Dynamic Graph Mixer for Learning in Incomplete High-Dimensional Tensors Supplementary File

This is the supplementary file for the paper entitled "Dynamic Graph Mixer-Incorporated Latent Factorization of Tensors". The proofs, details of experimental settings, and additional tables are put into this file and cited.

I. Proofs (Section IV)

This section provides detailed proofs for the defined two theorems in Section III.F to demonstrate that several key modules of DGM enable it achieves significantly higher expressiveness than conventional LFoT models in learning representations on an HDI tensor. More inference details are presented as the following.

A. Expressiveness Proof of Feature Fusion in DGM (Proof of Theorem 1

To prove ***Theorem* 1**, a lemma and a corollary are additionally shown for clarification. First, we consider the expressiveness of a vanilla LFoT model that does not account for the non-Euclidean structural characteristics and dimension-coupled tensor attributes, its learned representations are defined as $(\boldsymbol{p}_i, \boldsymbol{h}_j, \boldsymbol{o}_t) = f_\Omega(C_{i,j,t})$. To capture the implicit high-order connectivity including the intra- and inter-dimensional structural characteristics, the TNN module is designed for DGM. Hence, apart from message set corresponding to node i, j and time slot t, their neighborhood information are also captured and the representations are learned by $(\boldsymbol{p}_i, \boldsymbol{h}_j, \boldsymbol{o}_t) = f_\Omega(C_{i,j,t} \oplus C_{N(i,j,t)})$. The joint messages make DGM achieve higher expressiveness, which is demonstrated in ***Lemma* 1**.

***Lemma* 1.** *Suppose $f_\Omega(C_{i,j,t})$ is the function for representation learning that fails to capture implicit structural characteristics, and $f_\Omega(C_{i,j,t} \oplus C_{N(i,j,t)})$ represents the function that successfully captures these features, the objective functions corresponding to them are optimized as:*

$$\min_{\Omega} \mathbb{E}_{y_{i,j,t} \in Y_t} \left[L(y_{i,j,t}, f_\Omega \left(C_{i,j,t} \right) \right], \tag{S1a}$$

$$\min_{\Omega} \mathbb{E}_{y_{i,j,t} \in Y_t} \left[L(y_{i,j,t}, f_\Omega \left(C_{i,j,t} \oplus C_{N(i,j,t)} \right) \right]. \tag{S1b}$$

For any $y_{i,j,t} \in Y_t$, the learning function in Eq. (S1) achieves higher mutual information than that in Eq. (S1b):

$$I \left(y_{i,j,t}; f_\Omega \left(C_{i,j,t} \oplus C_{N(i,j,t)} \right) \right) \geq I \left(y_{i,j,t}; f_\Omega \left(C_{i,j,t} \right) \right). \tag{S2}$$

Proof. Several important properties regarding entropy and mutual information are required to prove ***Lemma* 1**. First, for any two variables (e.g., x and y), the joint entry between x and y are formulated as:

$$H(x, y) = H(x) + H(y|x) = H(y) + H(x|y), \tag{S3}$$

where $H(\cdot, \cdot)$ is the function of joint entropy, $H(\cdot|\cdot)$ is the function of conditional entropy, and $H(\cdot)$ is the function of marginal entropy. Moreover, two key properties of mutual information are used:

$$I(x; y) = H(x) - H(x|y) = H(y) - H(y|x), \tag{S4}$$

$$I(x; y|z) = H(x|z) - H(x|y, z) = H(y|z) - H(y|x, z), \tag{S5}$$

in which $I(\cdot; \cdot)$ is the function of mutual information and $I(\cdot; \cdot|\cdot)$ is the function of conditional mutual information. From Eq. (8), we know that different orders of latent features are fused for final output. This indicates that leveraging the joint messages of $C_{i,j,t} \oplus C_{N(i,j,t)}$ is equivalent to directly aggregating the ducal messages formulated by $C_{i,j,t}$ and $C_{N(i,j,t)}$ for the subsequent representation learning on an HDI tensor. Hence, we rewrite the mutual information between the ground-truth label $y_{i,j,t}$ and $C_{i,j,t} \oplus C_{N(i,j,t)}$ (i.e., $I(y_{i,j,t}; f_\Omega(C_{i,j,t} \oplus C_{N(i,j,t)}))$) as $I(y_{i,j,t}; f_\Omega(C_{i,j,t}), f_\Omega(C_{N(i,j,t)}))$, where $f_\Omega(C_{i,j,t})$ and $f_\Omega(C_{N(i,j,t)})$ are generated from a joint message distribution. According to Eq. (S4), we have:

$$I \left(y_{i,j,t}; f_\Omega \left(C_{i,j,t} \right), f_\Omega \left(C_{N(i,j,t)} \right) \right) = H \left(f_\Omega \left(C_{i,j,t} \right), f_\Omega \left(C_{N(i,j,t)} \right) \right) - H \left(f_\Omega \left(C_{i,j,t} \right), f_\Omega \left(C_{N(i,j,t)} \right) | y_{i,j,t} \right). \tag{S6}$$

