PaSca: a Graph Neural Architecture Search System under the Scalable Paradigm

Wentao Zhang, Yu Shen, Zheyu Lin, Yang Li,
Xiaosen Li, Wen Ouyang, Yangyu Tao, Zhi Yang, Bin Cui

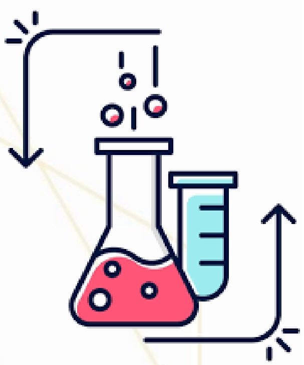
Peking University, Tencent Inc.

2022.04.27

Presentation Outline

1. Motivation

2. Method



3. Experiment

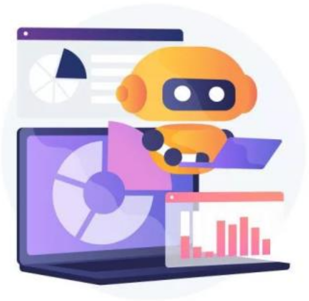
4. Conclusion

Motivation

Graph Neural Networks

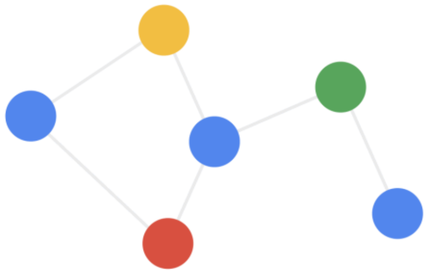
- Graph neural networks (GNNs) have been widely applied to web-based applications.

Social influence prediction



Recommendation system

- Neighborhood expansion in GNNs
  - Leads to exciting performance
  - Requires to gather information



Figures from internet

Neural Message Passing

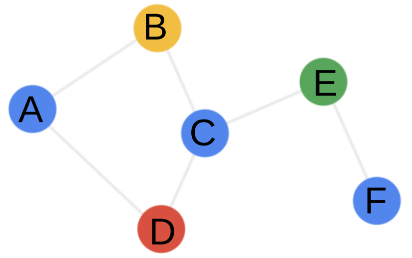
- Traditional GNN designs (e.g., GCN[1], GAT[2]) follow the neural message passing (NMP) paradigm:
  - Aggregate the neighborhood information (**Communication**)

$$\mathbf{m}_v^t \leftarrow \mathtt{aggregate}\left(\{\mathbf{h}_u^{t-1} | u \in \mathcal{N}_v\}\right)$$

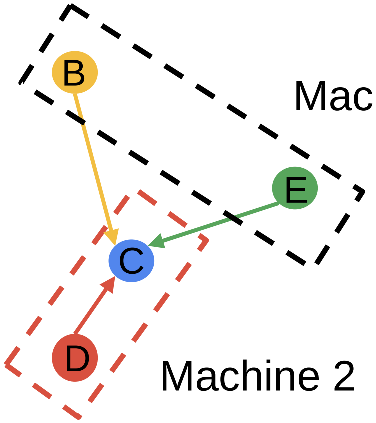
  - Update the message via neural networks (**Computation**)

$$\mathbf{h}_v^t \leftarrow \mathtt{update}(\mathbf{m}_v^t)$$

- Drawback: **Frequently** fetch information from other machines → **High** communication cost during each epoch on large datasets



Input Graph



GIF from https://blog.csdn.net/DreamHome_S/article/details/105619194

[1] Thomas N Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In ICLR.
[2] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In ICLR.

GNN Systems

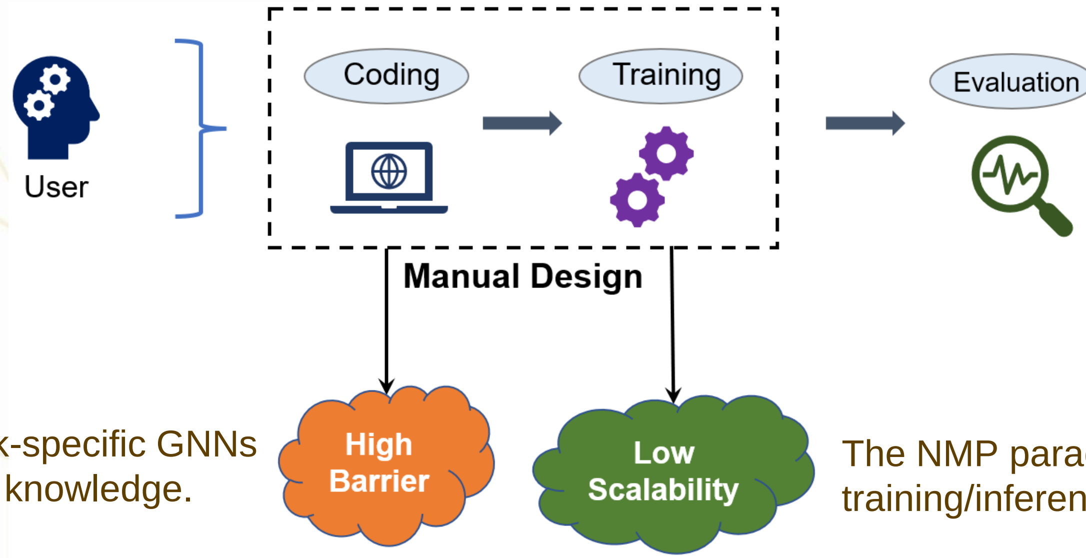
- Most GNN systems adopt the NMP paradigm.

DGL[1]

PyG[2]

- Challenges for web-scale graphs

Designing task-specific GNNs require expert knowledge.

