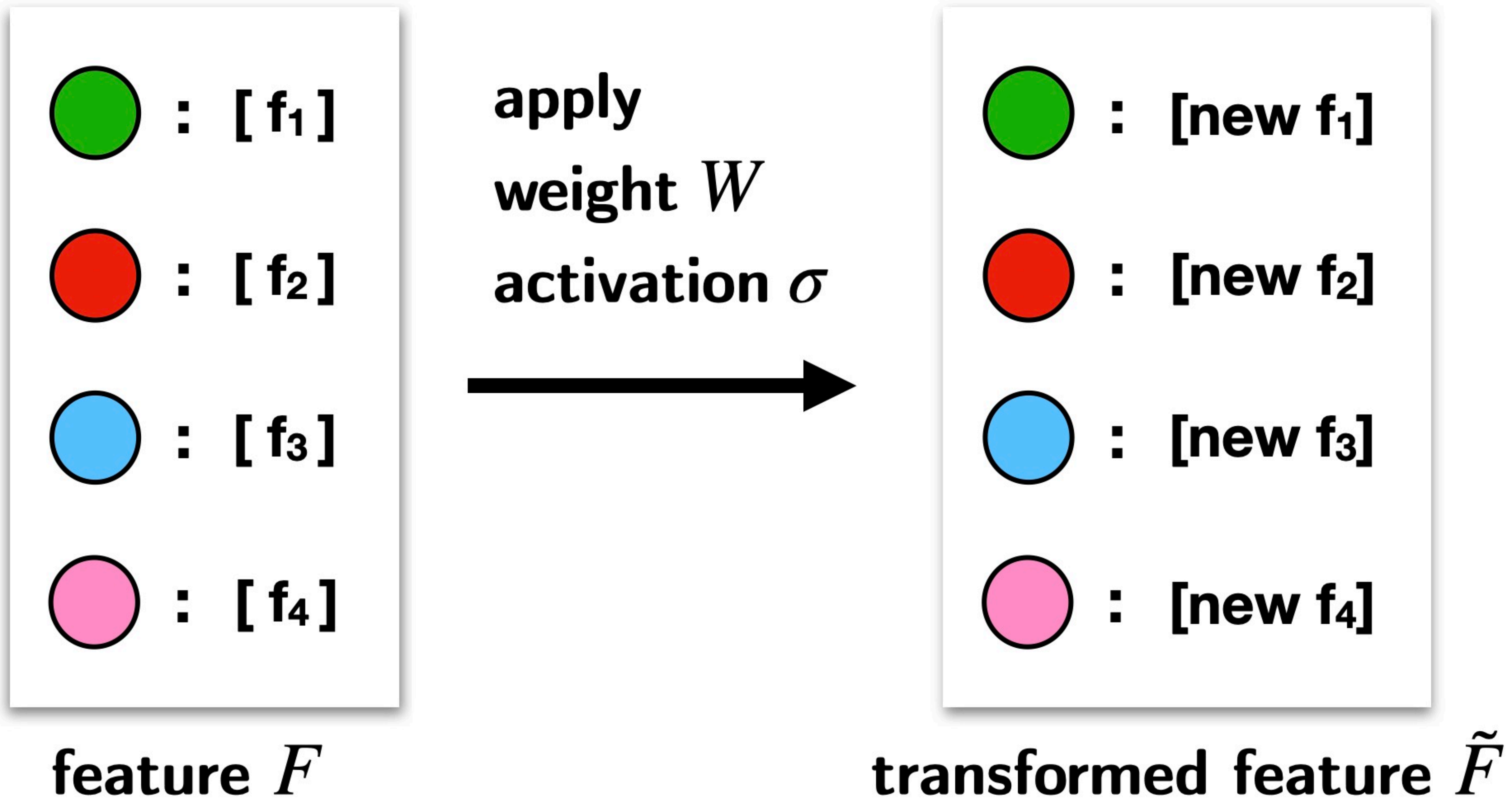
Improving Graph Neural Networks by Learning Continuous Edge Directions

Seong Ho Pahng[1,2] **Sahand Hormoz**[3,2,4]

[1]Department of Chemistry and Chemical Biology, Harvard University
[2]Department of Data Science, Dana-Farber Cancer Institute
[3]Department of Systems Biology, Harvard Medical School
[4]Broad Institute of MIT and Harvard

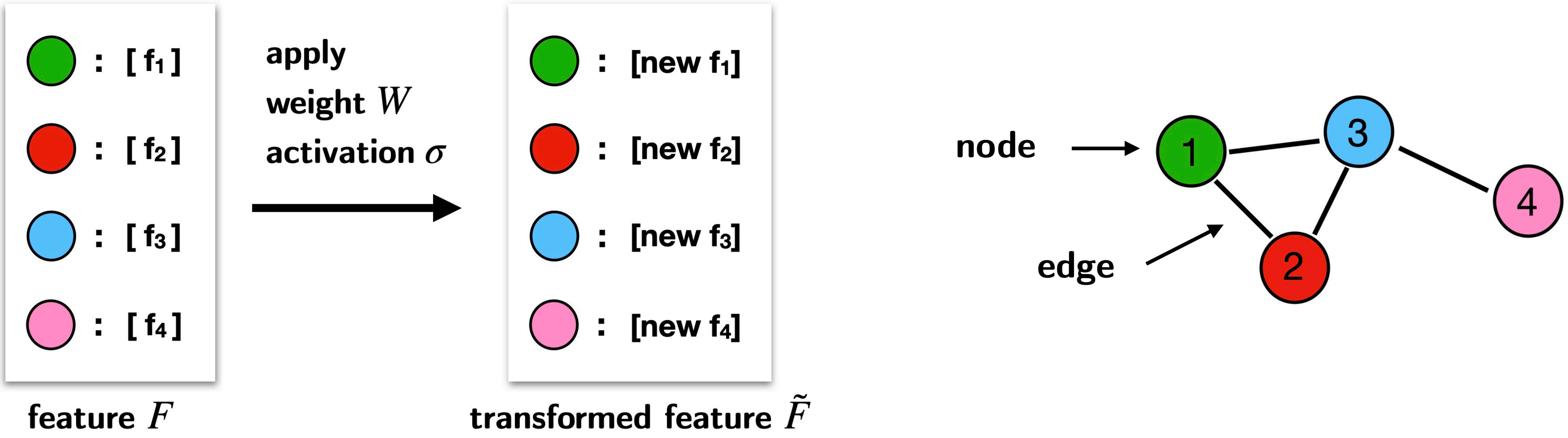
Neural network for graph-structured data

Conventional neural networks



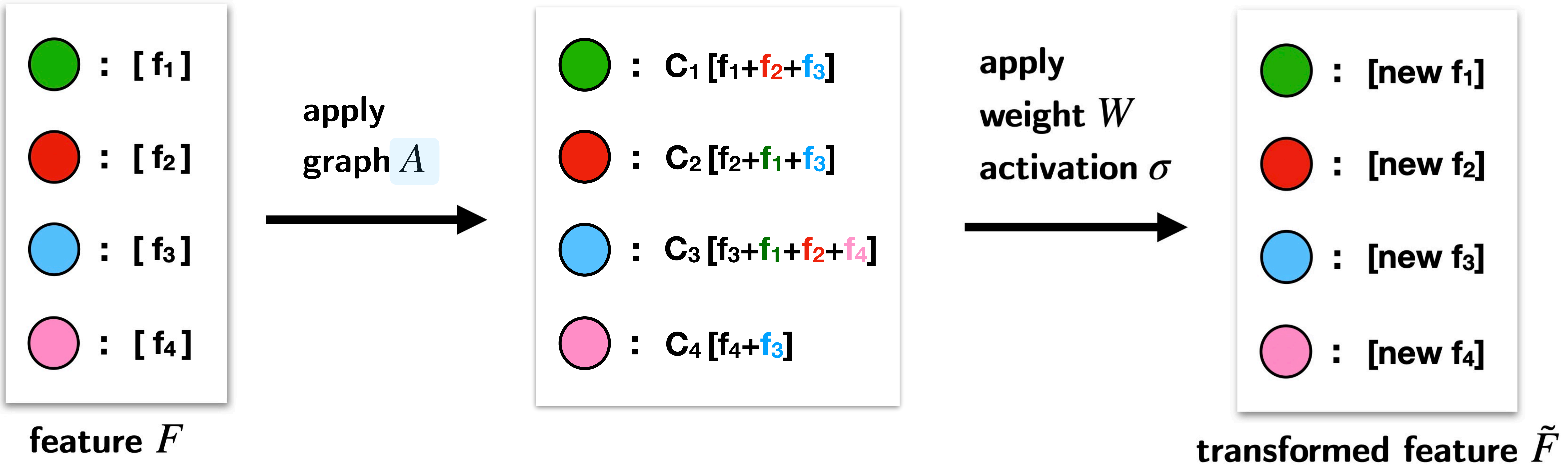
Neural network for graph-structured data

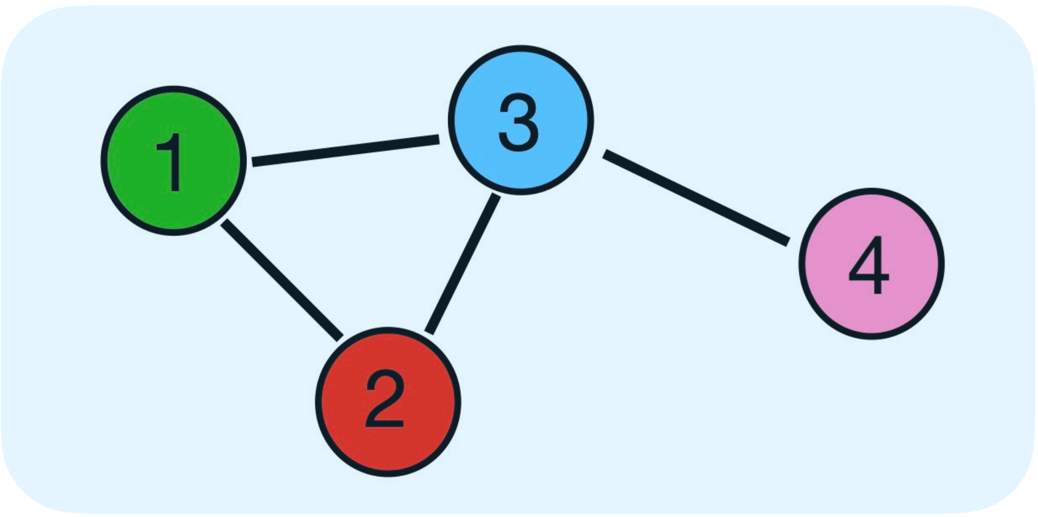
Conventional neural networks



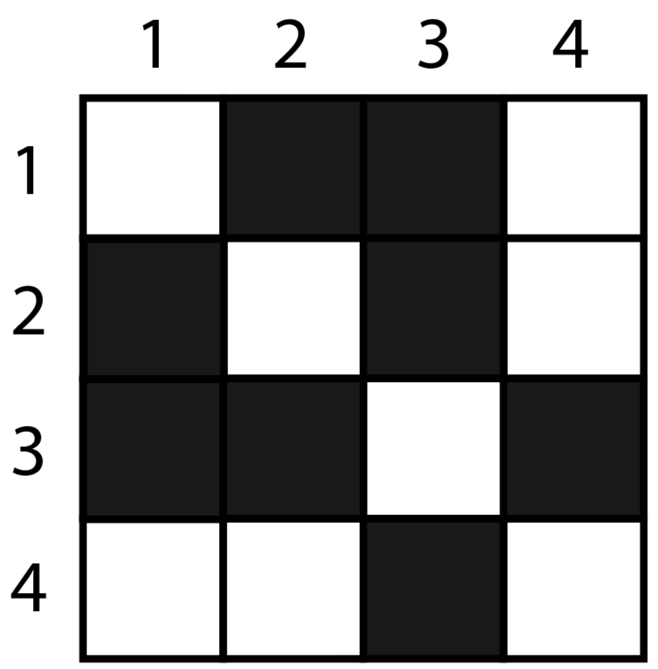
Neural network for graph-structured data

Graph neural networks (GNN)





Adjacency matrix $A =$



$\in \{0,1\}^{N \times N}$

□ : 0

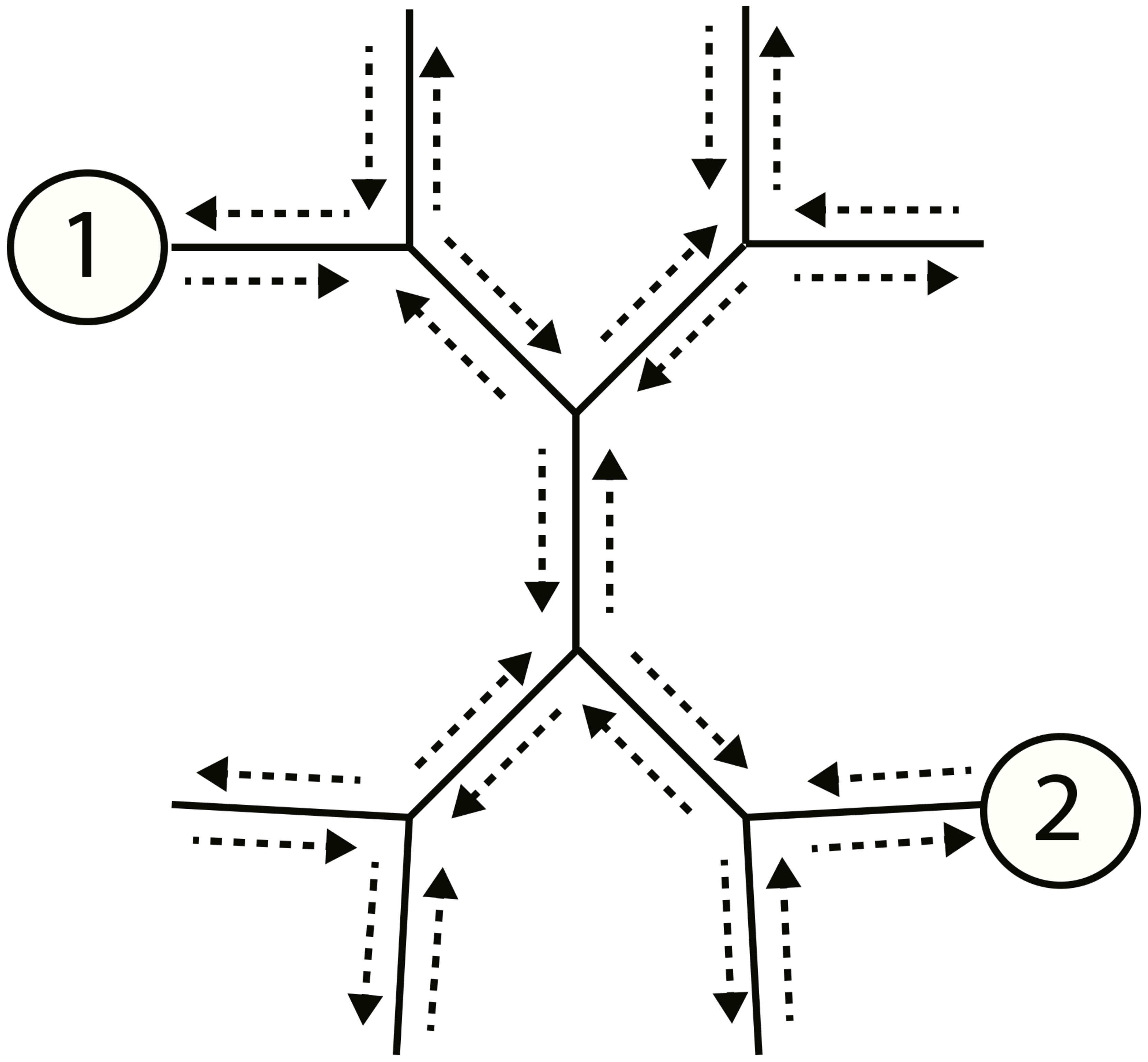
■ : 1

Problem with GNN using undirected graph

- **Multiplication of feature F with adjacency A leads to averaging over neighbors' features**
- **As we increase depths, each f_i of F converges toward a similar value**
- **This problem is known as "oversmoothing"**

