PURDUE
UNIVERSITY

GCoM

Expressive and Interpretable Graph Neural Networks

Pan Li

10/03/2022
Talk at FastML workshop

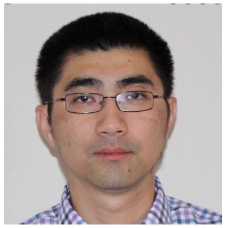
Collaborators

Siqi Miao

Daniel F. Guerrero

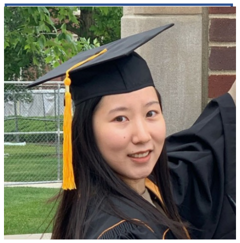
Mia Liu

Jacobo Konigsberg



Zhenbin Wu

Tianchun Li



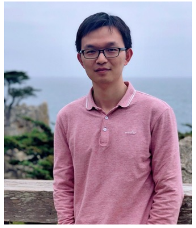
Shikun Liu

Yongbin Feng

Lisa Paspalaki

Nhan Tran

Yanbang Wang



Hongwei Wang

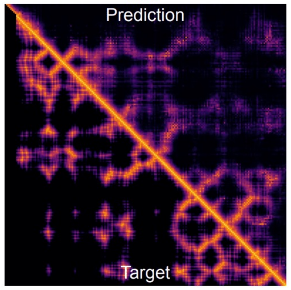
Jure Leskovec

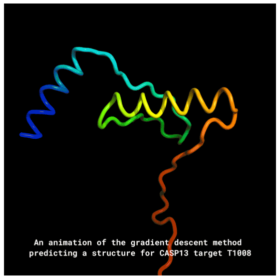
Yunan Luo

Deep Learning on Graphs in Science

- Protein folding

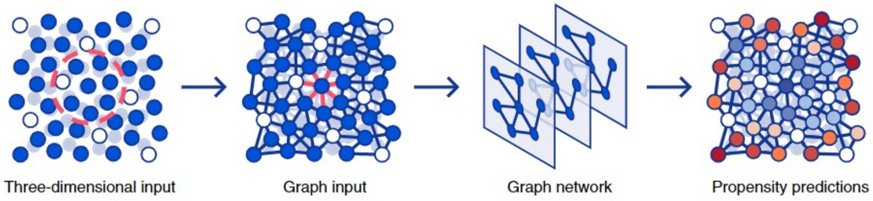
[Senior et al., Nature 2019]
[Jumper et al., Nature 2021]



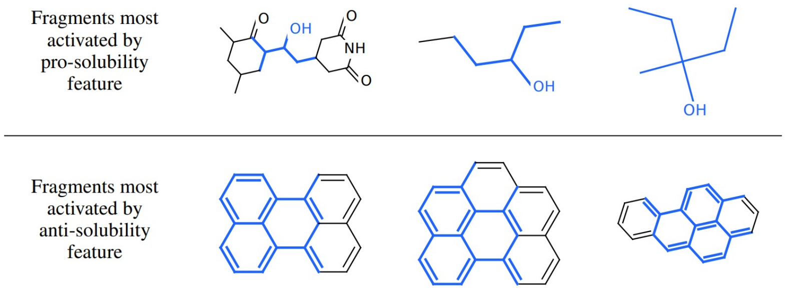

An animation of the gradient descent method predicting a structure for CASP13 target T1008

- Simulation of glass dynamics

[Baspt et al, Nature Physics 2021]

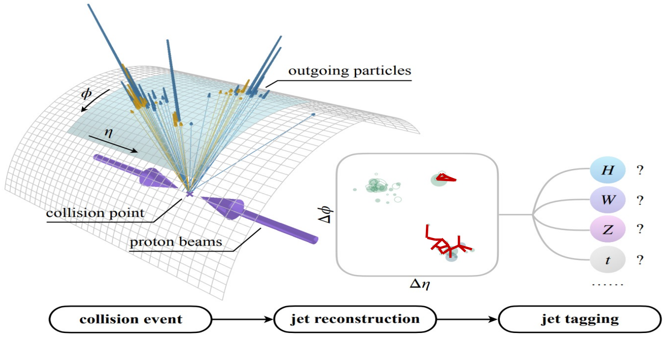


- Molecular Property Prediction


Fragments most activated by pro-solubility feature
O
OH
O
NH
O
OH
OH
Fragments most activated by anti-solubility feature

[Duvenaud et al., NeurIPS 2015]

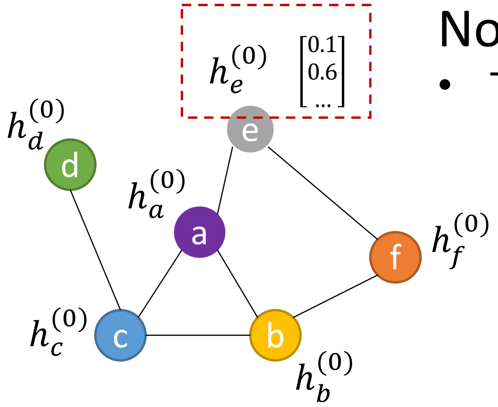
- Jet Tagging in HEP



Refined based on [Qu, Li, Qian, 2022]

Graph Neural Networks

Graph Data (A, X): the adjacency matrix A, possibly with node attributes X.

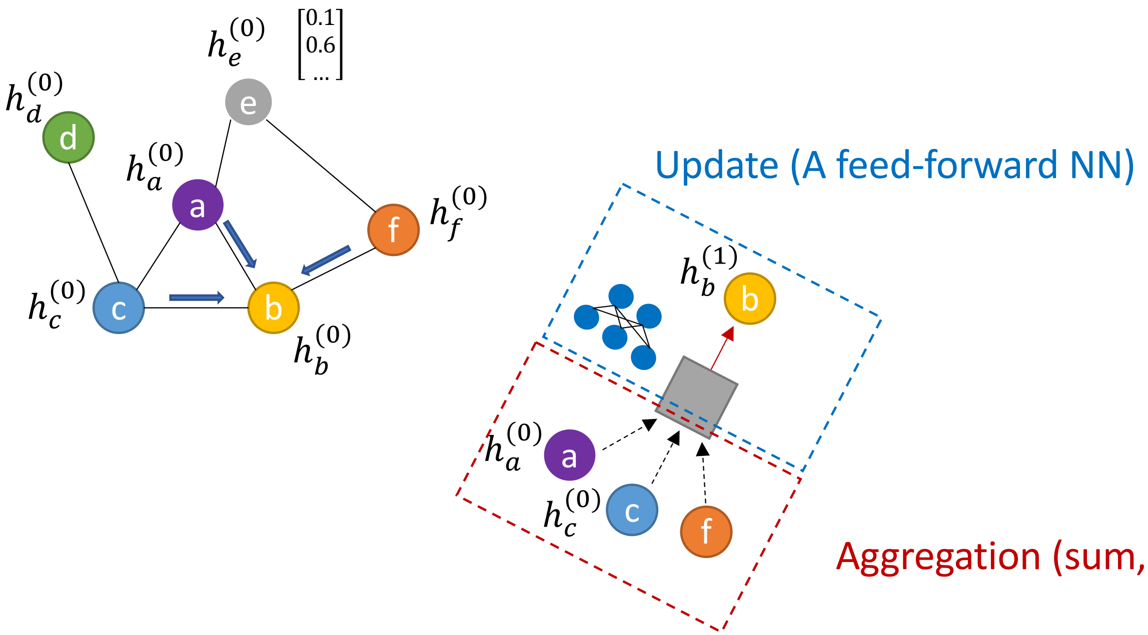


Node (feature) representation

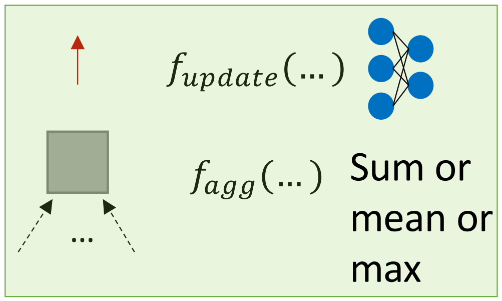
- Transformation of node attributes

Graph Neural Networks

Graph Data (A, X): the adjacency matrix A, possibly with node attributes X.



