source dataset. Different from these works, we adopt the method of unsupervised knowledge adaptation to improve the prediction accuracy of the target dataset by adapting the knowledge memorized in the pre-trained model learned from the source dataset, which does not require direct sharing of the source dataset.

3. Dataset and Demand Forecasting Problem Formulation

This section introduces the dataset collected in the Greater Sydney area that contains multiple transit modes. Then, the possible correlations/similarities among various transport modes are illustrated, highlighting the potential of knowledge adaptation from the source mode to help enhance the forecasting performance of the target mode.

3.1. Dataset and Mode Correlation

The dataset is collected from the Greater Sydney Area covering main public transport services, including buses, trains, ferries, and light rails, from 01/Apr/2017 to 30/Jun/2017. The dataset records traffic information of 24 hours a day, covering 6.37 million users. All lines’ information, including tap-on and tap-off location (e.g., name, longitude, and latitude of the station), time, and the number of passengers getting on and off, are used in experiments. The dataset does not involve personal information that can be used to identify individuals.

The potential correlations are visualized for four public transit modes (i.e., buses, trains, ferries, and light rails) in Sydney, where the Pearson correlation coefficient is adopted. The coefficient ranges from $-1$ to $1$, where the larger absolute value represents a higher correlation between the two modes. A positive value means a positive correlation, while a negative value means a negative correlation.

In particular, we calculate correlations between the station-based demands of different modes, and Fig. 3 shows the distribution of coefficient values for four modes where the X-axis represents the range of coefficient values, and the Y-axis represents proportions of stations in the corresponding range. As can be seen from these figures, the proportion of stations with a correlation coefficient greater than 0.6 is between 30% and 40% approximately, which illustrates the relatively wide and high correlations among any two modes.
3.2. Demand Forecasting Problem Formulation

In this subsection, we formulate the passenger demand forecasting problem to be studied. Note that the following problem formulation is based on station-based demand. It can be readily adapted to non-station-based travel modes such as taxi and ride-sourcing service by introducing the zone-based demand concept, where “station” is then replaced by “zone”.

**Demand Series.** In the multimodal transport system, we denote the demand of station $i$ at time step $t$ as a scalar $x_{D,i}^t$ for transport mode $D$ (e.g., train), which means the passenger demand between time step $t - 1$ and time step $t$. Then, the demand of station $i$ over time can be represented as a vector $X_{D,i} = \{x_{D,i}^1, x_{D,i}^2, \ldots, x_{D,i}^t, \ldots, x_{D,i}^T\}$ where $T$ is the total number of time steps. Similarly, all stations for transport mode $D$ at time step $t$ can be represented as a vector $X_D = \{x_{D,1}^t, x_{D,2}^t, \ldots, x_{D,i}^t, \ldots, x_{D,N_D}^t\}$. Moreover, we let $X_D = \{X_{D,1}^T, X_{D,2}^T, \ldots, X_{D,T}^T\}$ denote the demand series of transport mode $D$ over time.

**Demand Forecasting Problem with Unsupervised Knowledge Adaptation.** Given a sequence of demand $\{x_{D}^{T-\tau+1}, \ldots, x_{D}^{T-1}, x_{D}^{T}\}$ of the target mode $P$ where $\tau$ is the number of time steps utilized for prediction, the problem is defined as forecasting the demand of each station of the target mode in the future time step $T + 1$:

$$\hat{X}_{P}^{T+1} = \Gamma(X_{D}^{T-\tau+1}, \ldots, X_{P}^{T-1}, X_{P}^{T})$$

(1)

where $\Gamma(\cdot)$ is the prediction function to be learned by the model sharing network, and $\hat{X}_{P}^{T+1}$ represents the predicted demand value at time step $T + 1$ of the target dataset. The above station-based formulation can be readily adapted for zone/region-based formulation if we consider zone/region-based demand rather than station-based demand.

Given the potential correlations of travel patterns between the source mode $S$ and the target mode $P$, a pre-trained recurrent model $\Phi_s$ learned based on the passenger demand of the source transport mode $S$ can help to enhance the forecasting performance of $P$. Thus, Eq. (1) can be further written as:

$$\hat{X}_{P}^{T+1} = \Gamma(X_{D}^{T-\tau+1}, \ldots, X_{P}^{T-1}, X_{P}^{T}, \Phi_s)$$

(2)

It should be noted that the training procedure of the target mode only takes advantage of the pre-trained model $\Phi_s$, and does not access the exact source dataset. Such an unsupervised knowledge adaptation strategy is different from adaptation methods in the literature, which directly utilize the source data when training the forecasting model for the target dataset.
4. Methodology

In this section, we introduce the structure and methodology of the proposed Un-Kadf framework as depicted in Fig. 2. The first step is to deal with the demand data of the source dataset, where the temporal knowledge (not the data itself) can be recorded in the pre-trained recurrent network and further shared to the target transport mode. This is shown in Fig. 4a. Then, we elaborate on how to adapt useful knowledge from the pre-trained network to the target dataset to enhance the forecasting performance based on unsupervised knowledge adaptation. The structure of the model sharing network for unsupervised knowledge adaptation is shown in Fig. 4b.

(a) The Procedure of Source Knowledge Extraction: \( X^t_S, X^{Se}_S, X^{Sd}_S \), and \( h^t_S \) denote the raw demand, the encoded vector, the decoded vector, and the hidden state at time step \( t \), respectively. \( h^t_S \) is the hidden state emitted by the pre-trained adaptable recurrent network \( LSTM_A \). \( \hat{X}^t_{S+1} \) is the predicted demand at time step \( t + 1 \). \( S \) represents the source dataset.

(b) The Structure of Model Sharing Network: \( X^t_P, X^{PI}_P, X^{PH}_P \), and \( X^{Pd}_P \) denote the raw demand, the encoded individual knowledge part, the encoded sharing knowledge part, and the decoded vector at time step \( t \), respectively. \( h^t_{PI} \) and \( h^t_{PH} \) are according hidden states while \( c^t_{PI} \) and \( c^t_{PH} \) are according memory cell emitted by the individual recurrent network \( LSTM_I \) and sharing recurrent network \( LSTM_H \). \( \bar{c}^t_{Ps} \) is the memory cell emitted by pre-trained adaptable recurrent network \( LSTM_A \). And \( P \) represents the target dataset.

