ICML
International Conference
On Machine Learning

NVIDIA

Subgraphormer: Unifying Subgraph GNNs and Graph Transformers via Graph Products

GBS
(Technion)

Beatrice Bevilacqua
(Purdue)

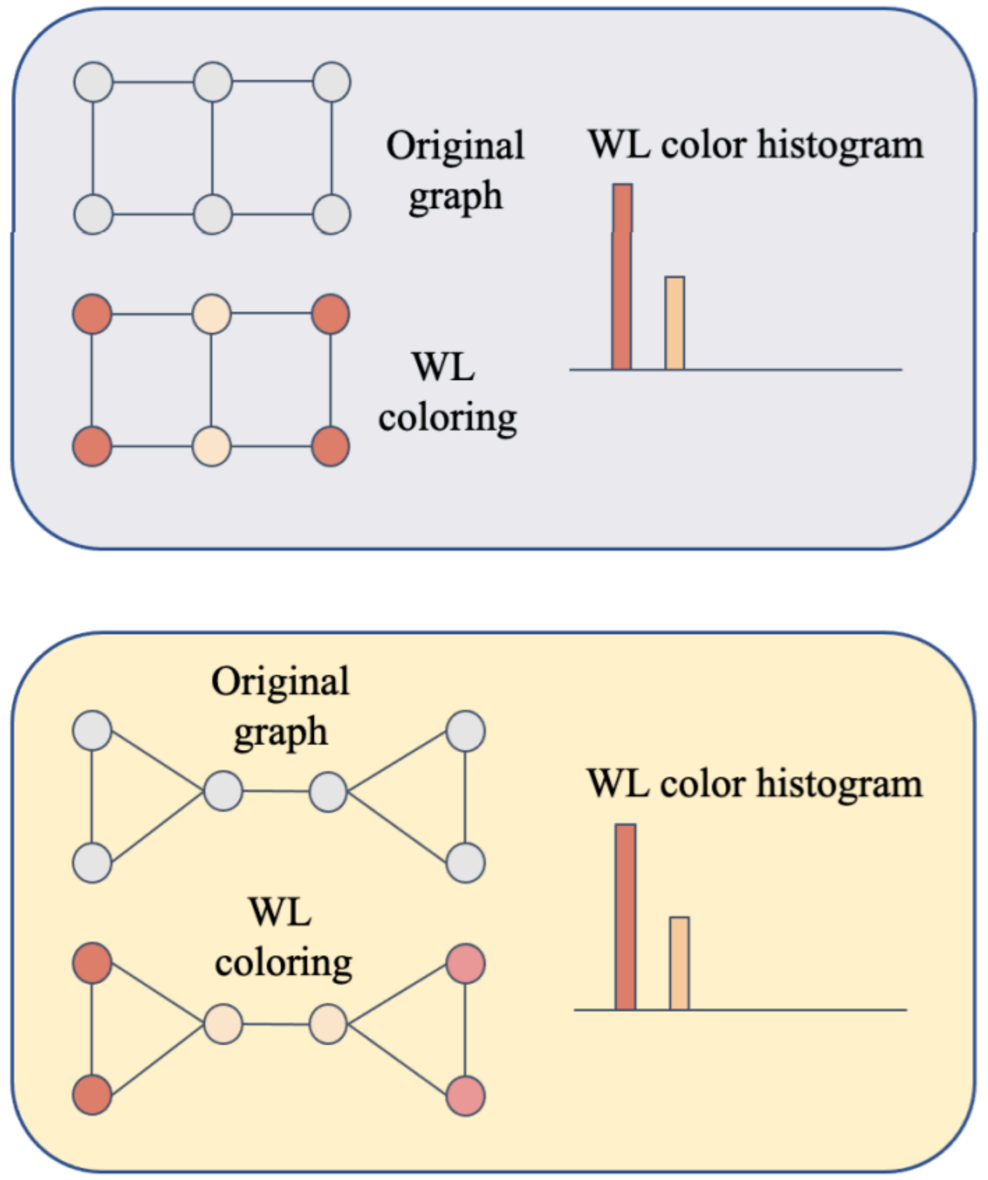
Haggai Maron
(Technion, Nvidia)

Outline

- Subgraph GNNs
- Graph Transformers
- **Subgraphormer: Unifying Subgraph GNNs and Graph Transformers via Graph Products**

Subgraph GNNs

- **Main idea -** graphs as sets of subgraphs
- **Motivation:** even if MPNNs can't distinguish two graphs, their subgraphs might be easily separable

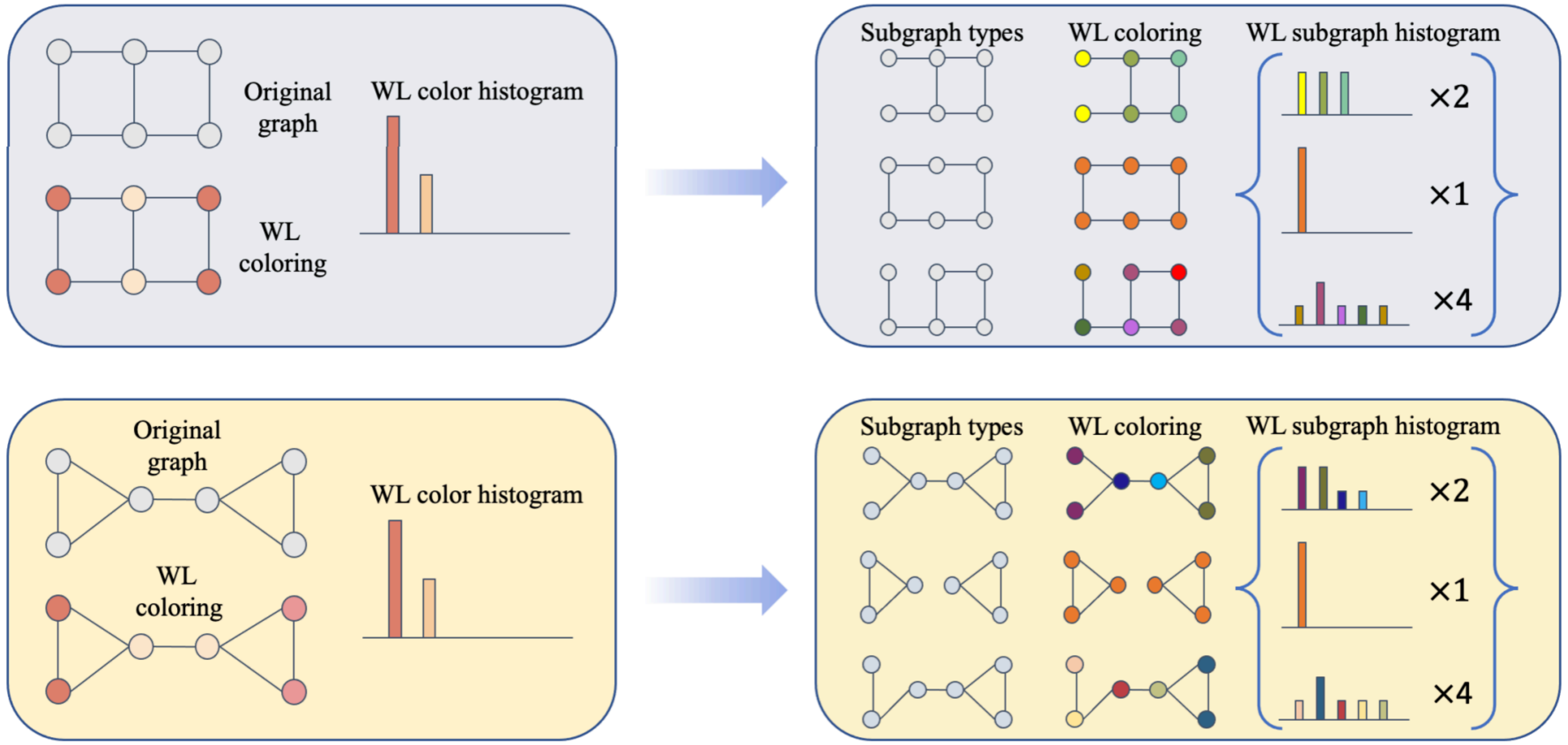


Taken from [1]

[1] Bevi. et al. 2022

Subgraph GNNs

- **Main idea -** graphs as sets of subgraphs
- **Motivation:** even if MPNNs can't distinguish two graphs, their subgraphs might be easily separable



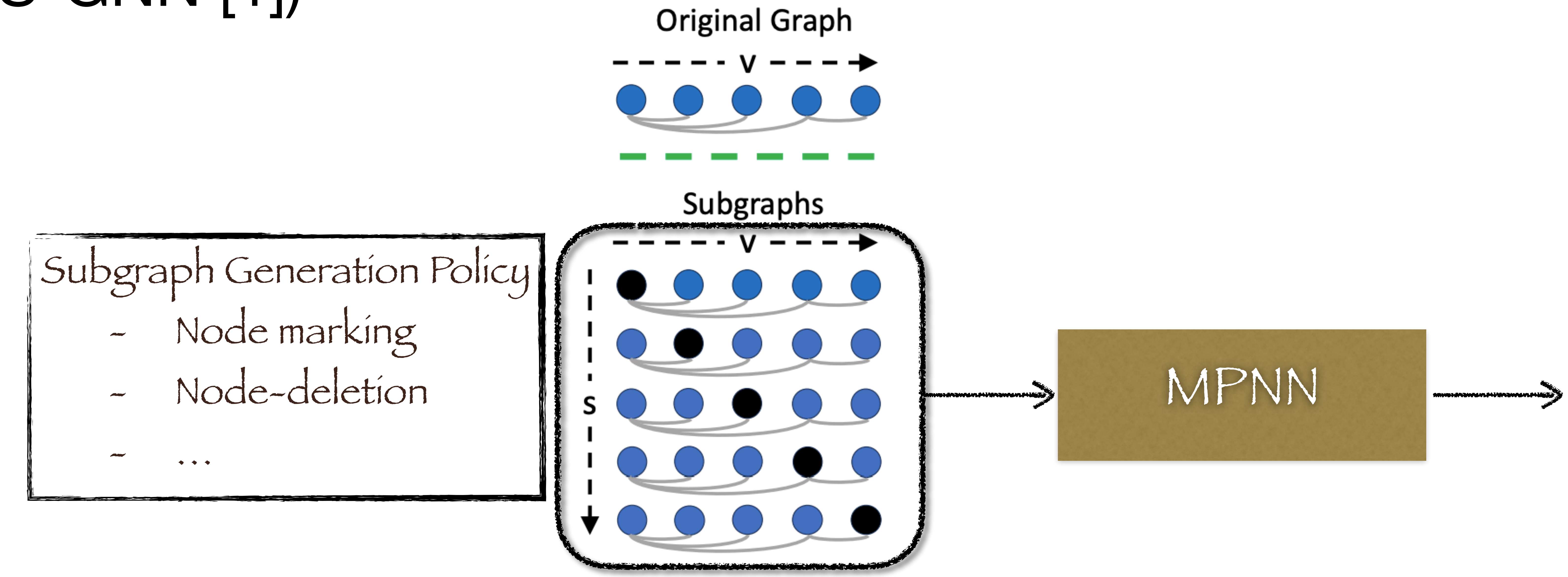
Taken from [1]

