Graph Neural Networks (GNNs)

Concepts, Implementations and Applications

Tirtharaj Dash

The Boolean Lab
University of California, San Diego
CA 92093, USA

October 9, 2022

UC San Diego
Boolean Lab

Graphs

Graph: Represents the relations between a collection of objects.

Example: Molecule-molecule interactions

- Objects: a set of molecules
- Relations: interactions between different molecues

Example: Atom-level intaractions within a molecule

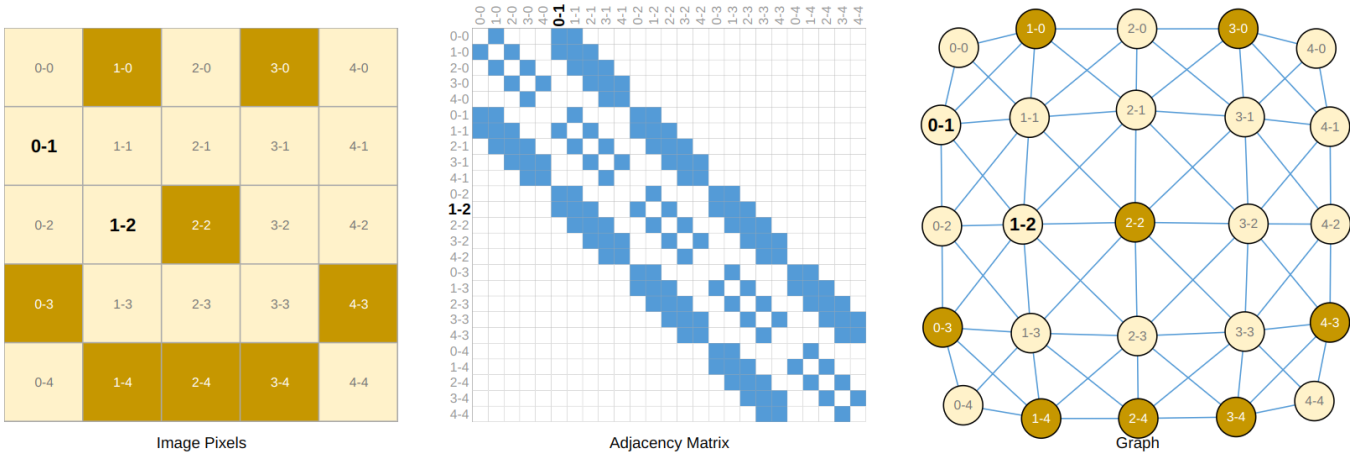
- Objects: a set of atoms
- Relations: different bonds between the atoms

UC San Diego
Boolean Lab

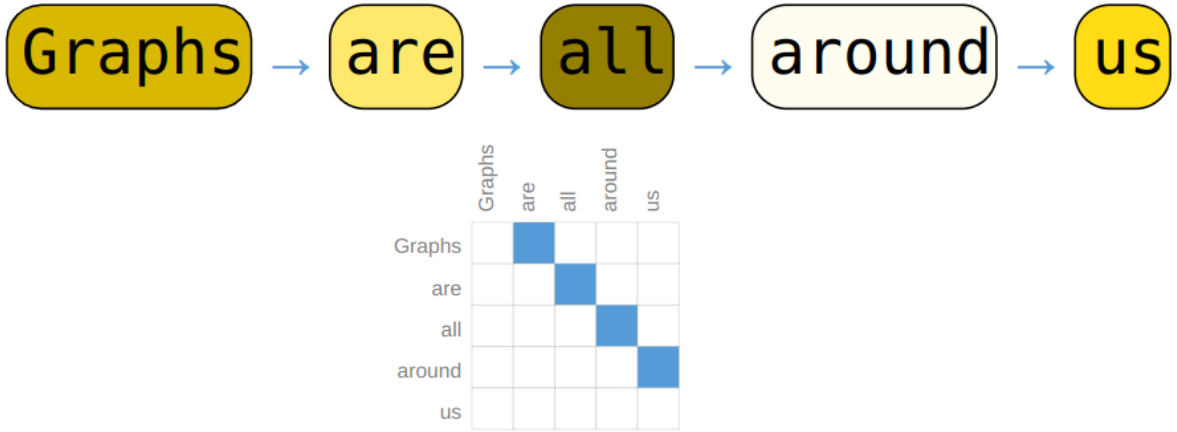
Graphs

Where else can we find graphs?

- Images as graphs



- Text as graphs



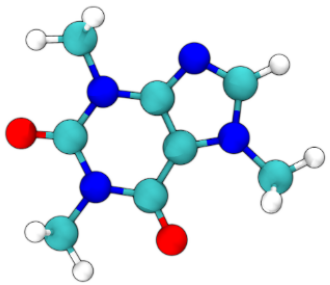
Graphs

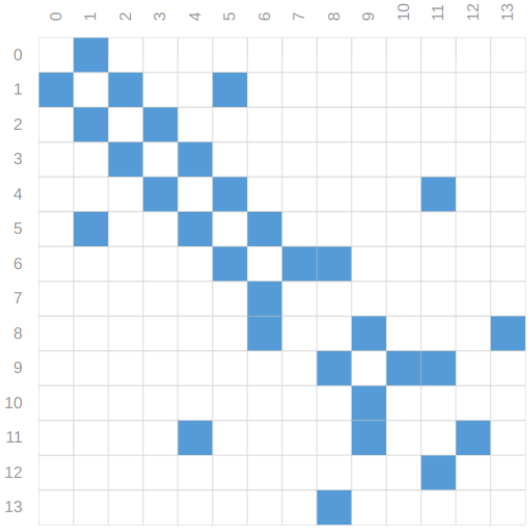
- Images and texts have some regular structures:
  - Pixels are connected in a grid.
  - Texts have line-like structure.

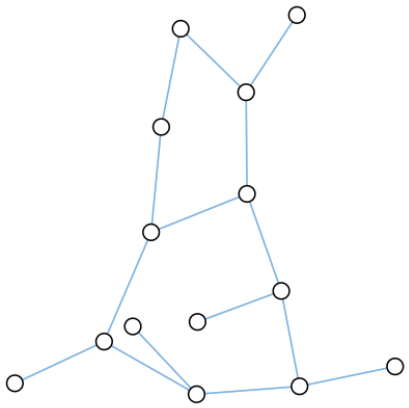
UC San Diego
Boolean Lab

Graphs

- There are, however, more complicated (heterogeneous) structures:







- Similar structures are seen in social networks as well.

Graphs

And, the simple definitions (let's just recall it):

Graph

1. A graph G is a pair (V, E) where V is a set of vertices (representing objects, E is a set of edges (representing relations and a subset of $V \times V$.
2. G is undirected if for every $(u, v) \in E$, $(v, u) \in E$.
3. A convenient way to represent a graph is by using a *adjacency matrix* $\mathbf{A} \in \mathbb{R}^{|V| \times |V|}$, where $\mathbf{A}[u, v] = 1$ if $(u, v) \in E$ and $\mathbf{A}[u, v] = 0$ otherwise.

Q. In practice, what can go wrong with representing graphs by adjacency matrix?

UC San Diego
Boolean Lab

Graphs

A. Memory requirement: $O(|V|^2)$

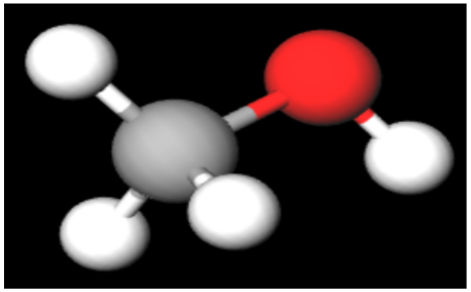
- Representing large graphs
- Representing sparse graphs

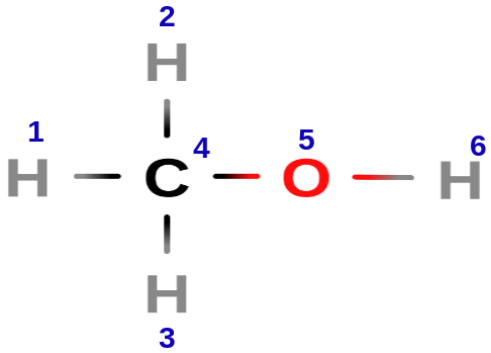
S. Use adjacency list: $O(|E|)$

UC San Diego
Boolean Lab

Graphs

Labelled graph: *Feature* information associated with a graph





Something like this, for the vertices:

$$
\begin{aligned}
v_1 &: [0,1,0] \\
v_2 &: [0,1,0] \\
v_3 &: [0,1,0] \\
v_4 &: [1,0,0] \\
v_5 &: [0,0,1] \\
v_6 &: [0,1,0]
\end{aligned}
$$

