How Attentive are Graph Attention Networks?

not that much…

Shaked Brody
Technion

Uri Alon
Language Technologies Institute
Carnegie Mellon University

Eran Yahav
Technion

https://arxiv.org › stat

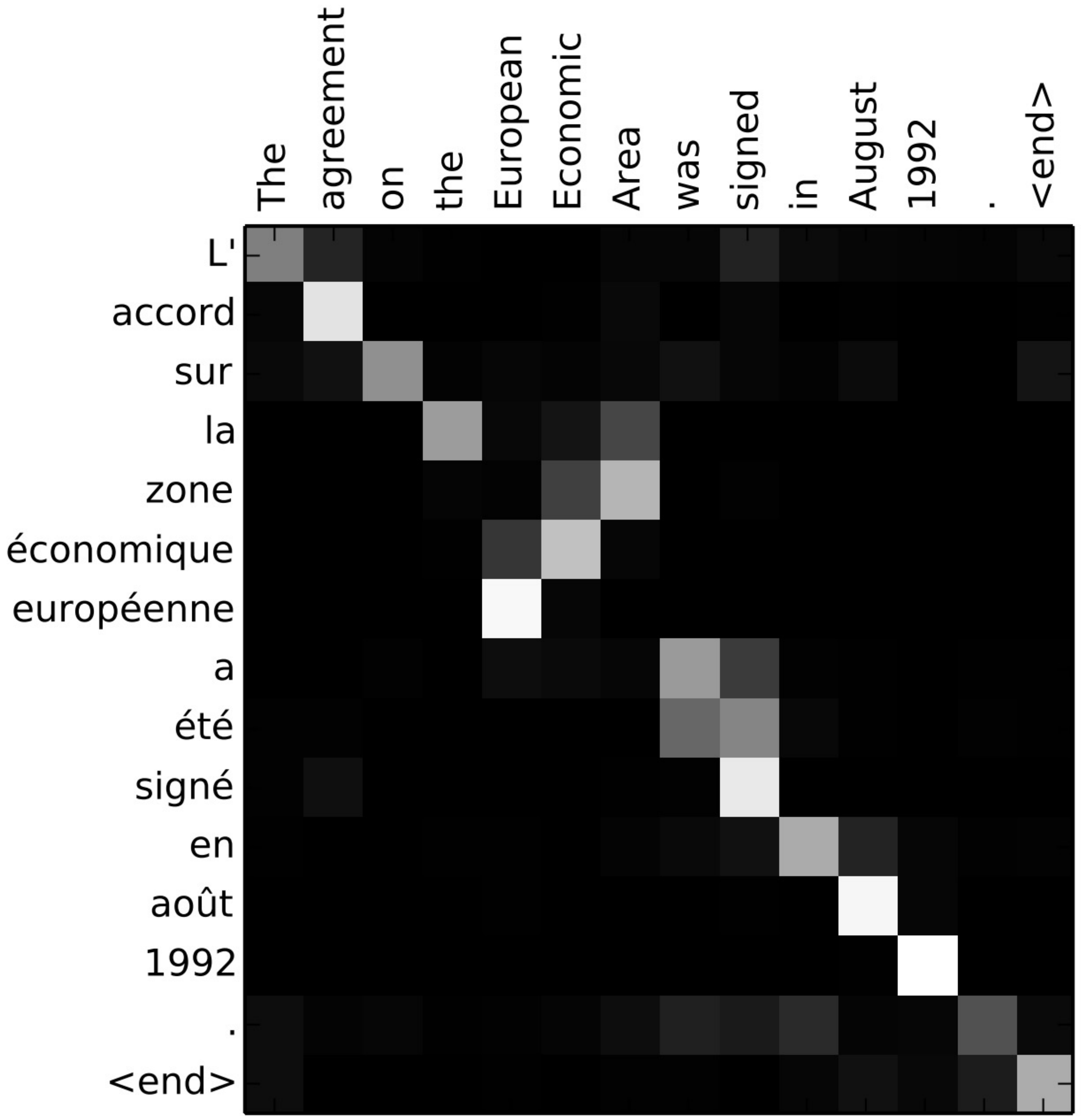
[1710.10903] Graph Attention Networks - arXiv

by P Veličković · 2017 · Cited by 5222 — Abstract: We present **graph attention networks** (GATs), novel neural network architectures that operate on graph-structured data, ...

we introduce an attention-based architecture to perform node classification of graph-structured data. The idea is to compute the hidden representations of each node in the graph, by attending over its neighbors, following a *self-attention* strategy. The attention architecture

But what kind of attention?

