


30th ACM International Conference on Information and Knowledge Management

Pooling Architecture Search for Graph Classification

Lanning Wei[1,3,+], Huan Zhao[3,+], Quanming Yao,[2,3], Zhiqiang He[1,4]

Institute of Computing Technology, Chinese Academy of Sciences[1]
Department of Electronic Engineering, Tsinghua University[2]
4Paradigm[3] Lenovo[4]
Equal contribution[+]

Outline

- Introduction
- Pooling architecture search
- Experiments

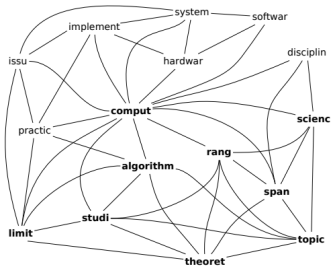
Outline

- Introduction
  - Graph classification
  - Adaptive pooling architectures
  - Pooling architecture search
- Pooling architecture search
- Experiments

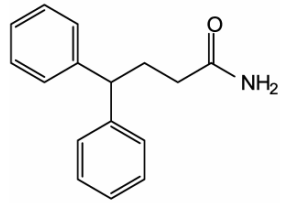
Graph classification task

- Various applications of graph classifications task.

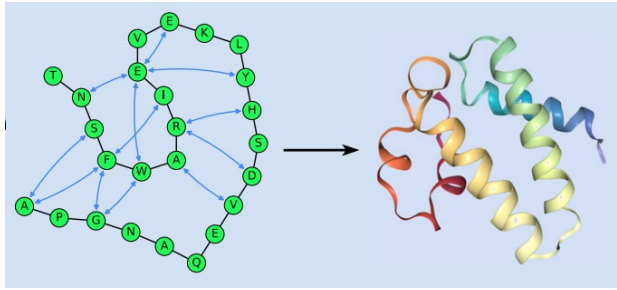
Social network[1]



Text graph[2]



Molecule[3]



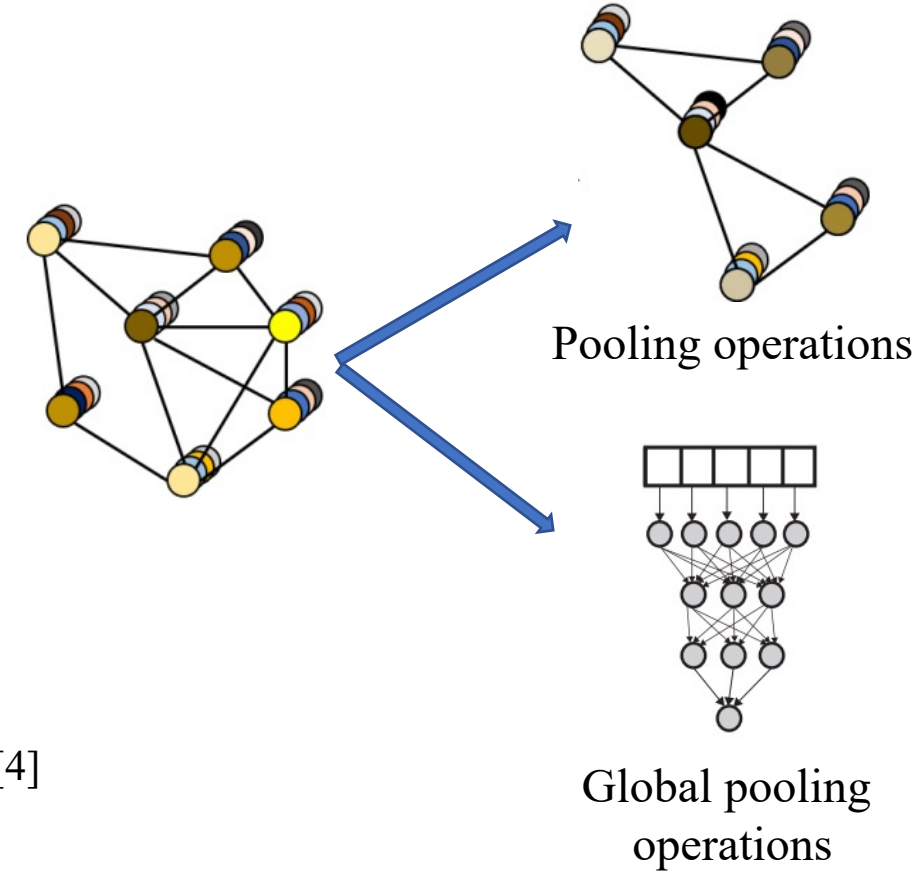
Protein[4]

- Represent a graph $G = (\mathbf{A}, \mathbf{H})$, where $\mathbf{A} \in \mathbb{R}^{N \times N}$ is the adjacency matrix, $\mathbf{H} \in \mathbb{R}^{N \times d}$ is the node features and $\mathbf{h}_v^l$ is the features of node v in l-th layer.
- Pooling methods proposed to learn the representations $\mathbf{h}_G \in \mathbb{R}^d$ of graphs.

