NEURAL INFORMATION PROCESSING SYSTEMS

Decoupling the Depth and Scope of Graph Neural Networks

Hanqing Zeng[1], Muhan Zhang[2], Yinglong Xia[3], Ajitesh Srivastava[1], Andrey Malevich[3], Rajgopal Kannan[4], Viktor Prasanna[1], Long Jin[3], Ren Chen[3]

1. USC 2. PKU 3. Facebook AI 4. US-ARL

NeurIPS 2021

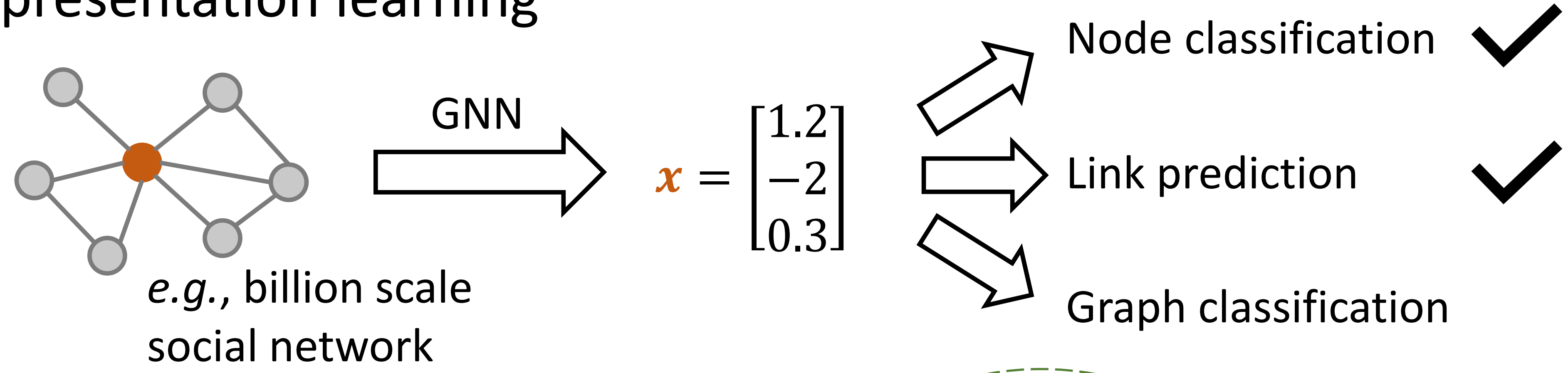
https://github.com/facebookresearch/shaDow_GNN

Outline

- Background
- Depth-scope decoupling
- Theoretical justifications
- Architecture designs
- Evaluation
- Conclusion

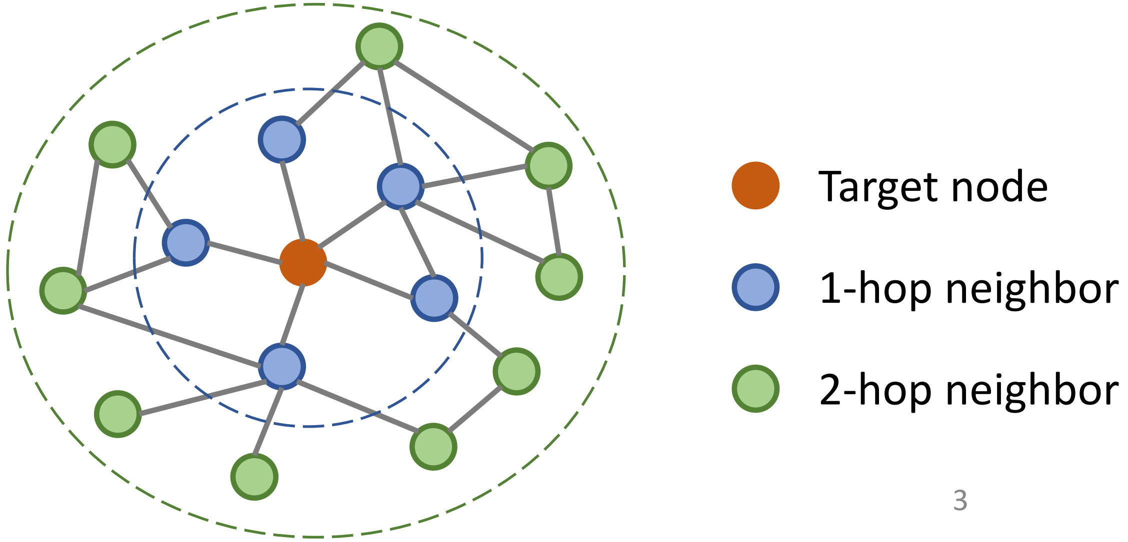
Background: Graph Neural Networks

Graph representation learning



Message passing in GNNs

- **Scope**: from what neighbors?
- **Depth**: by how many iterations / layers?



