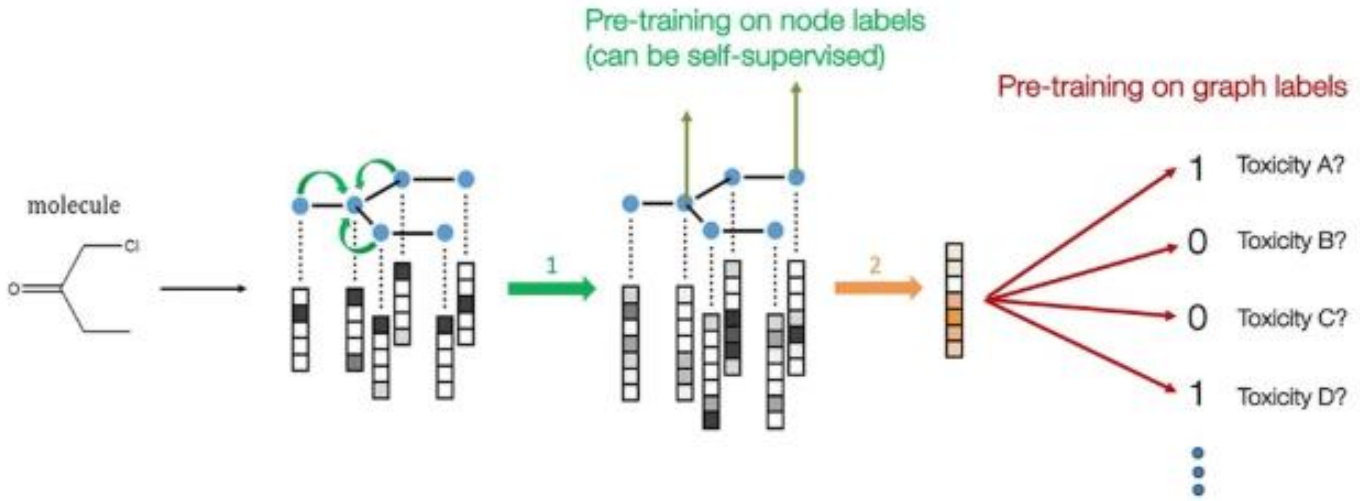
Strategies for Pre-training Graph Neural Networks

Jaewoong Shim

Seoul National University

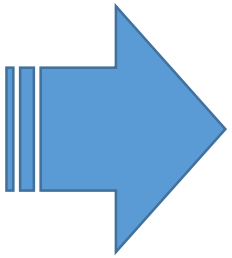


Hu, Weihua, et al. "Strategies for Pre-training Graph Neural Networks." *International Conference on Learning Representations*. 2020.

Pre-training GNN ?

