Model-Agnostic Augmentation for Accurate Graph Classification

Jaemin Yoo*, **Sooyeon Shim**, and **U Kang**

Seoul National University

* Currently in Carnegie Mellon University

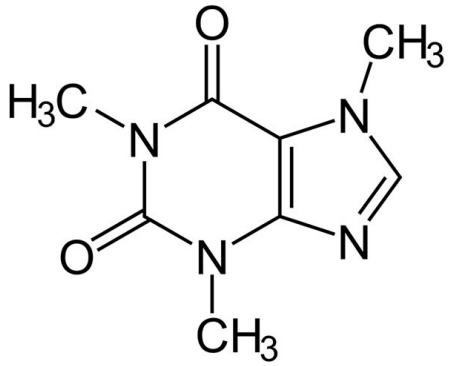
TheWebConf 2022

Outline

- **Introduction**
- Proposed Methods
- Experiments
- Conclusion

Graphs

- Graph data are common in real-world tasks
  - **Social network** represents friendships of people
  - **Review graph** represents user preferences
  - **Chemical compound** consists of many elements



https://www.shortstack.com/blog/best-social-networks-to-reach-specific-demographics/

Graph Classification

- **Given** a set $\{G_i, y_i\}_i$ of graphs and labels
  - Each graph has a node attribute matrix $\mathbf{X}$
- **Learn** a graph classifier f
- **Predict** the labels of unseen test graphs
- **Real-world applications:**
  - Predicting the toxicity of a chemical compound
  - Predicting the property of a social group

Data Augmentation

- **Data augmentation (DA)** is essential for ML
  - Increases the coverage of training data
  - Improves the generalizability of estimators
- An example of DA in the image domain:

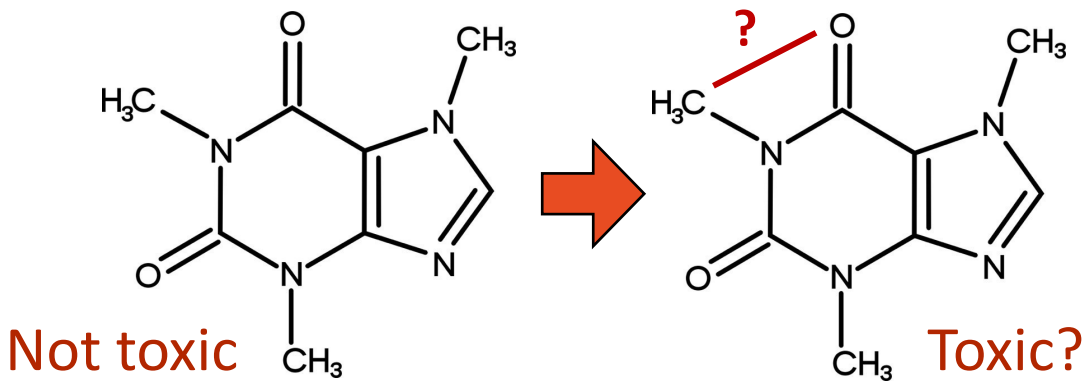
A Simple Framework for Contrastive Learning of Visual Representations (ICML 2020)

Graph Augmentation

- DA can also be done for graph classification
  - Recall that images are grid-structured graphs
- Consider an undirected graph $G = (\mathcal{V}, \mathcal{E}, \mathbf{X})$
  - $\mathcal{V}$ is the set of nodes
  - $\mathcal{E}$ is the set of edges
  - $\mathbf{X}$ is the node feature matrix of size $|\mathcal{V}| \times d$
- Any of $\mathcal{V}$, $\mathcal{E}$, and $\mathbf{X}$ can be the target of DA

Difficulties of Graph DA

- DA should preserve **semantic information**
  - In images, this can be easily verified visually
  - In graphs, even the change of a single edge can change the fundamental semantic information
- Let's consider a molecular graph:



Desired Properties (1)

- We have contradictory goals for effective DA
  - **Goal 1.** To make sufficient changes of a graph
  - **Goal 2.** To preserve semantic information
- We thus propose five **desired properties**
  - **Property 1.** Changes in $|\mathcal{E}|$ should be unbiased
  - **Property 2.** Connectivity should be preserved
  - **Property 3.** Node features should change
  - **Property 4.** The number of edges should change
  - **Property 5.** It should be done in linear time

Desired Properties (2)