[1] https://medium.com/analytics-vidhya/social-network-analytics-f082f4e21b16
[2] Rousseau F et al. Text categorization as a graph classification problem. IJCNLP 2015
[3] https://www.mdpi.com/2078-2489/1/2/60/htm
[4] Strokach A et al. Fast and flexible protein design using deep graph neural networks[J]. Cell Systems, 2020, 11(4): 402-411. e4.

Graph classification methods

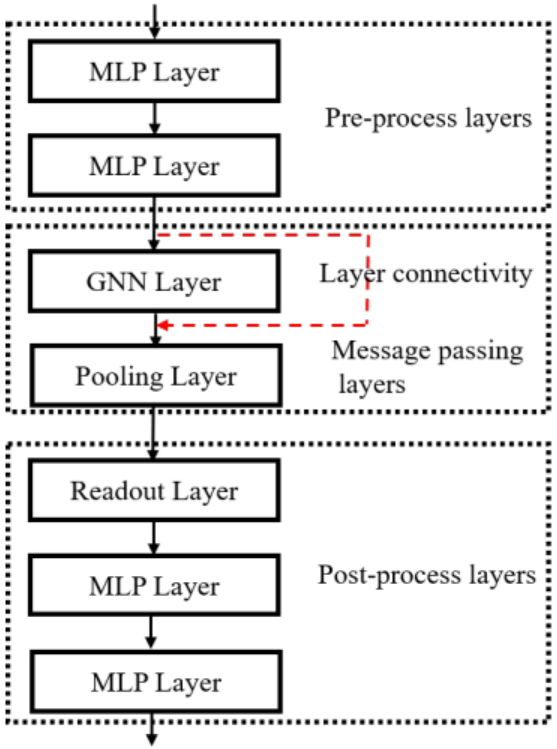
- Global pooling
  - Pool nodes at the end of GNN.
  - Representative works: GIN[1] / DGCNN[2]
  - Learn flat graph-level representation.
- Hierarchical pooling
  - Learn coarser and coarser graphs in the hierarchical pooling methods.
  - Representative works: DiffPool [3] / SAGPool [4] / ASAP [5]

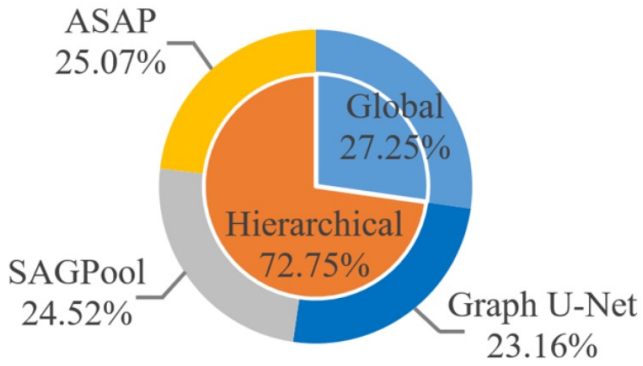


[1] Keyulu Xu et al. How powerful are graph neural networks? ICLR 2019
[2] Muhan Zhang et al. An end-to-end deep learning architecture for graph classification. AAAI 2018
[3] Zhitao Ying et al. Hierarchical graph representation learning with differentiable pooling. NeurIPS 2018
[4] Junhyun Lee et al. Self-Attention Graph Pooling. ICML 2019
[5] Ekagra Ranjan et al. ASAP: Adaptive Structure Aware Pooling for Learning Hierarchical Graph Representations. AAAI 2020

Adaptive pooling architectures

Considering the graph property, different graphs prefer different pooling methods.





- D&D, PROTEINS, BZR, COX2, IMDB-MULTI, ENZYMES
- smallworld-clustering, scalefree-clustering
- smallworld-const, scalefree-const
- smallworld-onehot, scalefree-onehot
- smallworld-pagerank, scalefree-pagerank

- No single human-designed pooling architecture can win in all cases.
- The hierarchical pooling method is not always superior to the global one.
- It is very important to design data-specific pooling architectures for the graph classification task.

Adaptive pooling architectures

- Neural architecture search (NAS) methods are proposed to design data-specific architectures.
- Most of the existing NAS methods for GNN are focusing on the aggregation and global pooling functions. They fail to obtain the hierarchical pooling paradigm.
- On top of the differentiable search algorithm, it is infeasible to directly relax the pooling operations due to the diverse nodes and edges in different pooling operations.

