On the Bottleneck of Graph Neural Networks and its Practical Implications


Uri Alon Eran Yahav

TECHNION | The Henry and Marilyn Taub
Faculty of Computer Science

On the Bottleneck of Graph Neural Networks and its Practical Implications

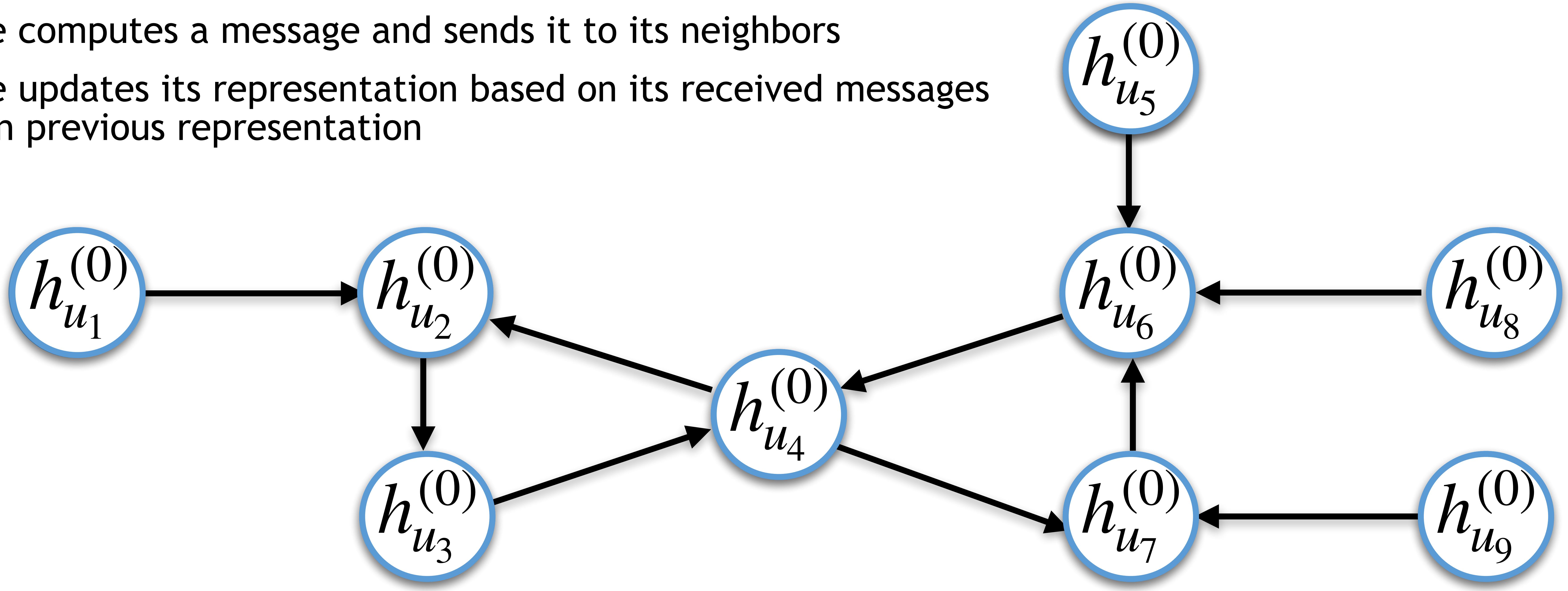
Uri Alon Eran Yahav

<u>Main Contribution</u>: GNNs suffer from a **bottleneck** that causes **over-squashing** when trying to capture long-range interactions

TECHNION | The Henry and Marilyn Taub
Faculty of Computer Science

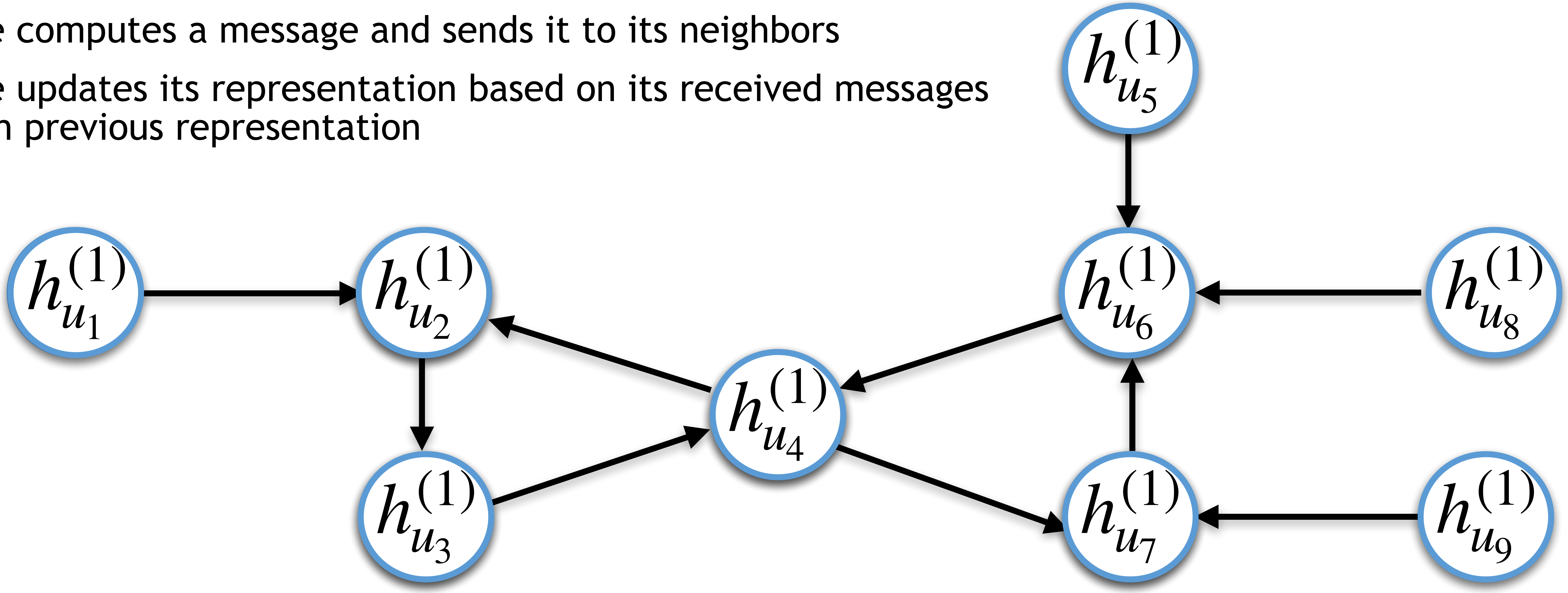
A GNN as a Message Passing Network [Gilmer, ICML'2017]

- Initial representations are embeddings or features
- At every message passing step (=layer):
  - Every node computes a message and sends it to its neighbors
  - Every node updates its representation based on its received messages and its own previous representation



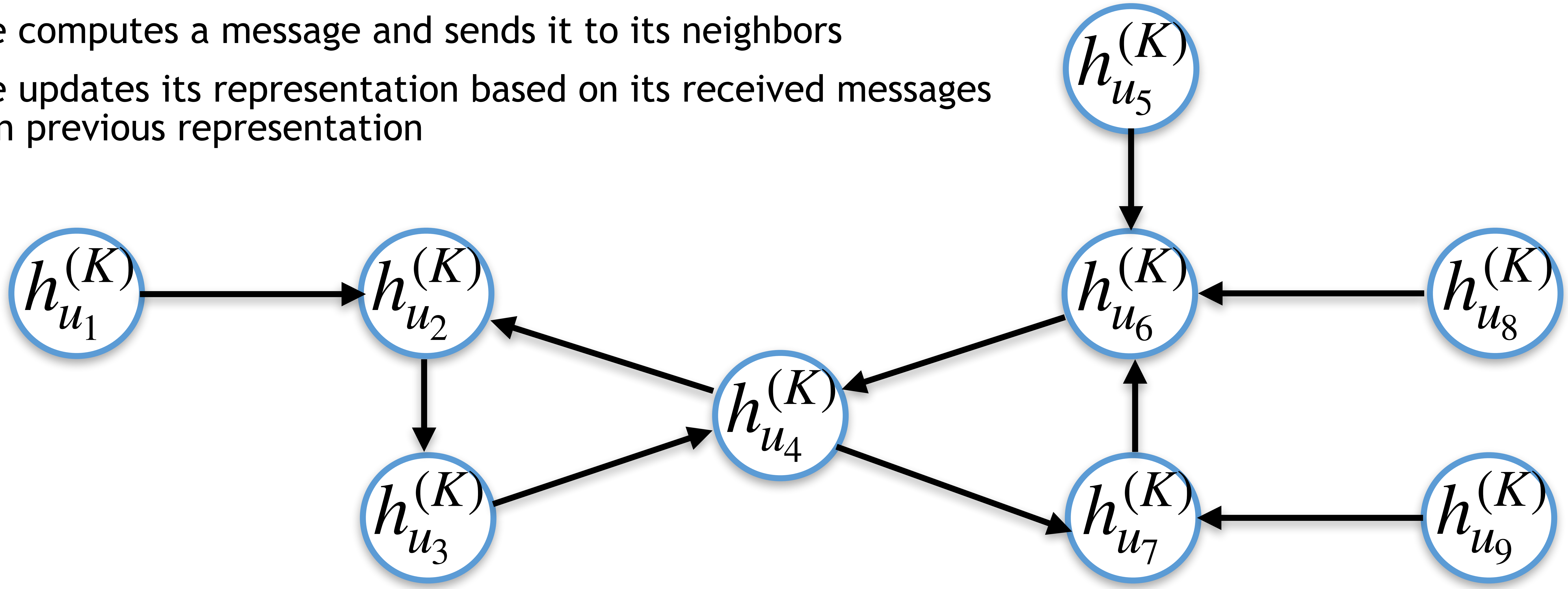
A GNN as a Message Passing Network [Gilmer, ICML'2017]

- Initial representations are embeddings or features
- At every message passing step (=layer):
  - Every node computes a message and sends it to its neighbors
  - Every node updates its representation based on its received messages and its own previous representation



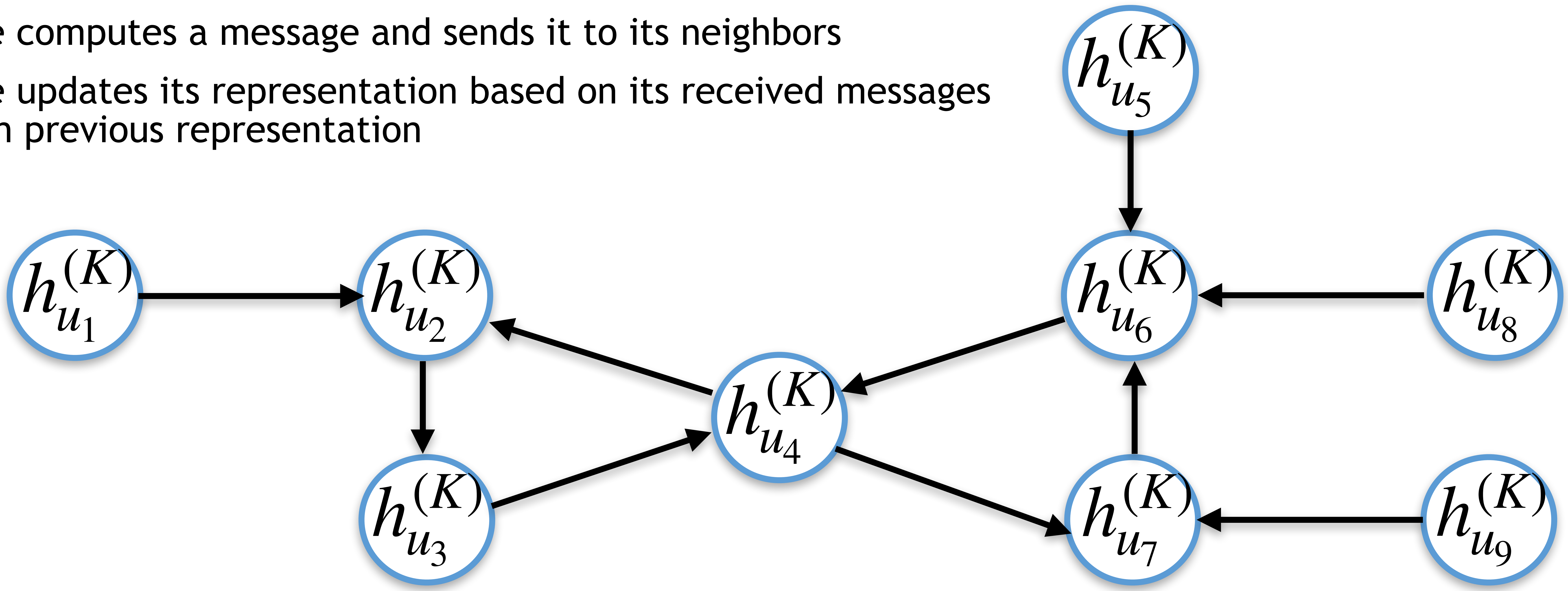
A GNN as a Message Passing Network [Gilmer, ICML'2017]

- Initial representations are embeddings or features
- At every message passing step (=layer):
  - Every node computes a message and sends it to its neighbors
  - Every node updates its representation based on its received messages and its own previous representation



A GNN as a Message Passing Network [Gilmer, ICML'2017]

- Initial representations are embeddings or features
- At every message passing step (=layer):
  - Every node computes a message and sends it to its neighbors
  - Every node updates its representation based on its received messages and its own previous representation



- Given $\{h_u^{(K)} \mid u \in V\}$:
  - Node classification, graph classification, link prediction...

What are graph neural networks good for?

- GNNs are good for **short-range** tasks:

What are graph neural networks good for?

- GNNs are good for **short-range** tasks:
  - Paper subject classification (Cora/Citeseer/Pubmed, Sen et al., 2008)

What are graph neural networks good for?

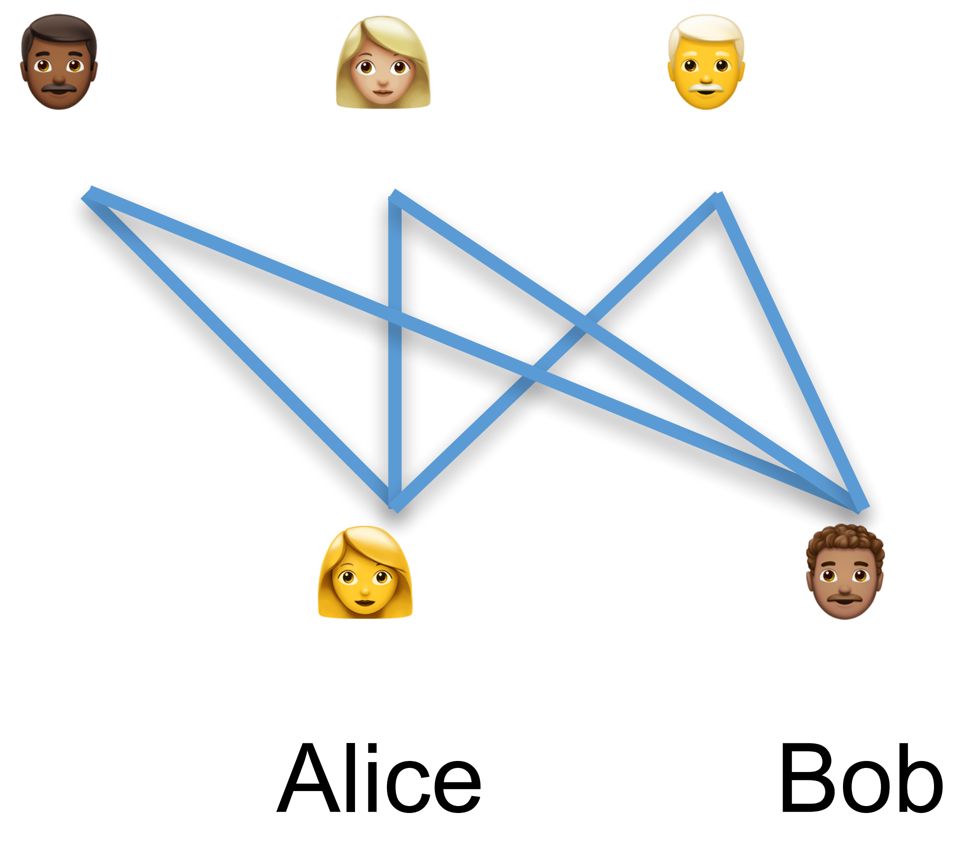
- GNNs are good for **short-range** tasks:
  - Paper subject classification (Cora/Citeseer/Pubmed, Sen et al., 2008)
  - Friendship/collaboration prediction (Open Graph Benchmark, Hu et al. 2020):

What are graph neural networks good for?

