Appendix E

PAPER E

Title: Online Edge Flow Imputation on Networks

Authors: **R. Money**, J. Krishnan, B. Beferull-Lozano, E. Isufi

Journal: IEEE Signal Processing Letters 2022

Online Edge Flow Imputation on Networks

R. Money, J. Krishnan, B. Beferull-Lozano, E. Isufi

Abstract: An online algorithm for missing data imputation for networks with signals defined on the edges is presented. Leveraging the prior knowledge intrinsic to real-world networks, we propose a bi-level optimization scheme that exploits the causal dependencies and the flow conservation, respectively via (i) a sparse line graph identification strategy based on a group-Lasso and (ii) a Kalman filtering-based signal reconstruction strategy developed using simplicial complex (SC) formulation. The advantages of this first SC-based attempt for time-varying signal imputation have been demonstrated through numerical experiments using EPANET models of both synthetic and real water distribution networks.

E.1 Introduction

Multivariate time series analysis is paramount in sensor, brain, and social networks, to name a few. Data generated from such interdependent systems can be represented as a time-varying graph, in which the recorded signals may be linked to the nodes [100, 101], or the edges [102], depending on the task at hand. Many applications including anomaly detection [103], time series forecasting [104], and missing data imputation [6] can benefit from learning and exploiting the graph structure. Among these applications, it is worth paying special attention to the missing data imputation [6, 18, 19] since many real-world systems are partially observed because of Re.g., sensor or communication failure, or simply the impossibility to have sensors in all locations. This paper focuses on time-varying data imputation on the edges of networks, such as water or traffic networks, referred to as *flow-based networks*. While there are methods for imputing data at the nodes [6, 18, 19, 105, 106], extending them to flow-based networks is not immediate.

Imputation in flow-based networks can benefit from simplicial complex (SC) formulations [45, 107], using algebraic tools from Hodge theory [108], [109] to encapsulate the adjacencies among the flow signals, e.g., the flow conservation in the network. In addition to this spatial information that SC encapsulates, one can also exploit the temporal priors, such as causal dependencies among the signals [1–4, 11, 21–23]. The flow signals are mostly interdependent in real-world systems, and their dependencies are often time-lagged in nature and cannot be observed physically. For instance, the flow in a pipe of a water network can influence the flow in another non-directly con-

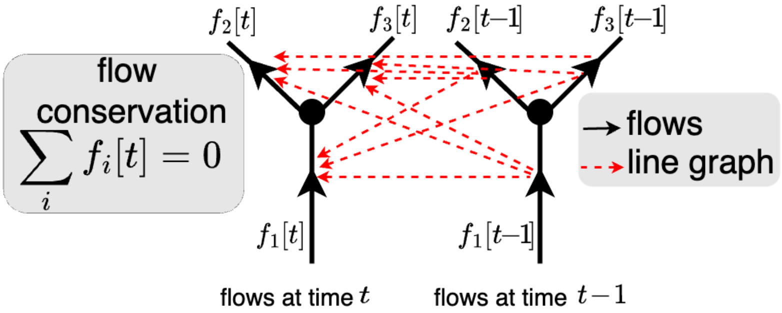

Figure E.1: Causal influence of $(t-1)$-th flows on t-th flows, represented using a line graph.

nected pipe in a time-lagged way. Similarly, a traffic block on a road can causally affect the traffic on another road. In such real-world networks, imputation can be enhanced by exploiting causal interactions between the flows. Imputation strategies utilizing both spatial and temporal dependencies have not been explored in flow-based networks.

This paper proposes a data imputation algorithm exploiting the spatio-temporal priors related to flow conservation and causal dependencies among flows. The algorithm learns a line graph connecting the flows, which stands in for an abstract representation of the time-lagged causal dependencies, as illustrated in Fig. E.1. One major challenge here is that a batch-based offline strategy is impractical in applications requiring real-time imputation of streaming flows. The proposed strategy learns a line graph in an online fashion. Using the learned line graph at each time step, a flow-conservation-based Kalman filter estimates the missing flows from streaming partial observations. The main contributions of this work are:

1. A method to estimate sparse causal dynamic dependencies among flows. This is achieved via a vector autoregressive model and a group-Lasso-based optimization framework. The latter is solved in an online fashion via composite objective mirror descent.

2. A Kalman-filter-based data imputation technique for streaming flows by exploiting the learned causality and the flow conservation devised via simplicial complexes.

3. The proposed algorithm can impute permanently unobserved flows, benefiting from the joint exploitation of the flow conservation and the causal dependencies.

To the best of our knowledge, this is the first work that considers multivariate time series data over simplicial complex. This work opens the door to the exploitation of learned line graphs and adjacency relationships among the time-varying signals over simplices (e.g., edge flows), which is useful in various applications such as forecasting, control strategy design, and change point detection.

E.2 Preliminaries

Consider a physically connected network $\mathcal{G} \triangleq (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ and $\mathcal{E}$ denote the sets of nodes and edges with cardinalities $V \triangleq |\mathcal{V}|$ and $E \triangleq |\mathcal{E}|$, respectively. We consider a flow-based network, for example, a water network with nodes as junctions, edges as pipes, and water flows as signals on the edges.

