Learning Causality for Modern Machine Learning

With a focus on the Out-of-Distribution Generalization challenge

Yongqiang Chen

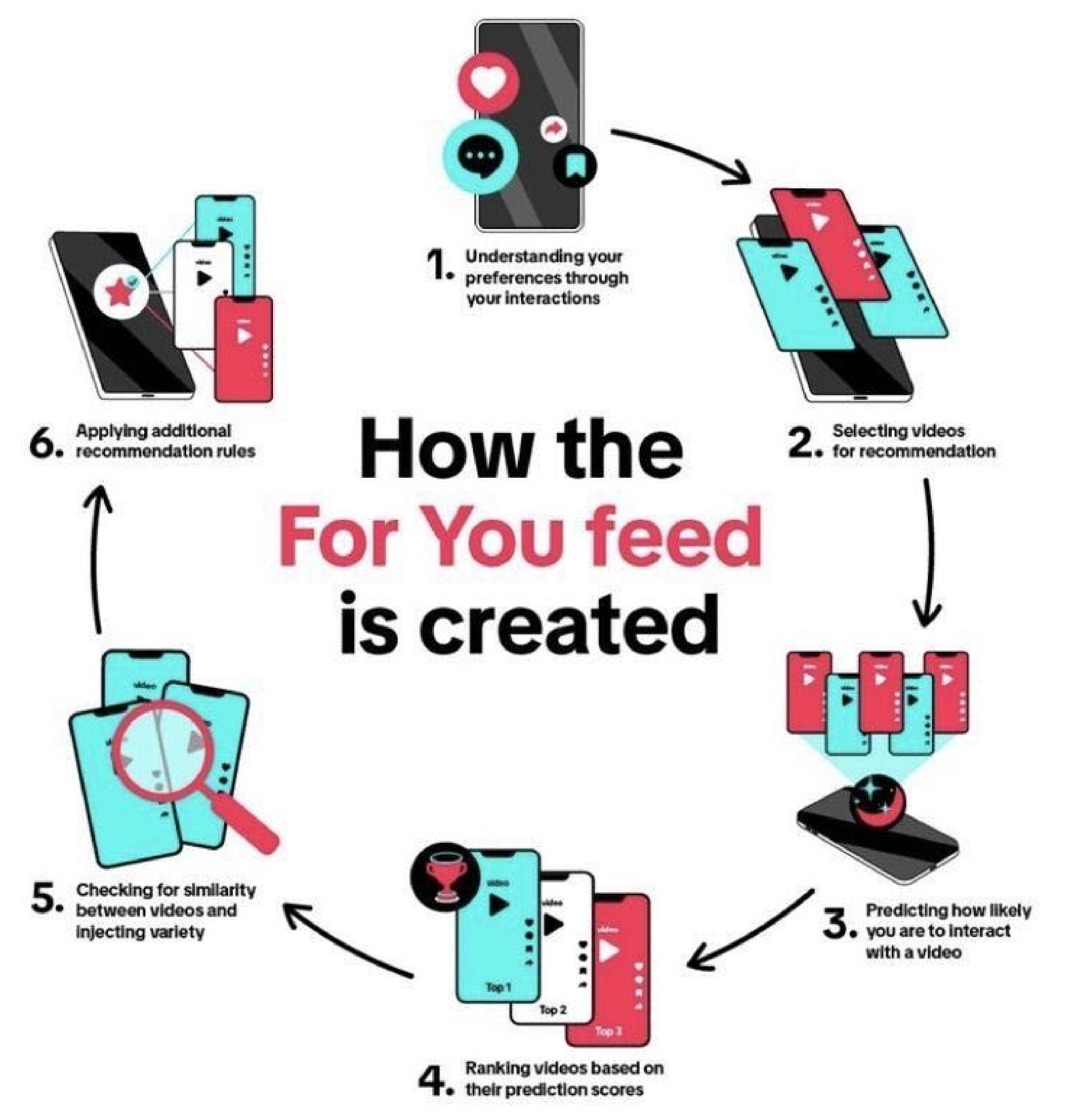
The Chinese University of Hong Kong

Machine Learning Systems are Everywhere

In our modern daily life, machine learning (ML) systems are everywhere.

Face ID

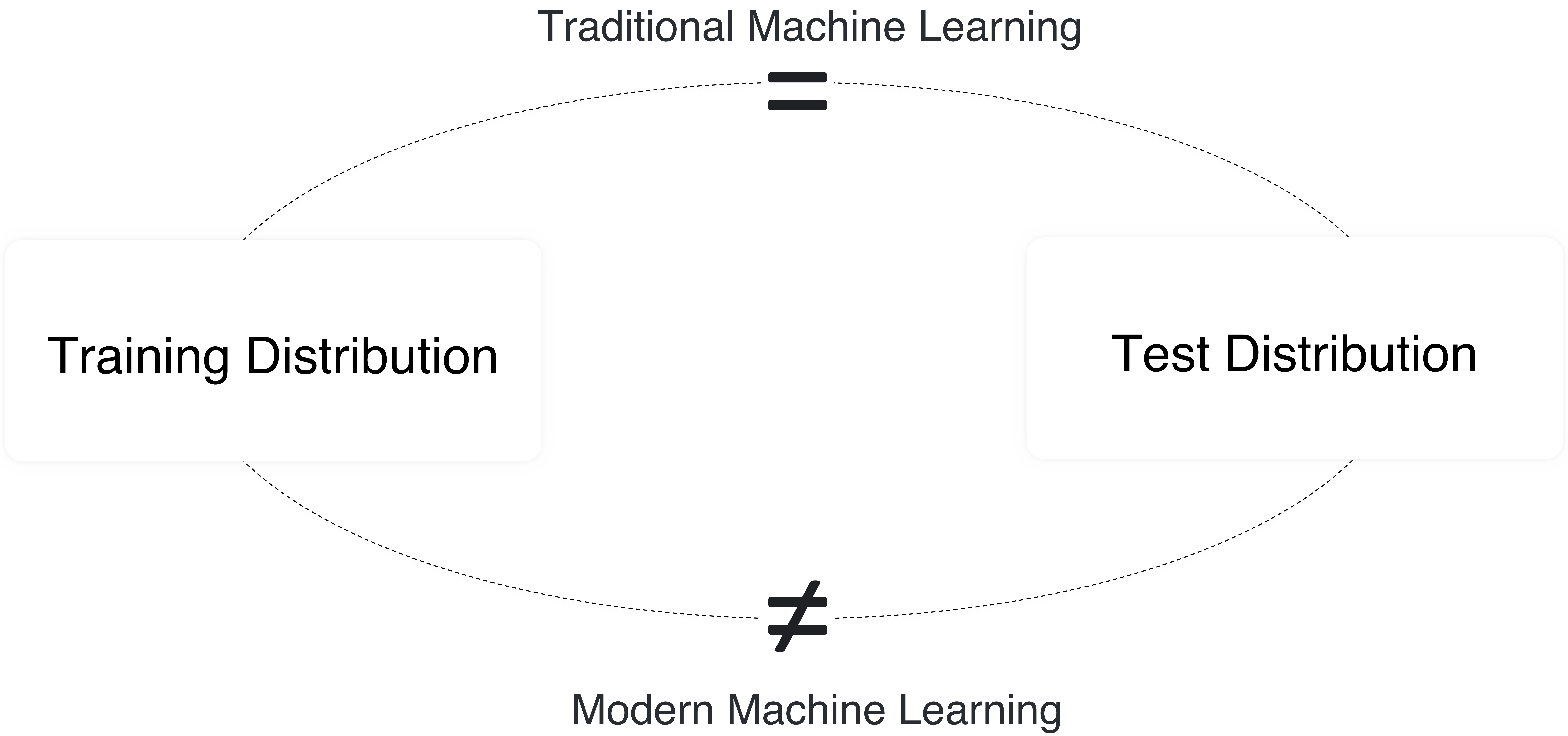
Autonomous Driving



Recommenders

From Traditional ML to Modern ML

There is a paradigm shift challening the **fundamental assumption** in ML:



The Out-of-Distribution Generalization Failure

Distribution shifts/changes are everywhere. Existing ML models deployed in the wild can fail short in generalizing to **new domains/environments**, or across subpopulations.

Training Environment

Waymo Open Challenge

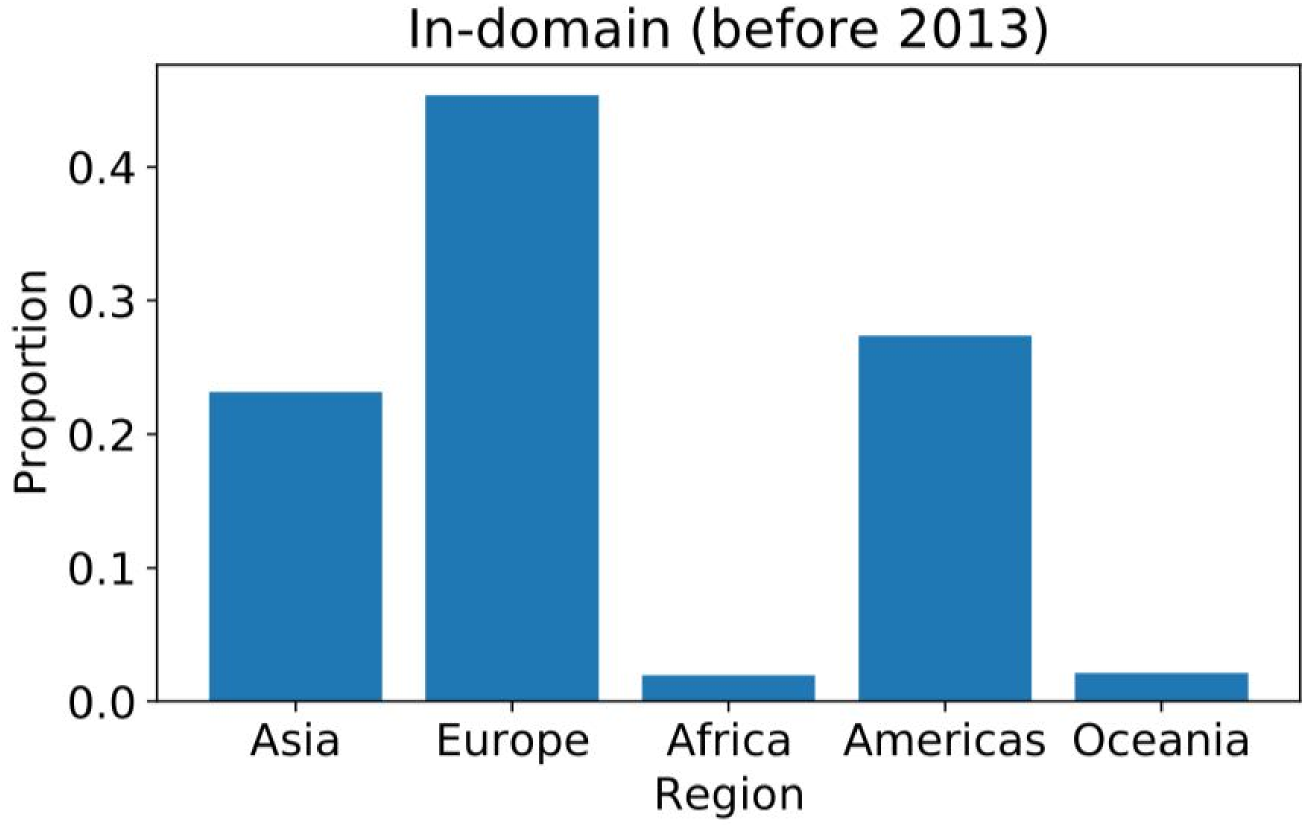
Various Test Environments

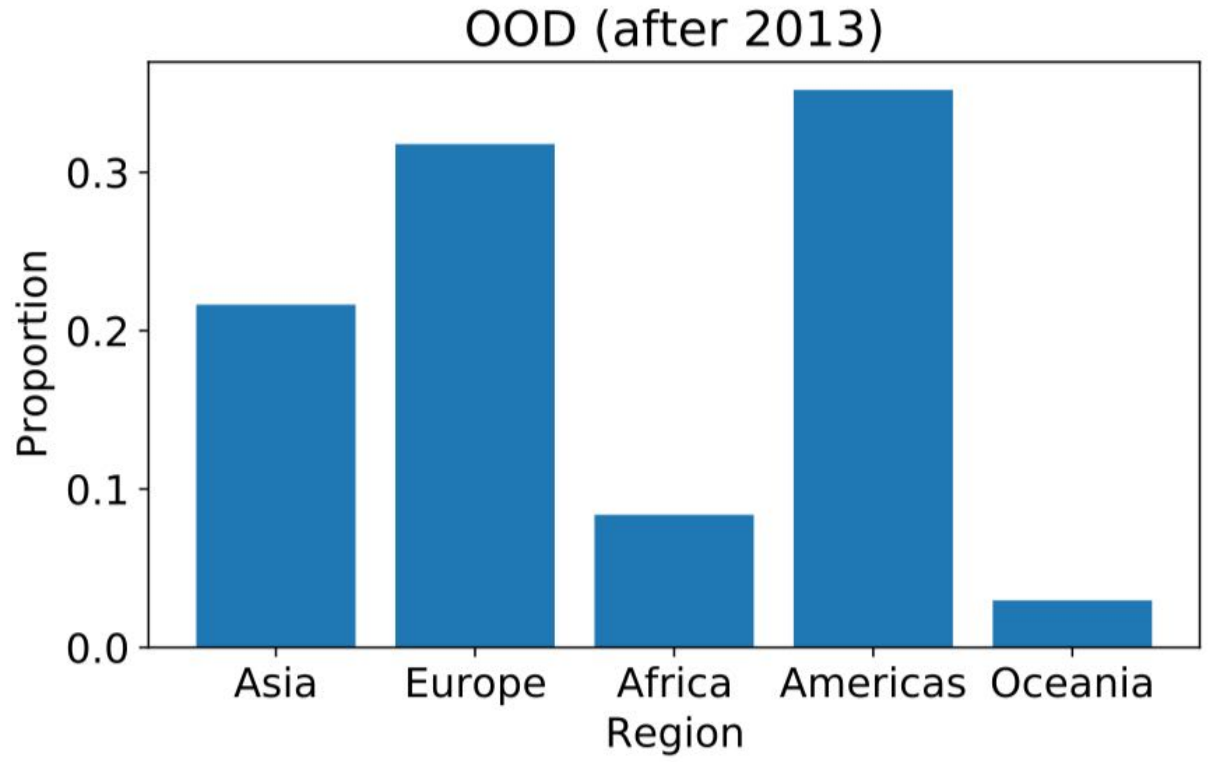
(Beery et al., 2018; Arjovsky et al., 2019; DeGrave et al. 2021; Koh et al., 2021)

The Out-of-Distribution Generalization Failure

Distribution shifts/changes are everywhere. Existing ML models deployed in the wild can fail short in generalizing to new domains/environments, or across **subpopulations**.

	Train			Test	
Satellite Image (x)					
Year / Region (d)	2002 / Americas	2009 / Africa	2012 / Europe	2016 / Americas	2017 / Africa
Building / Land Type (y)	shopping mall	multi-unit residential	road bridge	recreational facility	educational institution



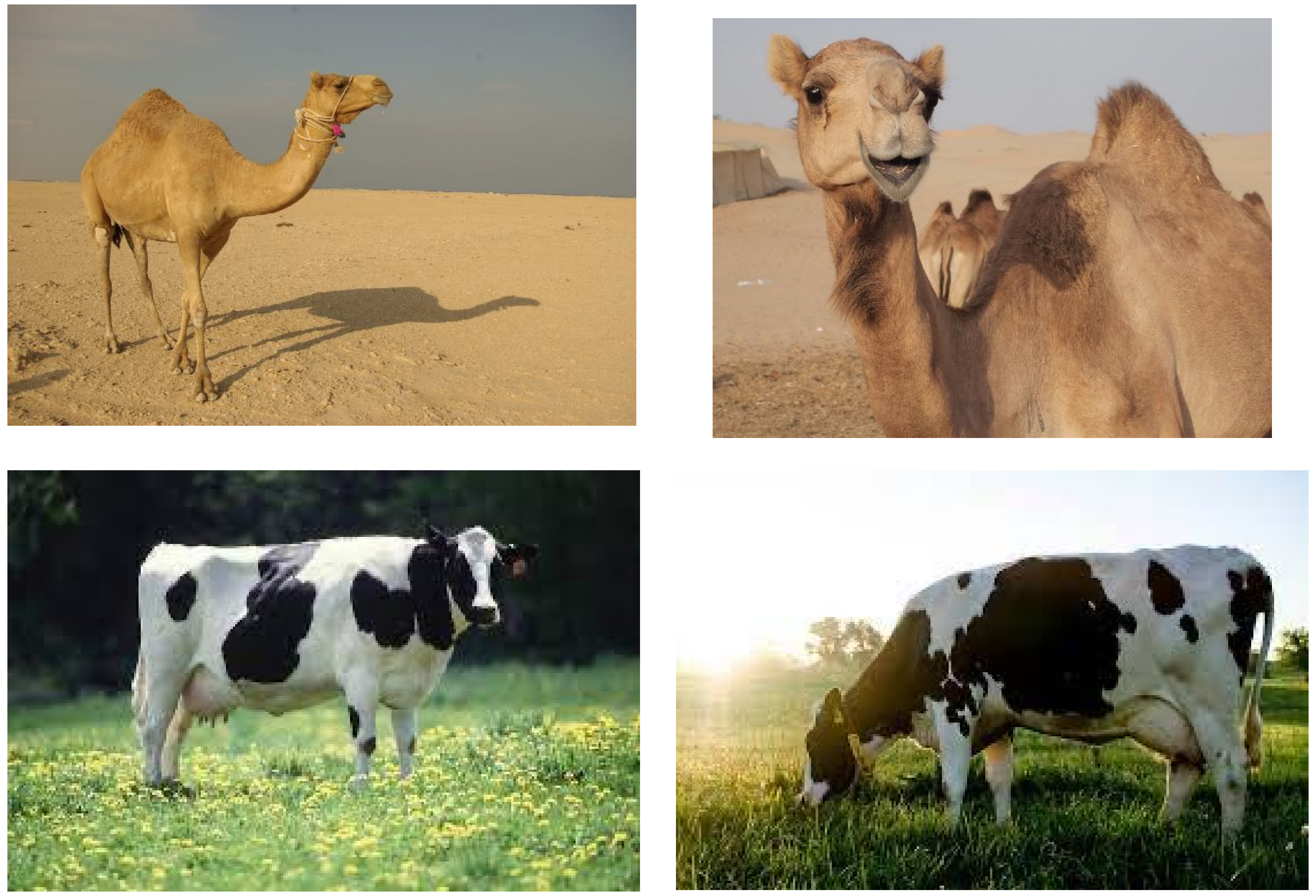


(Beery et al., 2018; Arjovsky et al., 2019; DeGrave et al. 2021; Koh et al., 2021)

How to Handle Changes?

What is unchanged across changes?

Train Environment

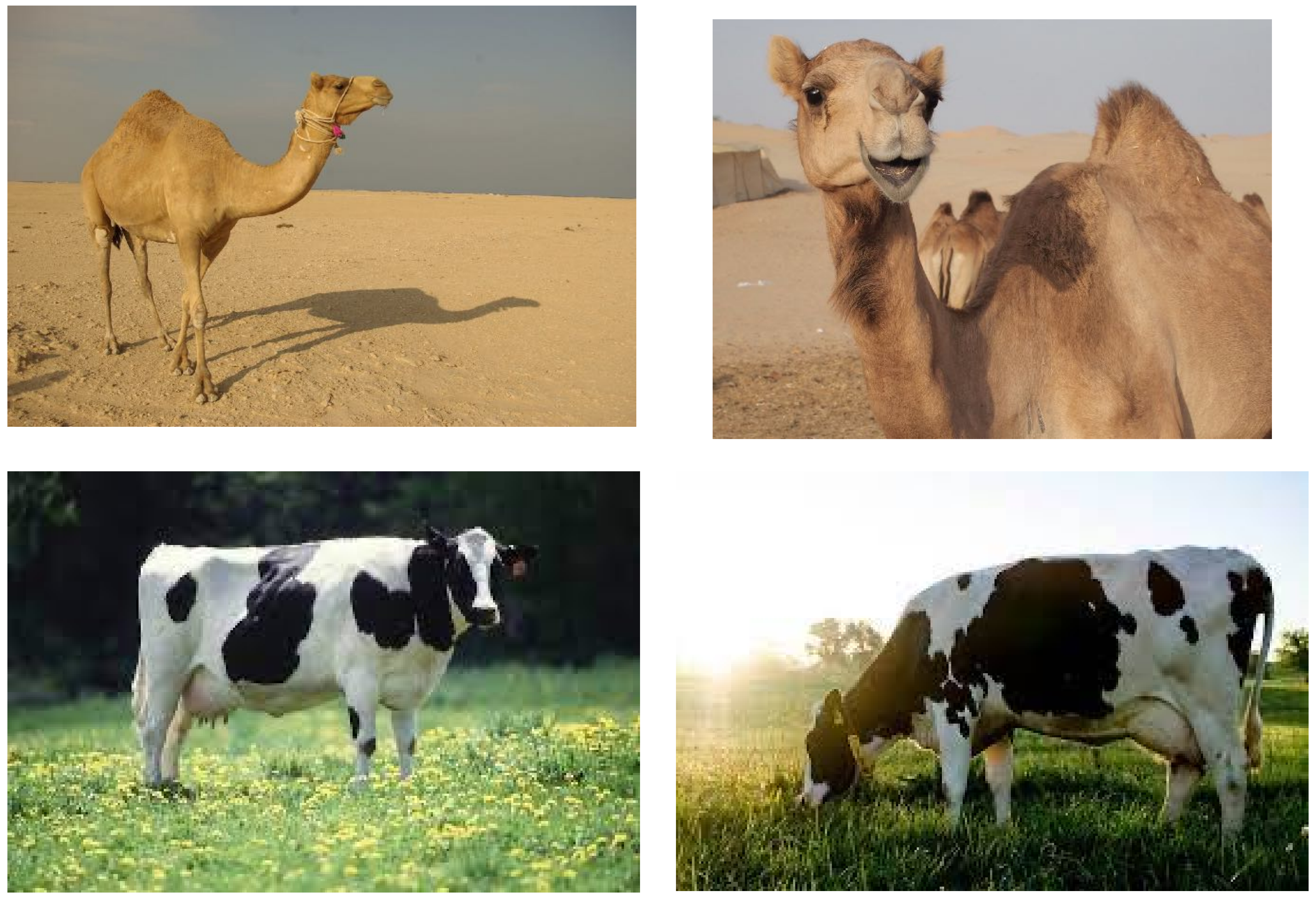


Test Environment

How to Handle Changes?

Causal Invariance Principle: The causal mechanism generating the target variable from its direct parents is independent from the changes.

Train Environment

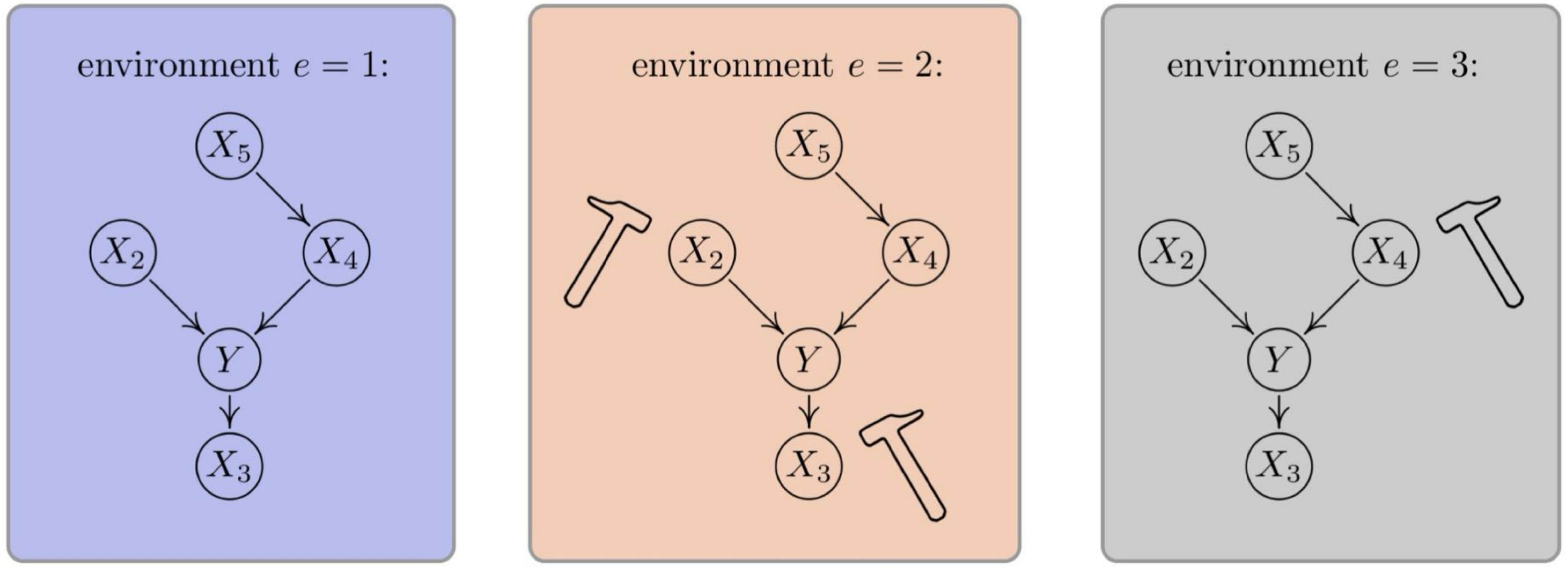


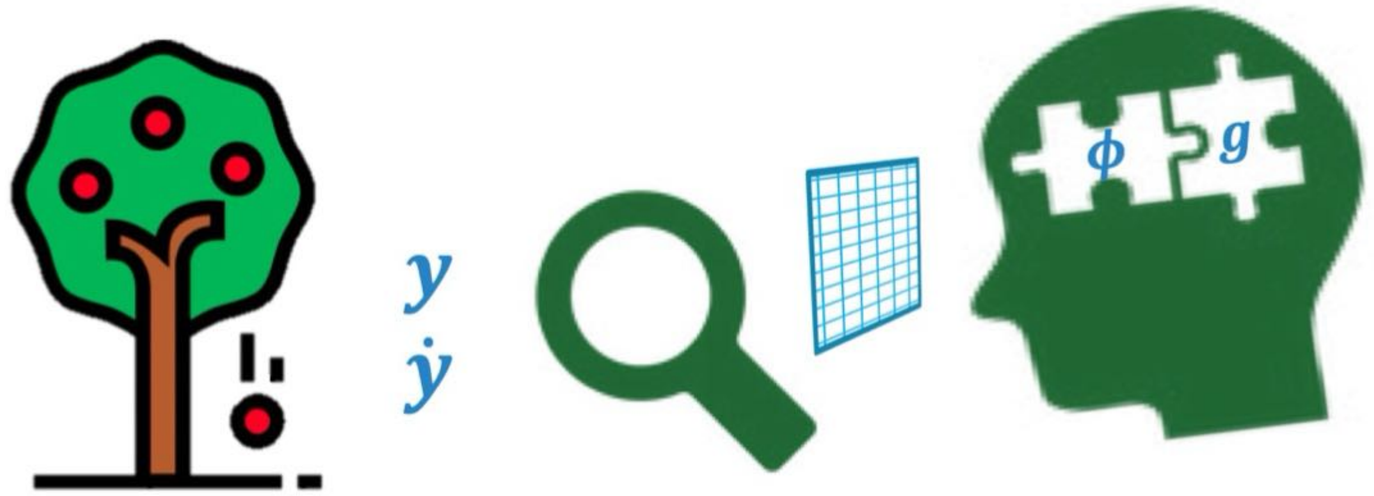
Test Environment

P(Label **|** Animal shapes etc.**) is invariant!**

How to Handle Changes?

Leveraging the principle of **causal invariance**, we can seek for

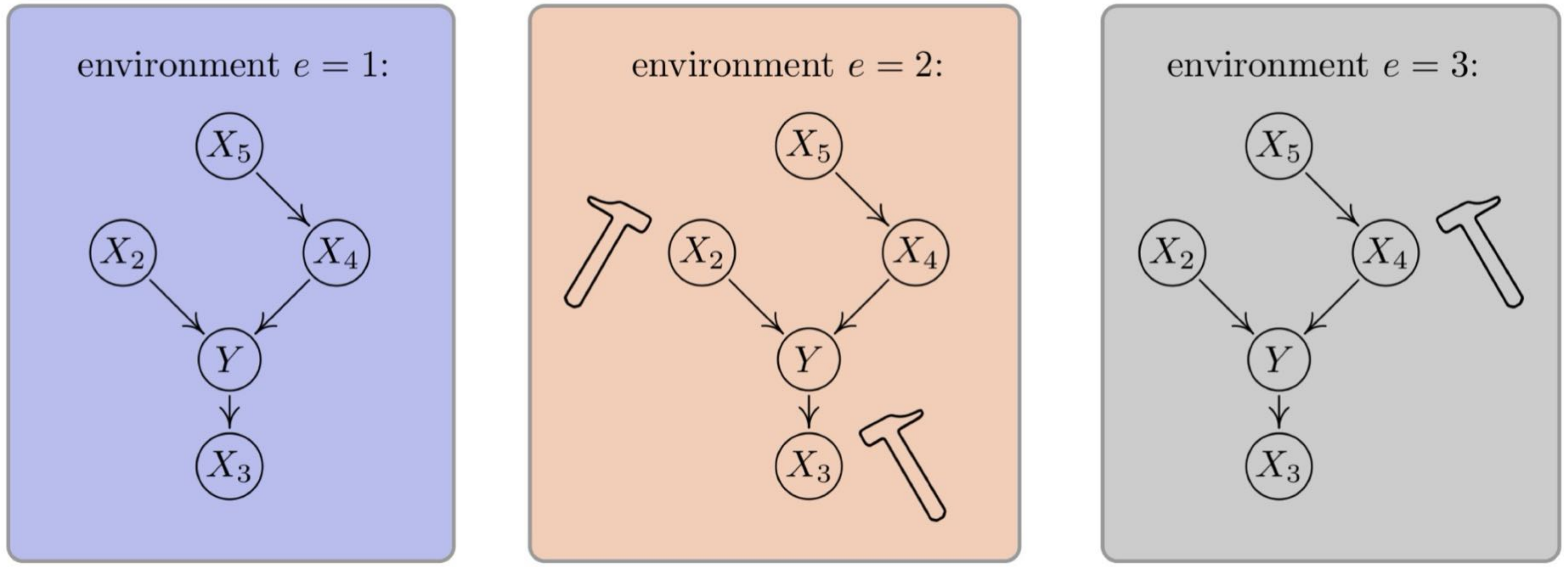


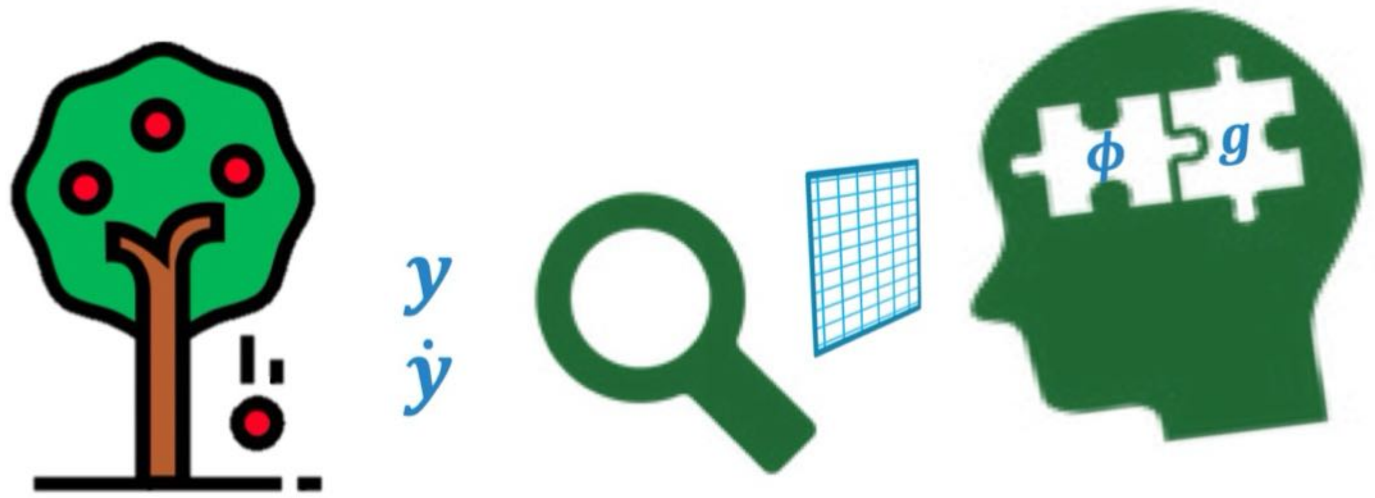


(Peters et al., 2015; Arjovsky et al., 2019; Bottou et al., 2021;)

How to Handle Changes?

