Missing Value Imputation for Multi-attribute Sensor Data Streams via Message Propagation (Appendix)

A APPENDIX

Due to space limit, we include minute details and less significant experimental results and analyses in this appendix.

A.1 The results of MAE on ICU and Airquality.

We provide MAE results of the ICU and Airquality datasets in Figure 1. Overall, the MAE results behave similarly to the MRE results reported in the paper. Referring to Figure 1 (a) and Figure 1 (b), MPIN consistently outperforms the other methods across varying ratios of streams. Also, the MAE of the neural network-based methods goes up when the ratio of streams is small (e.g., 1%). This is due to the lack of training data to impute missing values effectively. In contrast, the effect of the small ratio of streams to FP depends on the dataset. On ICU, the MAE of FP goes up, whereas on Airquality, the MAE goes down. We attribute this to the difference in the sparsity: the sparsity of ICU is much higher than that of Airquality, so FP is more easily affected by the small ratio of concurrent streams on ICU.

Referring to Figure 1 (c) and Figure 1 (d), the MAE of all neural network-based methods exhibits a decreasing MAE as the time-window length increases. Usually, a longer window means that more data instances occur in the time window, resulting in more training data for the imputation. Further, the decreasing tendencies of sequential neural networks such as BRITS and SAITS are more pronounced than that of MPIN, since sequential models can capture deeper dependencies from longer sequences. However, a longer time window incurs much more time cost for the sequential models, as reported in the paper. MPIN still achieves the lowest MAE across varying lengths of time windows.

A.2 Effect of Window Length on Wi-Fi Dataset.

As shown in Figures 2, 3, and 4, we provide experimental results on the Wi-Fi dataset on the effect of window length. Overall, we achieve similar conclusions from Wi-Fi as from the other datasets. First, referring to Figures 2 and 3, MPIN is always the most effective method across varying lengths of time windows. Referring to Figure 4, MPIN is much more efficient than the sequential neural network models. Next, FP is the most efficient method since it simply conducts matrix multiplication for imputation. Compared with FP, MPIN takes a little more time but is much better at imputation. Thus, MPIN is the most desirable imputer.

A.3 Effect of Methods of Graph Construction.

R1.W4

We compare two methods for building similarity graphs, namely the threshold-based and KNN-based methods using Euclidean distance as the similarity measure. Figure 5 shows that the KNN-based method is best in terms of imputation accuracy. This is because it can always ensure that each node (i.e., instance) has K edges, while the threshold-based method can not. Also, with the increase of K, the imputation accuracy first increases and then drops. The best performance is achieved when K=10, so we set K to 10 in the experiments. When K or the distance threshold are low, meaning that there are few edges for each node in the graph, the imputation accuracy is impaired because a node may not be able to capture enough information from neighbors. With the increase of K or the threshold, the imputation accuracy can be improved. However, when K or the threshold are too high, the performance degrades. This is because if a node has too many neighboring nodes, some of them may contribute noise because they are not so similar to the target node.

A.4 Effect of Similarity Function.

Considering the construction of a similarity graph, we use the Euclidean distance to calculate similarity since this yields more effective results compared with using Cosine similarity, as shown in Figure 6. In all three datasets, a lower MRE is obtained by using Euclidean distance as the similarity function.

A.5 Effect of Number of MsgProp Layer.

Referring to Figure 7, we achieve the best effectiveness on the three datasets when using 2 MsgProp layers. With only one MsgProp layer, MPIN cannot capture the correlations among data instances (i.e., nodes) well and cannot utilize the recursive imputation process. Using 3 or more layers may cause over-smoothing [1] due to the use of an internal message-passing module. A two-layer MPIN is the best option as it can exploit both the correlations among data instances in the similarity graph and avoid over-smoothing.

A.6 Effect of Internal Message Passing Module.

We vary the internal message passing module to consider typical message passing modules such as GAT [7], GCN [6], and GraphSAGE [4] to assess their effects on the imputation performance. The results are shown in Figure 8. Generally, GraphSAGE enables better performance than that of the other two modules. GraphSAGE follows the principle of sampling-and-aggregation that may be better for capturing the correlations among data instances to impute their missing values.

