AugWard: Augmentation-Aware Representation Learning for Accurate Graph Classification

Minjun Kim[1], Jaehyeon Choi[1], SeungJoo Lee[1], Jinhong Jung[2*], and U Kang[1*]

[1]Seoul National University, [2]Soongsil University

*Correspond to: ukang@snu.ac.kr, jinhong@ssu.ac.kr

Overview

- We propose **AugWard**, an augmentation-aware learning framework for accurate graph classification
  - AugWard enriches graph representations by capturing "augmentation-induced differences"
- AugWard is easily integrated with any method, enhancing their accuracy across various settings
  - Supervised, semi-supervised, transfer learning

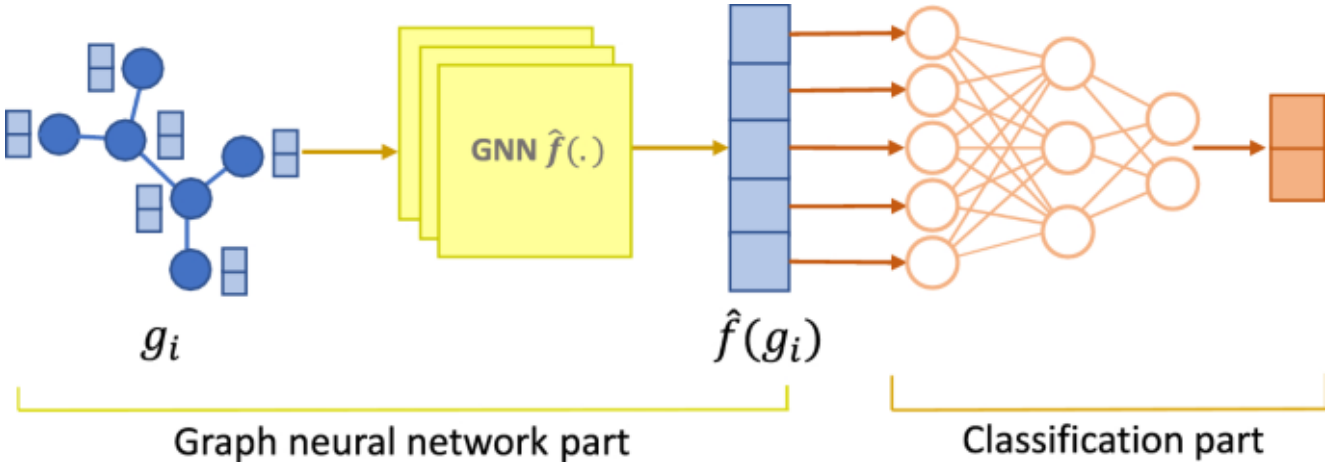
Outline

- **Introduction**
- Preliminaries
- Proposed Method
- Experiments
- Conclusion

Graph Classification

- **Task**: Classify a graph into pre-defined classes based on its structural properties and features

- Graph Neural Networks (GNNs) capture higher-order structures for accurate classification



[1] M. Do et al., (2023) "Two-Stage Training of Graph Neural Networks for Graph Classification" Neural Process Letters 55, pp. 2799–2823

Problem Definition

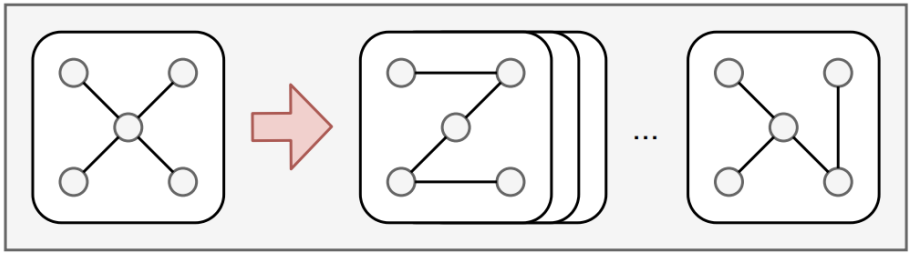
Graph Classification

- **Given**
  - A set $\boldsymbol{G} = \{\mathcal{G}_1, \cdots, \mathcal{G}_N\}$ of N distinct graphs
  - A set $\boldsymbol{Y} = \{y_1, \cdots, y_N\}$ of corresponding labels
  - A set $\boldsymbol{C}$ of classes

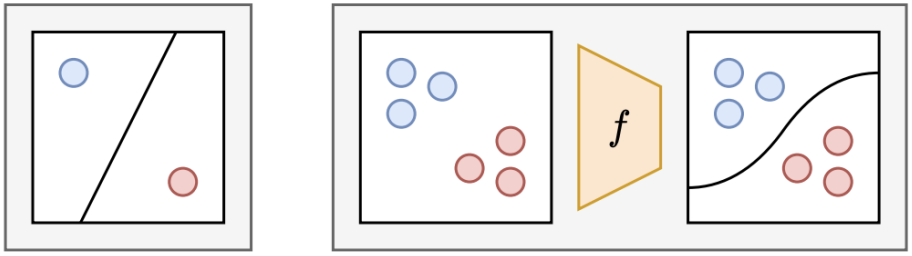
- **Predict**
  - $\forall \mathcal{G}_i \in \boldsymbol{G}, y \in \boldsymbol{C}$, the probability $P(y|\mathcal{G}_i)$

Graph Augmentation

- Graph augmentation generates variants of original graphs while preserving their labels
  - Mitigate the common issue of overfitting



