Supplementary Materials to: Missing Value Imputation for Multi-attribute Sensor Data Streams via Message Propagation

1 SUPPLEMENTARY EXPERIMENTS

The supplementary experiments consist of two parts: Section 1.1 show additional experimental results either in Wi-Fi dataset or with MAE metric and Section 1.2 evaluate the effect of key hyperparameters on the effectiveness of imputation.

1.1 Evaluation of Additional Experiments

MAE Results on ICU and Airquality Datasets. We complement MAE results of ICU and Airquality datasets in Figure 1.Overall, the results of MAE exhibit similar tendencies as that of MRE in the paper. Referring to (a) and (b), MPIN consistently outperforms the other methods in varying ratios of streams. Besides, the MAE of neural network-based methods would go 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. In ICU dataset, the MAE of FP goes up whereas in Airquality dataset, the MAE goes down. We attribute this to the difference in sparsity: the sparsity of ICU dataset is much higher than that of Airquality and thus FP is easier to be affected by the small ratio of concurrent streams in ICU.

Referring to (c) and (d), the MAE of all neural network-based methods have a decreasing tendencies of MAE with the increasing length of a time window. Usually a longer time window means more data instances involved in the time window and thus can contribute more training data for the imputation. Besides, the decreasing tendencies of sequential neural networks such as BRITS and SAITS is sharper than that of MPIN, since sequential models can capture deeper dependencies from longer sequences. However, a longer time window will incur much more time cost for those sequential models, as reported in the paper. Besides, MPIN still achieves the lowest MAE in varying lengths of time windows.

Effect of Window Length on Wi-Fi Dataset. As shown in Figures 2, 3 and 4, we complement the experimental results on Wi-Fi dataset regarding the effect of window length. Overall, we can achieve similar conclusions from Wi-Fi dataset as that from the other datasets. First, referring to Figures 2 and 3, MPIN is always the most effective method in varying lengths of time windows. Referring to Figure 4, MPIN is much more efficient than sequential neural network models. Besides, FP is the most efficient method since it simply conducts matrix multiplication for imputation. Compared with FP, MPIN consumes a little more time but achieves much more effectiveness in imputation. Thus, MPIN is the most desirable imputer.

Summary. From the above results, we can find that the results of MAE demonstrate similar tendencies as that of MRE reported in the paper. Also, the points derived from experimental results on Wi-Fi dataset are generally similar to that derived from ICU and Airquality datasets. This further justifies the generalizability of our proposals. Next, we will evaluate the effect of key hyperparameters on the effectiveness of imputation.

1.2 Evaluations of Key Hyperparameters

Effect of Number of MsgProp Layer. Referring to Figure 5, we can achieve the best effectiveness result in three datasets when the number of MsgProp layer equals to 2. With only one MsgProp layer, MPIN is hard to capture the correlations among data instances (i.e., nodes) well and cannot utilize the recursive imputation process. Besides, three-layer or even deeper MPIN may encounter the over-smoothing issue [1] due to the use of an internal message-passing module. A two-layer MPIN would be the best option as it can both exploit the correlations among data instances in the similarity graph and avoid the over-smoothing issue caused by deeper structure.

Effect of Type of Internal Message Passing Module. We vary the type of internal message passing module from typical message passing modules such as GAT [5], GCN [4], and GraphSAGE [3] to test their effects on the performance of imputation. The results are shown in Figure 6. Generally, the use of GraphSAGE can achieve a better effect than that of the other two modules. GraphSAGE following the principle of sampling-and-aggregation may be better to capture the correlations among data instances to impute their missing values.

Effect of Similarity Function. Regarding the construction of a similarity graph, we use the Euclidean distance function to calculate the similarity since it can lead to more effective results compared with Cosine similarity function, as shown in Figure 7. In all three datasets, a lower MRE can be obtained by using Euclidean distance as the similarity function.

Effect of K Selection of KNN. To select a proper K value for KNN to construct the similarity graph, we vary K from 5 to 20 to see the results of MRE under different K values. The results are reported in Figure 8. We can find that varying K values make no obvious difference in the effectiveness of imputation. This is probably due to the fact that the high sparsity of datasets themselves leaves very limited space to benefit from interacting with more or less neighbors. Despite that, MPIN can exploit the correlations among data instances in the similarity graph to enhance the effectiveness of imputation. We finally select K=6 as it performs better than the other K values in Airquality dataset.

Effect of Value Threshold. To select a proper value threshold for the data updating strategy, we vary the threshold values from 0.3 to 0.8 and the results of MRE are shown in Figure 9. As we can see, a lower MRE can be achieved when the threshold takes a value around 0.6. Besides, the results of MRE appear quite stable in varying thresholds and this gives the flexibility of deciding a proper value threshold.