![Fig. 4. The Overall Structure of Un-Kadf](image)

4.1. Source Knowledge Extraction

Transfer learning consists of a two-stage learning framework, i.e., the pre-training stage to capture knowledge from source task(s), and the fine-tune stage to transfer the captured knowledge to target task(s) (Pan and Yang, 2009). It provides a flexible manner to train efficiently and perform well in the computer vision (CV) area and the natural language process (NLP) area among tasks with similar distributions. As discussed in Subsection 3.1, similarities/correlations often exist among different transport modes, which can be potentially used to enhance demand forecasting. Following the success of transfer learning in CV and NLP areas to take advantage of the profitable knowledge learned from the source dataset, a pre-trained recurrent network is designed to extract and memorize the information, and an encoder-decoder framework is utilized to allow the pre-trained network to be adaptable to the target dataset. It is noteworthy that the pre-trained network rather than the source dataset is utilized for further demand forecasting, and there is no need of direct data sharing.
The following fully connected layer is used to combine hidden states predicted demand value of the target dataset as:

\[ X_{Sc}^t = \tanh(W_e X_S^t + b_e) \]  

where \( W_e \in \mathbb{R}^{K \times N_S} \) and \( b_e \in \mathbb{R}^K \) represent the weight matrix and bias vector, respectively.

Then, to extract the temporal information of the source dataset, the encoded vector is sent into the LSTM layer, following a fully connected layer to predict the demand data, a decoder is designed to recover the raw data, which is represented as:

\[ X_{Sd}^t = \tanh(W_{ds} X_{Sc}^t + b_{ds}) \]  

where \( W_{ds} \in \mathbb{R}^{N_S \times K} \) and \( b_{ds} \in \mathbb{R}^{N_S} \) represent the weight matrix and bias vector, respectively.

To ensure that the information recorded in the pre-trained network can describe patterns of the source dataset more comprehensively, we need to optimize the network to avoid adapting noises or useless knowledge (Pan and Yang, 2009). In the training process of the source dataset, the objective is to minimize the error between the real demand and the predicted values and the error between the real demand and recovered demand values. The loss function is defined as the mean squared error for time step length \( T \), which is formulated as follows:

\[ L(\theta) = \sum_{t=T}^{T+\tau} ||\hat{X}_S^t - X_S^t|| + \sum_{t=T}^{T+\tau-1} ||X_{Sd}^t - X_S^t|| \]  

where \( \hat{X}_S^{T+1} \) represents the predicted demand value at time step \( T + 1 \) of the source dataset and \( \theta \) denotes all the learnable parameters in the pre-trained network. And the network is trained via back-propagation and the Adam optimizer.

**4.2. Model Sharing Network**

As revealed in Subsection 3.1, two transport modes may share some similarities and hold disparities at the same time. Therefore, the raw demand data of the target dataset is encoded into two parts, i.e., the encoded individual knowledge part \( X_{PI}^t \in \mathbb{R}^K \) and the encoded sharing knowledge part \( X_{PH}^t \in \mathbb{R}^K \) which are calculated as:

\[ X_{PI}^t = \tanh(W_{PI} X_P^t + b_{PI}) \]  
\[ X_{PH}^t = \tanh(W_{PH} X_P^t + b_{PH}) \]

where \( W_{PI} \in \mathbb{R}^{K \times N_S} \), \( W_{PH} \in \mathbb{R}^{K \times N_S} \) are learnable weight matrices and \( b_{IS} \in \mathbb{R}^K \) and \( b_{IS} \in \mathbb{R}^K \) are bias vectors.

Then, to analyze the temporal knowledge of the individual and sharing parts independently, two different LSTM networks are applied to get hidden states and internal memory cells as:

\[ h_{PI}^t, c_{PI}^t = LSTM_I(X_{PI}^t, h_{PI}^{t-1}, c_{PI}^{t-1}) \]  
\[ h_{PH}^t, c_{PH}^t = LSTM_H(X_{PH}^t, h_{PH}^{t-1}, c_{PH}^{t-1}) \]

The following fully connected layer is used to combine hidden states \( h_{PI}^t \) and \( h_{PH}^t \) to get the predicted demand value of the target dataset as:

\[ \hat{X}_P^{t+1} = \tanh(W_P(h_{PI}^t + h_{PH}^t) + b_P) \]
where $W_P \in \mathbb{R}^{N_S \times 2K}$ and $b_P \in \mathbb{R}^{N_S}$ denote the weight matrix and bias vector, respectively.

Furthermore, as introduced in Subsection 4.1, the pre-trained network memorizes travel patterns of the source dataset. Therefore, motivated by the success of transfer learning for trajectory prediction and precipitation nowcasting (Yao et al., 2020), we adopt such a strategy for unsupervised knowledge adaptation. In detail, to make the sharing recurrent network LSTM$_H$ be able to represent the sharing knowledge of both datasets, the sharing knowledge part is also sent into the adaptable pre-trained recurrent network LSTM$_A$ to get the internal memory cell $c_{PS}^t$. The values of $c_{PS}^t$ and $c_{PH}^t$ are similar, which means that the parameters (e.g., weight matrices and bias vectors) of the pre-trained network and the sharing network are similar. The parameters of the pre-trained recurrent network LSTM$_A$ are frozen during the training process so that the sharing network can focus on more sharing patterns.

Similar to the decoder in the pre-trained recurrent network, encoded individual and sharing parts are recovered by the decoder to improve the representation ability of the encoded vectors as:

$$X_{Pd}^t = \tanh(W_{dP}(X_{PI}^t + X_{PH}^t) + b_{dP})$$  \hspace{1cm} (9)

where $W_{dP} \in \mathbb{R}^{N_S \times 2K}$ and $b_{dP} \in \mathbb{R}^{N_S}$ are the weight matrix and bias vector, respectively.

