TRANSFORMERS FOR GRAPHS

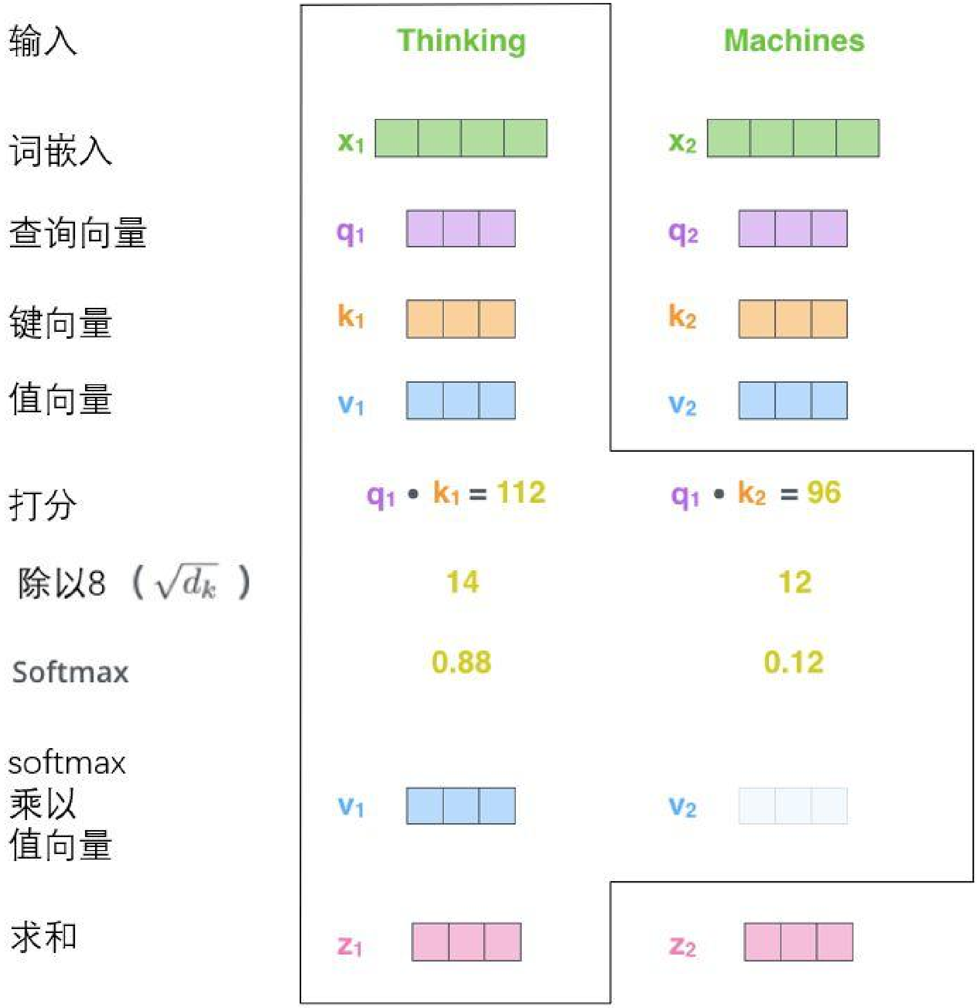
Chaojie Ji

INTRODUCTION

PROBLEMS OF GNNS

- The Message Passing paradigm is bounded by the Weisfeiler-Lehamn isomorphism hierarchy
- Over-smoothing problem caused by repeated local aggregation,
- Over-squashing problem due to the exponential computation cost with the increase of model depth

TRANSFORMERS



PROBLEMS OF TRANSFORMERS

- Target node neglects its local neighborhood, which causes over-fitting on large graphs

- Global receptive field of Transformer is costly

MAINSTREAM METHODS

- GNNs as An Auxiliary Module
  Directly injects GNNs into Transformer architecture

- Improved Positional Embedding from Graphs
  Compresses the graph structure into positional embedding vectors

