ICML 2022

DFG-NAS: Deep Graph Neural Architecture Search

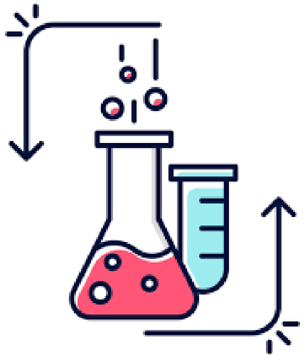
Wentao Zhang, Zheyu Lin, Yu Shen, Yang Li, Zhi Yang, Bin Cui

Peking University

Outline

1. Motivation

2. Method



3. Experiment

4. Conclusion

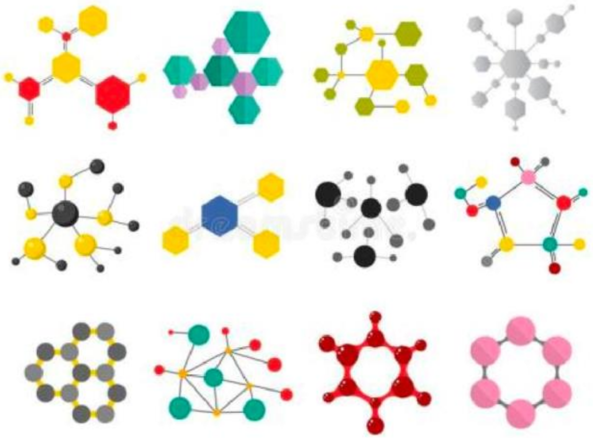
Motivation

Graph Neural Networks

Many real-world data are graphs.

Social Network

Knowledge Graph



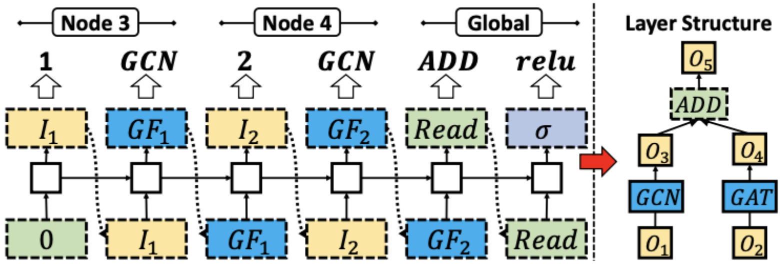
Drugs and New materials

- Graph Neural Networks (GNNs) have achieved great success in many graph-based applications.

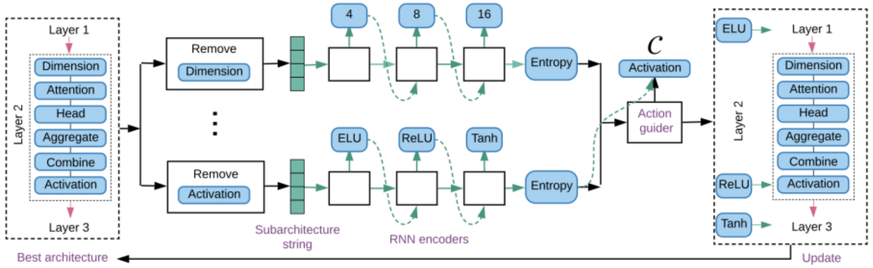
- The handcrafted GNN architectures can not behave well in all scenarios.

Automated Graph NAS

- Rich human expertise is required
  - Exploring a suitable GNN architecture in each scenario requires tremendous laborious trials and rich human expertise.
  - Graph Neural Architecture Search (G-NAS) methods are emerged to enable automatic design of the best graph neural architecture.



GraphNAS[1]



AutoGNN[2]

[1] Gao Y, Yang H, Zhang P, et al. Graph neural architecture search. *IJCAI 2021*: 1403-1409.