[1] Bevi. et al. 2022

Subgraph GNNs

- How can the graph representations be learned?
- Map a graph into a set of subgraphs (bag) via selection policy

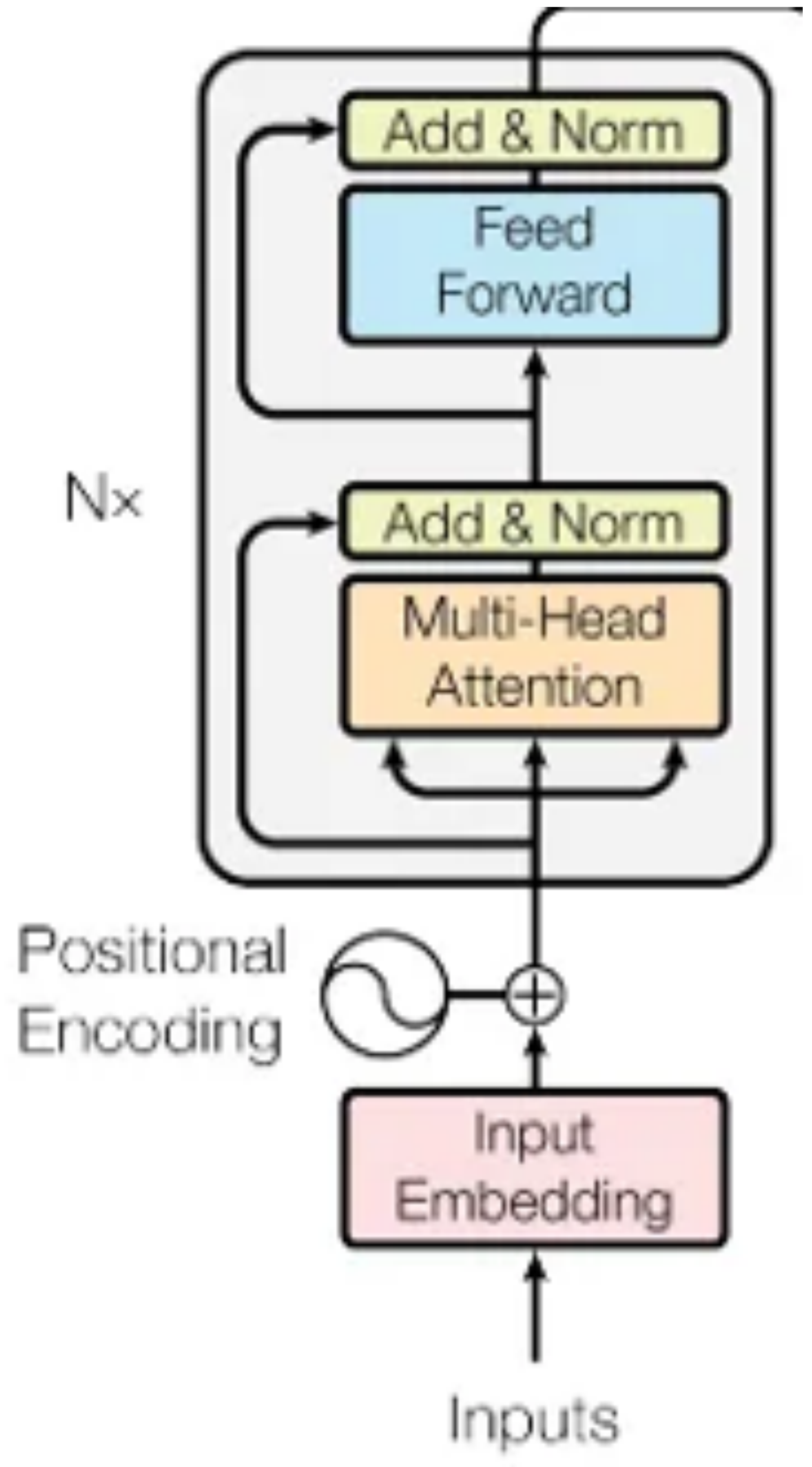
  For Isomorphic graphs the bag must (ideally) be the same
- Process the bag in a principled way, e.g., MPNN on each subgraph followed by pooling (DS-GNN [1])



[1] Bevi. et al. 2022

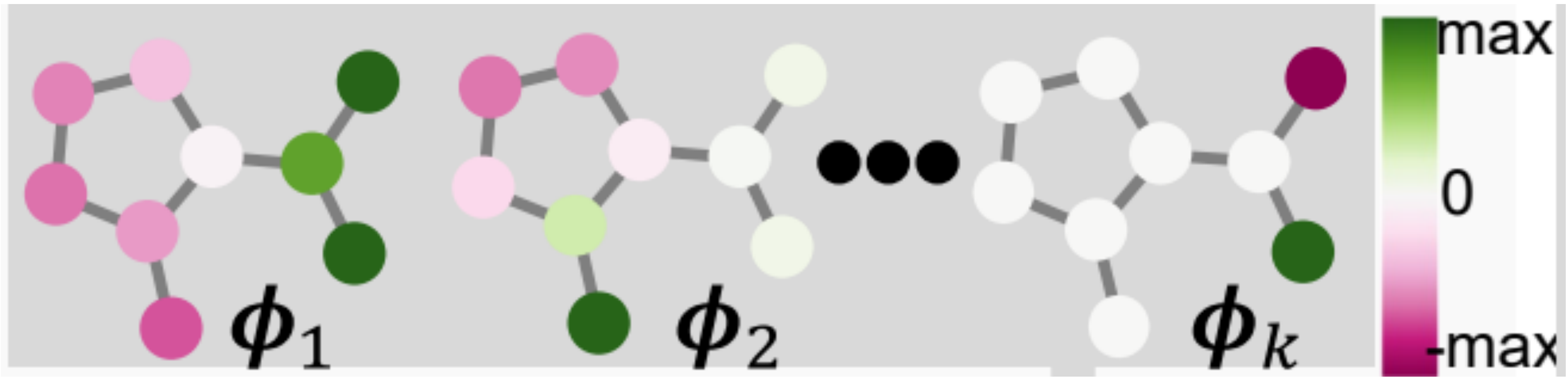
Graph Transformers

- **Recipe:**
  - Positional Encodings (PE)
  - Attention-based aggregations

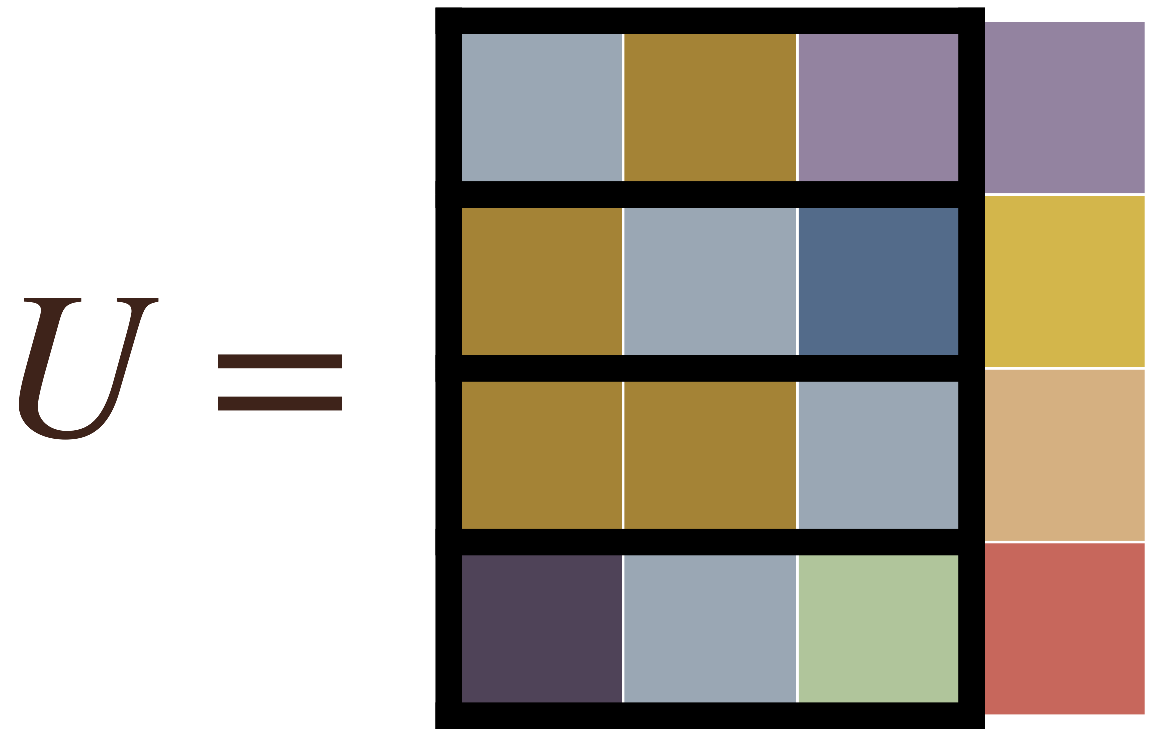


Graph Transformers

- Positional Encodings (PE):
  - Laplacian: $L = D - A$
  - Eigendecomposition: $L = U^T \Lambda U$
  - Use rows of U as node features — PE

