Staleness-based Subgraph Sampling for Training GNNs on Large-Scale Graphs

NEURAL INFORMATION PROCESSING SYSTEMS

NPGML @NeurIPS 2025

Meta

TEXAS A&M UNIVERSITY

Limei Wang[1*], Si Zhang[1], Hanqing Zeng[1], Hao Wu[1], Zhigang Hua[1], Kaveh Hassani[1], Andrey Malevich[1+], Bo Long[1+], and Shuiwang Ji[2]

[1] Meta, [2] Texas A&M University

* Worked done when the author was a PhD student at Texas A&M University and an intern at Meta

+ Worked done when the author at Meta

TL; DR: Design a simple yet effective subgraph sampling method for GNNs with historical embeddings [1][2][3]

- Based on theoretical analysis of the approximation error caused by using historical embeddings for out-of-batch neighbors
- Better performance compared to the default sampling method (as in Cluster-GCN[4])
- Do not bring additional computation overhead due to efficient staleness score calculation, improved re-sampling strategy, and faster training converge

Background - Training GNNs on large-scale graphs

- Mini-batch training is necessary
- Sampling-based methods are commonly used (our focus in this paper), basically we sample a subgraph as a mini-batch
  - There may (e.g. node/layer-wise sampling) or may not (e.g. cluster-based) be overlapping nodes between different subgraphs
- Another line of work is treating each node as an training example and do mini-batch training with fixed batch size, but each node is attached with fixed-dimensional information [5] (e.g. top-k ppr neighbors, top-k hop aggregated embeddings)

Background - GNNs with historical embeddings

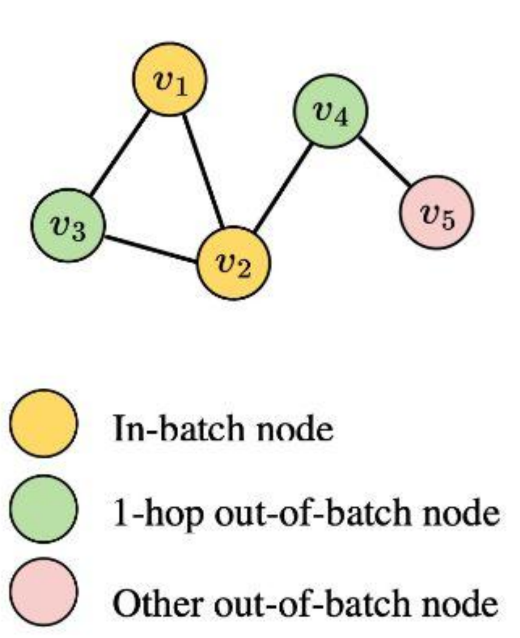
- Use historical embeddings for the unsampled neighbors

$$
\begin{aligned}
h_u^{l+1} &= f_\theta^{l+1}\left(h_u^l, \{h_v^l\}_{v\in\mathcal{N}(u)}\right)\\
&= f_\theta^{l+1}\left(h_u^l, \{h_v^l\}_{v\in\mathcal{N}(u)\cap\mathcal{B}} \cup \{h_v^l\}_{v\in\mathcal{N}(u)\setminus\mathcal{B}}\right)\\
&\approx f_\theta^{l+1}\left(h_u^l, \{h_v^l\}_{v\in\mathcal{N}(u)\cap\mathcal{B}} \cup \{\bar{h}_v^l\}_{v\in\mathcal{N}(u)\setminus\mathcal{B}}\right)
\end{aligned}
$$

