KDD 2024 Rebuttal Reviewer LzJ5

April 12, 2024

1 Reviewer LzJ5

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

1.1 Q1

A main concern is the novelty of the proposed REST. Although it is general, it is more like a training trick rather than a methodology.

1.2 A1

We respectfully disagree with the comment. Historical embedding methods have become crucial baselines in the existing large-scale GNN domain. While these methods are scalable, their impact is limited as they fail to address the bottleneck in performance and convergence. Although related works such as GraphFM or Refresh introduce technical modifications, their improvements are quite limited as they fail to address the primary concern of inferior performance and convergence.

We emphasize that the novelty extends beyond mere technical methodology changes. We provide a comprehensive analysis of staleness and inferior performance on large-scale problems, a research aspect that has not been explored before. Based on our analysis, we propose a general and straightforward training algorithm that efficiently addresses the bottleneck in all historical embedding approaches. Contrary to other fancy techniques, ours stands out in its simplicity and generality, which should be regarded as a merit.

Moreover, our proposed method can serve as a valuable asset for existing methods and future research endeavors, fostering advancements in large-scale GNN training.

1.3 Q2

Some clarifications are not clear.

1.4 A2

We are happy to clarify this:

(1) For lines 4-5: The sampling process in our training is iterative. We first sample F mini-batches $\mathbf{B}_1 \ldots \mathbf{B}_F$ and use them for F separate forward propagations respectively (as depicted by the blue ellipses in Figure 4). Then, we sample $\tilde{\mathbf{B}}$ (as indicated by the yellow ellipse in Figure 4) from another sampler. This operation provides us with flexibility to choose different batch sizes for $\mathbf{B}$ and $\tilde{\mathbf{B}}$.

(2) In REST-IS, for batch B, instead of randomly sampling in-batch nodes as done in REST, we specify that the in-batch nodes in batch B are the one-hop neighbors of batch $\tilde{B}$. In Formula 9, v represents an in-batch node of $\tilde{B}$, u is its one-hop neighbor, and w is the one-hop neighbor of u, which essentially represents the neighbor of v's neighbor. We utilize historical embeddings for those out-of-batch nodes w.

Thank you for pointing out our typo. We have updated formula 9 as follows to make our statement clearer:

$$
\begin{aligned}
h_v^{l+1} &= f_\theta^{(l+1)}(h_u^l, [h_u^l]_{u \in \mathcal{N}(v)}) && (1) \\
&= f_\theta^{(l+1)}(h_u^l, [h_u^l]_{u \in \mathcal{N}(v) \cap \tilde{B}} \cup [h_u^l]_{u \in \mathcal{N}(v) \setminus \tilde{B}}) && (2) \\
&\approx f_\theta^{(l+1)}\Bigg(h_u^l, [h_u^l]_{u \in \mathcal{N}(v) \cap \tilde{B}} \\
&\quad \cup \big[f_\theta^{(l+1)}(h_w^l, [h_w^l]_{w \in \mathcal{N}(u) \cap B} \cup \underbrace{[\bar{h}_w^l]_{w \in \mathcal{N}(u) \setminus B}}_{Historical})\big]_{u \in \mathcal{N}(v) \setminus \tilde{B}}\Bigg) && (3)
\end{aligned}
$$

1.5 Q3

The performance comparison and frequency analysis on different backbone models are missing.

1.6 A3

We opted for another backbone, GraphFM, to conduct performance and frequency analysis, as depicted in the following Table 1 and 2.

GraphFM exhibits a minor performance improvement through staleness reduction, yet it remains inadequate in achieving superior performance as it fails to fully tackle the staleness issue at its root, unlike REST. Conversely, our proposed method can be seamlessly applied to GraphFM, yielding even better performance, underscoring the generality of REST.

Furthermore, the frequency analysis indicates that higher frequencies tend to enhance performance, consistent with our claim in the submission.

Table 1: 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

Table 2: Frequency analysis with GraphFM on ogbn-products.