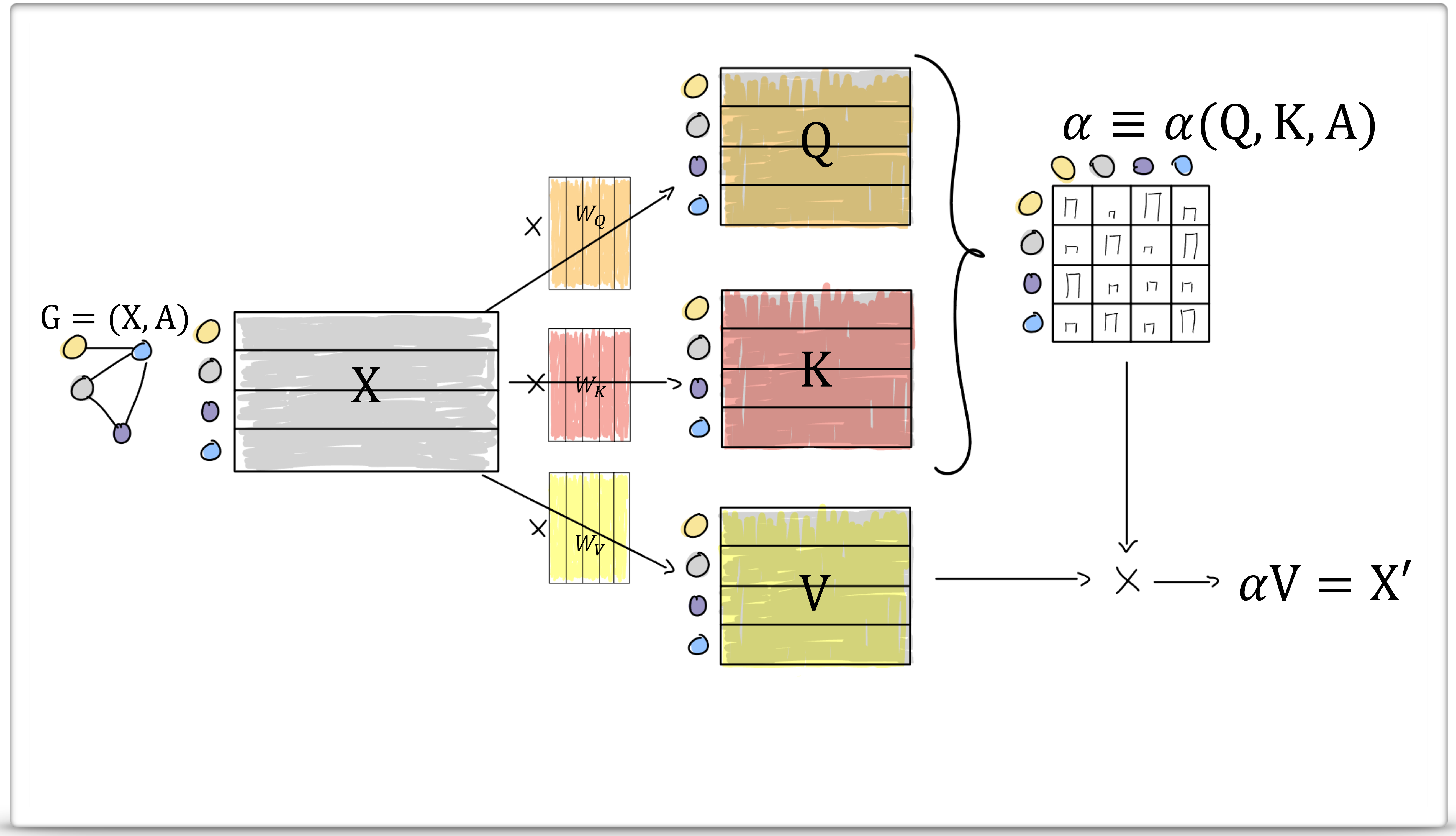


Taken from [1]



[1] Ramp. et al. 2022

Graph Transformers



- Attention
  - GAT [1]
  - GATV2 [2]
  - Standard Attention [3]
  - Sparse Attention [4]

[1] Velic. et al., 2018
[2] Brod. Et al., 2021
[3] Vasw et al., 2021
[4] Krzy. Et al., 2021

Subgraphormer

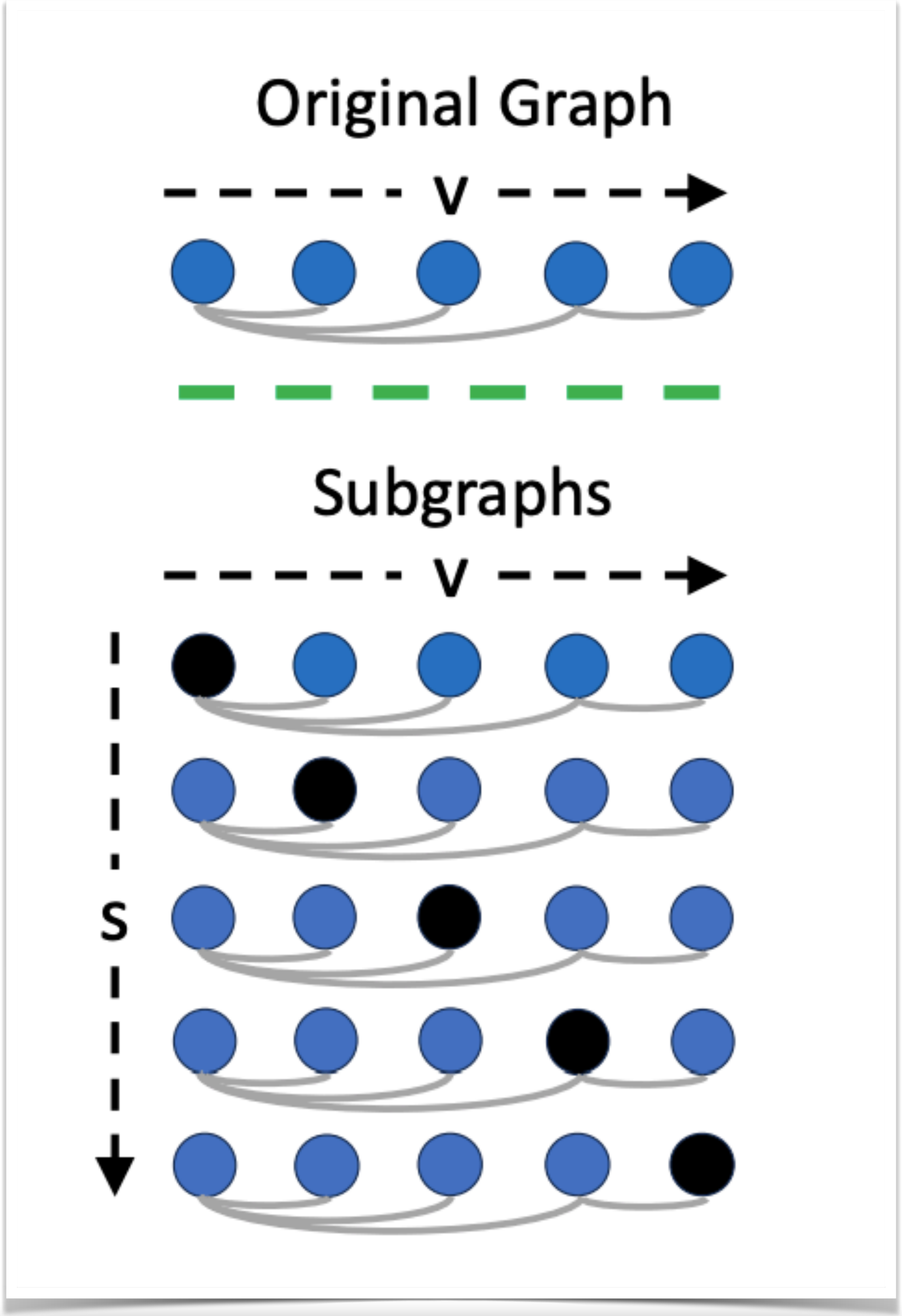
Main idea

- Two main components:
  - Attention based aggregations
  - Subgraph Positional encodings

Subgraphormer

Subgraph GNNs as MPNNs

Notation:
$\mathcal{X}(s, v)$ - the feature of node v in Subgraph s



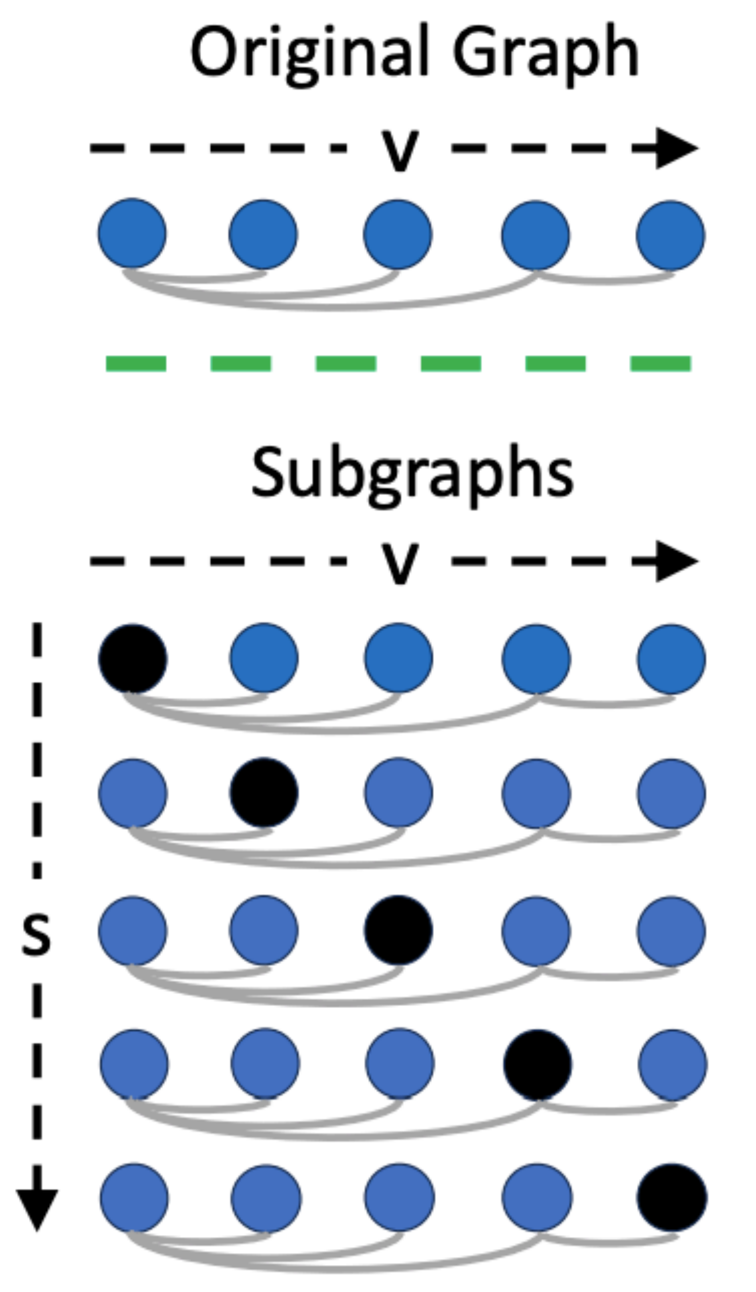
Subgraphormer

Subgraph GNNs as MPNNs

Notation:
$\mathcal{X}(s,v)$ - the feature of node v in Subgraph s

- The following update is the most expressive* Subgraph GNN (GNN-SSWL+ [1]):

$$\mathcal{X}(s,v)^{t+1} = f^t\Big(\mathcal{X}(s,v)^t, \mathcal{X}(v,v)^t, \{\mathcal{X}(s,v')^t\}_{v'\sim_G v}, \{\mathcal{X}(s',v)^t\}_{s'\sim_G s}\Big)$$