UC San Diego
Boolean Lab

Now,

Labelled Graph

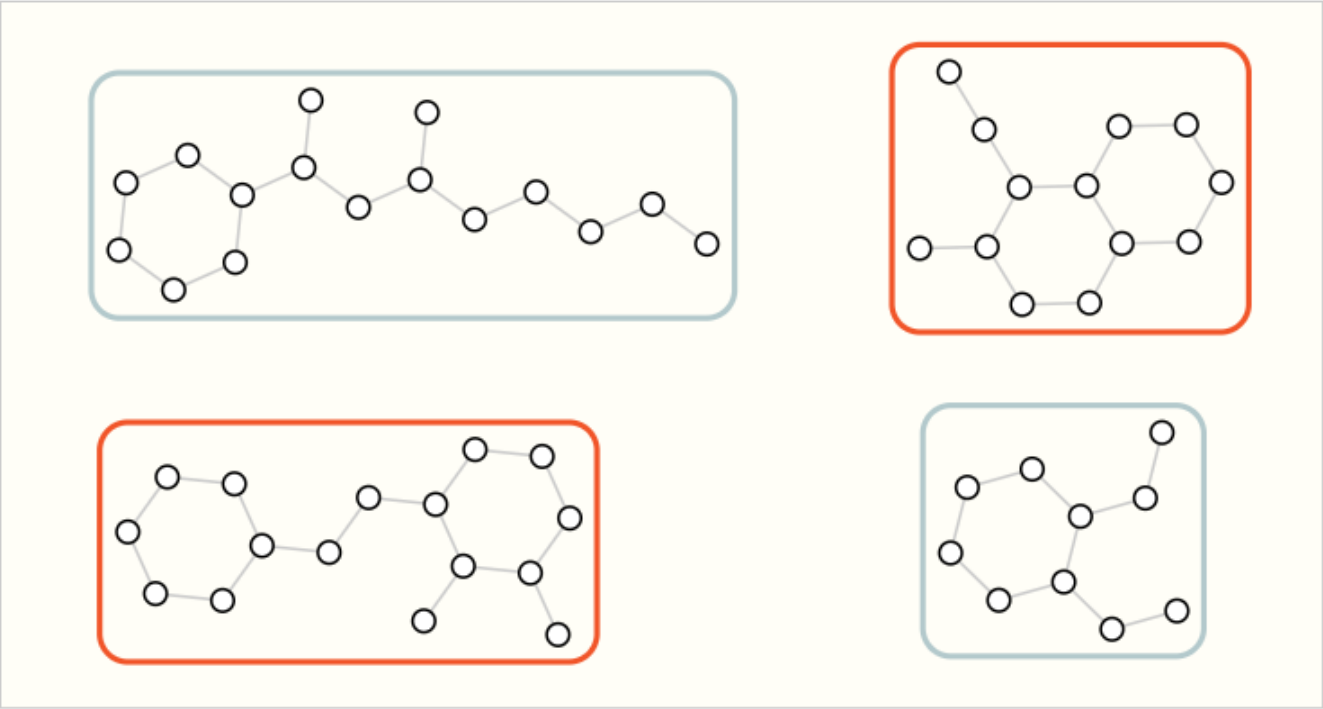
A labelled graph $G \in \mathcal{G}$ is represented as: $(V, E, \sigma, \psi, \epsilon)$, where

- V: a set of vertices
- E: a set of edges
- σ: neighbourhood function $\sigma : V \rightarrow 2^V$
- ψ: vertex-labelling function $\psi : V \rightarrow 2^{\mathcal{V}}$
- ϵ: edge-labelling function $\epsilon : E \rightarrow 2^{\mathcal{E}}$

UC San Diego
Boolean Lab

Learning Tasks on Graphs

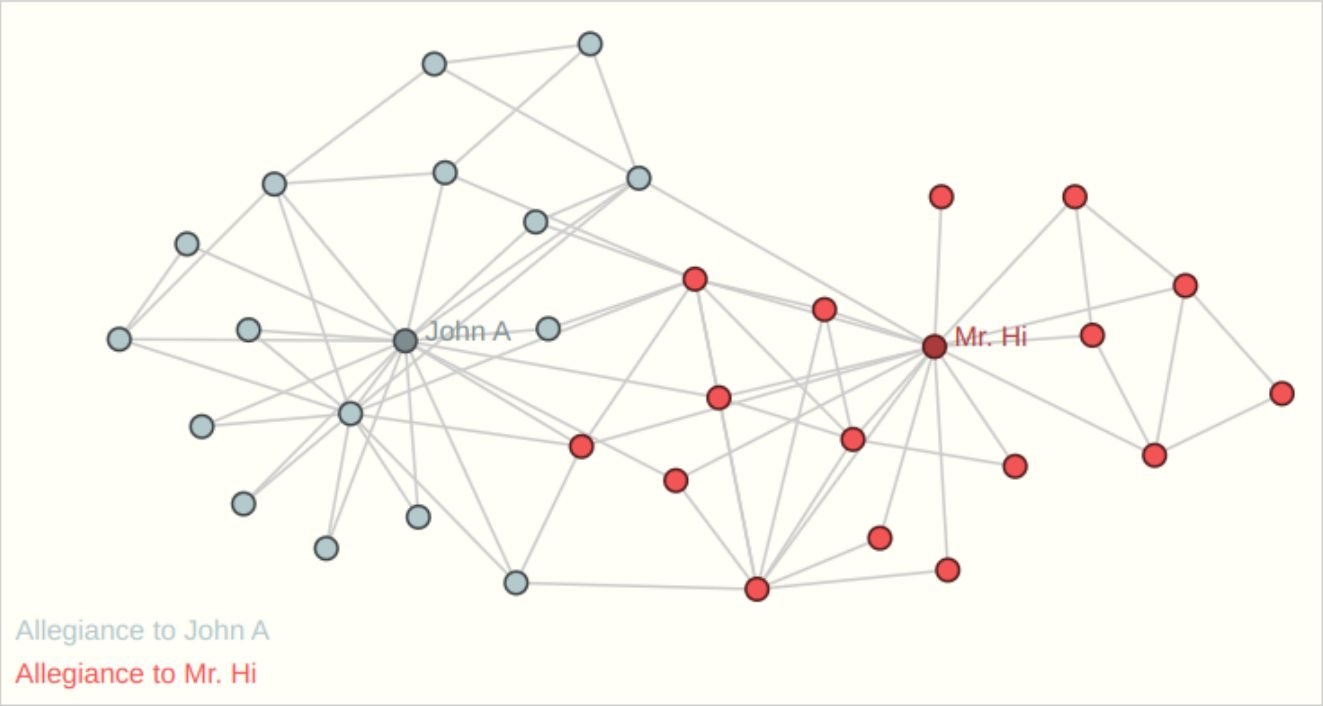
- Graph-level task: Predicting a property of the whole graph



- Predicting *efficacy* of a drug molecule

Learning Tasks on Graphs

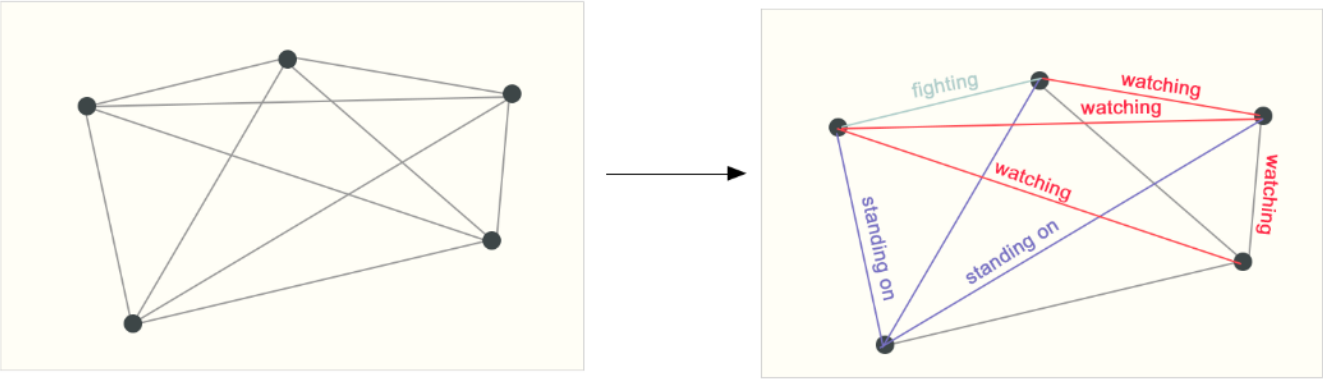
- Node-level task: Predicting a property of a node



- Predicting whether an aminoacid will be in an *interacting* chain

Learning Tasks on Graphs

- Edge-level task: Predicting a property of an edge (node-node pair)



  - Predicting what drug *treats* what disease

Pre-GNN Approaches to ML on Graphs

Node Statistics:

- Node degree: Measures the influence of a node on its neighbours
- Node centrality: Measures the importance of a node in the graph
- Clustering coefficient: Measures the diffences in properties of two nodes
- etc.

UC San Diego
Boolean Lab

Pre-GNN Approaches to ML on Graphs

Kernel methods:

- Bag of nodes: Aggregates node-level statisticss
- The Weisfeiler-Lehman (WL) kernel: Computes and compares two graph representations

UC San Diego
Boolean Lab

Pre-GNN Approaches to ML on Graphs

WL Kernel: Iterative neighbourhood aggregation

1. Assign an initial labelling to each node (usually, its degree):

$$\ell^{(0)}(v) = d_v \ \forall v \in V$$

2. Iteratively assign a new labelling to each node by hashing the multiset of the current labels within the node's neighborhood:

$$\ell^{(i)} = \mathrm{HASH}(\{\{\ell^{(i-1)}(u)\} \ \forall u \in \mathcal{N}(v)\})$$

3. After running K iterations of re-labelling (i.e., Step 2), we now have a label $\ell^{(K)}(v)$ for each node that summarizes the structure of its K-hop neighbourhood.