Oversmoothing leads to inefficient information propagation

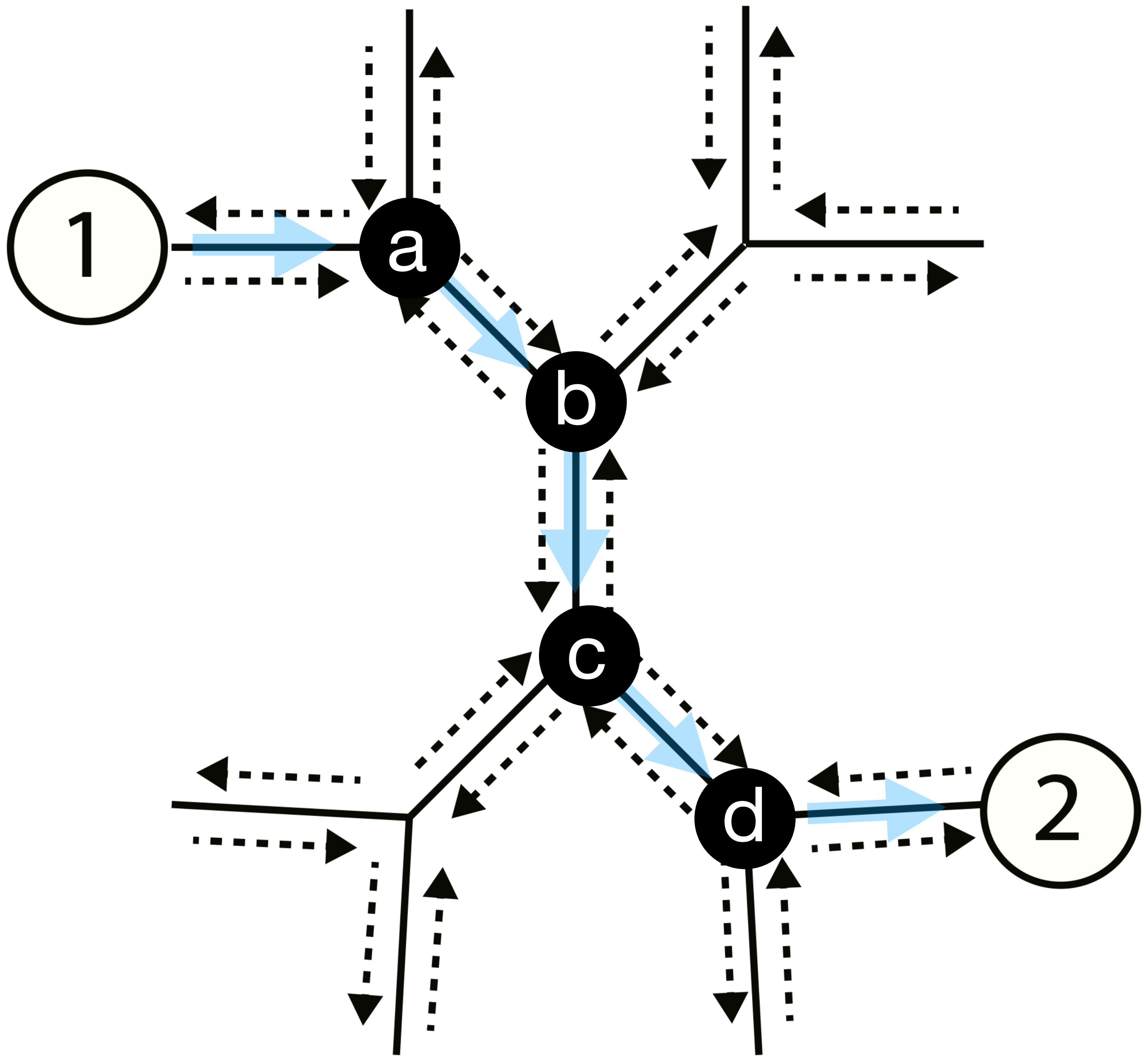
Picture for intuition



- **Task: send information from (1) to (2)**

Oversmoothing leads to inefficient information propagation

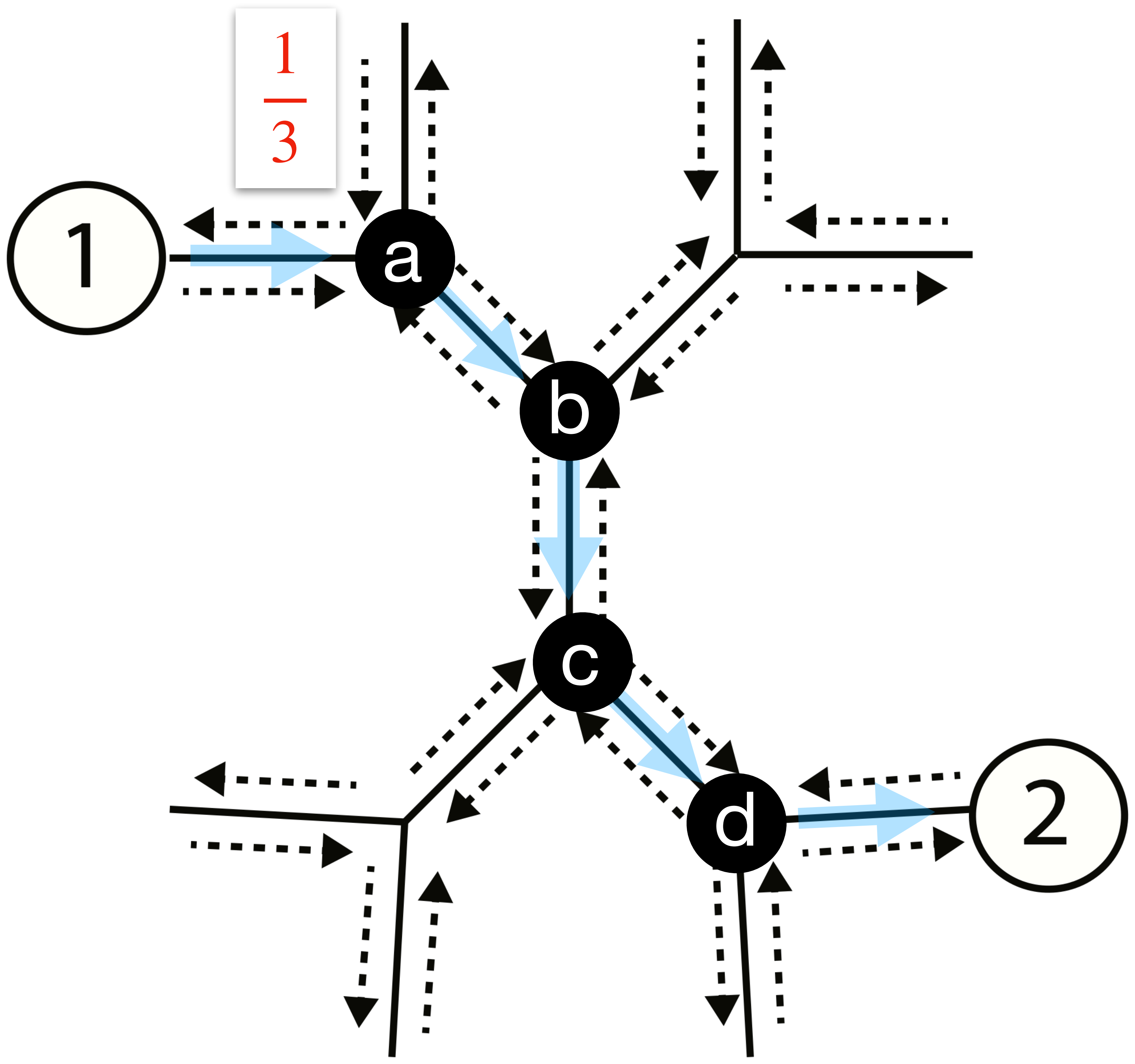
Picture for intuition



- **Task: send information from** (1) **to** (2)
- (1)**'s info. has to go through:** (a), (b), (c), (d)

Oversmoothing leads to inefficient information propagation

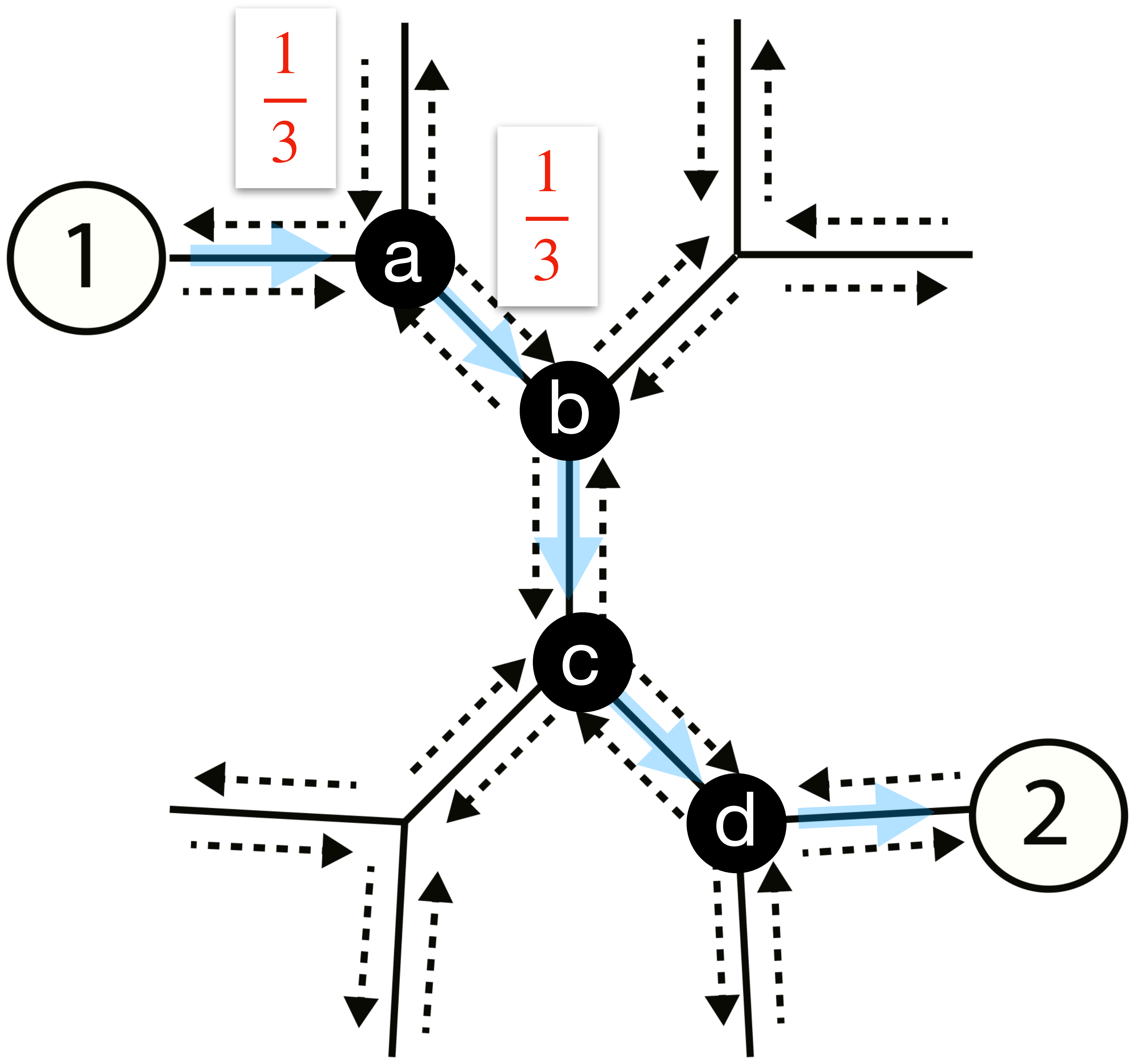
Picture for intuition



- Task: send information from (1) to (2)
- (1)'s info. has to go through: (a), (b), (c), (d)
- 1/3 of (a)'s received info. come from (1)

Overmoothing leads to inefficient information propagation

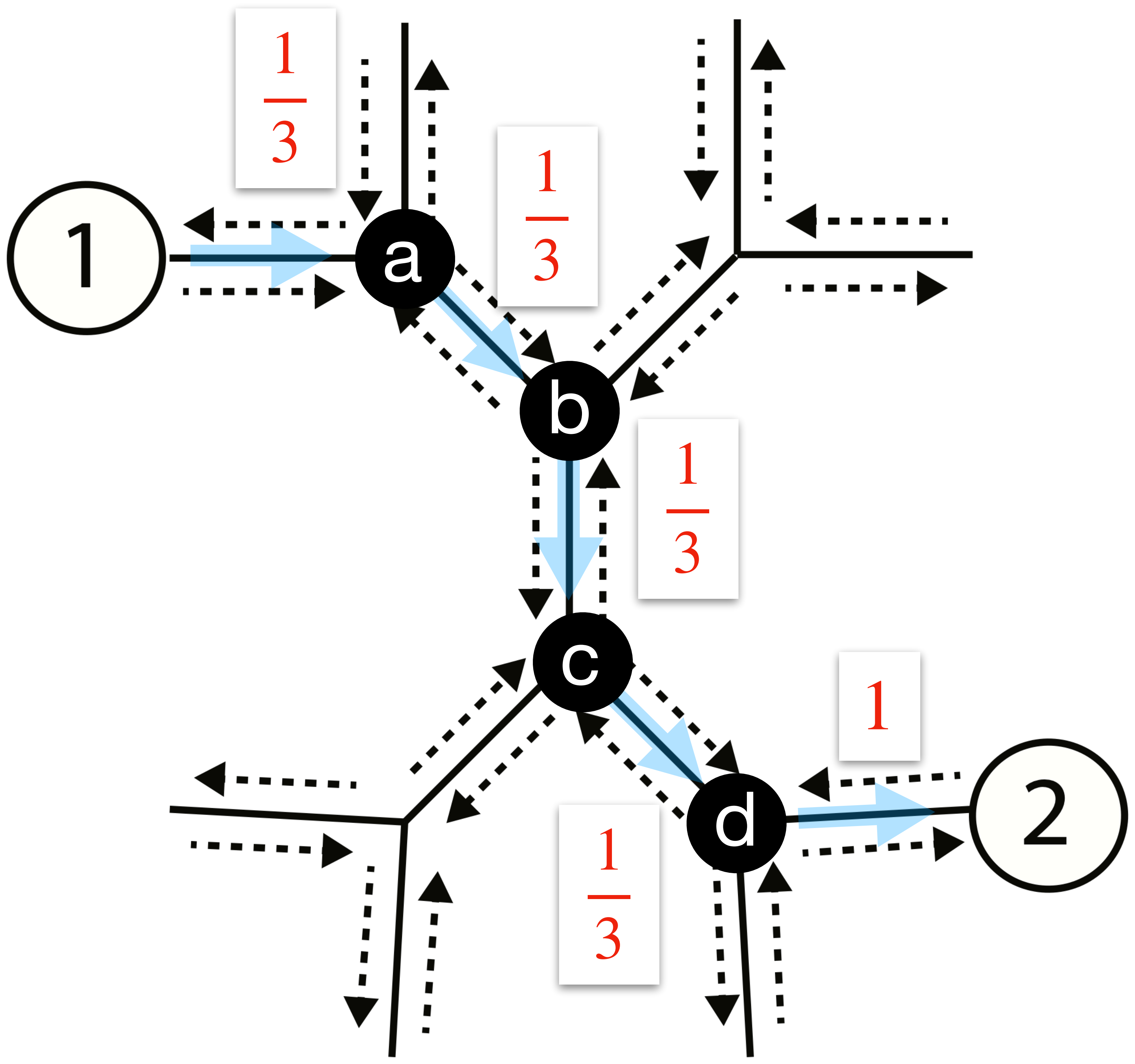
Picture for intuition



- Task: send information from 1 to 2
- 1's info. has to go through: a, b, c, d
- 1/3 of a's received info. come from 1
  - and 1/3 of b's from a

