SDM 2023 Tutorial

Augmentation Methods for Graph Learning

Tong Zhao, Kaize Ding, Wei Jin, Gang Liu, Meng Jiang, Neil Shah

2023-04-27

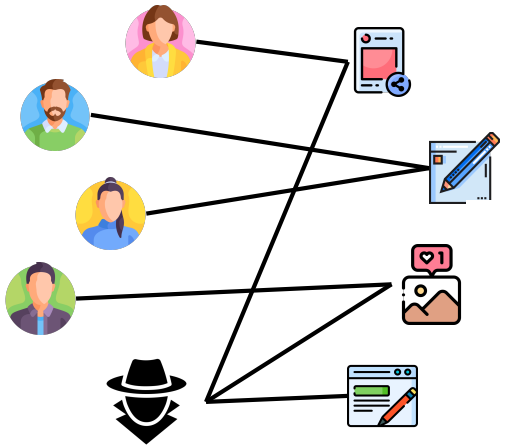
SIAM International Conference on Data Mining

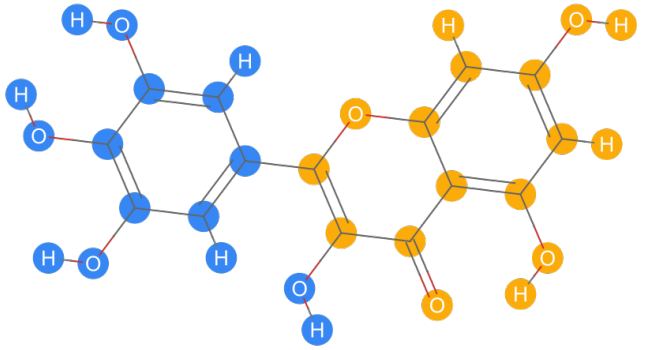




Graphs are everywhere

- Graph models interactions (edges) between objects (nodes).





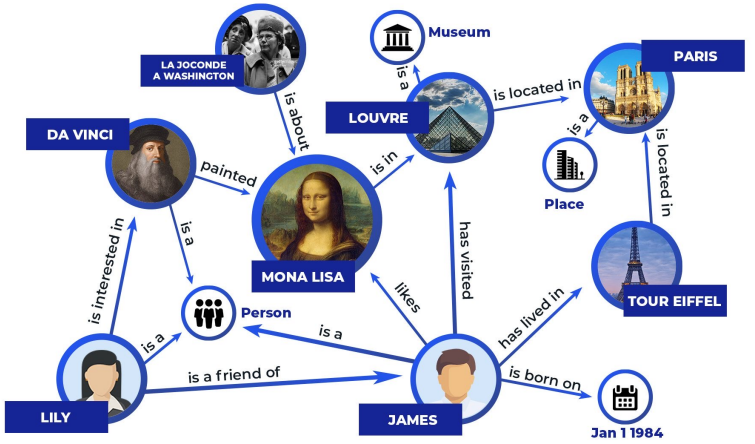
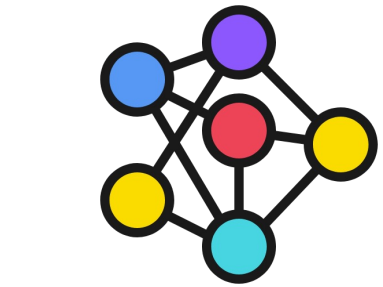


Image source: https://community.atlassian.com/t5/Confluence-questions/Knowledge-graph/qaq-p/1565284

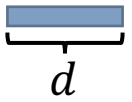
Background: Graph Machine Learning (GML)

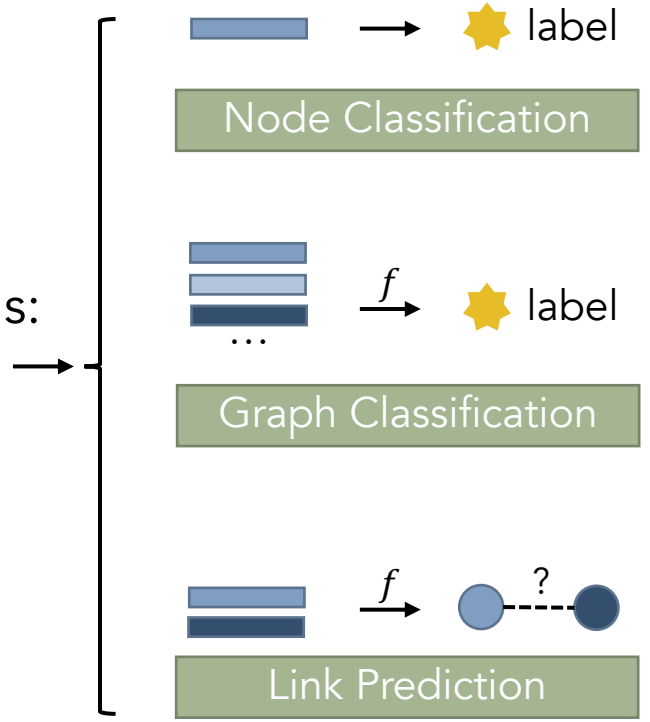
- GML learns low-dimensional representations from graph data.



Given graph data:
$G = (\mathcal{V}, \mathcal{E})$
raw node attributes (optional):
$\mathbf{x}_v \in \mathbb{R}^m, v \in \mathcal{V}$

Node representations:
$\boldsymbol{z}_v \in \mathbb{R}^d$

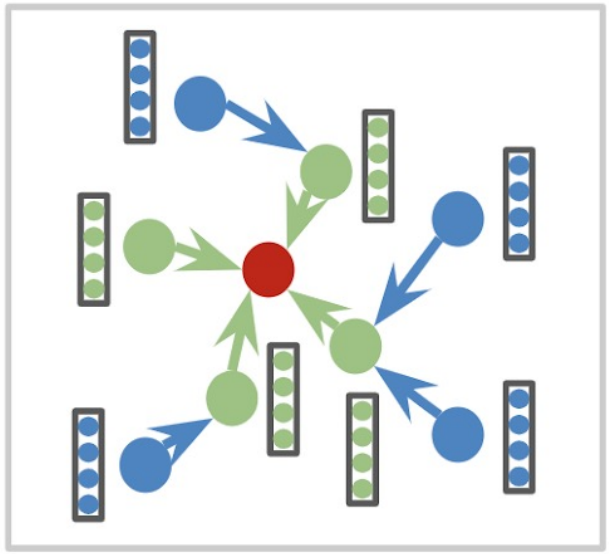




Background: Graph Neural Networks (GNNs)

Neighborhood Aggregation:

- Generate node representations based on local neighborhoods.

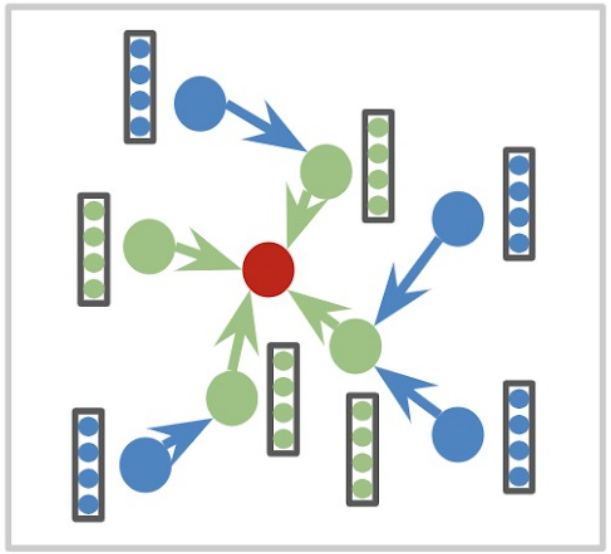


$$\boldsymbol{h}^l_{\mathcal{N}(v)} = \textsc{Aggregate}(\{\boldsymbol{h}^{l-1}_u | u \in \mathcal{N}(v)\}),$$
$$\boldsymbol{h}^l_v = \textsc{Update}(\boldsymbol{h}^{l-1}_v, \boldsymbol{h}^l_{\mathcal{N}(v)}).$$

Background: Graph Neural Networks (GNNs)

Neighborhood Aggregation:

- Generate node representations based on local neighborhoods.