problem

- Scarcity of labeled data
- Out-of-distribution prediction



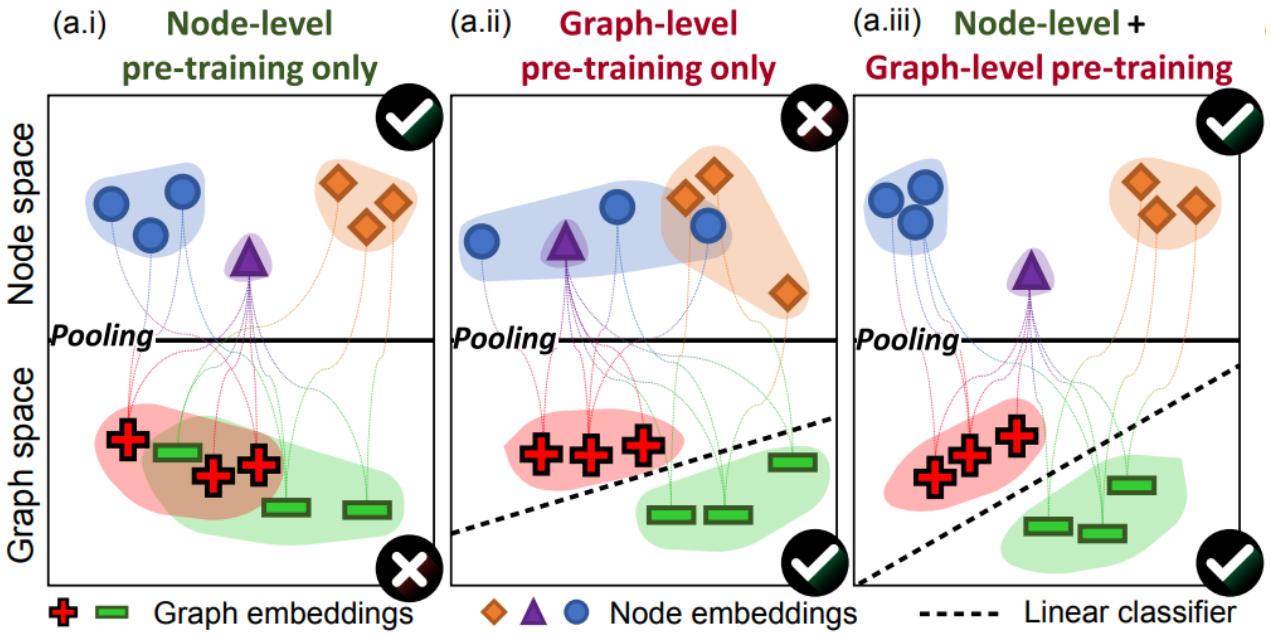
solution

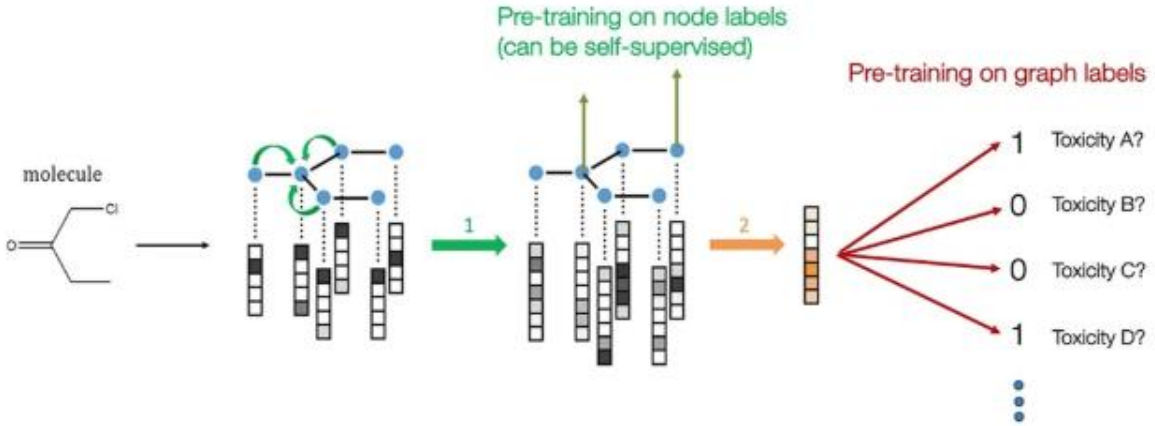
Pre-training a model on related tasks where data is abundant

- How to pre-train GNNs ?

Proposed strategies for pre-training GNN

- Pre-train both node and graph embeddings





Proposed strategies for pre-training GNN

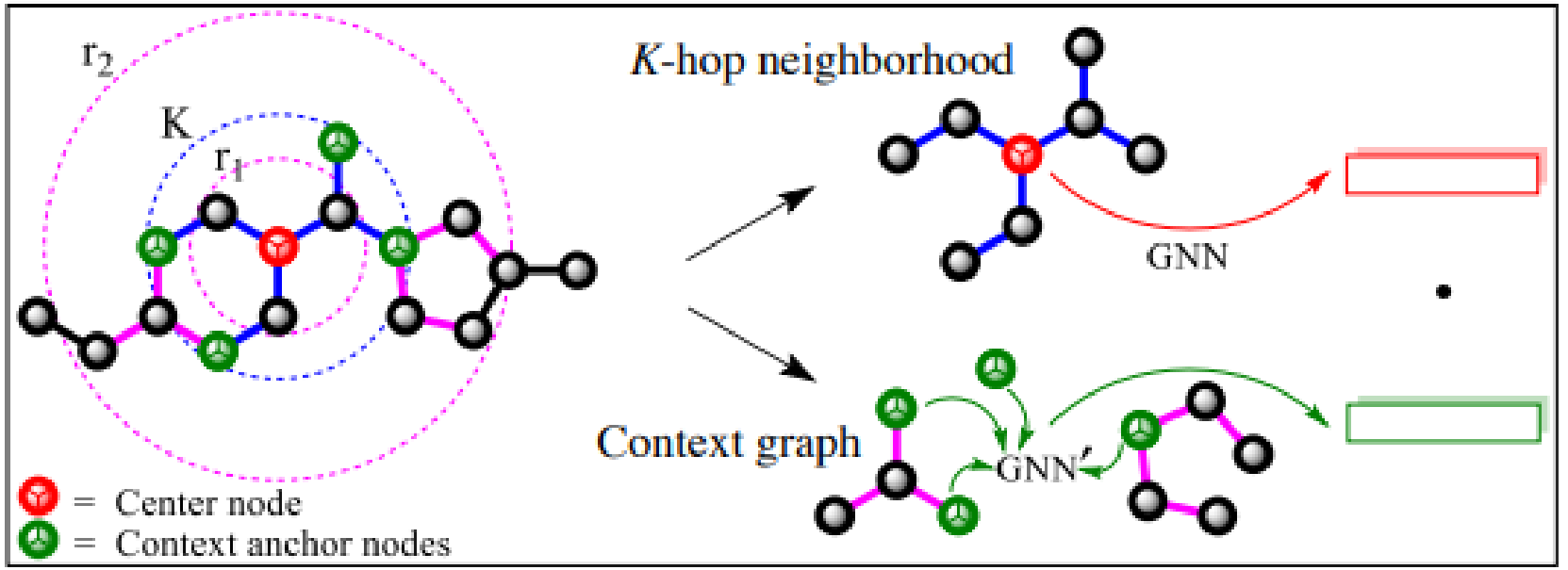
(b) Categorization of our pre-training methods

	Node-level	Graph-level
Attribute prediction	Attribute Masking	Supervised Attribute Prediction
Structure prediction	Context Prediction	Structural Similarity Prediction

Context Prediction

	Node-level	Graph-level
Attribute prediction	Attribute Masking	Supervised Attribute Prediction
Structure prediction	Context Prediction	Structural Similarity Prediction

