Graph Linear Convolution Pooling for Learning in Incomplete High-Dimensional Data Supplementary File

Fanghui Bi, Tiantian He, *Member, IEEE*, Yew Soon Ong, *Fellow, IEEE*, and Xin Luo, *Fellow, IEEE*

This is the supplementary file for the paper entitled "Graph Linear Convolution Pooling for Learning in Incomplete High-Dimensional Data". The supplementary symbol table, model discussions, proofs, details of experimental settings, and additional experimental results are put into this file and cited.

I. Supplementary Tables of Notations (Sec. II.A)

TABLE SI
Symbol Appointment.

Type	Symbol	Description
Set	M	Concerned node set.
	Λ, O	Known and unknown entry sets in an HDI matrix.
	Ψ, H, Ω	Training, validation, and testing datasets from Λ.
	X	Multiset containing the features of neighbor nodes.
	Γ	Element set in the multiset X.
	N(s), N(z)	First hop neighbor sets of nodes s and z.
	Φ	Entry set sampled randomly from Λ.
Set Element	s, z	Two example nodes in node set M.
	x, y	Features of neighbor nodes, which are elements of X/Γ.
Matrix	U	An |M|×|M| HDI matrix among M.
	$\hat{U}$	Rank-K approximation of U.
	$P^{(l)}$, $X^{(l)}$	|M|×K feature matrices of the l-th propagation layers.
	$W_P^{(l)}$, $W_X^{(l)}$	Weighted matrix for linear transformation on $P^{(l)}$ and $X^{(l)}$.
	A	|M|×|M| adjacency matrix of U.
	I	|M|×|M| identity matrix corresponding to A.
	$\tilde{A}$	Adjacency matrix with self-connections, i.e., $A+I$.
	$\hat{A}$	A's symmetrically normalized matrix.
	G	|M|×|M| weighted adjacency matrix of U.
	$\hat{G}$	G's symmetrically normalized matrix.
	D_A, D_G	Degree matrices of A and G.
	P, X	Output pooling feature matrices of nodes.
	H	Graph signal matrix.
	$\tilde{L}$	Normalized Laplacian matrix.
Vector	$\boldsymbol{p}_s^{(l)}$, $\boldsymbol{x}_z^{(l)}$	l-th order neighborhood representations of nodes s and z.
	$\boldsymbol{p}_s$, $\boldsymbol{x}_z$	Final latent features of nodes s and z.
Scalar	K	Dimensionality of latent feature space.
	$u_{s,z}$, $\hat{u}_{s,z}$	Single entries in U and $\hat{U}$.
	L, l	Layer number and index of graph convolution layer.
	$w(s)$, $w(z)$	Element sums of rows in G related to nodes s and z.
	η, λ	Learning rate and L_2 regularization coefficient.
	α	Coefficient controlling the pooling effect.
	B, b	Batch size and index of a mini-batch.
	T, t	Preset maximum and current training epochs.
	$\varphi_{s,z}^{(l)}$	Attention coefficient between s and z at the l-th layer.
Function	$\boldsymbol{\sigma}(\cdot)$	Activation function of complete message propagation.
	$\boldsymbol{\tau}(\cdot)$	Loss function.
	$\boldsymbol{\xi}(\cdot)$	A specific learning model.
	$\boldsymbol{\xi}^*(\cdot)$	The real mapping from feature space to target space.
	$\boldsymbol{\delta}(\cdot)$	Aggregation function based on Eq. (6).
	$\boldsymbol{\omega}(\cdot)$	Locality-enhance pooling function based on Eq. (14).
	$\boldsymbol{\rho}(\cdot)$	Auxiliary function introduced in Eq. (S9).
Algorithm Cost	$\mathcal{T}_1$	Algorithm 1's time cost.
	$\mathcal{S}_1$	Algorithm 1's storage cost.
Number Field	$\mathbb{R}$	Field of real numbers.
Feature Space	$\mathbb{X}$	Countable feature space.
Operator	$\circ$	Interaction operator.
	$\lceil\cdot\rceil$	Ceiling function of an involved number.
	$\|\cdot\|$	Cardinality of the involved set.

II. Supplementary Model Discussions (Secs. IV.A and IV.B)

A. Comparison with Related Approaches (Sec. IV.A)

Here, we present the discussions to the relation and difference between the proposed GLCPN and its closely related methods.

i. Relation to MF

GLCPN is a novel GNN-incorporated LFA model that revolutionizes existing MF-based approaches. It has the same trained parameters as a vanilla MF-based model, but it differs from previous MF-based ones, e.g., NeuMF [7] and MetaMF [16], as GLCPN ingeniously incorporates the interaction graph into the training process to exploit the high-order connectivity. It is worth noting that GLCPN reduces to a conventional MF-based LFA model with L_2 regularization when L=0.

ii. Relation to GAT

Graph attention networks are state-of-the-art methods and are widely used in many graph-related tasks [20], [30]. Their core idea is to adaptively compute the attention scores between the target node and its neighbors in the message aggregation process. Considering applying this strategy to our situation and taking $\boldsymbol{p}_s^{(l+1)}$ as an example, the attention weight of two nodes in the (l+1)-th layer is defined as:

$$\varphi_{s,z}^{(l)} = \frac{\exp\left(\text{LeakyReLU}\left(\left(\tilde{\boldsymbol{a}}^{(l)}\right)^T\left[\boldsymbol{p}_s^{(l)} \middle\| \boldsymbol{p}_z^{(l)}\right]\right)\right)}{\sum_{i\in\text{N}(s)}\exp\left(\text{LeakyReLU}\left(\left(\tilde{\boldsymbol{a}}^{(l)}\right)^T\left[\boldsymbol{p}_s^{(l)} \middle\| \boldsymbol{p}_i^{(l)}\right]\right)\right)}, \tag{S1}$$

where $\varphi_{s,z}^{(l)}$ is the attention coefficient between s and z, and $\tilde{\boldsymbol{a}}^{(l)}\in\mathbb{R}^{2K}$ is a shared weight vector for calculating the attention. Then, $\boldsymbol{p}_s^{(l+1)}$ is achieved as:

$$\boldsymbol{p}_s^{(l+1)} = \sum_{i\in\text{N}(s)} \varphi_{s,i}^{(l)} \boldsymbol{p}_i^{(l)}. \tag{S2}$$

Note that compared to the priori convolution operator in Eq. (6), the calculation for the attention coefficient is time-consuming. And it may cause negative effects because $P^{(0)}$ and $X^{(0)}$ are latent features optimized by the neural network, thereby resulting in accuracy loss.

iii. Relation to LightGCN

In [27], the authors propose a light graph convolutional network for top-K recommendations, which removes nonlinearities and weight matrices in graph convolutional networks. Our model differentiates from LightGCN in the following four aspects. First, LightGCN is designed for arbitrary graphs aiming to rank known interactions higher than unknown ones, while GLCPN targets weighted graphs and performs accurate estimation for missing data. Second, LightGCN treats a target node's neighbors uniformly. Third, the message-passing process in LightGCN is chaotic when dealing with an HDI matrix since it focuses on propagating messages between two kinds of entities. Fourth, LightGCN lacks special consideration for the 0-th layer representations and simply combines different layers of representations as the output representations.

iv. Relation to PageRank

PageRank is a classic algorithm that defines a random walk on the graph [34]. It computes the PageRank values of web pages by $V_{pr\_lim}=A_{rw}V_{pr\_lim}$, where $A_{rw}=AD^{-1}$, and V_{pr_lim} is the limit distribution of PageRank values. Considering our message propagation rules in Eq. (7), with $L\to\infty$, we obtain the limit approximations of node representations as:

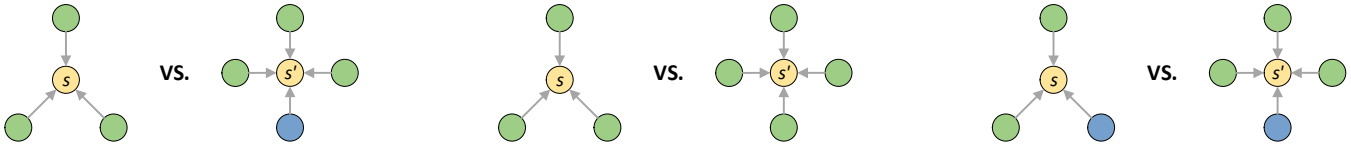
$$\begin{cases} P^{(L+1)} = P^{(L)} = \hat{G}P^{(L)}, \\ X^{(L+1)} = X^{(L)} = \hat{G}X^{(L)}. \end{cases} \tag{S3}$$

Apart from the difference that $\hat{G}$ is symmetrically normalized, Eq. (S3) has the same form as PageRank. This implies that our message propagation of infinite layers is closely related to PageRank. However, unlike the final stable probability distribution in PageRank, the limit node representations are over-smoothed when addressing an HDI matrix and thus less expressive [34].

B. Expressivity Analysis based on 1-WL Test (Sec. IV.B)

i. Example Illustration

To further clarify the discussions in Sec. IV.B, we illustrate several examples to show the distinguishing effect of different aggregators in Fig. S1. In the following examples, we mainly consider several aggregators including min, max, mean, and sum, which are widely used in existing GNNs such as GCNs and GATs. Different aggregators may learn the same embeddings for nodes s and s' even though the graph structures associated with them differ substantially. To be specific, all the aggregators of min, max, mean, and sum can succeed to distinguish the two graph structures given in Fig. S1(a). However, for the two graphs shown in Fig. S1(b), the aggregators of min, max, and mean fail to distinguish their structures, only sum aggregator succeeds. Besides, in another example of Fig. S1(c), the mean and sum aggregators succeed in distinguishing, but the min and max aggregators fail. According to Eq. (15), the equal representations for different nodes s and s' (i.e., $\boldsymbol{p}_s=\boldsymbol{p}_{s'}$) will result in the equal estimations (i.e., $\hat{u}_{s,z}=\hat{u}_{s',z}$), indicating a reduction in the solution space of $\boldsymbol{\xi}^*(\cdot)$. Hence, to learn more expressive representations from HDI data, a learning model based on GNNs should possess higher distinguishing ability [36], [39].



(a) Example 1 (b) Example 2 (c) Example 3

Fig. S1. Several examples of graph structures that illustrate the distinguishing ability of some aggregators, where the output representations for a graph structure can be viewed as the combinations of colors (features of the nodes adopted by GNNs). For example, in Fig. S1(a), without considering self-connections, the sum aggregator for node *s* fully preserves the three green features, whereas the max aggregator may only retain a single green feature.

ii. Proof of Theorem 1

To prove ***Theorem* 1**, two directions of the iff conditions should be considered. If the conditions that $\Gamma_1=\Gamma_2=\Gamma$ and $\kappa\cdot\sum_{y=x,y\in \mathrm{X}_1} g_{c_1,y}\big/\sqrt{w(y)}=\sum_{y=x,y\in \mathrm{X}_2} g_{c_2,y}\big/\sqrt{w(y)}$ are given, for $\kappa=\sqrt{w(c_2)/w(c_1)}$ and $x\in\Gamma$. According to Eq. (6), we have the following equations:

$$\begin{cases}\boldsymbol{\delta}(c_i,\mathrm{X}_i)=\sum\limits_{x\in \mathrm{X}_i}\hat{g}_{c_i,x}x,\\ \hat{g}_{c_i,x}=\dfrac{g_{c_i,x}}{\sqrt{w(c_i)\cdot w(x)}}.\end{cases}\tag{S4}$$

Given Eq. (S4), we derive:

$$\begin{cases}\boldsymbol{\delta}(c_1,\mathrm{X}_1)=\sum\limits_{x\in \mathrm{X}_1}\hat{g}_{c_1,x}x=\sum\limits_{x\in \mathrm{X}_1}\dfrac{g_{c_1,x}}{\sqrt{w(c_1)\cdot w(x)}}x,\\ \boldsymbol{\delta}(c_2,\mathrm{X}_2)=\sum\limits_{x\in \mathrm{X}_2}\hat{g}_{c_2,x}x=\sum\limits_{x\in \mathrm{X}_2}\dfrac{g_{c_2,x}}{\sqrt{w(c_2)\cdot w(x)}}x.\end{cases}\tag{S5}$$

With the condition that $\Gamma_1=\Gamma_2=\Gamma$, we have:

$$\begin{aligned}&\boldsymbol{\delta}(c_1,\mathrm{X}_1)-\boldsymbol{\delta}(c_2,\mathrm{X}_2)\\&=\sum_{x\in\Gamma}\left[\sum_{y=x,y\in \mathrm{X}_1}\frac{g_{c_1,y}}{\sqrt{w(c_1)\cdot w(y)}}-\sum_{y=x,y\in \mathrm{X}_2}\frac{g_{c_2,y}}{\sqrt{w(c_2)\cdot w(y)}}\right]\cdot x.\end{aligned}\tag{S6}$$

Given $\kappa\cdot\sum_{y=x,y\in \mathrm{X}_1} g_{c_1,y}\big/\sqrt{w(y)}=\sum_{y=x,y\in \mathrm{X}_2} g_{c_2,y}\big/\sqrt{w(y)}$, for $\kappa=\sqrt{w(c_2)/w(c_1)}$, based on Eq. (S6), we directly have $\boldsymbol{\delta}(c_1, \mathrm{X}_1)=\boldsymbol{\delta}(c_2, \mathrm{X}_2)$.