R3.D2

A.7 Effect of Value Threshold.

To select a proper threshold for the data update strategy, we vary the threshold from 0.3 to 0.8 and report the results of MRE in Figure 9. As we can see, a lower MRE can be achieved when the threshold is set to around 0.6. Next, the results of MRE appear quite stable across different thresholds, which gives the flexibility to select a proper value threshold.

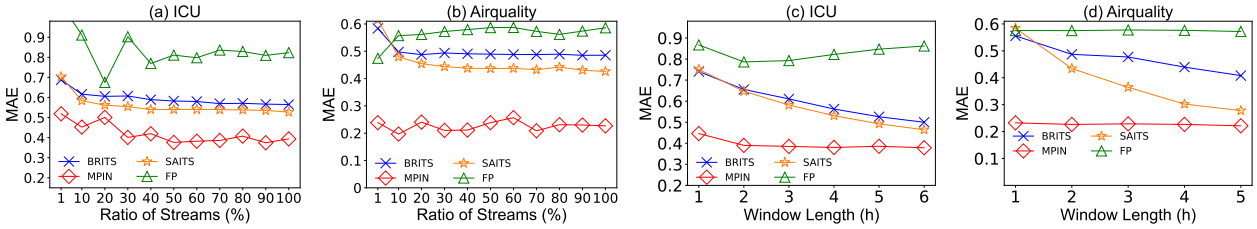

Figure 1: MAE results on ICU and Airquality.

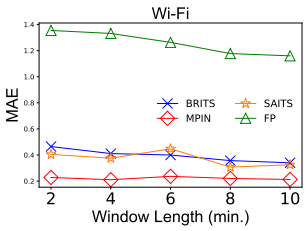

Figure 2: MAE vs. window length.

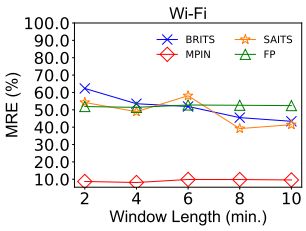

Figure 3: MRE vs. window length.

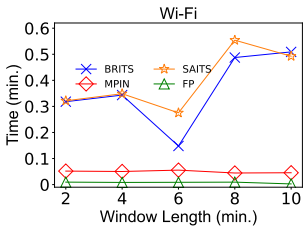

Figure 4: Time vs. window length.

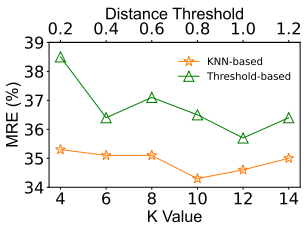

Figure 5: Effect of methods of graph construction.

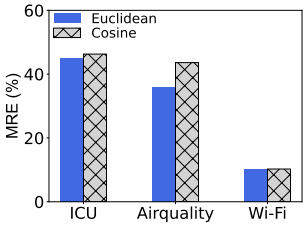

Figure 6: Effect of similarity function.

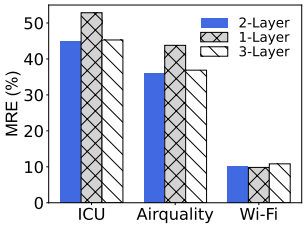

Figure 7: Effect of num of MsgProp layers.

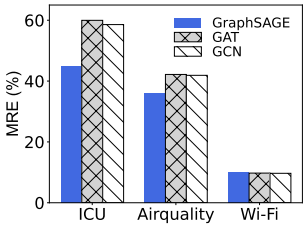

Figure 8: Effect of types of message passing.

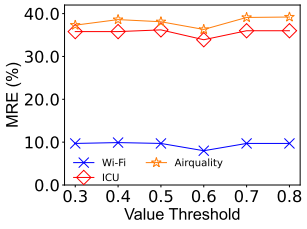

Figure 9: Effect of value threshold.

A.8 Performance on Classification Tasks.