Leveraging the principle of **causal invariance**, we can seek for





Invariant predictor $f = w \circ \phi$ implemented with **invariant risk minimization (IRM)**:

$$\min_{f=w\circ\varphi} \sum_{e\in\mathcal{E}_{\text{tr}}} \mathcal{L}_e(w\circ\varphi),$$

$$\text{s.t. } w \in \arg\min_{\bar{w}} \mathcal{L}_e(\bar{w}\circ\varphi), \ \forall e \in \mathcal{E}_{\text{tr}},$$

that is simultaneously optimal across different environments/domains.

(Peters et al., 2015; Arjovsky et al., 2019; Bottou et al., 2021;)

Learning Causality for Modern Machine Learning

Traditional ML assumes train and test data are **iid.**, i.e., independently sampled from an identical distribution, while data is often **OOD**, i.e., out-of-distribution, in real-world applications.

Objectives	**Causal Representation Learning on Graphs**: [NeurIPS'22 Spotlight, NeurIPS'23a]
Implications	**Useful Properties** of the Causal Representations: OOD Generalizability [NeurIPS'22, 23a], Adversarial Robustness [ICLR'22], Interpretability [ICML'24a]
Realizations	**Optimization & Feature Learning** schemes for Causal Representation Learning: [ICLR'23a, NeurIPS'23b]

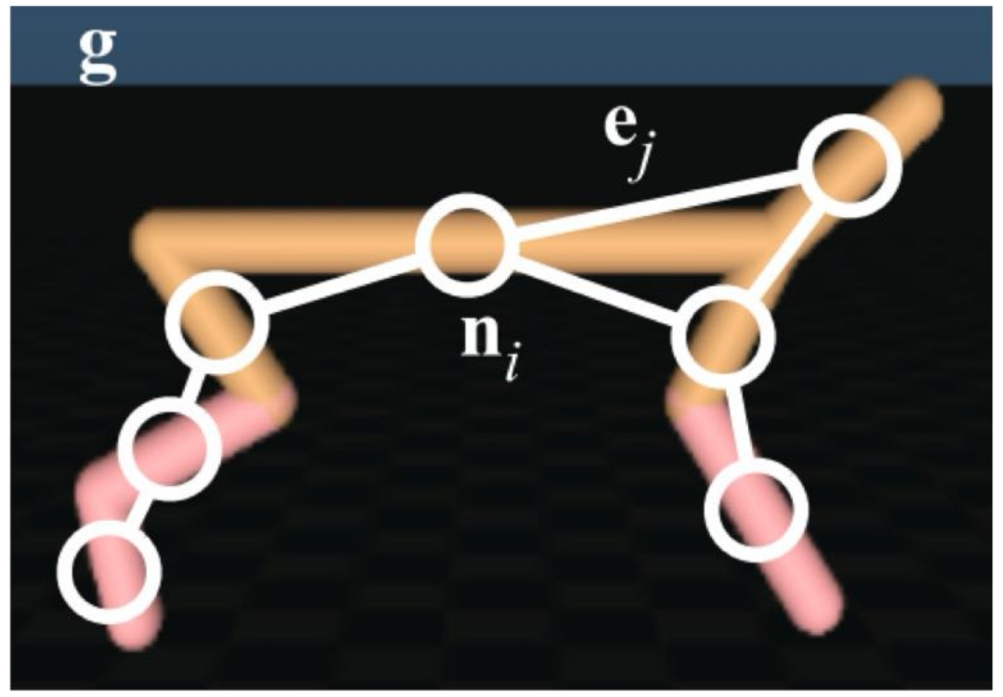
Learning Causality for Modern Machine Learning

Traditional ML assumes train and test data are **iid.**, i.e., independently sampled from an identical distribution, while data is often **OOD**, i.e., out-of-distribution, in real-world applications.

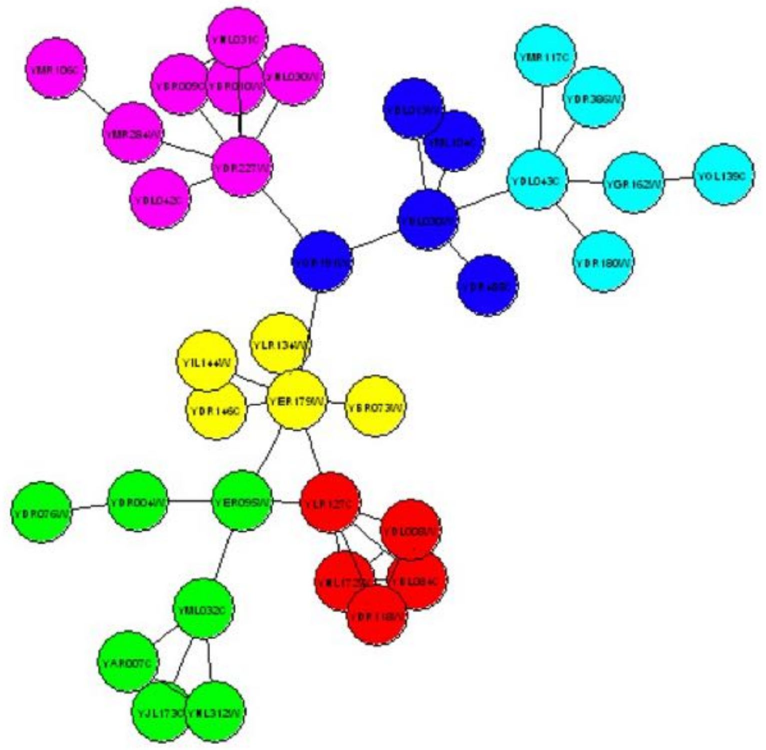
Objectives	**Causal Representation Learning on Graphs**: [NeurIPS’22 Spotlight, NeurIPS’23a]
Implications	**Useful Properties** of the Causal Representations: OOD Generalizability [NeurIPS’22, 23a], Adversarial Robustness [ICLR’22], Interpretability [ICML’24a]
Realizations	**Optimization & Feature Learning** schemes for Causal Representation Learning: [ICLR’23a, NeurIPS’23b]

Causal Representation Learning on Graphs

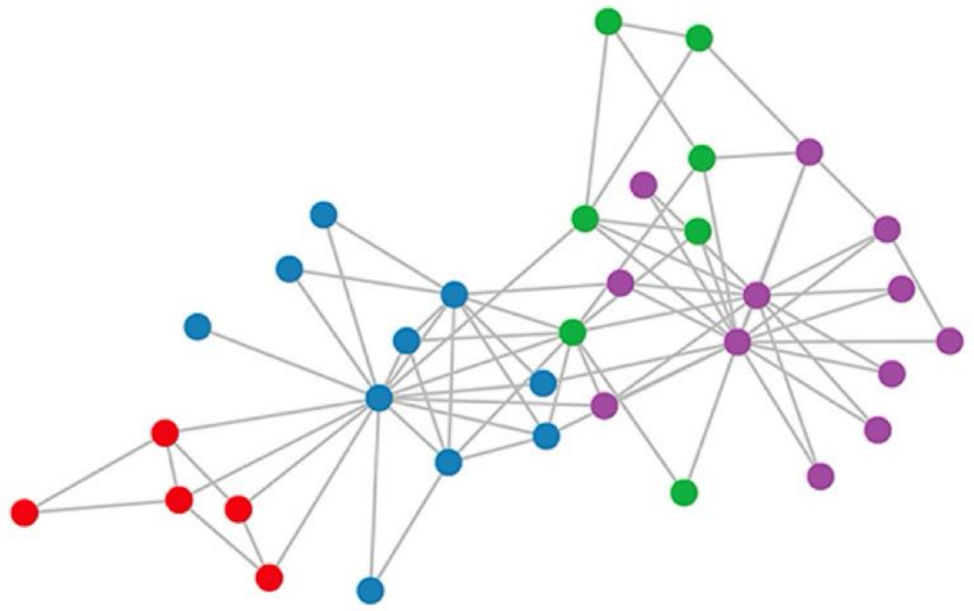
We seek to derive general causal representation learning objectives from a general view, i.e., graphs.



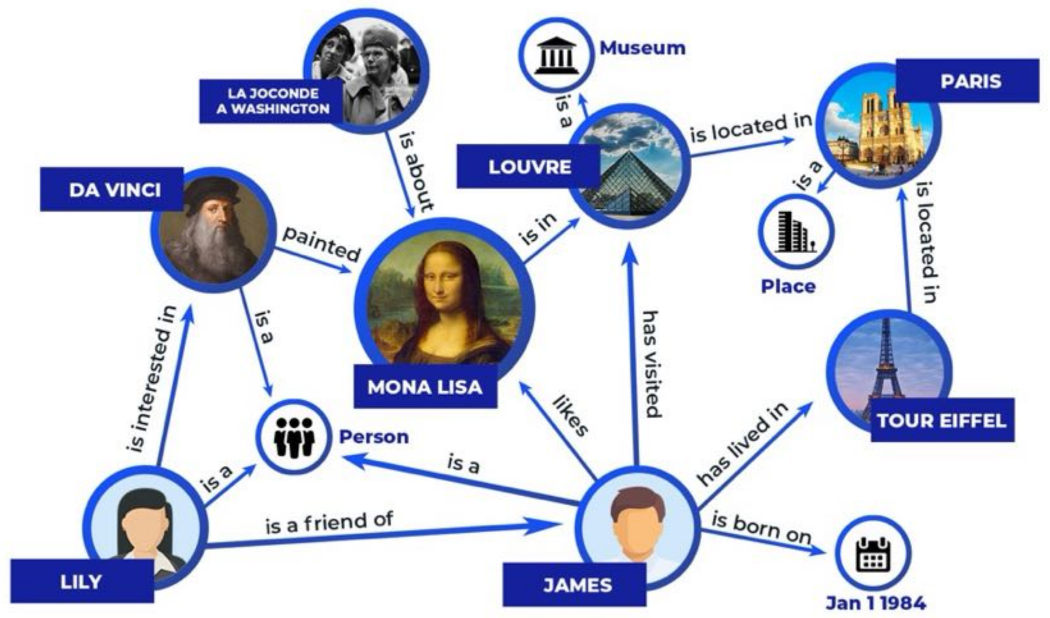
Model & Inference over the physical world



Protein Interaction Predictions

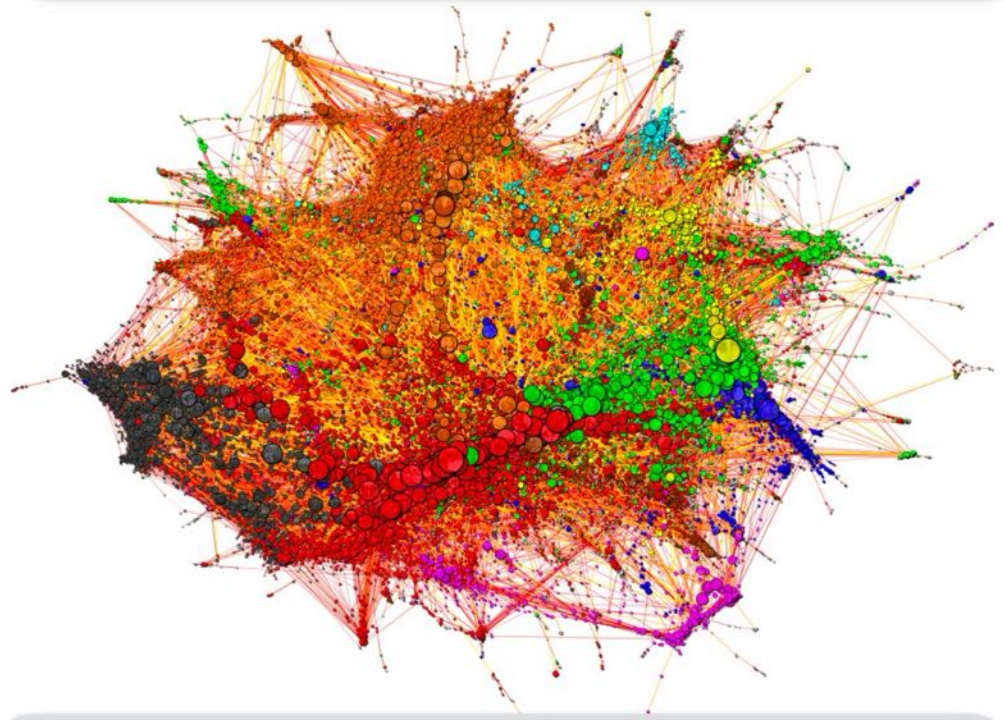


Social Network Analysis



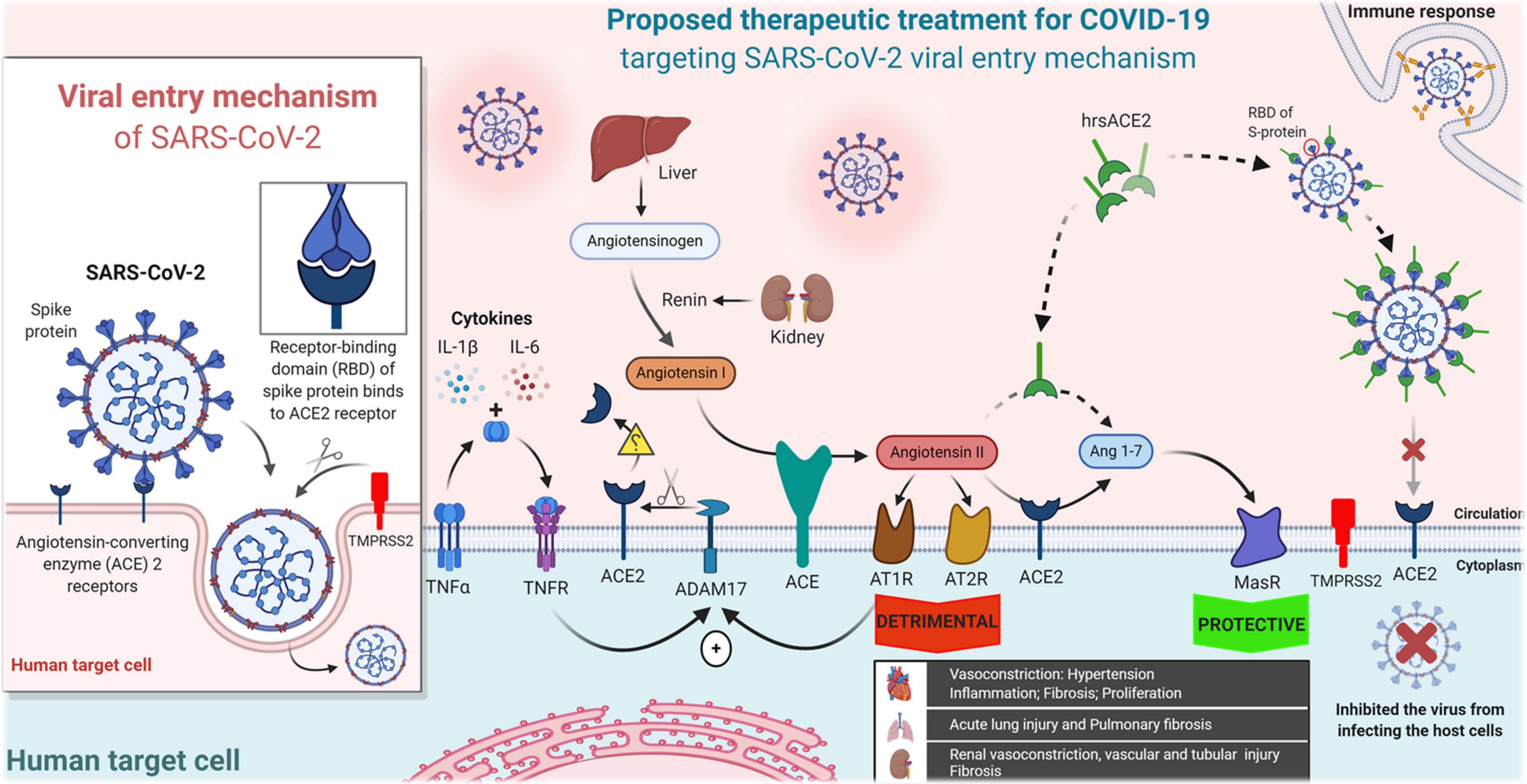
Knowledge Graph Completion & Analysis

Besides, GNNs can also process structures like image and text…



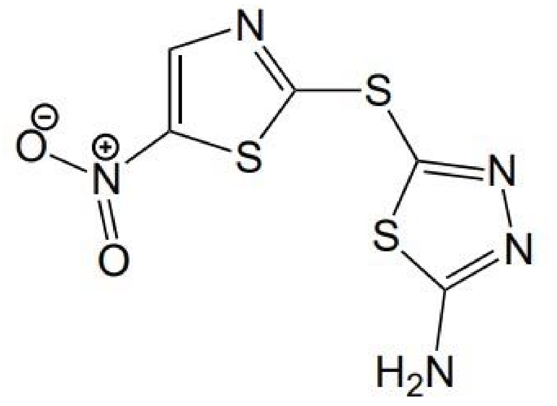
Recommender Systems

Out-of-Distribution Generalization on Graphs

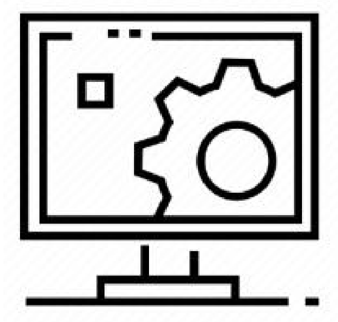


(El-Aziz et al., 2020)

Out-of-Distribution Generalization on Graphs



Compound

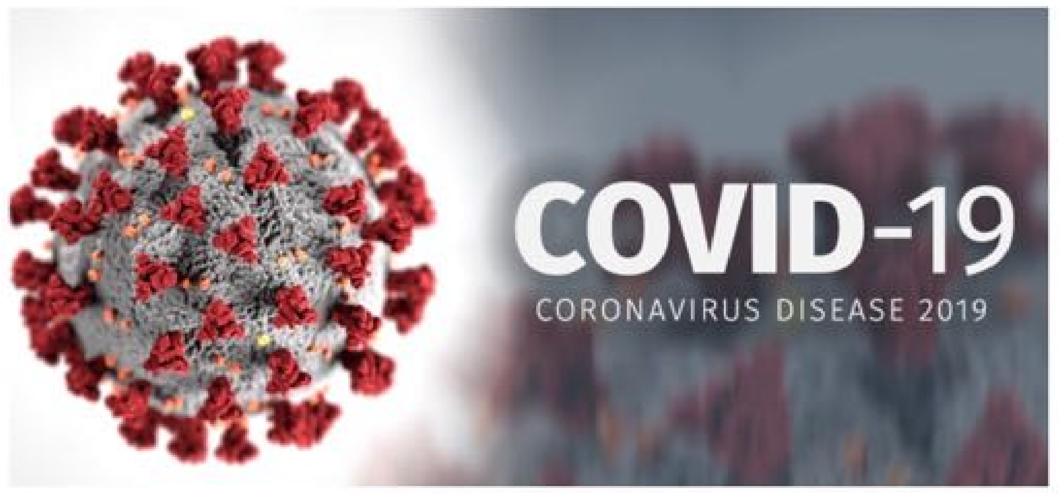


Virtual screening model

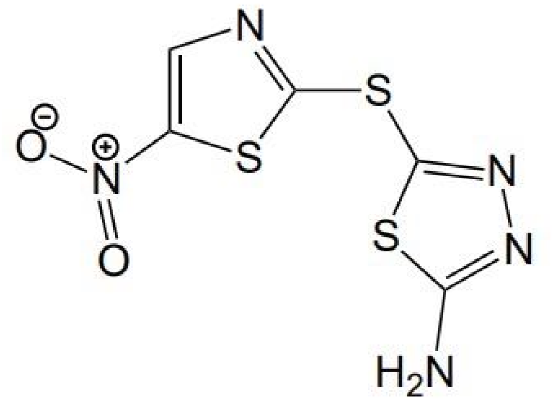
Prediction: good!

Experiments

X



Out-of-Distribution Generalization on Graphs



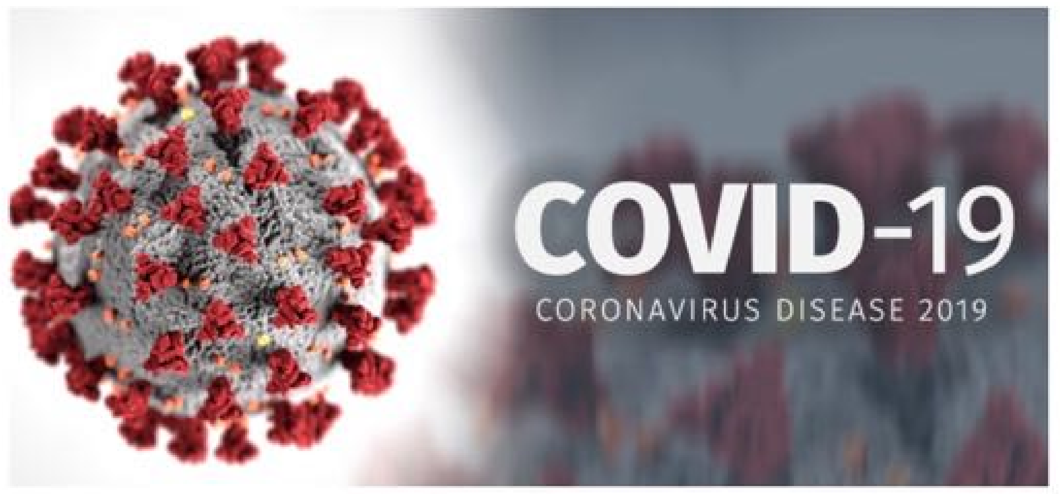
Compound

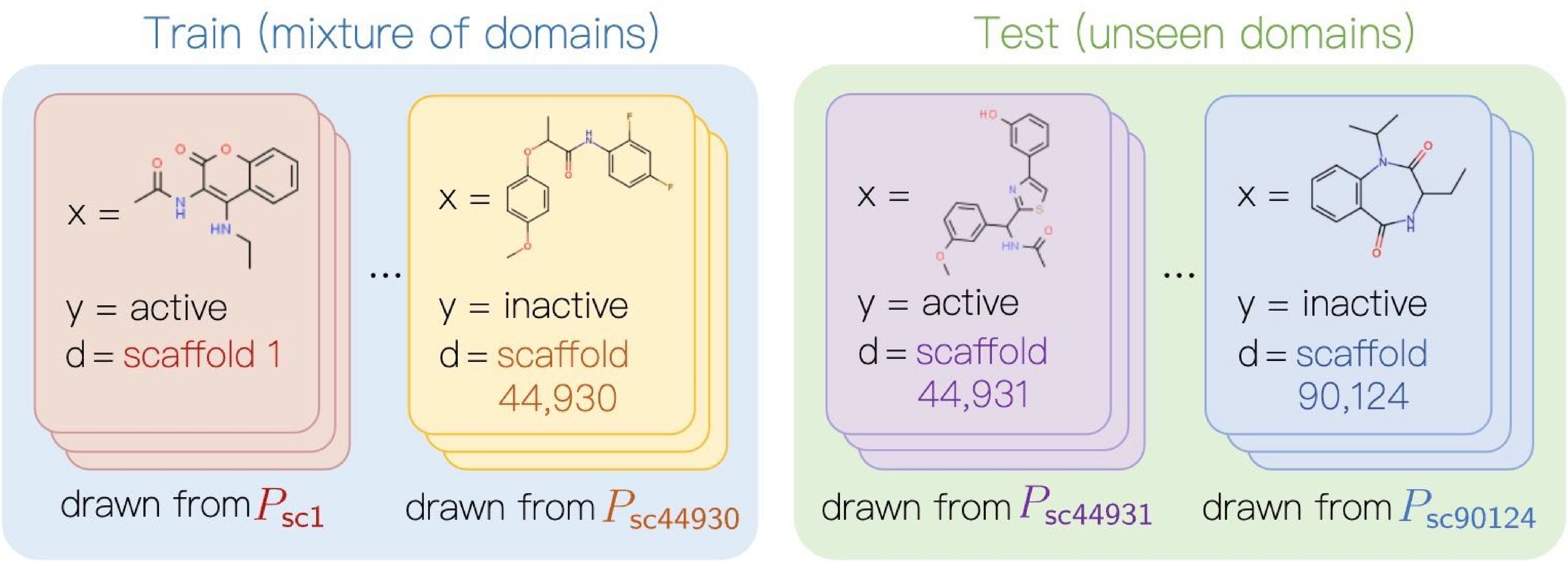
Virtual screening model

Prediction: good!

Experiments

X



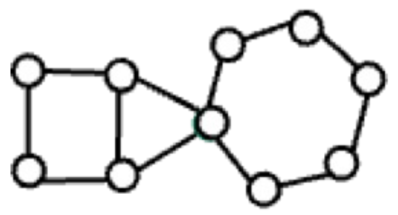


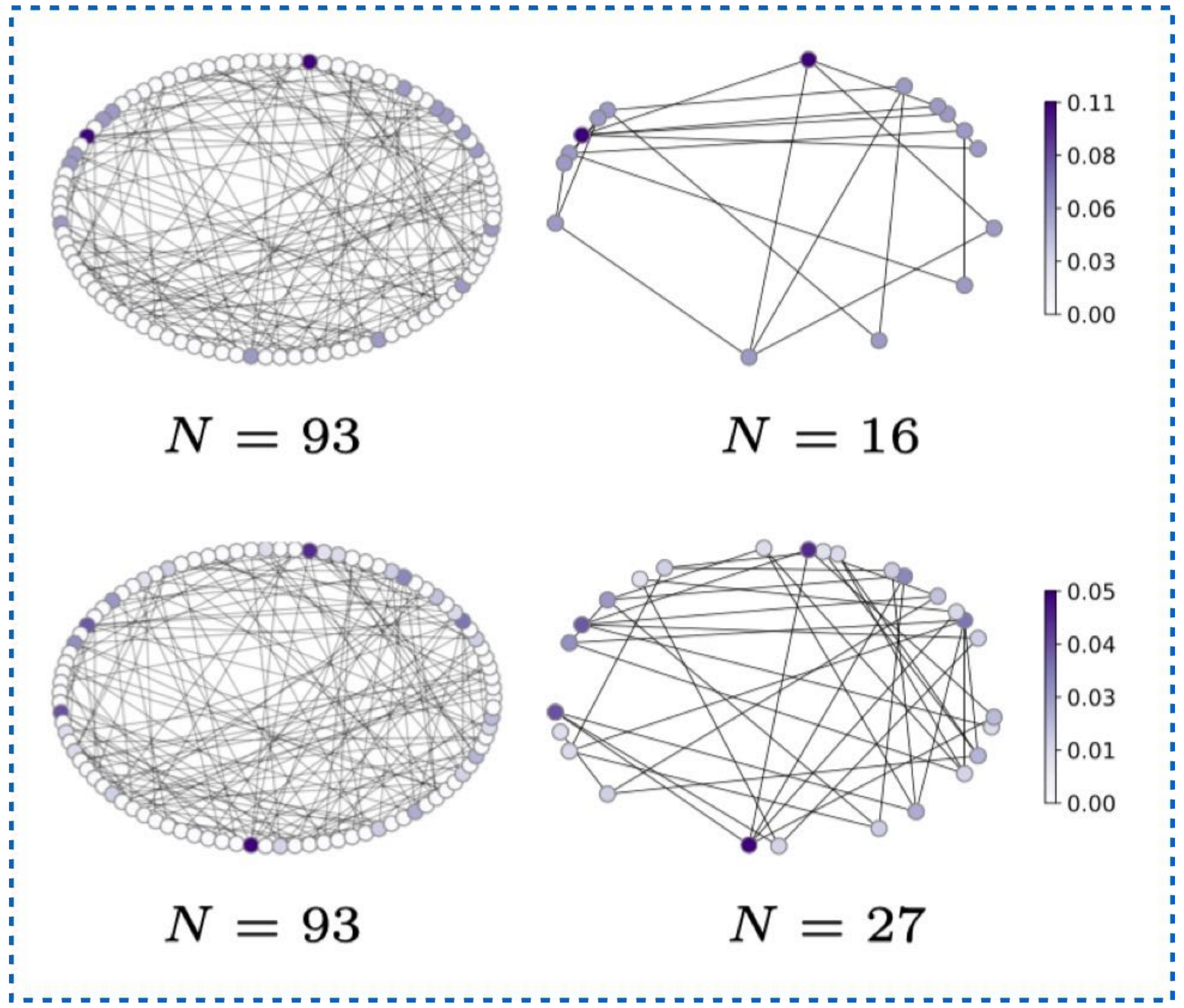
average precision = 27.2%

Dataset	Gap
DrugOOD-lbap-core-ic50-assay	17.64
DrugOOD-lbap-core-ic50-scaffold	17.60
DrugOOD-lbap-core-ic50-size	24.87
DrugOOD-lbap-refined-ic50-assay	16.55
DrugOOD-lbap-refined-ic50-scaffold	15.78
DrugOOD-lbap-refined-ic50-size	22.58
DrugOOD-lbap-general-ic50-assay	15.32
DrugOOD-lbap-general-ic50-scaffold	17.60
DrugOOD-lbap-general-ic50-size	23.72
DrugOOD-sbap-core-ic50-protein	21.84
DrugOOD-sbap-core-ic50-protein-family	17.68
DrugOOD-sbap-refined-ic50-protein	17.36
DrugOOD-sbap-refined-ic50-protein-family	15.82
DrugOOD-sbap-general-ic50-protein	16.86
DrugOOD-sbap-general-ic50-protein-family	10.58

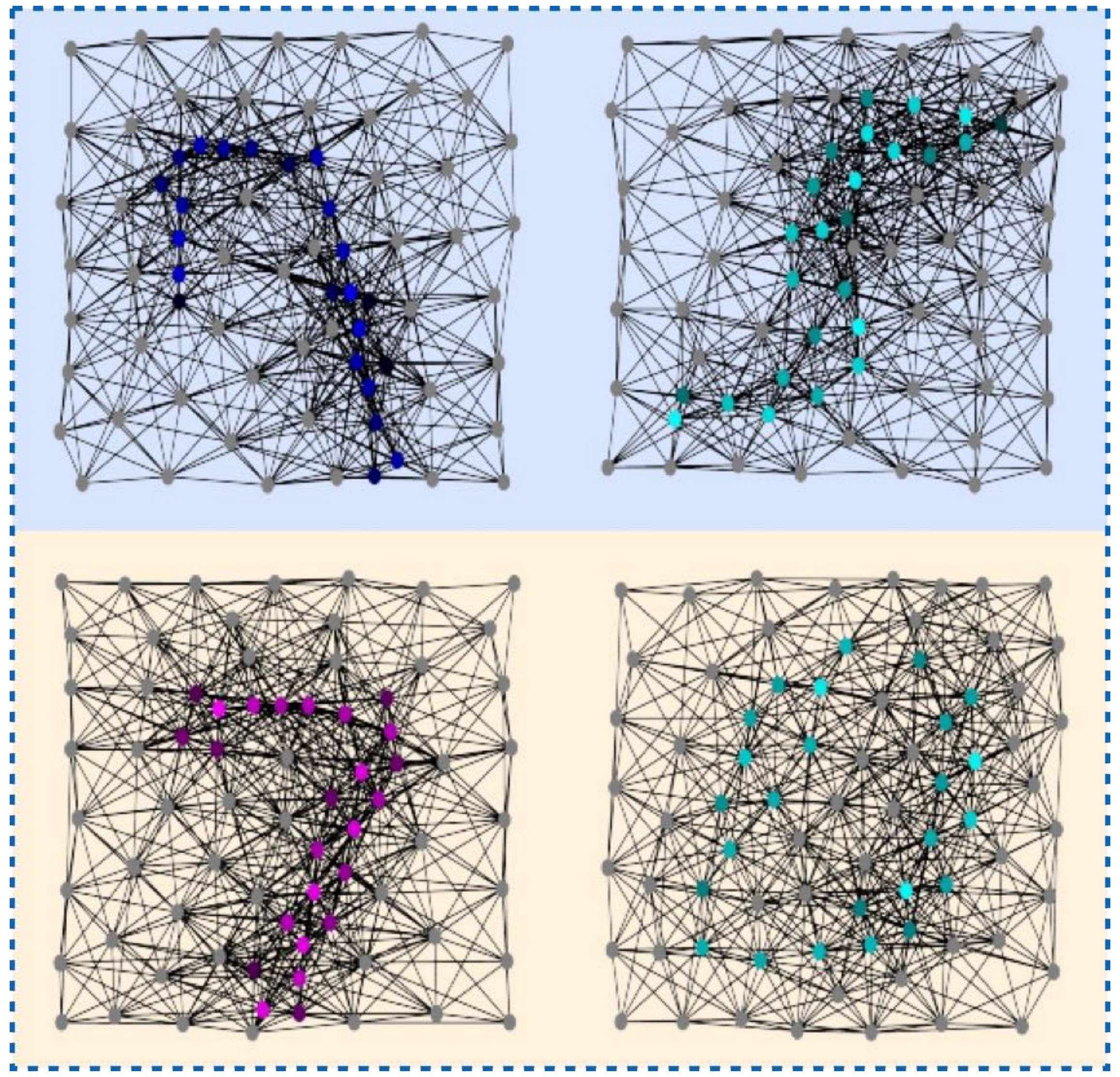
Out-of-Distribution Generalization on Graphs

