그래프 신경망 기반 화학/소재분야 연구 동향

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
- http://dsail.kaist.ac.kr/

Research Interest

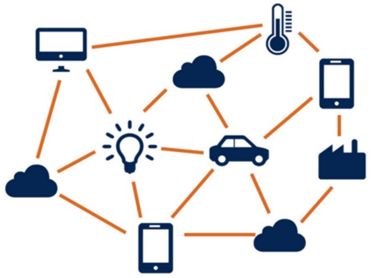
- **Multimodal Data Mining, Applied Machine Learning, Deep Learning**
  - *Mining meaningful knowledge from multimodal data to develop artificial intelligence solutions for various real-world applications across different disciplines*
  - Keywords: Multimodal user behavior analysis, Machine learning 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

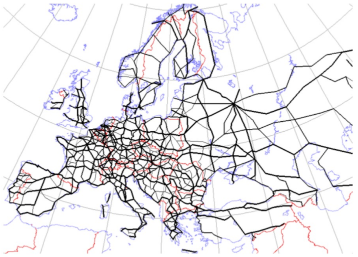
- Assistant Professor, **KAIST** (2020.11 – Present)
- Postdoctoral Research Fellow, **University of Illinois at Urbana-Champaign**, Dept. of Computer Science (2019. 1 – 2020. 10)
- Research Intern, **Microsoft Research Asia** (2017. 9 – 2017. 12)
- Research Intern, **NAVER** (2017. 3 – 2017. 6)

Research area Graphs are everywhere!

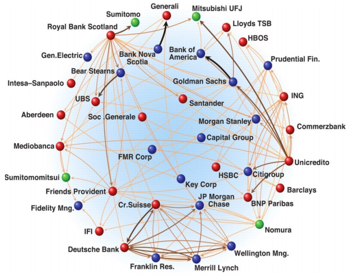
Social Network



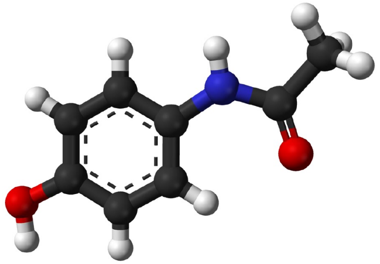
Internet of Things



Road Graph



Financial Graph

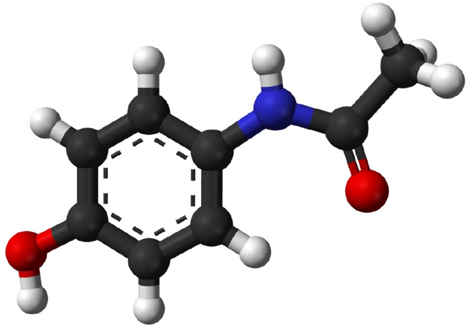


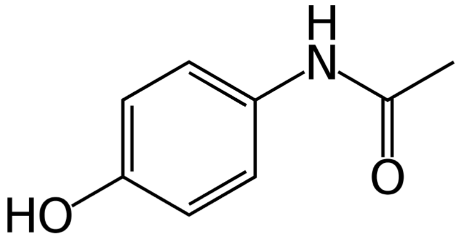
Molecular Graph

- 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**

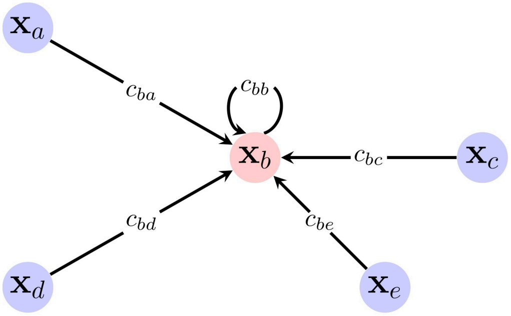
Introduction



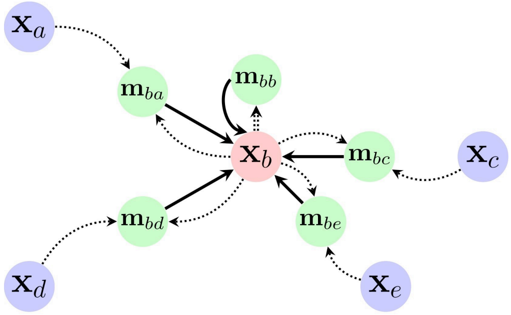


- A molecule can be represented as a graph
  - Atom in a molecule: Node in a graph
  - Bond in a molecule: Edge in a graph
- **Graph machine learning is widely being applied to chemistry / materials science**

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



Convolutional



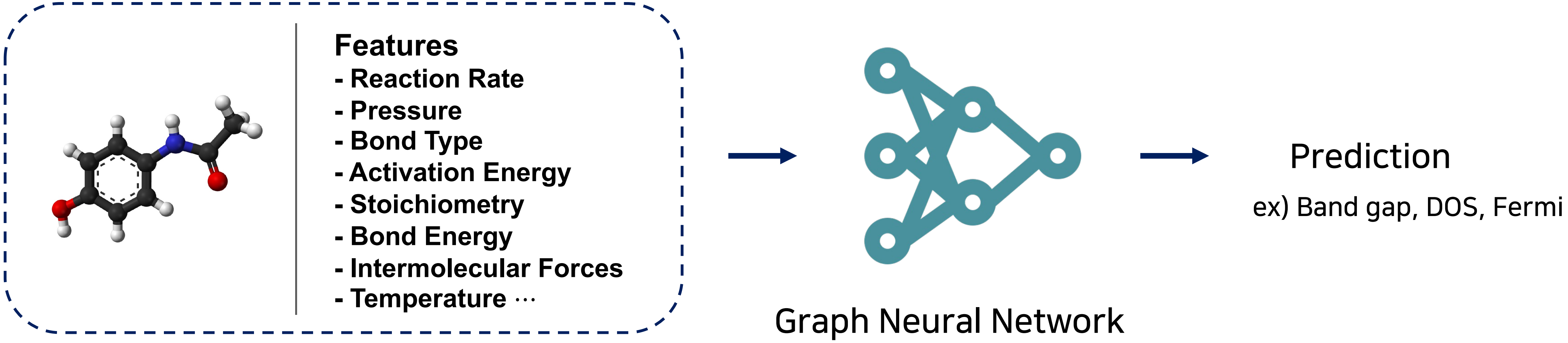
Message-passing

Table of Contents

시간	주제	주요 내용
1	그래프 신경망 개요	- 그래프 신경망 전반적인 소개 - 그래프 종류에 따른 다양한 그래프 신경망 소개
2	소재 물성 예측 연구	- 소재 물성 예측 연구 최신 동향 소개 - Transformer 기반 모델 소개 및 extrapolation을 위한 모델 소개
3	물질 간 화학 반응 예측 연구	- 물질 간 화학 반응 예측 연구 동향 소개 - 정보 이론(Information bottleneck) 및 인과추론(Causal inference) 기반 모델 소개

Introduction: Molecular Property Prediction

- Predict the properties of a molecule



Features

- **- Reaction Rate**
- **- Pressure**
- **- Bond Type**
- **- Activation Energy**
- **- Stoichiometry**
- **- Bond Energy**
- **- Intermolecular Forces**
- **- Temperature** ···

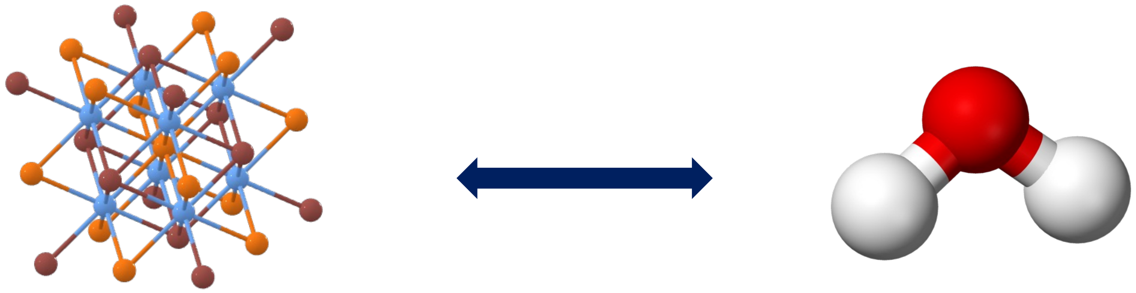
Graph Neural Network

Prediction

ex) Band gap, DOS, Fermi

Introduction: 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)

Table of Contents

시간	주제	주요 내용
1	그래프 신경망 개요	- 그래프 신경망 전반적인 소개 - 그래프 종류에 따른 다양한 그래프 신경망 소개
2	소재 물성 예측 연구	- 소재 물성 예측 연구 최신 동향 소개 - Transformer 기반 모델 소개 및 extrapolation을 위한 모델 소개
3	물질 간 화학 반응 예측 연구	- 물질 간 화학 반응 예측 연구 동향 소개 - 정보 이론(Information bottleneck) 및 인과추론(Causal inference) 기반 모델 소개

그래프 신경망 개요

14:00-15:00

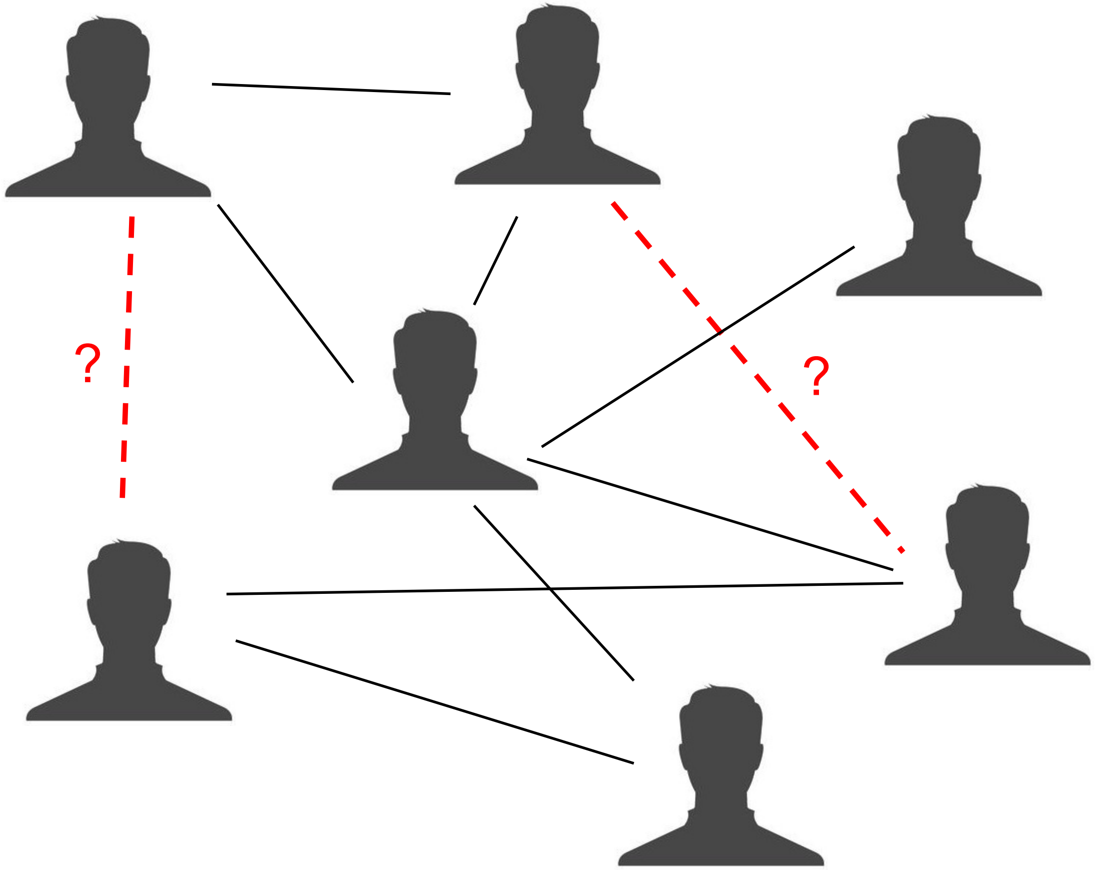
Outline

- Overview
- Node-level Network Embedding
  - Random walk-based Node Embeddings (DeepWalk / node2vec)
- Graph Neural Network (GNN)
  - Graph Convolutional Neural Network (GCN)
  - Graph Attention Network (GAT)
  - Relational GCN
  - GraphSAGE
  - Deep Graph Infomax (DGI)
  - GNNs with Edge Embeddings

Machine learning on graphs

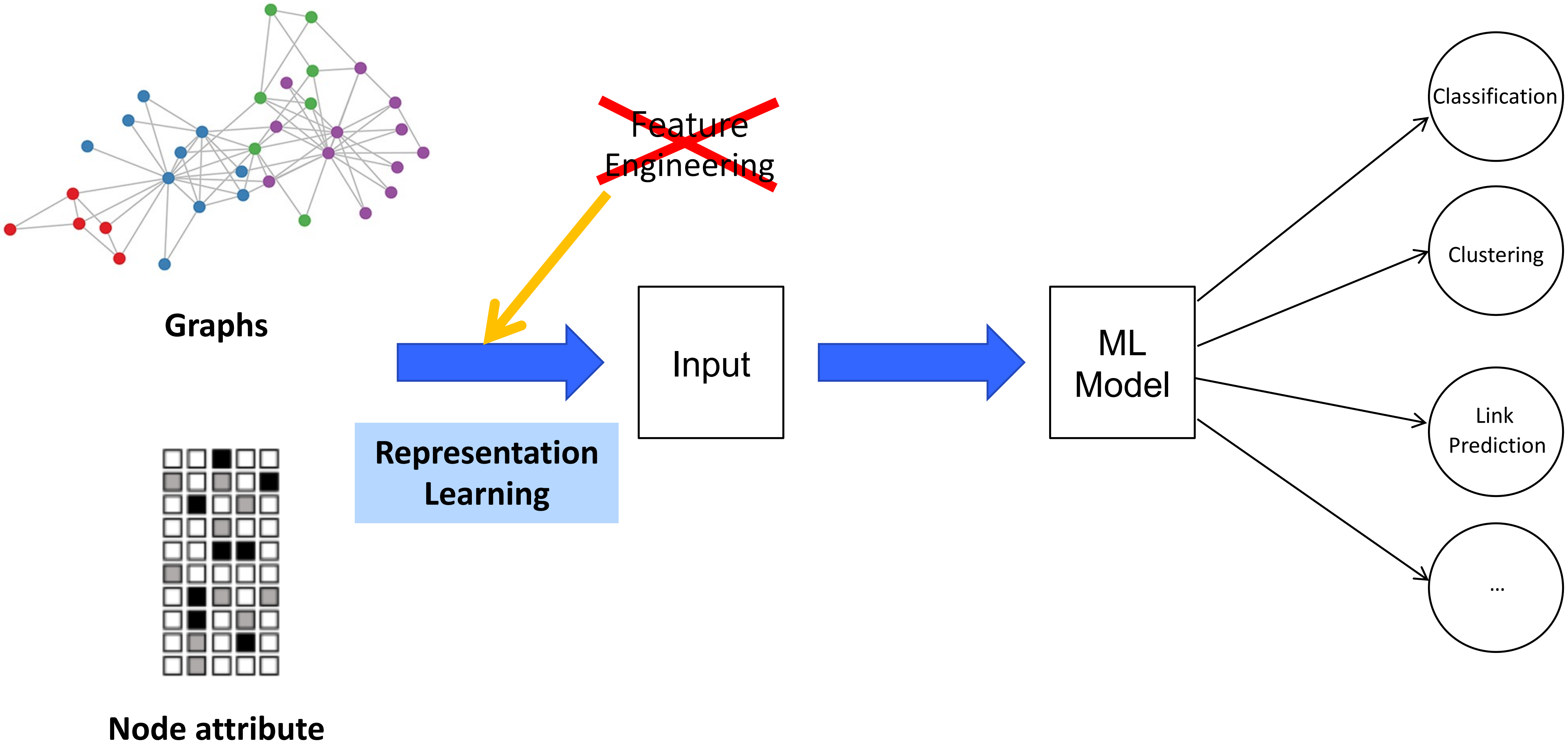
Classical ML tasks in graphs:

- 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



Graphs

Node attribute

~~Feature Engineering~~

Representation Learning

Input

ML Model

Classification

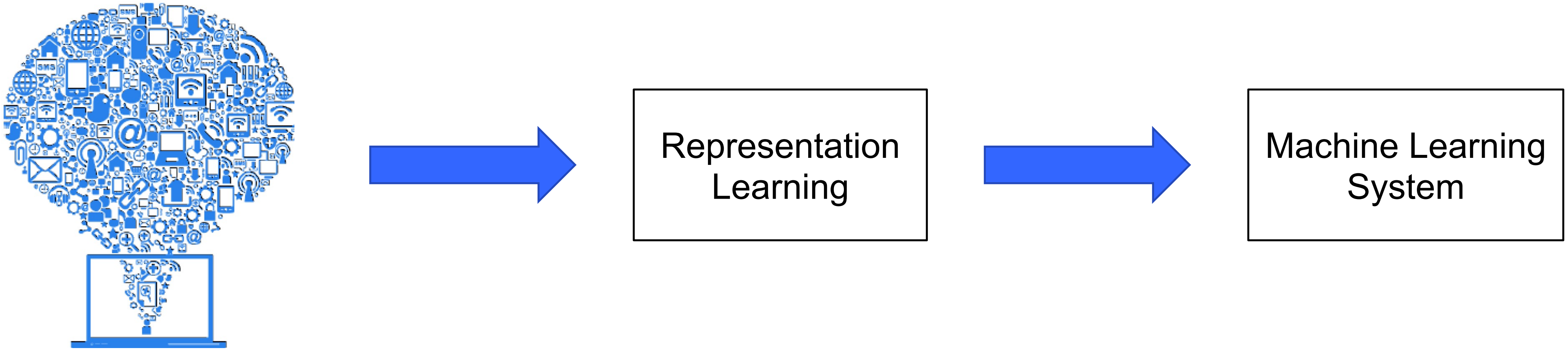
Clustering

Link Prediction

…

Machine learning in general

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



Raw data

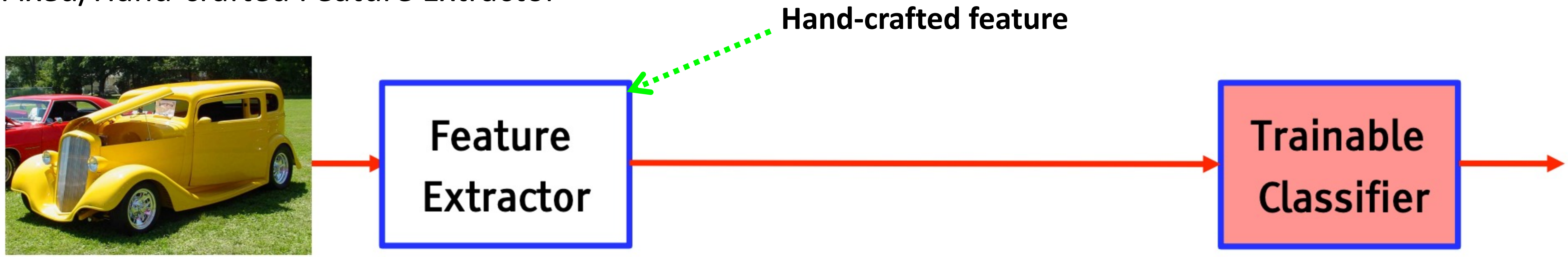
Representation Learning

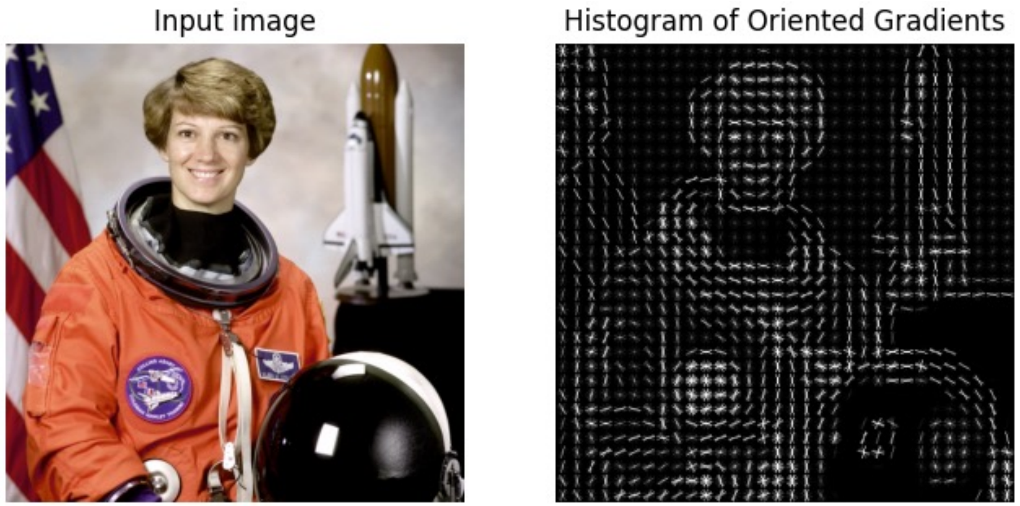
Machine Learning System

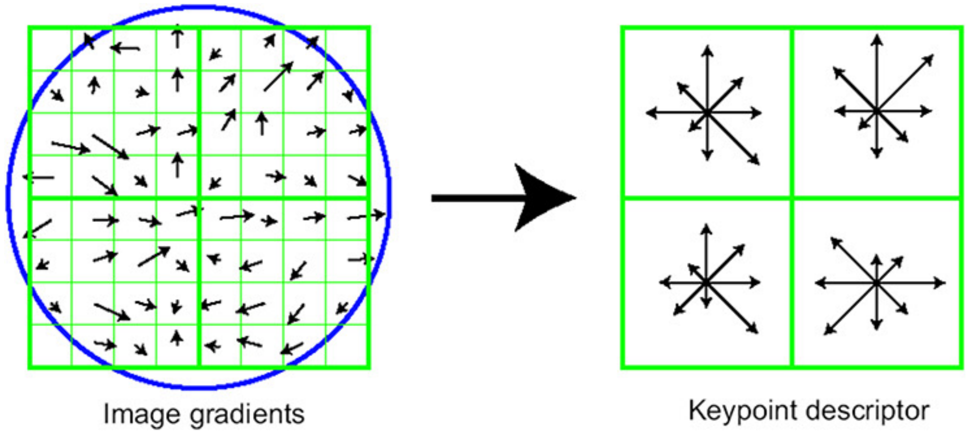
Good **Representation** is Essential for
Good **Machine Learning**

Traditional feature extraction for images

- Fixed/Hand-crafted Feature Extractor



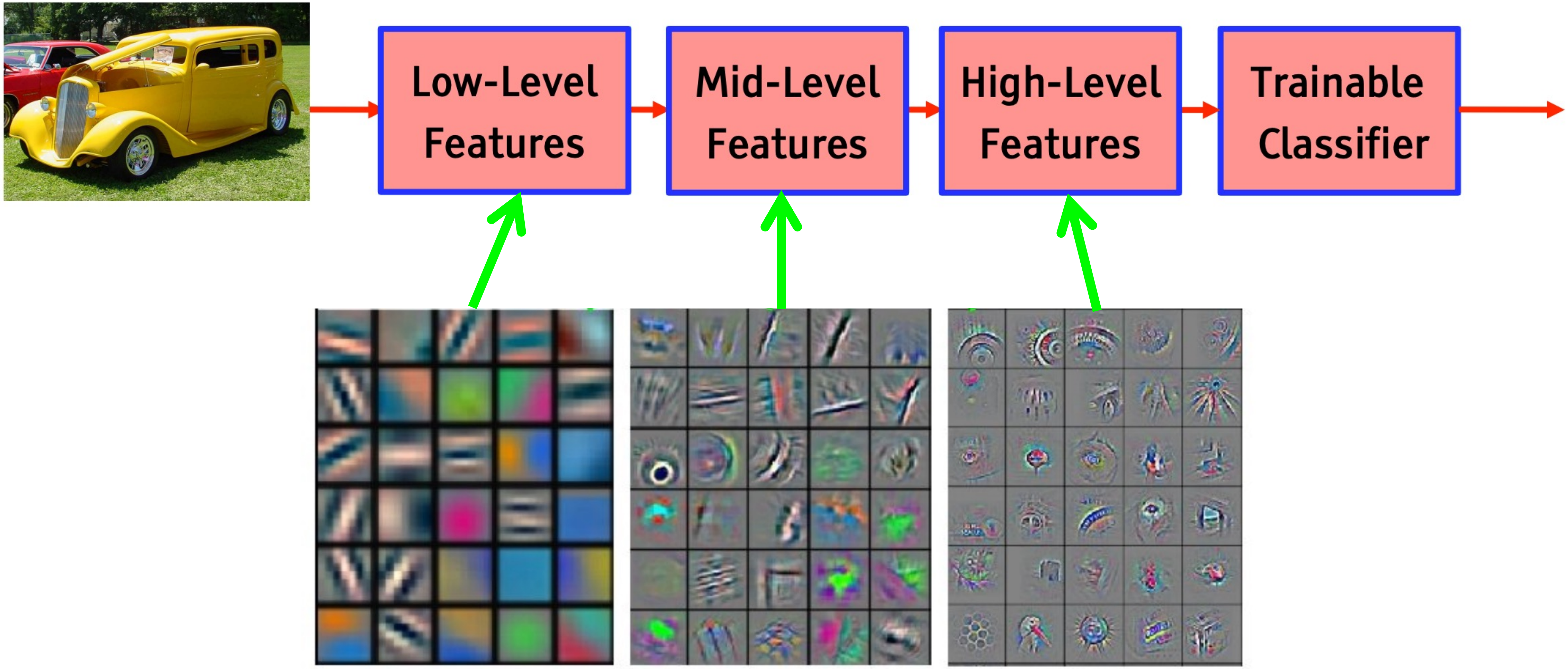




- Based on Yann Lecun's slides
- Lowe, David G. "Distinctive image features from scale-invariant keypoints." *International journal of computer vision* 60.2 (2004): 91-110.

Machine (Deep) learning based representation learning

- Multiple layers trained end-to-end



Low-Level Features

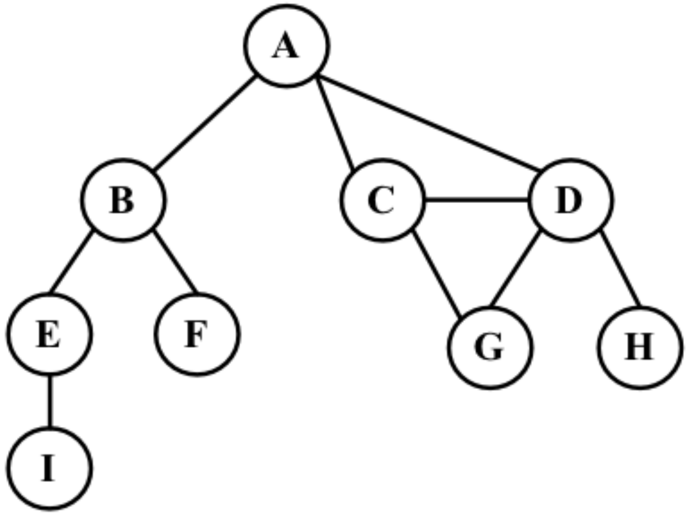
Mid-Level Features

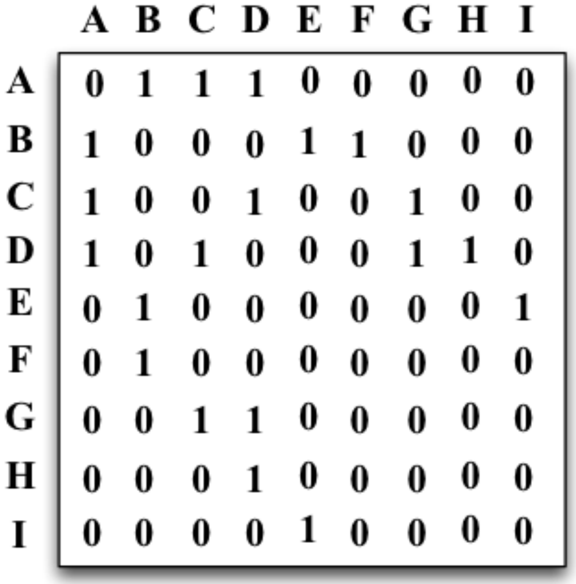
High-Level Features

Trainable Classifier

Based on Yann Lecun’s slides

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”
- …

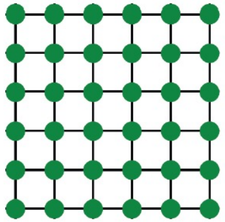
How to effectively and efficiently represent graphs is the key!

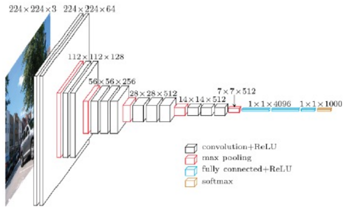
→ Deep learning-based approach?

(Figure credit) https://www.oreilly.com/library/view/learning-javascript-data/9781788623872/8a7d3187-7c57-418c-a426-3aceab96f47f.xhtml

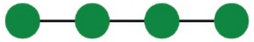
Challenges of graph representation learning

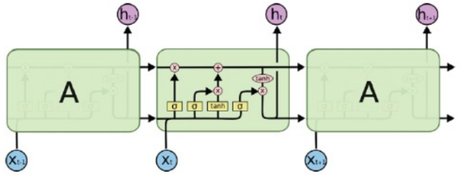
- Existing deep neural networks are designed for data with regular-structure (grid or sequence)
  - CNNs for fixed-size images/grids …





  - RNNs for text/sequences …



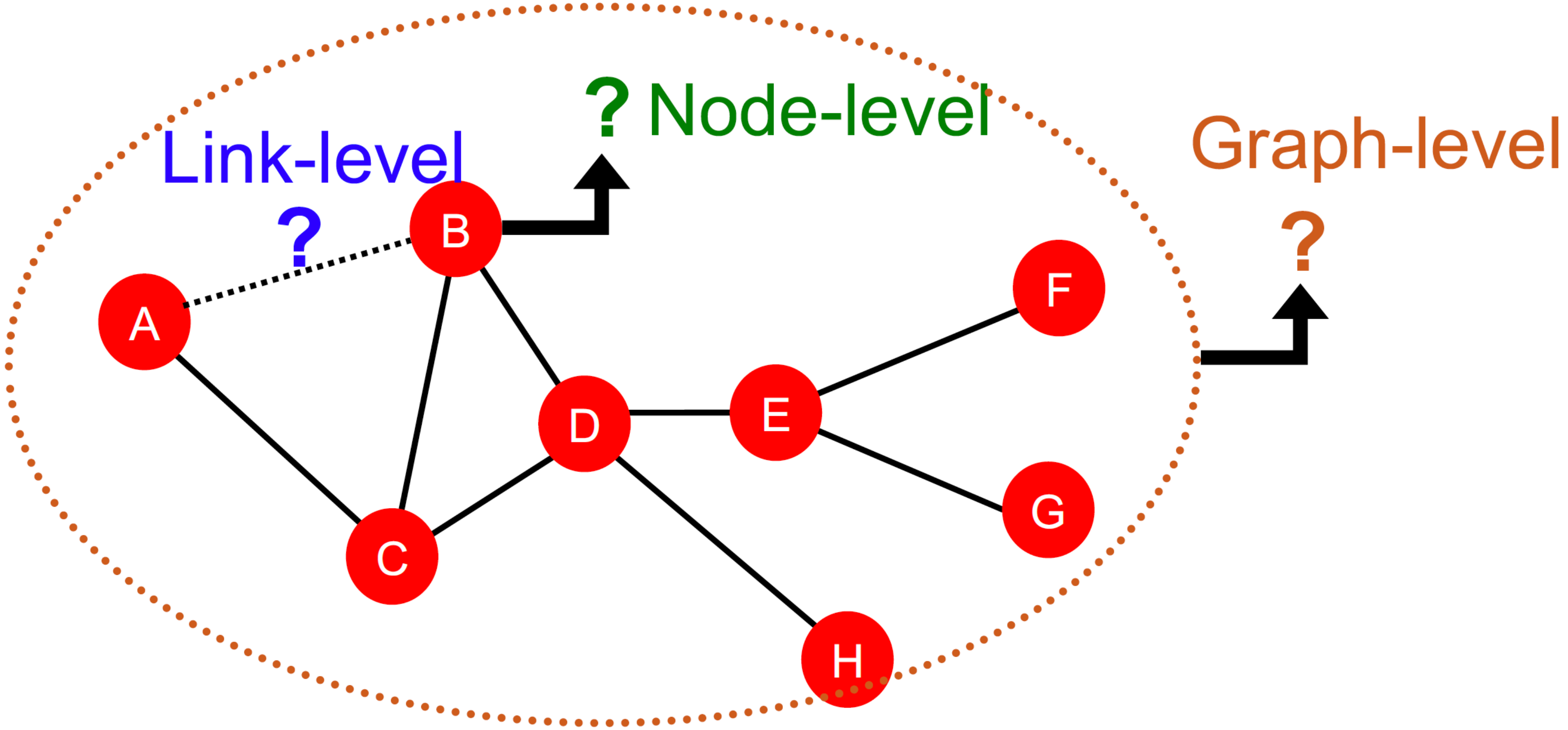


- **Graphs are very complex**
  - Arbitrary structures (no spatial locality like grids / no fixed orderings)
  - Heterogeneous: Directed/undirected, binary/weighted/typed, multimodal features
  - Large-scale: More than millions of nodes and billions of edges

(Figure credit) Tutorial on Graph Representation Learning, AAAI 2019

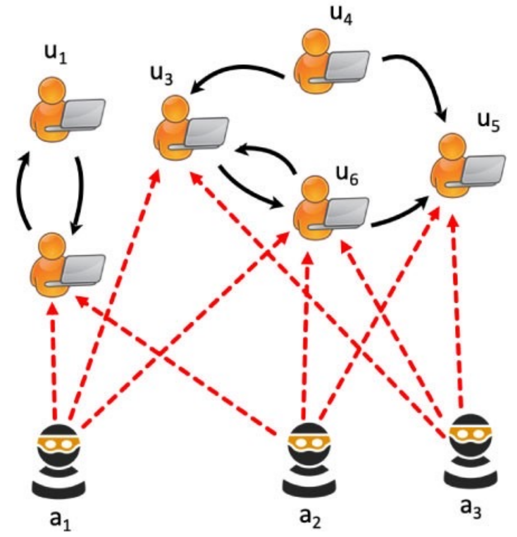
Typical tasks

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

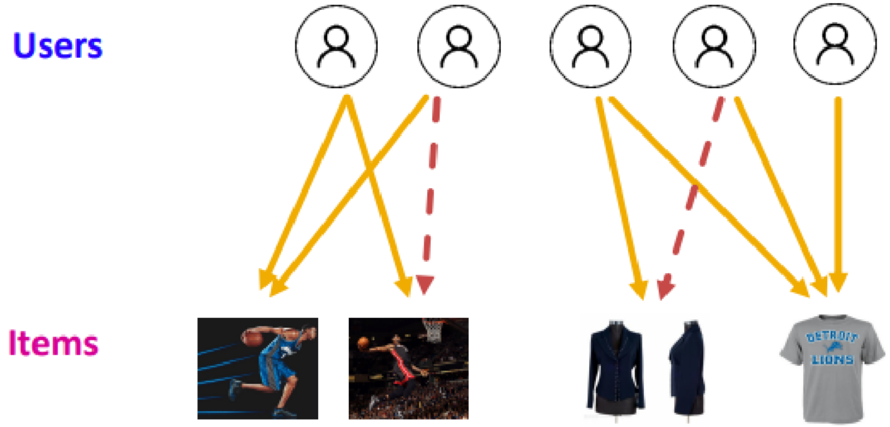


Typical tasks

- **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

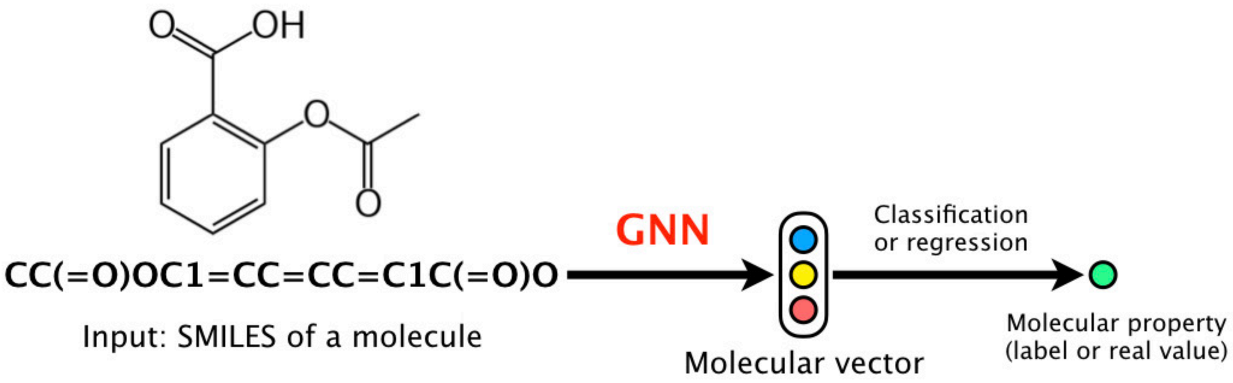


- Social network analysis (e.g., demographic info prediction)
- Spam / fraud detection (e.g., transaction networks)
- Link prediction (e.g., social networks, chemical interaction networks, biological networks, transportation networks)
- Knowledge graph population / completion / relation reasoning
- Recommender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



Typical tasks

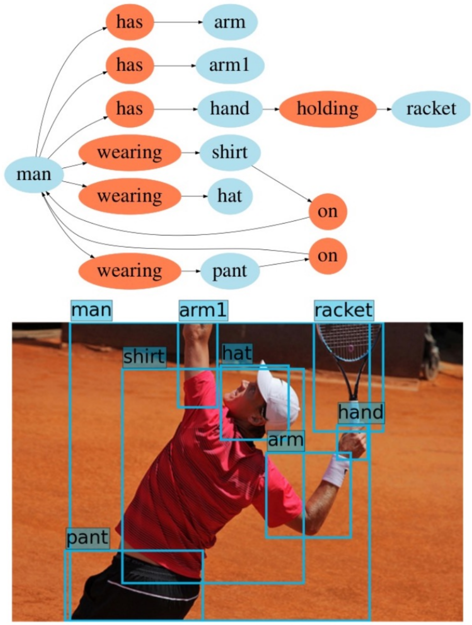
- **Graph-level tasks**
  - Graph label classification
  - Graph label regression



➢ Molecular property prediction

➢ Drug discovery

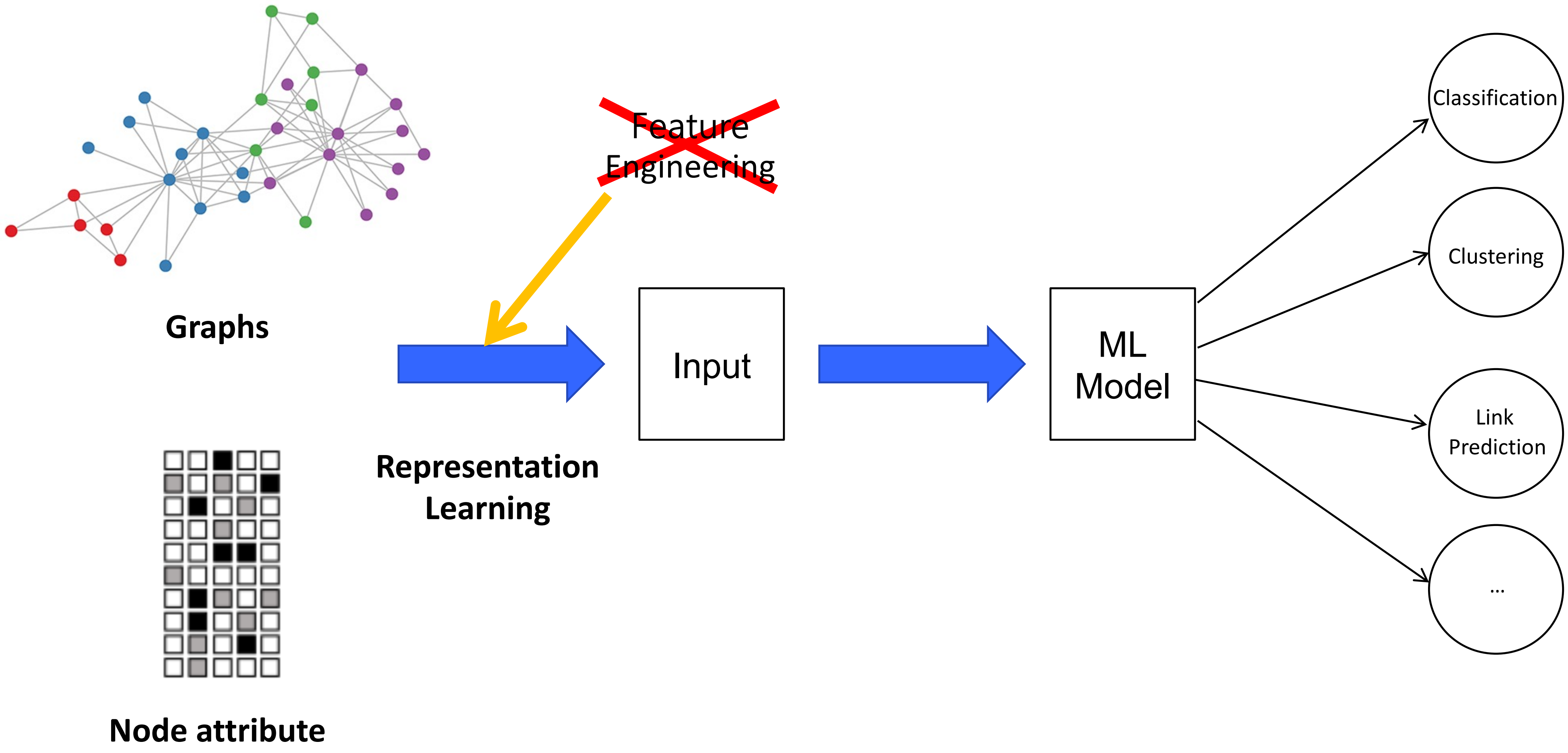
➢ Scene understanding (i.e., objects graph)



Outline

- Overview
- Node-level Network Embedding
  - Random walk-based Node Embeddings (DeepWalk / node2vec)
- Graph Neural Network (GNN)
  - Graph Convolutional Neural Network (GCN)
  - Graph Attention Network (GAT)
  - Relational GCN
  - GraphSAGE
  - Deep Graph Infomax (DGI)
  - GNNs with Edge Embeddings

Recall: Machine learning on graphs



Graphs

Node attribute

~~Feature Engineering~~

Representation Learning

Input

ML Model

Classification

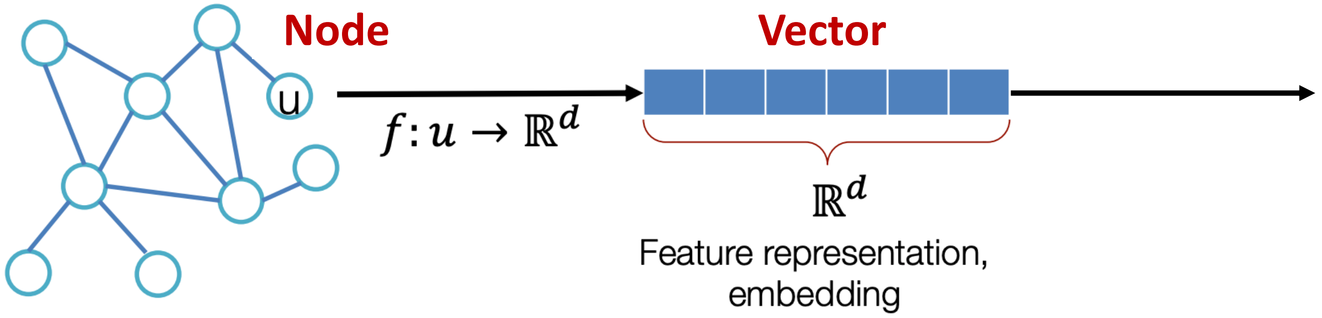
Clustering

Link Prediction

…

Graph representation learning

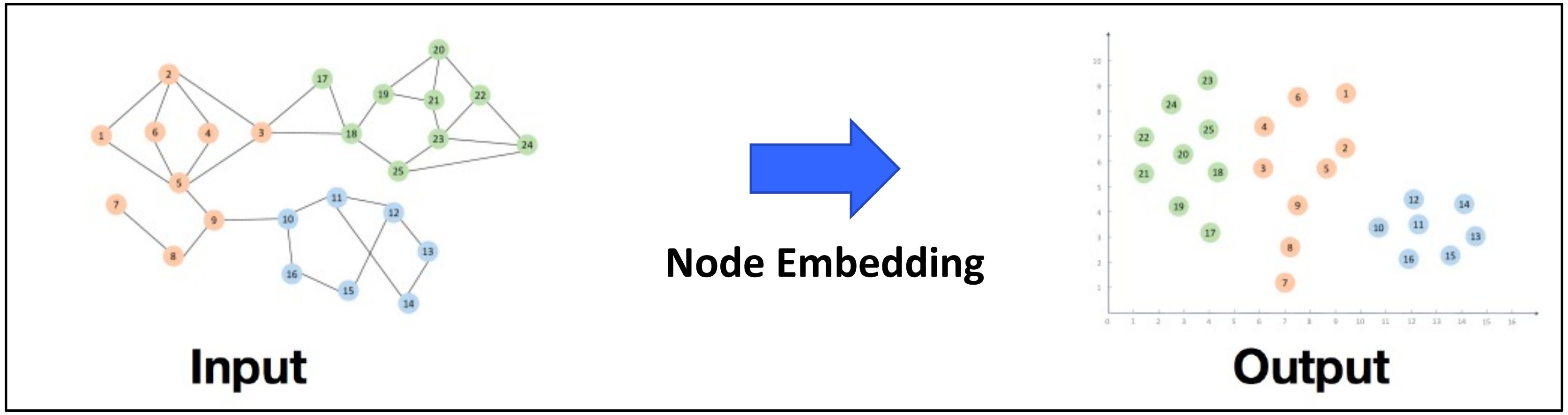
- **Goal**: Encode nodes so that **similarity in the embedding space** approximates **similarity in the original network**
- **Similar nodes in a network have similar vector representations**



Feature representation, embedding

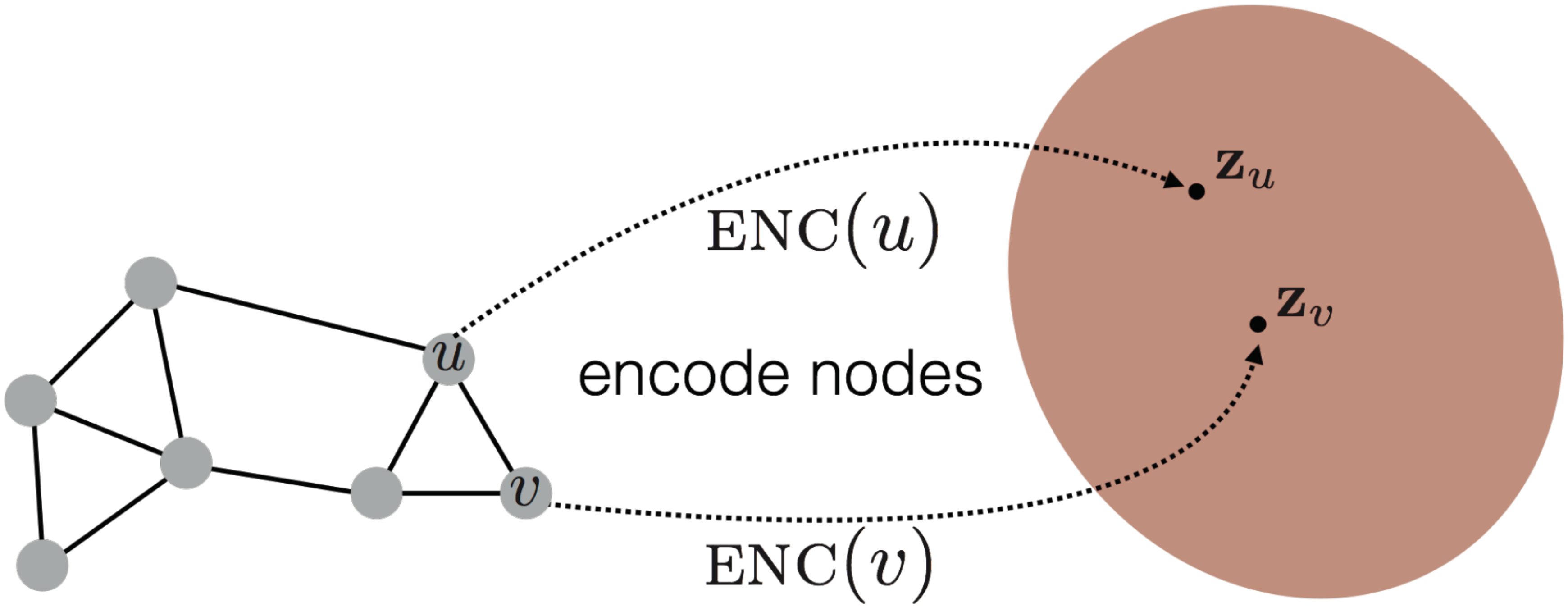
Tasks

- Node classification
- Clustering
- Link prediction
- …



Node embedding

- **Main idea:** Encode nodes so that **similarity in the embedding space** approximates **similarity in the graph**



Original graph

Embedding space
(Latent space)

- **Two things to consider**
  - 1. How to encode nodes?
    - Encoder
  - 2. How to define similarity in the embedding space?
    - Decoder (Similarity function)

Figures from http://web.stanford.edu/class/cs224w/

Encoder

- Maps each node to a low-dimensional vector

$$ENC(v) = \mathbf{z}_v$$

d-dimensional embedding vector

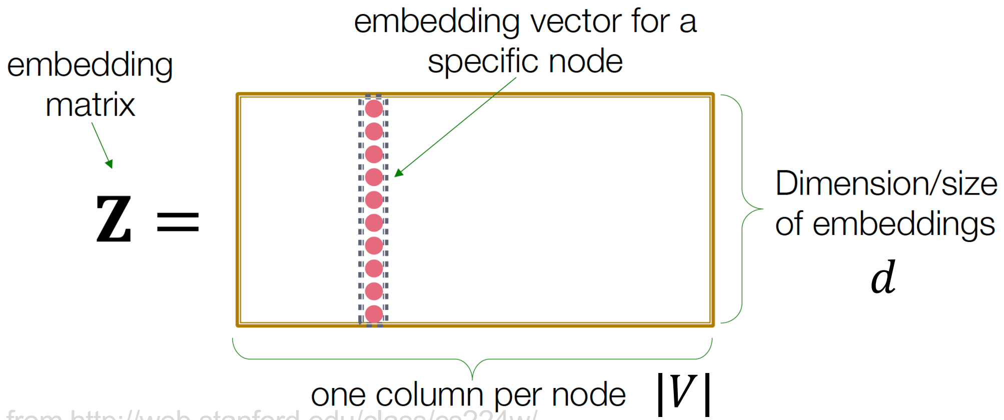
Node in the input graph

- **Simplest encoding approach**: Encoder is just an embedding-lookup (Shallow model)

$$ENC(v) = \mathbf{z}_v = \boldsymbol{Z} \cdot v$$

$$\boldsymbol{Z} \in R^{d\times|V|}$$

$$v \in I^{|V|}$$


embedding matrix
$\mathbf{Z} =$
embedding vector for a specific node
Dimension/size of embeddings
d
one column per node
$|V|$

ex) $node\ 4$

$$v = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Each node is assigned a unique embedding vector (i.e., we directly optimize the embedding of each node)

Figures from http://web.stanford.edu/class/cs224w/

Decoder (Similarity function)

- Specifies how the **relationships in the original graph** map to **relationships in embedding space**

Relationships in the original graph

$$similarity\ (u, v)$$

Similarity between node u and node v in the original network

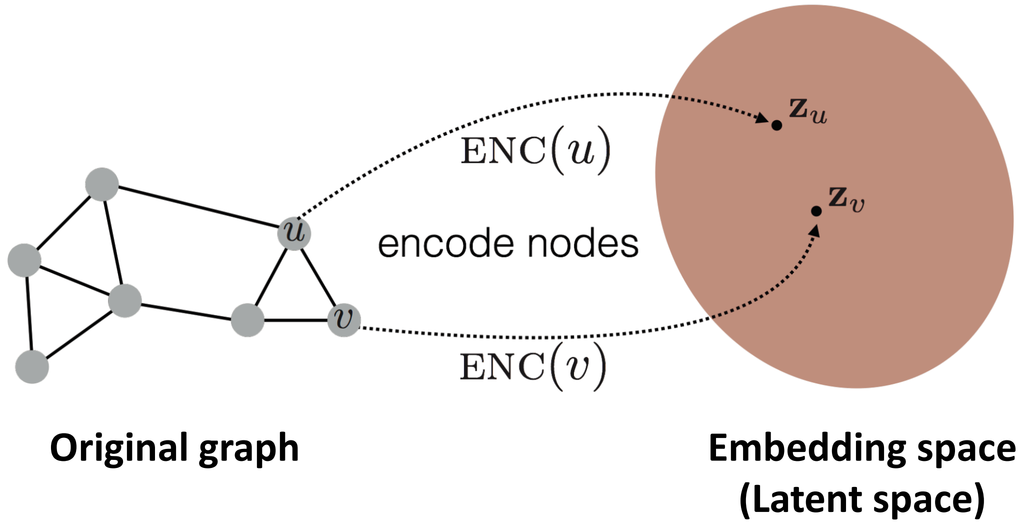
$$\approx$$

Relationships in embedding space

$$\boldsymbol{z}_v^T \boldsymbol{z}_u$$

Dot product between embeddings of node u and node v

Encoder + decoder framework



- **Encoder**: Embedding look-up (Shallow model)
  - Deep encoders (GNNs) later in the lecture
- **Decoder**: Based on dot product

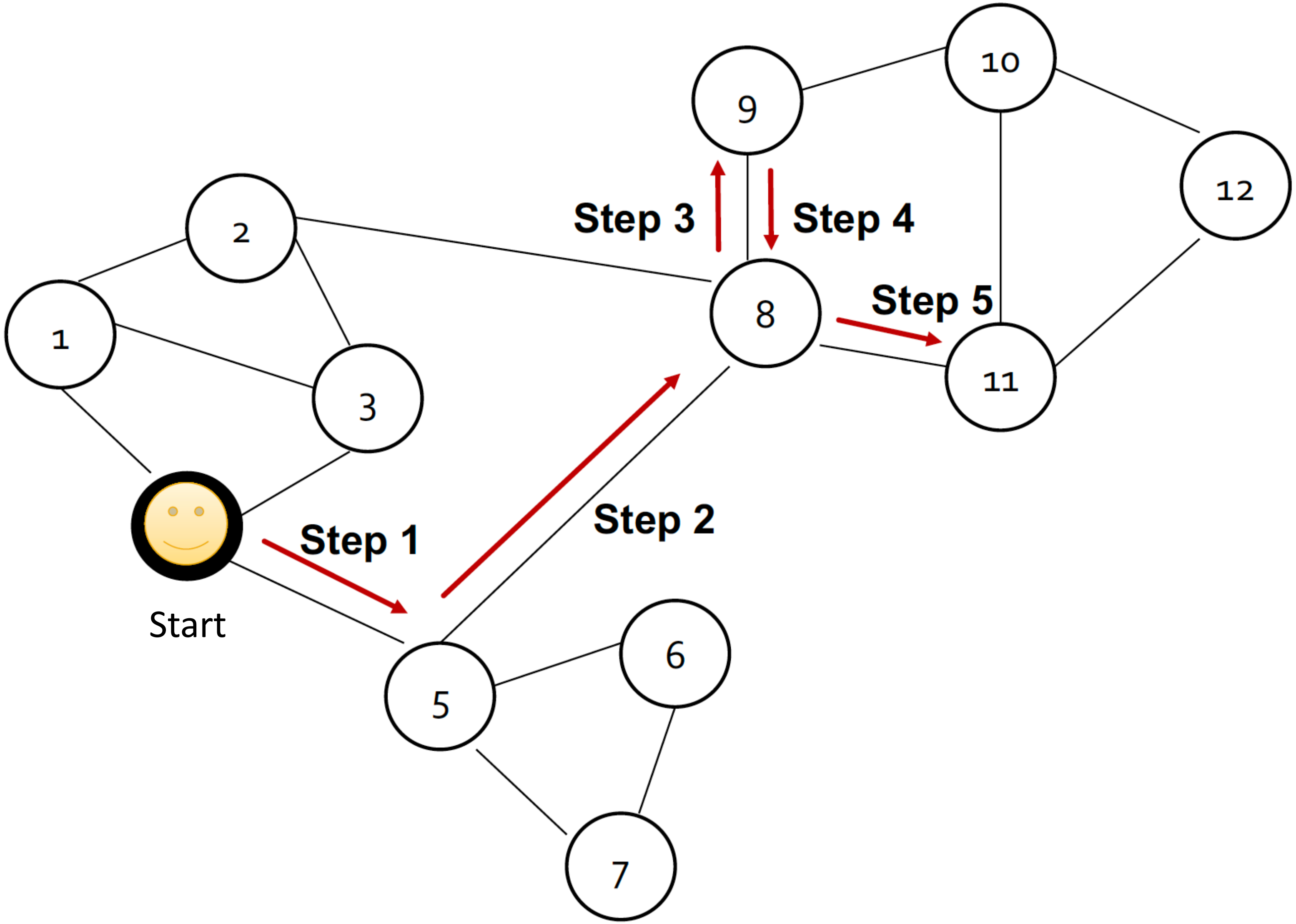
Objective

Maximize $\boldsymbol{z}_v^T \boldsymbol{z}_u$ for node pairs (u, v) that are **similar**

- **How can we define node similarity?**
- **Possible choice**
  - Are two nodes linked?
  - Do they share neighbors?
  - Do they have similar structural roles?
  - …

What is Random walk?

- **Given a graph and a starting node,**
  - 1. Select a neighbor of it at random,
  - 2. Move to this neighbor
  - Repeat 1,2
- Example of random walk
  - Start → 5 → 8 → 9 → 8 → 11
  - (Random) Sequence of nodes

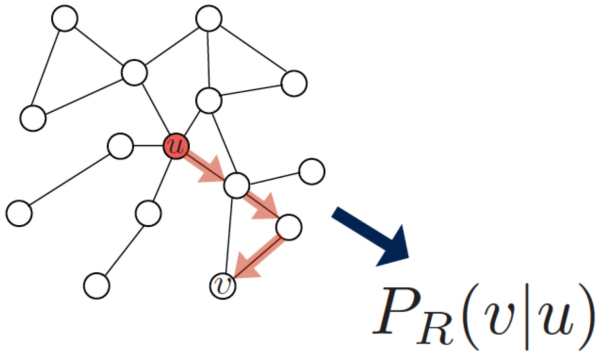


Figures from http://web.stanford.edu/class/cs224w/

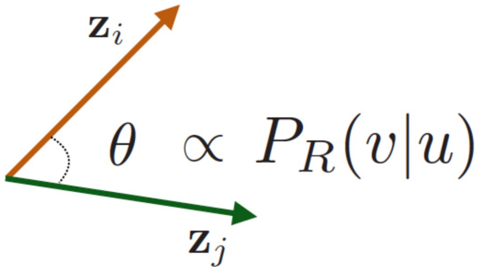
Random walk-based node embeddings: overview

- **Idea**: Learn node embedding such that nearby nodes in the graph are close together in the embedding space
- **Q.** Given a node u, how do we define nearby nodes?
- **A.** Through **random walk!**

- **Step 1.** Estimate the probability of visiting node v on a random walk starting from node u using some random walk strategy R



- **Step 2.** Optimize embeddings to encode these random walk statistics
  - e.g., If two nodes co-occur, maximize their similarity



cf) Dot product = cosine similarity, if node embeddings are unit vectors

Why random walk?

Random walk can reflect both **local** and **high-order** neighborhood information

Figures from http://web.stanford.edu/class/cs224w/

Random walk-based node embeddings: Detailed algorithm

- **Given**: $G = (V, E)$
- **Goal**: To learn a mapping function $f: u \rightarrow R^d$ for $u \in V$
  - $f(u) = \boldsymbol{z}_u \in R^d$
- **Step 1**: Run fixed-length random walks starting from each node u in the graph using some random walk strategy R
- **Step 2**: For each node u collect $N_R(u)$, the multiset of nodes visited on random walks starting from u
  - $N_R(u)$: Neighboring nodes of node u under random walk strategy R
- **Step 3**: Optimize embeddings according to the following objective
  - Objective: Given node u, predict its neighbors $N_R(u)$

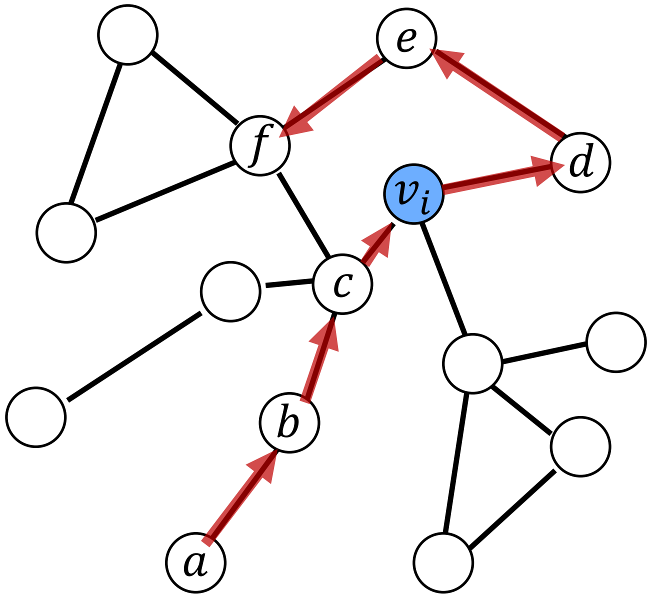
$$\max_f \sum_{u \in V} \log P(N_R(u) | \boldsymbol{z}_u)$$

(Maximum likelihood objective)

Given node u, we aim to maximize the probability of its neighboring nodes

i.e., we want to learn embedding of node u that is predictive of its neighboring nodes

How to define neighboring nodes?