- We have contradictory goals for effective DA
  - **Goal 1.** To make sufficient changes of a graph
  - **Goal 2.** To preserve semantic information
- We thus propose five **desired properties**
  - **Property 1.** Changes in $|\mathcal{E}|$ should be unbiased
  - **Property 2.** Connectivity should be preserved
  - **Property 3.** Node features should change
  - **Property 4.** The number of edges should change
  - **Property 5.** It should be done in linear time

Research Overview

- We propose two algorithms for effective DA
  - Our approaches satisfy all desired properties
  - They show the best performance in 9 datasets

Method	P1	P2	P3	P4	P5
DropEdge [28]				✓	✓
GraphCrop [35]			✓	✓	✓
NodeAug [38]			✓	✓	✓
MotifSwap [55]	✓	✓			
NodeSam (proposed)	✓	✓	✓	✓	✓
SubMix (proposed)	✓	✓	✓	✓	✓

Outline

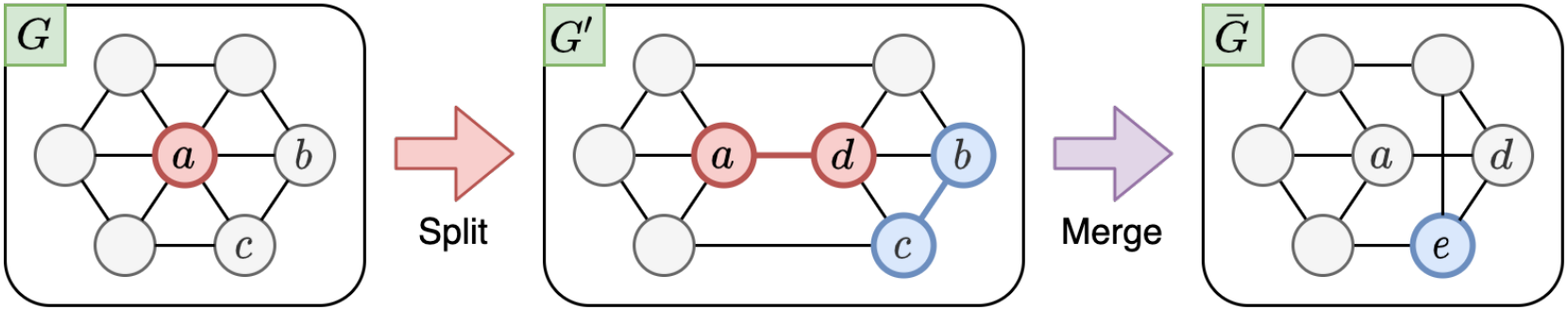
- Introduction
- **Proposed Methods**
- Experiments
- Conclusion

Proposed Methods

- We propose two algorithms for graph DA
  - They work in a **model-agnostic** way
  - Based on different motivations and ideas
- **NodeSam** (Node Split & Merge)
  - Makes balanced and stable changes
- **SubMix** (Subgraph Mix)
  - Makes a high degree of structural changes