Oversmoothing leads to inefficient information propagation

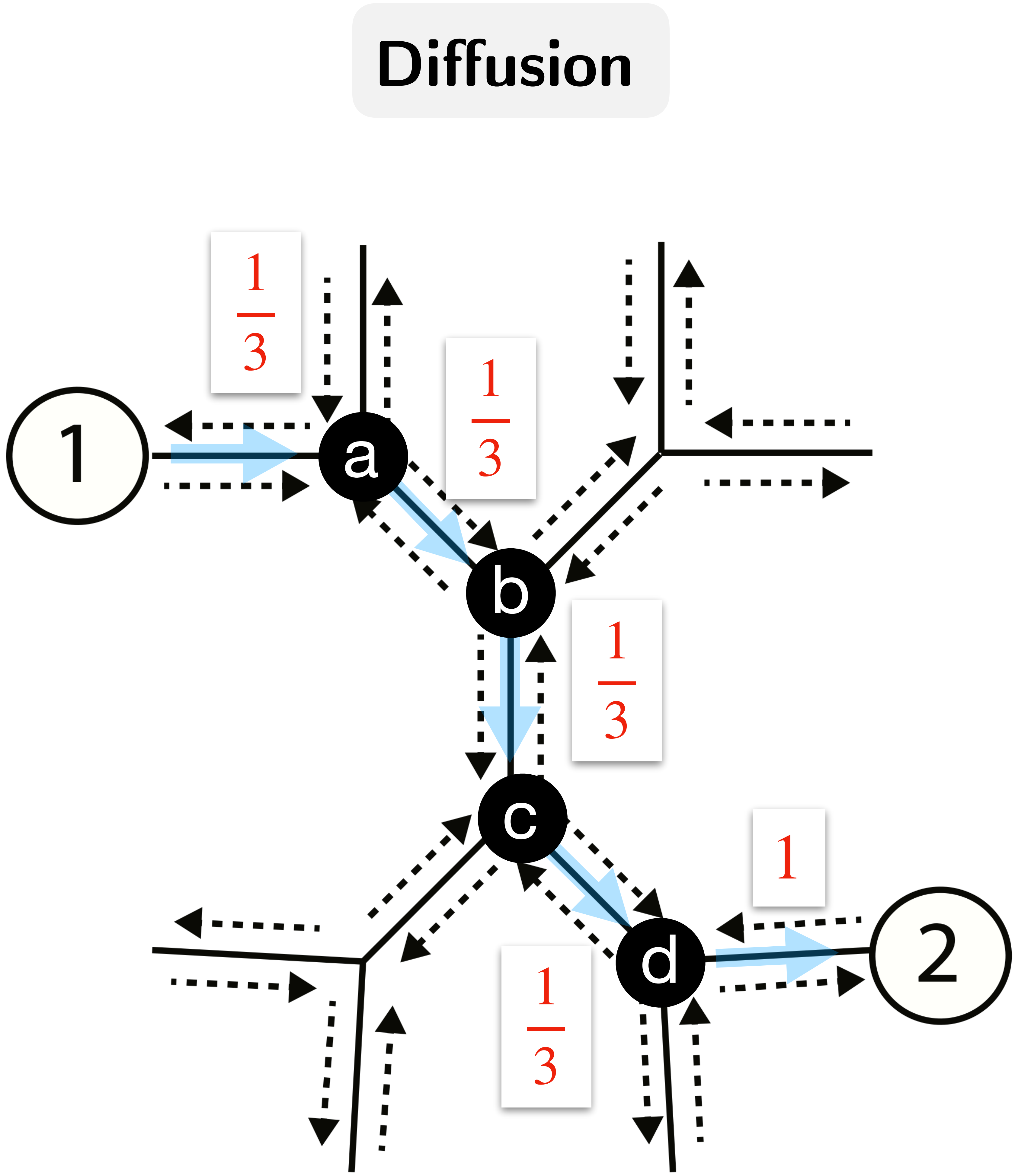
Picture for intuition

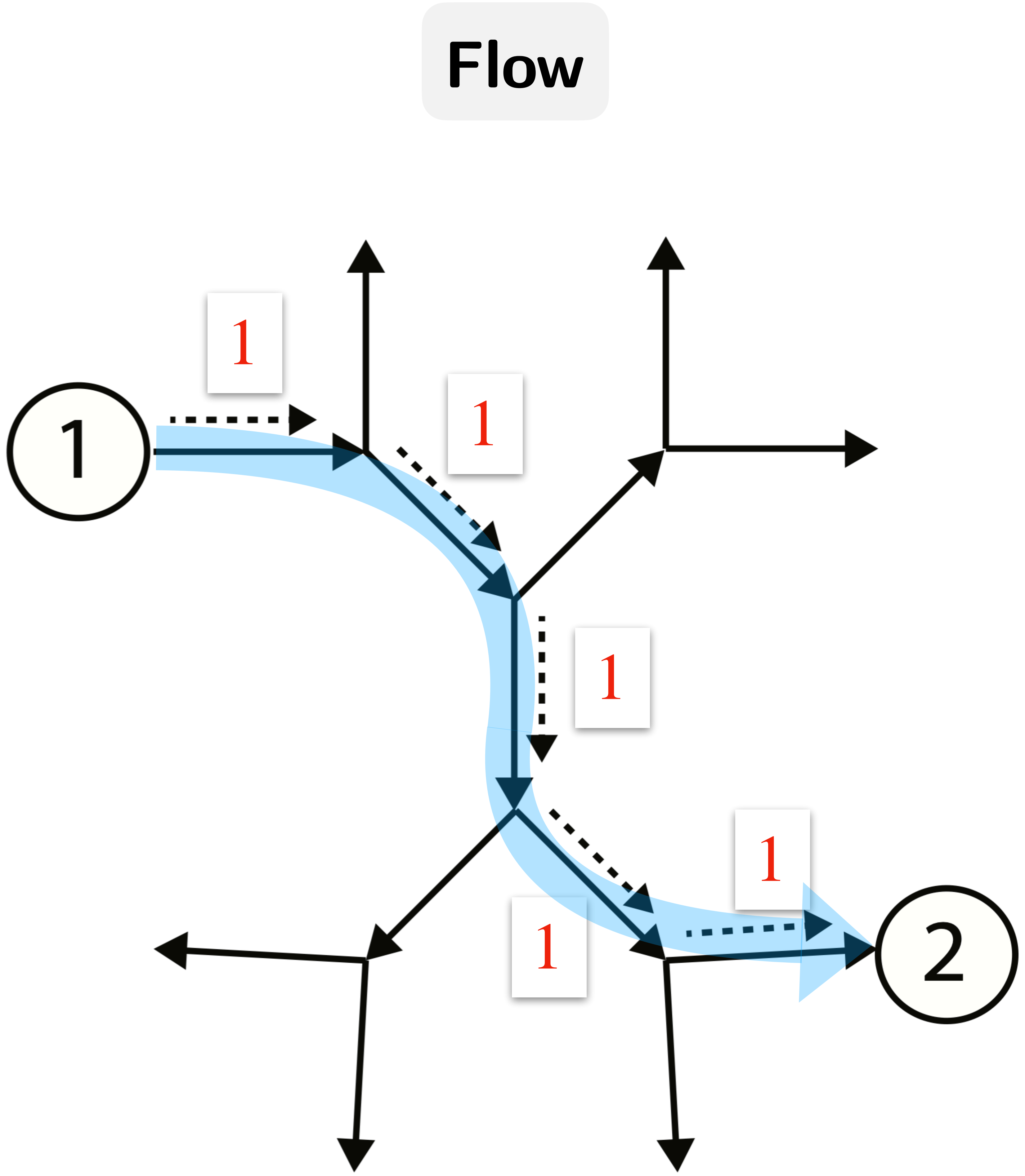


- Task: send information from 1 to 2
- 1's info. has to go through: a, b, c, d
- 1/3 of a's received info. come from 1
  - and 1/3 of b's from a

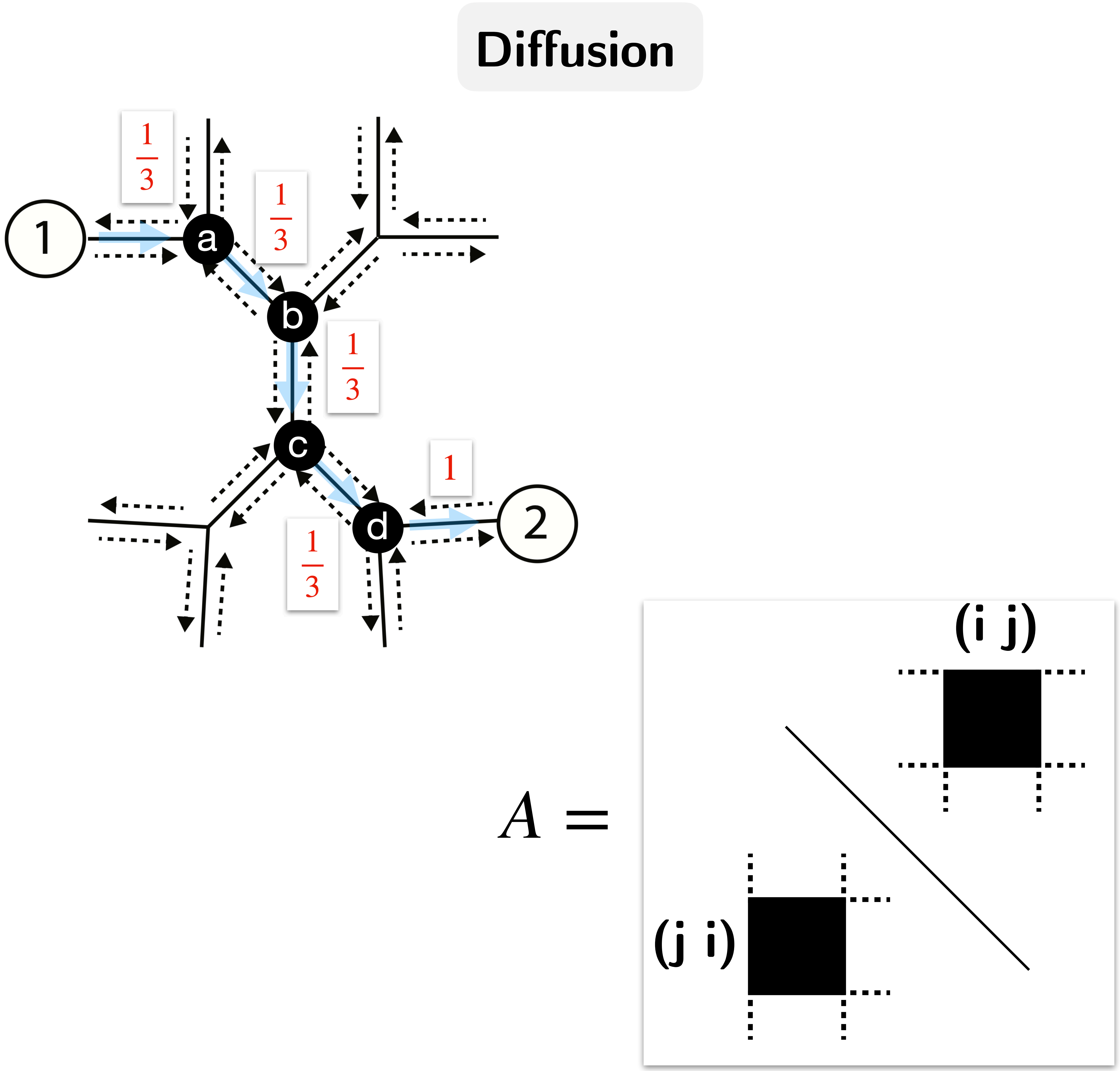
  ⋮
- only $(1/3)^4$'th of 2's info. originated from 1

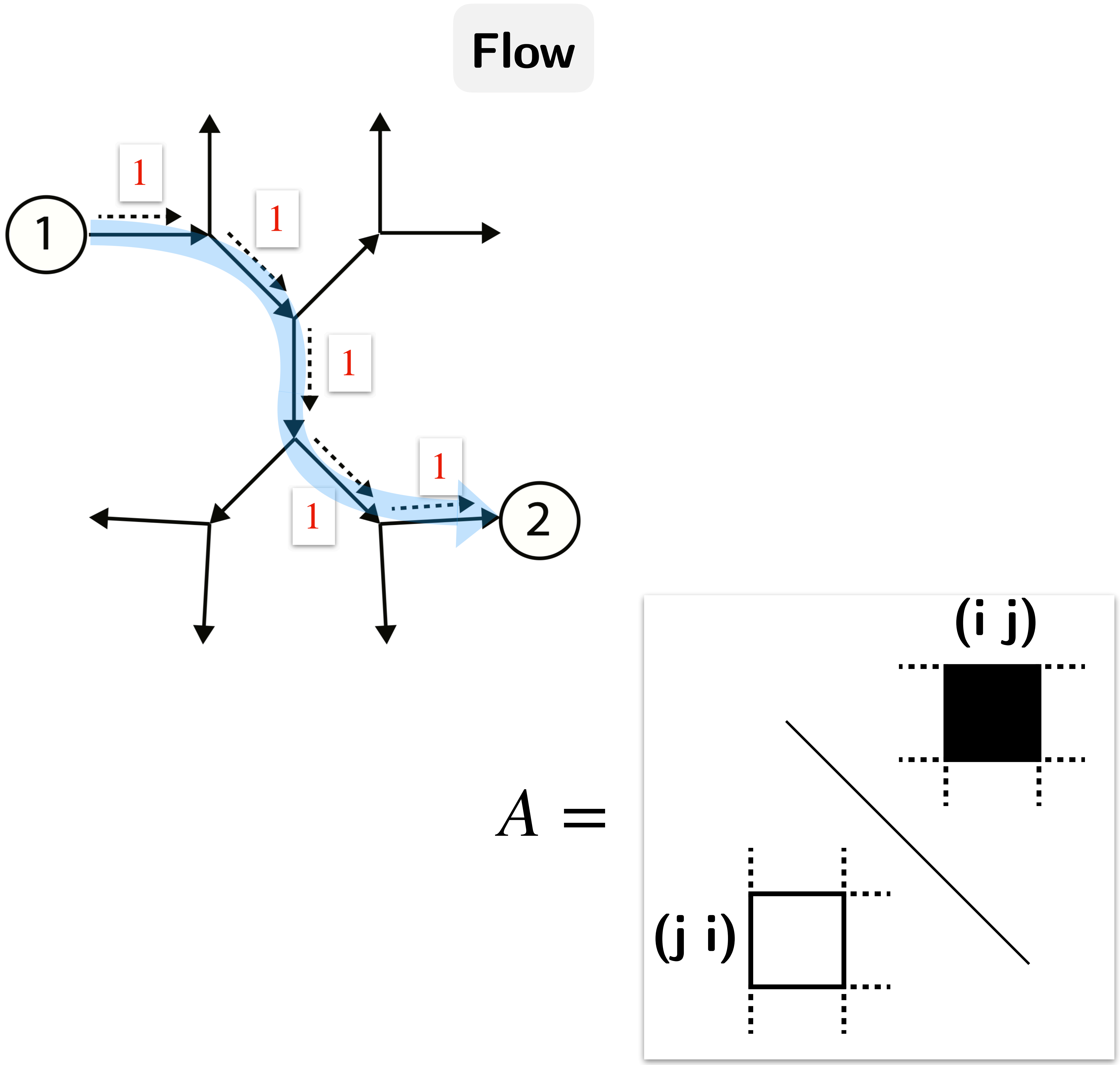
Remedy: Let information flow instead diffuse





