Transportation Science

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Real-Time Forecasting of Metro Origin-Destination Matrices with High-Order Weighted Dynamic Mode Decomposition

Zhanhong Cheng, Martin Trépanier, Lijun Sun

Published in Transportation Science on February 16, 2022 as DOI: 10.1287/trsc.2022.1128.
This article has not been copyedited or formatted. The final version may differ from this version.

http://pubsonline.informs.org/journal/trsc

TRANSPORTATION SCIENCE
Vol. 56, No. 4, July–August 2022, pp. 904–918
ISSN 0041-1655 (print), ISSN 1526-5447 (online)

Real-Time Forecasting of Metro Origin-Destination Matrices with High-Order Weighted Dynamic Mode Decomposition

Zhanhong Cheng,[a,b] **Martin Trépanier,**[b,c] **Lijun Sun**[a,b,*]

[a] Department of Civil Engineering, McGill University, Montreal, Quebec H3A 0C3, Canada; [b] Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT), Montreal, Quebec H2S 3H1, Canada; [c] Department of Mathematics and Industrial Engineering, Polytechnique Montreal, Montreal, Quebec H3T 1J4, Canada
*Corresponding author

Contact: zhanhong.cheng@mail.mcgill.ca, https://orcid.org/0000-0003-4770-4702 (ZC); mtrepanier@polymtl.ca, https://orcid.org/0000-0001-8408-8035 (MT); lijun.sun@mcgill.ca, https://orcid.org/0000-0001-9488-0712 (LS)

Received: May 3, 2021
Revised: October 4, 2021
Accepted: January 8, 2022
Published Online in Articles in Advance: February 16, 2022

https://doi.org/10.1287/trsc.2022.1128

Abstract. Forecasting short-term ridership of different origin-destination pairs (i.e., OD matrix) is crucial to the real-time operation of a metro system. However, this problem is notoriously difficult due to the large-scale, high-dimensional, noisy, and highly skewed nature of OD matrices. In this paper, we address the short-term OD matrix forecasting problem by estimating a low-rank high-order vector autoregression (VAR) model. We reconstruct this problem as a data-driven reduced-order regression model and estimate it using dynamic mode decomposition (DMD). The VAR coefficients estimated by DMD are the best-fit (in terms of Frobenius norm) linear operator for the rank-reduced full-size data. To address the practical issue that metro OD matrices cannot be observed in real time, we use the boarding demand to replace the unavailable OD matrices. Moreover, we consider the time-evolving feature of metro systems and improve the forecast by exponentially reducing the weights for historical data. A tailored online update algorithm is then developed for the high-order weighted DMD model (HW-DMD) to update the model coefficients at a daily level, without storing historical data or retraining. Experiments on data from two large-scale metro systems show that the proposed HW-DMD is robust to noisy and sparse data, and significantly outperforms baseline models in forecasting both OD matrices and boarding flow. The online update algorithm also shows consistent accuracy over a long time, allowing us to maintain an HW-DMD model at much low costs.

Funding: This work was supported by the Natural Sciences and Engineering Research Council of Canada, Mitacs, exo.quebec (https://exo.quebec/en), and the Canada Foundation for Innovation.

Keywords: origin-destination matrices • ridership forecasting • dynamic mode decomposition • public transport systems • high-dimensional time series • time-evolving system

1. Introduction

The metro is a green and efficient travel mode that plays an ever-important role in urban transportation. An accurate real-time ridership/demand forecast is crucial to the efficiency and reliability of metro systems. With the wide application of smart card systems and diverse types of sensors, forecasting real-time metro ridership has become an emerging research question in recent years. Existing research mainly focuses on forecasting the short-term (e.g., 15 or 30 minutes) boarding or alighting ridership at metro stations, such as Wei and Chen (2012), Sun, Leng, and Guan (2015), Li et al. (2017), Chen et al. (2019), Liu, Liu, and Jia (2019), and Zhang et al. (2021b). In contrast, forecasting the short-term ridership at origin-destination (OD) pairs of a metro system receives little attention. The ridership among all OD pairs of a metro system can be organized into a matrix. For simplicity, an "OD matrix" in this paper refers to such a ridership matrix at a certain (short) time interval.

Forecasting metro OD matrices has much broader applications than the station-level ridership forecast. For example, by assigning OD matrices to a metro network, we can predict and thus regulate the crowdedness of each train. The station-level boarding/alighting flow also can be calculated as the row and column sums of the OD matrix. However, the real-time forecast of metro OD matrices is extremely difficult for the following reasons. (1) The first challenge is the *high dimensionality*. The number of OD pairs of a metro system is the square of the number of stations, often tens of thousands in practice. (2) Short-term OD matrices of a metro system are often *sparse*, and the ridership/flow distribution within an OD matrix is highly *skewed* (see, e.g., Figure 3). (3) Unlike the boarding or alighting flow, a metro system's OD matrices cannot be obtained in real time (*delayed data availability*), because an OD matrix only becomes available after all the trips belonging to the OD matrix have reached their destinations. Lastly, (4) the complex dynamics of a metro system are *time-evolving*; a well-tuned model may have a short "shelf life" and has expensive retrain/retune costs in long-term maintenance. Although a few studies tried to forecast the real-time metro OD matrices by matrix factorization methods (Gong et al. 2018, 2022) or deep learning models (Toqué et al. 2016, Shen et al. 2020, Zhang et al. 2021c), no existing solution overcomes all four challenges.

This paper utilizes dynamic mode decomposition (DMD) (Schmid 2010)—a recent advance in the fluid dynamics community—to address the aforementioned challenges in a real-time metro OD matrix forecasting problem. DMD is a dimensionality reduction technique that extracts dominating dynamics (modes) from a sequence of high-dimensional vectors. The uniqueness of DMD is that it identifies the best-fit (in terms of Frobenius norm) linear operator that advances a high-dimensional vector sequence forward in time (Tu et al. 2014). We extend the original DMD model by a high-order vector autoregression to incorporate long-term temporal correlations. In dealing with the delayed data availability problem, we replace the latest OD matrices, which are unavailable, with snapshots of boarding flow. We also consider the time-evolving dynamics and introduce a forgetting ratio to reduce the weights of past data exponentially. We name the proposed model *high-order weighted dynamic mode decomposition* (HW-DMD). Moreover, we develop a tailored online update algorithm that updates an HW-DMD's coefficients daily without storing historical data or retraining the model, which greatly reduces the model maintenance costs for long-term implementations. Finally, the proposed model is tested on a Guangzhou metro data set with 159 stations and a Hangzhou metro data set with 80 stations. Our experiments show that HW-DMD can excellently handle the sparse, skewed, and noisy OD data and significantly outperforms baseline models in forecasting both the OD matrices and the boarding flow. The online update algorithm also shows consistent accuracy in updating an HW-DMD model over a long period. Although the online HW-DMD model is applied to the metro OD matrix forecasting problem, it can be readily applied to general (high-dimensional) traffic flow forecast problems, such as in recent studies about DMD-based traffic flow forecasting (Avila and Mezić 2020, Yu et al. 2021). The main contributions of this paper are as follows:

- This paper proposes an HW-DMD model that addresses various difficulties of real-time metro OD matrix forecasting. Experiments show that the forecast of HW-DMD is significantly better than existing models.
- The time-evolving dynamics of a transportation system and the maintenance/update of a forecasting model are often ignored in the literature. This paper considers the time-evolving feature of a metro system by reducing the weights for past data and shows improved performance. An online update algorithm is proposed to reduce the long-term maintenance cost of the HW-DMD model in a time-evolving metro system.
- We propose a DMD-based estimation and online update algorithm for large-scale high-order vector autoregression models with external covariates. The DMD-based estimation produces a best-fit linear operator for rank-reduced full-size data and is particularly useful for the forecast of high-dimensional data with low-rank properties.

The remainder of this paper is organized as follows. We review related work on short-term OD matrix forecasting in Section 2. Next, a description of the metro OD matrix forecasting problem is presented in Section 3. Section 4 briefly introduces the DMD algorithm, which serves as the base for the proposed HW-DMD model. Section 5 is the core part of this paper, where the model specification, estimation, and the online update method for HW-DMD are elaborated. In Section 6, we conduct numerical experiments on the two metro data sets. Conclusions and future directions are summarized in Section 7.

2. Related Work

In the literature, only a few studies have explored the real-time OD matrix forecasting problem for a "metro" system. Therefore, we extend the range to OD demand forecasting for general road transportation modes, such as the ride-hailing system and the highway tolling system. Note that, for a ride-hailing system, the origins and destinations are often defined as zones on a grid.

Matrix/tensor factorization is an effective method to tackle the high-dimensionality problem of OD matrix forecasting. For example, Ren and Xie (2017) applied canonical polyadic (CP) decomposition to an *origin* × *destination* × *vehicle_type* × *time* tensor from highway tolling data. Time series models were then built on the latent temporal matrix to forecast OD matrices. Dai, Sun, and Xu (2018) and Liu et al. (2020) used principal component analysis (PCA) to reduce the dimensionality of OD data and applied several machine learning models to the reduced data for OD flow forecasting. Gong et al. (2022) developed a matrix factorization model to forecast the OD matrices of a metro system. Their work highlights a solution to the delayed data availability problem and various spatial, and temporal regularization techniques are introduced to improve the forecast. In summary, the matrix/tensor factorization-based OD matrix forecasting consists of two components: (1) a dimensionality reduction by factorization and (2) a forecasting model applied to the reduced data.

Deep learning is another mainstream method for OD matrix forecasting. In an early study, Toqué et al. (2016) used long short-term memory (LSTM) networks to forecast the OD matrices of a transit network. They only applied the model to selected high-flow OD pairs because of the high dimensionality and sparsity problems. Convolutional neural networks (CNN) and graph convolutional networks (GNN) are two deep-learning models that greatly reduce the model size compared with a fully connected neural network. Recently, using CNN/GCN to capture spatial correlations and LSTM to capture temporal correlations started to become the "standard configuration" for deep-learning-based OD matrix forecasting. For example, Chu, Lam, and Li (2020) used multiscale convolutional LSTM to forecast the real-time taxi OD demand, and Wang et al. (2019b, 2020) used multitask learning to improve the OD flow forecast of GCN and LSTM networks. A large body of literature focused on better utilizing the spatial/semantic correlations by optimizing the GNN structure or incorporating side information, such as the local spatial context used by Liu et al. (2019), the spatio-temporal encoder-decoder residual multigraph convolutional network (ST-ED-RMGC) proposed by Ke et al. (2021), and the dynamic node-edge attention network (DNEAT) developed by Zhang et al. (2021a). Some studies combined deep-learning models with other models to complement each other. In this direction, Xiong et al. (2020) combined GCN with a Kalman filter to forecast the OD matrices of a turnpike network. Shen et al. (2020) mixed CNN with a gravity model to forecast OD matrices of a metro system. Hu et al. (2020) considered the travel time between OD pairs as a stochastic variable and developed a stochastic OD matrix forecasting model based on tensor factorization and GCN. Noursalehi, Koutsopoulos, and Zhao (2021) used discrete wavelet transform to decompose OD matrices into frequency domain; the outputs were fed into CNN and convolutional-LSTM networks for forecasting.