E.2.1 Modelling Flow Conservation in a Simplicial Complex

Given the set of nodes $\mathcal{V}$, a k-simplex $\mathcal{S}^k$ is a subset of $\mathcal{V}$ having $k+1$ distinctive elements [34], [35]. A simplicial complex (SC) of order K, denoted as Ψ^K, is a set of k-simplices for $k=0,1\ldots,K$ such that a simplex $\mathcal{S}^k\in\Psi^K$ only if all of its subsets also belong to Ψ^K. The typical low-order simplices, named after their geometrical shapes, are nodes (0-simplex), edges defined by two nodes (1-simplex), and triangles defined by three nodes (2-simplex). Let the number of k-simplices in Ψ^K be N_k. The proximities between different k-simplices in an SC can be represented using an incidence matrix $\mathbf{B}_k \in \mathbb{R}^{N_{k-1}\times N_k}, k \geq 1$, where the row and the column indices of $\mathbf{B}_k$ correspond to $(k-1)$- and k-simplices, respectively. The structure of an SC is encoded by Hodge Laplacians, constructed using $\mathbf{B}_k$'s as

$$\mathbf{L}_k = \begin{cases} \mathbf{B}_{k+1}\mathbf{B}_{k+1}^\top, & \text{for } k=0, \\ \mathbf{B}_k^\top\mathbf{B}_k + \mathbf{B}_{k+1}\mathbf{B}_{k+1}^\top, & \text{for } 1\leq k\leq K-1, \\ \mathbf{B}_K^\top\mathbf{B}_K, & \text{for } k=K, \end{cases} \tag{E.1}$$

where $\mathbf{L}_0$ is the graph Laplacian. The higher-order Laplacians $\mathbf{L}_k$, for $1\leq k\leq K-1$, consist of two terms: i) the *lower Laplacian*, $\mathbf{L}_k^l \triangleq \mathbf{B}_k^\top\mathbf{B}_k$, which encodes the adjacencies w.r.t. next-low-order simplices; and ii) the *upper Laplacian*, $\mathbf{L}_k^u \triangleq \mathbf{B}_{k+1}\mathbf{B}_{k+1}^\top$, which encodes the adjacencies w.r.t. next-high-order simplices.

In a SC, k-simplex signals are mappings from k-simplices to the real set $\mathbb{R}$. The 0-simplex, 1-simplex, and 2-simplex signals reside on the nodes, edges, and triangles, respectively. For flow-based networks, we consider 1-simplex signals or simply the flow signals. The flow signal at time t between two nodes i and j is defined as $f_{(i,j)}[t] = -f_{(j,i)}[t]$, $\forall$ $(i,j)\in\mathcal{E}$. We stack the flows into a vector $\tilde{\mathbf{f}}[t] = [f_1[t]\ f_2[t]\ \ldots\ f_E[t]]^\top$. The node-to-edge incidence matrix $\mathbf{B}_1 \in \mathbb{R}^{V\times E}$ has entries $\mathbf{B}_1(m,n) = 1$, if the flow n is leaving the node m, -1 if entering the node m, and 0 if the flow is not connected to m. According to the flow conservation principle, the sum of flows entering and leaving a node is zero, i.e., $\mathbf{B}_1\tilde{\mathbf{f}}[t] = \mathbf{0}\in\mathbb{R}^V$ [20]. The first-order lower Laplacian $\mathbf{L}_1^l$, can be used to model the flow conservation since it describes the relationship among the edges incidenting on a node, which is given by

$$\|\mathbf{B}_1\tilde{\mathbf{f}}[t]\|_2^2 = \tilde{\mathbf{f}}[t]^\top\mathbf{B}_1^\top\mathbf{B}_1\tilde{\mathbf{f}}[t] = \tilde{\mathbf{f}}[t]^\top\mathbf{L}_1^l\tilde{\mathbf{f}}[t] = 0. \tag{E.2}$$

One can also exploit the edge-to-triangle relationship of flows using $\mathbf{B}_2$, but we do not consider it since there is no contextual prior associated with $\mathbf{B}_2$.

E.2.2 Modelling Causal Dependencies using Line Graphs

We also take advantage from the fact that flows in a real-world network exhibit causal interactions. We construct a dynamic line graph connecting the flows using a P-th order dynamic VAR model to describe the time-lagged causal dependencies among the flows:

$$\tilde{\mathbf{f}}[t] = \sum_{p=1}^{P} \left[\tilde{\mathbf{A}}^{(p)}[t]\tilde{\mathbf{f}}[t-p] + \mathbf{b}^{(p)}[t] \right] + \mathbf{u}[t], \tag{E.3}$$

where $\tilde{\mathbf{A}}^{(p)}[t] \in \mathbb{R}^{E \times E}$ is the unknown weighted adjacency matrix of the line graph that captures the influence of the p-th time-lagged vector flow on the vector flow at time t, and $\mathbf{u}[t]$ is the process noise, Rwhich is assumed to be temporarily white and zero mean. The term $\mathbf{b}^{(p)}[t] \in \mathbb{R}^{E}$ is the bias component, which makes the model slightly different from a standard VAR model. We include the bias term since the normalization of the flow signals, which is a requirement for the subsequent formulation, cannot easily be achieved for permanently unobserved flows. Using an augumented matrix $\mathbf{A}^{(p)}[t] = [\tilde{\mathbf{A}}^{(p)}[t] \;\; \mathbf{b}^{(p)}[t]] \in \mathbb{R}^{E \times E+1}$ and the signal vector $\mathbf{f}[t] = [\tilde{\mathbf{f}}[t]^{\top}; 1]^{\top} \in \mathbb{R}^{E+1}$, (E.3) can be compactly written as

$$\mathbf{f}[t] = \sum_{p=1}^{P} \mathbf{A}^{(p)}[t]\mathbf{f}[t-p] + \mathbf{u}[t]. \tag{E.4}$$

E.3 Problem formulation

Assume that at a particular time t, only a subset of flows is observable. The observed flow vector is $\mathbf{f}_o[t] = \mathbf{M}[t]\mathbf{f}[t] \in \mathbb{R}^{E+1}$, where $\mathbf{M}[t] \in \mathbb{R}^{(E+1)\times(E+1)}$ is a diagonal masking matrix, with $\mathbf{M}(n,n)[t] = 0$ if the *n-th* flow is missing and $\mathbf{M}(n,n)[t] = 1$, otherwise. In this setting, some flows can be permanently unobserved. The goal is to find in an online fashion both a sequence of line graphs $\{\mathbf{A}^{(p)}[t]\}_{p,t}$, representing the causal dependencies between flows and the original signal $\mathbf{f}[t]$ from the partial observation $\mathbf{f}_o[t]$.