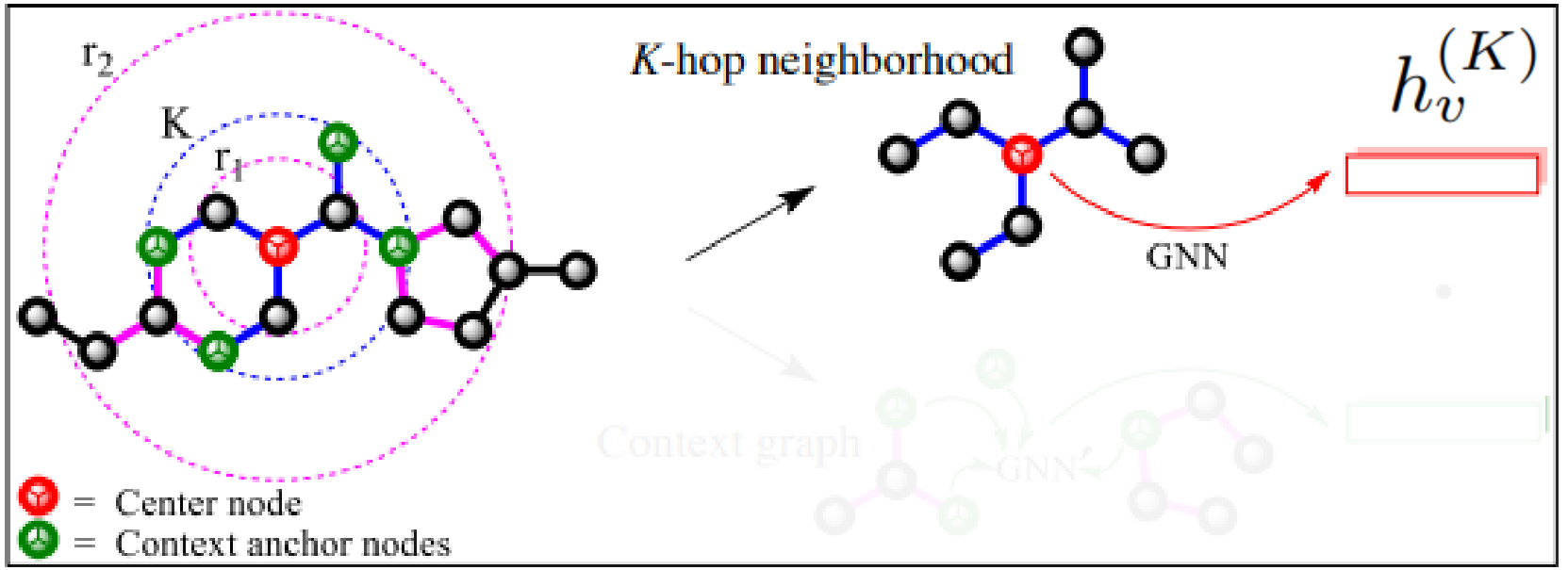
- 비슷한 structural context를 가진 node들은 비슷한 embedding을 갖도록
  - 2개의 GNN을 통해 학습을 수행



Context Prediction

	Node-level	Graph-level
Attribute prediction	Attribute Masking	Supervised Attribute Prediction
Structure prediction	Context Prediction	Structural Similarity Prediction

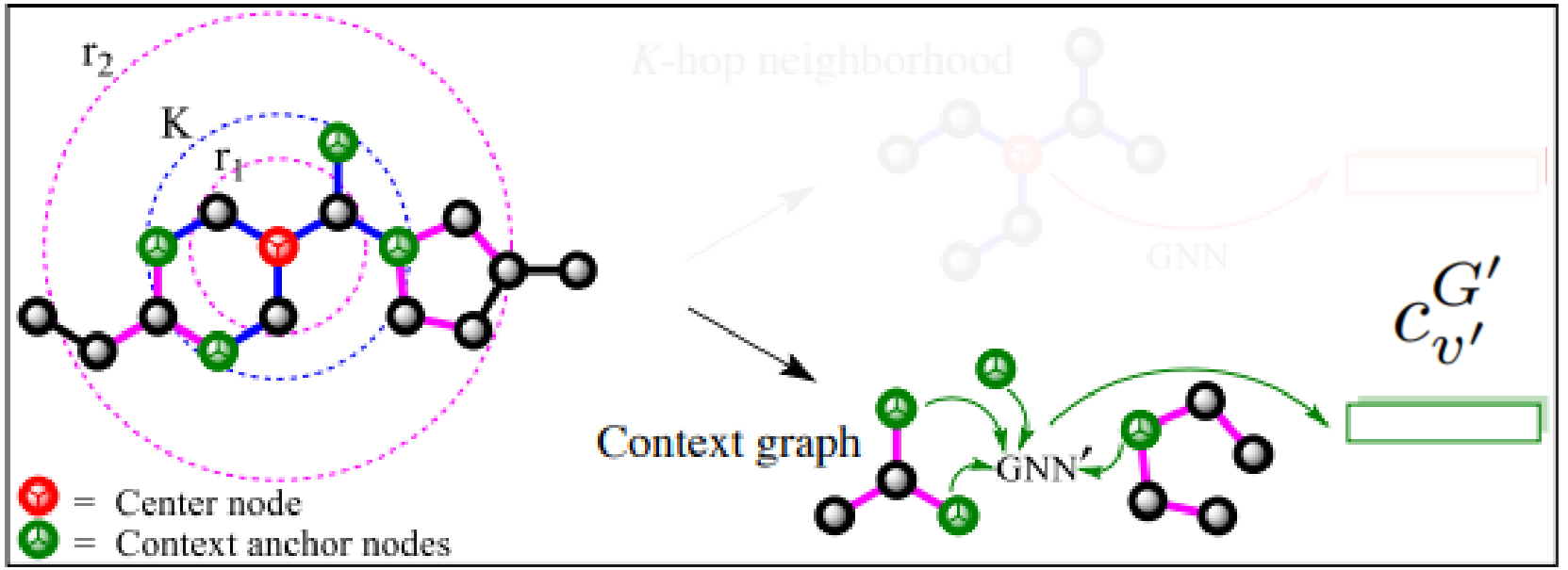
1. Main GNN (pre-trained model로 쓰임)
   - K-layer GNN을 통해 K-hop neighbor의 정보를 aggregate