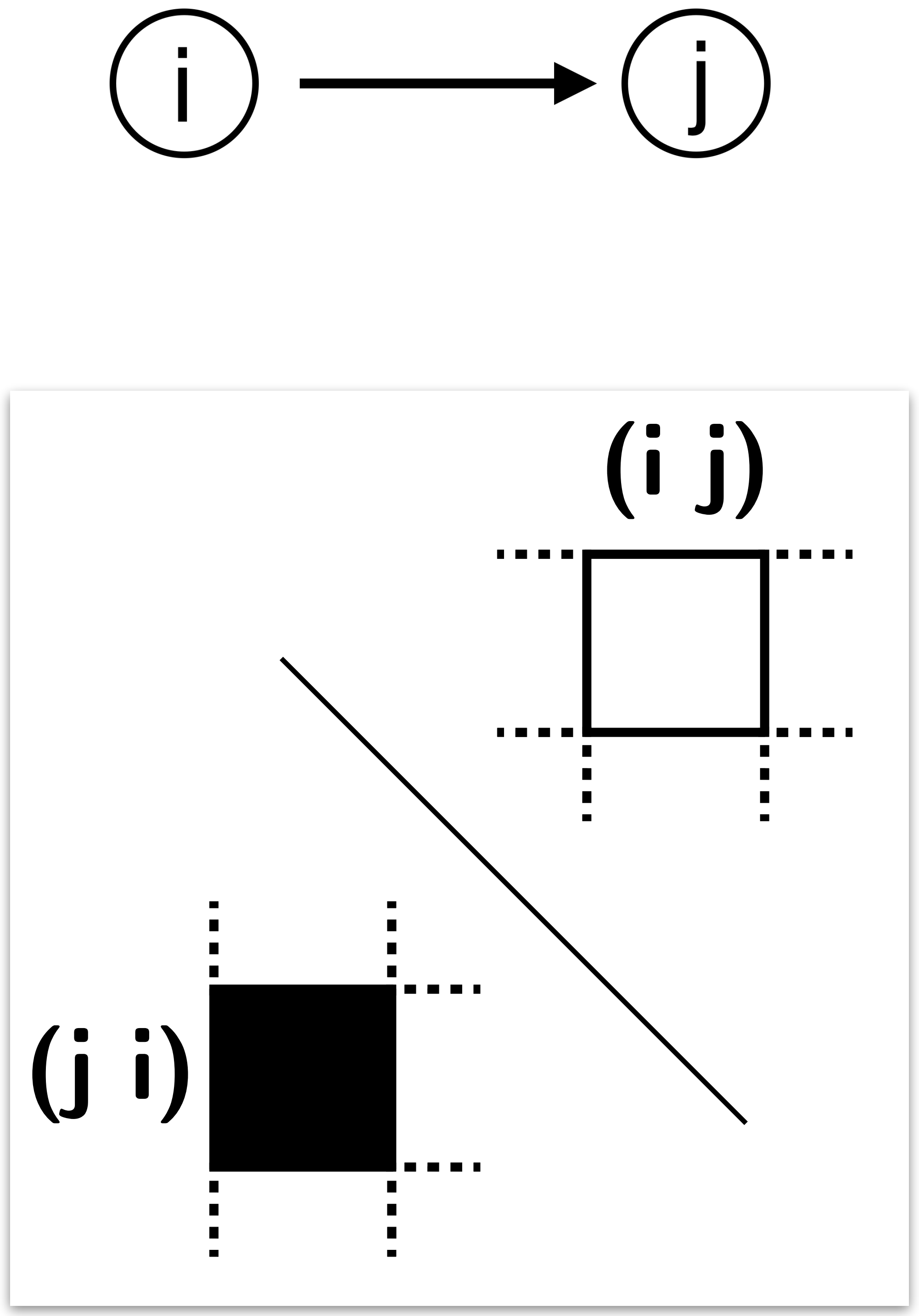
Remedy: Let information flow instead diffuse



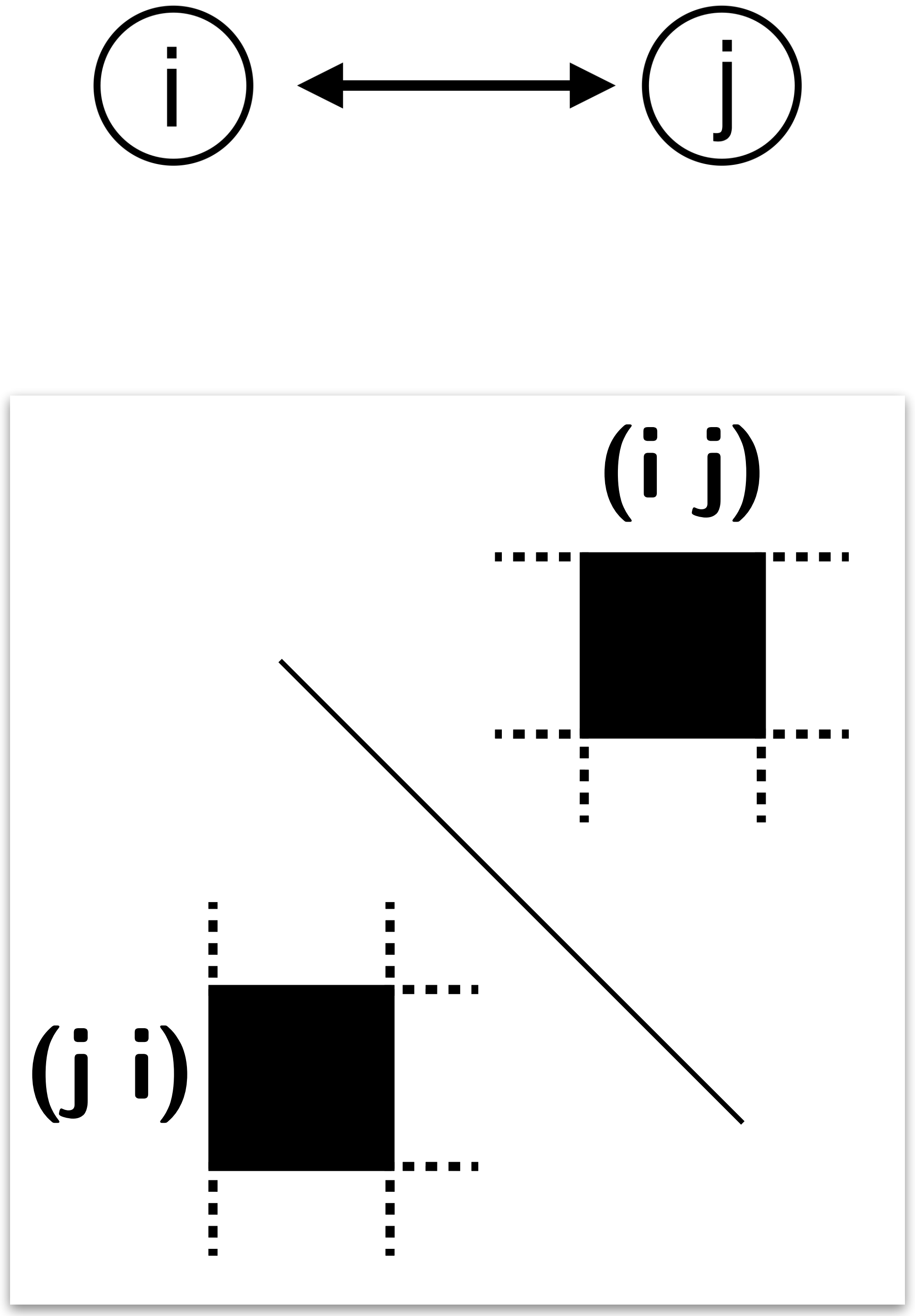


GNNs exclusively work with either undirected or directed graph

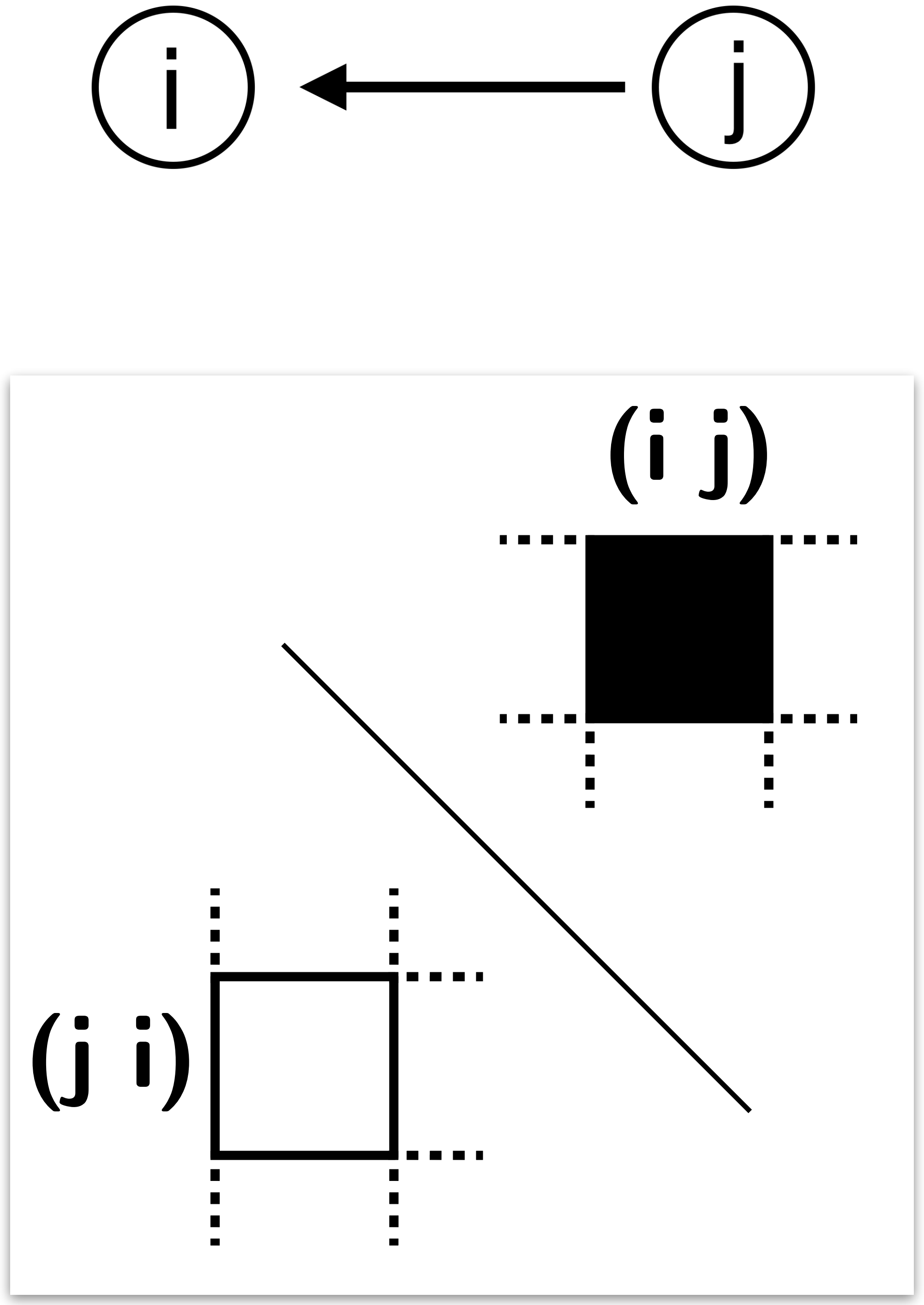
Directed (one way)



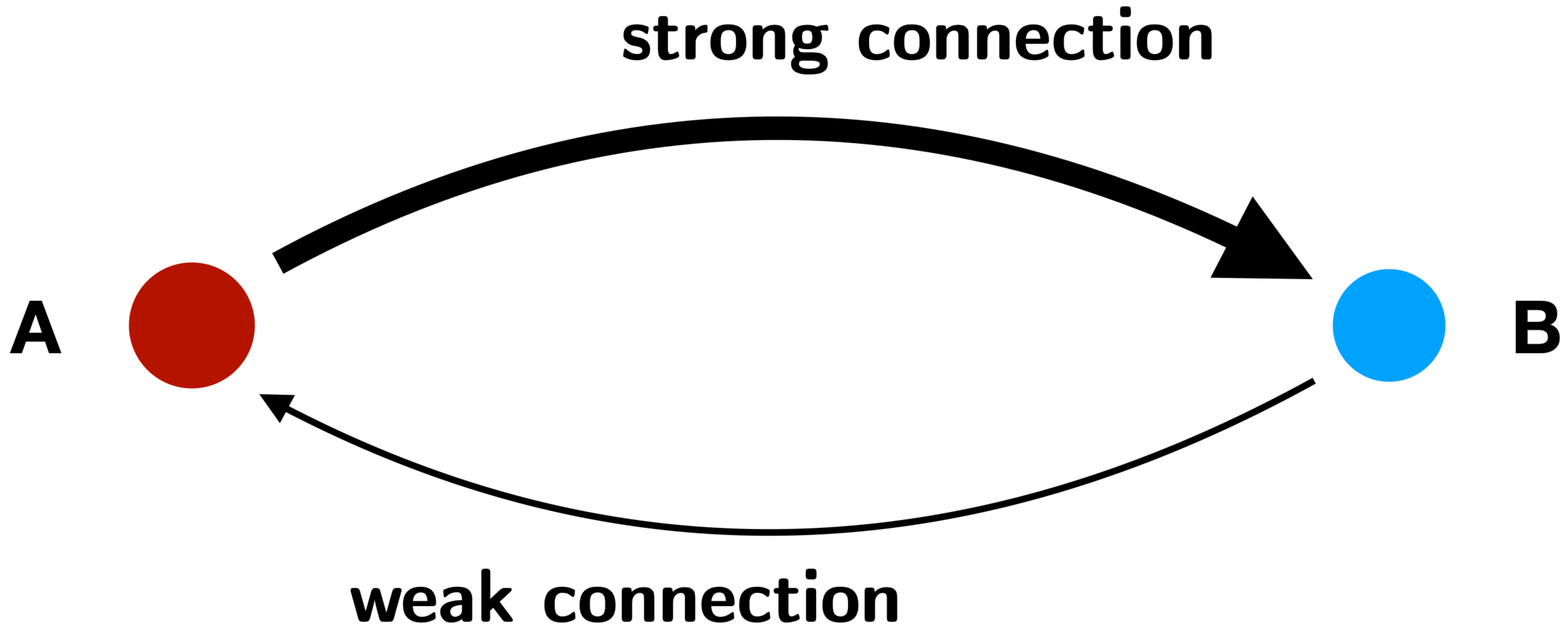
Undirected



Directed (the other way)