The performances of deep-learning models are often impaired by the noise in sparse metro OD matrices. To reduce the impact of the noise, Zhang et al. (2019b, 2021c) developed a metric called *OD attraction degree* (ODAD) to mask insignificant OD pairs. Zhang et al. (2019b) showed that masking near-zero OD pairs improves the forecasting accuracy of an LSTM. Based on ODAD, Zhang et al. (2021c) developed a channel-wise attentive split-CNN (CAS-CNN) model for metro OD matrix forecasting. Another merit of this work is that they considered the delayed data availability problem.

In summary, matrix/tensor factorization, CNN, and GCN all aim to reduce model size while maintaining spatial/temporal correlations/dependencies. The HW-DMD model proposed in this paper belongs to the matrix factorization category. Although some ride-hailing systems may not have the delayed data availability problem, most research essentially omitted this problem for simplicity. Particularly, RNN-based deep-learning models can barely work without the most recent OD matrices as inputs. In dealing with the delayed data availability problem, existing solutions (Xiong et al. 2020, Zhang et al. 2021c, Gong et al. 2022) used alternative quantities (e.g., boarding ridership, link flow) to compensate for the unavailable OD information. We also adopt this approach in the proposed model.

3. Problem Description

Many modern metro systems record passengers' entry and exit information using smart cards. We thus know the origin and destination stations, the start and end time for every trip in such a system. Given a fixed time interval (30 minutes in this paper), we denote by $o_{t,i,j}$ the number of trips that depart from station i at the tth interval to station j. We call $o_{t,i,j}$ an *OD flow*. Next, we can describe the number of trips between every OD pair in the system at the tth time interval by an *OD matrix*

$$O_t = \begin{bmatrix} o_{t,1,1} & \cdots & o_{t,1,s} \\ \vdots & \ddots & \vdots \\ o_{t,s,1} & \cdots & o_{t,s,s} \end{bmatrix} \in \mathbb{R}^{s\times s},$$

where s is the number of metro stations. The diagonal elements of metro OD matrices are always zero. We keep these zero elements because they have a negligible effect on the forecast. In our model, OD matrices are organized in a vector form,

$$\mathbf{f}_t = \mathrm{vec}(O_t) = [o_{t,1,1},\ldots,o_{t,s,1},o_{t,1,2},\ldots,o_{t,s,2},\ldots,o_{t,1,s},\ldots,o_{t,s,s}]^\top \in \mathbb{R}^n,$$

where $n = s\times s$ is the number of OD pairs. For convenience, we refer to $\mathbf{f}_t$ as an *OD snapshot*.

Note that OD snapshots are aggregated by the time when passengers enter the system; the exit time might be in a different time interval. Therefore, the true OD snapshot for interval t can only be obtained after all those passengers entered at interval t have reached their destinations; it cannot be observed in real time (i.e., the delayed data availability). In other words, we often do not have access to $\mathbf{f}_t$ when forecasting $\mathbf{f}_{t+1}$. In contrast, the boarding (entering) flow—another important quantity—is observable in real time. We denote by $b_{t,i}$ the number of passengers entering station i at interval t. In fact, we have $b_{t,i} = \sum_j o_{t,i,j}$. We define a *boarding snapshot* as a vector $\mathbf{b}_t = [b_{t,1}, b_{t,2},\ldots,b_{t,s}]^\top$.

The OD matrices/flow forecasting problem is to forecast future OD snapshots $\mathbf{f}_{t+1},\mathbf{f}_{t+2},\ldots,\mathbf{f}_{t+L}$ given a sequence of available historical OD snapshots $\mathbf{f}_1,\mathbf{f}_2,\ldots,\mathbf{f}_t$ and boarding snapshots $\mathbf{b}_1,\mathbf{b}_2,\ldots,\mathbf{b}_t$. The reason for using boarding snapshots is to compensate for the delayed data availability problem of recent OD snapshots.

4. Dynamic Mode Decomposition

Dynamic mode decomposition (DMD; Schmid 2010) was developed by the fluid dynamics community to extract dynamic features from high-dimensional data. To better illustrate our forecasting model, we briefly introduce DMD in this section.

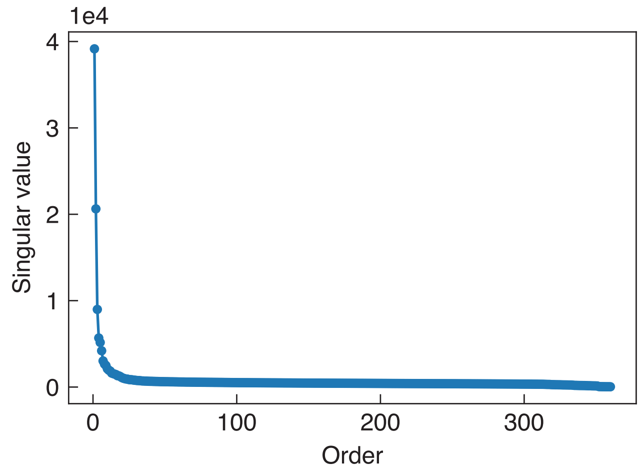
Consider using a linear dynamical system $\mathbf{f}_i \approx A\mathbf{f}_{i-1}$ for OD flow forecasting. Similar to many fluid problems, n is huge for an OD snapshot, and even storing $A \in \mathbb{R}^{n\times n}$ can be prohibitive. Therefore, DMD outputs the (leading) eigenvalues and eigenvectors of A without calculating the expensive A. The eigenvectors of A are referred to as the *DMD modes* and have clear physical meaning. Each DMD mode is associated with an oscillation frequency and a decay/growth rate determined by its eigenvalue. DMD is also connected to Koopman theory and can model complex nonlinear systems by constructing proper measurements (Rowley et al. 2009). There are many variant algorithms for DMD. We only present the *exact DMD* proposed by Tu et al. (2014), which is closely related to this paper.

We arrange OD snapshots into m-column matrices $Y_i = [\mathbf{f}_{i-m+1}, \mathbf{f}_{i-m+2}, \ldots, \mathbf{f}_i] \in \mathbb{R}^{n\times m}$. Typically, $m \ll n$. The linear dynamical system follows $Y_t \approx AY_{t-1}$. The exact DMD seeks the leading eigenvalues and eigenvectors of the best-fit linear operator A by the following procedure.

1. Compute the truncated singular value decomposition (SVD) of $Y_{t-1} \approx U\Sigma V^\top$, where $U \in \mathbb{R}^{n\times r}$, $\Sigma \in \mathbb{R}^{r\times r}$, and $V \in \mathbb{R}^{m\times r}$ and $r \ll m$.
2. Instead of computing the full matrix $A = Y_t Y_{t-1}^+ \approx Y_t V\Sigma^{-1}U^\top$.[1] We define a reduced matrix $\tilde{A} = U^\top AU \approx U^\top Y_t V\Sigma^{-1} \in \mathbb{R}^{r\times r}$. It can be proved that $\tilde{A}$ and A have the same nonzero leading eigenvalues (Tu et al. 2014).
3. Compute the eigenvalue decomposition $\tilde{A}W = W\Lambda$. The entries of the diagonal matrix Λ are also the eigenvalues of the full matrix A.
4. The DMD modes (eigenvectors of A) can be obtained by $\Phi = Y_t V\Sigma^{-1}W$.

Figure 1 shows the singular values of a 10-day-long Y_{t-1} from the Guangzhou metro system. A few leading singular values explain a significant portion of the variance, confirming the low-rank feature of OD snapshot data. The DMD-based model can thus greatly reduce the dimensionality/complexity of such a dynamic system. However, the exact DMD has some limitations for the OD flow forecasting problem. First, the complex temporal correlation of OD flow cannot be well captured by a linear dynamical system. Moreover, using the last OD snapshot is impractical since OD snapshots cannot be observed in real time. To address these problems, we propose our solution in the next section.

Figure 1. (Color online) The Singular Values of a 10-Day-Long Y_{t-1} Collected from the Guangzhou Metro Smart Card System



5. High-Order Weighted Dynamic Mode Decomposition

5.1. Model Specification

The forecasting formula of an exact DMD amounts to a high-dimensional vector autoregression of order 1. However, the latest OD snapshots are unknown at the time of forecasting. Therefore, we use the two latest snapshots of the boarding flow as a replacement. We regard OD snapshots of three or more intervals ago as available, because we find that more than 96% trips in our data set are completed within one hour (two lags). And we can use a high-order vector autoregression to capture the long-term correlations in OD snapshots. The forecasting model follows

$$\mathbf{f}_i \approx A_{t,1}\mathbf{f}_{i-q_1} + A_{t,2}\mathbf{f}_{i-q_2} + \ldots + A_{t,h}\mathbf{f}_{i-q_h} + A_{t,b1}\mathbf{b}_{i-1} + A_{t,b2}\mathbf{b}_{i-2} \quad \forall\, i \in \{q_h+1, q_h+2, \ldots, t\}, \tag{1}$$

where time lags for OD snapshots are positive integers satisfying $3 \le q_1 < \ldots < q_h < t$. Note that coefficient matrices $A_{t,1}, \ldots, A_{t,h} \in \mathbb{R}^{n\times n}$ and $A_{t,b1}, A_{t,b2} \in \mathbb{R}^{n\times s}$ are estimated using the data up to the latest (tth) time interval; they are reestimated when new data become available. This allows model coefficients to be time-varying. We will introduce how to update coefficient matrices using new observations without storing historical data in Section 5.3.

To express Equation (1) in a concise matrix form, let $Y_i = [\mathbf{f}_{i-m+1}, \mathbf{f}_{i-m+2}, \ldots, \mathbf{f}_i]$ and $B_i = [\mathbf{b}_{i-m+1}, \mathbf{b}_{i-m+2}, \ldots, \mathbf{b}_i]$, where $m = t - q_h$ is the number of target snapshots. Then, Equation (1) is equivalent to

$$Y_t \approx A_{t,1}Y_{t-q_1} + A_{t,2}Y_{t-q_2} + \ldots + A_{t,h}Y_{t-q_h} + A_{t,b1}B_{t-1} + A_{t,b2}B_{t-2} \tag{2}$$

$$= [A_{t,1}, A_{t,2}, \ldots A_{t,h}, A_{t,b1}, A_{t,b2}]\begin{bmatrix} Y_{t-q_1} \\ Y_{t-q_2} \\ \vdots \\ Y_{t-q_h} \\ B_{t-1} \\ B_{t-2} \end{bmatrix} \tag{3}$$

$$= G_t X_t, \tag{4}$$

where $G_t \in \mathbb{R}^{n\times(hn+2s)}$ and $X_t \in \mathbb{R}^{(hn+2s)\times m}$ are augmented matrices for coefficients and data, respectively. Note that, with this approach, we model forecasting as a regression problem without considering the intersequence dependence.