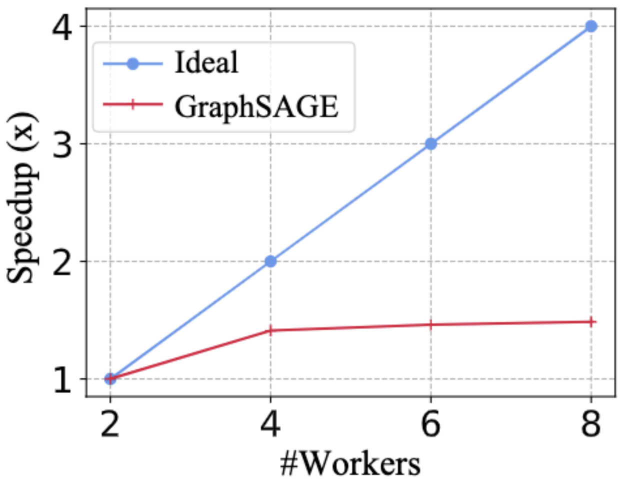


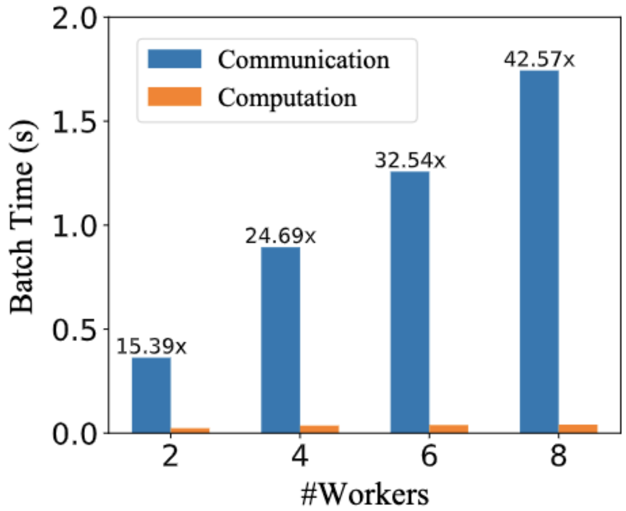
The NMP paradigm leads to high training/inference time.

[1] https://github.com/dmlc/dgl
[2] https://github.com/pyg-team/pytorch_geometric

Bottlenecks

- Scalability issue
  - The speedup decreases when using more workers.
  - The communication costs dominate the training process.



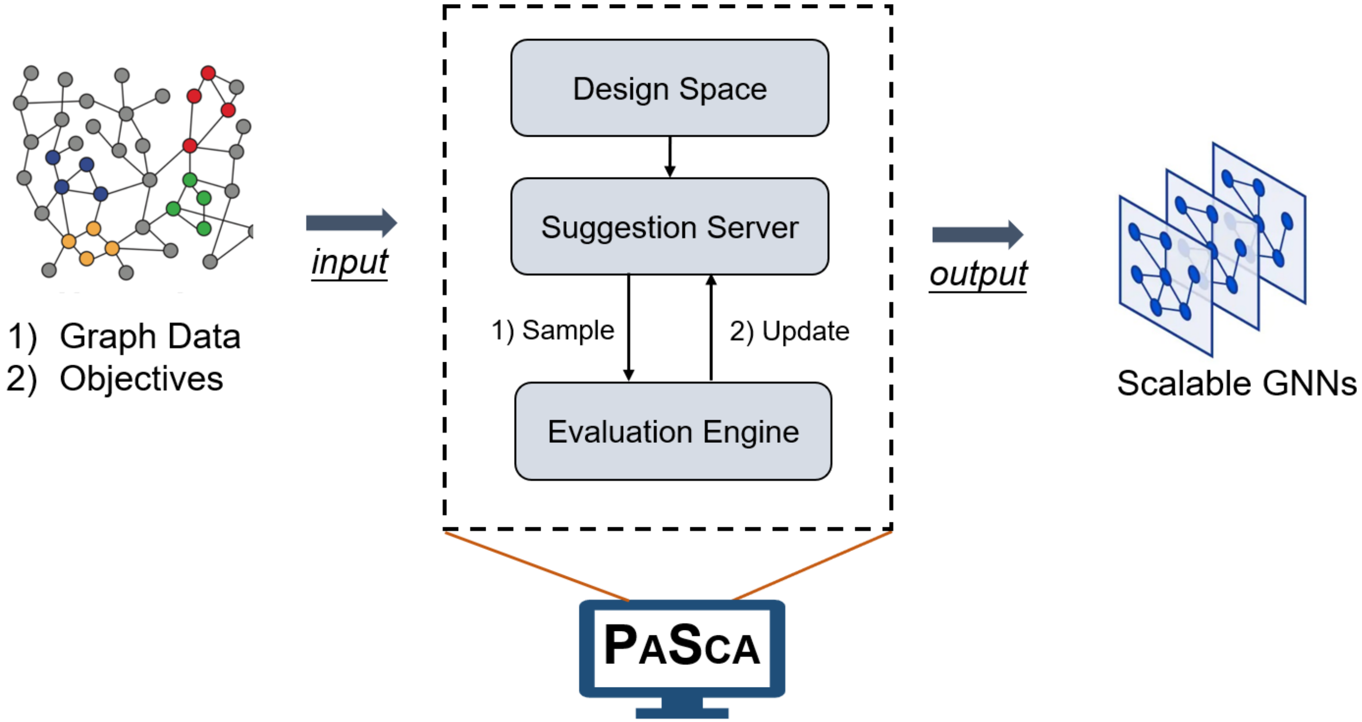


- Motivation: Can we propose a novel GNN system to support simple and scalable graph learning for large graphs?