E.4 Online estimation of the line graph and data

A naive one-step optimization strategy to estimate $\mathbf{A}^{(p)}[t]$ and $\mathbf{f}[t]$ leads to nonconvex formulations that are difficult to solve [6]. Hence, we propose a bi-level optimization problem with the following steps: *i*) *signal reconstruction-* missing flows are estimated using the observed flows by assuming a known line graph topology; and *ii*) *line graph identification-* line graph is estimated using the reconstructed signals.

E.4.1 Signal Reconstruction

Assume that we have an estimate at time t of the topology $\hat{\mathbf{A}}^{(p)}[t]$, $\forall p$ and estimates of P previous flow values $\{\hat{\mathbf{f}}[t-p]\}_{p=1}^{P}$. We propose a Kalman-filtering-based strategy for signal reconstruction, and to facilitate the formulation, the available data are arranged as

$$\hat{\mathbf{A}}^{\mathcal{S}}[t] \triangleq \left[\begin{array}{c|c} \multicolumn{2}{c}{\underbrace{\hat{\mathbf{A}}^{(1:P)}[t]}_{E \times P(E+1)}} \\ \mathbf{I}_{P(E+1)-E} & \underbrace{\mathbf{0}}_{(P(E+1)-E) \times E} \end{array}\right], \mathbf{C}^{\mathcal{S}}[t] \triangleq \left[\begin{array}{c|c} \underbrace{\mathbf{M}[t]}_{(E+1) \times (E+1)} & \underbrace{\mathbf{0}}_{(E+1) \times (P-1)(E+1)} \\ \underbrace{\mathbf{0}}_{(P-1)(E+1) \times (E+1)} & \mathbf{I}_{(P-1)(E+1)} \end{array}\right],$$

$$\begin{aligned} \mathbf{y}^{\mathcal{S}}[t] &\triangleq [\mathbf{f}_o[t]^\top; \hat{\mathbf{f}}[t-1 : t-P+1]^\top]^\top, \\ \hat{\mathbf{f}}^{\mathcal{S}}[t] &\triangleq [\hat{\mathbf{f}}[t]^\top; \hat{\mathbf{f}}[t-1]^\top; \ldots; \hat{\mathbf{f}}[t-P+1]^\top]^\top, \end{aligned} \tag{E.5}$$

where $\hat{\mathbf{A}}^{(1:P)}[t] = [\hat{\mathbf{A}}^{(1)}[t], \ldots, \hat{\mathbf{A}}^{(P)}[t]]$ and $\mathbf{I}_N$ denotes $N \times N$ identity matrix. A state-space representation capturing the VAR relationships (E.15) and the missing data modelling is

$$\hat{\mathbf{f}}^{\mathcal{S}}[t] = \hat{\mathbf{A}}^{\mathcal{S}}[t]\hat{\mathbf{f}}^{\mathcal{S}}[t-1] + \mathbf{v}_t, \tag{E.6}$$

$$\mathbf{y}^{\mathcal{S}}[t] = \mathbf{C}^{\mathcal{S}}[t]\hat{\mathbf{f}}^{\mathcal{S}}[t] + \mathbf{w}_t, \tag{E.7}$$

where $\hat{\mathbf{f}}^{\mathcal{S}}[t] \in \mathbb{R}^{P(E+1)}$ is current state vector, $\hat{\mathbf{A}}^{\mathcal{S}}[t] \in \mathbb{R}^{P(E+1) \times P(E+1)}$ is the state transition matrix and $\mathbf{y}^{\mathcal{S}}[t] \in \mathbb{R}^{P(E+1)}$, and $\mathbf{C}^{\mathcal{S}} \in \mathbb{R}^{P(E+1) \times P(E+1)}$ are the observed signal and the observation matrix, respectively. The process noise $\mathbf{v}_t$ and the observation noise $\mathbf{w}_t$ are assumed zero-mean Gaussian. The optimal estimates of $\hat{\mathbf{f}}^{\mathcal{S}}[t]$ can be obtained using a Kalman filter (KF) [33].

1) Prediction:

$$\hat{\mathbf{f}}^{\mathcal{S}}_{t|t-1} = \hat{\mathbf{A}}^{\mathcal{S}}[t]\hat{\mathbf{f}}^{\mathcal{S}}_{t-1|t-1}, \tag{E.8}$$

$$\mathbf{P}_{t|t-1} = \hat{\mathbf{A}}^{\mathcal{S}}[t]\mathbf{P}_{t-1|t-1}\hat{\mathbf{A}}^{\mathcal{S}}[t]^\top + \mathbf{Q}_t, \tag{E.9}$$

where $t|t-1$ refers to the estimate at time t given the observation up to $t-1$, $\mathbf{P}_{t|t-1} \in \mathbb{R}^{(E+1)P \times (E+1)P}$ is the prediction error covariance matrix and $\mathbf{Q}_t \in \mathbb{R}^{(E+1)P \times (E+1)P}$, the noise covariance matrix.

