

Learn When and Where to Connect: Adaptive Virtual Nodes for Dynamic Message Passing on Graphs

Jaejun Lee¹ and Joyce Jiyoung Whang^{2*}

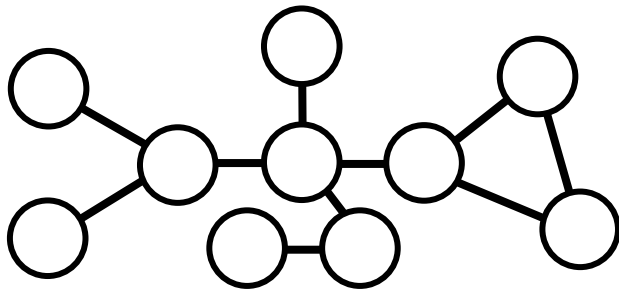
¹School of Computing, KAIST, ²Department of AI Computing, KAIST

* Corresponding Author

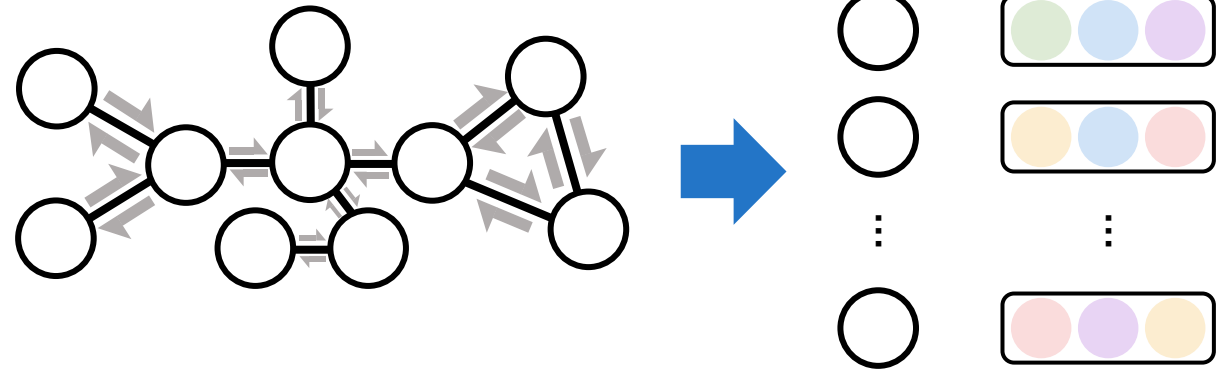
The 32nd SIGKDD Conference on Knowledge Discovery
and Data Mining (KDD 2026)