We next introduce a forgetting ratio ρ $(0 < \rho \le 1)$ that assigns small weights on snapshots to past days. This is because the dynamics of the system may change over time and we prefer to use the most recent dynamics to achieve accurate forecasting. The matrix G_t can be solved by the following optimization problem,

$$\min_{G_t} \sum_{i=1}^{m} \rho^{\text{day}(m)-\text{day}(i)} \|\mathbf{y}_i - G_t \mathbf{x}_i\|_F^2, \tag{5}$$

where $\mathbf{y}_i$ and $\mathbf{x}_i$ are the ith column of Y_t and X_t, respectively; day(i) represents the day of the snapshot $\mathbf{y}_i$. We assign the same weight for snapshots of the same day. The weight $\rho^{\text{day}(m)-\text{day}(m)} = \rho^0 = 1$ for the latest OD snapshot. For a snapshot in j days ago, the weight is ρ^j, which decreases exponentially. This weighting idea is similar to works by Alfatlawi and Srivastava (2020), Zhang et al. (2019a), and Kwak and Geroliminis (2021). For convenience, we define $\sigma = \sqrt{\rho}$ and the weighted version of Y_t and X_t as

$$Y_t^w = [\sigma^{\text{day}(m)-\text{day}(1)}\mathbf{y}_1, \sigma^{\text{day}(m)-\text{day}(2)}\mathbf{y}_2, \ldots, \mathbf{y}_m],$$
$$X_t^w = [\sigma^{\text{day}(m)-\text{day}(1)}\mathbf{x}_1, \sigma^{\text{day}(m)-\text{day}(2)}\mathbf{x}_2, \ldots, \mathbf{x}_m].$$

Then, the optimization problem in Equation (5) becomes an ordinary least squares problem:

$$\min_{G_t} \|Y_t^w - G_t X_t^w\|_F^2. \tag{6}$$

Figure (2) summarizes the overall structure of the proposed higher-order weighted DMD (HW-DMD) framework. The underlying forecasting model is a high-order vector autoregression with the boarding flow as extra inputs. A forgetting ratio is introduced to decrease the weights of past data exponentially on a daily basis. In Section 5.2, we will introduce a dimensionality reduction technique based on DMD to find a low-rank solution for this large model (with respect to number of parameters). Instead of full matrices $A_{t,(\cdot)}$, we seek $\tilde{A}_{t,(\cdot)}$—much smaller matrices—to capture the system's dynamic. Finally, an online update method is proposed in Section 5.3 to update the model coefficients incrementally without storing historical data. This provides a memory-saving solution that maintains an up-to-date model. Note that the same model framework can be easily extended to incorporate higher-order boarding flow or other external covariates (e.g., days of the week, alighting flow, holidays). For example, we can represent days of the week by one-hot encoding $\mathbf{w}_i \in \mathbb{R}^{7\times1}$ and add an additional regression term $A_{t,w}\mathbf{w}_i$ to Equation (1) to incorporate the weekly pattern. This paper only presents the model specified in Equation (1) for illustration.

5.2. Model Estimation

We prefer a low-rank approximation of G_t over a full matrix of the optimal solution of Equation (6). This is because storing the large full matrix is prohibitive, and the optimal solution often leads to overfitting problems, especially for the sparse and noisy OD data. Luckily, we can find a pretty good approximation thanks to the inherent low-rank nature of OD data.

Similar to the exact DMD, we first compute the truncated SVD on the weighted augmented data matrix $X_t^w \approx U_X \Sigma_X V_X^\top$, where we keep the r_X $(r_X \ll m)$ largest singular values and $U_X \in \mathbb{R}^{(hn+2s)\times r_X}$, $\Sigma_X \in \mathbb{R}^{r_X\times r_X}$, $V_X \in \mathbb{R}^{m\times r_X}$. As shown in Figure 1, a few leading singular values can well capture the entire data. Therefore, an approximation for coefficient matrices is

$$G_t = Y_t^w X_t^{w+} \approx Y_t^w V_X \Sigma_X^{-1} U_X^\top, \tag{7}$$

$$[A_{t,1}, \ldots A_{t,h}, A_{t,b1}, A_{t,b2}] \approx [Y_t^w V_X \Sigma_X^{-1} U_{X,1}^\top, \ldots, Y_t^w V_X \Sigma_X^{-1} U_{X,h}^\top, Y_t^w V_X \Sigma_X^{-1} U_{X,b1}^\top, Y_t^w V_X \Sigma_X^{-1} U_{X,b2}^\top], \tag{8}$$

where $U_X^\top = [U_{X,1}^\top, \ldots, U_{X,h}^\top, U_{X,b1}^\top, U_{X,b2}^\top]$, $U_{X,1}, \ldots, U_{X,h} \in \mathbb{R}^{n\times r_X}$, and $U_{X,b1}, U_{X,b2} \in \mathbb{R}^{s\times r_X}$. This step uses the result from a truncated SVD to replace the original X_t^w, which reduces the impact of the noise in the data.

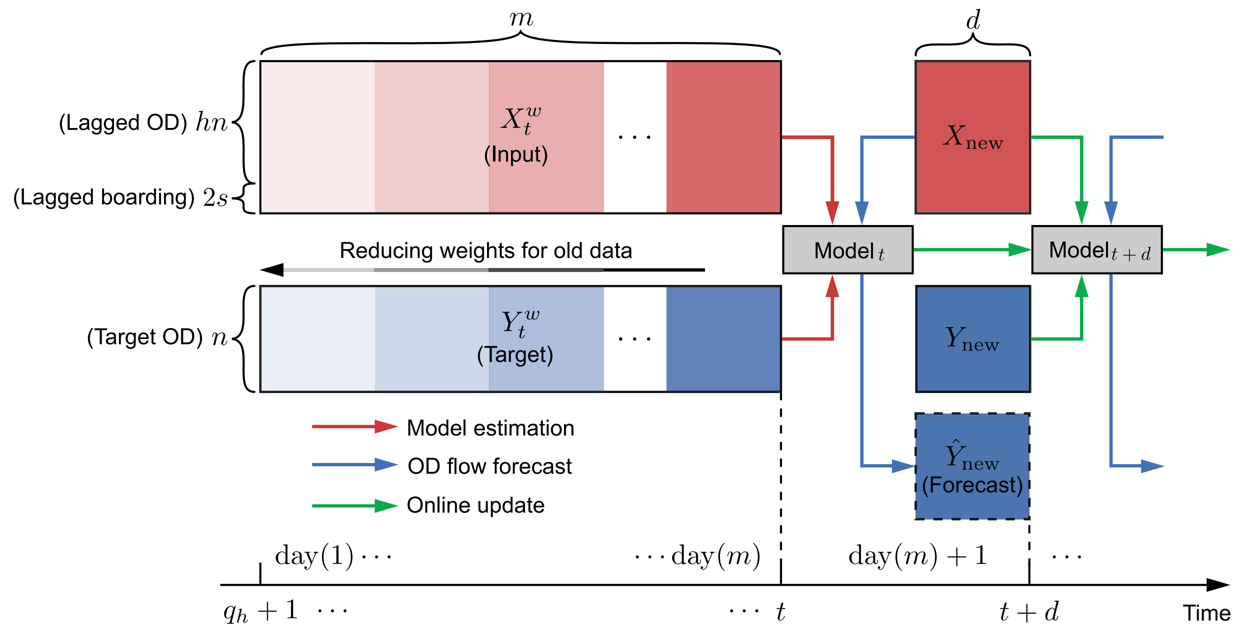
The results computed from Equations (7) and (8) are still prohibitive. Therefore, for each column in Y_t^w, we seek a transformation $\mathbf{y}_i^w \to \tilde{\mathbf{y}}_i^w$ such that $\tilde{\mathbf{y}}_i^w \in \mathbb{R}^{r_Y}$ with $r_Y \ll n$. In doing so, we compute another rank-r_Y truncated SVD of the target matrix $Y_t^w \approx U_Y \Sigma_Y V_Y^\top$. The columns of U_Y form an orthonormal basis; thus, the transformation $\tilde{\mathbf{y}}_i^w = U_Y^\top \mathbf{y}_i^w$ computes the coordinates of $\mathbf{y}_i^w$ on this basis, which compresses y_i^w from $\mathbb{R}^n$ to $\mathbb{R}^{r_Y}$. We can project coefficient matrices onto the same basis U_Y to greatly reduce the dimensionality:

$$\tilde{A}_{t,i} = U_Y^\top A_{t,i} U_Y \approx U_Y^\top Y_t^w V_X \Sigma_X^{-1} U_{X,i}^\top U_Y, \quad \forall i \in \{1, 2, \ldots, h\}, \tag{9}$$

$$\tilde{A}_{t,bj} = U_Y^\top A_{t,bj} \approx U_Y^\top Y_t^w V_X \Sigma_X^{-1} U_{X,bj}^\top, \quad \forall j \in \{1, 2\}, \tag{10}$$

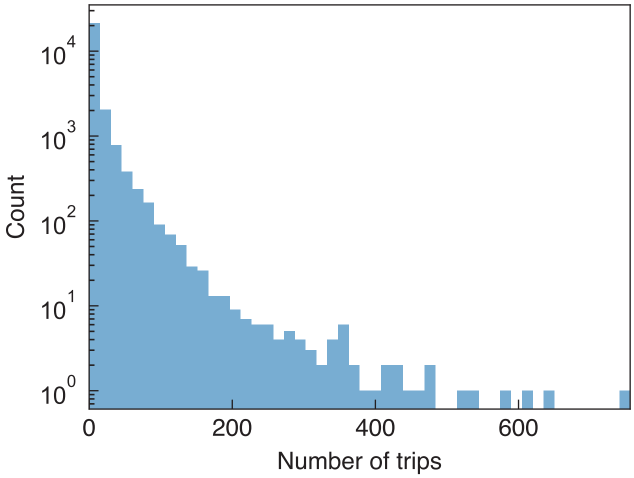
Published in Transportation Science on February 16, 2022 as DOI: 10.1287/trsc.2022.1128.
This article has not been copyedited or formatted. The final version may differ from this version.

Figure 2. (Color online) Model Framework for HW-DMD



Notes. Model input X contains hn rows for lagged OD snapshots and $2s$ rows for lagged boarding snapshots. Columns in Y and $\hat{Y}$ are, respectively, real and forecasted snapshots for OD flow. Model coefficients are estimated by weighted historical data (X_t^w and Y_t^w) and updated daily whenever new data come.

Figure 3. (Color online) The Histogram of $o_{i,j}$ in an OD Snapshot of a Morning Peak in Guangzhou



where $\tilde{A}_{t,i} \in \mathbb{R}^{r_Y \times r_Y}$ and $\tilde{A}_{t,bj} \in \mathbb{R}^{r_Y \times s}$. Finally, we can write the model of Equation (2) in the reduced-order subspace

$$\tilde{Y}_t \approx \tilde{A}_{t,1}\tilde{Y}_{t-q_1} + \tilde{A}_{t,2}\tilde{Y}_{t-q_2} + \ldots + \tilde{A}_{t,h}\tilde{Y}_{t-q_h} + \tilde{A}_{t,b1}B_{t-1} + \tilde{A}_{t,b1}B_{t-2},$$

where $\tilde{Y}_i = U_Y^\top Y_i$. The final forecast of an OD snapshot $\hat{\mathbf{y}}_i$ can be calculated by transforming back to the original basis by $\hat{\mathbf{y}}_i = U_Y \tilde{\mathbf{y}}_i$. With the reduced coefficient matrices $\tilde{A}_{t,(\cdot)}$ and projection bases U_Y, we avoid calculating and storing the giant coefficient matrices $A_{t,(\cdot)}$.

