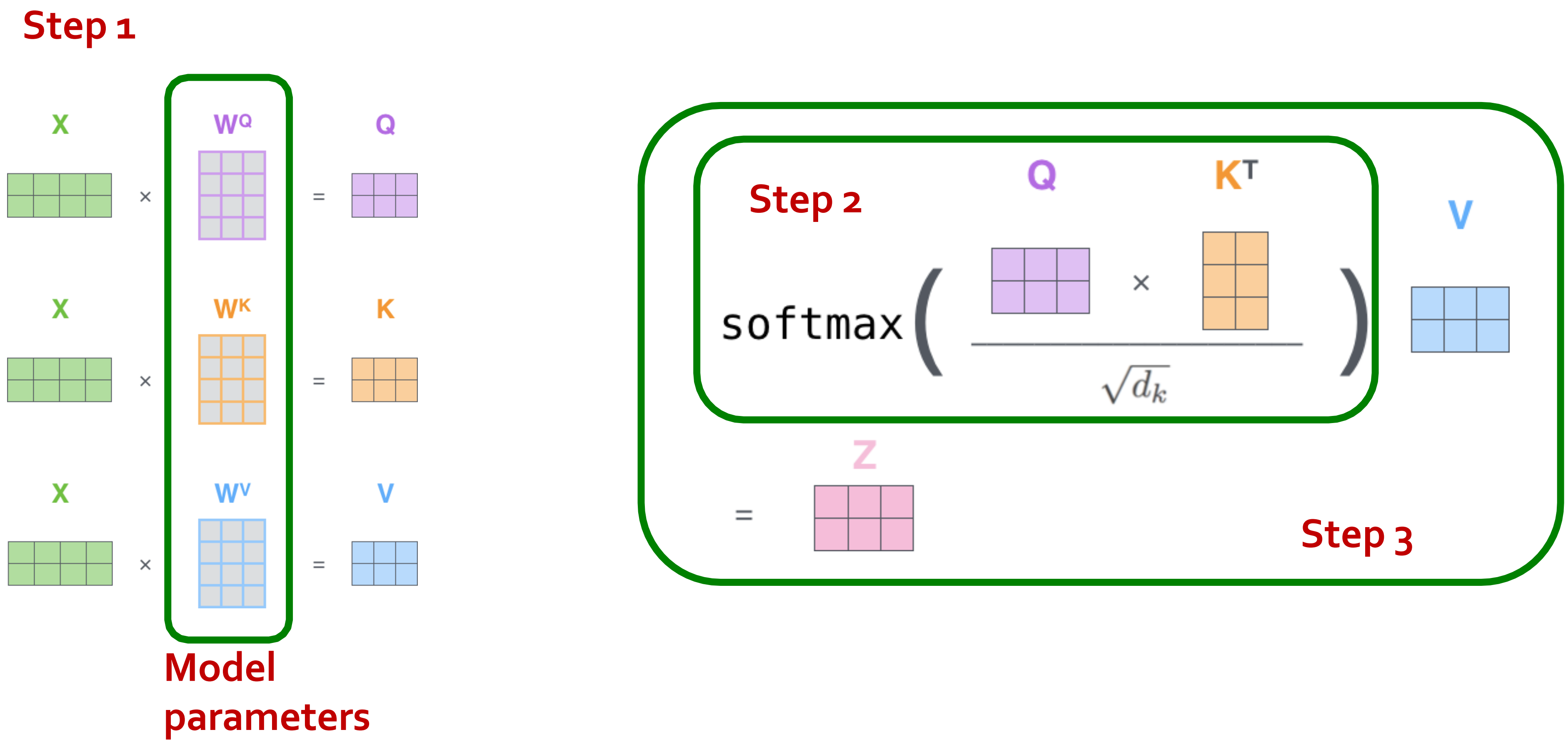
Self-attention

What is self attention?