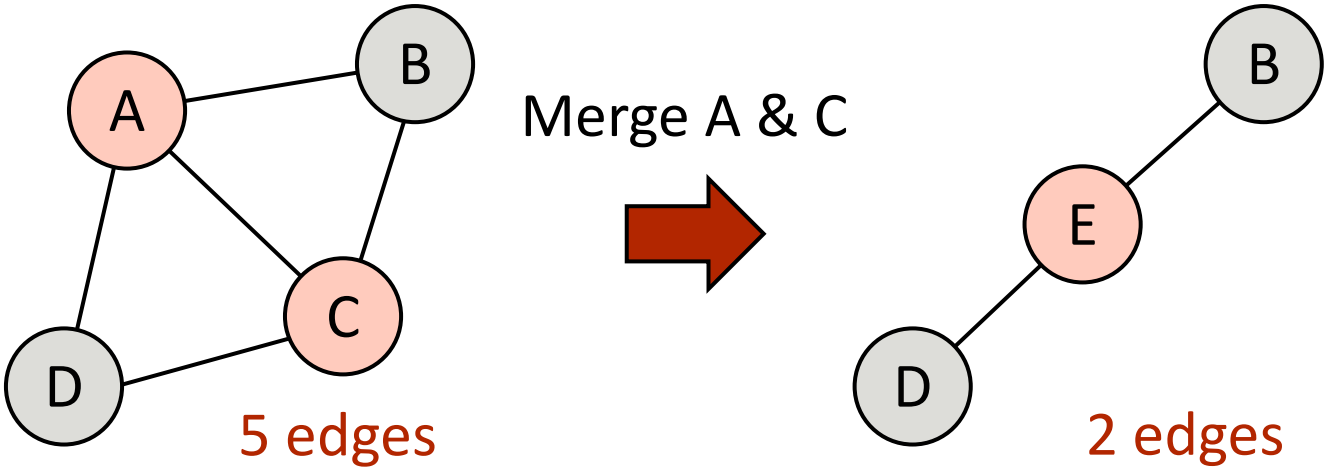
NodeSam: Overview

- **Goal:** To make balanced and stable changes
- Main idea is to conduct opposite operations
  - To **split** a random node into a pair of nodes
  - To **merge** a random pair of connected nodes
- This does not arbitrarily change connectivity



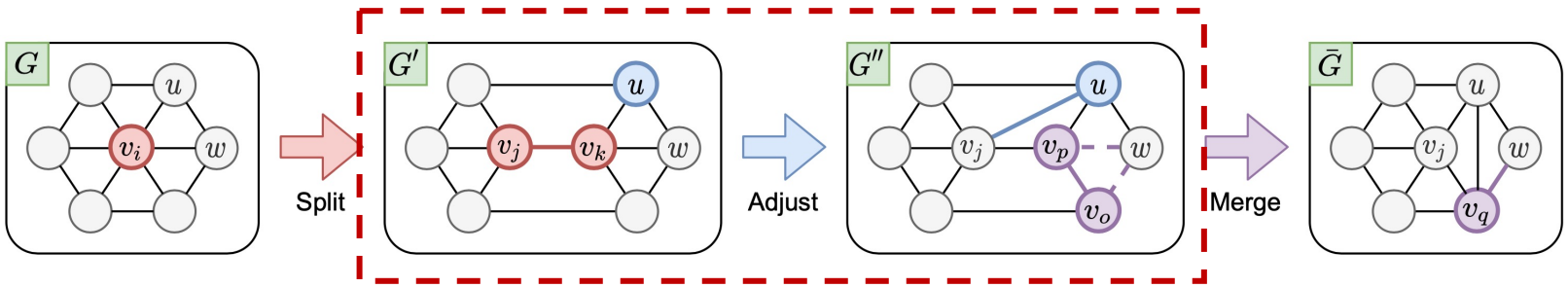
NodeSam: Limitation

- The basic version can **decrease** # of edges
  - Split always creates a single edge
  - Merge can remove **more than one** edge
- This violates our Property 1 for unbiasedness



NodeSam: Adjustment

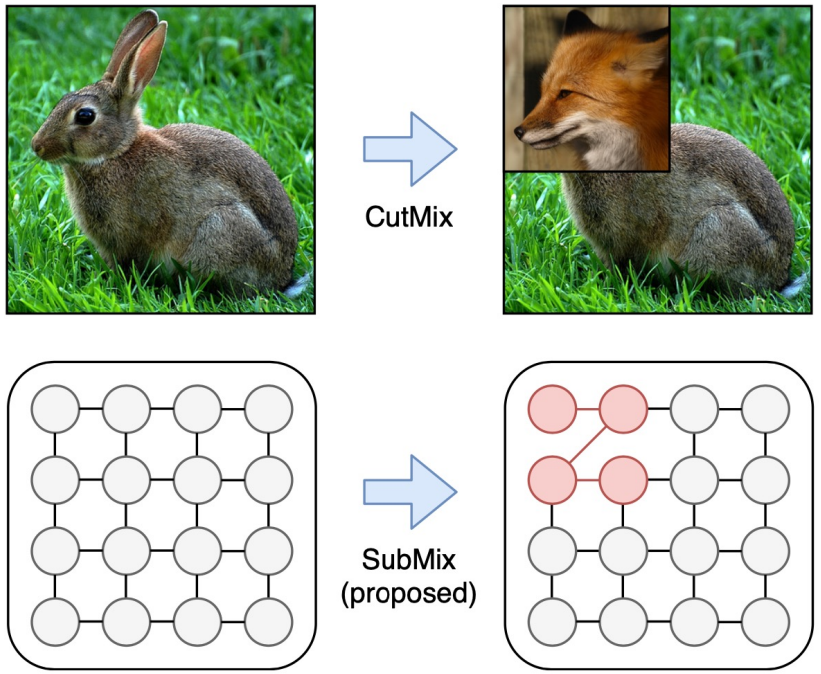
- We propose an **adjustment** operation
  - **Step 1.** Compute h_i, which is the expected number of edges that will be removed by Merge
  - **Step 2.** Insert h_i edges around the target node
- We also propose an optimization technique
  - Compute an estimation of h_i in linear time