Pooling architecture search

Pooling Architecture Search (PAS) for Graph Classification

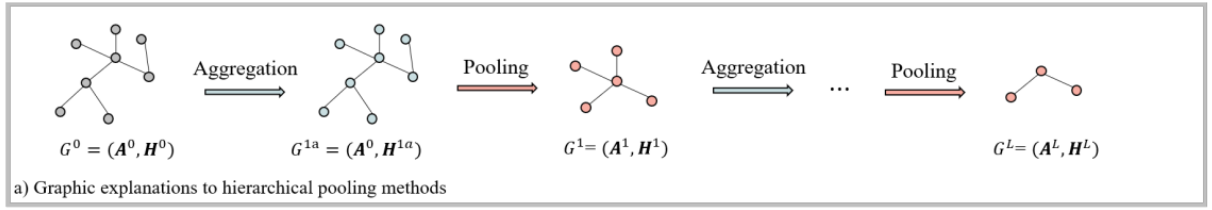
- The first method to learn data-specific pooling architectures.
- A unified framework consisting of 4 modules is defined to cover the existing pooling architectures.
- A customized and effective search space is proposed to enable the design of global and hierarchical pooling architectures.
- A coarsening strategy is designed to relax the search space and enable the usage of the differentiable search algorithm.

Outline

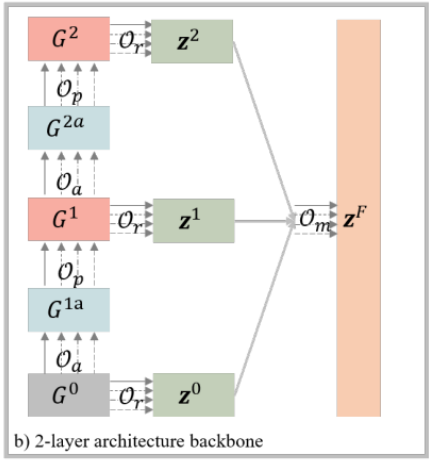
- Introduction
- Pooling architecture search
  - The unified framework
  - The design of the search space
  - Differentiable search algorithm
- Experiments

The unified framework

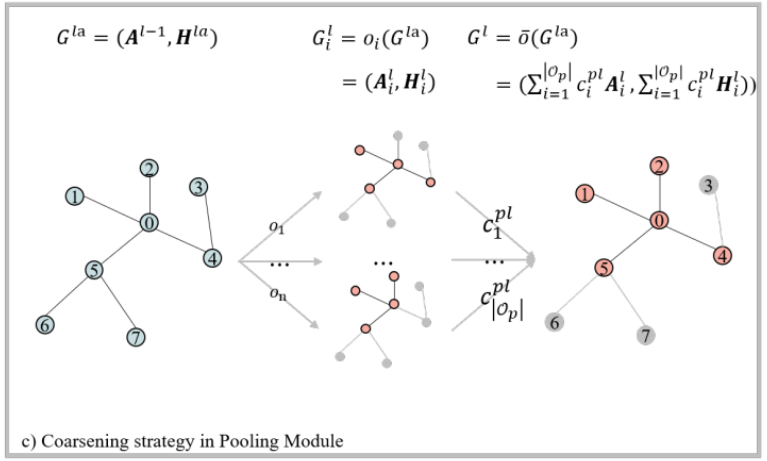
- The framework is constructed with four key modules derived from existing pooling methods.
- Design one novel search space on the proposed framework which can cover both the global and hierarchical pooling architectures.
- Design one coarsening strategy to relax the pooling module and enable the usage of the differentiable search algorithm.