Information flow in data need not be discrete



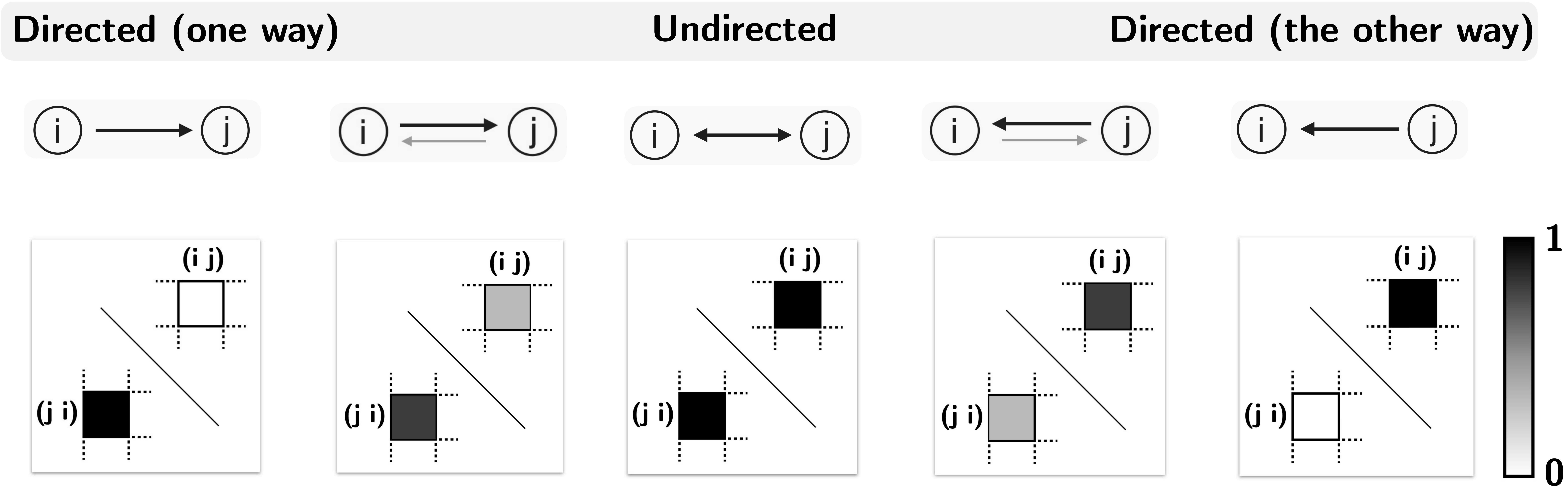
ex 1) Neuron A makes a strong synaptic connection to neuron B

Neuron B doesn't make as strong a synaptic connection to neuron A

ex 2) Country A export a lot of goods to country B

Country B doesn't export as much to country A

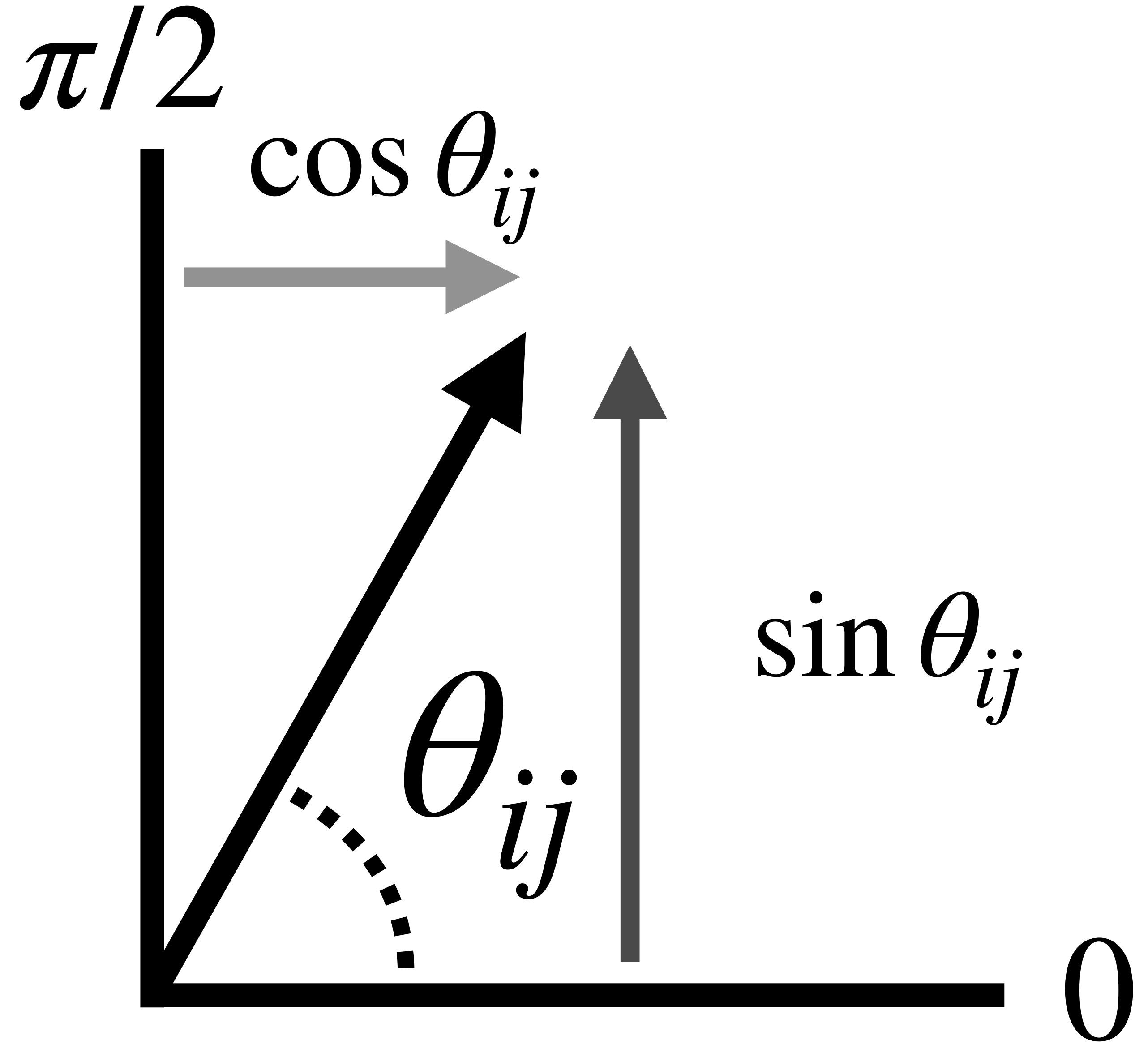
Information flow in data need not be discrete



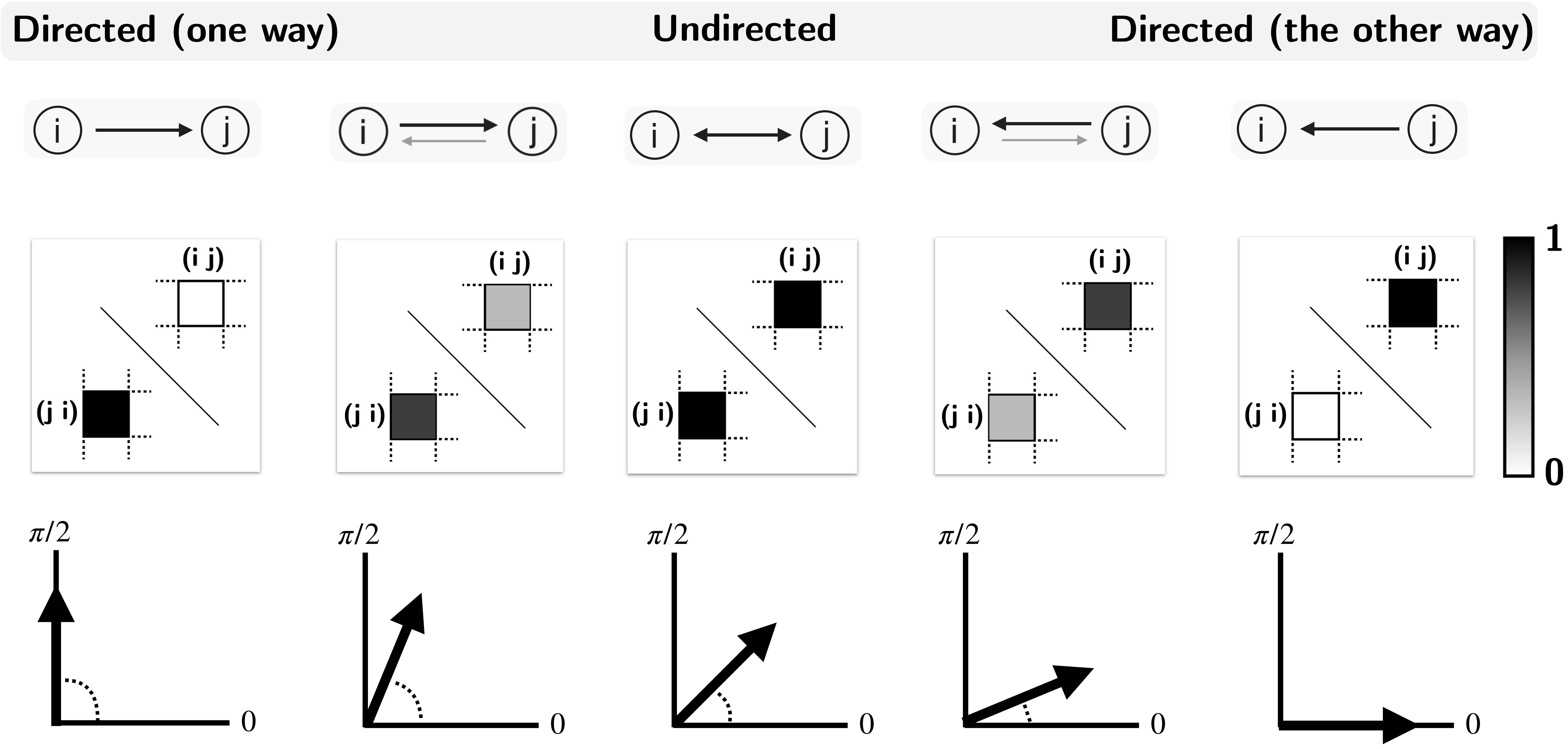
"Angle" θ_{ij} to capture continuously varying edge direction

$\cos(\theta_{ij})$: $i \leftarrow j$ **edge magnitude**

$\sin(\theta_{ij})$: $i \rightarrow j$ **edge magnitude**



"Angle" θ_{ij} to capture continuously varying edge direction



Constructing "fuzzy" graph Laplacian from angles

- **We encode $i \leftarrow j$ and $i \rightarrow j$ edges to real and imaginary parts of a complex number**

$$(\mathbf{L}_F)_{ij} = \begin{cases} 0 & \text{if } A_{ij} = A_{ji} = 0 \\ \exp(\mathrm{i}\theta_{ij}) & \text{otherwise} \end{cases}$$

- **Since $\theta_{ji} = \pi/2 - \theta_{ij}$, it follows that $\mathbf{L}_F = \mathrm{i}\mathbf{L}_F^\dagger$ $\longrightarrow$ $\mathbf{L}_F$ admits orthogonal eigenvectors**

Continuous Edge Direction (CoED) GNN

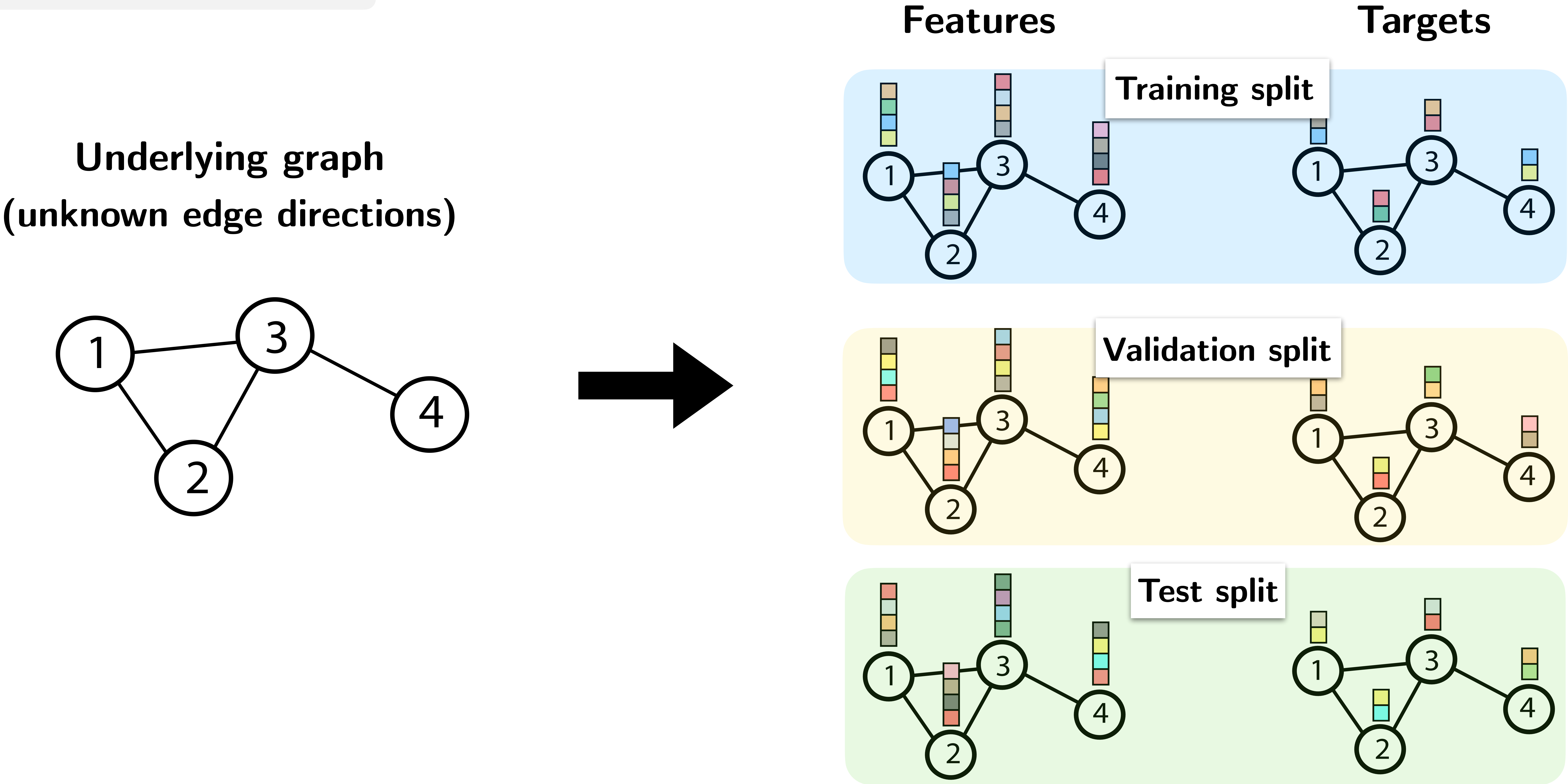
	over in-neighbors	over out-neighbors
Fuzzy adjacency	$\mathbf{A}_{\leftarrow} = \mathrm{Re}[\mathbf{L}_F]$	$\mathbf{A}_{\rightarrow} = \mathrm{Im}[\mathbf{L}_F]$
Propagator	$\mathbf{P}_{\leftarrow} = \mathbf{D}_{\leftarrow}^{-1/2}\mathbf{A}_{\leftarrow}\mathbf{D}_{\rightarrow}^{-1/2}$	$\mathbf{P}_{\rightarrow} = \mathbf{D}_{\rightarrow}^{-1/2}\mathbf{A}_{\rightarrow}\mathbf{D}_{\leftarrow}^{-1/2}$
Messages	$\mathbf{m}_{\leftarrow}^{(l)} = \mathbf{P}_{\leftarrow}\mathbf{F}^{(l-1)}$	$\mathbf{m}_{\rightarrow}^{(l)} = \mathbf{P}_{\rightarrow}\mathbf{F}^{(l-1)}$