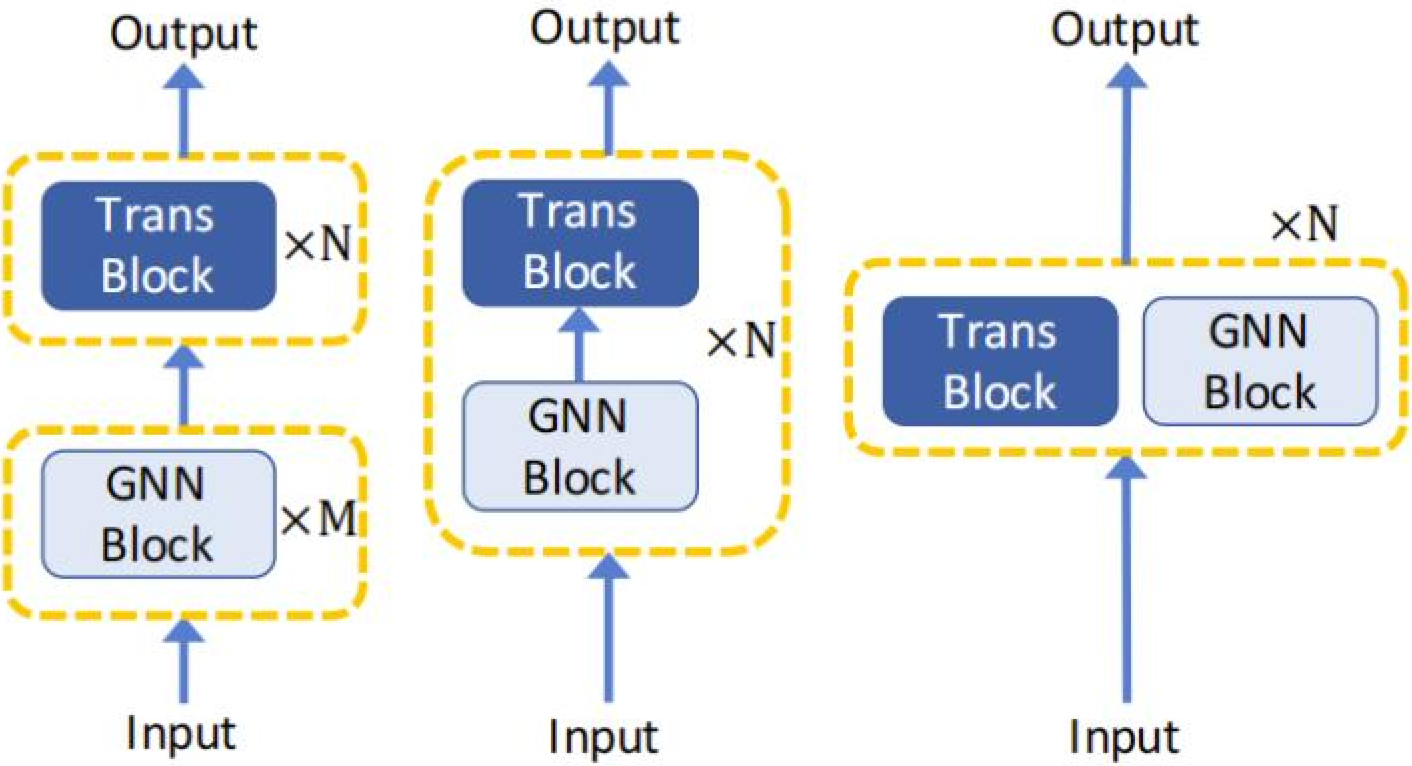
- Improved Attention Matrices from Graphs
  Injects graph bias terms into the attention computation, or narrow the receptive field

METHODS

GNNS AS AN AUXILIARY MODULE

- Pros
  - Remain the original pros of GNNs and Transformers
- Cons
  - Difficult to determine the best architecture to incorporate GNNs
  - A massive number of effort taken in hyper-parameter searching