Scalability & Expressivity Challenges

GNN designs by default (on large scale graphs):

Increasing the **depth** (L-layer GNN) ⟺ Expanding the **scope** (L-hop neighborhood)

Neighborhood explosion $\sim d^L$

Dilemma in deep GNN: *scalability-expressivity tradeoff*

- **Depth is important**: Experience from general deep learning
- **Depth is expensive**: Observation from graph message passing
- **Depth can cause training challenges**: Oversmoothing in GCN

Solution: Don’t forget the scope!

Depth-Scope Decoupling

Define ***scope*** independent of ***depth***

- Intuitions
  - Some neighbors are irrelevant → no need to pass their messages
  - Some neighbors are extra important → worth passing their messages many times
- Example: Deep (L'-layer) GNN on shallow (L-hop) subgraph, $L' > L$

Algorithm: generate embedding for a target node v of the full graph $\mathcal{G}$

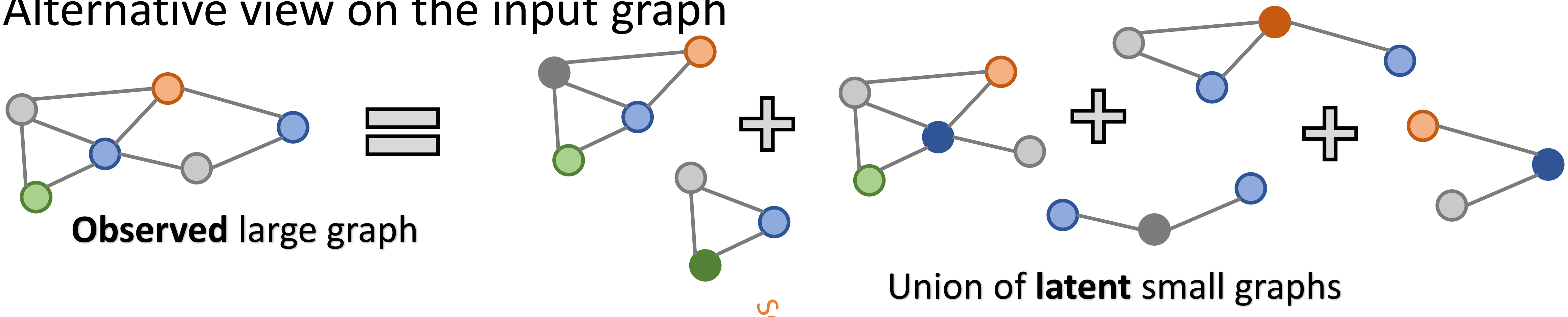
1. Extract a subgraph $\mathcal{G}_{[v]}$ around v
2. for round $i = 1$ to L':
   Perform message passing along all edges in $\mathcal{G}_{[v]}$
3. Take v's embedding from all node embeddings of $\mathcal{G}_{[v]}$

Depth-Scope Decoupling

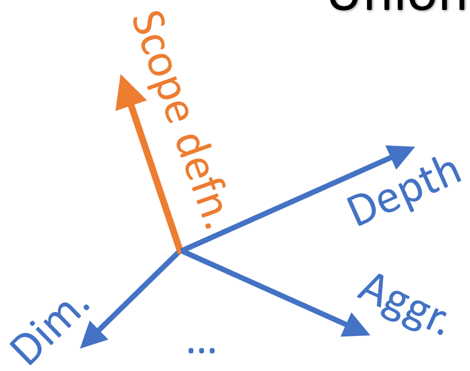
Interpretation

Scope
Depth
is a property of the
Data
Model

Alternative view on the input graph



Enlarging the GNN design space



Theoretical Justifications: Overview

Decoupling improves GNN expressive power, from the perspectives of

- Graph signal processing: decoupled-GCN avoids oversmoothing
- Function approximator: decoupled-SAGE learns target function better
- Topological information: decoupled-GIN exceeds 1-WL test

