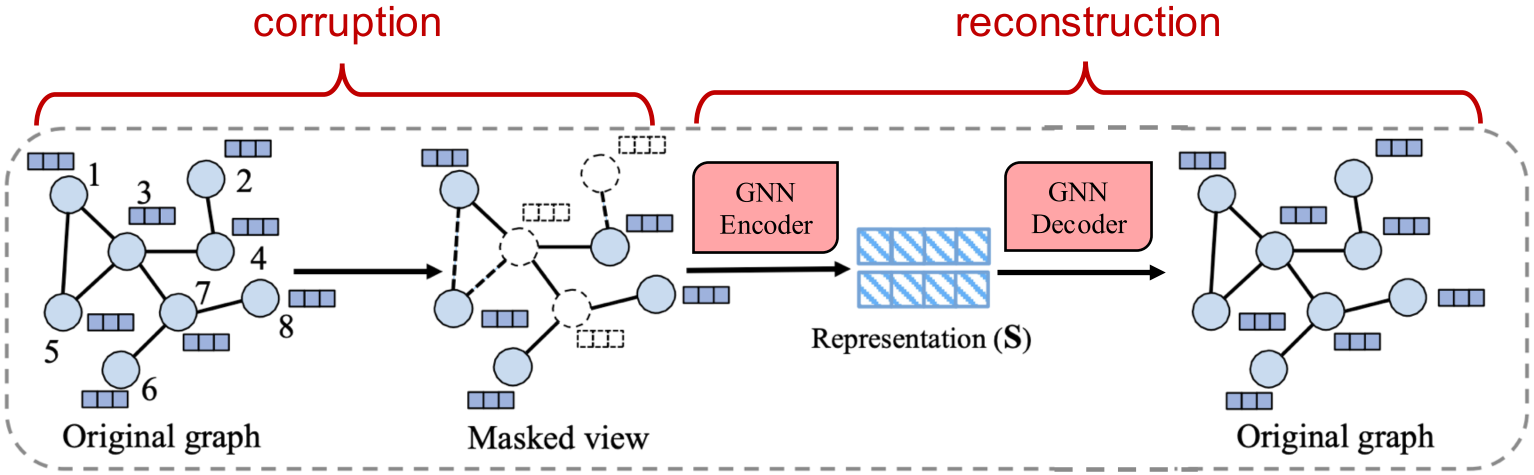
Graph Positional Autoencoders as Self-supervised Learners

Yang Liu*, Deyu Bo*, Wenxuan Cao, Yuan Fang, Yawen Li, Chuan Shi†

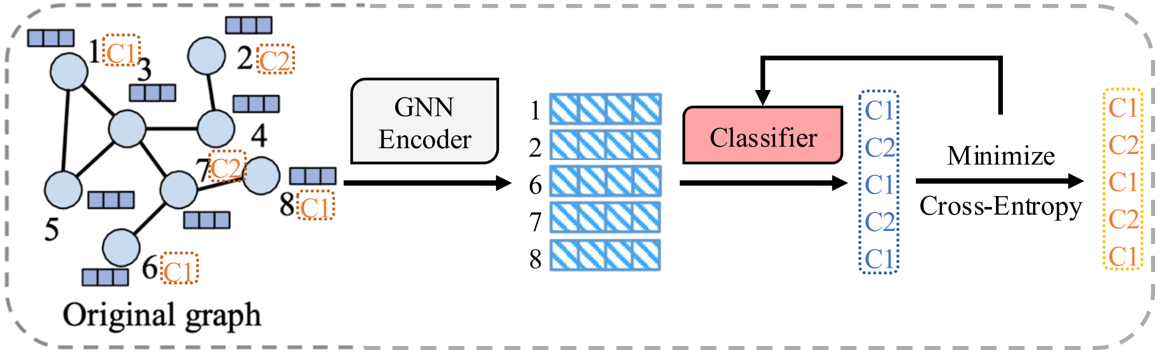
* Both authors contributed equally to this research. † Corresponding author.

Masked Graph Autoencoders

Maked Graph Autoencoders (Masked GAEs) follow a corruption-reconstruction framework, which learns graph representations by recovering the missing information of the incomplete input graphs.