01 Message Passing Neural Networks (MPNNs)



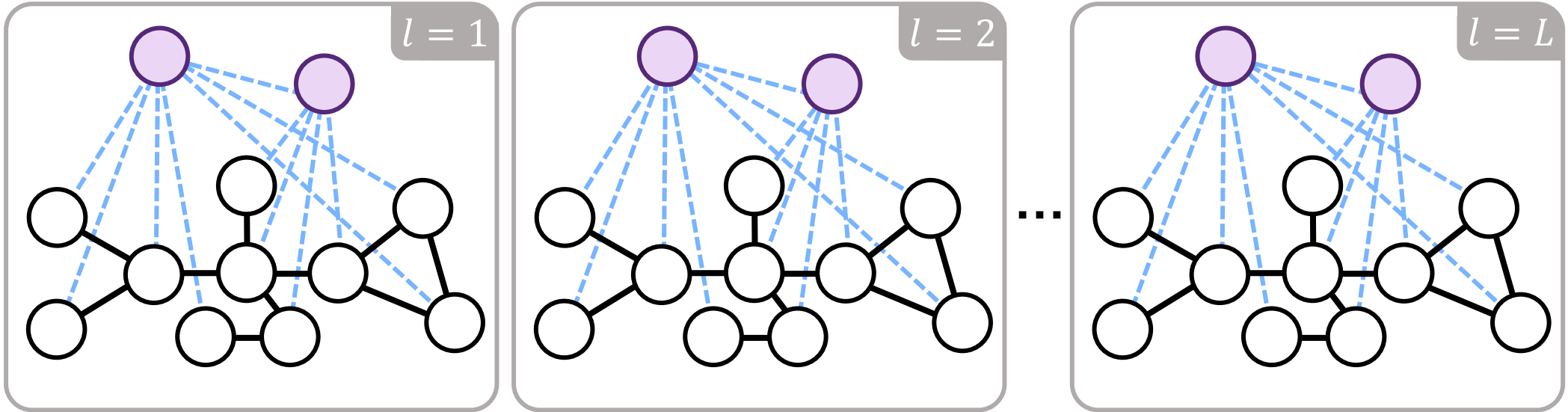
Graph



MPNN

01

Virtual Nodes (VNs) for MPNNs



Equal number of connected VNs per node

Connect a node to a VN independently of the other nodes

Connections are fixed before applying MPNNs

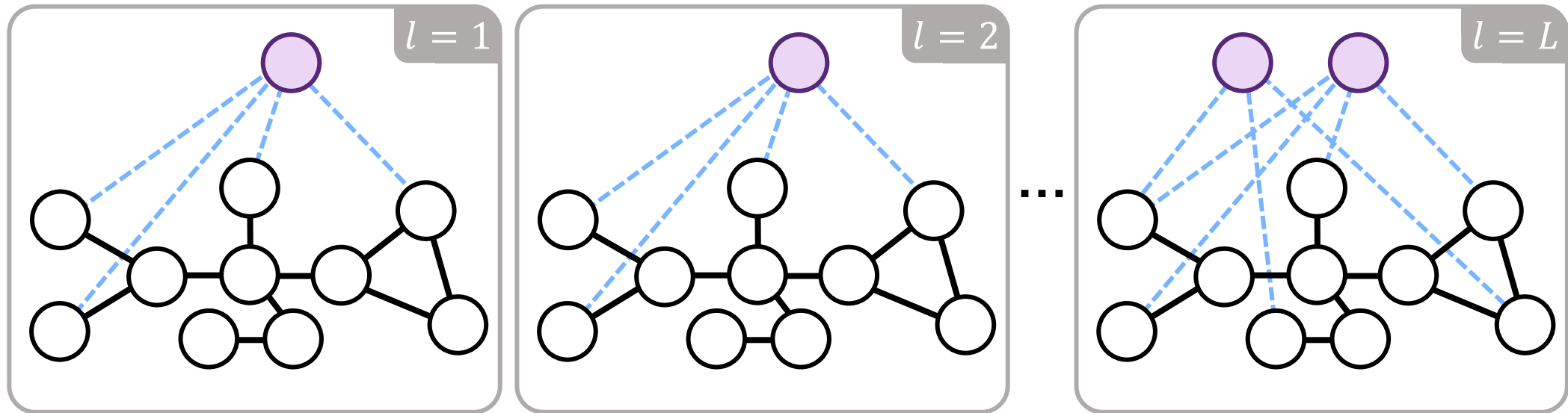
May introduce unnecessary connections

Some necessary connections may not be introduced

02

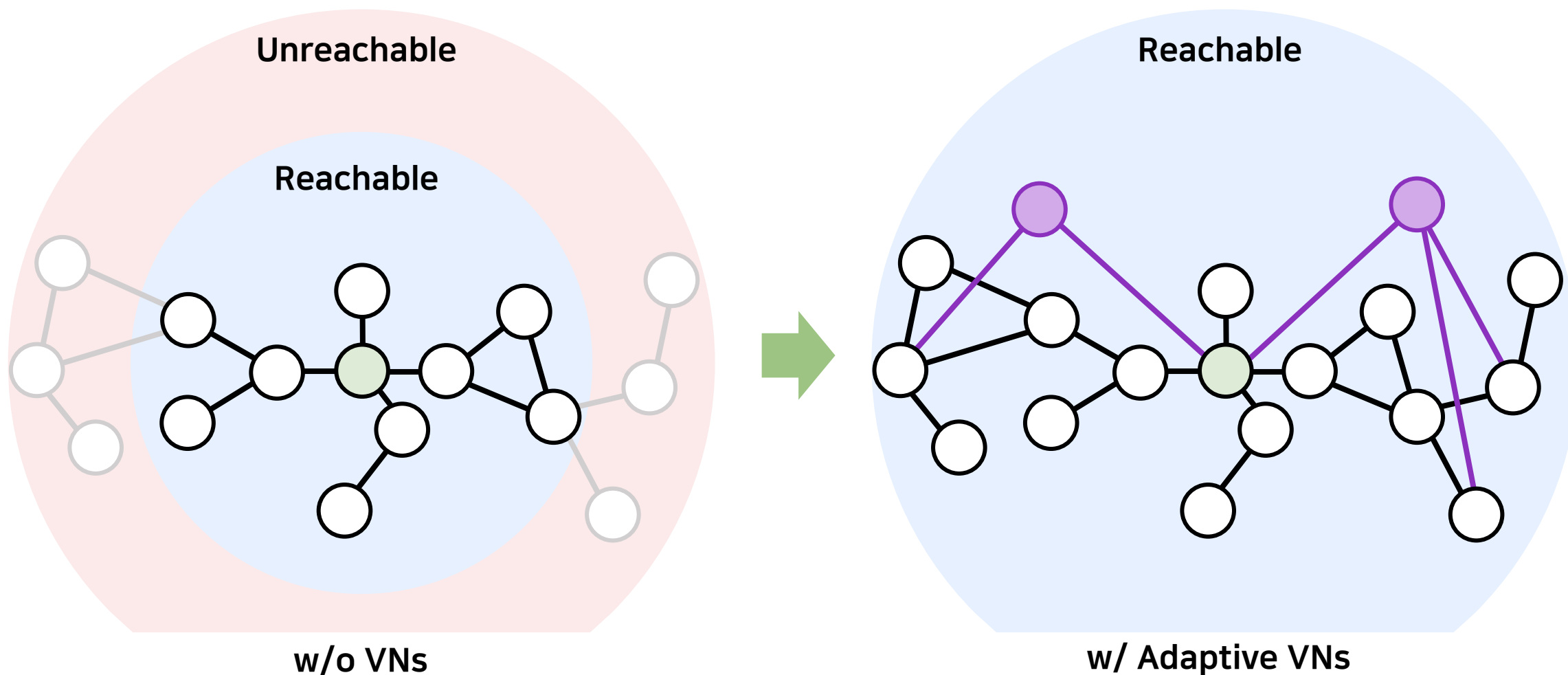
Adaptive Virtual Nodes for MPNNs

- Individual nodes may require a varying number of connections
- Establishing a node-VN connection should account for the other nodes connecting to that VN
- Additional VNs may need to be introduced at specific layers



02 Adaptive Virtual Nodes for Under-reaching

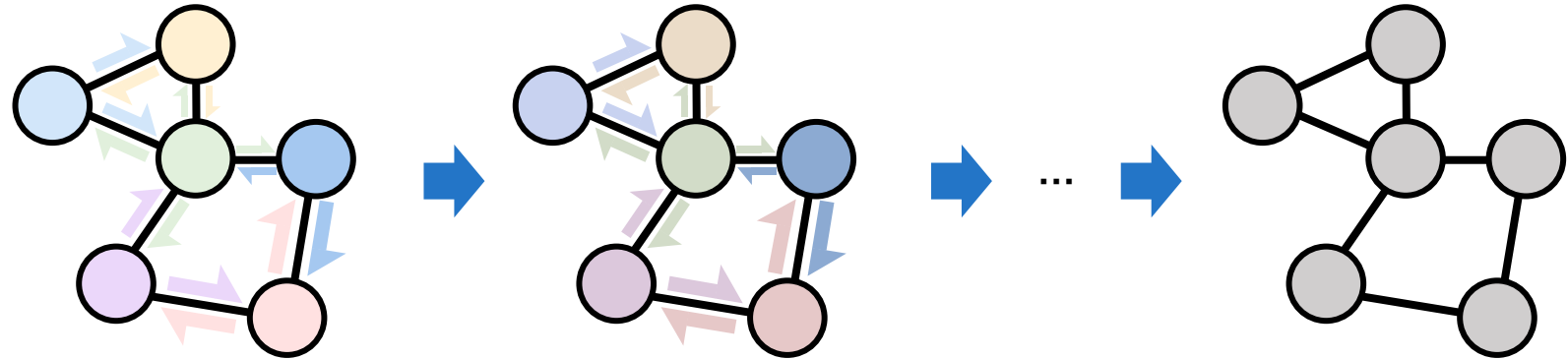
- Under-reaching: insufficiently deep layers cause MPNNs to fail to capture long-range dependencies



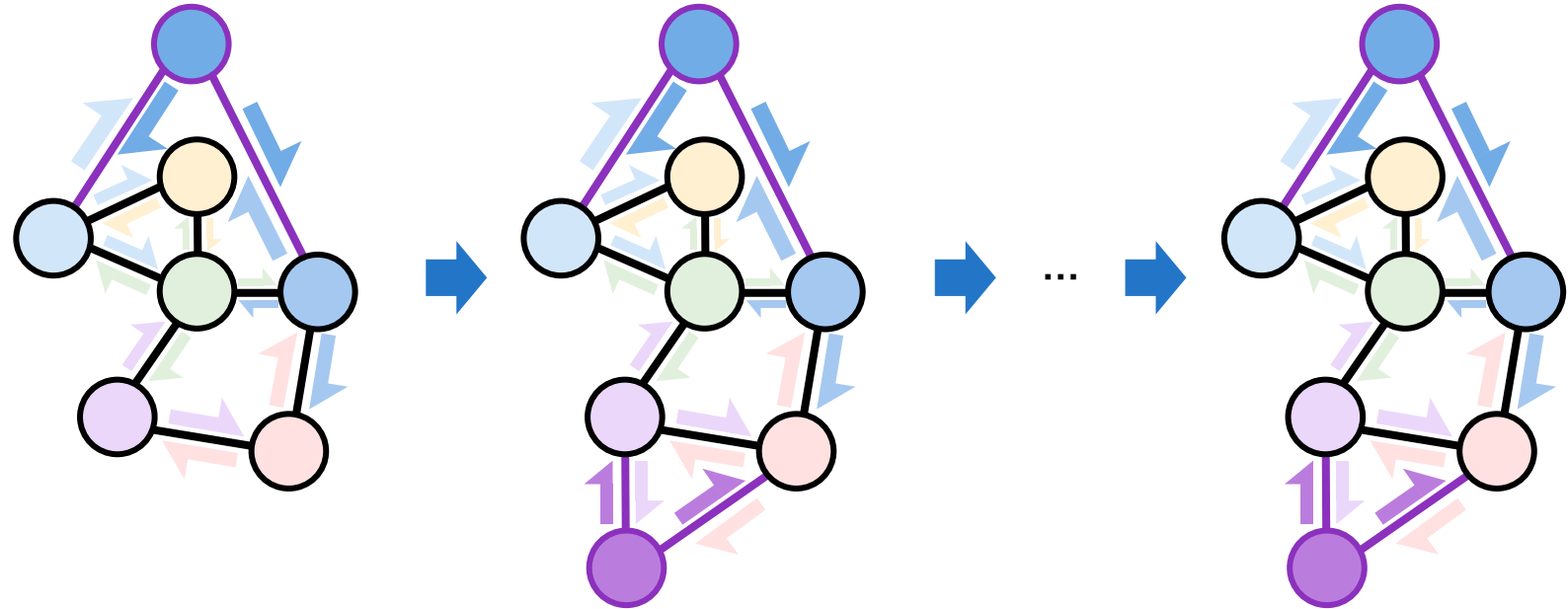
02 Adaptive Virtual Nodes for Over-smoothing

