Training Graph Neural Networks via Self-Supervised Learning: Experiments and Analysis

自監督學習於圖神經網路：實驗與分析

何青儒 Ho Ching-Ru

指導老師：顏佐榕、沈俊嚴

Data Science Degree Program, National Taiwan University.

July 4th, 2022

Outline

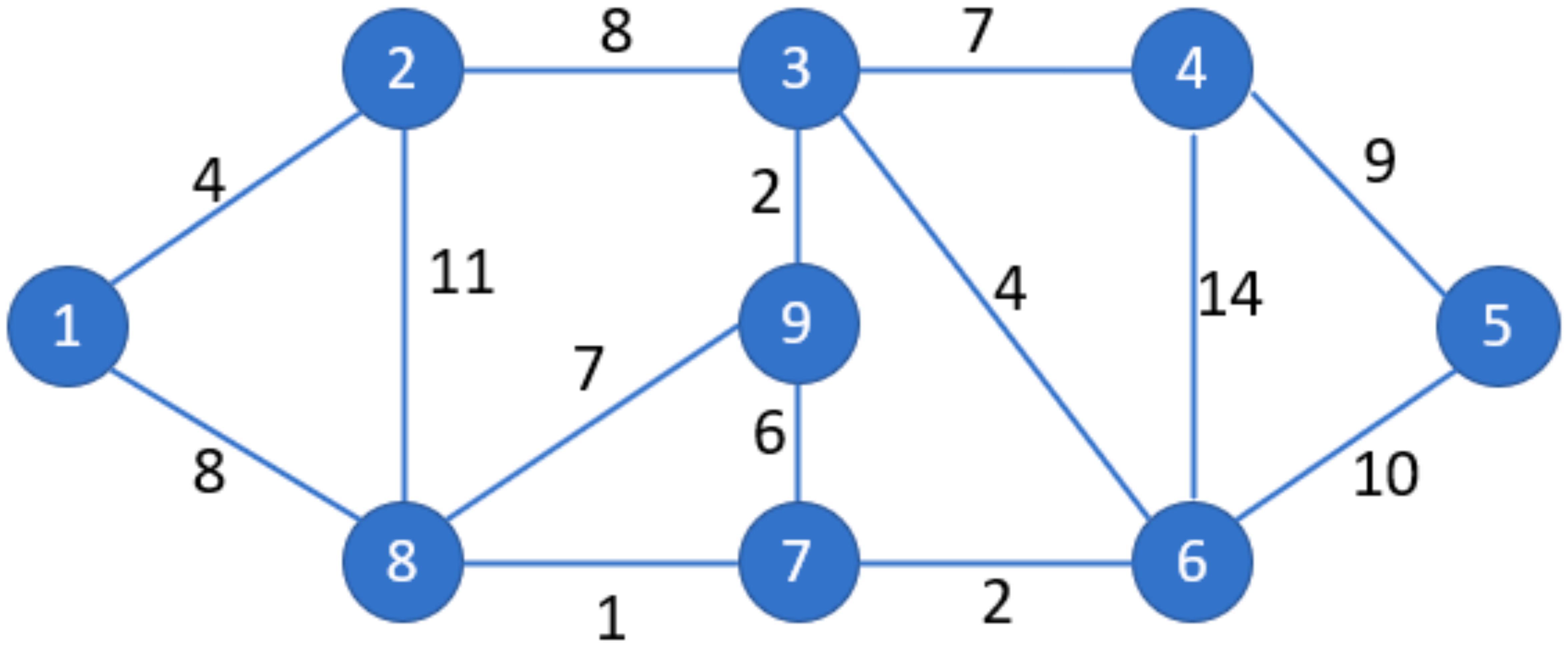
- Introduction
- Methodology
  - Dataset
  - Data Augmentation
  - SimCLR
  - Barlow Twins
  - Simsiam
- Results and Discussion

Introduction

Graph

A special data structure for non-tablize information

- Usage:
    - Knowledge Graph
    - Social Network
    - Recommendation System
    - Particle Simulation
    - Molecule Discovery
    - …etc.



A bi-directed graph with 9 nodes and 14 edges.

Graph Neural Network

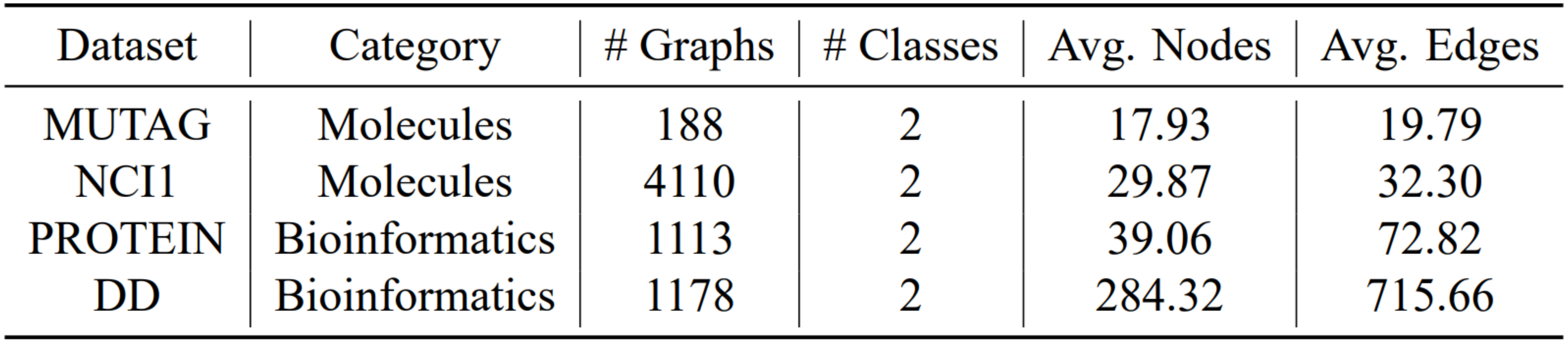
A branch of machine learning for graph-structured data

- By using different $\mathbf{COMBINE}(\cdot)$ and $\mathbf{AGGREGATE}(\cdot)$ operators, a number of architectures for encoding graph data have been proposed.
- e.g. GraphSAGE, Graph Convolutional Networks (GCN), Graph Isomorphism Network (GIN) …, etc.
- Our experiment: Graph Isomorphism Network:

$$\mathbf{h}_u^{(k)} = \mathrm{MLP}\Big(\big(1 + \epsilon^{(k)}\big) \cdot \mathbf{h}_u^{(k-1)} + \sum_{v \in \mathcal{N}_u} \mathbf{h}_v^{(k-1)} \Big)$$

Methodology

Datasets

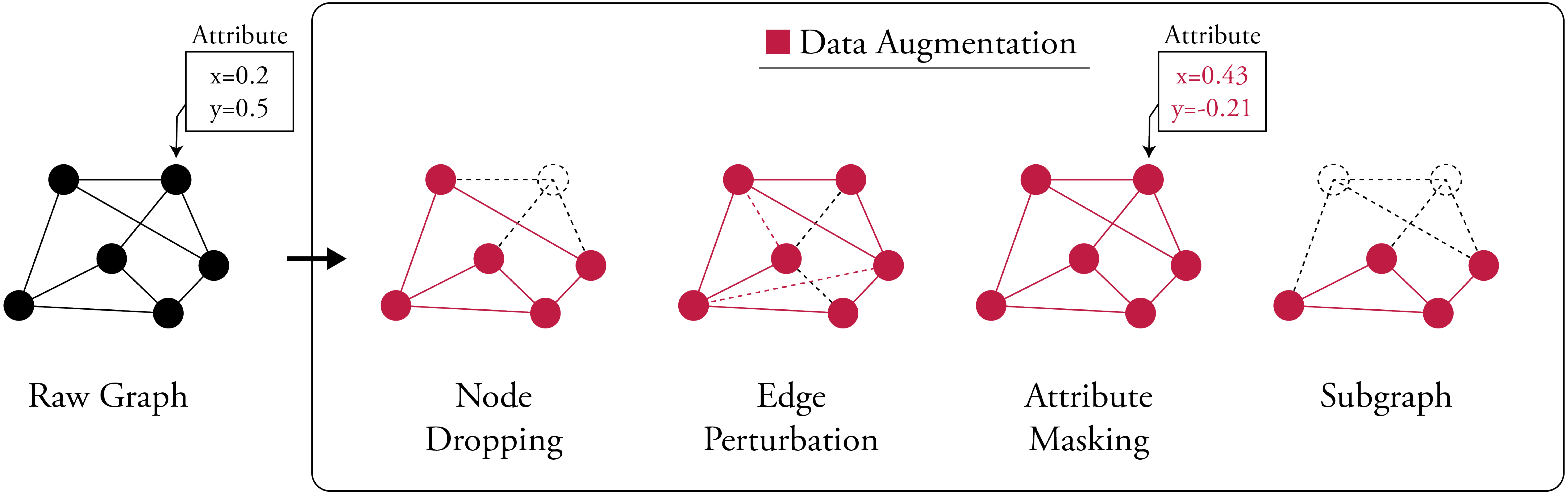


Dataset	Category	# Graphs	# Classes	Avg. Nodes	Avg. Edges
MUTAG	Molecules	188	2	17.93	19.79
NCI1	Molecules	4110	2	29.87	32.30
PROTEIN	Bioinformatics	1113	2	39.06	72.82
DD	Bioinformatics	1178	2	284.32	715.66

For more details, please refer to TUDataset.

Data Augmentation

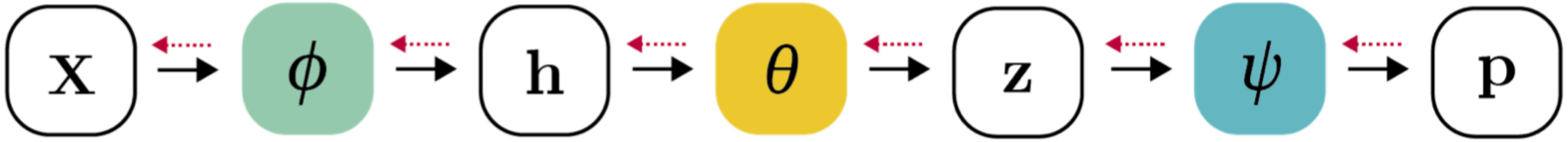
- Augmented ratio = 0.3



The red parts drawn in graphs are the output of data augmentation.

Notation

- Encoder: $\varphi(\cdot)$
- Projector: $\vartheta(\cdot)$
- Predictor: $\psi(\cdot)$
- Raw Data: $\mathbf{x}=\{\mathbf{u},\ \mathbf{e}\}$
- Embedding/representation: $\mathbf{h} = \varphi(\mathbf{x})$
- Projection: $\mathbf{z} = \vartheta(\mathbf{h})$
- Prediction: $\mathbf{p} = \psi(\mathbf{z})$



An illustration of relationships between embedding, projection and prediction.
Red arrow shows the step of back propagation.

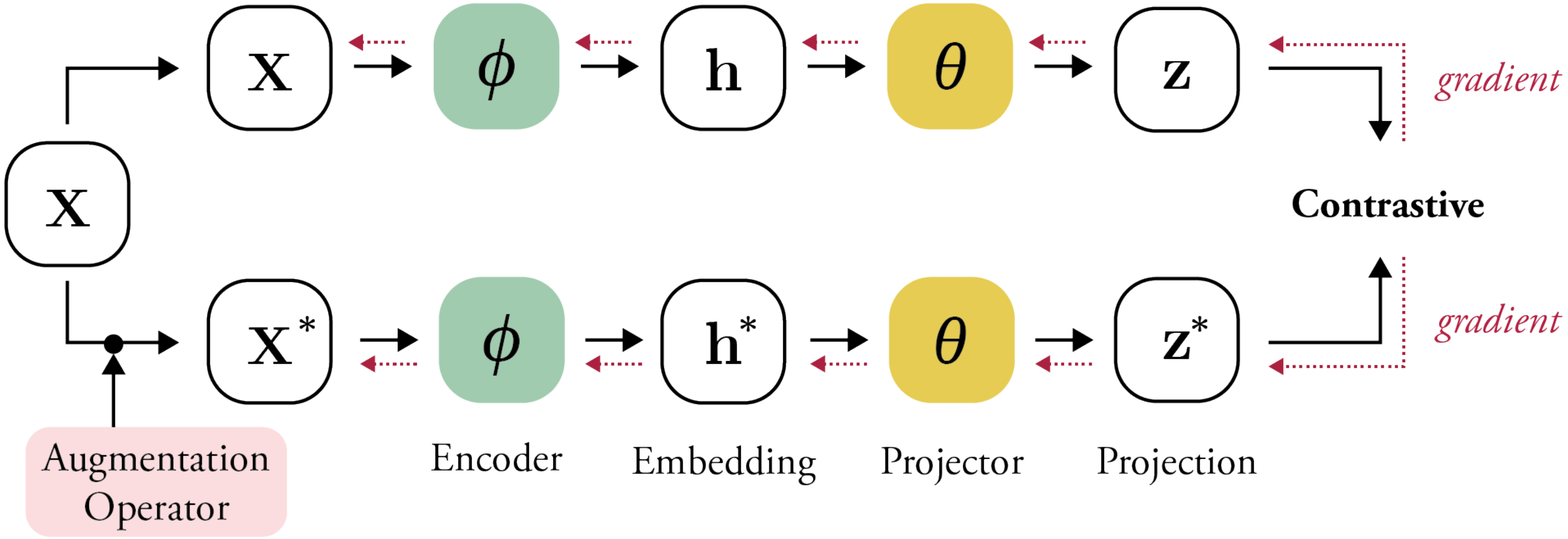
SimCLR

Contrastive Learning

$$Loss = -\log \frac{\exp\left(\mathbf{sim}(\mathbf{z}, \mathbf{z}^*)/\tau\right)}{\sum_{\mathbf{z}' \neq \mathbf{z}}^{N} \exp\left(\mathbf{sim}(\mathbf{z}, \mathbf{z}')/\tau\right))}$$

$$\mathbf{sim}(\mathbf{u}, \mathbf{v}) = \mathbf{u}^\intercal \mathbf{v}/(\|\mathbf{u}\|_2 \|\mathbf{v}\|_2)$$