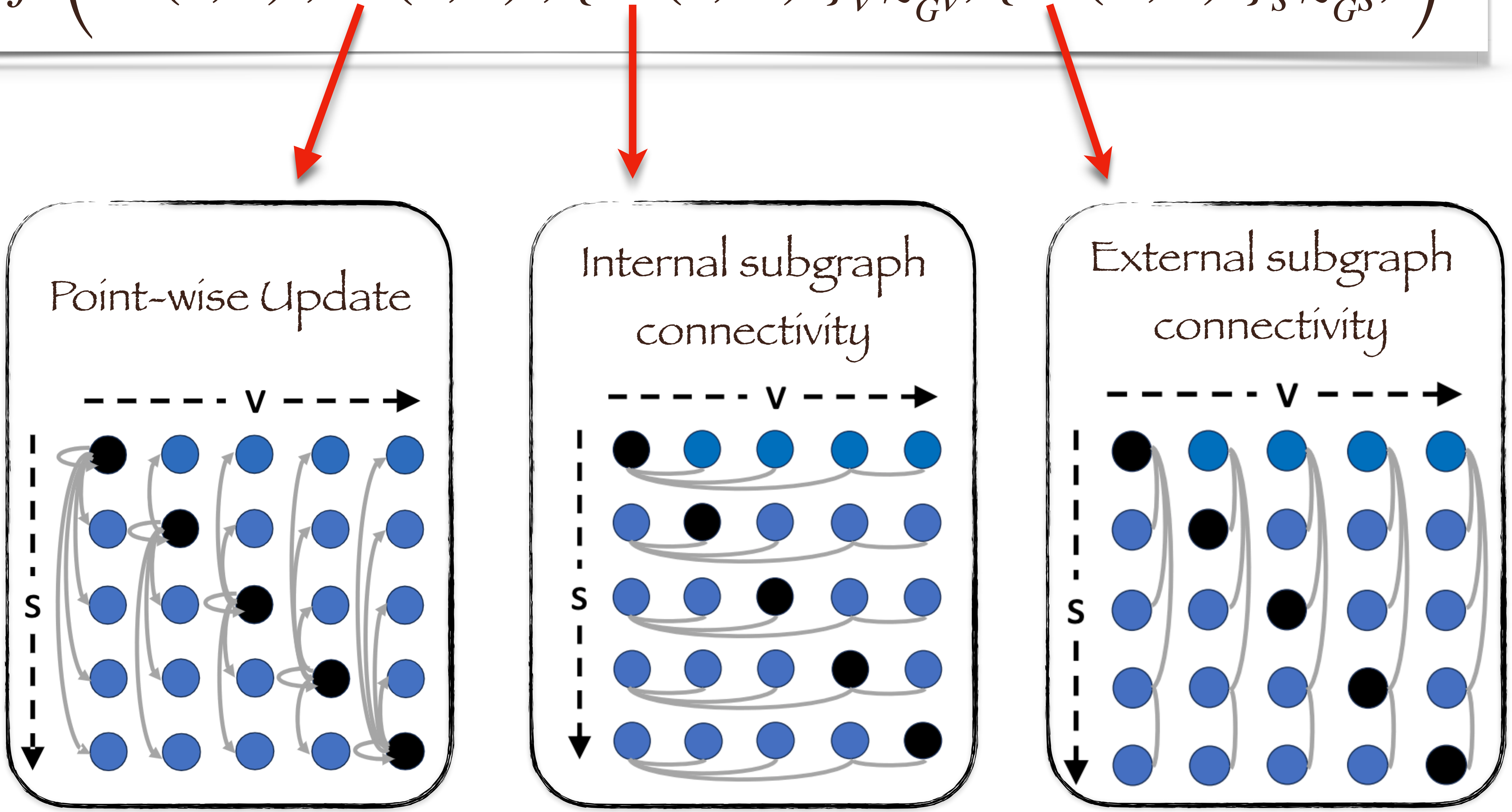
[1] Zhang et al. 2023

* Only internal/External are required for Maximal expressivity

Subgraphormer

Subgraph GNNs as MPNNs

$$\mathcal{X}(s,v)^{t+1} = f^t\Big(\mathcal{X}(s,v)^t, \mathcal{X}(v,v)^t, \{\mathcal{X}(s,v')^t\}_{v'\sim_G v}, \{\mathcal{X}(s',v)^t\}_{s'\sim_G s}\Big)$$



* Only internal/External are required for Maximal expressivity

Subgraphormer

Subgraph GNNs as MPNNs

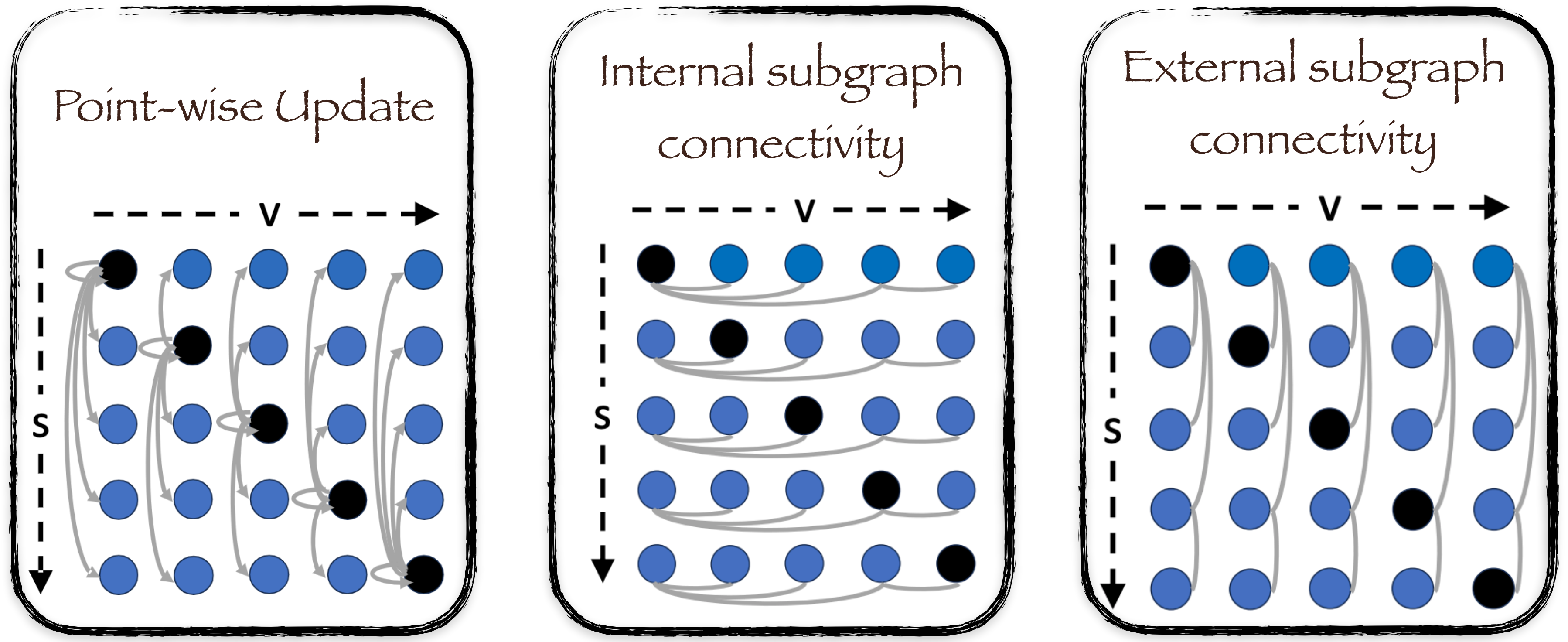
- Subgraph GNNs — just MPNNs on a product graph!
- Don't change the MPNN — change the graph!

Definition (Product Graph):
A product graph is a heterogeneous graph, defined by a feature matrix $\mathcal{X} \in \mathbb{R}^{n^2 \times d}$, and a **set** of adjacency matrices, $\mathcal{A} \in \mathbb{R}^{n^2 \times n^2}$.

Proposition 3.1 (GNN-SSWL+ as MPNNs):
GNN-SSWL+ update equation can be realized via RGCN layers on this product graph.

$$\mathcal{X}(s,v)^{t+1} = f^t\Big(\mathcal{X}(s,v)^t, \mathcal{X}(v,v)^t, \{\mathcal{X}(s,v')^t\}_{v' \sim_G v}, \{\mathcal{X}(s',v)^t\}_{s' \sim_G s}\Big) \sim \text{RGCN}$$

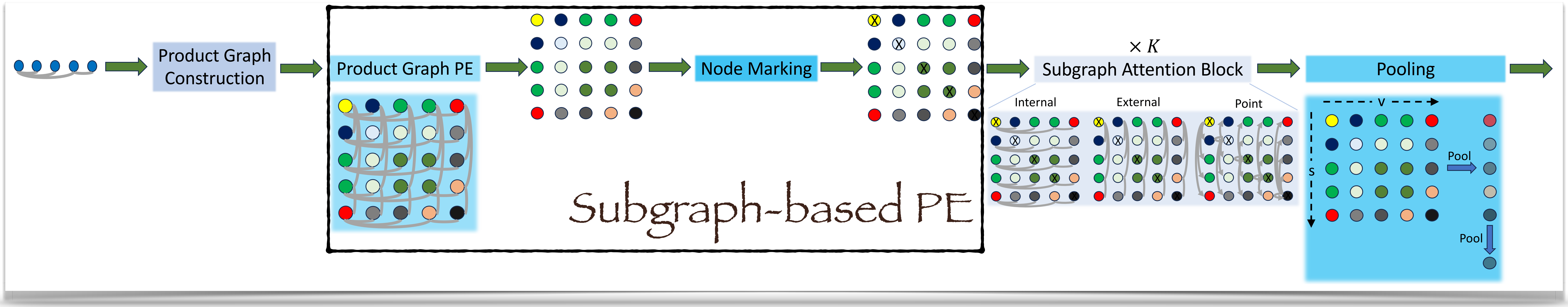
[1]



