arXiv:2001.02492v3 [math.OC] 28 Nov 2020

Nonlinear Traffic Prediction as a Matrix Completion Problem with Ensemble Learning

Wenqing Li[1], Chuhan Yang[2], and Saif Eddin Jabari*[1,2]

[1]New York University Abu Dhabi, Saadiyat Island, P.O. Box 129188, Abu Dhabi, U.A.E.
[2]New York University Tandon School of Engineering, Brooklyn NY, U.S.A.

Abstract. We focus on short-term traffic prediction for signalized traffic operations management. Specifically, we focus on predicting sensor states in high-resolution (second-by-second). Most work on traffic forecasting has focused on predicting aggregated traffic variables, typically over intervals that are no shorter than 5 minutes. We develop a (big) data-driven methodology for predicting sensor states in high-resolution. Our contributions can be summarized as offering three insights: first, we show how the prediction problem can be modeled as a matrix completion problem. Second, we employ a block-coordinate descent algorithm and demonstrate that the algorithm converges in sub-linear time to a block coordinate-wise optimizer. This allows us to capitalize on the "bigness" of high-resolution data in a computationally feasible way. Third, we develop an ensemble learning (or adaptive boosting) approach to reduce the training error to within any arbitrary error threshold. The latter utilizes past days so that the boosting can be interpreted as capturing periodic patterns in the data. The performance of the proposed method is analyzed theoretically and tested empirically using both simulated data and a real-world high-resolution traffic dataset from Abu Dhabi, UAE. Our experimental results show that the proposed method outperforms other state-of-the-art algorithms.

Keywords: Traffic prediction, high-resolution data, signalized intersections, adaptive control, matrix completion, kernel regression, sparse approximation, ensemble learning, adaptive boosting.

1 Introduction

Governments worldwide expend considerable effort and investment to manage the day-to-day operations of urban networks. Today's urban road networks are highly complex interacting systems of technologies that include sensory devices, communications technologies, and monitoring tools. These serve a diverse set of operational objectives, from rapid response to incidents (*safety*) to managing emergencies and special events (*security*) to the day-to-day operation of traffic lights and congestion (*efficiency*). Adaptive control systems, namely traffic lights [19] and ramp meters [62] are key technologies employed by operators to achieve these objectives. According to the Federal Highway Administration (FHWA), adaptive signal control technologies can improve travel time by more than 10% and such improvements can exceed 50% when they replace outdated controls [10]. Decision-making in these systems occurs at a very fine time cadence, e.g., actuated/adaptive traffic controllers update their controls every time a vehicle is detected by one of the sensors in the system. The control logic employed by adaptive controllers involves coordination among the controllers at different locations (e.g., adjacent intersections or consecutive ramp meters). Coordination essentially entails one controller informing another to *anticipate* vehicle arrivals, and then producing a coordinated response. In other words, coordination involves the use of short-term predictions of vehicle arrivals to sensor stations. In practice, this is done using rudimentary techniques, namely assuming uniform vehicle speeds between sensor locations. But such assumptions are only valid in low-volume traffic and prediction is high-volume traffic is more challenging.

Prediction problems in traffic have focused on applications that do not require the fine temporal resolutions

Corresponding author. Email: sej7@nyu.edu.

needed for adaptive control. They almost exclusively use data that is aggregated at the 5-15 minute level. Numerous studies have investigated the impact of data aggregation level on prediction accuracy [18, 25, 69] These studies generally agree that prediction accuracies drop as the aggregation levels get finer. [56] further demonstrated that data-driven techniques like neural networks tend to produce more accurate predictions than conventional statistical models (like time series) as the data aggregation levels get finer (from 10 minutes down to 1 minute). But they argued that finer aggregations are not necessary for the applications they considered. [70] argued that aggregation may result in bias, even for applications that do not necessarily operate on disaggregated data. They specifically pointed to non-stationarity, long-run memory, and structural change as features that are lost upon aggregation, but that can be of extreme importance, particularly in traffic management. This is especially the case for traffic signal operations and adaptive control.

The collection and archival of disaggregated traffic data is a growing trend in U.S. cities, but it is also emerging in other parts of the world. The first such system appeared in northern California for highway traffic operations [54, 55, 63]. Various traffic operations tools were developed using these data [76], two notable examples that requires disaggregated data are vehicle re-identification [9] and vehicle classification [72]. Similar systems were subsequently deployed for purposes of monitoring the performance of signalized arterials in real-time, in Texas [1], Indiana [64], and Minnesota [46]. These systems have dubbed the disaggregated sensor data *high-resolution data* or *event-based* data. These datasets consist of on/off sensor states over time, typically recorded on a second-by-second basis (which is the control cadence). The datasets also include the status of traffic signal heads (greed, orange, red) over time and the signal switch times. It is notable that these data do not typically correspond to raw sensor states; in the case of inductive loop detectors, the raw sensor states are inductance drops that the controller *interprets* as vehicle presence depending on controller sensitivity settings. In this paper, we will use the nomenclature "high-resolution data" to refer to these types of datasets, pertaining to traffic signal systems. The literature includes numerous examples of applications that utilize high-resolution data for traffic operations applications, including vehcile classification in interrupted traffic [45], detection of over-saturated traffic conditions [77], signal timing optimization [22, 23], and estimating red-light running frequencies [8].

Existing traffic data analysis methods are either model-based or data-driven. Techniques that are geared towards the estimation of traffic densities or speeds from both fixed and mobile sensors are examples of the former [26–29, 48, 57–59, 61, 79, 81]. Data-driven techniques are becoming more popular with the increasing availability of traffic data. For traffic prediction, data-driven methodologies fall in one of two major categories: parametric approaches and non-parametric approaches. Time series models dominate the category of parametric techniques. For example, auto-regressive integrated moving average (ARIMA) [7, 37, 39, 40, 67, 74] and vector variants (VARIMA) [16, 33–35, 66] have been widely used and demonstrated to be successful. These models assume a parametric linear relationship between the label and a finite number of past states of the label itself. Another class of methods combine *partial likelihood inference* with *generalized linear modeling* (GLM) techniques to generalize both the linear relationship between predictors (past states) and the predictions and the Gaussian noise assumption. An example of this class of models is the generalized auto-regressive moving average model (GARMA) [3], but we will refer to them more generally as *time series following generalized linear models*. [14] demonstrated how the basic ideas of exponential distribution families and monotone link functions used in GLMs generalize directly to the case of time series data. This readily provides an apparatus for generalizing linear time series to binary time series data (in a manner similar to logistic regression). To the best of our knowledge, these models have not been applied to traffic prediction problems. We will test them alongside other techniques in our experiments.

Non-parametric methods, on the other hand, do not assume any functional model forms and are typically data-driven. The basic idea behind non-parametric techniques is that they learn a general form from the historical data and use it to predict future data. Non-parametric methods can be divided into two types: non-parametric regression such as support vector regression (SVR) [24, 32, 75] and artificial neural networks (ANN) [36, 47, 49–51, 53]. Compared with time series models which assume that traffic data vary linearly over time, SVR and ANN techniques can capture nonlinear variations in traffic data. The advantage of SVR models is that they can learn representative features by using various kernels. For this reason, SVR has been successfully applied to predict traffic data such as flow [24, 32], headway [80], and travel time [11, 12, 30, 75]. [32] further proposed an online version of SVR, which can efficiently update the model when new data is added. Alternatively, ANNs are among the first non-parametric methods that have been applied to traffic prediction, and there is a wealth of literature on the subject, from simple multilayer perceptrons (MLPs) [49] to

more complicated architectures such as recurrent neural networks (RNNs) [36, 50, 53], convolutional neural networks (CNNs) [51], and even combinations of RNNs and CNNs [47].

To our best of knowledge, short-term prediction using high-resolution data remains an open problem. As discussed above the accuracy of traditional prediction techniques tend to drop as the temporal aggregation levels get finer. This paper develops a novel non-parametric traffic prediction method using high-resolution data for signalized traffic operations management applications, e.g., [41, 42]. When using high-resolution data, much larger volumes of data are required in order to observe patterns and produce predictions. One also improves accuracy with frequent updates, that is, updating the predictions in real-time as opposed to utilizing a parametric model that was fitted off-line. These two ingredients (large volumes of data and online estimation) are key aspects of the proposed techniques. The main challenge is, thus, computational/algorithmic in nature and the central theme of this paper is the algorithmic aspects of the problem. To address these challenges, we propose a matrix completion formulation and block coordinate descent method. Our contributions are threefold: First, we present a novel formulation of traffic prediction as a matrix completion problem. Our formulation extends traditional modeling approaches (e.g., vector autoregressive time series) in that the nonlinear dependence of the labels on the inputs can be accommodated with ease. The model can be described as being *transductive* in that we leverage statistical information in the testing data (e.g., correlations in the inputs). This, in turn, allows for a dynamic implementation that can adapt to streaming high-resolution data. We consider (and leverage), for the first time, the natural sparsity and "bigness" of high-resolution traffic data, in a way that is intrinsic to our formulation. Second, we develop a block coordinate descent technique to solve the matrix completion problem and analytically demonstrate that the algorithm converges to a block coordinate-wise minimizer (or Nash point). We also demonstrate analytically that the convergence rate is sub-linear, meaning that each iteration (which only involves a series of algebraic manipulations) produces an order of magnitude reduction in the distance from optimality. This means that only few iterations are required to solve the problem, allowing for real-time implementation. Third, the performance of the model is further boosted by forming an ensemble of matrix completion problems using datasets from past days. The merits of the proposed ensemble learning approach are twofold: training errors can be reduced to arbitrarily small numbers while capturing and exploiting trends in the data from past days. This also offers interpretability that is usually sacrificed with most large-scale machine learning tools (e.g., deep neural nets).

The remainder of this paper is organized as follows: Sec. 2 formulates the traffic prediction problem as a matrix completion problem. The block-coordinate descent algorithm is presented in Sec. 3 along with all convergence results. Sec. 4 develops the ensemble learning extension, presents an analysis of the training error, and brief analysis of the time complexity of the overall approach. Sec. 5 presents three sets of numerical experiments, which include both simulated and real-world data and Sec. 6 concludes the paper.

2 Traffic Prediction as a Matrix Completion Problem: Problem Formulation

2.1 Notation and Preliminaries

The task of traffic prediction is to learn a mapping between the output space of the predicted data (future data) and the input space of predictors (recent data). Let $\mathbf{x}(t) \in \{0,1\}^n$ represent a set of n network detector states at time $t \in \mathbb{Z}_+$; an element of the vector $\mathbf{x}(t)$ is 0 if the corresponding detector is unoccupied at time t and is equal to 1, otherwise. We consider as input at time step t the present detector state, and previous states up to a lag of size L. The parameter L depends on temporal correlations in the data, and has been comprehensively discussed in previous work, e.g., [71]. We represent this input at time t by applying a backshift operator up to order L, $\mathsf{B}_L : \mathbb{R}^n \to \mathbb{R}^{nL}$, which is given by

$$\mathsf{B}_L \mathbf{x}(t) = \begin{bmatrix} \mathbf{x}(t-L+1) \\ \vdots \\ \mathbf{x}(t) \end{bmatrix}. \tag{1}$$

The output of the prediction, performed at time t, which we denote by $\mathbf{y}(t) \in \mathbb{R}^n$, is simply the state of the n network detectors at some prediction horizon $H \in \mathbb{Z}_+$ time steps later, that is $\mathbf{y}(t) = \mathbf{x}(t+H)$. Let T_{tr} denote the number of training samples. The inputs in the training sample cover the interval $t \in \{1, \ldots, T_{\text{tr}}\}$, which we denote by $\{\mathsf{B}_L \mathbf{x}_{\text{tr}}(t), \mathbf{y}_{\text{tr}}(t)\}_{t=1}^{T_{\text{tr}}}$. As in (1)

$$\mathsf{B}_L \mathbf{x}_{\text{tr}}(t) = \begin{bmatrix} \mathbf{x}_{\text{tr}}(t-L+1) \\ \vdots \\ \mathbf{x}_{\text{tr}}(t) \end{bmatrix} \tag{2}$$

is the input associated with the sample at time t and the corresponding output vector is $\mathbf{y}_{\text{tr}}(t) = \mathbf{x}_{\text{tr}}(t+H) \in \mathbb{R}^n$.

The inputs and outputs at a single time step t are illustrated in Fig. 1

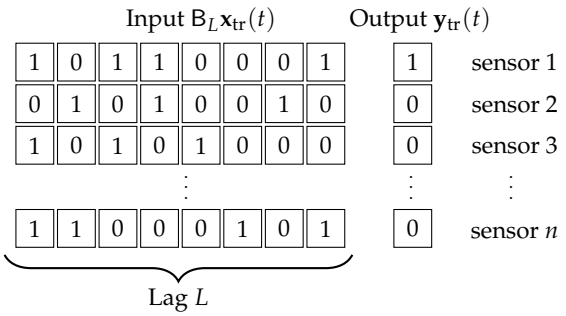


FIGURE 1: An Illustration of Training Inputs and Outputs at a Single Time Step

2.2 Traffic Prediction and Matrix Rank Minimization

Assuming a linear relationship between the input and output (which will be relaxed below), we may write

$$\mathbf{y}_{\text{tr}}(t) = \langle \mathbf{W}, \mathsf{B}_L \mathbf{x}_{\text{tr}}(t) \rangle = \mathbf{W}^\top \mathsf{B}_L \mathbf{x}_{\text{tr}}(t), \tag{3}$$

where $\mathbf{W} \in \mathbb{R}^{n \times nL}$ is the (regression) coefficients matrix and $\langle \cdot, \cdot \rangle$ is the inner product. Let T_{te} be the number of testing samples. We denote the set of testing data over the prediction interval $t \in \{T_{\text{tr}}+1, \ldots, T_{\text{tr}}+T_{\text{te}}\}$ by $\{\mathsf{B}_L \mathbf{x}_{\text{te}}(t), \mathbf{y}_{\text{te}}(t)\}_{t=T_{\text{tr}}+1}^{T_{\text{tr}}+T_{\text{te}}}$ and write their (linear) relationship as

$$\mathbf{y}_{\text{te}}(t) = \langle \mathbf{W}, \mathsf{B}_L \mathbf{x}_{\text{te}}(t) \rangle = \mathbf{W}^\top \mathsf{B}_L \mathbf{x}_{\text{te}}(t). \tag{4}$$

To set the prediction problem up as a matrix completion problem, we first define the input data matrices

$$\mathbf{X}_{\text{tr}} \equiv \begin{bmatrix} \mathsf{B}_L \mathbf{x}_{\text{tr}}(1) & \cdots & \mathsf{B}_L \mathbf{x}_{\text{tr}}(T_{\text{tr}}) \end{bmatrix} \in \mathbb{R}^{nL \times T_{\text{tr}}} \tag{5}$$

and

$$\mathbf{X}_{\text{te}} \equiv \begin{bmatrix} \mathsf{B}_L \mathbf{x}_{\text{te}}(T_{\text{tr}}+1) & \cdots & \mathsf{B}_L \mathbf{x}_{\text{te}}(T_{\text{tr}}+T_{\text{te}}) \end{bmatrix} \in \mathbb{R}^{nL \times T_{\text{te}}}. \tag{6}$$

Their corresponding output matrices are defined as

$$\mathbf{Y}_{\text{tr}} \equiv \begin{bmatrix} \mathbf{y}_{\text{tr}}(1) & \cdots & \mathbf{y}_{\text{tr}}(T_{\text{tr}}) \end{bmatrix} \in \mathbb{R}^{n \times T_{\text{tr}}} \tag{7}$$

and

$$\mathbf{Y}_{\text{te}} \equiv \begin{bmatrix} \mathbf{y}_{\text{te}}(T_{\text{tr}}+1) & \cdots & \mathbf{y}_{\text{te}}(T_{\text{tr}}+T_{\text{te}}) \end{bmatrix} \in \mathbb{R}^{n \times T_{\text{te}}}. \tag{8}$$

We then define the *joint matrix*, which concatenates both the training and the testing data, as

$$\mathbf{Z} \equiv \begin{bmatrix} \mathbf{Y}_{\text{tr}} & \mathbf{Y}_{\text{te}} \\ \mathbf{X}_{\text{tr}} & \mathbf{X}_{\text{te}} \end{bmatrix} \in \mathbb{R}^{(n+nL) \times (T_{\text{tr}}+T_{\text{te}})}. \tag{9}$$

The entries corresponding to $\mathbf{Y}_{\text{te}}$ in $\mathbf{Z}$ are unknown and constitute the solution of the prediction problem. Hence the prediction problem can be cast as a matrix completion problem. From (3) and (4), we have that

$$\begin{bmatrix} \mathbf{Y}_{\text{tr}} & \mathbf{Y}_{\text{te}} \end{bmatrix} = \langle \mathbf{W}, \begin{bmatrix} \mathbf{X}_{\text{tr}} & \mathbf{X}_{\text{te}} \end{bmatrix} \rangle, \tag{10}$$

which implies that the joint matrix $\mathbf{Z}$ as defined in (9) has low rank as a result of linear dependencies in the rows of $\mathbf{Z}$ implied by (10). Thus, removing entries corresponding to $\mathbf{Y}_{\text{te}}$ and completion of the joint matrix $\mathbf{Z}$ via rank-minimization techniques is equivalent to directly producing predictions (and bypassing the need to estimate $\mathbf{W}$). The higher the linear dependence in the matrix $\mathbf{Z}$ (represented by its rank), the more parsimonious the resulting prediction, which is a desirable feature from a statistical point of view.

2.3 Relaxing the Linearity Using Kernels

We relax the linearity assumption by mapping the inputs to a higher dimensional feature space, via kernel "basis" functions, and solve a linear prediction problem in the higher dimensional space [68]. In the training stage, the nonlinear relationship can be expressed as

$$\mathbf{y}_{\text{tr}}(t) = \langle \mathbf{W}, \phi(\mathsf{B}_L \mathbf{x}_{\text{tr}}(t)) \rangle, \tag{11}$$

where $\phi : \mathbb{R}^{nL} \to \mathbb{R}^h$ is the nonlinear function that maps the input space to the high-dimensional feature space ($h > nL$) and with slight notation abuse $\mathbf{W} \in \mathbb{R}^{n \times h}$ is the regression matrix. Note that the inner product of $\mathbf{W}$ and $\phi(\mathbf{x})$ can be learned using the *kernel trick* avoiding the need to compute the map ϕ; see for example [43, 65]. Mapping functions are related to kernels via

$$\mathsf{K}(\mathbf{x}_1, \mathbf{x}_2) = \langle \phi(\mathbf{x}_1), \phi(\mathbf{x}_2) \rangle = \phi(\mathbf{x}_1)^\top \phi(\mathbf{x}_2), \tag{12}$$

where K is a kernel (a weighted distance measure). We note that not all known kernel functions have known mapping functions (e.g., the widely used Gaussian kernels). We shall prescribe a kernel and treat the mapping function as an unknown. Specifically, we shall adopt a radial basis function with periodical patterns (RBFP), which appends a component that captures potential periodic patterns to traditional radial basis functions [44]:

$$\begin{aligned} &\mathsf{K}_P(\mathbf{x}_1(t_1), \mathbf{x}_2(t_2)) \\ &= \exp\big(-\gamma \|\mathsf{B}_L \mathbf{x}_1(t_1) - \mathsf{B}_L \mathbf{x}_2(t_2)\|_2^2 - \gamma_{\text{p}} d_P(t_1, t_2)^2 \big), \end{aligned} \tag{13}$$

where P is the period in the data (e.g., one signal cycle), γ and γ_{p} are weights associated with the radial basis func-

tion and the periodicity, respectively, and

$$d_P(t_1,t_2) \equiv \min\left\{|(t_1-t_2) \bmod P|, P-|(t_1-t_2) \bmod P|\right\} \tag{14}$$

is a temporal distance (modulo periodicity). The testing samples are related, similar to the training samples, as follows:

$$\mathbf{y}_{\text{te}}(t) = \langle \mathbf{W}, \phi(\mathsf{B}_L \mathbf{x}_{\text{te}}(t)) \rangle, \tag{15}$$

and the *joint matrix* in this kernelized setting is defined as:

$$\mathbf{Z} \equiv \begin{bmatrix} \mathbf{Y}_{\text{tr}} & \mathbf{Y}_{\text{te}} \\ \Phi(\mathbf{X}_{\text{tr}}) & \Phi(\mathbf{X}_{\text{te}}) \end{bmatrix} \in \mathbb{R}^{(n+h)\times(T_{\text{tr}}+T_{\text{te}})}, \tag{16}$$

where $\Phi : \mathbb{R}^{nL\times t} \to \mathbb{R}^{h\times t}$ simply applies ϕ to each column of its argument. That is

$$\Phi(\mathbf{X}_{\text{tr}}) \equiv \begin{bmatrix} \phi(\mathsf{B}_L \mathbf{x}_{\text{tr}}(1)) & \cdots & \phi(\mathsf{B}_L \mathbf{x}_{\text{tr}}(T_{\text{tr}})) \end{bmatrix} \tag{17}$$

and $\Phi(\mathbf{X}_{\text{te}})$ is defined in a similar way. To further simplify notation, we define the matrices $\Phi_{\text{tr}} \equiv \Phi(\mathbf{X}_{\text{tr}}) \in \mathbb{R}^{h\times T_{\text{tr}}}$ and $\Phi_{\text{te}} \equiv \Phi(\mathbf{X}_{\text{te}}) \in \mathbb{R}^{h\times T_{\text{te}}}$. The inner products of these matrices define the kernels that we will use below. For example we define the kernel $\mathbf{K}_{\text{tr,te}}$ as:

$$\mathbf{K}_{\text{tr,te}} \equiv \langle \Phi(\mathbf{X}_{\text{tr}}), \Phi(\mathbf{X}_{\text{te}}) \rangle = \begin{bmatrix} \mathsf{K}_P(\mathbf{x}_{\text{tr}}(1), \mathbf{x}_{\text{te}}(T_{\text{tr}})) & \cdots & \mathsf{K}_P(\mathbf{x}_{\text{tr}}(1), \mathbf{x}_{\text{te}}(T_{\text{tr}}+T_{\text{te}})) \\ \vdots & \ddots & \vdots \\ \mathsf{K}_P(\mathbf{x}_{\text{tr}}(T_{\text{tr}}), \mathbf{x}_{\text{te}}(T_{\text{tr}})) & \cdots & \mathsf{K}_P(\mathbf{x}_{\text{tr}}(T_{\text{tr}}), \mathbf{x}_{\text{te}}(T_{\text{tr}}+T_{\text{te}})) \end{bmatrix}. \tag{18}$$

From (11) and (15), we have that

$$\begin{bmatrix} \mathbf{Y}_{\text{tr}} & \mathbf{Y}_{\text{te}} \end{bmatrix} = \langle \mathbf{W}, \begin{bmatrix} \phi(\mathbf{X}_{\text{tr}}) & \phi(\mathbf{X}_{\text{te}}) \end{bmatrix} \rangle, \tag{19}$$

which (again) implies that $\mathbf{Z}$ is a low-rank matrix and that the prediction problem can be formulated as a low-rank matrix completion problem.

