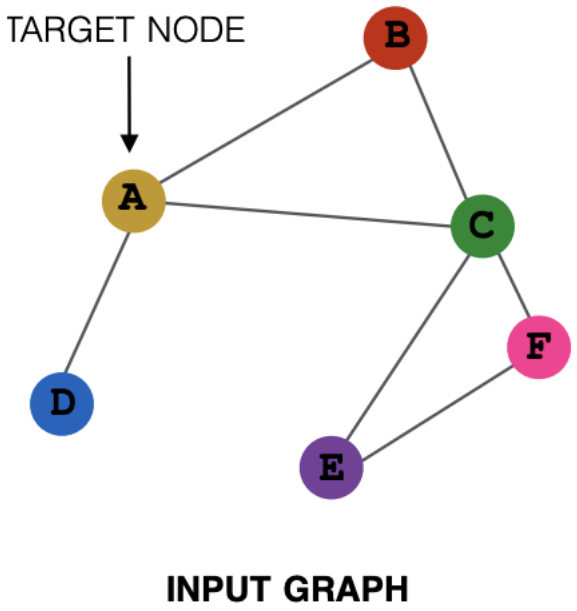
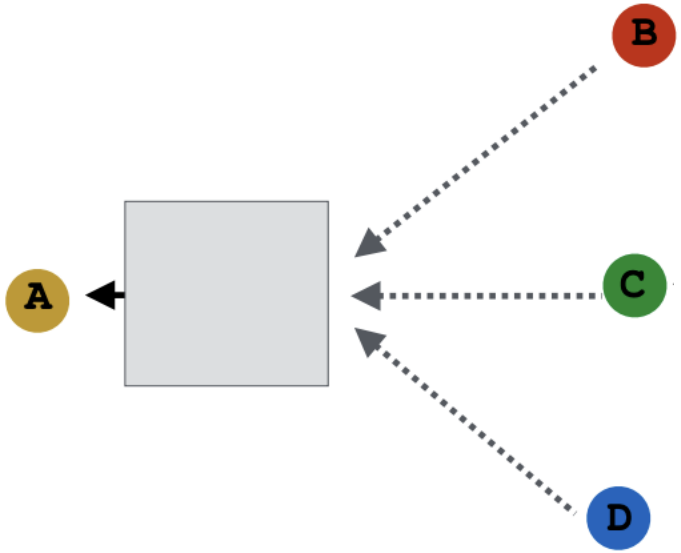
Over-smoothing of GNN

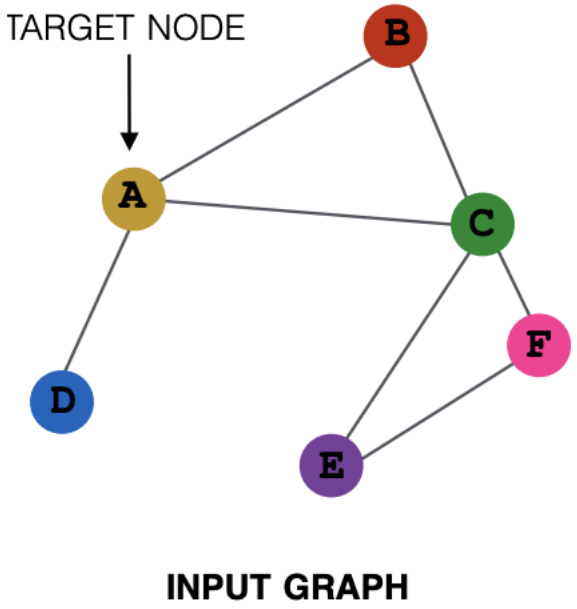
Neural Networks Design And Application

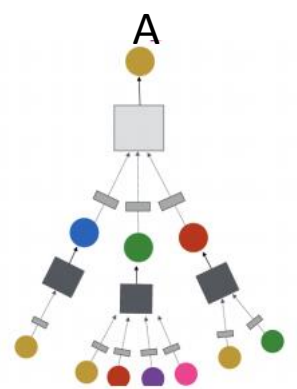
Graph networks: aggregate neighbors

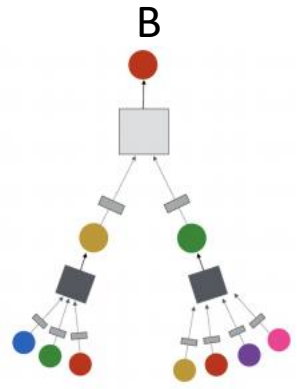


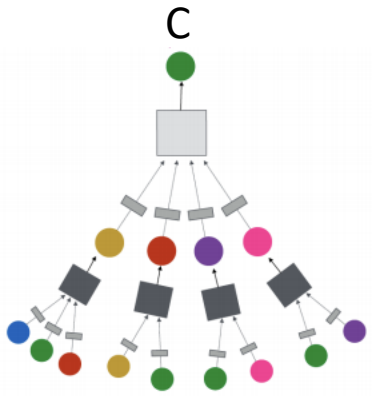


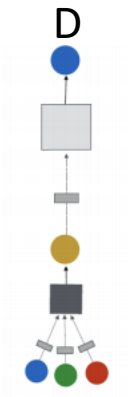
Graph networks: aggregate neighbors

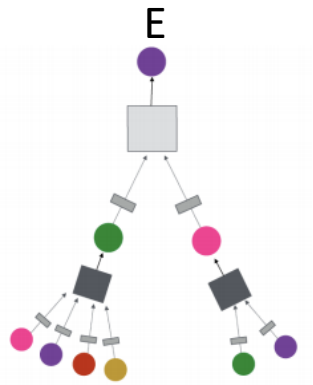


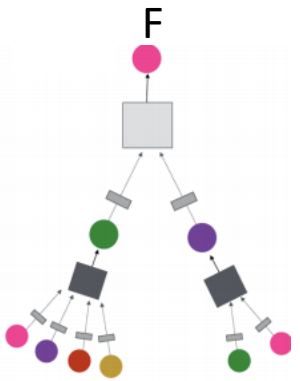




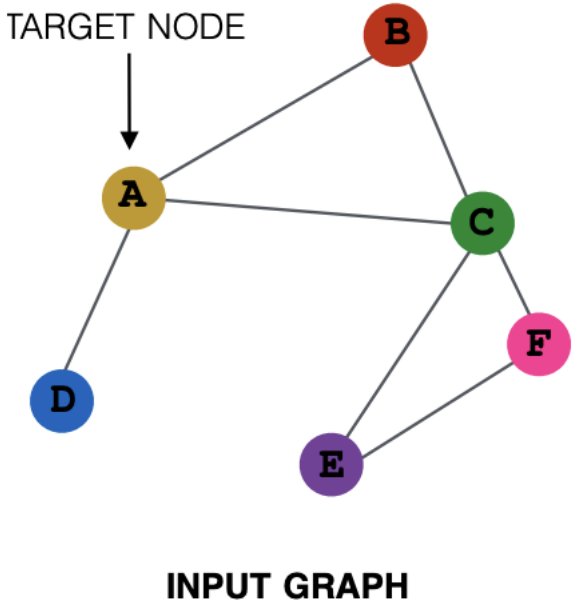




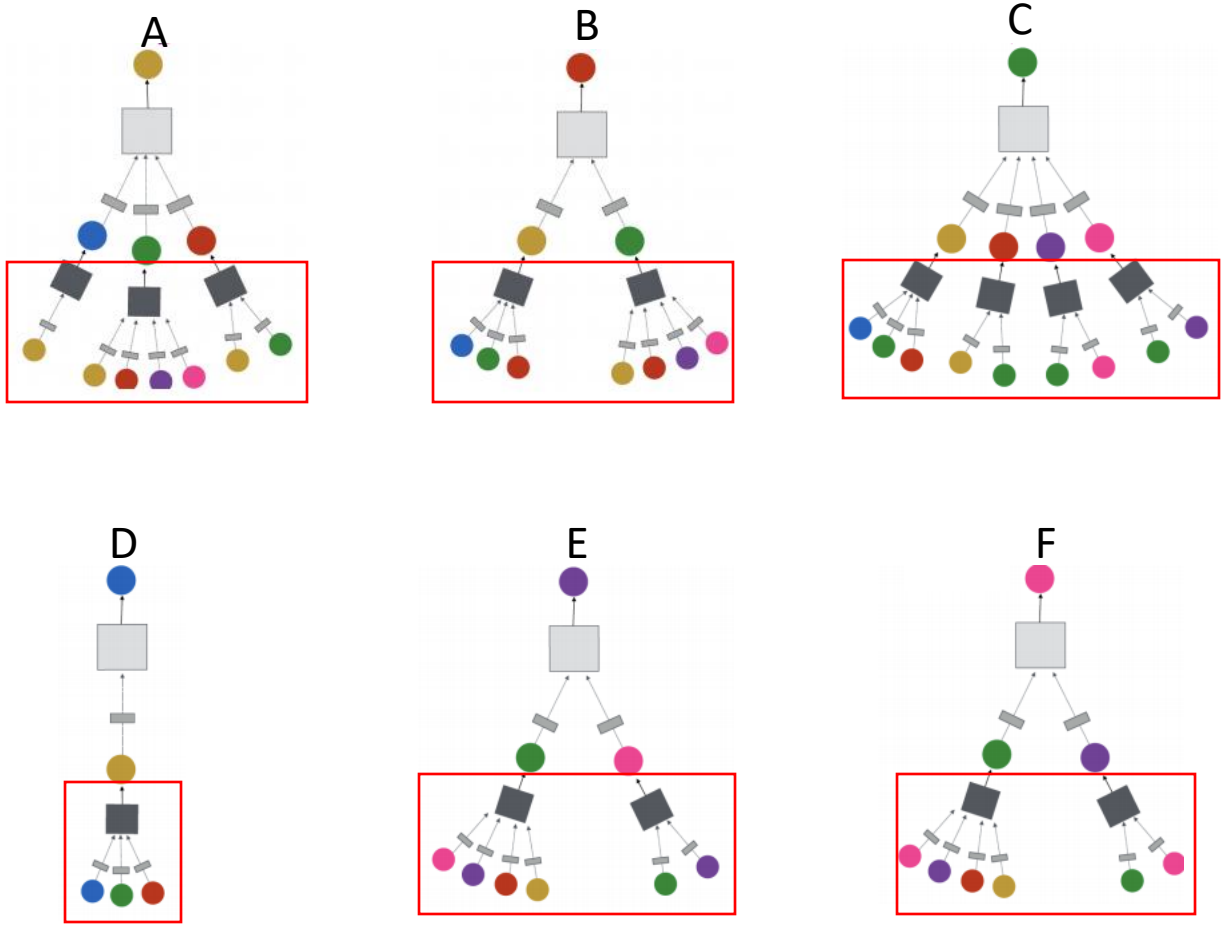




Graph networks: aggregate neighbors

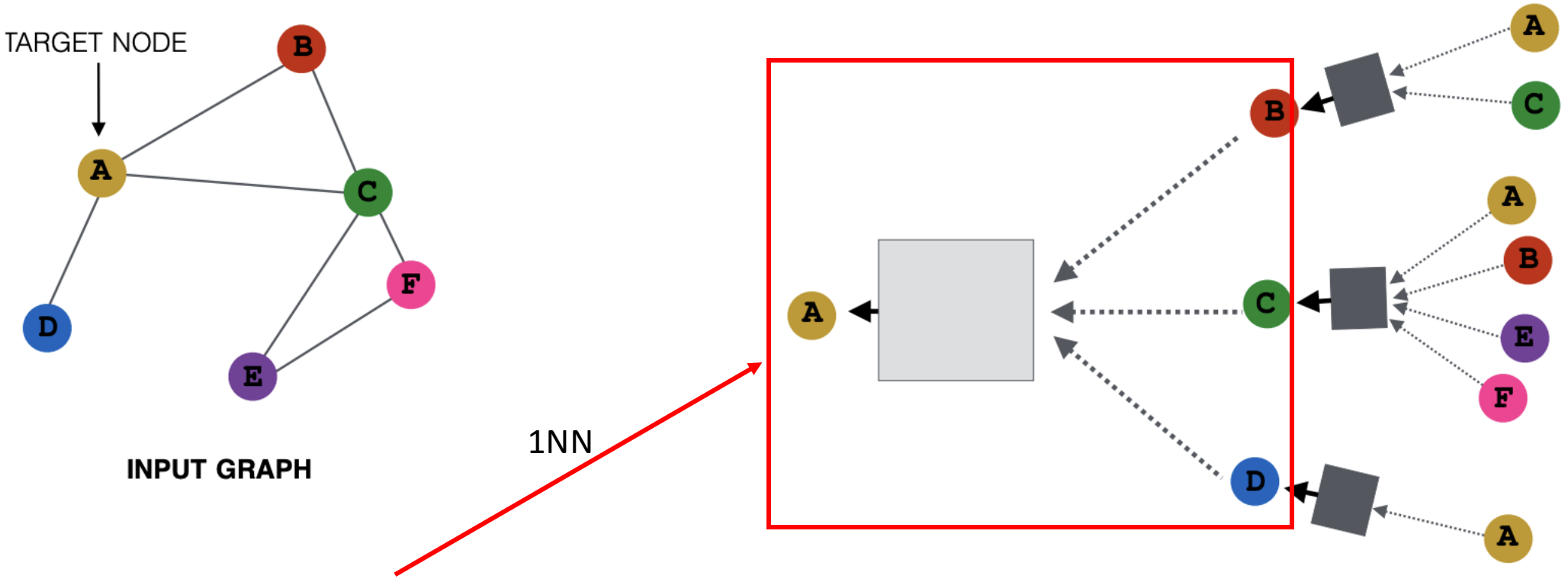


INPUT GRAPH



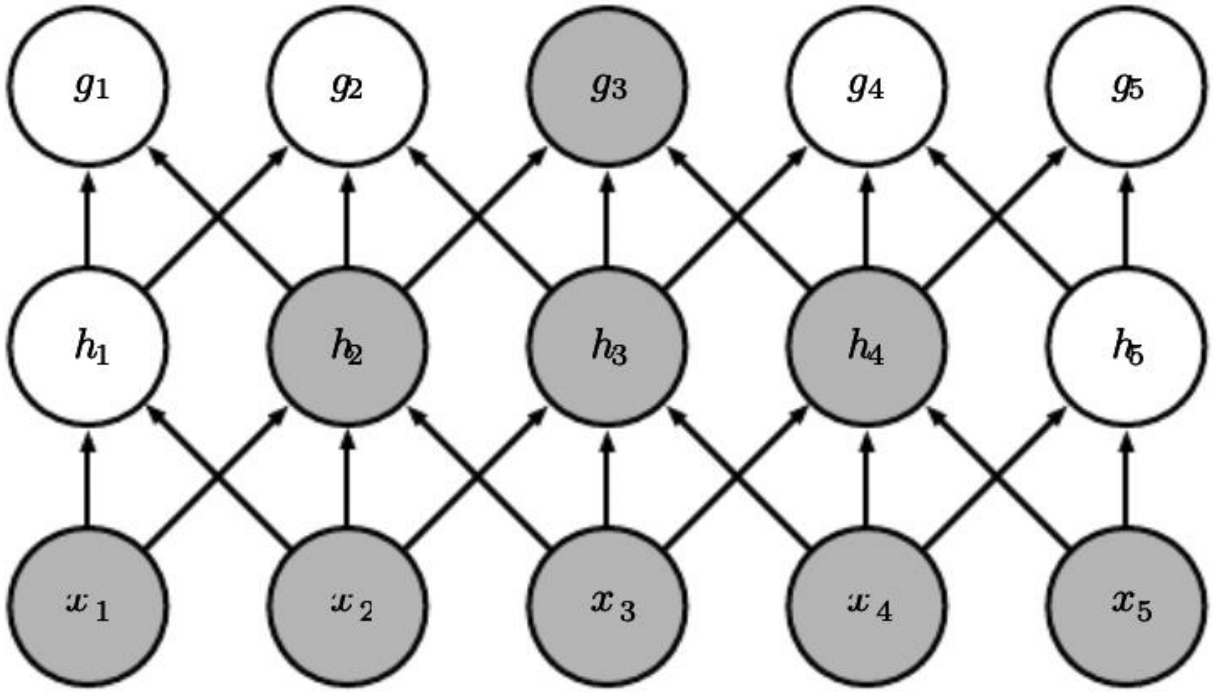
Two hops (two nearest neighbors, 2NN)

Graph networks: aggregate neighbors

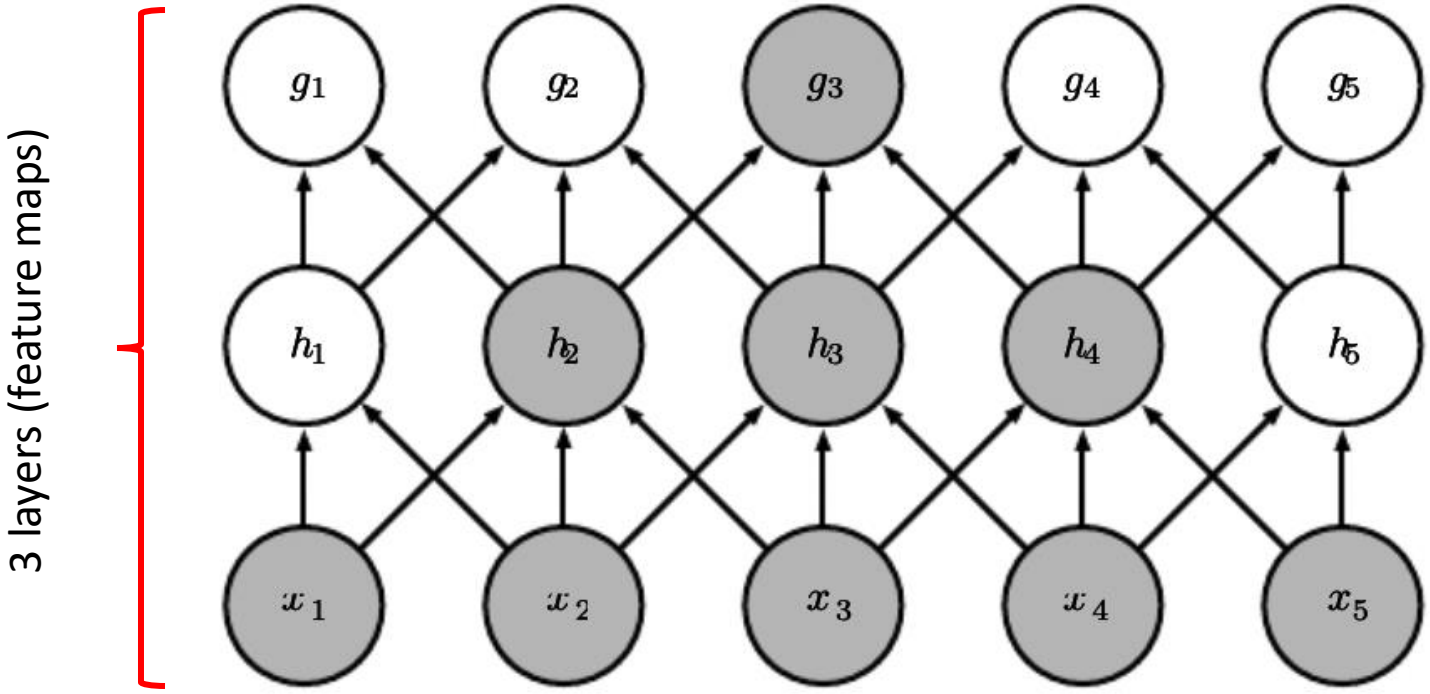


Q: can we add more NNs? → 3NN, 4NN, …
→ it will aggregate/cover all nodes in a graph