Based on Eq. (S3), the equation in Eq. (S6) is expended as:

$$\begin{aligned} & I \left(y_{i,j,t}; f_\Omega \left(C_{i,j,t} \right), f_\Omega \left(C_{N(i,j,t)} \right) \right) \\ =& H \left(f_\Omega \left(C_{i,j,t} \right), f_\Omega \left(C_{N(i,j,t)} \right) \right) + H \left(y_{i,j,t} \right) - H \left(f_\Omega \left(C_{i,j,t} \right), f_\Omega \left(C_{N(i,j,t)} \right), y_{i,j,t} \right). \end{aligned} \tag{S7}$$

Further, Eq. (S7) can be derived as:

$$I\left(y_{i,j,t}; f_\Omega\left(C_{i,j,t}\right), f_\Omega\left(C_{N(i,j,t)}\right)\right) = H\left(y_{i,j,t}\right) - H\left(y_{i,j,t}|f_\Omega\left(C_{i,j,t}\right), f_\Omega\left(C_{N(i,j,t)}\right)\right). \tag{S8}$$

An intermediate element, i.e., $H(y_{i,j,t}|C_{i,j,t})$ is introduced to Eq. (S8) to achieve the following inference:

$$\begin{aligned} &I\left(y_{i,j,t}; f_\Omega\left(C_{i,j,t}\right), f_\Omega\left(C_{N(i,j,t)}\right)\right) \\ &= H\left(y_{i,j,t}\right) - H\left(y_{i,j,t}|f_\Omega\left(C_{i,j,t}\right)\right) + H\left(y_{i,j,t}|f_\Omega\left(C_{i,j,t}\right)\right) - H\left(y_{i,j,t}|f_\Omega\left(C_{i,j,t}\right), f_\Omega\left(C_{N(i,j,t)}\right)\right). \end{aligned} \tag{S9}$$

Given Eqs. (S4) and (S5), we rewrite Eq. (S9) as:

$$I\left(y_{i,j,t}; f_\Omega\left(C_{i,j,t}\right), f_\Omega\left(C_{N(i,j,t)}\right)\right) = I\left(y_{i,j,t}; f_\Omega\left(C_{i,j,t}\right)\right) + I\left(y_{i,j,t}; f_\Omega\left(C_{N(i,j,t)}\right)|f_\Omega\left(C_{i,j,t}\right)\right). \tag{S10}$$

Note that the mutual information is nonnegative, thus the following inference holds for any learning function $f_\Omega(\cdot)$:

$$I\left(y_{i,j,t}; f_\Omega\left(C_{i,j,t}\right), f_\Omega\left(C_{N(i,j,t)}\right)\right) \geq I\left(y_{i,j,t}; f_\Omega\left(C_{i,j,t}\right)\right). \tag{S11}$$

It is evident that the high-order connectivity is widespread in HDI tensors and serves as key structural characteristics for learning effective representations. Besides, as shown in Fig. 2, numerous key implicit connections that may be overlooked by existing methods are modeled by CAN, indicating that $I(y_{i,j,t}; f_\Omega(C_{N(i,j,t)})|f_\Omega(C_{i,j,t}))$ is typically much larger than zero. To summarize, based on the aforementioned analyses, ***Lemma* 1** holds.□

Despite the effectiveness of CAN in capturing latent structural features, it cannot learn the dimension-coupled tensor attribute characteristics, which are described by highly-sparse and sophisticated weight values. To address this problem, the TNN module is proposed for DGM, enabling it to efficiently learn the latent attribute features. With this additional TNN module, DGM's expressiveness of representation learning on an HDI tensor is further enhanced. Formally, we have ***Corollary* 1** as follows:

***Corollary* 1.** *Let $g_\Theta(\cdot)$ be the learning function, $D_{i,j,t}$ be the message set of attribute features, and $f_\Omega(C_{i,j,t} \oplus C_{N(i,j,t)}) \oplus g_\Theta(D_{i,j,t})$ be the function for representation learning that simultaneously capture the structural and attribute characteristics, the corresponding objective function is optimized as:*

$$\min_{\Omega,\Theta} \mathbb{E}_{y_{i,j,t} \in Y_t}\left[L(y_{i,j,t}, f_\Omega\left(C_{i,j,t} \oplus C_{N(i,j,t)}\right) \oplus g_\Theta\left(D_{i,j,t}\right)\right]. \tag{S12}$$

For any $y_{i,j,t} \in Y_t$, the learning function in Eq. (S12) further improves the expressiveness:

$$I\left(y_{i,j,t}; f_\Omega\left(C_{i,j,t} \oplus C_{N(i,j,t)}\right) \oplus g_\Theta\left(D_{i,j,t}\right)\right) \geq I\left(y_{i,j,t}; f_\Omega\left(C_{i,j,t} \oplus C_{N(i,j,t)}\right)\right). \tag{S13}$$

