More Interpretable Graph Similarity Computation via Maximum Common Subgraph Inference

Zixun Lan[1 *] **Binjie Hong**[2 *] **Ye Ma**[3] **Fei Ma**[1 †]

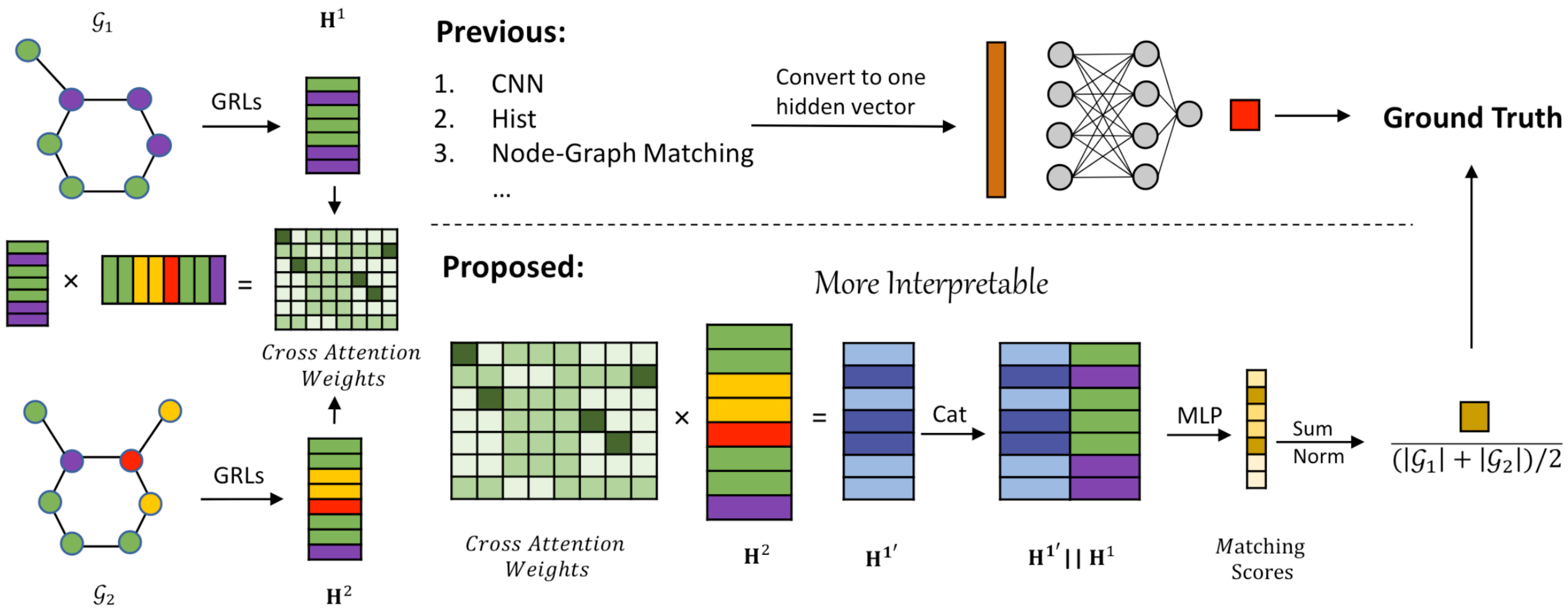
[1] Department of Applied Mathematics, School of Science
[2] Department of Information and Computing Science, School of Advanced Technology
[3] Department of Financial and Actuarial Mathematics, School of Science
Xi'an Jiaotong-Liverpool University, SIP, 215123 Suzhou, China

{zixun.lan19, binjie.hong19}@student.xjtlu.edu.cn, {ye.ma, fei.ma}@xjtlu.edu.cn

Introduction

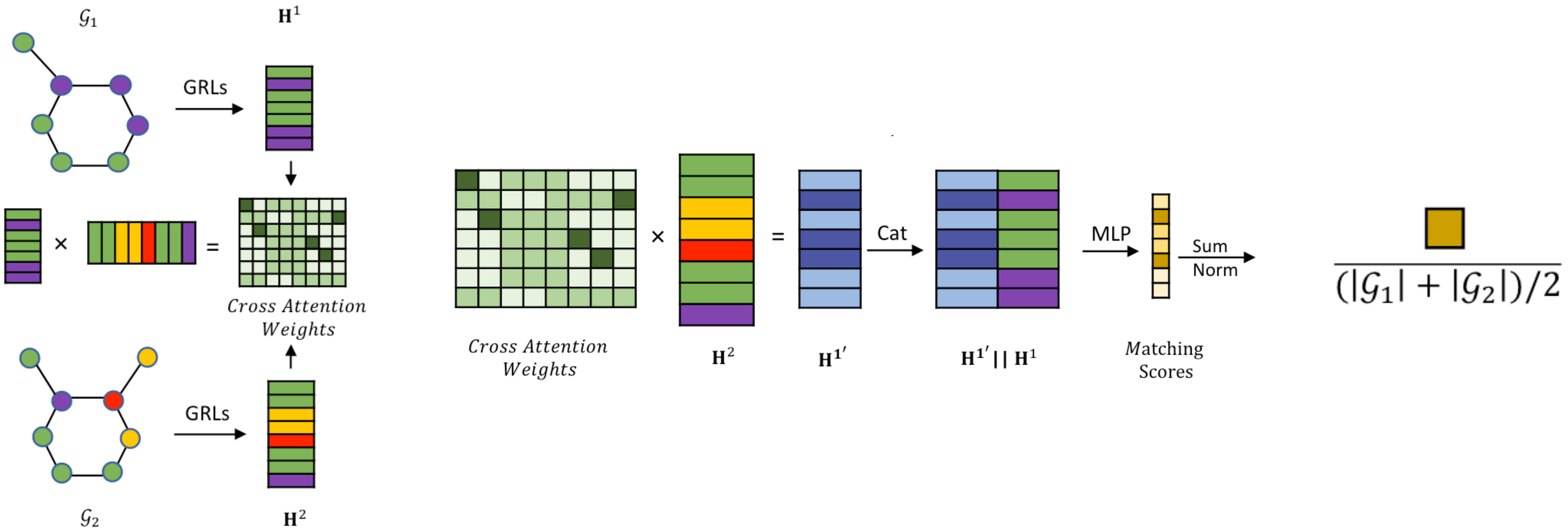
- Graph similarity measurement, which is to compute distance/similarity between two graphs, is a fundamental problem in graph-related tasks.
- Graph Edit Distance (**GED**) and Maximum Common Subgraph (**MCS**) are two domainagnostic graph similarity metrics, yet exact computation of both are known to be NP-hard.
- Previous methods lack interpretability despite the exploitation of interaction information.It is unclear what the final hidden vector represents and how to map it to ground truth.
- To cope with this limitation, this study proposes a more interpretable end-to-end paradigm for graph similarity learning, named Similarity Computation via Maximum Common Subgraph Inference (**INFMCS**).

