KDD 2024 Rebuttal Reviewer MQnk

April 12, 2024

1 Reviewer MQnk

Dear reviewer, thank you so much for your valuable comments. We address all of your concerns as follows

1.1 Q1

It still need some deeper analysis. For example, the different trend of approximation error on small graph(ogbn-arixv) and large graph(ogbn-products) in figure 3.

1.2 A1

We conclude that the differing trends of the two curves in Figure 3 are primarily due to the extent of staleness. In the case of ogbn-arixv, where the dataset size is smaller than ogbn-products, staleness is not as significant. Therefore, the approximation accumulates at a slower rate compared to ogbn-products (resulting in a smaller slope). Given the time constraints, we plan to conduct additional experiments to verify our conclusion after the rebuttal period.

1.3 Q2

The effectiveness should be illustrated on larger-graphs(e.g. ogbn-paper100M). On small graph the improvement is marginal without the std of the accuracy.

1.4 A2

We want to claim that we adhere to the experimental setups used in all baseline methods, including the selection of datasets to ensure fair comparison. For small dataset, we choose to use smaller batch size to demonstrate the effectiveness of our proposed methods. Given the time constrain, we will add std after the rebuttal period.

Following your recommendation, we carry out experiments using ogbn-papers100m, which is significantly larger in scale compared to other datasets. We report the performance and efficiency in Table 1.

The larger size of the ogbn-papers100M dataset exacerbates the staleness issue for GAS, leading to decreased accuracy and slower convergence, as anticipated. In contrast, REST showcases both high accuracy and efficiency, in line with our primary claim in the submission.

Table 1: Memory usage (MB) and running time (seconds) ogbn-papers100m.

Models	**Accuracy**	MEMORY(MB)	TIME(S)
GAS	64.9	15705	8840
REST	67.3	16808	4100

1.5 Q3

It needs provide the curves depicting the relationship between approximation error and epochs, as well as memory persistence and batch size, for backbones equipped with the proposed method. This will directly demonstrate the effectiveness of the proposed method and its underlying source.

1.6 A3

We present the two additional results regarding memory persistence and approximation error for REST+GAS on ogbn-arxiv as follows:

(1) Memory Persistence: It is evident that the persistence continuously decreases with the increase of the updating frequency. This directly demonstrates that staleness is reduced by REST.

(2) Approximation Error: It is also noticeable that the error value decreases compared to Figure 3 (a) in our submission. This provides another straightforward evidence showing the effectiveness of REST.