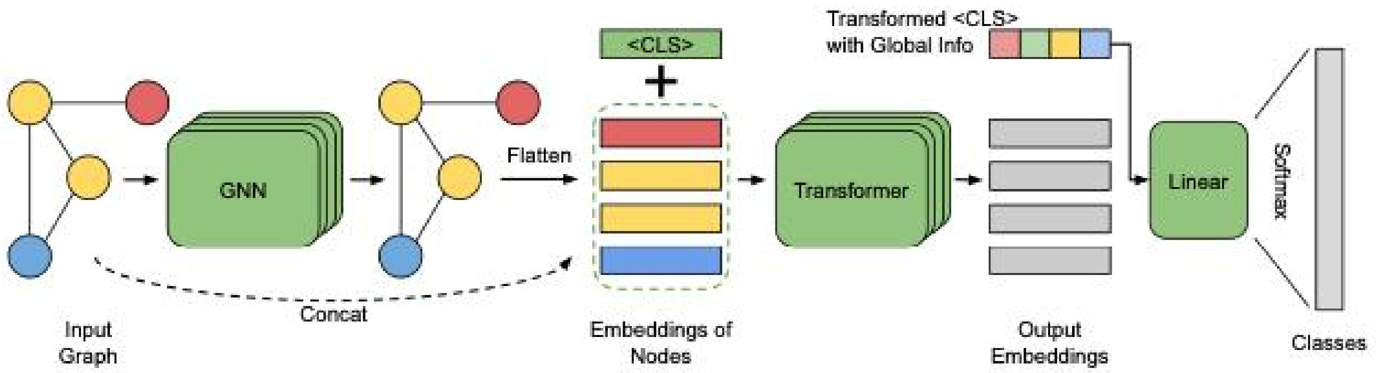
MAIN PATTERNS



(a) Before Trans (b) Alternatively (c) Parallel

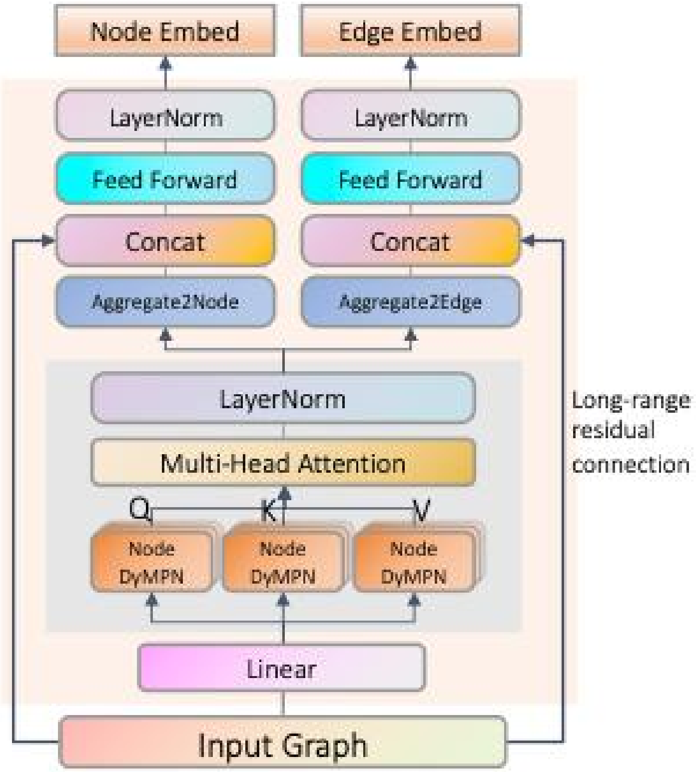
BEFORE TRANS

- [1] GraphTrans (NIPS-2021)
    - GNNs are applied to learn local representations
    - Transformer subnetwork explicitly computes all pairwise node interactions



BEFORE TRANS

- [1] GROVER (NIPS-2020)
  - Uses 2 GTransformer modules to represent node-level and edge-level features.
  - The inputs are first fed into a GNN to extract vectors as queries, keys, and values

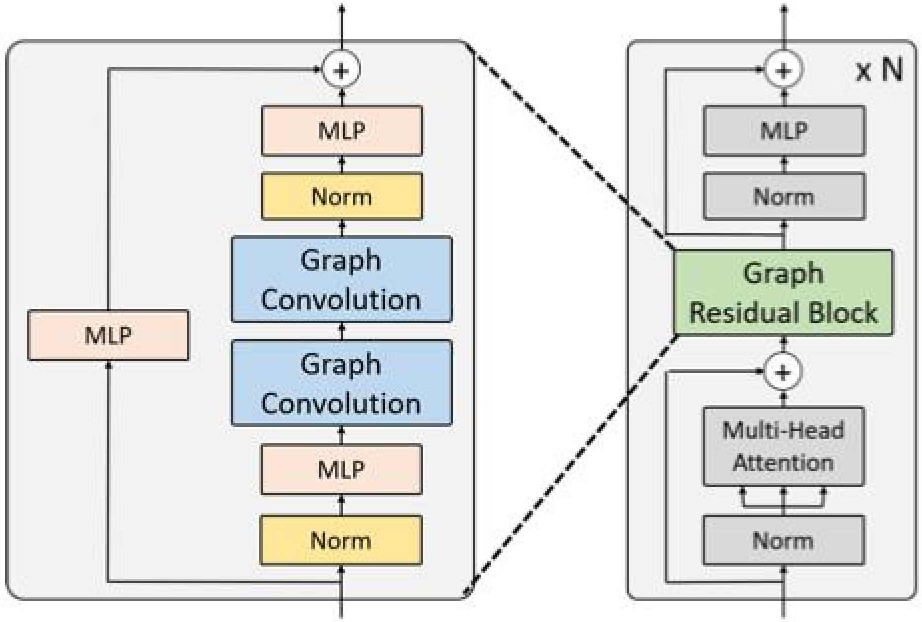


BEFORE TRANS

- [3] GraphiT (2021)
    - Adopts a GNN layer to produce a structure-aware representaion
    - Concatenate them as the input of Transformer Architecture