Aggregation (sum, mean, max pooling, attention, etc.)



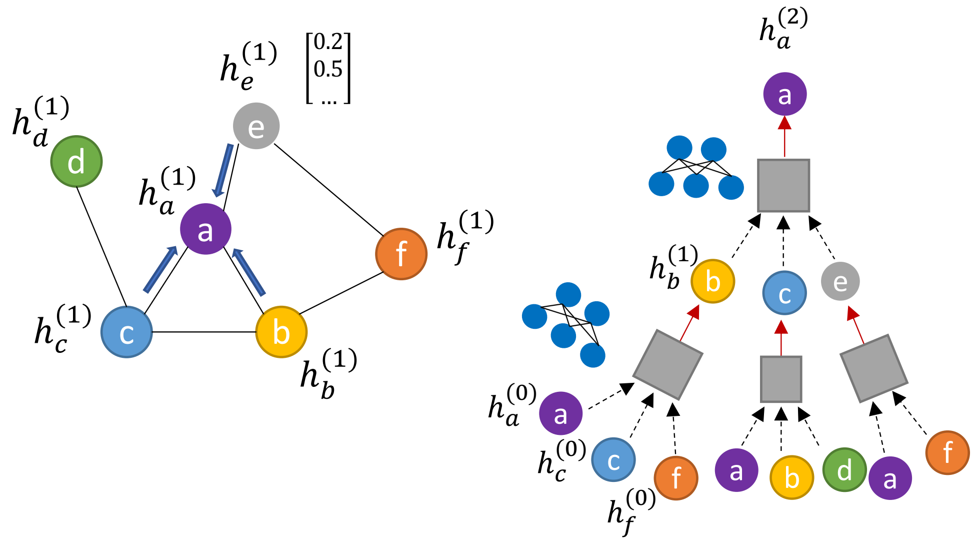
Graph neural network: one layer

$$h_v^{(t+1)} = f_{update}\left(h_v^{(t)}, f_{agg}\left(\{h_u^{(t)} \middle| u \in N_v\}\right)\right),$$

where N_v denotes the set of the neighbors of node v.

Graph Neural Networks

Graph Data (A, X): the adjacency matrix A, possibly with node attributes X.



Make prediction

1. [node level] Use node representations separately to predict node labels

2. [graph level] Aggregate all node representations to predict the graph label

$$h_G = \mathrm{POOL}\left(\left\{h_v^{(L)} \mid v \in V\right\}\right)$$

$$h_v^{(t+1)} = f_{update}\left(h_v^{(t)}, f_{agg}\left(\{h_u^{(t)} \middle| u \in N_v\}\right)\right),$$

where N_v denotes the set of the neighbors of node v.

Limitations of GNNs in Science

- Limited Expressive Power
  - Fail to represent some relations between input features and labels

- Hard to Interpret
  - Complicated architectures
  - Capture spurious correlations not effective patterns

- Subpar Generalization
  - Performance drop due to distribution shifts (simulation-based training -> real-data testing)

Limitations of GNNs in Science

- Limited Expressive Power
  - Fail to represent some relations between input features and labels

- Hard to Interpret
  - Complicated architectures
  - Capture spurious correlations not effective patterns

- Subpar Generalization
  - Performance drop due to distribution shifts (simulation-based training -> real-data testing)

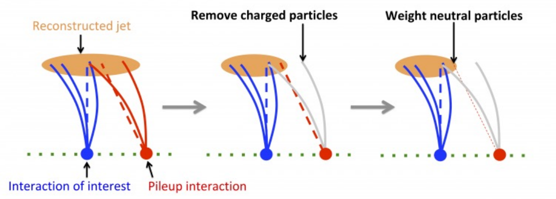


Image source: HOW CMS WEEDS OUT PARTICLES THAT PILE UP

Observed in pileup mitigation

Please check this with Shikun Liu on Wednesday.

Li et al., Semi-supervised Graph Neural Networks for Pileup Noise Removal, Neurips AI4Science workshop, 2021

Limitations of GNNs in Science

- Limited Expressive Power
  - Fail to represent some relations between input features and labels
- Hard to Interpret
  - Complicated architectures
  - Capture spurious correlations not effective patterns
- Subpar Generalization
  - Performance drop due to distribution shifts (simulation-based training -> real-data testing)

Expressive Power

- The target function $f: \mathcal{X} \rightarrow \mathcal{Y}$ --- unknown
- A model $f_\theta: \mathcal{X} \rightarrow \mathcal{Y}$ --- θ denotes the parameters

Can we expect $\sup_{X \in \mathcal{X}} |f(X) - f_\theta(X)|$ to be small for some θ?

For regular inputs $\mathcal{X} = \mathbb{R}^d$ and a fully-connected feedforward f_θ ✓

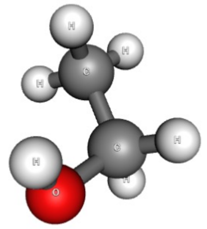
For graph inputs $\mathcal{X} = \mathcal{G} = \{0,1\}^{n \times n}$ and a GNN f_θ ✗

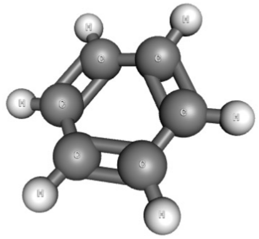
Hornik et al., Multilayer Feedforward Networks are Universal Approximators, Neural Networks, 1989

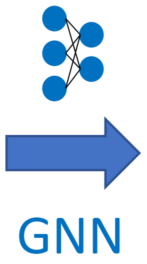
Xu et al., How powerful are graph neural networks? ICLR 2019,

Expressive Power for Graph-level Tasks

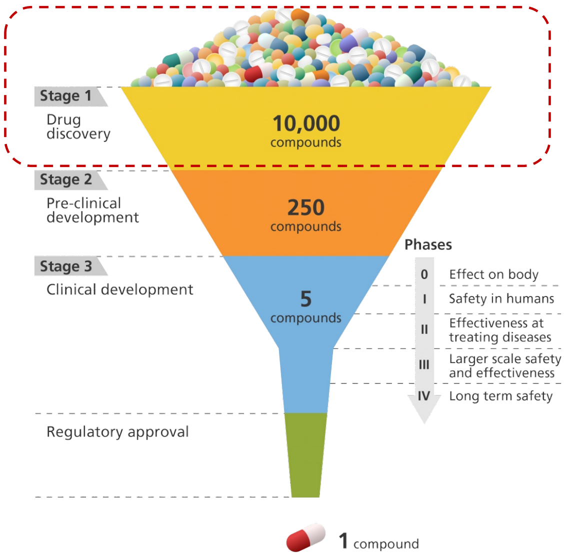
GNNs predict graph-level properties:







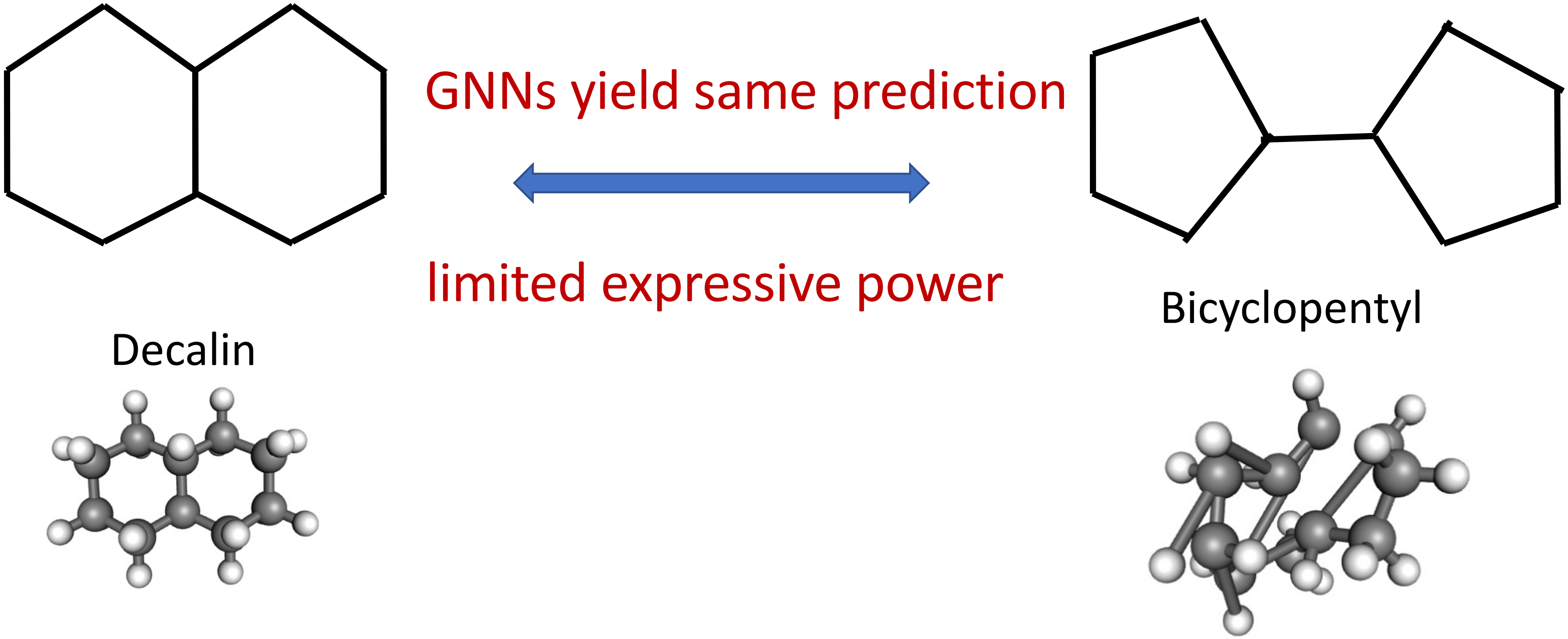
- Solubility
- Toxicity
- HOMO-LUMO energy gap
- effectiveness to certain disease



An illustration showing the different stages involved in developing a drug.
Image credit: Genome Research Limited

Expressive Power for Graph-level Tasks

GNNs fail in many cases. E.g., fail to give predictions of any different properties regarding the following molecules