[1] Schlic. et al. 2017

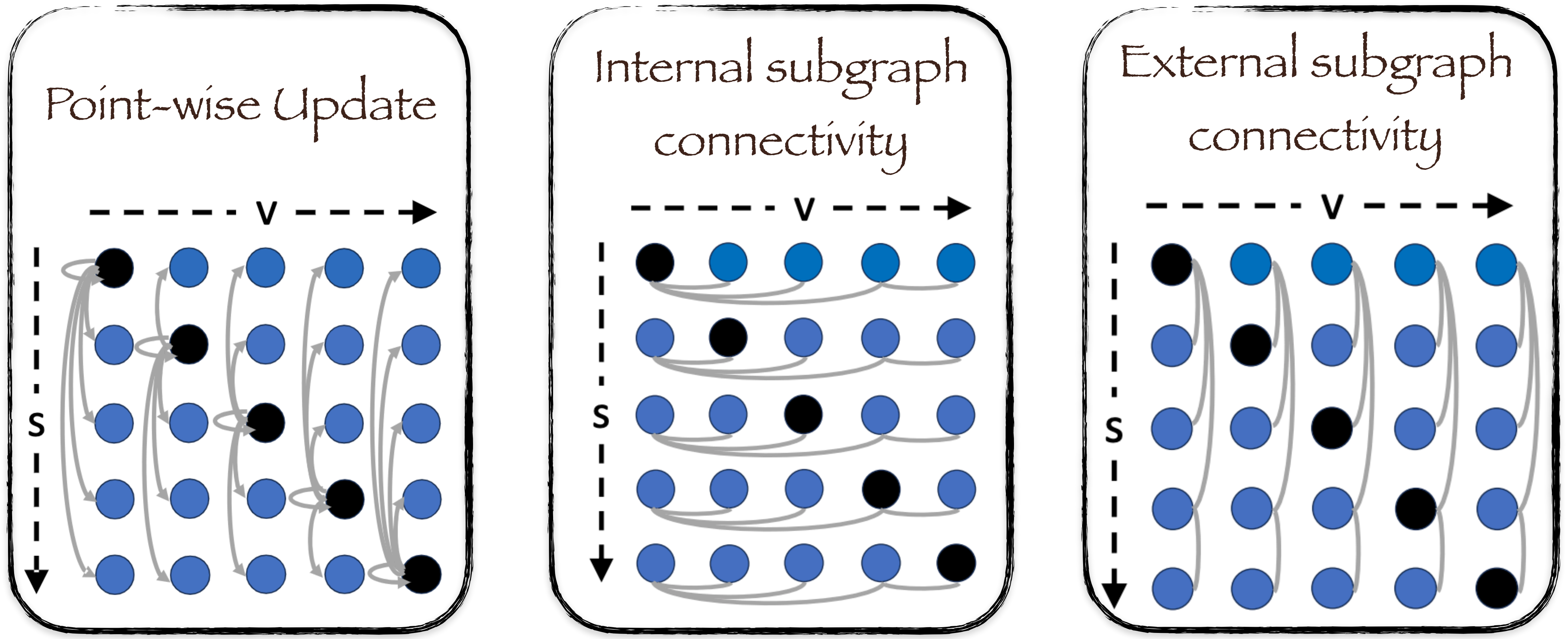
Subgraphormer

Subgraph-Based PE



Subgraphormer

Subgraph-Based PE



$\mathcal{A}_G$ $\mathcal{A}_{G^S}$

- **What is the challenge?**

1. Adjacency: Which adjacency should we use?

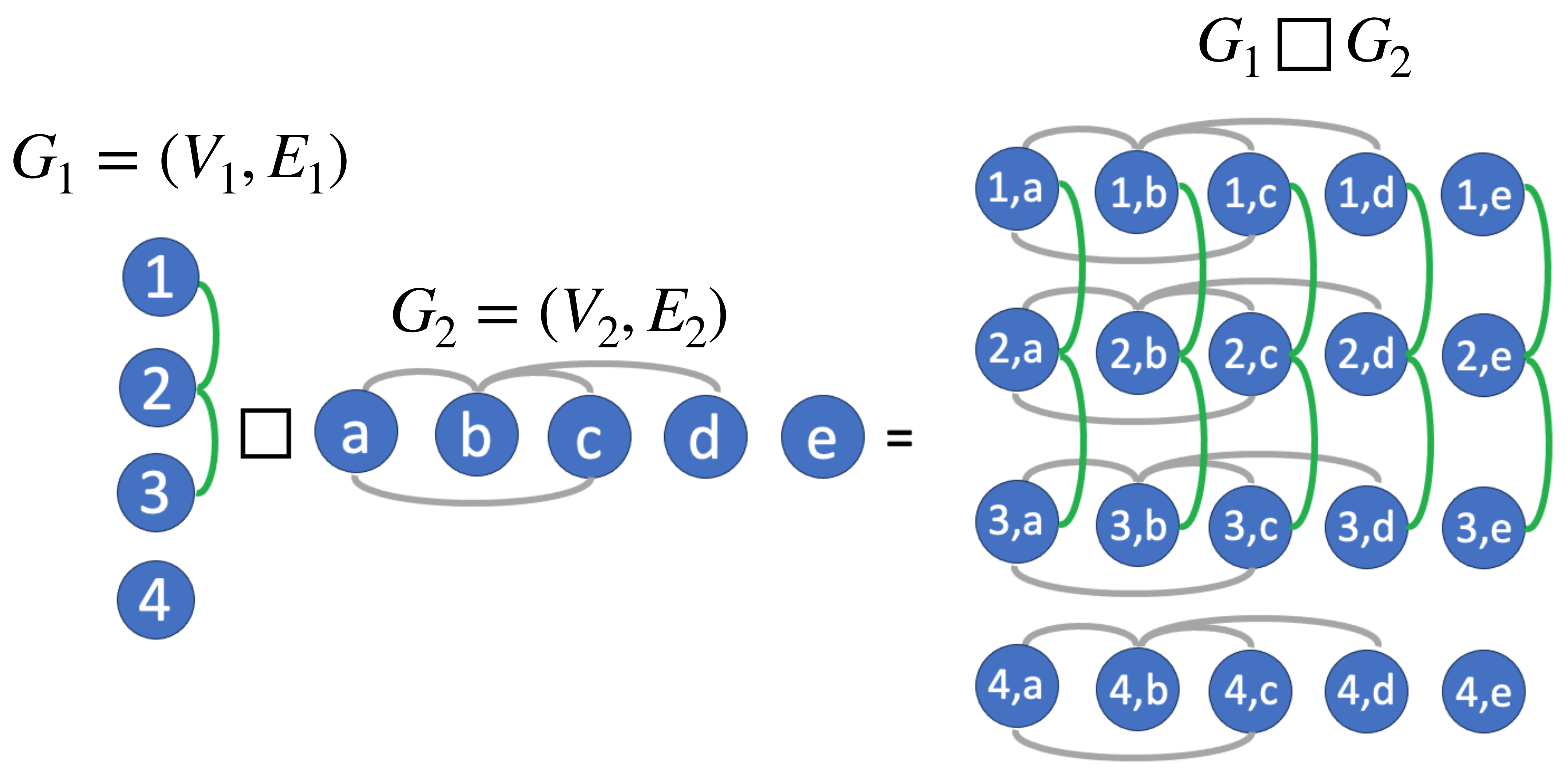
$\mathcal{A}_G, \mathcal{A}_{G^S}$ — hold information about the original graph's topology.

2. Efficiency: $\mathcal{A}_G, \mathcal{A}_{G^S} \in \mathbb{R}^{n^2 \times n^2}$, applying standard eigendecomposition is not an option — $\mathcal{O}(n^4 \cdot k)$

Subgraphormer

Subgraph-Based PE — Graph Cartesian product

- $G_1 = (V_1, E_1)$ with adjacency A_1
- $G_2 = (V_2, E_2)$ with adjacency A_2
- Cartesian Product Graph $G_1 \square G_2$
- Vertex set $V_{G_1 \square G_2} \triangleq V_1 \times V_2$
- $(x, y) \sim_{G_1 \square G_2} (x', y') \iff$
  - $x = x'$ and $y \sim_{G_2} y'$
  - $y = y'$ and $x \sim_{G_1} x'$



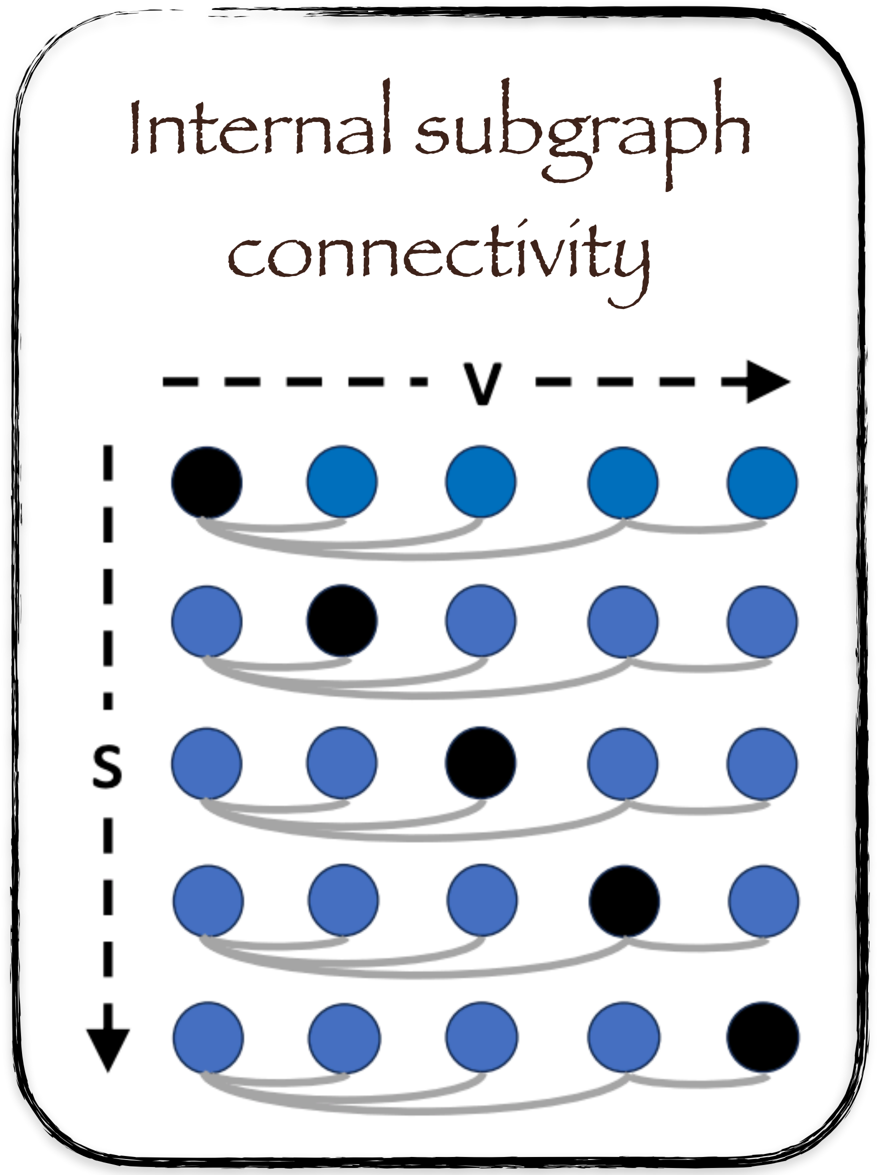
$$\mathscr{A}_{G_1 \square G_2} = A_2 \otimes I + I \otimes A_1$$