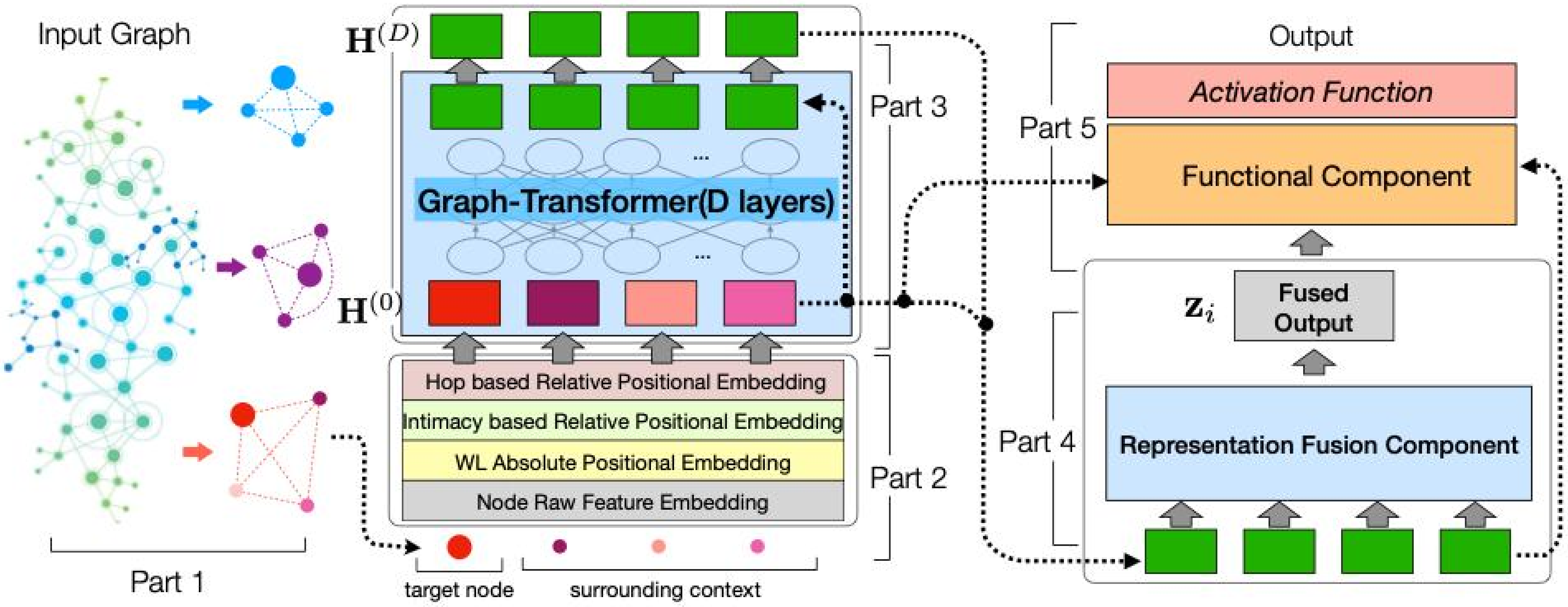
ALTERNATIVELY

- [4] Graphormer (ICCV 2021)
  - proposes a Graph Residual Block
  - MHSA is used to generate contextualized features
  - A GNN is applied to improve the local interactions

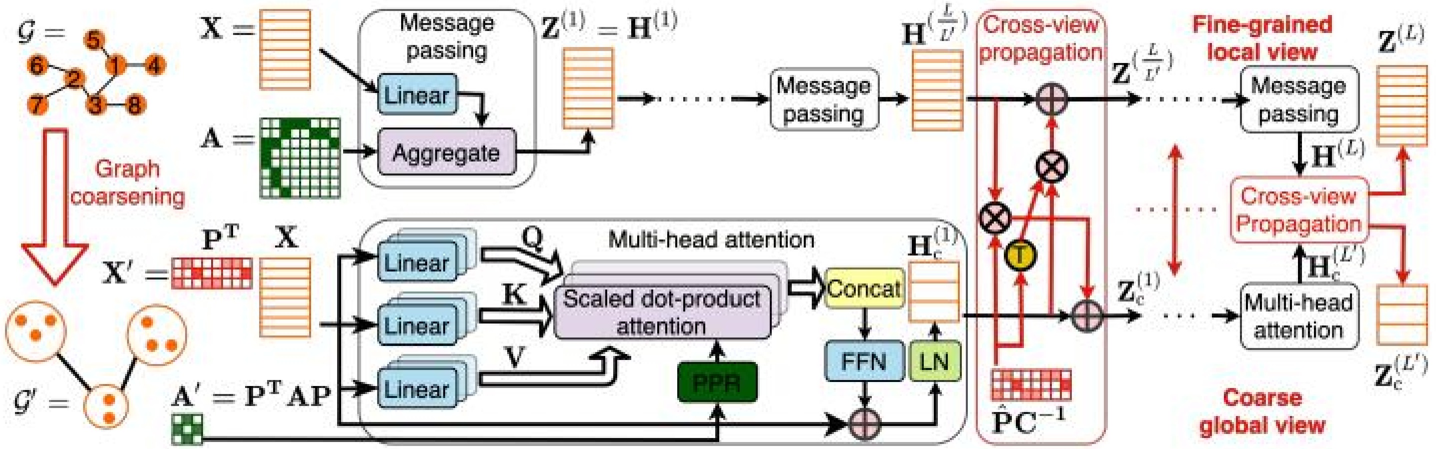


PARALLEL

- [5] Graph-BERT (ICCV 2021)



PARALLEL



- [14] Coarformer (2021)
  - It coarsens an input graph into partitions which are regarded as multiple super-nodes
  - Adjancy matrix and feature matrix are constructed fro these partitions
  - A GNN and a Transformer are utilized to learn its representation
  - A fusion model is designed to fuse these information

METHODS

IMPROVED POSITIONAL EMBEDDING FROM GRAPHS

- Pros
  - Convinent
- Cons
  - Copressing graph structure into fixed-sized vectors results in information loss

DENSE EMBEDDINGS

- [6] Graph Transformer (AAAI-2021)
    - Laplacian eigenvactors are derived by the factorization of the graph matrix
    - Eigenvectors of the k smallest non-trivial eigenvalues are regarded as the positional embeddings
- [7] EGT (KDD-2022)
    - SVD are pre-computed from graph structural matrix
    - Largest k singukar values and corresponding left and right singular vectos