### 4.3. Training Strategy

In the training process, one objective is to minimize the error between the values of internal memory cells obtained from the sharing recurrent network and pre-trained recurrent network to learn sharing travel patterns. Moreover, to enhance the forecasting performance of the target dataset in the training process of the model sharing recurrent network, the error between the true demand and the predicted values, and the error between the real demand and recovered demand data are needed to be minimized. The loss function is defined as the mean squared error for time step length $\tau$, which is represented as:

$$L_1 = \frac{1}{T+\tau} \sum_{t=T+1}^{T+\tau} ||\hat{X}_P^t - X_P^t||$$

$$L_2 = \frac{1}{T+\tau-1} \sum_{t=T}^{T+\tau-1} ||X_{Pd}^t - X_{PI}^t||$$

$$L_3 = \frac{1}{T+\tau-1} \sum_{t=T}^{T+\tau-1} ||c_{PH}^t - c_{PS}^t||$$

$$L(\Pi) = L_1 + \gamma \times L_2 + \beta \times L_3$$  \hspace{1cm} (10)

where $\Pi$ denotes all the learnable parameters in the model sharing network. $\gamma$ and $\beta$ are hyper-parameters to decide the weights of data recovery loss $L_2$ and memory update loss $L_3$, which will be evaluated in Subsection 5.3. And the network is trained via back-propagation and the Adam optimizer. Note the training procedures of the knowledge adaptation framework are unsupervised, which does not need the exact source data for optimization as discussed in Section 1.

### 5. Experiments

This section presents results from experiments carried out in this study. The experimental setup, including dataset setting, evaluation matrices, and network implementation, is presented at first. The components of the proposed model and hyper-parameter sensitivity are then discussed and analyzed. Furthermore, we compare the proposed model with a list of baseline models and state-of-the-art methods.
5.1. Experimental Setup

Dataset Setting. The demand data is normalized by Min-Max normalization for training and re-scaled to the actual values for evaluating the prediction performance. To test the performance of Un-Kadf, 60% data are used for training, 20% data are used for validation, and the rest for testing. In each experiment, we use the data of one transport mode as the source dataset and the data of another mode as the target dataset. The time step length we choose in the experiments is one hour. Since the data volume is too large, which contains some meaningless data (e.g., the demand is zero for more than 80% of time steps in one day), we drop the stations with zero demand per hour for more than 60% of time steps. The number of stations for the bus, train, light rail, and ferry stations is 472, 250, 23, and 33, respectively. And we use the previous 12 time steps (12 hours) to predict the public transport demand in the next time step (next one hour).

Evaluation Matrices. Three evaluation matrices are used to evaluate the proposed model: Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE).

Network Implementation. The batch size is set to 64 and the learning rate is fixed as 0.0001. 1000 episodes are run for model training. The proposed model is tuned with the hyper-parameters $\gamma$ and $\beta$ in the loss function Eq. (10) (from 0.1 to 1.0 with a step size of 0.1). The number of hidden states is fixed as 64 of encoder layers and LSTM layers for pre-trained model adaptation. Different hyper-parameter values for different transit modes are set, which are shown in Table 1, and the details of hyper-parameter tuning will be introduced in Subsection 5.3.

Table 1
Hyper-parameter Setting of Un-Kadf

| Mode     | Value of $\gamma$ | Value of $\beta$ |
|----------|-------------------|-------------------|
| Bus      | 0.4 Light Rail    | 1 Ferry           |
| Train    | 0.6 Light Rail    | 0.6 Ferry         |
| Light Rail | 0.9 Train        | 0.3 Ferry         |
| Ferry    | 0.1 Train         | 0.6 Light Rail    |
|          | Train             | 1 Light Rail      |
|          | 1 Light Rail      | 0.4 Ferry         |
|          | 0.7 Light Rail    | 0.9 Ferry         |
|          | 0.3 Train         | 0.1 Ferry         |
|          | 0.6 Train         | 0.6 Light Rail    |
|          | 1 Train           | 0.4 Light Rail    |
|          | 0.7 Light Rail    | 0.9 Ferry         |
|          | 0.6 Light Rail    | 0.6 Light Rail    |

5.2. Network Architecture Analysis

To provide a more concrete understanding of the effectiveness of the encoder-decoder framework and the unsupervised knowledge adaptation strategy described in Section 4, this subsection compares the proposed method against several variants as described below:

- **LSTM**: The LSTM layer is directly used to model long-and-short-term dependencies, followed by a fully connected layer to predict the demand for each transport mode independently.

- **Encoder-Adaptation**: The encoded vectors of the target dataset are only sent into LSTM layers for temporal knowledge extraction. The decoder to recover the raw demand data is removed in this architecture.

- **Encoder-Decoder**: The knowledge adaptation from the source dataset to the target dataset is removed, which means that only encoder and decoder components are used to help LSTM for features capturing and further demand prediction.

- **Encoder-LSTM**: This architecture removes both the decoder and adaptation module. The raw demand data is sent into encoders and LSTM layers for further prediction.

The forecasting results under different architectures are listed in Table 2. Since Encoder-Adaptation and Un-Kadf deal with two transport modes (i.e., the source mode and target mod), the results of them are calculated as average values of the forecasting results for each mode. Several observations are summarized as follows based on the experiments of network architectures:

First, the forecasting results of LSTM and Encoder-LSTM are similar, which do not show a significant difference. These results illustrate that the design of the independent encoder to split the target dataset into individual and sharing knowledge parts does not destroy the travel patterns.
Table 2
Network Architecture Analysis with Different variants

| Evaluation Matrices | Mode  | LSTM    | Encoder-Adaptation | Encoder-Decoder | Encoder-LSTM | Un-Kadf |
|---------------------|-------|---------|--------------------|-----------------|--------------|---------|
| MAE                 | Bus   | 8.750   | 8.233              | 8.397           | 8.545        | 7.841   |
|                     | Train | 24.357  | 22.810             | 21.863          | 24.071       | 19.614  |
|                     | Light Rail | 12.106 | 10.863             | 10.908          | 12.253       | 10.530  |
|                     | Ferry | 14.298  | 13.924             | 14.232          | 14.361       | 13.039  |
| RMSE                | Bus   | 20.108  | 18.824             | 19.274          | 19.832       | 17.814  |
|                     | Train | 70.262  | 65.662             | 63.635          | 69.422       | 54.693  |
|                     | Light Rail | 25.819 | 21.899             | 22.166          | 25.144       | 21.381  |
|                     | Ferry | 37.356  | 38.745             | 38.788          | 38.941       | 35.832  |
| MAPE                | Bus   | 0.169   | 0.155              | 0.157           | 0.168        | 0.149   |
|                     | Train | 0.157   | 0.151              | 0.146           | 0.158        | 0.130   |
|                     | Light Rail | 0.180 | 0.165              | 0.168           | 0.183        | 0.160   |
|                     | Ferry | 0.210   | 0.192              | 0.199           | 0.206        | 0.187   |

of demand data. It illustrates the rationality of the encoder introduced in Subsection 4.2. Second, Encoder-LSTM and LSTM yield higher MAE, RMSE, and MAPE than Encoder-Decoder, which means that the recovery procedure operated by the decoder is able to enhance the representation ability of the encoded individual and sharing knowledge vectors of the target dataset. Besides, when compared with Encoder-LSTM and LSTM, Encoder-Adaptation achieves higher accuracy for four transit modes. This highlights the usefulness of adapting the knowledge recorded in the pre-trained recurrent network optimized using the source dataset does improve the prediction performance of the target dataset. Furthermore, the overall model Un-Kadf gains lower values of MAE, RMSE, and MAPE than other listed architectures. In view of the above, the encoder-decoder framework that formulates encoded vectors and the unsupervised knowledge adaptation strategy to adapt the learned knowledge from the source dataset to the target dataset can both help improve the forecasting performance.

