



# 그래프 신경망 기반 화학/소재분야 연구 동향

(Out-of-Distribution 문제를 중심으로)

박찬영

**Assistant Professor, KAIST**

Industrial and Systems Engineering  
Graduate School of Data Science  
Graduate School of AI

cy.park@kaist.ac.kr

# BRIEF BIO



**Chanyoung Park, Ph.D.**

**Assistant Professor**

ISYSE KAIST

GSDS/GSAI KAIST

## Contact Information

- [cy.park@kaist.ac.kr](mailto:cy.park@kaist.ac.kr)
- <http://dsail.kaist.ac.kr/>

## ■ Research Interest

- **Multimodal Data Mining, Applied Machine Learning, Deep Learning**
  - *Mining meaningful knowledge from multimodal data to develop artificial intelligence solutions for various real-world applications across different disciplines*
  - Keywords: Multimodal user behavior analysis, Machine learning for graphs, Graph neural network, Graph representation learning
  - **Application domains:** Recommendation system, Social network analysis, Fraud detection, Sentiment analysis, Purchase/Click prediction, Anomaly detection, Knowledge-graph construction, Time-series analysis, Bioinformatics, Chemistry etc.

## ■ Professional Experience

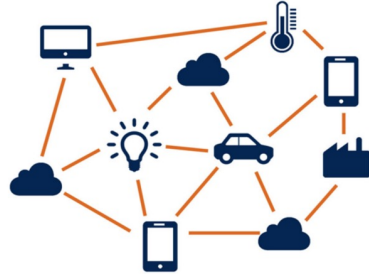
- Assistant Professor, **KAIST** (2020.11 – Present)
- Postdoctoral Research Fellow, **University of Illinois at Urbana-Champaign**, Dept. of Computer Science (2019. 1 – 2020. 10)
- Research Intern, **Microsoft Research Asia** (2017. 9 – 2017. 12)
- Research Intern, **NAVER** (2017. 3 – 2017. 6)



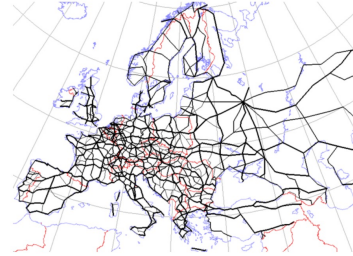
# Research area Graphs are everywhere!



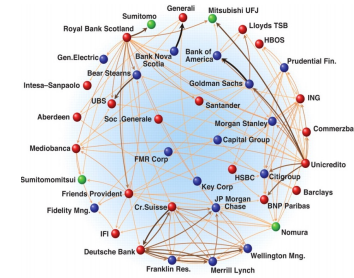
Social Network



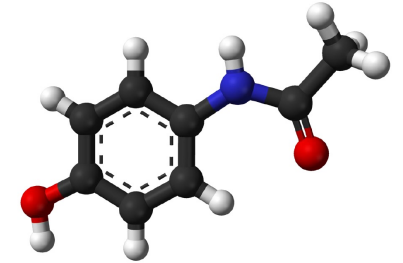
Internet of Things



Road Graph



Financial Graph

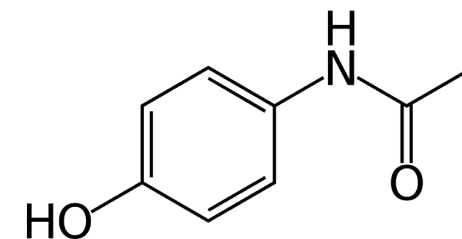
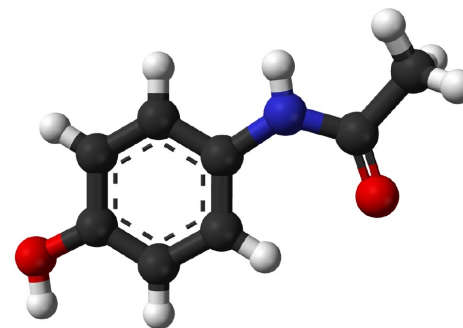


Molecular Graph

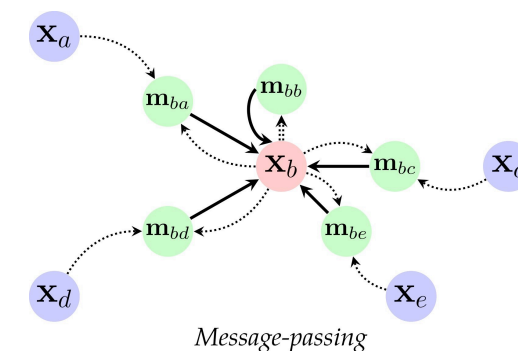
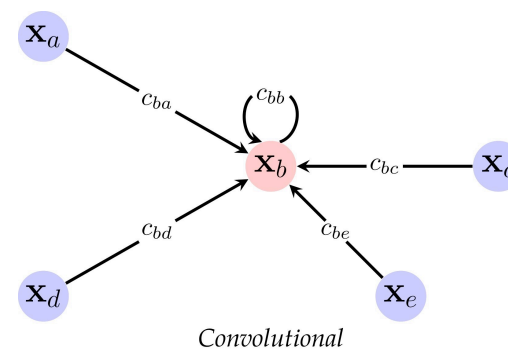
- Many problems in our real-life can be modeled as machine learning tasks **over large graphs**
- Our goal is to use graph as a tool for **solving real-world problems** by applying **graph mining techniques**

# Introduction

- A molecule can be represented as a graph
  - Atom in a molecule: Node in a graph
  - Bond in a molecule: Edge in a graph
- Graph machine learning is widely being applied to chemistry / materials science



- **Graph Neural Network** learns how to propagate messages between nodes
  - Variants of GNNs
    - Graph Convolutional Networks
    - Graph Attention Networks
    - Message Passing Neural Network

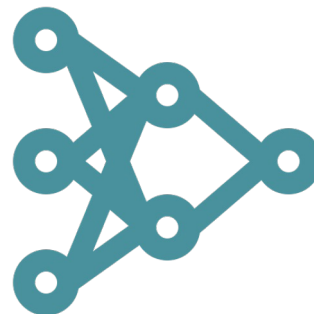
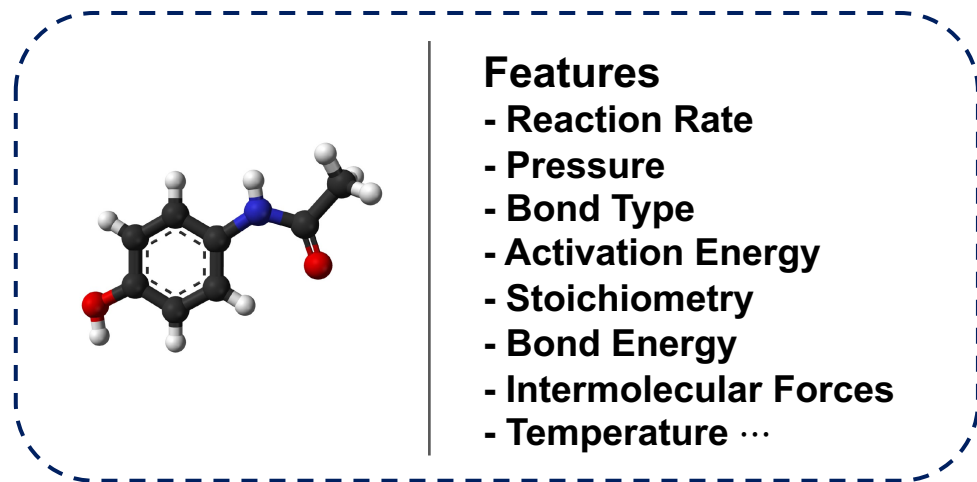


# Outline

- 그래프 신경망 개요 (20 mins)
  - 그래프 신경망 전반적인 소개
  - 그래프 종류에 따른 다양한 그래프 신경망 소개
- How to address Out-of-distribution problem (세부 기술 및 Q&A) (90~120 mins)
  - 소재 물성 예측 연구
    - 소재 물성 예측 연구 최신 동향 소개
    - Transformer 기반 모델 소개 → Prompt-based method
    - Extrapolation을 위한 모델 소개 → Nonlinearity encoding-based method
  - 물질 간 화학 반응 예측 연구
    - 물질 간 화학 반응 예측 연구 동향 소개
    - 정보 이론(Information bottleneck) 기반 모델 소개 → Information bottleneck-based method
    - 인과추론(Causal inference) 기반 모델 소개 → Causal inference-based method

# Introduction: Molecular Property Prediction

- Predict the properties of a molecule (소재 물성 예측)



Graph Neural Network



**Prediction**  
ex) Band gap, DOS, Fermi

# Introduction: Molecular Relational Learning

- Learn the interaction behavior between a pair of molecules (물질 간 화학 반응 예측)



- Examples

- Predicting **optical properties** when a chromophore (Chromophore) and solvent (Solvent) react
- Predicting **solubility** when a solute and solvent react
- Predicting **side effects** when taking two types of drugs simultaneously (Polypharmacy effect)

# Outline

- 그래프 신경망 개요 (20 mins)
  - 그래프 신경망 전반적인 소개
  - 그래프 종류에 따른 다양한 그래프 신경망 소개
- How to address Out-of-distribution problem (세부 기술 및 Q&A) (90~120 mins)
  - 소재 물성 예측 연구
    - 소재 물성 예측 연구 최신 동향 소개
    - Transformer 기반 모델 소개 → Prompt-based method
    - Extrapolation을 위한 모델 소개 → Nonlinearity encoding-based method
  - 물질 간 화학 반응 예측 연구
    - 물질 간 화학 반응 예측 연구 동향 소개
    - 정보 이론(Information bottleneck) 기반 모델 소개 → Information bottleneck-based method
    - 인과추론(Causal inference) 기반 모델 소개 → Causal inference-based method

# 그래프 신경망 개요

# Outline

- Overview
- Graph Neural Network (GNN)
  - Graph Convolutional Neural Network (GCN)
  - Graph Attention Network (GAT)
  - Relational GCN
  - GraphSAGE



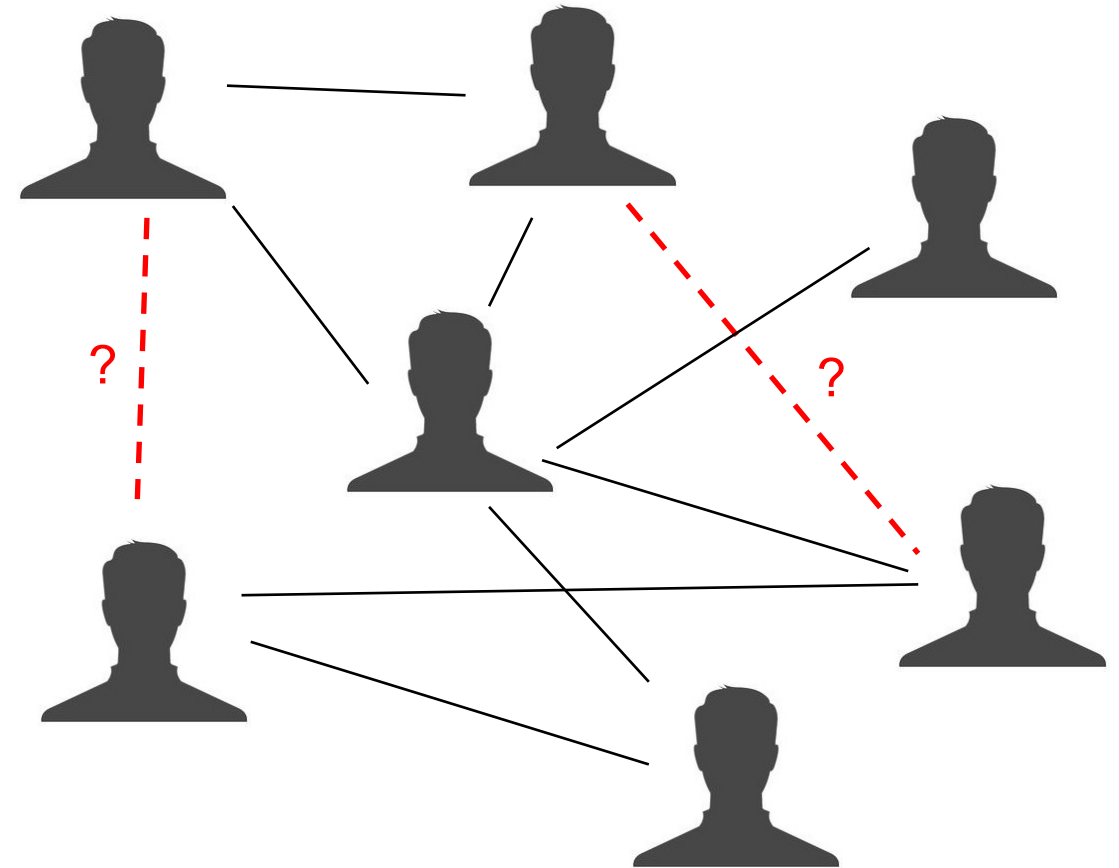
# Outline

- Overview
- Graph Neural Network (GNN)
  - Graph Convolutional Neural Network (GCN)
  - Graph Attention Network (GAT)
  - Relational GCN
  - GraphSAGE

# Machine learning on graphs

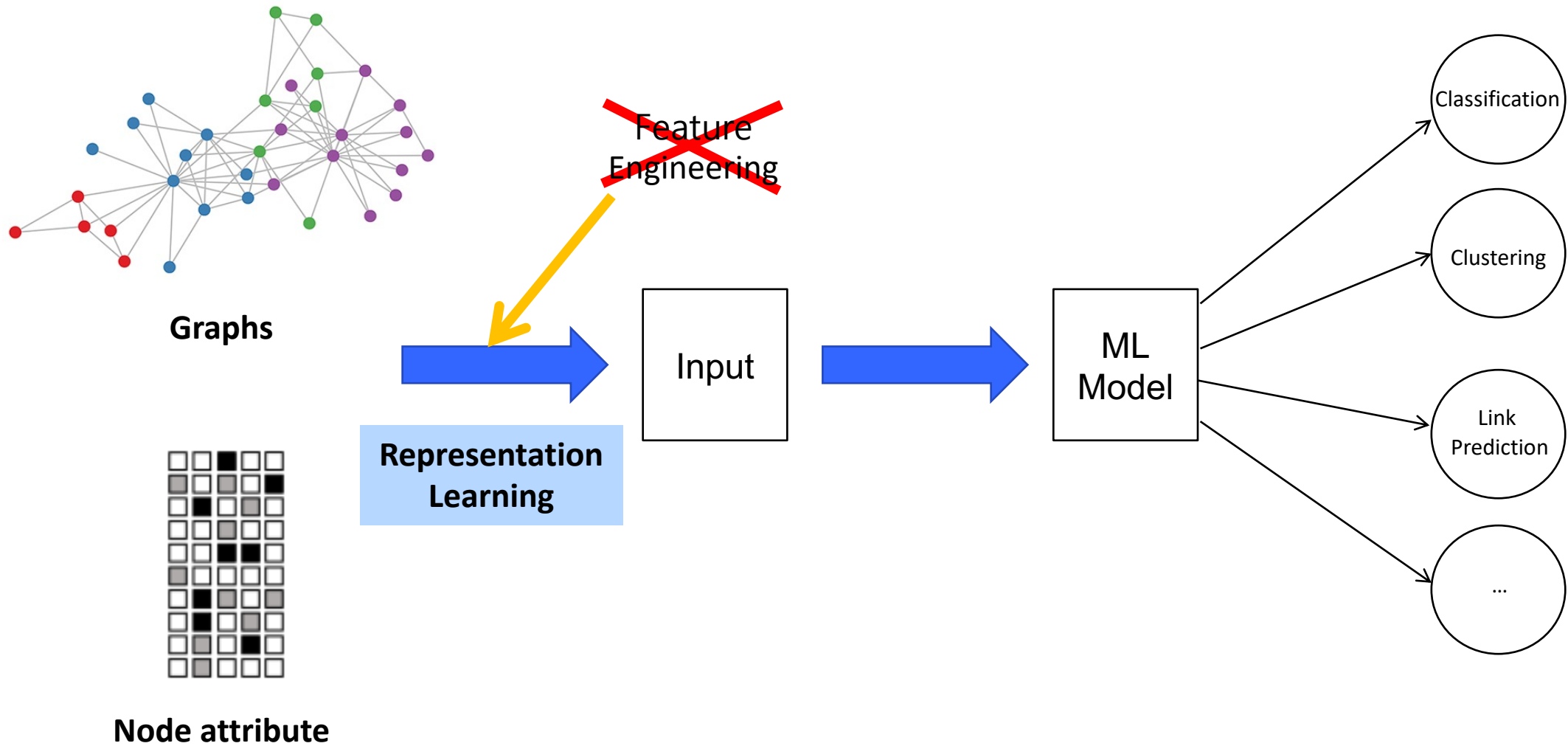
## Classical ML tasks in graphs:

- Node classification
  - Predict a type of a given node
- Link prediction
  - Predict whether two nodes are linked
- Community detection
  - Identify densely linked clusters of nodes
- Network similarity
  - How similar are two (sub)networks



**Link Prediction  
(Friend Recommendation)**

# Machine learning on graphs



# Machine learning in general

- Machine Learning = **Representation** + Objective + Optimization



Raw data



Representation  
Learning

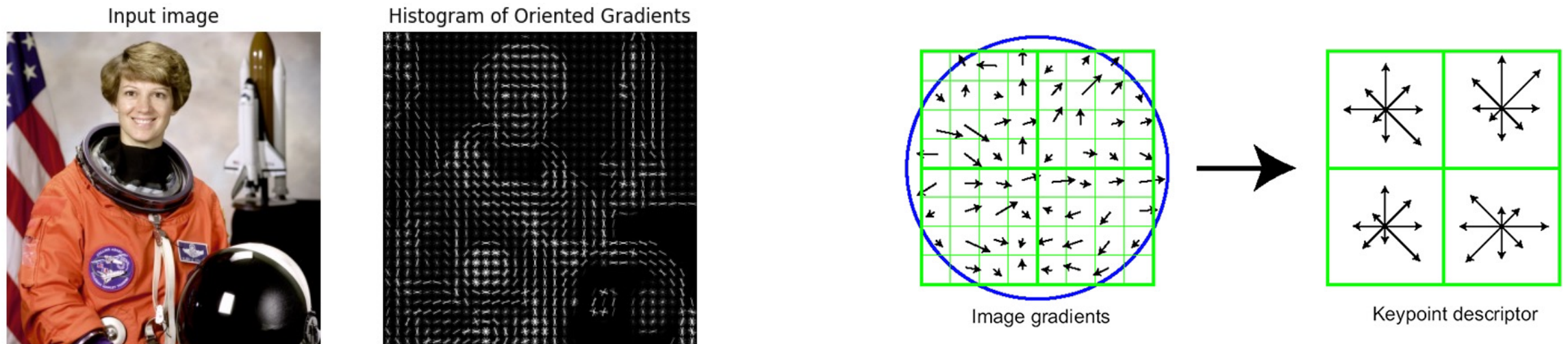
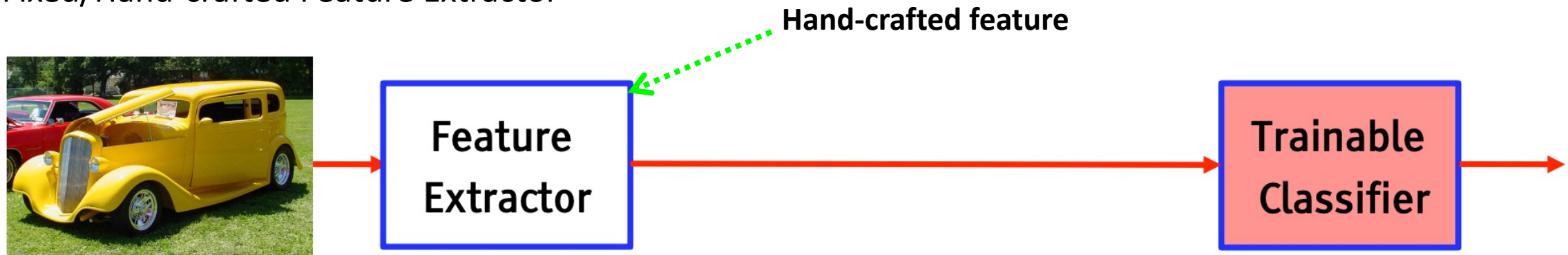


Machine Learning  
System

Good **Representation** is Essential for  
Good **Machine Learning**

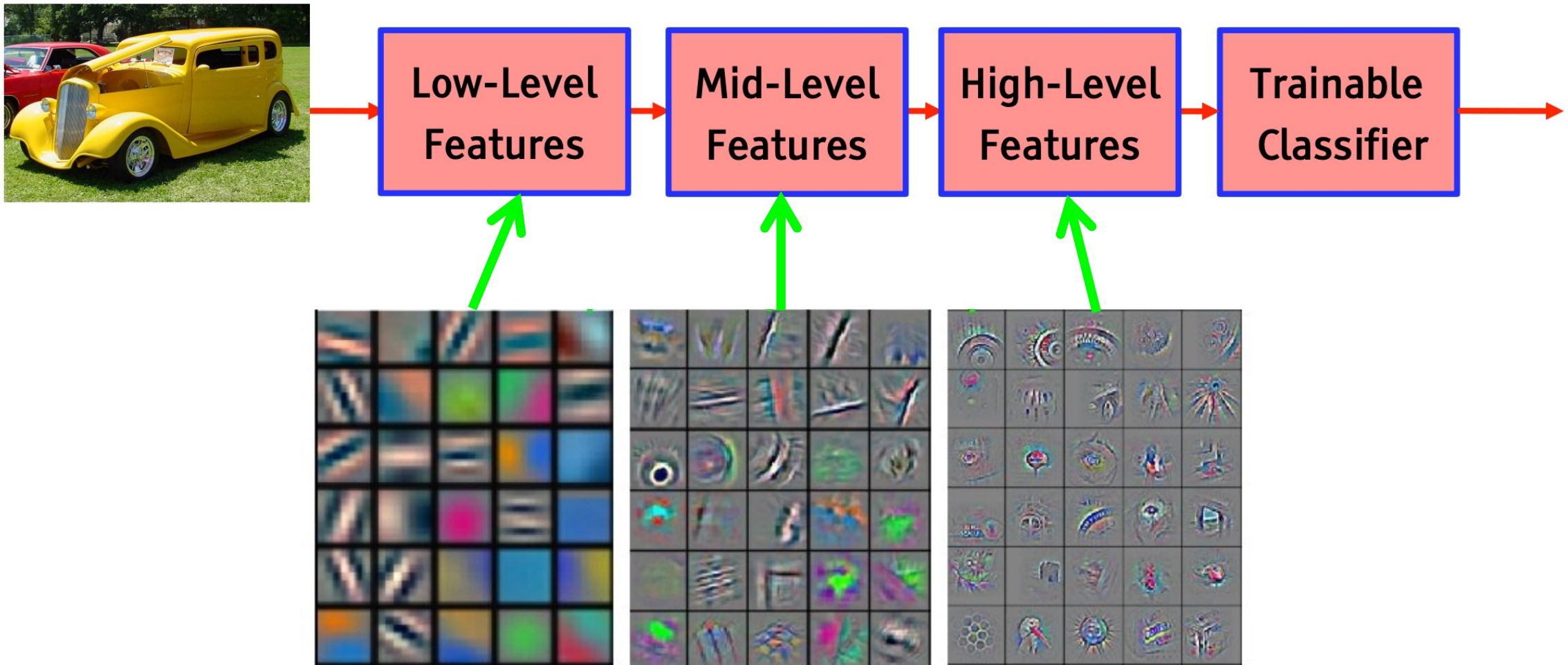
# Traditional feature extraction for images

- Fixed/Hand-crafted Feature Extractor

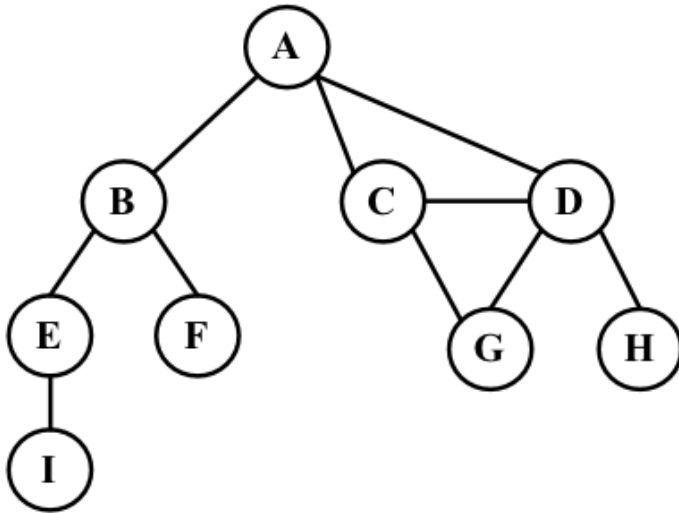


# Machine (Deep) learning based representation learning

- Multiple layers trained end-to-end



# Traditional graph representation



	A	B	C	D	E	F	G	H	I
A	0	1	1	1	0	0	0	0	0
B	1	0	0	0	1	1	0	0	0
C	1	0	0	1	0	0	1	0	0
D	1	0	1	0	0	0	1	1	0
E	0	1	0	0	0	0	0	0	1
F	0	1	0	0	0	0	0	0	0
G	0	0	1	1	0	0	0	0	0
H	0	0	0	1	0	0	0	0	0
I	0	0	0	0	1	0	0	0	0

Adjacency matrix

## Problems

- Suffer from data sparsity
- Suffer from high dimensionality
- High complexity for computation
- Does not represent “semantics”
- ...

**How to effectively and efficiently represent graphs is the key!**

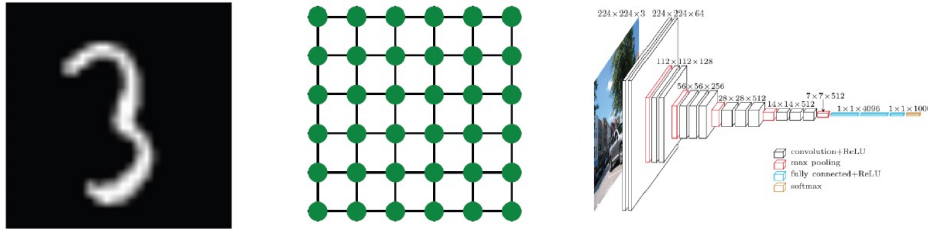
**→ Deep learning-based approach?**



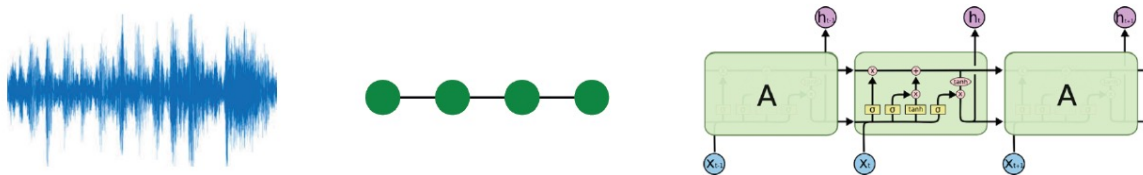
# Challenges of graph representation learning

- Existing deep neural networks are designed for data with regular-structure (grid or sequence)

- CNNs for fixed-size images/grids ...



- RNNs for text/sequences ...



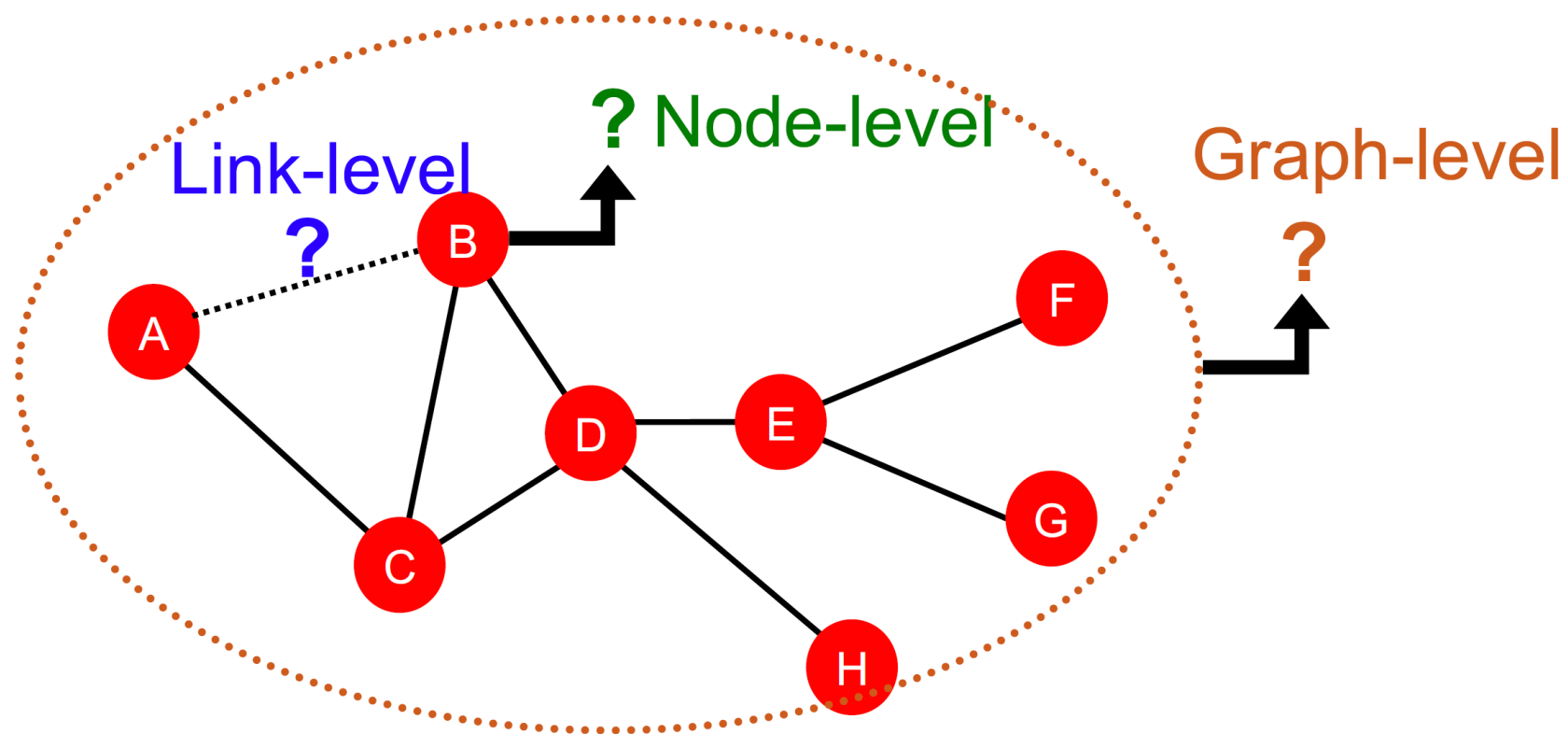
- Graphs are very complex**

- Arbitrary structures (no spatial locality like grids / no fixed orderings)
- Heterogeneous: Directed/undirected, binary/weighted/typed, multimodal features
- Large-scale: More than millions of nodes and billions of edges



# Typical tasks

- **Node-level** prediction
- **Edge-level** prediction
- **Graph-level** prediction



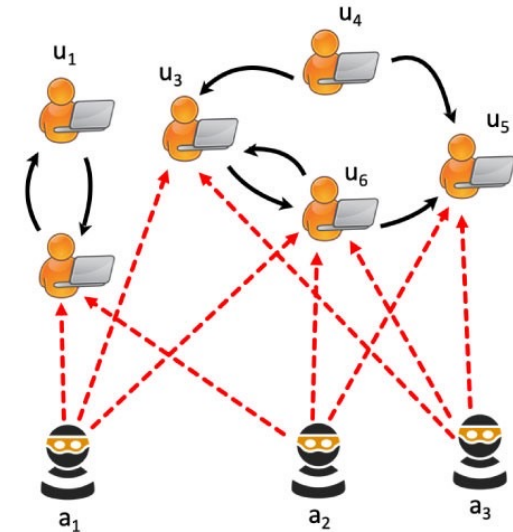
# Typical tasks

## ▪ Node-level tasks (or edge-level tasks)

- Node label classification, including node-level anomaly detection
- Node label regression
- Link label binary classification, i.e., link prediction
- Link label multi-class classification, i.e., relation classification

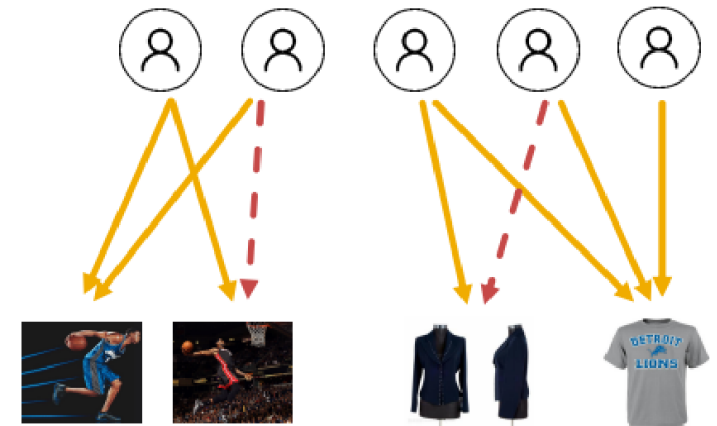


- Social network analysis (e.g., demographic info prediction)
- Spam / fraud detection (e.g., transaction networks)
- Link prediction (e.g., social networks, chemical interaction networks, biological networks, transportation networks)
- Knowledge graph population / completion / relation reasoning
- Recommender system (bipartite graphs, hyper-graphs)



Users

Items



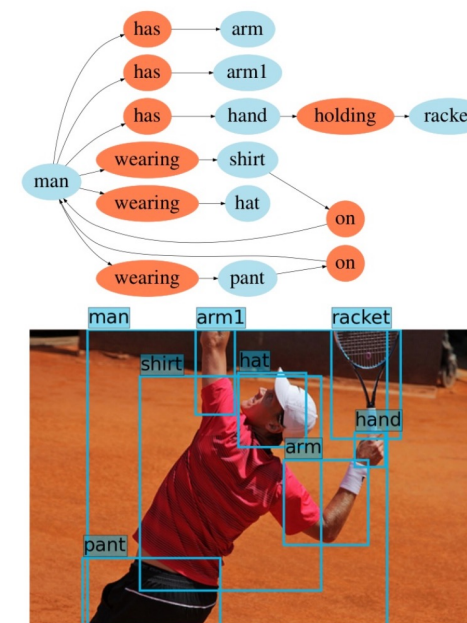
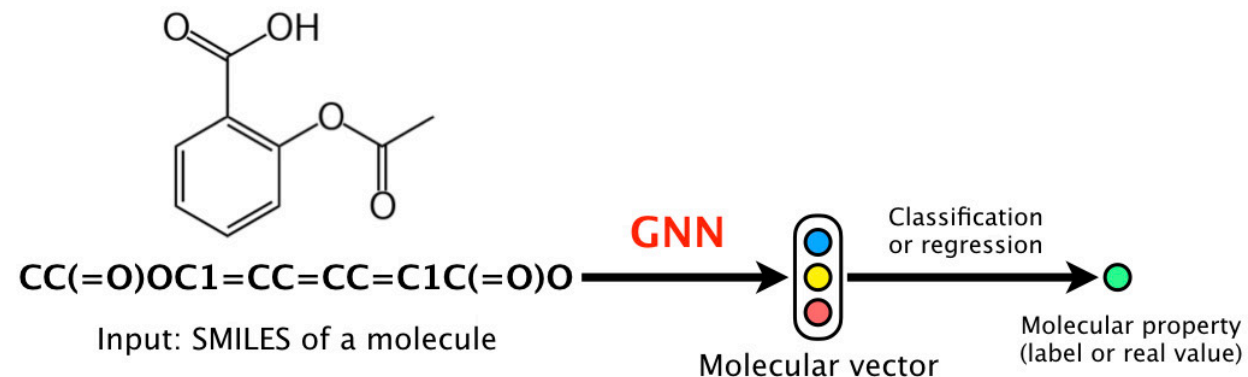
# Typical tasks

## ■ Graph-level tasks

- Graph label classification
- Graph label regression



- Molecular property prediction
- Drug discovery
- Scene understanding (i.e., objects graph)

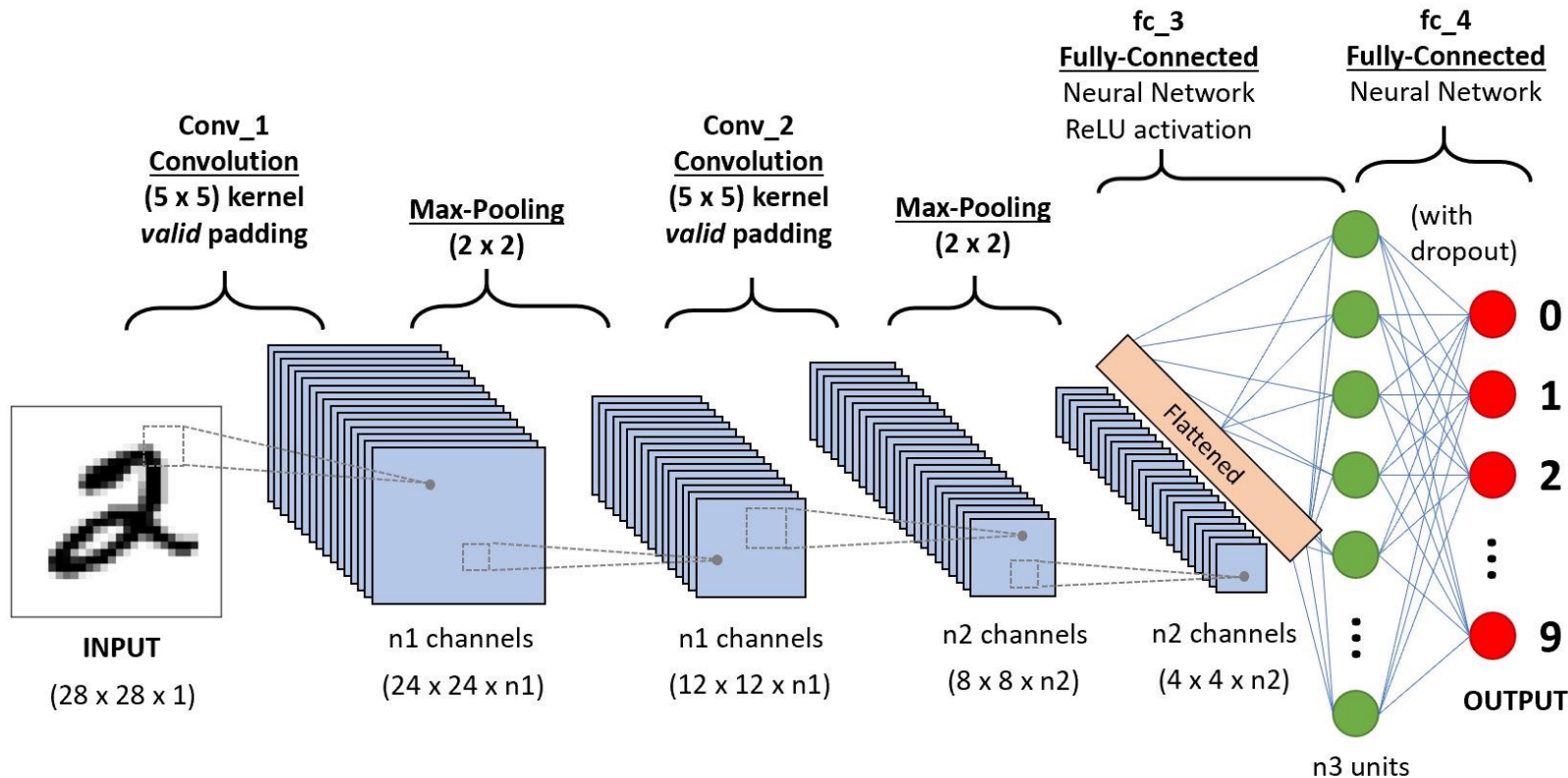


# Outline

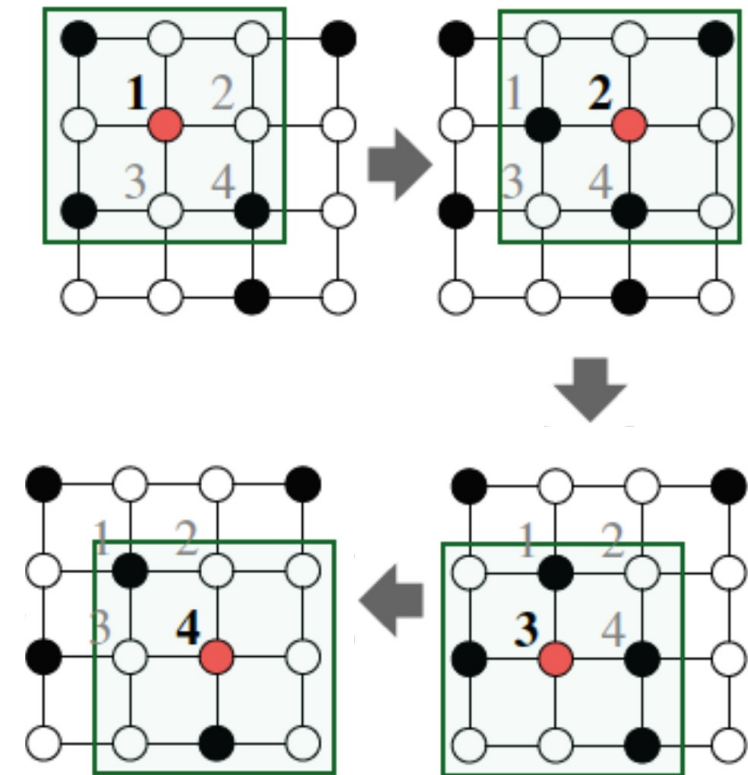
- Overview
- Graph Neural Network (GNN)
  - Graph Convolutional Neural Network (GCN)
  - Graph Attention Network (GAT)
  - Relational GCN

# Background: Convolutional neural networks for images

- Convolutional filters
  - Local feature detectors
  - A feature is learned in each **local receptive field** by a convolutional filter



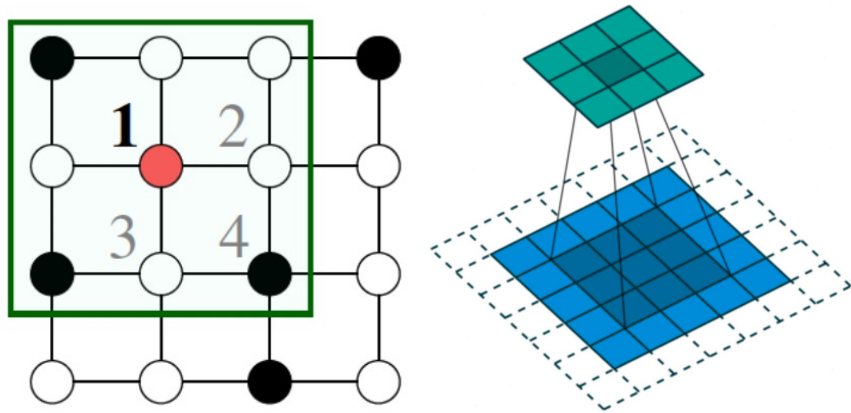
## CNN on an image



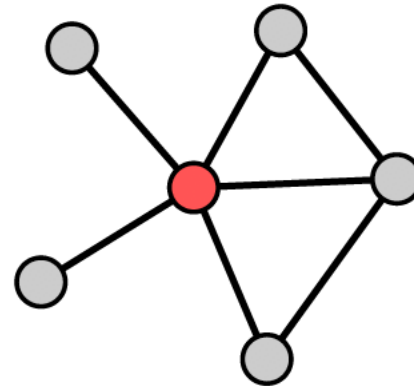
# From images to graphs: Local receptive field on graphs

- How should we define local receptive fields on graphs?

- Local subgraphs

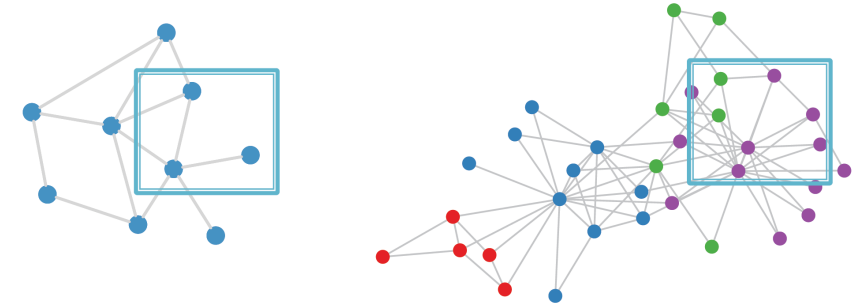


Image



Graph

## Graphs look like this

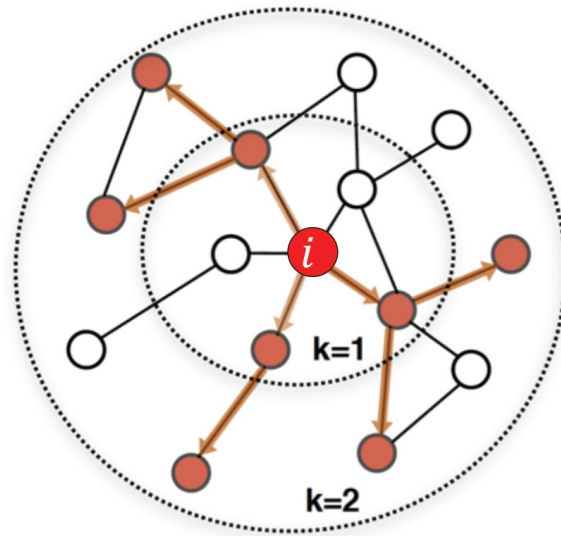


- There is no fixed notion of locality or sliding window on the graph
- No order among neighboring nodes
  - Permutation invariant

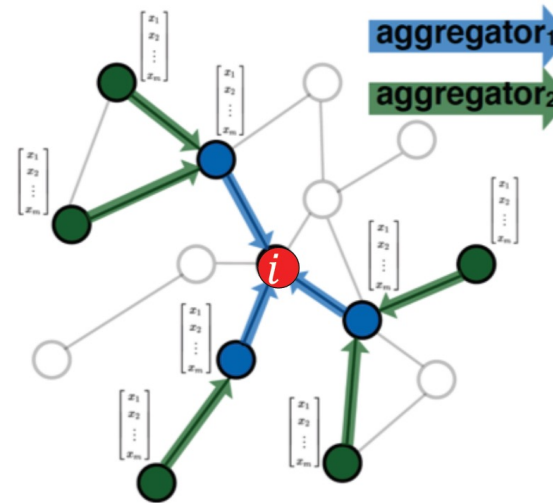
- Idea:** Transform information from the neighboring nodes and combine it
  - Step 1:** For each node  $v_i$ , transform “messages” from neighbors  $N(i)$ 
    - $W_j h_j$  for  $v_j \in N(i)$ ,  $h_j$ : “Message” from  $v_j$
  - Step 2:** Add them up:  $\sum_{v_j \in N(i)} W_j h_j$

# Graph Convolutional Network (GCN)

- **Idea:** Node's neighborhood defines a computation graph
  - Messages contain **relational information** + **attribute information**



Determine node  
computation graph

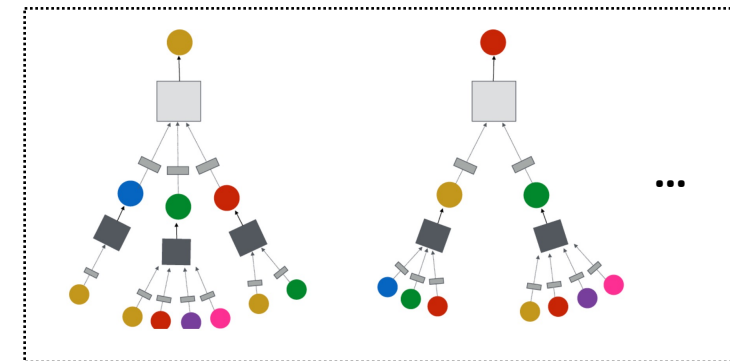


Propagate messages and  
transform information

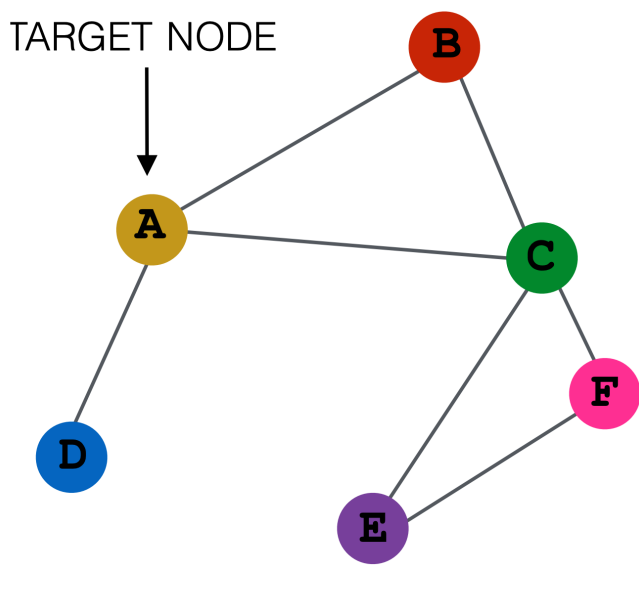
**Learn how to propagate information across the graph to compute node features**

# GCN: Neighborhood aggregation

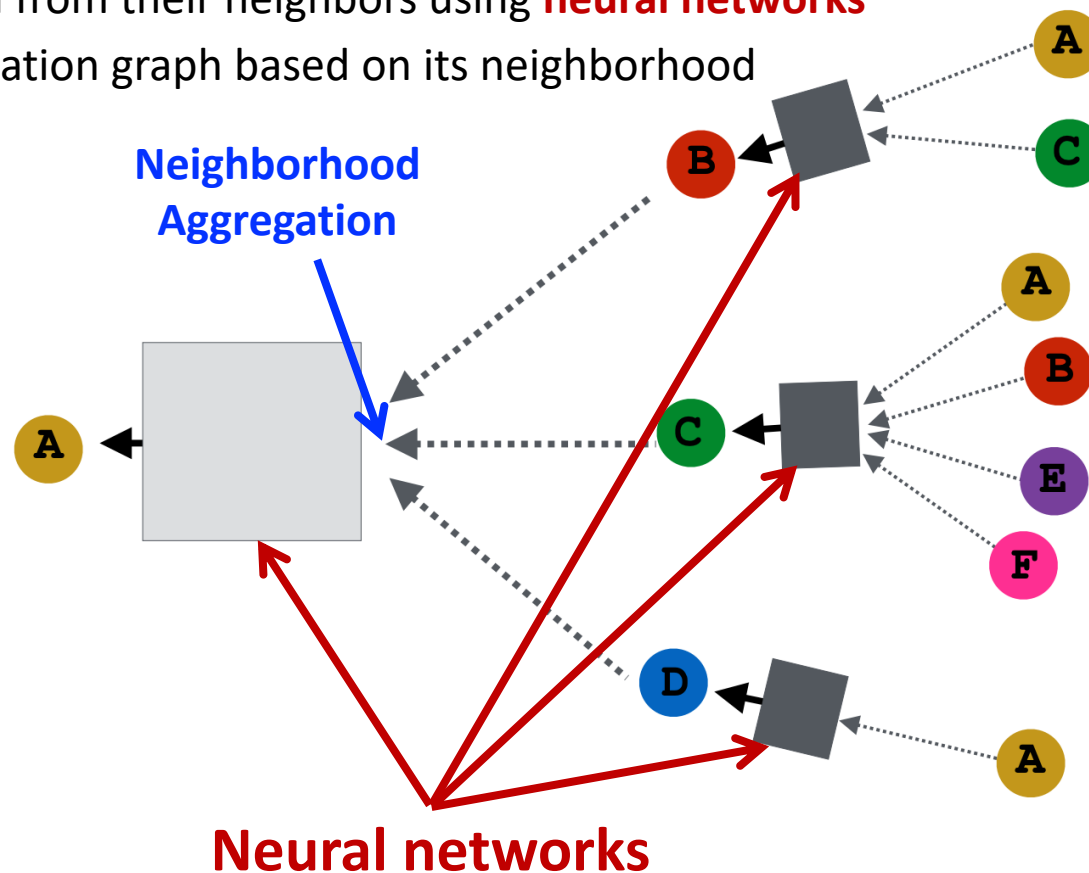
- Generate node embeddings based on local network neighborhoods
- **Neighborhood aggregation**
  - Nodes aggregate information from their neighbors using **neural networks**
  - Every node defines a computation graph based on its neighborhood



TARGET NODE



Input graph



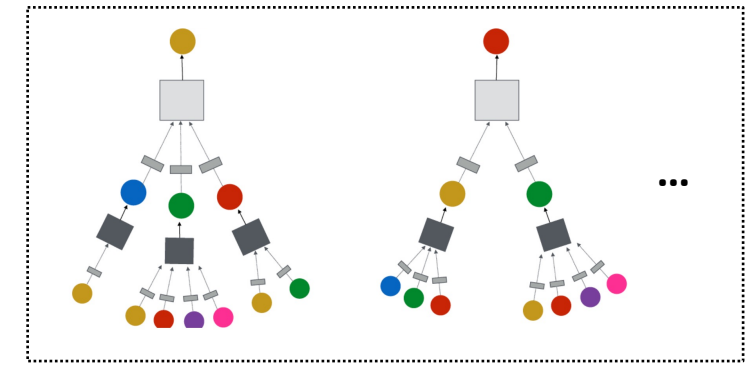
## ■ Things to consider

- 1. What kind of neural network?
- 2. How do we aggregate neighboring nodes?