DMD-based estimation is different from common dimensionality reduction techniques in several ways. For many matrix-factorization-based models and dynamic factor models, a forecasting model is estimated after performing dimensionality reduction (e.g., Ren and Xie 2017), or latent factors are constructed by keeping the most temporal dynamics (e.g., Forni et al. 2000; Lam, Yao, and Bathia 2011; Yu, Rao, and Dhillon 2016); the forecast ability is designed on the latent (size-reduced) data for these models. In contrast, DMD-based methods first estimate a forecasting model by a least-square fit of rank-reduced full-size data (i.e., Equation (7)) and then reduce the dimensionality of the linear operator by projecting to leading SVD modes (i.e., Equations (9)–(10)); the resulting linear operator captures the dynamics of the rank-reduced full-sized data. Although the forecast value $\hat{\mathbf{y}}_i$ by an HW-DMD is restricted on the column space of U_Y, it is already the best approximation in $\mathbb{R}^{r_Y}$ (in terms of Frobenius norm; Eckart and Young 1936) because the basis is determined by leading singular vectors. Besides, the rank truncation for the data also eases the noise and the overfitting problem. As noted by Schmid (2010), accurate identification of more than the first couple modes can be difficult on noisy data sets without this truncation step.

The major computational cost in parameter estimation of HW-DMD is the SVD part. Current numerical software can solve a large-scale SVD very efficiently.

Therefore, estimating the HW-DMD model is very fast. We can further derive the eigenvalues and eigenvectors of coefficient matrices $A_{t,i}$ (Proctor, Brunton, and Kutz 2016). But this step is not necessary for our task, since they are not used to generate the forecast and there is no clear physical meaning for eigenvectors in a high-order vector autoregression.

5.3. Online Update

A model trained by dated data may not reflect the recent dynamic in a system. Instead of retraining using entire data, we develop an online algorithm that updates HW-DMD day by day with new observations without storing historical data, as shown in Figure 2. Similar algorithms for online DMD have been developed by Hemati, Williams, and Rowley (2014), Zhang et al. (2019a), and Alfatlawi and Srivastava (2020). We extend the online DMD update algorithm to a high-order weighted version.

To illustrate the update algorithm, we need to reorganize Equations (7)–(10). Let $\tilde{X}_i^w = U_X^\top X_i^w$ and $\tilde{Y}_i^w = U_Y^\top Y_i^w$ be the projection of data to the coordinates of U_X and U_Y, respectively. Using the fact that $(U_X\tilde{X}_t^w)^+ = V_X\Sigma_X^{-1}U_X^\top$, we can rewrite Equation (7) as

$$\begin{aligned} G_t &\approx Y_t^w(U_X\tilde{X}_t^w)^+ \\ &= Y_t^w\tilde{X}_t^{w\top}\left(\tilde{X}_t^w\tilde{X}_t^{w\top}\right)^+U_X^\top. \end{aligned}$$

Therefore, Equations (9) and (10) become

$$\tilde{A}_{t,i} \approx \tilde{Y}_t^w\tilde{X}_t^{w\top}\left(\tilde{X}_t^w\tilde{X}_t^{w\top}\right)^+U_{X,i}^\top U_Y = PQ_X^+U_{X,i}^\top U_Y \quad \forall\, i \in \{1,\dots,h\}, \tag{11}$$

$$\tilde{A}_{t,bj} \approx \tilde{Y}_t^w\tilde{X}_t^{w\top}\left(\tilde{X}_t^w\tilde{X}_t^{w\top}\right)^+U_{X,bj}^\top = PQ_X^+U_{X,bj}^\top \quad \forall\, j \in \{1,2\}, \tag{12}$$

where $P = \tilde{Y}_t^w\tilde{X}_t^{w\top} \in \mathbb{R}^{r_Y\times r_X}$ and $Q_X = \tilde{X}_t^w\tilde{X}_t^{w\top} \in \mathbb{R}^{r_X\times r_X}$.

To facilitate the online update, we define an additional matrix $Q_Y = \tilde{Y}_t^w\tilde{Y}_t^{w\top} \in \mathbb{R}^{r_Y\times r_Y}$. After the reorganization, model coefficients are represented by three "core" matrices P, Q_X, Q_Y and two projection matrices U_X, U_Y. Note that these matrices are also time-varying. For simplicity, we omit the t subscript and regard that they are always "up-to-date." Moreover, there are two important properties for the core matrices.

Theorem 1. *Given new observations $Y_{\text{new}} \in \mathbb{R}^{n\times d}$ and $X_{\text{new}} \in \mathbb{R}^{(hn+2s)\times d}$ from a new day, where d is the number of snapshots per day, under the same projection matrices, the new core matrices can be updated by*

$$P \leftarrow \rho P + \tilde{Y}_{\text{new}}\tilde{X}_{\text{new}}^\top, \tag{13}$$

$$Q_X \leftarrow \rho Q_X + \tilde{X}_{\text{new}}\tilde{X}_{\text{new}}^\top, \tag{14}$$

$$Q_Y \leftarrow \rho Q_Y + \tilde{Y}_{\text{new}}\tilde{Y}_{\text{new}}^\top, \tag{15}$$

where $\tilde{X}_{\text{new}} = U_X^\top X_{\text{new}}$ and $\tilde{Y}_{\text{new}} = U_Y^\top Y_{\text{new}}$.

Proof of Theorem 1. Given new observations $Y_{\text{new}} \in \mathbb{R}^{n\times d}$ and $X_{\text{new}} \in \mathbb{R}^{(hn+2s)\times d}$ from a new day, under the same projection matrices, the new core matrix P can be computed by

$$\begin{aligned} \tilde{Y}_{t+d}^w\tilde{X}_{t+d}^{w\top} &= [\sigma\tilde{Y}_t^w, U_Y^\top Y_{\text{new}}][\sigma\tilde{X}_t^w, U_X^\top X_{\text{new}}]^\top \\ &= [\sigma\tilde{Y}_t^w, \tilde{Y}_{\text{new}}][\sigma\tilde{X}_t^w, \tilde{X}_{\text{new}}]^\top \\ &= \sigma^2\tilde{Y}_t^w\tilde{X}_t^{w\top} + \tilde{Y}_{\text{new}}\tilde{X}_{\text{new}}^\top \\ &= \rho P + \tilde{Y}_{\text{new}}\tilde{X}_{\text{new}}^\top. \end{aligned}$$

Therefore, P can be updated by $P \leftarrow \rho P + \tilde{Y}_{\text{new}}\tilde{X}_{\text{new}}^\top$. A similar proof applies to Q_X and Q_Y. □

Theorem 2. *Denote by $\bar{Y}_t^w = U_Y\tilde{Y}_t^w$ the recovered data from the reduced data. If $\mathbf{v}_i$ is the ith eigenvector of Q_Y, then $U_Y\mathbf{v}_i$ is the ith left singular vector of $\bar{Y}_t^w$. The same property applies to Q_X and $\bar{X}_t^w = U_X\tilde{X}_t^w$.*

Proof of Theorem 2. Compute SVD $\bar{Y}_t^w = \bar{U}\bar{\Sigma}\bar{V}^\top$; then

$$\bar{Y}_t^w\bar{Y}_t^{w\top} = \bar{U}\bar{\Sigma}\bar{V}^\top\bar{V}\bar{\Sigma}^\top\bar{U}^\top = \bar{U}(\bar{\Sigma}\bar{\Sigma})\bar{U}^\top, \tag{16}$$

$$\left(\bar{Y}_t^w\bar{Y}_t^{w\top}\right)\bar{U} = \bar{U}(\bar{\Sigma}\bar{\Sigma}) = \bar{U}\bar{\Lambda}. \tag{17}$$

Therefore, columns of $\bar{U}$ are the eigenvectors of $\bar{Y}_t^w\bar{Y}_t^{w\top}$ and the left singular vectors of $\bar{Y}_t^w$. Substitute $\bar{Y}_t^w\bar{Y}_t^{w\top} = U_YQ_YU_Y^\top$ to Equation (17); we have

$$\begin{aligned} (U_YQ_YU_Y^\top)\bar{U} &= \bar{U}\bar{\Lambda}, \\ Q_Y(U_Y^\top\bar{U}) &= (U_Y^\top\bar{U})\bar{\Lambda}. \end{aligned}$$

Define $V = U_Y^\top\bar{U}$. Then, each column $\mathbf{v}_i$ in V is a eigenvector for Q_Y and $U_Y\mathbf{v}_i = U_Y(U_Y^\top\bar{\mathbf{u}}_i) = \mathbf{u}_i$ is a singular vector of $\bar{Y}_t^w$. □

Theorem 1 is used to update the core matrices in a memory-saving way. Theorem 2 indicates that we can used the eigenvectors of Q_Y to approximate the left singular vectors of Y_t^w (because $Y_t^w \approx \bar{Y}_t^w$), which is crucial for updating the projection matrices. Based on these properties, we summarize the online update algorithm in the following three steps.

1. **Expand projection matrices**. Let $E_Y = Y_{\text{new}} - U_YU_Y^\top Y_{\text{new}}$ and $E_X = X_{\text{new}} = U_XU_X^\top X_{\text{new}}$ be the residuals that cannot be represented by the column space of U_X and U_Y. To incorporate these residuals, we expand projection matrices by $U_X \leftarrow [U_X, U_{E_X}]$ and $U_Y \leftarrow [U_Y, U_{E_Y}]$, where U_{E_X} and U_{E_Y} are the orthonormal bases (obtained by SVD or QR factorization) of E_X and E_Y, respectively.
2. **Update core matrices**. To align dimensions, we first pad P, Q_X, and Q_Y with zeros on the dimensions where U_X and U_Y expanded. Then we update core matrices by Equation (13)–(15).
3. **Compression**. The first two steps incorporate all new information at the cost of expanding dimensions. Next, we compress the model based on Theorem 2.

Denote V_X and V_Y as the matrices composed by the leading r_X and r_Y eigenvectors of Q_X and Q_Y, respectively. We can compress projection matrices by $U_X \leftarrow U_X V_X$, $U_Y \leftarrow U_Y V_Y$ to keep the leading singular vectors of $\tilde{X}^w_{t+d}$ and $\tilde{Y}^w_{t+d}$. The core matrices can be compressed accordingly by $Q_X \leftarrow V_X^\top Q_X V_X$, $Q_Y \leftarrow V_Y^\top Q_Y V_Y$, $P \leftarrow V_Y^\top P V_X$.

