KAIST
清華大學 Tsinghua University
UNIVERSITA DEGLI STUDI DI TORINO

A Tutorial on Hypergraph Neural Networks: An In-Depth and Step-by-Step Guide

Part 3. Message Passing


Sunwoo Kim*


Soo Yong Lee*


Yue Gao


Alessia Antelmi


Mirko Polato


Kijung Shin

Part 3. Message Passing on Hypergraphs

The slides are available at https://sites.google.com/view/hnn-tutorial

SCAN ME

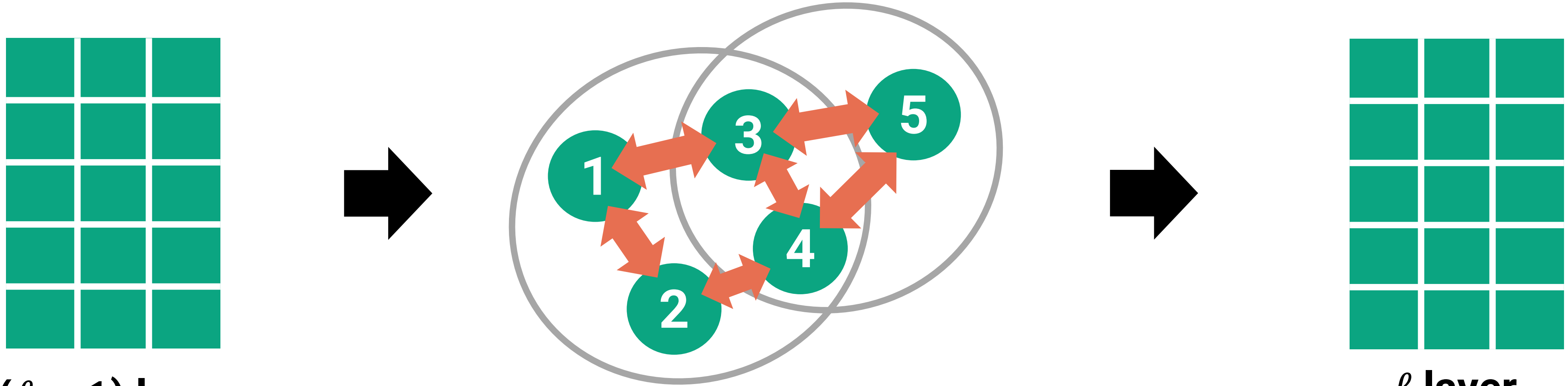
Presenters

Alessia Antelmi
Assistant Professor
University of Turin

Mirko Polato
Assistant Professor
University of Turin

What is message passing?

- HNNs learn node (and hyperedge) embeddings by aggregating information from other nodes (and hyperedges).
- This process is called **message passing**.



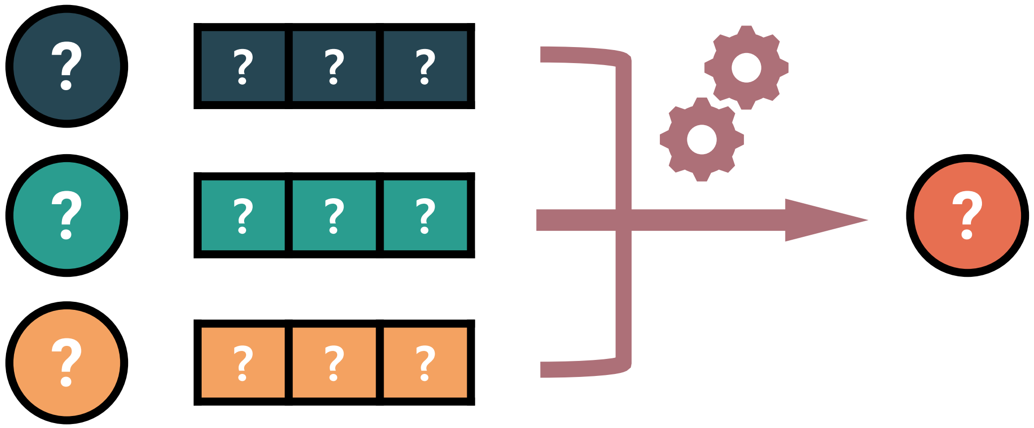
$(\ell - 1)$ layer node embeddings

Message passing

ℓ layer node embeddings

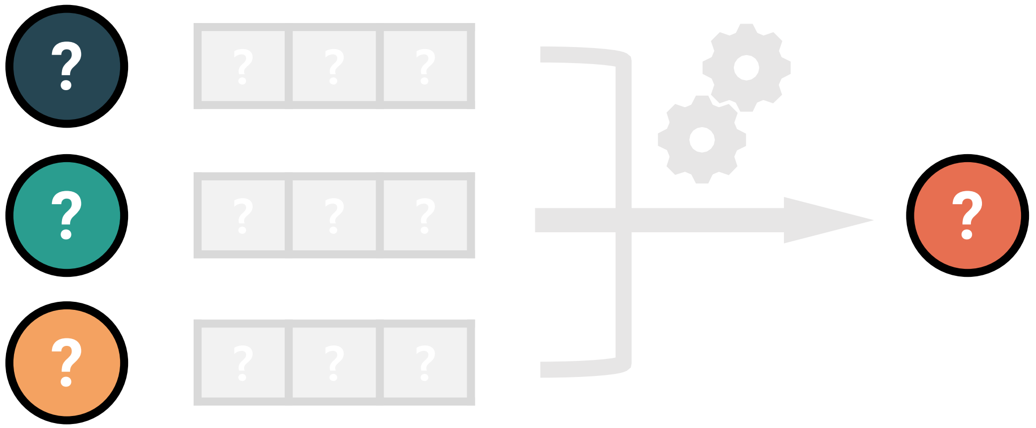
What is message passing? (cont.)

- In HNNs' message passing, some of the key issues involve:
  - Q1) **Whose** messages to aggregate
  - Q2) **What** messages to aggregate
  - Q3) **How** to aggregate messages



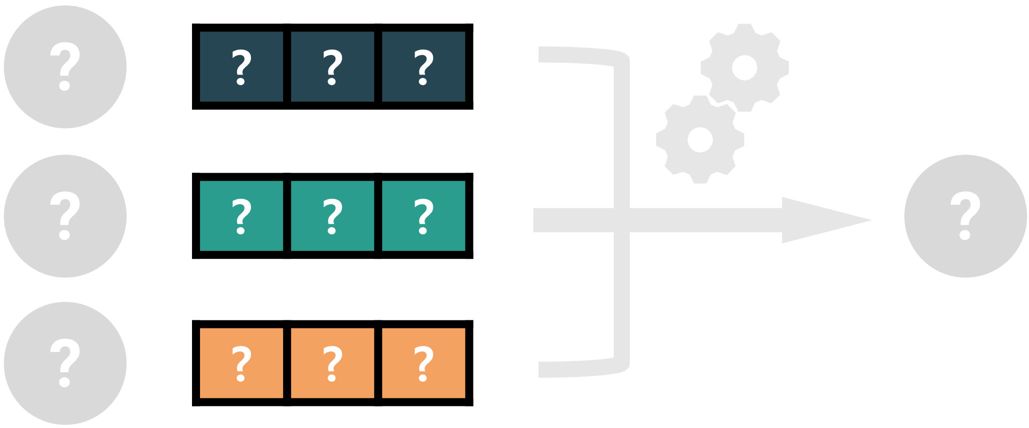
What is message passing? (cont.)

- In HNNs' message passing, some of the key issues involve:
  - Q1) **Whose** messages to aggregate
  - Q2) **What** messages to aggregate
  - Q3) **How** to aggregate messages



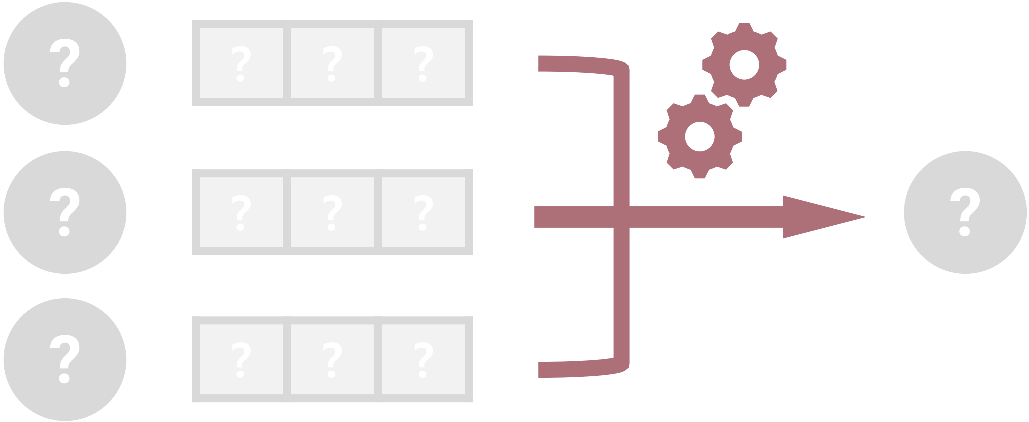
What is message passing? (cont.)

- In HNNs' message passing, some of the key issues involve:
  - Q1) **Whose** messages to aggregate
  - Q2) **What** messages to aggregate
  - Q3) **How** to aggregate messages



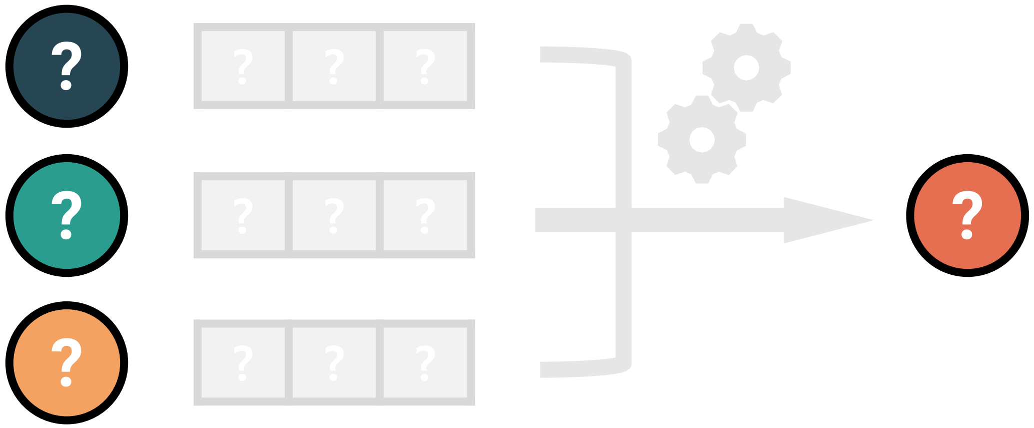
What is message passing? (cont.)

- In HNNs' message passing, some of the key issues involve:
  - Q1) **Whose** messages to aggregate
  - Q2) **What** messages to aggregate
  - Q3) **How** to aggregate messages



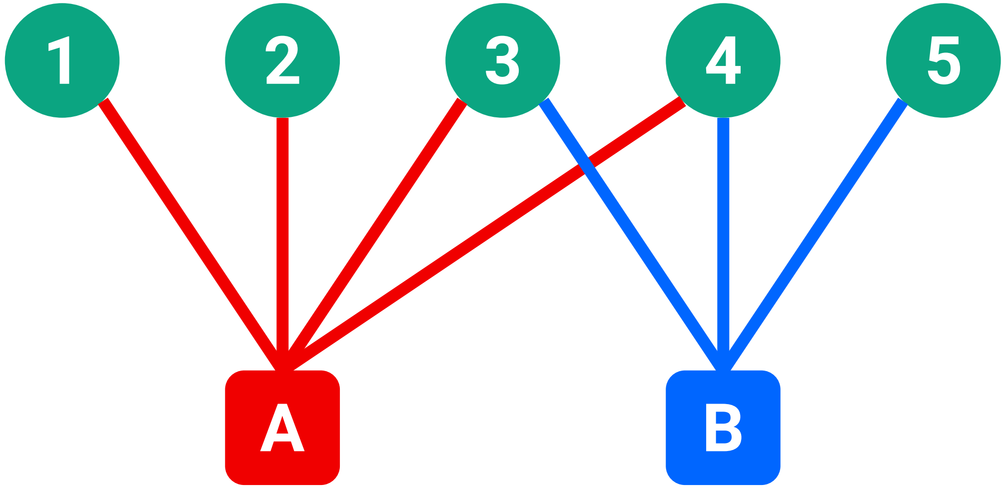
What is message passing? (cont.)

- In HNNs' message passing, some of the key issues involve:
  - Q1) **Whose** messages to aggregate
  - Q2) **What** messages to aggregate
  - Q3) **How** to aggregate messages

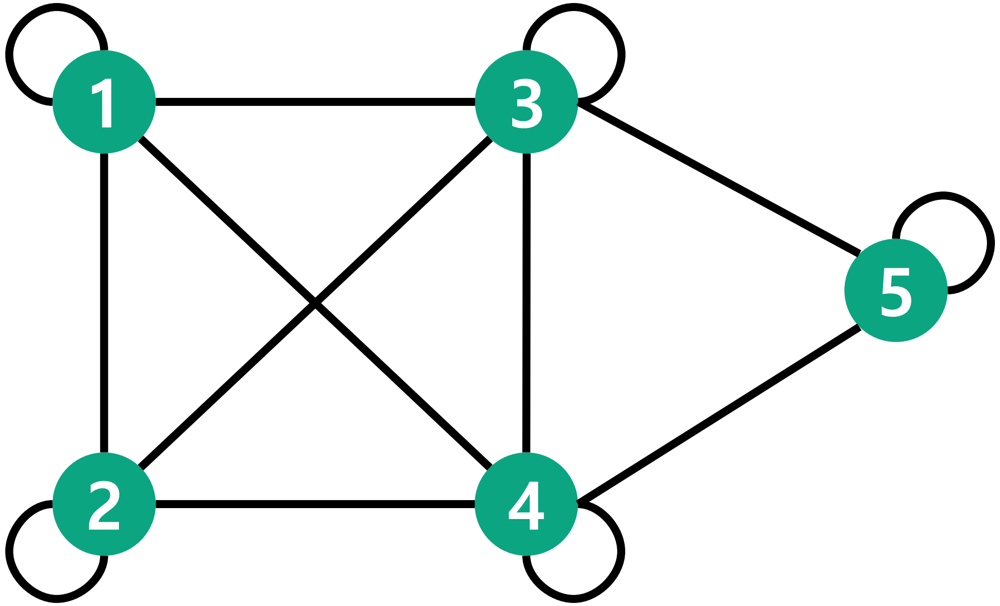


Q1) Whose messages to aggregate

- This is typically determined by how the input hypergraph is expressed.
  1. On **star**-expanded graph
  2. On **clique**-expanded graph



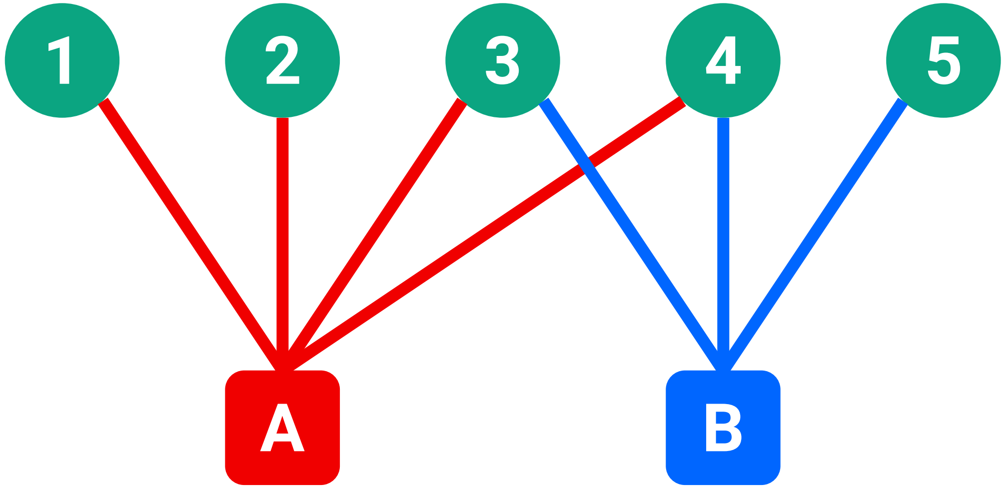
Star-expanded graph



Clique-expanded graph

Q1) Whose messages to aggregate

- This is typically determined by how the input hypergraph is expressed.
  1. On **star**-expanded graph
  2. On **clique**-expanded graph

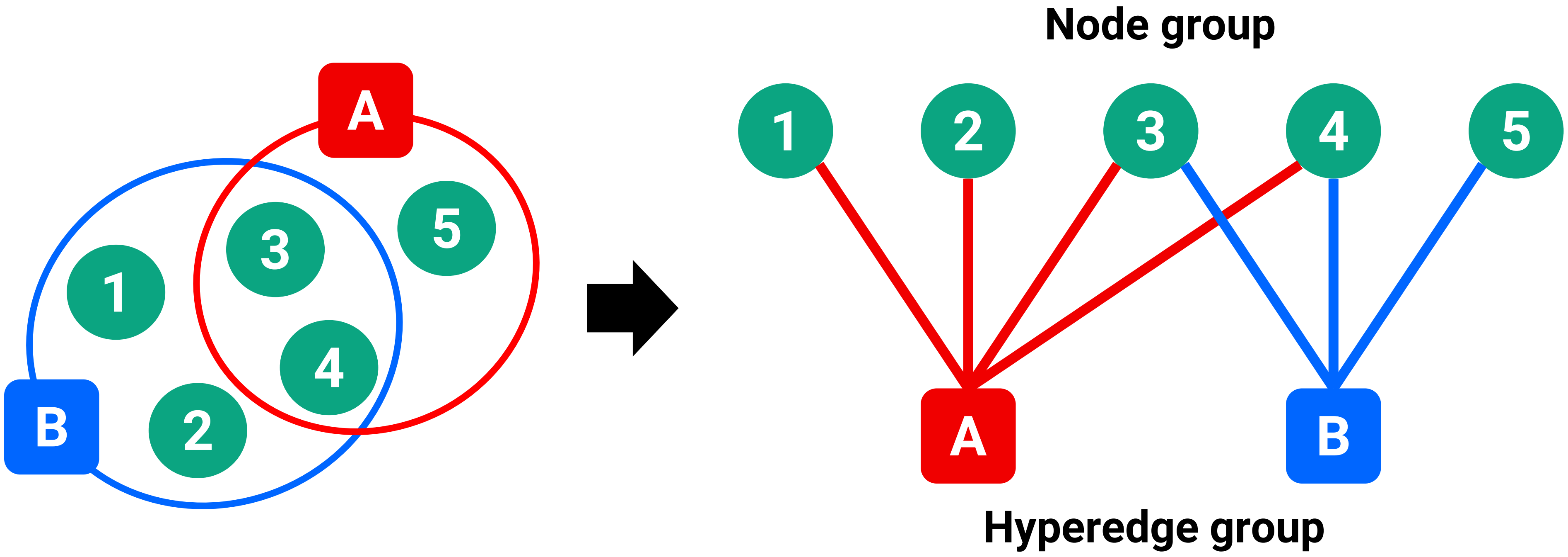


Star-expanded graph

Clique-expanded graph

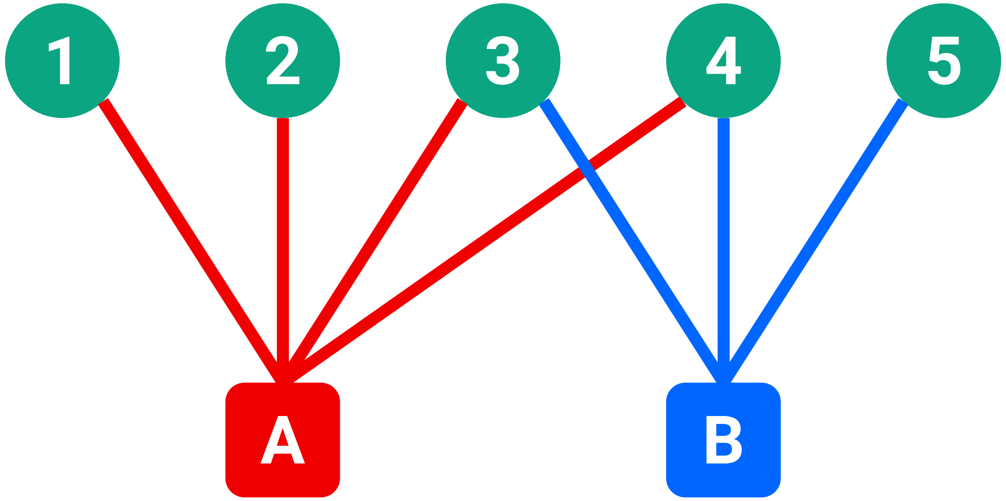
Message Target: On Star-expanded Graphs

- The star-expansion transforms a hypergraph into a bipartite graph.
  - Two groups of nodes: **Node group** and **Hyperedge group**.

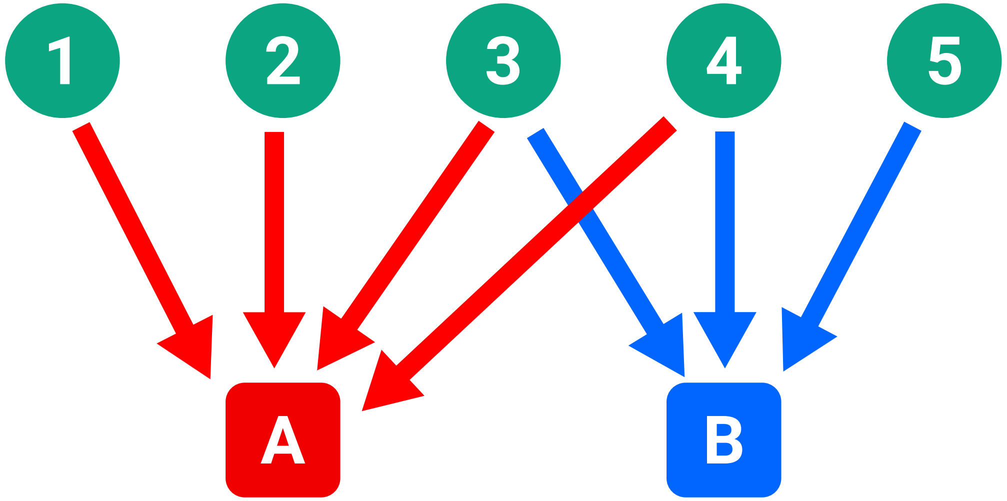


Message Target: On Star-expanded Graphs (cont.)

- On star-expanded graphs, HNNs typically do **two-stage message passing**.
  - **[$\mathcal{V} \rightarrow \mathcal{E}$]** From the node group to the hyperedge group.
  - [$\mathcal{E} \rightarrow \mathcal{V}$] From the hyperedge group to the node group.



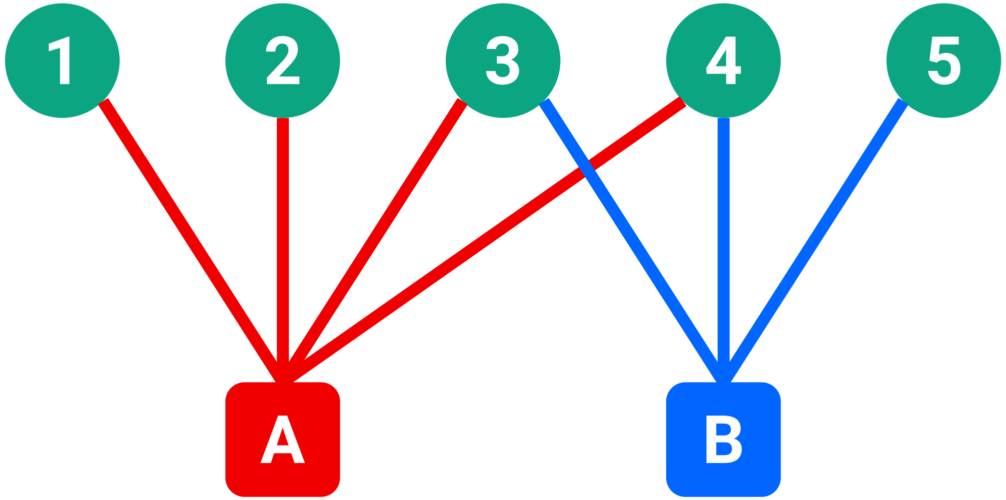
Star-expanded graph



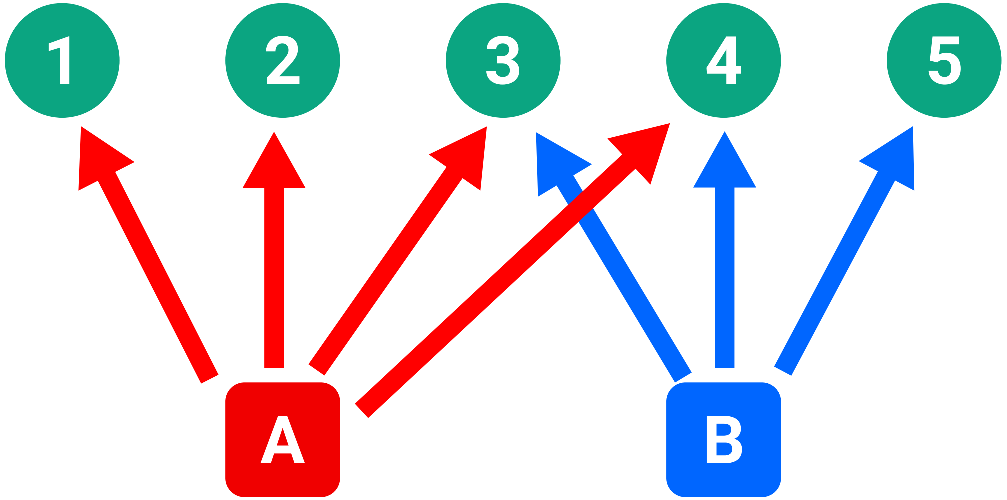
Message passing

Message Target: On Star-expanded Graphs (cont.)