If the conditions that $\boldsymbol{\delta}(c_1, \mathrm{X}_1)=\boldsymbol{\delta}(c_2, \mathrm{X}_2)$, we can prove that the conditions mentioned in ***Theorem* 1** are necessary by showing the contradictions while they are not satisfied. Given $\boldsymbol{\delta}(c_1, \mathrm{X}_1)=\boldsymbol{\delta}(c_2, \mathrm{X}_2)$, we have:

$$\begin{aligned}&\boldsymbol{\delta}(c_1,\mathrm{X}_1)-\boldsymbol{\delta}(c_2,\mathrm{X}_2)\\&=\sum_{x\in \mathrm{X}_1}\frac{g_{c_1,x}}{\sqrt{w(c_1)\cdot w(x)}}x-\sum_{x\in \mathrm{X}_2}\frac{g_{c_2,x}}{\sqrt{w(c_2)\cdot w(x)}}x=0.\end{aligned}\tag{S7}$$

First, we assume $\Gamma_1\neq\Gamma_2$ for all $\Gamma_1, \Gamma_2\in\mathbb{X}$, the following equations are achieved:

$$\begin{aligned}&\boldsymbol{\delta}(c_1,\mathrm{X}_1)-\boldsymbol{\delta}(c_2,\mathrm{X}_2)\\&=\sum_{x\in\Gamma_1\cap\Gamma_2}\left[\sum_{y=x,y\in \mathrm{X}_1}\frac{g_{c_1,y}}{\sqrt{w(c_1)\cdot w(y)}}-\sum_{y=x,y\in \mathrm{X}_2}\frac{g_{c_2,y}}{\sqrt{w(c_2)\cdot w(y)}}\right]\cdot x\\&+\sum_{x\in\Gamma_1\setminus\Gamma_2}\sum_{y=x,y\in \mathrm{X}_1}\frac{g_{c_1,y}}{\sqrt{w(c_1)\cdot w(y)}}\cdot x-\sum_{x\in\Gamma_2\setminus\Gamma_1}\sum_{y=x,y\in\Gamma_2}\frac{g_{c_2,y}}{\sqrt{w(c_2)\cdot w(y)}}\cdot x=0.\end{aligned}\tag{S8}$$

Since Eq. (S8) holds for any x, we could define a function $\boldsymbol{\rho}(\cdot)$ as:

$$x=\begin{cases}\boldsymbol{\rho}(x),\ \text{for } x\in\Gamma_1\bigcap\Gamma_2;\\ \boldsymbol{\rho}(x)-1,\ \text{for } x\in\Gamma_1\setminus\Gamma_2;\\ \boldsymbol{\rho}(x)+1,\ \text{for } x\in\Gamma_2\setminus\Gamma_1.\end{cases}\tag{S9}$$

And if Eq. (S8) holds, we also infer that:

$$\begin{aligned}
&\boldsymbol{\delta}(c_1,\mathrm{X}_1)-\boldsymbol{\delta}(c_2,\mathrm{X}_2)\\
&=\sum_{x\in\Gamma_1\cap\Gamma_2}\left[\sum_{y=x,y\in\Gamma_1}\frac{g_{c_1,y}}{\sqrt{w(c_1)\cdot w(y)}}-\sum_{y=x,y\in\Gamma_2}\frac{g_{c_2,y}}{\sqrt{w(c_2)\cdot w(y)}}\right]\cdot\boldsymbol{\rho}(x)\\
&+\sum_{x\in\Gamma_1\setminus\Gamma_2}\sum_{y=x,y\in\Gamma_1}\frac{g_{c_1,y}}{\sqrt{w(c_1)\cdot w(y)}}\cdot\boldsymbol{\rho}(x)-\sum_{x\in\Gamma_2\setminus\Gamma_1}\sum_{y=x,y\in\Gamma_2}\frac{g_{c_2,y}}{\sqrt{w(c_2)\cdot w(y)}}\cdot\boldsymbol{\rho}(x)=0.
\end{aligned}\tag{S10}$$

By substituting Eq. (S9) into Eq. (S10), we infer:

$$\begin{aligned}
&\boldsymbol{\delta}(c_1,\mathrm{X}_1)-\boldsymbol{\delta}(c_2,\mathrm{X}_2)\\
&=\sum_{x\in\Gamma_1\cap\Gamma_2}\left[\sum_{y=x,y\in\mathrm{X}_1}\frac{g_{c_1,y}}{\sqrt{w(c_1)\cdot w(y)}}-\sum_{y=x,y\in\mathrm{X}_2}\frac{g_{c_2,y}}{\sqrt{w(c_2)\cdot w(y)}}\right]\cdot x\\
&+\sum_{x\in\Gamma_1\setminus\Gamma_2}\sum_{y=x,y\in\Gamma_1}\frac{g_{c_1,y}}{\sqrt{w(c_1)\cdot w(y)}}\cdot(x+1)\\
&-\sum_{x\in\Gamma_2\setminus\Gamma_1}\sum_{y=x,y\in\Gamma_2}\frac{g_{c_2,y}}{\sqrt{w(c_2)\cdot w(y)}}\cdot(x-1)=0.
\end{aligned}\tag{S11}$$

As Eq. (S8) is equal to Eq. (S11), we have:

$$\begin{aligned}
&\boldsymbol{\delta}(c_1,\mathrm{X}_1)-\boldsymbol{\delta}(c_2,\mathrm{X}_2)\\
&=\sum_{x\in\Gamma_1\cap\Gamma_2}\left[\sum_{y=x,y\in\mathrm{X}_1}\frac{g_{c_1,y}}{\sqrt{w(c_1)\cdot w(y)}}-\sum_{y=x,y\in\mathrm{X}_2}\frac{g_{c_2,y}}{\sqrt{w(c_2)\cdot w(y)}}\right]\cdot x\\
&+\sum_{x\in\Gamma_1\setminus\Gamma_2}\sum_{y=x,y\in\Gamma_1}\frac{g_{c_1,y}}{\sqrt{w(c_1)\cdot w(y)}}\cdot x-\sum_{x\in\Gamma_2\setminus\Gamma_1}\sum_{y=x,y\in\Gamma_2}\frac{g_{c_2,y}}{\sqrt{w(c_2)\cdot w(y)}}\cdot x\\
&+\sum_{x\in\Gamma_1\setminus\Gamma_2}\sum_{y=x,y\in\Gamma_1}\frac{g_{c_1,y}}{\sqrt{w(c_1)\cdot w(y)}}-\sum_{x\in\Gamma_2\setminus\Gamma_1}\sum_{y=x,y\in\Gamma_2}\frac{g_{c_2,y}}{\sqrt{w(c_2)\cdot w(y)}}\\
&=\sum_{x\in\Gamma_1\setminus\Gamma_2}\sum_{y=x,y\in\mathrm{X}_1}\frac{g_{c_1,y}}{\sqrt{w(c_1)\cdot w(y)}}+\sum_{x\in\Gamma_2\setminus\Gamma_1}\sum_{y=x,y\in\Gamma_2}\frac{g_{c_2,y}}{\sqrt{w(c_2)\cdot w(y)}}=0.
\end{aligned}\tag{S12}$$

Since the terms in the above summation operators are positive, i.e., Eq. (S12) cannot hold obviously. Thus, the assumption that $\Gamma_1\neq\Gamma_2$ is false. Thus, given $\boldsymbol{\delta}(c_1, \mathrm{X}_1)=\boldsymbol{\delta}(c_2, \mathrm{X}_2)$, we have $S_1=S_2$.

Furthermore, based on $\Gamma_1=\Gamma_2=\Gamma$, we have the following inference:

$$\sum_{x\in\Gamma_1\setminus\Gamma_2}\sum_{y=x,y\in\mathrm{X}_1}\frac{g_{c_1,y}}{\sqrt{w(c_1)\cdot w(y)}}\cdot x-\sum_{x\in\Gamma_2\setminus\Gamma_1}\sum_{y=x,y\in\Gamma_2}\frac{g_{c_2,y}}{\sqrt{w(c_2)\cdot w(y)}}\cdot x=0.\tag{S13}$$

Hence, according to Eq. (S8) and Eq. (S13), we have:

$$\boldsymbol{\delta}(c_1,\mathrm{X}_1)-\boldsymbol{\delta}(c_2,\mathrm{X}_2)=\sum_{x\in\Gamma}\left[\sum_{y=x,y\in\Gamma_1}\frac{g_{c_1,y}}{\sqrt{w(c_1)\cdot w(y)}}-\sum_{y=x,y\in\mathrm{X}_2}\frac{g_{c_2,y}}{\sqrt{w(c_2)\cdot w(y)}}\right]\cdot x=0.\tag{S14}$$

Obviously, each term in the above summation is equal to zero:

$$\sum_{y=x,y\in\mathrm{X}_1}\frac{g_{c_1,y}}{\sqrt{w(c_1)\cdot w(y)}}-\sum_{y=x,y\in\mathrm{X}_2}\frac{g_{c_2,y}}{\sqrt{w(c_2)\cdot w(y)}}=0.\tag{S15}$$

Hence, we obtain:

$$\sum_{y=x,y\in\mathrm{X}_1}\frac{g_{c_1,y}}{\sqrt{w(c_1)}\cdot\sqrt{w(y)}}-\sum_{y=x,y\in\mathrm{X}_2}\frac{g_{c_2,y}}{\sqrt{w(c_2)}\cdot\sqrt{w(y)}}=\frac{1}{\sqrt{w(c_1)}}\cdot\sum_{y=x,y\in\mathrm{X}_1}\frac{g_{c_1,y}}{\sqrt{w(y)}}-\frac{1}{\sqrt{w(c_2)}}\cdot\sum_{y=x,y\in\mathrm{X}_2}\frac{g_{c_2,y}}{\sqrt{w(y)}}=0.\tag{S16}$$

Eq. (S16) can be rewritten as:

$$\sqrt{\frac{w(c_2)}{w(c_1)}}\cdot\sum_{y=x,y\in\mathrm{X}_1}\frac{g_{c_1,y}}{\sqrt{w(y)}}=\sum_{y=x,y\in\mathrm{X}_2}\frac{g_{c_2,y}}{\sqrt{w(y)}}.\tag{S17}$$

We further set $\kappa=\sqrt{w(c_2)/w(c_1)}$, we have $\kappa\cdot\sum_{y=x,y\in\mathrm{X}_1}g_{c_1,y}/\sqrt{w(y)}=\sum_{y=x,y\in\mathrm{X}_2}g_{c_2,y}/\sqrt{w(y)}$, for $x\in\Gamma$. Therefore, based on these inferences, ***Theorem* 1** holds.□

iii. Proof of Corollary 1

Following ***Theorem* 1**, the pooling function using the strategy in Eq. (14) can be denoted as $\boldsymbol{\omega}(c, \mathrm{X}^{(1)}, \mathrm{X}^{(2)},\ldots, \mathrm{X}^{(L)})=(\alpha+\varepsilon/|\mathrm{X}|)\cdot c+\sum_{l=\{1\sim L\},\, x\in \mathrm{X}^{(l)}} h(c, x)$, $c, \mathrm{X}^{(1)}, \mathrm{X}^{(2)},\ldots, \mathrm{X}^{(L)}\in\mathbb{X}$, where $|\mathrm{X}^{(1)}|=|\mathrm{X}^{(2)}|=\ldots=|\mathrm{X}^{(L)}|=|\mathrm{X}|$. According to ***Theorem* 1**, it is seen that when the condition that $\mathrm{X}_1^{(l)}=\{\Gamma, \mu_1\}$, $\mathrm{X}_2^{(l)}=\{\Gamma, \mu_2\}$, and $\kappa\cdot\sum_{y=x, y\in \mathrm{X}_1} g_{c_1, y}\Big/\sqrt{w(y)}=\sum_{y=x, y\in \mathrm{X}_2} g_{c_2, y}\Big/\sqrt{w(y)}$, for $\kappa=\sqrt{w(c_2)/w(c_1)}$, $x\in\Gamma$, and $l=\{1\sim L\}$ is fulfilled, $\boldsymbol{\delta}(c_1, \mathrm{X}_1^{(l)})-\boldsymbol{\delta}(c_2, \mathrm{X}_1^{(l)})=0$, for $l=\{1\sim L\}$, i.e., the aggregation function in Eq. (6) cannot distinguish different graph structures. To prove ***Corollary* 1**, it should be proved that $\boldsymbol{\omega}(\cdot)$ can correctly distinguish all different structures that the aggregation function in Eq. (6) fails previously. To do so, two cases require to be considered.

(1)$c_1\neq c_2$.

Given $\mathrm{X}_1^{(l)}=\{\Gamma, \mu_1\}$, $\mathrm{X}_2^{(l)}=\{\Gamma, \mu_2\}$, and $\kappa\cdot\sum_{y=x, y\in \mathrm{X}_1} g_{c_1, y}\Big/\sqrt{w(y)}=\sum_{y=x, y\in \mathrm{X}_2} g_{c_2, y}\Big/\sqrt{w(y)}$, for $\kappa=\sqrt{w(c_2)/w(c_1)}$, $x\in\Gamma$, and $l=\{1\sim L\}$, we have $\boldsymbol{\delta}(c_1, \mathrm{X}_1^{(l)})-\boldsymbol{\delta}(c_2, \mathrm{X}_2^{(l)})=0$, for $l=\{1\sim L\}$ based on ***Theorem* 1**. Hence, we have $\boldsymbol{\omega}(c_1, \mathrm{X}_1^{(1)}, \mathrm{X}_1^{(2)},\ldots, \mathrm{X}_1^{(L)})-\boldsymbol{\omega}(c_2, \mathrm{X}_2^{(1)}, \mathrm{X}_2^{(2)},\ldots, \mathrm{X}_2^{(L)})=(\alpha+\varepsilon/|\mathrm{X}_1|)\cdot c_1-(\alpha+\varepsilon/|\mathrm{X}_2|)\cdot c_2$. As $c_1\neq c_2$, $\boldsymbol{\omega}(c_1, \mathrm{X}_1^{(1)}, \mathrm{X}_1^{(2)},\ldots, \mathrm{X}_1^{(L)})\neq\boldsymbol{\omega}(c_2, \mathrm{X}_2^{(1)}, \mathrm{X}_2^{(2)},\ldots, \mathrm{X}_2^{(L)})$ is obvious.

