# Improving Out-of-Distribution Generalization in Graphs via Hierarchical Semantic Environments

Biomedical domain

**Yinhua Piao, Sangseon Lee, Yijingxiu Lu, Sun Kim.**

Bio & Health Informatics Lab

Department of Computer Science and Engineering

Seoul National University, Seoul, Korea

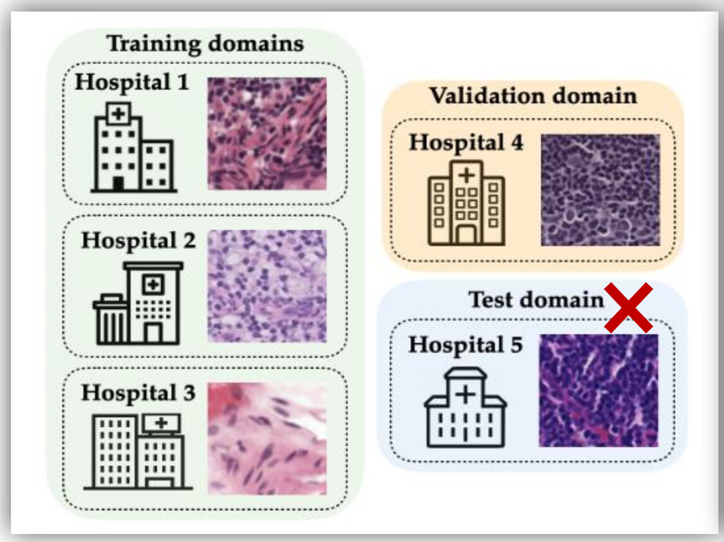
# Out-of-distribution Generalization

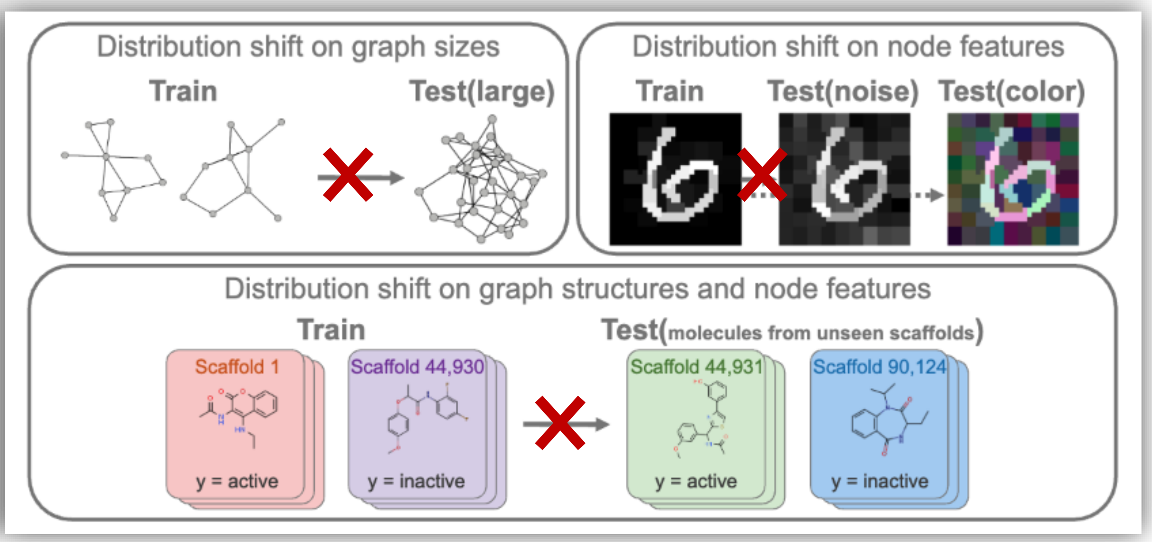
# Out-of-distribution Generalization

- Models learned with **Empirical Risk Minimization** often fail to generalize to OOD data.

$$\min_{f} \mathbb{E}_{(x,y)\sim \mathrm{P}_{\mathrm{train}}(\mathbf{x},\mathbf{y})}[l(f(x),y)], \qquad \mathrm{P}_{\mathrm{test}}(\mathrm{x},\mathrm{y}) \neq \mathrm{P}_{\mathrm{train}}(x,y)$$

- Applications: Molecule prediction, pandemic prediction, medical detection for COVID-19.

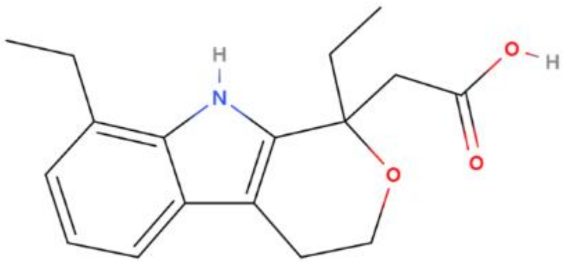

*(A robey, et al., 2021, Li, Haoyang, et al., 2022, Arjovsky et al., 2022)*

# OOD Generalization in Euclidean and Non-Euclidean Domains

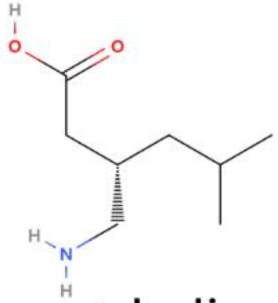
**Euclidean-based OOD**: featrure distribution shifts
**Non-euclidean-based OOD**: feature & structural shifts simultaneously.