# GCN: Basic approach

- 1. What kind of neural network?
  - Simple multiplication of weight matrices ( $B$  and  $W$ )
- 2. What kind of aggregation?
  - Average



Weight matrix

Embedding of  $v$  at layer  $l$

Total number of layers

$$h_v^{(l+1)} = \sigma \left( W_l \sum_{u \in N(v)} \frac{h_u^{(l)}}{|N(v)|} + B_l h_v^{(l)} \right), \quad \forall l \in \{0, 1, \dots, L-1\}$$

Initial embedding of  $v$

Feature of node  $v$

Average of neighboring nodes' previous layer embeddings

Final embedding of  $v$

How do we train the embeddings?

# GCN: Training

- We need to define the loss function on the embeddings
- We can feed the **final embeddings  $\mathbf{z}_v$**  into any loss function and run SGD to train the weight parameters
- **Types of loss function:** 1) Supervised loss, 2) Unsupervised loss

- **1) Supervised loss**

$$\min_{\theta} \sum_{v \in V} \mathcal{L}(y_v, f_{\theta}(\mathbf{z}_v))$$

- $y_v$ : Label of node  $v$
- $f_{\theta}$ : Classifier with parameter  $\theta$
- $\mathcal{L}$  could be squared error if  $y$  is real number (regression), or cross entropy if  $y$  is categorical (classification)

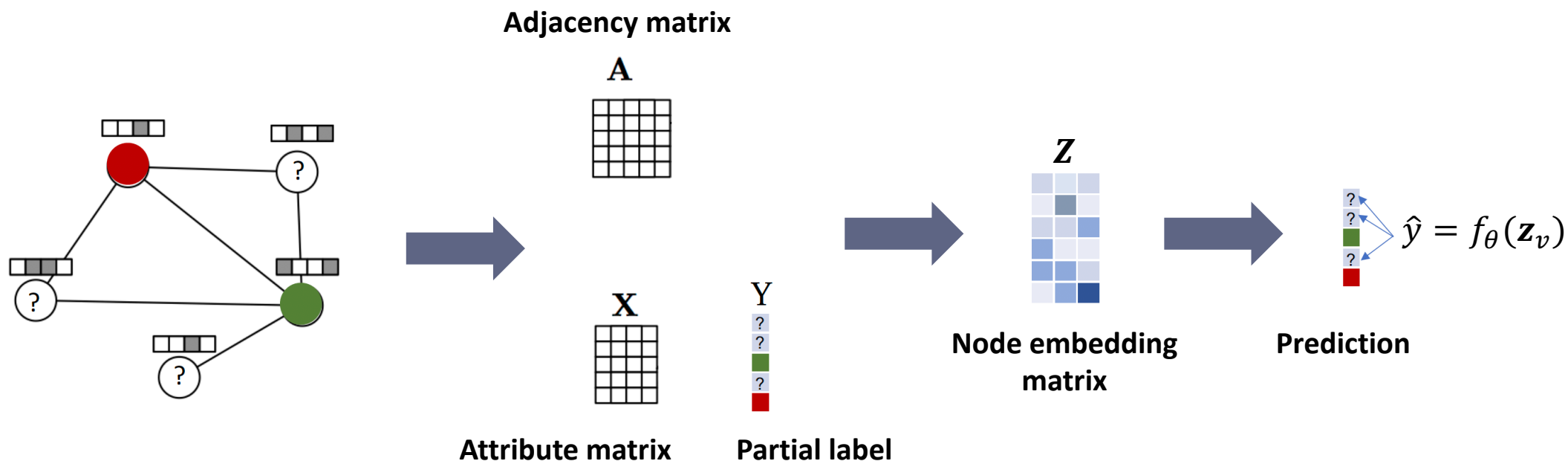
- **2) Unsupervised loss**

- No node label available
- We can use the graph structure as the supervision
  - e.g., adjacency information
  - In this case,  $\mathcal{L}$  is cross entropy ( $A_{v,u} = 1$  if an edge exists between node  $v$  and node  $u$ , otherwise 0)

$$\min_{\theta} \sum_{v, u \in V} \mathcal{L}(A_{v,u}, f_{\theta}(\mathbf{z}_v, \mathbf{z}_u)) \quad f_{\theta}: \text{Encoder}$$

# GCN: Supervised training

- Directly train the model for a supervised task (e.g., node classification)



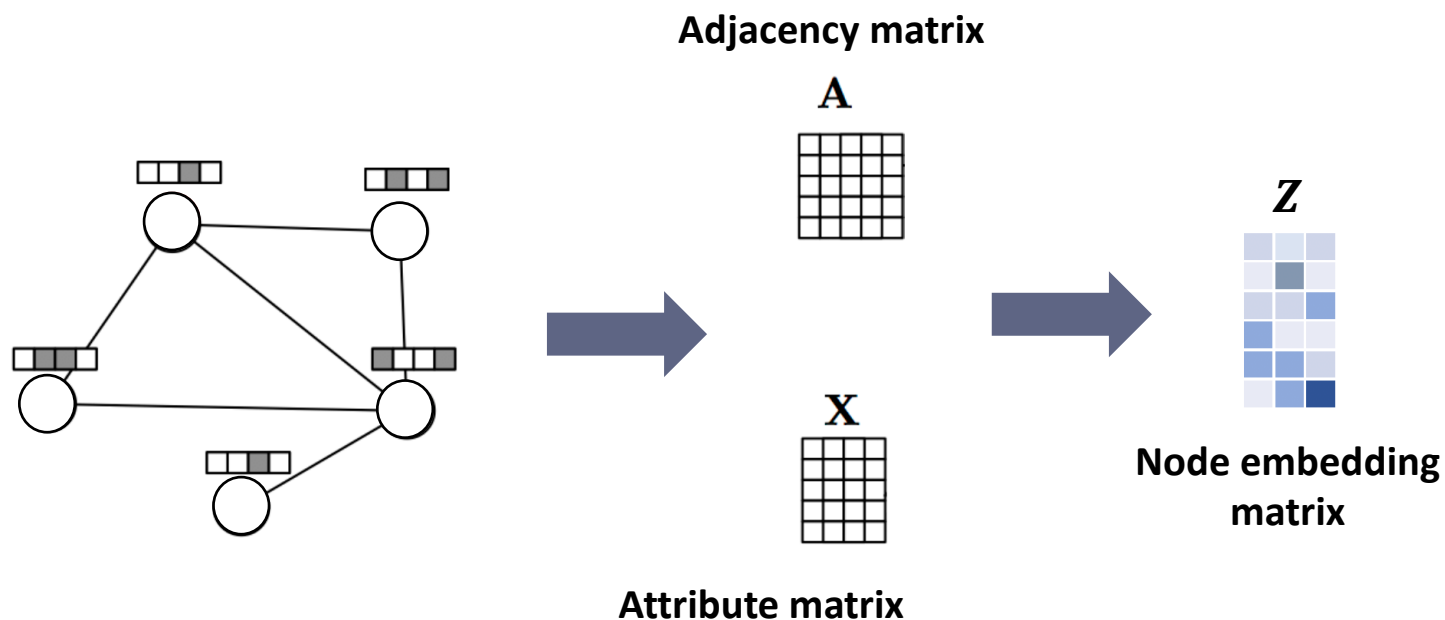
$$\mathcal{L} = - \sum_{v \in V} y_v \log f_{\theta}(z_v) + (1 - y_v) \log(1 - f_{\theta}(z_v))$$

Ground truth label

Model prediction

# GCN: Unsupervised training

- As we are not given node labels, we define our task to reconstruct the graph, i.e., Adjacency matrix



$$\mathcal{L} = - \sum_{v,u \in V} \boxed{A_{v,u}} \log \boxed{f_{\theta}(\mathbf{z}_v, \mathbf{z}_u)} + (1 - A_{v,u}) \log(1 - f_{\theta}(\mathbf{z}_v, \mathbf{z}_u))$$

Ground truth label

Model prediction

# Graph Attention Networks (GAT)

- **Idea:** Treat different neighboring nodes differently

$$\mathbf{h}_v^{(l+1)} = \sigma \left( \mathbf{W}_l \sum_{u \in N(v) \cup v} \alpha_{vu} \mathbf{h}_u^{(l)} \right)$$

Attention weight

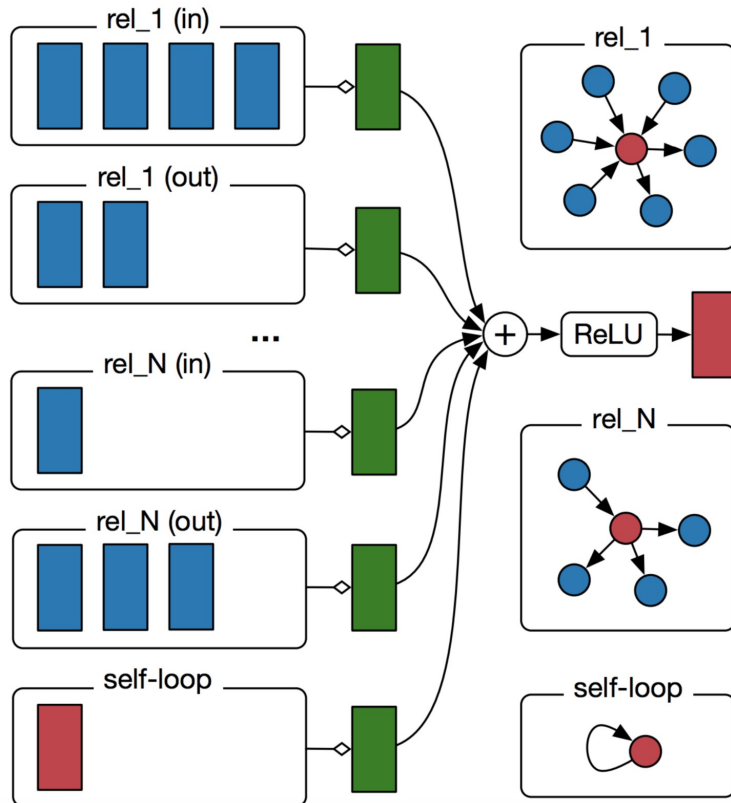
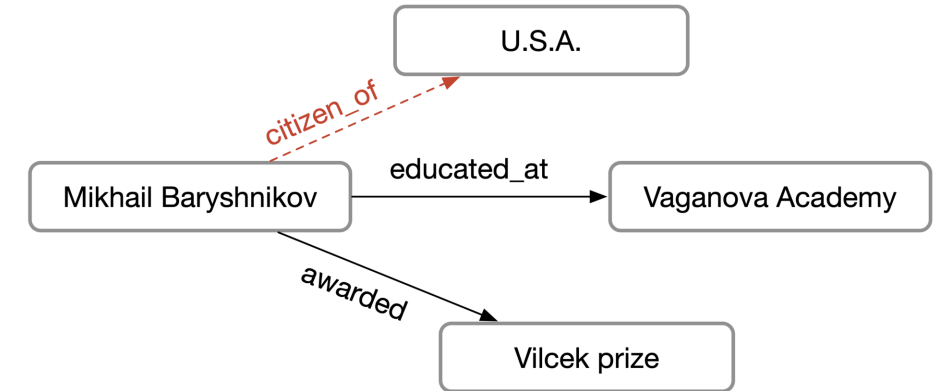
$$\mathbf{h}_v^{(l+1)} = \sigma \left( \mathbf{W}_l \sum_{u \in N(v)} \frac{\mathbf{h}_u^{(l)}}{|N(v)|} + \mathbf{B}_l \mathbf{h}_v^{(l)} \right) \quad (\text{GCN})$$

- $\alpha_{vu}$ : Importance of node  $u$  to node  $v$  as its neighboring node
- In GCN, the importance was heuristically defined based on the structural property of the graph (node degree)
  - $\alpha_{vu} = \frac{1}{|N(v)|}$ : Does not depend on the neighbors (it is fixed)
  - All neighboring nodes  $u \in N(v)$  are **equally important** to node  $v$

**Not all neighbors are equally important!**

# R-GCN: RELATIONAL GCN

- Knowledge graph is a type of multi-relational graph
  - Nodes are entities, the edges are relations labeled with their types



$$\mathbf{h}_v^{(l+1)} = \sigma \left( \mathbf{W}_l \sum_{u \in N(v)} \frac{\mathbf{h}_u^{(l)}}{|N(v)|} + \mathbf{B}_l \mathbf{h}_v^{(l)} \right) \quad (\text{GCN})$$



$$\mathbf{h}_v^{(l+1)} = \sigma \left( \sum_{r \in R} \mathbf{W}_l^r \sum_{u \in N(v)} \frac{\mathbf{h}_u^{(l)}}{|N(v)|} + \mathbf{B}_l \mathbf{h}_v^{(l)} \right) \quad (\text{R-GCN})$$

# Conclusion

- Overview
- Graph Neural Network (GNN)
  - Graph Convolutional Neural Network (GCN)
  - Graph Attention Network (GAT)
  - Relational GCN

# Outline

- 그래프 신경망 개요 (20 mins)
  - 그래프 신경망 전반적인 소개
  - 그래프 종류에 따른 다양한 그래프 신경망 소개
- How to address Out-of-distribution problem (세부 기술 및 Q&A) (90~120 mins)
  - 소재 물성 예측 연구
    - 소재 물성 예측 연구 최신 동향 소개
    - Transformer 기반 모델 소개 → Prompt-based method
    - Extrapolation을 위한 모델 소개 → Nonlinearity encoding-based method
  - 물질 간 화학 반응 예측 연구
    - 물질 간 화학 반응 예측 연구 동향 소개
    - 정보 이론(Information bottleneck) 기반 모델 소개 → Information bottleneck-based method
    - 인과추론(Causal inference) 기반 모델 소개 → Causal inference-based method



# Papers

## ■ Material property prediction

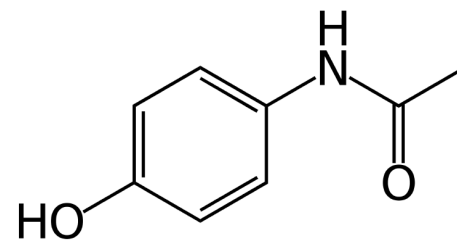
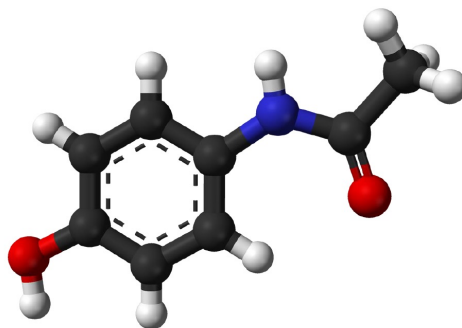
- Neural message passing for quantum chemistry. ICML 2017
- Schnet: a continuous-filter convolutional neural network for modeling quantum interactions. NeurIPS 2017
- Crystal Graph Convolutional Neural Networks for an Accurate and Interpretable Prediction of Material Properties. Phys. Rev. Lett. 2018
- Graph networks as a universal machine learning framework for molecules and crystals. Chem. Mater. 2019
- **Predicting Density of States via Multi-modal Transformer. ICLR Workshop 2023**

## ■ Extrapolation

- How Neural Networks Extrapolate: From Feedforward to Graph Neural Networks. ICLR 2021
- **Nonlinearity Encoding for Extrapolation of Neural Networks. KDD 2022**

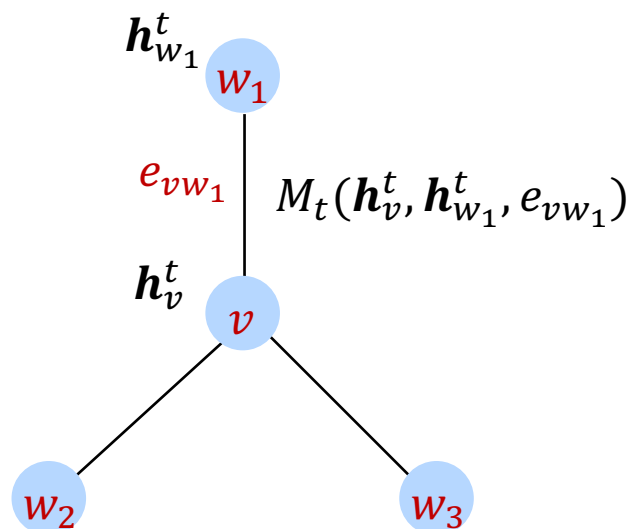
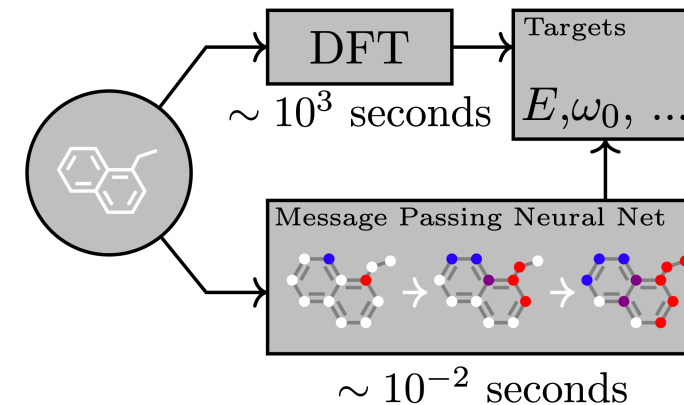
# Molecular Graphs

- Molecules can be represented as a graph with node features and edge features
  - Node features: atom type, atom charges...
  - Edge features: valence bond type...



# Message Passing Neural Network

- Unified various graph neural network and graph convolutional network approaches



$$m_v^{t+1} = \sum_{w \in N(v)} M_t(h_v^t, h_w^t, e_{vw})$$

Edge embedding

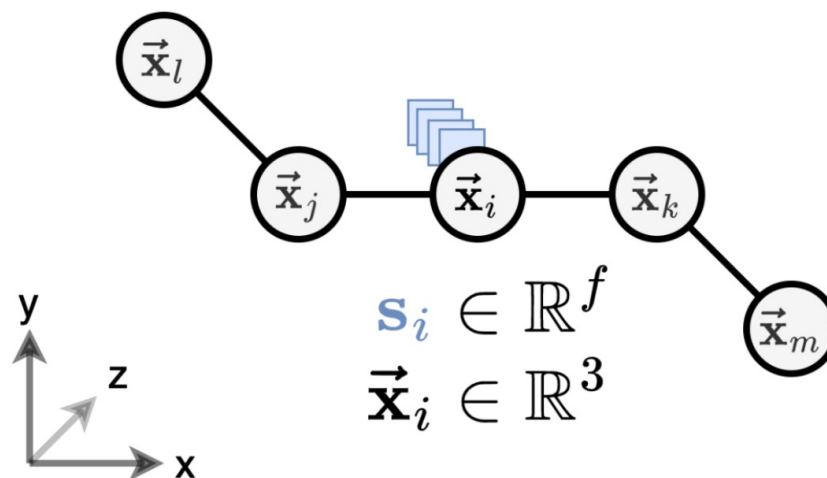
Neighbor of  $v$

$$h_v^{t+1} = U_t(h_v^t, m_v^{t+1})$$

$$\hat{y} = R(\{h_v^T \mid v \in G\})$$

# Geometric Graphs

- Sometimes, we also know the 3D positions of atoms, which is actually more informative
- A geometric graph  $G = (A, S, X)$  is a graph where each node is embedded in  $d$ -dimensional **Euclidean space**:

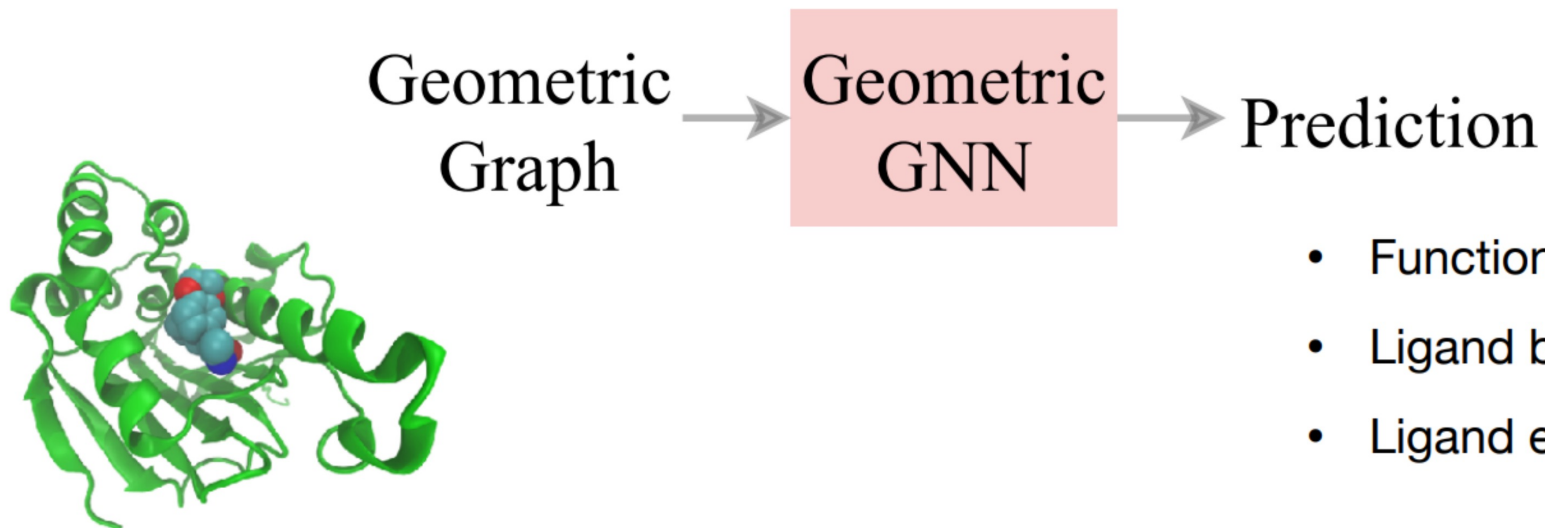


- $A$ : an  $n \times n$  adjacency matrix
- $S \in \mathbb{R}^{n \times f}$ : Scalar features (atom type, atom charges, ...)
- $X \in \mathbb{R}^{n \times d}$ : tensor features, e.g., coordinates

# Broad Impact on Sciences

## ■ Supervised Learning: Prediction

- Properties prediction
- 3D Protein-ligand interaction (binding)

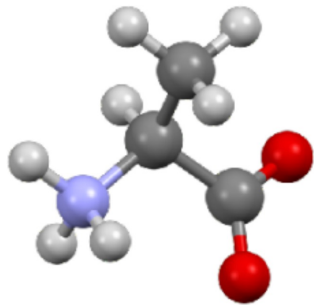


- Functional properties?
- Ligand binding affinity?
- Ligand efficacy?

# Broad Impact on Sciences

## ■ Supervised Learning: Structured Prediction

- Molecular Simulation

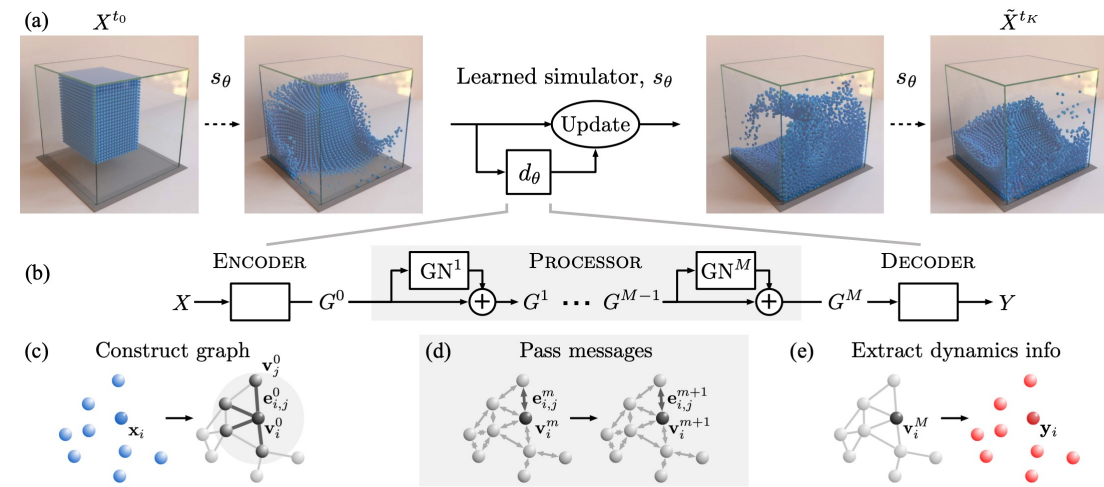
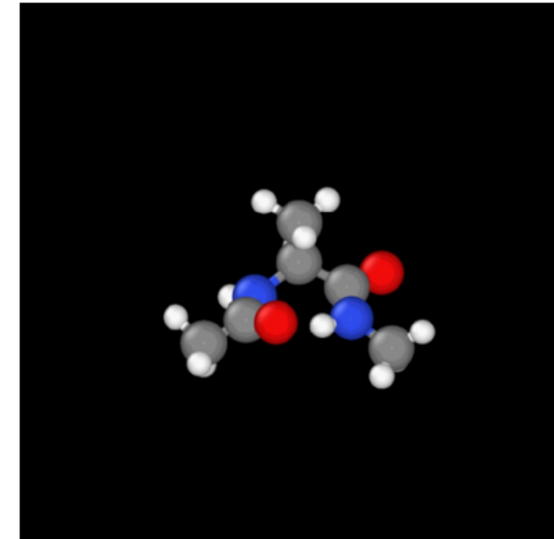
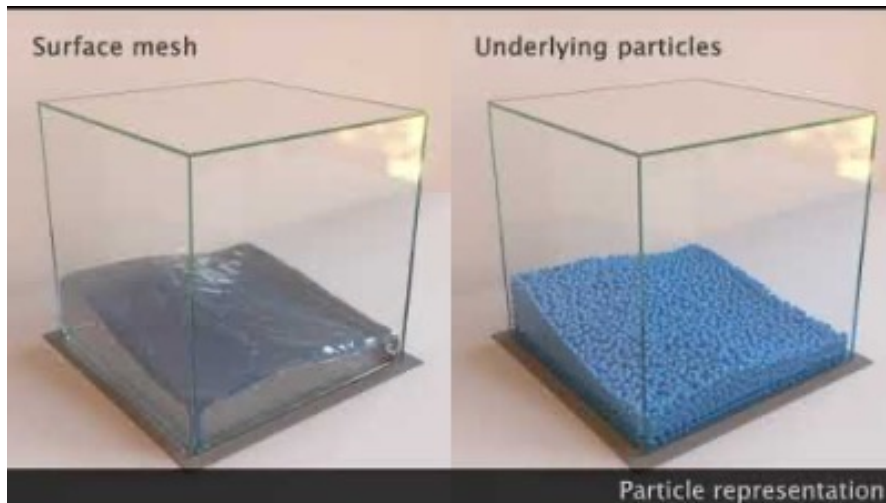


Current  
State

Geometric  
GNN

Next  
State

Dynamics  
Simulator



# Broad Impact on Sciences

- Generative Models
  - Drug or material design



Geometric  
Graph

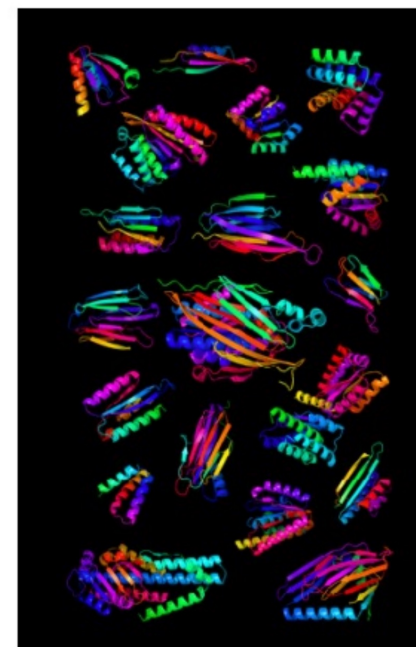


Geometric  
GNN

Generative  
Model

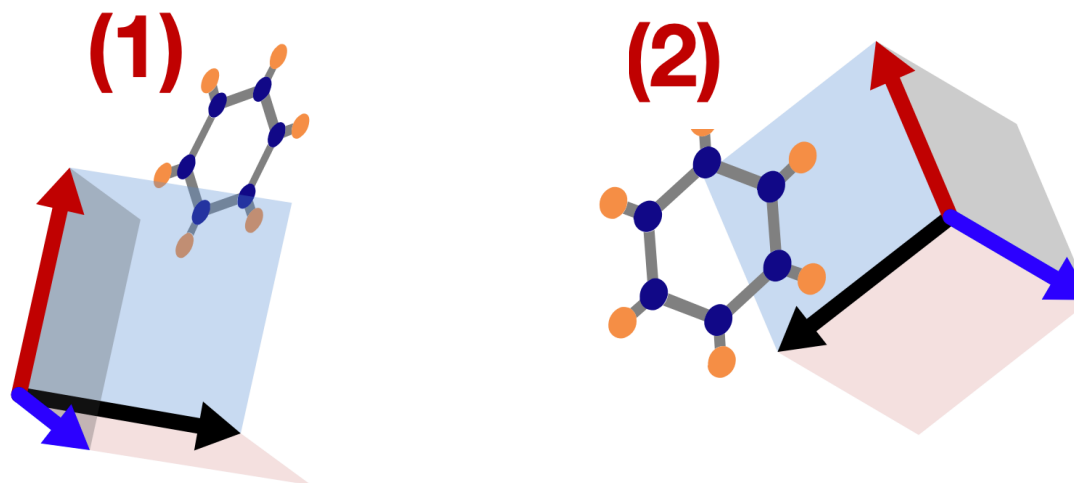


Geometric  
Graph



# Geometric graph is more challenging than Molecular graph

- To describe geometric graphs, we use **coordinate systems**
  - (1) and (2) use different coordinate systems to describe the **same** molecular geometry.
- We can describe the transform between coordinate systems with **symmetries** of Euclidean space
  - 3D rotations, translations



However, output of traditional GNNs given (1) and (2) are completely different!  
→ Enforcing symmetry is crucial (Invariant GNNs)



# Schnet: Overview

## Input

- Feature representations of  $n$  atoms  $X^l = (\mathbf{x}_1^l, \dots, \mathbf{x}_n^l)$  with  $\mathbf{x}_i^l \in R^F$
- At locations  $R = (\mathbf{r}_1, \dots, \mathbf{r}_n)$  with  $\mathbf{r}_i \in R^D$  ( $D = 3$  for 3-dim coordinates)

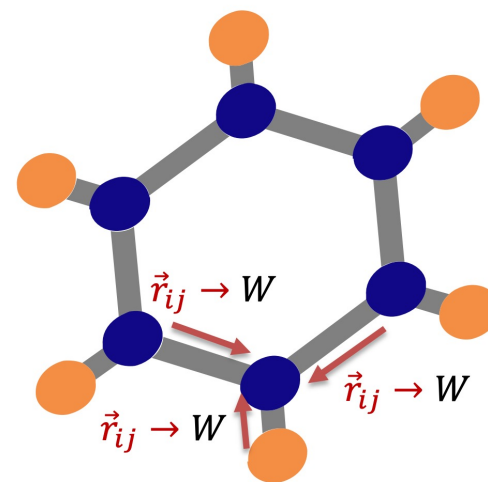
## Output

- Molecular total energy  $E(\mathbf{r}_1, \dots, \mathbf{r}_n)$

- SchNet updates the node embeddings at the  $l$ -th layer by message passing layers

$$\mathbf{x}_i^{l+1} = (X^l * W^l)_i = \sum_j \mathbf{x}_j^l \circ W^l(\mathbf{r}_i - \mathbf{r}_j),$$

- A filter generating function  $W^l: R^D \rightarrow R^F$  is determined by the relative position from neighbor atoms  $j$  to  $i$
- $\circ$  is the element-wise multiplication

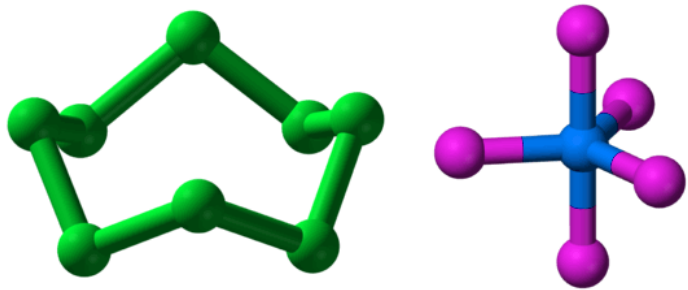


$\mathbf{x}^l$ : node embeddings at  $l$  layer  
 $\mathbf{r}$ : atomic coordinates

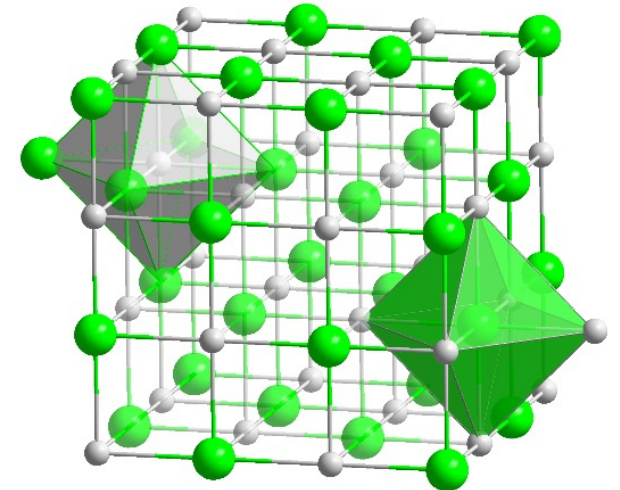
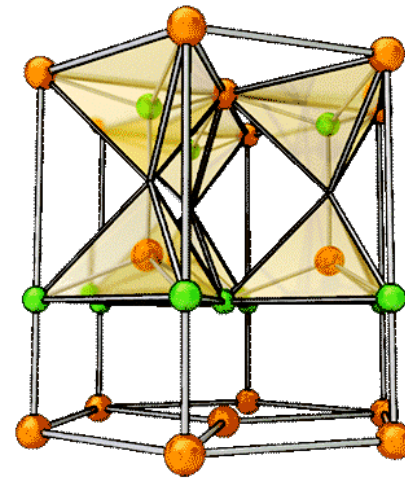
# Schnet: Invariance

- $W$  is invariant by scalarizing relative positions with relative distances ( $\|r_i - r_j\| = \|r_{ij}\| = d_{ij}$ )
  - $\|r_{ij}\|$  is invariant to **rotations** and **translations**
- Hence, each message passing layer  $W^l$  is invariant
- Aggregated node embeddings  $\sum_j \mathbf{x}_j^l \circ W^l(\mathbf{r}_i - \mathbf{r}_j)$  is invariant
- **Node embeddings are invariant!**

# Crystalline Materials



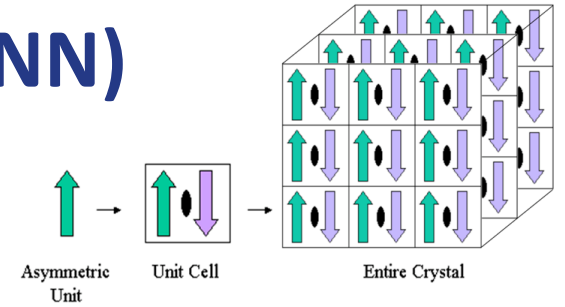
Molecules



Crystals

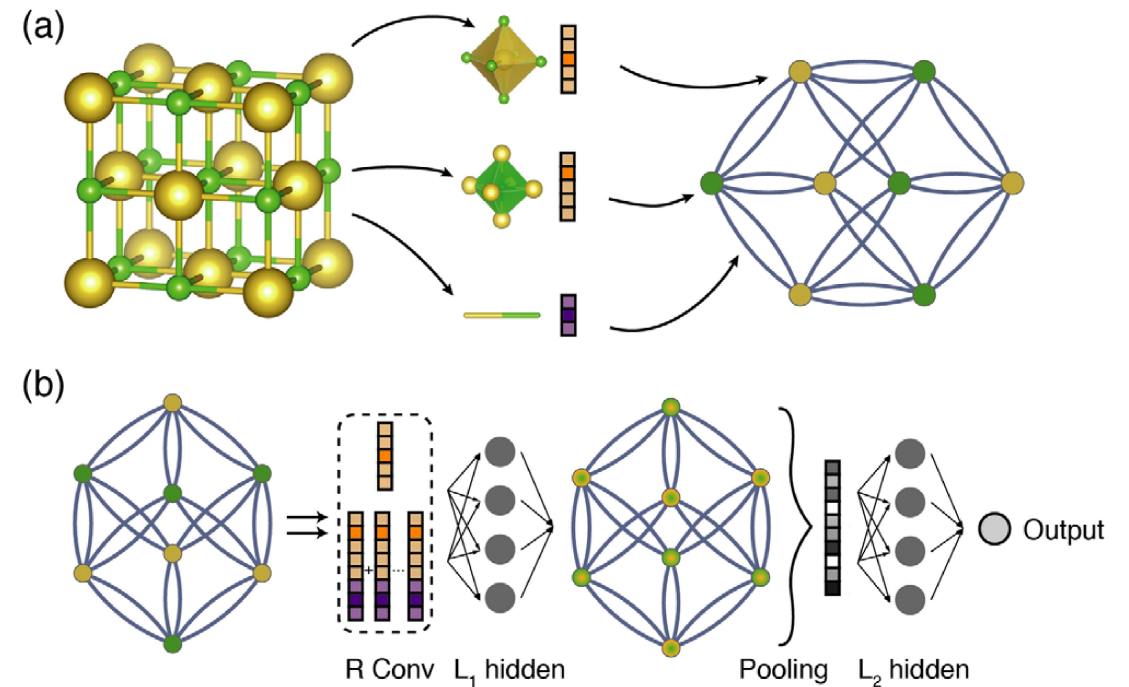
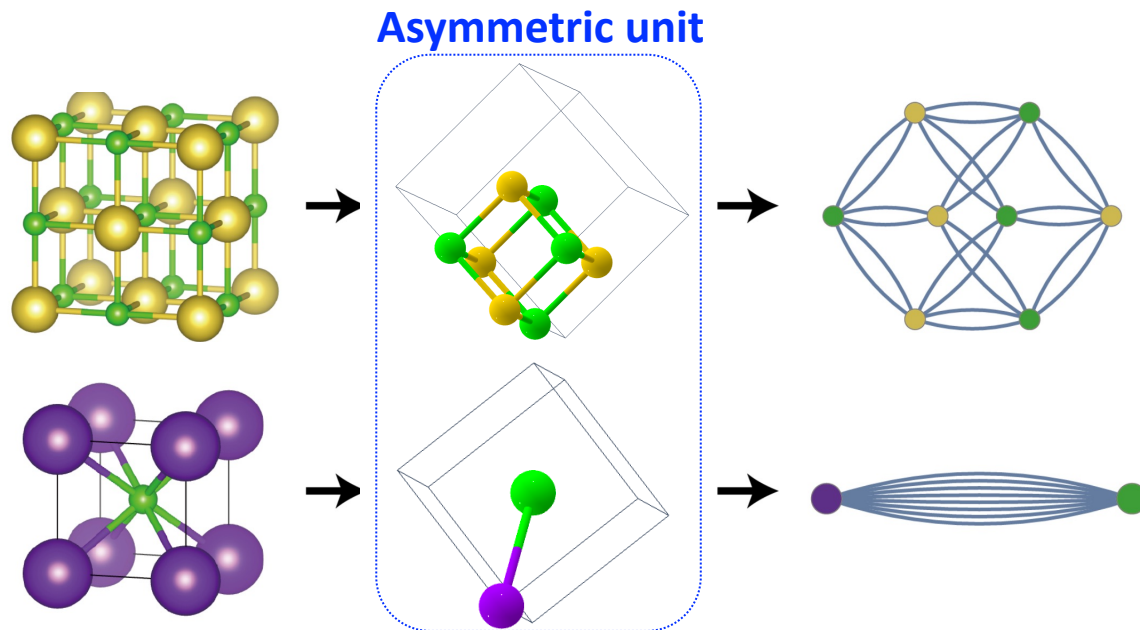
# Crystal Graph Convolutional Neural Networks (CGCNN)

- Goal: Predict material properties of periodic **crystal systems**
- Idea: Represent the crystal structure by a **crystal graph** that encodes both atomic information and bonding interactions between atoms (Distance between atoms  $\rightarrow$  Edges in a crystal graph)



## Undirected multigraph

- Multiple edges between the same pair of nodes
- Considers lattice periodicity



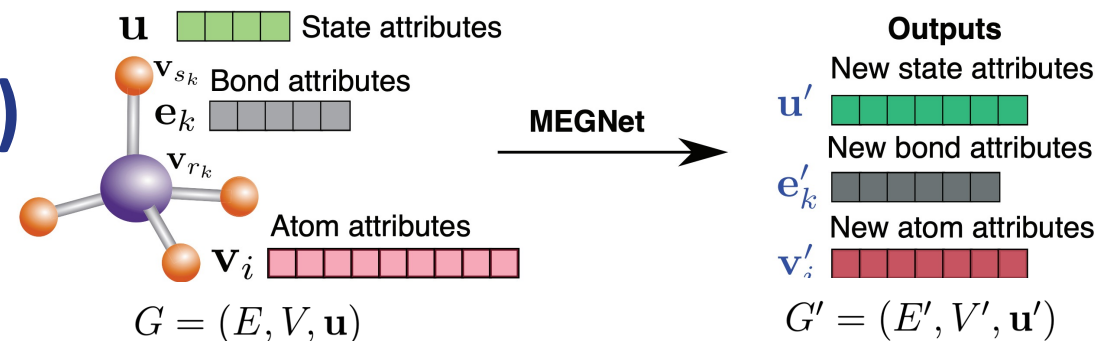
# MatErials Graph Network (MEGNet)

## ■ Motivation

- 1) Existing work either on molecular and crystal datasets
- 2) Global state (e.g., temperature) of each molecule/crystal is overlooked
  - Important for predicting state-dependent properties such as the free energy

Solved by adopting graph networks!

