Betty: Enabling Large-Scale GNN Training with Batch-Level Graph Partitioning and Tiered Memory

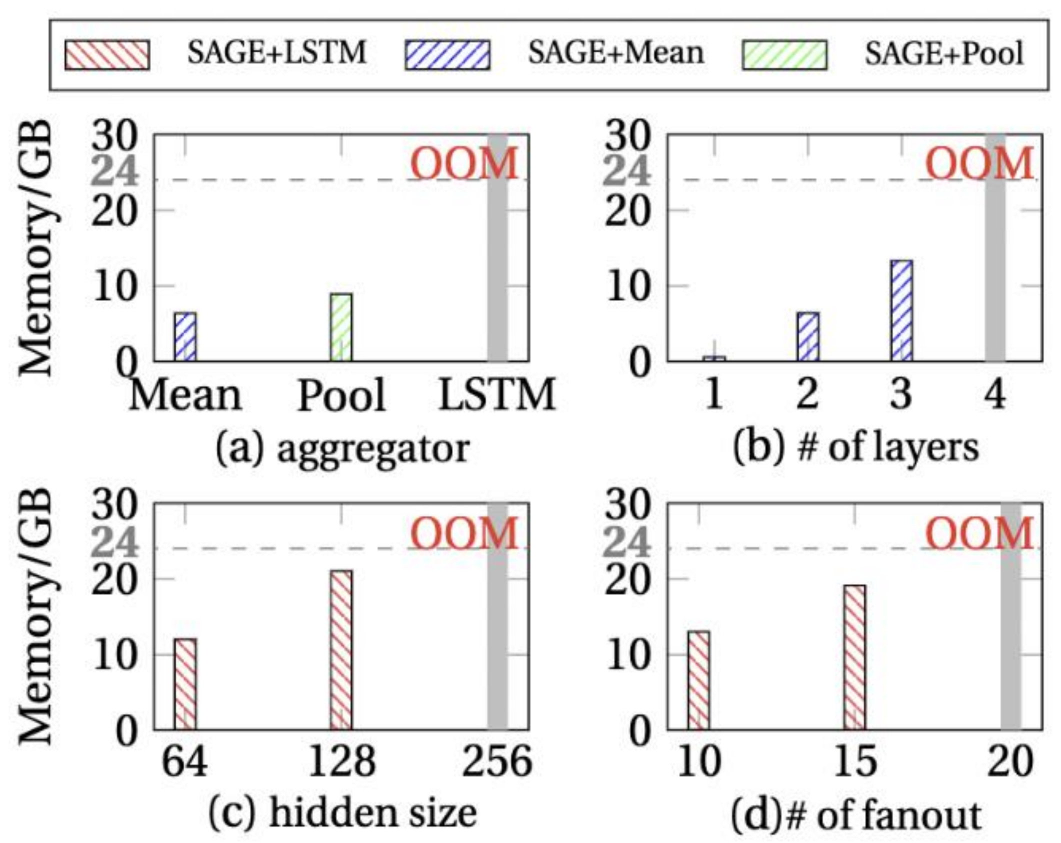
Shuangyan Yang[1], Minjia Zhang[2], Wenqian Dong[1,3] and Dong Li[1]

[1]University of California Merced, [2]Microsoft , [3]Florida International University

Motivation

Mitigate the memory bottleneck, and enable large-scale GNN training within a single GPU

❑ **Challenge** : **Easily exceeding GPU memory capacity.**



To work around the memory capacity bottleneck, prior work explored both algorithmic (sampling[1]) and system optimizations(DGL [2], PyTorch Geometric [3], and NeuGraph[4]).

❑ **Sampling**

➢ ***Pros***: Sample neighbors to compute the feature for a given node/subgraph.

reduce the sampling rate ⇨ reduce the number of neighbors participating in aggregation ⇨ reduce memory consumption

➢ ***Cons***: May cause loss of important neighbor information that hurts the final model accuracy.