SubMix: Overview

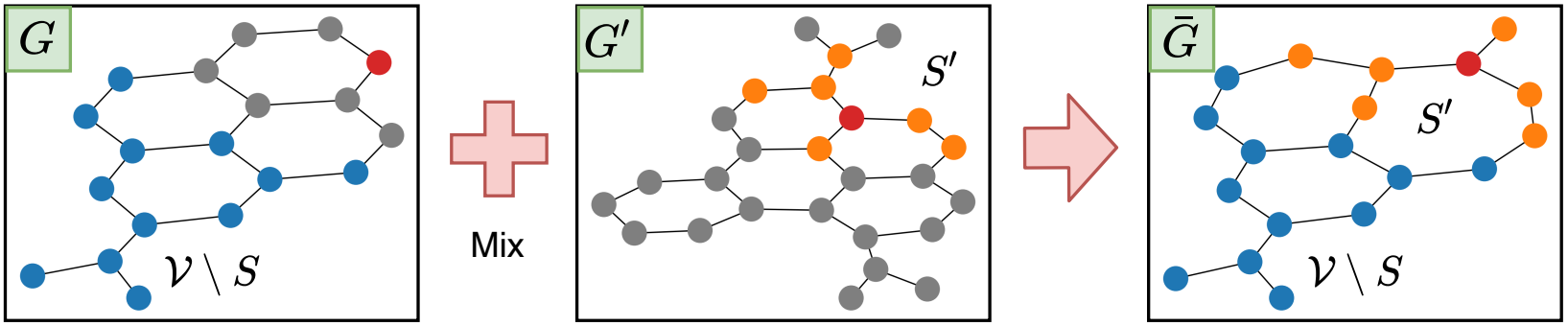
- **Goal:** To augment a graph in a subgraph level
  - Recall that NodeSam is a node-level algorithm

- SubMix generalizes **CutMix** into graphs
  - CutMix is a popular algorithm in images
  - The red subgraph is from another graph



SubMix: Random Walks

- How SubMix works given a set $\mathcal{G}$ of graphs:
  - Take two random graphs G and G' from $\mathcal{G}$
  - Select subgraphs S and S' by **random walks**
    - Each of S and S' is guaranteed to be connected
  - Replace S in G with S' from G', making $\bar{G}$



SubMix: Mixing Labels

- SubMix can select graphs with different labels
  - For example, G is toxic, while G' is not toxic
- The resulting (soft) label $\bar{\mathbf{y}}$ is computed as

$$\bar{\mathbf{y}} = q\mathbf{y} + (1-q)\mathbf{y}'$$

  - $\mathbf{y}$ and $\mathbf{y}'$ are one-hot labels of G and G', resp.
  - q is the ratio of edges of G included in the new $\bar{G}$

Outline

- Introduction
- Proposed Methods
- **Experiments**
- Conclusion

Datasets

- We use 9 datasets for graph classification
  - 7 molecular graphs and 2 large social networks
  - Each dataset consists of 344 to 144,033 graphs

Dataset	Graphs	Nodes	Edges	Features	Labels
D&D[1]	1,178	334,925	843,046	89	2
ENZYMES[1]	600	19,580	37,282	3	6
MUTAG[1]	188	3,371	3,721	7	2
NCI1[1]	4,110	122,747	132,753	37	2
NCI109[1]	4,127	122,494	132,604	38	2
PROTEINS[1]	1,113	43,471	81,044	3	2
PTC-MR[1]	344	4,915	5,054	18	2
COLLAB[1]	5,000	372,474	12,286,079	369	3
Twitter[1]	144,033	580,768	717,558	1,323	2

Experimental Setup

- We use GIN [ICLR'19] as a graph classifier
- We include the following baselines:
  - **DropEdge** removes an edge uniformly at random
  - **DropNode** removes a node uniformly at random
  - **GraphCrop** crops and returns a random subgraph
  - **MotifSwap** changes of an open triangle
  - …

Classification Accuracy

Q1. Do NS&SM outperform the baselines?