## ■ Considers topological distance and spatial distance (Manual features)



$$\bar{\mathbf{v}}_i^e = \frac{1}{N_i^e} \sum_{k=1}^{N_i^e} \{\mathbf{e}'_k\}_{r_k=i}$$

$$\bar{\mathbf{u}}^e = \frac{1}{N^e} \sum_{k=1}^{N^e} \{\mathbf{e}'_k\}$$

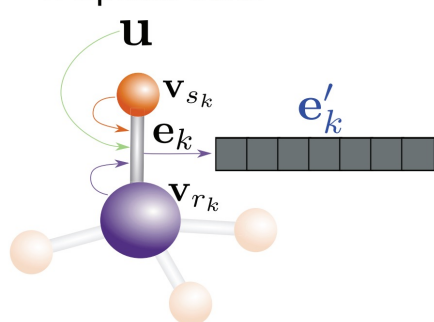
$$\mathbf{v}'_i = \phi_v(\bar{\mathbf{v}}_i^e \oplus \mathbf{v}_i \oplus \mathbf{u})$$

$$\bar{\mathbf{u}}^v = \frac{1}{N^v} \sum_{i=1}^{N^v} \{\mathbf{v}'_i\}$$

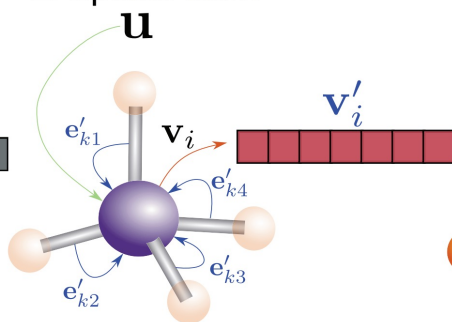
$$\mathbf{u}' = \phi_u(\bar{\mathbf{u}}^e \oplus \bar{\mathbf{u}}^v \oplus \mathbf{u})$$

### MEGNet update steps

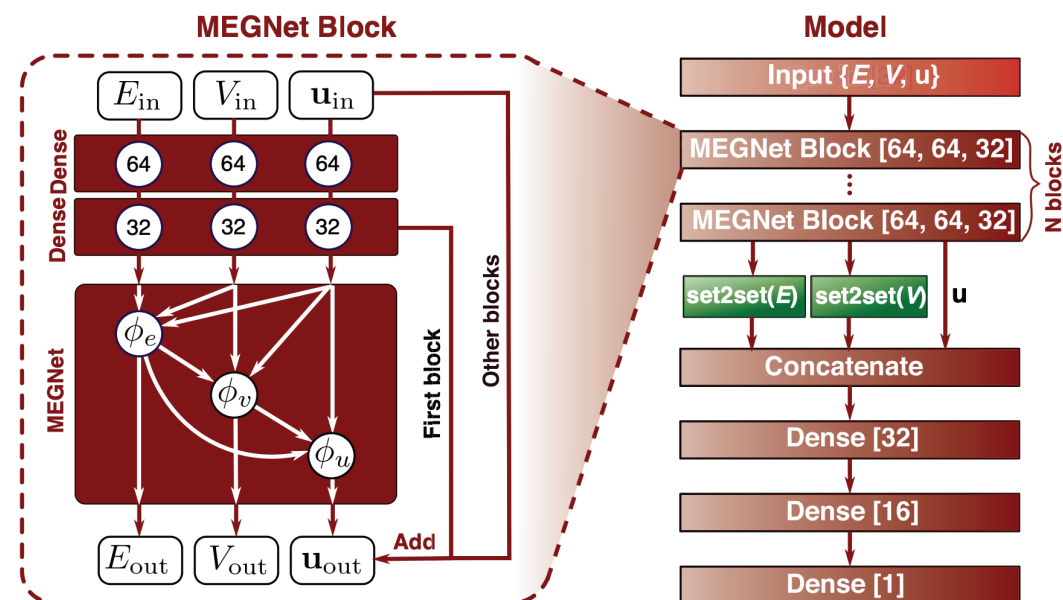
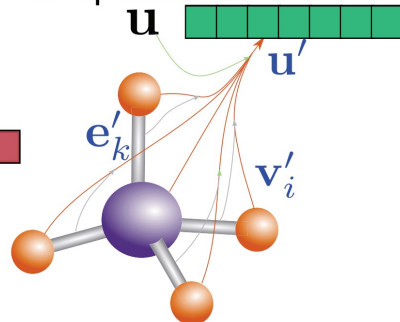
#### 1. Update bond



#### 2. Update atom



#### 3. Update state



# Outline

- 그래프 신경망 개요 (20 mins)
  - 그래프 신경망 전반적인 소개
  - 그래프 종류에 따른 다양한 그래프 신경망 소개
- How to address Out-of-distribution problem (세부 기술 및 Q&A) (90~120 mins)
  - 소재 물성 예측 연구
    - 소재 물성 예측 연구 최신 동향 소개
    - Transformer 기반 모델 소개 → **Prompt-based method**
    - Extrapolation을 위한 모델 소개 → Nonlinearity encoding-based method
  - 물질 간 화학 반응 예측 연구
    - 물질 간 화학 반응 예측 연구 동향 소개
    - 정보 이론(Information bottleneck) 기반 모델 소개 → Information bottleneck-based method
    - 인과추론(Causal inference) 기반 모델 소개 → Causal inference-based method

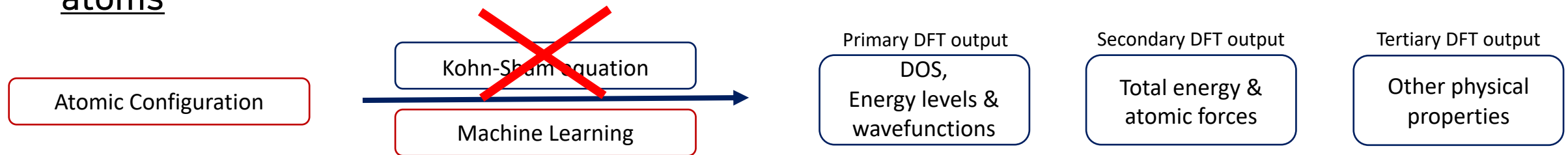
# Density of States Prediction of Crystalline Materials via Prompt-guided Multi-Modal Transformer (Under review)

Namkyeong Lee, Heewoong Noh, Sungwon Kim, Dongmin Hyun, Gyoung S. Na, Chanyoung Park  
(Based on ICLR 2023 ML4Materials Workshop paper)

**Density-functional theory (DFT)** is a computational quantum mechanical modelling method used in physics, chemistry and materials science to investigate the electronic structure (or nuclear structure) (principally the ground state) of many-body systems, in particular atoms, molecules, and the condensed phases. Using this theory, the properties of a many-electron system can be determined by using functionals, i.e. functions of another function. In the case of DFT, these are functionals of the spatially dependent electron density. DFT is among the

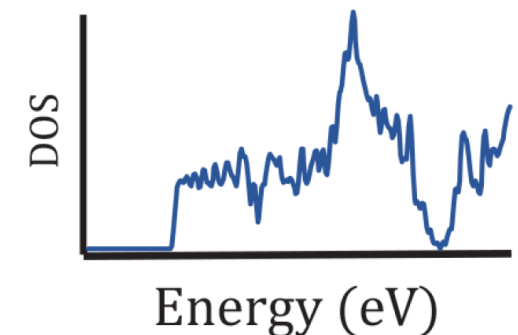
# Density Functional Theory (DFT)

- DFT calculations are used to determine the mechanisms of chemical reactions that are difficult to experimentally determine by considering the movements and reactions of electrons within atoms



- However, it is difficult and computationally expensive to compute DFT outputs based on Kohn-Sham equation
- In this work, we adopt GNNs to approximate Kohn-Sham Equation to predict DOS

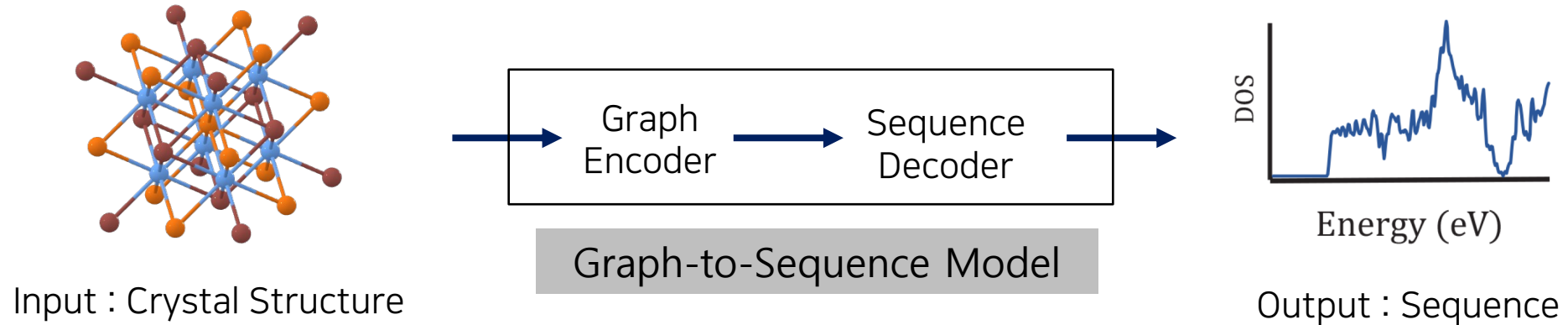
Main assumption: DOS is related to a sequence of energy



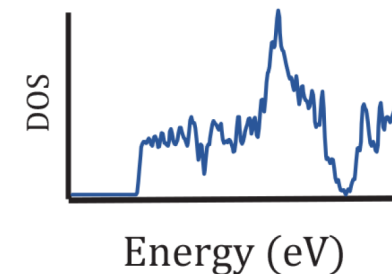


# Consider DOS as a sequence

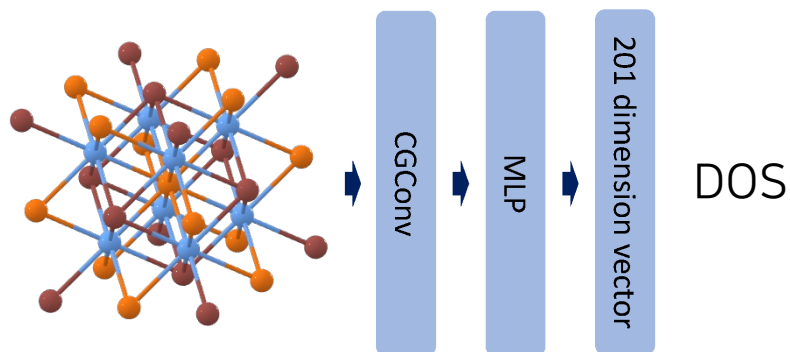
- Idea: DOS prediction = Graph-to-Sequence task



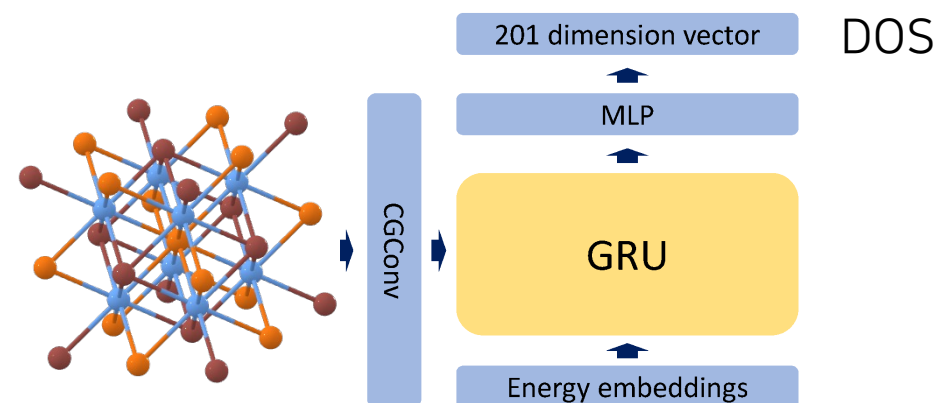
# Baseline methods



- **Baseline 1 - CGCNN**: Use Crystal Graph Convolution [1] to predict 201 DOS values at once
- **Baseline 2 - CGGRU**: Use graph embedding as the initial state of GRU and sequentially predict DOS given energy embeddings



Baseline 1: CGCNN



Baseline 2: CGGRU

- **Performance**: CGGRU > CGCNN (2% Gap in MSE)
- **Key Takeaways: Sequential modeling is important**
  - We need to explicitly capture the relationship between energies
  - What about adopting Transformer?

**Challenge:** Input types are different (Different modality)

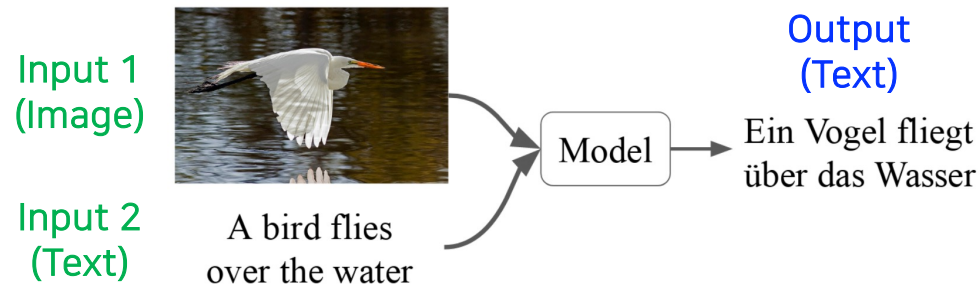
- Modality: Graph  $\neq$  Energy

# Multimodal transformer

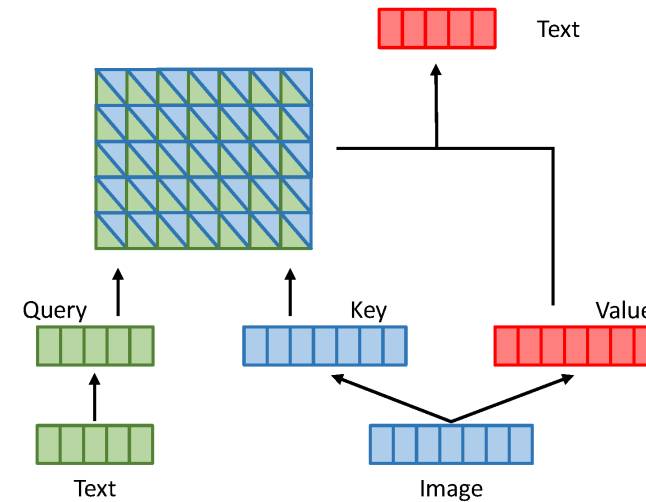
- How can we perform machine translation given both image and text data?

- Multi-modal Machine Translation

→ Multi-modal Transformer



Multi-modal Machine Translation



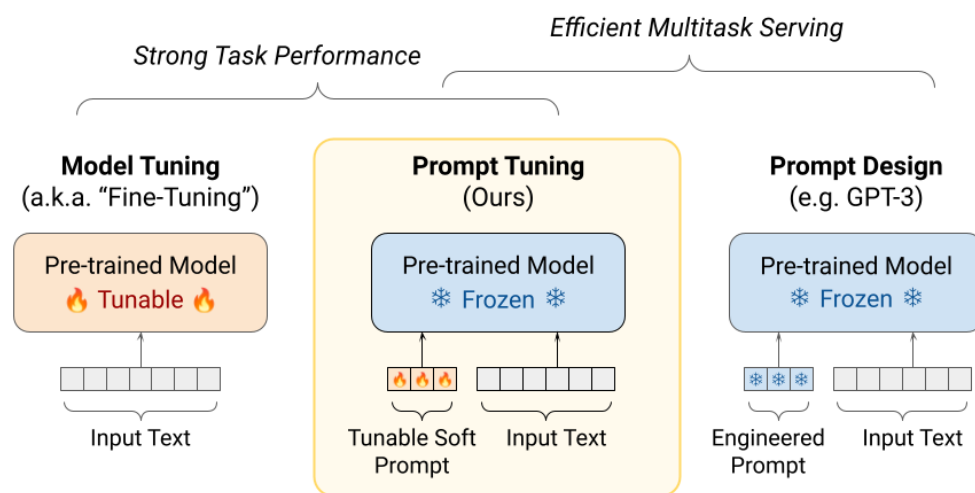
Multi-modal Transformer

- Multi-modal transformer assigns Query, Key, Value for different modalities
  - Query: Text
  - Key, Value: Image
- We refer to the interaction between Query and Key, and combine with Value to get Query embedding

# Preliminary: Prompt Tuning

## How to effectively fine-tune pre-trained models (LLMs) for downstream tasks?

- Prompt Design (e.g. GPT-3)
- cf) Fine-Tuning → Prompt Design → Prompt Tuning



Tunable Soft-prompts  $P_e \in \mathbb{R}^{p \times e}$

Concatenated  $[P_e; X_e] \in \mathbb{R}^{(p+n) \times e}$

ex) Sentiment Classification Task

- **Finetuning:** "This movie was amazing!" → Positive
- **Prompt design:** Engineered prompt + Input text (In-context learning)

$$\left[ \text{Is the following movie review positive or negative?} + \text{"This movie was amazing!"} \right]$$
  
**Engineered Prompt**
**Input text**

- **Prompt tuning:** Tunable soft prompt + Input text

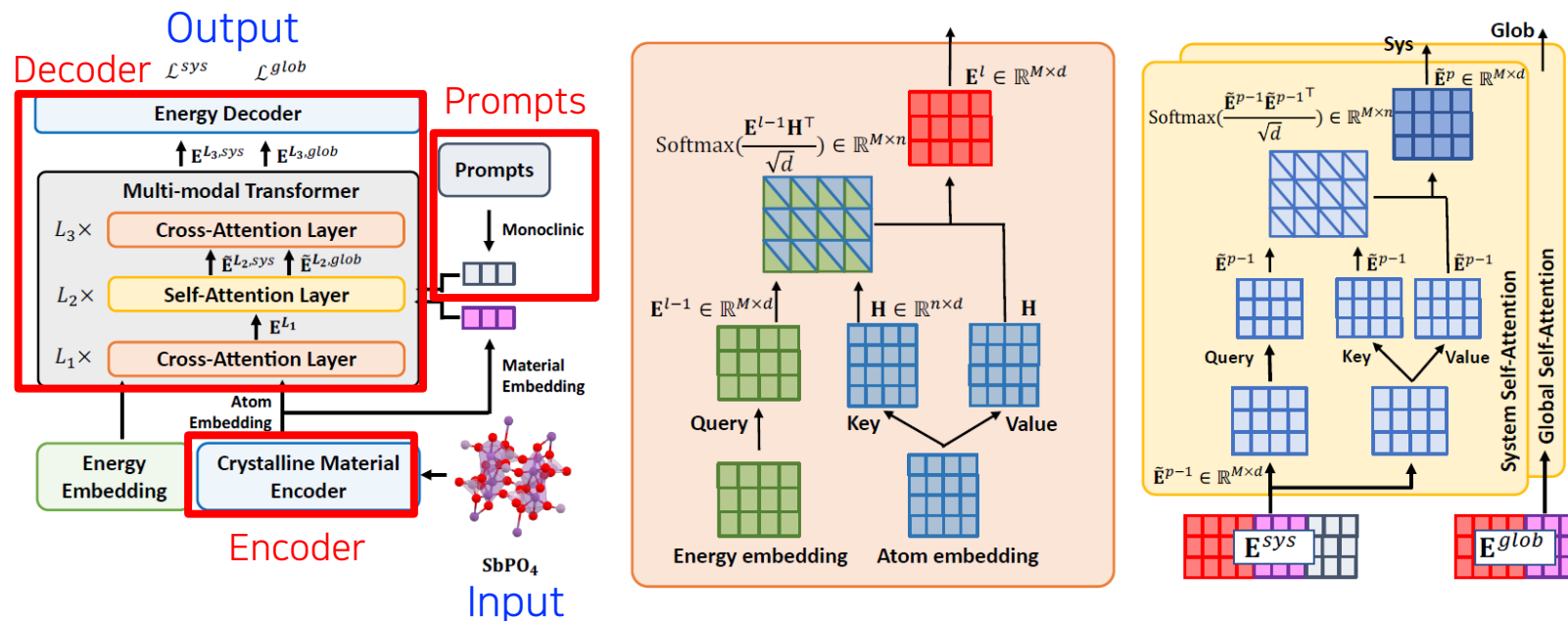
**Our idea:** There are 7 Widely known Crystal Systems

- i.e., Cubic, Hexagonal, ..., Triclinic
- Introduce 7 learnable prompts  $P \in \mathbb{R}^{7 \times d_p}$

- Incorporating structural information to the model by injecting prompts, not naively concatenating
- Later, we show that introducing prompts is helpful for addressing OOD problems

# Our proposed method: Prompt-guided DOSTransformer

- **Query**: Energy / **Key**, **Value**: Graph (Atom)
- We determine which atom to focus on at each energy level for DOS prediction
  - i.e., Crystal-specific energy embedding
- We utilize learnable prompts to guide the model to learn the crystal structural system-specific interaction between materials and energies



Proposed Model  
(Prompt-guided DOSTransformer)

# Result: In-distribution

: Phonon DOS

: Electron DOS

Model	Phonon DOS			Electron DOS			Physical Properties (MSE)		
	MSE	MAE	$R^2$	MSE	MAE	$R^2$	Bulk M.	Band G.	Ferm. E.
<b>Energy ✗</b>									
MLP	0.346 (0.004)	0.112 (0.001)	0.517 (0.005)	0.714 (0.013)	0.187 (0.001)	-0.146 (0.050)	0.720 (0.026)	1.425 (0.166)	5.039 (0.120)
Graph Network	0.359 (0.009)	0.108 (0.001)	0.502 (0.001)	0.319 (0.006)	0.113 (0.001)	0.530 (0.008)	0.725 (0.073)	0.784 (0.116)	3.849 (0.121)
E3NN	0.210 (0.004)	0.077 (0.001)	0.705 (0.007)	0.301 (0.002)	0.110 (0.000)	0.551 (0.009)	0.504 (0.033)	0.705 (0.073)	3.677 (0.139)
<b>Energy ✓</b>									
MLP	0.244 (0.000)	0.097 (0.001)	0.660 (0.002)	0.320 (0.015)	0.124 (0.004)	0.527 (0.020)	0.549 (0.007)	0.854 (0.046)	4.207 (0.165)
Graph Network	0.213 (0.006)	0.087 (0.001)	0.701 (0.010)	0.252 (0.003)	0.102 (0.001)	0.632 (0.002)	0.568 (0.093)	0.748 (0.068)	3.759 (0.135)
E3NN	0.200 (0.001)	0.074 (0.001)	0.724 (0.002)	0.295 (0.006)	0.111 (0.001)	0.562 (0.012)	0.451 (0.023)	0.872 (0.090)	3.780 (0.160)
DOSTransformer	<b>0.191</b> (0.003)	<b>0.071</b> (0.002)	<b>0.733</b> (0.004)	<b>0.225</b> (0.002)	<b>0.089</b> (0.001)	<b>0.671</b> (0.006)	<b>0.427</b> (0.024)	<b>0.455</b> (0.018)	<b>3.324</b> (0.036)

- It is beneficial to consider the energy level
  - However, a naïve consideration is not much helpful
- For Phonon DOS, predicting Bulk Modulus based on the output of our model is the best
- For Electron DOS, predicting Band Gap, Fermi Energy based on the output of our model is the best

# Result: Out-of-distribution

- **Scenario 1** - Train: binary and ternary / Test: Unary, Quaternary, and Quinary
- **Scenario 2** - Train: Cubic, Hexagonal, Tetragonal, Trigonal, and Orthorhombic / Test: rest
  - For Scenario 2: As no prompts are available for unseen crystal systems, we use the mean-pooled representation of the trained prompts
    - i.e., mean of cubic, hexagonal, tetragonal, trigonal and orthorhombic
  - DOSTransformer performs well in OOD

Table 2: The number of crystals according to the number of atom species (Scenario 1).

	Unary (1)	Binary (2)	Ternary (3)	Quaternary (4)	Quinary (5)	Senary (6)	Septenary (7)	Total
# Crystals	386	9,034	21,794	5,612	1,750	279	34	38,889

Table 3: The number of crystals according to different crystal systems (Scenario 2).

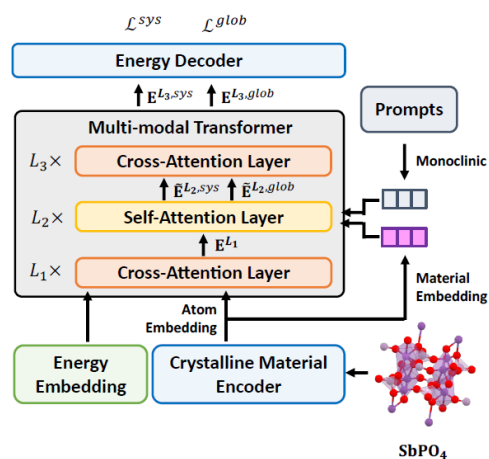
	Cubic	Hexagonal	Tetragonal	Trigonal	Orthorhombic	Monoclinic	Triclinic	Total
# Crystals	8,385	3,983	5,772	2,101	8,108	6,576	2,101	38,889

Model	# Atom Species			Crystal System		
	MSE	MAE	$R^2$	MSE	MAE	$R^2$
<b>Energy ✗</b>						
MLP	0.811 (0.001)	0.196 (0.0001)	-0.155 (0.004)	0.769 (0.019)	0.192 (0.002)	0.048 (0.025)
Graph Network	0.610 (0.017)	0.162 (0.003)	0.162 (0.028)	0.523 (0.032)	0.149 (0.004)	0.348 (0.048)
E3NN	0.546 (0.007)	0.153 (0.001)	0.232 (0.005)	0.422 (0.005)	0.134 (0.001)	0.484 (0.012)
<b>Energy ✓</b>						
MLP	0.510 (0.005)	0.154 (0.001)	0.304 (0.004)	0.430 (0.006)	0.142 (0.001)	0.479 (0.004)
Graph Network	0.481 (0.011)	0.145 (0.001)	0.353 (0.004)	0.388 (0.005)	0.129 (0.001)	0.533 (0.014)
E3NN	0.528 (0.012)	0.153 (0.000)	0.263 (0.008)	0.414 (0.001)	0.133 (0.001)	0.497 (0.006)
DOSTransformer	<b>0.450</b> (0.008)	<b>0.134</b> (0.001)	<b>0.402</b> (0.011)	<b>0.380</b> (0.005)	<b>0.123</b> (0.002)	<b>0.540</b> (0.009)



# Result: Fine-tuning in OOD scenario 2

Fine-tuning on  
10 % training data

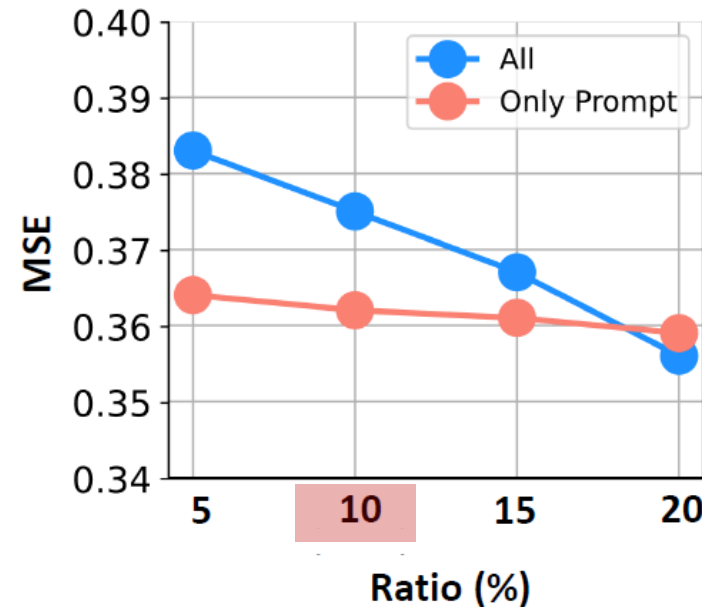


Fine-tuning all model  
parameters of DOSTransformer

Fine-tuning only the prompt and  
decoder of DOSTransformer

Model	MSE	MAE	$R^2$
<b>Energy ✗</b>			
MLP	0.762 (0.017)	0.190 (0.002)	0.042 (0.022)
Graph Network	0.504 (0.006)	0.150 (0.002)	0.371 (0.013)
E3NN	0.412 (0.002)	0.133 (0.001)	0.491 (0.002)
<b>Energy ✓</b>			
MLP	0.419 (0.002)	0.142 (0.001)	0.487 (0.006)
Graph Network	0.384 (0.001)	0.130 (0.001)	0.532 (0.005)
E3NN	0.413 (0.000)	0.134 (0.001)	0.494 (0.004)
<b>DOSTransformer</b>			
All	0.375 (0.009)	<b>0.123</b> (0.001)	0.543 (0.013)
Only Prompt	<b>0.362</b> (0.002)	<b>0.123</b> (0.001)	<b>0.559</b> (0.003)

Various training data ratio for Fine-tuning



- Additional fine-tuning achieves performance gain for all models
- “Only fine-tuning prompts” achieves more performance gain compared to fine-tuning the whole model
  - Fine-tuning only prompts enables the model to additionally learn from few new samples while fine-tuning all incur overfitting easily

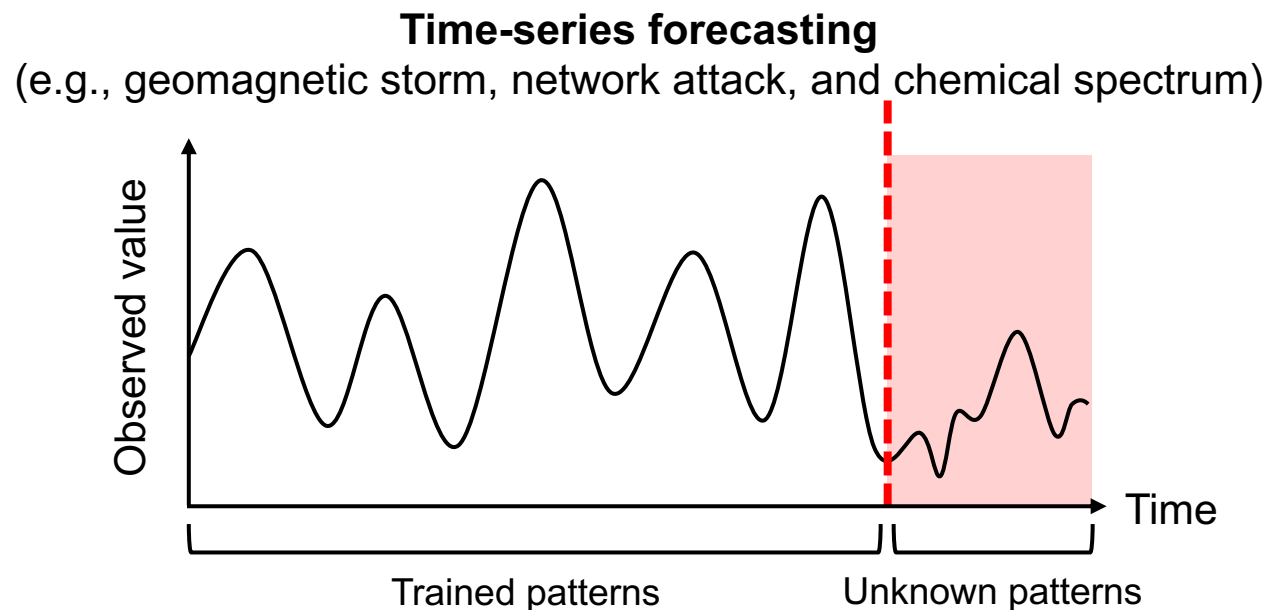
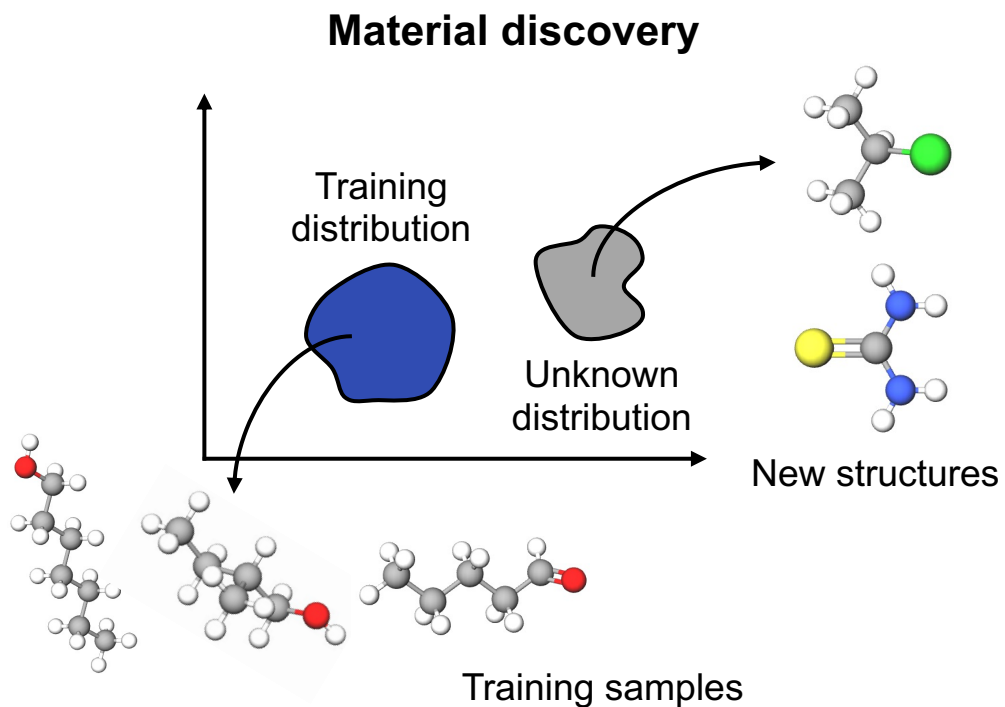


# Outline

- 그래프 신경망 개요 (20 mins)
  - 그래프 신경망 전반적인 소개
  - 그래프 종류에 따른 다양한 그래프 신경망 소개
- How to address Out-of-distribution problem (세부 기술 및 Q&A) (90~120 mins)
  - 소재 물성 예측 연구
    - 소재 물성 예측 연구 최신 동향 소개
    - Transformer 기반 모델 소개 → Prompt-based method
    - Extrapolation을 위한 모델 소개 → **Nonlinearity encoding-based method**
  - 물질 간 화학 반응 예측 연구
    - 물질 간 화학 반응 예측 연구 동향 소개
    - 정보 이론(Information bottleneck) 기반 모델 소개 → Information bottleneck-based method
    - 인과추론(Causal inference) 기반 모델 소개 → Causal inference-based method

# Introduction: Extrapolation

- **Goal:** Predict unseen data **outside the training distribution**
- Extrapolation is challenging because the input data usually follows an unknown distribution
- However, **extrapolation is common in scientific applications** in which discovering unobserved scientific knowledge is crucial



# Formal Definition of Extrapolation in Machine Learning

- **Given:** Prediction model  $f: \mathcal{X} \rightarrow \mathbb{R}$  trained on a training distribution  $\mathcal{D}$
- **Goal:** Minimize the following extrapolation error  $L_e$

$$\text{Extrapolation Error } L_e = \mathbb{E}_{(x,y) \sim \mathcal{X} \setminus \mathcal{D}} [L_s(y, f(x))]$$

Diagram illustrating the components of the extrapolation error formula:

- Extrapolation Error** (Gray box) is the overall quantity being minimized.
- $L_e$  (Gray box) is the extrapolation error.
- $\mathbb{E}$  (Black text) is the expectation operator.
- $(x,y)$  (Yellow and Blue boxes) is a sample from the data distribution.
- $\sim \mathcal{X} \setminus \mathcal{D}$  (Purple and Yellow boxes) indicates the sample is drawn from the data distribution excluding the training distribution.
- $L_s$  (Orange box) is the loss function (Cross entropy, MSE).
- $y$  (Blue box) is the target response.
- $f(x)$  (Green box) is the prediction model output.
- $x$  (Yellow box) is the input data.

Legend:

- Training distribution (Yellow box)
- Data distribution (Purple box)
- Prediction model (Green box)
- Loss function (Cross entropy, MSE) (Orange box)
- Input data (Yellow box)
- Target response (Blue box)

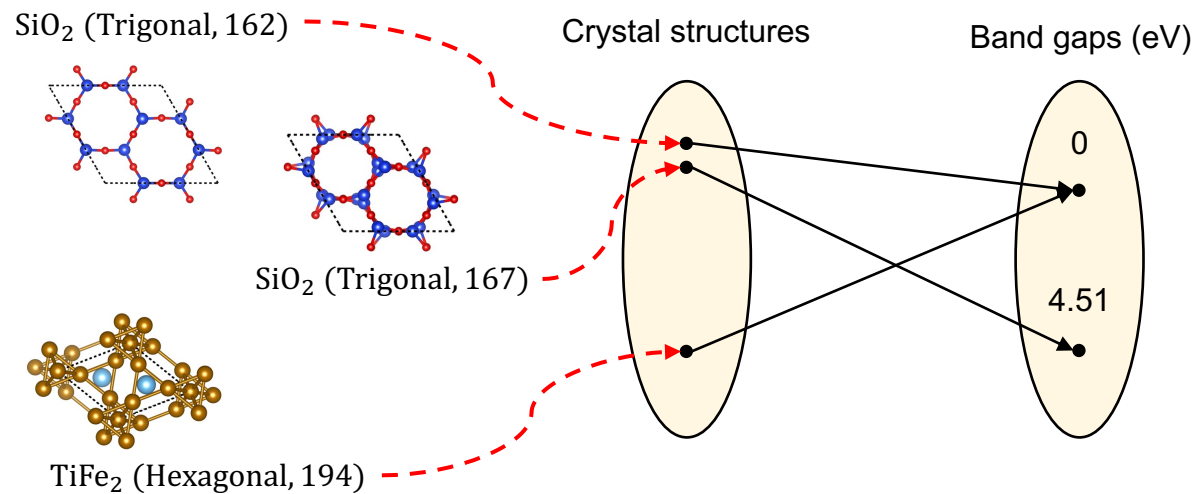
-  $(x,y)$ : A sample from out of training distribution  $\mathcal{X} \setminus \mathcal{D}$

- Machine learning achieved remarkable extrapolation performance in computer vision
- However, **extrapolation in scientific applications is still far from satisfactory**

# Why is Extrapolation Difficult in Scientific Data?

- **Nonlinear input-to-target relationship**

- Physical and chemical systems have severe **nonlinear relationships with their properties.**

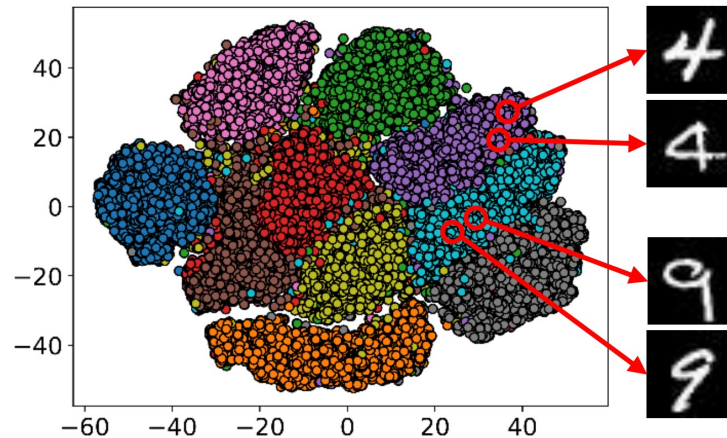


Two **similar structures** may have completely **different physical properties**,  
whereas two completely **different structures** may have **the same physical property**

# Image Dataset vs. Scientific Dataset

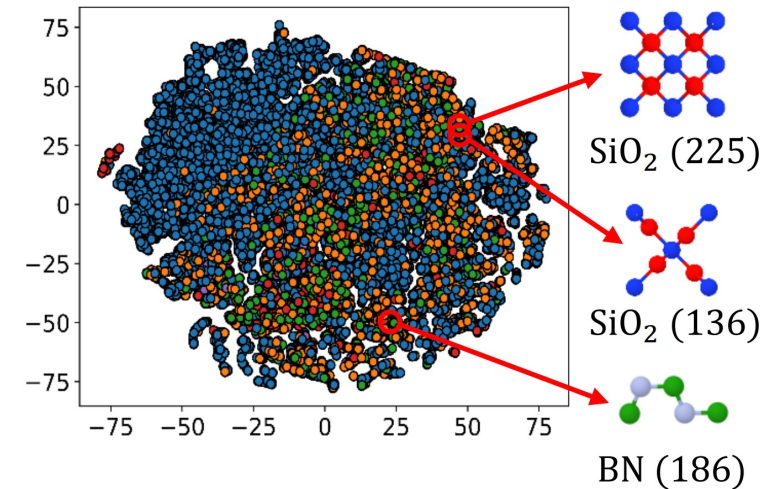
- T-SNE plots of MNIST and Material Project (MP) datasets
- Each point indicates an **image** or a **material** with **target response (label)** denoted by colors.
  - MNIST: class label
  - MP dataset: band gap

(a) MNIST dataset



Similar images share similar labels

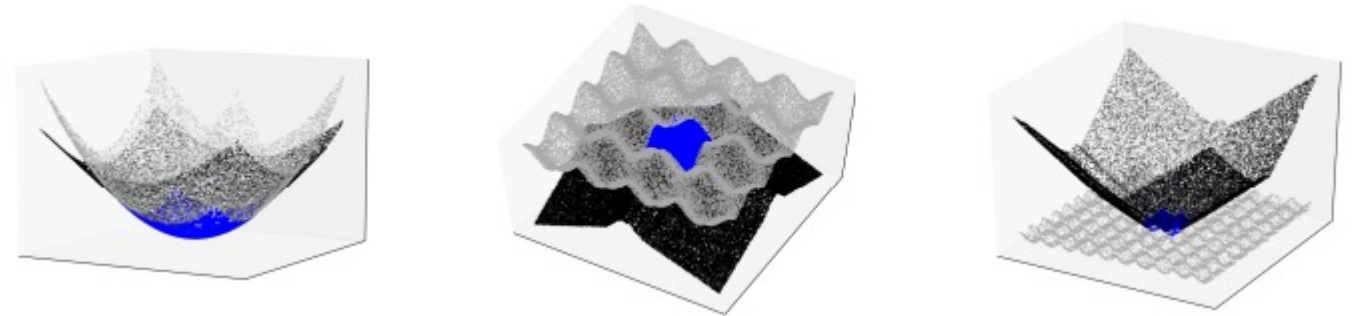
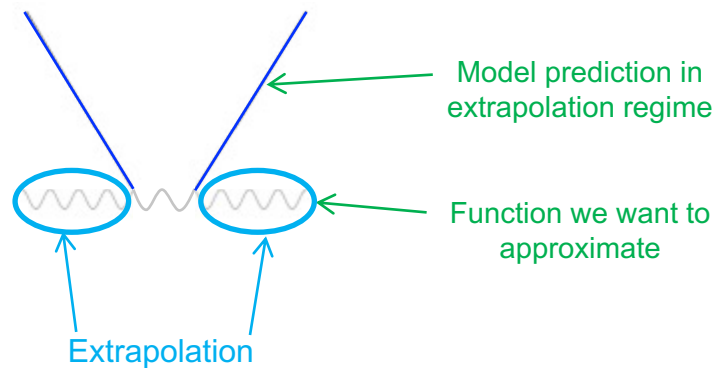
(b) MP dataset



Similar materials do not necessarily share similar labels

# How Neural Networks Extrapolate (Xu et al, ICLR21)

- **Theoretical findings in extrapolation:** Neural networks with ReLU → **simple linear regression in the extrapolation regime**



MLPs converge to linear functions outside the training data range

- **Proposed solution:** Remove nonlinearity from the data itself to linearize the problem
- **Limitation:** Requires domain knowledge to remove nonlinearity, and task-specific / data-specific

# Papers

## ■ Material property prediction

- Schnet: a continuous-filter convolutional neural network for modeling quantum interactions. NeurIPS 2017
- Neural message passing for quantum chemistry. ICML 2017
- Crystal Graph Convolutional Neural Networks for an Accurate and Interpretable Prediction of Material Properties. Phys. Rev. Lett. 2018
- Graph networks as a universal machine learning framework for molecules and crystals. Chem. Mater. 2019
- **Predicting Density of States via Multi-modal Transformer. ICLR Workshop 2023**

## ■ Extrapolation

- How Neural Networks Extrapolate: From Feedforward to Graph Neural Networks. ICLR 2021
- **Nonlinearity Encoding for Extrapolation of Neural Networks. KDD 2022**



# Nonlinearity Encoding for Extrapolation of Neural Networks

Gyoung S. Na<sup>1</sup> and Chanyoung Park<sup>2</sup>

<sup>1</sup>Korea Research Institute of Chemical Technology (KRICT), Republic of Korea

<sup>2</sup>Korea Advanced Institute of Science and Technology (KAIST), Republic of Korea

ngs0@kRICT.re.kr, cy.park@kaist.ac.kr

**KRICT**

**KAIST**

**DSAIL**

Data Science &  
Artificial Intelligence





# Related Work on Extrapolation

- **Representation learning**
  - Pros: Universally applicable method
  - Cons: Constraints on data distributions
- **Transfer learning**
  - Pros: Problem-specific methods, goal-directed learning
  - Cons: Source datasets, similar data distributions, re-training
- **Graph reformulation**
  - Pros: Easy to implement, theoretical backgrounds
  - Cons: Manual reformulation, white-box systems

Most existing studies mainly focus on **supporting extrapolation** rather than learning extrapolation models

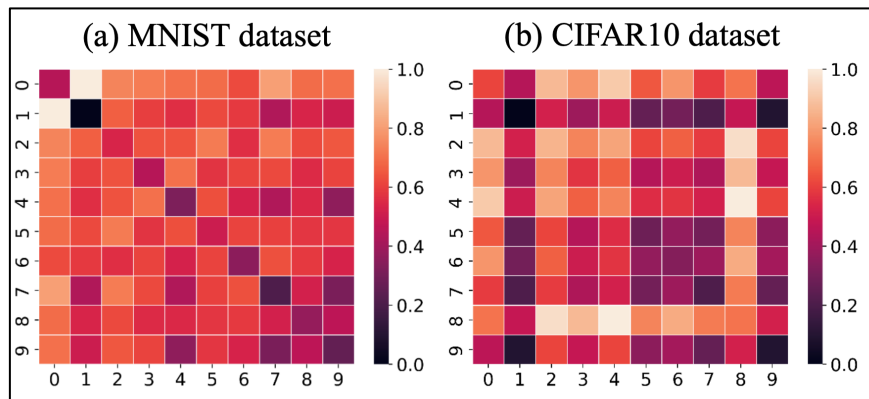
**Can we learn extrapolation** models?

# Can we learn extrapolation models?

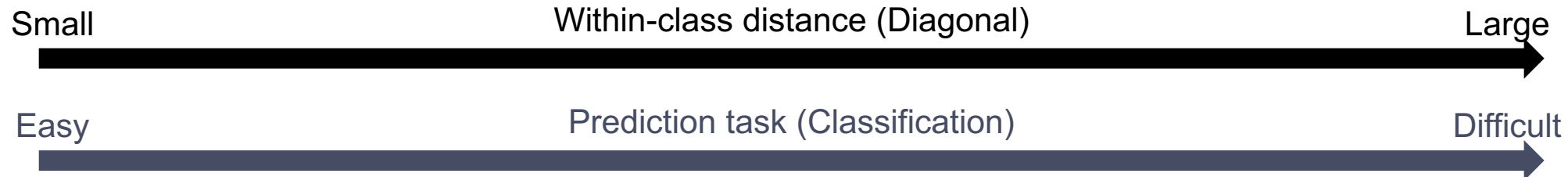
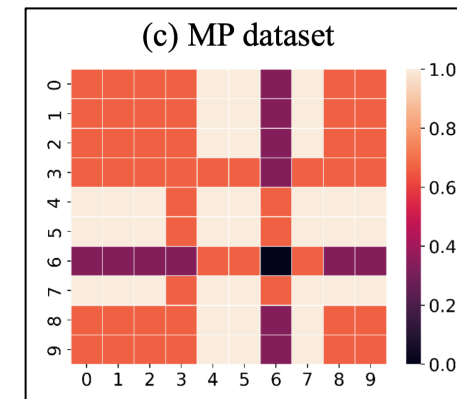
: Image Dataset vs. Scientific Dataset

- Heatmap visualization of **within-** and **between-class distances** on benchmark image and materials datasets

Image



Scientific dataset



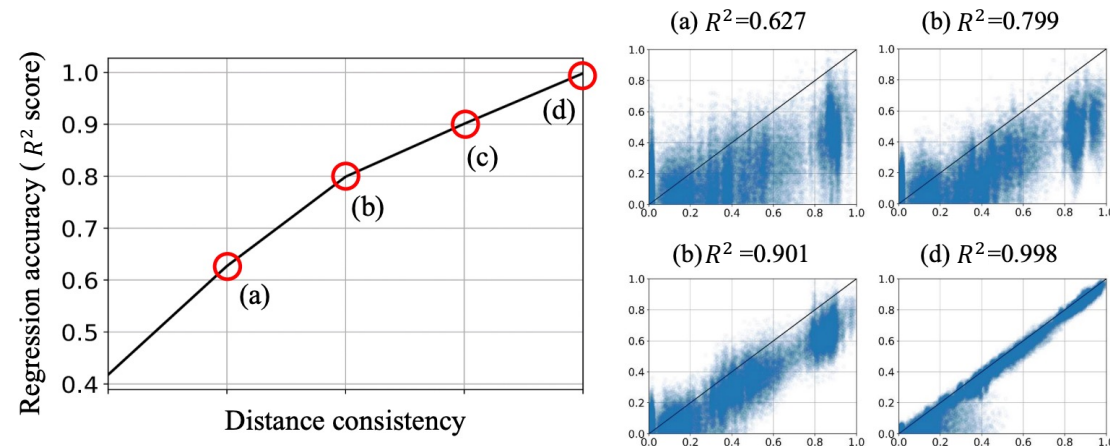
Prediction tasks can be made easier when,  
Two inputs with same label → Small input distance

**Distance  
Consistency!**

# Distance Consistency (DC)

- Consistency w.r.t. the distance between the inputs and their target responses
  - e.g., images > materials
- Extend our argument from classification to **regression**
  - Assume: Classification with infinite number of classes  $\approx$  regression

## Linear regression on synthetic datasets



High distance consistency  $\rightarrow$  High accuracy ( $R^2$  score)  $\rightarrow$  **Input-to-target relationship is made simple**

# Problem Reformulation of Extrapolation

- We reformulate the extrapolation problem as a **representation learning** problem aiming to **linearize the input-to-target relationships**



- **Our goal:** Increase the **distance consistency** aiming at **simplifying the input-to-target relationships**
  - **Given:** Two pairs of data samples  $(x_i, y_i), (x_j, y_j)$
  - **Define:** The distance between them

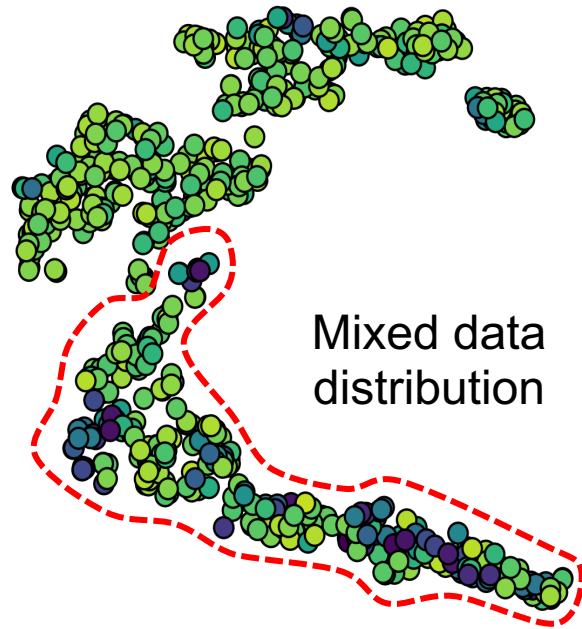
$$\begin{array}{ccc} \text{Dist. btw. targets} & & \\ d(d(x_i, x_j) - d(y_i, y_j)) & \xrightarrow{\text{Consider all } N^2 \text{ pairs}} & \sum_{i=1}^N \sum_{j=1}^N d(d(x_i, x_j) - d(y_i, y_j)) \\ \text{Dist. btw. inputs} & & \end{array}$$

We adopt **Wasserstein distance** to measure the distance consistency between input and target

# Problem Definition of Nonlinearity Encoding

- **Our method:** Automatic Nonlinearity Encoding (ANE)

Data distribution in the **original feature space**

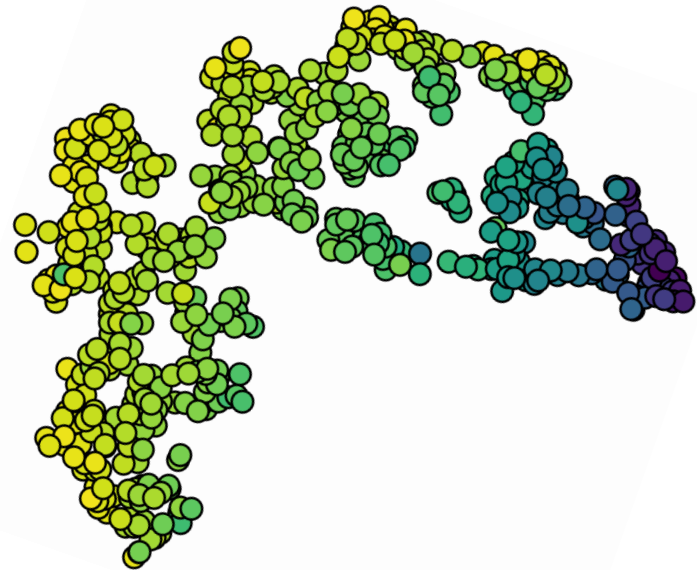


Hard

Nonlinearity  
Encoding



Data distribution in the **embedding space of ANE**



Easy

# Optimization: Decomposition of Lagrangian

- Our problem can be defined as follows:

$$\theta^* = \operatorname{argmin}_{\theta} \sum_{i=1}^N \sum_{j=1}^N \inf_{\pi \in \Pi} \int_{\mathcal{M} \times \mathcal{M}} \|r_{ij} - u_{ij}\|_p \pi(r_{ij}, u_{ij}) dr du$$

# training data

**Joint optimization  
w.r.t.  $\theta$  and  $\pi$**

- $r_{ij} = d(\phi(\mathbf{x}_i; \theta), \phi(\mathbf{x}_j; \theta))$ : **Dist. btw input data** in embedding space
- $u_{ij} = d(y_i, y_j)$ : **Dist. btw target data**

- We can define a **Lagrangian of the objective function** as (refer to **Kantorovich-Rubinstein duality**):

$$\begin{aligned} L_W = & \sum_{(i,j) \in \mathcal{N}} \sum_{(k,q) \in \mathcal{N} \setminus I_{ij}} (\|r_{ij} - u_{kq}\| - f(r_{ij}) - g(u_{kq})) \pi(r_{ij}, u_{kq}) + \sum_{(i,j) \in \mathcal{N}} \sum_{(k,q) \in \mathcal{N} \setminus I_{ij}} \|r_{ij} - u_{kq}\| \pi(r_{ij}, u_{kq}) \\ & + \sum_{(i,j) \in \mathcal{N}} (p(r_{ij}) - \sum_{(k,q) \in I_{ij}} \pi(r_{ij}, u_{kq})) f(r_{ij}) + \sum_{(i,j) \in \mathcal{N}} (p(u_{ij}) - \sum_{(k,q) \in \mathcal{N}} \pi(r_{kq}, u_{ij})) g(u_{ij}) + \sum_{(i,j) \in \mathcal{N}} \sum_{(k,q) \in \mathcal{N} \setminus I_{ij}} \pi(r_{kq}, u_{ij}) g(u_{ij}), \end{aligned}$$

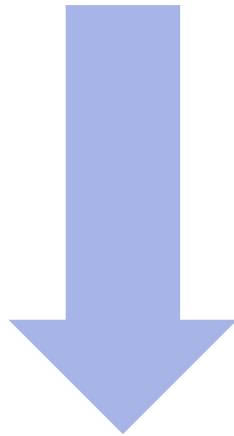
where  $\mathcal{N} = \{(i, j) \mid \text{for all } i, j \in \{1, 2, \dots, N\}\}$ , and  $I_{ij} = \{(k, q) \mid u_{ij} = u_{kq} \text{ for } (k, q) \in \mathcal{N}\}$ .