(2)$c_1=c_2$.

Similarly, we have $\boldsymbol{\omega}(c_1, \mathrm{X}_1^{(1)}, \mathrm{X}_1^{(2)},\ldots, \mathrm{X}_1^{(L)})-\boldsymbol{\omega}(c_2, \mathrm{X}_2^{(1)}, \mathrm{X}_2^{(2)},\ldots, \mathrm{X}_2^{(L)})=(\alpha+\varepsilon/|\mathrm{X}_1|)\cdot c_1-(\alpha+\varepsilon/|\mathrm{X}_2|)\cdot c_2$. Considering the condition that $c_1=c_2$, we thereby have the following inference:

$$\boldsymbol{\omega}\left(c_1, \mathrm{X}_1^{(1)}, \mathrm{X}_1^{(2)},\ldots, \mathrm{X}_1^{(L)}\right)-\boldsymbol{\omega}\left(c_2, \mathrm{X}_2^{(1)}, \mathrm{X}_2^{(2)},\ldots, \mathrm{X}_2^{(L)}\right)=\left(\alpha+\frac{\varepsilon}{|\mathrm{X}_1|}\right)c-\left(\alpha+\frac{\varepsilon}{|\mathrm{X}_2|}\right)c=\varepsilon\left(\frac{1}{|\mathrm{X}_1|}-\frac{1}{|\mathrm{X}_2|}\right)c, \tag{S18}$$

where $|\mathrm{X}_1|=|\mathrm{N}(c_1)|$ and $|\mathrm{X}_2|=|\mathrm{N}(c_2)|$. Since $|\mathrm{X}_1|\neq|\mathrm{X}_2|$, $\boldsymbol{\omega}(c_1, \mathrm{X}_1^{(1)}, \mathrm{X}_1^{(2)},\ldots, \mathrm{X}_1^{(L)})\neq\boldsymbol{\omega}(c_2, \mathrm{X}_2^{(1)}, \mathrm{X}_2^{(2)},\ldots, \mathrm{X}_2^{(L)})$, meaning that the locality-enhanced holistic pooling function $\boldsymbol{\omega}(\cdot)$ based on Eqs. (6) and (14) can successfully distinguish the graph structures that solely utilizing the aggregation function in Eq. (6) fails to distinguish previously. Thus, ***Corollary* 1** holds.□

C. Spectral Expressivity Analysis (Sec. IV.C)

Note that an l-order polynomial filter on a given graph signal H is defined as $\theta_l\tilde{L}^lH$, where θ_l represents the corresponding polynomial coefficients [19], [41]. According to Eq. (14), the first~L-th order polynomial filter applied to the graph signals $P^{(0)}$ and $X^{(0)}$ can be reformulated using the normalized Laplacian matrix $\tilde{L}$ as:

$$\sum_{l=1}^{L}\alpha(1-\alpha)^l\left(I-\tilde{L}\right)^l. \tag{S19}$$

Further, by expanding Eq. (19), the l-th polynomial filtering term $\theta_l\tilde{L}^l$ can be derived. Concretely, for $l\in[1, L]$, the filter coefficients are given as:

$$\theta_l=\sum_{i=l}^{L}\alpha(1-\alpha)^i(-1)^l\binom{i}{l}. \tag{S20}$$

From the filter coefficients, we observe that compared to the case without locality-enhanced pooling, GLCPN exhibits higher flexibility owing to its adjustable factor α. Therefore, according to prior studies [41], which suggest that flexible coefficients are significantly beneficial for preventing over-smoothing, this enhancement in GLCPN's ability to fit the coefficients of polynomial filters improves its capacity for effective representation learning in HDI data.

III. Supplementary Descriptions of General Settings (Sec. V.A)

A. Details of Datasets

Ten real-world industrial HDI datasets raised from four distinct applications are used in our experiments. They are *T-208964*, *T-227321*, *T-9544*, *T-9796*, *Astro-Ph*, *Cond-Mat*, *Cond-Mat-2003*, *Cond-Mat-2005*, *Throughput*, and *Kiba.* The detailed descriptions of them are presented as follows.

(1)PPI networks: *T-208964*, *T-227321*, *T-9544*, and *T-9796* are undirected PPI networks gathered from the protein interactome database STRING [1], [2]. They respectively record the protein-protein interactions for the species Macaca mulatta, Equus caballus, Pseudomonas aeruginosa, and Aspergillus nidulans. In a PPI network, the nodes represent specific proteins, the edges represent the observed protein-protein interactions, and the weights represent the confidences of interactomes between proteins. All the weight values have been normalized to the range between 0 and 1. *T-208964* consists of 5,565 proteins and 1,559,616 interactomes; *T-227321* consists of 7,963 proteins and 1,120,030 interactomes; *T-9544* consists of 20,462 proteins and 15,597,778 interactomes; and *T-9796* consists of 19,664 proteins and 12,149,608 interactomes. Their densities are respectively 5.04%, 1.77%, 3.73%, and 3.14%. More descriptions of them including the original datasets can be found in the following URLs: https://version-12-0.string-db.org/organism/208964, https://version-12-0.string-db.org/organism/227321, https://version-12-0.string-db.org/organism/9544, and https://version-12-0.string-db.org/organism/9796.

(2)Scientific collaboration networks: *Astro-Ph*, *Cond-Mat*, *Cond-Mat-2003*, and *Cond-Mat-2005* are from the incomplete matrix collection of University of Florida [42], which describe the co-authorships between scientists posting preprints on the archives of astrophysics (Astro-Ph) and condensed matter (Cond-Mat) at *www.arxiv.org* in different periods [5], [42]. To be specific, *Astro-Ph* and *Cond-Mat* record the co-authorships between Jan. 1, 1995 and Dec. 31, 1999; *Cond-Mat-2003* includes all preprints posted between Jan. 1, 1995 and June 30, 2003; and *Cond-Mat-2005* includes the co-authorship records between Jan. 1, 1995 and Mar. 31, 2005. The undirected networks are weighted, and the weights are assigned as described in their original papers. *Astro-Ph* consists of 16,706 nodes (i.e., the involved scientists) and 242,502 edges (i.e., the collaborations); *Cond-Mat* consists of 16,726 nodes and 95,188 edges; *Cond-Mat-2003* consists of 31,163 nodes and 240,058 edges; and *Cond-Mat-2005* consists of 40,421 nodes and 351,382 edges. They are all sparse and the densities are only 0.09%, 0.03%, 0.02%, and 0.02%. More details and the corresponding HDI matrices describing them can be found on https://sparse.tamu.edu/Newman/astro-ph, https://sparse.tamu.edu/Newman/cond-mat, https://sparse.tamu.edu/Newman/cond-mat-2003, and https://sparse.tamu.edu/Newman/cond-mat-2005.

(3)QoS network: *Throughput* is collected by the WS-Dream system [6], [17], which records the throughputs of cloud services invoked by users. It reflects the fundamental QoS on the user side and is known as one of the most widely-used public QoS datasets in the communication of service computing [6], [17]. This is a bipartite graph including 339 users, 5,825 cloud services, and 1,831,253 invocation records. Its density reaches 92.74%, which is hard to achieve in most of real application scenarios. Hence, we employ random sampling of test cases that include a variety of low-density data split ratios for train-validation-test, i.e., 10%-70%-20%, 20%-60%-20%, 30%-50%-20%, and 40%-40%-20%. This approach enables us to perform an exhaustive evaluation of all models' representation learning effectiveness in HDI data with a broad range of density conditions. They are first published at https://wsdream.github.io/dataset/wsdream_dataset1.html.

(4)Drug-Protein Interaction (DPI) network: *Kiba* is a benchmark dataset for predicting the binding affinity between drugs and targets, which is created by Jing Tang at the University of Helsinki [43]. It encompasses joint kinase inhibitor bioactivities from various sources including Kd, Ki, and IC50. For training and prediction purposes, the weights are processed into logarithmically normalized KIBA scores. This dataset consists of 118,254 DPI pairs, 2,111 chemical compounds, and 229 kinase targets, resulting a density of 24.46%. Its details can be found at https://researchportal.helsinki.fi/en/datasets/kiba-a-benchmark-dataset-for-drug-target-prediction and the latest dataset can be downloaded from there.

Their properties are summarized in Table II. Note that the above ten datasets involve four diverse representation learning tasks to HDI data, including PPI prediction, co-authorship analysis, cloud service selection, and drug target prediction. Obviously, their distribution characteristics are inherently different and the densities vary from 0.02% to 92.74%, providing supportive evaluation to the robustness of GLCPN's performance. Considering that the original datasets may be updated, for the convenience of model reproduction and dataset usage, we have placed the datasets directly used in this paper on GitHub. All datasets can be found at https://github.com/Oak-B/GLCPN/blob/main/ GLCPN-Datasets.zip.

B. Details of Evaluation Metrics

In our experiments, four classic and commonly-used evaluation metrics are adopted to assess the performance of the models. They include three regression error metrics (i.e., RMSE, MAE, and R^2) and one ranking metric (i.e., NDCG). Specifically, the RMSE is calculated as the square root of the average of the squared differences between the predicted and truth values. It is mathematically represented as:

$$RMSE = \sqrt{\left(\sum_{u_{s,z} \in \Omega} \left(u_{s,z} - \hat{u}_{s,z} \right)^2 \right) \Big/ |\Omega|},$$

where Ω indicates the testing set. The MAE, on the other hand, is the average of the absolute differences between the predicted and actual values, it is calculated as:

$$MAE = \left(\sum_{u_{s,z} \in \Omega} \left| u_{s,z} - \hat{u}_{s,z} \right|_{abs} \right) \Big/ |\Omega|.$$

While the RMSE and MAE provide a measure of the average prediction error, they do not give us a sense of how well a tested model is performing in terms of the variance. This is where the coefficient of determination (i.e., R^2) comes into play. Hence, in the context of our experiments, we further employ R^2 to evaluate the performance of our models involved in this paper, providing a more comprehensive understanding of model accuracy. It is achieved as:

$$R^2 = 1 - \frac{\sum_{u_{s,z} \in \Omega} \left(u_{s,z} - \hat{u}_{s,z} \right)^2}{\sum_{u_{s,z} \in \Omega} \left(u_{s,z} - \overline{u}_{s,z} \right)^2},$$

where $\overline{u}_{s,z}$ is the mean of the actual values in Ω. A higher R^2 value signifies that a tested model can better fit the testing case [44]. In many cases, more accurate predictions have practical application significance. For instance, in the development of new drugs, excessive blinding affinity might lead to side effects that render the drug unsuitable for treatment. Therefore, accurate DTA prediction is crucial for drug applications [37], [43]. Simultaneously, precise PPI prediction can reflect whether a drug can effectively interfere with the target PPI [1]. In the tasks of interest in this paper, our primary focus is on the predictive accuracy of a tested approach towards the true values. The above several metrics can well evaluate the performance of GLCPN. However, the ranking ability of a model is also significant. For instance, although service selection is a complex issue involving multiple factors, accurate QoS estimation provides informative references [6], [17]. At this point, if more precise ranking can also be achieved on certain QoS characteristics, it would be more suitable for users to choose cloud services with less response time or higher throughput. Thus, we additionally adopt the NDCG to further test a model's ranking performance. It is achieved as [10]:

$$DCG@k\left(m,\hat{U}\right) = \sum_{i=1}^{k} \frac{2^{u_{m,i}} - 1}{\log_2(i+1)},$$

$$IDCG@k\left(m,\Omega\right) = \sum_{i=1}^{k} \frac{2^{u_{m,i}} - 1}{\log_2(i+1)},$$

$$NDCG@k = \frac{1}{|\mathrm{M}|} \sum_{m \in \mathrm{M}} \frac{DCG@k\left(m,\hat{U}\right)}{IDCG@k\left(m,\Omega\right)},$$

where the DCG indicates the discounted cumulative gain calculated based on the predicted ranking, the IDCG indicates the ideal discounted cumulative gain obtained based on the real ranking, $u_{m,j}$ is the actual value of the j-th entry in the descending sorted sequence regarding m, and k is a threshold parameter determines the count of entries to be considered in the ranking list. Unless mentioned specifically, k is generally chosen as 20 in our implementation. It is worth noting that a smaller RMSE/MAE or a larger R^2/NDCG indicates a higher model performance. The above four metrics are sufficient to provide a comprehensive evaluation to GLCPN's performance in representation learning to HDI data.

C. Details of Models for Comparisons

The details all the approaches involved in our experiments are presented as below.

(1) NeuMF [7] is a commonly adopted neural MF baseline, which extends the inner product to multilayered perception and combines them into the process of MF for achieving nonlinear node representations.
(2) MetaMF [16] is a federated MF-based model for LFA, which considers the privacy of nodes and learns private node representations by building a federated and meta-learning framework.
(3) GC-MC [25] is a graph convolutional matrix completion model, which achieves a graph autoencoder to learn the first-order connectivity information of nodes and proposes node dropout to improve the robustness.
(4) NGCF [24] is a neural graph collaborative filtering model that performs MF with a standard graph convolutional network, thereby effectively addressing the high-order connectivity in the interaction graph.
(5) LightGCN [27] is a light graph convolutional network-based LFA model that removes the nonlinear activation and feature transformation to enjoy a fast learning process for accurate node representations.
(6) LR-GCCF [35] is a linear residual graph convolutional network-based MF model, which removes nonlinearities but reserves the feature transformation and presents an empirical explanation of layer concatenation.
(7) SGL-ED [38] is a robust self-supervised graph learning-based model that uses edge dropout to generate multiple views of a node and maximizes the agreement between different views of the same node.
(8) DGCN-HN [32] is a graph convolutional network-based model with hybrid normalization, which deepens the message propagation with two kinds of residual connection approaches.
(9) LightGCN$_{r\text{-}AdjNorm}$ [45] is a variant of lightGCN, which investigates the accuracy-novelty performance and incorporates a plugin called *r-AdjNorm* to control the normalization strength.
(10) HMLET [8] is a hybrid method of linear and nonlinear collaborative filtering. It is a GNN-based LFA method that leverages a gating module to choose linear or nonlinear propagation of each node.