Model-agnostic graph augmentation



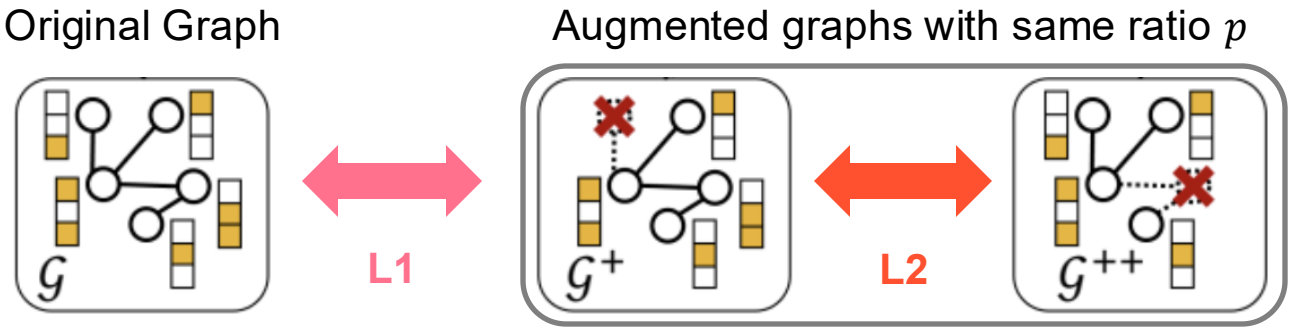
Baseline

Rich decision boundary learned by f

[1] J. Yoo et al., (2022) "Model-Agnostic Augmentation for Accurate Graph Classification" WWW 2022

Limitations of Previous Work

- Existing methods are suboptimal due to 2 major limitations by simplistic adaptation of augmentation:
  - **Limitation 1**. Ignorance of difference between original and augmented graphs
  - **Limitation 2**. Deceptive assumption that the perturbation ratio p ensures similarity among augmented graphs



Outline

- Introduction
- **Preliminaries**
- Proposed Method
- Experiments
- Conclusion

Graph Neural Networks

- Graph Neural Networks (GNNs) jointly train the encoder f_θ and the classifier g_ϕ for classification:

$$\mathbf{z}_\mathcal{G} = f_\theta(\mathcal{G}), \qquad \mathbf{p}_\mathcal{G} = g_\phi(\mathbf{z}_\mathcal{G})$$

  - $\mathbf{z}_\mathcal{G} \in \mathbb{R}^d$: representation of a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$
  - $\mathbf{p}_\mathcal{G} \in \mathbb{R}^{|\boldsymbol{C}|}$: predicted probabilities (i-th entry of $\mathbf{p}_\mathcal{G}$) $= P(y = i|\mathcal{G})$

$$\mathbf{z}_\mathcal{G} = \mathrm{READOUT}\left(\left\{\mathbf{h}_u^{(l)} \middle| \, u \in \mathcal{V}, l \in [L]\right\}\right),$$
$$\tilde{\mathbf{h}}_u^{(l)} = \mathrm{AGGREGATE}\left(\left\{\mathbf{h}_v^{(l-1)} : v \in \mathcal{N}_u\right\}\right), \mathbf{h}_u^{(l)} = \mathrm{COMBINE}\left(\mathbf{h}_u^{(l-1)}, \tilde{\mathbf{h}}_u^{(l)}\right).$$

  - $\mathbf{h}_u^{(0)} = \mathbf{X}_u$, $[L] = \{0, \cdots, L\}$, $\mathcal{N}_u$: set of neighbors for node u

Graph Augmentation

- Augmented graph $\mathcal{G}^+$ is randomly sampled as:

$$\mathcal{G}^+ \sim \mathcal{T}_p(\mathcal{G}^+|\mathcal{G})$$

  - $\mathcal{G}$: original graph
  - $\mathcal{T}_p(\cdot\,|\mathcal{G})$: augmentation distribution conditioned on $\mathcal{G}$
  - p: perturbation ratio (amount of change from $\mathcal{G}$)

- Various designs for $\mathcal{T}_p$
  - 1. Drop-based methods: remove or mask attributes of nodes and edges according to ratio p
  - 2. Mixup-based methods: fuse two graphs by ratio p

FGWD

Fused Gromov-Wasserstein Distance

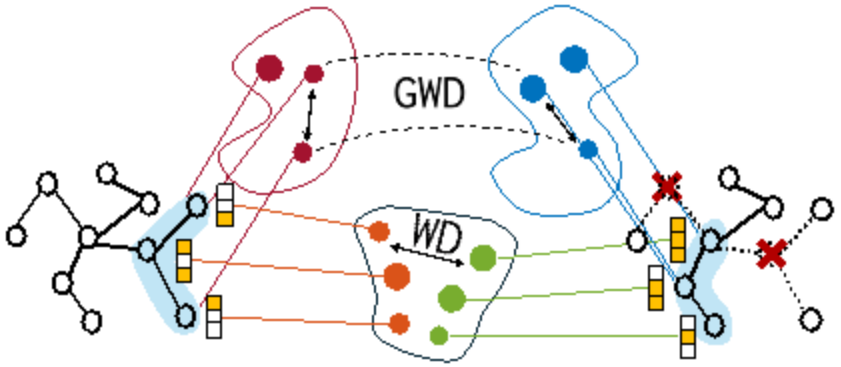
- FGWD measures the distance between two graphs $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$ and $\mathcal{G}^+ = (\mathcal{V}, \mathcal{E}, \mathbf{X}^+)$ by combining feature-level and structure-level differences

$$\mathrm{FGWD}_\alpha(\mathcal{G}, \mathcal{G}^+) = \min_{\boldsymbol{\pi} \in \Pi(\boldsymbol{\mu}, \boldsymbol{\nu})} \alpha \cdot WD(\mathbf{X}, \mathbf{X}^+, \boldsymbol{\pi}) + (1 - \alpha)\, GWD(\mathcal{E}, \mathcal{E}^+, \boldsymbol{\pi})$$