Decoupling improves GNN scalability

- Deep model + Large graph $\neq$ Exploding scope
- With fixed-size scope, complexity is linear with the model depth

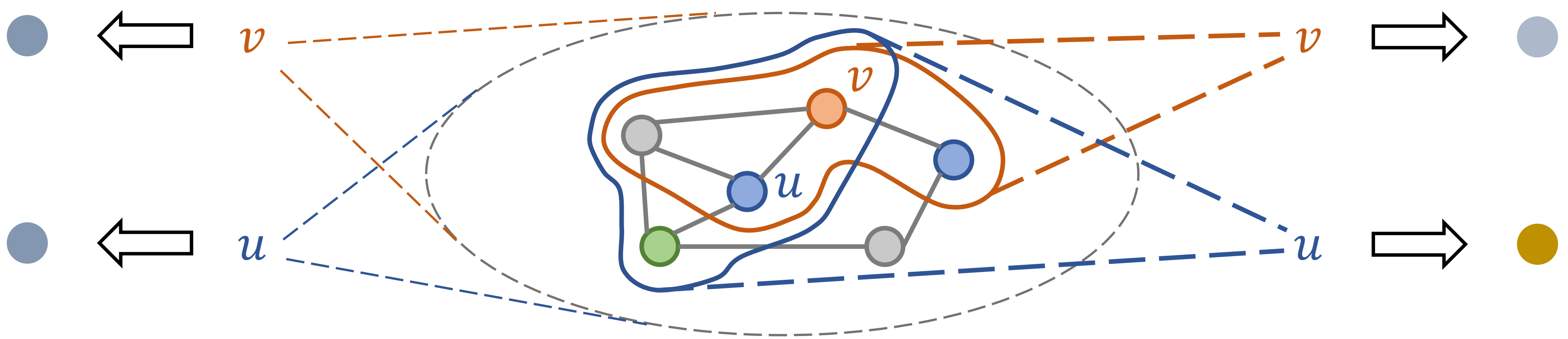
Theoretical Justification: Graph Signal Processing Perspective

Oversmoothing of deep GCN

- 1 layer: smoothing of direct neighbors
- Many layers: smoothing within the whole connected component (CC)
- ∞ layers: embedding only contains global info. of CC → indistinguishable

Local-smoothing of decoupled-GCN

- Scope is fully customized: $\mathcal{G}_{[u]} \neq \mathcal{G}_{[v]}$
- Many layers: smoothing within target node's own scope
- ∞ layers: different scope → distinctive embeddings



Theoretical Justification: Function Approximator Perspective

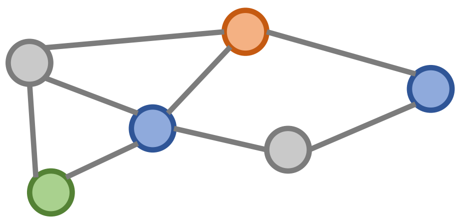
Decoupled-SAGE is more expressive than GraphSAGE

Consider neighborhood $\mathcal{G}'$ & function τ for linear comb. of $\mathcal{G}'$ features

- GraphSAGE cannot approximate τ well, even if $\mathcal{G}'$ is L-hop neighborhood
- Decoupled-SAGE can approximate τ where

Scope $\mathcal{G}_{[v]} = \mathcal{G}'$

Depth reduces the error exponentially



Neighborhood $\mathcal{G}'$

Normalized adj. matrix

Markov transition matrix

$$\widehat{\boldsymbol{A}} = \begin{bmatrix} 0 & 1/3 & 1/3 & 1/3 & 0 & 0 \\ 1/2 & 0 & 1/2 & 0 & 0 & 0 \\ 1/4 & 1/4 & 0 & 1/4 & 1/4 & 0 \\ 1/2 & 0 & 1/2 & 0 & 0 & 0 \\ 0 & 0 & 1/2 & 0 & 0 & 1/2 \\ 0 & 1/2 & 0 & 0 & 1/2 & 0 \end{bmatrix}$$