Method

Method Overview

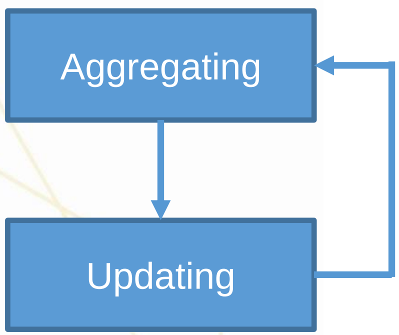
- Input: Graph dataset + Optimization objectives
- Output: **Scalable** GNNs that tackle the tradeoff between objectives well
- **End-to-end without further interaction**



Method Outline

- Scalable paradigm (SGAP)
  - Abstraction to define a scalable training process
- Auto-search system (PaSca)

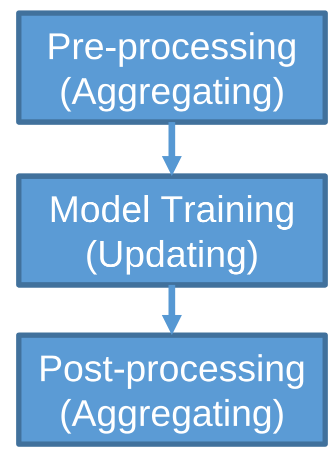
Unscalable Design



Fetch information **during** training

The number of training epochs

Scalable Paradigm



Fetch information **before and after** training

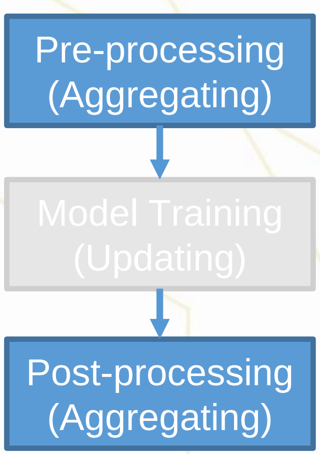
Twice

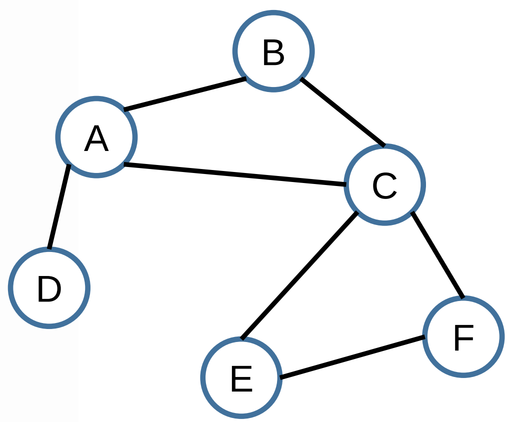
SGAP Abstraction

- Pre-processing
  - Aggregate messages (**features**) from neighbors
- Post-processing
  - Aggregate messages (**soft predictions**) from neighbors

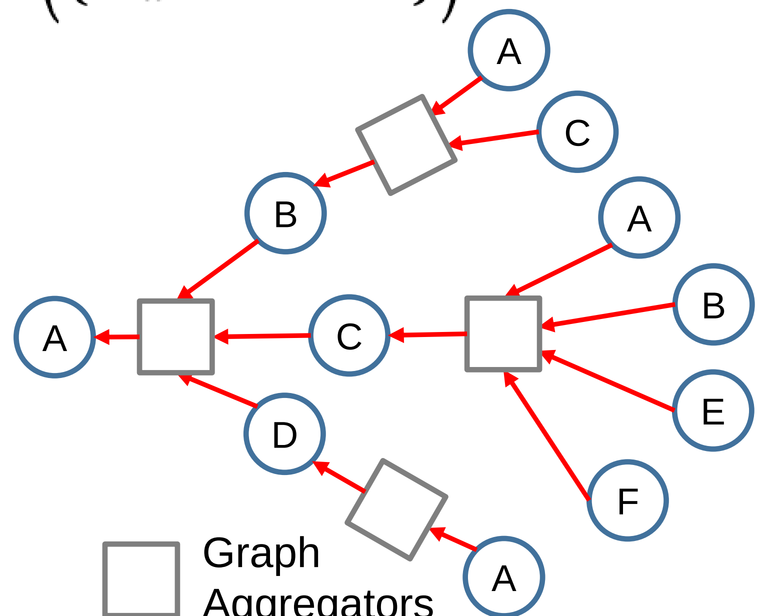
$$\mathbf{m}_v^t \leftarrow \texttt{graph\_aggregator}\left(\left\{\mathbf{m}_u^{t-1} \mid u \in \mathcal{N}_v\right\}\right)$$

Scalable Paradigm





Input Graph



Graph Aggregator

- Abstraction $\mathbf{m}_v^t \leftarrow \texttt{graph\_aggregator}\left(\{\mathbf{m}_u^{t-1} | u \in \mathcal{N}_v\}\right)$
- Augmented normalized adjacency (used in GCN[1])

$$\mathbf{m}_v^t = \sum_{u \in \mathcal{N}_v} \frac{1}{\tilde{d}_u} \mathbf{m}_u^{t-1}$$

- Personalized PageRank (used in APPNP[2])

$$\mathbf{m}_v^t = \alpha \mathbf{m}_v^0 + (1-\alpha) \sum_{u \in \mathcal{N}_v} \frac{1}{\sqrt{\tilde{d}_v \tilde{d}_u}} \mathbf{m}_u^{t-1}$$

- Triangle-induced adjacency (used MotifNet[3])

$$\mathbf{m}_v^t = \sum_{u \in \mathcal{N}_v} \frac{1}{d_v^{tri}} \mathbf{m}_u^{t-1}$$

[1] Thomas N Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In ICLR.