Random walk strategy	R
Example sequence	$a \to b \to c \to v_i \to d \to e \to f$
Window size=2	$a \mid\to b \to c \to v_i \to d \to e \mid\to f$
Center node	v_i
Neighborhood	$N_R(v_i) = b, c, d, e$

Random walk-based node embeddings: Optimization

$$\max_{f} \sum_{u\in V} \log P(\,N_R(u)|\boldsymbol{z}_u) \xrightarrow{\text{Equivalent}} L = \sum_{u\in V} \sum_{v\in N_R(u)} -\log(P(v|\boldsymbol{z}_u))$$

- **Intuition:** Optimize embeddings $\boldsymbol{z}_u$ to maximize the likelihood of random walk co-occurrences
- **Approach:** Parameterize $P(v|\boldsymbol{z}_u)$ using **softmax**

$$P(v|\boldsymbol{z}_u) = \frac{\exp(\boldsymbol{z}_u^T\boldsymbol{z}_v)}{\sum_{j\in V}\exp(\boldsymbol{z}_u^T\boldsymbol{z}_j)}$$

$$L = \underbrace{\sum_{u\in V}}_{\text{Sum over all nodes } u} \underbrace{\sum_{v\in N_R(u)}}_{\text{Sum over nodes } v \text{ seen on random walks starting from } u} -\log(\underbrace{\frac{\exp(\boldsymbol{z}_u^T\boldsymbol{z}_v)}{\sum_{j\in V}\exp(\boldsymbol{z}_u^T\boldsymbol{z}_j)}}_{\text{Predicted probability of } u \text{ and } v \text{ cooccurring on random walk}})$$

$$\boxed{\text{Optimizing random walk embeddings = Finding embeddings } \boldsymbol{z}_u \text{ that minimizes the loss } L}$$

Negative sampling

- $O(|V|^2)$ complexity
- We can approximate this normalization term

- **But, optimizing the loss L is expensive!**

$$L = \sum_{u \in V} \sum_{v \in N_R(u)} -\log\left(\frac{\exp(\mathbf{z}_u^T \mathbf{z}_v)}{\sum_{j \in V} \exp(\mathbf{z}_u^T \mathbf{z}_j)}\right)$$

$$\log\left(\frac{\exp(\mathbf{z}_u^T \mathbf{z}_v)}{\sum_{j \in V} \exp(\mathbf{z}_u^T \mathbf{z}_j)}\right) \approx \log\left(\sigma(\mathbf{z}_u^T \mathbf{z}_v)\right) - \sum_{j=1}^{k} \log(\sigma(\mathbf{z}_u^T \mathbf{z}_j)), \quad j \sim P_V$$

https://arxiv.org/pdf/1402.3722.pdf

- $\sigma(x) = \frac{1}{1+e^{-x}}$ (Sigmoid function)
  - Makes each term a “probability” between 0 and 1
- P_v: Random distribution over nodes

- **Instead of normalizing w.r.t. all nodes, just normalize against k random “negative samples” j**

- How do we sample from P_V to help the training process?
  - Sample k negative nodes considering the degree of each node

Random walk-based node embeddings: Optimization

- After we obtained the objective function, **how do we optimize (minimize) it?**

$$L = \sum_{u \in V} \sum_{v \in N_R(u)} -\log(P(v|\mathbf{z}_u))$$

- **Gradient descent:** A simple and the most common way to minimize L
  - Step 1: Randomly initialize $\mathbf{z}_i$ for all $i \in V$
  - Step 2: Iterate until convergence
    - For all $i \in V$, compute the derivative w.r.t. the loss L, i.e., $\frac{\partial L}{\partial \mathbf{z}_i}$
    - For all $i \in V$, update $\mathbf{z}_i \leftarrow \mathbf{z}_i - \eta \frac{\partial L}{\partial \mathbf{z}_i}$

- **Stochastic Gradient descent:** Instead of evaluating gradients over all examples, evaluate for a **single** node
  - Step 1: Randomly initialize $\mathbf{z}_i$ for all $i \in V$
  - Step 2: Iterate until convergence: $L^{(u)} = \sum_{v \in N_R(u)} -\log(P(v|\mathbf{z}_u))$
    - Sample a node i, for all $j \in N_R(i)$ compute the derivative w.r.t. the loss L, i.e., $\frac{\partial L^{(i)}}{\partial \mathbf{z}_j}$
    - For all $j \in N_R(i)$, update $\mathbf{z}_j \leftarrow \mathbf{z}_j - \eta \frac{\partial L^{(i)}}{\partial \mathbf{z}_j}$

Random walk-based node embeddings: Summary

- **Step 1**: Run fixed-length random walks starting from each node u in the graph using some random walk strategy R
- **Step 2**: For each node u collect $N_R(u)$, the multiset of nodes visited on random walks starting from u
- **Step 3**: Optimize embeddings according to the following objective using Stochastic Gradient Descent
  - The optimization can be made efficient through negative sampling technique!

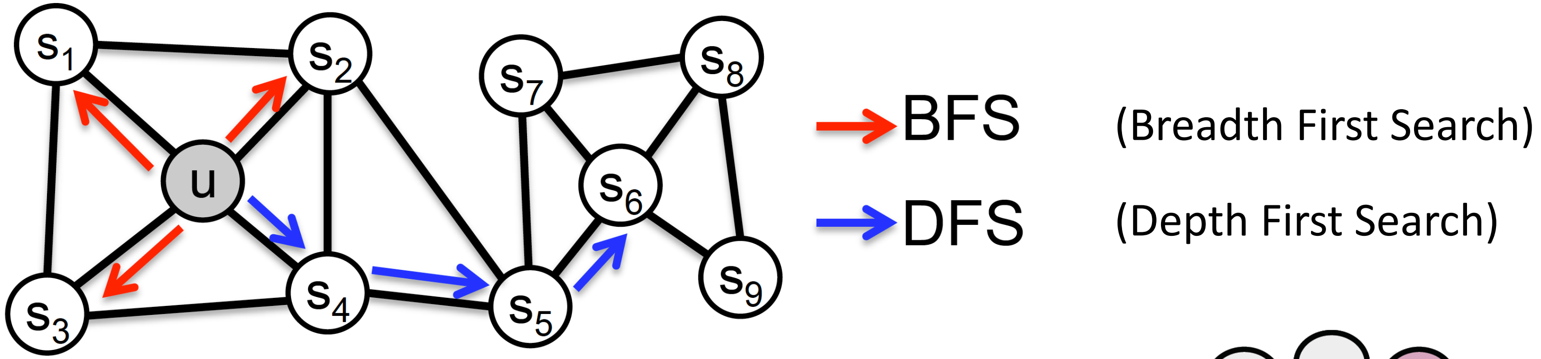
$$L = \sum_{u \in V} \sum_{v \in N_R(u)} -\log(P(v|\mathbf{z}_u)) = \sum_{u \in V} \sum_{v \in N_R(u)} -\log\left(\frac{\exp(\mathbf{z}_u^T \mathbf{z}_v)}{\sum_{j \in V} \exp(\mathbf{z}_u^T \mathbf{z}_j)}\right)$$

- So far, we have discussed about how to optimize embeddings given a random walk strategy $\boldsymbol{R}$
  - Different random walk strategies make different $N_R(u)$
  - **What strategies are there?**
    - 1. Idea of Deepwalk [1]: Simple random walk (we talked about so far)
    - 2. Idea of node2vec [2]: Biased random walk

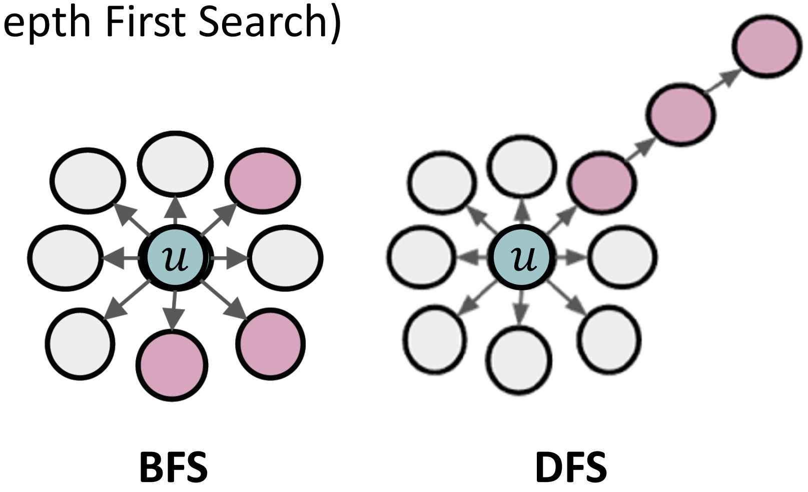
[1] Perozzi, Bryan, Rami Al-Rfou, and Steven Skiena. "Deepwalk: Online learning of social representations." KDD 2014.
[2] Grover, Aditya, and Jure Leskovec. "node2vec: Scalable feature learning for networks." KDD 2016.

Node2vec: Biased walks

- **Idea**: Use flexible, biased random walks that can trade off between **local** and **global** views of the network
  - Deepwalk's simple random walk mainly focuses on the global view



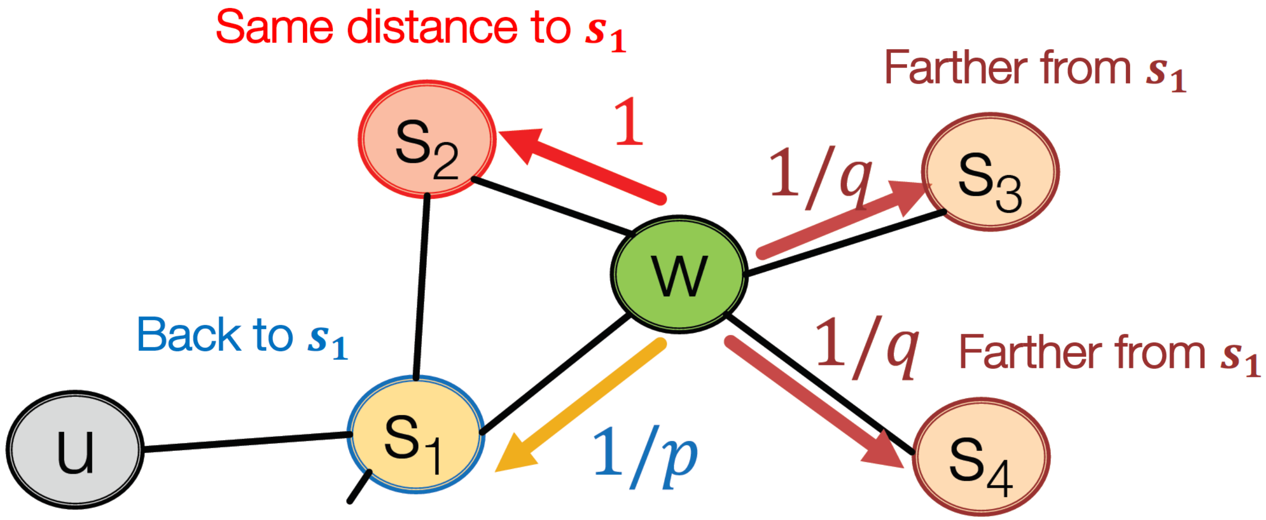
- **Two strategies to define a neighborhood** $N_R(u)$ of node u: BFS and DFS
- Example: Walk of length 3 from node u
  - $N_{BFS}(u) = \{s_1, s_2, s_3\}$ **Local** microscopic view
  - $N_{DFS}(u) = \{s_4, s_5, s_6\}$ **Global** macroscopic view



How can we interpolate between BFS and DFS?

Node2vec: Interpolating BFS and DFS

- Biased fixed-length random walk R starting from node u generates neighborhood $N_R(u)$
- **Two parameters to control the interpolation**
  - **Return parameter** p
    - Return back to the previous node
  - **In-out parameter** q
    - Moving outwards (DFS) vs. inwards (BFS)
    - Intuitively, q is the "ratio" of BFS vs. DFS



Same distance to $\boldsymbol{s_1}$
Farther from $\boldsymbol{s_1}$
1
$1/q$
s_2
s_3
w
Back to $\boldsymbol{s_1}$
$1/q$
Farther from $\boldsymbol{s_1}$
u
s_1
$1/p$
s_4

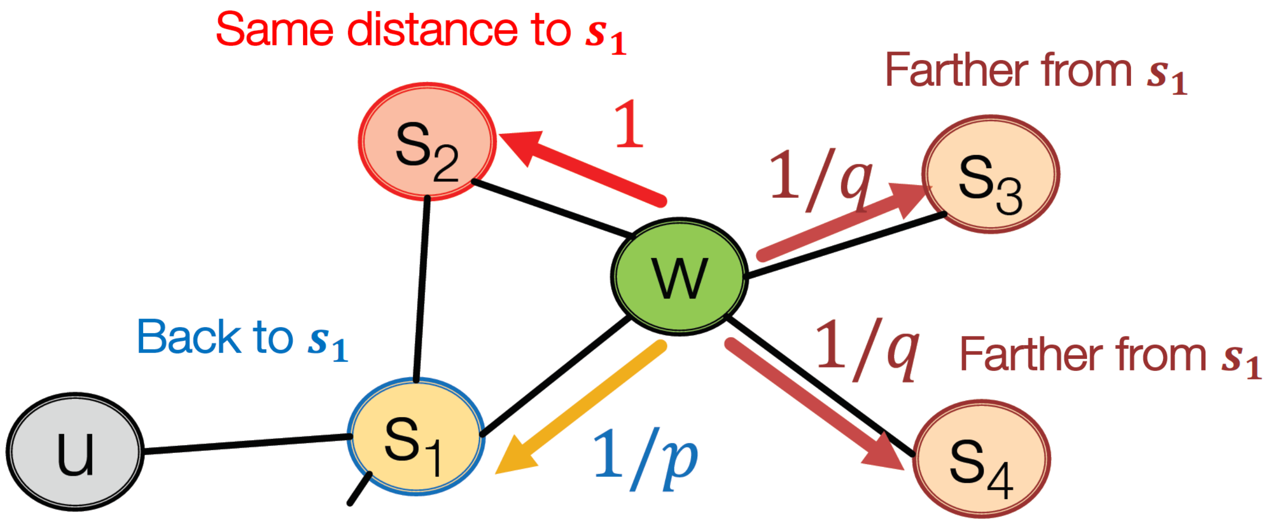
Current situation

- Random walk that started from node u just traversed edge (s_1, w) and is now at w
- At this point, neighbors of w can be s_1, s_2, s_3 or s_4

Idea of biased random walk
Remember where the walk came from!

Node2vec: Details of biased random walk

- **Main idea**: Move to neighbors considering where the walk came from



w →

Target neighbor	Prob.	Dist. (s_1, t)
s_1	$1/p$	0
s_2	1	1
s_3	$1/q$	2
s_4	$1/q$	2

- p, q model transition probabilities ($1/p$, $1/q$, 1 are unnormalized probabilities)
  - p: return parameter
  - q: walk away parameter
- **BFS-like walk**: Low value of p
- **DFS-like walk**: low value of q
- $N_R(u)$ are the nodes visited by the biased walk

Outline

- Overview
- Node-level Network Embedding
  - Random walk-based Node Embeddings (DeepWalk / node2vec)
- Graph Neural Network (GNN)
  - Graph Convolutional Neural Network (GCN)
  - Graph Attention Network (GAT)
  - Relational GCN
  - GraphSAGE
  - Deep Graph Infomax (DGI)
  - GNNs with Edge Embeddings

Recall: Encoder

- Maps each node to a low-dimensional vector

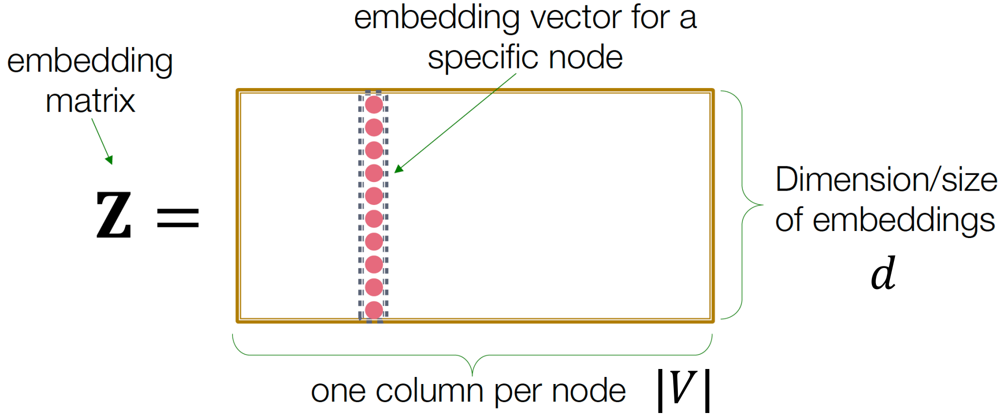
$$ENC(v) = \mathbf{z}_v$$

d-dimensional embedding vector

Node in the input graph

- **Simplest encoding approach**: Encoder is just an embedding-lookup (Shallow model)

$$ENC(v) = \mathbf{z}_v = \mathbf{Z} \cdot v \qquad \mathbf{Z} \in R^{d\times|V|} \qquad v \in I^{|V|}$$



embedding matrix
$\mathbf{Z} =$
embedding vector for a specific node
Dimension/size of embeddings d
one column per node $|V|$

ex) *node* 4

$$v = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Each node is assigned a unique embedding vector (i.e., we directly optimize the embedding of each node)

This lecture: Deep encoder

Deep graph encoder

- **Deep Encoder** = **Graph neural network** (GNN)

Shallow model

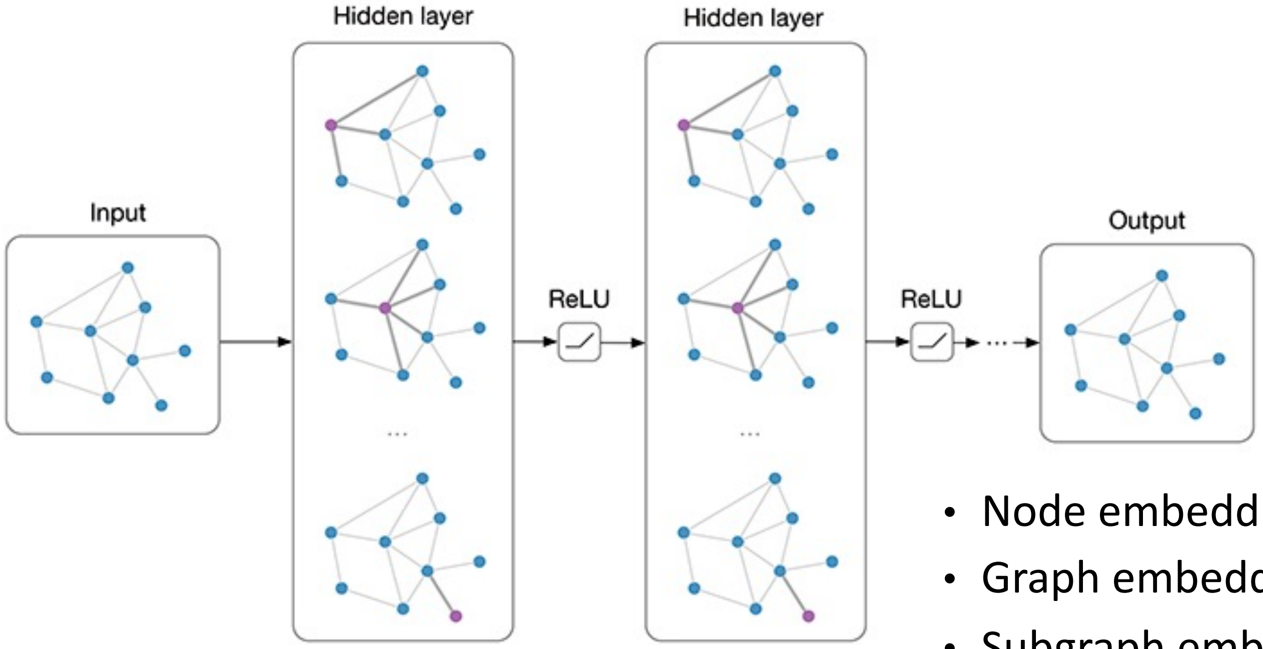
$$ENC(v) = \mathbf{z}_v$$

Deep model

$ENC(v) =$ Multiple layers of non-linear transformations-based on graph structure

Graph convolution

Graph convolution

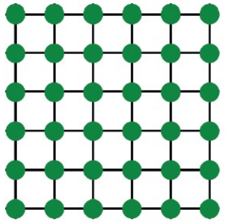


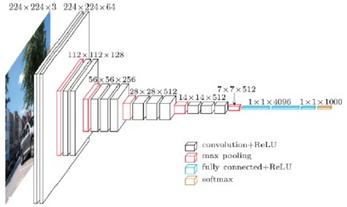
- Node embedding
- Graph embedding
- Subgraph embedding
- Edge embedding
- …

Can we use existing deep learning models? e.g., CNN, RNN, etc

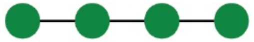
Challenges of graph representation learning

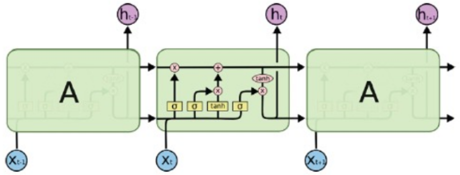
- Existing deep neural networks are designed for data with regular-structure (grid or sequence)
    - CNNs for fixed-size images/grids …





- RNNs for text/sequences …



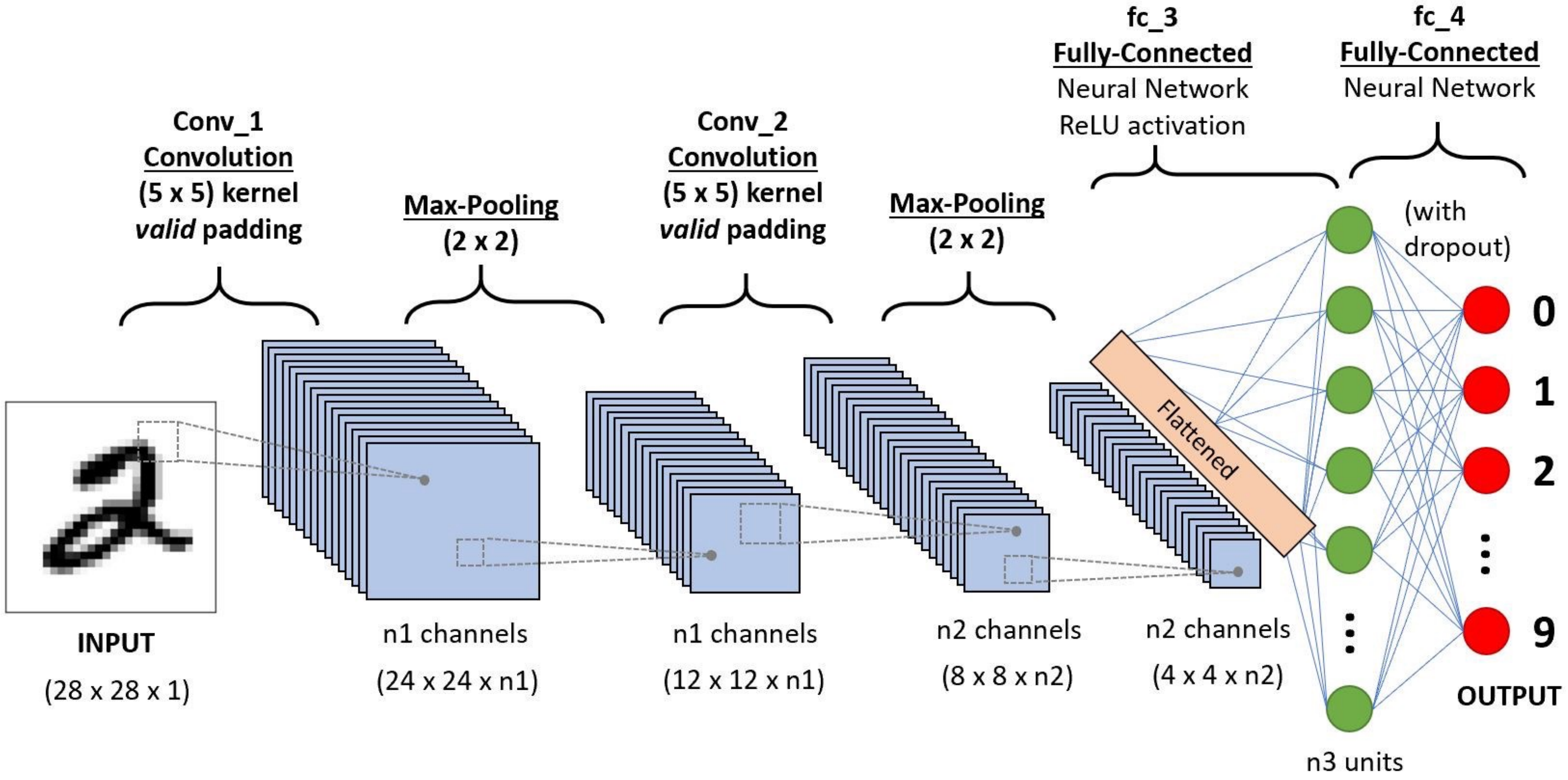


- **Graphs are very complex**
    - Arbitrary structures (no spatial locality like grids / no fixed orderings)
    - Heterogeneous: Directed/undirected, binary/weighted/typed, multimodal features
    - Large-scale: More than millions of nodes and billions of edges

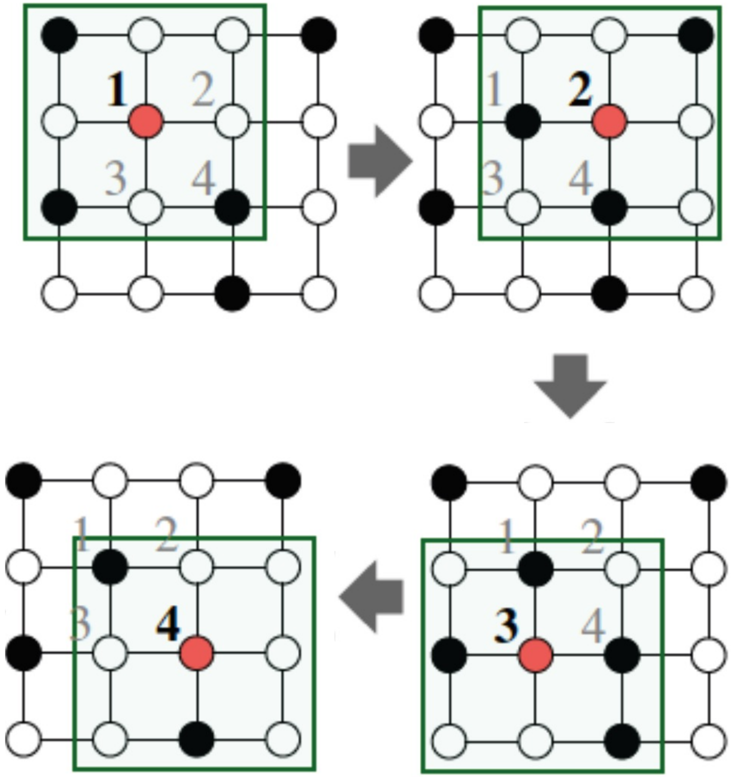
(Figure credit) Tutorial on Graph Representation Learning, AAAI 2019

Background: Convolutional neural networks for images

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



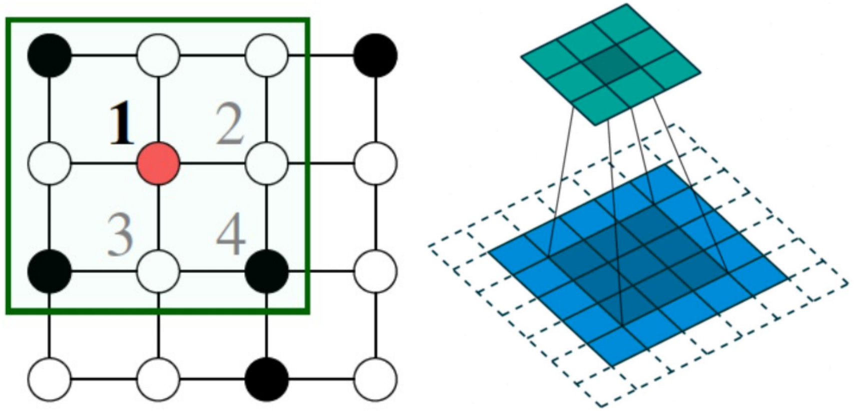
CNN on an image



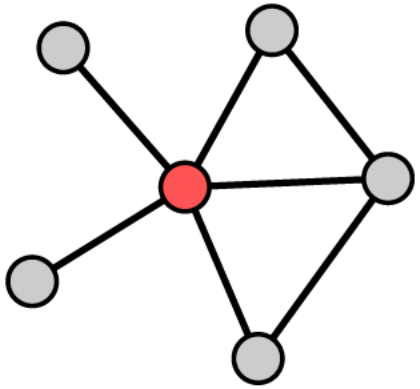
(Figure credit) https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

From images to graphs: Local receptive field on graphs

- How should we define local receptive fields on graphs?
  - Local subgraphs

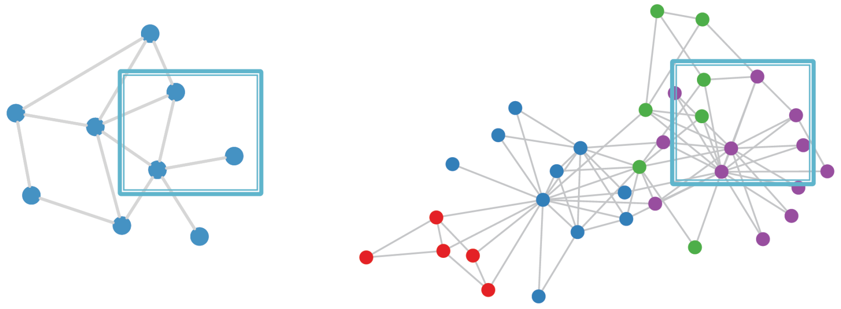


Image



Graph

Graphs look like this



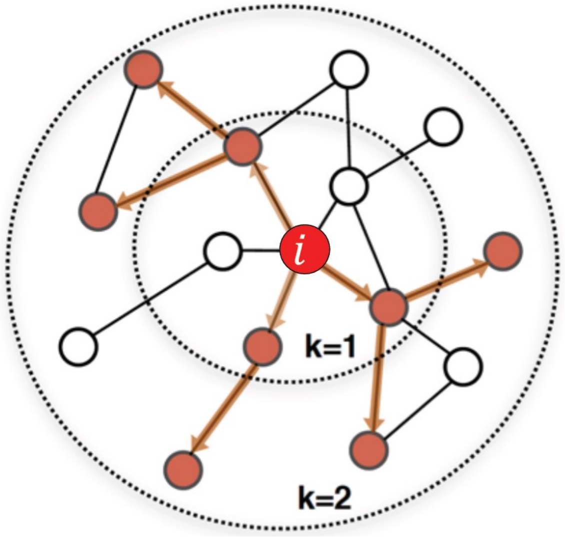
- There is no fixed notion of locality or sliding window on the graph
- No order among neighboring nodes
  - Permutation invariant

- **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: $\sum_{v_j \in N(i)} W_j h_j$

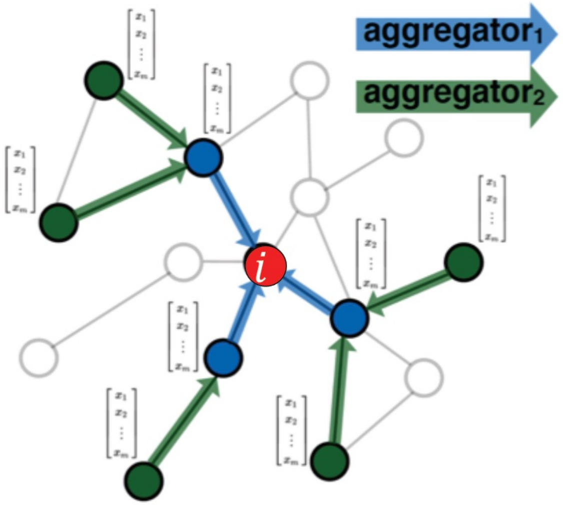
(Figure credit) https://deepgraphlearning.github.io/coursewebsite/schedule

Graph Convolutional Network (GCN)

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



Determine node computation graph

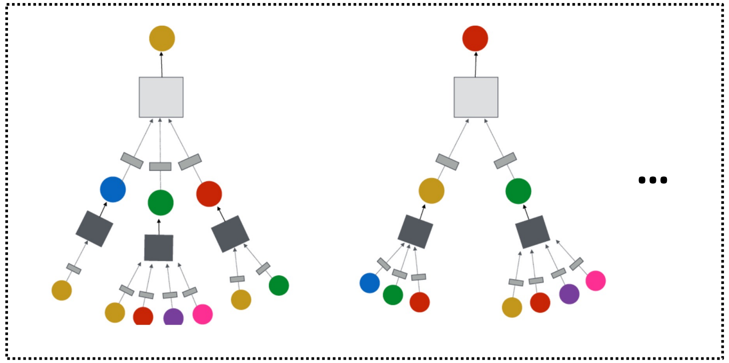


Propagate messages and transform information

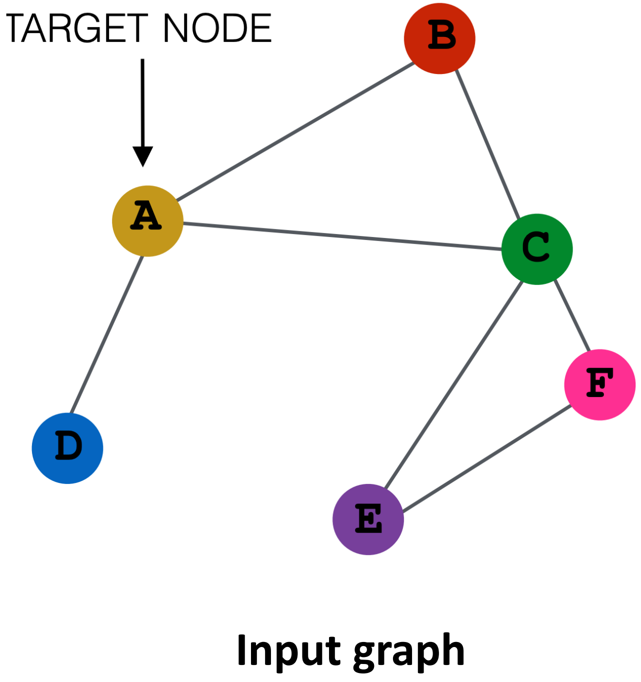
Learn how to propagate information across the graph to compute node features

Kipf, Thomas N., and Max Welling. "Semi-supervised classification with graph convolutional networks." ICLR 2017

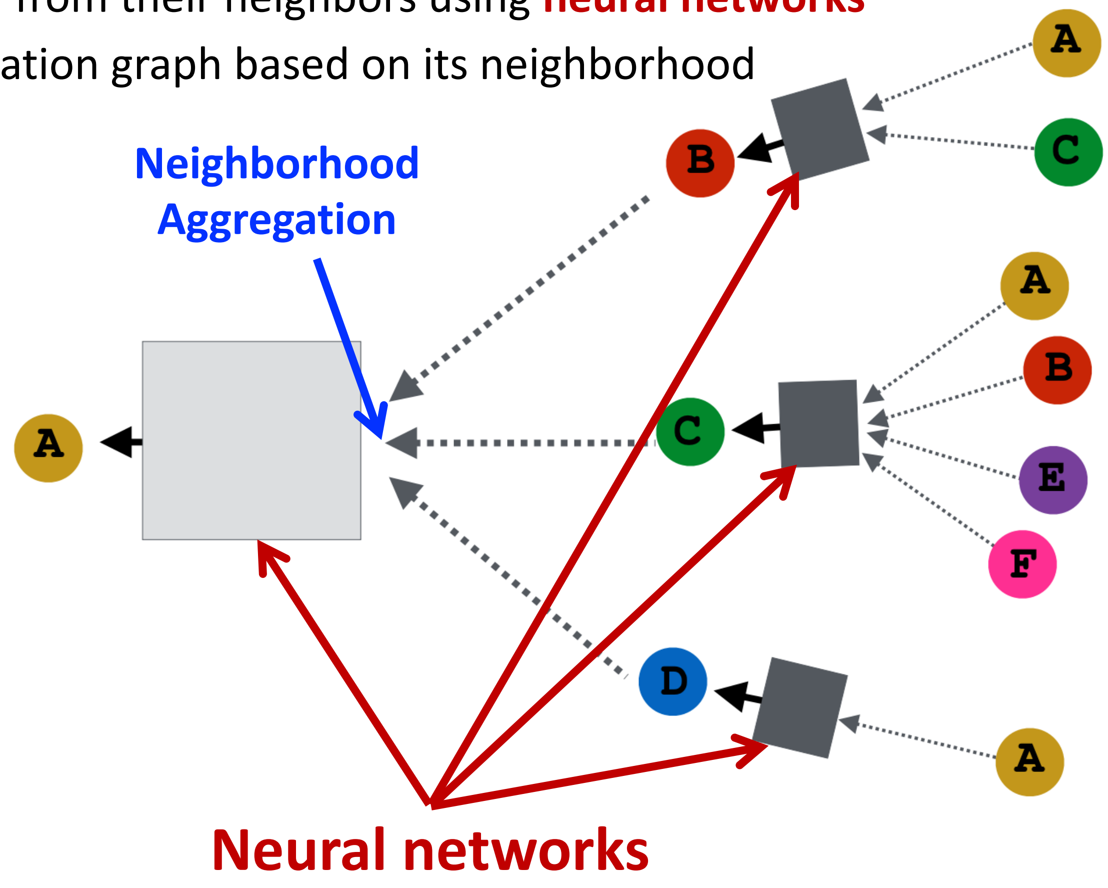
GCN: Neighborhood aggregation



- 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

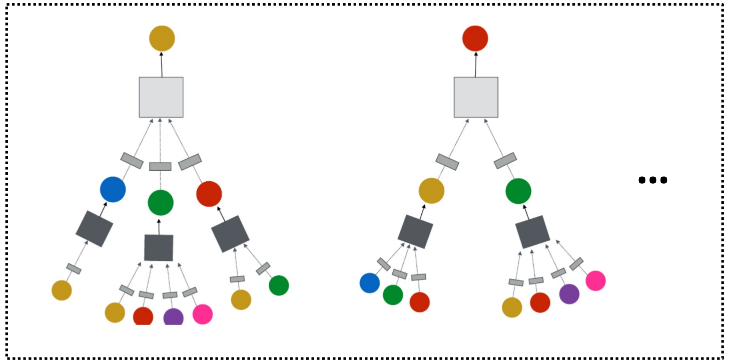

Neighborhood Aggregation

Neural networks

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

Kipf, Thomas N., and Max Welling. "Semi-supervised classification with graph convolutional networks." ICLR 2017

GCN: Basic approach



- **1. What kind of neural network?**
  - Simple multiplication of weight matrices ($\boxed{\boldsymbol{B}}$ and $\boxed{\boldsymbol{W}}$)
- **2. What kind of aggregation?**
  - Average

Weight matrix

Embedding of v at layer $\boldsymbol{l}$

Total number of layers

$$\boldsymbol{h}_v^{(l+1)} = \sigma\left(\boxed{\boldsymbol{W}_l} \boxed{\sum_{u\in N(v)} \frac{\boldsymbol{h}_u^{(l)}}{|N(v)|}} + \boxed{\boldsymbol{B}_l}\boxed{\boldsymbol{h}_v^{(l)}}\right), \ \forall l \in \{0,1,\dots,\boxed{L}-1\}$$

Initial embedding of v

$$\boxed{\boldsymbol{h}_v^0} = \boxed{\boldsymbol{x}_v}$$

Feature of node v

Average of neighboring nodes' previous layer embeddings

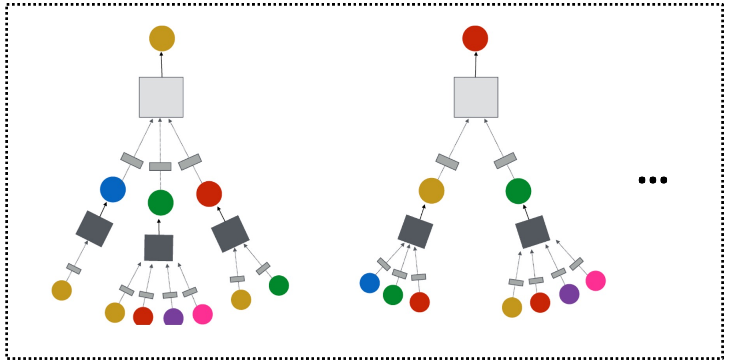
$$\boxed{\boldsymbol{z}_v} = \boldsymbol{h}_v^{(L)}$$

Final embedding of v

How do we train the embeddings?

Kipf, Thomas N., and Max Welling. "Semi-supervised classification with graph convolutional networks." ICLR 2017

GCN: Matrix formulation



- GCN can be efficiently computed in a matrix form

$$\boldsymbol{h}_v^{(l+1)} = \sigma\left(\boldsymbol{W}_l \boxed{\sum_{u\in N(v)} \frac{\boldsymbol{h}_u^{(l)}}{|N(v)|}} + \boldsymbol{B}_l \boldsymbol{h}_v^{(l)}\right), \quad \forall l \in \{0,1,\dots,L-1\}$$

$\boldsymbol{h}_v^{0} = \boldsymbol{x}_v$, $\boldsymbol{z}_v = \boldsymbol{h}_v^{(L)}$

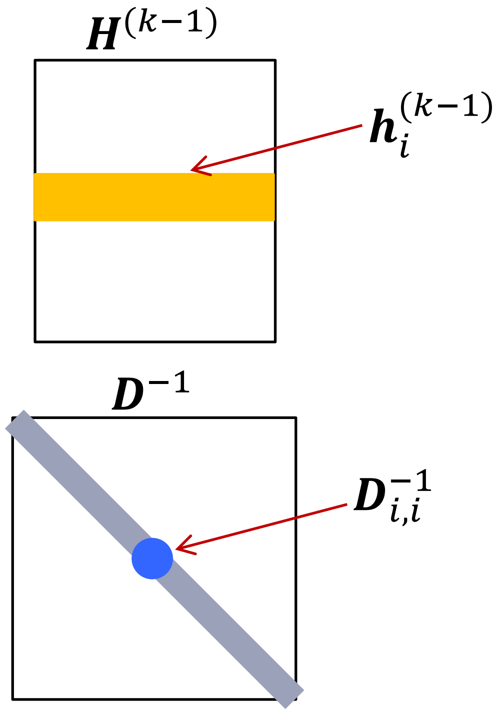
$\boxed{\boldsymbol{D}^{-1}\boldsymbol{A}\boldsymbol{H}^{(l)}}$ **(Matrix form)**

$$\boldsymbol{H}^{(l+1)} = \sigma(\boxed{\tilde{\boldsymbol{A}}\boldsymbol{H}^{(l)}\boldsymbol{W}_l^T} + \boxed{\boldsymbol{H}^{(l)}\boldsymbol{B}_l^T}) \text{ where } \tilde{\boldsymbol{A}} = \boldsymbol{D}^{-1}\boldsymbol{A}$$

Neighborhood aggregation

Self transformation

Since $\tilde{\boldsymbol{A}}$ is sparse, sparse matrix multiplication can be used (efficient)



$$\sum_{u\in N(v)} \boldsymbol{h}_u^{(l)} = \boldsymbol{A}_v \boldsymbol{H}^{(l)}$$

$$\boldsymbol{D}_{v,v} = Deg(v) = |N(v)|$$

$$\boldsymbol{D}_{v,v}^{-1} = \frac{1}{|N(v)|}$$

Kipf, Thomas N., and Max Welling. "Semi-supervised classification with graph convolutional networks." ICLR 2017

GCN: Training

- We need to define the loss function on the embeddings
- We can feed the **final embeddings $\boldsymbol{z}_v$** 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**

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

  - y_v: Label of node v
  - f_θ: Classifier with parameter θ
  - $\mathcal{L}$ 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 ($\boldsymbol{A}_{v,u} = 1$ if an edge exists between node v and node u, otherwise 0)

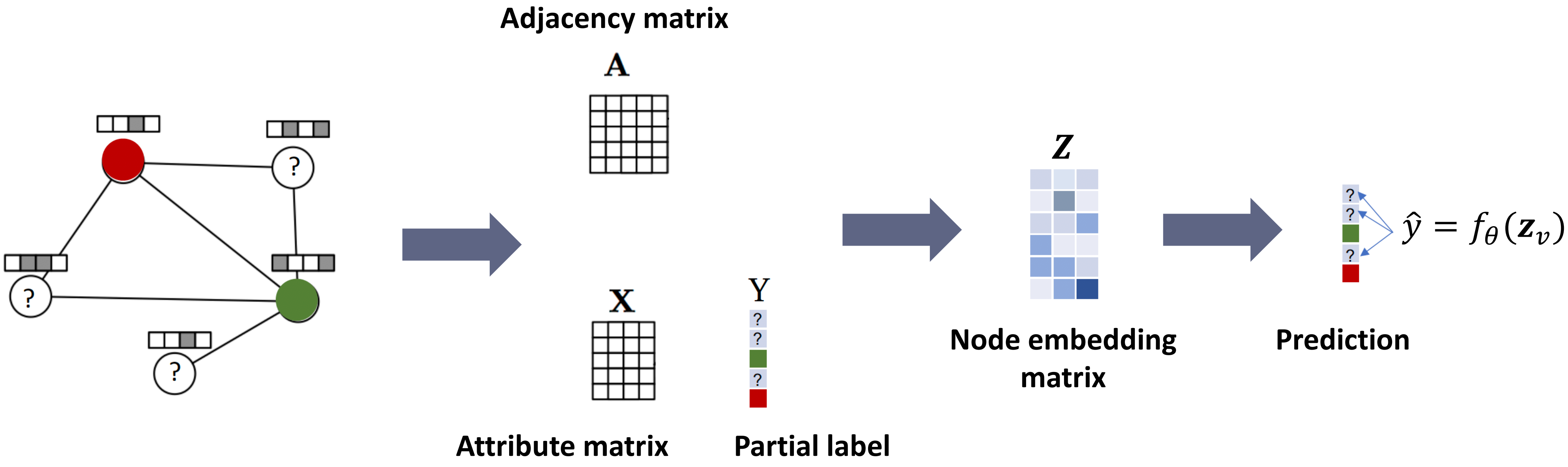
$$\min_{\theta} \sum_{v,u \in V} \mathcal{L}(\boldsymbol{A}_{v,u}, f_\theta(\boldsymbol{z}_v, \boldsymbol{z}_u))$$

f_θ: Encoder

Kipf, Thomas N., and Max Welling. "Semi-supervised classification with graph convolutional networks." ICLR 2017

GCN: Supervised training

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



$$\mathcal{L} = -\sum_{v,\in V} \boxed{y_v} \log \boxed{f_\theta(\boldsymbol{z}_v)} + (1 - y_v)\log(1 - f_\theta(\boldsymbol{z}_v))$$

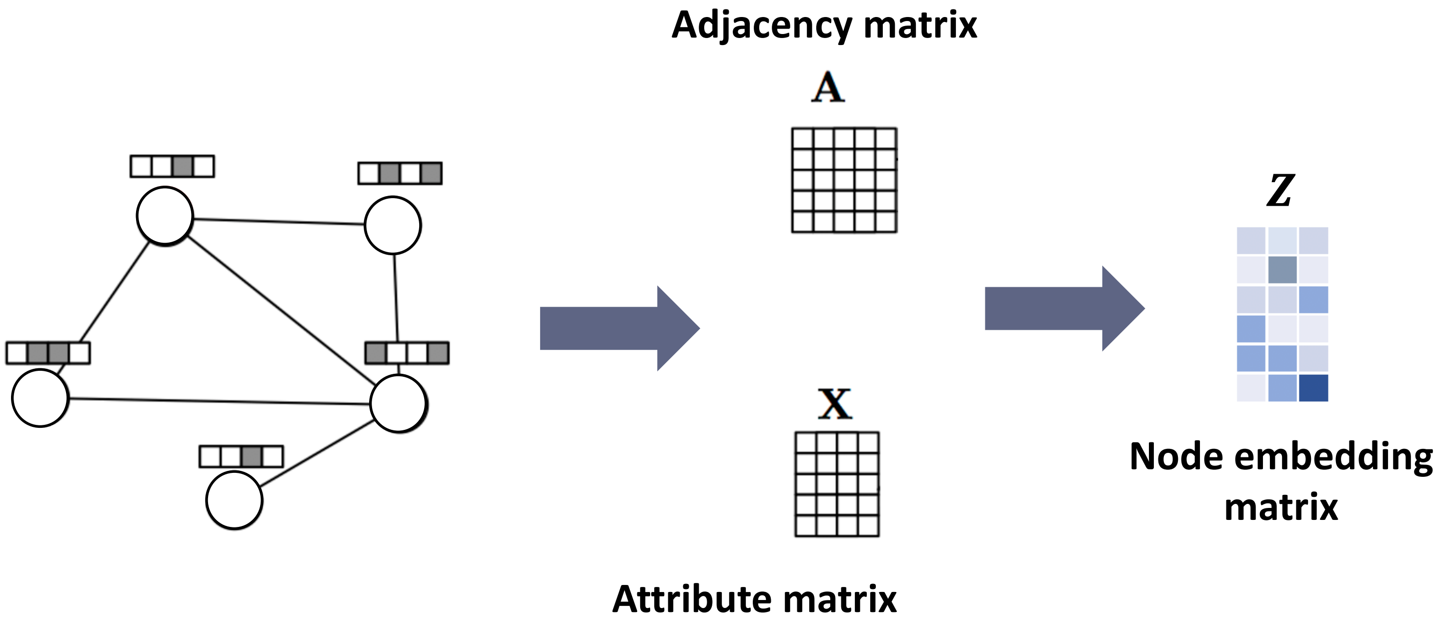
Ground truth label

Model prediction

Kipf, Thomas N., and Max Welling. "Semi-supervised classification with graph convolutional networks." ICLR 2017

GCN: Unsupervised training

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



Adjacency matrix

$\mathbf{A}$

$\mathbf{X}$

Attribute matrix

$\boldsymbol{Z}$

Node embedding matrix

$$\mathcal{L} = -\sum_{v,u \in V} \boxed{\boldsymbol{A}_{v,u}} \log \boxed{f_\theta(\boldsymbol{z}_v, \boldsymbol{z}_u)} + \left(1 - \boldsymbol{A}_{v,u}\right) \log(1 - f_\theta(\boldsymbol{z}_v, \boldsymbol{z}_u))$$

Ground truth label

Model prediction

Kipf, Thomas N., and Max Welling. "Semi-supervised classification with graph convolutional networks." ICLR 2017

Graph Attention Networks (GAT) (1/3)

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

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

Attention weight

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

  - α_{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!

Veličković, Petar, et al. "Graph attention networks." ICLR 2018

Graph Attention Networks (GAT) (2/3)

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

$$\boldsymbol{h}_v^{(l+1)} = \sigma\left(\boldsymbol{W}_l \sum_{u\in N(v)\cup v} \underbrace{\boxed{\alpha_{vu}}}_{\text{Attention weight}} \boldsymbol{h}_u^{(l)}\right)$$

- α_{vu}: Importance of node u to node v as its neighboring node

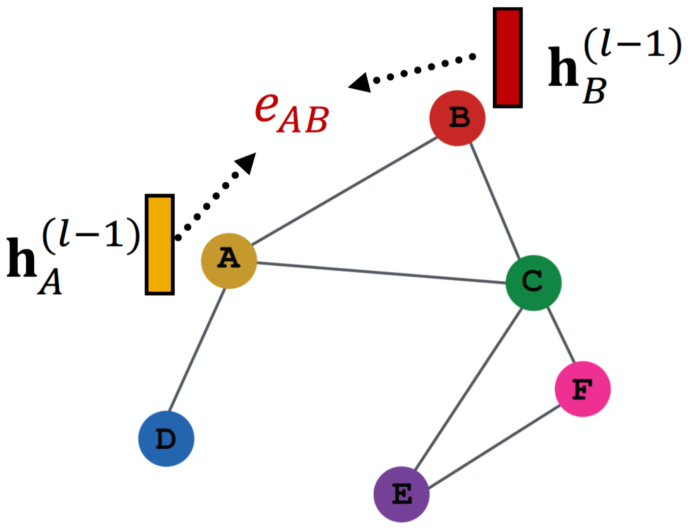
- Computing the attention weight

$$e_{vu} = a(\boldsymbol{W}_l \boldsymbol{h}_v^{(l)}, \boldsymbol{W}_l \boldsymbol{h}_u^{(l)})$$

(Importance of node u's message to node v)

$$\alpha_{vu} = \frac{\exp(e_{vu})}{\sum_{k\in N(v)} \exp(e_{vk})}$$

(Normalization)



$$e_{AB} = a(\boldsymbol{W}_{l-1}\boldsymbol{h}_A^{(l-1)}, \boldsymbol{W}_{l-1}\boldsymbol{h}_B^{(l-1)})$$

How do we define $a(\cdot)$?

Veličković, Petar, et al. "Graph attention networks." ICLR 2018

Graph Attention Networks (GAT) (3/3)

$$\alpha_{vu} = \frac{\exp(e_{uv})}{\sum_{k \in N(v)} \exp(e_{vk})}$$

- **Defining the function $\boldsymbol{a}(\cdot)$**

$$e_{vu} = a(\boldsymbol{W}_l \boldsymbol{h}_v^{(l)}, \boldsymbol{W}_l \boldsymbol{h}_u^{(l)})$$

(Importance of node u's message to node v)

$$= \mathrm{Linear}(\mathrm{Concat}\left(\boldsymbol{W}_l \boldsymbol{h}_v^{(l)}, \boldsymbol{W}_l \boldsymbol{h}_u^{(l)}\right))$$

Parameters of Linear layer is jointly trained end-to-end with other parameters of GAT

- **Multi-head attention** (Introduced in Transformer)
  - Create multiple attention scores using multiple copies of parameters

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

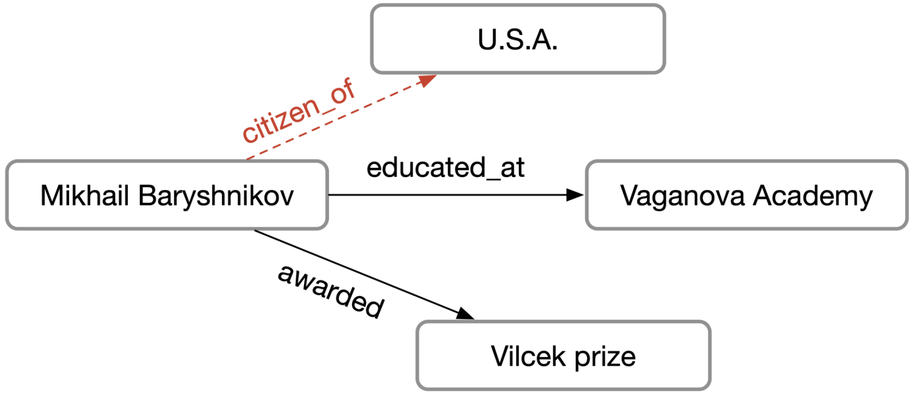
$$\boldsymbol{h}_v^{(l)}[2] = \sigma\left(\boldsymbol{W}_l \sum_{u \in N(v) \cup v} \alpha_{vu}^{[2]} \boldsymbol{h}_u^{(l)}\right)$$

$$\boldsymbol{h}_v^{(l)}[3] = \sigma\left(\boldsymbol{W}_l \sum_{u \in N(v) \cup v} \alpha_{vu}^{[3]} \boldsymbol{h}_u^{(l)}\right)$$

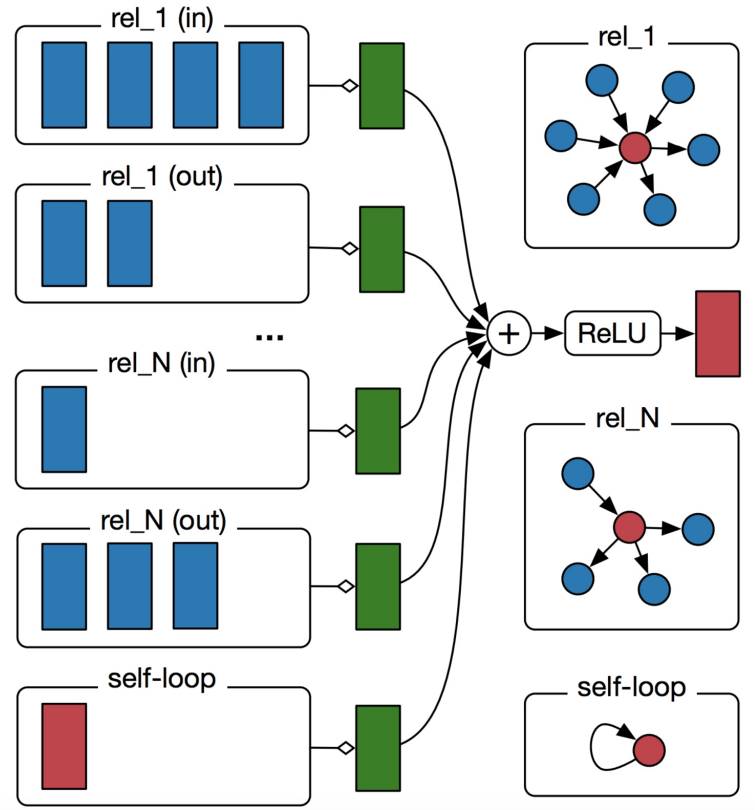
$$\boldsymbol{h}_v^{(l)} = \mathrm{AGG}(\boldsymbol{h}_v^{(l)}[1], \boldsymbol{h}_v^{(l)}[2], \boldsymbol{h}_v^{(l)}[3])$$

The final embedding aggregates the outputs of multi-head attention

R-GCN: RELATIONAL GCN



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



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

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

Schlichtkrull, Michael, et al. “Modeling Relational Data with Graph Convolutional Networks.” ESWC 2018

GraphSAGE (1/2)

- **Generalization of GCN/GAT**
  - So far we have aggregated the neighbor messages by taking their (weighted) average Can we do better?

$$\boldsymbol{h}_v^{(l+1)} = \sigma\left(\boldsymbol{W}_l \overbrace{\boxed{\sum_{u\in N(v)} \frac{\boldsymbol{h}_u^{(l)}}{|N(v)|}}}^{\text{Average of neighboring nodes}} \overbrace{\boxed{+}}^{\text{Add self representation}} \boldsymbol{B}_l \boldsymbol{h}_v^{(l)}\right)$$

GCN

$$\boldsymbol{h}_v^{(l+1)} = \sigma\left(\boldsymbol{W}_l \sum_{u\in N(v)\cup v} \overbrace{\boxed{\alpha_{vu}}}^{\text{Attention weight}} \boldsymbol{h}_u^{(l)}\right)$$

GAT

GraphSAGE

$$\boldsymbol{h}_v^{(l+1)} = \sigma\left(\left[\boldsymbol{W}_l \cdot \overbrace{\boxed{\mathrm{AGG}}}^{\text{Generalized representation}}\left(\left\{\boldsymbol{h}_u^{(l)} | u \in N(v)\right\}\right) \overbrace{\boxed{,}}^{\text{Generalized representation}} \boldsymbol{B}_l \boldsymbol{h}_v^{(l)}\right]\right)$$

Hamilton, William L., Rex Ying, and Jure Leskovec. "Inductive representation learning on large graphs." NeurIPS 2017

GraphSAGE (2/2)

- **Generalization of GCN/GAT**
  - So far we have aggregated the neighbor messages by taking their (weighted) average Can we do better?