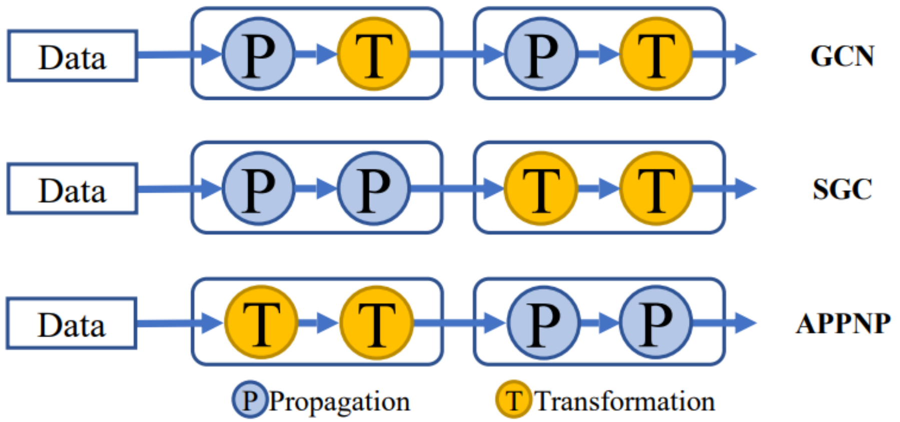
[2] Zhou K, Song Q, Huang X, et al. Auto-gnn: Neural architecture search of graph neural networks. *arXiv preprint* arXiv:1909.03184, 2019.

Drawbacks

Drawbacks of current G-NAS methods

1) Fixed Pipeline Pattern.

Existing methods adopt a fixed message-passing pipeline to organize two operations: **propagating** (P) representations of its neighbors and applying **transformation** (T) on the representations.



P-T-P-T : most G-NAS methods adopt the tight entanglement of applying transformation after propagation in each layer.
P-P-T-T or T-T-P-P : Several methods do the transformation or propagation first.

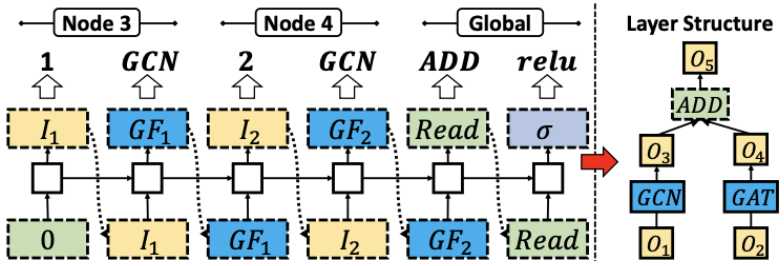
Specific P-T permutations and combinations are still fixed pipeline designs, limiting the expressive power of macro-architecture search space!

Drawbacks

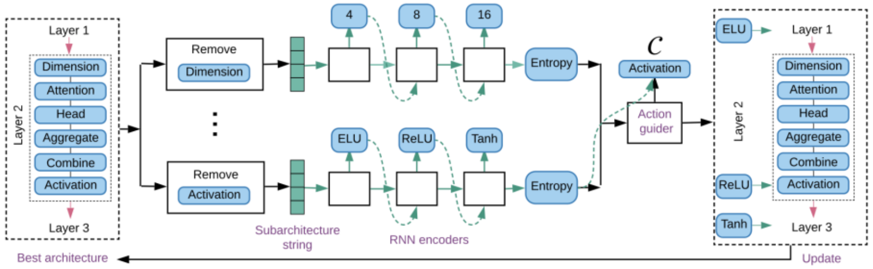
Drawbacks of current G-NAS methods

2) Restricted Pipeline Depth

- The performance decreases as the layers become deeper
- the existing G-NAS methods fix the number of layers to a small constant



GraphNAS



AutoGNN

Both AutoGNN and GraphNAS pre-define a very restricted GNN layer number (e.g., ≤ 3)

Observation

- The meaning of deep GNN
  - In fact, there are two potential benefits for deep GNN.
    - 1) Information Propogation: Shallow architecture can not involve the full graph information due a few propagations. (especially when the label, feature or edges is sparse)
    - 2) Nonlinear Transformation: The expressive power is low due to a few nonlinear transformation.

