$\mathcal{G}$-Mixup: Graph Data Augmentation for Graph Classification

Xiaotian Han[1], Zhimeng Jiang[1], Ninghao Liu[2], Xia Hu[3]

[1]Texas A&M University, [2]University of Georgia, [3]Rice University

Overview

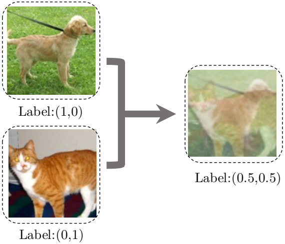
1. Background and Motivation
2. Methodology
   - $\mathcal{G}$-Mixup
   - Implementation
3. Experiments
   - Verification Experiments
   - Performance Experiments

Overview

1. Background and Motivation
2. Methodology
   - $\mathcal{G}$-Mixup
   - Implementation
3. Experiments
   - Verification Experiments
   - Performance Experiments

Mixup

Mixup is a cross-instance data augmentation method, which linearly interpolates random sample pair to generate more synthetic training data.



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

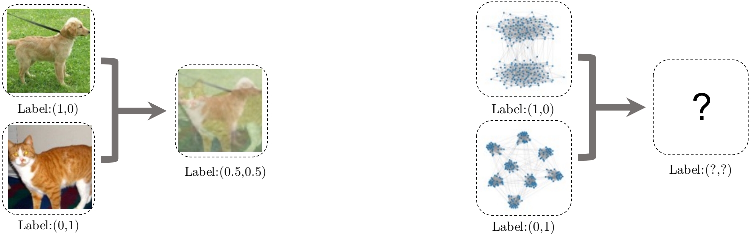
where $(\mathbf{x}_i, \mathbf{y}_i)$, $(\mathbf{x}_j, \mathbf{y}_j)$ are two samples randomly drawn from training data.

Mixup have been empirically and theoretically shown to improve the generalization and robustness of deep neural networks (H. Zhang et al., 2017; L. Zhang et al., 2021).

Can we mix up input graph pair to improve graph neural networks?

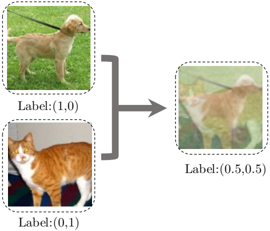
Challenges for Graph Mixup

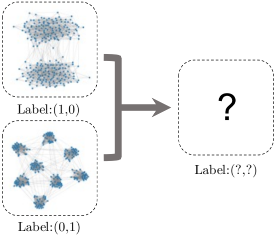
Graph data is different from image data:



Challenges for Graph Mixup

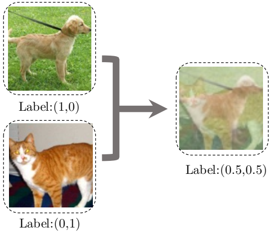
Graph data is different from image data:

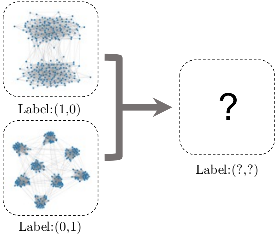




1. Image data is regular (image can be represented as matrix)
1. Graph data is irregular (the number of nodes)

Graph data is different from image data:



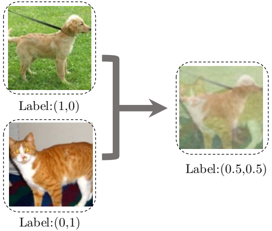


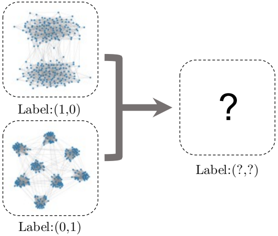
1. Image data is regular (image can be represented as matrix)
2. Image data is well-aligned (pixel to pixel correspondence)

1. Graph data is irregular (the number of nodes)
2. Graph data is not well-aligned (nodes not naturally ordered)

Challenges for Graph Mixup

Graph data is different from image data:



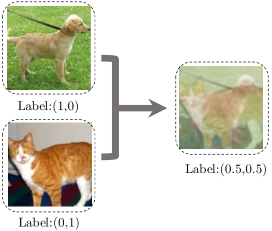


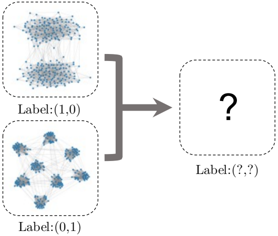
1. Image data is regular (image can be represented as matrix)
2. Image data is well-aligned (pixel to pixel correspondence)
3. Image data is grid-like data

1. Graph data is irregular (the number of nodes)
2. Graph data is not well-aligned (nodes not naturally ordered)
3. Graph has divergent topology information

Challenges for Graph Mixup

Graph data is different from image data:





1. Image data is regular (image can be represented as matrix)
2. Image data is well-aligned (pixel to pixel correspondence)
3. Image data is grid-like data

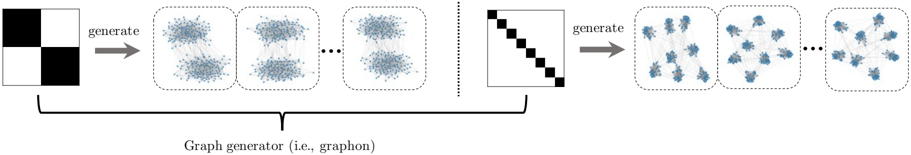
- Image is in Euclidean space

1. Graph data is irregular (the number of nodes)
2. Graph data is not well-aligned (nodes not naturally ordered)
3. Graph has divergent topology information

- Graph is in non-Euclidean space

Graph Generator: Graphon

The real-world graphs can be regarded as generated from generator (i.e., graphon[1]). For example,



The graphons of different graphs are **regular**, **well-aligned**, and **in Euclidean space**.

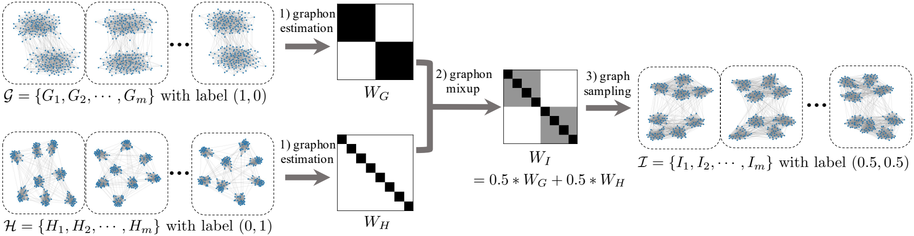
We propose to mix up graph generator (i.e., graphon) to achieve the input graph mixup.

[1] For ease of exposition, we use step function as grpahon in the following.

Overview

1. Background and Motivation
2. Methodology
   - $\mathcal{G}$-Mixup
   - Implementation
3. Experiments
   - Verification Experiments
   - Performance Experiments

We propose to mixup the generator (i.e., graphon) of graphs, mix up the graphons of different classes, and then generate synthetic graphs.



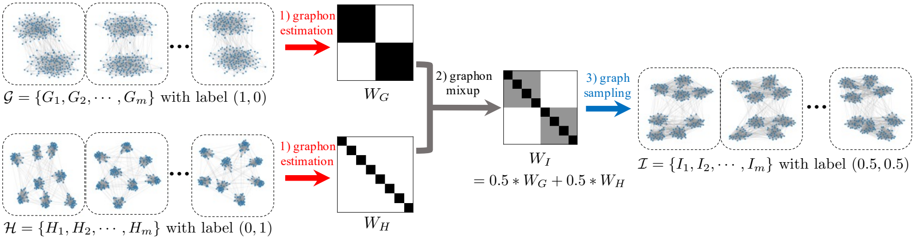
The formal mathematical expression are as follows:

(1) Graphon Estimation: $\mathcal{G} \rightarrow W_{\mathcal{G}}, \mathcal{H} \rightarrow W_{\mathcal{H}}$