2) Update: The KF update of the state vector can be expressed as convex optimization problem [46], [47]:

$$\begin{aligned} \underset{\hat{\mathbf{f}}^{\mathcal{S}}_{t|t}, \mathbf{w}_t}{\text{minimize}} \quad & \mathbf{w}_t^\top \mathbf{R}_t{}^{-1}\mathbf{w}_t + (\hat{\mathbf{f}}^{\mathcal{S}}_{t|t} - \hat{\mathbf{f}}^{\mathcal{S}}_{t|t-1})^\top \mathbf{P}^{-1}_{t|t-1}(\hat{\mathbf{f}}^{\mathcal{S}}_{t|t} - \hat{\mathbf{f}}^{\mathcal{S}}_{t|t-1}), \\ \text{subject to} \quad & \mathbf{y}^{\mathcal{S}}[t] = \mathbf{C}^{\mathcal{S}}[t]\hat{\mathbf{f}}^{\mathcal{S}}_{t|t} + \mathbf{w}_t. \end{aligned} \tag{E.10}$$

Solving (E.10) yields the standard KF update equation:

$$\hat{\mathbf{f}}^{\mathcal{S}}_{t|t} = \hat{\mathbf{f}}^{\mathcal{S}}_{t|t-1} + \mathbf{K}_t(\mathbf{y}^{\mathcal{S}}[t] - \mathbf{C}^{\mathcal{S}}[t]\hat{\mathbf{f}}^{\mathcal{S}}_{t|t-1}). \tag{E.11}$$

The covariance matrix can be updated as

$$\mathbf{P}_{t|t} = \mathbf{P}_{t|t-1} - \mathbf{K}_t\mathbf{C}^{\mathcal{S}}[t]\mathbf{P}_{t|t-1}. \tag{E.12}$$

where $\mathbf{K}_t = \mathbf{P}_{t|t-1}\mathbf{C}^{\mathcal{S}}[t]^\top(\mathbf{C}^{\mathcal{S}}[t]\mathbf{P}_{t|t-1}\mathbf{C}^{\mathcal{S}}[t]^\top + \mathbf{R}_t)^{-1}$ is the Kalman gain and $\mathbf{R}_t$ is the covariance matrix of the observation noise.

3) Flow-conservation update: The KF update problem (E.10), penalized with the flow conservation (E.2), can be written as

$$\begin{aligned} \underset{\hat{\mathbf{f}}^{\mathcal{S}}_{t|t}, \mathbf{w}_t}{\text{minimize}} \quad & \mathbf{w}_t^\top \mathbf{R}_t{}^{-1}\mathbf{w}_t + (\hat{\mathbf{f}}^{\mathcal{S}}_{t|t} - \hat{\mathbf{f}}^{\mathcal{S}}_{t|t-1})^\top \mathbf{P}^{-1}_{t|t-1}(\hat{\mathbf{f}}^{\mathcal{S}}_{t|t} - \hat{\mathbf{f}}^{\mathcal{S}}_{t|t-1}) \\ & \qquad + \mu \hat{\mathbf{f}}^{\mathcal{S}}_{t|t}[t]^\top \mathbf{L} \hat{\mathbf{f}}^{\mathcal{S}}_{t|t}[t], \\ \text{subject to} \quad & \mathbf{y}^{\mathcal{S}}[t] = \mathbf{C}^{\mathcal{S}}[t]\hat{\mathbf{f}}^{\mathcal{S}}_{t|t} + \mathbf{w}_t, \end{aligned} \tag{E.13}$$

where

$$\mathbf{L} = \begin{bmatrix} \tilde{\mathbf{L}}_1^l & \mathbf{0}_{(E+1)\times(P-1)(E+1)} \\ \mathbf{0}_{(P-1)(E+1)\times(E+1)} & \mathbf{0}_{(P-1)(E+1)\times(P-1)(E+1)} \end{bmatrix},$$

with $\tilde{\mathbf{L}}_1^l = [\mathbf{L}_1^l \ \mathbf{0}_E; \mathbf{0}_E^\top \ 0] \in \mathbb{R}^{(E+1)\times(E+1)}$, the Laplacian $\mathbf{L}_1^l$ padded with zero vector $\mathbf{0}_E \in \mathbb{R}^E$ to nullify the bias component in $\mathbf{f}[t]$ and μ is a hyperparameter. We regularize flow conservation instead of imposing it as a constraint, based on the assumption that the flow conservation is not strictly satisfied in real-world networks. The optimization problem (E.13) is quadratic with a closed-form solution (see, E.7.1):

$$\begin{aligned} \hat{\mathbf{f}}^{\mathcal{S}}_{t|t} =& (\mathbf{C}^{\mathcal{S}}[t]^\top \mathbf{R}_t^{-1}\mathbf{C}^{\mathcal{S}}[t] + \mathbf{P}^{-1}_{t|t-1} + 2\mu\mathbf{L})^{-1} \times \\ & (\mathbf{C}^{\mathcal{S}}[t]^\top \mathbf{R}_t{}^{-1}\mathbf{y}^{\mathcal{S}}[t] + \mathbf{P}^{-1}_{t|t-1}\hat{\mathbf{f}}^{\mathcal{S}}_{t|t-1}), \end{aligned} \tag{E.14}$$