OOD generalization on graphs is fundamentally more **challenging** than that on Euclidean data:

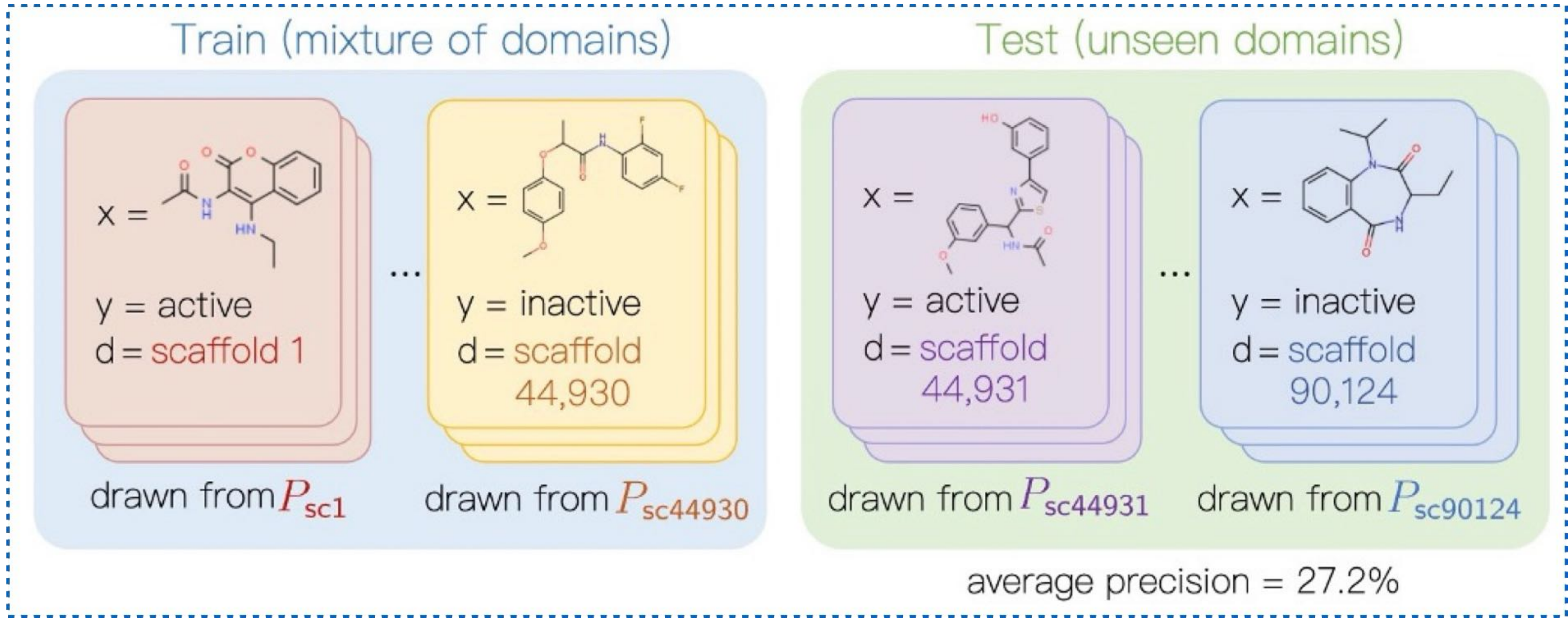
$f_{\mathrm{GNN}}($ {  } , { ○ ○ } $)$ = **“House”**



Structure-level shifts



Attribute-level shifts

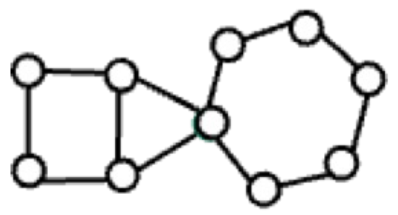


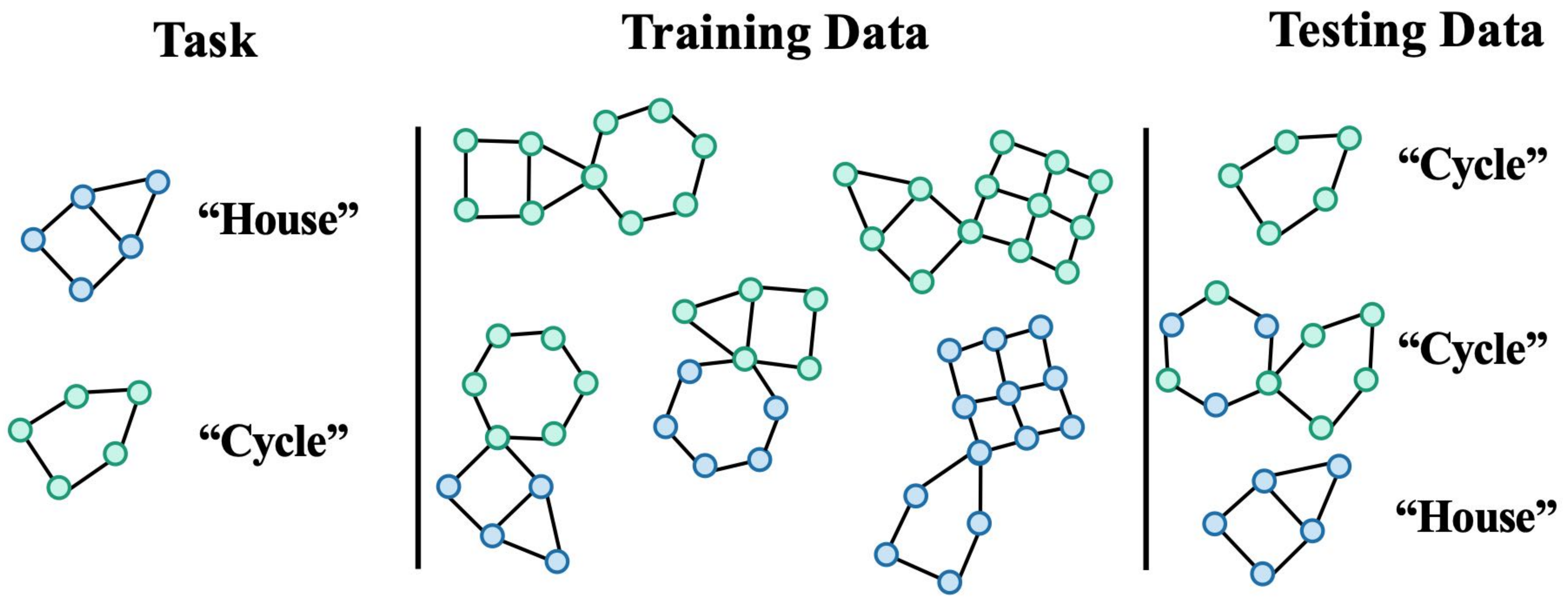
Mixture of structure-level and attribute-level shifts

(Knyazev et al. 2019; Hu et al., 2020; Koh et al., 2021; Gui et al., 2022; Chen et al., 2022)

Out-of-Distribution Generalization on Graphs

OOD generalization on graphs is fundamentally more **challenging** than that on Euclidean data:

$f_{\text{GNN}}(\{$  $\}$, $\{$  $\})$ = **"House"**



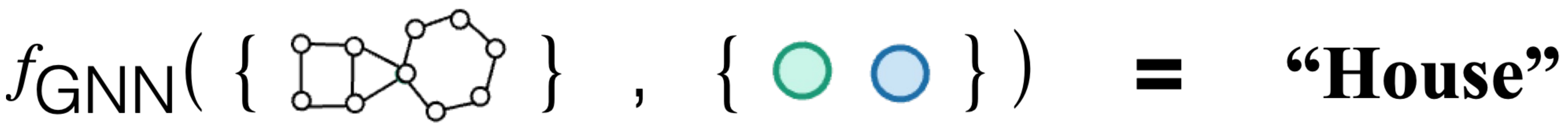
(Ying et al., 2019; Luo et al., 2020; Wu et al., 2022;)

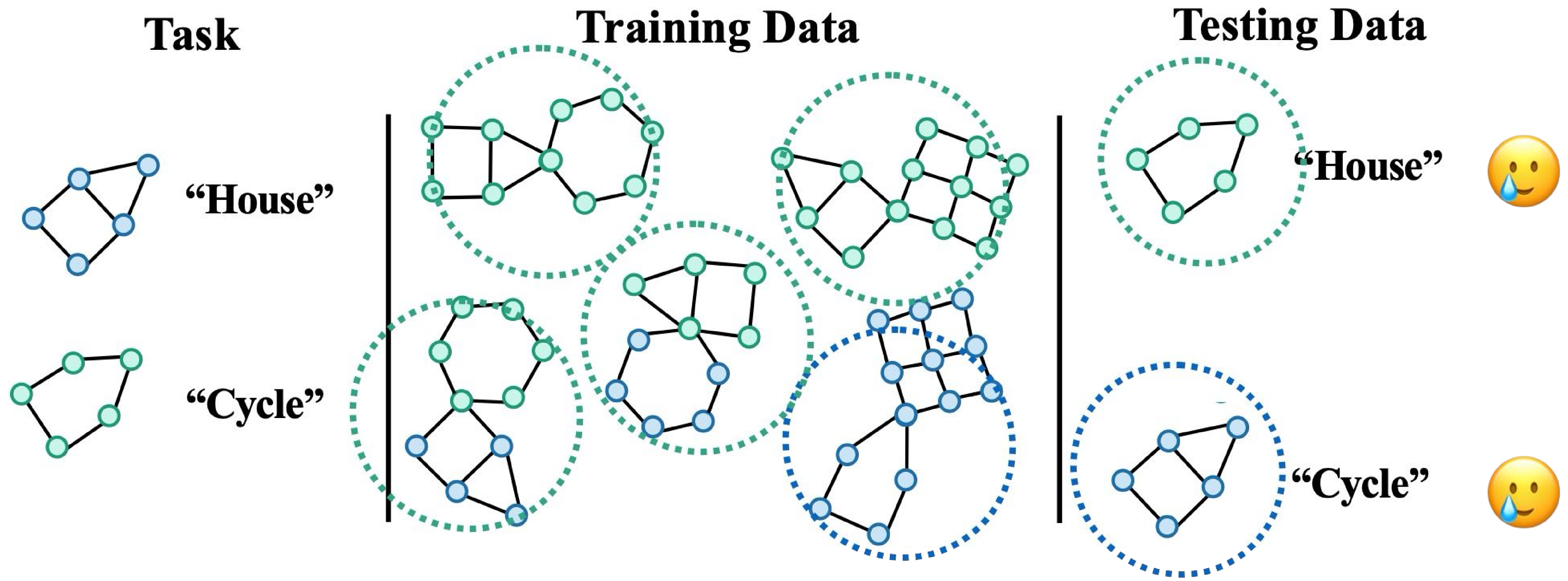
Specifically,

- Graphs are highly non-linear;

Out-of-Distribution Generalization on Graphs

OOD generalization on graphs is fundamentally more **challenging** than that on Euclidean data:





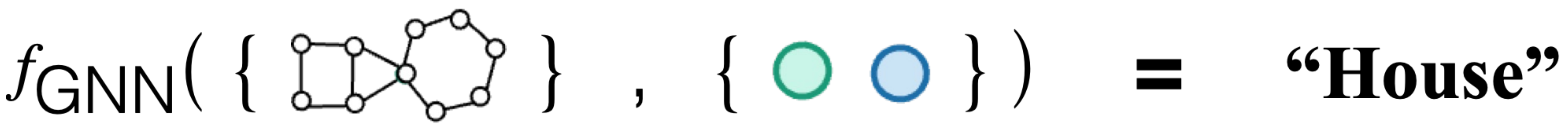
(Ying et al., 2019; Luo et al., 2020; Wu et al., 2022;)

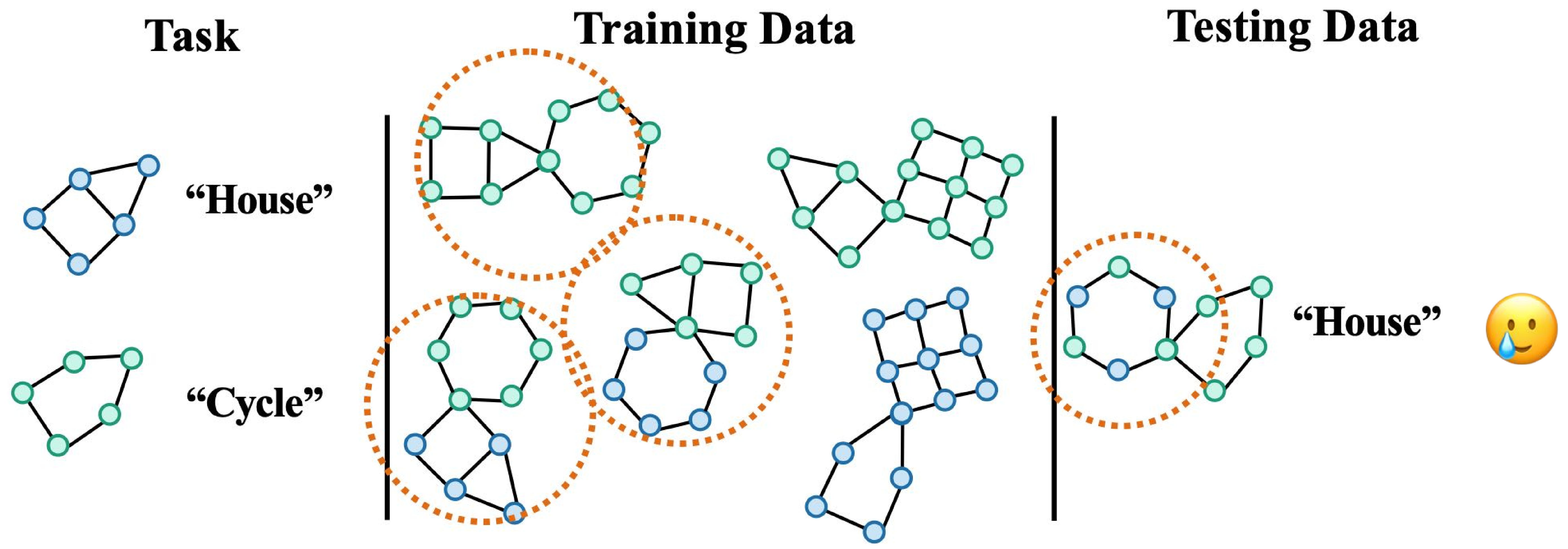
Specifically,

- Graphs are highly non-linear;
- There could be **attribute-level shifts**;

Out-of-Distribution Generalization on Graphs

OOD generalization on graphs is fundamentally more **challenging** than that on Euclidean data:





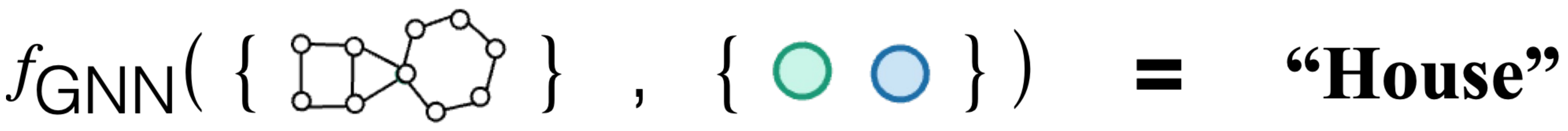
(Ying et al., 2019; Luo et al., 2020; Wu et al., 2022;)

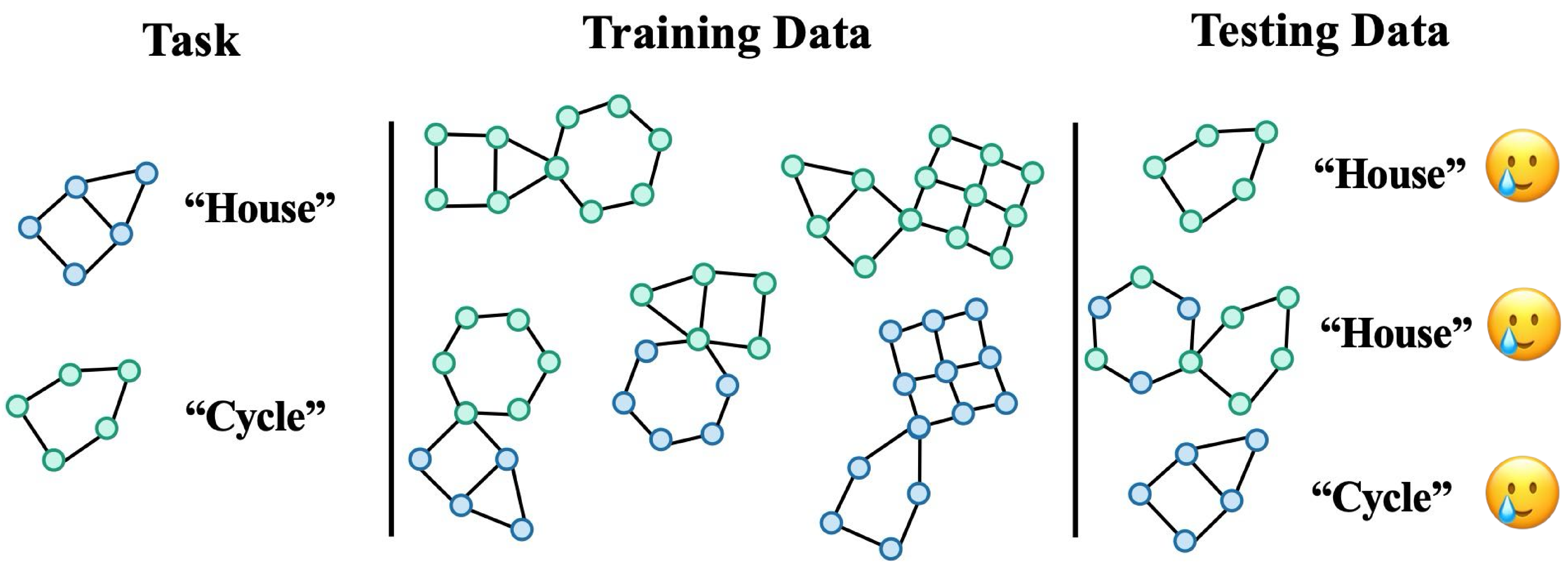
Specifically,

- Graphs are highly non-linear;
- There could be attribute-level shifts;
- There could be **structure-level shifts**;

Out-of-Distribution Generalization on Graphs

OOD generalization on graphs is fundamentally more **challenging** than that on Euclidean data:





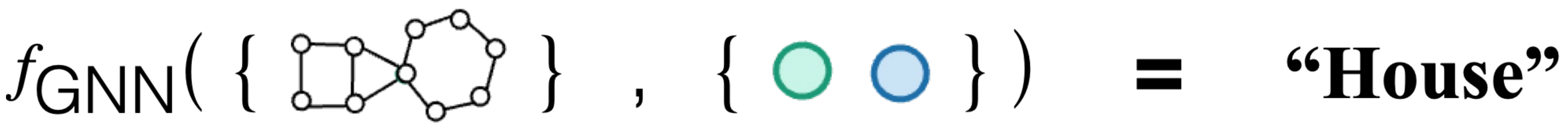
Specifically,

- Graphs are highly non-linear;
- There could be attribute-level shifts;
- There could be structure-level shifts;
- Both shifts can be **mixed**;

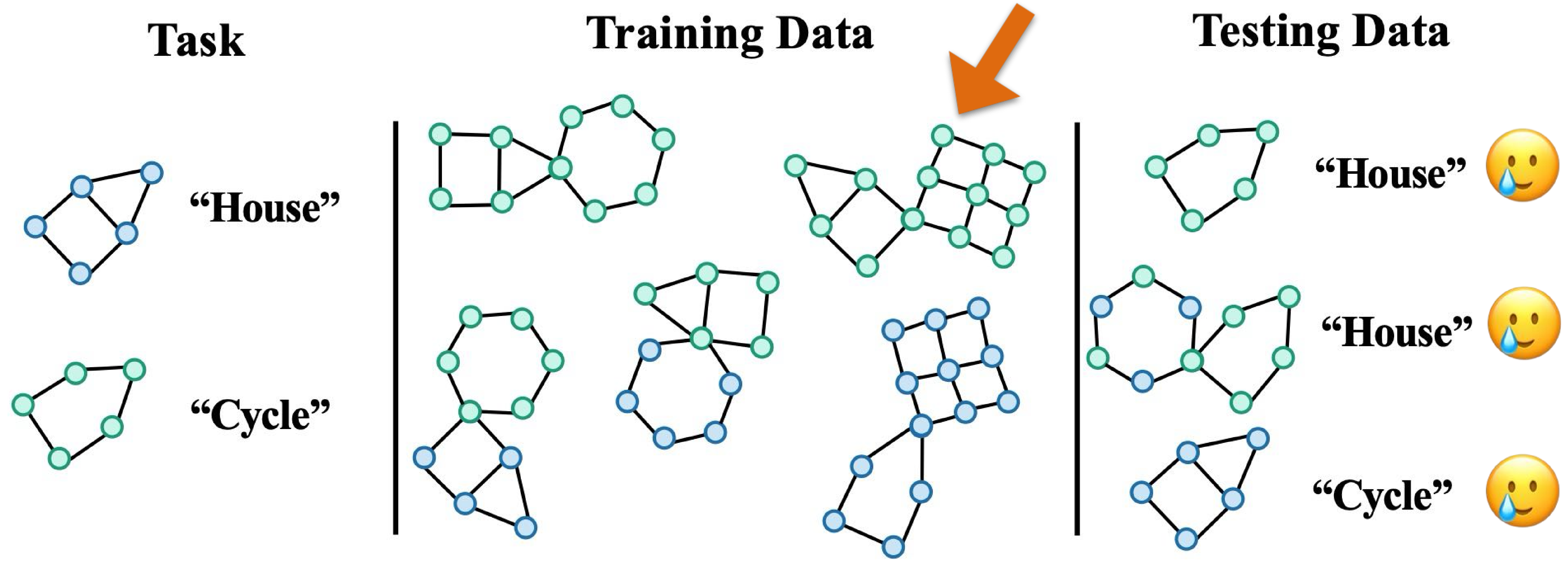
(Ying et al., 2019; Luo et al., 2020; Wu et al., 2022;)

Out-of-Distribution Generalization on Graphs

OOD generalization on graphs is fundamentally more **challenging** than that on Euclidean data:



No environment partitions

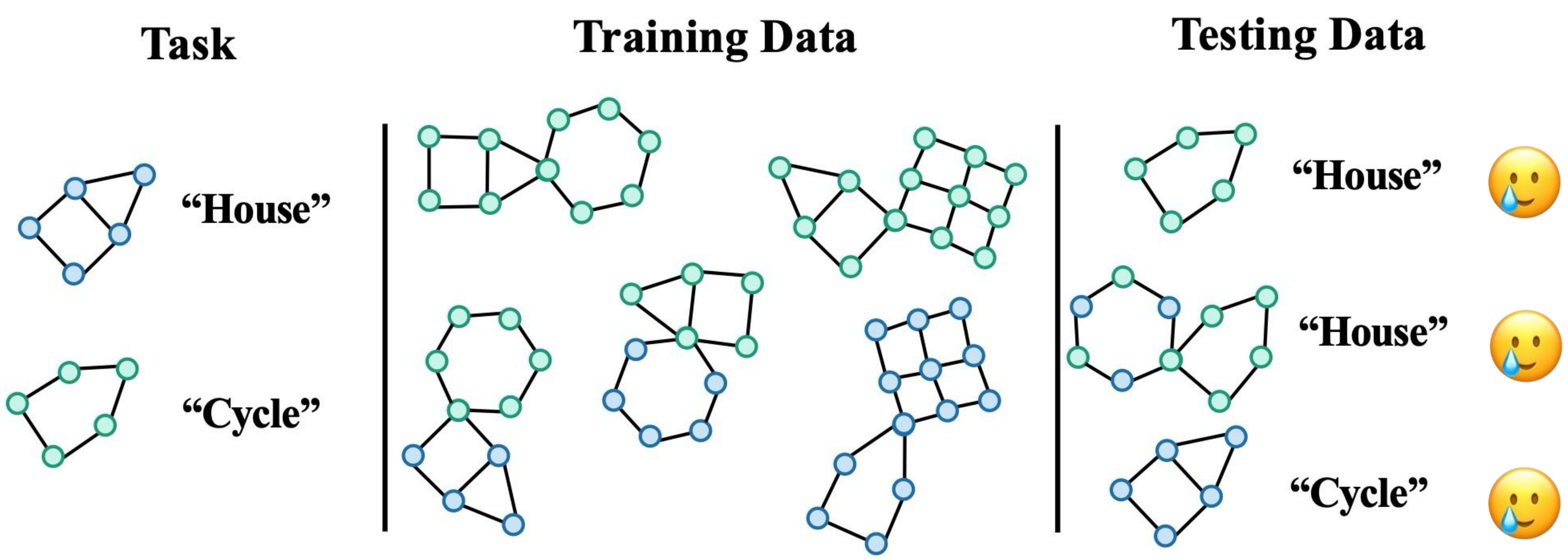


(Ying et al., 2019; Luo et al., 2020; Wu et al., 2022;)

Specifically,

- Graphs are highly non-linear;
- There could be attribute-level shifts;
- There could be structure-level shifts;
- Both shifts can be mixed;
- **Environment partitions** are expensive

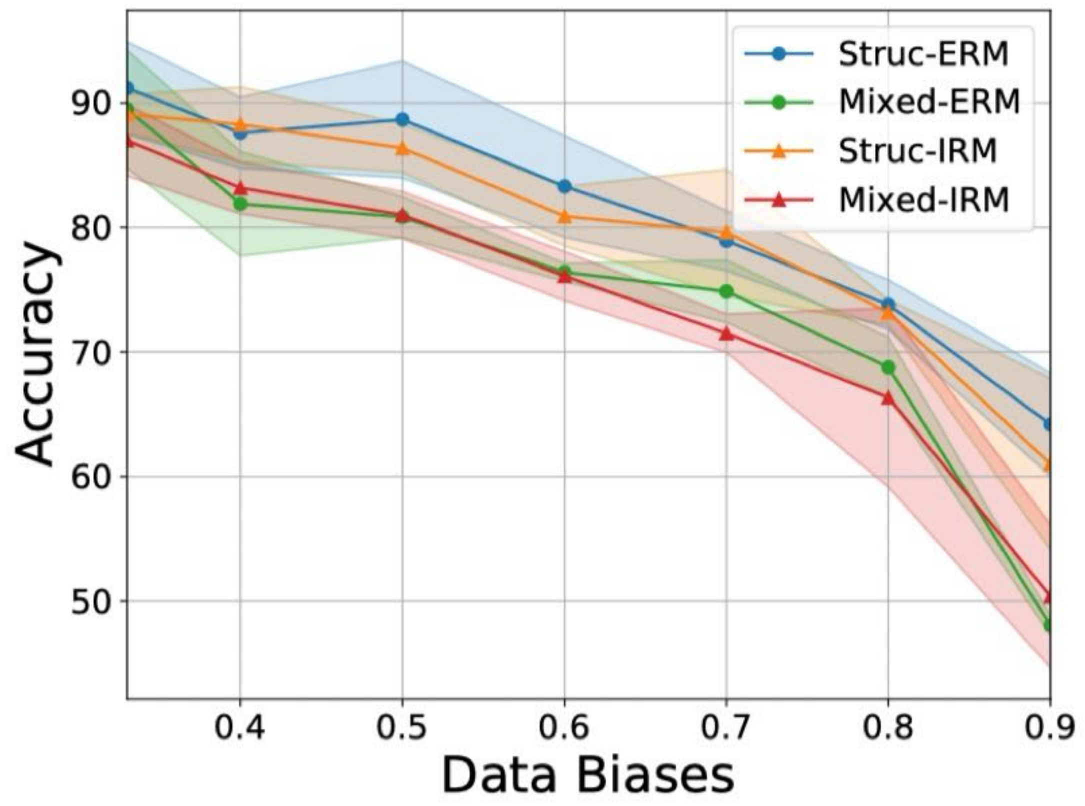
Out-of-Distribution Generalization on Graphs



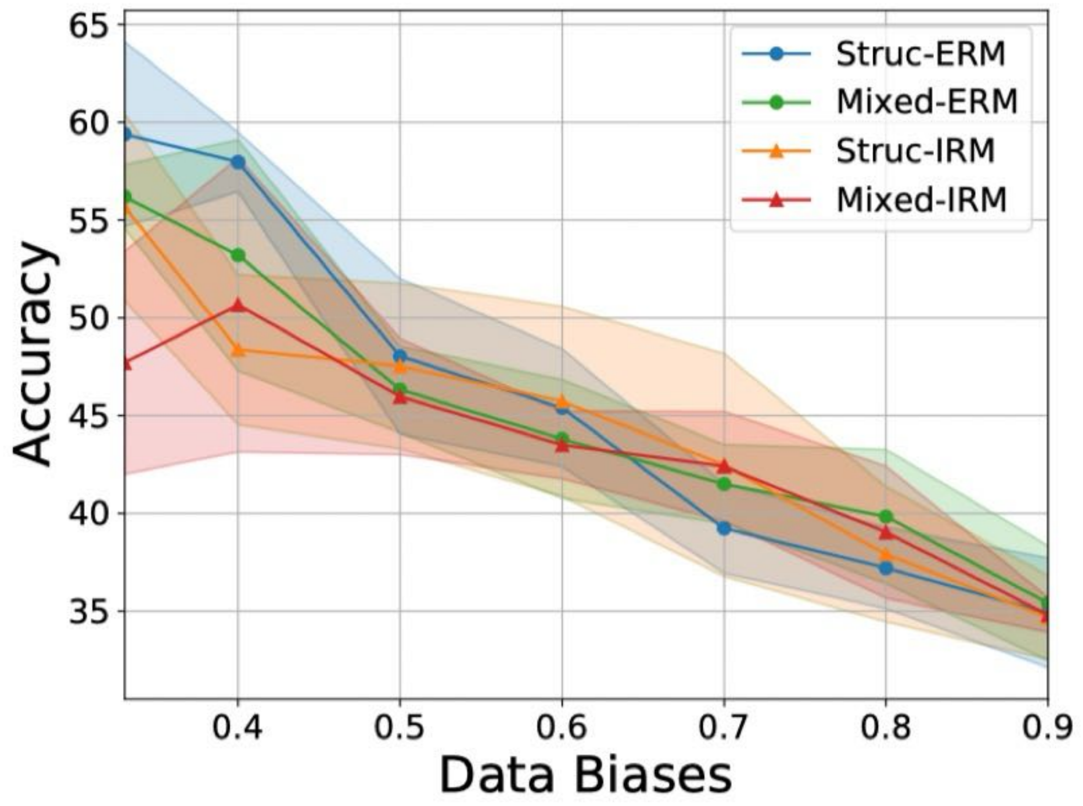
(Ying et al., 2019; Luo et al., 2020; Wu et al., 2022;)

OOD generalization on graphs are **much more challenging!**

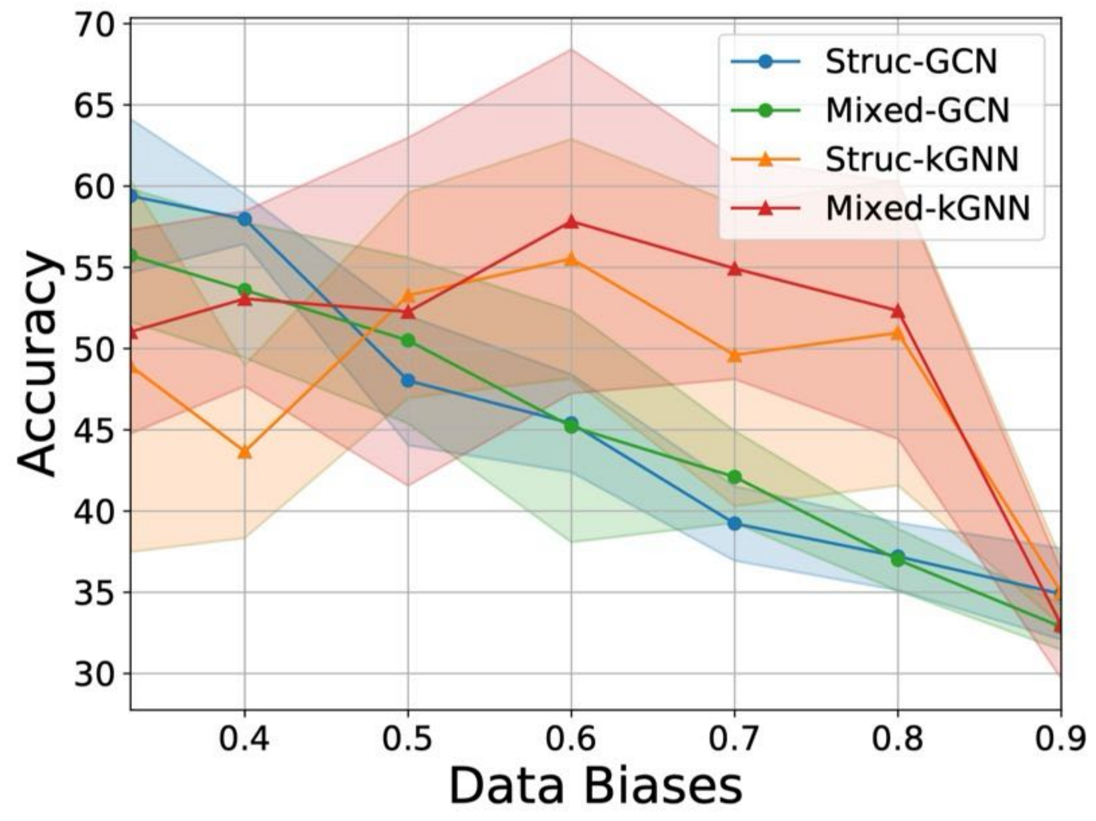
- Graphs are highly non-linear
- Attribute-level shifts
- Structure-level shifts
- Mixed shifts in different modes
- Expensive environment labels



Structure and attribute shifts



Mixed with **graph size** shifts



Structure and attribute shifts

OOD failures of GNNs ***training objectives*** and ***architectures***

How can we model the complicated graph distribution shifts?
And train a GNN to capture the desired invariance?