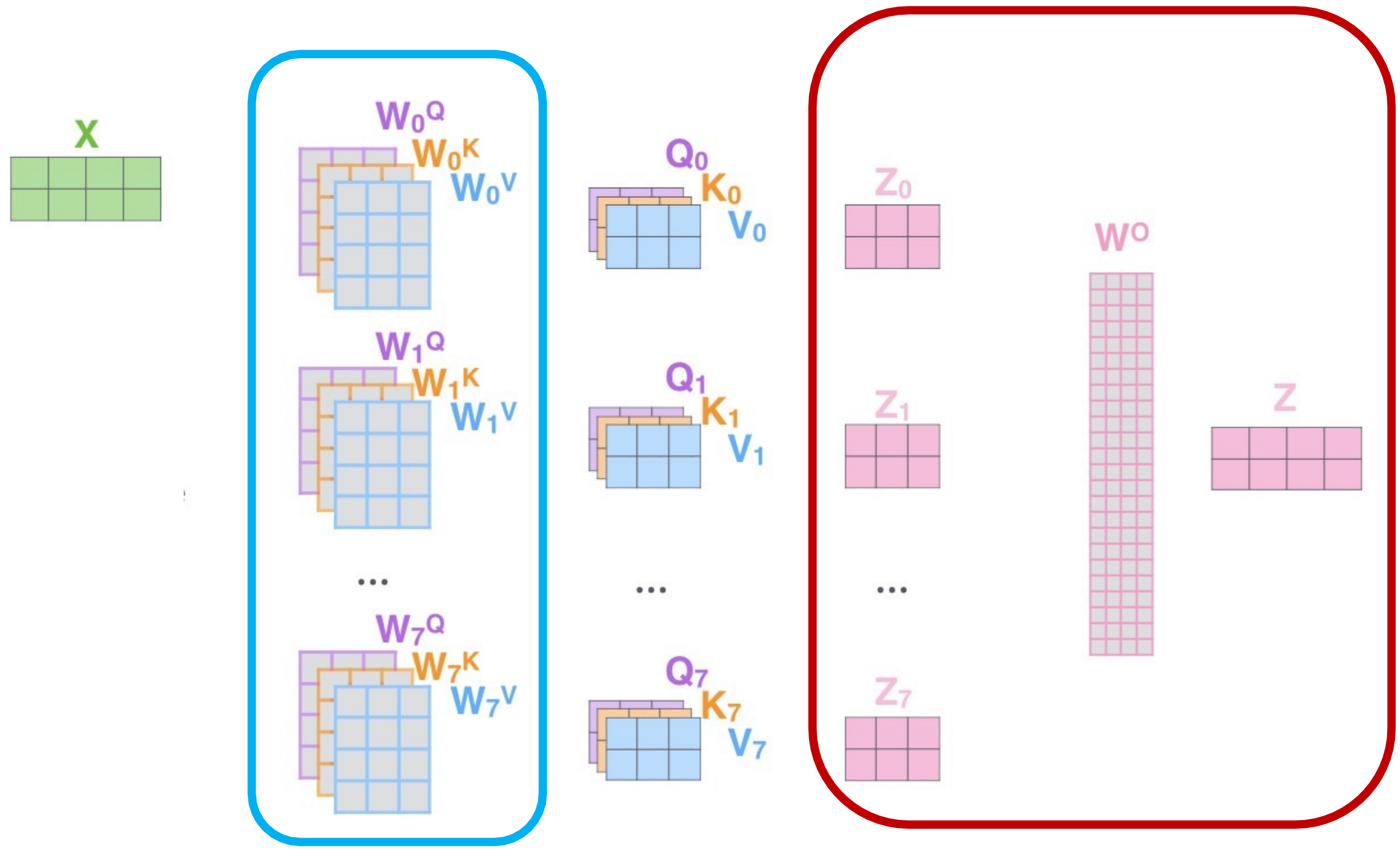


Multi-head self-attention

Do many self-attentions in parallel, and combine

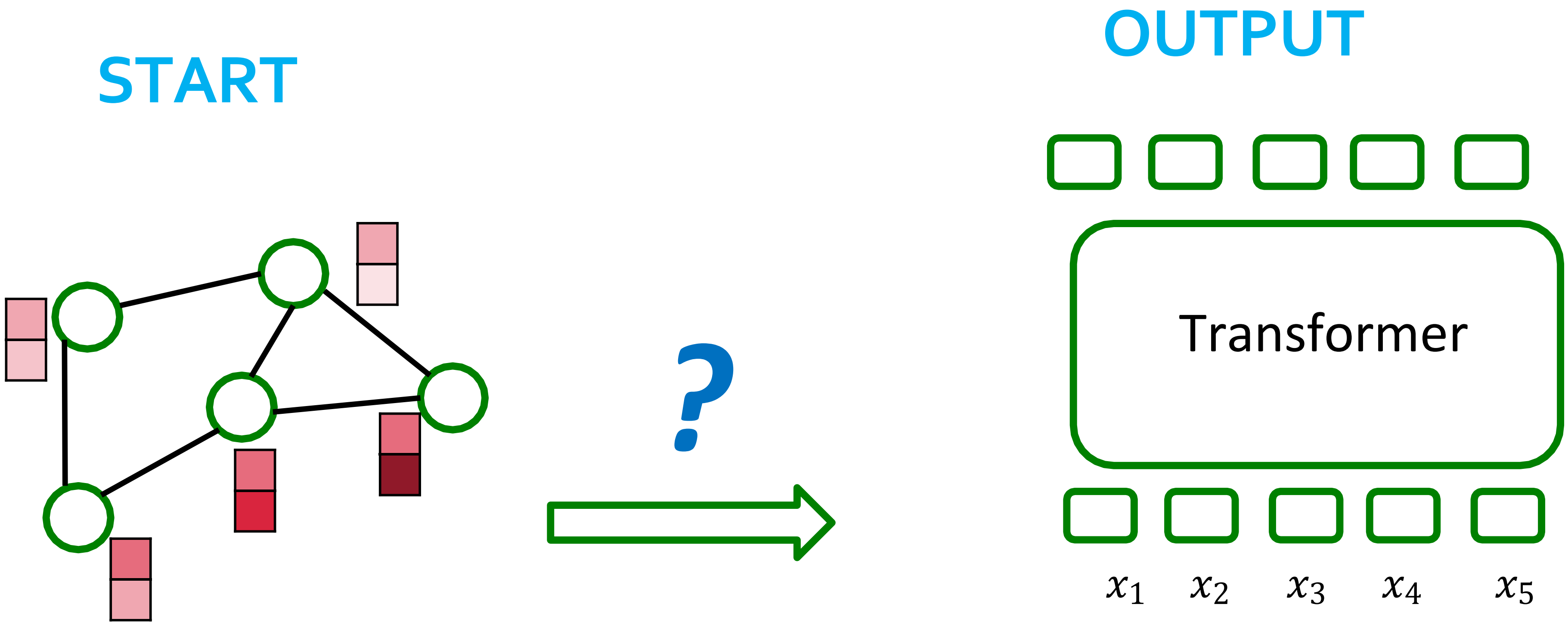
Different heads can learn different "similarities" between inputs

Each has own set of parameters



Processing Graphs with Transformers

We start with graph(s)
How to input a graph into a Transformer?