2.4 The Matrix Completion Problem

Our matrix completion problem seeks to find an estimate of the matrix $\mathbf{Z}$ in (16), denoted $\widehat{\mathbf{Z}}$. We seek to match $\mathbf{Z}$ on the kernelized training and testing inputs $\Phi(\mathbf{X}_{\text{tr}})$ and $\Phi(\mathbf{X}_{\text{te}})$, and the training outputs $\mathbf{Y}_{\text{te}}$. There are many ways to accomplish this; we will invoke the *principle of parsimony*. This results in a simple solution, one that is easy to interpret and extract insights from. To this end, our matrix completion problem seeks to find a low-rank approximation $\widehat{\mathbf{Z}}$ of the matrix $\mathbf{Z}$. Let $\mathsf{P}_\Omega : \mathbb{R}^{p\times q} \to \mathbb{R}^{p\times q}$ be a binary mask operator:

$$\left[\mathsf{P}_\Omega(\mathbf{M})\right]_{ij} = \mathbb{1}\{(i,j) \in \Omega\}\mathbf{M}_{ij}, \tag{20}$$

where $\mathbb{1}\{(i,j) \in \Omega\}$ is the indicator function, taking the value 1 if the condition $(i,j) \in \Omega$ is true and 0 otherwise. We utilize the binary mask to exclude the output testing matrix from $\mathbf{Z}$. That is, we define Ω as the set of indices corresponding to non-testing outputs, that is

$$\begin{aligned} \Omega \equiv &\{(i,j) : 1 \le i \le n+h, 1 \le j \le T_{\text{tr}}\} \\ &\cup \{(i,j) : n+1 \le i \le n+h, T_{\text{tr}}+1 \le j \le T_{\text{tr}}+T_{\text{te}}\}. \end{aligned} \tag{21}$$

Consequently,

$$\mathsf{P}_\Omega(\mathbf{Z}) = \begin{bmatrix} \mathbf{Y}_{\text{tr}} & \mathbf{0} \\ \Phi(\mathbf{X}_{\text{tr}}) & \Phi(\mathbf{X}_{\text{te}}) \end{bmatrix}. \tag{22}$$

The matrix completion problem is then formulated as a rank minimization problem:

$$\widehat{\mathbf{Z}} \equiv \underset{\mathbf{M}\in\mathbb{R}^{(n+h)\times(T_{\text{tr}}+T_{\text{te}})}}{\arg\min} \left\{\mathsf{rank}(\mathbf{M}) : \mathsf{P}_\Omega(\mathbf{M}-\mathbf{Z}) = \mathbf{0}\right\}. \tag{23}$$

The solution $\widehat{\mathbf{Z}}$ is interpreted as the lowest-rank matrix that matches $\mathbf{Z}$ exactly (via the constraint) in the entries corresponding to the training data, $\mathbf{Y}_{\text{tr}}$ and $\Phi(\mathbf{X}_{\text{tr}})$, and the input testing data, $\Phi(\mathbf{X}_{\text{te}})$. Rank minimization (23) is generally NP-hard but it was demonstrated in [5] that (23) can be solved exactly if certain sparsity conditions on $\mathbf{Z}$ hold, by using convex optimization techniques (via convex relaxation of the rank objective as shown below). The sparsity condition required is that $\mathbf{Z}$ be *sufficiently incoherent*, which is achieved when the number of observed entries in the matrix, $|\Omega|$ is $O(N^{1.2}\mathsf{rank}(\mathbf{Z})\log N)$, where $N \equiv \max\{n+h, T_{\text{tr}}+T_{\text{te}}\}$ in our context. This ensures faithful reconstruction. In our context such bounds can be easily ensured by simply using past days and longer lags (if necessary). A more useful application of this bound is to produce upper bound estimates of the rank of $\widehat{\mathbf{Z}}$, which will be needed to set up the problem (25) below. Henceforth, we will focus on solving a convex relaxation of the problem and refer readers to [5, 6] for further information on these bounds.

The main difficulty lies in minimizing the matrix rank. This can be observed by noting that $\mathsf{rank}(\mathbf{M}) = \|\sigma(\mathbf{M})\|_0$,

where $\sigma(\mathbf{M})$ is a vector of the singular values of $\mathbf{M}$ and $\|\cdot\|_0$ is the ℓ_0 pseudo-norm which counts the number of non-zero elements of its argument. We relax the objective function via a convex surrogate using the nuclear norm $\|\mathbf{M}\|_* = \mathsf{trace}(\sqrt{\mathbf{M}^\top \mathbf{M}}) = \|\sigma(\mathbf{M})\|_1$. (The ℓ_1 norm in this context is simply the sum of the singular values, since the singular values of a matrix are always non-negative.) This yields the relaxed problem:

$$\widehat{\mathbf{Z}} \equiv \underset{\mathbf{M}\in\mathbb{R}^{(n+h)\times(T_{\text{tr}}+T_{\text{te}})}}{\arg\min} \big\{\|\mathbf{M}\|_* : \mathsf{P}_\Omega(\mathbf{M}-\mathbf{Z}) = \mathbf{0}\big\}, \quad (24)$$

which is equivalent to the quadratic optimization problem [60, Lemma 5.1]

$$\begin{aligned}&\{\widehat{\mathbf{U}}, \widehat{\mathbf{V}}\}\\ &\equiv \underset{\substack{\mathbf{U}\in\mathbb{R}^{(n+h)\times r},\\ \mathbf{V}\in\mathbb{R}^{(T_{\text{tr}}+T_{\text{te}})\times r}}}{\arg\min} \big\{\|\mathbf{U}\|_{\text{F}}^2 + \|\mathbf{V}\|_{\text{F}}^2 : \mathsf{P}_\Omega(\mathbf{U}\mathbf{V}^\top - \mathbf{Z}) = \mathbf{0}\big\},\end{aligned} \quad (25)$$

where $\|\cdot\|_{\text{F}}$ is the Frobenius norm and r is chosen so that $r \geq \mathsf{rank}(\widehat{\mathbf{Z}})$. The latter formulation utilizes a bilinear representation $\mathbf{M} = \mathbf{U}\mathbf{V}^\top$ of the joint matrix and can be solved efficiently using alternating minimization techniques [31], an instance of which we shall develop in the next section. The latter formulation (25) can be expressed as an unconstrained optimization problem via the Lagrangian:

$$\begin{aligned}&\{\widehat{\mathbf{U}}, \widehat{\mathbf{V}}\}\\ &\equiv \underset{\substack{\mathbf{U}\in\mathbb{R}^{(n+h)\times r},\\ \mathbf{V}\in\mathbb{R}^{(T_{\text{tr}}+T_{\text{te}})\times r}}}{\arg\min} \|\mathsf{P}_\Omega(\mathbf{U}\mathbf{V}^\top - \mathbf{Z})\|_{\text{F}}^2 + \mu(\|\mathbf{U}\|_{\text{F}}^2 + \|\mathbf{V}\|_{\text{F}}^2),\end{aligned} \quad (26)$$

where $\mu > 0$ is a Lagrange multiplier.

Another simplification can be achieved by dividing each of the two matrices $\mathbf{U}$ and $\mathbf{V}$ into two blocks, a training block and a testing block:

$$\mathbf{U} = \begin{bmatrix}\mathbf{U}_{\text{tr}}\\ \mathbf{U}_{\text{te}}\end{bmatrix} \text{ and } \mathbf{V} = \begin{bmatrix}\mathbf{V}_{\text{tr}}\\ \mathbf{V}_{\text{te}}\end{bmatrix}, \quad (27)$$

where $\mathbf{U}_{\text{tr}} \in \mathbb{R}^{n\times r}$, $\mathbf{V}_{\text{tr}} \in \mathbb{R}^{T_{\text{tr}}\times r}$, $\mathbf{U}_{\text{te}} \in \mathbb{R}^{h\times r}$, and $\mathbf{V}_{\text{te}} \in \mathbb{R}^{T_{\text{te}}\times r}$ This decomposition allows us to bypass use of the binary mask P_Ω, simplifying the formulation further. We obtain the following optimization problem:

$$\begin{aligned}\{\widehat{\mathbf{U}}_{\text{tr}}, \widehat{\mathbf{U}}_{\text{te}}, \widehat{\mathbf{V}}_{\text{tr}}, \widehat{\mathbf{V}}_{\text{te}}\} \equiv \underset{\mathbf{U}_{\text{tr}},\mathbf{U}_{\text{te}},\mathbf{V}_{\text{tr}},\mathbf{V}_{\text{te}}}{\arg\min}\ & \|\mathbf{U}_{\text{tr}}\mathbf{V}_{\text{tr}}^\top - \mathbf{Y}_{\text{tr}}\|_{\text{F}}^2\\ &+ \|\mathbf{U}_{\text{te}}\mathbf{V}_{\text{tr}}^\top - \Phi_{\text{tr}}\|_{\text{F}}^2 + \|\mathbf{U}_{\text{te}}\mathbf{V}_{\text{te}}^\top - \Phi_{\text{te}}\|_{\text{F}}^2\\ &+ \mu(\|\mathbf{U}_{\text{tr}}\|_{\text{F}}^2 + \|\mathbf{U}_{\text{te}}\|_{\text{F}}^2 + \|\mathbf{V}_{\text{tr}}\|_{\text{F}}^2 + \|\mathbf{V}_{\text{te}}\|_{\text{F}}^2).\end{aligned} \quad (28)$$

Again, we do not prescribe the mapping functions, but shall prescribe kernels. For this purpose, we define $\mathbf{K}_{\text{tr,tr}} \equiv \langle\Phi_{\text{tr}}, \Phi_{\text{tr}}\rangle = \Phi_{\text{tr}}^\top\Phi_{\text{tr}}$, $\mathbf{K}_{\text{tr,te}} \equiv \langle\Phi_{\text{tr}}, \Phi_{\text{te}}\rangle = \Phi_{\text{tr}}^\top\Phi_{\text{te}}$, $\mathbf{K}_{\text{te,tr}} \equiv \langle\Phi_{\text{te}}, \Phi_{\text{tr}}\rangle = \Phi_{\text{te}}^\top\Phi_{\text{tr}}$, and $\mathbf{K}_{\text{te,te}} \equiv \langle\Phi_{\text{te}}, \Phi_{\text{te}}\rangle = \Phi_{\text{te}}^\top\Phi_{\text{te}}$.

To summarize, the key features of the proposed formulation (28), as they relate to traffic prediction, are:

1. The formulation inherits all the merits of conventional *vector* time series prediction methods, namely, vector auto-regressive models (VARs) [16, 66]. Specifically, our approach is capable of capturing and leveraging spatio-temporal dependencies.

2. The use of kernels allows us to capture nonlinearities (not possible with the VAR models in the literature), which one expects to see in traffic flow patterns, particularly in high-resolution traffic data.

3. The prediction results can be directly obtained without computing regression parameters. This is a particularly desirable feature for real-time implementations, where the parameters change throughout the day, from day-to-day, and throughout the year (seasonal effects).

4. The low rank approximation allows for insights to be easily extracted from the solution. The predictions produced are $\widehat{\mathbf{Y}}_{\text{te}} = \widehat{\mathbf{U}}_{\text{tr}}\widehat{\mathbf{V}}_{\text{te}}^\top$, where the n rows of $\widehat{\mathbf{U}}_{\text{tr}}$ capture the relationship between sensors (space) and r latent features, while the T_{te} rows of $\widehat{\mathbf{V}}_{\text{te}}$ capture the relationship between the prediction time steps and the r latent features. One can mine these matrices to find spatio-temporal correlations between different sensor and different times steps; see, e.g., [4].

Mathematically, the matrix completion problem (28) is not (in general) convex in all of its block coordinates $\{\mathbf{U}_{\text{tr}}, \mathbf{U}_{\text{te}}, \mathbf{V}_{\text{tr}}, \mathbf{V}_{\text{te}}\}$ simultaneously but it is convex in each of the blocks separately. (For intuition, readers may consider –as an exercise– convexity of the function $f(x, y) = (xy-1)^2$). Such problems are referred to as *block multi-convex* [78]. Let $F(\mathbf{U}_{\text{tr}}, \mathbf{U}_{\text{te}}, \mathbf{V}_{\text{tr}}, \mathbf{V}_{\text{te}})$ denote the objective function in (28):

$$\begin{aligned}F(\mathbf{U}_{\text{tr}}, \mathbf{U}_{\text{te}}, \mathbf{V}_{\text{tr}}, \mathbf{V}_{\text{te}}) &\equiv \|\mathbf{U}_{\text{tr}}\mathbf{V}_{\text{tr}}^\top - \mathbf{Y}_{\text{tr}}\|_{\text{F}}^2 + \|\mathbf{U}_{\text{te}}\mathbf{V}_{\text{tr}}^\top - \Phi_{\text{tr}}\|_{\text{F}}^2\\ &+ \|\mathbf{U}_{\text{te}}\mathbf{V}_{\text{te}}^\top - \Phi_{\text{te}}\|_{\text{F}}^2 + \mu(\|\mathbf{U}_{\text{tr}}\|_{\text{F}}^2 + \|\mathbf{U}_{\text{te}}\|_{\text{F}}^2 + \|\mathbf{V}_{\text{tr}}\|_{\text{F}}^2 + \|\mathbf{V}_{\text{te}}\|_{\text{F}}^2).\end{aligned} \quad (29)$$

We seek a solution $\{\overline{\mathbf{U}}_{\text{tr}}, \overline{\mathbf{U}}_{\text{te}}, \overline{\mathbf{V}}_{\text{tr}}, \overline{\mathbf{V}}_{\text{te}}\}$ for which the following variational inequalities hold for all $\mathbf{U}_{\text{tr}} \in \mathbb{R}^{n\times r}$, all $\mathbf{U}_{\text{te}} \in \mathbb{R}^{h\times r}$, all $\mathbf{V}_{\text{tr}} \in \mathbb{R}^{T_{\text{tr}}\times r}$, and all $\mathbf{V}_{\text{te}} \in \mathbb{R}^{T_{\text{te}}\times r}$:

$$\langle\partial_{\mathbf{U}_{\text{tr}}} F(\overline{\mathbf{U}}_{\text{tr}}, \overline{\mathbf{U}}_{\text{te}}, \overline{\mathbf{V}}_{\text{tr}}, \overline{\mathbf{V}}_{\text{te}}), \mathbf{U}_{\text{tr}} - \overline{\mathbf{U}}_{\text{tr}}\rangle \geq 0, \quad (30)$$

$$\left\langle \partial_{\mathbf{U}_{\text{te}}} F(\overline{\mathbf{U}}_{\text{tr}}, \overline{\mathbf{U}}_{\text{te}}, \overline{\mathbf{V}}_{\text{tr}}, \overline{\mathbf{V}}_{\text{te}}), \mathbf{U}_{\text{te}} - \overline{\mathbf{U}}_{\text{te}} \right\rangle \geq 0, \tag{31}$$

$$\left\langle \partial_{\mathbf{V}_{\text{tr}}} F(\overline{\mathbf{U}}_{\text{tr}}, \overline{\mathbf{U}}_{\text{te}}, \overline{\mathbf{V}}_{\text{tr}}, \overline{\mathbf{V}}_{\text{te}}), \mathbf{V}_{\text{tr}} - \overline{\mathbf{V}}_{\text{tr}} \right\rangle \geq 0, \tag{32}$$

and

$$\left\langle \partial_{\mathbf{V}_{\text{te}}} F(\overline{\mathbf{U}}_{\text{tr}}, \overline{\mathbf{U}}_{\text{te}}, \overline{\mathbf{V}}_{\text{tr}}, \overline{\mathbf{V}}_{\text{te}}), \mathbf{V}_{\text{te}} - \overline{\mathbf{V}}_{\text{te}} \right\rangle \geq 0, \tag{33}$$

where $\partial_{\mathbf{U}}$ denotes the partial derivative operator with respect to the matrix $\mathbf{U}$. This type of solution is referred to as a *block coordinate-wise minimizer* or a *Nash point* in the sense that one cannot further minimize F by changing any of the four block coordinates separately.

3 Block Coordinate Descent Algorithm

3.1 Block Coordinate Descent and Thresholding

The proposed block coordinate descent algorithm first updates the two $\mathbf{U}$ blocks, $\mathbf{U}_{\text{tr}}$ and $\mathbf{U}_{\text{te}}$, and then updates the two $\mathbf{V}$ blocks, $\mathbf{V}_{\text{tr}}$ and $\mathbf{V}_{\text{te}}$. The objective functions of the four sub-problems in iteration k are stated as

$$F_1^{[k]}(\mathbf{U}_{\text{tr}}) \equiv \|\mathbf{U}_{\text{tr}} \widehat{\mathbf{V}}_{\text{tr}}^{[k-1]\top} - \mathbf{Y}_{\text{tr}}\|_{\text{F}}^2 + 2\mu \|\mathbf{U}_{\text{tr}}\|_{\text{F}}^2, \tag{34}$$

$$F_2^{[k]}(\mathbf{U}_{\text{te}}) \equiv \|\mathbf{U}_{\text{te}} \widehat{\mathbf{V}}_{\text{tr}}^{[k-1]\top} - \Phi_{\text{tr}}\|_{\text{F}}^2 + \|\mathbf{U}_{\text{te}} \widehat{\mathbf{V}}_{\text{te}}^{[k-1]\top} - \Phi_{\text{te}}\|_{\text{F}}^2 + 2\mu \|\mathbf{U}_{\text{te}}\|_{\text{F}}^2, \tag{35}$$

$$F_3^{[k]}(\mathbf{V}_{\text{tr}}) \equiv \|\widehat{\mathbf{U}}_{\text{tr}}^{[k]} \mathbf{V}_{\text{tr}}^\top - \mathbf{Y}_{\text{tr}}\|_{\text{F}}^2 + \|\widehat{\mathbf{U}}_{\text{te}}^{[k]} \mathbf{V}_{\text{tr}}^\top - \Phi_{\text{tr}}\|_{\text{F}}^2 + 2\mu \|\mathbf{V}_{\text{tr}}\|_{\text{F}}^2, \tag{36}$$

and

$$F_4^{[k]}(\mathbf{V}_{\text{te}}) \equiv \|\widehat{\mathbf{U}}_{\text{te}}^{[k]} \mathbf{V}_{\text{te}}^\top - \Phi_{\text{te}}\|_{\text{F}}^2 + 2\mu \|\mathbf{V}_{\text{te}}\|_{\text{F}}^2. \tag{37}$$

The factors of two in the regularizers are unnecessary but we use them here to reduce clutter later on. The updates are obtained in closed form; the updated $\mathbf{U}$ blocks are given by:

$$\begin{aligned} \widehat{\mathbf{U}}_{\text{tr}}^{[k]} &= \underset{\mathbf{U}_{\text{tr}} \in \mathbb{R}^{n \times r}}{\arg\min}\, F_1^{[k]}(\mathbf{U}_{\text{tr}}) \\ &= \mathbf{Y}_{\text{tr}} \widehat{\mathbf{V}}_{\text{tr}}^{[k-1]} \left(\widehat{\mathbf{V}}_{\text{tr}}^{[k-1]\top} \widehat{\mathbf{V}}_{\text{tr}}^{[k-1]} + 2\mu \mathbf{I} \right)^{-1} \end{aligned} \tag{38}$$

and

$$\begin{aligned} \widehat{\mathbf{U}}_{\text{te}}^{[k]} &= \underset{\mathbf{U}_{\text{te}} \in \mathbb{R}^{h \times r}}{\arg\min}\, F_2^{[k]}(\mathbf{U}_{\text{te}}) = (\Phi_{\text{tr}} \widehat{\mathbf{V}}_{\text{tr}}^{[k-1]} + \Phi_{\text{te}} \widehat{\mathbf{V}}_{\text{te}}^{[k-1]}) \\ &\times (\widehat{\mathbf{V}}_{\text{tr}}^{[k-1]\top} \widehat{\mathbf{V}}_{\text{tr}}^{[k-1]} + \widehat{\mathbf{V}}_{\text{te}}^{[k-1]\top} \widehat{\mathbf{V}}_{\text{te}}^{[k-1]} + 2\mu \mathbf{I})^{-1}. \end{aligned} \tag{39}$$

The updated $\mathbf{V}$ blocks are then given by:

$$\begin{aligned} \widehat{\mathbf{V}}_{\text{tr}}^{[k]} &= \underset{\mathbf{V}_{\text{tr}} \in \mathbb{R}^{T_{\text{tr}} \times r}}{\arg\min}\, F_3^{[k]}(\mathbf{V}_{\text{tr}}) \\ &= (\mathbf{Y}_{\text{tr}}^\top \widehat{\mathbf{U}}_{\text{tr}}^{[k]} + \Phi_{\text{tr}}^\top \widehat{\mathbf{U}}_{\text{te}}^{[k]})(\widehat{\mathbf{U}}_{\text{te}}^{[k]\top} \widehat{\mathbf{U}}_{\text{te}}^{[k]} + \widehat{\mathbf{U}}_{\text{tr}}^{[k]\top} \widehat{\mathbf{U}}_{\text{tr}}^{[k]} + 2\mu \mathbf{I})^{-1} \end{aligned} \tag{40}$$

and

$$\widehat{\mathbf{V}}_{\text{te}}^{[k]} = \underset{\mathbf{V}_{\text{te}} \in \mathbb{R}^{T_{\text{te}} \times r}}{\arg\min}\, F_4^{[k]}(\mathbf{V}_{\text{te}}) = \Phi_{\text{te}}^\top \widehat{\mathbf{U}}_{\text{te}}^{[k]} (\widehat{\mathbf{U}}_{\text{te}}^{[k]\top} \widehat{\mathbf{U}}_{\text{te}}^{[k]} + 2\mu \mathbf{I})^{-1}. \tag{41}$$

Since Φ_{tr} and Φ_{te} are unknown, $\widehat{\mathbf{U}}_{\text{te}}^{[k]}$ cannot be calculated explicitly. However, this calculation is not required to produce a prediction: $\widehat{\mathbf{Y}}_{\text{te}} \equiv \widehat{\mathbf{U}}_{\text{tr}} \widehat{\mathbf{V}}_{\text{te}}^\top$. To produce this estimate, in each iteration k we need to be able to calculate $\widehat{\mathbf{U}}_{\text{tr}}^{[k]}$ and $\widehat{\mathbf{V}}_{\text{te}}^{[k]\top}$. To calculate these quantities, we need (i) $\widehat{\mathbf{V}}_{\text{tr}}^{[k-1]}$, (ii) $\Phi_{\text{te}}^\top \widehat{\mathbf{U}}_{\text{te}}^{[k]}$, and (iii) $\widehat{\mathbf{U}}_{\text{te}}^{[k]\top} \widehat{\mathbf{U}}_{\text{te}}^{[k]}$. We start with (iii): assuming that all required calculations have been performed for iteration $k-1$, we have from (39) that

$$\begin{aligned} \widehat{\mathbf{U}}_{\text{te}}^{[k]\top} \widehat{\mathbf{U}}_{\text{te}}^{[k]} &= (\widehat{\mathbf{V}}_{\text{tr}}^{[k-1]\top} \widehat{\mathbf{V}}_{\text{tr}}^{[k-1]} + \widehat{\mathbf{V}}_{\text{te}}^{[k-1]\top} \widehat{\mathbf{V}}_{\text{te}}^{[k-1]} + \mu \mathbf{I})^{-1} \\ &\times \Big(\widehat{\mathbf{V}}_{\text{tr}}^{[k-1]\top} \mathbf{K}_{\text{tr,tr}} \widehat{\mathbf{V}}_{\text{tr}}^{[k-1]} + \widehat{\mathbf{V}}_{\text{te}}^{[k-1]\top} \mathbf{K}_{\text{te,tr}} \widehat{\mathbf{V}}_{\text{tr}}^{[k-1]} \\ &+ \widehat{\mathbf{V}}_{\text{tr}}^{[k-1]\top} \mathbf{K}_{\text{tr,te}} \widehat{\mathbf{V}}_{\text{te}}^{[k-1]} + \widehat{\mathbf{V}}_{\text{te}}^{[k-1]\top} \mathbf{K}_{\text{te,te}} \widehat{\mathbf{V}}_{\text{te}}^{[k-1]} \Big) \\ &\times (\widehat{\mathbf{V}}_{\text{tr}}^{[k-1]\top} \widehat{\mathbf{V}}_{\text{tr}}^{[k-1]} + \widehat{\mathbf{V}}_{\text{te}}^{[k-1]\top} \widehat{\mathbf{V}}_{\text{te}}^{[k-1]} + \mu \mathbf{I})^{-1}, \end{aligned} \tag{42}$$

which only involves known quantities. Similarly, for (ii) we have from (39) that

$$\begin{aligned} \Phi_{\text{te}}^\top \widehat{\mathbf{U}}_{\text{te}}^{[k]} &= (\mathbf{K}_{\text{te,tr}} \widehat{\mathbf{V}}_{\text{tr}}^{[k-1]} + \mathbf{K}_{\text{te,te}} \widehat{\mathbf{V}}_{\text{te}}^{[k-1]}) \\ &\times (\widehat{\mathbf{V}}_{\text{tr}}^{[k-1]\top} \widehat{\mathbf{V}}_{\text{tr}}^{[k-1]} + \widehat{\mathbf{V}}_{\text{te}}^{[k-1]\top} \widehat{\mathbf{V}}_{\text{te}}^{[k-1]} + \mu \mathbf{I})^{-1}, \end{aligned} \tag{43}$$

which also only involves known quantities. For (i), we need to be able to calculate $\Phi_{\text{tr}}^\top \widehat{\mathbf{U}}_{\text{te}}^{[k-1]}$ and $\widehat{\mathbf{U}}_{\text{te}}^{[k-1]\top} \widehat{\mathbf{U}}_{\text{te}}^{[k-1]}$. These quantities also only involve known quantities; they are the same as (43) and (42), respectively (since k is arbitrary). Note that this also allows for calculating training estimates $\widehat{\mathbf{Y}}_{\text{tr}}^{[k]} \equiv \widehat{\mathbf{U}}_{\text{tr}}^{[k]} \widehat{\mathbf{V}}_{\text{tr}}^{[k]\top}$ in each iteration, which we will need in our ensemble approach presented below. The steps involved in performing a single update are summarized in Alg. 1.