Pairs with the same target distance

# Optimization: Model Parameter Optimization

- In the end, the representation learning problem to encode the nonlinearity is given by:

$$\theta^* = \operatorname{argmin}_{\theta} \sum_{i=1}^N \sum_{j=1}^N \inf_{\pi \in \Pi} \int_{\mathcal{M} \times \mathcal{M}} \|r_{ij} - u_{ij}\|_p \pi(r_{ij}, u_{ij}) dr du$$



- $r_{ij} = d(\phi(\mathbf{x}_i; \theta), \phi(\mathbf{x}_j; \theta))$ : **Dist. btw input data** in embedding space
- $u_{ij} = d(y_i, y_j)$ : **Dist. btw target data**

$$\theta^* = \operatorname{argmin}_{\theta} \sum_{i=1}^N \sum_{j=1}^N \|r_{ij} - u_{ij}\|$$

Enforce distance consistency  
between data pairs!

# Optimization: Model Parameter Optimization

## Training of ANE-based prediction model

**Input** : Training dataset  $\mathcal{D} = \{(\mathbf{x}_1, \mathbf{y}_1), \dots, (\mathbf{x}_N, \mathbf{y}_N)\}$ ;  
Embedding network  $\phi(\mathbf{x}; \boldsymbol{\theta})$ ; Prediction model  
 $f(\phi(\mathbf{x}; \boldsymbol{\theta}); \boldsymbol{\mu})$ ; Sampling method  $\psi(\mathbf{x}; \mathcal{D})$ ; Distance  
metric  $d$

```
1 repeat
2   for  $i = 1; i < N; i++$  do
3      $s = \psi(\mathbf{x}_i; \mathcal{D})$  // List of indices of the samples.
4     for  $j = 1; j < |s|; j++$  do
5        $r_{ij} = d(\phi(\mathbf{x}_i; \boldsymbol{\theta}), \phi(\mathbf{x}_{s_j}; \boldsymbol{\theta}))$  and  $u_{ij} = d(\mathbf{y}_i, \mathbf{y}_{s_j})$ 
6        $L_W += ||r_{ij} - u_{ij}||_2$ 
7     end
8   end
9   Optimize  $\boldsymbol{\theta}$  with respect to  $L_W$ .
10 until  $\boldsymbol{\theta}$  converged;
11 Optimize  $\boldsymbol{\mu}$  on  $\mathcal{Z} = \{(\phi(\mathbf{x}_1; \boldsymbol{\theta}^*), \mathbf{y}_1), \dots, (\phi(\mathbf{x}_N; \boldsymbol{\theta}^*), \mathbf{y}_N)\}$ .
12 Return  $\phi(\mathbf{x}; \boldsymbol{\theta}^*)$  and  $f(\phi(\mathbf{x}; \boldsymbol{\theta}^*); \boldsymbol{\mu}^*)$ 
```

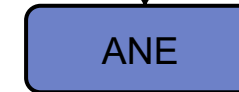
ANE

Prediction  
model

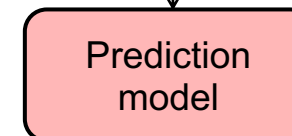


Training dataset  
 $\mathcal{D} = \{(\mathbf{x}_1, \mathbf{y}_1), \dots, (\mathbf{x}_N, \mathbf{y}_N)\}$

Data-agnostic!



Training dataset with  
nonlinearity encoding  
 $\mathcal{Z} = \{(\phi(\mathbf{x}_1; \boldsymbol{\theta}^*), \mathbf{y}_1), \dots, (\phi(\mathbf{x}_N; \boldsymbol{\theta}^*), \mathbf{y}_N)\}$



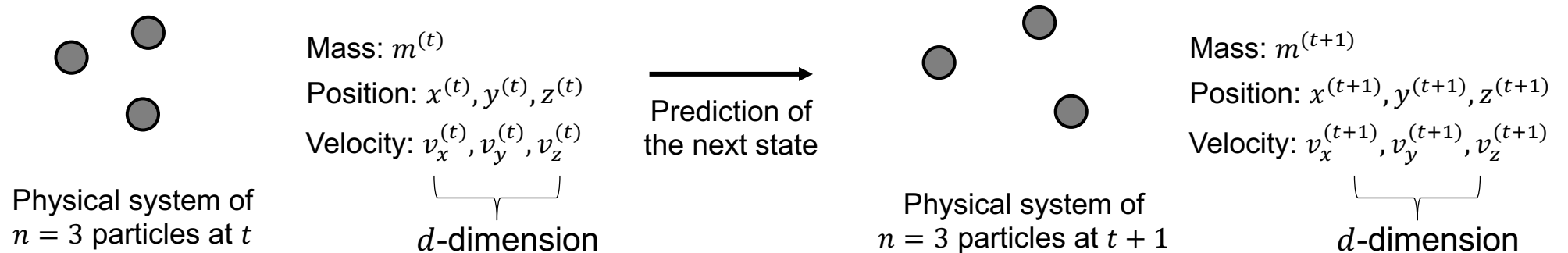


# Experiments

- Matrix-shaped data
- Graph-structured data
- Real-world data (Discovering Solar Cell Materials)

# Extrapolation on Matrix-Shaped Data: $n$ -Body Problem (1/3)

- **Task:** Given mass, position, and velocity of  $n$  particles, estimate future velocities of  $n$  particles



- **Data preprocessing:** 3-dimensional 3-body problem.  $\mathbf{x}_t \in \mathbb{R}^{3 \times 7}$  and  $\mathbf{y}_t \in \mathbb{R}^{3 \times 3} \leftarrow$  Matrix-shaped data
  - Simulated 10 datasets
  - **Train:** Observations in time  $[0, 80]$
  - **Test:** Predict velocity in future time  $(80, 100]$

# Extrapolation on Matrix-Shaped Data: $n$ -Body Problem (2/3)

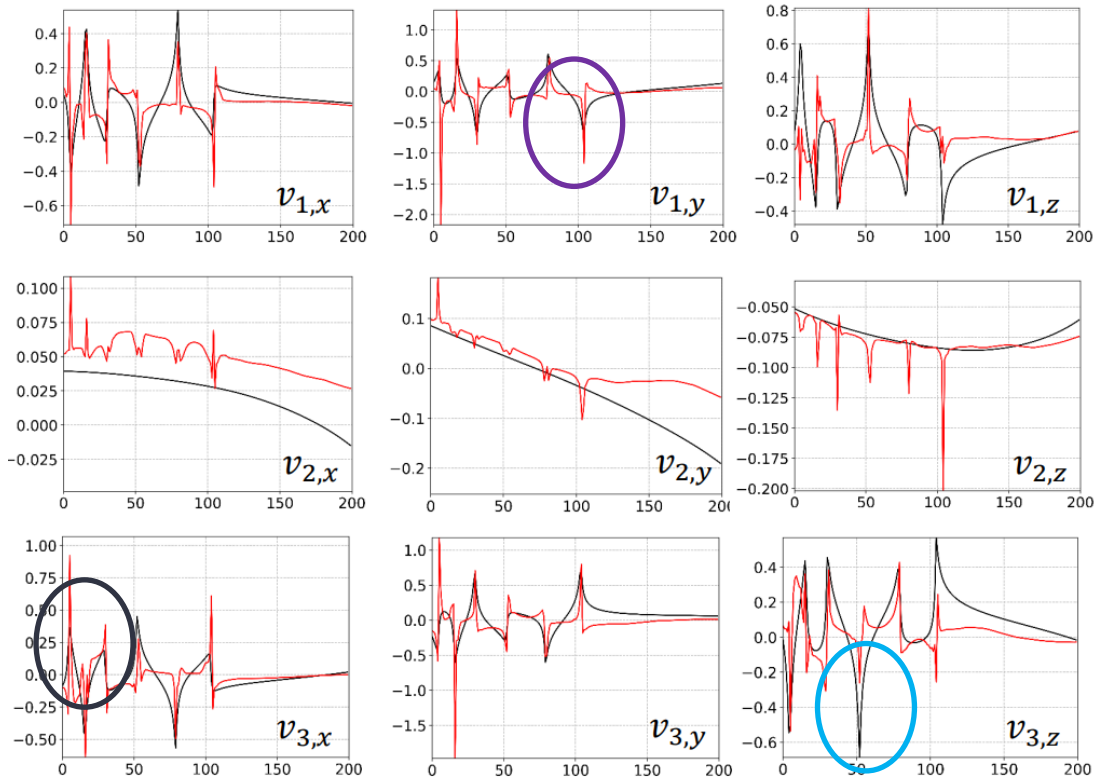
- **Metric:** Distance correlation (Corr) between the simulated (ground-truth) and predicted velocities
  - To measure how accurately the models predict future **trends** of the velocities

	Direct prediction method	GNN-based methods			Metric learning-based method		
Idx.	NBNet	GIN	MPNN	UMP	LRL-F	SLRL-F	ANE-F
1	0.32	0.54	0.35	0.25	0.43	0.53	<b>0.18</b>
2	0.49	0.54	0.53	<b>0.36</b>	0.52	0.49	0.45
3	0.57	0.54	0.53	0.46	0.52	0.59	<b>0.29</b>
4	0.25	0.68	0.26	0.26	0.09	0.07	<b>0.03</b>
5	0.66	0.93	0.71	0.69	0.85	0.65	<b>0.49</b>
6	<b>0.11</b>	0.22	0.17	0.16	0.12	0.12	0.02
7	0.75	0.94	0.63	0.67	0.61	0.44	<b>0.40</b>
8	0.44	0.85	0.26	0.29	0.27	0.38	<b>0.15</b>
9	0.39	0.26	0.10	0.70	0.18	0.40	<b>0.03</b>
10	0.64	0.72	0.55	0.54	0.53	0.37	<b>0.27</b>
mean	0.46	0.62	0.41	0.44	0.41	0.40	<b>0.23</b>
±std.	±0.19	±0.24	±0.20	±0.19	±0.23	±0.18	<b>±0.17</b>

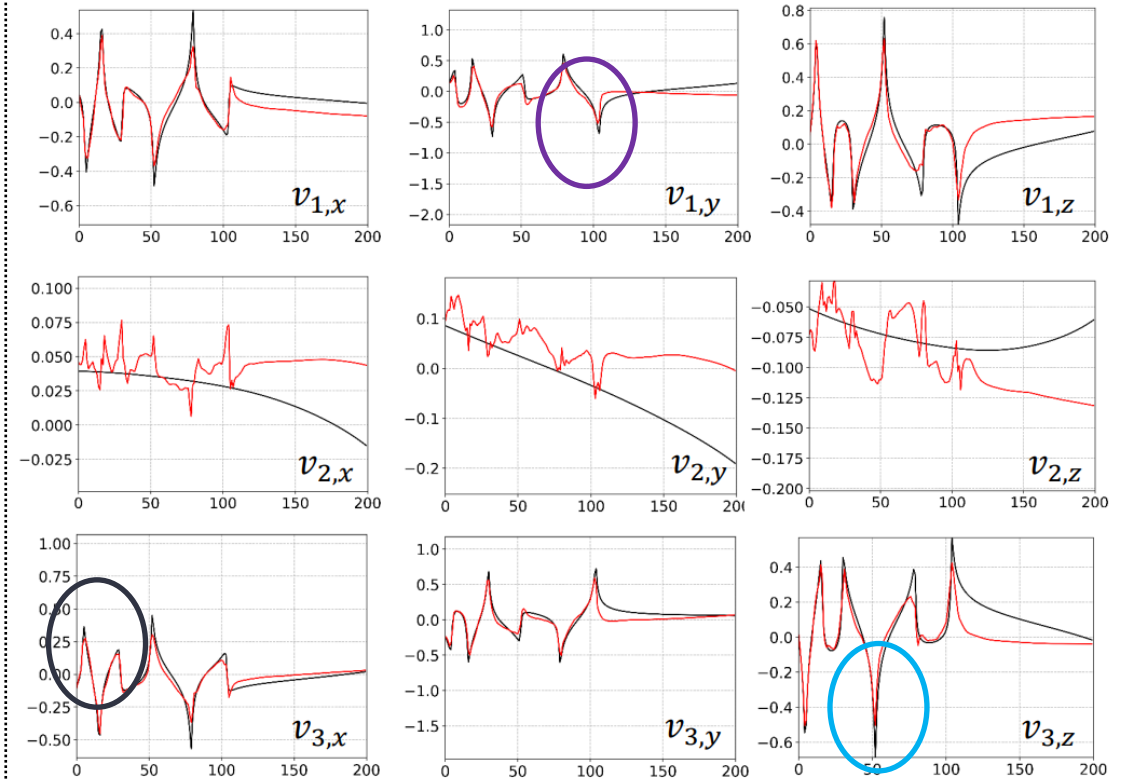
ANE generates input representations that are the most effective to reducing the extrapolation errors

# Extrapolation on Matrix-Shaped Data: $n$ -Body Problem (3/3)

State-of-the-art GNN-based method



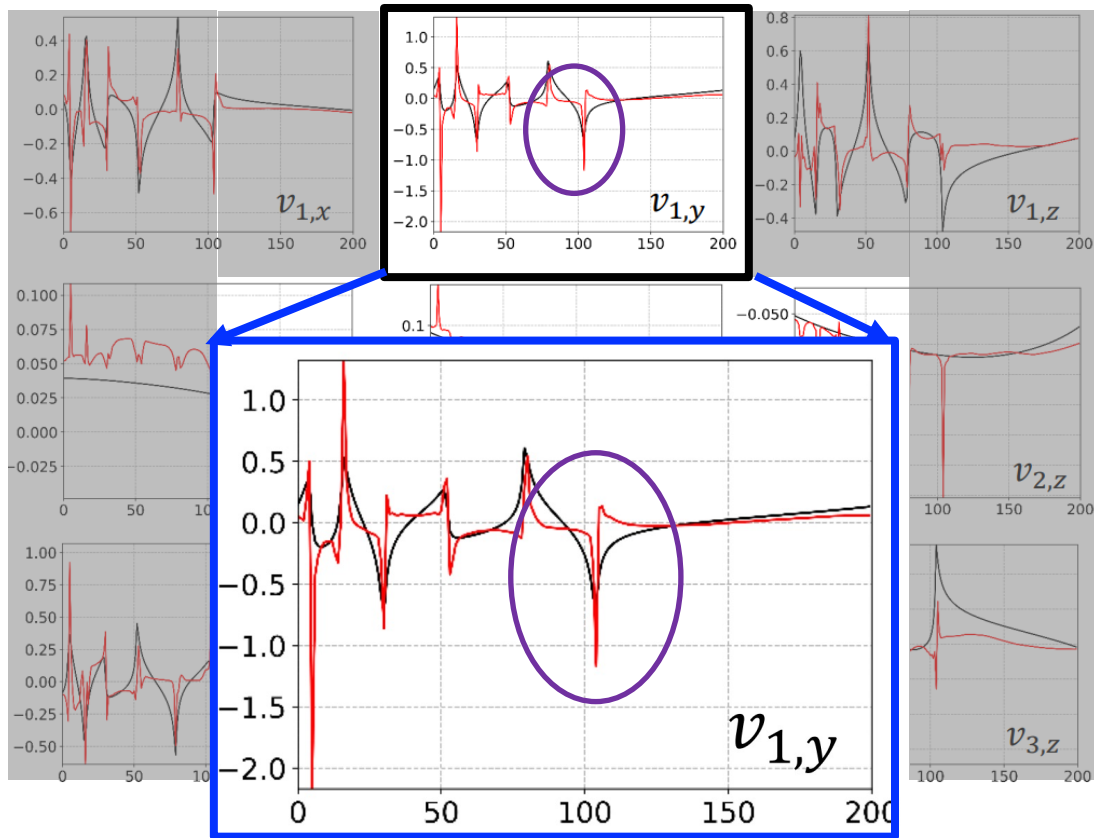
Ours (ANE-F)



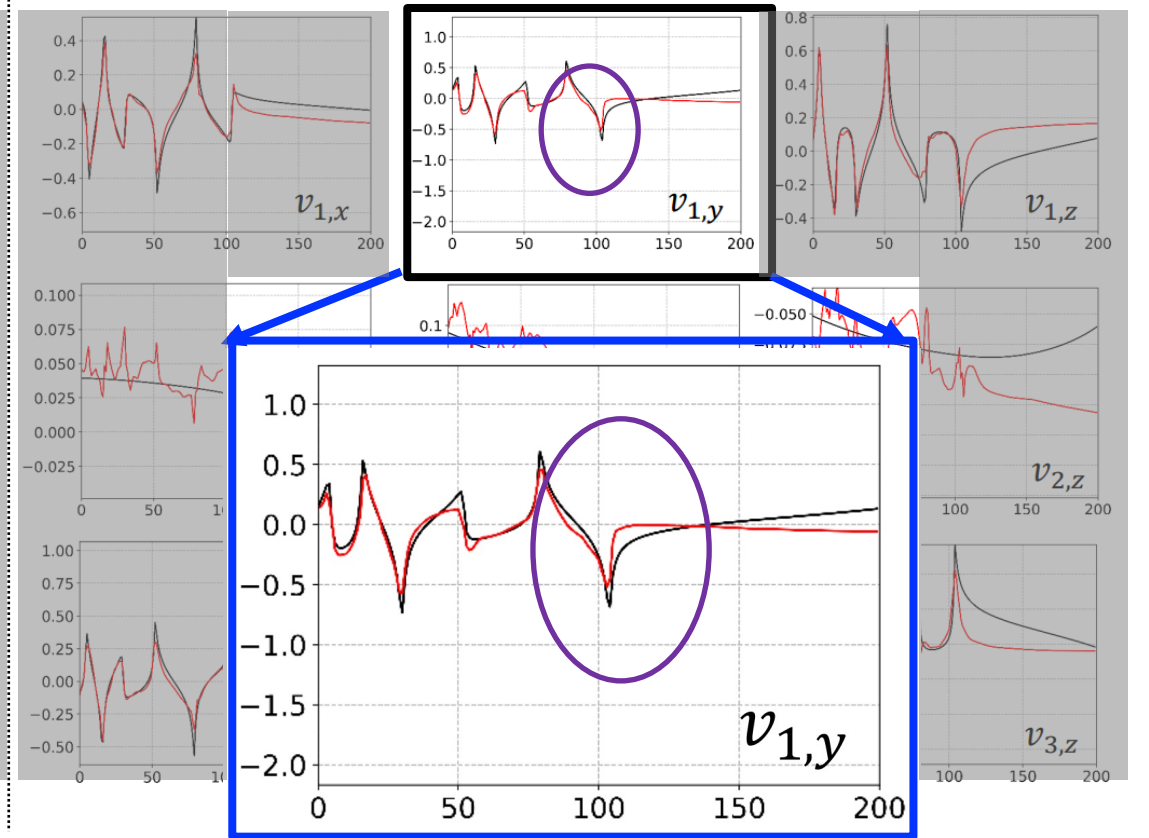
— Simulated velocity (ground truth) — Predicted velocity

# Extrapolation on Matrix-Shaped Data: $n$ -Body Problem (3/3)

State-of-the-art GNN-based method



Ours (ANE-F)

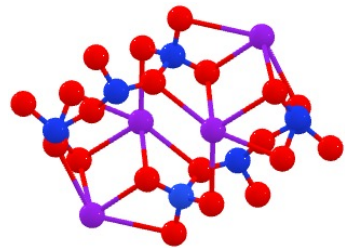


— Simulated velocity (ground truth) — Predicted velocity

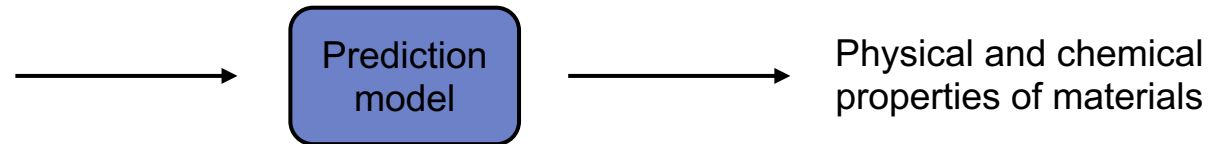
ANE is better at predicting **sudden explosions** of velocity

# Extrapolation on Graph-Structured Data: Materials Property Prediction

- **Task:** Predict four material properties (Formation energy, Band gap, Shear modulus, Bulk modulus)
  - Discovering novel materials is a fundamental task in various fields (e.g., semiconductor and renewable energy)



$\mathcal{V}$ : A set of nodes (atoms)  
 $\mathcal{U}$ : A set of edges (bondings)  
 $\mathbf{X}$ : Node feature matrix  
 $\mathbf{E}$ : Edge feature matrix



A material can be represented as an attributed graph  $G = (\mathcal{V}, \mathcal{U}, \mathbf{X}, \mathbf{E})$ .

- **Data preprocessing**
  - MPS dataset: Benchmark materials dataset containing 3,162 materials
  - **Train**: Materials that contain **only two types of elements** (i.e., Binary materials)
  - **Test**: Materials that contain **three/four types of elements** (i.e., Ternary and quaternary materials)

# Extrapolation on Graph-Structured Data: Materials Property Prediction

- **Metric:**  $R^2$  score

Method	Formation Energy	Band Gap	Shear Modulus	Bulk Modulus
GCN	0.662 ( $\pm 0.019$ )	0.254 ( $\pm 0.071$ )	0.526 ( $\pm 0.025$ )	0.574 ( $\pm 0.037$ )
MPNN	0.072 ( $\pm 0.052$ )	N/A	0.352 ( $\pm 0.344$ )	0.714 ( $\pm 0.007$ )
CGCNN	N/A	0.163 ( $\pm 0.424$ )	0.405 ( $\pm 0.441$ )	0.732 ( $\pm 0.011$ )
UMP	0.763 ( $\pm 0.042$ )	0.351 ( $\pm 0.069$ )	0.552 ( $\pm 0.003$ )	0.707 ( $\pm 0.022$ )
LRL-MPNN	0.819 ( $\pm 0.024$ )	0.259 ( $\pm 0.034$ )	0.704 ( $\pm 0.009$ )	0.769 ( $\pm 0.021$ )
SLRL-MPNN	0.841 ( $\pm 0.018$ )	0.396 ( $\pm 0.052$ )	0.693 ( $\pm 0.013$ )	0.767 ( $\pm 0.007$ )
ANE-MPNN	<b>0.879</b> <b>(<math>\pm 0.017</math>)</b>	<b>0.447</b> <b>(<math>\pm 0.055</math>)</b>	<b>0.716</b> <b>(<math>\pm 0.015</math>)</b>	<b>0.790</b> <b>(<math>\pm 0.011</math>)</b>

ANE-MPNN outperforms state-of-the-art GNNs and metric learning methods

# ANE for Discovering Solar Cell Materials

- **Task:** Predict band gaps of perovskites
  - c.f.) Perovskite has received significant attention as solar cell materials for renewable energy
  - Infer materials properties of crystal structures containing **unseen elemental combinations**
- **Data preprocessing**
  - Divided HOIP dataset by eliminating the materials that contain specific elements
    - **HOIP-HIGH:** HOIP – (Germanium (Ge) and Fluorine (F))
    - **HOIP-LOW:** HOIP – (Lead (Pb) and Iodine (I))
  - Range of band gaps between training and test data is completely different

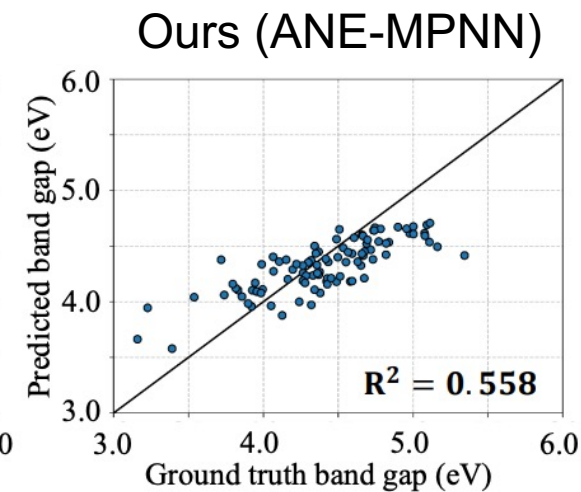
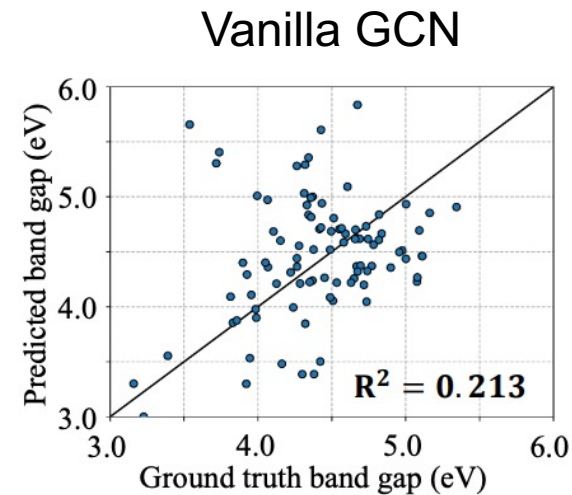


# ANE for Discovering Solar Cell Materials

- **Metric:**  $R^2$  score

N/A: negative  $R^2$

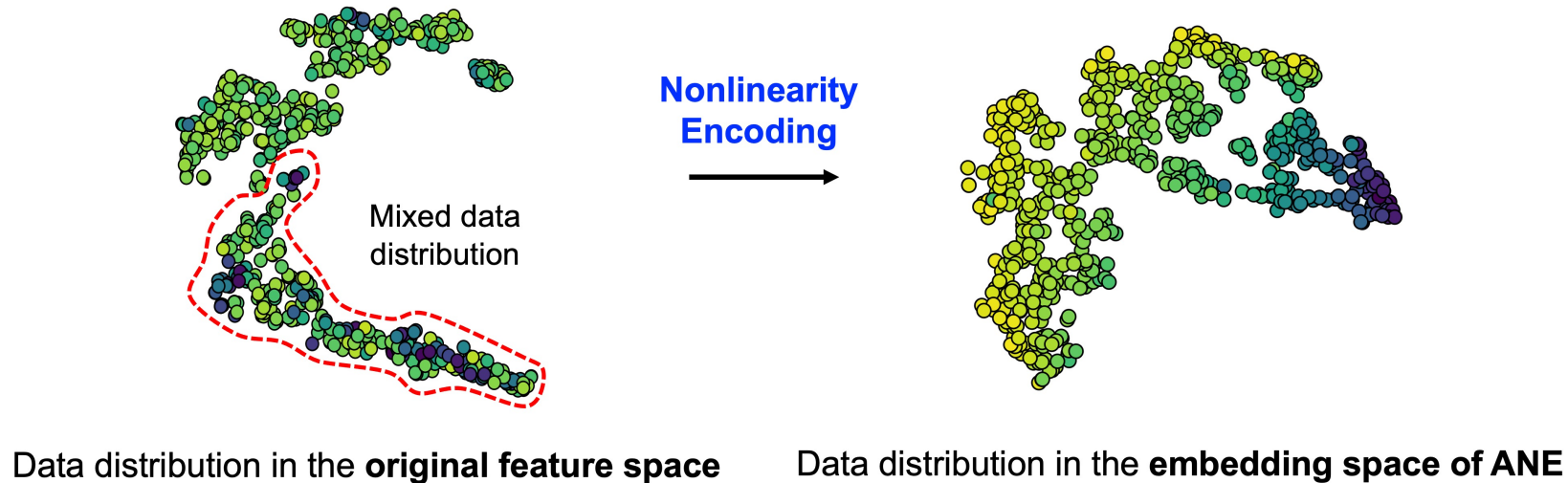
	Method	Dataset	
		HOIP-HIGH	HOIP-LOW
GNN methods	GCN	0.213( $\pm 0.162$ )	N/A
	MPNN	N/A	N/A
	CGCNN	N/A	N/A
	UMP	N/A	N/A
DML methods	LRL-MPNN	N/A	0.521( $\pm 0.131$ )
	SLRL-MPNN	0.182( $\pm 0.160$ )	0.486( $\pm 0.096$ )
	ANE-MPNN	<b>0.558(<math>\pm 0.044</math>)</b>	<b>0.664(<math>\pm 0.071</math>)</b>



ANE-MPNN roughly captured the relationships, while GCN fails to do so

# Conclusion

- Proposed a **data-agnostic** embedding method for improving the **extrapolation capabilities** of ML



- Maximized **distance consistency** between the inputs and their targets (Based on **Wasserstein distance**)
  - The **distance between two inputs** should be **determined based on the distance between their targets**
- Demonstrated the effectiveness in **various scientific applications of various data formats**

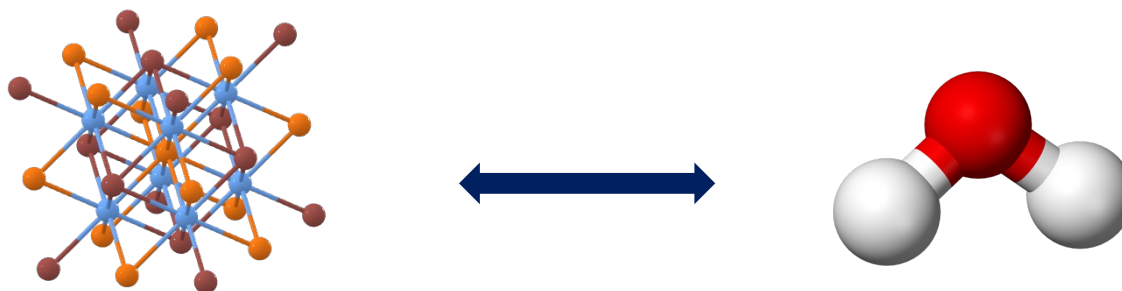
# Outline

- 그래프 신경망 개요 (20 mins)
  - 그래프 신경망 전반적인 소개
  - 그래프 종류에 따른 다양한 그래프 신경망 소개
- How to address Out-of-distribution problem (세부 기술 및 Q&A) (90~120 mins)
  - 소재 물성 예측 연구
    - 소재 물성 예측 연구 최신 동향 소개
    - Transformer 기반 모델 소개 → Prompt-based method
    - Extrapolation을 위한 모델 소개 → Nonlinearity encoding-based method
  - 물질 간 화학 반응 예측 연구
    - 물질 간 화학 반응 예측 연구 동향 소개
    - 정보 이론(Information bottleneck) 기반 모델 소개 → Information bottleneck-based method
    - 인과추론(Causal inference) 기반 모델 소개 → Causal inference-based method

# Introduction: Relational Learning

## ▪ Molecular Relational Learning

- Learn the interaction behavior between a pair of molecules



### • Examples

- Predicting **optical properties** when a chromophore (Chromophore) and solvent (Solvent) react
- Predicting **solubility** when a solute and solvent react
- Predicting **side effects** when taking two types of drugs simultaneously (Polypharmacy effect)

# Papers

## ■ General

- Modeling polypharmacy side effects with graph convolutional networks. Bioinformatics 2018
- Chemically interpretable graph interaction network for prediction of pharmacokinetic properties of drug-like molecules. AAAI 2020
- Multi-view graph contrastive representation learning for drug-drug interaction prediction. WWW 2021

## ■ Information bottleneck-based

- Graph information bottleneck for subgraph recognition. ICLR 2021
- Interpretable and generalizable graph learning via stochastic attention mechanism. ICML 2022
- Improving subgraph recognition with variational graph information bottleneck. CVPR 2022
- **Conditional Graph Information Bottleneck for Molecular Relational Learning. ICML 2023**

## ■ Causal inference-based

- Discovering invariant rationales for graph neural networks. ICLR 2022
- Debiasing Graph Neural Networks via Learning Disentangled Causal Substructure. NeurIPS 2022
- Causal attention for interpretable and generalizable graph classification. KDD 2022
- **Shift-robust molecular relational learning with causal substructure. KDD 2023**

# Papers

## ■ General

- Modeling polypharmacy side effects with graph convolutional networks. Bioinformatics 2018
- Chemically interpretable graph interaction network for prediction of pharmacokinetic properties of drug-like molecules. AAAI 2020
- Multi-view graph contrastive representation learning for drug-drug interaction prediction. WWW 2021

## ■ Information bottleneck-based

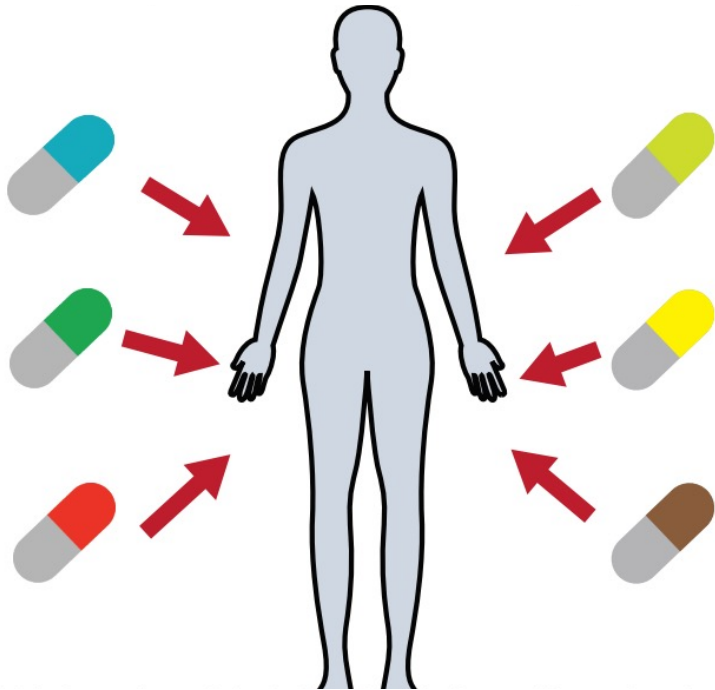
- Graph information bottleneck for subgraph recognition. ICLR 2021
- Interpretable and generalizable graph learning via stochastic attention mechanism. ICML 2022
- Improving subgraph recognition with variational graph information bottleneck. CVPR 2022
- **Conditional Graph Information Bottleneck for Molecular Relational Learning. ICML 2023**

## ■ Causal inference-based

- Discovering invariant rationales for graph neural networks. ICLR 2022
- Debiasing Graph Neural Networks via Learning Disentangled Causal Substructure. NeurIPS 2022
- Causal attention for interpretable and generalizable graph classification. KDD 2022
- **Shift-robust molecular relational learning with causal substructure. KDD 2023**

# Polypharmacy side effect

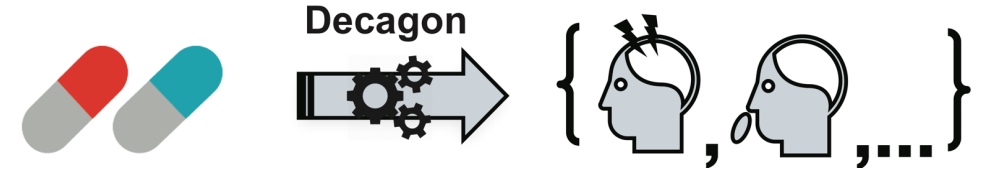
- Many patients **take multiple drugs** to treat complex or co-existing diseases
  - 25% of people ages 65-69 take more than 5 drugs
  - 46% of people ages 70-79 take more than 5 drugs
  - Many patients take more than 20 drugs to treat heart disease, depression, insomnia, etc.



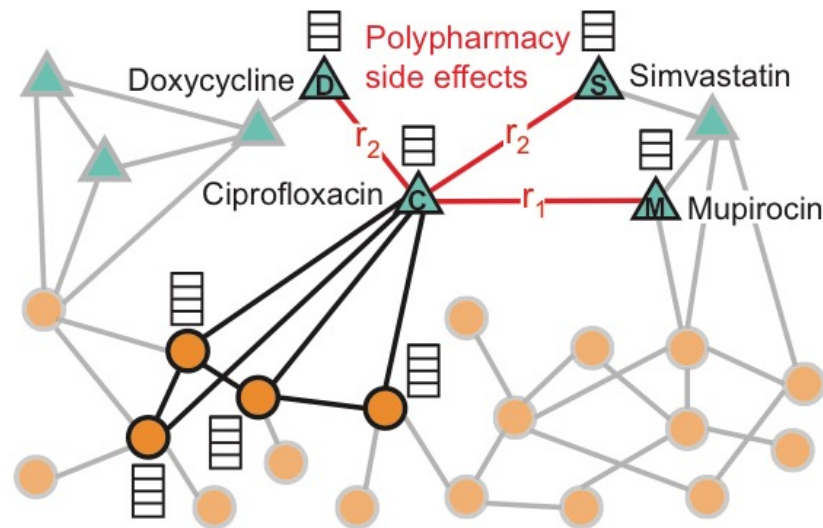
- Extremely difficult to identify
  - Impossible to test all combinations of drugs
  - Side effects not observed in controlled trials
- 15% of the U.S. population affected
  - Annual costs exceed \$177 billion

# Decagon: Overview

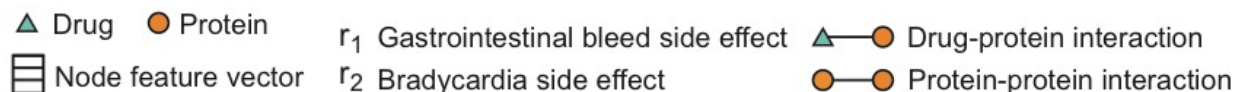
Given a drug pair, predict side effects of that drug pair



- Task: Predicting polypharmacy side-effect (Drug-drug interaction)
- Idea: Construct a multi-modal graph of following relations
  - 1. Protein-protein interaction
  - 2. Drug-protein interaction
  - 3. Drug-drug interaction (polypharmacy side effects; each side effect is an edge of a different type)



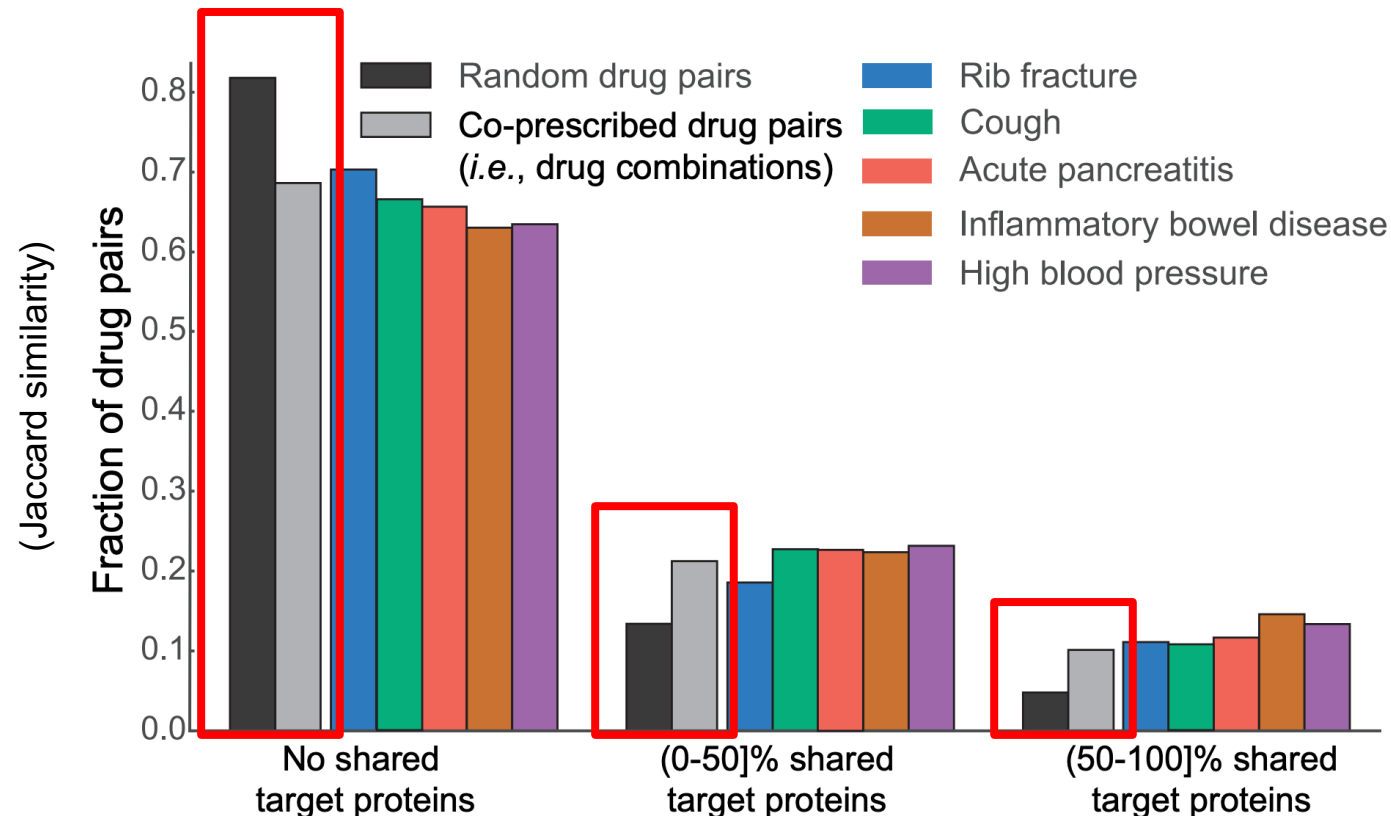
Multi-relational edge prediction model



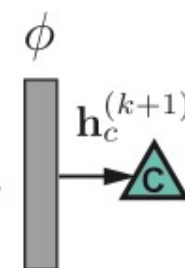
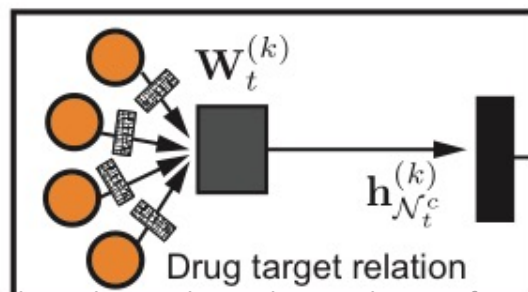
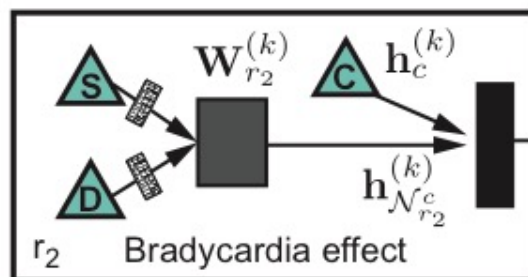
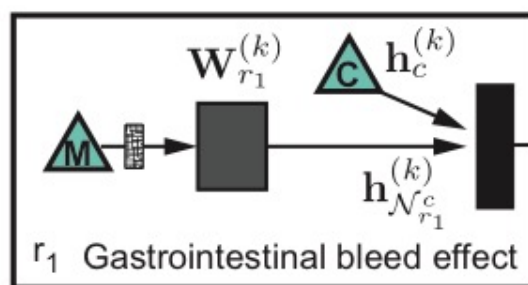
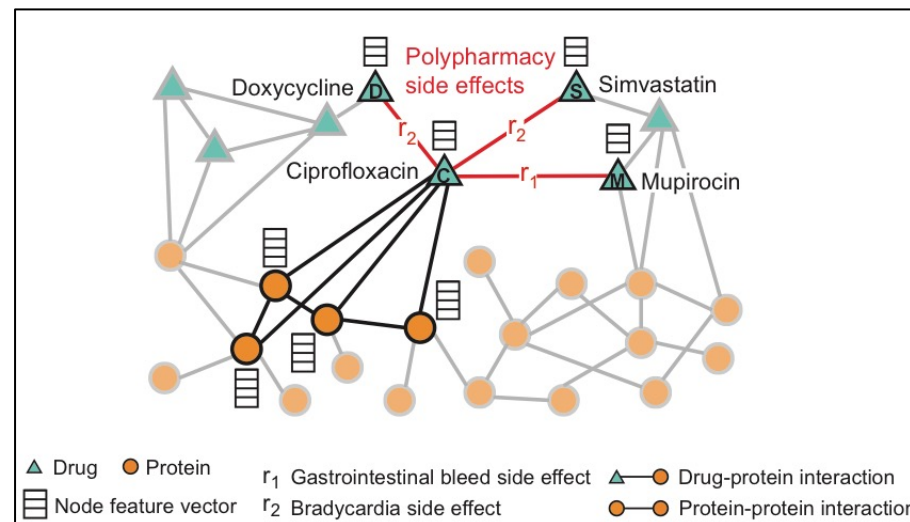


# Decagon: Exploratory Data Analysis (EDA)

- Observation: Co-prescribed drugs (i.e. drug combinations) tend to have more target proteins in common than random drug pairs
  - It is important to consider how proteins interact with each other and to be able to model longer chains of (indirect) interactions.