5.3. Hyper-parameter Sensitivity

The sensitivity of hyper-parameters $\gamma$ and $\beta$ in the loss function Eq. (10) to decide the weights of the data recovery loss and the memory update loss for Un-Kadf training are studied in this subsection.

Table 3
Forecasting Performance of Special Hyper-parameter Setting

| Hyper-parameter | $\gamma=0$ | $\beta=0$ | $\gamma=\beta=0$ |
|-----------------|------------|------------|------------------|
| Mode            | MAE        | RMSE       | MAPE             | MAE        | RMSE       | MAPE             |
| Bus             | 8.211      | 18.908     | 0.155            | 8.357      | 18.961     | 0.156            | 8.573      | 20.112     | 0.172 |
| Train           | 22.143     | 64.505     | 0.149            | 21.649     | 63.968     | 0.143            | 23.569     | 68.692     | 0.156 |
| Light Rail      | 10.899     | 22.219     | 0.164            | 10.849     | 22.060     | 0.164            | 12.750     | 25.278     | 0.184 |
| Ferry           | 13.781     | 37.870     | 0.191            | 14.163     | 38.409     | 0.195            | 14.364     | 39.646     | 0.208 |

Different settings of hyper-parameters are tested for the proposed model based on the validation set. The setting of hyper-parameters that yields the best performance is then utilized to conduct testing based on the testing set, which has been listed in Table 1. Due to the large number of experiments conducted for hyper-parameters (i.e., $10 \times 10 \times 3 = 300$ experiments conducted for each transport mode), we only show MAE for two groups of experimental results in the manuscript, i.e., the performance with $\gamma$ changing from 0.1 to 1.0 and $\beta$ fixed as the value yields the best performance for each mode is shown in Fig. 5, and the performance with $\beta$ changing from 0.1 to 1.0 and $\gamma$ fixed as the value yields the best performance for each mode is presented in Fig. 6. As can be seen from these figures, MAE does not change drastically when the hyper-parameters change, which illustrates the robustness of the proposed model with respect to these hyper-parameters. To further illustrate this, we calculate the standard deviation of the listed MAE in the aforementioned
figures. When fixing $\beta$, the values of standard deviation are 0.062 for the bus, 0.365 for the train, 0.172 for the light rail, and 0.336 for the ferry. When fixing $\gamma$, the values of standard deviation are 0.129 for the bus, 0.462 for the train, 0.150 for the light rail, and 0.267 for the ferry. These stationary experimental results illustrate the stability of Un-Kadf, which is not sensitive to the settings of hyper-parameters.

Furthermore, it is noteworthy that when the value of $\gamma$ is set to zero, the decoder module to recover the raw demand data becomes invalid, which becomes similar to the Encoder-Adaptation structure. And when the value of $\beta$ becomes zero, the knowledge learned from the source dataset recorded in the pre-trained network can not adapt to the target dataset for forecasting enhancement, which becomes similar to the Encoder-Decoder structure. Also, if $\gamma$ and $\beta$ are zero, the network becomes similar to Encoder-LSTM. The results (i.e., the average values of the forecasting results for each mode) of these three architectures are listed in Table 3, which is consistent with the performance shown in Table 2. Their less satisfactory performance demonstrates the effectiveness of the encoder-decoder framework and unsupervised knowledge adaptation via model sharing.

5.4. Overall Comparison

In this subsection, we compare the proposed model with several baseline models and state-of-the-art methods, which are summarized in the following.

- **Historical Average (HA)**: The predicted demand is computed as the average values of historical demand at the same time interval of every day.

- **Linear Regression (LR)**: The relations between variables are modeled to minimize the sum of the squares of the errors for forecasting.

- **eXtreme Gradient Boosting (Xgboost)** (Chen and Guestrin, 2016): Based on the gradient boosting tree, XGBoost is proposed to incorporate the advantages of Bagging integrated learning methods in the evolution process. And the tree-booster is used for demand forecasting.

- **Multilayer Perceptron (MLP)**: The neural network contains two fully connected layers to predict the demand.
• **Long-Short Term Memory (LSTM):** LSTM is a variant of Recurrent Neural Network, which is applied to process and analyze time-series information and further predict demand for each transit mode.

• **Dual-stage Attention-based Recurrent Neural Network (DA-RNN)** (Qin et al., 2017): In the construction of DA-RNN, the input attention mechanism is utilized to extract relevant driving series at each time step while the temporal attention mechanism is utilized to select relevant encoder hidden states across all time steps for time-series prediction. Since the original structure is used for one variant forecasting, we modify it to multiple variants.

• **Convolutional LSTM (ConvLSTM)** (Xingjian et al., 2015): ConvLSTM extends the structure of original LSTM to have convolutional structures to capture spatial and temporal correlations simultaneously.

• **BiLSTM-ED** (Fan et al., 2019): This network is proposed based on the framework of Transformer (Vaswani et al., 2017) which consists of the self-attention mechanism and fully connected layers without using recurrent networks or convolution operations.

• **Dual Self-Attention Network (DSANet)** (Huang et al., 2019): DSANet is composed of a global temporal convolution module and a local temporal convolution module to capture global and local temporal patterns, respectively. Moreover, the self-attention mechanism is used to model dependencies among various series. The traditional autoregressive linear model is integrated to improve the robustness of the model.

• **MT-LSTM:** Two LSTM layers are used to extract temporal correlations for each transport mode independently, and fully connected layers are adopted to analyze the implicit relations for multimodal demand forecasting.