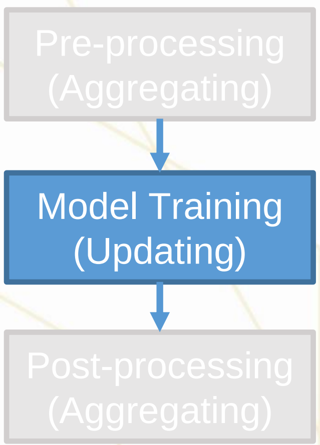
[2] Johannes Klicpera, Aleksandar Bojchevski, and Stephan Günnemann. 2019. Predict then Propagate: Graph Neural Networks meet Personalized PageRank. In ICLR.

[3] Federico Monti, Karl Otness, and Michael M Bronstein. 2018. Motifnet: a motif-based graph convolutional network for directed graphs. In 2018 IEEE Data Science Workshop (DSW). IEEE, 225–228.

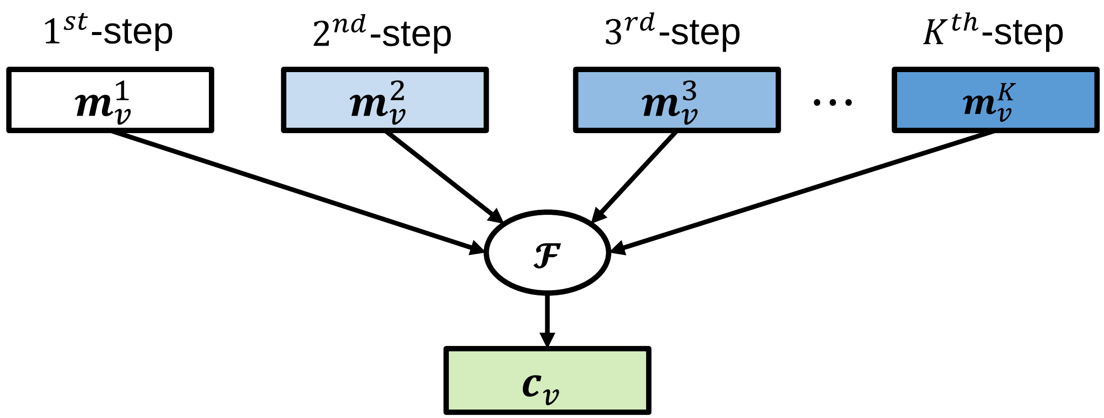
SGAP Abstraction

- Training
  - Aggregate the messages from the pre-processing stage
  - Update the combined message via dense layers

Scalable Paradigm



$$\mathbf{c}_v \leftarrow \texttt{message\_aggregator}(\mathcal{M}_v)$$



$$\mathbf{h}_v \leftarrow \texttt{message\_updater}(\mathbf{c}_v)$$

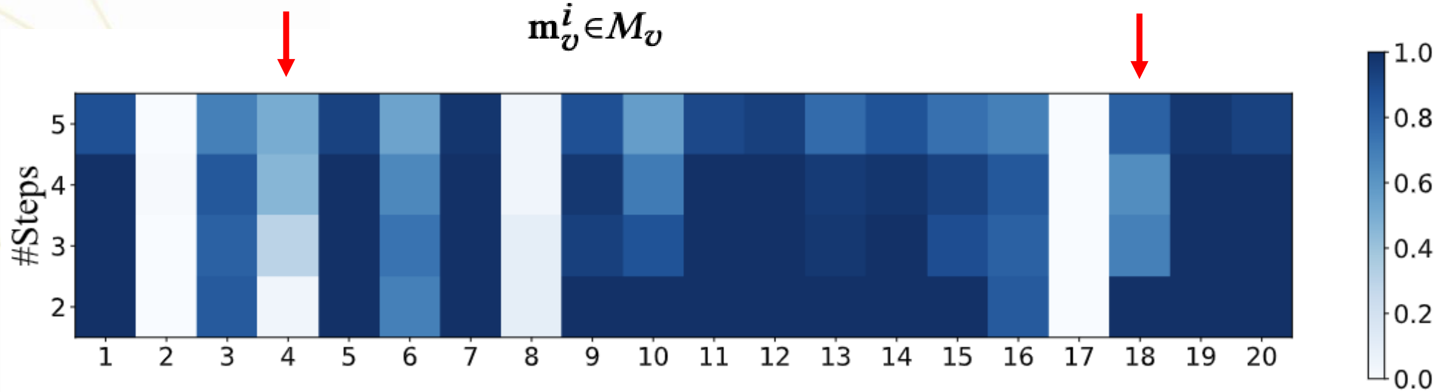
Message Aggregator

- Abstraction $\mathbf{c}_v \leftarrow \texttt{message\_aggregator}(\mathcal{M}_v)$
- Non-adaptive aggregator (mean, max)

$$c_{msg} \leftarrow \oplus_{\mathbf{m}_v^i \in M_v} w_i f(\mathbf{m}_v^i)$$

- Adaptive aggregator (gate with trainable parameters)

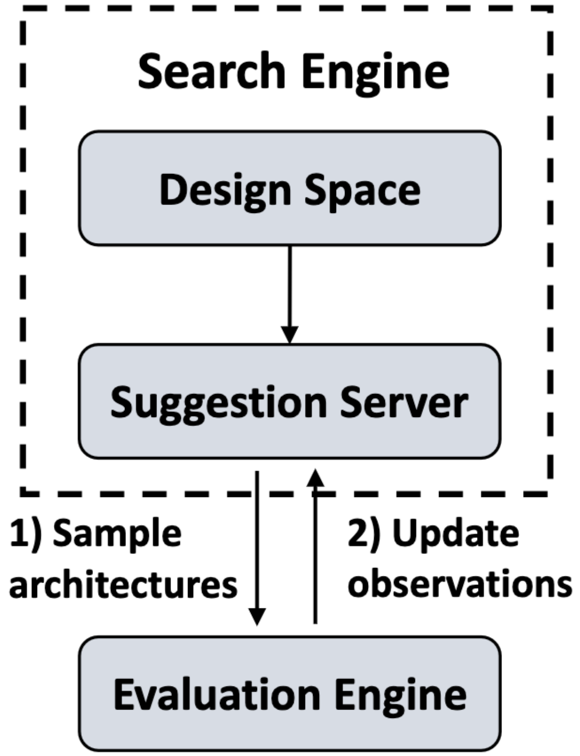
$$c_{msg} \leftarrow \sum_{\mathbf{m}_v^i \in M_v} w_i \mathbf{m}_v^i, \;\; w_i = \sigma(\mathbf{s}\mathbf{m}_v^i)$$



We should assign messages with different weights for different nodes!