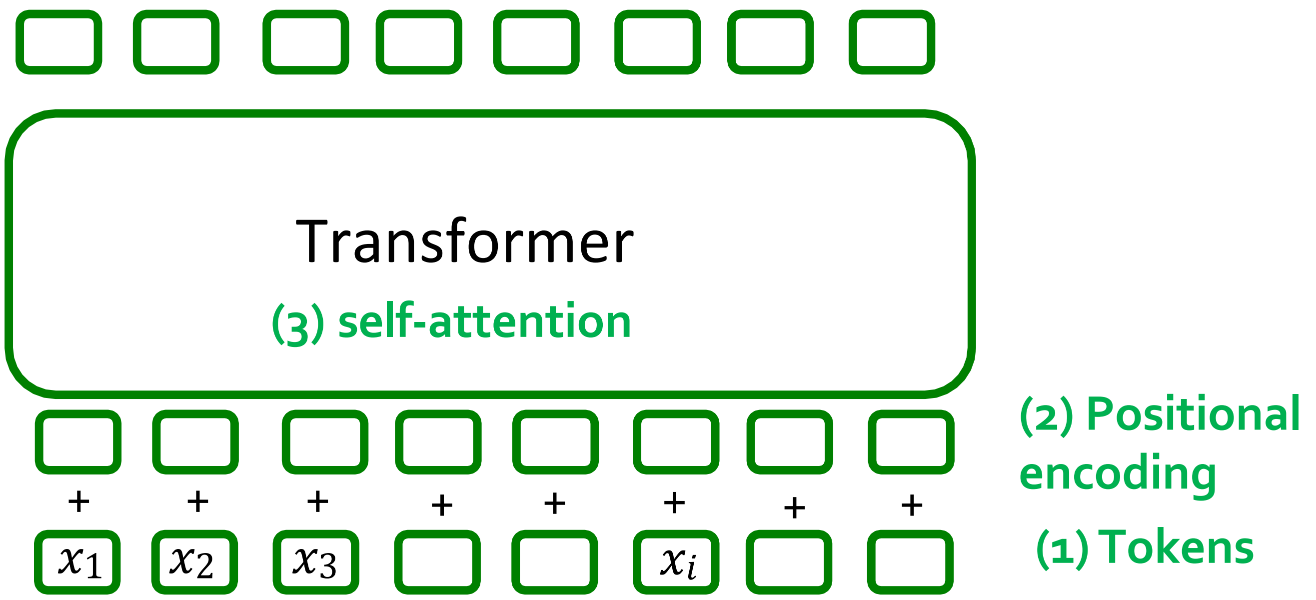


Components of a Transformer

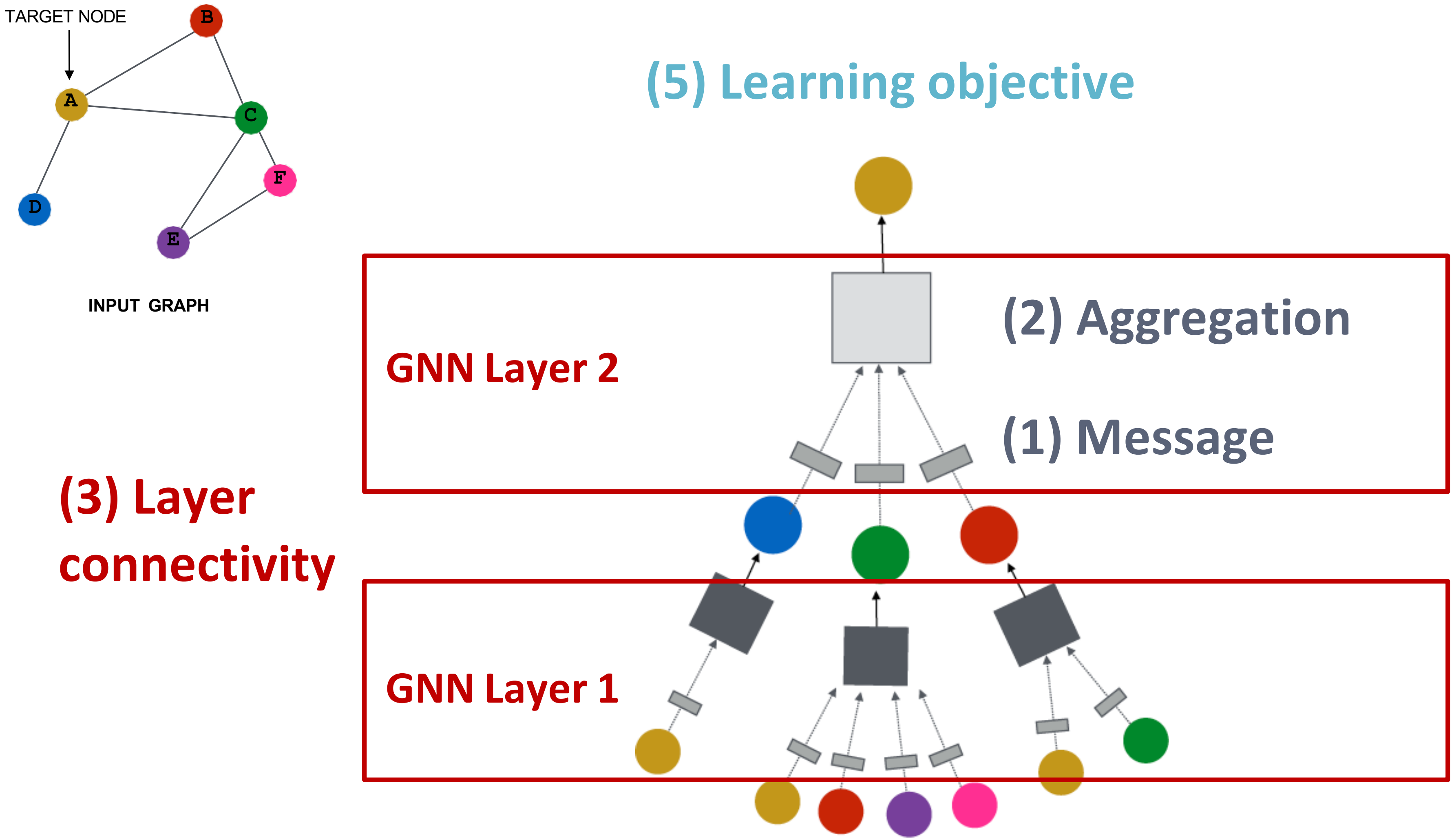
Key components of Transformer

- (1) tokenizing
- (2) positional encoding
- (3) self-attention

Key question: What should these be for a graph input?



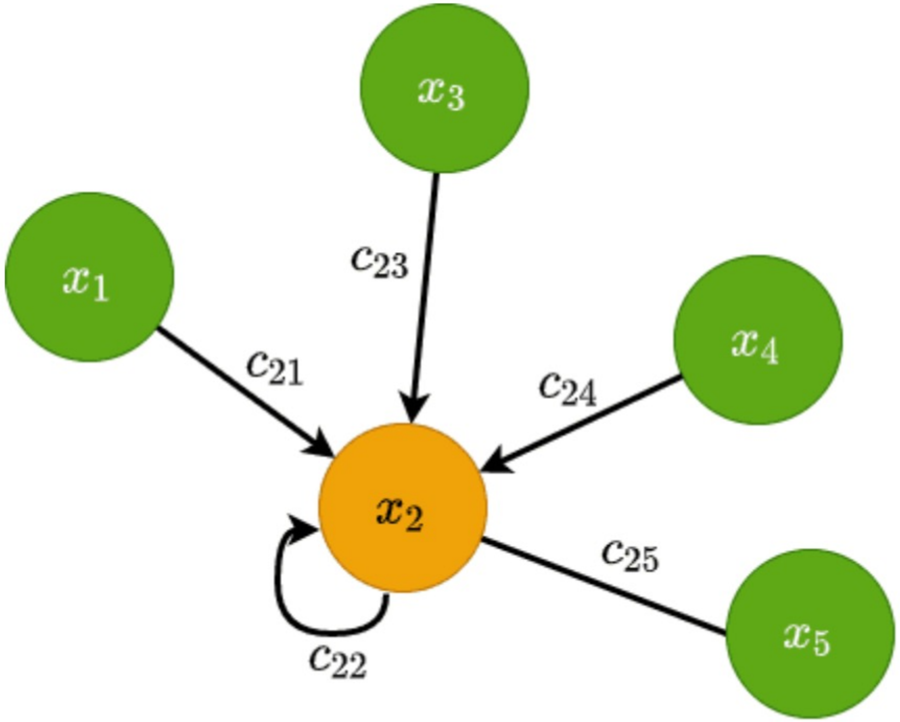
Recap: A General GNN Framework



Message Passing

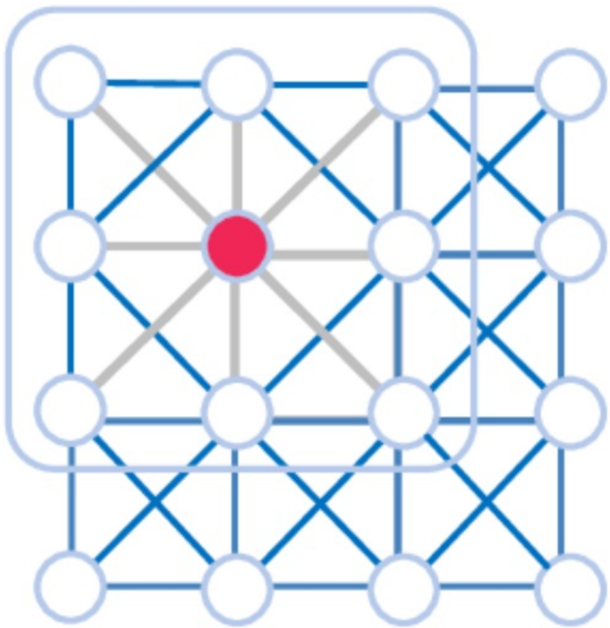
General rule for convolutional message passing:

$$h_i^l = W_s h_i^{l-1} + \sum_{v_j \in \mathcal{N}(v_i)} W_t h_j^{l-1},$$

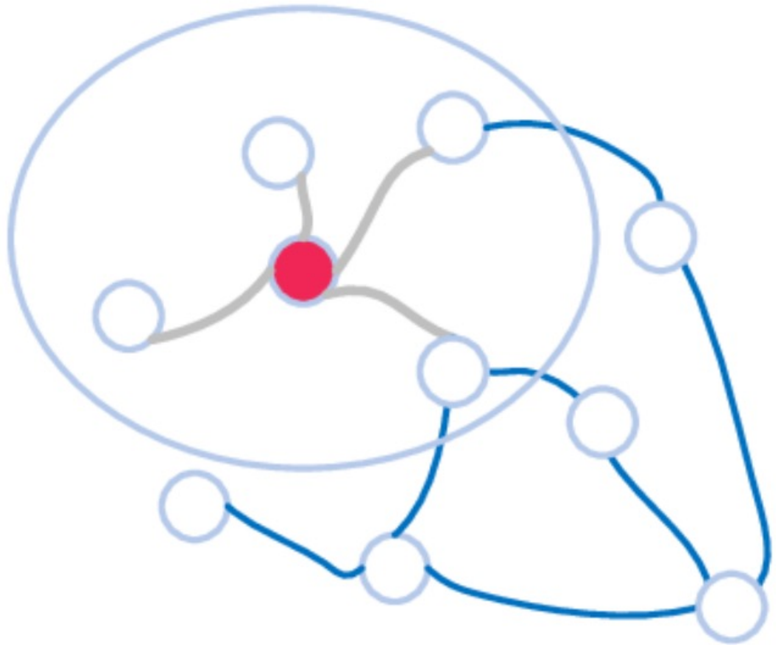


Convolution limitations

No ordering -> The weight is always the same!



(a)Convolution neural network



(b)Graph convolution network

Message Passing

General rule for attentive message passing:

$$h_i^l = W_s h_i^{l-1} + \sum_{v_j \in \mathcal{N}(v_i)} \alpha_{i,j}^{l-1} W_t h_j^{l-1}$$

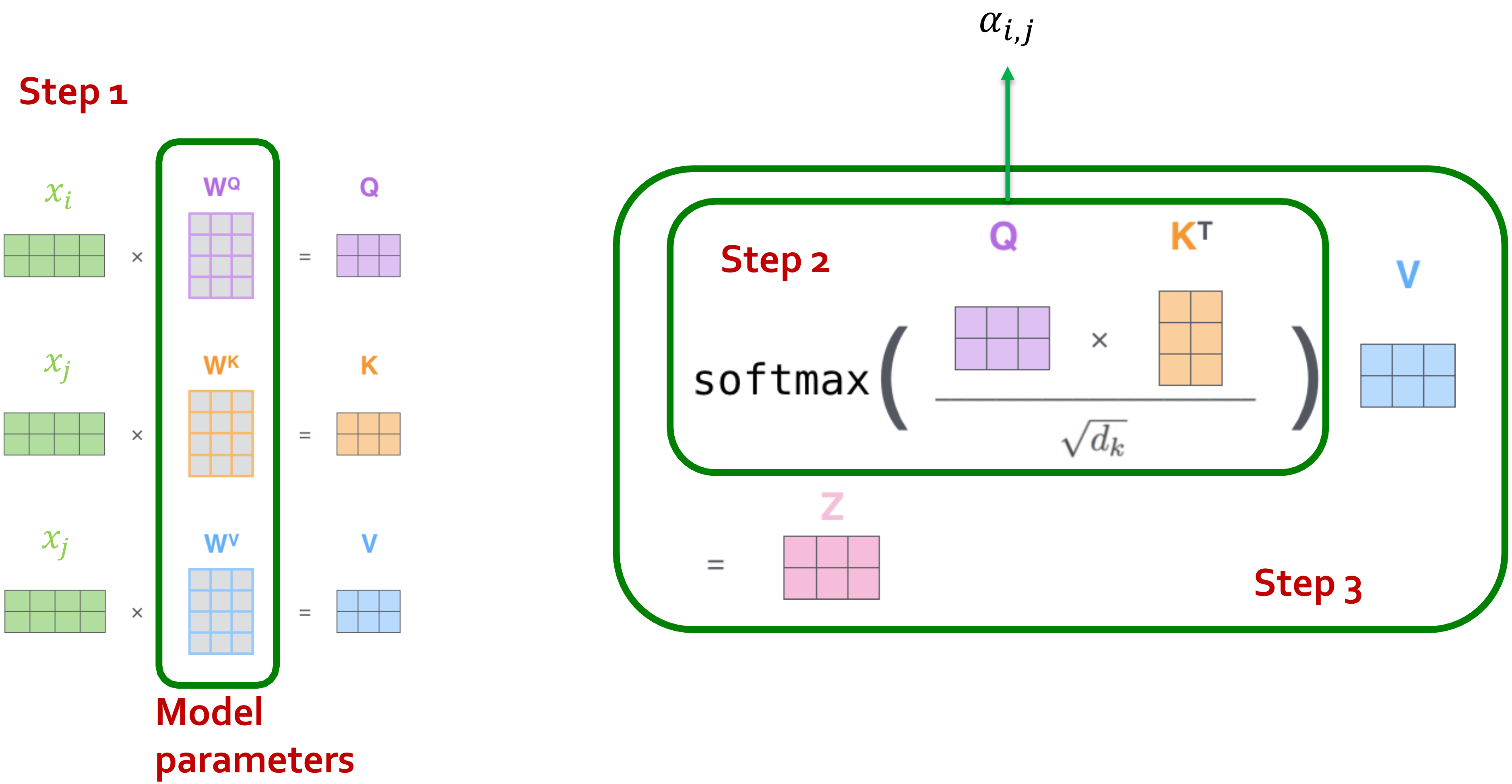
Where the attentive coefficients are calculated as:

$$\alpha_{i,j} = \mathrm{softmax}\left(\frac{(W_1 x_i)^T (W_2 x_j)}{\sqrt{d}}\right)$$

$$\alpha_{i,j} = \mathrm{softmax}\left(\frac{(W_1 x_i)^T (W_2 x_j + W_3 e_{i,j})}{\sqrt{d}}\right)$$

Self-attention

What is graph self attention?