Structural Causal Models for Graph Generation

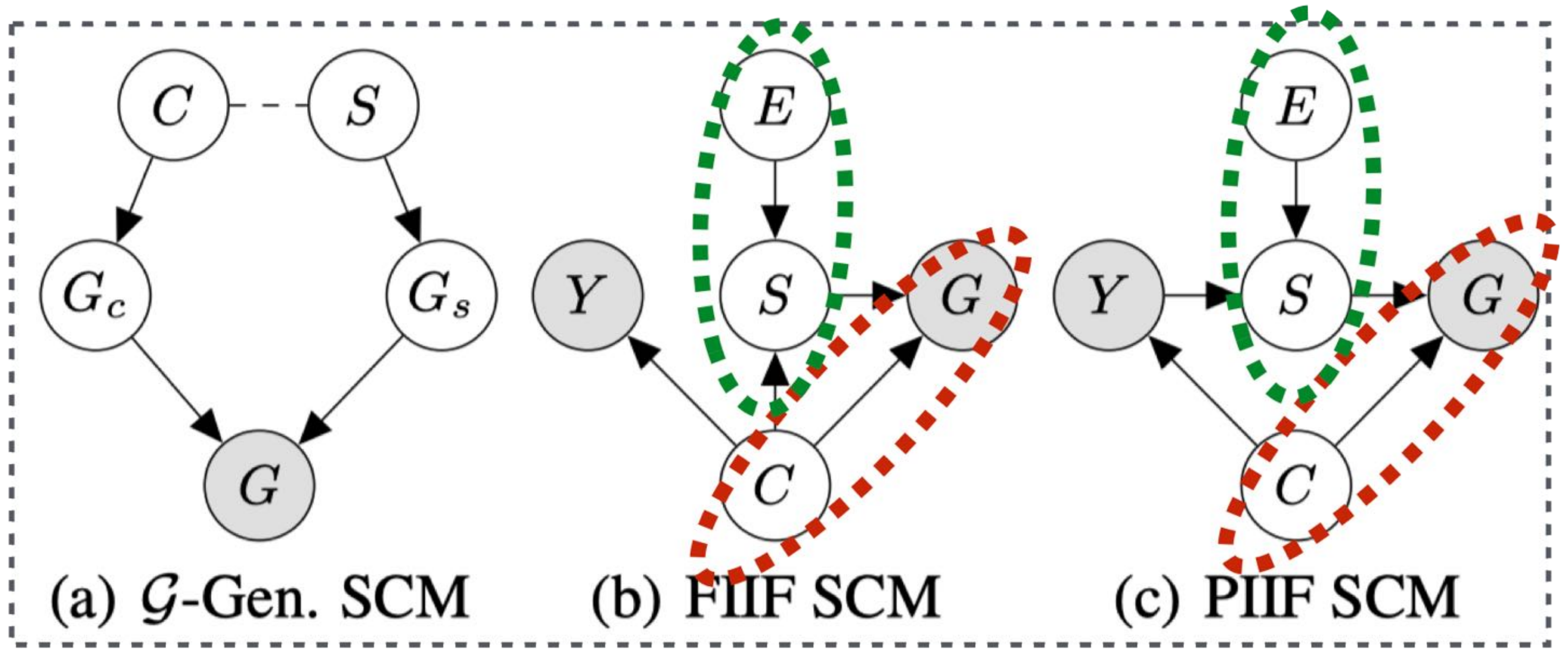
We formulate the **most comprehensive** causal models for distribution shifts on graphs.

Graph Generation Process:

$$f_{\text{gen}} : \mathcal{Z} \rightarrow \mathcal{G}$$

Invariant features C

S Spurious features



Structural Causal Models

Structural Causal Models for Graph Generation

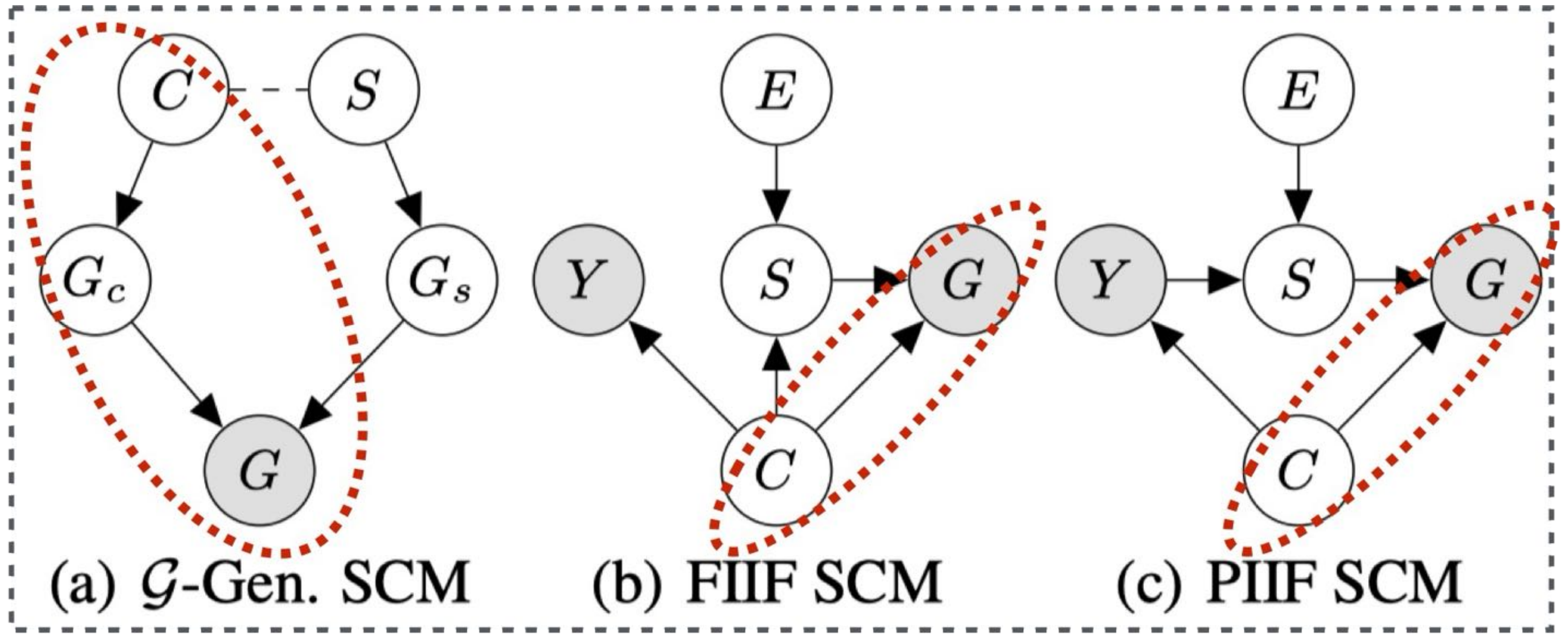
We formulate the **most comprehensive** causal models for distribution shifts on graphs.

Graph Generation Process:

$$f_{\text{gen}} : \mathcal{Z} \to \mathcal{G}$$

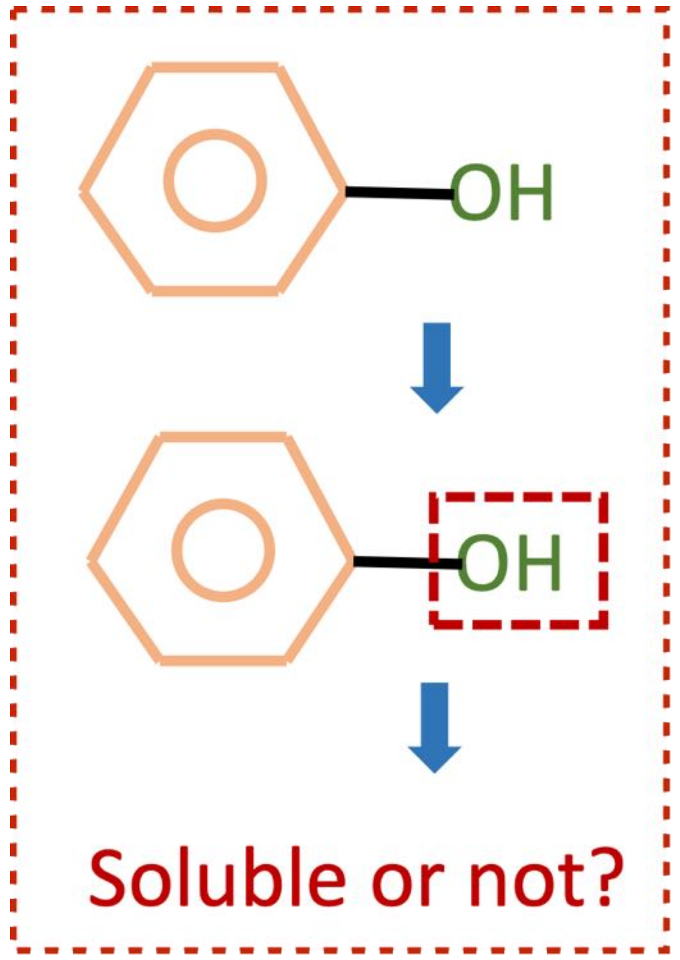
C Invariant features

S Spurious features



Structural Causal Models

Realistic examples

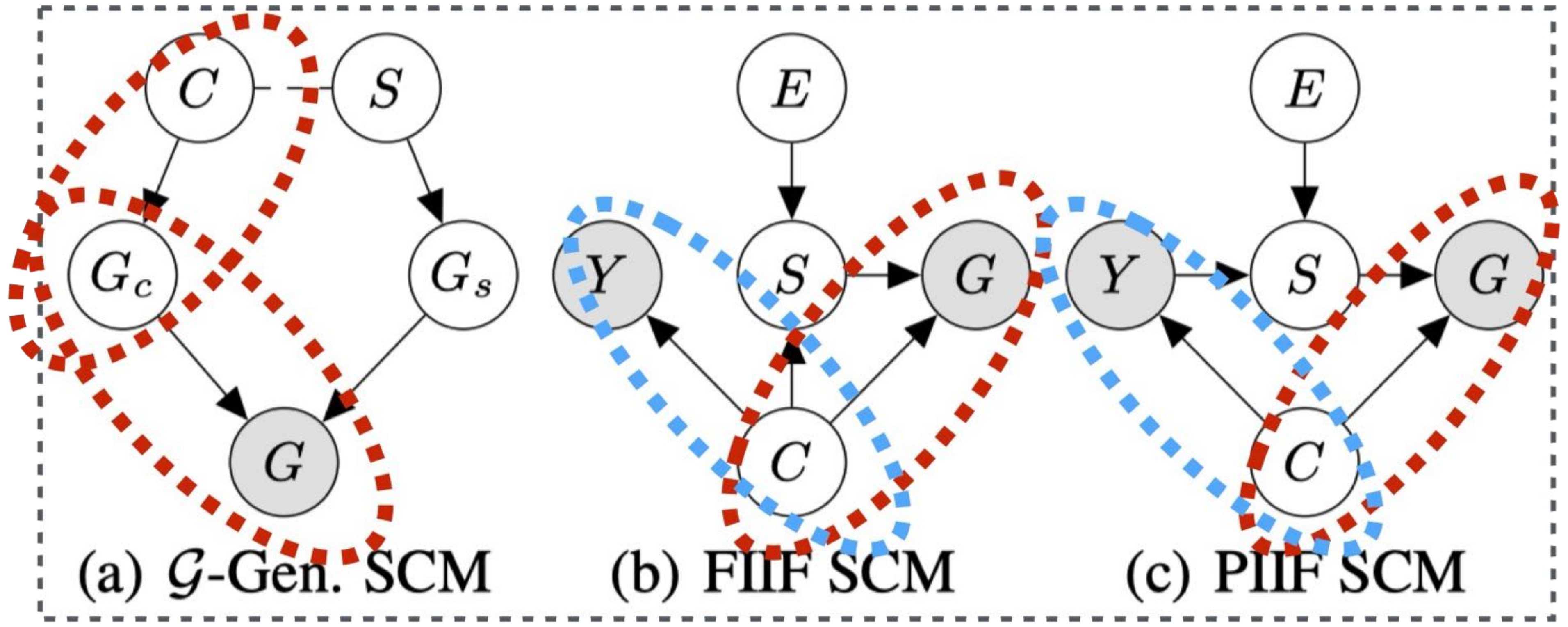


CIGA: Causality Inspired Invariant Graph LeArning

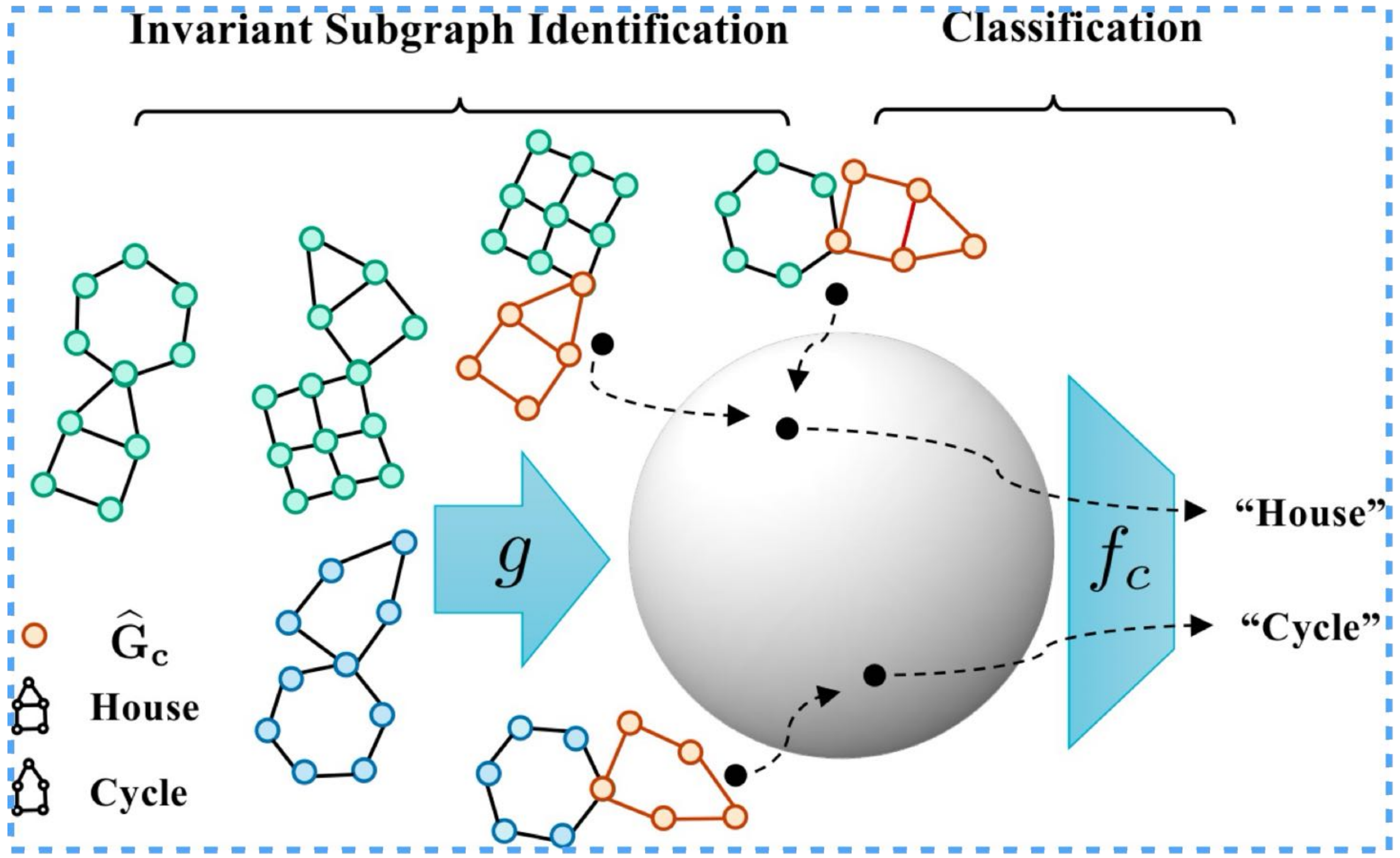
We propose a new framework, CIGA, that approaches the classification in two steps:

Step 1: Invariant subgraph identification

Featurizer GNN $g : \mathcal{G} \rightarrow \mathcal{G}_c$



Structural Causal Models



Step 2: Label prediction

Classifier GNN $f_c : \mathcal{G}_c \rightarrow \mathcal{Y}$

Overall objective

$$\boxed{\max_{f_c,\, g} I(\widehat{G}_c; Y),\ \text{s.t.}\ \widehat{G}_c \perp\!\!\!\perp E,\ \widehat{G}_c = g(G),}$$

Informative

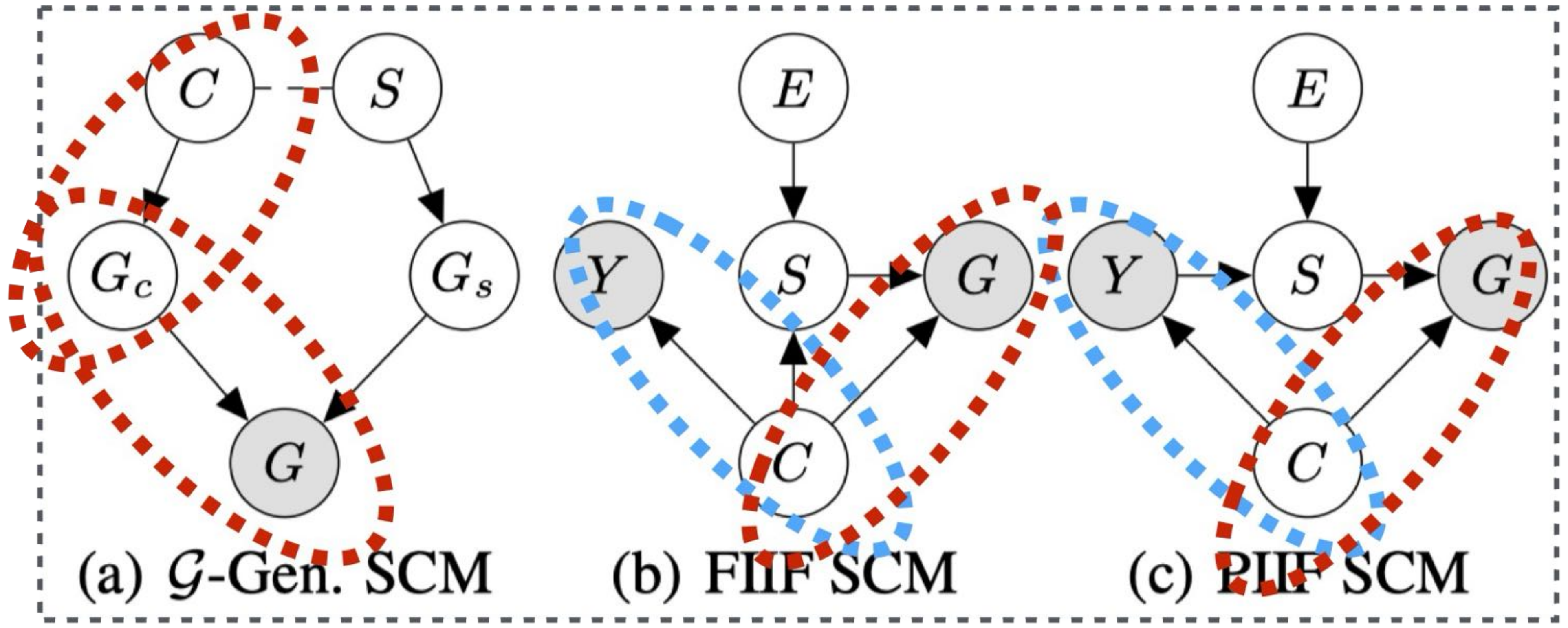
Invariant

CIGA: Causality Inspired Invariant Graph LeArning

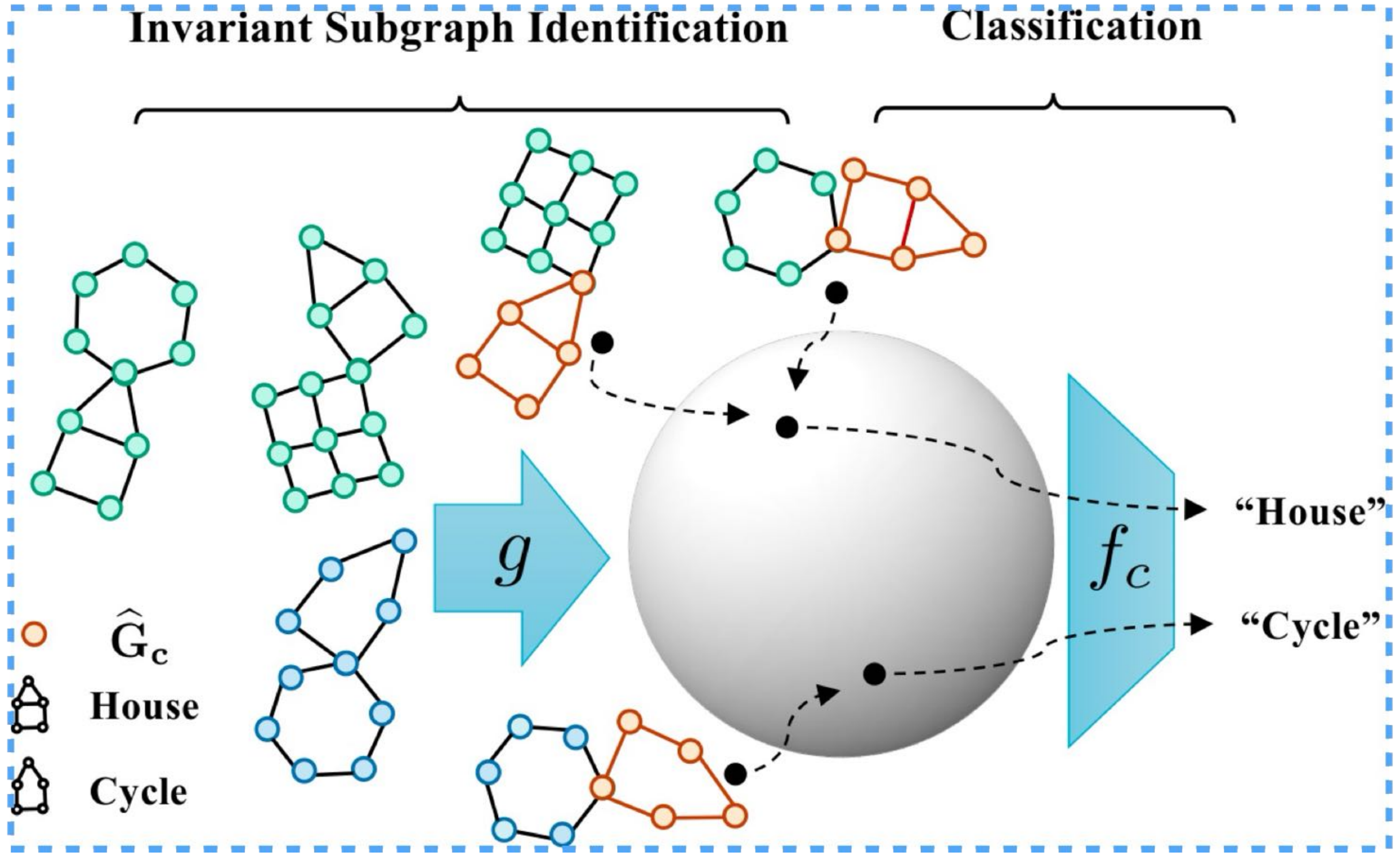
We propose a new framework, CIGA, that approaches the classification in two steps:

When $|G_c| = s_c$ is known and fixed,

$$G_c^{e_1} \in \arg\max_{\widehat{G}_c^{e_1}} I(\widehat{G}_c^{e_1};\widehat{G}_c^{e_2}|C=c) - I(\widehat{G}_c^{e_1};\widehat{G}_{c'}^{e_2}|C=c', c' \neq c),$$



Structural Causal Models



CIGAv1: using Y as a proxy of C

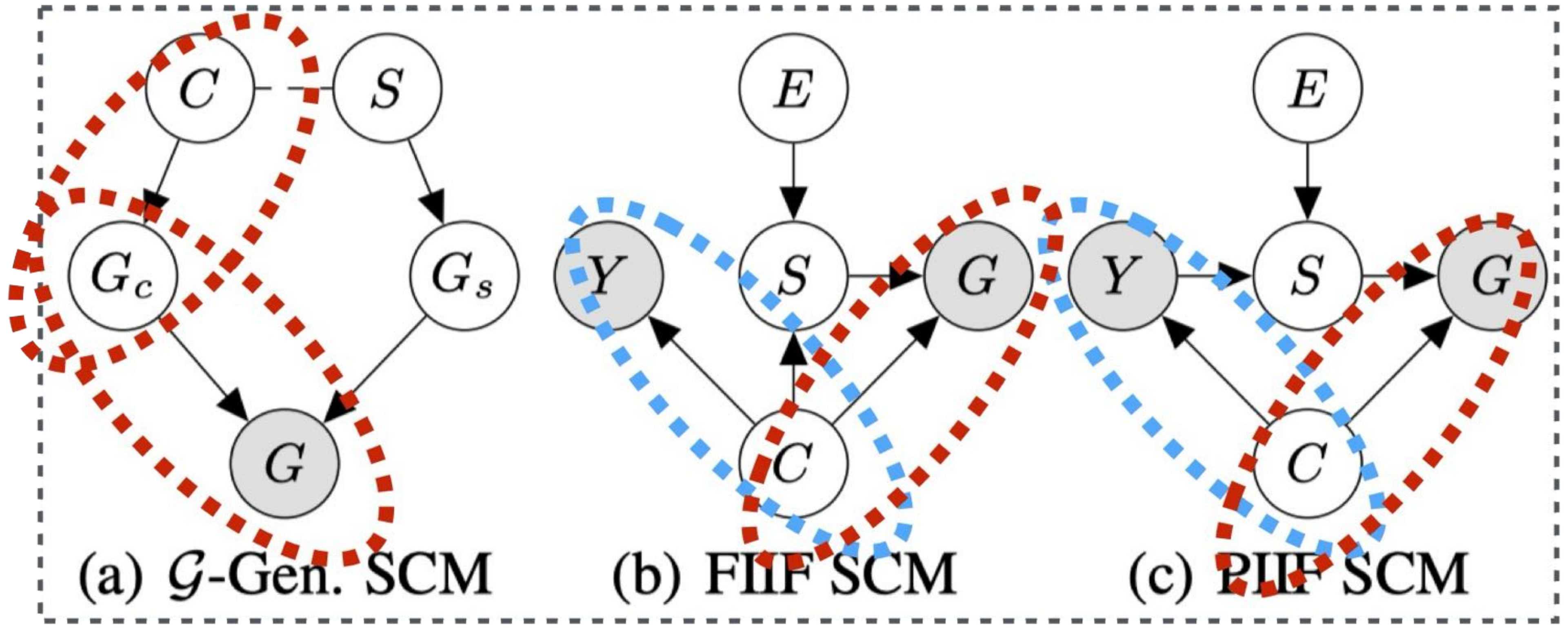
$$\max_{f_c,g} I(\widehat{G}_c;Y), \text{ s.t. } \widehat{G}_c \in \arg\max_{\widehat{G}_c=g(G),|\widehat{G}_c|\leq s_c} I(\widehat{G}_c;\widetilde{G}_c|Y),$$

MI estimation via ***supervised contrastive learning***

$$I(\widehat{G}_c;\widetilde{G}_c|Y) \approx \mathbb{E}_{\substack{\{\widehat{G}_c,\widetilde{G}_c\}\sim\mathbb{P}_g(G|\mathcal{Y}=Y)\\ \{G_c^i\}_{i=1}^M\sim\mathbb{P}_g(G|\mathcal{Y}\neq Y)}} \log \frac{e^{\phi(h_{\widehat{G}_c},h_{\widetilde{G}_c})}}{e^{\phi(h_{\widehat{G}_c},h_{\widetilde{G}_c})} + \sum_{i=1}^M e^{\phi(h_{\widehat{G}_c},h_{G_c^i})}},$$

CIGA: Causality Inspired Invariant Graph LeArning

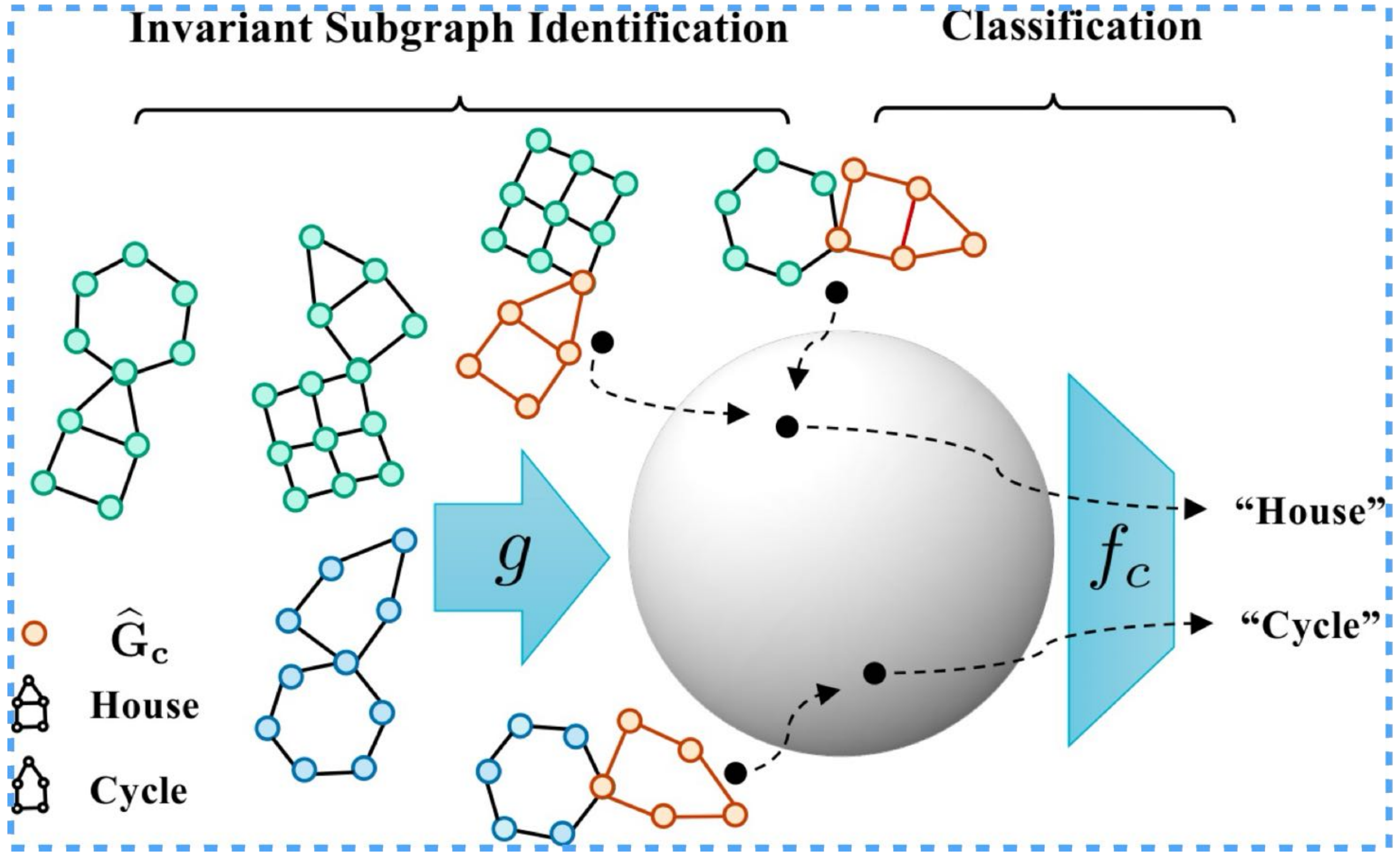
We propose a new framework, CIGA, that approaches the classification in two steps:



Structural Causal Models

CIGAv1: when $|G_c| = s_c$ is known and fixed

$$\max_{f_c, g} I(\widehat{G}_c; Y), \text{ s.t. } \widehat{G}_c \in \underset{\widehat{G}_c = g(G), |\widehat{G}_c| \leq s_c}{\arg\max} I(\widehat{G}_c; \widetilde{G}_c | Y),$$



CIGAv2: eliminate the size constraint

$$\max_{f_c, g} I(\widehat{G}_c; Y) + I(\widehat{G}_s; Y), \text{ s.t. } \widehat{G}_c \in \arg\max_{\widehat{G}_c = g(G)} I(\widehat{G}_c; \widetilde{G}_c | Y),$$
$$I(\widehat{G}_s; Y) \leq I(\widehat{G}_c; Y), \ \widehat{G}_s = G - g(G),$$

CIGA: Causality Inspired Invariant Graph LeArning

CIGA achieves the state-of-the-art OOD generalization performance under **30+** datasets and graph distribution shifts, including a OOD drug property prediction task.