- GNNs are good for **short-range** tasks:
  - Paper subject classification (Cora/Citeseer/Pubmed, Sen et al., 2008)
  - Friendship/collaboration prediction (Open Graph Benchmark, Hu et al. 2020):

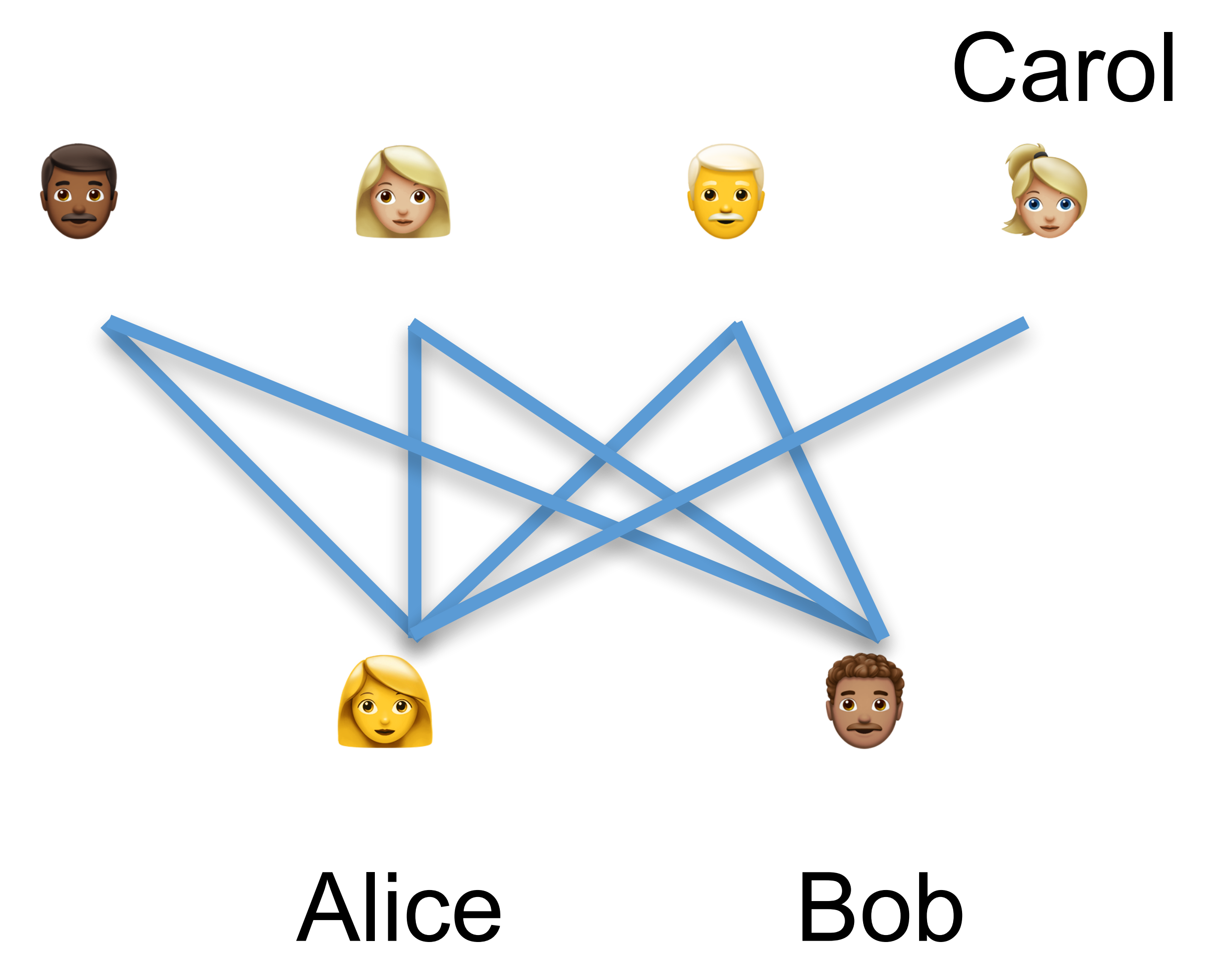
What are graph neural networks good for?

- GNNs are good for **short-range** tasks:
  - Paper subject classification (Cora/Citeseer/Pubmed, Sen et al., 2008)
  - Friendship/collaboration prediction (Open Graph Benchmark, Hu et al. 2020):



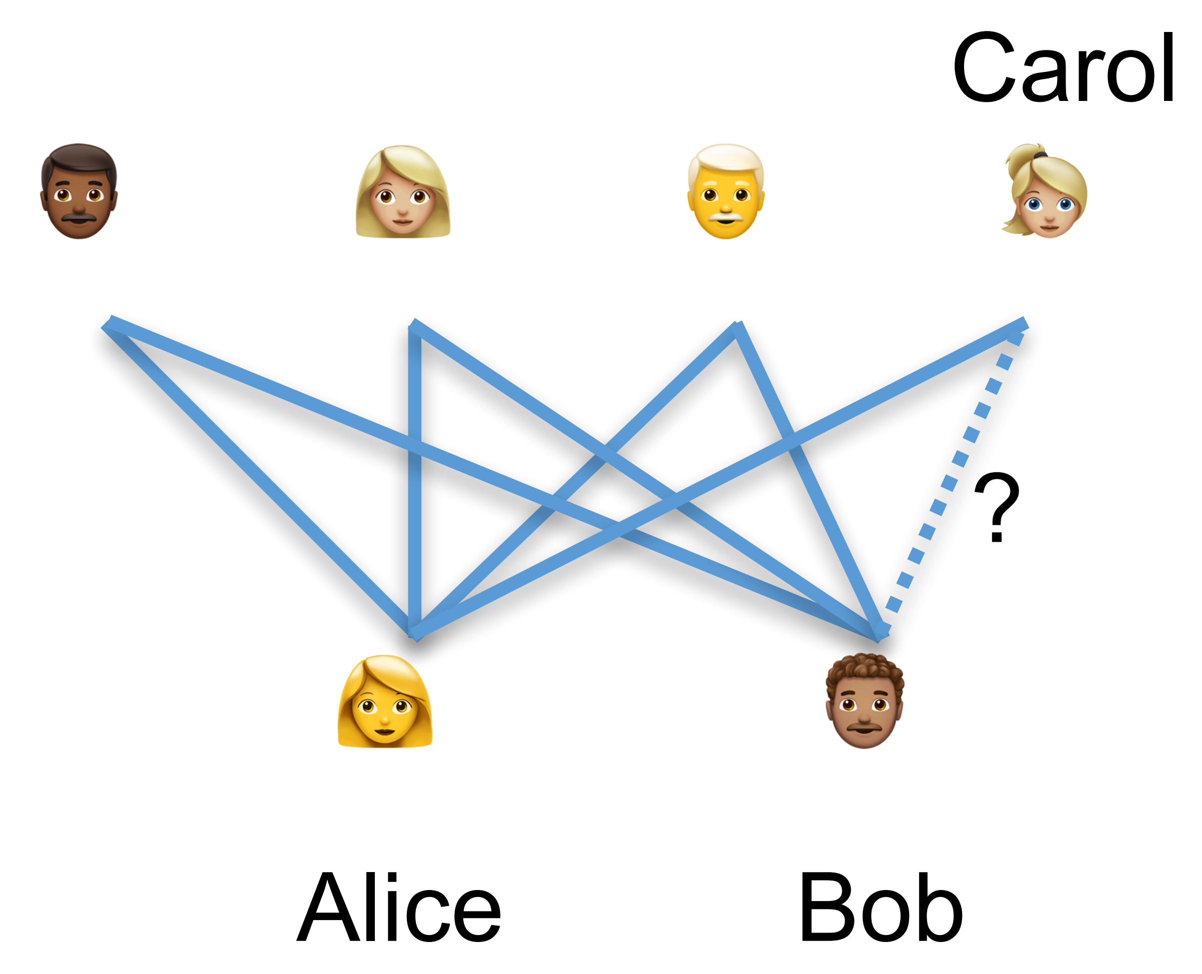
What are graph neural networks good for?

- GNNs are good for **short-range** tasks:
  - Paper subject classification (Cora/Citeseer/Pubmed, Sen et al., 2008)
  - Friendship/collaboration prediction (Open Graph Benchmark, Hu et al. 2020):



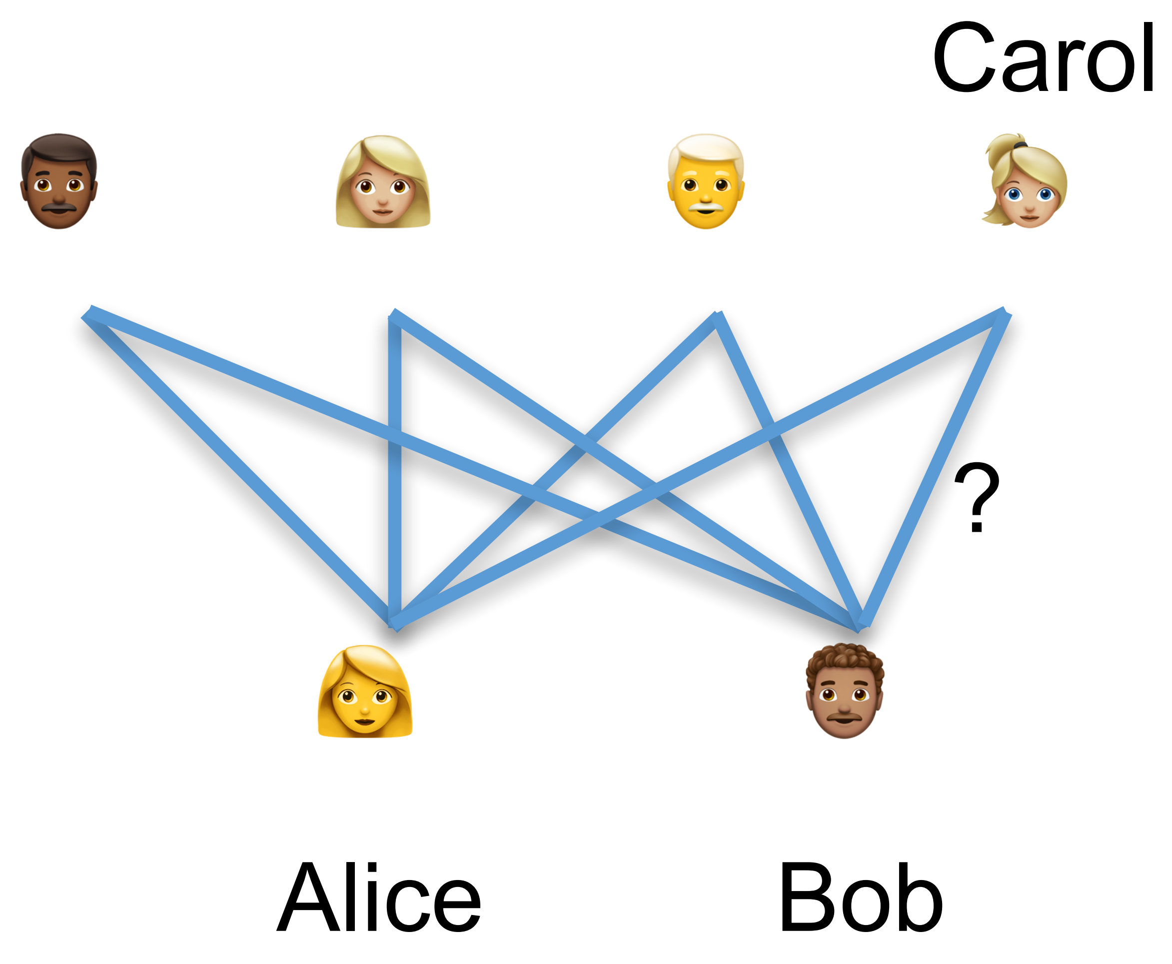
What are graph neural networks good for?

- GNNs are good for **short-range** tasks:
  - Paper subject classification (Cora/Citeseer/Pubmed, Sen et al., 2008)
  - Friendship/collaboration prediction (Open Graph Benchmark, Hu et al. 2020):



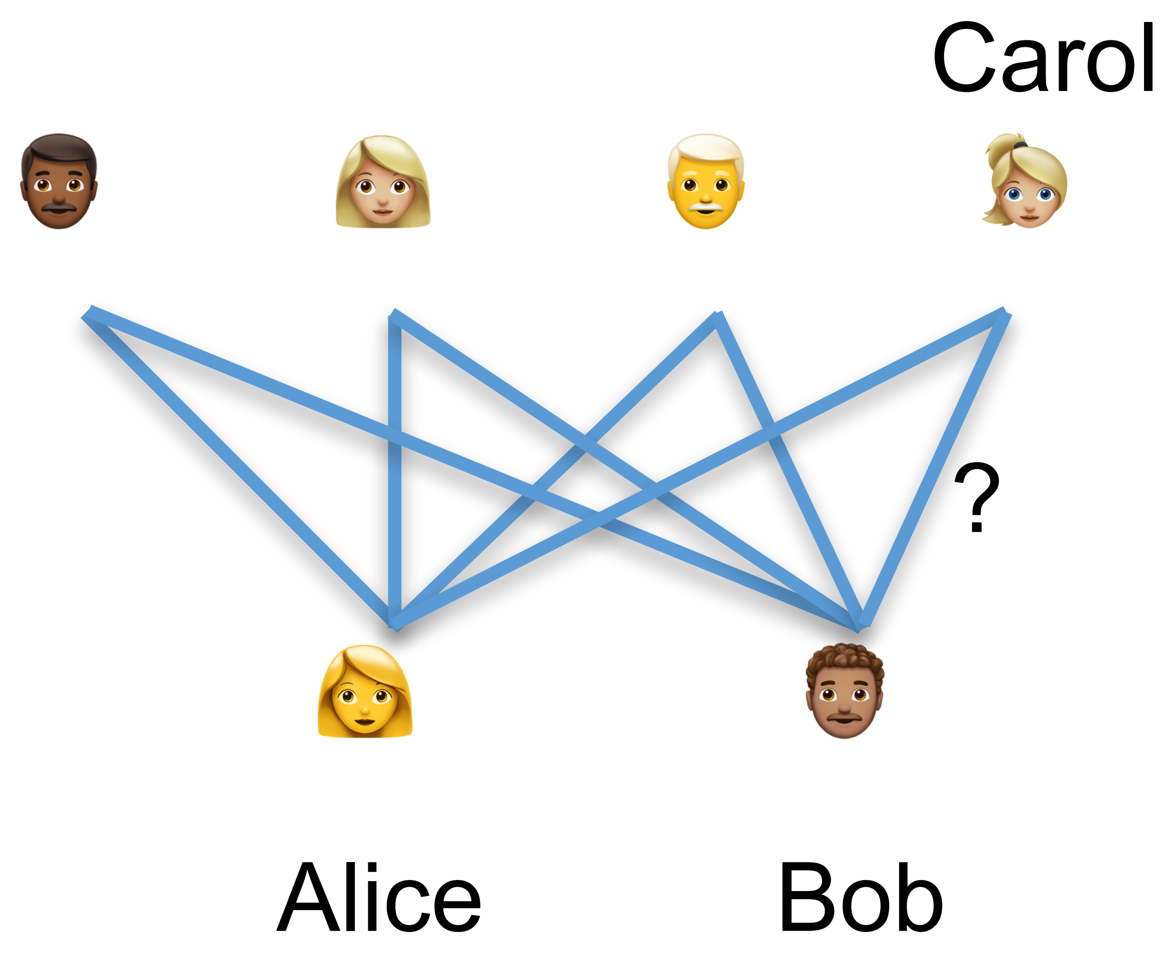
What are graph neural networks good for?