Recall: receptive field of CNN

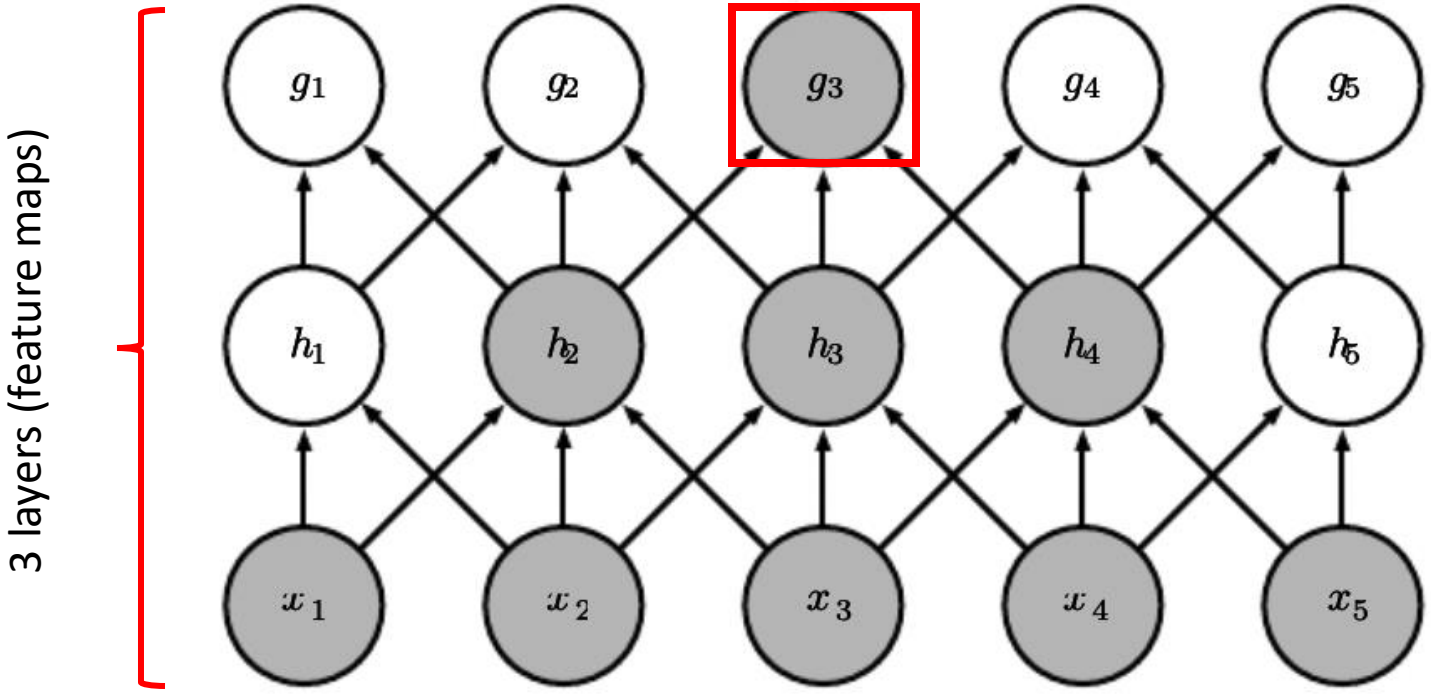


Recall: receptive field of CNN

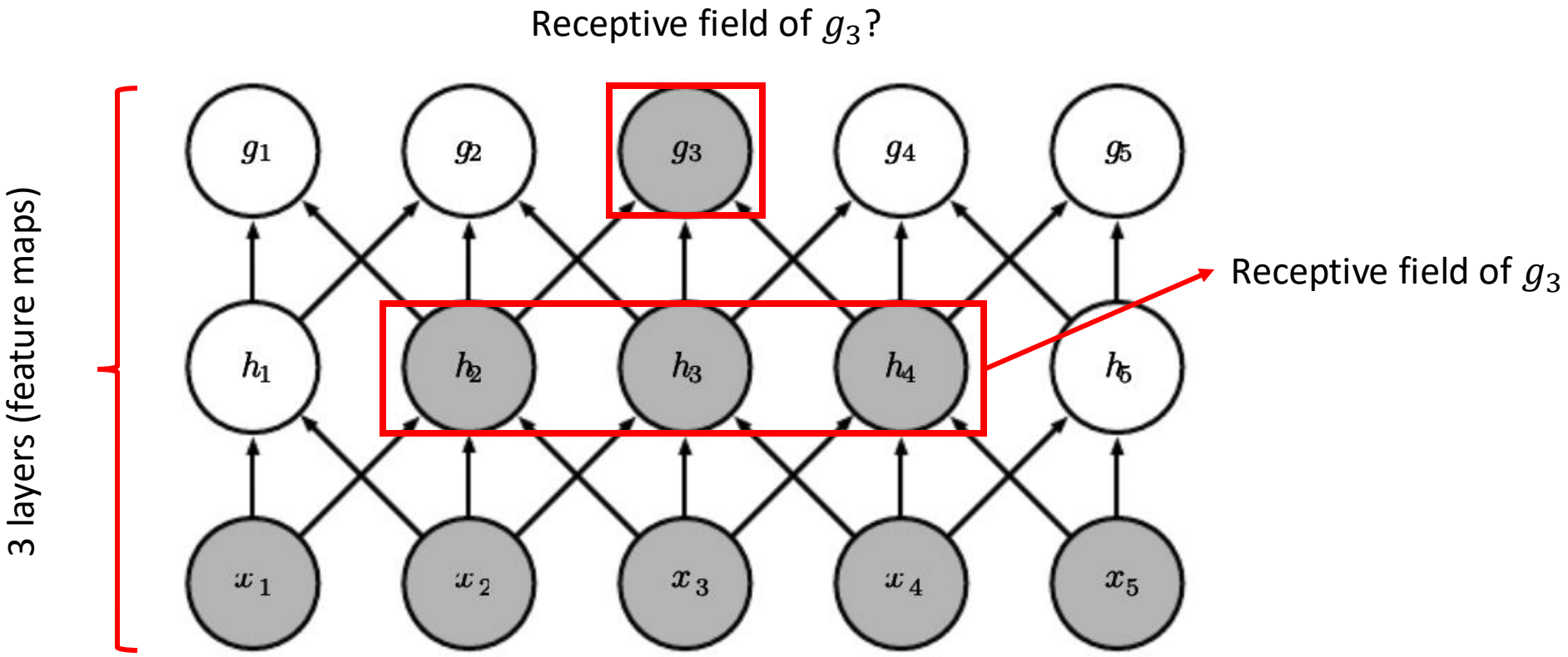


Recall: receptive field of CNN

Receptive field of g_3?

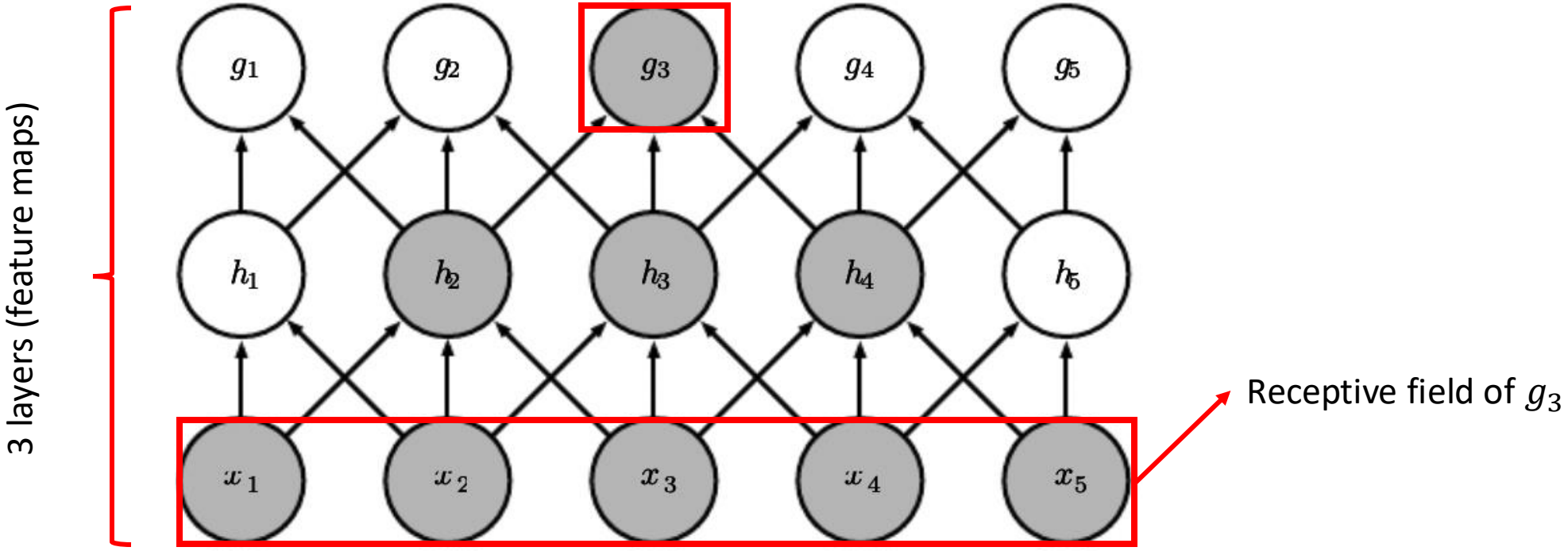


Recall: receptive field of CNN



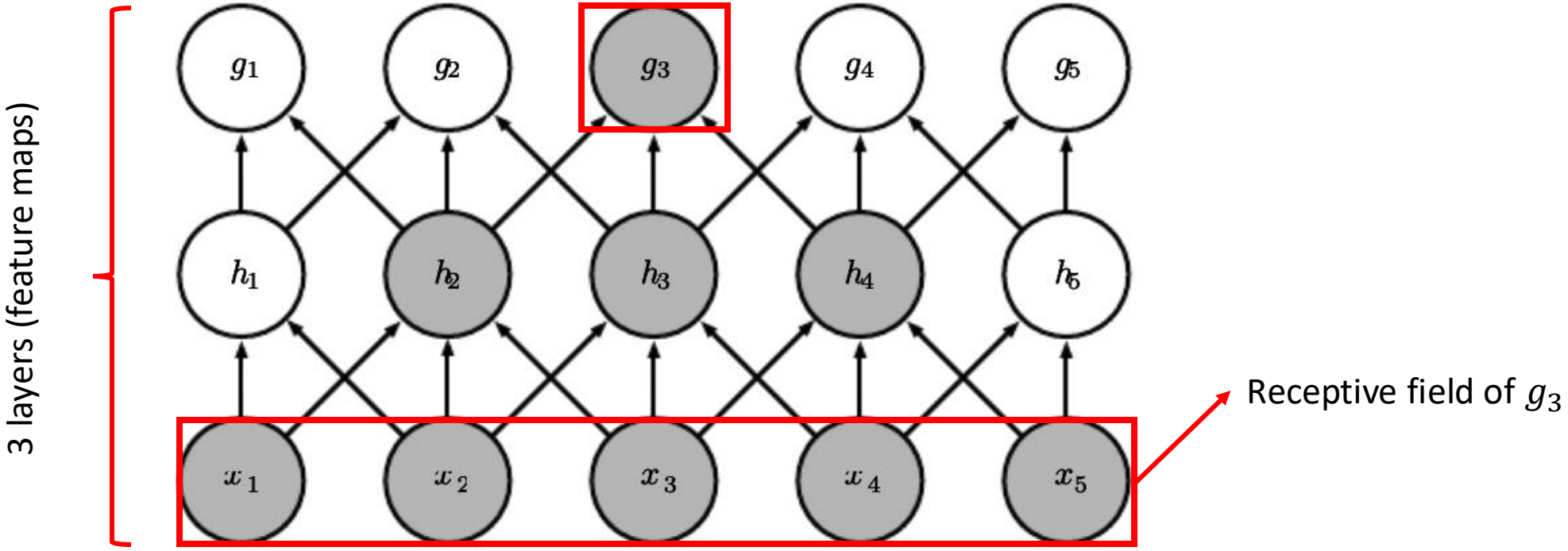
Recall: receptive field of CNN

Receptive field of g_3?



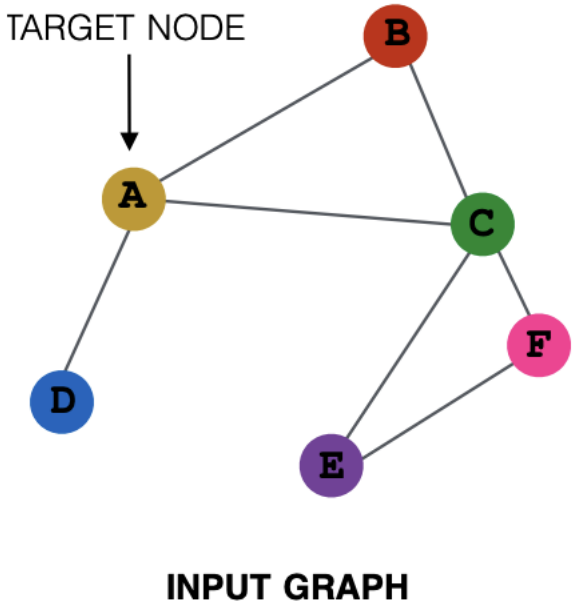
Recall: receptive field of CNN

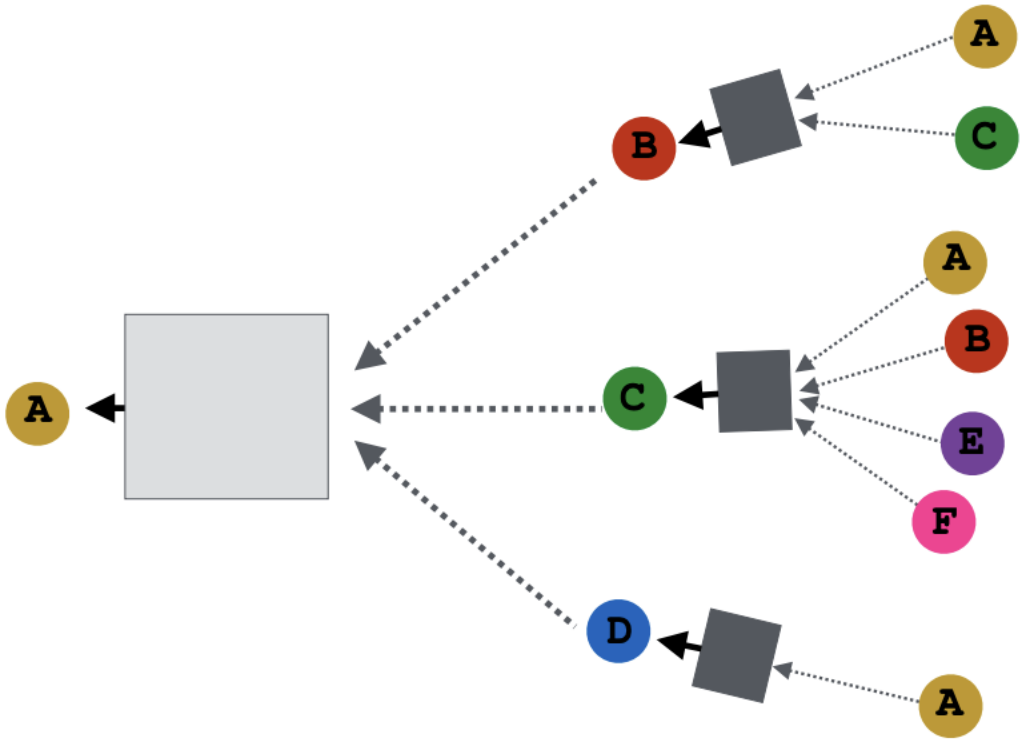
Receptive field of g_3?



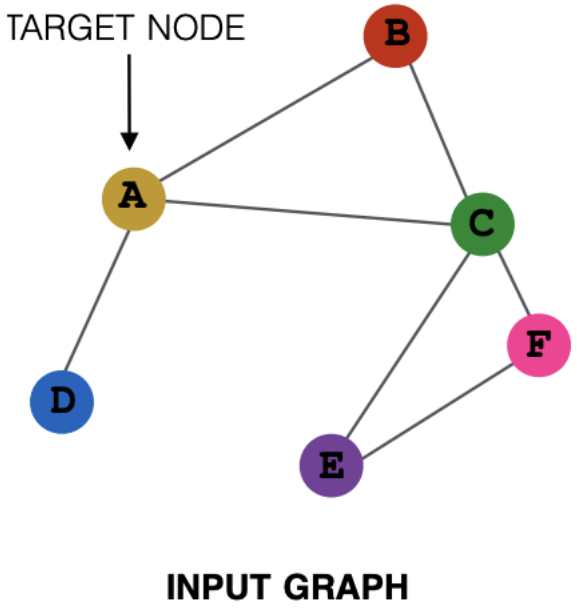
(input layer: pixel values → location information)

