A DATASETS

In this section, we include 14 node-level benchmark datasets (See Table 6 for statistics) and 7 graph-level benchmark datasets (See Table 7 for statistics) for experiments. These datasets are publicly available online and have been frequently adopted by other researchers.

Table 6: Dataset statistics of node-level benchmarks.

Data	# Nodes	# Edges	# Features	Split ratio	# Classes
Cora	2, 708	5, 429	1, 433	85/5/15	7
CiteSeer	3, 312	4, 660	3, 703	85/5/15	6
PubMed	19, 717	44, 338	500	85/5/15	3
BlogCatalog	5, 196	171, 743	8, 189	85/5/15	–
Flickr	7, 575	239, 738,	12, 047,	85/5/15	–
ogbl-ddi	4, 267	1, 334, 889	-	80/10/10	–
ogbl-collab	235, 868	1, 285, 465	128	92/4/4	–
ogbl-ppa	576, 289	30, 326, 273	58	70/20/10	–
Amazon-Computers	13, 752	245, 861	767	–	10
Amazon-Photo	7, 650	119, 081	745	–	8
Coauthor-CS	18, 333	81, 894	6, 805	–	15
Coauthor-Physics	34, 493	247, 962	8, 415	–	5
ogbn-arxiv	169, 343	1, 166, 243	128	–	40
ogbn-proteins	132, 534	39, 561, 252	8	–	112

Table 7: Dataset statistics of graph-level benchmarks.

Data	# graphs	Avg. # nodes	# classes
IMDB-B	1, 000	19.8	2
IMDB-M	1, 500	13.0	3
PROTEINS	1, 113	39.1	2
COLLAB	5, 000	74.5	3
MUTAG	188	17.9	2
REDDIT-B	2, 000	429.7	2
NCI1	4, 110	29.8	2

B MODEL DETAILS

In this section, we provide more details of the proposed S2GAE methods in terms of the model architecture, hyper-parameter, and hardware perspectives.

B.1 Details of the Model Architecture

Recall that our model S2GAE consists of two crucial components: GNN encoder and cross-correlation decoder. The GNN encoder is parameterized by classical GCN [19] modules. In experiments, we tested on two types of GNN variants: GCN [19] and SAGE [11]. The default choice is GCN. The cross-correlation decoder is parameterized by a 2-layer MLP with hidden dimension D. We adopt ReLU as the default hidden activation function.

B.2 Hyperparameter Configuration

To provide fair comparison with state-of-the-art competitors, we generally follow the same parameter settings across different baselines in terms of two downstream applications. In general, our model is optimized based on minibatch training following [15]. Specifically, in each step, we randomly sample a minibatch of positive edges from the training loader and then randomly generate one negative sample for each positive edge to construct the minibatch training set. It is worth noting that we do not conduct subgraph sampling for node representation as done in [11]. In contrast, we feed the whole adjacency matrix into the model to generate node representations based on GNN encoder.

Specifically, for three Planetoid datasets (Cora, CiteSeer, PubMed), two social networks (BlogCatalog and Flickr), two co-purchase networks (Amazon-Computers and Amazon-Photo), and two academic networks (Coauthor-CS and Coauthor-Physics), we adopt a two-layer GNN encoder with dimension 128. For six OGB datasets, we adopt a three-layer GNN encoder with hidden dimension 256. To be more specific, the batch size and learning rate for Cora, CiteSeer, Pubmed, BlogCatalog, and Flickr datasets are 1024 and 0.01, respectively. The batch size and learning ratio for ogbl-ddi, ogbl-collab, ogbl-ppa, ogbn-arxiv, and ogbn-proteins are fixed as 64 * 1024 and 0.001 as suggested in [15] [1].