- GNNs are good for **short-range** tasks:
  - Paper subject classification (Cora/Citeseer/Pubmed, Sen et al., 2008)
  - Friendship/collaboration prediction (Open Graph Benchmark, Hu et al. 2020):



What are graph neural networks good for?

- GNNs are good for **short-range** tasks:
  - Paper subject classification (Cora/Citeseer/Pubmed, Sen et al., 2008)
  - Friendship/collaboration prediction (Open Graph Benchmark, Hu et al. 2020):

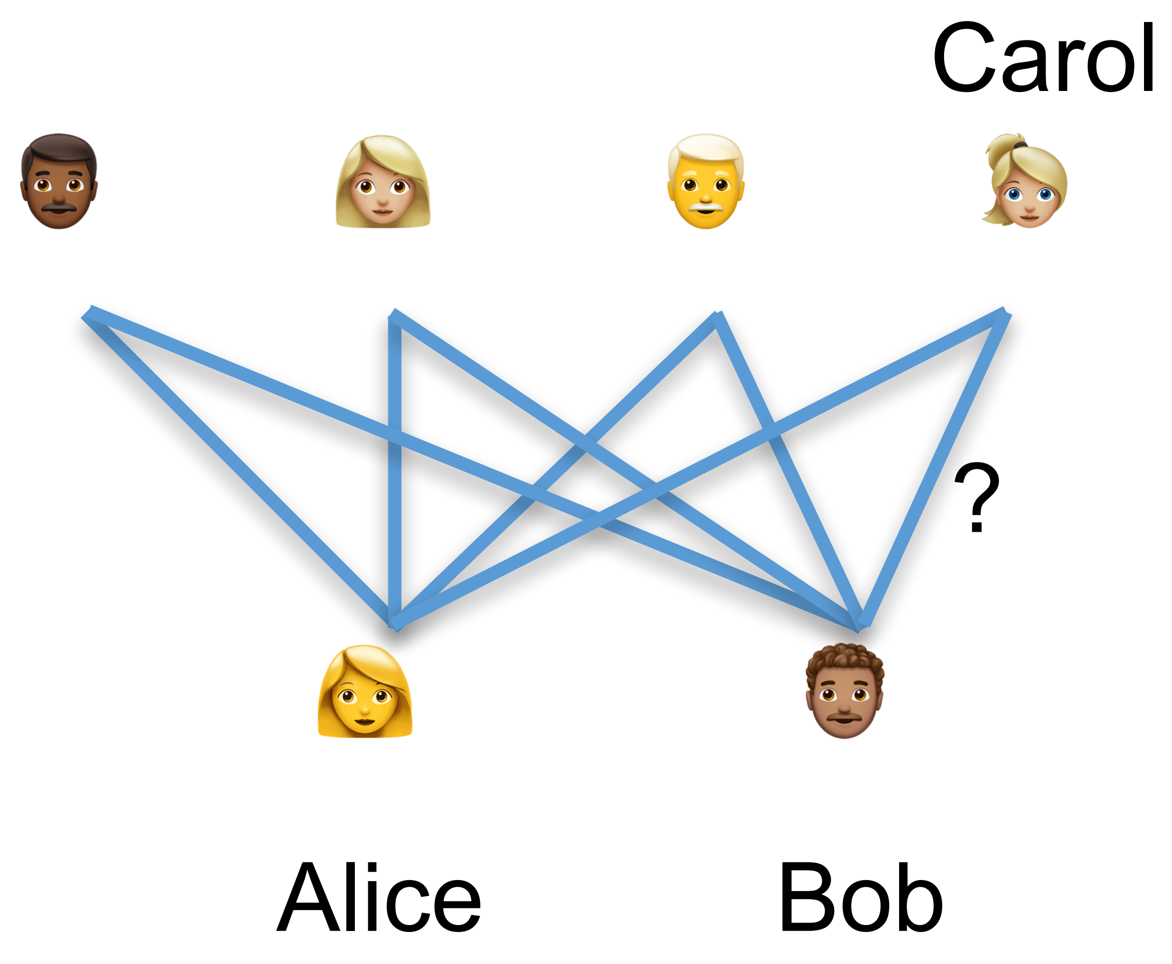


Very local property,

requires only 2-3 message-passing steps

What are graph neural networks good for?

- GNNs are good for **short-range** tasks:
  - Paper subject classification (Cora/Citeseer/Pubmed, Sen et al., 2008)
  - Friendship/collaboration prediction (Open Graph Benchmark, Hu et al. 2020):



Very local property,

requires only 2-3 message-passing steps

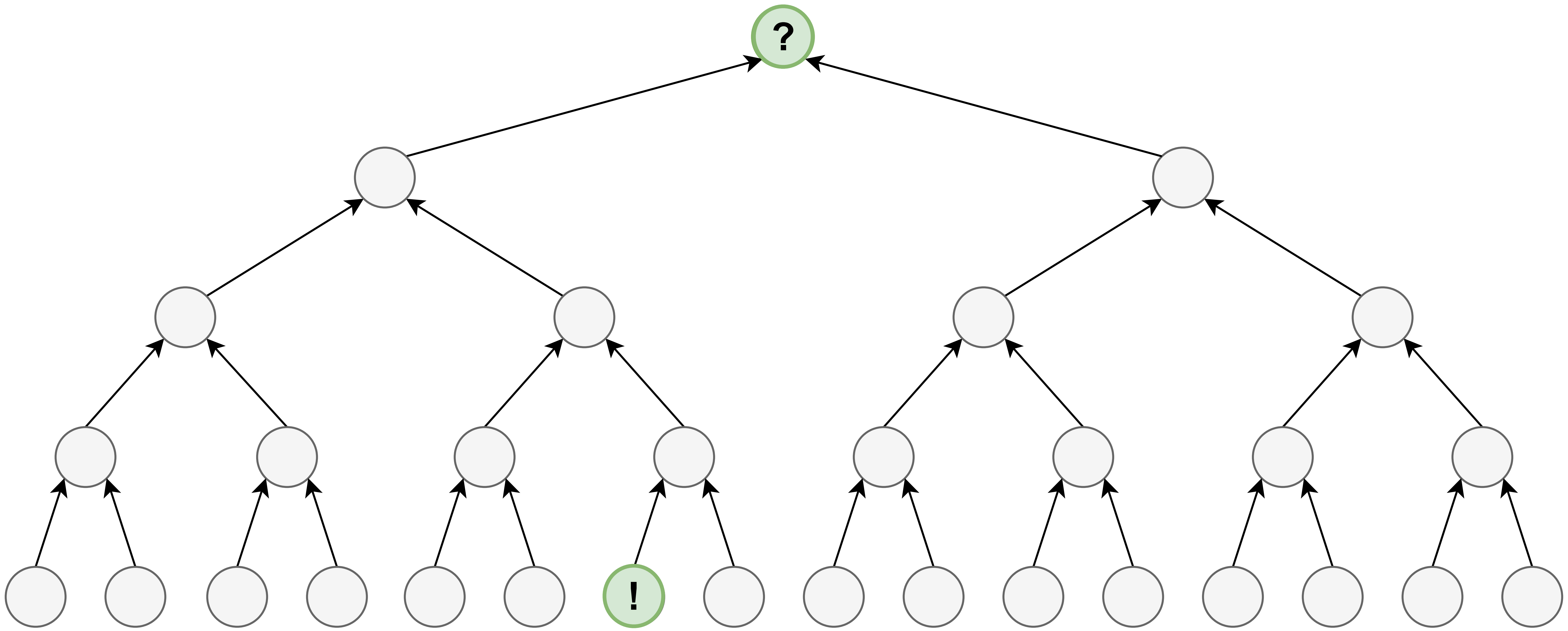
- But some tasks require **longer-range** interaction…

The GNN Bottleneck

Imagine that a prediction of a node depends on information coming from a distant node.

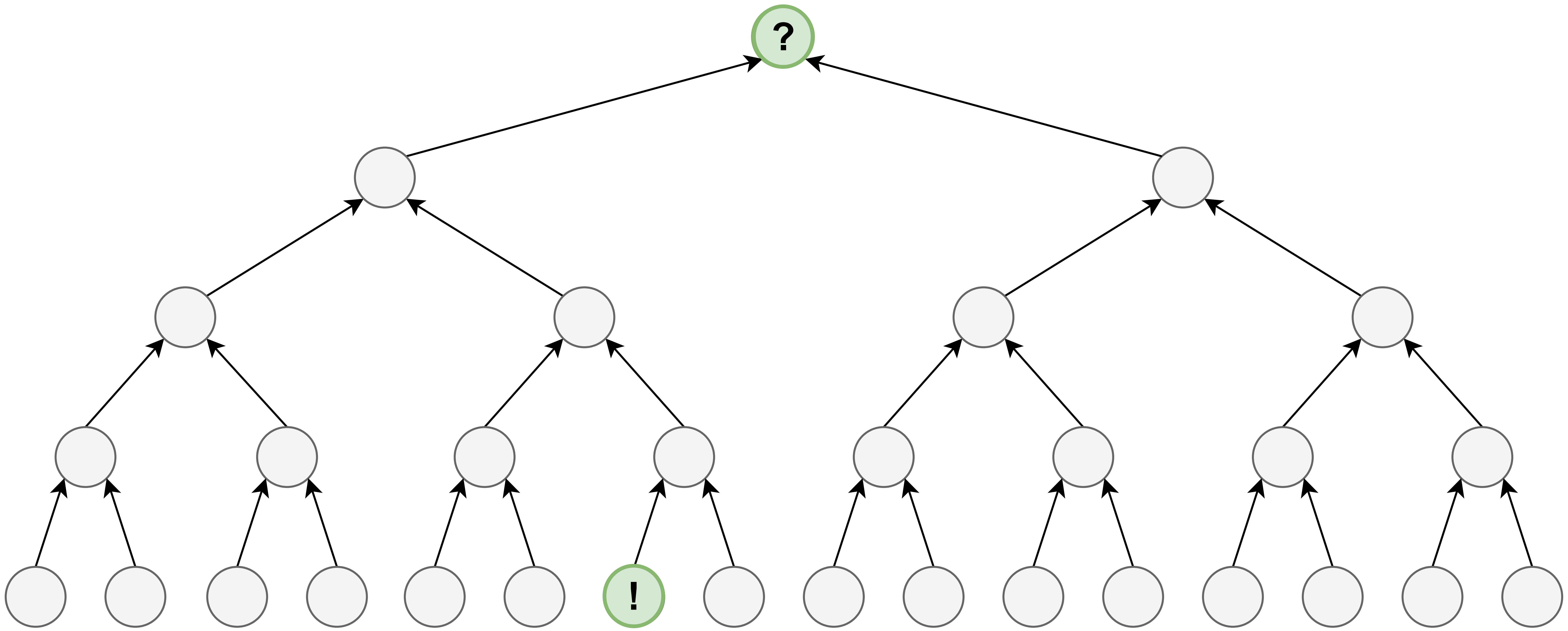
The GNN Bottleneck

Imagine that a prediction of a node depends on information coming from a distant node.



The GNN Bottleneck

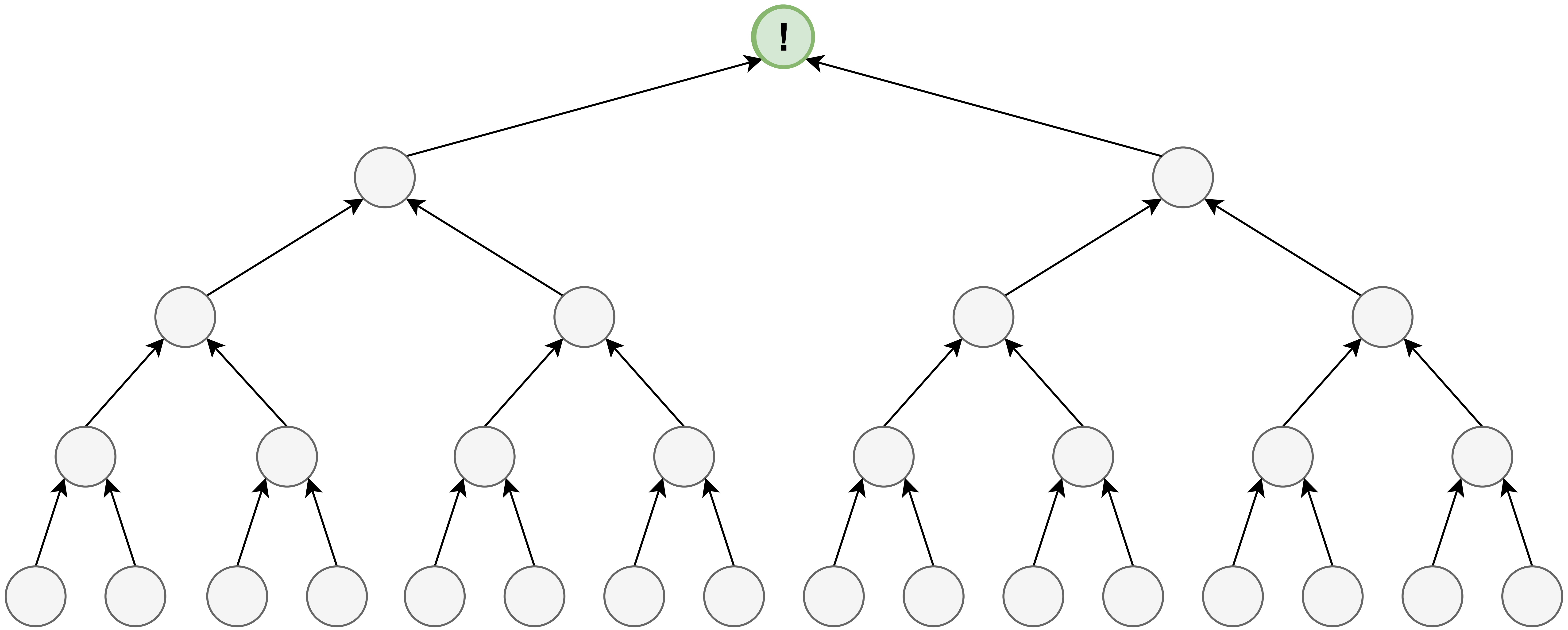
Imagine that a prediction of a node depends on information coming from a distant node.



We need: $Layers \geq Radius$

The GNN Bottleneck

Imagine that a prediction of a node depends on information coming from a distant node.



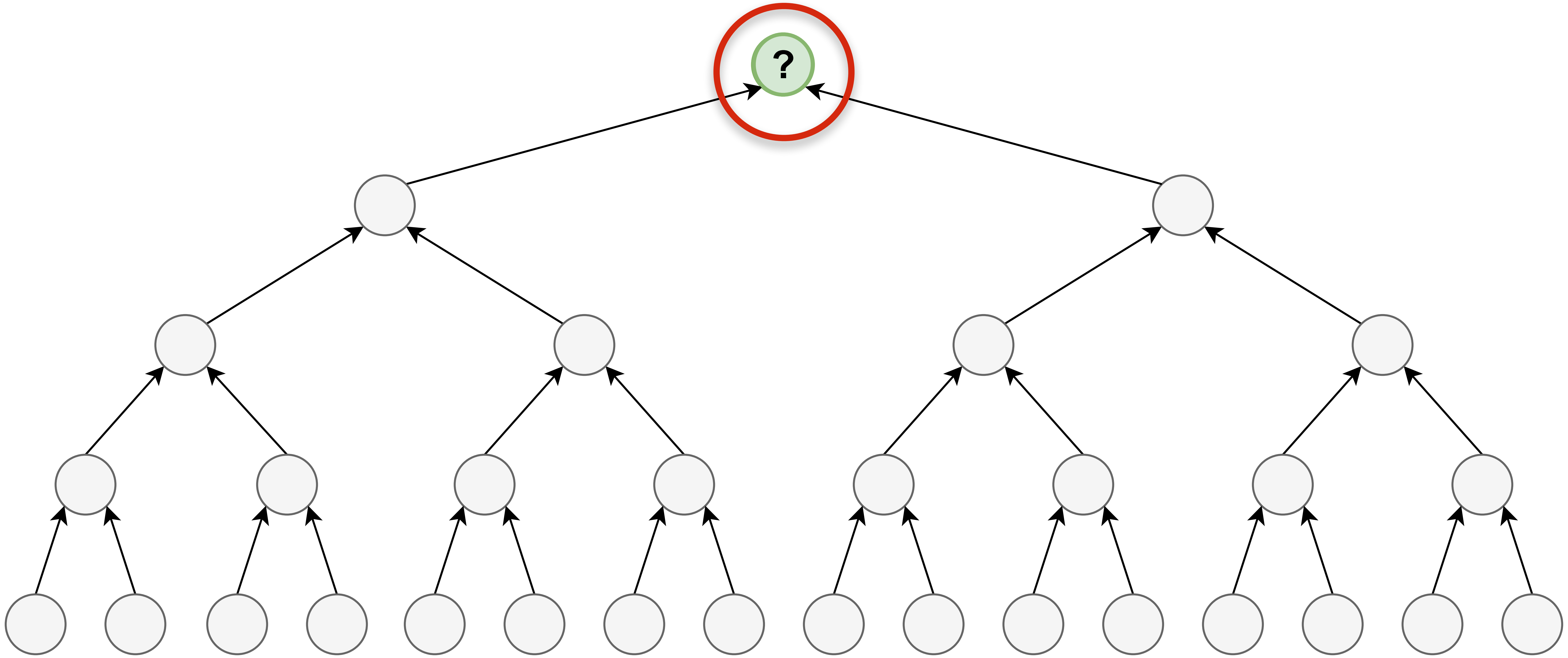
We need: $Layers \geq Radius$

In this case, we need at least 4 GNN layers for distant information to reach the target node.