- Over-smoothing: excessively deep layers can make indistinguishable representations

w/o VNs



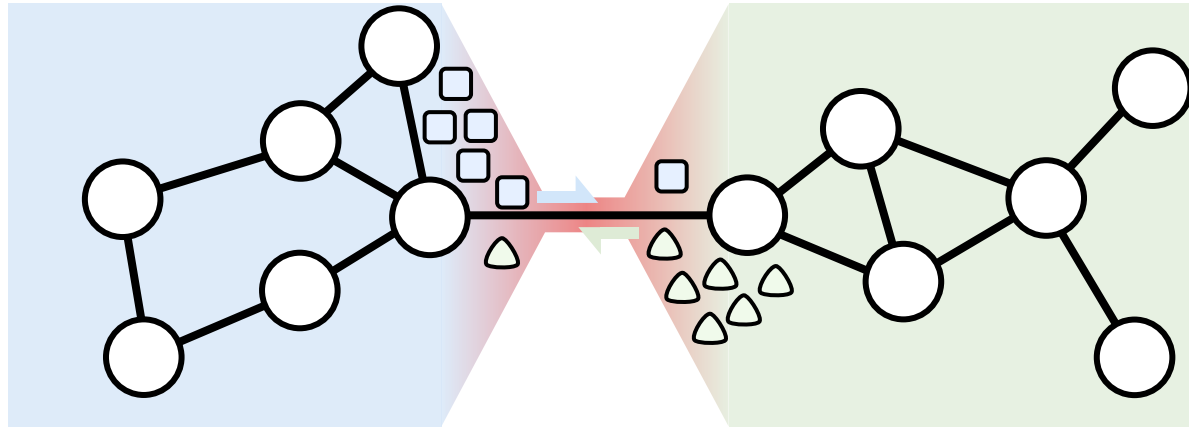
w/ Adaptive VNs



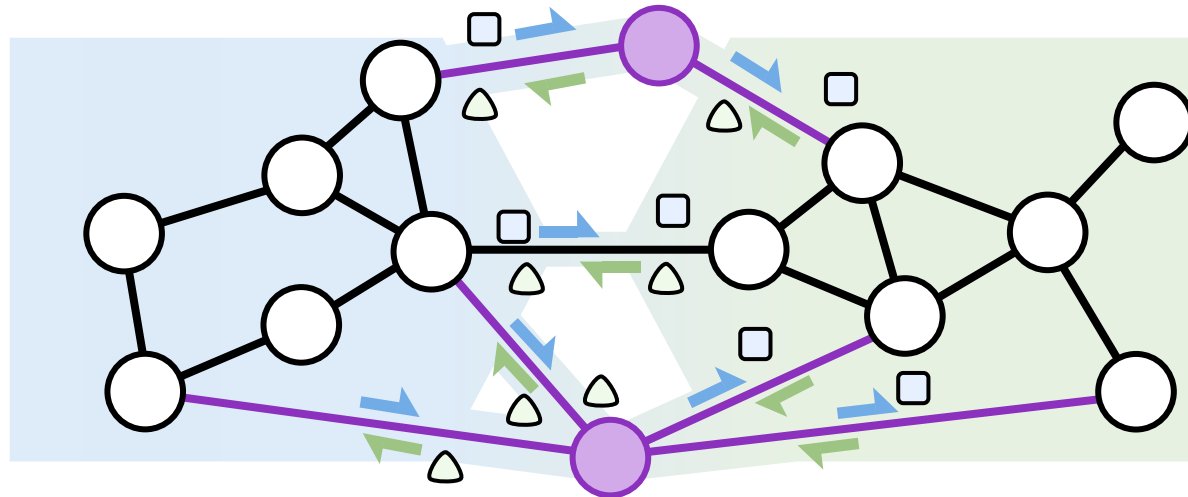
02 Adaptive Virtual Nodes for Over-squashing

- Over-squashing: fixed-size vectors may struggle to encode information from extensive receptive fields