We build our model based on Pytorch and PyG (PyTorch Geometric) library [8]. We train S2GAE for 200 epochs with Adam optimizer and early stopping with a patience of 50 epochs. Following common practice in [15, 16, 60], we employ a three-layer GNN encoder with dimension $d = 256$ for OGB benchmarks (ogbl-ddi, ogbl-collab, ogbl-ppa, ogbn-arxiv, and ogbn-proteins), while a two-layer GNN encoder with dimension $d = 128$ for other datasets. For graph-level classification, we strictly follow the architecture configurations of GraphMAE [14] on different datasets. There are two hyper-parameters in our model, i.e., masking ratio ω and hidden dimension D for the decoder network. We set $\omega = 0.7$ for link prediction while $\omega = 0.5$ for node and graph classification tasks. We search D in the set $\{128, 256, 512, 1024\}$ for all datasets. The default masking choice for three planetoid datasets and two social networks on link prediction task is *undirected masking*, while *directed masking* in all other scenarios. We use S2GAE refer S2GAE-GCN by default.

For all baselines, we tested their performances based on their open-resourced implementations. For datastes that are tested on original papers, we adopt the default hyperparameter configurations. For those datasets that are not tested on their papers, we search the best hyperparameters from the candidate pool suggested in original papers, and report the best results.

As most contrastive baselines are not originally evaluated on link prediction task, to make a fair comparison, we adopt fine-tune strategy to learn a downstream link predictor. Specifically, we adopt a two-layer MLPs to parameterize the link predictor and train it according to the training set, where the validation set is used to find the best hyper-parameters.

It is worth noting that we exclude GMAE for comparison because the source code of this paper was not open released. Moreover, their experimental settings for node classification and empirical datasets used for graph classification are different to us. Besides, MaskGAE is also excluded for graph classification task because their source code only support link prediction and node classification task. It's nontrivial to adopt their code for graph classification.

[1] https://github.com/snap-stanford/ogb/tree/master/examples/linkproppred

Table 8: Link prediction results on Planetoid benchmarks and social networks. The best results are highlighted.

	Cora AP	CiteSeer AP	PubMed AP	Blog. AP	Flickr AP	ogbl-ddi Hits@30	ogbl-collab Hits@100	ogbl-ppa Hits@50
DGI	90.61 ± 1.00	95.72 ± 0.10	92.23 ± 0.50	73.81 ± 1.72	92.38 ± 0.22	–	–	–
GIC	93.33 ± 0.70	**96.80 ± 0.50**	93.54 ± 0.30	61.96 ± 2.55	86.08 ± 0.16	–	–	–
MVGRL	87.78 ± 0.51	89.90 ± 0.42	87.69 ± 0.38	70.53 ± 1.98	91.32 ± 0.29	–	–	–
BGRL	86.59 ± 0.18	79.02 ± 0.22	95.56 ± 0.13	70.28 ± 1.77	88.58 ± 0.21	–	–	–
GAE	92.83 ± 0.03	91.68 ± 0.05	96.50 ± 0.02	84.87 ± 1.57	92.43 ± 0.26	51.56 ± 4.19	52.30 ± 1.01	**10.82 ± 1.04**
GraphSage	88.24 ± 0.87	87.90 ± 2.54	89.44 ± 0.82	77.09 ± 0.87	89.30 ± 0.16	65.80 ± 6.94	60.23 ± 1.20	8.92 ± 2.28
ARGE	93.23 ± 0.00	93.03 ± 0.00	97.11 ± 0.00	72.29 ± 0.36	87.05 ± 0.08	23.86 ± 6.53	37.66 ± 1.98	3.83 ± 0.84
GPT-GNN	92.53 ± 0.63	92.09 ± 0.76	97.79 ± 0.04	85.00 ± 0.29	93.05 ± 0.14	42.57 ± 6.01	51.32 ± 1.45	7.88 ± 0.92
RRL	89.60 ± 1.45	87.84 ± 1.18	91.47 ± 0.57	82.22 ± 0.86	92.76 ± 0.05	19.33 ± 2.51	40.52 ± 2.23	3.85 ± 0.83
GraphMAE	90.32 ± 0.01	92.66 ± 0.35	94.07 ± 0.02	77.54 ± 1.06	88.94 ± 0.02	–	33.62 ± 1.50	3.62 ± 0.91
MaskGAE	**96.29 ± 0.23**	**98.25 ± 0.16**	**98.99 ± 0.06**	80.12 ± 2.79	<u>92.87 ± 0.36</u>	19.27 ± 1.59	39.98 ± 0.74	3.77 ± 0.06
S2GAE-SAGE	94.50 ± 0.86	94.68 ± 0.34	97.11 ± 0.19	78.99 ± 1.91	91.71 ± 0.20	**75.18 ± 6.57**	55.44 ± 0.82	4.79 ± 0.16
S2GAE-GCN	94.46 ± 0.24	93.81 ± 0.40	98.22 ± 0.05	**86.12 ± 0.49**	**94.10 ± 0.04**	**75.02 ± 2.26**	**61.01 ± 1.18**	9.97 ± 1.55