The GNN Bottleneck

Imagine that a prediction of a node depends on information coming from a distant node.

t=0



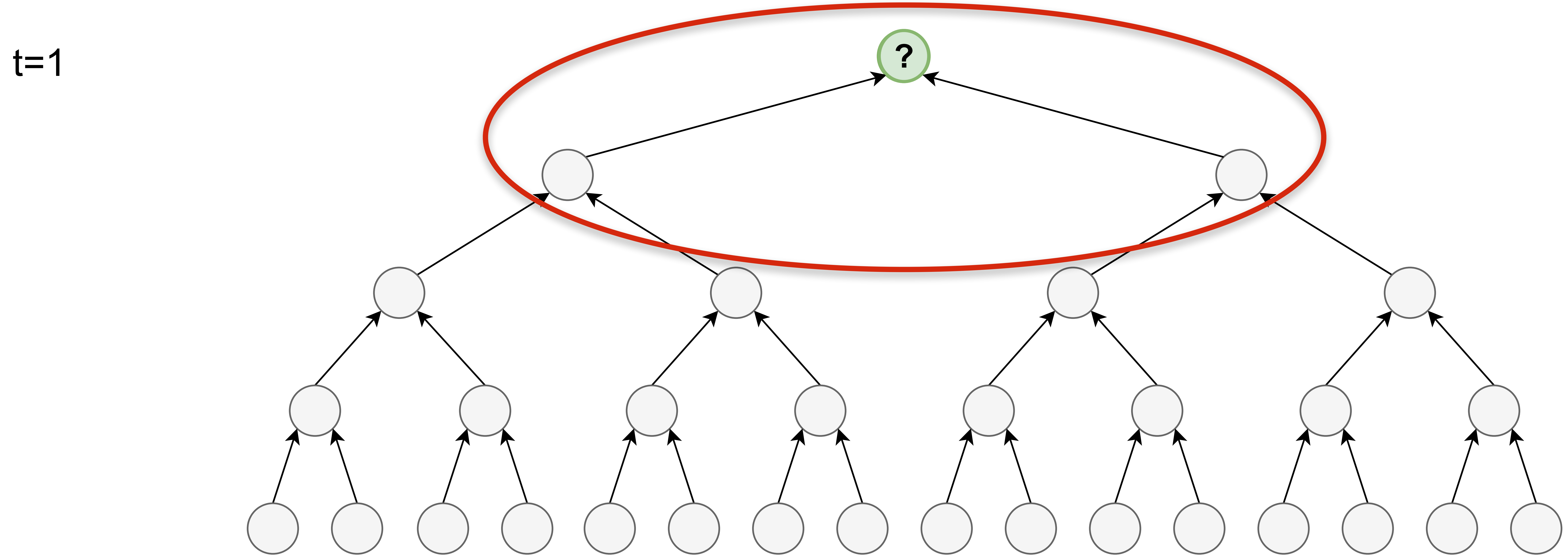
We need: $Layers \geq Radius$

In this case, we need at least 4 GNN layers for distant information to reach the target node.

However, the receptive field of the target node grows **exponentially** with the number of layers

The GNN Bottleneck

Imagine that a prediction of a node depends on information coming from a distant node.



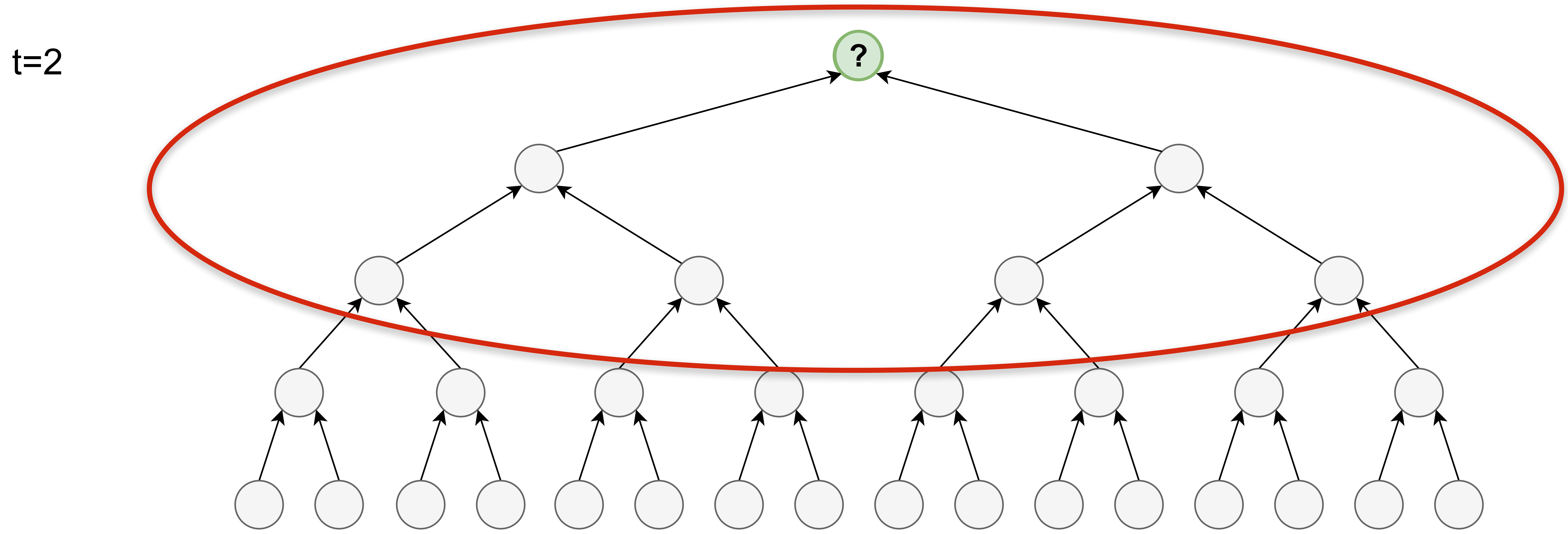
We need: $Layers \geq Radius$

In this case, we need at least **4** GNN layers for distant information to reach the target node.

However, the receptive field of the target node grows **exponentially** with the number of layers

The GNN Bottleneck

Imagine that a prediction of a node depends on information coming from a distant node.



We need: $Layers \geq Radius$

In this case, we need at least **4** GNN layers for distant information to reach the target node.

However, the receptive field of the target node grows **exponentially** with the number of layers

The GNN Bottleneck

Imagine that a prediction of a node depends on information coming from a distant node.

We need: $Layers \geq Radius$

In this case, we need at least **4** GNN layers for distant information to reach the target node.

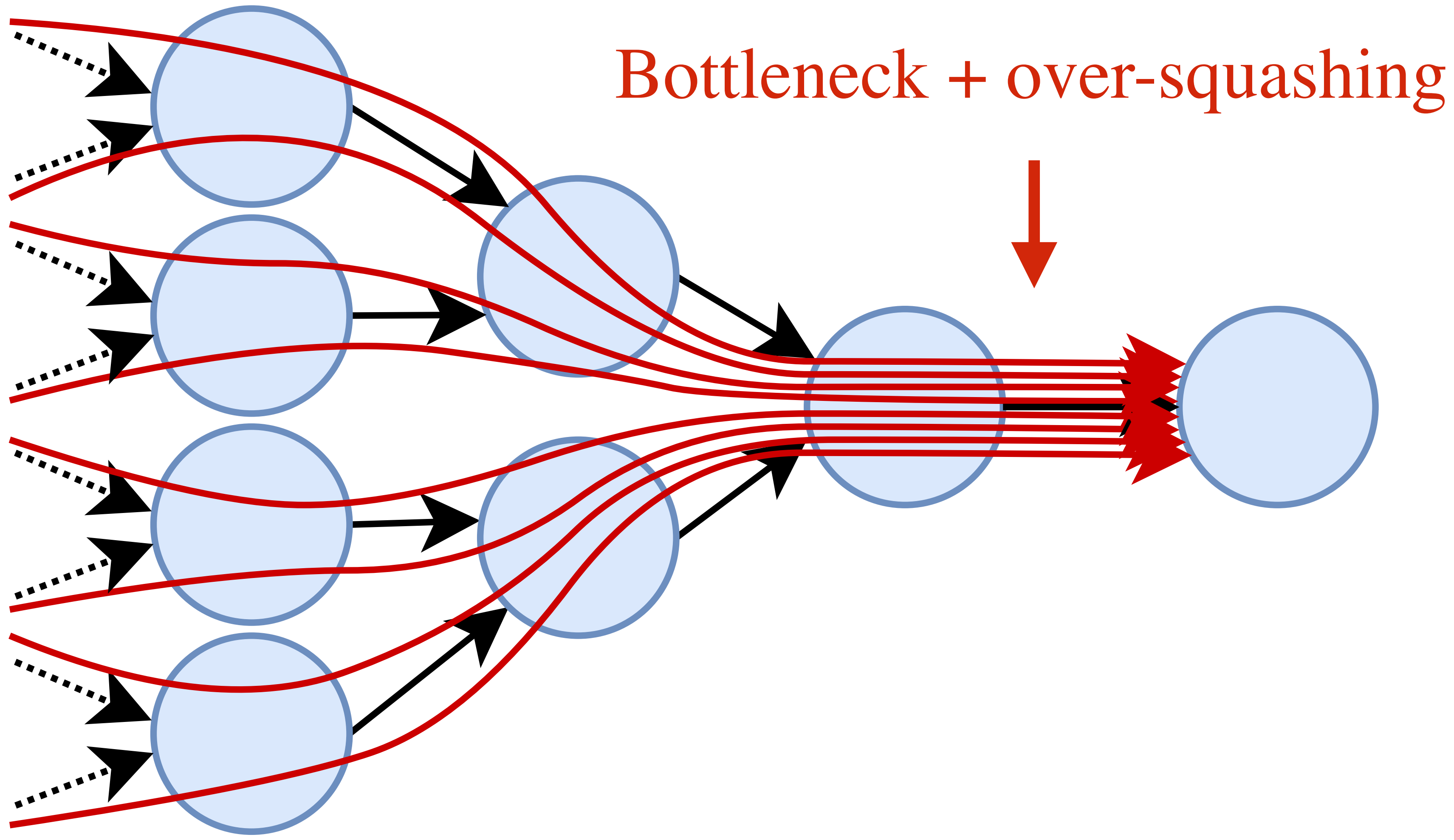
However, the receptive field of the target node grows **exponentially** with the number of layers

Over-squashing

To flow a message to a distance of 4, we need to squash $\mathbf{O}\left(\mathbf{degree}^4\right)$ messages into a single node vector.

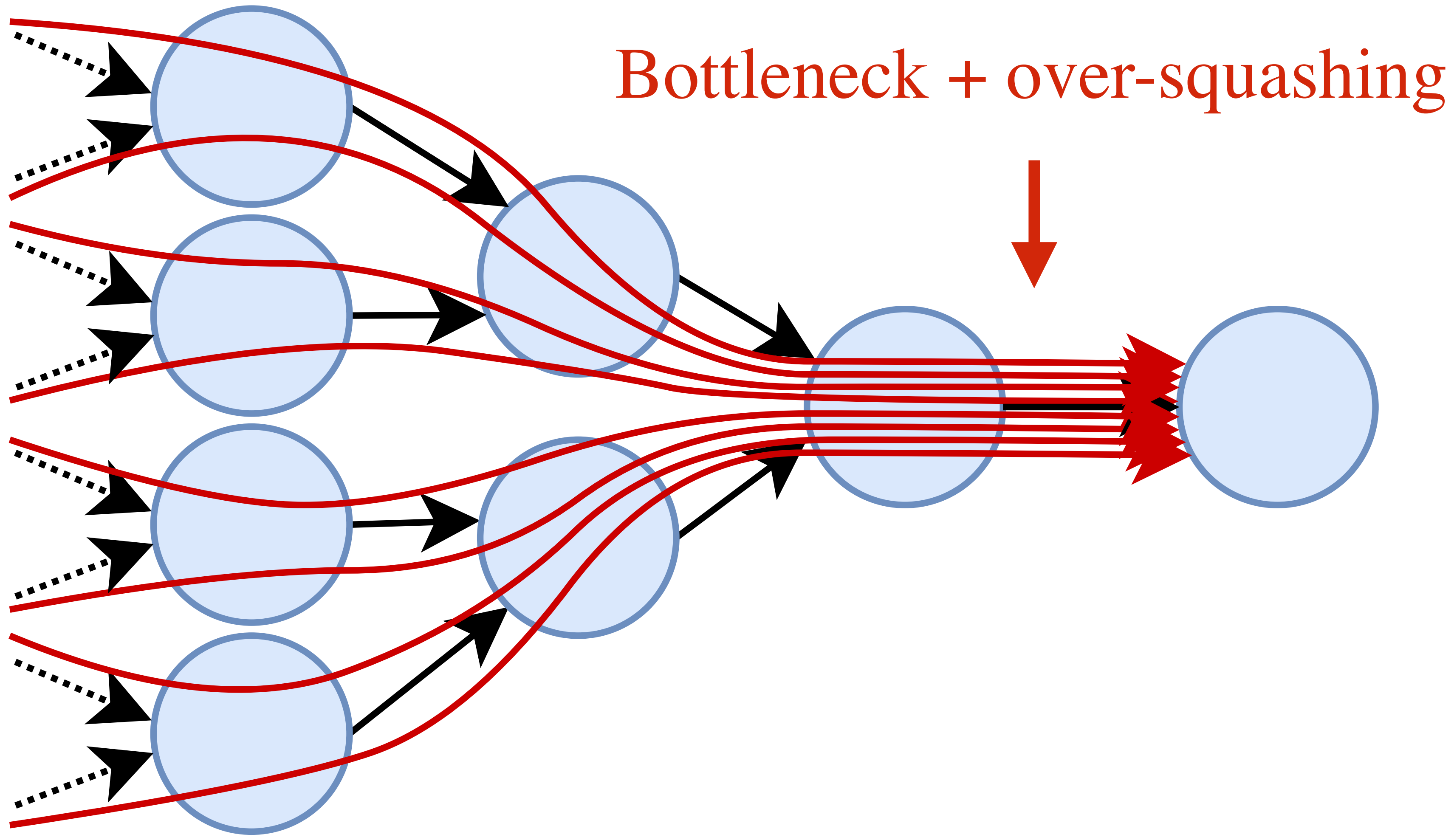
Over-squashing

To flow a message to a distance of 4, we need to squash $\mathbf{O}\left(\mathbf{degree}^4\right)$ messages into a single node vector.