Contrast



$\mathcal{G}_1$
$\mathbf{H}^1$
GRLs
Previous:
1. CNN
2. Hist
3. Node-Graph Matching
...
Convert to one hidden vector
Ground Truth
×
=
Cross Attention Weights
Proposed:
More Interpretable
$\mathcal{G}_2$
GRLs
$\mathbf{H}^2$
×
=
Cat
MLP
Sum
Norm
$(|\mathcal{G}_1| + |\mathcal{G}_2|)/2$
Cross Attention Weights
$\mathbf{H}^2$
$\mathbf{H}^{1'}$
$\mathbf{H}^{1'} \,||\, \mathbf{H}^1$
Matching Scores

Model Design



Model Design

Graph Similarity Learning

- Given a pair of input graphs $(\mathcal{G}_1, \mathcal{G}_2)$, the aim of graph similarity learning is to produce a similarity score $y = s(\mathcal{G}_1, \mathcal{G}_2) \in \mathcal{Y}$.
- For graph-graph classifcation task, the scalar y represents the class label, i.e., $y \in \mathcal{Y} = \{0,1\}$; for graph-graph regression task, the scalar y measures the graph similarity, i.e., $y \in \mathcal{Y} = [0,1]$.

Model Design

Similarity Computation

- Given the node representations of the last layer of the graph representation learning $\mathbf{H}^1 = [\mathbf{h}_1^1; \mathbf{h}_2^1; \cdots \mathbf{h}_{|\mathscr{V}_1|}^1] \in \mathscr{R}^{|\mathscr{V}_1| \times d}$ for $\mathscr{G}_1$ and $\mathbf{H}^2 = [\mathbf{h}_1^2; \mathbf{h}_2^2; \cdots \mathbf{h}_{|\mathscr{V}_2|}^2] \in \mathscr{R}^{|\mathscr{V}_2| \times d}$ for $\mathscr{G}_2$

$$a_{ij} = \frac{\exp\left(s_h\left(\mathbf{h}_i^1, \mathbf{h}_j^1\right) \times \tau_*^{-1}\right)}{\sum_{j'} \exp\left(s_h\left(\mathbf{h}_i^1, \mathbf{h}_{j'}^1\right) \times \tau_*^{-1}\right)}, \mathbf{h}_{i'}^1 = \sum_j a_{ij} \mathbf{h}_i^{(t)},$$

$$\hat{y} = \frac{\sum_i s_i}{(|\mathcal{G}_1| + |\mathcal{G}_2|)/2}, s_i = \operatorname{sigmoid}\left(\operatorname{MLP}\left(\mathbf{h}_i^1 \| \mathbf{h}_{i'}^1\right)\right).$$

Model Design

Loss Functions

- For the graph-graph classification task

$$\mathcal{L}_c = -\frac{1}{|\mathrm{D}|}\sum_{i=1}^{|\mathrm{D}|} y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)$$

- For the graph-graph regression task

$$\mathcal{L}_r = \frac{1}{|\mathrm{D}|}\sum_{i=1}^{|\mathrm{D}|} (y_i - \hat{y}_i)^2$$

Model Design

Graph Convolution with Transformer

- In this study, we use **GCN** to compute node-level embeddings.
- Over-smoothing constrains graph convolution from stacking multiple layers, resulting in a gap between the shallow **GCN** and the sizeable receptive field.
- To fill this gap, we stack some vanilla transformer encoder layers with graph convolution layers.