- On star-expanded graphs, HNNs typically do **two-stage message passing**.
  - [$\mathcal{V} \rightarrow \mathcal{E}$] From the node group to the hyperedge group.
  - [$\mathcal{E} \rightarrow \mathcal{V}$] From the hyperedge group to the node group.



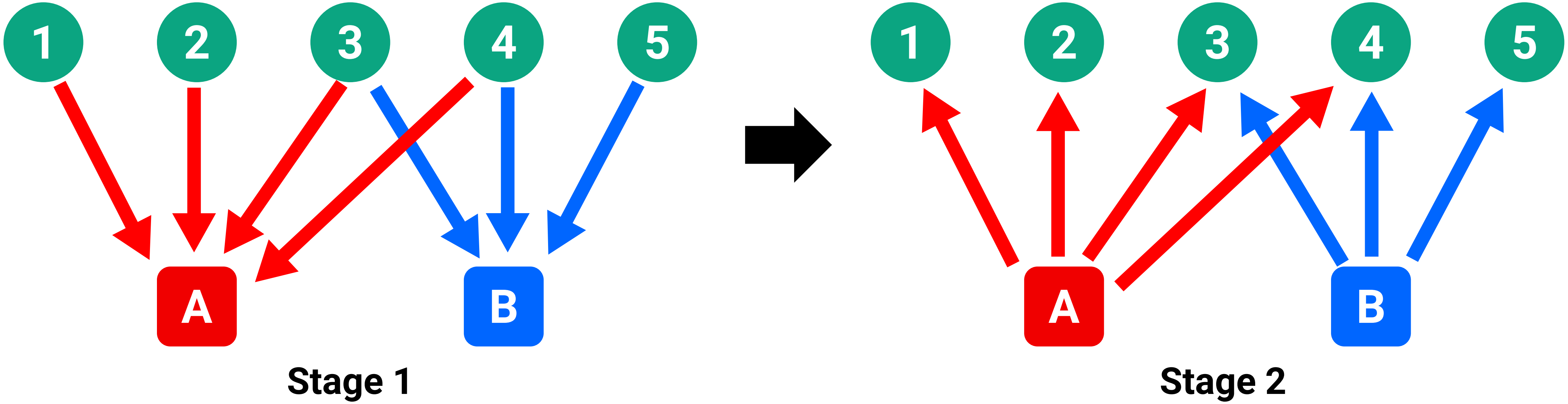
Star-expanded graph



Message passing

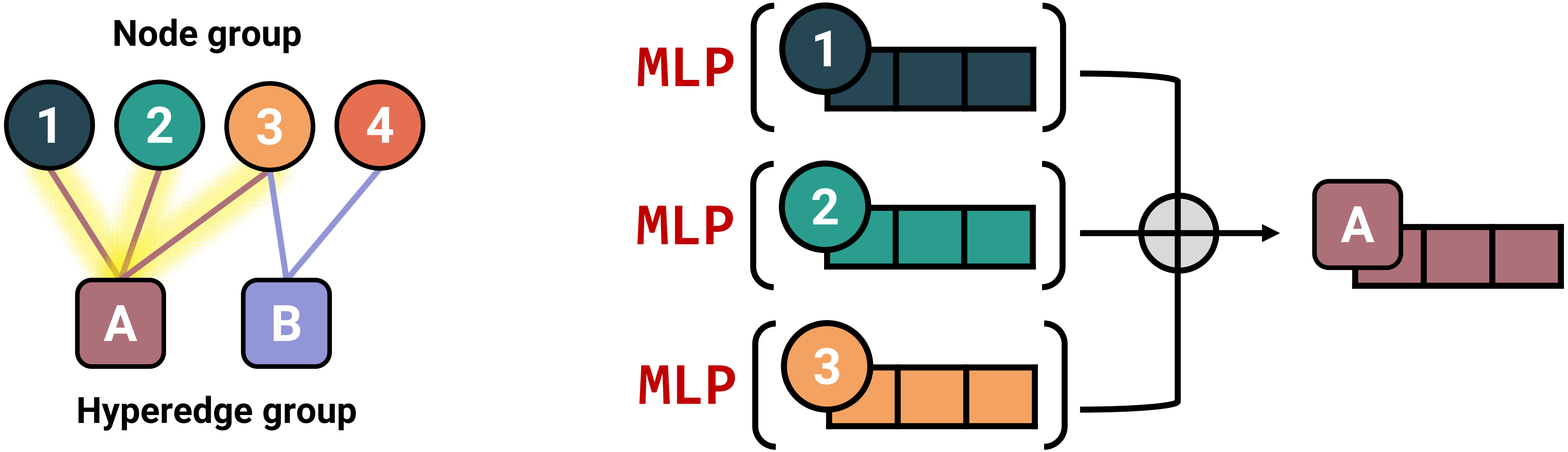
Message Target: On Star-expanded Graphs (cont.)

- This two-stage message passing can be either **sequential** or simultaneous.
  - **[Sequential]** The process of "$[\mathcal{V} \rightarrow \mathcal{E}]$ and $[\mathcal{E} \rightarrow \mathcal{V}]$" is done **in an order**.



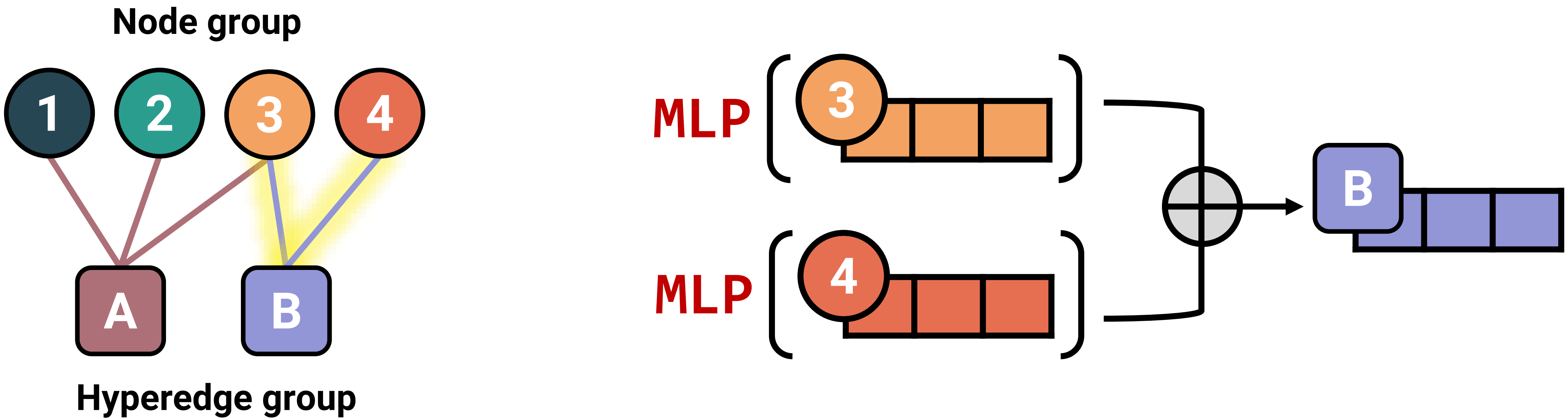
Message Target: On Star-expanded Graphs (cont.)

- A representative example: **ED-HNN** [Wang et al., 2023].
  - **[Stage 1]** Generate hyperedge embeddings by aggregating the embeddings of the constituent nodes $[\mathcal{V} \to \mathcal{E}]$.



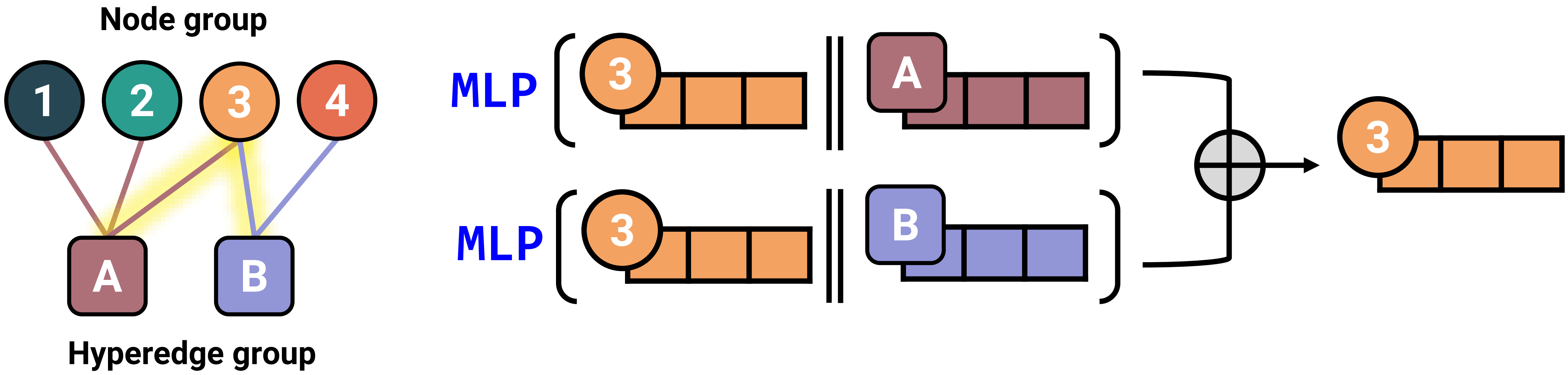
Message Target: On Star-expanded Graphs (cont.)

- A representative example: **ED-HNN** [Wang et al., 2023].
  - **[Stage 1]** Generate hyperedge embeddings by aggregating the embeddings of the constituent nodes $[\mathcal{V} \rightarrow \mathcal{E}]$.



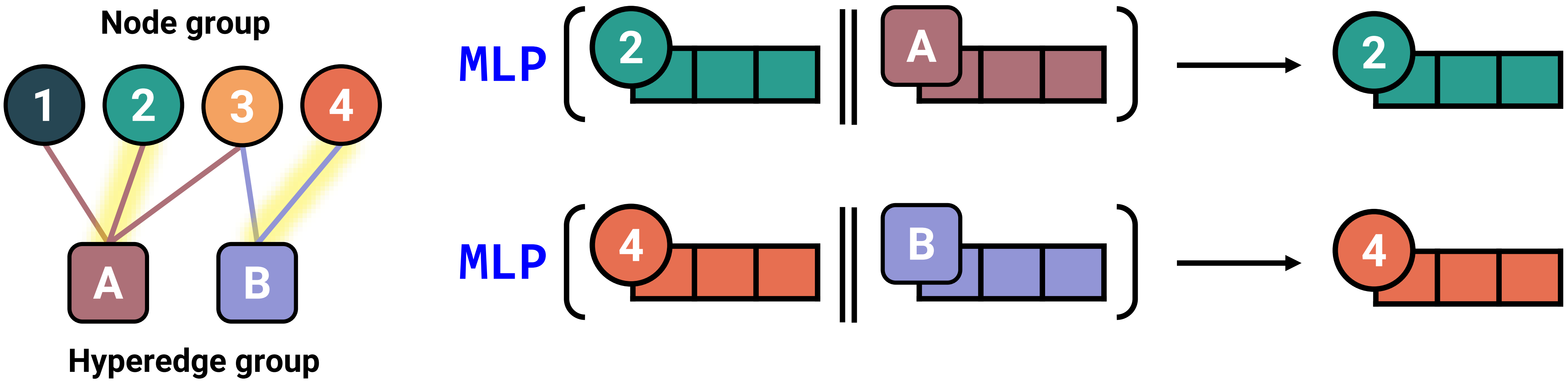
Message Target: On Star-expanded Graphs (cont.)

- A representative example: **ED-HNN** [Wang et al., 2023].
  - **[Stage 2]** Generate node embeddings by aggregating the embeddings of the adjacent hyperedges [$\mathcal{E} \rightarrow \mathcal{V}$].



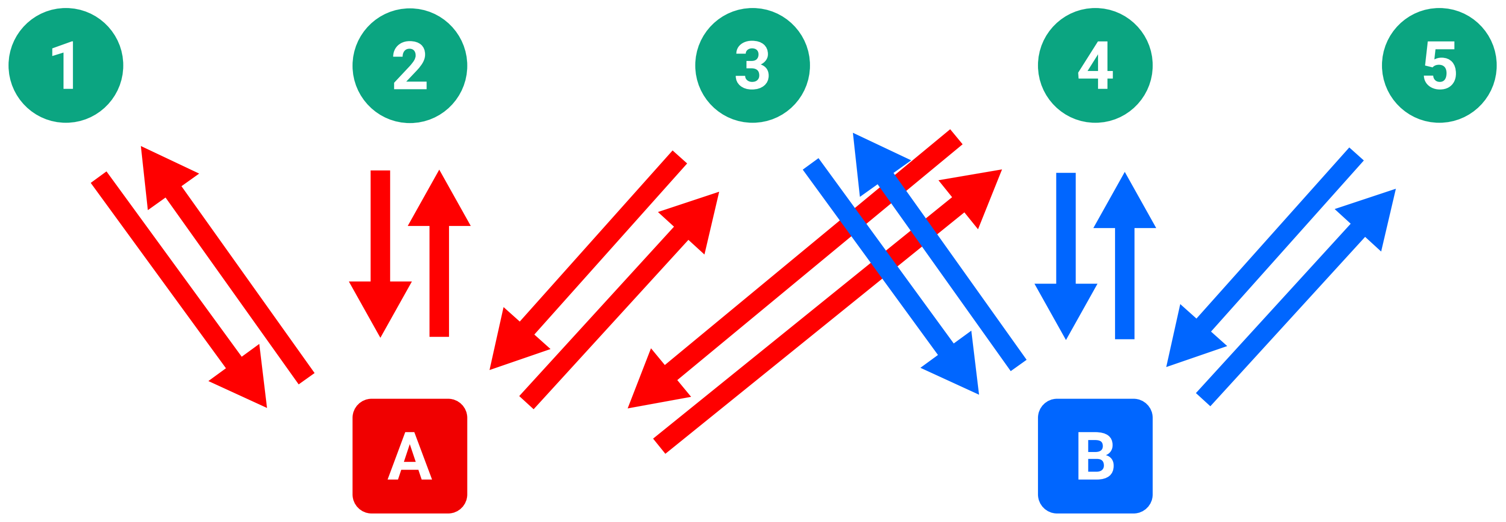
Message Target: On Star-expanded Graphs (cont.)

- A representative example: **ED-HNN** [Wang et al., 2023].
  - **[Stage 2]** Generate node embeddings by aggregating the embeddings of the adjacent hyperedges $[\mathcal{E} \rightarrow \mathcal{V}]$.



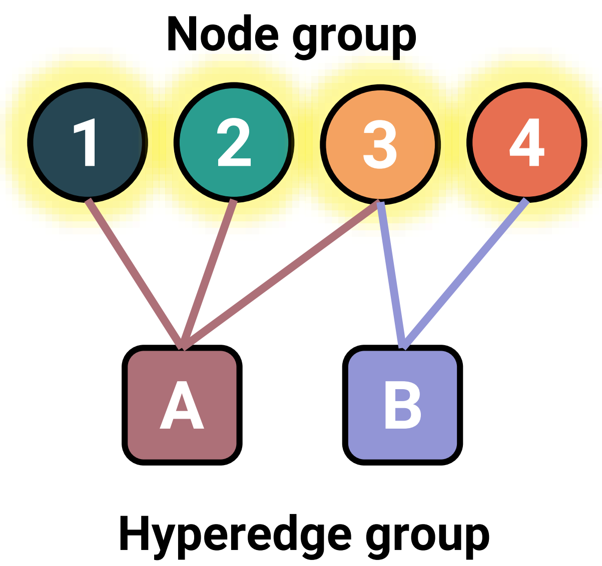
Message Target: On Star-expanded Graphs (cont.)

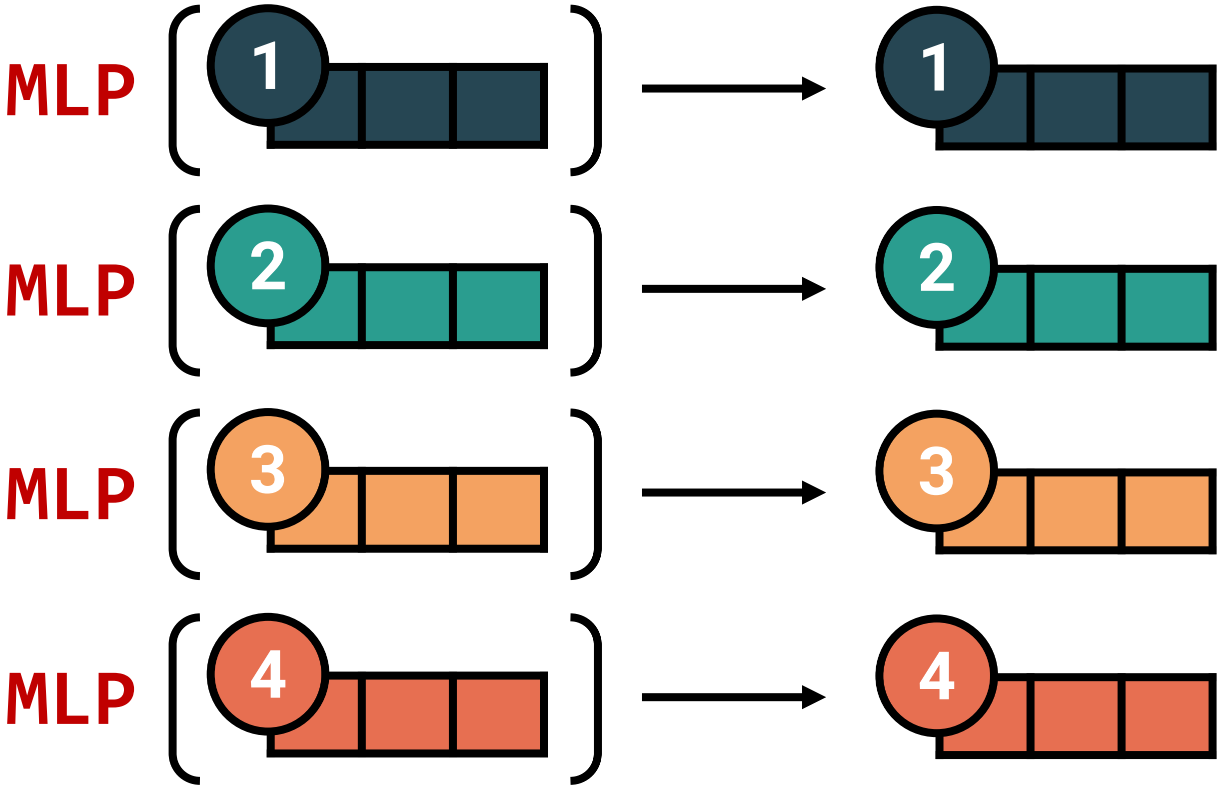
- This two-stage message passing can be either sequential or **simultaneous**.
  - **[Simultaneous]** The process of “$[\mathcal{V} \rightarrow \mathcal{E}]$ and $[\mathcal{E} \rightarrow \mathcal{V}]$” is done **together**.



Message Target: On Star-expanded Graphs (cont.)

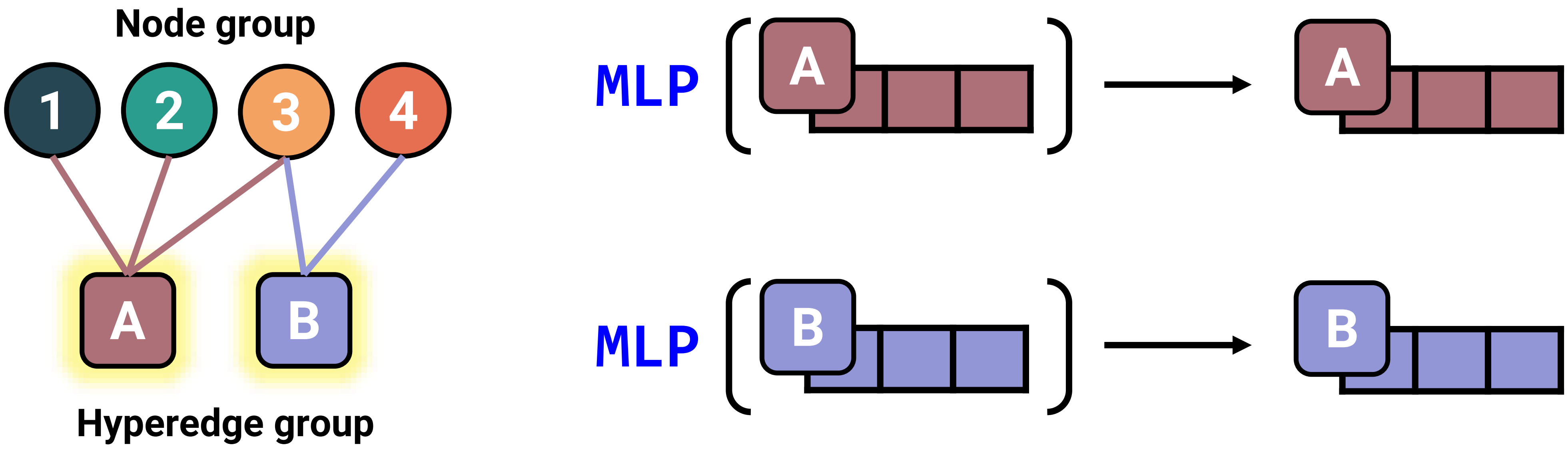
- A representative example: **HDS-ODE** [Yan et al., 2024].
  - **[Stage 1]** Encode **node** and hyperedge embeddings.





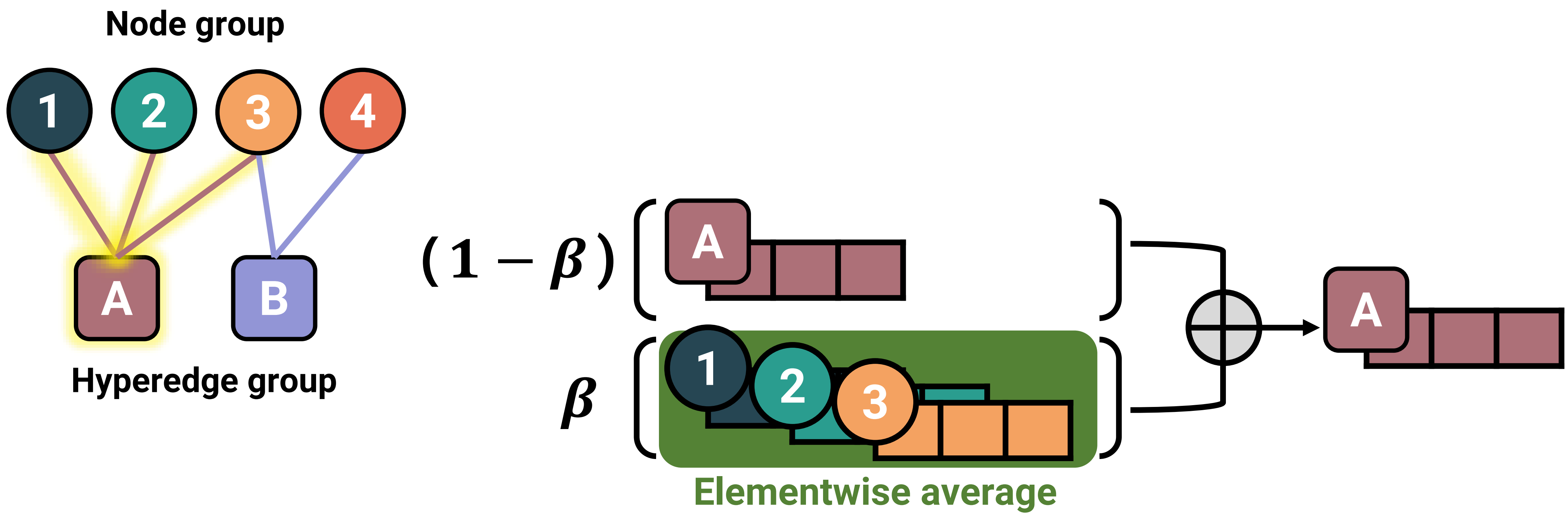
Message Target: On Star-expanded Graphs (cont.)