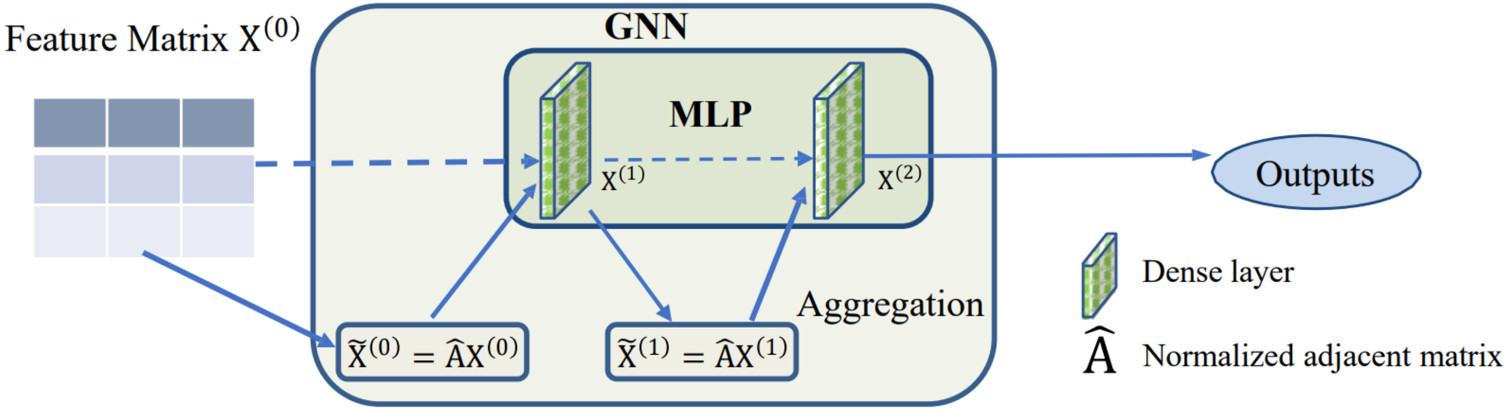


Figure 1. The framework of a two-layer GCN models.

Observation

The characteristics of P and T

- Different kind of datasets needs different pipeline patterns.

Table 1. Test accuracy of GNNs with different PT orders.

Methods	Cora	Citeseer	PubMed
PPTT	83.4±0.3	72.2±0.4	78.5±0.5
TTPP	82.8±0.2	71.8±0.3	79.8±0.3
PTPT	81.2±0.6	71.2±0.4	79.1±0.2

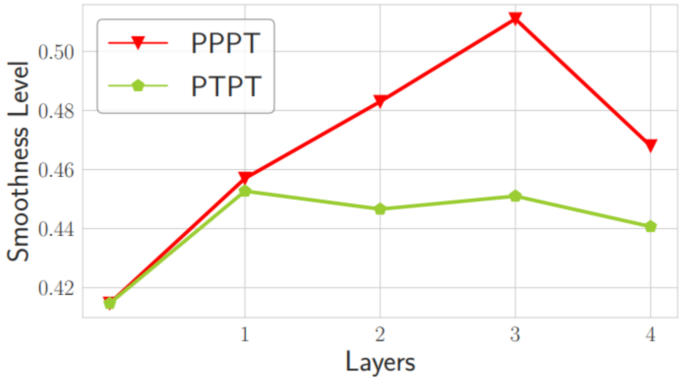


Figure 2. Smoothness of different **PT** orders.

- the smoothness increases, i.e., the node embedding becomes similar, by applying the P operation
- the smoothness decreases by applying the T operation, which implies that the T operation has the ability to alleviate the over-smoothing issue.

Outline

Method

Design Space

A new paradigm and design space

- The design space includes P-T permutations and combinations, and the number of P-T operations.