• **Knowledge Adaptation with Attentive Multi-task Memory Network (KA2M2)** (Li et al., 2021a): A memory-augmented recurrent network is proposed to capture and store the temporal information of each transit mode. And an attention-based knowledge adaptation module is integrated to adapt relevant knowledge from the station-intensive mode to the station-sparse mode for further forecasting. Note that KA2M2 requires direct data sharing among different transport modes.

---

**Fig. 6. Sensitivity Analysis of Hyper-parameter \( \beta \)**

(a) Bus

(b) Train

(c) Light Rail

(d) Ferry
Table 4
Hyper-parameter Settings of Compared Models with Single-Mode

| Model      | Hyper-parameter   | Bus      | Train    | Light Rail | Ferry    |
|------------|-------------------|----------|----------|------------|----------|
|            | No. of Layers     | 2        | 2        | 2          | 2        |
| MLP        | No. of Units of Layers | 512, 256, 128, 64 | 128, 64 | 64, 64 |
| LSTM       | No. of Layers     | 2        | 2        | 2          | 2        |
| DA-RNN     | No. of Units of Layers | 128, 64, 64 | 64, 64 | 64, 64 |
| ConvLSTM   | No. of Units of Layers | 64, 32, 64, 32 | 32, 32 | 128, 32 |
| Transformer| No. of Encoder Layers | 1        | 3        | 3          | 3        |
|            | No. of heads for multthead-attention models | 8        | 4        | 6          | 8        |
|            | No. of Units of Layer | 64, 128 | 128 | 128 |
| DSANet     | No. of Units of Recurrent Layer | 64        | 64 | 128 |
|            | No. of Units of Convolution Layer | 64        | 32 | 64 |
|            | Filter Length     | 3        | 3        | 5          | 5        |

Table 5
Hyper-parameter Settings of AK2M2

| Mode | Cooperative Mode | Epsilon in Loss Function | Gamma | No. of Units | No. of Memory Segments | Size of Each Segment |
|------|------------------|--------------------------|-------|--------------|------------------------|---------------------|
| Bus  | Train            | 0.8                      | 0.5   | 128          | 10                     | 60                  |
|      | Light Rail       | 0.8                      | 0.5   | 128          | 5                      | 60                  |
|      | Ferry            | 0.4                      | 0.5   | 128          | 5                      | 60                  |
| Train| Bus              | 0.8                      | 0.5   | 128          | 5                      | 60                  |
|      | Light Rail       | 0.2                      | 0.5   | 128          | 5                      | 60                  |
|      | Ferry            | 0.2                      | 0.5   | 128          | 5                      | 60                  |
| Light Rail| Bus            | 0.4                      | 0.5   | 128          | 5                      | 60                  |
|      | Train            | 0.8                      | 0.5   | 128          | 5                      | 60                  |
|      | Ferry            | 0.2                      | 0.5   | 128          | 5                      | 60                  |
| Ferry| Bus              | 0.6                      | 0.5   | 128          | 10                     | 60                  |
|      | Train            | 0.8                      | 0.5   | 128          | 5                      | 60                  |
|      | Light Rail       | 0.2                      | 0.5   | 64           | 5                      | 60                  |

- **Fine-Tune** (Hinton et al., 2006): Fine-tune is one of the transfer learning methods for deep learning, which needs a relatively small amount of data. By modifying the structure and selectively loading the weight of the pre-trained network optimized by the source dataset, this method retrains the model with the target dataset.

The tested model hyper-parameters are tuned on the validation dataset to locate the best settings for the evaluated deep-based methods for comparison. The hyper-parameter settings of compared models with single-mode (MLP, LSTM, DA-RNN, ConvLSTM, BiLSTM-ED, and DSANet) are summarized in Table 4 and the setting of AK2M2 is summarized in Table 5.

The performances of all the methods as mentioned above are summarized in Table 6 and Table 7. For the models dealing with multiple modes (i.e., MT-LSTM, AK2M2, Fine-Tune, and the proposed Un-Kadf framework), the listed results in Table 6 are the average values of the forecasting results for each mode. Table 7 presents the detailed performance of models dealing with various combinations of transport modes. According to these results, we have the following observations.

First of all, classical machine learning methods HA and LR obtain higher values in terms of MAE, RMSE, and MAPE than other strategies. The forecasting performance of Xgboost is better than HA, LR, and some deep learning models in some sources (e.g., more accurate than MLP and LSTM for the ferry), which implies the effectiveness of ensemble methods.

Second, the deep-learning-based strategies normally perform better than HA and LR since
neural networks can handle non-linear relations of large-scale demand data more thoroughly. And among the tested deep models, MLP gains less satisfactory results due to its limited ability for temporal information extraction. When compared to MLP, LSTM achieves higher accuracy on four modes, demonstrating the usefulness of the recurrent network, which is used as the fundamental component for demand forecasting in the current study.

Furthermore, DA-RNN, as a state-of-the-art strategy, does not obtain better results than LSTM on the demand forecasting of the bus with abundant stations since it is proposed for univariate time-series prediction. It realizes a competent performance on the other three modes (i.e., train, light rail, and ferry) with fewer stations. In addition, RNN-based and CNN-based models (i.e., ConvLSTM and DSANet) proposed for demand forecasting with the data of one transport mode yield relatively accurate results, implying their ability to handle temporal patterns via recurrent networks and spatial patterns via convolutional networks. BiLSTM-ED, a Transformer-based neural network, shows its superior capability to distinguish efficacious travel patterns from the raw demand data for further prediction. It is of our interest to further explore methods based on Transformer for multimodal demand forecasting in the future.

The strategies designed based on graph convolution networks (GCNs) have not been tested and compared in the experiments since the station-based demand data is relatively sparse for graph construction. Training on large-scale sparse networks by GCN remains challenging (Yadav et al., 2019), which is not the research focus of this work. However, GCN-based models have become powerful tools for transport analysis in numerous studies, which will be our future direction to study spatial-temporal correlations for unsupervised knowledge adaptation and transport prediction.