- **Loss function**: NT-Xent loss between projections



Simsiam

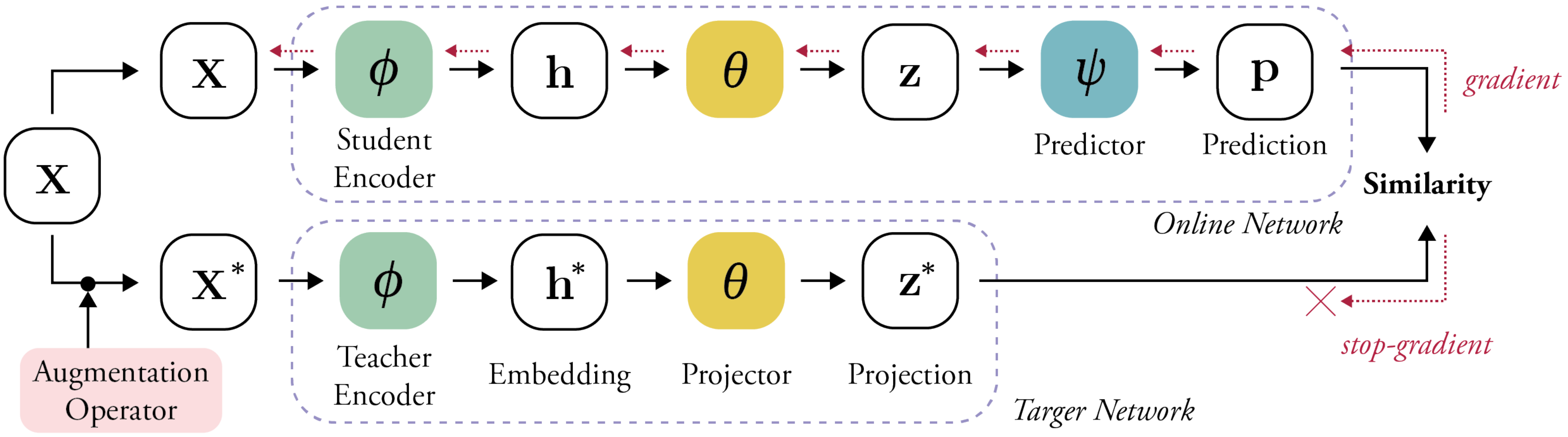
Distillation Learning

$$Loss = -\frac{1}{2}\sum_{\text{Dataset}}^{N}\Big(\mathbf{sim}(\mathbf{z}^*, \mathbf{p}) + \mathbf{sim}(\mathbf{z}, \mathbf{p}^*)\Big)$$

$$= -\frac{1}{2}\sum_{\text{Dataset}}^{N}\left(\frac{(\mathbf{z}^*)^\intercal \mathbf{p}}{\|\mathbf{z}^*\|_2\|\mathbf{p}\|_2} + \frac{\mathbf{z}^\intercal \mathbf{p}^*}{\|\mathbf{z}\|_2\|\mathbf{p}^*\|_2}\right)$$

- **Loss function**: cosine similarity between prediction and projection

.



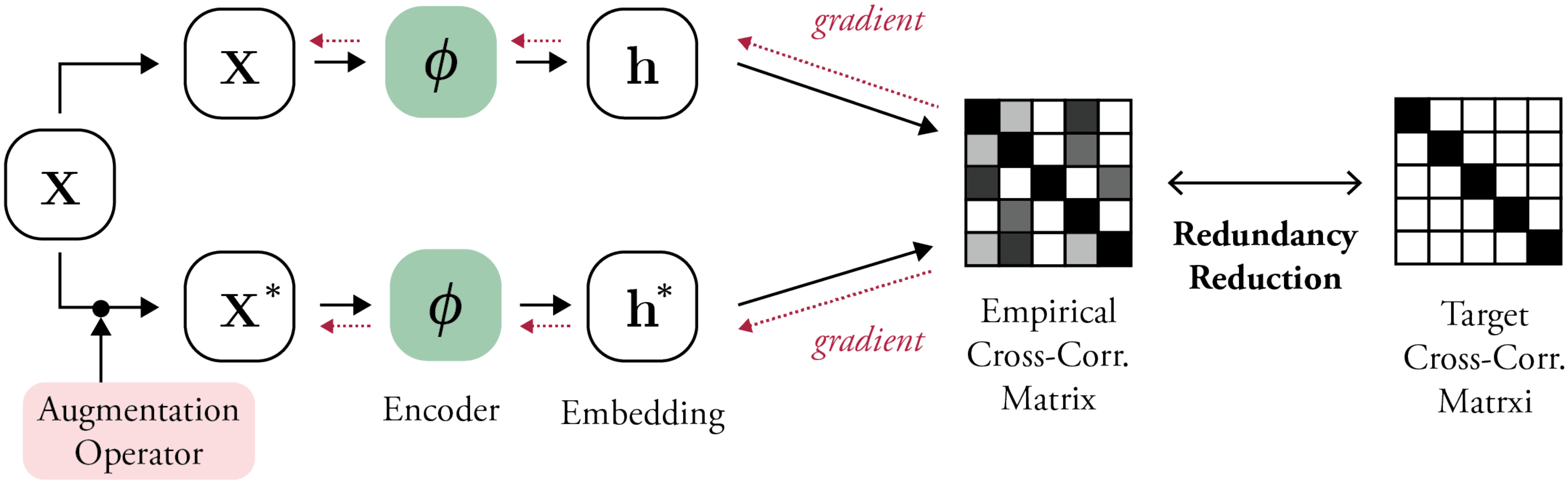
Barlow Twins

Redundancy Reduction

$$Loss = \underbrace{\sum_i (1 - \mathcal{C}_{ii})^2}_{\text{invariance term}} + \lambda \underbrace{\left(\sum_i \sum_{i \neq j} \mathcal{C}_{ij}^2\right)}_{\text{redundancy reduction term}}$$

- **Loss function**: invariance and redundancy reduction

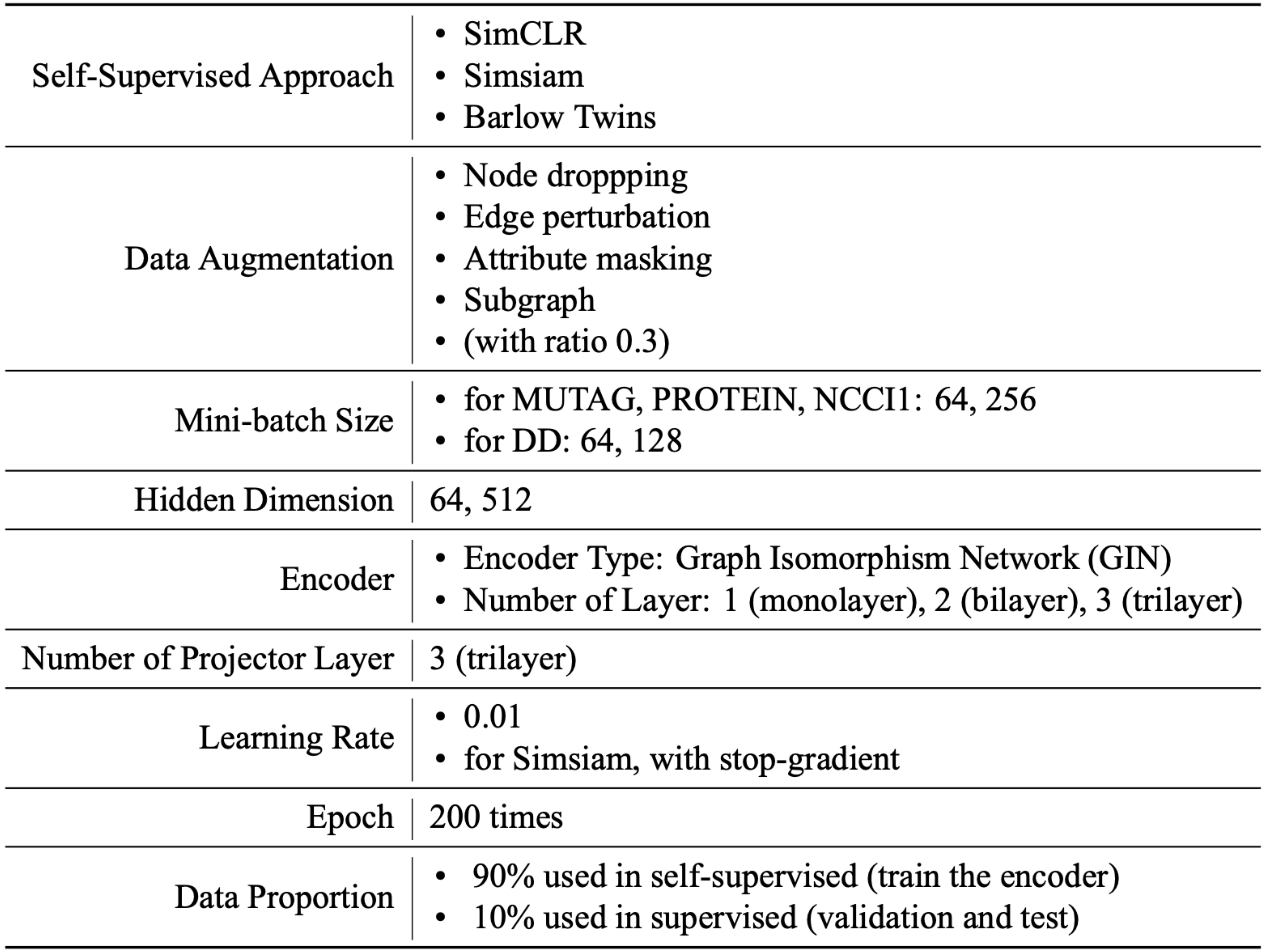
$$\mathcal{C}_{ij} = \frac{\sum_b \left((\tilde{\mathbf{h}}_{\mathbf{x}})_{b,i}\right) \cdot \left((\tilde{\mathbf{h}}^*_{\mathbf{x}})_{b,j}\right)}{\sqrt{\sum_b \left((\tilde{\mathbf{h}}_{\mathbf{x}})_{b,i}\right)} \cdot \sqrt{\sum_b \left((\tilde{\mathbf{h}}^*_{\mathbf{x}})_{b,j}\right)}}$$



Experiment Factors

144 experiments in total

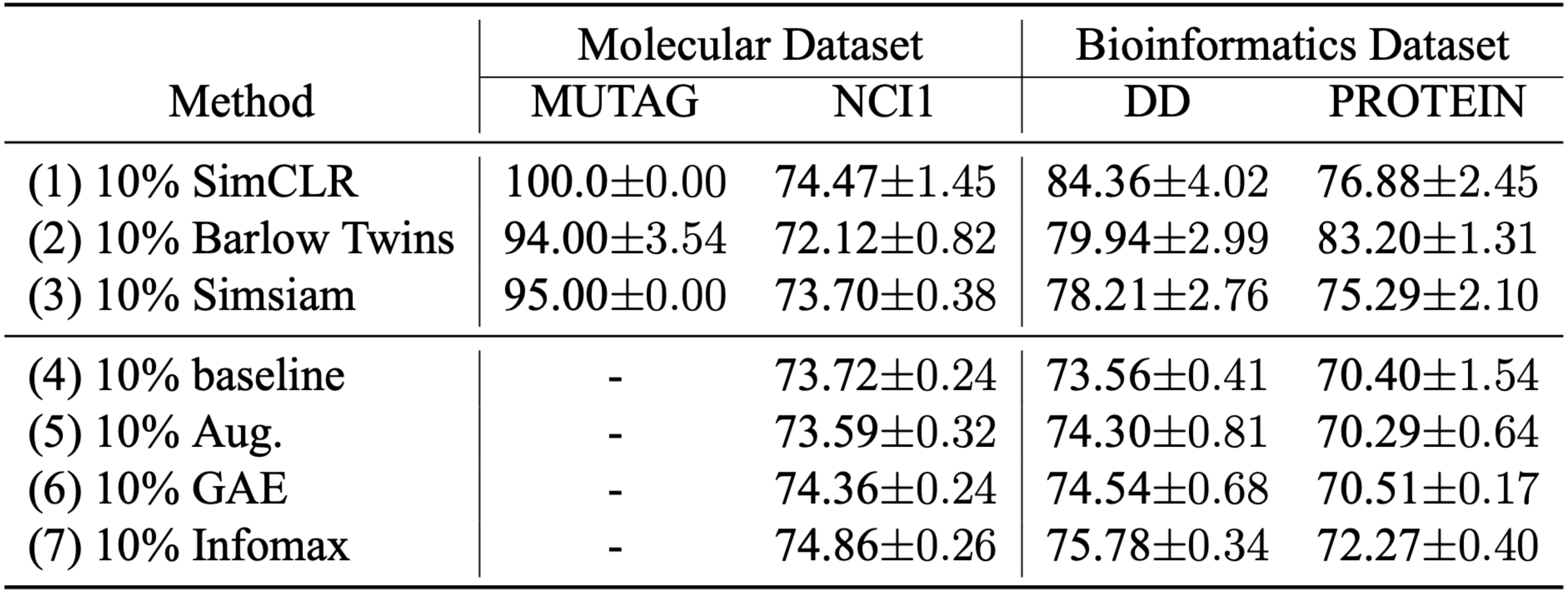
- Each experiment repeats 5 times to obtain an average accuracy and standard deviation.