Wasserstein Distance (Feature-level distance)

Gromov-Wasserstein Distance (Structure-level distance)

- α: balancing hyperparameter
- $\boldsymbol{\pi}$: coupling matrix
- $\Pi(\boldsymbol{\mu}, \boldsymbol{\nu})$: a set of all possible matchings of distributions $\boldsymbol{\mu} \in \mathbb{R}^{|\mathcal{V}|}$ and $\boldsymbol{\nu} \in \mathbb{R}^{|\mathcal{V}^+|}$



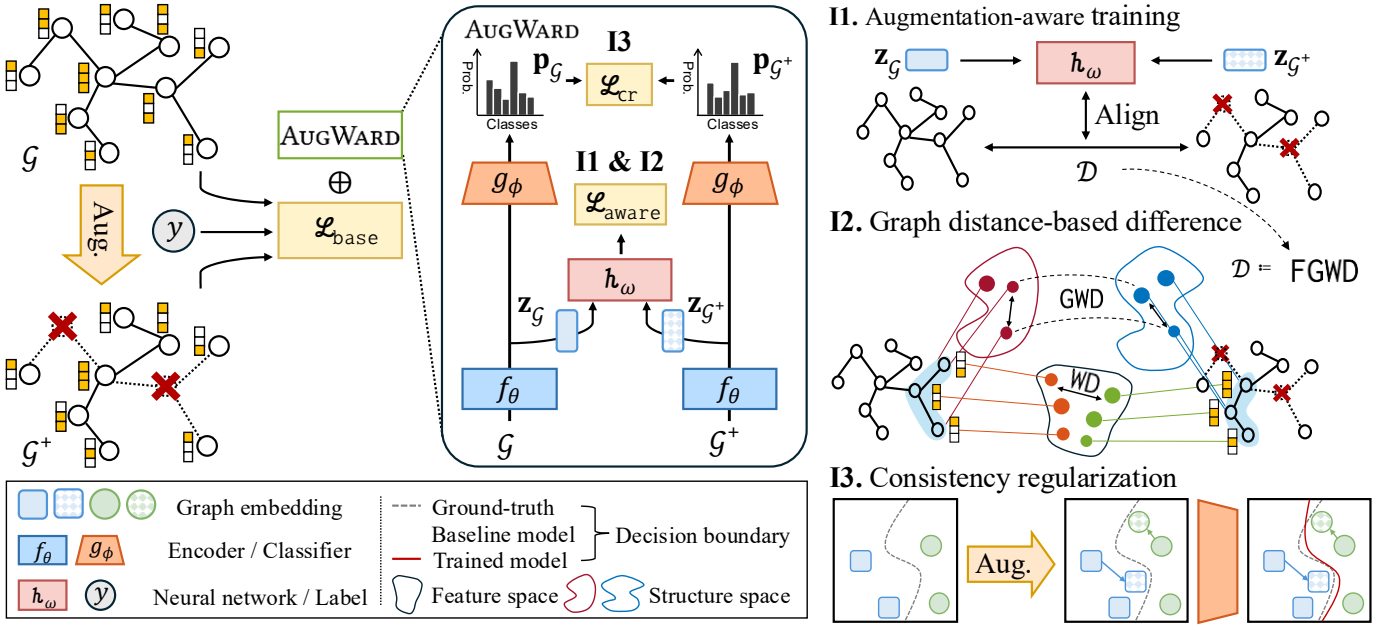
[1] T. Vayer et al., (2020). "Fused Gromov-Wasserstein distance for structured objects". Algorithms, 13(9), 212.