Model Design

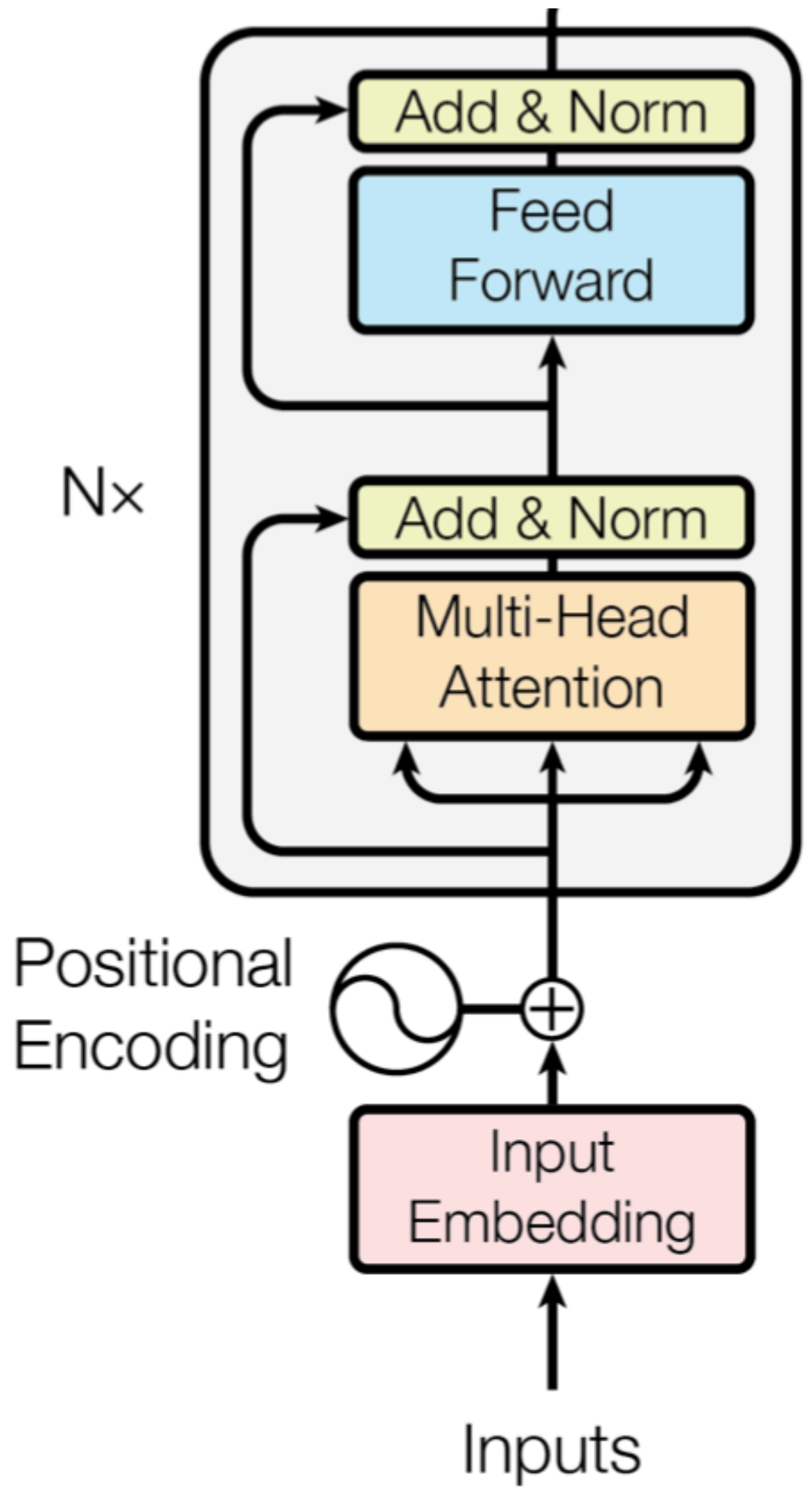
Graph Convolution with Transformer

$$Q^h = H^{(l)} W_Q^h, \quad K^h = H^{(l)} W_K^h, \quad V^h = H^{(l)} W_V^h,$$

$$A^h = \frac{Q^h {K^h}^\top}{\sqrt{d_K}}, H^h = \mathrm{softmax}(A^h) V^h,$$

$$H' = W \cdot \left(\|_h H^h \right) + H^{(l)}$$

$$H^{(l+1)} = \mathrm{FFN}\left(\mathrm{LN}\left(H'\right)\right) + H'.$$



Model Design

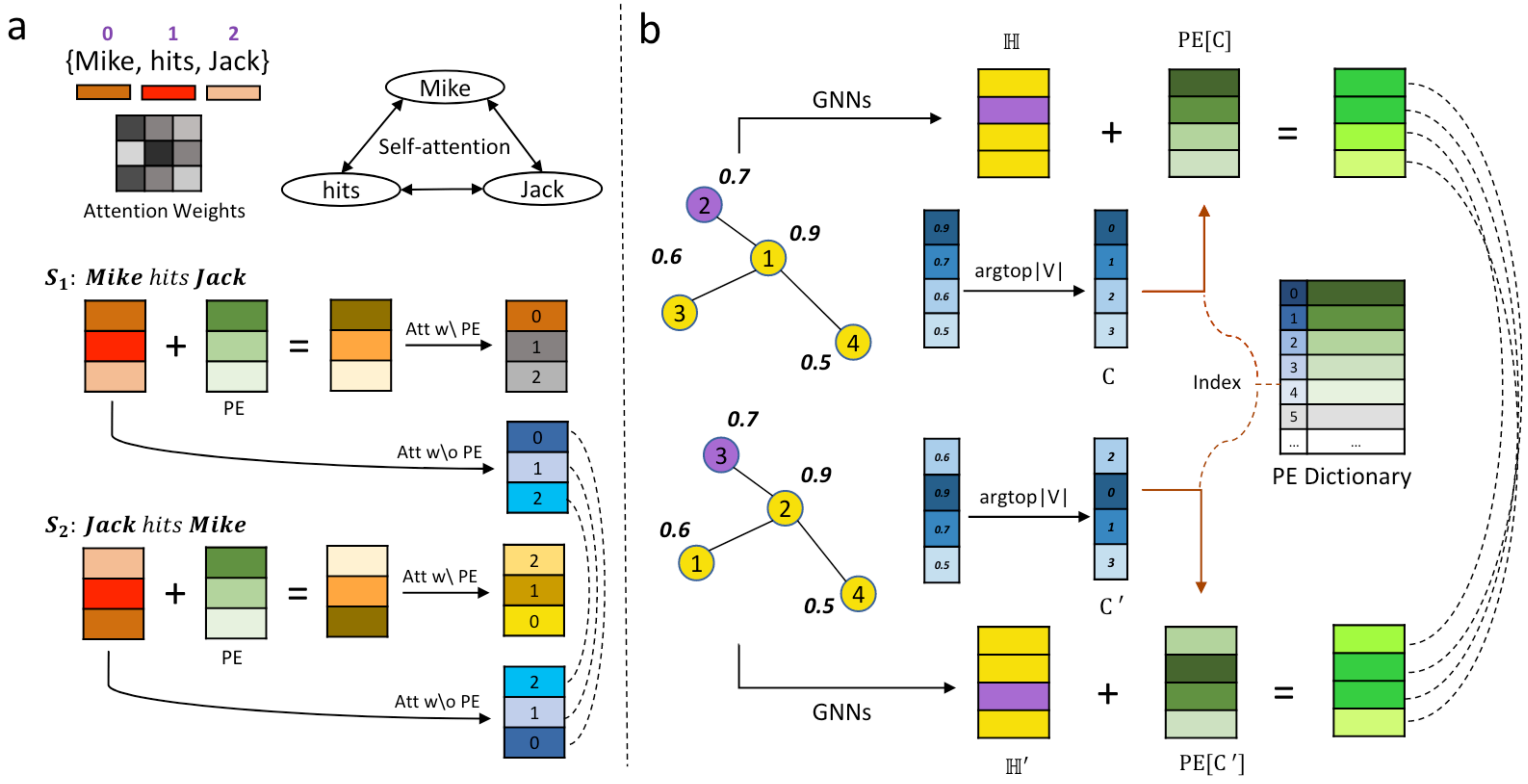
Positional Encoding

- For sentence representation, extensive experiments show the importance of Positional Encoding.
- However, graphs are permutation-invariant, resulting in no order for nodes. Thus, we propose a permutation-invariant node ordering $\mathbf{C} \in \mathscr{R}^{|\mathscr{V}|}$ based on closeness centrality

$$\mathbf{C} = \text{argtop}|\mathcal{V}| \left(\left[\mathrm{c}_1, \mathrm{c}_2, \cdots, \mathrm{c}_{|\mathcal{V}|} \right] \right), \mathrm{c}_i = \frac{n-1}{|\mathcal{V}|-1} \frac{n-1}{\sum_{j=1}^{n-1} d(j,i)}$$