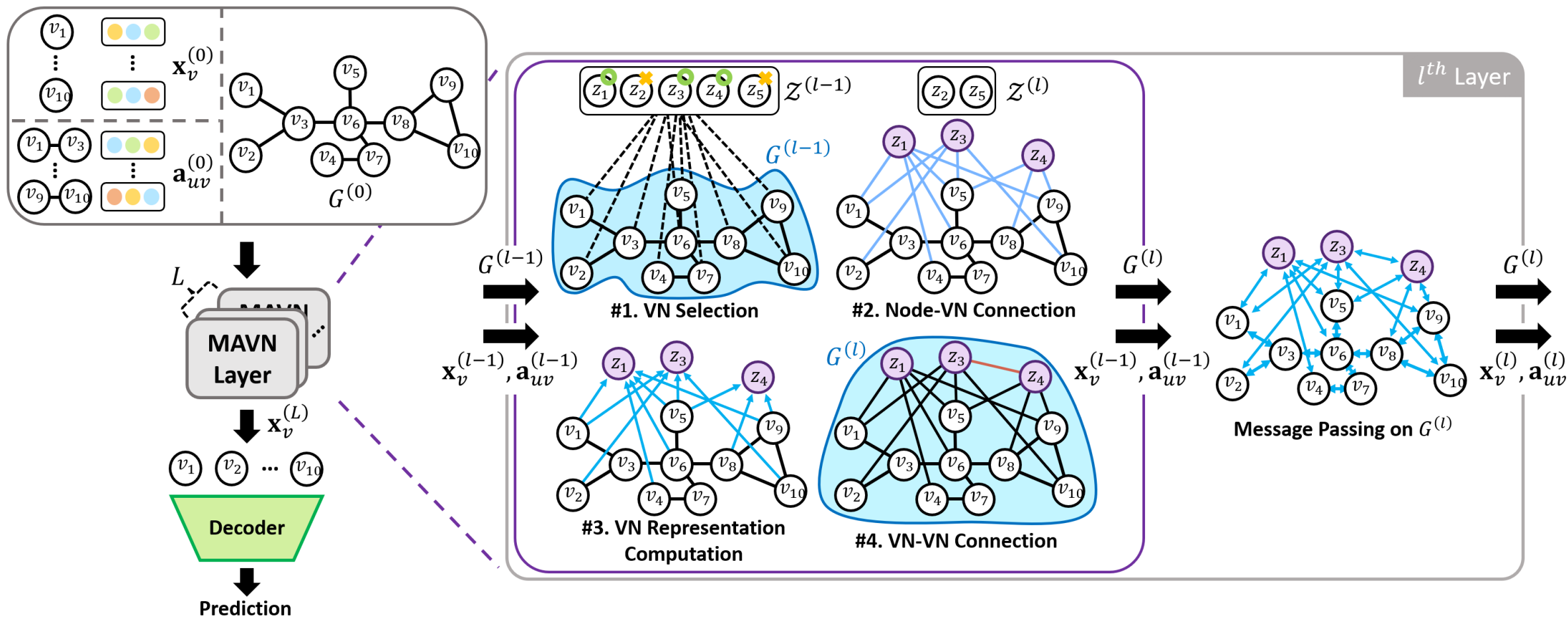
w/o VNs



w/ Adaptive VNs

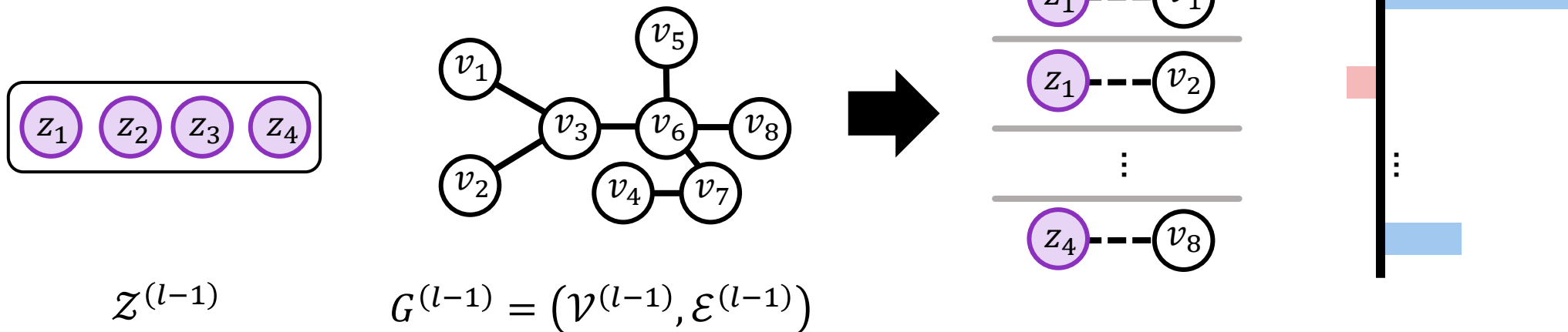


MAVN: Dynamic Message Passing with Adaptive Virtual Nodes



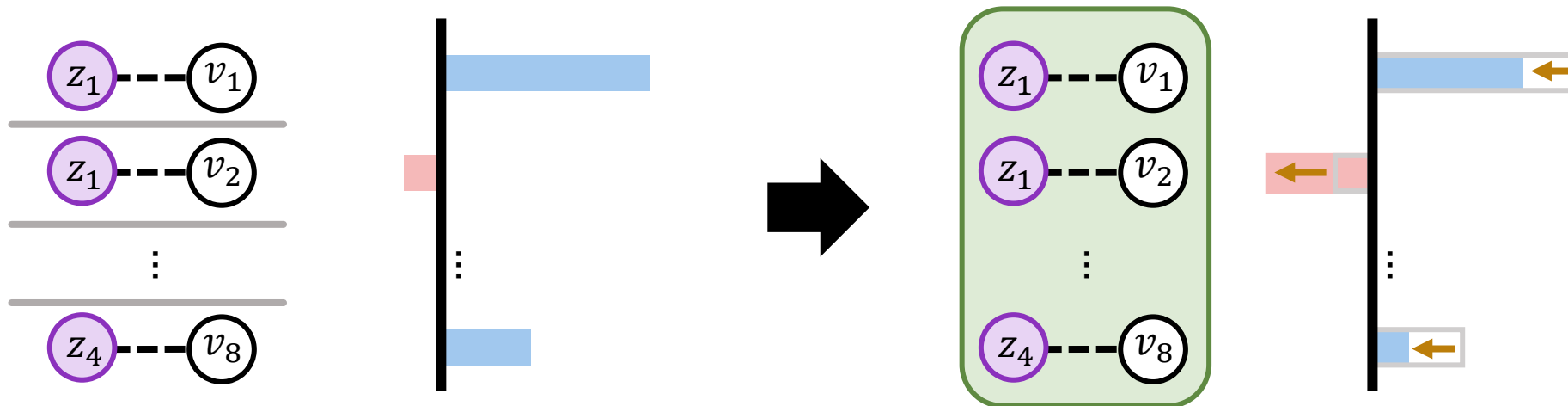
03 Relevance Score Computation

- The score used to select VNs, connect the selected VNs and nodes, and form connections between the selected VNs
 - Adjusted based on their relative importance across all candidates
- Computed for all pairs of the **remaining VNs** $\mathcal{Z}^{(l-1)}$ and the **nodes** $\mathcal{V}^{(l-1)}$ in the graph $G^{(l-1)}$ from the previous layer
 - $s_{vz}^{(l)} = \text{MLP}^{(l)}(\mathbf{x}_v^{(l-1)}) \cdot k_z^{(l)} / \sqrt{d_{\text{dot}}}, z \in \mathcal{Z}^{(l-1)}, v \in \mathcal{V}^{(l-1)}$



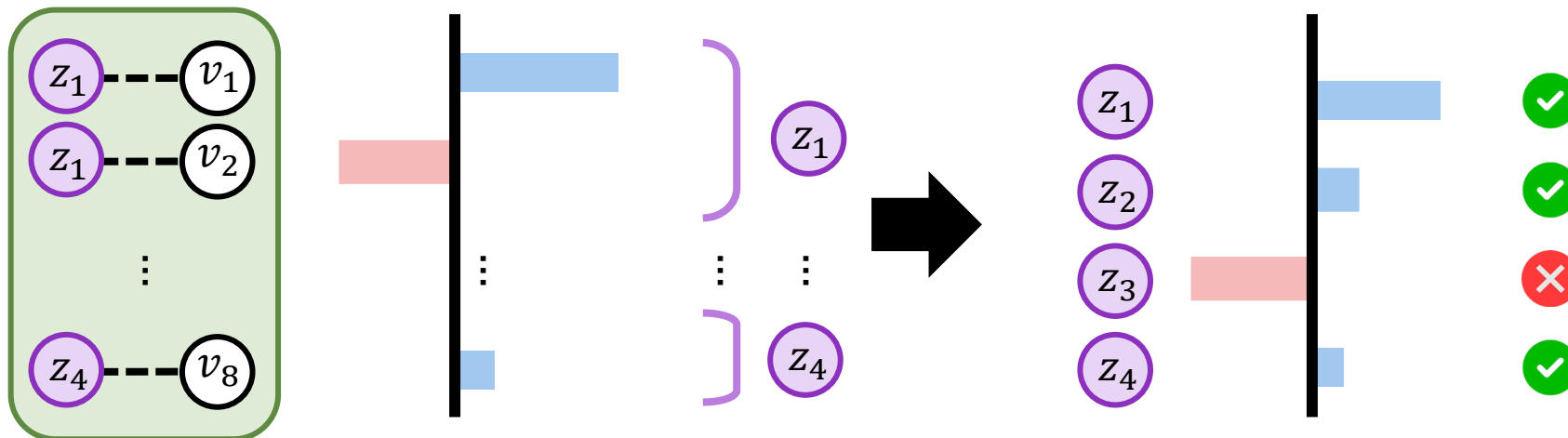
03 VN Selection

- Selects the VNs to introduce by considering the relative importance of the connections the VNs will form by introducing them
- Adjusts the relevance scores with their **relative significance among all node-VN pairs**
 - $\bar{s}_{vz}^{(l)} = s_{vz}^{(l)} + \alpha \cdot \text{logsoftmax}(\mathcal{S}^{(l)}) [s_{vz}^{(l)}], \mathcal{S}^{(l)} = \{s_{v'z'}^{(l)} | v' \in \mathcal{V}^{(l-1)}, z' \in \mathcal{Z}^{(l-1)}\}$



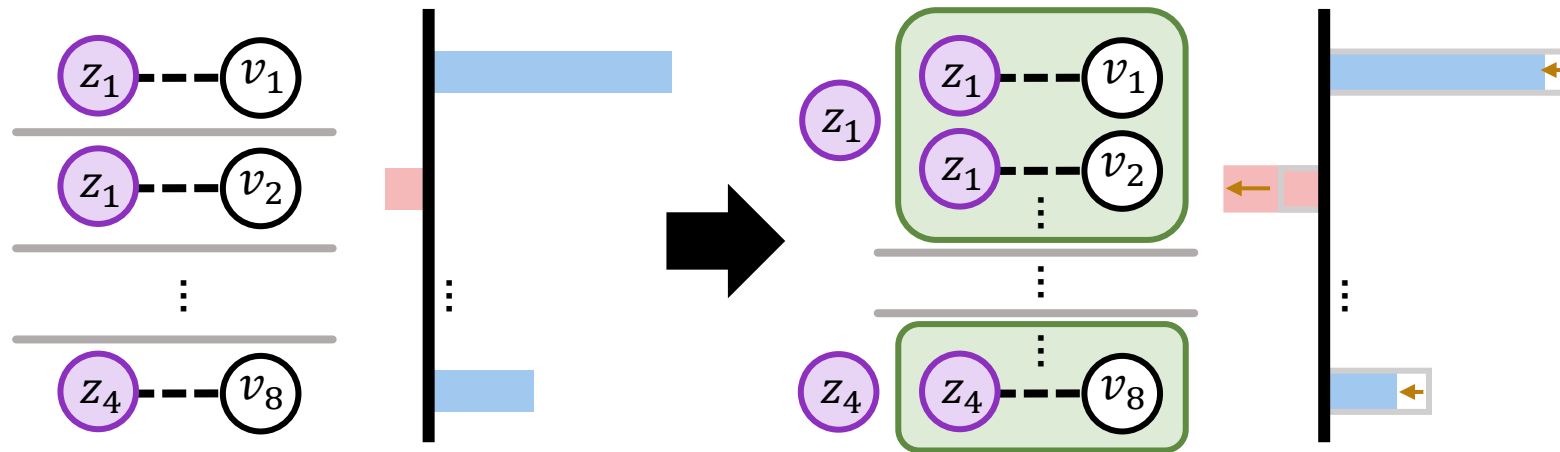
03 VN Selection

- Selects the VNs to introduce by considering the relative importance of the connections the VNs will form by introducing them
- Computes the score of a VN z by aggregating the connection scores of **all node-VN pairs involving z**
 - $s_z^{(l)} = \text{logmeanexp} \left(\left\{ \bar{s}_{vz}^{(l)} \mid v \in \mathcal{V}^{(l-1)} \right\} \right)$
- Selects z such that $\text{sigmoid} \left(s_z^{(l)} \right) \geq 0.5$



