Graph Transformer Networks

# A Brief Explanation of GTN

By Li Ziming

# Contents

1

# Research Background And Significance

1

# Background

- ▲ Recent years, GNNs have been used in various tasks such as graph classification, link prediction and so on and perform well.
- ▲ GNNs are designed to be used on fixed and homogeneous graph. But many networks such as citation networks are heterogeneous graph.
- ▲ We can design meta-paths manually to solve this problem, but the results will be significantly affected by the choice of these meta-paths.
- ▲ Based on these, GTN was developed learns to transform a heterogeneous input graph into useful meta-path graphs for each task and learn node representation.

1

## Significance

▲ Propose a novel framework GTN and it can learn a new graph structure which involves identifying useful meta-paths and multi-hop connections for learning effective node representations.

▲ The graph generation is interpretable and the model is able to predict effective meta-paths.

▲ The node representation learnt by GTN resulting in best performance in node classification tasks without using domain knowledge.

2

# Model And Key Methologies

2

Preliminaries

Meta-Path Generation

Trick

GTN

# Preliminaries

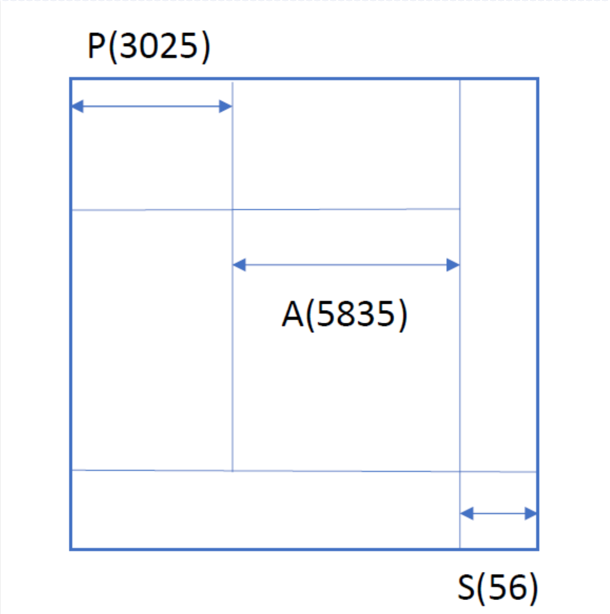
$\mathrm{T}^e$: The set of node edge types

