

# Graph Neural Networks for Molecular Structure Learning

Prof. O-Joun Lee

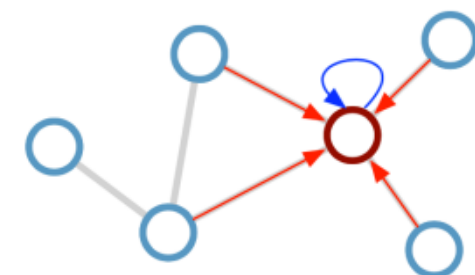
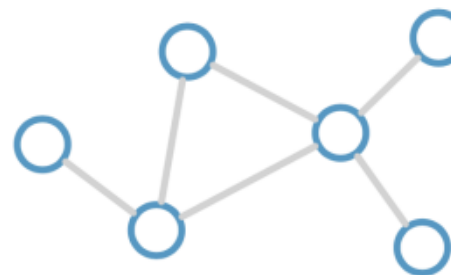
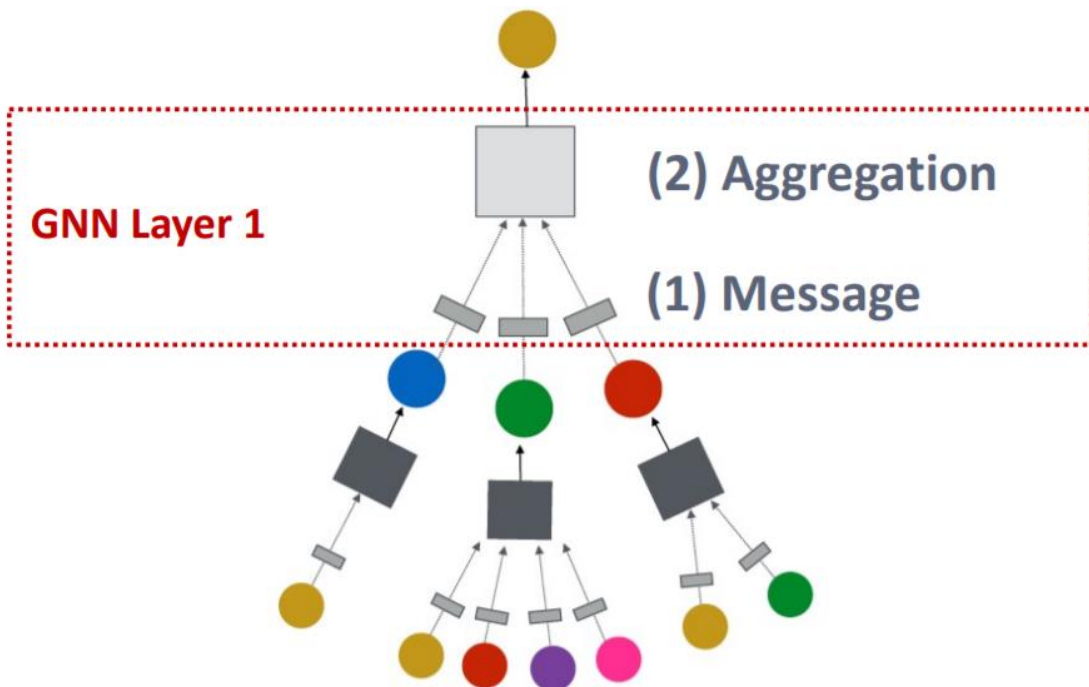
Dept. of Artificial Intelligence,  
The Catholic University of Korea  
*ojlee@catholic.ac.kr*

# Contents



- Introduction to GNNs and Molecules
- Molecules represented as Graphs
- GNNs for molecular learning tasks
  - Molecule Classification task
  - Molecule Regression task
- Equivariant GNNs
- Recent Advances in GNNs on Molecular Learning
- Practice code
  - Read molecules from SMILE strings, and visualize as 2D and 3D graphs
  - Applying GNNs models (GCN, Graphsage, GIN, GAT, GT) on benchmark molecular datasets on classification and regression tasks; and performance comparison.
  - Benchmarks: MoleculeNet

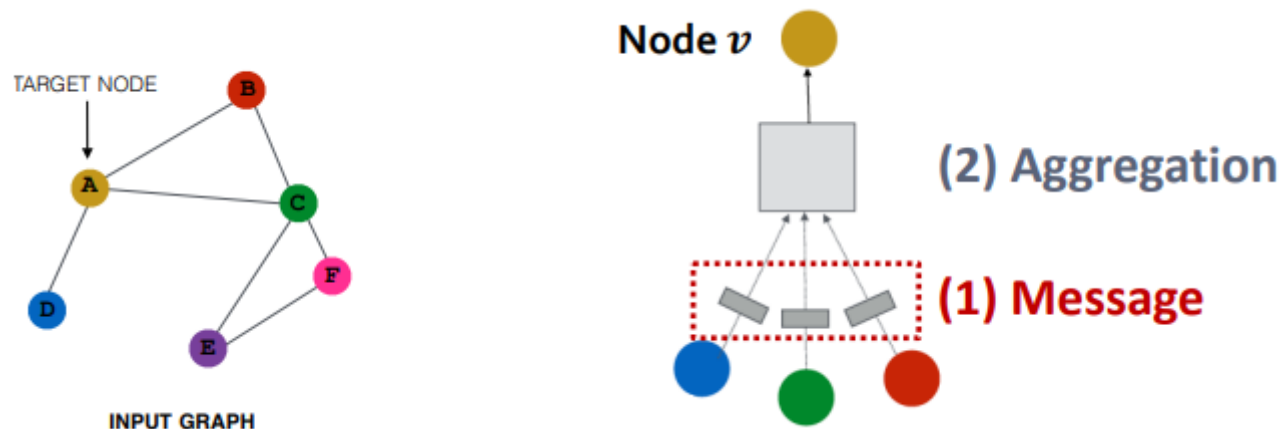
- GNN Layer = **Message** + **Aggregation**
  - Message COMPUTATION
    - how to make each neighborhood node as embedding?
  - Message AGGERGATION
    - how to combine those embeddings?



Update rule: 
$$\mathbf{h}_i^{(l+1)} = \sigma \left( \mathbf{h}_i^{(l)} \mathbf{W}_0^{(l)} + \sum_{j \in \mathcal{N}_i} \frac{1}{c_{ij}} \mathbf{h}_j^{(l)} \mathbf{W}_1^{(l)} \right)$$

- Intuition: Each node will create a message, which will be sent to other nodes later
- Example: A Linear layer  $\mathbf{m}_u^{(l)} = \mathbf{W}^{(l)} \mathbf{h}_u^{(l-1)}$ 
  - Multiply node features with weight matrix  $\mathbf{W}^{(l)}$

**Message function:**  $\mathbf{m}_u^{(l)} = \text{MSG}^{(l)} \left( \mathbf{h}_u^{(l-1)} \right)$



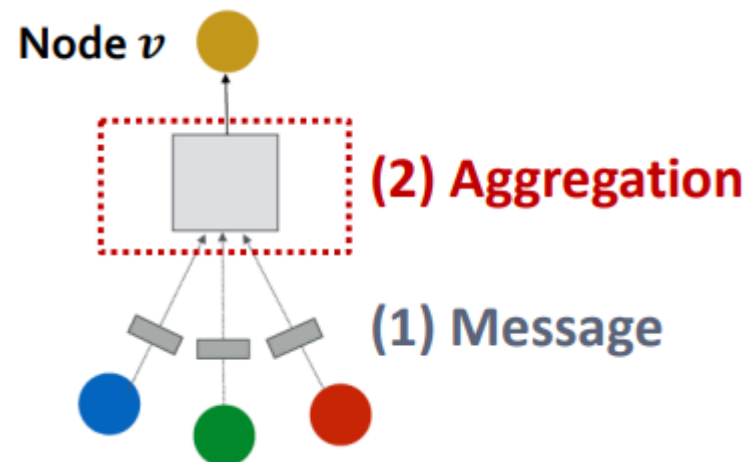
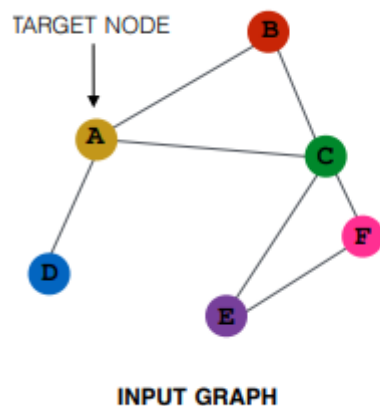


- Intuition: Each node will aggregate the messages from node  $v$ 's neighbors

$$\mathbf{h}_v^{(l)} = \text{AGG}^{(l)} \left( \left\{ \mathbf{m}_u^{(l)}, u \in N(v) \right\} \right)$$

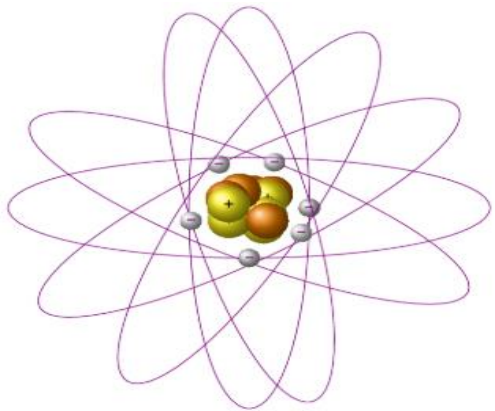
- Example: Sum( $\cdot$ ), Mean( $\cdot$ ) or Max( $\cdot$ ) aggregator

$$\mathbf{h}_v^{(l)} = \text{Sum}(\{\mathbf{m}_u^{(l)}, u \in N(v)\})$$

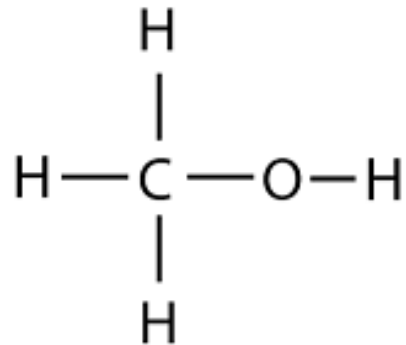


- Atoms are the smallest stable units of matter.
- They can combine to form molecules with complex shapes.
- The atomic components and unique structural shape of a particular molecule determine its chemical functions.