Generalized representation

Generalized representation
(Concatenation here)

$$\boldsymbol{h}_v^{(l+1)} = \sigma\left(\left[\boldsymbol{W}_l \cdot \boxed{\mathrm{AGG}}\left(\left\{\boldsymbol{h}_u^{(l)} | u \in N(v)\right\}\right)\boxed{,}\boldsymbol{B}_l \boldsymbol{h}_v^{(l)}\right]\right)$$

- **Variants of AGG**
  - **Mean:** Same as GCN
    - $\mathrm{AGG} = \sum_{u \in N(v)} \frac{\boldsymbol{h}_u^{(l)}}{|N(v)|}$
  - **Pool:** Transform neighbor vectors and apply symmetric vector function
    - $\mathrm{AGG} = \gamma\left(\left\{\mathrm{MLP}(\boldsymbol{h}_u^{(l)}) | u \in N(v)\right\}\right)$, where γ is element-wise mean/max/min
  - **LSTM:** Apply LSTM to shuffled neighbors
    - $\mathrm{AGG} = \mathrm{LSTM}\left(\left[\boldsymbol{h}_u^{(l)} | u \in \pi(N(v))\right]\right)$

Hamilton, William L., Rex Ying, and Jure Leskovec. "Inductive representation learning on large graphs." NeurIPS 2017

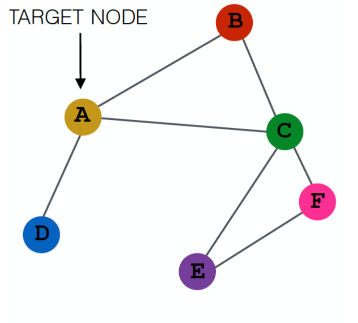
A GNN layer: Overview

- GNN layer = 1) **Message** + 2) **Aggregation**
- Compress a set of vectors into a single vector
- Different types of GNN layers
  - GCN, GraphSAGE, GAT, …

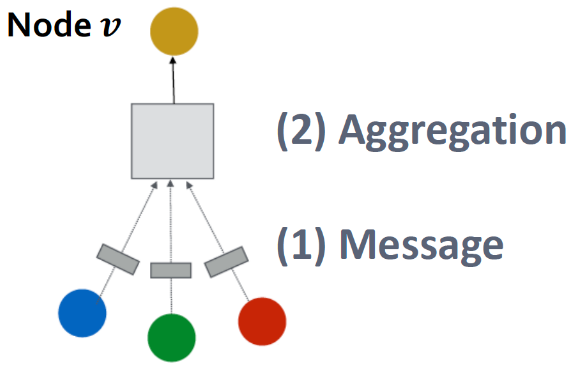
(GCN) $$\boldsymbol{h}_v^{(l+1)} = \sigma\left(\boldsymbol{W}_l \sum_{u\in N(v)} \frac{\boldsymbol{h}_u^{(l)}}{|N(v)|} + \boldsymbol{B}_l \boldsymbol{h}_v^{(l)}\right)$$

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

(GraphSAGE) $$\boldsymbol{h}_v^{(l+1)} = \sigma\left(\left[\boldsymbol{W}_l \cdot \mathrm{AGG}\left(\left\{\boldsymbol{h}_u^{(l)} | u \in N(v)\right\}\right), \boldsymbol{B}_l \boldsymbol{h}_v^{(l)}\right]\right)$$



INPUT GRAPH



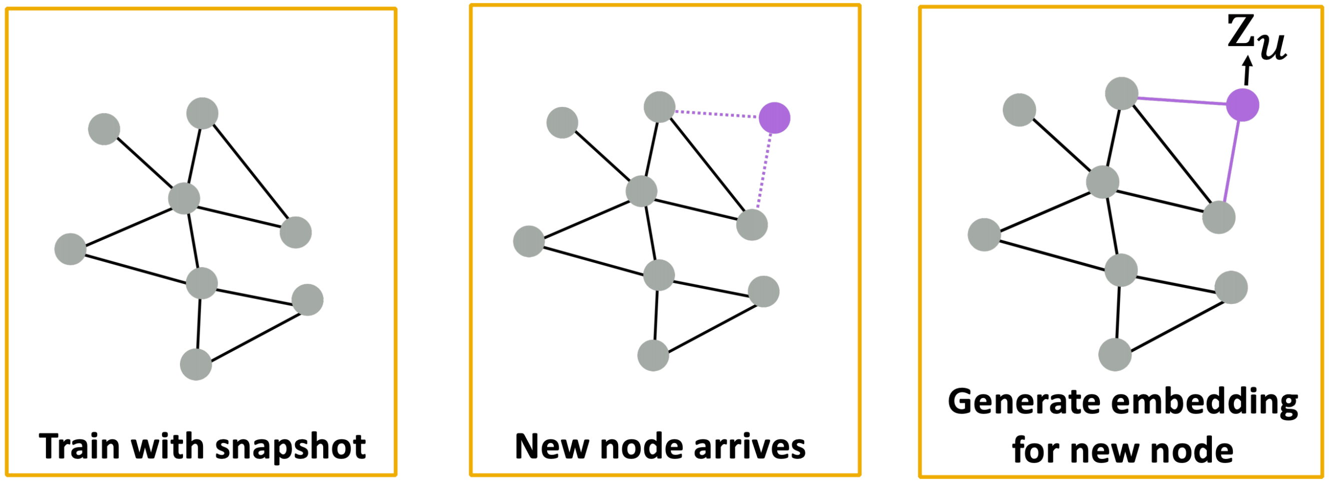
- (1) Message computation
  - Message function: $\boldsymbol{m}_u^{(l)} = \mathrm{MSG}^{(l)}(\boldsymbol{h}_u^{(l)})$
  - Example: A linear layer $\boldsymbol{m}_u^{(l)} = \boldsymbol{W}_l \boldsymbol{h}_u^{(l)}$

- (2) Aggregation
  - $\boldsymbol{h}_v^{(l)} = \mathrm{AGG}\left(\left\{\boldsymbol{m}_u^{(l)} | u \in N(v)\right\}\right)$
  - Example: SUM(), MEAN(), or MAX() aggregator
    - $\boldsymbol{h}_v^{(l)} = \mathrm{SUM}\left(\left\{\boldsymbol{m}_u^{(l)} | u \in N(v)\right\}\right) = \sum_{u\in N(v)} \boldsymbol{W}_l \boldsymbol{h}_u^{(l)}$

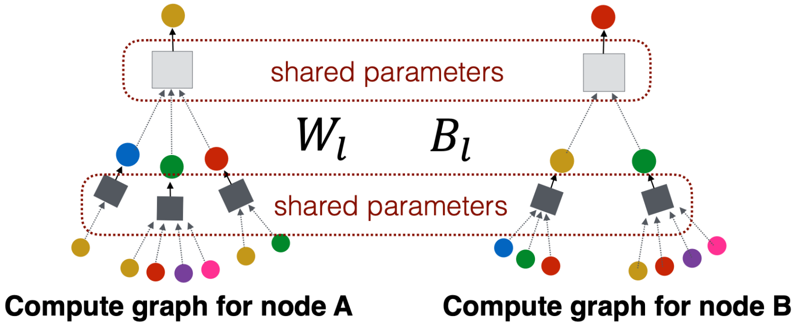
      $= \sum_{u\in N(v)} \boldsymbol{m}_u^{(l)}$

Inductive capability of GNN

- **Inductive learning**: We can obtain embeddings for nodes that have not appeared in the training time
  - e.g., In Amazon, new users are consistently added to the system, and it is impractical to re-train the system to get the embeddings for the new users



- This is possible because we do not train an embedding matrix as done in Deepwalk/node2vec
- Instead, we train **aggregator** and **transformer**



Background: Mutual Information (MI)

- Measures the amount of information that two variables share
- If X and Y are independent, then $P_{XY} = P_X P_Y$ → in this case, MI = 0

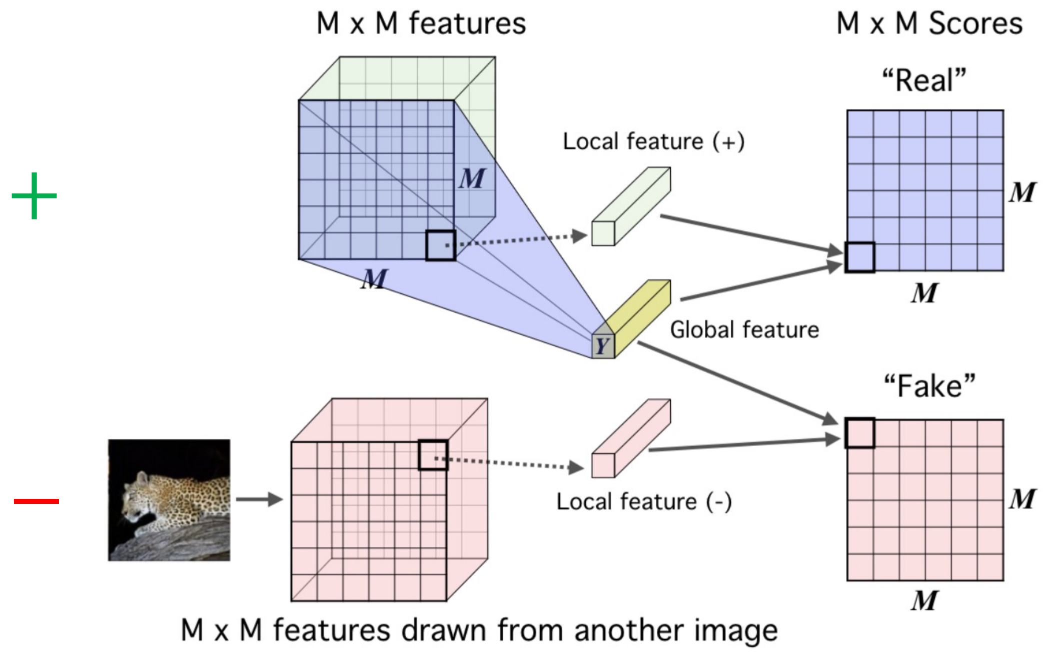
$$\begin{aligned} I(X;Y) &= \mathbb{E}_{P_{XY}}\left[\log \frac{P_{XY}}{P_X P_Y}\right] \\ &= D_{KL}(P_{XY} || P_X P_Y) \end{aligned}$$

- High MI? → One variable is always indicative of the other variable
- Recently, scalable estimation of mutual information was made both possible and practical through **Mutual Information Neural Estimation** (MINE) [1]

Belghazi, Mohamed Ishmael, et al. "Mutual information neural estimation." ICML 2018

Background: Deep infomax

- Unsupervised representation learning method for image data
- Intuition: **Maximize mutual information (MI)** between local patches and the global representation of an image



Discriminator tries to discriminate between "Real" and "Fake"

Deep Infomax (Hjelm et al, 2019)

Hjelm, R. Devon, et al. "Learning deep representations by mutual information estimation and maximization." ICLR 2019

Deep Graph Infomax (DGI) (1/2)

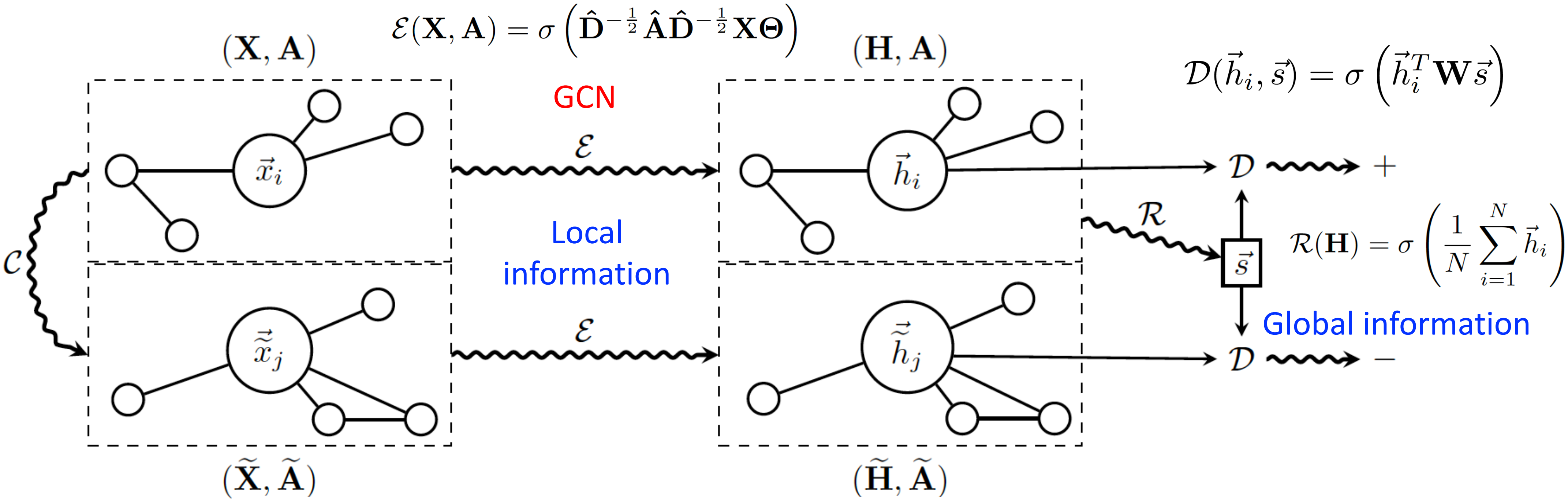
- Deep Graph Infomax (DGI) applies Deep Infomax on graph domain
- Unsupervised graph representation learning method that considers node features
- Notations
  - $X = \{\vec{x}_1, \vec{x}_2, \dots, \vec{x}_N\}$: A set of node features (N: num. nodes)
  - $A \in R^{N \times N}$: Adjacency matrix
- Learn a **GCN** encoder $\mathcal{E}(\mathbf{X}, \mathbf{A}) = \mathbf{H} = \{\vec{h}_1, \vec{h}_2, \dots, \vec{h}_N\}$ $\vec{h}_i \in \mathbb{R}^{F'}$
  - Generates node representations by **repeated aggregation over local node neighborhoods**
  - $\vec{h}_i$ summarizes a patch of the graph centered around node i (≈ patch representation)

Analogy: Local patch representation in an image == Node representation in a graph

Velickovic, Petar, et al. "Deep Graph Infomax." ICLR 2019

Deep Graph Infomax (DGI) (2/2)

- Unsupervised training of GNN considering the local and global structure of the graph



$$\mathcal{E}(\mathbf{X},\mathbf{A}) = \sigma\left(\hat{\mathbf{D}}^{-\frac{1}{2}}\hat{\mathbf{A}}\hat{\mathbf{D}}^{-\frac{1}{2}}\mathbf{X}\boldsymbol{\Theta}\right)$$

$$\mathcal{D}(\vec{h}_i,\vec{s}) = \sigma\left(\vec{h}_i^T\mathbf{W}\vec{s}\right)$$

$$\mathcal{R}(\mathbf{H}) = \sigma\left(\frac{1}{N}\sum_{i=1}^{N}\vec{h}_i\right)$$

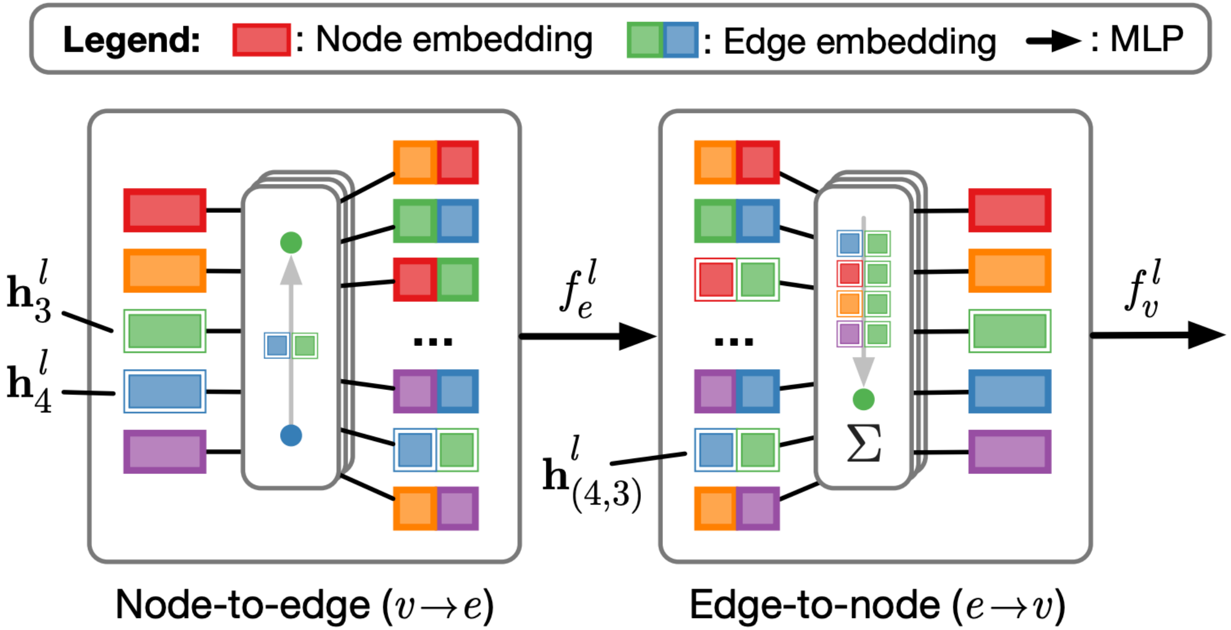
$$\mathcal{L} = \frac{1}{N+M}\left(\sum_{i=1}^{N}\mathbb{E}_{(\mathbf{X},\mathbf{A})}\left[\log\mathcal{D}\left(\vec{h}_i,\vec{s}\right)\right] + \sum_{j=1}^{M}\mathbb{E}_{(\widetilde{\mathbf{X}},\widetilde{\mathbf{A}})}\left[\log\left(1-\mathcal{D}\left(\vec{\widetilde{h}}_j,\vec{s}\right)\right)\right]\right)$$

Maximizes the mutual information between the local patches and the graph-level global representation

Velickovic, Petar, et al. "Deep Graph Infomax." ICLR 2019 https://www.youtube.com/watch?v=mdWQYYapvR8

GNNs with edge embeddings

- **Motivation**: Edges may also contain attributes
  - e.g., contents of papers written together by two authors



Node-to-edge ($v \rightarrow e$)

Edge-to-node ($e \rightarrow v$)

$$v \rightarrow e: \quad \mathbf{h}_{(i,j)}^l = f_e^l([\mathbf{h}_i^l, \mathbf{h}_j^l, \mathbf{x}_{(i,j)}])$$

$$e \rightarrow v: \quad \mathbf{h}_j^{l+1} = f_v^l([\textstyle\sum_{i \in \mathcal{N}_j} \mathbf{h}_{(i,j)}^l, \mathbf{x}_j])$$

Gilmer, Justin, et al. "Neural message passing for quantum chemistry." ICML 2017

Kipf, Thomas, et al. "Neural relational inference for interacting systems." ICML 2018

Gong, Liyu, and Qiang Cheng. "Exploiting edge features for graph neural networks." CVPR 2019.

Conclusion

- Overview
- Node-level Network Embedding
  - Random walk-based Node Embeddings (DeepWalk / node2vec)
- Graph Neural Network (GNN)
  - Graph Convolutional Neural Network (GCN)
  - Graph Attention Network (GAT)
  - GraphSAGE
  - Deep Graph Infomax (DGI)
  - GNNs with Edge Embeddings

소재 물성 예측 연구

15:00-16:00

Table of Contents

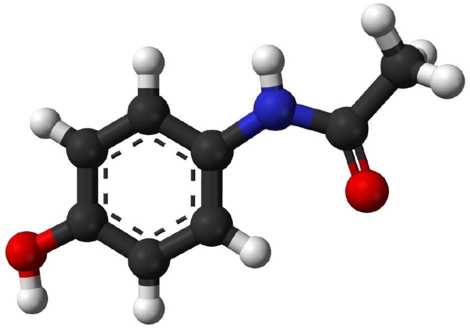
시간	주제	주요 내용
1	그래프 신경망 개요	- 그래프 신경망 전반적인 소개 - 그래프 종류에 따른 다양한 그래프 신경망 소개
2	소재 물성 예측 연구	- 소재 물성 예측 연구 최신 동향 소개 - Transformer 기반 모델 소개 및 extrapolation을 위한 모델 소개
3	물질 간 화학 반응 예측 연구	- 물질 간 화학 반응 예측 연구 동향 소개 - 정보 이론(Information bottleneck) 및 인과추론(Causal inference) 기반 모델 소개

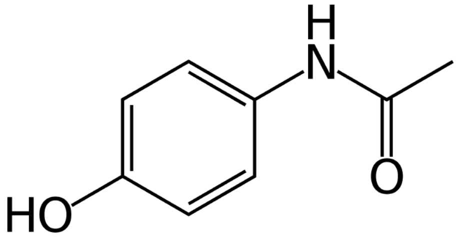
Papers

- 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
  - Graph convolutional neural networks with global attention for improved materials property prediction. Physical Chemistry Chemical Physics 2020
  - Direct prediction of phonon density of states with Euclidean neural networks. Advanced Science 2021
  - **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**

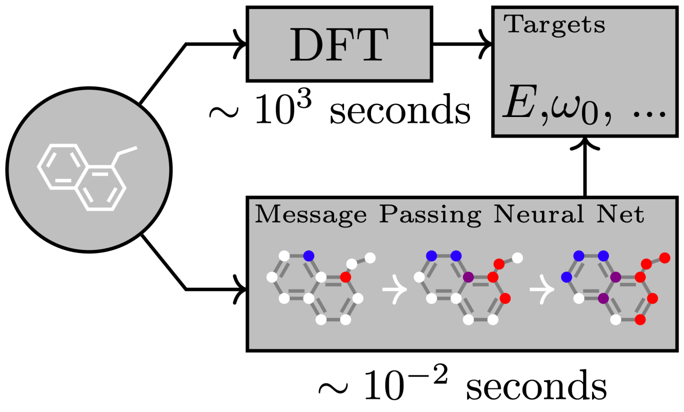
Molecular Graphs

- Molecules can be represented as a graph with node features and edge features
  - Node features: atom type, atom charges…
  - Edge features: valence bond type…
  - However, sometimes, we also know the **3D positions $\boldsymbol{x_i}$**, which is actually more **informative**

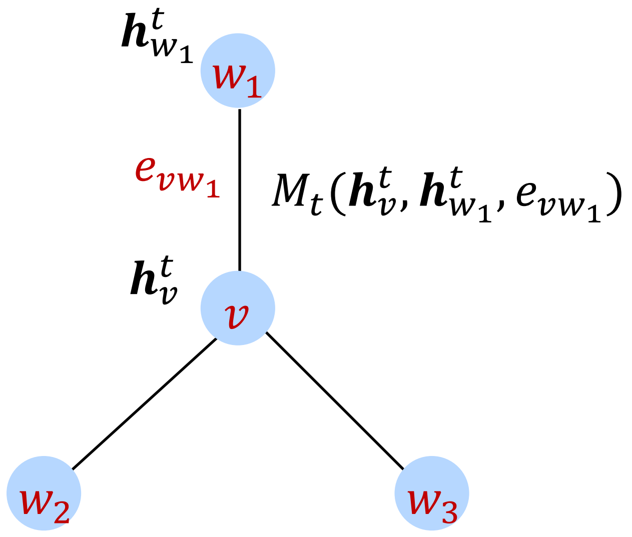




Message Passing Neural Network



- Unified various graph neural network and graph convolutional network approaches



Edge embedding

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

Neighbor of v

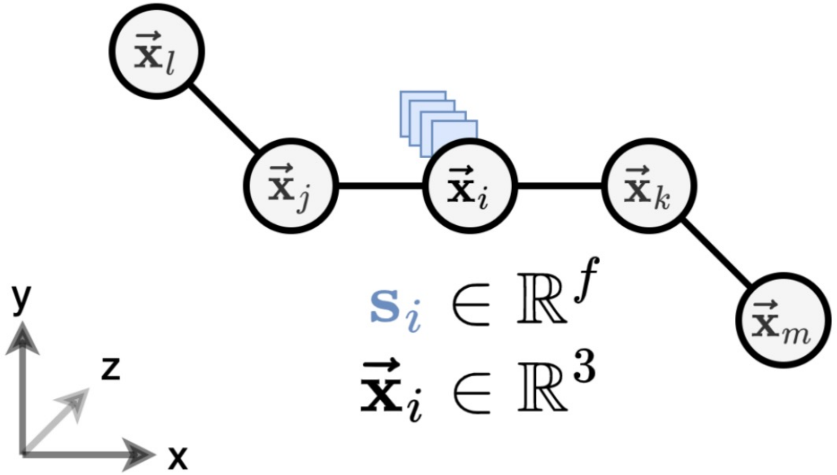
$$h_v^{t+1} = U_t(h_v^t, m_v^{t+1})$$

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

Neural message passing for quantum chemistry. ICML 2017

Geometric Graphs

- A geometric graph $G = (A, S, X)$ is a graph where each node is embedded in d-dimensional **Euclidean space**:

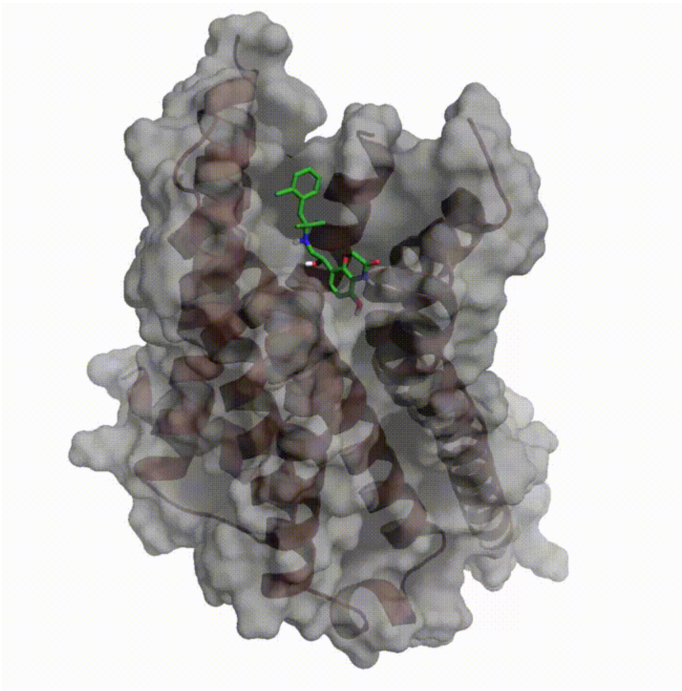


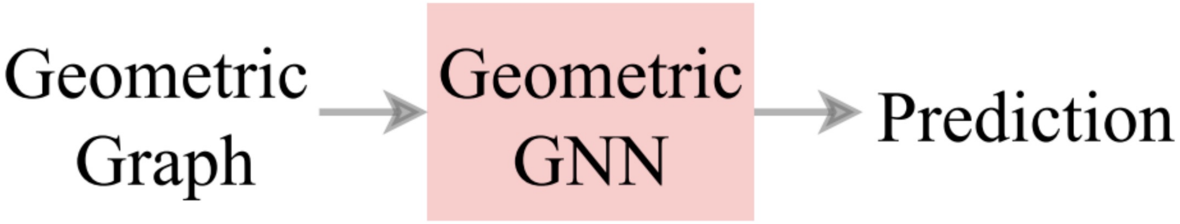
  - A: an $n \times n$ adjacency matrix
  - $S \in R^{n \times f}$: Scalar features (atom type, atom charges, …)
  - $X \in R^{n \times d}$: tensor features, e.g., coordinates

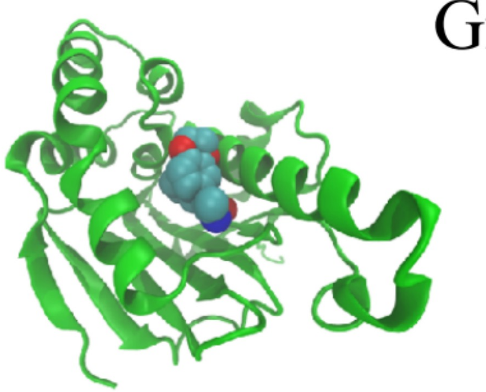
http://web.stanford.edu/class/cs224w/slides/17-geometric.pdf

Broad Impact on Sciences

- Supervised Learning: Prediction
  - Properties prediction
  - 3D Protein-ligand interaction (binding)





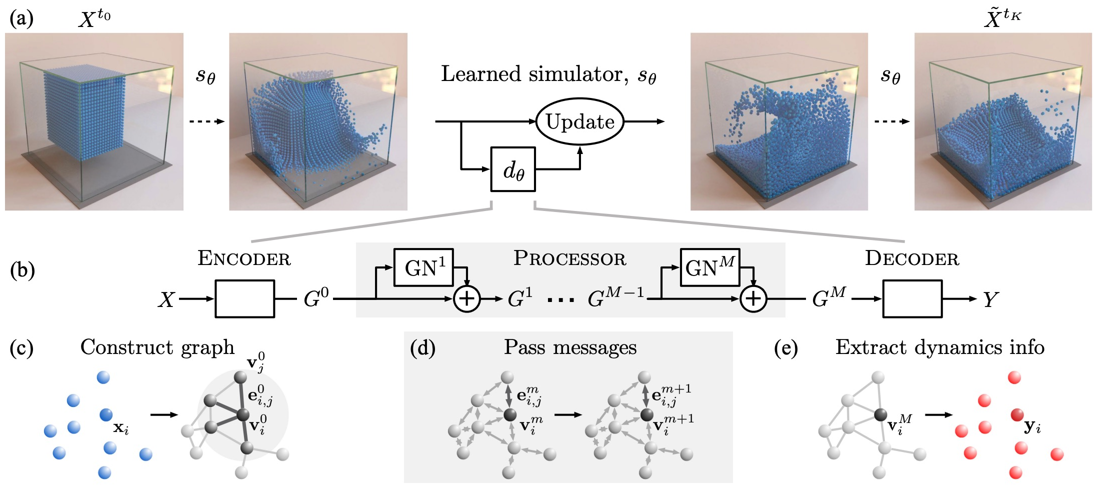


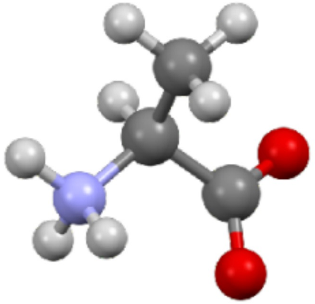
- Functional properties?
- Ligand binding affinity?
- Ligand efficacy?

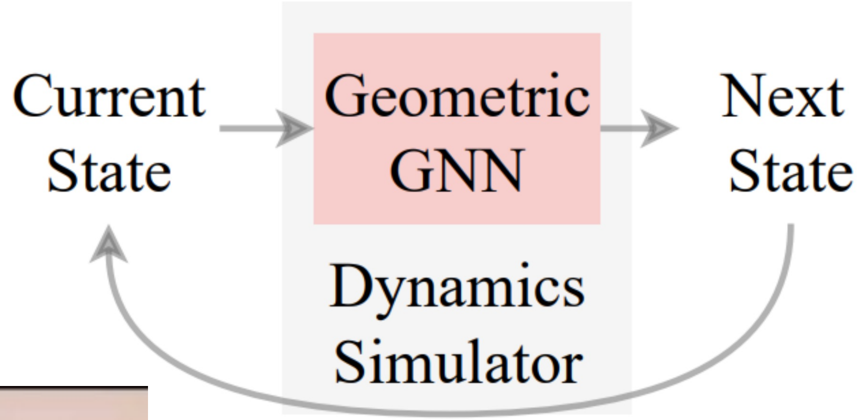
http://web.stanford.edu/class/cs224w/slides/17-geometric.pdf

Broad Impact on Sciences

- Supervised Learning: Structured Prediction
  - Molecular Simulation





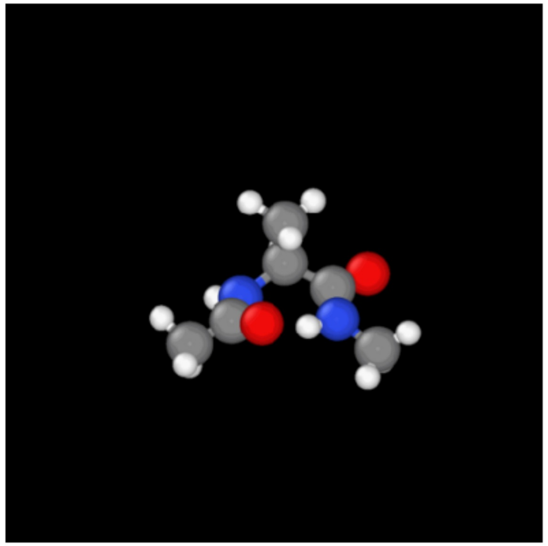


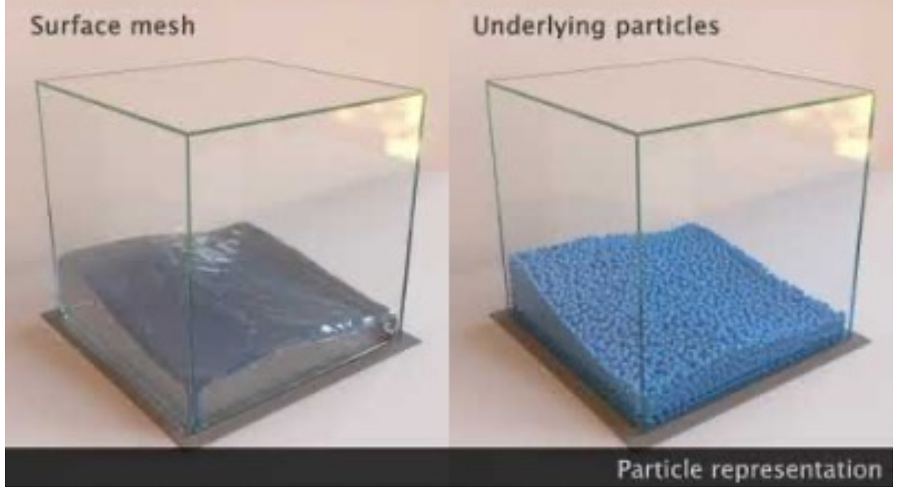
Current State

Geometric GNN

Dynamics Simulator

Next State

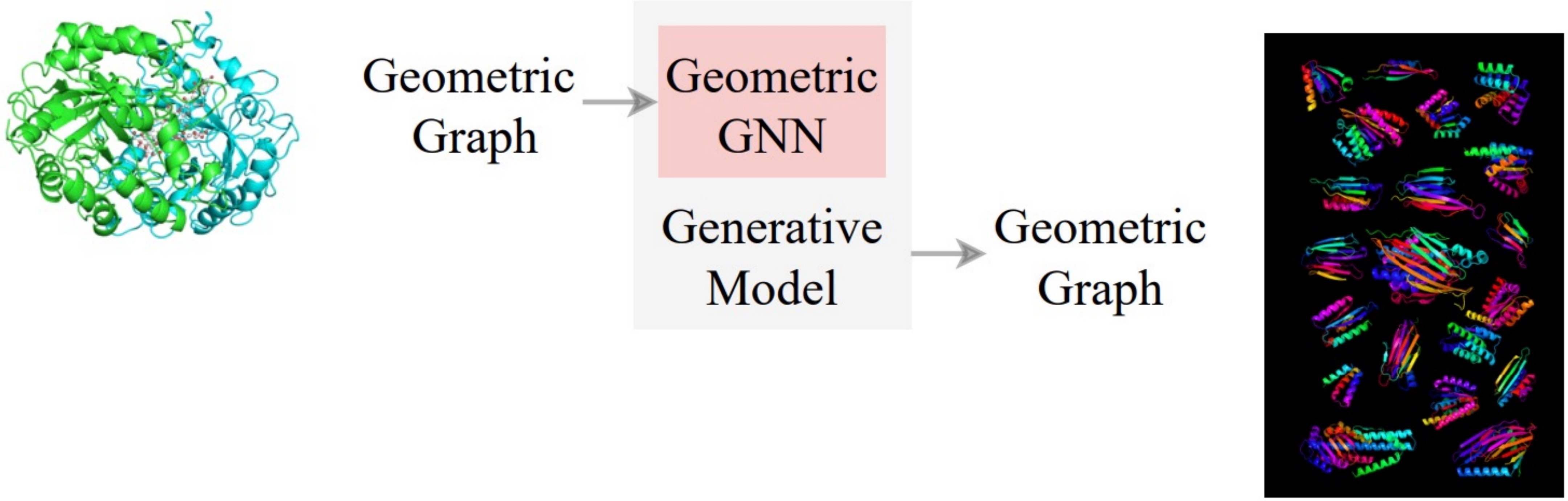




http://web.stanford.edu/class/cs224w/slides/17-geometric.pdf

Broad Impact on Sciences

- Generative Models
  - Drug or material design

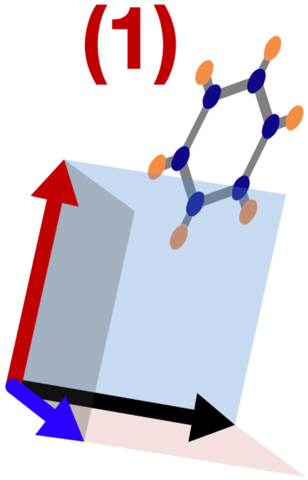


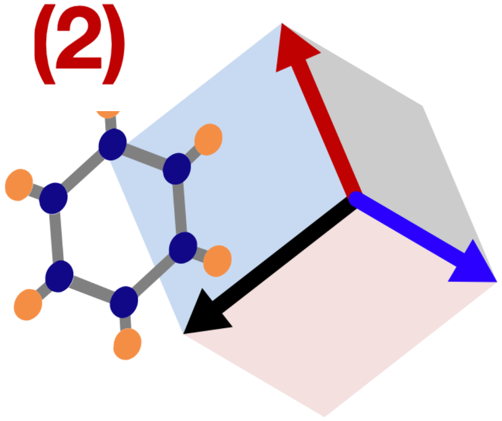
Geometric Graph → Geometric GNN / Generative Model → Geometric Graph

http://web.stanford.edu/class/cs224w/slides/17-geometric.pdf

Why is it hard?

- 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)

http://web.stanford.edu/class/cs224w/slides/17-geometric.pdf

Schnet: Overview

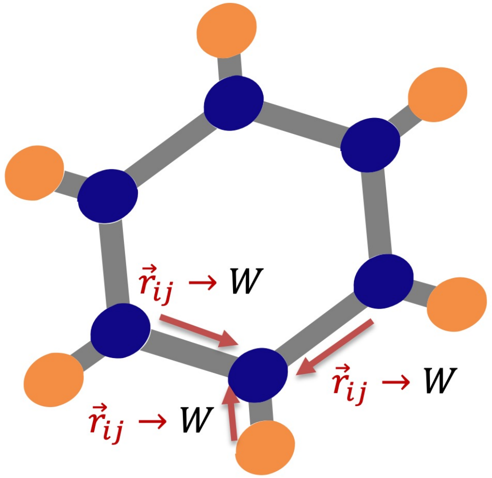
- **Input**
  - Feature representations of n objects $X^l = (\boldsymbol{x}_1^l, \ldots, \boldsymbol{x}_n^l)$ with $\boldsymbol{x}_i^l \in R^F$
  - At locations $R = (\boldsymbol{r}_1, \ldots, \boldsymbol{r}_n)$ with $\boldsymbol{r}_i \in R^D$ ($D = 3$ for 3-dim coordinates)
- **Output**
  - Molecular total energy $E(\boldsymbol{r}_1, \ldots, \boldsymbol{r}_n)$
  - Forces $F = (\boldsymbol{r}_1, \ldots, \boldsymbol{r}_n)$ acting on each atom

$$\mathbf{F}_i(\mathbf{r}_1, \ldots, \mathbf{r}_n) = -\frac{\partial E}{\partial \mathbf{r}_i}(\mathbf{r}_1, \ldots, \mathbf{r}_n).$$

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



x^l: node embeddings at l layer
r: atomic coordinates

SchNet: A continuous-filter convolutional neural network for modeling quantum interactions, NeurIPS 2017

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

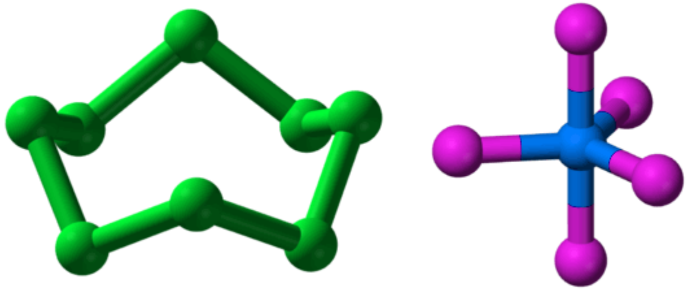
→ Node embeddings are invariant!

- Since d_{ij} is 1-dimensional, we expand to a higher dimension (i.e., 300-dim) via radial basis function

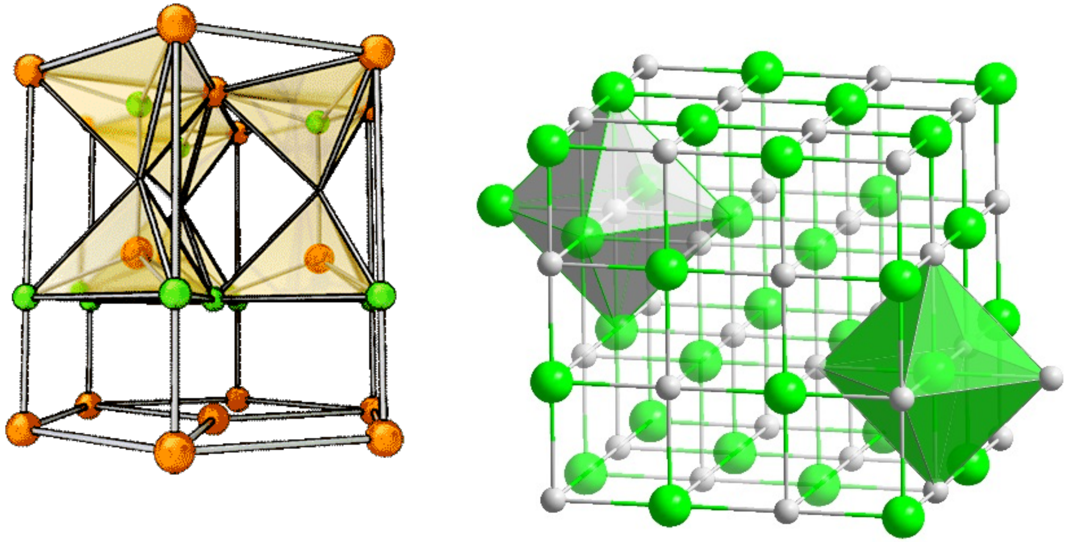
$$e_k(\mathbf{r}_i - \mathbf{r}_j) = \exp(-\gamma\|d_{ij} - \mu_k\|^2)$$

  - μ_k is chosen every 0.1A within $0A \leq \mu_k \leq 30A$ and $\gamma = 10A$

Crystalline Materials

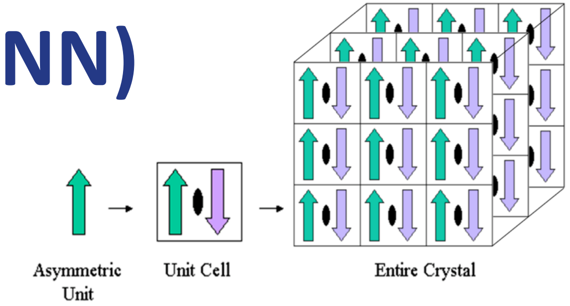


Molecules



Crystals

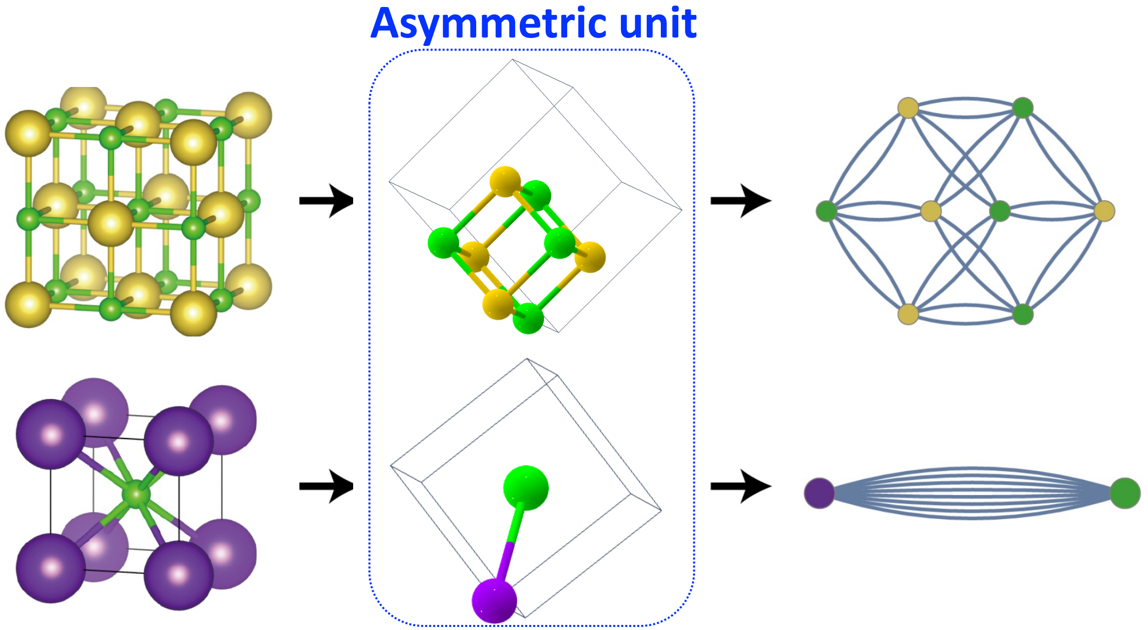
Crystal Graph Convolutional Neural Networks (CGCNN)

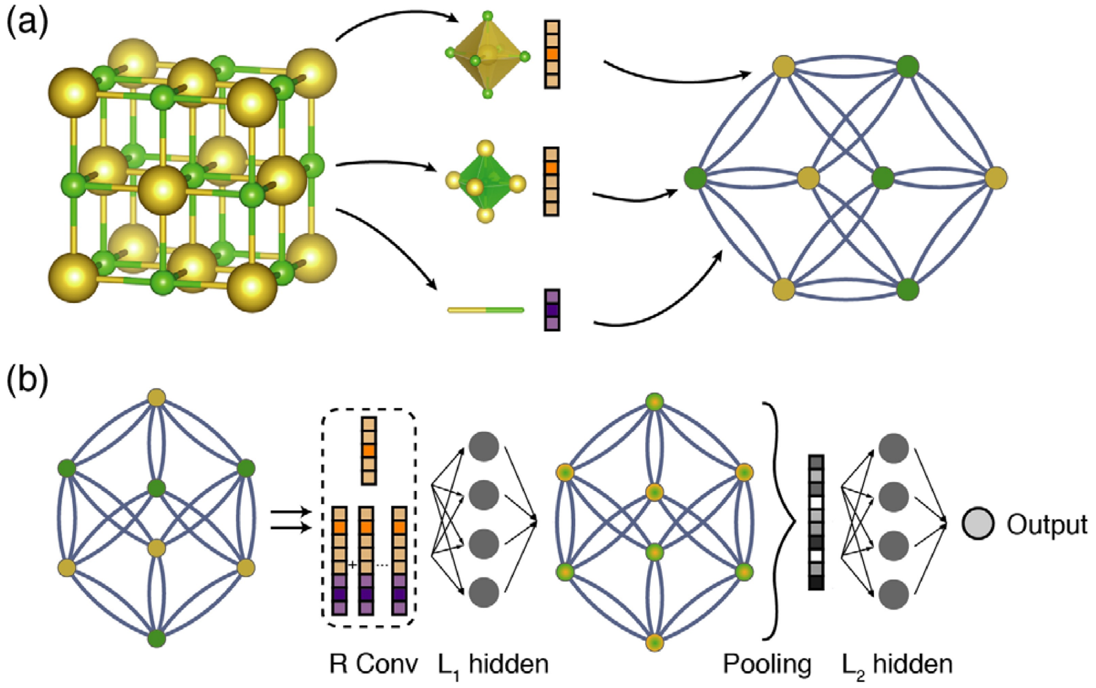


- Goal: Predict material properties of periodic **crystal systems**
- Idea: Represent the crystal structure by a **crystal graph** that encodes both atomic information and bonding interactions between atoms (Distance between atoms → Edges in a crystal graph)

Undirected multigraph

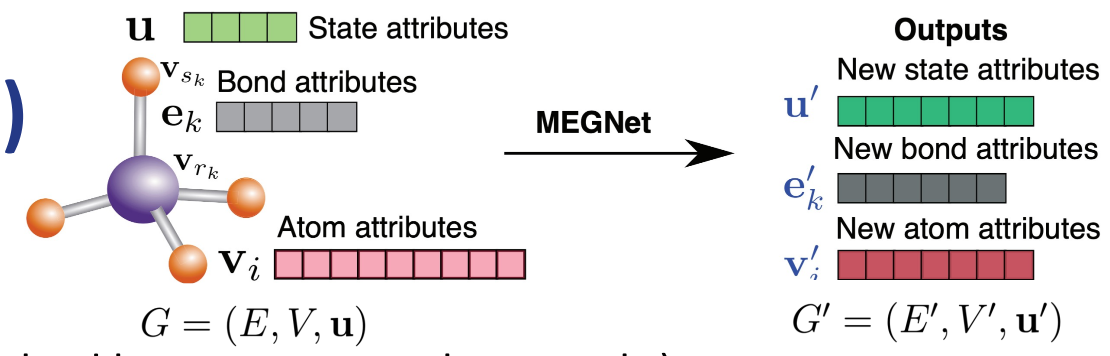
- Multiple edges between the same pair of nodes
- Considers lattice periodicity





Crystal Graph Convolutional Neural Networks for an Accurate and Interpretable Prediction of Material Properties, Phys. Rev. Lett. 2018

MatErials Graph Network (MEGNet)



- **Motivation**
  - 1) Existing work either on molecular and crystal datasets (solved by adopting graph networks)
  - 2) Global state (e.g., temperature) of each molecule/crystal is overlooked
    - Important for predicting state-dependent properties such as the free energy
- Considers topological distance and spatial distance (Manual features)

$$\mathbf{e}'_k = \phi_e\left(\mathbf{v}_{s_k} \bigoplus \mathbf{v}_{r_k} \bigoplus \mathbf{e}_k \bigoplus \mathbf{u}\right)$$

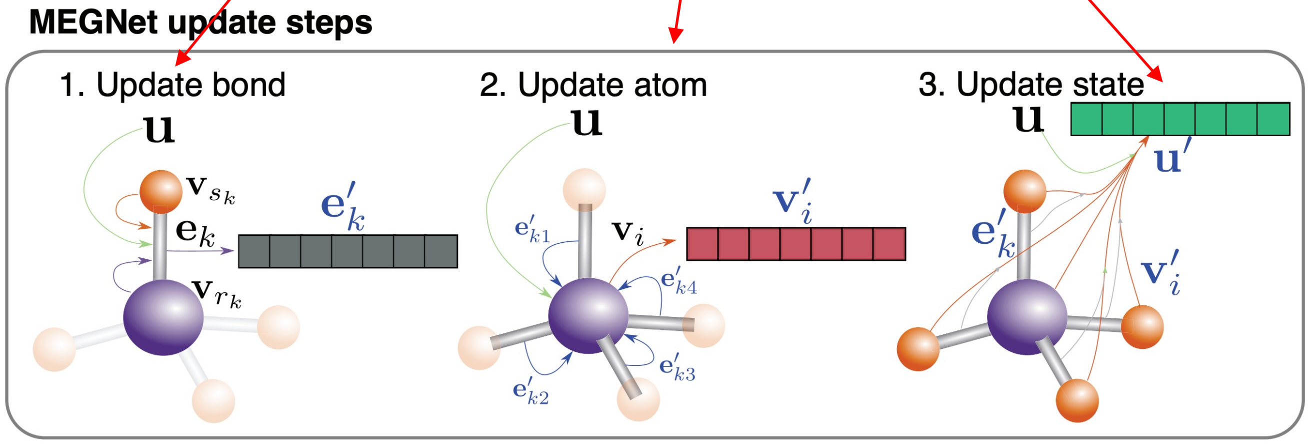
$$\bar{\mathbf{v}}_i^e = \frac{1}{N_i^e}\sum_{k=1}^{N_i^e}\{\mathbf{e}'_k\}_{r_k=i}$$

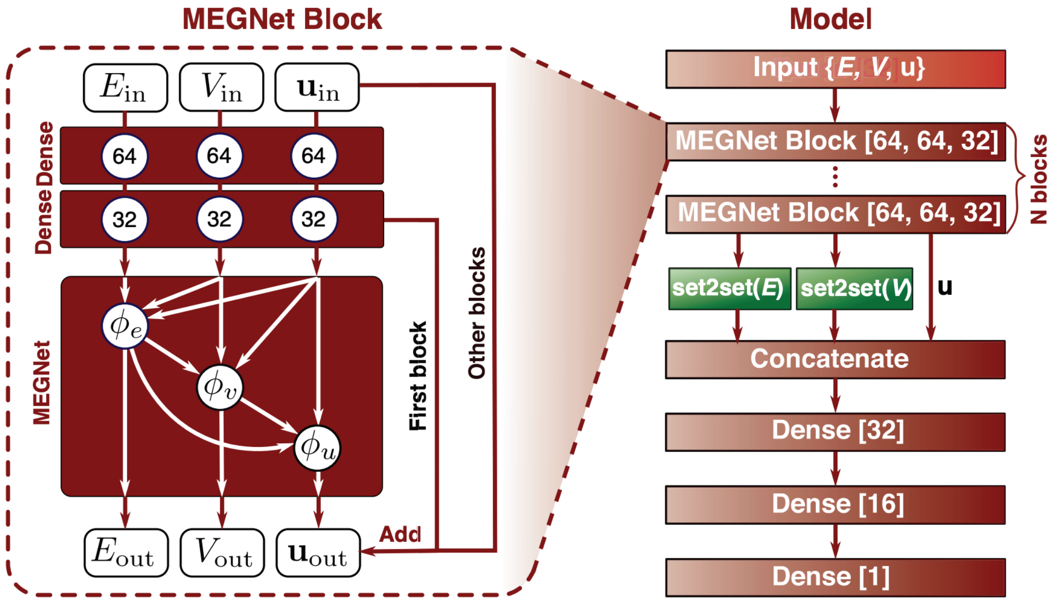
$$\mathbf{v}'_i = \phi_v\left(\bar{\mathbf{v}}_i^e \bigoplus \mathbf{v}_i \bigoplus \mathbf{u}\right)$$

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

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

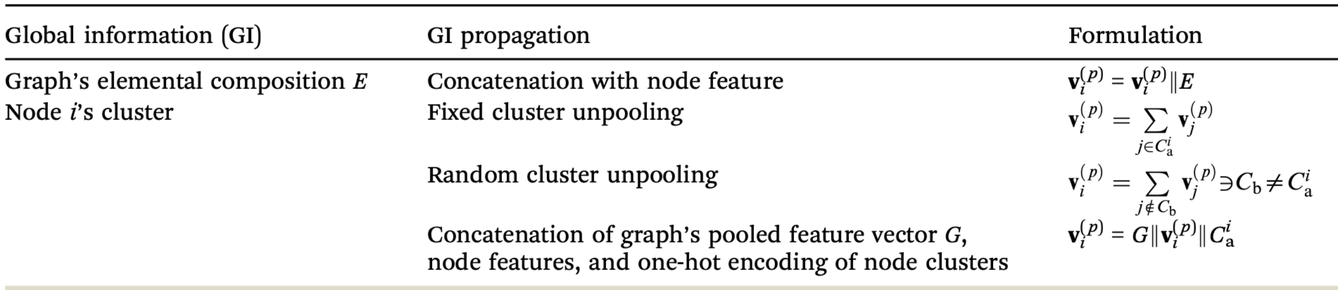
$$\mathbf{u}' = \phi_u\left(\bar{\mathbf{u}}^e \bigoplus \bar{\mathbf{u}}^v \bigoplus \mathbf{u}\right)$$





Graph networks as a universal machine learning framework for molecules and crystals. Chem. Mater. 2019

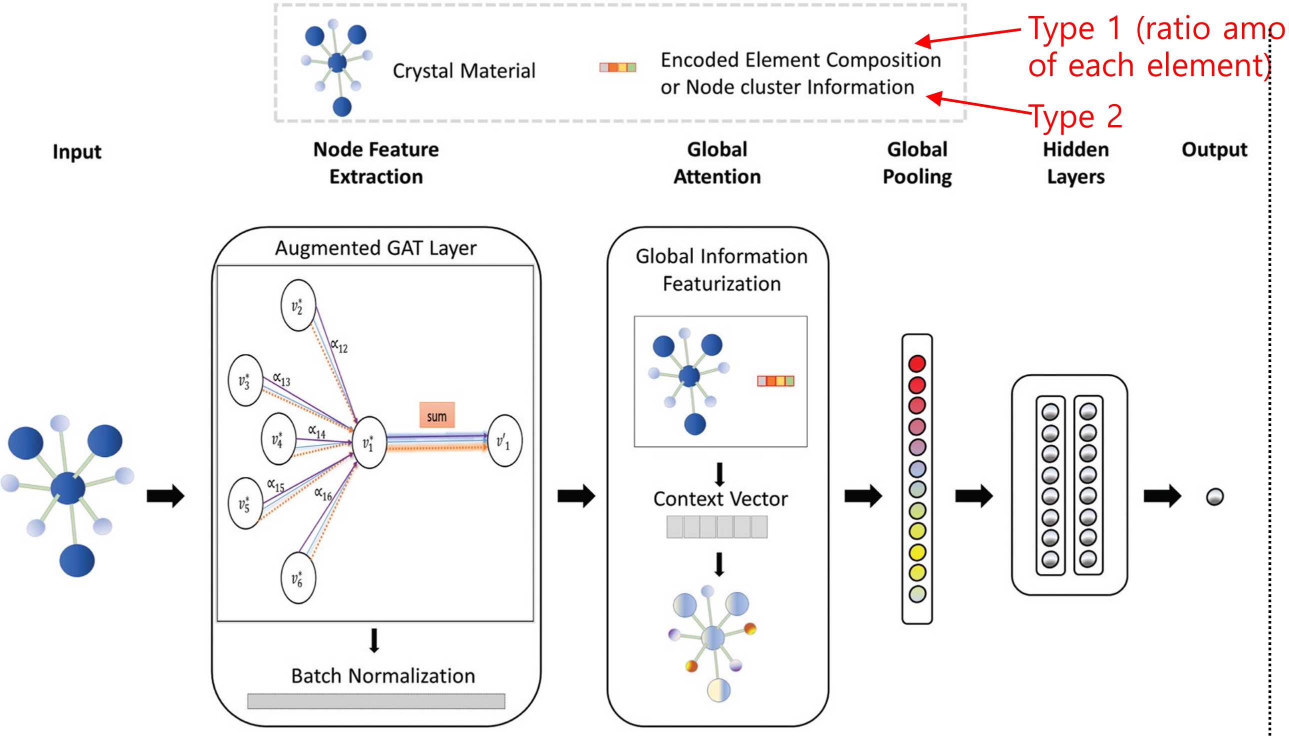
GATGNN



Global information (GI)	GI propagation	Formulation
Graph's elemental composition E	Concatenation with node feature	$\mathbf{v}_i^{(p)} = \mathbf{v}_i^{(p)} \Vert E$
Node i's cluster	Fixed cluster unpooling	$\mathbf{v}_i^{(p)} = \sum_{j\in C_a^i} \mathbf{v}_j^{(p)}$
	Random cluster unpooling	$\mathbf{v}_i^{(p)} = \sum_{j\notin C_b} \mathbf{v}_j^{(p)} \ni C_b \neq C_a^i$
	Concatenation of graph's pooled feature vector G, node features, and one-hot encoding of node clusters	$\mathbf{v}_i^{(p)} = G \Vert \mathbf{v}_i^{(p)} \Vert C_a^i$

- Motivation: Existing studies do not effectively **differentiate the contributions from different atoms**
  - Existing studies only emphasized on capturing local atomic environment

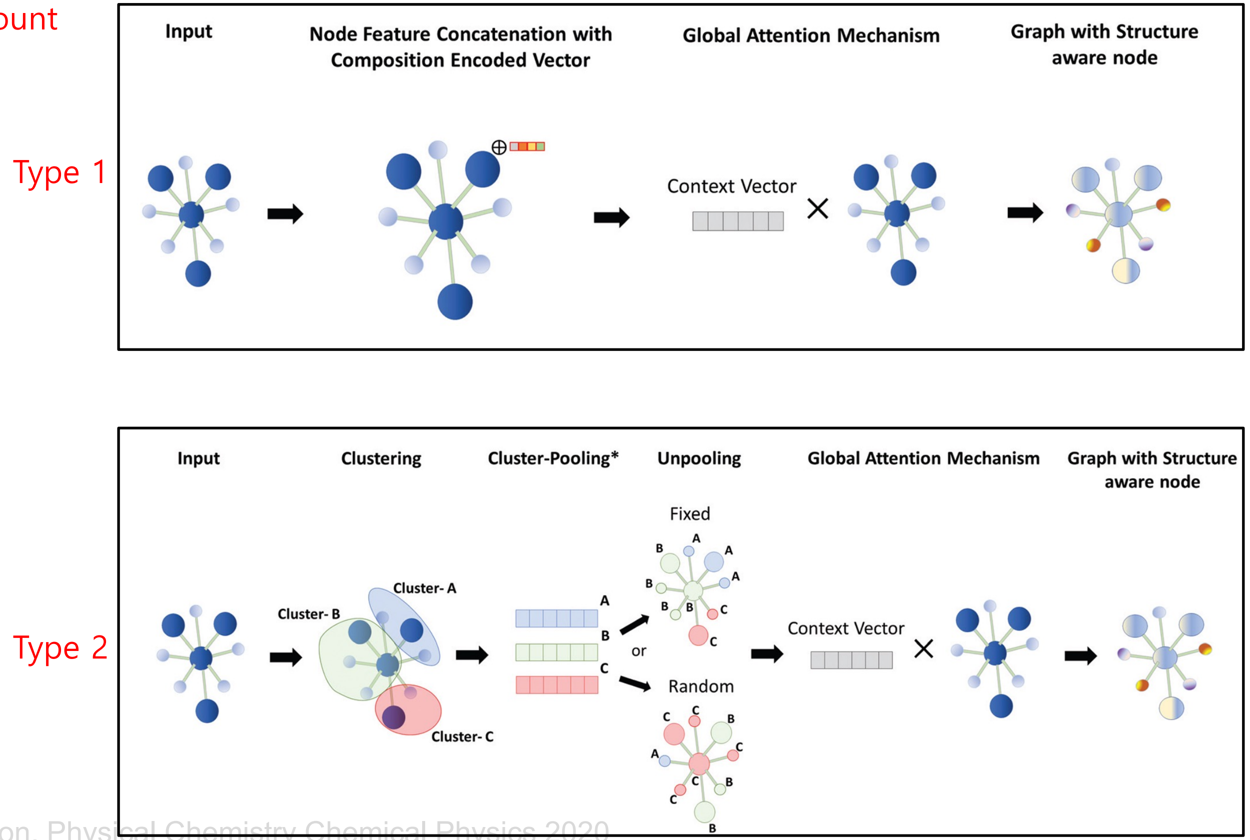
- Idea: **Global attention** (Global information – i) Entire crystal graph, ii) Crystal node's location in graph)



Type 1 (ratio amount of each element)

Type 2

GAT + edge information


Type 1
Input
Node Feature Concatenation with Composition Encoded Vector
Global Attention Mechanism
Graph with Structure aware node
Context Vector
×
Type 2
Input
Clustering
Cluster-Pooling*
Unpooling
Global Attention Mechanism
Graph with Structure aware node
Cluster- A
Cluster- B
Cluster- C
A
B
C
Fixed
or
Random
Context Vector
×

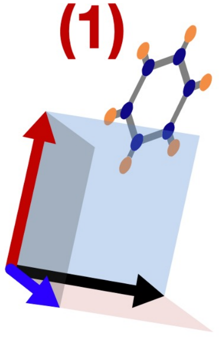
Graph convolutional neural networks with global attention for improved materials property prediction. Physical Chemistry Chemical Physics 2020

Symmetry

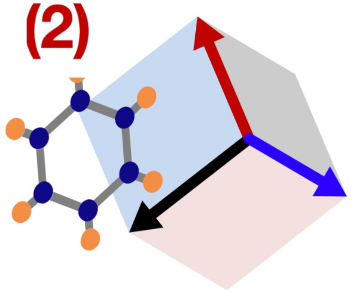
- Symmetry of **Inputs**
  - We want our GNNs to see (1) and (2) as the same system though described differently

    → Symmetry-aware GNNs

(1)

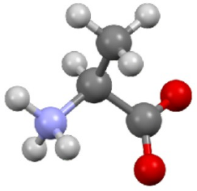


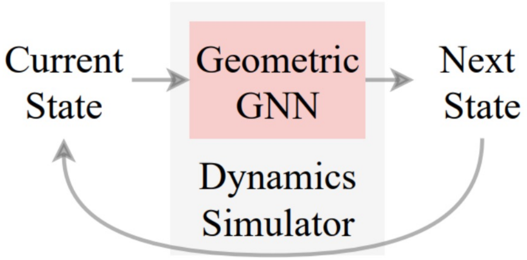
(2)

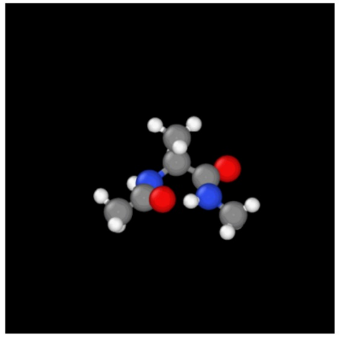


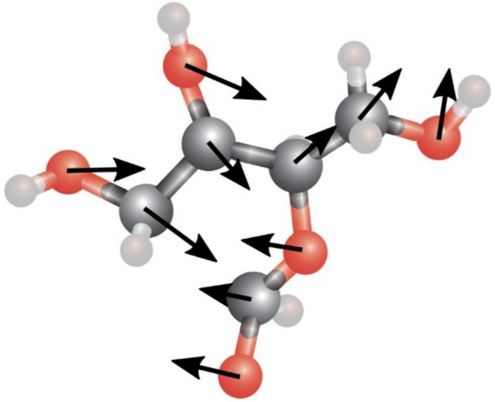
- Symmetry of **Outputs**
  - Beyond input space, output can also be tensors
  - Example: simulation (force prediction)
    - Given a molecule and a rotated copy, predicted forces should be the same up to rotation
    - i.e., Predicted forces are equivariant to rotation

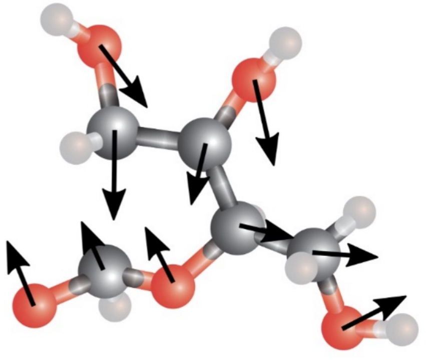
$$\mathbf{F}_i(\mathbf{r}_1, \ldots, \mathbf{r}_n) = -\frac{\partial E}{\partial \mathbf{r}_i}(\mathbf{r}_1, \ldots, \mathbf{r}_n).$$









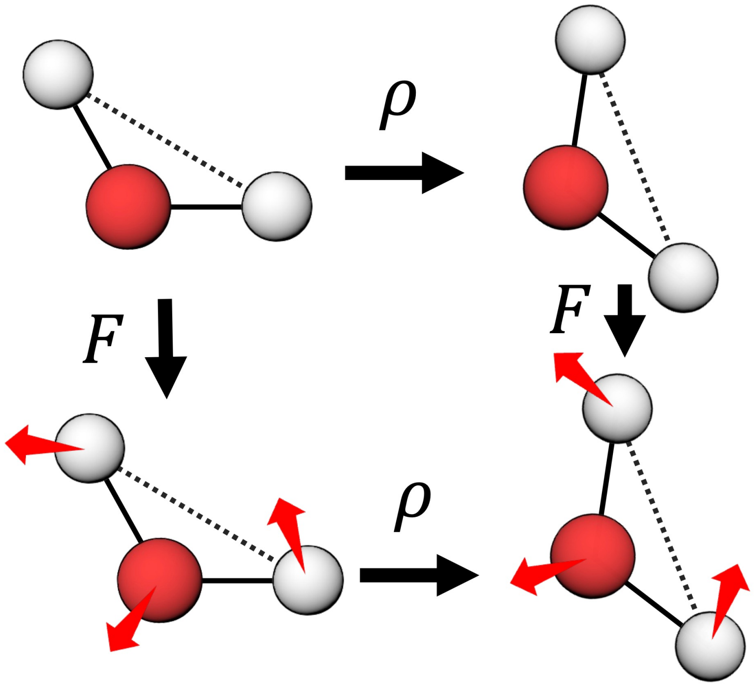


Equivariance

- Formal definition of Equivariance
  - A function $F: X \rightarrow Y$ is equivariant, if for a transformation ρ it satisfies:

$$F \circ \rho(x) = \rho \circ F(x)$$

  - The equation says that applying ρ on the input has the same effect as applying it to the output.