Besides the daily update, a more general setting can be updating the model for every k intervals or only doing the compression step when r_X or r_Y exceeds a threshold. This paper adopts the aforementioned daily update because metro systems often have a one-day periodicity. In terms of computational efficiency, the online update algorithm computes the SVD for d-column data matrices and eigenvalue decomposition of Q_X and Q_Y. The computation has a constant cost every day and it is significantly faster than retraining all data. In terms of memory efficiency, historical data are not required when updating the model. All we need to store are three "core" matrices and two projection matrices. Regarding the error, the online algorithm does not take into account the previously truncated part. This impact is negligible because the truncated part contains mostly noise, and past data are forgotten exponentially. Our experiments in Section 6.6 show that the online algorithm performs pretty close to or even slightly better than retraining.

5.4. Connections with Other DMD Models

The proposed HW-DMD is closely related to Hankel-DMD (Arbabi and Mezić 2017, Brunton et al. 2017, Avila and Mezić 2020) and DMD with control (DMDc; Proctor, Brunton, and Kutz 2016). Hankel-DMD uses Hankel data matrices as input and output to model a nonlinear dynamical system by a linear model; its DMD modes approximate to Koopman modes. There is another model also named *higher-order DMD* (HODMD; Le Clainche and Vega 2017), which requires Hankelizing data in its estimation and is essentially similar to Hankel-DMD. Instead, the proposed HW-DMD uses raw snapshots as the output (the left-hand side of Equation (2)) without using the Hankel structure. This formula is equivalent to a high-order vector autoregression model, which is neater and more suitable in the context of forecasting. Moreover, our model can use noncontinuous orders and external variables (e.g., the boarding flow). Essentially, the external variables of our model can be regarded as the control term of a DMDc model.

The three-step online update algorithm for HW-DMD in this paper inherits from the work of Hemati, Williams, and Rowley (2014). The original algorithm was developed for the exact DMD introduced in Section 4. Besides, the online DMD proposed by (Zhang et al. 2019a) considers the decaying weight of data, but the constant projection matrix in their assumption restricts the update effect. Alfatlawi and Srivastava (2020) proposed an online algorithm for weighted DMD using incremental SVD, which is a different technique from our method. Our contribution is extending the algorithm proposed by Hemati, Williams, and Rowley (2014) to a high-order weighted version with the consideration of external regression covariates.

6. Experiments

In this section, we compare the proposed HW-DMD with other forecasting models using real-world data. We begin with an introduction to data and experimental settings. Next, we compare model performances by forecasting the OD matrices and the boarding flow derived from the OD matrices. Finally, we examine the long-term effect of the online HW-DMD update algorithm. The code for the experiments is available from https://github.com/mcgill-smart-transport/high-order-weighted-DMD.

6.1. Data and Experimental Settings

We examine HW-DMD using the metro smart card data from two cities, Guangzhou and Hangzhou. Both data sets record the origin, destination, and entry and exit time of each metro trip. We focus on the forecast of workdays and connect each Friday to the next Monday. Details of the two data sets are as follows.

- Guangzhou metro data: This data set covers around 301 million trips among 159 metro stations in Guangzhou from July 1 to September 30, 2017. Guangzhou metro operates from 6:00 to 24:00. We use the first 20 weekdays (July 3 to July 28) as the training set, the following 10 weekdays (July 31 to Aug 11) as the validation set, and the following 10 weekdays (August 14 to August 25) as the test set. There are additional one-month data after the test set; we use these data to study the long-term effect of the online HW-DMD update algorithm.
- Hangzhou metro data[2]: This is an open data set that covers 80 effective stations of Hangzhou metro from January 1 to January 25, 2019. The operation hours are from 5:30 to 23:30. We use the first 10 weekdays (January 1 to January 14) for training, the following four weekdays (January 15 to January 18) for validation, and the remaining five weekdays (January 21 to January 25) for testing.

We aggregate OD snapshots by a 30-minute time interval, which means 36 snapshots per day for both cities. Note that a small interval may result in sparse OD matrices; we choose the 30-minute interval to balance the practical requirements. Figure 3 shows the distribution of $o_{i,j}$ from an OD snapshot of a typical morning peak in Guangzhou. The distribution roughly follows a power law, with most OD pairs having small volumes while a few of them are significantly larger. The highly skewed distribution is very difficult to be properly handled by conventional forecasting models.

The performance of a model is quantified using the root-mean-square error (RMSE), the weighted mean absolute percentage error (WMAPE), and the coefficient of determination (denoted as R^2):

$$\mathrm{RMSE}(\alpha,\hat{\alpha}) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\alpha_i-\hat{\alpha}_i)^2},$$

$$\mathrm{WMAPE}(\alpha,\hat{\alpha}) = \frac{\sum_i^N |\alpha_i-\hat{\alpha}_i|}{\sum_i^N |\alpha_i|}\times 100\%,$$

$$R^2(\alpha,\hat{\alpha}) = 1-\frac{\sum_{i=1}^{N}(\alpha_i-\hat{\alpha}_i)^2}{\sum_{i=1}^{N}(\alpha_i-\bar{\alpha})^2},$$

where α and $\hat{\alpha}$ are, respectively, the real and predicted values; $\bar{\alpha}$ is the average value of α; and N is the total number of elements under different time intervals and locations. The three performance metrics are computed for both OD flow o and boarding flow b (forecasted by $\hat{b}_{t,i} = \sum_j \hat{o}_{t,i,j}$).

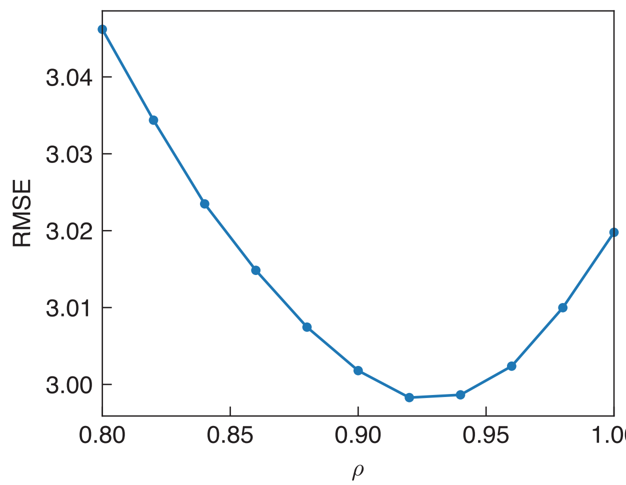
6.2. Hyperparameters

We use the online update algorithm for HW-DMD if not otherwise specified. Hyperparameters for HW-DMD include time lags $q_1,\ldots,q_h$, the SVD truncation rank r_X, r_Y, and the forgetting ratio ρ. These parameters are determined in sequential order.

We use the Guangzhou data set as an example to elaborate the hyperparameter tuning procedure. We first set $r_X = r_Y = 100$ and $\rho = 1$ and select time lags in a greedy manner. For time lags within one day ($3 \le q_i \le 36$), we repeatedly add a "currently best" time lag based on the RMSE of the validation set until a new lag brings no improvement or the number of lags reaches 10. This procedure selects {3, 4, 8, 14, 19, 28, 30, 33, 35, 36} as time lags. The considerable high-order time lags in the result indicate long-term autocorrelations of OD time series. For example, the lag 19 roughly equals a typical work duration (9.5 hours), which can be explained as a strong correlation between the departure trips for commuters in the morning and the returning trips in the afternoon (Cheng, Trépanier, and Sun 2021). The metro OD flow is also highly regular; the largest several lags (e.g., 33, 35, and 36) capture the one-day periodicity. Next, we determine r_X and r_Y by a grid search from 20 to 100 at an interval of 10. The best r_Y is 50. A larger r_X than 100 still brings a marginal improvement, but we truncate r_X at 100 to restrict the model size (r_X affects the size of U_X in the online update). Lastly, we set ρ to be 0.92 based on a line search from 0.8 to 1. As shown in Figure 4, we can see that assigning smaller weights for old data indeed improves the forecast. Because $0.92^8 \approx 0.51$, using $\rho = 0.92$ is roughly equivalent to halving the weight every eight days.

The hyperparameter tuning for the Hangzhou data set follows the same procedure. The selected hyperparameters for the Hangzhou data set are time lags = {3, 4, 6, 14, 18, 19, 28, 32, 35, 36}, $r_Y = 40$, $r_X = 100$, and $\rho = 0.92$.

Figure 4. (Color online) The Effect of ρ to the Forecast OD RMSE in the Validation set of the Guangzhou Data



6.3. Benchmark Models

We compare HW-DMD with the following benchmark models:

- HA: Historical average. For the OD flow at a certain period of the day (e.g., 7:00–7:30), HA uses the average OD flow at that period in the training set as the forecast value.
- TRMF: Temporal regularized matrix factorization (Yu, Rao, and Dhillon 2016). TRMF is a matrix factorization model that imposes autoregression (AR) processes on each temporal factor. We use time lags [1, ..., 10] for the AR processes. We search over {100, 300, 500, 1000, 1500, 2000, 2500, 3000} for the best regularization parameter and search from 30 to 150 with an interval of 10 for the best number of factors.
- ConvLSTM: Convolutional LSTM (Shi et al. 2015). It is a deep recurrent neural network model that forecasts future frames of matrix time series (e.g., videos). Here, we use it to forecast future OD matrices by the most recent 10 OD matrices. Following the work of Zhang et al. (2021c), we apply a three-layer LSTM structure with eight, eight, and one filter, respectively, and set the kernel size to be 3 × 3 for all convolutional layers in the model.
- FNN: A two-layer feedforward neural network. We use the OD snapshots of three to 10 lags ago and the boarding flow snapshot of one to two lags ago as the input features. We perform a grid search over the type of activation functions (linear, sigmoid, and relu) and the number of hidden layers (from 10 to 100 at an interval of 10) for the best model setting.
- SARIMA: Seasonal autoregressive integrated moving average. We only use SARIMA to forecast the

Table 1. Models' Performance for OD Flow Forecasting

		Guangzhou			Hangzhou		
Method	Criterion	One-step	Two-step	Three-step	One-step	Two-step	Three-step
HW-DMD	RMSE	**3.05**	**3.09**	**3.11**	**3.36**	**3.41**	**3.44**
$\rho = 0.92$	WMAPE	**29.65%**	**29.77%**	**29.79%**	**31.76%**	**31.96%**	**31.84%**
	R^2	**0.957**	**0.956**	**0.955**	**0.934**	**0.932**	**0.931**
HW-DMD	RMSE	3.08	3.12	3.14	3.40	3.45	3.48
$\rho = 1$	WMAPE	29.71%	29.87%	29.91%	31.94%	32.22%	32.13%
	R^2	0.956	0.955	0.954	0.933	0.930	0.929
TRMF	RMSE	3.22	3.24	3.26	3.80	3.89	3.96
	WMAPE	30.61%	30.72%	30.79%	34.02%	34.48%	34.82%
	R^2	0.952	0.951	0.951	0.916	0.912	0.908
FNN	RMSE	3.15	3.16	3.18	3.97	4.01	4.05
	WMAPE	30.23%	30.28%	30.32%	33.58%	33.63%	33.65%
	R^2	0.954	0.953	0.953	0.908	0.906	0.904
Conv-LSTM	RMSE	3.25	3.26	3.27	4.04	4.06	4.08
	WMAPE	30.11%	30.18%	30.23%	32.96%	32.92%	33.04%
	R^2	0.951	0.950	0.950	0.905	0.904	0.903
HA	RMSE	3.43	3.43	3.43	4.34	4.34	4.34
	WMAPE	31.21%	31.21%	31.21%	34.28%	34.28%	34.28%
	R^2	0.945	0.945	0.945	0.890	0.890	0.890

Note. Boldface indicates the best performance among all models.

boarding flow since SARIMA only handles one-dimensional time series. We use the order ARIMA $(2,0,1)(1,1,0)[36]$ for all the stations and fit 159 separate models. This model configuration is the same as that of Cheng, Trépanier, and Sun (2021) and was tested to be suitable for most metro stations.