Conclusions

Transformer for graph data:

1. Tokenizing corresponds to choose a neighbour
2. Positional embedding depends on the semantic of the graph
3. Self-attention is straightforward

Inputs	Model	Datasets		
		ogbn-products Test ACC	**ogbn-proteins Test ROC-AUC**	**ogbn-arxiv Test ACC**
$\mathbf{X}$	Multilayer Perceptron	0.6106 ± 0.0008	0.7204 ± 0.0048	0.5765 ± 0.0012
$\mathbf{X}, \mathbf{A}$	GCN	0.7851 ± 0.0011	0.8265 ± 0.0008	0.7218 ± 0.0014
	GAT	0.8002 ± 0.0063	0.8376 ± 0.0007	0.7246 ± 0.0013
	Graph Transformer	0.8137 ± 0.0047	0.8347 ± 0.0014	0.7292 ± 0.0010
$\mathbf{A}, \hat{\mathbf{Y}}$	GCN	0.7832 ± 0.0013	0.8083 ± 0.0021	0.7018 ± 0.0009
	GAT	0.7751 ± 0.0054	0.8247 ± 0.0033	0.7055 ± 0.0012
	Graph Transformer	0.7987 ± 0.0104	0.8160 ± 0.0007	0.7090 ± 0.0007
$\mathbf{X}, \mathbf{A}, \hat{\mathbf{Y}}$	GCN	0.7987 ± 0.0104	0.8247 ± 0.0032	0.7264 ± 0.0003
	GAT	0.8193 ± 0.0017	0.8556 ± 0.0009	0.7278 ± 0.0009
	Graph Transformer	$\mathbf{0.8256 \pm 0.0031}$	0.8560 ± 0.0003	$\mathbf{0.7311 \pm 0.0021}$
	∟ w/ Edge Feature	*	$\mathbf{0.8642 \pm 0.0008}$	*

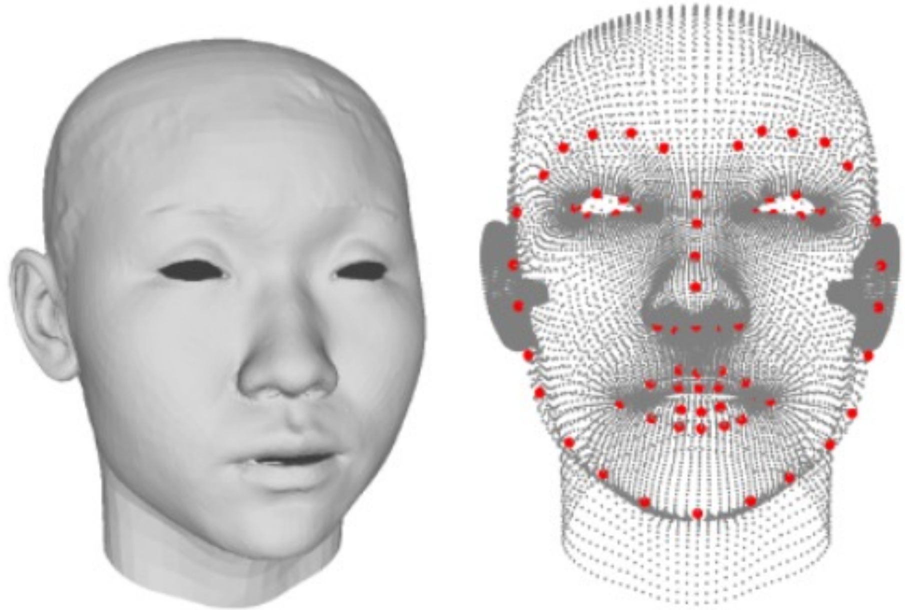
A light example

Two-layer transformer: not enough data to go deep

Adjacency matrix fully connected

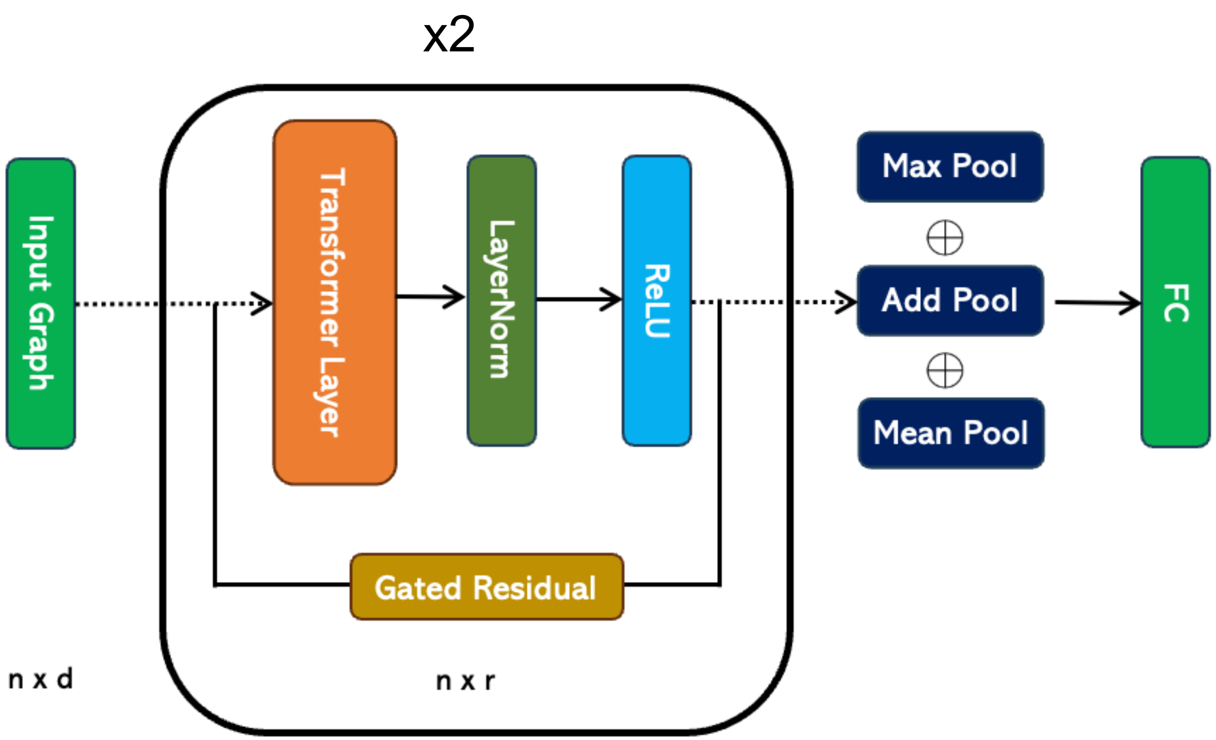
Nodes feature: PE, node pos., FPFH

Edge features: Spline
bewteen nodes



The model

Facescape dataset, few data -> a deep model could overfit



Results

Model	Accuracy
Transformer	89.07
Transformer without ReLU	81.79
Transformer without LayerNorm	86.34
Transformer without Residual	82.40
Transformer with one head	77.75
Transformer (3 layers)	51.07

