Efficient Methods for Training GNNs on Large Graphs

Sep. 2020

Dongyue Li

Real-World Graphs are Large

Table 3: Datasets used in system experiments. *Taobao-large* is six times large than *Taobao-small*.

Dataset	# user vertices	# item vertices	# user-item edges	# item-item edges	# attributes of user	# attributes of item
Taobao-small	147,970,118	9,017,903	442,068,516	224,129,155	27	32
Taobao-large	483,214,916	9,683,310	6,587,662,098	231,085,487	27	32

ZINC15

Welcome to ZINC, a free database of commercially-available compounds for virtual screening. ZINC contains over 230 million purchasable compounds in ready-to-dock, 3D formats. ZINC also contains over 750 million purchasable compounds you can search for analogs in under a minute.

ZINC is provided by the Irwin and Shoichet Laboratories in the Department of Pharmaceutical Chemistry at the University of California, San Francisco (UCSF). We thank NIGMS for financial support (GM71896).

To cite ZINC, please reference: Sterling and Irwin, *J. Chem. Inf. Model, 2015* *http://pubs.acs.org/doi/abs/10.1021/acs.jcim.5b00559*. You may also wish to cite our previous papers: Irwin, Sterling, Mysinger, Bolstad and Coleman, *J. Chem. Inf. Model, 2012* DOI: 10.1021/ci3001277 or Irwin and Shoichet, *J. Chem. Inf. Model. 2005;45(1):177-82* PDF, DOI.

Getting Started

- Getting Started
- What's New
- About ZINC 15 Resources
- Current Status / In Progress
- Why are ZINC results "estimates"?

Explore Resources

Chemistry
Tranches, Substances, 3D Representations, Rings, Patterns
And More
Catalogs, Genes, ATC Codes

Ask Questions

You can use ZINC for **general** questions such as

- How many substances in current clinical trials have PAINS patterns? (150)
- How many natural products have names in ZINC and are not for sale? (9296) get them as SMILES, names and calculated logP
- How many endogenous human metabolites are there? (47319) and how many of these can I buy? (8271) How many are FDA approved drugs? (94)
- How many compounds known to aggregate are in current clinical trials? (60)
- How many epigenetic targets have compounds known? (53) and Which of these substances can I buy? (278)
- How many ligands are there for the NMDA 1 ion channel GRIN1? (662) and How many of these are for sale? (60)
- More...

ZINC15 News

- 2018-02-14 - ZINC reaches 213,235,528 purchasable leadlike 3D!
- 2018-02-13 - ZINC reaches 736,001,654 purchasable molecules 2D!
- 2018-01-14 - Klara Anu is born! Welcome Klara Anu, sister to Lisa!
- 2018-01-01 - Chinzo Dandar joins our team. Welcome Chinzo! Follow us on twitter @chem4biology Known limitations What's new

Caveat Emptor: We do not guarantee the quality of any molecule for any purpose and take no responsibility for errors arising from the use of this database. ZINC is provided in the hope that it will be useful, but you must use it at your own risk.