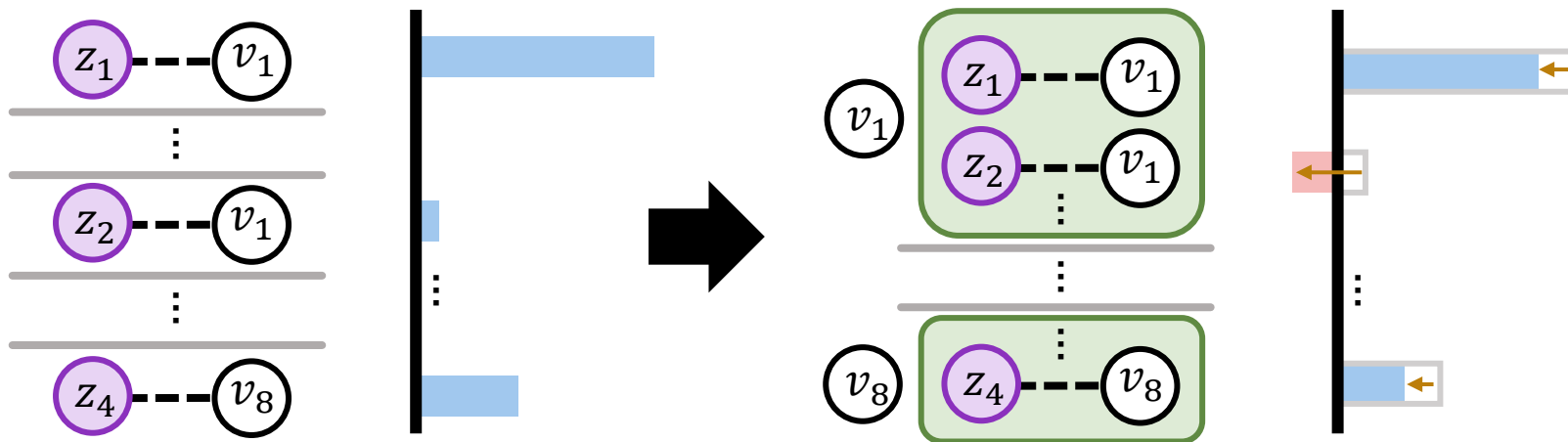
03 Connecting Nodes and VNs

- An edge between a node $v \in \mathcal{V}^{(l-1)}$ and a selected VN z is formed by considering the preference of the node v and the VN $z \in \hat{\mathcal{V}}^{(l)}$
 - **VN-level preference score** $\tilde{s}_{vz|z}^{(l)} = s_{vz}^{(l)} + \alpha \cdot \text{logsoftmax}(\mathcal{S}_{:z}^{(l)})[s_{vz}^{(l)}]$, $\mathcal{S}_{:z}^{(l)} = \{s_{v'z}^{(l)} | v' \in \mathcal{V}^{(l-1)}\}$
 - How significant is v for z **compared to all nodes in $\mathcal{V}^{(l-1)}$** ?



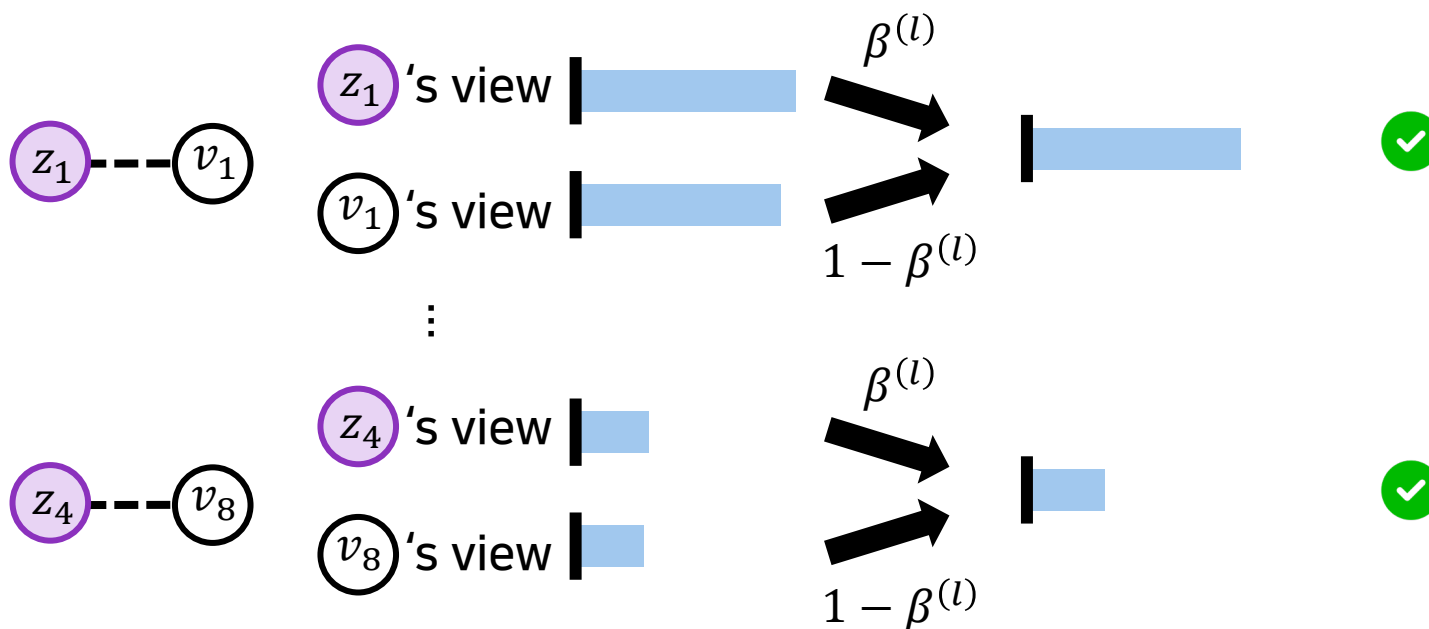
03 Connecting Nodes and VNs

- An edge between a node $v \in \mathcal{V}^{(l-1)}$ and a selected VN z is formed by considering the preference of the node v and the VN $z \in \hat{\mathcal{V}}^{(l)}$
- **Node-level preference score** $\tilde{s}_{vz|v}^{(l)} = s_{vz}^{(l)} + \alpha \cdot \text{logsoftmax}(\mathcal{S}_v^{(l)})[s_{vz}^{(l)}]$, $\mathcal{S}_{:z}^{(l)} = \{s_{vz'}^{(l)} | z' \in \hat{\mathcal{V}}^{(l)}\}$
 - How significant is z for v compared to all VNs in $\hat{\mathcal{V}}^{(l)}$?



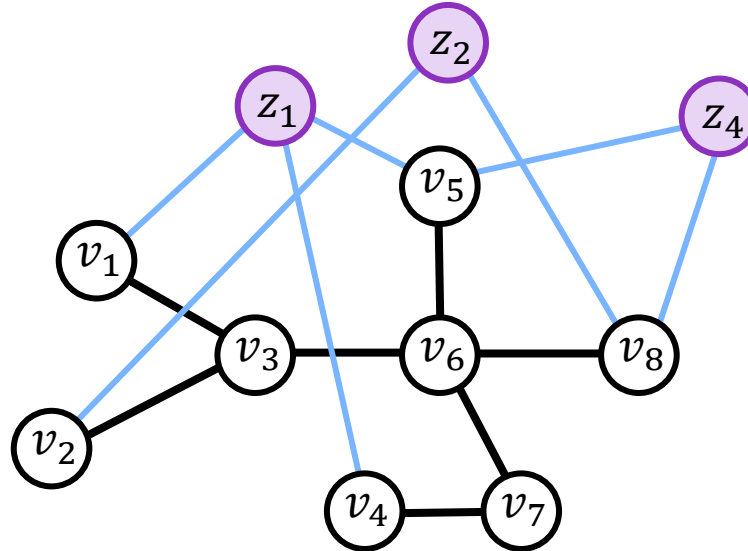
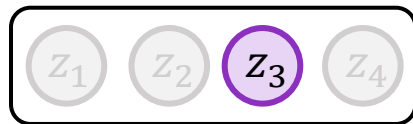
03 Connecting Nodes and VNs

- An edge between a node $v \in \mathcal{V}^{(l-1)}$ and a selected VN z is formed by considering the **preference of the node v** and the **VN $z \in \hat{\mathcal{V}}^{(l)}$**
 - Final edge score** $\tilde{s}_{vz}^{(l)} = \beta^{(l)} \cdot \tilde{s}_{vz|z}^{(l)} + (1 - \beta^{(l)}) \cdot \tilde{s}_{vz|v}^{(l)}$
 - Selected if $\text{sigmoid}(\tilde{s}_{vz}^{(l)}) \geq 0.5$
 - Note: Every selected VN is guaranteed to be connected to at least one node