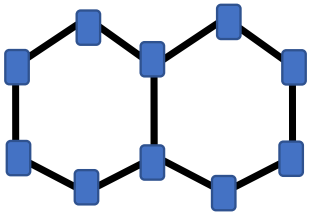
GNNs yield same prediction

limited expressive power

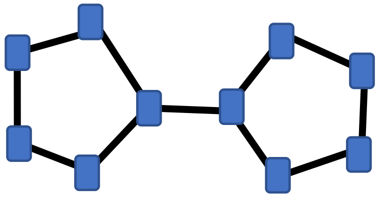
Decalin

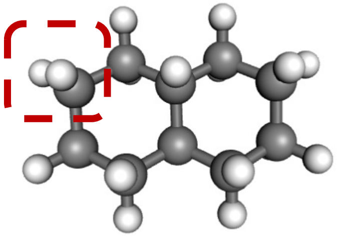
Bicyclopentyl

Expressive Power for Graph-level Tasks

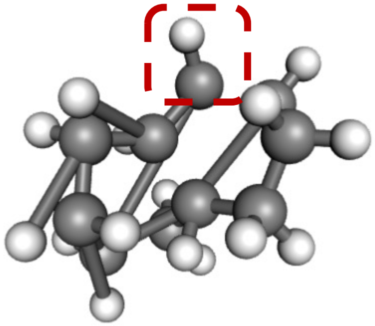


🟦 Node attributes

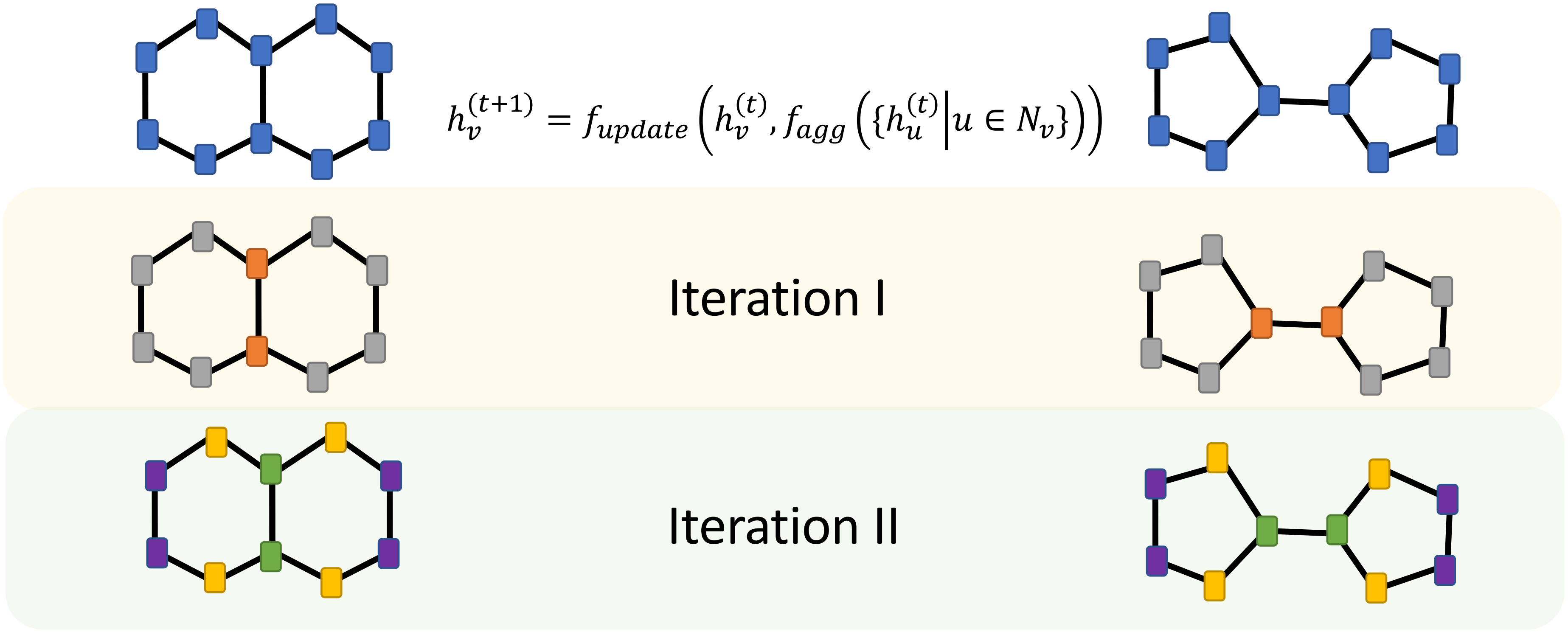




“one carbon atom with two hydrogen atoms” as a node



Expressive Power for Graph-level Tasks

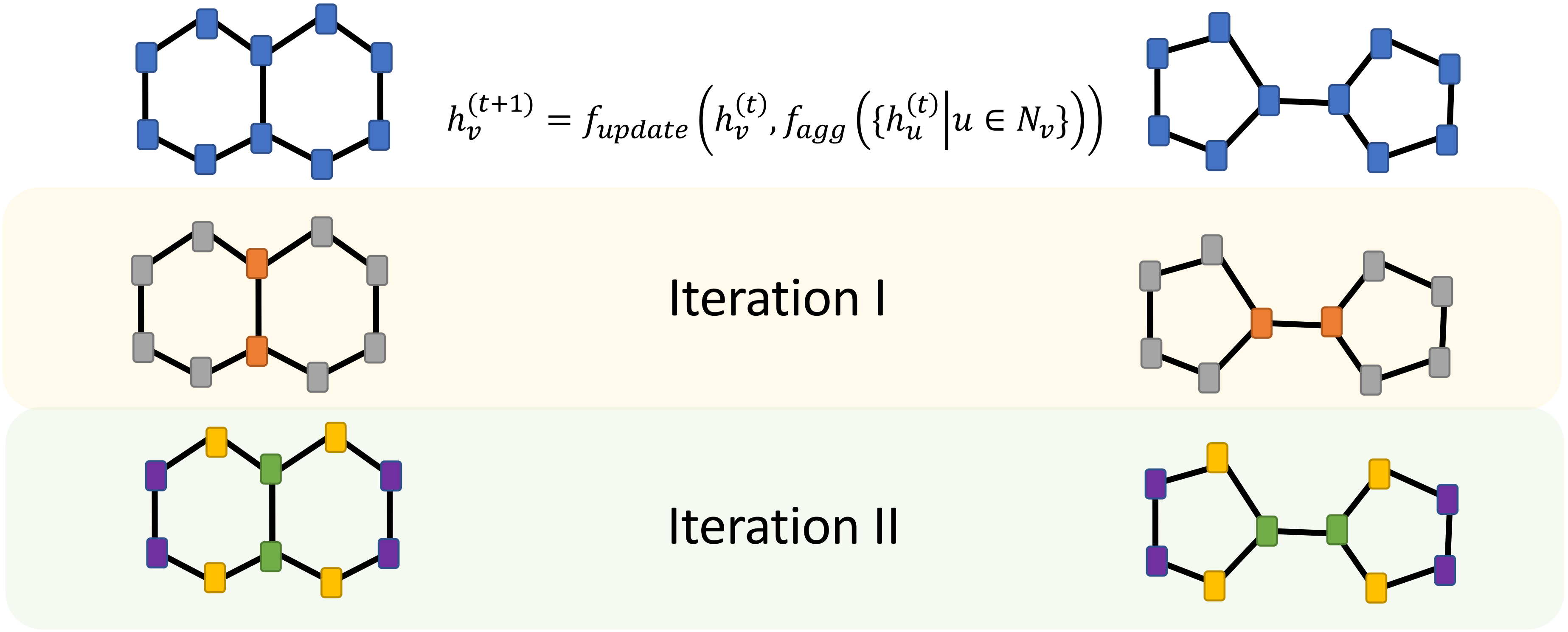


$$h_v^{(t+1)} = f_{update}\left(h_v^{(t)}, f_{agg}\left(\{h_u^{(t)} \middle| u \in N_v\}\right)\right)$$

Iteration I

Iteration II

Expressive Power for Graph-level Tasks



$$h_v^{(t+1)} = f_{update}\left(h_v^{(t)}, f_{agg}\left(\{h_u^{(t)} \middle| u \in N_v\}\right)\right)$$

Iteration I

Iteration II

…

Predict based on $h_G = \mathrm{POOL}\left(\left\{h_v^{(L)} \mid v \in V\right\}\right)$

The multi-sets of colors (node representations) on two graphs keep the same

No valid predictions…

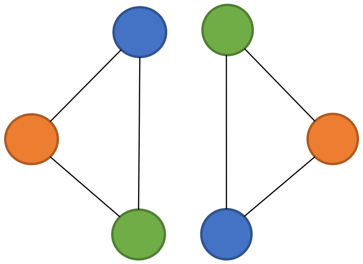
Expressive Power for Graph-level Tasks

- **Too symmetric? This is not an extreme case…**
  - Consider $A \in \mathbb{R}^{n \times n}$, $f(A) = trace(A^3)$
  - If let $A \in \{0,1\}^{n \times n}$ represent a graph, $A_{uv} = 1$ if (u, v) is an edge, $f(A)$ outputs the number of 3-cycles.

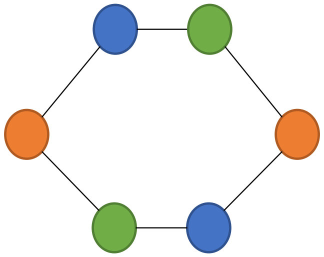
Expressive Power for Graph-level Tasks

- **Too symmetric? This is not an extreme case…**
  - Consider $A \in \mathbb{R}^{n\times n}$, $f(A) = trace(A^3)$
  - If let $A \in \{0,1\}^{n\times n}$ represent a graph, $A_{uv} = 1$ if (u, v) is an edge, $f(A)$ outputs the number of 3-cycles.
  - Consider a GNN $f_\theta(\cdot)$.

Have different numbers of 3-cycles while GNNs give them the same prediction



v.s.

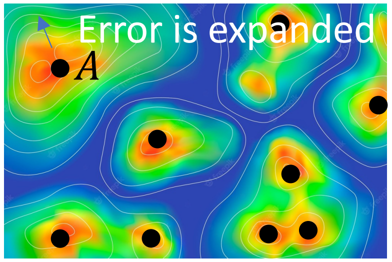


Expressive Power for Graph-level Tasks

- **Too symmetric? This is not an extreme case…**
  - Consider $A \in \mathbb{R}^{n\times n}$, $f(A) = trace(A^3)$
  - If let $A \in \{0,1\}^{n\times n}$ represent a graph, $A_{uv} = 1$ if (u, v) is an edge, $f(A)$ outputs the number of 3-cycles.
  - Consider a GNN $f_\theta(\cdot)$. $f_\theta(\cdot)$ cannot approximate $f(\cdot)$
  - A lot of input A's may cause such errors

Error for $A \in \mathbb{R}^{n\times n}$:

$$|f_\theta(A) - f(A)|$$



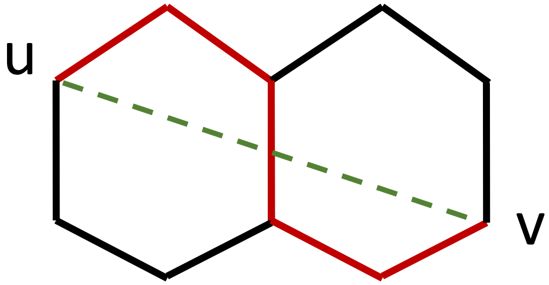
Note that f_θ is continuous

Solutions for Graph-level Expressive Power

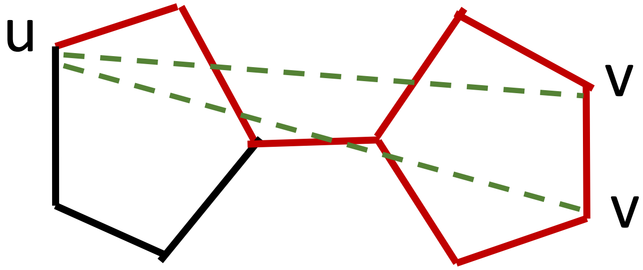
Solutions for Graph-level Expressive Power

Let us consider the 0-1 case: $A \in \{0,1\}^{n \times n}$
(a graph without weights on edges)

One key idea: Injecting structural (e.g., distance) features



For any node u, there is at most one node v whose shortest path distance to u is 5.

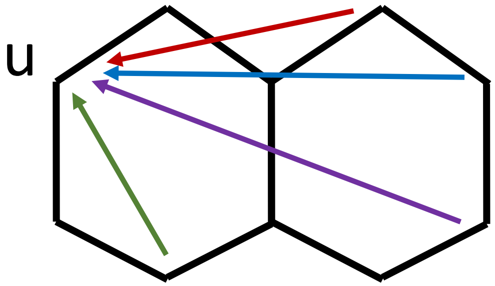


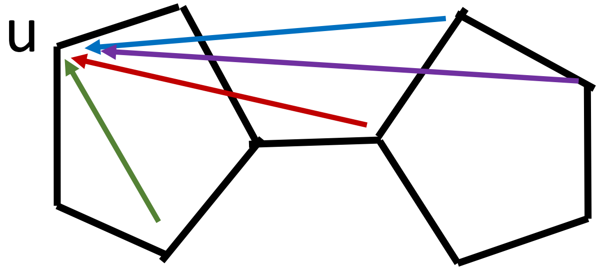
There exists a node u such that there are two nodes whose shortest path distance to u are 5.