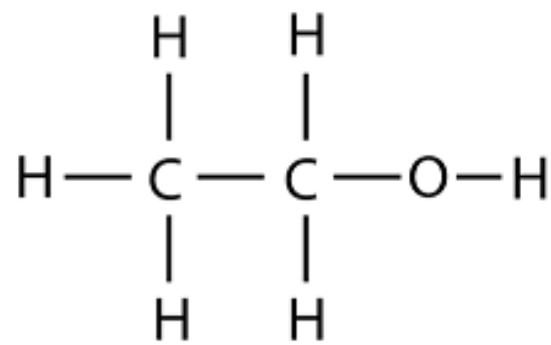
Carbon atom



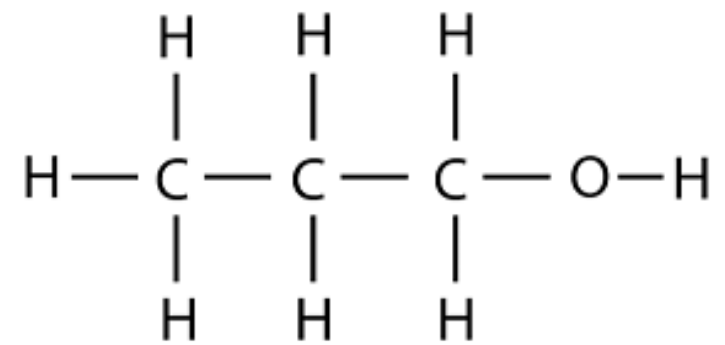
Methanol



Ethanol

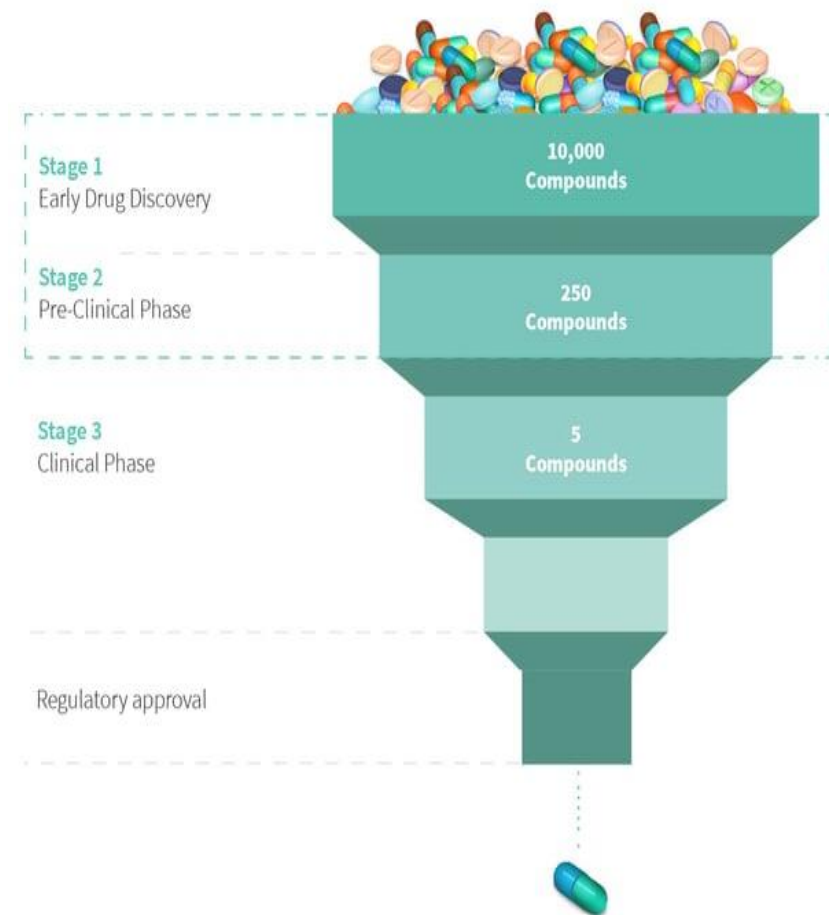


Propanol

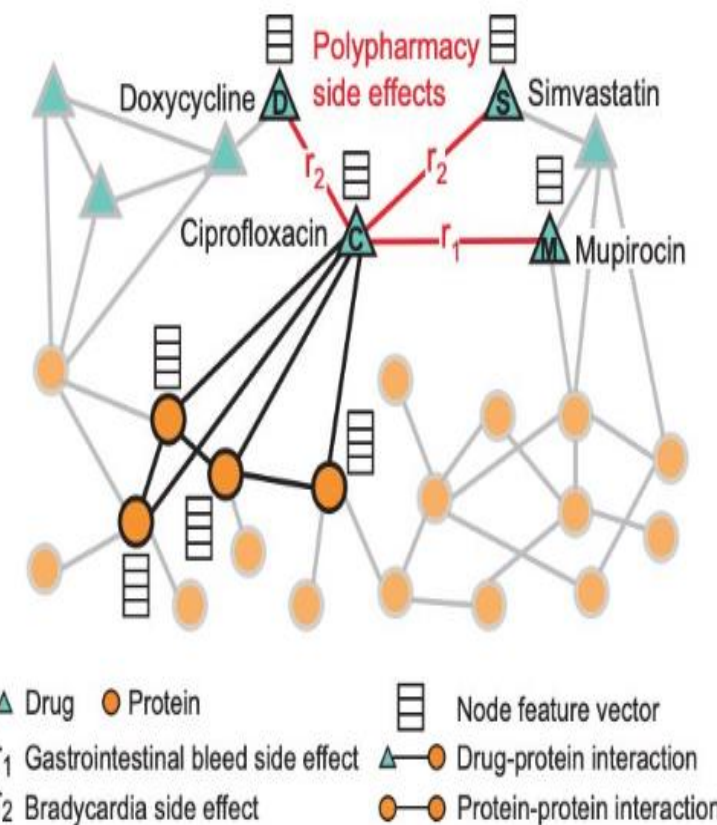


The alcohol group (**-OH group**) is made up of oxygen and hydrogen atoms.

- Application: drug design is very important:
  - Drug discovery is costly, time-expensive
- On average, for every 10,000 structures synthesized during design:
  - 250 will reach animal testing
  - 5 will reach phase I clinical trials
  - Only 1 reach the market place
- **Our target:**
- Use molecular databases and ML-based models to support searching the complex space of candidate.
- Want targeted molecules that optimise for specific properties.

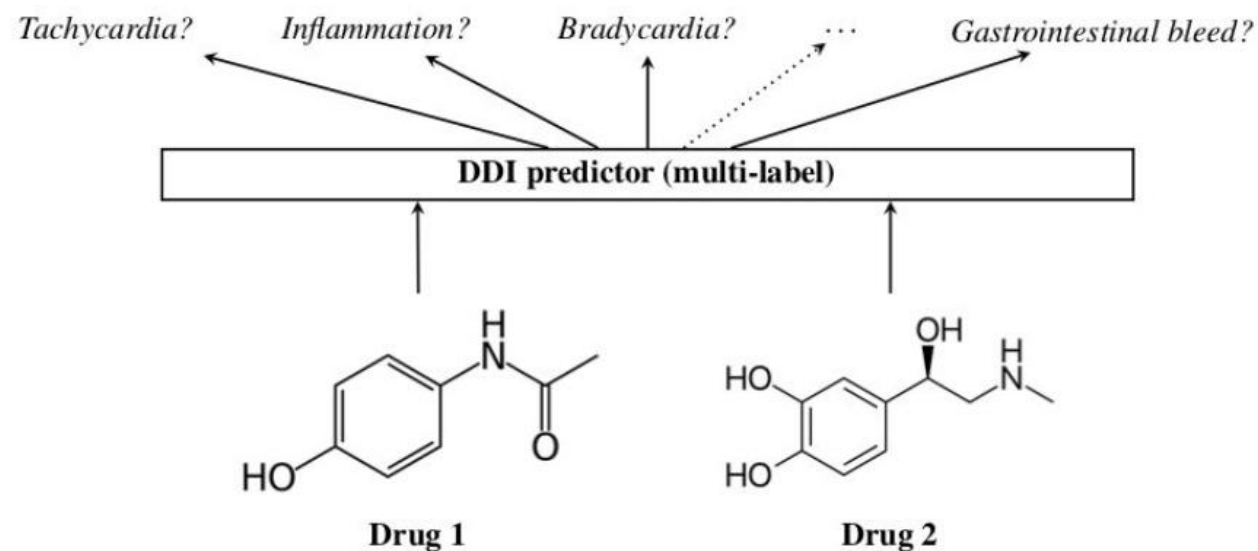
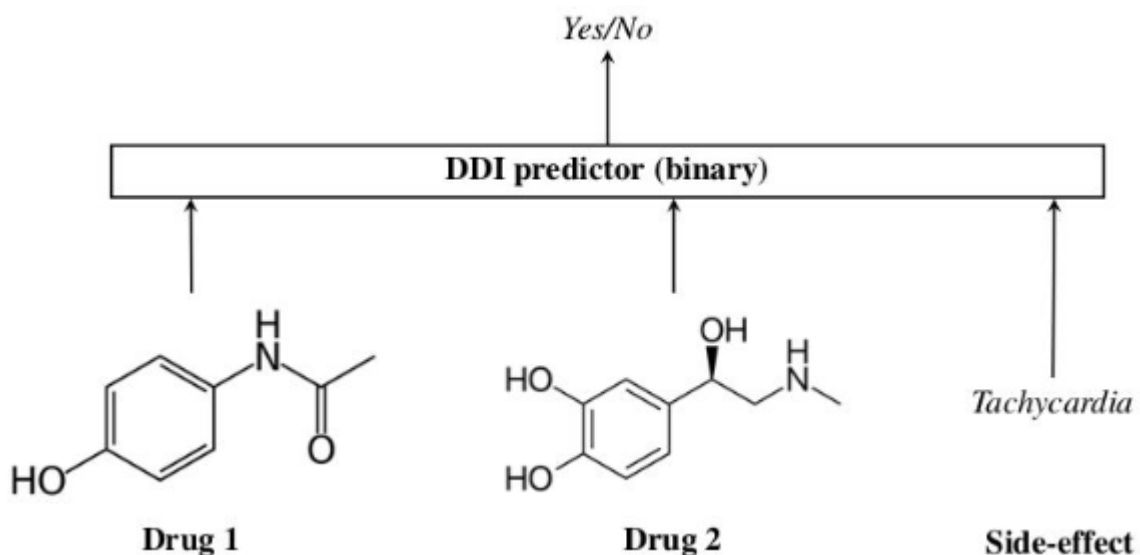


- The process of identifying interactions between drug combinations that may cause adverse side effects.
- Identifying how different drugs interact with each other for Patient safety & Treatment efficacy.
- **Our target:** Can model the complex associations between atoms and functional groups within and between drug molecules.