Deep decoupled-SAGE

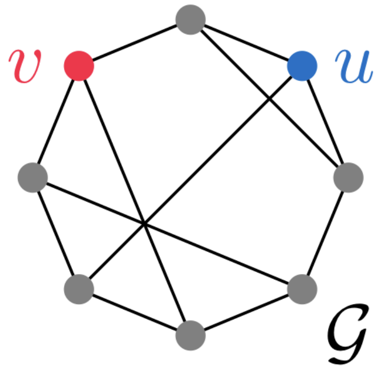
Markov chain convergence theorem

$\lim_{\ell \to \infty} \widehat{\boldsymbol{A}}^{\ell} \boldsymbol{X} \Rightarrow$ Approximate τ

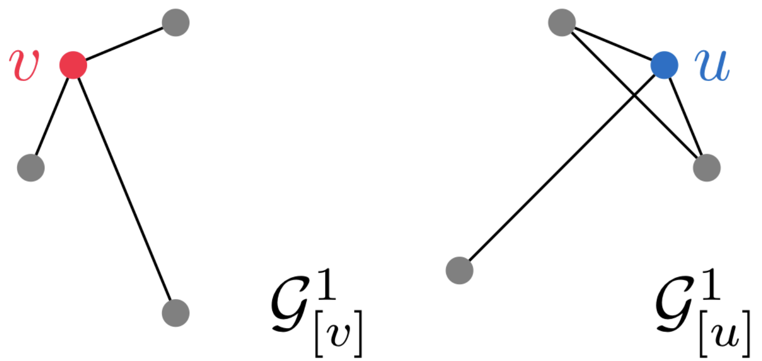
Theoretical Justification: Topology Information Perspective

Decoupled-GIN is more expressive than GIN/1-WL

- Challenge for GIN/1-WL: non-isomorphic regular graphs
- Benefit of decoupling: **subgraphs of a regular graph may not be regular**



Example 3-regular graph where GIN cannot distinguish u and v



Scope = 1-hop

Depth = 2

Decoupled-GIN can distinguish u and v

Architecture: Subgraph Extraction

Define scope $\mathcal{G}_{[v]}$ by extracting subgraphs around v

General approaches to preserve neighborhood characteristics

Heuristic based	Model based	Learning based

Example heuristic-based extraction function

- Identify important neighbors by Personalized PageRank (PPR) scores

```
1. Compute PPR score with target v as the root node
2. Take B neighbors N[v] with top PPR scores
3. Construct node-induced subgraph G[v] from N[v]
```

Architecture: READOUT & Ensemble

READOUT for node-& link-level tasks

- Two L-hop neighbors may only talk to each other after $2L$ layers
- Deep layers on shallow scope: each $\mathcal{G}_{[v]}$ node sees the complete $\mathcal{G}_{[v]}$ info.