Layer output embedding Layer weight matrix

$$\mathbf{H}^{(l)} = \sigma(\hat{\mathbf{A}}\mathbf{H}^{(l-1)}\mathbf{W}^{(l)})$$

Non-linearity

Aggregated neighbor embeddings normalized

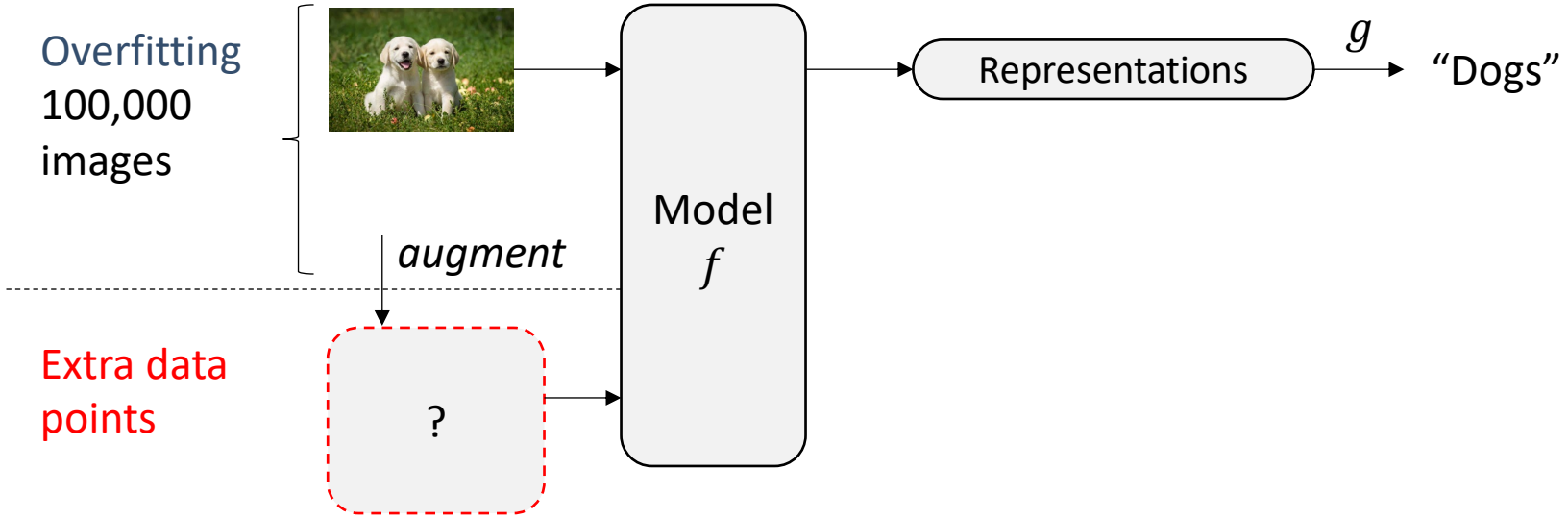
Background: Data Augmentation

- Wikipedia: Techniques used to increase the amount of data by adding *slightly modified* copies of already existing data or *newly created* synthetic data from existing data.

- Why data augmentation?
  - It helps reduce overfitting when training a machine learning model.
  - The acquisition of labeled graph data can be expensive.

Background: Data Augmentation

- Wikipedia: Techniques used to increase the amount of data by adding *slightly modified* copies of already existing data or *newly created* synthetic data from existing data.



Background: Data Augmentation

- Wikipedia: Techniques used to increase the amount of data by adding *slightly modified* copies of already existing data or *newly created* synthetic data from existing data.

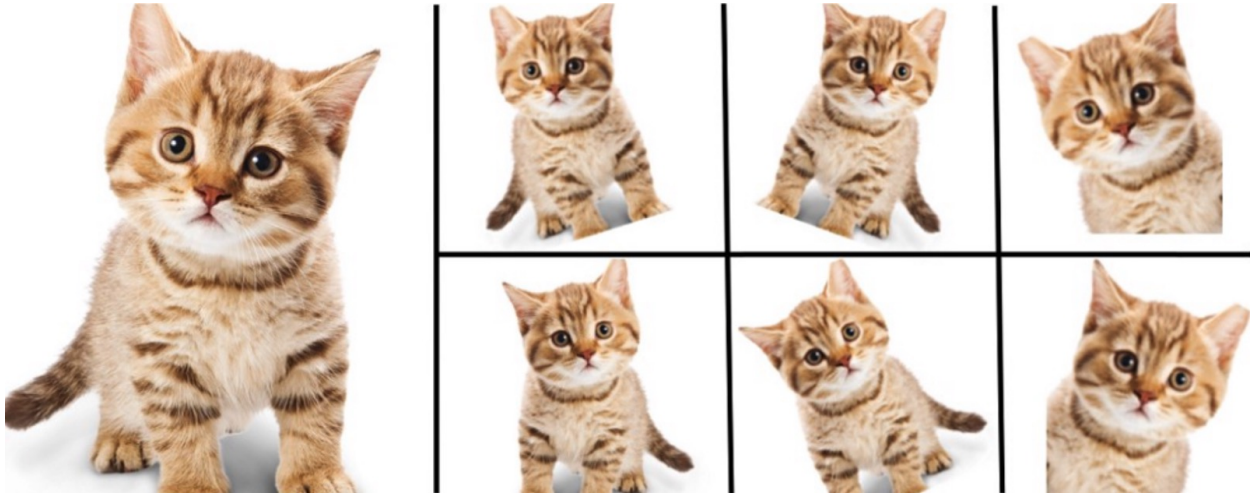


Image sources:
https://www.kdnuggets.com/2018/05/data-augmentation-deep-learning-limited-data.html
https://amitness.com/2020/05/data-augmentation-for-nlp/

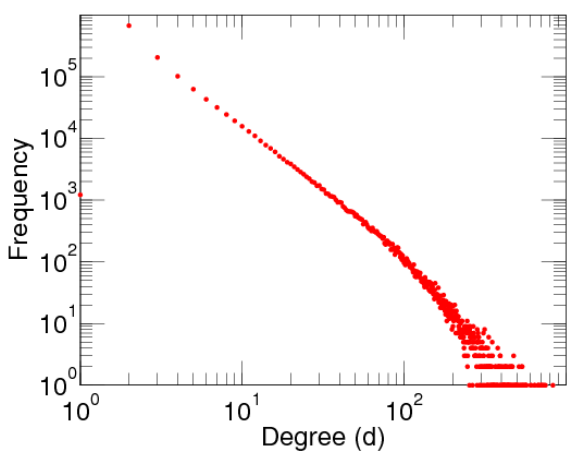
Why augment graphs?

1. Graphs are sparse.

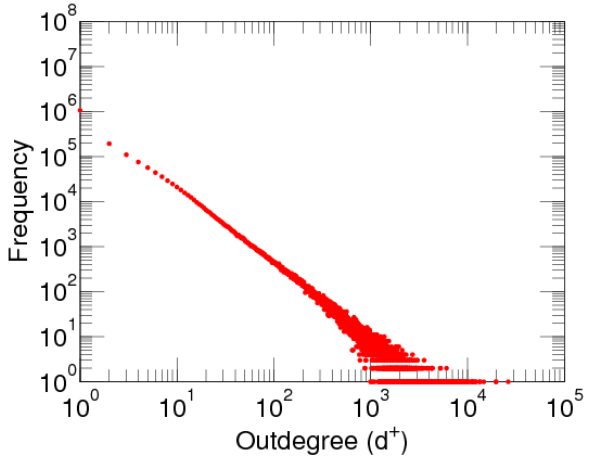
Impact: Overfitting, Poor performance.

Observation: Power-law distribution.

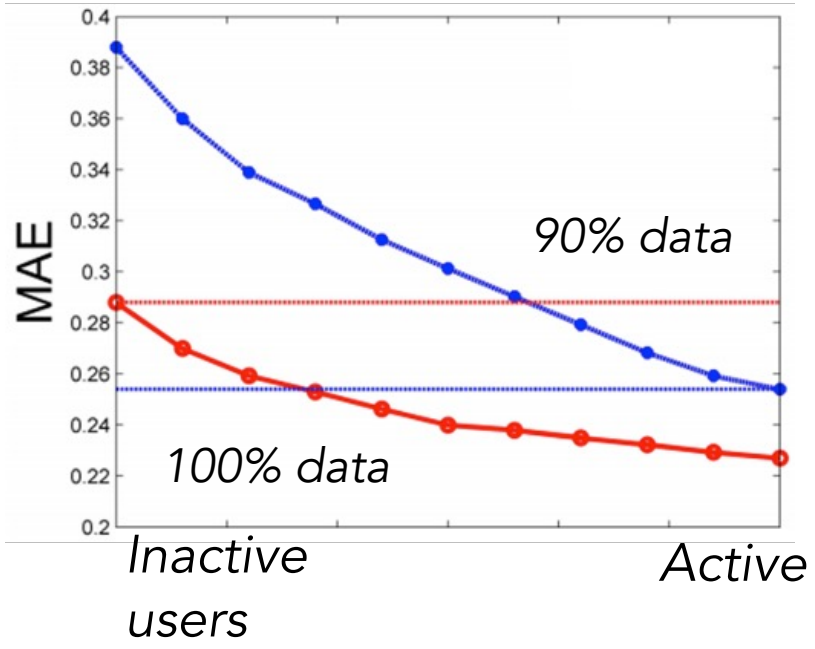
Netflix movies



Flickr friendship



Movie rating prediction on a "user-gives-rating-to-movie" graph



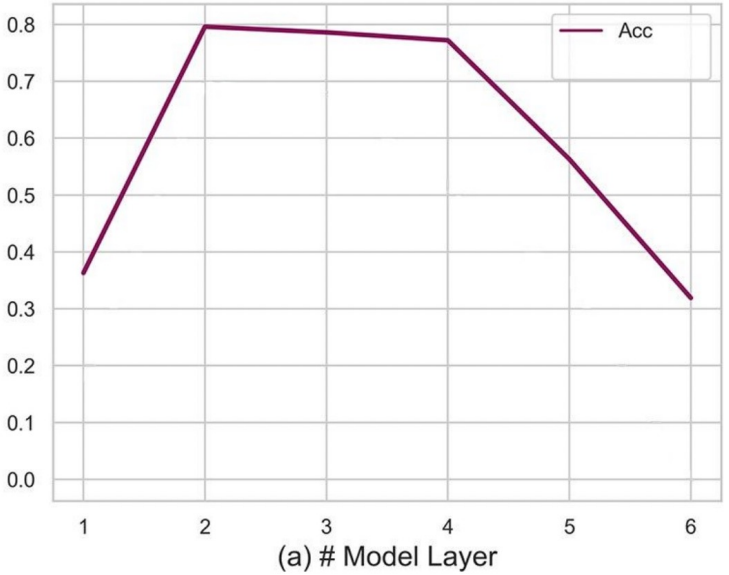
Why augment graphs?