Outline

- Introduction
- Preliminaries
- **Proposed Method**
- Experiments
- Conclusion

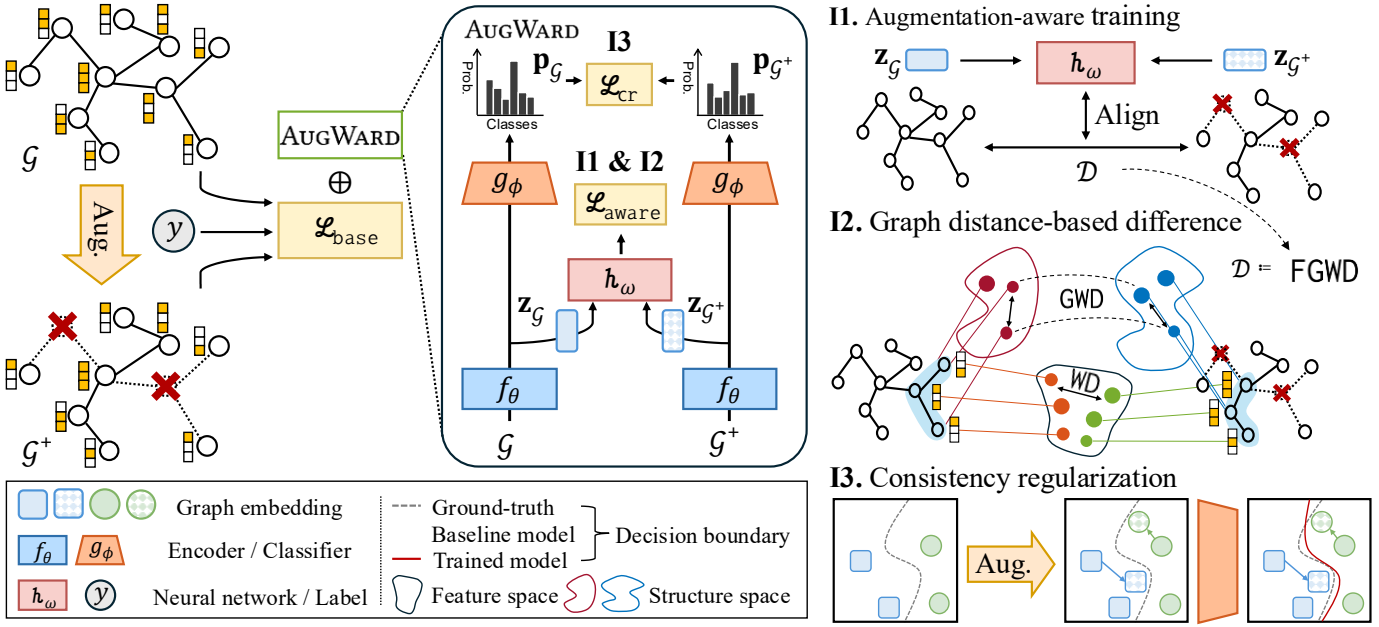
Overall Structure

- **AugWard** for accurate graph classification
  - Considers the *"augmentation-induced difference"*
  - Applicable to any augmentation $\mathcal{T}_p$ / encoder f_θ / classifier g_ϕ



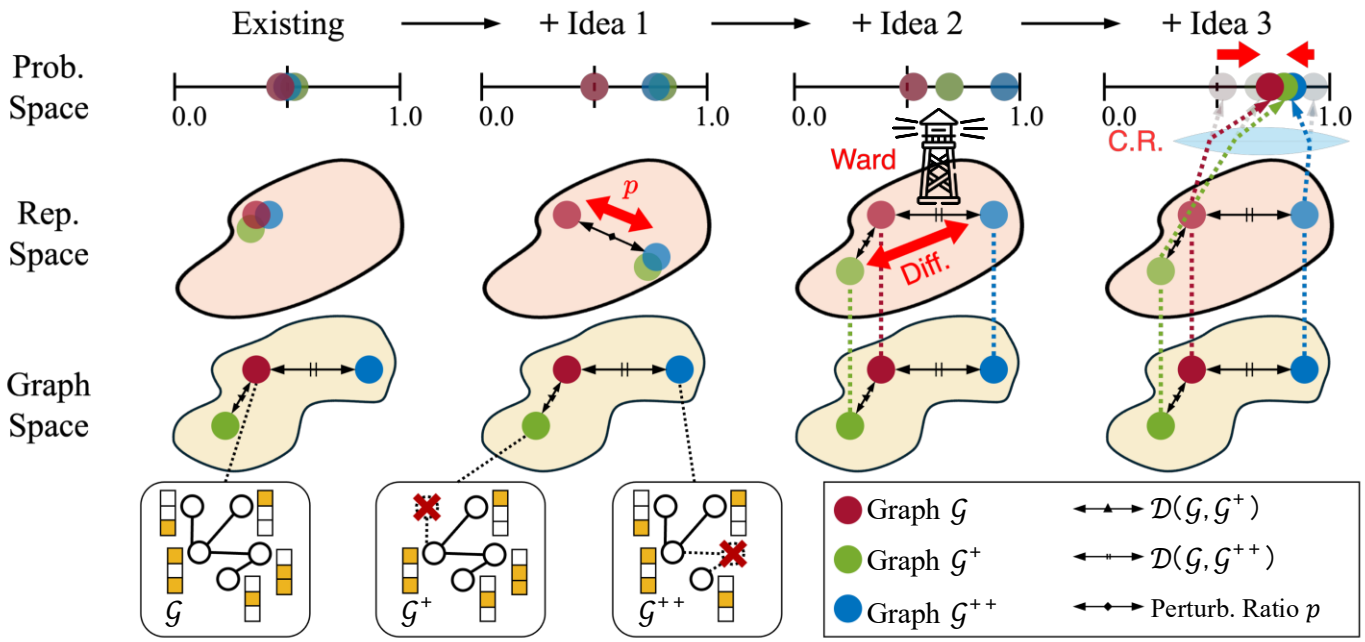
Overall Structure

- **AugWard** for accurate graph classification
  - **Idea 1.** Augmentation-aware training
  - **Idea 2.** Graph distance-based difference
  - **Idea 3.** Consistency regularization



Impact of Proposed Ideas

- The impact of proposed ideas
  - **Ideas 1 and 2** enrich the representation of each graph
  - **Idea 3** regulates the classifier for consistent predictions



Loss Function

- Combining all 3 ideas, we get the objective function
  - AugWard supports various learning paradigms with baseline loss $\mathcal{L}_{\text{base}}$:

Baseline loss (e.g., cross-entropy)

$$\mathcal{L}(\mathcal{G}, \mathcal{G}^+, y) = \mathcal{L}_{\text{base}}(\mathcal{G}, \mathcal{G}^+, y) + \mathcal{L}_{\text{AugWard}}(\mathcal{G}, \mathcal{G}^+),$$

$$\mathcal{L}_{\text{AugWard}}(\mathcal{G}, \mathcal{G}^+) = \lambda_{\text{aware}}\mathcal{L}_{\text{aware}}(\mathcal{G}, \mathcal{G}^+) + \lambda_{\text{cr}}\mathcal{L}_{\text{cr}}(\mathcal{G}, \mathcal{G}^+)$$