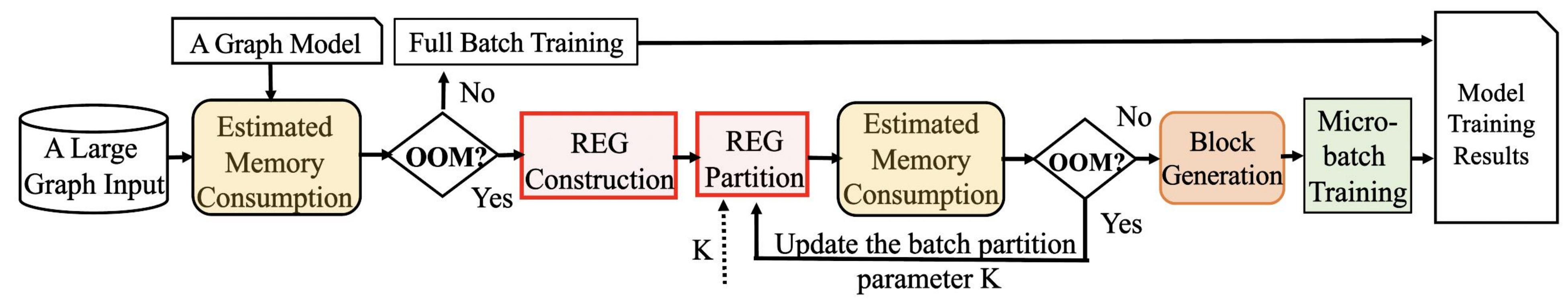
❑ **System optimizations**

➢ ***Pros***: support convenient and highly efficient graph operation primitives (e.g., aggregators) in terms of compute and memory efficiency.

➢ ***Cons***: GNN training can still run out of memory as more advanced configuration, especially when using more memory intensive aggregators.

Overview

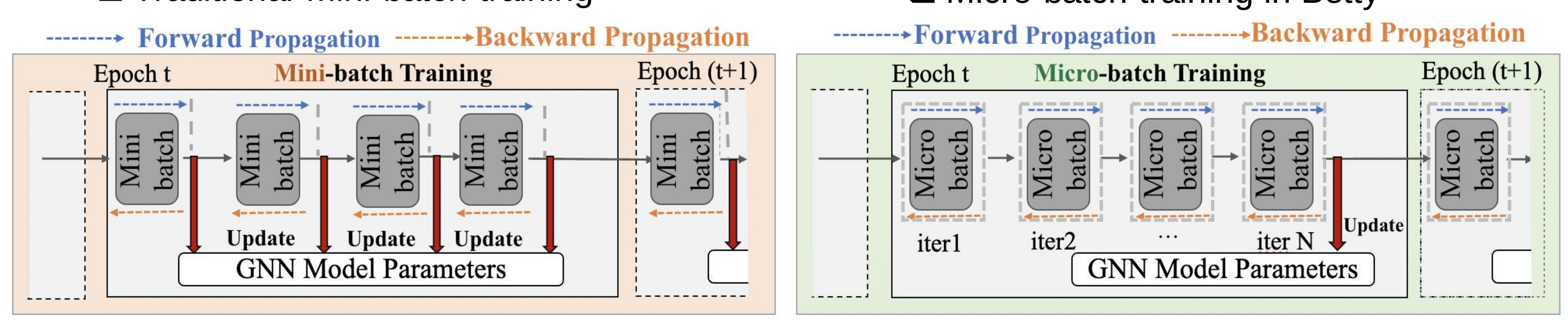
Betty introduces two novel techniques, redundancy-embedded graph (REG) partitioning and memory-aware partitioning.