- **Encoder:** GCN operating on the graph and produces embeddings for nodes
- **Decoder:** Tensor factorization model using these embeddings to model polypharmacy side effects



Query drug pair



$D_{r_1}$

$D_{r_2}$

$D_{r_n}$

Predictions

$$p(C, r_1, S)$$

$$p(C, r_2, S)$$

$$p(C, r_3, S)$$

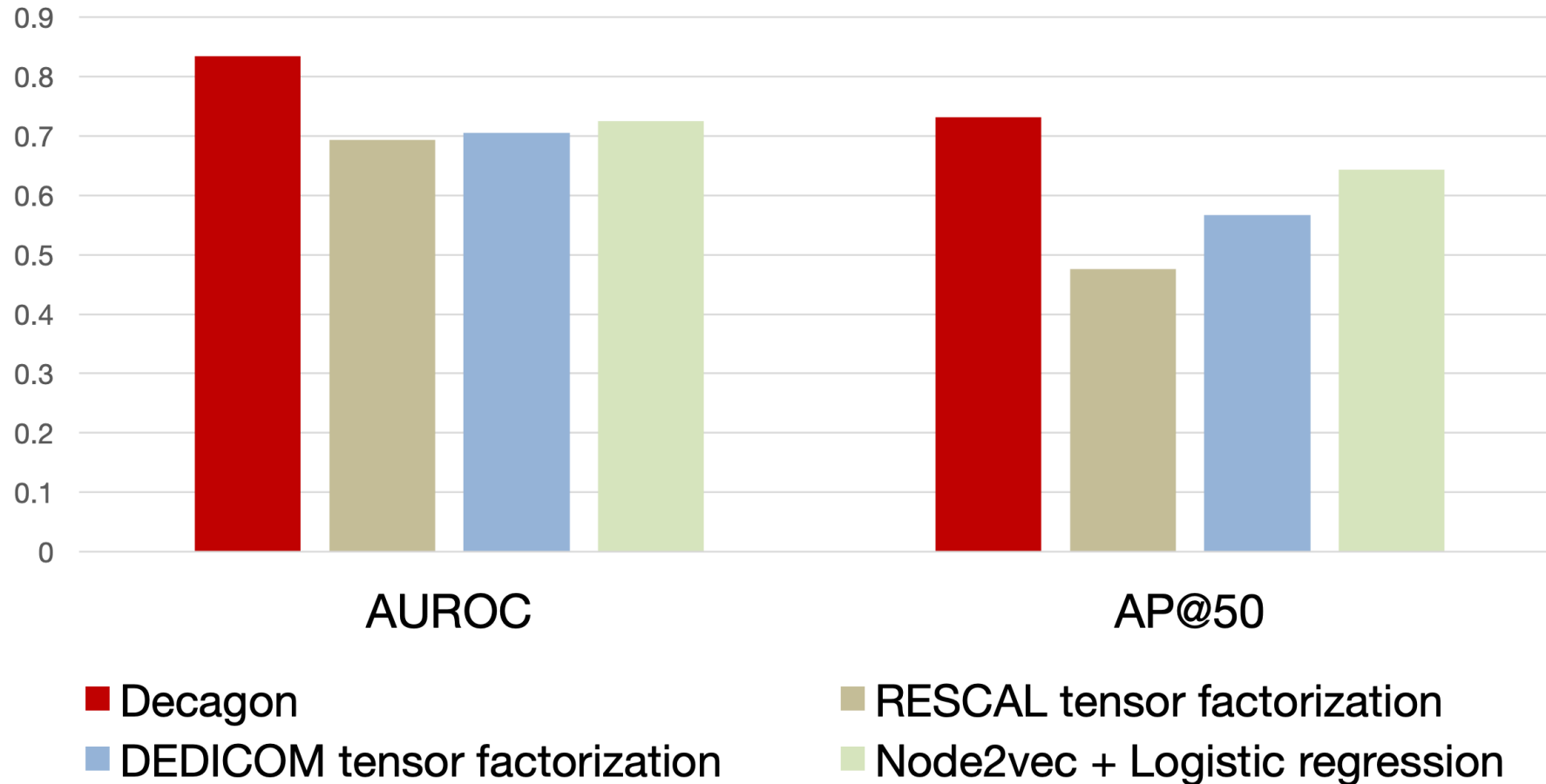
$$p(C, r_4, S)$$

...

$$p(C, r_n, S)$$

$r_1, r_2, r_3, \dots, r_n$  Polypharmacy side effects

# Decagon: Results (Side Effect Prediction)



36% average in AP@50 improvement over baselines

# Decagon: Results (Qualitative analysis)

**Table 4.** New polypharmacy side effect predictions given by (drug *i*, side effect type *r*, drug *j*) triples that were assigned the highest probability scores by *Decagon*

k	Polypharmacy effect r	Drug i	Drug j	Evidence
1	Sarcoma	Pyrimethamine	Aliskiren	Stage <i>et al.</i> (2015)
4	Breast disorder	Tolcapone	Pyrimethamine	Bicker <i>et al.</i> (2017)
6	Renal tubular acidosis	Omeprazole	Amoxicillin	Russo <i>et al.</i> (2016)
8	Muscle inflammation	Atorvastatin	Amlodipine	Banakh <i>et al.</i> (2017)
9	Breast inflammation	Aliskiren	Tioconazole	Parving <i>et al.</i> (2012)

## Case Report

### Severe Rhabdomyolysis due to Presumed Drug Interactions between Atorvastatin with Amlodipine and Ticagrelor

Iouri Banakh,<sup>1</sup> Kavi Haji,<sup>2,3</sup> Ross Kung,<sup>2</sup> Sachin Gupta,<sup>2,3</sup> and Ravindranath Tiruvoipati<sup>2,3</sup>

<sup>1</sup>Department of Pharmacy, Frankston Hospital, Peninsula Health, Frankston, VIC 3199, Australia  
<sup>2</sup>Department of Intensive Care Medicine, Frankston Hospital, Peninsula Health, Frankston, VIC 3199, Australia  
<sup>3</sup>School of Public Health, Faculty of Medicine, Nursing and Health Sciences, Monash University, Clayton, VIC 3800, Australia

Correspondence should be addressed to Iouri Banakh; ibanakh@phcn.vic.gov.au

Received 26 February 2017; Revised 19 April 2017; Accepted 4 May 2017; Published 25 May 2017

## Rhabdomyolysis

Article [Talk](#)

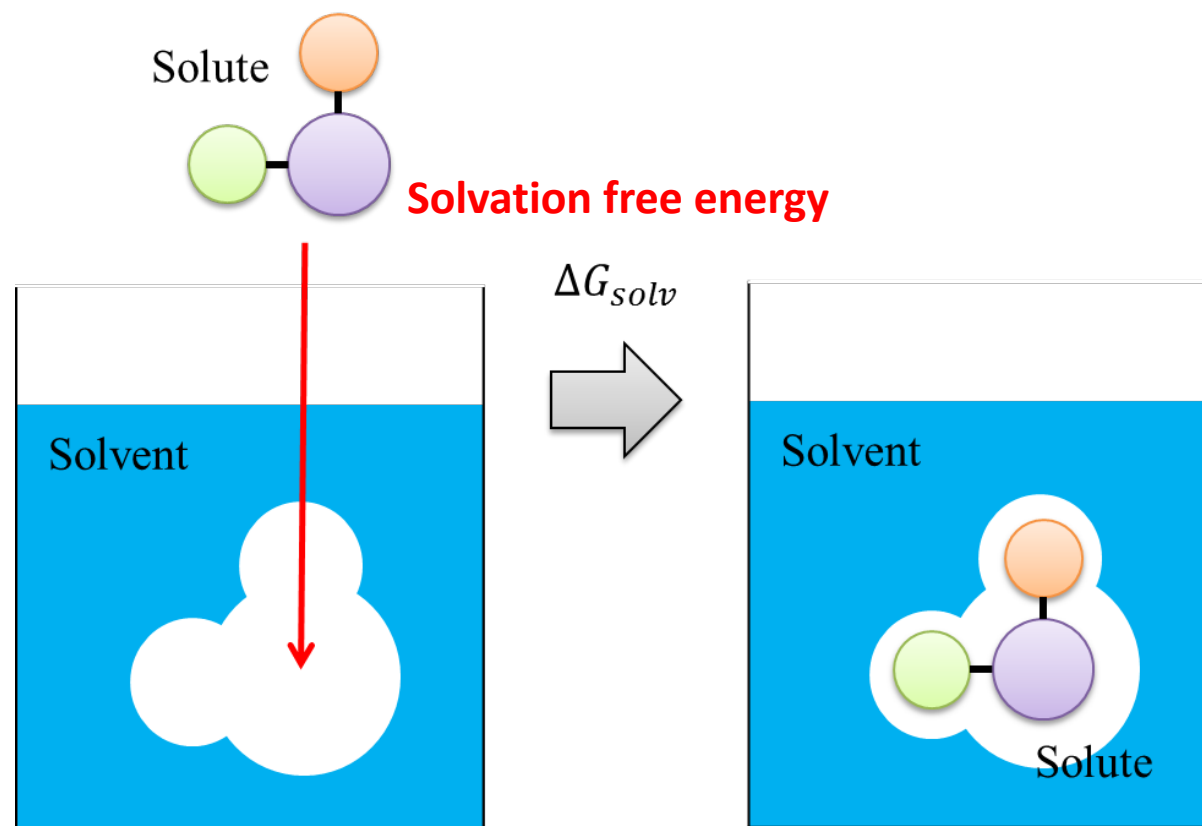
From Wikipedia, the free encyclopedia

Rhabdomyolysis (also called rhabdo) is a condition in which damaged skeletal muscle breaks down rapidly.<sup>[6][4][5]</sup> Symptoms may include muscle pains, weakness, vomiting, and confusion.<sup>[3][4]</sup> There may be tea-colored urine or an irregular heartbeat.<sup>[3][5]</sup> Some of the muscle breakdown products, such as the protein myoglobin, are harmful to the kidneys and can cause acute kidney injury.<sup>[7][3]</sup>

# Predicting Solvation Free Energy (용매화 자유 에너지)

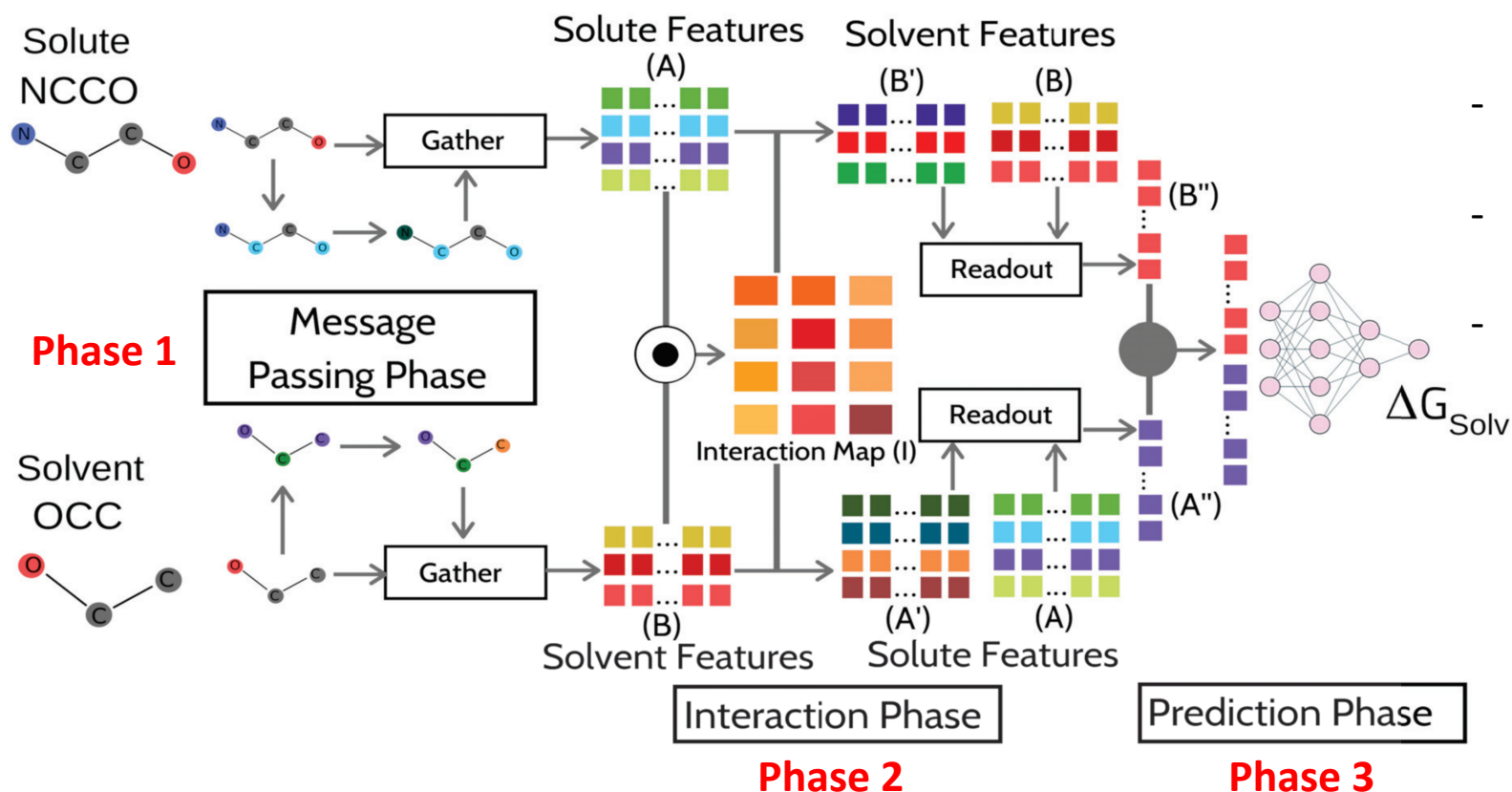
## ■ Solvation free energy

- Change in free energy for a molecule to be transferred from gas phase to a given solvent
- Quantifies **solubility** of drug molecules
  - A large negative value → high solubility
  - A lower magnitudes/positive value → poor solubility



# CIGIN: Overview

- Task: Predicting solvation free energy
- Previous studies considered only the solute for solvation free energy prediction and ignored the nature of the solvent



- **Phase 1:** Compute inter-atomic interaction within both solute and solvent
- **Phase 2:** Calculate a solute-solvent interaction map
- **Phase 3:** Predict the solvation free energies

# CIGIN: Model Architecture

## Phase 1: Message Passing Phase

$$m_v^{t+1} = \sum_{w \in N(v)} M_t(h_v^t, h_w^t, e_{vw}), \quad h_v^{t+1} = U_t(h_v^t, m_v^{t+1})$$

Message function:  $M_t$   
 Node feature:  $h_v^t$   
 Edge feature:  $e_{vw}$   
 Node update function:  $U_t$   
 Neighbors of  $v$ :  $N(v)$   
 Final feature of  $v$ :  $F_v$   
 $F_v = g(x_v, h_v^t), \forall v \in V$   
 set2set layer:  $g$

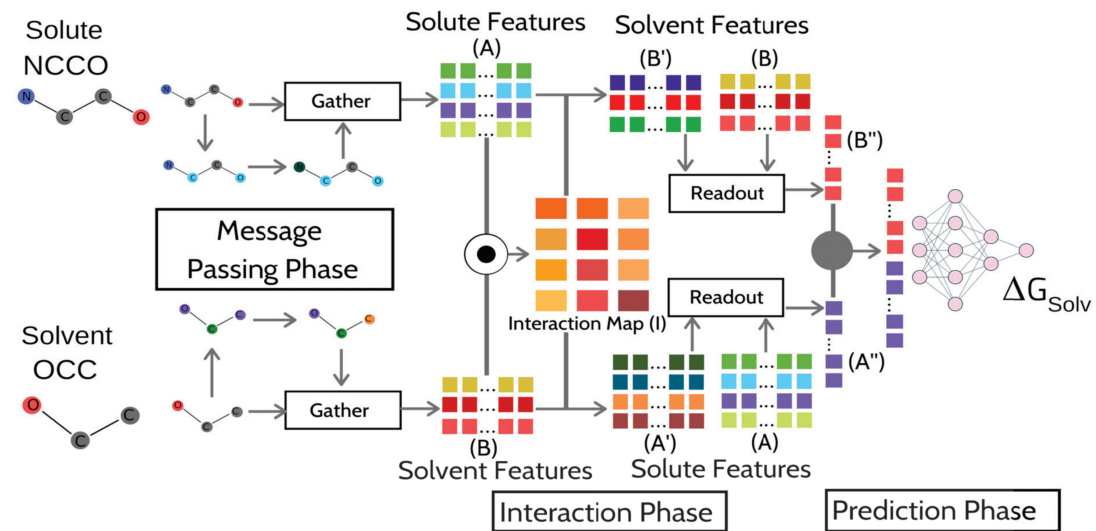
## Phase 2: Interaction Phase

$$f(A_n, B_m) = \tanh(A_n \cdot B_m)$$

$$I_{nm} = f(A_n, B_m), \forall n = 1, 2, 3..J, \forall m = 1, 2, 3, ..K$$

Atom  $n$  of solute:  $A_n$   
 Atom  $m$  of solvent:  $B_m$

$$A' = IB, \quad B' = I^T A$$



## Phase 3: Prediction Phase

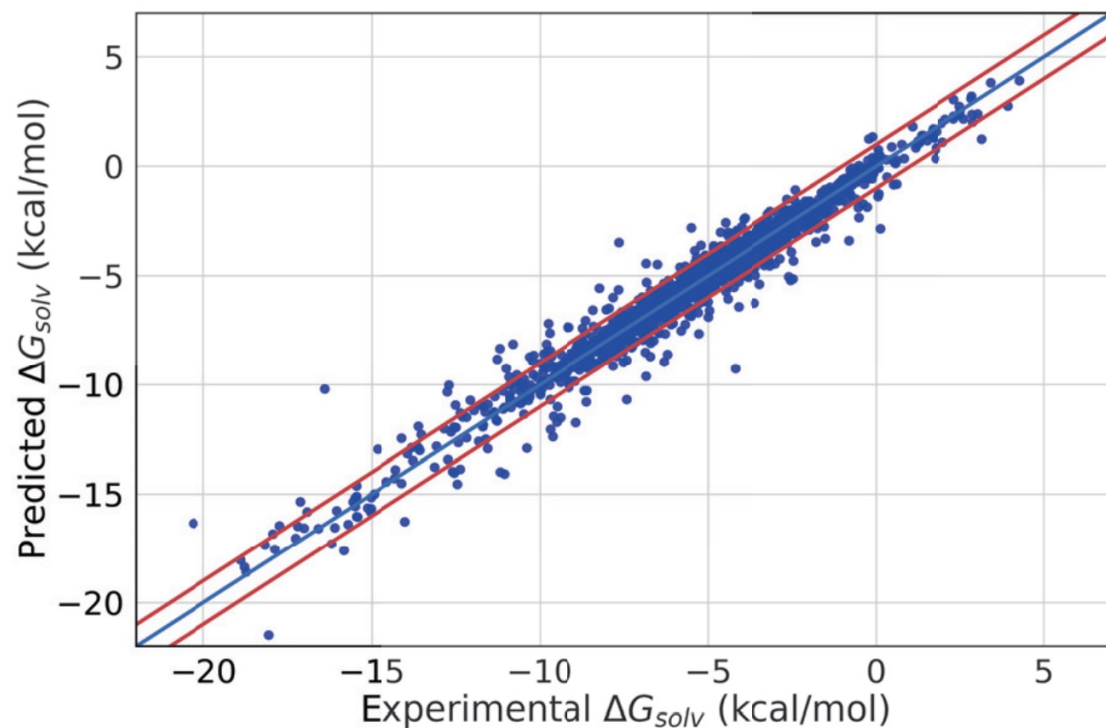
$$A'' = R_{solute}(A, A'), \quad B'' = R_{solvent}(B, B')$$

set2set layer:  $R_{solute}$   
 set2set layer:  $R_{solvent}$

$$\Delta G_{Solv} = f_{final}[\text{Concat}(A'', B'')]$$



# CIGIN: Results



Model	RMSE (kcal/mol)
Baseline model	$0.65 \pm 0.13$
CIGIN (sum pooling)	$0.61 \pm 0.12$
CIGIN (set2set)	<b><math>0.57 \pm 0.10</math></b>

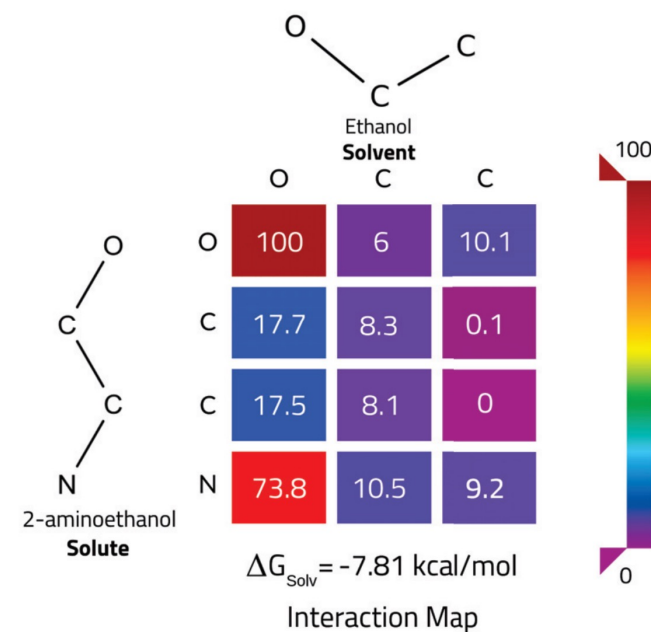


Figure 3: Heat map of the normalized (min-max) interaction map for 2-aminoethanol (solute) and ethanol(solvent) along with the predicted solvation free energy.



# MIRACLE

- Task: Predicting drug-drug interaction
- Key idea: Construct a graph-of-graphs

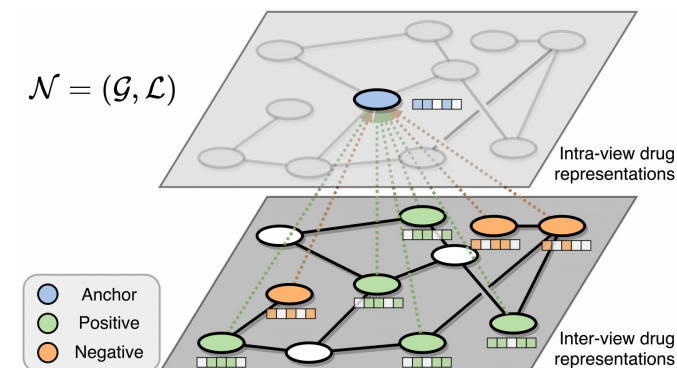
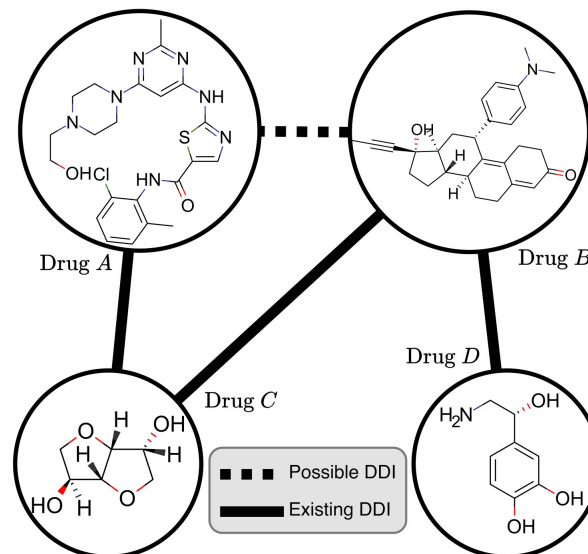
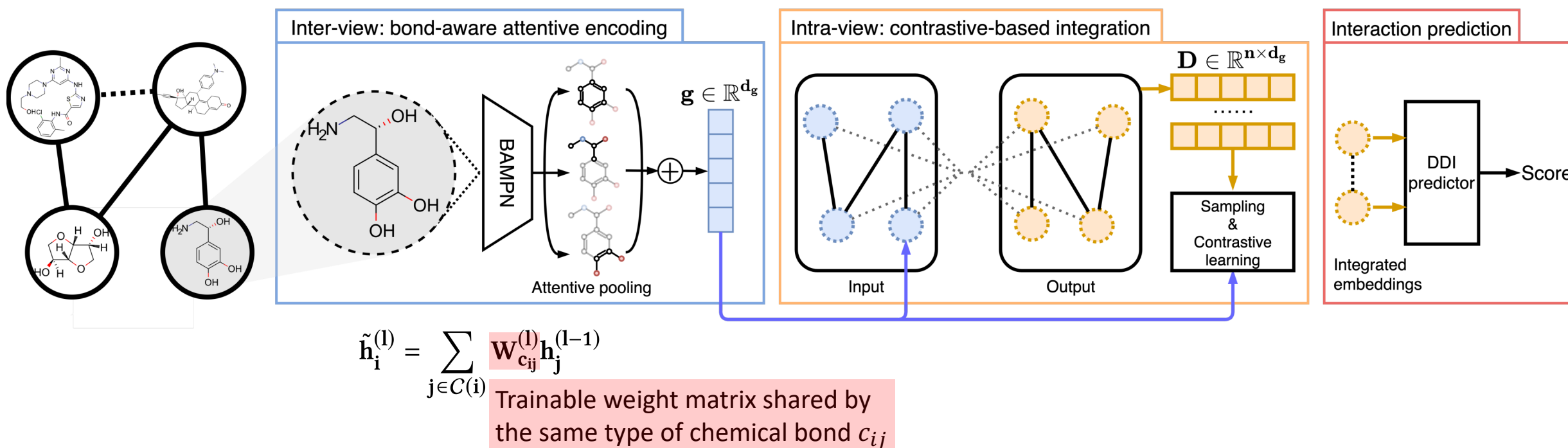


Figure 3: The proposed graph contrastive learning framework.



# Outline

- 그래프 신경망 개요 (20 mins)
  - 그래프 신경망 전반적인 소개
  - 그래프 종류에 따른 다양한 그래프 신경망 소개
- How to address Out-of-distribution problem (세부 기술 및 Q&A) (90~120 mins)
  - 소재 물성 예측 연구
    - 소재 물성 예측 연구 최신 동향 소개
    - Transformer 기반 모델 소개 → Prompt-based method
    - Extrapolation을 위한 모델 소개 → Nonlinearity encoding-based method
  - 물질 간 화학 반응 예측 연구
    - 물질 간 화학 반응 예측 연구 동향 소개
    - 정보 이론(Information bottleneck) 기반 모델 소개 → **Information bottleneck-based method**
    - 인과추론(Causal inference) 기반 모델 소개 → Causal inference-based method

# Papers

## ■ General

- Modeling polypharmacy side effects with graph convolutional networks. Bioinformatics 2018
- Chemically interpretable graph interaction network for prediction of pharmacokinetic properties of drug-like molecules. AAAI 2020
- Multi-view graph contrastive representation learning for drug-drug interaction prediction. WWW 2021

## ■ Information bottleneck-based

- Interpretable and generalizable graph learning via stochastic attention mechanism. ICML 2022
- Improving subgraph recognition with variational graph information bottleneck. CVPR 2022
- **Conditional Graph Information Bottleneck for Molecular Relational Learning. ICML 2023**

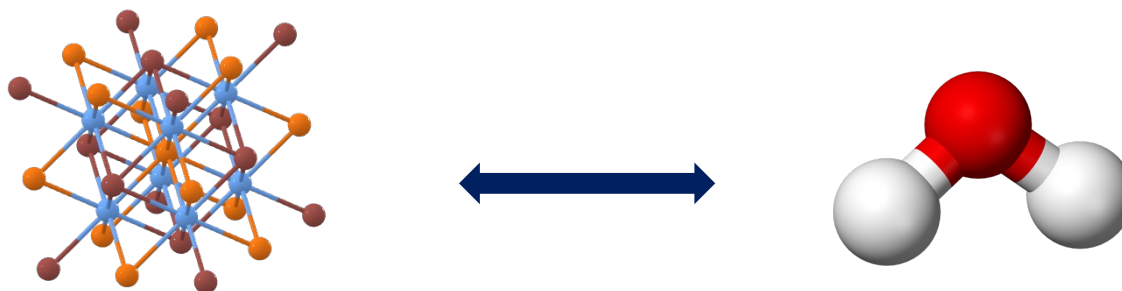
## ■ Causal inference-based

- Discovering invariant rationales for graph neural networks. ICLR 2022
- Debiasing Graph Neural Networks via Learning Disentangled Causal Substructure. NeurIPS 2022
- Causal attention for interpretable and generalizable graph classification. KDD 2022
- **Shift-robust molecular relational learning with causal substructure. KDD 2023**

# Introduction: Relational Learning

## ▪ Molecular Relational Learning

- Learn the interaction behavior between a pair of molecules



### • Examples

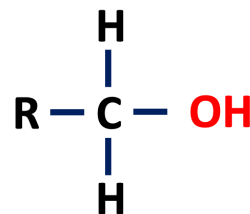
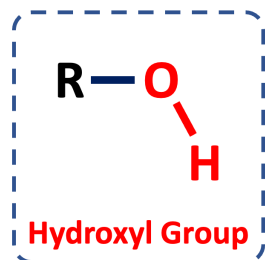
- Predicting **optical properties** when a chromophore (Chromophore) and solvent (Solvent) react
- Predicting **solubility** when a solute and solvent react
- Predicting **side effects** when taking two types of drugs simultaneously (Polypharmacy effect)

# Introduction: Functional Group

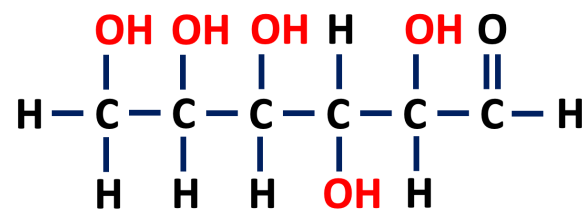
## ■ Functional Groups

- Specific atomic groups or structures that play an important role in determining the chemical reactivity of organic compounds
- Compounds with the same functional group generally have similar properties and undergo similar chemical reactions
- Examples
  - The hydroxyl group structure has the characteristic of **increasing the polarity** of the molecule  
→ Molecules containing hydroxyl structures, such as alcohol and glucose, commonly have a high solubility in water

### Functional Group



Alcohol

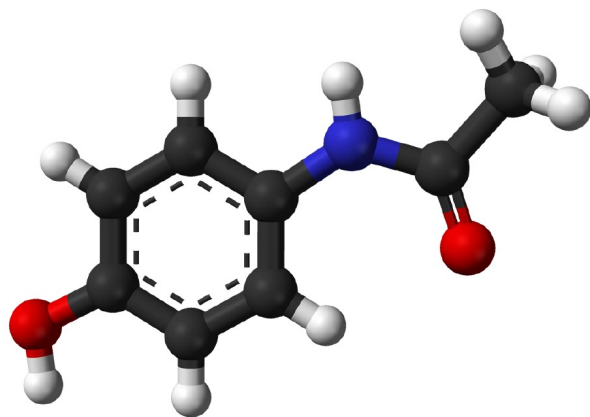


Glucose

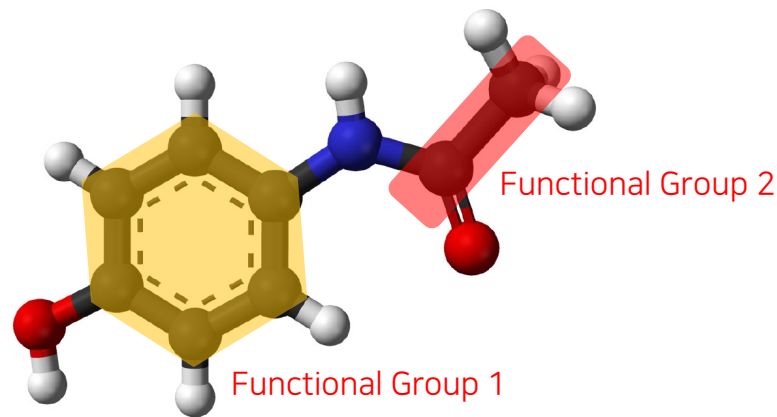
Hence, it is important to consider functional group for molecular relation learning

# Introduction: Representing Molecules as a Graph

- Molecule → Can be represented as a **graph**
- Functional Group → Can be represented as a **subgraph**



Molecule  
(=Graph)



Functional Group  
(=Subgraph)

Recently, information theory-based approaches have been proposed to detect important subgraph

# Information Bottleneck

- How can we find an important subgraph based on machine learning model?
- **Solution: Information Bottleneck Theory**
  - A theoretical approach to the trade-off between information **compression** and **preservation**
  - Given random variables  $X$  and  $Y$ , the Information Bottleneck principle aims to compress  $X$  to a bottleneck random variable  $T$ , while keeping the information relevant for predicting  $Y$
  - That is, the goal is to obtain  $T$  that compresses as much of information contained in  $X$  while still being able to predict  $Y$ 
    - Widely used to learn noisy robust representation

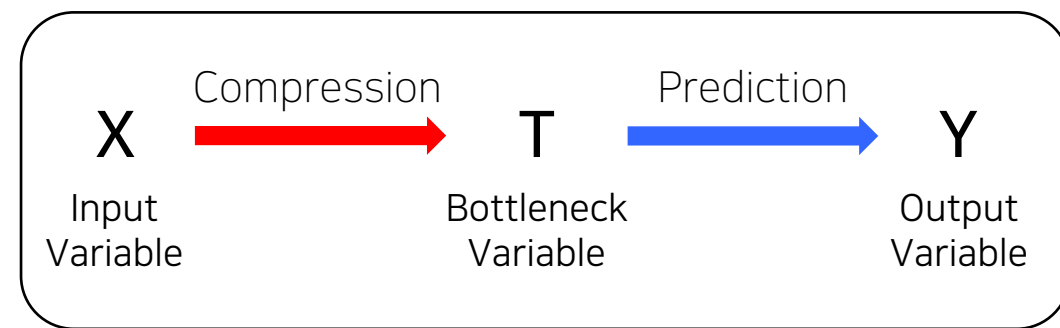
$$\min_T -I(Y; T) + \beta I(X; T)$$

Minimize MI between  $X$  and  $T$   
→  $T$  should contain minimal information about  $X$   
→ **Compression**

Maximize MI between  $T$  and  $Y$   
→  $T$  should contain as much information about  $Y$  as possible  
→ **Prediction**

## Information Bottleneck Objective

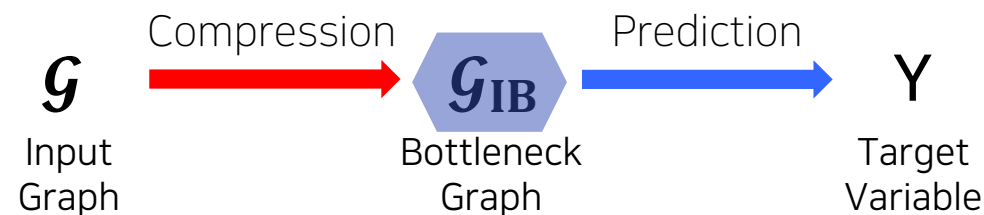
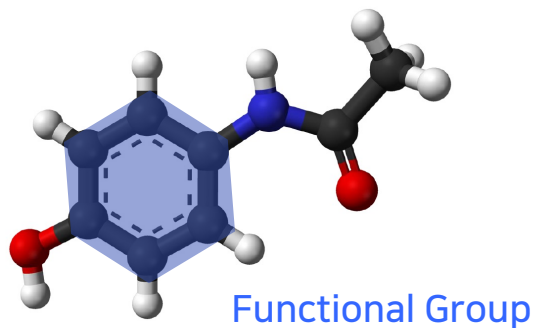
( $I(X, Y)$ ): Mutual information between  $X$  and  $Y$ )



# Graph Information Bottleneck: Overview

- How can we apply information bottleneck theory to graphs?
- Information Bottleneck Graph (IB-Graph)
  - To detect a subgraph that maximally preserves the property of the original graph
  - Subgraph becomes the bottleneck variable T
    - Problem formulation: Find Subgraph  $G_{IB}$  that is important for predicting Target Y

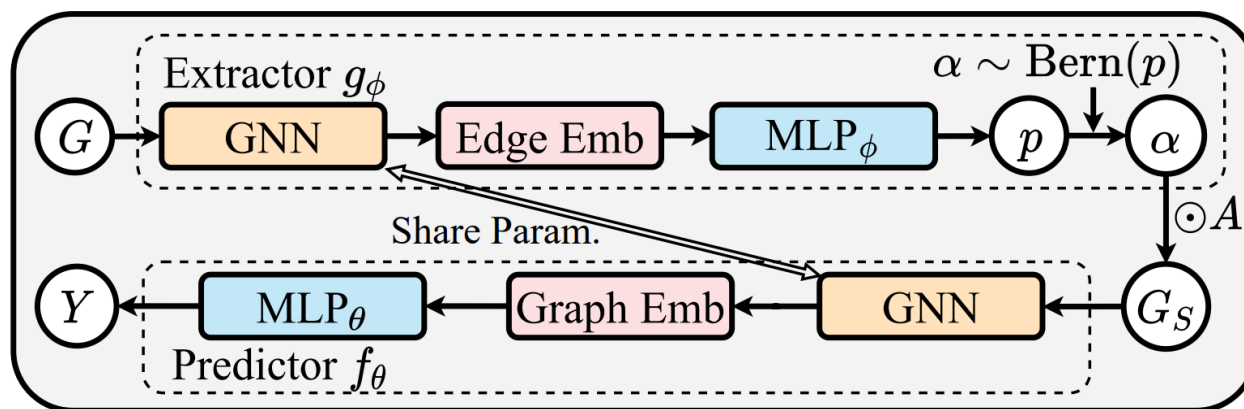
$$\mathcal{G}_{IB} = \arg \min_{\mathcal{G}_{IB}} -I(Y; \mathcal{G}_{IB}) + \beta I(\mathcal{G}; \mathcal{G}_{IB})$$





# Graph Information Bottleneck: Existing studies (1/2)

- Extract a subgraph in terms of edges
  - Model an edge based on Bernoulli distribution to perform graph compression

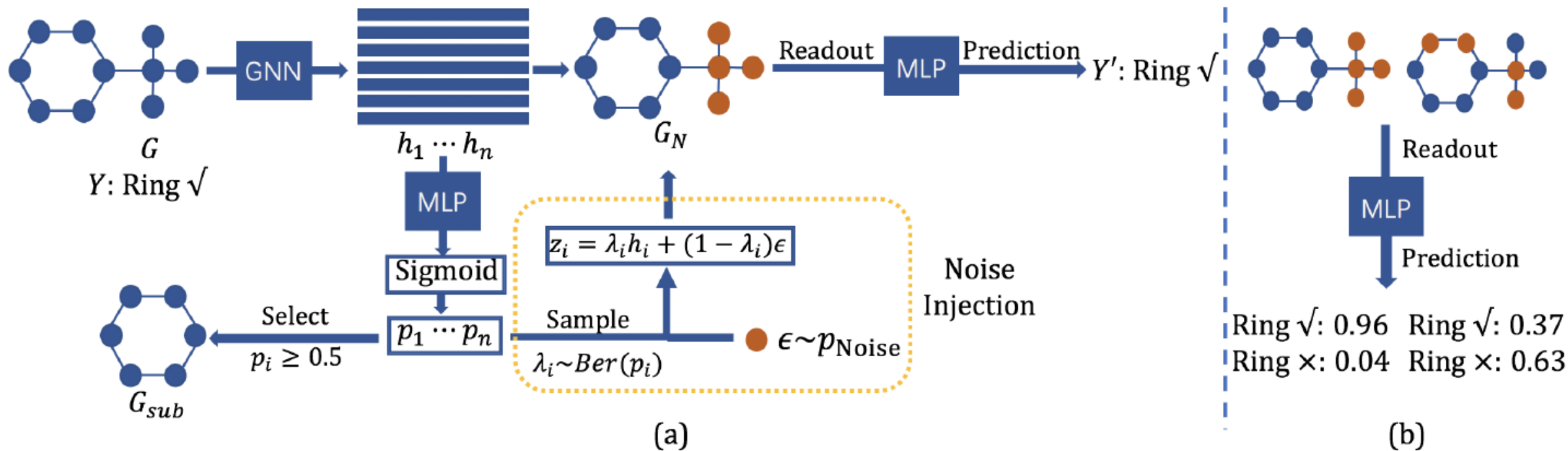


$$\min_{\phi} -I(G_S; Y) + \beta I(G_S; G), \text{ s.t. } G_S \sim g_\phi(G)$$

$$\min_{\theta, \phi} -\mathbb{E} [\log \mathbb{P}_\theta(Y|G_S)] + \beta \mathbb{E} [\text{KL}(\mathbb{P}_\phi(G_S|G) || \mathbb{Q}(G_S))], \text{ s.t. } G_S \sim \mathbb{P}_\phi(G_S|G)$$

## Graph Information Bottleneck: Existing studies (2/2)

- Extract a subgraph in terms of nodes
  - Inject noise into node embeddings to perform graph compression



However, the existing studies address single-input tasks, hence cannot be applied to relational learning tasks with two input graphs

# Conditional Graph Information Bottleneck for Molecular Relational Learning

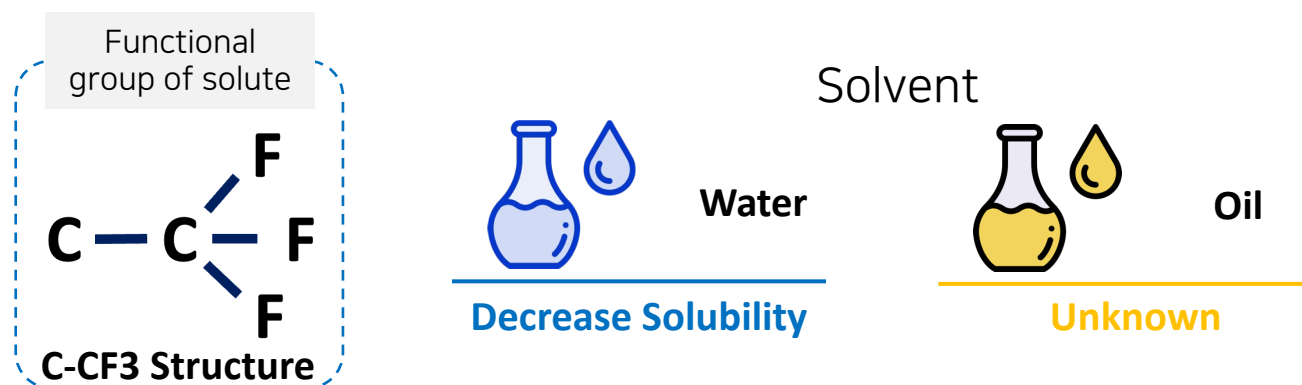
Namkyeong Lee, Dongmin Hyun, Gyoung S. Na, Sungwon Kim, Junseok Lee, Chanyoung Park

ICML 2023 - International Conference on Machine Learning

# Recall: Functional Group

## ▪ Functional Groups

- Specific atomic groups or structures that play an important role in determining the chemical reactivity of organic compounds
- Compounds with the same functional group generally have similar properties and undergo similar chemical reactions
- On the other hand, the role of functional group varies depending on which solvent the solute (Chromophore) reacts with
  - Examples: C-CF<sub>3</sub> structure decreases the solubility of a molecule in water
    - However, it is unknown how C-CF<sub>3</sub> structure affects the solubility of a molecule in oil
    - Hence, it is important to consider the paired solvent when detecting important substructure from solute



Existing approaches for information bottleneck cannot capture such a prior knowledge

# Proposed Method: Conditional Graph Information Bottleneck

- Conditional Information Bottleneck Graph (CIB-Graph)

- Consider Graph 2 (Solvent) when detecting the important subgraph from Graph 1 (Chromophore)

$$\mathcal{G}_{\text{IB}} = \arg \min_{\mathcal{G}_{\text{IB}}} -I(Y; \mathcal{G}_{\text{IB}}) + \beta I(\mathcal{G}; \mathcal{G}_{\text{IB}}) \quad \text{Graph Information Bottleneck}$$



$$\mathcal{G}_{\text{CIB}}^1 = \arg \min_{\mathcal{G}_{\text{CIB}}^1} -I(Y; \mathcal{G}_{\text{CIB}}^1 | \mathcal{G}^2) + \beta I(\mathcal{G}_{\text{CIB}}^1; \mathcal{G}^1 | \mathcal{G}^2) \quad \text{Conditional Graph Information Bottleneck (CGIB)}$$

# Proof of Lemma

$$\mathcal{G}_{\text{CIB}}^1 = \arg \min_{\mathcal{G}_{\text{CIB}}^1} -I(\mathbf{Y}; \mathcal{G}_{\text{CIB}}^1 | \mathcal{G}^2) + \beta I(\mathcal{G}_{\text{CIB}}^1; \mathcal{G}^1 | \mathcal{G}^2)$$

Conditional Graph  
Information Bottleneck  
(CGIB)

- By proving the following lemma, we show that minimizing the CGIB objective is equivalent to detecting **task relevant subgraph**

**Lemma 4.3.** (Nuisance Invariance) Given a pair of graphs  $(\mathcal{G}^1, \mathcal{G}^2)$  and its label information  $\mathbf{Y}$ , let  $\mathcal{G}_n^1$  be a task irrelevant noise in the input graph  $\mathcal{G}^1$ . Then, the following inequality holds:

$\mathcal{G}_n^1$ : Task irrelevant noise

$$I(\mathcal{G}_{\text{CIB}}^1; \mathcal{G}_n^1 | \mathcal{G}^2) \leq -I(\mathbf{Y}; \mathcal{G}_{\text{CIB}}^1 | \mathcal{G}^2) + I(\mathcal{G}^1; \mathcal{G}_{\text{CIB}}^1 | \mathcal{G}^2) \quad (6)$$

## Proof of Lemma 4.3

Assuming that  $\mathcal{G}^1$ ,  $\mathcal{G}_{\text{CIB}}^1$ ,  $\mathcal{G}_n^1$ ,  $\mathcal{G}^2$ , and  $\mathbf{Y}$  satisfy the Markov condition  $(\mathbf{Y}, \mathcal{G}_n^1, \mathcal{G}^2) \rightarrow \mathcal{G}^1 \rightarrow \mathcal{G}_{\text{CIB}}^1$ , we have the following inequality due to data processing inequality:

$$\begin{aligned} I(\mathcal{G}^1; \mathcal{G}_{\text{CIB}}^1 | \mathcal{G}^2) &= I(\mathcal{G}_{\text{CIB}}^1; \mathcal{G}^1, \mathcal{G}^2) - I(\mathcal{G}_{\text{CIB}}^1; \mathcal{G}^2) \\ &\geq I(\mathcal{G}_{\text{CIB}}^1; \mathbf{Y}, \mathcal{G}_n^1, \mathcal{G}^2) - I(\mathcal{G}_{\text{CIB}}^1; \mathcal{G}^2) \\ &= I(\mathcal{G}_{\text{CIB}}^1; \mathcal{G}_n^1, \mathcal{G}^2) + I(\mathcal{G}_{\text{CIB}}^1; \mathbf{Y} | \mathcal{G}_n^1, \mathcal{G}^2) - I(\mathcal{G}_{\text{CIB}}^1; \mathcal{G}^2) \\ &= I(\mathcal{G}_{\text{CIB}}^1; \mathcal{G}_n^1 | \mathcal{G}^2) + I(\mathcal{G}_{\text{CIB}}^1; \mathbf{Y} | \mathcal{G}_n^1, \mathcal{G}^2) \quad (1) \end{aligned}$$

Suppose that  $\mathcal{G}_n^1$  and  $\mathbf{Y}$ ,  $\mathcal{G}_n^1$  and  $\mathcal{G}^2$ , and joint random variable  $(\mathcal{G}_n^1, \mathcal{G}^2)$  and  $\mathbf{Y}$  are independent respectively. Then, for  $I(\mathcal{G}_{\text{CIB}}^1; \mathbf{Y} | \mathcal{G}_n^1, \mathcal{G}^2)$  we have:

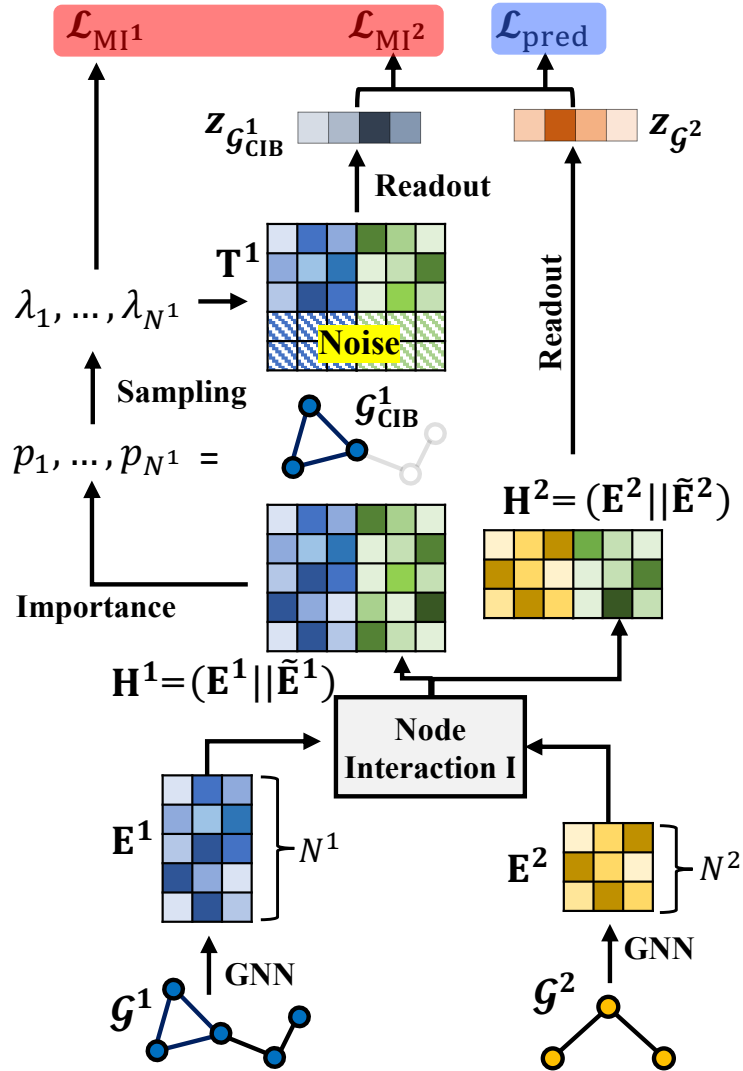
$$\begin{aligned} I(\mathcal{G}_{\text{CIB}}^1; \mathbf{Y} | \mathcal{G}_n^1, \mathcal{G}^2) &= H(\mathbf{Y} | \mathcal{G}_n^1, \mathcal{G}^2) - H(\mathbf{Y} | \mathcal{G}_n^1, \mathcal{G}_{\text{CIB}}^1, \mathcal{G}^2) \\ &\geq H(\mathbf{Y} | \mathcal{G}^2) - H(\mathbf{Y} | \mathcal{G}_{\text{CIB}}^1, \mathcal{G}^2) \\ &= I(\mathbf{Y}; \mathcal{G}_{\text{CIB}}^1 | \mathcal{G}^2) \quad (2) \end{aligned}$$

By plugging Equation (2) into Equation (1), we have:

$$\begin{aligned} I(\mathcal{G}^1; \mathcal{G}_{\text{CIB}}^1 | \mathcal{G}^2) &\geq I(\mathcal{G}_{\text{CIB}}^1; \mathcal{G}_n^1 | \mathcal{G}^2) + I(\mathbf{Y}; \mathcal{G}_{\text{CIB}}^1 | \mathcal{G}^2) \\ \therefore I(\mathcal{G}_{\text{CIB}}^1; \mathcal{G}_n^1 | \mathcal{G}^2) &\leq -I(\mathbf{Y}; \mathcal{G}_{\text{CIB}}^1 | \mathcal{G}^2) + I(\mathcal{G}^1; \mathcal{G}_{\text{CIB}}^1 | \mathcal{G}^2) \end{aligned}$$

By minimizing the CGIB objective function, the model learns  $\mathcal{G}_{\text{CIB}}^1$  with the smallest mutual information with task-irrelevant noise  $\mathcal{G}_n^1$ .

# Proposed Method: Conditional Graph Information Bottleneck



$$\min -I(Y; \mathcal{G}_{CIB}^1 | \mathcal{G}^2) + \beta I(\mathcal{G}^1; \mathcal{G}_{CIB}^1 | \mathcal{G}^2)$$

Overall procedure:

Decompose the conditional MI based on the chain rule of MI, and then *derive the upper bound* of the decomposed terms

Chain Rule of MI

$$\begin{aligned} I(X; Y, Z) &= I(X; Z) + I(X; Y|Z) \\ &= I(X; Y) + I(X; Z|Y) \end{aligned}$$

$$-I(Y; \mathcal{G}_{CIB}^1 | \mathcal{G}^2) = -I(Y; \mathcal{G}_{CIB}^1, \mathcal{G}^2) + I(Y; \mathcal{G}^2)$$

$$-I(Y; \mathcal{G}_{CIB}^1, \mathcal{G}^2) \leq \mathbb{E}_{\mathcal{G}_{CIB}^1, \mathcal{G}^2, Y} [-\log p_{\theta}(Y | \mathcal{G}_{CIB}^1, \mathcal{G}^2)]$$

Prediction Loss

$$I(\mathcal{G}^1; \mathcal{G}_{CIB}^1 | \mathcal{G}^2) = I(\mathcal{G}_{CIB}^1; \mathcal{G}^1, \mathcal{G}^2) - I(\mathcal{G}_{CIB}^1; \mathcal{G}^2)$$

$$I(\mathcal{G}_{CIB}^1; \mathcal{G}^1, \mathcal{G}^2) \leq \mathbb{E}_{\mathcal{G}^1, \mathcal{G}^2} \left[ -\frac{1}{2} \log A + \frac{1}{2N^1} A + \frac{1}{2N^1} B^2 \right]$$

$$:= \mathcal{L}_{MI^1}(\mathcal{G}_{CIB}^1, \mathcal{G}^1, \mathcal{G}^2)$$

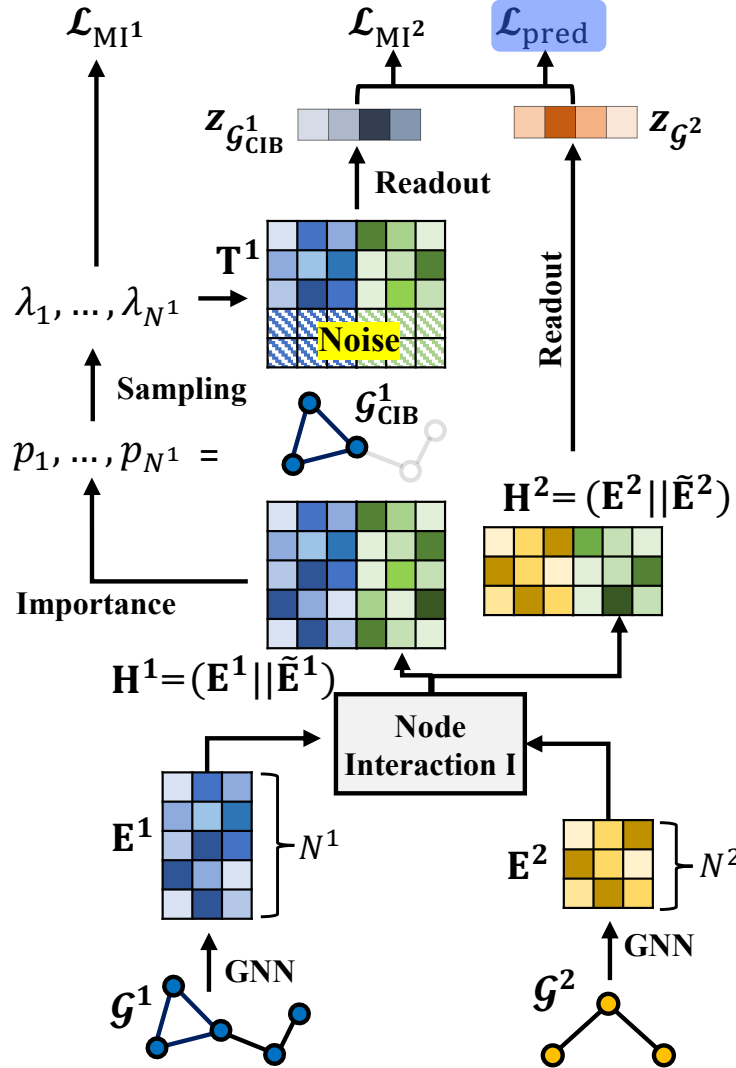
$$-I(\mathcal{G}_{CIB}^1; \mathcal{G}^2) \leq \mathbb{E}_{\mathcal{G}_{CIB}^1, \mathcal{G}^2} [-\log p_{\xi}(\mathcal{G}^2 | \mathcal{G}_{CIB}^1)]$$

$$:= \mathcal{L}_{MI^2}(\mathcal{G}_{CIB}^1, \mathcal{G}^2)$$

Compression Loss

$$\min -I(Y; \mathcal{G}_{\text{CIB}}^1 | \mathcal{G}^2) + \beta I(\mathcal{G}^1; \mathcal{G}_{\text{CIB}}^1 | \mathcal{G}^2)$$

# Proposed Method: Conditional Graph Information Bottleneck



## - Step 1: Optimizing the prediction loss

$$-I(Y; \mathcal{G}_{\text{CIB}}^1 | \mathcal{G}^2) = -I(Y; \mathcal{G}_{\text{CIB}}^1, \mathcal{G}^2) + I(Y; \mathcal{G}^2) \quad \because \text{Chain rule of mutual information}$$

Directly calculating the mutual Information is intractable;  
Instead, we *minimize the upper bound*

**Proposition.** (Upper bound of  $-I(Y; \mathcal{G}_{\text{CIB}}^1, \mathcal{G}^2)$ ) Given a pair of graph  $(\mathcal{G}^1, \mathcal{G}^2)$ , its label information  $Y$ , and the learned CIB-graph  $\mathcal{G}_{\text{CIB}}^1$ , we have:

$$-I(Y; \mathcal{G}_{\text{CIB}}^1, \mathcal{G}^2) \leq \mathbb{E}_{Y, \mathcal{G}_{\text{CIB}}^1, \mathcal{G}^2} [-\log p_{\theta}(Y | \mathcal{G}_{\text{CIB}}^1, \mathcal{G}^2)]$$

where  $p_{\theta}(Y | \mathcal{G}_{\text{CIB}}^1, \mathcal{G}^2)$  is variational approximation of  $p(Y | \mathcal{G}_{\text{CIB}}^1, \mathcal{G}^2)$ .