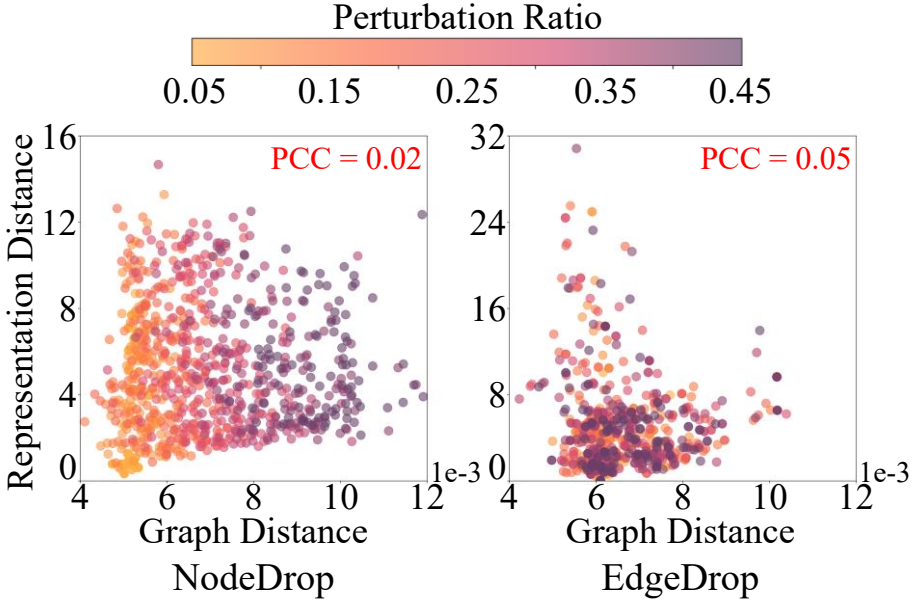
Idea 1. Augmentation-aware training
Idea 2. Graph distance-based difference

Idea 3. Consistency Regularization

Main Ideas

1. Augmentation-aware training

- **Challenge:** Capturing augmentation-induced difference
  - Existing methods ignore the difference between original and augmented graphs, or *augmentation-induced difference*
  - Pearson Correlation Coefficient (PCC) close to 0 indicates no strong correlation



** Euclidean $\left\|\mathbf{z}_{\mathcal{G}} - \mathbf{z}_{\mathcal{G}^+}\right\|_2^2$ and graph distances of $\mathcal{G}$ and 100 augmented graphs $\mathcal{G}^+$ per $p \in \{0.05, 0.1, \cdots, 0.45\}$

Main Ideas

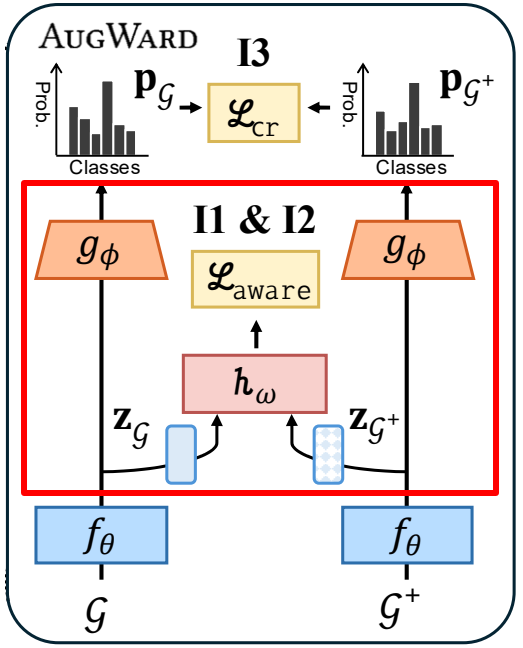
1. Augmentation-aware training

- **Idea:** Augmentation-aware training
  - **Intuition.** Encourage the encoder f_θ to align representation-level and graph-level differences
  - Train a neural network h_ω by optimizing $\mathcal{L}_{aware}$:

$$\mathcal{L}_{\text{aware}}(\mathcal{G}, \mathcal{G}^+) = \left\| h_\omega\left(\mathbf{z}_\mathcal{G}, \mathbf{z}_{\mathcal{G}^+}\right) - \mathcal{D}(\mathcal{G}, \mathcal{G}^+) \right\|^2$$

  - $\mathcal{D}(\cdot,\cdot)$: graph-level distance

"Which metric is suitable for $\mathcal{D}(\mathcal{G}, \mathcal{G}^+)$?"

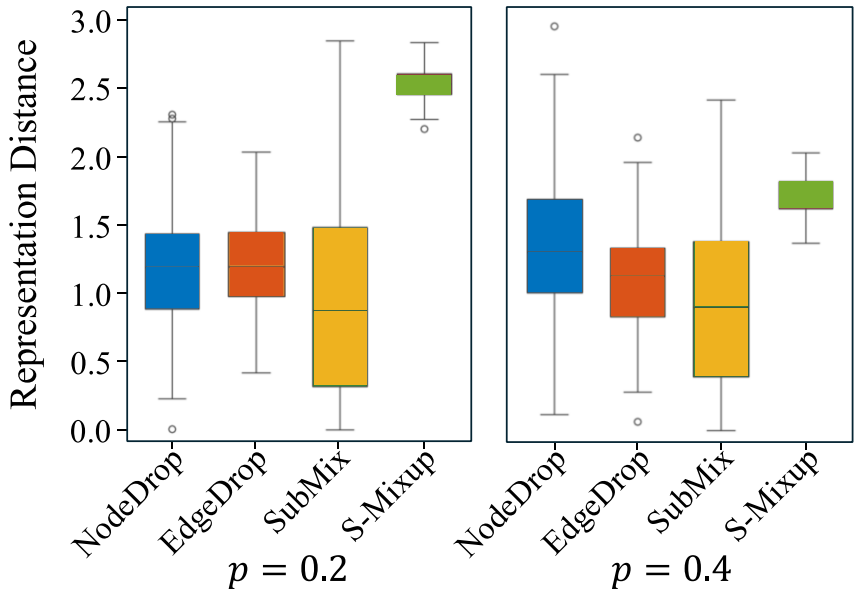


** A fully connected layer h_ω with the concatenation of $\mathbf{z}_\mathcal{G}$ and $\mathbf{z}_{\mathcal{G}^+}$ as input