Euclidean domain

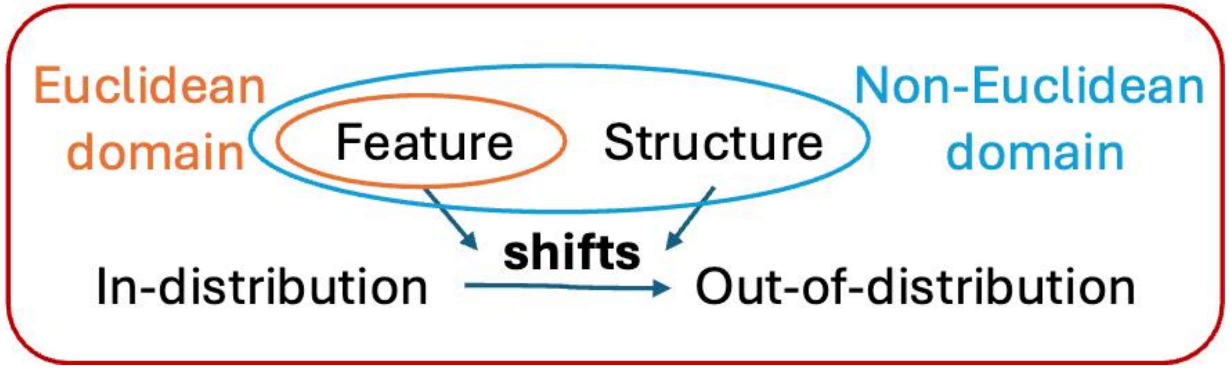
Non-Euclidean domain



etodolac

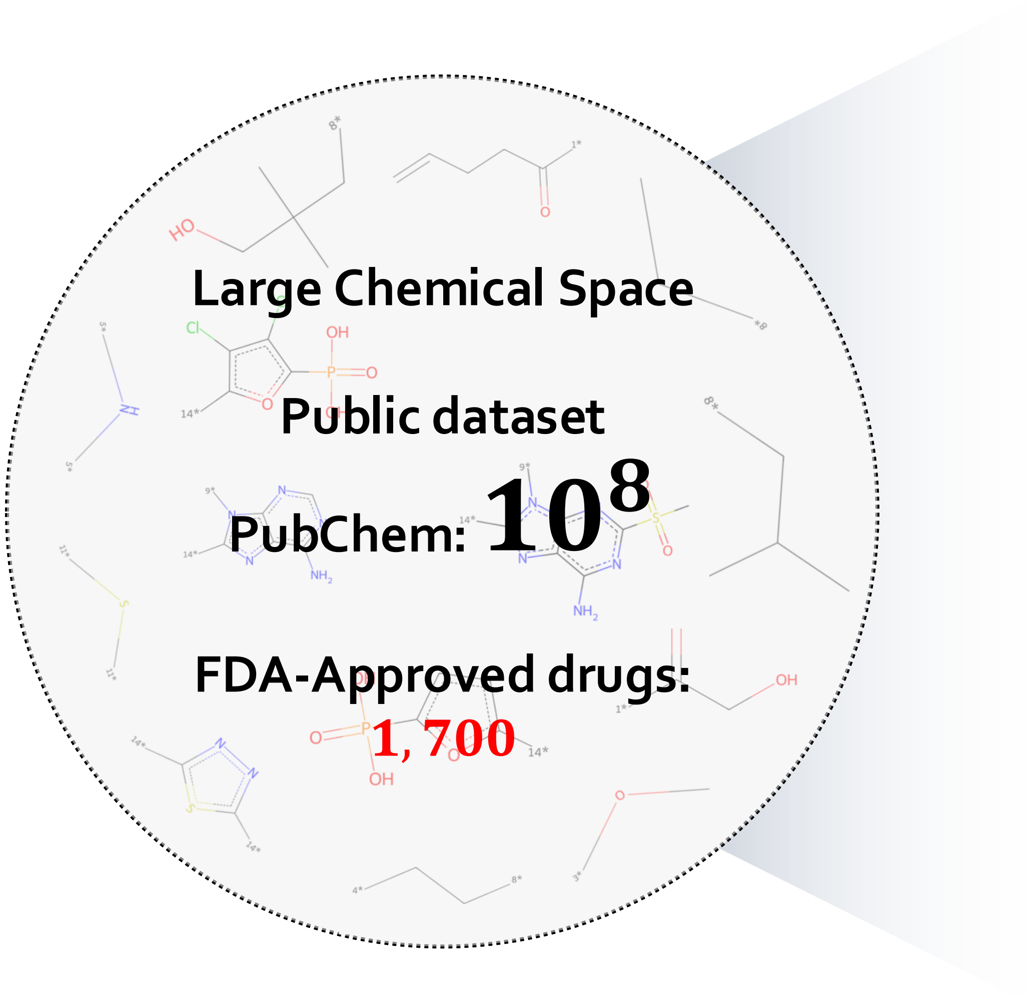


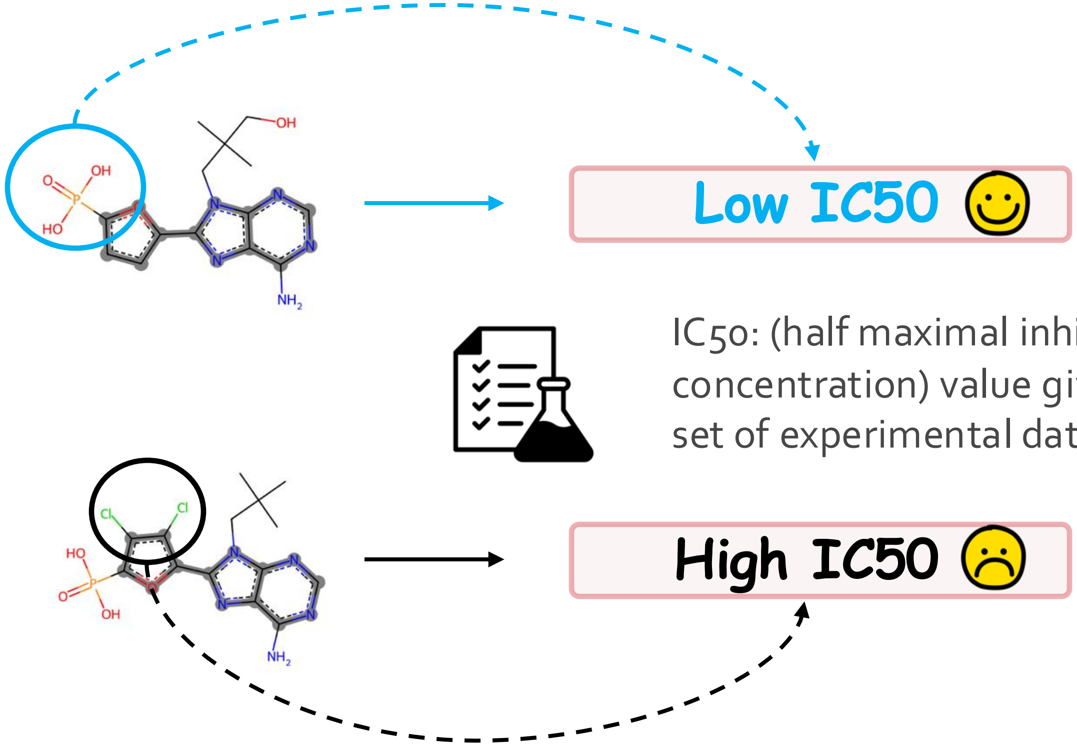
pregabalin



# Graph OOD Generalization in Chemical Domain

# Molecules and Molecular Property Prediction

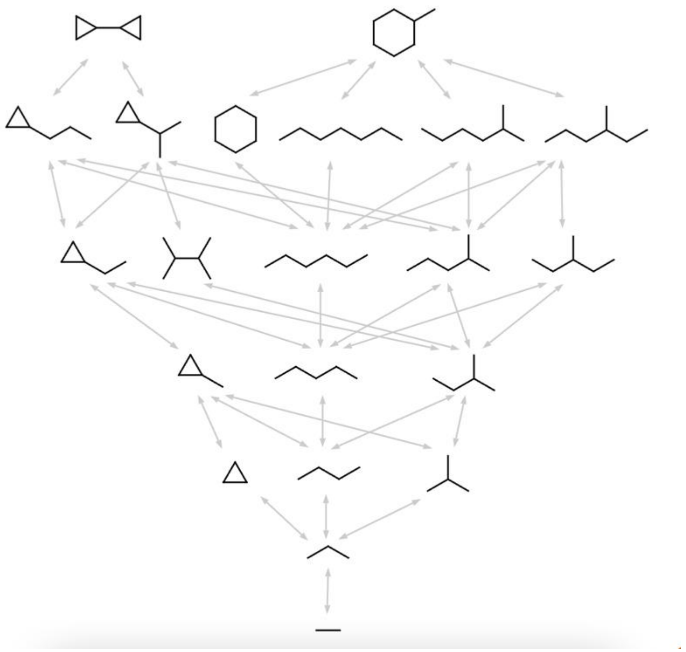



IC50: (half maximal inhibitory concentration) value given a set of experimental data.