E.4.2 Line Graph Identification

The element-wise version of (E.15) for the n^{th} flow is

$$f_n[t] = \sum_{n'=1}^{E+1}\sum_{p=1}^{P} a^{(p)}_{n,n'}[t] f_{n'}[t-p] + u_n[t], \tag{E.15}$$

where $a^{(p)}_{n,n'}[t] \in \mathbb{R}$ represents the influence of the p-th time-lagged value of flow n' on flow n. For notational convenience, we stack the elements of $a^{(p)}_{n,n'}[t]$ in the lexicographic order of the indices p, and n' to obtain $\boldsymbol{a}_n[t] \in \mathbb{R}^{(E+1)P}$ and also stack the same elements along index p to obtain $\boldsymbol{a}_{n,n'}[t] \in \mathbb{R}^P$. Assuming flows are known, the online topology identification can be formulated as [23, 48]

$$\widehat{\boldsymbol{a}}_n[t] = \arg \min_{\boldsymbol{a}_n \in \mathbb{R}^{(E+1)P}} \ell^n_t(\boldsymbol{a}_n) + \lambda \sum_{n'=1}^{E+1} \|\boldsymbol{a}_{n,n'}\|_2, \tag{E.16}$$

where $\ell^n_t(\boldsymbol{a}_n) = \frac{1}{2}[f_n[t] - \boldsymbol{a}_n^\top \hat{\mathbf{f}}^{\mathcal{S}}[t-1]]^2$ is the instantaneous loss function for a node n and λ is a hyperparameter. The second term is a group-lasso regularizer added in line with the assumption that the real-world dependencies are sparse.

In general, proximal algorithms can solve objective functions of the form (E.16) having a differentiable loss function and a non-differentiable regularizer. Following

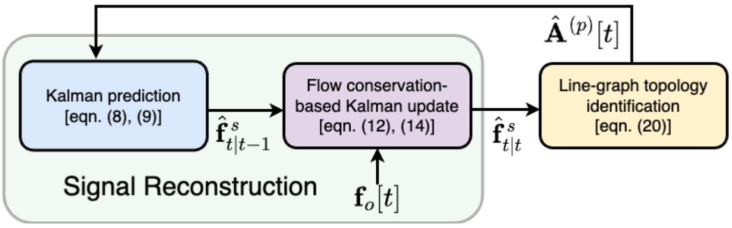
$\hat{\mathbf{A}}^{(p)}[t]$
Kalman prediction [eqn. (8), (9)]
$\hat{\mathbf{f}}^s_{t|t-1}$
Flow conservation-based Kalman update [eqn. (12), (14)]
$\hat{\mathbf{f}}^s_{t|t}$
Line-graph topology identification [eqn. (20)]
Signal Reconstruction
$\mathbf{f}_o[t]$

Figure E.2: Schematic representation of the proposed algorithm.

[48], we use online composite objective mirror descent (COMID), which is effective and comes with convergence guarantees. The online COMID update is

$$\widehat{\boldsymbol{a}}_n[t+1] = \arg \min_{\boldsymbol{a}_n \in \mathbb{R}^{(E+1)P}} J_t^{(n)}(\boldsymbol{a}_n), \tag{E.17}$$

$$\begin{aligned} \text{where } J_t^{(n)}(\boldsymbol{a}_n) &\triangleq \nabla \ell_t^n(\widehat{\boldsymbol{a}}_n[t])^\top (\boldsymbol{a}_n - \widehat{\boldsymbol{a}}_n[t]) \\ &+ \frac{1}{2\gamma_t}\|\boldsymbol{a}_n - \widehat{\boldsymbol{a}}_n[t]\|_2^2 + \lambda \sum_{n'=1}^{E+1} \|\boldsymbol{a}_{n,n'}\|_2. \end{aligned} \tag{E.18}$$

Equation (E.18) has the gradient of the loss $\ell_t^n(\boldsymbol{a}_n)$ as the first term, and the Bregman divergence and sparsity-promoting regularizer as the second and the third terms, respectively. Bregman divergence makes the algorithm more stable by constraining $\widehat{\boldsymbol{a}}_n[t+1]$ to be close to $\widehat{\boldsymbol{a}}_n[t]$ and it is chosen to be $B(\boldsymbol{a}_n, \widehat{\boldsymbol{a}}_n[t]) = \frac{1}{2}\|\boldsymbol{a}_n - \widehat{\boldsymbol{a}}_n[t]\|_2^2$ so that the COMID update has a closed-form solution [40] and $\gamma_t > 0$ is the corresponding step size. The gradient in (E.18) is evaluated as

$$\mathbf{v}_n[t] \triangleq \nabla \ell_t^n(\widehat{\boldsymbol{a}}_n[t]) = \hat{\mathbf{f}}^{\mathcal{S}}[t-1]\left(\boldsymbol{a}_n^\top \hat{\mathbf{f}}^{\mathcal{S}}[t-1] - f_n[t]\right) \tag{E.19}$$

The optimization problem is separable across nodes and a closed-form solution for (E.17) is obtained via the multidimensional shrinkage-thresholding operator [41]:

$$\widehat{\boldsymbol{a}}_{n,n'}[t+1] = \left(\widehat{\boldsymbol{a}}_{n,n'}[t] - \gamma_t \mathbf{v}_{n,n'}[t]\right)\left[1 - \frac{\gamma_t \lambda}{\|\widehat{\boldsymbol{a}}_{n,n'}[t] - \gamma_t \mathbf{v}_{n,n'}[t]\|_2}\right]_+, \tag{E.20}$$

where $[x]_+ = \max\{0, x\}$. A schematic representation of the proposed algorithm is shown in Fig. F.1. The computational complexity of the algorithm is mainly contributed by (E.14), and it is of order $\mathcal{O}\big(P^3(E+1)^3\big)$.