- A representative example: **HDS-ODE** [Yan et al., 2024].
  - **[Stage 1]** Encode node and **hyperedge** embeddings.



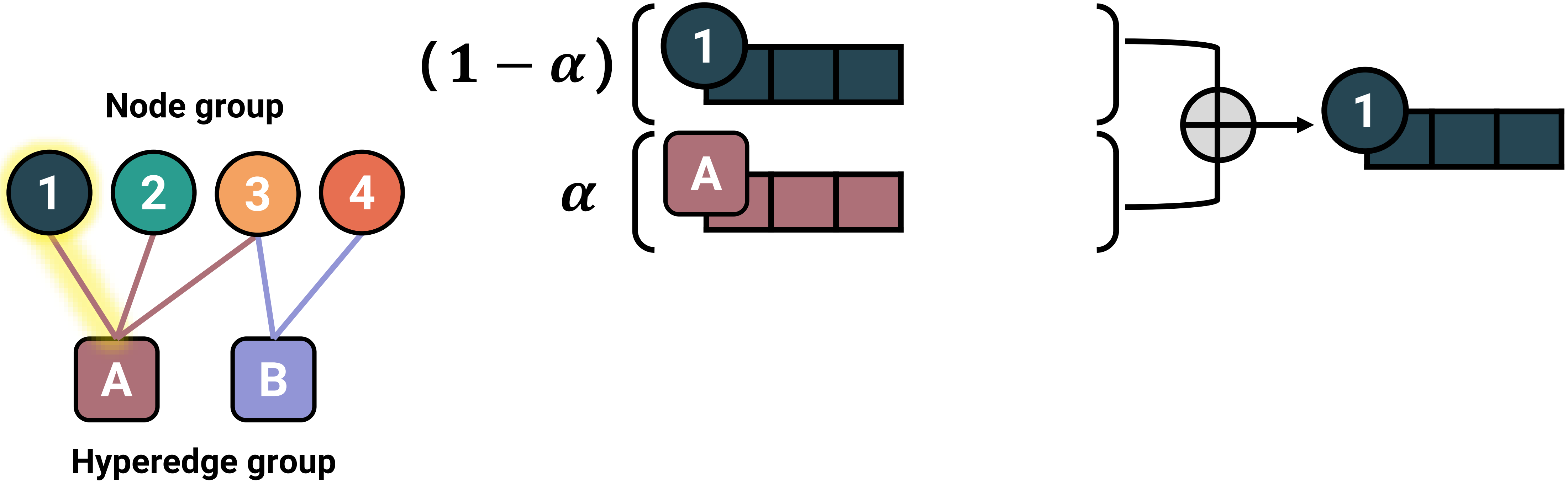
Message Target: On Star-expanded Graphs (cont.)

- A representative example: **HDS-ODE** [Yan et al., 2024].
  - **[Stage 2]** Aggregate adjacent node/**hyperedge** embeddings.



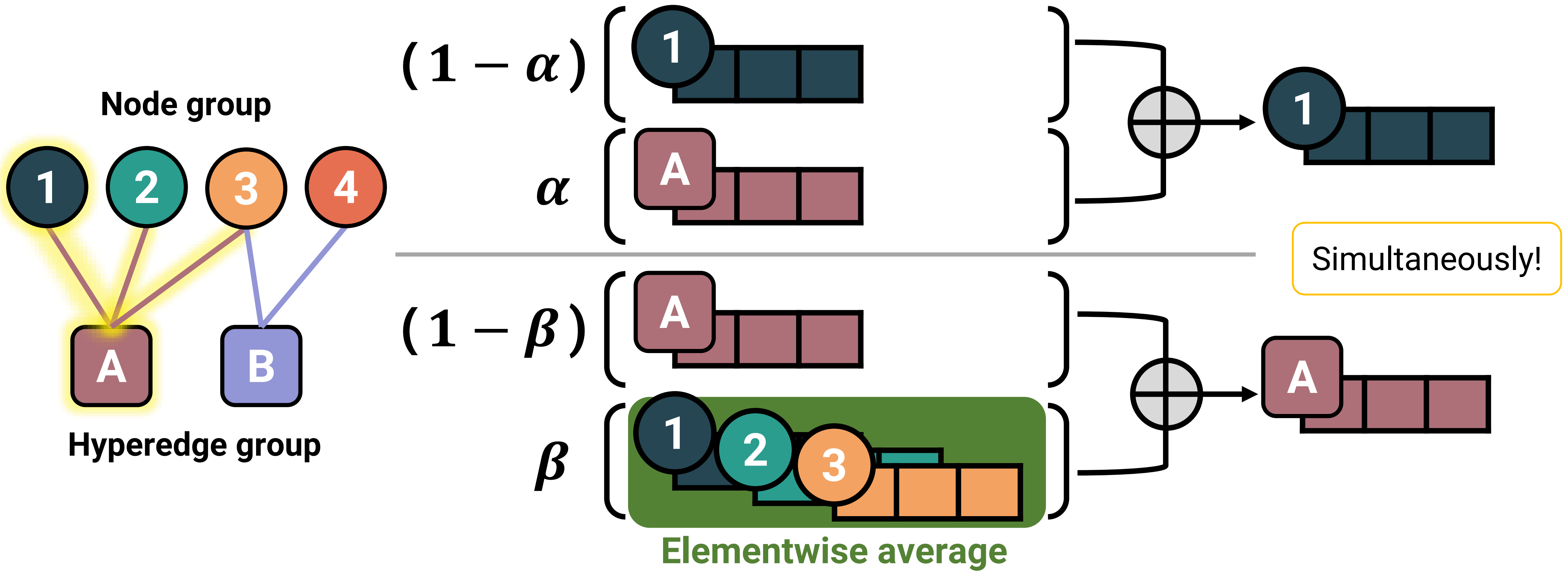
Message Target: On Star-expanded Graphs (cont.)

- A representative example: **HDS-ODE** [Yan et al., 2024].
  - **[Stage 2]** Aggregate adjacent **node**/hyperedge embeddings.



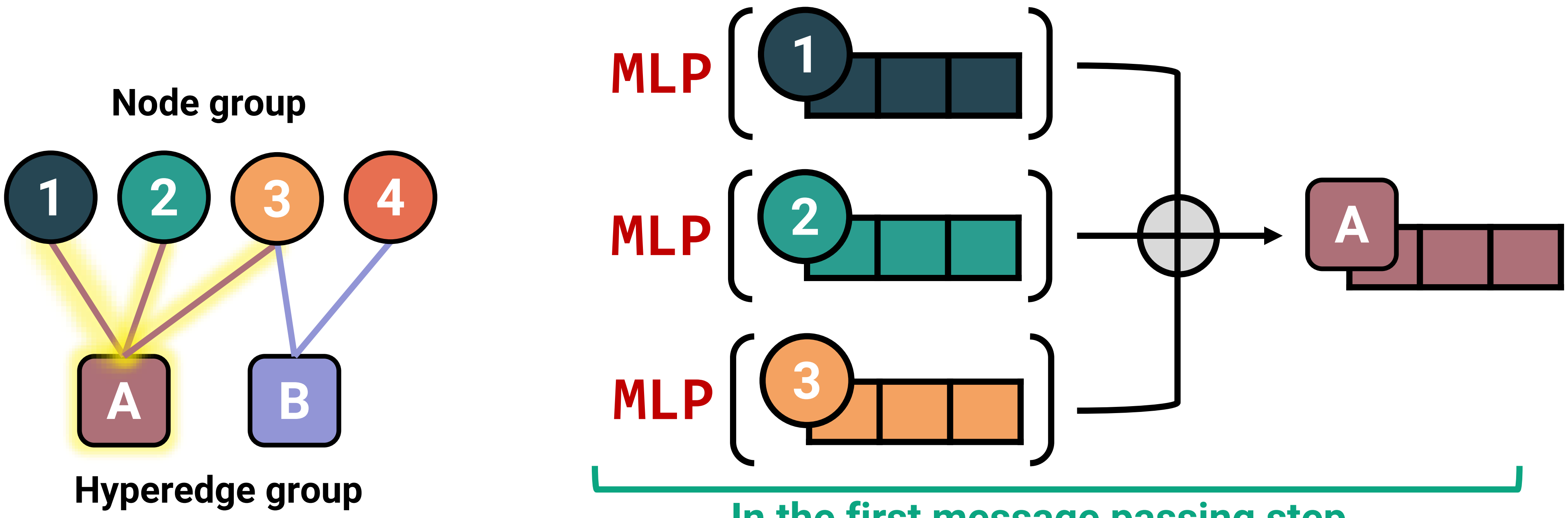
Message Target: On Star-expanded Graphs (cont.)

- A representative example: **HDS-ODE** [Yan et al., 2024].
  - **[Stage 2]** Aggregate adjacent node/hyperedge embeddings.



Sequential Passing vs Simultaneous Passing

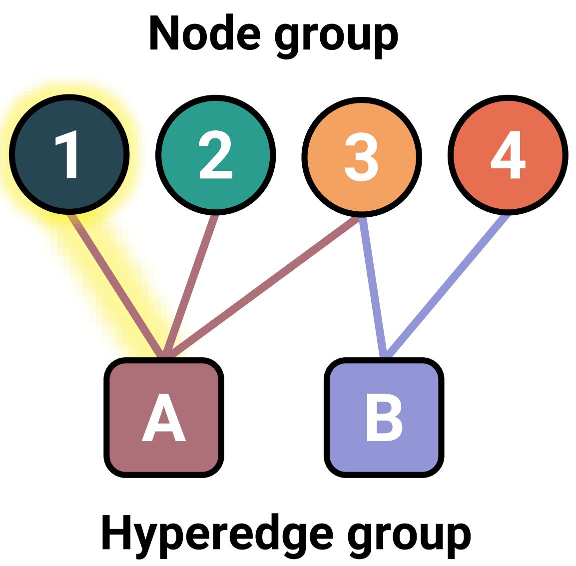
- **[Sequential passing]** When appropriate hyperedge features **are not given**, sequential passing would be spontaneous.

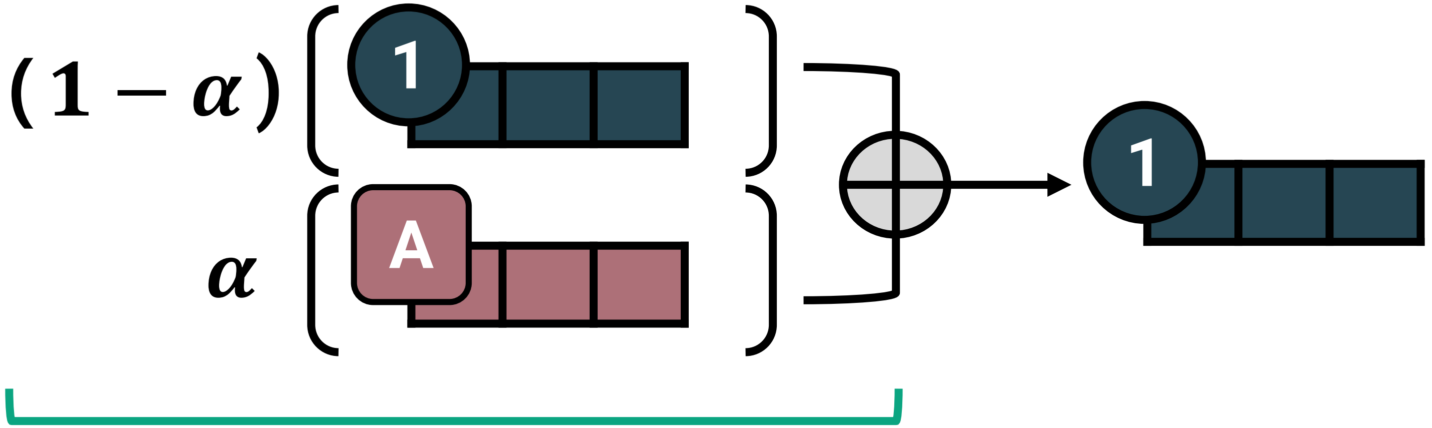


In the first message passing step,
input hyperedge features are not necessary.

Sequential Passing vs Simultaneous Passing (cont.)

- **[Simultaneous passing]** When appropriate hyperedge features **are given**, simultaneous passing would be spontaneous.

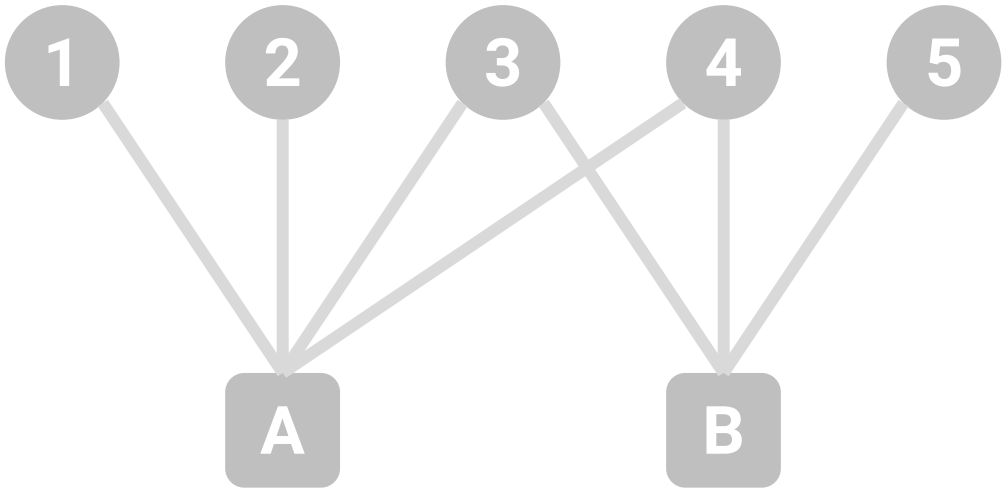




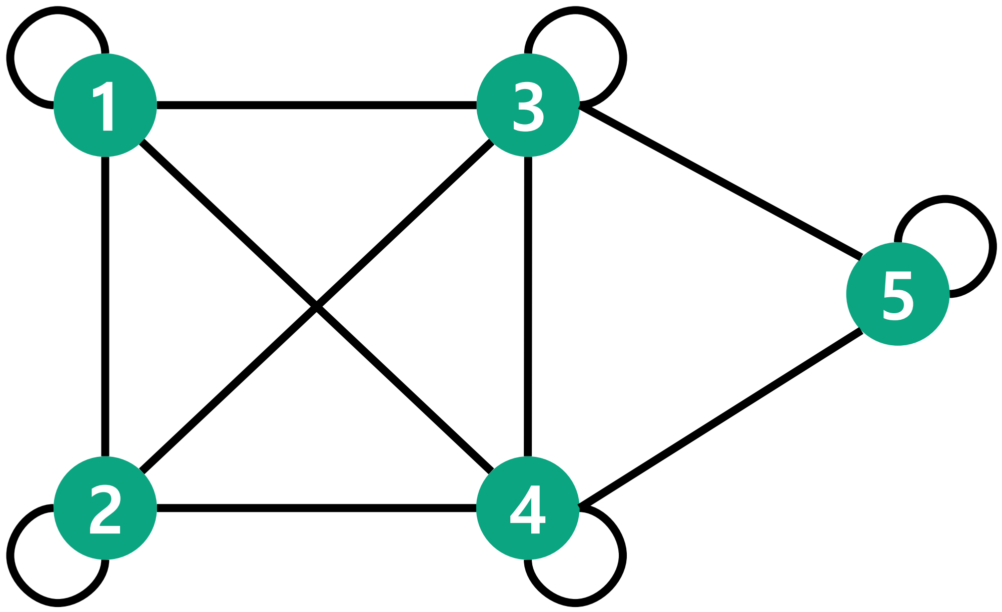
In the first message passing step, input hyperedge features are required.

Q1) Whose messages to aggregate

- This is typically determined by how the input hypergraph is expressed.
  1. On **star**-expanded graph
  2. On **clique**-expanded graph



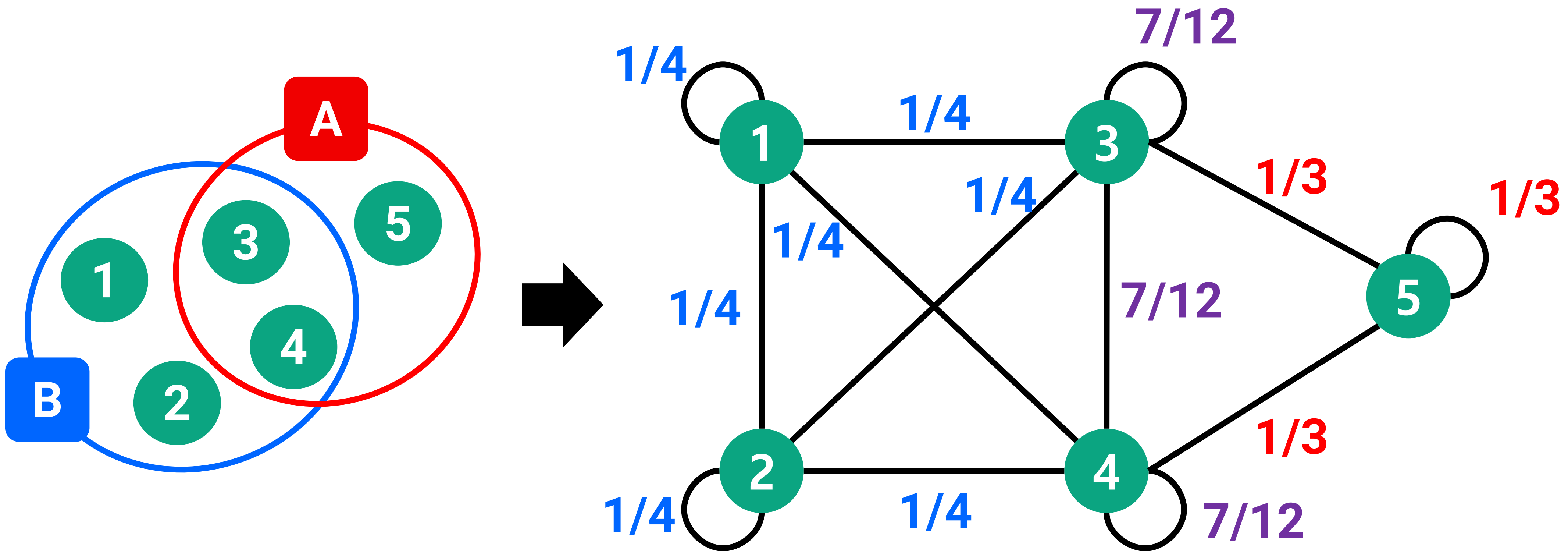
Star-expanded graph



Clique-expanded graph

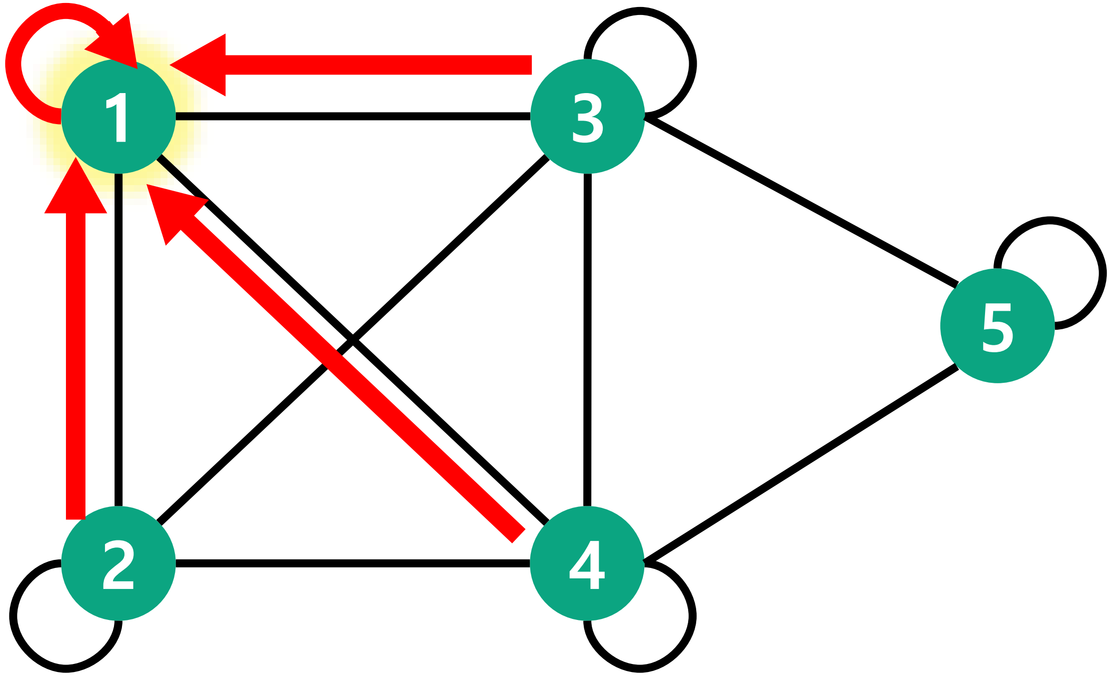
Q1) On Clique-expanded Graphs

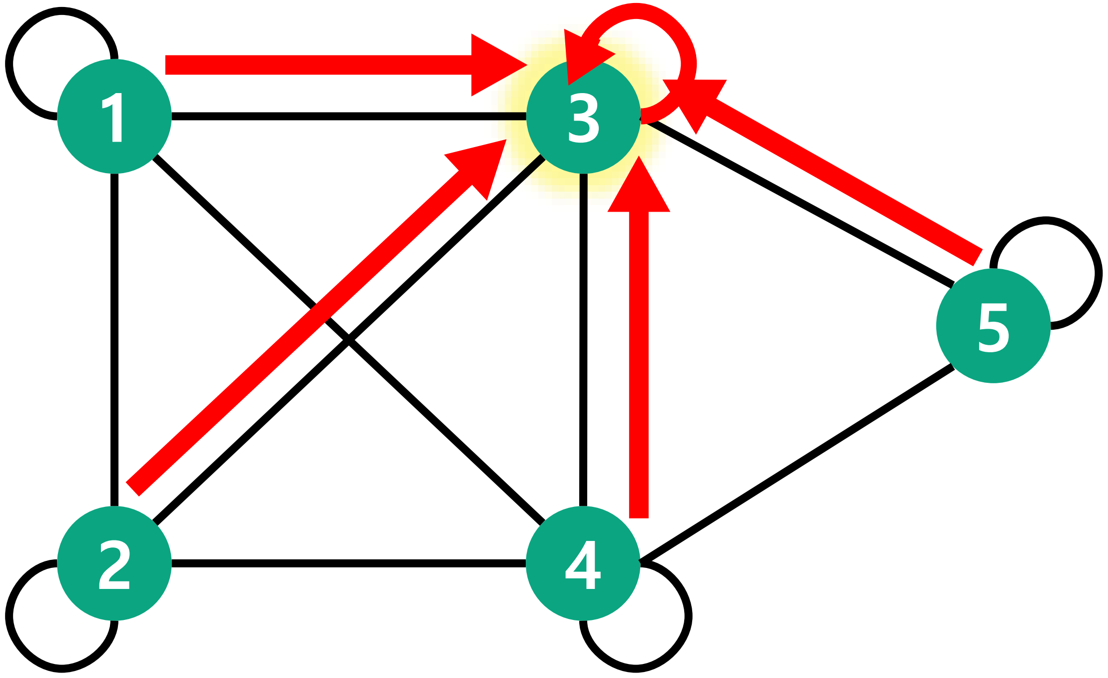
- (Recall) Clique-expansion transforms a hypergraph into a weighted graph.