Full Batch Training

- GCN (Kipf et al. ICLR 2017) in matrix operations

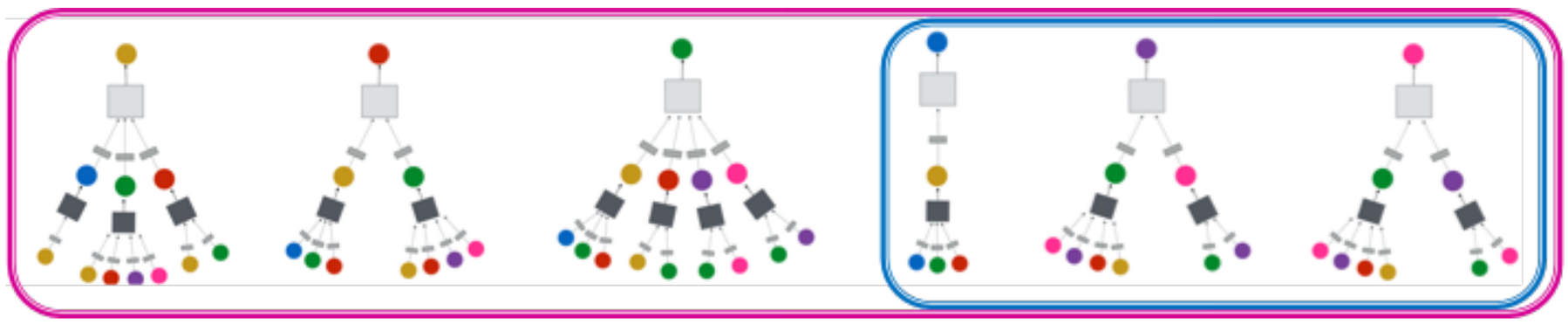
$$H^k = D^{-1/2} A D^{1/2} H^{k-1}$$

- Challenges:
  - Memory is not large enough to contain full matrix for large graphs
  - Full-batch training suffers from slow convergence rate (lack randomness)

Mini-Batch Training

- Utilize Stochastic Gradient Descent
  - Sample several nodes to compute loss function

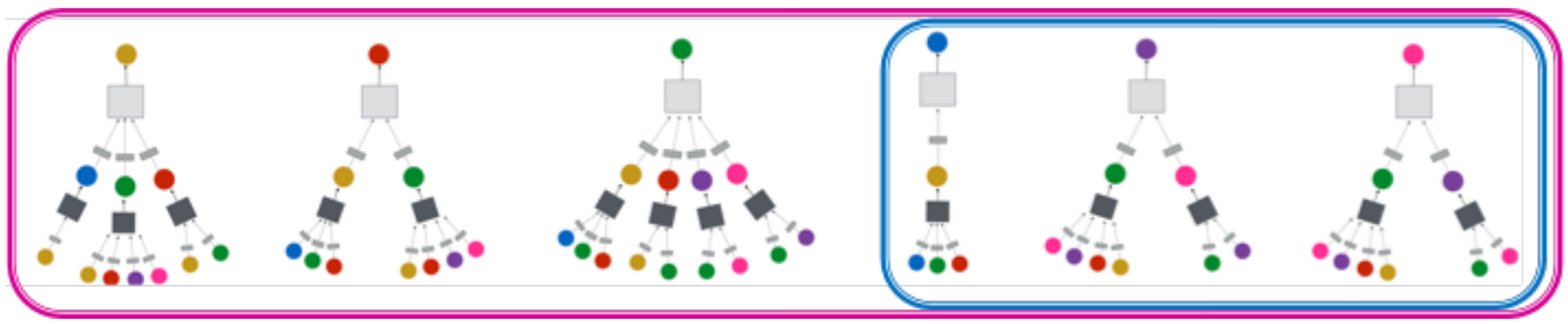
$$\nabla \mathcal{L} = \frac{1}{|\mathcal{V}_{\mathcal{L}}|} \sum_{v \in \mathcal{V}_{\mathcal{L}}} \nabla f(y_v, z_v^{(L)}) \approx \frac{1}{|\mathcal{V}_{\mathcal{B}}|} \sum_{v \in \mathcal{V}_{\mathcal{B}}} \nabla f(y_v, z_v^{(L)})$$