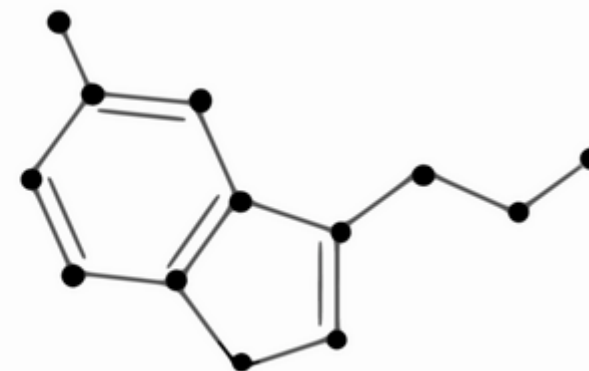
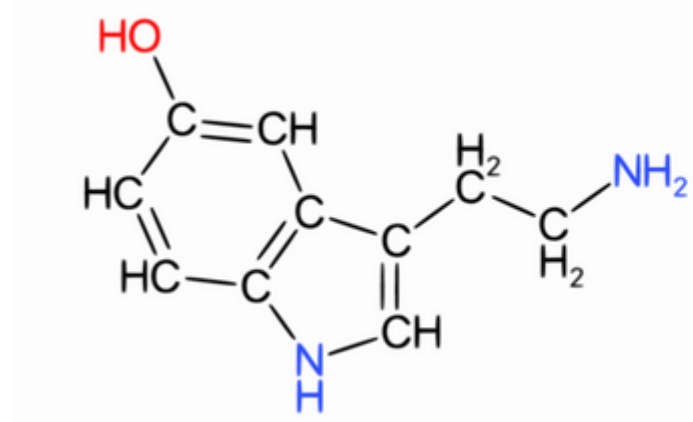
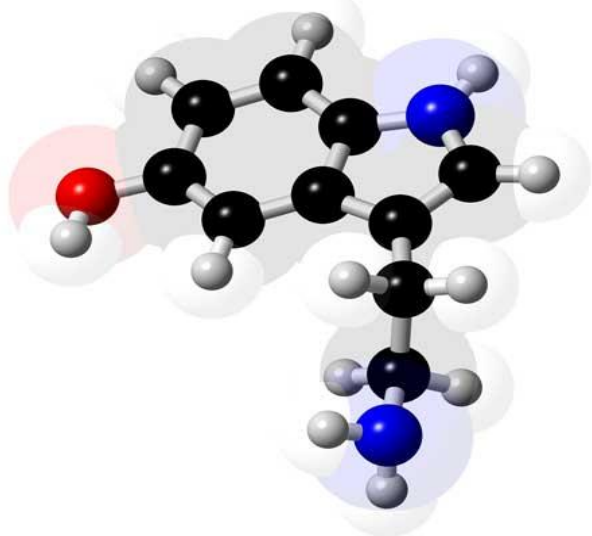




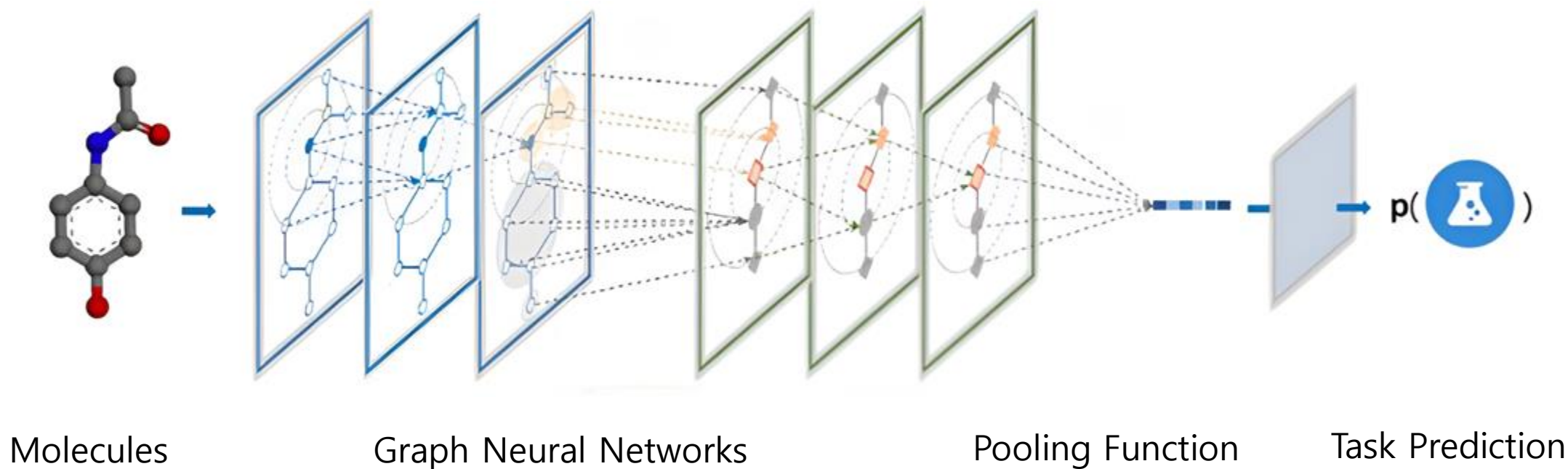
- Drug-drug interaction (DDI) prediction
- We aim to identify unwanted side effects that can occur when multiple drugs are taken together.
- For example:
  - Input: set of molecules w/wo side-effect
  - Output: predict the interaction (binary or multi-label classification)



- A molecular graph is a labeled graph:
  - The nodes and edges represent the atoms and bonds of the molecule, respectively.
- Molecule graph provides direct access to the graph underpinning all molecule objects (atoms, functional groups), allowing seamless integration with existing graph functionality.
  - E.g., Graph representation of the Serotonin ( $C_{10}H_{12}N_2O$ ) molecule with functional groups, i.e.,  $NH_2$ ,  $NH$ , and  $-OH$ .

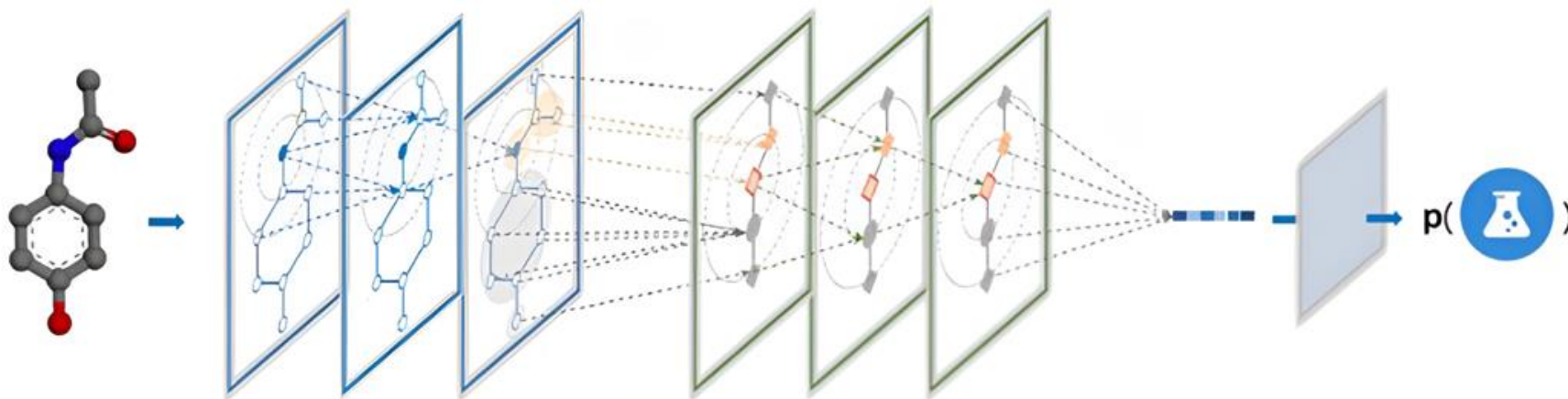


- Inputs:
  - Given a set of graph pairs  $D = \{(G_1, Y_1), (G_2, Y_2), \dots, (G_N, Y_N)\}$
- The goal is to train a GNN model  $M$  that predicts the target values of given arbitrary graph pairs in an end-to-end manner, i.e.,  $Y_i = M(G_i)$ . The target  $Y$  is a scalar value for regression tasks, while it is a binary class label  $Y \in \{0,1\}$  for (binary) classification tasks.

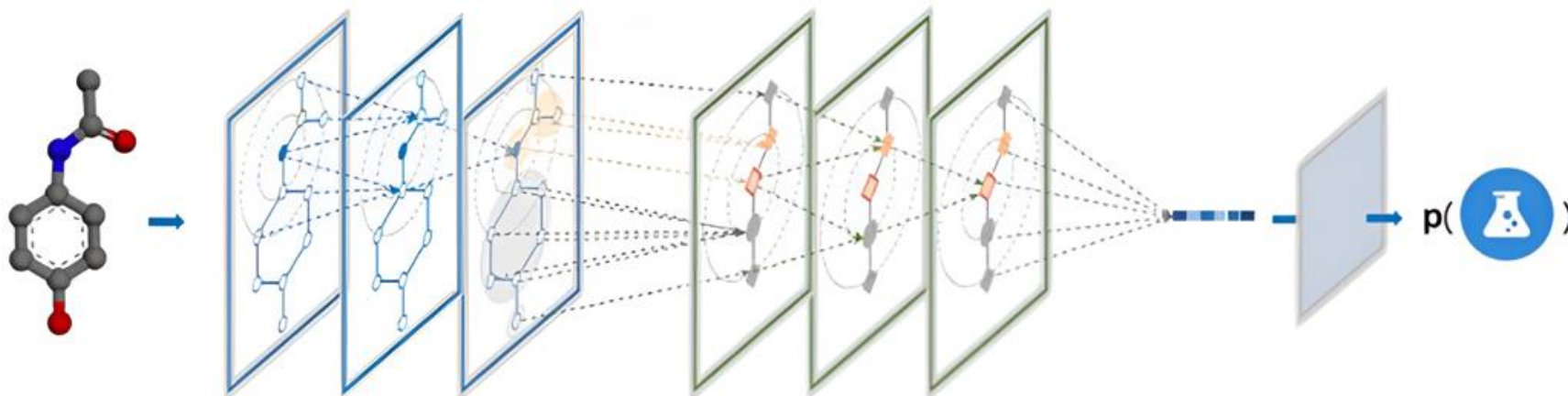


## ➤ Graph Representation of Molecules

- **Nodes (Atoms):** Each atom is represented as a node with features like atom type (e.g., carbon, oxygen), formal charge, hybridization, etc.
- **Edges (Bonds):** Bonds between atoms (e.g., single, double, aromatic) form the edges and carry properties that indicate bond strength, type, or length.
- **Graph Construction:** The molecule's 2D or 3D structure is converted into a graph that GNNs can process, encoding molecular connectivity and interactions.