→ READOUT all $\mathcal{G}_{[v]}$ embeddings

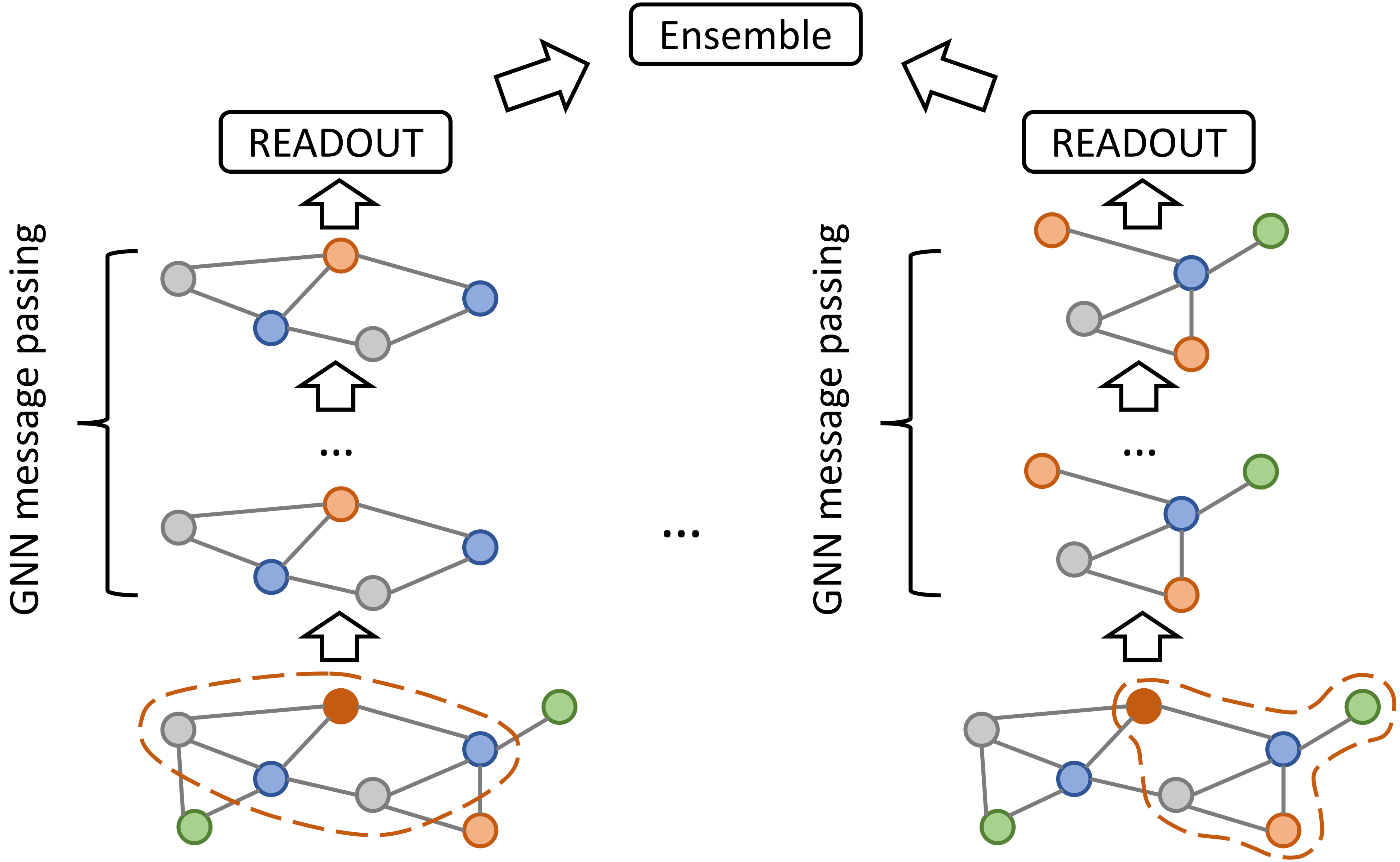
Ensemble of different subgraphs

- Different graph metrics captures different neighbor importance
- Design a single complicated subgraph extraction function?

→ Ensemble simple subgraph extractors

  - e.g., [L-hop] + [PPR]

Architecture: Full Picture



Target node

Evaluation: Setup

Tasks	node classification & link prediction
Datasets	7 graphs (up to 111M nodes)
	inductive & transductive
Backbone models	5 aggregation functions & residue connection
Training of baselines	full batch & GraphSAINT minibatch
Training of proposed	minibatch of independently constructed $\mathcal{G}_{[v]}$