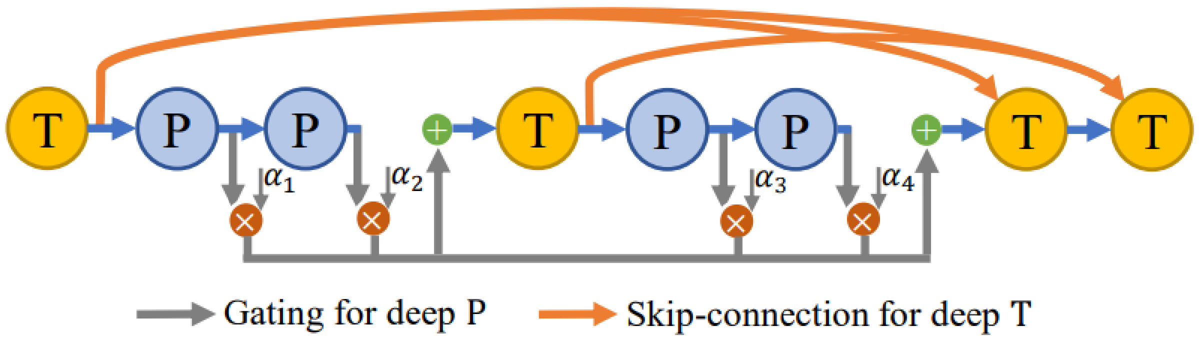


Figure 3. GNN pipeline example in the search space of DFG-NAS.

Propagation Connection: if the next operation is **T**, we assign a node-adaptive combination weight for the node embeddings propagated by all previous **P** operations.

Transformation Connection: the input of each T operation is the sum of the output of the last layer and the outputs of all previous T operations before the last layer.

Search Method

We adopt the evolutionary algorithm for G-NAS.

- Each GNN architecture is encoded as a sequence consisting of the **P** and **T** operations.

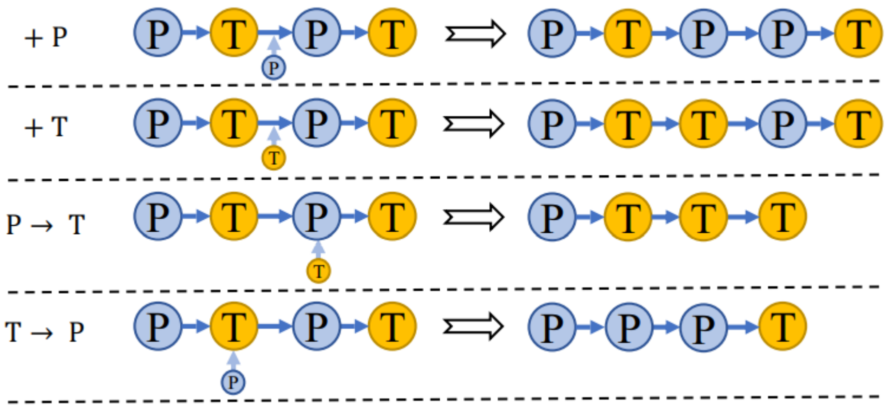


Figure 4. Overview of four different mutations.

Case 1 + P: Add a propagation operation.

Case 2 + T: Add a transformation operation.

Case 3 P → T: Replace a propagation operation by a transformation operation.

Case 4 T → P: Replace a transformation operation by a propagation operation.

Outline

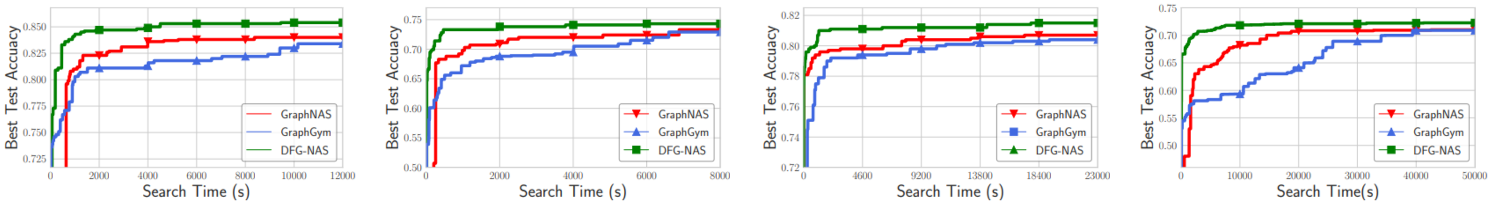
Experiments

Comparison with Existing GNNs

- DFG-NAS obtains the deepest architectures with high expressive power and achieves the best performance on all four datasets
- DFG-NAS consistently outperforms the compared G-NAS methods

Table 2. Test accuracy on the node classification task.