Self-Supervised Approach	• SimCLR • Simsiam • Barlow Twins
Data Augmentation	• Node droppping • Edge perturbation • Attribute masking • Subgraph • (with ratio 0.3)
Mini-batch Size	• for MUTAG, PROTEIN, NCCI1: 64, 256 • for DD: 64, 128
Hidden Dimension	64, 512
Encoder	• Encoder Type: Graph Isomorphism Network (GIN) • Number of Layer: 1 (monolayer), 2 (bilayer), 3 (trilayer)
Number of Projector Layer	3 (trilayer)
Learning Rate	• 0.01 • for Simsiam, with stop-gradient
Epoch	200 times
Data Proportion	• 90% used in self-supervised (train the encoder) • 10% used in supervised (validation and test)

Results and Discussion

Experimental results



Method	Molecular Dataset		Bioinformatics Dataset	
	MUTAG	NCI1	DD	PROTEIN
(1) 10% SimCLR	100.0±0.00	74.47±1.45	84.36±4.02	76.88±2.45
(2) 10% Barlow Twins	94.00±3.54	72.12±0.82	79.94±2.99	83.20±1.31
(3) 10% Simsiam	95.00±0.00	73.70±0.38	78.21±2.76	75.29±2.10
(4) 10% baseline	-	73.72±0.24	73.56±0.41	70.40±1.54
(5) 10% Aug.	-	73.59±0.32	74.30±0.81	70.29±0.64
(6) 10% GAE	-	74.36±0.24	74.54±0.68	70.51±0.17
(7) 10% Infomax	-	74.86±0.26	75.78±0.34	72.27±0.40

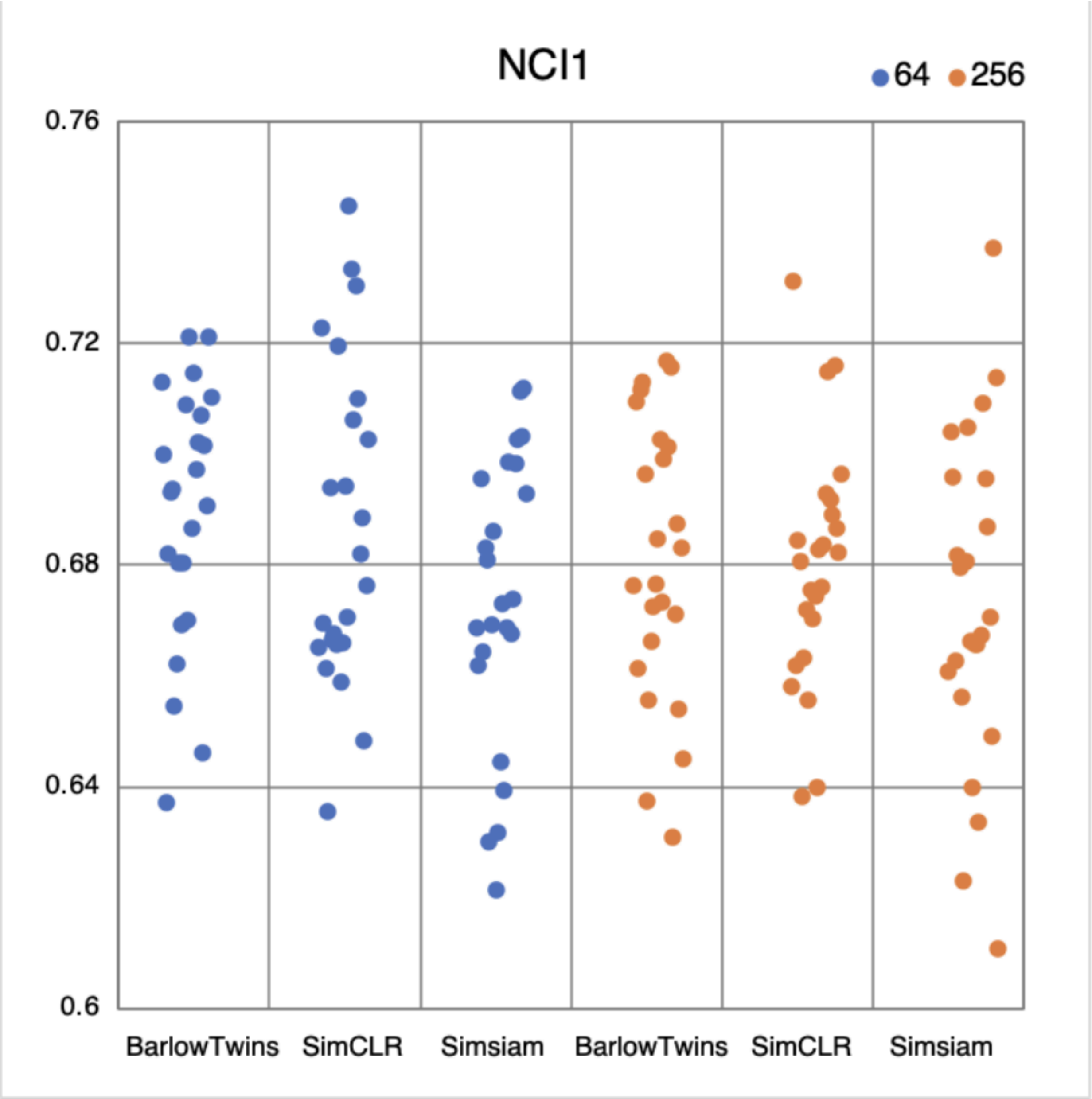
Results (1) to (3) are made by our experiments, and Result (4) to (7) refer to You et al. (2020)

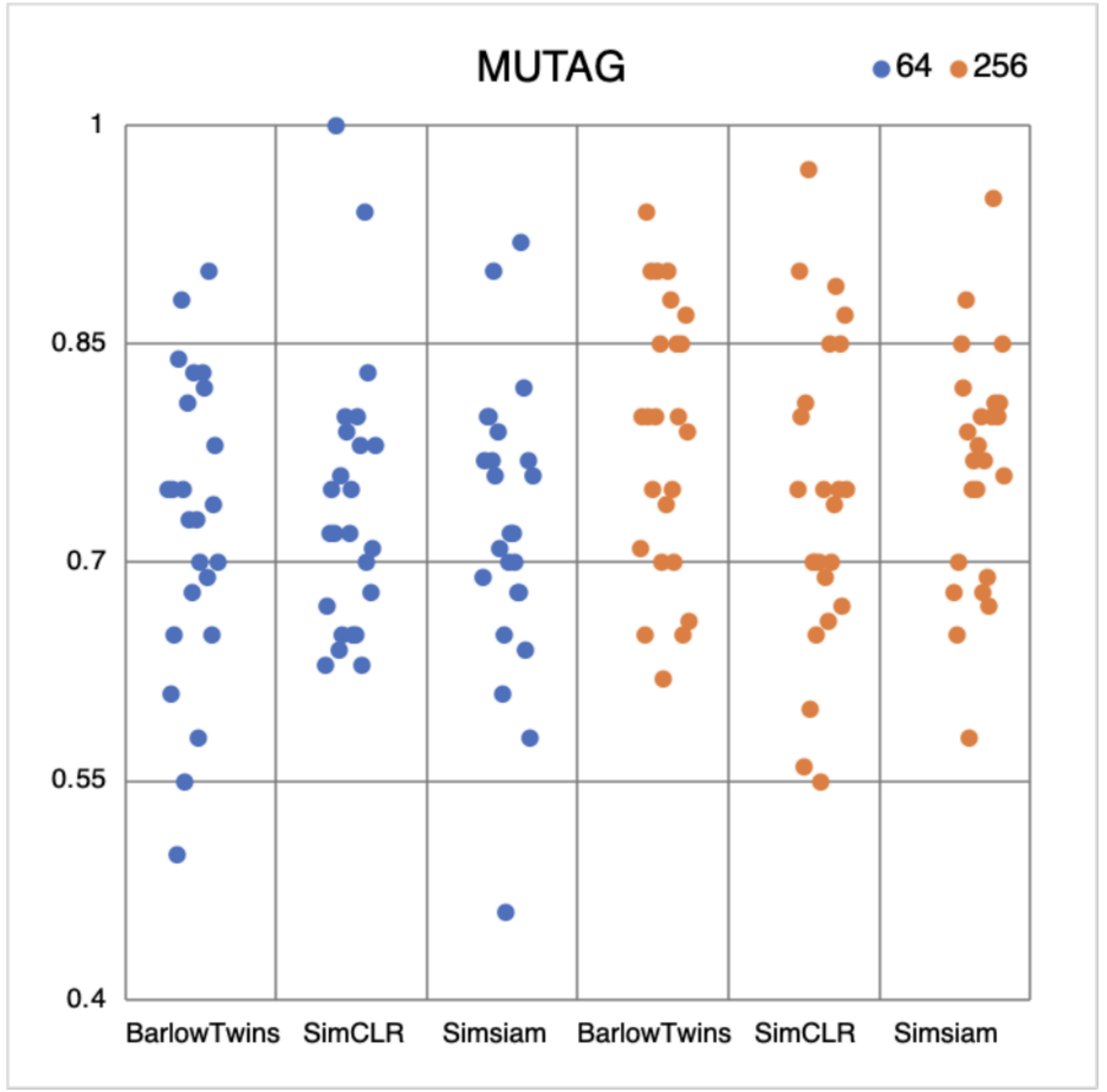
Batch size's effects on SimCLR are not apparent

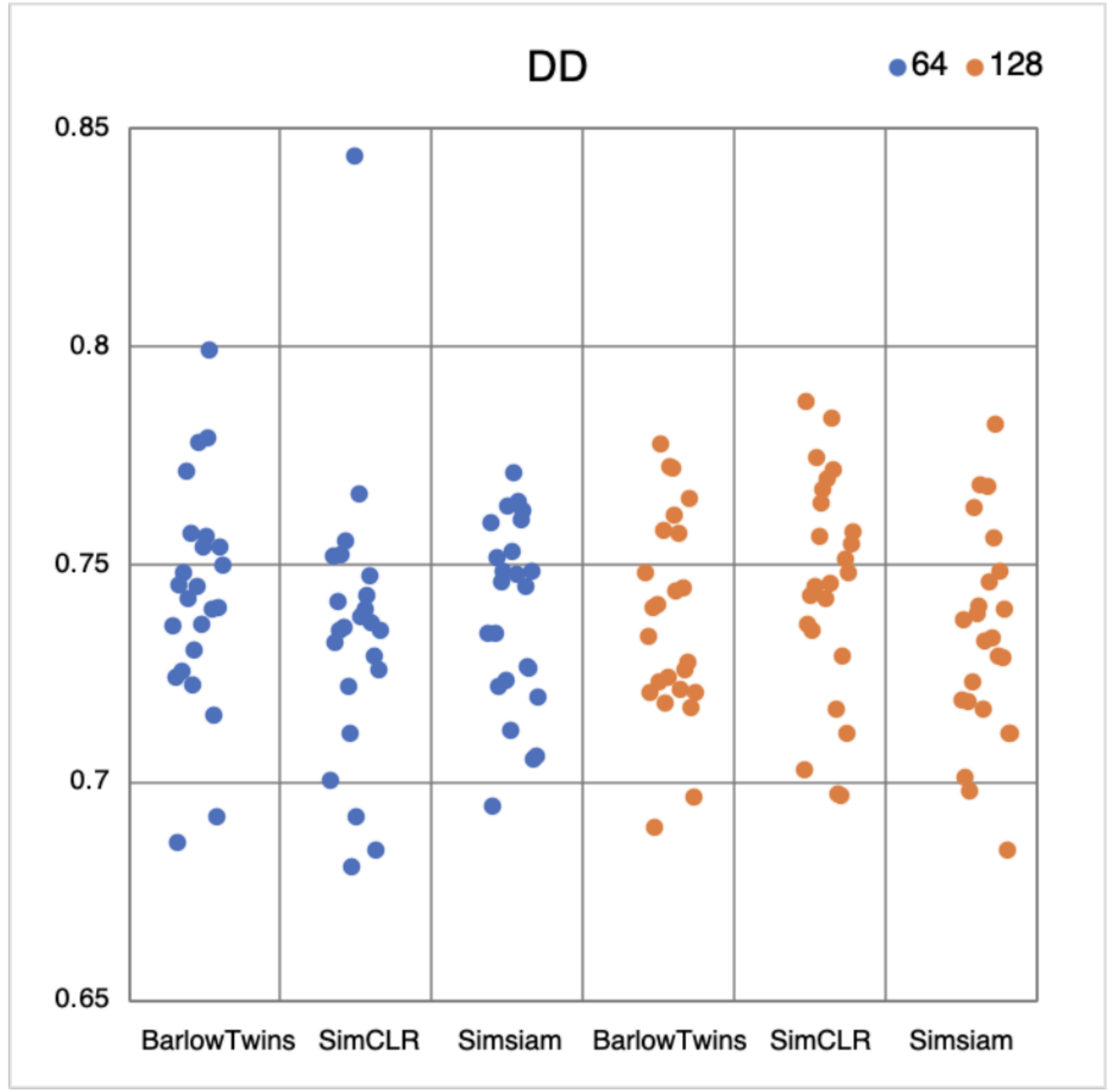
- In the paper of SimCLR, the authors pointed out that contrastive learning benefits more from a larger batch size and longer training time.
- They have tried different batch size from 256, 512, 1024, 2048, 4096 to 8192 on ResNet-50 model for image self-supervised task.
- However, due to the constraints of our computational power, we only applied 64 and 512 batch sizes (DD uses 64 and 128).
- In addition, the sizes of datasets we used are not large. All of them contain less than ten thousand samples for model training.
- This makes the effect of the batch size less apparent on the SimCLR method.