Pre-training of Maked Graph Autoencoders

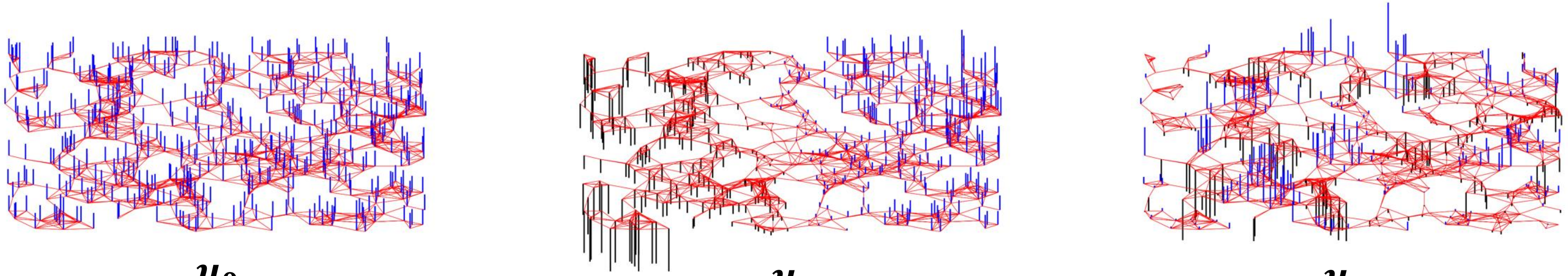


Fine-tuning for downstream task (e.g. node classification)

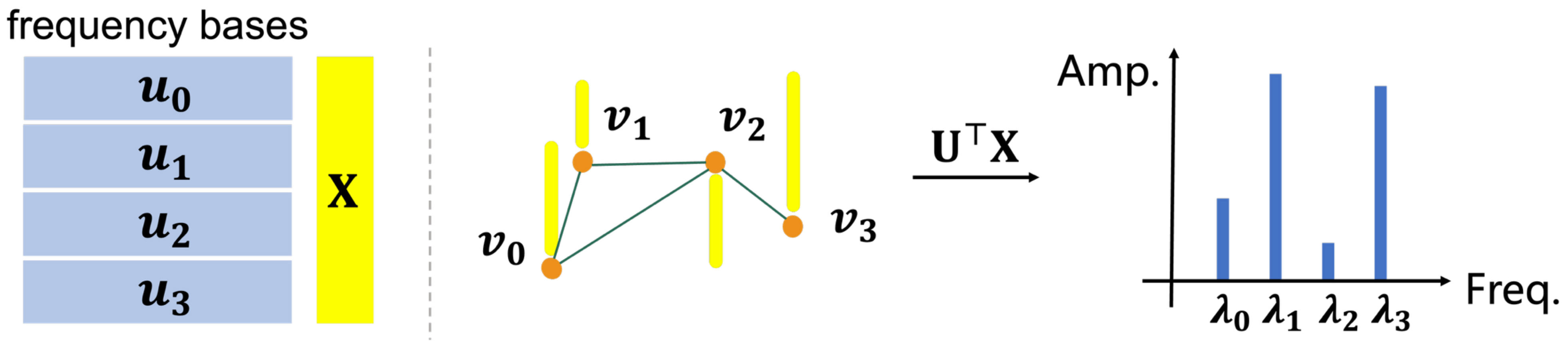
Frequency Bases in Graphs

- Eigenvectors of the graph Laplacian represent different frequencies, acting as frequency bases in the spectral domain.

Given eigen-decomposition $Lu_k = \lambda_k u_k$ with $u_k^\top u_k = 1$, we have $u_k^\top L u_k = u_k^\top \lambda_k u_k = \lambda_k$.
Since $u_k^\top L u_k = \sum_{(i,j)\in\mathcal{E}} \left(u_{i,k} - u_{j,k}\right)^2$, we obtain $\sum_{(i,j)\in\mathcal{E}} \left(u_{i,k} - u_{j,k}\right)^2 = \lambda_k$.
Therefore, λ_k reflects the frequency magnitude of u_k over the graph.



$\boldsymbol{u}_0$ $\boldsymbol{u}_1$ $\boldsymbol{u}_{50}$



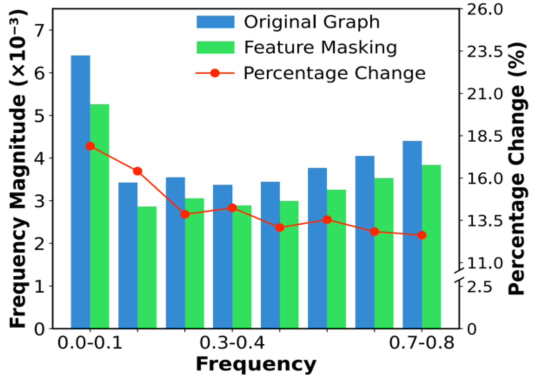
2 Motivation

- Limitations of existing methods:

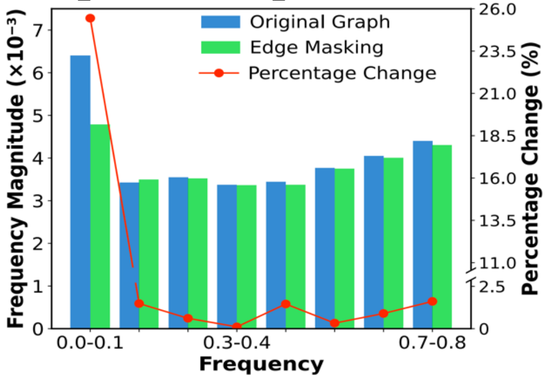
  Existing masked GAEs tend to focus on reconstructing low-frequency information of graphs while overlooking high-frequency information.

Model	Corruption			Reconstruction		
	Feature	Edge	Position	Feature	Edge	Other
GraphMAE [18]	✓	-	-	✓	-	-
StructMAE [32]	✓	-	-	✓	-	-
AUG-MAE [59]	✓	-	-	✓	-	-
S2GAE [50]	-	✓	-	-	✓	-
SeeGera [30]	✓	✓	-	✓	✓	-
Bandana [77]	-	✓	-	-	✓	-
MaskGAE [28]	-	✓	-	✓	-	Degree
GiGaMAE [45]	✓	✓	-	-	-	Latent
GraphPAE	✓	-	✓	✓	-	Position

Node Masking: $\mathbf{U}^\top\mathbf{X}$ vs $\mathbf{U}^\top\mathbf{X}'$



Edge Masking: $\mathbf{U}^\top\mathbf{X}$ vs $\mathbf{U}'^\top\mathbf{X}$

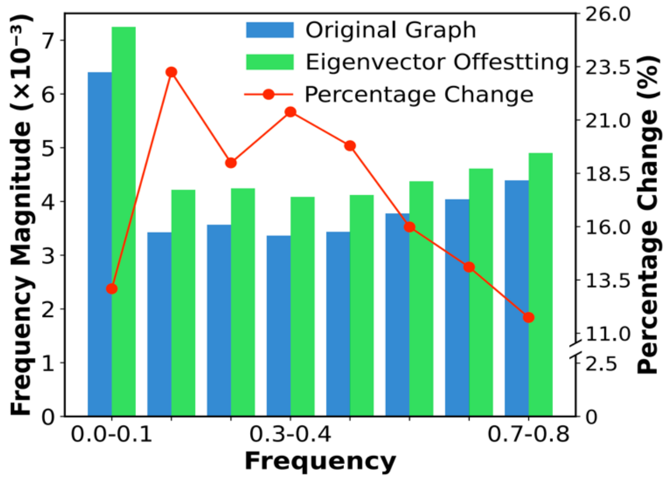


- **How can we design the corruption and reconstruction objectives to exploit the diverse frequency information?**

3 Inspired by Spectral Theory

- The eigenvectors of graph Laplacian correspond to different frequencies in the graph signal processing.
- Directly perturbing eigenvectors can explicitly corrupt corresponding frequency information in the graph.