(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: $\mathbf{y}_{\mathcal{I}} = \lambda \mathbf{y}_{\mathcal{G}} + (1 - \lambda) \mathbf{y}_{\mathcal{H}}$



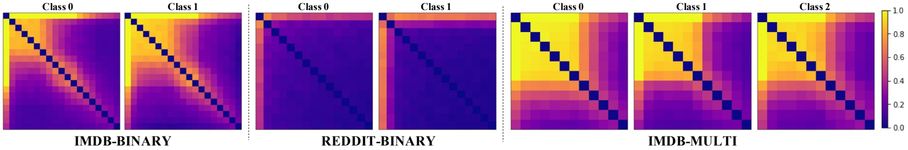
1. **Graphon Estimation.** We use the step function (Lovász, 2012; Xu et al., 2021) to approximate graphons. In general, the *step function* can be seen as a matrix $\mathbf{W} = [w_{kk'}] \in [0,1]^{K \times K}$, where $\mathbf{W}_{ij}$ is the probability that an edge exists between node i and node j.
2. **Synthetic Graphs Generation.** Generates an adjacency matrix $\boldsymbol{A} = [a_{ij}] \in \{0,1\}^{K \times K}$, whose element values follow the Bernoulli distributions $(\cdot)$ determined by the step function.

Overview

1. Background and Motivation
2. Methodology
   - $\mathcal{G}$-Mixup
   - Implementation
3. Experiments
   - Verification Experiments
   - Performance Experiments

Do different classes of graphs have different graphons?

We visualize the estimated graphons on IMDB-BINARY, REDDIT-BINARY, and IMDB-MULTI.

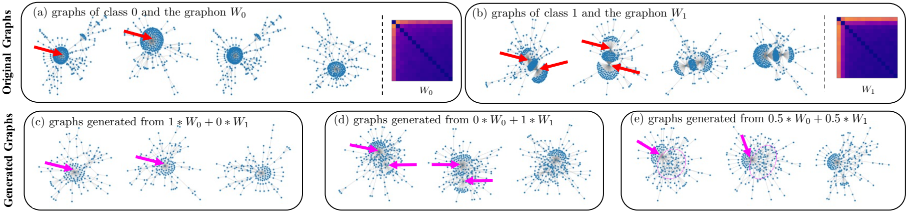


We make the following observations:

1. Real-world graphs of different classes have different graphons.
2. This observation lays a solid foundation for our proposed method.

What is $\mathcal{G}$-Mixup doing? A case study

We visualize the generated synthetic graphs on REDDIT-BINARY dataset.



We make the following observations:

1. The class 0 has one high-degree node while class 1 have two (a)(b).
2. The generated graphs based on
   - $(1 * W_0 + 0 * W_1)$ have one high-degree node (c).
   - $(0 * W_0 + 1 * W_1)$ have two high-degree nodes (d).
   - $(0.5 * W_0 + 0.5 * W_1)$ have a high-degree node and a dense subgraph (e).
3. Graphs generated by $\mathcal{G}$-Mixup are the mixture of original graphs.

Can $\mathcal{G}$-Mixup improve the performance of GNNs?

We use different GNNs for graph classification and report the performance comparisons of $\mathcal{G}$-Mixup.