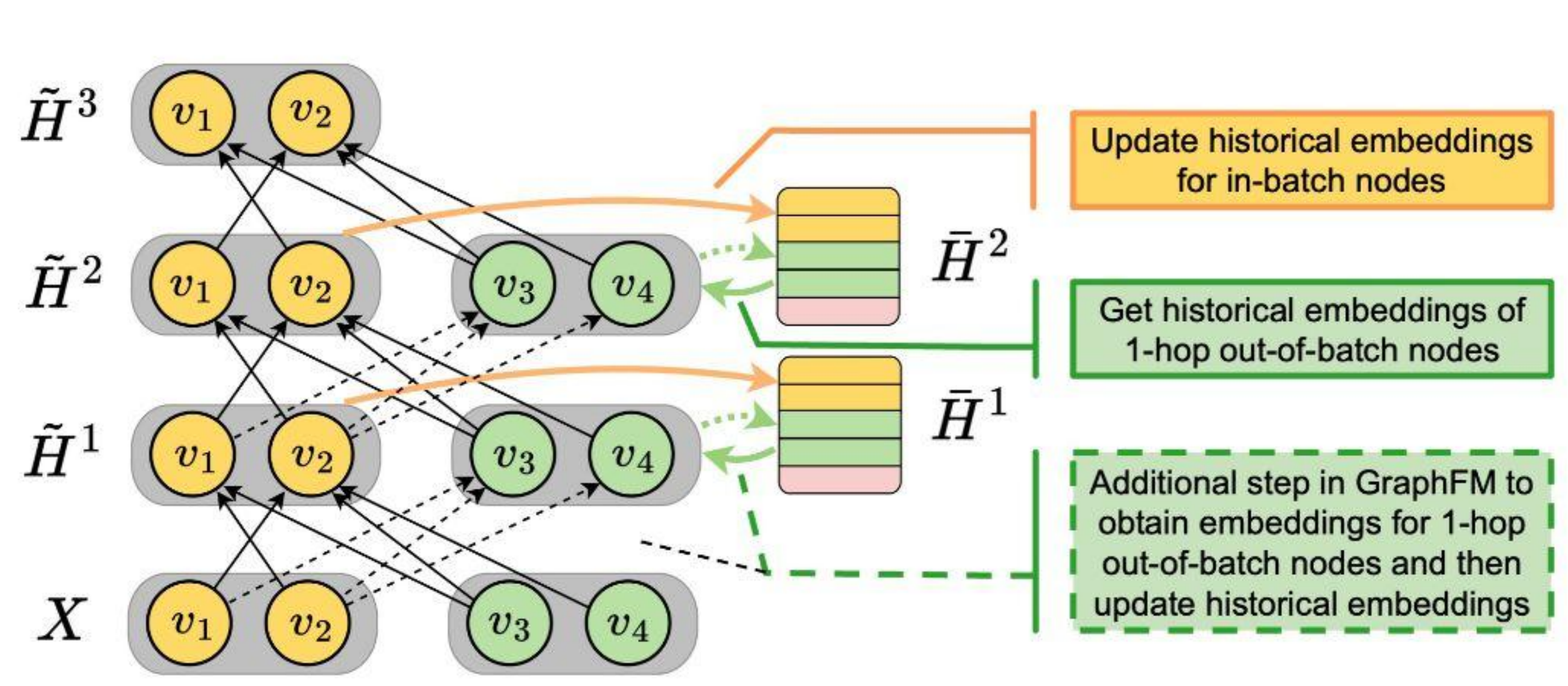
$v \in \mathcal{N}(u)\setminus\mathcal{B}$: unsampled neighbors