Over-squashing

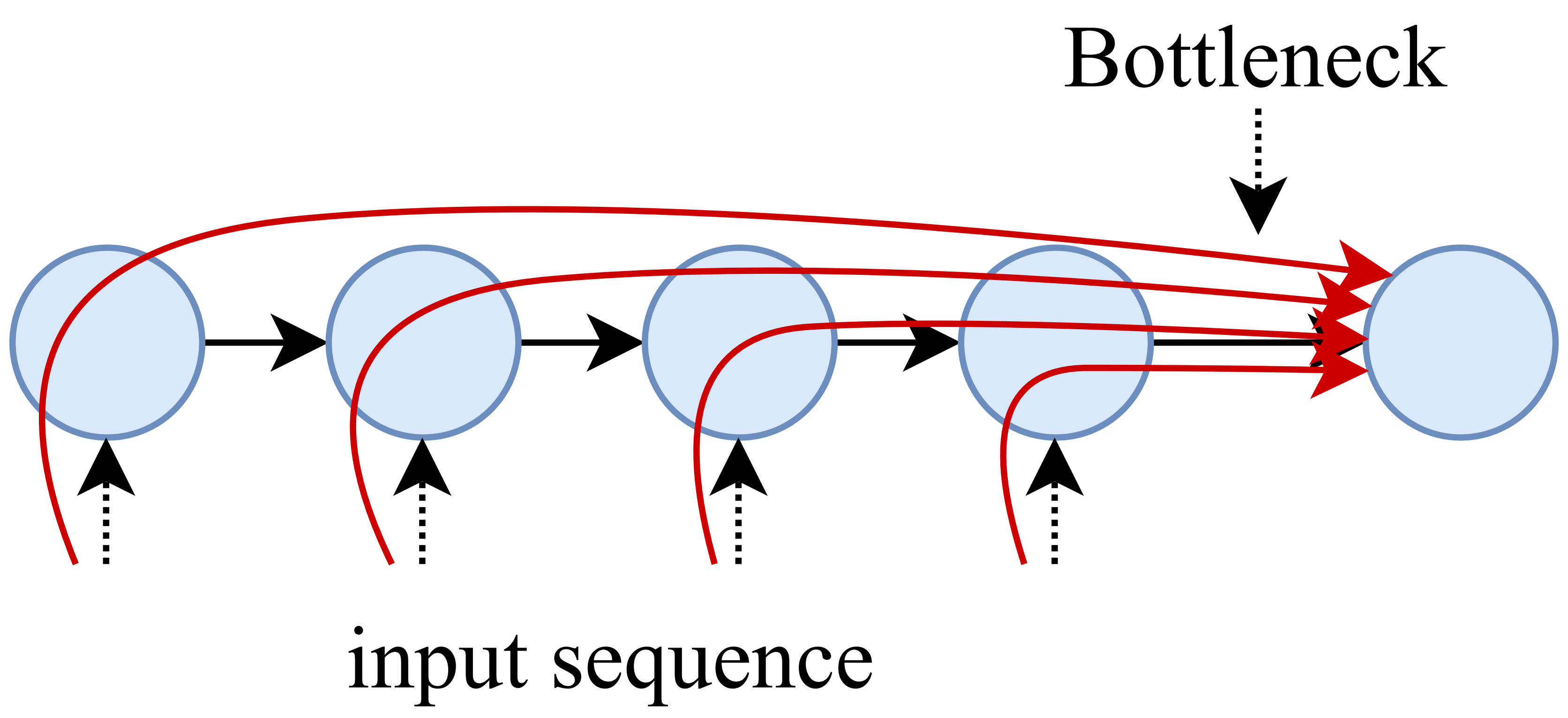
To flow a message to a distance of 4, we need to squash $\mathbf{O}\left(\mathbf{degree}^4\right)$ messages into a single node vector.



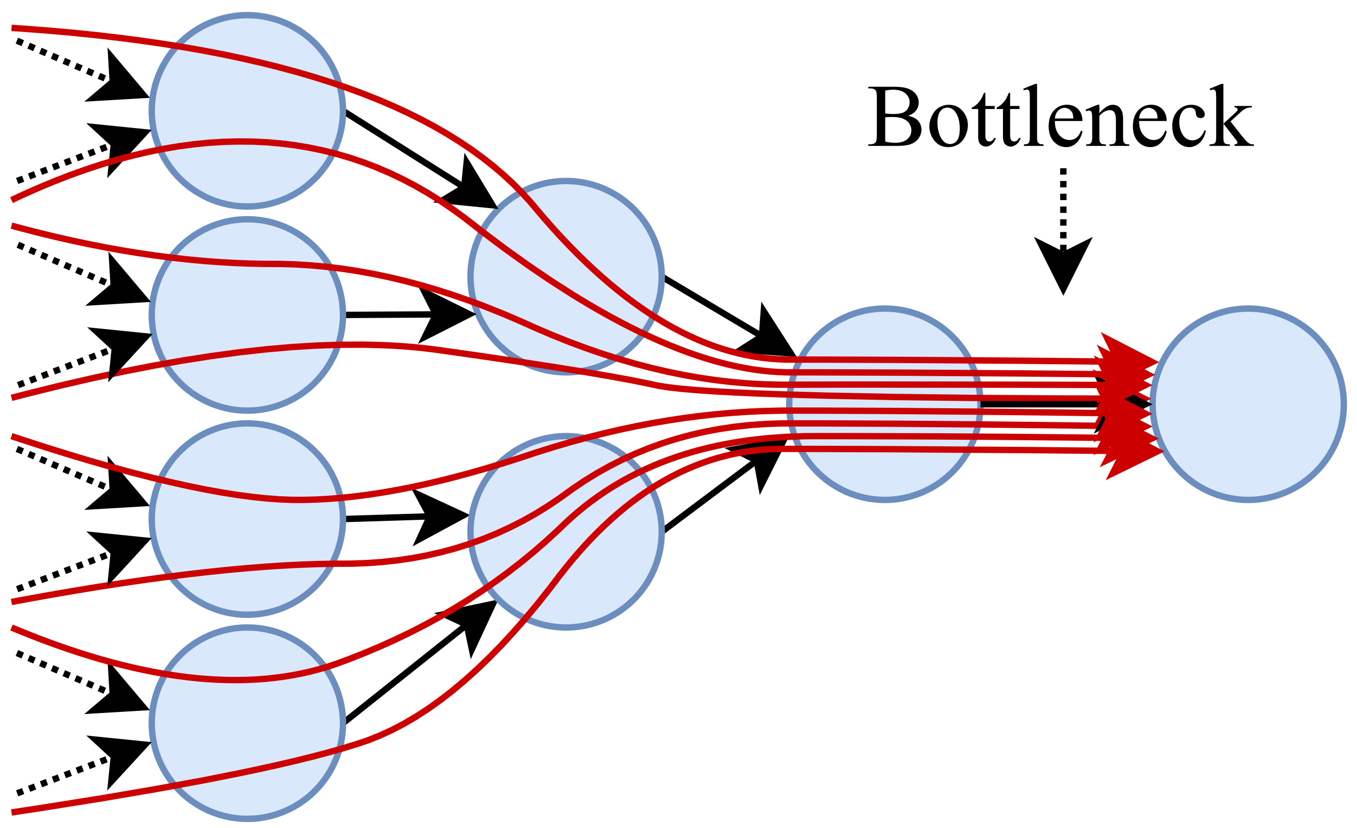
An exponential amount of information is squashed into a fixed-size vector.

Over-squashing

Actually, this is similar to the bottleneck of recurrent sequential models (before attention), except that the receptive field in RNNs grows **linearly**, while in GNNs it grows **exponentially**



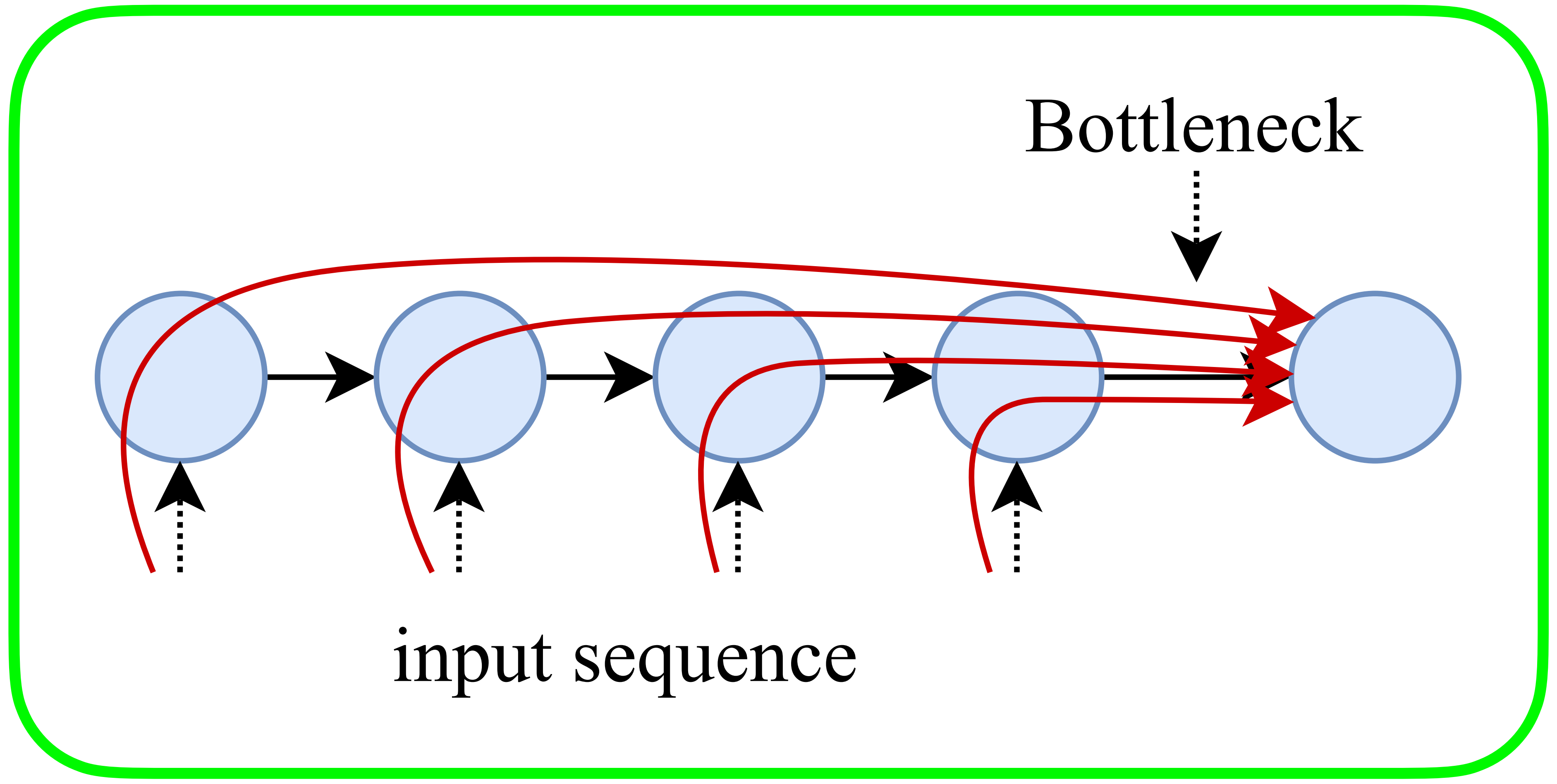
RNNs



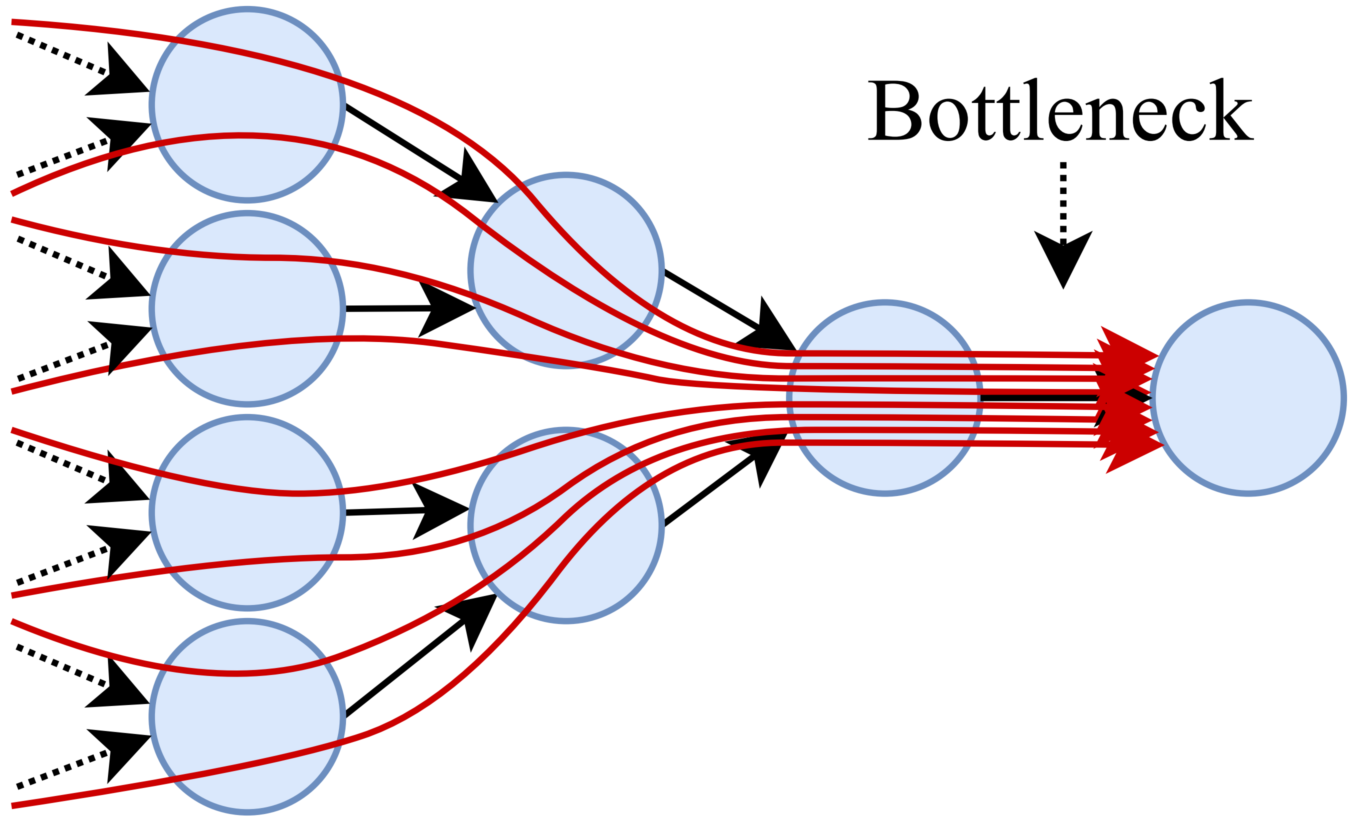
GNNs

Over-squashing

Actually, this is similar to the bottleneck of recurrent sequential models (before attention), except that the receptive field in RNNs grows **linearly**, while in GNNs it grows **exponentially**