Message Target: On Clique-expanded Graphs (cont.)

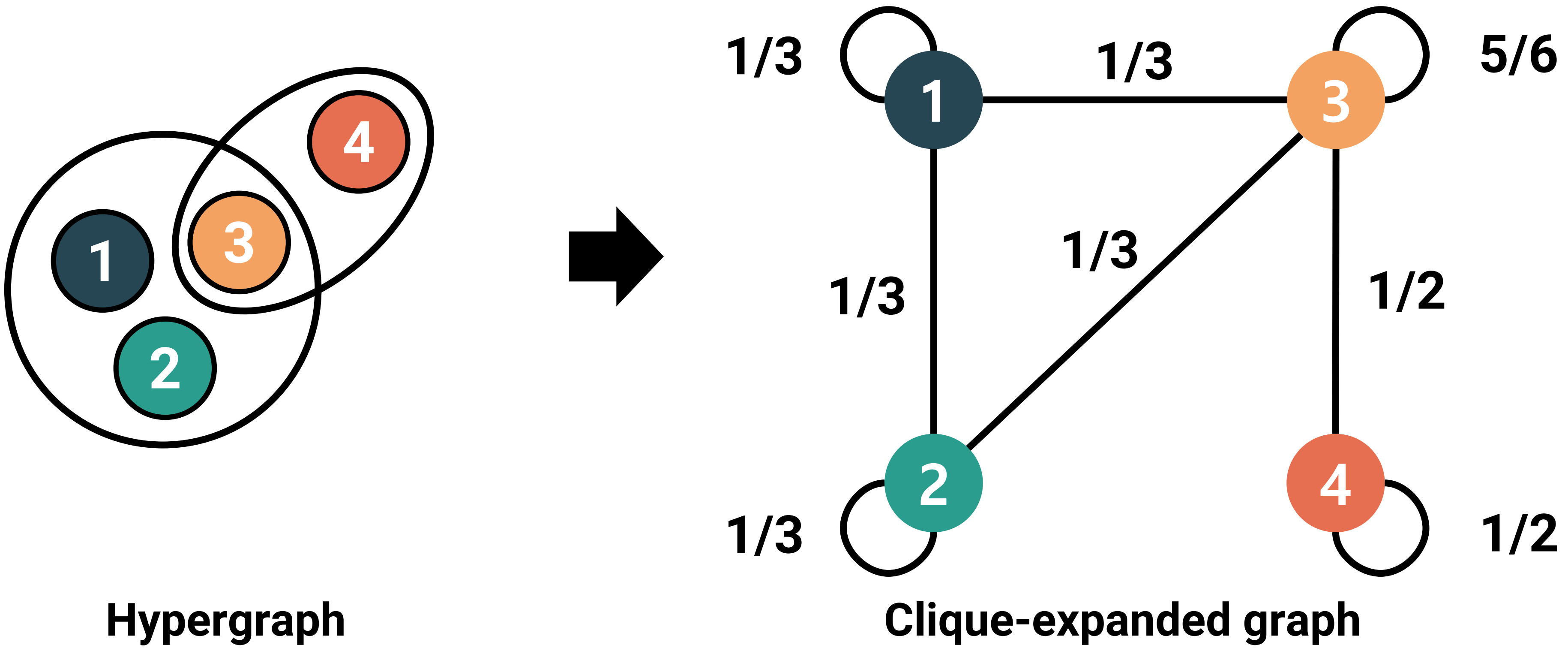
- Akin typical GNNs, HNNs based on clique-expansion perform message passing **between nodes**.





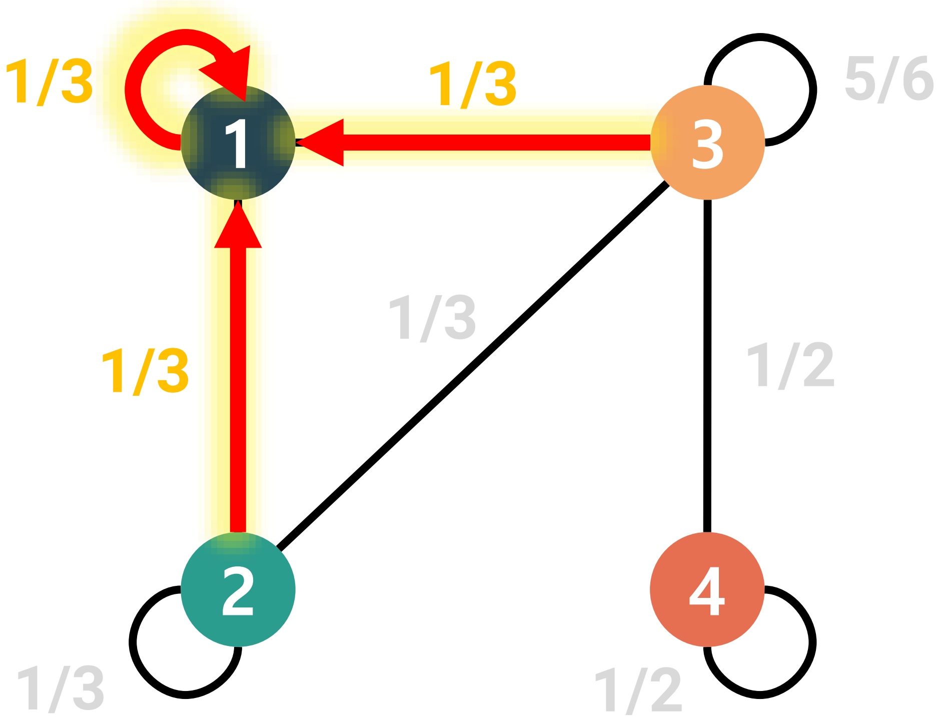
Message Target: On Clique-expanded Graphs (cont.)

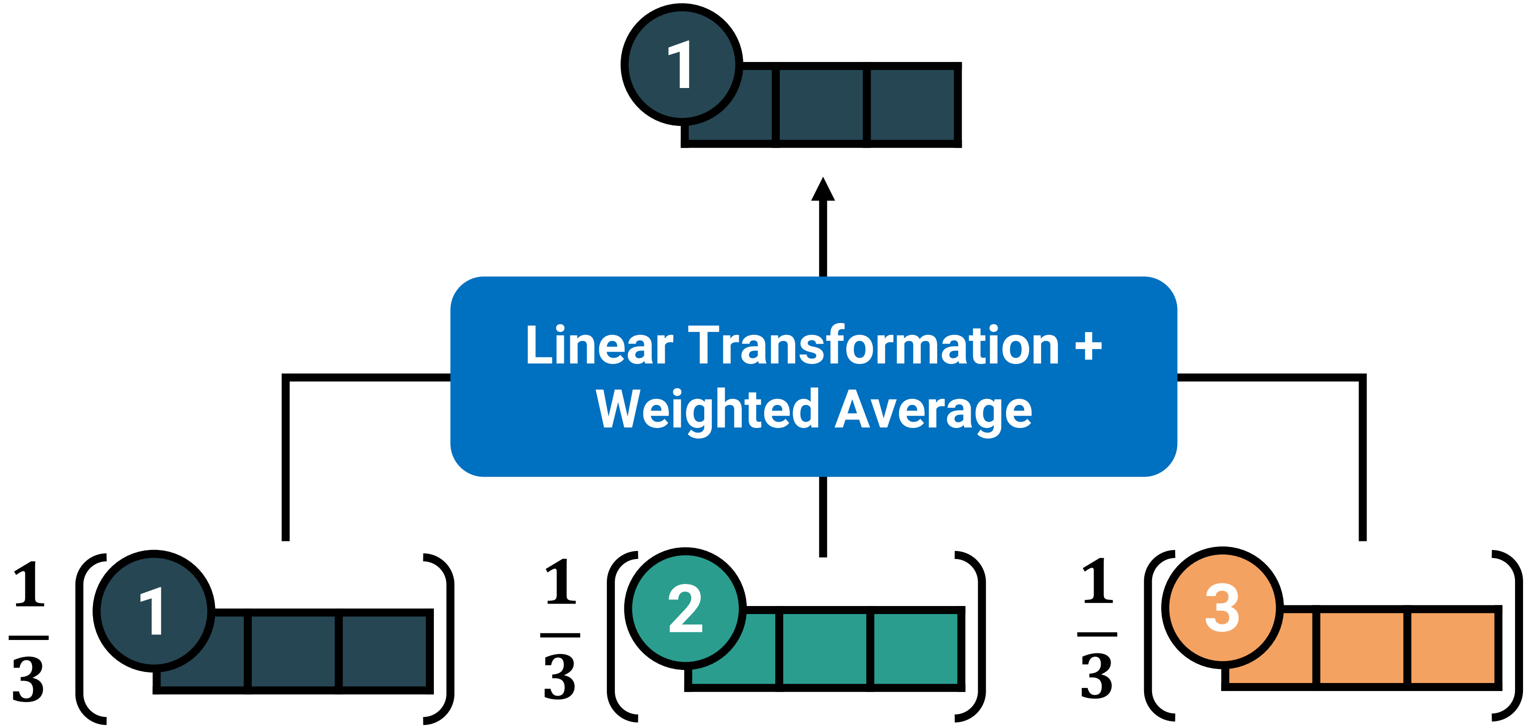
- A representative example: **HGNN** [Feng et al., 2019].
  - **[Stage 1]** Obtain the clique-expanded graph with edge weights.



Message Target: On Clique-expanded Graphs (cont.)

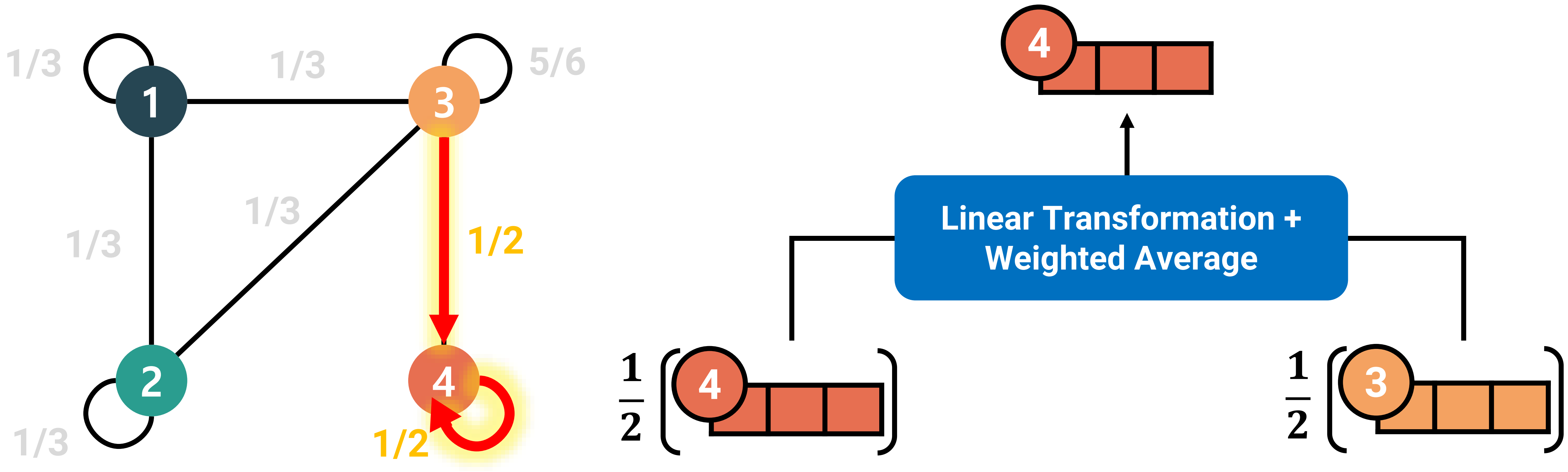
- A representative example: **HGNN** [Feng et al., 2019].
  - **[Stage 2]** Aggregate neighbor embeddings.





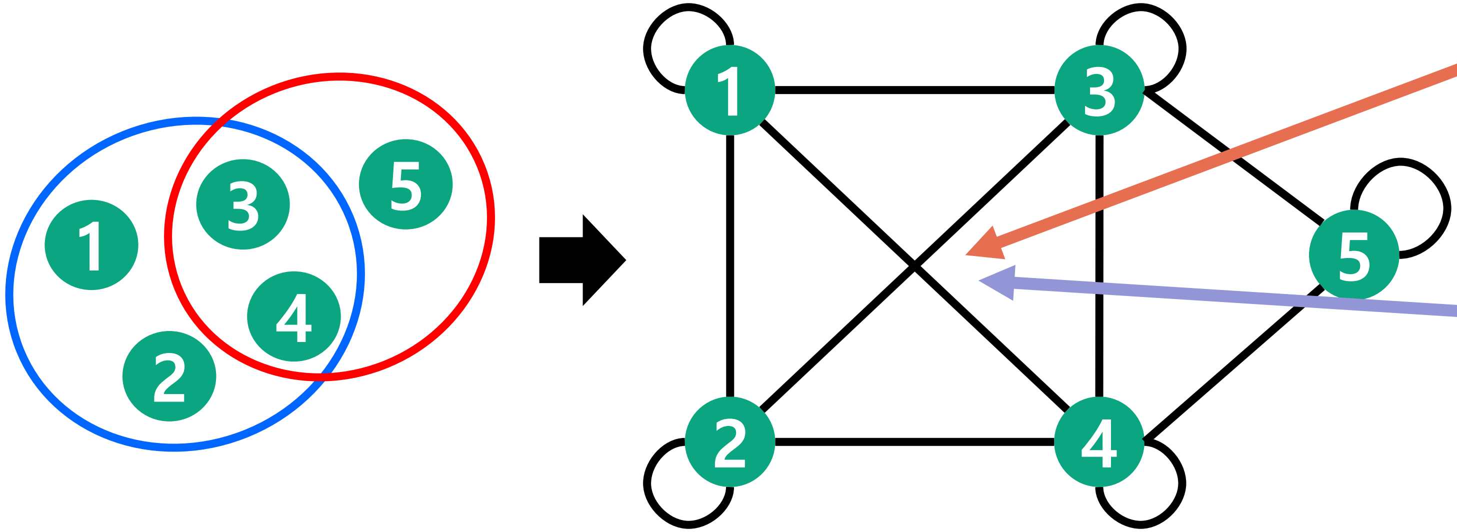
Message Target: On Clique-expanded Graphs (cont.)

- A representative example: **HGNN** [Feng et al., 2019].
  - **[Stage 2]** Aggregate neighbor embeddings.



Clique Expansion vs Star Expansion

- **[Pros of clique expansion]** Various advanced techniques for GNNs can be directly applied/extended.



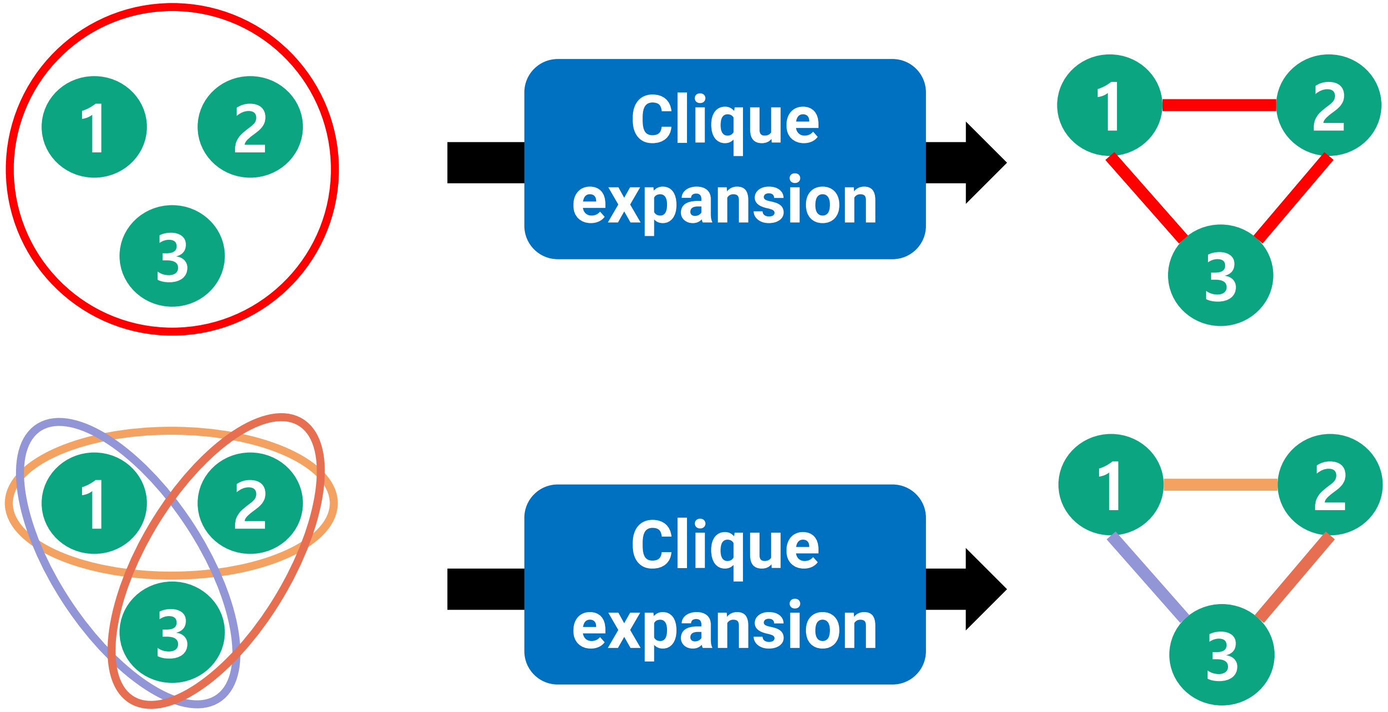
Re-normalization trick [Kipf and Welling, 2017]

Graph attention [Veličković et al., 2018]

•••

Clique Expansion vs Star Expansion (cont.)

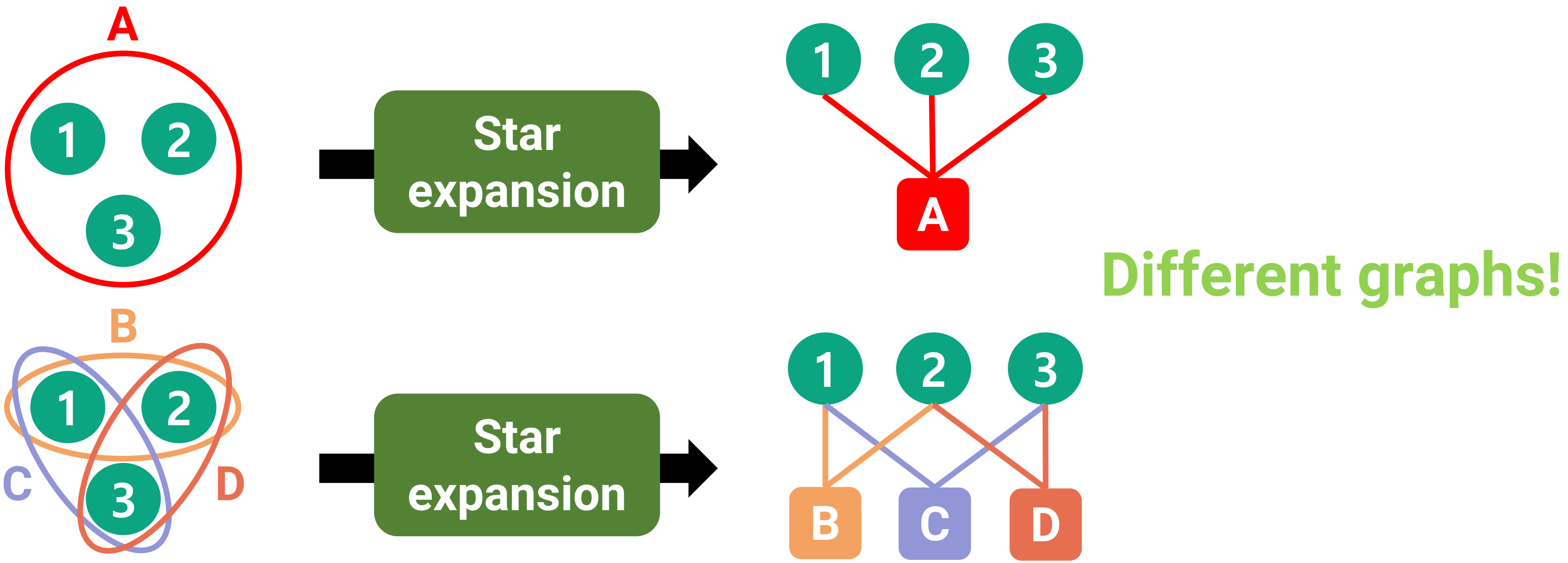
- **[Cons of clique expansion]** Higher-order information can be lost [Zhou et al., 2006].



Same graph!

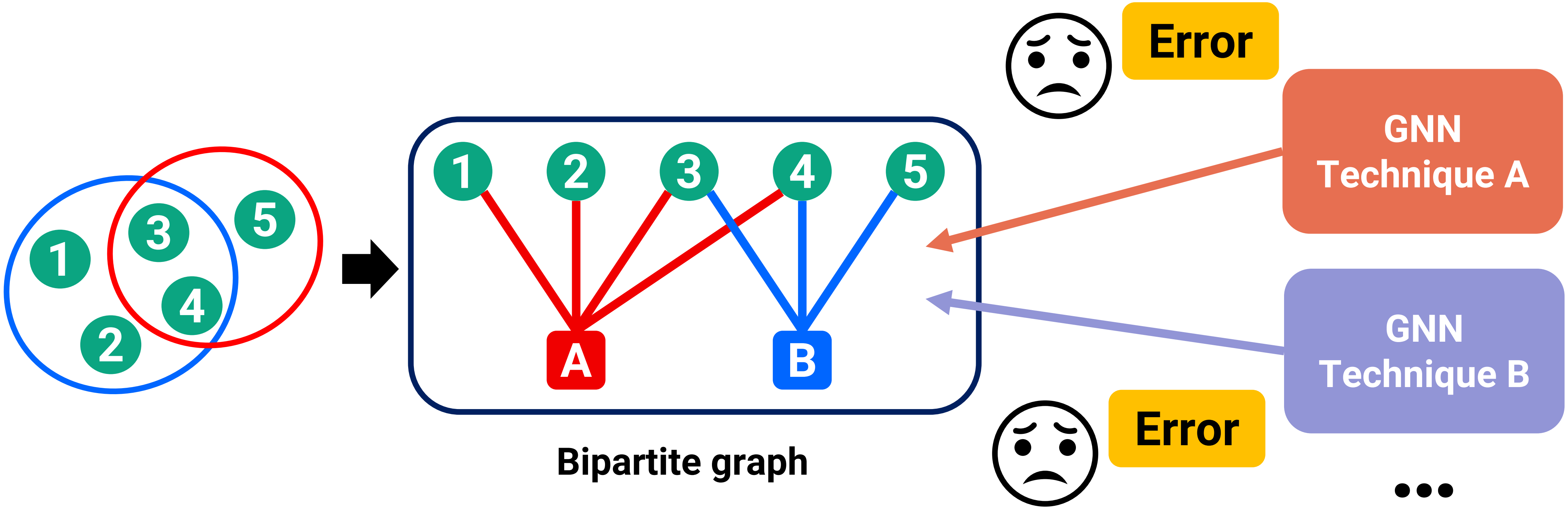
Clique Expansion vs Star Expansion (cont.)

- **[Pros of star expansion]** Can maintain higher-order information.



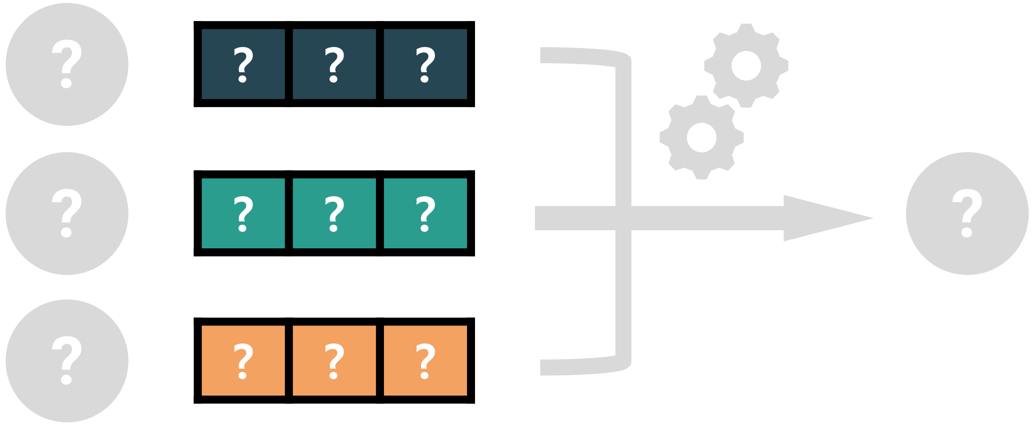
Clique Expansion vs Star Expansion (cont.)