Li et al., Distance encoding: Design provably more powerful neural networks for graph representation learning, NeurIPS 2020

Solutions for Graph-level Expressive Power

Let us consider the 0-1 case: $A \in \{0,1\}^{n \times n}$





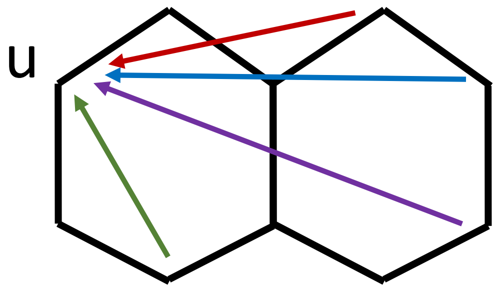
- Build a new fully connected graphs (transformers)
- Use distance over the original graph as edge features on the new graph

Graphomer achieves top-1 in KDD Cup's 2021 to predict molecular properties

Ying et al., Do Transformers Really Perform Bad for Graph Representation? NeurIPS 2021

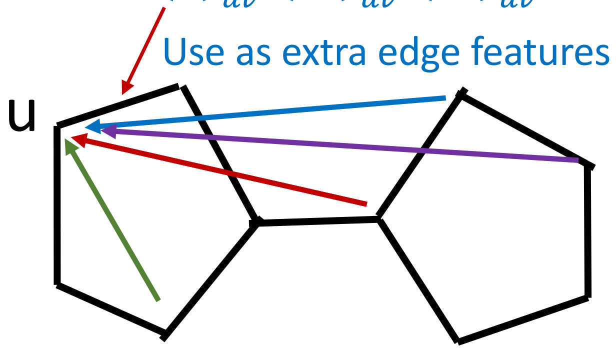
Solutions for Graph-level Expressive Power

How about the case when $A \in \mathbb{R}^{n \times n}$?



$[(\hat{A})_{uv}, (\hat{A}^2)_{uv}, (\hat{A}^3)_{uv}, \ldots]$

Use as extra edge features



- Build a new fully connected graph (transformers)
- Use $[(\hat{A})_{uv}, (\hat{A}^2)_{uv}, (\hat{A}^3)_{uv}, \ldots]$ as edge features for (u,v)

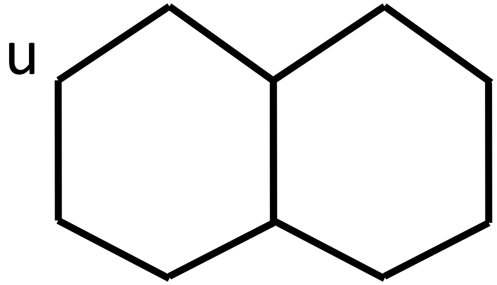
$\hat{A}$: Adding some row/column normalization is good for numerical stability

A more general structural feature

Chen et al., on the equivalence between graph isomorphism testing and function approximation with gnns. NeurIPS 2019

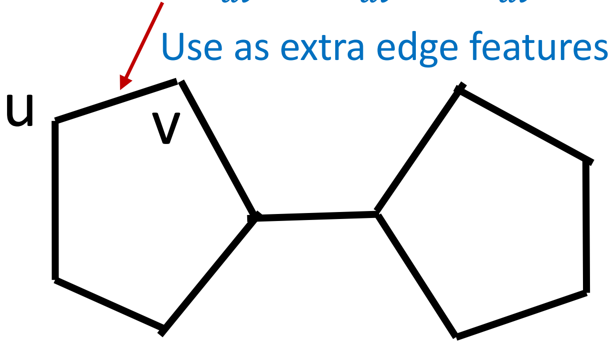
Solutions for Graph-level Expressive Power

How about the case when $A \in \mathbb{R}^{n\times n}$?



$[(\hat{A})_{uv}, (\hat{A}^2)_{uv}, (\hat{A}^3)_{uv}, \ldots]$

Use as extra edge features



For complexity consideration, this can be removed.

- Build ~~fully connected graphs~~ (transformers)
- Use $[(\hat{A})_{uv}, (\hat{A}^2)_{uv}, (\hat{A}^3)_{uv} \ldots]$ as edge features

$\hat{A}$: Adding some row/column normalization is good for numerical stability

Solutions for Graph-level Expressive Power

- Higher-order tensors: Computation complexity is high

Maron et al., Provably powerful graph networks, NeurIPS 2019

- Add random node features: Training is hard to converge

Sato et al., Random features strengthen graph neural networks, SDM 2021

Abboud et al., The surprising power of graph neural networks with random node initialization, IJCAI 2021

Limitations of GNNs in Science

- Limited Expressive Power
  - Fail to represent some relations between input features and labels

- Hard to Interpret
  - Complicated architectures
  - Capture spurious correlations not effective patterns

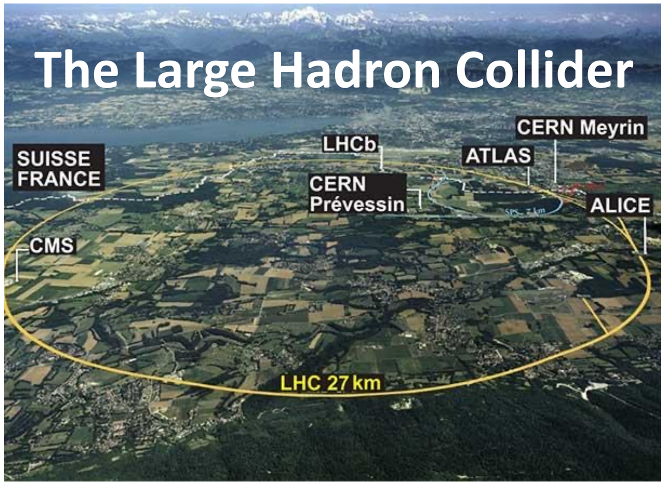
- Subpar Generalization
  - Performance drop due to distribution shifts

(simulation-based training -> real-data testing)

$\tau \rightarrow 3\mu$ Detection

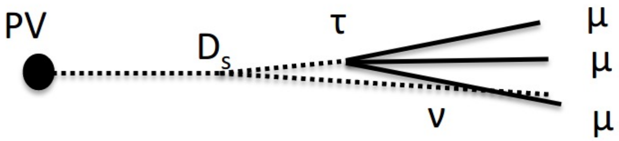
❑ Motivation

- Physics beyond the Standard Model
  - Search for **charged lepton flavor violating** decays
  - $\tau \longrightarrow 3\mu$ is the **cleanest signature**
- Extremely small branching ratio
  - Though may be enhanced by BSM physics
  - $\mathrm{BR}(\tau \rightarrow 3\mu) \sim O(10^{-8})$



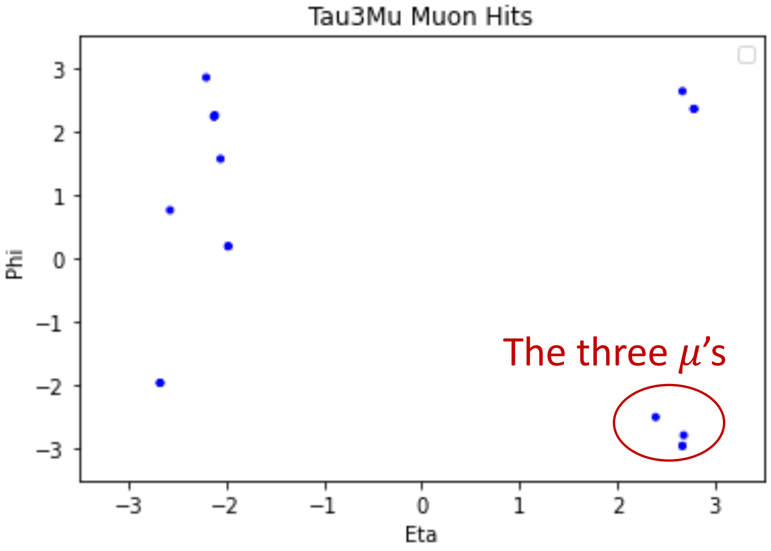
❑ Given an ML model, we want

- High trigger efficiency
- Low trigger rate



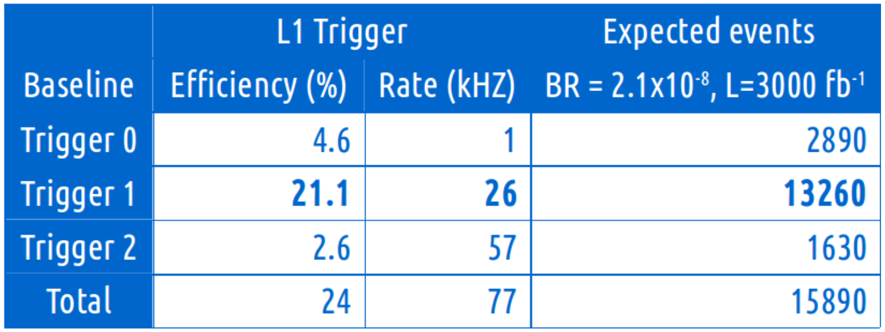
GNNs give super performance

❑ **We use muon hits left in the muon stations to make prediction.**



The three μ's

Traditional methods (pattern matching)