Receptive field of GCN

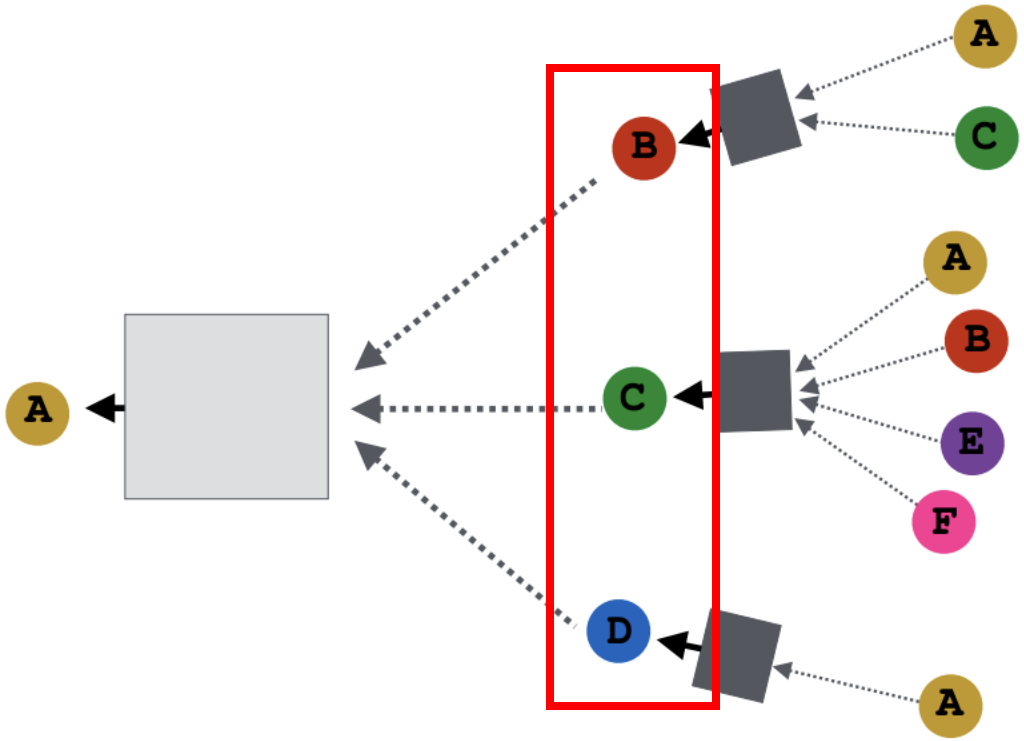




Receptive field of GCN

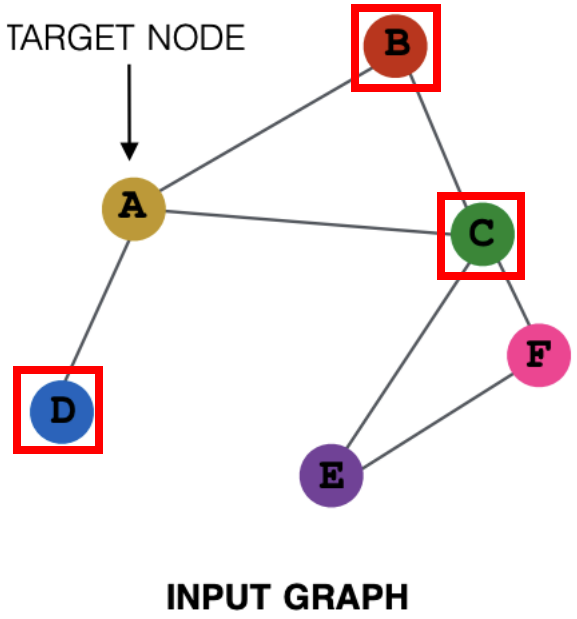


INPUT GRAPH

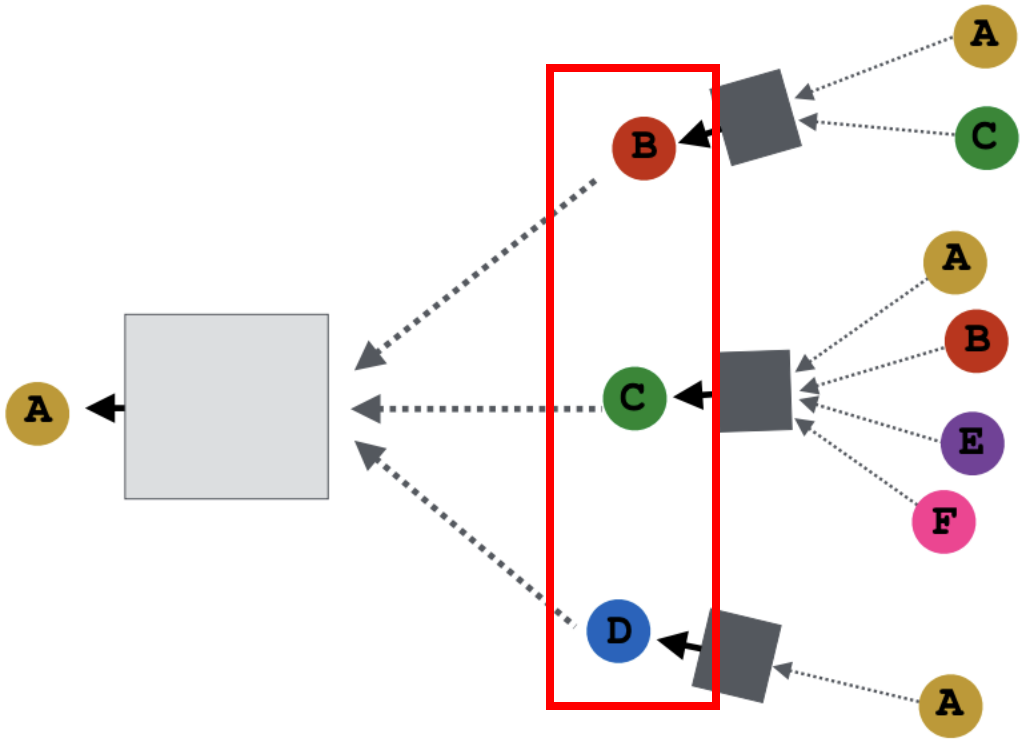


B, C, D

Receptive field of GCN

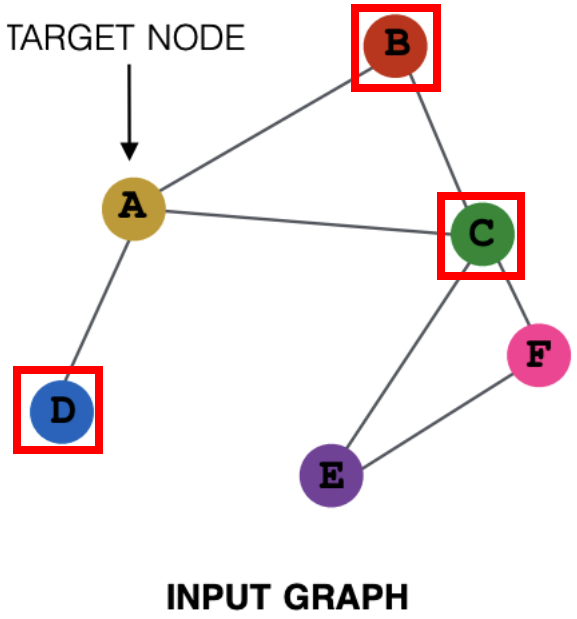


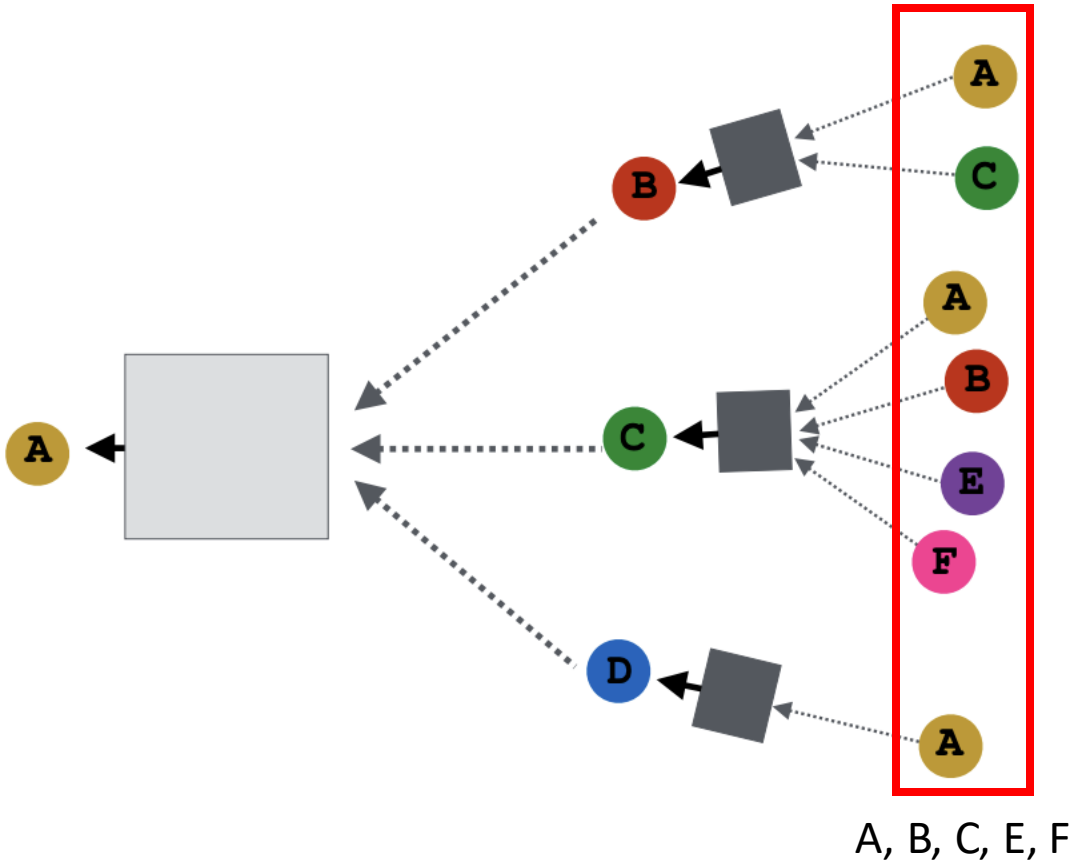
INPUT GRAPH



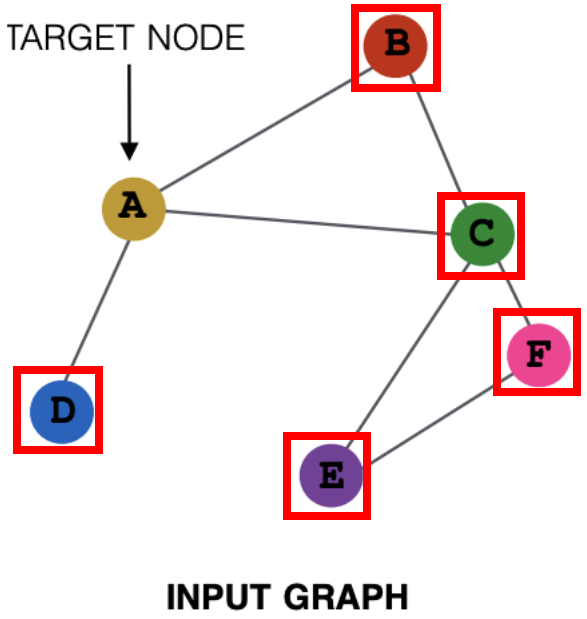
B, C, D

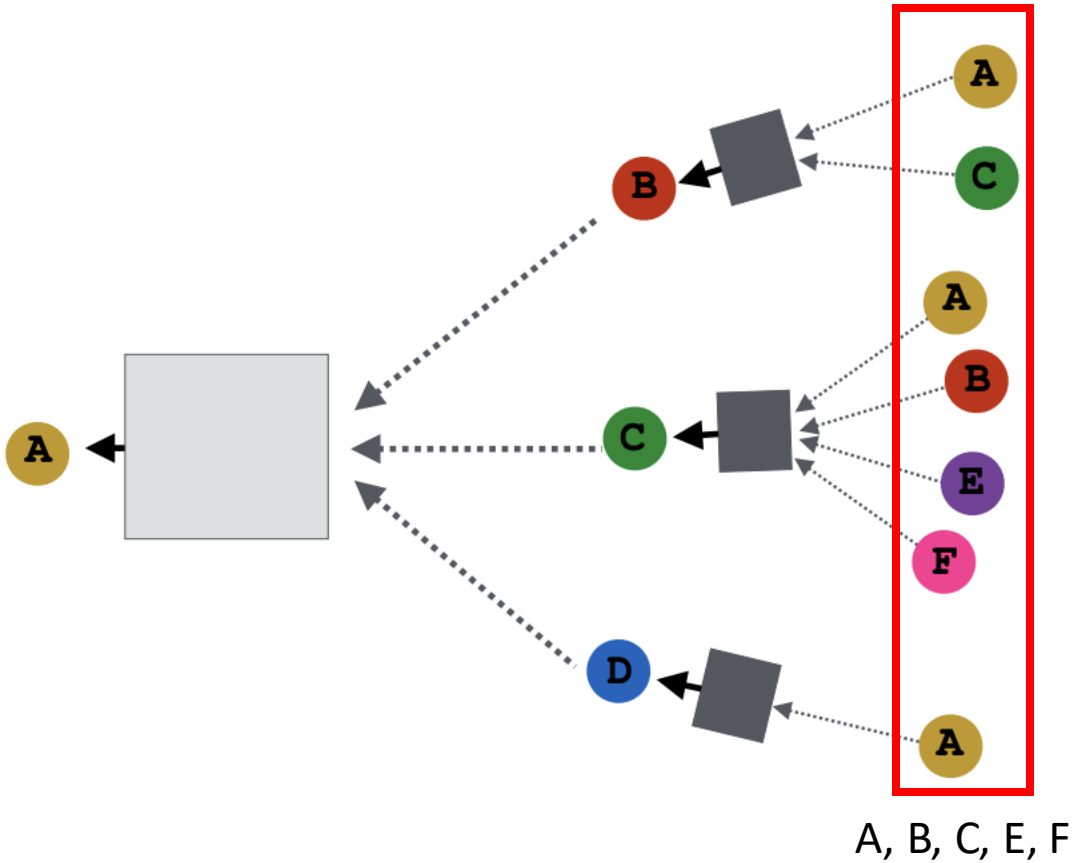
Receptive field of GCN



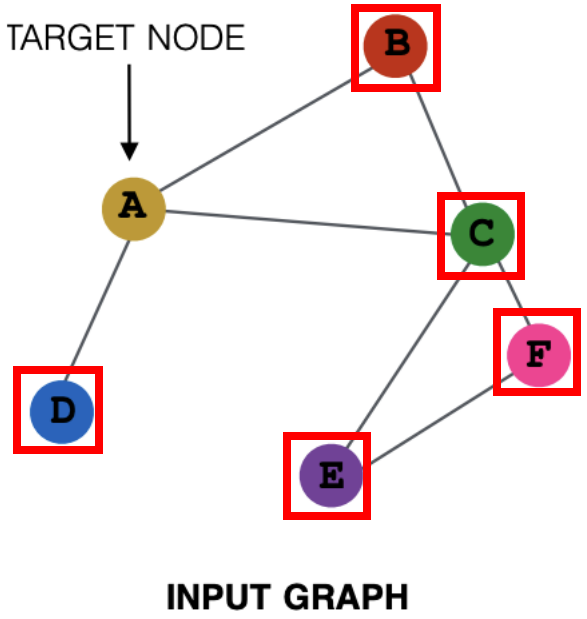


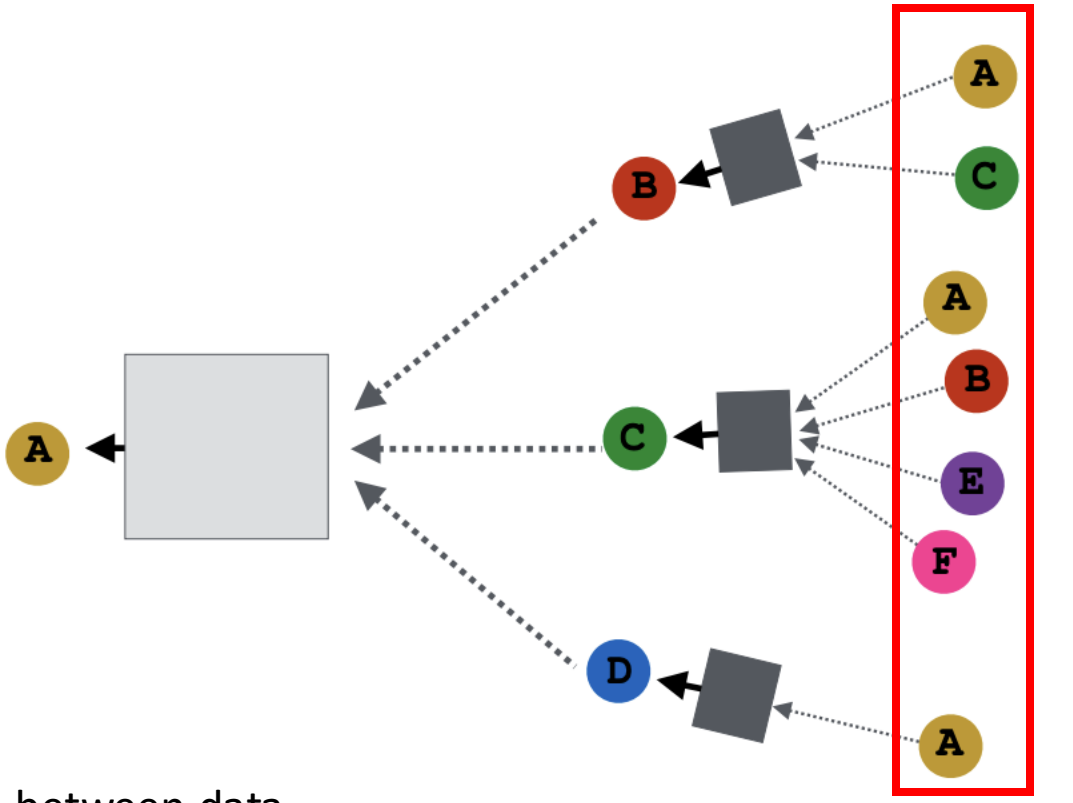
Receptive field of GCN





Receptive field of GCN



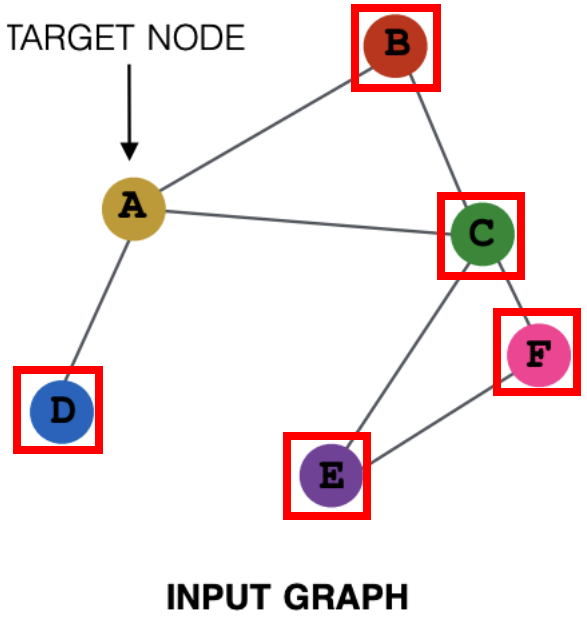


Cover the entire graph

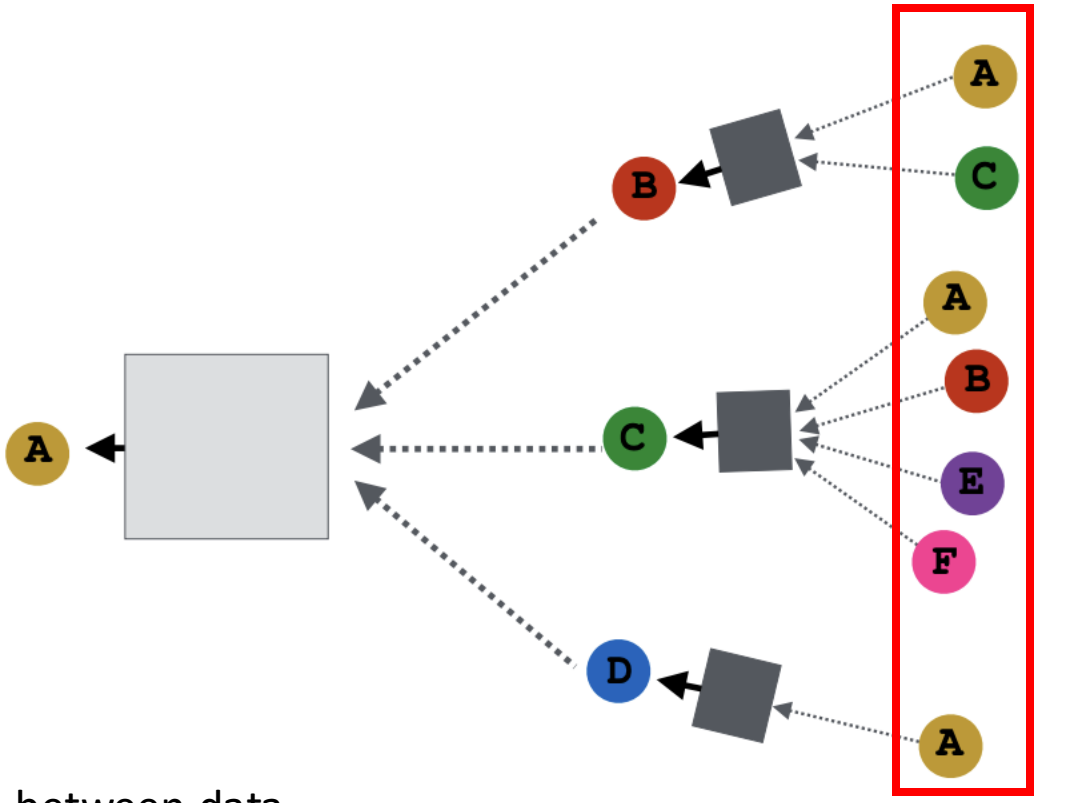
GCN: learn representation/feature capturing the correlation between data

A, B, C, E, F

Receptive field of GCN



INPUT GRAPH



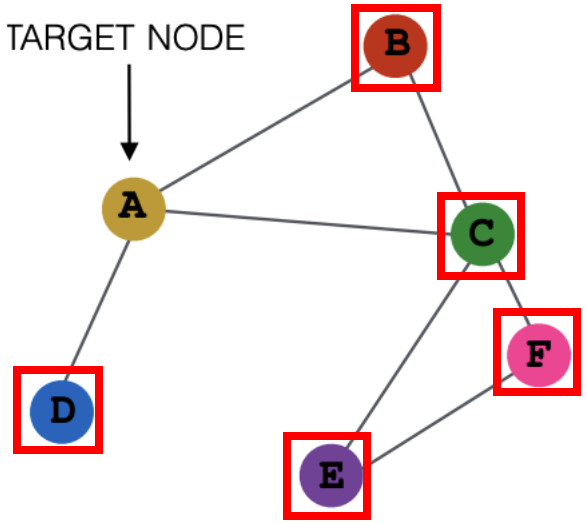
A, B, C, E, F

Cover the entire graph

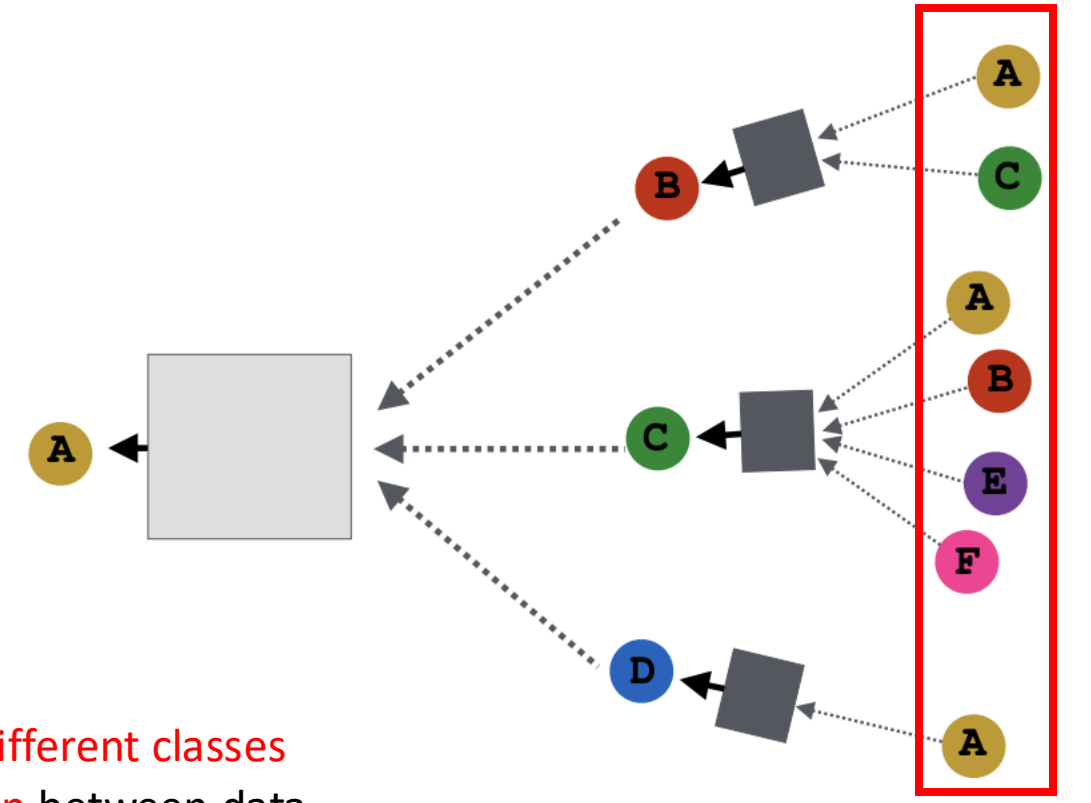
GCN: learn representation/feature capturing the correlation between data

Node classification: differentiate nodes from different classes

Receptive field of GCN



INPUT GRAPH



A, B, C, E, F

Cover the entire graph → correlation between different classes ?