Main Ideas

2. Graph Distance-based Difference

- **Challenge:** Measuring the difference gained from graph augmentation
  - Generated graphs at a fixed perturbation ratio p exhibit significant variations due to inherent randomness



** Euclidean $\left\|\mathbf{z}_{\mathcal{G}} - \mathbf{z}_{\mathcal{G}^+}\right\|_2^2$ distances between $\mathcal{G}$ and 100 augmented graphs $\mathcal{G}^+$ with p fixed at 0.2 and 0.4

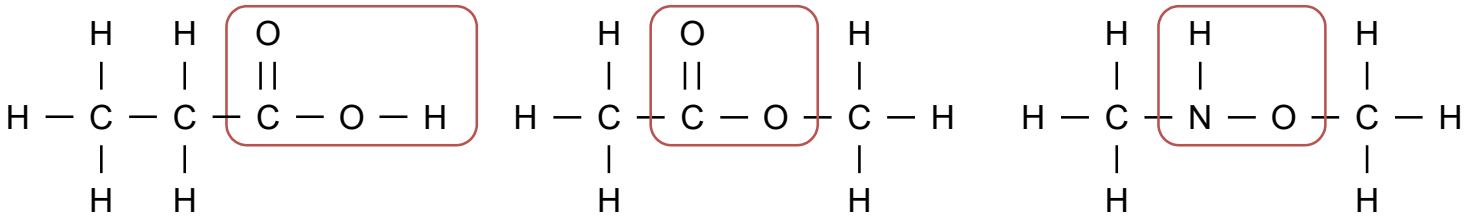
Main Ideas

2. Graph Distance-based Difference

- **Idea**: Graph Distance-based Difference
  - **Fused Gromov-Wasserstein Distance (FGWD)**: optimizes both differences in structures and features

$$\mathcal{L}_{\text{aware}} = \left\| h_\omega\left(\mathbf{z}_{\mathcal{G}}, \mathbf{z}_{\mathcal{G}^+}\right) - \text{FGWD}_\alpha(\mathcal{G}, \mathcal{G}^+) \right\|^2$$

  - Difference in either structure or feature leads to a significant distinction in their chemical type



(a) $C_3H_6O_2$ (Acid) (b) $C_3H_6O_2$ (Neutral) (c) C_2H_7NO (Base)

** (a) and (b) share graph features, while (b) and (c) exhibit identical graph structures

Main Ideas

3. Consistency Regularization

- **Motivation**: Training robust classifier
  - Given distinguishable representations, training the classifier robustly is crucial for better generalization
  - In node classification[1], matching predictions from different representations (same label) improves the generalization

- **Idea**: Consistency regularization
  - Matching two predictions $\mathbf{p}_{\mathcal{G}}$ and $\mathbf{p}_{\mathcal{G}^+}$

$$\mathcal{L}_{\text{cr}} = H\left(\mathbf{p}_{\mathcal{G}}, \mathbf{p}_{\mathcal{G}^+}\right) = -\sum_{i=1}^{|C|} P(y = i|\mathcal{G}) \cdot \log P(y = i|\mathcal{G}^+)$$

  - $H(\cdot,\cdot)$: cross-entropy loss

[1] W. Feng et al., (2020) "Graph Random Neural Network for Semi-Supervised Learning on Graphs" NeurIPS 2020

Outline

- Introduction
- Preliminaries
- Proposed Method
- **Experiments**
- Conclusion

Experimental Setup

- **Datasets**
  - Supervised, semi-supervised learning: 10 TUDatasets
  - Transfer learning: ZINC15 ➔ 8 MoleculeNet datasets
- **Augmentations**
  - Drop-based: NodeDrop, EdgeDrop, AttrMask, Subgraph, GraphAug
  - Mixup-based: SubMix, S-Mixup
- **Baselines**
  - Model: a 4-layered GIN
  - Semi-supervised (10%): Infograph, GraphCL, CuCo, GCL-SPAN
  - Transfer learning: ContextPred, GraphCL, MGSSL, GraphMAE