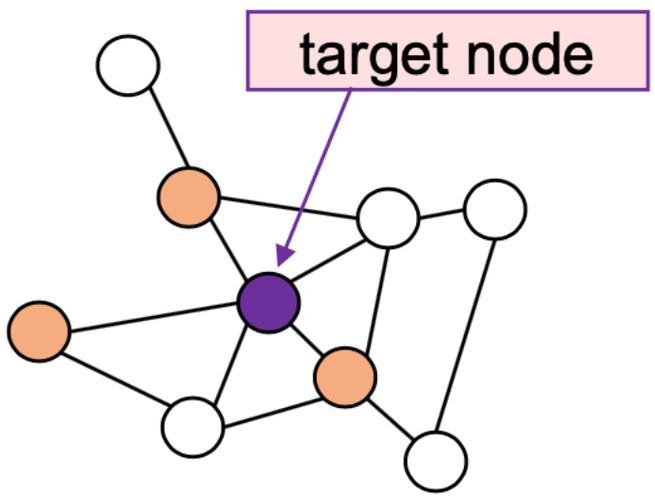
Vanilla Mini-Batch

Mini-Batch Training

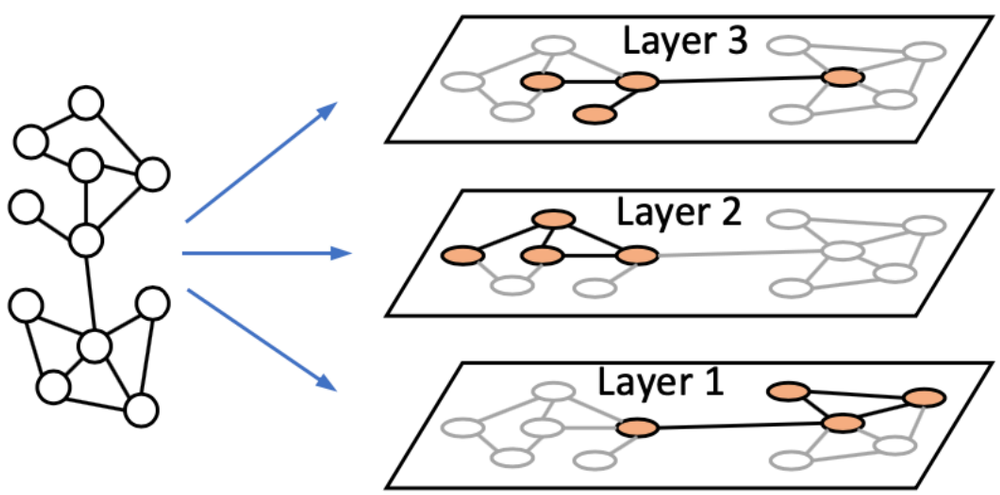
- Utilize Stochastic Gradient Descent
  - Too many nodes in one sample (**neighborhood expansion problem**)
    - One node samples nearly 2.4% nodes of the whole graph (NELL Dataset)
- Computes hidden layer feature repetitively (**Reuse Problem**)