Context Prediction

	Node-level	Graph-level
Attribute prediction	Attribute Masking	Supervised Attribute Prediction
Structure prediction	Context Prediction	Structural Similarity Prediction

2. Context GNN (auxiliary GNN)
   - Context graph of node v : r1-hop과 r2-hop의 사이에 있는 graph들
   - Context anchor node : context graph와 K-neighbor의 공통 node
   - Context graph $\xrightarrow[\text{Context GNN}]{}$ embeddings of context anchor nodes $\longrightarrow$ average

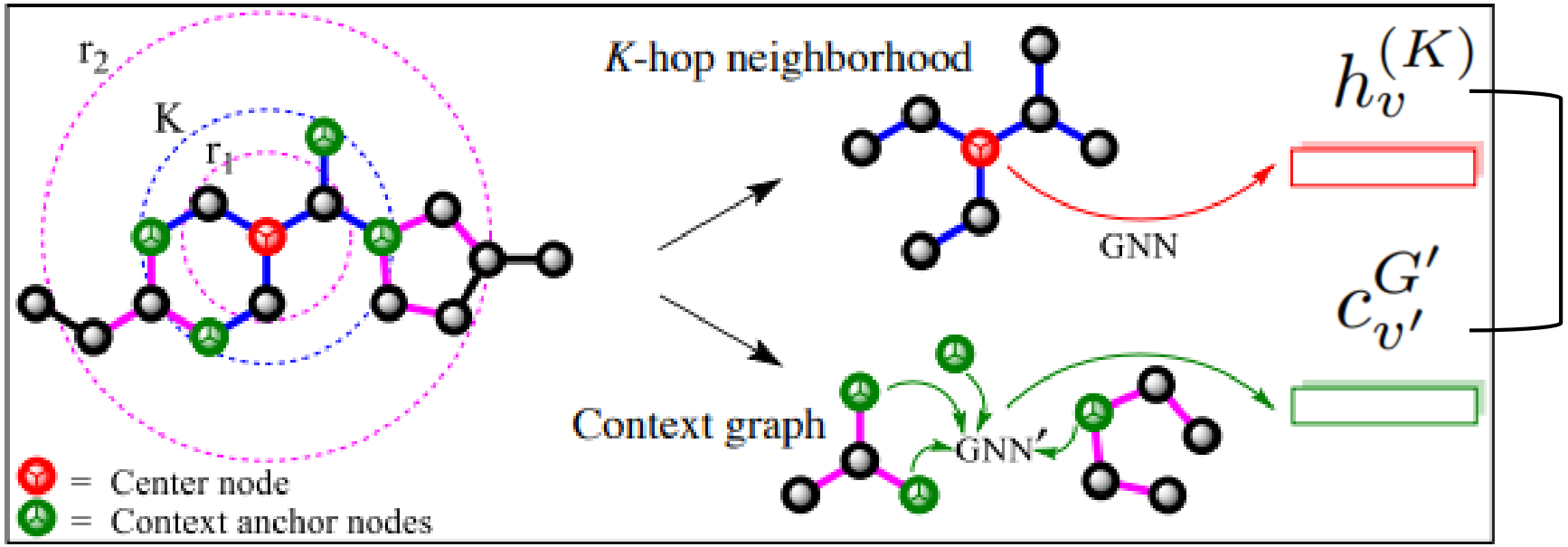


Context Prediction

	Node-level	Graph-level
Attribute prediction	Attribute Masking	Supervised Attribute Prediction
Structure prediction	Context Prediction	Structural Similarity Prediction