Method Outline

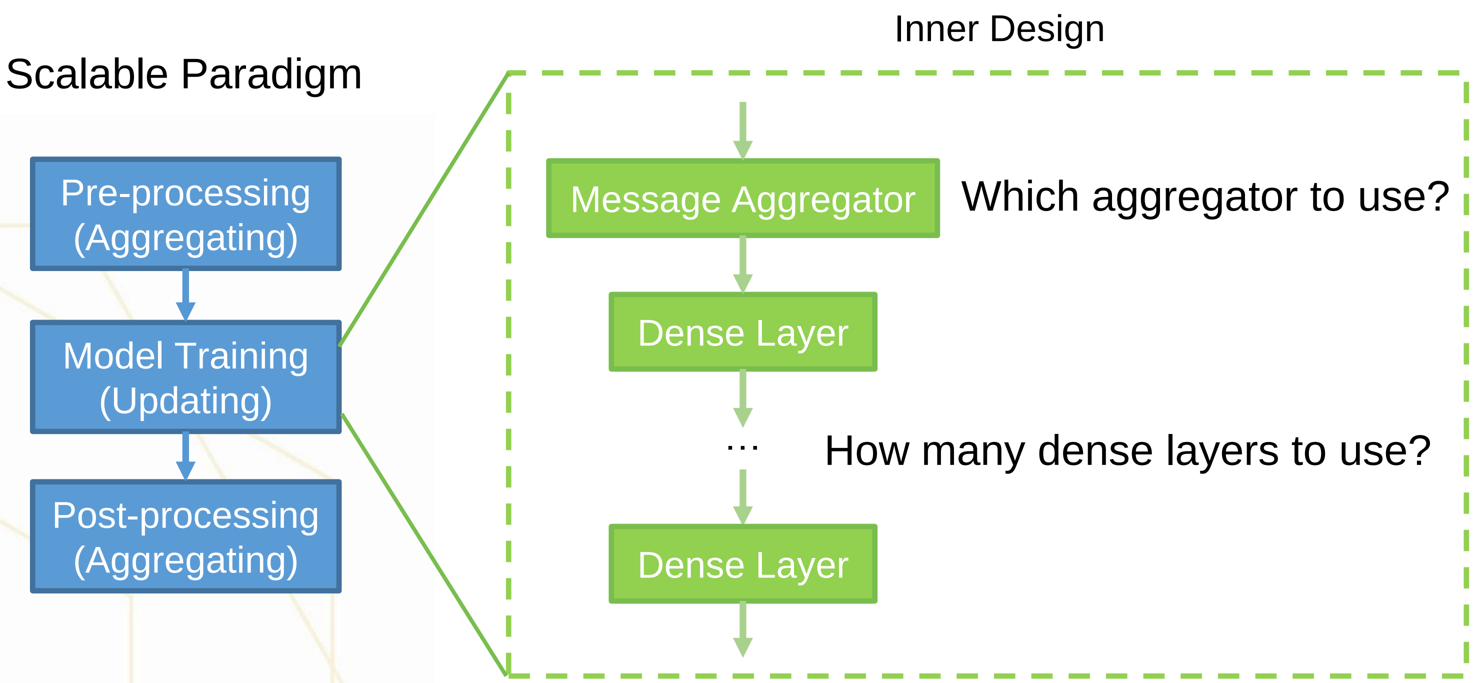
- Scalable paradigm (SGAP)
- Auto-search system (PaSca)
  - Two components
    - (Automatic) search engine
    - (Distributed) evaluation engine
  - The search engine **suggests** an configuration instance.
  - The evaluation engine **evaluates** the configuration instance.



Searching

Search Engine

- Tackle tradeoff between different objectives
- Design space: Choices of inner design (parameter) in three SGAP stages



Design Space

- 6 parameters to choose + 2 parameters for each stage
- Over 150k possible configuration instances

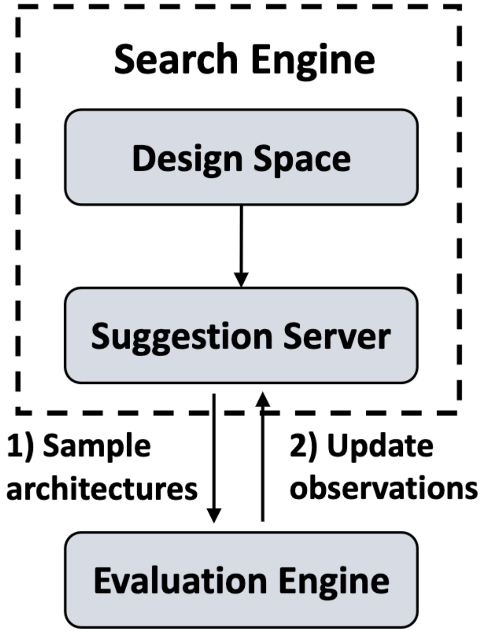
Stages	Name	Range/Choices	Type
Pre-processing	Aggregation steps (K_{pre}) Graph aggregators (GA_{pre})	[0, 10] {Aug.NA, PPR($\alpha = 0.1$), PPR($\alpha = 0.2$), PPR($\alpha = 0.3$), Triangle. IA}	Integer Categorical
Model training	Message aggregators (MA) Transformation steps (K_{trans})	{None, Mean, Max, Concatenate, Weighted, Adaptive} [1, 10]	Categorical Integer
Post-processing	Aggregation steps (K_{post}) Graph aggregators (GA_{post})	[0, 10] {Aug.NA, PPR($\alpha = 0.1$), PPR($\alpha = 0.2$), PPR($\alpha = 0.3$), Triangle. IA}	Integer Categorical

- The space also contains recent scalable architecture designs.