GCN: learn representation/feature capturing the correlation between data

Node classification: differentiate nodes from different classes

Receptive field of GCN

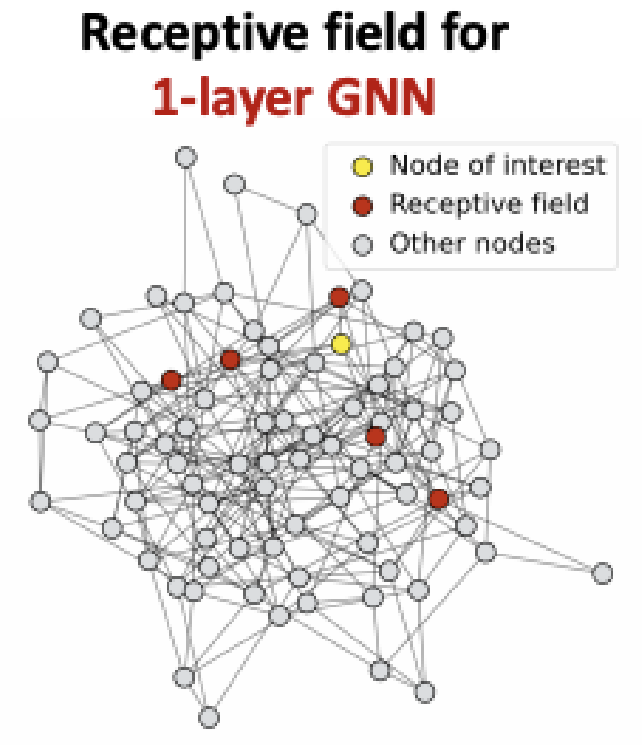
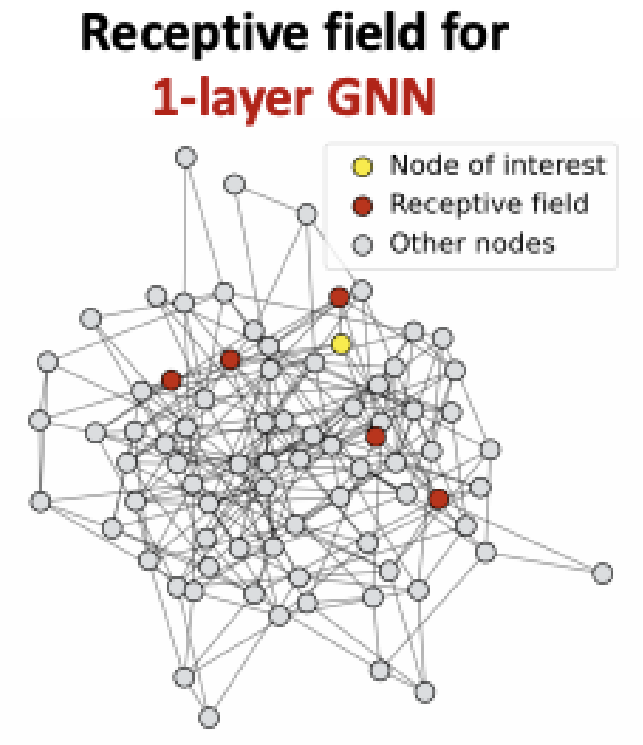


Image credit http://web.stanford.edu/class/cs224w/slides/07-GNN2.pdf

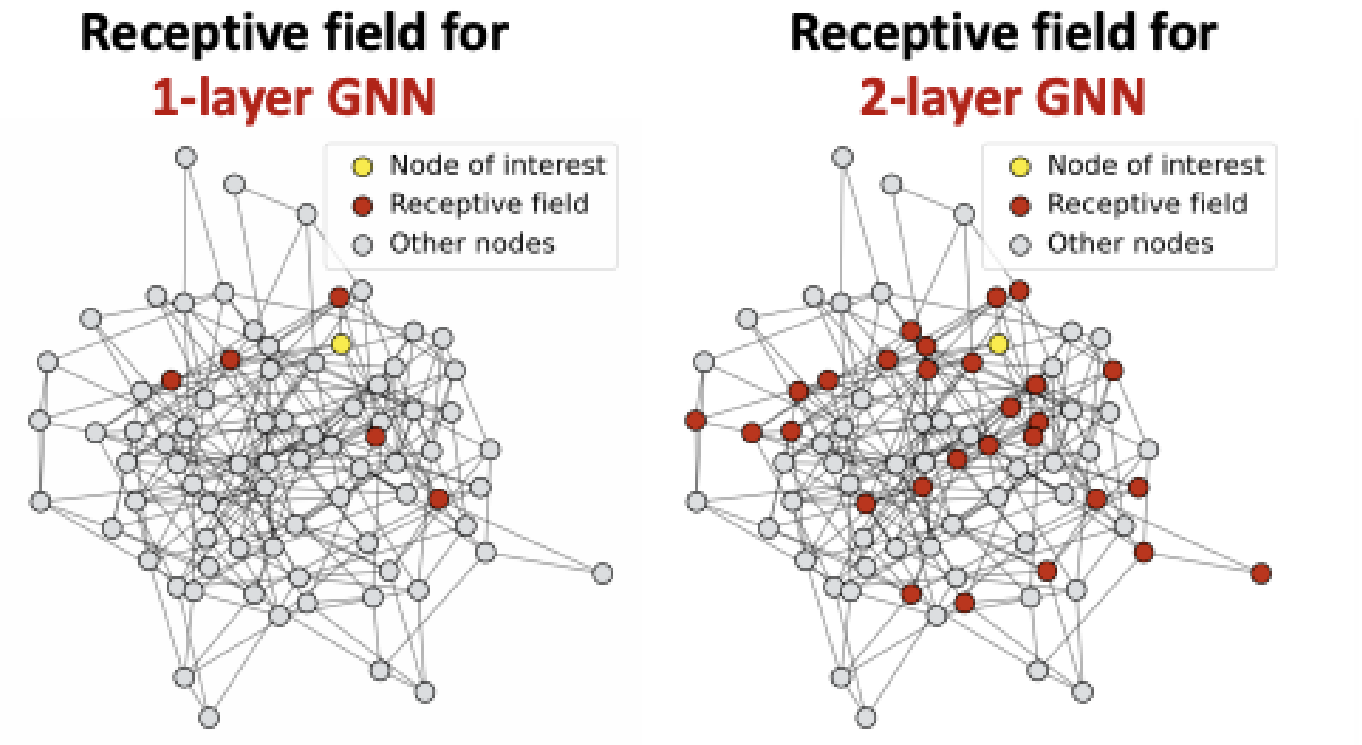
Receptive field of GCN



Only 5 nearest neighbors

Image credit http://web.stanford.edu/class/cs224w/slides/07-GNN2.pdf

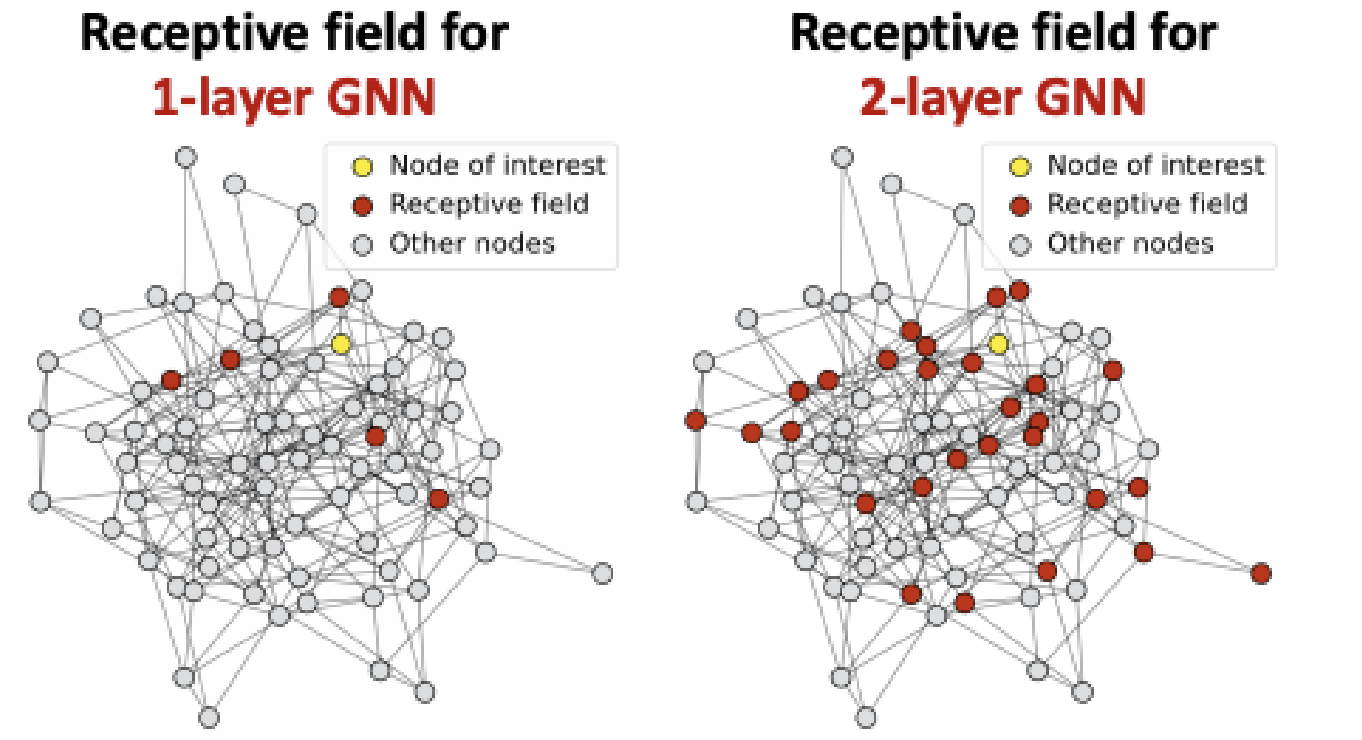
Receptive field of GCN



Only 5 nearest neighbors

Image credit http://web.stanford.edu/class/cs224w/slides/07-GNN2.pdf

Receptive field of GCN

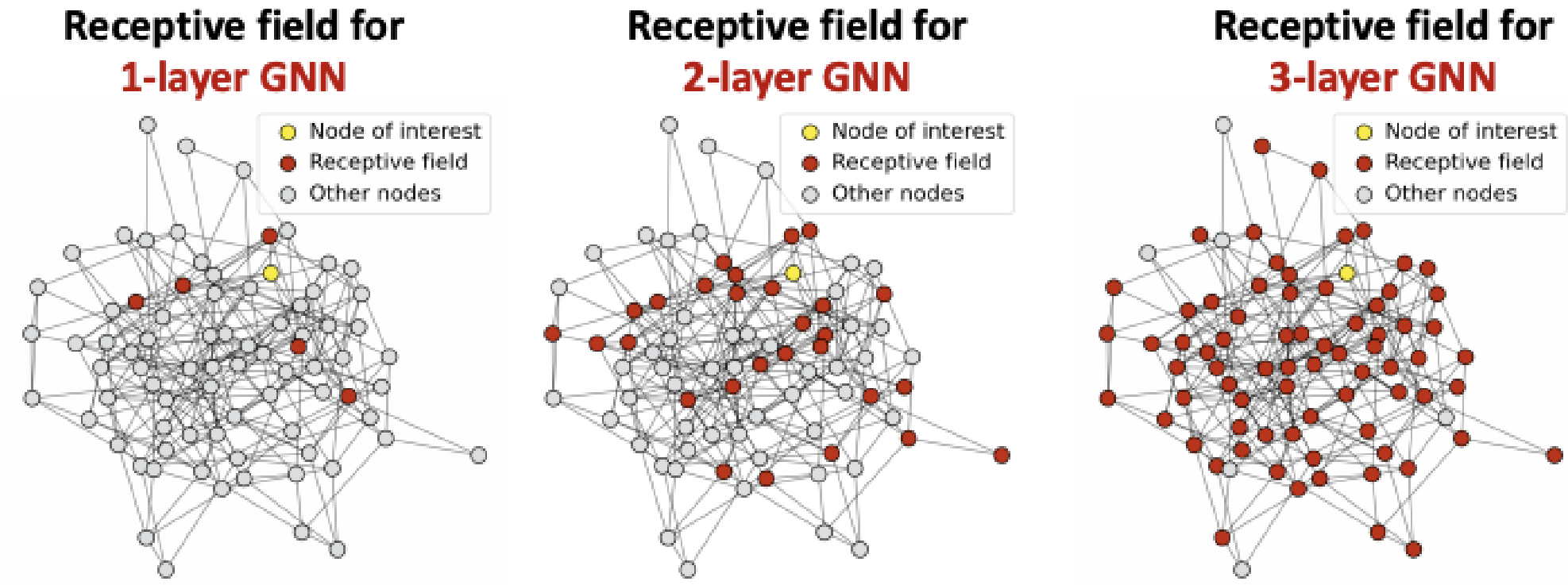


Only 5 nearest neighbors

Over 20 nodes

Image credit http://web.stanford.edu/class/cs224w/slides/07-GNN2.pdf

Receptive field of GCN

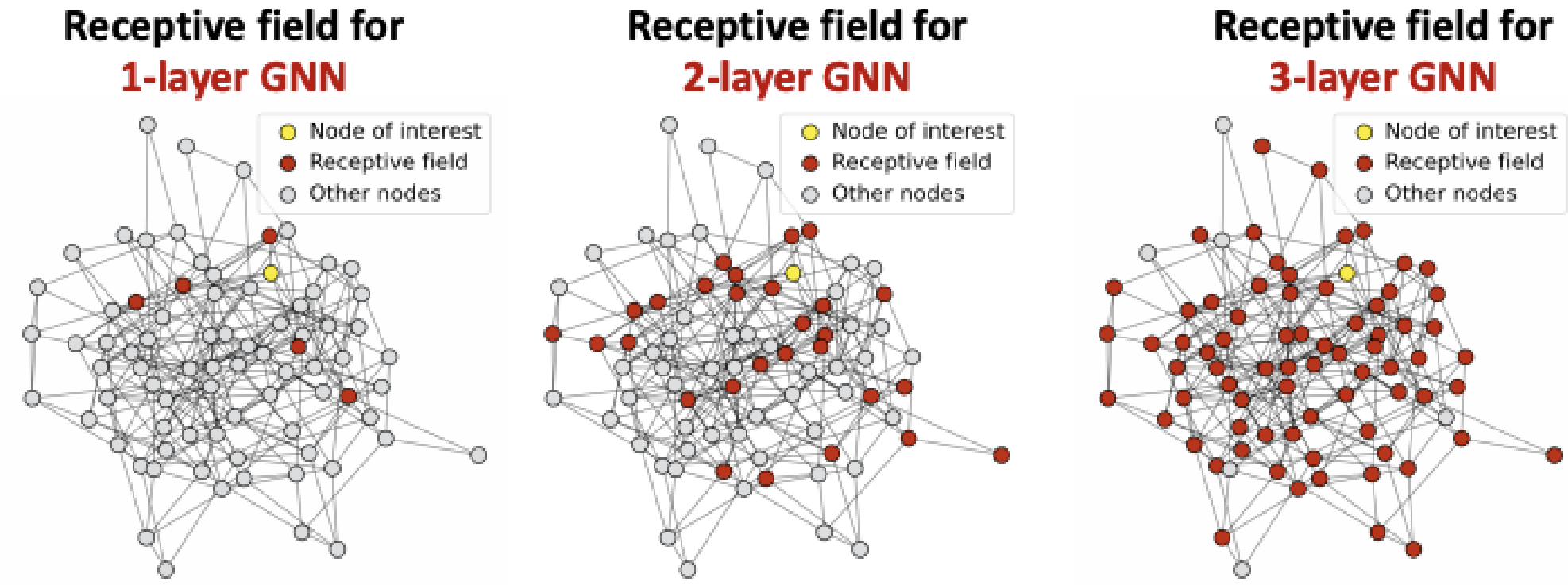


Only 5 nearest neighbors

Over 20 nodes

Image credit http://web.stanford.edu/class/cs224w/slides/07-GNN2.pdf

Receptive field of GCN



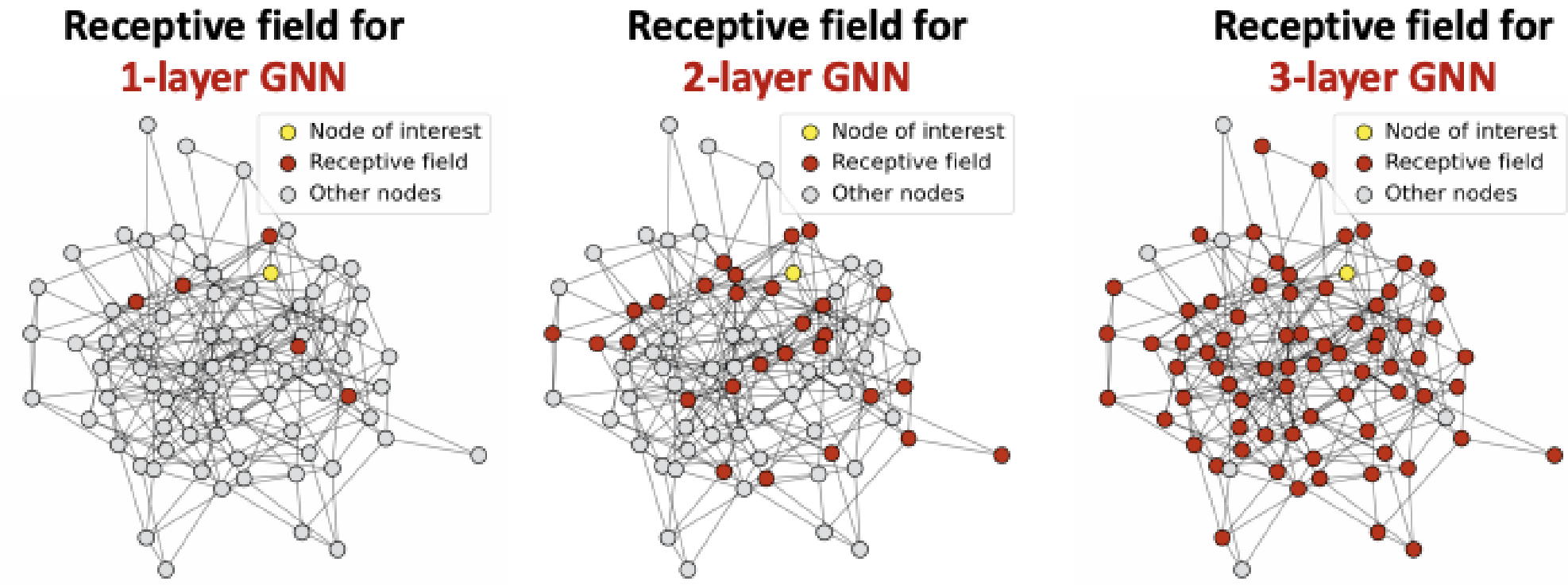
Only 5 nearest neighbors

Over 20 nodes

Almost all nodes

Image credit http://web.stanford.edu/class/cs224w/slides/07-GNN2.pdf

Receptive field of GCN



Only 5 nearest neighbors

Over 20 nodes

Almost all nodes
Aggregate node features regardless the clusters (communities)

Image credit http://web.stanford.edu/class/cs224w/slides/07-GNN2.pdf

Smooth a curve

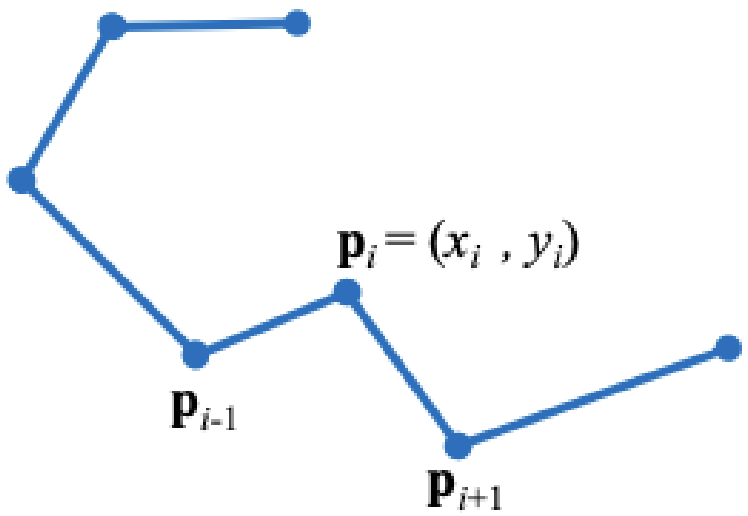
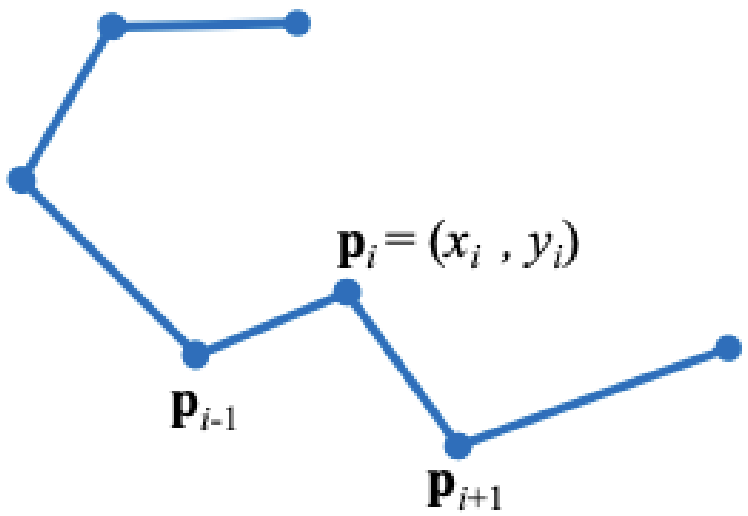


Image credit https://graphics.stanford.edu/courses/cs468-12-spring/LectureSlides/06_smoothing.pdf

Smooth a curve

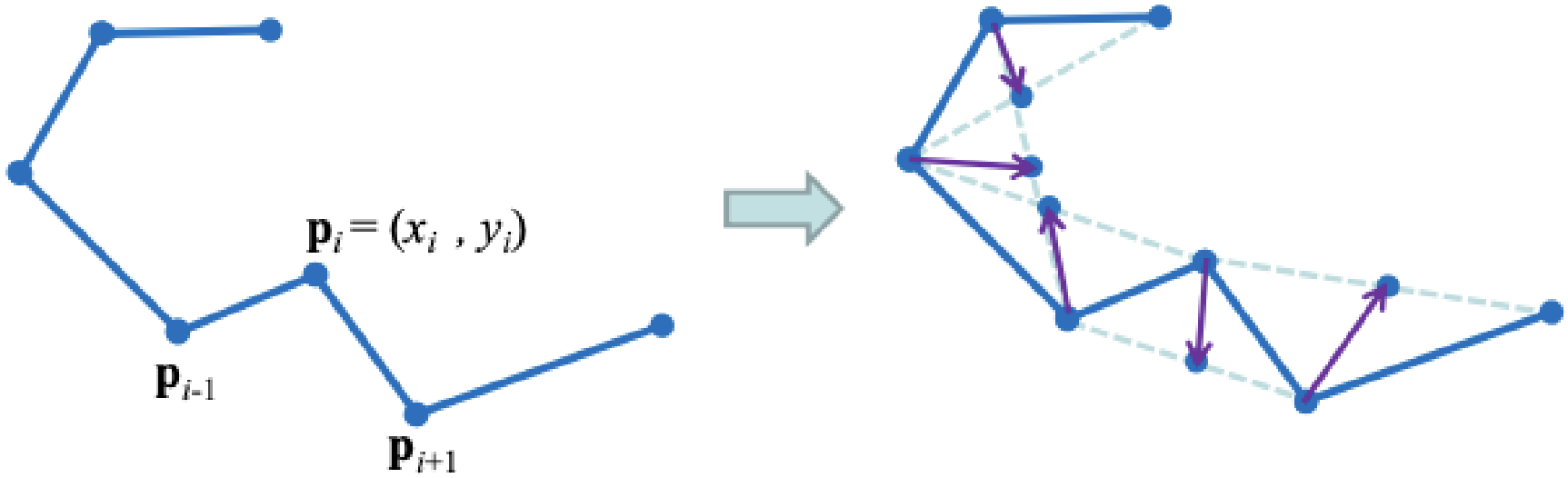


Q: how can we smooth this curve to be more like a straight line?