Soft thresholding: The predictions $\widehat{\mathbf{Y}}$ produced by Algorithm 1 will produce values that are not necessarily restricted to $\{0, 1\}$. For traffic signal operations, whether a vehicle is present or not needs to be decided before the control decisions can be made. For this reason, we employ the thresholding procedure given in Algorithm 2.

Algorithm 1: Block Coordinate Descent

Data: $\mathbf{Y}_{\text{tr}}$, $\mathbf{K}_{\text{tr,tr}}$, $\mathbf{K}_{\text{tr,te}}$, $\mathbf{K}_{\text{te,tr}}$, $\mathbf{K}_{\text{te,te}}$, $\widehat{\mathbf{U}}_{\text{tr}}^{[0]}$, $\widehat{\mathbf{V}}_{\text{tr}}^{[0]}$, $\widehat{\mathbf{V}}_{\text{te}}^{[0]}$
Result: $\widehat{\mathbf{Y}}_{\text{tr}}$, $\widehat{\mathbf{Y}}_{\text{te}}$

Initialize: $k \leftarrow\!\!\shortmid 1$
while *stopping criterion not met* **do**
  $\widehat{\mathbf{U}}_{\text{tr}}^{[k]} \leftarrow\!\!\shortmid \mathbf{Y}_{\text{tr}}\widehat{\mathbf{V}}_{\text{tr}}^{[k-1]}(\widehat{\mathbf{V}}_{\text{tr}}^{[k-1]\top}\widehat{\mathbf{V}}_{\text{tr}}^{[k-1]} + 2\mu\mathbf{I})^{-1}$
  $\mathbf{C}_1 \leftarrow\!\!\shortmid (\widehat{\mathbf{V}}_{\text{tr}}^{[k-1]\top}\widehat{\mathbf{V}}_{\text{tr}}^{[k-1]} + \widehat{\mathbf{V}}_{\text{te}}^{[k-1]\top}\widehat{\mathbf{V}}_{\text{te}}^{[k-1]} + 2\mu\mathbf{I})^{-1}$
  $\mathbf{C}_2 \leftarrow\!\!\shortmid \widehat{\mathbf{V}}_{\text{tr}}^{[k-1]\top}\mathbf{K}_{\text{tr,tr}}\widehat{\mathbf{V}}_{\text{tr}}^{[k-1]} + \widehat{\mathbf{V}}_{\text{te}}^{[k-1]\top}\mathbf{K}_{\text{te,tr}}\widehat{\mathbf{V}}_{\text{tr}}^{[k-1]}$
  $\mathbf{C}_3 \leftarrow\!\!\shortmid \widehat{\mathbf{V}}_{\text{tr}}^{[k-1]\top}\mathbf{K}_{\text{tr,te}}\widehat{\mathbf{V}}_{\text{te}}^{[k-1]} + \widehat{\mathbf{V}}_{\text{te}}^{[k-1]\top}\mathbf{K}_{\text{te,te}}\widehat{\mathbf{V}}_{\text{te}}^{[k-1]}$
  $\Phi_{\text{te}}^{\top}\widehat{\mathbf{U}}_{\text{te}}^{[k]} \leftarrow\!\!\shortmid (\mathbf{K}_{\text{te,tr}}\widehat{\mathbf{V}}_{\text{tr}}^{[k-1]} + \mathbf{K}_{\text{te,te}}\widehat{\mathbf{V}}_{\text{te}}^{[k-1]})\mathbf{C}_1$
  $\widehat{\mathbf{U}}_{\text{te}}^{[k]\top}\widehat{\mathbf{U}}_{\text{te}}^{[k]} \leftarrow\!\!\shortmid \mathbf{C}_1(\mathbf{C}_2 + \mathbf{C}_3)\mathbf{C}_1$
  $\mathbf{C}_4 \leftarrow\!\!\shortmid (\widehat{\mathbf{U}}_{\text{te}}^{[k]\top}\widehat{\mathbf{U}}_{\text{te}}^{[k]} + \widehat{\mathbf{U}}_{\text{tr}}^{[k]\top}\widehat{\mathbf{U}}_{\text{tr}}^{[k]} + 2\mu\mathbf{I})^{-1}$
  $\widehat{\mathbf{V}}_{\text{tr}}^{[k]} \leftarrow\!\!\shortmid (\mathbf{Y}_{\text{tr}}^{\top}\widehat{\mathbf{U}}_{\text{tr}}^{[k]} + \Phi_{\text{tr}}^{\top}\widehat{\mathbf{U}}_{\text{te}}^{[k]})\mathbf{C}_4$
  $\widehat{\mathbf{V}}_{\text{te}}^{[k]} \leftarrow\!\!\shortmid \Phi_{\text{te}}^{\top}\widehat{\mathbf{U}}_{\text{te}}^{[k]}(\widehat{\mathbf{U}}_{\text{te}}^{[k]\top}\widehat{\mathbf{U}}_{\text{te}}^{[k]} + 2\mu\mathbf{I})^{-1}$
  $k \leftarrow\!\!\shortmid k+1$
end
$\widehat{\mathbf{Y}}_{\text{tr}} \leftarrow\!\!\shortmid \widehat{\mathbf{U}}_{\text{tr}}^{[k]}\widehat{\mathbf{V}}_{\text{tr}}^{[k]\top}$ and $\widehat{\mathbf{Y}}_{\text{te}} \leftarrow\!\!\shortmid \widehat{\mathbf{U}}_{\text{tr}}^{[k]}\widehat{\mathbf{V}}_{\text{te}}^{[k]\top}$

Algorithm 2: Threshold Learning

Data: Predicted output matrices $\widehat{\mathbf{Y}}_{\text{tr}}$ and $\widehat{\mathbf{Y}}_{\text{te}}$, true training output $\mathbf{Y}_{\text{tr}}$
Result: Set of thresholds, one per row $\{\tau_j\}_{j\geq 1}$

for *each row j of* $\widehat{\mathbf{Y}}_{\text{tr}}$ **do**
  Store row j of $\widehat{\mathbf{Y}}_{\text{tr}}$ in a separate vector $\mathbf{y} \leftarrow\!\!\shortmid \widehat{\mathbf{Y}}_{j,\text{tr}}$
  Sort $\mathbf{y}$ in ascending order $\mathbf{y} \leftarrow\!\!\shortmid \text{sort}(\mathbf{y})$
  $c_j \leftarrow\!\!\shortmid \|\mathbf{y}\|_0$ `// number of non-zero elements in y`
  for $i \geq c_j$ **do**
    $\tau_{j,i} \leftarrow\!\!\shortmid \mathbf{y}_i$
    $\widehat{\mathbf{Y}}_{j,m,\text{tr}} \leftarrow\!\!\shortmid \mathbb{1}\{\widehat{\mathbf{Y}}_{j,m,\text{tr}} \geq \tau_{j,i}\}$ for all $m \geq 1$
    Calculate the error $e_{j,i} \leftarrow\!\!\shortmid \|\widehat{\mathbf{Y}}_{\text{tr}} - \mathbf{Y}_{\text{tr}}\|_0$
  end
  Set $\tau_j \leftarrow\!\!\shortmid \tau_{j,i^*}$, where $i^* \leftarrow\!\!\shortmid \arg\min_{i\geq c_j} e_{j,i}$
end

Since our approach is data-driven, it can be applied to prediction problems involving continuous data; in such cases, thresholding is not required. Our thresholding procedure can simply be described as one that produces cut-offs for each of the n network sensors separately. This resembles the operation performed by intersection controllers, which translate inductance drops at the sensors to on-off signals and this is tuned for each of the sensors separately. Our thresholding algorithm chooses cut-offs, denoted $\{\tau_j\}_{j=1}^{n}$ that result in the lowest training error. We note that thresholding as a post-processing step is common in classification and prediction problems in neural networks [52] and logistic regression based methods [17, 20] to project non-binary solutions to binary values.

3.2 Sublinear Convergence to a Block Coordinate-Wise Minimizer

In this section, we demonstrate that the block coordinate descent algorithm above converges to a coordinate-wise minimizer (a Nash point). The convergence follows from the strong convexity of the objective functions of the sub-problems as we demonstrate in Lemma 1 below. We will also prove that the convergence rate is sub-linear, and we will require estimates for the Lipschitz bounds on the gradients of our objective functions. Hence, before stating our results formally and proving them, we will next demonstrate that the sub-problems have strongly convex objective functions and provide Lipschitz bounds on their gradients.

Strong Convexity: A function $f : \mathbb{R}^{p\times q} \to \mathbb{R}$ is λ-strongly convex, for some $\lambda > 0$, if for all $\mathbf{G}_1, \mathbf{G}_2 \in \mathbb{R}^{p\times q}$

$$f(\mathbf{G}_1) - f(\mathbf{G}_2) \geq \langle \partial_{\mathbf{G}} f(\mathbf{G}_2), \mathbf{G}_1 - \mathbf{G}_2\rangle + \frac{\lambda}{2}\|\mathbf{G}_1 - \mathbf{G}_2\|_{\text{F}}^2 \tag{44}$$

or, equivalently,

$$\langle \partial_{\mathbf{G}} f(\mathbf{G}_1) - \partial_{\mathbf{G}} f(\mathbf{G}_2), \mathbf{G}_1 - \mathbf{G}_2\rangle \geq \lambda\|\mathbf{G}_1 - \mathbf{G}_2\|_{\text{F}}^2. \tag{45}$$

We will use definition (45) to demonstrate the strong convexity of $F_1^{[k]}$, $F_2^{[k]}$, $F_3^{[k]}$, and $F_4^{[k]}$, defined above, for any $k \geq 1$. We begin with $F_1^{[k]}$: for any $\mathbf{U}_1, \mathbf{U}_2 \in \mathbb{R}^{n\times r}$, we have that

$$\begin{aligned}&\langle \partial_{\mathbf{U}} F_1^{[k]}(\mathbf{U}_1) - \partial_{\mathbf{U}} F_1^{[k]}(\mathbf{U}_2), \mathbf{U}_1 - \mathbf{U}_2\rangle \\ &\quad= \langle 2(\mathbf{U}_1 - \mathbf{U}_2)(\widehat{\mathbf{V}}_{\text{tr}}^{[k-1]\top}\widehat{\mathbf{V}}_{\text{tr}}^{[k-1]} + \mu\mathbf{I}), \mathbf{U}_1 - \mathbf{U}_2\rangle \\ &= \text{trace}\big(2(\widehat{\mathbf{V}}_{\text{tr}}^{[k-1]\top}\widehat{\mathbf{V}}_{\text{tr}}^{[k-1]} + \mu\mathbf{I})(\mathbf{U}_1 - \mathbf{U}_2)^{\top}(\mathbf{U}_1 - \mathbf{U}_2)\big).\end{aligned} \tag{46}$$

Since $\mu > 0$, we have that $2\widehat{\mathbf{V}}_{\text{tr}}^{[k-1]\top}\widehat{\mathbf{V}}_{\text{tr}}^{[k-1]} + 2\mu\mathbf{I}$ is positive definite with positive eigenvalues. In particular the smallest eigenvalue, defined as

$$\underline{\lambda}_1^{[k]} \equiv \min_{1\leq i\leq r} \lambda(2\widehat{\mathbf{V}}_{\text{tr}}^{[k-1]\top}\widehat{\mathbf{V}}_{\text{tr}}^{[k-1]} + 2\mu\mathbf{I}) \tag{47}$$

is positive, i.e., $\underline{\lambda}_1^{[k]} > 0$, where $\lambda(\mathbf{M})$ is a vector of eigenvalues of $\mathbf{M}$. It follows immediately that (see, e.g., [13])

$$\langle \partial_{\mathbf{U}} F_1^{[k]}(\mathbf{U}_1) - \partial_{\mathbf{U}} F_1^{[k]}(\mathbf{U}_2), \mathbf{U}_1 - \mathbf{U}_2\rangle \geq \underline{\lambda}_1^{[k]}\|\mathbf{U}_1 - \mathbf{U}_2\|_{\text{F}}^2. \tag{48}$$

Hence, $F_1^{[k]}$ is $\underline{\lambda}_1^{[k]}$-strongly convex. Similarly,

$$\langle \partial_{\mathbf{U}} F_2^{[k]}(\mathbf{U}_1) - \partial_{\mathbf{U}} F_2^{[k]}(\mathbf{U}_2), \mathbf{U}_1 - \mathbf{U}_2 \rangle \geq \underline{\lambda}_2^{[k]} \|\mathbf{U}_1 - \mathbf{U}_2\|_{\mathrm{F}}^2 \tag{49}$$

for any $\mathbf{U}_1, \mathbf{U}_2 \in \mathbb{R}^{h \times r}$, where

$$\underline{\lambda}_2^{[k]} \equiv \min_{1 \leq i \leq r} \lambda(2\widehat{\mathbf{V}}_{\mathrm{tr}}^{[k-1]\top}\widehat{\mathbf{V}}_{\mathrm{tr}}^{[k-1]} + 2\widehat{\mathbf{V}}_{\mathrm{te}}^{[k-1]\top}\widehat{\mathbf{V}}_{\mathrm{te}}^{[k-1]} + 2\mu\mathbf{I}) \tag{50}$$

is the smallest eigenvalue of the matrix $2\widehat{\mathbf{V}}_{\mathrm{tr}}^{[k-1]\top}\widehat{\mathbf{V}}_{\mathrm{tr}}^{[k-1]} + 2\widehat{\mathbf{V}}_{\mathrm{te}}^{[k-1]\top}\widehat{\mathbf{V}}_{\mathrm{te}}^{[k-1]} + 2\mu\mathbf{I}$ and $\underline{\lambda}_2^{[k]} > 0$. Hence, $F_2^{[k]}$ is $\underline{\lambda}_2^{[k]}$-strongly convex. It can be similarly shown that $F_3^{[k]}$ and $F_4^{[k]}$ are $\underline{\lambda}_3^{[k]}$-strongly convex and $\underline{\lambda}_4^{[k]}$-strongly convex, respectively, where

$$\underline{\lambda}_3^{[k]} \equiv \min_{1 \leq i \leq r} \lambda(2\widehat{\mathbf{U}}_{\mathrm{tr}}^{[k]\top}\widehat{\mathbf{U}}_{\mathrm{tr}}^{[k]} + 2\widehat{\mathbf{U}}_{\mathrm{te}}^{[k]\top}\widehat{\mathbf{U}}_{\mathrm{te}}^{[k]} + 2\mu\mathbf{I}) \tag{51}$$

and

$$\underline{\lambda}_4^{[k]} \equiv \min_{1 \leq i \leq r} \lambda(2\widehat{\mathbf{U}}_{\mathrm{te}}^{[k]\top}\widehat{\mathbf{U}}_{\mathrm{te}}^{[k]} + 2\mu\mathbf{I}). \tag{52}$$

Lipschitz Bounds: We will now establish that the Lipschitz constants for $\partial_{\mathbf{U}} F_1^{[k]}$, $\partial_{\mathbf{U}} F_2^{[k]}$, $\partial_{\mathbf{V}} F_3^{[k]}$, and $\partial_{\mathbf{V}} F_4^{[k]}$ are the largest eigenvalues of the matrices above. For any $\mathbf{U}_1, \mathbf{U}_2 \in \mathbb{R}^{n \times r}$

$$\begin{aligned} &\|\partial_{\mathbf{U}} F_1^{[k]}(\mathbf{U}_1) - \partial_{\mathbf{U}} F_1^{[k]}(\mathbf{U}_2)\|_{\mathrm{F}}^2 \\ &\quad = \|2(\mathbf{U}_1 - \mathbf{U}_2)(\widehat{\mathbf{V}}_{\mathrm{tr}}^{[k-1]\top}\widehat{\mathbf{V}}_{\mathrm{tr}}^{[k-1]} + \mu\mathbf{I})\|_{\mathrm{F}}^2 \\ &\quad \leq (\overline{\lambda}_1^{[k]})^2 \|\mathbf{U}_1 - \mathbf{U}_2\|_{\mathrm{F}}^2, \end{aligned} \tag{53}$$

where

$$\overline{\lambda}_1^{[k]} \equiv \max_{1 \leq i \leq r} \lambda(2\widehat{\mathbf{V}}_{\mathrm{tr}}^{[k-1]\top}\widehat{\mathbf{V}}_{\mathrm{tr}}^{[k-1]} + 2\mu\mathbf{I}). \tag{54}$$

The inequality (53) follows from the bounds in [13] and we have the Lipschitz condition:

$$\|\partial_{\mathbf{U}} F_1^{[k]}(\mathbf{U}_1) - \partial_{\mathbf{U}} F_1^{[k]}(\mathbf{U}_2)\|_{\mathrm{F}}^2 \leq (\overline{\lambda}_1^{[k]})^2 \|\mathbf{U}_1 - \mathbf{U}_2\|_{\mathrm{F}}^2. \tag{55}$$

We can similarly establish the Lipschitz conditions

$$\|\partial_{\mathbf{U}} F_2^{[k]}(\mathbf{U}_1) - \partial_{\mathbf{U}} F_2^{[k]}(\mathbf{U}_2)\|_{\mathrm{F}}^2 \leq (\overline{\lambda}_2^{[k]})^2 \|\mathbf{U}_1 - \mathbf{U}_2\|_{\mathrm{F}}^2, \tag{56}$$

$$\|\partial_{\mathbf{V}} F_3^{[k]}(\mathbf{V}_1) - \partial_{\mathbf{V}} F_3^{[k]}(\mathbf{V}_2)\|_{\mathrm{F}}^2 \leq (\overline{\lambda}_3^{[k]})^2 \|\mathbf{V}_1 - \mathbf{V}_2\|_{\mathrm{F}}^2, \tag{57}$$

and

$$\|\partial_{\mathbf{V}} F_4^{[k]}(\mathbf{V}_1) - \partial_{\mathbf{V}} F_4^{[k]}(\mathbf{V}_2)\|_{\mathrm{F}}^2 \leq (\overline{\lambda}_4^{[k]})^2 \|\mathbf{V}_1 - \mathbf{V}_2\|_{\mathrm{F}}^2, \tag{58}$$

where the constants are given by

$$\overline{\lambda}_2^{[k]} \equiv \max_{1 \leq i \leq r} \lambda(2\widehat{\mathbf{V}}_{\mathrm{tr}}^{[k-1]\top}\widehat{\mathbf{V}}_{\mathrm{tr}}^{[k-1]} + 2\widehat{\mathbf{V}}_{\mathrm{te}}^{[k-1]\top}\widehat{\mathbf{V}}_{\mathrm{te}}^{[k-1]} + 2\mu\mathbf{I}), \tag{59}$$

$$\overline{\lambda}_3^{[k]} \equiv \max_{1 \leq i \leq r} \lambda(2\widehat{\mathbf{U}}_{\mathrm{tr}}^{[k]\top}\widehat{\mathbf{U}}_{\mathrm{tr}}^{[k]} + 2\widehat{\mathbf{U}}_{\mathrm{te}}^{[k]\top}\widehat{\mathbf{U}}_{\mathrm{te}}^{[k]} + 2\mu\mathbf{I}) \tag{60}$$

and

$$\overline{\lambda}_4^{[k]} \equiv \max_{1 \leq i \leq r} \lambda(2\widehat{\mathbf{U}}_{\mathrm{te}}^{[k]\top}\widehat{\mathbf{U}}_{\mathrm{te}}^{[k]} + 2\mu\mathbf{I}). \tag{61}$$

The smallest eigenvalues, $\{\underline{\lambda}_i^{[k]}\}_{i=1}^4$ are all bounded from below by 2μ for all k, which we need in the sequel. We can also produce estimates of the largest eigenvalues, $\{\overline{\lambda}_i^{[k]}\}_{i=1}^4$, by appeal to the Perron-Frobenius theorem. We denote these upper bounds by $L_1 \geq \overline{\lambda}_1^{[k]}$, $L_2 \geq \overline{\lambda}_2^{[k]}$, $L_3 \geq \overline{\lambda}_2^{[k]}$, and $L_4 \geq \overline{\lambda}_4^{[k]}$ (for all k) and define the upper bound $L_{\max} \equiv \max\{L_1^2, L_2^2, L_3^2, L_4^2\}$.