Model Design

Positional Encoding



Model Design

Graph Convolution with Transformer

- We denote the vanilla transformer encoder by $\mathrm{TranformerEncoder}(\cdot)$. Given a learnable Positional Encoding dictionary $\mathbf{PE}[\mathbf{C}] \in \mathscr{R}^{|\mathscr{V}| \times d}$, final node representations $\mathbf{H} \in \mathscr{R}^{|\mathscr{V}| \times d}$ is derived by

$$\mathbf{H} = \mathrm{TranformerEncoder}(\mathscr{H}), \mathscr{H} = \mathbb{H} + \mathbf{PE}[\mathbf{C}]$$

Evaluation

Graph-Graph Classification Task

Table 1: Graph-Graph classification results (AUC score) with standard deviation (in percentage).

Datasets	FFmpeg			OpenSSL		
	[3, 200]	[20, 200]	[50, 200]	[3, 200]	[20, 200]	[50, 200]
SimGNN	95.38±0.76	94.32±1.01	93.45±0.54	95.96±0.31	93.38±0.82	94.25±0.85
GMN	94.15±0.62	95.92±1.38	94.76±0.45	96.43±0.61	93.03±3.81	93.91±1.65
GraphSim	97.46±0.30	96.49±0.28	94.48±0.73	96.84±0.54	94.97±0.98	93.66±1.84
MGMN	98.07±0.06	98.29±0.10	97.83±0.11	96.90±0.10	97.31±1.07	95.87±0.88
PSimGNN	96.67±0.54	96.86±0.95	95.23±0.15	96.10±0.46	94.67±1.30	93.46±1.59
GOTSim	96.93±0.34	97.01±0.52	95.65±0.31	97.87±0.49	96.42±1.89	95.97±1.06
H2MN	98.28±0.20	98.54±0.14	98.30±0.29	98.27±0.16	98.47±0.38	97.78±0.75
INFMCS	**98.49±0.09**	**99.36±0.13**	**99.48±0.20**	**98.34±0.20**	**99.14±0.31**	**99.26±0.45**

Evaluation

Graph-Graph Regression Task

Table 2: Graph-Graph regression results about mse($\times 10^{-2}$), ρ and p@10 on the MCS metric.

Datasets	AIDS(2-15)			LINUX(2-15)			PTC_MM(all)		
Metrics	mse↓	ρ↑	p@10↑	mse↓	ρ↑	p@10↑	mse↓	ρ↑	p@10↑
EMBAVG	33.20	0.0045	0.0540	0.83	0.5922	0.1340	35.03	0.0497	0.3471
GMN	32.20	0.0039	0.0578	3.99	0.0561	0.1340	35.03	0.0370	0.3500
GraphSim	2.73	0.1688	0.0578	0.81	0.2260	0.1340	3.21	0.5001	0.3500
SimGNN	2.65	0.1784	0.0596	0.83	0.4281	0.2370	3.27	0.5280	0.3500
SMPNN	2.89	0.2046	0.1056	12.59	0.5502	0.4280	4.67	0.4558	0.4353
MGMN	1.69	0.5300	0.1683	0.87	0.5351	0.3664	1.43	0.7329	0.5200
PSimGNN	2.54	0.1031	0.0452	1.83	0.4311	0.2668	3.43	0.4359	0.4280
GOTSim	1.77	0.5550	0.1763	0.61	0.3752	0.2569	2.75	0.3495	0.3431
H2MN	1.29	0.6745	0.2097	0.44	0.6364	0.4795	1.07	0.8823	0.7182
INFMCS	**0.30**	**0.9352**	**0.7976**	**0.02**	**0.9814**	**0.8870**	**0.71**	**0.9205**	**0.7794**