3. Learning via negative sampling
   - Jointly learn the main GNN and context GNN

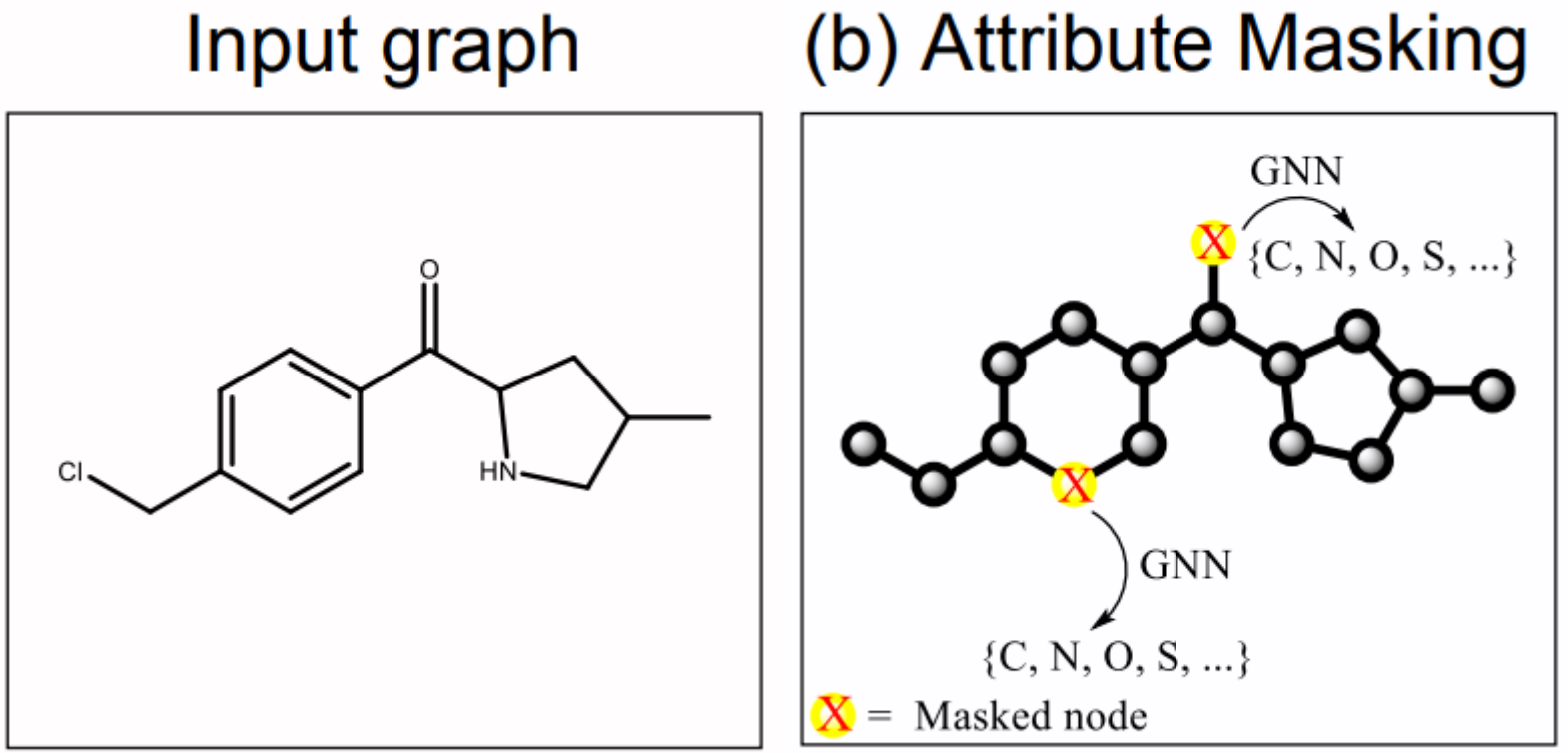
Learning objective : $\sigma\left(h_v^{(K)\top} c_{v'}^{G'}\right) \approx \mathbf{1}\{v \text{ and } v' \text{ are the same nodes}\}$



Attribute Masking

	Node-level	Graph-level
Attribute prediction	Attribute Masking	Supervised Attribute Prediction
Structure prediction	Context Prediction	Structural Similarity Prediction

- masked node(edge) attributes are predicted with GNN
  - Useful for richly-annotated graphs from scientific domains
    - Ex) Atom type in molecular graphs



Supervised Attribute prediction

	Node-level	Graph-level
Attribute prediction	Attribute Masking	Supervised Attribute Prediction
Structure prediction	Context Prediction	Structural Similarity Prediction

- Graph-level multi-task supervised pre-training
  - Jointly predict a diverse set of supervised labels of individual graphs
    - Ex) predict all the properties of molecules measured so far

Structural Similarity Prediction

	Node-level	Graph-level
Attribute prediction	Attribute Masking	Supervised Attribute Prediction
Structure prediction	Context Prediction	Structural Similarity Prediction