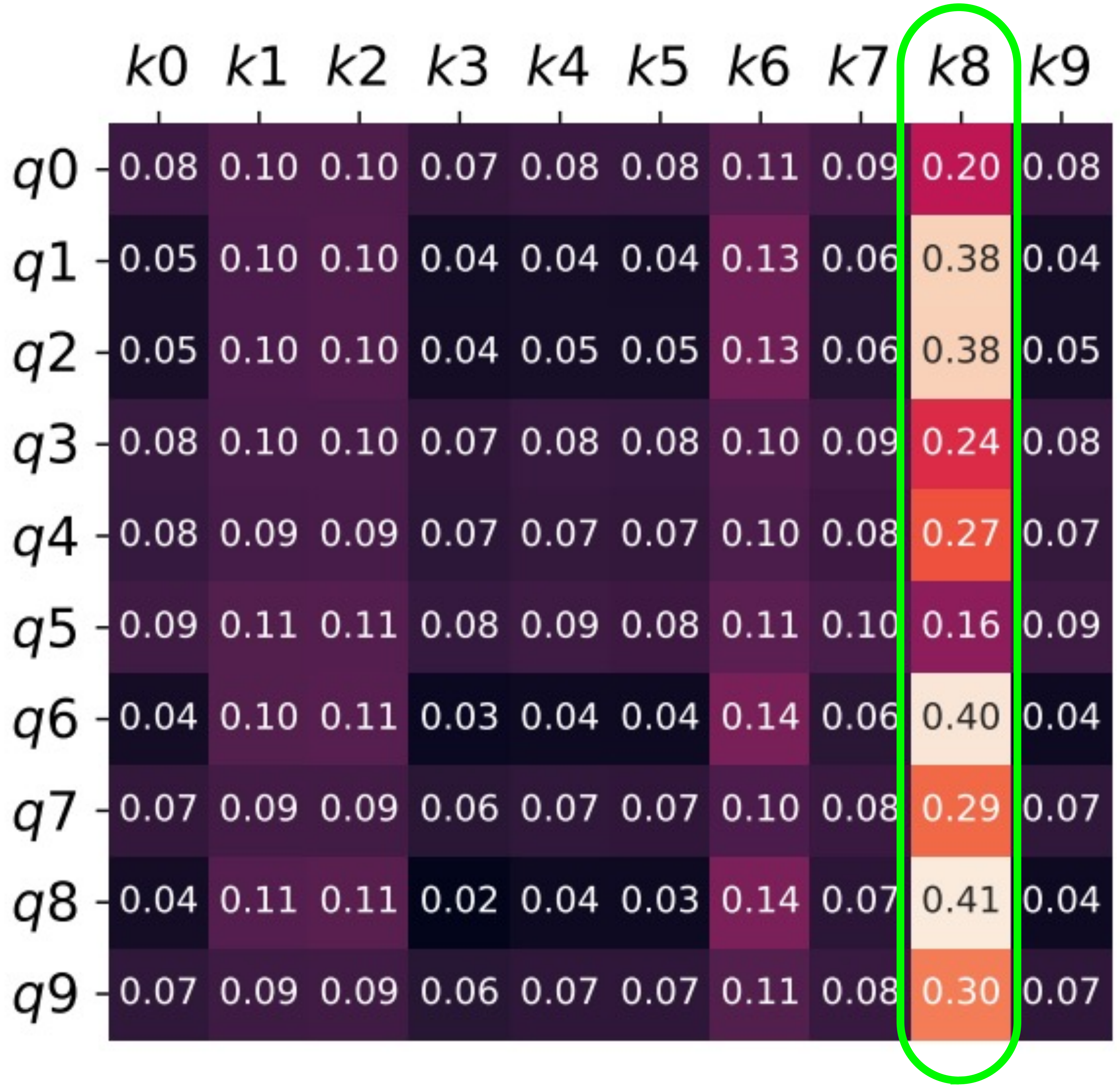
Attention



[Bahdanau et al., ICLR 2015]

The ability of different queries to learn to “focus” differently on a set of keys

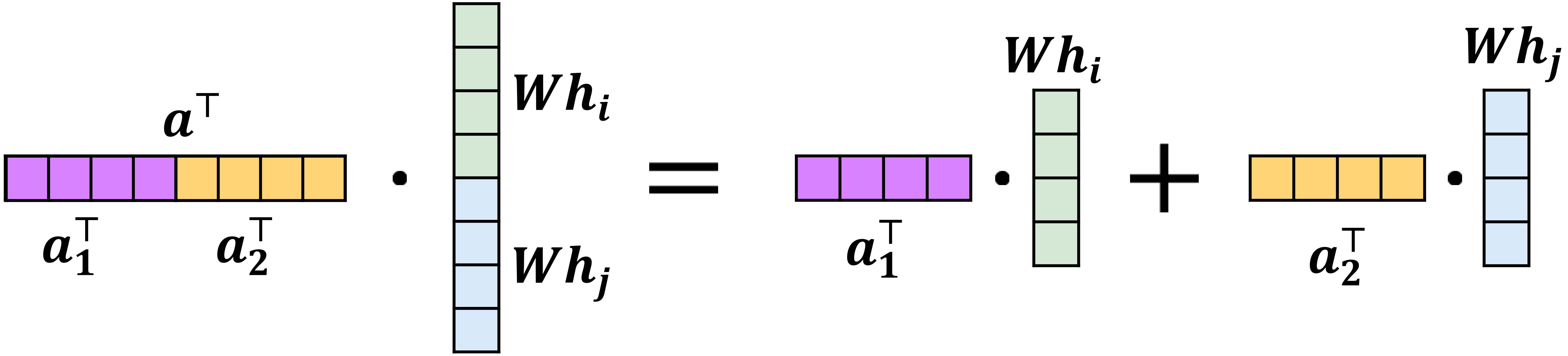
Graph Attention Networks (GAT) [Veličković et al., 2018]