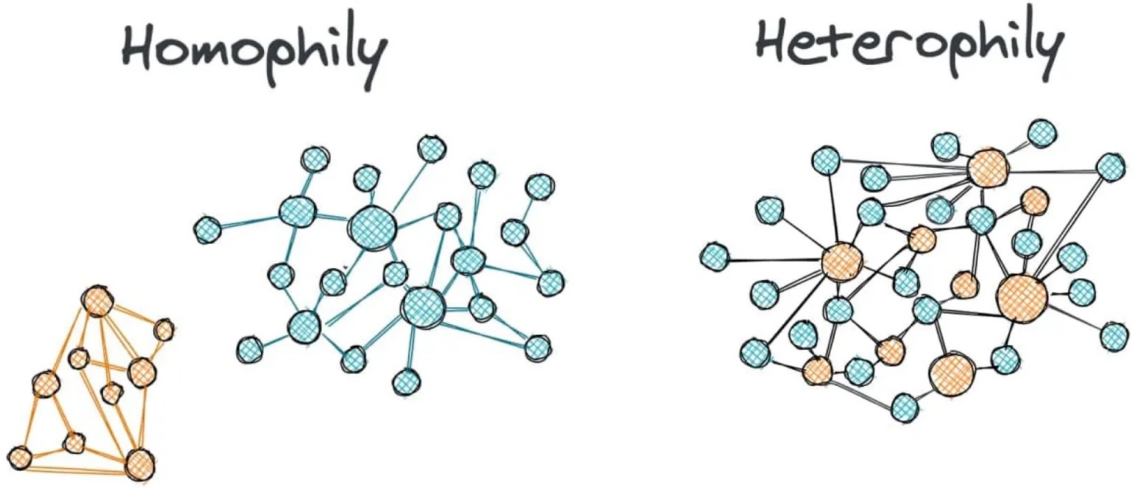
1. Graphs are sparse.
2. Real world graphs are incomplete and noisy.

Image sources:
https://xkcd.com/285/
https://medium.com/analytics-vidhya/social-network-analytics-f082f4e21b16

Why augment graphs?

1. Graphs are sparse.
2. Real world graphs are incomplete and noisy.
3. Over-smoothing and homophily assumption.

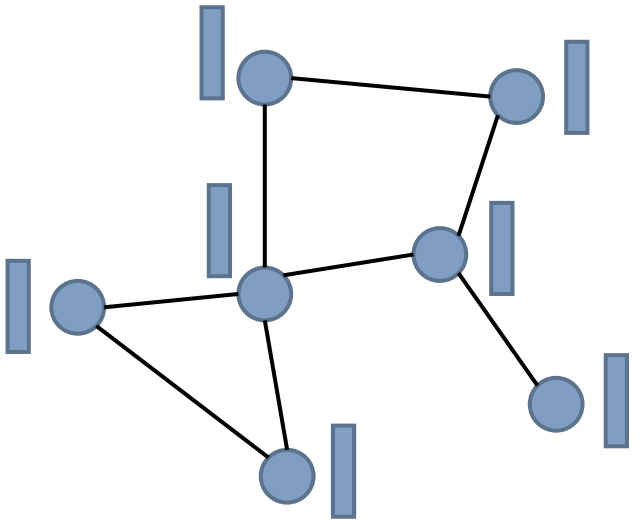




Chen, et al. Measuring and Relieving the Over-smoothing Problem for Graph Neural Networks from the Topological View. AAAI 2020.
https://graphml.substack.com/p/gml-newsletter-homophily-heterophily

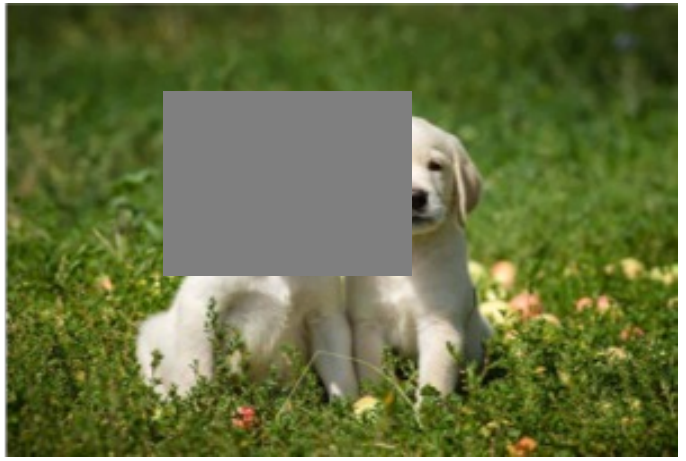
Graph Data Augmentation

- Structure Augmentation
  - Drop/add nodes/edges, etc.
- Feature Augmentation
  - Mask off features, etc.
- Label Augmentation
  - Label propagation, etc.



Rule-based vs. Learned Augmentations

- Rule-based augmentations
  - Designed based on heuristic rules
  - Usually efficient and scalable
  - Simple and easy to implement
    - Commonly used in self-supervised learning



- Learned augmentations
  - Involve learning during augmentation
  - Augmented data better fits GML models
    - Better performances in supervised learning

Outline

- Introduction
- Rule-based Augmentation Approaches
- Learned Augmentation Approaches
- Applications and Future Directions

Rule-based Augmentation Approaches

Tong Zhao, Snap Inc.

	Methodology	Representative Works	Task Level			Augmented Data		
			Node	Graph	Edge	Structure	Feature	Label
Rule-based GDA	Stochastic Dropping/Masking	DropEdge [87]	✓			✓		
		DropNode [27]		✓			✓	
		NodeDropping [127]		✓		✓		
		Feature Masking [100]	✓				✓	
		Feature Shuffling [106]	✓				✓	
		DropMessage [23]	✓		✓		✓	
		Subgraph Masking [127]		✓		✓	✓	
	Subgraph Cropping/Substituting	GraphCrop [111]		✓		✓		
		M-Evolve [145]		✓		✓		
		MoCL [97]		✓		✓	✓	
	Virtual Node	Graphormer [125]		✓		✓		
		GNN-CM$^+$/CM [45]			✓	✓		
	Mixup	Graph Mixup [115]	✓	✓				✓
		ifMixup [37]		✓		✓	✓	✓
		Graph Transparent [85]		✓		✓	✓	✓
		G-Mixup [39]		✓		✓	✓	✓
	SMOTE	GraphSMOTE [140]	✓				✓	
		GATSMOTE [75]	✓			✓		
		GNN-CL [70]	✓			✓	✓	
	Diffusion	GDA [60]	✓			✓		
	Counterfactual Augmentation	CFLP [141]			✓	✓		✓
	Attribute Augmentation	LA-GNN [74]	✓				✓	
		SR+DR [93]	✓				✓	
	Pseudo-labeling	Label Propagation [147]	✓					✓
		PTA [21]	✓					✓

Zhao, et al. Graph Data Augmentation for Graph Machine Learning: A Survey. 2022.

Rule-based Augmentation Approaches

-- Stochastic Dropping/Masking

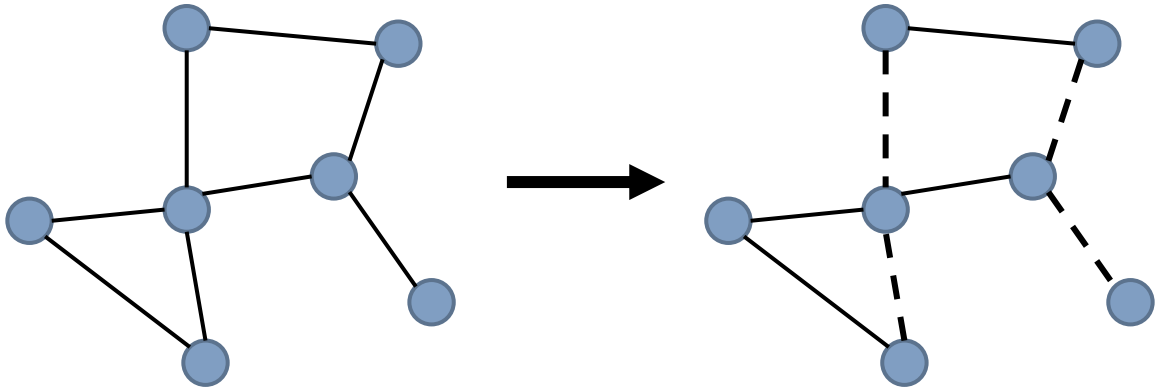
DropEdge

- Dropout on edges: randomly remove some edges at the beginning of every training epoch.