a) Graphic explanations to hierarchical pooling methods



b) 2-layer architecture backbone

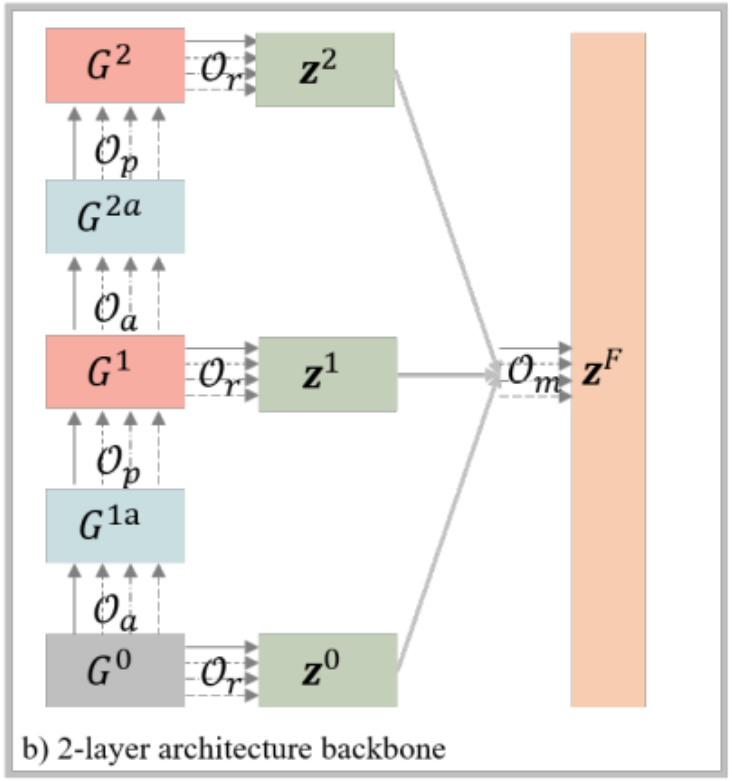


c) Coarsening strategy in Pooling Module

	Methods	Task	Search Space				Search
			A	P	R	M	Algorithm
GNNs	GraphSAGE [13]	Node	√	-	-	×	-
	JK-Net [42]	Node	√	-	-	√	-
	GIN [41]	Graph	√	×	√	×	-
	DiffPool [44]	Graph	√	√	√	×	-
	SAGPool [20]	Graph	√	√	√	√	-
NAS	GraphNAS [10]	Node	√	-	-	×	RL
	SNAG [50]	Node	√	-	-	√	RL
	RE-MPNN [16]	Graph	√	×	√	√	EA
	AutoGM [45]	Node	√	-	-	×	Bayesian
	SANE [51]	Node	√	-	-	√	Gradient Descent
	DSS [23]	Node	√	-	-	×	Gradient Descent
	GNAS [3]	Node	√	-	-	×	Gradient Descent
	PAS (proposed)	Graph	√	√	√	√	Gradient Descent

Search space

We provide a set of operations for each module.



b) 2-layer architecture backbone

Module name	Operations
Aggregation $\mathcal{O}_a$	GCN, GAT, SAGE, GIN, GRAPHCONV, MLP
Pooling $\mathcal{O}_p$	TOPKPOOL, SAGPOOL, ASAP, HOPPOOL_1, HOPPOOL_2, HOPPOOL_3, MLPPOOL, GCPOOL, GAPPOOL, NONE
Readout $\mathcal{O}_r$	GLOBAL_SORT, GLOBAL_ATT, SET2SET, GLOBAL_MEAN, GLOBAL_MAX, GLOBAL_SUM, ZERO
Merge $\mathcal{O}_m$	M_LSTM, M_CONCAT, M_MAX, M_MEAN, M_SUM

Pooling Module: Calculate the node scores and generate the coarsen graphs by selecting Top-K nodes.

$$\mathbf{S} = f_s(\mathbf{A}, \mathbf{H}), idx = \mathrm{TOP}_k(\mathbf{S}),$$
$$\mathbf{A}' = \mathbf{A}(idx, idx), \mathbf{H}' = \mathbf{H}(idx, :).$$

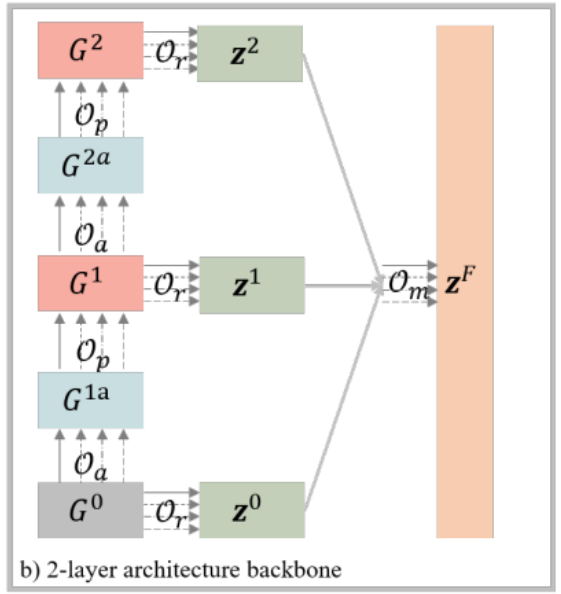
Search space size: $6^2 \times 10^2 \times 7^3 \times 5 = 6{,}174{,}000$

Differentiable search algorithm

- Differentiable search algorithm[1] is proposed to address the search efficiency problem.
- It relaxes the discrete search space into a continuous one by mixing the output of all candidate operations.
  - Relaxation function

$$\bar{o}(x) = \sum_{k=1}^{|\mathcal{O}|} c_k o_k(x) \qquad c_k \epsilon (0,1)$$

  - Mixed results in PAS
    - Aggregation: $\mathbf{H}^{la} = \sum_{i=1}^{|O_a|} c_i^{al} o_i(\mathbf{A}^{l-1}, \mathbf{H}^{l-1})$
    - Readout: $\mathbf{z}^{l} = \sum_{i=1}^{|O_r|} c_i^{rl} o_i(\mathbf{A}^{l}, \mathbf{H}^{l})$
    - Merge: $\mathbf{z}^{F} = \sum_{i=1}^{|O_m|} c_i^{m} o_i(\mathbf{z}^{0}, \mathbf{z}^{1}, \cdots, \mathbf{z}^{L})$



b) 2-layer architecture backbone

[1] Hanxiao Liu ret al. DARTS: Differentiable architecture search. ICLR 2019

Differentiable search algorithm

- It is unachievable to relax the pooling module since different coarse graphs contain diverse nodes and edges.