Static Attention

GAT uses an Addition of Two Dot Products

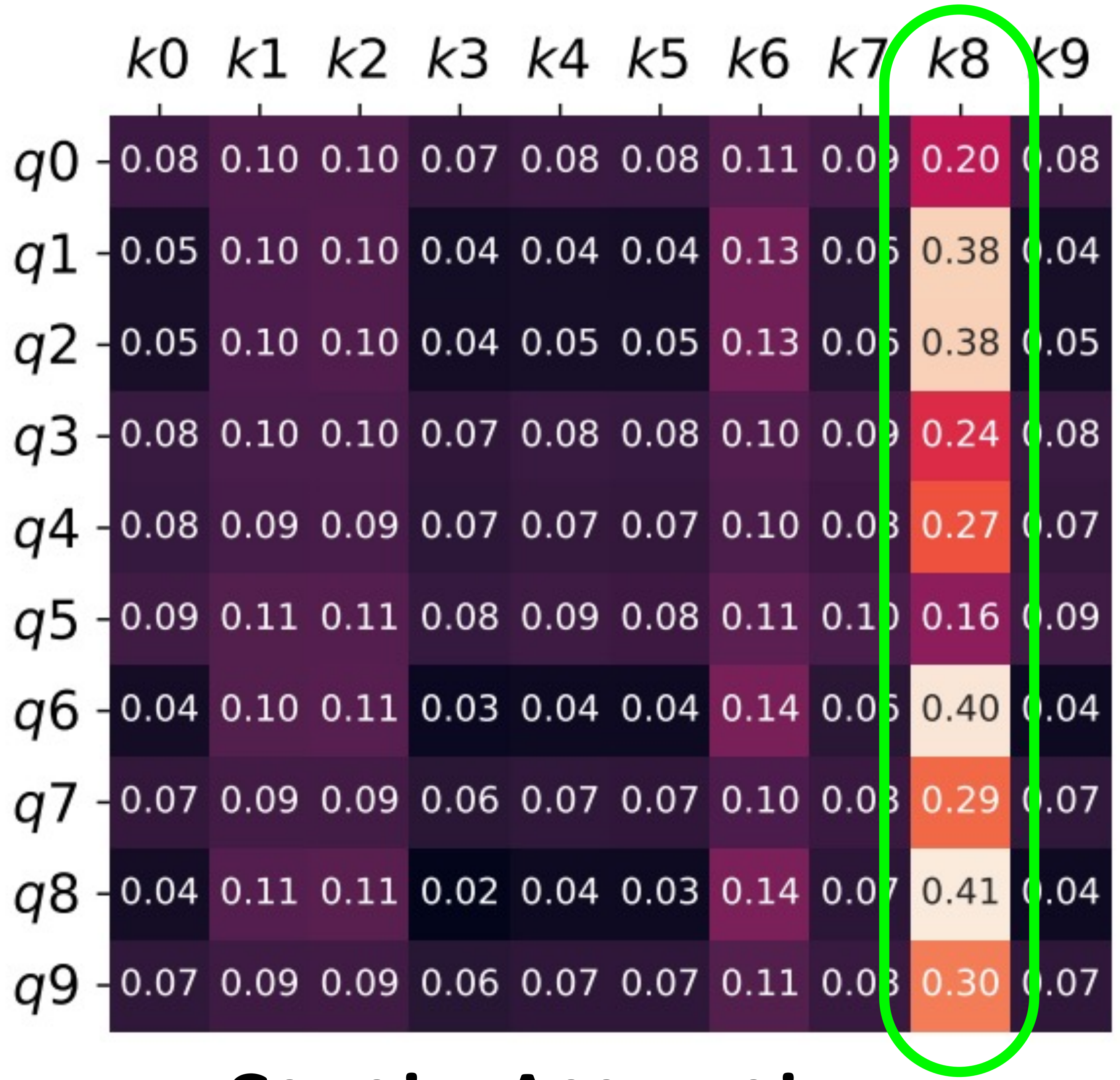
$$e(h_i, h_j) = \boxed{LeakyReLU}\left(\boxed{a}^\top [W\boxed{h_i} \,||\, W\boxed{h_j}]\right)$$



$$e(h_i, h_j) = \underbrace{\boxed{LeakyReLU}}_{\mathbb{R}^1 \to \mathbb{R}^1}\left(\underbrace{\boxed{a_1^\top \cdot Wh_i}}_{\in \mathbb{R}^1} + \underbrace{\boxed{a_2^\top \cdot Wh_j}}_{\in \mathbb{R}^1}\right)$$

GAT Attends to the Same Key Regardless of Query

$$e(h_i, h_j) = LeakyReLU\left(a_1^\top \cdot Wh_i + \underbrace{\boxed{a_2^\top \cdot Wh_j}}_{S_j}\right)$$