Moreover, MT-LSTM optimizes two modes simultaneously for demand forecasting, which performs better than LSTM training with one mode and demonstrates that the correlations/similarities between two modes can boost prediction tasks. AK2M2 is proposed for multi-mode transport forecasting, adapting the learned knowledge from the station-intensive mode to the station-sparse mode via optimizing two datasets simultaneously. AK2M2 achieves more comprehensive performance than the proposed model Un-Kadf on the average results of the light rail and the ferry, but works worse on the bus and the train as shown in Table 6. AK2M2 is proposed for station-sparse mode (light rail and ferry) forecasting improvement and requires simultaneous access to all data sources, it is expected that it should perform better regarding station-sparse mode. However, while the proposed method in this study does not require simultaneous access to all data sources, it produces very competitive forecasting results for station-sparse mode (light rail and ferry) and performs better (on average) than AK2M2 for the other two modes. Also, our work gains more accurate results on some pairs of modes involving station-sparse mode (e.g., light rail as the target dataset and the ferry as the source dataset) as listed in Table 7. These results verify the potential of the proposed unsupervised knowledge adaptation that avoids access to detailed source data and the requirement of data sharing.
| Evaluation Matrices | Mode | Single Mode | Cooperative Mode | Fine-Tune | MT-LSTM | Ak2M2 | My Model |
|---------------------|------|-------------|------------------|-----------|---------|-------|----------|
|                     |      |             |                  | Value     | Value   | Value | Value    |
|                     |      |             |                  | PCT       | PCT     | PCT   | PCT      |
| MAE                 | Bus  | 8.750       | Train            | 8.689     | 8.520   | 7.714 | 7.777    |
|                     |      |             | Light Rail       | 8.640     | 8.388   | 8.096 | 7.871    |
|                     |      |             | Ferry            | 8.780     | 8.711   | 8.265 | 7.871    |
|                     | Train| 24.357      | Bus              | 24.101    | 22.745  | 18.296| 15.564   |
|                     |      |             | Light Rail       | 23.183    | 23.753  | 22.453| 19.724   |
|                     |      |             | Ferry            | 24.489    | 22.725  | 22.408| 19.552   |
|                     | Light Rail | 12.106  | Bus              | 11.713    | 10.928  | 9.060 | 16.539   |
|                     |      |             | Light Rail       | 11.979    | 10.361  | 11.552| 10.582   |
|                     |      |             | Ferry            | 15.469    | 13.044  | 13.044| 13.179   |
| RMSE                | Bus  | 20.108      | Train            | 22.648    | 20.805  | 17.437| 18.409   |
|                     |      |             | Light Rail       | 21.173    | 19.947  | 19.084| 17.355   |
|                     |      |             | Ferry            | 20.451    | 20.281  | 19.265| 17.681   |
|                     | Train| 70.262      | Bus              | 67.237    | 64.148  | 60.100| 65.614   |
|                     |      |             | Light Rail       | 65.489    | 65.320  | 63.920| 54.469   |
|                     |      |             | Ferry            | 65.970    | 67.495  | 63.941| 53.996   |
| Light Rail          | Bus  | 22.955      | Train            | 22.392    | 20.101  | 20.102| 21.380   |
|                     |      |             | Light Rail       | 23.601    | 21.601  | 22.054| 22.160   |
|                     |      |             | Ferry            | 38.545    | 34.008  | 33.880| 34.746   |
|                     | Train| 40.545      | Bus              | 40.068    | 34.186  | 34.113| 35.462   |
|                     |      |             | Light Rail       | 40.068    | 34.186  | 32.460| 37.318   |
| MAPE                | Bus  | 0.169       | Train            | 0.166     | 0.163   | 0.145 | 0.148    |
|                     |      |             | Light Rail       | 0.166     | 0.163   | 0.161 | 0.151    |
|                     |      |             | Ferry            | 0.169     | 0.169   | 0.168 | 0.168    |
|                     | Train| 0.157       | Bus              | 0.154     | 0.144   | 0.124 | 0.129    |
|                     |      |             | Light Rail       | 0.150     | 0.146   | 0.141 | 0.131    |
|                     |      |             | Ferry            | 0.164     | 0.152   | 0.138 | 0.130    |
| Light Rail          | Bus  | 0.180       | Train            | 0.179     | 0.159   | 0.154 | 0.161    |
|                     |      |             | Light Rail       | 0.177     | 0.154   | 0.150 | 0.158    |
|                     |      |             | Ferry            | 0.179     | 0.172   | 0.164 | 0.162    |
| Ferry               | Bus  | 0.210       | Train            | 0.238     | 0.203   | 0.169 | 0.185    |
|                     |      |             | Light Rail       | 0.231     | 0.205   | 0.197 | 0.191    |

Fine-tune, as a popular and basic transfer learning strategy, achieves better performance on the bus, train, and light rail than LSTM but works worse than other state-of-the-art models. It indicates that utilizing the pre-trained model optimized by the source dataset is able to improve the forecasting performance of the target dataset.

In general, compared to the listed state-of-the-art methods training with single-mode forecasting, MT-LSTM, and fine-tune, Un-Kadf yields improvement in all the three evaluation matrices on four transit modes. When compared to AK2M2, it performs better on the bus and train, and worse but very close on the light rail and ferry. However, AK2M2 requires direct utilization of multiple datasets simultaneously for training to some extent, while the proposed Un-Kadf avoids direct data sharing.

Overall, our results indicate that using the information recorded in the pre-trained network introduced in Subsection 4.1 optimized by the source dataset can efficiently enhance the forecasting performance of the target dataset via the model sharing network discussed in Subsection 4.2 based on the strategy of unsupervised knowledge adaptation instead of depending on the data sharing.
6. Conclusion

This study demonstrates that unsupervised knowledge/information adaptation from the source transport mode to the target transport mode via model sharing is able to improve passenger demand prediction, where the Unsupervised Knowledge Adaptation Enhanced Demand Forecasting (Un-KaDeF) framework has been proposed. In particular, we propose to extract the source knowledge based on an encoder-decoder framework and record the knowledge in the pre-trained recurrent network. Then, the pre-trained network is adapted to the target dataset instead of direct data sharing to enhance the forecasting performance since similarities/relations among multiple transport modes are helpful for demand prediction. The evaluation results on real-world datasets collected from Sydney, including four public transit modes (bus, train, light rail, and ferry), show that the proposed approach is able to improve the forecasting performance of the target dataset and achieve more accurate results.

Although the proposed methodology is tested on a public multimodal transport system, it can be adapted to systems with other modes such as private cars and ride-hailing vehicles. It can also be used for the same transport mode managed by different institutions or operated in different cities. The zone-based and city-based demand prediction problems can also be accommodated by the proposed model in this study by replacing the station-based demand setting with zone/region-based demand setting. Generally, this study provides insights on using model sharing instead of the dependence on direct data sharing to improve demand prediction performance in multimodal transport systems.