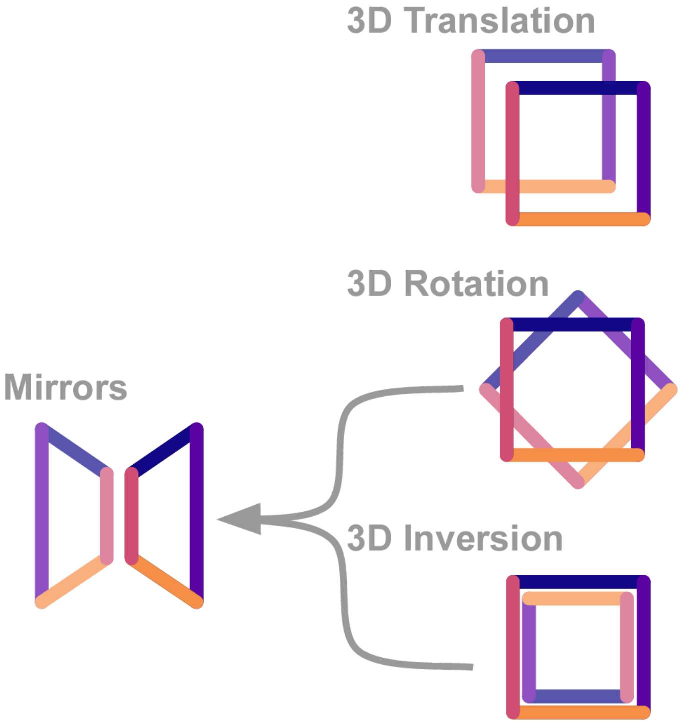
Definition of Invariance

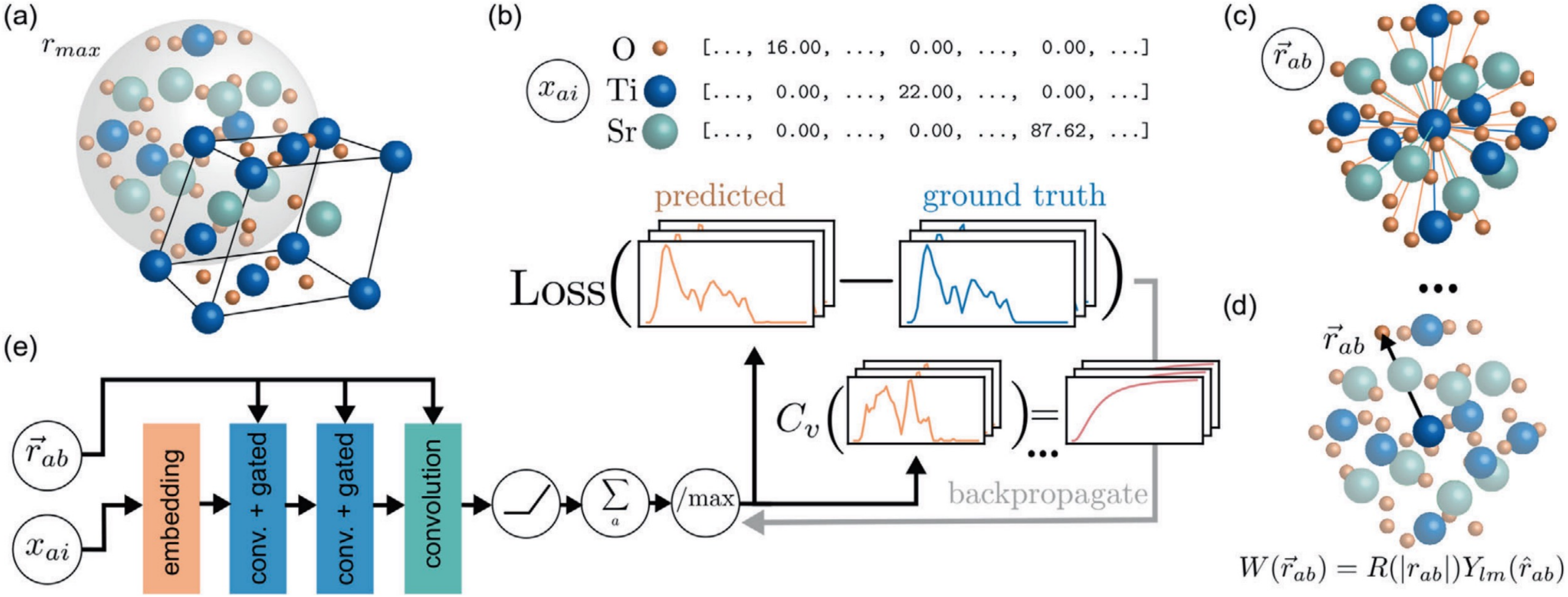
- A function $F: X \rightarrow Y$ is invariant if for a transformation ρ it satisfies:

$$F \circ \rho(x) = F(x)$$

E(3)NN

- Motivation: Can we preserve all geometric information of the input?
- Idea: Apply **Euclidean neural networks** (Capture full crystal symmetry)
  - Equivariant to 3D rotations, translations, and inversion
    - (b) Node feature: Mass weighted one-hot encoding
    - (c) Edge feature: Distance between atoms





Direct prediction of phonon density of states with Euclidean neural networks. Advanced Science 2021

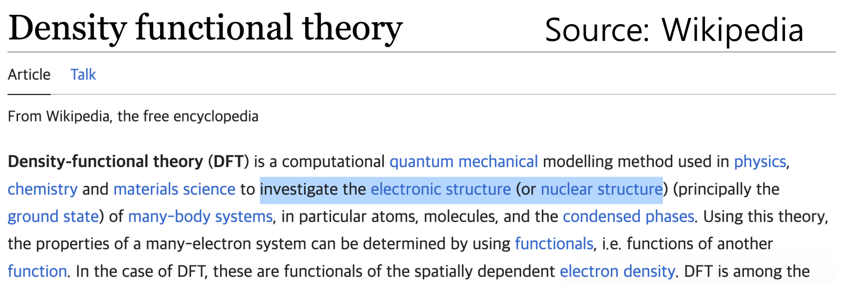
Table of Contents

시간	주제	주요 내용
1	그래프 신경망 개요	- 그래프 신경망 전반적인 소개 - 그래프 종류에 따른 다양한 그래프 신경망 소개
2	소재 물성 예측 연구	- 소재 물성 예측 연구 최신 동향 소개 - Transformer 기반 모델 소개 및 extrapolation을 위한 모델 소개
3	물질 간 화학 반응 예측 연구	- 물질 간 화학 반응 예측 연구 동향 소개 - 정보 이론(Information bottleneck) 및 인과추론(Causal inference) 기반 모델 소개

Density of States Prediction of Crystalline Materials via Prompt-guided Multi-Modal Transformer

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

Density Functional Theory (DFT)



Density functional theory

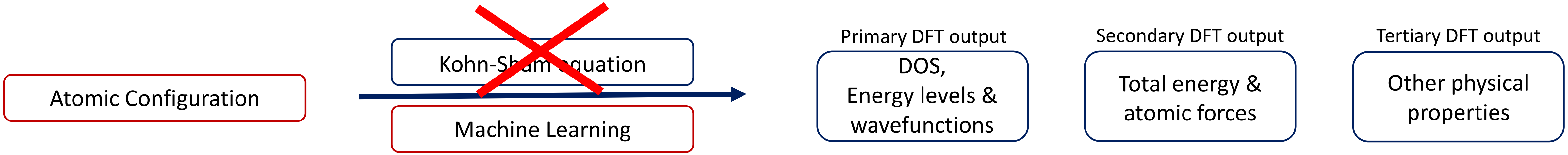
Source: Wikipedia

Article Talk

From Wikipedia, the free encyclopedia

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

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



Atomic Configuration

~~Kohn-Sham equation~~

Machine Learning

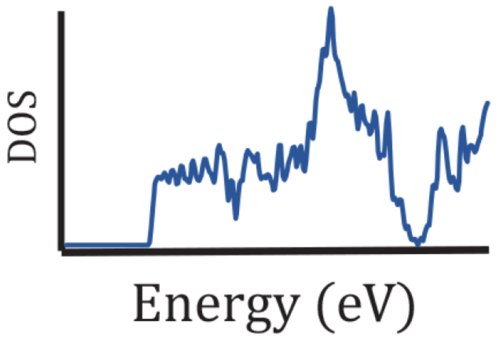
Primary DFT output: DOS, Energy levels & wavefunctions

Secondary DFT output: Total energy & atomic forces

Tertiary DFT output: Other physical properties

- However, it is difficult and computationally expensive to compute DFT outputs based on Kohn-Sham equation

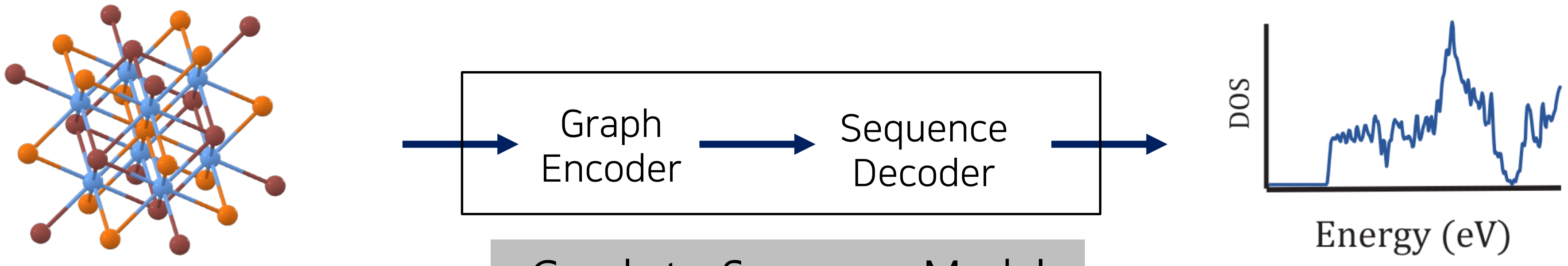
- In this work, we adopt GNNs to approximate Kohn-Sham Equation to predict DOS



Main assumption: DOS is related to a sequence of energy

Consider DOS as a sequence

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



Graph Encoder

Sequence Decoder

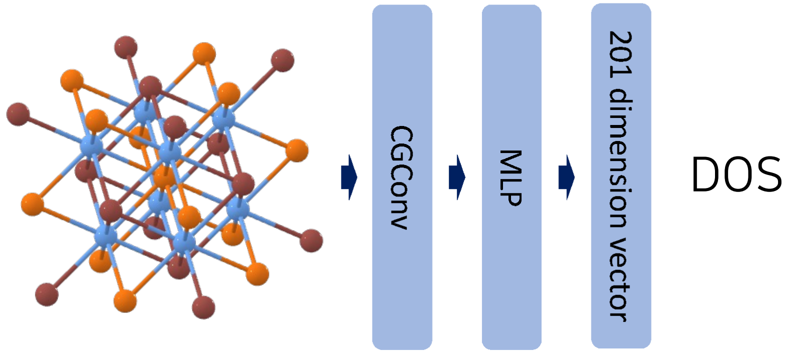
Graph-to-Sequence Model

Input : Crystal Structure

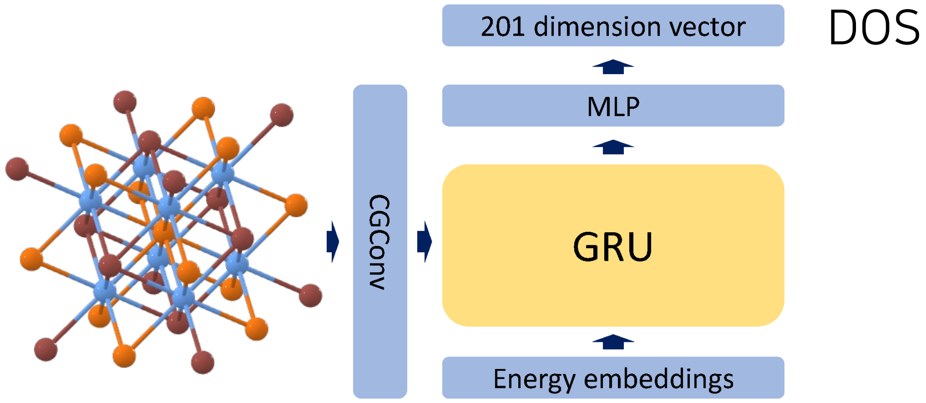
Output : Sequence

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)

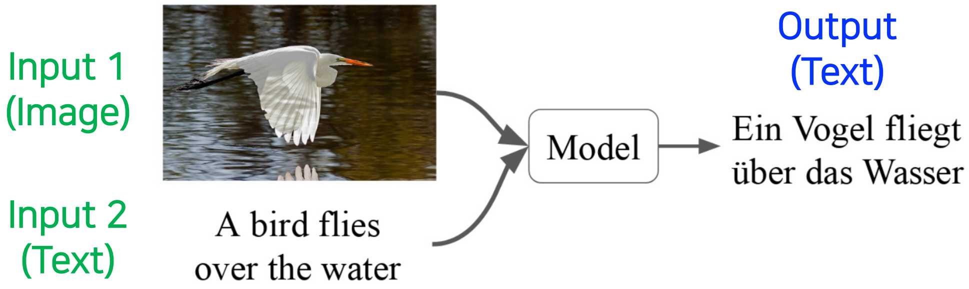
- Modality: Graph ≠ Energy

[1] XIE, Tian; GROSSMAN, Jeffrey C. Crystal graph convolutional neural networks for an accurate and interpretable prediction of material properties. Physical review letters, 2018, 120.14: 145301.

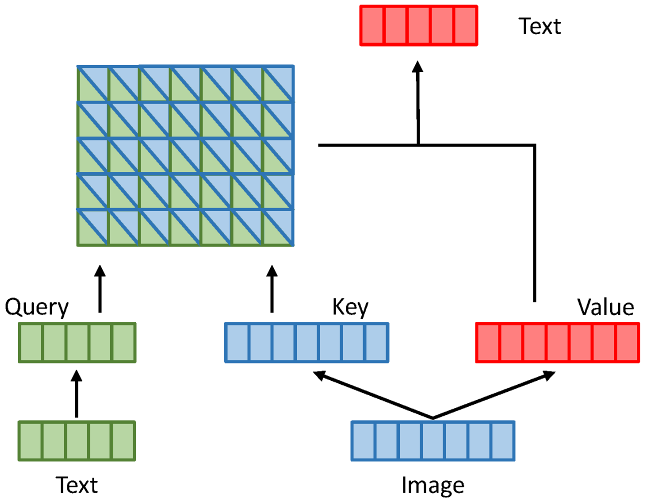
Multimodal transformer

- How can we perform machine translation given both image and text data?
  - Multi-modal Machine Translation

  → Multi-modal Transformer



Multi-modal Machine Translation



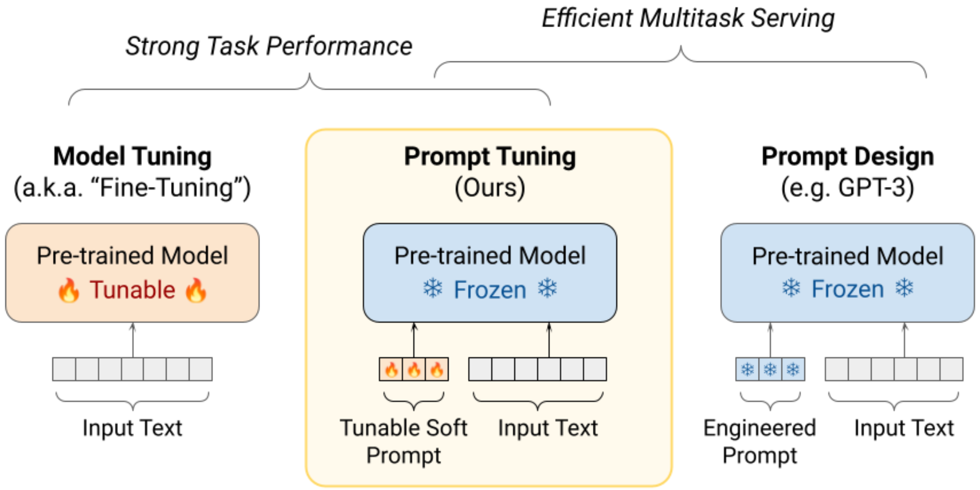
Multi-modal Transformer

- 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)

  → 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

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

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

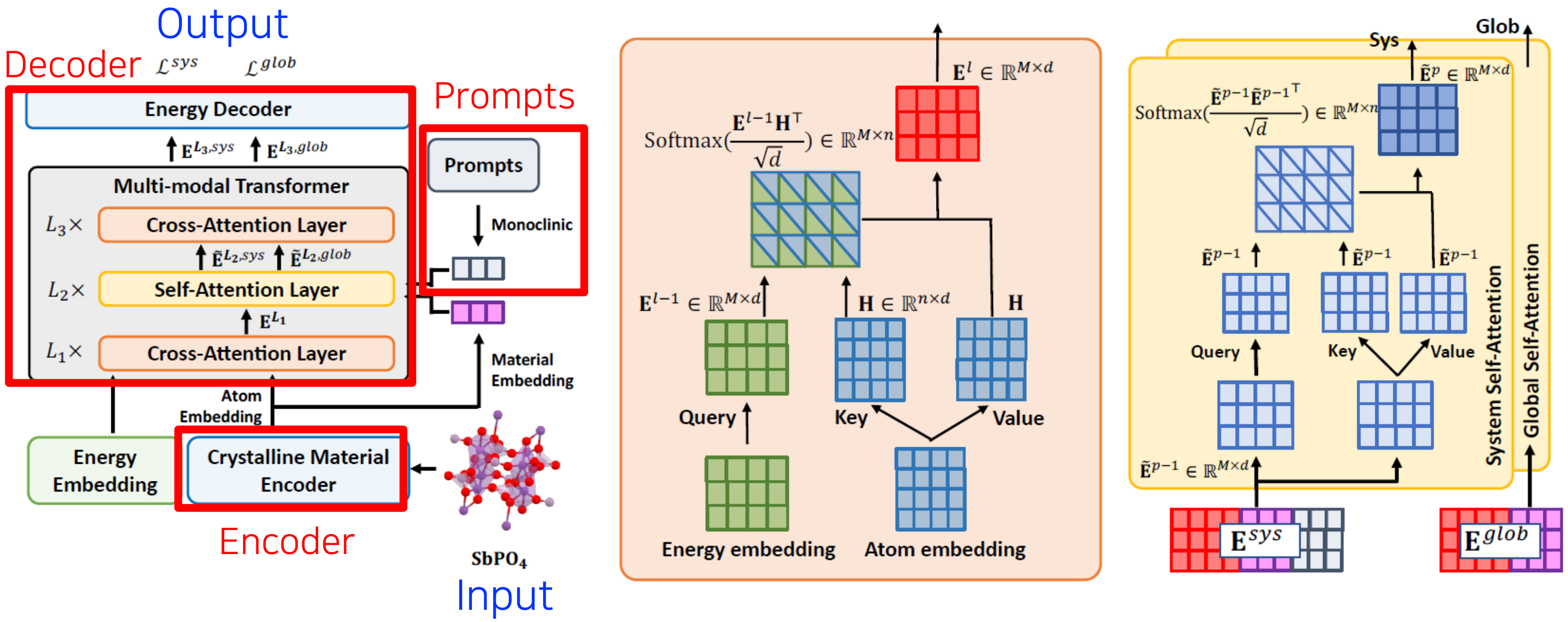
Our idea: There are 7 Widely known Crystal Systems
- i.e., Cubic, Hexagonal, ⋯, Triclinic
- Introduce 7 learnable prompts $P \in \mathbb{R}^{7 \times d_p}$

- Incorporating structural information to the model by injecting prompts, not naively concatenating

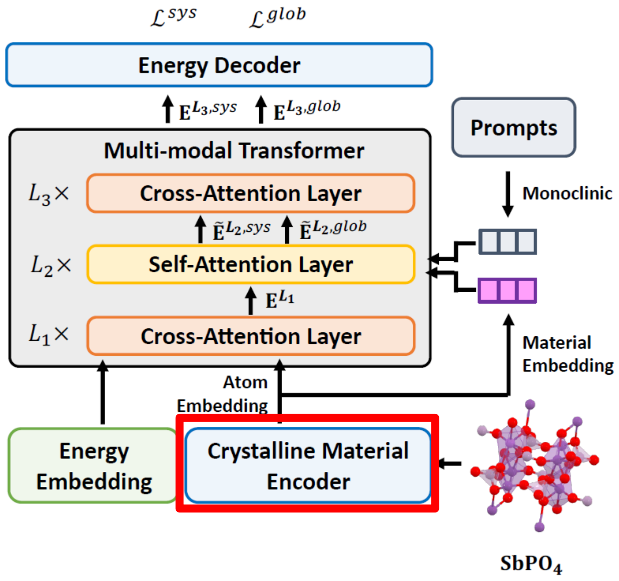
https://ai.googleblog.com/2022/02/guiding-frozen-language-models-with.html

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**
- We utilize learnable **prompts to guide the model to learn the crystal structural system-specific interaction** between materials and energies

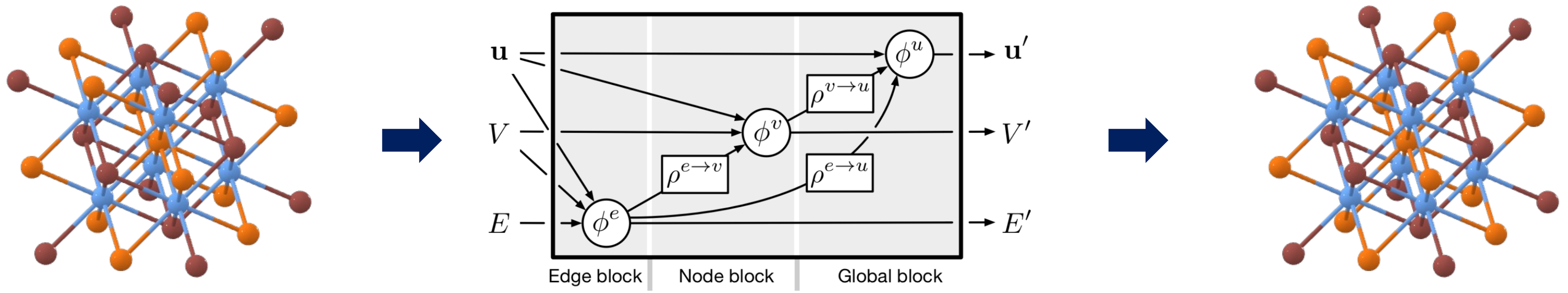

Output
Decoder
$\mathcal{L}^{sys}$
$\mathcal{L}^{glob}$
Energy Decoder
$\mathbf{E}^{L_3,sys}$
$\mathbf{E}^{L_3,glob}$
Multi-modal Transformer
$L_3\times$
Cross-Attention Layer
$\tilde{\mathbf{E}}^{L_2,sys}$
$\tilde{\mathbf{E}}^{L_2,glob}$
$L_2\times$
Self-Attention Layer
$\mathbf{E}^{L_1}$
$L_1\times$
Cross-Attention Layer
Prompts
Prompts
Monoclinic
Material Embedding
Atom Embedding
Energy Embedding
Crystalline Material Encoder
Encoder
$SbPO_4$
Input
Softmax($\frac{\mathbf{E}^{l-1}\mathbf{H}^\top}{\sqrt{d}}$) ∈ $\mathbb{R}^{M\times n}$
$\mathbf{E}^l \in \mathbb{R}^{M\times d}$
$\mathbf{E}^{l-1} \in \mathbb{R}^{M\times d}$
$\mathbf{H} \in \mathbb{R}^{n\times d}$
$\mathbf{H}$
Query
Key
Value
Energy embedding
Atom embedding
Sys
Glob
Softmax($\frac{\tilde{\mathbf{E}}^{p-1}\tilde{\mathbf{E}}^{p-1\top}}{\sqrt{d}}$) ∈ $\mathbb{R}^{M\times n}$
$\tilde{\mathbf{E}}^p \in \mathbb{R}^{M\times d}$
$\tilde{\mathbf{E}}^{p-1}$
$\tilde{\mathbf{E}}^{p-1}$
$\tilde{\mathbf{E}}^{p-1}$
Query
Key
Value
$\tilde{\mathbf{E}}^{p-1} \in \mathbb{R}^{M\times d}$
System Self-Attention
Global Self-Attention
$\mathbf{E}^{sys}$
$\mathbf{E}^{glob}$

Proposed Model
(Prompt-guided DOSTransformer)



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

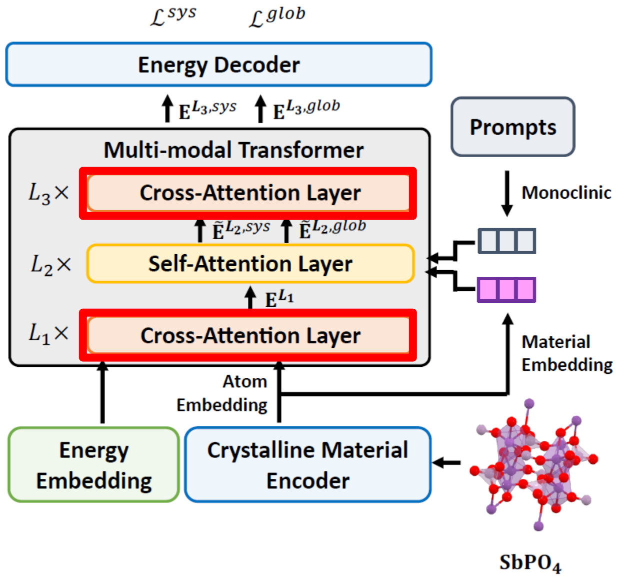


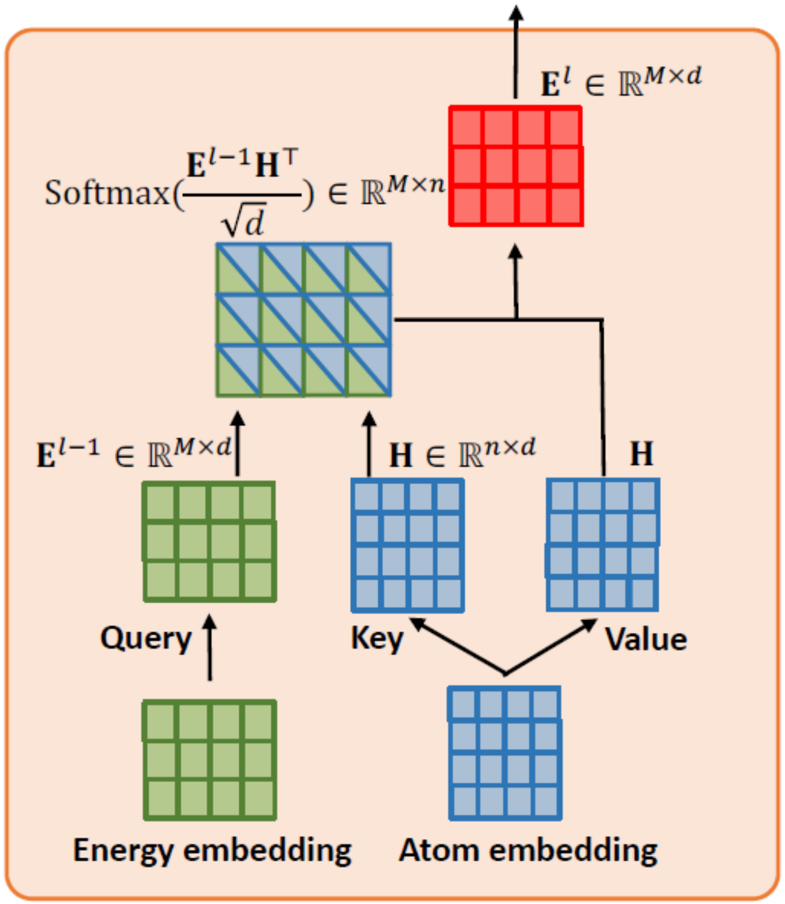
Architecture of Graph Network block

Battaglia, Peter W., et al. "Relational inductive biases, deep learning, and graph networks." *arXiv preprint arXiv:1806.01261* (2018).

PROMPT-GUIDED MULTI-MODAL TRANSFORMER

- Cross-Attention
- Obtain crystal-specific energy embedding $\boldsymbol{E}^l$

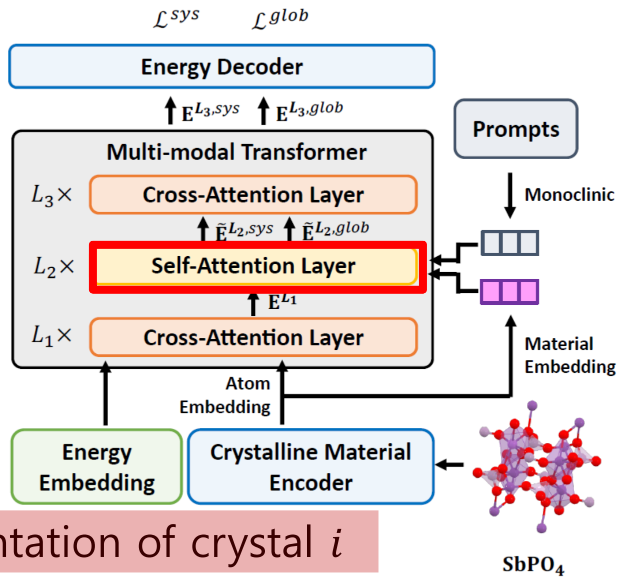




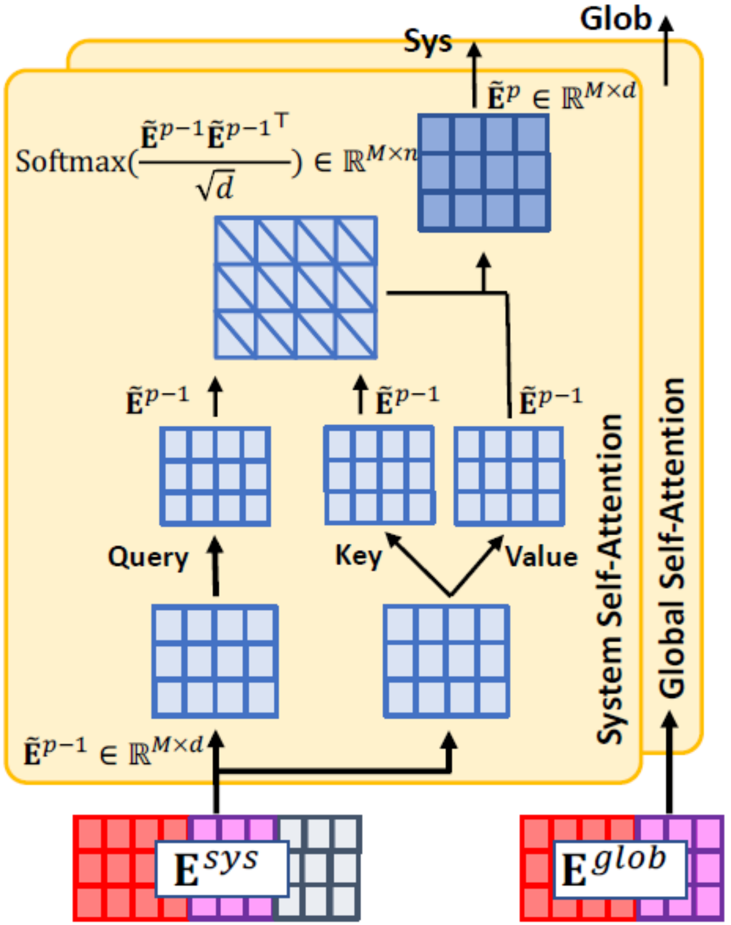
$\mathbf{E}^l \in \mathbb{R}^{M\times d}$
$\mathrm{Softmax}(\frac{\mathbf{E}^{l-1}\mathbf{H}^\top}{\sqrt{d}}) \in \mathbb{R}^{M\times n}$
$\mathbf{E}^{l-1} \in \mathbb{R}^{M\times d}$
$\mathbf{H} \in \mathbb{R}^{n\times d}$
$\mathbf{H}$
Query
Key
Value
Energy embedding
Atom embedding

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

PROMPT-GUIDED MULTI-MODAL TRANSFORMER



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



Glob
Sys
$\tilde{\mathbf{E}}^p \in \mathbb{R}^{M\times d}$
$\mathrm{Softmax}(\frac{\tilde{\mathbf{E}}^{p-1}\tilde{\mathbf{E}}^{p-1\top}}{\sqrt{d}}) \in \mathbb{R}^{M\times n}$
$\tilde{\mathbf{E}}^{p-1}$
$\tilde{\mathbf{E}}^{p-1}$
$\tilde{\mathbf{E}}^{p-1}$
Query
Key
Value
System Self-Attention
Global Self-Attention
$\tilde{\mathbf{E}}^{p-1} \in \mathbb{R}^{M\times d}$
$\mathbf{E}^{sys}$
$\mathbf{E}^{glob}$

Energy embedding

Sum-pooled representation of crystal i

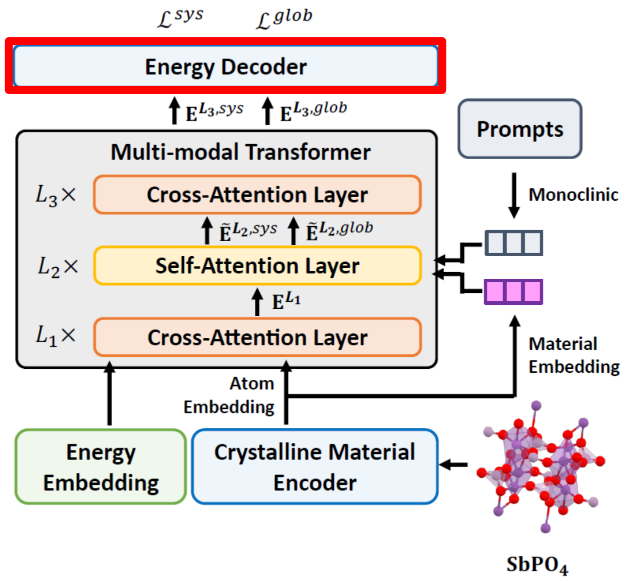
$$\mathbf{E}_j^{glob} = (\mathbf{E}_j^{L_1} || \mathbf{g}_i) \qquad \tilde{\mathbf{E}}_j^0 = \phi_1(\mathbf{E}_j^{glob})$$

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

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

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

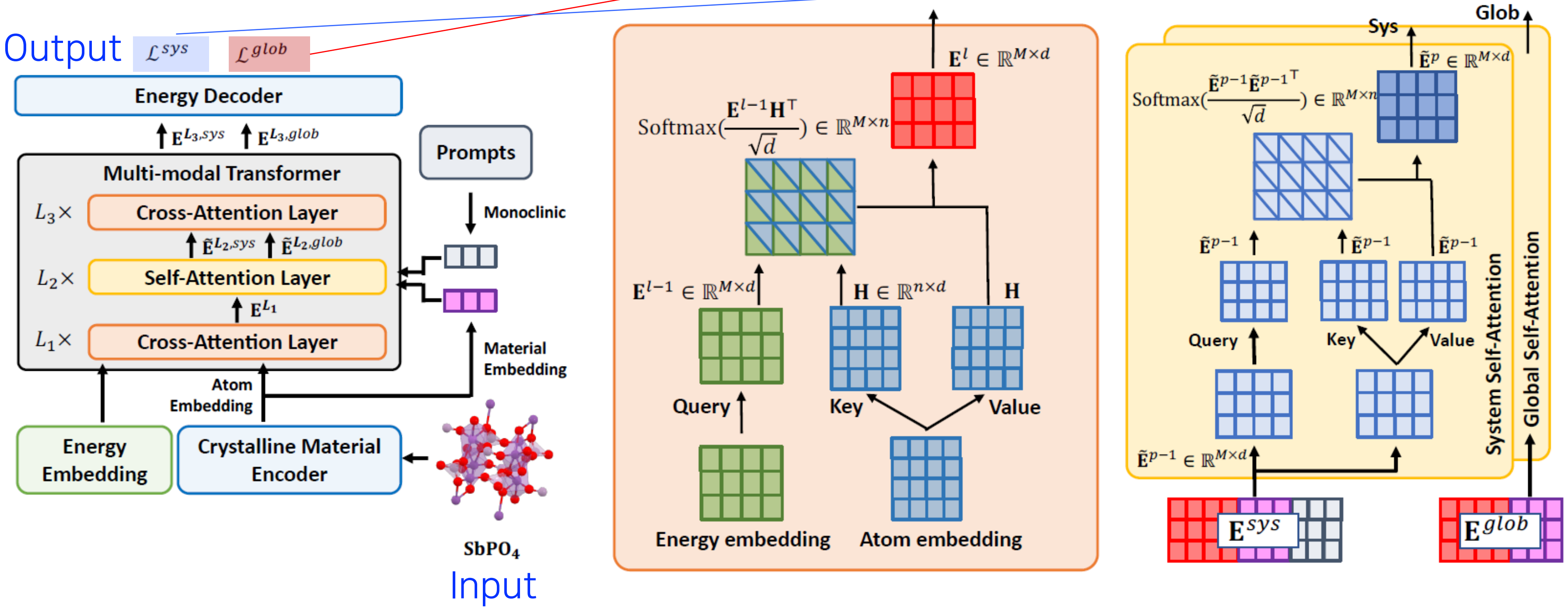
Our proposed method: Prompt-guided DOSTransformer

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

Total training loss Global loss Balancing term

$$\mathcal{L}^{total} = \mathcal{L}^{glob} + \beta \cdot \mathcal{L}^{sys}$$

Crystal System loss

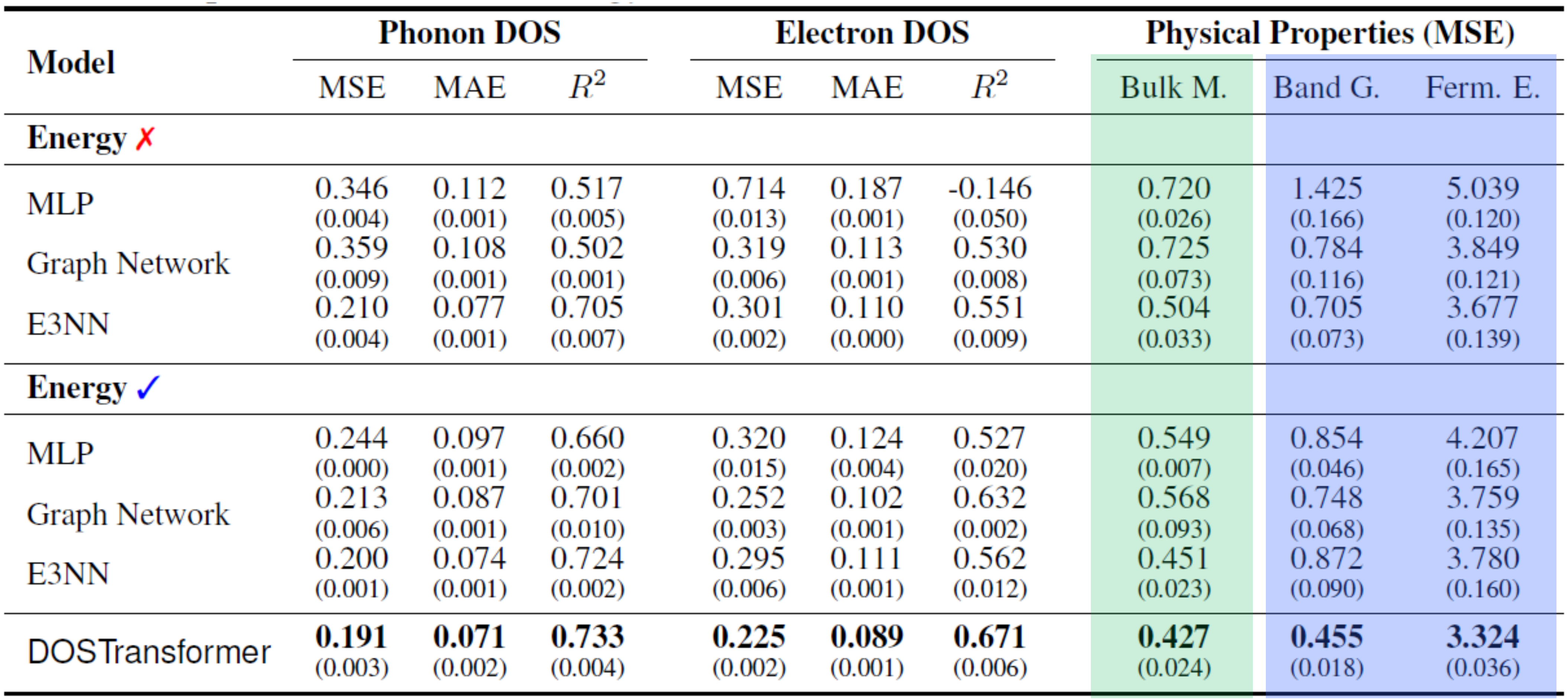


Output
$\mathcal{L}^{sys}$
$\mathcal{L}^{glob}$
Energy Decoder
$\mathbf{E}^{L_3,sys}$
$\mathbf{E}^{L_3,glob}$
Multi-modal Transformer
$L_3\times$
Cross-Attention Layer
$\tilde{\mathbf{E}}^{L_2,sys}$
$\tilde{\mathbf{E}}^{L_2,glob}$
$L_2\times$
Self-Attention Layer
$\mathbf{E}^{L_1}$
$L_1\times$
Cross-Attention Layer
Prompts
Monoclinic
Material Embedding
Atom Embedding
Energy Embedding
Crystalline Material Encoder
$SbPO_4$
Input
$\mathbf{E}^{l} \in \mathbb{R}^{M\times d}$
$\text{Softmax}(\frac{\mathbf{E}^{l-1}\mathbf{H}^\top}{\sqrt{d}}) \in \mathbb{R}^{M\times n}$
$\mathbf{E}^{l-1} \in \mathbb{R}^{M\times d}$
$\mathbf{H} \in \mathbb{R}^{n\times d}$
$\mathbf{H}$
Query
Key
Value
Energy embedding
Atom embedding
Sys
Glob
$\tilde{\mathbf{E}}^{p} \in \mathbb{R}^{M\times d}$
$\text{Softmax}(\frac{\tilde{\mathbf{E}}^{p-1}\tilde{\mathbf{E}}^{p-1\top}}{\sqrt{d}}) \in \mathbb{R}^{M\times n}$
$\tilde{\mathbf{E}}^{p-1}$
$\tilde{\mathbf{E}}^{p-1}$
$\tilde{\mathbf{E}}^{p-1}$
Query
Key
Value
$\tilde{\mathbf{E}}^{p-1} \in \mathbb{R}^{M\times d}$
System Self-Attention
Global Self-Attention
$\mathbf{E}^{sys}$
$\mathbf{E}^{glob}$

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.
Energy ✗									
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)
Energy ✓									
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	**0.191** (0.003)	**0.071** (0.002)	**0.733** (0.004)	**0.225** (0.002)	**0.089** (0.001)	**0.671** (0.006)	**0.427** (0.024)	**0.455** (0.018)	**3.324** (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

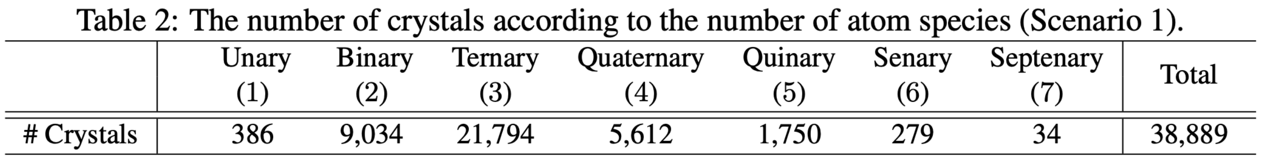


Table 2: The number of crystals according to the number of atom species (Scenario 1).

	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

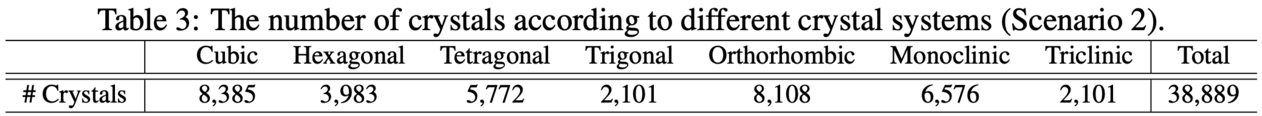
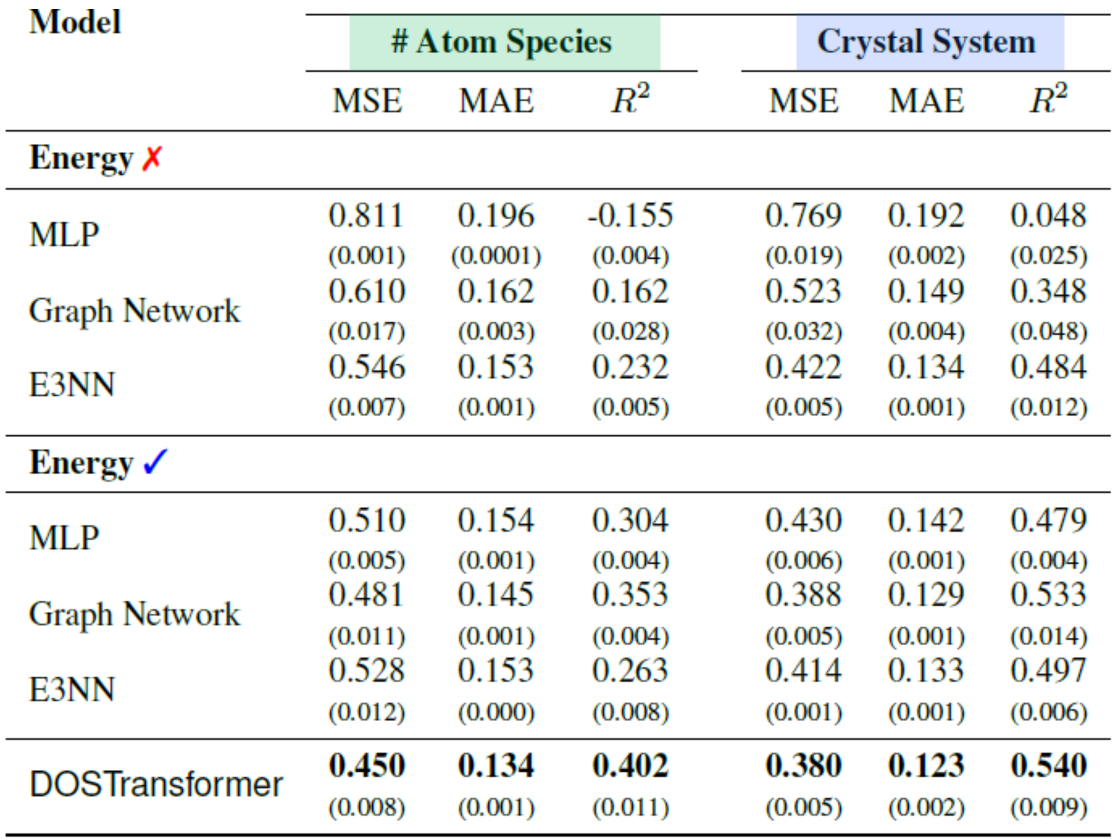


Table 3: The number of crystals according to different crystal systems (Scenario 2).

	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

- 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

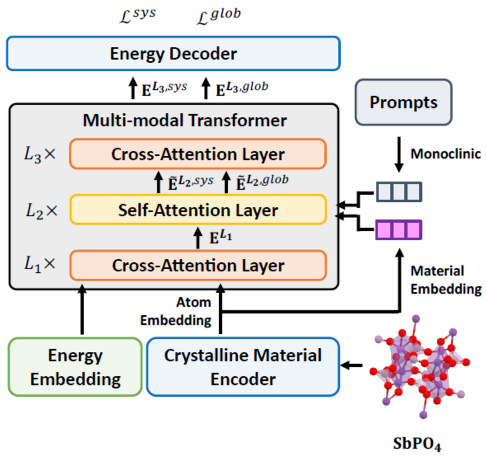


Model	# Atom Species			Crystal System		
	MSE	MAE	R^2	MSE	MAE	R^2
Energy ✗						
MLP	0.811 (0.001)	0.196 (0.0001)	-0.155 (0.004)	0.769 (0.019)	0.192 (0.002)	0.048 (0.025)
Graph Network	0.610 (0.017)	0.162 (0.003)	0.162 (0.028)	0.523 (0.032)	0.149 (0.004)	0.348 (0.048)
E3NN	0.546 (0.007)	0.153 (0.001)	0.232 (0.005)	0.422 (0.005)	0.134 (0.001)	0.484 (0.012)
Energy ✓						
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	**0.450** (0.008)	**0.134** (0.001)	**0.402** (0.011)	**0.380** (0.005)	**0.123** (0.002)	**0.540** (0.009)

- DOSTransformer performs well in OOD

Result: Fine-tuning in OOD scenario 2

Fine-tuning on 10 % training data

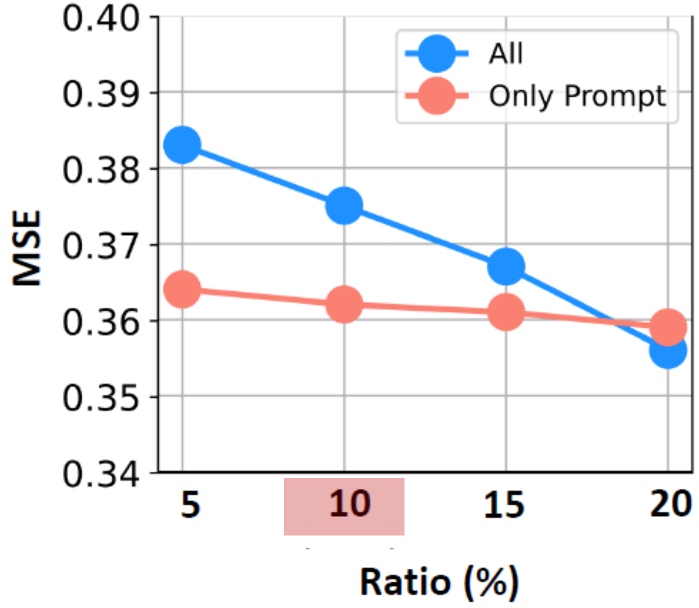


Fine-tuning all model parameters of DOSTransformer

Fine-tuning only the prompt and decoder of DOSTransformer

Model	MSE	MAE	R^2
Energy ✗			
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)
Energy ✓			
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)
DOSTransformer			
All	0.375 (0.009)	**0.123** (0.001)	0.543 (0.013)
Only Prompt	**0.362** (0.002)	**0.123** (0.001)	**0.559** (0.003)

Various training data ratio for Fine-tuning



- Additional fine-tuning achieves performance gain for all models
  - However, it was marginal due to a limited number of materials used for fine-tuning
- Only fine-tuning (prompts & decoder) model 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

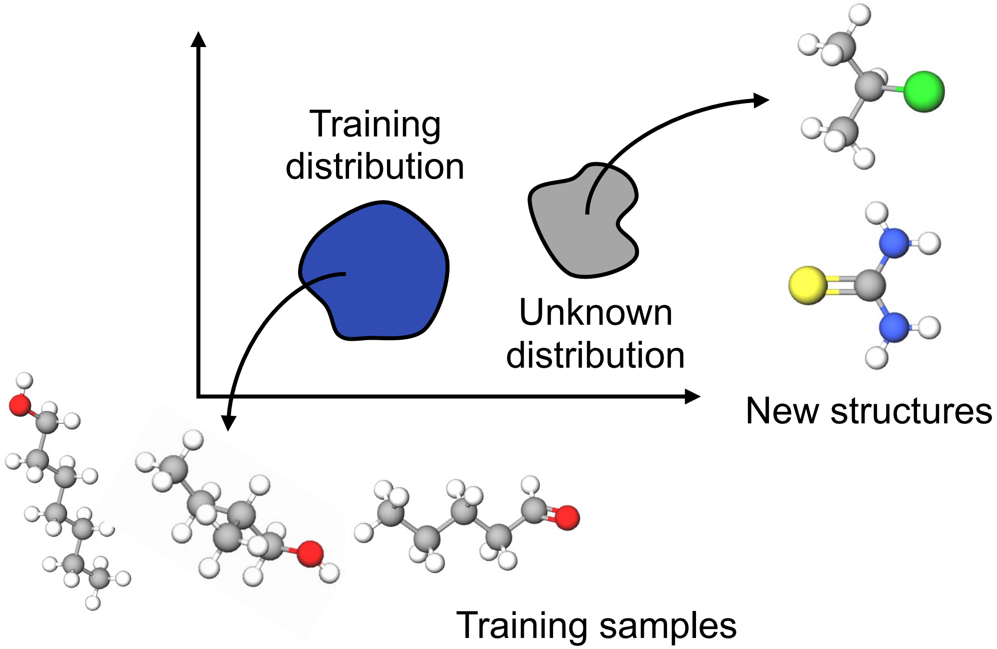
Table of Contents

시간	주제	주요 내용
1	그래프 신경망 개요	- 그래프 신경망 전반적인 소개 - 그래프 종류에 따른 다양한 그래프 신경망 소개
2	소재 물성 예측 연구	- 소재 물성 예측 연구 최신 동향 소개 - Transformer 기반 모델 소개 및 extrapolation을 위한 모델 소개
3	물질 간 화학 반응 예측 연구	- 물질 간 화학 반응 예측 연구 동향 소개 - 정보 이론(Information bottleneck) 및 인과추론(Causal inference) 기반 모델 소개

Introduction: Extrapolation

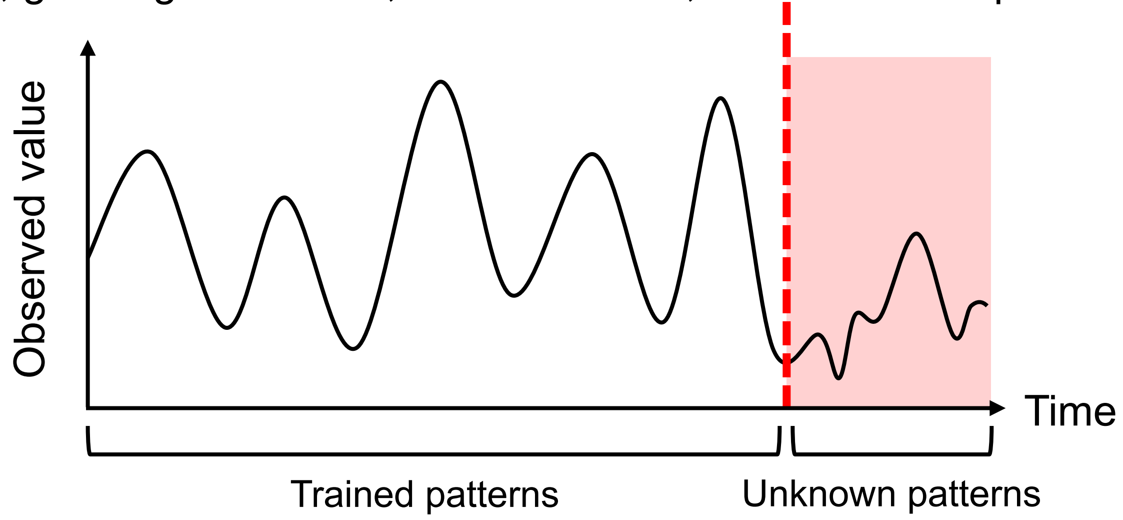
- **Goal:** Predict unseen data **outside the training distribution**
- Extrapolation is challenging because the input data usually follows an unknown distribution
- However, **extrapolation is common in scientific applications** in which discovering unobserved scientific knowledge is crucial

Material discovery



Time-series forecasting

(e.g., geomagnetic storm, network attack, and chemical spectrum)



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

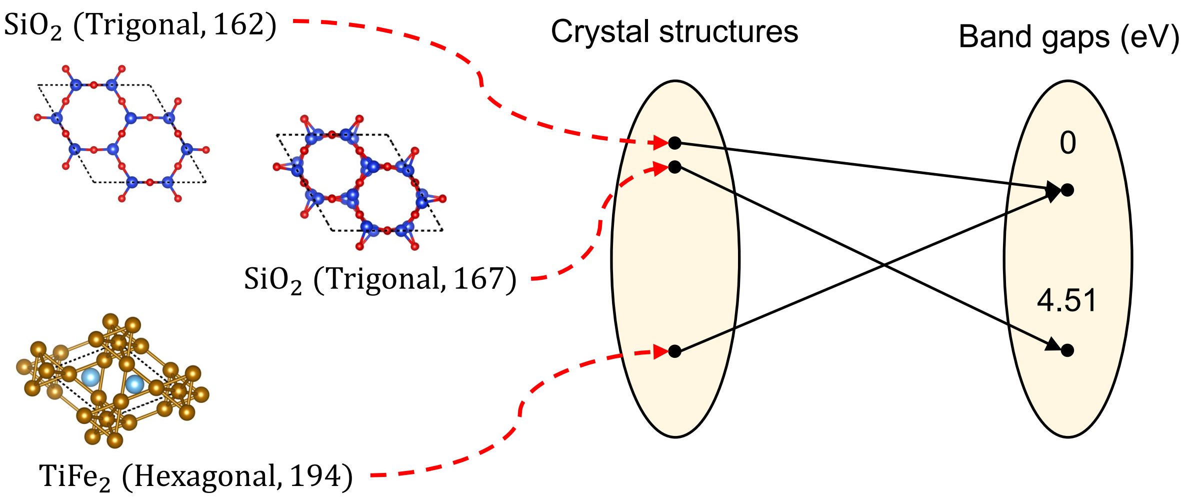
$$\underset{\text{Extrapolation Error}}{L_e} = \mathbb{E}_{(x,y)\sim \underbrace{\mathcal{X}}_{\text{Data distribution}}\setminus \overbrace{\mathcal{D}}^{\text{Training distribution}}}\Big[\underbrace{L_s}_{\text{Loss function (Cross entropy, MSE)}}(y, \overbrace{f}^{\text{Prediction model}}(\boldsymbol{x}))\Big]$$

- $(\underbrace{x}_{\text{Input data}}, \underbrace{y}_{\text{Target response}})$: 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**.

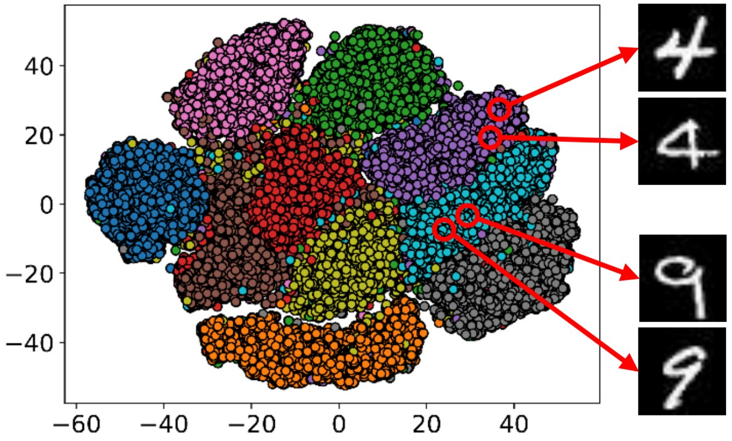


Two **similar structures** may have completely **different physical properties**, **whereas** two completely **different structures** may have **the same physical property**

Image Dataset vs. Scientific Dataset

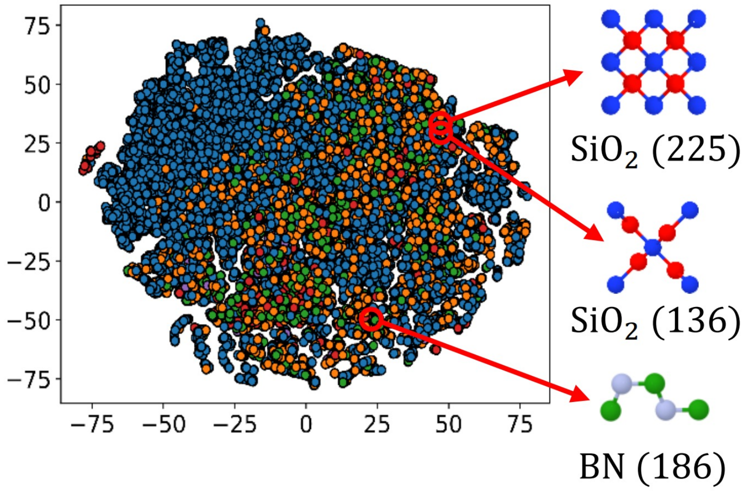
- T-SNE plots of MNIST and Material Project (MP) datasets
- Each point indicates an **image** or a **material** with **target response (label) denoted by colors**.
    - MNIST: class label
    - MP dataset: band gap

(a) MNIST dataset



Similar images share similar labels

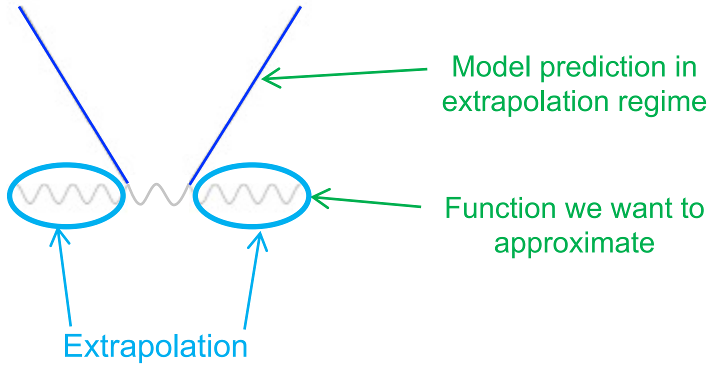
(b) MP dataset

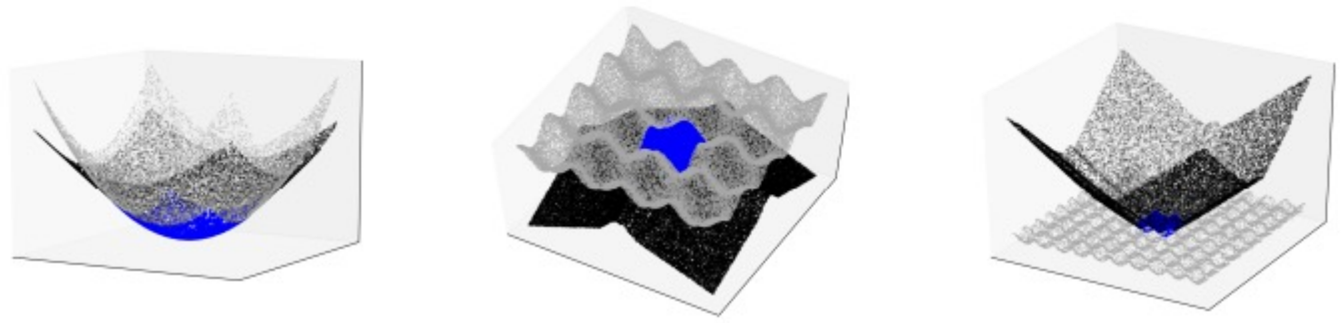


Similar materials do not necessarily share similar labels

How Neural Networks Extrapolate (Xu et al, ICLR21)

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


Model prediction in extrapolation regime
Function we want to approximate
Extrapolation



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
  - Graph convolutional neural networks with global attention for improved materials property prediction. Physical Chemistry Chemical Physics 2020
  - Direct prediction of phonon density of states with Euclidean neural networks. Advanced Science 2021
  - **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[1] and Chanyoung Park[2]

[1]Korea Research Institute of Chemical Technology (KRICT), Republic of Korea

[2]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

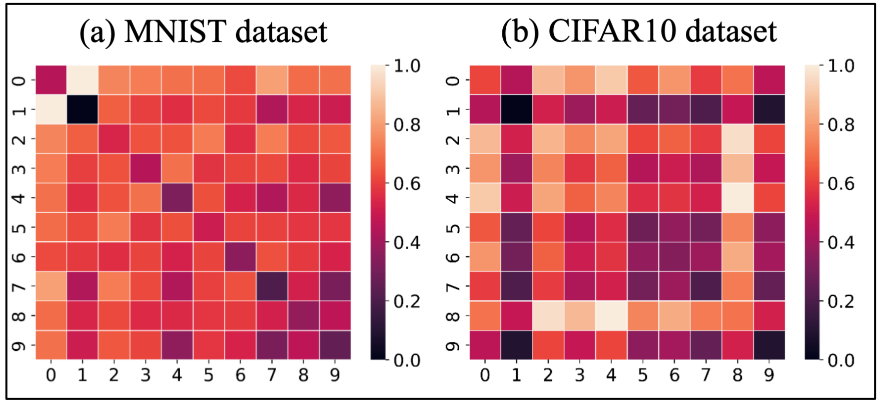
Can we learn extrapolation models?