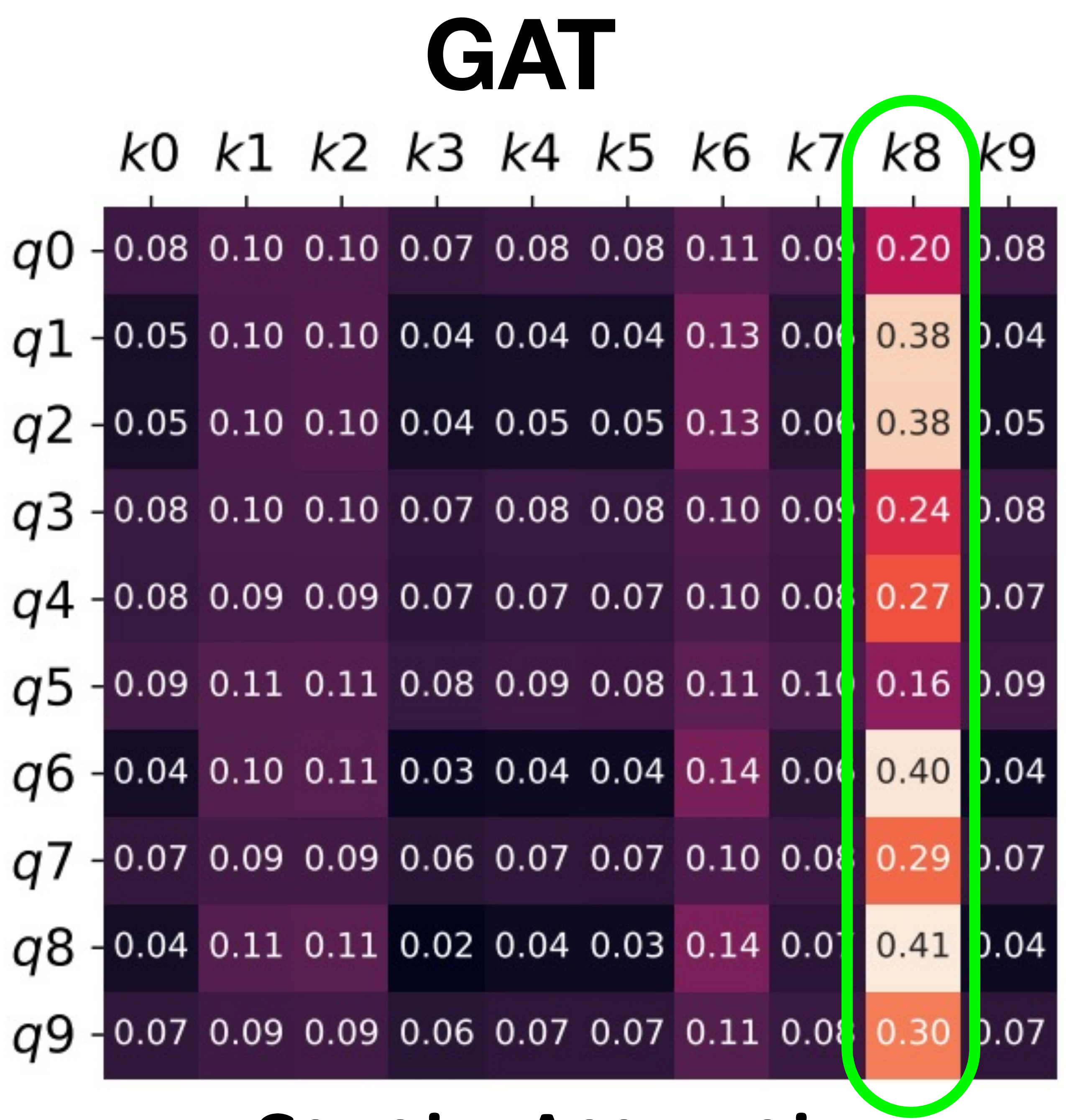
Static Attention

$$S_8 \geq S_6 \geq S_2 \geq \cdots$$

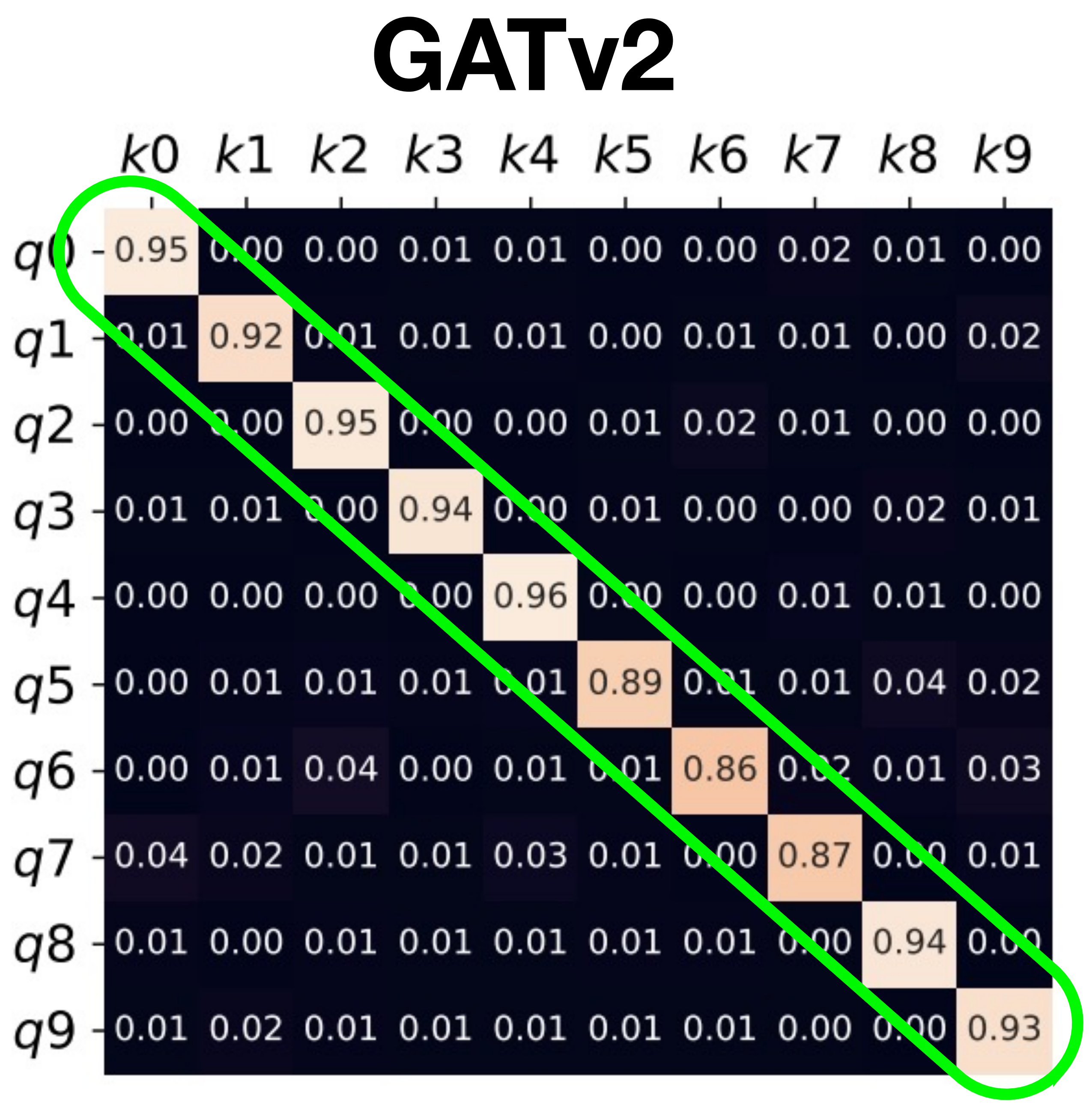
GATv2: Fixing Graph Attention Mechanism

GAT , Veličković et al., 2018: $e(h_i, h_j) = LeakyReLU\left(a^\top [W h_i \,||\, W h_j]\right)$

GATv2, this work: $e(h_i, h_j) = a^\top LeakyReLU\left(W \cdot [h_i \,||\, h_j]\right)$