$$\tilde{\mathbf{A}} = \mathbf{M} \odot \mathbf{A}$$

$$\mathbf{M} \in \{0, 1\}^{N \times N} \text{ s.t. } M_{i,j} = Bernoulli(\varepsilon)$$

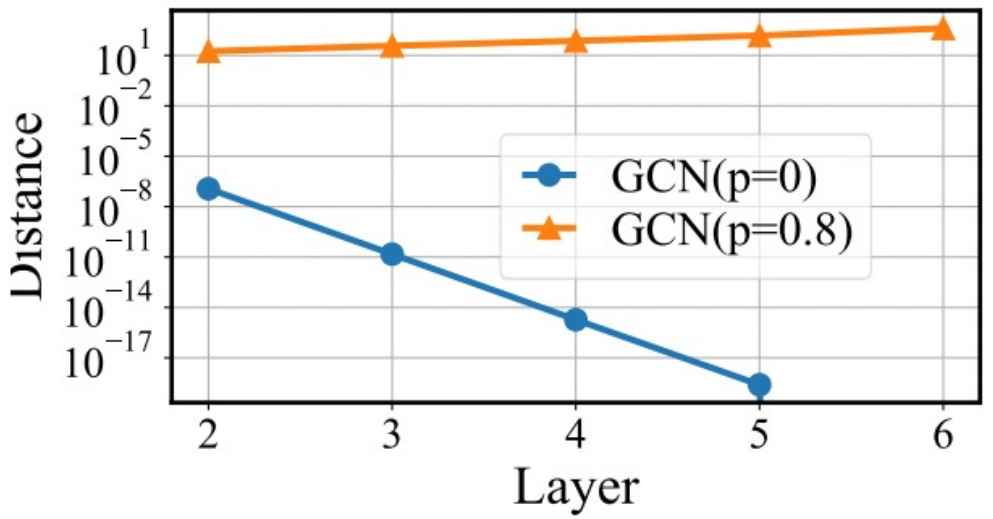
- Prevents overfitting and over-smoothing.



Yu, et al. DropEdge: Towards Deep Graph Convolutional Networks on Node Classification. ICLR 2020.

DropEdge

- Prevents over-smoothing.



Dataset	Backbone	2 layers		8 layers		32 layers	
		Orignal	DropEdge	Orignal	DropEdge	Orignal	DropEdge
Citeseer	GCN	75.90	**78.70**	74.60	**77.20**	59.20	**61.40**
	ResGCN	-	-	77.80	**78.80**	74.40	**77.90**
	JKNet	-	-	79.20	**80.20**	71.70	**80.00**
	IncepGCN	-	-	79.60	**80.50**	72.60	**80.30**
	GraphSAGE	78.40	**80.00**	74.10	**77.10**	37.00	**53.60**

Yu, et al. DropEdge: Towards Deep Graph Convolutional Networks on Node Classification. ICLR 2020.

Other Stochastic Masking/Dropping Methods

- Node Dropping
  - Randomly removing part of the nodes.
- Feature Masking
  - Randomly mask off node features.
  - Random row-shuffling on node feature matrix $\mathbf{X}$.
- Subgraph Masking
  - Randomly mask off a connected subgraph.

Feng, et al. Graph Random Neural Networks for Semi-supervised Learning on Graphs. NeurIPS 2020.
You, et al. Graph Contrastive Learning with Augmentations. NeurIPS 2020.
Thakoor, et al. Large-scale Representation Learning on Graphs via Bootstrapping. ICLR 2022.
Velickovic, et al. Deep Graph Infomax. ICLR 2019.

DropMessage

- Generic formulation of message passing-based GNNs:

$$\boldsymbol{h}^l_{\mathcal{N}(v)} = \textsc{Aggregate}(\{\boldsymbol{h}^{l-1}_u | u \in \mathcal{N}(v)\}),$$
$$\boldsymbol{h}^l_v = \textsc{Update}(\boldsymbol{h}^{l-1}_v, \boldsymbol{h}^l_{\mathcal{N}(v)}).$$

- DropMessage: random element-wise masking on the message vectors $\boldsymbol{h}^l_{\mathcal{N}(v)}$ for each node.

- Advantage: small sample variance
  - DropMessage presents the smaller sample variance than Dropout, DropEdge, and DropNode on message passing GNNs with same dropping rate.

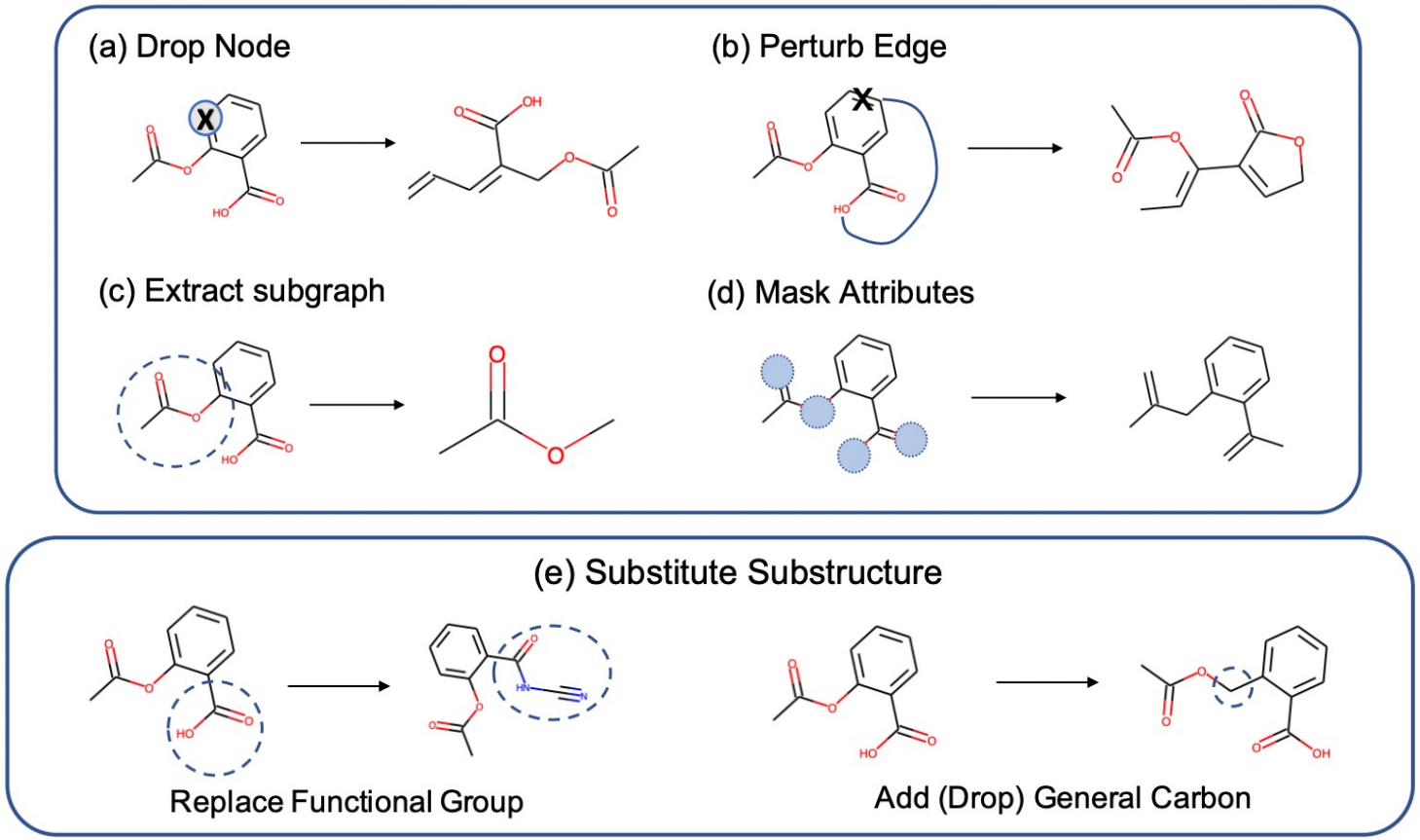
Fang, et al. DropMessage: Unifying Random Dropping for Graph Neural Networks. AAAI 2023.

Rule-based Augmentation Approaches

-- Subgraph Substituting

MoCL

- Subgraph substituting with domain knowledge.



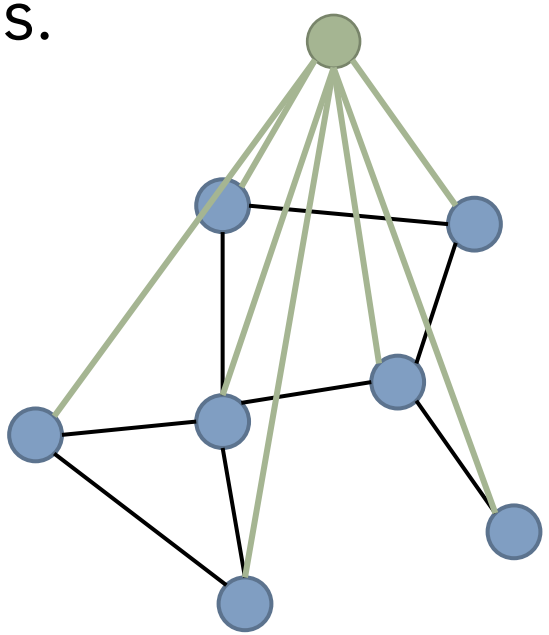
Sun, et al. MoCL: Data-driven Molecular Fingerprint via Knowledge-aware Contrastive Learning from Molecular Graph. KDD 2021.