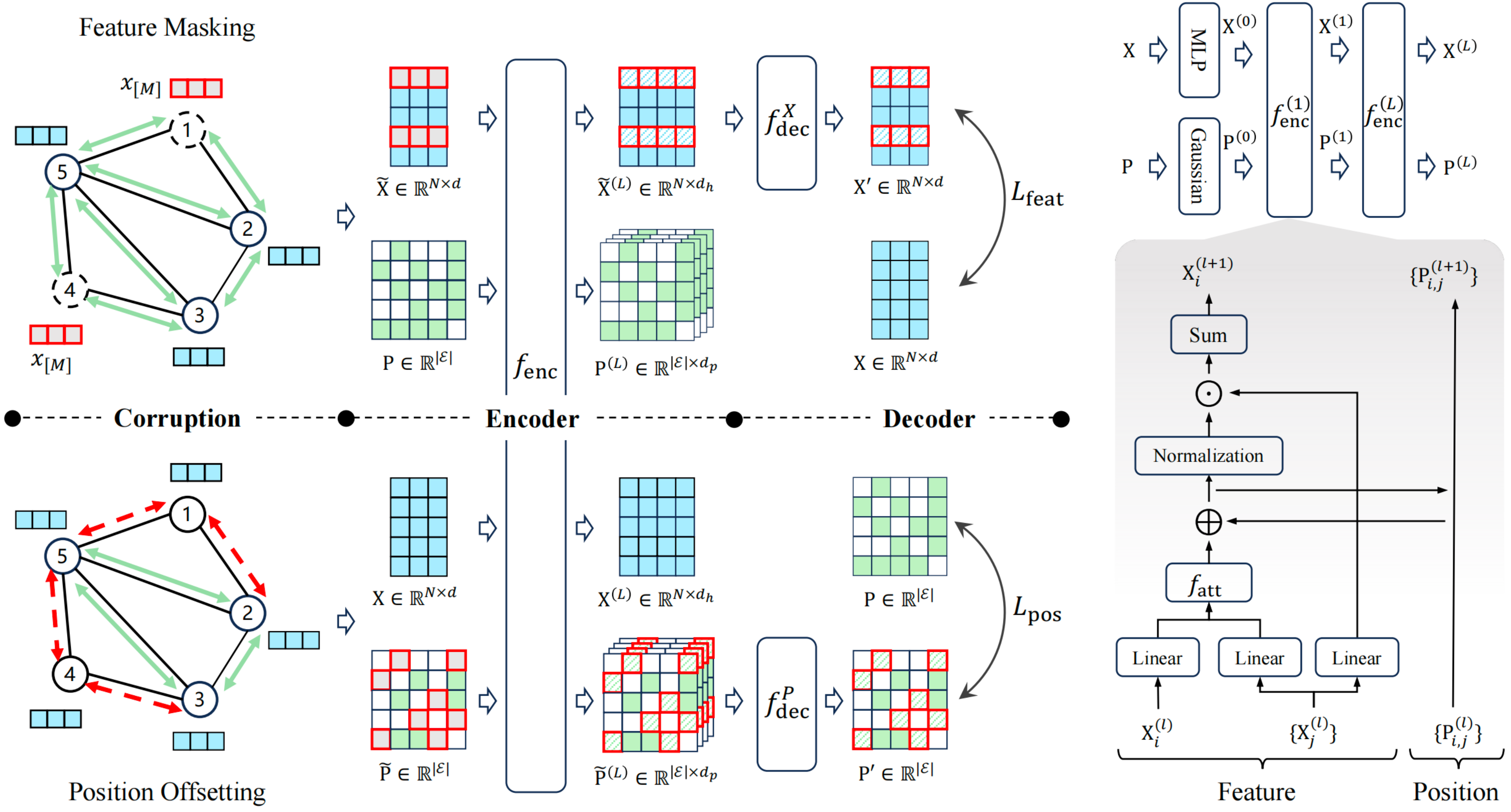
Eigenvector offsetting: $\mathbf{U}^{\top}\mathbf{X}$ vs $\mathbf{U}'^{\top}\mathbf{X}$



Incorporating eigenvector corruption-reconstruction into masked GAEs.

- Two main challenge of reconstructing the eigenvectors:
  - ✓ GNN Expressivity: Eigenvectors represent global structural patterns, which cannot be easily approximated by GNNs, whose expressiveness is bounded by the 1-WL test.
  - ✓ Eigenvector Ambiguity: Eigenvectors suffer from sign- and basis- ambiguity issues. Directly reconstructing eigenvectors leads to non-unique solutions, thereby affecting the robustness of GAEs.

4 GraphPAE Overall Framework



(a) Overall pipeline of GraphPAE.

(b) Encoder of GraphPAE

- Feature Masking.

Randomly sampling a subset of nodes $\widetilde{\mathcal{V}} \subset \mathcal{V}$ and reset their features. The corrupted feature matrix $\widetilde{\mathbf{X}}$ is defined as:

$$\widetilde{\mathbf{X}}_i = \begin{cases} \mathbf{x}_{[M]}, & \text{if } v_i \in \widetilde{\mathcal{V}} \\ \mathbf{X}_i, & \text{if } v_i \notin \widetilde{\mathcal{V}} \end{cases}$$

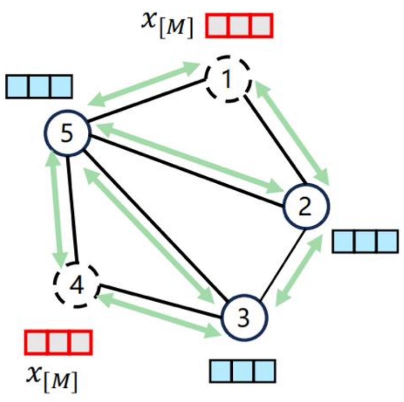
- Position Offsetting.