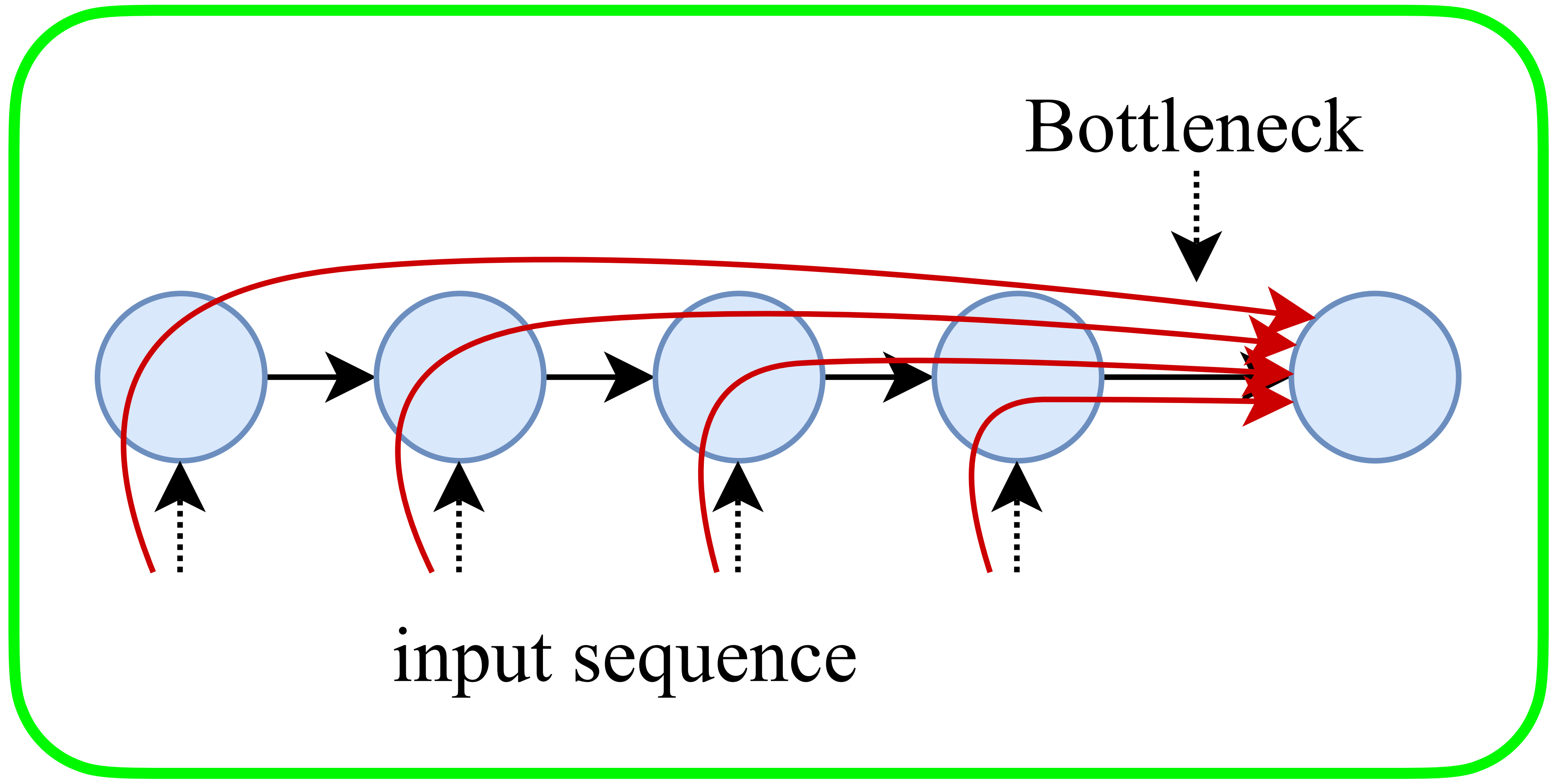
RNNs



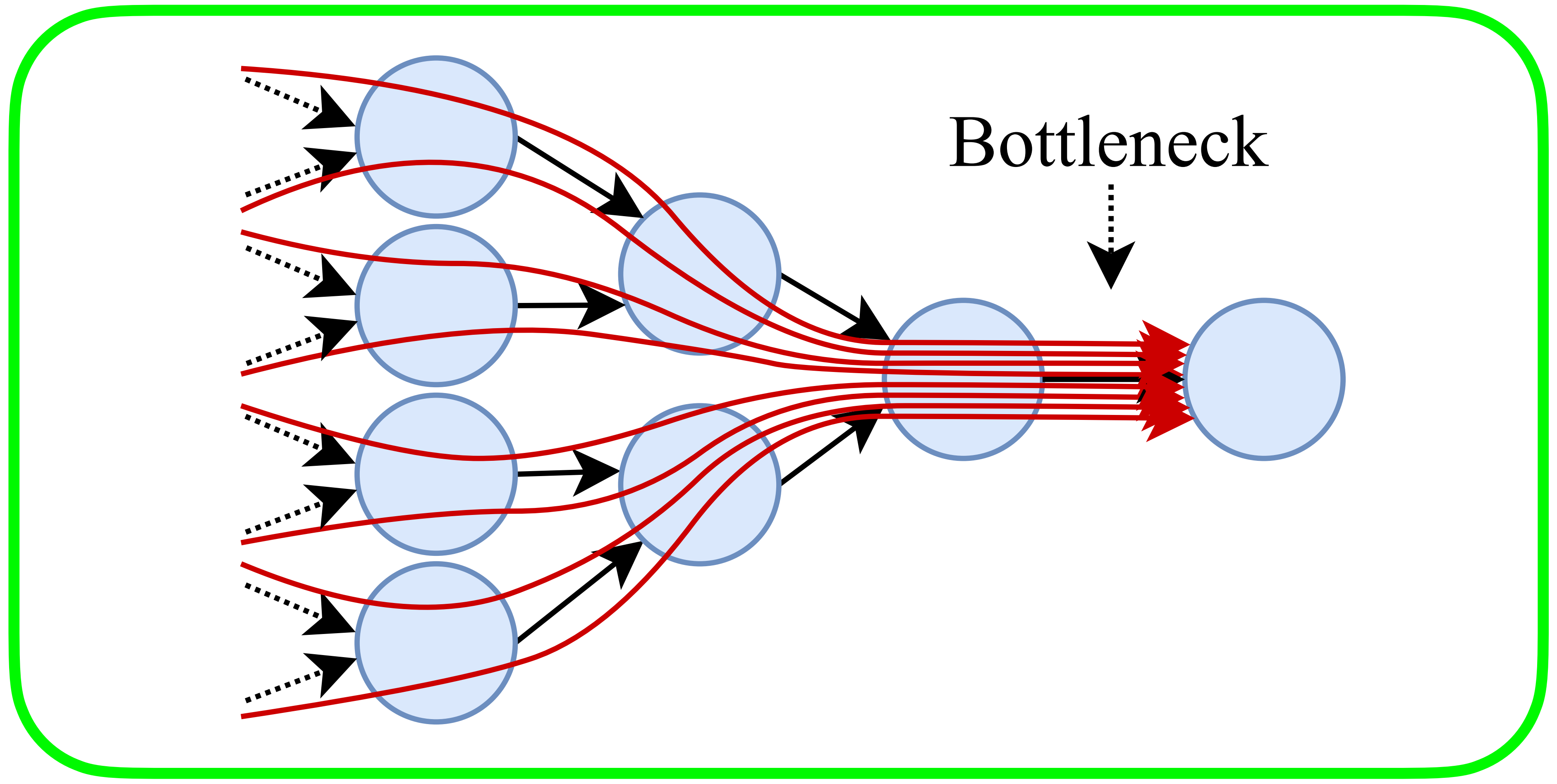
GNNs

Over-squashing

Actually, this is similar to the bottleneck of recurrent sequential models (before attention), except that the receptive field in RNNs grows linearly, while in GNNs it grows **exponentially**

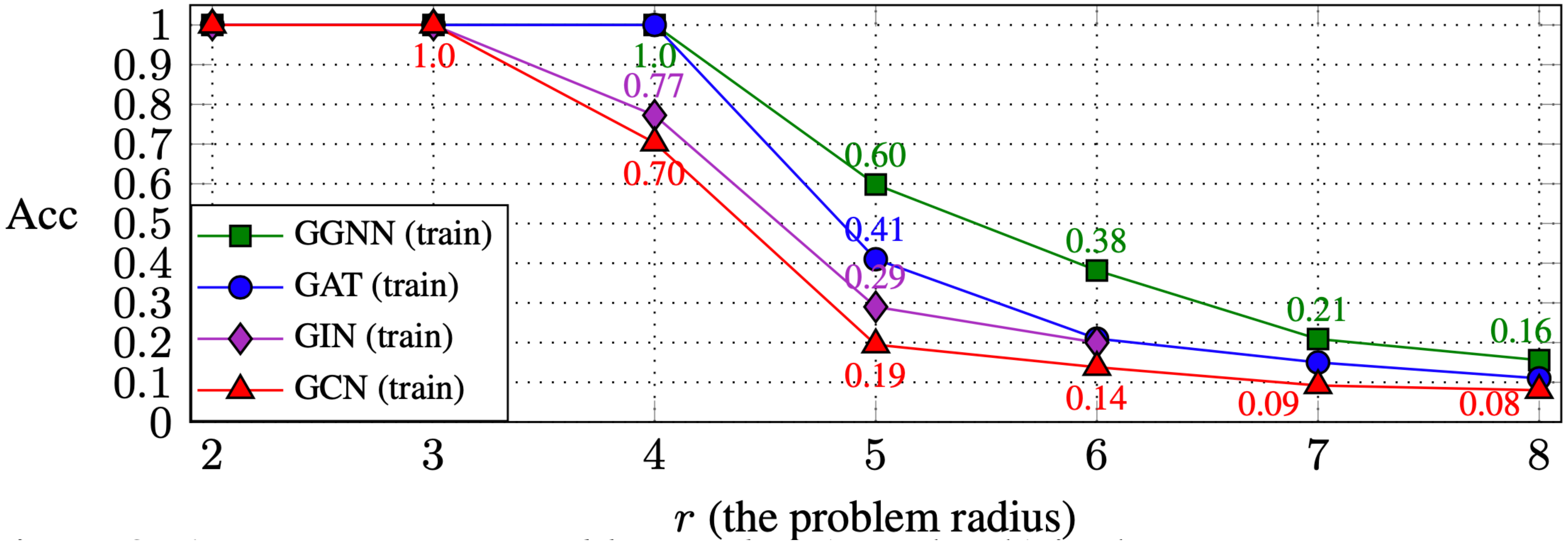


RNNs



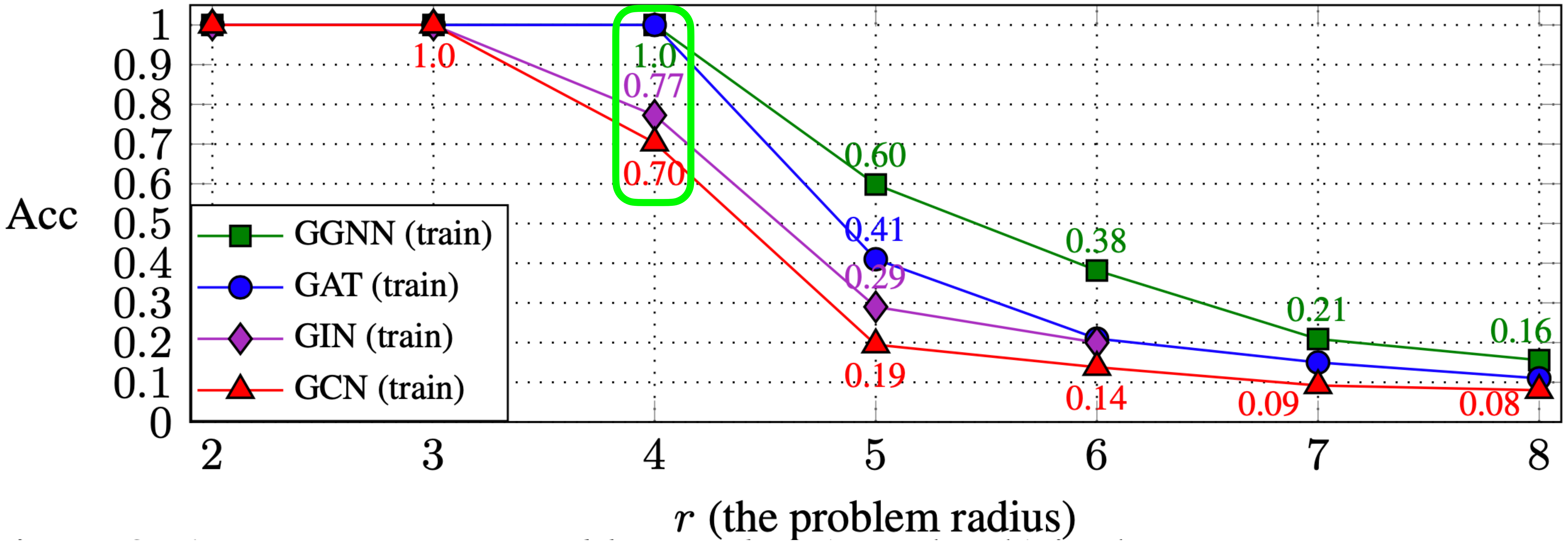
GNNs

Over-squashing prevents GNNs from **fitting the training data**



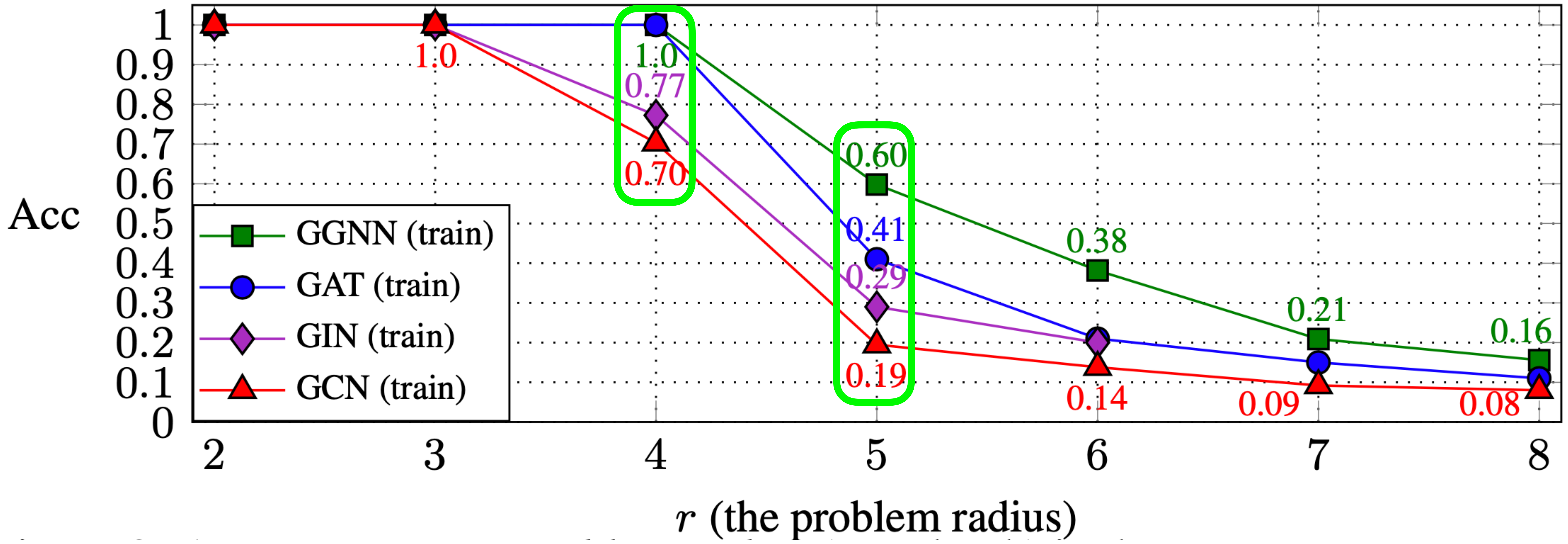
Over-squashing prevents GNNs from fitting the training data

- At a radius of **4**, some GNNs cannot even reach 100% **training** accuracy



Over-squashing prevents GNNs from fitting the training data

- At a radius of **4**, some GNNs cannot even reach 100% **training** accuracy
- At a radius of **5**, all GNNs could not reach 100% **training** accuracy



How long is “long-range”?