E.5 Experimental Results

We use flow data from a real water network and a synthetic network, both generated using the EPANET software. The flow signals are the hourly sampled volume of water in m^3/h. A demand-driven model is used to generate data such that the water flows meet the time-varying water demands at the nodes. We compare the results with the state-of-the-art algorithms Graph-based Semi-supervised learning for Edge Flows *(FlowSSL)* [20] and *Joint Signal and Topology Identification via Recursive Sparse Online learning (JSTIRSO)* [6]. FlowSSL exploits the flow conservation of the flows, whereas JSTIRSO uses a causal graph structure to impute the missing data. We compare the algorithms via the normalized mean squared error (NMSE):

$$\text{NMSE}_n(T) = \frac{\sum_{t=1}^T (f_n(t) - \hat{f}_n(t))^2}{\sum_{t=1}^T f_n(t)^2}. \tag{E.21}$$

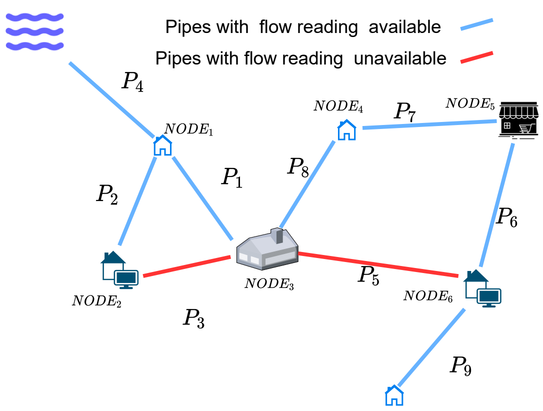

Figure E.3: Physical graph.

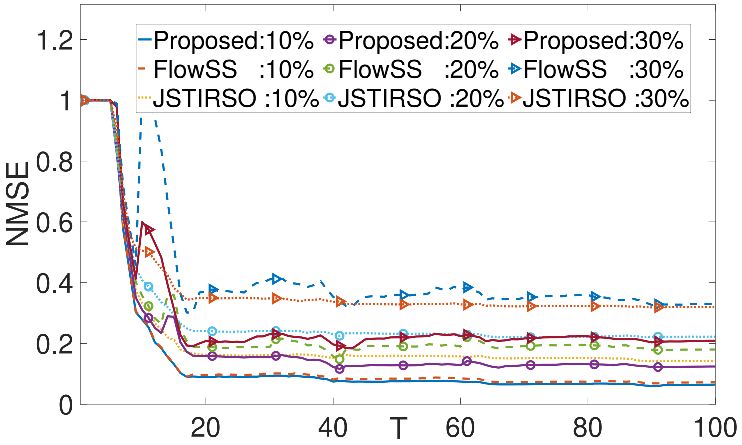

Figure E.4: Time varying random missing-flow patterns.

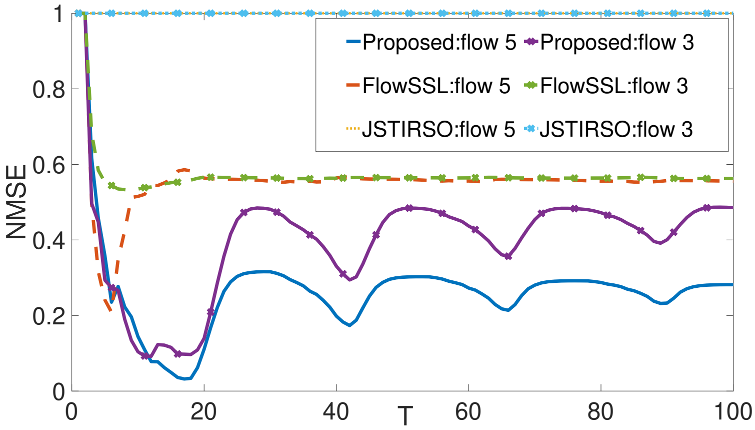

Figure E.5: Permanently unobserved flows.

Figure E.6: Synthetic Water Network Topology.

A total of 125 data samples are generated, and the initial 25 samples are used to tune the hyperparameters of all the algorithms to achieve the lowest NMSE averaged across all edges via grid search. The line graph is initialized with random values drawn from $\mathcal{N}(0, 1)$. The NMSEs are averaged over 50 runs of experiments.

E.5.1 Synthetic Water Network

A water distribution model, shown in E.3, is simulated, which consists of 1 reservoir, 9 pipes, and 8 nodes. Below, we examine two types of missing data patterns with the hyperparameter setting $(\mu, \lambda) = (0.5, 0.1)$.

E.5.1.1 Random variation in missing-flows

We assume that $10\%, 20\%$, and 30% of randomly chosen flows are missing at each time instant. NMSEs are plotted in Appendix E.4.2, which shows that the proposed method is better than the competitors because, unlike them, it takes full advantage of the flow conservation and causal dependencies. Going beyond 30% of missing data results in very high NMSEs by all algorithms, and is not included in Appendix E.4.2 to maintain the legibility.

E.5.1.2 Permanently unobserved flows

We consider flow-3 and flow-5 are permanently missing. The NMSEs for both the missing flows are shown in E.5. The proposed method provides better imputation performance compared to FlowSSL [20], whereas JSTIRSO [6] fails to reconstructs the missing signal since it does not exploit the flow conservation.