R1.W2

We have report on a case study on the ICU dataset. Specifically, we trained a classifier on the imputed sensor dataset to predict whether or not a patient in the ICU would survive [3]. We use 3 classification metrics (i.e., AUC, F1-score, and accuracy) to achieve a comprehensive evaluation. The results in Table 1. show that the performance of MPIN in downstream classification tasks is considerably better than those of the other imputation methods. This further demonstrates the significance of MPIN in imputing missing values in sensor data streams.

Table 1: Performance on Classification Task.

Method	MEAN	KNN	MICE	MF	FP	BRITS	SAITS	MPIN
AUC	0.492	0.525	0.536	0.504	0.749	0.776	0.787	**0.813**
F1-score	0.109	0.121	0.155	0.112	0.353	0.411	0.427	**0.459**
Accuracy	0.750	0.765	0.774	0.784	0.827	0.836	0.832	**0.856**

A.9 Evaluation on Synthetic Datasets.

R2.D1

We have conducted additional experiments on two synthetic datasets from an empirical study of imputation [5]. The performance of each imputation method on the synthetic datasets is shown in Table 2. We see that MPIN always achieves much better performance than the other methods across different settings. Only in very few cases is MPIN the second-best imputer. These findings exhibit similar tendencies as those obtained on the real datasets. Thus the study offers further evidence of the effectiveness of MPIN.

A.10 Message-passing vs. MsgProp

Though MsgProp uses message passing to capture correlations among instances in the graph, it is distinctive. Essentially, MsgProp equips an additional reconstruction process and computes reconstruction errors for imputation. To assess the significance of the distinction empirically, we replace MPIN's MsgProp layer with a message-passing layer so as to discard the intermediate reconstruction process. We apply this to the same imputation tasks and show the MRE results in Figure 10. Clearly, the pure message-passing mechanism performs poorly at imputation. In contrast, MsgProp is significantly more effective, as its MsgProp minimizes the Dirichlet energy in the graph, leading to improved reconstructed features (see Section 3.2 in the paper).

A.11 Ablation Study of Combined Loss

In MPIN, we calculate the loss at the output of each MsgProp layer and use the combined loss for backpropagation. To assess the effectiveness of this design, we compare it with the designs of using only the first-layer loss or using only the second-layer loss. The results reported in Figure 11 show that MPIN with its combined loss is much more effective than the two alternatives. This is because the reconstruction is conducted on each MsgProp layer and the

Table 2: Effectiveness Comparison on Synthetic Datasets (the unit of MRE is %).

Dataset	Synthetic Dataset 1						Synthetic Dataset 2					
Rate	50%		30%		10%		50%		30%		10%	
Metrics	MAE	MRE	MAE	MRE	MAE	MRE	MAE	MRE	MAE	MRE	MAE	MRE
MEAN	0.776	99.42	0.773	99.06	0.77	98.63	0.756	99.62	0.754	99.39	0.752	99.12
KNN	0.686	87.82	0.625	80.15	0.465	59.54	0.609	80.29	0.506	66.69	0.277	36.51
MICE	0.657	83.94	0.557	71.21	0.417	53.34	0.564	74.36	0.435	57.3	0.221	29.13
MF	0.646	82.73	0.555	71.12	0.418	53.61	0.585	77.18	0.477	62.87	0.262	34.47
FP	0.591	75.78	0.465	59.55	0.395	50.62	0.461	60.71	0.31	40.89	0.213	28.04
BRITS	0.478	58.47	0.476	58.33	0.469	57.48	0.219	26.42	0.199	23.98	0.186	22.41
SAITS	0.518	63.42	0.498	61.02	0.487	59.6	0.194	**23.4**	0.191	23.06	0.191	23.04
MPIN	**0.392**	**50.21**	**0.382**	**48.97**	**0.379**	**48.51**	**0.185**	24.44	**0.156**	**20.6**	**0.142**	**18.75**

reconstructed instance of the former layer is taken as the input to the next layer. Thus, it is necessary to consider and handle the error of each layer; otherwise, the model will suffer from accumulated errors.

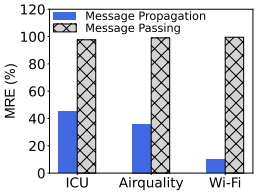

Figure 10: Message mechanism.