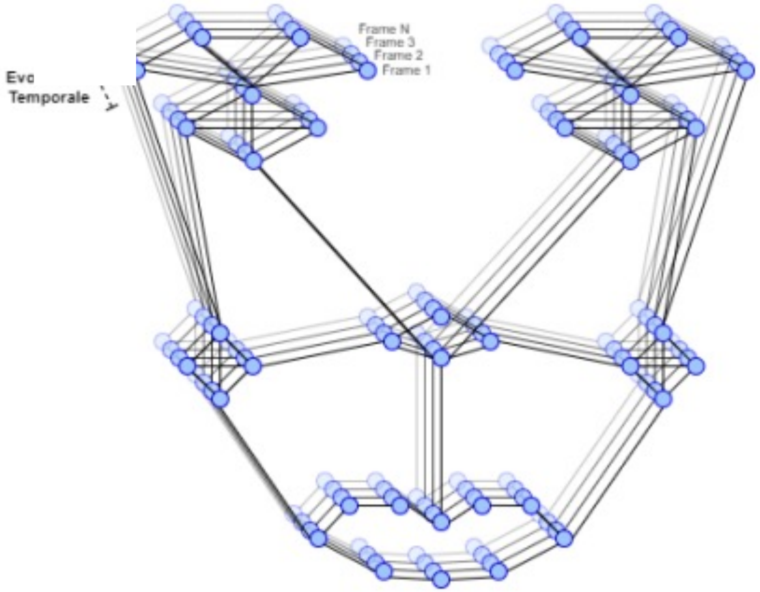
A more complex example

Spatiotemporal graph

Adjacency matrix adapted to temporal and spatial connections

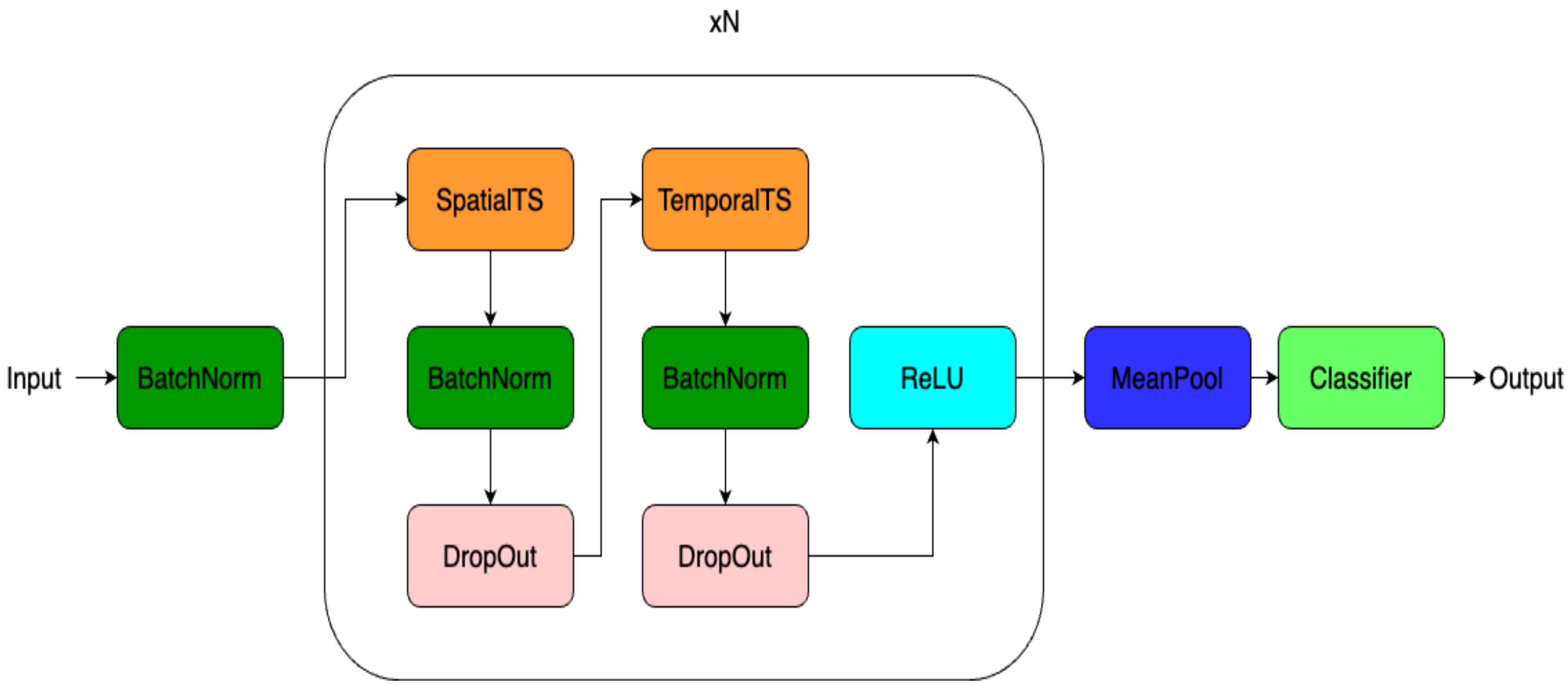
Ten-layers model

Nodes feature: node position, Gabor filters, MFCC



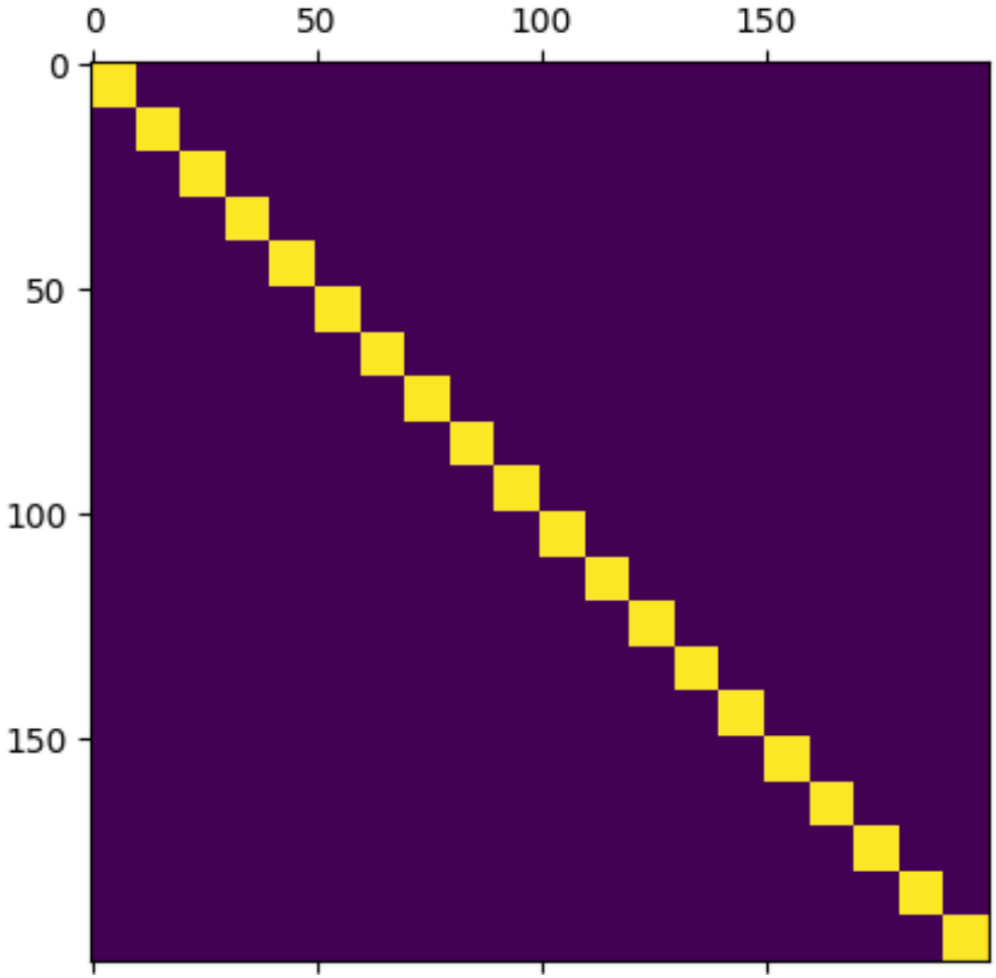
The model

Spatial and Temporal Transformer?