Theoretical results (Informal):
Given the previous SCMs, each solution to CIGAv1 or CIGAv2 elicits a GNN that is generalizable against various distribution shifts, with some mild assumptions on training environments, and the expressivity of GNNs encoders.

Table 1: OOD generalization performance on structure and mixed shifts for synthetic graphs.

	SPMotif-Struc†			SPMotif-Mixed†			
	bias=0.33	bias=0.60	bias=0.90	bias=0.33	bias=0.60	bias=0.90	Avg
ERM	59.49 (3.50)	55.48 (4.84)	49.64 (4.63)	58.18 (4.30)	49.29 (8.17)	41.36 (3.29)	52.24
ASAP	64.87 (13.8)	64.85 (10.6)	**57.29 (14.5)**	66.88 (15.0)	59.78 (6.78)	**50.45 (4.90)**	60.69
DIR	58.73 (11.9)	48.72 (14.8)	41.90 (9.39)	67.28 (4.06)	51.66 (14.1)	38.58 (5.88)	51.14
IRM	57.15 (3.98)	61.74 (1.32)	45.68 (4.88)	58.20 (1.97)	49.29 (3.67)	40.73 (1.93)	52.13
V-Rex	54.64 (3.05)	53.60 (3.74)	48.86 (9.69)	57.82 (5.93)	48.25 (2.79)	43.27 (1.32)	51.07
EIIL	56.48 (2.56)	60.07 (4.47)	55.79 (6.54)	53.91 (3.15)	48.41 (5.53)	41.75 (4.97)	52.73
IB-IRM	58.30 (6.37)	54.37 (7.35)	45.14 (4.07)	57.70 (2.11)	50.83 (1.51)	40.27 (3.68)	51.10
CNC	70.44 (2.55)	**66.79 (9.42)**	50.25 (10.7)	65.75 (4.35)	59.27 (5.29)	41.58 (1.90)	59.01
CIGAv1	**71.07 (3.60)**	63.23 (9.61)	51.78 (7.29)	**74.35 (1.85)**	**64.54 (8.19)**	49.01 (9.92)	**62.33**
CIGAv2	**77.33 (9.13)**	**69.29 (3.06)**	**63.41 (7.38)**	**72.42 (4.80)**	**70.83 (7.54)**	**54.25 (5.38)**	**67.92**
Oracle (IID)		88.70 (0.17)			88.73 (0.25)		

†Higher accuracy and lower variance indicate better OOD generalization ability.

CIGA outperforms previous methods under ***structure and mixed shifts*** by a significant margin up to ***10%***.

CIGA: Causality Inspired Invariant Graph LeArning

CIGA achieves the state-of-the-art OOD generalization performance under **30+** datasets and graph distribution shifts, including a OOD drug property prediction task.

Theoretical results (Informal):
Given the previous SCMs, each solution to CIGAv1 or CIGAv2 elicits a GNN that is generalizable against various distribution shifts, with some mild assumptions on training environments, and the expressivity of GNNs encoders.

Table 2: OOD generalization performance on complex distribution shifts for real-world graphs.

DATASETS	DRUG-ASSAY	DRUG-SCA	DRUG-SIZE	CMNIST-SP	GRAPH-SST5	TWITTER	AVG (RANK)†
ERM	71.79 (0.27)	68.85 (0.62)	66.70 (1.08)	13.96 (5.48)	43.89 (1.73)	60.81 (2.05)	54.33 (6.00)
ASAP	70.51 (1.93)	66.19 (0.94)	64.12 (0.67)	10.23 (0.51)	44.16 (1.36)	60.68 (2.10)	52.65 (8.33)
GIB	63.01 (1.16)	62.01 (1.41)	55.50 (1.42)	15.40 (3.91)	38.64 (4.52)	48.08 (2.27)	47.11 (10.0)
DIR	68.25 (1.40)	63.91 (1.36)	60.40 (1.42)	15.50 (8.65)	41.12 (1.96)	59.85 (2.98)	51.51 (9.33)
IRM	72.12 (0.49)	68.69 (0.65)	66.54 (0.42)	31.58 (9.52)	43.69 (1.26)	63.50 (1.23)	57.69 (4.50)
V-REX	72.05 (1.25)	68.92 (0.98)	66.33 (0.74)	10.29 (0.46)	43.28 (0.52)	63.21 (1.57)	54.01 (6.17)
EIIL	72.60 (0.47)	68.45 (0.53)	66.38 (0.66)	30.04 (10.9)	42.98 (1.03)	62.76 (1.72)	57.20 (5.33)
IB-IRM	72.50 (0.49)	68.50 (0.40)	66.64 (0.28)	**39.86 (10.5)**	40.85 (2.08)	61.26 (1.20)	58.27 (5.33)
CNC	72.40 (0.46)	67.24 (0.90)	65.79 (0.80)	12.21 (3.85)	42.78 (1.53)	61.03 (2.49)	53.56 (7.50)
CIGAv1	**72.71 (0.52)**	**69.04 (0.86)**	**67.24 (0.88)**	19.77 (17.1)	**44.71 (1.14)**	**63.66 (0.84)**	**56.19 (2.50)**
CIGAv2	**73.17 (0.39)**	**69.70 (0.27)**	**67.78 (0.76)**	**44.91 (4.31)**	**45.25 (1.27)**	**64.45 (1.99)**	**60.88 (1.00)**
ORACLE (IID)	85.56 (1.44)	84.71 (1.60)	85.83 (1.31)	62.13 (0.43)	48.18 (1.00)	64.21 (1.77)	

†Averaged rank is also reported in the blankets because of dataset heterogeneity. Lower rank is better.

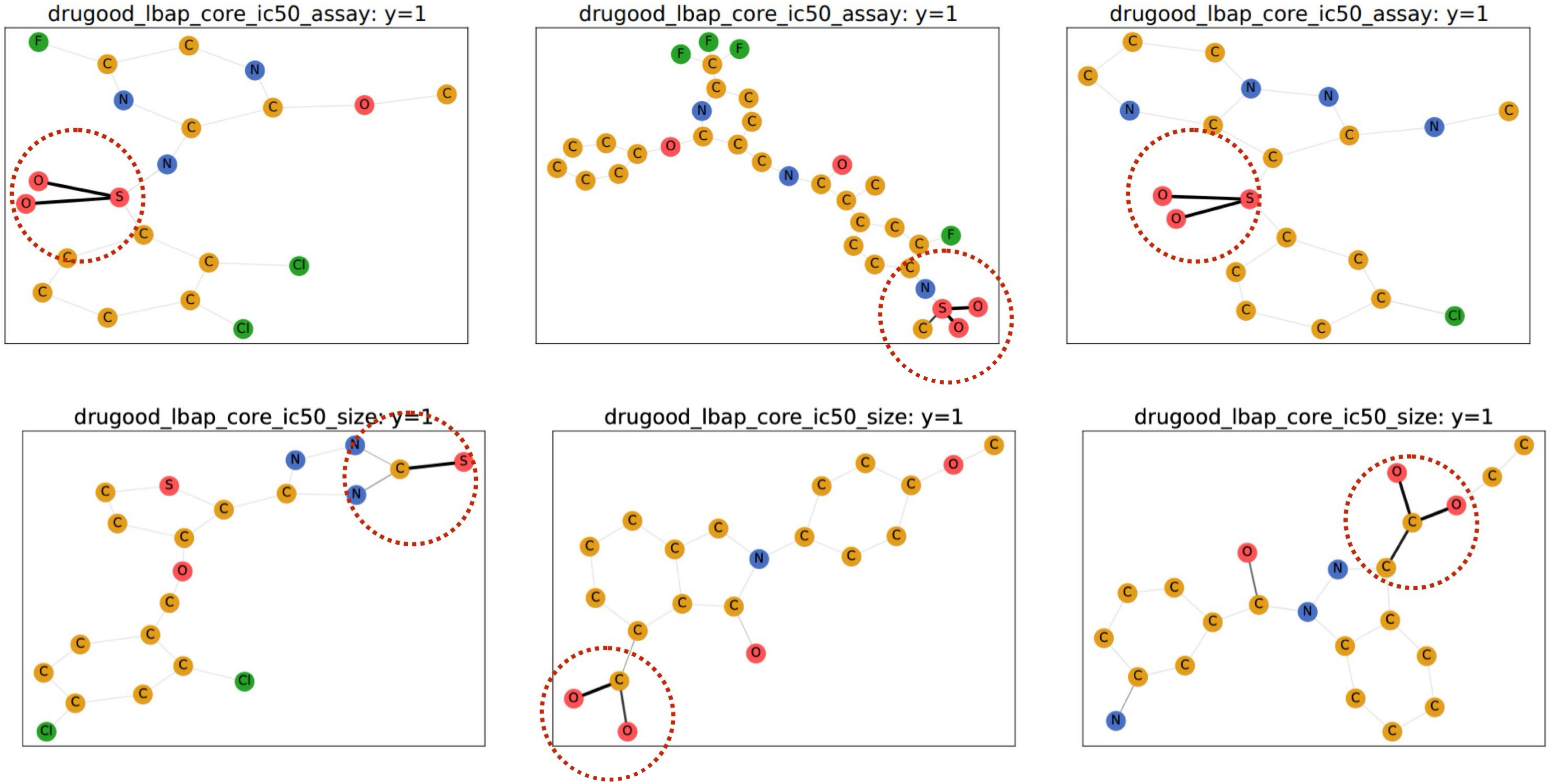
Table 3: OOD generalization performance on graph size shifts for real-world graphs in terms of Matthews correlation coefficient.

DATASETS	NCI1	NCI109	PROTEINS	DD	AVG
ERM	0.15 (0.05)	0.16 (0.02)	0.22 (0.09)	0.27 (0.09)	0.20
ASAP	0.16 (0.10)	0.15 (0.07)	0.22 (0.16)	0.21 (0.08)	0.19
GIB	0.13 (0.10)	0.16 (0.02)	0.19 (0.08)	0.01 (0.18)	0.12
DIR	0.21 (0.06)	0.13 (0.05)	0.25 (0.14)	0.20 (0.10)	0.20
IRM	0.17 (0.02)	0.14 (0.01)	0.21 (0.09)	0.22 (0.08)	0.19
V-REX	0.15 (0.04)	0.15 (0.04)	0.22 (0.06)	0.21 (0.07)	0.18
EIIL	0.14 (0.03)	0.16 (0.02)	0.20 (0.05)	0.23 (0.10)	0.19
IB-IRM	0.12 (0.04)	0.15 (0.06)	0.21 (0.06)	0.15 (0.13)	0.16
CNC	0.16 (0.04)	0.16 (0.04)	0.19 (0.08)	0.27 (0.13)	0.20
WL KERNEL	**0.39 (0.00)**	0.21 (0.00)	0.00 (0.00)	0.00 (0.00)	0.15
GC KERNEL	0.02 (0.00)	0.00 (0.00)	0.29 (0.00)	0.00 (0.00)	0.08
$\Gamma_{\text{1-HOT}}$	0.17 (0.08)	**0.25 (0.06)**	0.12 (0.09)	0.23 (0.08)	0.19
Γ_{GIN}	0.24 (0.04)	0.18 (0.04)	0.29 (0.11)	**0.28 (0.06)**	0.25
Γ_{RPGIN}	0.26 (0.05)	0.20 (0.04)	0.25 (0.12)	0.20 (0.05)	0.23
CIGAv1	0.22 (0.07)	**0.23 (0.09)**	**0.40 (0.06)**	**0.29 (0.08)**	**0.29**
CIGAv2	**0.27 (0.07)**	0.22 (0.05)	**0.31 (0.12)**	0.26 (0.08)	**0.27**
ORACLE (IID)	0.32 (0.05)	0.37 (0.06)	0.39 (0.09)	0.33 (0.05)	

CIGA outperforms previous methods under other ***realistic shifts*** by a significant margin up to ***10%***.

Causal Interpretable Patterns for Scientific Practice

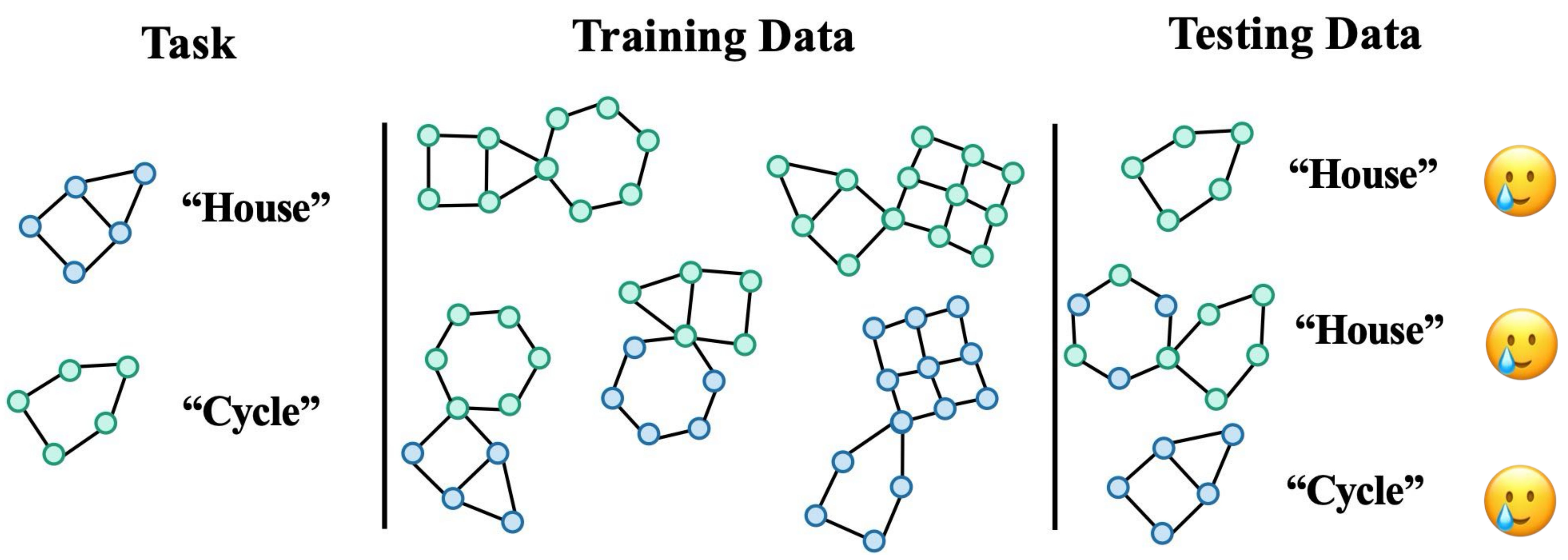
CIGA finds interesting critical functional groups/sub-molecules in OOD molecular affinity prediction.



(Ji et al., 2022;)

OOD Generalization Challenges Solved by CIGA

Let us trace back the challenges in OOD generalization on graphs…



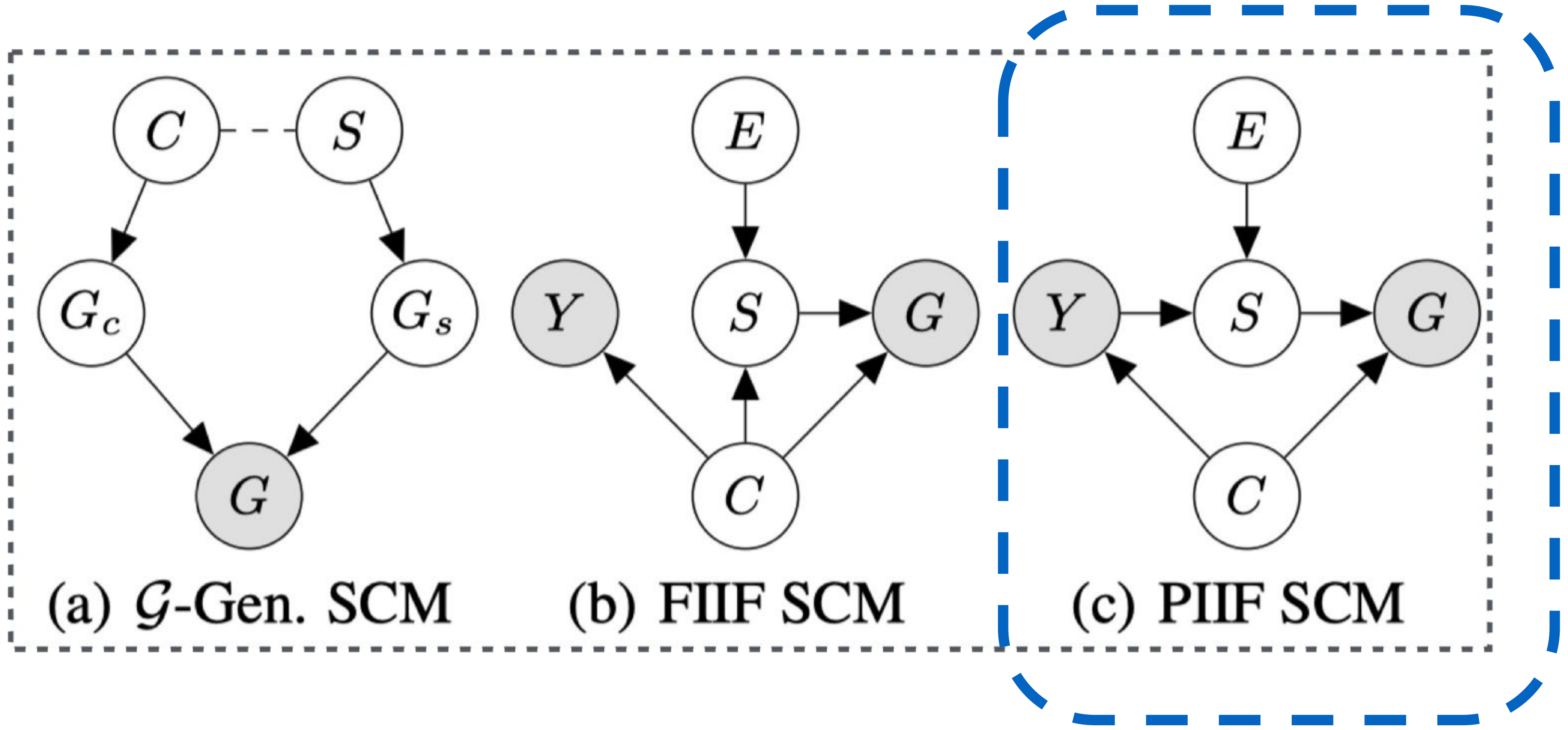
(Ying et al., 2019; Luo et al., 2020; Wu et al., 2022;)

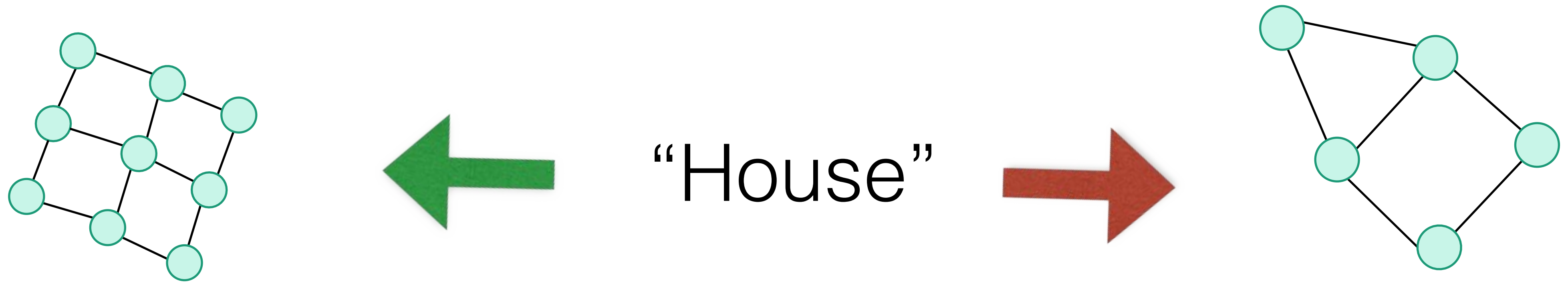
OOD generalization on graphs are **much more challenging!**

- Graphs are highly non-linear 😋
- Attribute-level shifts 😋
- Structure-level shifts 😋
- Mixed shifts in different modes 😋
- Expensive environment labels 🤔

The “Free Lunch Dilemma” in OOD Generalization on Graphs

Let us considering the data generative model with **Partial Informative Invariant Features**:



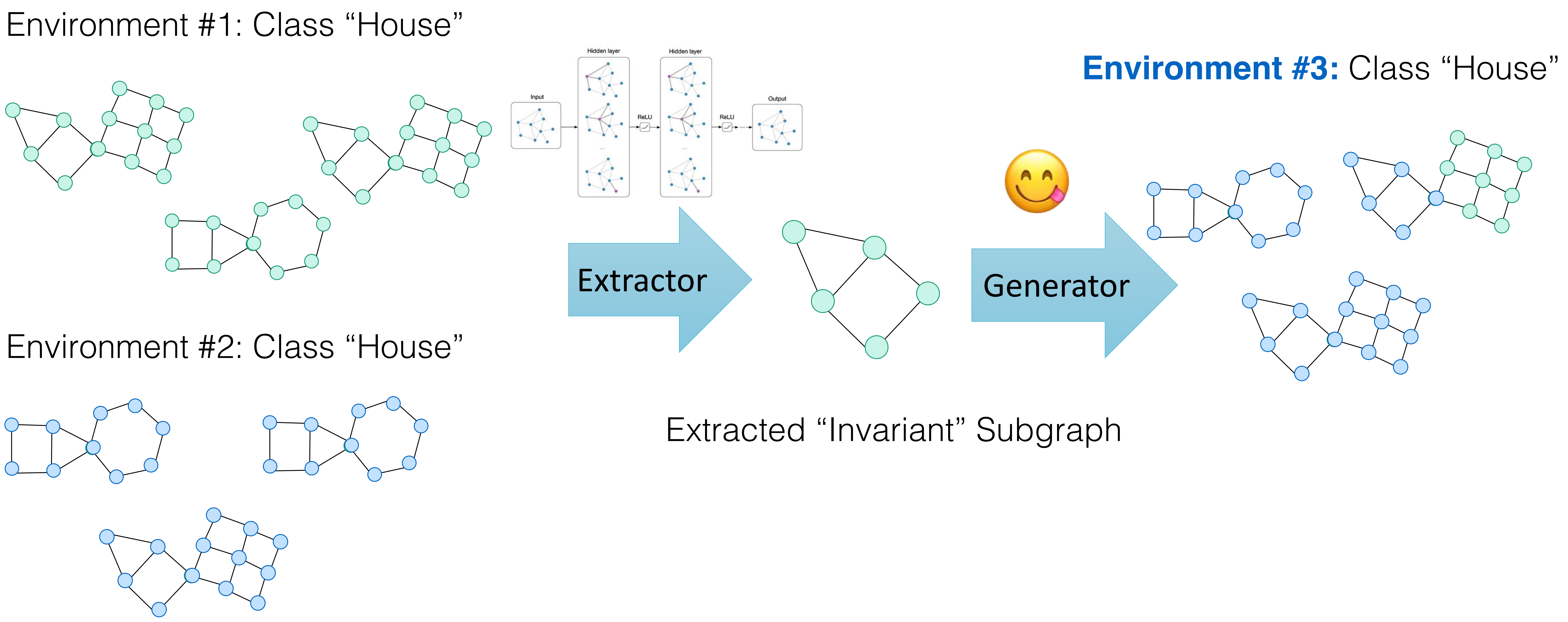


Spurious correlation Invariant correlation

(Yu et al., 2021; Miao et al., 2022;)

The “Free Lunch Dilemma” in OOD Generalization on Graphs

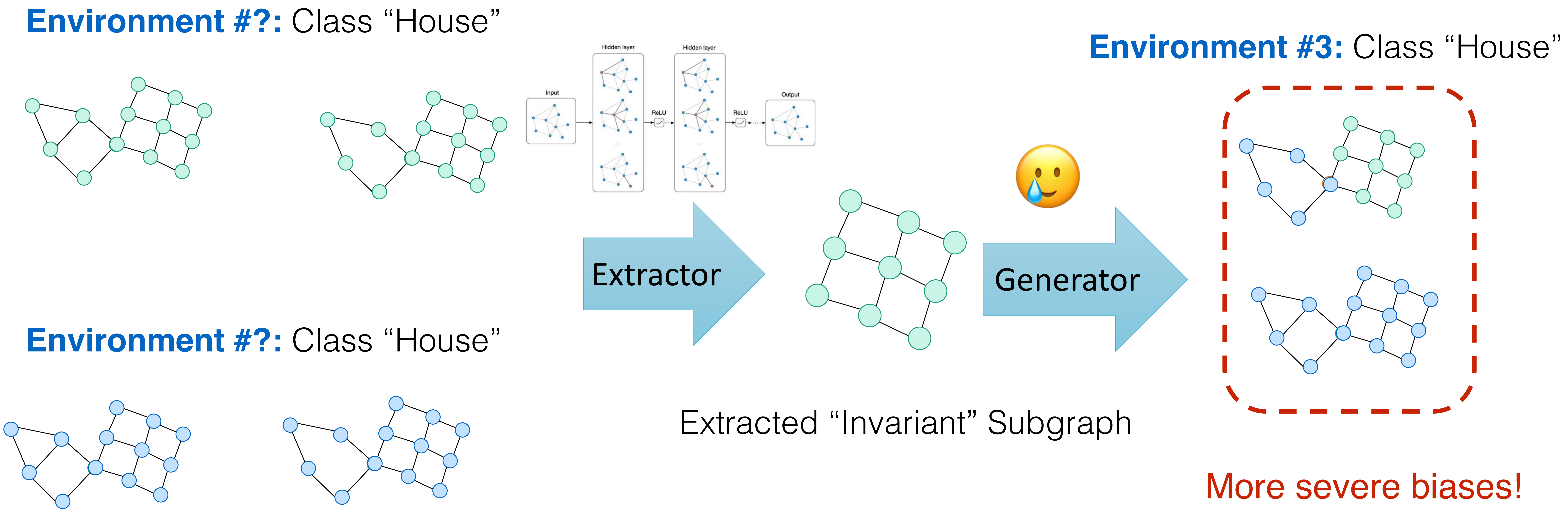
One line of works aim to generate **new environments** based on the existing extracted subgraphs:



(Wu et al., 2022ab; Liu et al., 2022)

The “Free Lunch Dilemma” in OOD Generalization on Graphs

One line of works aim to generate **new environments** based on the existing extracted subgraphs:

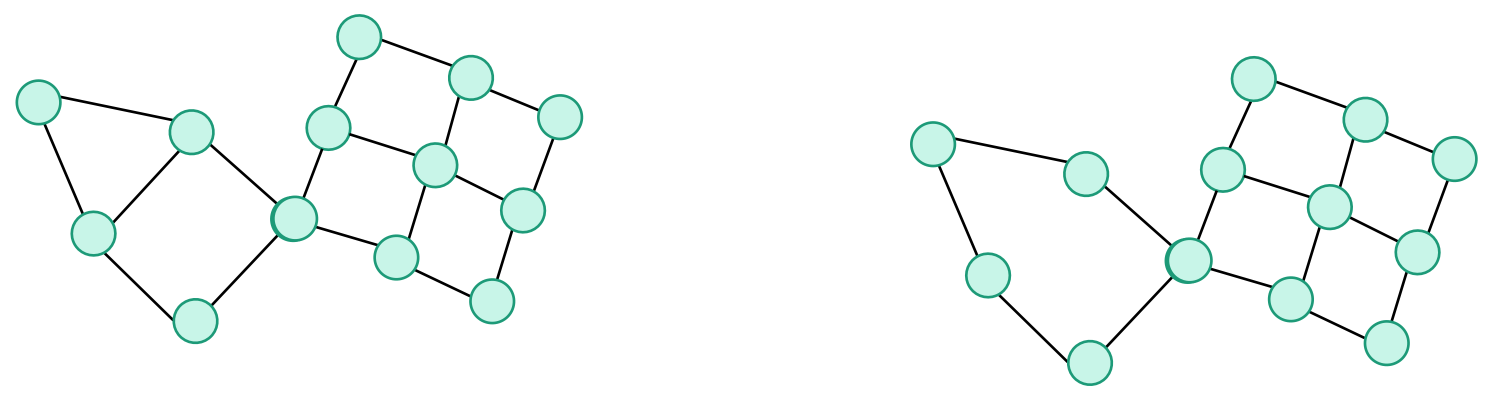


(Wu et al., 2022ab; Liu et al., 2022)

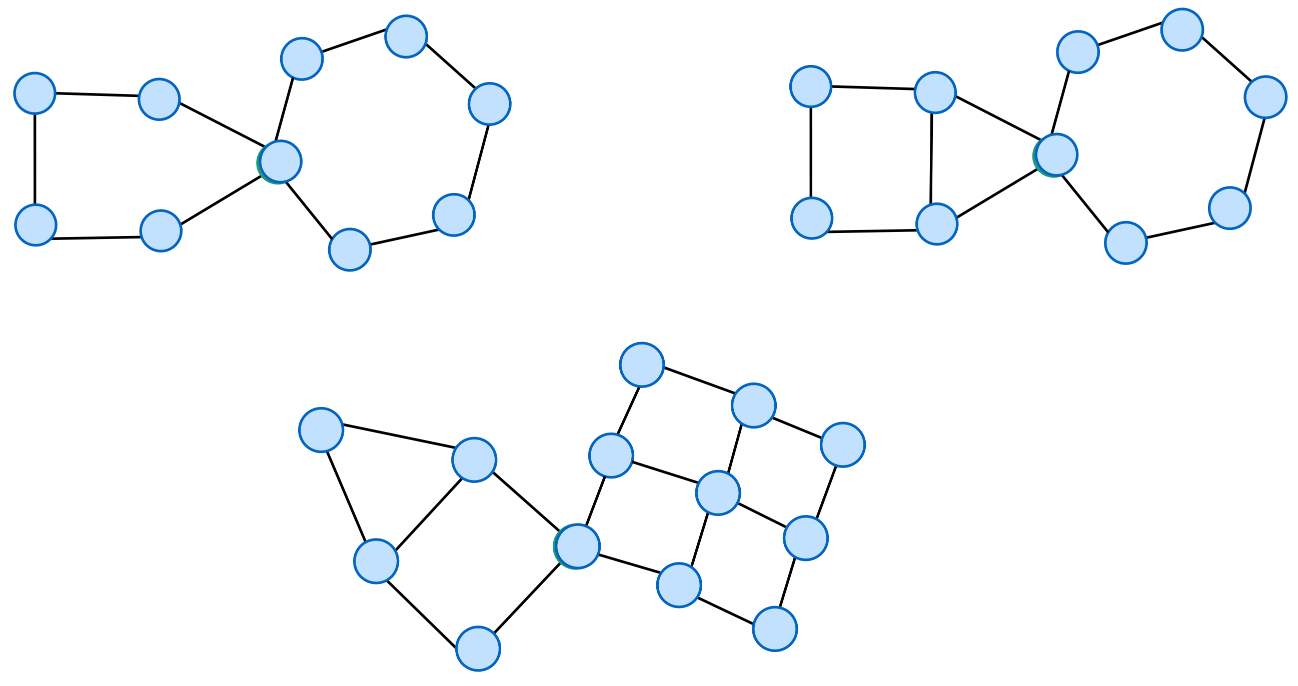
The “Free Lunch Dilemma” in OOD Generalization on Graphs