Graph representation: Aggregate the updated node labellings

UC San Diego
Boolean Lab

Pre-GNN Approaches to ML on Graphs

Limitations of Pre-GNN (or traditional) approaches:

- Require careful, hand-engineered statistics and measures
- Designing these features is time-consuming
- Hand-engineered features are not very flexible: Maynot adapt through the learning process

UC San Diego
Boolean Lab

ML on Graphs using GNNs

GNN is a special kind of deep neural network (DNN) where

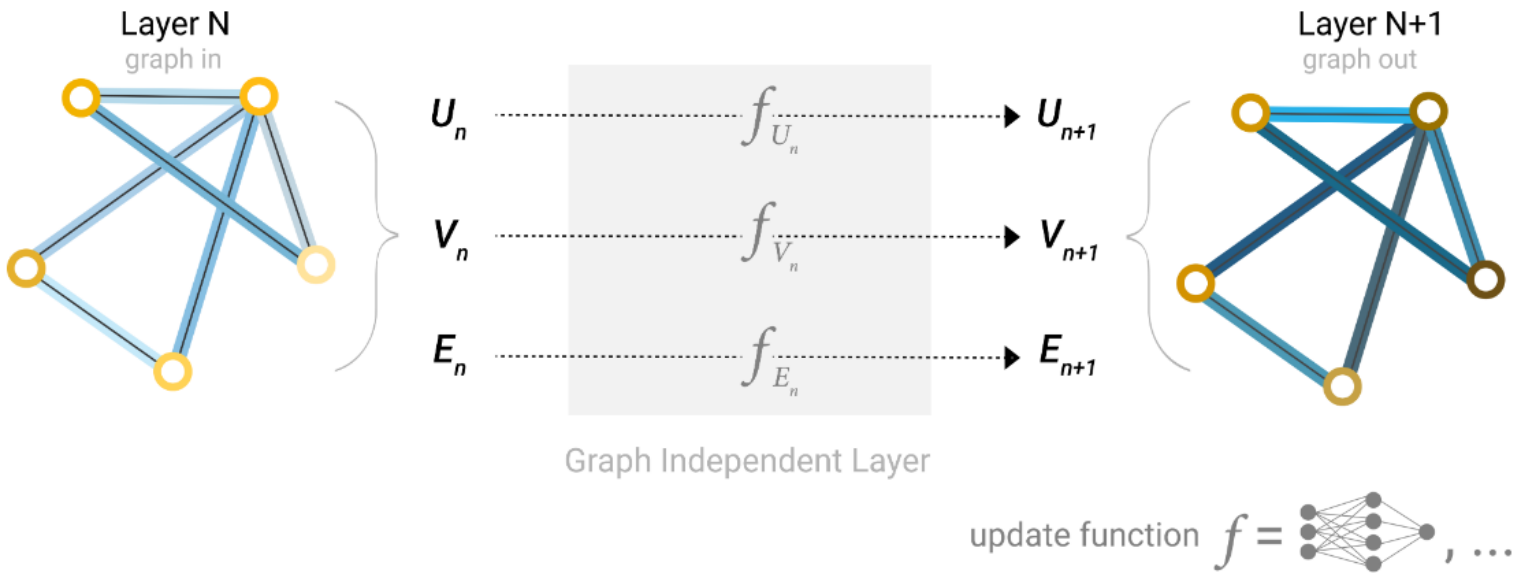
- The input is a labelled graph;
- Its output is *permutation equivariant**;
- It follows a "Graph-in, Graph-out" principle.

*Output of the GNN is unaffected by permutation (ordering) of the vertices or their labellings.

UC San Diego
Boolean Lab

ML on Graphs using GNNs

A simple GNN convolution step (also, called a 'layer') can be visualised this way:



Notice that the change is only to the descriptions associated with the graph (nodes, edges and graph), and not the structure of the graph.

ML on Graphs using GNNs

We can describe this using a function:

Relabel function

A Relabel function is defined as

$$Relabel : \mathcal{G} \rightarrow \mathcal{G}'.$$

That is,

$$Relabel((V, E, \sigma, \psi, \epsilon)) = (V, E, \sigma, \psi', \epsilon')$$

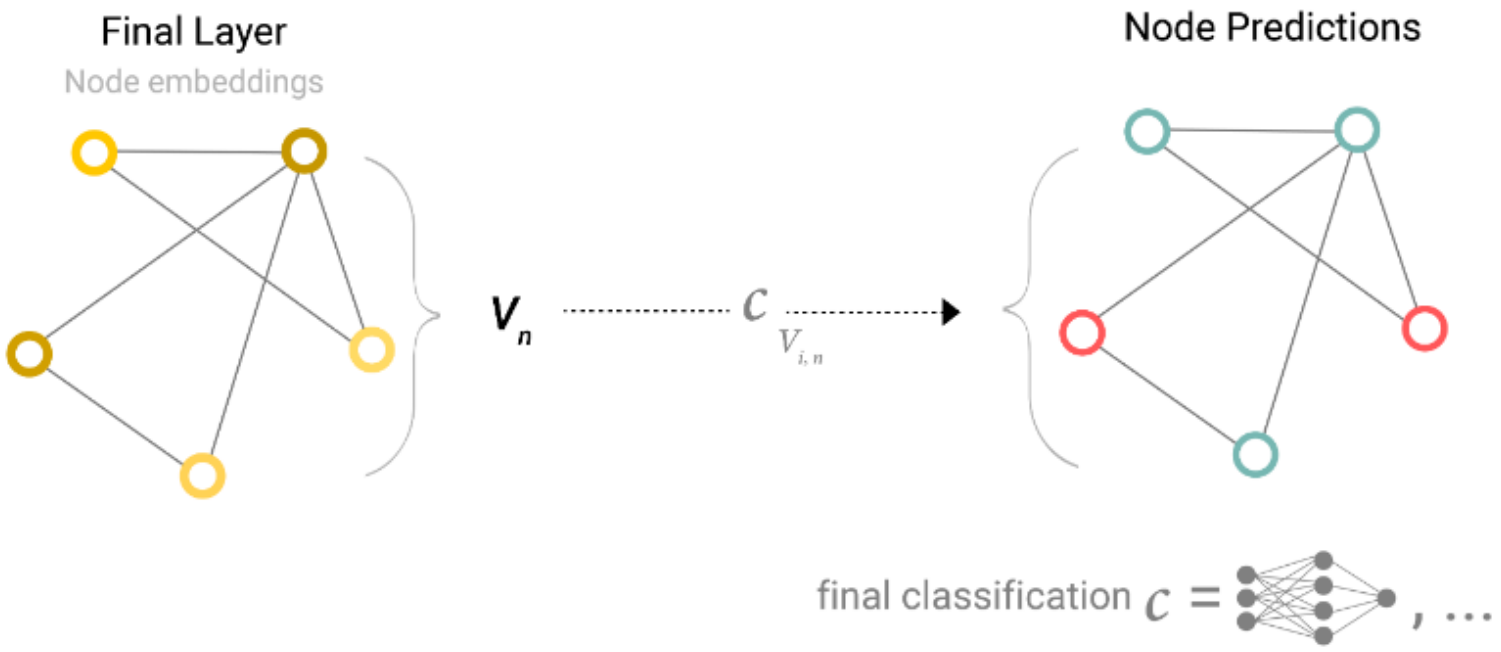
- *Relabel* is implemented using AGGREGATE-COMBINE procedures.
  - Graph convolution and pooling operations

UC San Diego
Boolean Lab

ML on Graphs using GNNs

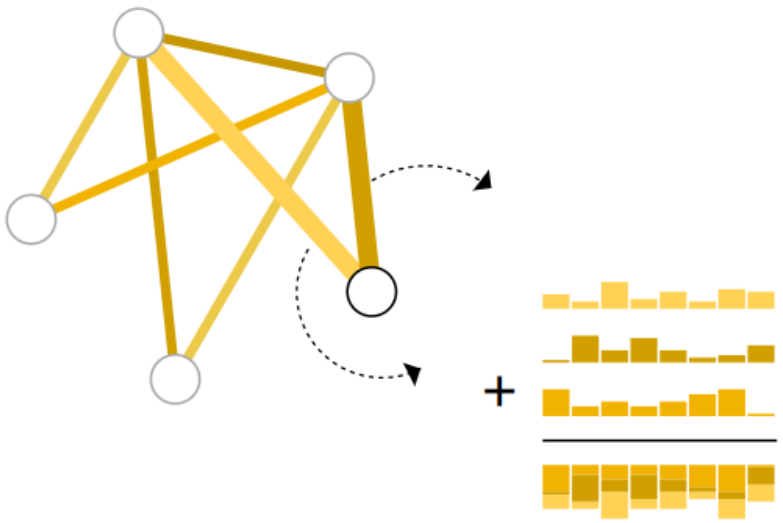
Now, how do we deal with various learning tasks: Prediction for nodes, edges, graphs?

Node-level prediction: Apply a classifier on the node attribute.



ML on Graphs using GNNs

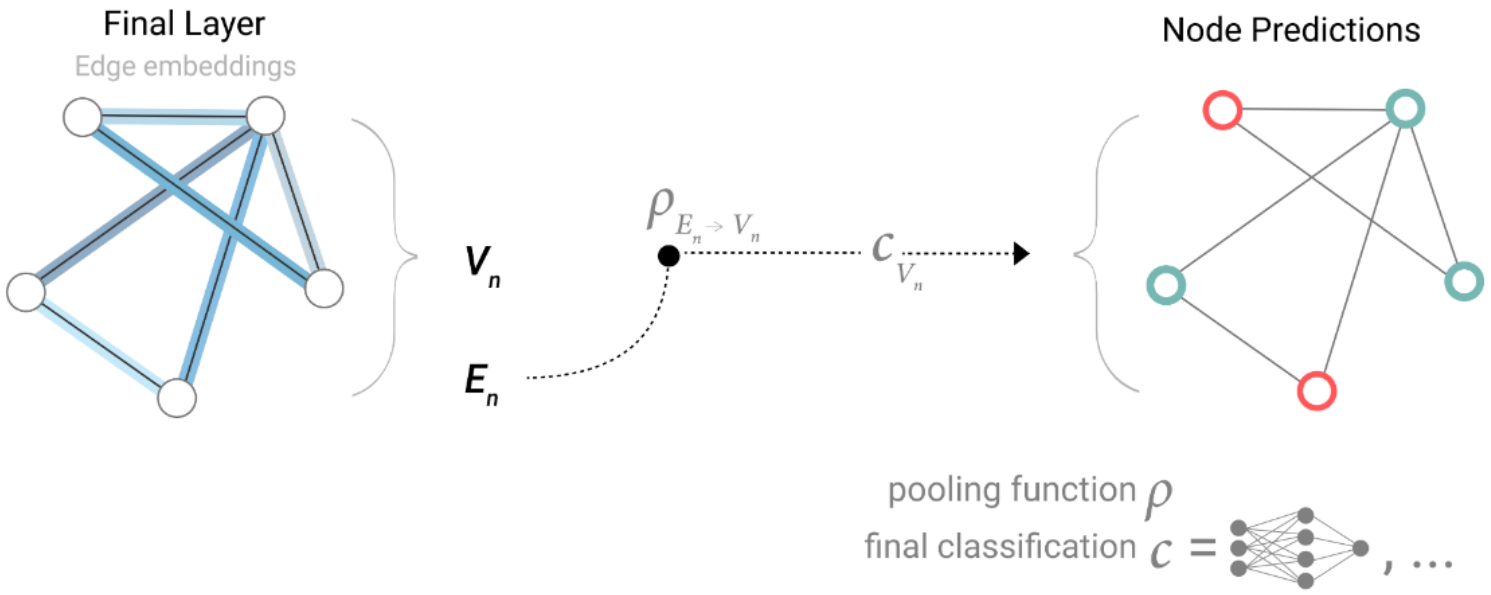
Things can be quite complicated: What if we don't have node-level information, just the edge-information and we want to do node-level prediction?