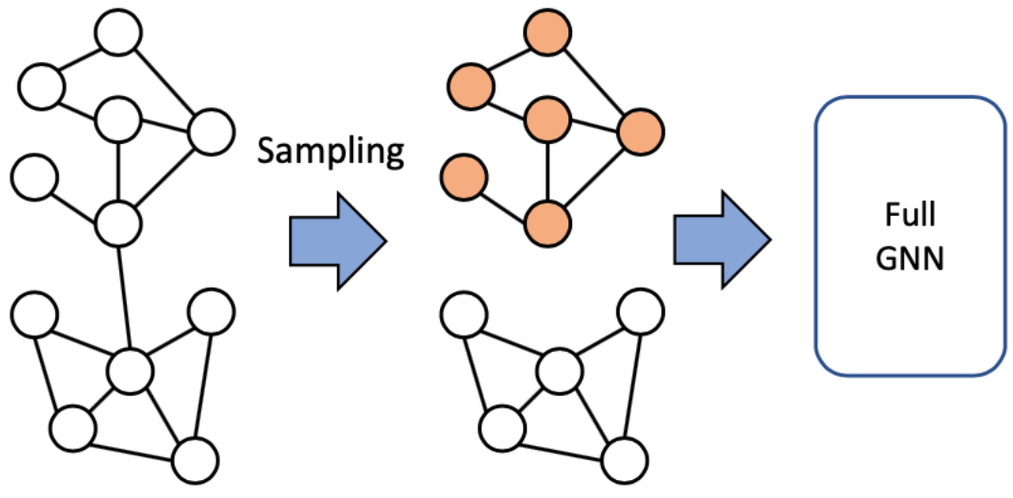
Three Sampling Paradigms



Node-wise Sampling



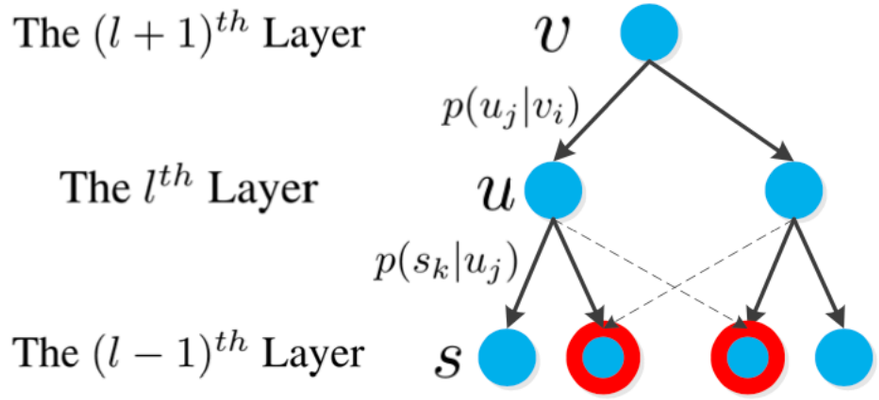
Layer-wise Sampling



Graph-wise Sampling

Node-Wise Sampling

- GraphSAGE (Hamilton et al. NIPS 2017)
  - Sample a fixed number of neighbors for each node



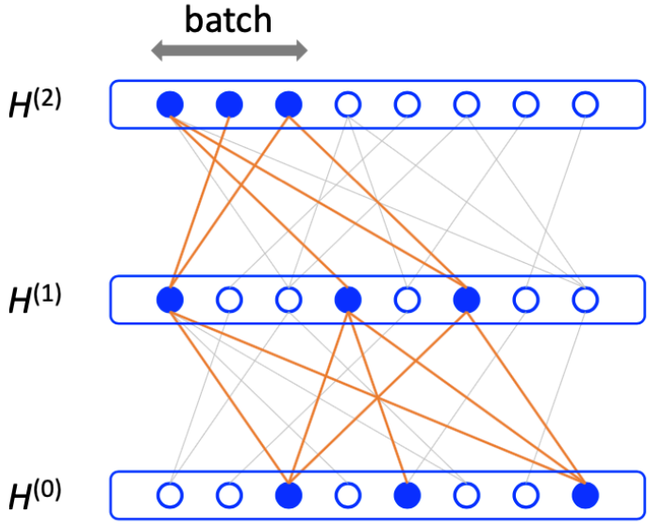
- VR-GCN (Chen et al. ICML 2018): buffer history features
  - A variance reduction method applied on GraphSAGE

Node-Wise Sampling

- Drawbacks
  - Node number in one sample grows exponentially
  - Still not solve the reuse problem

Layer-Wise Sampling

- Fast GCN (Chen et al. ICLR 2018)
  - Apply importance sampling on each layer



Graph convolution view

$$(PH^{(l)})_u = V \sum_{v=1}^{V} \frac{1}{V} P_{uv} h_v^{(l)} \approx \frac{V}{S} \sum_{v_s \sim q(v)} P_{uv} h_{v_s}^{(l)} / q(v_s)$$

Importance Probability

$$q(u) = \|\hat{A}(:,u)\|^2 / \sum_{u' \in V} \|\hat{A}(:,u')\|^2, \quad u \in V$$

- Nodes of consecutive layers may not be connected

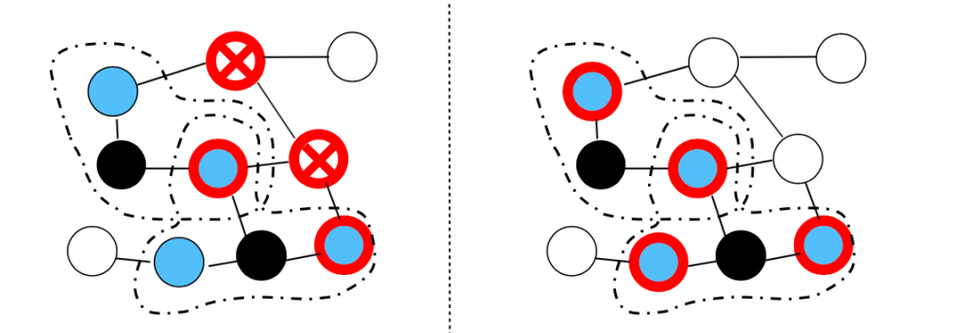
Layer-Wise Sampling

- Adaptive GCN (Huang[1] et al. NIPS 2018)
  - The lower layer is sampled based on the top one
  - Train to reduce variance explicitly in objective function