DENSE EMBEDDINGS

- [10] Graph Transformer (2021)
    - Rethinks the proposed Eigen PE and designs a learned positional encoding that takes advantage of full Laplacian spectrum

HEURISTIC EMBEDDINGS

- [8] Degree PE (2021)
  - proposes a degree PE that measures the degree centrality
- [9] (AAAI 2020)
  - utilizes tree structure. It adopts a distance from the root node as a flag of the importance

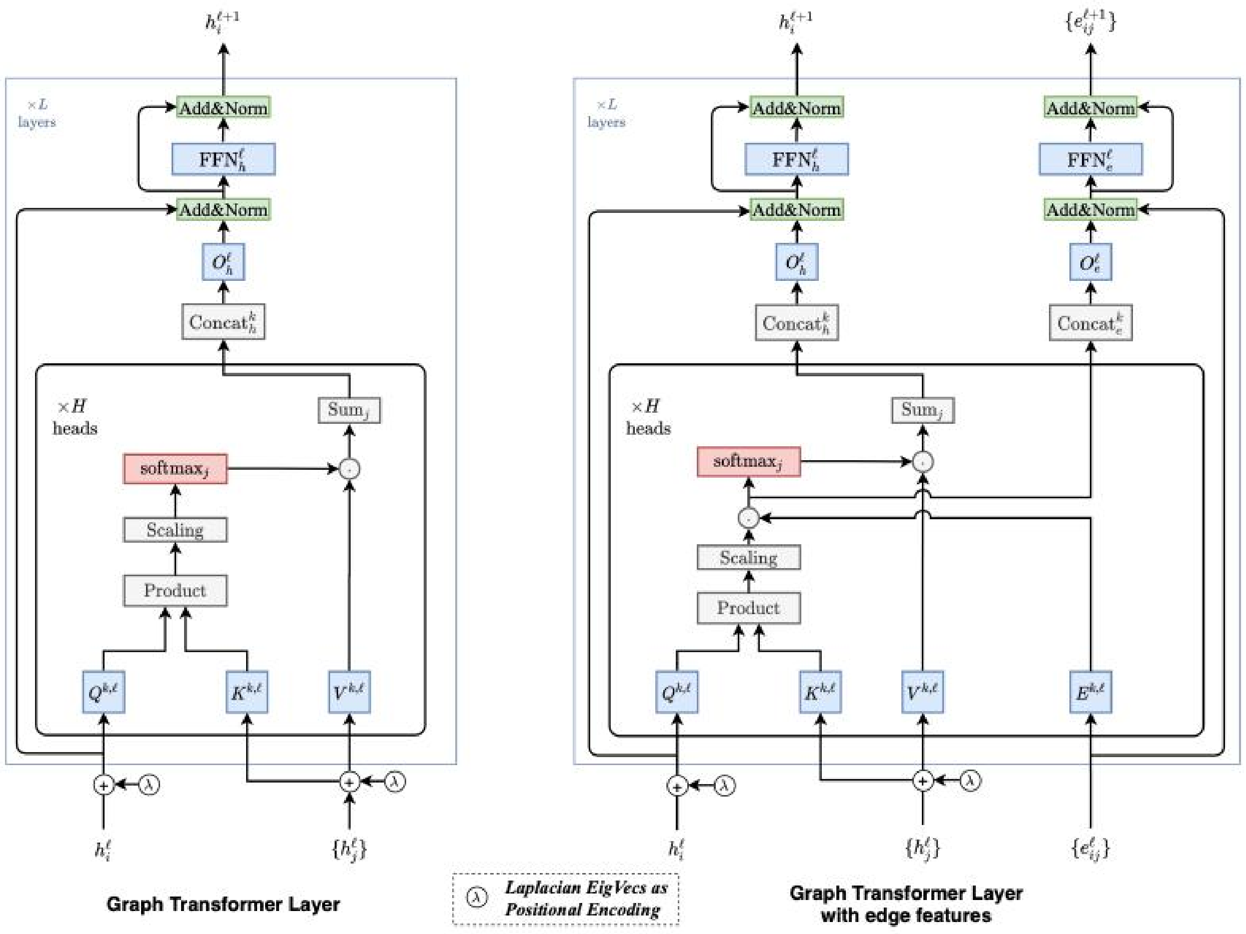
METHODS

IMPROVED ATTENTION MATRICES FROM GRAPHS

- Pros
  - Efficient
- Cons
  - Several pre-procession steps

RESTRICTING A NODE ONLY ATTENDING TO LOCAL NEIGHBORS

- [6] Graph Transformer (AAAI-2021)
  - The representation or the existence of an edge informs model which nodes should be attended to for a query node