Lemma 1 (Algorithm Convergence). *Let F, $F_1^{[k]}$, $F_2^{[k]}$, $F_3^{[k]}$, and $F_4^{[k]}$ be as defined in (29) and (34)–(37), and let the block updates $\widehat{\mathbf{U}}_{\mathrm{tr}}^{[k]}$, $\widehat{\mathbf{U}}_{\mathrm{te}}^{[k]}$, $\widehat{\mathbf{V}}_{\mathrm{tr}}^{[k]}$, and $\widehat{\mathbf{V}}_{\mathrm{te}}^{[k]}$ be as given in (38)–(41). Assume that the initial solution $\{\widehat{\mathbf{U}}_{\mathrm{tr}}^{[0]}, \widehat{\mathbf{U}}_{\mathrm{te}}^{[0]}, \widehat{\mathbf{V}}_{\mathrm{tr}}^{[0]}, \widehat{\mathbf{V}}_{\mathrm{te}}^{[0]}\}$ is such that $F(\widehat{\mathbf{U}}_{\mathrm{tr}}^{[0]}, \widehat{\mathbf{U}}_{\mathrm{te}}^{[0]}, \widehat{\mathbf{V}}_{\mathrm{tr}}^{[0]}, \widehat{\mathbf{V}}_{\mathrm{te}}^{[0]}) < \infty$. Then*

$$\begin{aligned} \lim_{K \to \infty} \sum_{k=1}^{K} \Big(&\|\widehat{\mathbf{U}}_{\mathrm{tr}}^{[k-1]} - \widehat{\mathbf{U}}_{\mathrm{tr}}^{[k]}\|_{\mathrm{F}}^2 + \|\widehat{\mathbf{U}}_{\mathrm{te}}^{[k-1]} - \widehat{\mathbf{U}}_{\mathrm{te}}^{[k]}\|_{\mathrm{F}}^2 \\ &+ \|\widehat{\mathbf{V}}_{\mathrm{tr}}^{[k-1]} - \widehat{\mathbf{V}}_{\mathrm{tr}}^{[k]}\|_{\mathrm{F}}^2 + \|\widehat{\mathbf{V}}_{\mathrm{te}}^{[k-1]} - \widehat{\mathbf{V}}_{\mathrm{te}}^{[k]}\|_{\mathrm{F}}^2 \Big) < \infty. \end{aligned} \tag{62}$$

Proof. Noting that $\widehat{\mathbf{U}}_{\mathrm{tr}}^{[k]}$ and $\widehat{\mathbf{U}}_{\mathrm{te}}^{[k]}$ minimize $F_1^{[k]}$ and $F_2^{[k]}$, respectively, we readily have that

$$\begin{aligned} &2F(\widehat{\mathbf{U}}_{\mathrm{tr}}^{[k-1]}, \widehat{\mathbf{U}}_{\mathrm{te}}^{[k-1]}, \widehat{\mathbf{V}}_{\mathrm{tr}}^{[k-1]}, \widehat{\mathbf{V}}_{\mathrm{te}}^{[k-1]}) - F_1^{[k]}(\widehat{\mathbf{U}}_{\mathrm{tr}}^{[k-1]}) \\ &\quad - F_2^{[k]}(\widehat{\mathbf{U}}_{\mathrm{te}}^{[k-1]}) - F_3^{[k]}(\widehat{\mathbf{V}}_{\mathrm{tr}}^{[k-1]}) - F_4^{[k]}(\widehat{\mathbf{V}}_{\mathrm{te}}^{[k-1]}) \geq 0. \end{aligned} \tag{63}$$

Similarly, since $\widehat{\mathbf{V}}_{\mathrm{tr}}^{[k]}$ and $\widehat{\mathbf{V}}_{\mathrm{te}}^{[k]}$ minimize $F_3^{[k]}$ and $F_4^{[k]}$, respectively, we have that

$$\begin{aligned} &2F(\widehat{\mathbf{U}}_{\mathrm{tr}}^{[k]}, \widehat{\mathbf{U}}_{\mathrm{te}}^{[k]}, \widehat{\mathbf{V}}_{\mathrm{tr}}^{[k]}, \widehat{\mathbf{V}}_{\mathrm{te}}^{[k]}) - F_1^{[k]}(\widehat{\mathbf{U}}_{\mathrm{tr}}^{[k]}) \\ &\quad - F_2^{[k]}(\widehat{\mathbf{U}}_{\mathrm{te}}^{[k]}) - F_3^{[k]}(\widehat{\mathbf{V}}_{\mathrm{tr}}^{[k]}) - F_4^{[k]}(\widehat{\mathbf{V}}_{\mathrm{te}}^{[k]}) \leq 0. \end{aligned} \tag{64}$$

Then, for all $k \geq 1$

$$\begin{aligned} &F(\widehat{\mathbf{U}}_{\mathrm{tr}}^{[k-1]}, \widehat{\mathbf{U}}_{\mathrm{te}}^{[k-1]}, \widehat{\mathbf{V}}_{\mathrm{tr}}^{[k-1]}, \widehat{\mathbf{V}}_{\mathrm{te}}^{[k-1]}) - F(\widehat{\mathbf{U}}_{\mathrm{tr}}^{[k]}, \widehat{\mathbf{U}}_{\mathrm{te}}^{[k]}, \widehat{\mathbf{V}}_{\mathrm{tr}}^{[k]}, \widehat{\mathbf{V}}_{\mathrm{te}}^{[k]}) \\ &\geq \frac{1}{2}\Big(F_1^{[k]}(\widehat{\mathbf{U}}_{\mathrm{tr}}^{[k-1]}) - F_1^{[k]}(\widehat{\mathbf{U}}_{\mathrm{tr}}^{[k]}) + F_2^{[k]}(\widehat{\mathbf{U}}_{\mathrm{te}}^{[k-1]}) - F_2^{[k]}(\widehat{\mathbf{U}}_{\mathrm{te}}^{[k]}) \\ &+ F_3^{[k]}(\widehat{\mathbf{V}}_{\mathrm{tr}}^{[k-1]}) - F_3^{[k]}(\widehat{\mathbf{V}}_{\mathrm{tr}}^{[k]}) + F_4^{[k]}(\widehat{\mathbf{V}}_{\mathrm{te}}^{[k-1]}) - F_4^{[k]}(\widehat{\mathbf{V}}_{\mathrm{te}}^{[k]}) \Big). \end{aligned} \tag{65}$$

Since $F_1^{[k]}$ is strongly convex, we have that

$$\frac{1}{2}\big(F_1^{[k]}(\widehat{\mathbf{U}}_{\text{tr}}^{[k-1]}) - F_1^{[k]}(\widehat{\mathbf{U}}_{\text{tr}}^{[k]})\big) \geq \frac{1}{2}\langle \partial_{\mathbf{U}} F_1^{[k]}(\widehat{\mathbf{U}}_{\text{tr}}^{[k]}), \widehat{\mathbf{U}}_{\text{tr}}^{[k-1]} - \widehat{\mathbf{U}}_{\text{tr}}^{[k]}\rangle + \frac{\underline{\lambda}_1^{[k]}}{4}\|\widehat{\mathbf{U}}_{\text{tr}}^{[k-1]} - \widehat{\mathbf{U}}_{\text{tr}}^{[k]}\|_{\text{F}}^2. \quad (66)$$

Since $\partial_{\mathbf{U}} F_1^{[k]}(\widehat{\mathbf{U}}_{\text{tr}}^{[k]}) = \mathbf{0}$ and $\underline{\lambda}_1^{[k]} \geq 2\mu$ in accord with the definition (47), we have that

$$\frac{1}{2}\big(F_1^{[k]}(\widehat{\mathbf{U}}_{\text{tr}}^{[k-1]}) - F_1^{[k]}(\widehat{\mathbf{U}}_{\text{tr}}^{[k]})\big) \geq \frac{\mu}{2}\|\widehat{\mathbf{U}}_{\text{tr}}^{[k-1]} - \widehat{\mathbf{U}}_{\text{tr}}^{[k]}\|_{\text{F}}^2. \quad (67)$$

Similarly, we have that

$$\frac{1}{2}\big(F_2^{[k]}(\widehat{\mathbf{U}}_{\text{te}}^{[k-1]}) - F_2^{[k]}(\widehat{\mathbf{U}}_{\text{te}}^{[k]})\big) \geq \frac{\mu}{2}\|\widehat{\mathbf{U}}_{\text{te}}^{[k-1]} - \widehat{\mathbf{U}}_{\text{te}}^{[k]}\|_{\text{F}}^2, \quad (68)$$

$$\frac{1}{2}\big(F_3^{[k]}(\widehat{\mathbf{V}}_{\text{tr}}^{[k-1]}) - F_3^{[k]}(\widehat{\mathbf{V}}_{\text{tr}}^{[k]})\big) \geq \frac{\mu}{2}\|\widehat{\mathbf{V}}_{\text{tr}}^{[k-1]} - \widehat{\mathbf{V}}_{\text{tr}}^{[k]}\|_{\text{F}}^2, \quad (69)$$

and

$$\frac{1}{2}\big(F_4^{[k]}(\widehat{\mathbf{V}}_{\text{te}}^{[k-1]}) - F_4^{[k]}(\widehat{\mathbf{V}}_{\text{te}}^{[k]})\big) \geq \frac{\mu}{2}\|\widehat{\mathbf{V}}_{\text{te}}^{[k-1]} - \widehat{\mathbf{V}}_{\text{te}}^{[k]}\|_{\text{F}}^2. \quad (70)$$

Combining (65) with (67)-(70) and summing over k from $k = 1$ to $k = K$, we have that

$$\frac{2}{\mu}\Big(F(\widehat{\mathbf{U}}_{\text{tr}}^{[0]}, \widehat{\mathbf{U}}_{\text{te}}^{[0]}, \widehat{\mathbf{V}}_{\text{tr}}^{[0]}, \widehat{\mathbf{V}}_{\text{te}}^{[0]}) - F(\widehat{\mathbf{U}}_{\text{tr}}^{[K]}, \widehat{\mathbf{U}}_{\text{te}}^{[K]}, \widehat{\mathbf{V}}_{\text{tr}}^{[K]}, \widehat{\mathbf{V}}_{\text{te}}^{[K]})\Big) \geq \sum_{k=1}^{K}\Big(\|\widehat{\mathbf{U}}_{\text{tr}}^{[k-1]} - \widehat{\mathbf{U}}_{\text{tr}}^{[k]}\|_{\text{F}}^2 + \|\widehat{\mathbf{U}}_{\text{te}}^{[k-1]} - \widehat{\mathbf{U}}_{\text{te}}^{[k]}\|_{\text{F}}^2 + \|\widehat{\mathbf{V}}_{\text{tr}}^{[k-1]} - \widehat{\mathbf{V}}_{\text{tr}}^{[k]}\|_{\text{F}}^2 + \|\widehat{\mathbf{V}}_{\text{te}}^{[k-1]} - \widehat{\mathbf{V}}_{\text{te}}^{[k]}\|_{\text{F}}^2\Big). \quad (71)$$

Noting that $F(\mathbf{U}_{\text{tr}}, \mathbf{U}_{\text{te}}, \mathbf{V}_{\text{tr}}, \mathbf{V}_{\text{te}}) \geq 0$ for any $\{\mathbf{U}_{\text{tr}}, \mathbf{U}_{\text{te}}, \mathbf{V}_{\text{tr}}, \mathbf{V}_{\text{te}}\}$ and taking $K \to \infty$ completes the proof. □

Corollary 1. *The sequence* $\{F(\widehat{\mathbf{U}}_{\text{tr}}^{[k]}, \widehat{\mathbf{U}}_{\text{te}}^{[k]}, \widehat{\mathbf{V}}_{\text{tr}}^{[k]}, \widehat{\mathbf{V}}_{\text{te}}^{[k]})\}_{k\geq 0}$ *is non-increasing. Moreover, there exists a constant* $0 < C_0 \leq \frac{2}{\mu}F(\widehat{\mathbf{U}}_{\text{tr}}^{[0]}, \widehat{\mathbf{U}}_{\text{te}}^{[0]}, \widehat{\mathbf{V}}_{\text{tr}}^{[0]}, \widehat{\mathbf{V}}_{\text{te}}^{[0]})$ *such that*

$$C_0 \geq \|\widehat{\mathbf{U}}_{\text{tr}}^{[k-1]} - \widehat{\mathbf{U}}_{\text{tr}}^{[k]}\|_{\text{F}}^2 + \|\widehat{\mathbf{U}}_{\text{te}}^{[k-1]} - \widehat{\mathbf{U}}_{\text{te}}^{[k]}\|_{\text{F}}^2 + \|\widehat{\mathbf{V}}_{\text{tr}}^{[k-1]} - \widehat{\mathbf{V}}_{\text{tr}}^{[k]}\|_{\text{F}}^2 + \|\widehat{\mathbf{V}}_{\text{te}}^{[k-1]} - \widehat{\mathbf{V}}_{\text{te}}^{[k]}\|_{\text{F}}^2 \quad (72)$$

for any $k \geq 1$.

Proof. The first assertion follows immediately from (65) and (67) - (70). The second assertion follows immediately from the first assertion and (71). □

Lemma 1 implies convergence (in the ℓ_2 sense) to a limit point, that is, (62) implies that as $k \to \infty$, $\|\widehat{\mathbf{U}}_{\text{te}}^{[k-1]} - \widehat{\mathbf{U}}_{\text{te}}^{[k]}\|_{\text{F}}^2 \to 0$, $\|\widehat{\mathbf{U}}_{\text{tr}}^{[k-1]} - \widehat{\mathbf{U}}_{\text{tr}}^{[k]}\|_{\text{F}}^2 \to 0$, $\|\widehat{\mathbf{V}}_{\text{tr}}^{[k-1]} - \widehat{\mathbf{V}}_{\text{tr}}^{[k]}\|_{\text{F}}^2 \to 0$, $\|\widehat{\mathbf{V}}_{\text{te}}^{[k-1]} - \widehat{\mathbf{V}}_{\text{te}}^{[k]}\|_{\text{F}}^2 \to 0$, and that there exists a solution $\{\overline{\mathbf{U}}_{\text{tr}}, \overline{\mathbf{U}}_{\text{te}}, \overline{\mathbf{V}}_{\text{tr}}, \overline{\mathbf{V}}_{\text{te}}\}$, such that

$$\|\widehat{\mathbf{U}}_{\text{tr}}^{[k]} - \overline{\mathbf{U}}_{\text{tr}}\|_{\text{F}}^2 + \|\widehat{\mathbf{U}}_{\text{te}}^{[k]} - \overline{\mathbf{U}}_{\text{te}}\|_{\text{F}}^2 + \|\widehat{\mathbf{V}}_{\text{tr}}^{[k]} - \overline{\mathbf{V}}_{\text{tr}}\|_{\text{F}}^2 + \|\widehat{\mathbf{V}}_{\text{te}}^{[k]} - \overline{\mathbf{V}}_{\text{te}}\|_{\text{F}}^2 \underset{k\to\infty}{\longrightarrow} 0. \quad (73)$$

We prove that the convergence rate is sub-linear. To do so, we next state a well-known result (see for example [2, Lemma 3.5]), which we provide for the sake of completeness.

Lemma 2. *Let* $\{q_k\}_{k\geq 0}$ *be a non-increasing sequence of positive real numbers and let* $0 < \overline{q} < \infty$ *and* $0 < C < \infty$ *be two positive constants. Define* $B \equiv \frac{C}{\overline{q}}$ *and assume that (i)* $q_0 < \overline{q}$ *and (ii)* $q_{k-1} - q_k \geq C^{-1}q_{k-1}^2$. *Then*

$$q_k \leq \frac{C}{B + k}. \quad (74)$$

Proof. The second condition implies that

$$\frac{1}{q_k} - \frac{1}{q_{k-1}} = \frac{q_{k-1} - q_k}{q_{k-1}q_k} \geq \frac{q_{k-1}^2}{C(q_{k-1}q_k)} = \frac{1}{C}\frac{q_{k-1}}{q_k} \geq \frac{1}{C}. \quad (75)$$

Then

$$\frac{1}{q_k} \geq \frac{1}{q_0} + \frac{k}{C} \geq \overline{q}^{-1} + \frac{k}{C}, \quad (76)$$

which completes the proof. □

Lemma 3. *Assume the conditions of Lemma 1 hold. Assume that* $\partial_{\mathbf{U}} F_1^{[k]}(\widehat{\mathbf{U}}_{\text{tr}}^{[k-1]}) \neq \mathbf{0}$, $\partial_{\mathbf{U}} F_2^{[k]}(\widehat{\mathbf{U}}_{\text{te}}^{[k-1]}) \neq \mathbf{0}$, $\partial_{\mathbf{V}} F_3^{[k]}(\widehat{\mathbf{V}}_{\text{tr}}^{[k-1]}) \neq \mathbf{0}$, *and* $\partial_{\mathbf{V}} F_4^{[k]}(\widehat{\mathbf{V}}_{\text{te}}^{[k-1]}) \neq \mathbf{0}$. *Let*

$$\Sigma^{[k]} \equiv \Big(F_1^{[k]}(\widehat{\mathbf{U}}_{\text{tr}}^{[k]}) + F_2^{[k]}(\widehat{\mathbf{U}}_{\text{te}}^{[k]}) + F_3^{[k]}(\widehat{\mathbf{V}}_{\text{tr}}^{[k]}) + F_4^{[k]}(\widehat{\mathbf{V}}_{\text{te}}^{[k]})\Big)^2 \quad (77)$$

and

$$\Delta^{[k]} \equiv \|\partial_{\mathbf{U}} F_1^{[k]}(\widehat{\mathbf{U}}_{\text{tr}}^{[k-1]})\|_{\text{F}}^2 + \|\partial_{\mathbf{U}} F_2^{[k]}(\widehat{\mathbf{U}}_{\text{te}}^{[k-1]})\|_{\text{F}}^2 + \|\partial_{\mathbf{V}} F_3^{[k]}(\widehat{\mathbf{V}}_{\text{tr}}^{[k-1]})\|_{\text{F}}^2 + \|\partial_{\mathbf{V}} F_4^{[k]}(\widehat{\mathbf{V}}_{\text{te}}^{[k-1]})\|_{\text{F}}^2. \quad (78)$$

Then there exists a positive constant $0 < \overline{C}_0 < \infty$*, which only depends on the initial solution* $(\widehat{\mathbf{U}}_{\text{tr}}^{[0]}, \widehat{\mathbf{U}}_{\text{te}}^{[0]}, \widehat{\mathbf{V}}_{\text{tr}}^{[0]}, \widehat{\mathbf{V}}_{\text{te}}^{[0]})$ *such that*

$$\overline{C}_0 \geq 1 + \frac{\widetilde{C}\Sigma^{[k]}}{\Delta^{[k]}} \tag{79}$$

for any positive constant $0 < \widetilde{C} < \infty$ *and all* $k \geq 1$.

Proof. Define a constant $\widetilde{C}_1$ so that $\widetilde{C}_1 \geq F_1^{[0]}(\widehat{\mathbf{U}}_{\text{tr}}^{[0]}) + F_2^{[0]}(\widehat{\mathbf{U}}_{\text{te}}^{[0]}) + F_3^{[0]}(\widehat{\mathbf{V}}_{\text{tr}}^{[0]}) + F_4^{[0]}(\widehat{\mathbf{V}}_{\text{te}}^{[0]})$. Then from (63), (64), and Corollary 1, we have that $\widetilde{C}_1 \geq F_1^{[k]}(\widehat{\mathbf{U}}_{\text{tr}}^{[k]}) + F_2^{[k]}(\widehat{\mathbf{U}}_{\text{te}}^{[k]}) + F_3^{[k]}(\widehat{\mathbf{V}}_{\text{tr}}^{[k]}) + F_4^{[k]}(\widehat{\mathbf{V}}_{\text{te}}^{[k]})$ for all $k \geq 1$. Next, define

$$\begin{aligned}\mathcal{K} \equiv \big\{k : \partial_{\mathbf{U}} F_1^{[k]}(\widehat{\mathbf{U}}_{\text{tr}}^{[k-1]}) \neq \mathbf{0}, \partial_{\mathbf{U}} F_2^{[k]}(\widehat{\mathbf{U}}_{\text{te}}^{[k-1]}) \neq \mathbf{0}, \\ \partial_{\mathbf{V}} F_3^{[k]}(\widehat{\mathbf{V}}_{\text{tr}}^{[k-1]}) \neq \mathbf{0}, \partial_{\mathbf{V}} F_4^{[k]}(\widehat{\mathbf{V}}_{\text{te}}^{[k-1]}) \neq \mathbf{0}\big\}.\end{aligned} \tag{80}$$

We immediately have that there exists constants $0 < \widetilde{C}_2^{[k]} < \infty$ so that

$$\begin{aligned}\widetilde{C}_2^{[k]} \leq \|\partial_{\mathbf{U}} F_1^{[k]}(\widehat{\mathbf{U}}_{\text{tr}}^{[k-1]})\|_{\text{F}}^2 + \|\partial_{\mathbf{U}} F_2^{[k]}(\widehat{\mathbf{U}}_{\text{te}}^{[k-1]})\|_{\text{F}}^2 \\ + \|\partial_{\mathbf{V}} F_3^{[k]}(\widehat{\mathbf{V}}_{\text{tr}}^{[k-1]})\|_{\text{F}}^2 + \|\partial_{\mathbf{V}} F_4^{[k]}(\widehat{\mathbf{V}}_{\text{te}}^{[k-1]})\|_{\text{F}}^2\end{aligned} \tag{81}$$

for all $k \in \mathcal{K}$. Define the lower bound

$$\widetilde{C}_2 \equiv \inf_{k \in \mathcal{K}} \widetilde{C}_2^{[k]}; \tag{82}$$

then, for all $k \in \mathcal{K}$,

$$\frac{1}{\widetilde{C}_2} \geq \frac{1}{\Delta^{[k]}}. \tag{83}$$

Letting $\overline{C}_0 = 1 + \widetilde{C}_2^{-1}\widetilde{C}\widetilde{C}_1^2$ completes the proof. □

The set $\mathcal{K}$ defined in (80) is the set of iteration indices before convergence is achieved. It follows immediately from Lemma 3 that before the algorithm converges (i.e., for $k \in \mathcal{K}$) there exists a positive constant $0 < \overline{C}_0 < \infty$ such that

$$\overline{C}_0 \Delta^{[k]} \geq \Delta^{[k]} + \Sigma^{[k]}. \tag{84}$$

We next prove our first main result related to the speed of convergence of Algorithm 1. Specifically, we prove that the distance from the final solution shrinks in inverse proportion to the number of iterations, that is very few iterations are required to achieve convergence and the number of iterations can be specified beforehand.

Theorem 3.1 (Sub-linear Convergence Rate). *Let* F, $F_1^{[k]}$, $F_2^{[k]}$, $F_3^{[k]}$, *and* $F_4^{[k]}$ *be as defined in* (29) *and* (34)–(37)*, and let the block updates* $\widehat{\mathbf{U}}_{\text{tr}}^{[k]}$, $\widehat{\mathbf{U}}_{\text{te}}^{[k]}$, $\widehat{\mathbf{V}}_{\text{tr}}^{[k]}$, *and* $\widehat{\mathbf{V}}_{\text{te}}^{[k]}$ *be as given in* (38)–(41)*. Then, there exists two positive constants* $0 < B < \infty$ *and* $0 < C < \infty$ *such that, for any* $k \geq 0$,

$$\left|F(\widehat{\mathbf{U}}_{\text{tr}}^{[k]}, \widehat{\mathbf{U}}_{\text{te}}^{[k]}, \widehat{\mathbf{V}}_{\text{tr}}^{[k]}, \widehat{\mathbf{V}}_{\text{te}}^{[k]}) - F(\overline{\mathbf{U}}_{\text{tr}}, \overline{\mathbf{U}}_{\text{te}}, \overline{\mathbf{V}}_{\text{tr}}, \overline{\mathbf{V}}_{\text{te}})\right| \leq \frac{C}{B+k}. \tag{85}$$

Proof. The case $k = 0$ is trivial so we will focus on $k \geq 1$. From (65), (67) - (70), and the Lipschitz bounds (55) - (56), we have that

$$\begin{aligned}&F(\widehat{\mathbf{U}}_{\text{tr}}^{[k-1]}, \widehat{\mathbf{U}}_{\text{te}}^{[k-1]}, \widehat{\mathbf{V}}_{\text{tr}}^{[k-1]}, \widehat{\mathbf{V}}_{\text{te}}^{[k-1]}) - F(\widehat{\mathbf{U}}_{\text{tr}}^{[k]}, \widehat{\mathbf{U}}_{\text{te}}^{[k]}, \widehat{\mathbf{V}}_{\text{tr}}^{[k]}, \widehat{\mathbf{V}}_{\text{te}}^{[k]}) \\ &\quad\geq \frac{\mu}{2L_{\max}}\Big(\|\partial_{\mathbf{U}} F_1^{[k]}(\widehat{\mathbf{U}}_{\text{tr}}^{[k-1]}) - \partial_{\mathbf{U}} F_1^{[k]}(\widehat{\mathbf{U}}_{\text{tr}}^{[k]})\|_{\text{F}}^2 \\ &\qquad + \|\partial_{\mathbf{U}} F_2^{[k]}(\widehat{\mathbf{U}}_{\text{te}}^{[k-1]}) - \partial_{\mathbf{U}} F_2^{[k]}(\widehat{\mathbf{U}}_{\text{te}}^{[k]})\|_{\text{F}}^2 \\ &\qquad + \|\partial_{\mathbf{V}} F_3^{[k]}(\widehat{\mathbf{V}}_{\text{tr}}^{[k-1]}) - \partial_{\mathbf{V}} F_3^{[k]}(\widehat{\mathbf{V}}_{\text{tr}}^{[k]})\|_{\text{F}}^2 \\ &\qquad + \|\partial_{\mathbf{V}} F_4^{[k]}(\widehat{\mathbf{V}}_{\text{te}}^{[k-1]}) - \partial_{\mathbf{V}} F_4^{[k]}(\widehat{\mathbf{V}}_{\text{te}}^{[k]})\|_{\text{F}}^2\Big) \\ &\quad= \frac{\mu}{2L_{\max}}\Big(\|\partial_{\mathbf{U}} F_1^{[k]}(\widehat{\mathbf{U}}_{\text{tr}}^{[k-1]})\|_{\text{F}}^2 + \|\partial_{\mathbf{U}} F_2^{[k]}(\widehat{\mathbf{U}}_{\text{te}}^{[k-1]})\|_{\text{F}}^2 \\ &\qquad + \|\partial_{\mathbf{V}} F_3^{[k]}(\widehat{\mathbf{V}}_{\text{tr}}^{[k-1]})\|_{\text{F}}^2 + \|\partial_{\mathbf{V}} F_4^{[k]}(\widehat{\mathbf{V}}_{\text{te}}^{[k-1]})\|_{\text{F}}^2\Big).\end{aligned} \tag{86}$$