E.5.2 Cherry Hills Water Networks

Cherry Hills is a real water network consisting of 40 pipes and 36 nodes [110]. We assume a reference flow direction as in Fig. E.7, and the hyperparamters are tuned to $(\mu, \lambda) = (50, 0.04)$. We examine four different scenarios in which $20\%, 30\%, 40\%$, and 50% of the flows are randomly missing at each time stamp. The average NMSEs computed from the estimates of random missing flows are plotted in Fig. E.11, where the proposed method outperforms the other two algorithms, especially with a significant margin for the 50% missing case. NMSEs of all algorithms is very high when more than 50% of flows are missing. The experiment is repeated with $15\%, 20\%$, and 25% of permanently missing flows, and the results are plotted in Fig. E.10, where the proposed algorithm outperforms the competitors in all the cases.

One instance of the learned line graph (T=100, p=3) is shown in Fig. E.8. We wish to note that the line graph is an abstract graph induced by the various physics-based equations describing the space-temporal variation of the flows. Although one could attempt to analyse the line graph using the underlying differential equations governing the space-time system, this is a daunting complex process, which is beyond the scope of this study. However, a good prediction implies necessarily that the data-driven line graph is close to the unknown real graph. To demonstrate the importance of the learned line graph, we repeat the Kalman prediction using a random line graph without considering any relation to the data. NMSEs obtained for permanently missing flows at t=100, using random and learned line graphs, are 1.08 and 0.06, respectively. Similar results were obtained for all the other experiments highlighting the role of the learned line graph.

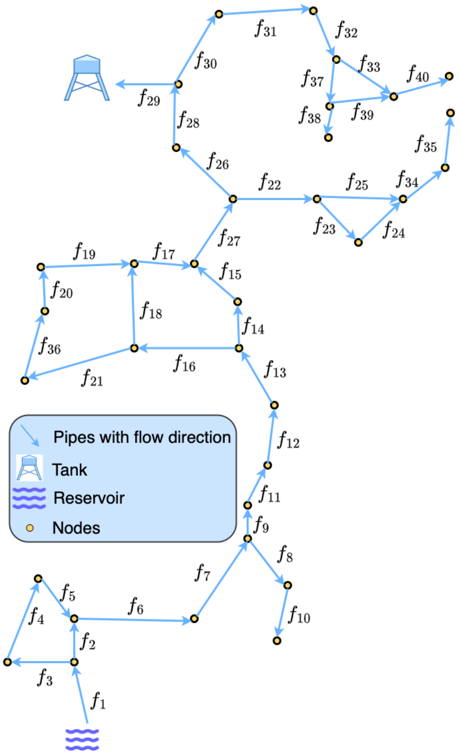

Figure E.7: Cherry Hills Flows.

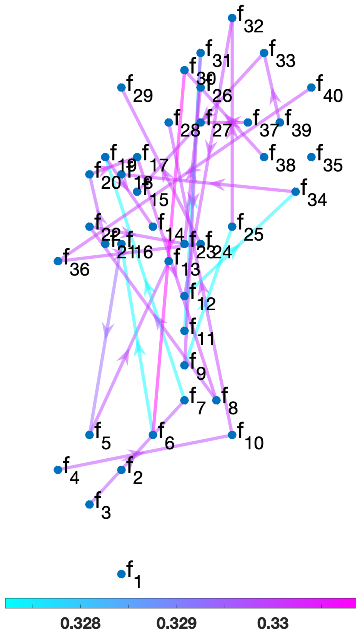

Figure E.8: Estimated Line Graph.
Figure E.9: Cherry Hills Water Network.

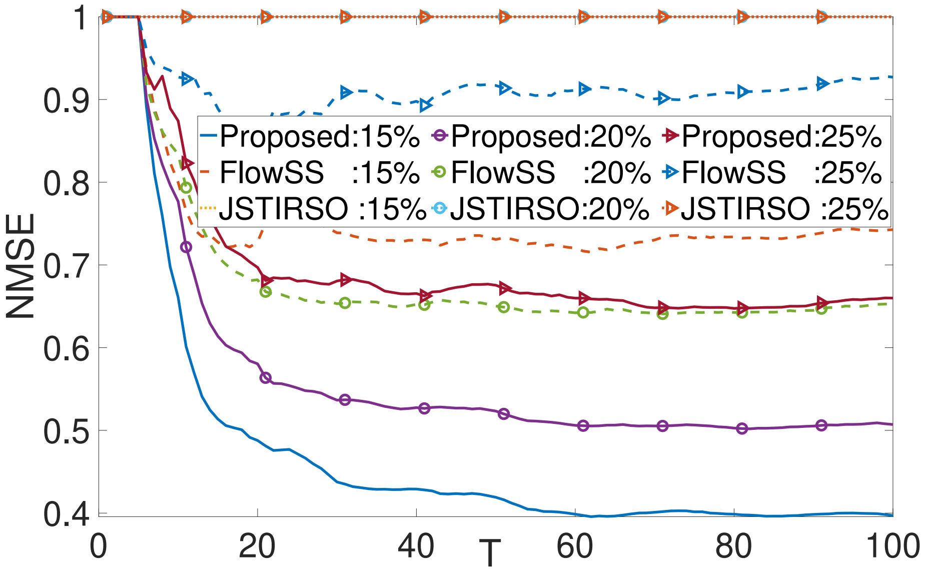

Figure E.10: Permanently Missing Flows.