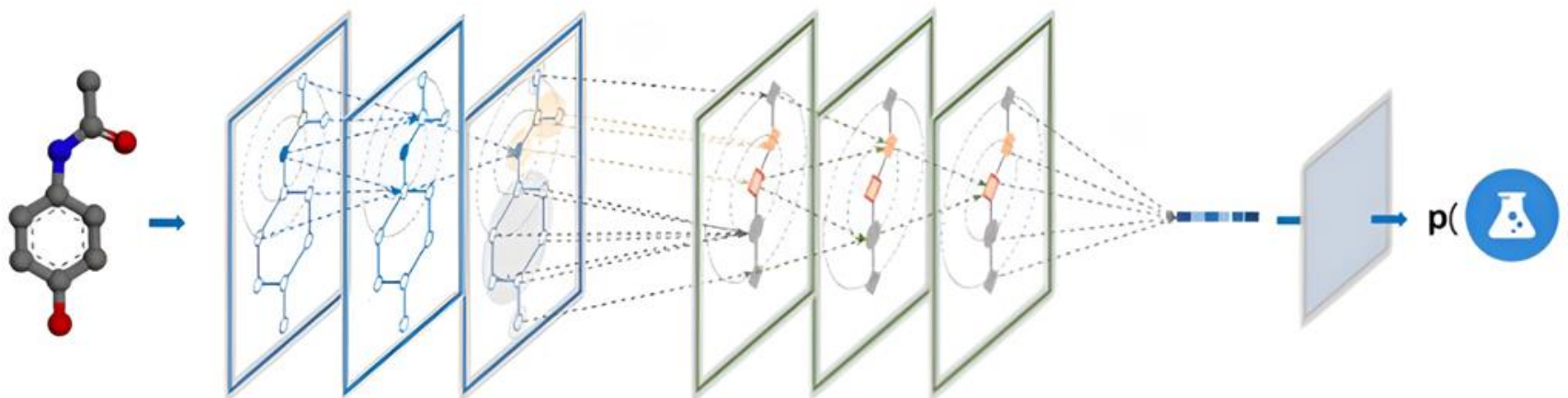


- Feature Extraction and Embedding Initialization
  - **Node Features:** Each atom's characteristics are embedded as vectors, capturing information such as atomic number, charge, and valency.
  - **Edge Features:** Bond characteristics can also be embedded to inform GNNs about bond types, order, or lengths (if using 3D data).



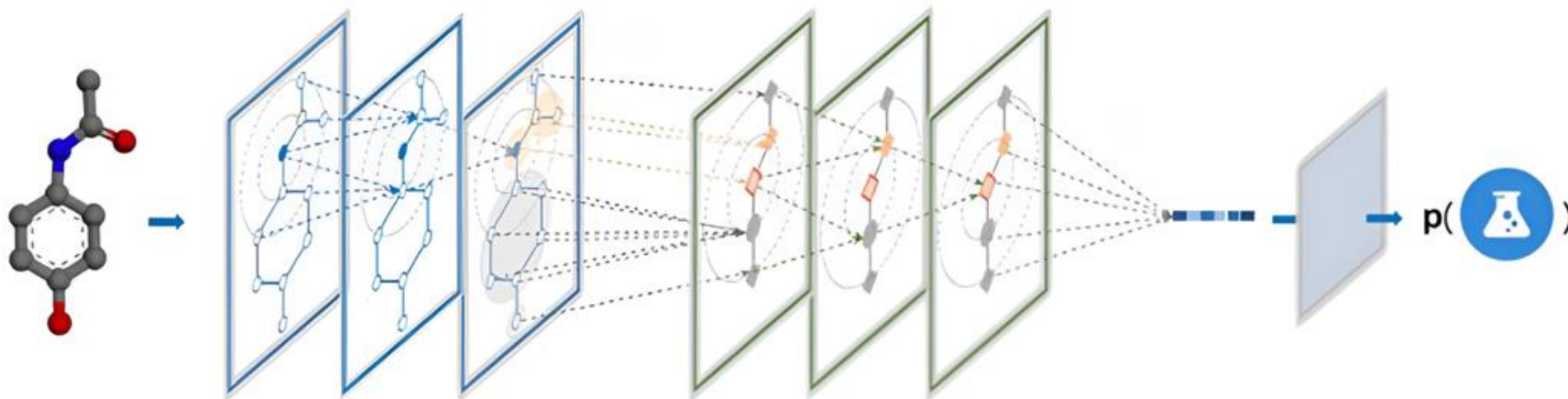


- Message Passing and Aggregation:
  - **Message Passing:** In each layer, nodes aggregate features from their neighboring nodes, “passing messages” to capture local interactions and structural information.
  - **Aggregation Function:** Common aggregation functions (mean, sum, max) help summarize neighboring information, creating a new node embedding that reflects both the node’s features and its neighborhood.
  - **Multiple Layers:** Stacking multiple GNN layers allows the model to capture information from nodes further away, which helps in learning both local and global molecular structures.

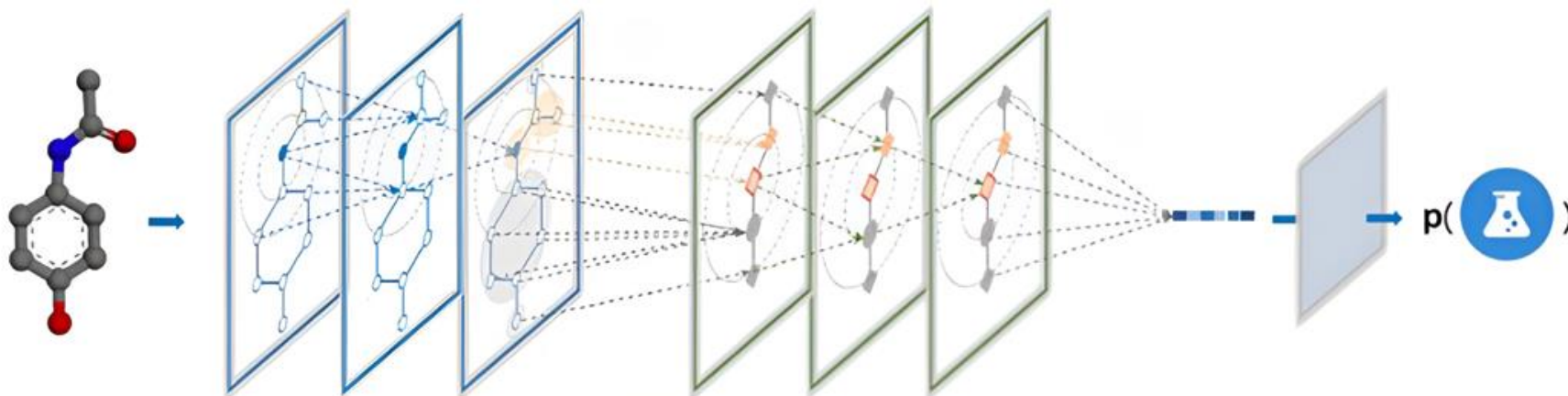


➤ Pooling and Readout:

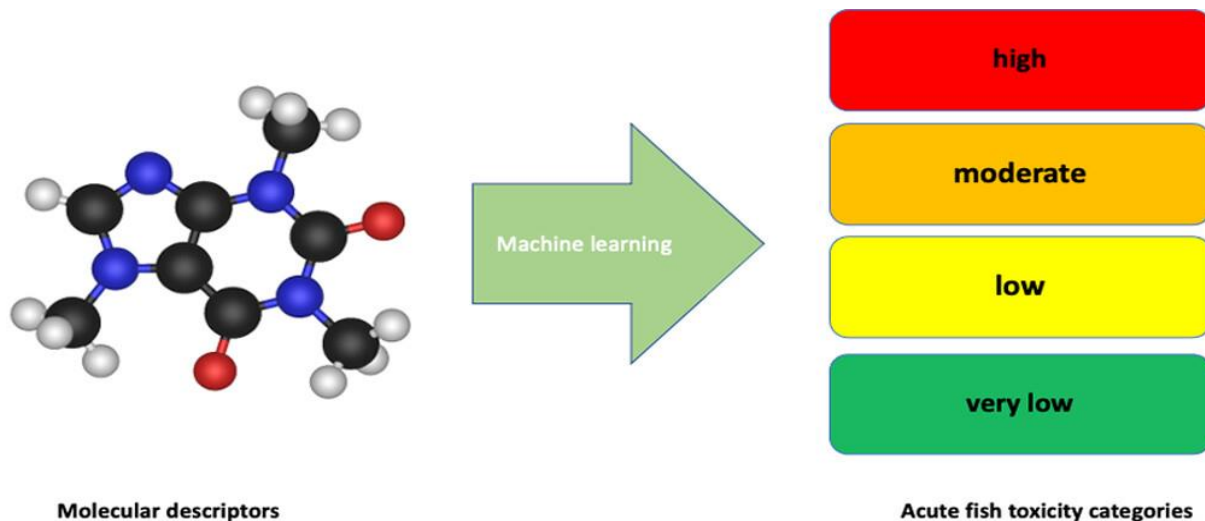
- **Global Pooling:** Since the final output is a single prediction per molecule, GNNs often use pooling methods to combine all node embeddings into a single graph-level representation. Pooling techniques like **mean pooling**, **max pooling**, or **attention pooling** summarize node embeddings into a fixed-size vector.
- **Hierarchical Pooling:** In complex molecules, hierarchical pooling (based on substructures or motifs) can capture higher-level chemical motifs, allowing the model to distinguish functional groups or rings.