03 Connecting Nodes and VNs

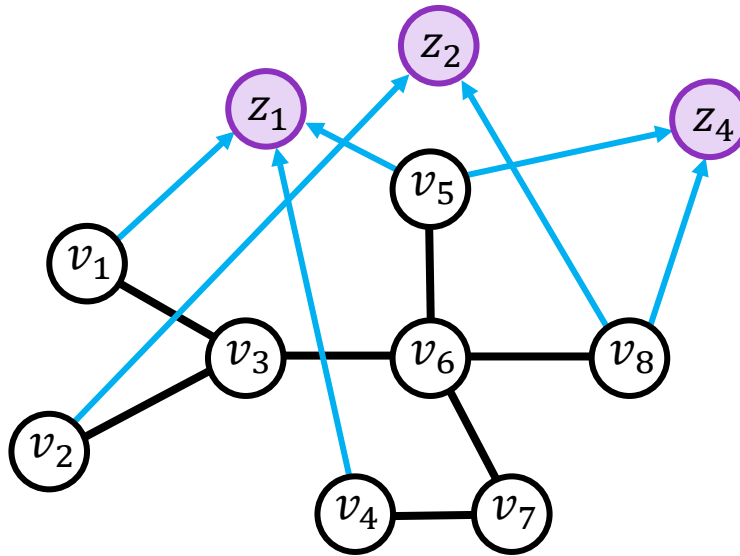
- An edge between a node $v \in \mathcal{V}^{(l-1)}$ and a selected VN z is formed by considering the preference of the node v and the VN $z \in \hat{\mathcal{V}}^{(l)}$
 - Final edge score $\tilde{s}_{vz}^{(l)} = \beta^{(l)} \cdot \tilde{s}_{vz|z}^{(l)} + (1 - \beta^{(l)}) \cdot \tilde{s}_{vz|v}^{(l)}$
 - Selected if $\text{sigmoid}(\tilde{s}_{vz}^{(l)}) \geq 0.5$
 - Note: Every selected VN is guaranteed to be connected to at least one node



03 Computing VNs' Representations

- Computed by **aggregating** the representations of **its connected nodes**

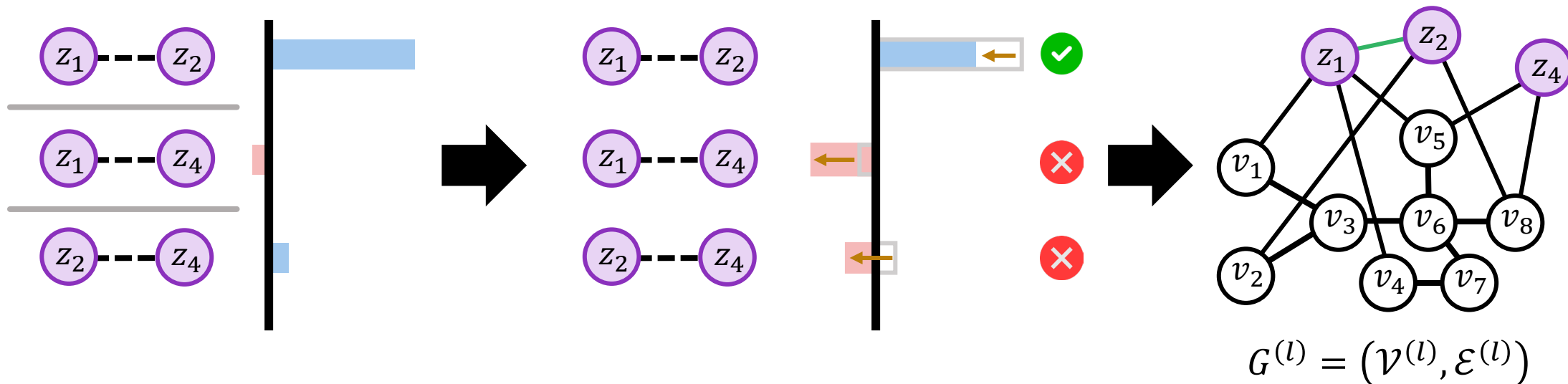
- $\mathbf{x}_z^{(l-1)} = \boldsymbol{\gamma}^{(l)} \odot \mathbf{q}_z^{(l)} + (\mathbf{1} - \boldsymbol{\gamma}^{(l)}) \odot \left(\sum_{v \in \mathcal{N}_z^{(l)}} \frac{p_{vz}^{(l)} \mathbf{x}_v^{(l-1)}}{c_z^{(l)}} \right), p_{vz}^{(l)} = \text{sigmoid}(\tilde{s}_{vz}^{(l)})$
 - $c_z^{(l)}$ is a constant that determines the aggregation type



03

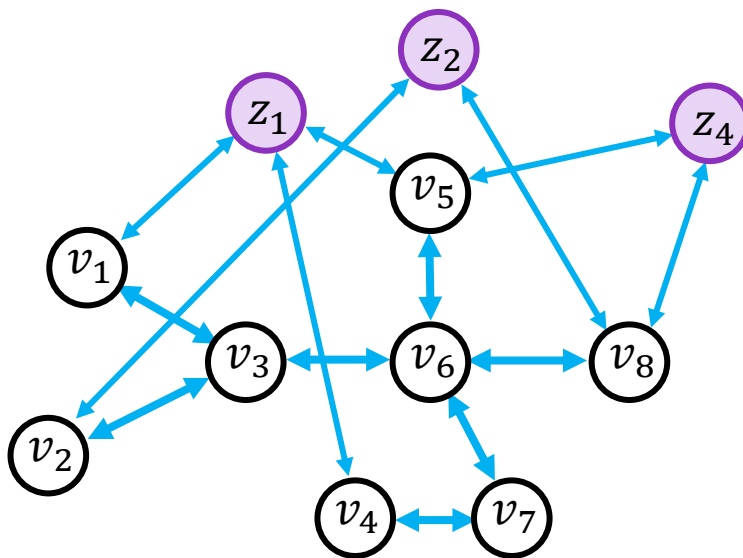
Selective Connections between VNs

- Edges between the selected VNs are formed by **considering both VNs' viewpoints**
 - Relevance score $s_{zu}^{(l)} = \text{MLP}^{(l)}(\mathbf{x}_z^{(l-1)}) \cdot \text{MLP}^{(l)}(\mathbf{x}_u^{(l-1)}) / \sqrt{d_{\text{dot}}}$
 - Both VNs' viewpoints
 - $\bar{s}_{zu|z}^{(l)} = s_{zu}^{(l)} + \alpha \cdot \text{logsoftmax}\left(\left\{s_{zu'}^{(l)} \mid u' \in \hat{\mathcal{V}}^{(l)} \setminus \{z\}\right\}\right) [s_{zu}^{(l)}]$, $\bar{s}_{zu|u}^{(l)} = s_{zu}^{(l)} + \alpha \cdot \text{logsoftmax}\left(\left\{s_{z'u}^{(l)} \mid z' \in \hat{\mathcal{V}}^{(l)} \setminus \{u\}\right\}\right) [s_{zu}^{(l)}]$
 - Final edge score $\bar{s}_{zu}^{(l)} = \frac{1}{2}(\bar{s}_{zu|z}^{(l)} + \bar{s}_{zu|u}^{(l)})$



Dynamic Message Passing with Adaptively Introduced Virtual Nodes

- An MPNN layer computes representation at the l -th layer using the graph $G^{(l)}$
 - The original graph $G^{(0)}$ **dynamically** and **iteratively** changes to $G^{(l)}$ for $l = 1, \dots, L$
- MAVN **only utilizes node representations** and does not assume any specific MPNN architecture
 - Architecture-agnostic and **can be seamlessly applied to any MPNNs**



- A single-layer MAVN **is capable of constructing any message passing paths that involve VNs**
 - Thus, MAVN can **simulate any message passing construction strategies** that connect VNs and nodes, including those employed by existing VN-based methods

Theorem 1. Given a graph $G = (\mathcal{V}, \mathcal{E})$ with K sets of nodes $\mathcal{V}_1, \mathcal{V}_2, \dots, \mathcal{V}_K \subseteq \mathcal{V}$, if node representations are uniquely distinguishable (i.e., $x_u = x_v \Leftrightarrow u = v, \forall u, v \in \mathcal{V}$), there exists a parameter configuration of a single-layer MAVN that introduces K virtual nodes $z_1, z_2, \dots, z_K$ into G , where each z_i is connected to a node $v \in \mathcal{V}$ if and only if $v \in \mathcal{V}_i$.