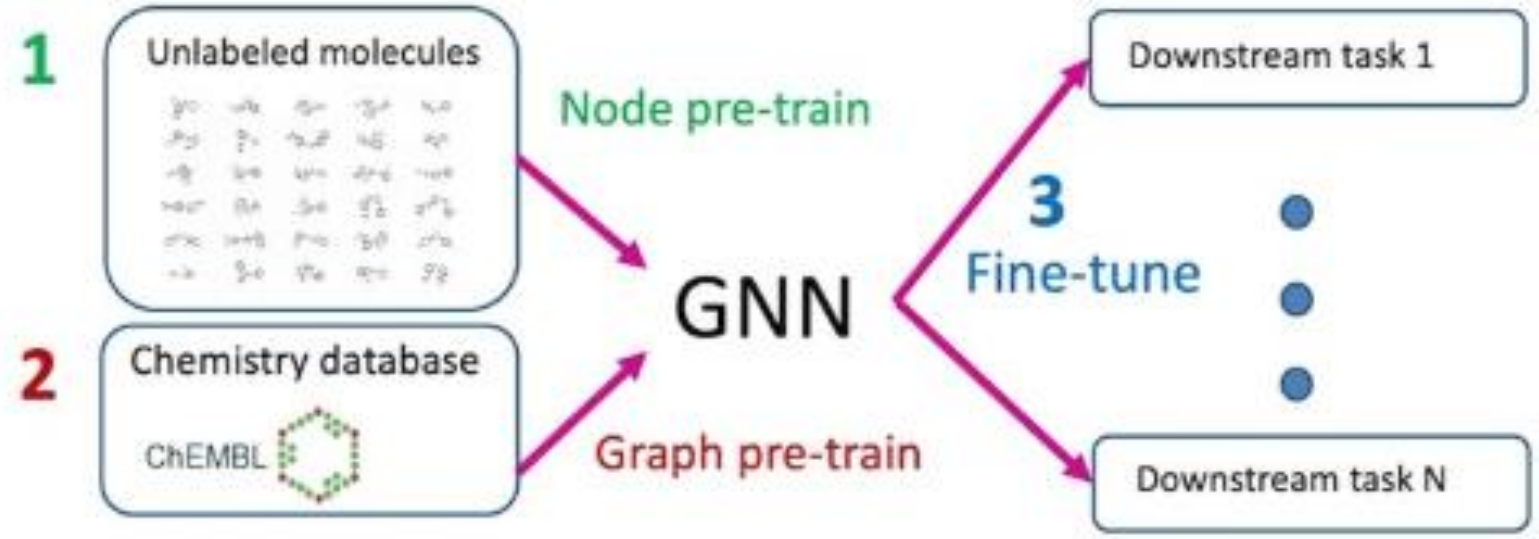
- Predict the structural similarity of two graphs
  - Ex) graph edit distance, graph structure similarity

- Ground truth graph distance is difficult to be found
- Quadratic number of graph pairs in large dataset

→ **Future work**

Overall Strategy

1. Node-level pre-training on unlabeled data
2. Graph-level pre-training on labeled data
3. Fine-tune on downstream data



Experiments and Results

	Chemistry			Biology		
	Non-pre-trained	Pre-trained	Gain	Non-pre-trained	Pre-trained	Gain
GIN	67.0	**74.2**	**+7.2**	64.8 ± 1.0	**74.2 ± 1.5**	**+9.4**
GCN	**68.9**	72.2	+3.4	63.2 ± 1.0	70.9 ± 1.7	+7.7
GraphSAGE	68.3	70.3	+2.0	65.7 ± 1.2	68.5 ± 1.5	+2.8
GAT	66.8	60.3	-6.5	**68.2 ± 1.1**	67.8 ± 3.6	-0.4

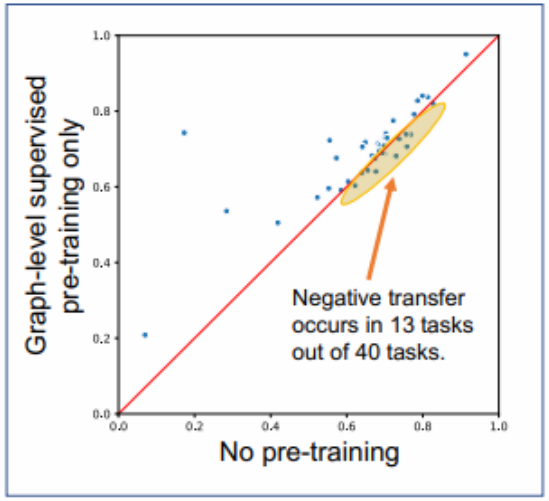
Table 2: **Test ROC-AUC (%) performance of different GNN architectures with and without pre-training.** Without pre-training, the less expressive GNNs give slightly better performance than the most expressive GIN because of their smaller model complexity in a low data regime. However, with pre-training, the most expressive GIN is properly regularized and dominates the other architectures. For results split by chemistry datasets, see Table 4 in Appendix H. Pre-training strategy for chemistry data: Context Prediction + Graph-level supervised pre-training; pre-training strategy for biology data: Attribute Masking + Graph-level supervised pre-training.

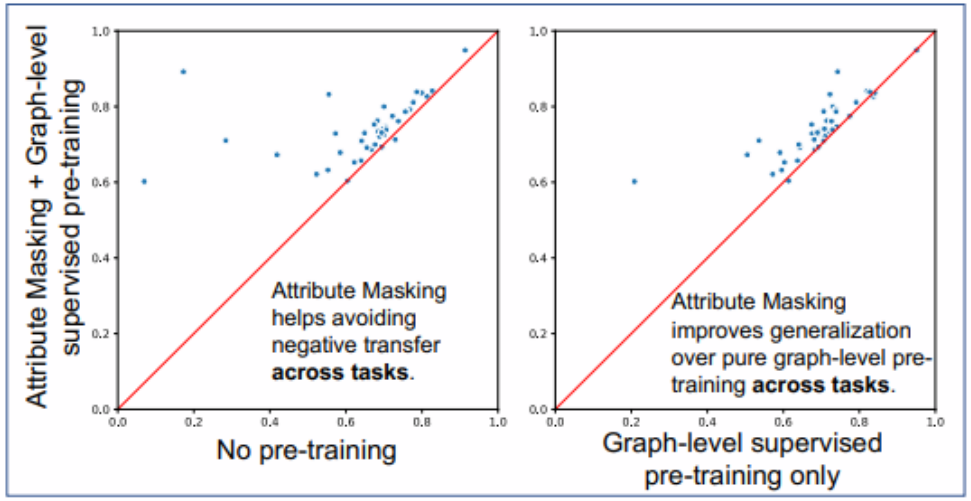
Experiments and Results

Dataset	BBBP	Tox21	ToxCast	SIDER	ClinTox	MUV	HIV	BACE	**Average**
# Molecules	2039	7831	8575	1427	1478	93087	41127	1513	/
# Binary prediction tasks	1	12	617	27	2	17	1	1	/