# Challenge of Subgraph Mining

- **Subgraph mining**



Subgraph is unknown !

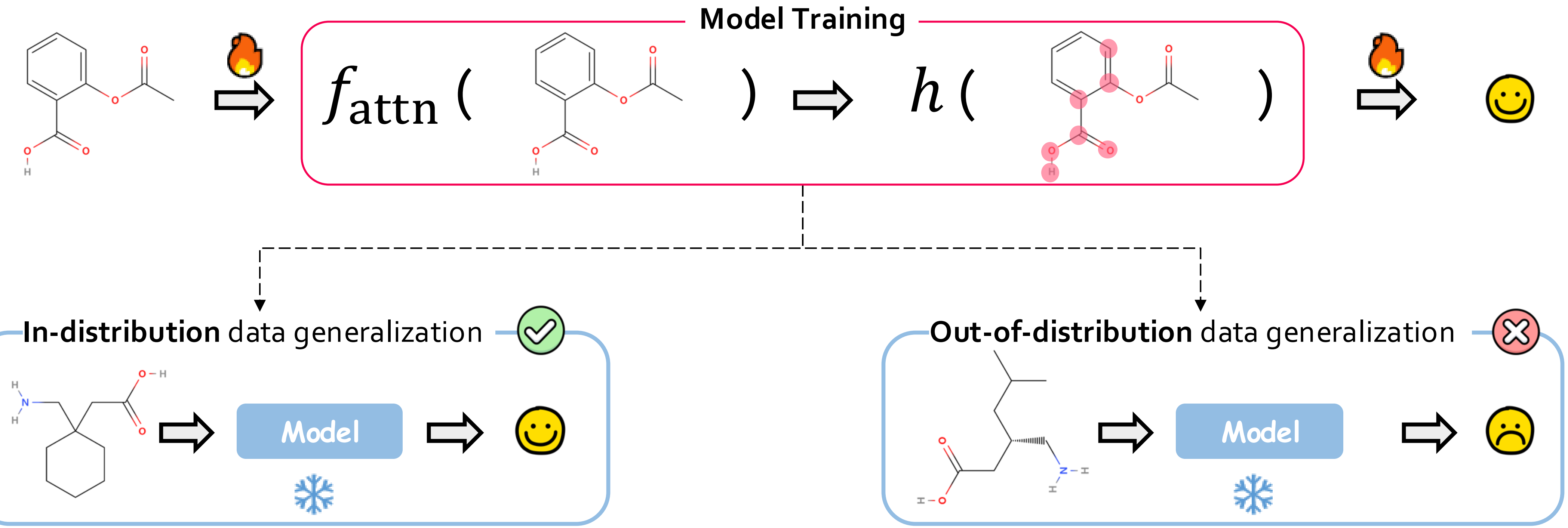
For a molecule with $n$ bonds, number of possible subgraphs:

**$2^n$combinations**
(exponential growth)

Huge chemical subgraph search space

*SmallWorld: Efficient Maximum Common Subgraph Searching of Large Chemical Databases.*
*ACS National Meeting, Philadelphia, USA 22nd August 2012*

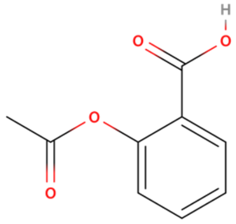
# Subgraph modelling is important in graph OOD generalization

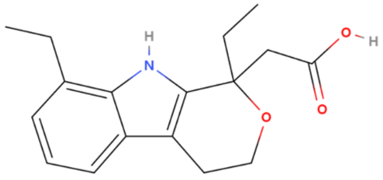



Poor generalization on out-of-distribution data.

# An example

**Train data**
(Ring structured)



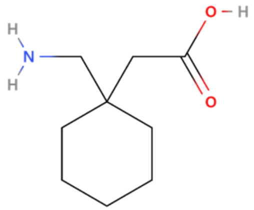


***Aspirin*** ***Etodolac***

Model 🔥

Relieve pain

**In-distribution Test data**
(Ring-structured)

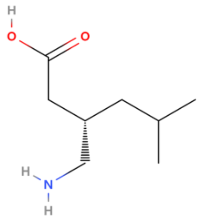


***Gabapentin***

Model ❄

Relieve pain ?

**Out-of-distribution Test data**
(non-Ring structured)



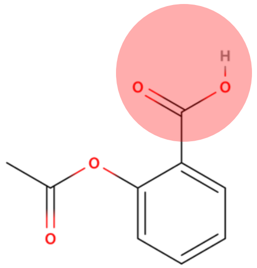
***Pregabalin***

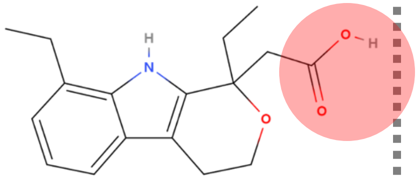
Model ❄

Relieve pain ?

# An example

**Train data**
(Ring structured)



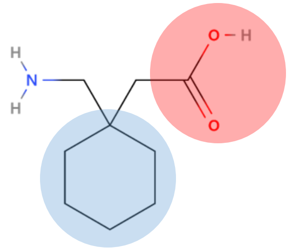


***Aspirin*** ***Etodolac***

Model

Relieve pain

**In-distribution Test data**
(Ring-structured)

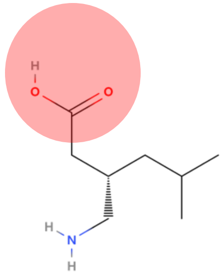


***Gabapentin***

Model

Relieve pain

**Out-of-distribution Test data**
(non-Ring structured)



***Pregabalin***

NO RING?

Model

Others

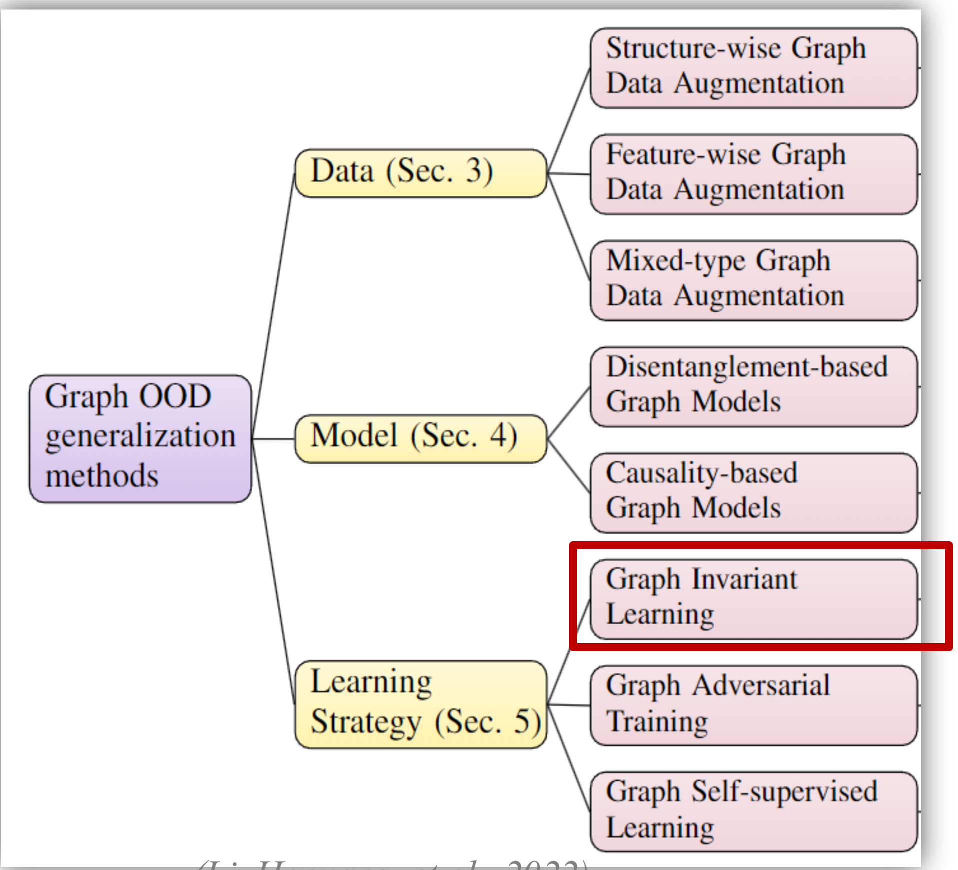
More challenging on unseen and highly different molecules (size, scaffolds, etc)