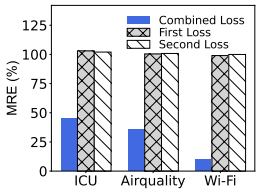

Figure 11: Loss vs. MRE.

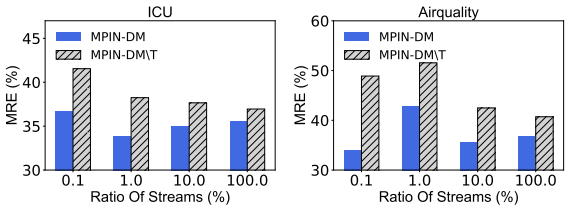

Figure 12: Effect of transfer mechanism.

A.12 Effect of Transfer Mechanism of Model Update

In the design of the model update mechanism (see Figure 6 in the paper), we adopt a selective approach to parameter reuse, rather than blindly reusing all parameters from the best-ever state of MPIN. Specifically, we reuse the parameters of the core part, while we retrain the reconstruction part from scratch. This approach draws inspiration from transfer learning. In our case, the model update process involves transferring the knowledge acquired from one graph to another. The experimental results, shown in Figure 12, validate the effectiveness of this approach. In comparison, the MPIN-DM\T method copies all parameters directly from the best-ever state, which is much less effective than our designed model update mechanism. This difference may occur because reusing all parameters can lead to overfitting, while selective parameter reuse based on transfer helps mitigate this issue.

A.13 Differences between Label Propagation and MPIN

The differences between MPIN and label propagation (i.e., LP) [8] are substantial. MPIN and LP differ not only in how they function but also in the targeted tasks. First, most of the differences between FP and MPIN also apply to LP and MPIN, since LP is similar to FP in terms of function. Second, LP and MPIN target different kinds of tasks. As the name implies, label propagation is used for semi-supervised classification tasks on complete data. In contrast, MPIN aims to impute missing attributes in data instances in sensor data streams, which can then be used for classification tasks in downstream applications. Third, LP requires labels to conduct classification tasks, while MPIN requires no labels to perform the process of imputation.

R1.W1

A.14 Proof of Lemmas

Here, we provide the proofs of Lemmas 3 and 4.

Lemma 3. *$\forall \mathbf{x}_i \in V$, we have $0 \leq \varphi(\mathbf{x}_i) \leq \mathtt{D} - 1$.*

Proof. Referring to Equation 9 in the paper, the importance score $\varphi(\mathbf{x}_i)$ reaches its maximum value when the first term, i.e., $OR(\mathbf{x}_i)$, reaches its maximum value and the second term, i.e., $\frac{1}{(|V|-1)} \sum_{\mathbf{x}_k \in V \setminus \mathbf{x}_i} OOR(\mathbf{x}_i, \mathbf{x}_k)$, reaches its minimum value; $\mathbf{x}_i$ reaches its lowest value when the condition is the opposite.

Consider the highest $\varphi(\mathbf{x}_i)$ score. The maximum observation ratio score $OR(\mathbf{x}_i)$ equals $\mathtt{D}$ when $\mathbf{x}_i[d] = 1, \forall 0 \leq d < \mathtt{D}$. As at least one dimension was observed in any of the other instances $\mathbf{x}_k \in V \setminus \mathbf{x}_i$, the minimum value of the second term cannot be lower than 1. Therefore, the highest $\varphi(\mathbf{x}_i)$ is achieved as $\mathtt{D} - 1$.

Consider the lowest $\varphi(\mathbf{x}_i)$ score. The minimum observation ratio score $OR(\mathbf{x}_i)$ equals 1 in the case where only one dimension was observed in $\mathbf{x}_i$. In this case, $\forall \mathbf{x}_k \in V \setminus \mathbf{x}_i$ we have $OOR(\mathbf{x}_i, \mathbf{x}_k) = \mathbf{m}_i^\top \mathbf{m}_k \leq 1$ according to Equation 11. Therefore, the lowest $\varphi(\mathbf{x}_i)$ must be no less than $1 - 1 = 0$.