Proof. In DGM, the CAN module learns the messages of non-Euclidean structure and the TNN module learns the messages of tensor attributes. The joint effect of structural and attribute messages is then combined with Eq. (13). The above procedure is equivalent to define dual messages of $f_\Omega(C_{i,j,t} \oplus C_{N(i,j,t)})$ and $g_\Theta(D_{i,j,t})$ for learning representations. Thus, similar to the analysis of $I(y_{i,j,t}; f_\Omega(C_{i,j,t} \oplus C_{N(i,j,t)})$, the mutual information between $y_{i,j,t}$ and $f_\Omega(C_{i,j,t} \oplus C_{N(i,j,t)}) \oplus g_\Theta(D_{i,j,t})$ (i.e., $I(y_{i,j,t}; f_\Omega(C_{i,j,t} \oplus C_{N(i,j,t)}) \oplus g_\Theta(D_{i,j,t}))$ can be rewritten as $I(y_{i,j,t}; f_\Omega(C_{i,j,t} \oplus C_{N(i,j,t)}), g_\Theta(D_{i,j,t})$. And based on Eqs. (S3)-(S5) we have the following reference:

$$\begin{aligned} &I\left(y_{i,j,t}; f_\Omega\left(C_{i,j,t} \oplus C_{N(i,j,t)}\right), g_\Theta\left(D_{i,j,t}\right)\right) \\ =&H\left(f_\Omega\left(C_{i,j,t} \oplus C_{N(i,j,t)}\right), g_\Theta\left(D_{i,j,t}\right)\right) - H\left(f_\Omega\left(C_{i,j,t} \oplus C_{N(i,j,t)}\right), g_\Theta\left(D_{i,j,t}\right)|y_{i,j,t}\right) \\ =&H\left(f_\Omega\left(C_{i,j,t} \oplus C_{N(i,j,t)}\right), g_\Theta\left(D_{i,j,t}\right)\right) + H\left(y_{i,j,t}\right) - H\left(f_\Omega\left(C_{i,j,t} \oplus C_{N(i,j,t)}\right), g_\Theta\left(D_{i,j,t}\right), y_{i,j,t}\right) \\ =&H\left(y_{i,j,t}\right) - H\left(y_{i,j,t}|f_\Omega\left(C_{i,j,t} \oplus C_{N(i,j,t)}\right), g_\Theta\left(D_{i,j,t}\right)\right) \\ =&H\left(y_{i,j,t}\right) - H\left(y_{i,j,t}|f_\Omega\left(C_{i,j,t} \oplus C_{N(i,j,t)}\right)\right) + H\left(y_{i,j,t}|f_\Omega\left(C_{i,j,t} \oplus C_{N(i,j,t)}\right)\right) \\ &- H\left(y_{i,j,t}|f_\Omega\left(C_{i,j,t} \oplus C_{N(i,j,t)}\right), g_\Theta\left(D_{i,j,t}\right)\right) \\ =&I\left(y_{i,j,t}; f_\Omega\left(C_{i,j,t} \oplus C_{N(i,j,t)}\right)\right) + I\left(y_{i,j,t}; g_\Theta\left(D_{i,j,t}\right)|f_\Omega\left(C_{i,j,t} \oplus C_{N(i,j,t)}\right)\right) \\ \geq&I\left(y_{i,j,t}; f_\Omega\left(C_{i,j,t} \oplus C_{N(i,j,t)}\right)\right), \end{aligned} \tag{S14}$$

where the inequality in Eq. (S14) is derived from the nonnegativity of mutual information. Overall, ***Corollary* 1** holds.□

In real scenarios, the sparsity of an HDI tensor determines each node's sensitivity to high-order connectivity information, thus capturing these features can significantly enhances a model's representation learning ability. Furthermore, the attribute characteristics of an HDI tensor serve as another important message source, which can also greatly improve a model's expressiveness. According to the above analyses, it is seen that both CAN and TNN are fundamental for DGM to learn more accurate representations. In conclusion, ***Theorem* 1** is established.□

B. Expressiveness Proof of Conjoint Attention Mechanism

Another fundamental improvement of DGM is the application of conjoint attentions to graph message passing within the CAN module. Leveraging latent features to calculate the connection strength is a common technique used in existing graph attention networks, and the learned representation for a node is denoted as $z_i^{FA} = r_\Psi(F_i)$. However, they never consider capturing structural interventions in the computation process. Particularly in our case, the graph structure becomes more complex due to its time-varying properties, which underscores the necessity of further exploring structural characteristics. Hence, CAN consider the joint effect of latent features and structural interventions to achieve the representation for node i as $z_i^{CA} = r_\Psi(F_i \oplus S_i)$. Based on the probability definition of mutual information [21], [22], we have the following inference for any $i \in N$:

$$I\left(X_i; z_i^{CA}\right) - I\left(X_i; z_i^{FA}\right) = \mathbb{E}\left[\log \frac{\mathbb{P}\left(X_i, z_i^{CA}\right)}{\mathbb{P}\left(X_i\right) \cdot \mathbb{P}\left(z_i^{CA}\right)}\right] - \mathbb{E}\left[\log \frac{\mathbb{P}\left(X_i, z_i^{FA}\right)}{\mathbb{P}\left(X_i\right) \cdot \mathbb{P}\left(z_i^{FA}\right)}\right], \tag{S15}$$

where $\mathbb{E}[\cdot]$ is the function of expected value and $\mathbb{P}(\cdot)$ is the function of probability. According to the properties of the expectation function, external variables unrelated to the probability distribution do not affect the expected value. Hence, we infer:

$$I\left(X_i; z_i^{CA}\right) - I\left(X_i; z_i^{FA}\right) = \mathbb{E}\left[\log \frac{\mathbb{P}\left(X_i, z_i^{CA}\right)}{\mathbb{P}\left(z_i^{CA}\right)} \cdot \frac{\mathbb{P}\left(z_i^{FA}\right)}{\mathbb{E}\left(X_i, z_i^{FA}\right)}\right]. \tag{S16}$$

Since Eq. (3) only involves weighted message aggregation without introducing any perturbations, the mutual information $I(X_i; r_\Psi(F_i \oplus S_i))$ can be rewritten as $I(X_i; r_\Psi(F_i), r_\Psi(S_i))$. Hence, we infer:

$$I\left(X_i; z_i^{CA}\right) - I\left(X_i; z_i^{FA}\right) = \mathbb{E}\left[\log \frac{\mathbb{P}\left(X_i, r_\Psi\left(F_i\right), r_\Psi\left(S_i\right)\right)}{\mathbb{P}\left(r_\Psi\left(F_i\right), r_\Psi\left(S_i\right)\right)} \cdot \frac{\mathbb{P}\left(r_\Psi\left(F_i\right)\right)}{\mathbb{P}\left(X_i, r_\Psi\left(F_i\right)\right)}\right], \tag{S17}$$

Based on the properties of joint probability and mutual information, we further derive Eq. (S17) as:

$$\begin{aligned}
&I\left(X_i; z_i^{CA}\right) - I\left(X_i; z_i^{FA}\right) \\
=&\mathbb{E}\left[\log \frac{\mathbb{P}\left(X_i, r_\Psi\left(F_i\right), r_\Psi\left(S_i\right)\right)}{\mathbb{P}\left(r_\Psi\left(S_i\right) | r_\Psi\left(F_i\right)\right) \cdot \mathbb{P}\left(X_i, r_\Psi\left(F_i\right)\right)}\right] \\
=&\mathbb{E}\left[\log \frac{\mathbb{P}\left(X_i, r_\Psi\left(S_i\right) | r_\Psi\left(F_i\right)\right)}{\mathbb{P}\left(r_\Psi\left(S_i\right) | r_\Psi\left(F_i\right)\right) \cdot \mathbb{P}\left(X_i | r_\Psi\left(F_i\right)\right)}\right] \\
=&I\left(X_i; r_\Psi\left(S_i\right) | r_\Psi\left(F_i\right)\right) \geq 0.
\end{aligned} \tag{S18}$$

From Eq. (S18), it is evident that z_i^{CA} achieves higher expressiveness than vanilla methods only using feature attentions. Hence, the conjoint attention mechanism further improves DGM's learning ability on an HDI tensor and ***Theorem* 2** holds.□

II. General Settings (Section V.A)

A. Details of Comparison Methods

The details of comparison models are summarized as follows:

- **EvolveGCN** [28] is an evolving GCN model, which is designed for dynamic graph representation learning. It captures the spatial patterns using GCNs and learn the temporal patterns using Recurrent Neural Networks (RNNs) to evolve parameters.
- **WD-GCN** [29] is a waterfall dynamic GCN method. It combines the modules of GCN and Long Short-Term Memory (LSTM) network to capture the spatial and temporal patterns within an HDI tensor.
- **SGP** [15] is a scalable GNN-based dynamic predictor. It achieves high efficiency and accuracy by improving the learning scalability based on the combination of GCNs randomized RNNs.
- **JMP-GCF** [30] is a joint multi-gained popularity-aware graph collaborative filter model. It learns multi-gained latent features within HDI data by combining several popularity-varying graph diffusion networks.
- **GRU-GCN** [31] is a novel combination method based on GRUs and GCNs. It first theoretically demonstrates that the method based on "time-then-graph" possesses higher expressivity than the approach on "time-and-graph".
- **APAN** [32] is an asynchronous propagation attention network. It is designed for higher inference efficiency, which can be used in the tasks of representation learning to continuous and discrete dynamic graphs.
- **TM-GCN** [33] is a dynamic GCN model based on tensor M-product framework. It follows the paradigm of Message Passing Neural Network (MPNN) on an HDI tensor to achieve effective dynamic representation learning.
- **MegaCRN** [34] is a meta-graph convolutional recurrent network-based LFoT model. It implements a meta-node bank for graph structure learning, in which the problems of spatiotemporal heterogeneity and non-stationary are addressed.
- **DDSTGCN** [35] is a dual dynamic spatiotemporal GCN. It generates a dual hypergraph according to the original dynamic graph structure in an HDI tensor, such that the characteristics of nodes and edges are simultaneously learned.