(11) HCCF [28] is a self-supervised hypergraph contrastive collaborative filtering model, which constructs a hypergraph with cross-view contrastive learning to capture the local and global collaborative relations.
(12) GTN [29] is a graph trend filtering network-based LFA model, which notices the non-adaptive propagation and non-robustness of existing approaches and proposes a principled technique to capture the adaptive reliability.
(13) CIGCN [46] is a channel-independent graph convolutional network-based model that performs transformation with diagonal parameter matrices, thus keeping independent embedding dimensions.
(14) JMP-GCF [33] is a GNN-based joint multi-grained popularity-aware collaborative filtering model, which constructs different granularities of popularity features and jointly learns the signals.
(15) LightGCL [31] is a simple graph contrastive learning-based LFA model, which uses singular value decomposition to refine the global collaborative relation and generate robust contrastive views.
(16) MGDN [47] is a Markov graph diffusion model, which incorporates an untrainable Markov process into traditional one-layer network representation learning to capture context features of vertices.
(17) PopGo [48] is a popularity-agnostic features aware approach, which considers the compositional degree of each node-node pair and learns the popularity representations to generate unbiased node representations.
(18) GLCPN is the proposed model in this paper.

D. Details of Training Settings

We apply the following established settings to the models involved in our experiments.

i. General Settings

For fair comparison, some essential general configurations for all the models including GLCPN are adopted. They include the initial method, optimization approach, early stopping strategies, data partitioning methods, strategies for tuning common hyperparameters, cross-validation settings, and sparse computation implementations. Specifically, the initial state of trainable variables is generated randomly using the normal distribution version of the Xavier method [49], and we use the Adam optimizer to train a target model. Each model's training process terminates if: a) the training epoch (t) reaches a preset threshold (T), i.e., one thousand epochs, or b) the representation ability keeps degrading for ten epochs. For the other nine datasets apart from *Throughput*, we employ the 70%-10%-20% train-validation-test settings, i.e., 70% of Λ are randomly chosen as the training set Ψ for model inference, 10% as the validation set H to monitor the training process, and the remaining 20% as the testing set Ω to test the achieved model's performance, where Ψ, H, and Ω are disjoint. Considering *Throughput*, we divide it into proportions of 10%-70%-20%, 20%-60%-20%, 30%-50%-20%, and 40%-40%-20% respectively, to validate the performance of GLCPN under different density conditions. In individually-built cases Ψ and H, we tune all the hyperparameters to obtain the lowest validation error on Ψ and then test the generalization performance on Φ with the optimized parameters. For all involved models, the dimensionality of latent feature space (K) is fixed to 64. The batch size is fixed to 2^{11} on the other eight datasets except for *T-9544* and *T-9796*. On these two larger-scale datasets, the batch size is fixed to 2^{14} for acceleration. The learning rate (η) is tuned in the range of {1e-5, 5e-5, 1e-4, 5e-4, 1e-3, 5e-3, 1e-2} and the L_2 regularization coefficient (λ) is tuned in the range of {1e-5, 1e-4, 1e-3, 1e-2, 1e-1}. Once the hyperparameters are determined, the splitting, training, and testing process is performed ten times to record the final averaged results and standard deviations. The sparse computation of each graph convolutional layer of GNN-based models is implemented by using the sparse matrix operators of PyTorch Geometric.

ii. Configurations of GLCPN

Apart from the common hyperparameters such as the learning rate and the L_2 regularization coefficient, the proposed GLCPN introduces a fundamental coefficient (α) controlling the pooling effect. For the three hyperparameters, considering that excessive hyperparameter tuning may affect the fairness of experimental comparisons, we simply set them in our comparison experiments. To be specific, on all the PPI/QoS/DPI datasets, the learning rate is set to 1e-2, the L_2 regularization coefficient is set to 1e-4, and the pooling coefficient is set to 0.1; and on the four scientific collaboration networks, we set the learning rate at 1e-3, the L_2 regularization coefficient at 1e-4, and the pooling coefficient at 0.5. As for the number of graph convolutional layer (L), we tune it in the scale of {1, 2, 3, 4} like other GNN-based models. More implementation details of GLCPN can be found in our codes shared at https://github.com/Oak-B/GLCPN/blob/main/GLCPN-Codes.zip.

iii. Configurations of Comparison Methods

We reproduced the source codes of all the comparison models based on their sharing repositories in PyTorch. To fairly compare the models, we maintain the structure of each model with care while adding L_2 regularization to those models do not use regularization, and we modify the classification loss to regression loss for certain models to suit the tasks at hand. For the non-GNN models such as NeuMF [7] and MetaMF [16], their structures are implemented according to the best suggestions of the authors. For GC-MC [25], we adopt a one-layer network structure for its graph encoder module. For all the other GNN-based LFA models like NGCF [24], DGCN-HN [32], and MGDN [47], the graph convolutional layer number is tuned among {1, 2, 3, 4}. For SGL [38], we choose to implement its edge dropout version, i.e., SGL-ED, and we implement the "End" version of HMLET [8]. For the study introducing *r-AdjNorm* [45], we incorporate it into LightGCN to achieve LightGCN$_{r\text{-}AdjNorm}$, which is also compared in the original paper. For HCCF [28], the number of hypergraph convolutional layer is the same as that of its GCN layer. For JMP-GCF [33], we adopt the architectures equipped with three granularities of popularity features. Besides, there are too many other unique hyperparameters to tune easily, so we use the recommended settings in their papers.

IV. Supplementary Tables of Empirical Studies (Secs. V.B and V.D)

A. Results of Comparison Experiments (Sec. V.B)

- Table SII reports the supplementary results regarding accuracy comparisons.

TABLE SII
SUPPLEMENTARY RESULTS OF THE PERFORMANCE ON PPI PREDICTION AND CP PREDICTION.

Method	*T-208964*		*T-227321*		*Astro-Ph*		*Cond-Mat-2005*	
	RMSE ↓	**MAE** ↓	**RMSE** ↓	**MAE** ↓	**RMSE** ↓	**MAE** ↓	**RMSE** ↓	**MAE** ↓
NeuMF	0.1880±9.1E-2	0.1429±8.7E-2	0.1352±3.1E-4	0.0949±2.3E-4	0.5058±9.1E-2	0.1997±4.0E-2	0.9979±3.1E-3	0.4564±4.4E-3
MetaMF	0.1438±2.4E-3	0.0918±2.2E-3	0.1413±5.8E-4	0.0917±5.4E-4	0.5129±4.0E-3	0.2367±6.8E-3	0.8492±2.5E-3	0.4018±3.2E-3
GC-MC	0.1280±5.4E-4	0.0954±1.6E-3	0.1276±2.4E-4	0.0907±4.4E-4	0.4741±1.4E-2	0.2231±3.9E-3	0.8241±2.6E-2	0.4057±1.2E-2
NGCF	0.1370±1.4E-3	0.1007±1.3E-3	0.1417±2.2E-3	0.1003±1.9E-3	0.6277±1.1E-2	0.3002±6.7E-3	1.0315±1.8E-2	0.6546±1.1E-2
LightGCN	0.1187±3.3E-4	0.0839±1.9E-4	0.1183±3.6E-4	0.0818±1.7E-4	0.4673±1.6E-2	0.1813±3.6E-3	0.8169±1.8E-2	0.3518±2.2E-3
LR-GCCF	0.1265±2.5E-4	0.0905±2.8E-4	0.1241±2.9E-4	0.0870±2.8E-4	0.4933±1.6E-2	0.2138±3.3E-3	0.8441±1.7E-2	0.3858±2.6E-3
DGCN-HN	0.1232±4.1E-4	0.0876±2.1E-4	0.1222±2.7E-4	0.0851±1.4E-4	0.4757±1.7E-2	0.1856±3.6E-3	0.8229±1.9E-2	0.3489±1.7E-3
GLCPN (Ours)	**0.1161±3.7E-4**	**0.0810±3.2E-4**	**0.1164±2.8E-4**	**0.0800±9.7E-5**	**0.4527±1.7E-2**	**0.1628±3.7E-3**	**0.8015±1.8E-2**	**0.3200±2.4E-3**
Method	*T-9544*		*T-9796*		*Cond-Mat*		*Cond-Mat-2003*	
	RMSE ↓	**MAE** ↓	**RMSE** ↓	**MAE** ↓	**RMSE** ↓	**MAE** ↓	**RMSE** ↓	**MAE** ↓
NeuMF	0.1064±4.6E-5	0.0710±5.2E-5	0.1375±7.6E-5	0.0998±5.8E-5	0.9384±9.0E-3	0.4856±4.0E-3	1.0164±9.4E-3	0.4557±5.1E-3
MetaMF	0.1117±1.4E-3	0.0698±1.1E-3	0.1393±2.9E-3	0.0992±2.0E-3	0.8310±3.0E-2	0.4471±1.3E-2	0.8650±1.5E-3	0.4151±4.7E-3
GC-MC	0.0944±1.6E-4	0.0644±7.5E-4	0.1284±9.2E-5	0.0950±6.1E-4	**0.7931±2.7E-2**	0.4536±2.0E-2	0.8385±8.2E-3	0.4291±5.2E-3
NGCF	0.0979±8.0E-5	0.0672±1.7E-4	0.1337±3.7E-4	0.0987±3.3E-4	1.0414±2.0E-2	0.7225±1.0E-2	1.0547±1.2E-2	0.6819±1.2E-2
LightGCN	0.0894±6.2E-5	0.0584±5.2E-5	0.1244±1.5E-4	0.0898±8.9E-5	0.8182±3.1E-2	0.3894±7.4E-3	0.8348±9.1E-3	0.3629±4.1E-3
LR-GCCF	0.0916±6.9E-5	0.0600±5.2E-5	0.1266±3.3E-5	0.0916±3.7E-5	0.8725±2.7E-2	0.4471±8.9E-3	0.8685±1.7E-2	0.4002±8.9E-3
DGCN-HN	0.0905±5.9E-5	0.0594±3.9E-5	0.1246±1.0E-4	0.0899±4.2E-5	0.8430±3.2E-2	0.3984±8.4E-3	0.8482±1.0E-2	0.3612±4.0E-3
SGL-ED	0.1089±1.1E-4	0.0726±6.7E-5	0.1594±2.3E-4	0.1181±1.1E-4	0.8479±2.3E-2	0.4020±6.8E-3	0.8461±9.3E-3	0.3610±3.9E-3
LightGCN$_{r\text{-}AdjNorm}$	0.0894±5.7E-5	0.0583±4.6E-5	0.1242±7.5E-5	0.0896±4.6E-5	0.8199±3.3E-2	0.3870±7.8E-3	0.8361±9.2E-3	0.3612±4.1E-3
HMLET	0.0900±4.5E-5	0.0592±7.9E-5	0.1252±1.8E-4	0.0904±8.4E-5	0.8517±3.3E-2	0.4083±9.2E-3	0.8562±1.1E-2	0.3646±4.6E-3
HCCF	0.0992±1.0E-3	0.0642±7.6E-4	0.1335±4.9E-4	0.0931±4.7E-4	0.8551±3.5E-2	0.4287±9.5E-3	0.8695±1.1E-2	0.3859±4.5E-3
GTN	0.0995±3.1E-5	0.0681±2.9E-5	0.1313±1.4E-4	0.0951±7.6E-5	0.8261±3.2E-2	0.3942±8.2E-3	0.8383±9.6E-3	0.3647±4.1E-3
CIGCN	0.0918±1.6E-4	0.0603±1.2E-4	0.1271±9.3E-5	0.0922±7.3E-5	0.8712±3.4E-2	0.4100±9.5E-3	0.8692±1.3E-2	0.3632±3.8E-3
JMP-GCF	0.1330±1.1E-3	0.0903±8.8E-4	0.1611±1.7E-3	0.1121±4.5E-4	0.8255±3.2E-2	0.4036±7.5E-3	0.8455±9.1E-3	0.3762±3.9E-3
LightGCL	0.0929±2.7E-5	0.0613±1.3E-5	0.1284±5.2E-5	0.0932±7.1E-5	0.8343±3.3E-2	0.3933±8.2E-3	0.8451±1.0E-2	0.3609±4.1E-3
GLCPN (Ours)	**0.0876±6.3E-5**	**0.0569±3.6E-5**	**0.1229±9.7E-5**	**0.0882±8.3E-5**	0.8159±3.0E-2	**0.3709±8.9E-3**	**0.8281±8.7E-3**	**0.3340±3.5E-3**

- Tables SIII-SIV report the training times achieved by GLCPN and its peers. It is seen that GLCPN costs less training time than other models in most cases, indicating that the computational efficiency of the proposed GLCPN also significantly outperforms other baselines. These notable results are obtained because GLCPN adopts linear message propagation and emphasizes the weights of trainable parameters during the learning process. For example, on the PPI datasets of million-scale (*T-208964* and *T-227321*) and ten million-scale (*T-9544* and *T-9796*), despite the significant fluctuations in training time for all methods due to differences in data scale, the efficiency of GLCPN remains consistently higher than other approaches.

TABLE SIII
THE TRAINING TIME IN RMSE (SEC.) ACHIEVED BY GLCPN AND ITS PEERS.