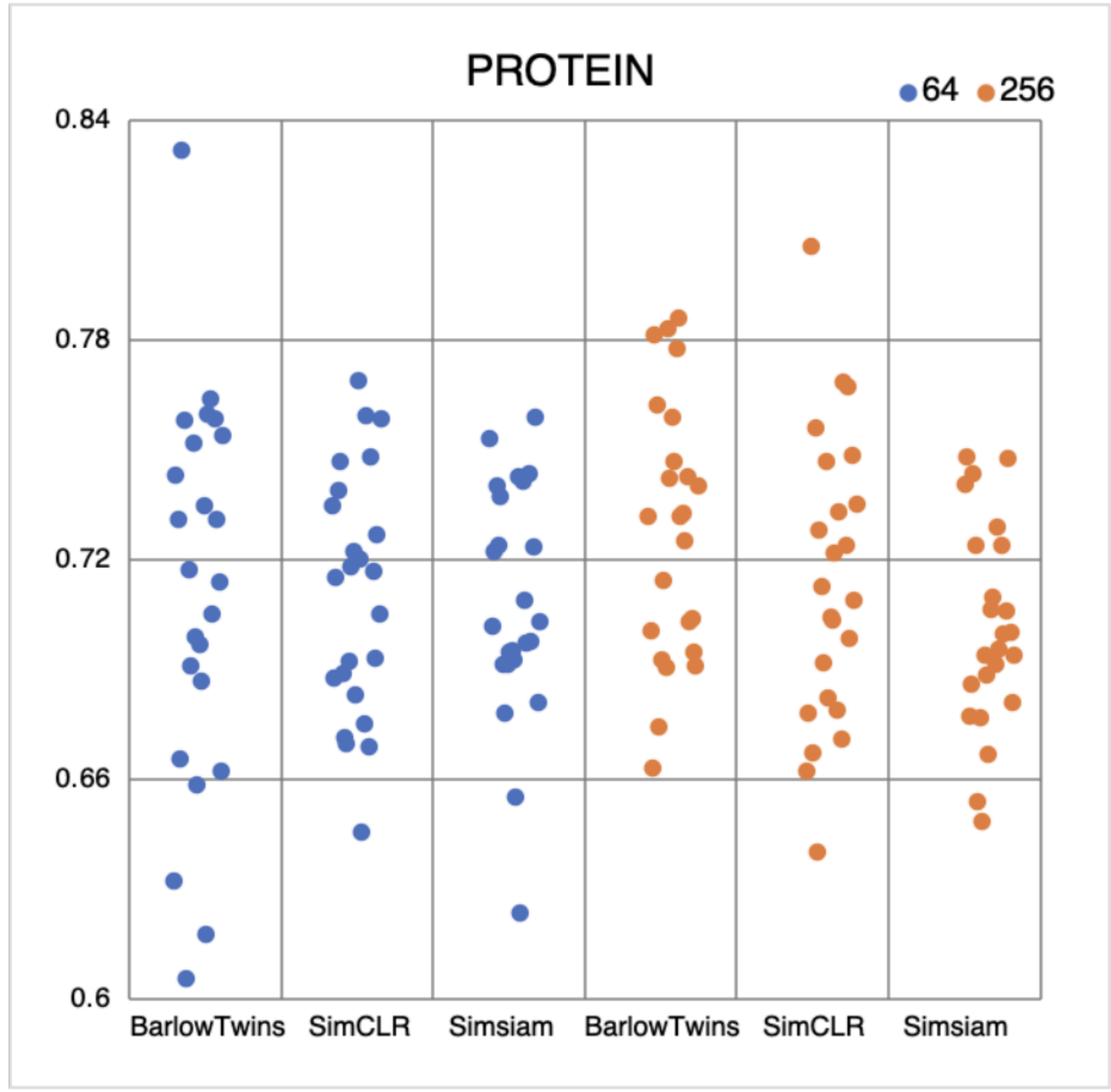
Batch size's effects on SimCLR are not apparent

- If the dataset is medium or small size, it seems that increasing the batch size may not be effective for getting better model performance under self-supervised learning.







