KDD 2024 Rebuttal Reviewer 1Qrn

April 12, 2024

1 Reviewer 1Qrn

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

1.1 Q1

The authors only study the staleness of GAS, while they claim this is a commom problem for various historical embedding methods, like VR-GCN and GraphFM. Similar studies on VR-GCN or GraphFM are expected in Section 3.

1.2 A1

Following your suggestion, we have included analyses similar to those in Section 3 of our submission for GraphFM. Due to time limitations, we will add analyses for VR-GCN in our revision after the rebuttal period.

From the results, we can make the similar conclusion as GAS case:

(1) In terms of convergence analysis in Figure 1 and 2, GraphSAGE shows slower convergence and inferior performance compared to GraphFM on the small dataset ogbn-arxiv when using a larger batch size. Conversely, GraphSAGE converges more swiftly and achieves superior performance compared to GraphFM in all other scenarios. Despite GraphFM's utilization of current one hop neighbors to mitigate staleness, its impact is limited, underscoring the necessity and importance of REST.

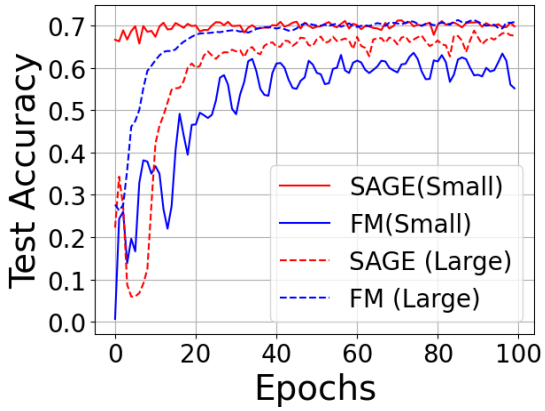

Figure 1: ogbn-arxiv

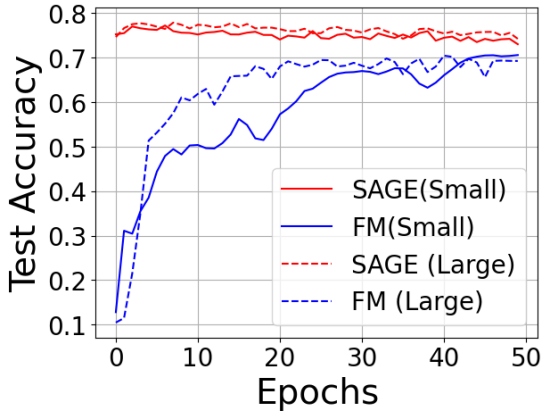

Figure 2: ogbn-products

(2) Regarding approximation errors in Figure 3 and 4, GraphFM also experiences staleness accumulation between each layer. While it can alleviate the approximation error introduced by staleness, it incurs additional approximation error by using biased one-hop neighbors for the momentum combination. This occurs because these nodes lose aggregations from their out-of-batch neighbors, exacerbating the overall approximation error.

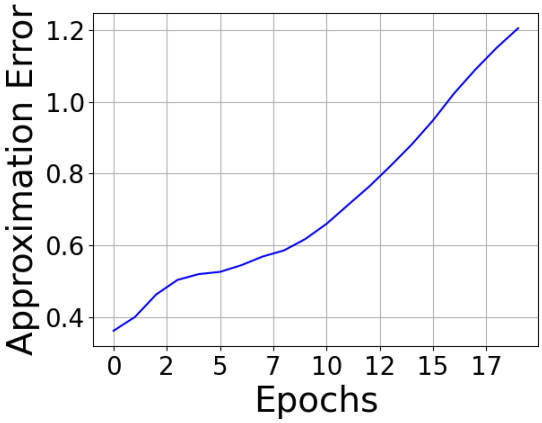

Figure 3: ogbn-arxiv

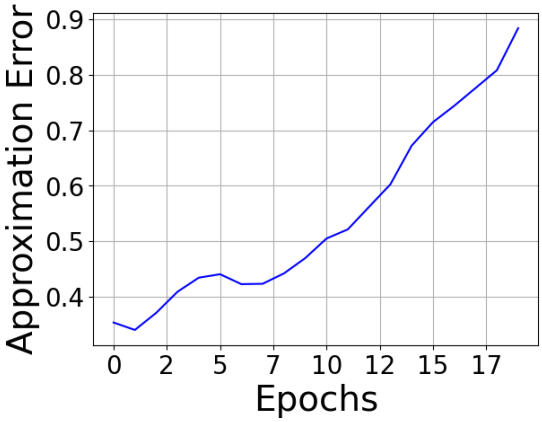

Figure 4: ogbn-products

1.3 Q2

As the goal of using historical embeddings is to reduce computation cost, while REST will introduce extra overhead as it will do F+1 forwards in one mini-batch. An ablation study about the relationship between the computation efficiency, converge time and the setting of F is expected.