- **[Cons of star expansion]** Direct extension of advanced GNN techniques can be non-trivial.



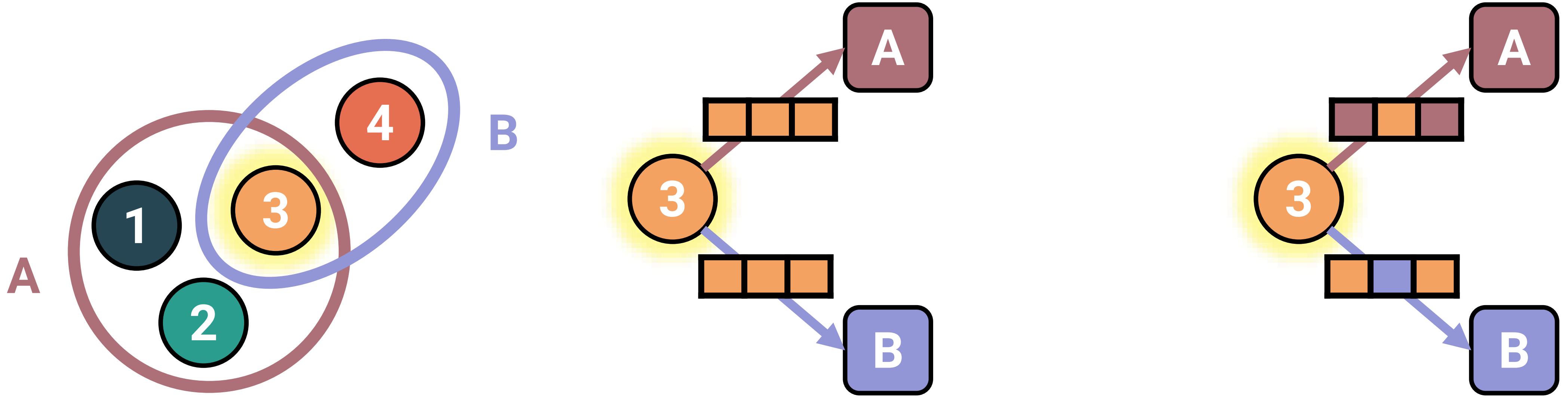
Q2) What messages to aggregate

- In HNNs' message passing, there are three key questions:
  - Q1) **Whose** messages to aggregate
  - Q2) **What** messages to aggregate
  - Q3) **How** to aggregate messages



Q2) What messages to aggregate (cont.)

- Possible message representations:
  1. **Hyperedge-consistent** messages
  2. **Hyperedge-dependent** messages

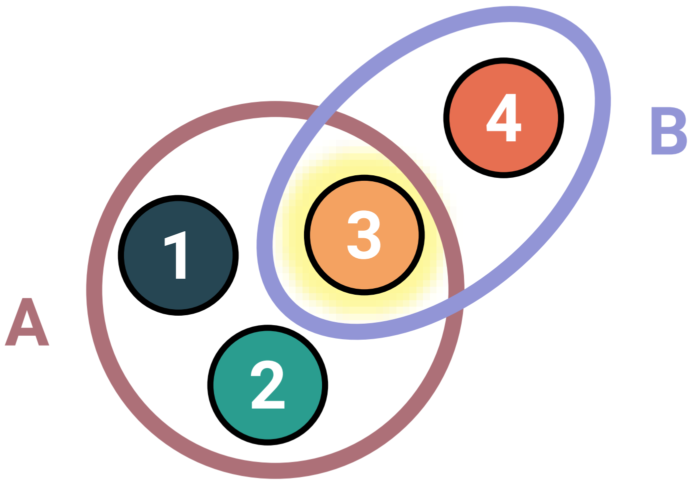


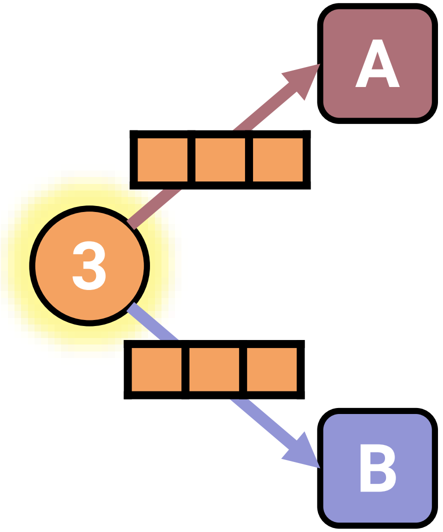
(1) Same messages

(2) Different messages

Q2) What messages to aggregate (cont.)

- Possible message representations:
  1. **Hyperedge-consistent** messages
  2. **Hyperedge-dependent** messages



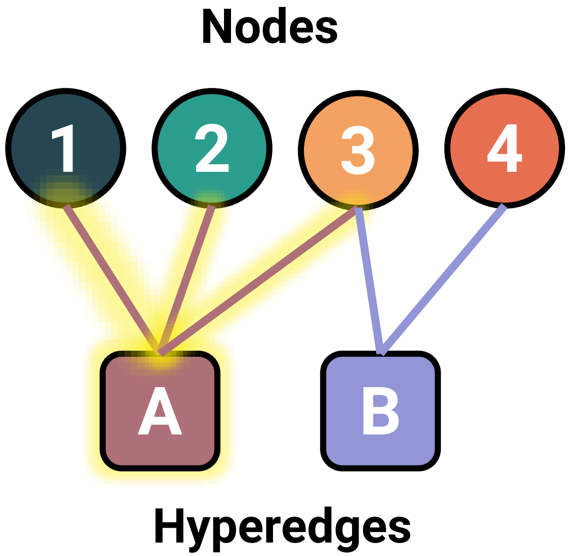


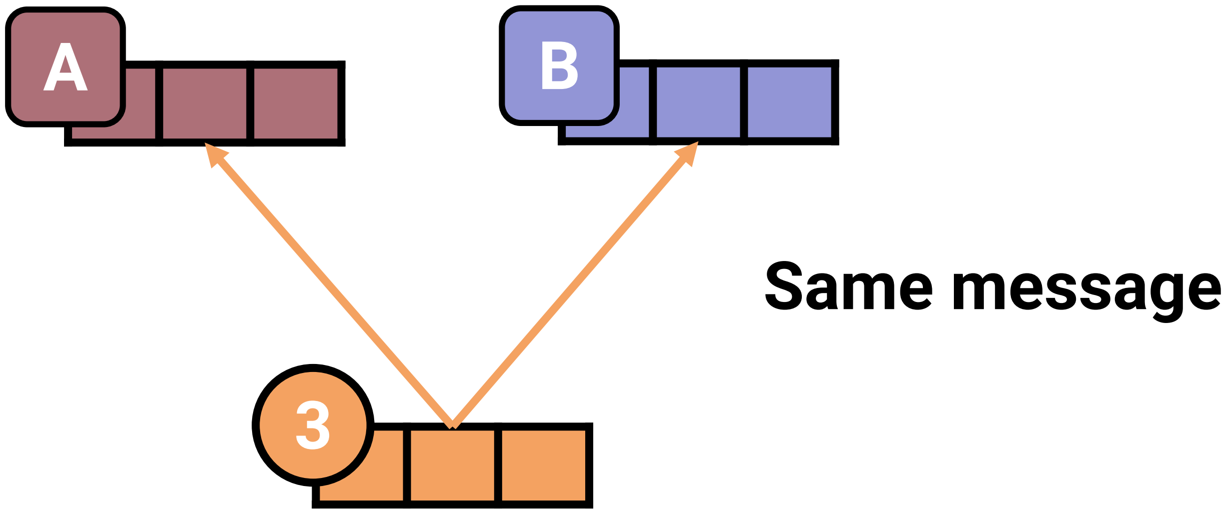
(1) Same messages

(2) Different messages

Q2) Hyperedge-Consistent Messages

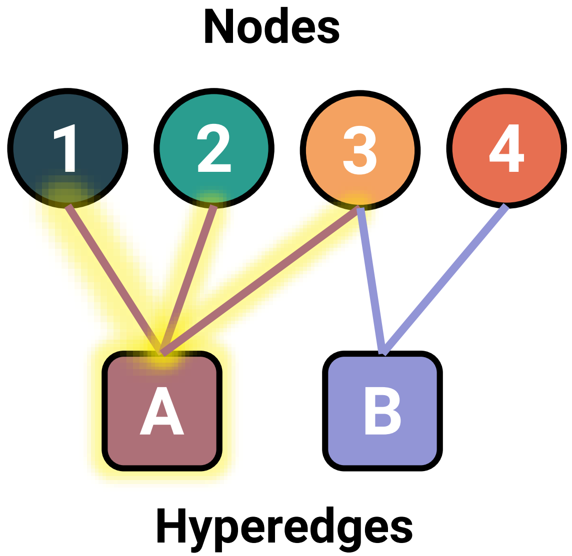
- In many HNNs, the node representation **remains the same** across all aggregations.

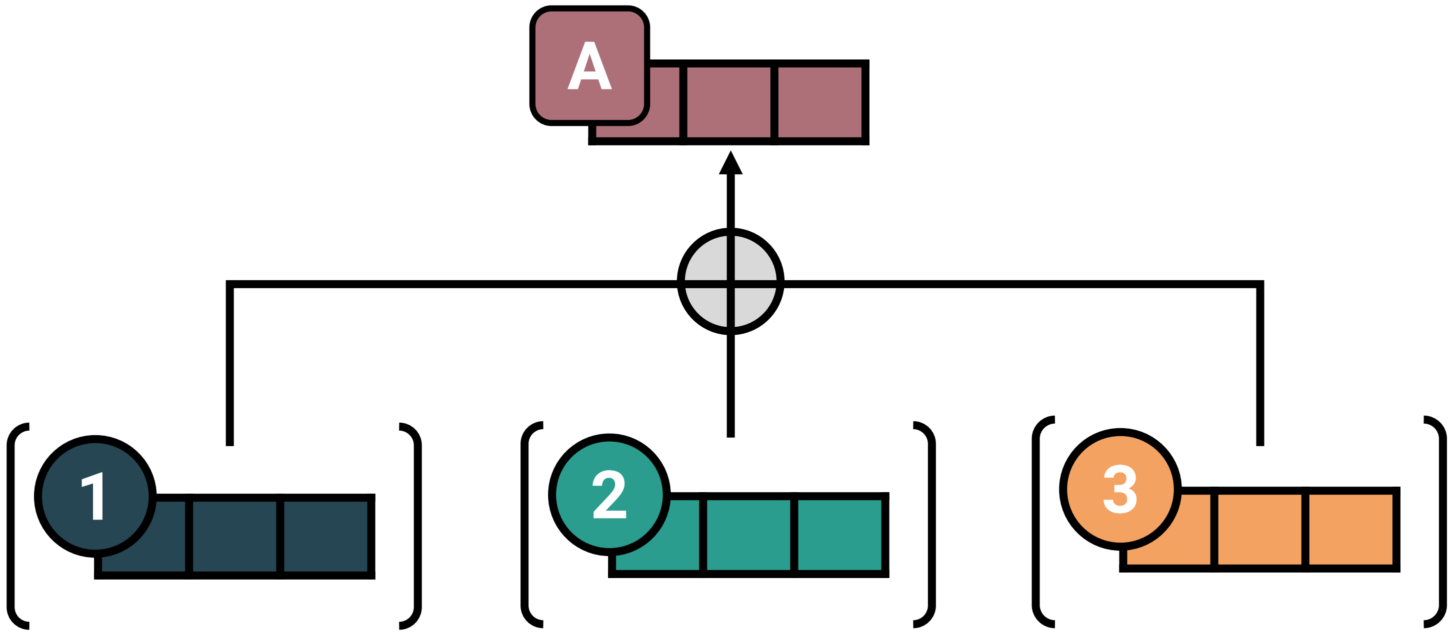




Message Vector: Hyperedge-Consistent

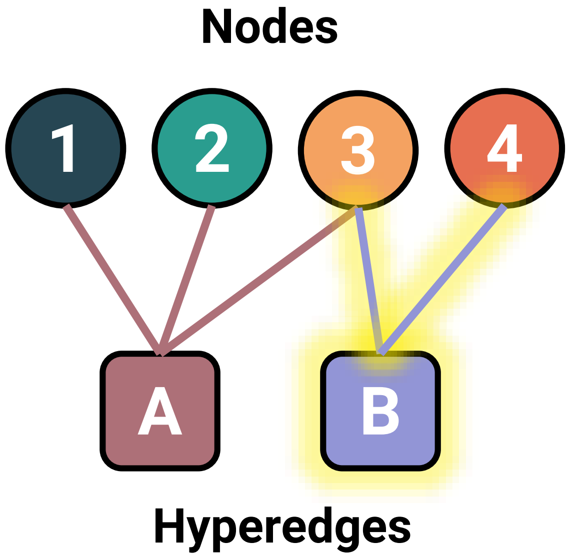
- In many HNNs, the node representation **remains the same** across all aggregations.
- A representative example is **UniGIN** [Huang et al., 2021].

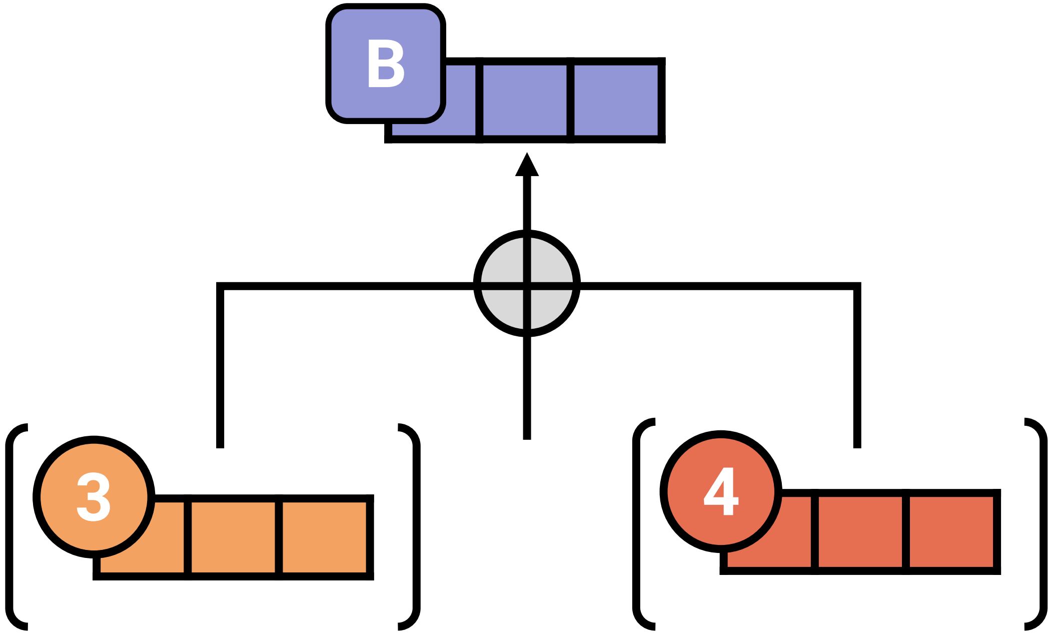




Message Vector: Hyperedge-Consistent (cont.)

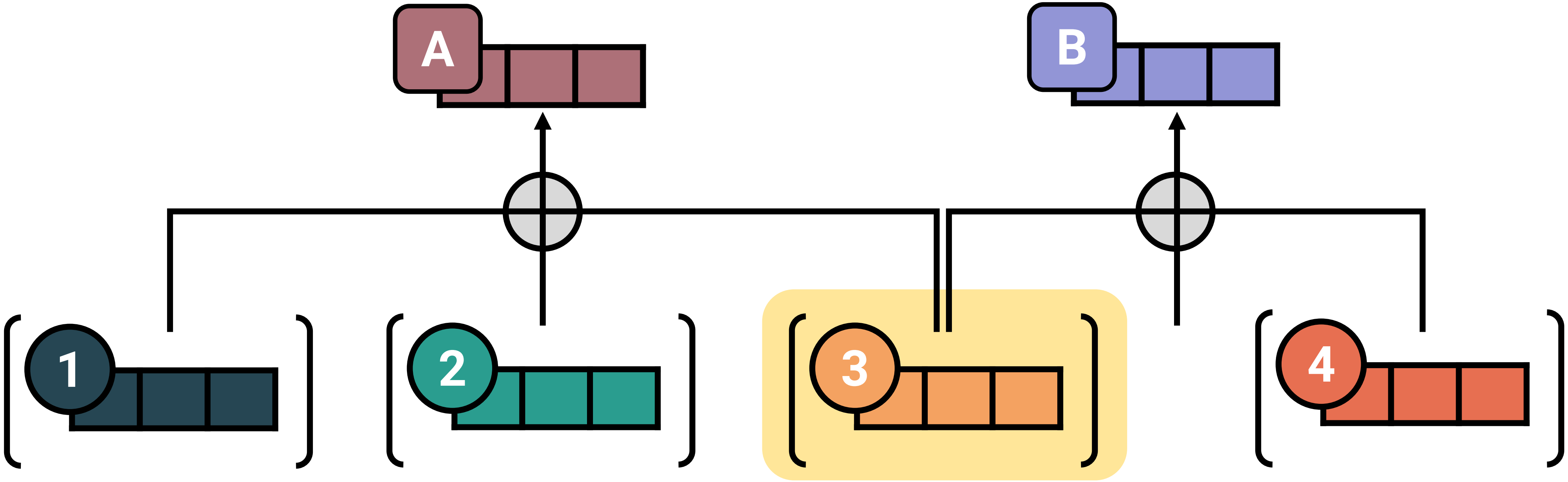
- In many HNNs, the node representation **remains the same** across all aggregations.
- A representative example is **UniGIN** [Huang et al., 2021].





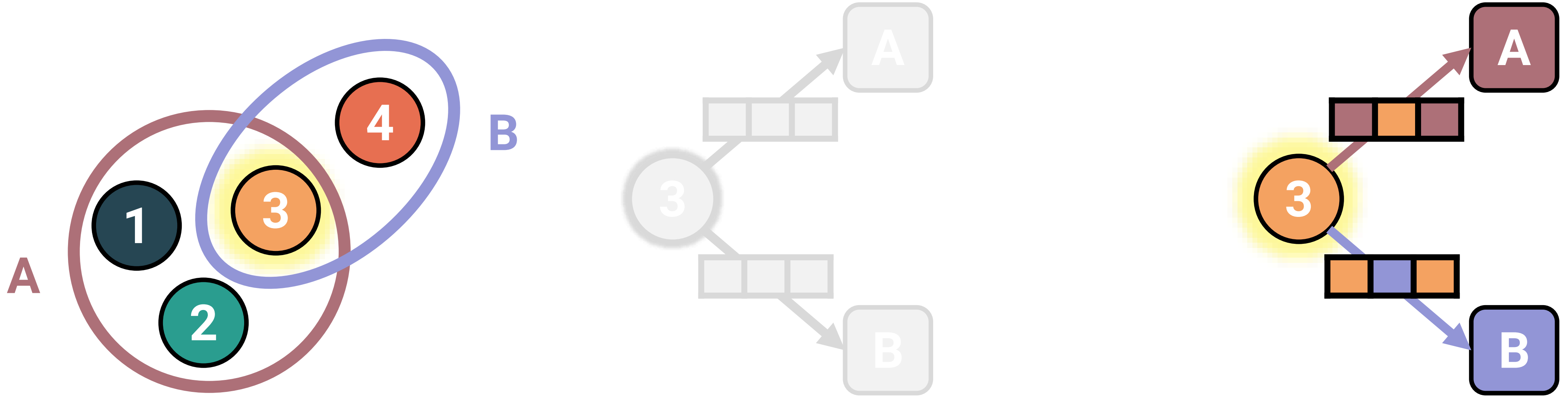
Message Vector: Hyperedge-Consistent (cont.)

- In many HNNs, the node representation **remains the same** across all aggregations.
- A representative example is **UniGIN** [Huang et al., 2021].



Q2) What messages to aggregate (cont.)

- Possible message representations:
  - Hyperedge-consistent messages
  - **Hyperedge-dependent** messages

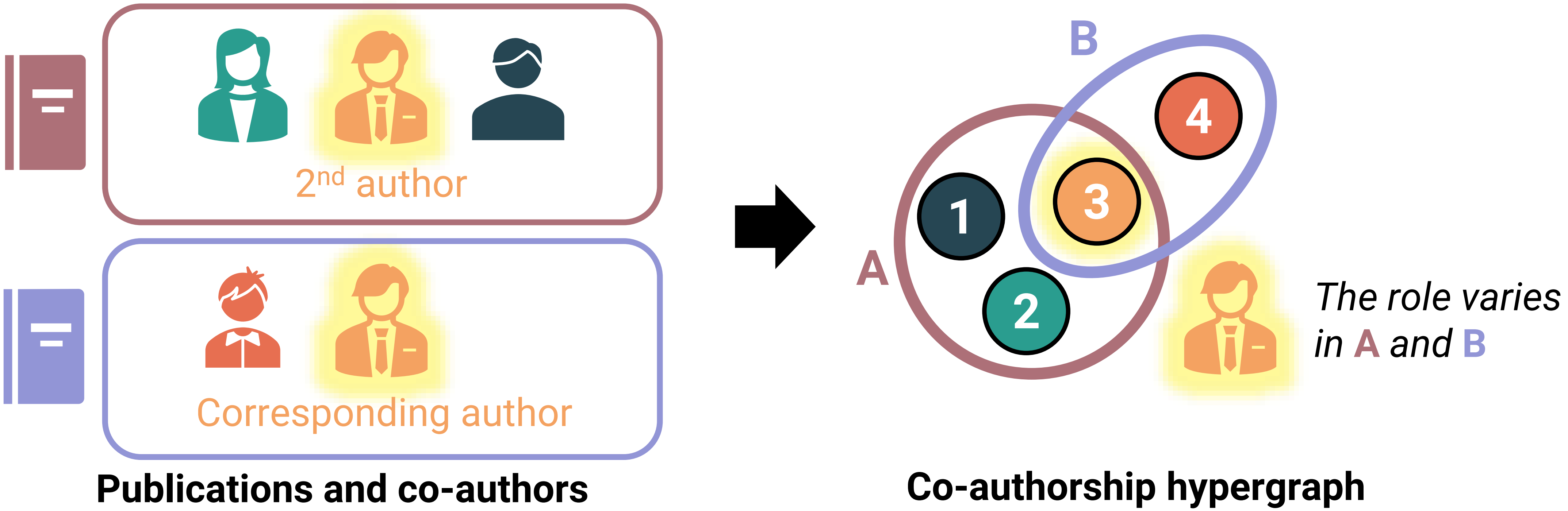


(1) Same messages

(2) Different messages

Q2) Hyperedge-Dependent Messages

- The role of a node may vary based on the hyperedges it is involved in.
- To model such characteristics, some advanced HNNs adopt **hyperedge-dependent** messages.



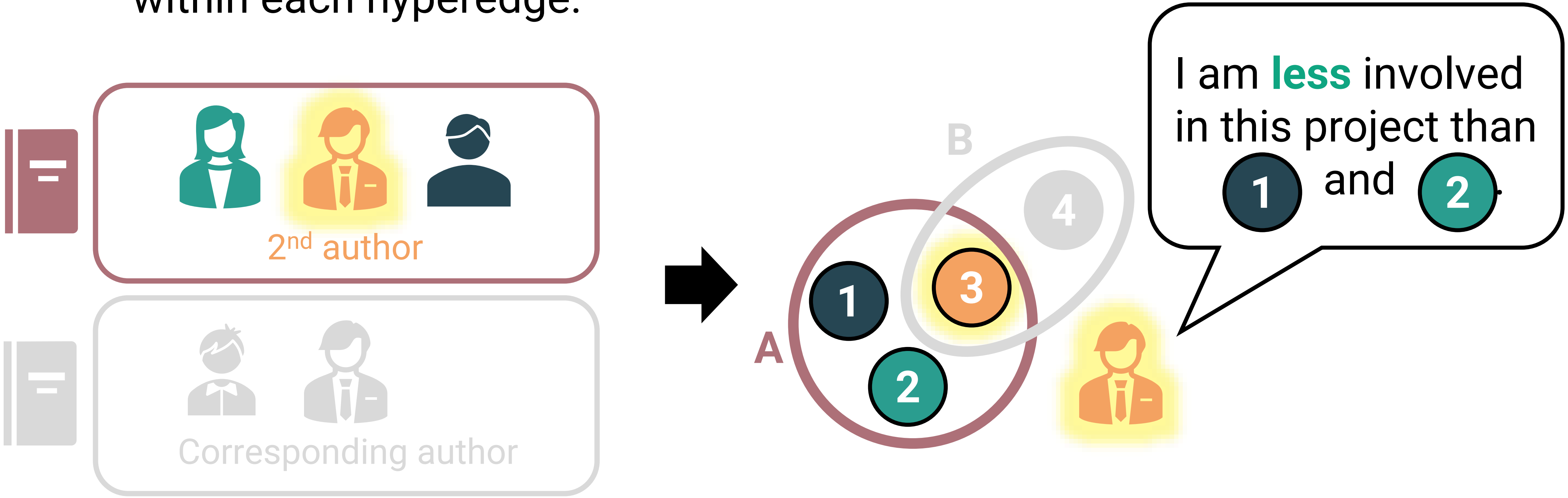
Q2) Hyperedge-Dependent Messages

- The role of a node may vary by the hyperedges it is involved in.
- To model such characteristics, some advanced HNNs adopt hyperedge-dependent messages.