Pre-training strategy		Out-of-distribution prediction (scaffold split)								
Graph-level	Node-level									
–	–	65.8 ±4.5	74.0 ±0.8	63.4 ±0.6	57.3 ±1.6	58.0 ±4.4	71.8 ±2.5	75.3 ±1.9	70.1 ±5.4	67.0
–	Infomax	**68.8 ±0.8**	75.3 ±0.5	62.7 ±0.4	58.4 ±0.8	69.9 ±3.0	75.3 ±2.5	76.0 ±0.7	75.9 ±1.6	70.3
–	EdgePred	67.3 ±2.4	76.0 ±0.6	64.1 ±0.6	60.4 ±0.7	64.1 ±3.7	74.1 ±2.1	76.3 ±1.0	79.9 ±0.9	70.3
–	AttrMasking	64.3 ±2.8	76.7 ±0.4	64.2 ±0.5	61.0 ±0.7	71.8 ±4.1	74.7 ±1.4	77.2 ±1.1	79.3 ±1.6	71.1
–	ContextPred	68.0 ±2.0	75.7 ±0.7	63.9 ±0.6	60.9 ±0.6	65.9 ±3.8	75.8 ±1.7	77.3 ±1.0	79.6 ±1.2	70.9
Supervised	–	68.3 ±0.7	77.0 ±0.3	64.4 ±0.4	62.1 ±0.5	57.2 ±2.5	79.4 ±1.3	74.4 ±1.2	76.9 ±1.0	70.0
Supervised	Infomax	68.0 ±1.8	77.8 ±0.3	64.9 ±0.7	60.9 ±0.6	**71.2 ±2.8**	**81.3 ±1.4**	77.8 ±0.9	80.1 ±0.9	72.8
Supervised	EdgePred	66.6 ±2.2	**78.3 ±0.3**	**66.5 ±0.3**	**63.3 ±0.9**	70.9 ±4.6	78.5 ±2.4	77.5 ±0.8	79.1 ±3.7	72.6
Supervised	AttrMasking	66.5 ±2.5	77.9 ±0.4	65.1 ±0.3	**63.9 ±0.9**	**73.7 ±2.8**	**81.2 ±1.9**	77.1 ±1.2	80.3 ±0.9	73.2
Supervised	ContextPred	**68.7 ±1.3**	**78.1 ±0.6**	65.7 ±0.6	62.7 ±0.8	**72.6 ±1.5**	**81.3 ±2.1**	**79.9 ±0.7**	**84.5 ±0.7**	**74.2**

Pre-training strategy		Out-of-dist.
Graph-level	Node-level	(species split)
–	–	64.8 ±1.0
–	Infomax	64.1 ±1.5
–	EdgePred	65.7 ±1.3
–	ContextPred	65.2 ±1.6
–	AttrMasking	64.4 ±1.3
Supervised	–	69.0 ±2.4
Supervised	Infomax	72.8 ±1.5
Supervised	EdgePred	72.3 ±1.4
Supervised	ContextPred	73.8 ± 1.0
Supervised	AttrMasking	**74.2 ±1.5**





Experiments and Results

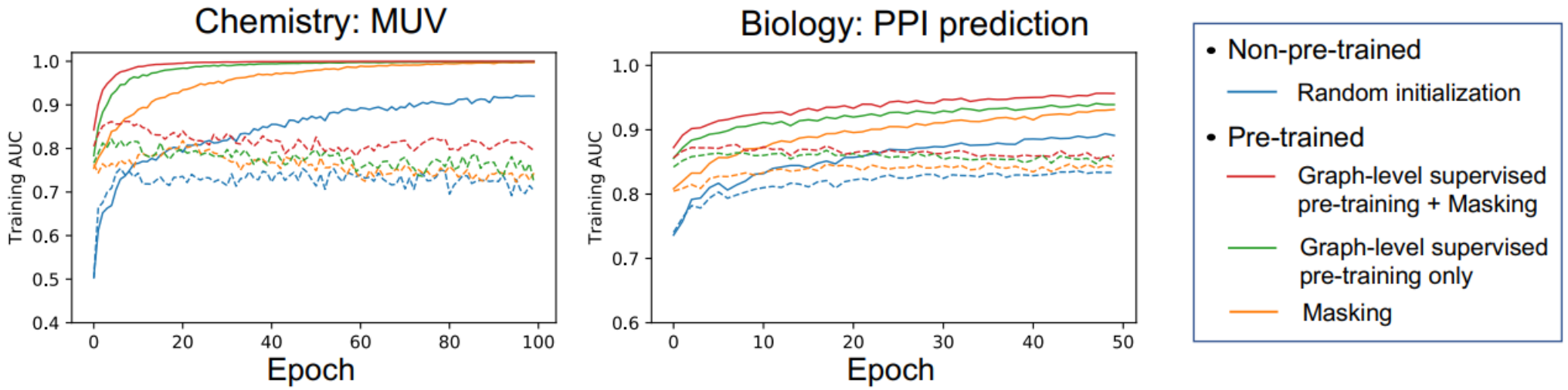


Figure 4: **Training and validation curves of different pre-training strategies on GINs.** Solid and dashed lines indicate training and validation curves, respectively.