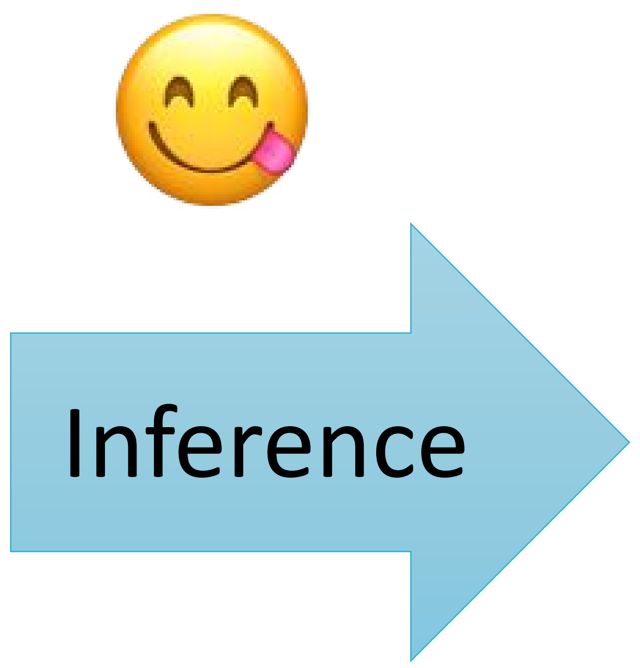
Another line of works aim to **infer environment labels** for learning the underlying invariance:

Environment #?: Class “House”

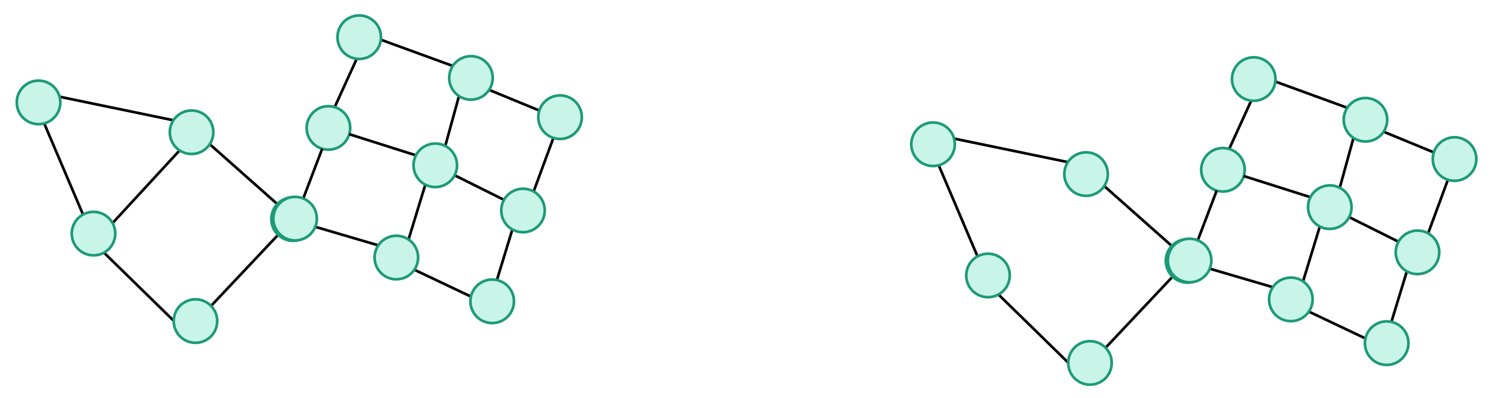


Environment #?: Class “House”

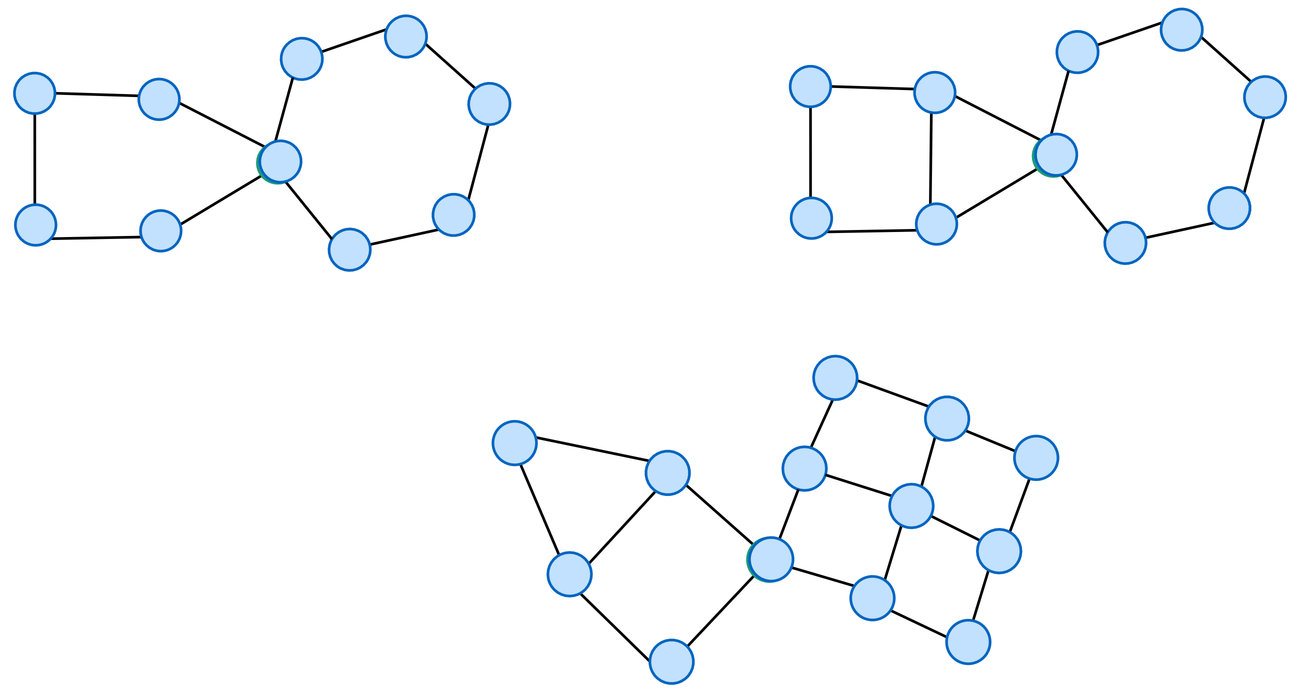




Environment #1: Class “House”



Environment #2: Class “House”

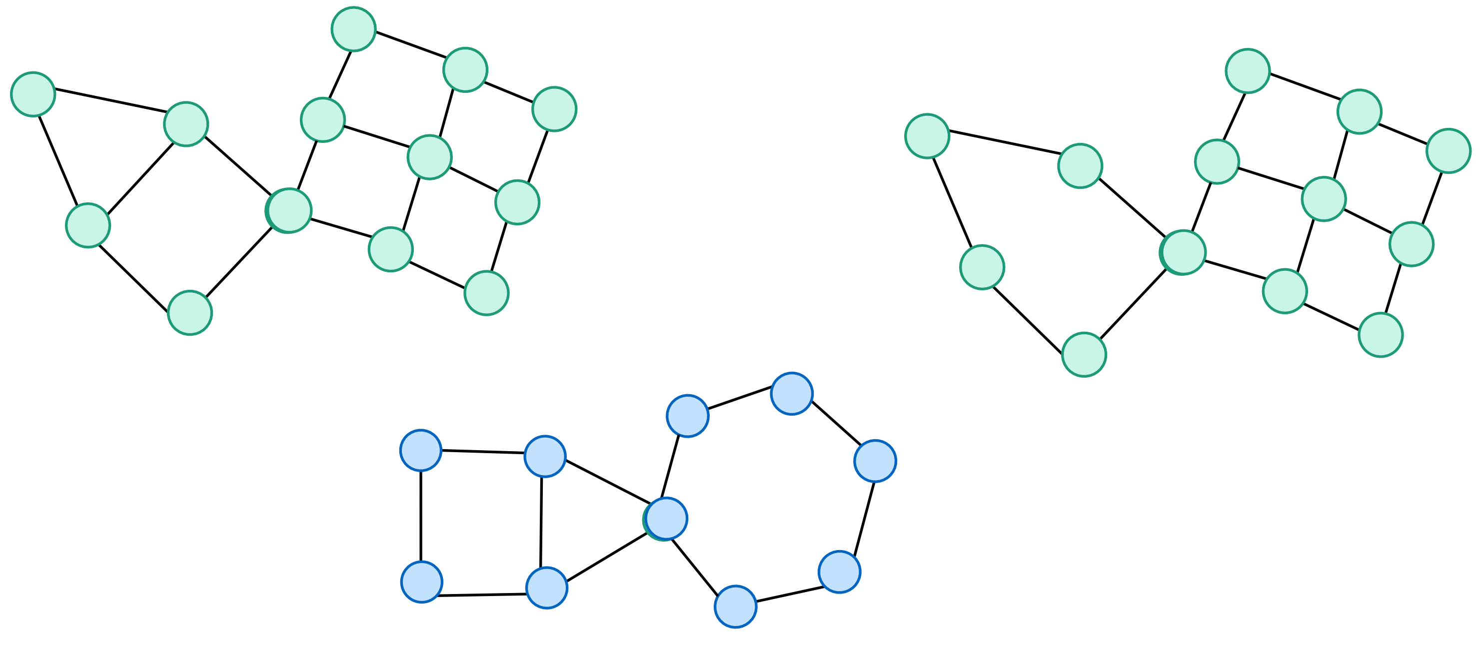


(Li et al., 2022; Yang et al., 2022)

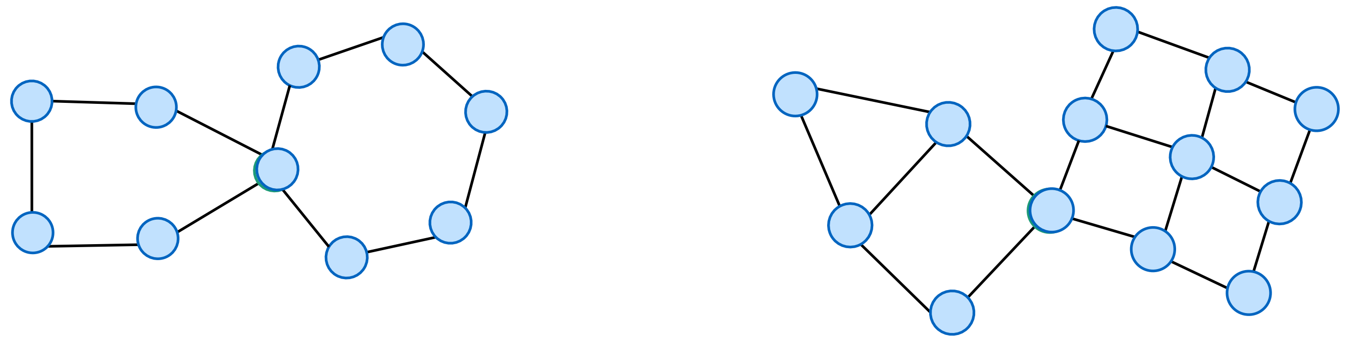
The “Free Lunch Dilemma” in OOD Generalization on Graphs

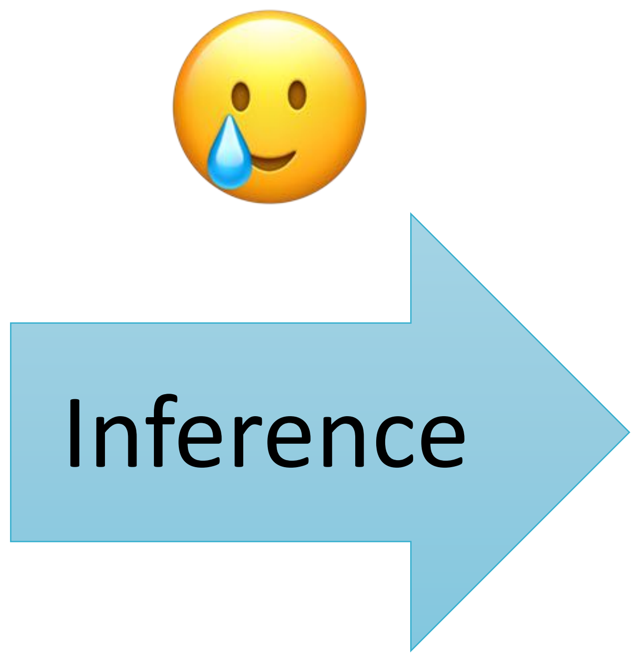
Another line of works aim to **infer environment labels** for learning the underlying invariance:

Environment #?: Class “Grid”



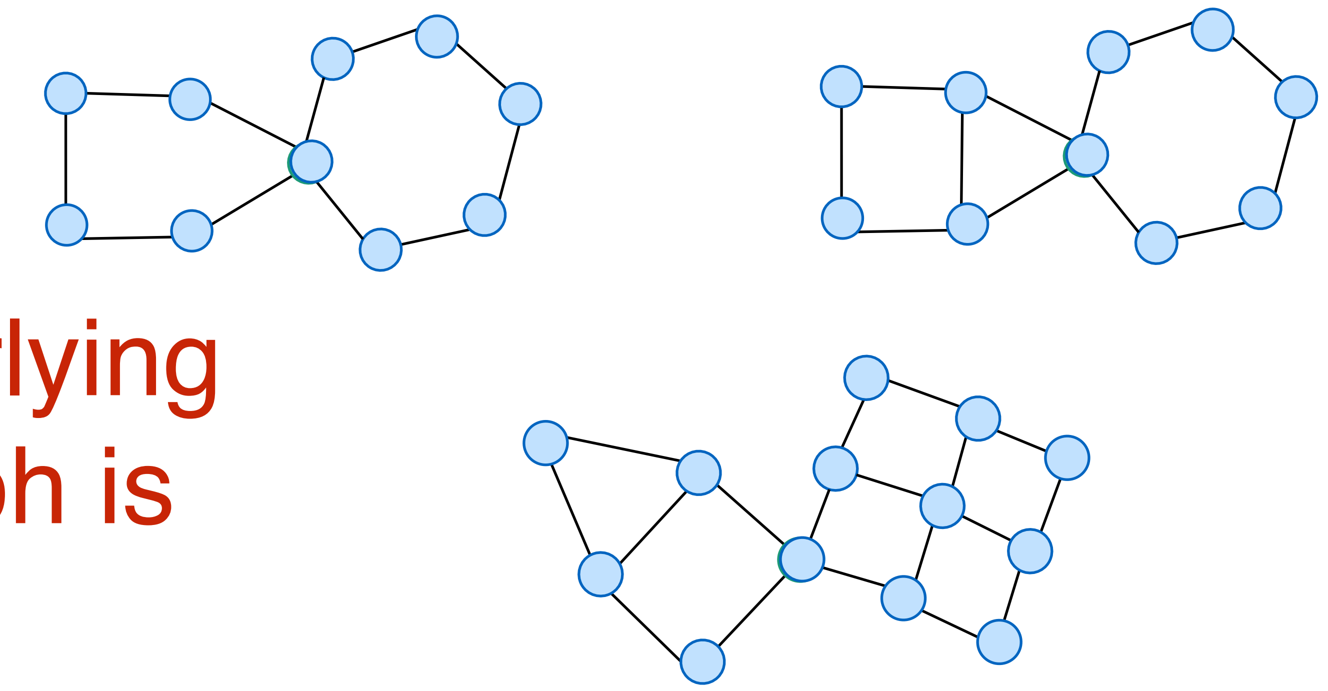
Environment #?: Class “Grid”





Environment #1: Class “Grid”

Environment #2: Class “Grid”



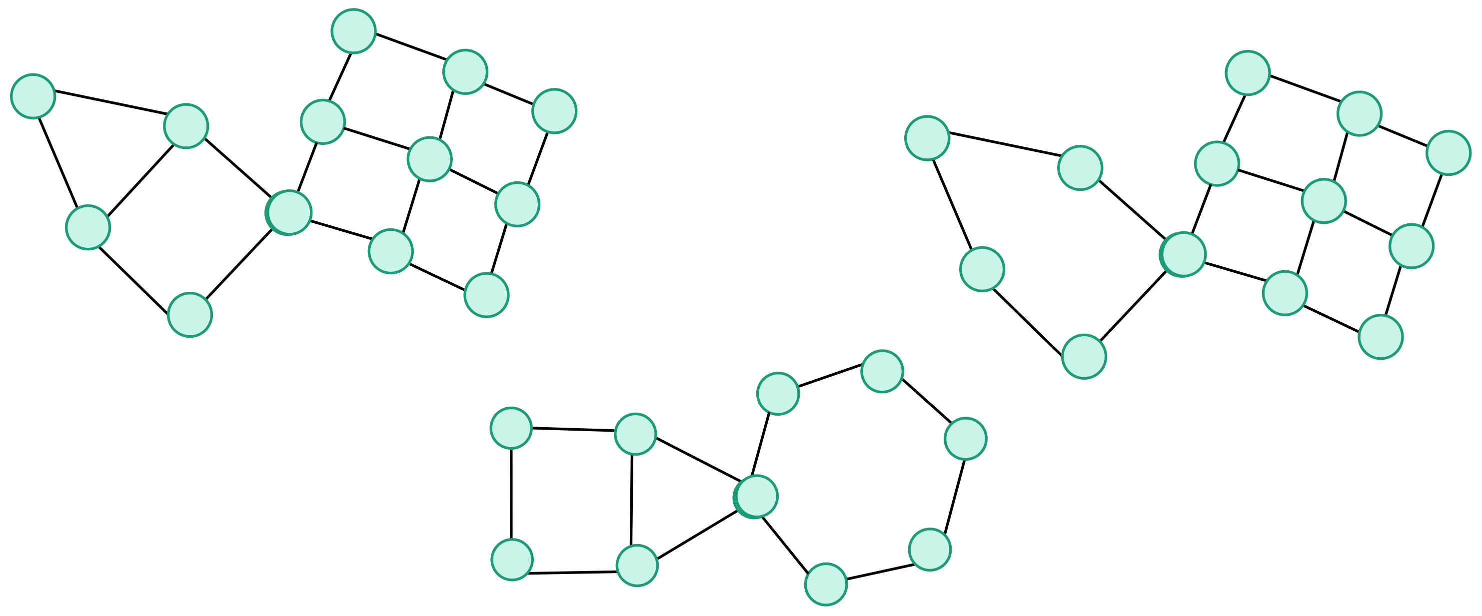
What if the underlying invariant subgraph is REVERSED?

(Li et al., 2022; Yang et al., 2022)

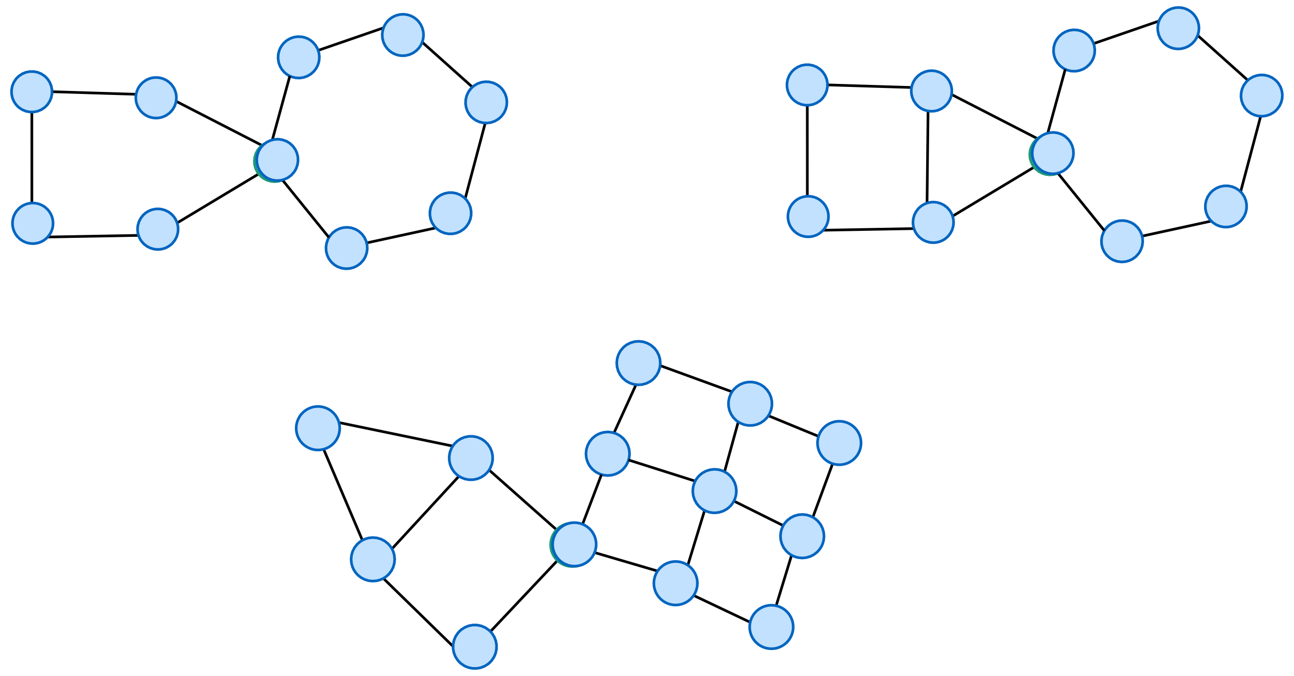
Impossibility Results for OOD Generalization on Graphs

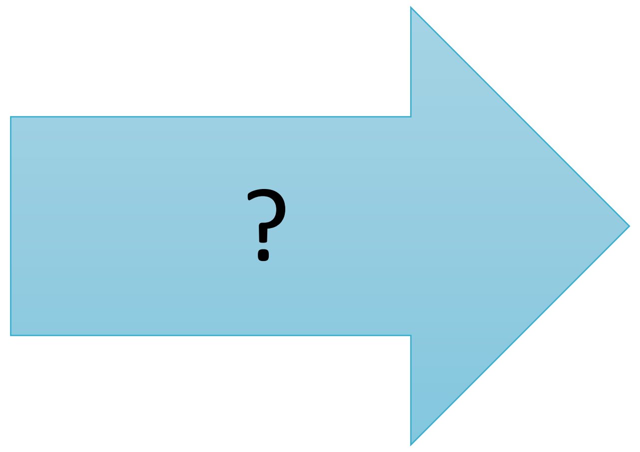
OOD generalization on graphs is fundamentally more **challenging** than that on Euclidean data:

Environment #?: Class "House"



Environment #?: Class "House"





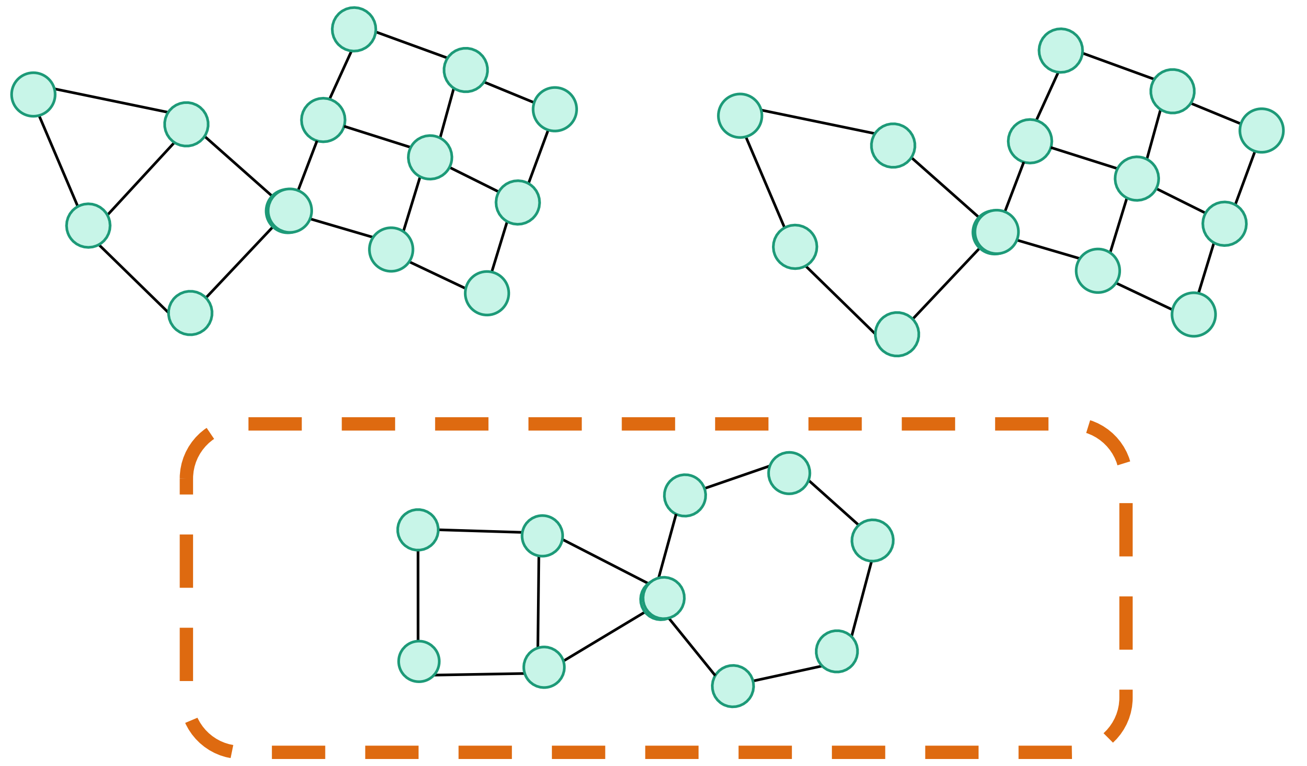
No Free Lunch in Graph OOD (Informal)
It is fundamentally ***impossible*** to identify the underlying invariant subgraph without further inductive biases.

What are the minimal sufficient inductive biases for invariant graph representation learning?

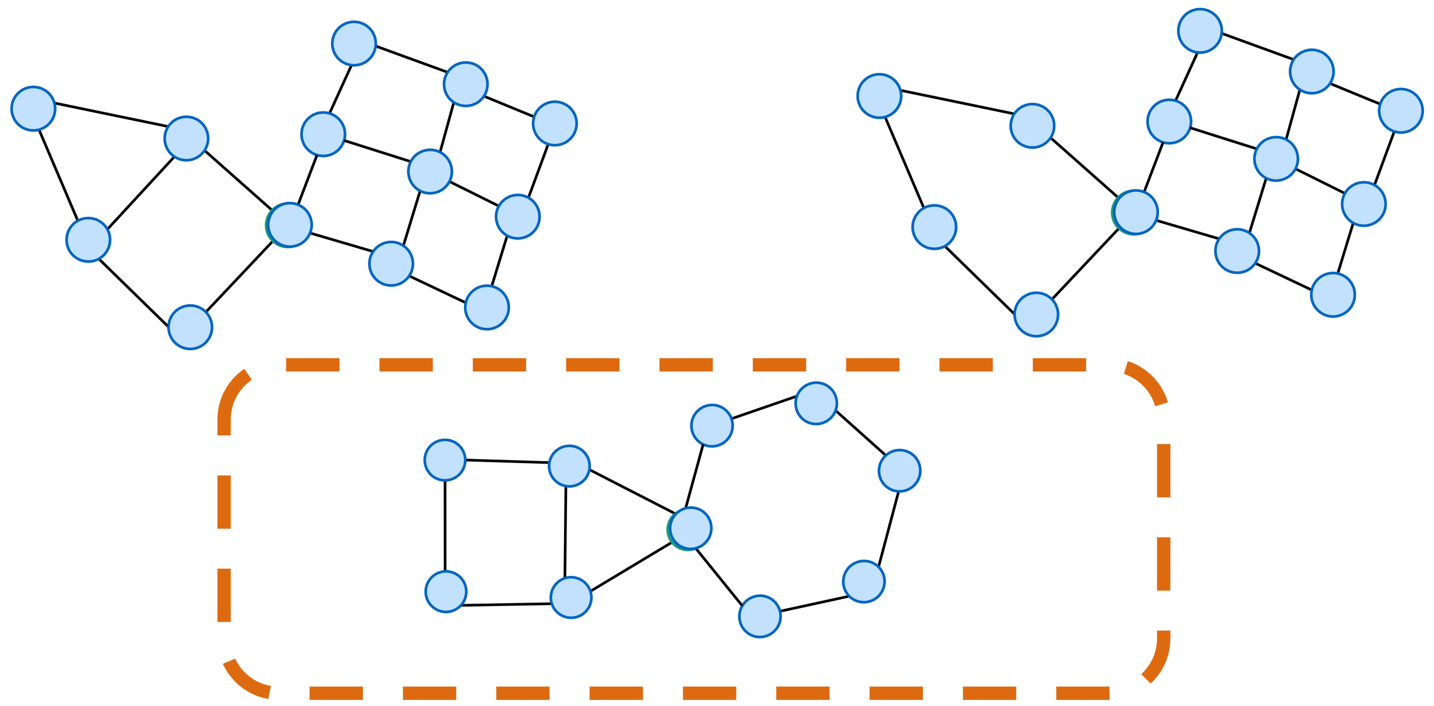
Failures of Environment Generation

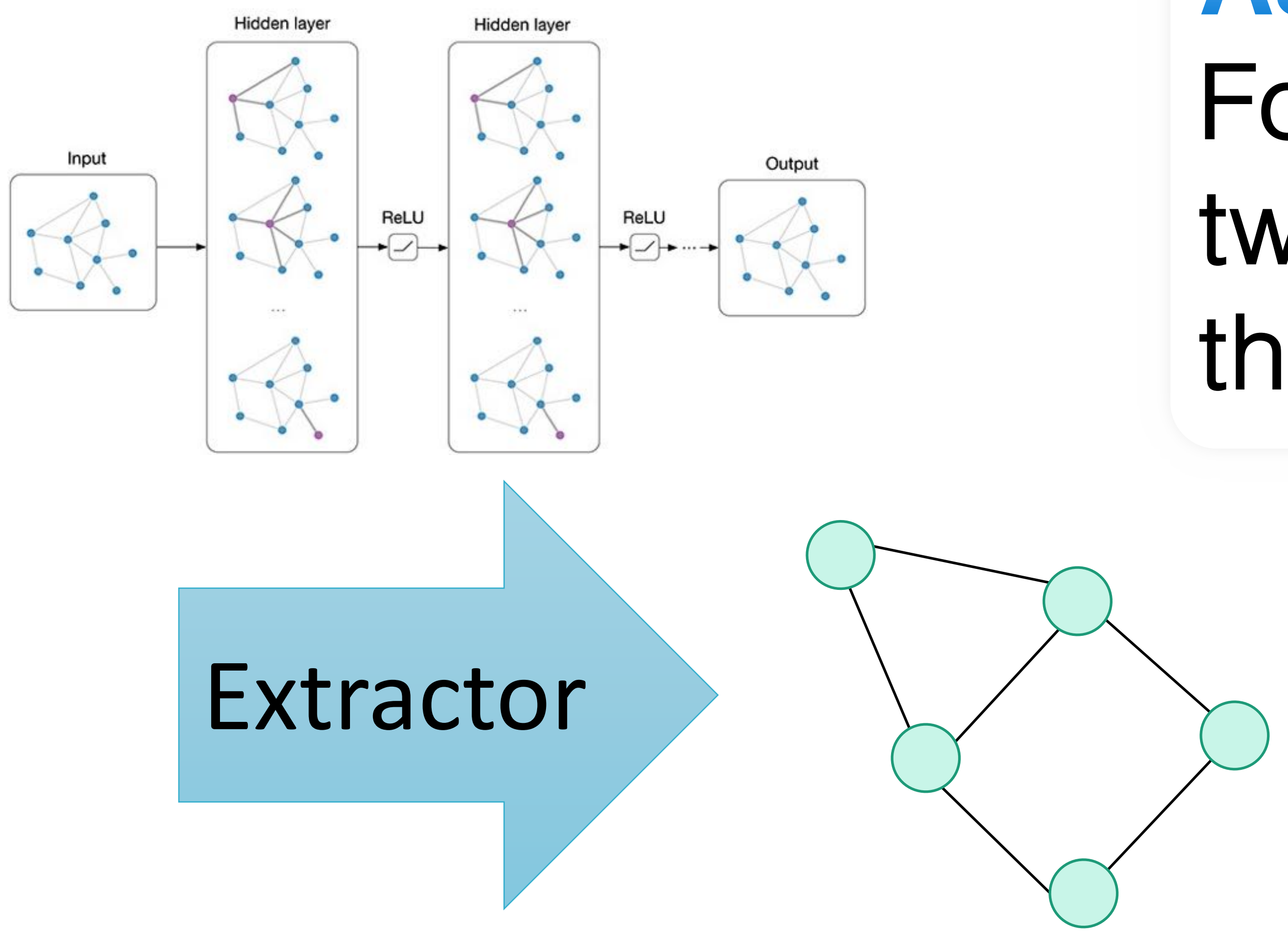
How can we address **environments generation** failures?

