A Supplementary Material of SA-MLP

A.1 Details of Datasets

We provide the details of the five homophilic datasets (connected nodes tend to be the same label) and three heterophilic datasets (labels of connected nodes tend to be different) in the following:

- Homophilic Datasets
  - *Citeseer, Pubmed, Cora* Kipf & Welling (2017): For the basic citation datasets, nodes correspond to papers, edges correspond to citation links, the sparse bag-of-words are the feature representation of each node, and the label of each node represents the topic of the paper. Note that we use the public ten data split(48%/32%/20% for Train/Val/Test) in Pei et al. (2020); Zhu et al. (2020) to reduce randomness and enhance reproducibility. Compared with the GLNN that uses only 20 nodes of each class for training, the results of our splits are more stable and reduce the possibility of overfitting Zhu et al. (2020).
  - *Arxiv* Hu et al. (2020): The Arxiv dataset is a large-scale citation network collected from all Computer Science arXiv papers. Each node is an arXiv paper, and edges are citation relations between papers. The features are 128-dimensional averaged word embeddings of each paper, and labels are subject areas of papers.
  - *Products* Hu et al. (2020): The Products is a large-scale Amazon product co-purchasing network. Nodes represent products sold in Amazon, edges indicate the products purchased together, and features are 100-dimensional bag-of-words features.
- Heterophilic Datasets
  - *Squirrel, Chameleon* Pei et al. (2020): Chameleon and Squirrel are web pages extracted from different topics in Wikipedia. Similar to WebKB, nodes and edges denote the web pages and hyperlinks among them, respectively, and informative nouns in the web pages are employed to construct the node features in the bag-of-word form. Webpages are labeled in terms of the average monthly traffic level.
  - *Arxiv-year* Lim et al. (2021): Modifying node labels of the Arxiv dataset to the year of paper, and the goal is to predict the year of paper publication that allows for evaluation of GNNs in large-scale non-homophilous settings.

A.2 Additional Comparison of Citation Datasets

We provide additional experiments for the comparison under the random splits (20 labeled nodes of each class during training) of citation datasets used in GLNN and report the mean test accuracy in Table 6. We can observe that SA-MLP also consistently outperforms GLNN, NOSMOG and teacher GNN. Moreover, we conduct the production (*prod*) scenario that involves both inductive (*ind*) and transductive (*trans*) settings used in GLNN (see Table 6). We find that SA-MLP can also achieve the best performance across all scenarios. However, the performance of these splits that only contained 20 labeled nodes is susceptible to hyper-parameters since the few training data cause the risk of overfitting.

Table 6: Inductive, transductive, and production scenario of citation datasets under random splits.

Dataset	Eval	SAGE	GLNN	NOSMOG	SA-MLP
Cora	*prod*	79.53	77.82	81.02	**81.21**
	ind	81.03	73.21	81.36	**81.03**
	trans	79.16	78.97	80.93	**81.47**
Citeseer	*prod*	68.06	69.08	70.60	**70.67**
	ind	69.14	68.48	70.30	**70.53**
	trans	67.79	69.23	70.67	**70.81**
Pubmed	*prod*	74.77	74.67	75.82	**76.12**
	ind	75.07	74.52	75.87	**76.04**
	trans	74.70	74.70	75.80	**76.15**

Table 7: Comparison with GLNN+ in large-scale datasets.

Dataset	Setting	SAGE	GLNN	GLNN+	SA-MLP
Arxiv	*trans*	70.92	63.46	**72.15**	71.54
	online	67.69	56.35	56.56	**68.01**
Products	*trans*	78.61	68.86	77.65	**79.02**
	online	65.55	62.45	62.58	**67.46**
Arixv-year	*trans*	51.85	46.22	51.02	**53.31**
	online	48.42	36.92	36.81	**49.55**

Table 8: Speed comparison between SA-MLP and other inference acceleration of SAGE. Numbers (in ms) are inference time on 10 randomly chosen nodes. "*' indicates our implementation based on released codes of GLNN.

Model	Structure	Arxiv	Products
SAGE	✓	489.49	2071.30
QSAGE	✓	433.90	1946.49
PSAGE	✓	465.43	2001.46
NSSAGE	✓	91.03	107.31
GLNN+		3.34	7.56
SAGE*	✓	386.85	1957.11
GLNN+*		3.45	8.64
NOSMOG	✓	1.36	1.34
SA-MLP	✓	**1.18**	**1.12**

A.3 Additional Comparison of GLNN+

GLNN also provides a larger GLNN+, which scales hidden dimension from 256 to 1024 for Arxiv (GLNNw4) and 2048 for Product (GLNNw8), with a larger capacity but a slower speed. We provide additional experiments for the GLNN+ of the *trans* and *online* for large-scale OGB datasets. We omit other datasets since the performance of GLNN+ is similar to that of GLNN. From Table 7, we find that GLNN+ can improve the performance of large-scale datasets under the *trans* setting. However, it achieves similar results to GLNN under the *online* setting, which implies that the improvement of *trans* for OGB datasets is due to the memory capacity, i.e., the larger parameters of GLNN+ can memorize all the teacher outputs. It still does not fully understand the structure information and generalizes limitedly on unseen test nodes under the *online* setting. However, the improvement over both *trans* and *online* of our SA-MLP is due to explicit structure awareness.

A.4 Additional Inference Time Comparison

Following the settings GLNN Zhang et al. (2022), we also compare SA-MLP with other inference acceleration techniques with GNNs, including vanilla SAGE, quantized SAGE from FP32 to INT8 (QSAGE), SAGE with 50% weights pruned (PSAGE), and inference with neighbor sampling with fan-out 15 (NSSAGE). All GNNs have three layers and 256 hidden units, while GLNN+ has 1024 hidden units for Arxiv and 2048 for Products to achieve optimal performance Zhang et al. (2022). As shown in Table 8, all MLP-like students achieve substantially faster inference than GNN variants. Moreover, SA-MLP is the only method that processes structured inputs explicitly but offers the fastest inference speed without sacrificing performance. Following the original paper of NOSMOG, we set the hidden size to 256 and achieved the best performance. Although it processed the pre-compputed deepwalk embedding, the Setting the hidden Without considering the time cost of preprocessing by DeepWalk, NoSMOG's inference time is significantly less than that of GLNN. The reason is that it uses 256 as the hidden size to achieve optimal performance. Compared to GLNN+ and NOSMOG, the speedup can be attributed to Pytorch's sparse tensor multiplication with structure inputs and the smaller hidden unit size (128 for Arxiv and 64 for Products) employed in SA-MLP.