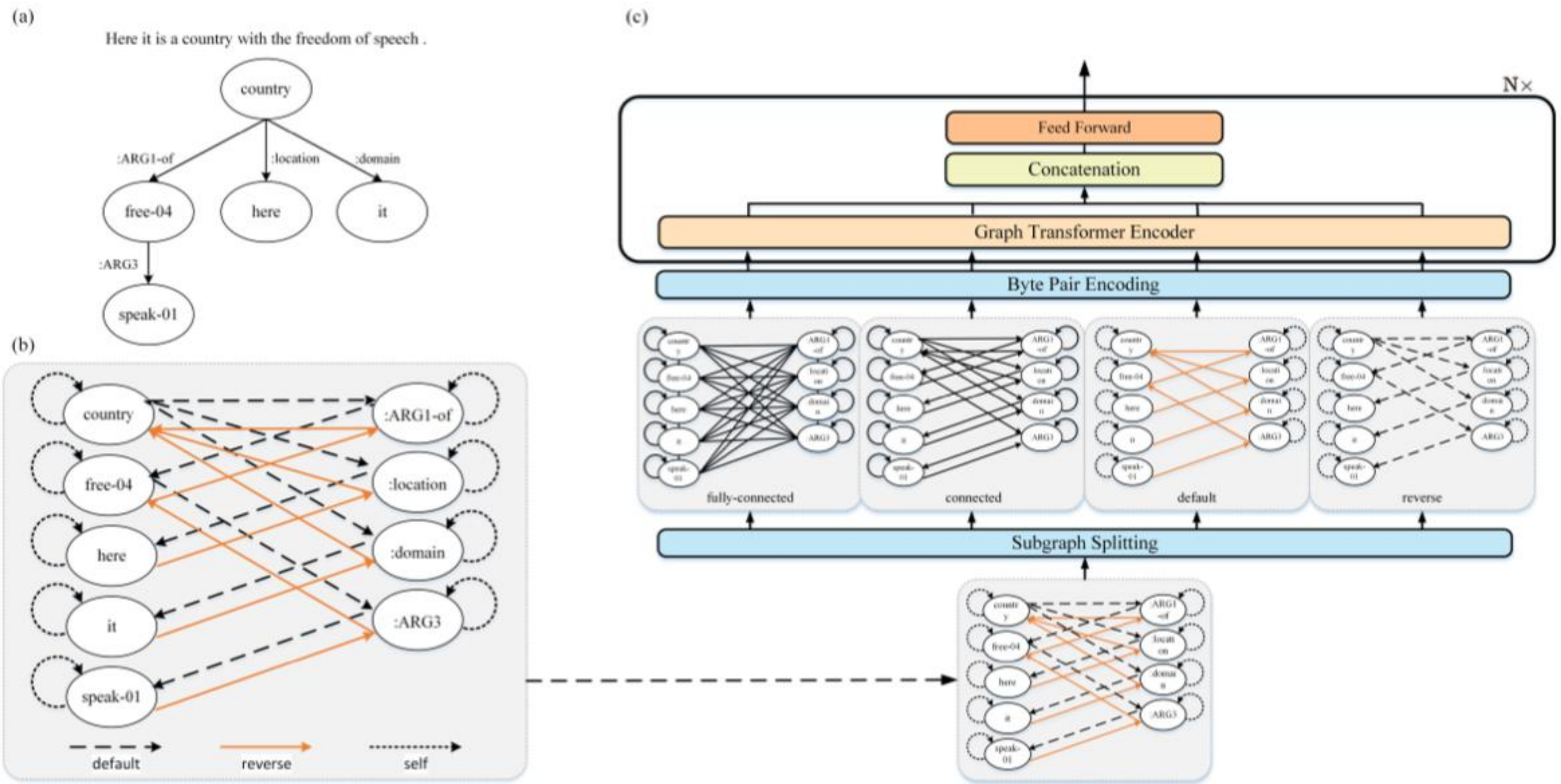


ENFORCE THE HEADS TO ATTEND TO DIFFERENT SUBSPACES

- [8] Graph Transformer (ACL-2020)
  - constructs Levi graphs in accordance to an input graph
  - splits this graph into multuiple subgraphs according to edge types
  - The corresponding adjacent matrices are assigned to different attention heads