Models	Pre-processing	Model training		Post-processing
	GA_{pre}	MA	K_{trans}	GA_{post}
SGC	Aug.NA	None	1	/
SIGN	Optional	Concatenate	1	/
S^{2}GC	PPR	Mean	1	/
GBP	Aug.NA	Weighted	≥ 2	/
PaSca-APPNP	/	/	≥ 2	PPR

Suggestion Server

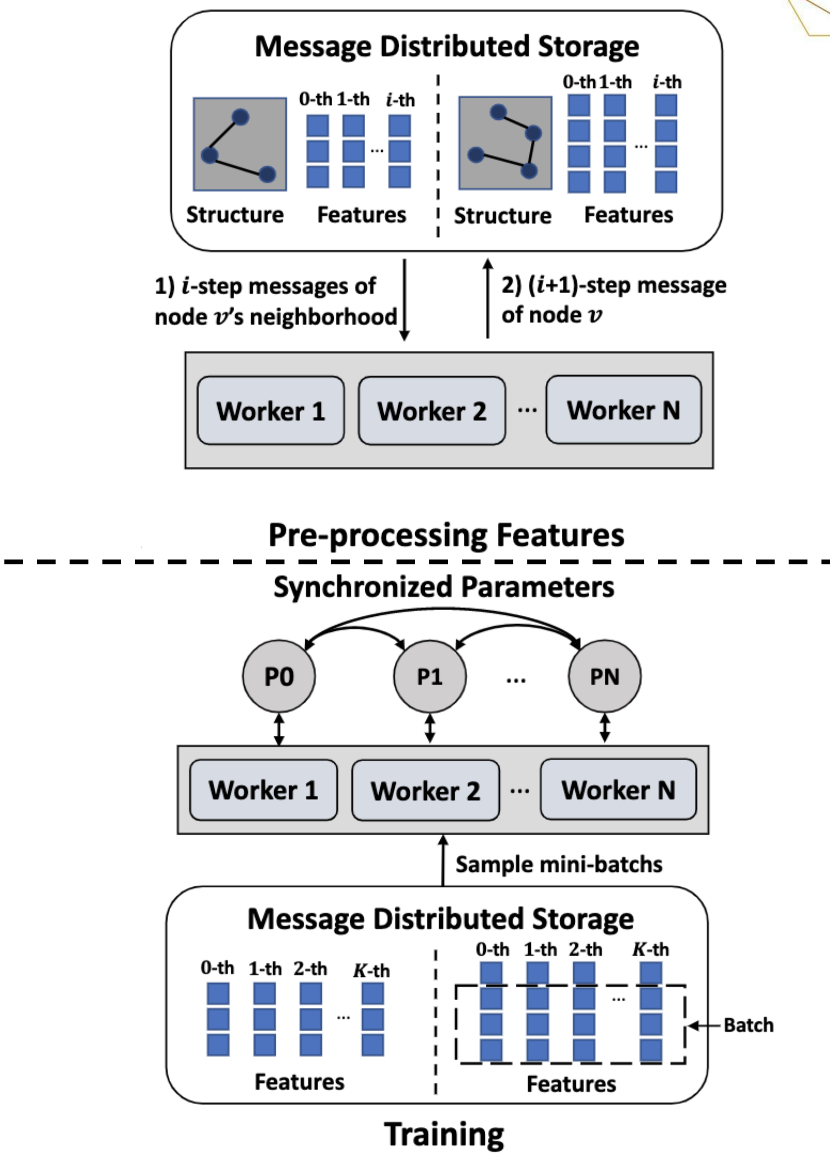
- **Model** the relationship between instances and objective values
- **Suggest** the instance that is expected to tackle the tradeoff well
- **Update** the history with observed performance



Searching

Evaluation Engine

- Graph data aggregator
  - Partition large graphs
  - Compute the (i+1)th-step messages after all ith-step messages are ready
- Neural architecture trainer
  - Mini-batch training
  - Asynchronous training via a parameter server



Experiment

Settings

- Dataset

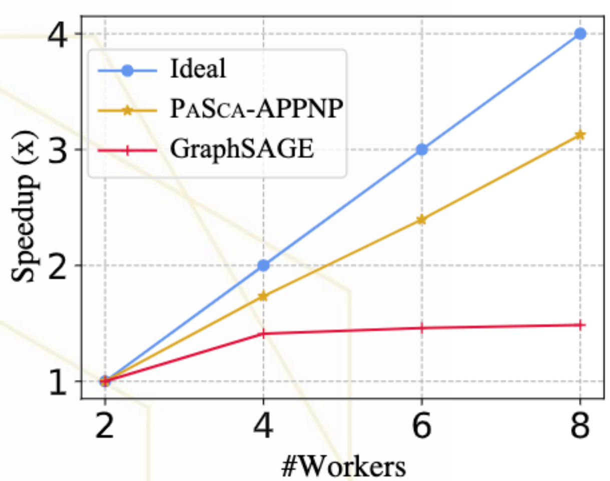
Dataset	#Nodes	#Features	#Edges	#Classes	#Train/Val/Test	Task type	Description
Cora	2,708	1,433	5,429	7	140/500/1000	Transductive	citation network
Citeseer	3,327	3,703	4,732	6	120/500/1000	Transductive	citation network
Pubmed	19,717	500	44,338	3	60/500/1000	Transductive	citation network
Amazon Computer	13,381	767	245,778	10	200/300/12881	Transductive	co-purchase graph
Amazon Photo	7,487	745	119,043	8	160/240/7,087	Transductive	co-purchase graph
ogbn-products	2,449,029	100	61,859,140	47	195922/489811/204126	Transductive	co-purchase network
Coauthor CS	18,333	6,805	81,894	15	300/450/17,583	Transductive	co-authorship graph
Coauthor Physics	34,493	8,415	247,962	5	100/150/34,243	Transductive	co-authorship graph
Flickr	89,250	500	899,756	7	44,625/22,312/22,312	Inductive	image network
Reddit	232,965	602	11,606,919	41	155,310/23,297/54,358	Inductive	social network
Industry	1,000,000	64	1,434,382	253	5,000/10,000/30,000	Transductive	user-video graph