Aggregate information
from adjacent edges

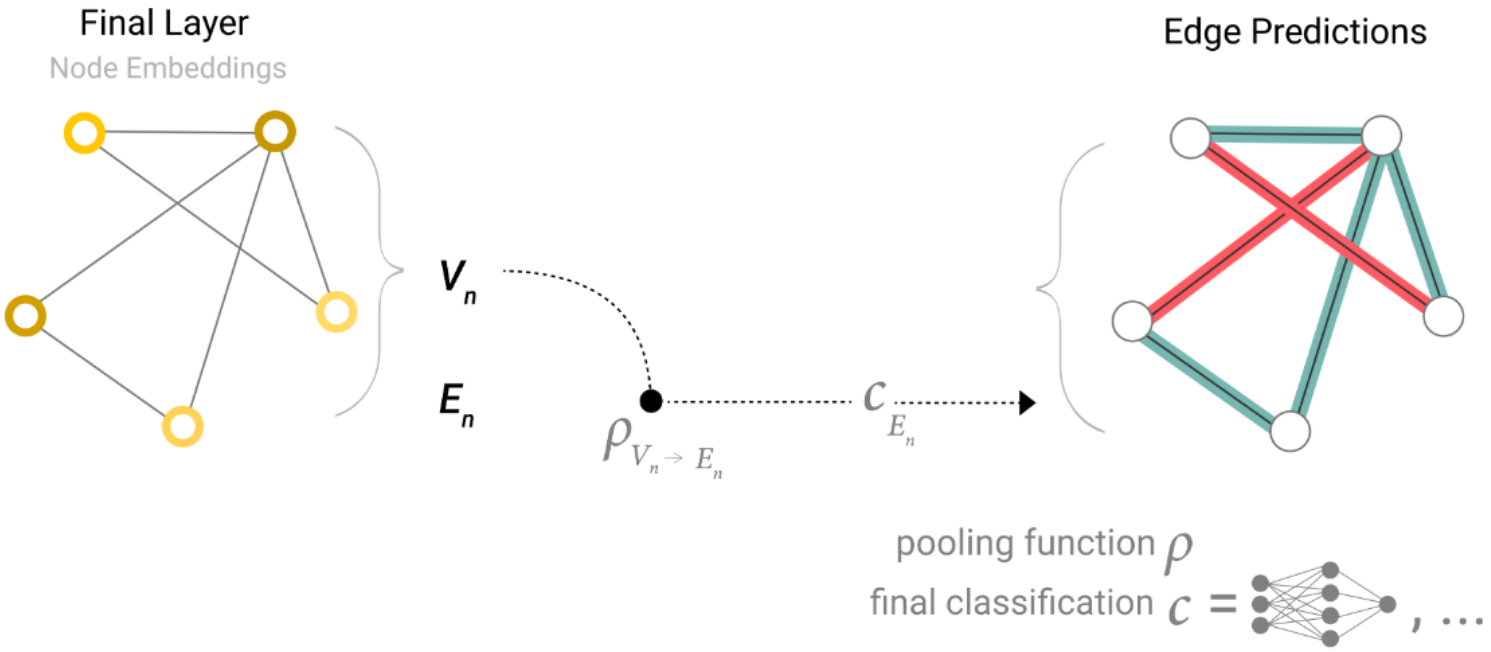
ML on Graphs using GNNs

Pooling: collect information from edges and give them to nodes for prediction.



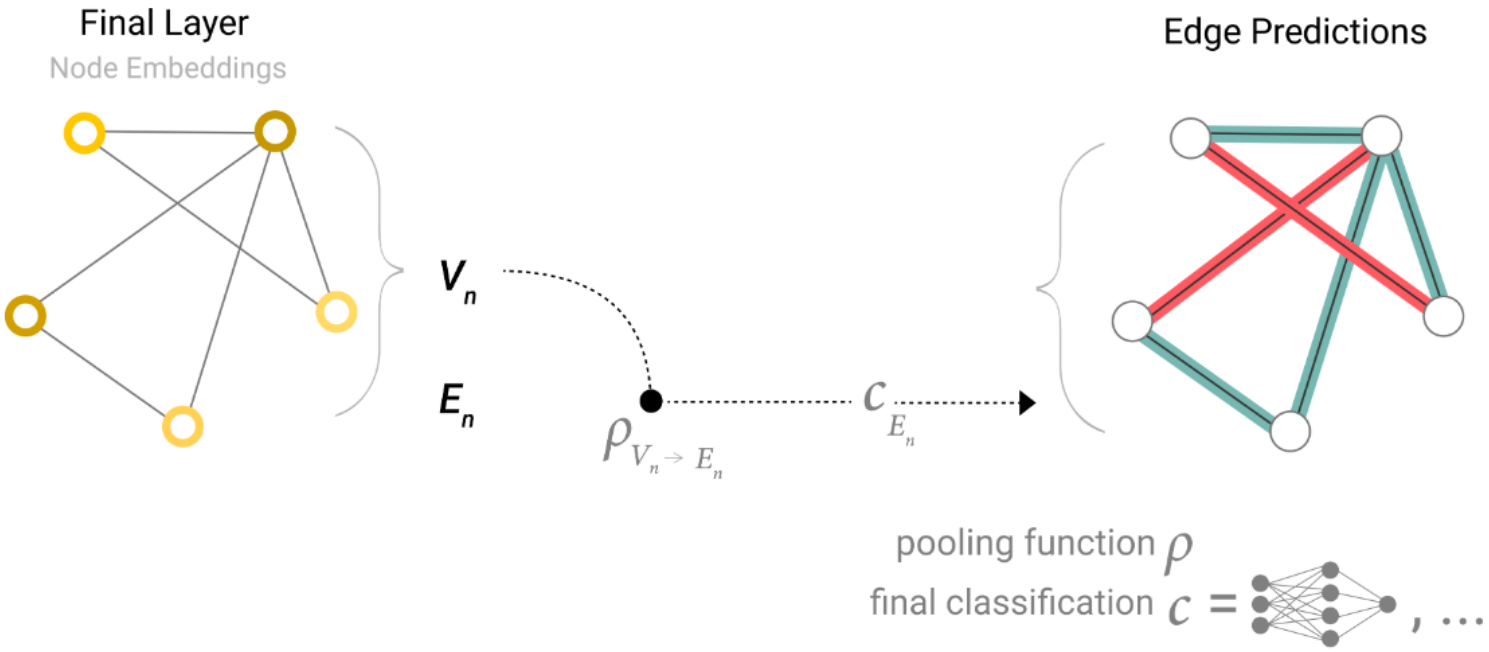
ML on Graphs using GNNs

Similarly, we can use only the node-level information and use it for edge-level prediction.



ML on Graphs using GNNs

Further, we can use only the node-level information and use it for graph-level prediction.



Global average pooling: Gather all available node information together and aggregate them and use it for prediction by a classifier*.

*A differentiable function (an MLP, mostly).

ML on Graphs using GNNs

This process of obtaining a graph-level representation can be described using a *vectorise* function:

Vec function

A vectorise function is defined as

$$Vec : \mathcal{G}' \rightarrow \mathbb{R}^d$$

- This function implemented by a READOUT procedure.
  - Global or hierarchical pooling operations

UC San Diego
Boolean Lab

ML on Graphs using GNNs

Then, a graph-level prediction can be described as:

NN function

A NN function is defined as

$$NN : \mathbb{R}^d \rightarrow \mathcal{Y}$$

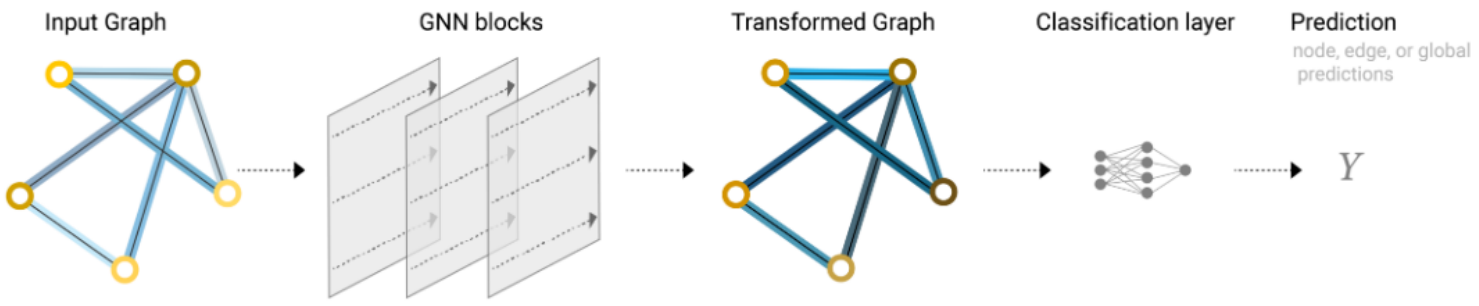
where $\mathcal{Y}$ denotes a set of class-labels or it is simply $\mathbb{R}$ for regression.