How to learn an effective subgraph for the unknown molecule's property?

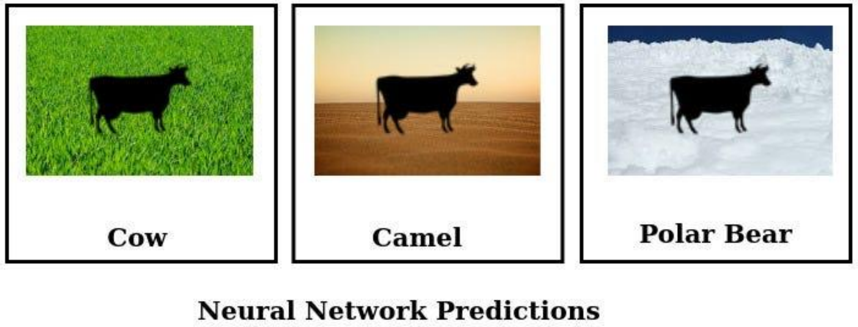
# Related works on Graph OOD Generalization

# Graph Invariant Learning

Invariant Learning treats the cause of distribution shifts between testing and training data as a potential unknown environmental variable e ∈ Ɛ (*hospital*, *size*, *color*, *scaffold*, *etc*).




(Li, Haoyang, et al., 2022)




Emprical Risk Minimization

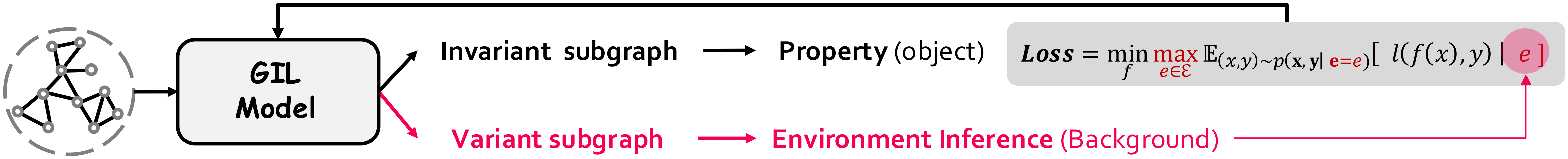
$$\min_{f} \mathbb{E}_{(x,y)\sim p(x,y))}[l(f(x),y)]$$

Invariant Risk Minimization

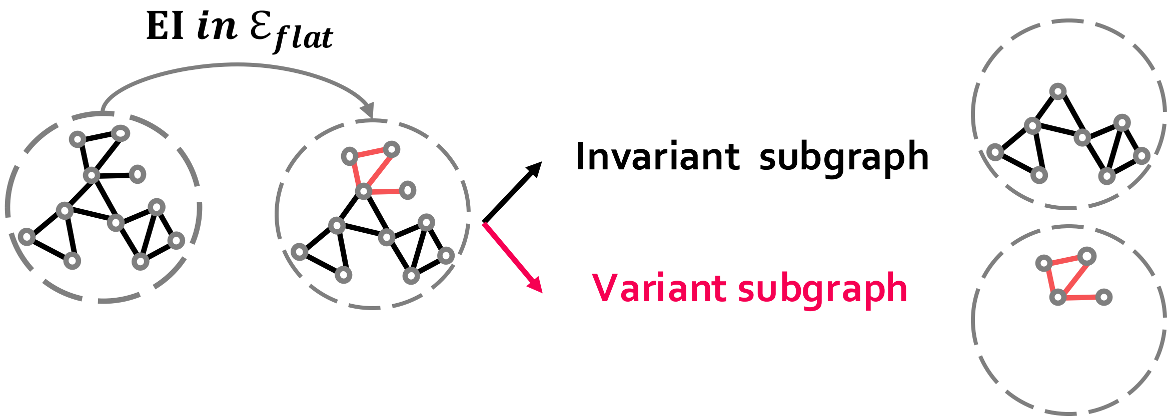
$$\min_{f} \max_{e\in\mathcal{E}} \mathbb{E}_{(x,y)\sim p(\mathbf{x},\,\mathbf{y}|\ \mathbf{e}=e)}[l(f(x),y)|e]$$

# Graph Invariant Learning – Existing works

- Graph Invariant Learning (GIL) improves Model generalization on OOD data.



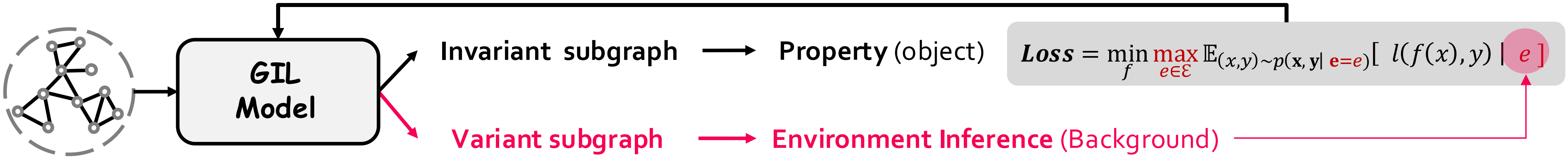

Existing GIL Method: Flat environment inference.