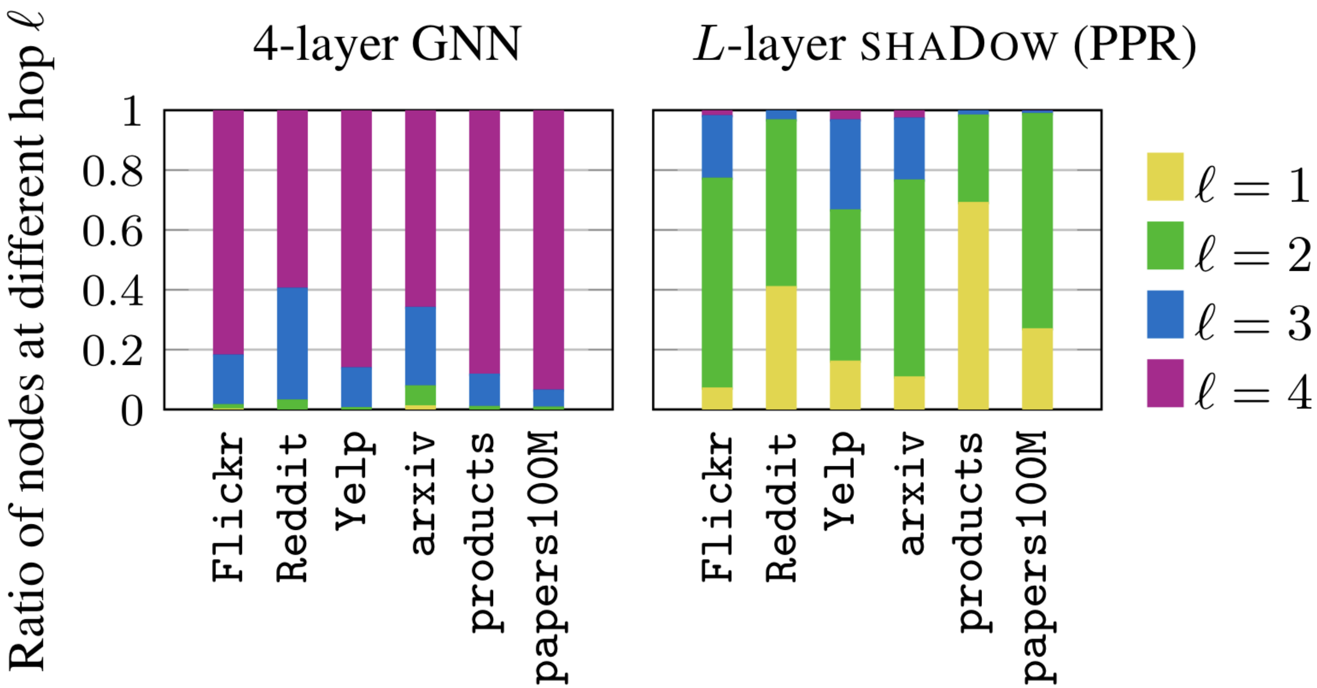
Practical design: **shaDow-GNN** (Decoupled GNN on shallow subgraphs)

- Scope: based on 2-hop / PPR (top 200 nodes)
- Depth: 3- / 5-layer

Evaluation: Neighborhood Composition

How many neighbors are ℓ hops away from the target node?

- Scope of normal GNN
  - Dominated by distant neighbors
  - Size grows rapidly
- Scope of shaDow-GNN
  - Consists of nearby neighbors
  - Size is small regardless of number of layers ($<$ 200 neighbors)



Evaluation: Baseline Comparisons

- Decoupling improves **accuracy** at lower computation **cost**
- Decoupling is a general design principle applicable to **various backbones**
- **Subgraph extraction** algorithms are important