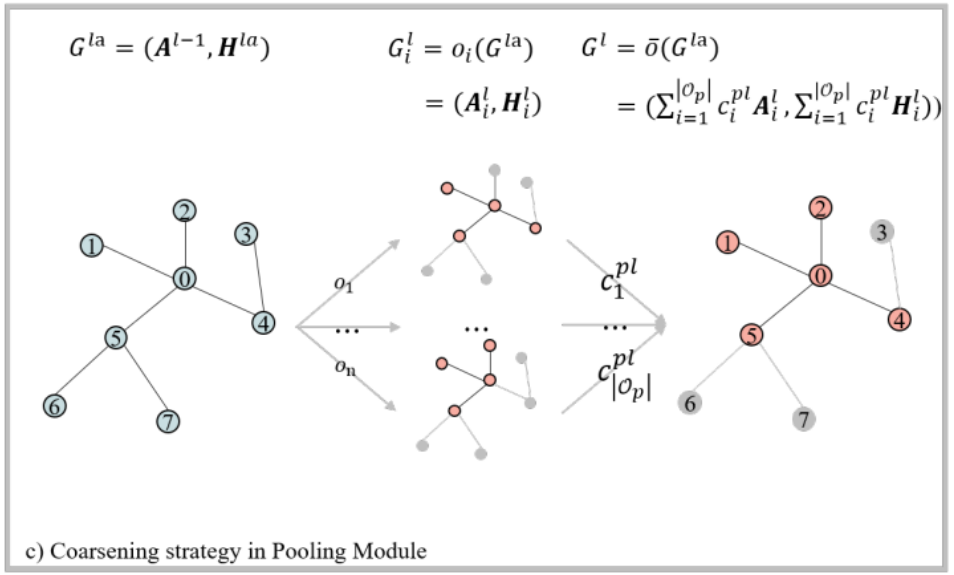
$V_1 = \{0,1,4,5\},\ A_1 \epsilon \mathbb{R}^{4\times4}, H_1 \epsilon \mathbb{R}^{4\times d}$

$V_n = \{0,1,2,5\},\ A_n \epsilon \mathbb{R}^{4\times4}, H_n \epsilon \mathbb{R}^{4\times d}$

Mismatched nodes

- Coarsening strategy
  - In pooling operations, keep the same "shape" when formulating the coarsen graphs.
  - For unselected nodes and edges (in grey), we set the features and weights to 0.

$$G^l = (A^l, H^l) = \left(\sum_{i=1}^{|O_p|} c_i^{pl} A_i^l, \sum_{i=1}^{|O_p|} c_i^{pl} H_i^l\right)$$



Outline

- Introduction
- Pooling architecture search
- Experiments
  - Experimental settings
  - Performance comparisons
  - Ablation study
  - The efficiency of PAS

Experimental settings

- Datasets
  - 6 real-world datasets from 3 domains.
- Baselines
  - Global pooling methods:
    - GNN + Global pooling
    - GNN + JK + Global pooling
    - DGCNN [1]
  - Hierarchical pooling methods:
    - ASAP[2] / SAGPOOL [3] / Graph U-Net [4]
    - DiffPool [5] / StructPool [6]
  - NAS methods:
    - RL based: GraphNAS [7] / SNAG [8]
    - EA / Random / Bayesian

Dataset	# of Graphs	# of Feature	# of Classes	Avg.# of Nodes	Avg.# of Edges	Domain
D&D	1,178	89	2	384.3	715.7	Bioinfo
PRO	1,113	3	2	39.1	72.8	Bioinfo
IMDB-B	1,000	0	2	19.8	96.5	Social
IMDB-M	1,500	0	3	13	65.9	Social
COX2	467	3	2	41.2	43.5	Chemistry
NCI109	4127	0	2	29.69	32.13	Chemistry

[1] Muhan Zhang et al. An end-to-end deep learning architecture for graph classification. AAAI 2018

[2] Ekagra Ranjan et al. ASAP: Adaptive Structure Aware Pooling for Learning Hierarchical Graph Representations. AAAI 2020

[3] Junhyun Lee et al. Self-Attention Graph Pooling. ICML 2019

[4] Gao H et al. Graph u-nets. ICML 2019

[5] Zhitao Ying et al. Hierarchical graph representation learning with differentiable pooling. NeurIPS 2018

[6] Hao Yuan et al. StructPool: Structured graph pooling via conditional random fields. ICLR 2020

[7] Yang Gao et al. Graphnas: Graph neural architecture search with reinforcement learning. IJCAI 2020