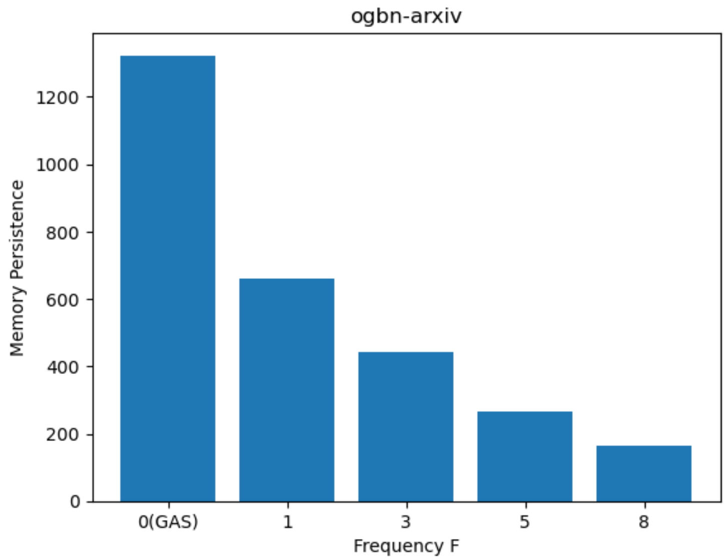

Figure 1: Memory Persistence on ogbn-arxiv

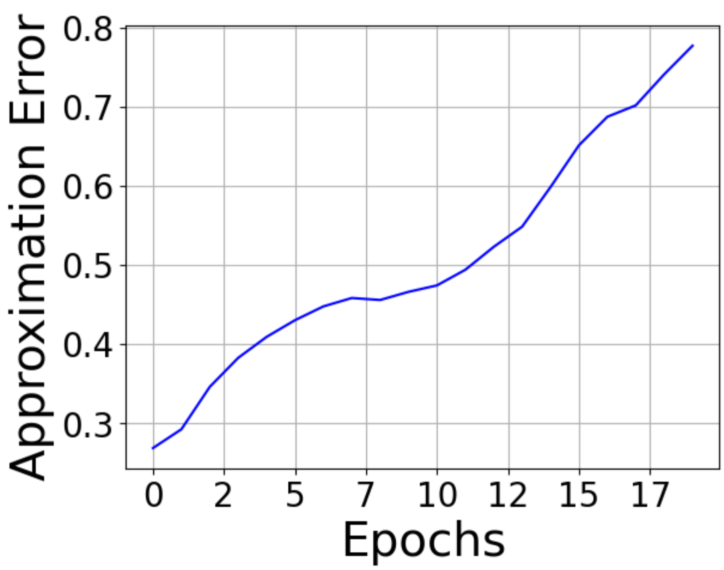

Figure 2: Approximation Error on ogbn-arxiv

1.7 Q4

How about the other methods mentioned (GraphFM OB, ReFresh. . .) perform? Can they decrease the feature staleness and approximation error? Why their performance are missing in the experiments section.

1.8 A4

GraphFM utilizes one-hop out-of-batch nodes to compensate for staleness of historical embeddings, while Refresh doesn't directly reduce staleness but instead avoids the use of embeddings that are too stale. We opted to use GAS as it's the most representative model and the basis of all related models. Following your suggestions, we have incorporated Refresh as Table 2 shown.

More importantly, REST can be integrated with any of them. We use GraphFM as an example to comprehensively verify our claim. Please refer to the Table 3 for the performance of GraphFM+REST. GraphFM shows a slight performance improvement by reducing staleness, but it still falls short of achieving superior performance as it doesn't fully address the staleness issue at its source, unlike REST. Conversely, our proposed method can be easily applied to GraphFM and achieve even better performance, highlighting the generality of REST.

Table 2: Accuracy comparison (%) with major baselines.

	# nodes	169K	2.4M
	# edges	1.2M	61.9M
Framework	**GNNs**	`ogbn-arxiv`	`ogbn-products`
GraphFM	GCN	71.8	76.8
	GCNII	72.9	77.4
Refresh	GCN	70.5	78.3
REST (Ours)	GCN	72.2	79.6
	APPNP	72.4	80.0
	GCNII	**73.1**	79.8
REST-IS (Our)	GCN	72.3	78.6
	APPNP	72.4	**80.5**
	GCNII	72.8	79.6

Table 3: Accuracy (%) improvement for GraphFM.

Dataset	Backbone	Parts	Batch Size	FM	**+REST**	**+REST-IS**
ogbn-products	**GCN**	**70**	5	76.3	77.9	78.0
			10	76.9	79.9	78.8
	APPNP	**40**	5	76.2	80.2	80.6
			10	77.1	80.3	80.6
	GCNII	**150**	5	75.3	76.2	76.6
			20	77.4	80.2	80.0
ogbn-arxiv	**GCN**	**80**	5	68.5	71.8	72.0
			10	70.5	72.0	72.4
			20	70.9	72.2	72.5
			40	71.8	72.5	72.7
	APPNP	**40**	5	70.3	72.0	72.4
			10	70.5	72.2	72.4
			20	71.5	72.3	72.3
	GCNII	**40**	5	70.6	72.7	72.8
			10	72.0	72.7	72.8
			20	73.1	73.2	73.1