Environment #?: Class "House"



Environment #?: Class "House"





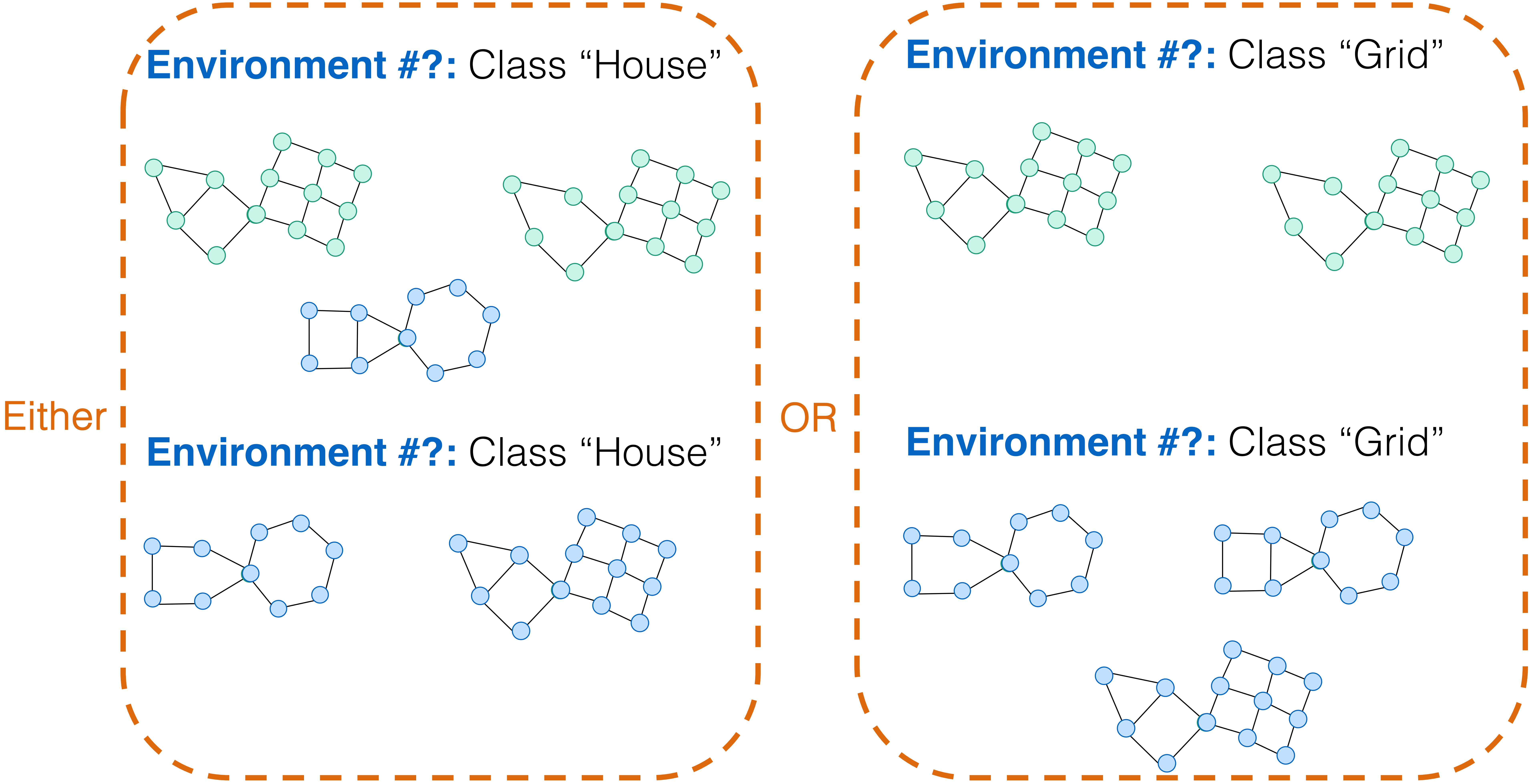
Extracted Invariant Subgraph

Assumption 1 (Variation Sufficiency)
For any spurious subgraph, there exists two underlying environments, such that the spurious correlation varies.

(Wu et al., 2022ab; Liu et al., 2022)

Failures of Environment Inference

How can we address **environment inference** failures?



Assumption 2 (Variation Consistency)
For all environments, either spurious correlation is stronger or weaker.

(Li et al., 2022; Yang et al., 2022)

Invariant Graph Learning with Minimal Assumptions

How can we address **environment inference** failures?

Assumption 1 (Variation Sufficiency)
For any spurious subgraph, there exists two underlying environments, such that the spurious correlation varies.

Environment Generation?

🥲 More assumptions needed!

Environment Inference?

Assumption 2 (Variation Consistency)
For all environments, either spurious correlation is stronger or weaker.

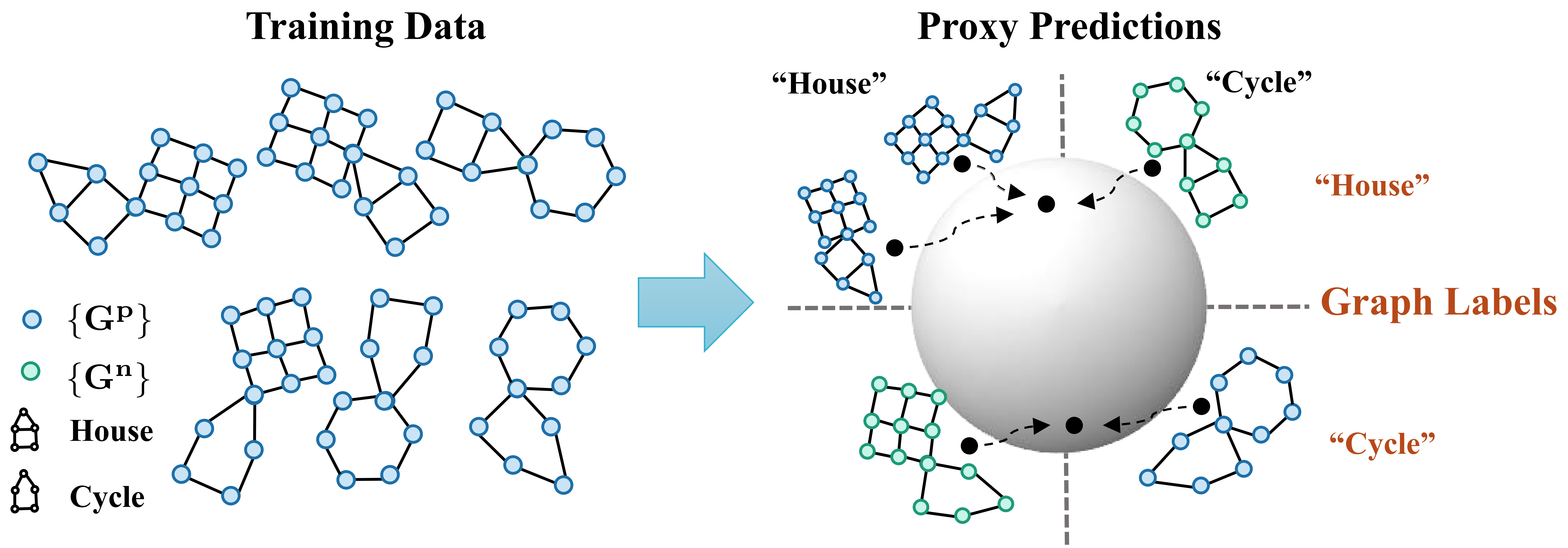
Spurious correlation stronger:
DisC (Fan et al., 2022)

Invariant correlation stronger:
CIGA (Chen et al., 2022)

(*Lin et al., 2022; Fan et al., 2022; Chen et al., 2022)*

GALA: invariant GrAph Learning Assistant

Improving the contrastive invariant subgraph extraction via an **Environment Assistant**:



Proof-of-Concept Experiments

Theorem 1 (Informal)
Given the same data generation process, and the aforementioned variation sufficiency and variation consistency assumptions, when the environment assistant model learns properly distinguishes the variations of the spurious subgraphs, GALA provably identifies the invariant subgraph for OOD generalization.

Datasets	$\{0.8, 0.6\}$	$\{0.8, 0.7\}$	$\{0.8, 0.9\}$	$\{0.7, 0.9\}$	Avg.
ERM	77.33±0.47	75.65±1.62	51.37±1.20	42.73±3.82	61.77
IRM	78.32±0.70	75.13±0.77	50.76±2.56	41.32±2.50	61.38
V-Rex	77.69±0.38	74.96±1.40	49.47±3.36	41.65±2.78	60.94
IB-IRM	78.00±0.68	73.93±0.79	50.93±1.87	42.05±0.79	61.23
EIIL	76.98±1.24	74.25±1.74	51.45±4.92	39.71±2.64	60.60
XGNN	83.84±0.59	83.05±0.20	53.37±1.32	38.28±1.71	64.63
GREA	82.86±0.50	82.72±0.50	50.34±1.74	39.01±1.21	63.72
GSAT	80.54±0.88	78.11±1.23	48.63±2.18	36.62±0.87	63.32
CAL	76.98±6.03	62.95±8.58	51.57±6.33	46.23±3.93	59.43
MoleOOD	49.93±2.25	49.85±7.31	38.49±4.25	34.81±1.65	43.27
GIL	83.51±0.41	82.67±1.18	51.76±4.32	40.07±2.61	64.50
DisC	60.47±17.9	54.29±15.0	45.06±7.82	39.42±8.59	50.81
CIGA	84.03±0.53	83.21±0.30	57.87±3.38	43.62±3.20	67.18
GALA	**84.27**±0.34	**83.65**±0.44	**76.42**±3.53	**72.50**±1.06	**79.21**
Oracle	84.73±0.36	85.42±0.25	84.28±0.15	78.38±0.19	

Stronger spurious correlations

Stronger invariant correlations

Real-World Experiments

GALA consistently improves the OOD generalization performance under various real-world graph distribution shifts on a number of realistic graph benchmarks:

Datasets	EC50-Assay	EC50-Sca	EC50-Size	Ki-Assay	Ki-Sca	Ki-Size	CMNIST-sp	Graph-SST2	Avg.(Rank)[†]
ERM	76.42±1.59	64.56±1.25	61.61±1.52	74.61±2.28	69.38±1.65	76.63±1.34	21.56±5.38	81.54±1.13	65.79 (6.50)
IRM	77.14±2.55	64.32±0.42	62.33±0.86	75.10±3.38	69.32±1.84	76.25±0.73	20.25±3.12	82.52±0.79	65.91 (6.13)
V-Rex	75.57±2.17	64.73±0.53	62.80±0.89	74.16±1.46	71.40±2.77	76.68±1.35	30.71±11.8	81.11±1.37	67.15 (5.25)
IB-IRM	64.70±2.50	62.62±2.05	58.28±0.99	71.98±3.26	69.55±1.66	70.71±1.95	23.58±7.96	81.56±0.82	62.87 (10.6)
EIIL	64.20±5.40	62.88±2.75	59.58±0.96	74.24±2.48	69.63±1.46	76.56±1.37	23.55±7.68	82.46±1.48	64.14 (8.00)
XGNN	72.99±2.56	63.62±1.35	62.55±0.81	72.40±3.05	72.01±1.34	73.15±2.83	20.96±8.00	82.55±0.65	65.03 (7.13)
GREA	66.87±7.53	63.14±2.19	59.20±1.42	73.17±1.80	67.82±4.67	73.52±2.75	12.77±1.71	82.40±1.98	62.36 (10.1)
GSAT	76.07±1.95	63.58±1.36	61.12±0.66	72.26±1.76	70.16±0.80	75.78±2.60	15.24±3.72	80.57±0.88	64.35 (8.63)
CAL	75.10±2.71	64.79±1.58	63.38±0.88	75.22±1.73	71.08±4.83	72.93±1.71	23.68±4.68	82.38±1.01	66.07 (5.38)
DisC	61.94±7.76	54.10±5.69	57.64±1.57	54.12±8.53	55.35±10.5	50.83±9.30	50.26±0.40	76.51±2.17	56.59 (12.4)
MoleOOD	61.49±2.19	62.12±1.91	58.74±1.73	75.10±0.73	60.35±11.3	73.69±2.29	21.04±3.36	81.56±0.35	61.76 (10.0)
GIL	70.56±4.46	61.59±3.16	60.46±1.91	75.25±1.14	70.07±4.31	75.76±2.23	12.55±1.26	83.31±0.50	63.69 (8.00)
CIGA	75.03±2.47	65.41±1.16	64.10±1.08	73.95±2.50	71.87±3.32	74.46±2.32	15.83±2.56	82.93±0.63	65.45 (5.88)
GALA	**77.56**±2.88	**66.28**±0.45	**64.25**±1.21	**77.92**±2.48	**73.17**±0.88	**77.40**±2.04	**68.94**±0.56	**83.60**±0.66	**73.64 (1.00)**
Oracle	84.77±0.58	82.66±1.19	84.53±0.60	91.08±1.43	88.58±0.64	92.50±0.53	67.76±0.60	91.40±0.26	

[†] Averaged rank is also reported in the parentheses because of dataset heterogeneity. A lower rank is better.

Learning Causality for Modern Machine Learning

Traditional ML assumes train and test data are **iid.**, i.e., independently sampled from an identical distribution, while data is often **OOD**, i.e., out-of-distribution, in real-world applications.

Objectives

Causal Representation Learning on Graphs: [NeurIPS’22 Spotlight, NeurIPS’23a]

Implications

Useful Properties of the Causal Representations:
OOD Generalizability [NeurIPS’22, 23a],
Adversarial Robustness [ICLR’22],
Interpretability [ICML’24a]

Realizations

Optimization & Feature Learning schemes for Causal Representation Learning: [ICLR’23a, NeurIPS’23b]

Learning Causality for Modern Machine Learning

Traditional ML assumes train and test data are **iid.**, i.e., independently sampled from an identical distribution, while data is often **OOD**, i.e., out-of-distribution, in real-world applications.

Objectives

Causal Representation Learning on Graphs: [NeurIPS'22 Spotlight, NeurIPS'23a]

Implications

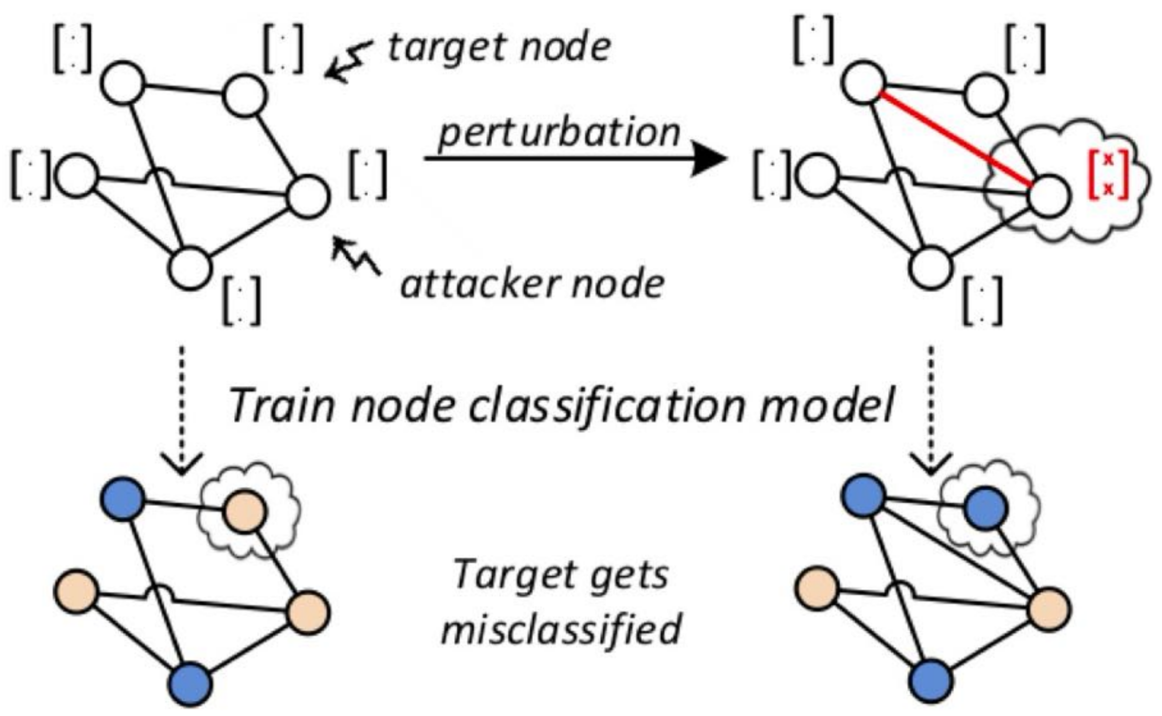
Useful Properties of the Causal Representations:
OOD Generalizability [NeurIPS'22, 23a],
Adversarial Robustness [ICLR'22],
Interpretability [ICML'24a]

Realizations

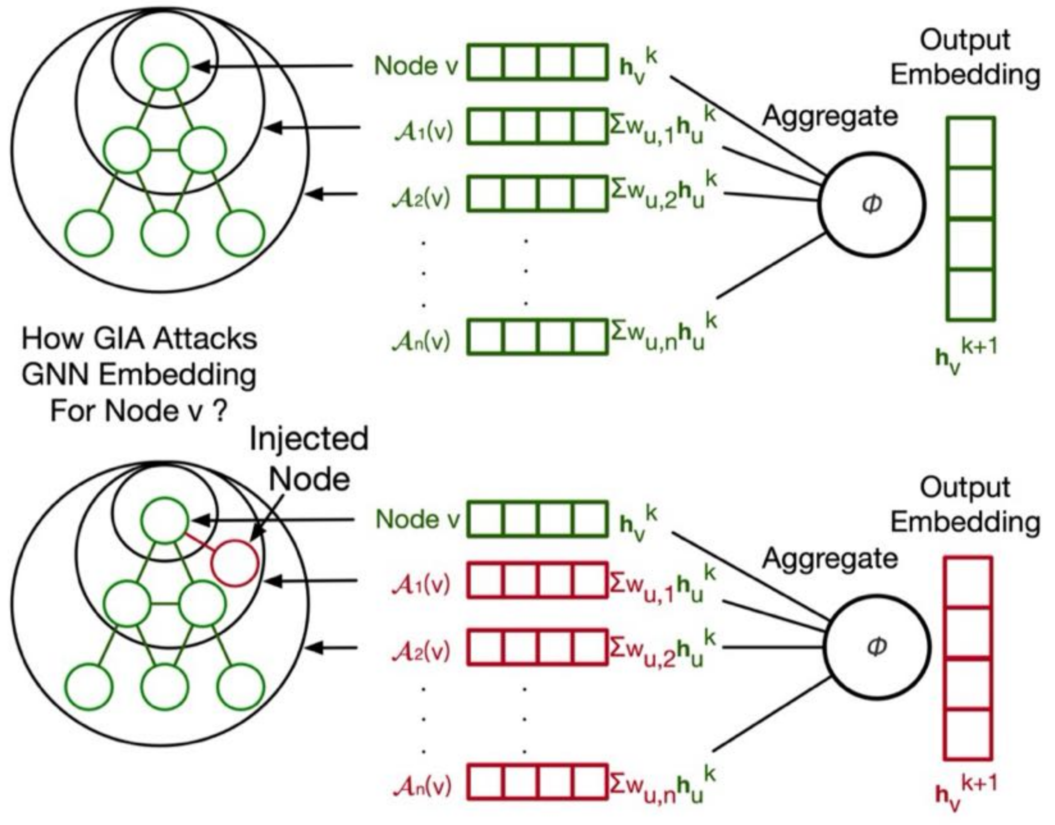
Optimization & Feature Learning schemes for Causal Representation Learning: [ICLR'23a, NeurIPS'23b]

Adversarial Attack on Graph Neural Networks

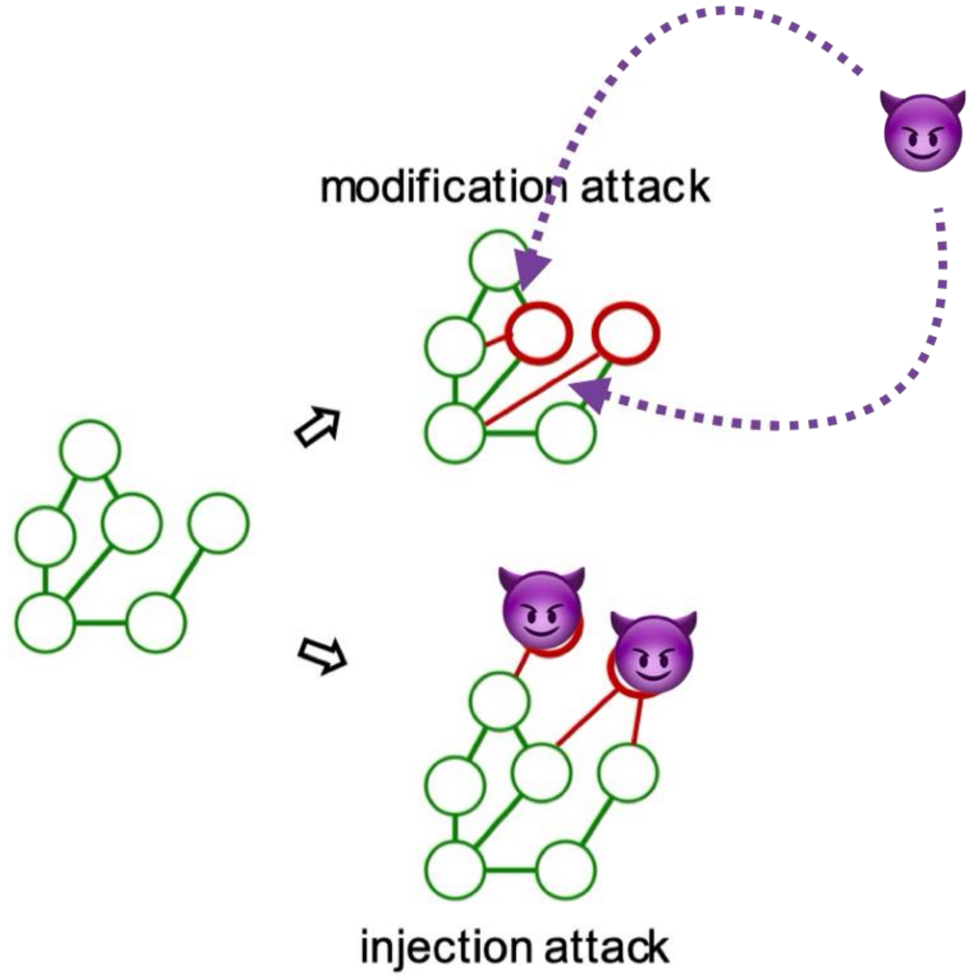
Graph adversarial attacks aim to degenerate the performance by maliciously perturbing graphs:



(Zügner et al., 2018)



(Zou et al., 2020)



GMA vs. GIA

Adversarial Objective:

$$\min \mathcal{L}_{\text{atk}}(f_{\theta^*}(G')), \ \ \text{s.t.} \|G' - G\| \le \underbrace{\triangle}_{\text{perturbation budgets}}$$

Graph Modification Attack (GMA):

$$\triangle_A + \triangle_X \le \triangle \in \mathbb{Z}, \ \|A' - A\|_0 \le \triangle_A \in \mathbb{Z}, \ \|X' - X\|_\infty \le \epsilon \in \mathbb{R}$$

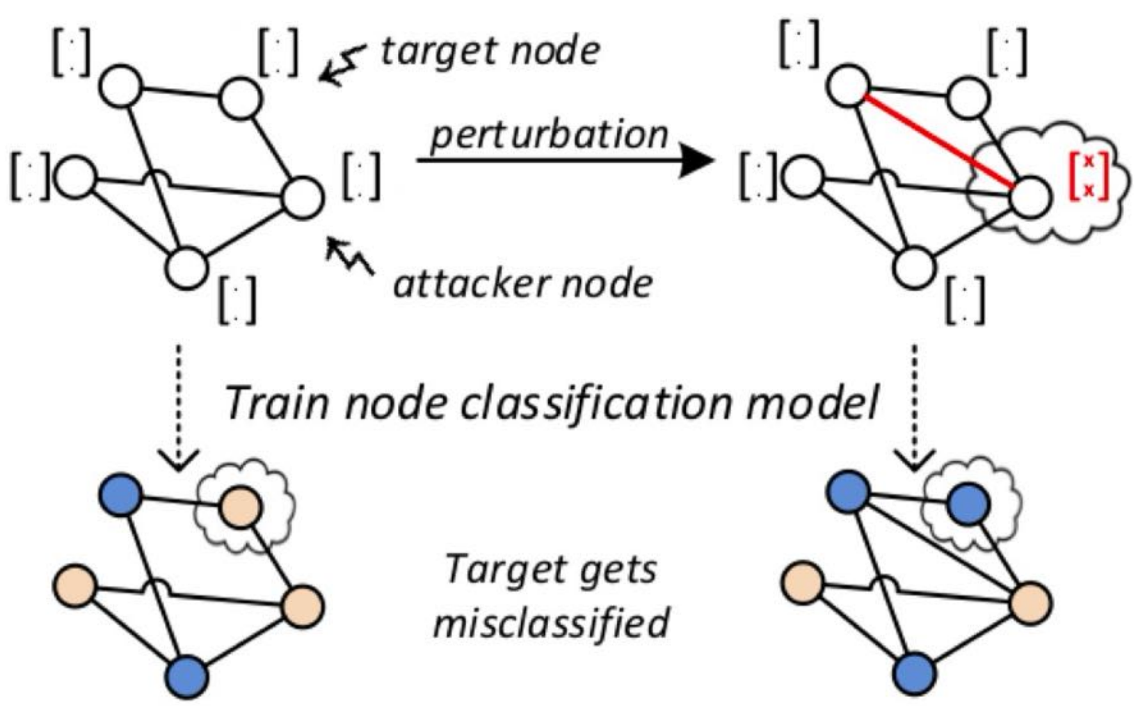
Sometimes Expensive

Modifying edges

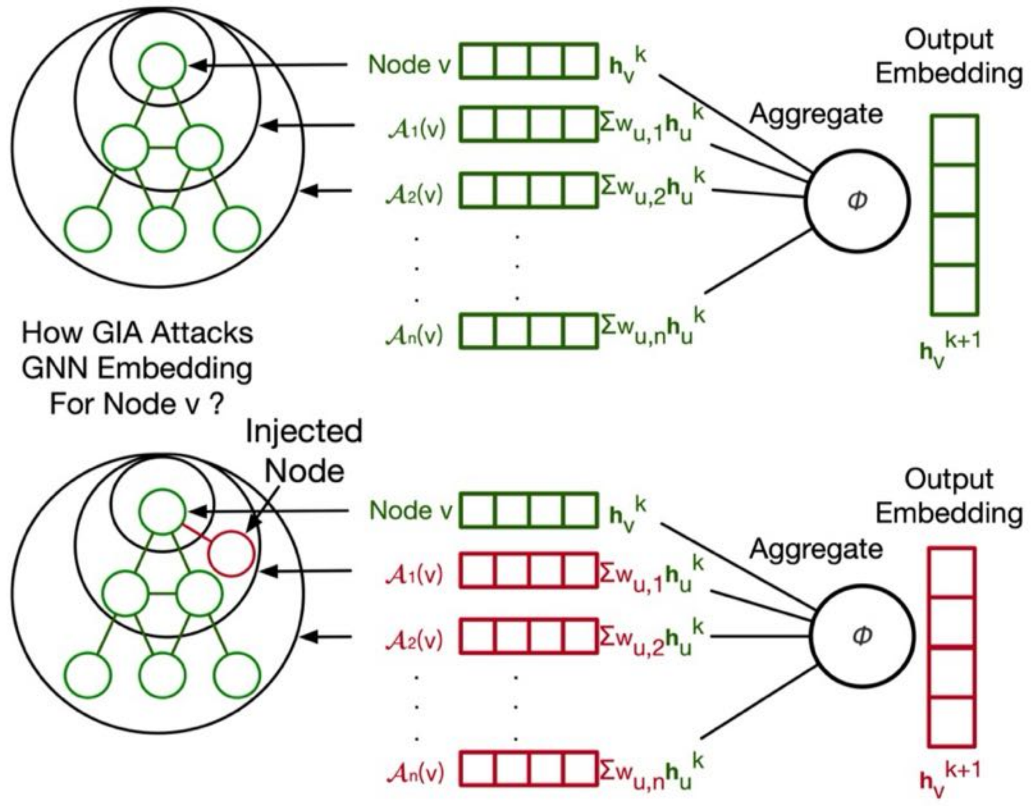
Perturbing node features

Adversarial Attack on Graph Neural Networks

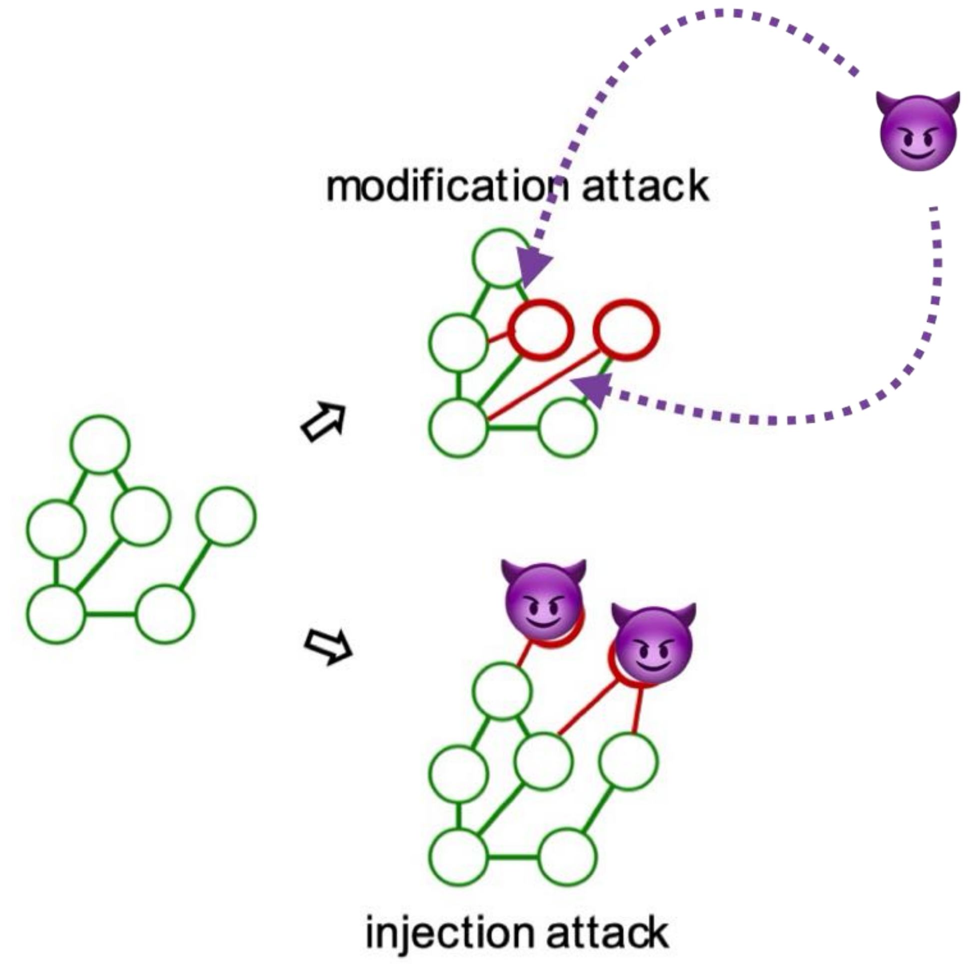
Graph adversarial attacks aim to degenerate the performance by maliciously perturbing graphs:



(Zügner et al., 2018)



(Zou et al., 2020)



GMA vs. GIA

Adversarial Objective:

$$\min \mathscr{L}_{\mathrm{atk}}(f_{\theta^*}(G')),\ \ \mathrm{s.t.} \|G' - G\| \leq \underbrace{\triangle}_{\text{perturbation budgets}}$$

Graph Injection Attack (GIA):

$$X' = \begin{bmatrix} X \\ X_{\mathrm{atk}} \end{bmatrix}, A' = \begin{bmatrix} A & A_{\mathrm{atk}} \\ A_{\mathrm{atk}}^T & O_{\mathrm{atk}} \end{bmatrix}, \quad |V_{\mathrm{atk}}| \leq \triangle \in \mathbb{Z},\ 1 \leq d_u \leq b \in \mathbb{Z}, X_u \in \mathscr{D}_X \subseteq \mathbb{R}^d, \forall u \in V_{\mathrm{atk}}$$

Practical

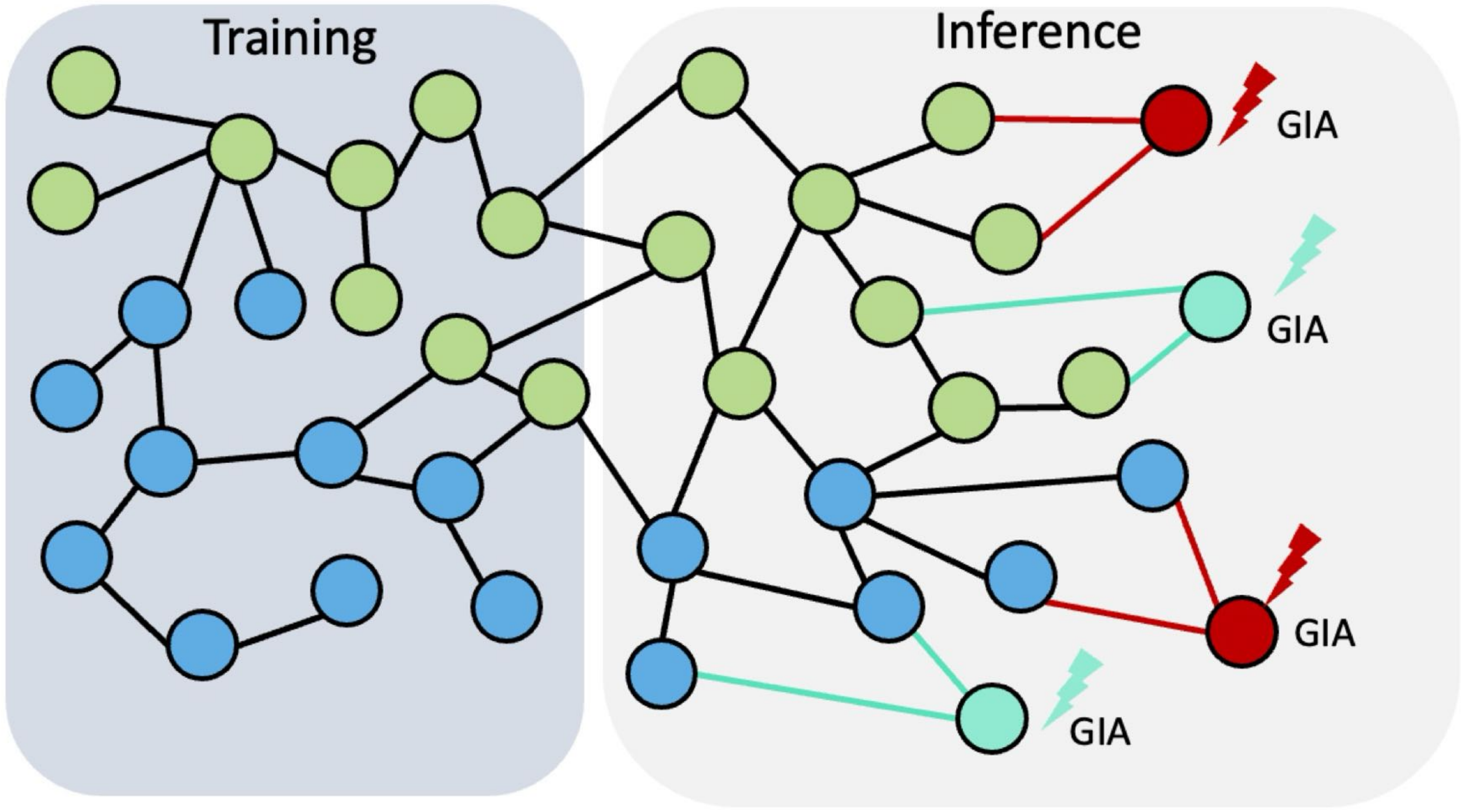
Injecting nodes

Carefully injected connections

Carefully crafted node features

Adversarial Attack on Graph Neural Networks

We compare GMA and GIA in a unified setting.