Can we learn extrapolation models?

: Image Dataset vs. Scientific Dataset

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

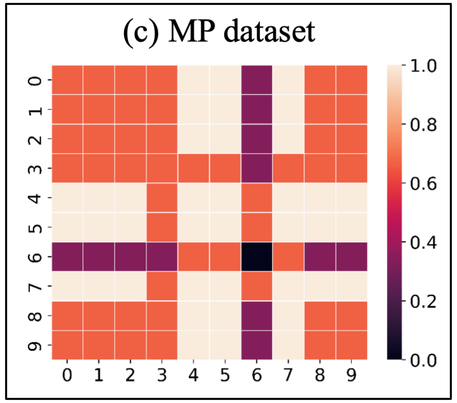
Image



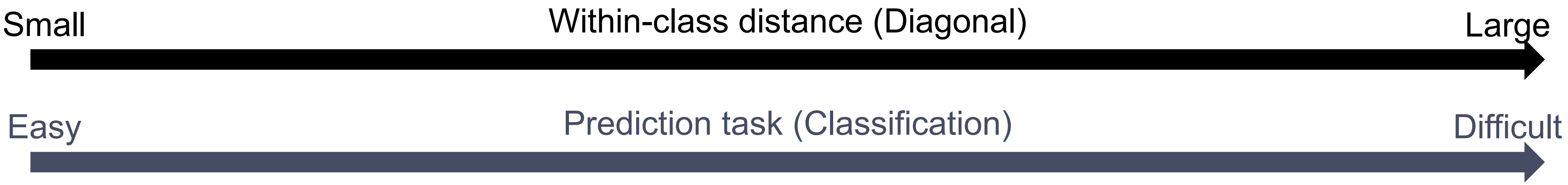
(a) MNIST dataset

(b) CIFAR10 dataset

Scientific dataset



(c) MP dataset



Small Within-class distance (Diagonal) Large

Easy Prediction task (Classification) Difficult

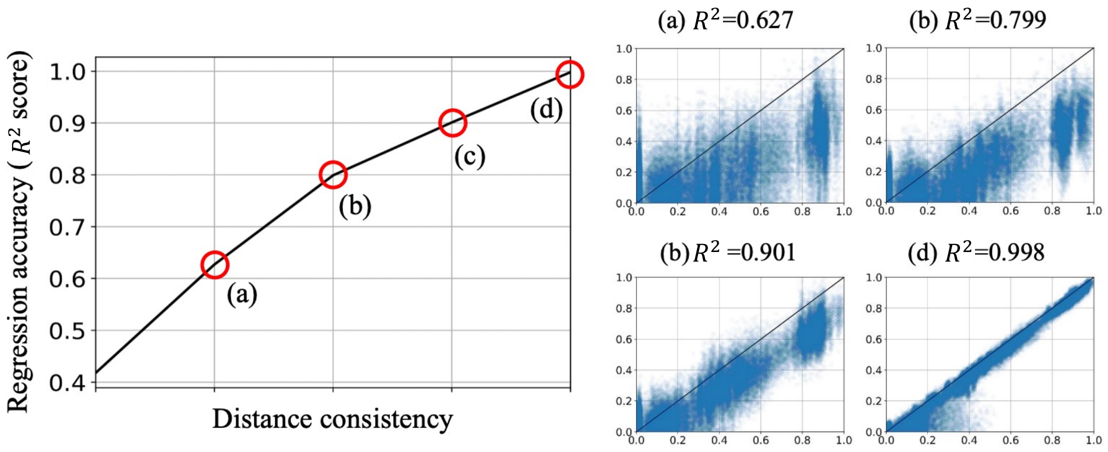
Prediction tasks can be made easier when,
Two inputs with same label → Small input distance

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 ≈ regression

Linear regression on synthetic datasets



High distance consistency → High accuracy (R^2 score) → **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 $(\boldsymbol{x}_i, y_i), (\boldsymbol{x}_j, y_j)$
  - **Define**: The distance between them

$$d(\underbrace{d(\boldsymbol{x}_i, \boldsymbol{x}_j)}_{\text{Dist. btw. inputs}} - \overbrace{d(y_i, y_j)}^{\text{Dist. btw. targets}}) \xrightarrow[\text{Consider all } N^2 \text{ pairs}]{} \sum_{i=1}^{N} \sum_{j=1}^{N} d(d(\boldsymbol{x}_i, \boldsymbol{x}_j) - d(y_i, y_j))$$

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

Nonlinearity Encoding based on Wasserstein Distance

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

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

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 $\boldsymbol{d}(\boldsymbol{x}, \boldsymbol{x}')$** for all $(\boldsymbol{x}, \boldsymbol{x}') \in \mathcal{X}\times\mathcal{X}$, where $d: \mathcal{X}\times\mathcal{X} \to [0, \infty)$ is a distance metric.*

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

$$(p = 1)$$

➡

Distance consistency btw input and target!

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

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

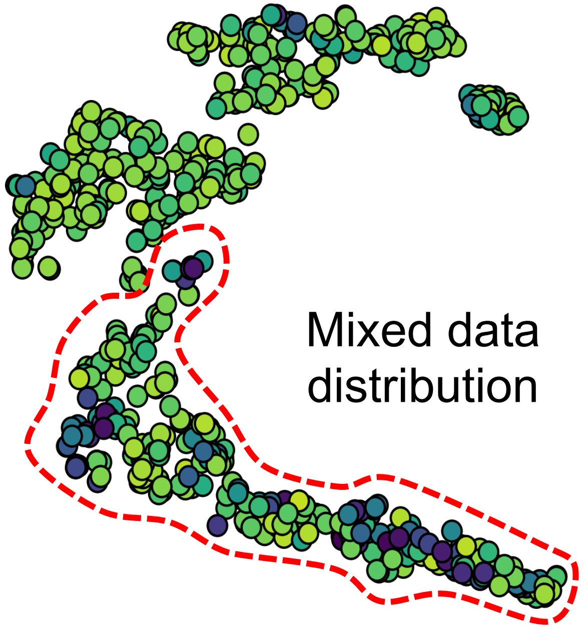
Our goal: **Maximize the distance consistency** between input and target

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

Problem Definition of Nonlinearity Encoding

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

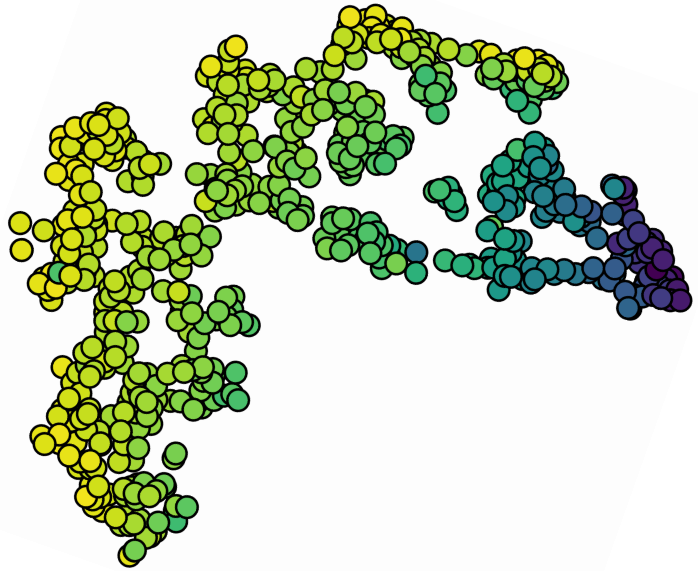
Data distribution in the **original feature space**



Hard

Nonlinearity Encoding →

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



Easy

Optimization: Decomposition of Lagrangian

- Our problem can be defined as follows:

Joint optimization w.r.t. θ and π

# training data

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \sum_{i=1}^{N} \sum_{j=1}^{N} \inf_{\pi\in\Pi} \int_{\mathcal{M}\times\mathcal{M}} \left\| r_{ij} - u_{ij} \right\|_p \pi\left(r_{ij}, u_{ij}\right) \mathrm{d}r\mathrm{d}u$$

- $r_{ij} = d\left(\phi(\mathbf{x}_i;\theta), \phi\left(\mathbf{x}_j;\theta\right)\right)$: **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 **Kantorovich-Rubinstein duality** [6]):

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

$$+ \sum_{(i,j)\in\mathcal{N}} \left(p\left(r_{ij}\right) - \sum_{(k,q)\in I_{ij}} \pi\left(r_{ij}, u_{kq}\right) \right) f\left(r_{ij}\right) + \sum_{(i,j)\in\mathcal{N}} \left(p\left(u_{ij}\right) - \sum_{(k,q)\in\mathcal{N}} \pi\left(r_{kq}, u_{ij}\right)\right) g\left(u_{ij}\right) + \sum_{(i,j)\in\mathcal{N}} \sum_{(k,q)\in\mathcal{N}\setminus I_{ij}} \pi\left(r_{kq}, u_{ij}\right) g\left(u_{ij}\right),$$

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:

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \sum_{i=1}^{N} \sum_{j=1}^{N} \inf_{\pi \in \Pi} \int_{\mathcal{M} \times \mathcal{M}} \left\| r_{ij} - u_{ij} \right\|_p \pi\left(r_{ij}, u_{ij}\right) \mathrm{d}r\mathrm{d}u$$

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

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

Enforce distance consistency between data pairs!

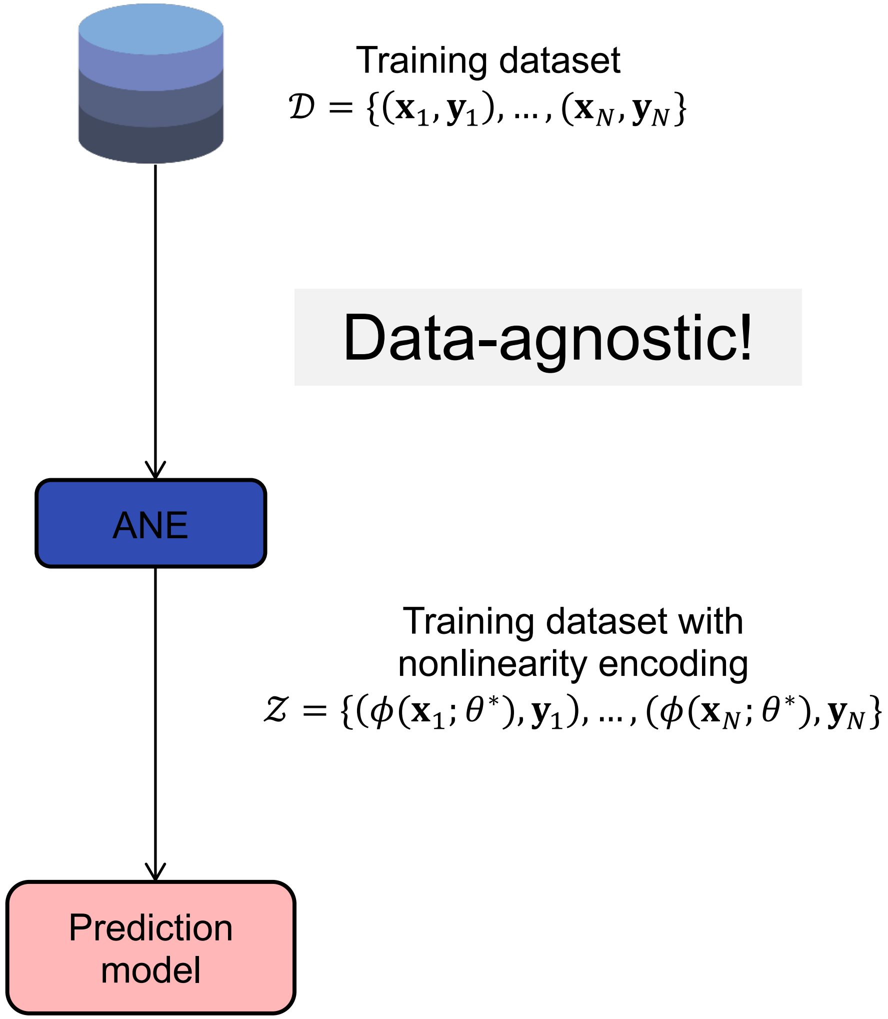
Optimization: Model Parameter Optimization

Training of ANE-based prediction model

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

ANE

Prediction model


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

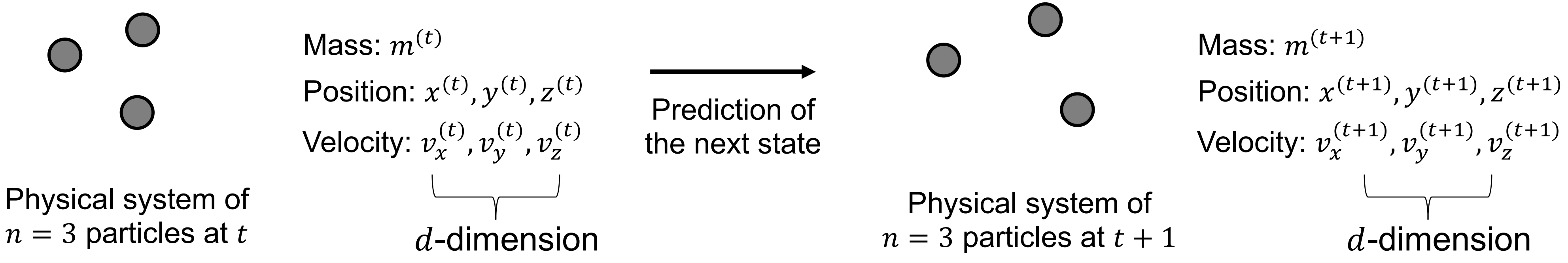
Data-agnostic!

Experiments

- Matrix-shaped data
- Graph-structured data
- Time-series data

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



Physical system of $n = 3$ particles at t

Mass: $m^{(t)}$

Position: $x^{(t)}, y^{(t)}, z^{(t)}$

Velocity: $\underbrace{v_x^{(t)}, v_y^{(t)}, v_z^{(t)}}_{d\text{-dimension}}$

Prediction of the next state

Physical system of $n = 3$ particles at $t + 1$

Mass: $m^{(t+1)}$

Position: $x^{(t+1)}, y^{(t+1)}, z^{(t+1)}$

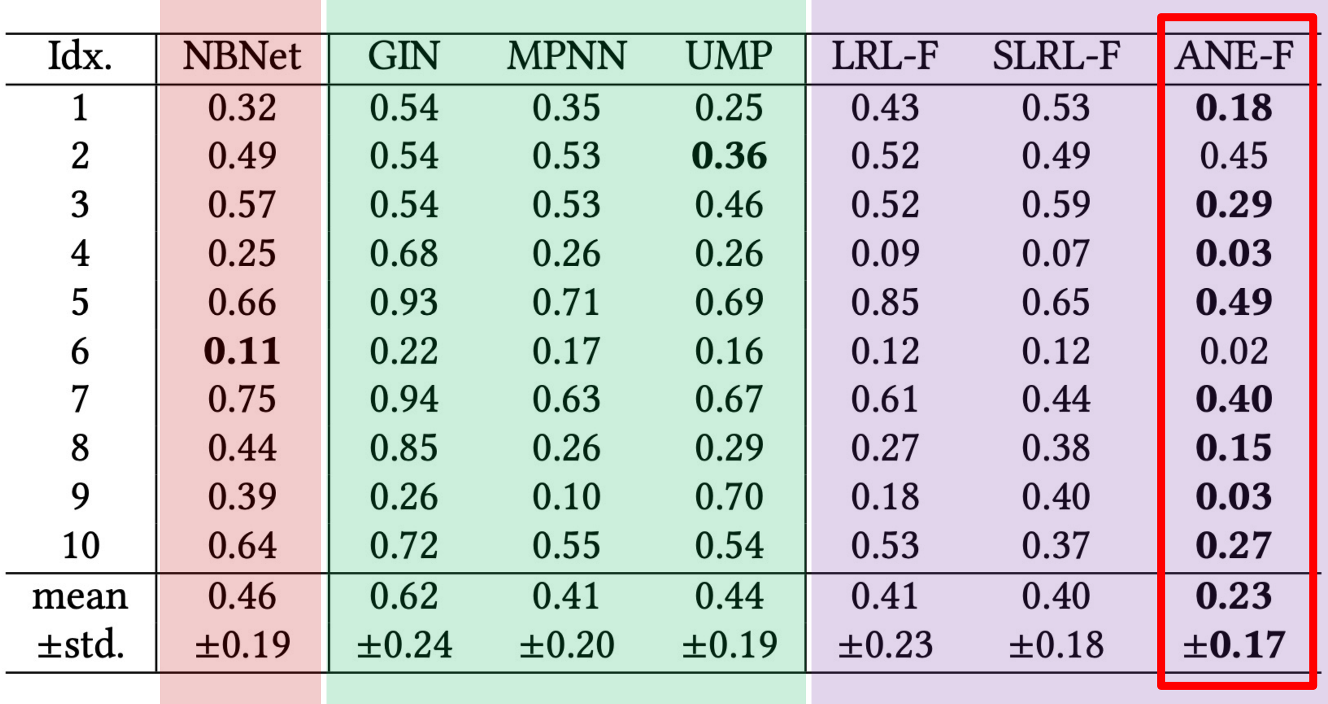
Velocity: $\underbrace{v_x^{(t+1)}, v_y^{(t+1)}, v_z^{(t+1)}}_{d\text{-dimension}}$

- **Data preprocessing:** 3-dimensional 3-body problem. $\boldsymbol{x}_t \in \mathbb{R}^{3\times7}$ and $\boldsymbol{y}_t \in \mathbb{R}^{3\times3}$ ← 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

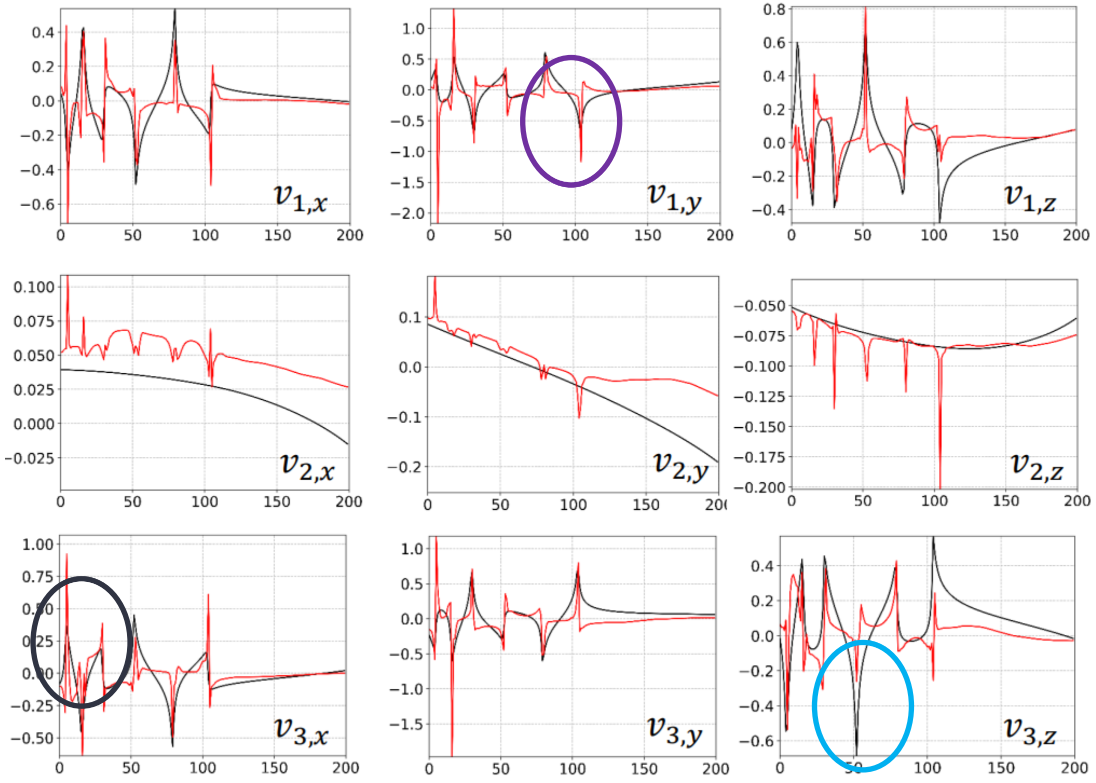


Idx.	NBNet	GIN	MPNN	UMP	LRL-F	SLRL-F	ANE-F
1	0.32	0.54	0.35	0.25	0.43	0.53	**0.18**
2	0.49	0.54	0.53	**0.36**	0.52	0.49	0.45
3	0.57	0.54	0.53	0.46	0.52	0.59	**0.29**
4	0.25	0.68	0.26	0.26	0.09	0.07	**0.03**
5	0.66	0.93	0.71	0.69	0.85	0.65	**0.49**
6	**0.11**	0.22	0.17	0.16	0.12	0.12	0.02
7	0.75	0.94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

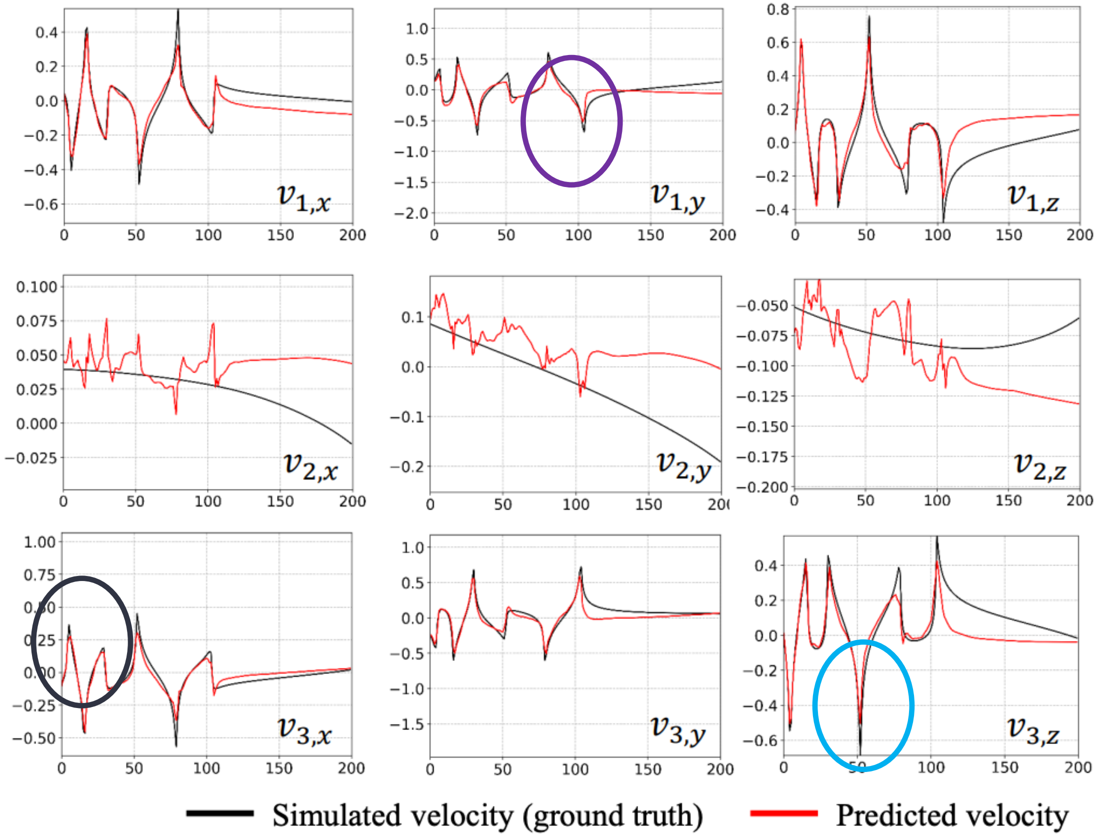
ANE generates input representations that are the most effective to reducing the extrapolation errors

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

State-of-the-art GNN-based method

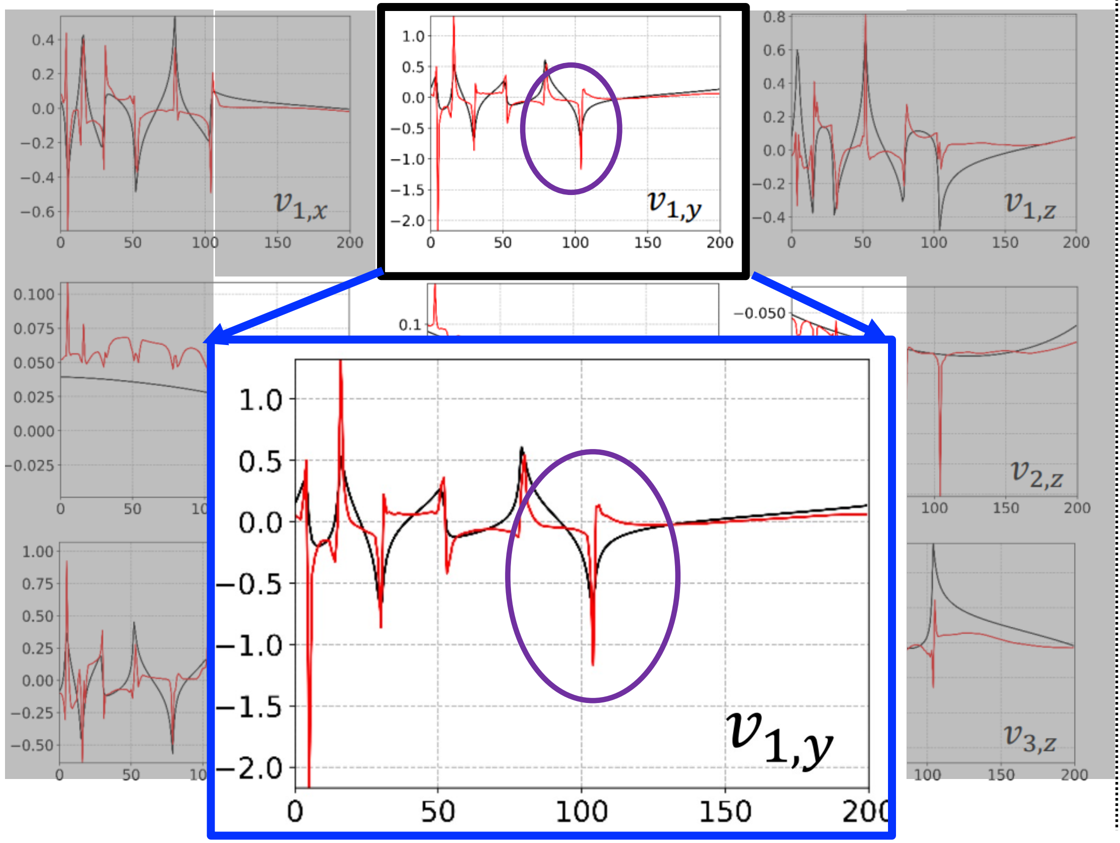


Ours (ANE-F)

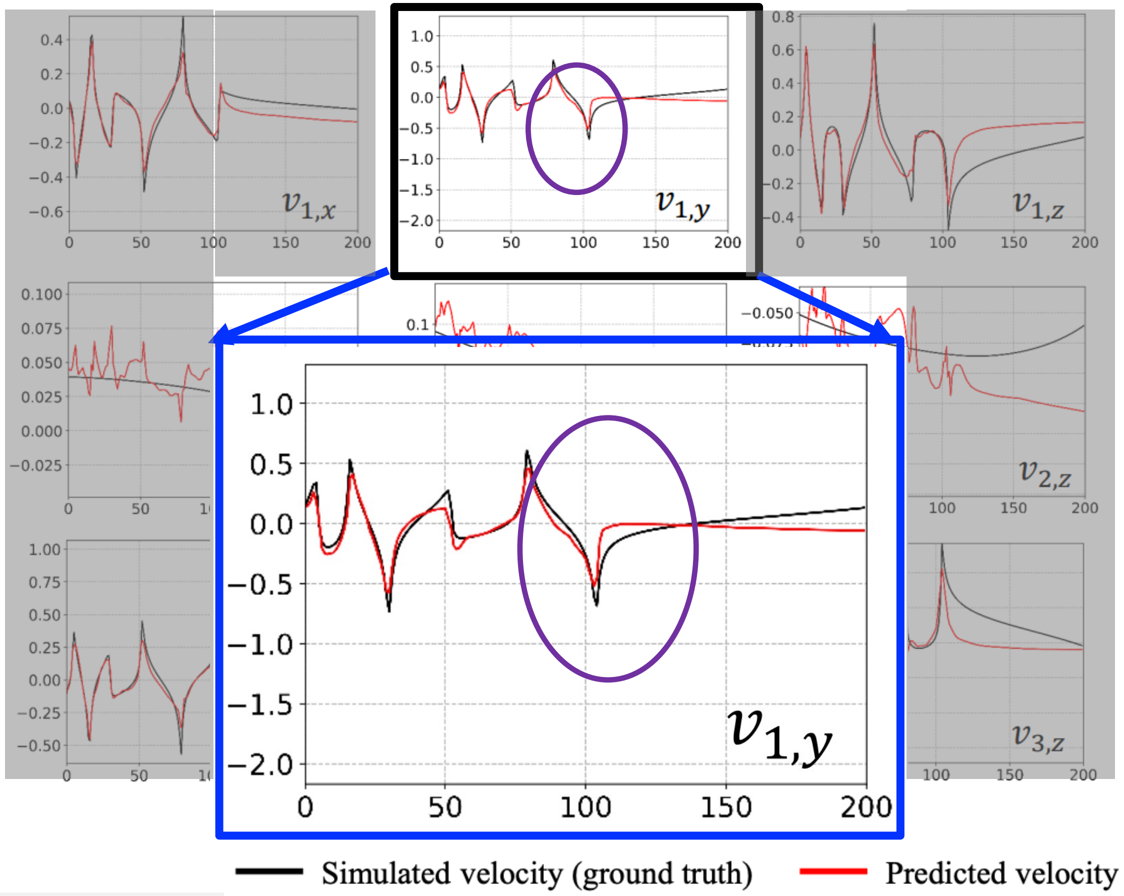


Extrapolation on Matrix-Shaped Data: n-Body Problem (3/3)

State-of-the-art GNN-based method



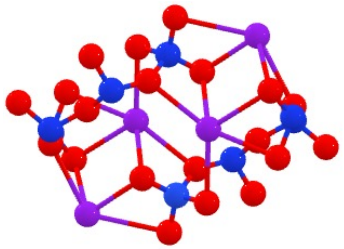
Ours (ANE-F)



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

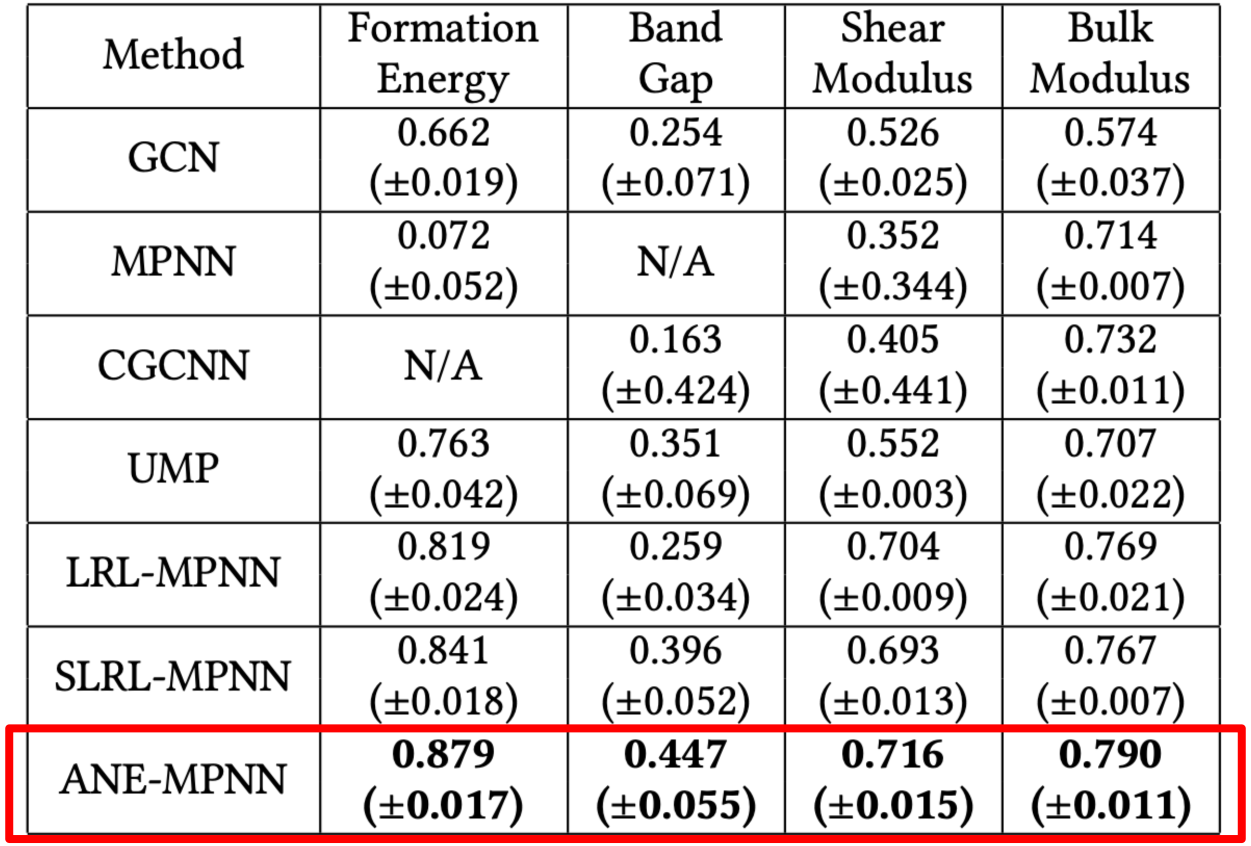
→ Prediction model → Physical and chemical properties of materials

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



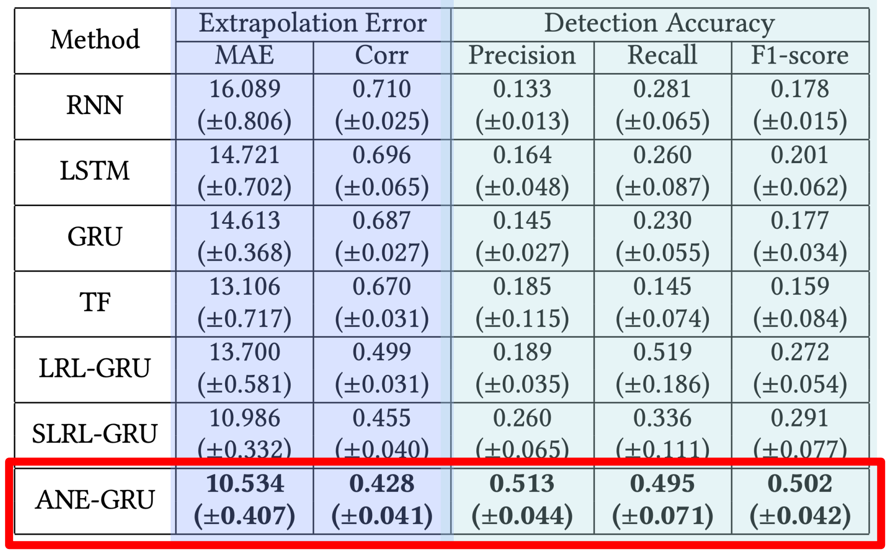
Method	Formation Energy	Band Gap	Shear Modulus	Bulk Modulus
GCN	0.662 (±0.019)	0.254 (±0.071)	0.526 (±0.025)	0.574 (±0.037)
MPNN	0.072 (±0.052)	N/A	0.352 (±0.344)	0.714 (±0.007)
CGCNN	N/A	0.163 (±0.424)	0.405 (±0.441)	0.732 (±0.011)
UMP	0.763 (±0.042)	0.351 (±0.069)	0.552 (±0.003)	0.707 (±0.022)
LRL-MPNN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

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

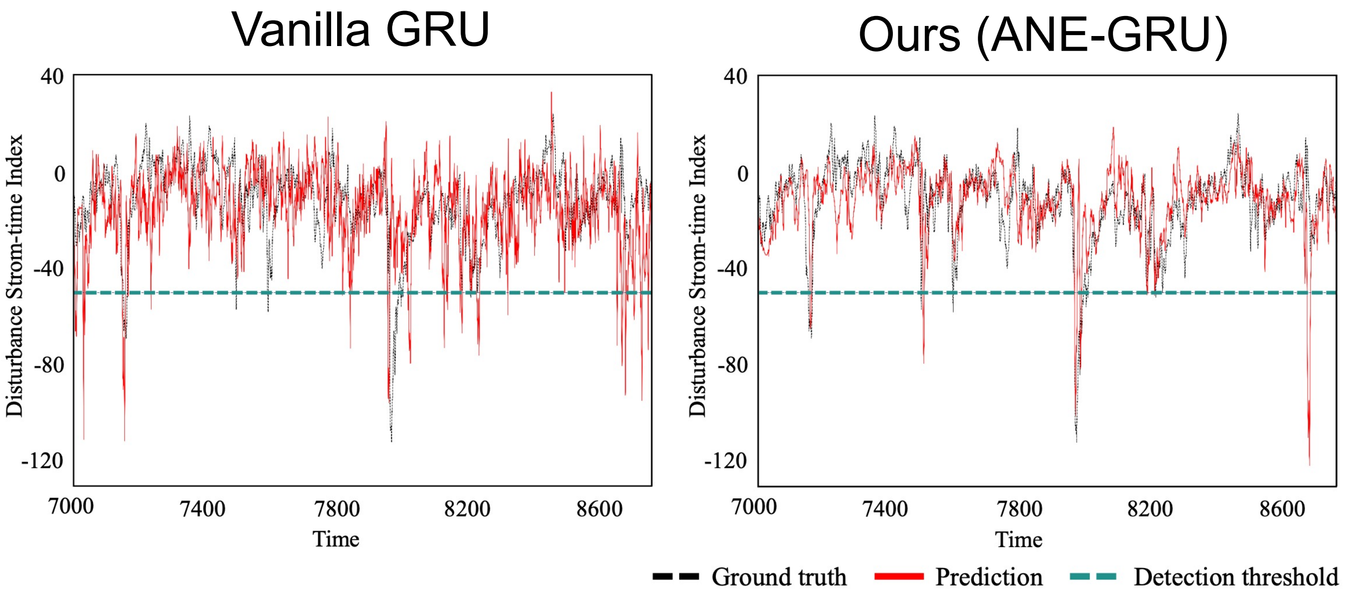
Extrapolation on Time-Series Data: Geomagnetic Storm Forecasting

- **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 (¾ used for training, ¼ used for test)

Task 1 Task 2



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



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

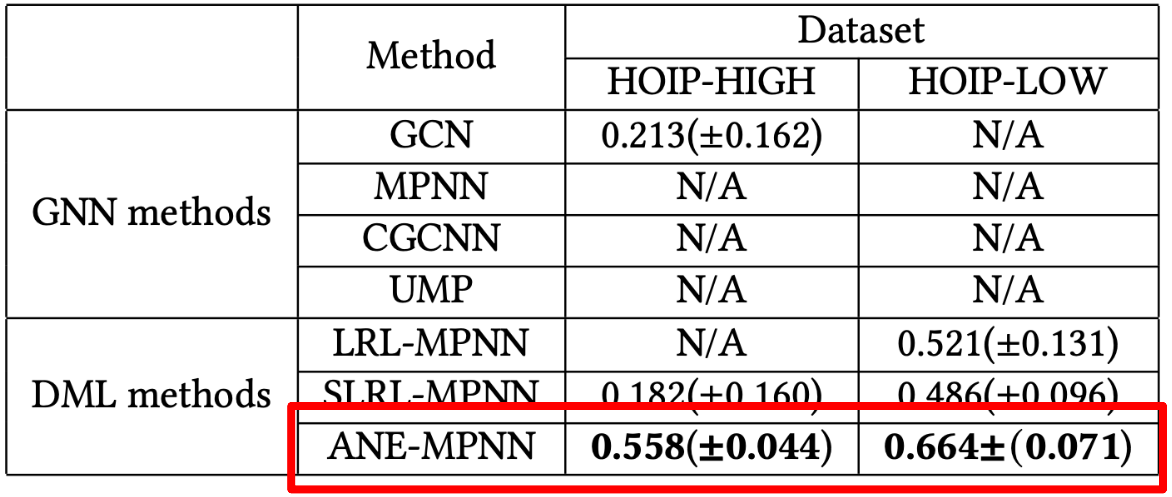
ANE for Discovering Solar Cell Materials

- **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

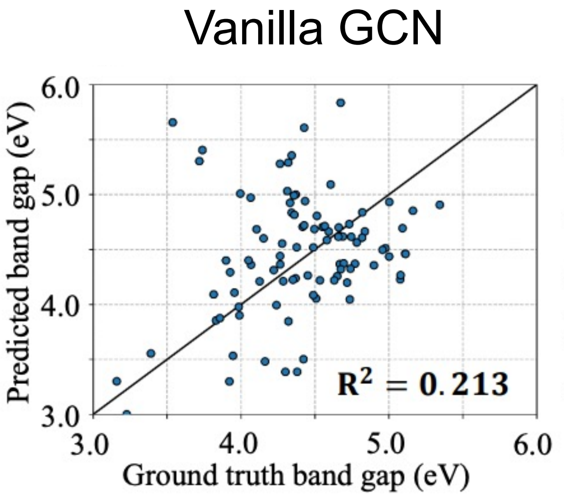
ANE for Discovering Solar Cell Materials

- **Metric:** R^2 score

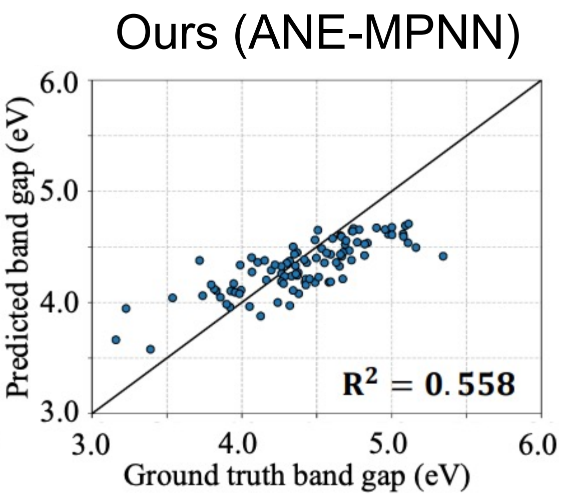
N/A: negative R^2



	Method	Dataset	
		HOIP-HIGH	HOIP-LOW
GNN methods	GCN	0.213(±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(±0.131)
	SLRL-MPNN	0.182(±0.160)	0.486(±0.096)
	ANE-MPNN	**0.558(±0.044)**	**0.664±(0.071)**



Vanilla GCN

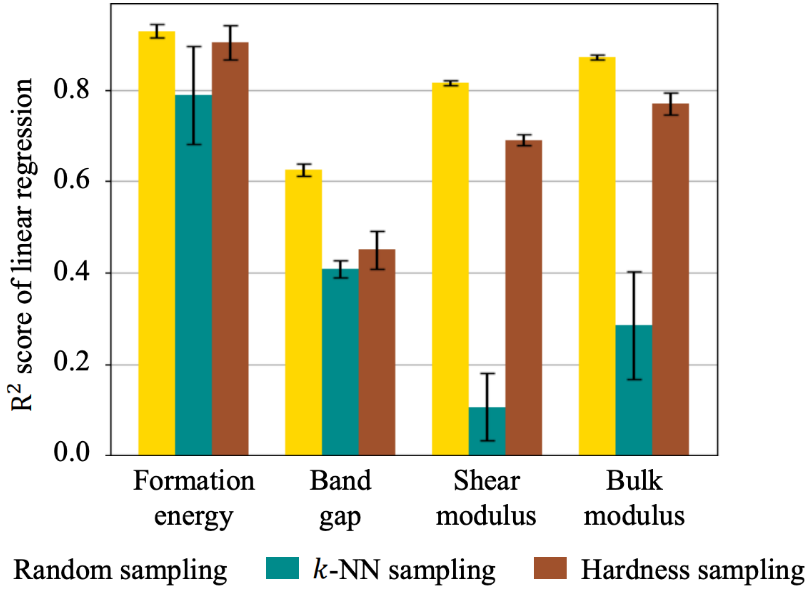


Ours (ANE-MPNN)

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

Sampling Strategies and Extrapolation

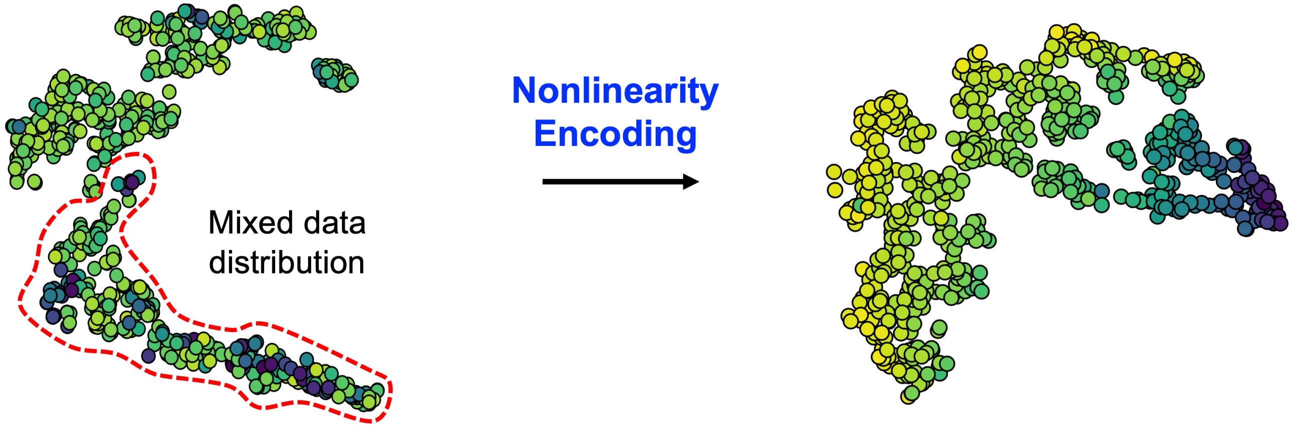
- Time complexity of the training process of ANE: $\theta^* = \underset{\theta}{\operatorname{argmin}} \sum_{i=1}^{N} \sum_{j=1}^{N} \left\| r_{ij} - u_{ij} \right\| \rightarrow \mathbf{O}(\boldsymbol{N}^{\mathbf{2}})$
- Three sampling strategies to reduce the time complexity:
  - **Random sampling:** 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
(∵ Random sampling = Density-based sampling)

Conclusion

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



Data distribution in the **original feature space**

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

- 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**

물질 간 화학 반응 예측 연구

16:00-17:00

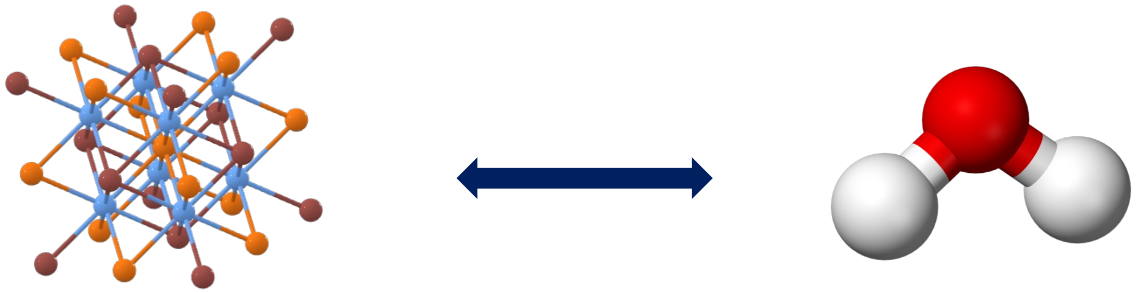
Table of Contents

시간	주제	주요 내용
1	그래프 신경망 개요	- 그래프 신경망 전반적인 소개 - 그래프 종류에 따른 다양한 그래프 신경망 소개
2	소재 물성 예측 연구	- 소재 물성 예측 연구 최신 동향 소개 - Transformer 기반 모델 소개 및 extrapolation을 위한 모델 소개
3	물질 간 화학 반응 예측 연구	- 물질 간 화학 반응 예측 연구 동향 소개 - 정보 이론(Information bottleneck) 및 인과추론(Causal inference) 기반 모델 소개

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

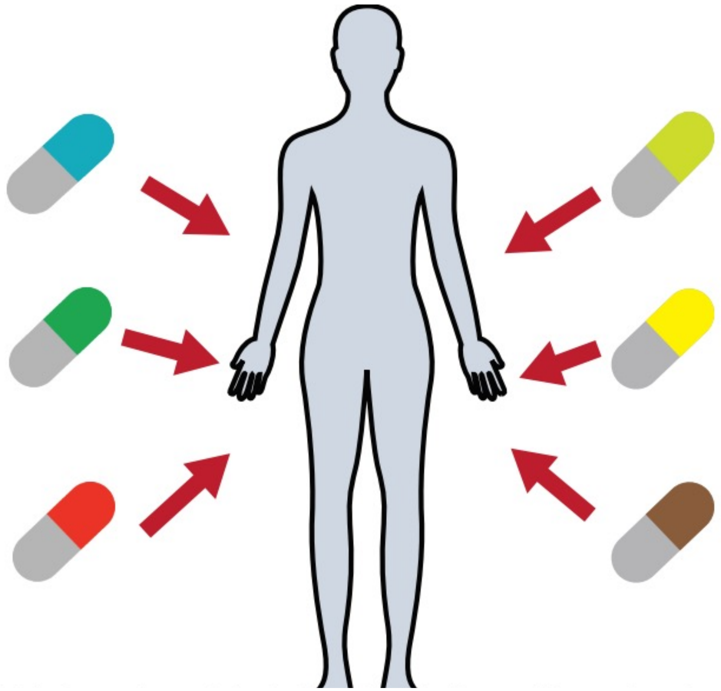
- General
    - Modeling polypharmacy side effects with graph convolutional networks. Bioinformatics 2018
    - Chemically interpretable graph interaction network for prediction of pharmacokinetic properties of drug-like molecules. AAAI 2020
    - SSI-DDI: substructure-substructure interactions for drug-drug interaction prediction. Briefings in Bioinformatics 2021
    - 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
  - SSI-DDI: substructure-substructure interactions for drug-drug interaction prediction. Briefings in Bioinformatics 2021
  - 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**

Polypharmarcy 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.

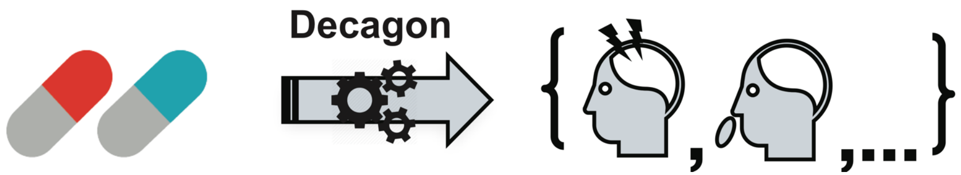


- Extremely difficult to identify
  - Impossible to test all combinations of drugs
  - Side effects not observed in controlled trials
- 15% of the U.S. population affected
  - Annual costs exceed $177 billion

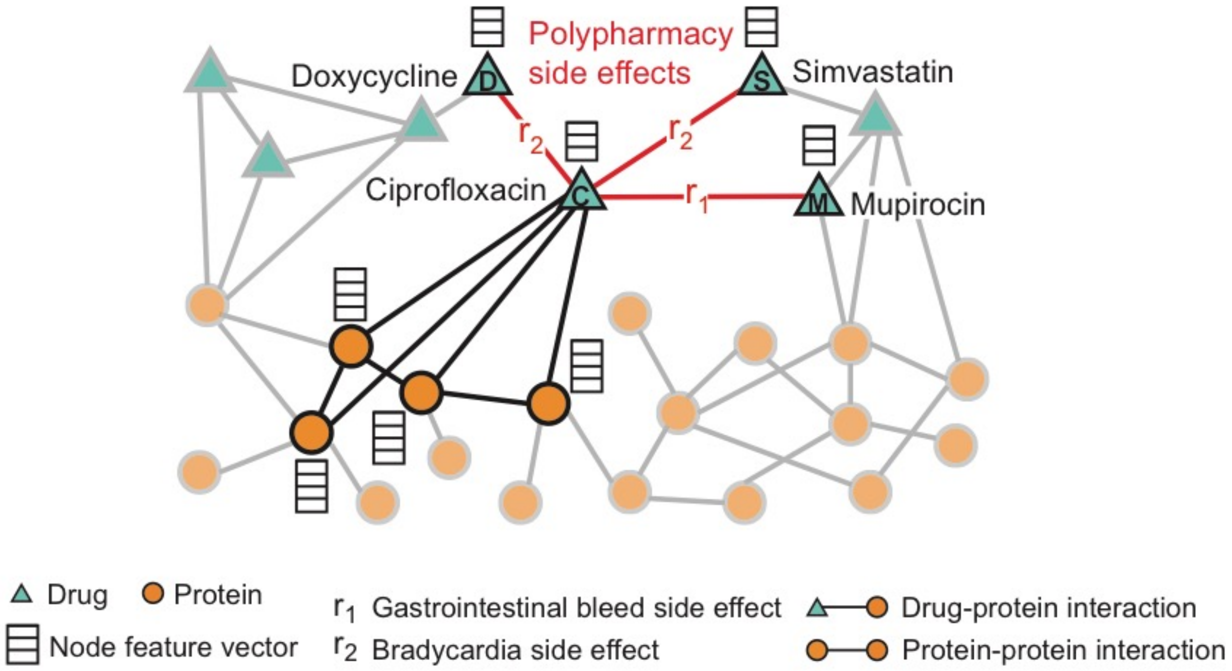
Charlesworth, Christina J., et al. "Polypharmacy among adults aged 65 years and older in the United States: 1988–2010." *Journals of Gerontology Series A: Biomedical Sciences and Medical Sciences* 70.8 (2015): 989-995.
Kantor, Elizabeth D., et al. "Trends in prescription drug use among adults in the United States from 1999-2012." *Jama* 314.17 (2015): 1818-1830.

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)

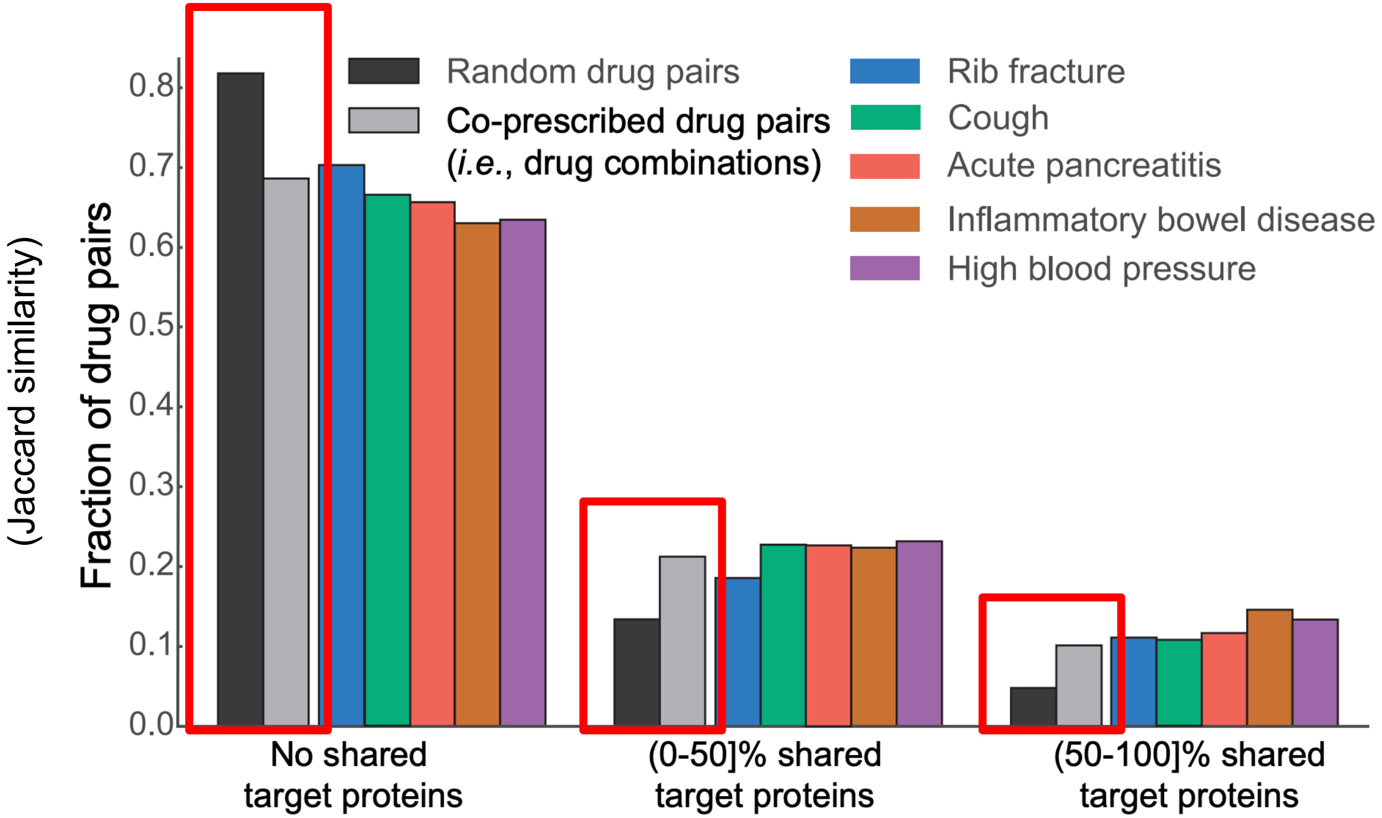

Polypharmacy side effects
Doxycycline
D
Simvastatin
S
r_2
r_2
Ciprofloxacin
C
r_1
M
Mupirocin
Drug
Protein
Node feature vector
r_1 Gastrointestinal bleed side effect
r_2 Bradycardia side effect
Drug-protein interaction
Protein-protein interaction

Multi-relational edge prediction model

Modeling polypharmacy side effects with graph convolutional networks. Bioinformatics 2018

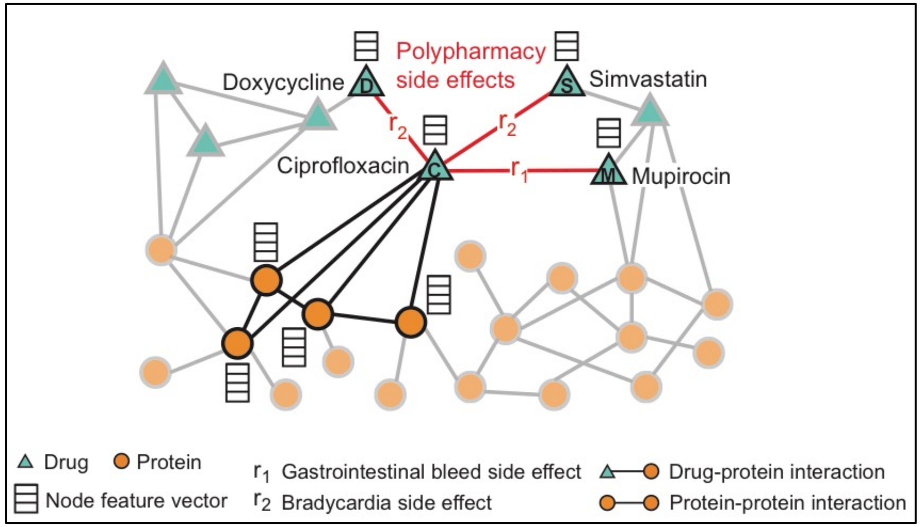
Decagon: Exploratory Data Analysis (EDA)

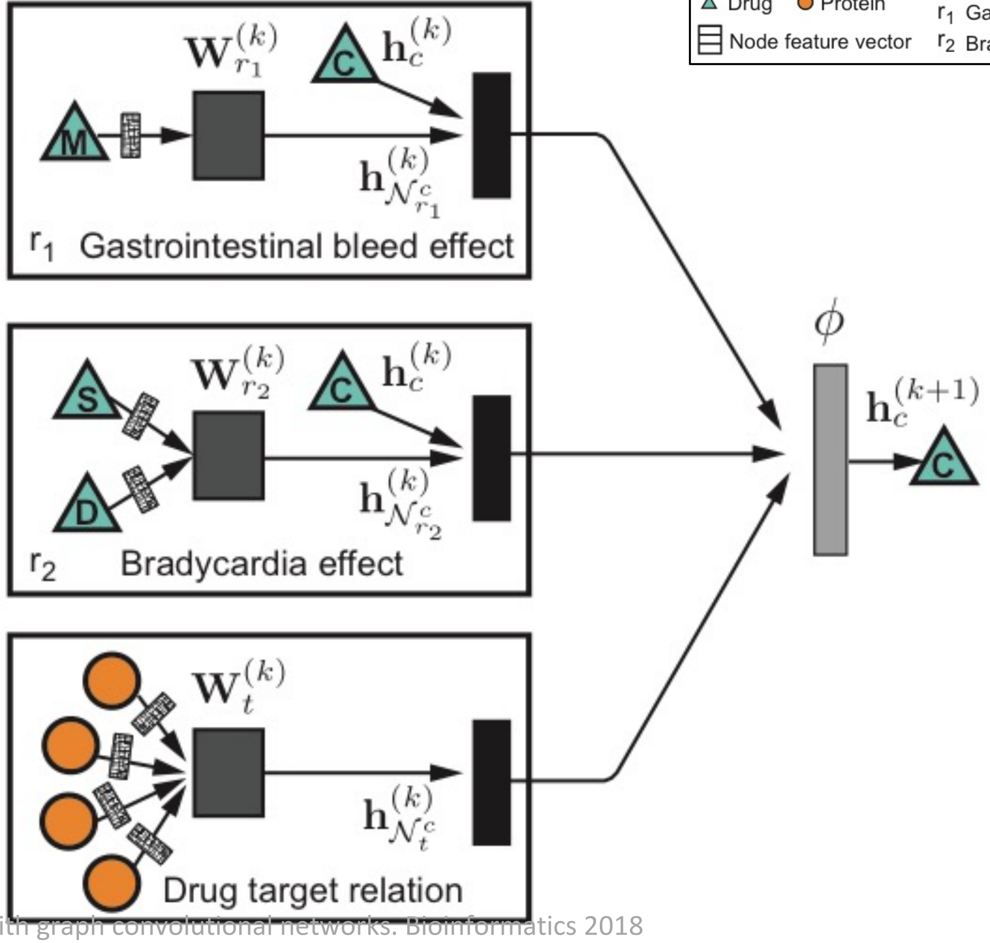
- Observation: Co-prescribed drugs (i.e. drug combinations) tend to have more target proteins in common than random drug pairs
  - It is important to consider how proteins interact with each other and to be able to model longer chains of (indirect) interactions.