- **CTGCN** [16] is a k-core substructure learning-based temporal GCN method. It applies GCNs to constructed k-core subgraphs for learning local structural information and uses RNNs to capture dynamics.
- **hetGNN-LSTM** [36] is a combination method based on a heterogeneous GNN and an LSTM network. It constructs multiple graphs by considering different types of node interactions, thus learning more comprehensive dynamic spatial patterns.
- **MGDN** [6] is a Markov diffusion network-based LFoT model. It incorporates the idea of distance learning into its Markov diffusion process and uses a Multi-Layered Perceptron to learn the diffusion features.
- **GraphMixer** [25] is an MLP mixer-based model. It notes that previous complex methods suffer from insufficient representational capacity due to structural redundancy, and thus proposes a simple yet effective approach for dynamic graphs.
- **PGCN** [14] is a progressive GCN-based model for HDI tensor learning. It captures the temporal patterns using the dilated causal convolution and capture the trend similarities with progressive GCNs.
- **DGM** is the dynamic graph mixer-based LFoT model proposed in this paper.

B. Details of Training Settings

We consistently adopt the following settings: All trainable variables of a model are randomly initialized by using the Xavier method [26], and they are trained by using the Adam optimizer [38]. In individually-built training and validation datasets, the hyperparameters are tuned with care and the achieved best settings are applied to the testing set. For all methods, the dimensionality of latent feature space is fixed at ten, the batch size is set to 2^{15}, and the coefficients controlling L_2 regularization strength and learning rate are tuned among $\{1e-5, 1e-4, 1e-3, 5e-3, 1e-2\}$. For the GNN-based models, the graph convolutional layer number is tuned in the range of $\{1, 2, 3, 4\}$. For the proposed DGM, we also tune the layer number of TNN module in $\{1, 2, 3, 4\}$. Besides, for the other models involved in our comparison experiments, their unique structures and hyperparameters are set according to the suggestions of their authors. Considering the effects of random factors such as parameter initialization, we apply the ten-fold cross validation to report the averaged results and the corresponding standard deviations. Once the training epochs reach the upper bound, which is set 1000 in our implementations, or the accuracy stops to increase for twenty epochs, we terminate our training process of a model.

III. Supplementary Experimental Tables (Section V.B

Based on the results shown in Tables III-V, we conduct Wilcoxon signed-ranks test to determine whether there is a statistically significant difference between DGM and its peers. Tables SI and SII report the detailed test results.

TABLE SI
WILCOXON SIGNED-RANKS TEST OUTCOMES ON ACCURACY. (**For DGM, high R+ value stands for high accuracy. **With the significance level of 0.01, we hypotheses the accepted hypotheses.*)