- This function is implemented using a multilayered perceptron (MLP).

UC San Diego
Boolean Lab

ML on Graphs using GNNs

That's it! We are now familiar with the basic principles of GNNs.



Somethings to always remember:

- Representations are *tensors*.
- GNN operations are on those tensors.

ML on Graphs using GNNs

GCN [Kipf and Welling, ICLR 2017]:

- Weighted aggregation of neighour node information

$$\mathbf{X}^{(\ell)} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{D}^{-\frac{1}{2}}\mathbf{X}^{(\ell-1)}\mathbf{W}^{(\ell)}\right)$$

UC San Diego
Boolean Lab

ML on Graphs using GNNs

- For a single node in the graph, v_i the equation is simply,

$$\mathbf{x}_{im} = \sigma\left(\frac{1}{d_i}\tilde{a}_{ij}x_{jk}w_{mk}\right)$$

Here:

- j: index for the neighbour node
- k: index for the node feature
- m: index for the *output* node feature
- w_{mk}: trainable weight parameter
- $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}_N$: Adjacency matrix with added self-connections (N is the number of nodes in the graph)

UC San Diego
Boolean Lab

ML on Graphs using GNNs

- Some things to note:
  - The equation, as before, communicates with the immediate neighbours, only once.
    Meaning: One layer of computation $\leftrightarrow$ 1-hop communication
  - In many real-world problems, the information from distant neighbours may be useful.
    Meaning: Mulitlayer computation $\leftrightarrow$ Multihop communication
  - Stack multiple layers of GNNs with different lengths of output node feature vectors.

UC San Diego
Boolean Lab

ML on Graphs using GNNs

- There are now numerous implementations of GNN layers. Some popular ones are:
  - GCN [Kipf and Welling, ICLR 2017]
  - GraphSAGE [Hamilton et al., NIPS 2017]
  - GAT [Veličković et al., ICLR 2018]
  - k-GNN [Morris et al., AAAI 2019]
  - ARMA [Bianchi et al., IEEE TPAMI 2021]
  - ...
- Similarly, there are different implementations of pooling layers.
- A comprehensive survey on various GNN layers can be found here: [Wu et al., IEEE TNNLS 2020].

UC San Diego
Boolean Lab

ML on Graphs using GNNs

We have prepared a notebook for you to play with:

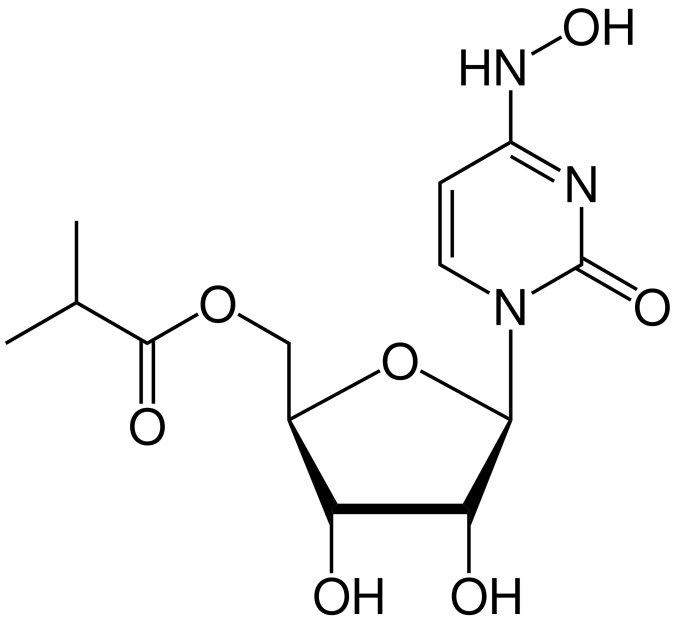
GNN Tutorial

Contributors:

- Shreyas Bhat
- Soham Chitnis

UC San Diego
Boolean Lab

Application of GNNs in Science



Q. Is this molecule effective against Covid-19 virus (*pos*)?

Application of GNNs in Science

GNNs have been successfully used for solving many real-world problems, including the one being discussed here.

But, there have been none or only a few attempts at incorporating domain-knowledge into GNNs.

UC San Diego
Boolean Lab

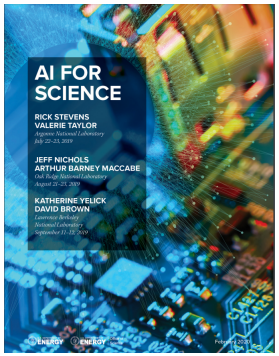
Application of GNNs in Science

Why is the inclusion of domain-knowledge into GNNs important?

- The incorporation of domain-knowledge is the first of the 3 Grand Challenges in developing AI systems:

  ``ML and AI are generally domain-agnostic.... Off-the-shelf practice treats [each of these] datasets in the same manner and ignores domain knowledge...

  Improving our ability to systematically incorporate diverse forms of domain knowledge can impact every aspect of AI ...''



(AI for Sc. Report, 2020)

Read the AI for Science Report.

UC San Diego Boolean Lab

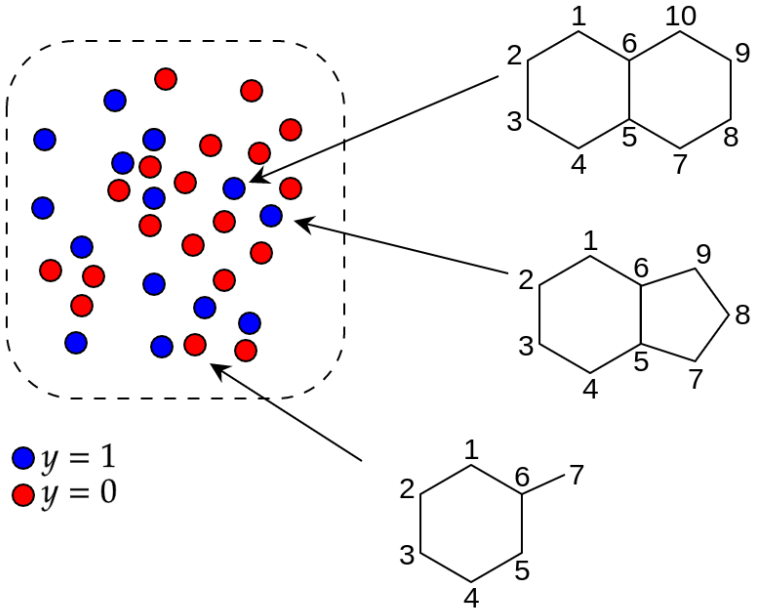
Application of GNNs in Science

Our interest is in real-world scientific problems where:

- Data are relational, naturallty represented as graphs;
- Domain-knowledge is available.

UC San Diego
Boolean Lab

Application 1: The discriminative problem



Graph space: Each instance (● or ●) is a graph.

Application of GNNs in Science

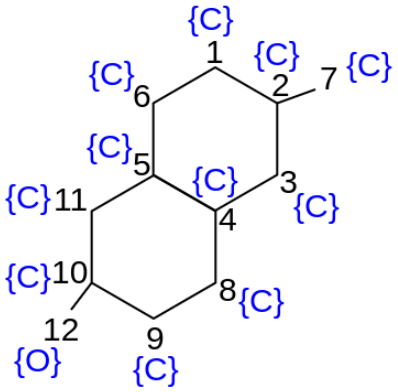
Let's assume we are dealing with problems in drug-discovery.

We have some domain-knowledge: Knowledge about various functional groups and various other structures.

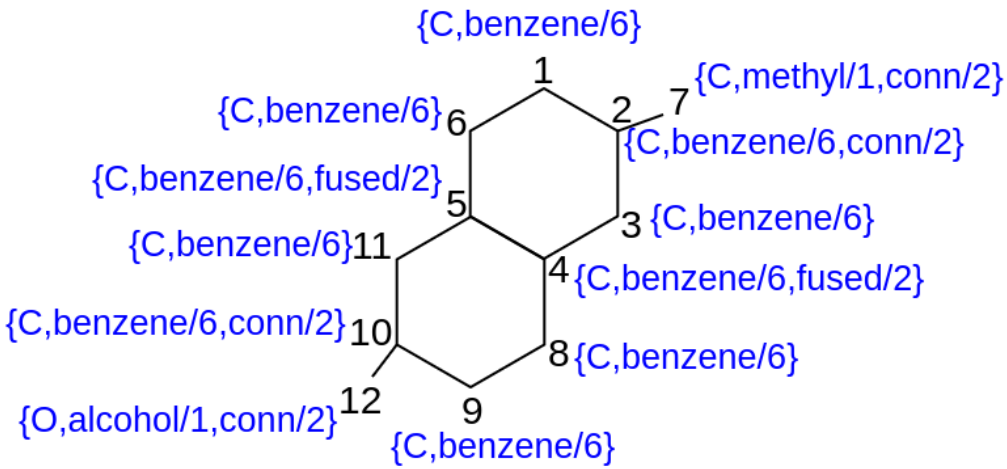
UC San Diego
Boolean Lab

Application of GNNs in Science

Simplified inclusion of domain-knowledge into GNNs:



(Before inclusion)



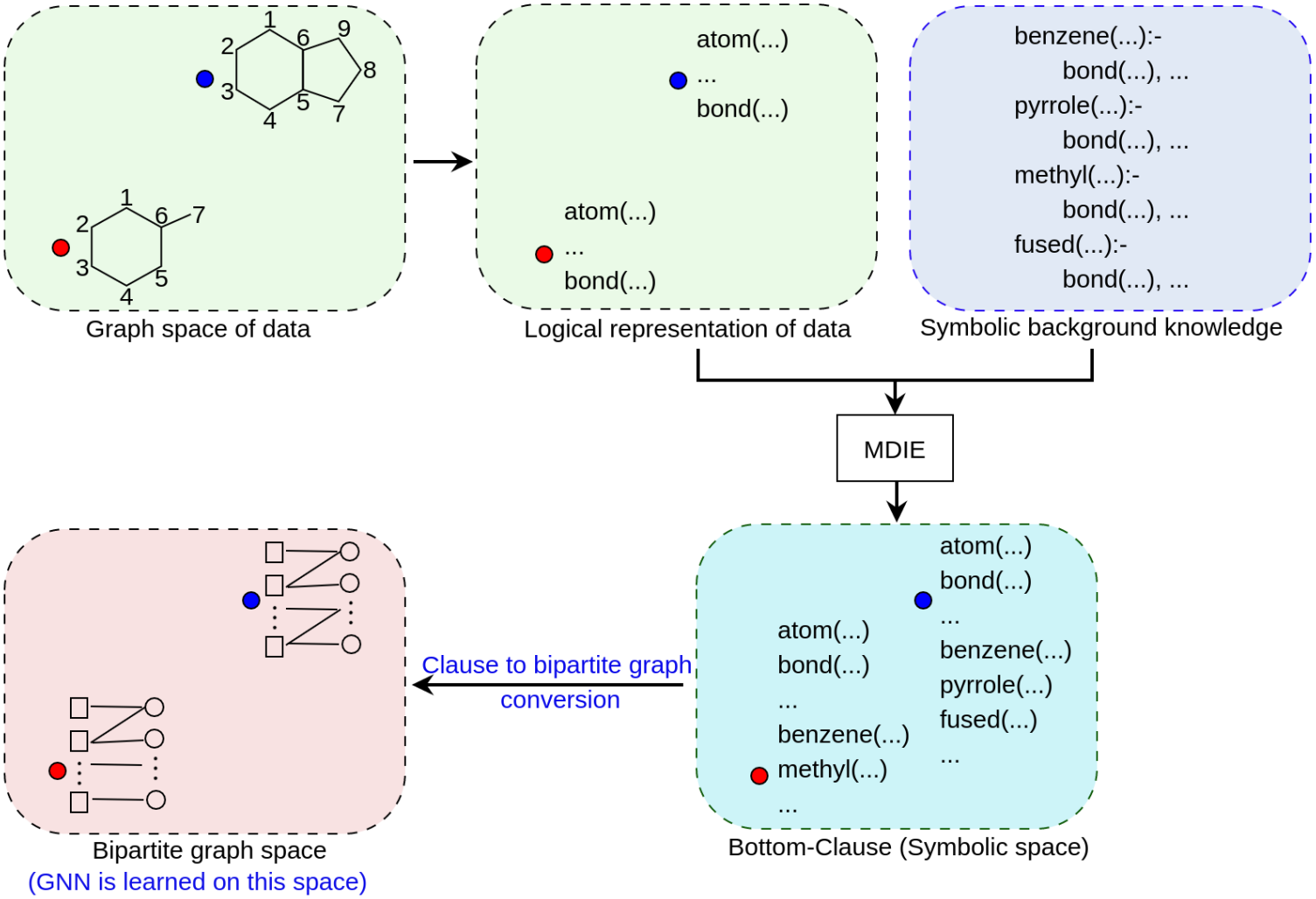
(After inclusion)

Based on: Vertex-Enrichment [Dash et al., MLJ 2021].

UC San Diego
Boolean Lab

Application of GNNs in Science

Complete inclusion of domain-knowledge into GNNs:



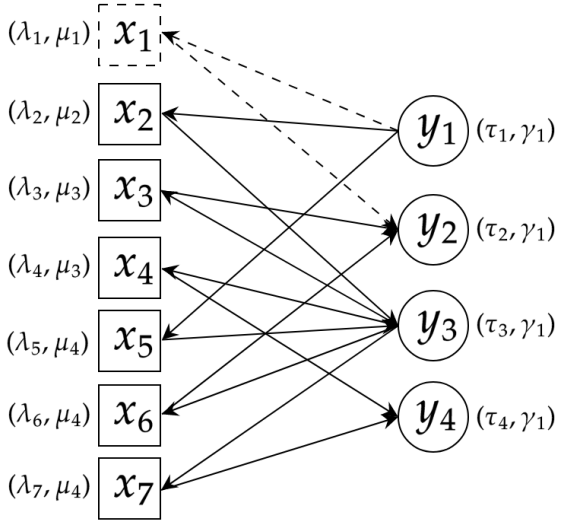
UC San Diego
Boolean Lab

Application of GNNs in Science

An example, on a simple family relation:

$$gparent(henry, john) \leftarrow$$
$$father(henry, jane),$$
$$mother(jane, john)$$

(Before inclusion)



(After inclusion)

Background knowledge:

$$parent(X, Y) \leftarrow father(X, Y)$$
$$parent(X, Y) \leftarrow mother(X, Y)$$

$$mother(jane, alice) \leftarrow$$

Based on: [Dash et al., MLJ 2022].

UC San Diego
Boolean Lab

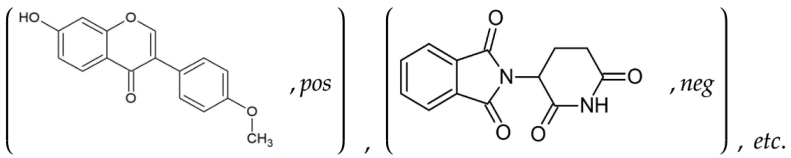
Large-scale Empirical Evaluation of GNNs

Datasets. NCI-50 datasets (chemical compounds and their activities)

- Number of datasets: 73 (approx. $220,000$ instances)
- A summary:

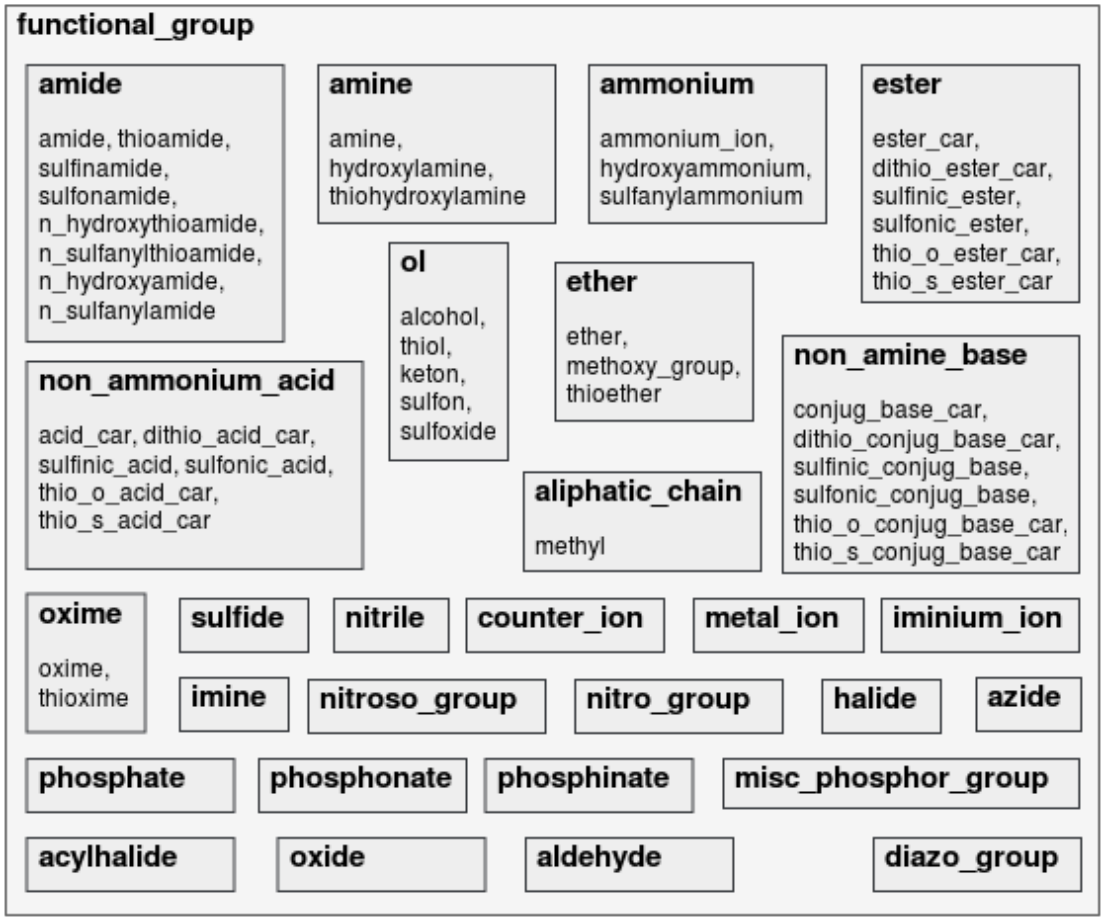
Avg. # of instances	Avg. # of atoms per instance	Avg. # of bonds per instance	% of positives
3032	24	51	0.4–0.9

- Each compound has an associated anti-cancer activity (positive or negative).