Design

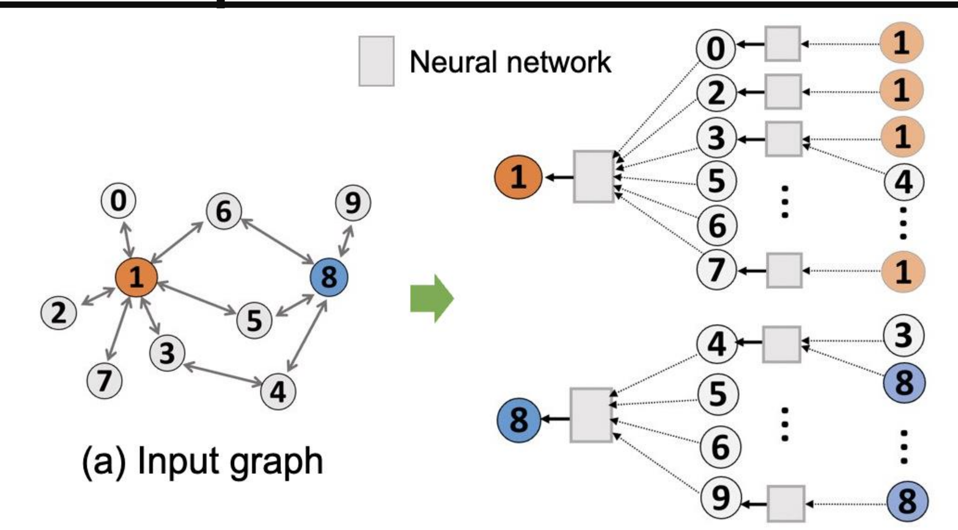
➢ **Batch-Level Partitioning:** reduces the memory consumption via the batch-level partitioning and using *both CPU and GPU memory* to enable training of advanced GNNs *on single GPU*.

❑ Traditional mini-batch training ❑ Micro-batch training in Betty



➢ **Partitioning the Multi-Level Bipartite for Micro–batch GNN Training**

Dividing each batch into K micro-batches, each micro-batch is still a hierarchical bipartite that is a subgraph of the original bipartite.



(a) Input graph

(b) Two-level bipartite structure

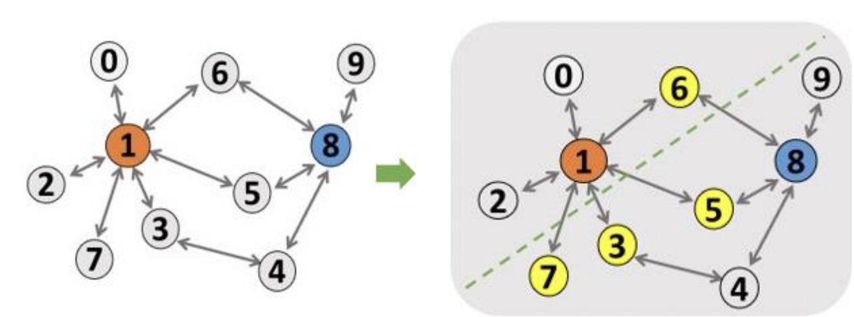
Full batch bipartite

K way partition

K micro-batch bipartite

➢ **Redundancy Reduction:**

Reduce the number of redundant node introduced by the partition of multi-level bipartite structure.



Partition : Redundant nodes {3,5,6,7}

➢ **Redundancy-Embedded Graph (REG) Construction and Partition**

➢ In REG output node u edge weight = # of shared neighbors output node v

➢ **Reducing Maximal Memory Footprint**

- Memory-aware Partitioning.
- Partition memory estimation

Experiment Results

Betty breaks the memory capacity constraint, reduce the peak memory consumption up to 48.3%.

➢ Dataset: Cora, Pubmed, Reddit, ogbn-arxiv and ogbn-products

➢ Baselines

- We evaluate the scalability of GNN training,
  - Aggregator
  - Number of model layers.
  - Hidden size
  - Fanout
- We use three common graph partition algorithms: range partition, random partition, and Metis[5]. (The partition is applied on the IDs of output nodes.)