[8] Huan Zhao et al. Simplifying Architecture Search for Graph Neural Network. Arxiv 2020

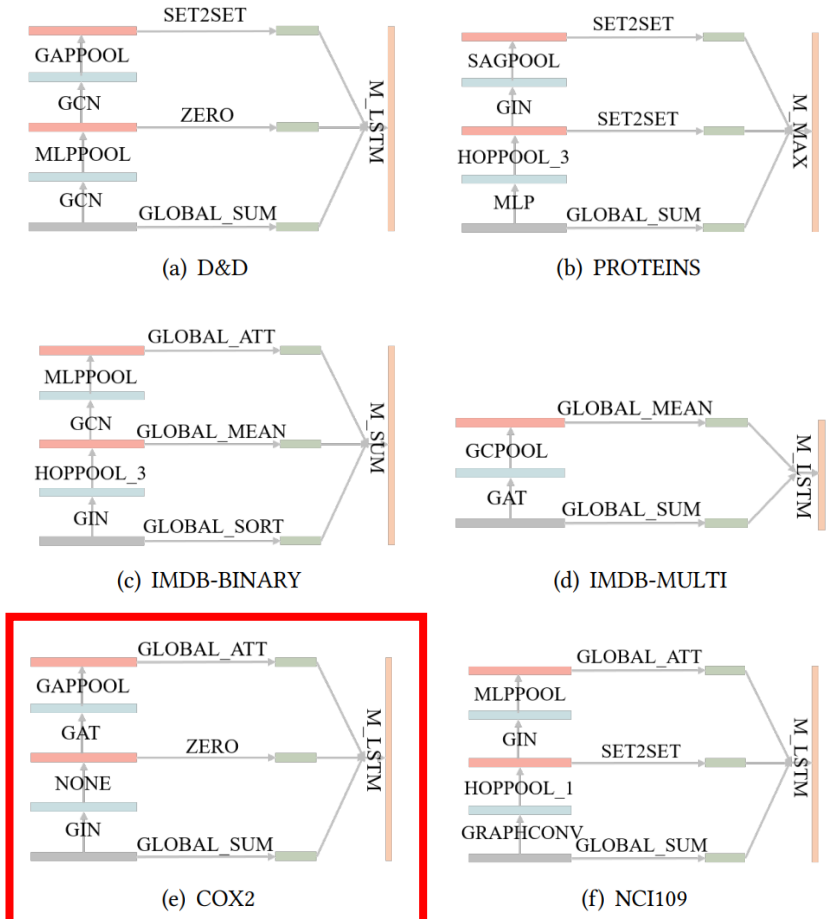
Performance comparisons

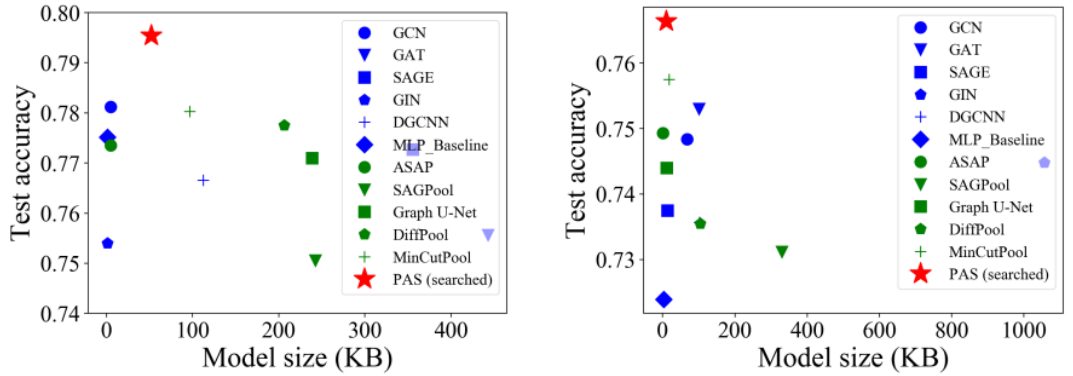
Table 4: Performance comparisons of PAS and all baselines. We report the mean test accuracy and the standard deviation by 10-fold cross-validation. The best results in different groups of baselines are underlined, and the best result on each dataset is in boldface.