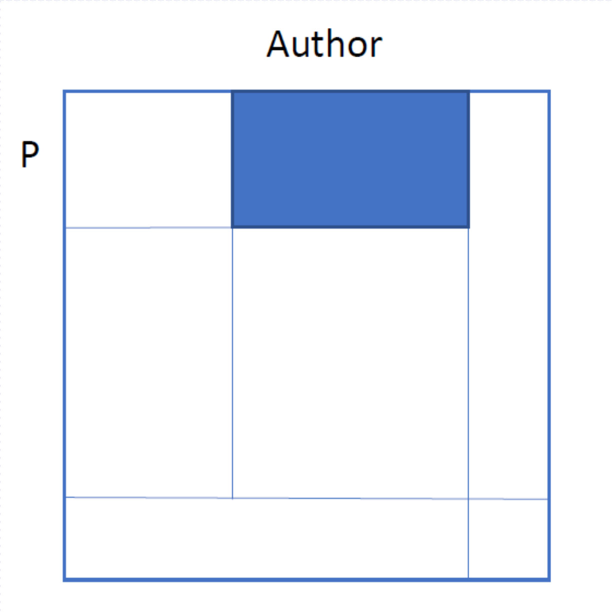
$\mathrm{T}^v$: The set of node types

$\{A_k\}_{k=1}^{K}$: Adjacency Matrices, $K = |T^e|$

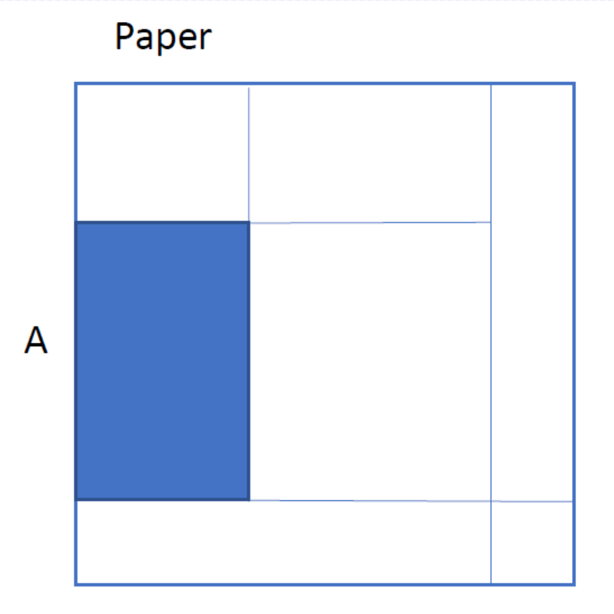
ACM data set: Paper(3025), Author(5835), Subject(56)

$\mathrm{T}^e: \{PA, AP, PS, SP\}$




$A_{PA} = Adjacency$




$A_{AP} = Adjacency$

# Meta-Path Generation

First we need ask two questions:

- What is meta-path?
- How can we aggregate meta-path?

Meta-Path:

$p$ is a path on the heterogeneous graph G that is connected with heterogeneous edges, $v1 \xrightarrow{t1} v2 \xrightarrow{t2} \ldots \xrightarrow{tl} v_{l+1}$, where $t_l \in \mathrm{T}^e$denotes an $l-th$ edge type of meta-path.

We have a set of edge,$\mathrm{T}^e$: $\{PA, AP, PS, SP\}$ .

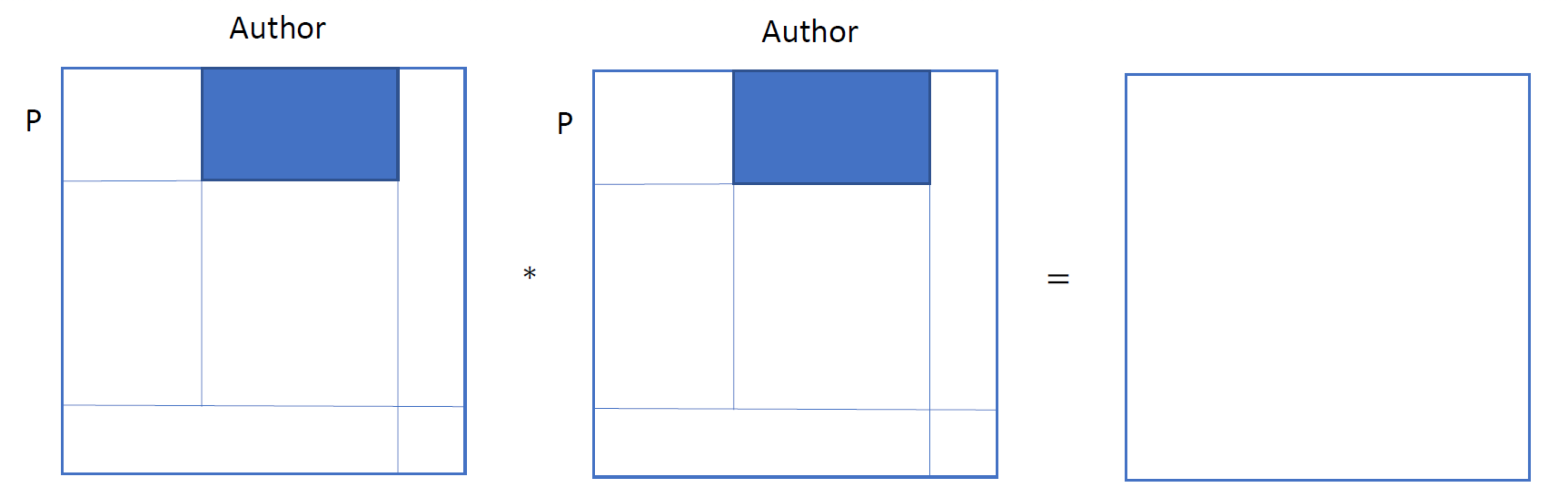
We assume that we have one meta-path as PAP. Then we can get the representation of it by this way:

$$meta-path1 = \{PAP\} = \{PA, AP\} = Adj_{PA} * Adj_{AP}$$

We multiply the matrix in the path.