Evaluation

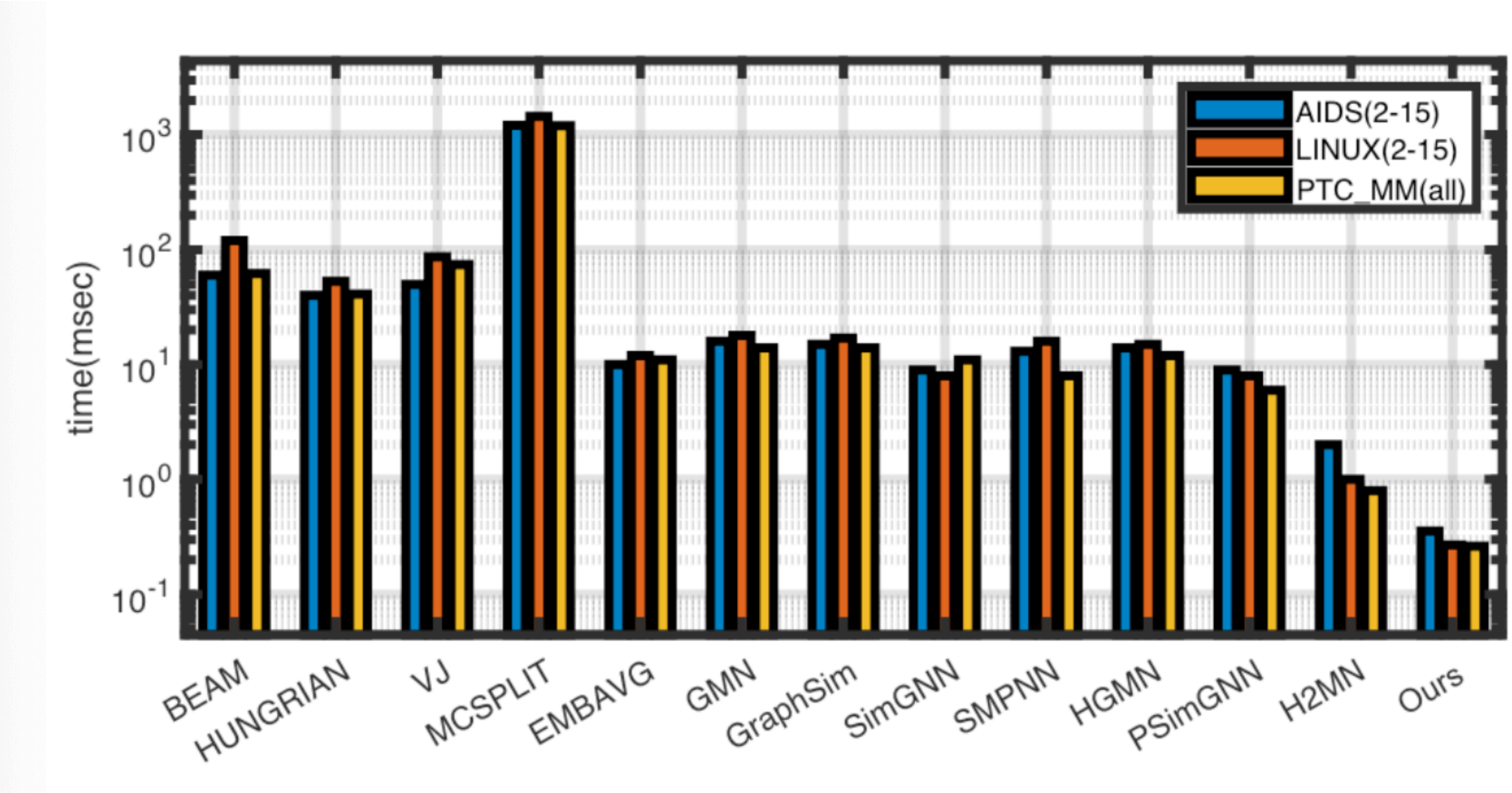
Graph-Graph Regression Task

Table 3: Graph-Graph regression results about mse($\times 10^{-2}$) on the synthetic datasets.

Datasets	BA100		BA200		BA300	
Metric	mse(MCS)	mse(GED)	mse(MCS)	mse(GED)	mse(MCS)	mse(GED)
EMBAVG	16.21	10.581	20.24	9.171	21.79	12.732
GMN	16.21	8.831	20.24	9.002	20.14	8.756
GraphSim	0.20	0.065	0.44	0.140	0.57	0.062
SimGNN	0.20	**0.060**	0.05	0.180	0.02	0.110
SMPNN	1.10	22.530	0.32	23.920	0.24	24.290
MGMN	0.35	1.033	0.27	0.901	0.44	0.071
PSimGNN	0.48	1.932	0.51	1.366	0.67	0.103
H2MN	0.02	0.187	0.01	0.532	0.02	0.034
INFMCS	**5.49e-7**	**0.061**	**1.80e-5**	**0.011**	**0.003**	**0.005**