$v \in \mathcal{N}(u)\cap\mathcal{B}$: sampled neighbors

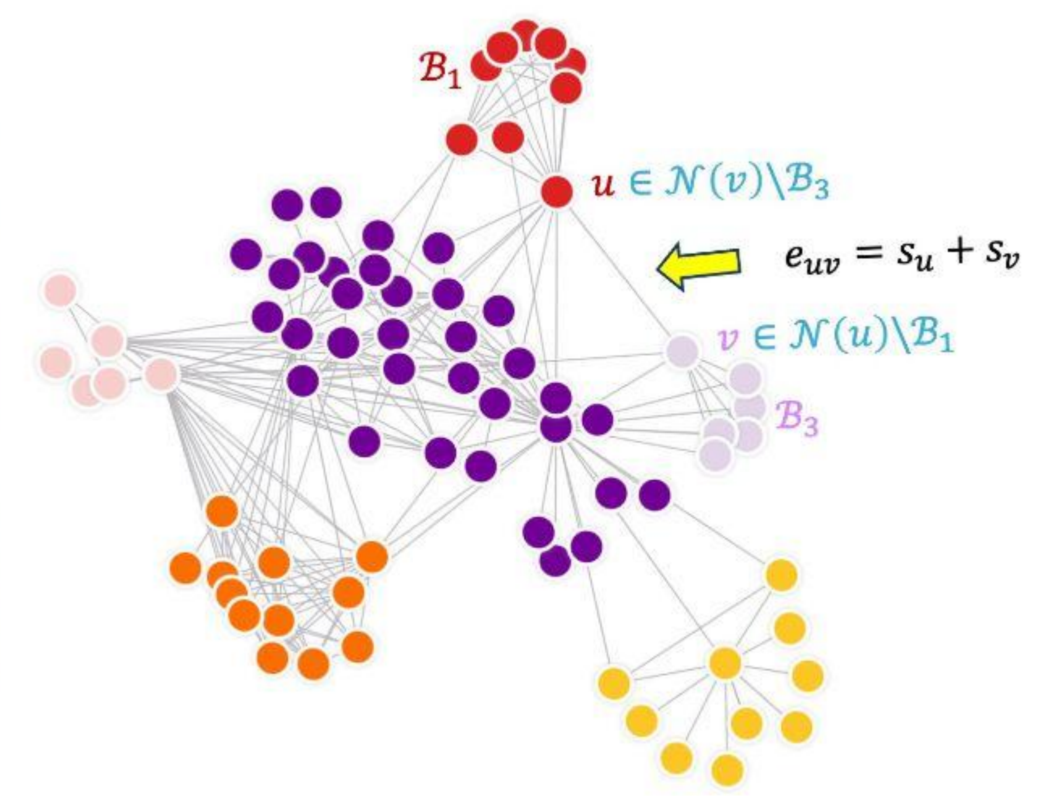
$\bar{h}_v^l$: historical embeddings



Example graph



Historical embedding-based methods



Staleness-based subgraph sampling

Method - Staleness-based subgraph sampling

- Minimize the appromization error from using historical embeddings

$$\|h_u^L - \tilde{h}_u^L\|_2^2$$

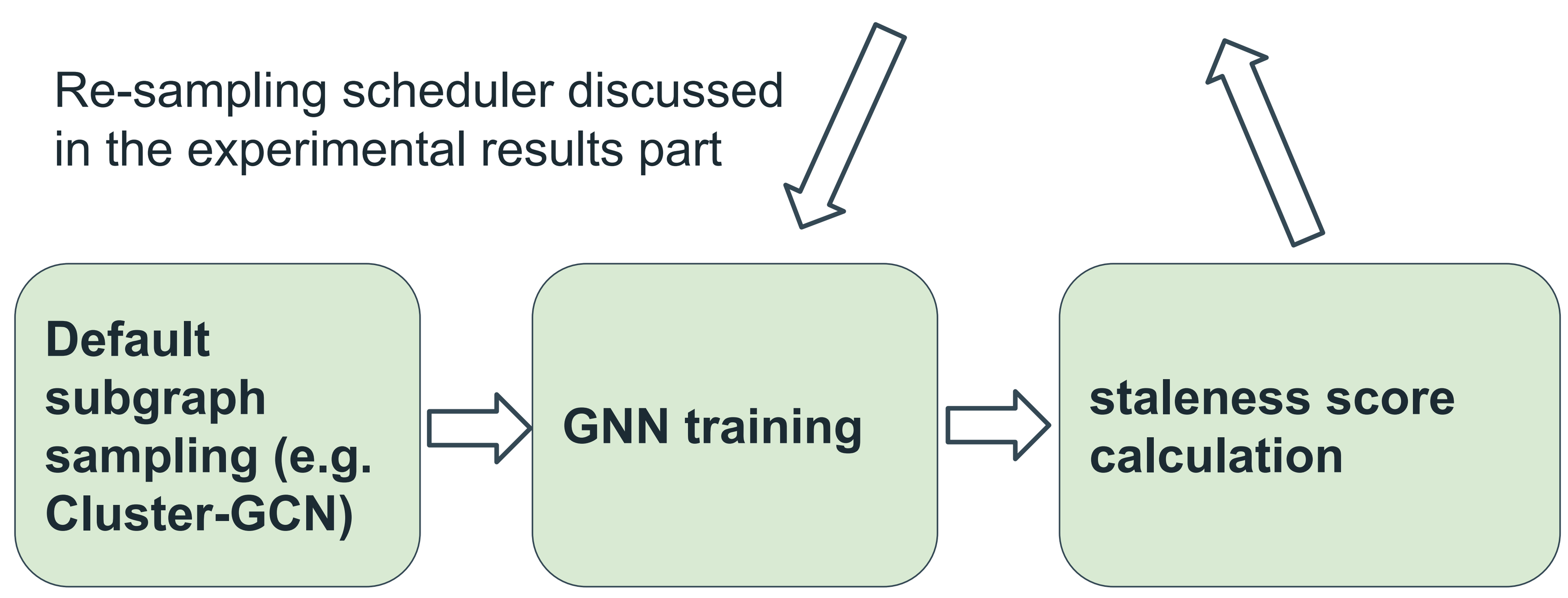
- Equivalent to minimize the weighted sum of the staleness scores for all out-of-batch neighbors $\arg\min_{\mathcal{B}} \sum_{u\in\mathcal{B}} \sum_{v\in\mathcal{N}(u)\setminus\mathcal{B}} \sum_{\ell} C_v^\ell s_v^\ell$