Rule-based Augmentation Approaches

-- Virtual Node

Virtual Node

- An additional node that connects to all nodes.
  - Mostly used in graph-level tasks.
  - Similar to `[CLS]` token in language modeling.
  - Learns graph representation in parallel with the node representations.



- GNN-CM: virtual node for link prediction.
  - One common virtual node for each cluster in the graph.

Gilmer, et al. Neural Message Passing for Quantum Chemistry. ICML 2017.
Pham, et al. Graph Classification via Deep Learning with Virtual Nodes. IJCAI 2017.
Ying, et al. Do Transformers really Perform Badly for Graph Representations? NeurIPS 2021.
Hwang, et al. Revisiting Virtual Nodes in Graph Neural Networks for Link Prediction. 2022.

Rule-based Augmentation Approaches

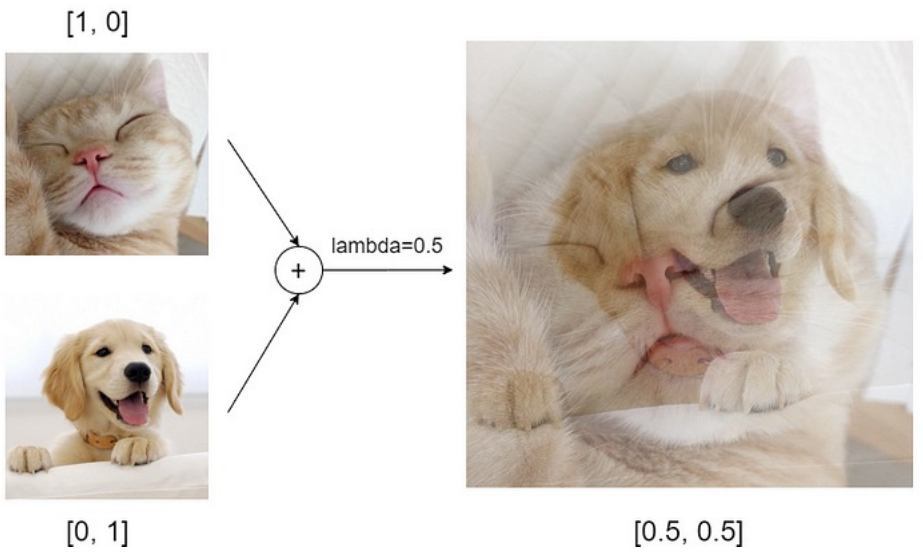
-- Mixup

Background: Mixup

- Mixup: generates a weighted combination of random pairs from the training data.

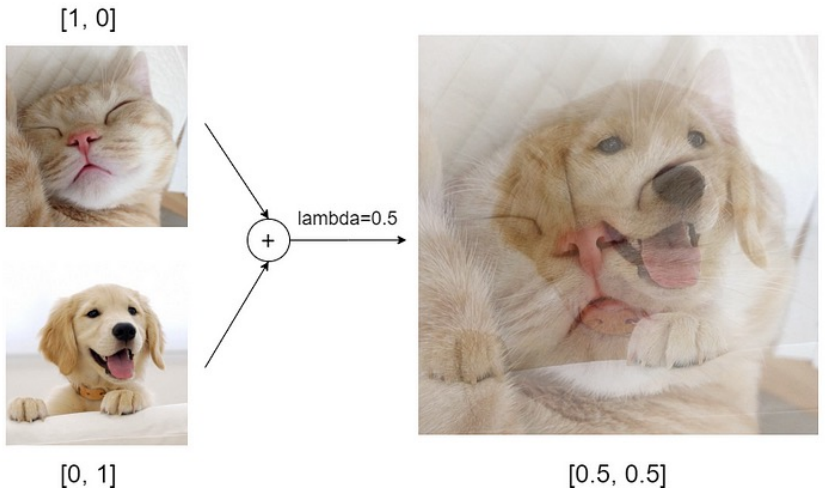
$$\tilde{\boldsymbol{x}} = \lambda \boldsymbol{x}_i + (1-\lambda)\boldsymbol{x}_j,$$
$$\tilde{\boldsymbol{y}} = \lambda \boldsymbol{y}_i + (1-\lambda)\boldsymbol{y}_j.$$

- Manifold Mixup: interpolating hidden states.

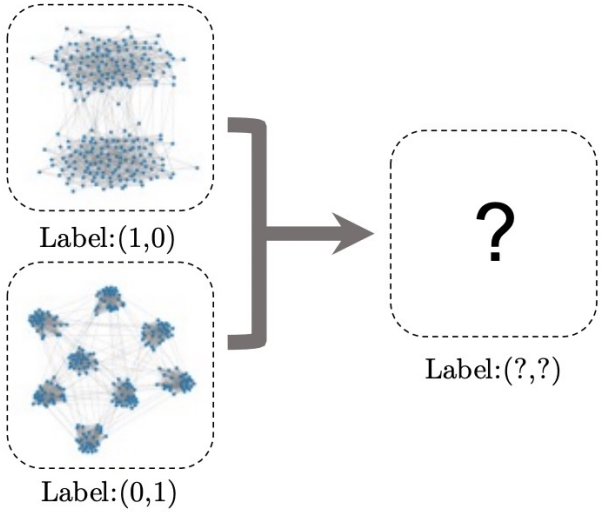


Zhang, et al. Mixup: Beyond Empirical Risk Minimization. ICLR 2018.
Verma, et al. Manifold Mixup: Better Representations by Interpolating Hidden States. ICML 2019.
Image source: https://medium.com/@wolframalphav1.0/easy-way-to-improve-image-classifier-performance-part-1-mixup-augmentation-with-codes-33288db92de5

Challenges for Mixup on Graphs

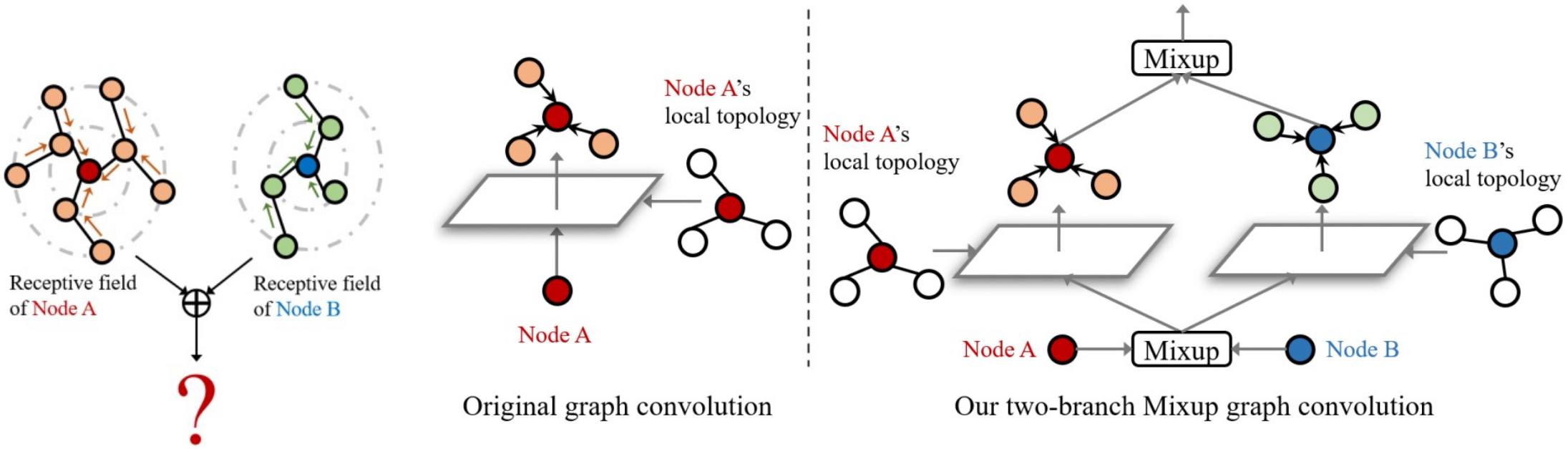


1. Image data is regular
2. Images are well-aligned
3. Images are in Euclidean space



1. Graph data is irregular
2. Graphs are not well-aligned
3. Graphs are in non-Euclidean space

Graph Mixup



Mixup on node features

$$\tilde{\mathbf{x}}_{ij} = \lambda \mathbf{x}_i + (1-\lambda)\mathbf{x}_j,$$

$$\tilde{\mathbf{h}}_{ij,i}^{(l)} = \text{AGGREGATE}\left(\tilde{\mathbf{h}}_{ij}^{(l-1)}, \left\{\mathbf{h}_k^{(l-1)} \middle| k \in \mathcal{N}(i)\right\}, \mathbf{W}^{(l)}\right),$$

$$\tilde{\mathbf{h}}_{ij,j}^{(l)} = \text{AGGREGATE}\left(\tilde{\mathbf{h}}_{ij}^{(l-1)}, \left\{\mathbf{h}_k^{(l-1)} \middle| k \in \mathcal{N}(j)\right\}, \mathbf{W}^{(l)}\right),$$