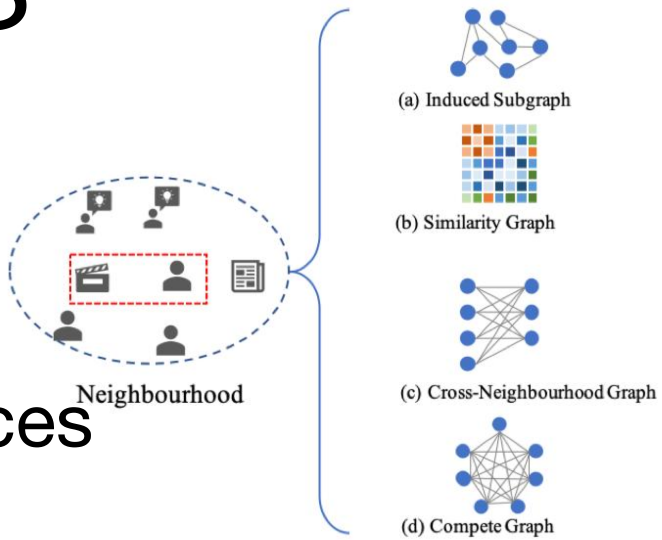
ENFORCE THE HEADS TO ATTEND TO DIFFERENT SUBSPACES

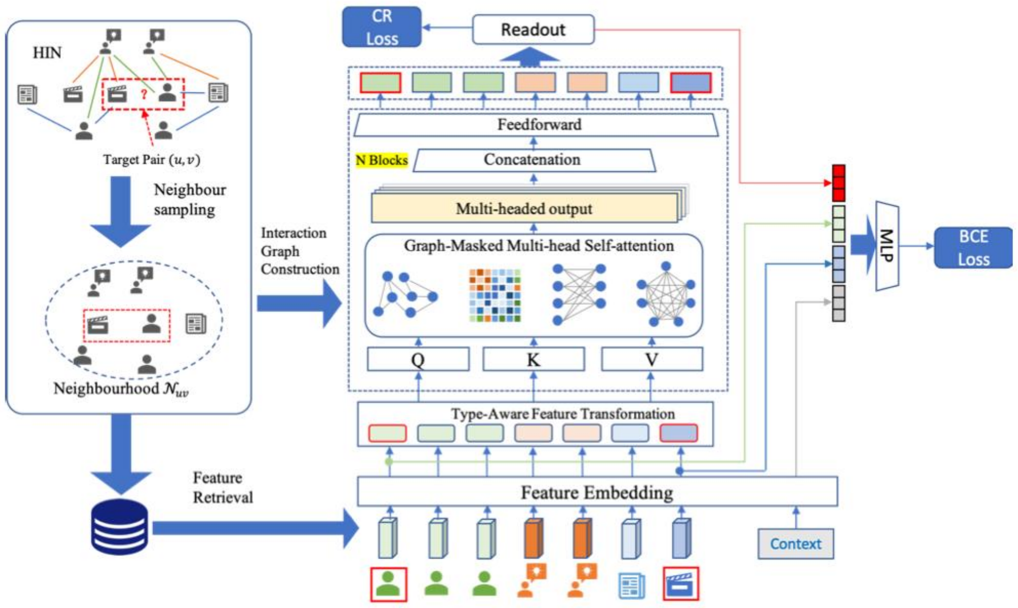
- [8] Graph Transformer (ACL-2020)



ENFORCE THE HEADS TO ATTEND TO DIFFERENT SUBSPACES

- [11] Graph-masked Transformer (2022)
  - designs 4 types of interaction graphs
  - enfore heads to attend to different subspaces