Feature update

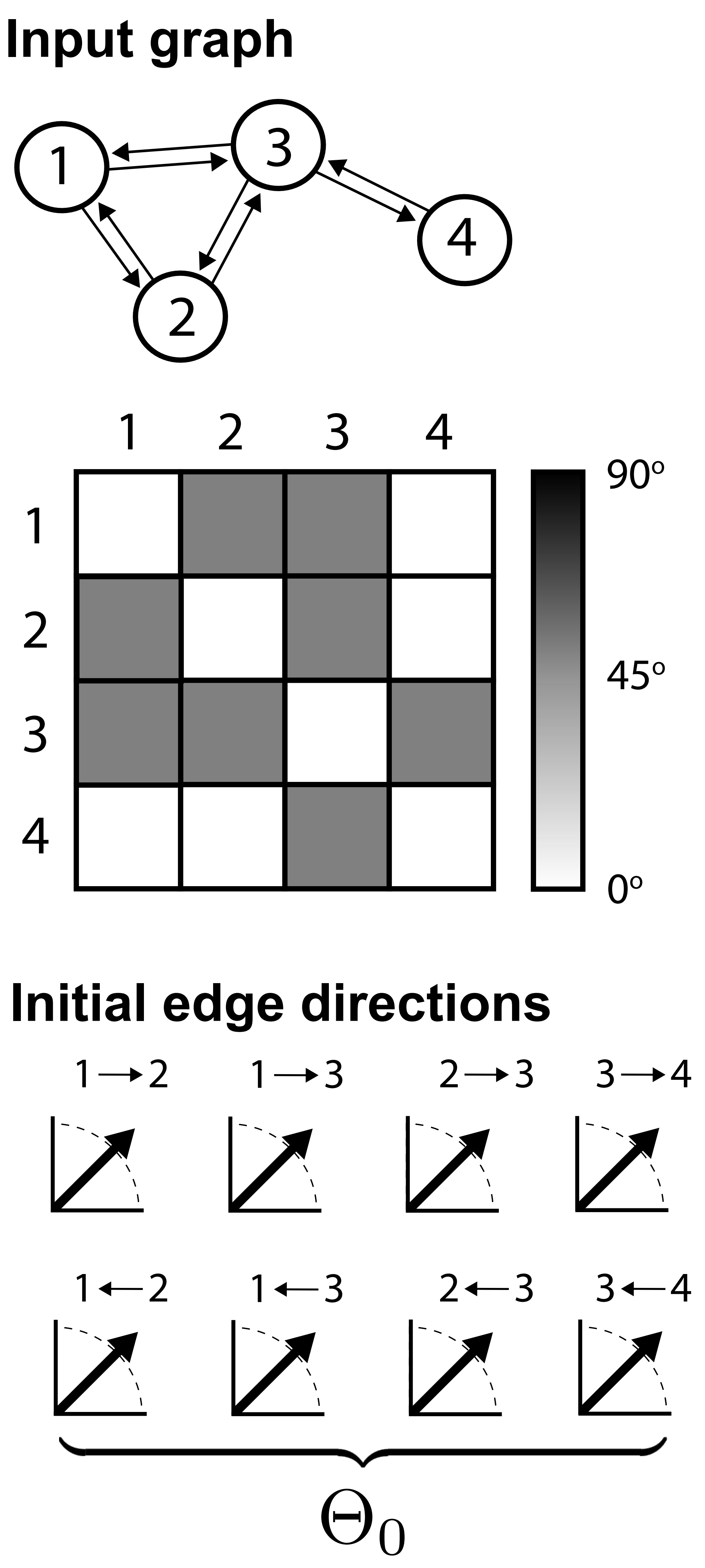
$$\mathbf{F}^{(l)} = \sigma\big(\mathbf{F}^{(l-1)}\mathbf{W}_{\mathrm{self}}^{(l)} + \mathbf{m}_{\leftarrow}^{(l)}\mathbf{W}_{\leftarrow}^{(l)} + \mathbf{m}_{\rightarrow}^{(l)}\mathbf{W}_{\rightarrow}^{(l)} + \mathbf{B}^{(l)}\big)$$

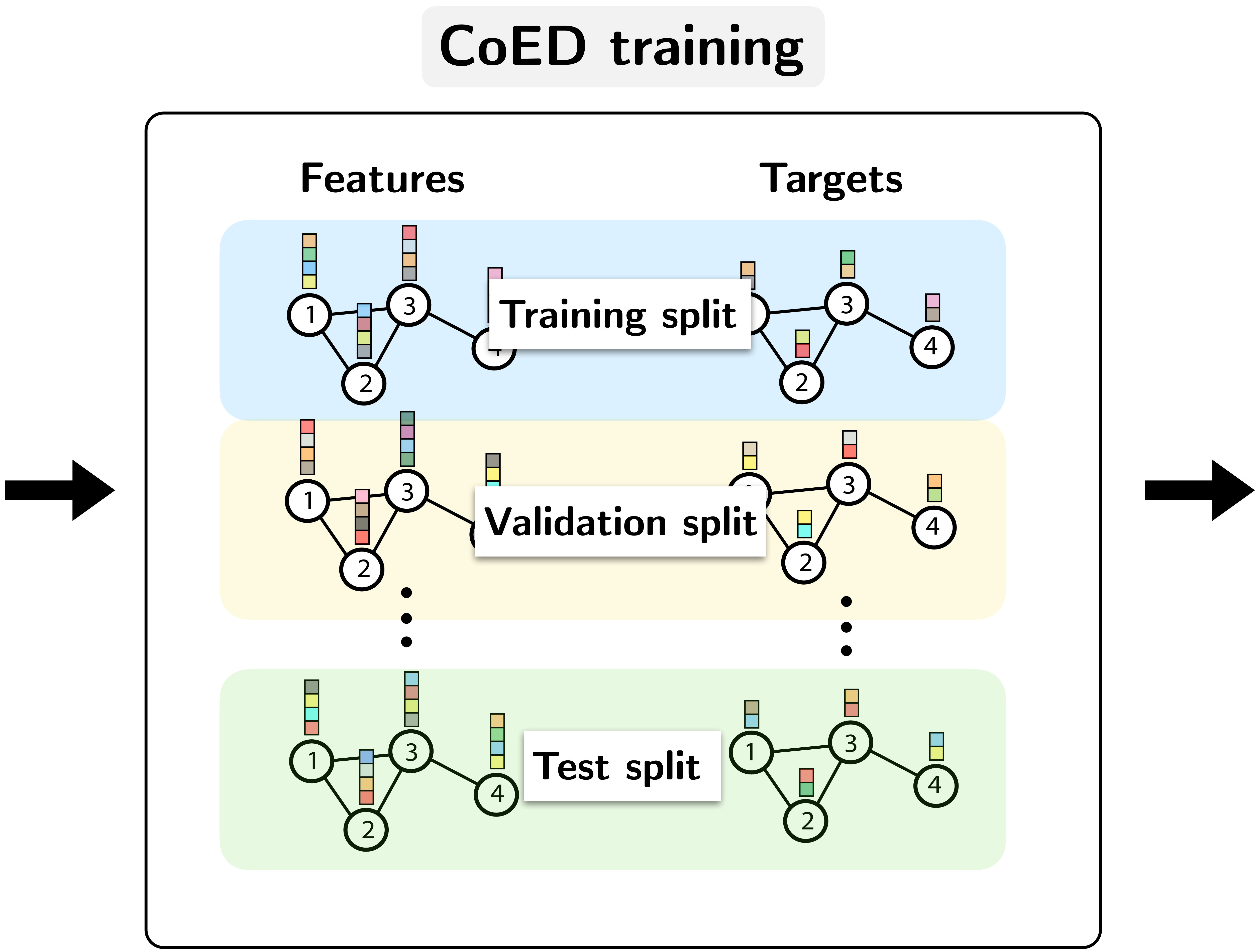
Learning edge directions (+GNN) on Graph Ensemble Data