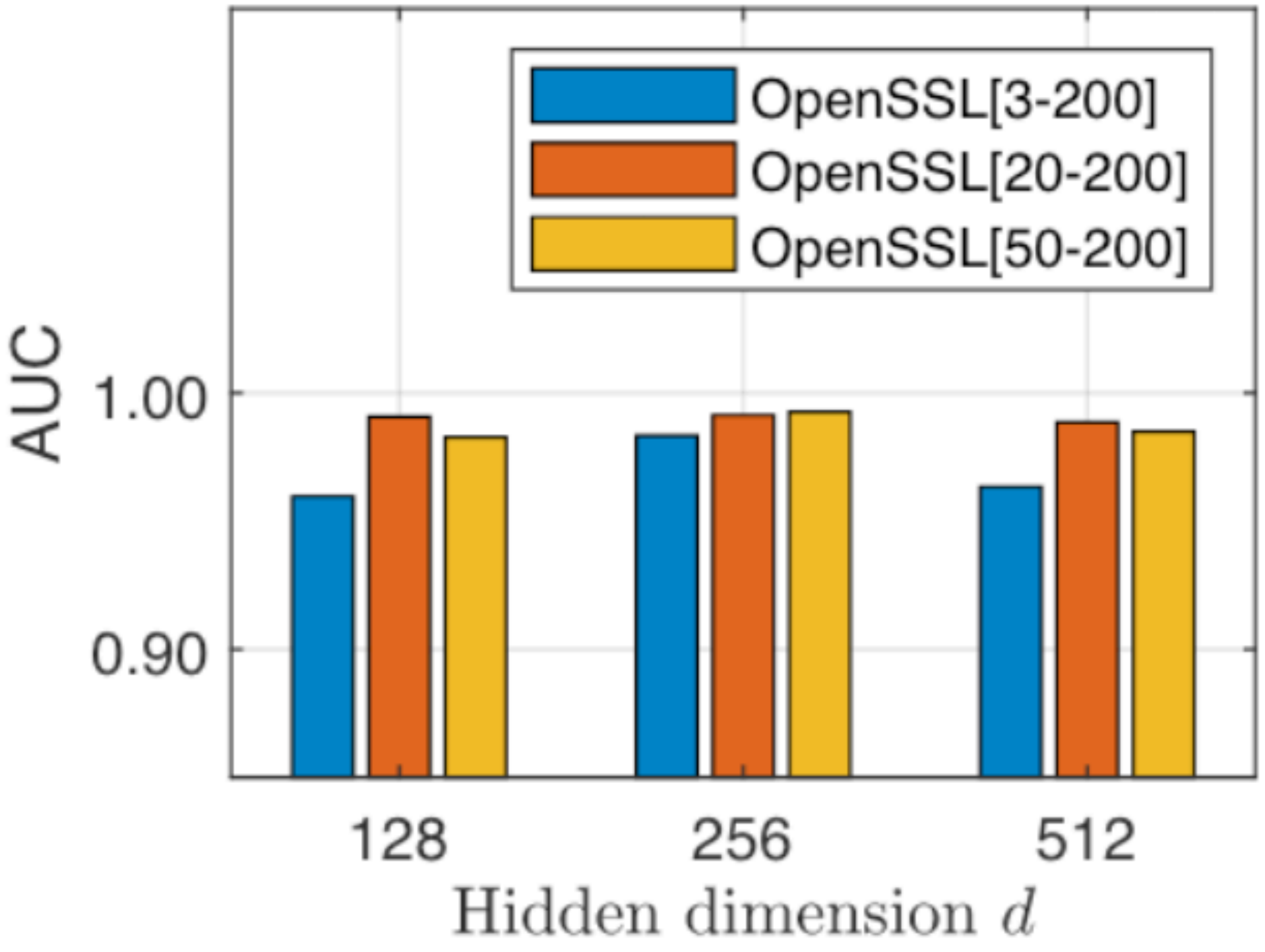
Evaluation

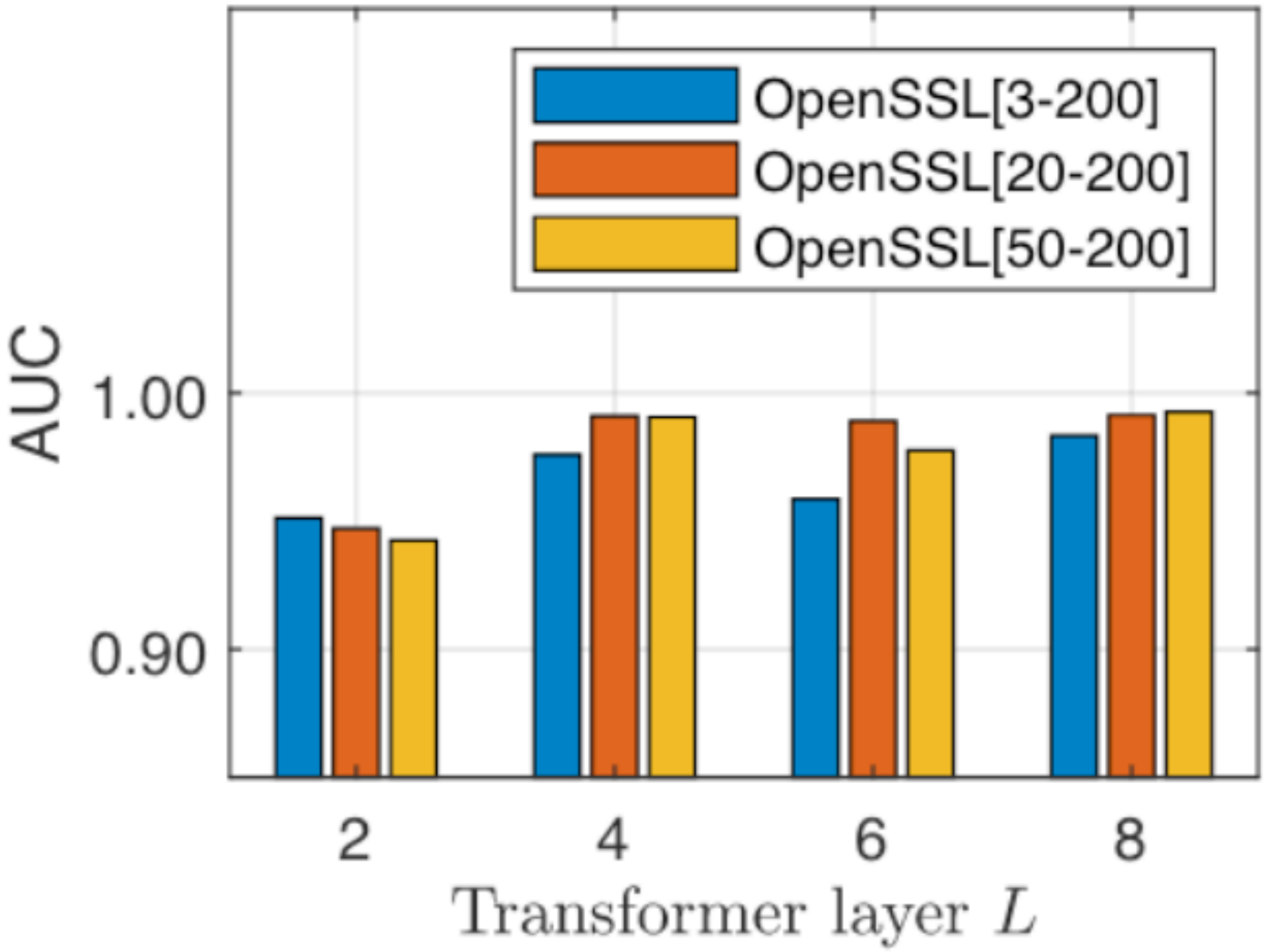
Efficiency



Evaluation

Hyperparameter sensitivity analysis





Evaluation

Ablation Study

Table 4: Ablation study on the FFmpeg.

(AUC score)	3-200	20-200	50-200
H2MN-H	97.50	98.12	98.05
BASE	98.16	98.83	98.87
BASE+T	97.13	98.20	98.48
BASE+H	98.01	98.42	98.56
BASE+T+PE	98.49	99.36	99.49