Image credit https://graphics.stanford.edu/courses/cs468-12-spring/LectureSlides/06_smoothing.pdf

Smooth a curve

Laplacian smoothing



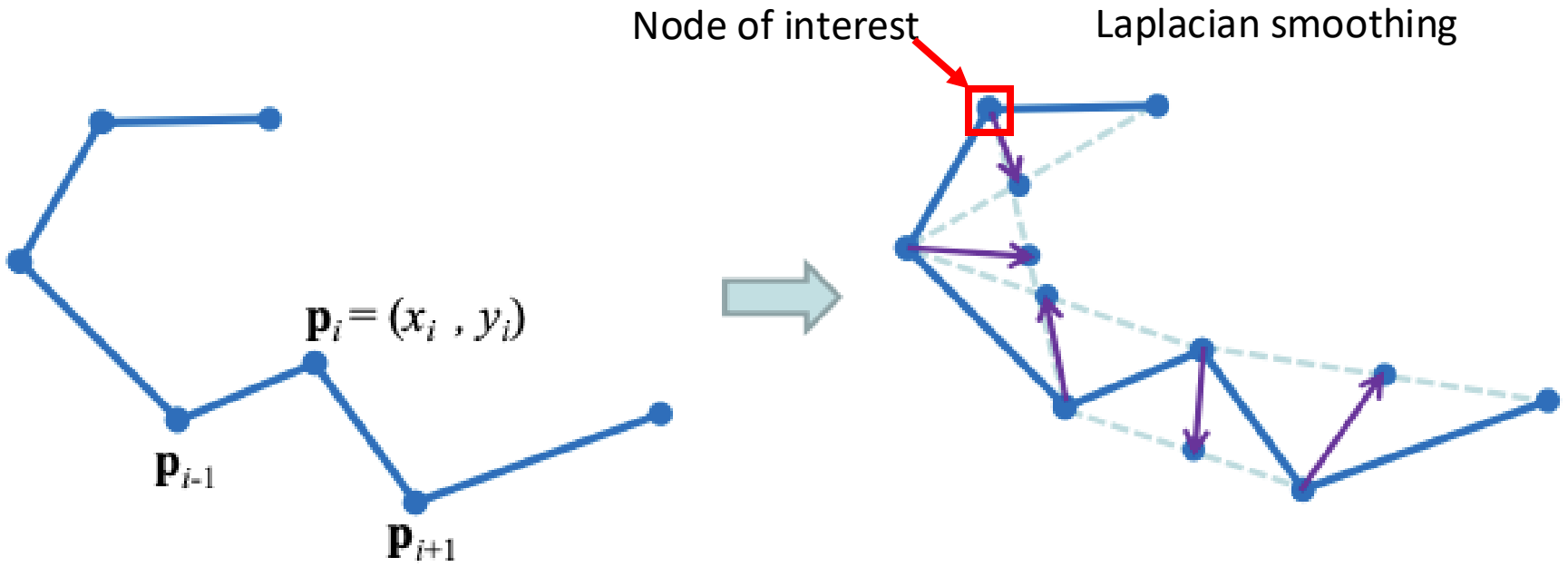
Q: how can we smooth this curve to be more like a straight line?

$$(\mathbf{p}_{i-1} + \mathbf{p}_{i+1})/2 - \mathbf{p}_i$$

$$L(\mathbf{p}_i) = \frac{1}{2}(\mathbf{p}_{i+1} - \mathbf{p}_i) + \frac{1}{2}(\mathbf{p}_{i-1} - \mathbf{p}_i)$$

Image credit https://graphics.stanford.edu/courses/cs468-12-spring/LectureSlides/06_smoothing.pdf

Smooth a curve



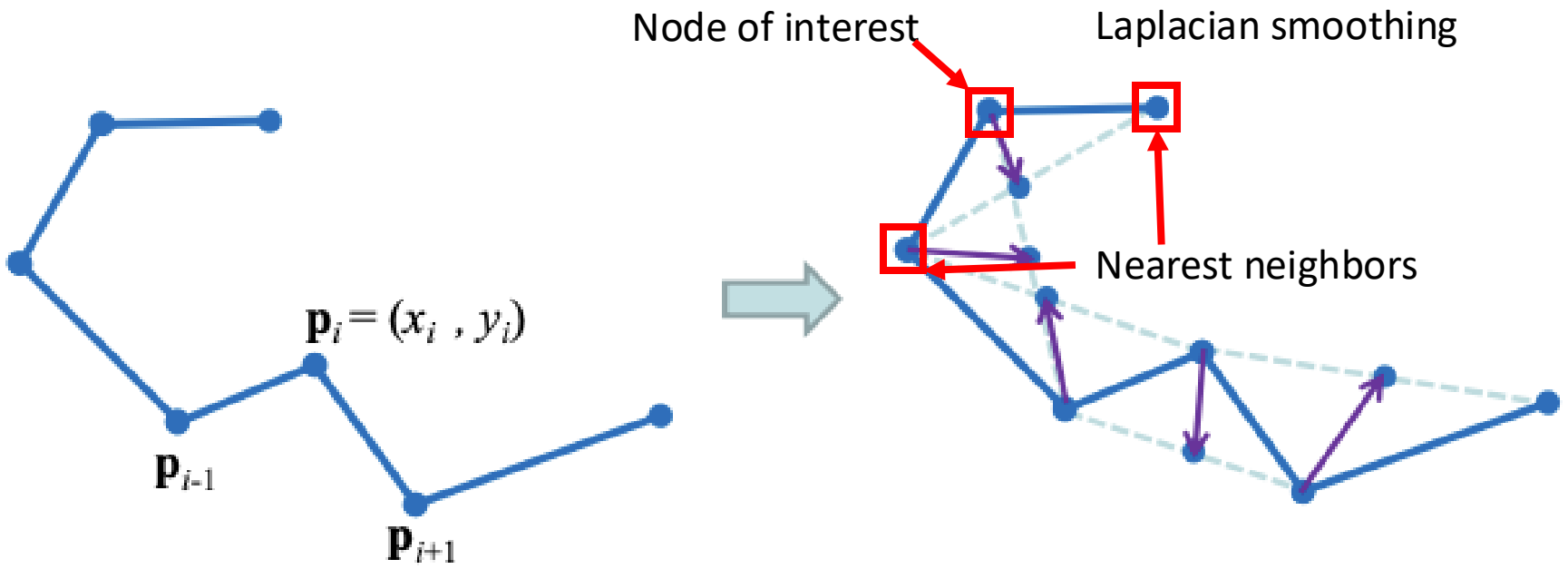
Q: how can we smooth this curve to be more like a straight line?

$$(\mathbf{p}_{i-1} + \mathbf{p}_{i+1})/2 - \mathbf{p}_i$$

$$L(\mathbf{p}_i) = \frac{1}{2}(\mathbf{p}_{i+1} - \mathbf{p}_i) + \frac{1}{2}(\mathbf{p}_{i-1} - \mathbf{p}_i)$$

Image credit https://graphics.stanford.edu/courses/cs468-12-spring/LectureSlides/06_smoothing.pdf

Smooth a curve



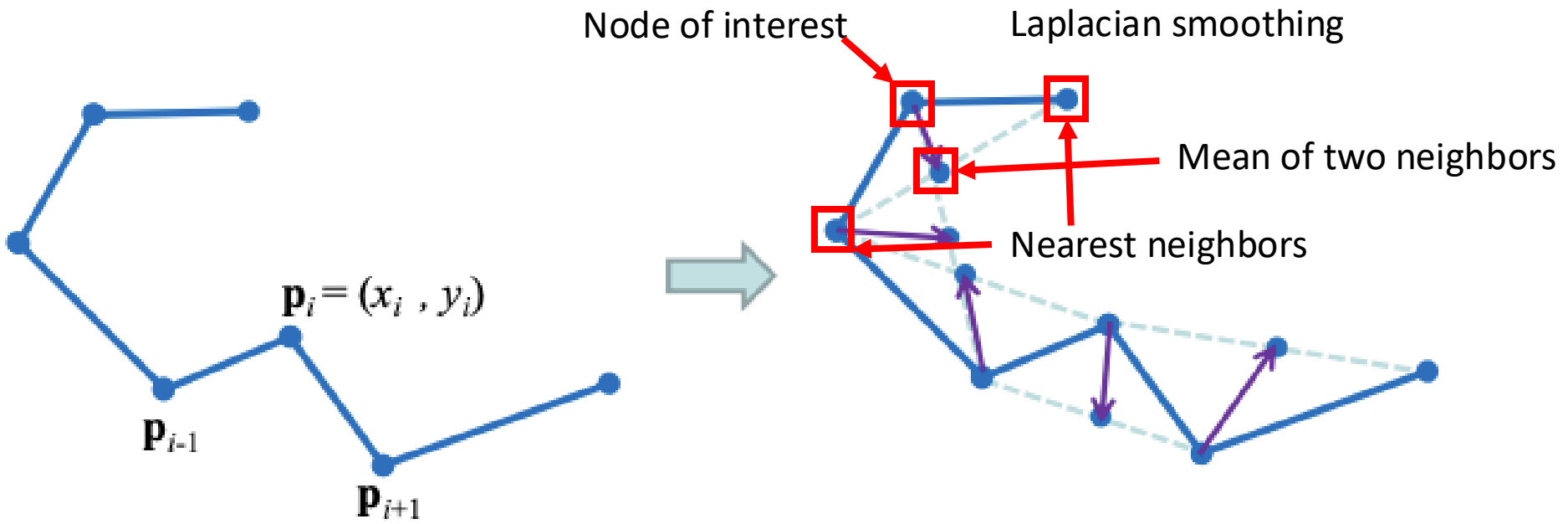
Q: how can we smooth this curve to be more like a straight line?

$$(\mathbf{p}_{i-1} + \mathbf{p}_{i+1})/2 - \mathbf{p}_i$$

$$L(\mathbf{p}_i) = \frac{1}{2}(\mathbf{p}_{i+1} - \mathbf{p}_i) + \frac{1}{2}(\mathbf{p}_{i-1} - \mathbf{p}_i)$$

Image credit https://graphics.stanford.edu/courses/cs468-12-spring/LectureSlides/06_smoothing.pdf

Smooth a curve



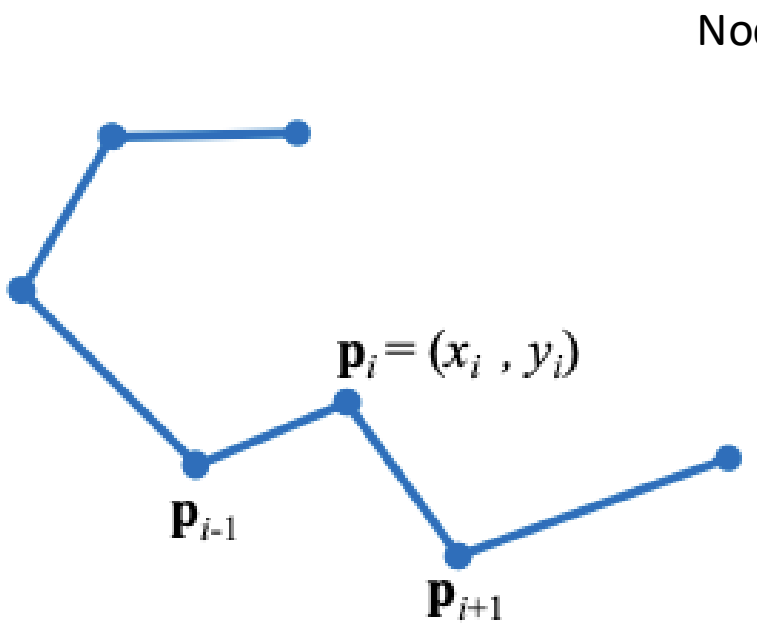
Q: how can we smooth this curve to be more like a straight line?

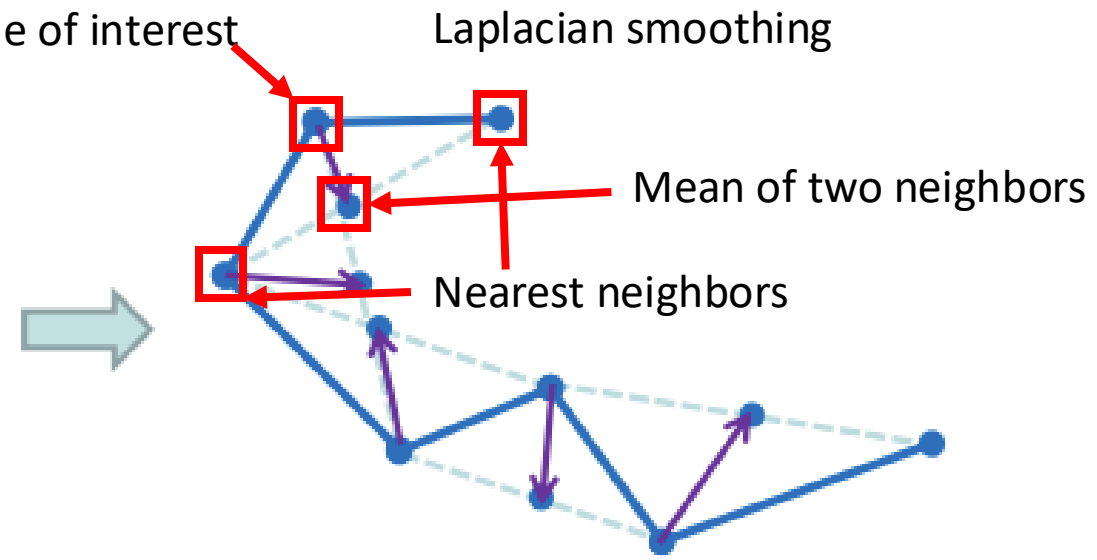
$$(\mathbf{p}_{i-1} + \mathbf{p}_{i+1})/2 - \mathbf{p}_i$$

$$L(\mathbf{p}_i) = \frac{1}{2}(\mathbf{p}_{i+1} - \mathbf{p}_i) + \frac{1}{2}(\mathbf{p}_{i-1} - \mathbf{p}_i)$$

Image credit https://graphics.stanford.edu/courses/cs468-12-spring/LectureSlides/06_smoothing.pdf

Smooth a curve





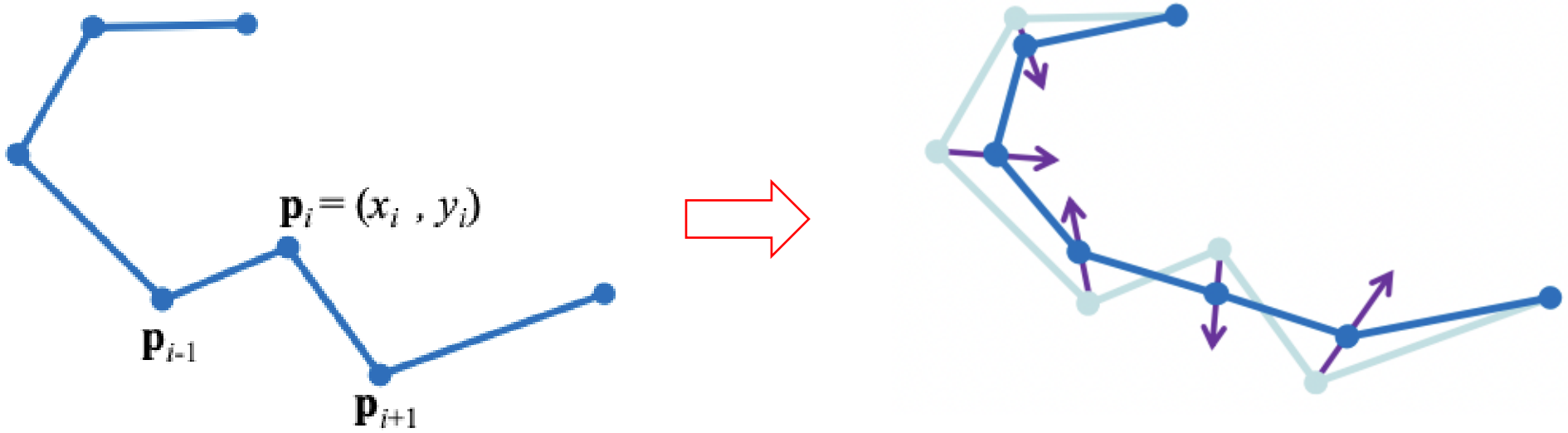
Q: how can we smooth this curve to be more like a straight line?

$$(\mathbf{p}_{i-1} + \mathbf{p}_{i+1})/2 - \mathbf{p}_i$$

$$L(\mathbf{p}_i) = \frac{1}{2}(\mathbf{p}_{i+1} - \mathbf{p}_i) + \frac{1}{2}(\mathbf{p}_{i-1} - \mathbf{p}_i)$$

Image credit https://graphics.stanford.edu/courses/cs468-12-spring/LectureSlides/06_smoothing.pdf

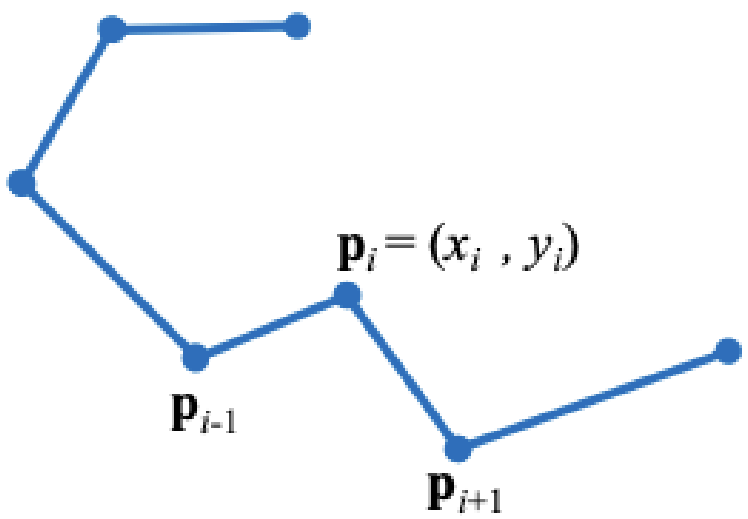
Smooth a curve

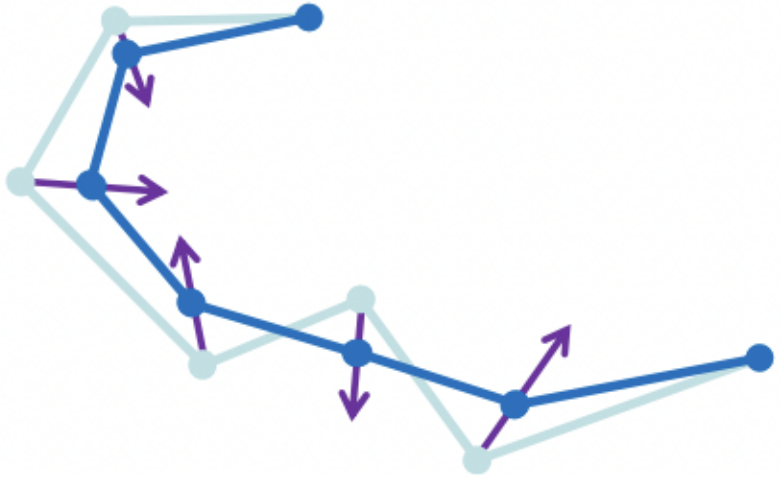


Q: how can we smooth this curve to be more like a straight line?