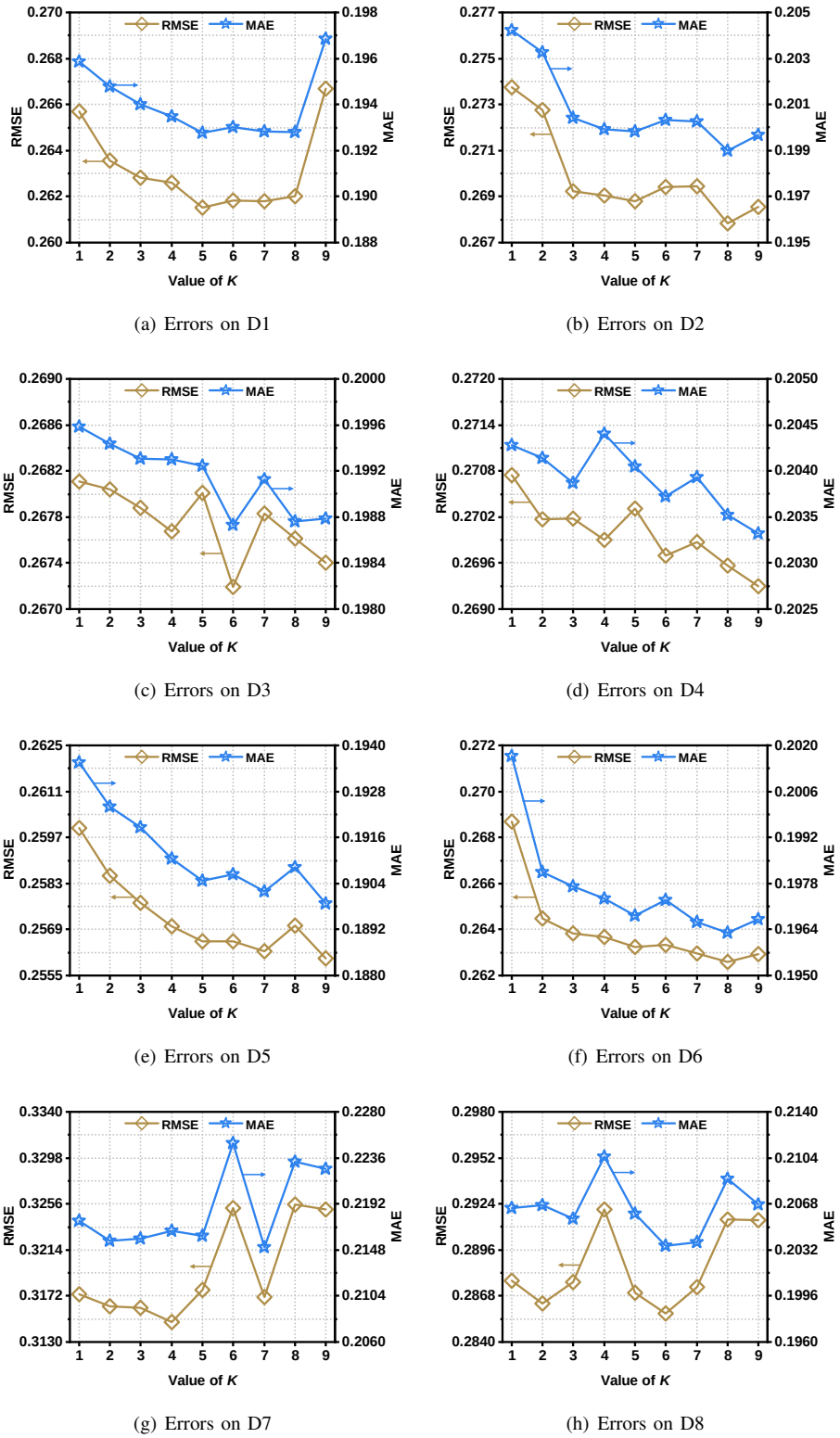
Comparision	R+*	R-	p-value**	Comparision	R+*	R-	p-value**
DGM vs EvolveGCN	136	0	**2.41E-4**	DGM vs MegaCRN	136	0	**2.41E-4**
DGM vs WD-GCN	136	0	**2.41E-4**	DGM vs DDSTGCN	136	0	**2.41E-4**
DGM vs SGP	136	0	**2.41E-4**	DGM vs CTGCN	136	0	**2.41E-4**
DGM vs JMP-GCF	136	0	**2.41E-4**	DGM vs hetGNN-LSTM	136	0	**2.41E-4**
DGM vs GRU-GCN	136	0	**2.41E-4**	DGM vs MGDN	136	0	**2.41E-4**
DGM vs APAN	136	0	**2.41E-4**	DGM vs GraphMixer	136	0	**2.41E-4**
DGM vs TM-GCN	136	0	**2.41E-4**	DGM vs PGCN	136	0	**2.41E-4**

TABLE SII
WILCOXON SIGNED-RANKS TEST OUTCOMES ON EFFICIENCY. (**For DGM, high R+ value stands for high efficiency. **With the significance level of 0.01, we hypotheses the accepted hypotheses.*)

Comparision	R+*	R-	p-value**	Comparision	R+*	R-	p-value**
DGM vs EvolveGCN	136	0	**2.41E-4**	DGM vs MegaCRN	136	0	**2.41E-4**
DGM vs WD-GCN	136	0	**2.41E-4**	DGM vs DDSTGCN	136	0	**2.41E-4**
DGM vs SGP	136	0	**2.41E-4**	DGM vs CTGCN	136	0	**2.41E-4**
DGM vs JMP-GCF	136	0	**2.41E-4**	DGM vs hetGNN-LSTM	136	0	**2.41E-4**
DGM vs GRU-GCN	136	0	**2.41E-4**	DGM vs MGDN	136	0	**2.41E-4**
DGM vs APAN	136	0	**2.41E-4**	DGM vs GraphMixer	136	0	**2.41E-4**
DGM vs TM-GCN	136	0	**2.41E-4**	DGM vs PGCN	136	0	**2.41E-4**

IV. Supplementary Figures of Experimental Results (Section V.C

- Fig. S1 (as discussed in Section V.C.i) illustrates how the errors of DGM vary with changes in K.
- Fig. S2 (as discussed in Section V.C.ii) illustrates how the errors of DGM vary with changes in L.
- Fig. S3 (as discussed in Section V.C.iii) illustrates how the errors of DGM vary with changes in R.