Method	Layers	Flickr		Reddit		Yelp		ogbn-arxiv		ogbn-products	
		Accuracy	Cost	Accuracy	Cost	F1-micro	Cost	Accuracy	Cost	Accuracy	Cost
GCN	3	0.5159±0.0017	2E0	0.9532±0.0003	6E1	0.4028±0.0019	2E1	0.7170±0.0026	1E1	0.7567±0.0018	5E0
	5	0.5217±0.0016	2E2	0.9495±0.0012	1E3	OOM	1E3	0.7186±0.0017	1E3	OOM	9E2
GCN	3	0.5155±0.0027	2E0	0.9523±0.0003	6E1	0.5110±0.0012	2E1	0.7093±0.0003	1E1	**0.8003**±0.0024	5E0
+ GraphSAINT-RW	5	0.5165±0.0026	2E2	0.9523±0.0012	1E3	0.5012±0.0021	1E3	0.7039±0.0020	1E3	0.7992±0.0021	9E2
SHADOW-GCN	3	0.5262±0.0018	**(1)**	0.9581±0.0004	**(1)**	0.5255±0.0012	**(1)**	0.7192±0.0025	**(1)**	0.7778±0.0030	**(1)**
+PPR	5	**0.5270**±0.0024	1E0	**0.9583**±0.0002	1E0	**0.5272**±0.0018	2E0	**0.7207**±0.0030	2E0	0.7844±0.0029	2E0
GraphSAGE	3	0.5140±0.0014	3E0	0.9653±0.0002	5E1	0.6178±0.0033	2E1	0.7192±0.0027	1E1	0.7858±0.0013	4E0
	5	0.5154±0.0052	2E2	0.9626±0.0004	1E3	OOM	2E3	0.7193±0.0037	1E3	OOM	1E3
GraphSAGE	3	0.5176±0.0032	3E0	0.9671±0.0003	5E1	0.6453±0.0011	2E1	0.7107±0.0003	1E1	0.7923±0.0023	4E0
+ GraphSAINT-RW	5	0.5201±0.0032	2E2	0.9670±0.0010	1E3	0.6394±0.0003	2E3	0.7013±0.0021	1E3	0.7964±0.0022	1E3
SHADOW-SAGE	3	0.5288±0.0014	1E0	0.9660±0.0003	1E0	0.6493±0.0001	1E0	0.7163±0.0012	1E0	0.7993±0.0012	1E0
+ 2-hop	5	0.5338±0.0038	2E0	0.9661±0.0002	2E0	0.6503±0.0001	2E0	0.7183±0.0012	2E0	0.8014±0.0020	2E0
SHADOW-SAGE	3	0.5344±0.0028	**(1)**	**0.9693**±0.0002	**(1)**	**0.6512**±0.0002	**(1)**	0.7234±0.0032	**(1)**	0.7945±0.0021	**(1)**
+ PPR	5	**0.5424**±0.0025	2E0	0.9691±0.0003	2E0	0.6502±0.0001	2E0	**0.7255**±0.0013	2E0	**0.8043**±0.0026	2E0
GAT	3	0.5070±0.0032	2E1	OOM	3E2	OOM	2E2	0.7201±0.0011	1E2	OOM	3E1
	5	0.5164±0.0033	2E2	OOM	2E3	OOM	2E3	OOM	3E3	OOM	2E3
GAT	3	0.5225±0.0053	2E1	0.9671±0.0003	3E2	0.6459±0.0002	2E2	0.6977±0.0003	1E2	0.8027±0.0028	3E1
+ GraphSAINT-RW	5	0.5153±0.0034	2E2	0.9651±0.0024	2E3	0.6478±0.0012	2E3	0.6954±0.0013	3E3	0.7990±0.0072	2E3
SHADOW-GAT	3	**0.5383**±0.0032	**(1)**	0.9703±0.0010	**(1)**	**0.6549**±0.0002	**(1)**	0.7243±0.0011	**(1)**	0.8014±0.0012	**(1)**
+ PPR	5	0.5342±0.0023	2E0	**0.9710**±0.0008	2E0	0.6537±0.0004	2E0	**0.7283**±0.0027	2E0	**0.8094**±0.0034	2E0

Evaluation: Scaling to 100M-Node Graph

OGB leaderboard comparison

- Higher accuracy
- 3 orders of magnitude smaller neighborhood size

Table 2: Leaderboard comparison on papers100M

Method	Test accuracy	Val accuracy	Neigh size
GraphSAGE+incep	0.6707±0.0017	0.7032±0.0011	4E5
SIGN-XL	0.6606±0.0019	0.6984±0.0006	> 4E5
SGC	0.6329±0.0019	0.6648±0.0020	> 4E5
SHADOW-GAT$_{200}$	0.6681±0.0016	0.7019±0.0011	2E2
SHADOW-GAT$_{400}$	**0.6710**±0.0015	**0.7067**±0.0012	3E2

Memory consumption

- Lowest in both CPU and GPU
- Train & inference the 100M graph on a low-end server

Memory consumption of the ogbn-papers100M leaderboard methods

Method	CPU RAM	GPU memory
GraphSAGE+incep	80GB	16GB
SIGN-XL	>682GB	4GB
SGC	>137GB	4GB
SHADOW-GAT	**80GB**	**4GB**

Conclusion

General design principle to decouple the depth & scope of GNNs

- Theoretical benefits in expressivity & scalability
- Empirical performance gain in accuracy & cost
- Flexibility w.r.t. GNN architecture, subgraph extraction algorithms & learning tasks

Public implementations

- Official code:

https://github.com/facebookresearch/shaDow_GNN

- PyG version:

https://pytorch-geometric.readthedocs.io/en/latest/modules/loader.html#torch_geometric.loader.ShaDowKHopSampler

- DGL version:

https://docs.dgl.ai/en/latest/_modules/dgl/dataloading/shadow.html

Thank you!