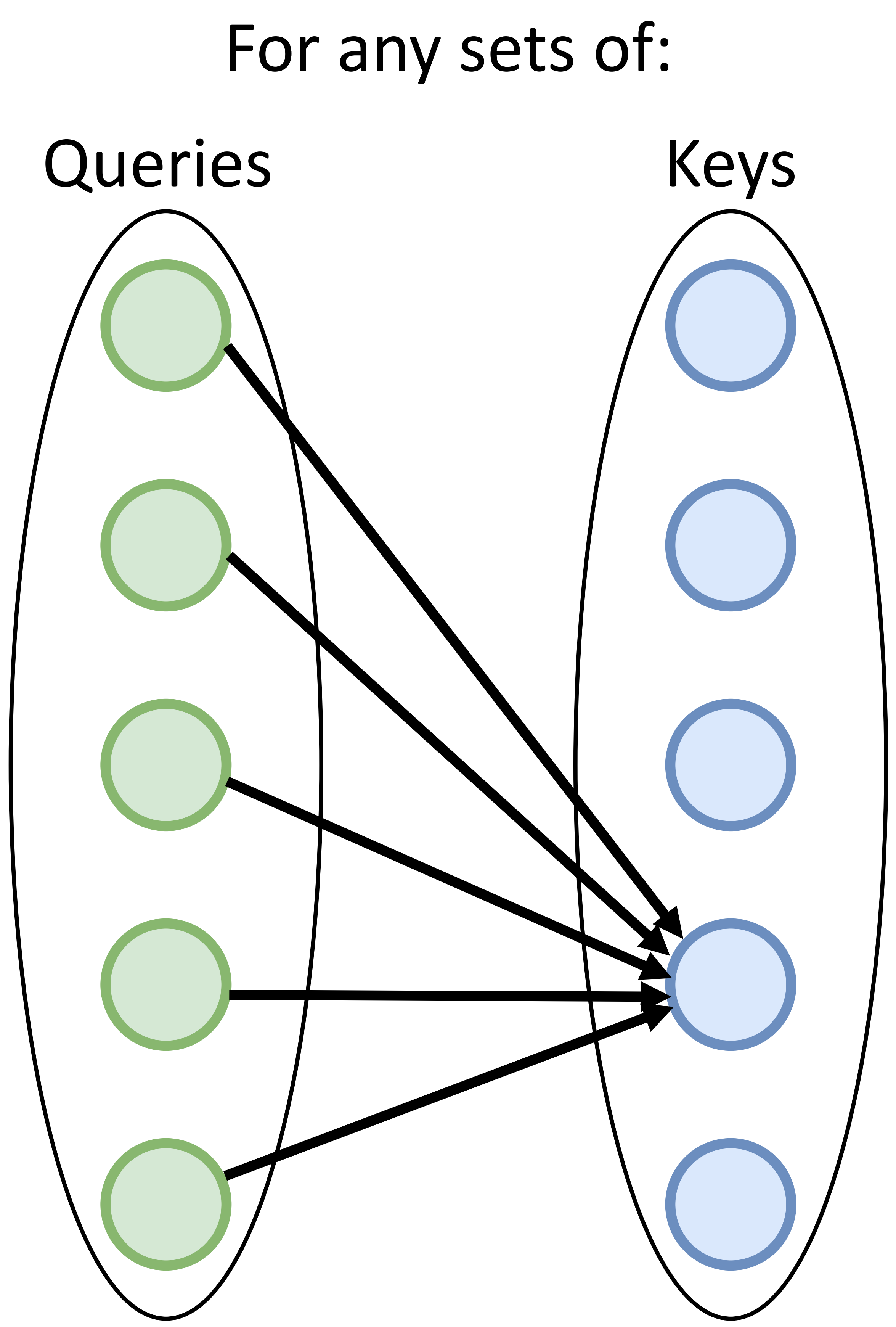


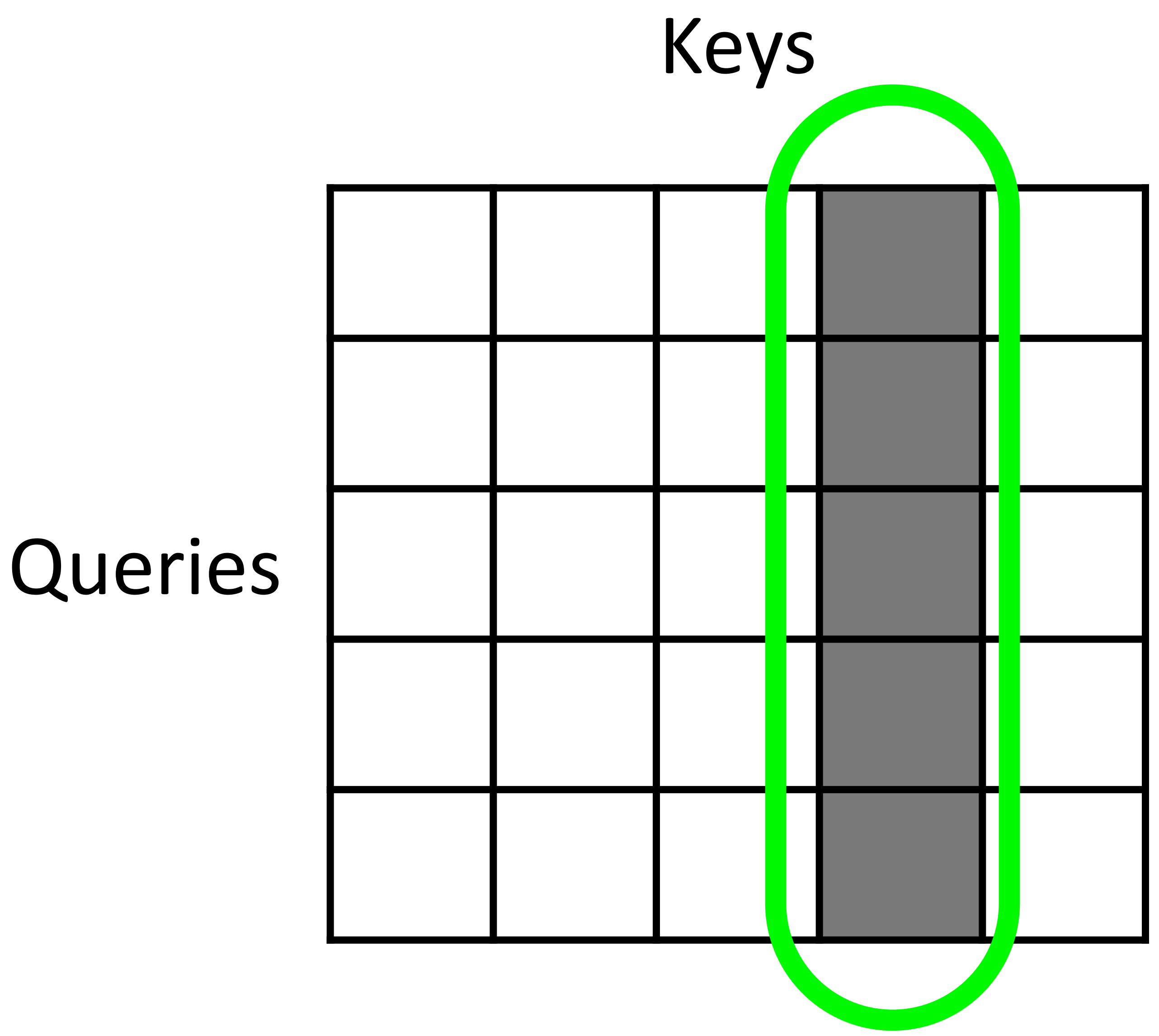
Static Attention



Dynamic Attention