$$\mathcal{L} = \frac{1}{n}\sum_{i=1}^{n} \mathcal{L}_c(y_i, \bar{y}(\hat{\mu}_q(v_i))) + \lambda \mathrm{Var}_q(\hat{\mu}_q(v_i)),$$

Can be estimated by the sampled instances.

- Layer-Dependent GCN (Zou et al. NIPS 2019)
  - Only sample neighbors of last layer

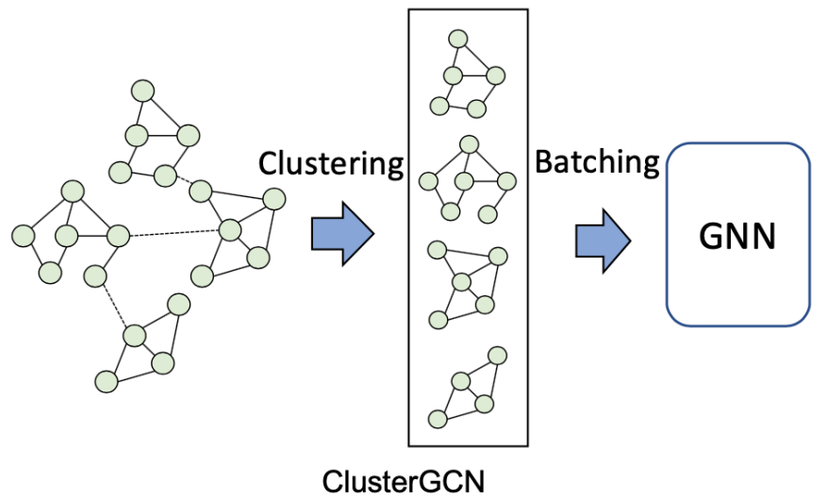


Layer-wise Importance Sampling (FastGCN)

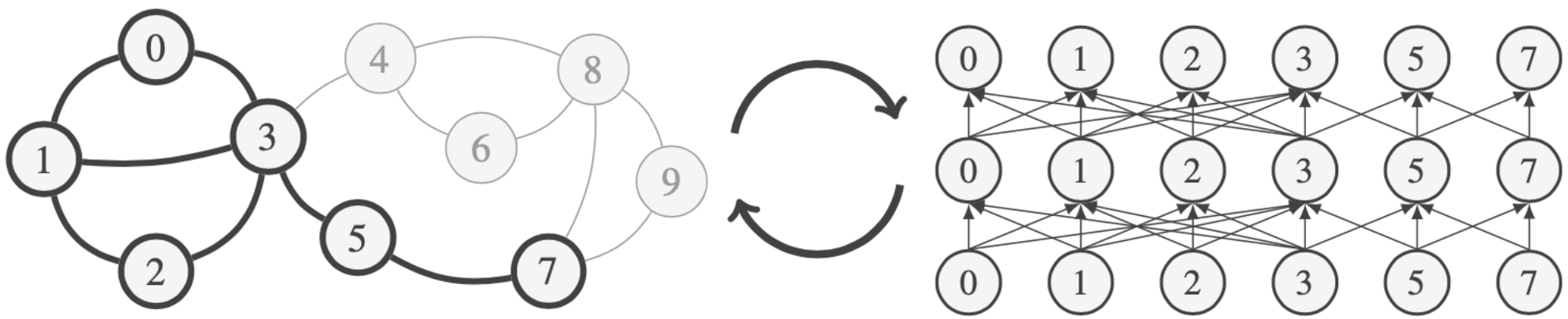
Layer-Dependent Importance Sampling (LADIES)

Graph-Wise Sampling

- Cluster-GCN (Chiang et al. KDD 2019)
  - Use clustered subgraphs for training
  - No theoretical guarantee