Method	*T-208964*	*T-227321*	*T-9544*	*T-9796*	*Astro-Ph*	*Cond-Mat*	*Cond-Mat-2003*	*Cond-Mat-2005*
NeuMF	3784±1328.82	2847±888.49	23639±3281.13	15262±2313.04	3124±1319.35	705±42.48	2485±516.66	3041±843.18
MetaMF	16636±5717.27	21886±775.19	86717±8762.90	100478±34208.66	19±0.28	9±1.63	36±0.28	61±13.49
GC-MC	7264±2205.15	3598±2076.02	111765±25822.30	53694±8577.76	24±3.35	17±2.33	51±4.53	77±12.26
NGCF	2394±232.15	1801±168.76	27007±2315.49	15677±1372.02	180±97.91	7±1.26	14±3.72	21±5.98
LightGCN	6504±91.71	3641±79.94	86409±4060.43	47689±2030.77	291±18.47	39±2.60	140±11.96	270±46.18
LR-GCCF	**812±12.78**	522±12.12	23776±382.51	12475±346.35	56±35.34	16±7.04	24±5.96	36±1.19
DGCN-HN	4109±152.29	2698±41.52	80054±9638.93	29526±807.53	110±2.65	18±1.68	206±269.24	133±18.17
SGL-ED	1909±36.29	1024±17.55	27716±332.19	16827±11.01	60±2.82	15±0.64	45±2.21	77±9.62
LightGCN$_{r\text{-}AdjNorm}$	6386±68.83	3507±47.87	80932±622.02	47631±210.16	299±24.09	38±1.41	126±9.59	213±62.21
HMLET	2706±74.28	1274±3.51	40161±3369.92	15008±148.85	140±13.83	29±2.40	69±6.52	105±13.40
HCCF	2148±458.26	1307±119.87	31986±9000.12	13559±1278.28	98±3.94	36±2.25	69±4.48	92±6.29
GTN	5418±537.70	**218±315.74**	23748±1717.06	15310±697.19	44±3.44	8±0.64	25±2.68	44±8.21
CIGCN	1793±48.35	962±26.48	29599±322.72	14960±598.34	91±3.36	43±1.99	91±3.42	134±3.75
JMP-GCF	6290±2004.81	4118±758.49	64241±20949.85	37383±16126.03	123±8.32	16±1.57	96±3.01	230±29.74
LightGCL	3390±42.10	1900±27.05	41094±218.09	22592±278.36	65±2.76	28±1.30	54±2.22	75±4.14
GLCPN (Ours)	1243±328.76	337±50.30	**18053±805.84**	**10682±1625.76**	**10±0.66**	**5±0.36**	**14±0.32**	**19±1.52**

TABLE SIV
THE TRAINING TIME IN MAE (SEC.) ACHIEVED BY GLCPN AND ITS PEERS.

Method	*T-208964*	*T-227321*	*T-9544*	*T-9796*	*Astro-Ph*	*Cond-Mat*	*Cond-Mat-2003*	*Cond-Mat-2005*
NeuMF	3546±1292.37	2993±970.69	23721±3496.87	15111±2262.50	3291±1367.57	717±32.25	2477±515.63	3059±836.57
MetaMF	17226±4821.14	21905±790.42	86846±8947.68	85208±37758.77	571±80.59	11±2.98	57±19.28	75±13.74
GC-MC	2543±242.30	2248±876.67	54936±13745.18	17854±5042.50	32±5.97	23±3.85	64±5.28	97±14.80
NGCF	2242±159.11	1834±123.43	28132±3352.69	14422±1371.31	335±41.57	8±1.95	19±4.85	29±6.76
LightGCN	5763±122.28	3500±78.80	78178±3396.68	43853±1915.68	293±15.35	47±5.34	206±7.42	435±30.94
LR-GCCF	**712±41.90**	491±20.89	20161±818.08	10973±459.04	30±4.12	19±9.76	19±3.09	27±3.61
DGCN-HN	2487±84.92	2433±38.51	102976±10275.42	23993±212.93	95±6.41	21±1.64	198±209.61	182±7.46
SGL-ED	1681±32.21	973±22.12	25686±760.97	15297±485.69	64±2.17	13±0.71	39±3.85	78±6.25
LightGCN$_{r\text{-}AdjNorm}$	5758±92.65	3394±34.17	74130±997.08	43757±289.41	320±11.69	53±3.67	213±13.72	417±20.05
HMLET	2285±71.03	1200±42.06	36989±2171.40	13019±360.84	152±16.69	27±1.92	74±9.68	141±17.84
HCCF	1801±294.71	1317±137.64	25744±5430.47	20815±5353.89	64±4.11	25±0.66	44±2.60	56±2.80
GTN	4944±899.33	2100±256.52	21000±1241.13	13453±556.08	49±2.66	6±0.59	23±2.97	47±7.44
CIGCN	1549±56.24	846±23.25	27687±845.58	12942±614.46	79±1.85	37±1.22	82±3.54	130±1.91
JMP-GCF	1103±625.83	791±335.47	24117±7551.16	12243±3910.36	77±16.68	20±4.30	100±17.31	204±35.47
LightGCL	2982±23.92	1731±23.45	40326±463.23	20638±204.73	59±1.03	21±0.68	39±3.49	65±3.94
GLCPN (Ours)	728±136.53	**440±133.24**	**14200±1727.31**	**8362±1621.56**	**6±0.01**	**3±0.18**	**9±0.40**	**11±0.60**

- Tables SV-SVI report the Friedman test results. Statistically, the comparison results demonstrate that GLCPN significantly outperforms its peers in all four learning tasks regarding HDI data. For instance, as recorded in Tables II-V, the average MAE gains in four tasks are 2.42%, 6.24%, 13.62%, and 8.82%, and the average R^2 gains are 3.62%, 20.83%, 1.60%, and 17.48%. The results of the Friedman test shown in Tables SV-SVI also provide solid support for the superiority of GLCPN. Taking the NDCG as an example (Table SV), the F-rank value of GLCPN is 1.00, which is evidently lower than that of other models, whose best F-rank value is achieved at 3.33. And the Friedman test results in efficiency (Table SVI) are also highly supportive.

TABLE SV
FRIEDMAN TEST RESULTS BASED ON THE RECORDS IN TABLES III-VI.

Method	**F-rank (RMSE)** ↓	**F-rank (MAE)** ↓	**F-rank (R^2)** ↓	**F-rank (NDCG)** ↓
SGL-ED	6.11	5.06	5.44	6.33
LightGCN$_{r\text{-}AdjNorm}$	6.11	5.00	5.89	6.33
HMLET	3.94	3.94	3.89	3.33
HCCF	9.11	9.33	8.89	7.33
GTN	6.11	7.78	6.11	6.89
CIGCN	5.00	4.17	6.44	5.00
JMP-GCF	10.11	10.33	10.00	10.11
LightGCL	8.56	8.17	8.56	9.00
MGDN	4.06	4.78	4.11	3.78
PopGo	5.89	6.44	5.67	6.89
GLCPN (Ours)	**1.00**	**1.00**	**1.00**	**1.00**

TABLE SVI
FRIEDMAN TEST RESULTS BASED ON THE RECORDS IN TABLES SII-SIII.

Method	**F-rank** ↓	**Method**	**F-rank** ↓	**Method**	**F-rank** ↓	**Method**	**F-rank** ↓
NeuMF	12.06	LightGCN	14.19	LightGCN$_{r\text{-}AdjNorm}$	13.56	CIGCN	7.94
MetaMF	11.00	LR-GCCF	2.94	HMLET	8.91	JMP-GCF	9.22
GC-MC	9.90	DGCN-HN	11.34	HCCF	7.75	LightGCL	8.38
NGCF	6.25	SGL-ED	6.16	GTN	5.19	GLCPN (Ours)	**1.22**

B. Results of Ablation Studies (Sec. V.D)

- Table SVII summarizes the supplementary module ablation results in MAE.

TABLE SVII
SUPPLEMENTARY RESULTS (MAE) OF MODULE ABLATIONS.

Method	*T-208964*	*T-227321*	*Astro-Ph*	*Cond-Mat-2005*	*Throughput*	*Kiba*
GLCPN-w/o-L	0.0847±1.3E-4	0.0811±2.0E-4	0.1859±1.2E-3	0.3501±2.7E-3	0.1719±1.5E-3	0.0136±5.8E-5
GLCPN-w/o-P	0.0813±3.2E-5	0.0785±1.7E-5	0.1691±5.2E-4	0.3332±2.8E-4	0.1733±4.2E-4	0.0124±8.5E-5
GLCPN-w/o-H	0.1109±1.1E-4	0.1090±1.6E-4	0.1751±2.0E-4	0.3426±9.3E-4	0.2843±1.3E-3	0.0283±3.8E-4
GLCPN-w/o-L&P	0.0841±1.2E-4	0.0817±1.3E-4	0.1901±1.7E-3	0.3519±1.8E-3	0.1912±1.5E-3	0.0136±1.8E-4
GLCPN-w/o-L&H	0.1023±4.4E-4	0.0953±1.2E-3	0.1721±6.8E-4	0.3320±1.8E-3	0.1834±4.0E-3	0.0133±5.7E-4
GLCPN-w/o-P&H	0.1194±1.6E-4	0.1230±3.2E-5	0.1935±2.5E-4	0.3637±4.1E-4	0.3378±1.1E-3	0.0282±7.1E-4
GLCPN-w/o-L&P&H	0.1061±9.7E-4	0.0993±3.1E-4	0.1813±1.6E-3	0.3455±3.1E-3	0.2196±2.5E-2	0.0131±3.1E-4
GLCPN	**0.0804**±1.4E-5	**0.0780**±5.7E-5	**0.1625**±6.5E-4	**0.3246**±4.5E-4	**0.1639**±4.2E-4	**0.0123**±7.5E-5

- Tables SVIII-SIX summarize the supplementary ablation results regarding linear message propagation, in which both the training and validation errors are recorded.

TABLE SVIII
SUPPLEMENTARY ABLATION RESULTS (RMSE) REGARDING LINEAR MESSAGE PROPAGATION.

Dataset		*T-227321*	*T-9544*	*T-9796*	*Astro-Ph*	*Cond-Mat*	*Cond-Mat-2003*
Training RMSE	GLCPN-MF	**0.0695±5.8E-3**	**0.0627±3.4E-5**	0.0990±5.6E-5	**0.2180±2.2E-2**	**0.3729±3.7E-2**	**0.4610±2.3E-2**
	GLCPN-A	0.0774±2.2E-4	0.0713±5.1E-5	0.0957±6.5E-4	0.2691±1.4E-2	0.6571±2.9E-2	0.7264±1.4E-2
	GLCPN-T	0.0931±2.6E-3	0.0692±1.5E-3	0.1022±1.8E-3	0.4432±3.0E-2	0.6819±3.1E-2	0.8024±1.0E-2
	GLCPN-A&T	0.0791±1.7E-3	0.0654±9.1E-4	0.1007±1.3E-3	0.4022±1.3E-2	0.6534±4.3E-2	0.8138±1.1E-2
	GLCPN	0.0758±3.3E-4	0.0711±5.0E-5	**0.0951±4.9E-4**	0.2615±1.9E-2	0.5423±1.9E-2	0.5958±2.8E-2
Validation RMSE	GLCPN-MF	0.1174±2.4E-4	0.0865±6.1E-5	0.1232±1.5E-4	0.4585±8.4E-3	0.8622±3.1E-2	0.8902±5.6E-2
	GLCPN-A	0.1118±4.6E-4	0.0835±6.1E-5	0.1195±7.7E-5	0.4393±9.5E-3	0.7640±2.6E-2	0.8466±5.0E-2
	GLCPN-T	0.1233±4.8E-4	0.0883±4.8E-4	0.1247±3.0E-4	0.4509±9.4E-3	0.7604±2.3E-2	0.8419±4.9E-2
	GLCPN-A&T	0.1143±3.4E-4	0.0840±4.8E-4	0.1210±1.4E-4	0.4470±1.0E-2	**0.7601±2.2E-2**	0.8381±5.1E-2
	GLCPN	**0.1112±4.3E-4**	**0.0831±5.0E-5**	**0.1195±3.9E-5**	**0.4273±8.1E-3**	0.7674±2.9E-2	**0.8272±5.2E-2**

TABLE SIX
SUPPLEMENTARY ABLATION RESULTS (MAE) REGARDING LINEAR MESSAGE PROPAGATION.

Dataset		*T-208964*	*T-227321*	*T-9544*	*T-9796*	*Astro-Ph*	*Cond-Mat*	*Cond-Mat-2003*	*Cond-Mat-2005*
Training MAE	GLCPN-MF	**0.0427±1.8E-4**	0.0730±3.7E-4	**0.0535±4.7E-4**	**0.0482±7.3E-4**	0.0821±6.1E-3	**0.1266±7.0E-3**	**0.1623±7.4E-3**	**0.1715±1.3E-3**
	GLCPN-A	0.0460±7.4E-5	**0.0694±6.5E-4**	0.0603±4.1E-4	0.0525±3.3E-4	**0.0748±1.2E-3**	0.3253±6.2E-3	0.3327±6.0E-3	0.3394±6.5E-3
	GLCPN-T	0.0471±1.8E-3	0.0755±1.5E-3	0.0726±1.7E-2	0.0659±1.7E-3	0.1693±8.2E-3	0.3327±1.5E-2	0.3458±2.6E-2	0.3355±2.1E-2
	GLCPN-A&T	0.0445±1.1E-4	0.0732±2.1E-4	0.0543±7.1E-4	0.0556±1.6E-3	0.1553±8.3E-3	0.3173±1.0E-2	0.3235±3.1E-2	0.3146±2.6E-2
	GLCPN	0.0456±7.9E-5	0.0699±8.3E-4	0.0593±2.5E-4	0.0512±1.1E-4	0.0886±4.4E-3	0.2447±1.6E-2	0.1999±1.3E-2	0.2098±1.7E-2
Validation MAE	GLCPN-MF	0.0563±3.2E-5	0.0891±1.2E-4	0.0848±1.4E-4	0.0780±5.6E-3	0.1726±7.9E-4	0.4262±6.8E-3	0.3626±1.1E-2	0.3400±6.6E-3
	GLCPN-A	0.0532±7.6E-5	0.0858±5.6E-5	0.0771±3.3E-4	0.0755±1.7E-4	0.1688±1.5E-3	0.3812±3.8E-3	0.3721±7.7E-3	0.3608±6.5E-3
	GLCPN-T	0.0576±6.4E-4	0.0902±1.3E-4	0.0923±8.2E-3	0.0859±3.5E-4	0.1911±1.9E-3	0.3836±4.1E-3	0.3655±6.9E-3	0.3642±9.1E-3
	GLCPN-A&T	0.0543±7.2E-5	0.0867±2.3E-4	0.0802±1.8E-4	0.0786±2.0E-4	0.1846±3.9E-3	0.3799±3.6E-3	0.3594±7.6E-3	0.3467±9.0E-3
	GLCPN	**0.0527±6.0E-5**	**0.0857±3.0E-5**	**0.0763±2.7E-4**	**0.0749±2.2E-4**	**0.1595±1.6E-4**	**0.3686±5.3E-3**	**0.3422±9.1E-3**	**0.3286±5.5E-3**

- Tables SX-SXI record the supplementary ablation results regarding priori convolution operator, including validation errors, training epochs, time cost per epoch, and total time cost.