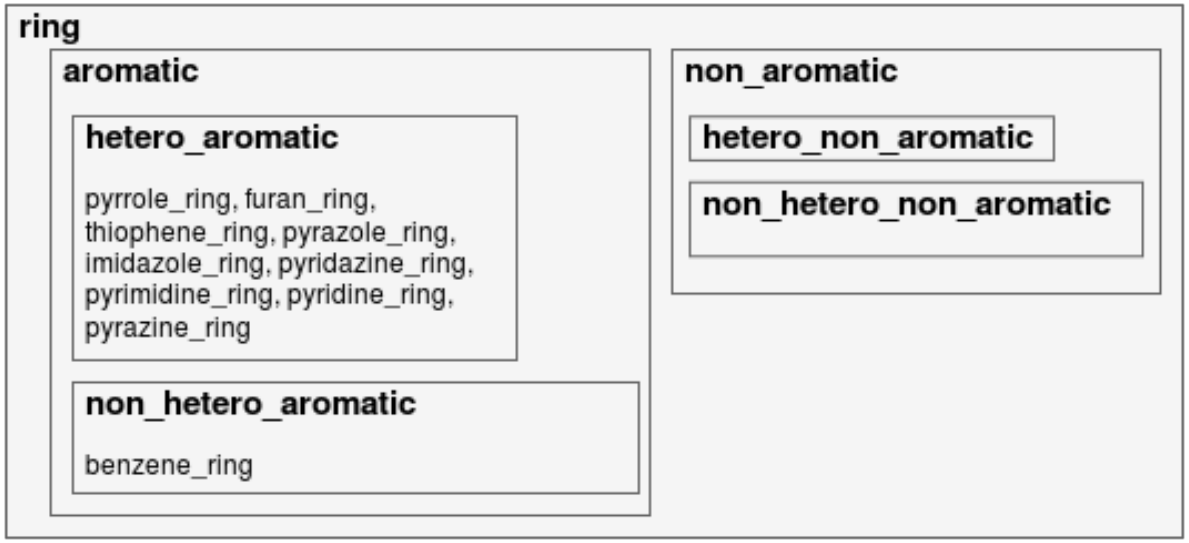
Application of GNNs in Science

Background Knowledge. Facts and definitions of chemical structures



UC San Diego
Boolean Lab

Application of GNNs in Science



In overall, we have about 100 domain-relations.

Application of GNNs in Science

GNNs. 5 different variants (as discussed before, GCN, GAT, etc.)

UC San Diego
Boolean Lab

Evaluation of VEGNNs:

- GNNs with domain-knowledge (VEGNNs) are better than GNNs without domain-knowledge.

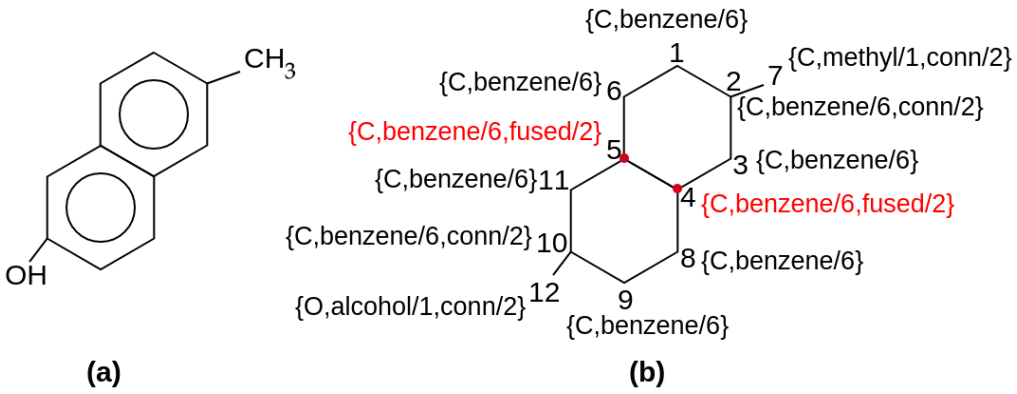
GNN Variant	Accuracy (*VEGNN* vs. *GNN*) Higher/Lower/Equal (p-value)
GNN_1	48/14/11 (< 0.001)
GNN_2	48/19/6 (0.005)
GNN_3	53/11/9 (< 0.001)
GNN_4	54/12/7 (< 0.001)
GNN_5	43/19/11 (0.002)

UC San Diego
Boolean Lab

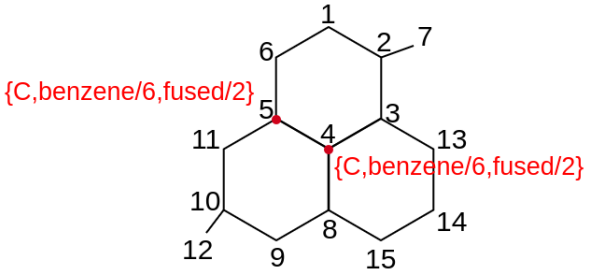
Application of GNNs in Science

But, there are some limitations:

- Vertex-enrichment simplifies domain-knowledge.



Similarly,



Evaluation of BotGNNs:

- GNNs with domain-knowledge (BotGNNs) are better than GNNs without domain-knowledge.

GNN Variant	Accuracy (*BotGNN* vs. *GNN*) Higher/Lower/Equal (p-value)
1	59/5/9 (< 0.001)
2	59/8/6 (< 0.001)
3	61/2/10 (< 0.001)
4	63/1/9 (< 0.001)
5	60/4/9 (< 0.001)

UC San Diego
Boolean Lab

Application of GNNs in Science

also:

- BotGNNs are superior to VEGNNs.

GNN Variant	Accuracy (*BotGNN* vs. *VEGNN*) Higher/Lower/Equal (p-value)
1	54/11/8 (< 0.001)
2	61/9/3 (< 0.001)
3	54/10/9 (< 0.001)
4	55/11/7 (< 0.001)
5	52/9/12 (< 0.001)

UC San Diego
Boolean Lab

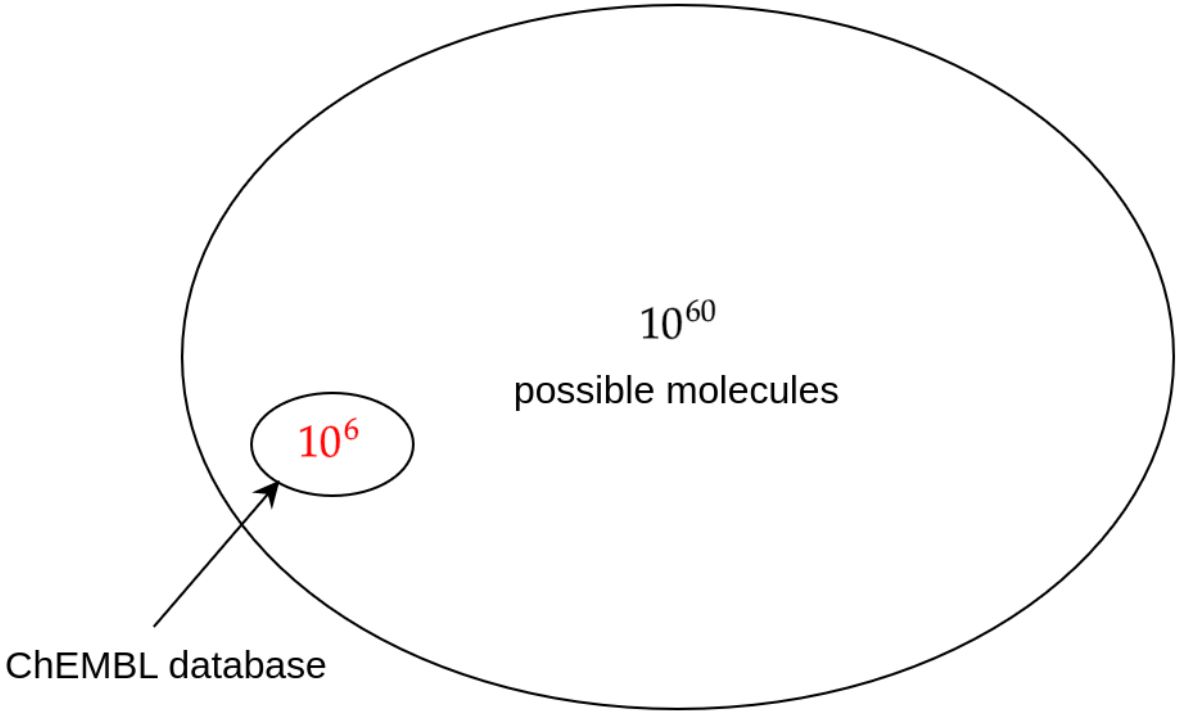
Application 2: The generative problem

- To generate new small molecules which could act as inhibitors of a biological target (e.g. JAK2 protein).
- There is limited prior information on the target-specific inhibitors.
- We want to investigate whether domain-knowledge can assist in generating such molecules.