Evaluation

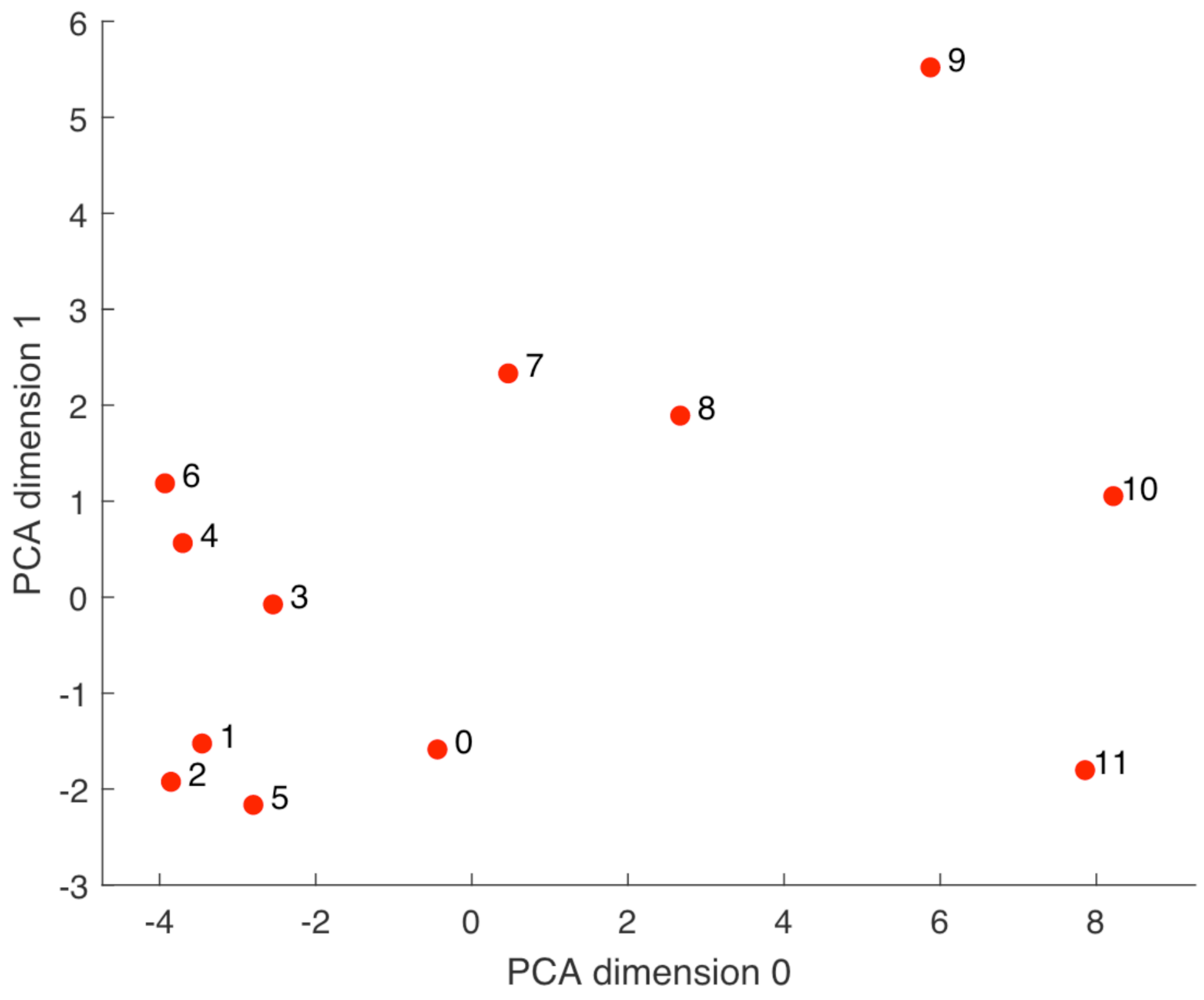
Ablation Study

Table 5: Ablation study on the MCS metric.

(mse$\times 10^{-2}$)	AIDS	LINUX	PTC_MM
H2MN-H	1.63	0.56	1.18
BASE	1.41	0.36	0.98
BASE+T	3.21	0.93	1.24
BASE+H	1.70	0.21	1.02
BASE+T+PE	0.30	0.02	0.71

Evaluation

Positional Encoding



Evaluation

Infer MCS and Interpretability analysis

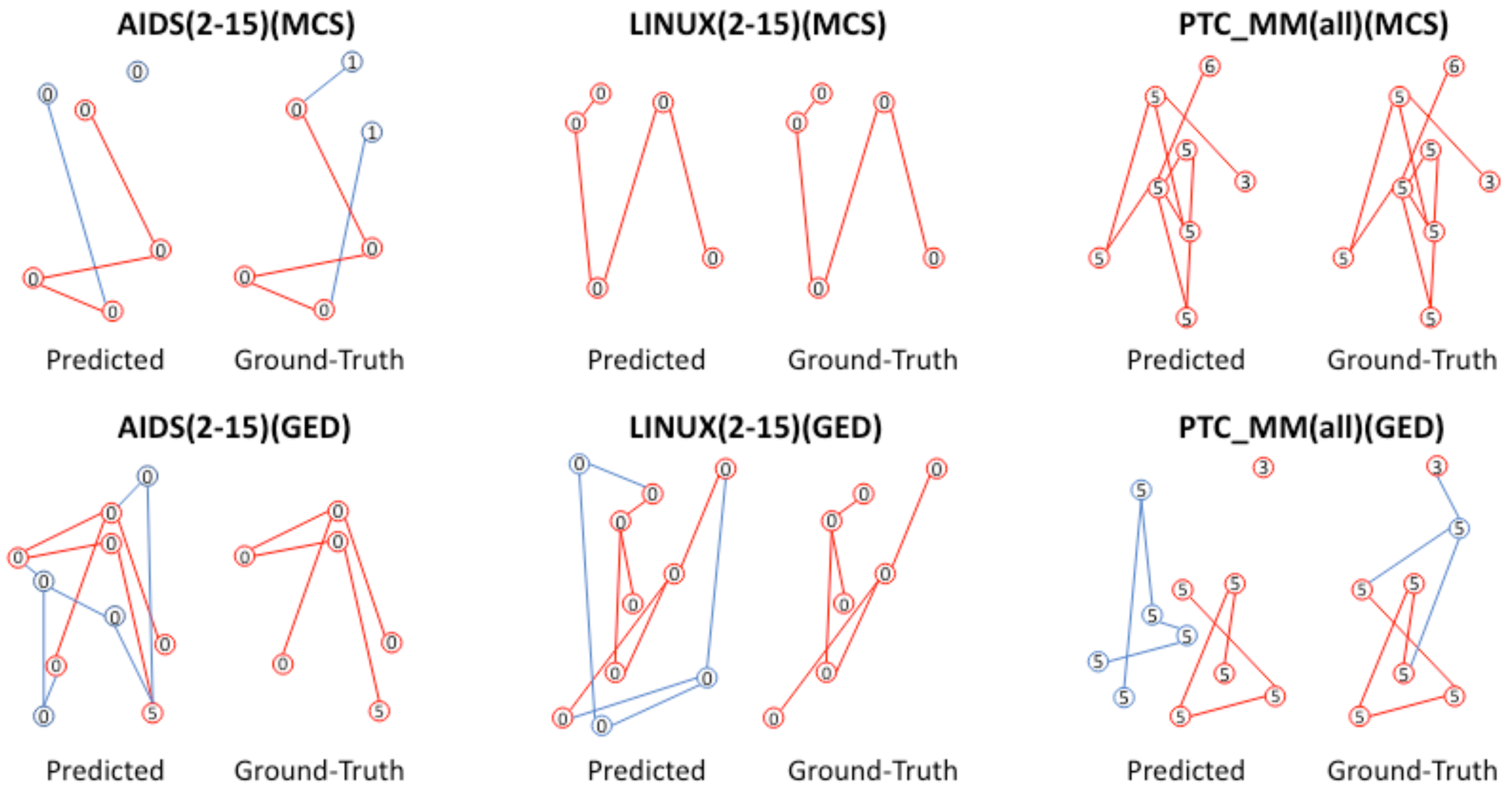


Figure 6: Visualizations of inferring MCS.