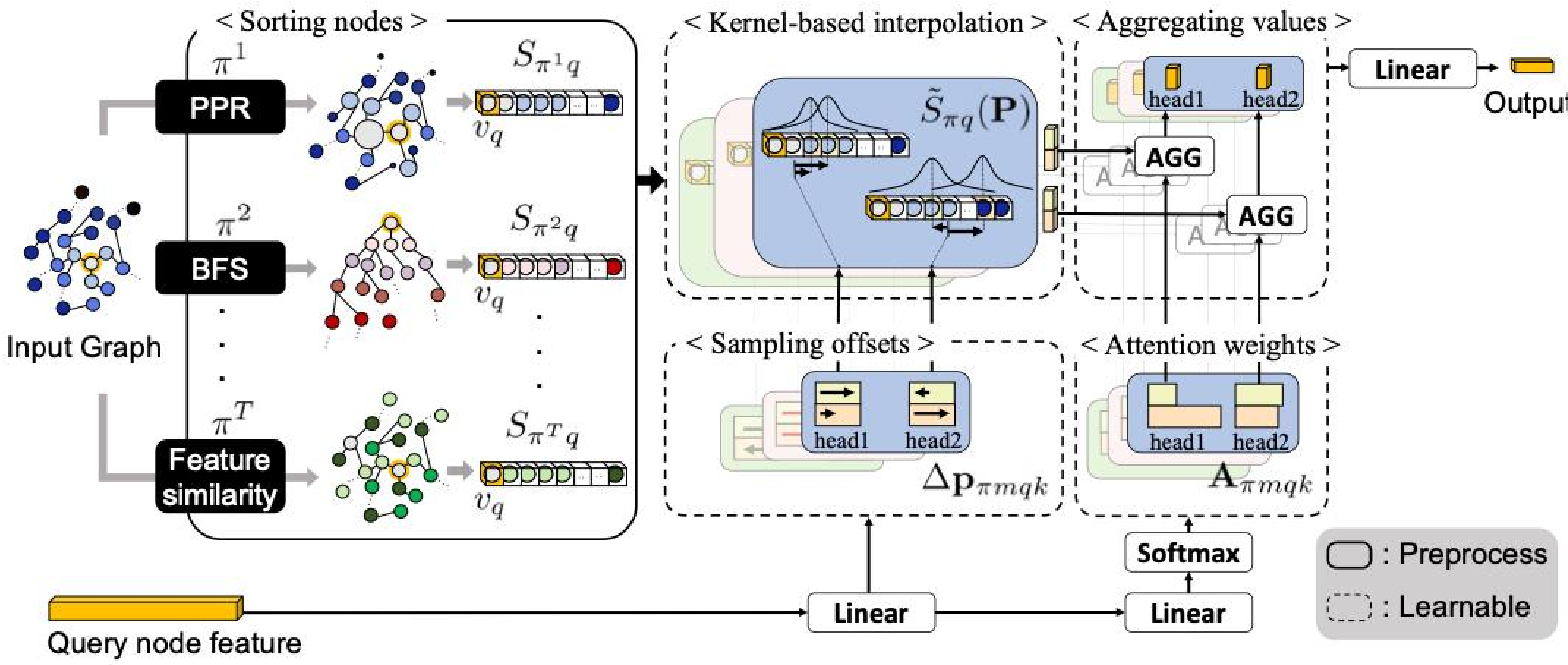


ENFORCE THE HEADS TO ATTEND TO DIFFERENT SUBSPACES

- [15] DGT (2022)
  - NodeSort module converts a graph into several sorted sequence. Several strategies can be used to generate sequences (e.g. personalized PageRank, BFS, and feature similarity)
  - DGA, a sparse attention, dynamically samples key/value pairs from the set of sorted sequences of node features

ENFORCE THE HEADS TO ATTEND TO DIFFERENT SUBSPACES

- [15] DGT (2022)



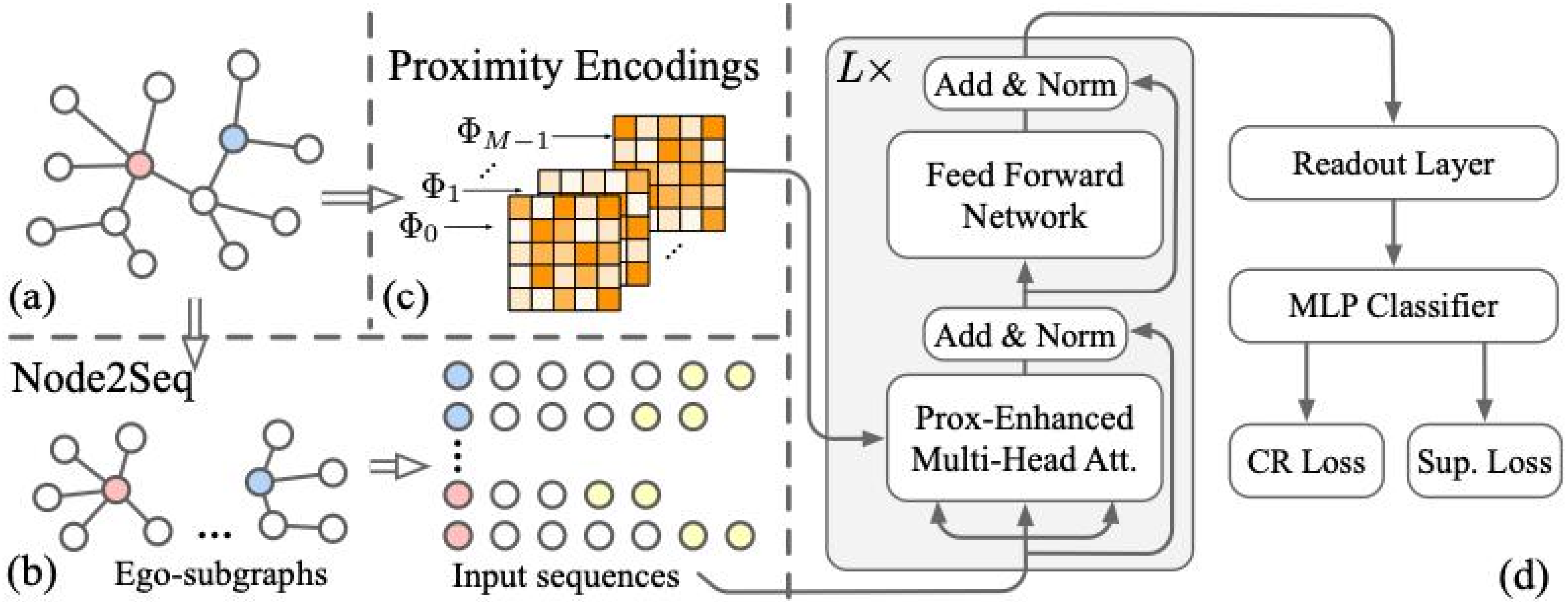
ADD SOFT GRAPH BIAS

- [8] Graph Transformer (ACL-2020)
    - proposes a Spatial Encoding Mechanism
    - measures the spatial realtion between 2 nodes
    - assigns each feasible value of the distance a learnable scale parameter as a graph bias term B^s

$$\mathbf{A} = (\frac{1}{\sqrt{d}}\mathbf{XQ}(\mathbf{XK})^\top) + \mathbf{B}^s$$

ADD SOFT GRAPH BIAS

- [12] Gohpormer (2021)
    - Samples ego-graphs to supplement local information
    - For each node pair, a structural encoding function is used to derived whether global node exists in this node pair



THANK YOU