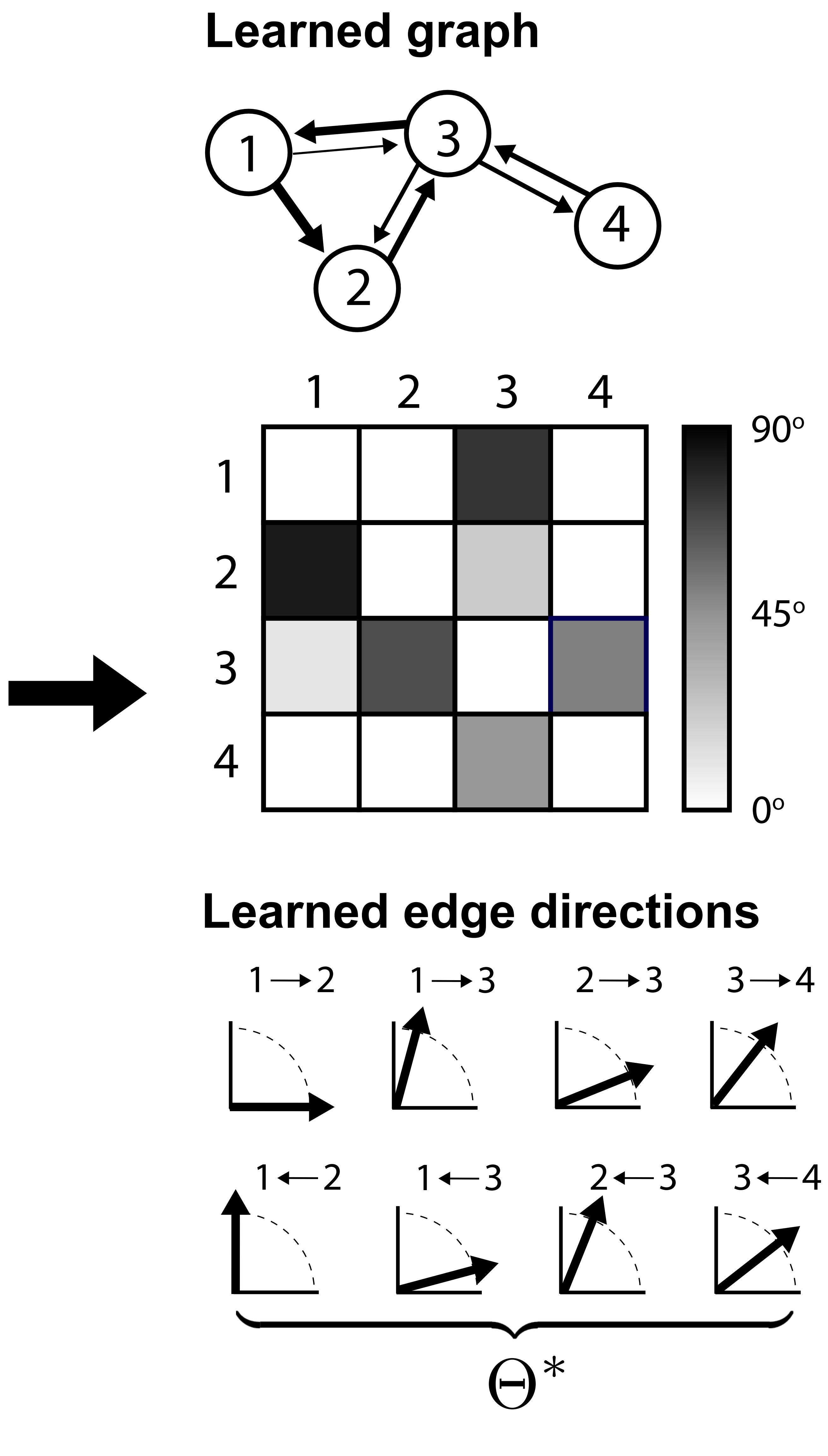
Graph ensemble data



Learning edge directions (+GNN) on Graph Ensemble Data



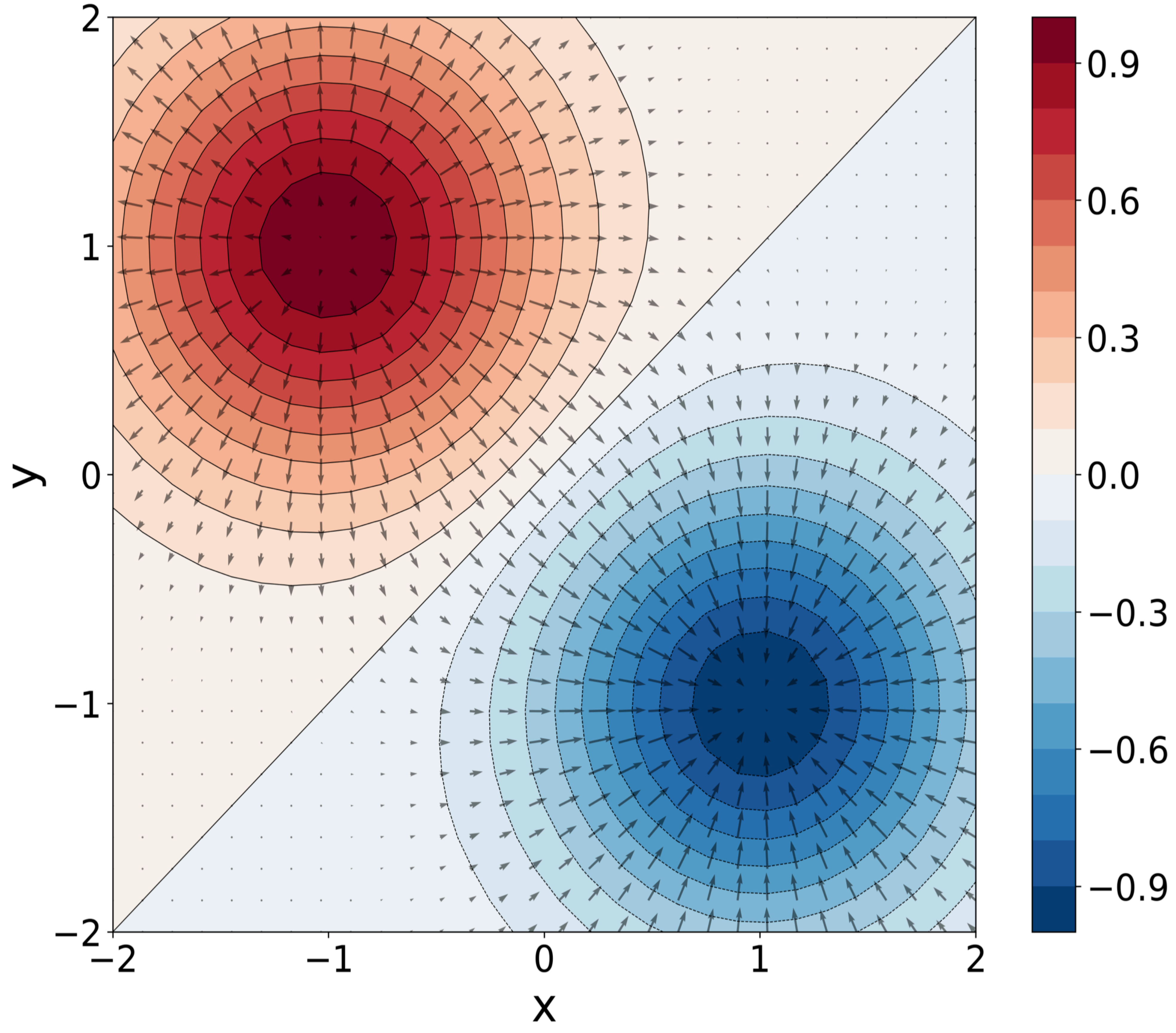




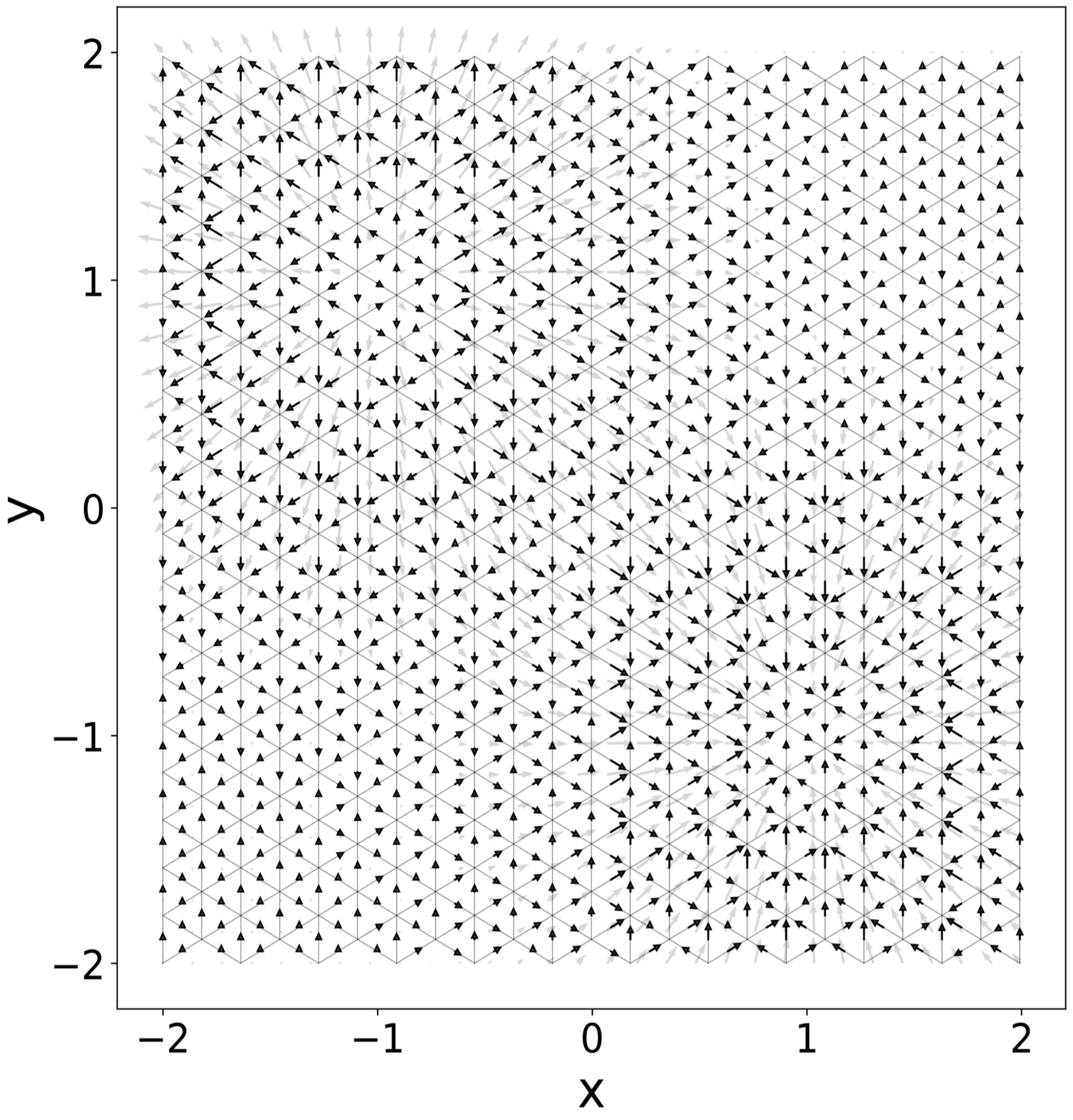
Experiments on synthetic graph ensemble dataset

1. Directed flow on triangular lattice graph

Gradient of 2-D potential $V \in [-2,2]^2$

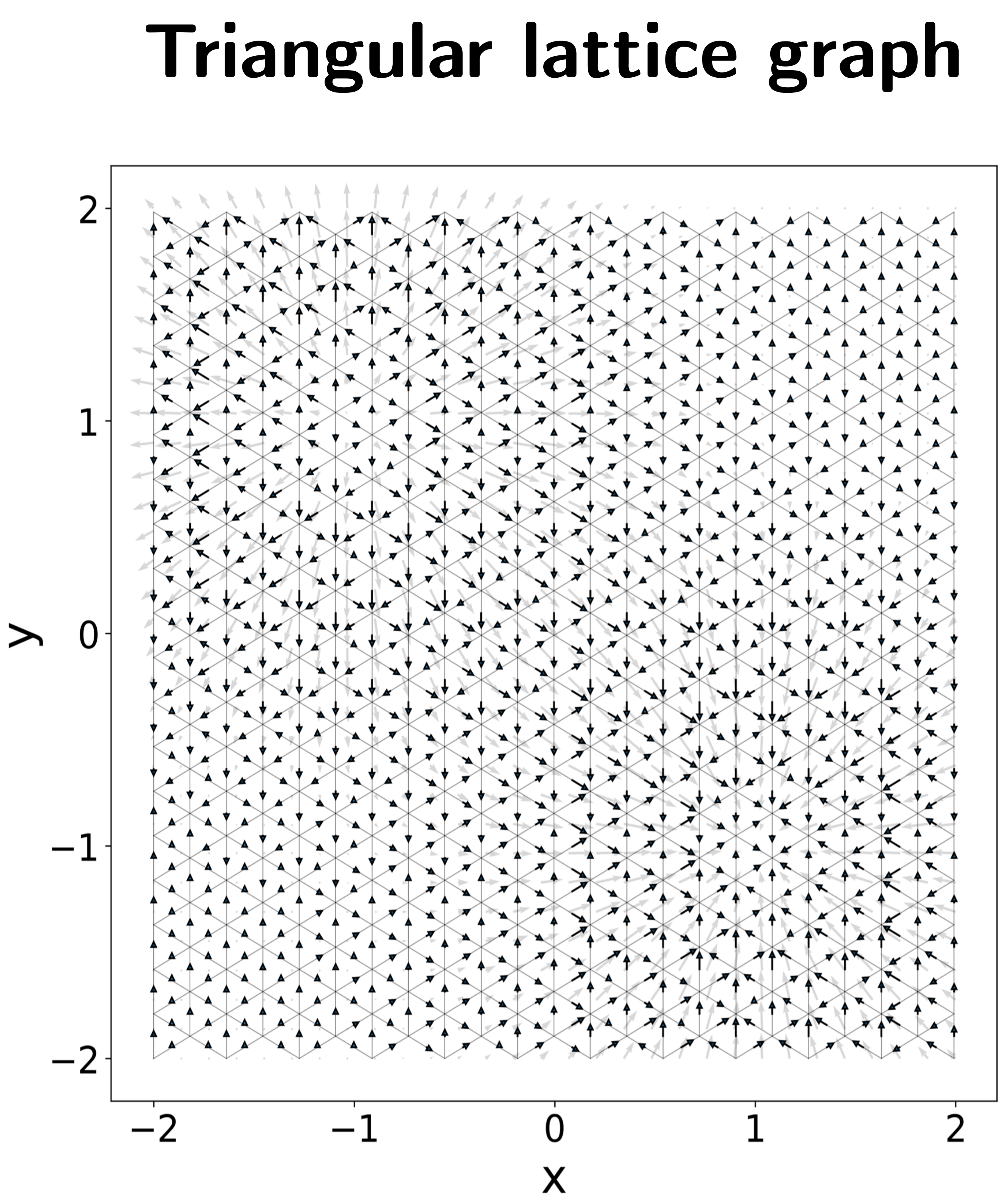


Triangular lattice graph



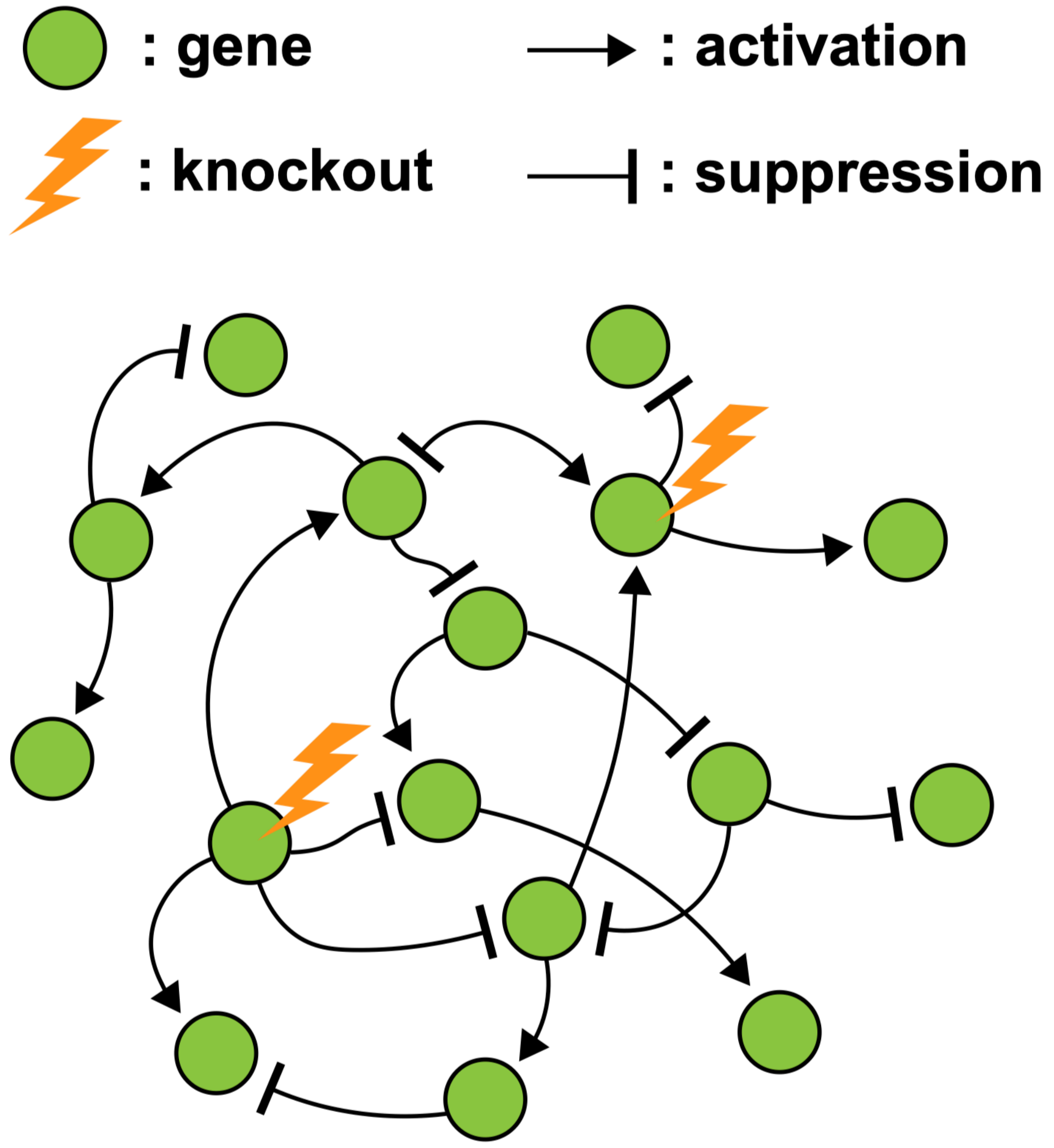
Experiments on synthetic graph ensemble dataset

1. Directed flow on triangular lattice graph



$$\mathbf{F}^{(l)} = \sigma\big(\mathbf{F}^{(l-1)}\mathbf{W}_{\mathcal{N}_{\text{self}}} + \mathbf{m}^{(l)}_{\leftarrow}\mathbf{W}_{\mathcal{N}_{\leftarrow}} + \mathbf{m}^{(l)}_{\rightarrow}\mathbf{W}_{\mathcal{N}_{\rightarrow}} + \mathbf{B}^{(l)}\big)$$

2. Gene regulatory network



$$\frac{dc_i}{dt} = \sum_{j \in \mathcal{N}(i)} \left(\gamma_{ij}^{\text{act}} F^{\text{act}}(c_j, K_{ij}) + \gamma_{ij}^{\text{sup}} F^{\text{sup}}(c_j, K_{ij})\right) - c_i$$

Experiments on synthetic graph ensemble dataset

- **For lattice graph, all models were shown undirected graph.**
- **For GRN graph, all models were shown the ground truth directed graph**

Best regression error

	Lattice	GRN
GCN	77.56 ±0.47	69.38 ±0.62
GAT	9.41 ±0.05	12.07 ±1.50
GraphGPS	3.47 ±0.14	25.16 ±1.56
MagNet	75.06 ±0.03	43.42 ±4.34
Chung	8.03 ±0.03	62.95±0.78
DRew	28.55 ±0.02	69.92 ±0.15
FLODE	7.54±0.05	70.31 ±0.03
CoED	**1.36 ±0.06**	**5.02 ±0.45**

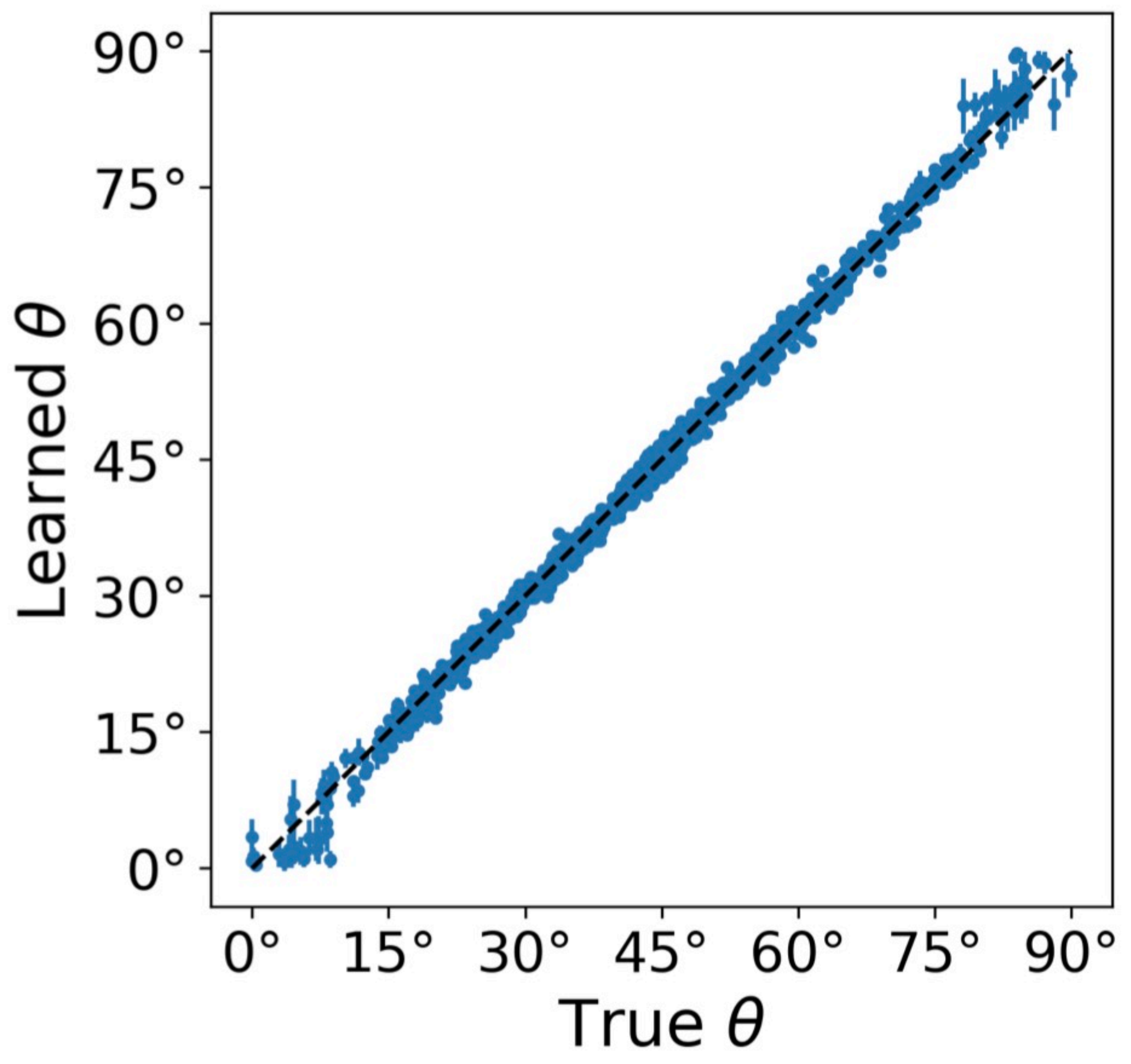
Experiments on synthetic graph ensemble dataset