UC San Diego
Boolean Lab

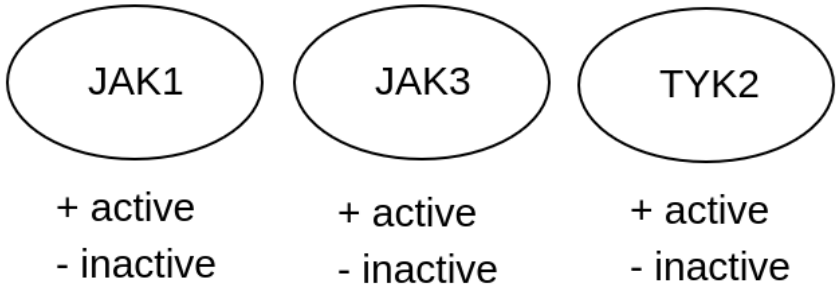
Application of GNNs in Science

It is a hard problem: Searching the space of molecules



Application of GNNs in Science

However, we have some information of some of the inhibitors for some related proteins:

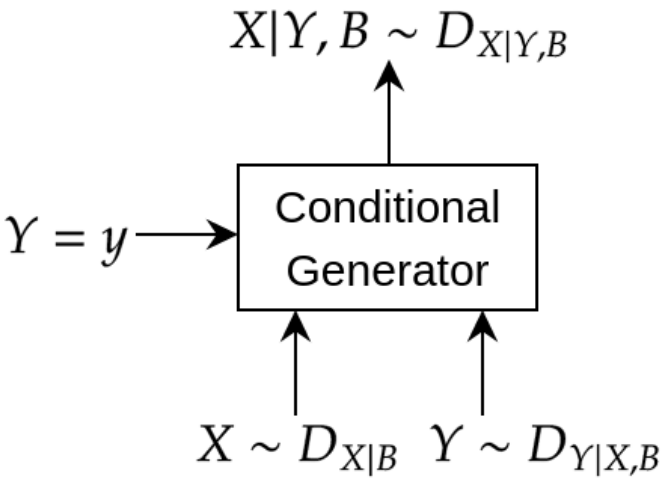


(JAK-homologous proteins)

Application of GNNs in Science

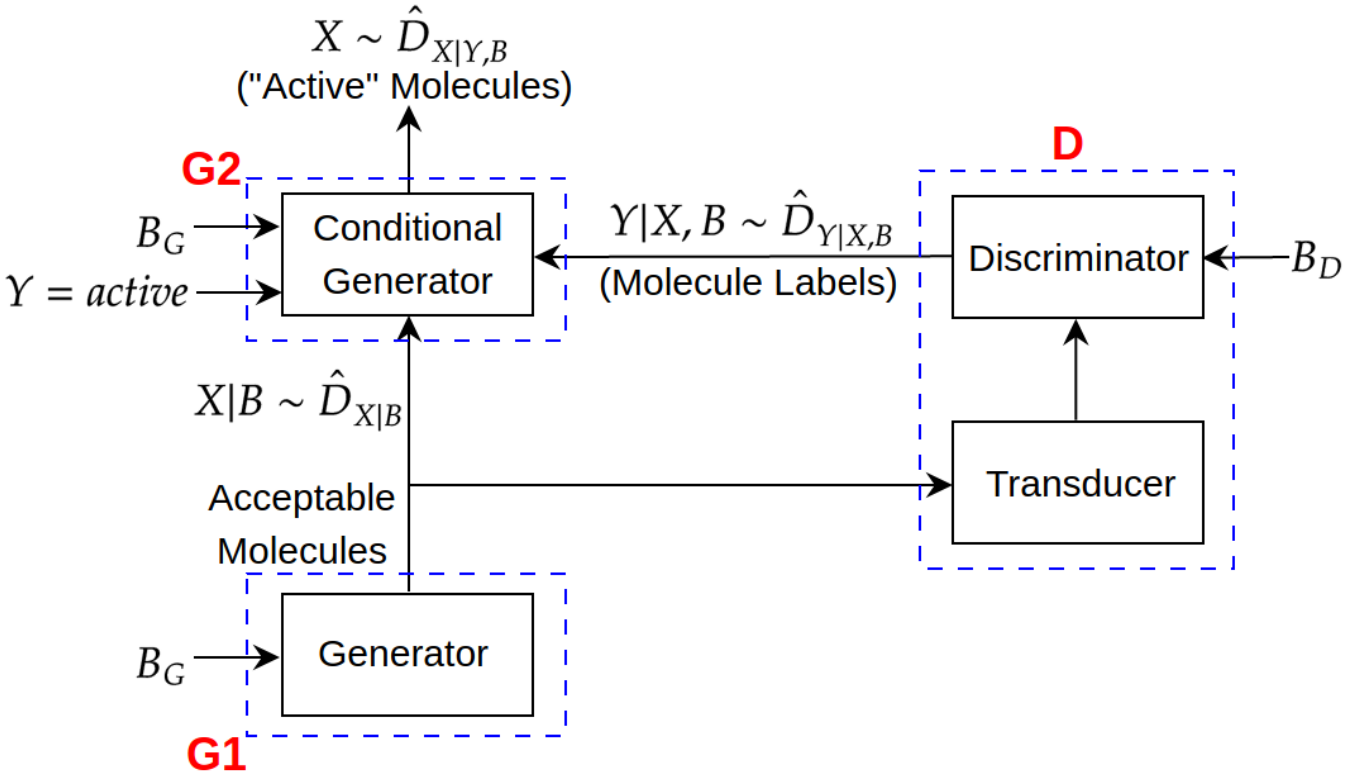
The Idea.

- Molecules and their activities are instances of r.v. X and Y (resp.)
- We want to draw instances from the conditional distribution $D_{X|Y,B}$



(Conditional generation of data)

System Design. Approximating the distributions



Empirical Evaluation of the System

Data.

- Δ: CHEMBL database (1.9M unlabelled SMILES)
- JAK2-Homologues: 4300 labelled SMILES
- JAK2 data: 4100 labelled SMILES (for the proxy model)

Background Knowledge.

- B_G: Constraints on bulk-molecular properties from the literature
- B_D: Functional groups, rings, fused and connected structures

Generators.

- G_1 and G_2: LSTM-based VAEs
- D: BotGNN

UC San Diego
Boolean Lab

Evaluation: Quantitative

As compared to the state-of-the-art approach:

- Our system generates significantly higher proportion of *active* molecules that are *active* for JAK2 inhibition.
- Our system generates significantly higher proportion of molecules that are *similar* to JAK2 inhibitors

Comparison is against [Krishnan et al, JCIM 2021].

UC San Diego
Boolean Lab

Application of GNNs in Science

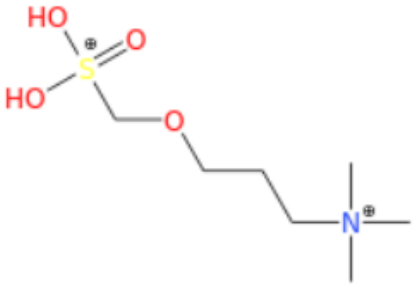
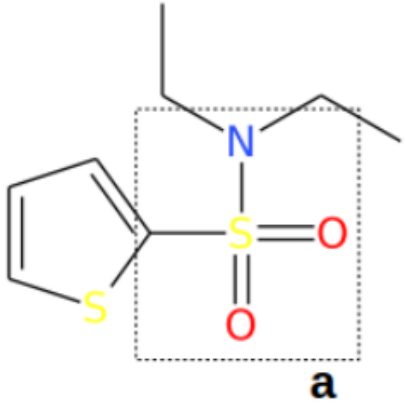
Evaluation: By an expert (computational chemist)

- 10 generates molecules where evaluated:
  - 5 similar to JAK2 inhibitors
  - 5 dissimilar to JAK2 inhibitors
- The expert picked 3 molecules, dissimilar to JAK2 inhibitors that were novel and worth investigating further.

UC San Diego
Boolean Lab

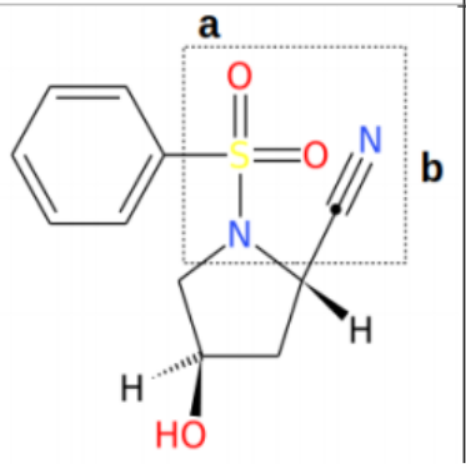
Application of GNNs in Science

- Dissimilar and highly active molecules:

ID	Structure	Descriptors	Assessment
551		$Act = 9.12$ $Sim = 0.15$	This molecule has very low similarity to known JAK2 inhibitors. Also none of the groups specific to JAK2 could be identified by the substructure search. Discard this molecule.
1548		$Act = 9.04$ $Sim = 0.22$	This molecule has very low similarity to known JAK2 inhibitors. Also none of the groups specific to JAK2 could be identified by the substructure search. However, the sulfonamide group commonly found in JAK family inhibitors was found to be present (highlighted)

UC San Diego
Boolean Lab

Application of GNNs in Science

1562		$Act = 9.49$ $Sim = 0.32$	Despite low similarity to existing JAK2 inhibitors, 1562 had one JAK2-selective subgroup and a group common to JAK inhibitors, indicating potential to act as JAK family inhibitor, but the selectivity to JAK2 cannot be confirmed. Possibly interesting new scaffold (highlighted) and worth pursuing further.

Fan mail:

I just saw your preprint ... ***the last molecule seems indeed quite promising.***

From: a researcher at a prominent research lab in Europe

UC San Diego
Boolean Lab

Summary of the Talk

We discussed some

- Concepts
- Implementations and
- Real-world scientific Applications

of Graph Neural Networks or GNNs.

Follow our research on neuro-symbolic learning:

https://github.com/tirtharajdash/NeSy

Reach me at:

🏠 https://tirtharajdash.github.io
@ tdash@[health.]ucsd.edu

UC San Diego
Boolean Lab

Credits

Some contents are based on:

- https://www.cs.mcgill.ca/~wlh/grl_book/
- https://distill.pub/2021/gnn-intro/
- https://distill.pub/2021/understanding-gnns/
- https://dmol.pub/dl/gnn.html

Some contents of the notebooks are based on:

- https://pytorch-geometric.readthedocs.io/

UC San Diego
Boolean Lab