	Method	D&D	PROTEINS	IMDB-BINARY	IMDB-MULTI	COX2	NCI109
Global pooling	GCN	0.7812±0.0433	0.7484±0.0282	0.7267±0.0642	0.5040±0.0302	0.7923±0.0219	0.7344 ± 0.0192
	GAT	0.7556±0.0372	0.7530±0.0372	0.7407±0.0453	0.4967±0.0430	0.8156±0.0417	0.7410 ± 0.0245
	SAGE	0.7727±0.0406	0.7375±0.0297	0.7217±0.0529	0.4853±0.0543	0.8031±0.0594	0.7553 ± 0.0164
	GIN	0.7540±0.0368	0.7448±0.0278	0.7167±0.0277	0.4980±0.0250	0.8309±0.0417	0.7456 ± 0.0210
	GCN-JK	0.7769±0.0235	0.7539±0.0517	0.7347±0.0445	0.4886±0.0496	0.7966±0.0226	0.7383 ± 0.0188
	GAT-JK	0.7809±0.0618	0.7457±0.0405	0.7327±0.0468	0.4953±0.0384	0.8072±0.0287	0.7172 ± 0.0292
	SAGE-JK	0.7735±0.0420	0.7494±0.0355	0.7287±0.0618	0.4973±0.0340	0.7990±0.0415	0.7325 ± 0.0356
	GIN-JK	0.7513±0.0395	0.7312±0.0451	0.7207±0.0486	0.5080±0.0302	0.8117±0.0467	0.7441 ±0.0220
	DGCNN	0.7666±0.0403	0.7357±0.0469	0.7367±0.0570	0.4900±0.0356	0.7985±0.0264	0.7506 ± 0.0165
	MLP-Baseline	0.7752±0.0390	0.7239±0.0353	0.7287±0.0520	0.4980±0.0428	0.7838±0.0104	0.6445 ± 0.0160
Hierarchical pooling	ASAP	0.7735±0.0415	0.7493±0.0357	0.7427±0.0397	0.5013±0.0344	0.8095±0.0320	0.7376 ± 0.0224
	SAGPool	0.7506±0.0506	0.7312±0.0447	0.7487±0.0409	0.4933±0.0490	0.7945±0.0298	0.6489 ± 0.0315
	Graph U-Net	0.7710±0.0517	0.7440±0.0349	0.7317±0.0484	0.4880±0.0319	0.8030±0.0421	0.7279 ± 0.0229
	DiffPool	0.7775±0.0400	0.7355±0.0322	0.7186±0.0563	0.4953±0.0398	0.7966±0.0264	0.7315 ± 0.0214
	MinCutPool	0.7803±0.0363	0.7575±0.0380	0.7077±0.0489	0.4900±0.0283	0.8007±0.0385	0.7405±0.0248
NAS	GraphNAS	0.7198±0.0454	0.7251±0.0336	0.7110±0.0230	0.4693±0.0364	0.7773±0.0140	0.7228 ± 0.0228
	GraphNAS-WS	0.7674±0.0455	0.7520±0.0251	0.7360±0.0463	0.4827±0.0350	0.7816±0.0058	0.7049 ± 0.0192
	SNAG	0.7223±0.0386	0.7053±0.0311	0.7250±0.0461	0.4933±0.0289	0.7903±0.0212	0.7090±0.0224
	SNAG-WS	0.7351±0.0303	0.7233±0.0244	0.7360±0.0516	0.5000±0.0248	0.8054±0.0381	0.7063±0.0160
	EA	0.7514±0.0301	0.7341±0.0298	0.7400±0.0412	0.4860±0.0405	0.7945±0.0159	0.7324±0.0126
	Random	0.7792±0.0482	0.7394±0.0423	0.7210±0.0554	0.4980±0.0398	0.8073±0.0231	0.7306±0.0241
	Bayesian	0.7555±0.0321	0.7314±0.0239	0.7270±0.0335	0.4980±0.0397	0.8029±0.0172	0.7204±0.0114
	PAS	**0.7896±0.0368**	**0.7664±0.0329**	**0.7510±0.0532**	**0.5220±0.0373**	**0.8344±0.0633**	**0.7684±0.0272**

- No absolute winner from human-designed models and PAS consistently outperforms all baselines.
- Compared with NAS baselines, the proposed differentiable search algorithm provide performance gain.

Searched architectures



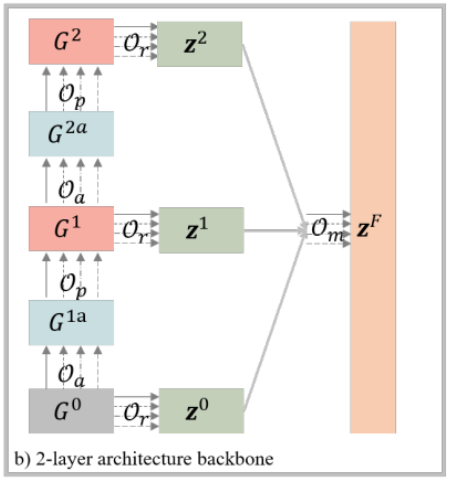


(a) D&D (b) PROTEINS

- Data-specific pooling architectures are obtained on each dataset.
- Global pooling architecture on COX2 is obtained by PAS.
- These searched architectures can achieve SOTA performance with moderate size in terms of the model parameters.