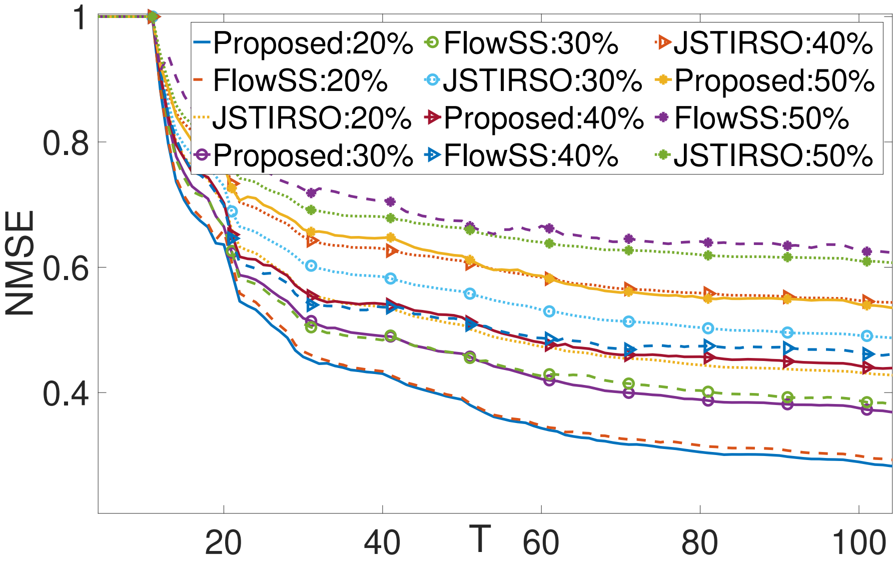

Figure E.11: Randomly Missing Flows.

Figure E.12: Cherry Hills Water Network:NMSE.

E.6 Conclusion

We proposed a novel missing data imputation scheme for flow-based networks. The proposed algorithm comprises a simplicial-complex-based Kalman filter and a group-lasso-based optimization strategy to take advantage of the flow conservation and causal dependency of real-world networks. This study paves the way for exploring higher order connectivity in real-life networks using simplicial complexes.

E.7 Supplementary Material

E.7.1 Derivation of Flow-Conservation-based Kalman Filter

The optimization problem (E.13) is a convex quadratic optimization problem that yields flow-conservation-based Kalman updates. We adopt a similar strategy as followed in [47] to obtain a closed-form solution. We first reformulate the problem (E.13) by substituting the constraint $\mathbf{w}_t = \mathbf{y}^{\mathcal{S}}[t] - \mathbf{C}^{\mathcal{S}}[t]\hat{\mathbf{f}}^{\mathcal{S}}_{t|t}$ in the objective function:

$$\begin{aligned}\underset{\hat{\mathbf{f}}^{\mathcal{S}}_{t|t}}{\text{minimize}} \quad & (\mathbf{y}^{\mathcal{S}}[t] - \mathbf{C}^{\mathcal{S}}[t]\hat{\mathbf{f}}^{\mathcal{S}}_{t|t})^{\top}\mathbf{R}_t{}^{-1}(\mathbf{y}^{\mathcal{S}}[t] - \mathbf{C}^{\mathcal{S}}[t]\hat{\mathbf{f}}^{\mathcal{S}}_{t|t}) \\ & +(\hat{\mathbf{f}}^{\mathcal{S}}_{t|t} - \hat{\mathbf{f}}^{\mathcal{S}}_{t|t-1})\mathbf{P}^{-1}_{t|t-1}(\hat{\mathbf{f}}^{\mathcal{S}}_{t|t} - \hat{\mathbf{f}}^{\mathcal{S}}_{t|t-1})^{\top} + \mu(\hat{\mathbf{f}}^{\mathcal{S}}_{t|t})^{\top}\mathbf{L}\hat{\mathbf{f}}^{\mathcal{S}}_{t|t}, \qquad \text{(E.22)}\end{aligned}$$

where

$$\underbrace{\mathbf{L}}_{P(E+1)\times P(E+1)} \triangleq \left[\begin{array}{c|c} \underbrace{\tilde{\mathbf{L}}^{l}_{1}}_{(E+1)\times(E+1)} & \underbrace{\mathbf{0}}_{(E+1)\times(P-1)(E+1)} \\ \hline \underbrace{\mathbf{0}}_{(P-1)(E+1)\times(E+1)} & \underbrace{\mathbf{0}}_{(P-1)(E+1)\times(P-1)(E+1)} \end{array}\right].$$

Next, we differentiate the objective function with respect to $\hat{\mathbf{f}}^{\mathcal{S}}_{t|t}$ and equate to 0 to find the optimum $\hat{\mathbf{f}}^{\mathcal{S}}_{t|t}$:

$$\begin{aligned}&-2\mathbf{C}^{\mathcal{S}}[t]^{\top}\mathbf{R}_t^{-1}(\mathbf{y}^{\mathcal{S}}[t] - \mathbf{C}^{\mathcal{S}}[t]\hat{\mathbf{f}}^{\mathcal{S}}_{t|t}) \\ &\qquad + 2\mathbf{P}^{-1}_{t|t-1}(\hat{\mathbf{f}}^{\mathcal{S}}_{t|t} - \hat{\mathbf{f}}^{\mathcal{S}}_{t|t-1}) + 2\mu\mathbf{L}\mathbf{f}[t] = 0 \qquad \text{(E.23)} \\ &\implies \hat{\mathbf{f}}^{\mathcal{S}}_{t|t} = (\mathbf{C}^{\mathcal{S}}[t]^{\top}\mathbf{R}^{-1}\mathbf{C}^{\mathcal{S}}[t] + \mathbf{P}^{-1}_{t|t-1} + 2\mu\mathbf{L})^{-1}\times \\ &\qquad (\mathbf{C}^{\mathcal{S}\top}\mathbf{R}^{-1}\mathbf{Y}^{\mathcal{S}}[t] + \mathbf{P}^{-1}_{t|t-1}\hat{\mathbf{f}}^{\mathcal{S}}_{t|t-1}), \qquad \text{(E.24)}\end{aligned}$$

which is the required flow-conservation-based Kalman filter solution.