By convexity of $F_1^{[k]}$ and the Cauchy-Schwartz inequality, we have that

$$\begin{aligned}F_1^{[k]}(\widehat{\mathbf{U}}_{\text{tr}}^{[k-1]}) - F_1^{[k]}(\widehat{\mathbf{U}}_{\text{tr}}^{[k]}) &\leq \langle \partial_{\mathbf{U}} F_1^{[k]}(\widehat{\mathbf{U}}_{\text{tr}}^{[k-1]}), \widehat{\mathbf{U}}_{\text{tr}}^{[k-1]} - \widehat{\mathbf{U}}_{\text{tr}}^{[k]} \rangle \\ &\leq \|\partial_{\mathbf{U}} F_1^{[k]}(\widehat{\mathbf{U}}_{\text{tr}}^{[k-1]})\|_{\text{F}} \|\widehat{\mathbf{U}}_{\text{tr}}^{[k-1]} - \widehat{\mathbf{U}}_{\text{tr}}^{[k]}\|_{\text{F}}.\end{aligned} \tag{87}$$

Then

$$\|\partial_{\mathbf{U}} F_1^{[k]}(\widehat{\mathbf{U}}_{\text{tr}}^{[k-1]})\|_{\text{F}}^2 \geq \frac{1}{C_0}\big(F_1^{[k]}(\widehat{\mathbf{U}}_{\text{tr}}^{[k-1]}) - F_1^{[k]}(\widehat{\mathbf{U}}_{\text{tr}}^{[k]})\big)^2, \tag{88}$$

where C_0 is the positive constant of Corollary 1. We can write similar bounds for $F_2^{[k]}$, $F_3^{[k]}$, and $F_2^{[k]}$ to obtain

$$\begin{aligned}&\frac{1}{4C_0}\Big(F_1^{[k]}(\widehat{\mathbf{U}}_{\text{tr}}^{[k-1]}) + F_2^{[k]}(\widehat{\mathbf{U}}_{\text{te}}^{[k-1]}) + F_3^{[k]}(\widehat{\mathbf{V}}_{\text{tr}}^{[k-1]}) \\ &\quad + F_4^{[k]}(\widehat{\mathbf{V}}_{\text{te}}^{[k-1]}) - F_1^{[k]}(\widehat{\mathbf{U}}_{\text{tr}}^{[k]}) - F_2^{[k]}(\widehat{\mathbf{U}}_{\text{te}}^{[k]}) - F_3^{[k]}(\widehat{\mathbf{V}}_{\text{tr}}^{[k]}) \\ &- F_4^{[k]}(\widehat{\mathbf{V}}_{\text{te}}^{[k]})\Big)^2 \leq \|\partial_{\mathbf{U}} F_1^{[k]}(\widehat{\mathbf{U}}_{\text{tr}}^{[k-1]})\|_{\text{F}}^2 + \|\partial_{\mathbf{U}} F_2^{[k]}(\widehat{\mathbf{U}}_{\text{te}}^{[k-1]})\|_{\text{F}}^2 \\ &\qquad + \|\partial_{\mathbf{V}} F_3^{[k]}(\widehat{\mathbf{V}}_{\text{tr}}^{[k-1]})\|_{\text{F}}^2 + \|\partial_{\mathbf{V}} F_4^{[k]}(\widehat{\mathbf{V}}_{\text{te}}^{[k-1]})\|_{\text{F}}^2,\end{aligned} \tag{89}$$

where the left hand side follows from the triangle inequal-

ity. Applying the reverse triangle inequality, we get

$$\frac{1}{4C_0}\Big(F_1^{[k]}(\widehat{\mathbf{U}}_{\text{tr}}^{[k-1]}) + F_2^{[k]}(\widehat{\mathbf{U}}_{\text{te}}^{[k-1]}) + F_3^{[k]}(\widehat{\mathbf{V}}_{\text{tr}}^{[k-1]}) + F_4^{[k]}(\widehat{\mathbf{V}}_{\text{te}}^{[k-1]})\Big)^2 \leq \frac{1}{4C_0}\Big(F_1^{[k]}(\widehat{\mathbf{U}}_{\text{tr}}^{[k]}) + F_2^{[k]}(\widehat{\mathbf{U}}_{\text{te}}^{[k]}) + F_3^{[k]}(\widehat{\mathbf{V}}_{\text{tr}}^{[k]}) + F_4^{[k]}(\widehat{\mathbf{V}}_{\text{te}}^{[k]})\Big)^2 + \big\|\partial_{\mathbf{U}} F_1^{[k]}(\widehat{\mathbf{U}}_{\text{tr}}^{[k-1]})\big\|_{\text{F}}^2 + \big\|\partial_{\mathbf{U}} F_2^{[k]}(\widehat{\mathbf{U}}_{\text{te}}^{[k-1]})\big\|_{\text{F}}^2 + \big\|\partial_{\mathbf{U}} F_3^{[k]}(\widehat{\mathbf{V}}_{\text{tr}}^{[k-1]})\big\|_{\text{F}}^2 + \big\|\partial_{\mathbf{U}} F_4^{[k]}(\widehat{\mathbf{V}}_{\text{te}}^{[k-1]})\big\|_{\text{F}}^2. \quad (90)$$

It can be easily shown that

$$F(\widehat{\mathbf{U}}_{\text{tr}}^{[k-1]}, \widehat{\mathbf{U}}_{\text{te}}^{[k-1]}, \widehat{\mathbf{V}}_{\text{tr}}^{[k-1]}, \widehat{\mathbf{V}}_{\text{te}}^{[k-1]}) \leq F_1^{[k]}(\widehat{\mathbf{U}}_{\text{tr}}^{[k-1]}) + F_2^{[k]}(\widehat{\mathbf{U}}_{\text{te}}^{[k-1]}) + F_3^{[k]}(\widehat{\mathbf{V}}_{\text{tr}}^{[k-1]}) + F_4^{[k]}(\widehat{\mathbf{V}}_{\text{te}}^{[k-1]}). \quad (91)$$

Hence,

$$\frac{1}{4C_0}\Big(F(\widehat{\mathbf{U}}_{\text{tr}}^{[k-1]}, \widehat{\mathbf{U}}_{\text{te}}^{[k-1]}, \widehat{\mathbf{V}}_{\text{tr}}^{[k-1]}, \widehat{\mathbf{V}}_{\text{te}}^{[k-1]})\Big)^2 \leq \frac{1}{4C_0}\Big(F_1^{[k]}(\widehat{\mathbf{U}}_{\text{tr}}^{[k]}) + F_2^{[k]}(\widehat{\mathbf{U}}_{\text{te}}^{[k]}) + F_3^{[k]}(\widehat{\mathbf{V}}_{\text{tr}}^{[k]}) + F_4^{[k]}(\widehat{\mathbf{V}}_{\text{te}}^{[k]})\Big)^2 + \big\|\partial_{\mathbf{U}} F_1^{[k]}(\widehat{\mathbf{U}}_{\text{tr}}^{[k-1]})\big\|_{\text{F}}^2 + \big\|\partial_{\mathbf{U}} F_2^{[k]}(\widehat{\mathbf{U}}_{\text{te}}^{[k-1]})\big\|_{\text{F}}^2 + \big\|\partial_{\mathbf{U}} F_3^{[k]}(\widehat{\mathbf{V}}_{\text{tr}}^{[k-1]})\big\|_{\text{F}}^2 + \big\|\partial_{\mathbf{U}} F_4^{[k]}(\widehat{\mathbf{V}}_{\text{te}}^{[k-1]})\big\|_{\text{F}}^2. \quad (92)$$

From (84) (and Lemma 3) we have that there exists a positive constant $\overline{C}_0$ which only depends on the initial solution so that (92) implies that

$$\frac{1}{4C_0\overline{C}_0}\Big(F(\widehat{\mathbf{U}}_{\text{tr}}^{[k-1]}, \widehat{\mathbf{U}}_{\text{te}}^{[k-1]}, \widehat{\mathbf{V}}_{\text{tr}}^{[k-1]}, \widehat{\mathbf{V}}_{\text{te}}^{[k-1]})\Big)^2 \leq \big\|\partial_{\mathbf{U}} F_1^{[k]}(\widehat{\mathbf{U}}_{\text{tr}}^{[k-1]})\big\|_{\text{F}}^2 + \big\|\partial_{\mathbf{U}} F_2^{[k]}(\widehat{\mathbf{U}}_{\text{te}}^{[k-1]})\big\|_{\text{F}}^2 + \big\|\partial_{\mathbf{U}} F_3^{[k]}(\widehat{\mathbf{V}}_{\text{tr}}^{[k-1]})\big\|_{\text{F}}^2 + \big\|\partial_{\mathbf{U}} F_4^{[k]}(\widehat{\mathbf{V}}_{\text{te}}^{[k-1]})\big\|_{\text{F}}^2. \quad (93)$$

Combining this with (86), we get

$$F(\widehat{\mathbf{U}}_{\text{tr}}^{[k-1]}, \widehat{\mathbf{U}}_{\text{te}}^{[k-1]}, \widehat{\mathbf{V}}_{\text{tr}}^{[k-1]}, \widehat{\mathbf{V}}_{\text{te}}^{[k-1]}) - F(\widehat{\mathbf{U}}_{\text{tr}}^{[k]}, \widehat{\mathbf{U}}_{\text{te}}^{[k]}, \widehat{\mathbf{V}}_{\text{tr}}^{[k]}, \widehat{\mathbf{V}}_{\text{te}}^{[k]}) \geq \frac{\mu}{8L_{\max}C_0\overline{C}_0}\Big(F(\widehat{\mathbf{U}}_{\text{tr}}^{[k-1]}, \widehat{\mathbf{U}}_{\text{te}}^{[k-1]}, \widehat{\mathbf{V}}_{\text{tr}}^{[k-1]}, \widehat{\mathbf{V}}_{\text{te}}^{[k-1]})\Big)^2. \quad (94)$$

Since $F(\widehat{\mathbf{U}}_{\text{tr}}^{[k]}, \widehat{\mathbf{U}}_{\text{te}}^{[k]}, \widehat{\mathbf{V}}_{\text{tr}}^{[k]}, \widehat{\mathbf{V}}_{\text{te}}^{[k]}) \geq F(\overline{\mathbf{U}}_{\text{tr}}, \overline{\mathbf{U}}_{\text{te}}, \overline{\mathbf{V}}_{\text{tr}}, \overline{\mathbf{V}}_{\text{te}})$ for all k, (94) implies that

$$F(\widehat{\mathbf{U}}_{\text{tr}}^{[k-1]}, \widehat{\mathbf{U}}_{\text{te}}^{[k-1]}, \widehat{\mathbf{V}}_{\text{tr}}^{[k-1]}, \widehat{\mathbf{V}}_{\text{te}}^{[k-1]}) - F(\widehat{\mathbf{U}}_{\text{tr}}^{[k]}, \widehat{\mathbf{U}}_{\text{te}}^{[k]}, \widehat{\mathbf{V}}_{\text{tr}}^{[k]}, \widehat{\mathbf{V}}_{\text{te}}^{[k]}) \geq \frac{\mu}{8L_{\max}C_0\overline{C}_0}\Big(F(\widehat{\mathbf{U}}_{\text{tr}}^{[k-1]}, \widehat{\mathbf{U}}_{\text{te}}^{[k-1]}, \widehat{\mathbf{V}}_{\text{tr}}^{[k-1]}, \widehat{\mathbf{V}}_{\text{te}}^{[k-1]}) - F(\overline{\mathbf{U}}_{\text{tr}}, \overline{\mathbf{U}}_{\text{te}}, \overline{\mathbf{V}}_{\text{tr}}, \overline{\mathbf{V}}_{\text{te}})\Big)^2. \quad (95)$$

We next add and subtract $F(\overline{\mathbf{U}}_{\text{tr}}, \overline{\mathbf{U}}_{\text{te}}, \overline{\mathbf{V}}_{\text{tr}}, \overline{\mathbf{V}}_{\text{te}})$ on the left hand side, and we invoke Lemma (2) with

$$q_k \equiv F(\widehat{\mathbf{U}}_{\text{tr}}^{[k]}, \widehat{\mathbf{U}}_{\text{te}}^{[k]}, \widehat{\mathbf{V}}_{\text{tr}}^{[k]}, \widehat{\mathbf{V}}_{\text{te}}^{[k]}) - F(\overline{\mathbf{U}}_{\text{tr}}, \overline{\mathbf{U}}_{\text{te}}, \overline{\mathbf{V}}_{\text{tr}}, \overline{\mathbf{V}}_{\text{te}}) \quad (96)$$

and note that $\{q_k\}_{k\geq 0}$ is a non-negative sequence by the first assertion of Corollary 1. Letting $B = 16L_{\max}\overline{C}_0\mu^{-2}$ and $C = 8L_{\max}C_0\overline{C}_0\mu^{-1}$ completes the proof. □

The result in Theorem 3.1 suggests that one can determine a number of iterations as a stopping criterion for Algorithm 1. More importantly, the theorem says that this number need not be large in order to get close to the limit. The next theorem provides our second main result. It demonstrates that the limiting solution $\{\overline{\mathbf{U}}_{\text{tr}}, \overline{\mathbf{U}}_{\text{te}}, \overline{\mathbf{V}}_{\text{tr}}, \overline{\mathbf{V}}_{\text{te}}\}$ implied by Lemma 1 is a block coordinate-wise minimizer of the matrix completion problem (28).

Theorem 3.2 (Block Corrdinate-Wise Minimizer). *Under the assumptions of Lemma 1, the variational inequalities (30)-(33) hold for all* $\mathbf{U}_{\text{tr}} \in \mathbb{R}^{n\times r}$, *all* $\mathbf{U}_{\text{te}} \in \mathbb{R}^{h\times r}$, *all* $\mathbf{V}_{\text{tr}} \in \mathbb{R}^{T_{\text{tr}}\times r}$, *and all* $\mathbf{V}_{\text{te}} \in \mathbb{R}^{T_{\text{te}}\times r}$.

Proof. In the limit, we have that

$$\overline{\mathbf{U}}_{\text{tr}} = \underset{\mathbf{U}_{\text{tr}}\in\mathbb{R}^{n\times r}}{\arg\min} \|\mathbf{U}_{\text{tr}}\overline{\mathbf{V}}_{\text{tr}}^{\top} - \mathbf{Y}_{\text{tr}}\|_{\text{F}}^2 + 2\mu\|\mathbf{U}_{\text{tr}}\|_{\text{F}}^2. \quad (97)$$

Hence, for all $\mathbf{U}_{\text{tr}} \in \mathbb{R}^{n\times r}$

$$F(\mathbf{U}_{\text{tr}}, \overline{\mathbf{U}}_{\text{te}}, \overline{\mathbf{V}}_{\text{tr}}, \overline{\mathbf{V}}_{\text{te}}) \geq F(\overline{\mathbf{U}}_{\text{tr}}, \overline{\mathbf{U}}_{\text{te}}, \overline{\mathbf{V}}_{\text{tr}}, \overline{\mathbf{V}}_{\text{te}}). \quad (98)$$

In particular, consider the matrix $\overline{\mathbf{U}}_{\text{tr}} + h(\mathbf{U}_{\text{tr}} - \overline{\mathbf{U}}_{\text{tr}})$, where h is a scalar and $\mathbf{U}_{\text{tr}}$ is any $\mathbb{R}^{n\times r}$ matrix. From (98), we have for each $h > 0$ that

$$\frac{1}{h}\Big(F(\overline{\mathbf{U}}_{\text{tr}} + h(\mathbf{U}_{\text{tr}} - \overline{\mathbf{U}}_{\text{tr}}), \overline{\mathbf{U}}_{\text{te}}, \overline{\mathbf{V}}_{\text{tr}}, \overline{\mathbf{V}}_{\text{te}}) - F(\overline{\mathbf{U}}_{\text{tr}}, \overline{\mathbf{U}}_{\text{te}}, \overline{\mathbf{V}}_{\text{tr}}, \overline{\mathbf{V}}_{\text{te}})\Big) \geq 0 \quad (99)$$

is true for all $\mathbf{U}_{\text{tr}} \in \mathbb{R}^{n\times r}$. Since F is continuous in all block coordinates, upon taking $h \to 0$ the left-hand side converges to the directional partial derivative of F along $\mathbf{U}_{\text{tr}} - \overline{\mathbf{U}}_{\text{tr}}$ and since the inequality is true for all h, it is true in the limit. Then, since the directional derivative is simply the inner product of the derivative and the direction, it follows that the first variational inequality,

$$\big\langle \partial_{\mathbf{U}_{\text{tr}}} F(\overline{\mathbf{U}}_{\text{tr}}, \overline{\mathbf{U}}_{\text{te}}, \overline{\mathbf{V}}_{\text{tr}}, \overline{\mathbf{V}}_{\text{te}}), \mathbf{U}_{\text{tr}} - \overline{\mathbf{U}}_{\text{tr}} \big\rangle \geq 0, \quad (100)$$

is true for all $\mathbf{U}_{\text{tr}} \in \mathbb{R}^{n\times r}$. The same reasoning can be used to demonstrate that the variational inequalities (31), (32), and (33) also hold for $\overline{\mathbf{U}}_{\text{te}}$, $\overline{\mathbf{V}}_{\text{tr}}$, and $\overline{\mathbf{V}}_{\text{te}}$, respectively. □

The interpretation of the solution as a Nash point (an equilibrium) can be seen immediately upon examining the inequalities

$$F(\mathbf{U}_{\text{tr}}, \overline{\mathbf{U}}_{\text{te}}, \overline{\mathbf{V}}_{\text{tr}}, \overline{\mathbf{V}}_{\text{te}}) \geq F(\overline{\mathbf{U}}_{\text{tr}}, \overline{\mathbf{U}}_{\text{te}}, \overline{\mathbf{V}}_{\text{tr}}, \overline{\mathbf{V}}_{\text{te}}), \tag{101}$$

$$F(\overline{\mathbf{U}}_{\text{tr}}, \mathbf{U}_{\text{te}}, \overline{\mathbf{V}}_{\text{tr}}, \overline{\mathbf{V}}_{\text{te}}) \geq F(\overline{\mathbf{U}}_{\text{tr}}, \overline{\mathbf{U}}_{\text{te}}, \overline{\mathbf{V}}_{\text{tr}}, \overline{\mathbf{V}}_{\text{te}}), \tag{102}$$

$$F(\overline{\mathbf{U}}_{\text{tr}}, \overline{\mathbf{U}}_{\text{te}}, \mathbf{V}_{\text{tr}}, \overline{\mathbf{V}}_{\text{te}}) \geq F(\overline{\mathbf{U}}_{\text{tr}}, \overline{\mathbf{U}}_{\text{te}}, \overline{\mathbf{V}}_{\text{tr}}, \overline{\mathbf{V}}_{\text{te}}), \tag{103}$$

and

$$F(\overline{\mathbf{U}}_{\text{tr}}, \overline{\mathbf{U}}_{\text{te}}, \overline{\mathbf{V}}_{\text{tr}}, \mathbf{V}_{\text{te}}) \geq F(\overline{\mathbf{U}}_{\text{tr}}, \overline{\mathbf{U}}_{\text{te}}, \overline{\mathbf{V}}_{\text{tr}}, \overline{\mathbf{V}}_{\text{te}}) \tag{104}$$

for all $\mathbf{U}_{\text{tr}} \in \mathbb{R}^{n\times r}$ $\mathbf{U}_{\text{te}} \in \mathbb{R}^{h\times r}$, $\mathbf{V}_{\text{tr}} \in \mathbb{R}^{T_{\text{tr}}\times r}$, and $\mathbf{V}_{\text{te}} \in \mathbb{R}^{T_{\text{te}}\times r}$. The inequalities (102), (103), and (104) can be established in the same way that (101) was established in the proof of Theorem 3.2.

4 Ensemble Learning and Time Complexity

4.1 Incorporating Historical Patterns

We extend the solution approach presented above to capture and incorporate patterns that may exist in the data. The extension leverages the "bigness" of high-resolution traffic data. To this end, we employ a boosting technique in which the ensembles represent past days. Boosting has become a standard component of online learning algorithms; it has been established that they improve solution quality, both experimentally and analytically [15].

Our proposed ensemble learning implementation considers past ensembles of training datasets as part of the prediction process. Without loss of generality, let $\mathcal{D}$ be set of indices of past ensembles and each ensemble corresponds to a day in the past with lower indices representing more recent days, in particular, $d = 1$ corresponds to the 'present'. The set $\mathcal{D}$ is typically chosen to include 4-5 weeks of past data and either week days or week ends are chosen based on whether the present day is a week day or week end. Let $\theta(t, d)$ denote that weight associated with time step t in ensemble d. The joint matrix is now given by

$$\mathbf{Z}(\Theta) \equiv \begin{bmatrix} \mathbf{Y}_{|\mathcal{D}|,\text{tr}} & \cdots & \mathbf{Y}_{1,\text{tr}} & \mathbf{Y}_{\text{te}} \\ \Theta_{|\mathcal{D}|} \odot \Phi(\mathbf{X}_{|\mathcal{D}|,\text{tr}}) & \cdots & \Theta_1 \odot \Phi(\mathbf{X}_{1,\text{tr}}) & \Phi_{\text{te}} \end{bmatrix}, \tag{105}$$

where $\mathbf{Y}_{d,\text{tr}} \equiv [\mathbf{y}_{d,\text{tr}}(1) \cdots \mathbf{y}_{d,\text{tr}}(T_{\text{tr}})]$ and $\mathbf{X}_{d,\text{tr}} \equiv [\mathsf{B}_L \mathbf{x}_{d,\text{tr}}(1) \cdots \mathsf{B}_L \mathbf{x}_{d,\text{tr}}(T_{\text{tr}})]$ are the output and input matrices, respectively, corresponding to day d, Φ applies the (unknown) mapping function to each of the columns of its argument,

$$\Theta_d \equiv \left[\theta(1, d)\mathbf{e} \cdots \theta(T_{\text{tr}}, d)\mathbf{e}\right] \in \mathbb{R}^{h\times T_{\text{tr}}} \tag{106}$$

is a matrix of weights corresponding to day d, $\mathbf{e}$ is vector of 1s of dimension h, and $\odot$ is the component-wise (or Hadamard) product. Given $\{\theta(t, d)\}_{1\leq t\leq T_{\text{tr}}, d\in\mathcal{D}}$, the prediction problem is simply a matrix completion problem, which is solved by block coordinate descent. Here, the training data are given by the augmented matrices

$$\widetilde{\mathbf{Y}}_{\text{tr}} \equiv [\mathbf{Y}_{|\mathcal{D}|,\text{tr}} \cdots \mathbf{Y}_{1,\text{tr}}] \in \mathbb{R}^{n\times|\mathcal{D}|T_{\text{tr}}} \tag{107}$$

and

$$\widetilde{\Phi}_{\text{tr}}(\Theta) \equiv \left[\Theta_{|\mathcal{D}|} \odot \Phi(\mathbf{X}_{|\mathcal{D}|,\text{tr}}) \quad \cdots \quad \Theta_1 \odot \Phi(\mathbf{X}_{1,\text{tr}})\right] \in \mathbb{R}^{h\times|\mathcal{D}|T_{\text{tr}}}. \tag{108}$$

The joint matrix is given by

$$\mathbf{Z}(\Theta) \equiv \begin{bmatrix} \widetilde{\mathbf{Y}}_{\text{tr}} & \mathbf{Y}_{\text{te}} \\ \widetilde{\Phi}_{\text{tr}}(\Theta) & \Phi_{\text{te}} \end{bmatrix} \in \mathbb{R}^{(n+h)\times(|\mathcal{D}|T_{\text{tr}}+T_{\text{te}})}. \tag{109}$$

The overall prediction algorithm is depicted in Fig. 2. The procedure begins with an initial set of weights,

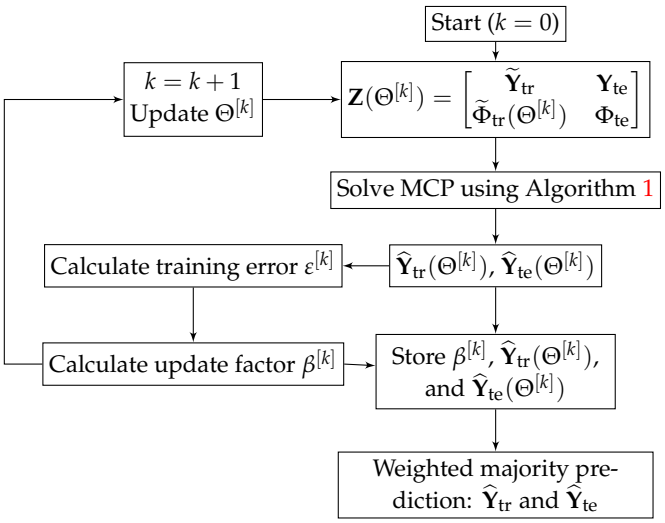


FIGURE 2: An Illustration of the Overall Prediction Algorithm with Adaptive Boosting

which can chosen in a variety of ways, e.g., equal weights: $\theta^{[0]}(t, d) \propto 1$ for all $t \in \{1, \ldots, T_{\text{tr}}\}$ and $d \in \mathcal{D}$, or weights that favor more recent days: $\theta^{[0]}(t, d) \propto d^{-1}$. (In our experiments, we use the former.) We then solve the matrix

completion problem using Algorithm 1 and use the results to calculate the training error, which is given as

$$\varepsilon^{[k]} = \frac{\sum_{(t,d)} \theta^{[k]}(t,d)\mathbb{1}\{\widehat{\mathbf{y}}^{[k]}_{d,\text{tr}}(t) \neq \mathbf{y}_{d,\text{tr}}(t)\}}{\sum_{(t,d)} \theta^{[k]}(t,d)}, \tag{110}$$

where $\widehat{\mathbf{y}}^{[k]}_{d,\text{tr}}(t)$ is the training output produced by Algorithm 1 using the weights determined in iteration k, $\Theta^{[k]}$. This error metric is bounded between 0 and 1, $\varepsilon^{[k]} = 0$ indicates that $\widehat{\mathbf{y}}^{[k]}_{d,\text{tr}}(t) = \mathbf{y}_{d,\text{tr}}(t)$ for all (t,d) pairs and, at the other extreme, $\varepsilon^{[k]} = 1$ indicates that $\widehat{\mathbf{y}}^{[k]}_{d,\text{tr}}(t) = \mathbf{y}_{d,\text{tr}}(t)$ for none of the (t,d) pairs. The error is used to calculate an *update factor* as follows:

$$\beta^{[k]} = \log \frac{1-\varepsilon^{[k]}}{\varepsilon^{[k]}}, \tag{111}$$

which is used to update the weight as

$$\theta^{[k+1]}(t,d) = \theta^{[k]}(t,d)\exp\left(\beta^{[k]}\mathbb{1}\{\widehat{\mathbf{y}}^{[k]}_{d,\text{tr}}(t) \neq \mathbf{y}_{d,\text{tr}}(t)\}\right). \tag{112}$$

The logarithm in (111) is used to mitigate potential computational instabilities due to large values. As $\varepsilon^{[k]} \to 0$, $\beta^{[k]} \to \infty$. The limit corresponds to the case where $\mathbb{1}\{\widehat{\mathbf{y}}^{[k]}_{d,\text{tr}}(t) \neq \mathbf{y}_{d,\text{tr}}(t)\} = 0$ for all (t,d) pairs (i.e., a perfect match) so that the exponential function on the right-hand side of (112) is 1. This means that in the case of a perfect match, the weights do not change. On the other hand, as $\varepsilon^{[k]} \to 1$, $\beta^{[k]} \to -\infty$ and $\theta^{[k+1]}(t,d) \to 0$ in this case.