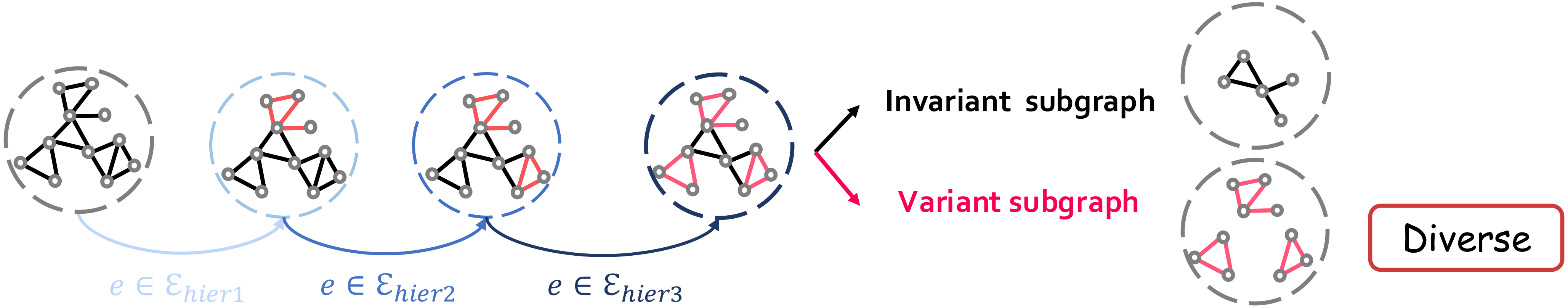

Ignore implicit hierarchy of environment → inductive bias

# Graph Invariant Learning – Our works

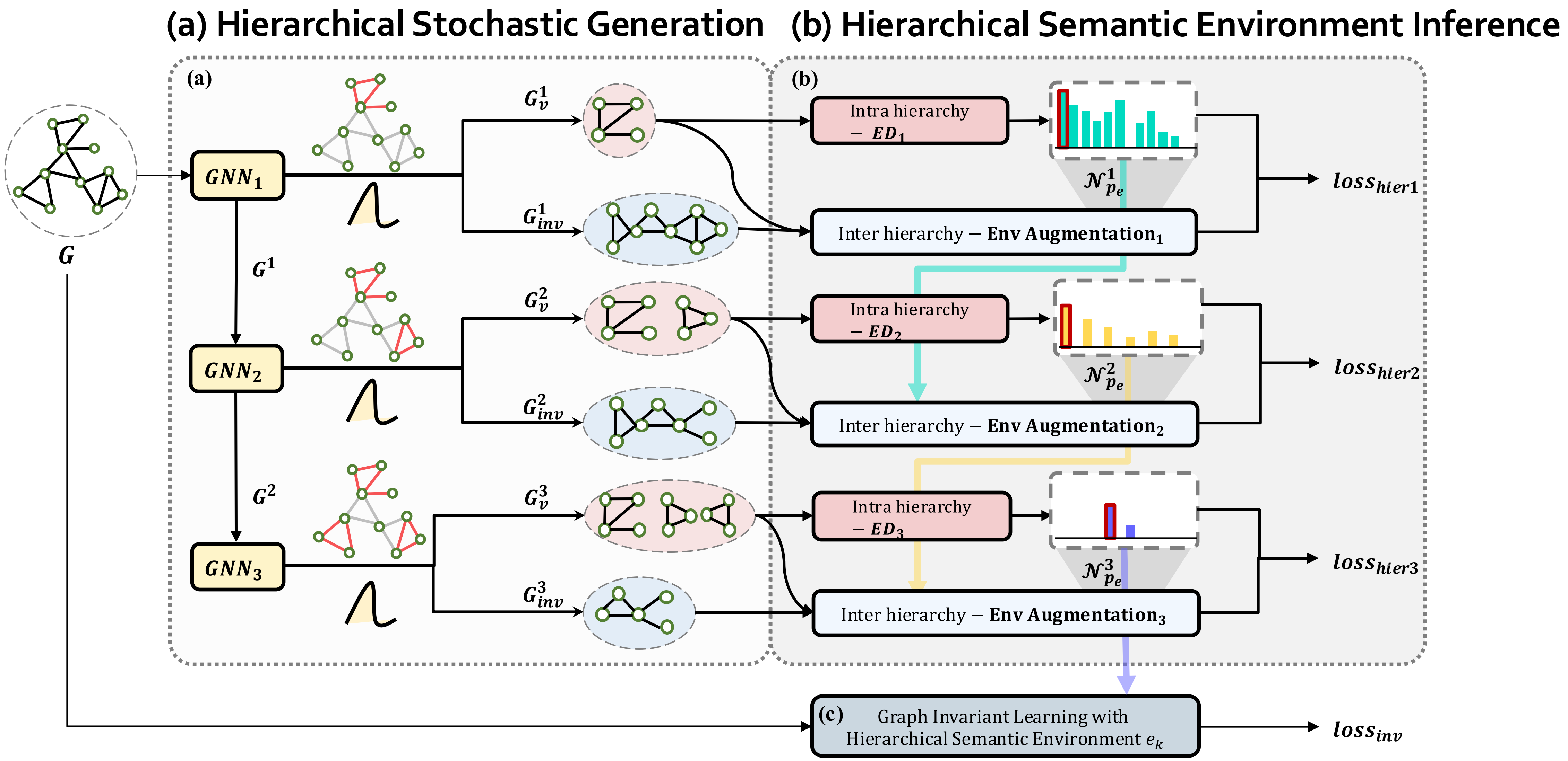
➢ **Graph Invariant Learning (GIL) improves Model generalization on OOD data.**




Our Method: Hierarchical environment inference.



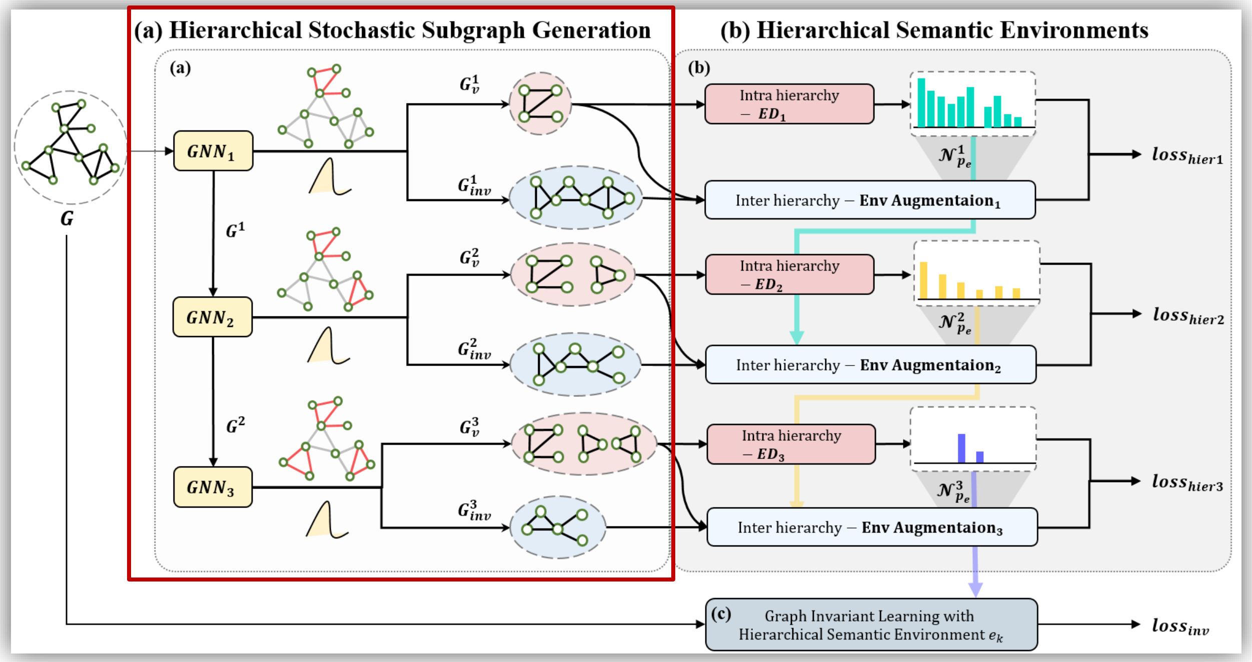
# Our Approach



# Our Approach

## (a) Hierarchical Stochastic Generation

- Input: $G = (V, A), \mathrm{h} = \mathrm{GNN}(\mathrm{V},\ \mathrm{A})$
  - Probability distribution of edge:
    $$s_{ij} = S(h_{ij}) = \sigma\left(\mathrm{MLP}([h_i, h_j])\right)$$
  - Stochastic Edge Selection via Gumbel-softmax:
    $$\mathbf{p}_{ij} \in \{0,1\} \sim Bern(s_{ij})$$
  - Hierarchical Stochastic Subgraph Generation
    $$N_{ij}^{k} = N_{ij}^{k-1} + 1\{N_{ij}^{k-1} = 0 \text{ and } \mathbf{p}_{ij}^{k} > T\},$$
    $$A_v^k \leftarrow A \odot N^k, \qquad A_{inv}^k \leftarrow A - A_v^k$$
- Output: $\{G_v^k$: variant subgraph, $G_{inv}^k$: invariant subgraph$\}$