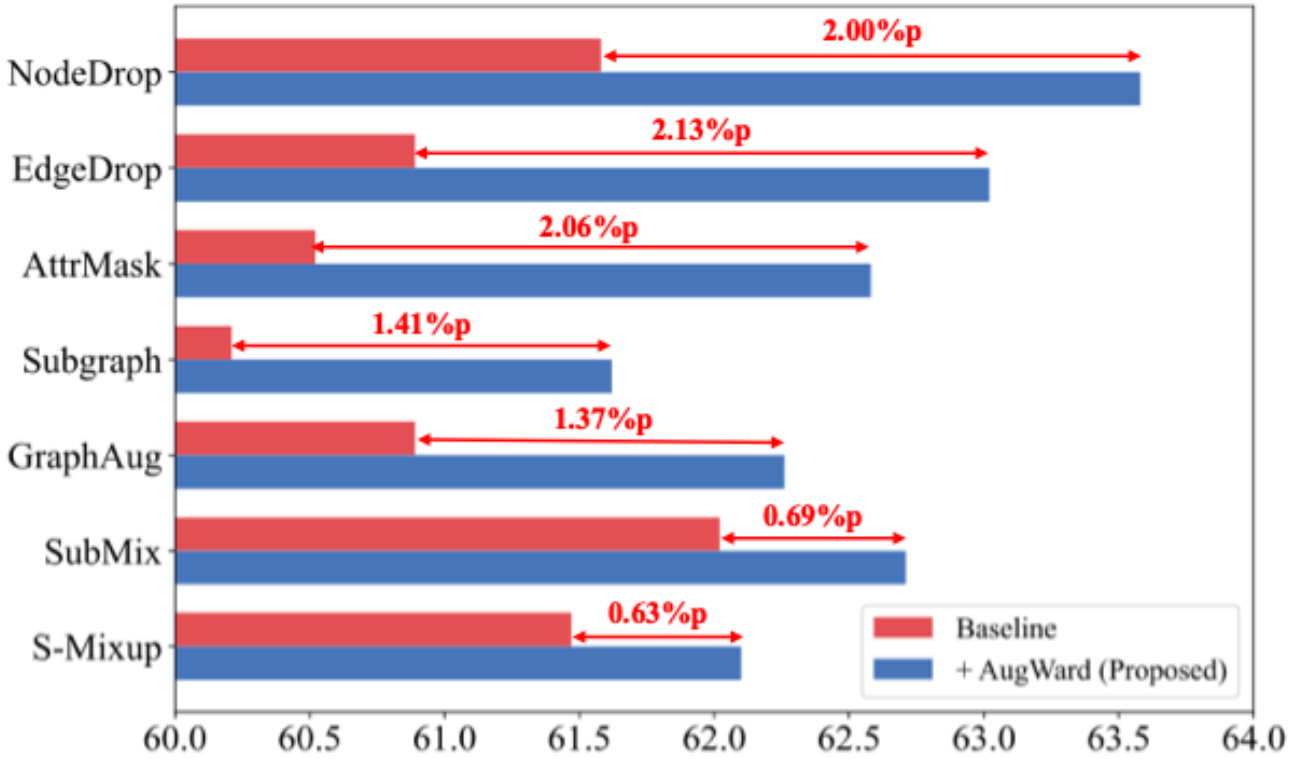
Experimental Questions

- We perform experiments on the following questions:
  - Q1. Accuracy in supervised graph classification
  - Q2. Accuracy in semi-supervised graph classification
  - Q3. Representation transferability
  - Q4. Runtime analysis
  - Q5. Ablation study

Experiments

1. Accuracy in supervised graph classification

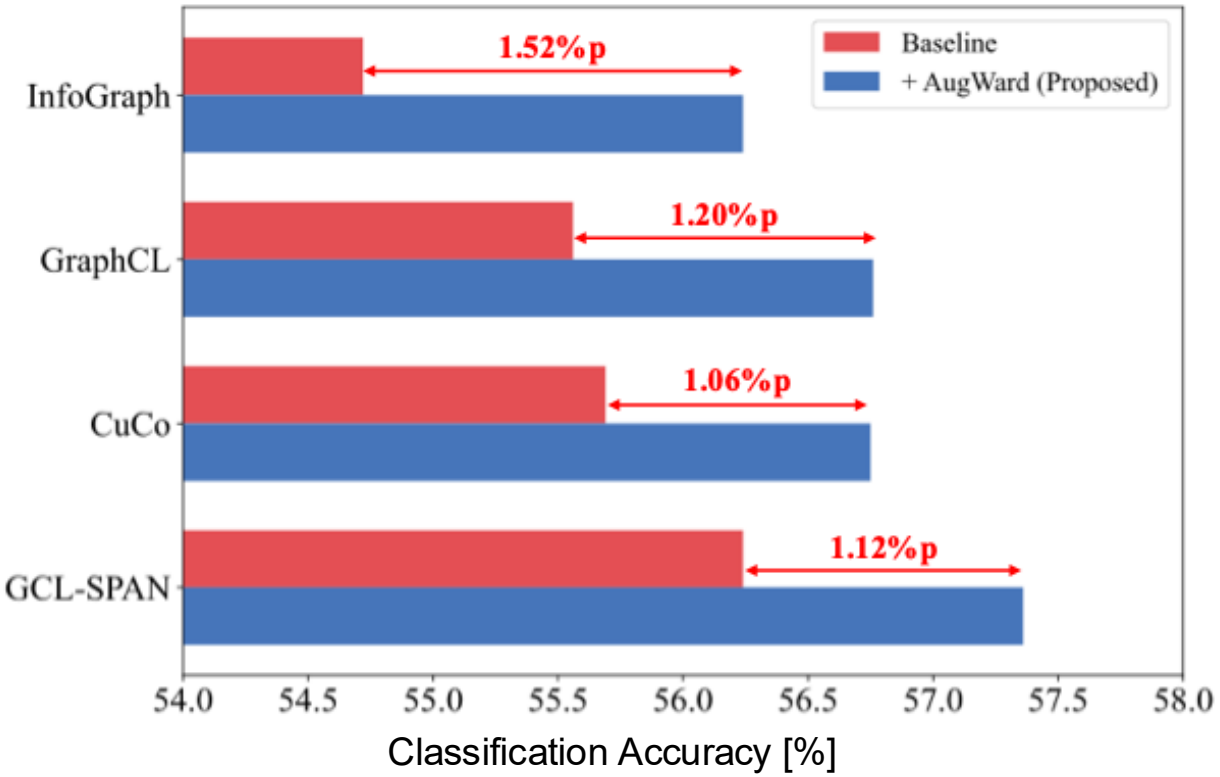
- AugWard consistently enhances classification accuracy
  - Up to **2.13%p** in average accuracy