Ablation study

- Ablation studies on 4 modules:
  - Aggregation Module: fixed aggregators
  - Pooling Module: fixed NONE operation
  - Readout Module: fixed GLOBAL_MEAN operation
  - Merge Module: remove Merge Module



b) 2-layer architecture backbone

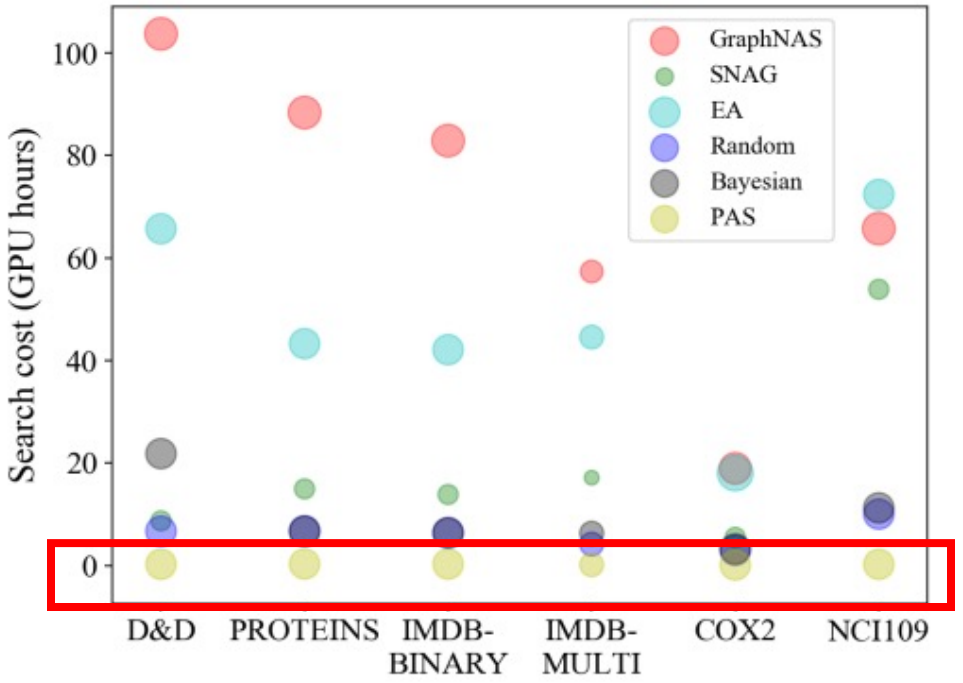
Fixed		D&D	IMDB-MULTI
Aggregation	PAS-GCN	0.7835±0.0407	0.5027±0.0409
	PAS-GAT	0.7878±0.0376	0.5087±0.0417
Pooling	PAS-Global	0.7708±0.0330	0.5173±0.0447
Readout	PAS-FR	0.7436±0.0472	0.5033±0.0436
Merge	PAS-RM	0.7682±0.0336	0.5047±0.0380
	PAS	**0.7896±0.0368**	**0.5220±0.0373**

Module name	Operations
Aggregation O_a	GCN, GAT, SAGE, GIN, GRAPHCONV, MLP
Pooling O_p	TOPKPOOL, SAGPOOL, ASAP, HOPPOOL_1, HOPPOOL_2, HOPPOOL_3, MLPPOOL, GCPOOL, GAPPOOL, NONE
Readout O_r	GLOBAL_SORT, GLOBAL_ATT, SET2SET, GLOBAL_MEAN, GLOBAL_MAX, GLOBAL_SUM, ZERO
Merge O_m	M_LSTM, M_CONCAT, M_MAX, M_MEAN, M_SUM

- Conclusion: It is important for graph classification to search for the combinations of operations from the four essential modules.

The efficiency of PAS

- Search cost of PAS
  - The size of each circle represents the size of the search space each method uses.
  - EA, Random and Bayesian baselines have the same search space as PAS.
  - GraphNAS focus on the design aggregation operations and SNAG provided additional merge modules.



- Conclusions:
  - PAS has the smallest search cost due to the designed coarsen strategy.
  - PAS helps to find SOTA performance compared with EA, Random and Bayesian.

Summary

- PAS learns data-specific pooling architectures for the graph classification task automatically.
- Design a unified framework consisting of four essential modules and provide a novel search space on top of this framework.
- A coarsen strategy is developed to relax the search space into continuous thus the differentiable algorithm can be utilized.
- Future work: investigate the connections between the graph properties and the learned pooling architectures.

Code

- Code: https://github.com/AutoML-Research/PAS
- More related methods: https://github.com/AutoML-Research
  - AutoSF: Searching Scoring Functions for Knowledge Graph Embedding (ICDE 2020)
  - DiffMG: Differentiable Meta Graph Search for Heterogeneous Graph Neural Networks (KDD 2021)
  - Search to aggregate neighborhood for graph neural network (ICDE 2021)

Thank you !
Q&A

weilanning18z@ict.ac.cn