Image credit https://graphics.stanford.edu/courses/cs468-12-spring/LectureSlides/06_smoothing.pdf

Smooth a curve



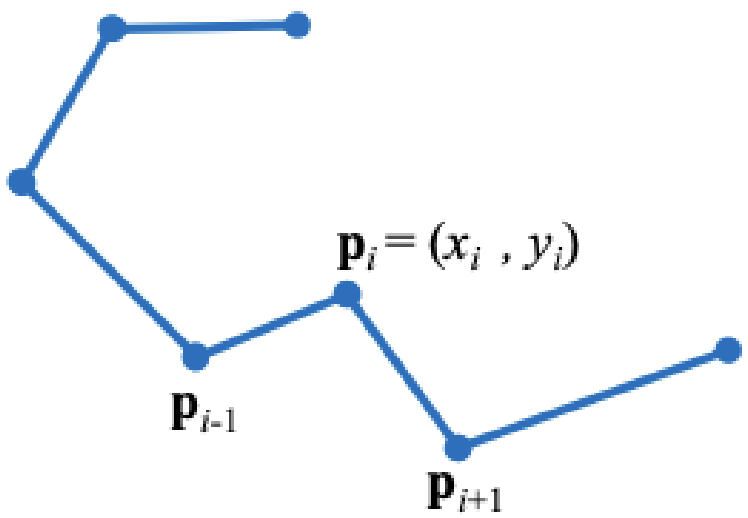


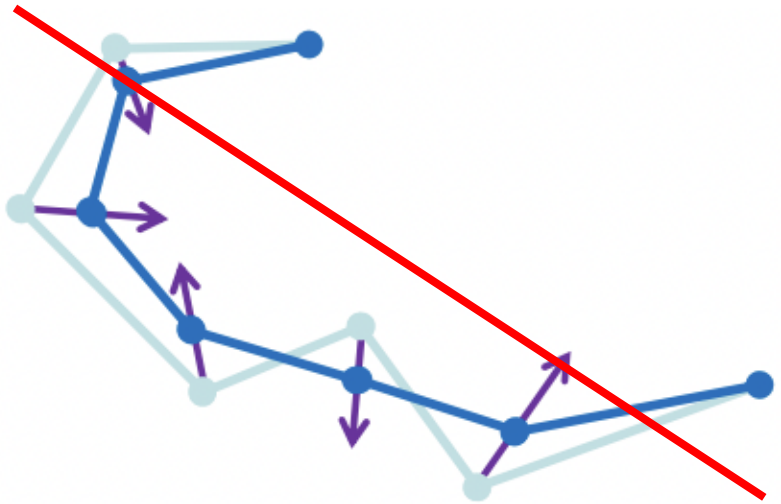
Q: what if we repeat for many times?

Q: how can we smooth this curve to be more like a straight line?

Image credit https://graphics.stanford.edu/courses/cs468-12-spring/LectureSlides/06_smoothing.pdf

Smooth a curve



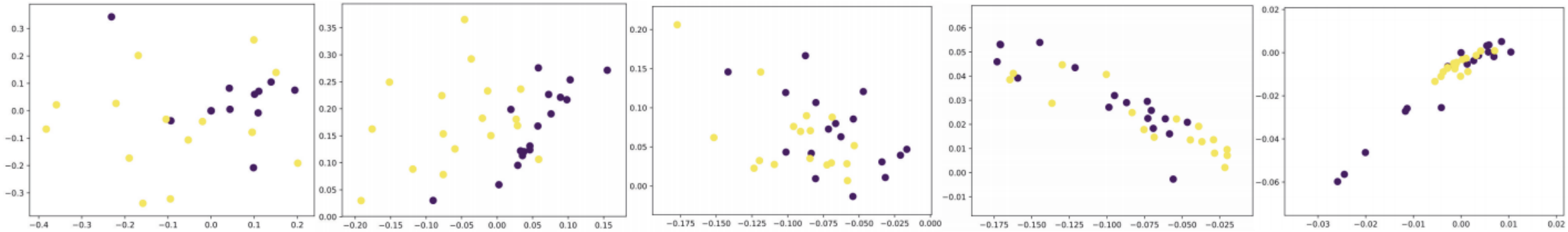


Q: what if we repeat for many times?
Converge to a straight line?

Q: how can we smooth this curve to be more like a straight line?

Image credit https://graphics.stanford.edu/courses/cs468-12-spring/LectureSlides/06_smoothing.pdf

Oversmoothing of GCN

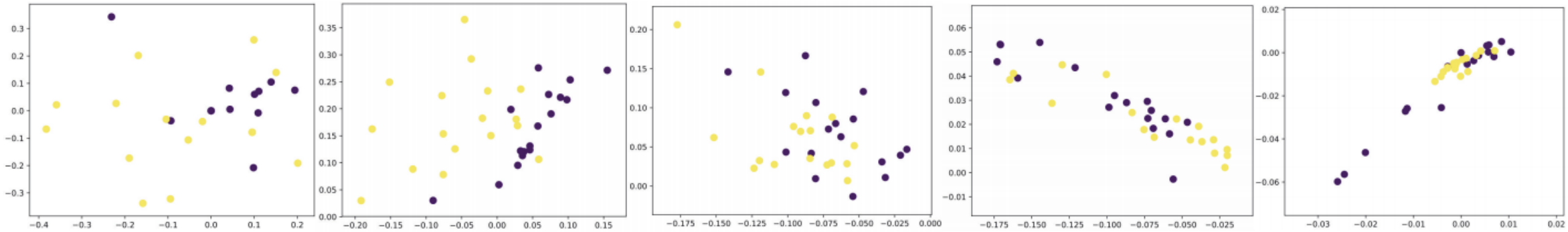


(a) 1-layer (b) 2-layer (c) 3-layer (d) 4-layer (e) 5-layer

Figure 2: Vertex embeddings of Zachary's karate club network with GCNs with 1,2,3,4,5 layers.

Image credit https://arxiv.org/pdf/1801.07606.pdf.
Deeper Insights into Graph Convolutional Networks for Semi-Supervised Learning. In AAAI 2018.

Oversmoothing of GCN



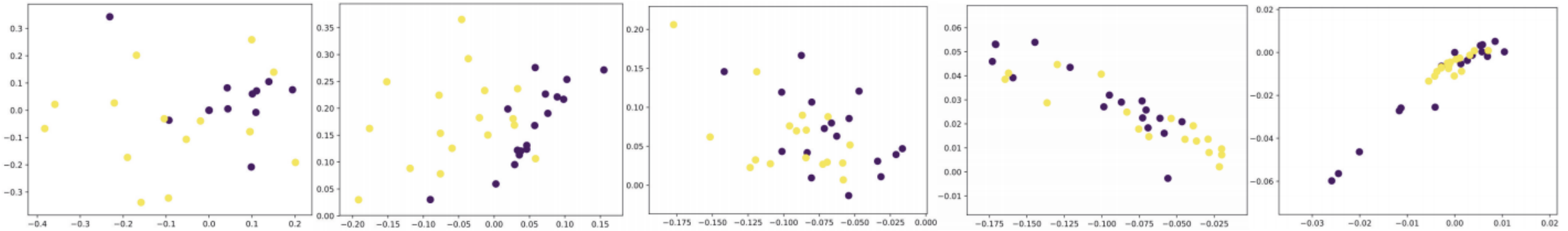
(a) 1-layer (b) 2-layer (c) 3-layer (d) 4-layer (e) 5-layer

Figure 2: Vertex embeddings of Zachary's karate club network with GCNs with 1,2,3,4,5 layers.

34 vertices of two classes and 78 edges

Image credit https://arxiv.org/pdf/1801.07606.pdf.
Deeper Insights into Graph Convolutional Networks for Semi-Supervised Learning. In AAAI 2018.

Oversmoothing of GCN



(a) 1-layer (b) 2-layer (c) 3-layer (d) 4-layer (e) 5-layer

Larger receptive field

Figure 2: Vertex embeddings of Zachary's karate club network with GCNs with 1,2,3,4,5 layers.

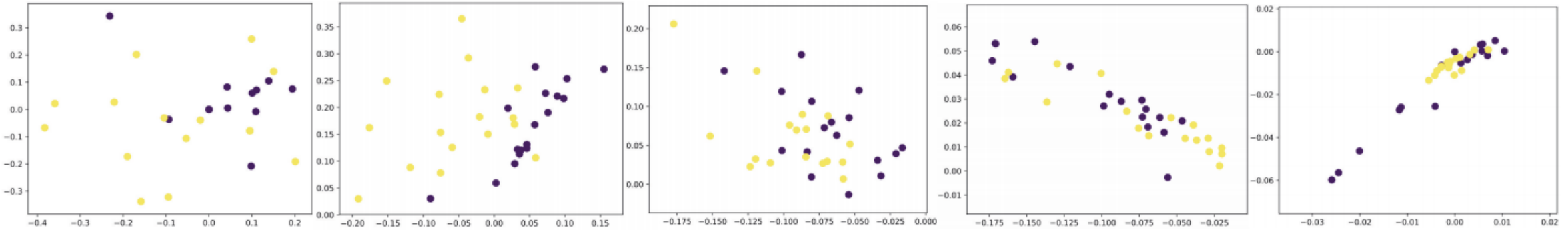
34 vertices of two classes and 78 edges

Image credit https://arxiv.org/pdf/1801.07606.pdf.

Deeper Insights into Graph Convolutional Networks for Semi-Supervised Learning. In AAAI 2018.

Oversmoothing of GCN

Two classes: yellow vs. blue



(a) 1-layer (b) 2-layer (c) 3-layer (d) 4-layer (e) 5-layer

Larger receptive field

Figure 2: Vertex embeddings of Zachary's karate club network with GCNs with 1,2,3,4,5 layers.

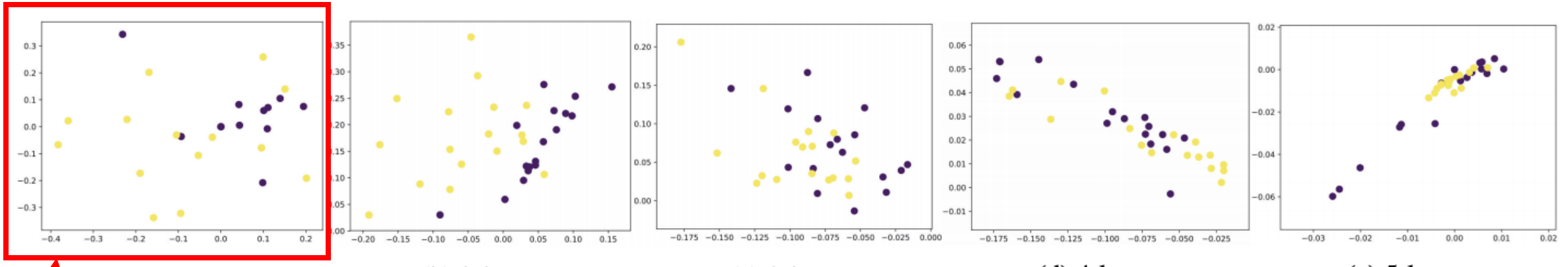
34 vertices of two classes and 78 edges

Image credit https://arxiv.org/pdf/1801.07606.pdf.

Deeper Insights into Graph Convolutional Networks for Semi-Supervised Learning. In AAAI 2018.

Oversmoothing of GCN

Two classes: yellow vs. blue



(a) 1-layer (b) 2-layer (c) 3-layer (d) 4-layer (e) 5-layer

Larger receptive field

Figure 2: Vertex embeddings of Zachary's karate club network with GCNs with 1,2,3,4,5 layers.

34 vertices of two classes and 78 edges

Q: is it linearly separable (can we use a straight line to separate two classes well)?

Image credit https://arxiv.org/pdf/1801.07606.pdf.
Deeper Insights into Graph Convolutional Networks for Semi-Supervised Learning. In AAAI 2018.

Oversmoothing of GCN

Two classes: yellow vs. blue

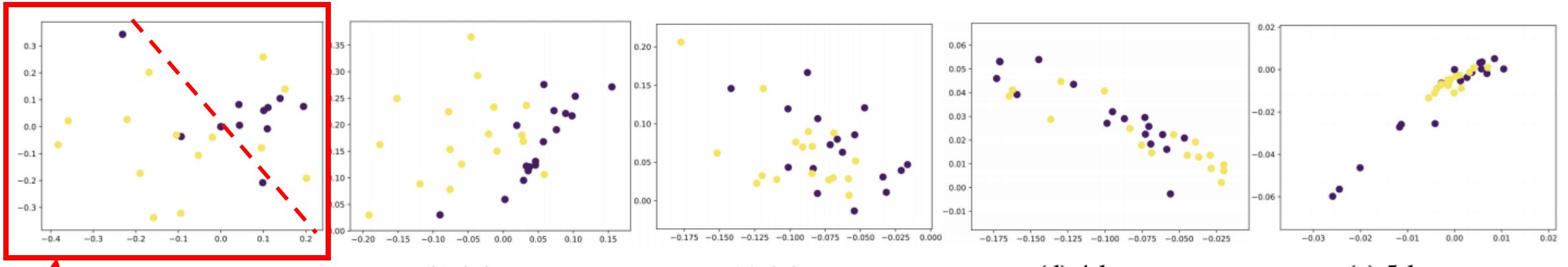


Figure 2: Vertex embeddings of Zachary's karate club network with GCNs with 1,2,3,4,5 layers.

34 vertices of two classes and 78 edges

Q: is it linearly separable (can we use a straight line to separate two classes well)?

Image credit https://arxiv.org/pdf/1801.07606.pdf.
Deeper Insights into Graph Convolutional Networks for Semi-Supervised Learning. In AAAI 2018.

Oversmoothing of GCN

Two classes: yellow vs. blue

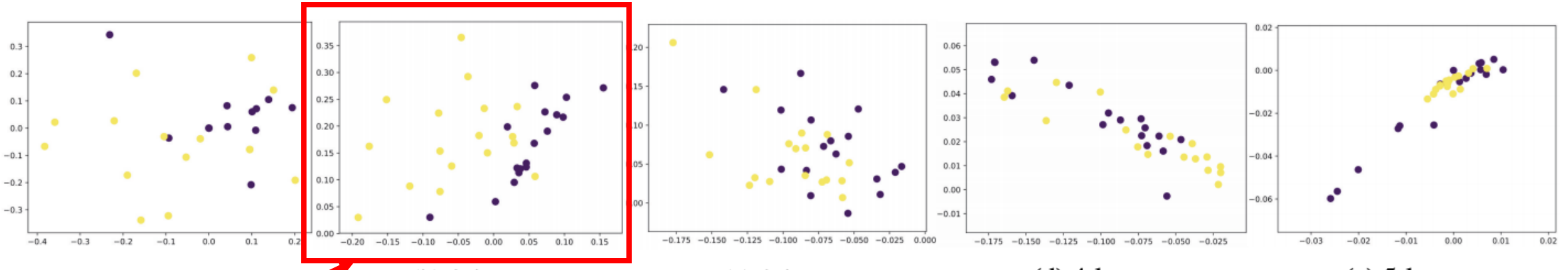


Figure 2: Vertex embeddings of Zachary's karate club network with GCNs with 1,2,3,4,5 layers.

34 vertices of two classes and 78 edges

Q: is it linearly separable (can we use a straight line to separate two classes well)?

Image credit https://arxiv.org/pdf/1801.07606.pdf.
Deeper Insights into Graph Convolutional Networks for Semi-Supervised Learning. In AAAI 2018.

Oversmoothing of GCN

Two classes: yellow vs. blue

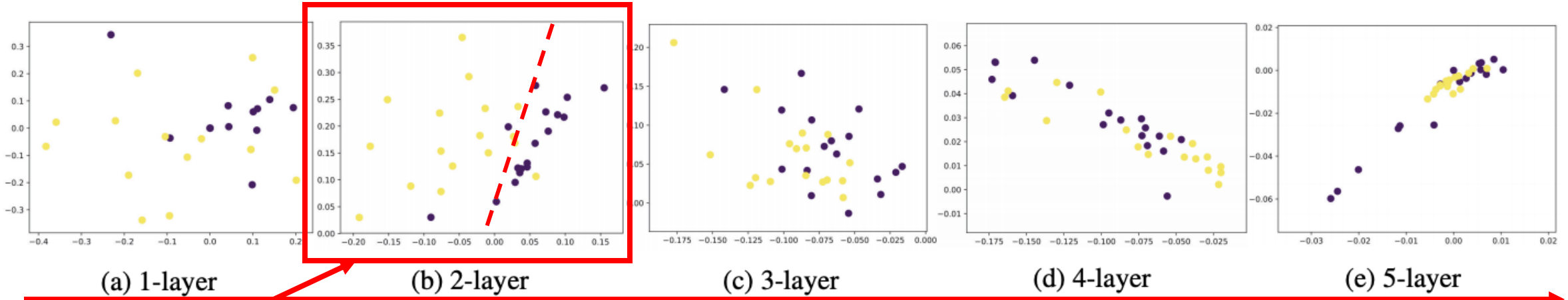


Figure 2: Vertex embeddings of Zachary's karate club network with GCNs with 1,2,3,4,5 layers.

34 vertices of two classes and 78 edges

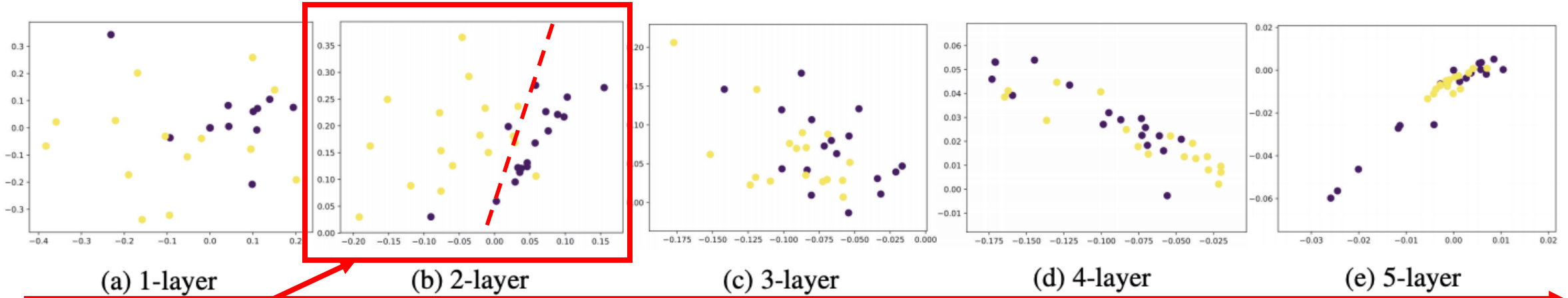
Q: is it linearly separable (can we use a straight line to separate two classes well)?

Image credit https://arxiv.org/pdf/1801.07606.pdf.
Deeper Insights into Graph Convolutional Networks for Semi-Supervised Learning. In AAAI 2018.

Oversmoothing of GCN

Two classes: yellow vs. blue

A bit better



Larger receptive field

Figure 2: Vertex embeddings of Zachary's karate club network with GCNs with 1,2,3,4,5 layers.

34 vertices of two classes and 78 edges

Q: is it linearly separable (can we use a straight line to separate two classes well)?

Image credit https://arxiv.org/pdf/1801.07606.pdf.
Deeper Insights into Graph Convolutional Networks for Semi-Supervised Learning. In AAAI 2018.