Dataset	**Freduency**	Acc
ogbn-products	2	80.0
	3	80.1
	4	80.4
	5	80.6

1.7 Q4

I wonder why REST can result in less running time than original methods (Tables 6 and 7). My understanding is the proposed REST will introduce more forward processes for the model. Please correct me if there has been a misunderstanding. Besides, the running time with different frequencies is missing.

1.8 A4

We appreciate the opportunity to clarify your question. As you mentioned, REST indeed incurs a longer running time per epoch due to the implementation of multiple forward propagations compared to original methods. However, it's noteworthy that REST requires significantly fewer epochs to converge, as illustrated in Figure 9 and 10 where epochs are used as the unit of measurement in the appendix. Consequently, the overall running time for REST is shorter than that of the original methods, as indicated in Figure 5 and 6 where time is used as the unit of measurement. Therefore, we claim that REST can achieve a double win in both performance and efficiency.

We present the following two convergence curves to illustrate the convergence concerning epochs and running time for different frequencies, following your suggestion. As observed from the results, higher frequencies tend to enhance convergence, as shown in Figure 1 (epochs as unit). However, they also entail extra computation overhead. Hence, the actual convergence time remains relatively consistent across different frequencies, as illustrated in Figure 2 (time as unit). Nevertheless, all cases exhibit significant improvements compared to GAS.

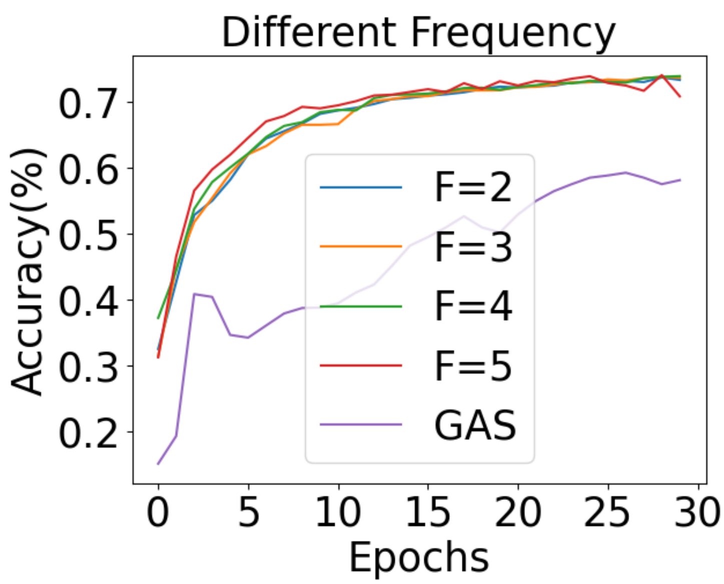

Figure 1: Convergence w.r.t epochs on the ogbn-products

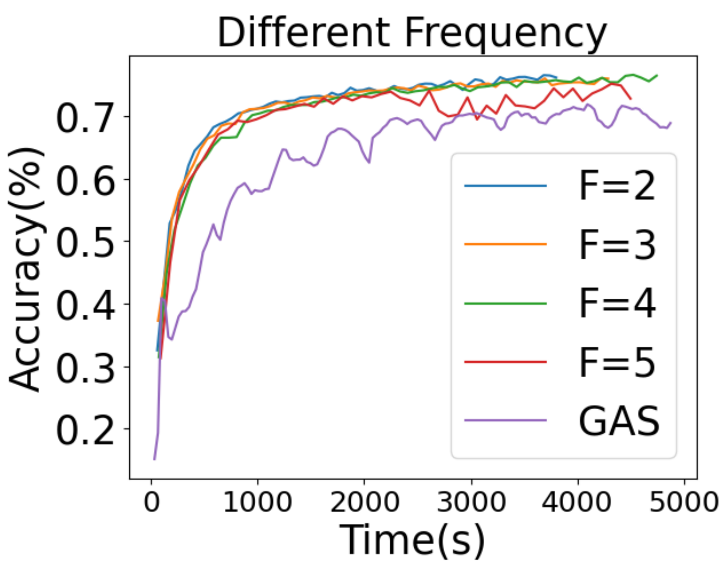

Figure 2: Convergence w.r.t time on ogbn-products