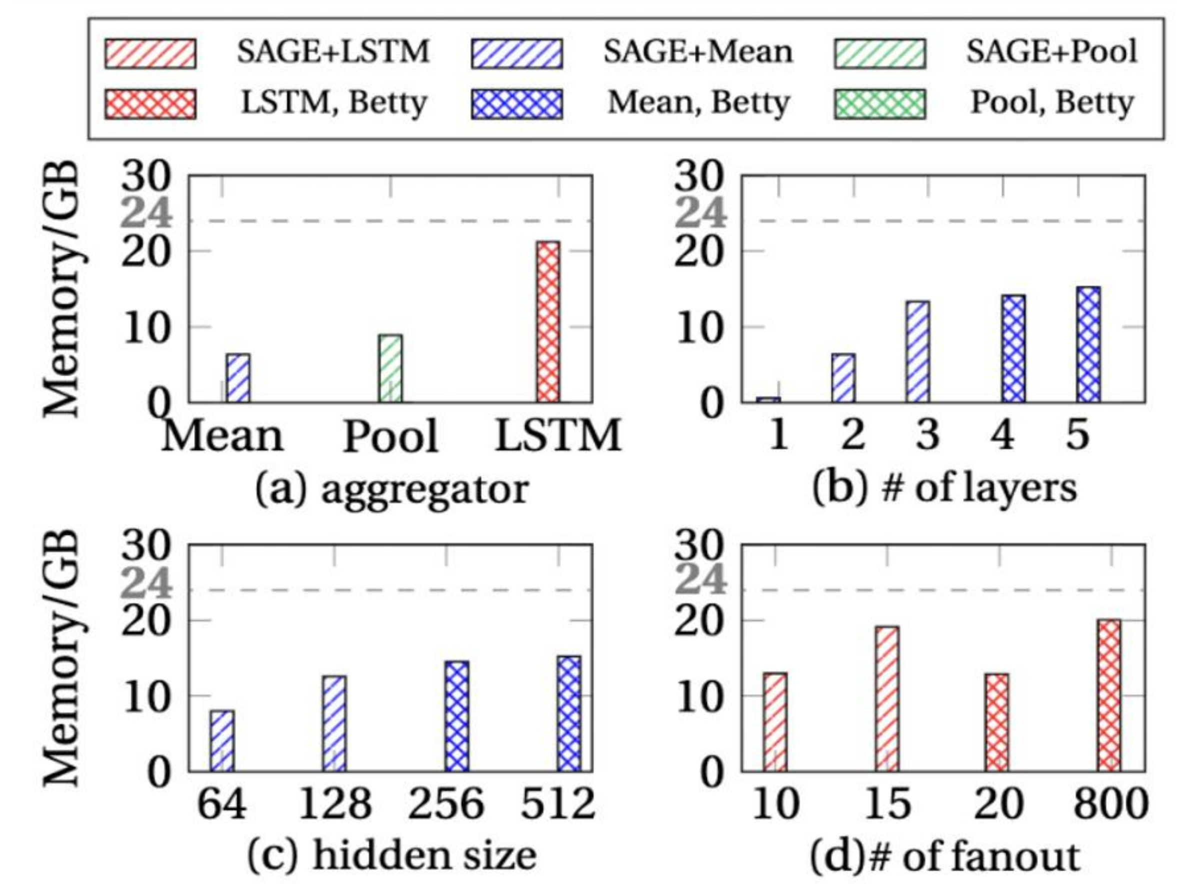
➢ Enable advanced and efficient GNN training with hybrid CPU-GPU memory.

➢ A transparent solution that does not require any hyperparameter tuning and preserve model convergence.

Open source

SCAN ME

For more details



➢ Compared with other graph partition methods, Betty can:

➢ reduce max memory consumption by 48.3% and 37.7% on average,

➢ reduce the node redundancy by up to 49.2% and 28.4% on average.

➢ improve computation efficiency by 20.6%, 21.1%, and 22.9%, when the number of batches increases (number of redundant nodes increases).

❑ References

[1] Da Zheng, Xiang Song, Chengru Yang, Dominique LaSalle, Qidong Su, Minjie Wang, Chao Ma, and George Karypis. Distributed hybrid cpu and gpu training for graph neural networks on billion-scale graphs. arXiv preprint arXiv:2112.15345, 2021

[2] DGL. Deep Graph Library. https://www.dgl.ai/

[3] PyG. PyTorch Geometric. https://pytorch-geometric.readthedocs.io.

[4] Lingxiao Ma, Zhi Yang, Youshan Miao, Jilong Xue, Ming Wu, Lidong Zhou, and Yafei Dai. {NeuGraph}: Parallel deep neural network computation on large graphs. In 2019 USENIX Annual Technical Conference (USENIX ATC 19), pages 443–458, 2019.

[5] George Karypis and Vipin Kumar. Metis–unstructured graph partitioning and sparse matrix ordering system, version 2.0. 1995