- GraphSAINT (Zeng et al. ICLR 2020) *State-of-Art
  - Sample a subgraph for training in full-batch manner



Summary

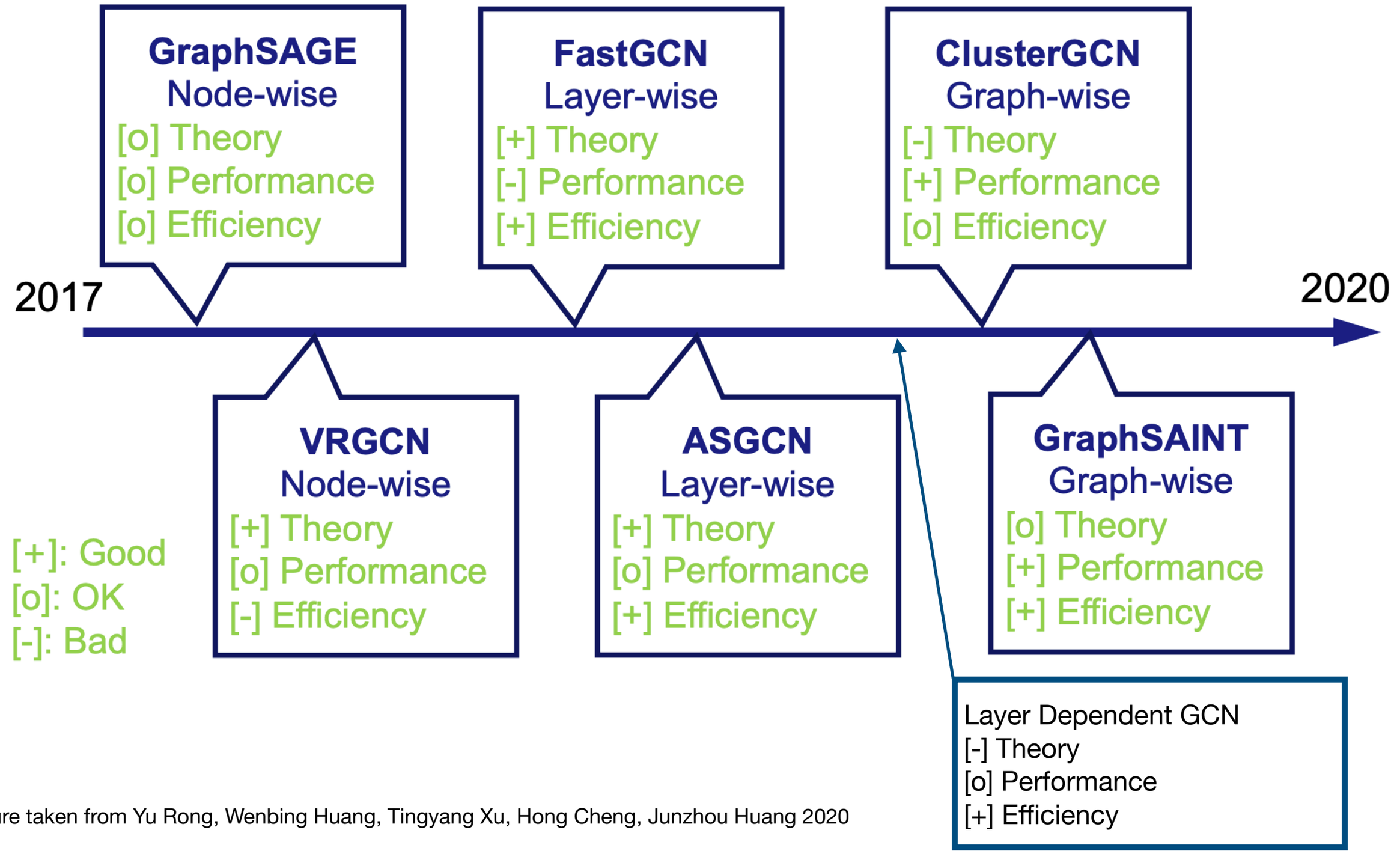


Figure taken from Yu Rong, Wenbing Huang, Tingyang Xu, Hong Cheng, Junzhou Huang 2020