A. They show the best performance in general

- NodeSam makes the highest avg. accuracy
- SubMix achieves the best average rank

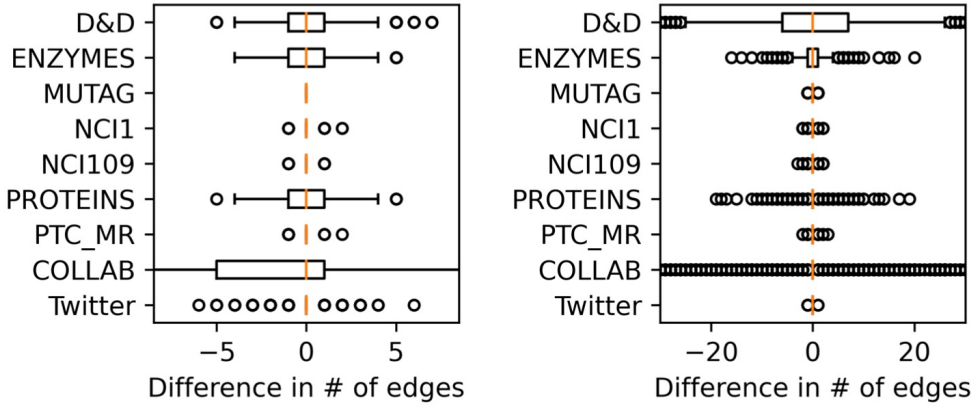
Method	D&D	ENZY.	MUTAG	NCI1	N109	PROT.	PTC-MR	COLLAB	Twitter	Average	Rank
Baseline	76.40 (4)	50.33 (10)	89.94 (4)	82.68 (9)	81.80 (9)	75.38 (9)	63.94 (7)	82.66 (7)	66.05 (7)	74.35 (8)	7.33 ± 2.18
GraphCrop	77.08 (2)	51.00 (9)	77.11 (10)	80.46 (10)	79.77 (10)	75.20 (10)	61.87 (10)	83.50 (2)	66.15 (3)	72.46 (10)	7.33 ± 3.77
DropEdge	76.14 (6)	53.67 (6)	81.93 (9)	82.82 (7)	82.60 (7)	75.74 (4)	63.68 (8)	82.50 (9)	66.05 (8)	73.90 (9)	7.11 ± 1.62
NodeAug	76.14 (8)	54.67 (5)	86.14 (7)	83.16 (4)	82.36 (8)	75.56 (6)	66.24 (2)	81.32 (10)	65.98 (9)	74.62 (7)	6.56 ± 2.55
AddEdge	76.14 (7)	55.17 (3)	85.67 (8)	83.99 (2)	83.06 (5)	75.38 (8)	64.27 (5)	82.80 (5)	66.10 (6)	74.73 (6)	5.44 ± 2.07
ChangeAttr	75.72 (9)	53.33 (8)	90.44 (3)	83.02 (5)	83.57 (2)	75.47 (7)	62.47 (9)	82.76 (6)	66.36 (2)	74.79 (5)	5.67 ± 2.83
DropNode	75.55 (10)	55.17 (3)	87.28 (6)	82.85 (6)	83.04 (6)	75.65 (5)	**66.59 (1)**	82.54 (8)	66.11 (5)	74.97 (4)	5.56 ± 2.60
MotifSwap	76.23 (5)	53.50 (7)	90.47 (2)	82.82 (7)	83.28 (4)	75.92 (3)	65.79 (3)	82.84 (3)	65.93 (10)	75.20 (3)	4.89 ± 2.62
SubMix	**78.10 (1)**	57.50 (2)	89.94 (4)	**84.33 (1)**	**84.37 (1)**	**76.19 (1)**	63.97 (6)	**83.74 (1)**	**66.44 (1)**	76.06 (2)	**2.00 ± 1.80**
NodeSam	76.57 (3)	**60.00 (1)**	**90.96 (1)**	83.33 (3)	83.52 (3)	76.10 (2)	65.48 (4)	82.82 (4)	66.13 (4)	**76.10 (1)**	2.78 ± 1.20

Changes of Graph Size

Q2. How do NS&SM change the graph size?