- Classification Layer:
  - **Fully Connected Layer:** The pooled embedding is typically passed through a fully connected layer for classification.
  - **Output:** The model predicts a class label, such as “toxic” vs. “non-toxic” or the presence of a certain property (e.g., drug-like properties, fluorescence).



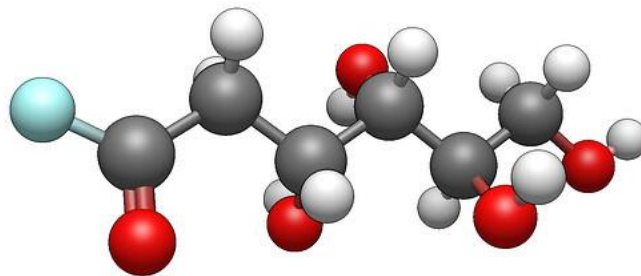
- **Goal:** Assign a categorical label to a molecule (e.g., toxic vs. non-toxic, active vs. inactive).
- **Applications:**
  - **Drug Discovery:** Predicting if a molecule is likely to be effective against a target disease.
  - **Toxicology:** Classifying compounds as toxic or non-toxic to ensure safety in drugs or materials.
  - **Materials Science:** Identifying molecules that meet specific properties like biodegradability, fluorescence, or electrical conductivity.



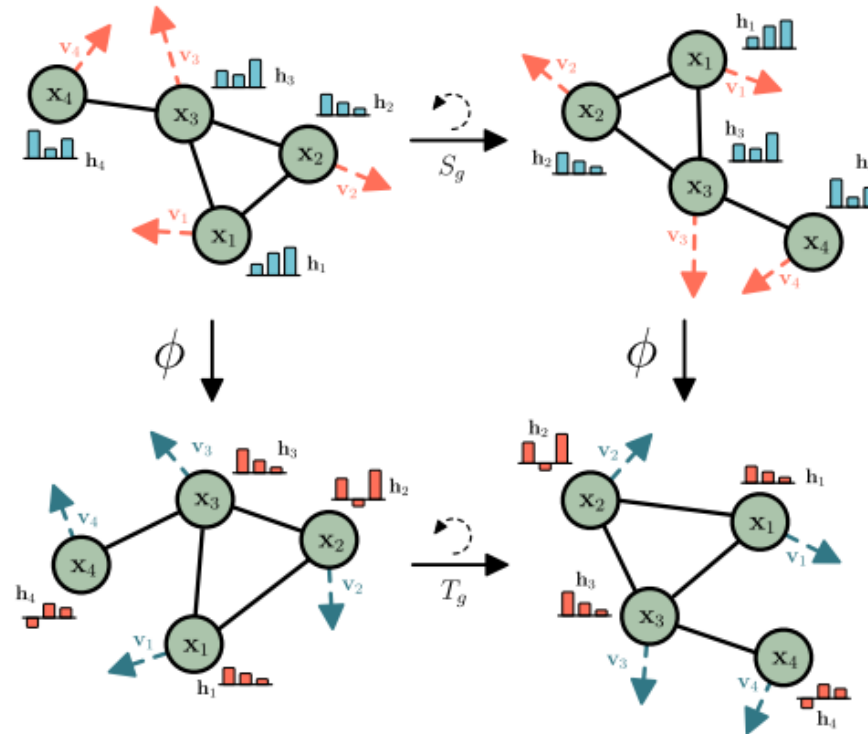
- **Goal:** Predict continuous properties of molecules, such as boiling point, binding affinity, or toxicity level.
- Applications
  - **Chemical and Physical Property Prediction:** Estimating properties like melting point, solubility, or partition coefficient.
  - **Drug Development:** Predicting binding affinities to determine how strongly a drug molecule interacts with a target.



- Equivariant Graph Neural Networks (GNNs) are designed to preserve certain symmetry properties, particularly rotation and translation, in their architecture.
- This is especially important in tasks involving 3D spatial data, such as molecular modeling or 3D object recognition, where the relationships between nodes must remain consistent despite changes in their spatial configurations.
- **Why 3D coordinates?**
  - The use of 3D coordinates is crucial for various tasks in molecular modeling and related fields, particularly when properties and behaviors of atoms and molecules are directly linked to their spatial arrangements.
  - In molecular learning, the spatial arrangement of atoms greatly influences their interactions, stability, and reactivity. Properties such as bond lengths, angles, and dihedral angles are inherently 3D.



- Rotation equivariance in graph neural networks (GNNs) refers to the property that a model's output remains consistent under rotations of the graph structure.
- Example of rotation equivariance on a graph with a Graph neural network



- Three types of equivariance on a set of particles  $x$ :
  - **Translation equivariance.** Translating the input by  $g \in R^n$  results in an equivalent translation of the output. Let  $x + g$  be shorthand for  $(x_1 + g, \dots, x_M + g)$ . Then  $y + g = \phi(x + g)$
  - **Rotation (and reflection) equivariance.** For any orthogonal matrix  $Q \in R^{n \times n}$ , let  $Qx$  be shorthand for  $(Qx_1, \dots, Qx_M)$ . Then rotating the input results in an equivalent rotation of the output  $Qy = \phi(Qx)$ .
  - **Permutation equivariance.** Permuting the input results in the same permutation of the output  $P(y) = \phi(P(x))$  where  $P$  is a permutation on the row indexes.

- Equivariant Graph Convolutional Layer (EGCL) takes as input the set of node embeddings  $h^l = \{h_0^l, \dots, h_{M-1}^l\}$ , coordinate embeddings  $x^l = \{x_0^l, \dots, x_{M-1}^l\}$  and edge information  $E = (e_{ij})$  and outputs a transformation on  $h^{l+1}$  and  $x^{l+1}$ . Concisely:  $h^{l+1}, x^{l+1} = \text{EGCL}[h^l, x^l, E]$ .
- The equations that define this layer are the following:

$$\begin{aligned}\mathbf{m}_{ij} &= \phi_e \left( \mathbf{h}_i^l, \mathbf{h}_j^l, \|\mathbf{x}_i^l - \mathbf{x}_j^l\|^2, a_{ij} \right) \\ \mathbf{x}_i^{l+1} &= \mathbf{x}_i^l + C \sum_{j \neq i} (\mathbf{x}_i^l - \mathbf{x}_j^l) \phi_x (\mathbf{m}_{ij}) \\ \mathbf{m}_i &= \sum_{j \neq i} \mathbf{m}_{ij} \\ \mathbf{h}_i^{l+1} &= \phi_h (\mathbf{h}_i^l, \mathbf{m}_i)\end{aligned}$$

➤ **Motivation:**

- Labeled molecules only occupy an extremely small portion of the enormous chemical space since they can only be obtained from wet-lab experiments or quantum chemistry calculations, which are time-consuming and expensive.
- Directly training GNNs on small labeled molecule datasets in a supervised fashion is prone to over-fitting and the trained GNNs can hardly generalize to out-of-distribution data.

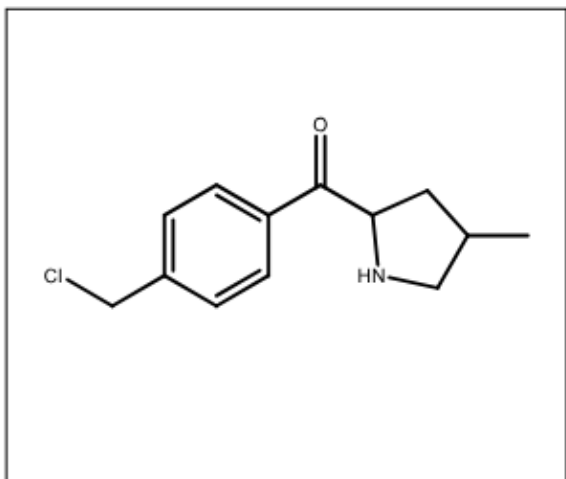
➤ **Question:** How to pretrain GNNs on large-scale unlabeled data via self-supervised pretraining?

- Node-level strategies
- Contrastive learning strategies
- Subgraph-level strategies

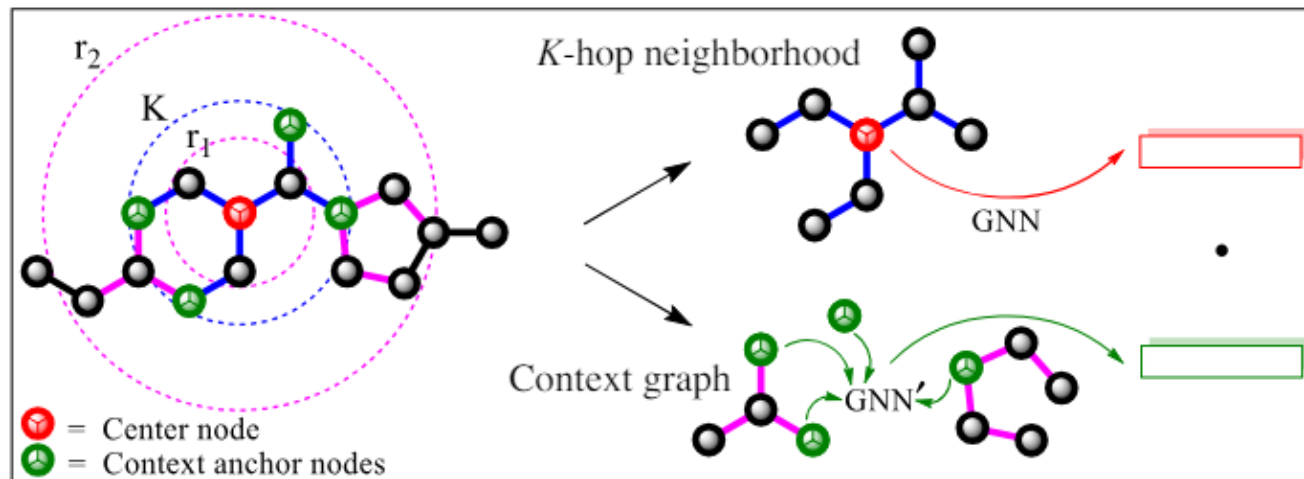


- Node-level strategies: Develop some strategies to relieve the scarce task-specific labels problem
  - In Context Prediction: Nodes in similar contexts obtain nearby embeddings
  - In Attribute Masking: Capture domain knowledge from masking node attributes.