Summary

	Full-Batch	Node-Wise	Layer-Wise	Graph-Wise
Memory	Bad	Good	Good	Good
Time per Epoch	Good	Bad	Good	Good
Convergence	Bad	Good	Good	Good

Note: Graph-wise sampling is a special layer-wise sampling method

Summary

Computation and Memory Complexity

Table 1: Time and space complexity of GCN training algorithms. L is number of layers, N is number of nodes, $\|A\|_0$ is number of nonzeros in the adjacency matrix, and F is number of features. For simplicity we assume number of features is fixed for all layers. For SGD-based approaches, b is the batch size and r is the number of sampled neighbors per node. Note that due to the variance reduction technique, VR-GCN can work with a smaller r than GraphSAGE and FastGCN. For memory complexity, LF^2 is for storing $\{W^{(l)}\}_{l=1}^{L}$ and the other term is for storing embeddings. For simplicity we omit the memory for storing the graph (GCN) or sub-graphs (other approaches) since they are fixed and usually not the main bottleneck.

	GCN [9]	Vanilla SGD	GraphSAGE [5]	FastGCN [1]	VR-GCN [2]	Cluster-GCN
Time complexity	$O(L\|\|A\|\|_0 F + LNF^2)$	$O(d^L NF^2)$	$O(r^L NF^2)$	$O(rLNF^2)$	$O(L\|\|A\|\|_0 F + LNF^2 + r^L NF^2)$	$O(L\|\|A\|\|_0 F + LNF^2)$
Memory complexity	$O(LNF + LF^2)$	$O(bd^L F + LF^2)$	$O(br^L F + LF^2)$	$O(brLF + LF^2)$	$O(LNF + LF^2)$	$O(bLF + LF^2)$

Graph-sampling is most memory-effective

Variance Analysis

Table 2: Summary of Complexity and Variance [6]. Here ϕ denotes the upper bound of the ℓ_2 norm of embedding vector, $\Delta\phi$ denotes the bound of the norm of the difference between the embedding and its history, D denotes the average degree, b denotes the batch size, and $\bar{V}(b)$ denotes the average number of nodes which are connected to the nodes sampled in the training batch.

Methods	Memory Complexity	Time Complexity	Variance
Full-Batch [12]	$\mathcal{O}(L\|\mathcal{V}\|K + LK^2)$	$\mathcal{O}(L\|\|\mathbf{A}\|\|_0 K + L\|\mathcal{V}\|K^2)$	0
GraphSage [9]	$\mathcal{O}(bKs_{node}^{L-1} + LK^2)$	$\mathcal{O}(bKs_{node}^{L} + bK^2 s_{node}^{L-1})$	$\mathcal{O}\big(D\phi\|\mathbf{P}\|_F^2/(\|\mathcal{V}\|s_{node})\big)$
VR-GCN [3]	$\mathcal{O}(L\|\mathcal{V}\|K + LK^2)$	$\mathcal{O}(bDKs_{node}^{L-1} + bK^2 s_{node}^{L-1})$	$\mathcal{O}\big(D\Delta\phi\|\mathbf{P}\|_F^2/(\|\mathcal{V}\|s_{node})\big)$
FastGCN [4]	$\mathcal{O}(LKs_{layer} + LK^2)$	$\mathcal{O}(LKs_{layer}^2 + LK^2 s_{layer})$	$\mathcal{O}(\phi\|\mathbf{P}\|_F^2/s_{layer})$
LADIES	$\mathcal{O}(LKs_{layer} + LK^2)$	$\mathcal{O}(LKs_{layer}^2 + LK^2 s_{layer})$	$\mathcal{O}\big(\phi\|\mathbf{P}\|_F^2\bar{V}(b)/(\|\mathcal{V}\|s_{layer})\big)$

Potential Directions

- Extend efficient training for more powerful models
- Apply pruning method to graph sampling
  - Make sampled node grows faster than linear but slower than exponential (performance will presumably be better)
- Accelerators for training on large graphs with subgraph sampling