Applying TRMF, ConvLSTM, and FNN to the original data (or after a normalization) can hardly obtain a forecast better than HA. This phenomenon was also found by Gong et al. (2018, 2022). This is because the OD data are high-dimensional, sparse, noisy, and highly skewed. To improve the forecasting of these models, we apply TRMF, ConvLSTM, and FNN to the residuals after subtracting the HA from the original data. This "mean-removal" processing also weakens the data's periodicity; therefore, we do not use seasonal lags in these models. Besides, because the standard TRMF and ConvLSTM cannot use the boarding flow as extra inputs, we ignore the delayed data availability problem for these models and assume that all historical OD snapshots are available.

6.4. Forecast Result

We apply trained models to the test set and forecast OD matrices of the next three steps at each time interval. Note that OD snapshots of one to two lags ago are unknown; they are replaced by previously forecasted OD snapshots when doing multistep rolling forecasting by HW-DMD/FNN. Next, the boarding flow can be calculated from OD matrices. We compare the forecast accuracy of models in terms of OD flow and boarding flow.

Table 1 shows the results of OD flow forecast. We can see that HW-DMD with a forgetting ratio $\rho = 0.92$ outperforms other models in all evaluation metrics. Even the three-step forecast of HW-DMD is better than the one-step forecast of other models. The advantage of HW-DMD over other models is more significant in the Hangzhou data set. Although TRMF, FNN, and ConvLSTM are trained on the residuals after subtracting the HA from the original data, the improvement of these models compared with HA is limited. In contrast, HW-DMD is directly applied to the original data but provides a significantly better forecast, demonstrating its strong prediction power in handling the sparse, noisy, and high-dimensional OD data. Besides, the performance of the "unweighted" HW-DMD ($\rho = 1$) is slightly behind the weighted version but still better than other models.

Examining the aggregated boarding flow is important because it reflects if the forecast errors in the OD matrices are properly distributed, which is crucial when using OD matrices in traffic assignments. Moreover, the boarding flow itself is of interest to many applications. Table 2 shows the boarding flow forecasting; all models except SARIMA calculate boarding flow by OD matrices. The two HW-DMD models are the best models in most cases. The only exception is that FNN slightly outperforms HW-DMD for the three-step forecast of the Guangzhou data set. Importantly, HW-DMD is the only model that outperforms SARIMA, a well-established boarding flow forecasting model, in both data sets, showing that the forecast of HW-DMD accurately reflects the marginal distribution of OD matrices.

The magnitude of OD flow in a metro system varies significantly in time and space dimensions. Therefore, we further compare HW-DMD with other models

Published in Transportation Science on February 16, 2022 as DOI: 10.1287/trsc.2022.1128.
This article has not been copyedited or formatted. The final version may differ from this version.

Table 2. Models' Performance for Boarding Flow Forecasting

		Guangzhou			Hangzhou		
Method	Criterion	One-step	Two-step	Three-step	One-step	Two-step	Three-step
HW-DMD	RMSE	**93.99**	**102.61**	107.58	**50.08**	**54.14**	**56.32**
$\rho = 0.92$	WMAPE	**6.09%**	**6.68%**	6.98%	**7.38%**	**8.05%**	**8.12%**
	R^2	**0.991**	**0.989**	0.988	**0.989**	**0.988**	**0.987**
HW-DMD	RMSE	94.51	102.46	106.55	51.28	55.66	58.45
$\rho = 1$	WMAPE	6.18%	6.74%	6.98%	7.54%	8.29%	8.43%
	R^2	0.991	0.989	0.988	0.989	0.987	0.986
TRMF	RMSE	126.03	127.87	128.65	77.70	81.19	83.12
	WMAPE	7.92%	8.07%	8.13%	10.00%	10.55%	10.81%
	R^2	0.983	0.983	0.983	0.975	0.972	0.971
FNN	RMSE	101.93	104.00	**106.06**	67.16	68.83	70.77
	WMAPE	6.44%	6.58%	**6.69%**	9.00%	9.22%	9.50%
	R^2	0.989	0.989	**0.988**	0.981	0.980	0.979
Conv-LSTM	RMSE	117.16	121.22	123.40	71.46	75.75	78.07
	WMAPE	6.87%	7.19%	7.35%	8.83%	9.63%	9.98%
	R^2	0.985	0.984	0.984	0.978	0.976	0.974
HA	RMSE	136.56	136.56	136.56	88.25	88.25	88.25
	WMAPE	8.38%	8.38%	8.38%	11.09%	11.09%	11.09%
	R^2	0.980	0.980	0.980	0.967	0.967	0.967
SARIMA	RMSE	110.23	120.60	126.52	55.59	60.97	64.66
	WMAPE	7.15%	7.65%	7.93%	7.86%	8.28%	8.50%
	R^2	0.987	0.985	0.983	0.987	0.984	0.982

Note. Boldface indicates the best performance among all models.

under different scenarios. Panels (a) and (c) of Figure 5 show the forecast RMSE at different times of a day. We can see that the RMSE of HW-DMD is the smallest in most time slots, particularly for the Hangzhou data set. Other models, such as Conv-LSTM, perform slightly better in the early morning and late night, but the difference is close, and the total network OD flow of these periods is pretty small. Panels (b) and (d) of Figure 5 show the forecast RMSE for OD pairs with different flow magnitudes. The forecast RMSE of HW-DMD is considerably lower than other models for high-flow OD pairs (average half-hour OD flow larger than 2^4). Note that the number of OD pairs drops exponentially with the increase of OD flow, showing the superior forecast capability of HW-DMD for highly skewed data.

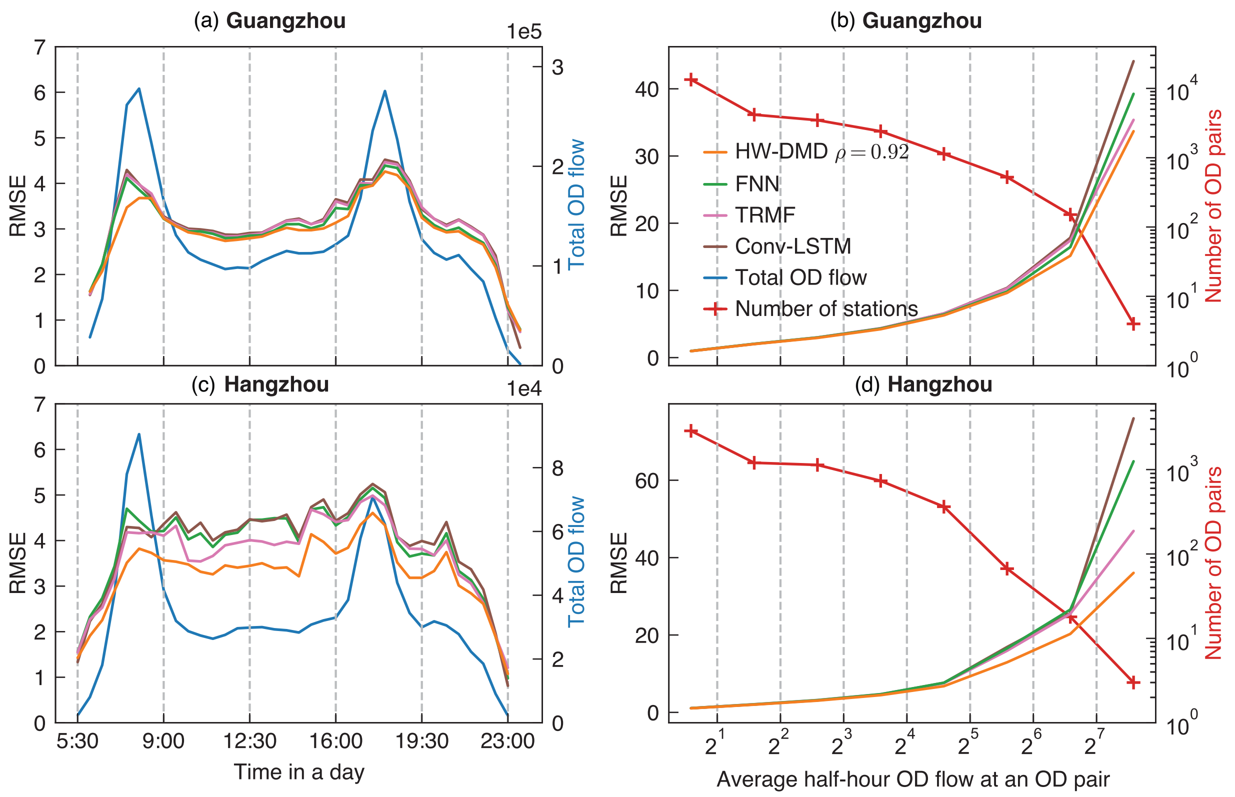
Finally, we show the real and one-step forecast of OD flow at four representative OD pairs of the Guangzhou metro in Figure 6. The OD flow exhibits a clear daily periodicity, explaining why HA already works reasonably well. Compared with FNN, HW-DMD is better at forecasting the fluctuation of high-flow OD pairs, as shown in panels (a) and (b) of Figure 6. In Figure 6(a), the forecast of HW-DMD is often lower than the real value; this is hard to avoid since there is a two-lag delay when collecting the real OD flow. More OD pairs in the system are like panels (c) and (d) of Figure 6 with a low flow but high noise. Under such high volatility, the forecast by HW-DMD reflects a smooth average value. In fact, the performances of other models are often undermined by noise. The SVD truncation to the data greatly enhances HW-DMD's ability in handling the noise data (Figure 1). Overall, HW-DMD achieves a great balance between forecasting and noise reduction, which is particularly hard for such a high-dimensional system with diverse flow magnitudes.

6.5. Effect of the Low-Rank Assumption

The demands of majority OD pairs are small and sparse by nature, making it difficult for a forecasting model to distinguish random fluctuation (noise) and intrinsic dynamic patterns. Taking the OD pair shown in Figure 6(d) as an example, the randomness in this OD pair is quite large compared with its average flow (low signal-to-noise ratio). A good forecasting should be robust to the noise while maintaining accurate cumulative effects of OD pairs in total (e.g., the boarding flow). This section evaluates the impact of using the low-rank assumption on forecasting and noise filtering.