Conclusions

- Systematic study of pre-training GNNs
- Develop a novel strategy for pre-training GNNs
- Consider both node-level and graph-level pre-training with an expressive GNN.

Reproducing

코드가 공개된 논문입니다.

https://github.com/snap-stanford/pretrain-gnns

코드 실행 구조

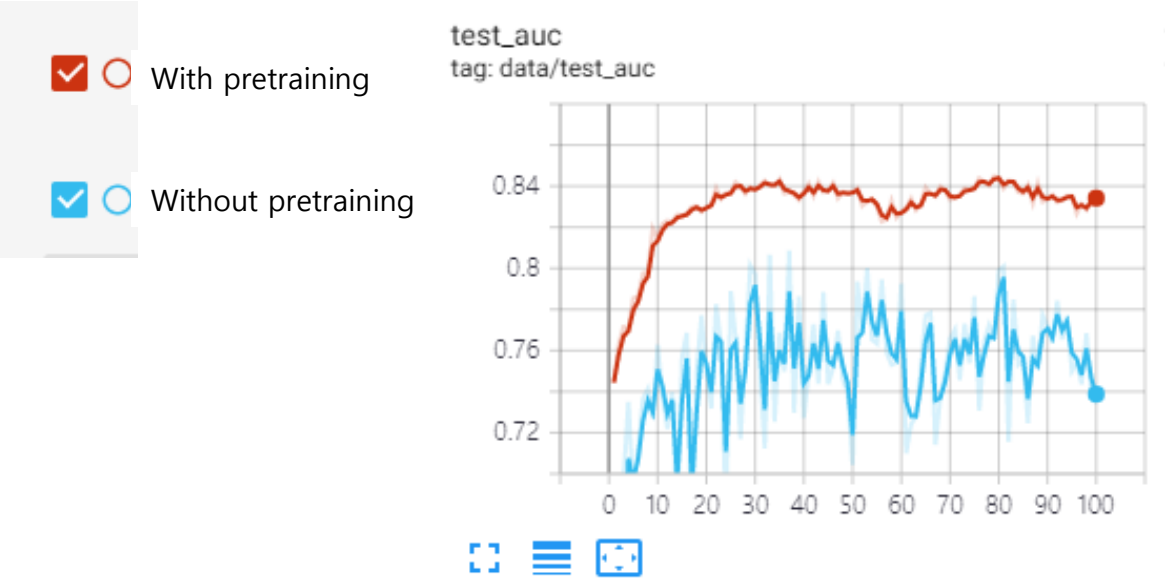
1. Self-supervised pre-training
    - Pretrain_masking.py
    - Pretrain_contextpred.py
    - Pretrain_edgepred.py
    - Pretrain_deepgraphinfomax.py

2. Supervised pre-training
    - Pretrain_supervised.py

3. Fine-tuning
    - Finetune.py

코드 실행 예시

- Pretraining 방법
  - Context prediction + supervised pretraining
- 데이터셋 : BACE 데이터 셋
  - **BACE.** Qualitative binding results for a set of inhibitors of human β-secretase 1 (Subramanian et al., 2016).
- Finetuning 시행




```
(pretraingnn) woong@ubuntu:~/pretraingnn/pretrain-gnns/chem$ sh finetune_tune_woong.sh 1 0
MoleculeDataset(1513)
scaffold
Data(edge_attr=[66, 2], edge_index=[2, 66], fold=[1], id=[1], x=[31, 2], y=[1])
Adam (
Parameter Group 0
    amsgrad: False
    betas: (0.9, 0.999)
    eps: 1e-08
    lr: 0.001
    weight_decay: 0

Parameter Group 1
    amsgrad: False
    betas: (0.9, 0.999)
    eps: 1e-08
    lr: 0.001
    weight_decay: 0
)
====epoch 1
Iteration: 100%|| 38/38 [00:01<00:00, 21.95it/s]
====Evaluation
omit the training accuracy computation
Iteration: 100%|| 5/5 [00:00<00:00, 24.16it/s]
Iteration: 100%|| 5/5 [00:00<00:00, 20.63it/s]
train: 0.000000 val: 0.692674 test: 0.744392
```

```
====epoch 99
Iteration: 100%|| 38/38 [00:01<00:00, 23.54it/s]
====Evaluation
omit the training accuracy computation
Iteration: 100%|| 5/5 [00:00<00:00, 22.06it/s]
Iteration: 100%|| 5/5 [00:00<00:00, 22.68it/s]
train: 0.000000 val: 0.695604 test: 0.834290

====epoch 100
Iteration: 100%|| 38/38 [00:01<00:00, 32.81it/s]
====Evaluation
omit the training accuracy computation
Iteration: 100%|| 5/5 [00:00<00:00, 21.97it/s]
Iteration: 100%|| 5/5 [00:00<00:00, 23.23it/s]
train: 0.000000 val: 0.696703 test: 0.835159
```