1.4 A2

We present the relationship between computation efficiency and the setting of F in Table 1, accompanied by convergence curves corresponding to different frequencies in Figure 5 and 6. As observed from the results, higher frequencies tend to enhance convergence, as shown in Figure 5 (epoch as unit). However,

they also entail extra computation overhead. Hence, the actual convergence time remains relatively consistent across different frequencies, as illustrated in Table 1 and Figure 6 (time as unit). Nevertheless, all cases exhibit significant improvements compared to GAS.

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

Dataset	**Freduency**	MEMORY(MB)	TIME(S)
ogbn-products	2	9295	1053
	3	9295	1188
	4	9295	1200
	5	9295	1204

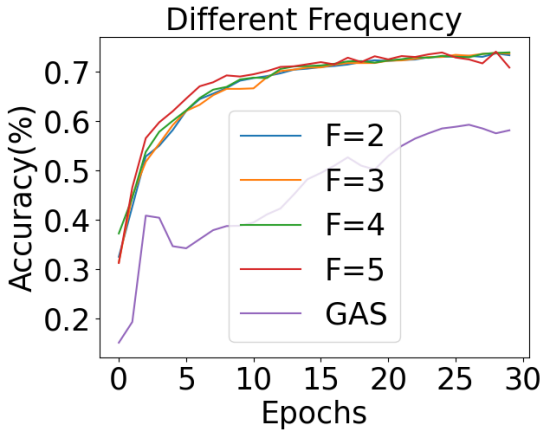

Figure 5: Convergence w.r.t epochs on the ogbn-products

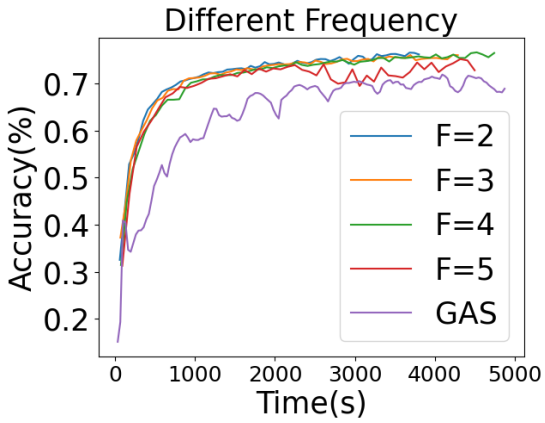

Figure 6: Convergence w.r.t time on ogbn-products

1.5 Q3

In Figure 5 and 6, a baseline model like SAGE or GCN are required. It is important to compare the convergence speed of GAS-REST with SAGE or GCN

1.6 A3

In our preliminary section, we offer a comparison with baseline models such as GraphSAGE. For reference, please refer Figure 1 in our submission, which depicts the performance over epochs for ogbn-arxiv and ogbn-products across different batch sizes.

To comprehensively address your concerns, we have included Figures 7 and 8, showcasing experiments on ogbn-arxiv with both small and large batch sizes, utilizing time as the unit of measurement. The results indicate that our model's convergence is nearly equivalent to that of GraphSAGE.

However, it's important to note that REST outperforms GraphSAGE in terms of performance and only requires a much smaller memory cost (please refer Table 2). Hence, REST demonstrates its advantage over GraphSAGE.

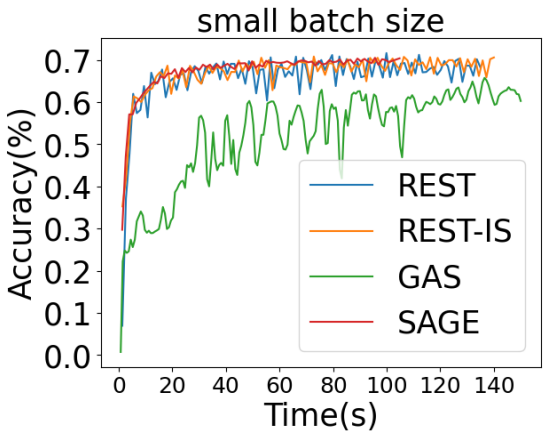

Figure 7: Convergence on ogbn-arxiv.