Adding random offsets to node position:

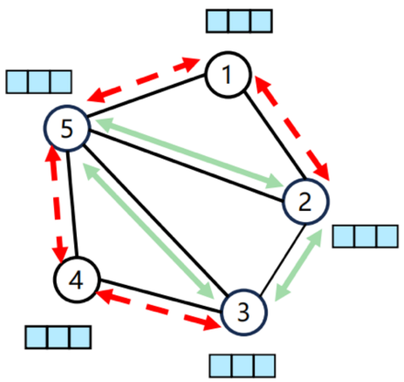
$$\widetilde{\mathbf{U}}_i = \begin{cases} \mathbf{U}_i + \delta, & \text{if } v_i \in \widetilde{\mathcal{V}} \\ \mathbf{U}_i, & \text{if } v_i \notin \widetilde{\mathcal{V}} \end{cases} \qquad \delta \in \mathbb{R}^K \text{ is sampled from } \mathcal{U}(-\mu_p, \mu_p)$$

- Relative Positional Encoding.

$$\mathbf{P}_{i,j} = \begin{cases} \|\mathbf{U}_i - \mathbf{U}_j\|_2, & \text{if } \mathbf{A}_{i,j} = 1 \\ 0, & \text{otherwise} \end{cases}$$



Feature Masking



Position Offsetting

4 GraphPAE Encoder

- Overall formulation.

$$\mathbf{X}_i^{(l+1)}, \mathbf{P}_i^{(l+1)} = f_{\text{enc}}^{(l+1)}\left(\mathbf{X}_i^{(l)}, \left\{\mathbf{X}_j^{(l)}\right\}_{j\in\mathcal{N}_i}, \mathbf{P}_i^{(l)}\right)$$

$$\mathbf{P}_{i,j}^{(0)} = \text{MLP}\left(\left[G(\mathbf{P}_{i,j};\mu_1,\sigma),\cdots,G(\mathbf{P}_{i,j};\mu_d,\sigma)\right]\right) \quad \mathbf{X}_i^{(0)} = \text{MLP}\left(\mathbf{X}_i\right).$$

$$G(\mathbf{P}_{i,j};\mu_k,\sigma) = \exp\left(-\left(\mathbf{P}_{i,j}-\mu_k\right)^2/2\sigma^2\right)$$

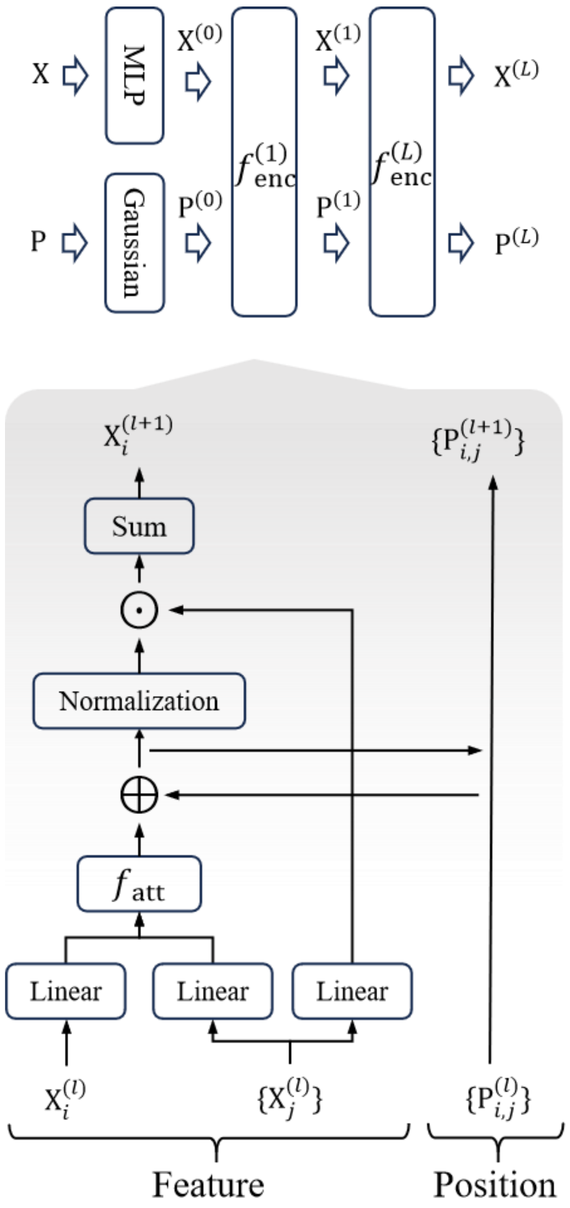
- Feature Path: PE-enhanced MPNNs.