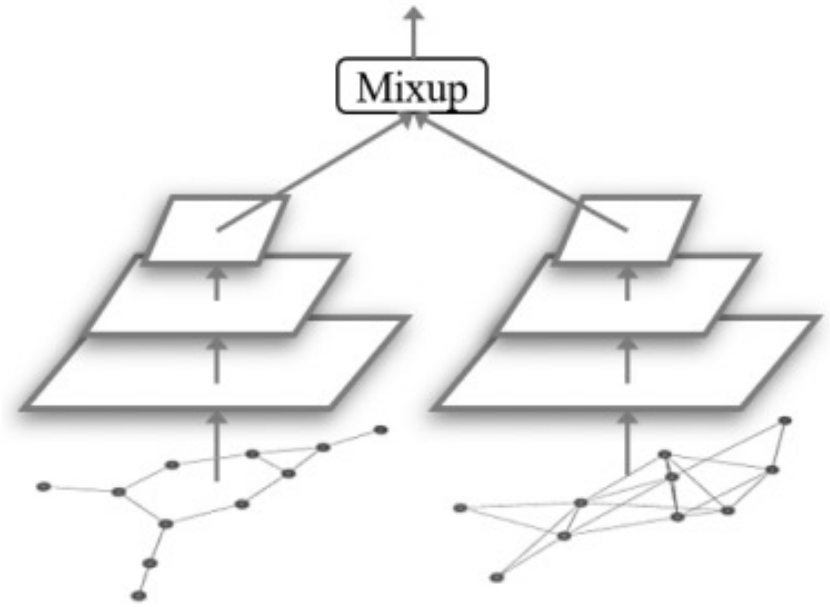
Mixup on neighbor features

$$\tilde{\mathbf{h}}_{ij}^{(l)} = \lambda \tilde{\mathbf{h}}_{ij,i}^{(l)} + (1-\lambda)\tilde{\mathbf{h}}_{ij,j}^{(l)},$$

Wang, et al. Mixup for Node and Graph Classification. TheWebConf 2021.

Graph Mixup

- Graph Mixup for graph representation learning.
  - Interpolating the latent representations of two graphs.



$$\tilde{\mathbf{h}}_{G_1 G_2} = \lambda \mathbf{h}_{G_1} + (1-\lambda)\mathbf{h}_{G_2},$$
$$\tilde{\mathbf{y}}_{G_1 G_2} = \lambda \mathbf{y}_{G_1} + (1-\lambda)\mathbf{y}_{G_2}.$$

Wang, et al. Mixup for Node and Graph Classification. TheWebConf 2021.

ifMixup

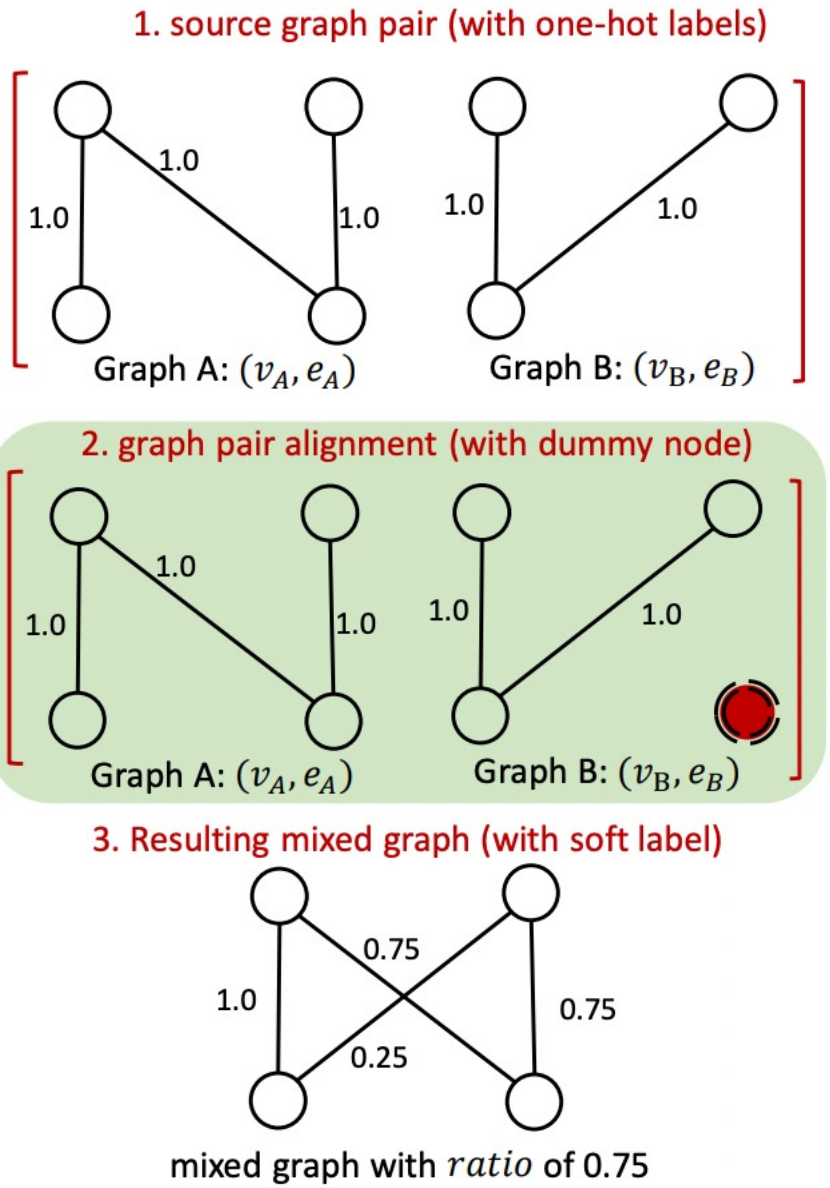
- Motivation: Can we directly mix up a pair of graph inputs?

Algorithm 1: The mixing schema in ifMixup

Input: a graph pair $G_A = (v_A, e_A)$ and $G_B = (v_B, e_B)$ (all edges in e have weight 1) **Parameter**: mixing ratio $\lambda \in (0, 1)$

Output: a mixed graph $\widetilde{G} = (\widetilde{v}, \widetilde{e})$

Compute max node number, $n = \max(n_A, n_B)$
if $n_A < n$ **then**
    Add $n - n_A$ dummy nodes to G_A
else if $n_B < n$ **then**
    Add $n - n_B$ dummy nodes to G_B
end if
$\widetilde{e} = \lambda e_A + (1 - \lambda) e_B$
$\widetilde{v} = \lambda v_A + (1 - \lambda) v_B$
return mixed graph $\widetilde{G} = (\widetilde{v}, \widetilde{e})$



Guo, et al. ifMixup: Interpolating Graph Pair to Regularize Graph Classification. AAAI 2023.

ifMixup

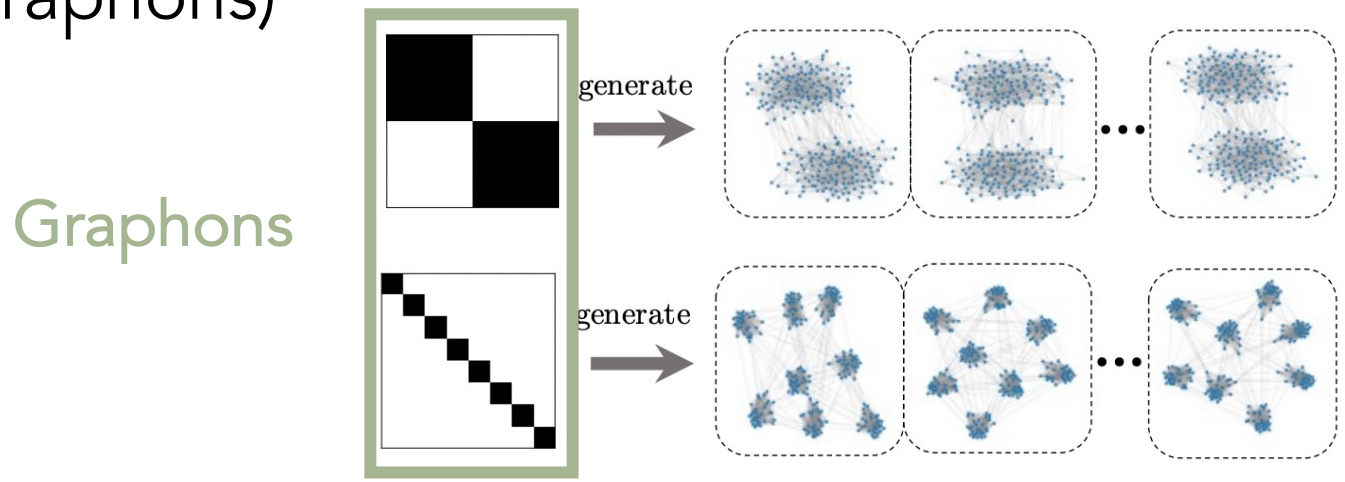
- Outperforms Graph Mixup and other stochastic dropping augmentations on graph classification.