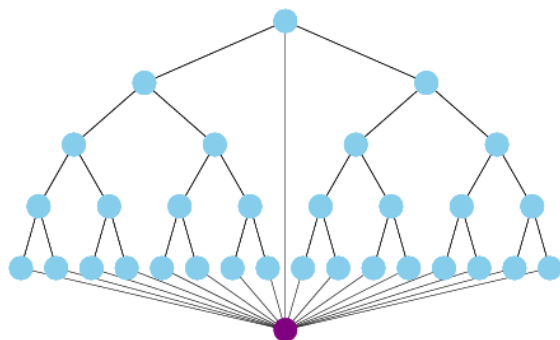
→ For any graph structure augmented by connecting VNs to nodes, there exists **a single-layer MAVN** that constructs the same structure

Corollary 1. Any graph structure obtained by connecting VNs to nodes in the original graph, such as those produced by IPR-MPNN or N², can be simulated by a single-layer MAVN

→ MAVN can simulate graph structures of **existing VN-based methods**

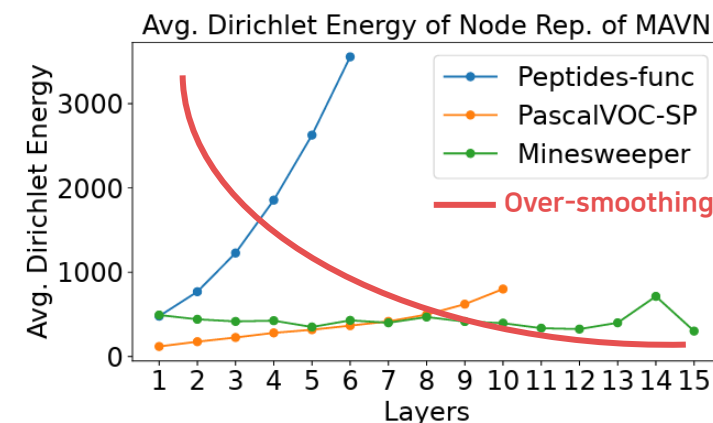
04 Mitigating Under-reaching and Over-smoothing

- Under-reaching: experiment on a synthetic dataset
 - Predict the number of leaf nodes labeled '1' using only the root node's representation in a single-layer setup



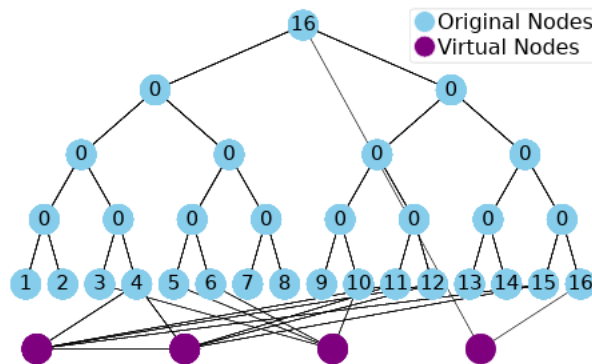
depth	GCN	MAVN-GCN
4	0.14	1.00
5	0.11	1.00
6	0.08	1.00

- Over-smoothing: Experiments on real-world datasets
 - Measures the **Dirichlet energy** of node representations at each layer
 - Over-smoothing is defined as **an exponential decrease in the Dirichlet energy** with increasing layers



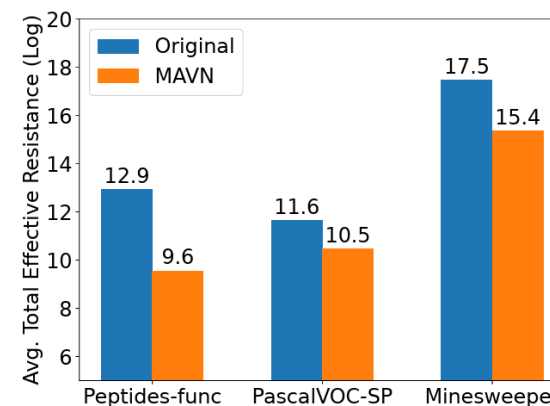
04 Mitigating Over-squashing

- Experiment on a synthetic dataset
 - Predict the value label of a leaf node that shares the same key label as the root node, using the representation of the root node



depth	GCN	MAVN-GCN
4	0.54	1.00
5	0.22	1.00
6	0.10	1.00

- Experiments on real-world datasets
 - Measures **effective graph resistance**, an upper bound on the extent of over-squashing

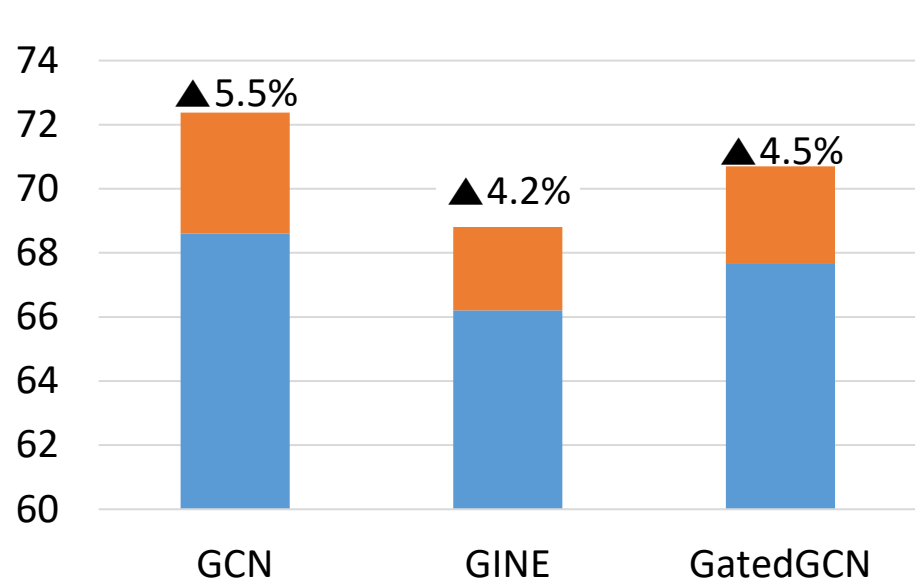


05 Experimental Results

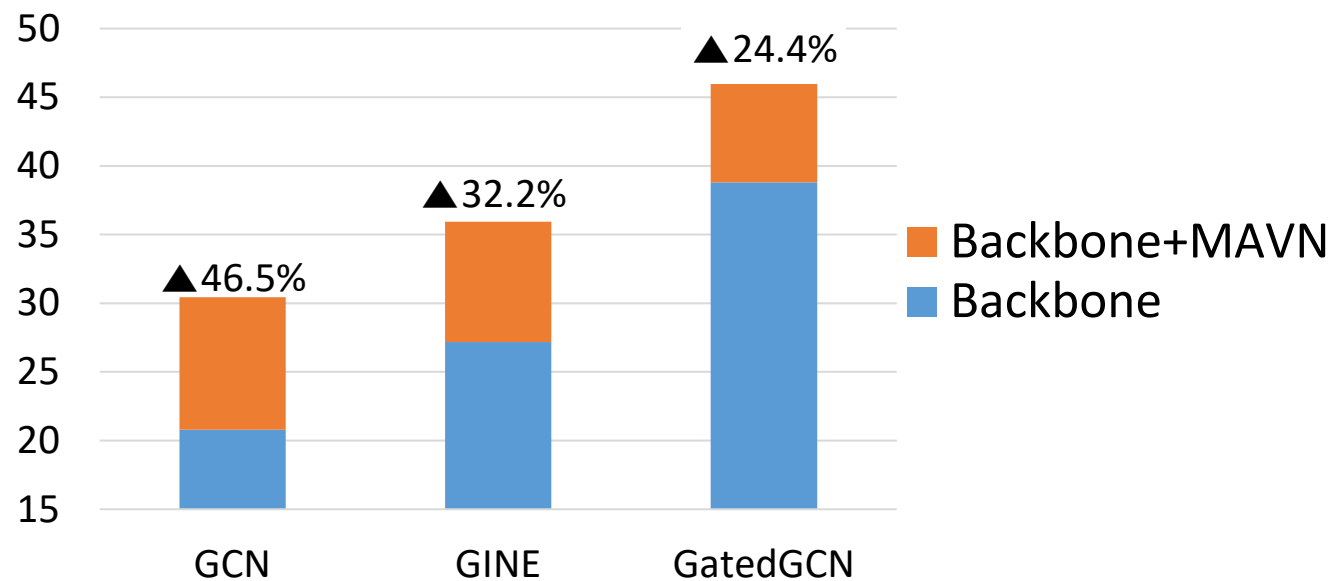
- Datasets
 - 4 datasets from [Long Range Graph Benchmark](#) (LRGB)
 - 5 **heterophilic graph datasets**
- Backbones
 - LRGB: GCN, GINE, GatedGCN
 - Heterophilic graph datasets: GCN, GAT, GraphSAGE