Challenge: We often do not know the role played by a node in its hyperedges.

Message Vector: Hyperedge-Dependent

- A representative example: **WHATsNet** [Choe et al., 2023].
  - Employ a **positional encoding** that models the relative position of nodes within each hyperedge.

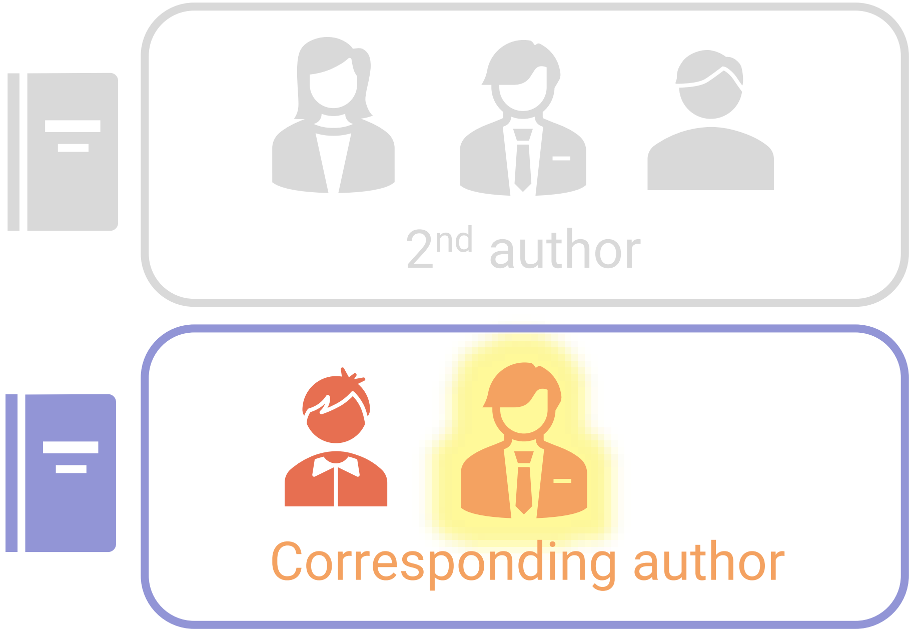


Publications and co-authors

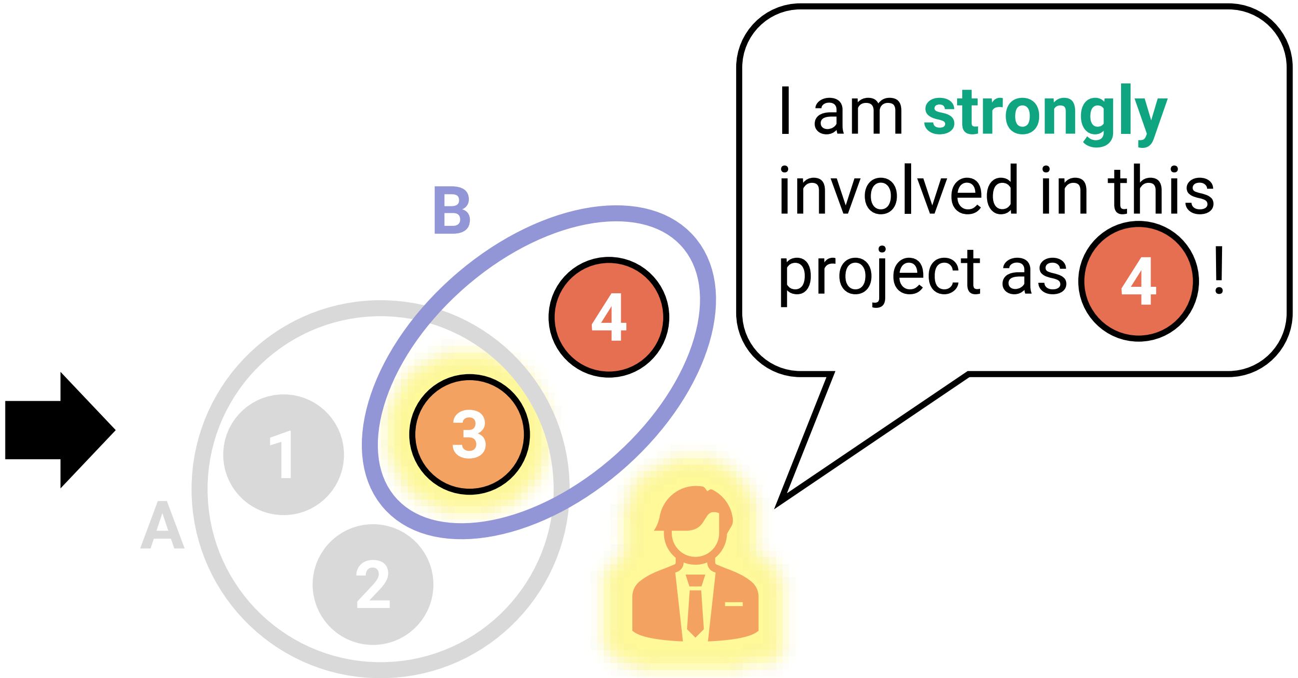
Co-authorship hypergraph

Message Vector: Hyperedge-Dependent (cont.)

- A representative example: **WHATsNet** [Choe et al., 2023].
  - Employ a **positional encoding** that models the relative position of nodes within each hyperedge.



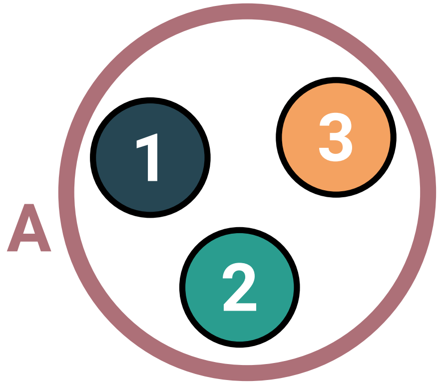
Publications and co-authors



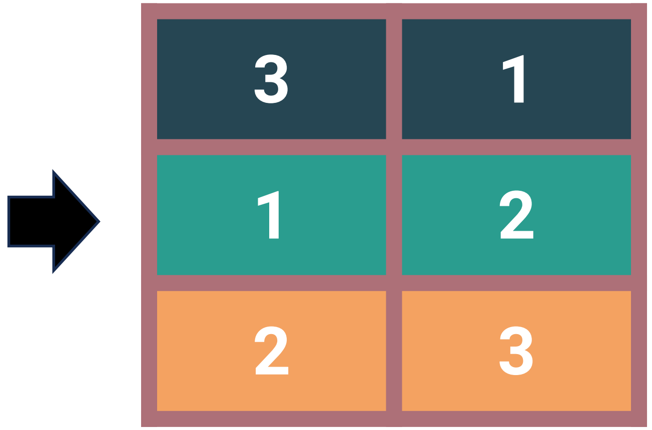
Co-authorship hypergraph

Message Vector: Hyperedge-Dependent (cont.)

- A representative example: **WHATsNet** [Choe et al., 2023].
  - It uses **within-hyperedge order** of node centrality measures as it's positional encoding.



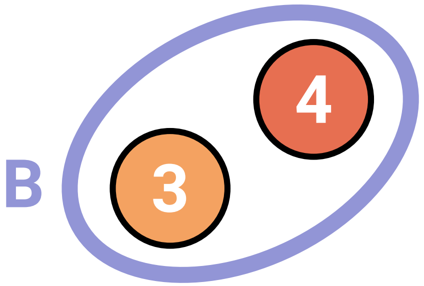
Node	Node Degree	PageRank Score
1	7	0.02
2	3	0.03
3	5	0.05



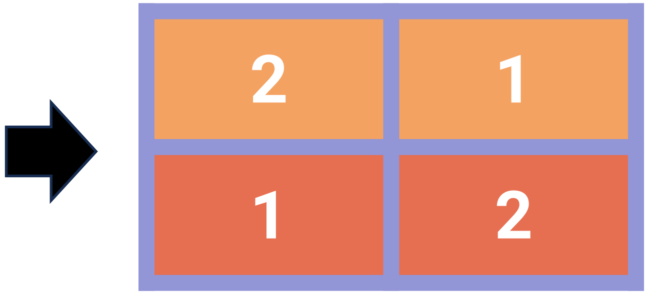
Positional encoding in A

Message Vector: Hyperedge-Dependent (cont.)

- A representative example: **WHATsNet** [Choe et al., 2023].
  - It uses **within-hyperedge order** of node centrality measures as it's positional encoding.



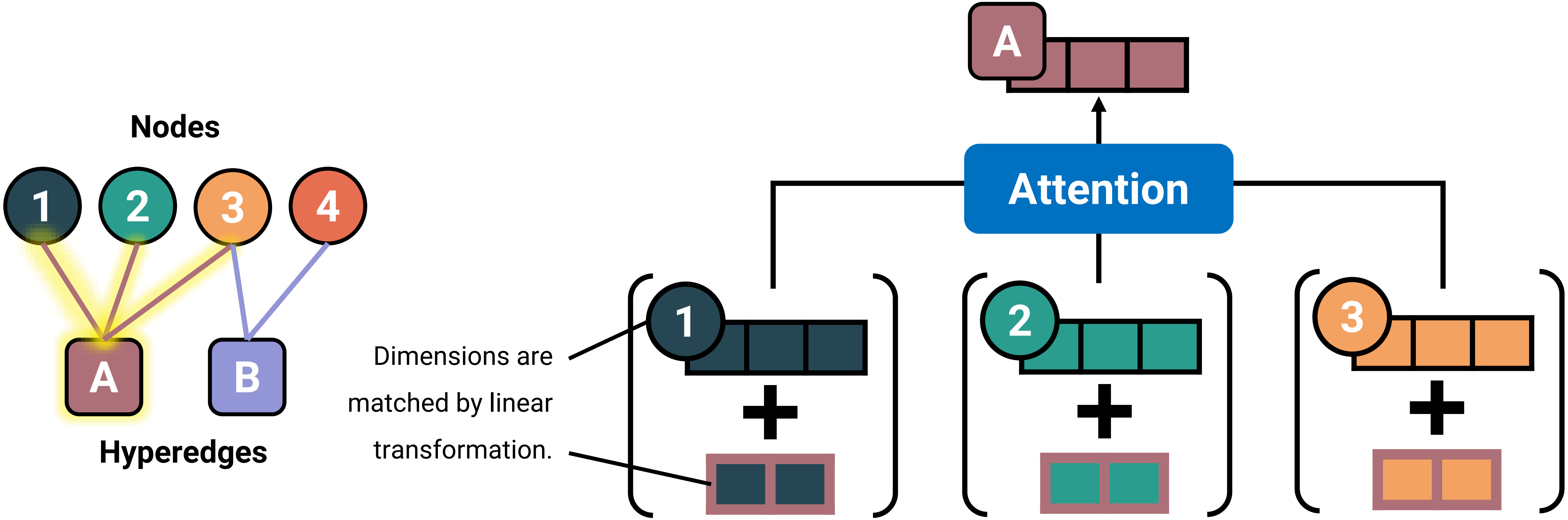
	Node Degree	PageRank Score
3	5	0.03
4	4	0.05



Positional encoding in B

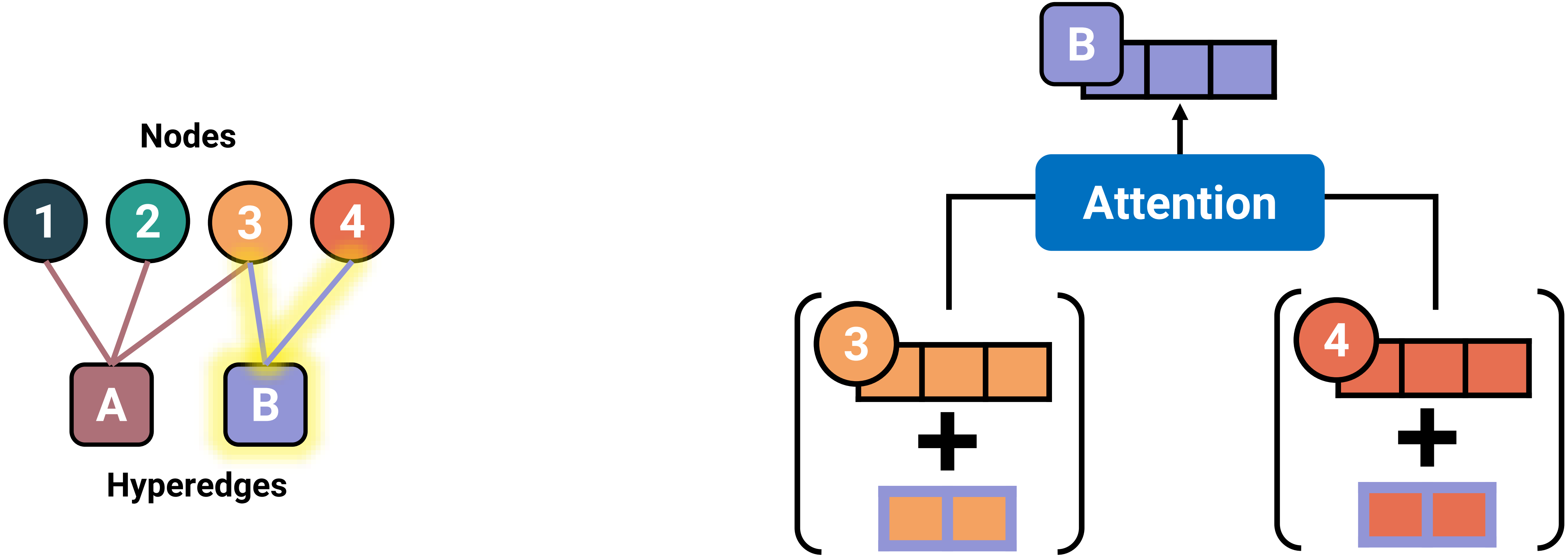
Message Vector: Hyperedge-Dependent (cont.)

- To obtain hyperedge embeddings, WHATsNet aggregates hyperedge-dependent node embeddings, which are **summation of node embeddings and positional encodings**.



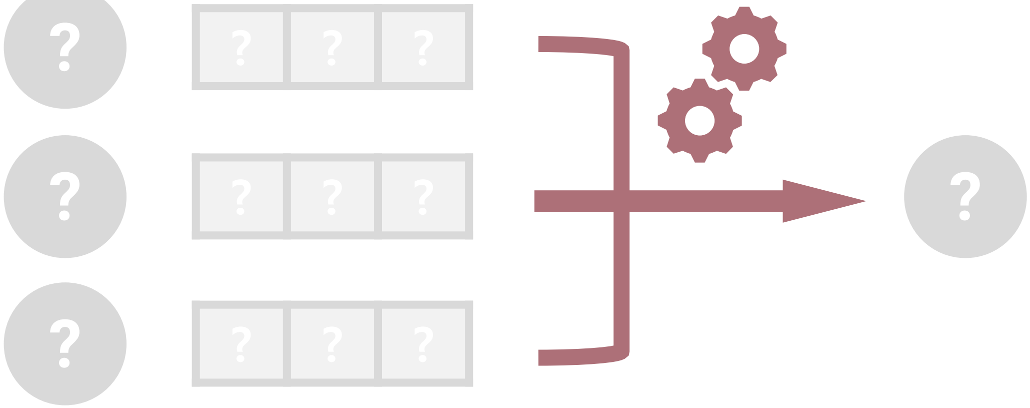
Message Vector: Hyperedge-Dependent (cont.)

- To obtain hyperedge embeddings, WHATsNet aggregates hyperedge-dependent node embeddings, which are **summation of node embeddings and positional encodings**.



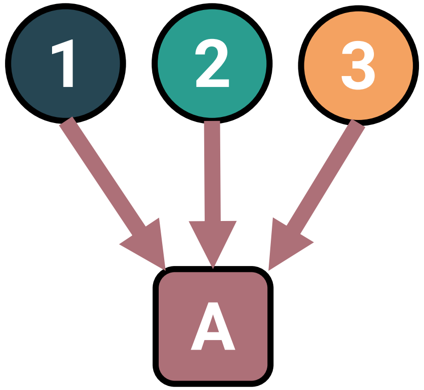
What is message passing?

- In HNNs' message passing, there are three key questions:
  - Q1) **Whose** messages to aggregate
  - Q2) **What** messages to aggregate
  - Q3) **How** to aggregate messages

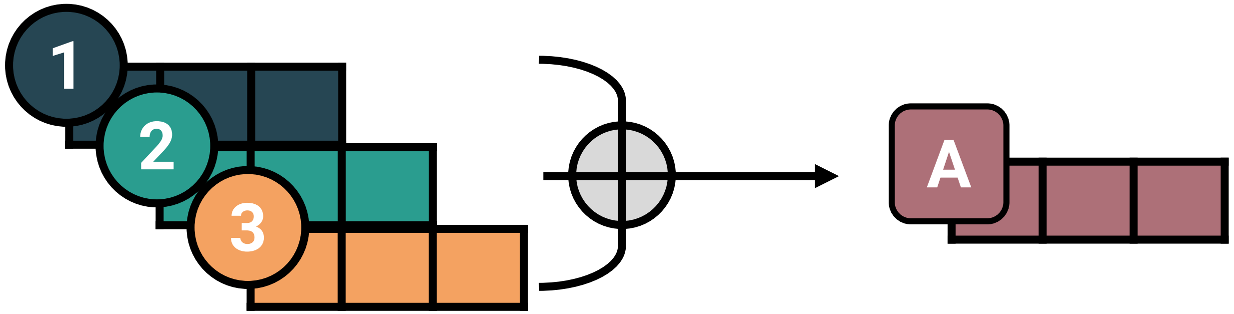


Q3) How to Aggregate Messages

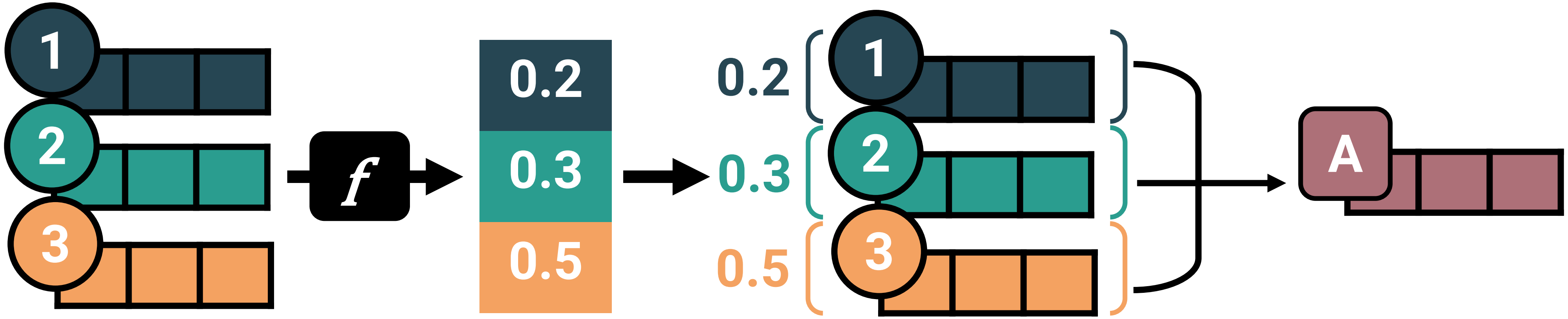
- Various aggregation functions are possible.
  1. **Fixed** pooling function
  2. **Learnable** pooling function



(1) Fixed pooling

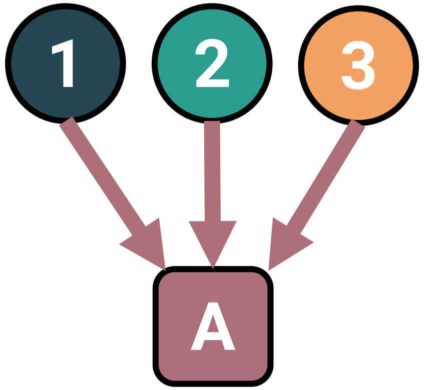


(2) Learnable pooling

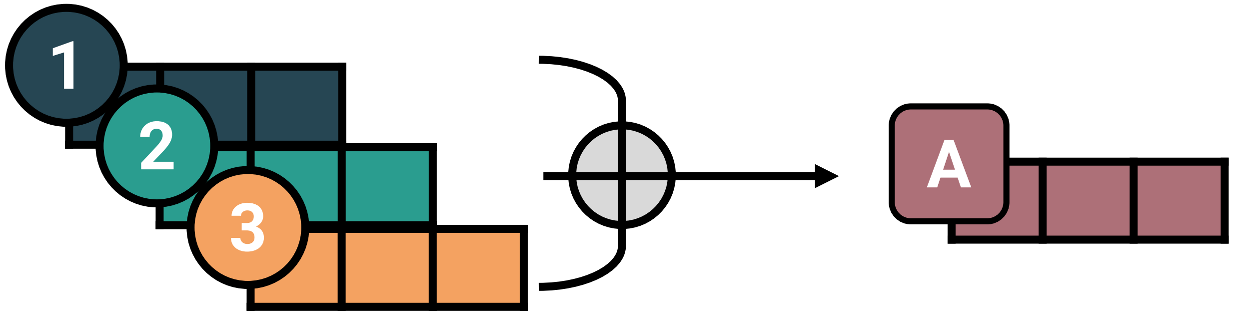


Q3) How to Aggregate Messages

- Various aggregation functions are possible.
  1. **Fixed** pooling function
  2. **Learnable** pooling function



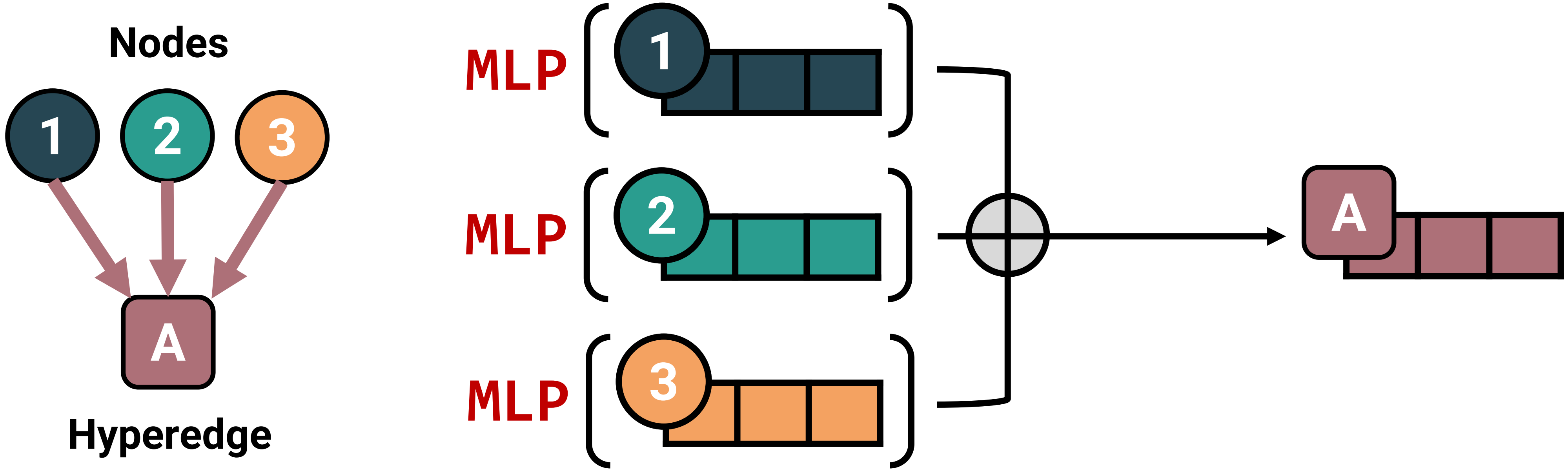
(1) Fixed pooling



(2) Learnable pooling

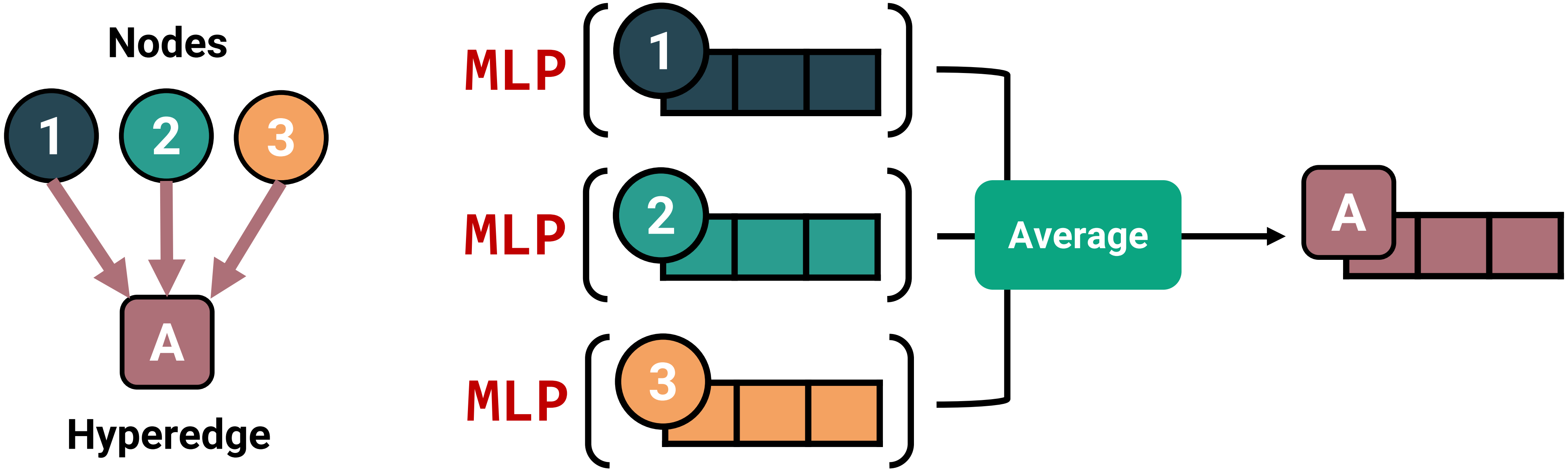
Message Aggregation: Fixed Pooling

- For simplicity, several HNNs use simple fixed pooling function.
  - Notable examples are **summation** or average.
  - An example is **ED-HNN** (described in star-expansion message passing).