- Combinatorially, to fit the dataset: $2^{32\cdot d} > \frac{\left(2^r\right)!}{(2)^{2^r-1}}$

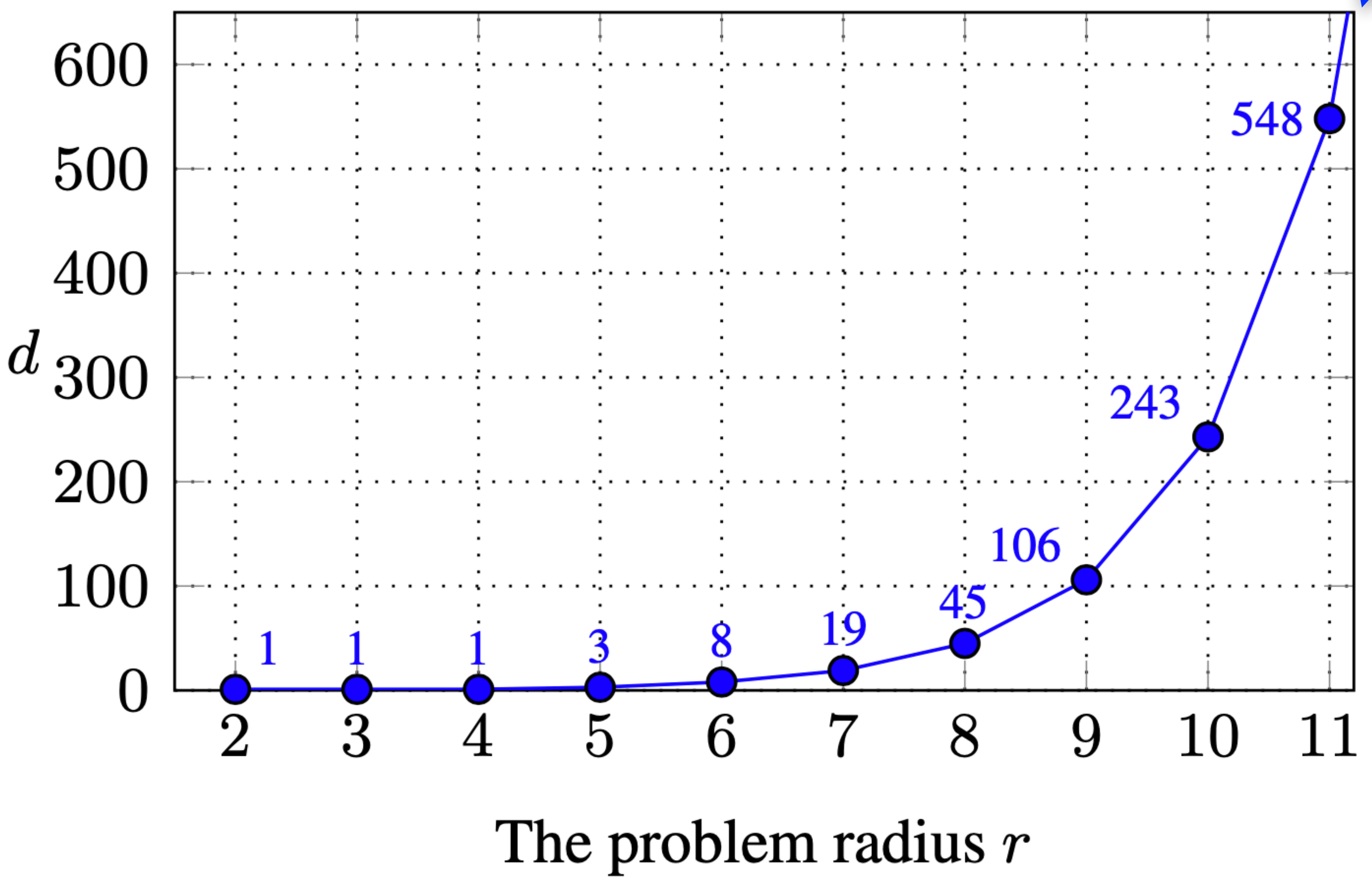
How long is “long-range”?

- Combinatorially, to fit the dataset: $\boxed{2^{32\cdot d} > \dfrac{\left(2^r\right)!}{(2)^{2^r-1}}}$

How long is “long-range”?

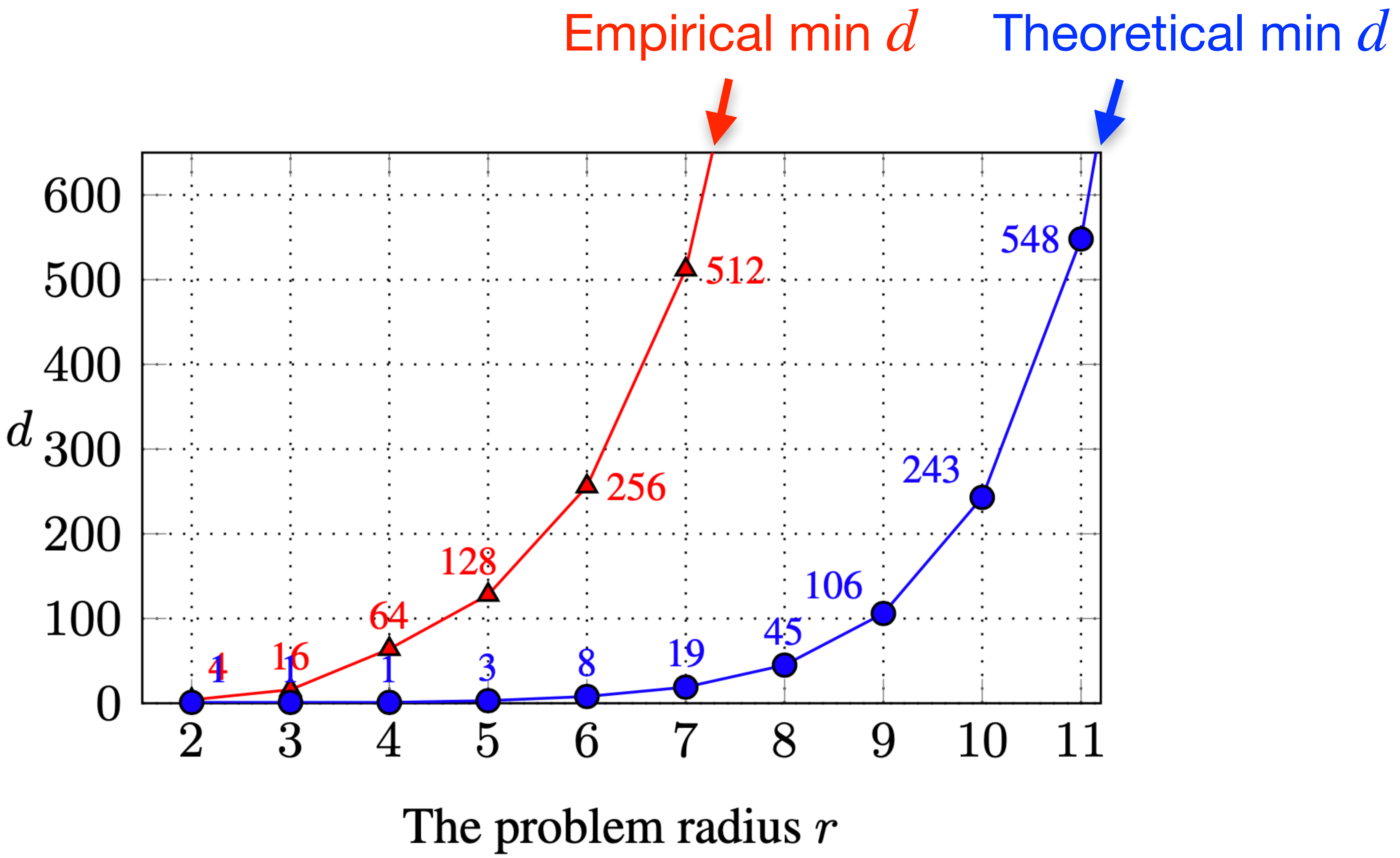
- Combinatorially, to fit the dataset: $\boxed{2^{32\cdot d} > \frac{(2^r)\,!}{(2)^{2^r-1}}}$

Theoretical min d $\quad 2^{32\cdot d} > \frac{(2^r)\,!}{(2)^{2^r-1}}$



How long is “long-range”?

- Combinatorially, to fit the dataset: $\boxed{2^{32\cdot d} > \frac{\left(2^{r}\right)!}{(2)^{2^{r}-1}}}$



$$2^{32\cdot d} > \frac{\left(2^{r}\right)!}{(2)^{2^{r}-1}}$$

GCN and GIN suffer from over-squashing **more** than GAT and GGNN

- GCN $\mathbf{h}_v^{(k)} = ReLU\left(W^{(k)} \sum_{u \in \mathcal{N}_v \cup \{v\}} \frac{1}{c_{v,u}} \mathbf{h}_u^{(k-1)}\right)$
- GIN $\mathbf{h}_v^{(k)} = MLP^{(k)}\left(\left(1 + \epsilon^{(k)}\right) \mathbf{h}_v^{(k-1)} + \sum_{u \in \mathcal{N}_v} \mathbf{h}_u^{(k-1)}\right)$
- GAT $\mathbf{h}_v^{(k)} = ReLU\left(MultiHeadAttention\left(\mathcal{N}_v \mid \mathbf{h}_v^{(k-1)}\right)\right)$
- GGNN $\mathbf{h}_v^{(k)} = GRU\left(\mathbf{h}_v^{(k-1)}, \sum_{u \in \mathcal{N}_v} W_{neighbor} \mathbf{h}_u^{(k-1)}\right)$

GCN and GIN suffer from over-squashing **more** than GAT and GGNN

- GCN $\mathbf{h}_v^{(k)} = ReLU\left(W^{(k)} \boxed{\sum_{u \in \mathcal{N}_v \cup \{v\}} \frac{1}{c_{v,u}} \mathbf{h}_u^{(k-1)}}\right)$
- GIN $\mathbf{h}_v^{(k)} = MLP^{(k)}\left(\left(1 + \epsilon^{(k)}\right) \mathbf{h}_v^{(k-1)} + \boxed{\sum_{u \in \mathcal{N}_v} \mathbf{h}_u^{(k-1)}}\right)$
- GAT $\mathbf{h}_v^{(k)} = ReLU\left(MultiHeadAttention\left(\mathcal{N}_v \mid \mathbf{h}_v^{(k-1)}\right)\right)$
- GGNN $\mathbf{h}_v^{(k)} = GRU\left(\mathbf{h}_v^{(k-1)}, \sum_{u \in \mathcal{N}_v} W_{neighbor} \mathbf{h}_u^{(k-1)}\right)$

GCN and GIN suffer from over-squashing **more** than GAT and GGNN

- GCN $\quad \mathbf{h}_v^{(k)} = ReLU\left(W^{(k)} \boxed{\sum_{u \in \mathcal{N}_v \cup \{v\}} \frac{1}{c_{v,u}} \mathbf{h}_u^{(k-1)}}\right)$
- GIN $\quad \mathbf{h}_v^{(k)} = MLP^{(k)}\left(\left(1 + \epsilon^{(k)}\right) \mathbf{h}_v^{(k-1)} + \boxed{\sum_{u \in \mathcal{N}_v} \mathbf{h}_u^{(k-1)}}\right)$
- GAT $\quad \mathbf{h}_v^{(k)} = ReLU\left(\boxed{MultiHeadAttention\left(\mathcal{N}_v \mid \mathbf{h}_v^{(k-1)}\right)}\right)$
- GGNN $\quad \mathbf{h}_v^{(k)} = \boxed{GRU}\left(\mathbf{h}_v^{(k-1)}, \sum_{u \in \mathcal{N}_v} W_{neighbor} \mathbf{h}_u^{(k-1)}\right)$

Public datasets

- To break the bottleneck:
  - We modified the last GNN layer to be fully-adjacent (FA) - every node has an edge to every other node

Public datasets

- To break the bottleneck:
  - We modified the last GNN layer to be fully-adjacent (FA) - every node has an edge to every other node
  - Re-trained without adding weights, without any hyperparameter tuning

Public datasets

- To break the bottleneck:
  - We modified the last GNN layer to be fully-adjacent (FA) - every node has an edge to every other node
  - Re-trained without adding weights, without any hyperparameter tuning
    - A temporary solution, just to show that the bottleneck is so prevalent and untreated — that *even the simplest solution helps*.

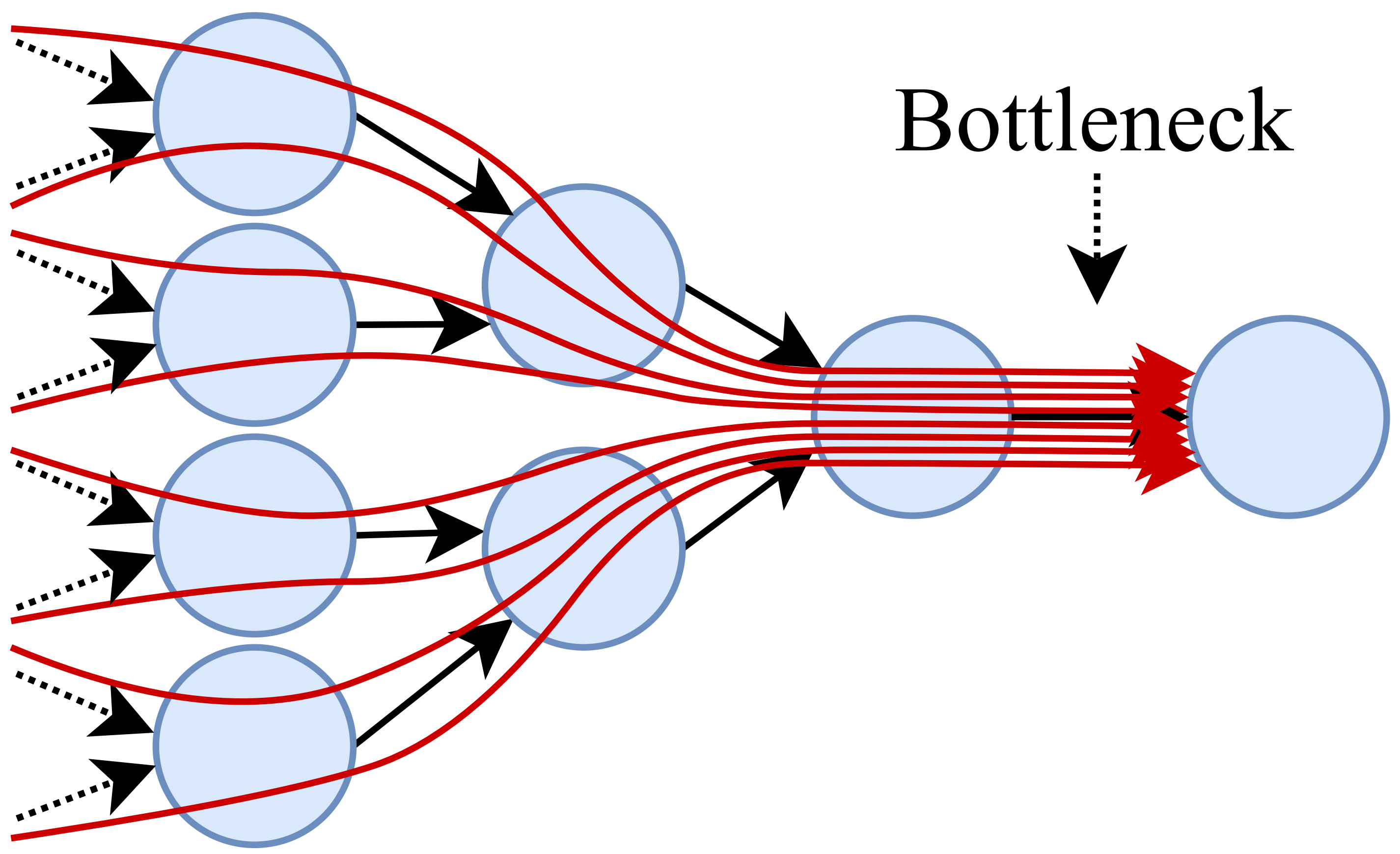
Public datasets

- To break the bottleneck:
  - We modified the last GNN layer to be fully-adjacent (FA) - every node has an edge to every other node
  - Re-trained without adding weights, without any hyperparameter tuning
    - A temporary solution, just to show that the bottleneck is so prevalent and untreated – that *even the simplest solution helps*.