Finally, the test predictions produced over the K iterations are combined to produce the final prediction. Let $\alpha^{[k]}$ denote the weight assigned to the prediction produced in iteration k and let it be defined as follows:

$$\alpha^{[k]} \equiv \frac{\beta^{[k]}}{\sum_{j=1}^{K}\beta^{[j]}}. \tag{113}$$

The combined predictions are given by

$$\widehat{\mathbf{Y}}_{\text{tr}} = \sum_k \alpha^{[k]}\widehat{\mathbf{Y}}_{\text{tr}}(\Theta^{[k]}) \tag{114}$$

and

$$\widehat{\mathbf{Y}}_{\text{te}} = \sum_k \alpha^{[k]}\widehat{\mathbf{Y}}_{\text{te}}(\Theta^{[k]}). \tag{115}$$

This is followed by a thresholding step to translate the predictions to labels in $\{0,1\}$. The prediction algorithm is summarized in Algorithm 3 below. Theorem 4.1 provides a bound on the training error (expressed as the number of mis-matched columns in $\widehat{\mathbf{Y}}_{\text{tr}}$). The error bound given in the theorem illustrates the fast reduction in error with number of iterations. As long as $\varepsilon^{[k]} < 0.5$, the error drops quickly. The bound we give is a specialization of the well known result in [15, Theorem 6] to our context.

Algorithm 3: Prediction with Ensemble Learning

Data: Joint matrix $\mathbf{Z}(\Theta^{[0]})$ and K (maximum number of iterations)
Result: $\widehat{\mathbf{Y}}_{\text{tr}}$ and $\widehat{\mathbf{Y}}_{\text{te}}$
Initialize: $k \leftarrow 0$ and initial weights $\Theta^{[0]} \equiv [\Theta^{[0]}_{|\mathcal{D}|} \cdots \Theta^{[0]}_1]$
while $k < K$ **do**
  $\widetilde{\Phi}_{\text{tr}}(\Theta^{[k]}) \leftarrow [\Theta^{[k]}_{|\mathcal{D}|} \odot \Phi(\mathbf{X}_{|\mathcal{D}|,\text{tr}}) \cdots \Theta^{[k]}_1 \odot \Phi(\mathbf{X}_{1,\text{tr}})]$
  Calculate $\widehat{\mathbf{Y}}_{\text{tr}}(\Theta^{[k]})$ and $\widehat{\mathbf{Y}}_{\text{te}}(\Theta^{[k]})$ using Algorithm 1
  Calculate $\varepsilon^{[k]}$ and $\beta^{[k]}$ using (110) and (111), respectively
  $k \leftarrow k+1$ and update $\Theta^{[k]}$ using (112)
end
Calculate the weighted majority predictions using (114)-(115) and apply thresholding (Algorithm 2).

Theorem 4.1 (Training Error of Algorithm 3). *Assume that $\theta^{[0]}(t,d) = 1$ for all (t,d)-pairs and let $\boldsymbol{\tau} \in \mathbb{R}^n$ denote a vector of thresholds. Let*

$$\epsilon(K) \equiv \left|\{(t,d) : \widehat{\mathbf{y}}^{[K]}_{d,\text{tr}}(t) \neq \mathbf{y}_{d,\text{tr}}(t)\}\right| \tag{116}$$

denote the training sample error at the end of step K. Then

$$\epsilon(K) \leq 2^K|\mathcal{D}|T_{\text{tr}}\prod_{k=0}^{K}\left(1-\varepsilon^{[k]}\right)^{\frac{n-\|\boldsymbol{\tau}\|_1}{n}}\left(\varepsilon^{[k]}\right)^{\frac{\|\boldsymbol{\tau}\|_1}{n}}. \tag{117}$$

Proof. Let (t,d) be such that $\mathbb{1}\{\widehat{\mathbf{y}}_{d,\text{tr}}(t) \neq \mathbf{y}_{d,\text{tr}}(t)\} = 1$. Since $\sum_{k=0}^{K}\alpha^{[k]} = 1$ we have that $\mathbf{y}_{d,\text{tr}}(t) = \sum_{k=0}^{K}\alpha^{[k]}\mathbf{y}_{d,\text{tr}}(t)$, thus

$$\sum_{k=0}^{K}\alpha^{[k]}\left\|\widehat{\mathbf{y}}^{[k]}_{d,\text{tr}}(t) - \mathbf{y}_{d,\text{tr}}(t)\right\|_1 \geq \|\boldsymbol{\tau}\|_1. \tag{118}$$

Multiplying both sides by $n^{-1}\sum_{j=0}^{K}\beta^{[j]}$, we get the inequality

$$\sum_{k=0}^{K}\frac{\beta^{[k]}}{n}\left\|\widehat{\mathbf{y}}^{[k]}_{d,\text{tr}}(t) - \mathbf{y}_{d,\text{tr}}(t)\right\|_1 \geq \frac{1}{n}\sum_{j=0}^{K}\beta^{[j]}\|\boldsymbol{\tau}\|_1. \tag{119}$$

Since $\mathbb{1}\{\widehat{\mathbf{y}}_{d,\text{tr}}^{[k]}(t) \neq \mathbf{y}_{d,\text{tr}}(t)\} \geq n^{-1}\|\widehat{\mathbf{y}}_{d,\text{tr}}^{[k]}(t) - \mathbf{y}_{d,\text{tr}}(t)\|_1$ for all k, we have that

$$\sum_{k=0}^{K} \beta^{[k]} \mathbb{1}\{\widehat{\mathbf{y}}_{d,\text{tr}}^{[k]}(t) \neq \mathbf{y}_{d,\text{tr}}(t)\} \geq \frac{1}{n}\sum_{j=0}^{K} \beta^{[j]}\|\boldsymbol{\tau}\|_1. \tag{120}$$

Define $\mathcal{M}^{[k]} \equiv \{(t,d) : \widehat{\mathbf{y}}_{d,\text{tr}}^{[k]}(t) \neq \mathbf{y}_{d,\text{tr}}(t)\}$. It follows from (120) that

$$\begin{aligned}
&\sum_{(t,d)} \theta^{[K+1]}(t,d) \\
&\quad \geq \sum_{(t,d)\in\mathcal{M}^{[K]}} \theta^{[0]}(t,d) \exp\Big(\sum_{k=0}^{K} \beta^{[k]} n^{-1}\|\boldsymbol{\tau}\|_1\Big) \\
&\quad = \epsilon(K) \prod_{k=0}^{K} \exp(\beta^{[k]} n^{-1}\|\boldsymbol{\tau}\|_1),
\end{aligned} \tag{121}$$

where $\epsilon(K) = \sum_{(t,d)\in\mathcal{M}^{[K]}} \theta^{[0]}(t,d)$ follows from the initialization assumption. Next, we have that

$$\begin{aligned}
&\sum_{(t,d)} \theta^{[K+1]}(t,d) = \sum_{(t,d)} \theta^{[K]}(t,d)\Big(\frac{1-\varepsilon^{[K]}}{\varepsilon^{[K]}}\Big)^{\mathbb{1}\{\widehat{\mathbf{y}}_{d,\text{tr}}^{[K]}(t)\neq \mathbf{y}_{d,\text{tr}}(t)\}} \\
&= \sum_{(t,d)} \theta^{[K]}(t,d)\Big(1 - \Big(1 - \frac{1-\varepsilon^{[K]}}{\varepsilon^{[K]}}\Big)\mathbb{1}\{\widehat{\mathbf{y}}_{d,\text{tr}}^{[K]}(t) \neq \mathbf{y}_{d,\text{tr}}(t)\}\Big) \\
&\quad = \sum_{(t,d)} \theta^{[K]}(t,d) - \sum_{(t,d)} \theta^{[K]}(t,d)\varepsilon^{[K]}\Big(1 - \frac{1-\varepsilon^{[K]}}{\varepsilon^{[K]}}\Big) \\
&\quad = \sum_{(t,d)} 2\theta^{[K]}(t,d)(1-\varepsilon^{[K]}),
\end{aligned} \tag{122}$$

where the second to last equality follows from (110). This implies that

$$\sum_{(t,d)} \theta^{[K+1]}(t,d) = 2^K|\mathcal{D}|T_{\text{tr}}\prod_{k=0}^{K}(1-\varepsilon^{[k]}). \tag{123}$$

Combining (121) with (123) and utilizing the definition of $\beta^{[k]}$, (111), completes the proof. □

4.2 Time Complexity Analysis

In each iteration, the block-coordinate descent algorithm (Algorithm 1) solves four least-squared (LS) problems but the solutions are given in closed form. The complexity of calculating $\widehat{\mathbf{U}}_{\text{tr}}^{[k]}$, using (38), is $O(r^2 T_{\text{tr}})$ (the complexity of matrix multiplication and inversion of a symmetric positive definite matrix). The complexity of updating $\widehat{\mathbf{V}}_{\text{tr}}^{[k]}$, using (40), is $O(\max\{r^3, T_{\text{tr}}^2 r, T_{\text{tr}}T_{\text{te}}r\})$, which is typically equal to $O(T_{\text{tr}}^2 r)$ as $T_{\text{tr}} > T_{\text{te}} > r$ will be the case in most settings. Similarly, updating $\widehat{\mathbf{V}}_{\text{te}}^{[k]}$ has a time complexity of $O(\max\{r^3, T_{\text{te}}^2 r, T_{\text{tr}}T_{\text{te}}r\})$, which is typically $O(T_{\text{tr}}T_{\text{te}}r)$. Therefore, the complexity in each iteration of Algorithm 1 is $O(\max\{r^3, T_{\text{tr}}^2 r, T_{\text{te}}^2 r, T_{\text{tr}}T_{\text{te}}r\}) = O(T_{\text{tr}}^2 r)$. Let K_{BCD} denote a the number of block-coordinate descent iterations, which can be determined a priori in accord with Theorem 3.1. Then, the overall complexity of Algorithm 1 is $O(K_{\text{BCD}}T_{\text{tr}}^2 r)$.

The analysis above implies that the time complexity of a single iteration of Algorithm 3 is $O(K_{\text{BCD}}|\mathcal{D}|^2 T_{\text{tr}}^2 r)$. The time complexity of the soft-thresholding algorithm is $O(|\mathcal{D}|^2 T_{\text{tr}}^2 n)$. The training error bound given in Theorem 4.1 can be used to estimate a number of iterations needed to get the training error to within a pre-specified error bound. Let K_{AB} be the number of iterations of Algorithm 3 to be performed. The overall complexity of our proposed approach is then $O(K_{\text{AB}}K_{\text{BCD}}T_{\text{tr}}^2 r + |\mathcal{D}|^2 T_{\text{tr}}^2 n)$. In most cases, we will have that $r > n$ so that the total complexity can be simply be stated as $O(K_{\text{AB}}K_{\text{BCD}}T_{\text{tr}}^2 r)$. One can further reduce the computational complexity of Algorithm 1 by utilizing more efficient matrix multiplication and inversion techniques, our time complexity bounds assume standard matrix algebra techniques are used.

5 Experimental Results

5.1 Network Descriptions and Performance Metrics

Simulated Data: We test the proposed method using both reproducible toy examples with simulated and a real-work high-resolution dataset. For the former, we generated synthetic data using a microscopic simulation model of a hypothetical network of four 4-leg intersections, depicted in Fig. 3. The model was developed using the

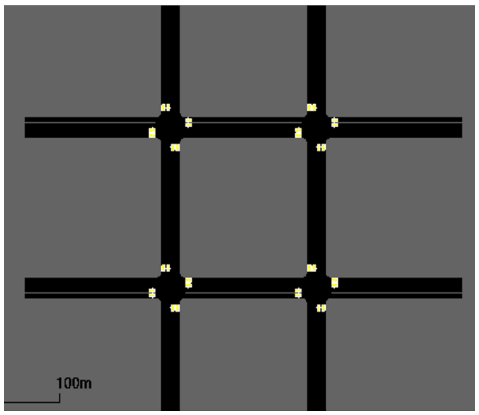


FIGURE 3: Layout of the Synthetic Network

open-source simulator SUMO (Simulation of Urban MObility) [38].

The intersections are operated using a fixed-time controller with a cycle length of 90 seconds. All approaches are 200 meters long and have two lanes with point sensors capable of recording high-resolution data located at the stop lines. We, thus, have 32 sensors in total. The simulation time horizon is 2 days (172,000 seconds). The average occupancy of these 32 sensors over the two day period is around 15%, hourly averaged occupancy profiles of two of the sensors are shown in Fig. 4. The experiments and results are presented in Sec. 5.3.

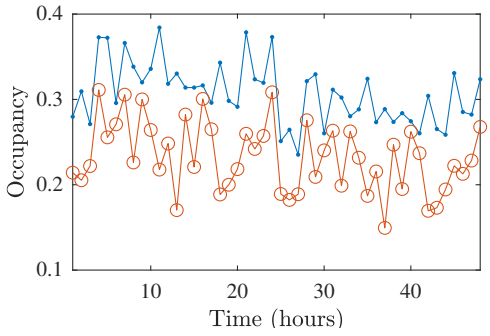


FIGURE 4: Occupancy Profiles of Two Sensors

Real-World Dataset: The second dataset is a real-world dataset provided by the Abu Dhabi Department of Transportation. The dataset was obtained for Al Zahiyah in downtown Abu Dhabi and consists of two corridors with the eleven intersections shown in Fig. 5. These intersections, currently operated by a commercial adaptive control tool, are located in the central business district in the city and are known to have heavy queuing during peak demand periods. For our experiments, we utilize second-by-second data obtained from point sensors along the direction that is highlighted in the figure for seven adjacent intersections. The sensors used in this study are those that the controller uses to make *signal extension* decisions. Each of the intersections has three or four through lanes in each direction with one sensor in each lane. Each of the intersections also has two left-turn lanes approaching the stop-line. The sensor stations used are located nearer the stop-lines of the intersections for intersections 4, 5, and 7, and the datasets for these sensor include the left-turn lanes in our experiments (as illustrated in Fig. 5). The total number of sensors in these experiments is 31. A schematic of one of the intersections is provided in Fig. 6. The sensors used are those in the advanced position (not the stop-line sensors).

Performance Metrics: We employ two performance metrics in our comparisons, the first is a traditional mean absolute error (MAE):

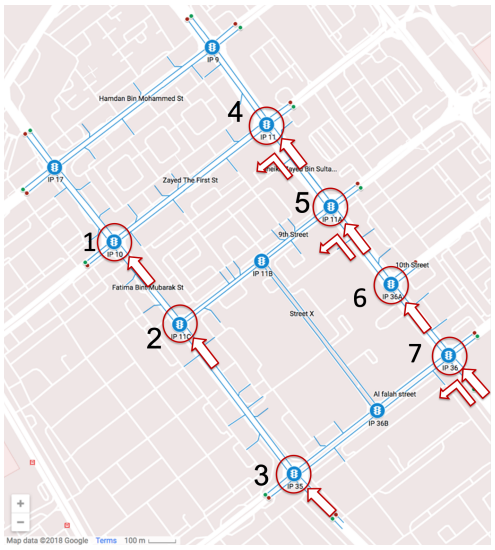


FIGURE 5: Layout of the Downtown Abu Dhabi Network

FIGURE 6: A Schematic of One of the Intersections Used in the Experiment

$$\epsilon_{\text{MAE}} = \frac{1}{nT_{\text{te}}} \sum_{t=1}^{T_{\text{te}}} \left\| \widehat{\boldsymbol{y}}_{\text{te}}(t) - \boldsymbol{y}_{\text{te}}(t) \right\|_1, \tag{124}$$

where $\widehat{\boldsymbol{y}}_{\text{te}}(t)$ and $\boldsymbol{y}_{\text{te}}(t)$ are columns $t \in \{1, \ldots, T_{\text{te}}\}$ of $\widehat{\mathbf{Y}}_{\text{te}}$ and $\mathbf{Y}_{\text{te}}$, respectively. The second metric is the Skorokhod M_1 metric:

$$d_{\mathrm{M}_1}(\widehat{\mathbf{Y}}_{\text{te}}, \mathbf{Y}_{\text{te}}) \equiv \inf_{\substack{(\widehat{Y},\widehat{y})\in\Pi(\widehat{\mathbf{Y}}_{\text{te}}),\\ (Y,y)\in\Pi(\mathbf{Y}_{\text{te}})}} \max\left\{ \|\widehat{Y} - Y\|_\infty, \|\widehat{y} - y\|_\infty \right\}, \tag{125}$$

where $\|\cdot\|_\infty$ is the uniform norm and $\Pi(\mathbf{Y})$ is the set of parametric representations of the rows of $\mathbf{Y}$. The Skorokhod M_1 metric is a particularly suitable metric for processes with jumps, more specifically, it allows for comparisons between processes with jumps and those free of jumps. We refer to [73] for a more detailed description of the metric. In our context, it is chosen for its ability to compare high-resolution signals.

Finally, all the tests are run on a 2.7 GHz intel Core i7 Processor with 16 GB of RAM.

5.2 Small Illustrative Example

Our first example utilizes a single sensor station form the real-world dataset to illustrate the proposed techniques. The three sensors located at intersection 2 in Fig. 5 are used in this example, and we denote the occupancies of the three sensors at time t by $s_1(t)$, $s_2(t)$, and $s_3(t)$ for lanes 1, 2, and 3, respectively. We set the lag to $L = 3$ seconds and the prediction horizon to $H = 2$ seconds. We wish to predict the occupancies of sensor 2 in time steps $t+12,\ldots,t+16$ utilizing past data from all three sensors. The structure of the joint matrix $\mathbf{Z}$ in (9) (without kernels) for this problem is

$$\mathbf{Z} = \begin{bmatrix} \mathbf{Y}_{\text{tr}} = \begin{bmatrix} s_2(t+2) & \cdots & s_2(t+11) \end{bmatrix} & \mathbf{Y}_{\text{te}} = \begin{bmatrix} s_2(t+12) & \cdots & s_2(t+16) \end{bmatrix} \\ \mathbf{X}_{\text{tr}} = \begin{bmatrix} s_3(t) & \cdots & s_3(t+9) \\ s_2(t) & \cdots & s_2(t+9) \\ s_1(t) & \cdots & s_1(t+9) \\ s_3(t-1) & \cdots & s_3(t+8) \\ s_2(t-1) & \cdots & s_2(t+8) \\ s_1(t-1) & \cdots & s_1(t+8) \\ s_3(t-2) & \cdots & s_3(t+7) \\ s_2(t-2) & \cdots & s_2(t+7) \\ s_1(t-2) & \cdots & s_1(t+7) \end{bmatrix} & \mathbf{X}_{\text{te}} = \begin{bmatrix} s_3(t+10) & \cdots & s_3(t+14) \\ s_2(t+10) & \cdots & s_2(t+14) \\ s_1(t+10) & \cdots & s_1(t+14) \\ s_3(t+9) & \cdots & s_3(t+14) \\ s_2(t+9) & \cdots & s_2(t+13) \\ s_1(t+9) & \cdots & s_1(t+13) \\ s_3(t+8) & \cdots & s_3(t+12) \\ s_2(t+8) & \cdots & s_2(t+12) \\ s_1(t+8) & \cdots & s_1(t+12) \end{bmatrix} \end{bmatrix}. \tag{126}$$

The entries of this matrix are provided in Fig. 7, and color coded according to whether the occupancy is 0 or 1. *Note the difference in the occupancy profiles corresponding to sensors 1 and 2 (s_1 and s_2) from times $t-1$ to $t+13$ (rows 6 and 7 in Fig. 7). Over the entire 15 second period, the aggregated occupancies are the same but the traffic patters are different.*

To solve the prediction problem, we set $\mu \equiv 10^{-6}$ and $r = 1$ and list the training results of the (unkernalized) problem without adaptive boosting in Table 1 and the testing results in Table 2. The convergence criterion used is $\epsilon^{\text{rel}}(k) < 10^{-4}$, where $\epsilon^{\text{rel}}(k)$ is the relative error in iteration k, calculated as

$$\epsilon^{\text{rel}}(k) \equiv \max\left\{ \frac{\|\mathbf{U}_{\text{tr}}^{[k]} - \mathbf{U}_{\text{tr}}^{[k-1]}\|_{\text{F}}}{\|\mathbf{U}_{\text{tr}}^{[k-1]}\|_{\text{F}}}, \frac{\|\mathbf{V}_{\text{tr}}^{[k]} - \mathbf{V}_{\text{tr}}^{[k-1]}\|_{\text{F}}}{\|\mathbf{V}_{\text{tr}}^{[k-1]}\|_{\text{F}}}, \frac{\|\mathbf{V}_{\text{te}}^{[k]} - \mathbf{V}_{\text{te}}^{[k-1]}\|_{\text{F}}}{\|\mathbf{V}_{\text{te}}^{[k-1]}\|_{\text{F}}} \right\}. \tag{127}$$

Note from Table 1 that beyond iteration $k = 5$, changes in the results are small. This illustrates the fast convergence rate of the algorithm. Convergence of the Block-Coordinate Descent (BCD) algorithm is illustrated in Fig. 8 for this example. We also see that the algorithm misclassified the sensor state at time $t+13$ in the testing sample in Table 2. The boosting procedure corrects this as depicted in Tables 3 and 4 below. Note that the iterations