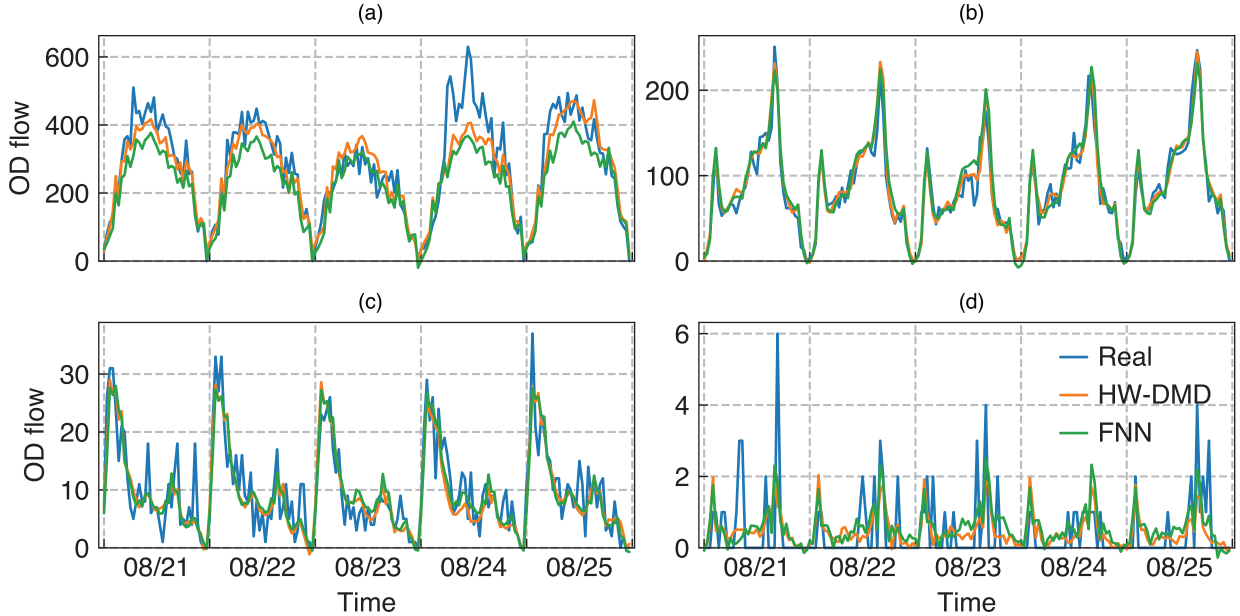
According to Section 5.2, the forecast of HW-DMD is always on the column space of U_Y. Therefore, the best possible value of an OD snapshot $\hat{\mathbf{y}}_i$ calculated by HW-DMD is the rank-reduced full-size data, that is, $U_Y U_Y^\top \mathbf{y}_i$, which is the upper bound of an HW-DMD's forecast ability. Figure 7 shows how well this low-rank approximation fits the original data. We can see that the low-rank approximation keeps most information for the high-demand OD pair of Figure 7(a). In contrast, most fluctuations in the sparse-demand OD pair of Figure 7(b) are truncated. By comparing with HA, we can see that the low-rank approximation reflects the average daily pattern of the sparse-

Figure 5. (Color online) The RMSE of OD Flow Forecasting at Different Times and Different OD Pairs



Notes. Panels (a) and (c) show the RMSE of OD matrix forecasting at every 30-minute interval, along with the total OD flow in the network. Using 2^i as boundaries, we divide OD pairs into groups according to their average half-hour OD flow; the forecast RMSE at each group and the number of OD pairs of each group are shown in panels (b) and (d).

Figure 6. (Color online) The Real and Forecasted Time Series of Four Selected OD Pairs of the Guangzhou Metro



Notes. Panel (a) shows the busiest OD pair in the Guangzhou metro data set. Panels (a)–(d) are in a flow-decreasing order.

demand OD pair, which is a reasonable approximation when considering the cumulative effects of OD pairs. Therefore, the rank truncation is crucial for filtering the noise in a large number of sparse-demand OD pairs.

Table 3 further quantitatively evaluates the differences between the original OD data and its low-rank approximation. The results in Table 3 are the forecast upper bound of HW-DMD under the current rank-reduced space. By comparing Table 3 with the forecast of HW-DMD in Tables 1 and 2, we can see that a significant portion of the forecast error of HW-DMD essentially attributes to the rank truncation, but there is still space to improve the current HW-DMD model (e.g., by higher order, larger r_X, more regression covariates).

In choosing the rank-reduced space, the two rank parameters in HW-DMD balance the trade-off between forecast accuracy and model complexity. Based on the results of hyperparameter tuning, a further increase in rank r_Y may result in overfitting (bringing the noise into the rank-reduced target data). We can further slightly improve the forecasting accuracy of HW-DMD by increasing the rank r_X (related to the rank-reduced input data), but we here prefer a compact model with a smaller r_X at the cost of slight accuracy loss. Lastly, the current HW-DMD chooses the rank-reduced space purely based on the leading singular values, which may be sensitive and not optimal when encountering significant data anomalies and failures (Duke, Soria, and Honnery 2012). Using optimized DMD (Chen, Tu, and Rowley 2012) or combining with an anomaly detection algorithm (Scherl et al. 2020) could further improve the current HW-DMD.

6.6. Effect of the Online Update

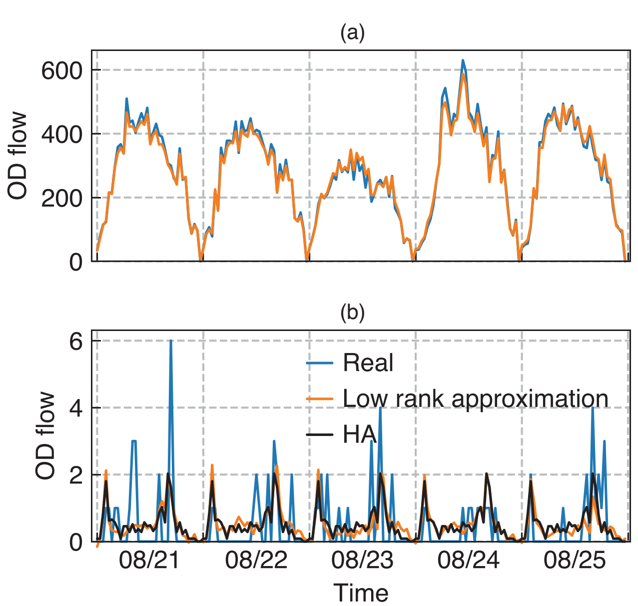
The online update algorithm proposed in Section 5.3 can update HW-DMD's parameters daily without storing historical data, which is computationally more efficient. On the Guangzhou metro training set, it takes 18.7 ± 0.43 seconds to train an HW-DMD model, whereas the online update only takes 1.0 ± 0.03 seconds for each day.[3] Other benchmark models have much longer training times than HW-DMD (more than one minute for FNN and more than 20 minutes for TRMF and Conv-LSTM). Besides the training time, we particularly care about whether errors will accumulate if we keep using the online update algorithm for a long time. Therefore, we apply the online update algorithm to all of the two-month data after the training set of the Guangzhou data set to evaluate its long-term effect. In comparison, we retrain two HW-DMD models (with $\rho = 0.92$ and 1, respectively) every day using all historical data up until the latest. The results are shown in Figure 8. We summarize the key findings for Figure 8 as follows.

Table 3. The Difference Between Original Data and the Low-Rank Approximation

Variable	Criterion	Guangzhou	Hangzhou
OD flow	RMSE	2.82	3.00
	WMAPE	28.80%	30.65%
	R^2	0.963	0.947
Boarding flow	RMSE	64.15	36.89
	WMAPE	4.69%	5.95%
	R^2	0.996	0.994

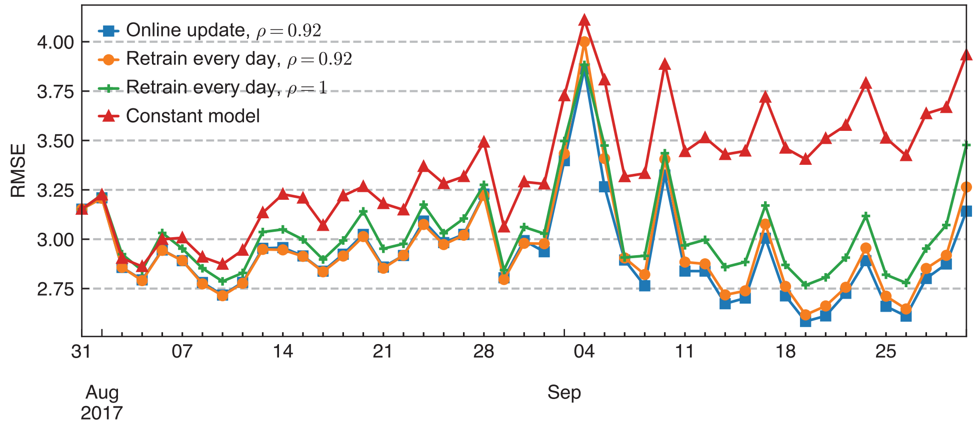
- The RMSE of a constant model gradually increases over time. This indicates that the metro system's dynamics are time-evolving; thus, forecasting models should be updated/retrained regularly for better performance.
- The RMSE curve of the online update algorithm clings to the model ($\rho = 0.92$) retrained every day by entire historical data, showing that the online HW-DMD update algorithm works consistently well in long-term applications. For a large training set (e.g., after September in Figure 8), the online update approach even performs slightly better than retraining.
- Properly reducing the weight for old data improves the forecast. Compare $\rho = 0.92$ with $\rho = 1$ for the two retrained models; the benefits of forgetting the old data become more significant as the training data increase.
- The OD flow of certain weekdays can be harder to forecast, especially for the forecast of September. The RMSEs on Fridays are significantly higher than on other weekdays.

Figure 7. (Color online) Comparing OD Flow with Its Low-Rank Approximation in Two Guangzhou Metro OD Pairs



Note. Panels (a) and (b) correspond, respectively, to panels (a) and (d) in Figure 6.

Figure 8. (Color online) The Evolvement of Forecast RMSE Under Different HW-DMD Update Methods (Guangzhou Data Set)



Notes. Each marker represents the RMSE of forecasted OD flow during a day. Numbered dates in the horizontal axis are Mondays; weekends are skipped.

Many forecasting models do not consider the time-evolving dynamics of a metro system. Regular retraining can be prohibitive, especially for complicated models (e.g., deep learning models). This experiment shows that the online update algorithm for HW-DMD is a memory-saving and accurate approach to keep an HW-DMD model up to date.

7. Conclusions and Discussion

This paper proposes a high-order weighted dynamic mode decomposition (HW-DMD) model to solve the real-time short-term OD matrix forecasting problem in metro systems. Experiments show that HW-DMD significantly outperforms common forecasting models under the high-dimensional, sparse, noisy, and skewed OD data. Particularly, we address the delayed data availability problem and the time-evolving dynamics of metro systems, which are often ignored in the literature. The idea of the forgetting rate and online update in dealing with a time-evolving system is also beneficial for other forecasting models. Moreover, the implementation of HW-DMD is simple, and the computation is very efficient, providing a promising solution to general high-dimensional time-series forecasting problems.