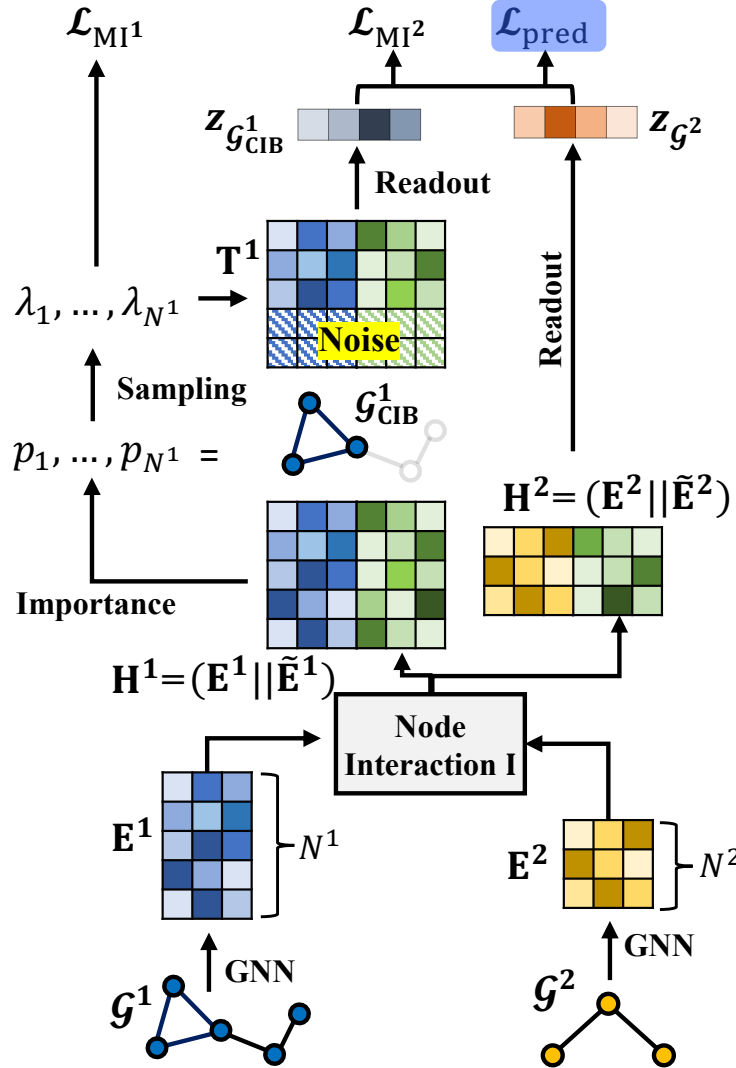
**Proof.** By the definition of mutual information and introducing variational approximation  $p_{\theta}(Y | \mathcal{G}_{\text{CIB}}^1, \mathcal{G}^2)$  of intractable distribution  $p(Y | \mathcal{G}_{\text{CIB}}^1, \mathcal{G}^2)$ , we have:

$$\begin{aligned} I(Y; \mathcal{G}_{\text{CIB}}^1, \mathcal{G}^2) &= \mathbb{E}_{Y, \mathcal{G}_{\text{CIB}}^1, \mathcal{G}^2} \left[ \log \frac{p(Y | \mathcal{G}_{\text{CIB}}^1, \mathcal{G}^2)}{p(Y)} \right] & I(X; Y) &= \mathbb{E}_{p(x, y)} \left[ \log \frac{p(x|y)}{p(x)} \right] = \mathbb{E}_{p(x, y)} \left[ \log \frac{p(y|x)}{p(y)} \right] \\ &= \mathbb{E}_{Y, \mathcal{G}_{\text{CIB}}^1, \mathcal{G}^2} \left[ \log \frac{p_{\theta}(Y | \mathcal{G}_{\text{CIB}}^1, \mathcal{G}^2)}{p(Y)} \right] + \mathbb{E}_{\mathcal{G}_{\text{CIB}}^1, \mathcal{G}^2} [p(Y | \mathcal{G}_{\text{CIB}}^1, \mathcal{G}^2) || p_{\theta}(Y | \mathcal{G}_{\text{CIB}}^1, \mathcal{G}^2)] \\ &\geq \mathbb{E}_{Y, \mathcal{G}_{\text{CIB}}^1, \mathcal{G}^2} \left[ \log \frac{p_{\theta}(Y | \mathcal{G}_{\text{CIB}}^1, \mathcal{G}^2)}{p(Y)} \right] & \because \text{Non-negativity of KL divergence} \\ &= \mathbb{E}_{Y, \mathcal{G}_{\text{CIB}}^1, \mathcal{G}^2} [\log p_{\theta}(Y | \mathcal{G}_{\text{CIB}}^1, \mathcal{G}^2)] + H(Y) \end{aligned}$$



$$\min -I(Y; \mathcal{G}_{\text{CIB}}^1 | \mathcal{G}^2) + \beta I(\mathcal{G}^1; \mathcal{G}_{\text{CIB}}^1 | \mathcal{G}^2)$$

# Proposed Method: Conditional Graph Information Bottleneck



## - Step 1: Optimizing the prediction loss

$$-I(Y; \mathcal{G}_{\text{CIB}}^1 | \mathcal{G}^2) = -I(Y; \mathcal{G}_{\text{CIB}}^1, \mathcal{G}^2) + I(Y; \mathcal{G}^2) \quad \because \text{Chain rule of mutual information}$$

Directly calculating the mutual Information is intractable;  
Instead, we *minimize the upper bound*

Proposition. (Upper bound of  $-I(Y; \mathcal{G}_{\text{CIB}}^1, \mathcal{G}^2)$ ) Given a pair of graph  $(\mathcal{G}^1, \mathcal{G}^2)$ , its label information  $Y$ , and the learned CIB-graph  $\mathcal{G}_{\text{CIB}}^1$ , we have:

$$-I(Y; \mathcal{G}_{\text{CIB}}^1, \mathcal{G}^2) \leq \mathbb{E}_{Y, \mathcal{G}_{\text{CIB}}^1, \mathcal{G}^2} [-\log p_{\theta}(Y | \mathcal{G}_{\text{CIB}}^1, \mathcal{G}^2)]$$

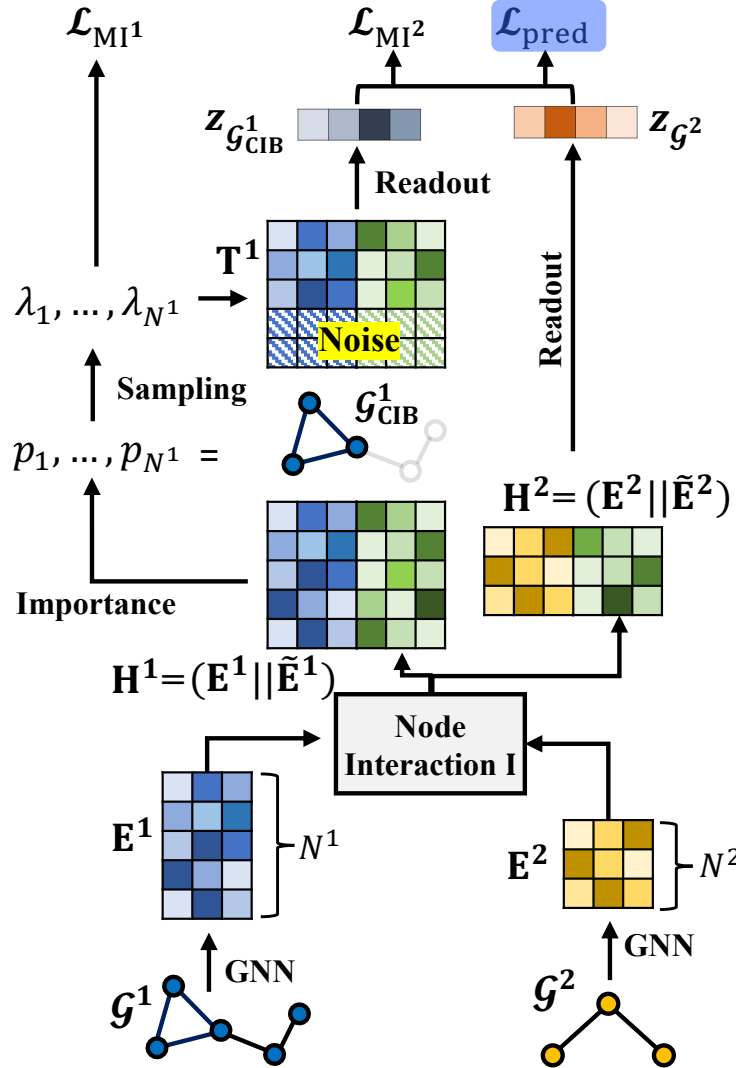
where  $p_{\theta}(Y | \mathcal{G}_{\text{CIB}}^1, \mathcal{G}^2)$  is variational approximation of  $p(Y | \mathcal{G}_{\text{CIB}}^1, \mathcal{G}^2)$ .

## Implementation.

- Consider  $p_{\theta}(Y | \mathcal{G}_{\text{CIB}}^1, \mathcal{G}^2)$  as a predictor parameterized by  $\theta$ , which outputs the model prediction  $\hat{Y}$  based on the input pair  $(\mathcal{G}_{\text{CIB}}^1, \mathcal{G}^2)$ .
- The upper bound is minimized by minimizing the prediction loss  $\mathcal{L}_{\text{pred}}(Y, \mathcal{G}_{\text{CIB}}^1, \mathcal{G}^2)$

$$\min -I(Y; \mathcal{G}_{\text{CIB}}^1 | \mathcal{G}^2) + \beta I(\mathcal{G}^1; \mathcal{G}_{\text{CIB}}^1 | \mathcal{G}^2)$$

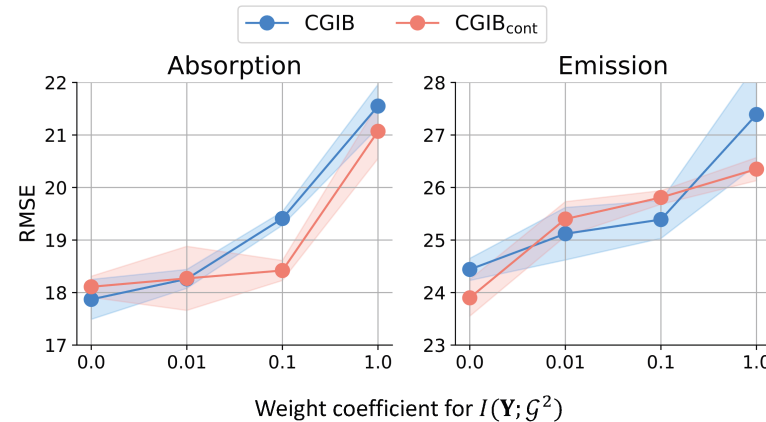
# Proposed Method: Conditional Graph Information Bottleneck



## - Step 1: Optimizing the prediction loss

$$-I(Y; \mathcal{G}_{\text{CIB}}^1 | \mathcal{G}^2) = -I(Y; \mathcal{G}_{\text{CIB}}^1, \mathcal{G}^2) + \underbrace{I(Y; \mathcal{G}^2)}_{\text{Chain rule of mutual information}} \quad \because \text{Chain rule of mutual information}$$

The 2<sup>nd</sup> term is empirically found to be not helpful



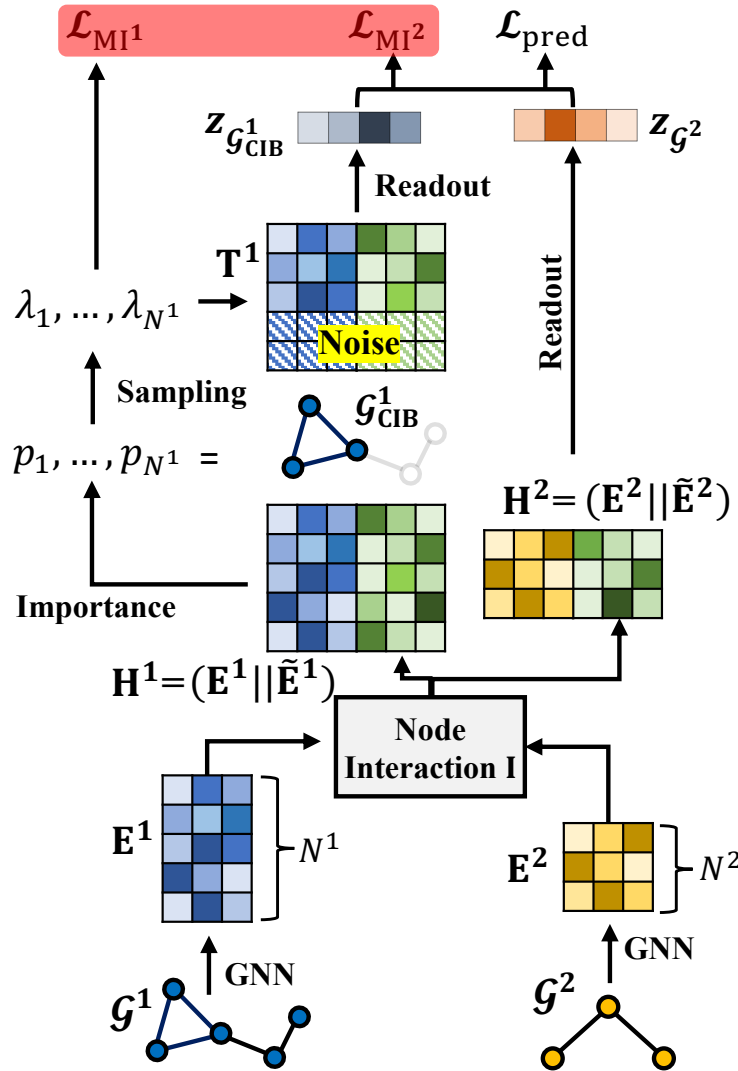
We treat  $r(Y)$  as fixed spherical Gaussian,  
 $I(Y; \mathcal{G}^2) \leq \mathbb{E}_{\mathcal{G}^2} [KL(p_{\xi}(Y | \mathcal{G}^2) || r(Y))]$   
 where  $r(Y) \sim N(Y | 0, 1)$

Increasing the contribution of this term deteriorates the model performance

Hence, we removed  $I(Y; \mathcal{G}^2)$  from the model

$$\min -I(Y; \mathcal{G}_{\text{CIB}}^1 | \mathcal{G}^2) + \beta I(\mathcal{G}^1; \mathcal{G}_{\text{CIB}}^1 | \mathcal{G}^2)$$

# Proposed Method: Conditional Graph Information Bottleneck



## - Step 2: Optimizing the compression loss

$$I(\mathcal{G}^1; \mathcal{G}_{\text{CIB}}^1 | \mathcal{G}^2) = \underbrace{I(\mathcal{G}_{\text{CIB}}^1; \mathcal{G}^1, \mathcal{G}^2)}_{L_{\text{MI}^1}} - \underbrace{I(\mathcal{G}_{\text{CIB}}^1; \mathcal{G}^2)}_{L_{\text{MI}^2}} \quad \because \text{Chain rule of mutual information}$$

### - $L_{\text{MI}^1}$ : Compression through Noise Injection

- \* Injecting noise into unimportant nodes
- Remaining nodes are important nodes

### - $L_{\text{MI}^2}$ : Solute Prediction

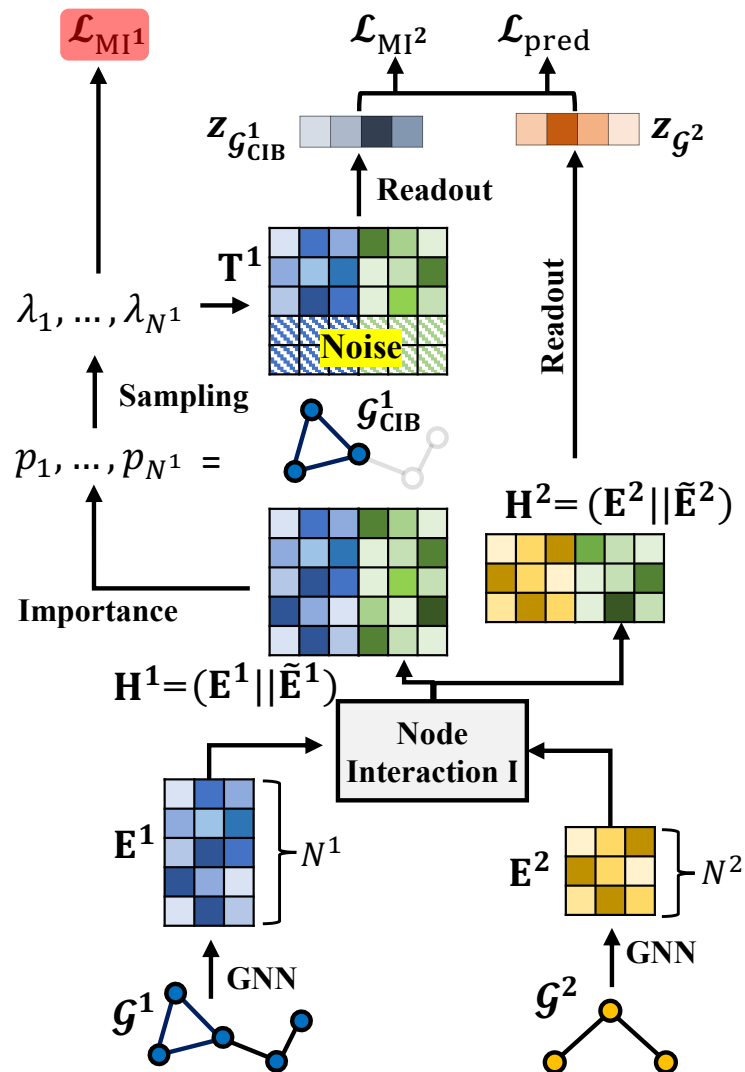
- \* Encourage  $\mathcal{G}_{\text{CIB}}^1$ , which is compressed conditioned on  $\mathcal{G}^2$ , to contain as much information about  $\mathcal{G}^2$  as possible

- \* This is the term that arises from the Conditional Mutual Information

→ Key to success of CGIB! Enables the conditional information compression of CGIB

$$\min -I(Y; \mathcal{G}_{\text{CIB}}^1 | \mathcal{G}^2) + \beta I(\mathcal{G}^1; \mathcal{G}_{\text{CIB}}^1 | \mathcal{G}^2)$$

# Proposed Method: Conditional Graph Information Bottleneck



## - Step 2: Optimizing the compression loss

$$I(\mathcal{G}^1; \mathcal{G}_{\text{CIB}}^1 | \mathcal{G}^2) = \underbrace{I(\mathcal{G}_{\text{CIB}}^1; \mathcal{G}^1, \mathcal{G}^2)} - I(\mathcal{G}_{\text{CIB}}^1; \mathcal{G}^2) \quad \because \text{Chain rule of mutual information}$$

### 1. Compression through Noise Injection

\* Injecting noise into unimportant nodes

- $H_i^1$  : Representation of node  $i$  of  $\mathcal{G}^1$  that contains information about both  $\mathcal{G}^1, \mathcal{G}^2$
- $p_i = \text{MLP}(H_i^1)$  : Important of node  $i$  of  $\mathcal{G}^1$
- $T_i^1 = \lambda_i H_i^1 + (1 - \lambda_i) \varepsilon$  (Replace  $H_i^1$  with noise  $\varepsilon$  depending on the important of node  $i$ ) where  $\lambda_i \sim \text{Bernoulli}(p_i)$  and  $\varepsilon \sim N(\mu_{H^1}, \sigma_{H^1}^2)$

Intuition) Unimportant nodes would not affect the model performance even if they are replaced with noise

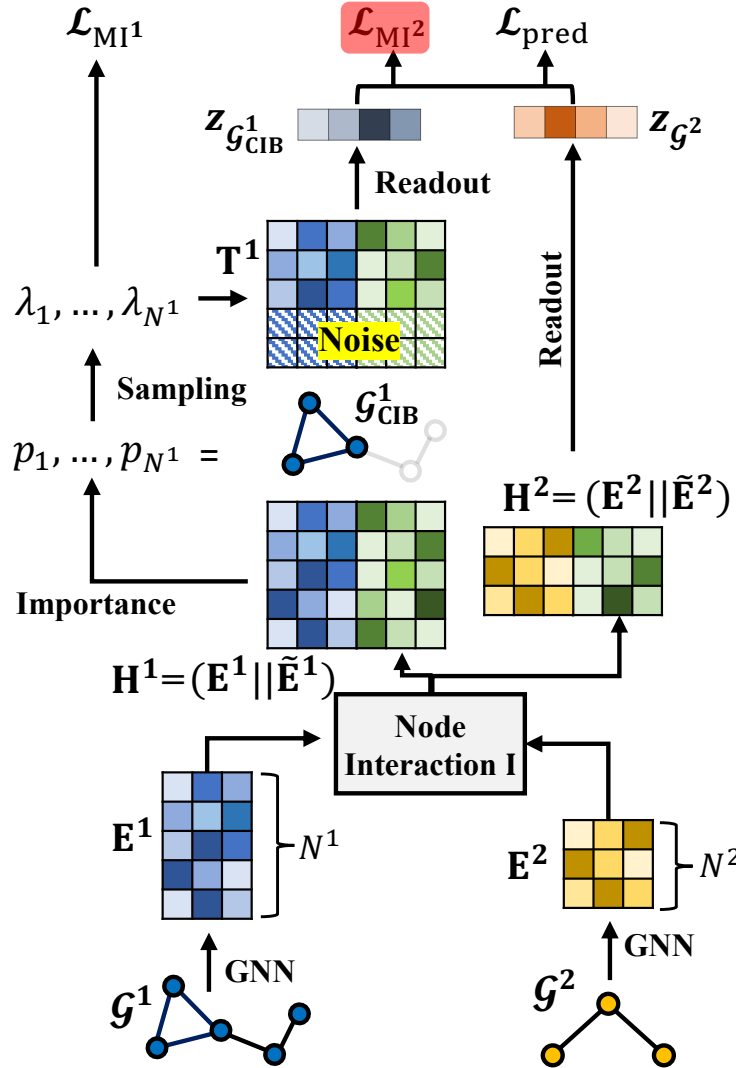
Upper bound of  $I(\mathcal{G}_{\text{CIB}}^1; \mathcal{G}^1, \mathcal{G}^2)$

$$I(\mathcal{G}_{\text{CIB}}^1; \mathcal{G}^1, \mathcal{G}^2) \leq \mathbb{E}_{\mathcal{G}^1, \mathcal{G}^2} \left[ -\frac{1}{2} \log A + \frac{1}{2N^1} A + \frac{1}{2N^1} B^2 \right] \quad \text{where } A = \sum_{j=1}^{N^1} (1 - \lambda_j)^2 \text{ and } B = \frac{\sum_{j=1}^{N^1} \lambda_j (H_j^1 - \mu_{H^1})^2}{\sigma_{H^1}^2}$$

$$:= \mathcal{L}_{\text{MI}^1}(\mathcal{G}_{\text{CIB}}^1, \mathcal{G}^1, \mathcal{G}^2)$$

$$\min -I(Y; \mathcal{G}_{\text{CIB}}^1 | \mathcal{G}^2) + \beta I(\mathcal{G}^1; \mathcal{G}_{\text{CIB}}^1 | \mathcal{G}^2)$$

# Proposed Method: Conditional Graph Information Bottleneck



## - Step 2: Optimizing the compression loss

$$I(\mathcal{G}^1; \mathcal{G}_{\text{CIB}}^1 | \mathcal{G}^2) = I(\mathcal{G}_{\text{CIB}}^1; \mathcal{G}^1, \mathcal{G}^2) - \underbrace{I(\mathcal{G}_{\text{CIB}}^1; \mathcal{G}^2)}_{\text{Chain rule of mutual information}} \quad \because \text{Chain rule of mutual information}$$

## 2. Solute Prediction

Encourage  $\mathcal{G}_{\text{CIB}}^1$ , which is compressed conditioned on  $\mathcal{G}^2$ , to contain as much information about  $\mathcal{G}^2$  as possible

Intuition) Make use of  $\mathcal{G}^2$  when detecting  $\mathcal{G}_{\text{CIB}}^1$

### 1) Variational IB-based approach

Derive upper bound similar to the prediction loss

$$-I(\mathcal{G}_{\text{CIB}}^1; \mathcal{G}^2) \leq \mathbb{E}_{\mathcal{G}_{\text{CIB}}^1, \mathcal{G}^2} [-\log p_{\xi}(\mathcal{G}^2 | \mathcal{G}_{\text{CIB}}^1)] := \mathcal{L}_{\text{MI}^2}(\mathcal{G}_{\text{CIB}}^1, \mathcal{G}^2)$$

### 2) Contrastive Learning-based approach

- Minimizing the contrastive loss = Maximizing the mutual information

- Hence, minimize  $-I(\mathcal{G}_{\text{CIB}}^1; \mathcal{G}^2)$  by minimizing the contrastive loss  $\rightarrow \text{CGIB}_{\text{cont}}$

$$\mathcal{L}_{\text{MI}^2} = -\frac{1}{K} \sum_{i=1}^K \log \frac{\exp(\text{sim}(\mathbf{z}_{\mathcal{G}_{\text{CIB},i}^1}, \mathbf{z}_{\mathcal{G}_i^2})/\tau)}{\sum_{j=1, j \neq i}^K \exp(\text{sim}(\mathbf{z}_{\mathcal{G}_{\text{CIB},i}^1}, \mathbf{z}_{\mathcal{G}_j^2})/\tau)}$$

# Experiments: Dataset

- 1) Chromophore dataset
  - Predicting Absorption max, Emission max, Lifetime
- 2) Solvation Free Energy dataset
  - MNSol / FreeSolv / CompSol / Abraham / CombiSolv
- 3) Drug-Drug Interaction dataset
  - ZhangDDI / ChChMiner

Dataset		$\mathcal{G}^1$	$\mathcal{G}^2$	# $\mathcal{G}^1$	# $\mathcal{G}^2$	# Pairs	Task
Chromophore <sup>1</sup>	Absorption	Chrom.	Solvent	6416	725	17276	reg.
	Emission	Chrom.	Solvent	6412	1021	18141	reg.
	Lifetime	Chrom.	Solvent	2755	247	6960	reg.
MNSol <sup>2</sup>		Solute	Solvent	372	86	2275	reg.
FreeSolv <sup>3</sup>		Solute	Solvent	560	1	560	reg.
CompSol <sup>4</sup>		Solute	Solvent	442	259	3548	reg.
Abraham <sup>5</sup>		Solute	Solvent	1038	122	6091	reg.
CombiSolv <sup>6</sup>		Solute	Solvent	1495	326	10145	reg.
ZhangDDI <sup>7</sup>		Drug	Drug	544	544	40255	cls.
ChChMiner <sup>8</sup>		Drug	Drug	949	949	21082	cls.

# Result: Main table

	Chromophore			MNSol	FreeSolv	CompSol	Abraham	CombiSolv
	Absorption	Emission	Lifetime					
GCN	25.75 (1.48)	31.87 (1.70)	0.866 (0.015)	0.675 (0.021)	1.192 (0.042)	0.389 (0.009)	0.738 (0.041)	0.672 (0.022)
GAT	26.19 (1.44)	30.90 (1.01)	0.859 (0.016)	0.731 (0.007)	1.280 (0.049)	0.387 (0.010)	0.798 (0.038)	0.662 (0.021)
MPNN	24.43 (1.55)	30.17 (0.99)	0.802 (0.024)	0.682 (0.017)	1.159 (0.032)	0.359 (0.011)	0.601 (0.035)	0.568 (0.005)
GIN	24.92 (1.67)	32.31 (0.26)	0.829 (0.027)	0.669 (0.017)	1.015 (0.041)	0.331 (0.016)	0.648 (0.024)	0.595 (0.014)
CIGIN	19.32 (0.35)	25.09 (0.32)	0.804 (0.010)	0.607 (0.024)	0.905 (0.014)	0.308 (0.018)	0.411 (0.008)	0.451 (0.009)
CGIB	<b>17.87</b> (0.38)	24.44 (0.21)	0.796 (0.010)	0.568 (0.013)	<b>0.831</b> (0.012)	0.277 (0.008)	0.396 (0.009)	0.428 (0.009)
CGIB <sub>cont</sub>	18.11 (0.20)	<b>23.90</b> (0.35)	<b>0.771</b> (0.005)	<b>0.538</b> (0.007)	0.852 (0.022)	<b>0.276</b> (0.017)	<b>0.390</b> (0.006)	<b>0.422</b> (0.005)

Performance on Molecular Interaction (Regression)

## Observations

- Outperforms baselines on both Molecular Interaction / Drug-Drug Interaction tasks

	(a) Transductive				(b) Inductive			
	ZhangDDI		ChChMiner		ZhangDDI		ChChMiner	
	AUROC	Accuracy	AUROC	Accuracy	AUROC	Accuracy	AUROC	Accuracy
GCN	91.64 (0.31)	83.31 (0.61)	94.71 (0.33)	87.36 (0.24)	68.39 (1.85)	63.78 (1.55)	73.63 (0.44)	67.07 (0.66)
GAT	92.10 (0.28)	84.14 (0.38)	96.15 (0.53)	89.49 (0.88)	69.99 (2.95)	64.41 (1.39)	75.72 (1.66)	68.77 (1.48)
MPNN	92.34 (0.35)	84.56 (0.31)	96.25 (0.53)	90.02 (0.42)	71.54 (1.24)	65.12 (1.14)	75.45 (0.32)	68.24 (1.42)
GIN	93.16 (0.04)	85.59 (0.05)	97.52 (0.05)	91.89 (0.66)	72.74 (1.32)	66.16 (1.21)	74.63 (0.48)	67.80 (0.46)
SSI-DDI	92.74 (0.12)	84.61 (0.18)	98.44 (0.08)	93.50 (0.16)	73.29 (2.23)	66.53 (1.31)	78.24 (1.29)	70.69 (1.47)
MIRACLE	93.05 (0.07)	84.90 (0.36)	88.66 (0.37)	84.29 (0.14)	73.23 (3.32)	50.00 (0.00)	60.25 (0.56)	50.09 (0.11)
CIGIN	93.28 (0.13)	85.54 (0.30)	98.51 (0.10)	93.77 (0.25)	74.02 (0.10)	66.81 (0.09)	79.23 (0.51)	71.56 (0.38)
CGIB	<b>94.27</b> (0.47)	<b>86.88</b> (0.56)	98.80 (0.04)	<b>94.69</b> (0.16)	74.59 (0.88)	<b>67.65</b> (1.07)	81.14 (1.20)	72.47 (0.16)
CGIB <sub>cont</sub>	93.78 (0.62)	86.36 (0.75)	<b>98.84</b> (0.31)	94.52 (0.38)	<b>75.08</b> (0.34)	67.31 (0.82)	<b>81.51</b> (0.67)	<b>74.29</b> (0.14)

Performance on Drug-Drug Interaction (Classification)

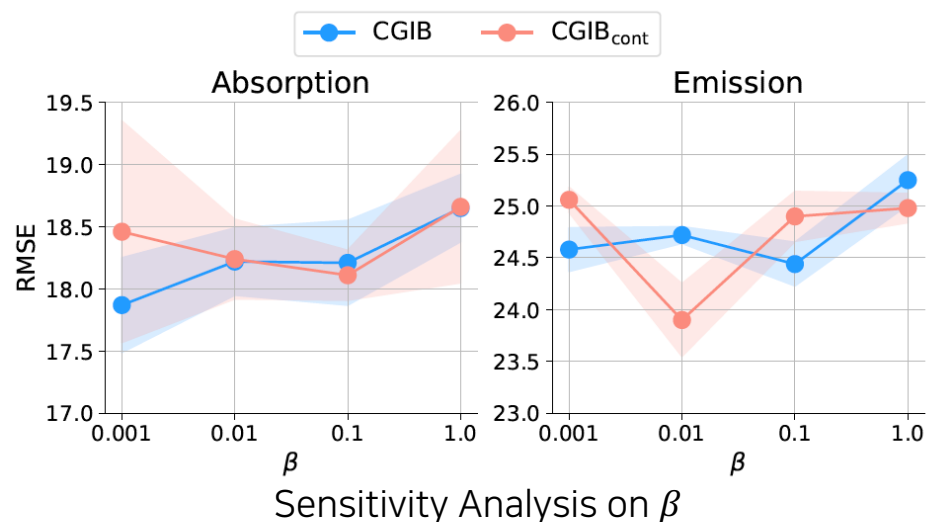
Evaluation on drugs unseen during training

## Observations

- Improvement gap is larger in inductive setting
  - ∴ By detecting function group that is basic in nature → helps generalization



# Result: Analysis on $\beta$



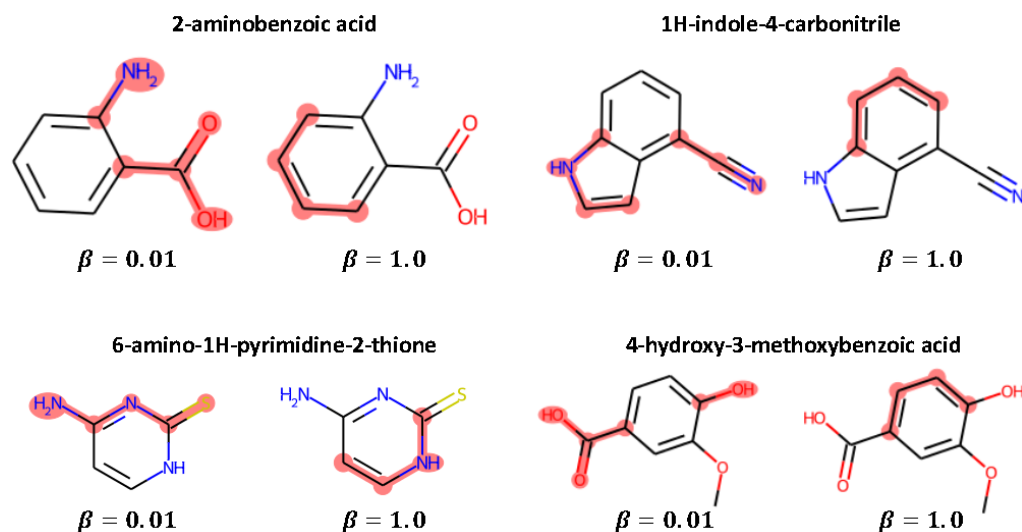
$$\min -I(Y; \mathcal{G}_{\text{CIB}}^1 | \mathcal{G}^2) + \beta I(\mathcal{G}^1; \mathcal{G}_{\text{CIB}}^1 | \mathcal{G}^2)$$

- $\beta$  Controls Trade-off btw prediction and compression

As  $\beta$  increases, Compression > Prediction

## Observations - Sensitivity Analysis

- $\beta = 1.0$ : Poor performance in general (focus on compression)
  - However, the model fails to detect functional group when  $\beta$  is too small  $\rightarrow$  poor generalization
- $\rightarrow$  Hence, finding an appropriate  $\beta$  is crucial



Qualitative Analysis on  $\beta$

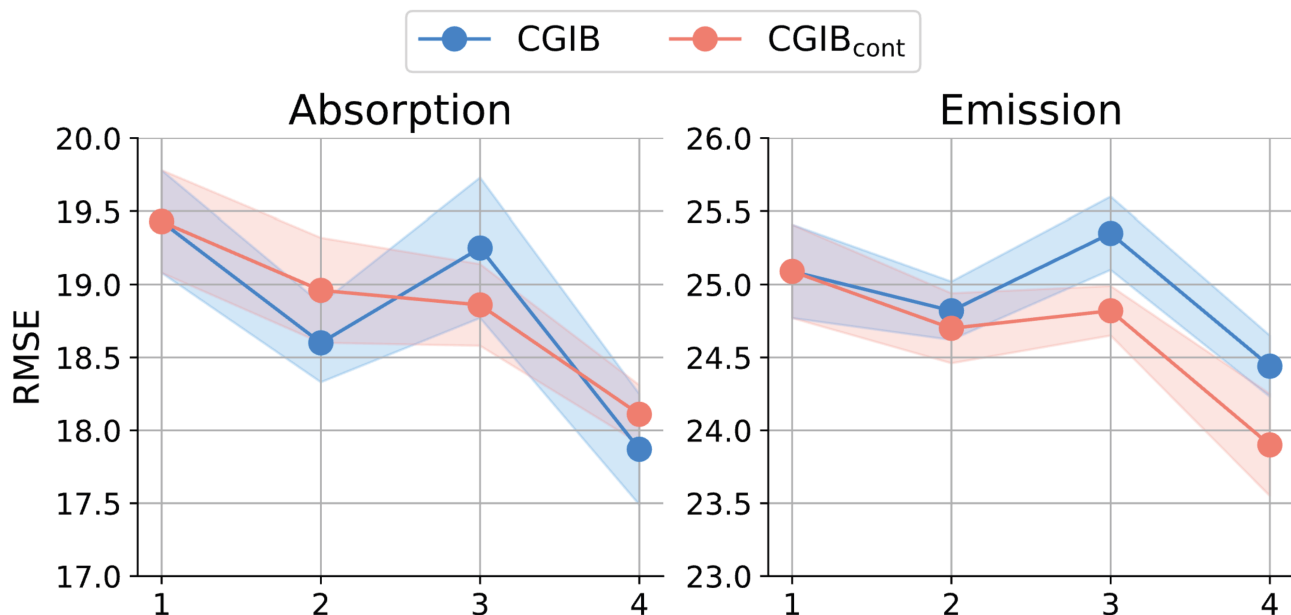
## Observations - Qualitative analysis

- $\beta = 1.0 \rightarrow$  CGIB focuses on compression  
e.g., CGIB focuses an aromatic ring, which is not relevant to chemical reactions
- $\beta = 0.0 \rightarrow$  CGIB does not compress
- $\beta = 0.01 \rightarrow$  Balance between prediction and compression  
e.g., CGIB focuses on external part, which generally more relevant to chemical reactions



# Result: Ablation studies

$$\begin{aligned} & \min -I(Y; \mathcal{G}_{\text{CIB}}^1 | \mathcal{G}^2) + \beta I(\mathcal{G}^1; \mathcal{G}_{\text{CIB}}^1 | \mathcal{G}^2) \\ &= \min -I(Y; \mathcal{G}_{\text{CIB}}^1 | \mathcal{G}^2) + \beta (I(\mathcal{G}_{\text{CIB}}^1; \mathcal{G}^1, \mathcal{G}^2) - I(\mathcal{G}_{\text{CIB}}^1; \mathcal{G}^2)) \\ &= \min -I(Y; \mathcal{G}_{\text{CIB}}^1, \mathcal{G}^2) + \beta (I(\mathcal{G}_{\text{CIB}}^1; \mathcal{G}^1, \mathcal{G}^2) - I(\mathcal{G}_{\text{CIB}}^1; \mathcal{G}^2)) \end{aligned}$$



## Observations - Ablation Studies

- Considering conditional MI is the key for success in relational learning
- A naïve consideration of  $\mathcal{G}^1$  and  $\mathcal{G}^2$  rather performs worse than considering  $\mathcal{G}^1$  only

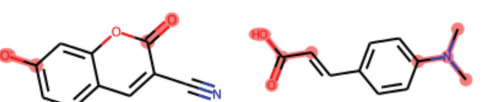
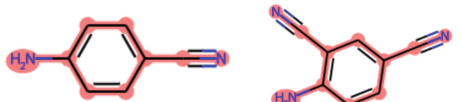
- Without IB  $\rightarrow \min - I(Y; \mathcal{G}^1, \mathcal{G}^2)$  (Same as CIGIN)
- $I(\mathcal{G}_{\text{CIB}}^1; \mathcal{G}^1)$   $\rightarrow \min - I(Y; \mathcal{G}_{\text{CIB}}^1, \mathcal{G}^2) + I(\mathcal{G}_{\text{CIB}}^1; \mathcal{G}^1)$  (Same as VGIB)
- $I(\mathcal{G}_{\text{CIB}}^1; \mathcal{G}^1, \mathcal{G}^2)$   $\rightarrow \min - I(Y; \mathcal{G}_{\text{CIB}}^1, \mathcal{G}^2) + I(\mathcal{G}_{\text{CIB}}^1; \mathcal{G}^1, \mathcal{G}^2)$
- $I(\mathcal{G}^1; \mathcal{G}_{\text{CIB}}^1 | \mathcal{G}^2)$   $\rightarrow \min - I(Y; \mathcal{G}_{\text{CIB}}^1, \mathcal{G}^2) + I(\mathcal{G}^1; \mathcal{G}_{\text{CIB}}^1 | \mathcal{G}^2)$  (Same as CGIB)

Importance of IB

Importance of conditional IB

Importance of valid conditional IB

# Result: Qualitative analysis

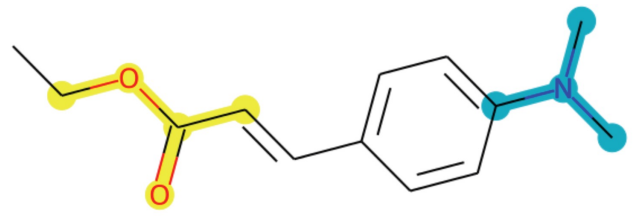
Solvent ( $G^2$ )	(a) Ordinary solvents	(b) Liquid oxygen solvent
Chromophore ( $G^1$ )		

(a) Chromophore ( $G^1$ ) interact with ordinary solvents ( $G^2$ )

Focus on external parts → Aligns with domain knowledge

(b) Chromophore ( $G^1$ ) interact with liquid oxygen solvents ( $G^2$ )

Focus on all parts → Aligns with domain knowledge

Solvent ( $G^2$ )	(c) <div>Ethanol, THF 1-hexanol, 1-butanol</div> <div>benzene</div>
Chromophore ( $G^1$ ) EDAC	 <div>Oxygen-Carbon</div> <div>Nitrogen-Carbon</div>

(c) Chromophore ( $G^1$ ) interacts with various solvents ( $G^2$ ) (e.g., Trans-ethyl p-(dimethylamino) cinnamate (EDAC))

\* Detected parts in chromophore depend on the polarity of solvent

- Case 1: High polarity solvent (Ethanol, THF, 1-hexanol, 1-butanol)

Structure with high polarity is detected (e.g., Oxygen-carbon)

→ Interact with high polarity solvent

- Case 2: Low polarity solvent (Benzene solvent)

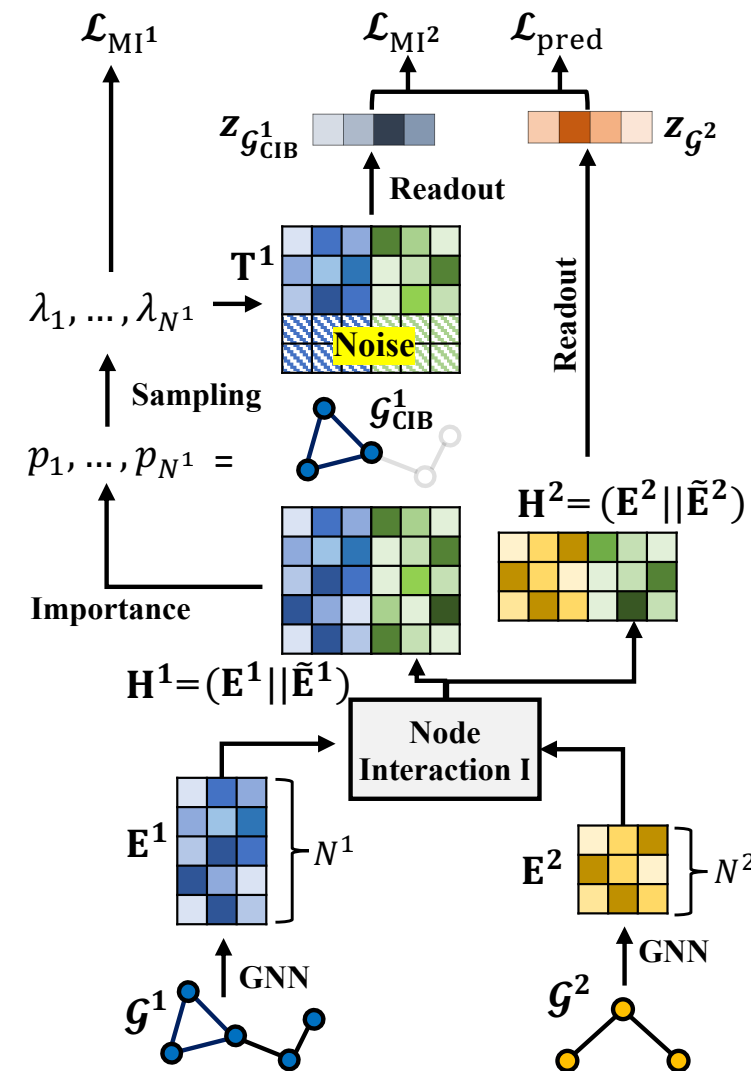
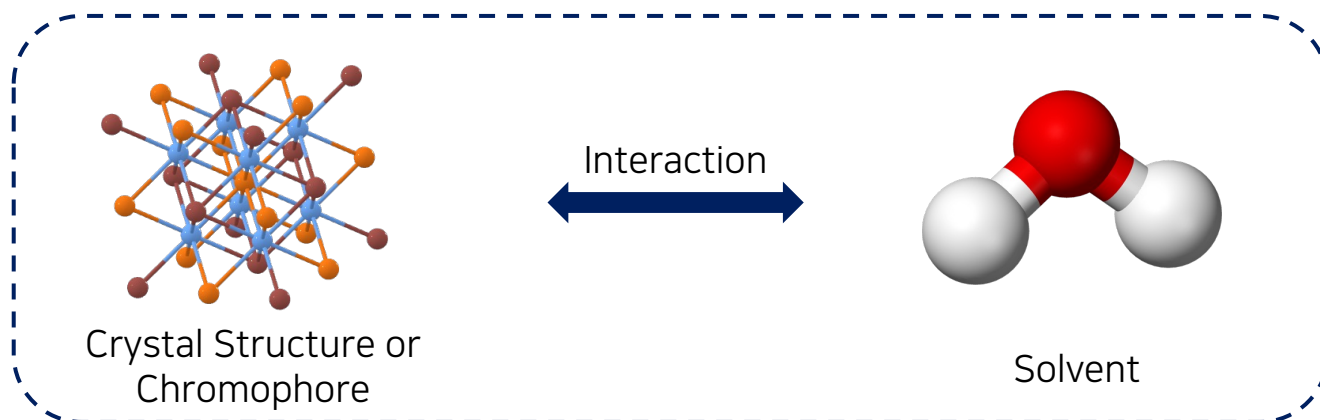
Structure with low polarity is detected (e.g., Nitrogen-Carbon)

→ Interact with low polarity solvent

Detected structure of Chromophore ( $G^1$ ) depends on the paired solvents ( $G^2$ )

# Conclusion

- Proposed a method for tackling relation learning tasks, which are crucial in materials science
  - Based on Conditional Information Bottleneck
- It is crucial to consider Graph 2 (Solvent) when detecting the important subgraph from Graph 1 (Chromophore)
  - i.e., Make use of  $\mathcal{G}^2$  when detecting  $\mathcal{G}_{\text{CIB}}^1$  of  $\mathcal{G}^1$
- CGIB has interpretability, which makes it highly practical



# Outline

- 그래프 신경망 개요 (20 mins)
  - 그래프 신경망 전반적인 소개
  - 그래프 종류에 따른 다양한 그래프 신경망 소개
- How to address Out-of-distribution problem (세부 기술 및 Q&A) (90~120 mins)
  - 소재 물성 예측 연구
    - 소재 물성 예측 연구 최신 동향 소개
    - Transformer 기반 모델 소개 → Prompt-based method
    - Extrapolation을 위한 모델 소개 → Nonlinearity encoding-based method
  - 물질 간 화학 반응 예측 연구
    - 물질 간 화학 반응 예측 연구 동향 소개
    - 정보 이론(Information bottleneck) 기반 모델 소개 → Information bottleneck-based method
    - 인과추론(Causal inference) 기반 모델 소개 → Causal inference-based method

# Papers

## ■ General

- Modeling polypharmacy side effects with graph convolutional networks. Bioinformatics 2018
- Chemically interpretable graph interaction network for prediction of pharmacokinetic properties of drug-like molecules. AAAI 2020
- Multi-view graph contrastive representation learning for drug-drug interaction prediction. WWW 2021