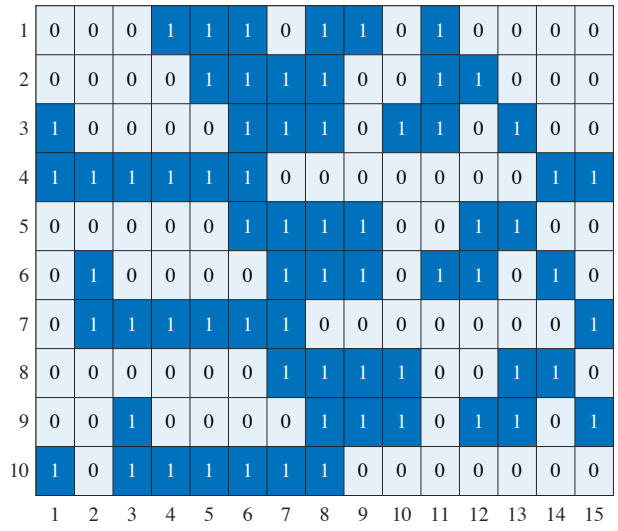


FIGURE 7: Entries of $\mathbf{Z}$ for the Small Illustrative Example

TABLE 1: Training Results Without Adaptive Boosting (G.T. = Ground Truth)

	k	$t+2$	$t+3$	$t+4$	$t+5$	$t+6$	$t+7$	$t+8$	$t+9$	$t+10$	$t+11$
$\widehat{\mathbf{Y}}_{\text{tr}}^{[k]}$	1	-0.0915	0.0884	0.1208	0.8264	0.8682	0.7644	-0.1001	0.5218	0.8142	0.0580
	5	0.0209	0.0535	0.1454	1.0242	1.0542	0.9922	0.1287	0.9451	0.9304	0.0597
	10	0.0208	0.0531	0.1439	1.0224	1.0520	0.9908	0.1278	0.9452	0.9330	0.0662
	end	0.0205	0.0523	0.1421	1.0218	1.0513	0.9908	0.1261	0.9464	0.9354	0.0686
	G.T.	0	0	0	1	1	1	0	1	1	0

TABLE 2: Testing Results Without Adaptive Boosting (A.T. = After Thresholding, G.T. = Ground Truth)

	k	$t+12$	$t+13$	$t+14$	$t+15$	$t+16$
$\widehat{\mathbf{Y}}_{\text{te}}^{[k]}$	1	0.4694	0.4454	0.4194	0.2675	0.2434
	5	0.1098	0.1289	0.1143	0.0635	0.0770
	10	0.2360	0.2351	0.1254	0.0651	0.0391
	end	0.2613	0.2626	0.1418	0.0730	0.0407
	A.T.	1	1	0	0	0
	G.T.	1	0	0	0	0

TABLE 3: Training Results With Adaptive Boosting (A.T. = After Thresholding, G.T. = Ground Truth)

	k	$t+2$	$t+3$	$t+4$	$t+5$	$t+6$	$t+7$	$t+8$	$t+9$	$t+10$	$t+11$
$\widehat{\mathbf{Y}}_{\text{tr}}(\Theta^{[k]})$	1	0.0205	0.0523	0.1421	1.0218	1.0513	0.9908	0.1261	0.9464	0.9354	0.0686
	2	0.0222	0.0568	0.1545	1.0192	1.0482	0.9865	0.1408	0.9469	0.9336	0.0765
	3	0.0247	0.0635	0.1655	1.0136	1.0432	0.9798	0.1804	0.9486	0.9324	0.0900
	4	0.0272	0.0711	0.1670	1.0045	1.0371	0.9715	0.2112	0.9527	0.9349	0.1104
$\widehat{\mathbf{Y}}_{\text{tr}}$	Eq. (114)	0.0229	0.0587	0.1531	1.0169	1.0466	0.9844	0.1581	0.9479	0.9341	0.0814
	A.T.	0	0	0	1	1	1	0	1	1	0
	G.T.	0	0	0	1	1	1	0	1	1	0

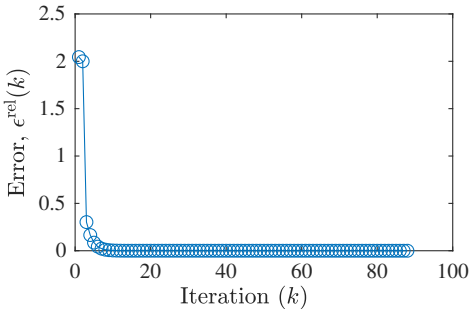


FIGURE 8: Convergence of BCD

TABLE 4: Testing Results With Adaptive Boosting (A.T. = After Thresholding, G.T. = Ground Truth)

	k	$t+12$	$t+13$	$t+14$	$t+15$	$t+16$
$\widehat{\mathbf{Y}}_{\text{te}}^{[k]}$	1	0.2613	0.2626	0.1418	0.0730	0.0407
	2	0.2583	0.2472	0.1293	0.0639	0.0345
	3	0.2546	0.2246	0.1120	0.0522	0.0268
	4	0.2609	0.2068	0.0984	0.0437	0.0211
$\widehat{\mathbf{Y}}_{\text{te}}$	Eq. (115)	0.1781	0.1597	0.0811	0.0389	0.0204
	A.T.	1	0	0	0	0
	G.T.	1	0	0	0	0

in the tables below are the *outer* iterations of the boosting procedure: the first outer iteration produces the results observed above (without boosting).

5.3 Simulation Experiments

Description of Experiments: We test the proposed method using the simulated data described in Sec. 5.1. Specifically, we test the approach under varying horizons and lags, $H \in \{1, 10, 60, 120\}$ and $L \in \{10, 30, 60, 120\}$ seconds. The historical data used for boosting consists of $|\mathcal{D}| = 2$ consecutive days in each experiment, and the number of training and testing time steps are $T_{\text{tr}} = 540$ and $T_{\text{te}} = 60$ seconds, respectively.

Impact of Choice of Lag (L) and Horizon (H): Since all entries in both $\widehat{\mathbf{Y}}_{\text{te}}$ and $\mathbf{Y}_{\text{te}}$ are binary, we have that $\epsilon_{\text{MAE}} \in [0,1]$ and $d_{\text{M}_1}(\widehat{\mathbf{Y}}_{\text{te}}, \mathbf{Y}_{\text{te}}) \in [0,1]$. We can hence measure *accuracy* using $1 - \epsilon_{\text{MAE}}$ and $1 - d_{\text{M}_1}$, where we have dropped the arguments from the latter where no confusion may arise. Mean accuracy and standard deviations (calculated over the 32 sensors) are summarized in Table 5

and Table 6 using different lags and prediction horizons. The entries in bold are those with the highest accuracy for each prediction horizon.

The average testing accuracy in Table 5 is no lower than 91.78% and reaches 95.22%, while the greatest standard deviation does not exceed 0.0761. The worst case prediction accuracy (when $|\mathcal{D}| \times L \times H = 2 \times 120 \times 60$) is greater than 76.9% with probability 97.5% (calculated as 0.9178 - 1.96×0.0761 corresponding to an accuracy that is 1.96 standard deviations below the mean). In other words, 97.5% of the cases have an accuracy that exceeds 76.9%. However, with a proper choice of the lag parameter (in this case $L = 60$ seconds), the worst case accuracy is greater than 87.6% with probability 97.5% (when $|\mathcal{D}| \times L \times H = 2 \times 60 \times 10$).

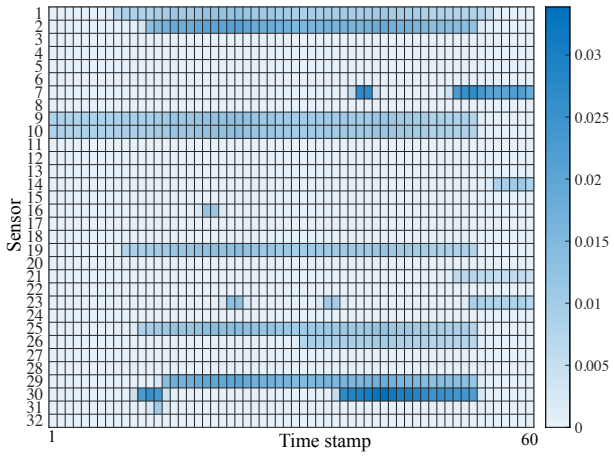
We see the same results when measuring accuracy using $1 - d_{\mathrm{M}_1}$ in Table 6. The worst case prediction accuracy (when $|\mathcal{D}| \times L \times H = 2 \times 120 \times 60$) is greater than 72.4% with probability 97.5% but with a proper choice of the lag parameter, the worse case accuracy is greater than 85.2% with probability 97.5%.

For both accuracy metrics used in Tables 5 and 6, we observe that the accuracy tends to decrease as H gets larger. However, increasing the lag L is not observed to improve the accuracy, a lag of $L = 60$ seconds seems to be the best choice in our experiment.

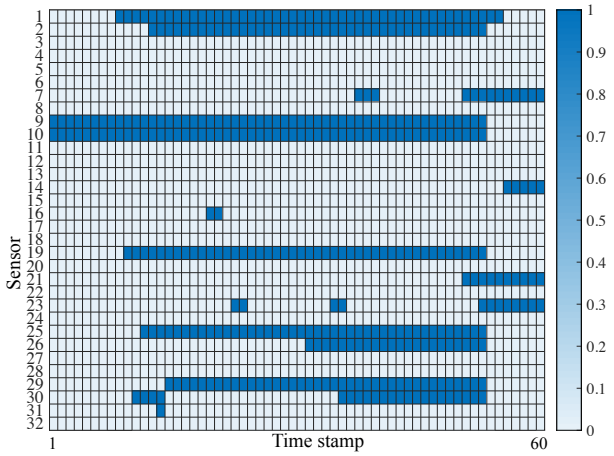
We further perform paired t-tests for different lags and horizons to check whether H and L play a significant role in the resulting predictions. We summarize the results in Table 7, where H is fixed and L is varied in the top part of the table, and H is varied and L is fixed in the bottom part of the table. The test result is either 1 or 0 indicating whether a hypothesis that the two models being compared are the same can be rejected (1) or not (0) at a 0.05 significance level. The results consistently suggest that the difference is significant in all cases when either L is varied or H is.

We provide an illustration of the resulting predictions for all 32 sensors in Figures 9 and 10, for a high-demand scenario and a low demand scenario, respectively. These are the resulting 10 second forecasts produced over a 60-second horizon. The figures illustrate the results before thresholding, after thresholding, and include the ground truth for comparison. These figures provide an illustration of how our method captures the sensor state dynamics.

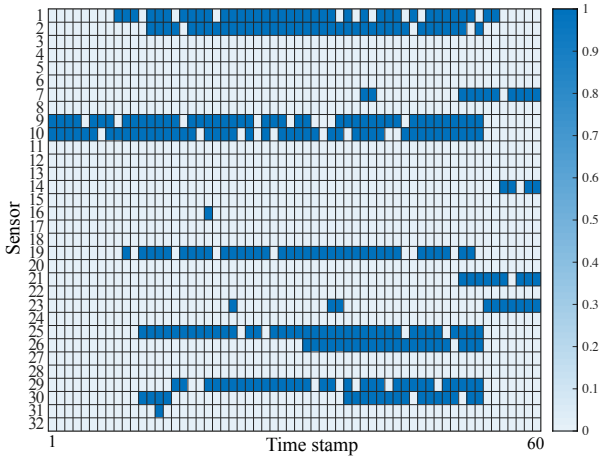
Impact of Rank Parameter (r) and Algorithm Convergence: We next investigate the performance of our approach for different choices of the input r in (25) and (28). We do so for the case $L = 60$ and $H = 10$ seconds. The objective values achieved when the algorithm converges for differ-



(a) Results Before Thresholding



(b) Results After Thresholding



(c) Ground Truth

FIGURE 9: Example Comparison Under High Demands

TABLE 5: Testing Accuracy Using MAE ($1-\epsilon_{\text{MAE}}$: Mean $\pm$ Std)

$\lvert\mathcal{D}\rvert \times L$	$H=1$	$H=10$	$H=60$
2×30	0.9395 ± 0.0340	0.9317 ± 0.0304	0.9268 ± 0.0268
2×60	$\mathbf{0.9522 \pm 0.0211}$	$\mathbf{0.9378 \pm 0.0313}$	$\mathbf{0.9325 \pm 0.0125}$
2×120	0.9448 ± 0.0311	0.9319 ± 0.0455	0.9178 ± 0.0761

TABLE 6: Testing Accuracy Using the Skorokhod M_1 Metric ($1-d_{M_1}$: Mean $\pm$ Std)

$\lvert\mathcal{D}\rvert \times L$	$H=1$	$H=10$	$H=60$
2×30	$\mathbf{0.9557 \pm 0.0520}$	0.9259 ± 0.0429	0.9111 ± 0.0457
2×60	0.9464 ± 0.0385	$\mathbf{0.9378 \pm 0.0313}$	$\mathbf{0.9269 \pm 0.0381}$
2×120	0.9387 ± 0.0401	0.9292 ± 0.0349	0.9155 ± 0.0977

TABLE 7: Paired t-test for Different Lags and Horizons (Significance Level = 0.05)

	$(H, \lvert\mathcal{D}\rvert \times L)$	$(1, 2\times 30)$	$(1, 2\times 60)$	$(1, 2\times 120)$
$(H, \lvert\mathcal{D}\rvert \times L)$	$(1, 2\times 30)$	0	1	1
	$(1, 2\times 60)$	1	0	1
	$(1, 2\times 120)$	1	1	0
	$(H, \lvert\mathcal{D}\rvert \times L)$	$(1, 2\times 30)$	$(10, 2\times 30)$	$(60, 2\times 30)$
$(H, \lvert\mathcal{D}\rvert \times L)$	$(1, 2\times 30)$	0	1	1
	$(10, 2\times 30)$	1	0	1
	$(60, 2\times 30)$	1	1	0

ent values of r are depicted in Fig. 11. We see that the lowest objective value is achieved when $r = 60$ indicating a highly sparse matrix (an order of magnitude smaller than both $nh \sim |\mathcal{D}|Ln = 3,840$ and $|\mathcal{D}|T_{\text{tr}} + T_{\text{te}} = 1,140$). Fig. 12 further illustrates the sublinear convergence rate when $r = 15$.

Comparison with Other Techniques: We benchmark our approach against the following prediction techniques:

1. Time series following generalized linear models (TS-GLM): We use the framework described in [14] to generalize a linear vector auto-regressive time series to binary data. We use a 'logit' link function for this purpose.

2. Support vector regression (SVR): We choose the SVR model with RBF kernels for comparison.

3. Recurrent neural networks (RNN): RNNs are powerful tools for time series. In our experiments, we employ a long short-term memory (LSTM) architecture [21], similar to the neural network architecture used in [53] to predict traffic flows; we use four layers (input, LSTM layer, fully-connected layer and output).

We compare these three techniques with two implementations of the proposed approach, the first employs block coordinate decent (BCD) without boosting, Algorithm 2, and the boosted-BCD (B-BCD) in Algorithm 3.

TABLE 8: Accuracy Comparisons

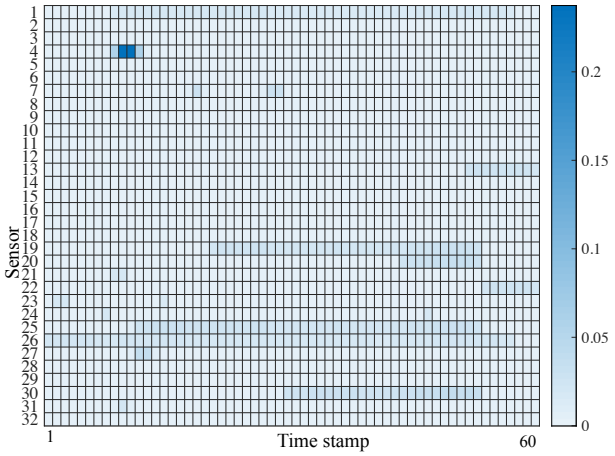
Method	$1-d_{M_1}$	$1-\epsilon_{\text{MAE}}$
BCD	0.9101 ± 0.0219	0.9289 ± 0.0215
B-BCD	0.9557 ± 0.0520	0.9522 ± 0.0211
TS-GLM	0.8209 ± 0.0166	0.8911 ± 0.0828
SVR	0.8754 ± 0.0697	0.8915 ± 0.0319
RNN	0.9473 ± 0.0315	0.9652 ± 0.0456

The accuracy results are summarized in Table 8. The results indicate that our B-BCD outperforms the other methods under both metrics, matched only by the RNN. The differences in accuracy are also significant at the 0.05 level, as per the paired t-tests shown in Tables 9 and 10.

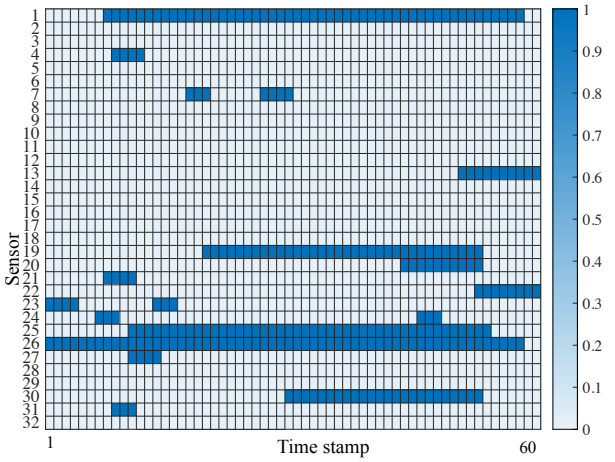
TABLE 9: Paired t-test for the Different Methods Under the Accuracy Metric $1-\epsilon_{\text{MAE}}$ (Significance Level = 0.05)

Method	BCD	B-BCD	TS-GLM	SVR	RNN
BCD	0	1	1	1	1
B-BCD	1	0	1	1	0
TS-GLM	1	1	0	0	1
SVR	1	1	0	0	1
RNN	1	0	1	1	0

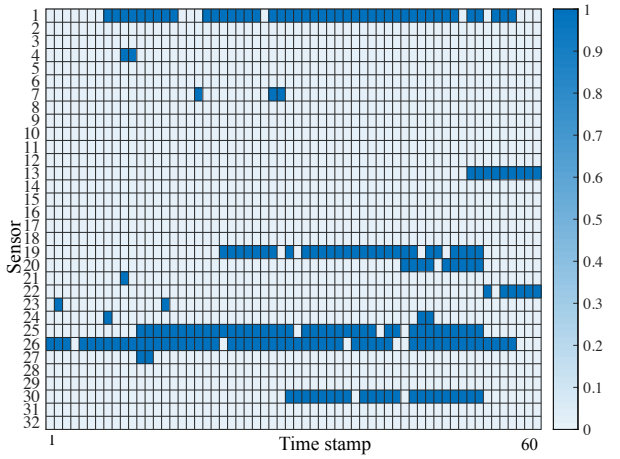
The two table show that we can reject the hypothesis that the accuracy achieved by B-BCD or RNN can be achieved by any of the other methods. It is not surprising that the RNN performs so well as a result of



(a) Results Before Thresholding



(b) Results After Thresholding



(c) Ground Truth

FIGURE 10: Example Comparison Under Low Demands

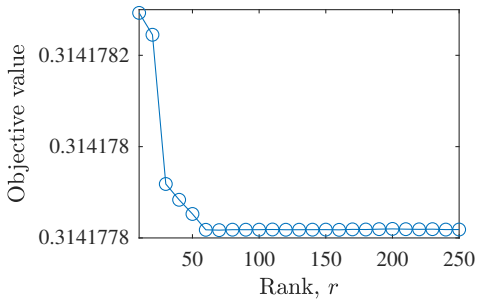


FIGURE 11: Relationship Between r and Objective Function Value at Convergence

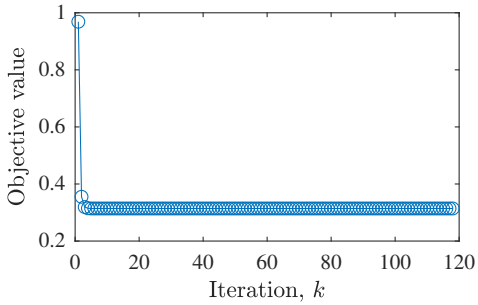


FIGURE 12: Sublinear Convergence Rate

TABLE 10: Paired t-test for the Different Methods Under the Accuracy Metric $1 - d_{\mathrm{M}_1}$ (Significance Level = 0.05)

Method	BCD	B-BCD	TS-GLM	SVR	RNN
BCD	0	1	1	1	1
B-BCD	1	0	1	1	0
TS-GLM	1	1	0	1	1
SVR	1	1	1	0	1
RNN	1	0	1	1	0

the sophisticated representations afforded by deep neural networks. It is, however, notable that RNNs are not amenable to real-time implementation due to the heavy computational costs required to train them. Moreover, interpreting the results is not straightforward with deep neural networks in general. These are the key advantages of our approach: B-BCD can achieve the performance accuracy of a deep neural network, but can be implemented in real-time, it is also easy to extract insights from the results.

5.4 Real World Example

Description of Experiments: The experiments conducted in this section are meant to test the performance of the

proposed approach on the larger real-world dataset, described in Sec. 5.1. We use seven weeks of high-resolution data, 49 days, from the beginning of the first week of December, 2018 to the end of the third week of January, 2019. We do not distinguish workdays from weekends (as inputs) in our experiments as the proposed method is essentially a dynamic learning approach that is adaptive to time-varying changes and within-week patterns. The horizons and lags tested in these experiments are $H \in \{1, 10, 60, 120\}$ and $L \in \{10, 30, 60, 120\}$ seconds, and the historical data used for boosting consists of $|\mathcal{D}| = 20$ consecutive days in each experiment, and the number of training and testing time steps are $T_{\text{tr}} = 500$ and $T_{\text{te}} = 100$ seconds, respectively. We benchmark our approach against the same prediction techniques described in Sec. 5.3 at the end of this section, specifically, we compare both BCD and B-BCD against TS-GLM, SVR, and RNN.

Impact of Choice of Lag (L) and Horizon (H): Again, we measure accuracy using $1 - \epsilon_{\text{MAE}}$ and $1 - d_{\text{M}_1}$, and we report mean accuracy and standard deviations (calculated over the 31 sensors) in Table 11 and Table 12 using different lags and prediction horizons. The entries in bold are those with the highest accuracy for each prediction horizon.

We see roughly the same pattern in both tables and similar results to those we observed in Sec. 5.3, namely that the accuracy decreases as H increases and that longer lags (L) do not necessarily mean improved accuracy. The mean accuracies (taken over all 31 sensors) is no lower than 79.01% and reaches 90.91%, while the corresponding standard deviations do not exceed 2.52%. The lowest accuracies under $1 - \epsilon_{\text{MAE}}$ and $1 - d_{\text{M}_1}$, respectively, exceed 75.07% and 76.94% in 97.5% of the cases.

We also conduct pair t-tests to investigate the impact of the variables H and L and present the results in Table 13 below. The top part of the table summarizes the results of changing L, the bottom part summarizes the effect of changing H. All results suggest that the differences between the different models are significant at the 0.05 level.

Choice of Rank Parameter (r) and Algorithm Convergence: We select the case $L = 30$ and $H = 10$ seconds to test the impact of the rank parameter (r). The objective values achieved when the algorithm converges under different values of r are depicted in Fig. 13. We see that the lowest objective value is achieved when $r = 95$ indicating a highly sparse matrix (two orders of magnitude smaller than both $nh \sim |\mathcal{D}|Ln = 18,600$ and $|\mathcal{D}|T_{\text{tr}} + T_{\text{te}} = 10,100$). Figures 14 and 15 illustrate the sub-linear convergence rate of the algorithm for the same inputs ($L = 30$ seconds, $H = 10$ seconds, and $r = 95$).