$$\alpha_{i,j}^{(l)} = f_{\text{att}}\left(\mathbf{X}_i^{(l)}, \mathbf{X}_j^{(l)}\right), \quad \alpha_{i,j}^{(l)} \in \mathbb{R}^d,$$

$$\mathbf{X}_i^{(l+1)} = \sum_{j\in\mathcal{N}_i}\left(\alpha_{i,j}^{(l)} + \mathbf{P}_{i,j}^{(l)}\right) \odot \text{MLP}\left(\mathbf{X}_j^{(l)}\right),$$

- Position Path: Refine Node Positions.

$$\mathbf{P}_{i,j}^{(l+1)} = \alpha_{i,j}^{(l)} + \mathbf{P}_{i,j}^{(l)}.$$



- Feature Reconstruction.

$$\mathbf{X}_i' = f_{\text{dec}}^X\left(\widetilde{\mathbf{X}}_i^{(L)}\right),$$

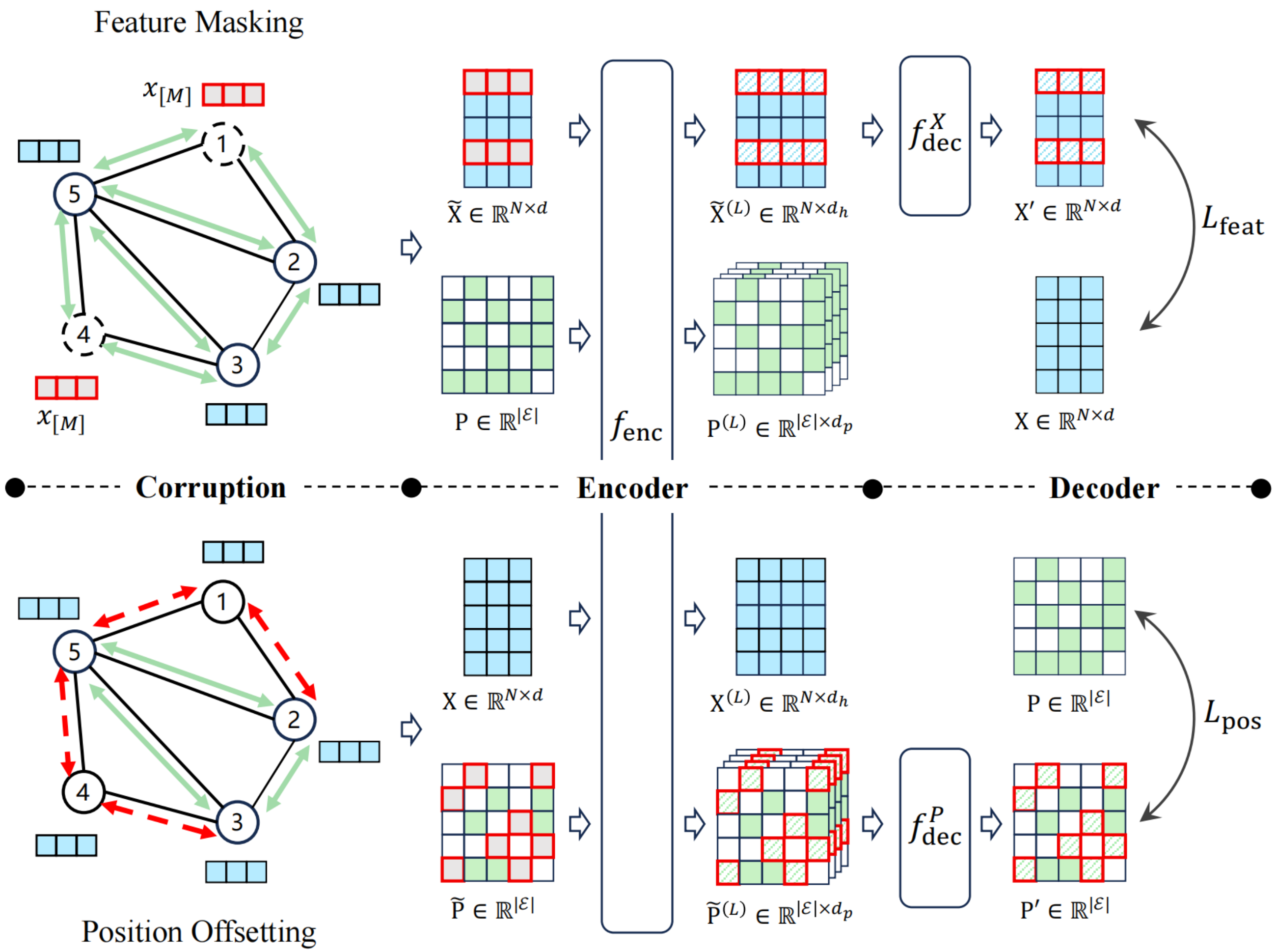
$$\mathcal{L}_{\text{feat}} = \frac{1}{|\widetilde{\mathcal{V}}|}\sum_{v_i \in \widetilde{\mathcal{V}}}\left(1 - \frac{\mathbf{X}_i^T \mathbf{X}_i'}{\|\mathbf{X}_i\| \cdot \|\mathbf{X}_i'\|}\right)^{\gamma},$$