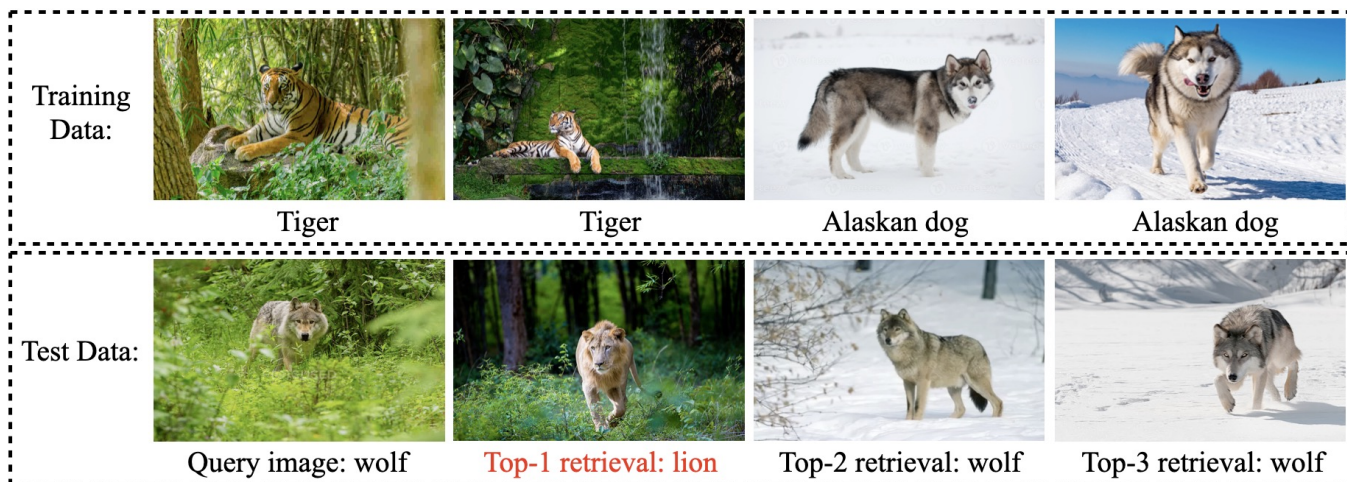
## ■ Information bottleneck-based

- Graph information bottleneck for subgraph recognition. ICLR 2021
- Interpretable and generalizable graph learning via stochastic attention mechanism. ICML 2022
- Improving subgraph recognition with variational graph information bottleneck. CVPR 2022
- **Conditional Graph Information Bottleneck for Molecular Relational Learning. ICML 2023**

## ■ Causal inference-based

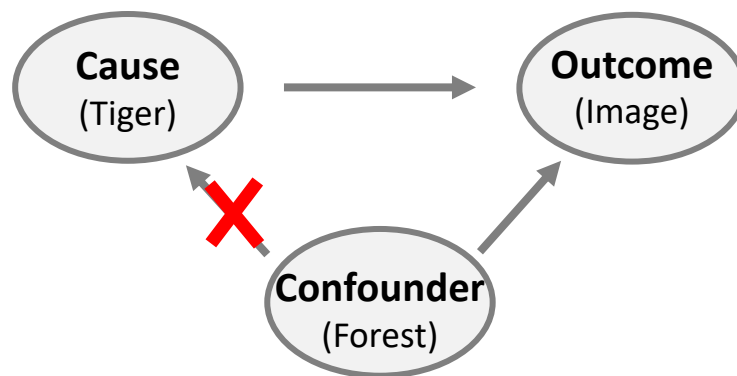
- Discovering invariant rationales for graph neural networks. ICLR 2022
- Debiasing Graph Neural Networks via Learning Disentangled Causal Substructure. NeurIPS 2022
- Causal attention for interpretable and generalizable graph classification. KDD 2022
- **Shift-robust molecular relational learning with causal substructure. KDD 2023**

# Background Causal Inference



- Due to the empirical process of **data collection**, the data for machine learning is heavily **biased**
- Context of the given data becomes a confounder that misleads the machine learning model to learn **spurious correlations (shortcut)** between pixels and labels

ex) Spurious correlation between “Forest” and “Tiger”

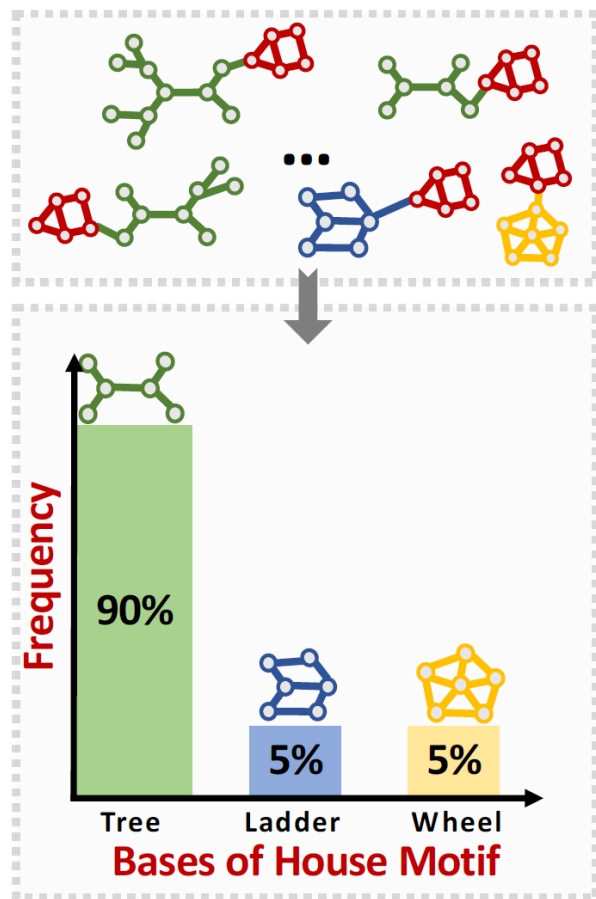


Structure Causal Model (SCM)

Causal Inference aims to improve model performance by **removing spurious correlations**

# Background

Causal Inference for graph structured data

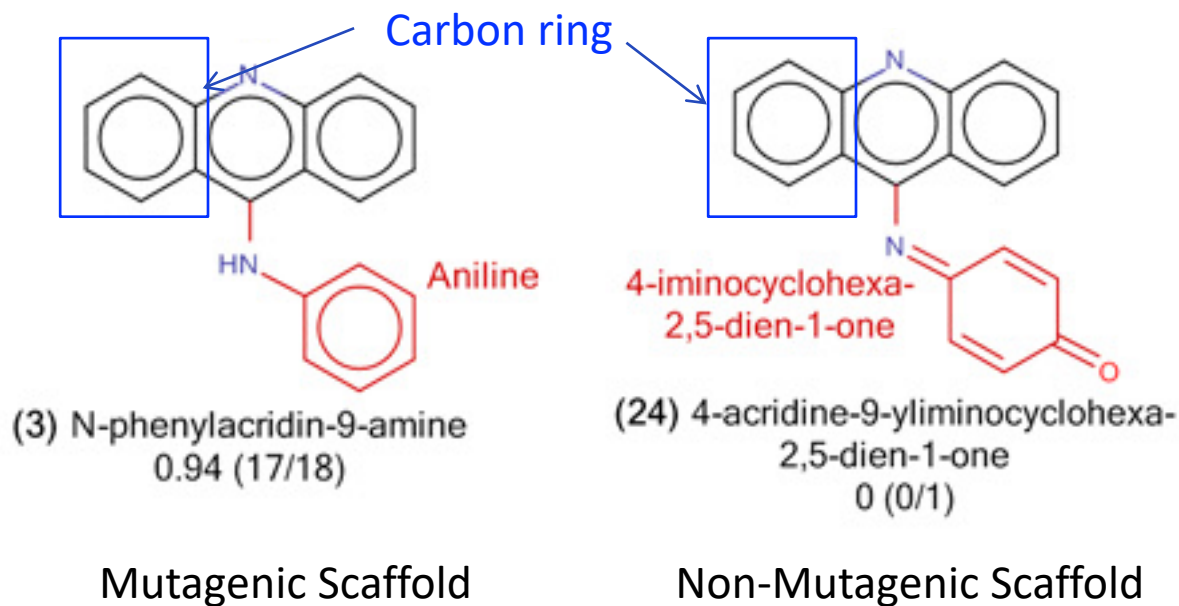


- **Task:** Determining whether a graph contains **House Motifs**
  - **Observation:** Statistical Shortcuts link the **Tree motifs** with **House motifs**
- When facing with out-of-distribution (OOD) data, statistical shortcuts will severely deteriorate the model performance (since the shortcuts will change)

# Background

## Causal Inference for graph structured data

- Example of spurious correlation in molecule property prediction
  - Instead of probing into the causal effect of the functional groups, model focuses on “carbon rings” as the cues of the mutagenic class

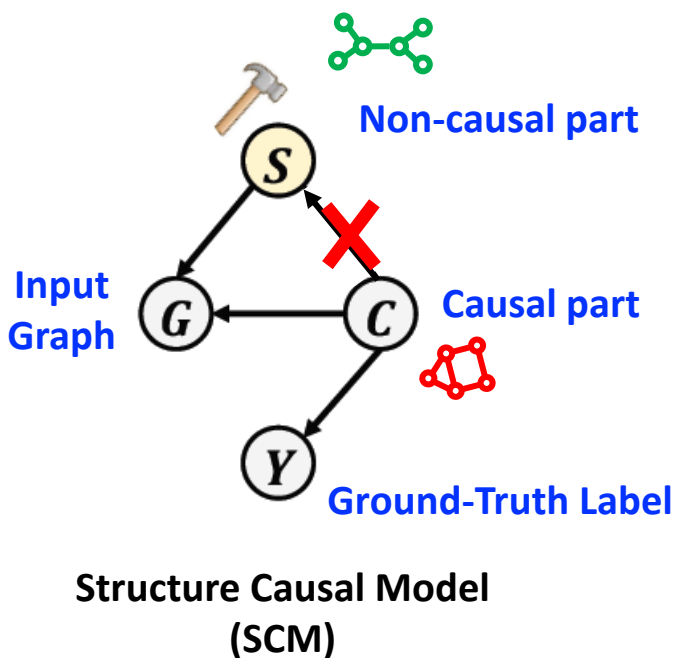


→ In fact, “Carbon ring” has no relationship with mutagenicity

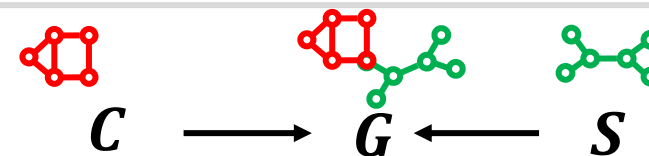


# Discovering Invariant Rationales for Graph Neural Networks (1/4)

- Key idea: Causal patterns are stable (invariant) to distribution shift
  - Causal patterns** (e.g., Tiger) to the labels remain stable across **confounder** (or environments) (e.g., forest, snow), while the relations between the confounder (e.g., forest, snow) and the labels (e.g., contains Tiger or not) vary



Input graph  $G$  consists of two disjoint part:  
- Causal part  $C$  and Non-causal part  $S$



Causal part  $C$  only determines target value  $Y$



Dependency between  $C$  and  $S$

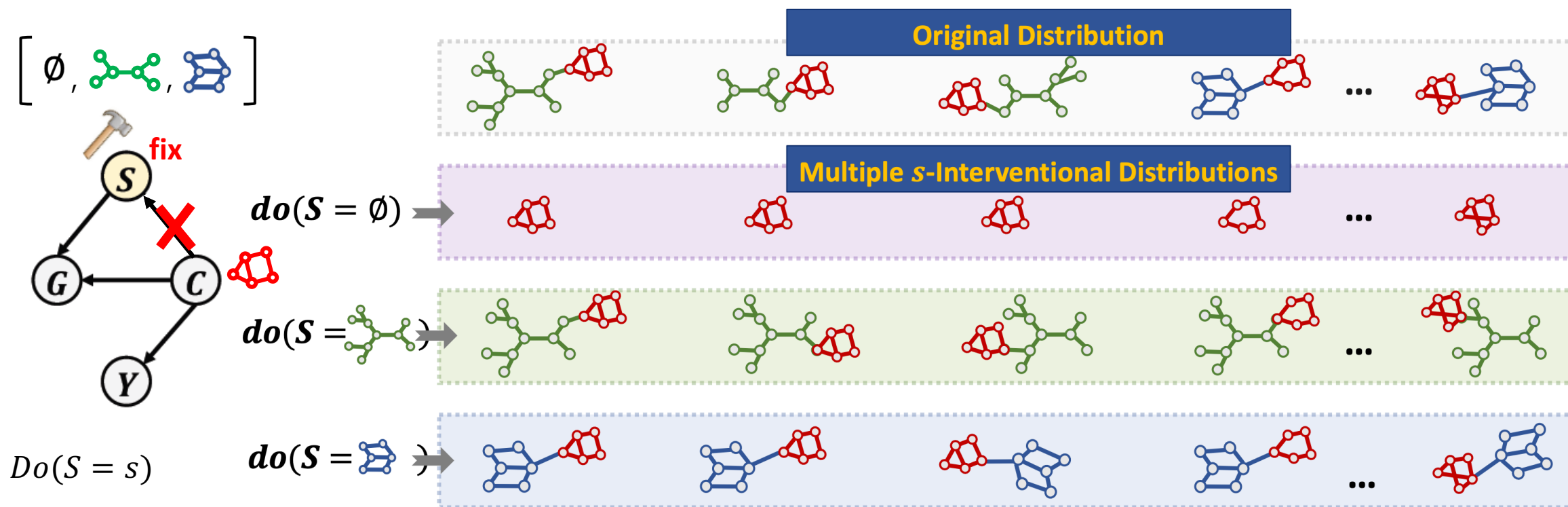
→ Create spurious correlation between  $S$  and  $Y$  ( $S \leftarrow C \rightarrow Y$ )



# Discovering Invariant Rationales for Graph Neural Networks (2/4)

- Research question: How to get multiple environments from a standard training set?

→ Causal intervention



Generate  $s$ -interventional distribution by doing intervention on  $S$

# Discovering Invariant Rationales for Graph Neural Networks (3/4)

**Definition 1 (DIR Principle)** *An intrinsically-interpretable model  $h$  satisfies the DIR principle if it*

- 1. minimizes all  $s$ -interventional risks:  $\mathbb{E}_s[\mathcal{R}(h(G), Y | do(S = s))]$ , and simultaneously*
- 2. minimizes the variance of various  $s$ -interventional risks:  $\text{Var}_s(\{\mathcal{R}(h(G), Y | do(S = s))\})$ ,*

*where the  $s$ -interventional risk is defined over the  $s$ -interventional distribution for specific  $s \in \mathbb{S}$ .*

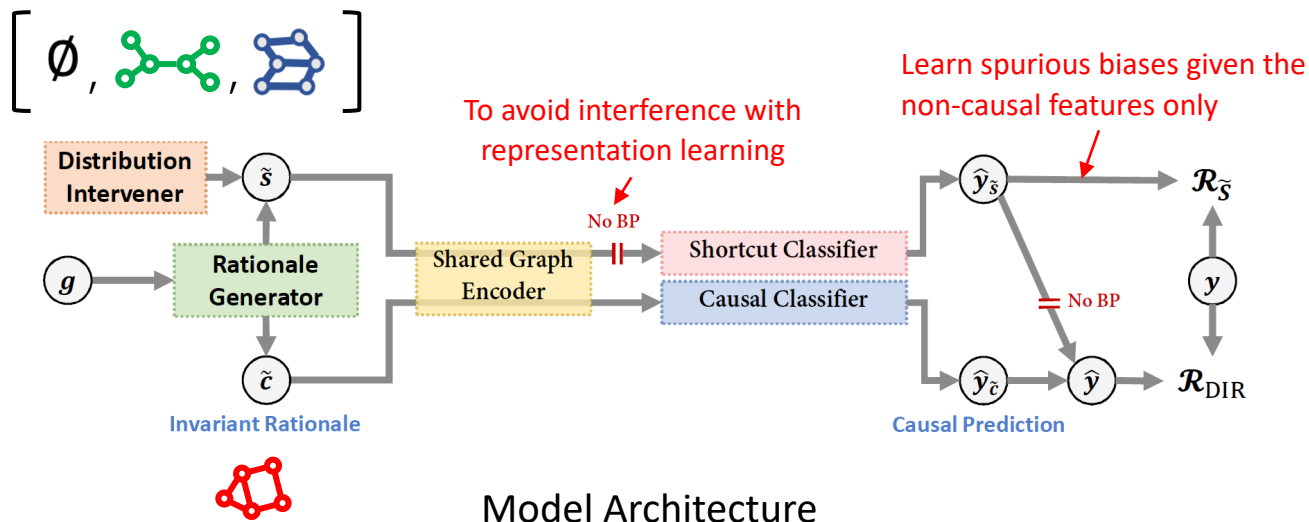
Guided by the proposed principle, we design the learning strategy of DIR as:

$$\min \mathcal{R}_{\text{DIR}} = \mathbb{E}_s[\mathcal{R}(h(G), Y | do(S = s))] + \lambda \text{Var}_s(\{\mathcal{R}(h(G), Y | do(S = s))\}), \quad (4)$$

where  $\mathcal{R}(h(G), Y | do(S = s))$  computes the risk under the  $s$ -interventional distribution, which we will elaborate in Section 2.4.  $\text{Var}(\cdot)$  calculates the variance of risks over different  $s$ -interventional distributions;  $\lambda$  is a hyper-parameter to control the strength of invariant learning.

1. Minimize the risk under all  $s$ -interventional distributions
2. Minimize variance of risk over different  $s$ -interventional distributions

# Discovering Invariant Rationales for Graph Neural Networks (4/4)



Model Architecture

## Rationale Generator

- Split the input graph instance  $g = (\mathcal{V}, \mathcal{E})$  into two subgraphs: **causal part**  $\tilde{c}$  and **non-causal part**  $\tilde{s}$

$$\mathbf{Z} = \text{GNN}_1(g), \quad \mathbf{M}_{ij} = \sigma(\mathbf{Z}_i^\top \mathbf{Z}_j), \quad \text{Generate mask}$$

$$\mathcal{E}_{\tilde{c}} = \text{Top}_r(\mathbf{M} \odot \mathbf{A}), \quad \mathcal{E}_{\tilde{s}} = \text{Top}_{1-r}((1 - \mathbf{M}) \odot \mathbf{A})$$

## Distribution Intervener

- Collects non-causal part of all instances into a memory bank as  $\tilde{\mathcal{S}}$
- Samples memory  $\tilde{s}_j \in \tilde{\mathcal{S}}$  to conduct intervention  $do(S = \tilde{s}_j)$ , constructing an intervened pair  $(\tilde{c}_i, \tilde{s}_j)$

## Model Prediction

$$\hat{y} = \hat{y}_{\tilde{c}} \odot \sigma(\hat{y}_{\tilde{s}})$$

## Optimization

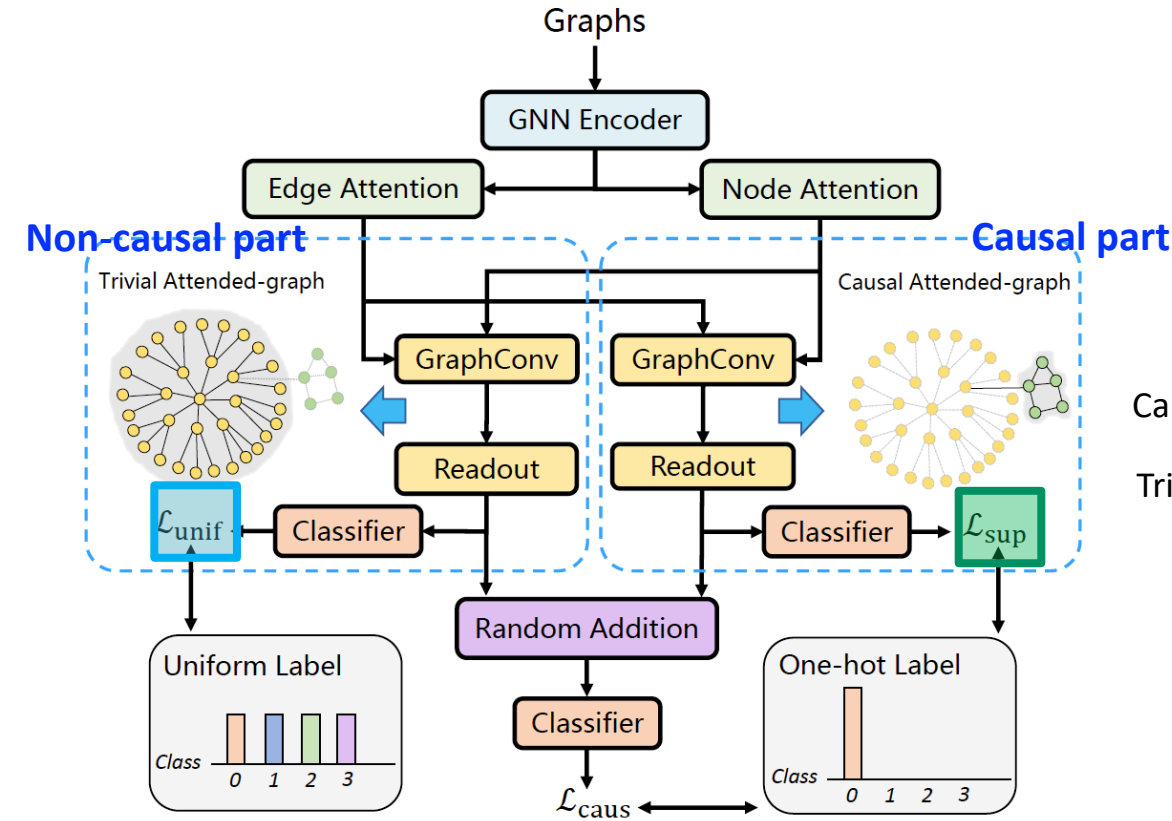
$$\min_{\phi_s} \mathcal{R}_{\tilde{\mathcal{S}}} + \min_{\gamma, \theta, \phi_c} \mathcal{R}_{\text{DIR}}$$

$$\mathcal{R}_{\text{DIR}} = \mathbb{E}_s[\mathcal{R}(h(G), Y | do(S = s))] + \lambda \text{Var}_s(\{\mathcal{R}(h(G), Y | do(S = s))\})$$

$$\mathcal{R}_{\tilde{\mathcal{S}}} = \mathbb{E}_{(g, y) \in \mathcal{O}, \tilde{s} = g/h_{\tilde{c}}(g)} l(\hat{y}_{\tilde{s}}, y)$$

# Causal Attention for Interpretable and Generalizable Graph Classification (1/2)

Task: Graph Classification → “How to classify biased graph datasets?”



Model Architecture

## Soft Mask Estimation

Separate the causal and shortcut features from the full graphs

## Disentanglement

Separate the causal and shortcut features from the full graphs

$$\text{Causal graph } \mathbf{h}_{\mathcal{G}_c} = f_{\text{readout}}(\text{GConv}_c(\mathbf{A} \odot \mathbf{M}_a, \mathbf{X} \odot \mathbf{M}_x)), \quad \mathbf{z}_{\mathcal{G}_c} = \Phi_c(\mathbf{h}_{\mathcal{G}_c})$$

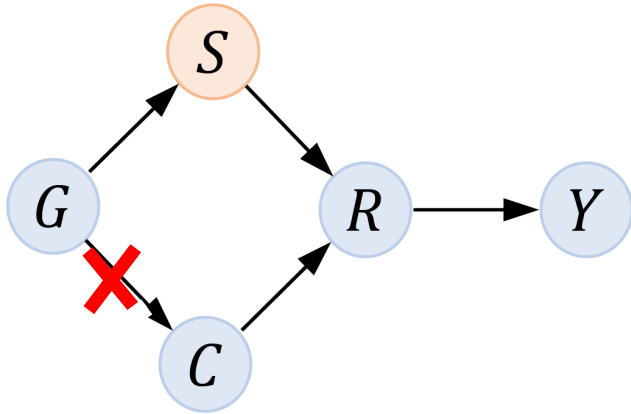
$$\text{Trivial graph } \mathbf{h}_{\mathcal{G}_t} = f_{\text{readout}}(\text{GConv}_t(\mathbf{A} \odot \bar{\mathbf{M}}_a, \mathbf{X} \odot \bar{\mathbf{M}}_x)), \quad \mathbf{z}_{\mathcal{G}_t} = \Phi_t(\mathbf{h}_{\mathcal{G}_t})$$

$$\mathcal{L}_{\text{sup}} = -\frac{1}{|\mathcal{D}|} \sum_{\mathcal{G} \in \mathcal{D}} \mathbf{y}_{\mathcal{G}}^{\top} \log(\mathbf{z}_{\mathcal{G}_c}) \quad \text{Causal graph} \rightarrow \text{Ground truth label prediction}$$

$$\mathcal{L}_{\text{unif}} = \frac{1}{|\mathcal{D}|} \sum_{\mathcal{G} \in \mathcal{D}} \text{KL}(\mathbf{y}_{\text{unif}}, \mathbf{z}_{\mathcal{G}_t}) \quad \text{Trivial graph} \rightarrow \text{Random label prediction}$$

# Causal Attention for Interpretable and Generalizable Graph Classification (2/2)

Task: Graph Classification → “How to classify biased graph datasets?”



$G$  : graph data  
 $C$  : causal feature  
 $S$  : shortcut feature  
 $R$  : representation  
 $Y$  : prediction

Structure Causal Model (SCM)

$$\begin{aligned}
 P(Y|do(C)) &= P_m(Y|C) \\
 &= \sum_{s \in \mathcal{T}} P_m(Y|C, s) P_m(s|C) \quad (\text{Bayes Rule}) \\
 &= \sum_{s \in \mathcal{T}} P_m(Y|C, s) P_m(s) \quad (\text{Independency}) \\
 &= \sum_{s \in \mathcal{T}} P(Y|C, s) P(s),
 \end{aligned}$$

Backdoor Adjustment

← Confounder Set

Causal Intervention via Backdoor adjustment

## Challenges

- Confounder set  $\mathcal{T}$  is commonly unobservable and hard to obtain

**Solution: Let's make implicit intervention on representation level!**

Causal part

Non-causal part from different graphs

$$\begin{aligned}
 \mathbf{z}_{\mathcal{G}'} &= \Phi(\mathbf{h}_{\mathcal{G}_c} + \mathbf{h}_{\mathcal{G}_{t'}}) \\
 \mathcal{L}_{\text{caus}} &= -\frac{1}{|\mathcal{D}| \cdot |\hat{\mathcal{T}}|} \sum_{\mathcal{G} \in \mathcal{D}} \sum_{t' \in \hat{\mathcal{T}}} \mathbf{y}_{\mathcal{G}}^{\top} \log(\mathbf{z}_{\mathcal{G}'} )
 \end{aligned}$$

# Shift-Robust Molecular Relational Learning with Causal Substructure

Namkyeong Lee, Kanghoon Yoon, Gyoung S. Na, Sein Kim, Chanyoung Park

KDD 2023 - International Conference on Knowledge Discovery and Data Mining

# Recall: Relational Learning

## ▪ Molecular Relational Learning

- Learn the interaction behavior between a pair of molecules

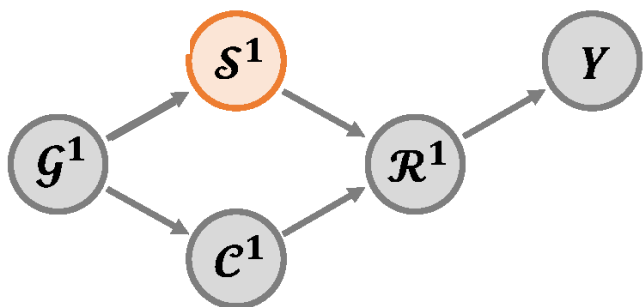


### • Examples

- Predicting **optical properties** when a chromophore (Solute) and solvent (Solvent) react
- Predicting **solubility** when a solute and solvent react
- Predicting **side effects** when taking two types of drugs simultaneously (Polypharmacy effect)



# Shift-Robust Molecular Relational Learning with Causal Substructure



$\mathcal{G}^1$  : Molecule 1

$\mathcal{C}^1$  : Causal Substructure in Molecule 1

$\mathcal{S}^1$  : Shortcut Substructure in Molecule 1

$\mathcal{R}^1$  : Molecule 1 Representation

$Y$  : Target Value

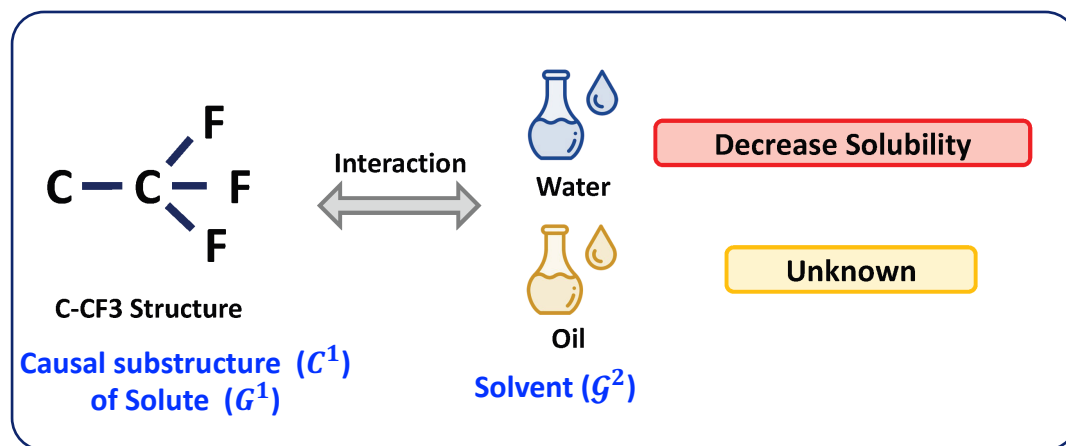
Structure Causal Model (SCM) for  
Molecular Relational Learning

Why not  $\mathcal{S}^2$  and  $\mathcal{C}^2$ ?

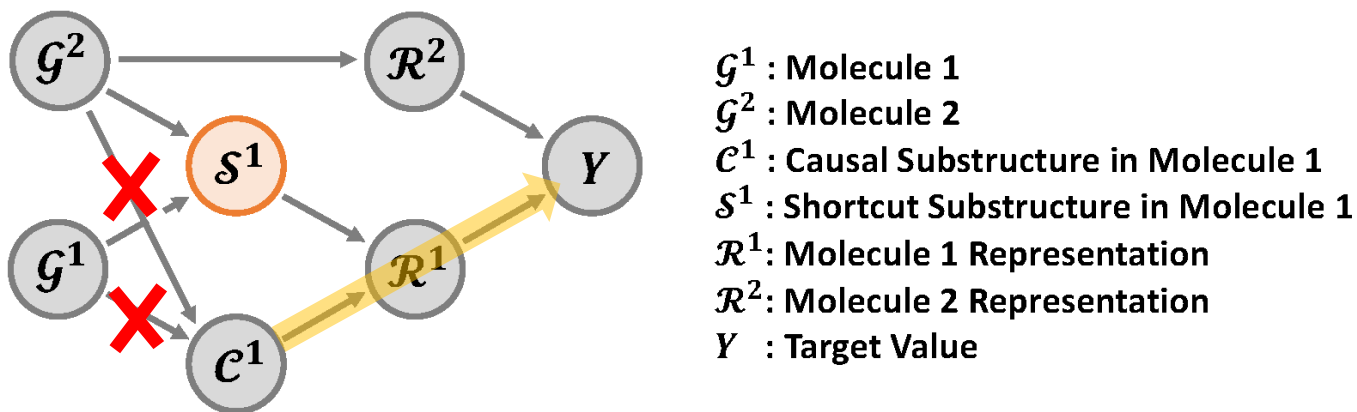
Key causal-effect relationship  
in molecular relational learning

$$\underline{\mathcal{G}^1 \longrightarrow \mathcal{C}^1 \longleftarrow \mathcal{G}^2}$$

Causal substructure  $\mathcal{C}^1$  of molecule  $\mathcal{G}^1$   
→ Determined by not only  $\mathcal{G}^1$  but also  $\mathcal{G}^2$



# Methodology Backdoor adjustment



Structure Causal Model (SCM) for  
Molecular Relational Learning

➔ Causality we are interested in ( $\mathcal{C}^1 \rightarrow Y$ )

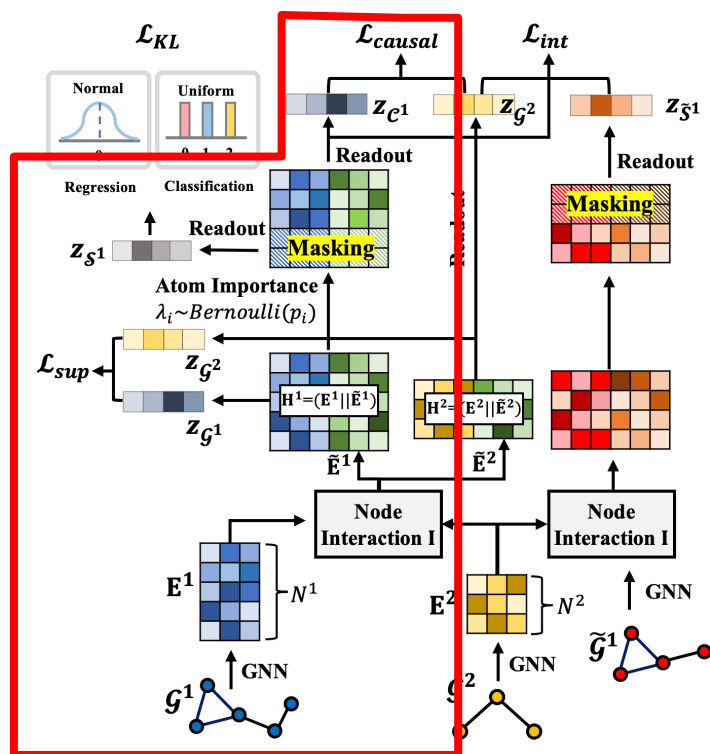
$$\begin{aligned}
 P(Y|do(\mathcal{C}^1), \mathcal{G}^2) &= \tilde{P}(Y|\mathcal{C}^1, \mathcal{G}^2) \\
 &= \sum_s \tilde{P}(Y|\mathcal{C}^1, \mathcal{G}^2, s) \cdot \tilde{P}(s|\mathcal{C}^1, \mathcal{G}^2) \text{ (Bayes' Rule)} \\
 &= \sum_s \tilde{P}(Y|\mathcal{C}^1, \mathcal{G}^2, s) \cdot \tilde{P}(s|\mathcal{G}^2) \text{ (Independence)} \\
 &= \sum_s P(Y|\mathcal{C}^1, \mathcal{G}^2, s) \cdot P(s|\mathcal{G}^2),
 \end{aligned}$$

← Confounder Set

Backdoor Adjustment

Alleviate confounding effect via Backdoor adjustment!

# Methodology Causal molecular relational learner



## Disentangling with Atom Representation Masks

- Separate the causal substructure  $\mathcal{C}^1$  and shortcut substructure  $\mathcal{S}^1$  from  $\mathcal{G}^1$ 
  - Not trivial to explicitly manipulate molecular structure
  - Let's separate in representation space by masking atom representation!

$$p_i = \text{MLP}(\mathbf{H}_i^1) \quad \text{Importance of atom } i$$

$$\mathbf{C}_i^1 = \lambda_i \mathbf{H}_i^1 + (1 - \lambda_i) \epsilon \quad \text{Causal substructure}$$

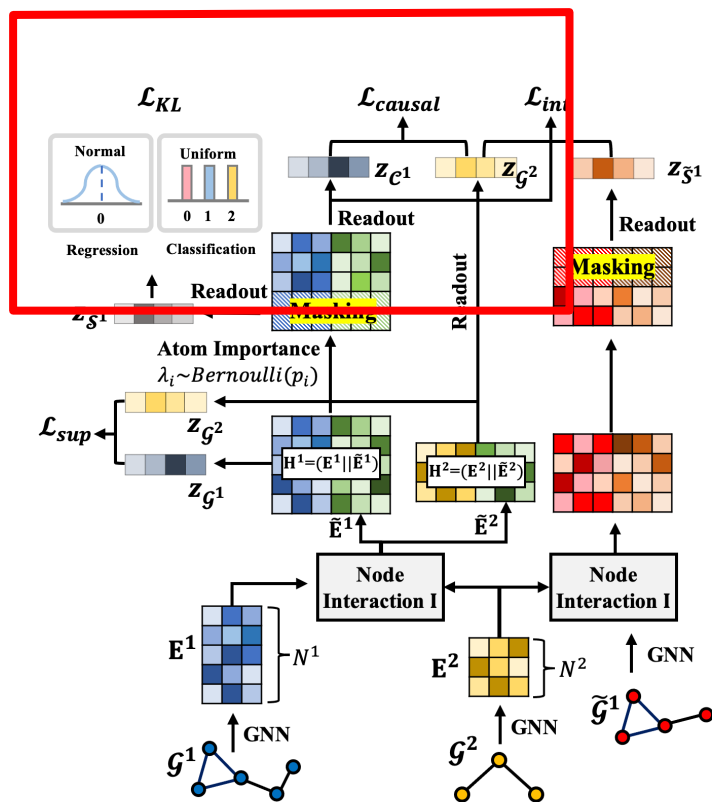
$$\mathbf{S}_i^1 = (1 - \lambda_i) \mathbf{H}_i^1 \quad \text{Shortcut substructure}$$

$$\left. \begin{array}{l} \text{Causal substructure} \\ \text{Shortcut substructure} \end{array} \right\} \text{ where } \lambda_i \sim \text{Bernoulli}(p_i) \quad \epsilon \sim N(\mu_{\mathbf{H}^1}, \sigma_{\mathbf{H}^1}^2)$$

- Gumbel sigmoid approach for differentiable optimization of  $p_i$

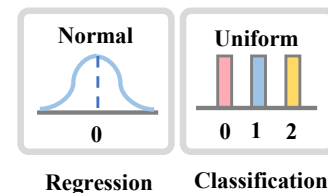
$$\lambda_i = \text{Sigmoid}(1/t \log[p_i/(1 - p_i)] + \log[u/(1 - u)]), \quad u \sim \text{Uniform}(0, 1)$$

# Methodology Causal molecular relational learner



## Disentangling with Atom Representation Masks

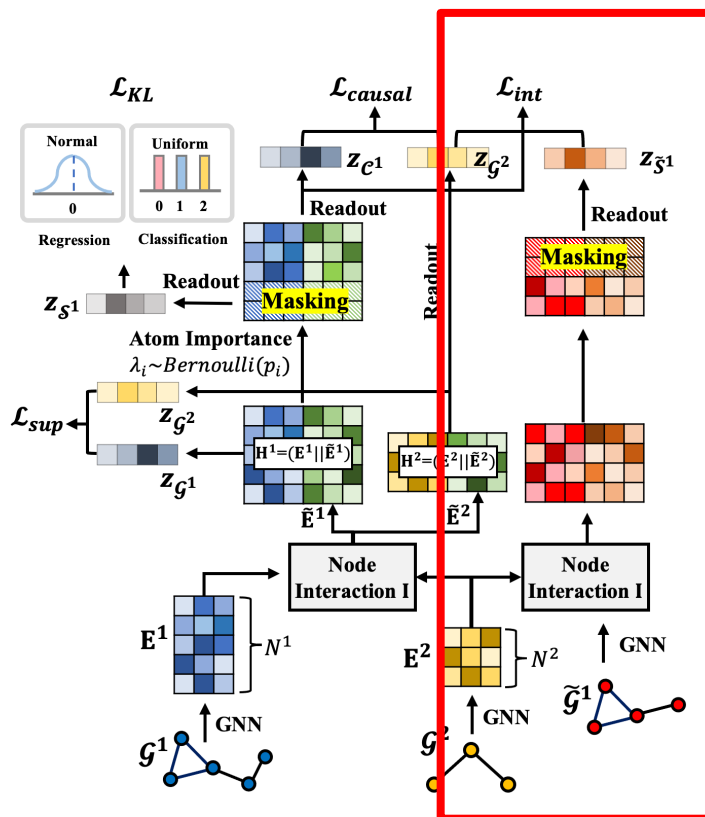
- Causal substructure  $\mathcal{C}^1$ 
  - Cross entropy loss for classification  $\rightarrow \mathcal{L}_{causal}(Y, z_{C^1}, z_{G^2})$
  - RMSE loss for Regression
- Shortcut substructure  $\mathcal{S}^1$ 
  - Learn non informative distribution  $\rightarrow \mathcal{L}_{KL}(Y_{rand}, z_{S^1})$



# Methodology Causal molecular relational learner

$$\begin{aligned}
 P(Y|do(C^1), \mathcal{G}^2) &= \tilde{P}(Y|C^1, \mathcal{G}^2) \\
 &= \sum_s \tilde{P}(Y|C^1, \mathcal{G}^2, s) \cdot \tilde{P}(s|C^1, \mathcal{G}^2) \quad (\text{Bayes' Rule}) \\
 &= \sum_s \tilde{P}(Y|C^1, \mathcal{G}^2, s) \cdot \tilde{P}(s|\mathcal{G}^2) \quad (\text{Independence}) \\
 &= \sum_s P(Y|C^1, \mathcal{G}^2, s) \cdot P(s|\mathcal{G}^2),
 \end{aligned}$$

Backdoor Adjustment



## Conditional Causal Intervention via backdoor adjustment

- Straightforward approach → Generate an intervened molecule structure

### Challenges

- 1) Molecules exist on the basis of various domain knowledge in molecular science
- 2) Intervention space on  $\mathcal{C}^1$  should be conditioned on the paired molecule  $\mathcal{G}^2$

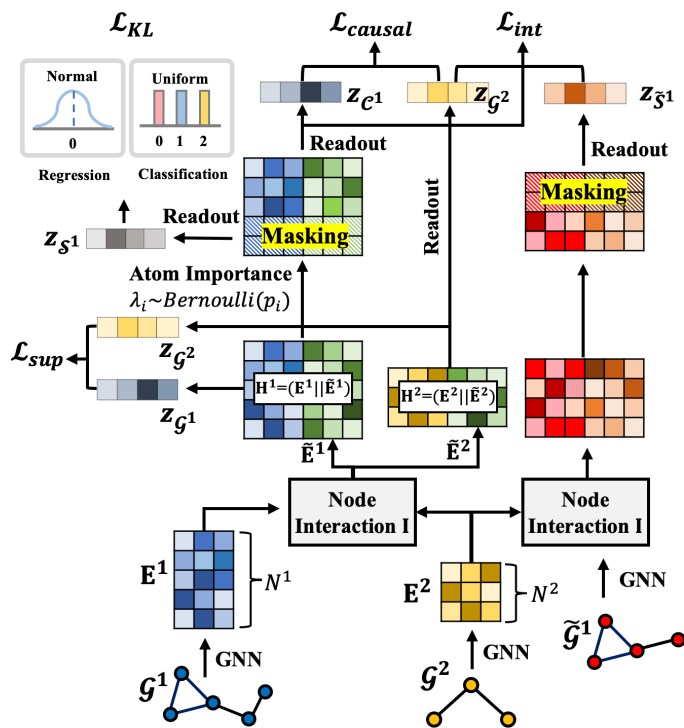
### Our Solution

- Obtain shortcut substructure  $\tilde{\mathcal{S}}^1$  by modeling interaction with other molecules  $\tilde{\mathcal{G}}^1$  and molecule  $\mathcal{G}^2$

$$\mathcal{L}_{int} = \sum_{(\mathcal{G}^1, \mathcal{G}^2) \in \mathcal{D}} \sum_{\tilde{\mathcal{S}}^1} \mathcal{L}(Y, z_{C^1}, z_{\mathcal{G}^2}, z_{\tilde{\mathcal{S}}^1})$$

Causal part of  $\mathcal{G}^1$   
Non-causal part of  $\mathcal{G}^1$  from different graphs  
 $\mathcal{G}^2$

# Methodology Causal molecular relational learner



## Final Objective

$$\mathcal{L}_{final} = \mathcal{L}_{sup} + \mathcal{L}_{causal} + \lambda_1 \cdot \mathcal{L}_{KL} + \lambda_2 \cdot \mathcal{L}_{int}$$

- $\mathcal{L}_{sup}$  : loss with paired graph  $(\mathcal{G}^1, \mathcal{G}^2)$  and target  $Y$
- $\mathcal{L}_{causal}$  : loss with causal substructure
- $\mathcal{L}_{KL}$  : loss with shortcut substructure
- $\lambda_1, \lambda_2$ : weight hyperparameters for  $\mathcal{L}_{KL}$  and  $\mathcal{L}_{int}$

# Experiments

## Dataset description

Dataset		$\mathcal{G}^1$	$\mathcal{G}^2$	# $\mathcal{G}^1$	# $\mathcal{G}^2$	# Pairs	Task
Chromophore <sup>3</sup>	Absorption	Chrom.	Solvent	6416	725	17276	MI
	Emission	Chrom.	Solvent	6412	1021	18141	MI
	Lifetime	Chrom.	Solvent	2755	247	6960	MI
MNSol <sup>4</sup>		Solute	Solvent	372	86	2275	MI
FreeSolv <sup>5</sup>		Solute	Solvent	560	1	560	MI
CompSol <sup>6</sup>		Solute	Solvent	442	259	3548	MI
Abraham <sup>7</sup>		Solute	Solvent	1038	122	6091	MI
CombiSolv <sup>8</sup>		Solute	Solvent	1495	326	10145	MI
ZhangDDI <sup>9</sup>		Drug	Drug	544	544	40255	DDI
ChChMiner <sup>10</sup>		Drug	Drug	949	949	21082	DDI
DeepDDI <sup>11</sup>		Drug	Drug	1704	1704	191511	DDI
AIDS <sup>12</sup>		Mole.	Mole.	700	700	490K	SL
LINUX <sup>12</sup>		Program	Program	1000	1000	1M	SL
IMDB <sup>12</sup>		Ego-net.	Ego-net.	1500	1500	2.25M	SL
OpenSSL <sup>13</sup>		Flow	Flow	4308	4308	18.5M	SL
FFmpeg <sup>13</sup>		Flow	Flow	10824	10824	117M	SL

### Molecular Interaction Dataset

- Predicting Chromophores' Absorption max, Emission max, Lifetime
- Predicting Solvation Free Energy of molecules (MNSol, FreeSolv, CompSol, Abraham, CombiSolv)
- Regression Task

### Drug-Drug Interaction Dataset

- Zhang DDI, ChChMiner, DeepDDI
- Classification Task

### Graph Similarity Learning Dataset

- How similar are the paired graphs? (ex. GED)
- AIDS, LINUX, IMDB, OpenSSL, Ffmpeg
- Regression Task / Classification Task

# Experiments Overall Performance

	Chromophore			MNSol	FreeSolv	CompSol	Abraham	CombiSolv
	Absorption	Emission	Lifetime					
GCN	25.75 (1.48)	31.87 (1.70)	0.866 (0.015)	0.675 (0.021)	1.192 (0.042)	0.389 (0.009)	0.738 (0.041)	0.672 (0.022)
GAT	26.19 (1.44)	30.90 (1.01)	0.859 (0.016)	0.731 (0.007)	1.280 (0.049)	0.387 (0.010)	0.798 (0.038)	0.662 (0.021)
MPNN	24.43 (1.55)	30.17 (0.99)	0.802 (0.024)	0.682 (0.017)	1.159 (0.032)	0.359 (0.011)	0.601 (0.035)	0.568 (0.005)
GIN	24.92 (1.67)	32.31 (0.26)	0.829 (0.027)	0.669 (0.017)	1.015 (0.041)	0.331 (0.016)	0.648 (0.024)	0.595 (0.014)
CIGIN	19.32 (0.35)	25.09 (0.32)	0.804 (0.010)	0.607 (0.024)	0.905 (0.014)	0.308 (0.018)	0.411 (0.008)	0.451 (0.009)
CMRL	<b>17.93</b> (0.31)	<b>24.30</b> (0.22)	<b>0.776</b> (0.007)	<b>0.551</b> (0.017)	<b>0.815</b> (0.046)	<b>0.255</b> (0.011)	<b>0.374</b> (0.011)	<b>0.421</b> (0.008)

Performance on molecular interaction prediction task

	AIDS			LINUX			IMDB			FFmpeg	OpenSSL
	MSE	$\rho$	p@10	MSE	$\rho$	p@10	MSE	$\rho$	p@10	AUROC	AUROC
SimGNN	1.376	0.824	0.400	2.479	0.912	0.635	1.264	0.878	0.759	93.45	94.25
GMN	4.610	0.672	0.200	2.571	0.906	0.888	4.422	0.725	0.604	94.76	93.91
GraphSim	1.919	0.849	0.446	0.471	0.976	0.956	0.743	0.926	0.828	94.48	93.66
HGMN	1.169	<b>0.905</b>	0.456	0.439	0.985	0.955	0.335	0.919	0.837	97.83	95.87
H <sup>2</sup> MN <sub>RW</sub>	0.936	0.878	0.496	0.136	0.988	0.970	0.296	0.918	0.872	<b>99.05</b>	92.21
H <sup>2</sup> MN <sub>NE</sub>	0.924	0.883	0.511	0.130	0.990	0.978	0.297	0.889	0.875	98.16	<b>98.25</b>
CMRL	<b>0.770</b>	0.899	<b>0.574</b>	<b>0.094</b>	<b>0.992</b>	<b>0.989</b>	<b>0.263</b>	<b>0.944</b>	<b>0.879</b>	98.69	96.57

Performance on graph similarity learning task

## Observations

### 1. CMRL outperforms all other baseline methods

→ It is crucial to discover causally related substructure in molecules

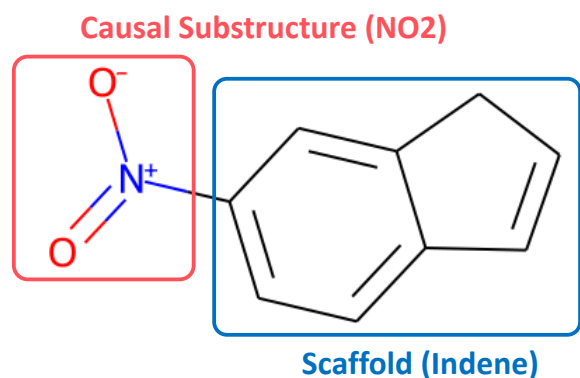
### 2. Wide applicability of CMRL beyond molecules

→ Performs well in dataset that contains core substructure

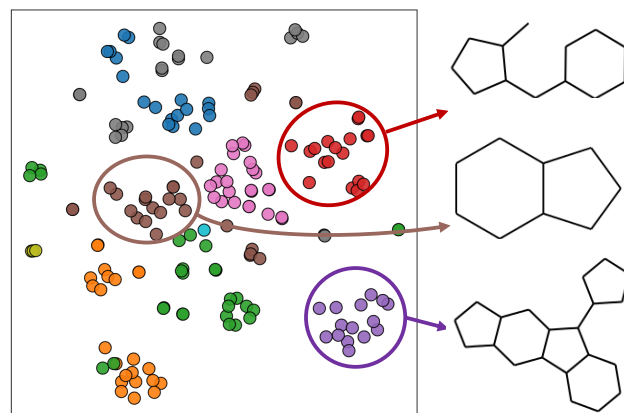


# Experiments Out-of-distribution performance

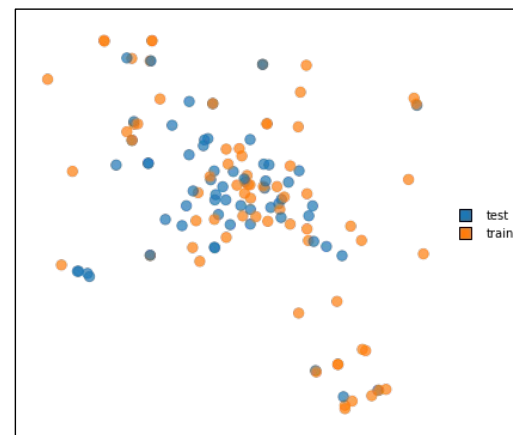
In OOD experiment, we assess the model's performance on molecules belonging to new scaffold classes (functional group)