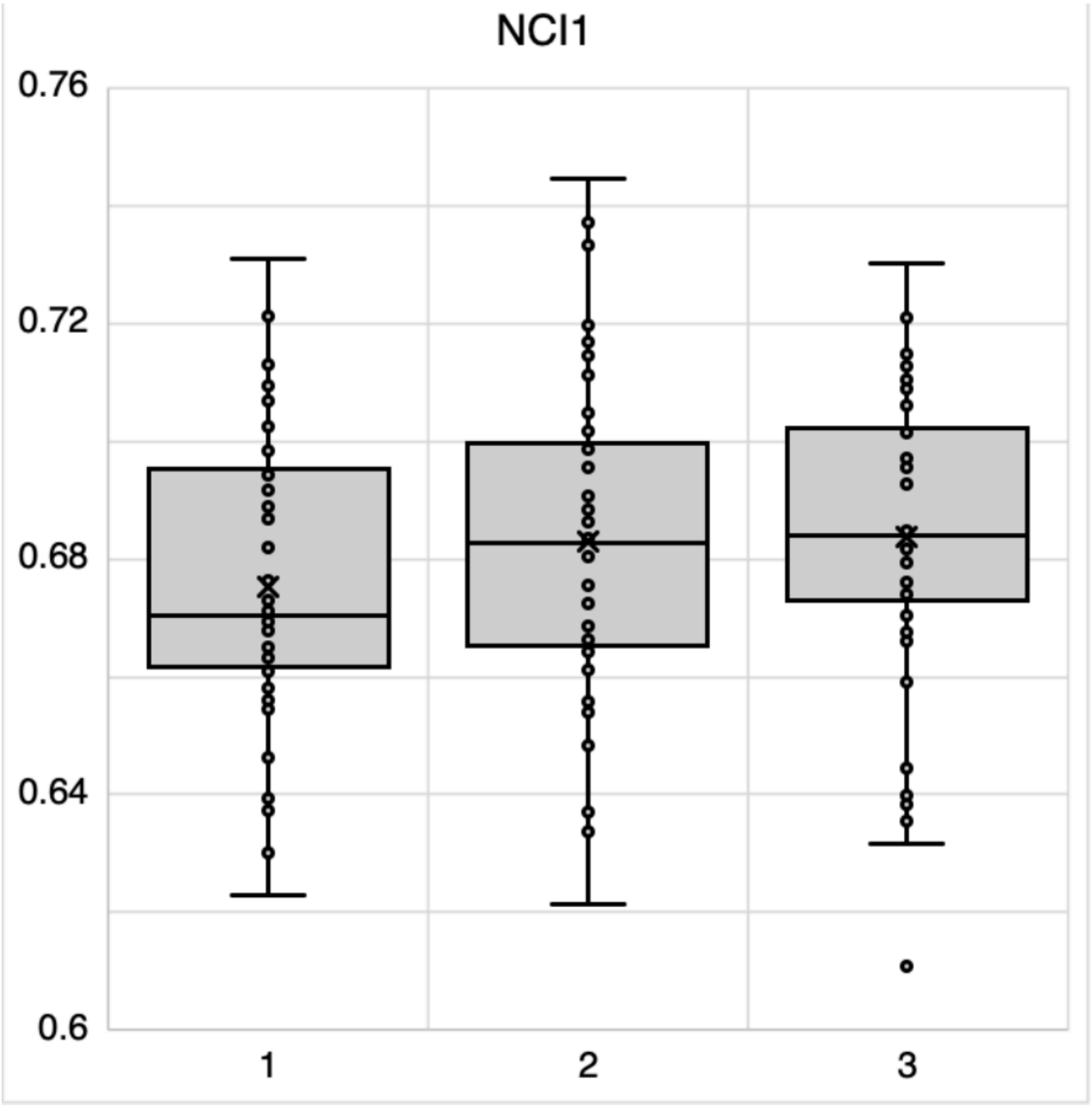


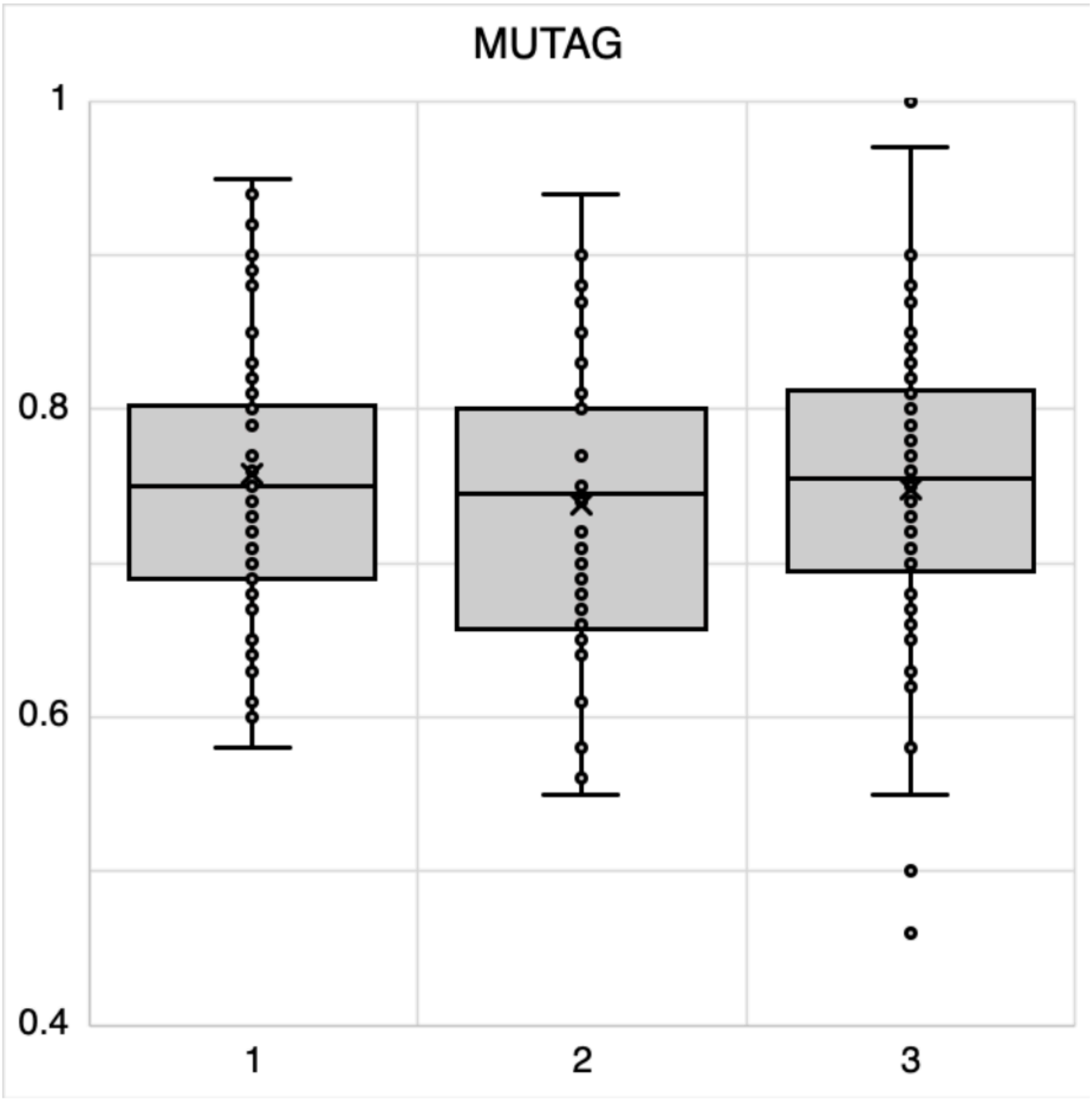
Deeper encoders have better performance

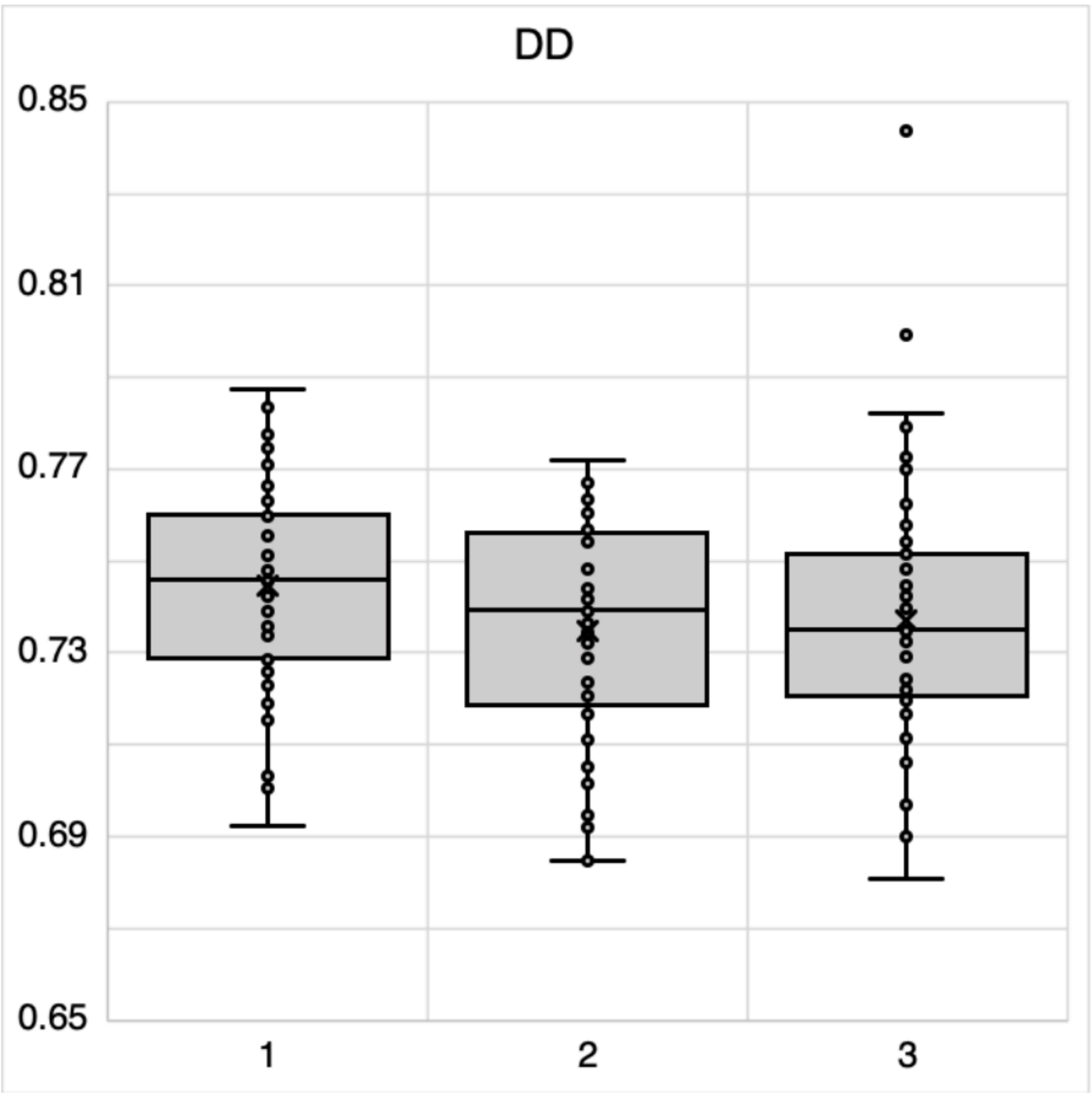
- Generally, models using a deeper encoder can capture more features and they are believed to have better performance than models employing a shallow encoder.
- Does this concept also work on graph data and self-supervised learning?
- We used a Graph Isomorphism Network (GIN) encoder to generate embeddings of the samples, and try three types of MLPs encoder.

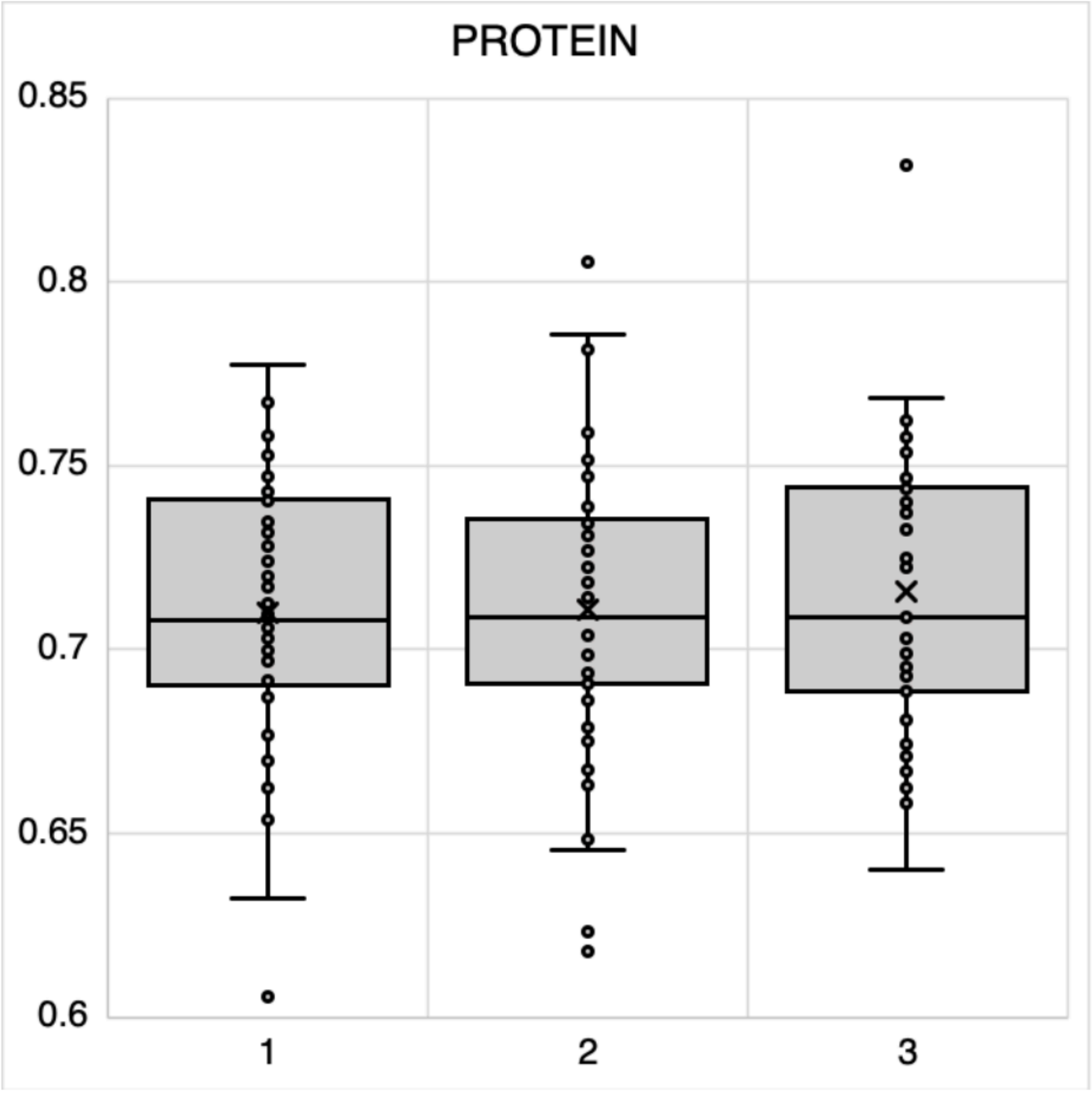
Deeper encoders have better performance

- The hypothesis that the deeper encoder also benefits on graph-structred model, is verified through the experiment results.







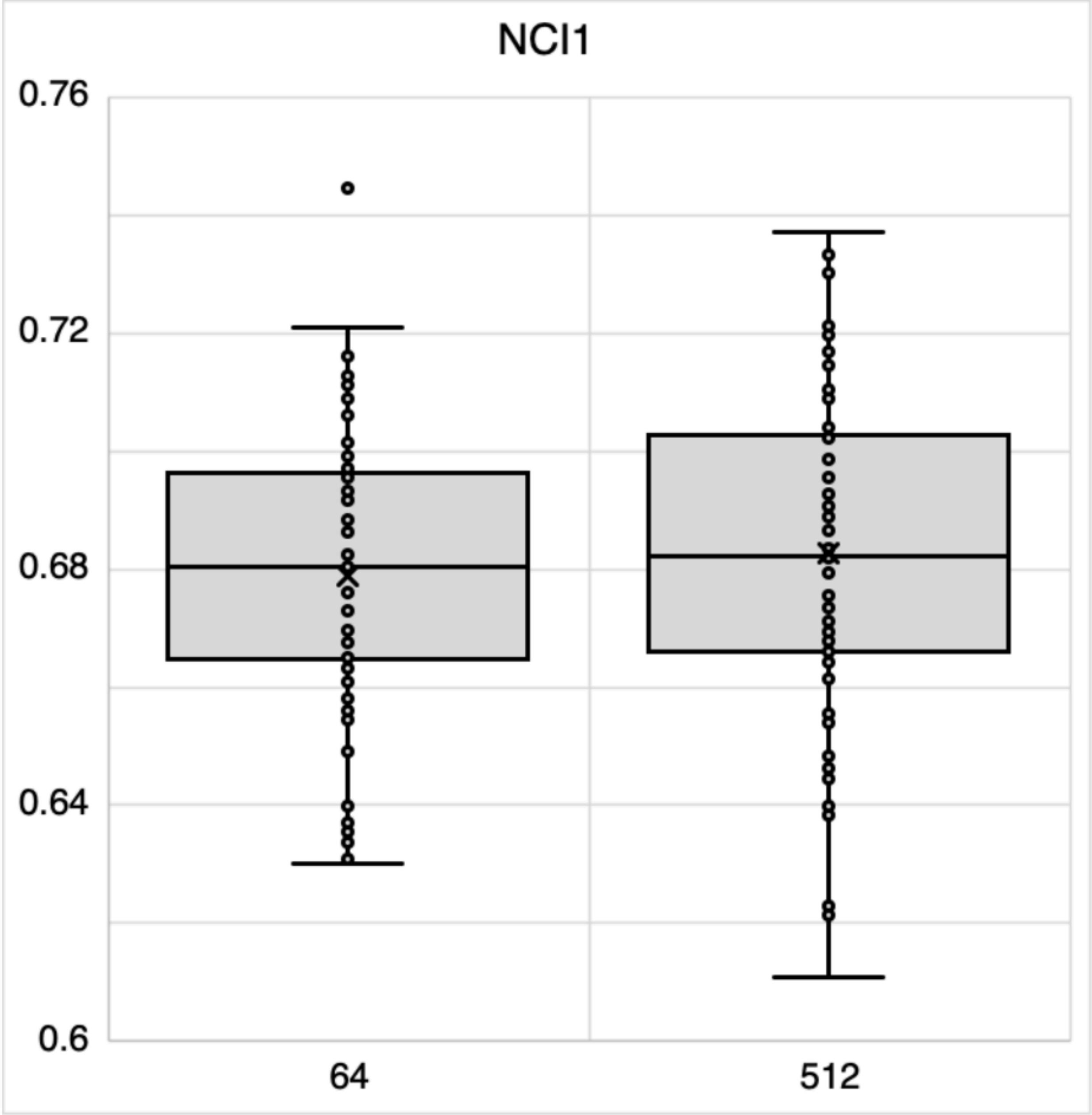


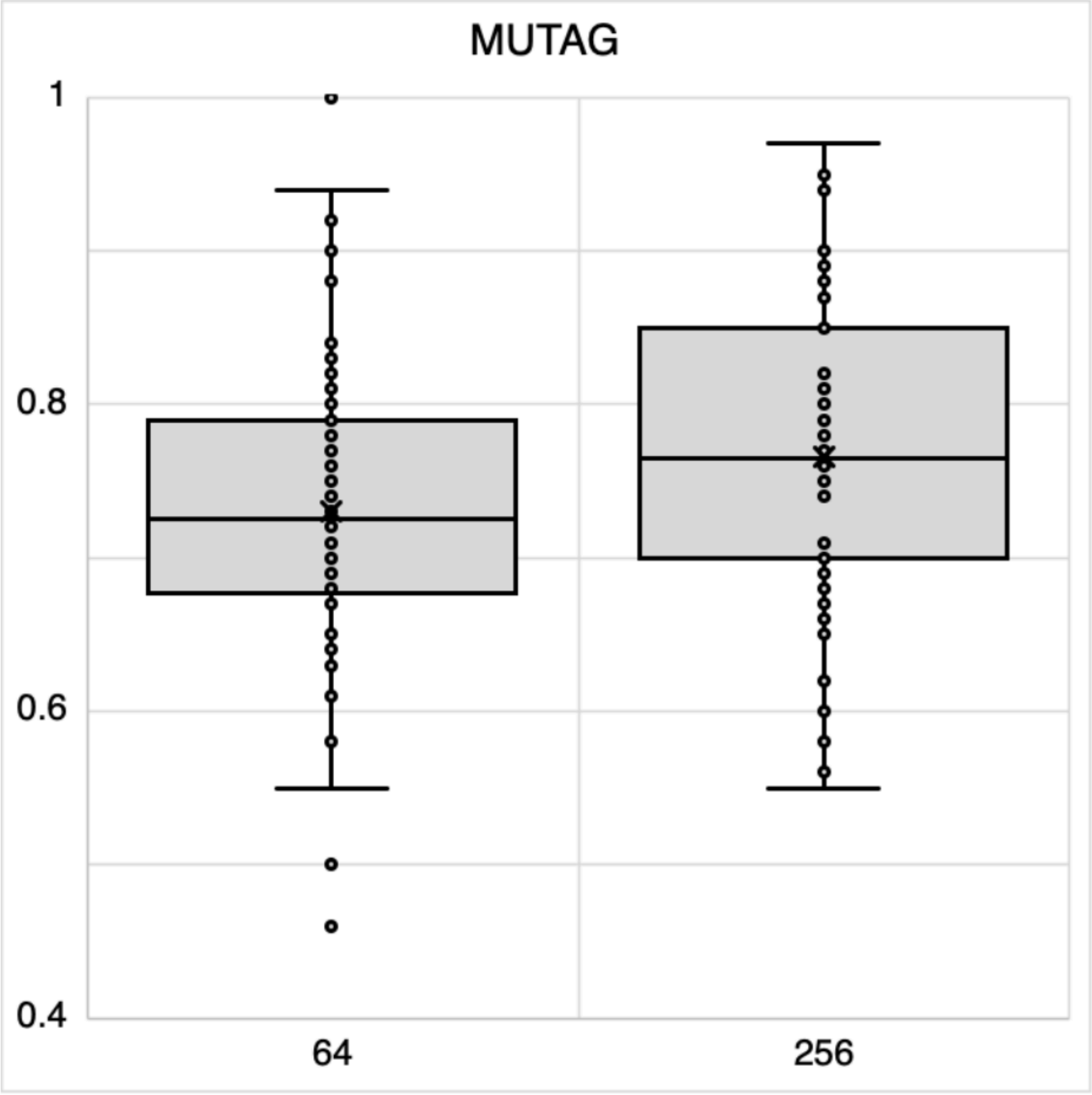
Hidden dimension has little effects on model performance