Oversmoothing of GCN

Two classes: yellow vs. blue

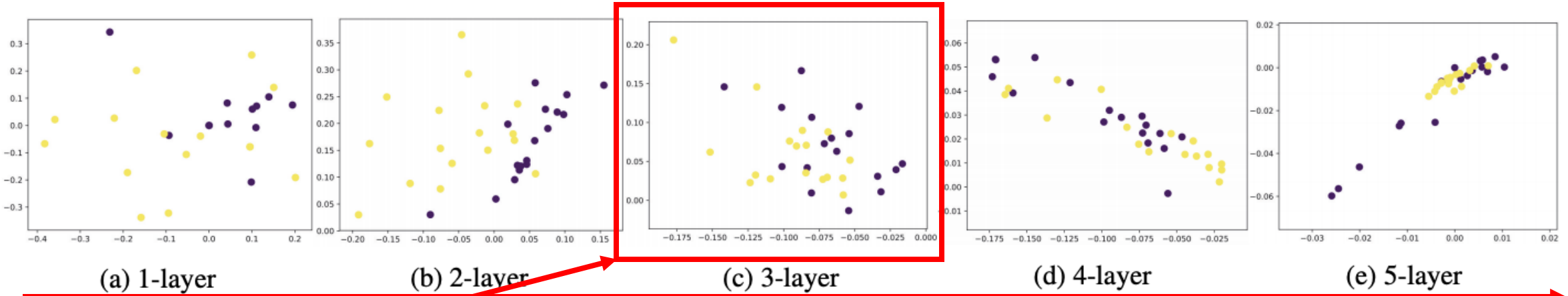


Figure 2: Vertex embeddings of Zachary's karate club network with GCNs with 1,2,3,4,5 layers.

34 vertices of two classes and 78 edges

Q: is it linearly separable (can we use a straight line to separate two classes well)?

Image credit https://arxiv.org/pdf/1801.07606.pdf.
Deeper Insights into Graph Convolutional Networks for Semi-Supervised Learning. In AAAI 2018.

Oversmoothing of GCN

Two classes: yellow vs. blue

NO!

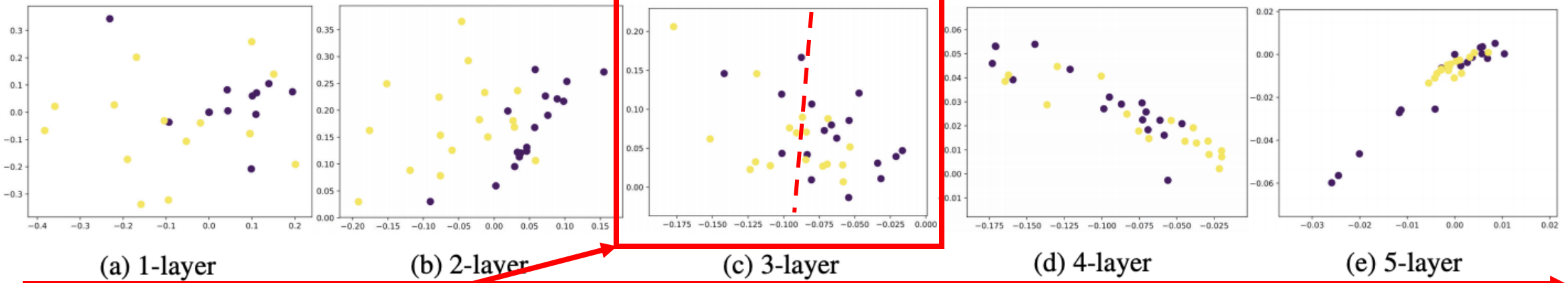


Figure 2: Vertex embeddings of Zachary's karate club network with GCNs with 1,2,3,4,5 layers.

34 vertices of two classes and 78 edges

Q: is it linearly separable (can we use a straight line to separate two classes well)?

Image credit https://arxiv.org/pdf/1801.07606.pdf.
Deeper Insights into Graph Convolutional Networks for Semi-Supervised Learning. In AAAI 2018.

Oversmoothing of GCN

Two classes: yellow vs. blue

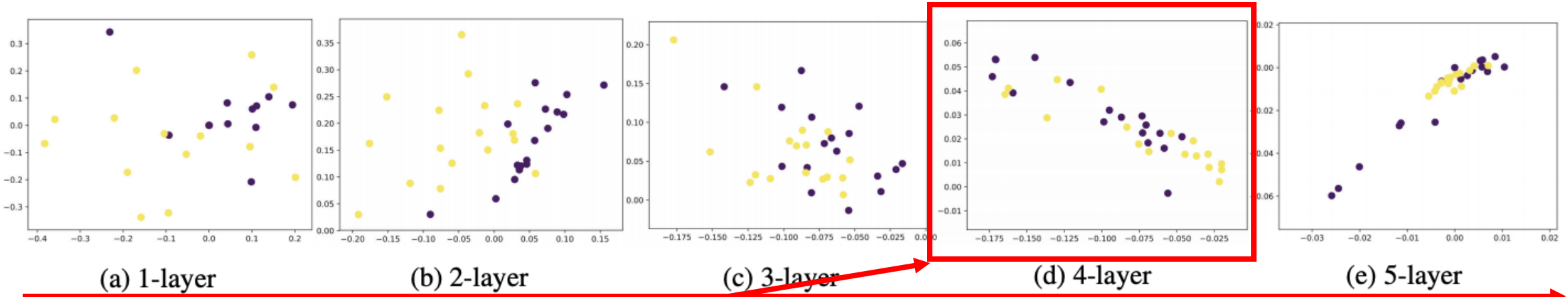


Figure 2: Vertex embeddings of Zachary's karate club network with GCNs with 1,2,3,4,5 layers.

34 vertices of two classes and 78 edges

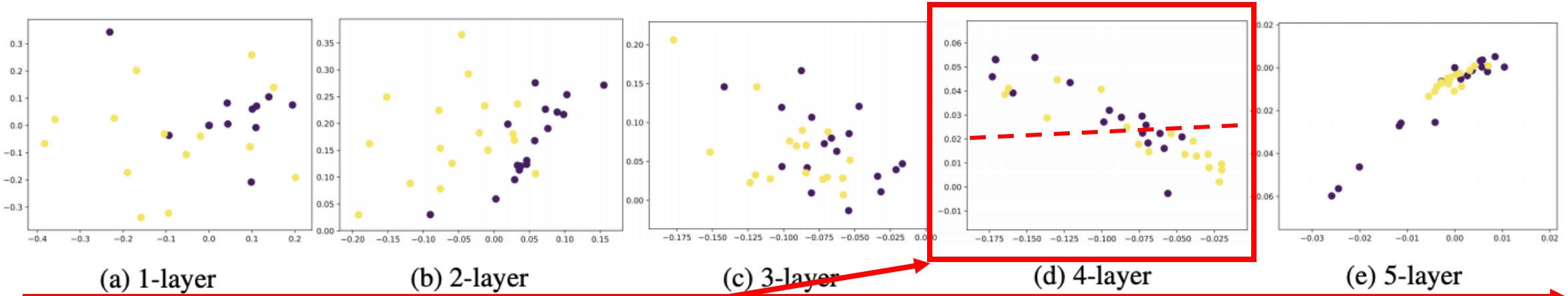
Q: is it linearly separable (can we use a straight line to separate two classes well)?

Image credit https://arxiv.org/pdf/1801.07606.pdf.
Deeper Insights into Graph Convolutional Networks for Semi-Supervised Learning. In AAAI 2018.

Oversmoothing of GCN

Two classes: yellow vs. blue

NO!



(a) 1-layer (b) 2-layer (c) 3-layer (d) 4-layer (e) 5-layer

Larger receptive field

Figure 2: Vertex embeddings of Zachary's karate club network with GCNs with 1,2,3,4,5 layers.

34 vertices of two classes and 78 edges

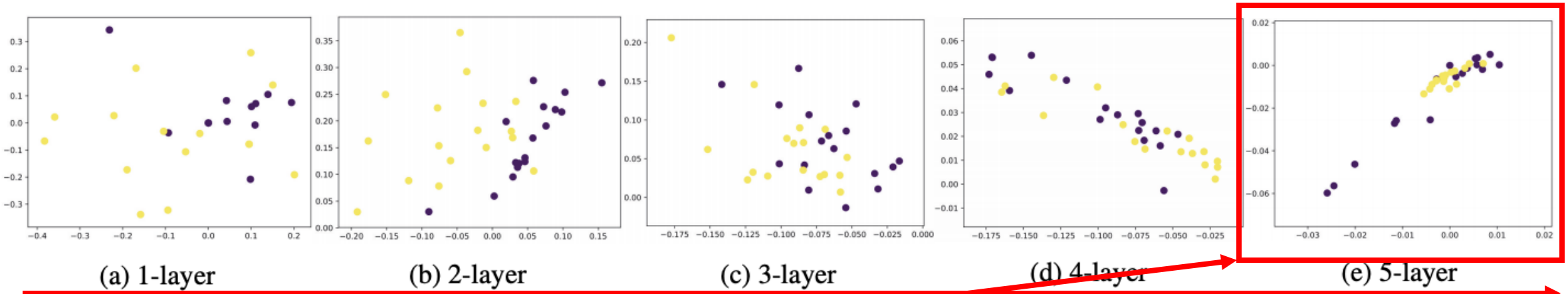
Q: is it linearly separable (can we use a straight line to separate two classes well)?

Image credit https://arxiv.org/pdf/1801.07606.pdf.

Deeper Insights into Graph Convolutional Networks for Semi-Supervised Learning. In AAAI 2018.

Oversmoothing of GCN

Two classes: yellow vs. blue



(a) 1-layer (b) 2-layer (c) 3-layer (d) 4-layer (e) 5-layer

Larger receptive field

Figure 2: Vertex embeddings of Zachary's karate club network with GCNs with 1,2,3,4,5 layers.

34 vertices of two classes and 78 edges

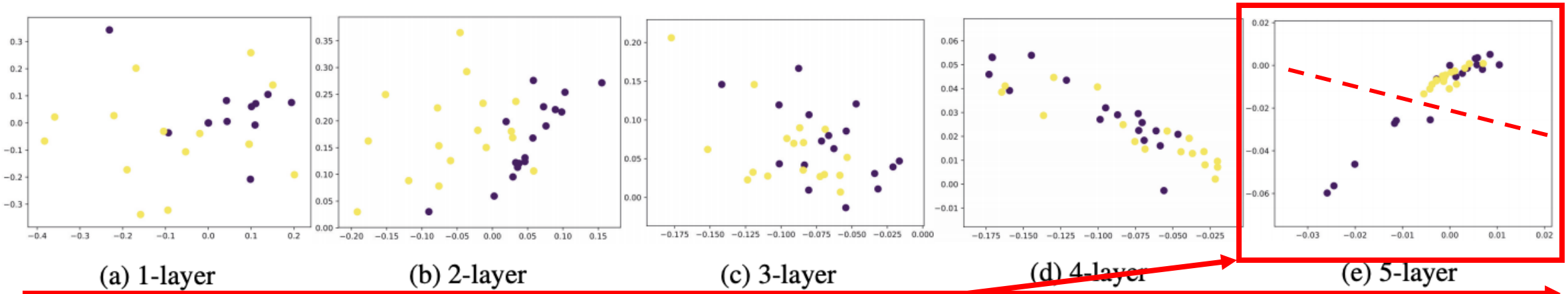
Q: is it linearly separable (can we use a straight line to separate two classes well)?

Image credit https://arxiv.org/pdf/1801.07606.pdf.
Deeper Insights into Graph Convolutional Networks for Semi-Supervised Learning. In AAAI 2018.

Oversmoothing of GCN

Two classes: yellow vs. blue

NO!



(a) 1-layer (b) 2-layer (c) 3-layer (d) 4-layer (e) 5-layer

Larger receptive field

Figure 2: Vertex embeddings of Zachary's karate club network with GCNs with 1,2,3,4,5 layers.

34 vertices of two classes and 78 edges

Q: is it linearly separable (can we use a straight line to separate two classes well)?

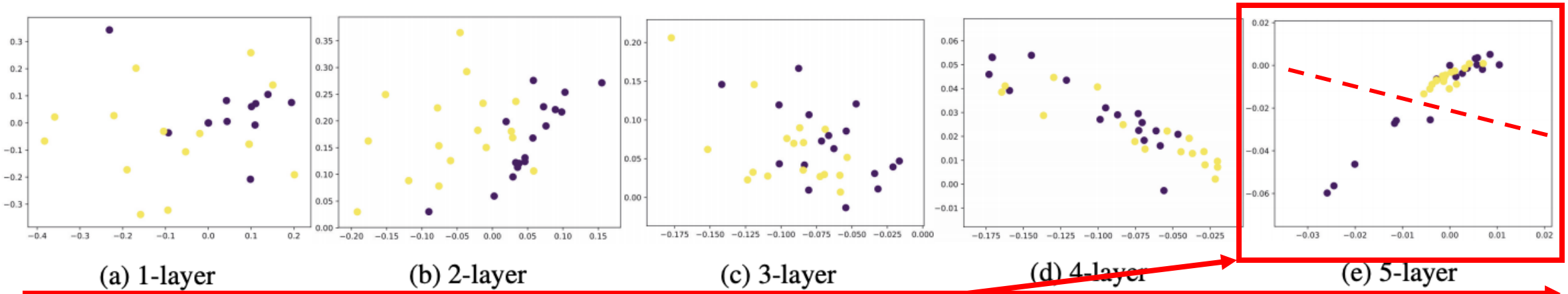
Image credit https://arxiv.org/pdf/1801.07606.pdf.

Deeper Insights into Graph Convolutional Networks for Semi-Supervised Learning. In AAAI 2018.

Oversmoothing of GCN

Two classes: yellow vs. blue

NO!



(a) 1-layer (b) 2-layer (c) 3-layer (d) 4-layer (e) 5-layer

Larger receptive field

Figure 2: Vertex embeddings of Zachary's karate club network with GCNs with 1,2,3,4,5 layers.

34 vertices of two classes and 78 edges

Q: is it linearly separable (can we use a straight line to separate two classes well)?

Q: will you choose a 2-layer or 5-layer GCN for the node classification on this dataset?

Image credit https://arxiv.org/pdf/1801.07606.pdf.

Deeper Insights into Graph Convolutional Networks for Semi-Supervised Learning. In AAAI 2018.

Oversmoothing of GCN

Two classes: yellow vs. blue

Nodes features are not linearly separable

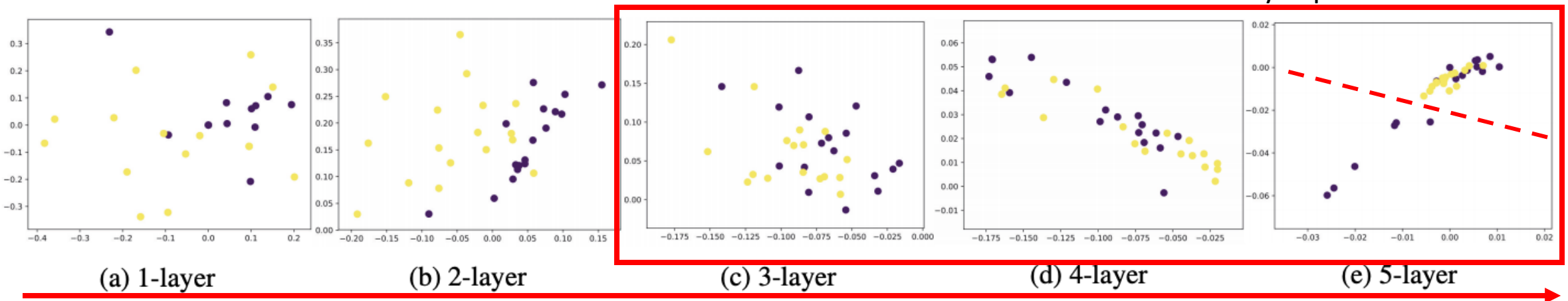


Figure 2: Vertex embeddings of Zachary's karate club network with GCNs with 1,2,3,4,5 layers.

34 vertices of two classes and 78 edges

Q: is it linearly separable (can we use a straight line to separate two classes well)?

Q: will you choose a 2-layer or 5-layer GCN for the node classification on this dataset?

Image credit https://arxiv.org/pdf/1801.07606.pdf.

Deeper Insights into Graph Convolutional Networks for Semi-Supervised Learning. In AAAI 2018.

Solutions to oversmoothing of GCN

- Properly set the number of GCN layers

Solutions to oversmoothing of GCN

- Properly set the number of GCN layers (k hops-away neighbors)

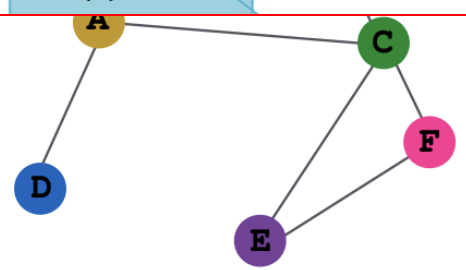
Solutions to oversmoothing of GCN

- Properly set the number of GCN layers (k hops-away neighbors)
- Increase the number of layers that do not aggregate neighbors
  - Use MLP to aggregate neighbors' feature from the previous layer

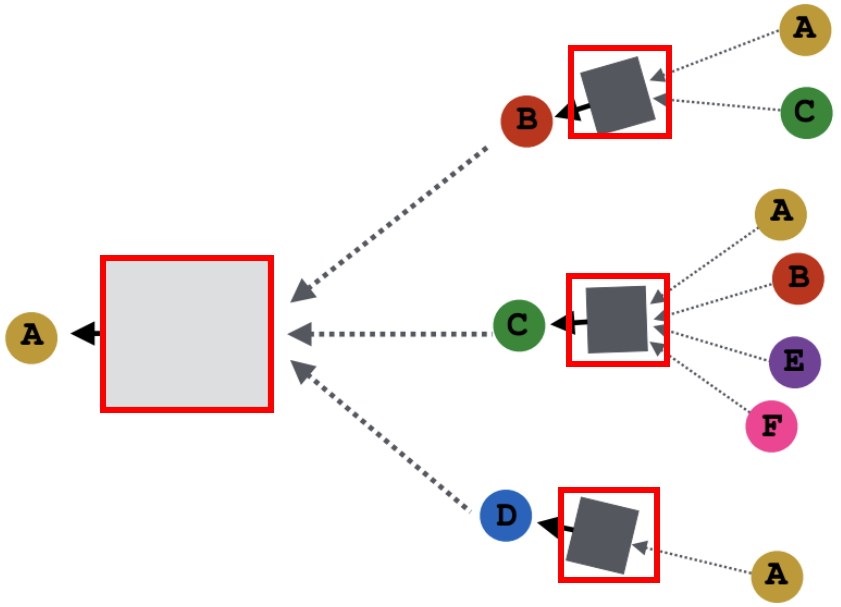
Solutions to oversmoothing of GCN

- Properly set the number of GCN layers (k hops-away neighbors)
- Increase the number of layers that do not aggregate neighbors
  - Use MLP to aggregate neighbors' feature from the previous layer

$$h_v^{(l+1)} = \sigma\left(W_l \sum_{u \in N(v)} \frac{h_u^{(l)}}{|N(v)|} + B_l h_v^{(l)}\right), \forall l \in \{0, \ldots, L-1\}$$



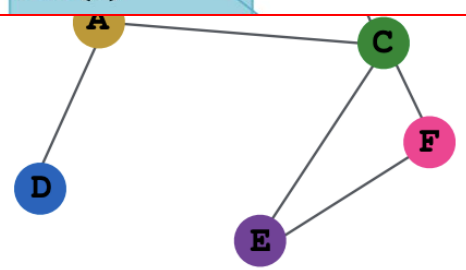
INPUT GRAPH



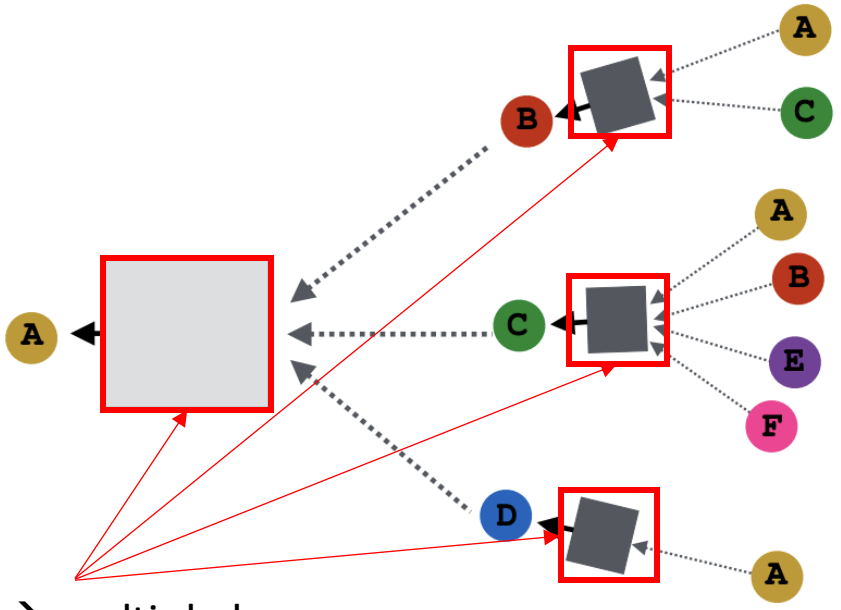
Solutions to oversmoothing of GCN

- Properly set the number of GCN layers (k hops-away neighbors)
- Increase the number of layers that do not aggregate neighbors
  - Use MLP to aggregate neighbors' feature from the previous layer

$$h_v^{(l+1)} = \sigma(W_l \sum_{u \in N(v)} \frac{h_u^{(l)}}{|N(v)|} + B_l h_v^{(l)}), \forall l \in \{0, \ldots, L-1\}$$