# Meta-Path Generation

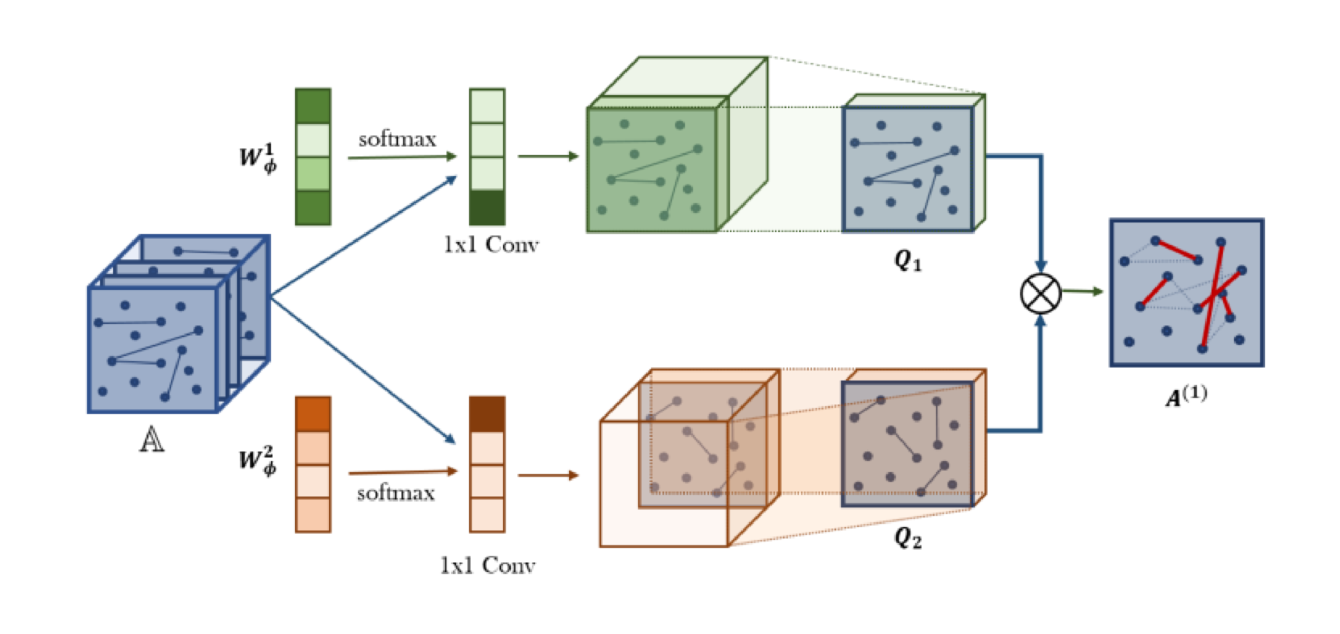
And we can see some interesting thing
when we try to get a meta-path that doesn't exist:




$A_{PA} = Adjacency$ $\qquad$ $A_{PA} = Adjacency$

$meta - path = \{PA, PA\} = \ Adj_{PA} \ * \ Adj_{PA} = Nan$ ,we get a null matrix, which suggests this method is right.