(a) Errors on D1

(b) Errors on D2

(c) Errors on D3

(d) Errors on D4

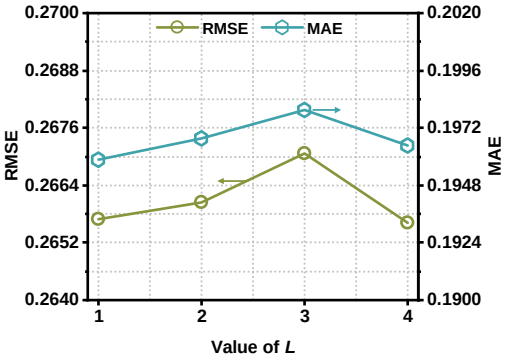
(e) Errors on D5

(f) Errors on D6

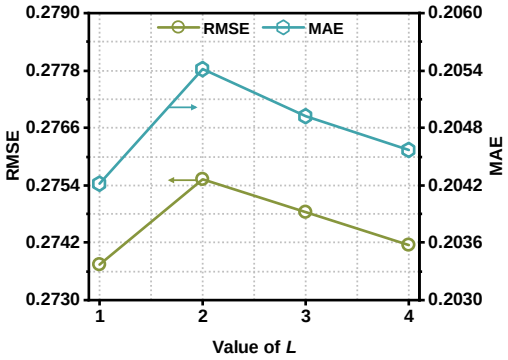
(g) Errors on D7

(h) Errors on D8

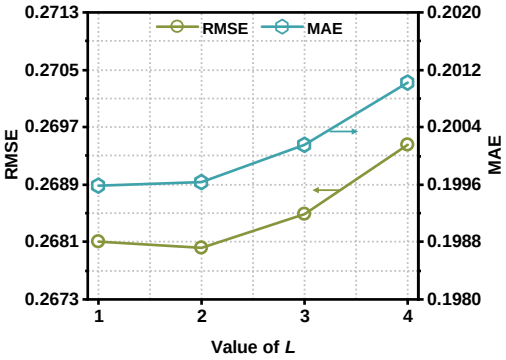
Fig. S1. Errors of DGM as K varies while fixing others on D1-8.