  where the staleness score $s_v^\ell = \|h_v^\ell - \bar{h}_v^\ell\|$
- Considering all M mini-batches, the overall minimization objective is

$$
\begin{aligned}
&\arg\min_{\{\mathcal{B}_1,\ldots,\mathcal{B}_M\}} \sum_{\mathcal{B}_i\in\{\mathcal{B}_1,\ldots,\mathcal{B}_M\}} \sum_{u\in\mathcal{B}_i} \sum_{v\in\mathcal{N}(u)\setminus\mathcal{B}_i} \sum_{\ell} C_v^\ell s_v^\ell\\
&\text{subject to} \quad \mathcal{V} = \mathcal{B}_1 \cup \mathcal{B}_2 \cup \ldots \cup \mathcal{B}_M\\
&\qquad \mathcal{B}_i \cap \mathcal{B}_j = \varnothing \quad \text{for all} \quad i \neq j, 1 \le i, j \le M
\end{aligned}
$$

- Equivalent to graph partitioning objective [6] (to the right)

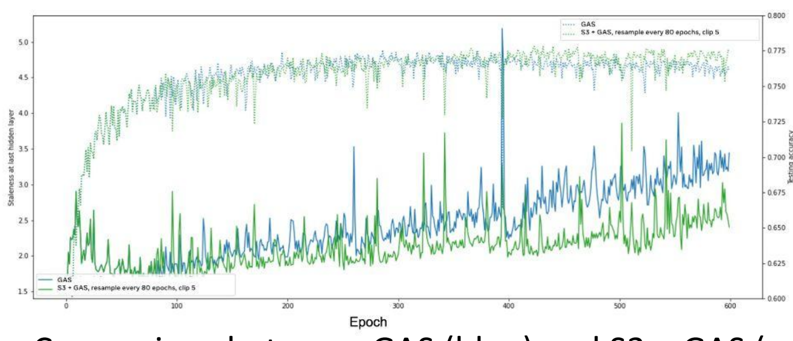
Re-sampling scheduler discussed in the experimental results part

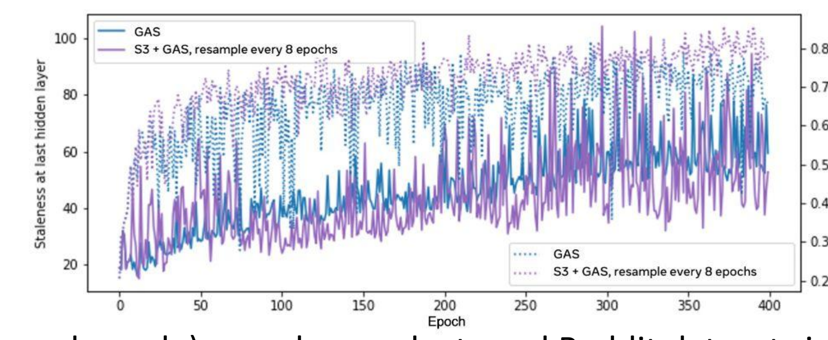


$$
\begin{aligned}
&\arg\min_{\{\mathcal{B}_1,\ldots,\mathcal{B}_M\}} \sum_{u\in\mathcal{B}_i, v\in\mathcal{B}_j, i\neq j, (u,v)\in\mathcal{E}} \sum_{\ell} C_u^\ell s_u^\ell + C_v^\ell s_v^\ell\\
&\text{subject to} \quad \mathcal{V} = \mathcal{B}_1 \cup \mathcal{B}_2 \cup \ldots \cup \mathcal{B}_M\\
&\qquad \mathcal{B}_i \cap \mathcal{B}_j = \varnothing \quad \text{for all} \quad i \neq j, 1 \le i, j, \le M
\end{aligned}
$$