Methods	Cora	Citeseer	PubMed	ogbn-arxiv
Alternate P and T				
GCN	81.3±0.6	71.1±0.1	78.8±0.4	71.7±0.3
GAT	82.9±0.2	70.8±0.5	79.1±0.1	71.9±0.2
GraphSAGE	79.2±0.6	71.6±0.5	77.4±0.5	71.5±0.3
T before P				
APPNP	83.1±0.5	71.8±0.4	80.1±0.2	72.0±0.1
AP-GCN	83.4±0.3	71.3±0.5	79.7±0.3	71.9±0.2
DAGNN	84.3±0.2	73.3±0.6	80.5±0.5	72.0±0.3
P before T				
SGC	81.7±0.2	71.3±0.2	78.8±0.1	71.6±0.3
SIGN	82.1±0.3	72.4±0.8	79.5±0.5	71.9±0.1
S^2GC	82.7±0.3	73.0±0.2	79.9±0.3	71.8±0.3
G-NAS Methods				
GraphNAS	83.7±0.4	73.5±0.3	80.5±0.3	71.7±0.2
AutoGNN	83.6±0.3	73.8±0.7	79.7±0.4	/
GraphGym	83.5±0.2	73.4±0.3	80.3±0.2	71.6±0.3
DFG-NAS	**85.2±0.2**	**74.1±0.4**	**81.1±0.3**	**72.3±0.2**

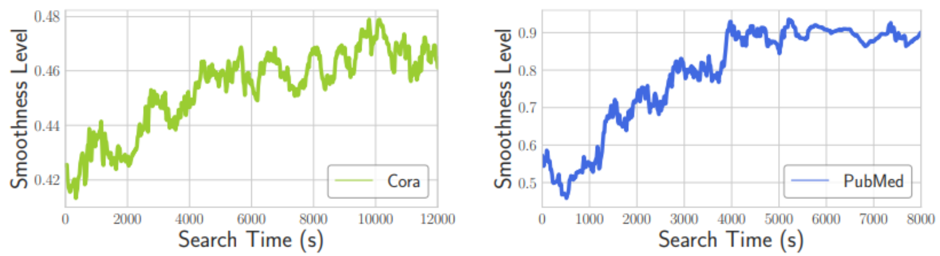


(a) Cora (b) Citeseer (c) Pubmed (d) ogbn-arxiv

Figure 5. Test results during neural architecture search on four datasets.

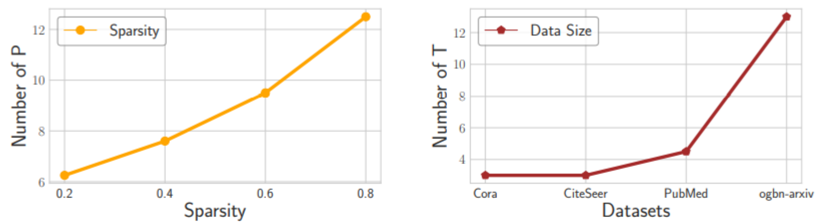
Interpretability

- Though graph datasets differ in the requirement of smoothness, both two curves are generally saturating.
- The average number of **P** in top-10 architectures increases when the dataset grows sparser.
- The average number of **T** is similar in Core and Citeseer, and the number further on larger datasets PubMed and ogbn-arxiv.



(a) Cora (b) Pubmed

Figure 6. Average smoothness over iterations on two datasets.



(a) Trend of **P** on sparsity (b) Trend of **T** on data size

Figure 7. Left: Average number of **P** operations along with the increased sparsity on PubMed. Right: Average number of **T** operations along with the increased nodes of different datasets.

Outline

Conclusion

Conclusion

1. By decoupling the P and T operations, DFG-NAS suggests a transition from studying specific fixed GNN pipelines to studying the GNN pipeline design space.

2. By further adding gating and skip-connection mechanisms, DFG-NAS could support both deep propagation and transformation, which has the ability to explore the best architecture design to push forward the GNN performance boundary.

3. Empirical results demonstrate that DFG-NAS achieves an accuracy and efficiency improvement over state-of-the-art G-NAS methods.

Thanks