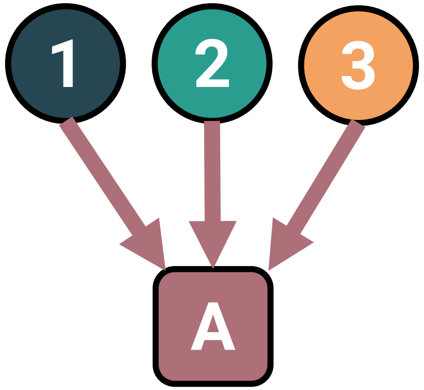
Message Aggregation: Fixed Pooling (cont.)

- For simplicity, several HNNs use simple fixed pooling function.
  - Notable examples are summation or **average**.
  - An example is **HDS-ODE** (described in star-expansion message passing).



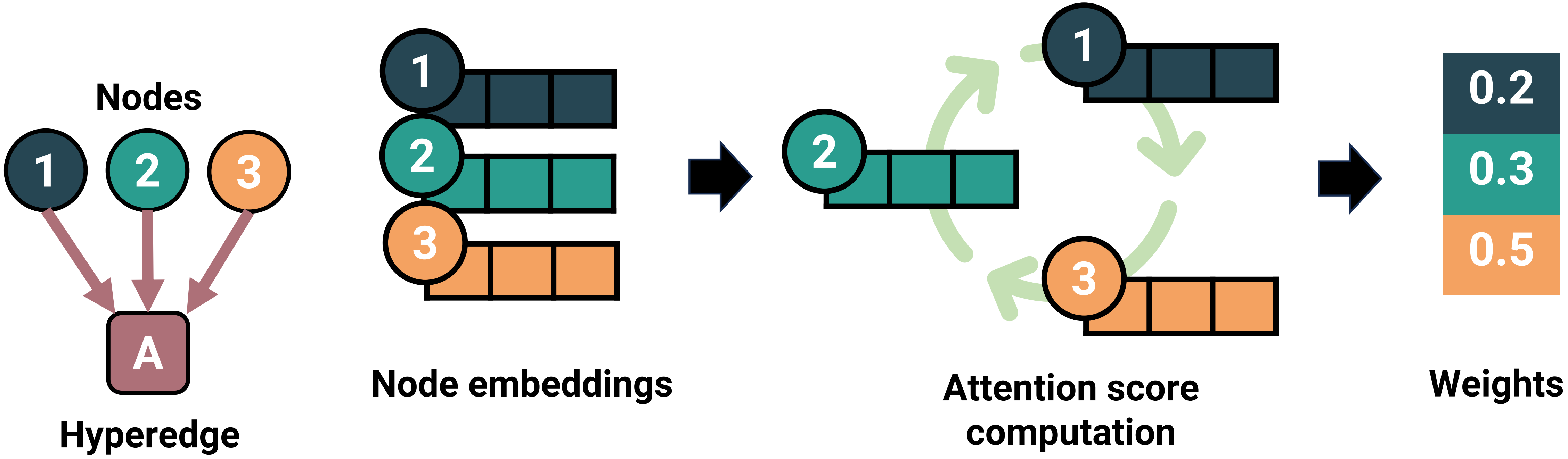
Q3) How to Aggregate Messages

- Various aggregation functions are possible.
  1. **Fixed** pooling function
  2. **Learnable** pooling function



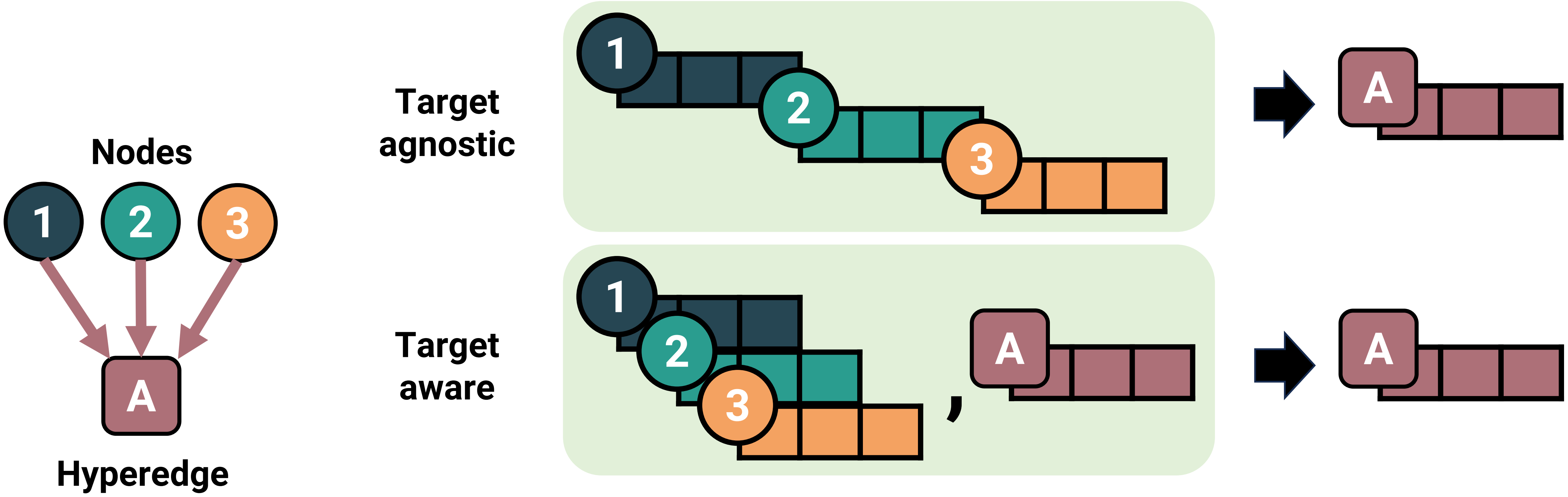
Message Aggregation: Learnable Pooling

- Several HNNs **adaptively aggregate** node/hyperedge messages.
  - To this end, the **attention** mechanism is widely used.
  - The attention mechanism assigns **different weights** to messages.



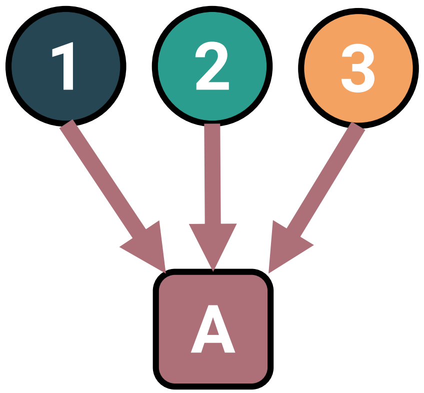
Message Aggregation: Learnable Pooling (cont.)

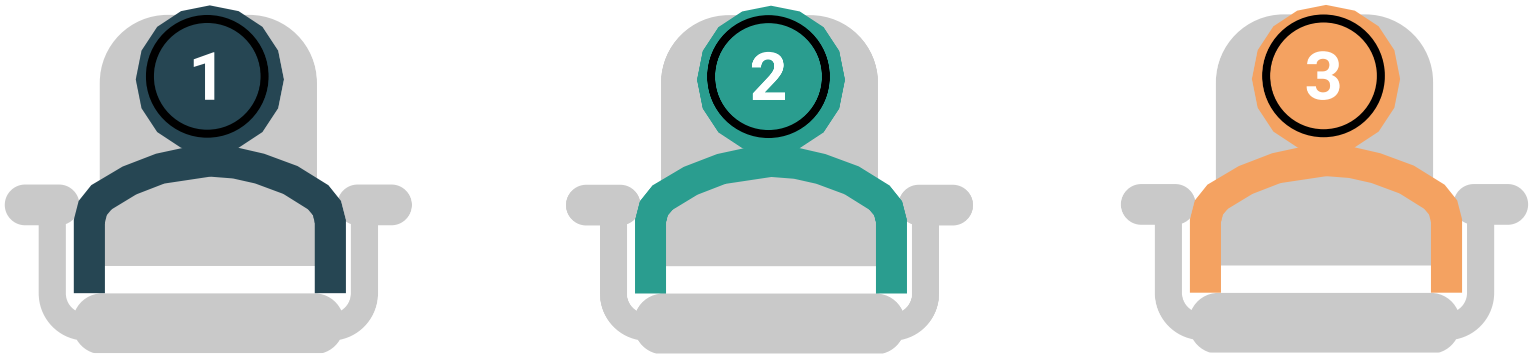
- The attention mechanism of HNNs may depend on the **target**.
  1. **Target-agnostic** attention
  2. **Target-aware** attention



Message Aggregation: Learnable Pooling (cont.)

- **Target-agnostic** attention focuses on the relationship between senders.
- A notable example is **AllSet** [Chien et al., 2022].

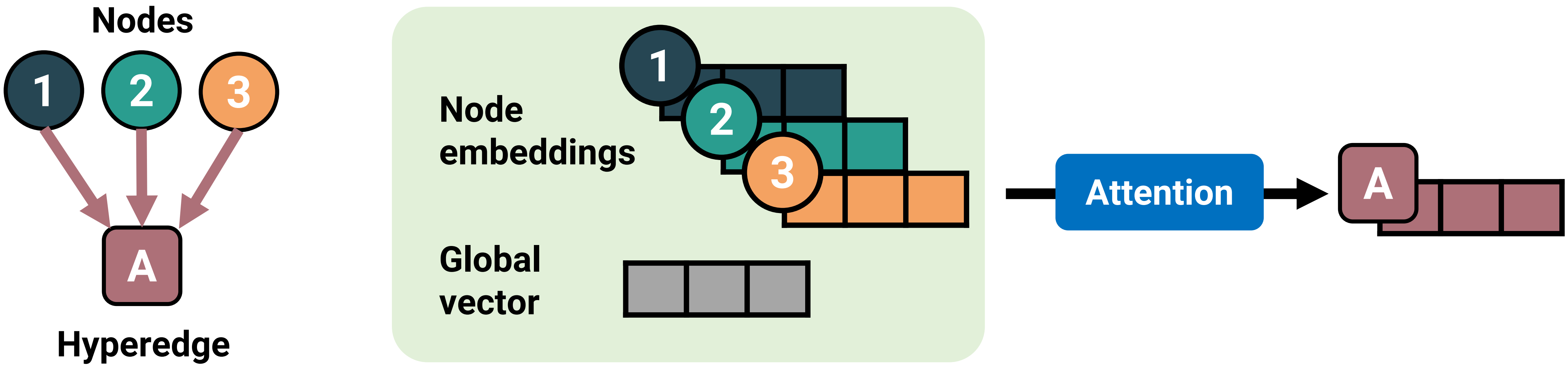




Who should get more weights among us?

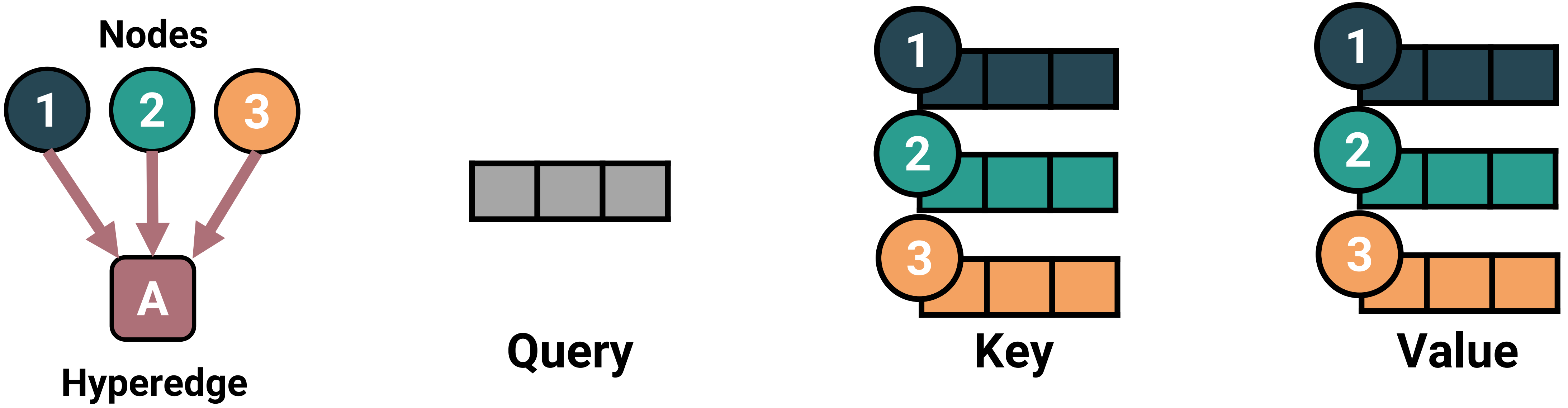
Message Aggregation: Learnable Pooling (cont.)

- **AllSet** leverages popular **Query-Key-Value** attention architecture for pooling.
  - **Query**: Learnable global vector
  - **Key**: Embeddings to be aggregated
  - **Value**: Embeddings to be aggregated



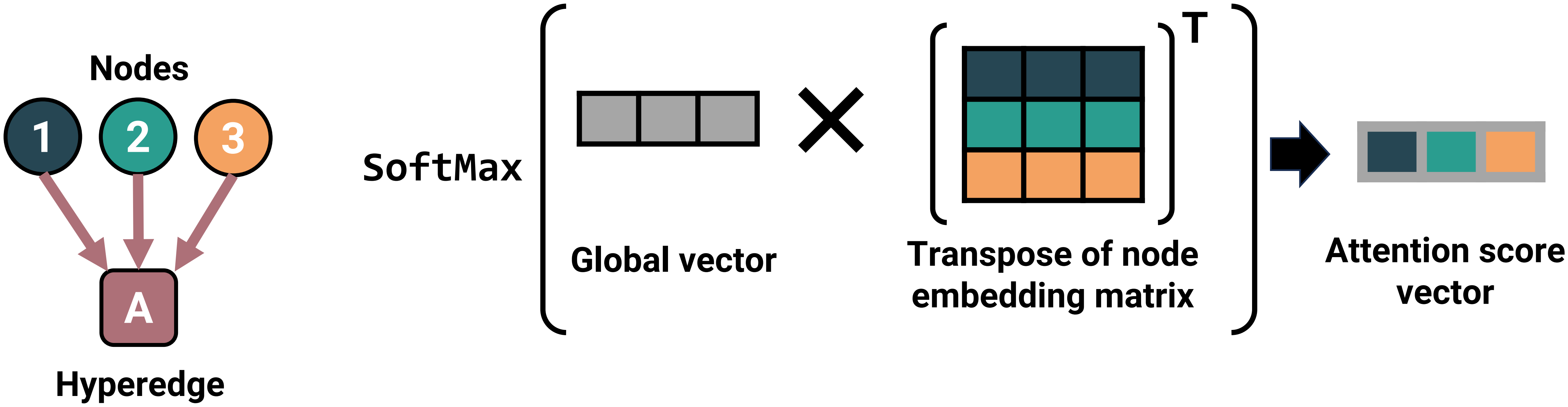
Message Aggregation: Learnable Pooling (cont.)

- **AllSet** leverages popular **Query-Key-Value** attention architecture for pooling.
  - **Query**: Learnable global vector
  - **Key**: Embeddings to be aggregated
  - **Value**: Embeddings to be aggregated



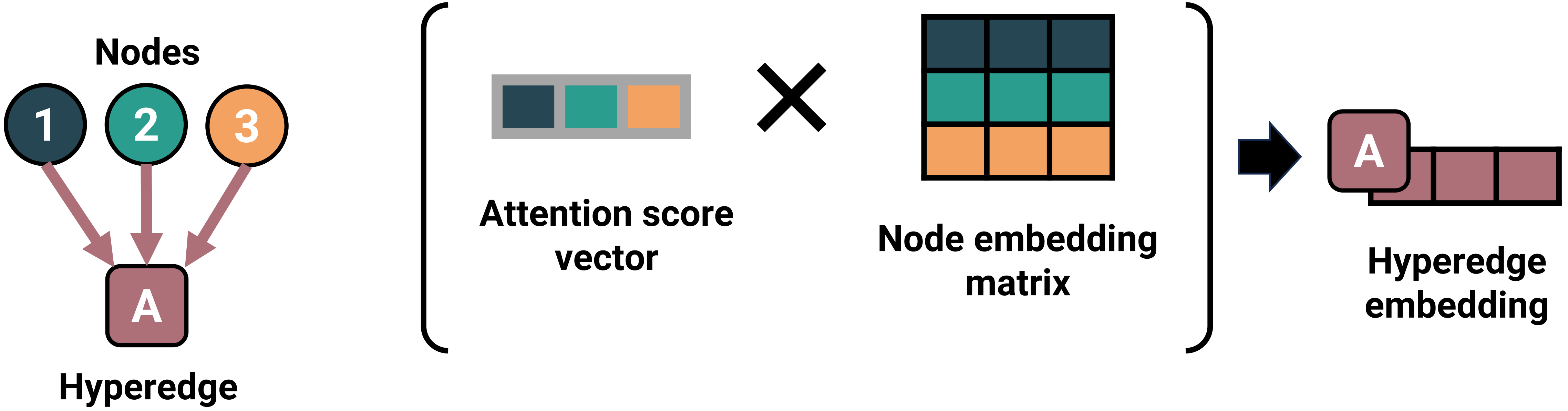
Message Aggregation: Learnable Pooling (cont.)

- **AllSet** leverages popular **Query-Key-Value** attention architecture for pooling.
  - By multiplying Query and Key, AllSet obtains **attention scores**.



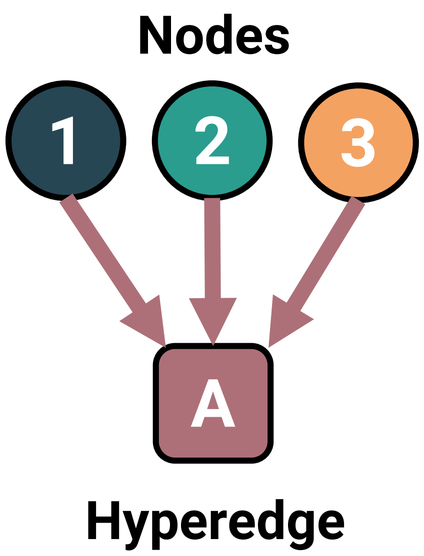
Message Aggregation: Learnable Pooling (cont.)

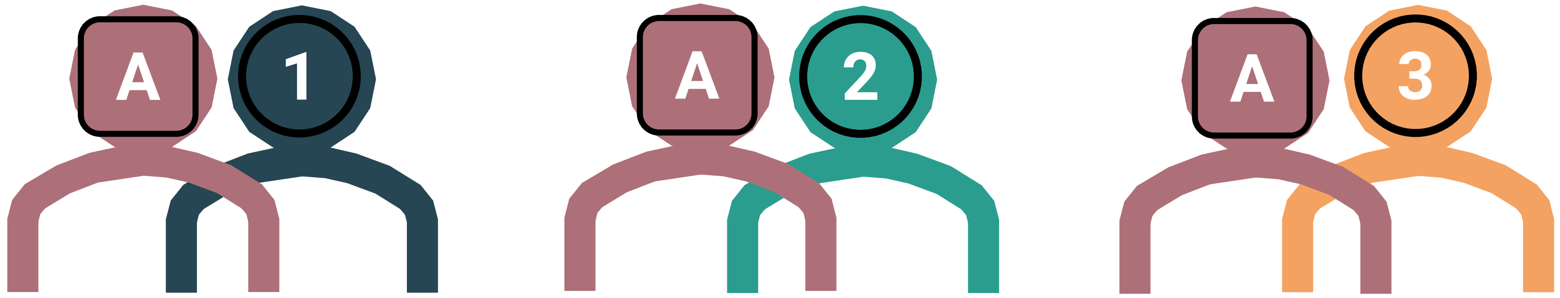
- **AllSet** leverages popular **Query-Key-Value** attention architecture for pooling.
  - Then, **AllSet** multiplies the attention score vector with the node embedding matrix.



Message Aggregation: Learnable Pooling (cont.)

- Target-aware attention focuses on the relationship between senders and the target.
- A notable example is **HyGNN** [Saifuddin et al., 2023].