TABLE SX
SUPPLEMENTARY ABLATION RESULTS (RMSE) REGARDING PRIORI CONVOLUTION OPERATOR.

Dataset		*T-208964*	*T-227321*	*T-9544*	*T-9796*	*Astro-Ph*	*Cond-Mat*	*Cond-Mat-2003*
RMSE	GLCPN-B	0.1135±9.5E-5	0.1392±8.6E-4	0.1142±3.8E-4	0.1125±3.4E-4	0.4325±9.0E-3	0.7734±2.7E-2	0.8376±5.2E-2
	GLCPN-SA	**0.1112±1.8E-4**	0.1370±3.3E-3	0.1266±7.7E-4	0.1235±5.9E-4	0.4328±9.4E-3	0.7782±3.0E-2	0.8481±5.1E-2
	GLCPN	0.1115±9.9E-5	**0.1363±3.0E-3**	**0.1119±3.6E-4**	**0.1112±4.3E-4**	**0.4273±8.1E-3**	**0.7674±2.9E-2**	**0.8272±5.2E-2**
Epochs	GLCPN-B	71±30.16	**85±12.76**	115±10.34	**100±6.26**	17±0.98	39±4.77	21±2.73
	GLCPN-SA	88±26.53	128±21.67	123±3.67	114±9.22	17±1.17	32±6.36	12±0.75
	GLCPN	**71±30.16**	86±14.12	**104±10.35**	114±19.96	**12±1.36**	**28±1.96**	**12±1.94**
Time Cost Per Epoch	GLCPN-B	35±0.08	22±0.32	28±0.05	15±0.07	0.96±0.01	0.24±0.00	1.09±0.00
	GLCPN-SA	183±0.59	109±0.21	107±0.17	55±0.75	3.54±0.05	0.97±0.01	3.56±0.06
	GLCPN	**35±0.40**	**22±0.28**	**28±0.13**	**15±0.09**	**0.96±0.01**	**0.24±0.00**	**1.09±0.01**
Total Time Cost	GLCPN-B	2531±1074.25	1891±296.56	3264±289.51	**1472±90.81**	16±1.10	9±1.12	22±2.92
	GLCPN-SA	16102±4888.56	13992±2354.94	13103±373.16	6351±553.40	59±3.90	31±0.04	42±2.85
	GLCPN	**2497±1060.24**	**1880±292.48**	**2966±302.48**	1681±303.08	**11±1.35**	**7±0.43**	**13±2.15**

TABLE SXI
SUPPLEMENTARY ABLATION RESULTS (MAE) REGARDING PRIORI CONVOLUTION OPERATOR.

Dataset		*T-208964*	*T-227321*	*T-9544*	*T-9796*	*Astro-Ph*	*Cond-Mat*	*Cond-Mat-2003*	*Cond-Mat-2005*
MAE	GLCPN-B	0.0736±5.4E-5	0.0987±5.6E-4	0.0780±2.4E-4	0.0761±1.5E-4	0.1667±4.8E-4	0.3759±5.1E-3	0.3536±8.2E-3	0.3398±6.1E-3
	GLCPN-SA	0.0738±1.5E-4	0.0991±2.3E-3	0.0892±5.0E-4	0.0854±4.6E-4	0.1637±9.5E-4	0.3782±5.6E-3	0.3651±9.0E_3	0.3508±5.7E-3
	GLCPN	**0.0721±6.7E-5**	**0.0971±1.3E-3**	**0.0763±2.7E-4**	**0.0749±2.2E-4**	**0.1595±1.6E-4**	**0.3686±5.3E-3**	**0.3422±9.1E-3**	**0.3286±5.5E-3**
Epochs	GLCPN-B	54±8.15	52±12.60	**71±13.16**	**89±12.19**	12±1.33	60±8.57	31±1.26	19±1.96
	GLCPN-SA	**42±12.58**	**37±10.79**	107±8.14	107±10.57	15±0.89	37±3.08	**12±1.10**	**9±3.20**
	GLCPN	50±12.09	56±9.68	75±12.27	146±15.77	**11±0.75**	**27±6.31**	20±3.01	12±2.04
Time Cost Per Epoch	GLCPN-B	35±0.09	22±0.33	29±0.09	15±0.08	0.96±0.01	0.24±0.00	1.09±0.00	2.13±0.01
	GLCPN-SA	183±0.61	109±0.56	107±0.10	56±0.76	3.55±0.05	0.97±0.01	3.56±0.07	6.85±0.18
	GLCPN	**35±0.40**	**22±0.32**	**29±0.19**	**15±0.08**	**0.96±0.01**	**0.24±0.00**	**1.09±0.01**	**2.13±0.01**
Total Time Cost	GLCPN-B	1923±287.08	**1172±295.15**	**2018±369.88**	**1312±179.24**	12±1.19	15±2.21	34±1.28	40±4.12
	GLCPN-SA	7626±2289.23	4053±1191.31	11493±874.54	5932±611.35	53±2.80	36±0.03	43±3.22	65±22.74
	GLCPN	**1744±412.10**	1242±198.53	2140±340.74	2137±221.26	**10±0.75**	**6±1.47**	**22±3.36**	**25±4.41**

- Tables SXII-SXIII record the supplementary ablation results regarding locality-enhanced pooling, including the RMSE and MAE of GLCPN and its variants.

TABLE SXII
SUPPLEMENTARY ABLATION RESULTS (RMSE) REGARDING LOCALITY-ENHANCED POOLING.

Method	*T-9544*	*T-9796*	*Cond-Mat*	*Cond-Mat-2003*
GLCPN-S	0.1331$_{\pm 9.1E-5}$	0.1660$_{\pm 2.4E-4}$	0.7816$_{\pm 2.6E-2}$	0.8445$_{\pm 5.0E-2}$
GLCPN-SS	0.0866$_{\pm 3.4E-5}$	0.1220$_{\pm 2.5E-4}$	0.7780$_{\pm 2.6E-2}$	0.8453$_{\pm 4.8E-2}$
GLCPN-M	0.0845$_{\pm 6.9E-5}$	0.1216$_{\pm 9.7E-5}$	0.7690$_{\pm 2.9E-2}$	0.8276$_{\pm 5.1E-2}$
GLCPN-C	0.0864$_{\pm 8.6E-5}$	0.1230$_{\pm 9.5E-5}$	0.7912$_{\pm 2.9E-2}$	0.8392$_{\pm 5.2E-2}$
GLCPN	**0.0831$_{\pm 5.0E-5}$**	**0.1195$_{\pm 3.9E-5}$**	**0.7674$_{\pm 2.9E-2}$**	**0.8272$_{\pm 5.2E-2}$**

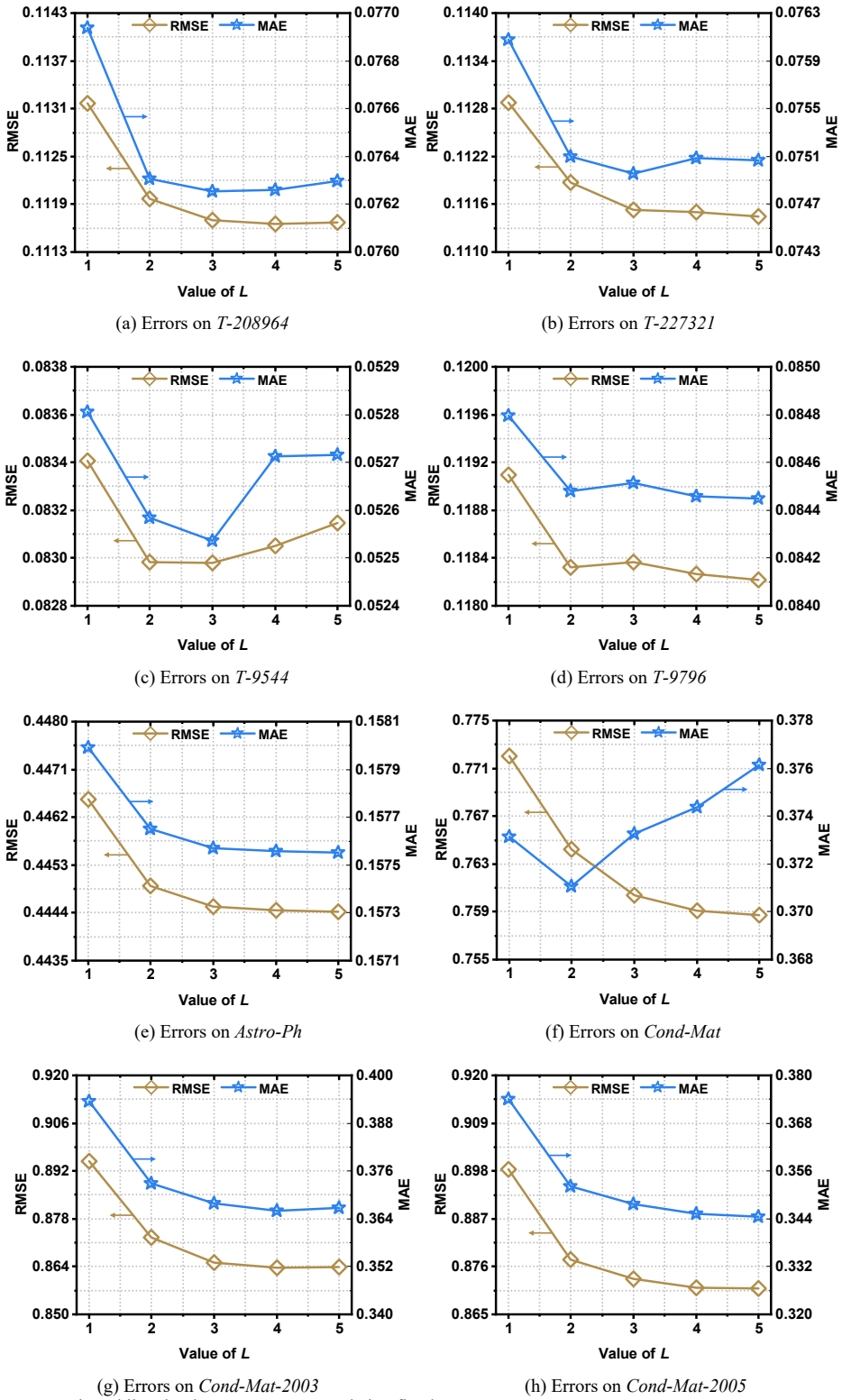
TABLE SXIII
SUPPLEMENTARY ABLATION RESULTS (MAE) REGARDING LOCALITY-ENHANCED POOLING.

Method	*T-208964*	*T-227321*	*T-9544*	*T-9796*	*Astro-Ph*	*Cond-Mat*	*Cond-Mat-2003*	*Cond-Mat-2005*
GLCPN-S	0.0909$_{\pm 3.1E-4}$	0.1187$_{\pm 7.3E-4}$	0.1152$_{\pm 1.9E-3}$	0.1123$_{\pm 5.0E-4}$	0.1812$_{\pm 1.1E-3}$	0.3905$_{\pm 4.8E-3}$	0.3705$_{\pm 8.5E-3}$	0.3594$_{\pm 5.5E-3}$
GLCPN-SS	0.0565$_{\pm 7.2E-5}$	0.0872$_{\pm 7.8E-5}$	0.0821$_{\pm 1.8E-4}$	0.0801$_{\pm 2.4E-4}$	0.1777$_{\pm 1.7E-3}$	0.3844$_{\pm 7.2E-3}$	0.3621$_{\pm 5.6E-3}$	0.3513$_{\pm 1.2E-2}$
GLCPN-M	0.0548$_{\pm 6.3E-5}$	0.0875$_{\pm 7.7E-5}$	0.0794$_{\pm 1.7E-4}$	0.0782$_{\pm 2.0E-4}$	0.1647$_{\pm 6.7E-4}$	0.3694$_{\pm 6.0E-3}$	0.3454$_{\pm 8.9E-3}$	0.3330$_{\pm 5.3E-3}$
GLCPN-C	0.0562$_{\pm 8.0E-5}$	0.0889$_{\pm 8.6E-5}$	0.0853$_{\pm 1.8E-4}$	0.0826$_{\pm 2.6E-4}$	0.1671$_{\pm 1.1E-3}$	0.3820$_{\pm 5.4E-3}$	0.3497$_{\pm 8.8E-3}$	0.3358$_{\pm 5.4E-3}$
GLCPN	**0.0527$_{\pm 6.0E-5}$**	**0.0857$_{\pm 3.0E-5}$**	**0.0763$_{\pm 2.7E-4}$**	**0.0749$_{\pm 2.2E-4}$**	**0.1595$_{\pm 1.6E-4}$**	**0.3686$_{\pm 5.3E-3}$**	**0.3422$_{\pm 9.1E-3}$**	**0.3286$_{\pm 5.5E-3}$**

V. Supplementary Figures of Empirical Studies (Sec. V.C)

The hyperparameter sensitivity test results have been drawn in Figs. S2-S7, which are discussed in Sec. V.C. Their details are described as follows:

- Figs. S2-S3 plot the errors and epochs of GLCPN as L varies;
- Figs. S4-S5 plot the errors and epochs of GLCPN as K varies;
- Figs. S6-S7 plot the errors and epochs of GLCPN as α varies.