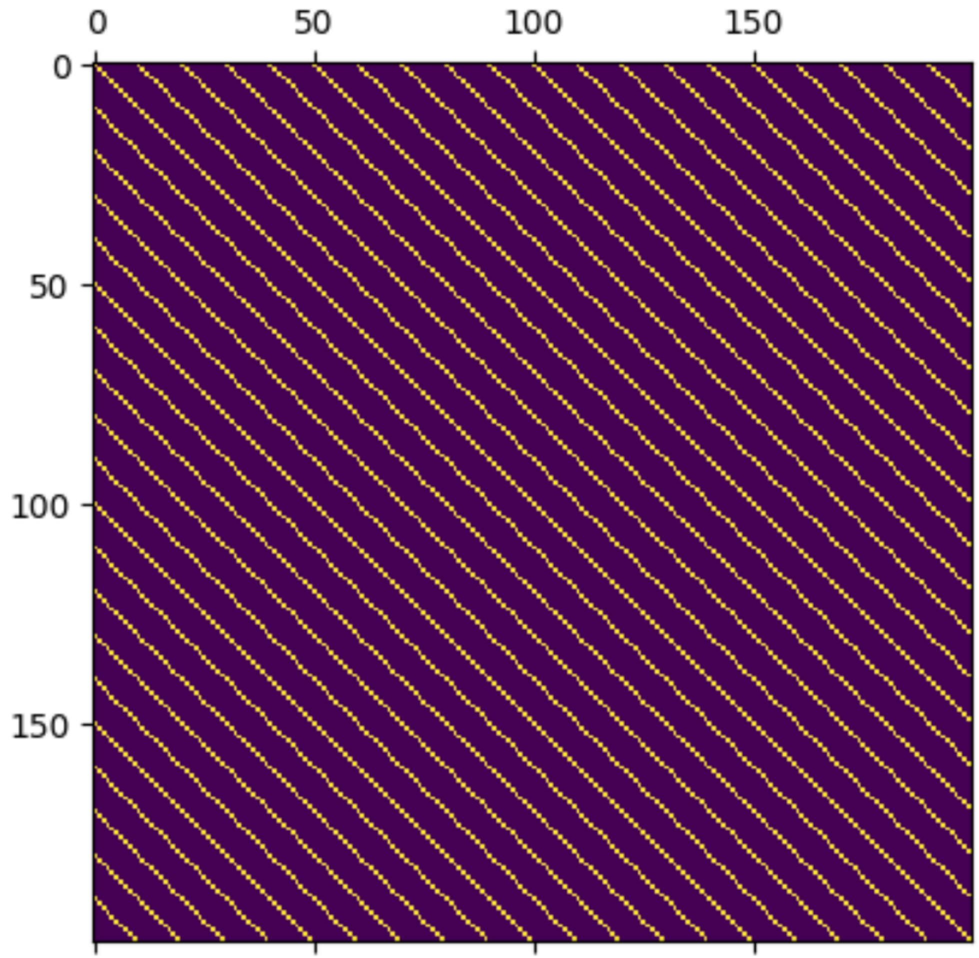
Adjacency matrices

Spatial adjacency matrix



Adjacency matrices

Temporal adjacency matrix



Further step: intra-attention

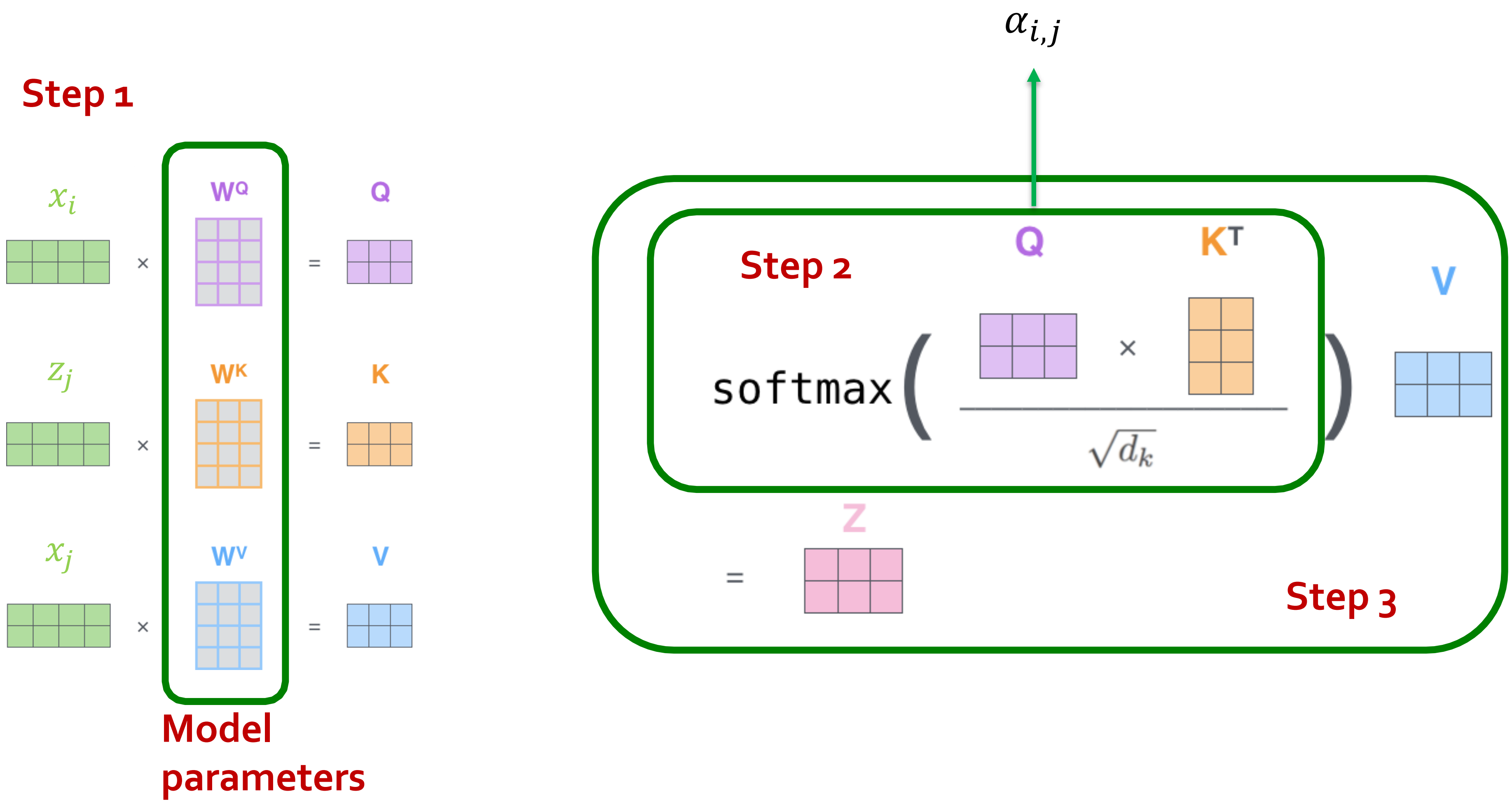
Since we have different modalities, we can attention one of them with respect to another