	GIN Baseline	ifMixup	MixupGraph	DropEdge	DropNode	Attr. Masking
PTC_MR	0.644±0.007	**0.672±0.005**	0.631±0.005	0.669±0.003	0.663±0.006	0.657± 0.004
NCI109	0.820±0.002	**0.837±0.004**	0.822±0.008	0.792±0.002	0.796±0.002	0.822± 0.002
NCI1	0.818±0.009	**0.839±0.004**	0.822±0.001	0.791±0.005	0.785±0.003	0.825± 0.002
MUTAG	0.886±0.011	**0.890±0.006**	0.884±0.009	0.854±0.003	0.859±0.003	0.881± 0.004
ENZYMES	0.526±0.014	0.543±0.005	0.521±0.007	0.488±0.015	0.528±0.002	**0.544± 0.013**
PROTEINS	0.745±0.003	**0.754±0.002**	0.744±0.005	0.749±0.002	0.751±0.005	0.748 ±0.010
IMDB-M	0.519±0.001	**0.532±0.001**	0.518±0.004	0.517±0.003	0.516±0.002	0.520± 0.004
IMDB-B	0.762±0.004	**0.765±0.005**	0.761±0.001	0.762±0.005	0.764±0.006	0.761± 0.004

Guo, et al. ifMixup: Interpolating Graph Pair to Regularize Graph Classification. AAAI 2023.

G-Mixup

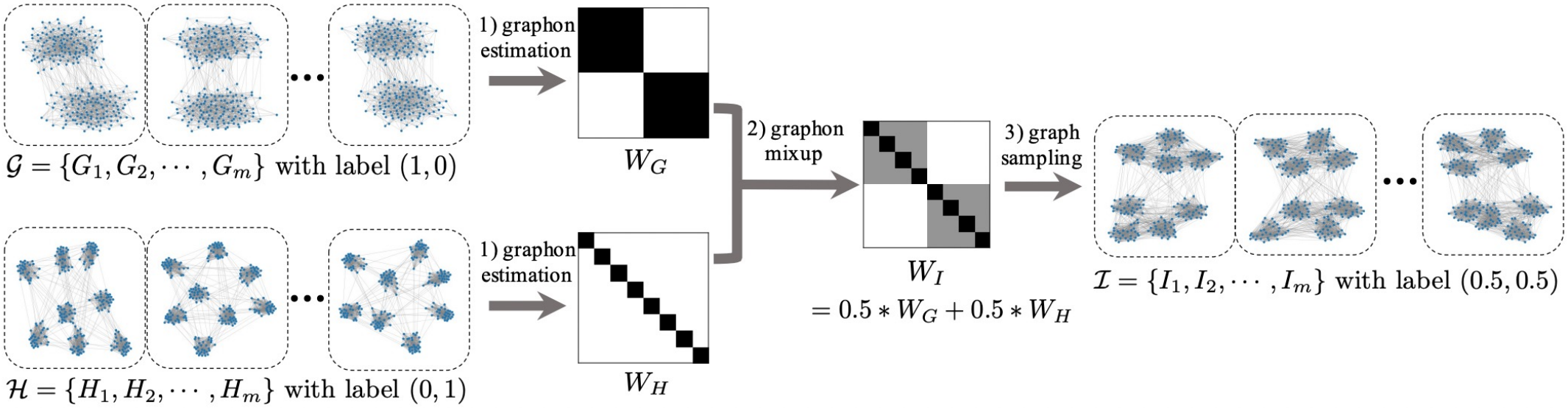
- Idea: Mixup on classes instead of data examples.
- Motivation: graphs can be generated from generators (e.g., graphons)



- Graphons are **regular**, **well-aligned**, and in **Euclidean space**.

Han, et al. G-Mixup: Graph Data Augmentation for Graph Classification. ICML 2022.

G-Mixup



1. Graphon estimation: $\mathcal{G} \to W_{\mathcal{G}}, \mathcal{H} \to W_{\mathcal{H}}$
2. Graphon Mixup: $W_{\mathcal{I}} = \lambda W_{\mathcal{G}} + (1 - \lambda) W_{\mathcal{H}}$
3. Graph Generation: $\{I_1, I_2, \cdots, I_m\} \overset{\text{i.i.d}}{\sim} \mathbb{G}(K, W_{\mathcal{I}})$
4. Label Mixup: $\mathbf{y}_{\mathcal{I}} = \lambda \mathbf{y}_{\mathcal{G}} + (1 - \lambda) \mathbf{y}_{\mathcal{H}}$

Han, et al. G-Mixup: Graph Data Augmentation for Graph Classification. ICML 2022.

Rule-based Augmentation Approaches

-- Diffusion

Graph Diffusion Convolution

- Motivation: message passing GNN layers are limited with one-hop neighbors.

- Proposed: generalized graph diffusion

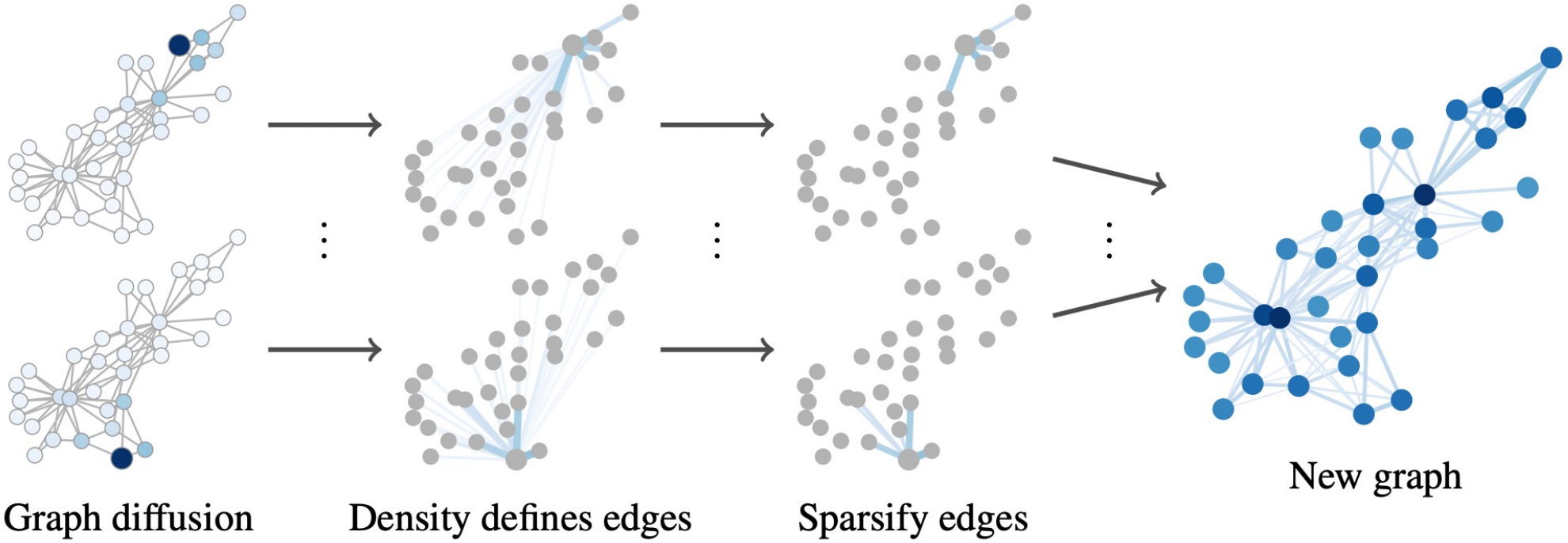
$$\tilde{\mathbf{A}} = \sum_{k=0}^{\infty} \theta_k \mathbf{T}^k$$

- $\mathbf{T}$: transition matrix. E.g., $\mathbf{T} = \mathbf{A}\mathbf{D}^{-1}$
- θ : pre-defined diffusion variants. E.g., $\theta_k^{\mathrm{PPR}} = \alpha(1-\alpha)^k$

$$\tilde{\mathbf{A}}^{\mathrm{PPR}} = \alpha(\mathbf{I}_N - (1-\alpha)\mathbf{T})^{-1}$$

Gasteiger, et al. Diffusion Improves Graph Learning. NeurIPS 2019.

Graph Diffusion Convolution



- Sparsification:
  - Top-k entries per column or thresholding on edge weights.

Gasteiger, et al. Diffusion Improves Graph Learning. NeurIPS 2019.