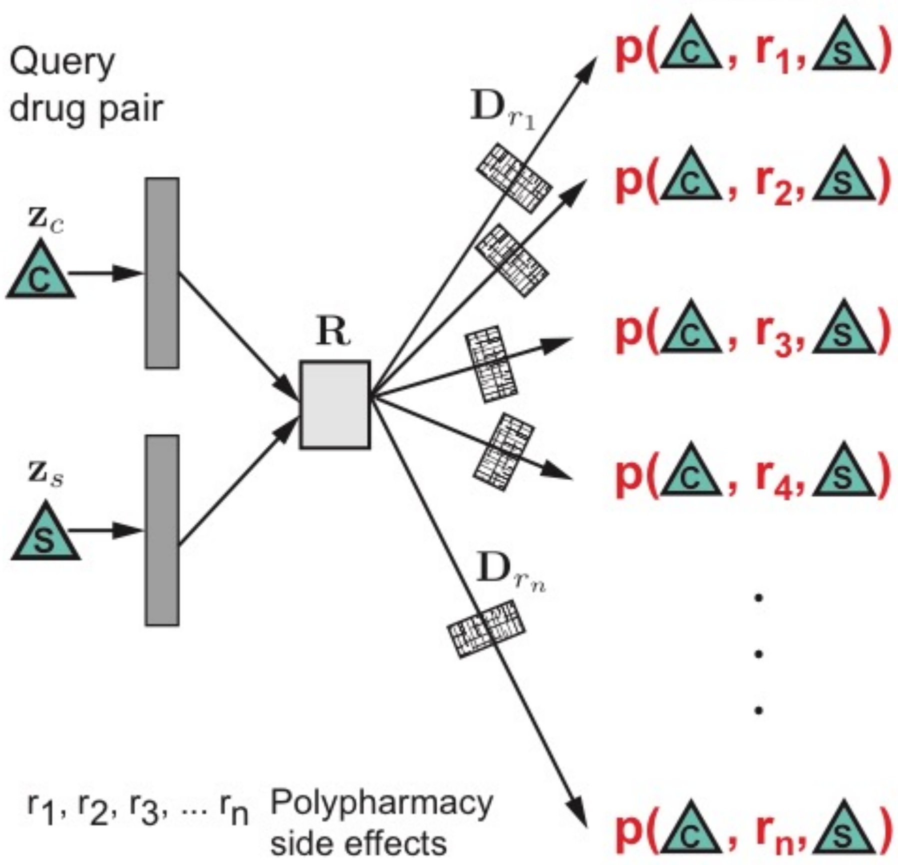


Modeling polypharmacy side effects with graph convolutional networks. Bioinformatics 2018

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


Polypharmacy side effects
Doxycycline
D
Simvastatin
S
r2
r2
Ciprofloxacin
C
r1
Mupirocin
M
Drug
Protein
Node feature vector
r1 Gastrointestinal bleed side effect
r2 Bradycardia side effect
Drug-protein interaction
Protein-protein interaction

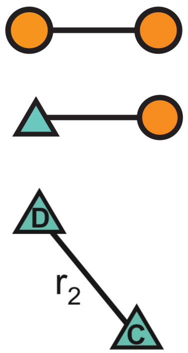




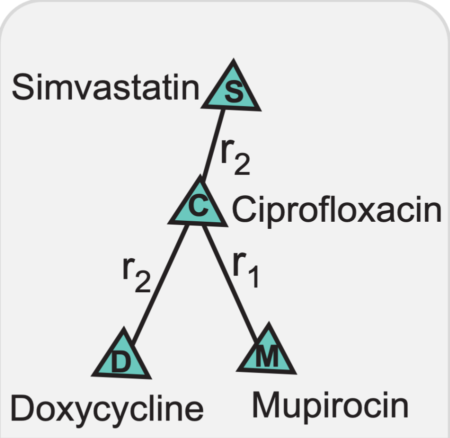
Modeling polypharmacy side effects with graph convolutional networks. Bioinformatics 2018

Decagon: Experiments

- Dataset
  - Protein-protein interactions: Physical interactions in humans [720 k edges]
  - Drug-protein relationships [19 k edges]
  - Side effects of drug pairs: National adverse event reporting system [4.6 M edges]
  - Additional side information
  - **Final graph has 966 different edge types**
    - Multi-relational link prediction



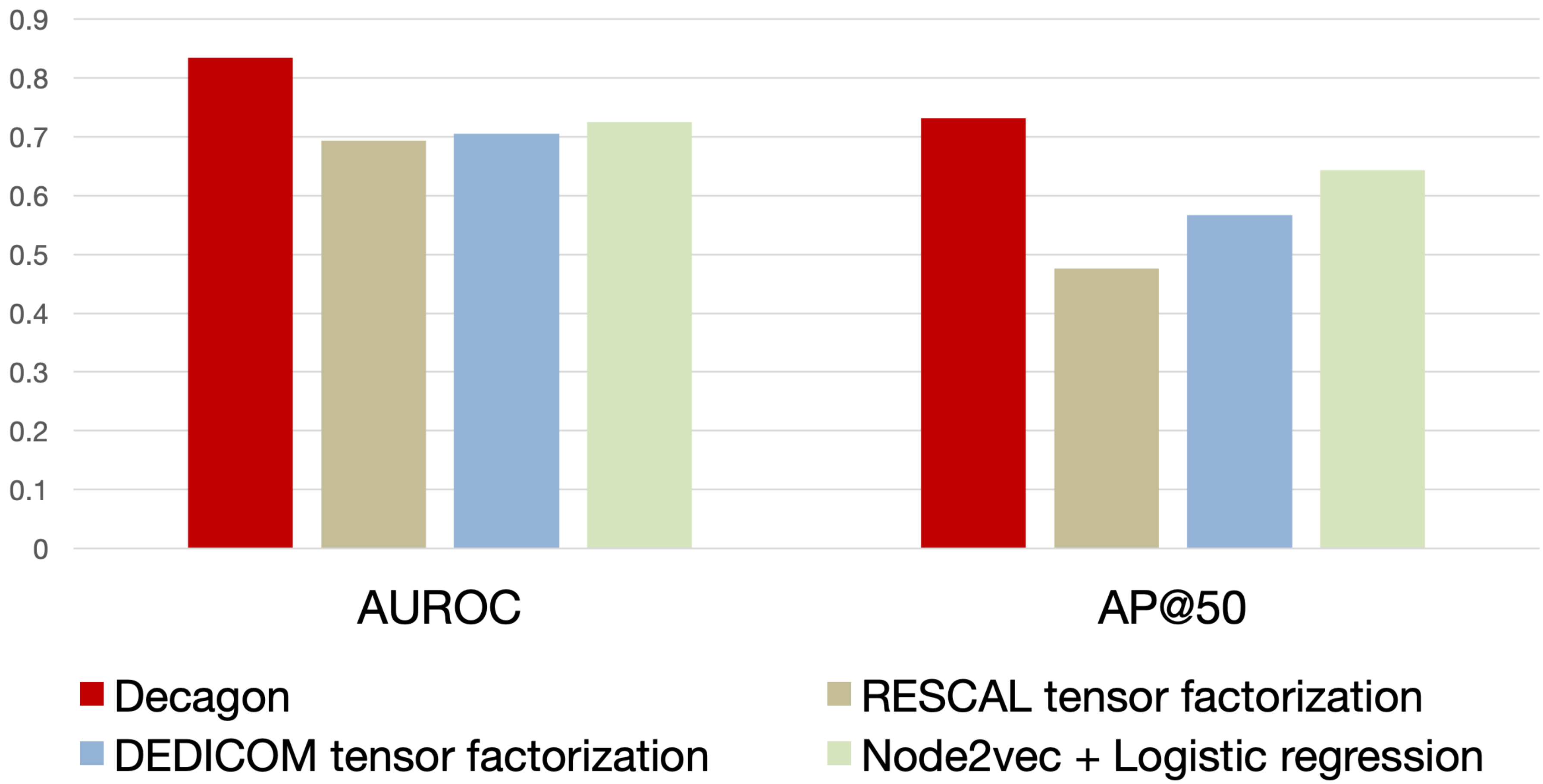
- Setup
  - Construct a heterogeneous graph of all the data
  - Side-effect centric evaluation:
    - Train: Fit a model on known side effects of drug pairs
    - Test: Given a query drug pair, predict all types of side effects



Drug pair c, d leads to side effect r_2

Modeling polypharmacy side effects with graph convolutional networks. Bioinformatics 2018

Decagon: Results (Side Effect Prediction)



36% average in AP@50 improvement over baselines

Modeling polypharmacy side effects with graph convolutional networks. Bioinformatics 2018

Decagon: Results

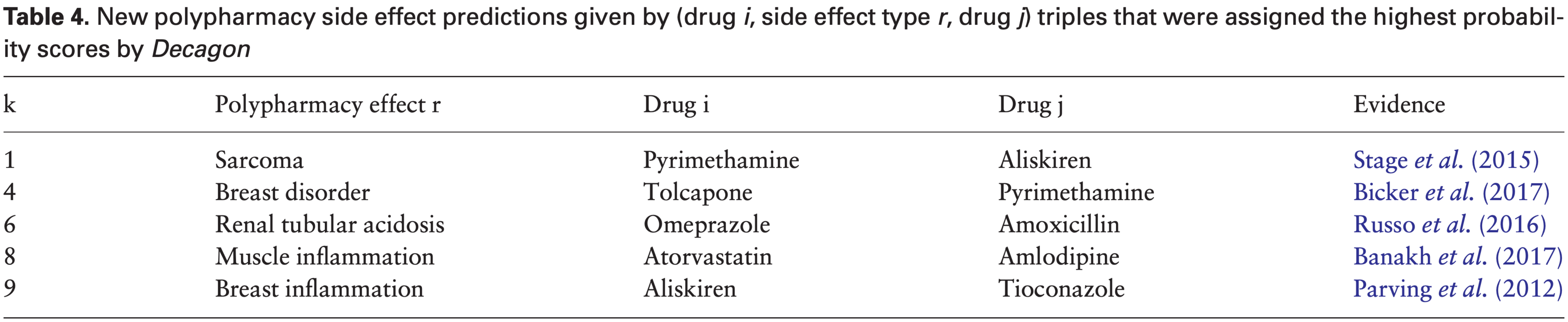
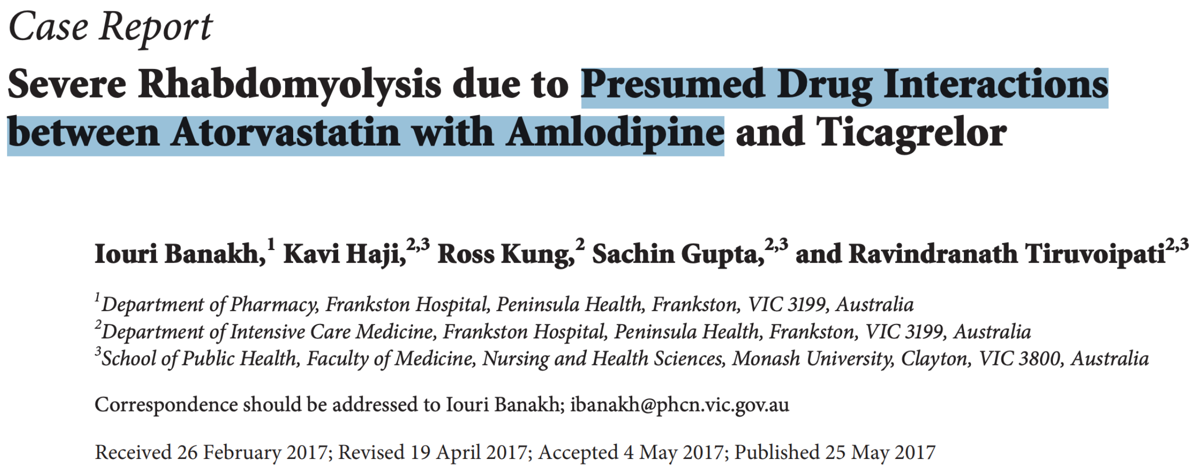


Table 4. New polypharmacy side effect predictions given by (drug *i*, side effect type *r*, drug *j*) triples that were assigned the highest probability scores by *Decagon*

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



Case Report

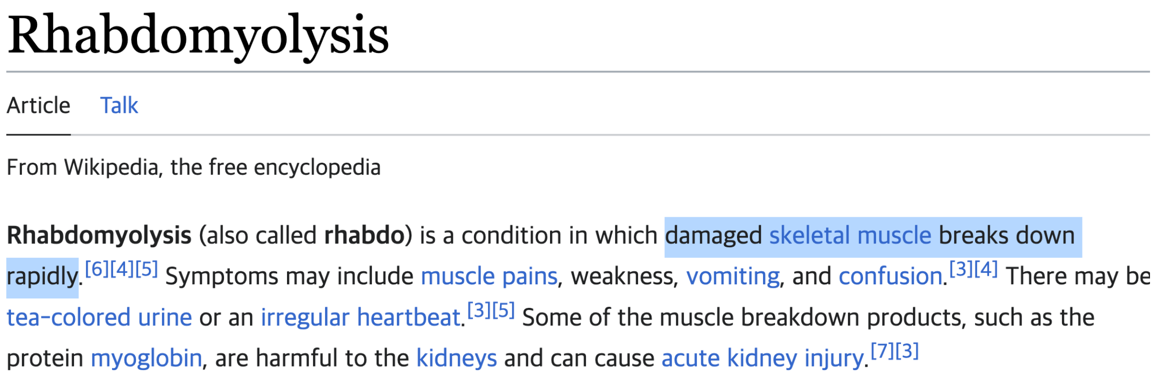
Severe Rhabdomyolysis due to Presumed Drug Interactions between Atorvastatin with Amlodipine and Ticagrelor

Iouri Banakh,[1] Kavi Haji,[2,3] Ross Kung,[2] Sachin Gupta,[2,3] and Ravindranath Tiruvoipati[2,3]

[1] *Department of Pharmacy, Frankston Hospital, Peninsula Health, Frankston, VIC 3199, Australia*
[2] *Department of Intensive Care Medicine, Frankston Hospital, Peninsula Health, Frankston, VIC 3199, Australia*
[3] *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

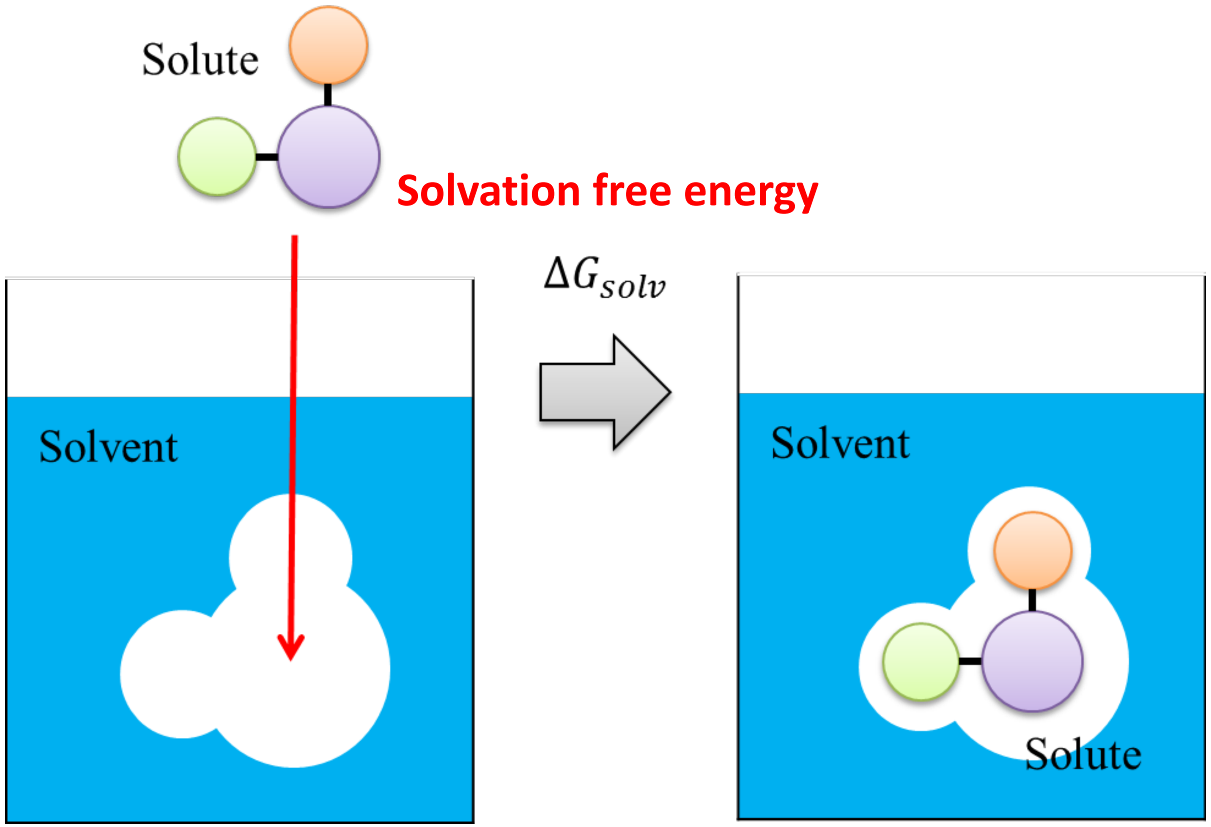
From Wikipedia, the free encyclopedia

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

Banakh, Iouri, et al. "Severe rhabdomyolysis due to presumed drug interactions between atorvastatin with amlodipine and ticagrelor." *Case reports in critical care* 2017 (2017).

Predicting Solvation Free Energy (용매화 자유 에너지)

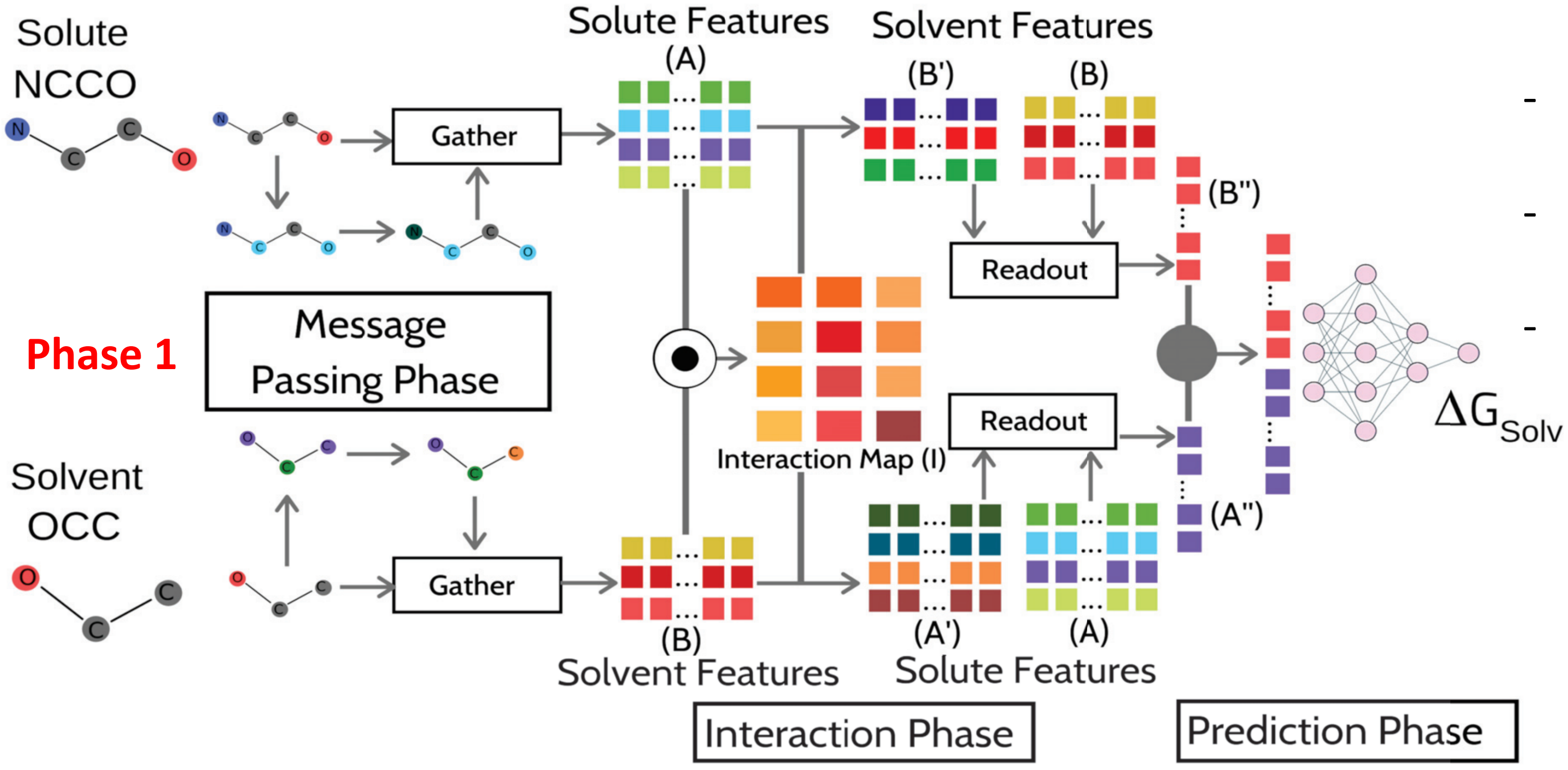
- **Solvation free energy**
  - Change in free energy for a molecule to be transferred from gas phase to a given solvent
  - Quantifies **solubility** of drug molecules
    - A large negative value → high solubility
    - A lower magnitudes/positive value → poor solubility



https://www.ibric.org/upload/geditor/201704/0.95691900_1493095501.png

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

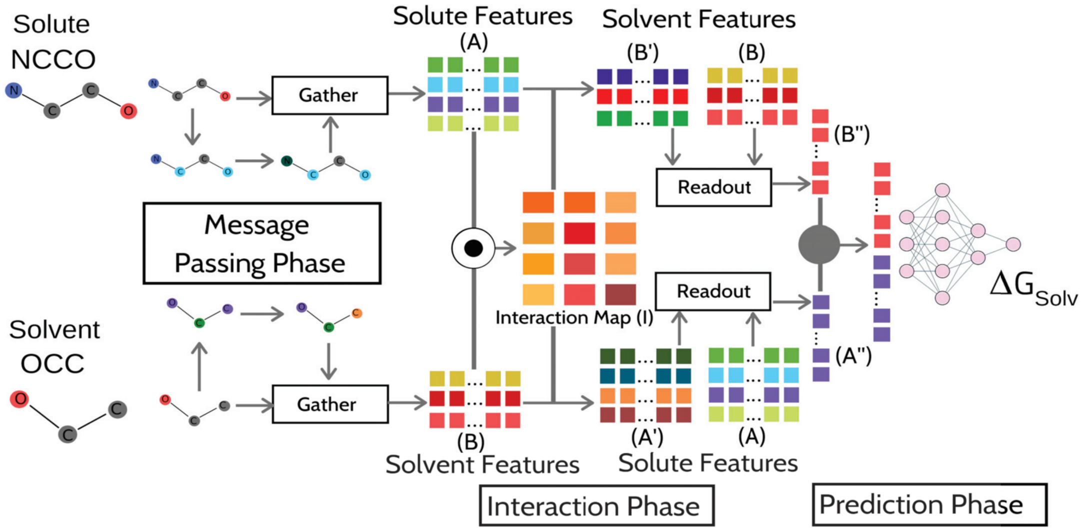
Phase 2

Phase 3

- **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

Chemically interpretable graph interaction network for prediction of pharmacokinetic properties of drug-like molecules. AAAI 2020

CIGIN: Model Architecture



- **Phase 1**: Message Passing Phase

Message function Node feature

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

Neighbors of v Edge feature Node update function

Final feature of v

$$F_v = g(x_v, h_v^t), \forall v \in V$$

set2set layer

- **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 Atom m of solvent

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

- **Phase 3**: Prediction Phase

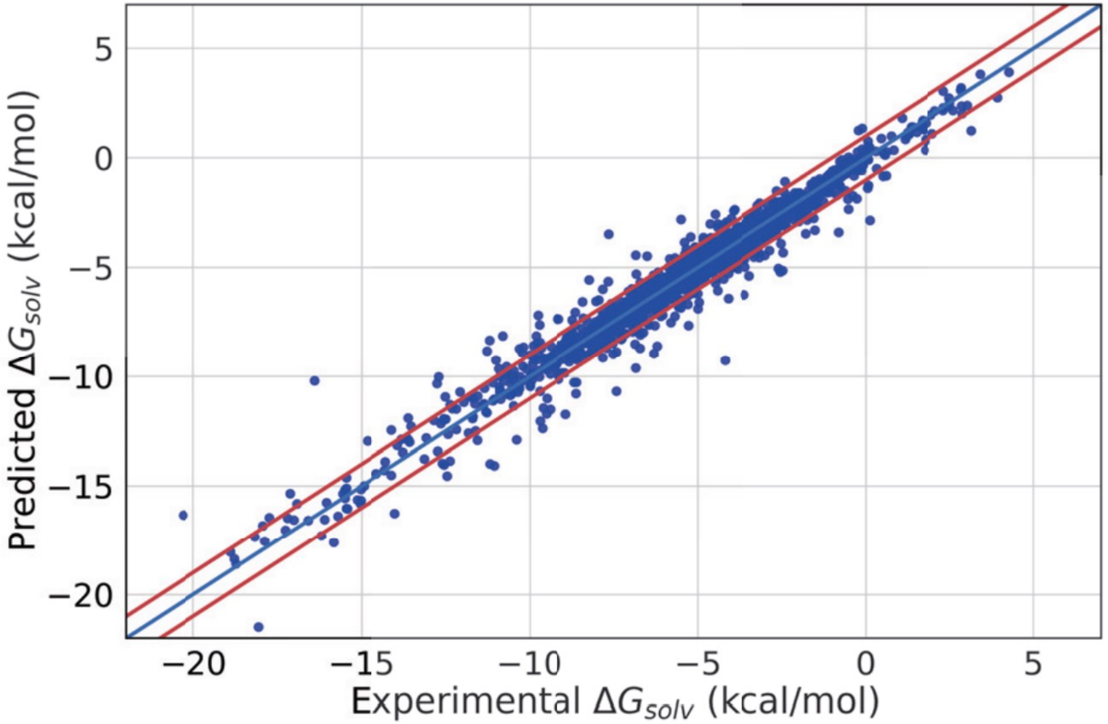
$$A'' = R_{solute}(A, A'), \quad B'' = R_{solvent}(B, B')$$

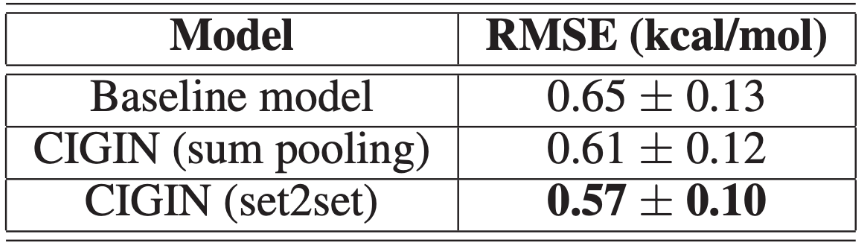
set2set layer set2set layer

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

Chemically interpretable graph interaction network for prediction of pharmacokinetic properties of drug-like molecules. AAAI 2020

CIGIN: Results





Model	RMSE (kcal/mol)
Baseline model	0.65 ± 0.13
CIGIN (sum pooling)	0.61 ± 0.12
CIGIN (set2set)	**0.57 ± 0.10**

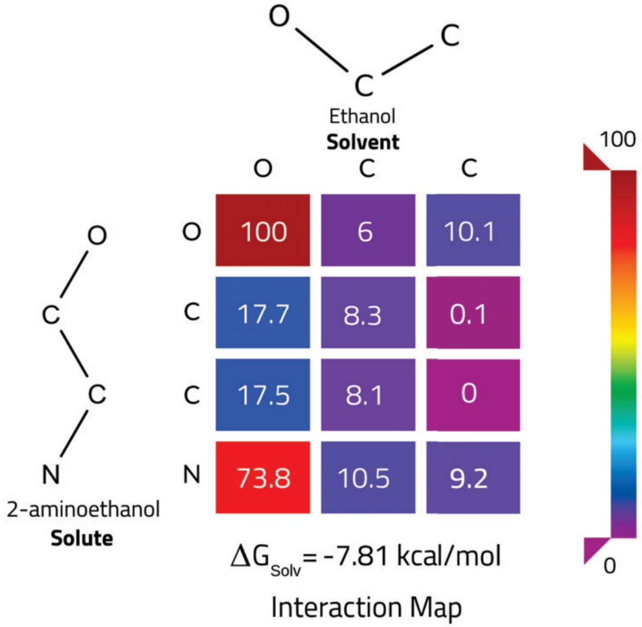
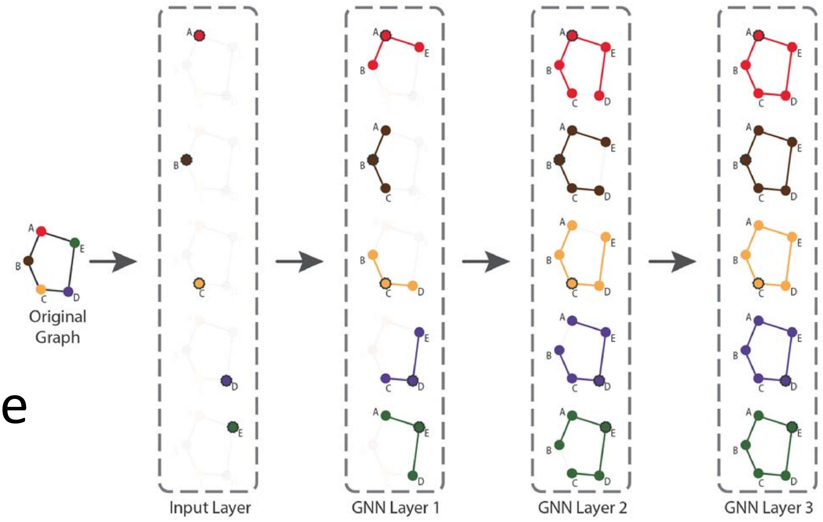


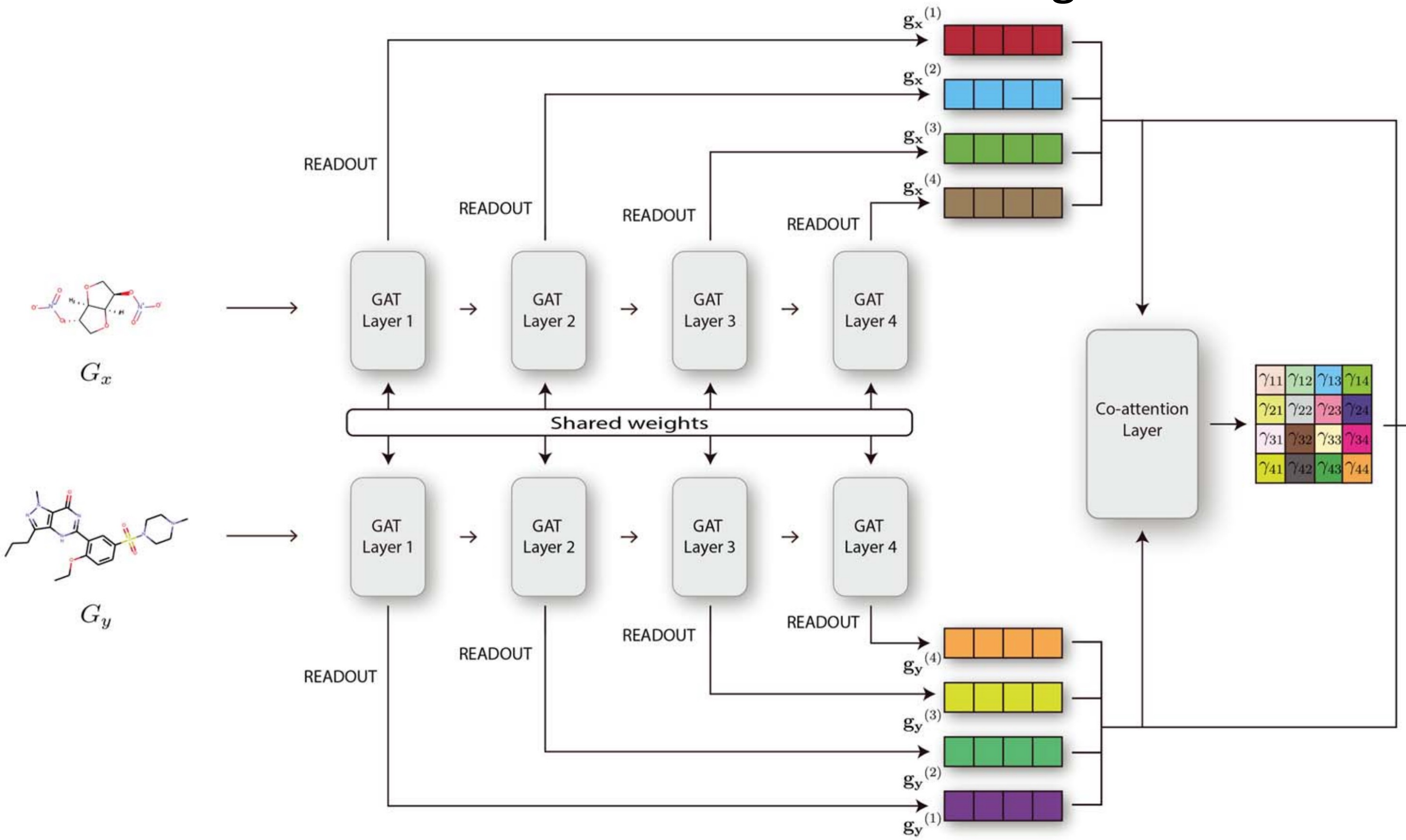
Figure 3: Heat map of the normalized (min-max) interaction map for 2-aminoethanol (solute) and ethanol(solvent) along with the predicted solvation free energy.

Chemically interpretable graph interaction network for prediction of pharmacokinetic properties of drug-like molecules. AAAI 2020

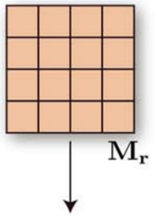
SSI–DDI

- Task: Predicting drug-drug interaction
- Key idea: Consider substructure instead of whole drugs' structure





G_x
G_y
GAT Layer 1
GAT Layer 2
GAT Layer 3
GAT Layer 4
Shared weights
READOUT
$g_x^{(1)}$
$g_x^{(2)}$
$g_x^{(3)}$
$g_x^{(4)}$
$g_y^{(4)}$
$g_y^{(3)}$
$g_y^{(2)}$
$g_y^{(1)}$
Co-attention Layer
γ_{11} γ_{12} γ_{13} γ_{14}
γ_{21} γ_{22} γ_{23} γ_{24}
γ_{31} γ_{32} γ_{33} γ_{34}
γ_{41} γ_{42} γ_{43} γ_{44}



$$p(G_x, G_y, r) = \sigma\left(\sum_i \sum_j \gamma_{ij} \mathbf{g_x}^{(i)^T} \mathbf{M_r} \mathbf{g_y}^{(j)}\right)$$

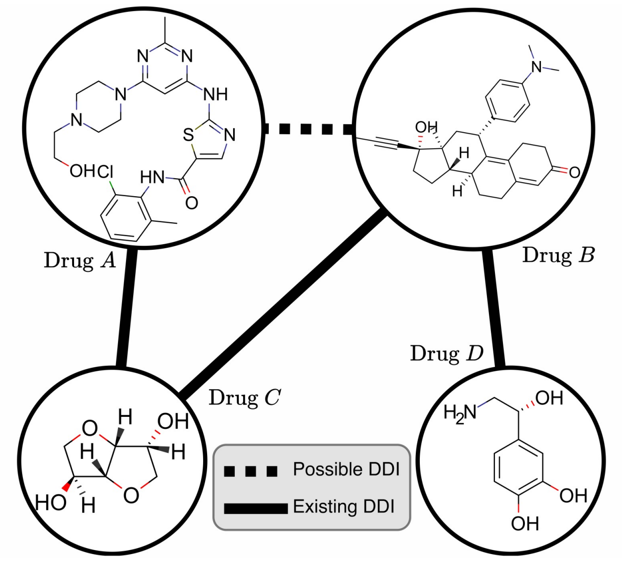
$$\gamma_{ij} = \mathbf{b}^T \tanh\left(\mathbf{W}_x \mathbf{g}_x^{(i)} + \mathbf{W}_y \mathbf{g}_y^{(j)}\right)$$

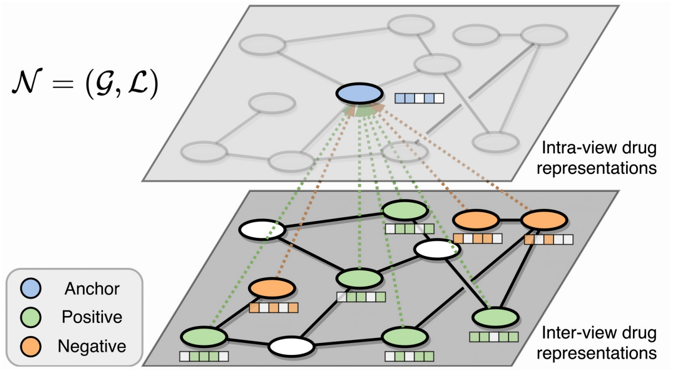
$$i = j = 1, \ldots, L,$$

SSI-DDI: substructure-substructure interactions for drug-drug interaction prediction. Briefings in Bioinformatics 2021

MIRACLE

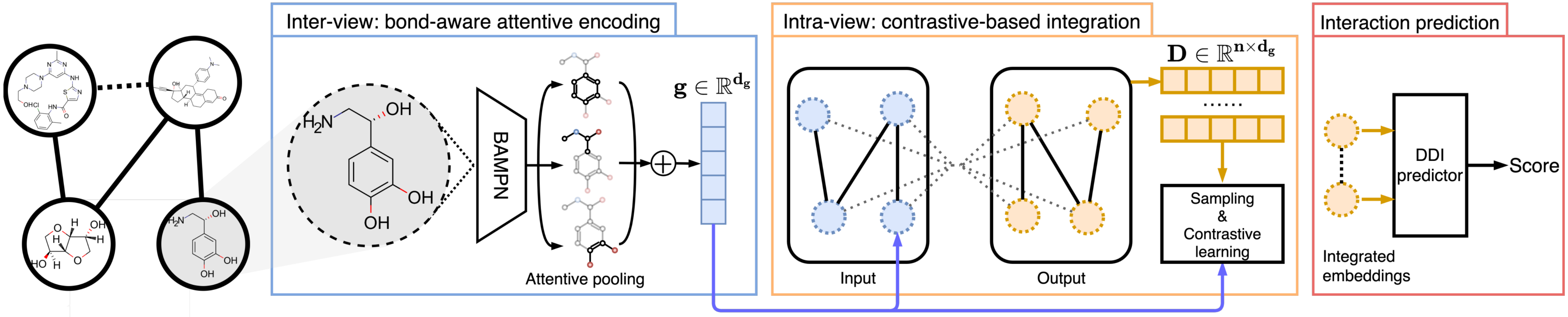
- Task: Predicting drug-drug interaction
- Key idea: Construct a graph-of-graphs





$\mathcal{N} = (\mathcal{G}, \mathcal{L})$
Intra-view drug representations
Anchor
Positive
Negative
Inter-view drug representations

Figure 3: The proposed graph contrastive learning framework.



Inter-view: bond-aware attentive encoding
BAMPN
Attentive pooling
$\mathbf{g} \in \mathbb{R}^{\mathbf{d_g}}$
Intra-view: contrastive-based integration
Input
Output
$\mathbf{D} \in \mathbb{R}^{\mathbf{n} \times \mathbf{d_g}}$
Sampling & Contrastive learning
Interaction prediction
DDI predictor
Score
Integrated embeddings

$$\tilde{\mathbf{h}}_{\mathbf{i}}^{(\mathbf{l})} = \sum_{\mathbf{j} \in C(\mathbf{i})} \mathbf{W}_{\mathbf{c_{ij}}}^{(\mathbf{l})} \mathbf{h}_{\mathbf{j}}^{(\mathbf{l-1})}$$

Trainable weight matrix shared by the same type of chemical bond c_{ij}

Multi-view graph contrastive representation learning for drug-drug interaction prediction. WWW 2021

Table of Contents

시간	주제	주요 내용
1	그래프 신경망 개요	- 그래프 신경망 전반적인 소개 - 그래프 종류에 따른 다양한 그래프 신경망 소개
2	소재 물성 예측 연구	- 소재 물성 예측 연구 최신 동향 소개 - Transformer 기반 모델 소개 및 extrapolation을 위한 모델 소개
3	물질 간 화학 반응 예측 연구	- 물질 간 화학 반응 예측 연구 동향 소개 - 정보 이론(Information bottleneck) 및 인과추론(Causal inference) 기반 모델 소개

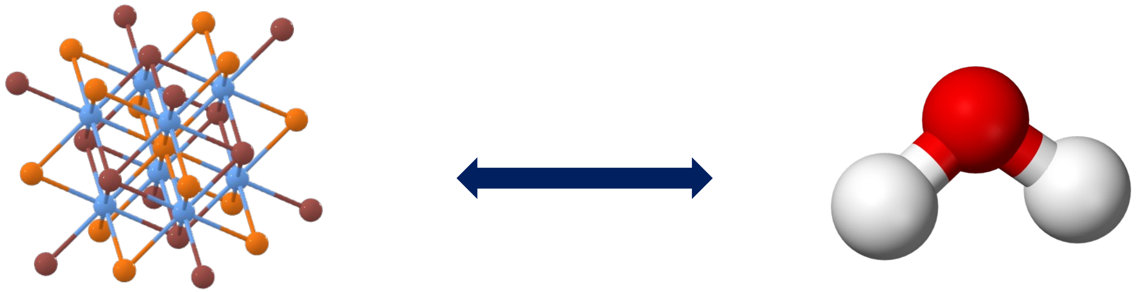
Papers

- 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
  - SSI-DDI: substructure-substructure interactions for drug-drug interaction prediction. Briefings in Bioinformatics 2021
  - 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**

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)

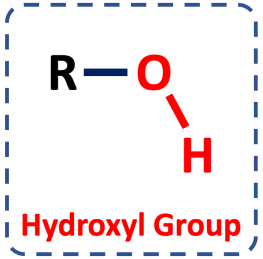
Introduction: Functional Group

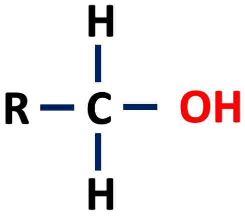
▪ Functional Groups

- 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

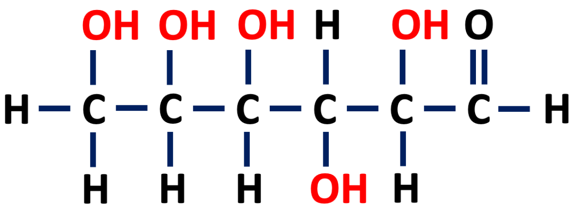
  → Molecules containing hydroxyl structures, such as alcohol and glucose, commonly have a high solubility in water

Functional Group





Alcohol

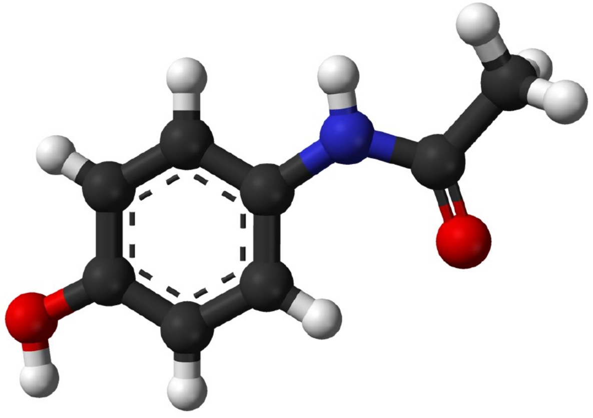


Glucose

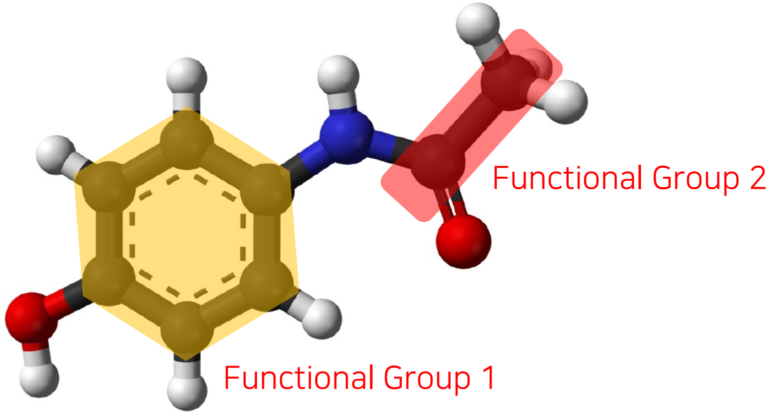
Hence, it is important to consider functional group for molecular relation learning

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

→ <u>Compression</u>

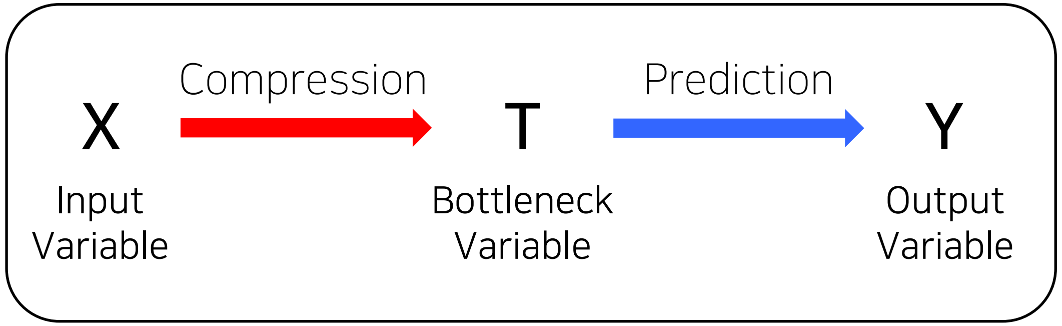
Maximize MI between T and Y

→ T should contain as much information about Y as possible

→ <u>Prediction</u>

Information Bottleneck Objective

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

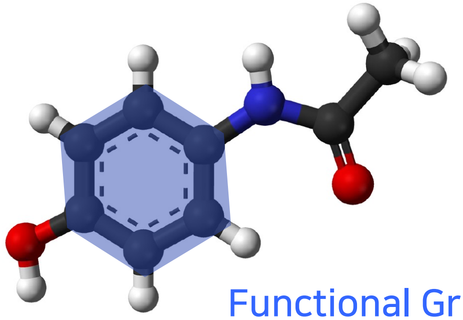


Graph Information Bottleneck: Overview

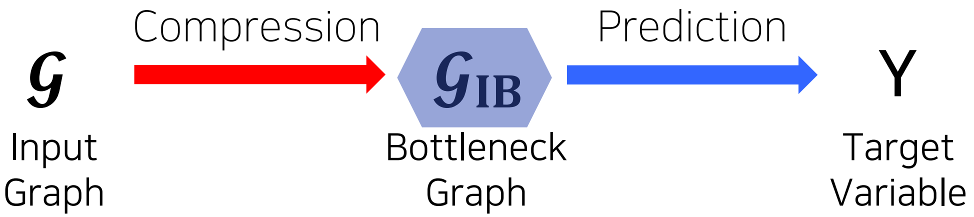
- How can we apply information bottleneck theory to graphs?
- Information Bottleneck Graph (IB-Graph)
  - To detect a subgraph that maximally preserves the property of the original graph
  - Subgraph becomes the bottleneck variable T

    → Problem formulation: Find Subgraph G_{IB} that is important for predicting Target Y

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



Functional Group

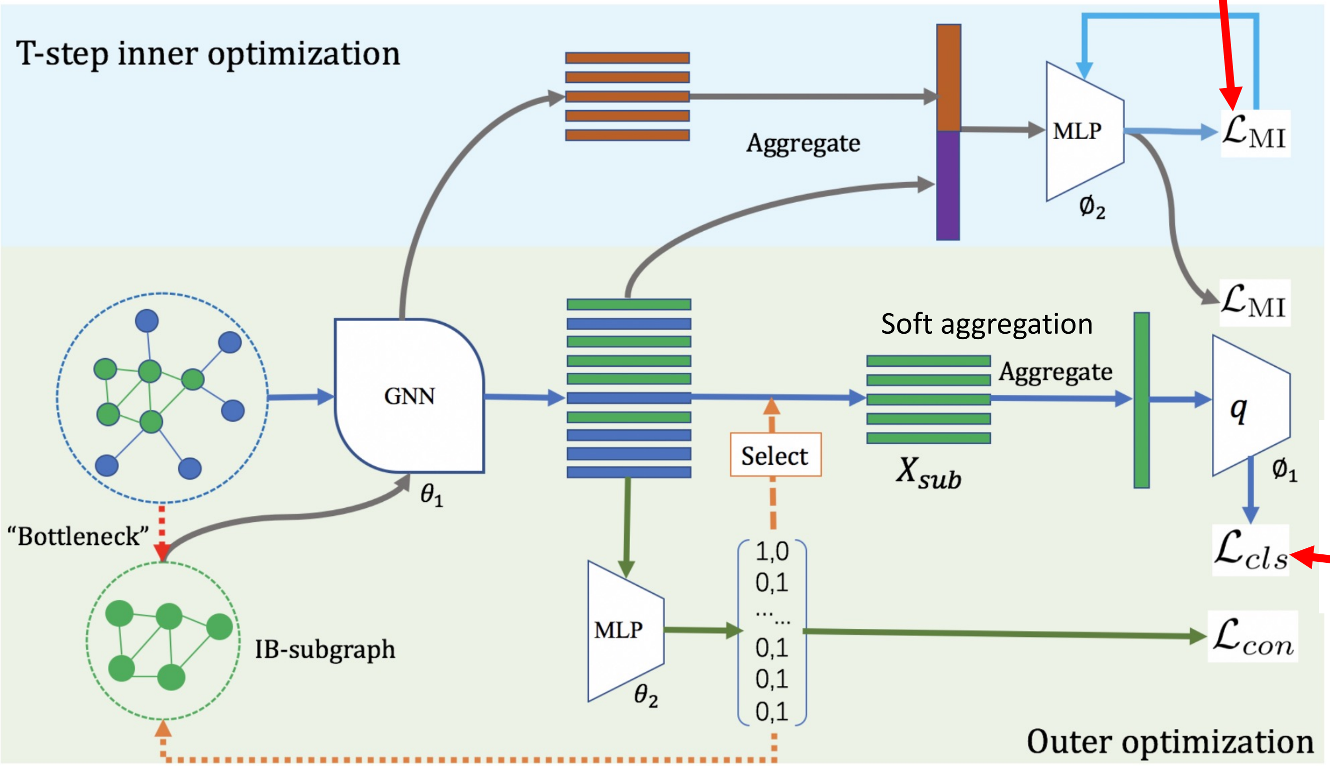


Graph Information Bottleneck: Existing studies (1/3)

GIB objective

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

$$\max_{\phi_2} \quad \mathcal{L}_{\text{MI}}(\phi_2,\mathcal{G}_{sub}) = \frac{1}{N}\sum_{i=1}^{N} f_{\phi_2}(\mathcal{G}_i,\mathcal{G}_{sub_i}) - \log\frac{1}{N}\sum_{i=1,j\neq i}^{N} e^{f_{\phi_2}(\mathcal{G}_i,\mathcal{G}_{sub_j})}$$



Final objective

$$\min_{\mathcal{G}_{sub},\phi_1} \quad \mathcal{L}(\mathcal{G}_{sub},\phi_1,\phi_2^*) = \mathcal{L}_{cls}(q_{\phi_1}(y|\mathcal{G}_{sub}),y_{gt}) + \beta\mathcal{L}_{\text{MI}}(\phi_2^*,\mathcal{G}_{sub})$$

$$\text{s.t.} \quad \phi_2^* = \underset{\phi_2}{\arg\max}\, \mathcal{L}_{\text{MI}}(\phi_2,\mathcal{G}_{sub}).$$

Perform T steps

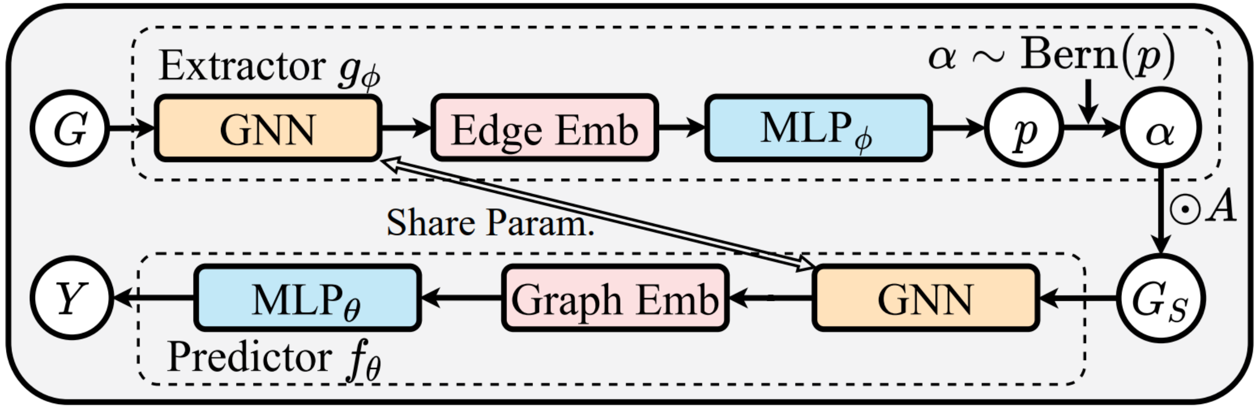
$$I(Y,\mathcal{G}_{sub}) \geq \int p(y,\mathcal{G}_{sub}) \log q_{\phi_1}(y|\mathcal{G}_{sub}) dy\, d\mathcal{G}_{sub}$$

$$\approx \frac{1}{N}\sum_{i=1}^{N} \log q_{\phi_1}(y_i|\mathcal{G}_{sub_i}) =: -\mathcal{L}_{cls}(q_{\phi_1}(y|\mathcal{G}_{sub}),y_{gt})$$

Graph information bottleneck for subgraph recognition. ICLR 2021

Graph Information Bottleneck: Existing studies (2/3)

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



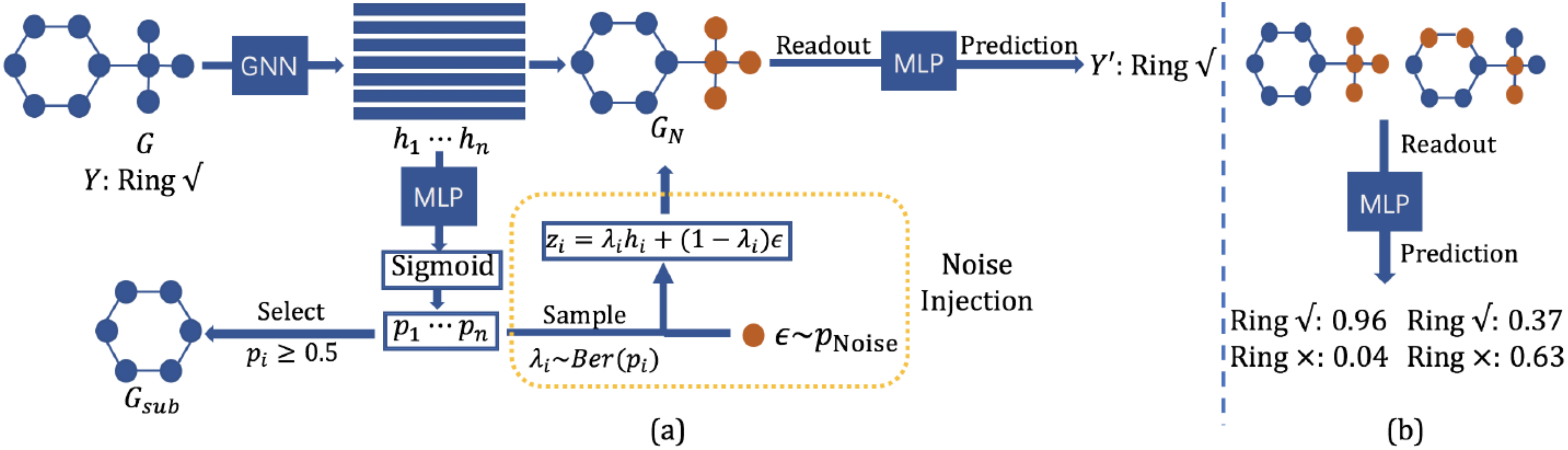
$$\min_{\phi} -I(G_S;Y) + \beta I(G_S;G), \text{ s.t. } G_S \sim g_\phi(G)$$

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

Miao, Siqi, Mia Liu, and Pan Li. "Interpretable and generalizable graph learning via stochastic attention mechanism." ICML 2022

Graph Information Bottleneck: Existing studies (3/3)

- 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

Yu, Junchi, Jie Cao, and Ran He. "Improving subgraph recognition with variational graph information bottleneck."CVPR 2022.

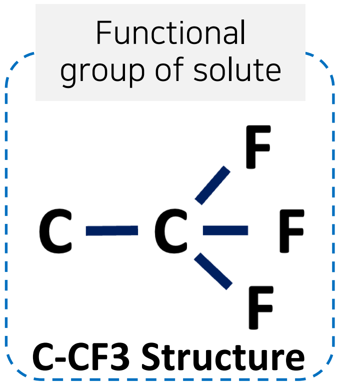
Conditional Graph Information Bottleneck for Molecular Relational Learning

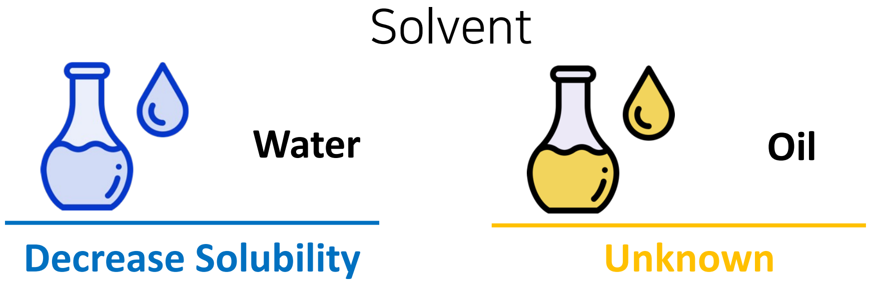
Namkyeong Lee, Dongmin Hyun, Gyoung S. Na, Sungwon Kim, Junseok Lee, Chanyoung Park

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-CF3 structure decreases the solubility of a molecule in water
    - However, it is unknown how C-CF3 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)

$$\mathcal{G}_{\text{IB}} = \underset{\mathcal{G}_{\text{IB}}}{\arg\min} -I(\mathrm{Y};\mathcal{G}_{\text{IB}}) + \beta I(\mathcal{G};\mathcal{G}_{\text{IB}})$$

Graph Information Bottleneck

↓

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

Conditional Graph Information Bottleneck (CGIB)

Proof of Lemma

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

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 $\mathcal{G}^1_n$ be a task irrelevant noise in the input graph $\mathcal{G}^1$. Then, the following inequality holds:*

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

Proof of Lemma 4.3

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

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

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

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

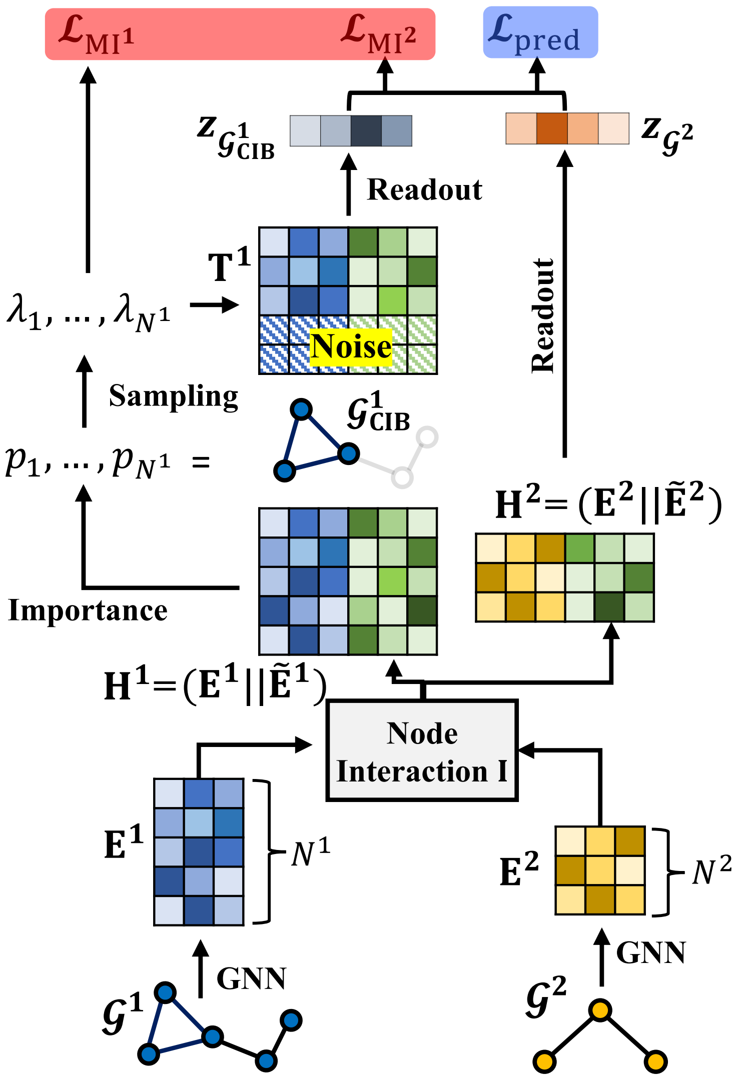
By plugging Equation (2) into Equation (1), we have:

$$I(\mathcal{G}^1; \mathcal{G}^1_{\mathrm{CIB}} | \mathcal{G}^2) \geq I(\mathcal{G}^1_{\mathrm{CIB}}; \mathcal{G}^1_n | \mathcal{G}^2) + I(Y; \mathcal{G}^1_{\mathrm{CIB}} | \mathcal{G}^2)$$

$$\therefore \; I(\mathcal{G}^1_{\mathrm{CIB}}; \mathcal{G}^1_n | \mathcal{G}^2) \leq -I(Y; \mathcal{G}^1_{\mathrm{CIB}} | \mathcal{G}^2) + I(\mathcal{G}^1; \mathcal{G}^1_{\mathrm{CIB}} | \mathcal{G}^2)$$

By minimizing the CGIB objective function, the model learns a CIB-Graph with the smallest mutual information with task-irrelevant noise.

