KDD 2024 Rebuttal Reviewer w4qu

April 12, 2024

1 Reviewer w4qu

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

1.1 Q1

In particular, historical embedding methods need to consume massive memory spaces to store node embeddings of all layers. Naturally, how can they scale to the setting with more than 100M nodes,>5 layers and embedding dimension of >256 ?

1.2 A1

We're pleased to address the query. In default settings, the historical embeddings of each layer are stored in the CPU rather than the GPU to alleviate memory constraints. However, for extremely large datasets like ogbn-papers100m, memory can be stored on disk instead of CPU memory to better accommodate memory limitations.

1.3 Q2

Missing discussion and comparison with two recent progresses [1,2].

[1] Hanqing Zeng, et al. Decoupling the depth and scope of graph neural networks. NeurIPS 2021.

[2] Zhihao Shi, et al. LMC: Fast training of GNNs via subgraph sampling with provable convergence. ICLR 2023

1.4 A2

We have included a performance comparison between REST, ShadowGNN, and LMC in Table 1 below. We use GCN as GNN backbone model and add GAS case in the table for your convenience. We will update our submission after the rebuttal period.

Table 1: Accuracy (%) on ogbn-arxiv and ogbn-products.

Models	OGBN-ARXIV	OGBN-PRODUCTS
Shadow	71.8	79.9
GAS	71.7	76.7
GAS+REST	72.2	79.6
LMC	71.4	77.5
LMC+REST	72.6	80.1

Given our focus on addressing the staleness problem in existing historical embedding methods, LMC stands out as one such method that harnesses historical gradients to expedite convergence. Due to the versatility of our model, it can be easily applied to LMC. Specifically, akin to the forward pass, we conduct backward propagation F times to refresh the gradient memory bank without updating model parameters. Subsequently, we execute a standard backward propagation to update the model parameters. We include additional analysis in this scenario, including efficiency analysis as follows. We set F=1 and use GCN as bakbone for simplicity. For LMC, we adhere to their experiment settings as outlined in their official repository. Our proposed method enhances performance, accelerates the training process, and maintains a comparable memory cost. All results underscore the versatility and benefits of our approach.

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

Models	OGBN-ARXIV			OGBN-PRODUCTS		
	Acc	Memory	Running time	Acc	Memory	Running time
LMC	71.4	558	66	77.5	10982	1520
LMC+REST	72.6	584	41	80.1	11139	925

1.5 Q3

Achieving an accuracy of 80.5% on ogbn-products is not a promising and persuasive results since many methods on leaderboard can achieve an accuracy up to 85%.

1.6 A3

Our research aims to address the staleness issue prevalent in current historical embedding methods. To ensure a fair comparison, we have selected several of the most representative SOTA models, such as GAS and GraphFM, as our baselines. It's important to note that none of these models achieve the same performance as ours. However, it's also crucial to acknowledge that performance may vary depending on factors like the choice of the base GNN models and any additional features or techniques employed (such as C&S, SCR, SLE, or LMs/LLMs as enhancers).

In our study, we don't use any extra enhancements or tricks, and sticking to base models like GCN, APPNP, and GCNII to maintain consistency across our baselines.Importantly, we emphasize that our proposed method is orthogonal to these base GNN models and enhancers/tricks. Due to time constraints, we leave further exploration of these areas for future work.