- Position Reconstruction.

$$\mathbf{P}_{i,j}' = f_{\text{dec}}^P\left(\widetilde{\mathbf{P}}_{i,j}^{(L)}\right)$$

$$\mathcal{L}_{\text{pos}}^{i,j} = \begin{cases} \frac{\left(\mathbf{P}_{i,j}' - \mathbf{P}_{i,j}\right)^2}{2}, & \text{if } |\mathbf{P}_{i,j}' - \mathbf{P}_{i,j}| < 1 \\ |\mathbf{P}_{i,j}' - \mathbf{P}_{i,j}| - \frac{1}{2}, & \text{otherwise} \end{cases}$$

$$\mathcal{L}_{\text{pos}} = \frac{1}{\sum_{v_i \in \widetilde{\mathcal{V}}} |\mathcal{N}_i|} \sum_{v_i \in \widetilde{\mathcal{V}}, j \in \mathcal{N}_i} \mathcal{L}_{\text{pos}}^{i,j}$$



- Overall Loss Function. $\mathcal{L} = \mathcal{L}_{\text{feat}} + \alpha \mathcal{L}_{\text{pos}}$

Performance of Node Classification

Dataset	Small Graphs				Large Graphs	
	BlogCatalog	Chameleon	Squirrel	Actor	arXiv-year	Penn94
Supervised	80.52±2.10	80.02±0.87	71.91±1.03	33.93±2.47	46.02±0.26	81.53±0.55
DGI	72.07±0.16	43.83±0.14	34.56±0.10	27.98±0.09	-	-
BGRL	79.74±0.46	61.24±1.07	43.24±0.52	26.61±0.57	<u>41.43±0.04</u>	63.31±0.49
MVGRL	63.24±0.94	73.19±0.42	60.09±0.44	34.64±0.20	-	-
CCA-SSG	74.00±0.28	75.00±0.75	61.58±1.98	27.79±0.58	40.78±0.01	62.63±0.20
Sp^2GCL	72.73±0.46	78.88±1.04	62.61±0.87	<u>34.70±0.92</u>	39.09±0.02	68.80±0.45
VGAE	60.47±1.84	62.32±1.90	42.50±1.35	31.57±0.75	36.39±0.21	55.31±0.28
GraphMAE	79.90±1.13	<u>79.50±0.57</u>	61.13±0.60	32.15±1.33	40.30±0.04	67.97±0.21
GraphMAE2	77.34±0.12	79.13±0.19	<u>70.27±0.88</u>	34.48±0.26	38.97±0.03	67.86±0.42
MaskGAE	73.10±0.08	74.50±0.87	68.53±0.44	33.44±0.34	40.59±0.04	63.84±0.03
S2GAE	75.76±0.43	60.24±0.37	68.60±0.56	26.17±0.38	40.32±0.12	<u>70.24±0.09</u>
AUG-MAE	<u>82.03±0.69</u>	70.10±1.88	62.57±0.67	33.42±0.38	37.10±0.13	69.90±0.43
GraphPAE	**85.76±1.22**	**80.51±1.25**	**72.05±1.40**	**38.55±1.35**	**41.85±0.04**	**71.79±0.37**

Table 1. Node classification results of different graph self-supervised learning.