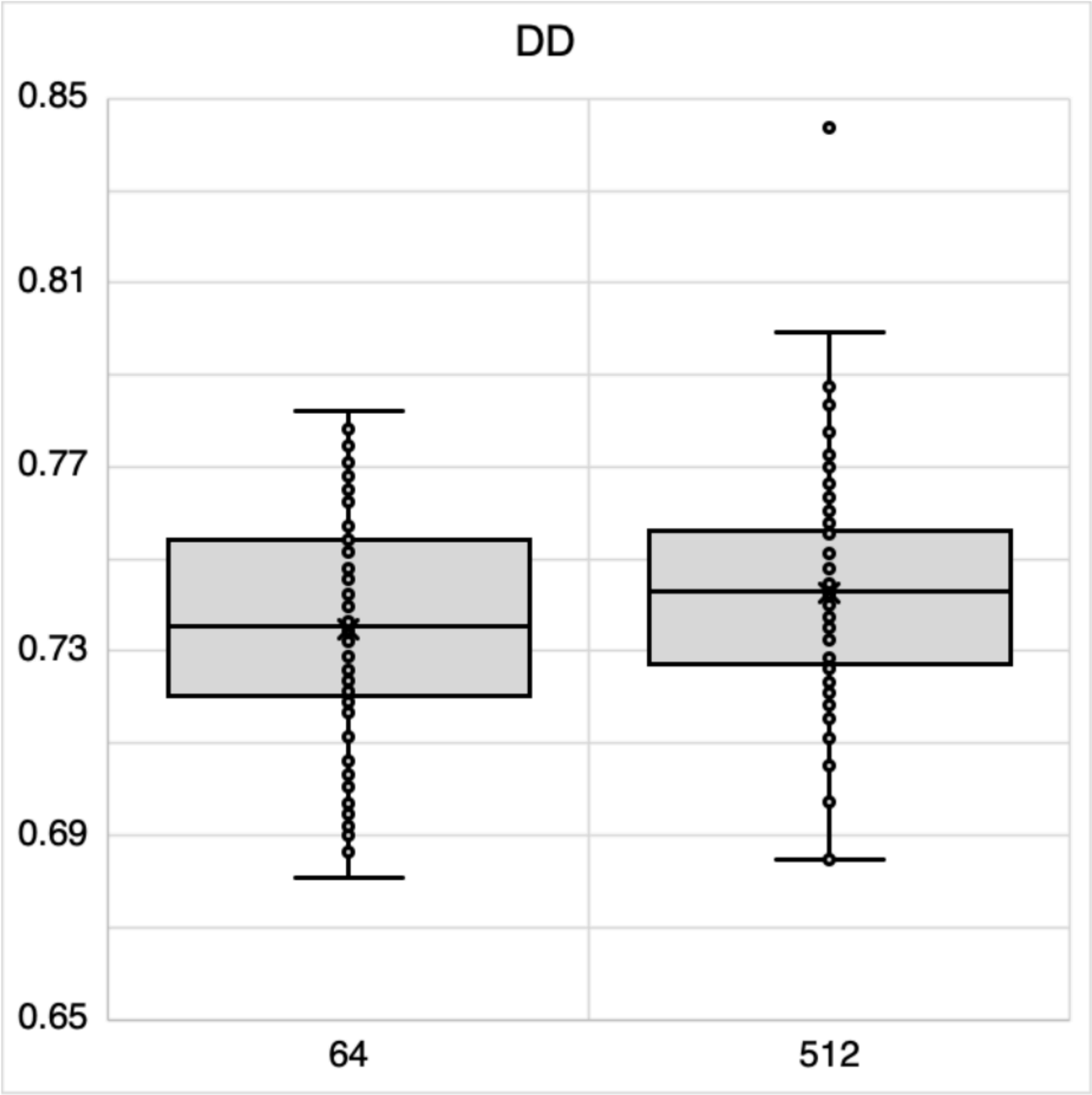
- In most state-of-the-art neural networks, especially in image and NLP detection and classification tasks, the hidden dimension of each encoder layer plays an influential role in model performance.
- Does this concept also work on graph data and self-supervised learning?
- However, for models with 512 hidden dimension, even the vector size is 8 times larger than those with 64 hidden dimensions, the performance does not have significant improvements.

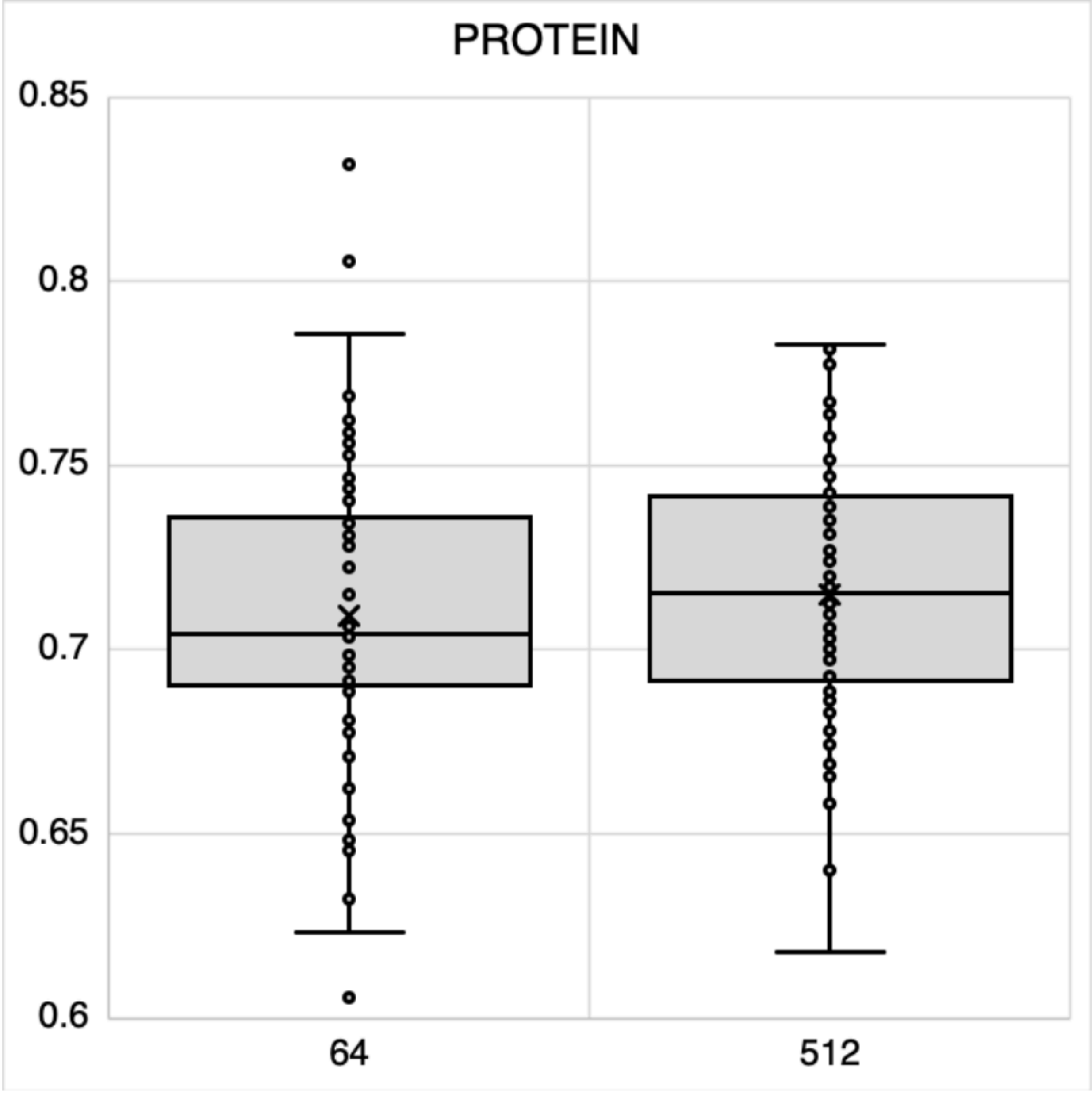
Hidden dimension has little effects on model performance

- For an image data, which consist of discrete pixels, with larger hidden dimension, the neuron can capture more information via pixels clustering. However, graph data are more abstract.