	L1 Trigger		Expected events
Baseline	Efficiency (%)	Rate (kHZ)	BR = 2.1×10^{-8}, L=3000 fb^{-1}
Trigger 0	4.6	1	2890
Trigger 1	**21.1**	**26**	**13260**
Trigger 2	2.6	57	1630
Total	24	77	15890

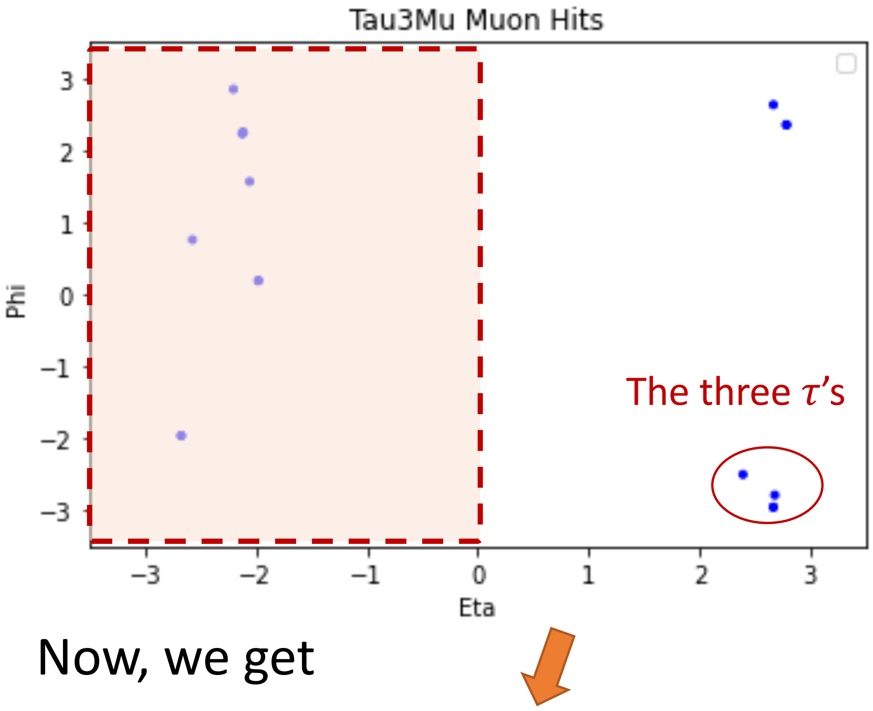
GNN-based methods

<u>92% efficiency @ 10kHz rate</u>

Can we trust this performance?

Problem: Spurious Correlations

- Positive samples: Only use the endcap (a half of space Eta>0 or Eta< 0) without true signals
- Negative samples: Randomly choose one endcap (a half of space Eta>0 or Eta< 0)

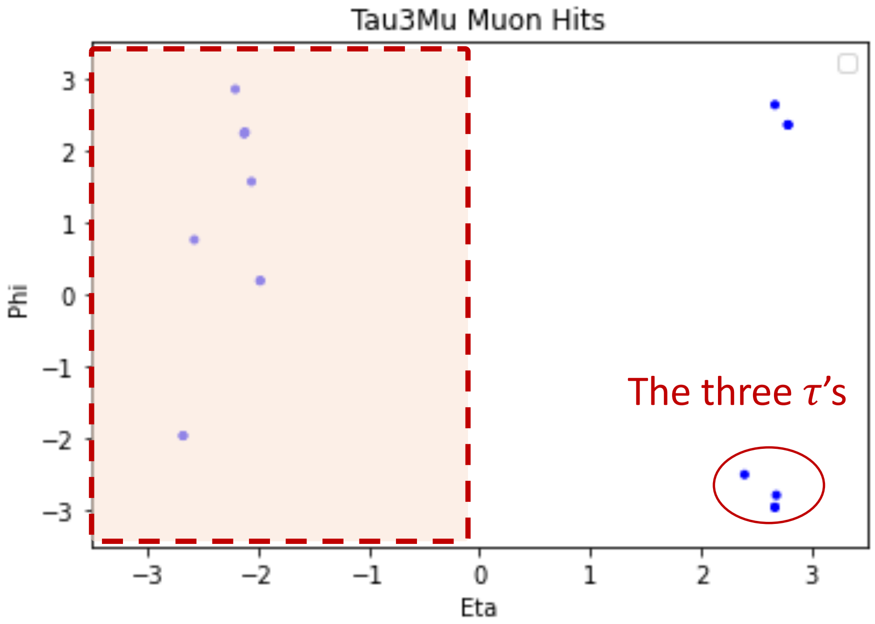


Now, we get

87% efficiency @ 10kHz rate

Problem: Spurious Correlations

- Positive samples: Only use the endcap (a half of space Eta>0 or Eta< 0) without true signals
- Negative samples: Randomly choose one endcap (a half of space Eta>0 or Eta< 0)



Why can it happen？

Either the simulator or pre-processing injects spurious correlations.

Now, we get

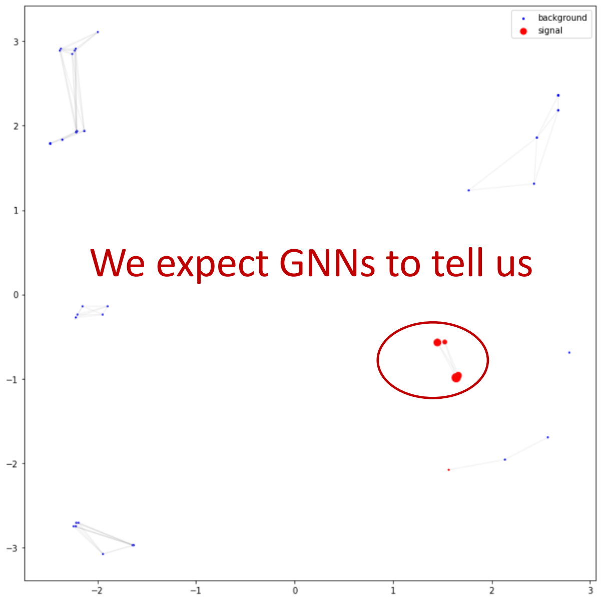
87% efficiency @ 10kHz rate

Two endcaps: **92% efficiency @ 10kHz rate**

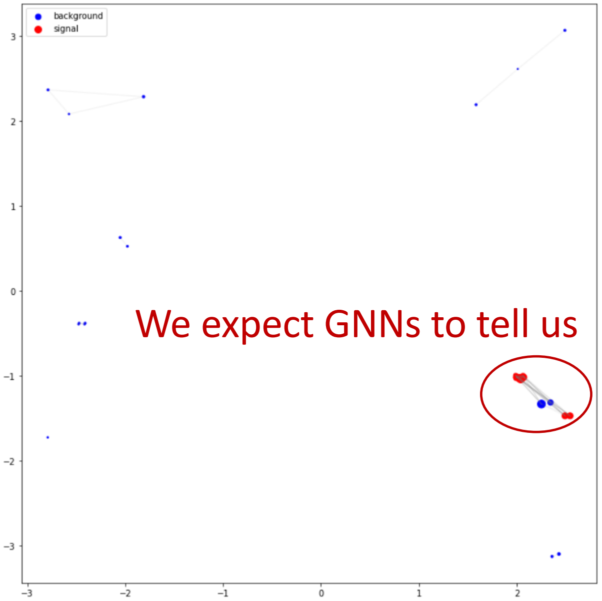
Traditional: **24% efficiency @ 77kHz rate**

Design Interpretable and Trustworthy GNNs

Can we **check patterns learned** by GNNs to see if we can trust them?