To sum up, we have $0 \leq \varphi(\mathbf{x}_i) \leq \mathtt{D} - 1$. □

Lemma 4. *Given the corresponding gram mask matrix $\mathbf{M}^{GM}$, the importance scores of data instances in the set V can be computed*

jointly by

$$\boldsymbol{\varphi} = \frac{(|V| * \text{diag}^{-1}(\mathbf{M}^{GM}) - \mathbf{M}^{GM} \cdot \mathbf{1}^{|V|\times 1})}{|V| - 1}, \tag{1}$$

where $\boldsymbol{\varphi} \in \mathbb{R}^{|V|\times 1}$ and $\boldsymbol{\varphi}[i]$ captures the importance score of the i-th data instance $\mathbf{x}_i$ in data chunk $\mathcal{X}_a$, $\text{diag}^{-1}(\cdot)$ gets the diagonal vector from the input matrix, and $*$ denotes the element-wise scalar product.

Proof. Following the literature [2], we have $\mathbf{M}^{\text{GM}}[i,i] = \mathbf{m}_i^\top \mathbf{m}_i$ and $\mathbf{M}^{\text{GM}}[i,k] = \mathbf{m}_i^\top \mathbf{m}_k$. With $\boldsymbol{\varphi}'$ being the numerator of Equation 1, we have

$$\begin{aligned}
\boldsymbol{\varphi}' &= |V| * \text{diag}^{-1}(\mathbf{M}^{\text{GM}}) - \mathbf{M}^{\text{GM}} \cdot \mathbf{1}^{|V|\times 1} \\
&= [|V|\mathbf{m}_i^\top \mathbf{m}_i - \sum\nolimits_{k=0}^{|V|-1} \mathbf{m}_i^\top \mathbf{m}_k : i = 0, \ldots, |V| - 1] \\
&= [(|V| - 1)\mathbf{m}_i^\top \mathbf{m}_i - \sum\nolimits_{k=0 \wedge k \neq i}^{|V|-1} \mathbf{m}_i^\top \mathbf{m}_k : i = 0, \ldots, |V| - 1] \\
&= [(|V| - 1)OR(\mathbf{x}_i) - \sum\nolimits_{\mathbf{x}_k \in V \setminus \mathbf{x}_i} OOR(\mathbf{x}_i, \mathbf{x}_k) : i = 0, \ldots, |V| - 1].
\end{aligned}$$

Dividing $\boldsymbol{\varphi}'$ by $(|V| - 1)$, we get $\boldsymbol{\varphi} = [\varphi(\mathbf{x}_i) : i = 0, \ldots, |V| - 1]$. Therefore, $\boldsymbol{\varphi}[i]$ equals the importance score of $\mathbf{x}_i$ (see Equation 9). The lemma is thus proved. □

REFERENCES

[1] Deli Chen, Yankai Lin, Wei Li, Peng Li, Jie Zhou, and Xu Sun. 2020. Measuring and relieving the over-smoothing problem for graph neural networks from the topological view. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 34. 3438–3445.

[2] Petros Drineas, Michael W Mahoney, and Nello Cristianini. 2005. On the Nyström Method for Approximating a Gram Matrix for Improved Kernel-Based Learning. *journal of machine learning research* 6, 12 (2005).

[3] Wenjie Du, David Côté, and Yan Liu. 2023. SAITS: Self-attention-based imputation for time series. *Expert Syst. Appl.* 219 (2023), 119619.

[4] William L. Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive Representation Learning on Large Graphs. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*. 1024–1034.

[5] Mourad Khayati, Alberto Lerner, Zakhar Tymchenko, and Philippe Cudré-Mauroux. 2020. Mind the Gap: An Experimental Evaluation of Imputation of Missing Values Techniques in Time Series. *Proc. VLDB Endow.* 13, 5 (2020), 768–782.

[6] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

[7] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

[8] Xiaojin Zhu and Zoubin Ghahramani. 2002. *Learning from Labeled and Unlabeled Data with Label Propagation.* Technical Report CMU-CALD-02-107. School of Computer Science, Carnegie Mellon University. 19 pages.