A. They make sufficient and unbiased changes

- We measure the graph size by # of edges
- The variance of changes is larger with SubMix

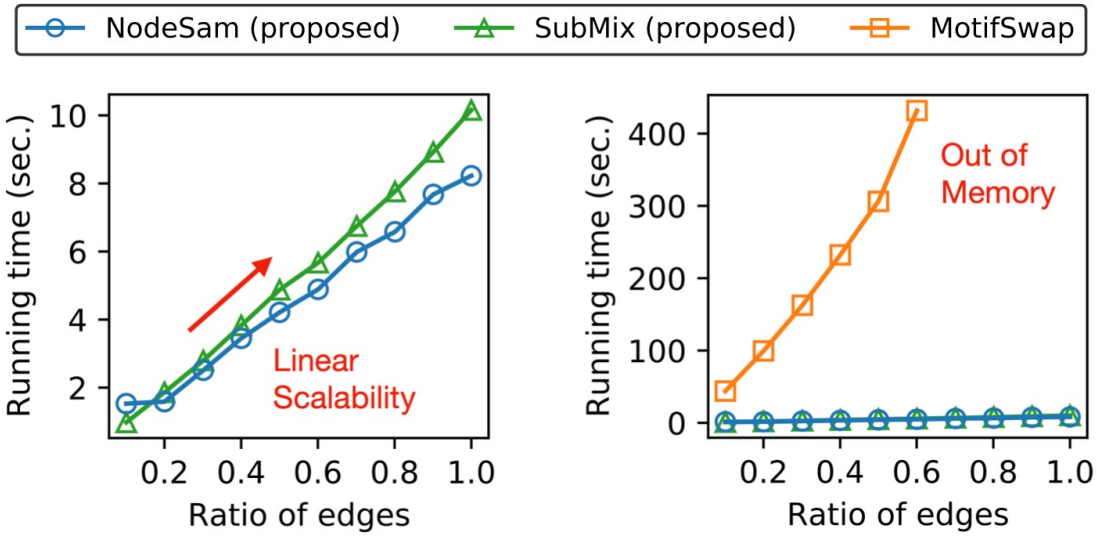


Scalability

Q3. How scalable are NS&SM to large graphs?

A. They show linear scalability with # of edges

- MotifSwap, the best competitor, causes OOM

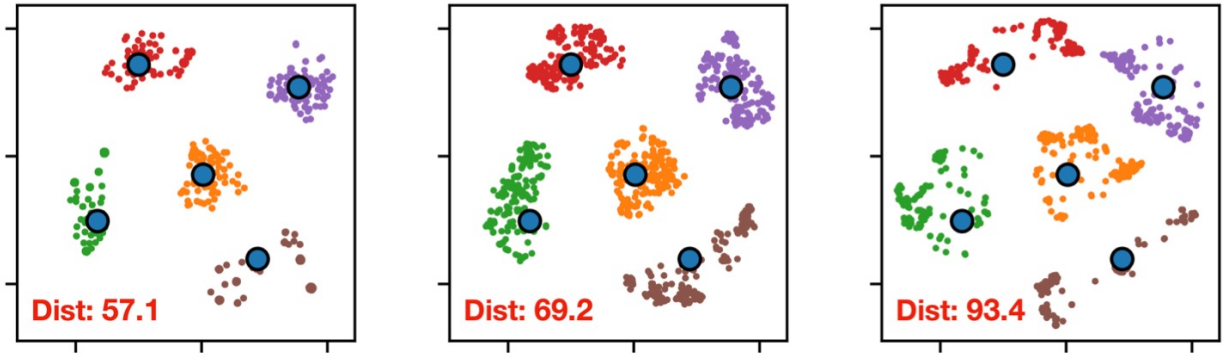


Visualization

Q4. How are augmented graphs distributed?

A. We visualize the space of augmentation

- MotifSwap makes the smallest changes
- SubMix makes the largest changes as we expect



(a) MotifSwap (b) NodeSam (c) SubMix

Outline

- Introduction
- Proposed Methods
- Experiments
- **Conclusion**

Conclusion

- We propose **NodeSam** and **SubMix**
  - Augmentation algorithms for graph classification
  - Can be used with any type of graph classifiers
- **Contributions**
  - First comprehensive work for model-agnostic DA
  - They satisfy five desired properties for graph DA
  - They show the best performance in 9 datasets

Thank you!

Jaemin Yoo (jaeminyoo@cmu.edu)

https://github.com/snudatalab/GraphAug