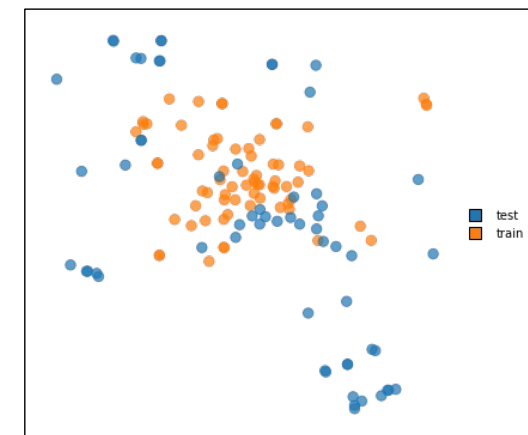
Molecule: 6-nitro-1H-indene



TSNE embeddings



Random Split



Scaffold Split

TSNE on splitted data (Train / Test)

Different scaffolds exhibit totally different distribution

# Experiments Out-of-distribution performance

In OOD experiment, we assess the model's performance on molecules belonging to new scaffold classes (functional group)

	(a) In-Distribution						(b) Out-of-Distribution					
	ZhangDDI		ChChMiner		DeepDDI		ZhangDDI		ChChMiner		DeepDDI	
	AUROC	Accuracy	AUROC	Accuracy	AUROC	Accuracy	AUROC	Accuracy	AUROC	Accuracy	AUROC	Accuracy
GCN	91.64 (0.31)	83.31 (0.61)	94.71 (0.33)	87.36 (0.24)	92.02 (0.01)	86.96 (0.02)	70.61 (2.32)	64.22 (1.64)	74.17 (0.89)	67.56 (1.29)	76.38 (0.43)	67.92 (0.81)
GAT	92.10 (0.28)	84.14 (0.38)	96.15 (0.53)	89.49 (0.88)	92.01 (0.02)	86.99 (0.05)	73.15 (2.50)	65.14 (2.47)	75.64 (0.99)	68.61 (0.72)	76.44 (1.27)	67.94 (1.38)
MPNN	92.34 (0.35)	84.56 (0.31)	96.25 (0.53)	90.02 (0.42)	92.02 (0.02)	86.97 (0.01)	72.39 (1.70)	64.55 (1.75)	76.40 (0.91)	68.51 (0.71)	79.03 (0.81)	71.23 (0.90)
GIN	93.16 (0.04)	85.59 (0.05)	97.52 (0.05)	91.89 (0.66)	92.03 (0.00)	87.02 (0.03)	75.04 (0.63)	67.14 (1.03)	74.32 (2.93)	67.49 (2.44)	78.61 (0.58)	70.33 (1.11)
MIRACLE	93.05 (0.07)	84.90 (0.36)	88.66 (0.37)	84.29 (0.14)	62.23 (0.75)	62.35 (0.30)	59.57 (0.90)	52.31 (2.24)	73.28 (0.71)	50.49 (0.59)	62.32 (1.63)	51.30 (0.29)
SSI-DDI	92.74 (0.12)	84.61 (0.18)	98.44 (0.08)	93.50 (0.16)	93.97 (0.38)	88.44 (0.39)	71.67 (4.71)	65.78 (3.02)	75.59 (1.93)	68.75 (1.41)	80.41 (1.74)	72.05 (1.47)
CIGIN	93.28 (0.13)	85.54 (0.30)	98.51 (0.10)	93.77 (0.25)	99.12 (0.03)	96.55 (0.11)	73.99 (1.74)	66.44 (1.07)	80.24 (2.00)	73.28 (1.08)	83.78 (0.87)	74.07 (1.19)
CMRL	93.73 (0.15)	86.32 (0.23)	98.70 (0.05)	94.26 (0.28)	99.13 (0.02)	96.70 (0.12)	75.30 (1.39)	67.76 (1.41)	82.05 (0.67)	74.21 (0.78)	83.83 (0.97)	75.20 (0.66)

Performance on drug-drug interaction task

## Observation

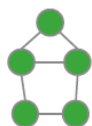
**CMRL outperforms previous work on out-of-distribution scenarios**

→ Learning causal substructure enhances the generalization ability of the model

# Experiments Synthetic dataset experiments

In synthetic dataset experiment, we assess the model's performance on various levels of bias in datasets

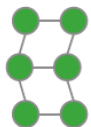
Causal subgraphs:



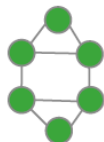
House



Cycle



Grid

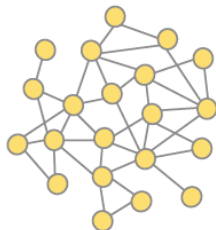


Diamond

Shortcut subgraphs:



Tree



BA  
(Barabasi-Albert)

**Task:** Predict whether a pair of graphs contain the same **causal substructure**

Positive pair

- a pair that shares the same causal substructure
- e.g., {House-Tree, House-BA} → Positive

Negative pair

- a pair that each graph has a different causal substructure
- e.g., {House-Tree, Cycle-Tree} → Negative

Dataset bias

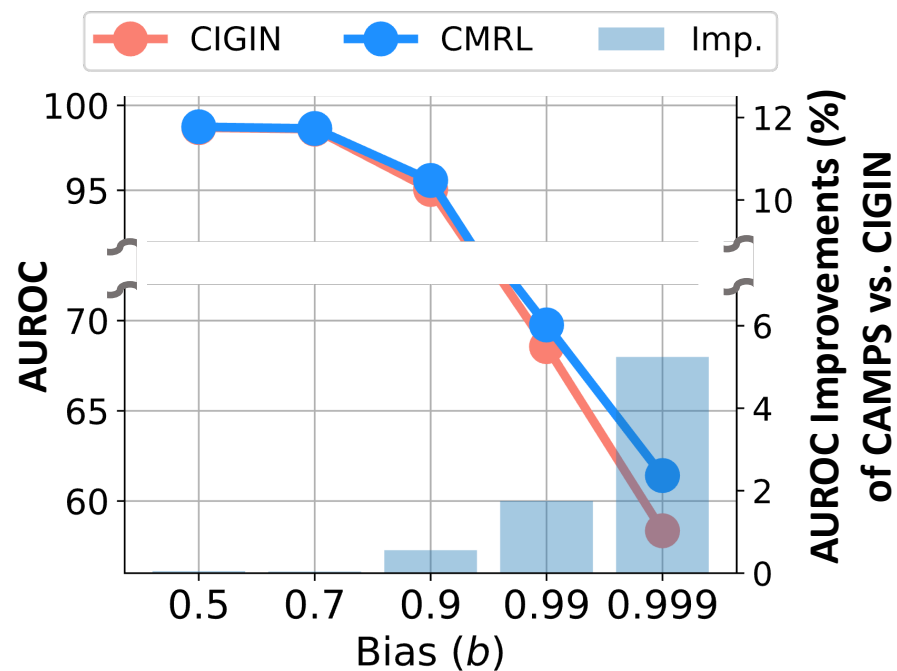
- the ratio of the positive pairs containing “BA” shortcut substructures

$$\begin{aligned}\text{bias}(b) &= \frac{\text{Number of positive pairs with BA substructure}}{\text{Number of positive pairs}} \\ &= \frac{\#\{\text{Causal-BA, Causal-BA}\}}{\#\{\text{Causal-Tree, Causal-Tree}\} + \#\{\text{Causal-BA, Causal-BA}\}}\end{aligned}$$

- Bias level  $b$  increases  
→ “BA” substructures dominates model prediction

# Experiments Synthetic dataset experiments

In synthetic dataset experiment, we assess the model's performance on various levels of bias in datasets



## Observations

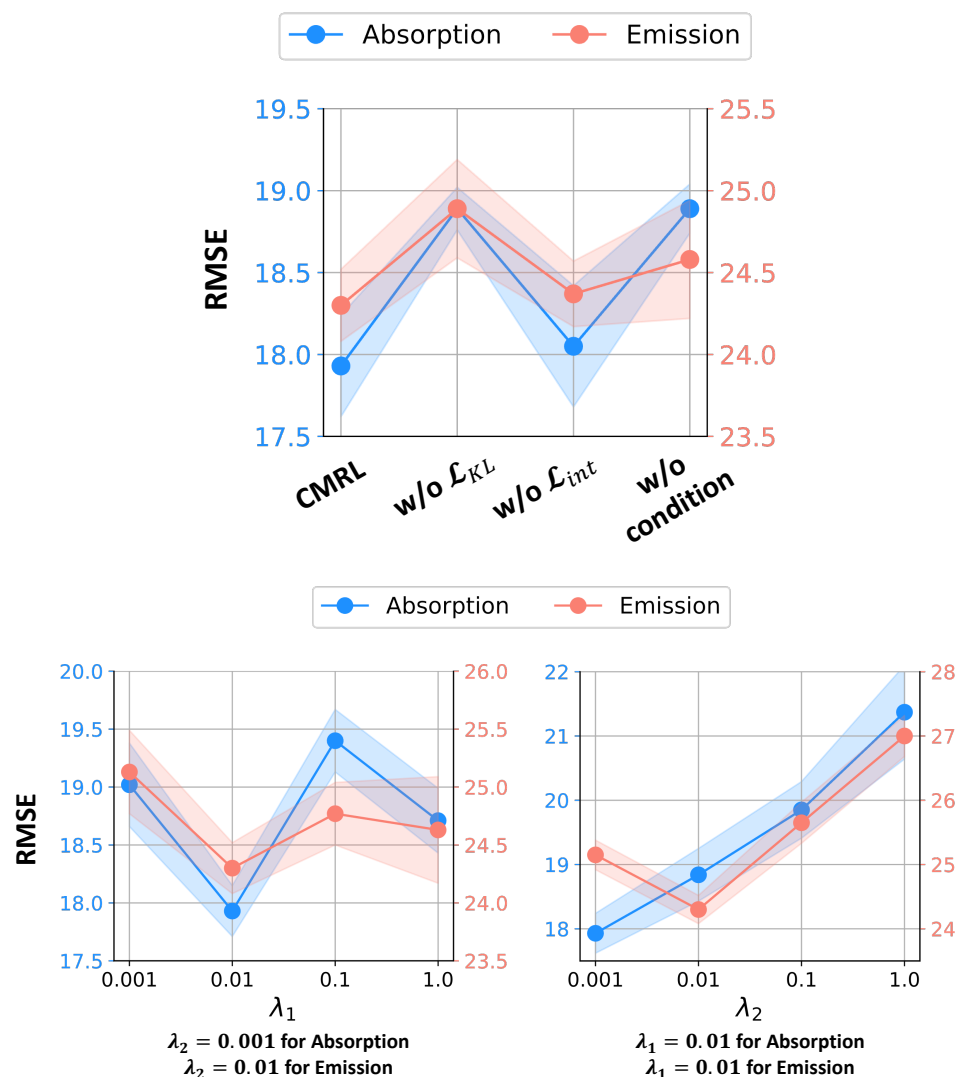
**1. Models' performance degrades as the bias gets severe**

→ “BA” shortcut confound the model

**2. Performance gap between CMRL and CIGIN gets larger as the bias gets severe**

→ Importance of learning causality between the substructure and target

# Experiments Model analysis



## Observations in Ablation Studies

**Naïve intervention whose confounders are not conditioned on paired molecule  $\mathcal{G}^2$**

→ Performs worse than the model without intervention

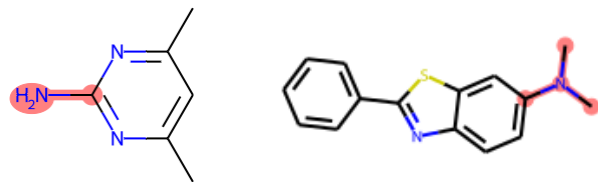
→ Wideness of intervention space introduces noisy signal during model training

## Observations in Sensitivity Analysis

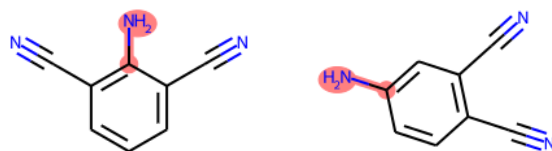
1. Optimal point for  $\lambda_2$  exists that balances between the noisiness and robustness
2. No certain relationship between model performance and  $\lambda_1$

Training objective  $\mathcal{L}_{final} = \mathcal{L}_{sup} + \mathcal{L}_{causal} + \lambda_1 \cdot \mathcal{L}_{KL} + \lambda_2 \cdot \mathcal{L}_{int}$

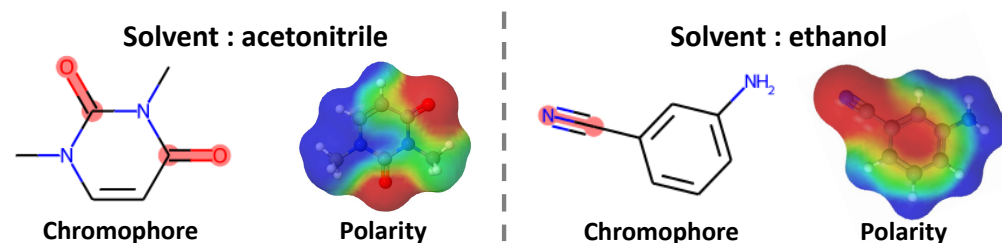
# Experiments Qualitative analysis



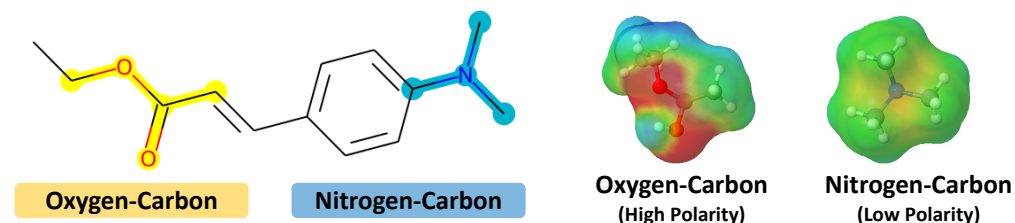
(a) Reaction with ordinary solvent



(b) Reaction with single bond



(c) Reaction with polar solvents



(d) Chromophore: EDAC

Solvents: 1-propanol, 1-butanol

## Observations

### 1. Discovered causal substructure aligns with well-known chemical domain knowledge

- CMRL selects edge substructure → Chemical reactions usually happen around ionized atoms
- CMRL concentrates on single-bonded substructure → Single-bonded substructures are more likely to undergo chemical reactions

### 2. When reacting with polar solvents, CMRL focuses on the edge substructures of high polarity

### 3. Selected important substructures of chromophore varies as the solvent varies

# Outline

- 그래프 신경망 개요 (20 mins)
  - 그래프 신경망 전반적인 소개
  - 그래프 종류에 따른 다양한 그래프 신경망 소개
- How to address Out-of-distribution problem (세부 기술 및 Q&A) (90~120 mins)
  - 소재 물성 예측 연구
    - 소재 물성 예측 연구 최신 동향 소개
    - Transformer 기반 모델 소개 → Prompt-based method
    - Extrapolation을 위한 모델 소개 → Nonlinearity encoding-based method
  - 물질 간 화학 반응 예측 연구
    - 물질 간 화학 반응 예측 연구 동향 소개
    - 정보 이론(Information bottleneck) 기반 모델 소개 → Information bottleneck-based method
    - 인과추론(Causal inference) 기반 모델 소개 → Causal inference-based method

# Papers: Material property prediction

## ■ Material property prediction

- Neural message passing for quantum chemistry. ICML 2017
- Schnet: a continuous-filter convolutional neural network for modeling quantum interactions. NeurIPS 2017
- Crystal Graph Convolutional Neural Networks for an Accurate and Interpretable Prediction of Material Properties. Phys. Rev. Lett. 2018
- Graph networks as a universal machine learning framework for molecules and crystals. Chem. Mater. 2019
- **Predicting Density of States via Multi-modal Transformer. ICLR Workshop 2023**

## ■ Extrapolation

- How Neural Networks Extrapolate: From Feedforward to Graph Neural Networks. ICLR 2021
- **Nonlinearity Encoding for Extrapolation of Neural Networks. KDD 2022**



# Papers: Molecular Relational Learning

## ■ General

- Modeling polypharmacy side effects with graph convolutional networks. Bioinformatics 2018
- Chemically interpretable graph interaction network for prediction of pharmacokinetic properties of drug-like molecules. AAAI 2020
- Multi-view graph contrastive representation learning for drug-drug interaction prediction. WWW 2021

## ■ Information bottleneck-based

- Interpretable and generalizable graph learning via stochastic attention mechanism. ICML 2022
- Improving subgraph recognition with variational graph information bottleneck. CVPR 2022
- **Conditional Graph Information Bottleneck for Molecular Relational Learning. ICML 2023**

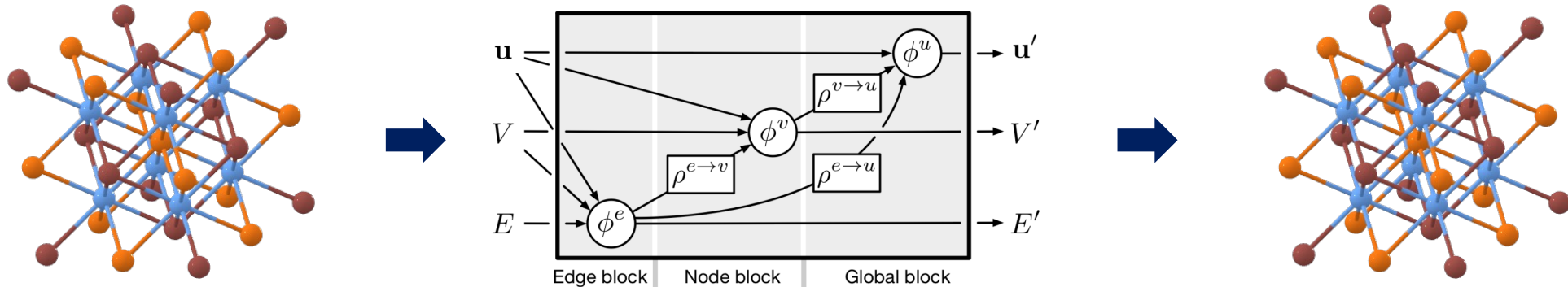
## ■ Causal inference-based

- Discovering invariant rationales for graph neural networks. ICLR 2022
- Debiasing Graph Neural Networks via Learning Disentangled Causal Substructure. NeurIPS 2022
- Causal attention for interpretable and generalizable graph classification. KDD 2022
- **Shift-robust molecular relational learning with causal substructure. KDD 2023**

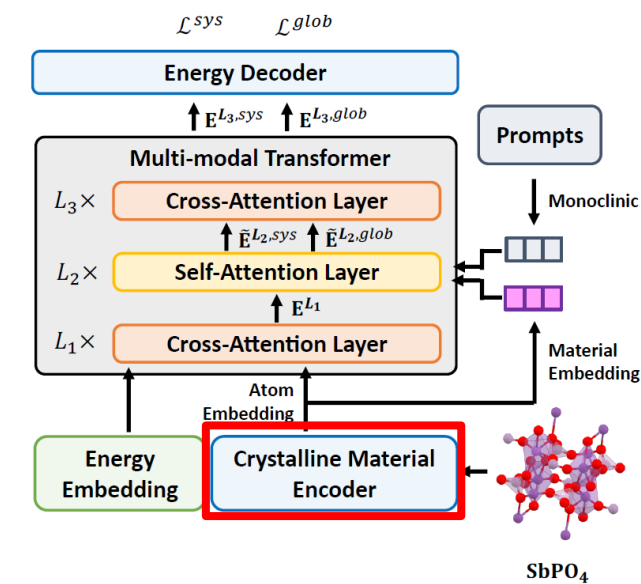
Thanks for listening!

# CRYSTAL ENCODER

- Graph Network: graph-to-graph function
  - Input: graph, Output: graph
    - Structure of input and output are equivalent
  - MLP is used to represent node/edge/graph of the output
  - Graph network can model the interaction between nodes
  - We can stack multiple blocks of graph network

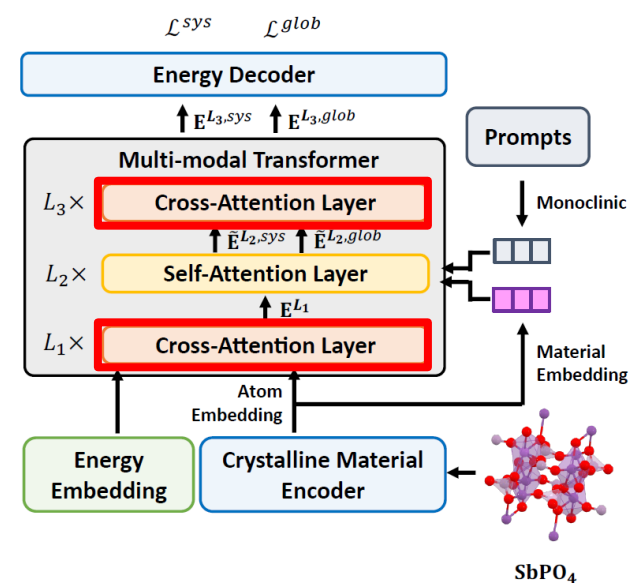
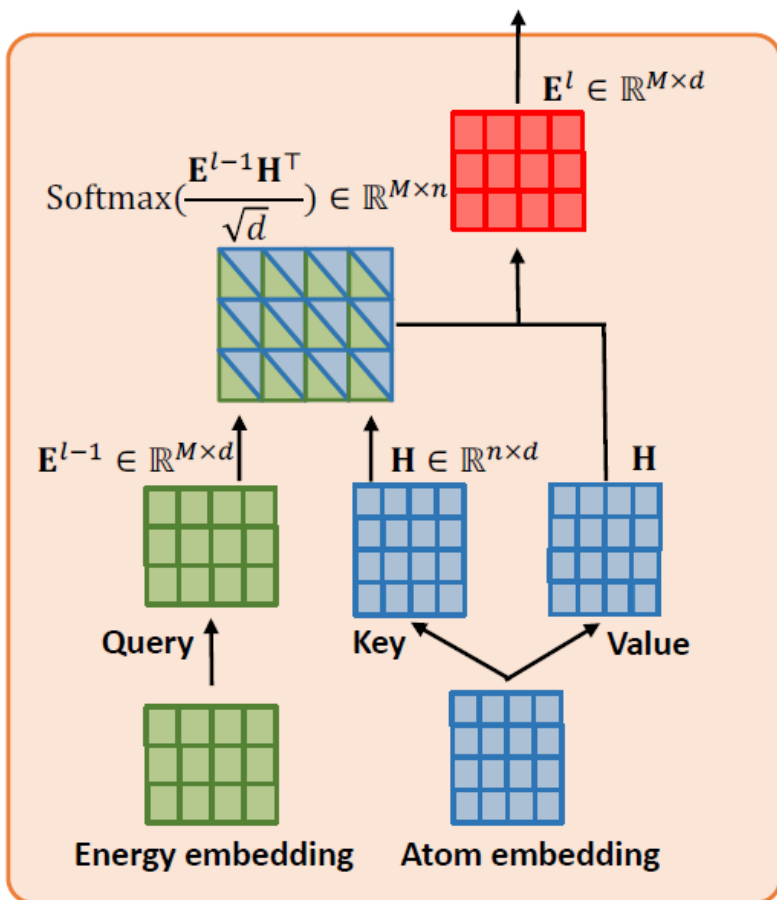


Architecture of Graph Network block



# PROMPT-GUIDED MULTI-MODAL TRANSFORMER

- Cross-Attention
- Obtain crystal-specific energy embedding  $E^l$

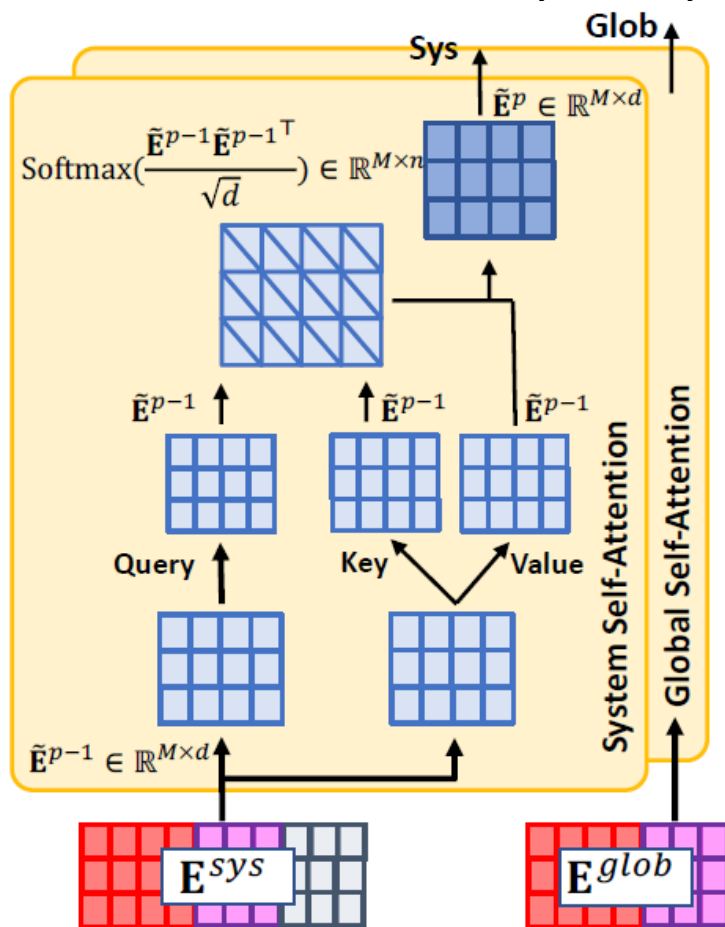


$$E^l = \text{Cross-Attention}(Q_{E^{l-1}}, K_H, V_H) \in \mathbb{R}^{M \times d}$$

$$= \text{Softmax}\left(\frac{E^{l-1} H^T}{\sqrt{d}}\right) H,$$

# PROMPT-GUIDED MULTI-MODAL TRANSFORMER

- Global Self-Attention
- System Self-Attention with Crystal System Prompts



Energy embedding

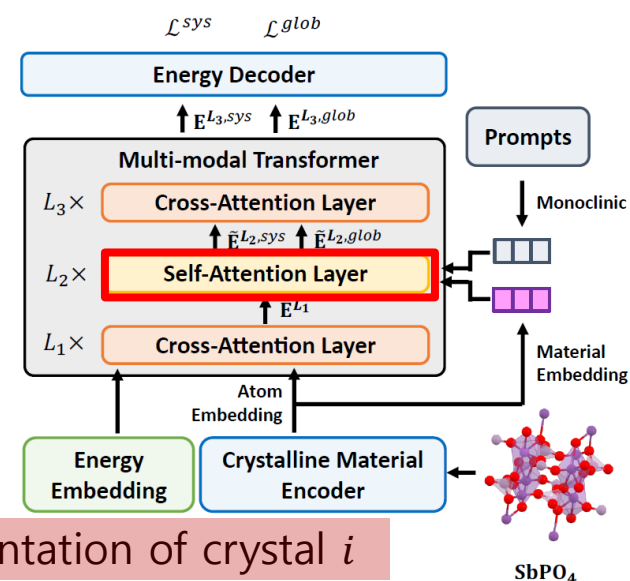
Sum-pooled representation of crystal  $i$

$$\mathbf{E}_j^{glob} = (\mathbf{E}_j^{L_1} || \mathbf{g}_i) \quad \tilde{\mathbf{E}}_j^0 = \phi_1(\mathbf{E}_j^{glob})$$

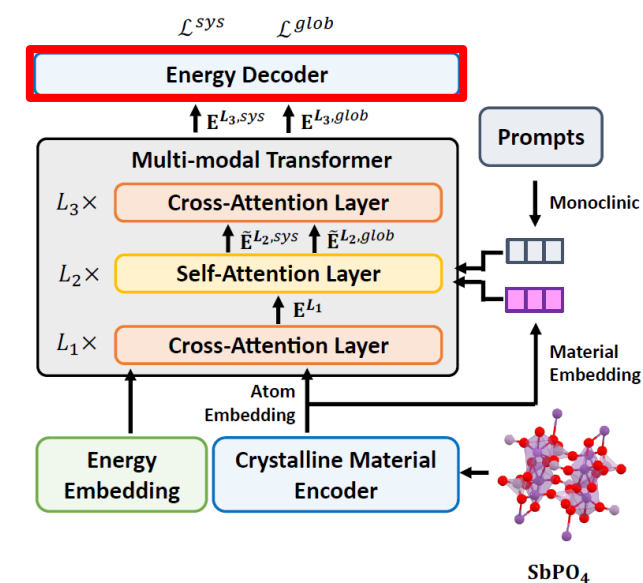
$$\mathbf{E}_j^{sys} = (\mathbf{E}_j^{L_1} || \mathbf{g}_i || \mathbf{P}_k) \quad \tilde{\mathbf{E}}_j^0 = \phi_2(\mathbf{E}_j^{sys})$$

Learnable prompts representing one of the 7 crystal systems

$$\begin{aligned} \tilde{\mathbf{E}}^p &= \text{Self-Attention}(\mathbf{Q}_{\tilde{\mathbf{E}}^{p-1}}, \mathbf{K}_{\tilde{\mathbf{E}}^{p-1}}, \mathbf{V}_{\tilde{\mathbf{E}}^{p-1}}) \in \mathbb{R}^{M \times d} \\ &= \text{Softmax}(\frac{\tilde{\mathbf{E}}^{p-1} \tilde{\mathbf{E}}^{p-1 \top}}{\sqrt{d}}) \tilde{\mathbf{E}}^{p-1}, \end{aligned}$$



# ENERGY DECODER



Crystal-specific energy embedding of crystal  $i$  at energy level  $j$

$$\hat{\mathbf{Y}}_j^i = \phi_{pred}(\mathbf{E}_j^{L_3,i})$$

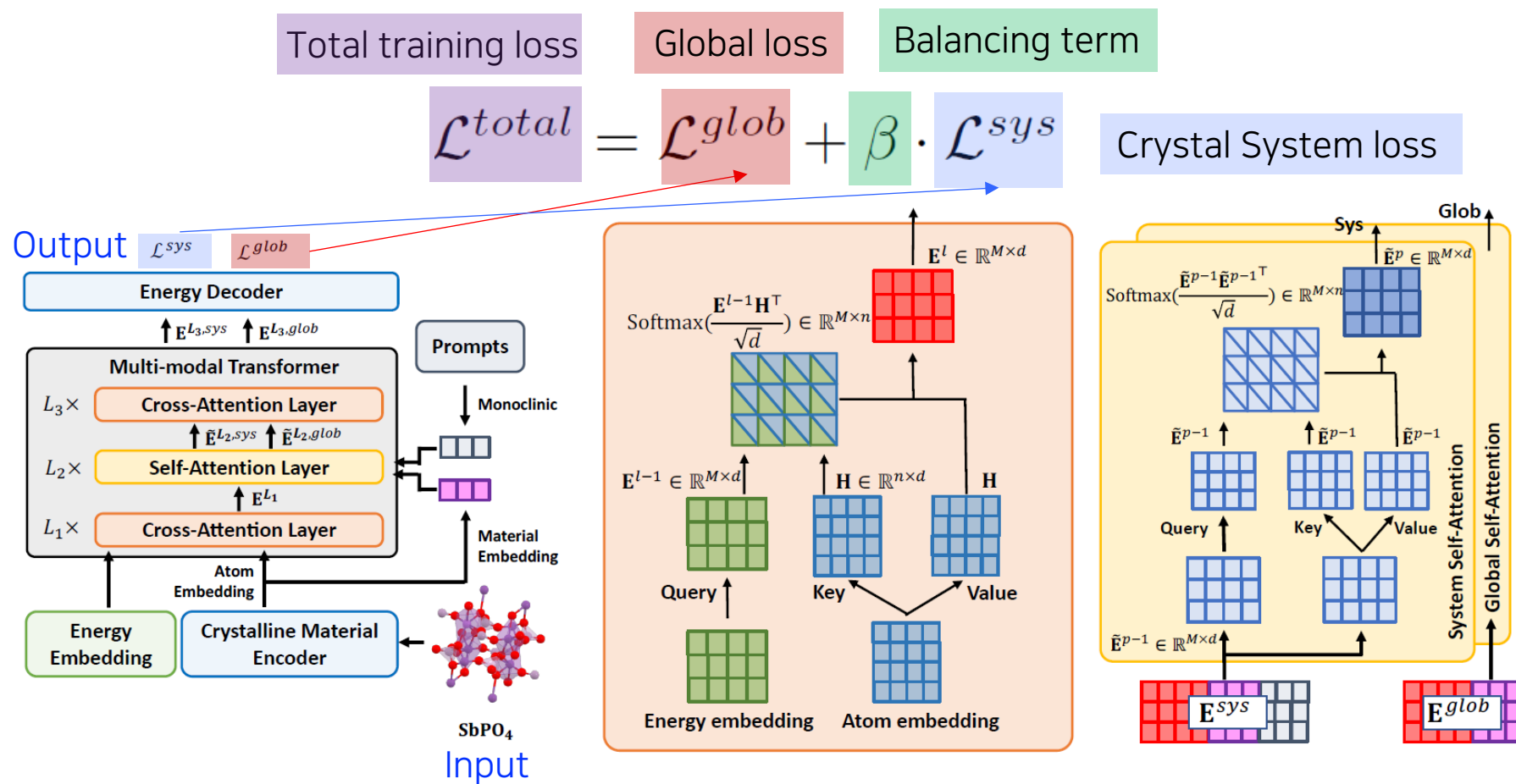
Predicted DOS of crystal  $i$  at energy level  $j$

MLP for predicting DOS

$$\phi_{pred} : \mathbb{R}^d \rightarrow \mathbb{R}^1$$

# Our proposed method: Prompt-guided DOSTransformer

- Using RMSE loss & 2 Forward Passes (System and Global energy embedding)



# Nonlinearity Encoding based on Wasserstein Distance

- For a set of probability measures  $\Pi$  on  $\Omega \times \Omega$ , Wasserstein distance is defined by an optimization problem as:

$$W_p = \left( \inf_{\pi \in \Pi} \int_{\Omega \times \Omega} \|\mathbf{x} - \mathbf{y}\|_p \pi(\mathbf{x}, \mathbf{y}) d\mathbf{x} d\mathbf{y} \right)^{1/p}$$

## Why Wasserstein distance?

Many scientific data has unknown and arbitrary shaped distributions

- However, there is a problem in applying Wasserstein distance in our task
  - Wasserstein distance is defined only for the **data distributions of the same dimensionality**.
- Our task:** Regression
  - Input: Vector ( $\in \mathbb{R}^d$ )
  - Target: Scalar ( $\in \mathbb{R}$ )

Dimension mismatch!



# Nonlinearity Encoding based on Wasserstein Distance

- Instead, we define **distance distribution** to apply Wasserstein distance between **two distributions of different dimensions**

*Definition)* For a  $n$ -dimensional space  $\mathcal{X} \subseteq \mathbb{R}^n$ , **distance distribution  $\mathcal{K}$  is defined as a probability distribution of pairwise distances  $d(x, x')$  for all  $(x, x') \in \mathcal{X} \times \mathcal{X}$ , where  $d: \mathcal{X} \times \mathcal{X} \rightarrow [0, \infty)$  is a distance metric.**

$$W_p = \left( \inf_{\pi \in \Pi} \int_{\Omega \times \Omega} \|\mathbf{x} - \mathbf{y}\|_p \pi(\mathbf{x}, \mathbf{y}) d\mathbf{x} d\mathbf{y} \right)^{1/p}$$

$(p = 1)$



**Distance consistency** btw input and target!

$$W_1(\mathcal{K}_x, \mathcal{K}_y; \pi, \theta) = \inf_{\pi \in \Pi} \int_{\mathcal{M} \times \mathcal{M}} \|r - u\| \pi(r, u) dr du$$

- $r = d(\phi(\mathbf{x}; \theta), \phi(\mathbf{x}'; \theta))$ : **Dist. btw input data** in embedding space
- $u = d(y, y')$ : **Dist. btw target data**

**Our goal: Maximize the distance consistency** between input and target

→ The distance between two inputs should be determined based on the distance between their targets

# Extrapolation on Time-Series Data: Geomagnetic Storm Forecasting

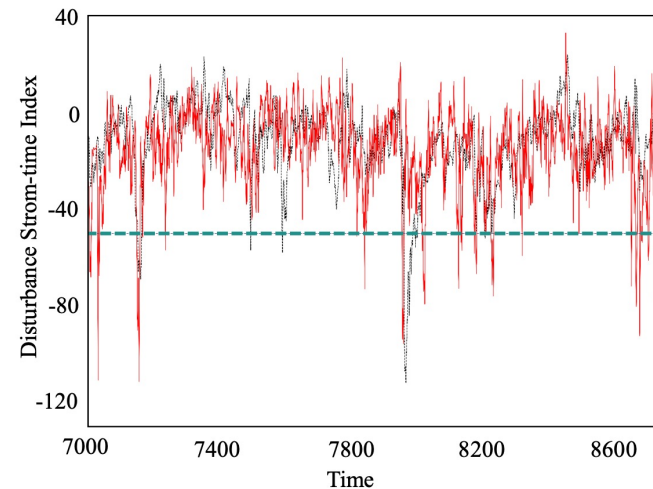
- **Task:** 1) Predict geomagnetic storm, 2) Detect geomagnetic storm
- **Data preprocessing**
  - Dataset: MagNet NASA dataset
  - 1-year geomagnetic storm data is divided into 4 sequential periods ( $\frac{3}{4}$  used for training,  $\frac{1}{4}$  used for test)

Task 1

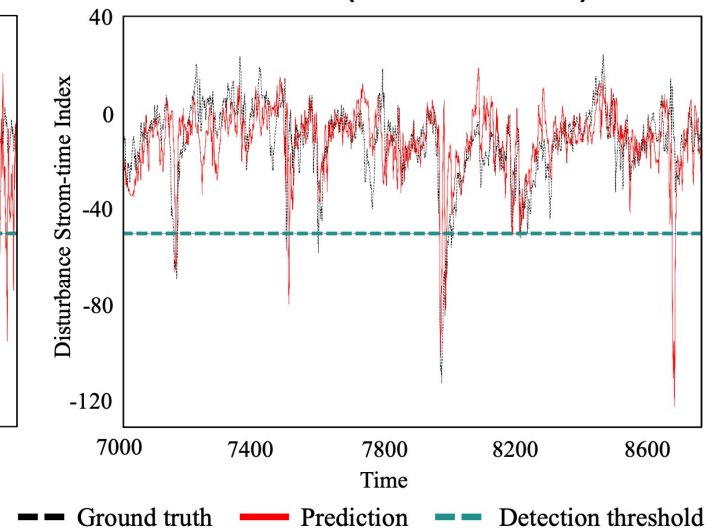
Task 2

Method	Extrapolation Error		Detection Accuracy		
	MAE	Corr	Precision	Recall	F1-score
RNN	16.089 ( $\pm 0.806$ )	0.710 ( $\pm 0.025$ )	0.133 ( $\pm 0.013$ )	0.281 ( $\pm 0.065$ )	0.178 ( $\pm 0.015$ )
LSTM	14.721 ( $\pm 0.702$ )	0.696 ( $\pm 0.065$ )	0.164 ( $\pm 0.048$ )	0.260 ( $\pm 0.087$ )	0.201 ( $\pm 0.062$ )
GRU	14.613 ( $\pm 0.368$ )	0.687 ( $\pm 0.027$ )	0.145 ( $\pm 0.027$ )	0.230 ( $\pm 0.055$ )	0.177 ( $\pm 0.034$ )
TF	13.106 ( $\pm 0.717$ )	0.670 ( $\pm 0.031$ )	0.185 ( $\pm 0.115$ )	0.145 ( $\pm 0.074$ )	0.159 ( $\pm 0.084$ )
LRL-GRU	13.700 ( $\pm 0.581$ )	0.499 ( $\pm 0.031$ )	0.189 ( $\pm 0.035$ )	0.519 ( $\pm 0.186$ )	0.272 ( $\pm 0.054$ )
SLRL-GRU	10.986 ( $\pm 0.332$ )	0.455 ( $\pm 0.040$ )	0.260 ( $\pm 0.065$ )	0.336 ( $\pm 0.111$ )	0.291 ( $\pm 0.077$ )
<b>ANE-GRU</b>	<b>10.534</b> <b>(<math>\pm 0.407</math>)</b>	<b>0.428</b> <b>(<math>\pm 0.041</math>)</b>	<b>0.513</b> <b>(<math>\pm 0.044</math>)</b>	<b>0.495</b> <b>(<math>\pm 0.071</math>)</b>	<b>0.502</b> <b>(<math>\pm 0.042</math>)</b>

Vanilla GRU



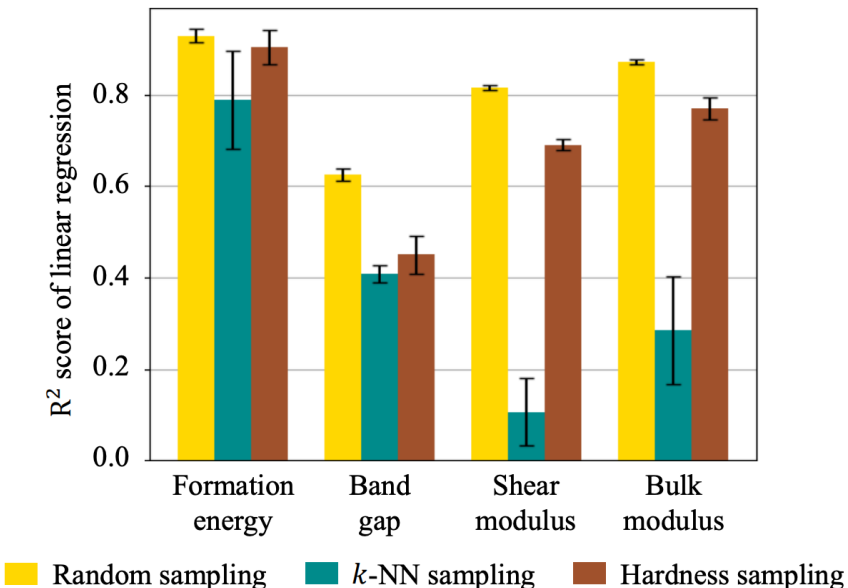
Ours (ANE-GRU)



ANE-GRU outperforms GRU, and ANE achieved further improvement over metric learning-based approaches

# Sampling Strategies and Extrapolation

- Time complexity of the training process of ANE:  $\theta^* = \operatorname{argmin}_{\theta} \sum_{i=1}^N \sum_{j=1}^N \|r_{ij} - u_{ij}\| \rightarrow \mathbf{O}(N^2)$
- Three sampling strategies to reduce the time complexity:
  - **Random sampling:** selecting a data point randomly at each iteration
  - **$k$ -NN sampling:** selecting  $k$  nearest data points for an anchor data
  - **Hardness sampling:** selecting  $k$  data points based on the training errors (top- $k$  largest errors)

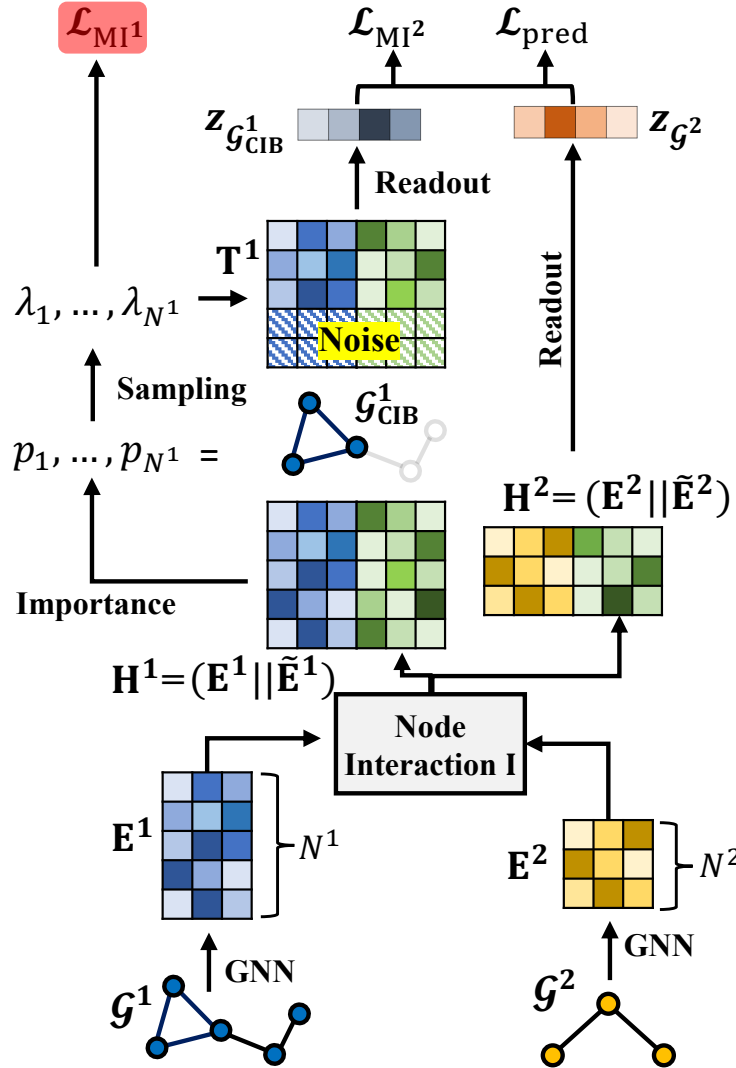


**Random sampling performs the best despite its simplicity**  
( $\because$  Random sampling = Density-based sampling)

$$\min -I(Y; \mathcal{G}_{\text{CIB}}^1 | \mathcal{G}^2) + \beta I(\mathcal{G}^1; \mathcal{G}_{\text{CIB}}^1 | \mathcal{G}^2)$$

# Proposed Method: Conditional Graph Information Bottleneck

$$I(\mathcal{G}^1; \mathcal{G}_{\text{CIB}}^1 | \mathcal{G}^2) = \underbrace{I(\mathcal{G}_{\text{CIB}}^1; \mathcal{G}^1, \mathcal{G}^2)}_{\text{Chain rule of mutual information}} - I(\mathcal{G}_{\text{CIB}}^1; \mathcal{G}^2)$$



Upper bound of  $I(\mathcal{G}_{\text{CIB}}^1; \mathcal{G}^1, \mathcal{G}^2)$

$$I(\mathcal{G}_{\text{CIB}}^1; \mathcal{G}^1, \mathcal{G}^2) \leq \mathbb{E}_{\mathcal{G}^1, \mathcal{G}^2} \left[ -\frac{1}{2} \log A + \frac{1}{2N^1} A + \frac{1}{2N^1} B^2 \right] \quad \text{where } A = \sum_{j=1}^{N^1} (1 - \lambda_j)^2 \text{ and } B = \frac{\sum_{j=1}^{N^1} \lambda_j (H_j^1 - \mu_{H^1})^2}{\sigma_{H^1}}$$

$$:= \mathcal{L}_{MI^1}(\mathcal{G}_{\text{CIB}}^1, \mathcal{G}^1, \mathcal{G}^2)$$

**Proof.** Given the perturbed graph  $\mathcal{G}_{\text{CIB}}^1$  and its representation  $z_{\mathcal{G}_{\text{CIB}}^1}$ , we assume there is no information loss during the readout process, i.e.,  $I(z_{\mathcal{G}_{\text{CIB}}^1}; \mathcal{G}^1, \mathcal{G}^2) = I(\mathcal{G}_{\text{CIB}}^1; \mathcal{G}^1, \mathcal{G}^2)$ .

$$\begin{aligned} I(z_{\mathcal{G}_{\text{CIB}}^1}; \mathcal{G}^1, \mathcal{G}^2) &= \mathbb{E}_{z_{\mathcal{G}_{\text{CIB}}^1}, \mathcal{G}^1, \mathcal{G}^2} \left[ -\log \frac{p_{\phi}(z_{\mathcal{G}_{\text{CIB}}^1} | \mathcal{G}^1, \mathcal{G}^2)}{p(z_{\mathcal{G}_{\text{CIB}}^1})} \right] \\ &= \mathbb{E}_{\mathcal{G}^1, \mathcal{G}^2} \left[ -\log \frac{p_{\phi}(z_{\mathcal{G}_{\text{CIB}}^1} | \mathcal{G}^1, \mathcal{G}^2)}{q(z_{\mathcal{G}_{\text{CIB}}^1})} \right] - \mathbb{E}_{z_{\mathcal{G}_{\text{CIB}}^1}, \mathcal{G}^1, \mathcal{G}^2} [KL(p(z_{\mathcal{G}_{\text{CIB}}^1}) || q(z_{\mathcal{G}_{\text{CIB}}^1}))] \\ &\leq \mathbb{E}_{z_{\mathcal{G}_{\text{CIB}}^1}, \mathcal{G}^1, \mathcal{G}^2} [KL(p_{\phi}(z_{\mathcal{G}_{\text{CIB}}^1} | \mathcal{G}^1, \mathcal{G}^2) || q(z_{\mathcal{G}_{\text{CIB}}^1}))] \quad (1) \quad \because \text{Non-negativity of KL divergence} \end{aligned}$$

Assuming that  $q(z_{\mathcal{G}_{\text{CIB}}^1})$  is Gaussian distribution.

The noise  $\varepsilon \sim N(\mu_{H^1}, \sigma_{H^1})$  is sampled from Gaussian distribution where  $\mu_{H^1}$  and  $\sigma_{H^1}$  are mean and variance of  $H^1$ .

Thus,  $q(z_{\mathcal{G}_{\text{CIB}}^1}) = N(N^1 \mu_{H^1}, N^1 \sigma_{H^1})$  (2)  $\because$  Summation of Gaussian is Gaussian

And,  $p(z_{\mathcal{G}_{\text{CIB}}^1} | \mathcal{G}^1, \mathcal{G}^2) = N(N^1 \mu_{H^1} + \sum_{j=1}^{N^1} \lambda_j H_j^1 - \sum_{j=1}^{N^1} \lambda_j \mu_{H^1}, \sum_{j=1}^{N^1} (1 - \lambda_j)^2 \sigma_{H^1}^2)$  (3)

By plugging Equation (2) and (3) into (1), we have:

$$-I(\mathcal{G}_{\text{CIB}}^1; \mathcal{G}^1, \mathcal{G}^2) \leq \mathbb{E}_{\mathcal{G}^1, \mathcal{G}^2} \left[ -\frac{1}{2} \log A + \frac{1}{2N^1} A + \frac{1}{2N^1} B^2 \right] + C \quad \text{where } A = \sum_{j=1}^{N^1} (1 - \lambda_j)^2 \text{ and } B = \frac{\sum_{j=1}^{N^1} \lambda_j (H_j^1 - \mu_{H^1})^2}{\sigma_{H^1}}$$