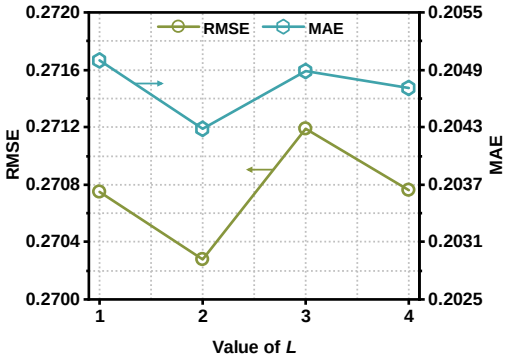
(a) Errors on D1



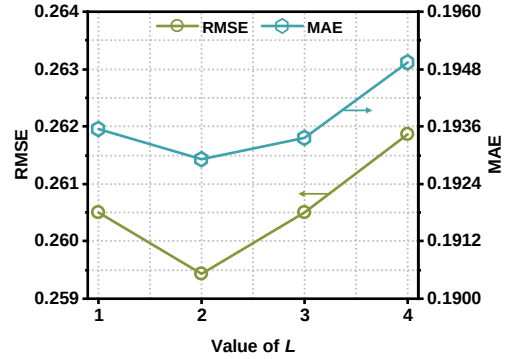
(b) Errors on D2



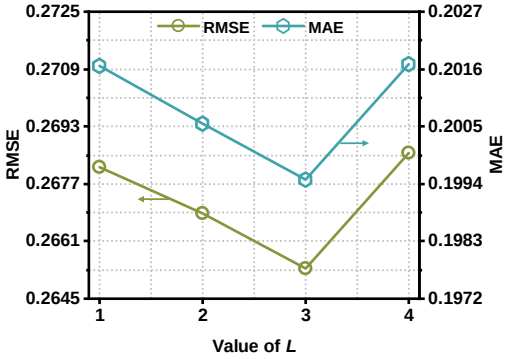
(c) Errors on D3



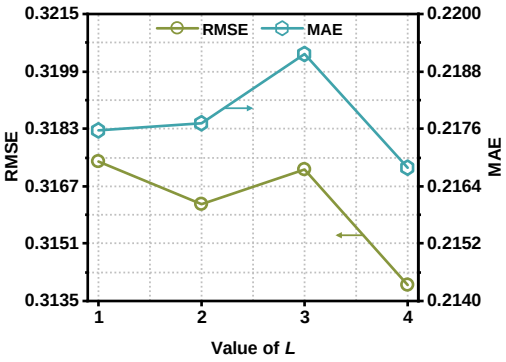
(d) Errors on D4



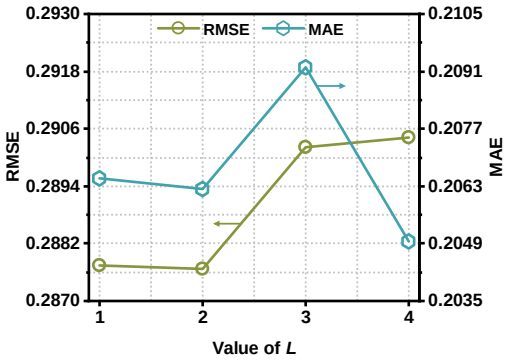
(e) Errors on D5



(f) Errors on D6

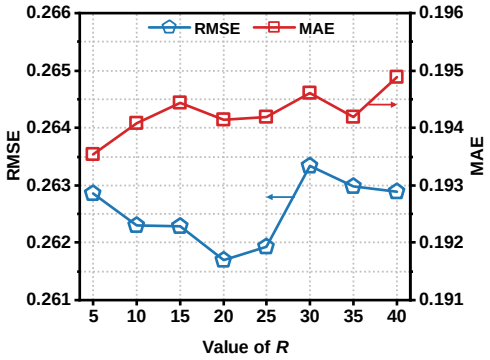


(g) Errors on D7

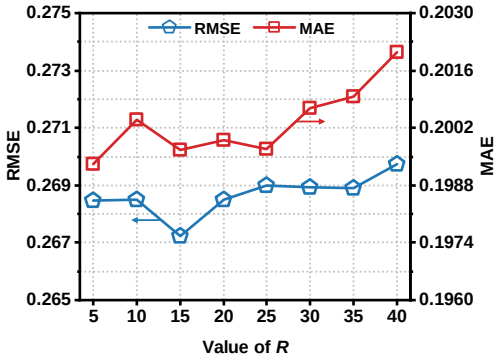


(h) Errors on D8

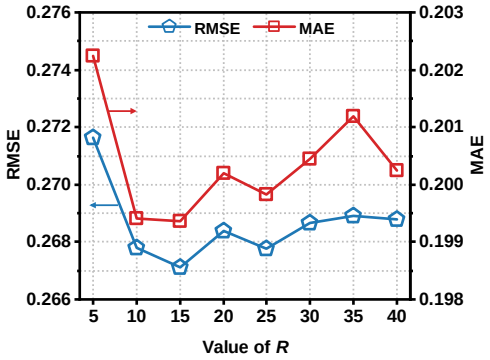
Fig. S2. Errors of DGM as L varies while fixing others on D1-8.



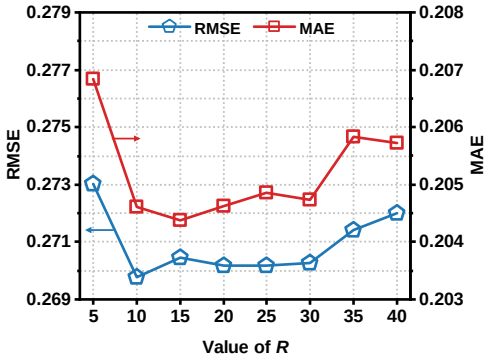
(a) Errors on D1



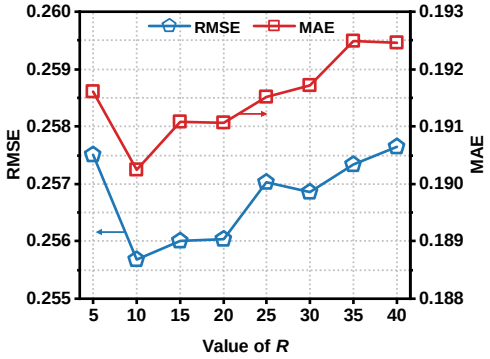
(b) Errors on D2



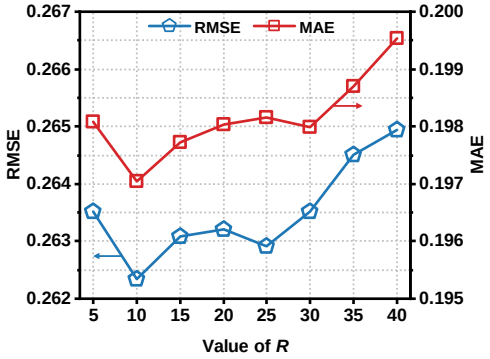
(c) Errors on D3



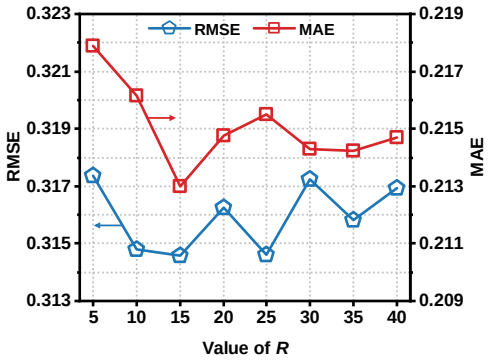
(d) Errors on D4



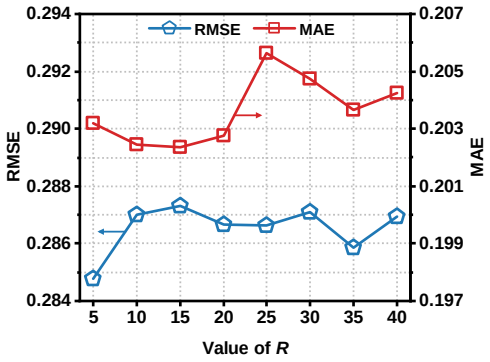
(e) Errors on D5



(f) Errors on D6



(g) Errors on D7



(h) Errors on D8

Fig. S3. Errors of DGM as R varies while fixing others on D1-8.