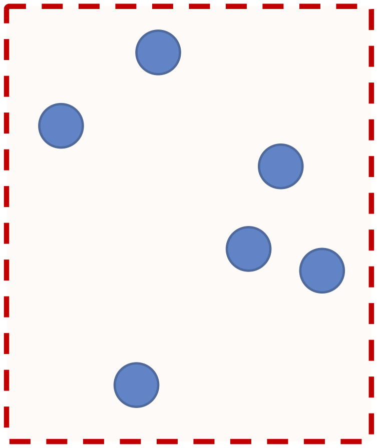
We expect GNNs to tell us



We expect GNNs to tell us

Solution: Learnable Randomness Injection

Constrain the amount of information that the model can use from the data

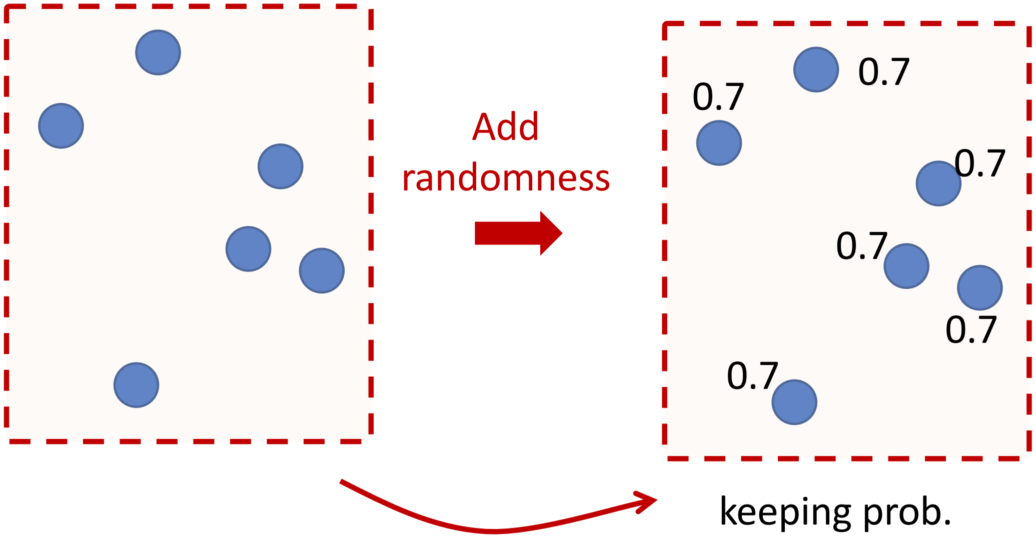


Miao et al., Interpretable geometric deep learning via learnable randomness injection, in submission

Miao et al., Interpretable and Generalizable Graph Learning via Stochastic Attention Mechanism, ICML 2022

Solution: Learnable Randomness Injection

- A trainable model output the probabilities to drop/keep points



Add randomness

0.7

0.7

0.7

0.7

0.7

0.7

keeping prob.

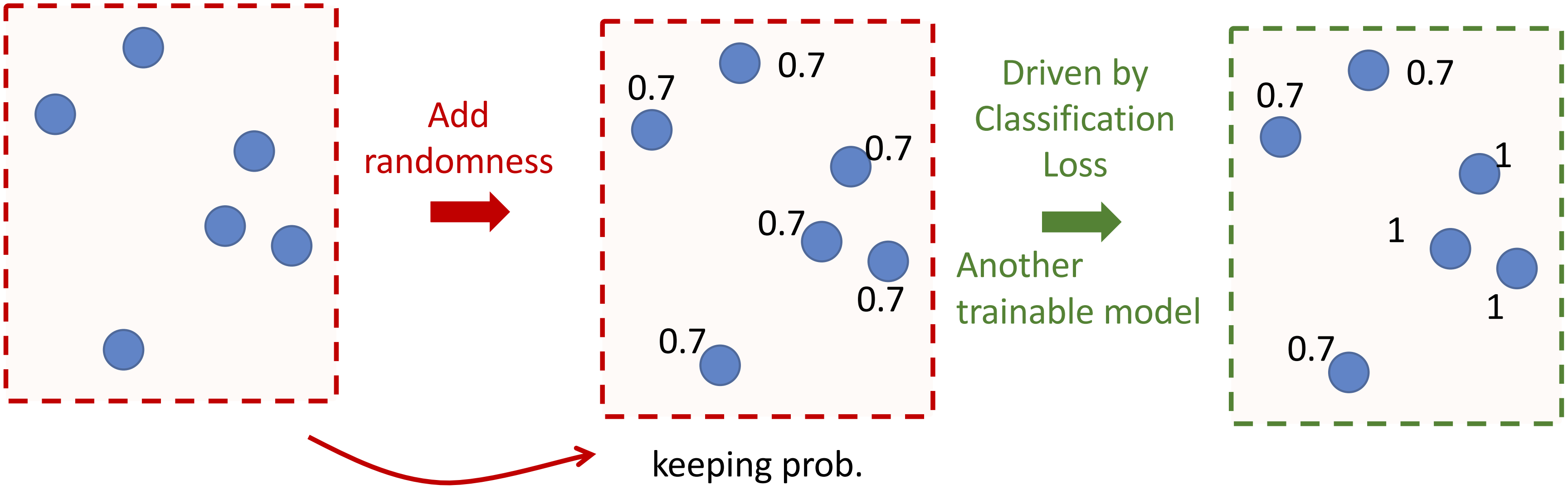
a trainable model

(graph construction + GNN)

Miao et al., Interpretable geometric deep learning via learnable randomness injection, in submission

Solution: Learnable Randomness Injection

- A trainable model output the probabilities to drop/keep points
- Another trainable model encodes the perturbed data to predict labels



Add randomness

Driven by Classification Loss

Another trainable model

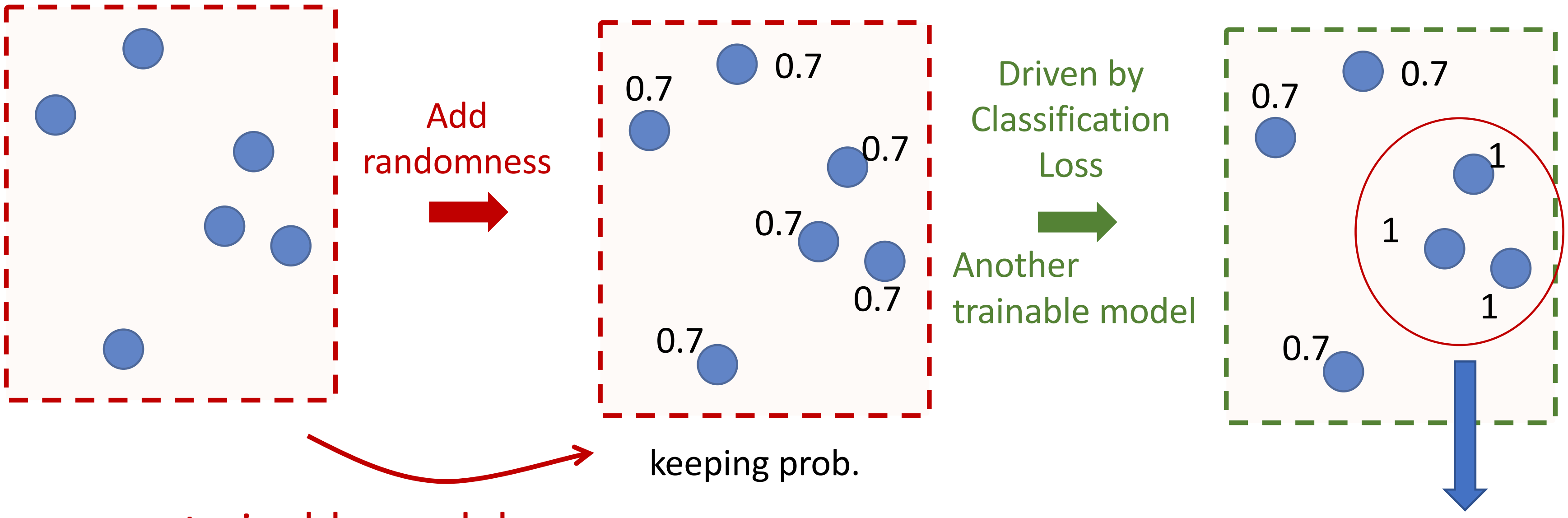
keeping prob.