Summary. We have evaluated the effect of various hyperparameters on the effectiveness of imputation. Also, we give some analysis based on observations from the experimental results. Finally, the

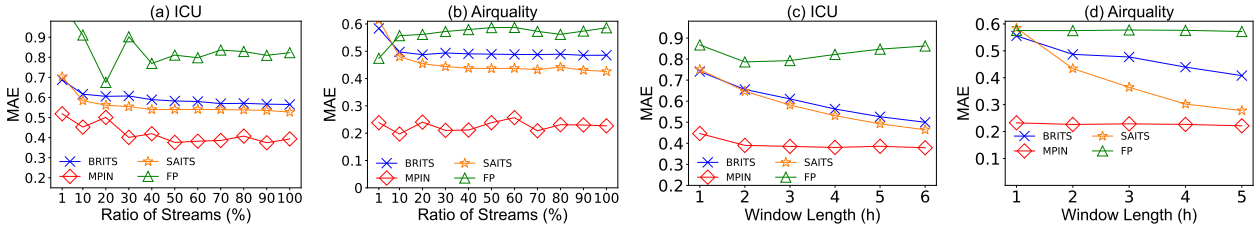

Figure 1: MAE results on ICU and Airquality.

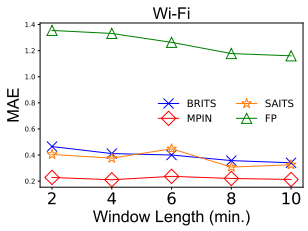

Figure 2: MAE vs. window length.

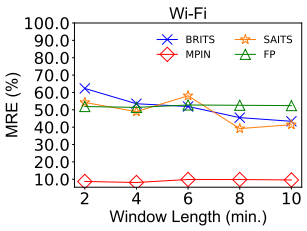

Figure 3: MRE vs. window length.

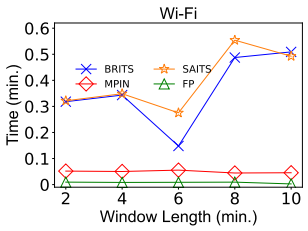

Figure 4: Time vs. window length.

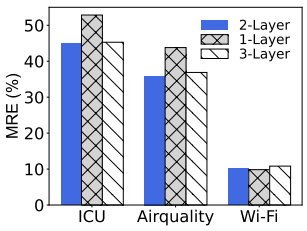

Figure 5: Effect of num of MsgProp layers.

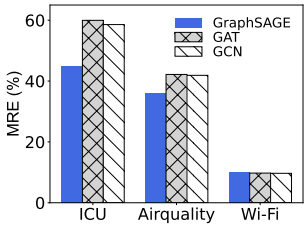

Figure 6: Effect of type of message passing.

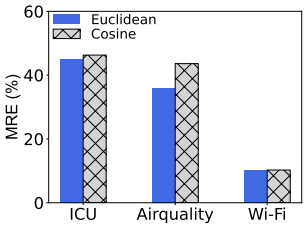

Figure 7: Effect of similarity function.

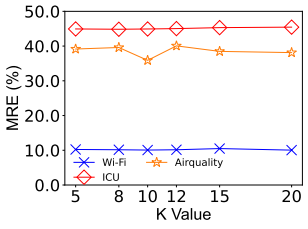

Figure 8: Effect of K value.

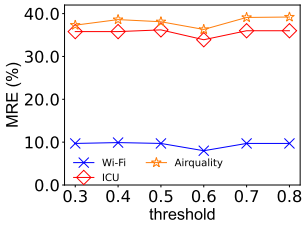

Figure 9: Effect of value threshold.

optimal value of those hyperparameters are decided based on these results.

2 PROOF OF LEMMAS

This section provides proof of Lemma 3 and Lemma 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 highest when the first term $OR(\mathbf{x}_i)$ reaches its maximal and the second term $\frac{1}{(|V|-1)} \sum_{\mathbf{x}_k \in V \setminus \mathbf{x}_i} OOR(\mathbf{x}_i, \mathbf{x}_k)$ reaches its minimal; $\mathbf{x}_i$ reaches its lowest when the condition is opposite.

Consider the highest $\varphi(\mathbf{x}_i)$ score. The maximal 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 minimal 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 minimal observation ratio score $OR(\mathbf{x}_i)$ equals 1 in the case that 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*

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

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 the data chunk $\mathcal{X}_a$, $\mathrm{diag}^{-1}(\cdot)$ gets the diagonal vector from the input matrix, and $*$ refers to the element-wise scalar product.

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

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

Dividing $\boldsymbol{\varphi}'$ by $(|V| - 1)$, we have $\boldsymbol{\varphi} = [\varphi(\mathbf{x}_i) : i = 0, \ldots, |V| - 1]$. 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] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. *Advances in neural information processing systems* 30 (2017).

[4] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).

[5] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, Yoshua Bengio, et al. 2017. Graph attention networks. *stat* 1050, 20 (2017), 10–48550.