Experiments

2. Accuracy in semi-supervised graph classification

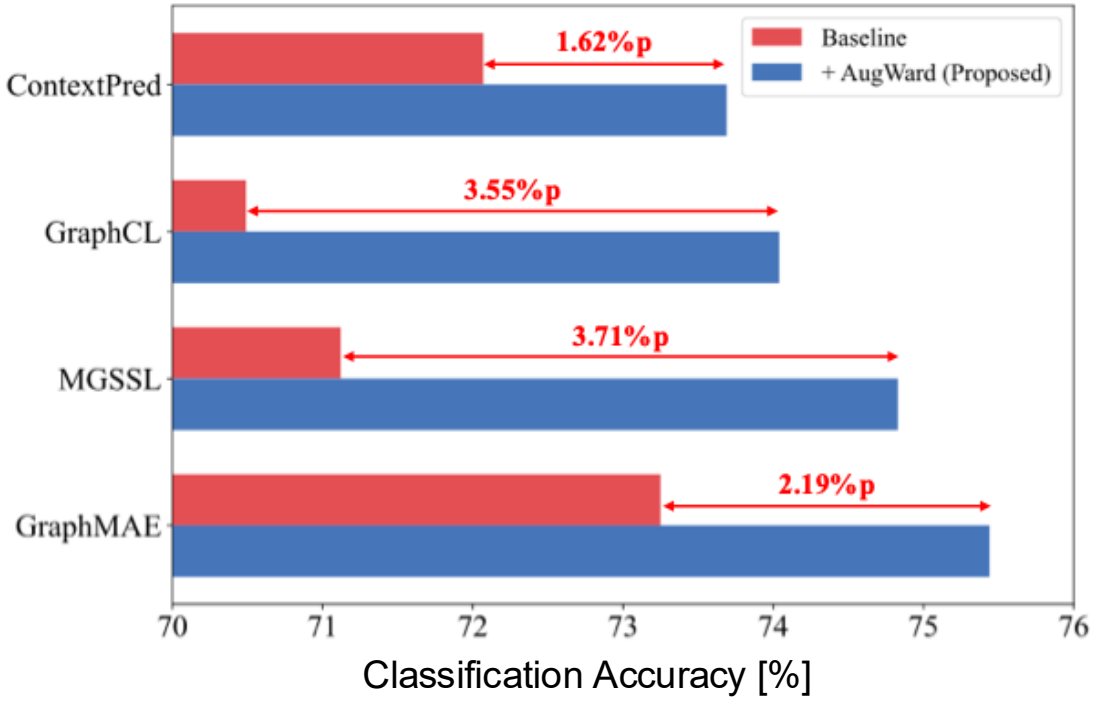
- AugWard is also beneficial in semi-supervised setting
  - Up to **1.52%p** increase in average accuracy



Experiments

3. Representation Transferability

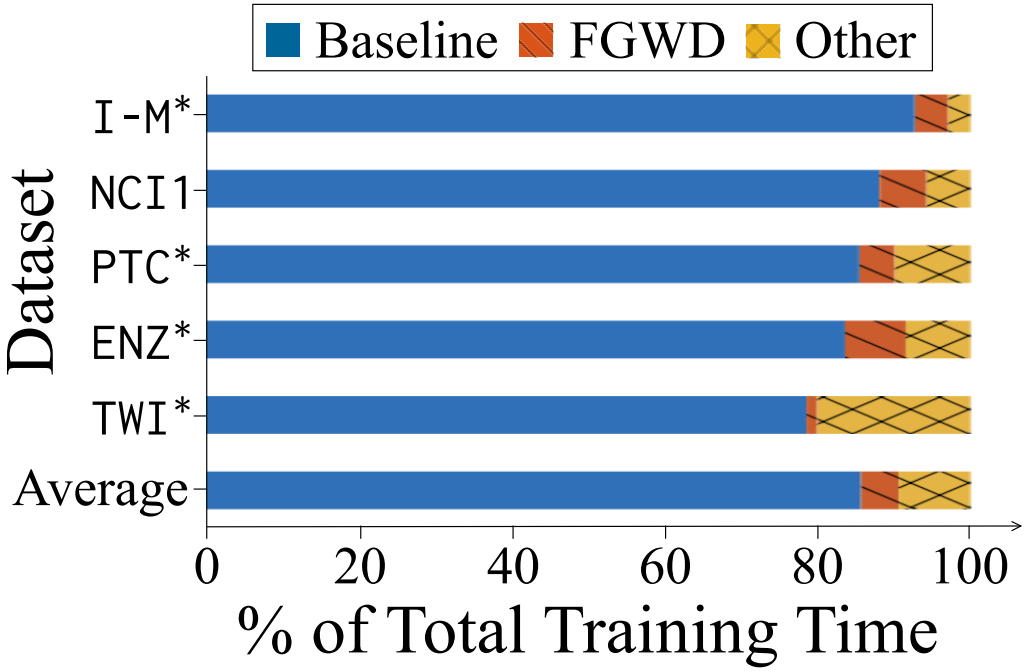
- AugWard offers more expressive representations
  - Improves the performance of transfer learning models; up to **3.71%p** in average accuracy



Experiments

4. Runtime Analysis

- The overhead from AugWard is marginal
  - Computing FGWD takes 4.89% of total time in average



Experiments

5. Ablation Study

- All ideas contribute to the enhanced performance
  - Considering the *augmentation-induced difference* is beneficial, even with simple heuristics

Variants	Ideas	Avg.	Imp.
A: GIN + NodeDrop	Existing	61.58	-
A + p	I1	61.90	+0.32
A + NFs	I1	62.50	+0.92
A + AMs	I1	62.55	+0.98
A + Edge Jaccard	I1	62.53	+0.95
A + FGWD	I1+I2	63.04	+1.46
A + **AugWard**	I1+I2+I3	**63.58**	**+2.00**

Naïve metrics for graph-level distance

FGWD (Proposed)

Outline

- Introduction
- Preliminaries
- Proposed Method
- Experiments
- **Conclusion**

Conclusion

- We propose **AugWard** for graph classification
  - AugWard considers the augmentation-induced differences
- **Main ideas**
  - **Ideas 1 & 2.** Augmentation-aware training with FGWD
  - **Ideas 3.** Consistency regularization
- **Experiments**
  - AugWard consistently enhances classification performance across various learning settings

Thank you !

Minjun Kim (minjun.kim@snu.ac.kr)

Paper

GitHub