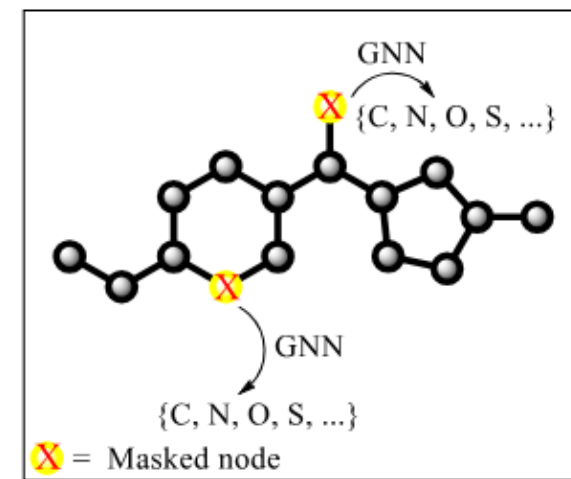
Input molecules



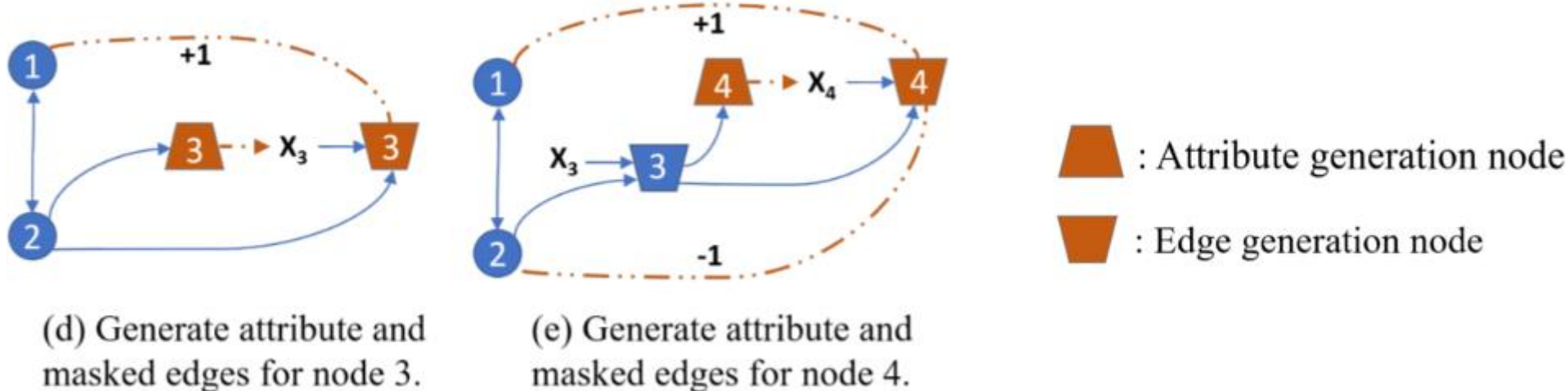
Context Prediction



Attribute Masking

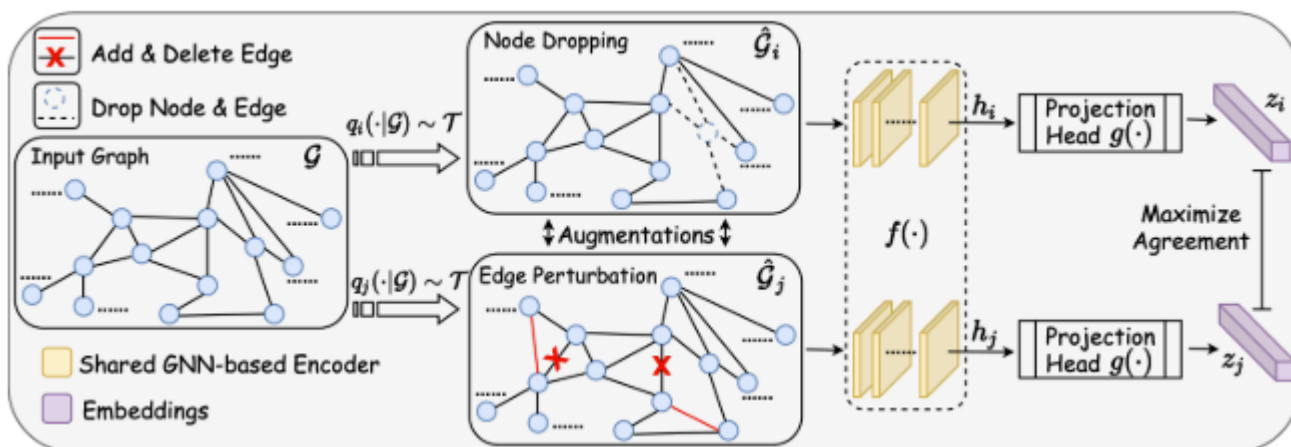


- Pre-train large-scale graph with reconstructing the input graph, which utilizes the unlabeled data for label scarcity issue. Decompose the reconstruction process into two coupled steps:
  - Attribute generation: given observed edges, generate **node attributes**
  - Edge generation: given generated attributes, generate **masked edges**



- Contrastive learning methods: Design types of graph augmentations to incorporate various impacts in different settings

Data augmentation	Type	Underlying Prior
Node dropping	Nodes, edges	Vertex missing does not alter semantics.
Edge perturbation	Edges	Semantic robustness against connectivity variations.
Attribute masking	Nodes	Semantic robustness against losing partial attributes.
Subgraph	Nodes, edges	Local structure can hint the full semantics.


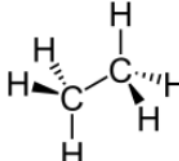


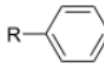
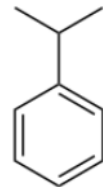


Contrastive Loss:

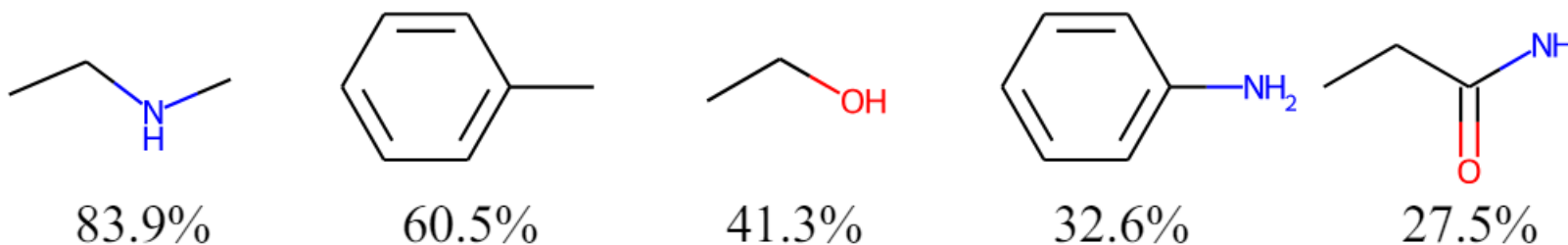
$$\ell_n = -\log \frac{\exp(\text{sim}(\mathbf{z}_{n,i}, \mathbf{z}_{n,j})/\tau)}{\sum_{n'=1, n' \neq n}^N \exp(\text{sim}(\mathbf{z}_{n,i}, \mathbf{z}_{n',j})/\tau)}$$

- GCC: Graph Contrastive Coding for Graph Neural Network Pre-Training (Qiu et al. KDD2020)
- Graph Contrastive Learning with Augmentations (You et al. NIPS2020)
- Graph Contrastive Learning with Adaptive Augmentation (Zhu et al. WWW2021)

- Limitations of node-level strategies:
  - Models can not be expressive to capture directly the functional groups (characteristic chemical reactions).
  - Contrastive learning strategies can destroy the molecular structures by data augmentation.
  - E.g., Hydrocarbons are a class of molecule that is defined by functional groups called hydrocarbyls that contain only carbon and hydrogen

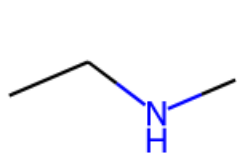
Chemical class	Group	Formula	Structural Formula	Prefix	Suffix	Example
Alkane	Alkyl	$R(CH_2)_nH$		alkyl-	-ane	 Ethane
Alkene	Alkenyl	$R_2C=CR_2$		alkenyl-	-ene	 Ethylene (Ethere)
Alkyne	Alkynyl	$RC\equiv CR'$	$R-C\equiv C-R'$	alkynyl-	-yne	$H-C\equiv C-H$ Acetylene (Ethyne)
Benzene derivative	Phenyl	$RC_6H_5$ $RPh$		phenyl-	-benzene	 Cumene (Isopropylbenzene)

- Subgraph-level strategies: Leverage domain knowledge to enhance GNNs for capturing functional groups (subgraphs).
  - capture the regularities of atomic combinations
  - discover repetitive patterns in molecules, thus more capable of generating realistic molecules.
- It is able to reflect chemical properties, since it has been revealed that the chemical properties are closely related to certain subgraphs
  - E.g., The frequent subgraphs are discovered in ZINC dataset:

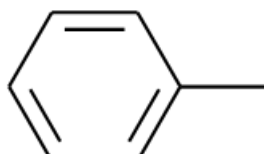




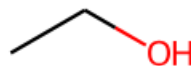
- Leverage domain knowledge to enhance GNNs for capturing functional groups (subgraphs).
  - capture the regularities of atomic combinations
  - discover repetitive patterns in molecules, thus more capable of generating realistic molecules.
- It is able to reflect chemical properties, since it has been revealed that the chemical properties are closely related to certain subgraphs
  - E.g., The frequent subgraphs are discovered in ZINC dataset:



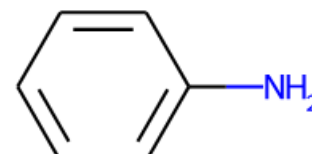
83.9%



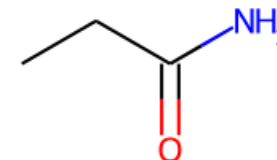
60.5%



41.3%

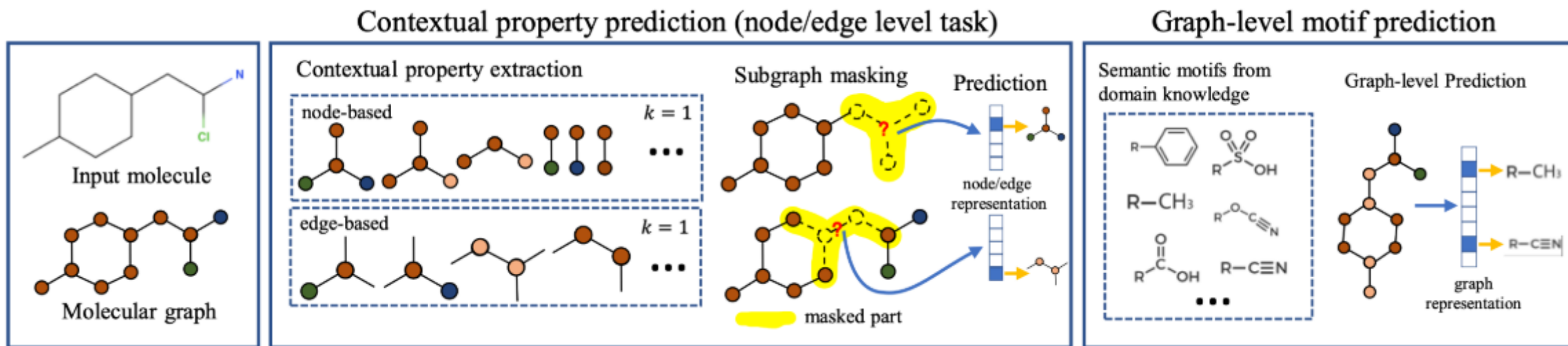


32.6%

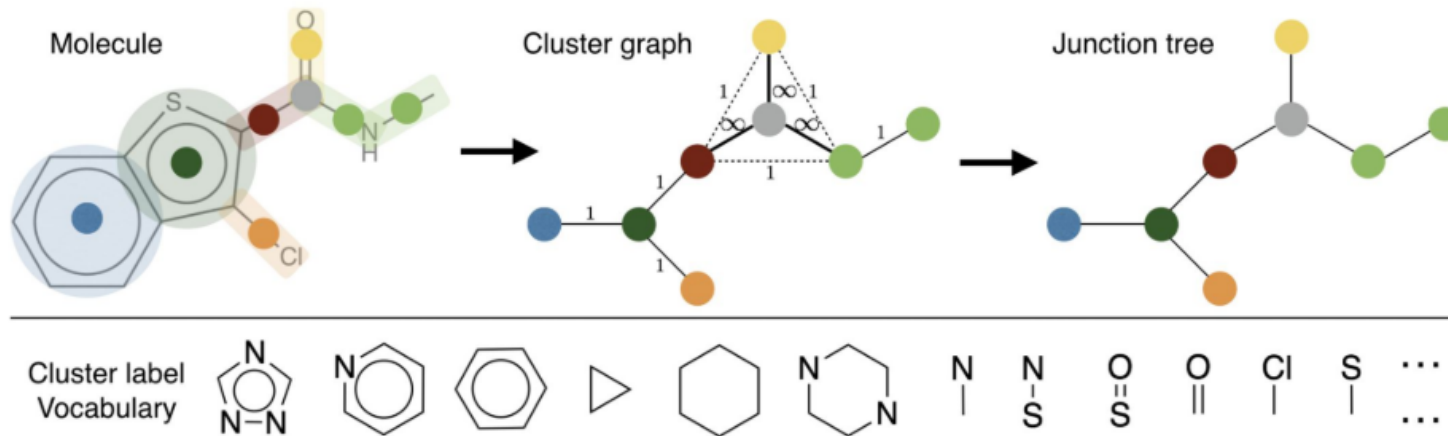


27.5%

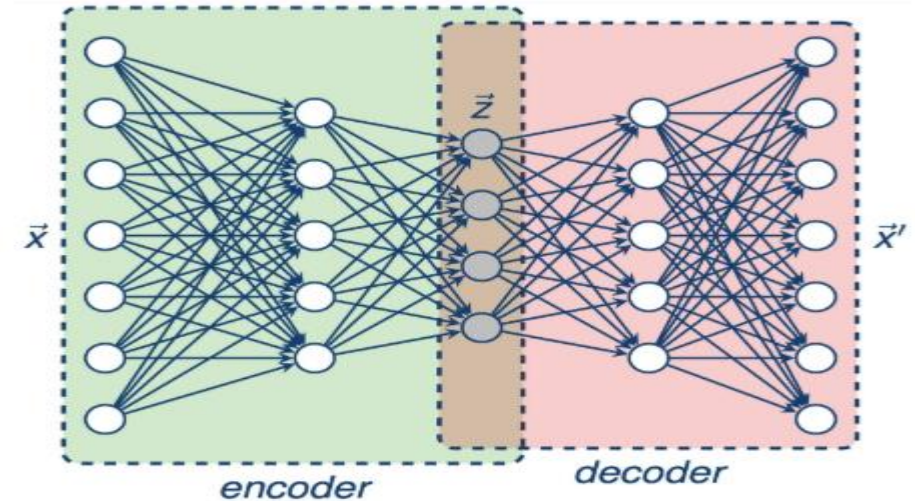
- GROVER: Node-level and Subgraph-level combination
- Designed self-supervised tasks in node-, edge- and graph-level, learn rich **structural** and **semantic** information of molecules
  - Contextual property prediction: predict **masked node/edge attributes** set
  - Motif prediction : predict the classes of **the motif** that occur in a given molecule



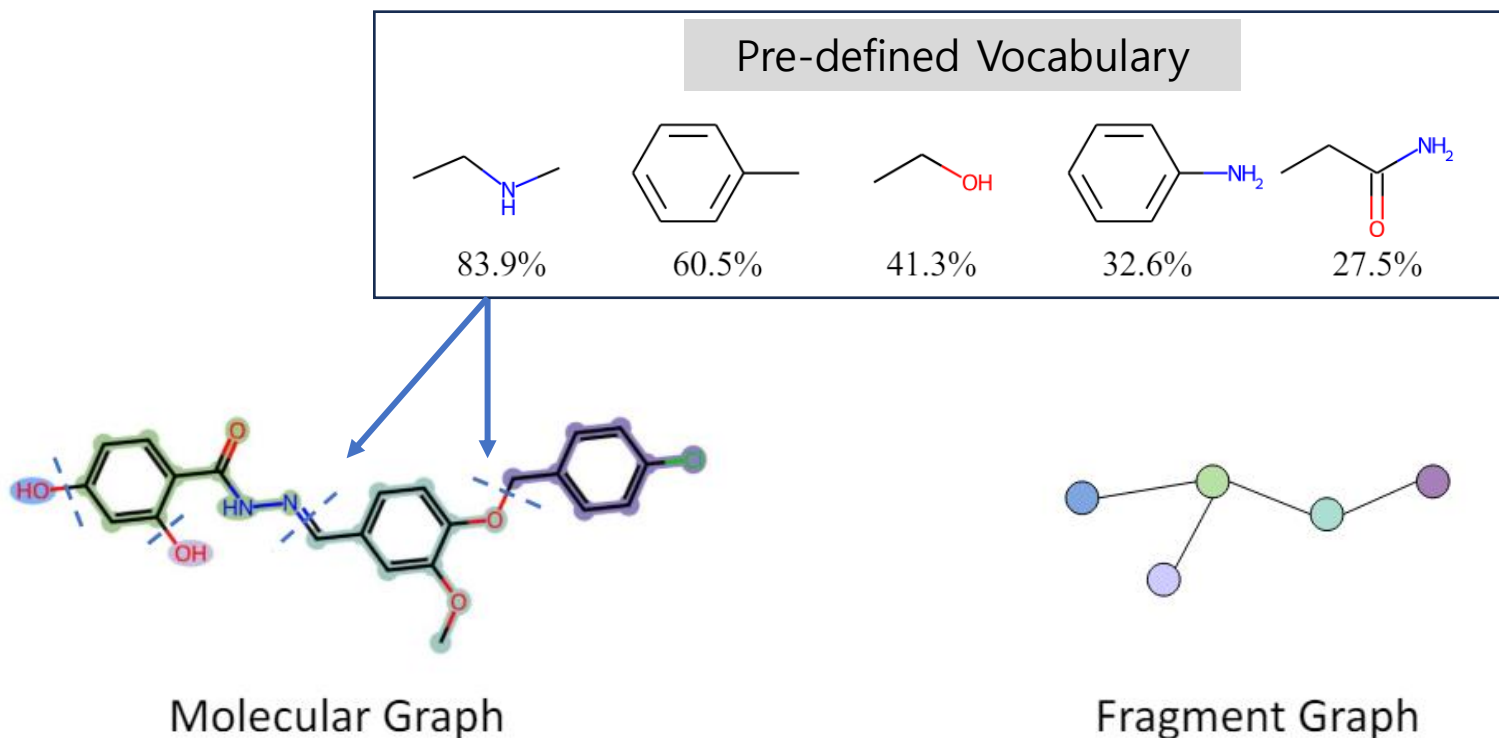
- **Key idea:** Clustering Atoms to generate Junction tree



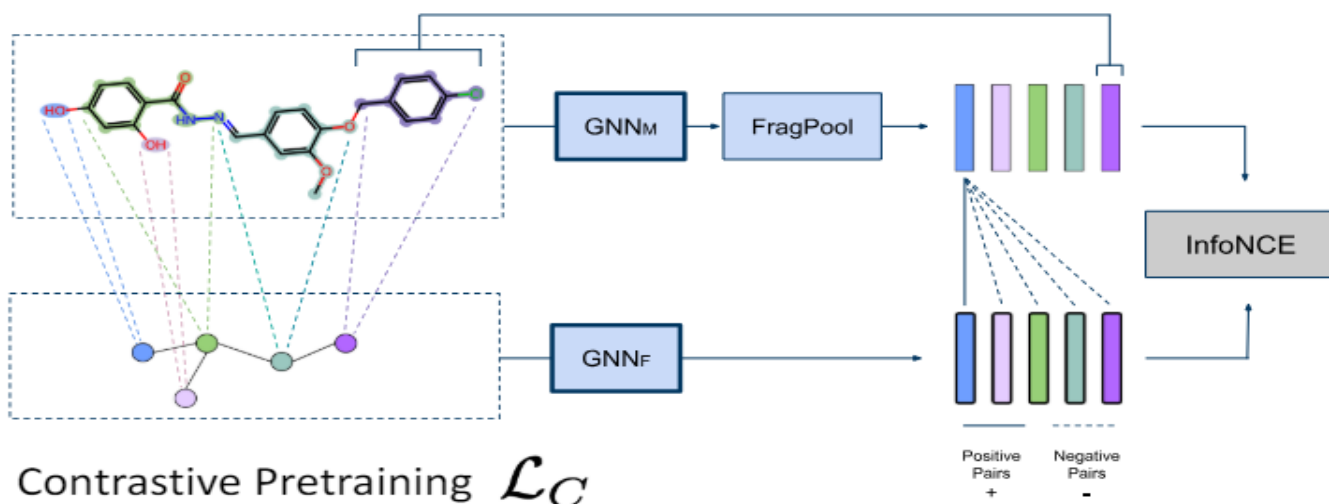
# Variational Autoencoder



- Fragment-based Pretraining and Finetuning on Molecular Graphs (NeuIPS'23)
- A pretraining task at the subgraph level.
- Prepare molecular graphs and fragment graphs and utilizing them in both pretraining and finetuning. Molecules are fragmented according to a vocabulary extracted via existing frequency-based methods.