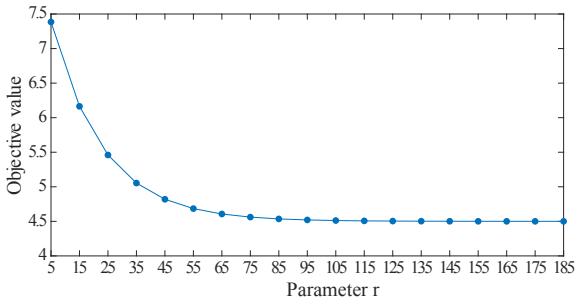


FIGURE 13: Relationship Between $r \geq \text{rank}(\widehat{\mathbf{Z}})$ and Objective Function Value at Convergence

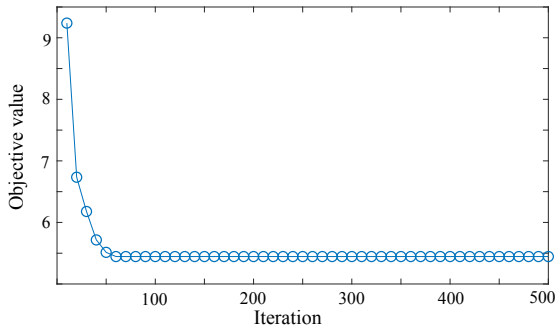


FIGURE 14: Convergence Rate

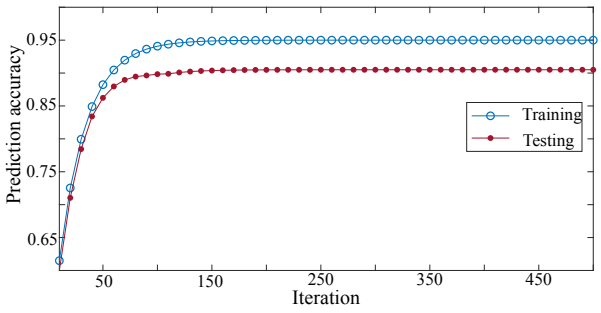


FIGURE 15: Illustration of Sub-Linear Convergence Rate

Comparisons: We now show the results of our comparisons. We first provide the accuracy results for all of the methods that we apply, averaged over all 31 sensors along with standard deviations in Table 14. The results suggest that our B-BCD outperforms all other methods except for the RNN. Paired t-tests were carried out under both accuracy metrics, and the results are listed in Tables 15 and 16. Under both metrics, we see that the differences between B-BCD and all other approaches (excluding RNN) are significant. Again, RNN is a good benchmark for compari-

TABLE 11: Testing Accuracy Using MAE for Real-World Example with 31 sensors ($1-\epsilon_{\text{MAE}}$: Mean $\pm$ Std)

$\lvert\mathcal{D}\rvert \times L$	$H=1$	$H=10$	$H=60$	$H=120$
20×10	0.8758 ± 0.0205	0.8609 ± 0.0225	0.8475 ± 0.0255	0.8138 ± 0.0199
20×30	$\mathbf{0.9091 \pm 0.0211}$	$\mathbf{0.9012 \pm 0.0269}$	$\mathbf{0.8539 \pm 0.0215}$	$\mathbf{0.8432 \pm 0.0197}$
20×60	0.8764 ± 0.0252	0.8631 ± 0.0215	0.8353 ± 0.0192	0.8155 ± 0.0197
20×120	0.8443 ± 0.0217	0.8401 ± 0.0145	0.8178 ± 0.0278	0.7901 ± 0.0201

TABLE 12: Testing Accuracy Using the Skorokhod M_1 Metric for Real-World Example with 31 sensors ($1-d_{M_1}$: Mean $\pm$ Std)

$\lvert\mathcal{D}\rvert \times L$	$H=1$	$H=10$	$H=60$	$H=120$
20×10	0.8618 ± 0.0197	0.8509 ± 0.0215	0.8453 ± 0.0221	0.8124 ± 0.0203
20×30	$\mathbf{0.8858 \pm 0.0213}$	$\mathbf{0.8719 \pm 0.0235}$	$\mathbf{0.8419 \pm 0.0225}$	$\mathbf{0.8332 \pm 0.0257}$
20×60	0.8517 ± 0.0201	0.8438 ± 0.0211	0.8359 ± 0.0197	0.8158 ± 0.0211
20×120	0.8313 ± 0.0227	0.8398 ± 0.0195	0.8268 ± 0.0218	0.8108 ± 0.0211

TABLE 13: Paired t-test for Different Lags and Horizons Using Real-World Data (Significance Level = 0.05)

$(H, \lvert\mathcal{D}\rvert \times L)$	$(1, 20\times 10)$	$(1, 20\times 30)$	$(1, 20\times 60)$	$(1, 20\times 120)$
$(1, 20\times 10)$	0	1	1	1
$(1, 20\times 30)$	1	0	1	1
$(1, 20\times 60)$	1	1	0	1
$(1, 20\times 120)$	1	1	1	0
$(H, \lvert\mathcal{D}\rvert \times L)$	$(1, 20\times 10)$	$(10, 20\times 30)$	$(60, 20\times 30)$	$(120, 20\times 30)$
$(1, 20\times 30)$	0	1	1	1
$(10, 20\times 30)$	1	0	1	1
$(60, 20\times 30)$	1	1	0	1
$(120, 20\times 30)$	1	1	1	0

son, but not implementation due to the heavy computational costs that come with fitting them. Our B-BCD approach can achieve accuracy that is comparable to a well-trained RNN, but offers computational advantages.

TABLE 14: Accuracy Comparisons for Real-World Problem

Method	$1-d_{M_1}$	$1-\epsilon_{\text{MAE}}$
BCD	0.8101 ± 0.0219	0.8589 ± 0.0215
B-BCD	0.8858 ± 0.0213	0.9091 ± 0.0211
TS-GLM	0.8018 ± 0.0608	0.8184 ± 0.0489
SVR	0.7867 ± 0.0161	0.8015 ± 0.0135
RNN	0.8703 ± 0.0305	0.9127 ± 0.0192

TABLE 15: Paired t-test for the Different Methods Under the Accuracy Metric $1-\epsilon_{\text{MAE}}$ (Significance Level = 0.05)

Method	BCD	B-BCD	TS-GLM	SVR	RNN
BCD	0	1	1	1	1
B-BCD	1	0	1	1	0
TS-GLM	1	1	0	0	1
SVR	1	1	0	0	1
RNN	1	0	1	1	0

TABLE 16: Paired t-test for the Different Methods Under the Accuracy Metric $1-d_{M_1}$ (Significance Level = 0.05)

Method	BCD	B-BCD	TS-GLM	SVR	RNN
BCD	0	1	1	1	1
B-BCD	1	0	1	1	0
TS-GLM	1	1	0	1	1
SVR	1	1	1	0	1
RNN	1	0	1	1	0

6 Conclusions

Our contribution can broadly be described as specializing contemporary convex optimization techniques to traffic prediction. These techniques have revolutionized a variety of applications that involve large volumes of data, from image processing to online recommender systems. However, they seem to have found little or no application in transportation science and traffic management. Our analysis demonstrates that one can solve large prediction problems in real-time (sub-linear convergence rate) and that the solutions obtained are block coordinate-wise minimizers. We also demonstrated that training error can be made arbitrarily small with ensemble learning. The

latter are typically used to amalgamate results from heterogeneous techniques. Our implementation uses historical data as predictors. We elected to perform ensemble learning in this way for the sake of interpretability of our results.

The analysis in this paper culminates in a bound on the training error. This result is a specialization of a well known bound that comes with the AdaBoost algorithm. Our bound allows for any type of thresholding. Our results do not say anything about how these bounds generalize to testing errors but our empirical results suggest that the out-of-sample errors are similar to the training error, with differences in accuracy that do not exceed 10%. Future work can be conducted along two separate lines: The first is generalization errors and bounds on out-of-sample errors. The second involves investigations of what should be considered an acceptable level of error. This will depend on the application; for example, for signal timing optimization a lag of three seconds in a sensor actuation prediction can trigger a signal status decision that results in unwanted congestion. We leave these questions to future research.

Acknowledgment

This work was supported by the NYUAD Center for Interacting Urban Networks (CITIES), funded by Tamkeen under the NYUAD Research Institute Award CG001 and by the Swiss Re Institute under the Quantum Cities™ initiative. The authors would also like to acknowledge in-kind support received from the Abu Dhabi Department of Transportation, in the form of the high-resolution traffic data that were used in our experiments. The opinions expressed in this article are those of the authors alone do not represent the opinions of CITIES or the Abu Dhabi Department of Transportation.

References

[1] Kevin Balke, Hassan Charara, and Ricky Parker. *Development of a traffic signal performance measurement system (TSPMS)*. Texas Department of Transportation, Report No. FHWA/TX-05/0-4422-2, 2005.

[2] Amir Beck and Luba Tetruashvili. On the convergence of block coordinate descent type methods. *SIAM journal on Optimization*, 23(4):2037–2060, 2013.

[3] Michael A Benjamin, Robert A Rigby, and D Mikis Stasinopoulos. Generalized autoregressive moving average models. *Journal of the American Statistical association*, 98 (461):214–223, 2003.

[4] Armelle Brun, Marharyta Aleksandrova, and Anne Boyer. Can latent features be interpreted as users in matrix factorization-based recommender systems? In *2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)*, volume 2, pages 226–233. IEEE, 2014.

[5] Emmanuel J Candès and Benjamin Recht. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 9(6):717–772, 2009.

[6] Emmanuel J. Candès and Terence Tao. The power of convex relaxation: Near-optimal matrix completion. *IEEE Transactions on Information Theory*, 56(5):2053–2080, 2009.

[7] Chenyi Chen, Jianming Hu, Qiang Meng, and Yi Zhang. Short-time traffic flow prediction with ARIMA-GARCH model. In *2011 IEEE Intelligent Vehicles Symposium (IV)*, pages 607–612, 2011.

[8] Peng Chen, Guizhen Yu, Xinkai Wu, Yilong Ren, and Yueguang Li. Estimation of red-light running frequency using high-resolution traffic and signal data. *Accident Analysis & Prevention*, 102:235–247, 2017.

[9] Benjamin Coifman and Michael Cassidy. Vehicle reidentification and travel time measurement on congested freeways. *Transportation Research Part A: Policy and Practice*, 36 (10):899–917, 2002.

[10] Eddie Curtis. *Adaptive Signal Control Technology, September 8, 2017, Retrived October 30, 2020*. Federal Highway Administration, 2017. URL https://www.fhwa.dot.gov/innovation/everydaycounts/edc-1/asct.cfm.

[11] Deepthi Dilip, Nikolaos Freris, and Saif Eddin Jabari. Sparse estimation of travel time distributions using Gamma kernels. In *Proceedings of the 96th Annual Meeting of the Transportation Research Board, Washington D.C.*, pages No. 17–02971, 2017.

[12] Deepthi Dilip, Nikolaos Freris, and Saif Eddin Jabari. Systems and methods for sparse travel time estimation. *US Patent Application*, pages 15/893,524, 2018.

[13] Yuguang Fang, Kenneth A Loparo, and Xiangbo Feng. Inequalities for the trace of matrix product. *IEEE Transactions on Automatic Control*, 39(12):2489–2490, 1994.

[14] Konstantinos Fokianos and Benjamin Kedem. Partial likelihood inference for time series following generalized linear models. *Journal of Time Series Analysis*, 25(2):173–197, 2004.

[15] Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

[16] Bidisha Ghosh, Biswajit Basu, and Margaret O'Mahony. Multivariate short-term traffic flow forecasting using time-series analysis. *IEEE Transactions on Intelligent Transportation Systems*, 10(2):246–254, 2009.

[17] Andrew Goldberg, Ben Recht, Junming Xu, Robert Nowak, and Jerry Zhu. Transduction with matrix completion: Three birds with one stone. In *Advances in neural information processing systems*, pages 757–765, 2010.

[18] Jianhua Guo, Billy M Williams, and Brian L Smith. Data collection time intervals for stochastic short-term traffic flow forecasting. *Transportation Research Record*, 2024(1):18–26, 2007.

[19] Qiangqiang Guo, Li Li, and Xuegang Jeff Ban. Urban traffic signal control with connected and automated vehicles: A survey. *Transportation research part C: emerging technologies*, 101:313–334, 2019.

[20] Frank E Harrell Jr. *Regression modeling strategies: with applications to linear models, logistic and ordinal regression, and survival analysis*. Springer International Publishing, Cham, Switzerland, 2015.

[21] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[22] Heng Hu and Henry X Liu. Arterial offset optimization using archived high-resolution traffic signal data. *Transportation Research Part C: Emerging Technologies*, 37:131–144, 2013.

[23] Heng Hu, Xinkai Wu, and Henry X Liu. Managing oversaturated signalized arterials: a maximum flow based approach. *Transportation Research Part C: Emerging Technologies*, 36:196–211, 2013.

[24] Jianming Hu, Pan Gao, Yunfei Yao, and Xudong Xie. Traffic flow forecasting with particle swarm optimization and support vector regression. In *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 2267–2268, 2014.

[25] Sherif Ishak and Haitham Al-Deek. Performance evaluation of short-term time-series traffic prediction model. *Journal of Transportation Engineering*, 128(6):490–498, 2002.

[26] Saif Eddin Jabari. *A stochastic model of macroscopic traffic flow: Theoretical foundations*. Ph.D. Dissertation. The University of Minnesota Twin Cities, Minneapolis, MN, 2012.

[27] Saif Eddin Jabari and Henry Liu. A stochastic model of traffic flow: Theoretical foundations. *Transportation Research Part B: Methodological*, 46(1):156–174, 2012.

[28] Saif Eddin Jabari and Henry Liu. A stochastic model of traffic flow: Gaussian approximation and estimation. *Transportation Research Part B: Methodological*, 47:15–41, 2013.

[29] Saif Eddin Jabari, Deepthi Dilip, DianChao Lin, and Bilal Thonnam Thodi. Learning traffic flow dynamics using random fields. *IEEE Access*, 7:130566–130577, 2019.

[30] Saif Eddin Jabari, Nikolaos Freris, and Deepthi Dilip. Sparse travel time estimation from streaming data. *Transportation Science*, 54(1):1–20, 2020.

[31] Prateek Jain, Praneeth Netrapalli, and Sujay Sanghavi. Low-rank matrix completion using alternating minimization. In *Proceedings of the 45th Annual ACM Symposium on Theory of Computing*, pages 665–674, 2013.

[32] Young-Seon Jeong, Young-Ji Byon, Manoel Mendonca Castro-Neto, and Said M Easa. Supervised weighting-online learning algorithm for short-term traffic flow prediction. *IEEE Transactions on Intelligent Transportation Systems*, 14(4):1700–1707, 2013.

[33] Yiannis Kamarianakis and Poulicos Prastacos. Forecasting traffic flow conditions in an urban network: Comparison of multivariate and univariate approaches. *Transportation Research Record*, 1857(1):74–84, 2003.

[34] Yiannis Kamarianakis and Poulicos Prastacos. Space–time modeling of traffic flow. *Computers & Geosciences*, 31(2): 119–133, 2005.

[35] Yiannis Kamarianakis, Wei Shen, and Laura Wynter. Real-time road traffic forecasting using regime-switching space-time models and adaptive LASSO. *Applied Stochastic Models in Business and Industry*, 28(4):297–315, 2012.

[36] Danqing Kang, Yisheng Lv, and Yuan-yuan Chen. Short-term traffic flow prediction with LSTM recurrent neural network. In *Proceedings of the 20th IEEE International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–6, 2017.

[37] Sung-Eun Kim and Debbie Niemeier. A weighted autoregressive model to improve mobile emissions estimates for locations with spatial dependence. *Transportation Science*, 35(4):413–424, 2001.

[38] Daniel Krajzewicz, Jakob Erdmann, Michael Behrisch, and Laura Bieker. Recent development and applications of SUMO-Simulation of Urban MObility. *International Journal on Advances in Systems and Measurements*, 5(3&4), 2012.

[39] S Vasantha Kumar and Lelitha Vanajakshi. Short-term traffic flow prediction using seasonal ARIMA model with limited input data. *European Transport Research Review*, 7(3):21, 2015.

[40] Semin Kwak and Nikolas Geroliminis. Traffic forecasting for freeway networks by a localized linear regression time series model with a graph data dimensional reduction method. In *Proceedings of the 19th Swiss Transport Research Conference*, pages 1–6, 2019.

[41] Li Li and Saif Eddin Jabari. Position weighted backpressure intersection control for urban networks. *Transportation Research Part B: Methodological*, 128:435–461, 2019.

[42] Li Li, Victor Okoth, and Saif Eddin Jabari. Backpressure control with estimated queue lengths for urban network traffic. *IET Intelligent Transport Systems*, In Press:1–9, 2020. URL https://arxiv.org/abs/2006.15549.

[43] Wenqing Li, Chunhui Zhao, and Furong Gao. Linearity evaluation and variable subset partition based hierarchical process modeling and monitoring. *IEEE Transactions on Industrial Electronics*, 65(3):2683–2692, 2017.

[44] Marco Lippi, Matteo Bertini, and Paolo Frasconi. 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, 2013.

[45] Henry X Liu and Jie Sun. Length-based vehicle classification using event-based loop detector data. *Transportation Research Part C: Emerging Technologies*, 38:156–166, 2014.

[46] Henry X Liu, Wenteng Ma, Heng Hu, Xinkai Wu, and Guizhen Yu. SMART-SIGNAL: Systematic monitoring of arterial road traffic signals. In *2008 11th International IEEE Conference on Intelligent Transportation Systems*, pages 1061–1066, 2008.

[47] Yipeng Liu, Haifeng Zheng, Xinxin Feng, and Zhonghui Chen. Short-term traffic flow prediction with Conv-LSTM. In *Proceedings of the 9th International Conference on Wireless Communications and Signal Processing (WCSP)*, pages 1–6, 2017.

[48] Jing Lu and Carolina Osorio. A probabilistic traffic-theoretic network loading model suitable for large-scale network analysis. *Transportation Science*, 52(6):1509–1530, 2018.

[49] Yisheng Lv, Yanjie Duan, Wenwen Kang, Zhengxi Li, and Fei-Yue Wang. Traffic flow prediction with big data: A deep learning approach. *IEEE Transactions on Intelligent Transportation Systems*, 16(2):865–873, 2014.

[50] Xiaolei Ma, Zhimin Tao, Yinhai Wang, Haiyang Yu, and Yunpeng Wang. Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. *Transportation Research Part C: Emerging Technologies*, 54:187–197, 2015.

[51] Xiaolei Ma, Zhuang Dai, Zhengbing He, Jihui Ma, Yong Wang, and Yunpeng Wang. Learning traffic as images: A deep convolutional neural network for large-scale transportation network speed prediction. *Sensors*, 17(4):818, 2017.

[52] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. Rectifier nonlinearities improve neural network acoustic models. In *Proceedings of the 30th International Conference on Machine Learning*, volume 28, pages 1–6, 2013.

[53] Jonathan Mackenzie, John F Roddick, and Rocco Zito. An evaluation of HTM and LSTM for short-term arterial traffic flow prediction. *IEEE Transactions on Intelligent Transportation Systems*, 20(5):1847–1857, 2018.

[54] Adolf May, Benjamin Coifman, Randall Cayford, and Greg Merritt. *Automatic diagnostics of loop detectors and the data collection system in the Berkeley highway lab*. California PATH Research Report UCB-ITS-PRR-2004-13, 2004. URL https://escholarship.org/uc/item/3qx70239.

[55] Adolf D May, Randall Cayford, Ben Coifman, and Greg Merritt. *Loop detector data collection and travel time measurement in the Berkeley highway laboratory*. California PATH Research Report UCB-ITS-PRR-2003-17, 2003. URL https://escholarship.org/uc/item/0248v7w8.

[56] Cheol Oh, Stephen G Ritchie, and Jun-Seok Oh. Exploring the relationship between data aggregation and predictability to provide better predictive traffic information. *Transportation Research Record*, 1935(1):28–36, 2005.

[57] Carolina Osorio and Gunnar Flötteröd. Capturing dependency among link boundaries in a stochastic dynamic network loading model. *Transportation Science*, 49(2):420–431, 2014.

[58] Carolina Osorio and Vincenzo Punzo. Efficient calibration of microscopic car-following models for large-scale stochastic network simulators. *Transportation Research Part B: Methodological*, 119:156–173, 2019.

[59] Mohsen Ramezani, Jack Haddad, and Nikolas Geroliminis. Dynamics of heterogeneity in urban networks: aggregated traffic modeling and hierarchical control. *Transportation Research Part B: Methodological*, 74:1–19, 2015.

[60] Benjamin Recht, Maryam Fazel, and Pablo A Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM Review*, 52(3):471–501, 2010.

[61] T. Seo, A. Bayen, T. Kusakabe, and Y. Asakura. Traffic state estimation on highway: A comprehensive survey. *Annual Reviews in Control*, 43:128–151, 2017.

[62] Khaled Shaaban, Muhammad Asif Khan, and Rida Hamila. Literature review of advancements in adaptive ramp metering. *Procedia Computer Science*, 83:203–211, 2016.

[63] Alexander Skabardonis, Karl Petty, Hisham Noeimi, Daniel Rydzewski, and Pravin Varaiya. I-880 field experiment: Data-base development and incident delay estimation procedures. *Transportation Research Record*, 1554(1): 204–212, 1996.

[64] Edward J Smaglik, Anuj Sharma, Darcy M Bullock, James R Sturdevant, and Gary Duncan. Event-based data collection for generating actuated controller performance measures. *Transportation Research Record*, 2035(1):97–106, 2007.

[65] Alex J Smola and Bernhard Schölkopf. A tutorial on support vector regression. *Statistics and Computing*, 14(3):199–222, 2004.

[66] James H Stock and Mark W Watson. Vector autoregressions. *Journal of Economic Perspectives*, 15(4):101–115, 2001.

[67] Lu Sun. Stochastic projection-factoring method based on piecewise stationary renewal processes for mid-and long-term traffic flow modeling and forecasting. *Transportation Science*, 50(3):998–1015, 2015.

[68] Johan AK Suykens and Joos Vandewalle. Least squares support vector machine classifiers. *Neural Processing Letters*, 9(3):293–300, 1999.

[69] Huachun Tan, Yuankai Wu, Bin Shen, Peter J Jin, and Bin Ran. Short-term traffic prediction based on dynamic tensor completion. *IEEE Transactions on Intelligent Transportation Systems*, 17(8):2123–2133, 2016.

[70] Eleni Vlahogianni and Matthew Karlaftis. Temporal aggregation in traffic data: Implications for statistical characteristics and model choice. *Transportation Letters*, 3(1):37–49, 2011.

[71] Wenyi Wang and Albert K Wong. Autoregressive model-based gear fault diagnosis. *Journal of Vibration and Acoustics*, 124(2):172–179, 2002.

[72] Yinhai Wang and Nancy L Nihan. Dynamic estimation of freeway large-truck volumes based on single-loop measurements. *Intelligent Transportation Systems*, 8(3):133–141, 2004.

[73] Ward Whitt. *Stochastic-process limits: An introduction to stochastic-process limits and their application to queues*. Springer–Verlag, New York, NY, 2002.

[74] Billy M Williams and Lester A Hoel. Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results. *Journal of Transportation Engineering*, 129(6):664–672, 2003.

[75] Chun-Hsin Wu, Jan-Ming Ho, and Der-Tsai Lee. Travel-time prediction with support vector regression. *IEEE Transactions on Intelligent Transportation Systems*, 5(4):276–281, 2004.

[76] Xinkai Wu and Henry X Liu. Using high-resolution event-based data for traffic modeling and control: An overview. *Transportation Research Part C: Emerging Technologies*, 42:28–43, 2014.

[77] Xinkai Wu, Henry X Liu, and Douglas Gettman. Identification of oversaturated intersections using high-resolution traffic signal data. *Transportation Research Part C: Emerging Technologies*, 18(4):626–638, 2010.

[78] Yangyang Xu and Wotao Yin. A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion. *SIAM Journal on Imaging Sciences*, 6(3):1758–1789, 2013.

[79] Mehmet Yildirimoglu and Nikolas Geroliminis. Experienced travel time prediction for congested freeways. *Transportation Research Part B: Methodological*, 53:45–63, 2013.

[80] Guohui Zhang and Yinhai Wang. A gaussian kernel-based approach for modeling vehicle headway distributions. *Transportation Science*, 48(2):206–216, 2013.

[81] Fangfang Zheng, Saif Eddin Jabari, Henry Liu, and DianChao Lin. Traffic state estimation using stochastic Lagrangian dynamics. *Transportation Research Part B: Methodological*, 115:143–165, 2018.