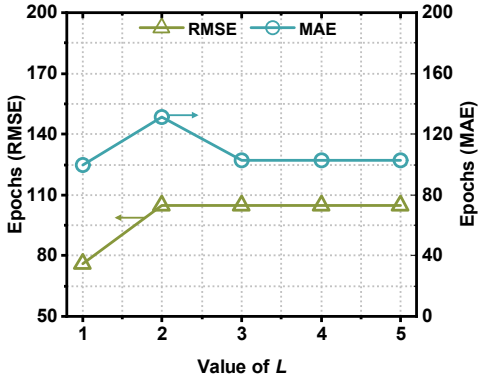
(a) Errors on *T-208964* (b) Errors on *T-227321*

(c) Errors on *T-9544* (d) Errors on *T-9796*

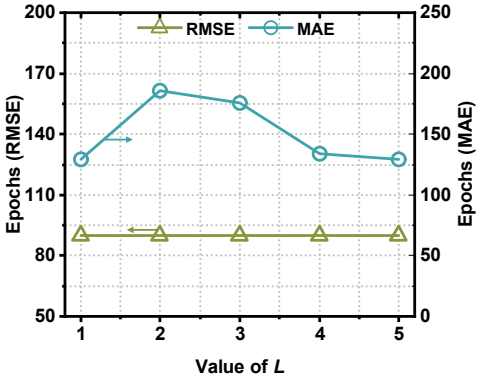
(e) Errors on *Astro-Ph* (f) Errors on *Cond-Mat*

(g) Errors on *Cond-Mat-2003* (h) Errors on *Cond-Mat-2005*

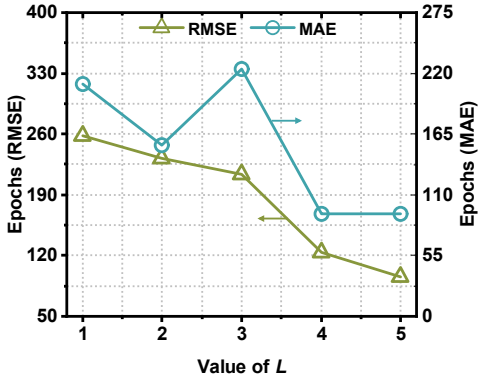
Fig. S2. Errors of GLCPN as L varies while other hyperparameters are being fixed.



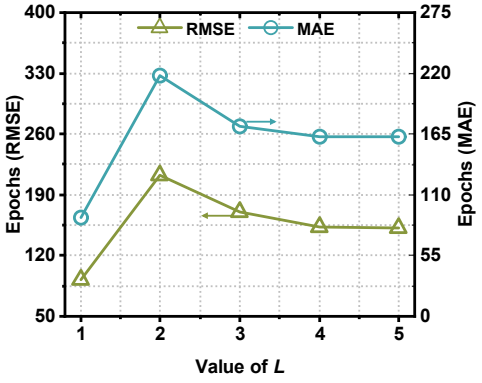
(a) Epochs on *T-208964*



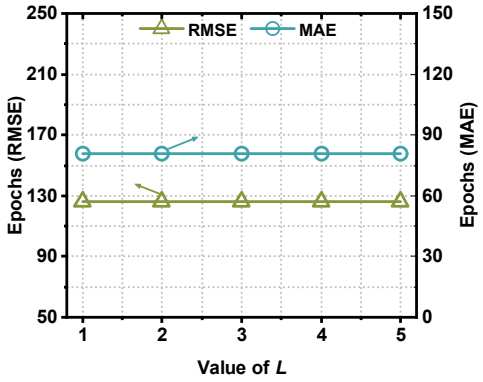
(b) Epochs on *T-227321*



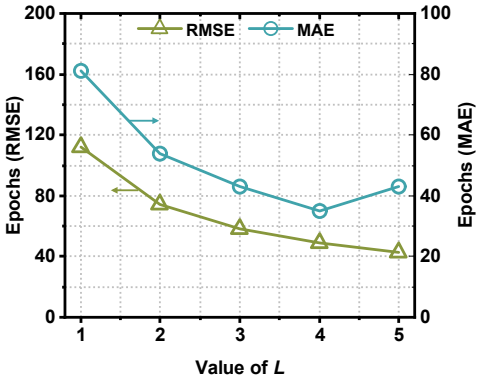
(c) Epochs on *T-9544*



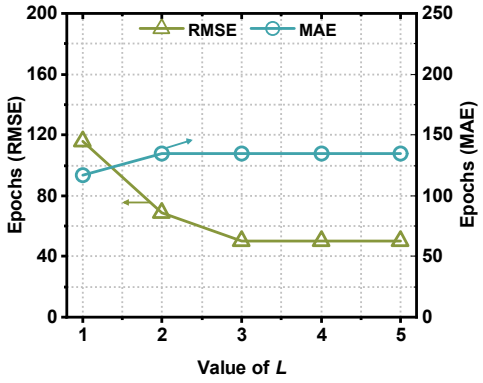
(d) Epochs on *T-9796*



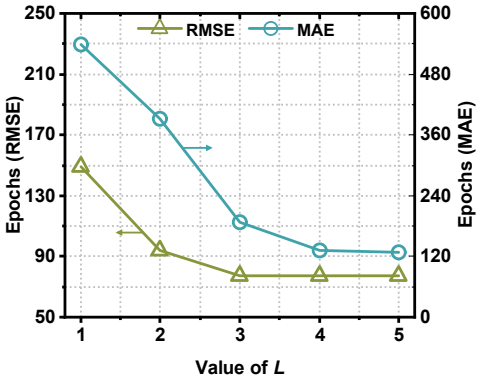
(e) Epochs on *Astro-Ph*



(f) Epochs on *Cond-Mat*

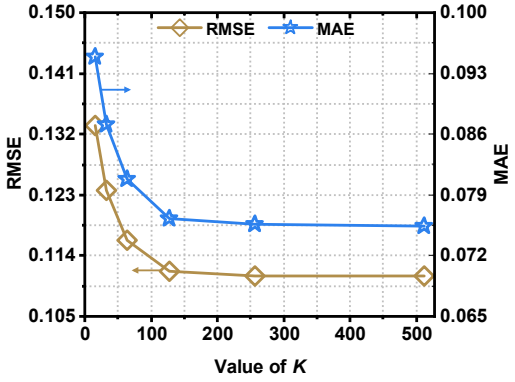


(g) Epochs on *Cond-Mat-2003*

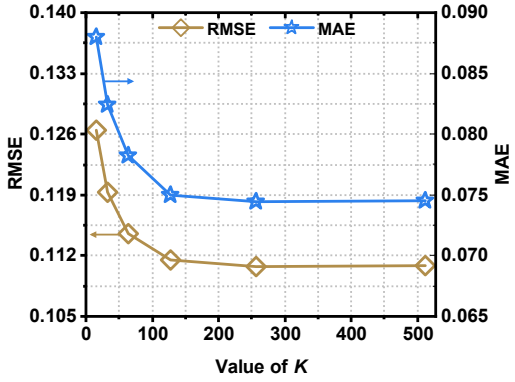


(h) Epochs on *Cond-Mat-2005*

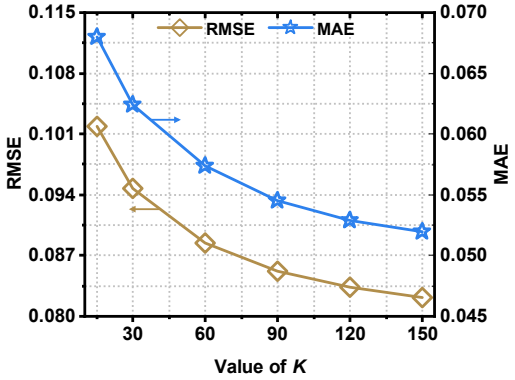
Fig. S3. Training epochs of GLCPN as L varies while other hyperparameters are being fixed.



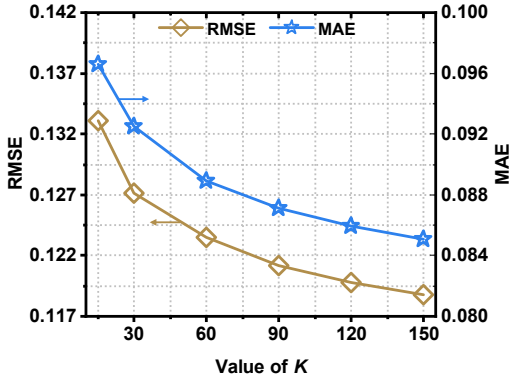
(a) Errors on *T-208964*



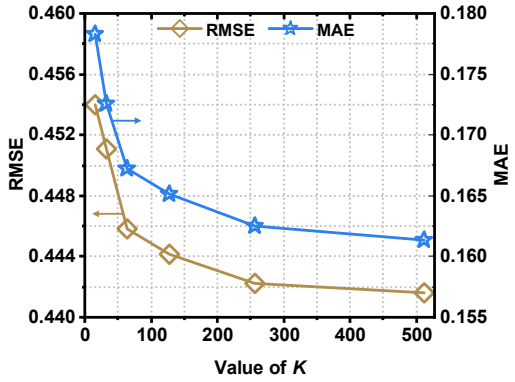
(b) Errors on *T-227321*



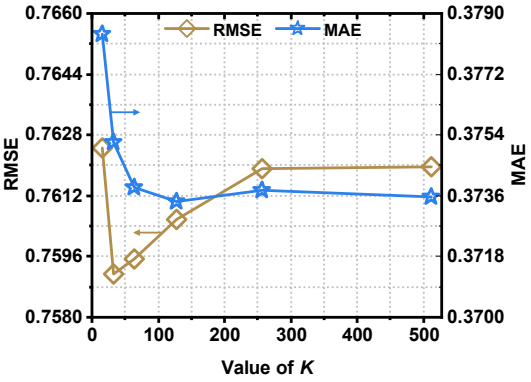
(c) Errors on *T-9544*



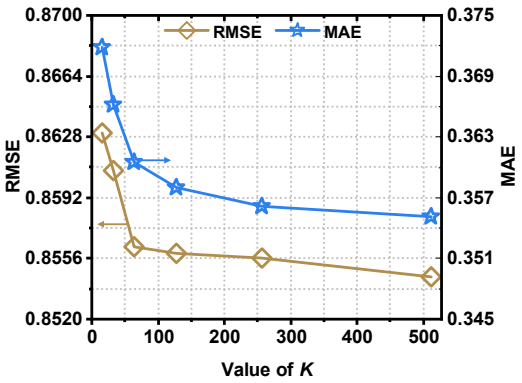
(d) Errors on *T-9796*



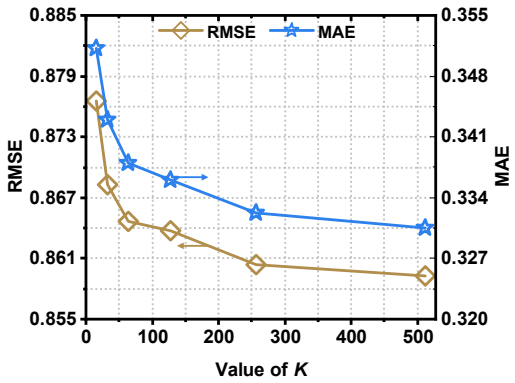
(e) Errors on *Astro-Ph*



(f) Errors on *Cond-Mat*

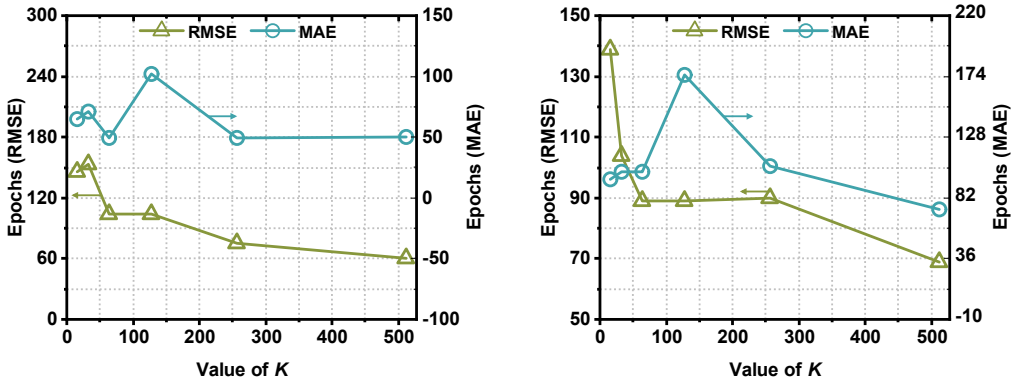


(g) Errors on *Cond-Mat-2003*



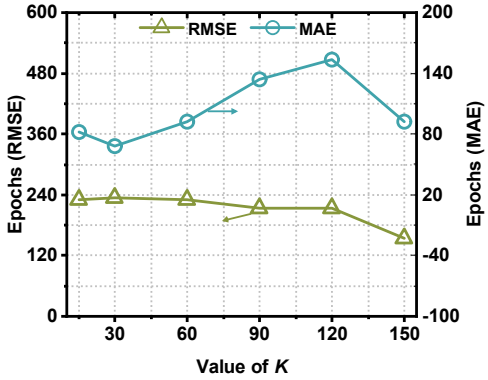
(h) Errors on *Cond-Mat-2005*

Fig. S4. Errors of GLCPN as K varies while other hyperparameters are being fixed.