a trainable model
(graph construction + GNN)

Miao et al., Interpretable geometric deep learning via learnable randomness injection, in submission

Solution: Learnable Randomness Injection

- A trainable model output the probabilities to drop/keep points
- Another trainable model encodes the perturbed data to predict labels
- Rank the probabilities to provide important patterns



Add randomness

Driven by Classification Loss

Another trainable model

keeping prob.

a trainable model

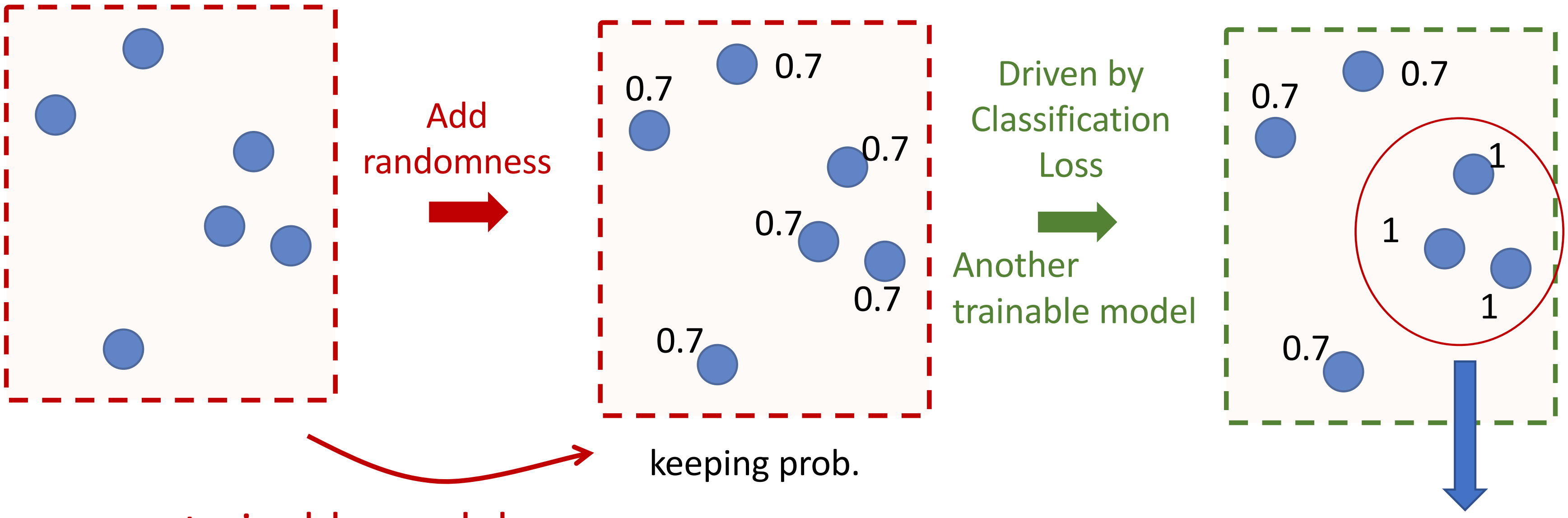
(graph construction + GNN)

Provide interpretation for checking

Miao et al., Interpretable geometric deep learning via learnable randomness injection, in submission

Solution: Learnable Randomness Injection

- A trainable model output the probabilities to drop/keep points
- Another trainable model encodes the perturbed data to predict labels
- Rank the probabilities to provide important patterns
- The detected points by our methods match the $\tau \rightarrow 3\mu$ signals with 80% ROC AUC



Add randomness

Driven by Classification Loss

Another trainable model

keeping prob.

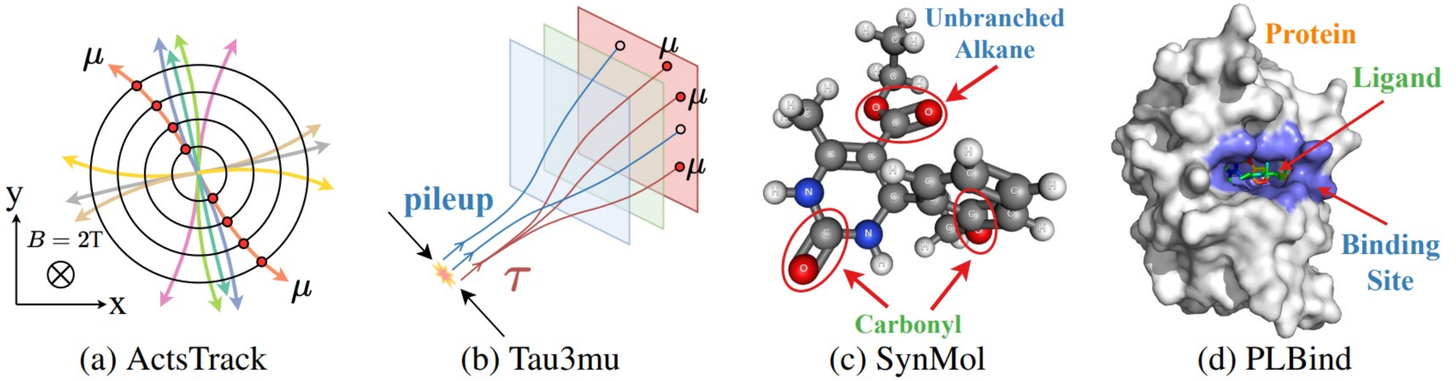
a trainable model
(graph construction + GNN)

Provide interpretation for checking

Miao et al., Interpretable geometric deep learning via learnable randomness injection, in submission

Check Papers and Code

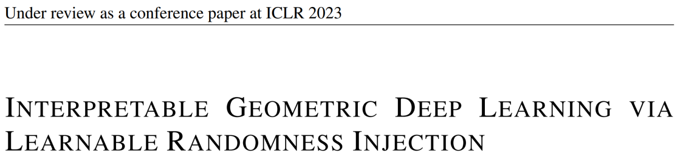
More Applications



(a) ActsTrack (b) Tau3mu (c) SynMol (d) PLBind

Figure 1: Illustrations of the four scientific datasets in this work to study interpretable GDL models.

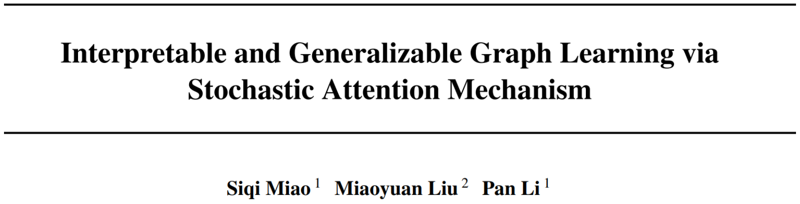
Point cloud part is under review at ICLR 2023



Under review as a conference paper at ICLR 2023

INTERPRETABLE GEOMETRIC DEEP LEARNING VIA LEARNABLE RANDOMNESS INJECTION

Applied to 2-D molecules



Interpretable and Generalizable Graph Learning via Stochastic Attention Mechanism

Siqi Miao [1] **Miaoyuan Liu** [2] **Pan Li** [1]

Code is online:
https://github.com/Graph-COM/GSAT

Miao et al., Interpretable and Generalizable Graph Learning via Stochastic Attention Mechanism, ICML 2022
Miao et al., Interpretable geometric deep learning via learnable randomness injection, in submission

Takeaways

Three problems of GNNs in scientific applications…

- Limited Expressive Power

  Adding structural features, e.g., $[(\hat{A})_{uv}, (\hat{A}^2)_{uv}, (\hat{A}^3)_{uv}, \ldots]$ as edge features

- Hard to Interpret

  Constraining information during the model training by adding randomness

- Subpar Generalization

Takeaways

Three problems of GNNs in scientific applications…

- Limited Expressive Power

  Adding structural features, e.g., $[(\hat{A})_{uv}, (\hat{A}^2)_{uv}, (\hat{A}^3)_{uv}, \ldots]$ as edge features

- Hard to Interpret and trust

  Constraining information during the model training by adding randomness

- Subpar Generalization

Thank you!