- For lattice graph, all models were shown undirected graph.
- For GRN graph, all models were shown the ground truth directed graph

Best regression error

	Lattice	GRN
GCN	77.56 ±0.47	69.38 ±0.62
GAT	9.41 ±0.05	12.07 ±1.50
GraphGPS	3.47 ±0.14	25.16 ±1.56
MagNet	75.06 ±0.03	43.42 ±4.34
Chung	8.03 ±0.03	62.95±0.78
DRew	28.55 ±0.02	69.92 ±0.15
FLODE	7.54±0.05	70.31 ±0.03
CoED	**1.36 ±0.06**	**5.02 ±0.45**

Recovery of true angles



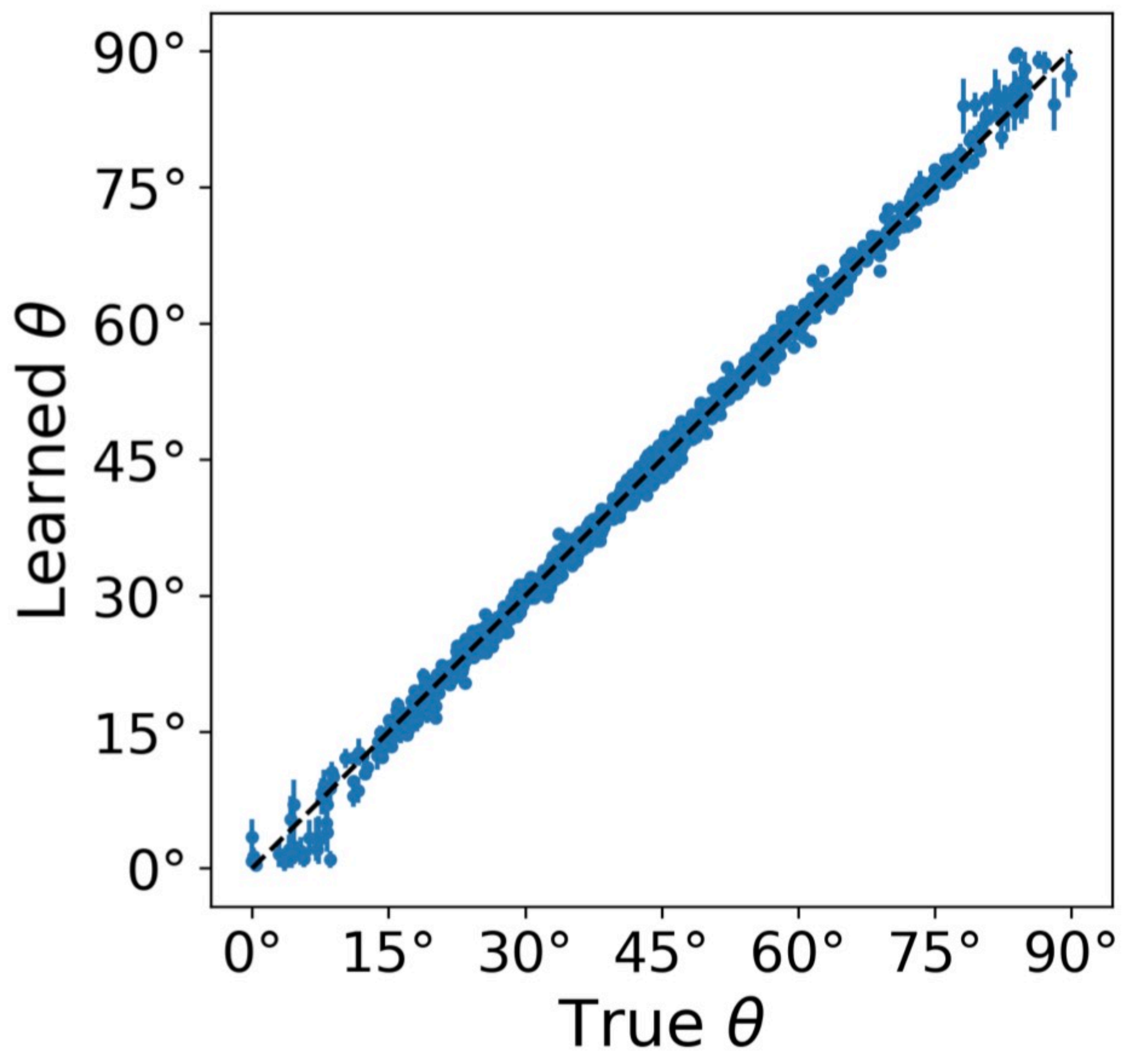
Experiments on synthetic graph ensemble dataset

- **For lattice graph, all models were shown undirected graph.**
- **For GRN graph, all models were shown the ground truth directed graph**

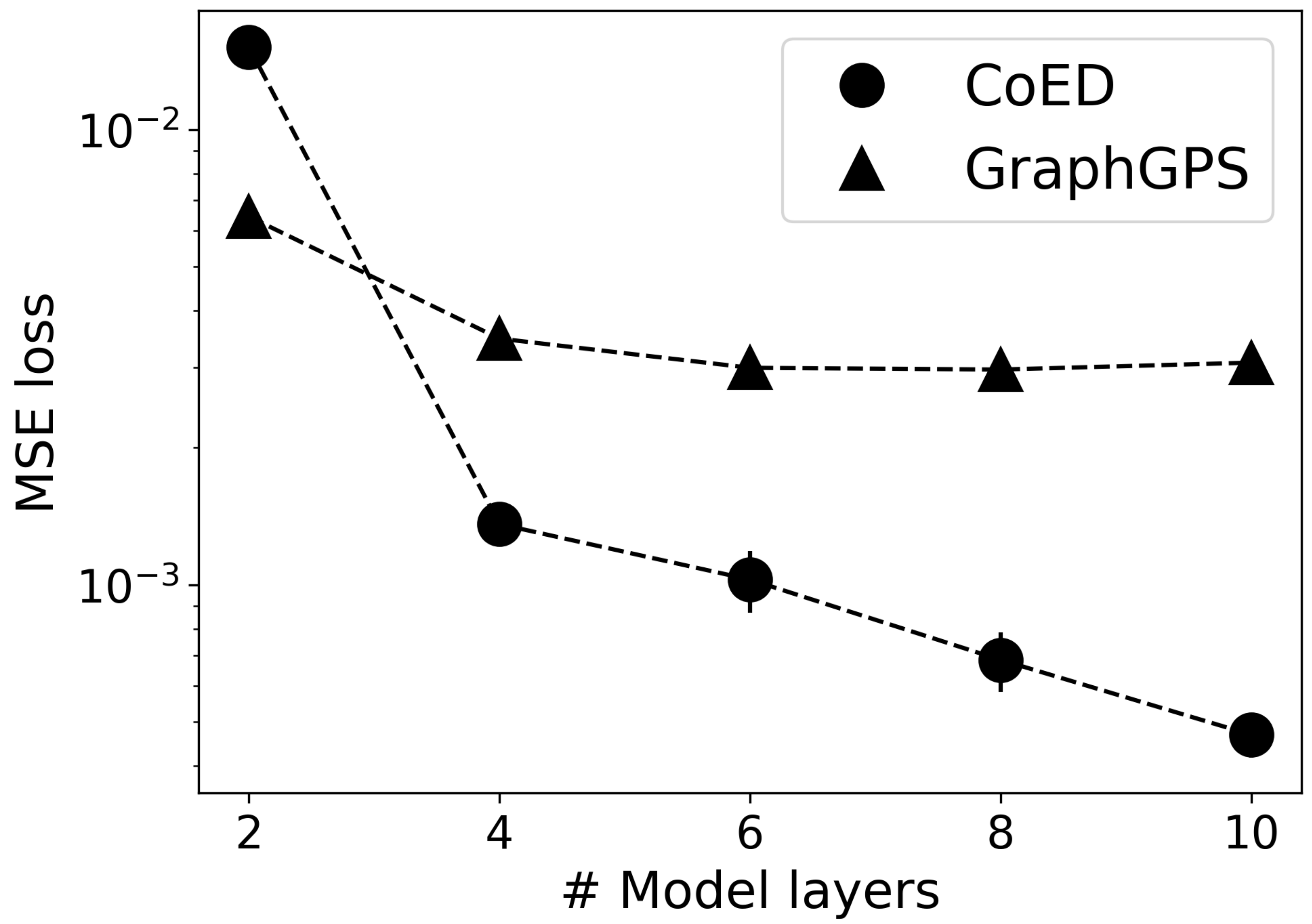
Best regression error

	Lattice	GRN
GCN	77.56 ±0.47	69.38 ±0.62
GAT	9.41 ±0.05	12.07 ±1.50
GraphGPS	3.47 ±0.14	25.16 ±1.56
MagNet	75.06 ±0.03	43.42 ±4.34
Chung	8.03 ±0.03	62.95±0.78
DRew	28.55 ±0.02	69.92 ±0.15
FLODE	7.54±0.05	70.31 ±0.03
CoED	**1.36 ±0.06**	**5.02 ±0.45**

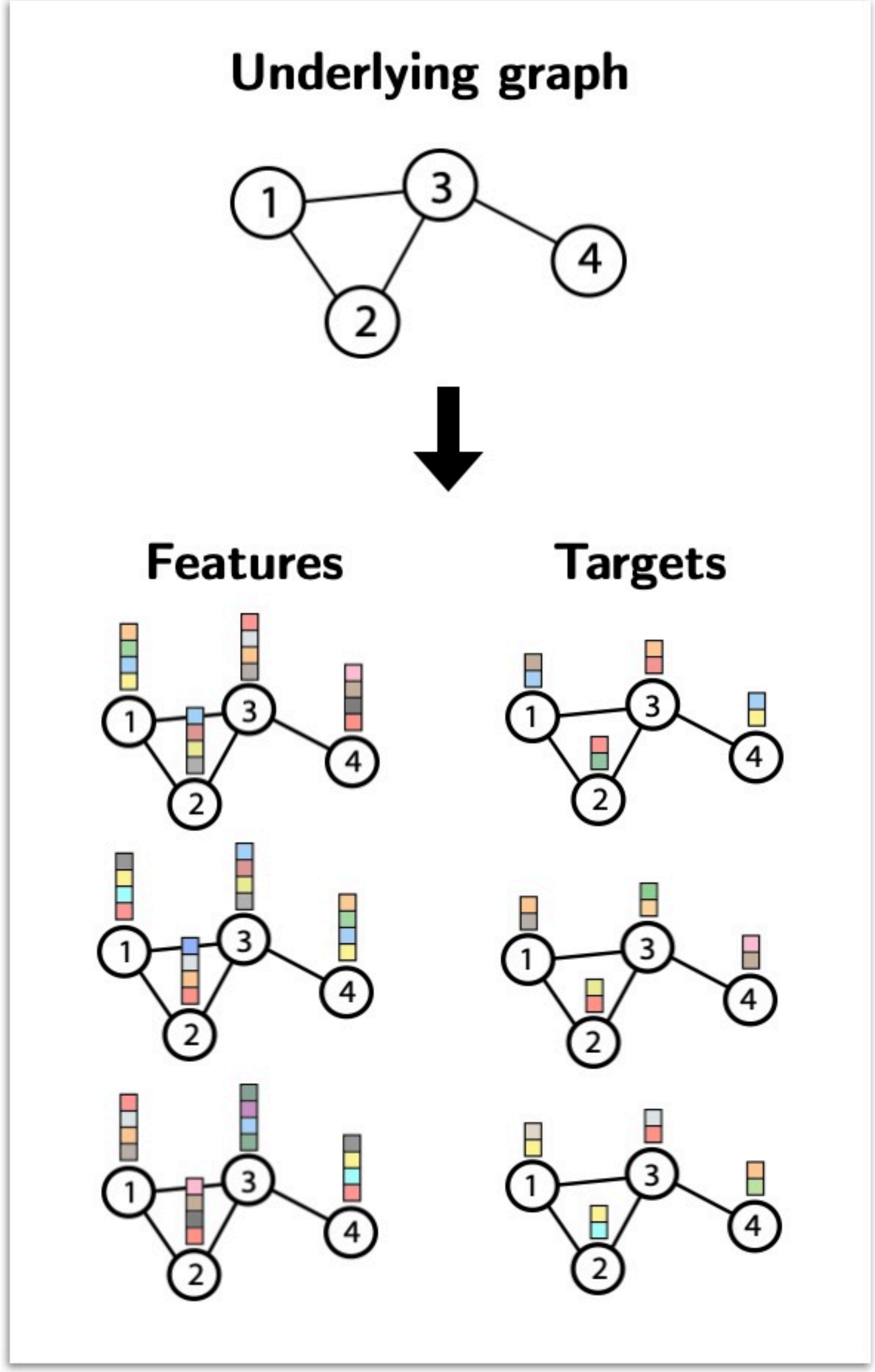
Recovery of true angles



CoED improves as depth ↑



Experiments on real graph ensemble dataset



Perturb-seq

Gene expression levels under different perturbations

Web traffic

Daily visit counts of webpages over time

Power grid

Optimal operating values under different loading conditions

Experiments on real graph ensemble dataset

- CoED achieves the lowest regression error

	Perturb-seq	Web traffic	Power grid
GCN	4.13±0.08	7.07±0.03	28.56±6.08
MagNet	4.11±0.01	6.94±0.02	18.05±2.77
GAT	3.85±0.03	6.00±0.03	13.57±1.73
DirGCN	5.46±0.26	6.72±0.04	6.15±0.84
DirGAT	3.98±0.07	6.55±0.04	3.28±0.17
CoED	**3.56±0.03**	**5.76±0.05**	**2.91±0.11**

Thank you for tuning in!

Paper: https://arxiv.org/abs/2410.14109

Code: https://github.com/hormoz-lab/coed-gnn