In the future, this work can be extended in several lines. For instance, motivated by the success of CNN-based and GCN-based models for demand prediction, the spatial-temporal relations among various stations or regions can be studied and further adapted via unsupervised knowledge adaptation strategies. Also, other types of features, such as weather and Point of Interest information, can be investigated for forecasting enhancement via moderate modifications of the proposed model.

Appendix A. Mechanism of LSTM

Owing to the scheme of the memory unit and gate mechanism (Hochreiter and Schmidhuber, 1997), long-term dependencies can be extracted and exploding/vanishing problems can be solved compared to classical Recurrent Neural Network (RNN) by LSTM, motivating the heavily usage in natural language process area and further in transport prediction problems. In detail, LSTM consists of an input gate $i^t \in \mathbb{R}^m$, a forget gate $f^t \in \mathbb{R}^m$, an output gate $o^t \in \mathbb{R}^m$, an internal memory cell $c^t \in \mathbb{R}^m$, and a hidden state $h^t \in \mathbb{R}^m$ where $m$ is the dimension of the hidden state. The derivation of LSTM layer is represented as:

\[
\begin{align*}
    i^t &= \sigma(W_i x^t + U_i h^{t-1} + b_i) \\
    f^t &= \sigma(W_f x^t + U_f h^{t-1} + b_f) \\
    o^t &= \sigma(W_o x^t + U_o h^{t-1} + b_o) \\
    \theta^t &= \tanh(W_\theta x^t + U_\theta h^{t-1} + b_\theta) \\
    c^t &= f^t \odot c^{t-1} + i^t \odot \theta^t \\
    h^t &= o^t \odot \tanh(c^t) 
\end{align*}
\]

where $\odot$ represents element-wise multiplication, $\sigma$ denotes the logistic sigmoid function $\sigma(u) = 1/(1 + e^{-u})$, $x^t \in \mathbb{R}^n$ is the input vector at the current time step $t$ ($n$ denotes the input dimension), and $W_i, W_f, W_o, W_\theta \in \mathbb{R}^{m \times n}$ and $U_i, U_f, U_o, U_\theta \in \mathbb{R}^{m \times m}$ are weight matrices, and $b_i, b_f, b_o, b_\theta \in \mathbb{R}^m$ are bias vectors. And we abbreviate the input data and output data of the LSTM layer as $h^t, c^t = LSTM(x^t, h^{t-1}, c^{t-1})$.

References

Bai, L., Yao, L., Kanhere, S. S., Wang, X., and Sheng, Q. Z. (2019). STG2seq: spatial-temporal graph to sequence model for multi-step passenger demand forecasting. In 28th International Joint Conference on Artificial Intelligence (IJCAI 2019), pages 1981–1987.
Bai, L., Yao, L., Li, C., Wang, X., and Wang, C. (2020). Adaptive graph convolutional recurrent network for traffic forecasting. Advances in Neural Information Processing Systems, 33:17804–17815.

Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 785–794.

Chen, X., Yang, J., and Sun, L. (2020). A nonconvex low-rank tensor completion model for spatiotemporal traffic data imputation. Transportation Research Part C: Emerging Technologies, 117:102673.

Cheng, Z., Trepanier, M., and Sun, L. (2022). Real-time forecasting of metro origin-destination matrices with high-order weighted dynamic mode decomposition. Transportation Science, pages 1–15.

Cui, Z., Ke, R., Pu, Z., Ma, X., and Wang, Y. (2020). Learning traffic as a graph: A gated graph wavelet recurrent neural network for network-scale traffic prediction. Transportation Research Part C: Emerging Technologies, 115:102620.

Devlin, J., Chang, M., Lee, K., and Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 4171–4186.

Fan, C., Zhang, Y., Pan, Y., Li, X., Zhang, C., Yuan, R., Wu, D., Wang, W., Pei, J., and Huang, H. (2019). Multi-horizon time series forecasting with temporal attention learning. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pages 2527–2535.

Feng, X., Ling, X., Zheng, H., Chen, Z., and Xu, Y. (2018). Adaptive multi-kernel svm with spatial–temporal correlation for short-term traffic flow prediction. IEEE Transactions on Intelligent Transportation Systems, 20(6):2001–2013.

Finn, C. and Levine, S. (2018). Meta-learning and universality: Deep representations and gradient descent can approximate any learning algorithm. In International Conference on Learning Representations.

Geng, X., Li, Y., Wang, L., Zhang, L., Yang, Q., Ye, J., and Liu, Y. (2019). Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 33, pages 3656–3663.

Guo, K., Hu, Y., Qian, Z., Liu, H., Zhang, K., Sun, Y., Gao, J., and Yin, B. (2020). Optimized graph convolution recurrent neural network for traffic prediction. IEEE Transactions on Intelligent Transportation Systems, 22(6):1388–1398.

Guo, S., Lin, Y., Li, S., Chen, Z., and Wan, H. (2019). Deep spatial–temporal 3d convolutional neural networks for traffic data forecasting. IEEE Transactions on Intelligent Transportation Systems, 20(10):3913–3926.

Hinton, G. E., Osindero, S., and Teh, Y. W. (2006). A fast learning algorithm for deep belief nets. Neural Computation, 18(7):1527–1554.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8):1735–1780.

Huang, S., Wang, D., Wu, X., and Tang, A. (2019). Dsanet: Dual self-attention network for multivariate time series forecasting. In Proceedings of the 28th ACM International Conference on Information and Knowledge Management, pages 2129–2132.

Jin, G., Cui, Y., Zeng, L., Tang, H., Feng, Y., and Huang, J. (2020). Urban ride-hailing demand prediction with multiple spatio-temporal information fusion network. Transportation Research Part C: Emerging Technologies, 117:102665.

Ke, J., Feng, S., Zhu, Z., Yang, H., and Ye, J. (2021). Joint predictions of multi-modal ride-hailing demands: A deep multi-task multi-graph learning-based approach. Transportation Research Part C: Emerging Technologies, 127:103063.

Ke, J., Yang, H., Zheng, H., Chen, X., Jia, Y., Gong, P., and Ye, J. (2018). Hexagon-based convolutional neural network for supply-demand forecasting of ride-sourcing services. IEEE
Kim, T., Sharda, S., Zhou, X., and Pendyala, R. M. (2020). A stepwise interpretable machine learning framework using linear regression (lr) and long short-term memory (lstm): City-wide demand-side prediction of yellow taxi and for-hire vehicle (fhv) service. *Transportation Research Part C: Emerging Technologies*, 120:102786.