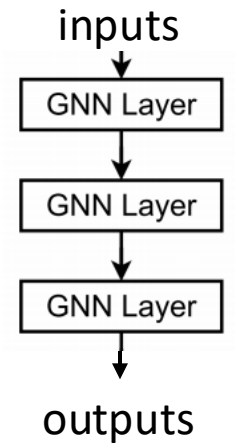
INPUT GRAPH



One layer → multiple layers

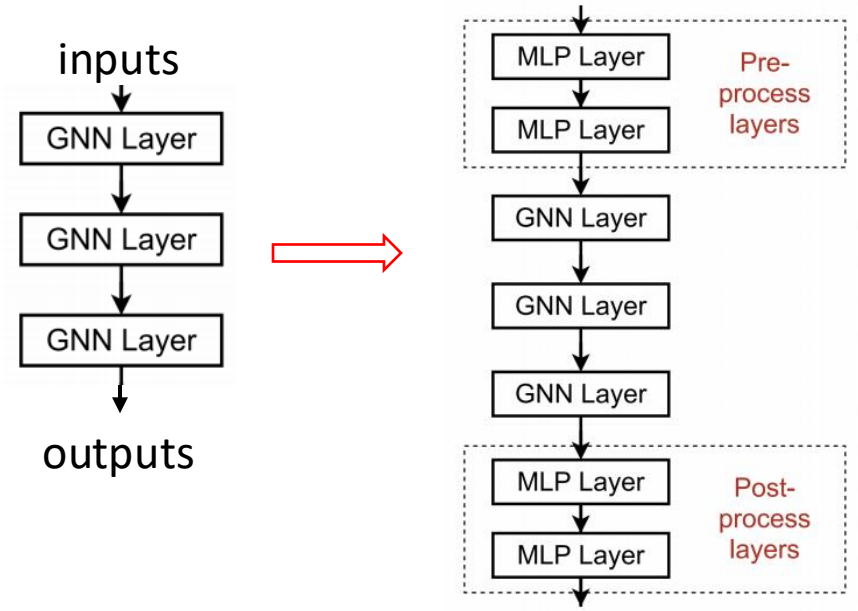
Solutions to oversmoothing of GCN

- Properly set the number of GCN layers (k hops-away neighbors)
- Increase the number of layers that do not aggregate neighbors
  - Use MLP to aggregate neighbors' feature from the previous layer
  - Use layers that do not aggregate neighbors



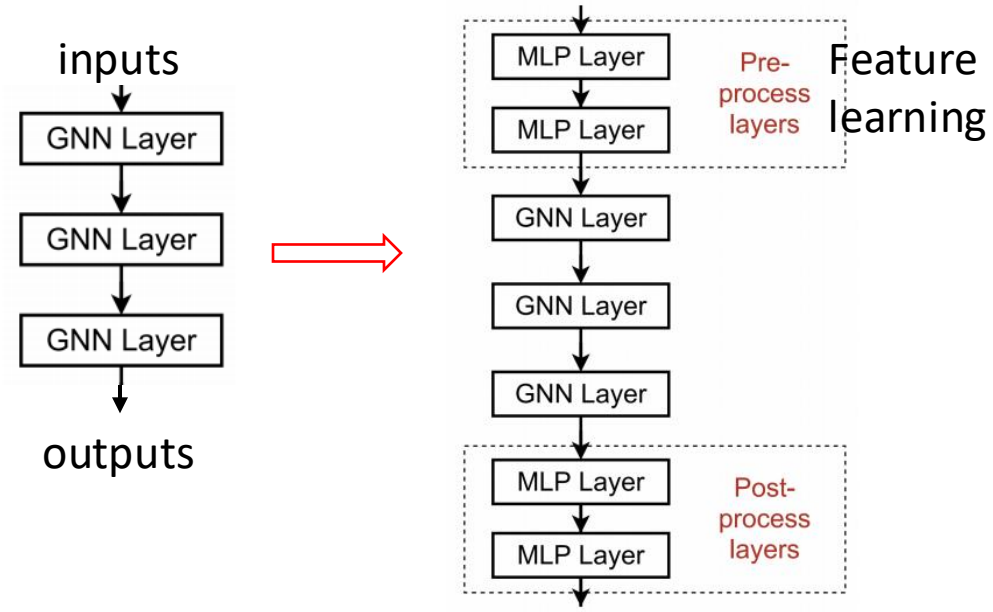
Solutions to oversmoothing of GCN

- Properly set the number of GCN layers (k hops-away neighbors)
- Increase the number of layers that do not aggregate neighbors
  - Use MLP to aggregate neighbors' feature from the previous layer
  - Use layers that do not aggregate neighbors



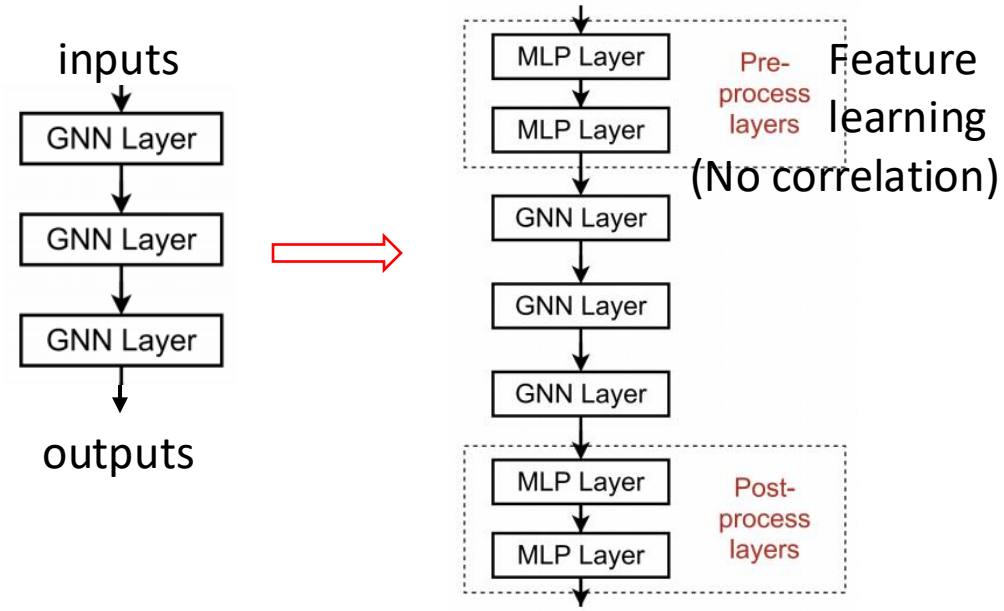
Solutions to oversmoothing of GCN

- Properly set the number of GCN layers (k hops-away neighbors)
- Increase the number of layers that do not aggregate neighbors
  - Use MLP to aggregate neighbors' feature from the previous layer
  - Use layers that do not aggregate neighbors



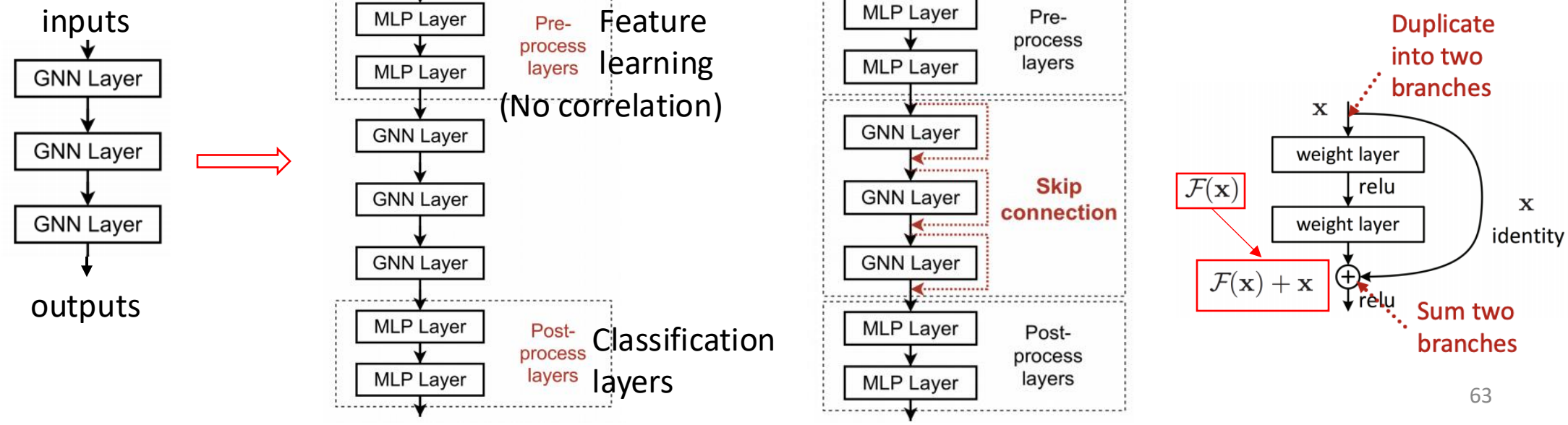
Solutions to oversmoothing of GCN

- Properly set the number of GCN layers (k hops-away neighbors)
- Increase the number of layers that do not aggregate neighbors
  - Use MLP to aggregate neighbors' feature from the previous layer
  - Use layers that do not aggregate neighbors



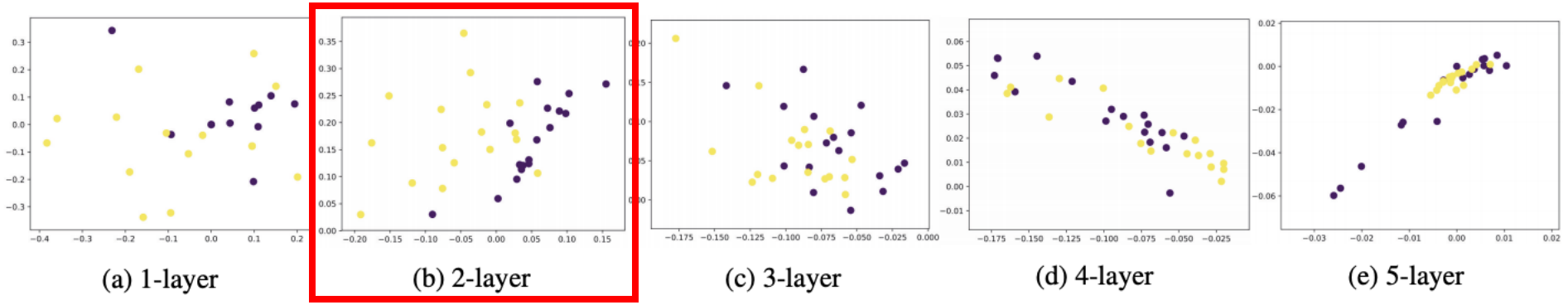
Solutions to oversmoothing of GCN

- Properly set the number of GCN layers (k hops-away neighbors)
- Increase the number of layers that do not aggregate neighbors
  - Use MLP to aggregate neighbors' feature from the previous layer
  - Use layers that do not aggregate neighbors



Why skip connection works?

discriminative features



(a) 1-layer (b) 2-layer (c) 3-layer (d) 4-layer (e) 5-layer

Figure 2: Vertex embeddings of Zachary's karate club network with GCNs with 1,2,3,4,5 layers.

Why skip connection works?

discriminative features

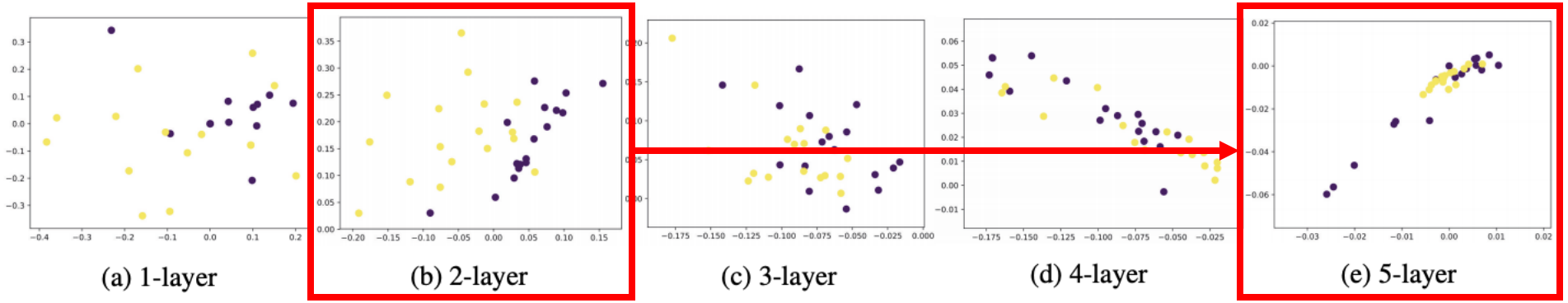


Figure 2: Vertex embeddings of Zachary's karate club network with GCNs with 1,2,3,4,5 layers.

Why skip connection works?

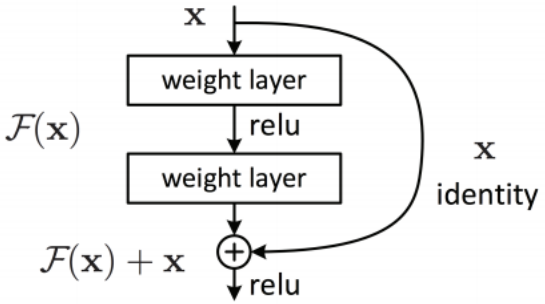
- **A standard GCN layer**
- $\mathbf{h}_v^{(l)} = \sigma\left(\sum_{u \in N(v)} \mathbf{W}^{(l)} \frac{\mathbf{h}_u^{(l-1)}}{|N(v)|}\right)$

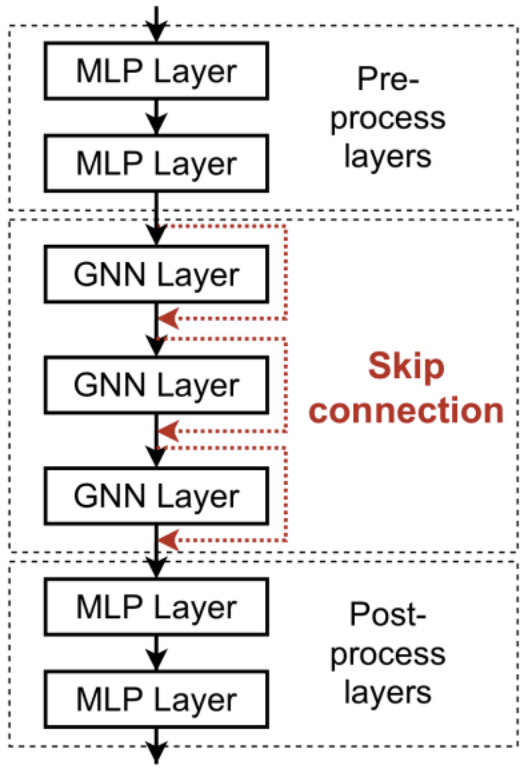
This is our $F(\mathbf{x})$

Why skip connection works?

- **A standard GCN layer**
- $\mathbf{h}_v^{(l)} = \sigma\left(\sum_{u \in N(v)} \mathbf{W}^{(l)} \frac{\mathbf{h}_u^{(l-1)}}{|N(v)|}\right)$

This is our $F(\mathbf{x})$





Why skip connection works?

- **A standard GCN layer**

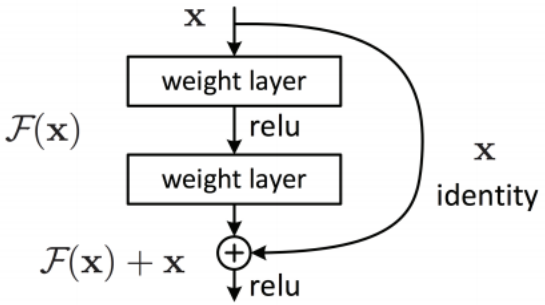
- $\mathbf{h}_v^{(l)} = \sigma\left(\sum_{u \in N(v)} \mathbf{W}^{(l)} \frac{\mathbf{h}_u^{(l-1)}}{|N(v)|}\right)$

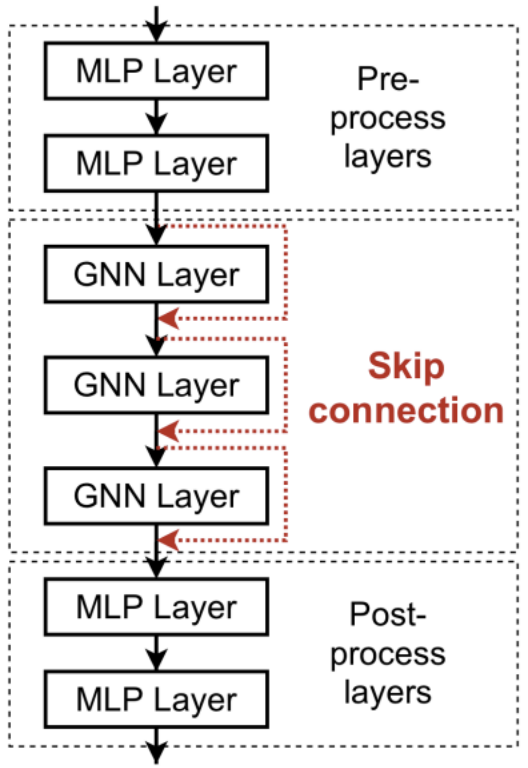
This is our $F(\mathbf{x})$

- **A GCN layer with skip connection**

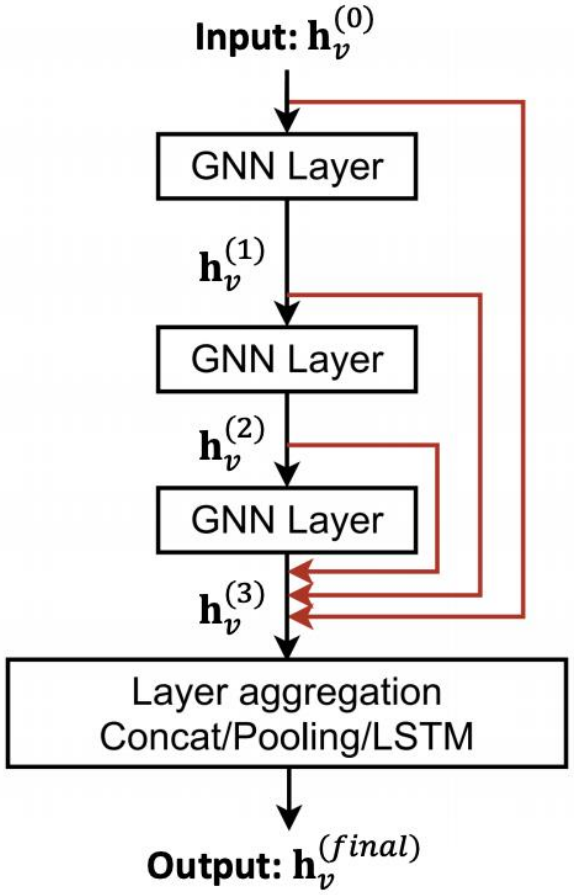
- $\mathbf{h}_v^{(l)} = \sigma\left(\sum_{u \in N(v)} \mathbf{W}^{(l)} \frac{\mathbf{h}_u^{(l-1)}}{|N(v)|} + \mathbf{h}_v^{(l-1)}\right)$

$F(\mathbf{x})$ + $\mathbf{x}$

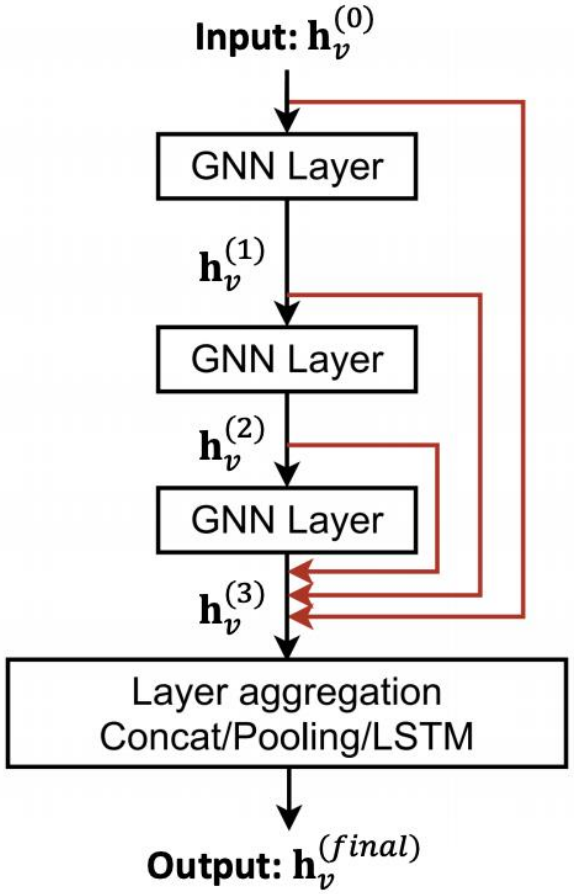




Other ways to add skip connections



Other ways to add skip connections



Oversmoothing is still an open problem