Static Attention

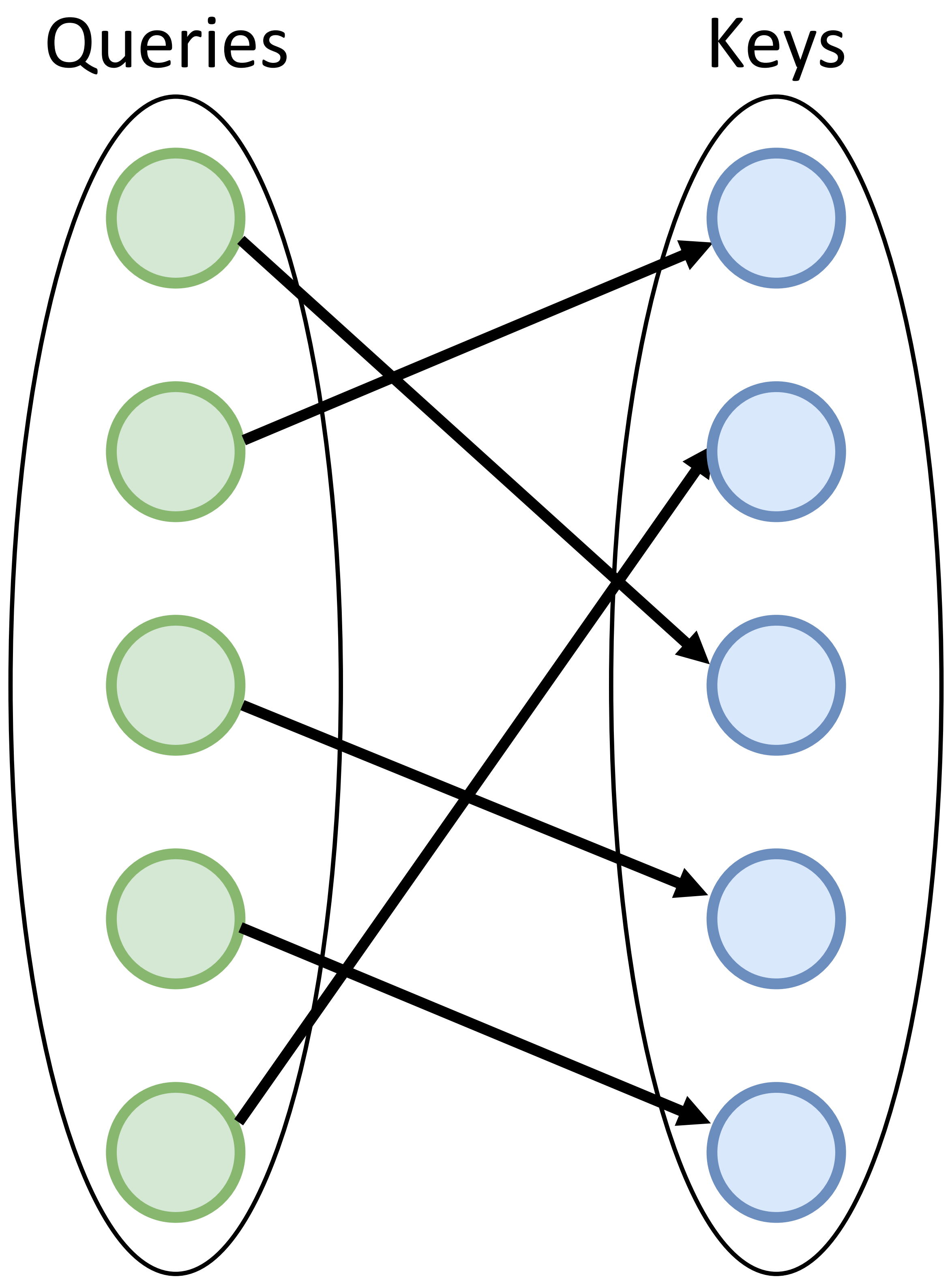


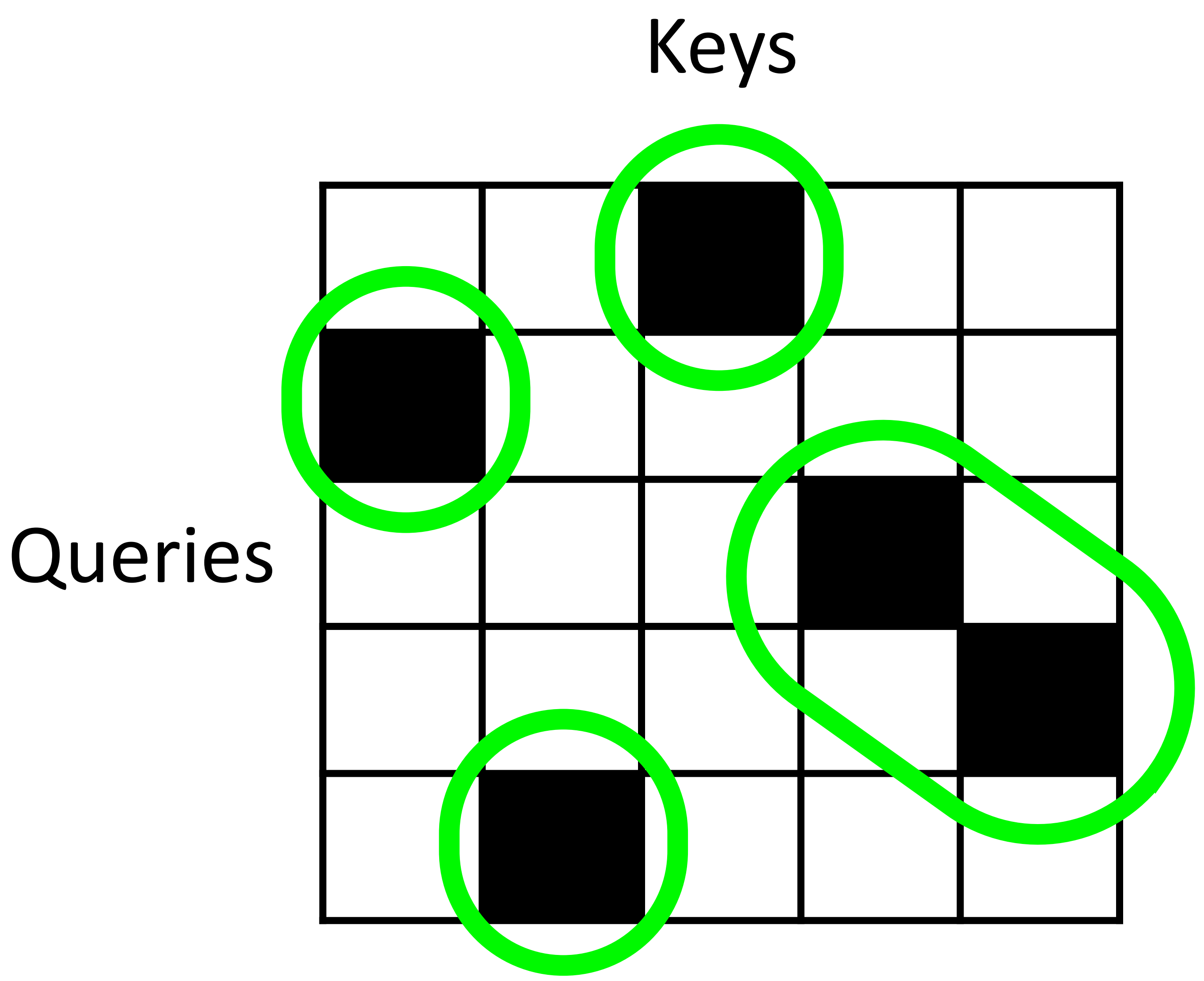


There is always a key that gets the most attention, regardless of the query

Dynamic Attention

For any set of queries, keys,
and any desired mapping between them:

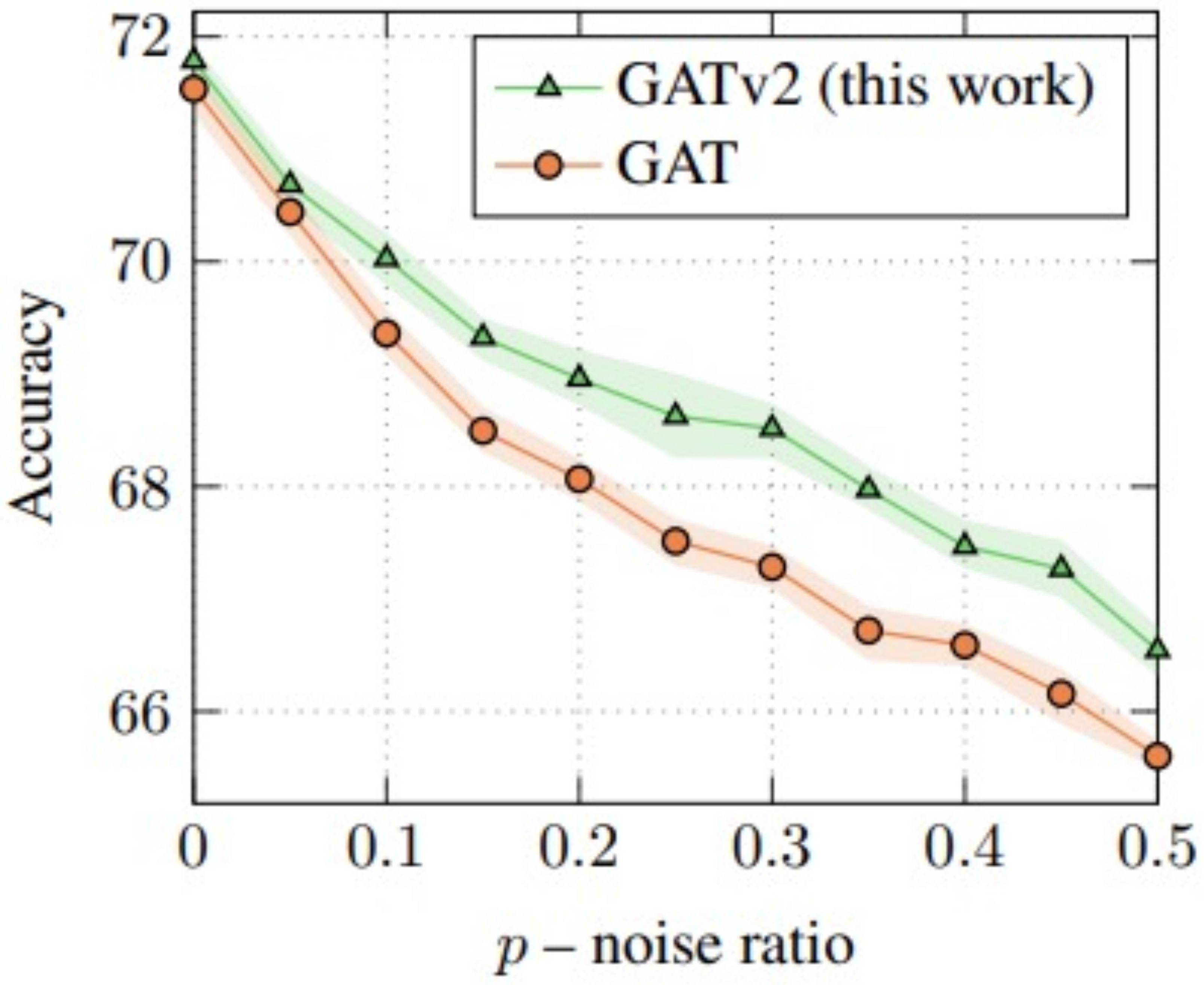




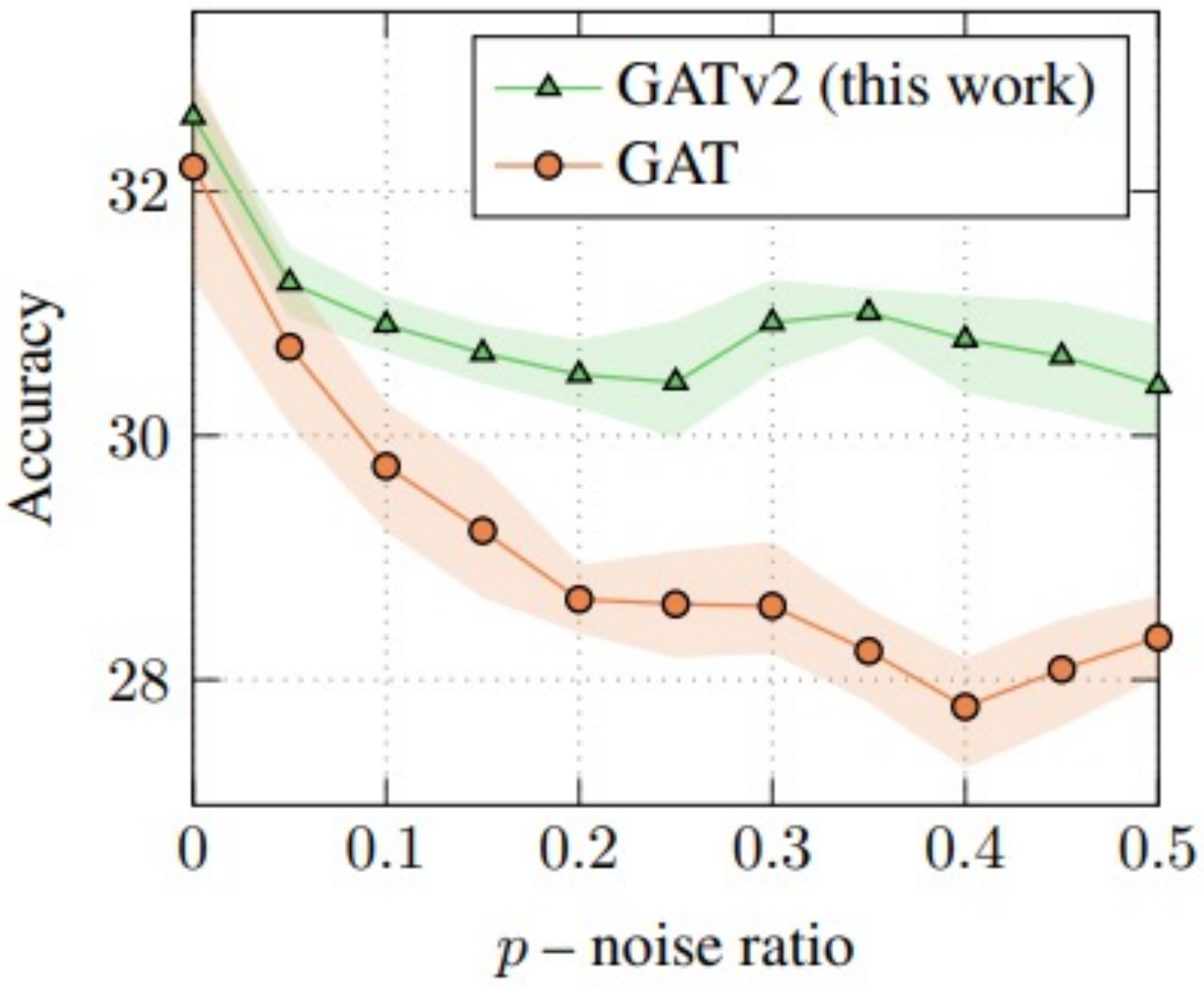
There exist learned parameters that “implement” this mapping

Experimental Results

- **GATv2** always outperformed **GAT** in 12 benchmarks of node- link- and graph-prediction
- **GATv2** is more robust to noisy edges (which did not exist in the original graph)



(a) **ogbn-arxiv**



(b) **ogbn-mag**

Summary

- Define **static attention** vs. **dynamic attention**
- **GAT** computes **static attention**
- **GATv2**: a simple modification that is strictly more expressive than **GAT**
  - More accurate across 12 benchmarks and more robust to noise
- Use **GATv2** instead of **GAT** whenever possible
- **GATv2** is available on:
  - PyTorch Geometric: **from** torch_geometric.nn **import** GATv2Conv
  - DGL: **from** dgl.nn.pytorch **import** GATv2Conv
  - TensorFlow GNN: **from** tensorflow_gnn.keras.layers **import** GATv2

shakedbr@cs.technion.ac.il

http://shakedbr.cswp.cs.technion.ac.il