# Our Approach

## (b) Hierarchical Semantic Environments

- Intra-Hierarchy Environment Diversification.

$$L_{ED} = -\frac{1}{K}\sum_{k}\sum_{e_k}\max_{e_k}\log(P(\boldsymbol{e_k}|f^{e}(G_{v_k}, y))$$

Variant subgraph

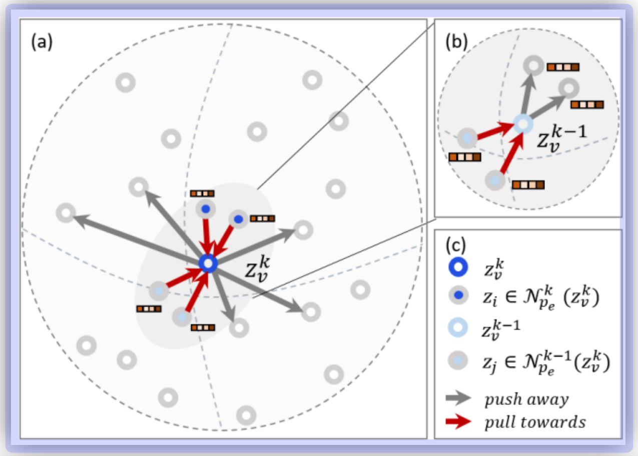
- Inter-Hierarchy Environment Augmentation

$$L_{\text{EnvCon}} = \mathbb{E}_{x\sim p_d}\left[\frac{1}{K}\sum_k L_{\text{InfoNCE}}(z, N_{p_e}^{k}(z), \tau)\right],$$

$$L_{\text{LabelCon}} = \mathbb{E}_{x\sim p_d}\left[\frac{1}{K}\sum_k L_{\text{InfoNCE}}(z, N_{p_y}(z), \tau)\right], where$$

$$N_{p_e}^{k}(z) = N_{p_e}^{k-1}(z) \cup \{z_i \mid e_i^k = e_z^k, z_i \in N(z)\},$$

$$N_{p_y}(z) = \{z_i \mid y_i = y_z, z_i \in N(z)\}$$



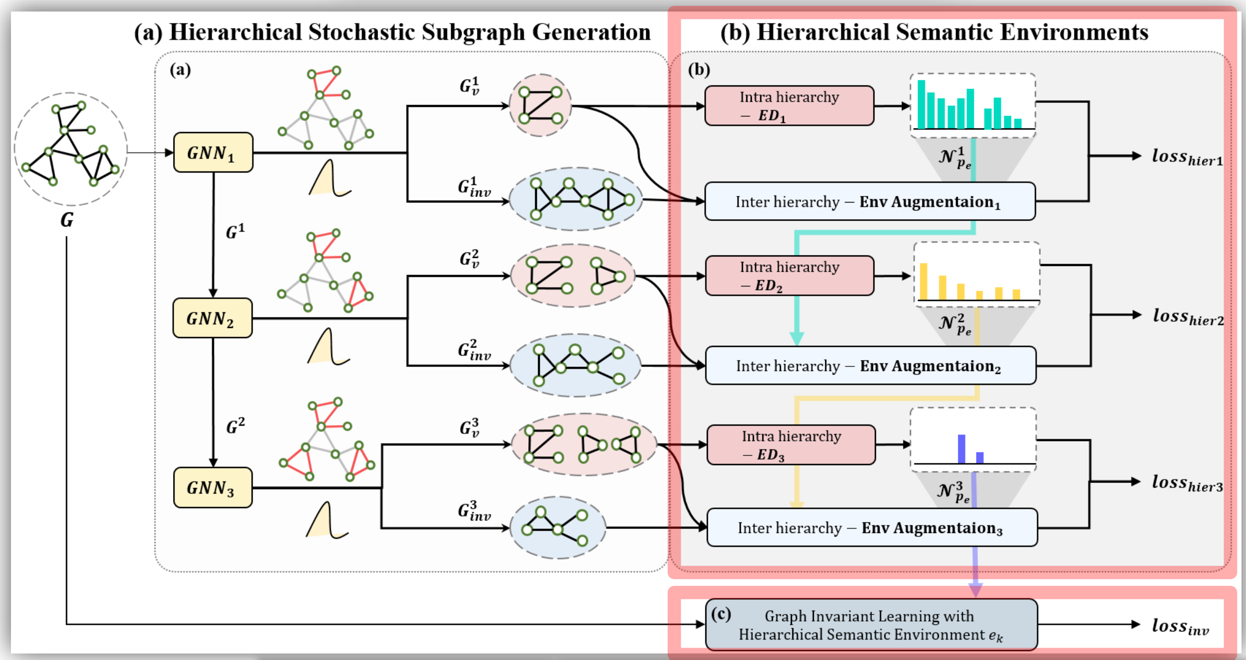

## (c) Graph invariant learning with HSE

- Hierarchical Semantic Environment Learning

$$L_{HEI} = L_{ED} + \alpha \cdot L_{\text{EnvConv}} + \beta \cdot L_{\text{LabelCon}}$$

- Robust GIL with HSE at k-th hierarchy .

$$\min_f L_{cls}^{\boldsymbol{e_k}}(f) + Var\ (L_{cls}^{\boldsymbol{e_k}}(f))\ s.t.\ \boldsymbol{e_k} = \arg\min_{\boldsymbol{e_k}} L_{HEI}$$