- +1% accuracy increase in Variable Misuse
- -40% error reduction in predicting quantum chemical properties of molecules (“QM9”)
- -5% error reduction in classifying biochemical compounds (“NCI1”)
- -12% error reduction in classifying enzymes (“ENZYMES”)

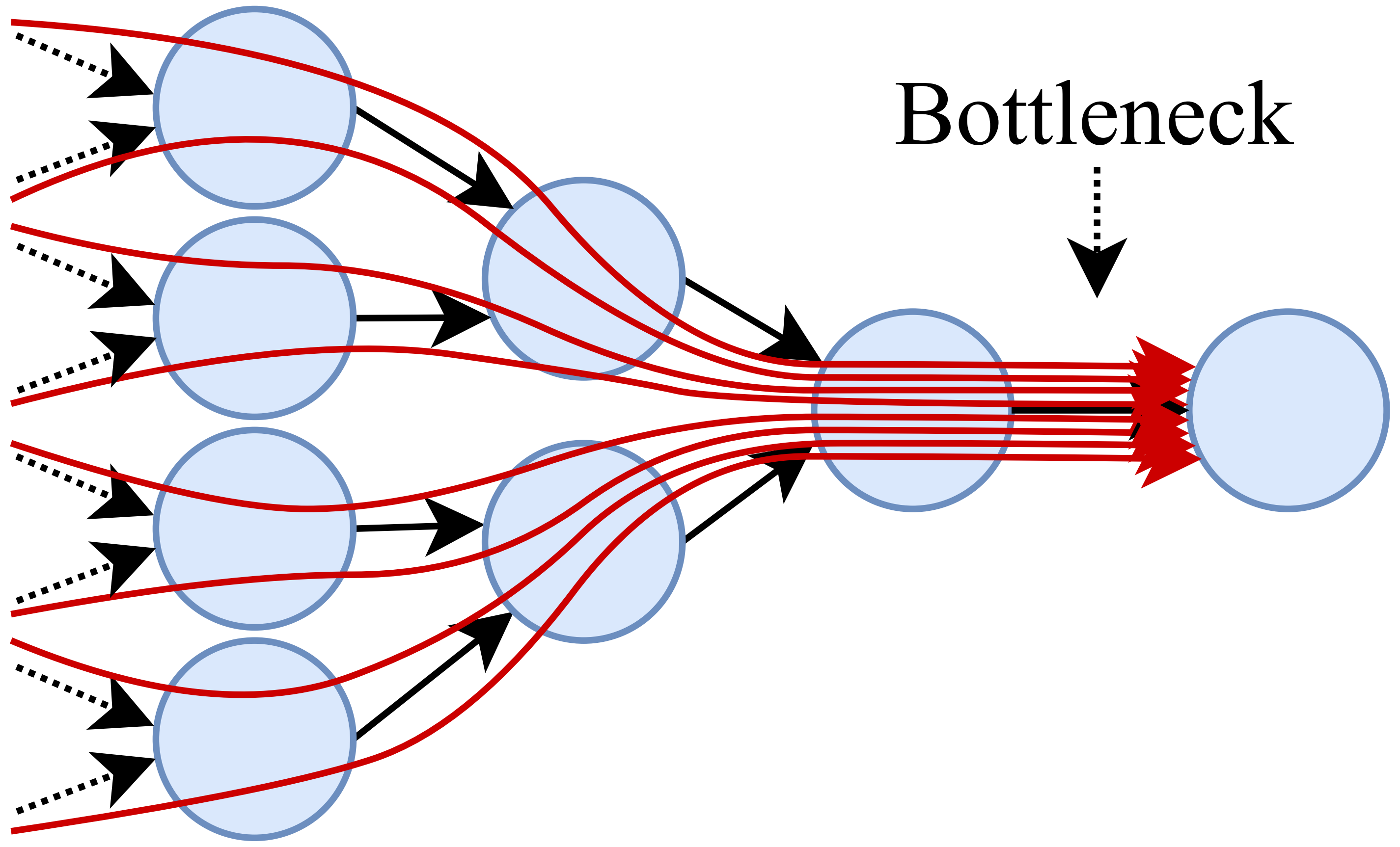
Summary

- To pass long-range messages - we need many GNN layers



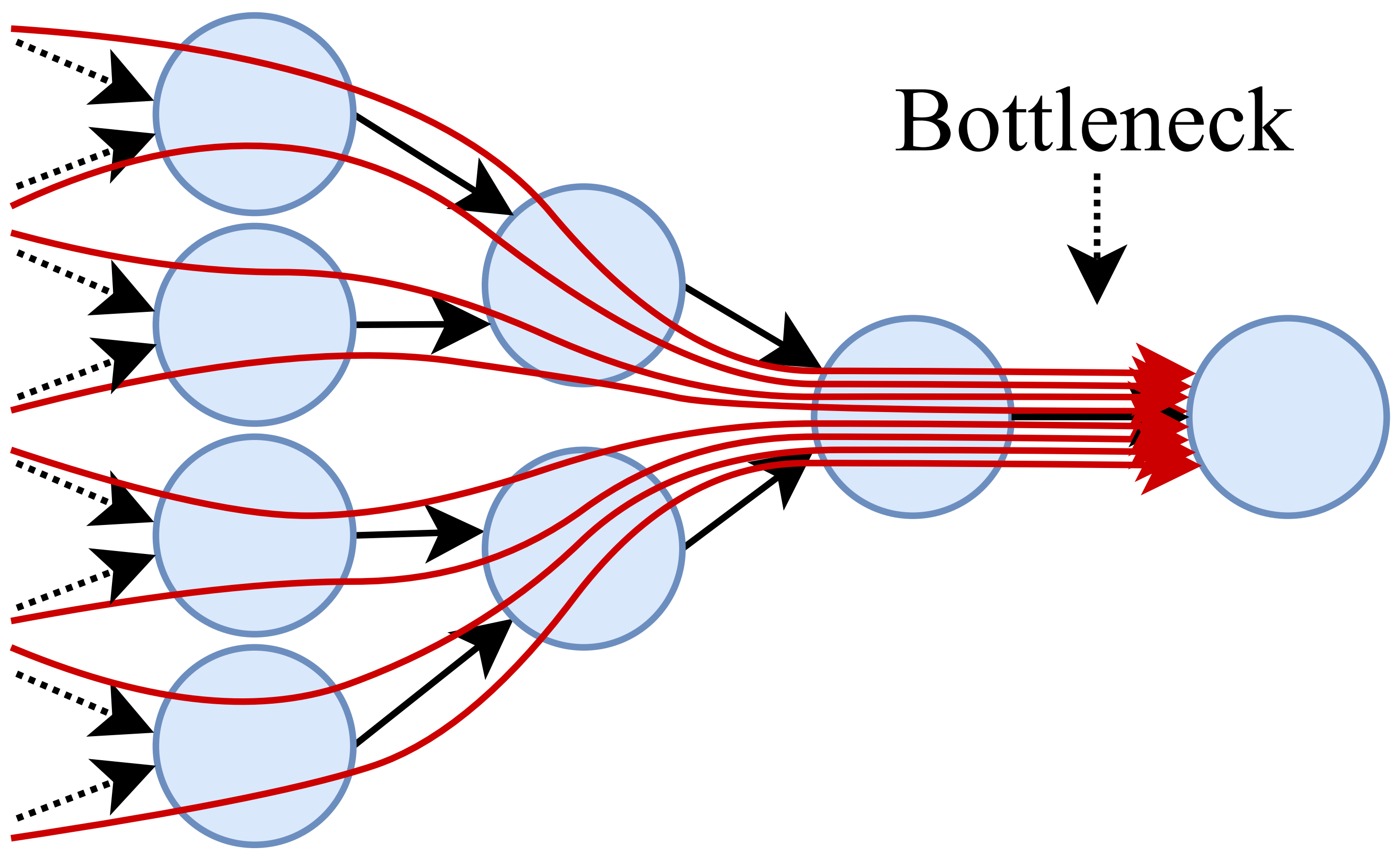
Summary

- To pass long-range messages - we need many GNN layers
- A node's receptive field grows **exponentially** with the number of layers



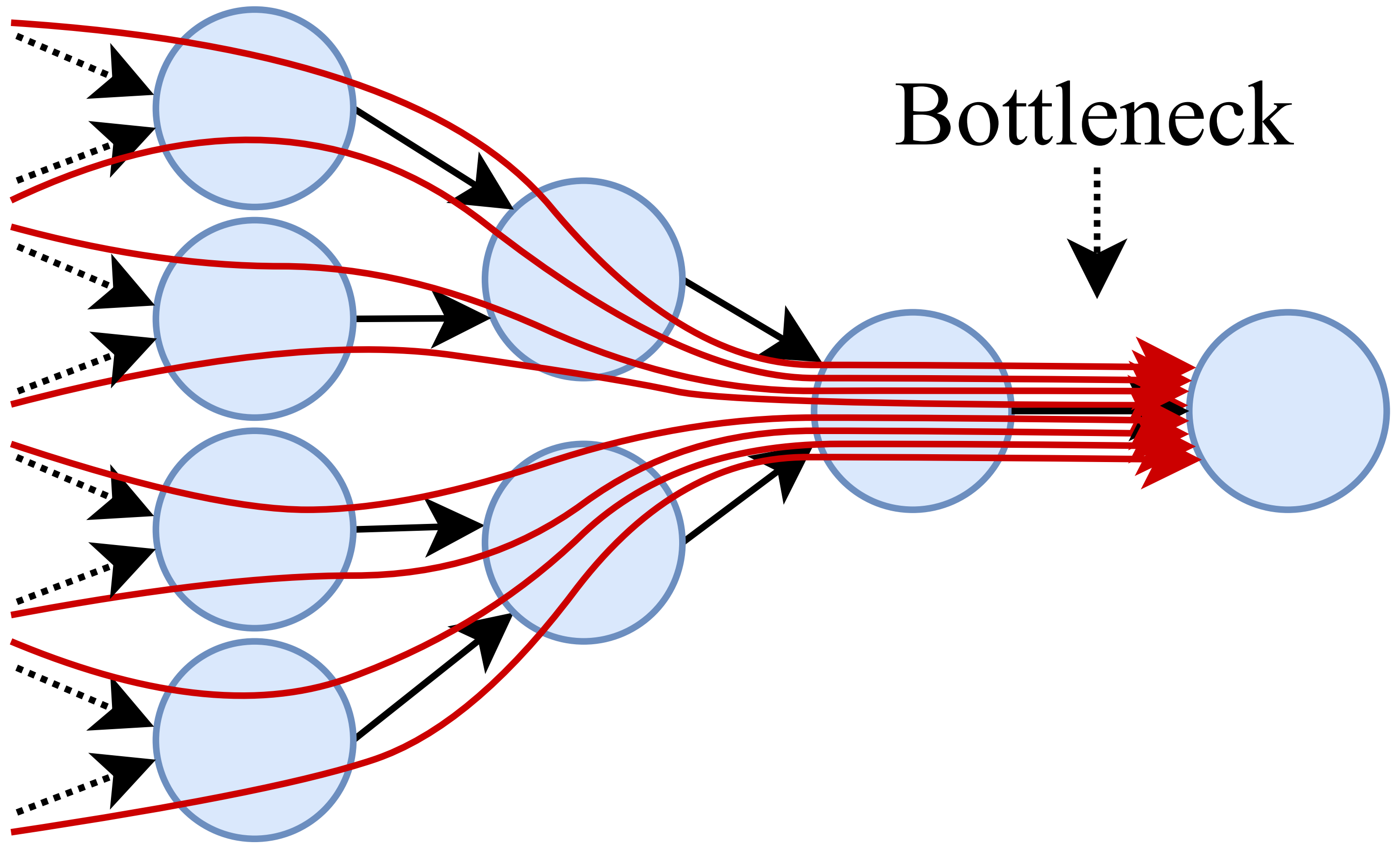
Summary

- To pass long-range messages - we need many GNN layers
- A node’s receptive field grows **exponentially** with the number of layers
  - ➡ Leads to a **bottleneck** and **over-squashing**



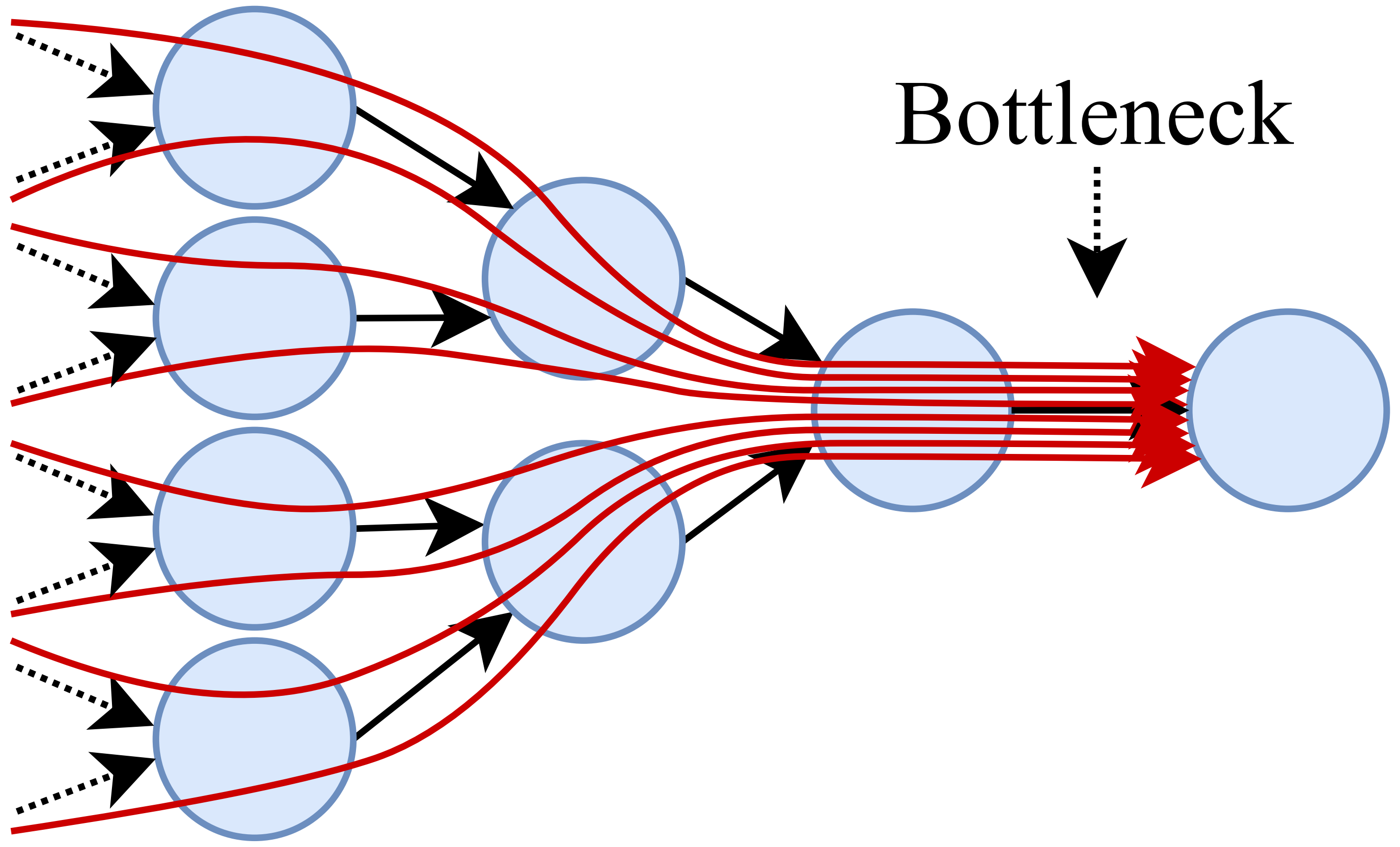
Summary

- To pass long-range messages - we need many GNN layers
- A node's receptive field grows **exponentially** with the number of layers
  - ➡ Leads to a **bottleneck** and **over-squashing**
- **GCN** and **GIN** suffer from over-squashing **more** than others



Summary

- To pass long-range messages - we need many GNN layers
- A node's receptive field grows **exponentially** with the number of layers
  - ➡ Leads to a **bottleneck** and **over-squashing**
- **GCN** and **GIN** suffer from over-squashing **more** than others
- SoTA models can be **improved** by simply considering the bottleneck



Summary

- To pass long-range messages - we need many GNN layers
- A node's receptive field grows **exponentially** with the number of layers
    - ➡ Leads to a **bottleneck** and **over-squashing**
- **GCN** and **GIN** suffer from over-squashing **more** than others
- SoTA models can be **improved** by simply considering the bottleneck

http://urialon.ml
urialon@cs.technion.ac.il

ICLR: **May 5th, 9AM PDT**
(Poster session 8)