Example: Audio (z) -> key

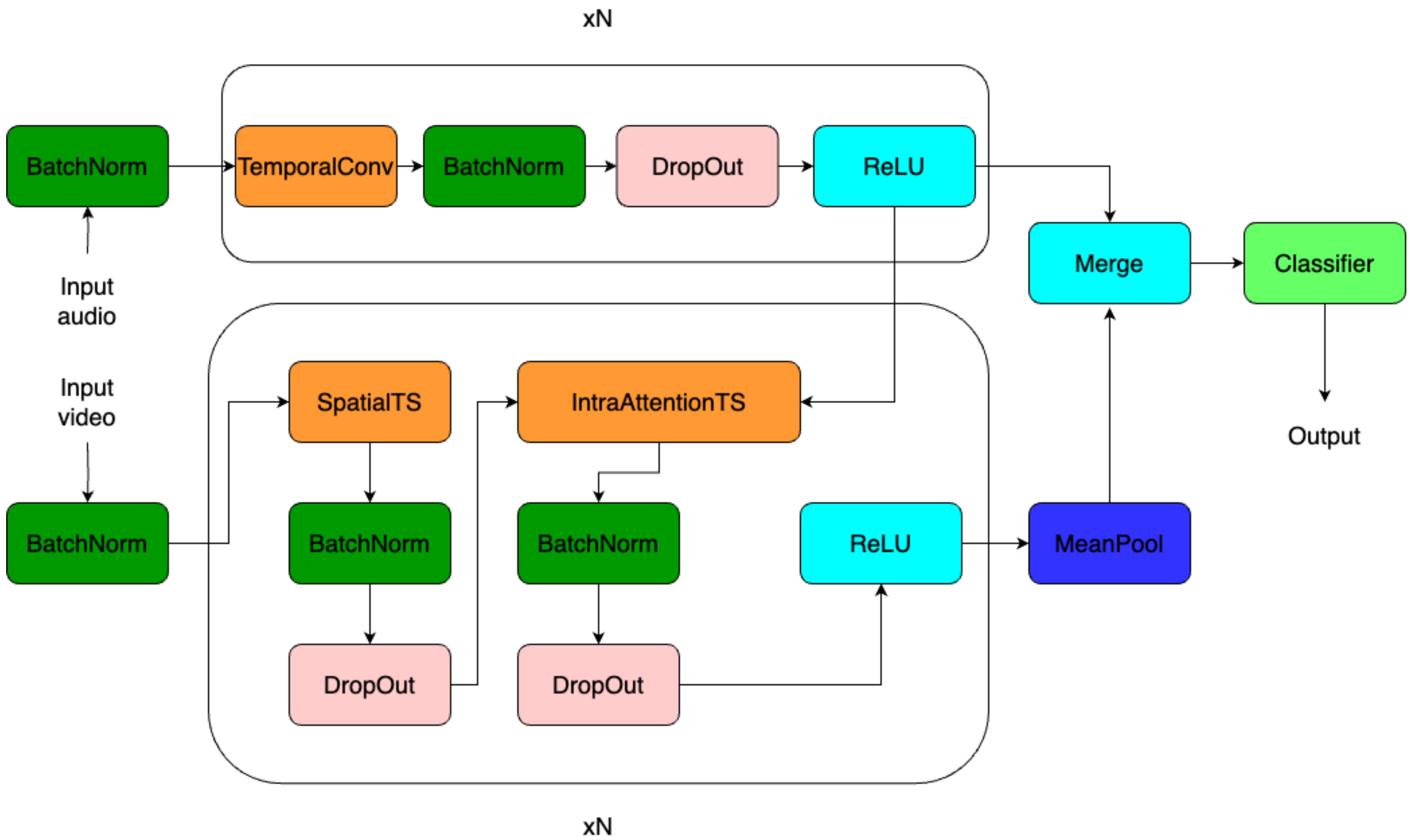
Video (x) -> query

Video (x) -> value

Intra-attention



The model


xN
BatchNorm
TemporalConv
BatchNorm
DropOut
ReLU
Merge
Classifier
Input
audio
Input
video
SpatialTS
IntraAttentionTS
Output
BatchNorm
BatchNorm
BatchNorm
ReLU
MeanPool
DropOut
DropOut
xN