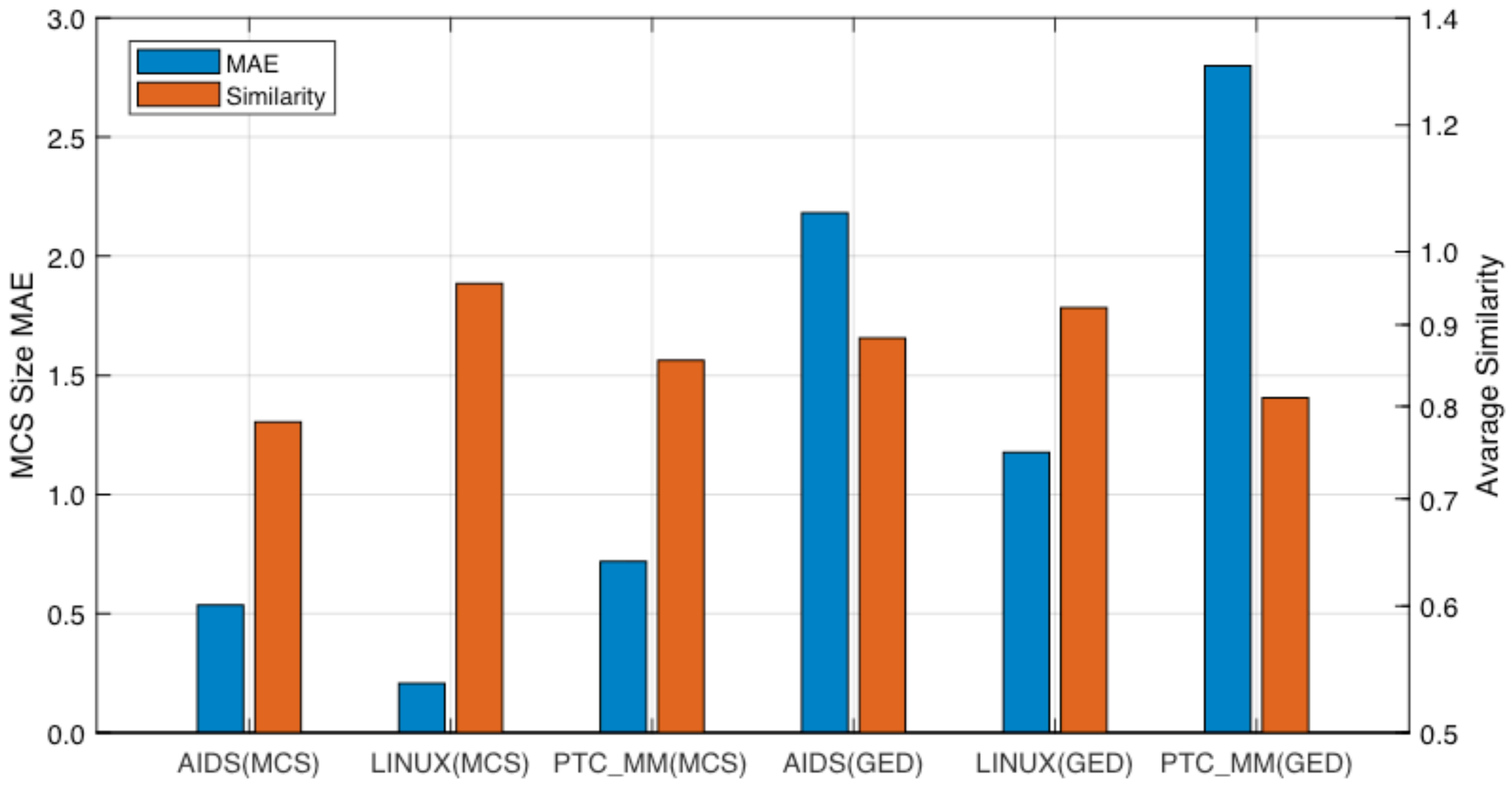


Figure 7: Interpretability analysis.

Evaluation

Case Study

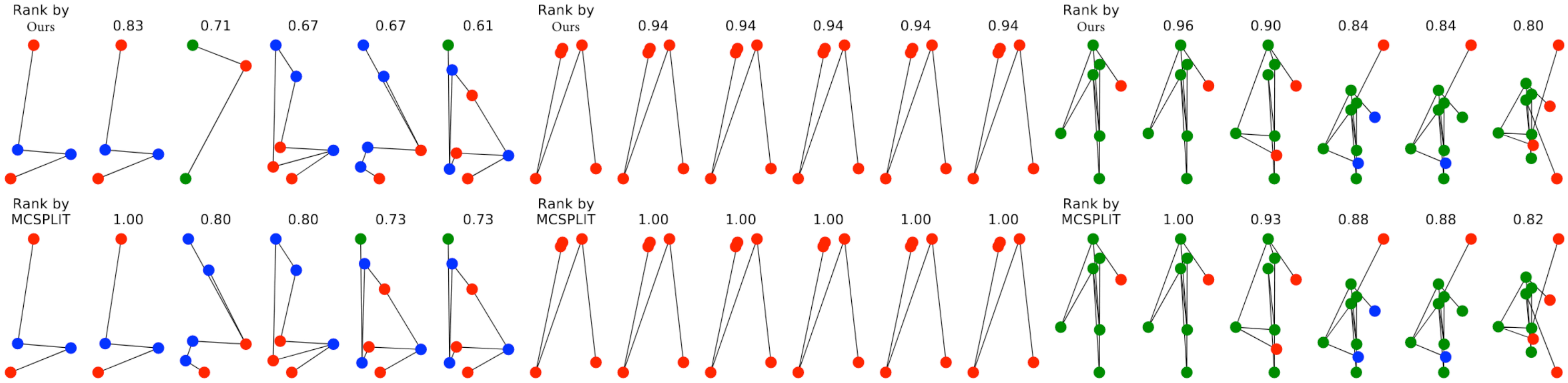


Figure 8: Visualization of ranking results. From left to right: AIDS, LINUX, PTC_MM.