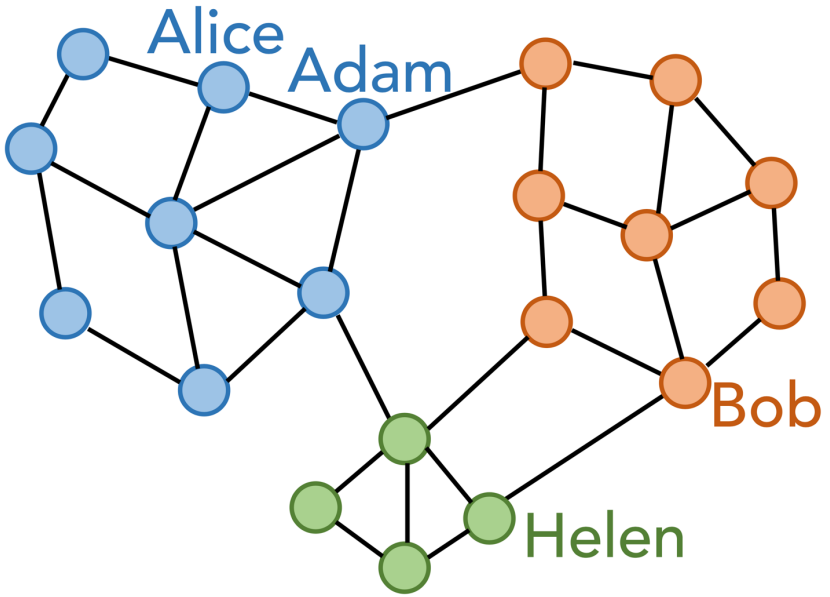
Rule-based Augmentation Approaches

-- Counterfactual Augmentation

Counterfactual Links

Alice and Adam have their personal interests,
AND they were in the same community, AND they were friends.

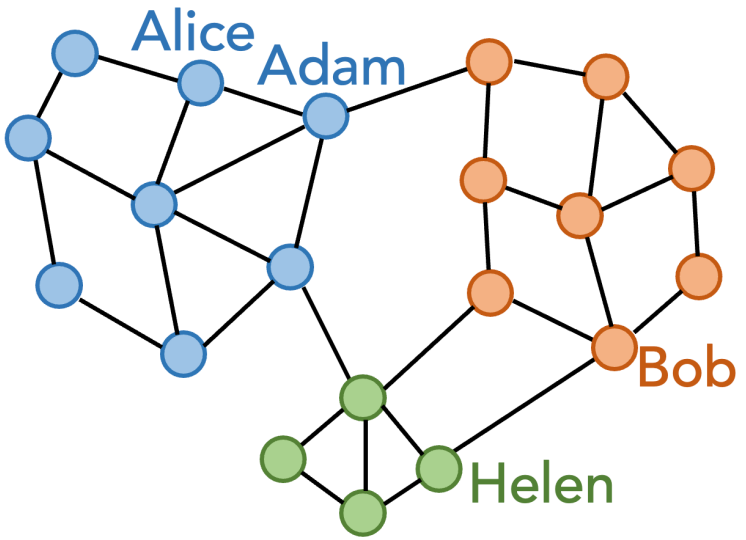
Counterfactual question:
If they were NOT in the same community,
would they still be friends?



If Helen is very similar with Alice,
AND Bob is very similar with Adam,
THEN the answer is YES, because Helen and Bob are friends.

Zhao, et al. Learning from Counterfactual Links for Link Prediction. ICML 2022.

Counterfactual Links



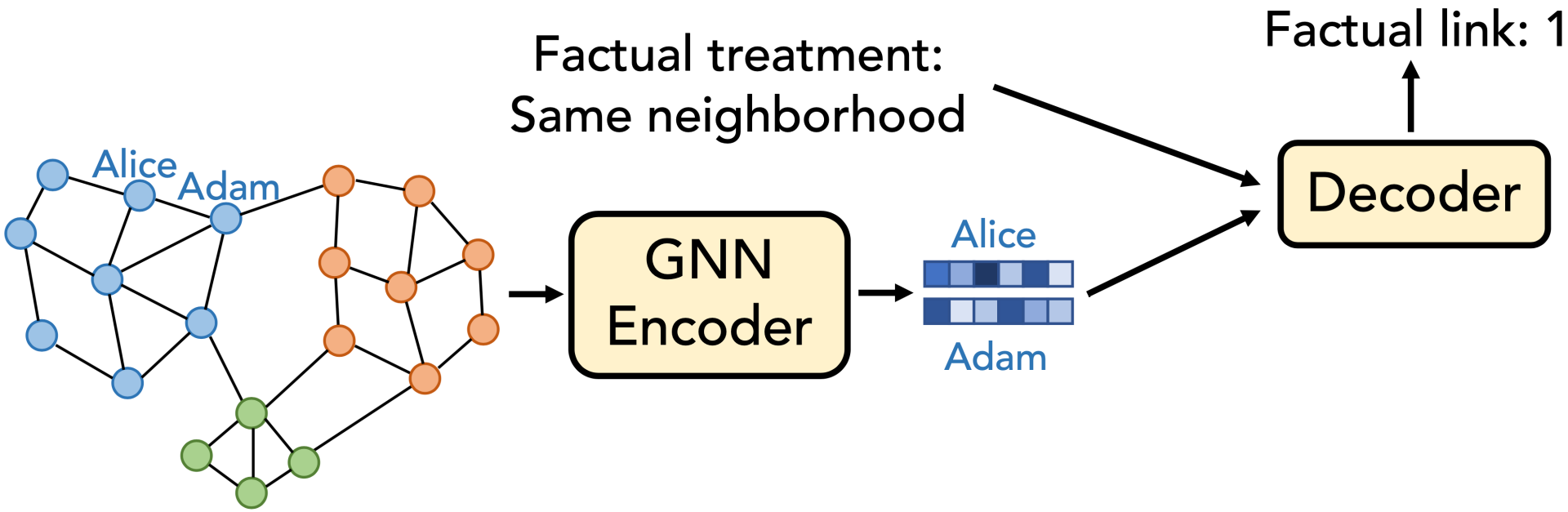
(Alice, Adam) —Most similar / Different community→ (Helen, Bob)

Factual link: 1

Counterfactual link: 1

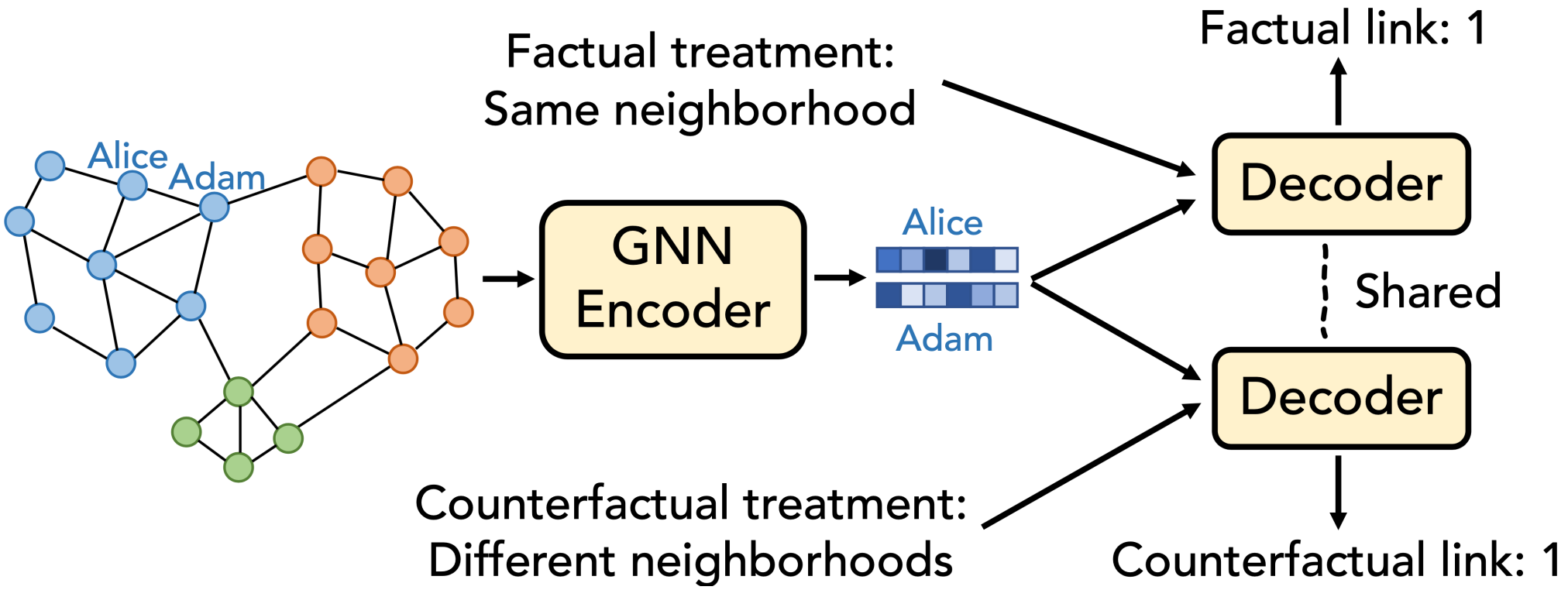
Zhao, et al. Learning from Counterfactual Links for Link Prediction. ICML 2022.

Learning from Counterfactual Links



Zhao, et al. Learning from Counterfactual Links for Link Prediction. ICML 2022.

Learning from Counterfactual Links



Zhao, et al. Learning from Counterfactual Links for Link Prediction. ICML 2022.