Subgraphormer

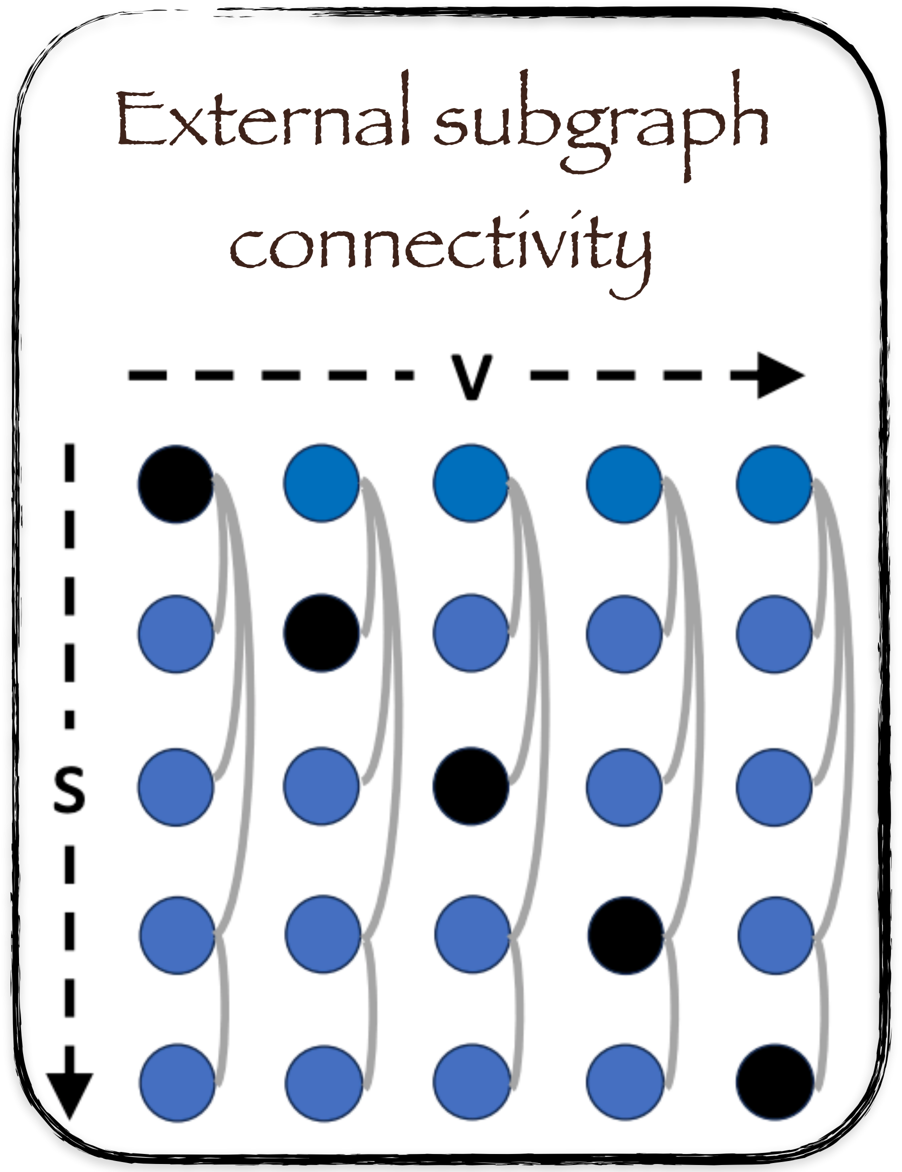
Subgraph-Based PE — Graph Cartesian product

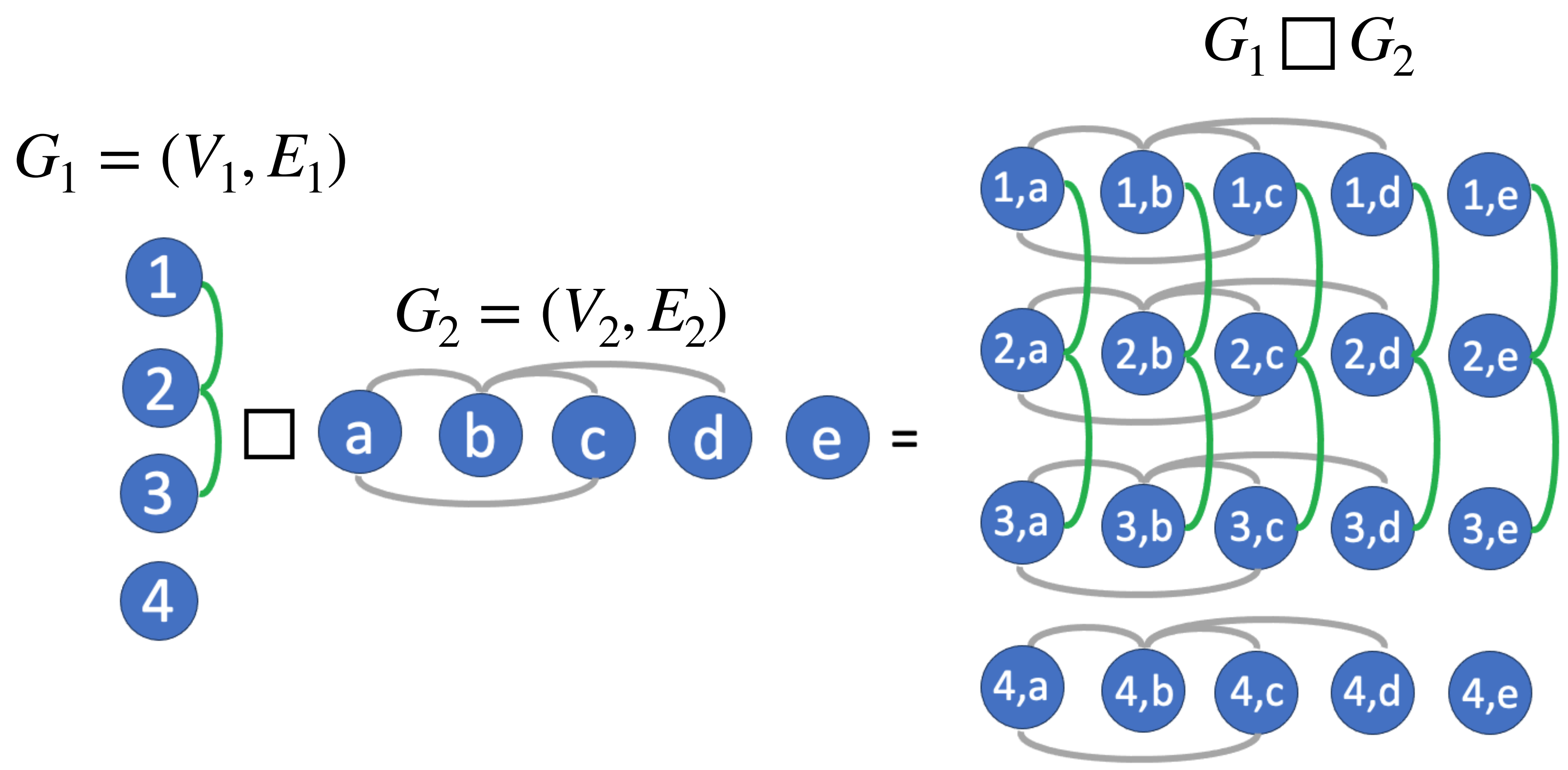
- The internal and external connectivities have a special structure