- Insights
  - SGAP is ***more scalable*** than other paradigms.
  - The search results of PaSca can ***tackle the tradeoff*** between different objectives ***well***.
  - The search results achieve ***higher predictive performance***.

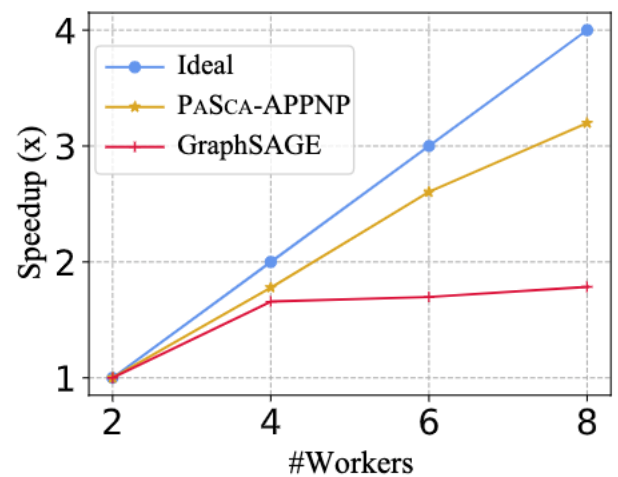
Scalability Analysis

- Baseline
  - SGAP: APPNP under SGAP with PaSca evaluation engine
  - NMP: GraphSAGE with DistDGL

- The SGAP architecture achieves a near-linear speedup and is closer to the ideal speedup.



Reddit (>230K nodes)



ogbn-product (>2.4M nodes)

Search Representatives

- Representatives (on the Pareto Front)
  - Searched instances from SGAP design space that tackle the trade-off well
  - PaSca-V3 achieves lower predictive error but requires longer inference time than PaSca-V2.
- Our search results also include GBP[1], a SOTA scalable design.