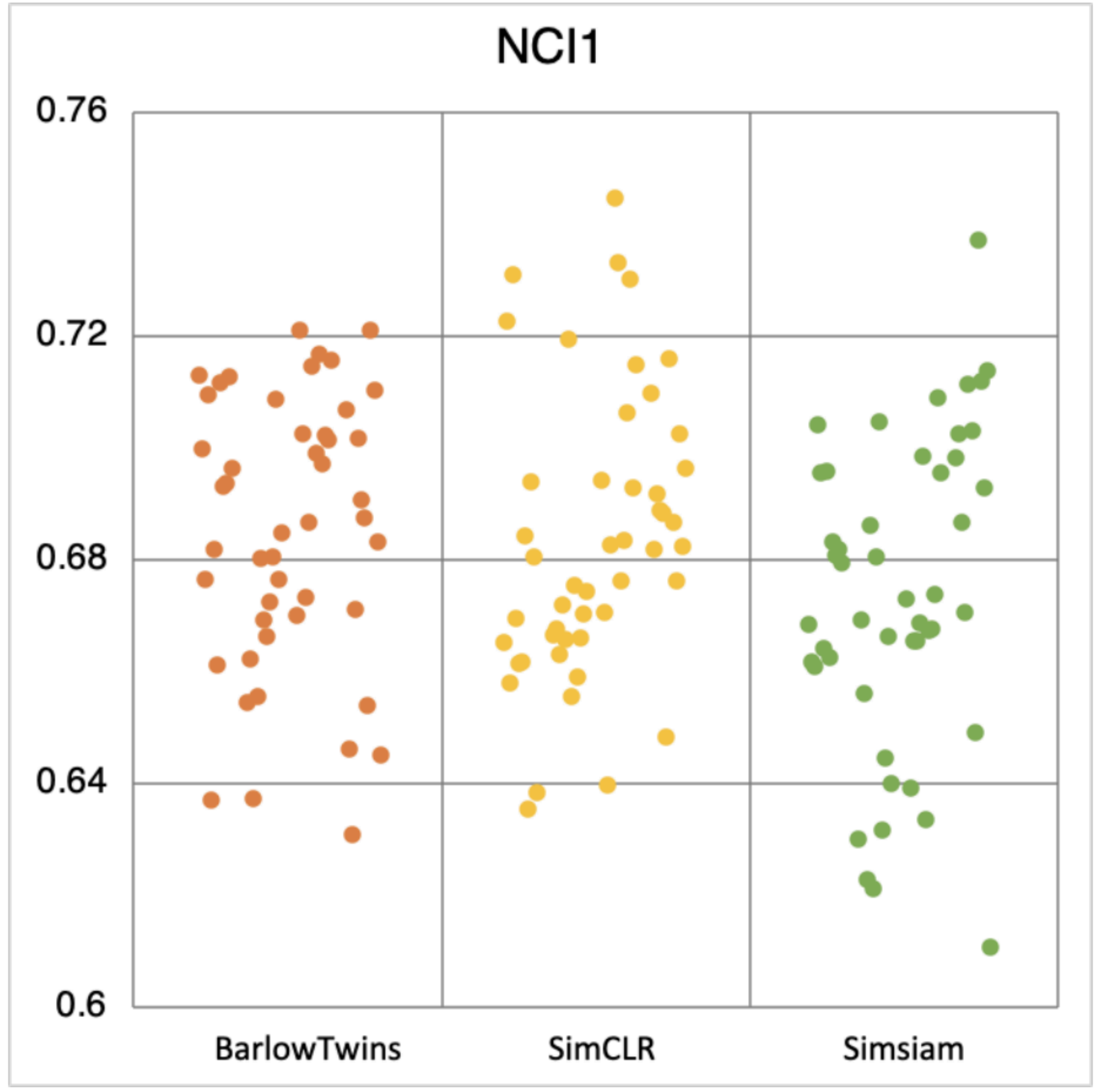


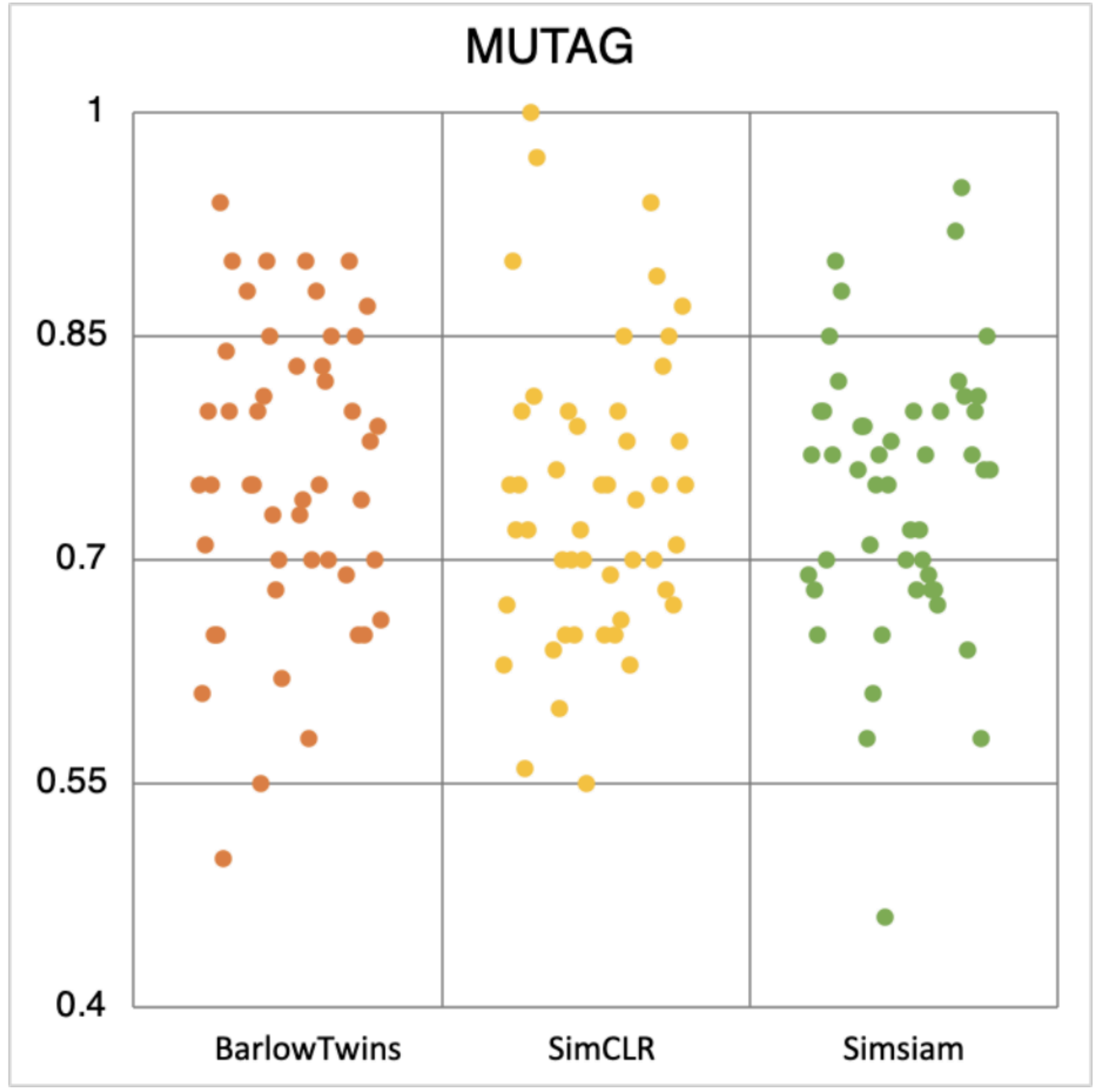
SimCLR performs better in molecular datasets

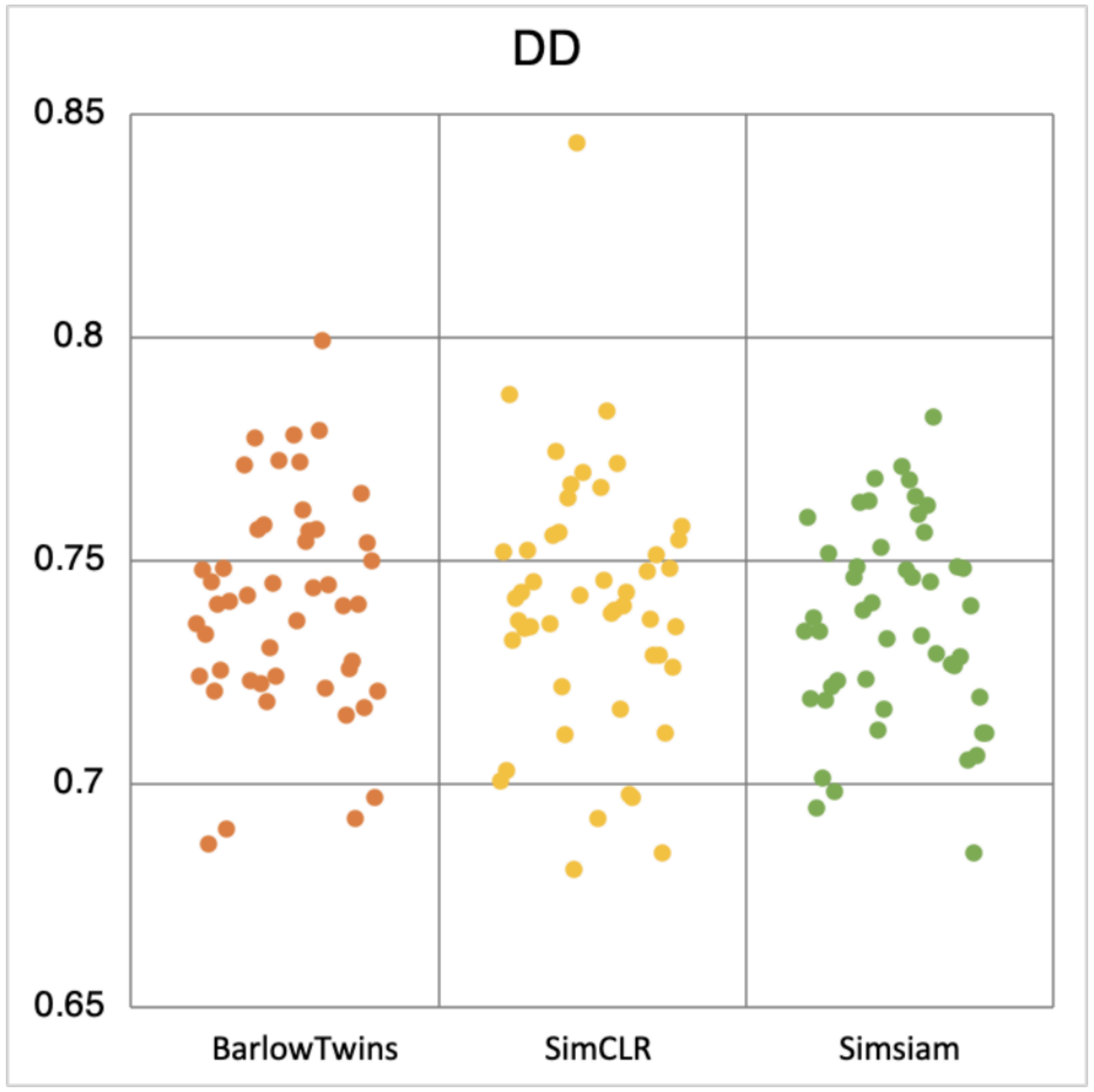
- **Bioinformatics datasets: DD, PROTEIN**
  - Node: an amino acid
  - Edge: if both nodes are less than 6 Angstroms apart
- **Molecular datasets: NCI1, MUTAG**
  - Node: an atom
  - Edge: the chemical bond connecting two nodes

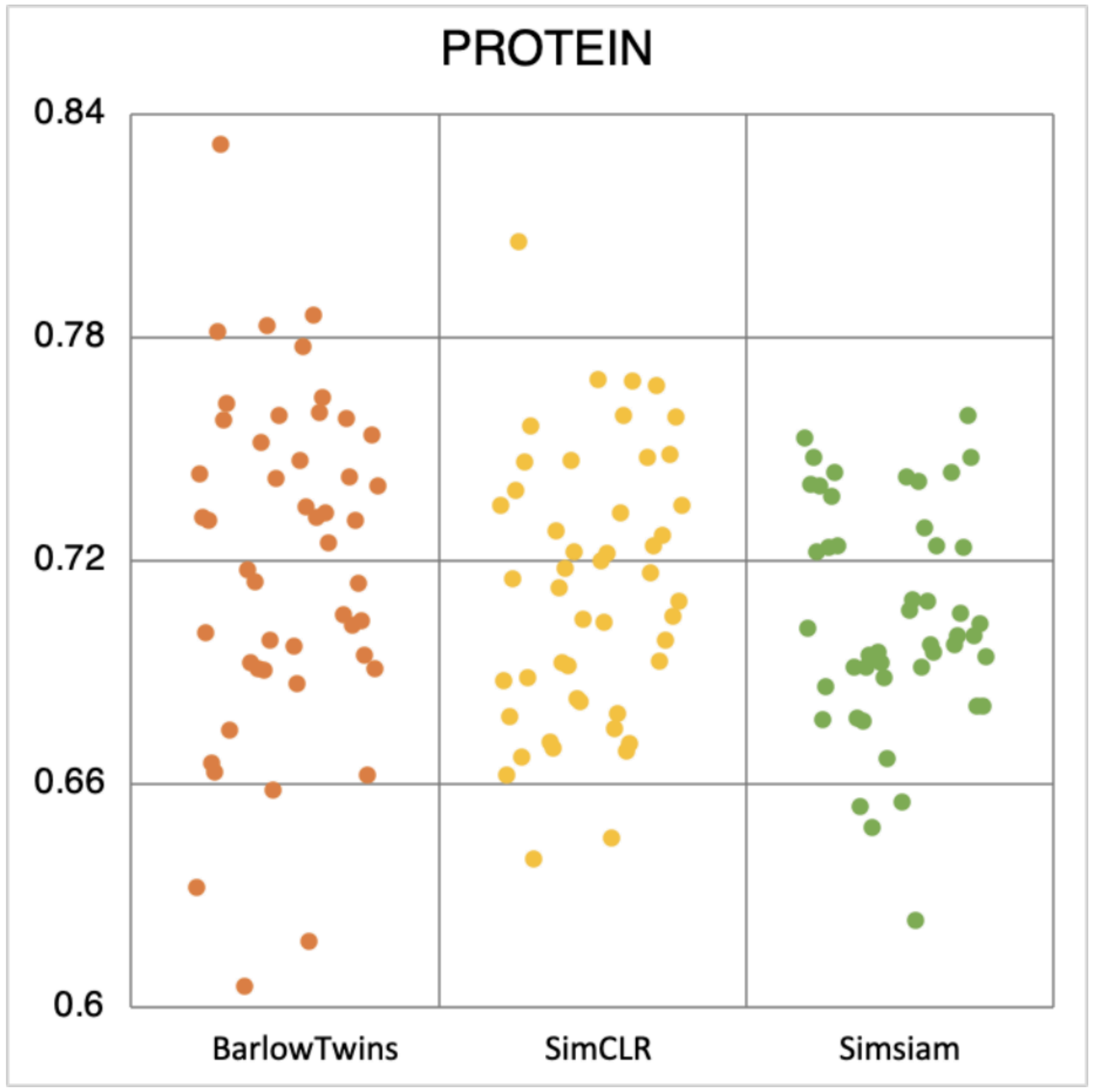
SimCLR performs better in molecular datasets

- Unlike Simsiam only using one projection, or Barlow Twins using embeddings directly, in SimCLR architecture, the raw data as well as the augmented data are encoded into projections, which might affect performance.