Performance of Graph Prediction

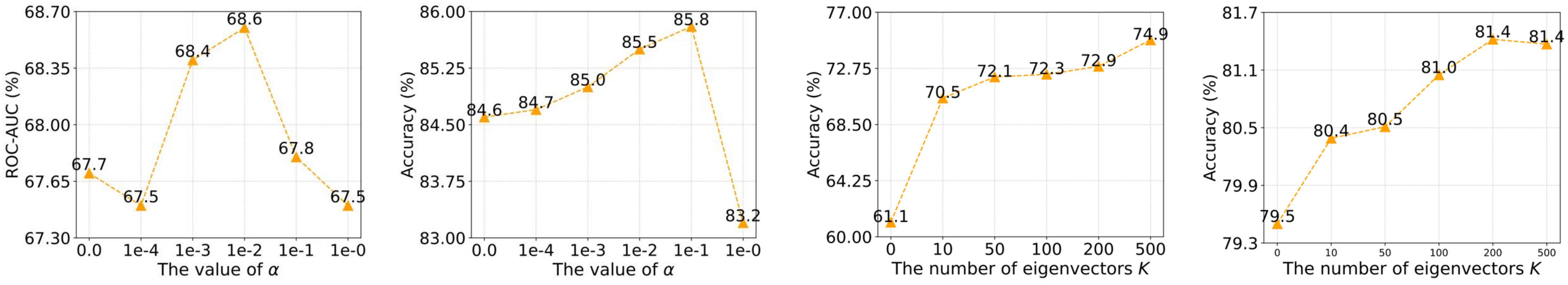
Task	Regression (Metric: RMSE ↓)			Classification (Metric: ROC-AUC% ↑)			
Dataset	molesol	molipo	molfreesolv	molbace	molbbbp	molclintox	moltocx21
Supervised	1.173±0.057	0.757±0.018	2.755±0.349	80.42±0.96	68.17±1.48	88.14±2.51	74.91±0.51
InfoGraph	1.344±0.178	1.005±0.023	10.005±8.147	73.64±3.64	66.33±2.79	64.50±5.32	69.74±0.57
GraphCL	1.272±0.089	0.910±0.016	7.679±2.748	73.32±2.70	68.22±2.19	74.92±4.42	72.40±1.07
MVGRL	1.433±0.145	0.962±0.036	9.024±1.982	74.88±1.43	67.24±3.19	73.84±2.75	70.48±0.83
JOAO	1.285±0.121	0.865±0.032	5.131±0.782	74.43±1.94	67.62±1.29	71.28±4.12	71.38±0.92
Sp^2GCL	1.235±0.119	0.835±0.026	4.144±0.573	78.76±1.43	**68.72±1.53**	80.88±3.86	73.06±0.75
GraphMAE	1.050±0.034	0.850±0.022	2.740±0.233	79.14±1.31	66.55±1.78	80.56±5.55	73.84±0.58
GraphMAE2	1.225±0.081	0.885±0.019	2.913±0.293	80.74±1.53	67.67±1.44	75.75±3.65	72.93±0.69
StructMAE	1.499±0.043	1.089±0.002	2.568±0.262	77.75±0.42	65.66±1.16	79.42±4.56	71.13±0.61
AUG-MAE	1.248±0.026	0.917±0.013	2.395±0.158	78.54±2.49	67.05±0.63	82.66±1.98	74.33±0.07
GraphPAE	**1.015±0.045**	**0.810±0.018**	**2.058±0.188**	**81.11±1.24**	68.56±0.71	**82.69±3.39**	**74.46±0.54**

Table 2. Graph regression and classification results of different graph self-supervised learning on OGB datasets.

Ablation Studies

Exp No.	Corrupt Info. Feature	Corrupt Info. Position	Recon Info. Feature	Recon Info. Position	Dual-Path	Node-level Blog (↑)	Node-level Squirrel (↑)	Graph-level Bace (↑)	Graph-level Bbbp (↑)	Graph-level Freesolv (↓)
a	✓	✓	✓			82.8±1.7	66.4±1.6	78.4±1.2	66.4±1.7	2.79±0.40
b	✓	✓	✓	✓		83.5±1.0	68.5±0.9	78.9±2.1	66.8±0.6	2.44±0.36
c	✓		✓		✓	84.6±1.6	71.3±0.9	79.4±3.4	67.7±0.9	2.20±0.14
d	✓	✓	✓	✓	✓	**85.8±1.2**	**72.1±1.4**	**81.1±1.2**	**68.6±0.7**	**2.06±0.19**

Table 3. Ablation studies of position reconstruction and framework design.



(a) Molbbbp (b) BlogCatalog (c) Squirrel (d) Chameleon

Figure 1: Influence of loss weight $\mathcal{L}_{pos}$ and the number of eigenvectors K.

Paper

Code

Thanks

Q&A

liuyangjanet@bupt.edu.cn