05 Performance on LRGB

- An overall average improvement of 17.1% across 4 datasets



Peptides-func

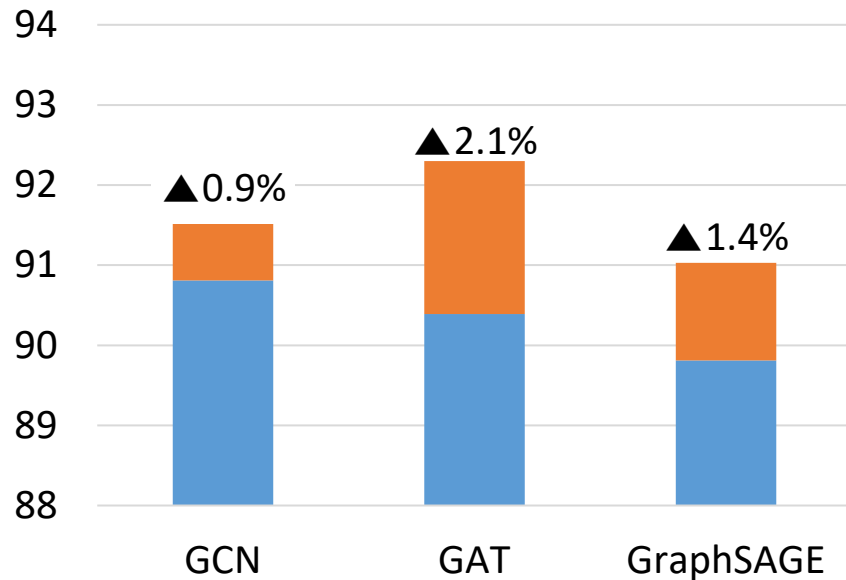


PascalVOC-SP

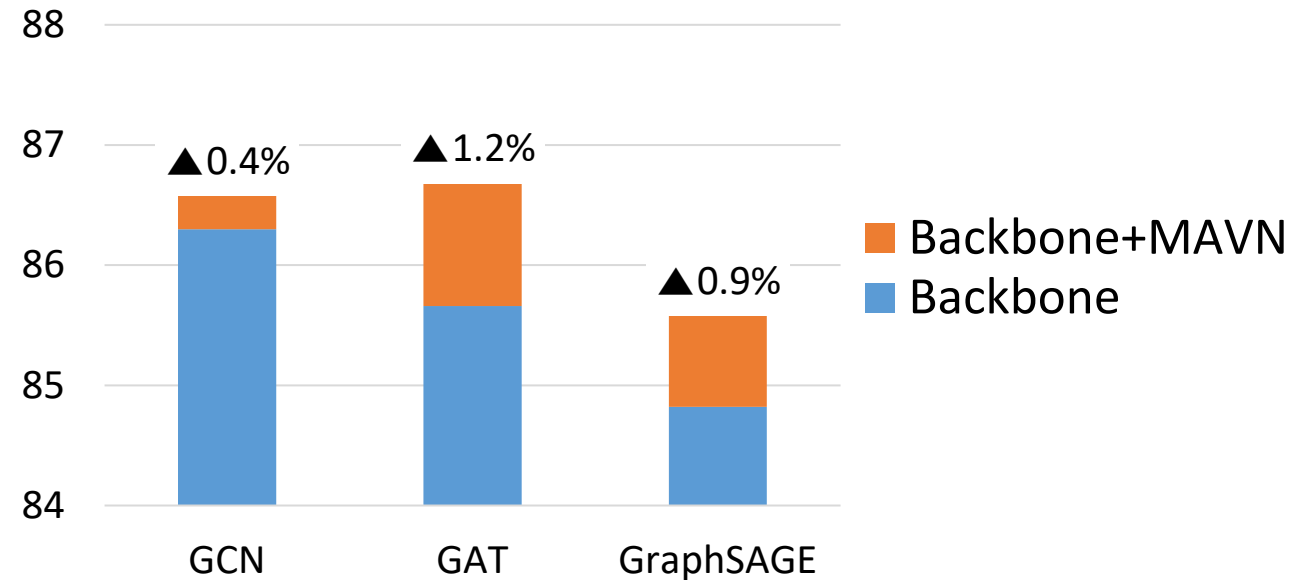
05

Performance on Heterophilic Graphs

- An overall average improvement of 0.9% across 5 graphs



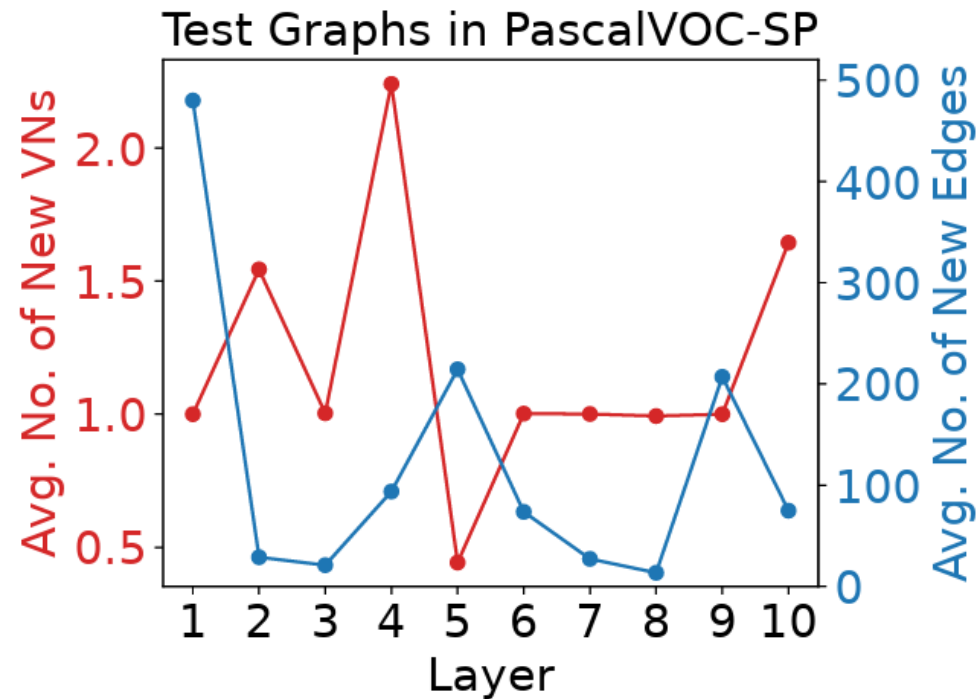
roman-empire



tolokers

05 Case Study

- Both the number of new VNs and their connectivity patterns vary across layers
 - Indicates that MAVN **adaptively introduces VNs** and **flexibly forms connections**



05 Conclusion

- Propose MAVN, a novel MPNN framework that **learns to determine when and where to introduce and connect VNs**
 - Allows unconstrained and flexible connectivity between nodes and VNs
- MAVN **makes no assumptions about the backbone architecture** and **relies solely on node representations** to dynamically update message passing paths
 - Agnostic to the choice of backbone MPNNs
- Theoretically, for **any node-VN connectivity pattern**, there exists a set of parameters of MAVN that can simulate the pattern
- Experimental results on nine datasets, each evaluated with three different backbone MPNNs, show that **MAVN consistently improves the backbone MPNNs' performance**

Thank You!

Our datasets and codes are available at:

<https://github.com/bdi-lab/MAVN>

You can find us at:

{jjlee98, jjwhang}@kaist.ac.kr

<https://bdi-lab.kaist.ac.kr>