Proposed Method: Conditional Graph Information Bottleneck



$\mathcal{L}_{\text{MI}^1}$
$\mathcal{L}_{\text{MI}^2}$
$\mathcal{L}_{\text{pred}}$
$\mathbf{z}_{\mathcal{G}^1_{\text{CIB}}}$
$\mathbf{z}_{\mathcal{G}^2}$
Readout
$\mathbf{T}^1$
$\lambda_1, \dots, \lambda_{N^1}$
Noise
Readout
Sampling
$\mathcal{G}^1_{\text{CIB}}$
$p_1, \dots, p_{N^1}$ =
$\mathbf{H}^2 = (\mathbf{E}^2 || \tilde{\mathbf{E}}^2)$
Importance
$\mathbf{H}^1 = (\mathbf{E}^1 || \tilde{\mathbf{E}}^1)$
Node Interaction I
$\mathbf{E}^1$
N^1
$\mathbf{E}^2$
N^2
GNN
GNN
$\mathcal{G}^1$
$\mathcal{G}^2$

Chain Rule of MI

$$I(X;Y,Z) = I(X;Z) + I(X;Y|Z)$$
$$= I(X;Y) + I(X;Z|Y)$$

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

Overall procedure:

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

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

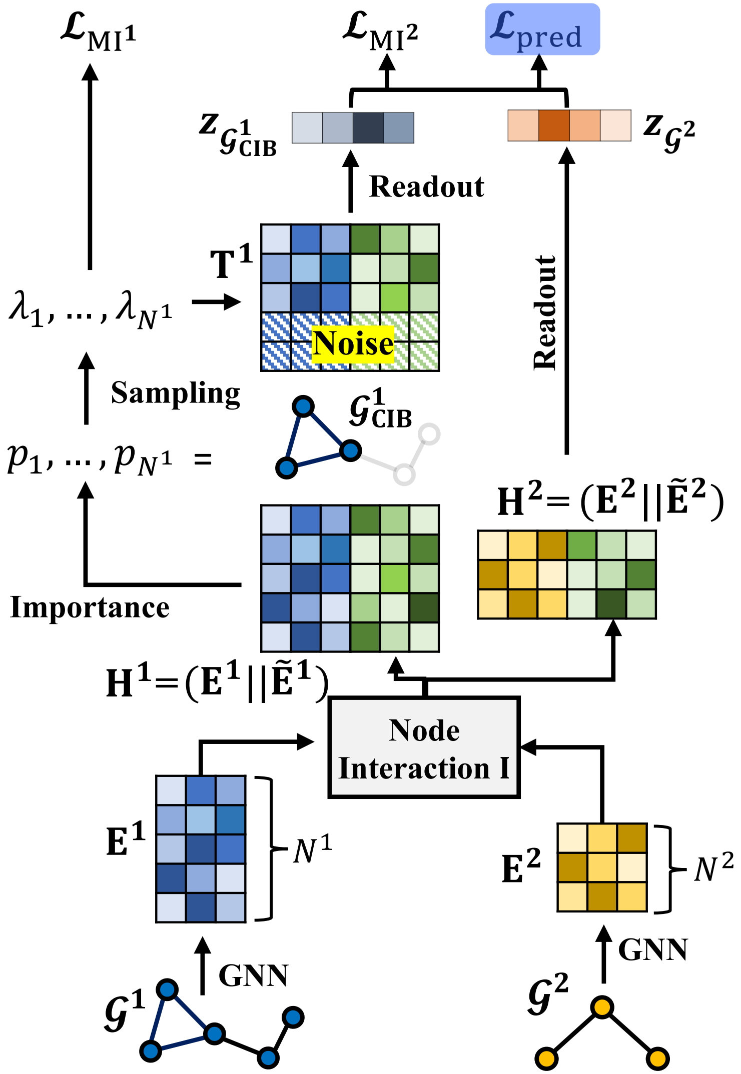
Prediction Loss

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

Compression Loss

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

Proposed Method: Conditional Graph Information Bottleneck



$\mathcal{L}_{\mathrm{MI}^1}$
$\mathcal{L}_{\mathrm{MI}^2}$
$\mathcal{L}_{\mathrm{pred}}$
$z_{\mathcal{G}^1_{\mathrm{CIB}}}$
$z_{\mathcal{G}^2}$
Readout
$\mathbf{T}^1$
$\lambda_1, \dots, \lambda_{N^1}$
Noise
Readout
Sampling
$\mathcal{G}^1_{\mathrm{CIB}}$
$p_1, \dots, p_{N^1}$ =
$\mathbf{H}^2 = (\mathbf{E}^2 || \tilde{\mathbf{E}}^2)$
Importance
$\mathbf{H}^1 = (\mathbf{E}^1 || \tilde{\mathbf{E}}^1)$
Node Interaction I
$\mathbf{E}^1$
N^1
$\mathbf{E}^2$
N^2
GNN
GNN
$\mathcal{G}^1$
$\mathcal{G}^2$

- Step 1: Optimizing the prediction loss

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

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

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

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

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

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

$$I(Y;\mathcal{G}^1_{\mathrm{CIB}},\mathcal{G}^2) = \mathbb{E}_{Y,\mathcal{G}^1_{\mathrm{CIB}},\mathcal{G}^2}\left[\log\frac{p(Y|\mathcal{G}^1_{\mathrm{CIB}},\mathcal{G}^2)}{p(Y)}\right]$$

$$I(X;Y) = \mathbb{E}_{p(x,y)}\left[\log\frac{p(x|y)}{p(x)}\right] = \mathbb{E}_{p(x,y)}\left[\log\frac{p(y|x)}{p(y)}\right]$$

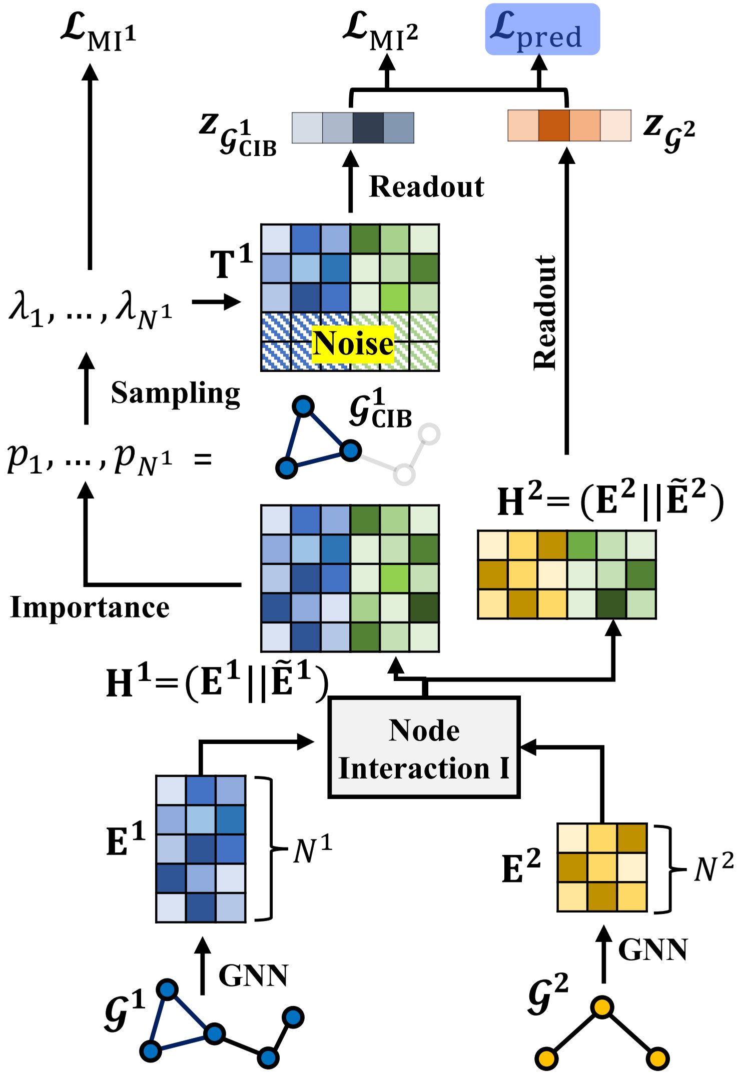
$$= \mathbb{E}_{Y,\mathcal{G}^1_{\mathrm{CIB}},\mathcal{G}^2}\left[\log\frac{p_\theta(Y|\mathcal{G}^1_{\mathrm{CIB}},\mathcal{G}^2)}{p(Y)}\right] + \mathbb{E}_{\mathcal{G}^1_{\mathrm{CIB}},\mathcal{G}^2}[p(Y|\mathcal{G}^1_{\mathrm{CIB}},\mathcal{G}^2)||p_\theta(Y|\mathcal{G}^1_{\mathrm{CIB}},\mathcal{G}^2)]$$

$$\ge \mathbb{E}_{Y,\mathcal{G}^1_{\mathrm{CIB}},\mathcal{G}^2}\left[\log\frac{p_\theta(Y|\mathcal{G}^1_{\mathrm{CIB}},\mathcal{G}^2)}{p(Y)}\right]$$ ∵ Non-negativity of KL divergence

$$= \mathbb{E}_{Y,\mathcal{G}^1_{\mathrm{CIB}},\mathcal{G}^2}[\log p_\theta(Y|\mathcal{G}^1_{\mathrm{CIB}},\mathcal{G}^2)] + H(Y)$$

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

Proposed Method: Conditional Graph Information Bottleneck



$\mathcal{L}_{\text{MI}^1}$
$\mathcal{L}_{\text{MI}^2}$
$\mathcal{L}_{\text{pred}}$
$z_{\mathcal{G}^1_{\text{CIB}}}$
$z_{\mathcal{G}^2}$
Readout
Readout
$\mathbf{T}^1$
$\lambda_1, \dots, \lambda_{N^1}$
Noise
Sampling
$\mathcal{G}^1_{\text{CIB}}$
$p_1, \dots, p_{N^1}$ =
$\mathbf{H}^2 = (\mathbf{E}^2 || \tilde{\mathbf{E}}^2)$
Importance
$\mathbf{H}^1 = (\mathbf{E}^1 || \tilde{\mathbf{E}}^1)$
Node Interaction I
$\mathbf{E}^1$
N^1
$\mathbf{E}^2$
N^2
GNN
GNN
$\mathcal{G}^1$
$\mathcal{G}^2$

- Step 1: Optimizing the prediction loss

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

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

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

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

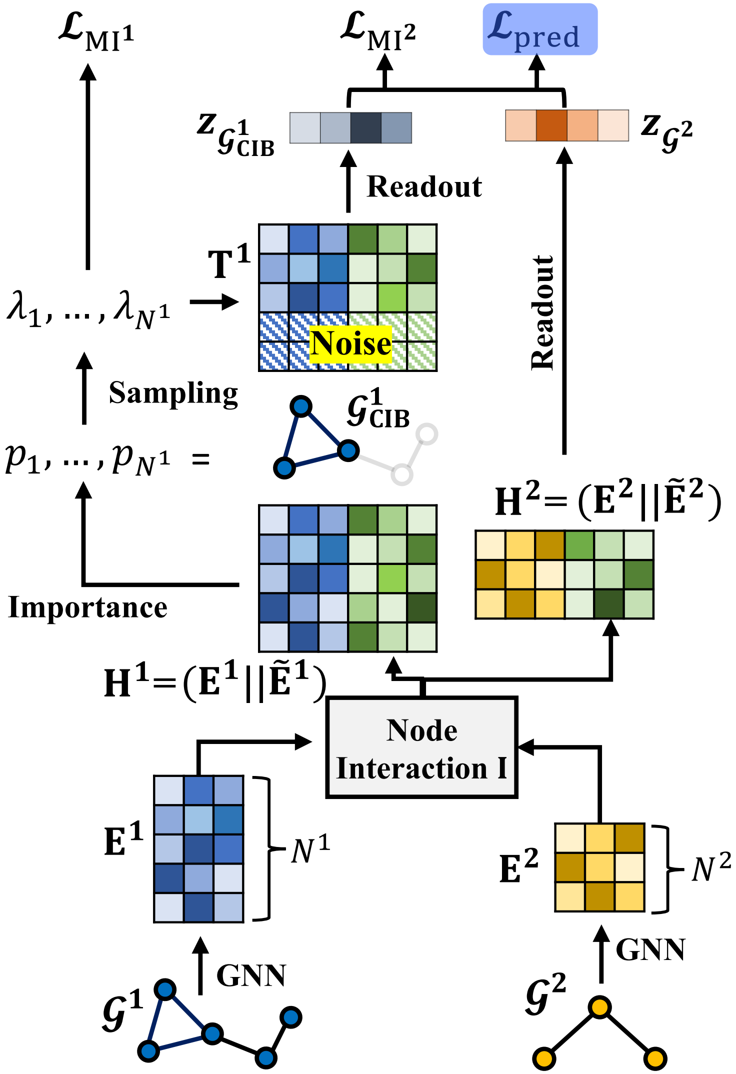
where $p_\theta(Y|\mathcal{G}^1_{\text{CIB}},\mathcal{G}^2)$ is variational approximation of $p(Y|\mathcal{G}^1_{\text{CIB}},\mathcal{G}^2)$.

Implementation.

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

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

Proposed Method: Conditional Graph Information Bottleneck



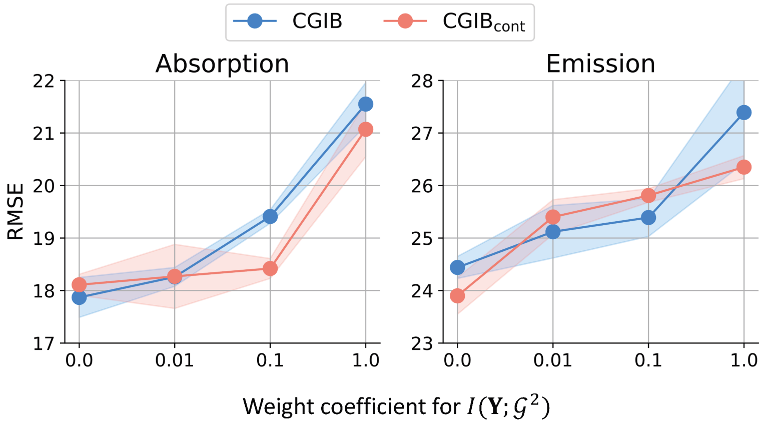
$\mathcal{L}_{\mathrm{MI}^1}$
$\mathcal{L}_{\mathrm{MI}^2}$
$\mathcal{L}_{\mathrm{pred}}$
$z_{\mathcal{G}^1_{\mathrm{CIB}}}$
$z_{\mathcal{G}^2}$
Readout
$\mathbf{T}^1$
$\lambda_1,\ldots,\lambda_{N^1}$
Noise
Readout
Sampling
$\mathcal{G}^1_{\mathrm{CIB}}$
$p_1,\ldots,p_{N^1}$ =
$\mathbf{H}^2 = (\mathbf{E}^2||\tilde{\mathbf{E}}^2)$
Importance
$\mathbf{H}^1 = (\mathbf{E}^1||\tilde{\mathbf{E}}^1)$
Node Interaction I
$\mathbf{E}^1$
N^1
$\mathbf{E}^2$
N^2
GNN
GNN
$\mathcal{G}^1$
$\mathcal{G}^2$

- Step 1: Optimizing the prediction loss

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

∵ Chain rule of mutual information

The 2nd term is empirically found to be not helpful



We treat $r(\mathrm{Y})$ as fixed spherical Gaussian,

$$I(Y;\mathcal{G}^2) \le \mathbb{E}_{\mathcal{G}^2}\left[KL(p_\xi(Y|\mathcal{G}^2)||r(Y))\right]$$

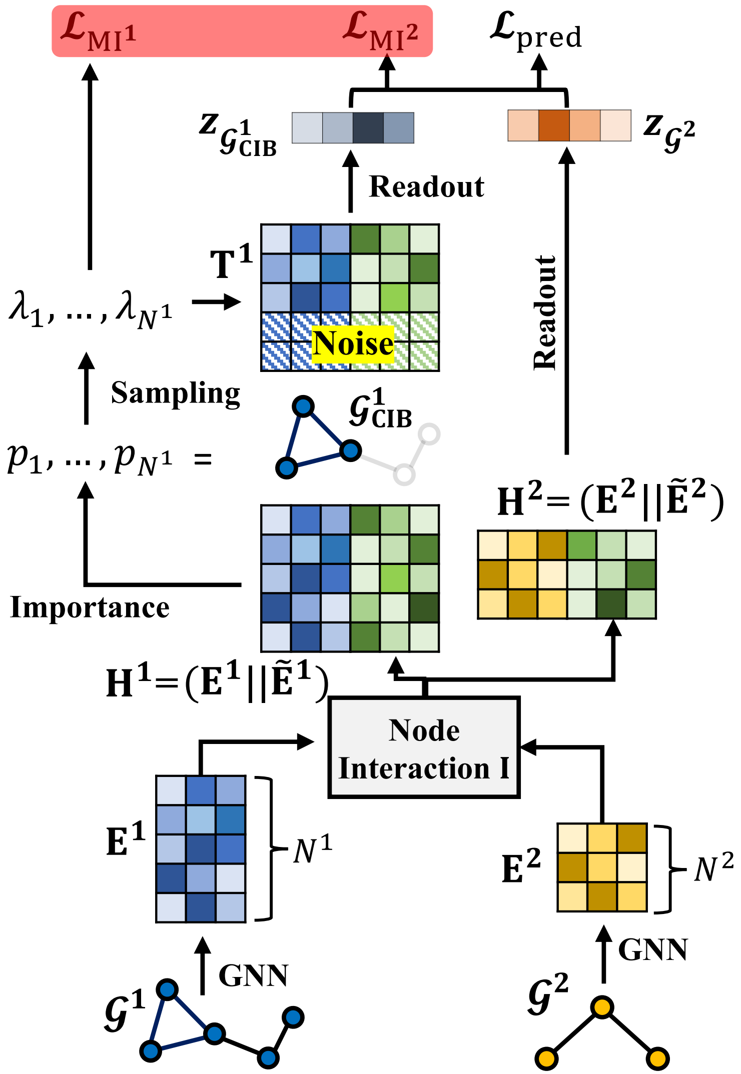
where $r(Y) \sim N(\mathrm{Y}|0,1)$

Increasing the contribution of this term deteriorates the model performance

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

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

Proposed Method: Conditional Graph Information Bottleneck



$\mathcal{L}_{\text{MI}^1}$
$\mathcal{L}_{\text{MI}^2}$
$\mathcal{L}_{\text{pred}}$
$z_{\mathcal{G}^1_{\text{CIB}}}$
$z_{\mathcal{G}^2}$
Readout
T^1
$\lambda_1, \dots, \lambda_{N^1}$
Noise
Readout
Sampling
$\mathcal{G}^1_{\text{CIB}}$
$p_1, \dots, p_{N^1}$ =
$H^2 = (E^2 || \tilde{E}^2)$
Importance
$H^1 = (E^1 || \tilde{E}^1)$
Node Interaction I
E^1
N^1
E^2
N^2
GNN
GNN
$\mathcal{G}^1$
$\mathcal{G}^2$

- Step 2: Optimizing the compression loss

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

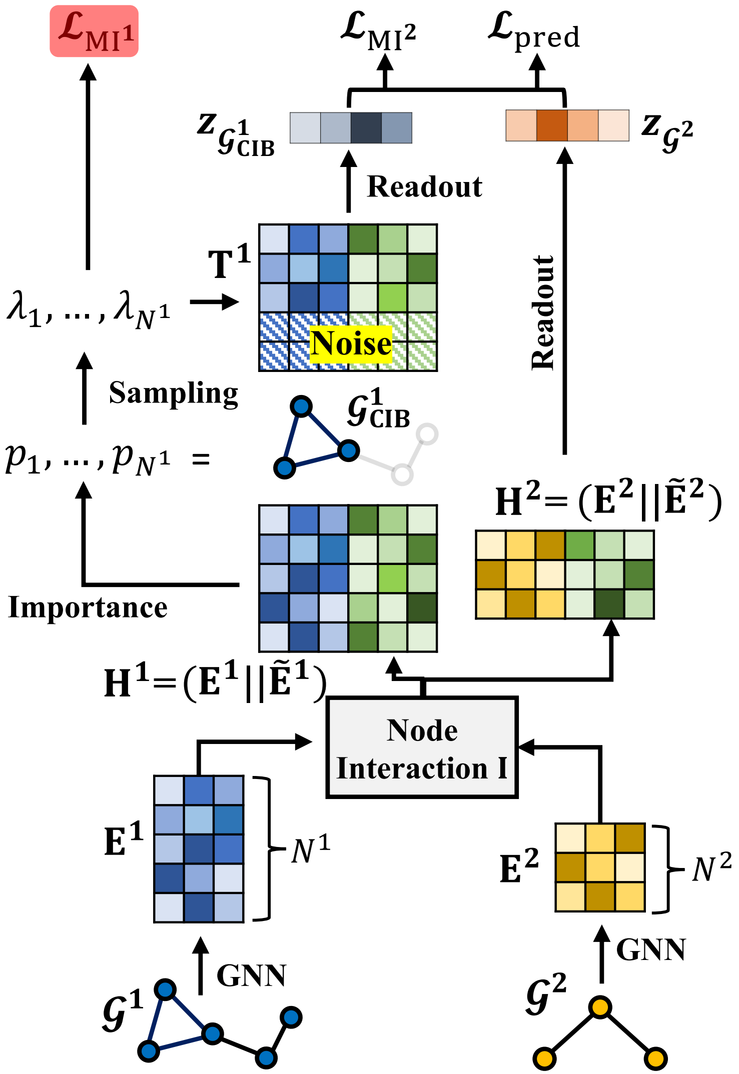
∵ Chain rule of mutual information

- L_{MI^1}: Compression through Noise Injection
  * Injecting noise into unimportant nodes
  → Remaining nodes are important nodes

- L_{MI^2}: Solute Prediction
  * Encourage $\mathcal{G}^1_{\text{CIB}}$, 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

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

Proposed Method: Conditional Graph Information Bottleneck



$\mathcal{L}_{\mathrm{MI}^1}$
$\mathcal{L}_{\mathrm{MI}^2}$
$\mathcal{L}_{\mathrm{pred}}$
$z_{\mathcal{G}^1_{\mathrm{CIB}}}$
$z_{\mathcal{G}^2}$
Readout
Readout
$\mathbf{T}^1$
$\lambda_1, \ldots, \lambda_{N^1}$
Noise
Sampling
$\mathcal{G}^1_{\mathrm{CIB}}$
$p_1, \ldots, p_{N^1}$ =
$\mathbf{H}^2 = (\mathbf{E}^2 || \tilde{\mathbf{E}}^2)$
Importance
$\mathbf{H}^1 = (\mathbf{E}^1 || \tilde{\mathbf{E}}^1)$
Node Interaction I
$\mathbf{E}^1$
N^1
$\mathbf{E}^2$
N^2
GNN
GNN
$\mathcal{G}^1$
$\mathcal{G}^2$

- Step 2: Optimizing the compression loss

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

1. Compression through Noise Injection

* Injecting noise into unimportant nodes

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

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

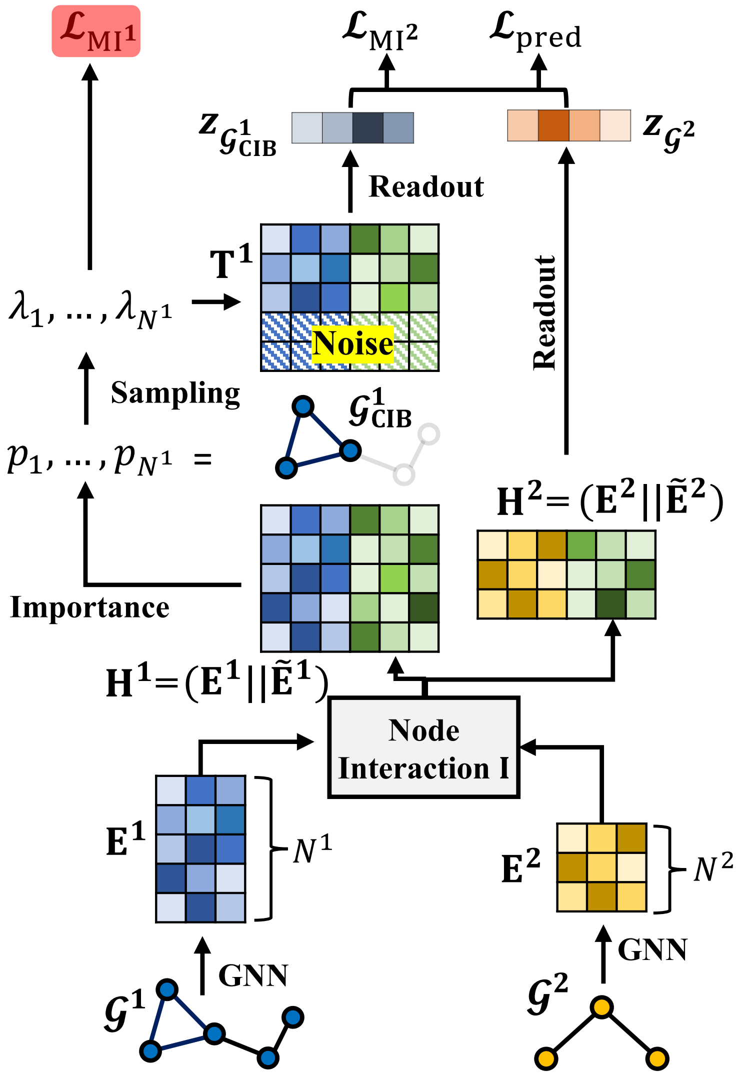
Upper bound of $\mathrm{I}(\mathcal{G}^1_{\mathrm{CIB}};\mathcal{G}^1,\mathcal{G}^2)$

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

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

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

Proposed Method: Conditional Graph Information Bottleneck


$\mathcal{L}_{\mathrm{MI}^1}$
$\mathcal{L}_{\mathrm{MI}^2}$
$\mathcal{L}_{\mathrm{pred}}$
$z_{\mathcal{G}^1_{\mathrm{CIB}}}$
$z_{\mathcal{G}^2}$
Readout
Readout
$\mathbf{T}^1$
$\lambda_1,\ldots,\lambda_{N^1}$
Noise
Sampling
$\mathcal{G}^1_{\mathrm{CIB}}$
$p_1,\ldots,p_{N^1}$ =
Importance
$\mathbf{H}^2 = (\mathbf{E}^2||\tilde{\mathbf{E}}^2)$
$\mathbf{H}^1 = (\mathbf{E}^1||\tilde{\mathbf{E}}^1)$
Node Interaction I
$\mathbf{E}^1$
N^1
$\mathbf{E}^2$
N^2
GNN
GNN
$\mathcal{G}^1$
$\mathcal{G}^2$

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

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

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

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

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

$$\mathrm{I}\left(z_{\mathcal{G}^1_{\mathrm{CIB}}};\mathcal{G}^1,\mathcal{G}^2\right) = \mathbb{E}_{z_{\mathcal{G}^1_{\mathrm{CIB}}},\mathcal{G}^1,\mathcal{G}^2}\left[-\log\frac{p_\phi\,(z_{\mathcal{G}^1_{\mathrm{CIB}}}|\mathcal{G}^1,\mathcal{G}^2)}{p(z_{\mathcal{G}^1_{\mathrm{CIB}}})}\right]$$

$$= \mathbb{E}_{\mathcal{G}^1,\mathcal{G}^2}\left[-\log\frac{p_\phi\,(z_{\mathcal{G}^1_{\mathrm{CIB}}}|\mathcal{G}^1,\mathcal{G}^2)}{q(z_{\mathcal{G}^1_{\mathrm{CIB}}})}\right] - \mathbb{E}_{z_{\mathcal{G}^1_{\mathrm{CIB}}},\mathcal{G}^1,\mathcal{G}^2}\left[KL(p\left(z_{\mathcal{G}^1_{\mathrm{CIB}}}\right)||q\left(z_{\mathcal{G}^1_{\mathrm{CIB}}}\right))\right]$$

$$\le \mathbb{E}_{z_{\mathcal{G}^1_{\mathrm{CIB}}},\mathcal{G}^1,\mathcal{G}^2}\left[KL(p_\phi\,(z_{\mathcal{G}^1_{\mathrm{CIB}}}|\mathcal{G}^1,\mathcal{G}^2)||q\left(z_{\mathcal{G}^1_{\mathrm{CIB}}}\right))\right] \quad (1) \qquad \because \text{Non-negativity of KL divergence}$$

Assuming that $q\left(z_{\mathcal{G}^1_{\mathrm{CIB}}}\right)$ is Gaussian distribution.
The noise $\varepsilon \sim N(\mu_{\mathbf{H}^1},\sigma_{\mathbf{H}^1})$ is sampled from Gaussian distribution where $\mu_{\mathbf{H}^1}$ and $\sigma_{\mathbf{H}^1}$ are mean and variance of $\mathbf{H}^1$.

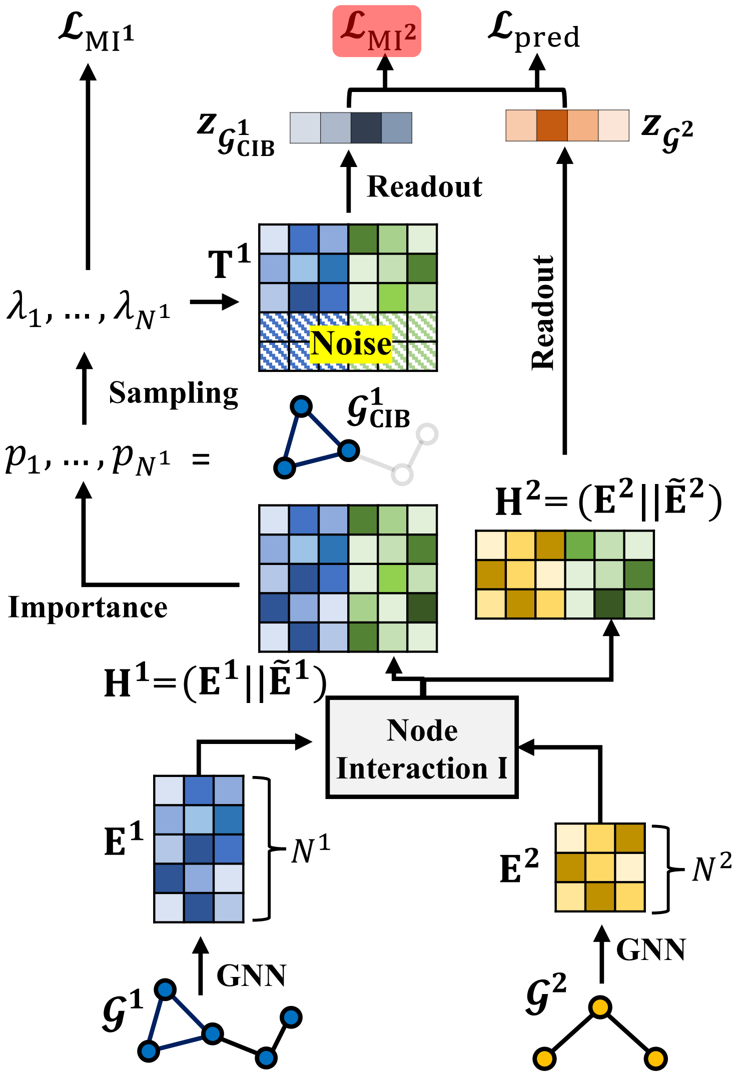
Thus, $q\left(z_{\mathcal{G}^1_{\mathrm{CIB}}}\right) = N(N^1\mu_{\mathbf{H}^1}, N^1\sigma_{\mathbf{H}^1})$ (2) ∵ Summation of Gaussian is Gaussian
And, $p\left(z_{\mathcal{G}^1_{\mathrm{CIB}}}|\mathcal{G}^1,\mathcal{G}^2\right) = N(N^1\mu_{\mathbf{H}^1} + \sum_{j=1}^{N^1}\lambda_j\mathbf{H}^1_j - \sum_{j=1}^{N^1}\lambda_j\mu_{\mathbf{H}^1}, \sum_{j=1}^{N^1}\left(1-\lambda_j\right)^2\sigma^2_{\mathbf{H}^1})$ (3)

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

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

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

Proposed Method: Conditional Graph Information Bottleneck



$\mathcal{L}_{\mathrm{MI}^1}$
$\mathcal{L}_{\mathrm{MI}^2}$
$\mathcal{L}_{\mathrm{pred}}$
$\mathbf{z}_{\mathcal{G}^1_{\mathrm{CIB}}}$
$\mathbf{z}_{\mathcal{G}^2}$
Readout
Readout
$\mathbf{T}^1$
$\lambda_1,\dots,\lambda_{N^1}$
Noise
Sampling
$\mathcal{G}^1_{\mathrm{CIB}}$
$p_1,\dots,p_{N^1}$ =
$\mathbf{H}^2 = (\mathbf{E}^2||\tilde{\mathbf{E}}^2)$
Importance
$\mathbf{H}^1 = (\mathbf{E}^1||\tilde{\mathbf{E}}^1)$
Node Interaction I
$\mathbf{E}^1$
N^1
$\mathbf{E}^2$
N^2
GNN
GNN
$\mathcal{G}^1$
$\mathcal{G}^2$

- Step 2: Optimizing the compression loss

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

2. Solute Prediction

Encourage $\mathcal{G}^1_{\mathrm{CIB}}$, 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 $\mathcal{G}^1_{\mathrm{CIB}}$

1) Variational IB-based approach

Derive upper bound similar to the prediction loss

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

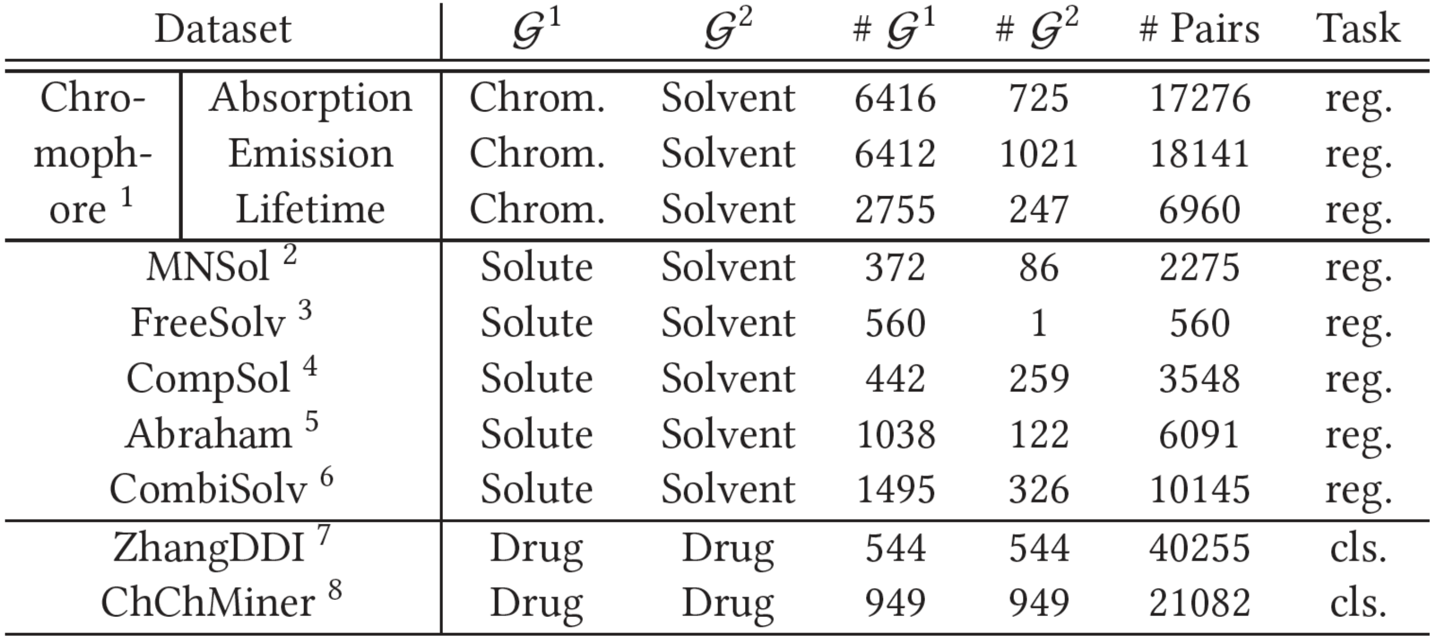
2) Contrastive Learning-based approach

- Minimizing the contrastive loss is proven to be equivalent to maximizing the mutual information
- Hence, minimize $-I(G^1_{\mathrm{CIB}};G^2)$ by minimizing the contrastive loss → $\mathrm{CGIB_{cont}}$

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

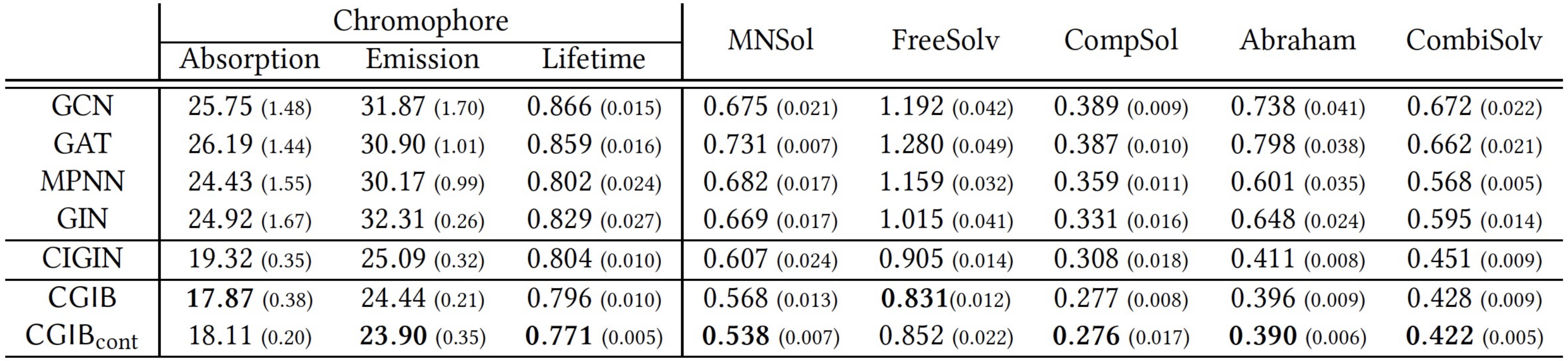
Experiments: Dataset

- 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 [1]	Absorption	Chrom.	Solvent	6416	725	17276	reg.
	Emission	Chrom.	Solvent	6412	1021	18141	reg.
	Lifetime	Chrom.	Solvent	2755	247	6960	reg.
MNSol [2]		Solute	Solvent	372	86	2275	reg.
FreeSolv [3]		Solute	Solvent	560	1	560	reg.
CompSol [4]		Solute	Solvent	442	259	3548	reg.
Abraham [5]		Solute	Solvent	1038	122	6091	reg.
CombiSolv [6]		Solute	Solvent	1495	326	10145	reg.
ZhangDDI [7]		Drug	Drug	544	544	40255	cls.
ChChMiner [8]		Drug	Drug	949	949	21082	cls.

Result: Main table



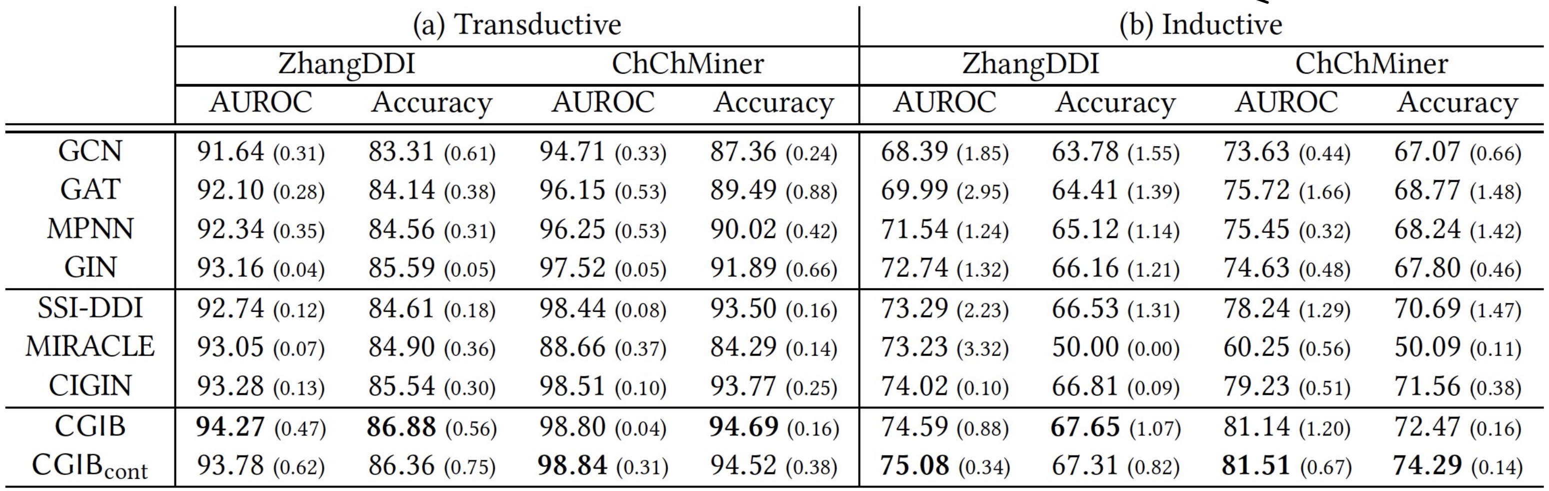
	Chromophore			MNSol	FreeSolv	CompSol	Abraham	CombiSolv
	Absorption	Emission	Lifetime					
GCN	25.75 (1.48)	31.87 (1.70)	0.866 (0.015)	0.675 (0.021)	1.192 (0.042)	0.389 (0.009)	0.738 (0.041)	0.672 (0.022)
GAT	26.19 (1.44)	30.90 (1.01)	0.859 (0.016)	0.731 (0.007)	1.280 (0.049)	0.387 (0.010)	0.798 (0.038)	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	**17.87** (0.38)	24.44 (0.21)	0.796 (0.010)	0.568 (0.013)	**0.831**(0.012)	0.277 (0.008)	0.396 (0.009)	0.428 (0.009)
CGIB$_{cont}$	18.11 (0.20)	**23.90** (0.35)	**0.771** (0.005)	**0.538** (0.007)	0.852 (0.022)	**0.276** (0.017)	**0.390** (0.006)	**0.422** (0.005)

Performance on Molecular Interaction (Regression)

Observations

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

Evaluation on drugs unseen during training



	(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	**94.27** (0.47)	**86.88** (0.56)	98.80 (0.04)	**94.69** (0.16)	74.59 (0.88)	**67.65** (1.07)	81.14 (1.20)	72.47 (0.16)
CGIB$_{cont}$	93.78 (0.62)	86.36 (0.75)	**98.84** (0.31)	94.52 (0.38)	**75.08** (0.34)	67.31 (0.82)	**81.51** (0.67)	**74.29** (0.14)

Performance on Drug-Drug Interaction (Classification)

Observations

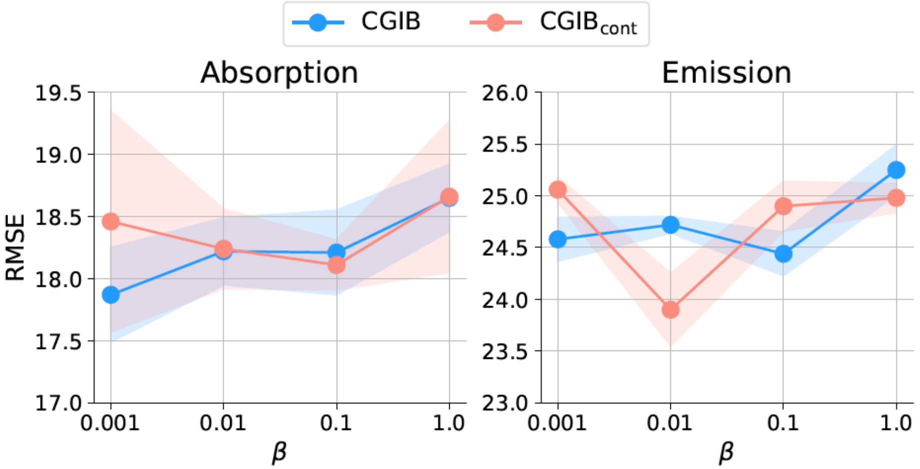
- Improvement gap is larger in inductive setting
  - ∵ By detecting function group that is basic in nature → helps generalization

Result: Analysis on β

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

- β Controls Trade-off btw prediction and compression

As β increases, Compression > Prediction

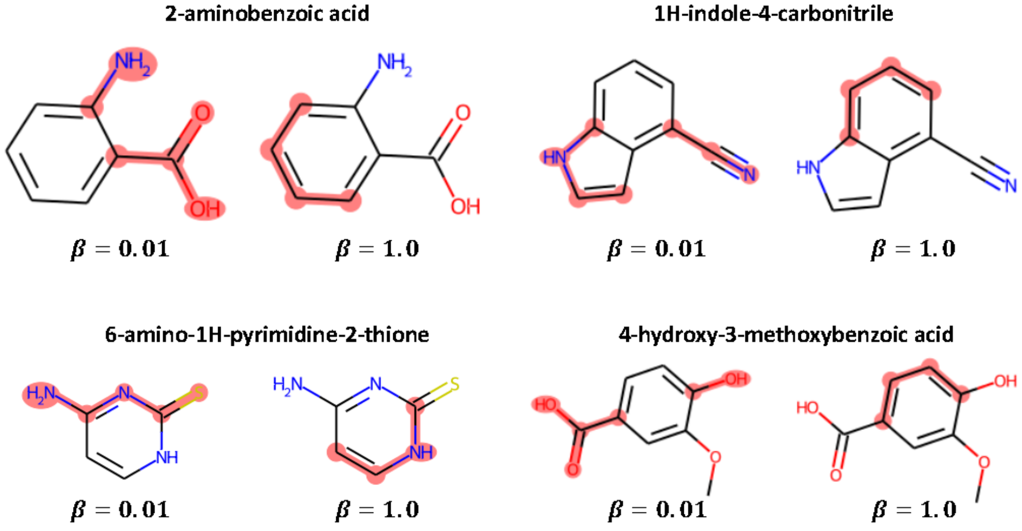


Sensitivity Analysis on β

Observations - Sensitivity Analysis

- $\beta = 1.0$: Poor performance in general (focus on compression)
- However, the model fails to detect functional group when β is too small → poor generalization

→ Hence, finding an appropriate β is crucial



Qualitative Analysis on β

Observations - Qualitative analysis

- $\beta = 1.0$ → CGIB focuses on compression

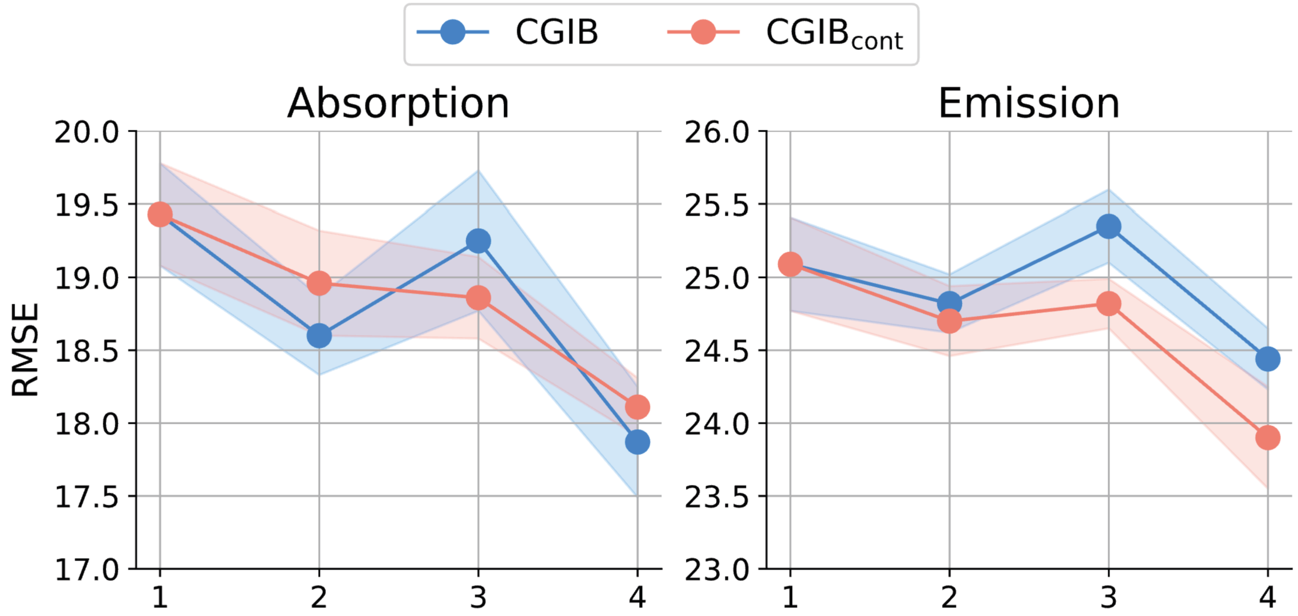
e.g., CGIB focuses an aromatic ring, which is not relevant to chemical reactions

- $\beta = 0.01$ → CGIB focuses on prediction

e.g., CGIB focuses on external part, which generally more relevant to chemical reactions

Result: Ablation studies

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



Observations - Ablation Studies

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

1. Without IB → $\min\ -\ I(Y; \mathcal{G}^1, \mathcal{G}^2)$ **(Same as CIGIN)**

2. $I(\mathcal{G}^1_{\mathrm{CIB}}; \mathcal{G}^1)$ → $\min\ -\ I(Y; \mathcal{G}^1_{\mathrm{CIB}}, \mathcal{G}^2) + I(\mathcal{G}^1_{\mathrm{CIB}}; \mathcal{G}^1)$ **(Same as VGIB)**

3. $I(\mathcal{G}^1_{\mathrm{CIB}}; \mathcal{G}^1, \mathcal{G}^2)$ → $\min\ -\ I(Y; \mathcal{G}^1_{\mathrm{CIB}}, \mathcal{G}^2) + I(\mathcal{G}^1_{\mathrm{CIB}}; \mathcal{G}^1, \mathcal{G}^2)$

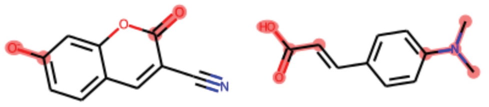
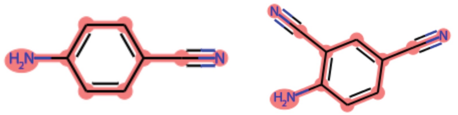
4. $I(\mathcal{G}^1; \mathcal{G}^1_{\mathrm{CIB}} | \mathcal{G}^2)$ → $\min\ -\ I(Y; \mathcal{G}^1_{\mathrm{CIB}}, \mathcal{G}^2) + I(\mathcal{G}^1; \mathcal{G}^1_{\mathrm{CIB}} | \mathcal{G}^2)$ **(Same as CGIB)**

Importance of IB

Importance of conditional IB

Importance of valid conditional IB

Result: Qualitative analysis

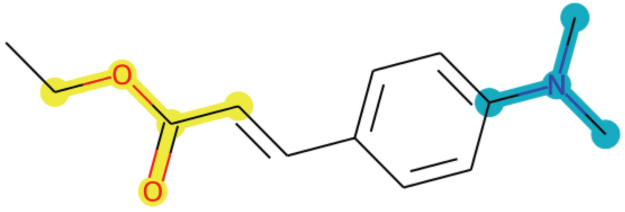
Solvent ($\mathcal{G}^2$)	(a) Ordinary solvents	(b) Liquid oxygen solvent
Chromophore ($\mathcal{G}^1$)		

(a) Chromophore ($\boldsymbol{G}^1$) interact with ordinary solvents ($\boldsymbol{G}^2$)

Focus on external parts → Aligns with domain knowledge

(b) Chromophore ($\boldsymbol{G}^1$) interact with liquid oxygen solvents ($\boldsymbol{G}^2$)

Focus on all parts → Aligns with domain knowledge

Solvent ($\mathcal{G}^2$)	(c) Ethanol, THF 1-hexanol, 1-butanol benzene
Chromophore ($\mathcal{G}^1$) EDAC	 Oxygen-Carbon Nitrogen-Carbon

(c) Chromophore ($\boldsymbol{G}^1$) interacts with various solvents ($\boldsymbol{G}^2$) (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)

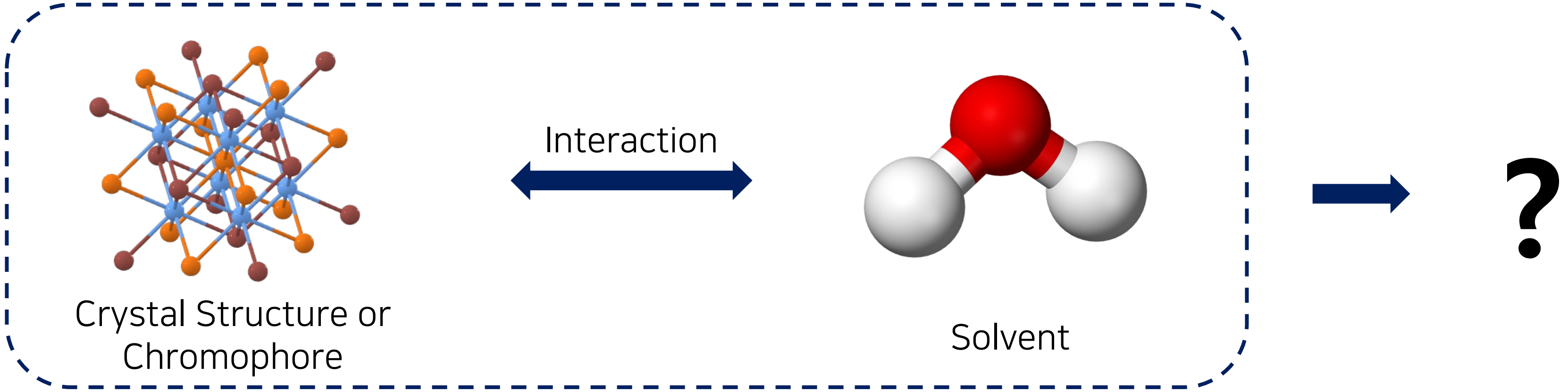
Structure with low polarity is detected (e.g., Nitrogen-Carbon)

→ 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 $\mathcal{G}^1_{\text{CIB}}$ of $\mathcal{G}^1$
- CGIB has interpretability, which makes it highly practical



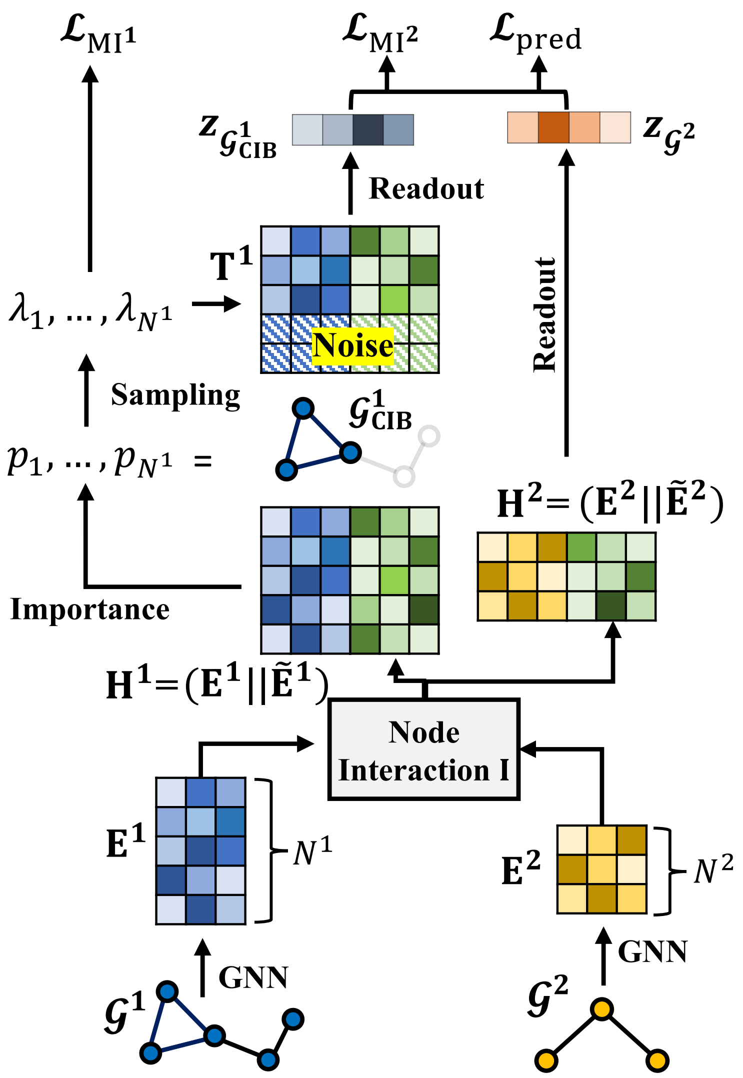

$\mathcal{L}_{\text{MI}^1}$
$\mathcal{L}_{\text{MI}^2}$
$\mathcal{L}_{\text{pred}}$
$z_{\mathcal{G}^1_{\text{CIB}}}$
$z_{\mathcal{G}^2}$
Readout
$\mathbf{T}^1$
Noise
$\lambda_1, \ldots, \lambda_{N^1}$
Readout
Sampling
$\mathcal{G}^1_{\text{CIB}}$
$p_1, \ldots, p_{N^1}$ =
$\mathbf{H}^2 = (\mathbf{E}^2 || \tilde{\mathbf{E}}^2)$
Importance
$\mathbf{H}^1 = (\mathbf{E}^1 || \tilde{\mathbf{E}}^1)$
Node Interaction I
$\mathbf{E}^1$
N^1
$\mathbf{E}^2$
N^2
GNN
GNN
$\mathcal{G}^1$
$\mathcal{G}^2$

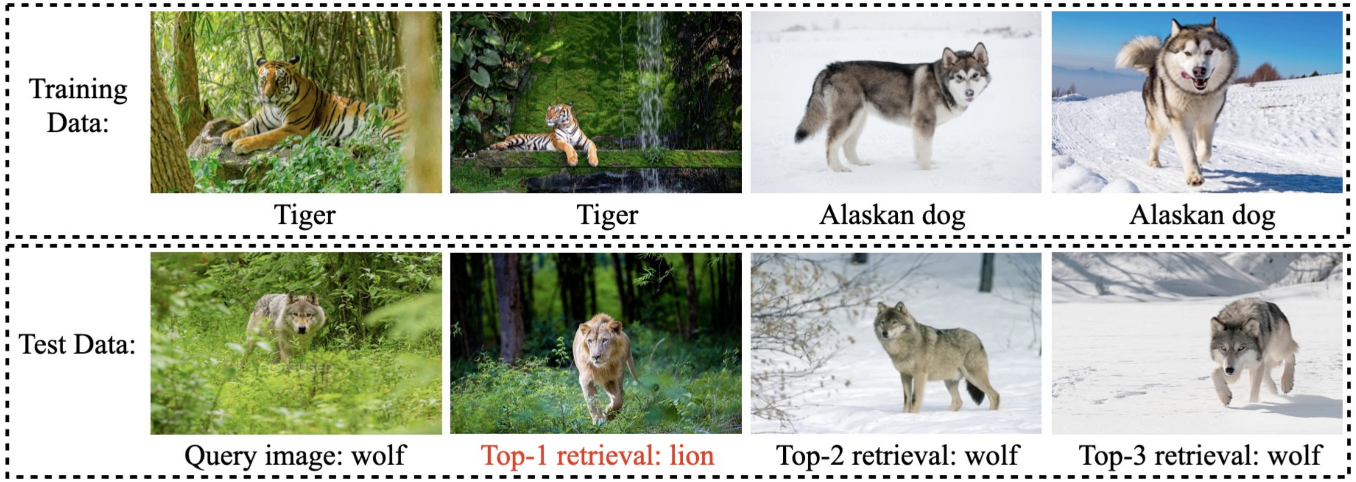
Table of Contents

시간	주제	주요 내용
1	그래프 신경망 개요	- 그래프 신경망 전반적인 소개 - 그래프 종류에 따른 다양한 그래프 신경망 소개
2	소재 물성 예측 연구	- 소재 물성 예측 연구 최신 동향 소개 - Transformer 기반 모델 소개 및 extrapolation을 위한 모델 소개
3	물질 간 화학 반응 예측 연구	- 물질 간 화학 반응 예측 연구 동향 소개 - 정보 이론(Information bottleneck) 및 인과추론(Causal inference) 기반 모델 소개

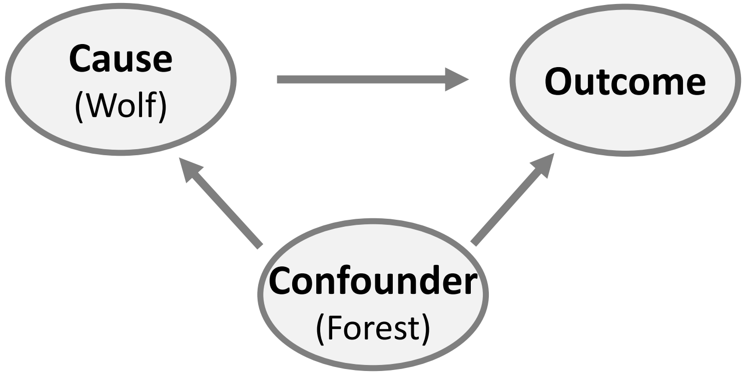
Papers

- 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
  - SSI-DDI: substructure-substructure interactions for drug-drug interaction prediction. Briefings in Bioinformatics 2021
  - 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



ex) Spurious correlation between "Forest" and "Lion"

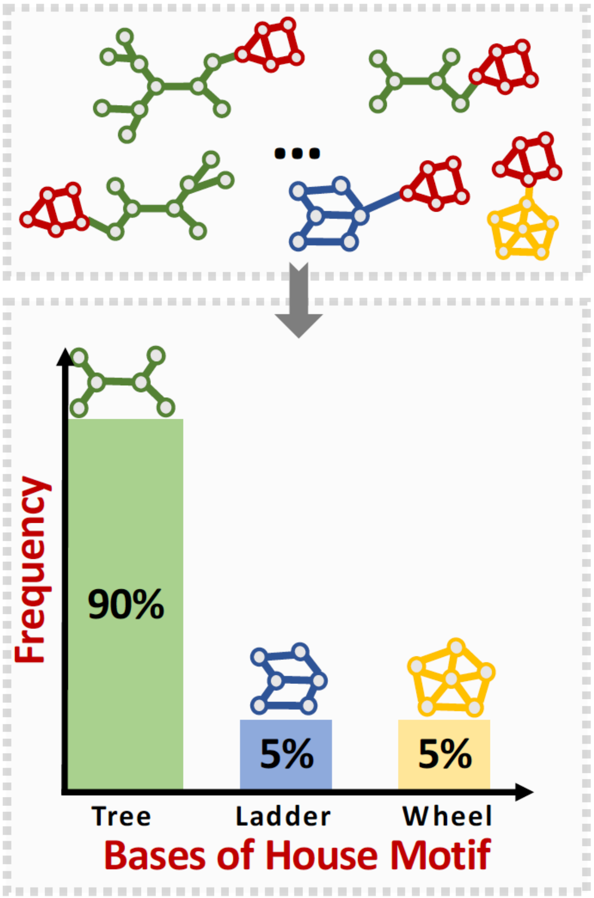


- Due to the empirical process of data collection, the data for machine learning is heavily biased

- Context of the given data becomes a confounder that misleads the machine learning model to learn spurious correlations between pixels and labels

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

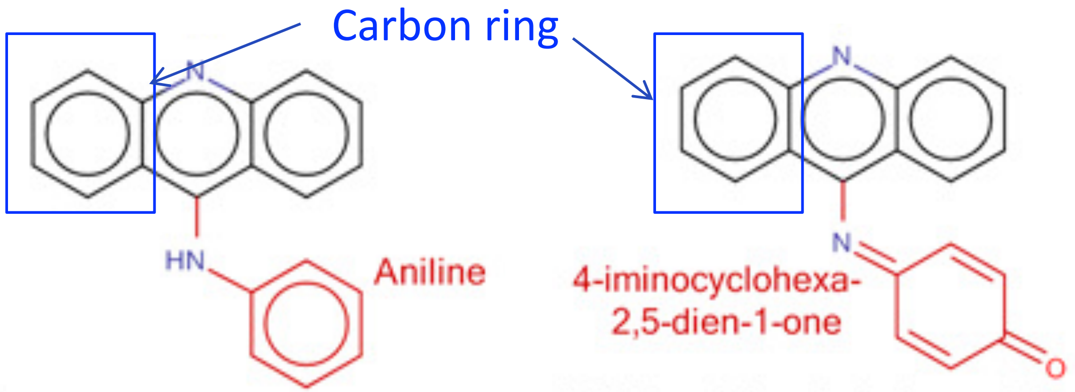
- Observation: Statistical Shortcuts link the Tree motifs with House motifs

→ When facing with out-of-distribution (OOD) data, statistical shortcuts will severely deteriorate the model performance (since the shortcuts will change)

Discovering Invariant Rationales for Graph Neural Networks. ICLR 2022

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



(3) N-phenylacridin-9-amine
0.94 (17/18)

(24) 4-acridine-9-yliminocyclohexa-2,5-dien-1-one
0 (0/1)

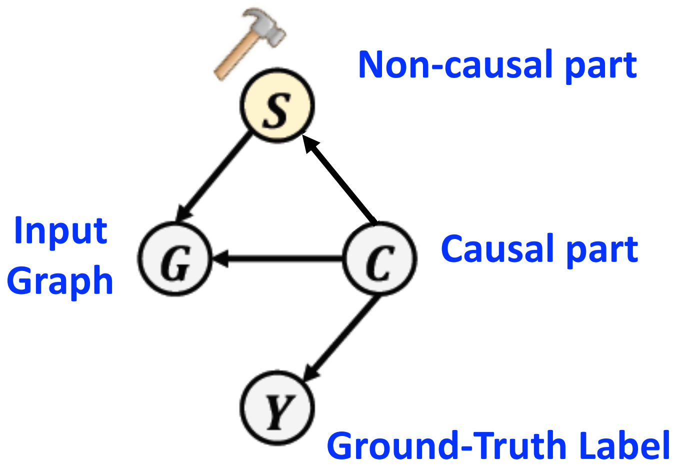
Mutagenic Scaffold

Non-Mutagenic Scaffold

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

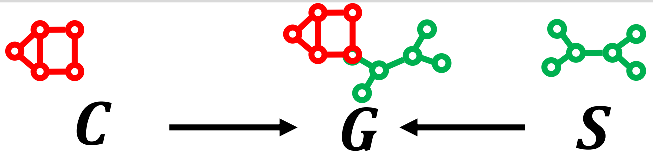
Discovering Invariant Rationales for Graph Neural Networks (1/4)

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

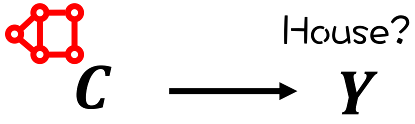


Structure Causal Model (SCM)

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



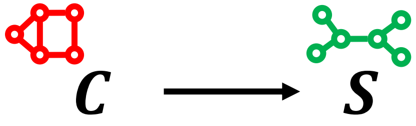
Causal part $\boldsymbol{C}$ only determines target value $\boldsymbol{Y}$



Dependency between $\boldsymbol{C}$ and $\boldsymbol{S}$

➔Create spurious correlation between $\boldsymbol{S}$ and $\boldsymbol{Y}$

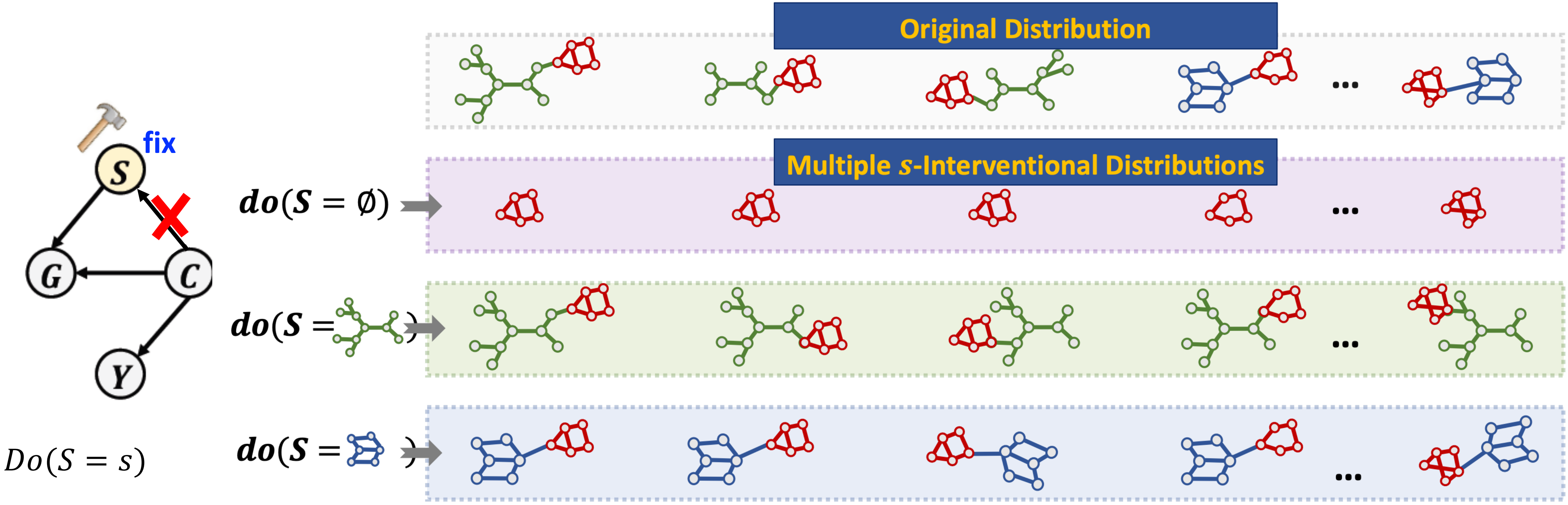
(S←C→Y)



Discovering Invariant Rationales for Graph Neural Networks. ICLR 2022

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

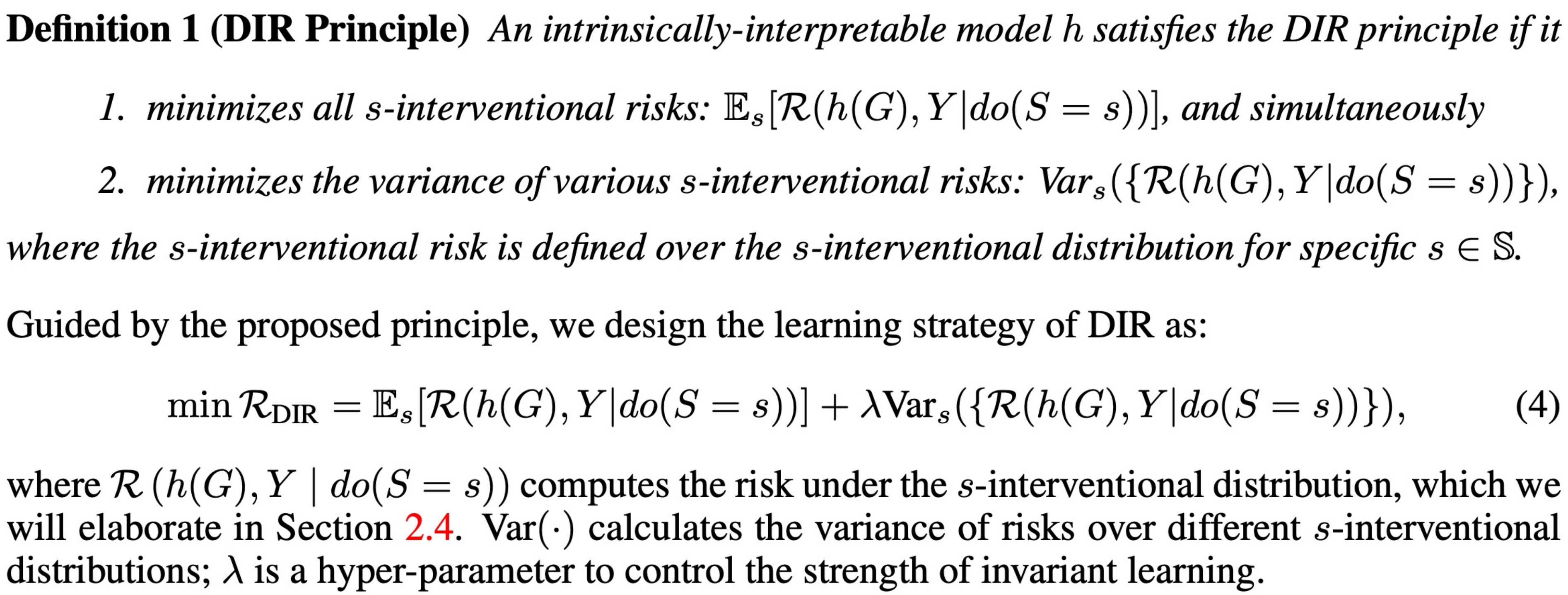
- Research question: How to get multiple environments from a standard training set?
  → Causal intervention


fix
S
G
C
Y
$Do(S = s)$
Original Distribution
Multiple s-Interventional Distributions
$\boldsymbol{do(S = \emptyset)}$
$\boldsymbol{do(S =}$ $\boldsymbol{)}$
$\boldsymbol{do(S =}$ $\boldsymbol{)}$
...