Experimental results

# Nodes		89K	230K	717K	169K	2.4M
# Edges		450K	11.6M	7.9M	1.2M	61.9M
Method		Flickr	Reddit	Yelp	ogbn-arxiv	ogbn-products
VR-GCN		0.482 ± 0.003	0.964 ± 0.001	0.640 ± 0.002	–	–
FastGCN		0.504 ± 0.001	0.924 ± 0.001	0.265 ± 0.053	–	–
GraphSAINT		0.511 ± 0.001	0.966 ± 0.001	0.653 ± 0.003	–	0.791 ± 0.002
Cluster-GCN		0.481 ± 0.005	0.954 ± 0.001	0.609 ± 0.005	–	0.790 ± 0.003
SIGN		0.514 ± 0.001	0.968 ± 0.000	0.631 ± 0.003	0.720 ± 0.001	0.776 ± 0.001
GraphSAGE		0.501 ± 0.013	0.953 ± 0.001	0.634 ± 0.006	0.715 ± 0.003	0.783 ± 0.002
GCN	GAS	0.534 ± 0.001	0.954 ± 0.000	–	0.715 ± 0.002	0.767 ± 0.002
	S3 + GAS	0.545 ± 0.001	0.955 ± 0.000	–	0.724 ± 0.002	0.771 ± 0.002
GCNII	GAS	0.554 ± 0.003	0.967 ± 0.000	0.639 ± 0.003	0.725 ± 0.003	0.770 ± 0.002
	S3 + GAS	0.567 ± 0.002	0.969 ± 0.001	0.652 ± 0.003	0.735 ± 0.002	0.778 ± 0.002
GCN	FM	0.535 ± 0.002	0.953 ± 0.000	–	0.715 ± 0.003	0.767 ± 0.001
	S3 + FM	0.549 ± 0.001	0.952 ± 0.000	–	0.722 ± 0.002	0.770 ± 0.002
GCNII	FM	0.547 ± 0.003	0.965 ± 0.001	0.641 ± 0.003	0.725 ± 0.003	0.771 ± 0.002
	S3 + FM	0.566 ± 0.003	0.969 ± 0.000	0.652 ± 0.002	0.733 ± 0.003	0.776 ± 0.001
GCN	LMC	0.538 ± 0.001	0.954 ± 0.000	–	0.714 ± 0.002	0.765 ± 0.002
	S3 + LMC	0.541 ± 0.001	0.955 ± 0.000	–	0.721 ± 0.002	0.770 ± 0.002
GCNII	LMC	0.554 ± 0.005	0.969 ± 0.000	0.647 ± 0.003	0.728 ± 0.002	0.769 ± 0.002
	S3 + LMC	0.562 ± 0.002	0.969 ± 0.000	0.650 ± 0.003	0.731 ± 0.001	0.773 ± 0.002





Comparison between GAS (blue) and S3 + GAS (green and purple) on ogbn-products and Reddit datasets in terms of staleness scores (solid line) and testing accuracy (dashed line).

		Flickr & GCNII	ogbn-arxiv & GCNII
Epochs	LMC	356	197.4
	S3 + LMC	**211.4**	**180**
Runtime (s)	LMC	475	178
	S3 + LMC	**304**	**175**

Running time (s)	ogbn-arxiv	ogbn-products
Training per epoch	1	37
Staleness score calculation	0	3
Re-sampling from scratch	3	120
Re-sampling with fast refinement	2	48

S3 + GAS	No resampling	Resampling once	Resample every a fixed number of epochs 80	40	20	8	1
Flickr	0.5667	0.5683	**0.5729**	0.5711	0.5692	0.5715	0.5703
ogbn-arxiv	0.7250	0.7278	0.7291	0.7300	**0.7303**	0.7294	0.7292
ogbn-products	0.7991	0.8001	0.8030	**0.8069**	0.8035	–	–

[1] M. Fey et al. “GNNAutoScale: Scalable and expressive graph neural networks via historical embeddings.” ICML 2021

[2] H. Yu et al. “GraphFM: Improving large-scale gnn training via feature momentum.” ICML 2022

[3] Z. Shi et al. “LMC: Fast training of GNNs via subgraph sampling with provable convergence.” ICLR 2023

[4] W.-L. Chiang et al. “Cluster-GCN: An efficient algorithm for training deep and large graph convolutional networks.” KDD 2019

[5] Fu, Dongqi, et al. "VCR-Graphormer: A mini-batch graph transformer via virtual connections." ICLR 2024

[6] G. Karypis et al. “A fast and high quality multilevel scheme for partitioning irregular graphs.” SIAM Journal on scientific Computing 1998