# Meta-Path Generation




And we get $A^{(1)}$ by multiply $Q_1, Q_2$, to prevent it from been too large, we do normalization to the result with its degree matrix.

$$A^{(1)} = D^{-1}Q_1Q_2$$

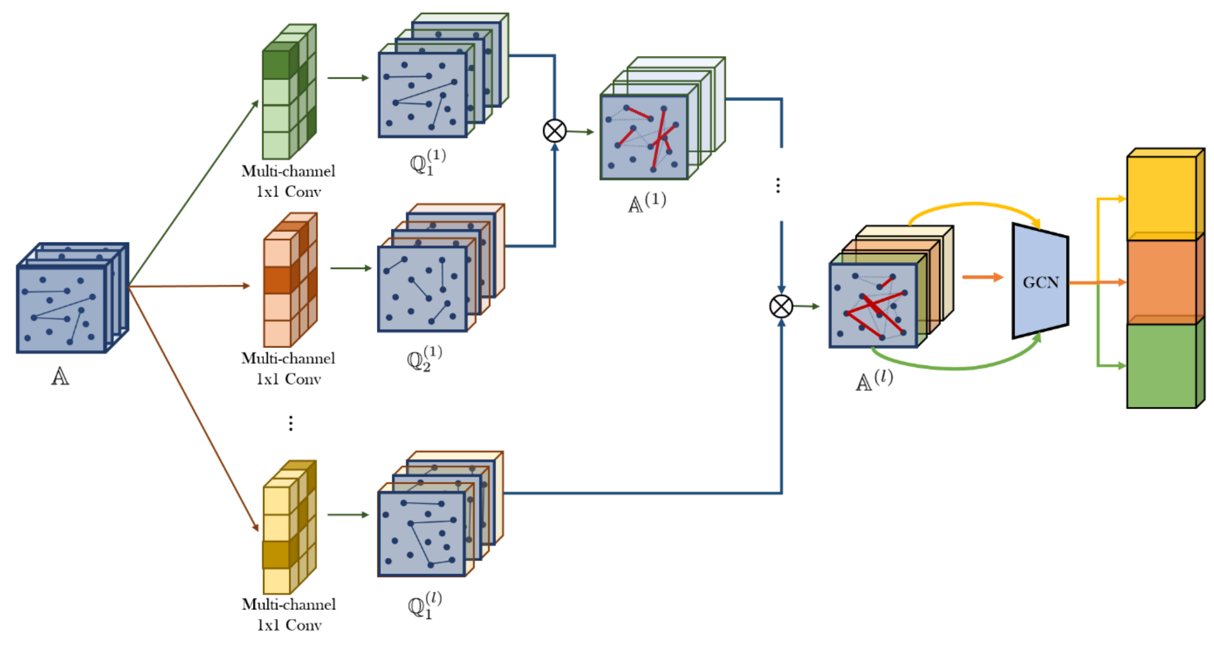
We have a set of edge,$\mathrm{T}^e$: $\{PA, AP, PS, SP\}$ , Adjacency matrices: $\mathbb{A} = \{A_1, A_2, A_3, A_4\}$,

Parameters of convolution layers: $W_\phi^1 = \{\alpha_1^1, \alpha_2^1, \alpha_3^1, \alpha_4^1\}, W_\phi^2 = \{\alpha_1^2, \alpha_2^2, \alpha_3^2, \alpha_4^2\}$

And then we get matrix $Q$ by:

$$Q = F\left(\mathbb{A}; W_\phi\right) = \phi(\mathbb{A}; softmax(W_\phi))$$

# GTN




After the stack of $l$ GT layers, a GCN is applied to each channel of meta-path tensor $\mathbb{A}^{(l)} \epsilon R^{N*N*C}$, and multiple node representations are concatenated as

$$Z = \overset{C}{\underset{i=1}{\|}} \sigma(\tilde{D}_i^{-1} \tilde{A}_i^{(l)} XW)$$

To consider multiple types of meta-paths simultaneously, we set the output channels of this 1 * 1 convolution to $C$, and $Q$ become tensor with dimension $N * N * C$ instead of matrix. It is beneficial to learn different node representations via multiple different graph structures.