- Fragment-based Pretraining and Finetuning on Molecular Graphs (NeuIPS'23)
- Fragmenting original molecules as a bags of groups:
  - Use Prior Knowledge to build the Dictionary to decompose molecules as a bag of functional groups



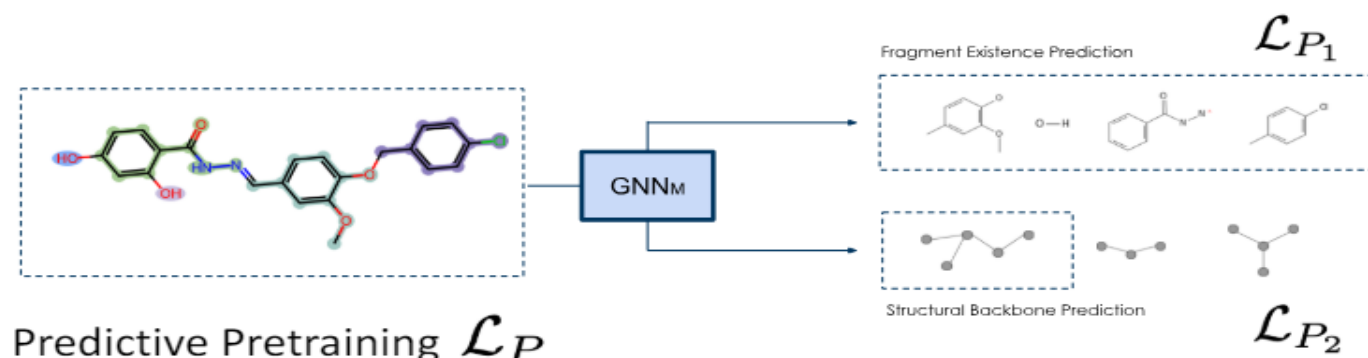
$GNN_M$  Process molecular graphs, encoding local patterns

$GNN_F$  Process fragment graphs, encoding global patterns

Contrastively enforces both local and global structural patterns into node embeddings

$\mathcal{L}_{P_1}$  Fragment existence prediction

$\mathcal{L}_{P_2}$  Main structure backbone prediction

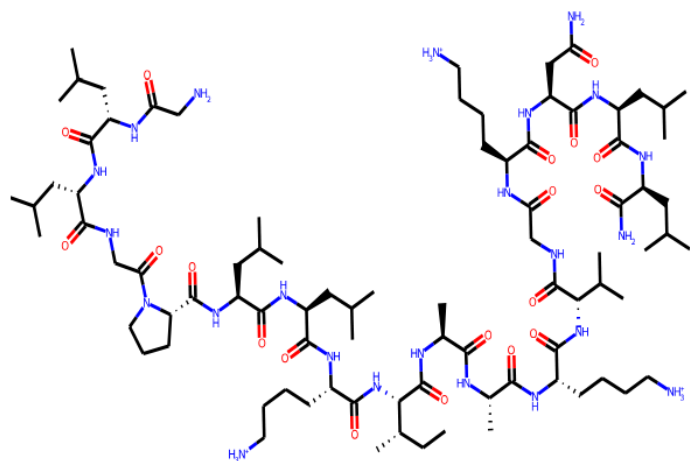


Combined Pretraining

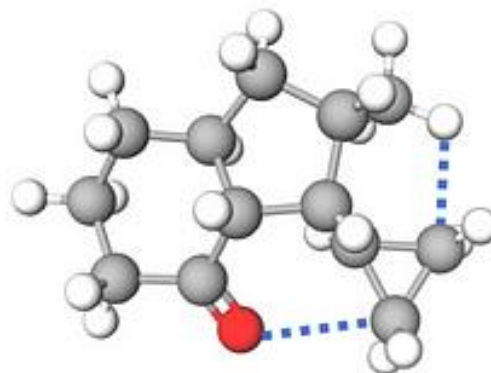
$$\mathcal{L}_P = \mathcal{L}_{P_1} + \mathcal{L}_{P_2}$$

$$\mathcal{L} = \alpha \mathcal{L}_P + (1 - \alpha) \mathcal{L}_C$$

- Efficiency on large-scale graphs: The molecules in the existing benchmarks are of small sizes, *i.e.*, number of nodes.
  - If graph sizes are large, GNNs would potentially lose out signals flowing from distant nodes due to over-squashing.
  - It is challenging to capture information exchange among distant nodes, where interactions are required far away from the near-local neighborhood.



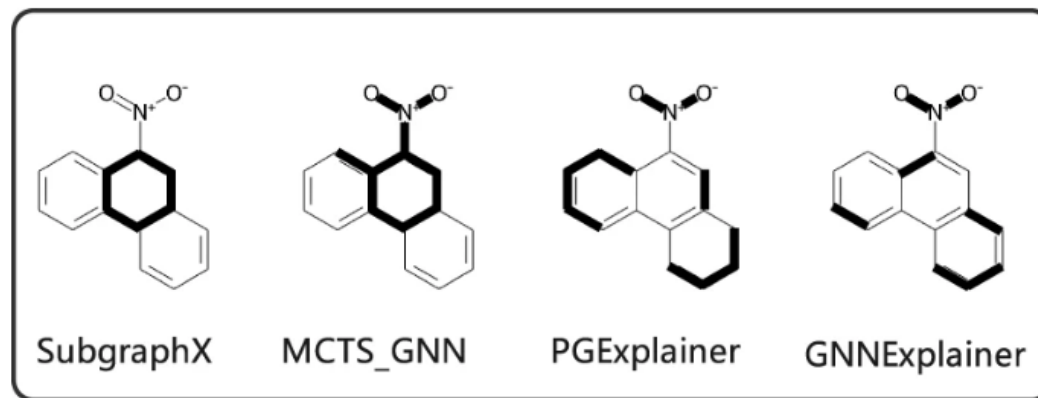
Long peptide sequence: “GLLGPLLKIAAKVGKNLL”



A molecule with long range interactions.  
The **dotted** lines show 3D atomic contact.

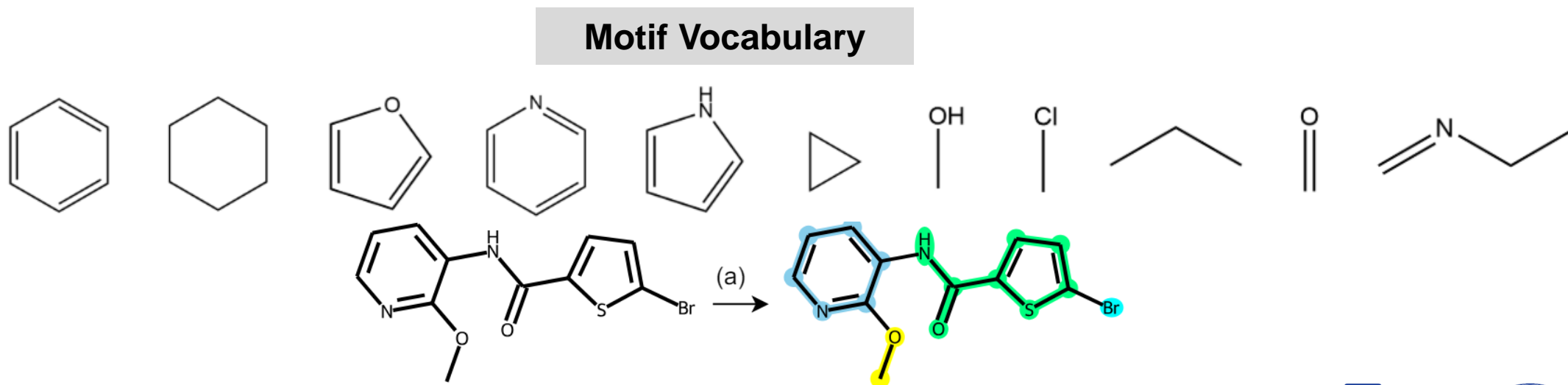


- Explainability: Unlike images and texts, molecules are not grid-like data, which means there is no locality information and each node has different numbers of neighbors.
  - The adjacency matrices represent the topology information and only contain discrete values. The discrete adjacency matrices cannot be optimized in the same manner.
  - It is in general difficult to understand the reasoning which leads to the models' output predictions.
    - E.g., Different explainable strategies can provide different solutions



- Gnnexplainer: generating explanations for graph neural networks. (NeurIPS 2019).
- Parameterized explainer for graph neural network. (*Adv. neural Inf. Process. Syst* 2020).
- On explainability of graph neural networks via subgraph explorations. (PMLR 2021).

- The use of domain knowledge: To capture functional groups: existing studies focus on extracting frequent subgraphs to generate dictionary.
- **It is challenging:**
  - More than one solution exists to break the molecules into bags of fragments.
  - The optimal settings are not easy to find (dictionary size, rules,...). There exist more than one solutions to break molecules.
  - It hard to generalize large-scale molecules in varied domains.





네트워크 과학연구실  
NETWORK SCIENCE LAB



가톨릭대학교  
THE CATHOLIC UNIVERSITY OF KOREA