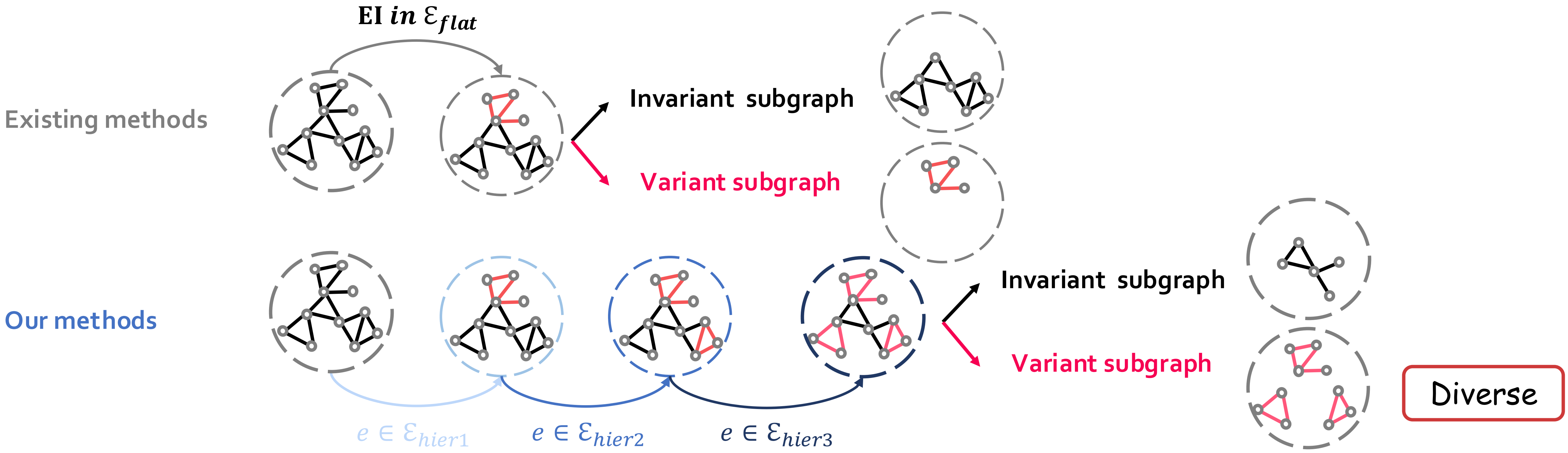


# Flat Env Inference vs Hierarchical Env Inference

**Two extreme cases in Flat Env Inference approaches:**

1. Too Few Environments → embedding space collapse.
2. Too many individual Environments → ignore dependency between environments.

*-> can also view as an adaptive parameter search problem.*



# Experiments and Results

# Results

- Compared to ERM, Euclidean-based invariant learning show degraded performance.
- Graph-based OOD methods exhibit better performance.
- Our method outperforms all baselines, especially when comes to complex dataset (*-SCA).

**Table 1. Statistics of the datasets used in experiments. The number of nodes and edges are averaged numbers among all datasets.**

| DATASETS | #TRAINING | #VALIDATION | #TESTING | #LABELS | #ENVS | #NODES | #METRICS |
|---|---|---|---|---|---|---|---|
| CMINST-SP | 40,000 | 5,000 | 15,000 | 2 | N.A. | 56.90 | ACC |
| GRAPH-SST5 | 6,090 | 1,186 | 2,240 | 5 | N.A. | 19.85 | ACC |
| IC50-ASSAY | 34,179 | 19,028 | 19,032 | 2 | 311 | 34.58 | ROC-AUC |
| IC50-SCA | 21,519 | 19,041 | 19,048 | 2 | 6,881 | 39.38 | ROC-AUC |
| IC50-SIZE | 36,597 | 17,660 | 16,415 | 2 | 190 | 37.99 | ROC-AUC |
| EC50-ASSAY | 4,540 | 2,572 | 2,490 | 2 | 47 | 39.81 | ROC-AUC |
| EC50-SCA | 2,570 | 2,532 | 2,533 | 2 | 850 | 56.84 | ROC-AUC |
| EC50-SIZE | 4,684 | 2,313 | 2,398 | 2 | 167 | 48.40 | ROC-AUC |

**Table 2. Test ROC-AUC of various models on DrugOOD benchmark datasets. The mean ± standard deviation of all models is reported as an average of 5 executions of each model. The best methods are highlighted in bold and the second best methods are underlined.**

| METHODS | IC50-ASSAY | IC50-SCA | IC50-SIZE | EC50-ASSAY | EC50-SCA | EC50-SIZE |
|---|---|---|---|---|---|---|
| ERM[51] | 71.79±0.27 | 68.85±0.62 | 66.70±1.08 | 76.42±1.59 | 64.56±1.25 | 62.79±1.15 |
| IRM[23] | 72.12±0.49 | 68.69±0.65 | 66.54±0.42 | 76.51±1.89 | 64.82±0.55 | 63.23±0.56 |
| V-REX[25] | 72.05±1.25 | 68.92±0.98 | 66.33±0.74 | 76.73±2.26 | 62.83±1.20 | 59.27±1.65 |
| EIIL[26] | 72.60±0.47 | 68.45±0.53 | 66.38±0.66 | 76.96±0.25 | 64.95±1.12 | 62.65±1.88 |
| IB-IRM[24] | 72.50±0.49 | 68.50±0.40 | 66.64±0.28 | 76.72±0.98 | 64.43±0.85 | 64.10±0.61 |
| GREA [36] | 72.77±1.25 | 68.33±0.32 | 66.16±0.46 | 72.44±2.55 | <u>67.98±1.00</u> | 63.93±3.01 |
| CIGAV1 [33] | 72.71±0.52 | 69.04±0.86 | 67.24±0.88 | 78.46±0.45 | 66.05±1.29 | <u>66.01±0.84</u> |
| CIGAV2 [33] | <u>73.17±0.39</u> | <u>69.70±0.27</u> | <u>67.78±0.76</u> | - | - | - |
| MOLEOOD [31] | 71.38±0.68 | 68.02±0.55 | 66.51±0.55 | 73.25±1.24 | 66.69±0.34 | 65.09±0.90 |
| GALA [34] | - | - | - | <u>79.24±1.36</u> | 66.00±1.86 | <u>66.01±0.84</u> |
| OURS | **74.01±0.11** | **70.72±0.30** | **68.64±0.23** | **80.82±0.21** | **69.73±0.21** | **66.87±0.38** |

# Results

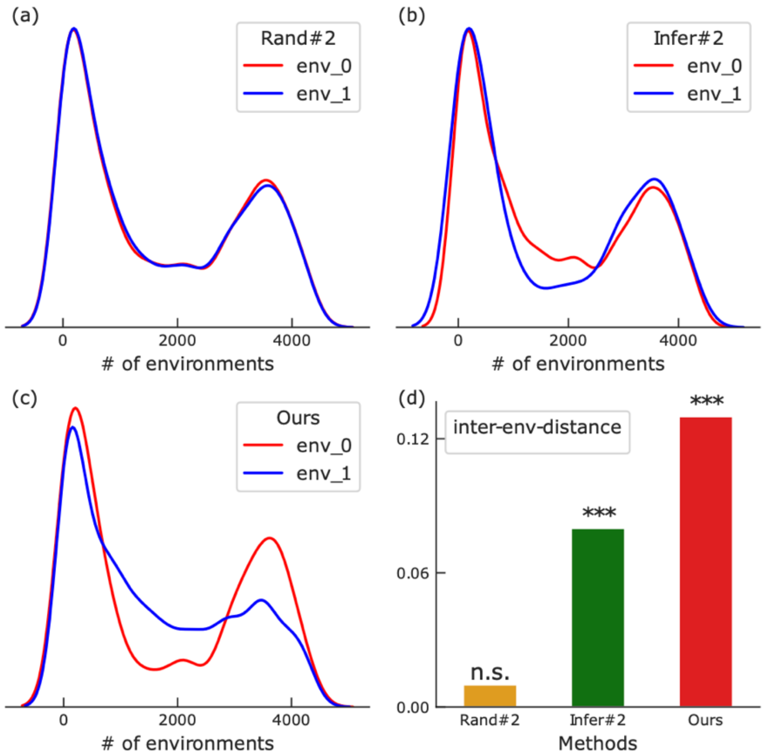
## ➢ Discussion on Diversity of Inferred Environments

(a) Random sampling methods.