# Trick

We have almost finished the introduction to GTN, but there is still one question. It was generated after I read the paper:

- We can see that as soon as we do the matrix multiplication, we get a meta-path of length at least 2, what to do with the meta-path represented by the original edges?

It is significant, so the authors didn't ignore it. To preserve the features of the graph itself, they added an identify matrix $I$ to the matrix $A$, because of

$$I * A_1 = A_1$$

In this way they solved this problem, and when they did comparison experiment, they found that adding this identify matrix or not makes a big difference.

# 3

# Experiments

# 3

① Are the new graph structures generated by GTN effective for learning node representation?

② Can GTN adaptively produce a variable length of meta-paths depending on datasets?

③ How can we interpret the importance of each meta-path from the adjacency matrix generated by GTNs?

# Results on Node Classification

Table 2: Evaluation results on the node classification task (F1 score).

| | DeepWalk | metapath2vec | GCN | GAT | HAN | GTN$_{-I}$ | **GTN (proposed)** |
|---|---|---|---|---|---|---|---|
| DBLP | 63.18 | 85.53 | 87.30 | 93.71 | 92.83 | 93.91 | **94.18** |
| ACM | 67.42 | 87.61 | 91.60 | 92.33 | 90.96 | 91.13 | **92.68** |
| IMDB | 32.08 | 35.21 | 56.89 | 58.14 | 56.77 | 52.33 | **60.92** |

By analysing the result of our experiment represented on Table 2, we will answer the research Q1 and Q2. GTN achieves the highest performance on all the datasets against all network embedding methods and graph neural network methods.

GTN model achieved the best performance compared to all other baselines on all the datasets even though the GTN model uses only one GCN layer whereas GCN, GAT and HAN use at least two layers. It demonstrates that the GTN can learn a new graph structure which consists of useful meta-paths for learning more effective node representation.

# Interpretation of Graph Transformer Networks

| Dataset | Predefined Meta-path | Meta-path learnt by GTNs | |
|---|---|---|---|
| | | Top 3 (between target nodes) | Top 3 (all) |
| DBLP | APCPA, APA | APCPA, APAPA, APA | CPCPA, APCPA, CP |
| ACM | PAP, PSP | PAP, PSP | APAP, APA, SPAP |
| IMDB | MAM, MDM | MDM, MAM, MDMDM | DM, AM, MDM |

By analysing the result of our experiment represented on Table 3, we will answer the research Q3. Comparison with predefined meta-paths and top-ranked meta-paths by GTNs. GTN found important meta-paths that are consistent with pre-defined meta-paths between target nodes (a type of nodes with labels for node classifications). Also, new relevant meta-paths between all types of nodes are discovered by GTNs. For example, in the DBLP dataset GTN ranks CPCPA as most importance meta-paths, which is not included in the predefined meta-path set. It makes sense that author's research area (label to predict) is relevant to the venues where the author publishes.

4

# Conclusion And My Thoughts

4

95%

## Conclusion

GTN transforms a heterogeneous graph into multiple new graphs defined by meta-paths with arbitrary edge types and arbitrary length up to one less than the number of Graph Transformer layers while it learns node representation via convolution on the learnt meta-path graphs. The learnt graph structures lead to more effective node representation resulting in state-of-the art performance, without any predefined meta-paths from domain knowledge.

Interesting future directions include studying the efficacy of GT layers combined with different classes of GNNs rather than GCNs and so on.

4

100%

## My Thoughts

When I read the Experiments part of the paper, I found that simple GAN performs better than HAN on all three datasets, and the difference with GTN is small. GAN is GCN with attention mechanism, and doesn't consider the kind of edges or the kind of nodes. However, it performs better than HAN which takes these into account and uses the attention mechanism. This makes me think about the meaning of meta-path, deep learning is data-driven learning, maybe GAN digs some information that can represent the node type or edge type during training, and from the result it is better than setting meta-path. The improvement is also very limited. Of course, I am amazed by the way GTN automatically mines the meta-path and handles heterogeneous graphs.

# Thanks!

by Li Ziming