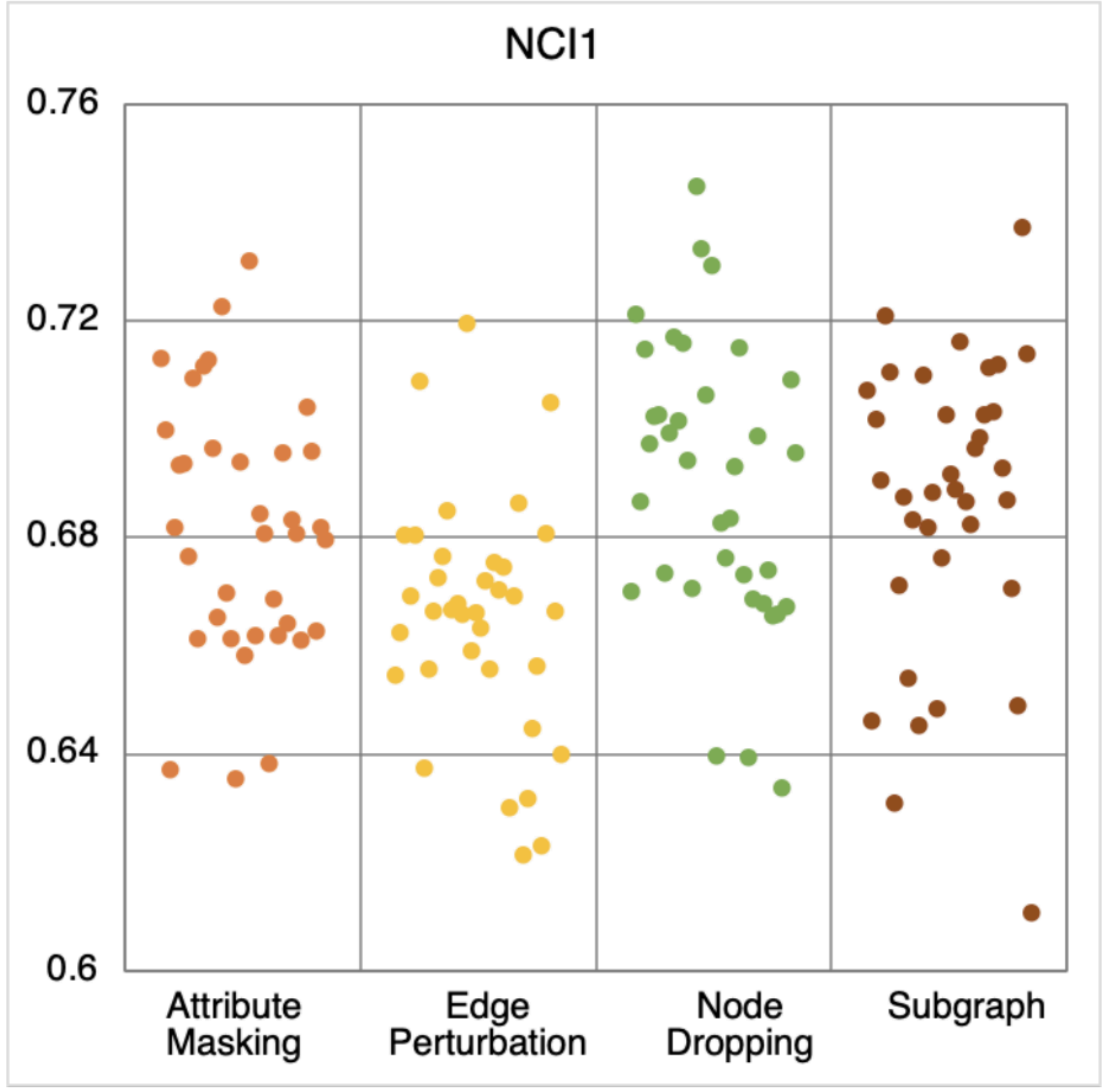


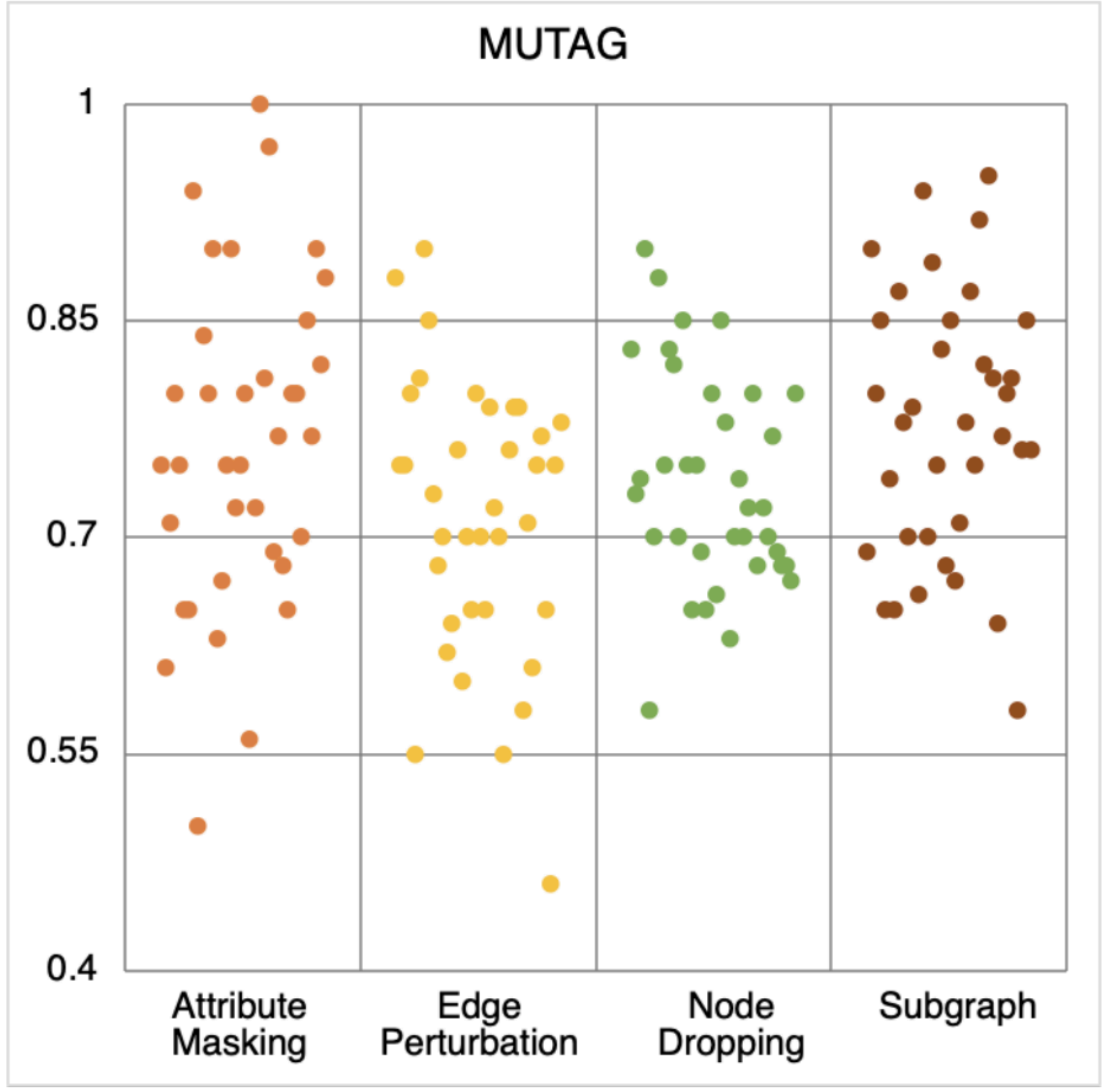
Subgraph performs stable in bioinformatics dataset

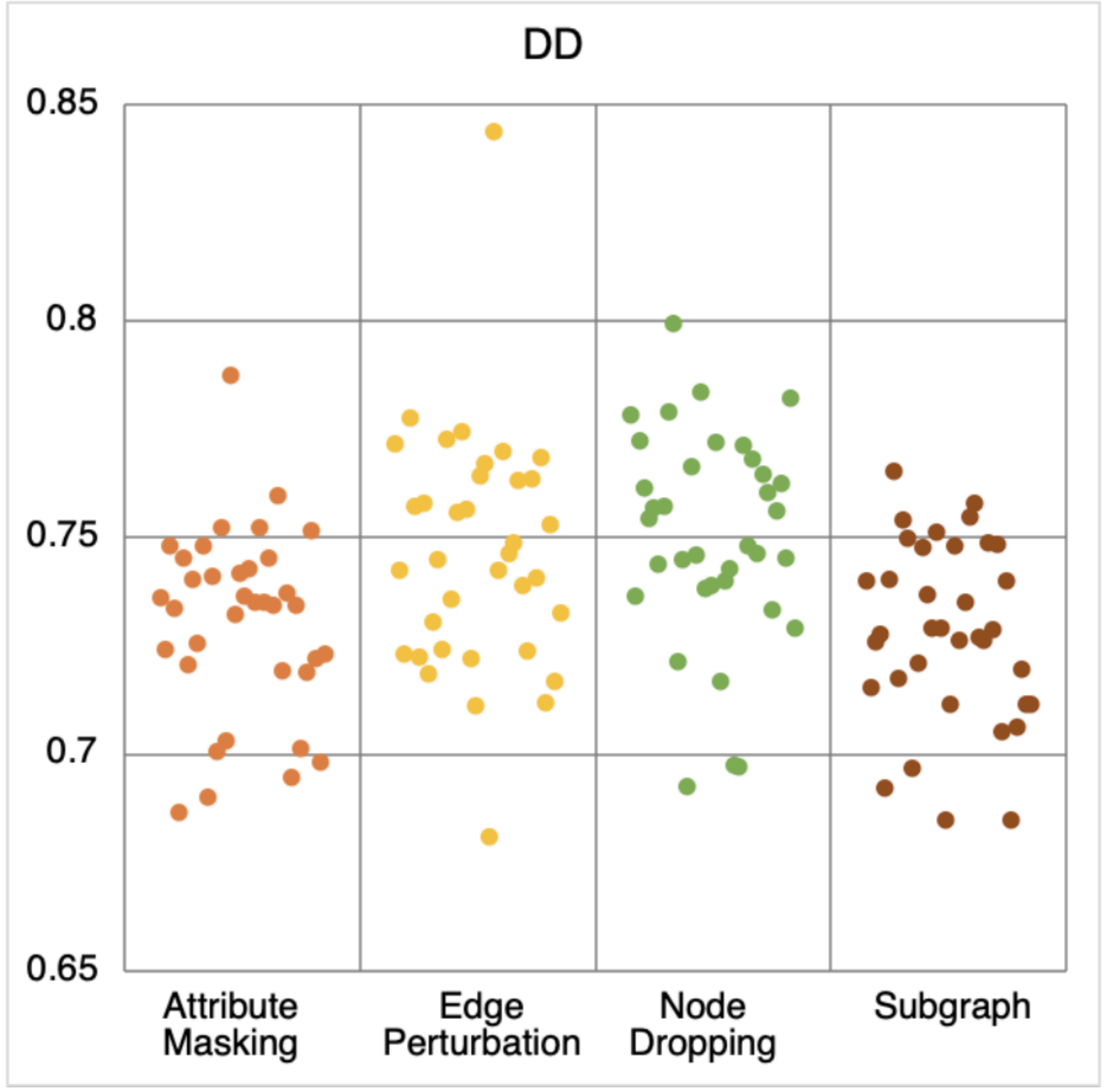
- **Target of MUTAG:**
    - to predict mutagenicity on Salmonella typhimurium.
- **Target of NCI1:**
    - to classify whether the chemical is positive or negative for cell lung cancer.
- **Target of PROTEINS and DD:**
    - to predict which proteins are enzymes or non-enzymes.

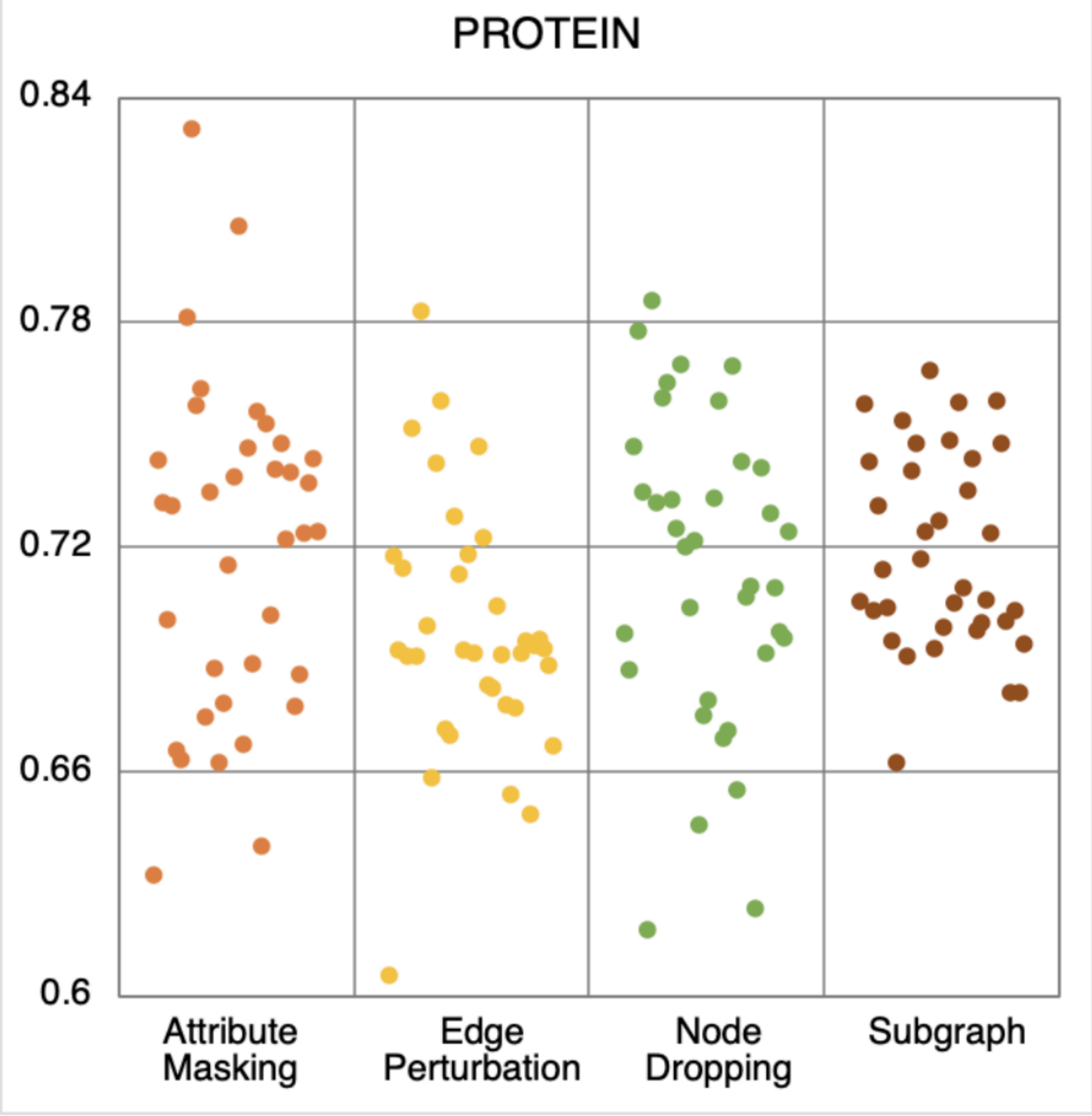
Subgraph performs stable in bioinformatics dataset

- Molecular: function group (e.g. C2H5OH)
- Bioinformatics/Protein: long polypeptide









Thank you for listening.
FAQ