We discuss several future research directions. (1) Current HW-DMD reshapes OD matrices into vectors for dimensionality reduction. However, performing dimensionality reduction directly on OD matrices may better utilize the column/row-wise correlations and produce more concise models (Chen, Xiao, and Yang 2021; Gong et al. 2022). A difficulty in this direction is that the low-rank feature in metro OD matrices is relatively weak because the diagonal elements of metro OD matrices are all zeros. (2) Another future direction is to use a nonlinear model instead of the current linear model in the reduced space, such as the deep factor model (Wang et al. 2019a). But a limitation for a nonlinear model is that an online update method may be difficult to derive or even impossible. (3) Lastly, current HW-DMD uses external features, such as the boarding flow, simply as covariates. Incorporating more general features (e.g., weather, events) and network structure to improve the HW-DMD is worth investigating.

Acknowledgments

A preliminary version of this paper was accepted for presentation at the 15th CASPT international conference (CASPT2021) on public transport, organized by Dr. Yuval Hadas, Prof. Avi Ceder, and Prof. Oded Cats.

Endnotes

[1] We use $(\cdot)^+$ to denote the Moore–Penrose inverse of a matrix.

[2] See https://doi.org/10.5281/zenodo.3145404.

[3] We report the mean ± standard deviation of 20 runs. Tests were run on a computer with an Intel Core i7-8700 processor and 24 gigabytes of RAM.

References

Alfatlawi M, Srivastava V (2020) An incremental approach to online dynamic mode decomposition for time-varying systems with applications to EEG data modeling. *J. Comput. Dynam.* 7(2): 209–241.

Arbabi H, Mezić I (2017) Ergodic theory, dynamic mode decomposition, and computation of spectral properties of the Koopman operator. *SIAM J. Appl. Dynamical Systems* 16(4):2096–2126.

Avila A, Mezić I (2020) Data-driven analysis and forecasting of highway traffic dynamics. *Nature Comm.* 11(1):1–16.

Brunton SL, Brunton BW, Proctor JL, Kaiser E, Kutz JN (2017) Chaos as an intermittently forced linear system. *Nat. Comm.* 8(1):1–9.

Chen KK, Tu JH, Rowley CW (2012) Variants of dynamic mode decomposition: Boundary condition, Koopman, and Fourier analyses. *J. Nonlinear Sci.* 22(6):887–915.

Chen R, Xiao H, Yang D (2021) Autoregressive models for matrix-valued time series. *J. Econometrics* 222(1):539–560.

Chen E, Ye Z, Wang C, Xu M (2019) Subway passenger flow prediction for special events using smart card data. *IEEE Trans. Intelligent Transportation Systems* 21(3):1109–1120.

Cheng Z, Trépanier M, Sun L (2021) Incorporating travel behavior regularity into passenger flow forecasting. *Transportation Res. Part C Emerging Tech.* 128:103200.

Chu KF, Lam AY, Li VO (2020) Deep multi-scale convolutional LSTM network for travel demand and origin-destination predictions. *IEEE Trans. Intelligent Transportation Systems* 21(8): 3219–3232.

Dai X, Sun L, Xu Y (2018) Short-term origin-destination based metro flow prediction with probabilistic model selection approach. *J. Advanced Transportation* 2018:5942763.

Duke D, Soria J, Honnery D (2012) An error analysis of the dynamic mode decomposition. *Experiments Fluids* 52(2):529–542.

Eckart C, Young G (1936) The approximation of one matrix by another of lower rank. *Psychometrika* 1(3):211–218.

Forni M, Hallin M, Lippi M, Reichlin L (2000) The generalized dynamic-factor model: Identification and estimation. *Rev. Econom. Statist.* 82(4):540–554.

Gong Y, Li Z, Zhang J, Liu W, Zheng Y (2022) Online spatiotemporal crowd flow distribution prediction for complex metro system. *IEEE Trans. Knowledge Data Engrg.* 34(2):865–880.

Gong Y, Li Z, Zhang J, Liu W, Zheng Y, Kirsch C (2018) Network-wide crowd flow prediction of Sydney trains via customized online non-negative matrix factorization. *Proc. 27th ACM Internat. Conf. Inform. Knowledge Management* (ACM, New York), 1243–1252.

Hemati MS, Williams MO, Rowley CW (2014) Dynamic mode decomposition for large and streaming datasets. *Physics Fluids* 26(11):111701.

Hu J, Yang B, Guo C, Jensen CS, Xiong H (2020) Stochastic origin-destination matrix forecasting using dual-stage graph convolutional, recurrent neural networks. *Proc. 36th Internat. Conf. Data Engrg. (ICDE)* (IEEE, Piscataway, NJ), 1417–1428.

Ke J, Qin X, Yang H, Zheng Z, Zhu Z, Ye J (2021) Predicting origin-destination ride-sourcing demand with a spatio-temporal encoder-decoder residual multi-graph convolutional network. *Transportation Res. Part C Emerging Tech.* 122:102858.

Kwak S, Geroliminis N (2021) Travel time prediction for congested freeways with a dynamic linear model. *IEEE Trans. Intelligent Transportation Systems* 22(12):7667–7677.

Lam C, Yao Q, Bathia N (2011) Estimation of latent factors for high-dimensional time series. *Biometrika* 98(4):901–918.

Le Clainche S, Vega JM (2017) Higher order dynamic mode decomposition. *SIAM J. Appl. Dynamical Systems* 16(2):882–925.

Li Y, Wang X, Sun S, Ma X, Lu G (2017) Forecasting short-term subway passenger flow under special events scenarios using multiscale radial basis function networks. *Transportation Res. Part C Emerging Tech.* 77:306–328.

Liu Y, Liu Z, Jia R (2019) DeepPF: A deep learning based architecture for metro passenger flow prediction. *Transportation Res. Part C Emerging Tech.* 101:18–34.

Liu J, Zheng F, van Zuylen HJ, Li J (2020) A dynamic OD prediction approach for urban networks based on automatic number plate recognition data. *Transportation Res. Procedia* 47:601–608.

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

Noursalehi P, Koutsopoulos HN, Zhao J (2021) Dynamic origin-destination prediction in urban rail systems: A multi-resolution spatio-temporal deep learning approach. *IEEE Trans. Intelligent Transportation Systems*, ePub ahead of print January 11, https://doi.org/10.1109/TITS.2020.3047047.

Proctor JL, Brunton SL, Kutz JN (2016) Dynamic mode decomposition with control. *SIAM J. Appl. Dynamical Systems* 15(1):142–161.

Ren J, Xie Q (2017) Efficient od trip matrix prediction based on tensor decomposition. *Proc. 18th IEEE Internat. Conf. Mobile Data Management* (IEEE, Piscataway, NJ), 180–185.

Rowley CW, Mezić I, Bagheri S, Schlatter P, Henningson D (2009) Spectral analysis of nonlinear flows. *J. Fluid Mech.* 641(1):115–127.

Scherl I, Strom B, Shang JK, Williams O, Polagye BL, Brunton SL (2020) Robust principal component analysis for modal decomposition of corrupt fluid flows. *Physical Rev. Fluids* 5(5):054401.

Schmid PJ (2010) Dynamic mode decomposition of numerical and experimental data. *J. Fluid Mech.* 656:5–28.

Shen L, Shao Z, Yu Y, Chen X (2020) Hybrid approach combining modified gravity model and deep learning for short-term forecasting of metro transit passenger flows. *Transp. Res. Record* 2675(1):25–38.

Shi X, Chen Z, Wang H, Yeung DY, Wong WK, Woo W-C (2015) Convolutional LSTM network: A machine learning approach for precipitation nowcasting. *Adv. Neural Inform. Processing Systems* 28:802–810.

Sun Y, Leng B, Guan W (2015) A novel wavelet-SVM short-time passenger flow prediction in Beijing subway system. *Neurocomput.* 166:109–121.

Toqué F, Côme E, El Mahrsi MK, Oukhellou L (2016) Forecasting dynamic public transport origin-destination matrices with long-short term memory recurrent neural networks. *Proc.* 19th *Internat. Conf. Intelligent Transportation Systems* (IEEE, Piscataway, NJ), 1071–1076.

Tu JH, Rowley CW, Luchtenburg DM, Brunton SL, Kutz JN (2014) On dynamic mode decomposition: Theory and applications. *J. Comput. Dynamics* 1(2):391–421.

Wang S, Miao H, Chen H, Huang Z (2020) Multi-task adversarial spatial-temporal networks for crowd flow prediction. *Proc. 29th ACM Internat. Conf. Inform. Knowledge Management* (ACM, New York), 1555–1564.

Wang Y, Smola A, Maddix DC, Gasthaus J, Foster D, Januschowski T (2019a) Deep factors for forecasting. *Internat. Conf. Machine Learn. (ICML)*, 6607–6617.

Wang Y, Yin H, Chen H, Wo T, Xu J, Zheng K (2019b) Origin-destination matrix prediction via graph convolution: A new perspective of passenger demand modeling. *Proc. 25th ACM SIGKDD Internat. Conf. Knowledge Discovery Data Mining* (ACM, New York), 1227–1235.

Wei Y, Chen MC (2012) Forecasting the short-term metro passenger flow with empirical mode decomposition and neural networks. *Transportation Res. Part C Emerging Tech.* 21(1):148–162.

Xiong X, Ozbay K, Jin L, Feng C (2020) Dynamic origin-destination matrix prediction with line graph neural networks and Kalman filter. *Transportation Res. Rec.* 2674(8):491–503.

Yu HF, Rao N, Dhillon IS (2016) Temporal regularized matrix factorization for high-dimensional time series prediction. *Adv. Neural Inform. Processing Systems* 29:847–855.

Yu Y, Zhang Y, Qian S, Wang S, Hu Y, Yin B (2021) A low rank dynamic mode decomposition model for short-term traffic flow prediction. *IEEE Trans. Intelligent Transportation Systems* 22(10):6547–6560.

Zhang J, Chen F, Wang Z, Liu H (2019b) Short-term origin-destination forecasting in urban rail transit based on attraction degree. *IEEE Access* 7:133452–133462.

Zhang H, Rowley CW, Deem EA, Cattafesta LN (2019a) Online dynamic mode decomposition for time-varying systems. *SIAM J. Appl. Dynamical Systems* 18(3):1586–1609.

Zhang D, Xiao F, Shen M, Zhong S (2021a) DNEAT: A novel dynamic node-edge attention network for origin-destination demand prediction. *Transportation Res. Part C Emerging Tech.* 122:102851.

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

Zhang J, Chen F, Cui Z, Guo Y, Zhu Y (2021b) Deep learning architecture for short-term passenger flow forecasting in urban rail transit. *IEEE Trans. Intelligent Transportation Systems* 22(11):7004–7014.