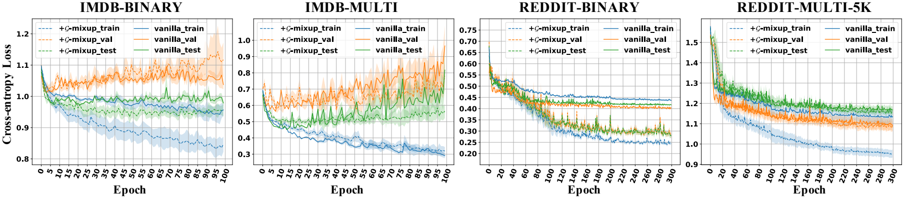
	Dataset	IMDB-B	IMDB-M	REDD-B	REDD-M5	REDD-M12
	#graphs	1000	1500	2000	4999	11929
	#classes	2	3	2	5	11
	#avg.nodes	19.77	13.00	429.63	508.52	391.41
	#avg.edges	96.53	65.94	497.75	594.87	456.89
GCN	vanilla	72.18	48.79	78.82	45.07	46.90
	w/ Dropedge	72.50	49.08	81.25	51.35	47.08
	w/ DropNode	72.00	48.58	79.25	49.35	47.93
	w/ Subgraph	68.50	49.58	74.33	48.70	47.49
	w/ M-Mixup	72.83	49.50	75.75	49.82	46.92
	w/ $\mathcal{G}$-Mixup	**72.87**	**51.30**	**89.81**	**51.51**	**48.06**
GIN	vanilla	71.55	48.83	92.59	55.19	50.23
	w/ Dropedge	**72.20**	48.83	92.00	55.10	49.77
	w/ DropNode	72.16	48.33	90.25	53.26	49.95
	w/ Subgraph	68.50	47.25	90.33	54.60	49.67
	w/ M-Mixup	70.83	49.88	90.75	54.95	49.81
	w/ $\mathcal{G}$-Mixup	71.94	**50.46**	**92.90**	**55.49**	**50.50**

	Method	IMDB-B	IMDB-M	REDD-B	REDD-M5k
TopKPool	vanilla	72.37	50.57	90.30	45.07
	w/ Dropedge	71.75	48.75	88.96	**47.43**
	w/ DropNode	69.16	48.50	81.33	46.15
	w/ Subgraph	67.83	50.83	86.08	45.75
	w/ M-Mixup	71.83	51.22	87.58	45.60
	w/ $\mathcal{G}$-Mixup	**72.80**	**51.30**	**90.40**	46.48
DiffPool	vanilla	71.68	47.75	78.40	31.61
	w/ Dropedge	69.16	49.44	76.00	34.46
	w/ DropNode	70.25	46.83	76.68	33.10
	w/ Subgraph	69.50	46.00	76.06	31.65
	w/ M-Mixup	66.50	45.16	78.37	34.46
	w/ $\mathcal{G}$-Mixup	**73.25**	**50.70**	**78.87**	**38.42**
MincutPool	vanilla	73.25	49.04	84.95	49.32
	w/ Dropedge	69.16	49.66	81.37	47.20
	w/ DropNode	73.50	49.91	85.68	46.82
	w/ Subgraph	70.25	48.18	84.91	49.22
	w/ M-Mixup	70.62	49.96	85.12	47.20
	w/ $\mathcal{G}$-Mixup	**73.93**	**50.29**	**85.87**	**50.12**

We make the following observation:

1. $\mathcal{G}$-Mixup can improve the performance of GNNs on various datasets.

We present the training/validation/test curves on IMDB-BINARY, IMDB-MULTI, REDDIT-BINARY and REDDIT-MULTI-5K with GCN.



We make the following observations:

1. The loss curves of $\mathcal{G}$-Mixup are lower than the vanilla model.
2. $\mathcal{G}$-Mixup can improve the generalization of graph neural networks.

References I

Lovász, L. (2012). *Large networks and graph limits* (Vol. 60). American Mathematical Soc.

Xu, H., Luo, D., Carin, L., & Zha, H. (2021). Learning graphons via structured gromov-wasserstein barycenters. *Proceedings of the AAAI Conference on Artificial Intelligence*, 10505–10513.

Zhang, H., Cisse, M., Dauphin, Y. N., & Lopez-Paz, D. (2017). Mixup: Beyond empirical risk minimization. *International Conference on Learning Representations*.

Zhang, L., Deng, Z., Kawaguchi, K., Ghorbani, A., & Zou, J. (2021). How does mixup help with robustness and generalization? *International Conference on Learning Representations*.

$\mathcal{G}$-Mixup: Graph Data Augmentation for Graph Classification

Xiaotian Han[1], Zhimeng Jiang[1], Ninghao Liu[2], Xia Hu[3]
[1]Texas A&M University, [2]University of Georgia, [3]Rice University

Q&A