Table 9: Hyper-parameter analysis of S2GAE.

	$D = 128$	$D = 256$	$D = 512$	$D = 1024$
Cora	92.85 ± 0.22	93.52 ± 0.23	93.19 ± 0.34	93.47 ± 0.64
CiteSeer	93.42 ± 0.40	93.29 ± 0.49	93.65 ± 0.56	93.11 ± 0.51
PubMed	98.40 ± 0.05	98.45 ± 0.03	98.51 ± 0.07	98.58 ± 0.05

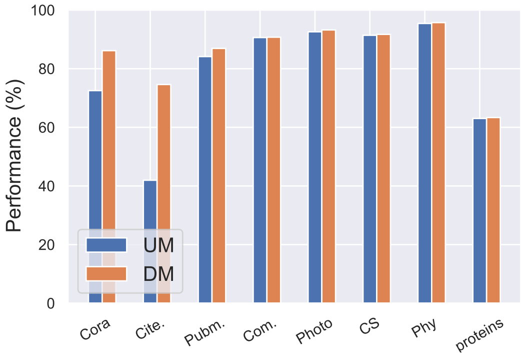

Figure 4: Ablation study of graph masking strategies on node classification.

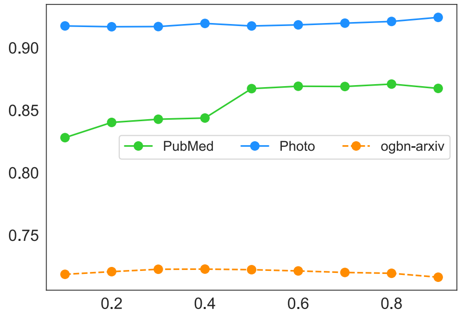

Figure 5: Ablation study of masking ratio on node classification.

Table 10: S2GAE vs. GAE in terms of running time (in second) and memory (GB) costs on large-scale OGB datasets.

	Method	ogbn-arxiv	ogbn-proteins	ogbl-collab	ogbl-ppa
memory	GAE	0.41	3.85	0.55	2.26
	S2GAE	0.66	1.73	0.89	2.47
time	GAE	1.57	674.00	2.00	188.09
	S2GAE	3.53	244.89	2.30	98.77

B.3 Hardware

We conduct all the experiments on a server with AMD EPYC 7282 16-Core processors, 252 GB memory, and four NVIDIA GeForce RTX 3090 GPUs.

C HYPER-PARAMETER ANALYSIS

We study the impact of hidden dimension D of the 2-layer MLP in our decoder network (in Eq. (5)). Table 9 shows the results on three Plantetoid datsets. We can observe that our model MGAE performs consistently good in a wide range of dimension choices. Similar observations are obtained by other datasets. In experiments, we set $D = 512$ for OGB datasets, while $D = 256$ for other datasets.

D EFFICIENCY ANALYSIS

Besides, we also explore the efficiency of S2GAE against vanilla autoencoder baseline GAE. Table 10 shows the results of memory and running time costs per epoch. We can observe that S2GAE performs slightly lower than GAE on datasets that have relatively fewer edges, i.e., ogbn-arxiv and ogbl-collab, while it is more efficient on large datasets with more edges, i.e., ogbn-proteins and ogbl-ppa. This is reasonable because masking a large and dense graph can reduce the graph convolution costs and the number of recovered edges. For relatively small datasets, the additional edge masking cost itself may be larger than what is saved.