(b) Flat environment inference methods

**(c) Our hierarchical environment inference methods**

(d) K-S test to calculate the diversity of three methods.




## ➢ Case study of Hierarchical Semantic Environment.

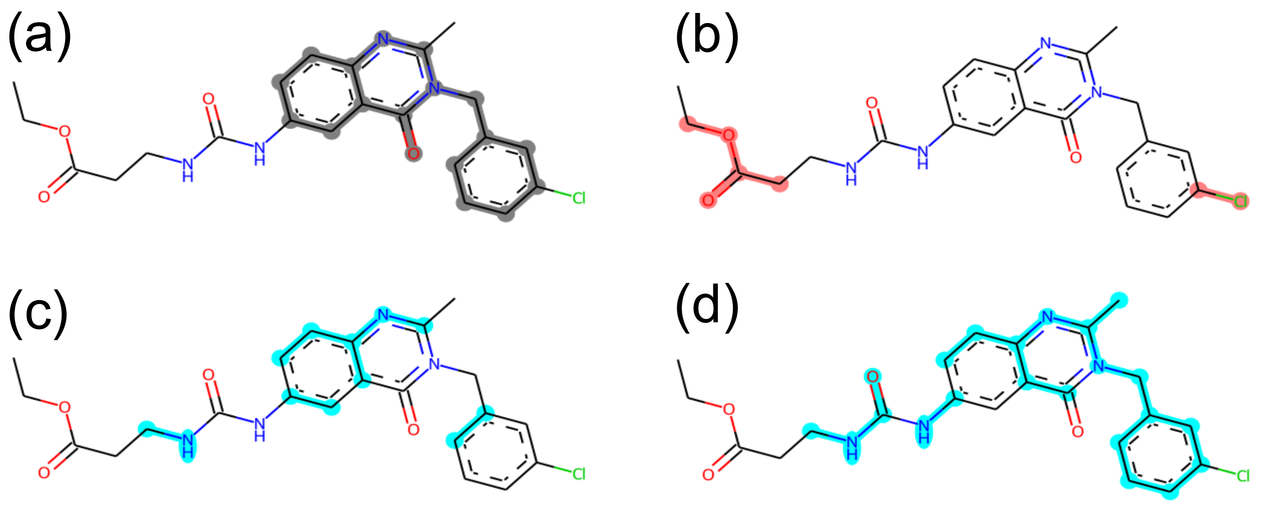
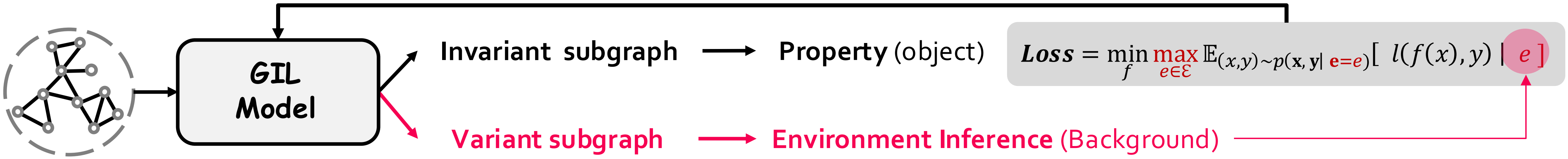
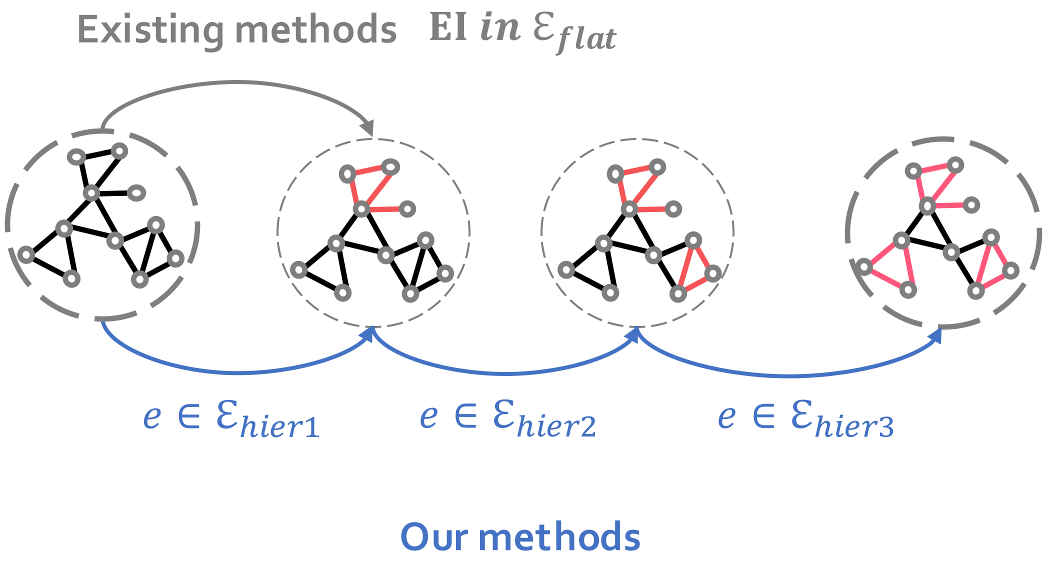



Figure 6. (a) Grey: Scaffold. (b) Red: Functional group (Chlorobenzene), and structural alert COC(=O)C from PubChem. (c, d) Blue: Learned variant subgraphs from the first and second hierarchies (the rest of the graph is considered learned invariant).

# Conclusion




**Sufficiency** **Efficiency** **Interpretability**

- Explicit Subgraph Modeling.
- Diverse environment inference.
- Env diversification loss instead of clustering methods.

# Thank you for listening!

# Results

- **Effect of Hierarchical Semantic Environments (Table 4)**
  - Environment Inference
  - Hierarchical Environment Inference
- **Sensitive Analysis on Hierarchy (Table 5)**

**Table 4. Ablation studies on IC50-SCA and IC50-SIZE datasets.**

| METHODS | IC50-SCA | IC50-SIZE |
|---|---|---|
| w/ $\text{env}_{\#\text{non-infer(rand)=2}}$ | 68.54±0.64 | 67.63±0.33 |
| w/ $\text{env}_{\#\text{non-infer=real}}$ | 68.77±0.72 | 67.60±0.32 |
| w/ $\text{env}_{\#\text{infer}=2}$ | 69.14±0.80 | 67.55±0.34 |
| w/ $\text{env}_{\#\text{infer}=\text{real}}$ | 69.08±0.64 | 67.74±0.13 |
| **w/ $\text{env}_{\#\text{hier-infer}}$ (OURS)** | **70.72±0.30** | **68.64±0.23** |

**Table 5. Sensitivity analysis on generated environments.**
(#$e_p$ denotes the number of provided environments in datasets.)

| CONFIGURATIONS | IC50-SCA | IC50-SIZE |
|---|---|---|
| #real = [#$e_p$] | 69.08±0.64 | 67.60±0.32 |
| #env = [5] | 69.35±0.67 | 67.70±0.50 |
| #env = [2] | 69.14±0.80 | 67.73±0.64 |
| #env=[#$e_p \to$ #$e_p/2 \to 5$] | 70.12±0.14 | 68.62±0.34 |
| **#env=[#$e_p \to$ #$e_p/2 \to 2$]** | **70.72±0.30** | **68.64±0.23** |