Generate s-interventional distribution by doing intervention on $\boldsymbol{S}$

Discovering Invariant Rationales for Graph Neural Networks. ICLR 2022

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: $Var_s(\{\mathcal{R}(h(G), Y|do(S=s))\})$,*

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:

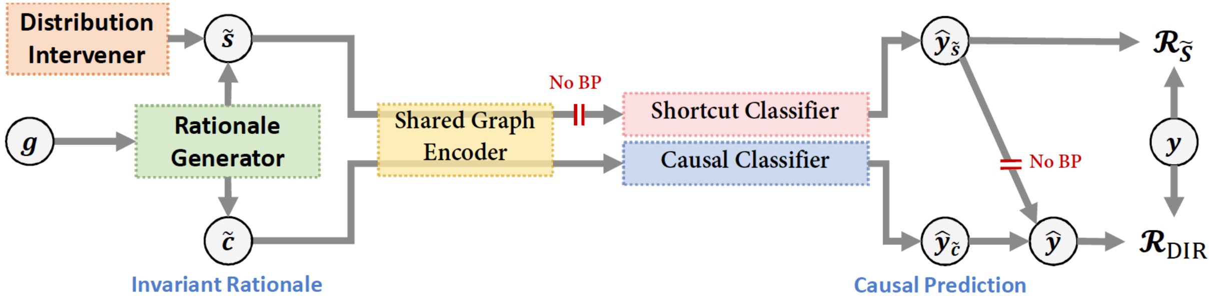
$$\min \mathcal{R}_{\text{DIR}} = \mathbb{E}_s[\mathcal{R}(h(G), Y|do(S=s))] + \lambda \text{Var}_s(\{\mathcal{R}(h(G), Y|do(S=s))\}), \tag{4}$$

where $\mathcal{R}(h(G), Y \mid do(S=s))$ 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; λ 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. ICLR 2022

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 $\tilde{s}$

$$\mathbf{Z} = \text{GNN}_1(g), \quad \mathbf{M}_{ij} = \sigma(\mathbf{Z}_i^\top \mathbf{Z}_j),$$ 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 $\tilde{\mathbb{S}}$
- Samples memory $\tilde{s}_j \in \tilde{\mathbb{S}}$ 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

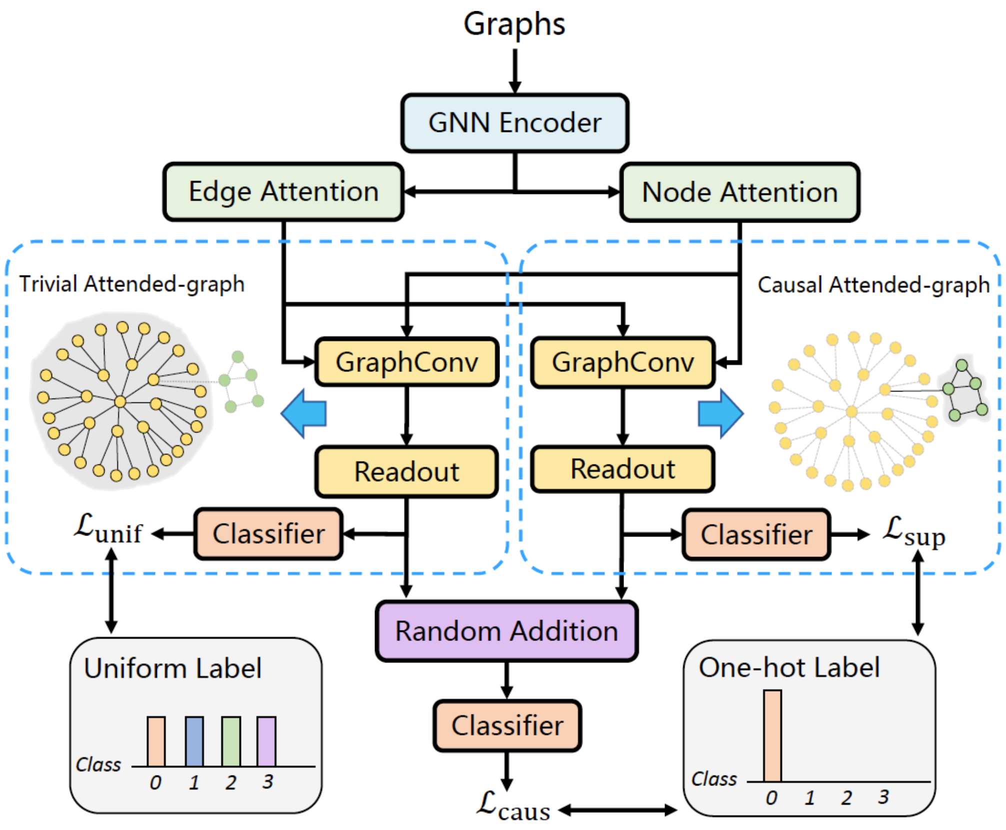
$$\mathcal{R}(h(G), Y | do(S = \tilde{s})) = \mathbb{E}_{(g,y) \in \mathcal{O}, S = \tilde{s}, C = h_{\tilde{C}}(g)} l(\hat{y}, y)$$

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

Discovering Invariant Rationales for Graph Neural Networks. ICLR 2022

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

<u>Task: Graph Classification</u> → "How to classify biased graph datasets?"



Model Architecture

<u>Soft Mask Estimation</u>

Separate the causal and shortcut features from the full graphs

<u>Disentanglement</u>

Separate the causal and shortcut features from the full graphs

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

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

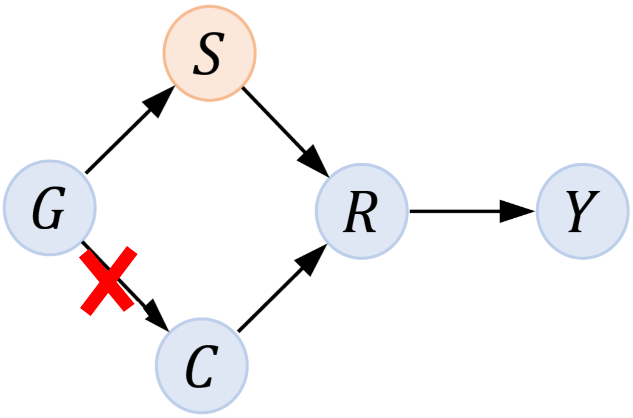
$\mathcal{L}_{\text{sup}} = -\frac{1}{|\mathcal{D}|} \sum_{\mathcal{G} \in \mathcal{D}} \mathbf{y}_{\mathcal{G}}^{\top} \log(\mathbf{z}_{\mathcal{G}_c})$ Causal graph → Ground truth label prediction

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

Causal Attention for Interpretable and Generalizable Graph Classification. KDD 2022

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)

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

Confounder Set

Backdoor Adjustment

Causal Intervention via Backdoor adjustment

Challenges

1) Confounder set $\mathcal{T}$ is commonly unobservable and hard to obtain
2) Difficult to directly manipulate graph data (∵Discrete nature)

→ Let's make implicit intervention on representation level!

$$\mathbf{z}_{\mathcal{G}'} = \Phi(\mathbf{h}_{\mathcal{G}_c} + \mathbf{h}_{\mathcal{G}_{t'}})$$

Trivial graph from different graphs

$$\mathcal{L}_{\text{caus}} = -\frac{1}{|\mathcal{D}| \cdot |\hat{\mathcal{T}}|} \sum_{\mathcal{G}\in\mathcal{D}} \sum_{t'\in\hat{\mathcal{T}}} \mathbf{y}_{\mathcal{G}}^{\top} \log\left(\mathbf{z}_{\mathcal{G}'}\right)$$

Causal Attention for Interpretable and Generalizable Graph Classification. KDD 2022

Shift-Robust Molecular Relational Learning with Causal Substructure

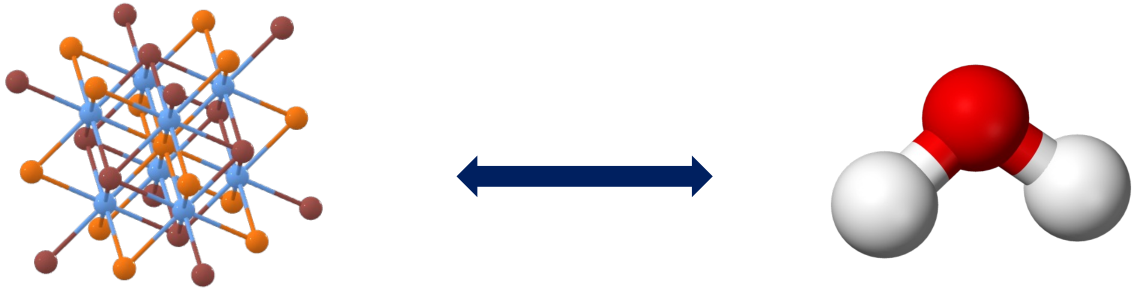
Namkyeong Lee, Kanghoon Yoon, Gyoung S. Na, Sein Kim, Chanyoung Park

KDD 2023 - International Conference on Knowledge Discovery and Data Mining

Recall: Relational Learning

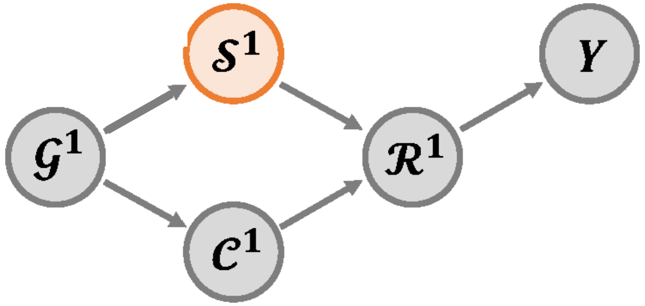
▪ Molecular Relational Learning

- 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

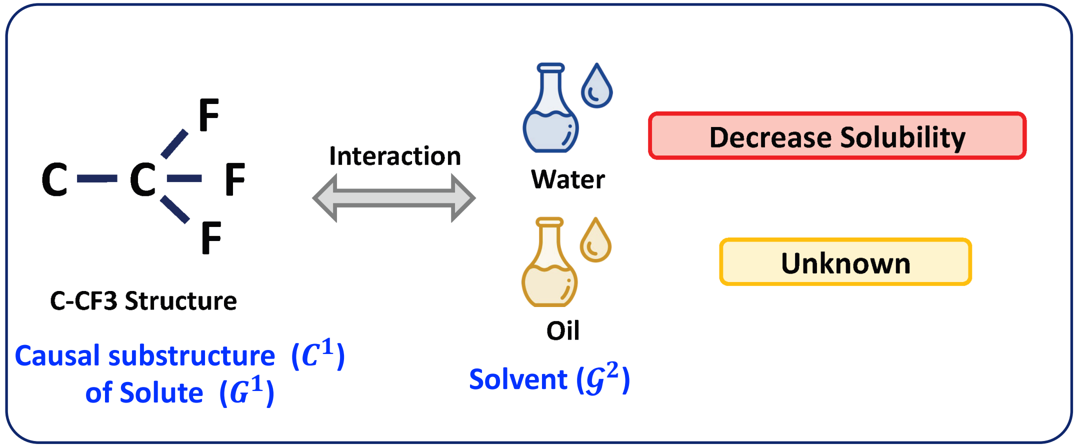
Structure Causal Model (SCM) for
Molecular Relational Learning

Why not G^2 and C^2?

Key causal-effect relationship
in molecular relational learning

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

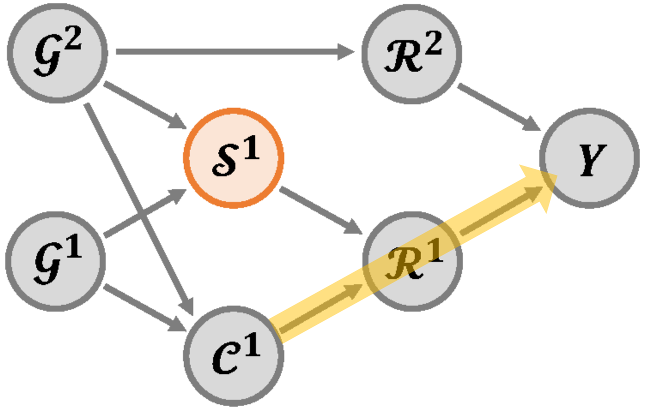
Causal substructure $\mathcal{C}^1$ of molecule $\mathcal{G}^1$
→ Determined by not only $\mathcal{G}^1$ but also $\mathcal{G}^2$



Causal substructure (C^1)
of Solute (G^1)

Solvent ($\mathcal{G}^2$)

Methodology Causality in molecular relational learning



$\mathcal{G}^1$: Molecule 1
$\mathcal{G}^2$: Molecule 2
$\mathcal{C}^1$: Causal Substructure in Molecule 1
$\mathcal{S}^1$: Shortcut Substructure in Molecule 1
$\mathcal{R}^1$: Molecule 1 Representation
$\mathcal{R}^2$: Molecule 2 Representation
Y : Target Value

Structure Causal Model (SCM) for
Molecular Relational Learning

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

- 4 Backdoor paths that confound the model

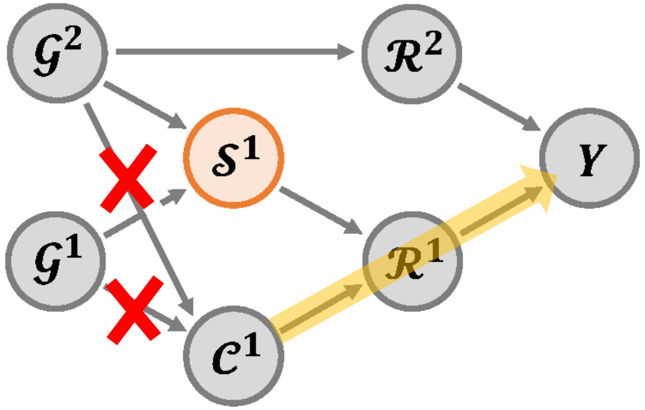
1. $\mathcal{C}^1 \leftarrow \mathcal{G}^1 \rightarrow \mathcal{S}^1 \leftarrow \mathcal{G}^2 \rightarrow \mathcal{R}^2 \rightarrow Y$
2. $\mathcal{C}^1 \leftarrow \mathcal{G}^2 \rightarrow \mathcal{R}^2 \rightarrow Y$
3. $\mathcal{C}^1 \leftarrow \mathcal{G}^2 \rightarrow \mathcal{S}^1 \rightarrow \mathcal{R}^1 \rightarrow Y$
4. $\mathcal{C}^1 \leftarrow \mathcal{G}^1 \rightarrow \mathcal{S}^1 \rightarrow \mathcal{R}^1 \rightarrow Y$ ⬅

- In molecular relational learning,

→ $\mathcal{G}^2$ is given and utilized during model prediction,
all paths are blocked except for

$\mathcal{C}^1 \leftarrow \mathcal{G}^1 \rightarrow \mathcal{S}^1 \rightarrow \mathcal{R}^1 \rightarrow Y$ Only remaining backdoor path!

Methodology Backdoor adjustment



$\mathcal{G}^1$: Molecule 1
$\mathcal{G}^2$: Molecule 2
$\mathcal{C}^1$: Causal Substructure in Molecule 1
$\mathcal{S}^1$: Shortcut Substructure in Molecule 1
$\mathcal{R}^1$: Molecule 1 Representation
$\mathcal{R}^2$: Molecule 2 Representation
Y : Target Value

Structure Causal Model (SCM) for Molecular Relational Learning

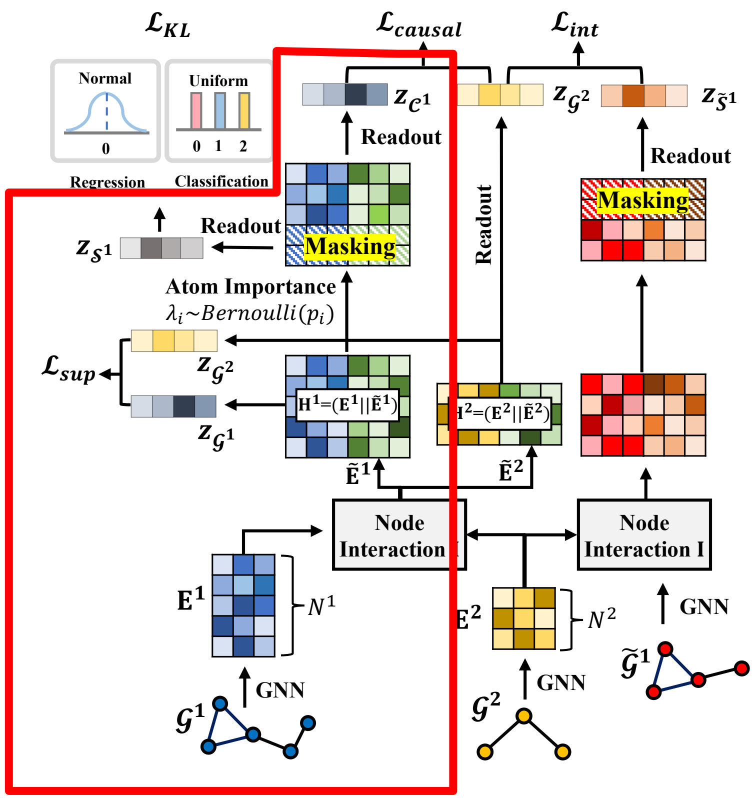
$$\begin{aligned}
P(\mathrm{Y}|do(C^1),\mathcal{G}^2) &= \tilde{P}(\mathrm{Y}|C^1,\mathcal{G}^2) \\
&= \sum_s \tilde{P}(\mathrm{Y}|C^1,\mathcal{G}^2,s)\cdot\tilde{P}(s|C^1,\mathcal{G}^2) \text{ (Bayes' Rule)} \\
&= \sum_s \tilde{P}(\mathrm{Y}|C^1,\mathcal{G}^2,s)\cdot\tilde{P}(s|\mathcal{G}^2) \text{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

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!

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

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

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

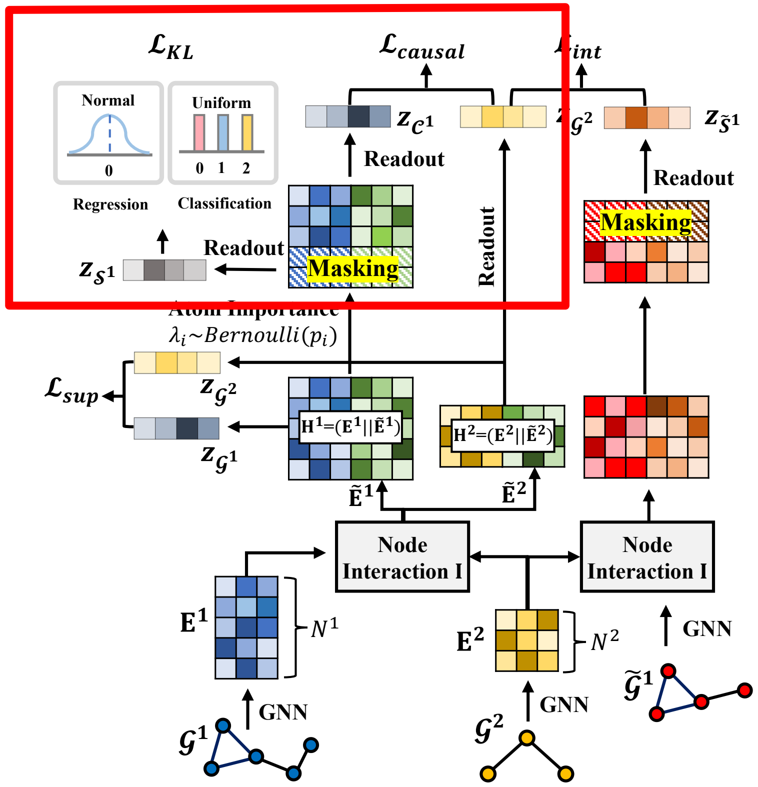
where $\lambda_i \sim \mathrm{Bernoulli}(p_i)$ $\epsilon \sim N(\mu_{\mathrm{H}^1}, \sigma_{\mathrm{H}^1}^2)$

- Gumbel sigmoid approach for differentiable optimization of p_i

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

Shift-Robust Molecular Relational Learning with Causal Substructure. KDD 2023

Methodology Causal molecular relational learner



Disentangling with Atom Representation Masks

- Causal substructure $\mathcal{C}^1$

→ Cross entropy loss for classification

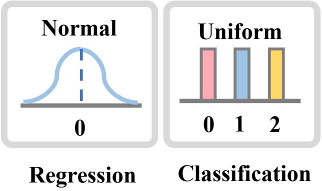
→ RMSE loss for Regression

$$\mathcal{L}_{causal}(\mathrm{Y}, z_{\mathcal{C}^1}, z_{\mathcal{G}^2})$$

- Shortcut substructure $\mathcal{S}^1$

→ Learn non informative distribution

$$\mathcal{L}_{KL}(\mathrm{Y}_{rand}, z_{\mathcal{S}^1})$$

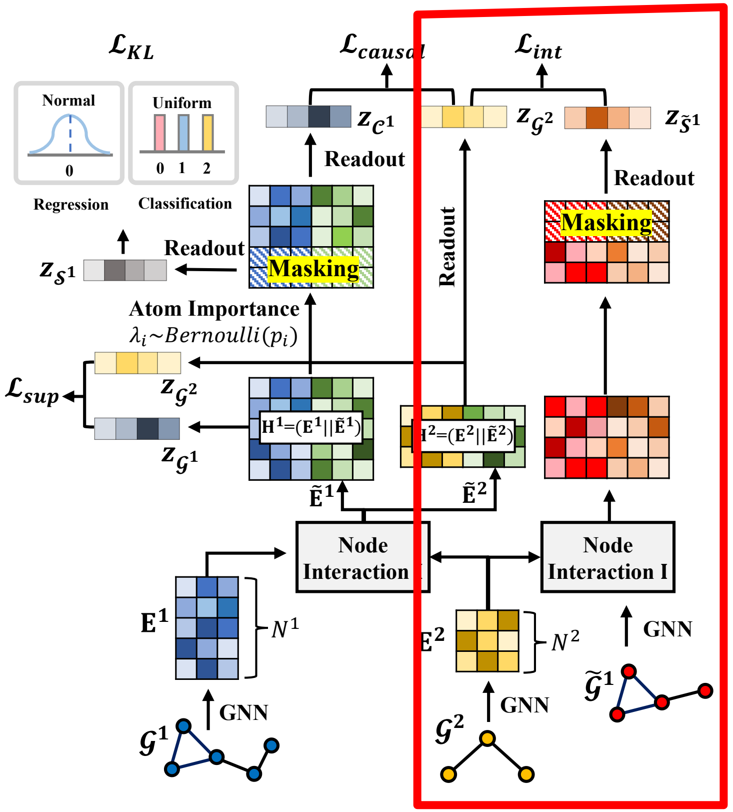


Shift-Robust Molecular Relational Learning with Causal Substructure. KDD 2023

Methodology Causal molecular relational learner

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

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 $\boldsymbol{\mathcal{C}^1}$ should be conditioned on the paired molecule $\boldsymbol{\mathcal{G}^2}$

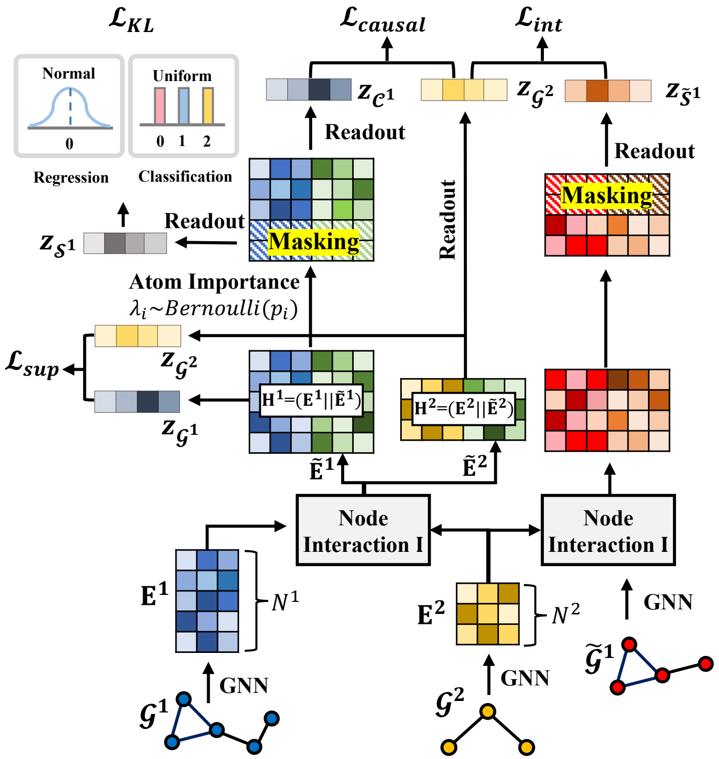
Our Solution

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

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

Shift-Robust Molecular Relational Learning with Causal Substructure. KDD 2023

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 $(\boldsymbol{\mathcal{G}}^1, \boldsymbol{\mathcal{G}}^2)$ and target $\boldsymbol{Y}$
- $\mathcal{L}_{causal}$: loss with causal substructure
- $\mathcal{L}_{KL}$: loss with shortcut substructure
- λ_1, λ_2: weight hyperparameters for $\mathcal{L}_{KL}$ and $\mathcal{L}_{int}$

Theoretical analysis

Training objective of CMRL $-\ell = -\sum_{i=1}^{n} \log q(\mathrm{Y}_i | C_i^1, \mathcal{G}_i^2)$

Expand by multiplying and dividing q

$$-\ell = \sum_{i=1}^{n} \log \frac{p(\mathrm{Y}_i|C_i^1, \mathcal{G}_i^2)}{q(\mathrm{Y}_i|C_i^1, \mathcal{G}_i^2)} + \sum_{i=1}^{n} \log \frac{p(\mathrm{Y}_i|\mathcal{G}_i^1, \mathcal{G}_i^2)}{p(\mathrm{Y}_i|C_i^1, \mathcal{G}_i^2)} - \sum_{i=1}^{n} \log p(\mathrm{Y}_i|\mathcal{G}_i^1, \mathcal{G}_i^2)$$

$$= \mathbb{E}\left[\log \frac{p(\mathrm{Y}|C^1, \mathcal{G}^2)}{q(\mathrm{Y}|C^1, \mathcal{G}^2)}\right] + \mathbb{E}\left[\log \frac{p(\mathrm{Y}|\mathcal{G}^1, \mathcal{G}^2)}{p(\mathrm{Y}|C^1, \mathcal{G}^2)}\right] - \mathbb{E}\left[\log p(\mathrm{Y}|\mathcal{G}^1, \mathcal{G}^2)\right],$$

$$\mathbb{E}\left[\log \frac{p(\mathrm{Y}|\mathcal{G}_i^1, \mathcal{G}_i^2)}{p(\mathrm{Y}|C_i^1, \mathcal{G}_i^2)}\right] = \mathbb{E}\left[\log \frac{p(\mathrm{Y}|C_i^1, \mathcal{S}_i^1, \mathcal{G}_i^2)}{p(\mathrm{Y}|C_i^1, \mathcal{G}_i^2)}\right]$$

$$= \sum_{i=1}^{n} p(\mathcal{G}_i^1, \mathcal{G}_i^2, \mathrm{Y}_i) \log \frac{p(\mathrm{Y}_i|C_i^1, \mathcal{S}_i^1, \mathcal{G}_i^2)}{p(\mathrm{Y}_i|C_i^1, \mathcal{G}_i^2)}$$

$$= \sum_{i=1}^{n} p(\mathcal{G}_i^1, \mathcal{G}_i^2, \mathrm{Y}_i) \log \frac{p(\mathrm{Y}_i|C_i^1, \mathcal{S}_i^1, \mathcal{G}_i^2)}{p(\mathrm{Y}_i|C_i^1, \mathcal{G}_i^2)} \frac{p(\mathcal{S}_i^1|C_i^1, \mathcal{G}_i^2)}{p(\mathcal{S}_i^1|C_i^1, \mathcal{G}_i^2)}$$

$$= \sum_{i=1}^{n} p(\mathcal{G}_i^1, \mathcal{G}_i^2, \mathrm{Y}_i) \log \frac{p(\mathcal{S}_i^1, \mathrm{Y}_i|C_i^1, \mathcal{G}_i^2)}{p(\mathrm{Y}_i|C_i^1, \mathcal{G}_i^2) \cdot p(\mathcal{S}_i^1|C_i^1, \mathcal{G}_i^2)}$$

$$= I(\mathcal{S}^1; \mathrm{Y}|C^1, \mathcal{G}^2)$$

$$\min \mathbb{E}\left[\log \frac{p(\mathrm{Y}|C^1, \mathcal{G}^2)}{q(\mathrm{Y}|C^1, \mathcal{G}^2)}\right] + I(\mathcal{S}^1; \mathrm{Y}|C^1, \mathcal{G}^2) + H(\mathrm{Y}|\mathcal{G}^1, \mathcal{G}^2)$$

1. Likelihood ratio between true distribution and predicted distribution
2. Conditional Mutual Information
3. Irreducible constant inherent in the datasets

We can explain the behavior of CMRL in two perspective

Shift-Robust Molecular Relational Learning with Causal Substructure. KDD 2023

Theoretical analysis

$$\min \mathbb{E}\left[\log \frac{p(\mathrm{Y}|C^1,\mathcal{G}^2)}{q(\mathrm{Y}|C^1,\mathcal{G}^2)}\right] + I(\mathcal{S}^1;\mathrm{Y}|C^1,\mathcal{G}^2) + H(\mathrm{Y}|\mathcal{G}^1,\mathcal{G}^2)$$

Perspective 1. CMRL learns informative causal substructure

- Minimize $I(\mathcal{S}^1;\mathrm{Y}|C^1,\mathcal{G}^2)$

Disentangle the shortcut substructure $\boldsymbol{S^1}$ that is no longer needed in predicting the label $\boldsymbol{Y}$ when the context $\boldsymbol{\mathcal{C}^1}$ and $\boldsymbol{\mathcal{G}^2}$ given.

- Chain rule of MI $I(\mathcal{S}^1;\mathrm{Y}|C^1,\mathcal{G}^2) = I(\mathcal{G}^1,\mathcal{G}^2;\mathrm{Y}) - I(C^1,\mathcal{G}^2;\mathrm{Y})$

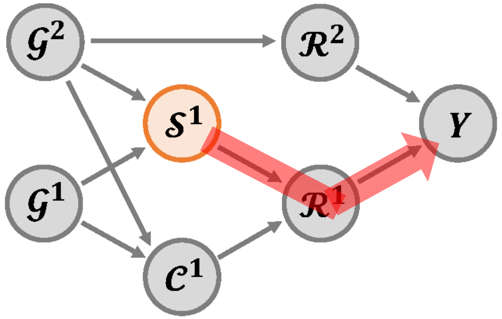
Encourages the causal substructure $\boldsymbol{\mathcal{C}^1}$ and paired molecule $\boldsymbol{\mathcal{G}^2}$ to contain enough information on target $\boldsymbol{Y}$.

Shift-Robust Molecular Relational Learning with Causal Substructure. KDD 2023

Theoretical analysis

$$\min \mathbb{E}\left[\log \frac{p(\mathrm{Y}|C^1,\mathcal{G}^2)}{q(\mathrm{Y}|C^1,\mathcal{G}^2)}\right] + I(\mathcal{S}^1;\mathrm{Y}|C^1,\mathcal{G}^2) + H(\mathrm{Y}|\mathcal{G}^1,\mathcal{G}^2)$$

Perspective 2. CMRL reduces model bias with causal view



$\mathcal{G}^1$: Molecule 1
$\mathcal{G}^2$: Molecule 2
$\mathcal{C}^1$: Causal Substructure in Molecule 1
$\mathcal{S}^1$: Shortcut Substructure in Molecule 1
$\mathcal{R}^1$: Molecule 1 Representation
$\mathcal{R}^2$: Molecule 2 Representation
Y : Target Value

➡ Model bias

- Based on information leakage,

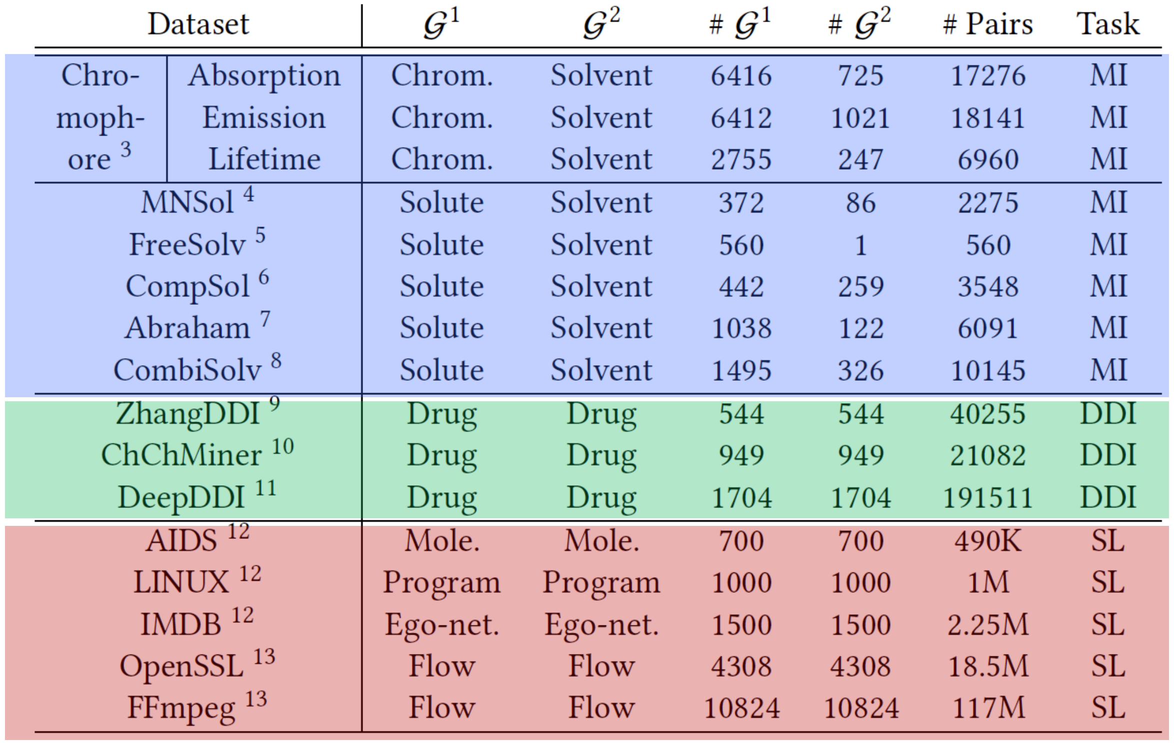
→ Model bias can be quantified based on mutual information

- Again, several backdoor paths are blocked by conditioning on $\mathcal{C}^1$ and $\mathcal{G}^2$

→ Enable the direct measure of model bias!
→ Finally, Loss term minimize the model bias

Shift-Robust Molecular Relational Learning with Causal Substructure. KDD 2023

Experiments Dataset description



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

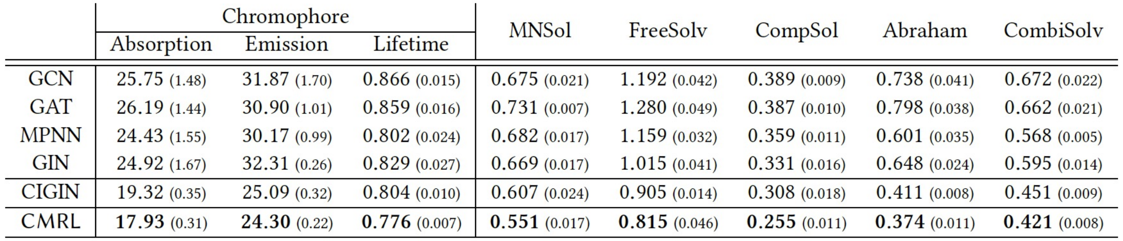
→ How similar are the paired graphs? (ex. GED)

→ AIDS, LINUX, IMDB, OpenSSL, Ffmpeg

→ Regression Task / Classification Task

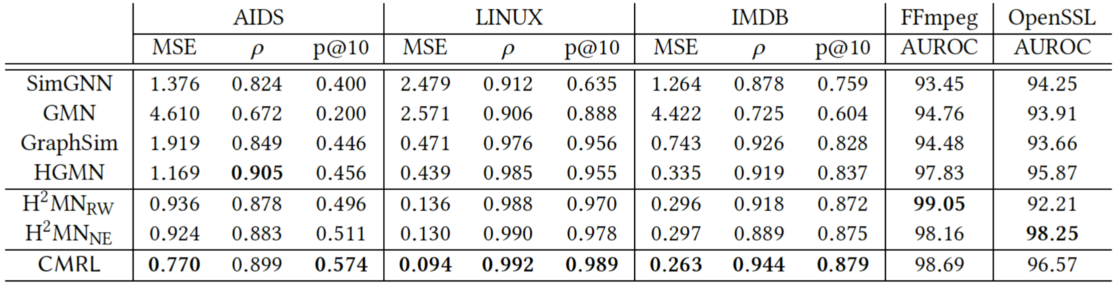
Shift-Robust Molecular Relational Learning with Causal Substructure. KDD 2023

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	**17.93** (0.31)	**24.30** (0.22)	**0.776** (0.007)	**0.551** (0.017)	**0.815** (0.046)	**0.255** (0.011)	**0.374** (0.011)	**0.421** (0.008)

Performance on molecular interaction prediction task



	AIDS			LINUX			IMDB			FFmpeg	OpenSSL
	MSE	ρ	p@10	MSE	ρ	p@10	MSE	ρ	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	**0.905**	0.456	0.439	0.985	0.955	0.335	0.919	0.837	97.83	95.87
H^2MN_{RW}	0.936	0.878	0.496	0.136	0.988	0.970	0.296	0.918	0.872	**99.05**	92.21
H^2MN_{NE}	0.924	0.883	0.511	0.130	0.990	0.978	0.297	0.889	0.875	98.16	**98.25**
CMRL	**0.770**	0.899	**0.574**	**0.094**	**0.992**	**0.989**	**0.263**	**0.944**	**0.879**	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

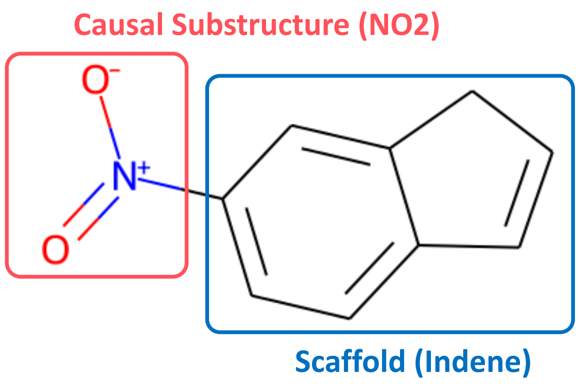
2. Wide applicability of CMRL beyond molecules

→ Performs well in dataset that contains core substructure

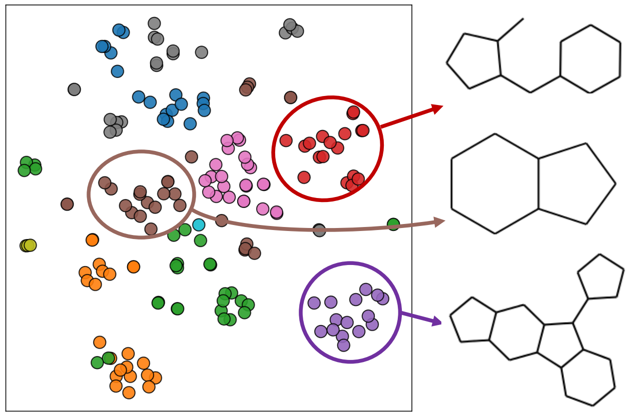
Shift-Robust Molecular Relational Learning with Causal Substructure. KDD 2023

Experiments Out-of-distribution performance

In out-of-distribution experiment, we assess the model's performance on molecules belonging to new scaffold classes

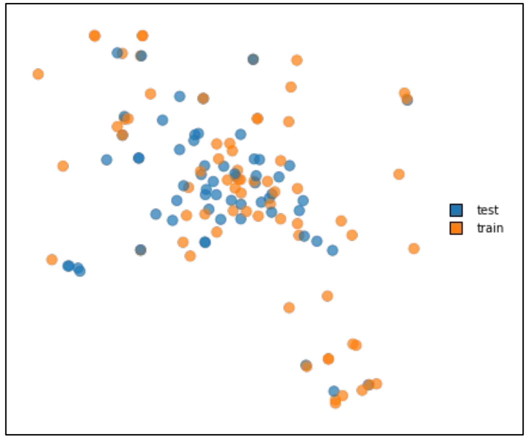


Molecule: 6-nitro-1H-indene

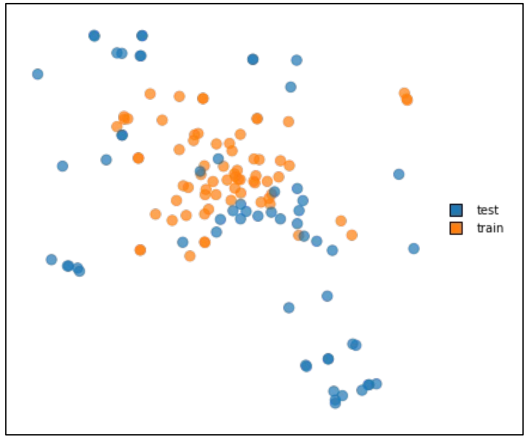


TSNE embeddings

Different scaffolds exhibit totally different distribution



Random Split



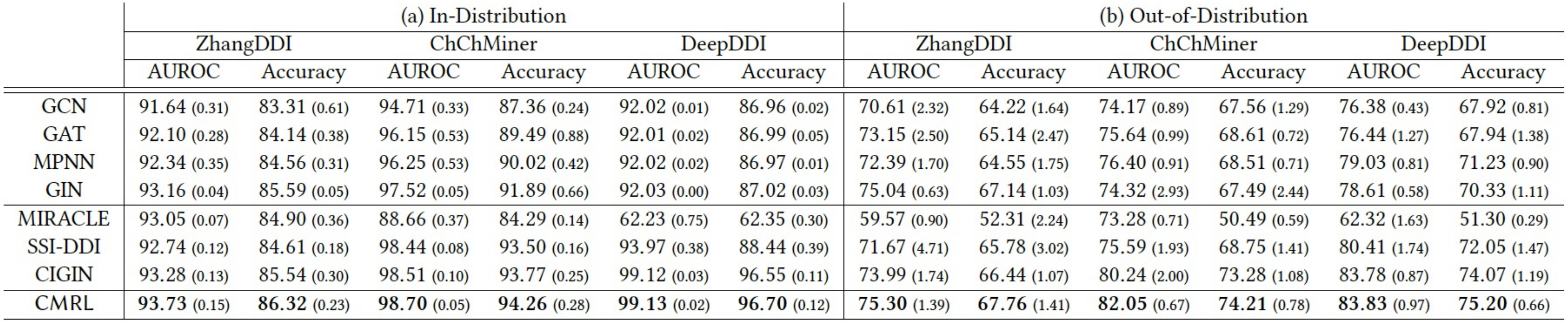
Scaffold Split

TSNE on splitted data (Train / Test)

Shift-Robust Molecular Relational Learning with Causal Substructure. KDD 2023

Experiments Out-of-distribution performance

In out-of-distribution experiment, we assess the model's performance on molecules belonging to new scaffold classes



	(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

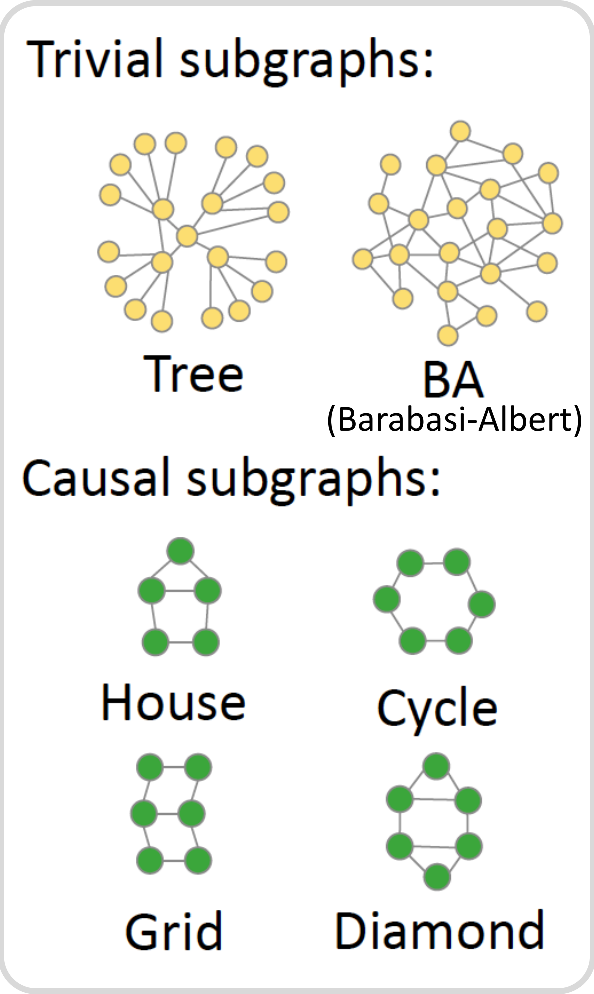
CMRL outperforms previous work on out-of-distribution scenarios

→ Learning causal substructure enhances the generalization ability of the model

Shift-Robust Molecular Relational Learning with Causal Substructure. KDD 2023

Experiments Synthetic dataset experiments

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



Positive pair

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

Negative pair

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

Dataset bias

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

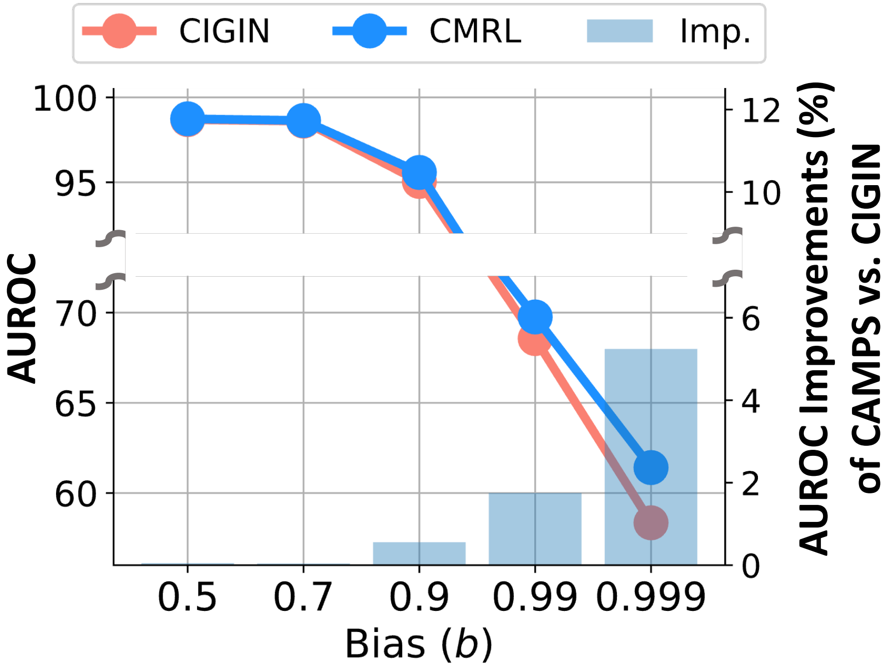
$$\text{bias}(b) = \frac{\text{Number of positive pairs with BA substructure}}{\text{Number of positive pairs}}$$
$$= \frac{\#\{\text{Causal-BA, Causal-BA}\}}{\#\{\text{Causal-Tree, Causal-Tree}\} + \#\{\text{Causal-BA, Causal-BA}\}}$$

- 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



<u>Observations</u>

1. Models' performance degrades as the bias gets severe

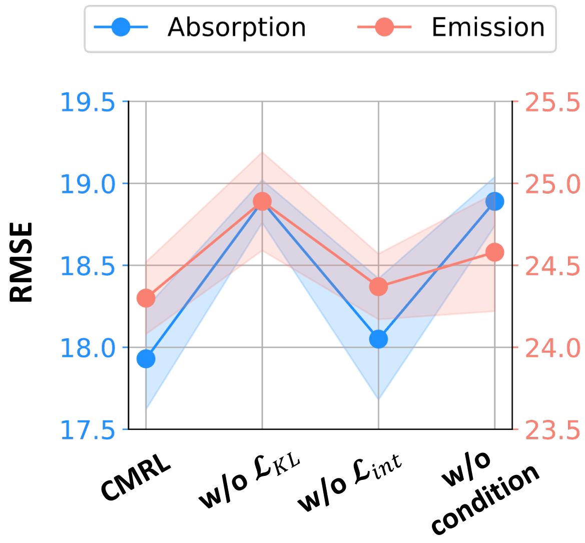
→ "BA" shortcut confound the model

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

→ Importance of learning causality between the substructure and target

Shift-Robust Molecular Relational Learning with Causal Substructure. KDD 2023

Experiments Model analysis

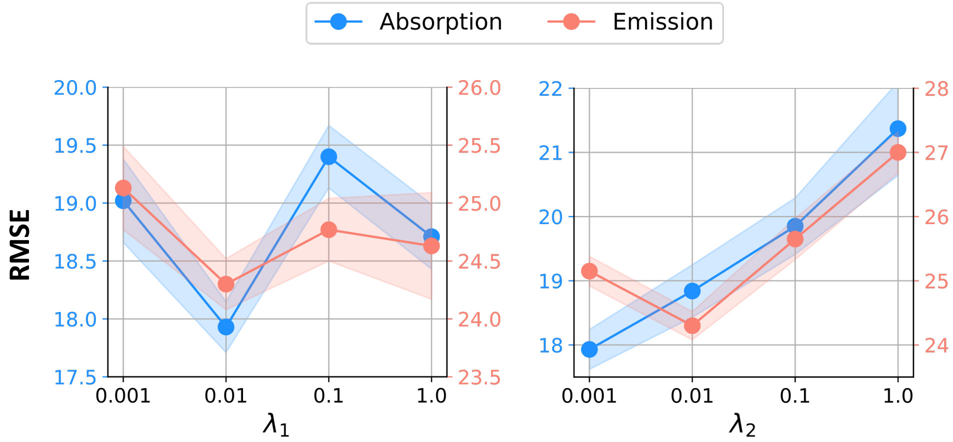


Observations in Ablation Studies

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



$\lambda_2 = 0.001$ for Absorption
$\lambda_2 = 0.01$ for Emission

$\lambda_1 = 0.01$ for Absorption
$\lambda_1 = 0.01$ for Emission

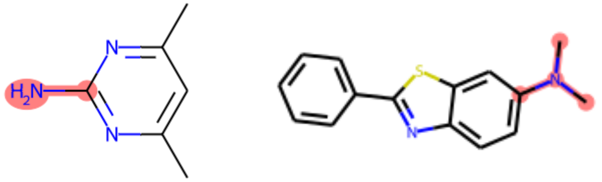
Observations in Sensitivity Analysis

1. **Optimal point for λ_2 exist balancing the noisiness and robustness**
2. **No certain relationship between model performance and λ_1**

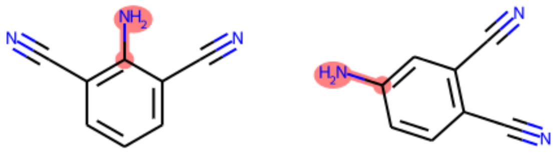
Training objective $\mathcal{L}_{final} = \mathcal{L}_{sup} + \mathcal{L}_{causal} + \lambda_1 \cdot \mathcal{L}_{KL} + \lambda_2 \cdot \mathcal{L}_{int}$

Shift-Robust Molecular Relational Learning with Causal Substructure. KDD 2023

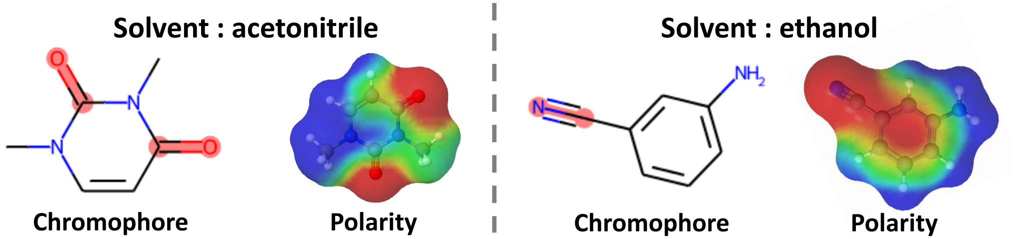
Experiments Qualitative analysis



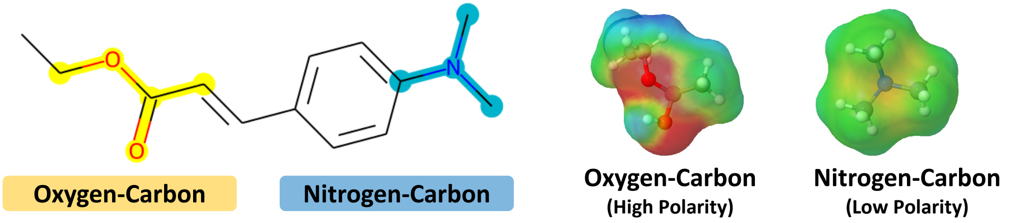
(a) Reaction with ordinary solvent



(b) Reaction with single bond



(c) Reaction with polar solvents



(d) Chromophore: EDAC

Solvents: 1-propanol, 1-butanol

Observations

1. Discovered causal substructure aligns to 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

Table of Contents

시간	주제	주요 내용
1	그래프 신경망 개요	- 그래프 신경망 전반적인 소개 - 그래프 종류에 따른 다양한 그래프 신경망 소개
2	소재 물성 예측 연구	- 소재 물성 예측 연구 최신 동향 소개 - Transformer 기반 모델 소개 및 extrapolation을 위한 모델 소개
3	물질 간 화학 반응 예측 연구	- 물질 간 화학 반응 예측 연구 동향 소개 - 정보 이론(Information bottleneck) 및 인과추론(Causal inference) 기반 모델 소개

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
  - Graph convolutional neural networks with global attention for improved materials property prediction. Physical Chemistry Chemical Physics 2020
  - Direct prediction of phonon density of states with Euclidean neural networks. Advanced Science 2021
  - **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: 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
  - SSI-DDI: substructure-substructure interactions for drug-drug interaction prediction. Briefings in Bioinformatics 2021
  - 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**

Thanks for listening!