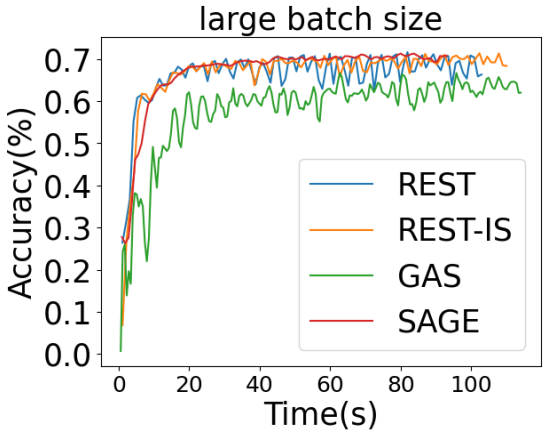

Figure 8: Convergence on ogbn-arxiv.

1.7 Q4

Table 5 does not include traditional GNN models like vanilla SAGE or vanilla APANN.

1.8 A4

We have included GraphSAGE in Table 5, as you suggested. We will add other traditional models after the rebuttal period. Please find the updated table below.

It's important to note that GraphSAGE may still encounter the neighbor explosion problem, leading to out-of-memory (OOM) errors for ogbn-products and significantly higher memory costs for ogbn-arxiv in our experiments. Similarly to the previous answer, REST exhibits a similar convergence rate to GraphSAGE but achieves better performance and much lower memory costs. These results highlight the advantages of our proposed method.

Table 2: Memory usage (MB) and running time (seconds) on ogbn-arxiv and ogbn-products.

Dataset	**Batch Size**	Memory(MB)				Time(s)				Accuracy			
		SAGE	GAS	REST	REST-IS	SAGE	GAS	REST	REST-IS	SAGE	GAS	REST	REST-IS
ogbn-products	5	OOM	8913	9295	10059	N/A	2600	1170	1312	N/A	75.2	79.7	80.5
	10	OOM	13406	13495	14753	N/A	1890	940	1200	N/A	76.9	80.0	80.4
ogbn-arxiv	5	3011	790	837	703	39.3	67.5	39	42	70.9	69.4	71.3	71.9
	10	3156	997	1115	948	37.2	54.0	31.5	33	71.2	70.1	72.3	72.4
	20	3323	1486	1505	1262	25.8	39.6	24.0	19.5	71.5	71.5	72.4	72.4

1.9 Q5

As the authors claims that REST is orthogonal to historical embedding, the performance of REST with GraphFM or others must be provided.

1.10 A5

Please refer to the Table 3 for the performance of GraphFM+REST. Compared with GAS, 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. Furthermore, our proposed method can be easily applied to GraphFM and achieve even better performance, highlighting the generality of REST.

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

1.11 Q6

What is the difference between REST and accumulating losses from multiple forwards and doing backward once?

1.12 A6

REST aims to refresh the memory bank to reduce staleness, whereas gradient accumulation simulates the effect of increasing the batch size. Hence, gradient accumulation introduces additional computation overhead. In terms of runtime, it prolongs the time since every forward pass requires gradient computation. Regarding memory usage, gradients must be stored for each forward pass until backward propagation is executed with accumulation. This accumulation of gradients across multiple forward passes can result in increased memory consumption, particularly with large models or datasets. To demonstrate the efficiency contrast between REST and gradient accumulation, we present the following table, showcasing the superior efficiency of our design. Note that REST can also be combined with gradient accumulation since they are orthogonal techniques. We leave this as a future work.

Table 4: Memory usage (MB) and running time (seconds) ogbn-products.

	Memory(MB)	Time/Epoch(s)
REST	9295	33
Gradient Accumulation	14537	39

1.13 Q7

Figure 9 and Figure 10 presents the converge curve of GAS and REST as Figure 5 and 6, but why they are different?

1.14 A7

The X-axis differs between them. Figure 5 and 6 utilize time (seconds) as the unit, while Figure 9 and 10 use epochs as the unit. Although the running time per epoch is longer for REST due to multiple forward propagations, REST requires significantly fewer epochs to converge, as demonstrated in Figure 9 and 10. Consequently, the overall running time is smaller than GAS, especially when the batch size is small (please refer to the first figure in Figure 5 and 6 in our submission). We showcase the advantage of our model in this comprehensive manner. Therefore, we claim that REST can achieve a double win in both performance and efficiency.

1.15 Q8

The caption of Figure 8 overlaps with Figures of Figure 7.

1.16 A8

Thanks for pointing out. We will make revision after the rebuttal period.