Li, C., Bai, L., Liu, W., Yao, L., and Waller, S. T. (2020a). Graph neural network for robust public transit demand prediction. *IEEE Transactions on Intelligent Transportation Systems*, pages 1–13.

Li, C., Bai, L., Liu, W., Yao, L., and Waller, S. T. (2020b). Knowledge adaption for demand prediction based on multi-task memory neural network. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 715–724.

Li, C., Bai, L., Liu, W., Yao, L., and Waller, S. T. (2021a). A multi-task memory network with knowledge adaptation for multimodal demand forecasting. *Transportation Research Part C: Emerging Technologies*, 131:103352.

Li, J., Guo, F., Sivakumar, A., Dong, Y., and Krishnan, R. (2021b). Transferability improvement in short-term traffic prediction using stacked lstm network. *Transportation Research Part C: Emerging Technologies*, 124:102977.

Li, J., Zhang, K., Shen, L., Wang, Z., Guo, F., Angeloudis, P., Chen, X. M., and Hu, S. (2021c). A domain adaptation framework for short-term traffic prediction. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pages 3564–3569. IEEE.

Li, Y., Zhu, Z., Kong, D., Xu, M., and Zhao, Y. (2019). Learning heterogeneous spatial-temporal representation for bike-sharing demand prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 1004–1011.

Lippi, M., Bertini, M., and Frasconi, P. (2013). Short-term traffic flow forecasting: An experimental comparison of time-series analysis and supervised learning. *IEEE Transactions on Intelligent Transportation Systems*, 14(2):871–882.

Liu, L., Qiu, Z., Li, G., Wang, Q., Ouyang, W., and Lin, L. (2019). Contextualized spatial-temporal network for taxi origin-destination demand prediction. *IEEE Transactions on Intelligent Transportation Systems*, 20(10):3875–3887.

Lv, Y., Duan, Y., Kang, W., Li, Z., and Wang, F. (2014). Traffic flow prediction with big data: a deep learning approach. *IEEE Transactions on Intelligent Transportation Systems*, 16(2):865–873.

Ma, W. and Qian, Z. S. (2018). Estimating multi-year 24/7 origin-destination demand using high-granular multi-source traffic data. *Transportation Research Part C: Emerging Technologies*, 96:96–121.

Ma, X., Zhang, J., Du, B., Ding, C., and Sun, L. (2018). Parallel architecture of convolutional bi-directional lstm neural networks for network-wide metro ridership prediction. *IEEE Transactions on Intelligent Transportation Systems*, 20(6):2278–2288.

Moreira-Matias, L., Gama, J., Ferreira, M., Mendes-Moreira, J., and Damas, L. (2013). Predicting taxi–passenger demand using streaming data. *IEEE Transactions on Intelligent Transportation Systems*, 14(3):1393–1402.

Pan, S. J. and Yang, Q. (2009). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359.

Peng, Y., Liang, T., Hao, X., Chen, Y., Li, S., and Yi, Y. (2021). Cnn-gru-am for shared bicycles demand forecasting. *Computational Intelligence and Neuroscience*, 2021.

Qiu, Y., Song, D., Cheng, H., Cheng, W., Jiang, G., and Cottrell, G. W. (2017). A dual-stage attention-based recurrent neural network for time series prediction. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 2627–2633. AAAI Press.

Tan, M. and Le, Q. (2019). Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pages 6105–6114.

Tang, J., Liang, J., Liu, F., Hao, J., and Wang, Y. (2021). Multi-community passenger demand prediction at region level based on spatio-temporal graph convolutional network. *Transportation Research Part C: Emerging Technologies*, 124:102951.
Toman, P., Zhang, J., Ravishanker, N., and Konduri, K. C. (2020). Dynamic predictive models for ridesourcing services in new york city using daily compositional data. *Transportation Research Part C: Emerging Technologies*, 121:102833.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Wang, L., Geng, X., Ma, X., Liu, F., and Yang, Q. (2019). Cross-city transfer learning for deep spatio-temporal prediction. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 1893–1899.

Xingjian, S., Chen, Z., Wang, H., Yeung, D. Y., Wong, W. K., and Woo, W. C. (2015). Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *Advances in Neural Information Processing Systems*, pages 802–810.

Xu, J., Rahmatizadeh, R., Bölöni, L., and Turgut, D. (2017). Real-time prediction of taxi demand using recurrent neural networks. *IEEE Transactions on Intelligent Transportation Systems*, 19(8):2572–2581.

Xue, R., Sun, D. J., and Chen, S. (2015). Short-term bus passenger demand prediction based on time series model and interactive multiple model approach. *Discrete Dynamics in Nature and Society*, 2015.

Yadav, P., Nimishakavi, M., Yadati, N., Vashisht, S., Rajkumar, A., and Talukdar, P. (2019). Lovasz convolutional networks. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1978–1987.

Yao, H., Liu, Y., Wei, Y., Tang, X., and Li, Z. (2019). Learning from multiple cities: A meta-learning approach for spatial-temporal prediction. In *The World Wide Web Conference*, pages 2181–2191.

Yao, Z., Wang, Y., Long, M., and Wang, J. (2020). Unsupervised transfer learning for spatiotemporal predictive networks. In *International Conference on Machine Learning*, pages 10778–10788.

Ye, J., Sun, L., Du, B., Fu, Y., Tong, X., and Xiong, H. (2019). Co-prediction of multiple transportation demands based on deep spatio-temporal neural network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 305–313.

Zhang, J., Che, H., Chen, F., Ma, W., and He, Z. (2021). Short-term origin-destination demand prediction in urban rail transit systems: A channel-wise attentive split-convolutional neural network method. *Transportation Research Part C: Emerging Technologies*, 124:102928.

Zhang, Z., Li, M., Liu, X., Wang, Y., and He, F. (2019). Multistep speed prediction on traffic networks: A deep learning approach considering spatio-temporal dependencies. *Transportation Research Part C: Emerging Technologies*, 105:297–322.

Zhou, F., Li, L., Zhang, K., and Trajcevski, G. (2021). Urban flow prediction with spatial–temporal neural odes. *Transportation Research Part C: Emerging Technologies*, 124:102912.