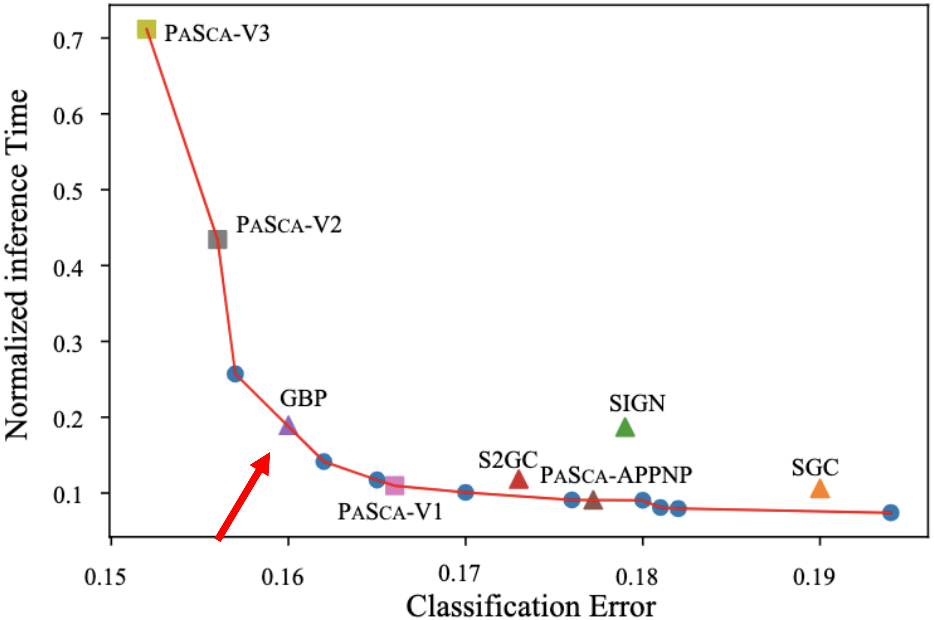


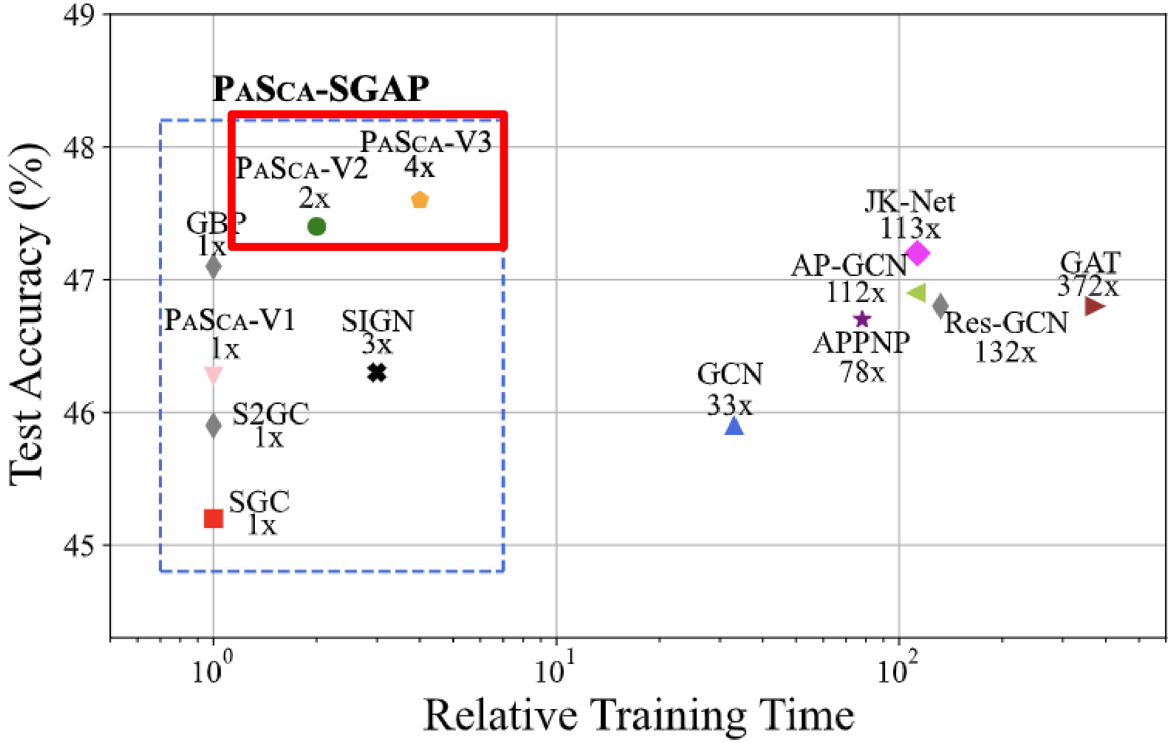
Table 3: Scalable GNNs found by PaSca.

Models	Pre-processing			Model training	Post-processing	
	GA_{pre}	MA	K_{pre}	K_{trans}	GA_{post}	K_{post}
PaSca-V1	PPR($\alpha = 0.1$)	Weighted	3	2	/	/
PaSca-V2	Aug.NA	Adaptive	6	2	/	/
PaSca-V3	Aug.NA	Adaptive	6	3	PPR ($\alpha = 0.3$)	4

[1]Chen M, Wei Z, Ding B, et al. 2020. Scalable graph neural networks via bidirectional propagation[J]. In NeurIPS.

Search Representatives

- The search results tackle the tradeoff well.
- PaSca V2 and V3 achieve better accuracy than the SOTA JK-Net and require significantly short training time.



[1] Xu K, Li C, Tian Y, et al. 2018. Representation learning on graphs with jumping knowledge networks. In ICML.

Predictive Performance

- SGAP architectures achieve competitive results compared with unscalable paradigms.
- PaSca-V3 achieves the best test results across different datasets.

Type	Models	Cora	Citeseer	PubMed	Amazon Computer	Amazon Photo	Coauthor CS	Coauthor Physics	Industry
NMP	GCN	81.8±0.5	70.8±0.5	79.3±0.7	82.4±0.4	91.2±0.6	90.7±0.2	92.7±1.1	45.9±0.4
	GAT	83.0±0.7	72.5±0.7	79.0±0.3	80.1±0.6	90.8±1.0	87.4±0.2	90.2±1.4	46.8±0.7
	JK-Net	81.8±0.5	70.7±0.7	78.8±0.7	82.0±0.6	91.9±0.7	89.5±0.6	92.5±0.4	47.2±0.3
	ResGCN	82.2±0.6	70.8±0.7	78.3±0.6	81.1±0.7	91.3±0.9	87.9±0.6	92.2±1.5	46.8±0.5
DNMP	APPNP	83.3±0.5	71.8±0.5	80.1±0.2	81.7±0.3	91.4±0.3	92.1±0.4	92.8±0.9	46.7±0.6
	AP-GCN	83.4±0.3	71.3±0.5	79.7±0.3	83.7±0.6	92.1±0.3	91.6±0.7	93.1±0.9	46.9±0.7
SGAP	SGC	81.0±0.2	71.3±0.5	78.9±0.5	82.2±0.9	91.6±0.7	90.3±0.5	91.7±1.1	45.2±0.3
	SIGN	82.1±0.3	72.4±0.8	79.5±0.5	83.1±0.8	91.7±0.7	91.9±0.3	92.8±0.8	46.3±0.5
	S^2GC	82.7±0.3	73.0±0.2	79.9±0.3	83.1±0.7	91.6±0.6	91.6±0.6	93.1±0.8	45.9±0.4
	GBP	83.9±0.7	72.9±0.5	80.6±0.4	83.5±0.8	92.1±0.8	92.3±0.4	93.3±0.7	47.1±0.6
	PaSca-V1	83.4±0.5	72.2±0.5	80.5±0.4	83.7±0.7	92.1±0.7	91.9±0.3	93.2±0.6	46.3±0.4
	PaSca-V2	84.4±0.3	73.1±0.3	80.7±0.7	84.1±0.7	92.4±0.7	92.6±0.4	93.6±0.8	47.4±0.6
	PaSca-V3	**84.6±0.6**	**73.4±0.5**	**80.8±0.6**	**84.8±0.7**	**92.7±0.8**	**92.8±0.5**	**93.8±0.9**	**47.6±0.3**

Conclusion

Conclusion

- We present PaSca, a novel auto-search system to construct and explore scalable GNNs, rather than studying individual designs.
- Representative architectures from PaSca outperforms SOTA GNNs in terms of predictive performance, efficiency, and scalability.
- PaSca can help researchers explore design space for scalable GNNs and understand different design choices.
- The code is available at https://github.com/PKU-DAIR/SGL.

Thanks for listening

Q&A