$\mathscr{A}_G$



$\mathscr{A}_{G^S}$



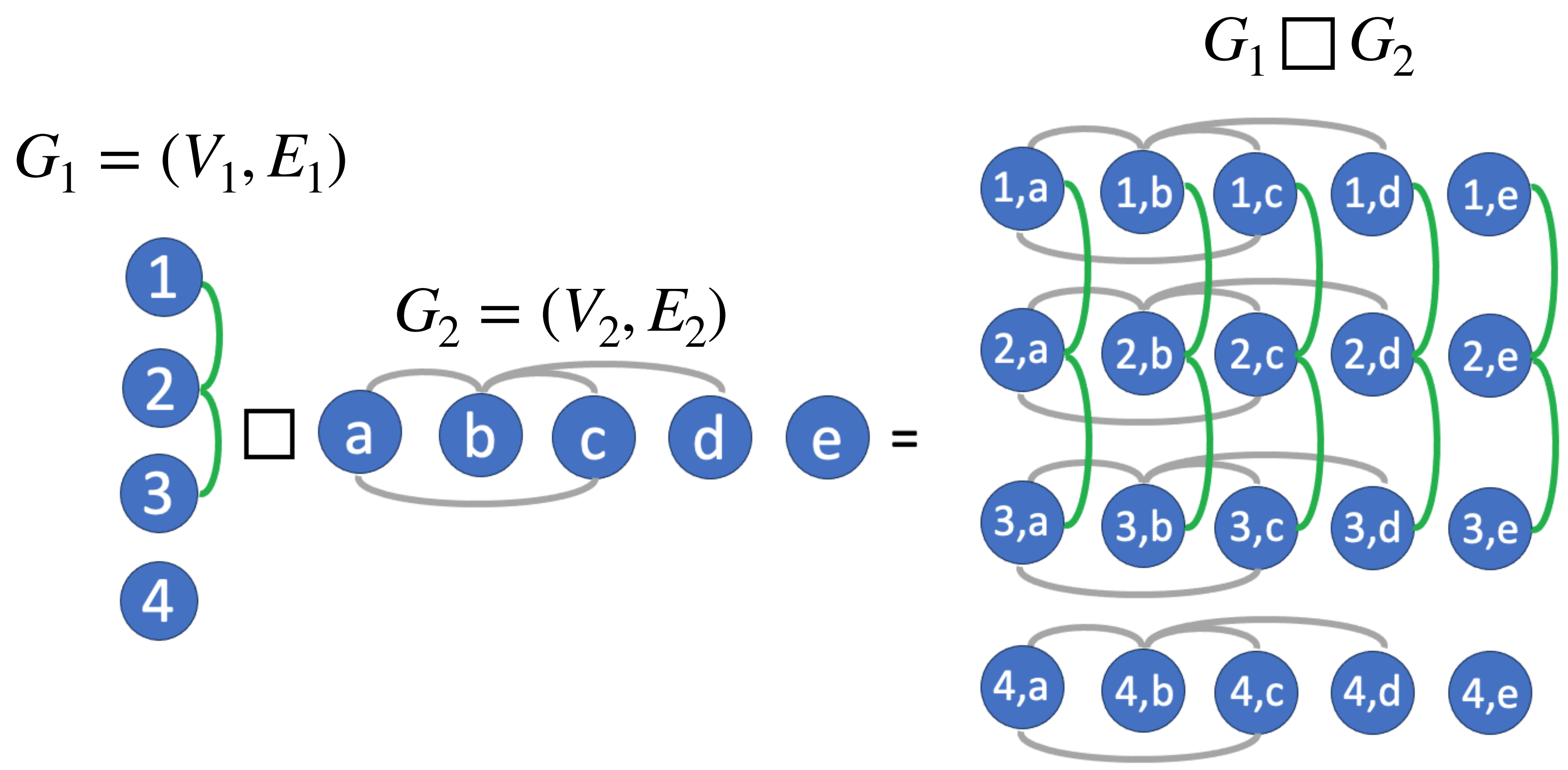


$$\mathscr{A}_{G_1 \square G_2} = A_2 \otimes I + I \otimes A_1$$

Subgraphormer

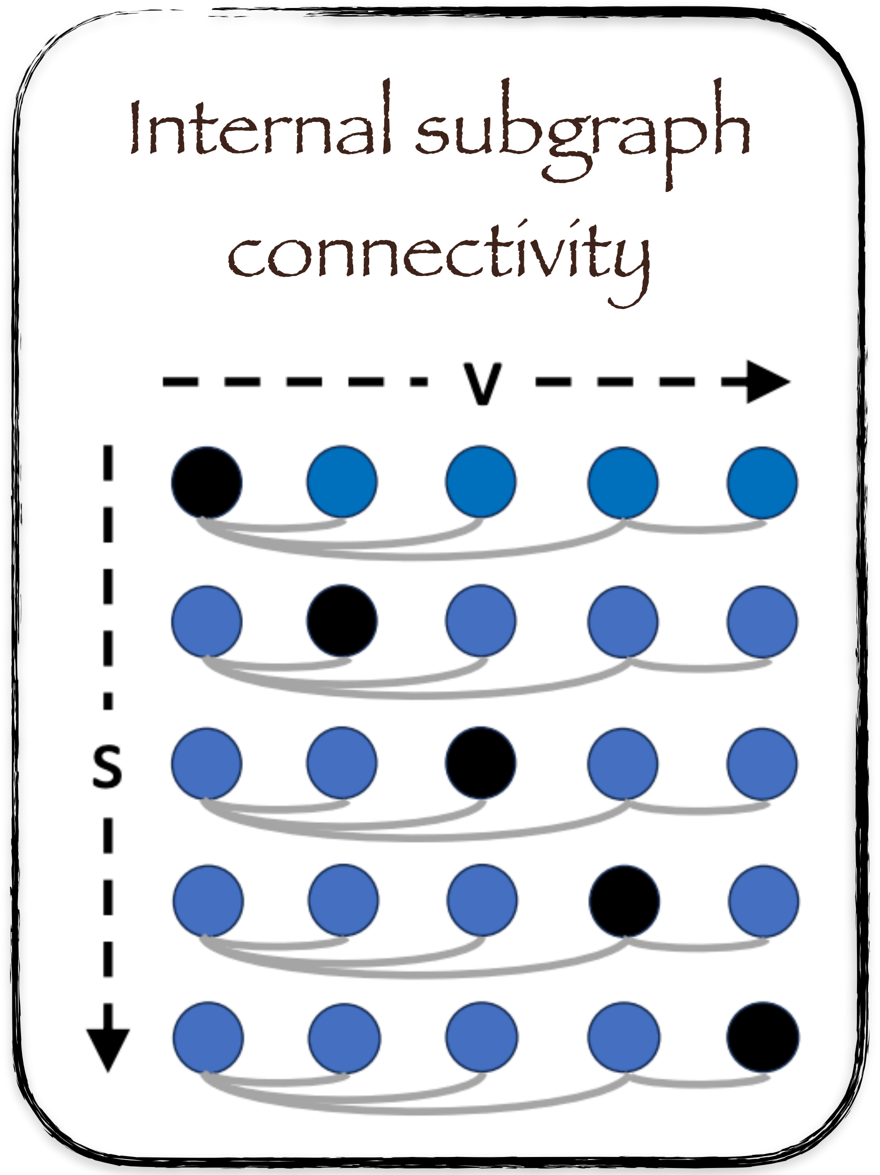
Subgraph-Based PE — Graph Cartesian product

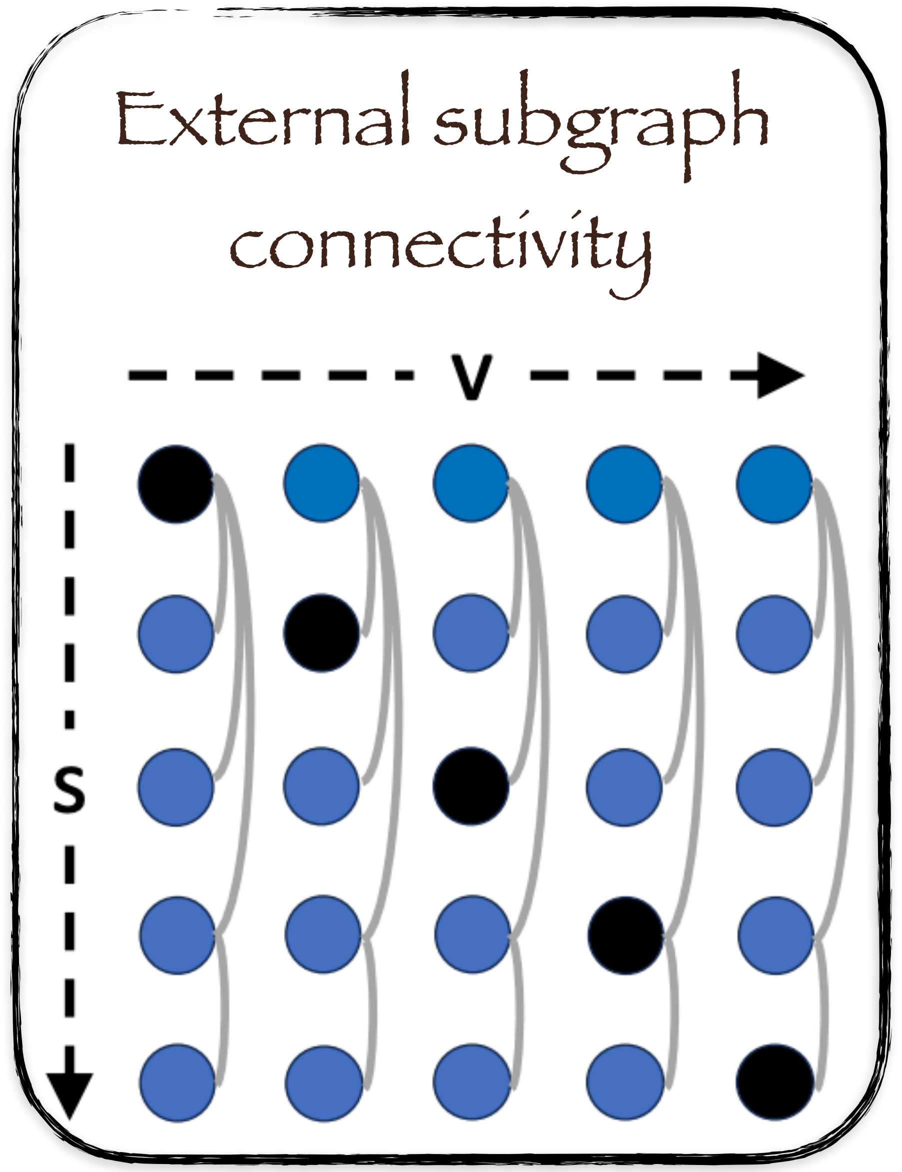
- The internal and external connectivities have a special structure



$$\mathcal{A}_{G_1 \square G_2} = A_2 \otimes I + I \otimes A_1$$

$$\mathcal{A}_G = I \otimes A \qquad \mathcal{A}_{G^S} = A \otimes I$$





Proposition 3.2:

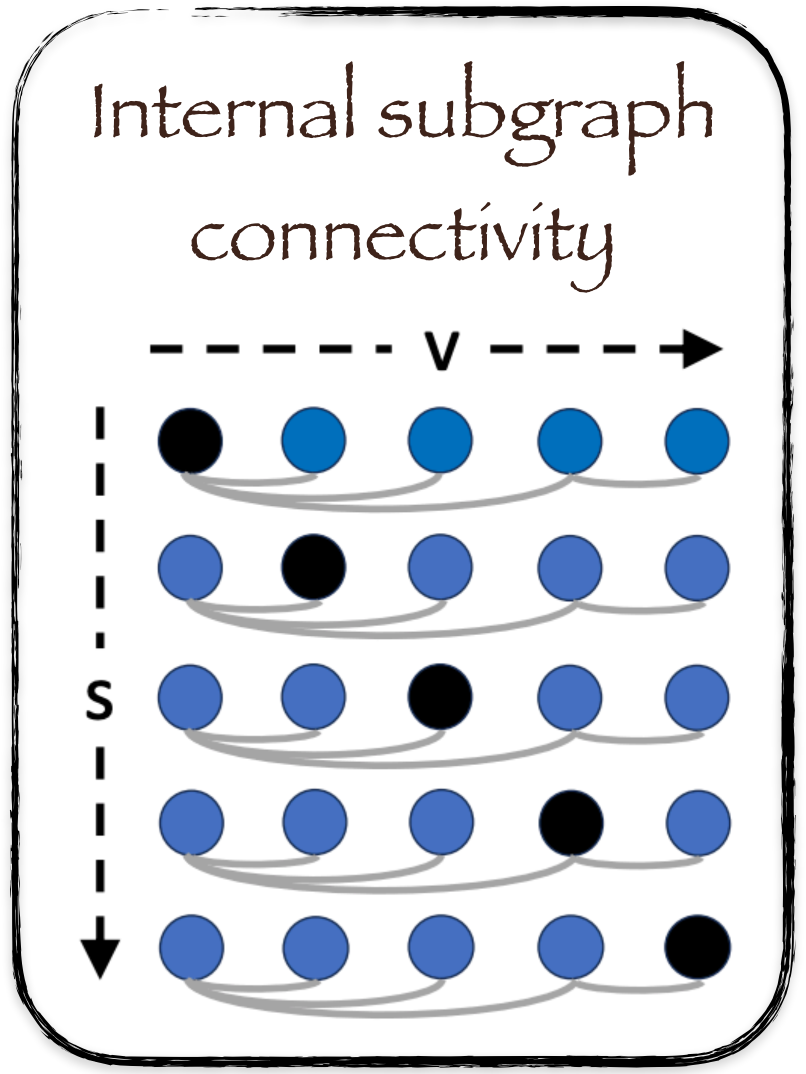
Taking $G \square G$ we get internal and external adjacencies