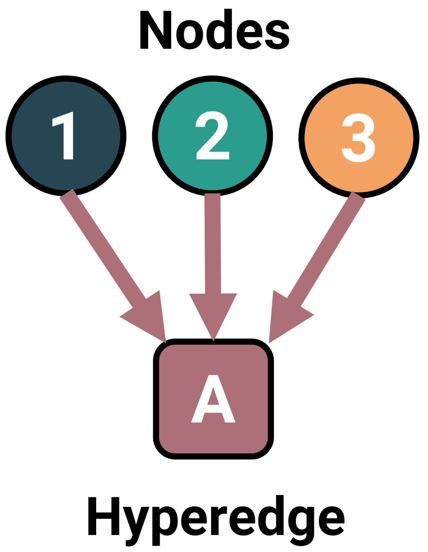


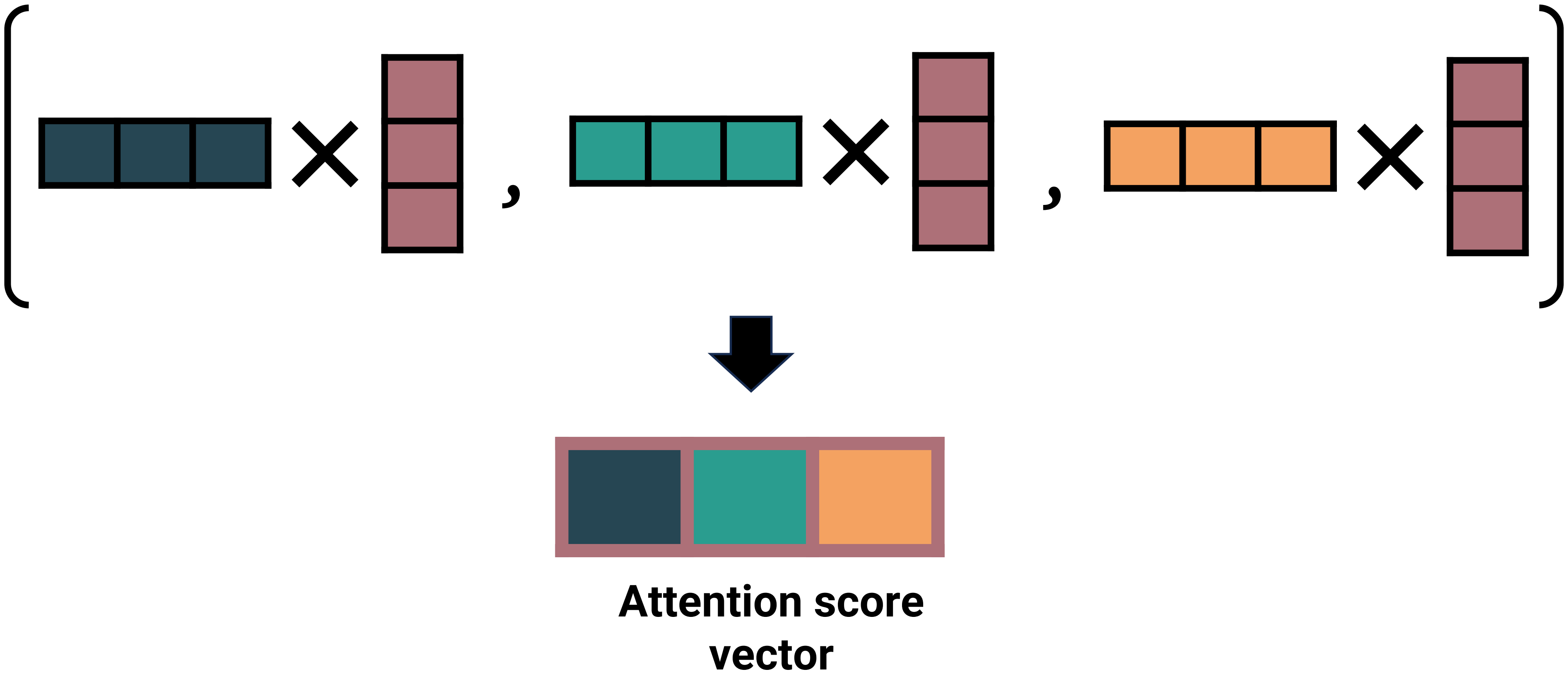


Who is closer to the target?

Message Aggregation: Learnable Pooling (cont.)

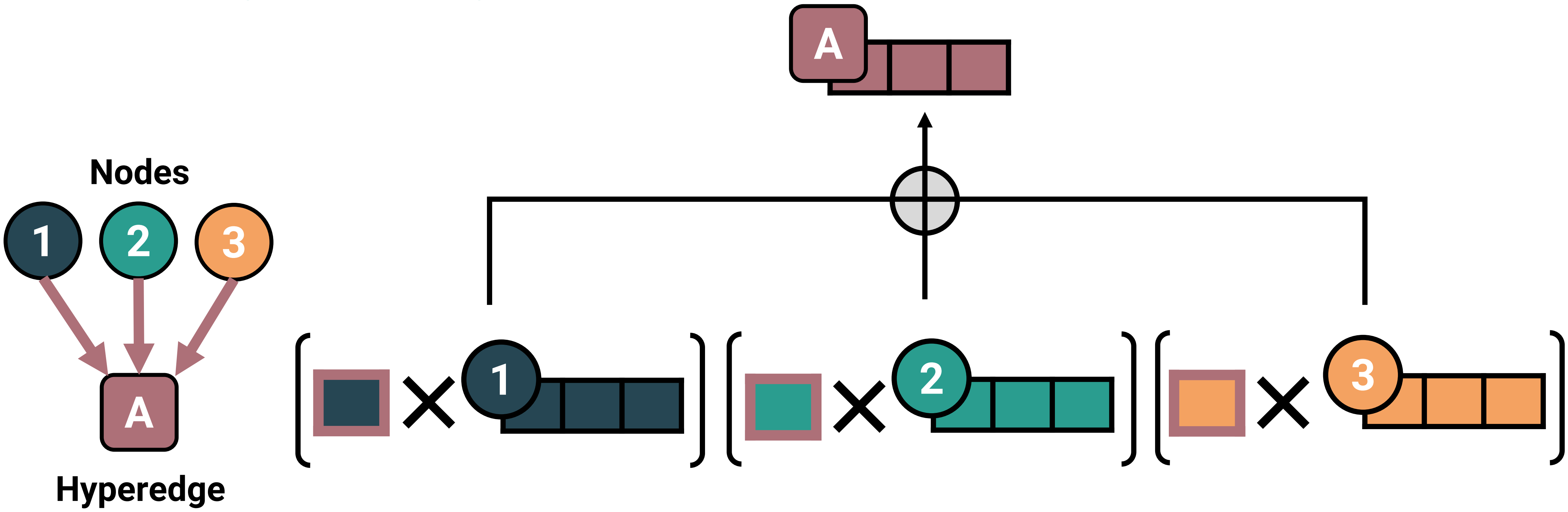
- **HyGNN** considers the similarity between the sender and receiver.
  - It uses the inner product operation to obtain attention scores.





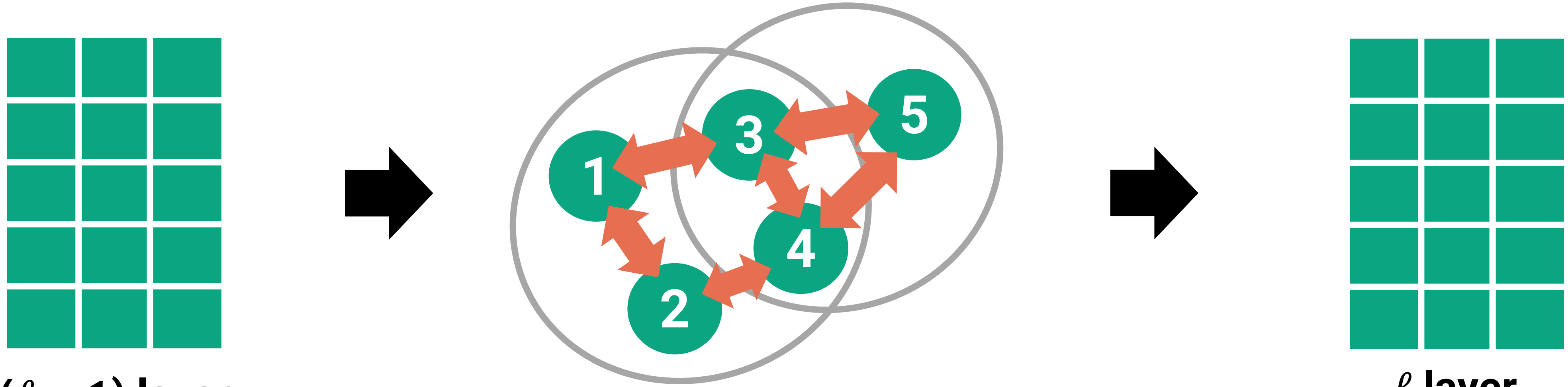
Message Aggregation: Learnable Pooling (cont.)

- **HyGNN** considers the similarity between the sender and receiver.
  - By using the attention score, HyGNN aggregate embeddings via the **weighted average**.



Part 3 Summary

- HNNs learn node (and hyperedge) embeddings by aggregating information from other nodes (and hyperedges).
- This process is called **message passing**.



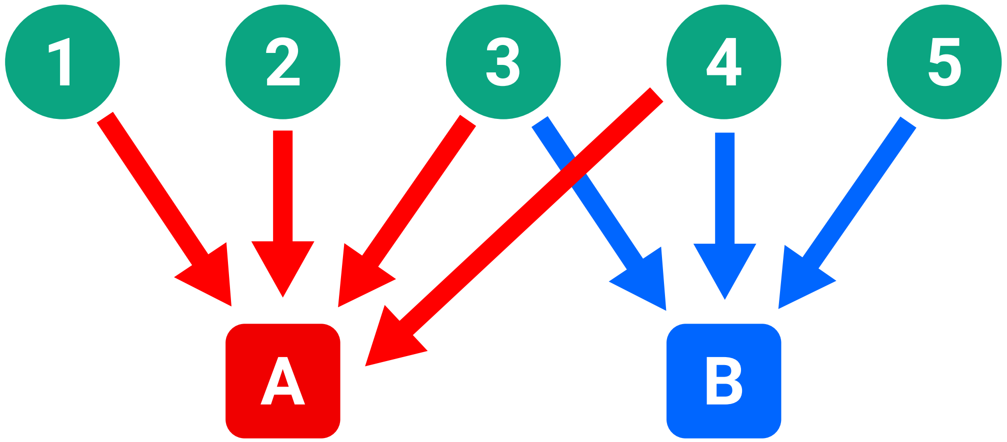
$(\ell - 1)$ layer node embeddings

Message passing

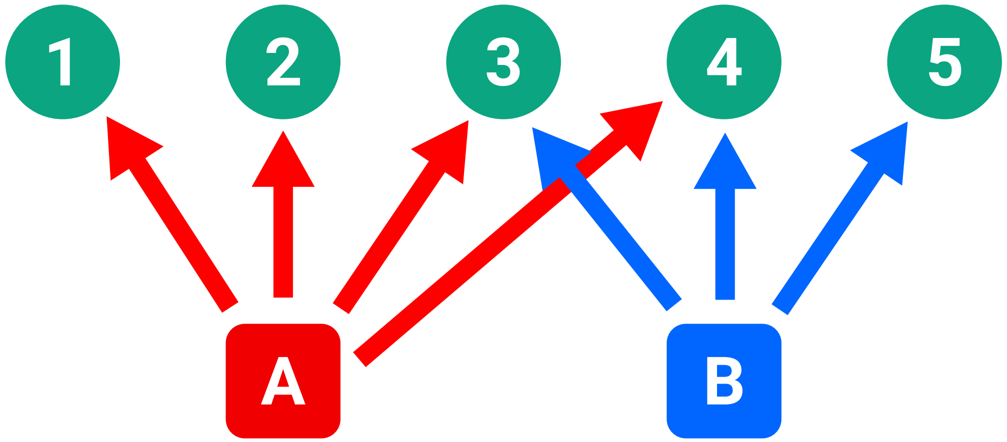
ℓ layer node embeddings

Part 3 Summary (cont.)

- There are three key components in HNNs' message passing:
  - 1) Message passing target selection
    - Two-stage: [Nodes to hyperedges] and [Hyperedges to nodes]
    - One-stage: Nodes to nodes



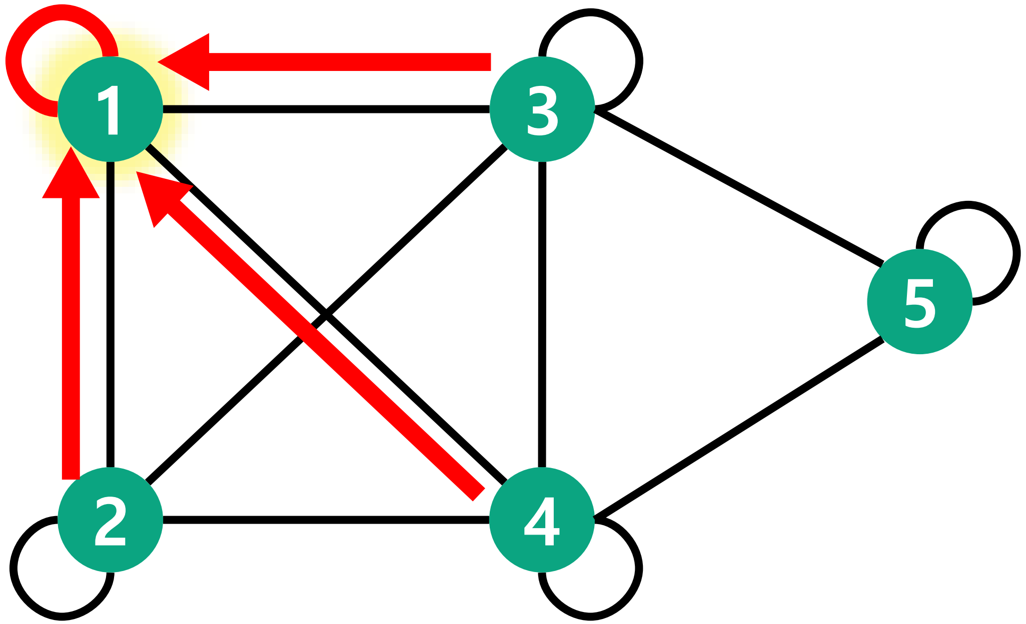
Nodes to hyperedges

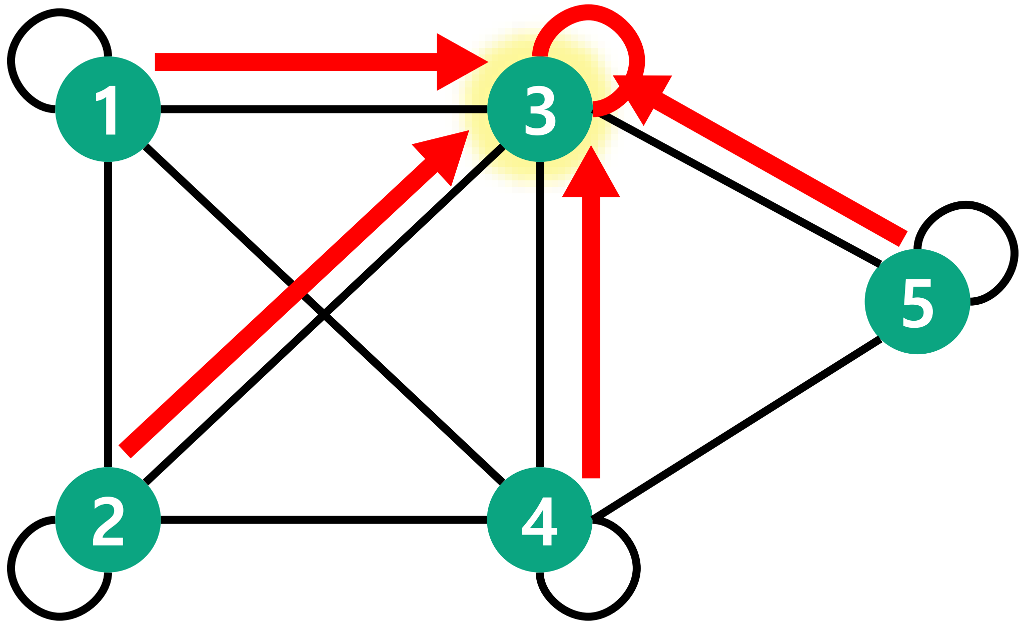


Hyperedges to nodes

Part 3 Summary (cont.)

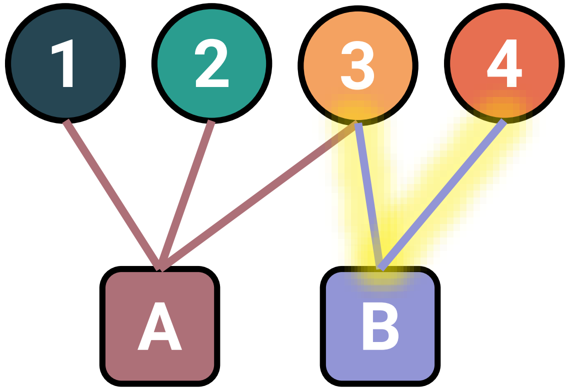
- There are three key components in HNNs' message passing:
  - 1) Message passing target selection
    - Two-stage: [Nodes to hyperedges] and [Hyperedges to nodes]
    - One-stage: Nodes to nodes

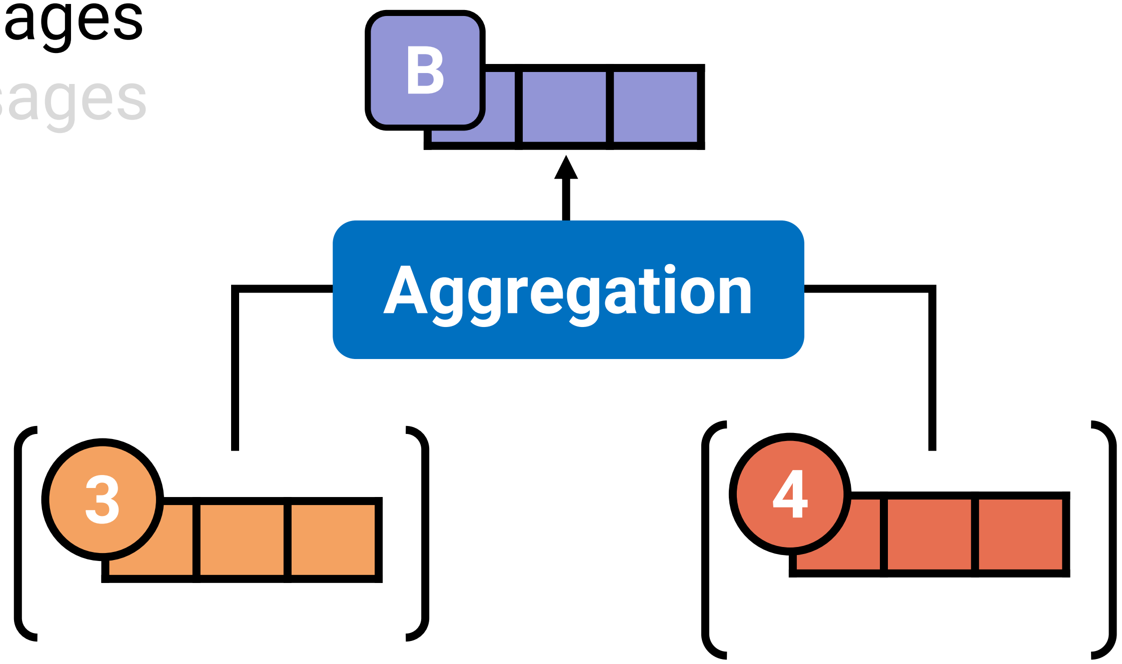




Part 3 Summary (cont.)

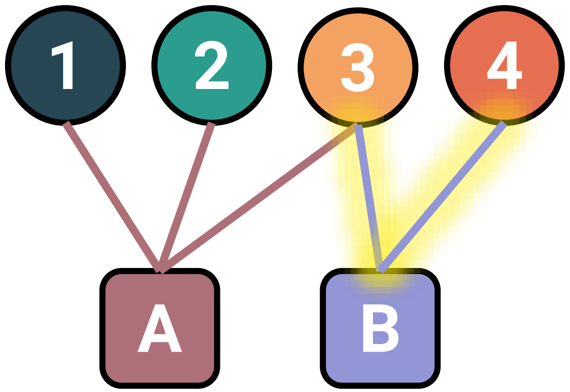
- There are three key components in HNNs' message passing:
  - 2) Message representations
    - Hyperedge-consistent messages
    - Hyperedge-dependent messages

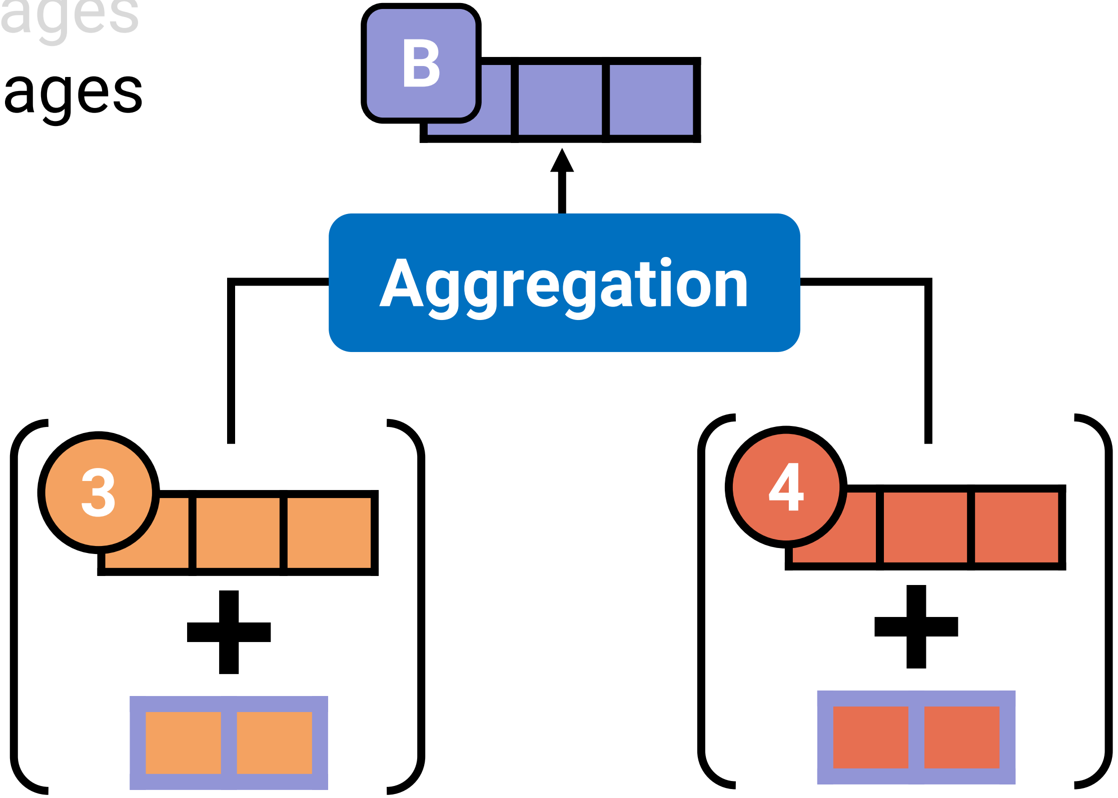




Part 3 Summary (cont.)

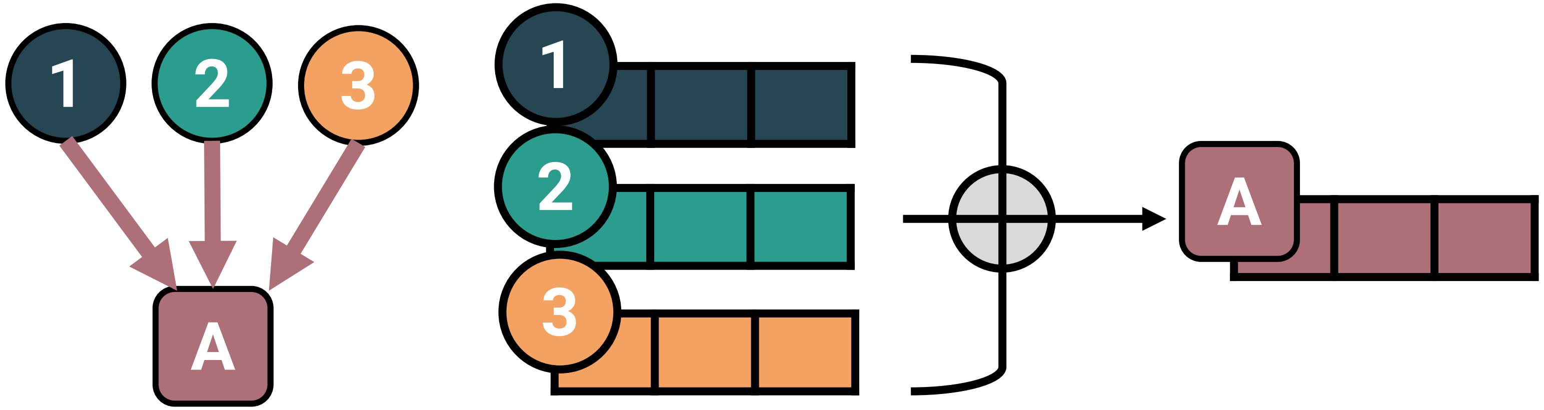
- There are three key components in HNNs' message passing:
  - 2) Message representations
    - Hyperedge-consistent messages
    - Hyperedge-dependent messages





Part 3 Summary (cont.)

- There are three key components in HNNs' message passing:
  - 3) Message aggregation functions
    - Fixed pooling function
    - Learnable pooling function



Part 3 Summary (cont.)

- There are three key components in HNNs' message passing:
  - 3) Message aggregation functions
    - Fixed pooling function
    - Learnable pooling function