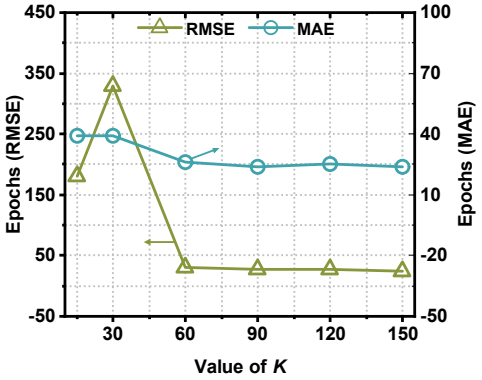


(a) Epochs on *T-208964*

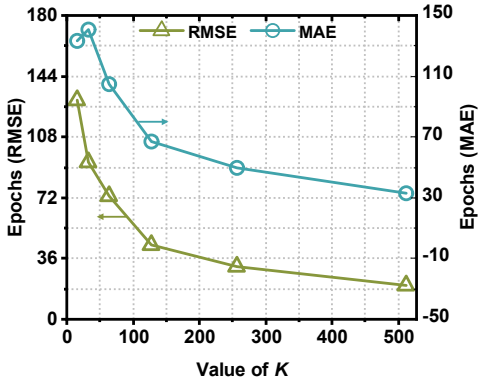
(b) Epochs on *T-227321*



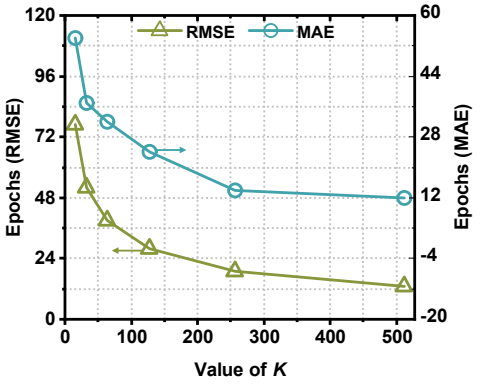
(c) Epochs on *T-9544*



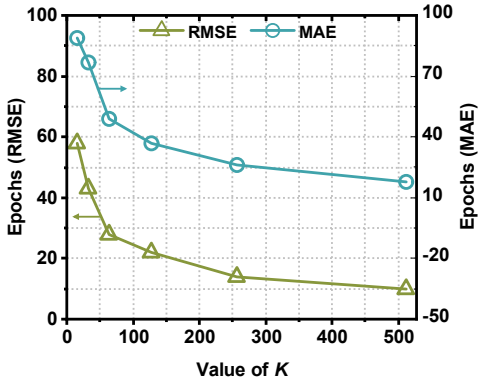
(d) Epochs on *T-9796*



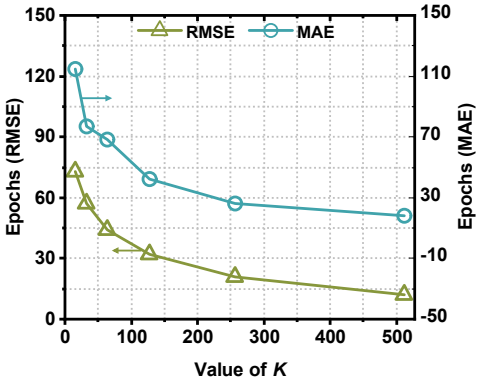
(e) Epochs on *Astro-Ph*



(f) Epochs on *Cond-Mat*

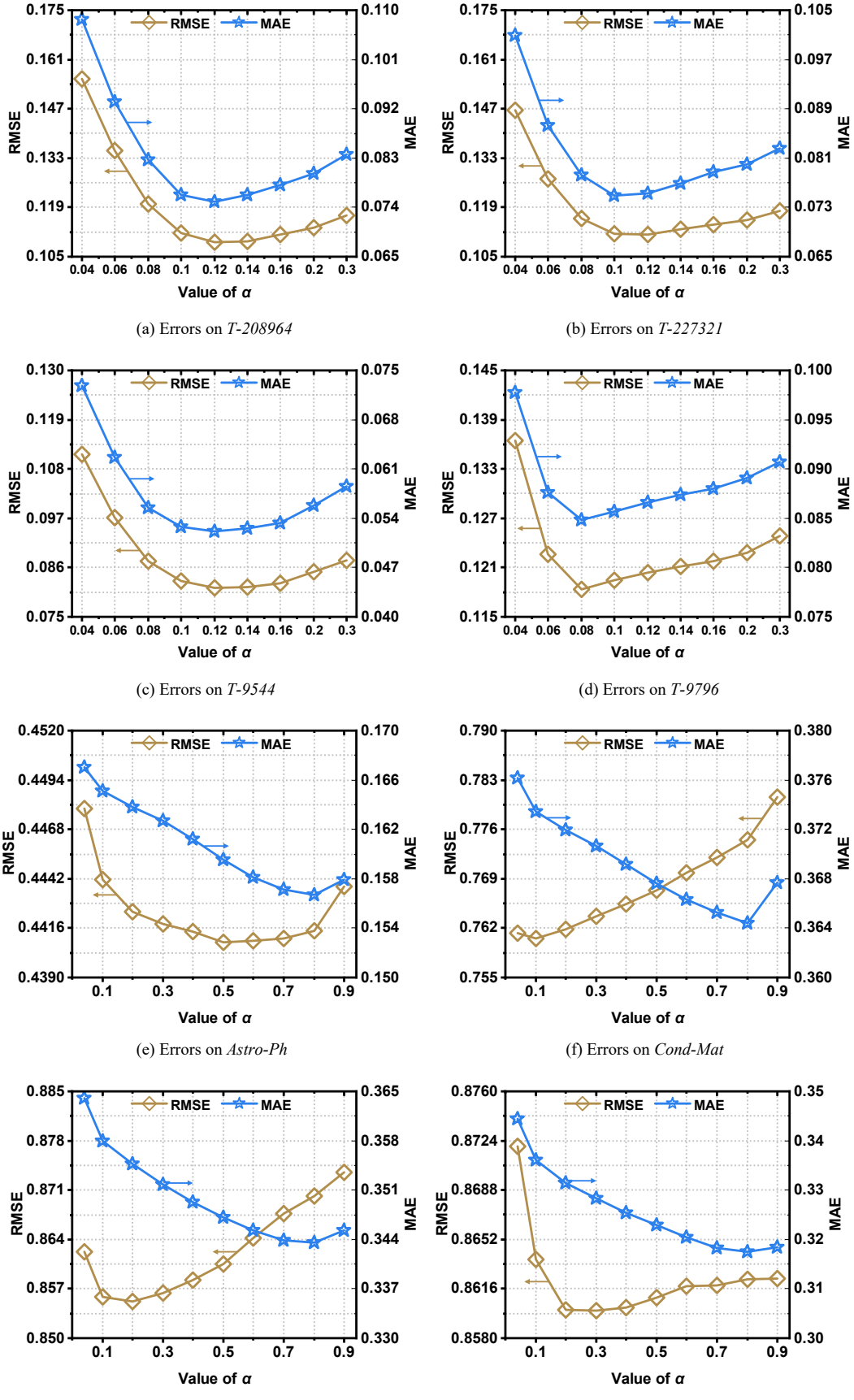


(g) Epochs on *Cond-Mat-2003*



(h) Epochs on *Cond-Mat-2005*

Fig. S5. Training epochs of GLCPN as K varies while other hyperparameters are being fixed.



(a) Errors on *T-208964*

(b) Errors on *T-227321*

(c) Errors on *T-9544*

(d) Errors on *T-9796*

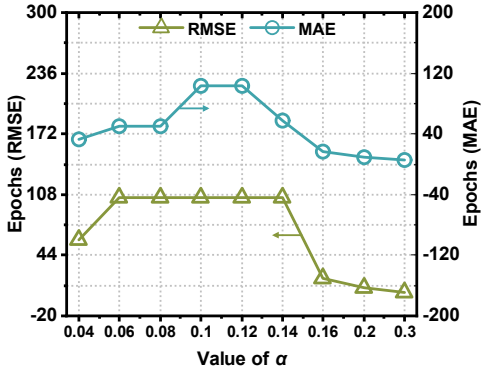
(e) Errors on *Astro-Ph*

(f) Errors on *Cond-Mat*

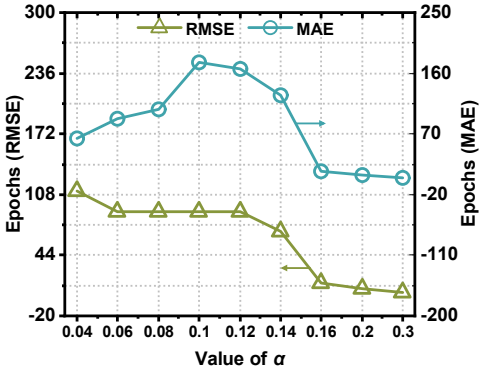
(g) Errors on *Cond-Mat-2003*

(h) Errors on *Cond-Mat-2005*

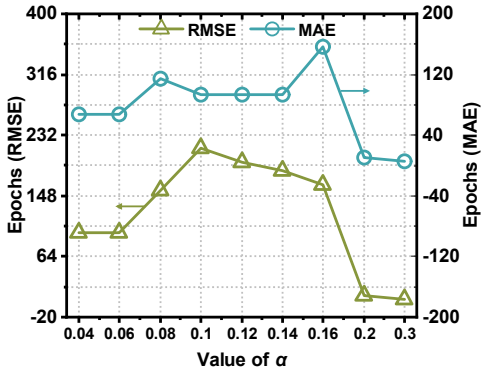
Fig. S6. Errors of GLCPN as α varies while other hyperparameters are being fixed.



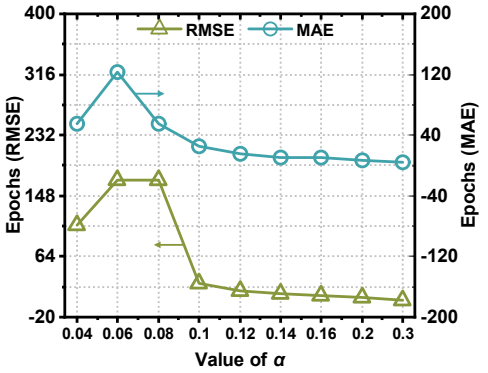
(a) Epochs on *T-208964*



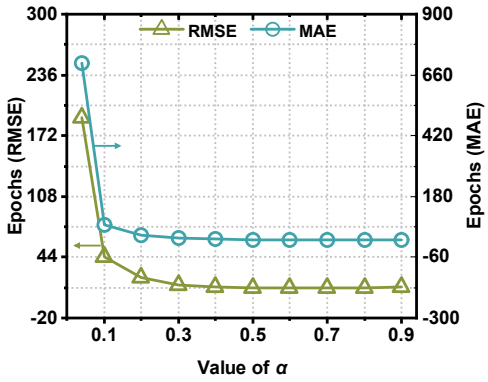
(b) Epochs on *T-227321*



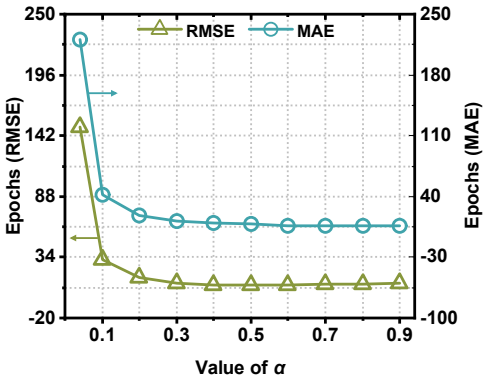
(c) Epochs on *T-9544*



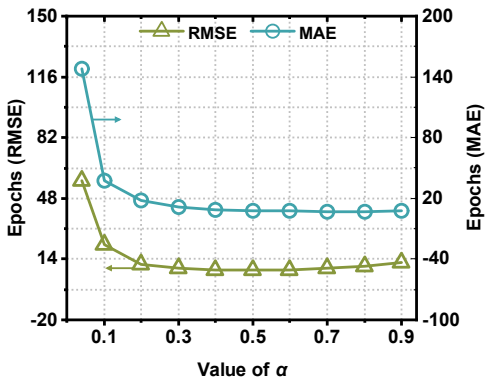
(d) Epochs on *T-9796*



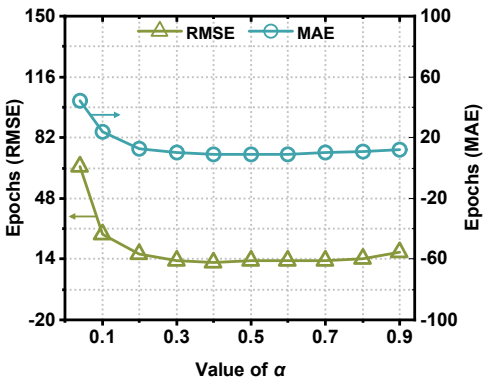
(e) Epochs on *Astro-Ph*



(f) Epochs on *Cond-Mat*



(g) Epochs on *Cond-Mat-2003*



(h) Epochs on *Cond-Mat-2005*

Fig. S7. Training epochs of GLCPN as α varies while other hyperparameters are being fixed.