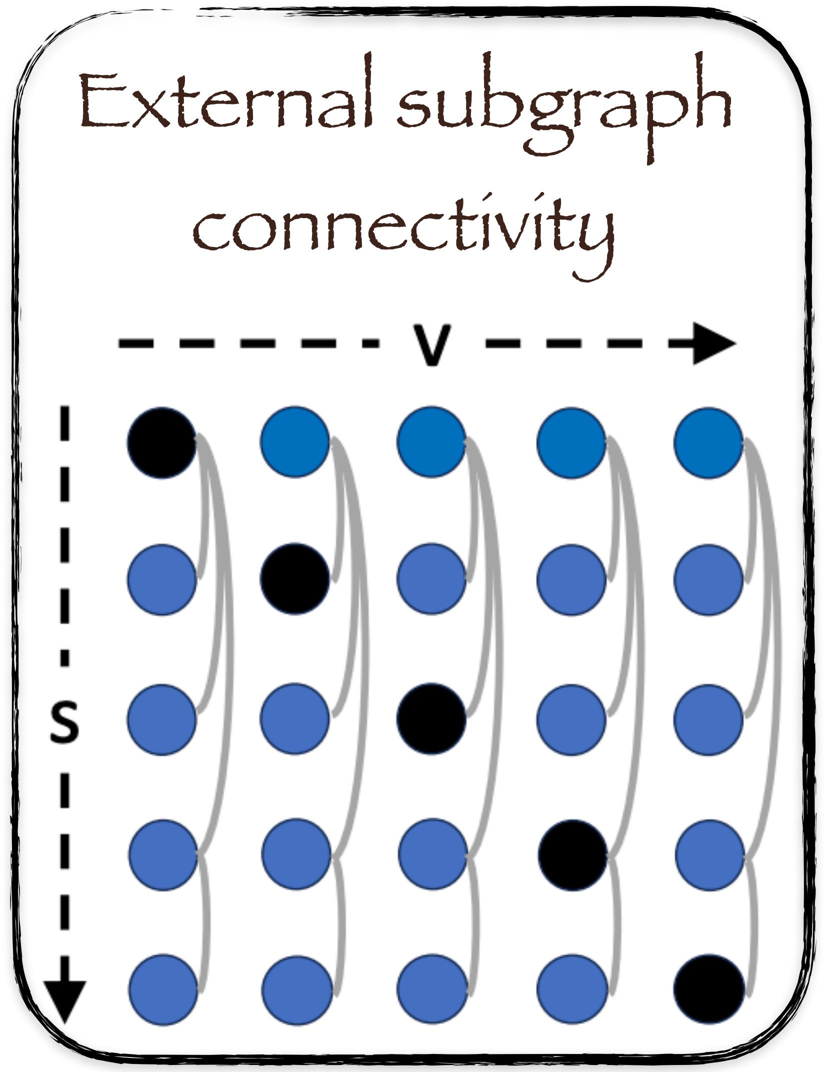
$$\mathcal{A}_{G \square G} = \overbrace{A_{G^S}}^{A \otimes I} + \overbrace{A_G}^{I \otimes A}$$

Subgraphormer

Subgraph-Based PE — Graph Cartesian product

$$\mathscr{A}_G = I \otimes A \qquad \mathscr{A}_{G^S} = A \otimes I$$





- The internal and external connectivities have a special structure

Proposition 3.2:
Taking $G \square G$ we get internal and external adjacencies

$$\mathscr{A}_{G \square G} = \overbrace{A_{G^S}}^{A \otimes I} + \overbrace{A_G}^{I \otimes A}$$

Proposition (Product Graph eigendecomposition) [1]: The eigenvectors and eigenvalues of $\mathscr{L}_{G \square G}$ are $\{(v_i \otimes v_j, \lambda_i + \lambda_j)\}_{i,j=1}^{n^2}$ where $\{(v_i, \lambda_i)\}_{i=1}^{n}$ are the eigenvectors and eigenvalues of the Laplacian matrix of G.

$\Rightarrow$

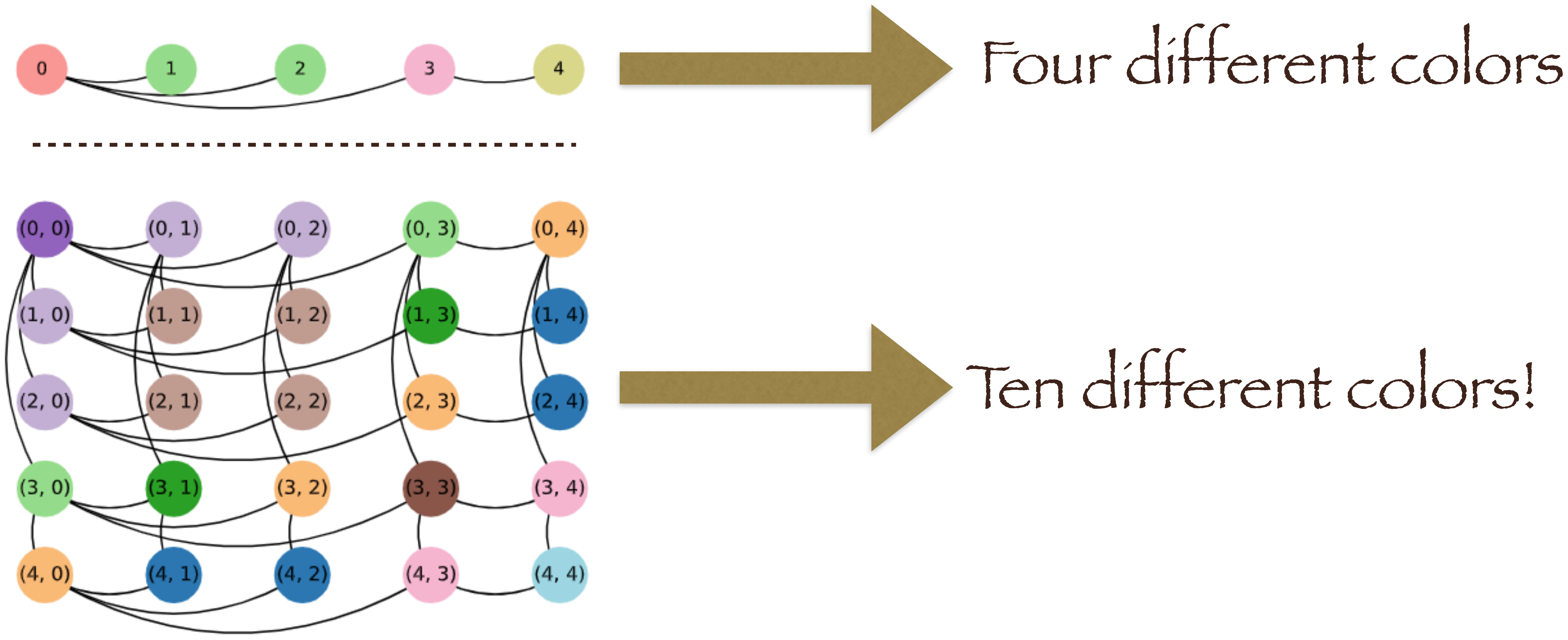
Only requires Eigendecomposition of the original (smaller) graph!

[1] Barik et al. 2015

Subgraphormer

Subgraph-Based PE — Visualization

- Visualization of the first (non - trivial) eigenvector



Subgraphormer

Experiments

Can Subgraphormer outperform Subgraph GNNs and Graph Transformers in real-world benchmarks?

Table 1: On the ZINC datasets, `Subgraphormer` outperforms Graph Transformers and Subgraph GNNs. The top three results are reported as **First**, **Second**, and **Third**.

Model ↓ / Dataset →	Param.	ZINC-12K (MAE ↓)	ZINC-FULL (MAE ↓)
GSN (Bouritsas et al., 2022)	500k	0.101±0.010	-
CIN (small) (Bodnar et al., 2021)	100k	0.094±0.004	0.044±0.003
GIN (Xu et al., 2018)	500k	0.163±0.004	-
PPGN++(6) (Puny et al., 2023)	500k	0.071±0.001	**0.020**±0.001
SAN (Kreuzer et al., 2021)	509k	0.139±0.006	-
URPE (Luo et al., 2022)	492k	0.086±0.007	0.028±0.002
GPS (Rampášek et al., 2022)	424k	**0.070**±0.004	-
Graphormer (Ying et al., 2021)	489k	0.122±0.006	0.052±0.005
Graphormer-GD (Zhang et al., 2023b)	503k	0.081±0.009	0.025±0.004
K-Subgraph SAT (Chen et al., 2022)	523k	0.094±0.008	-
NGNN (Zhang and Li, 2021)	500k	0.111±0.003	0.029±0.001
DS-GNN (Bevilacqua et al., 2022)	100k	0.116±0.009	-
DSS-GNN (Bevilacqua et al., 2022)	100k	0.102±0.003	0.029±0.003
GNN-AK (Zhao et al., 2022)	500k	0.105±0.010	-
GNN-AK+ (Zhao et al., 2022)	500k	0.091±0.002	-
SUN (Frasca et al., 2022)	526k	0.083±0.003	0.024±0.003
OSAN (Qian et al., 2022)	500k	0.154±0.008	-
DS-GNN (Bevilacqua et al., 2023)	500k	0.087±0.003	-
GNN-SSWL (Zhang et al., 2023a)	274k	0.082±0.003	0.026±0.001
GNN-SSWL+ (Zhang et al., 2023a)	387k	**0.070**±0.005	**0.022**±0.001
`Subgraphormer`	293k	**0.067**±0.007	**0.020**±0.002
`Subgraphormer + PE`	293k	**0.063**±0.001	**0.023**±0.001

Thanks for listening!