We adopt a unified setting, which is also used by Graph Robustness Benchmark *(Zheng et al., NeurIPS 2021)*.

Evasion: Attack happens at testing time.

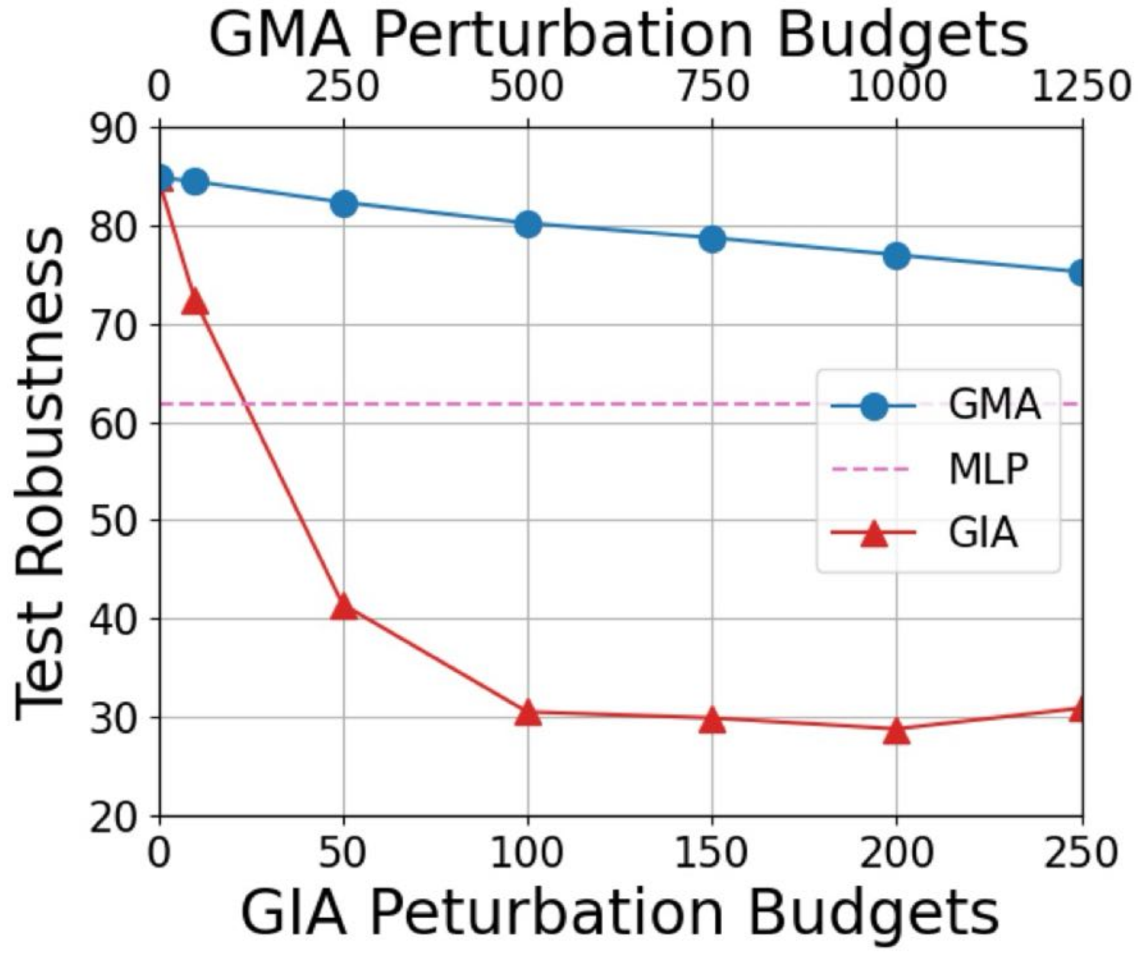
Inductive: Test nodes and corresponding edges are invisible to the model during training, i.e., $G_{\text{train}} \subseteq G,\ G_{\text{test}} = G'$.

Blackbox: The adversary can not access the architecture or the parameters of the target model.

Find out more about the motivation for adopting this setting in our paper :)

Adversarial Attack on Graph Neural Networks

In general, GIA is more powerful than GMA.



GMA vs. GIA

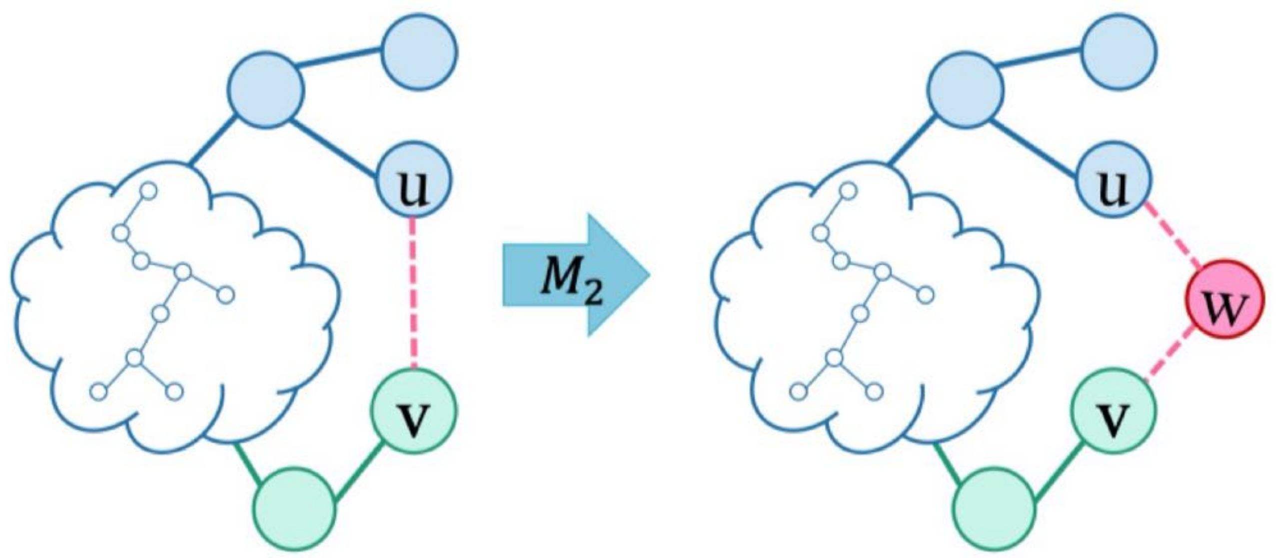
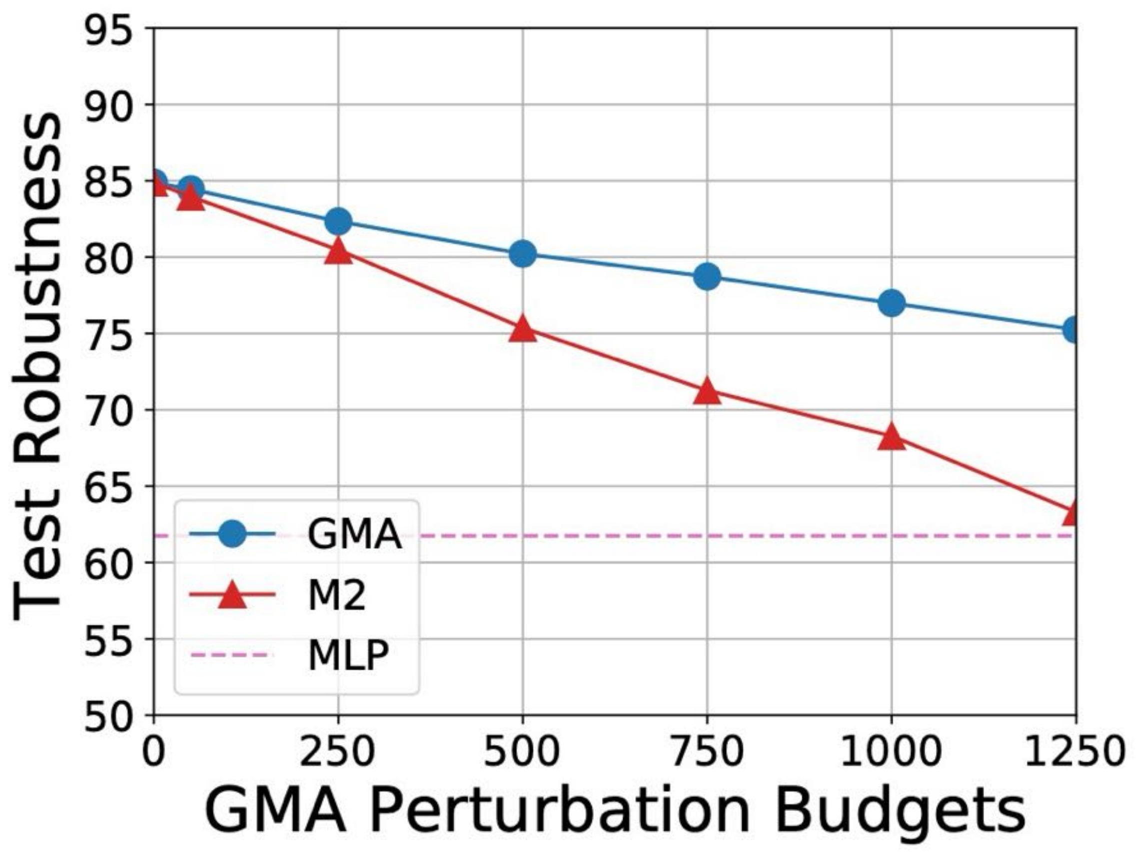


Illustration of $\mathscr{M}_2$ mapping



GMA vs. GIA with $\mathscr{M}_2$

Theorem 1 (GIA is more harmful than GMA)

Given moderate perturbation budgets $\triangle_{\text{GIA}}$ for GIA and $\triangle_{\text{GMA}}$ for GMA, that is, let $\triangle_{\text{GIA}} \leq \triangle_{\text{GMA}} \ll |V| \leq |E|$, for a fixed linearized GNN f_θ trained on G, assume that G has no isolated nodes, and both GIA and GMA follow the **optimal strategy**, then,
$\forall \triangle_{\text{GMA}} \geq 0, \exists \triangle_{\text{GIA}} \leq \triangle_{\text{GMA}}$,

$$\mathscr{L}_{\text{atk}}(f_\theta(G'_{\text{GIA}})) - \mathscr{L}_{\text{atk}}(f_\theta(G'_{\text{GMA}})) \leq 0,$$

where G'_{GIA} and G'_{GMA} are perturbed graphs generated by GIA and GMA, respectively.

Adversarial Attack on Graph Neural Networks

In general, GIA is more powerful than GMA. But, what is the price?

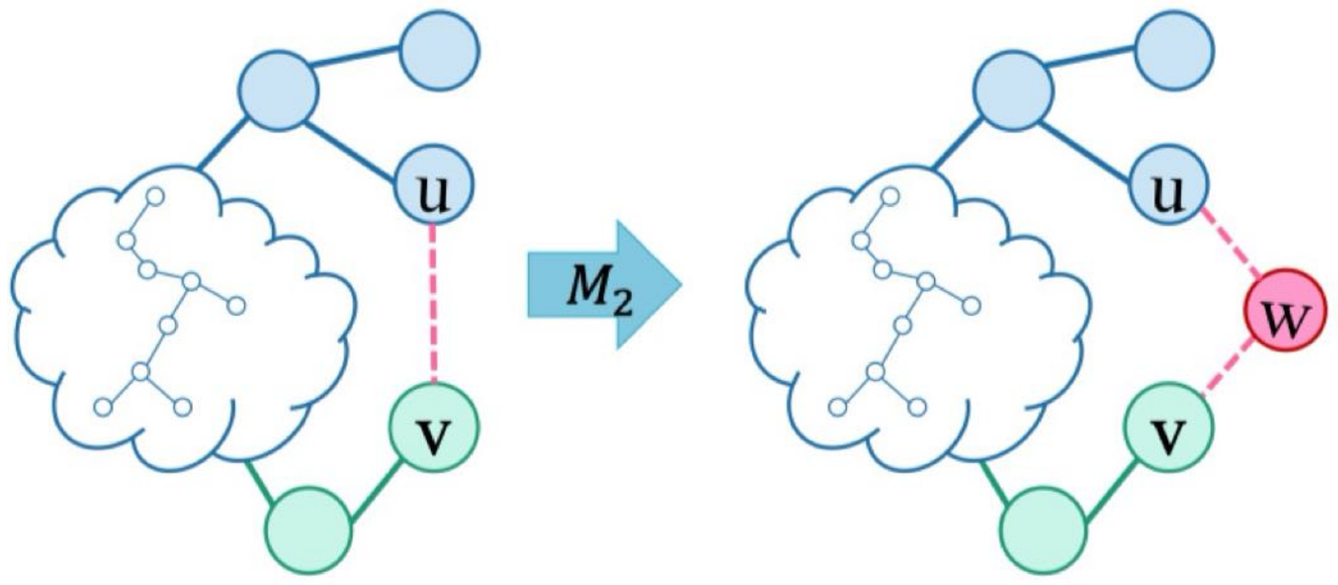


Illustration of $\mathscr{M}_2$ mapping

Given the example of $\mathscr{M}_2$, assume GIA uses PGD to optimize X_w iteratively, we find:

$$\mathrm{sim}(X_u, X_w)^{(t+1)} \leq \mathrm{sim}(X_u, X_w)^{(t)},$$

where t is the number of optimization steps and $\mathrm{sim}(\cdot)$ is the cosine similarity.

Definition 3 (Node-Centric Homophily)

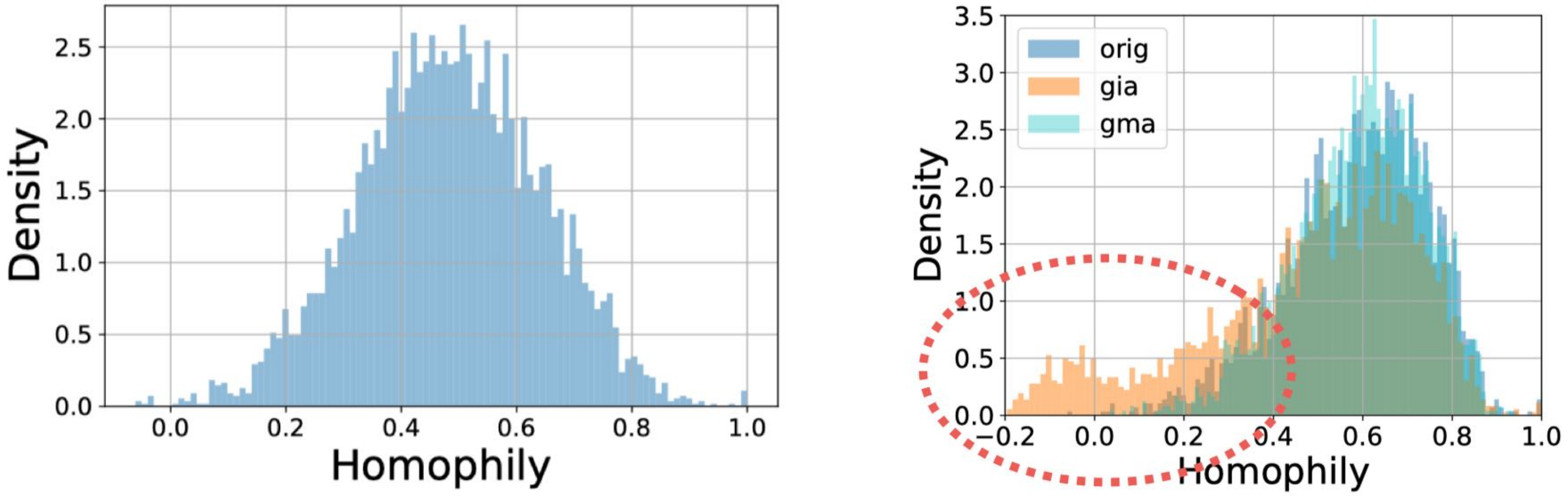
The homophily of a node u can be defined with the similarity between the features of node u and the aggregated features of its neighbors*:

$$h_u = \mathrm{sim}(r_u, X_u),\ r_u = \sum_{j \in \mathcal{N}(u)} \frac{1}{\sqrt{d_j d_u}} X_j,$$

where d_u is the degree of node u and $\mathrm{sim}(\cdot)$ is a similarity metric, e.g., cosine similarity.

Adversarial Attack on Graph Neural Networks

In general, GIA is more powerful than GMA. But, what is the price?



Homophily changes before and after attacks

GIA provably leads more damage to the homophily of the original graph than GMA

Definition 3 (Homophily Defenders)

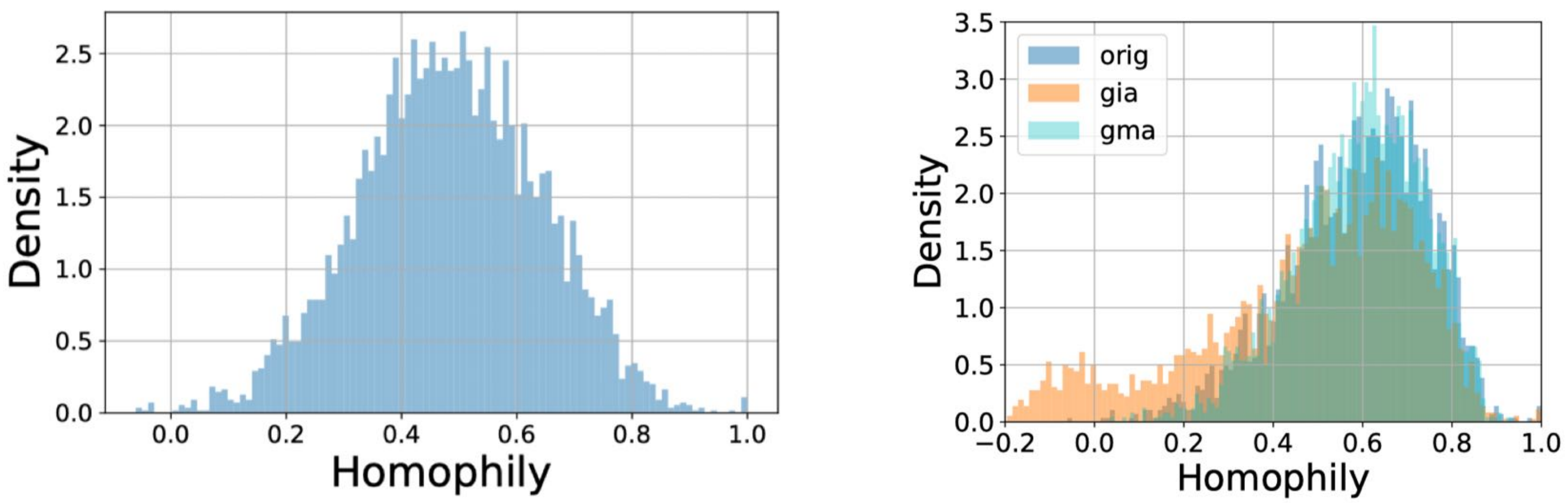
The homophily defenders can be implemented via edge pruning*:

$$H_u^{(k)} = \mathrm{READOUT}(W_k \cdot \mathrm{AGG}(\mathbb{I}_{\mathrm{con}}(u,v)\{H_v^{(k-1)}\} \mid v \in \mathcal{N}(u) \cup \{u\})),$$

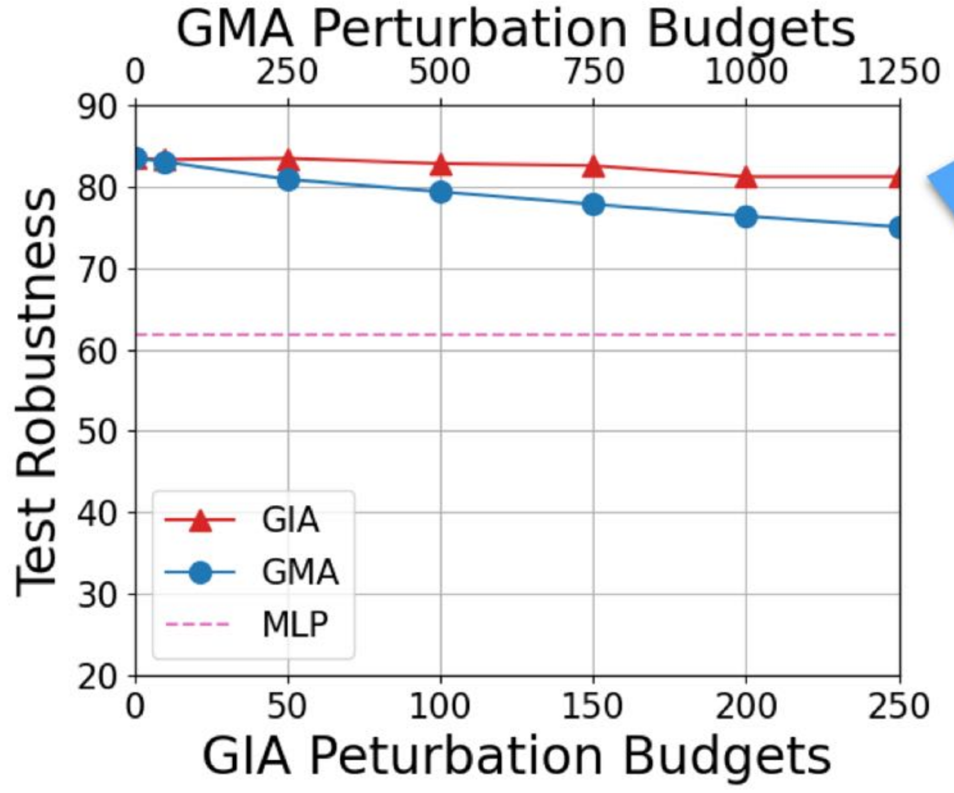
where $\mathbb{I}_{\mathrm{con}}(u,v)$ elaborates the pruning condition for edge (u,v).

Adversarial Attack on Graph Neural Networks

In general, GIA is more powerful than GMA. But, what is the price?



Homophily changes before and after attacks



GMA vs. GIA when with defense

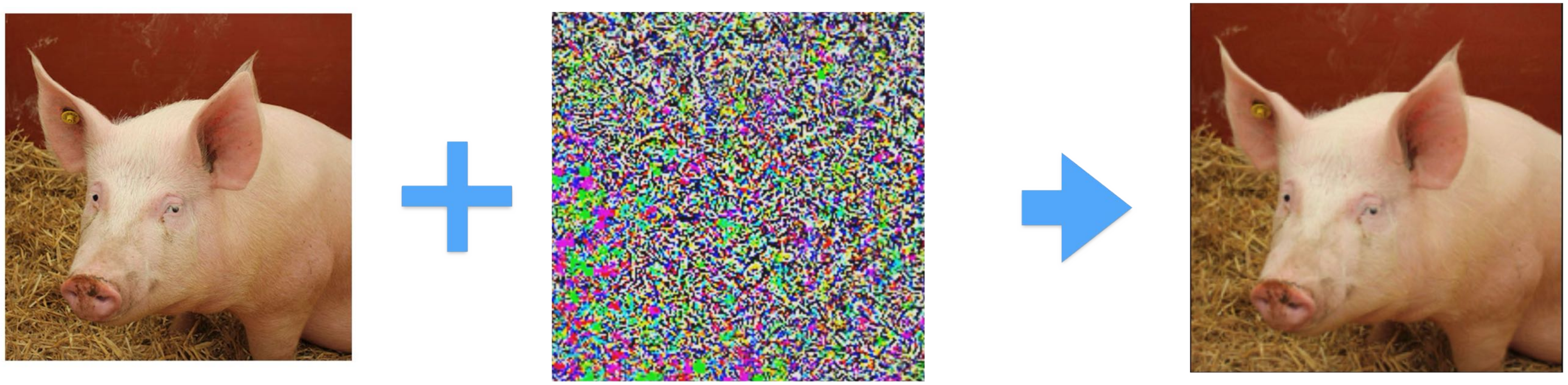
GIA almost loses its power!

Theorem 2 (GIA loses power when against homophily defenders)

Given conditions in Theorem 1, consider a GIA attack, which **(i)** is mapped by $\mathscr{M}_2$ from from a GMA attack that only performs edge addition perturbations, and **(ii)** uses a linearized GNN trained with at least one node from each class in G as the surrogate model, and **(iii)** optimizes the malicious node features with PGD. Assume that G has no isolated node, and has node features as $X_u = \frac{C}{C-1} e_{Y_u} - \frac{1}{C-1}\mathbf{1} \in \mathbb{R}^d$ where Y_u is the label of node u and $e_{Y_u} \in \mathbb{R}^d$ is a one-hot vector with the Y_u-th entry being 1 and others being 0. Let the minimum similarity for any pair of nodes connected in G be $s_G = \min_{(u,v)\in E} \mathrm{sim}(X_u, X_v)$ implemented with cosine similarity. For a homophily defender g_θ that prunes edges (u, v) if $\mathrm{sim}(X_u, X_v) \leq s_G$, we have:

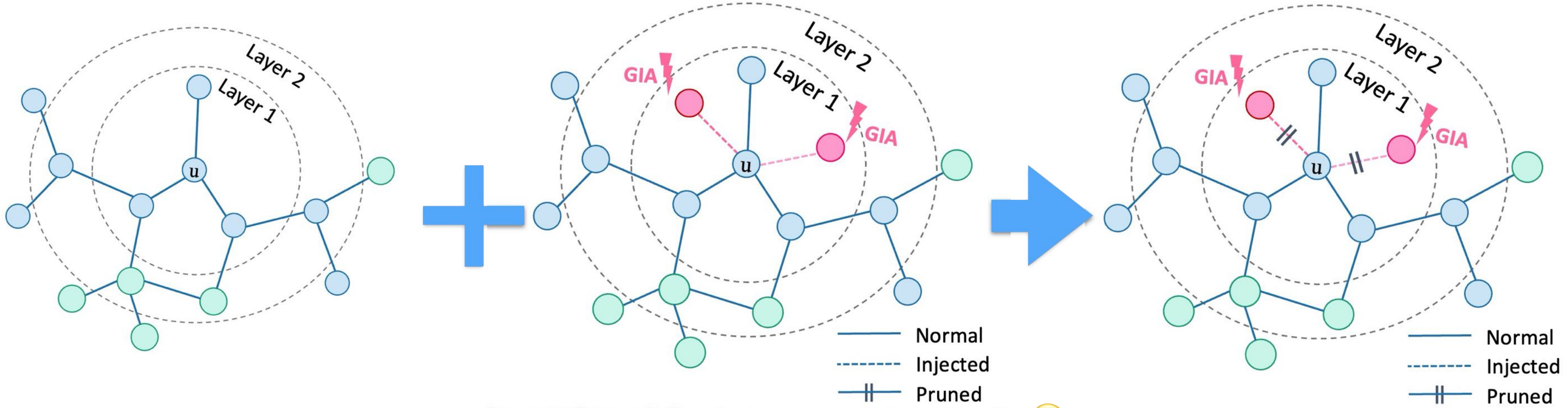
New Definition of Adversarial Attack on Graphs

We rethink the **ill-defined** unnoticeability constraints for prevalent graph adversarial attacks…



Prediction: Pig **Unnoticeable** Adversarial noise 😋 Prediction: Airliner

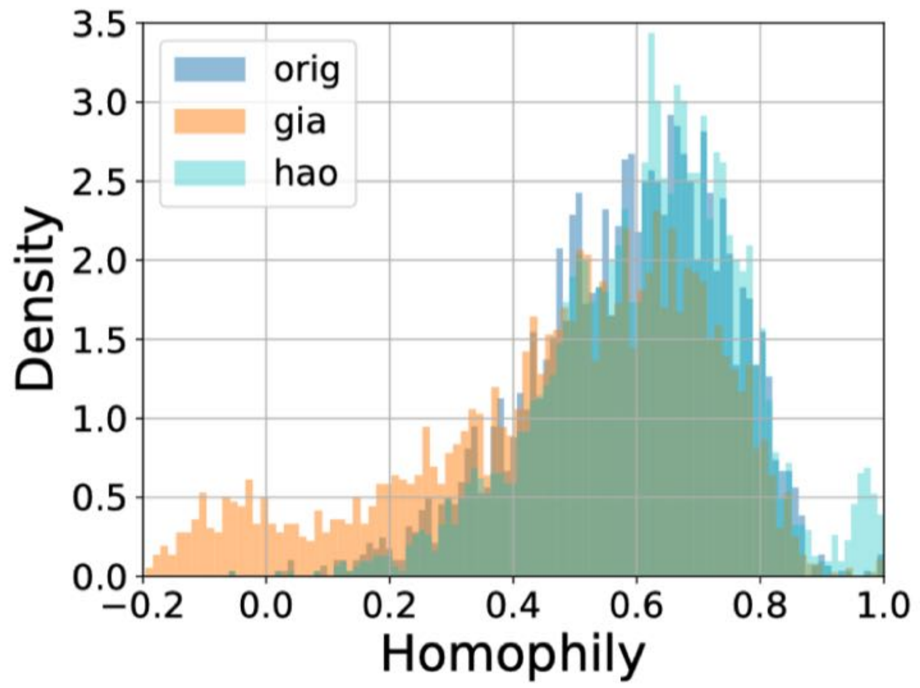
(Szegedy et al., 2014; Goodfellow et al., 2015; Kolter and Madry et al. 2019)



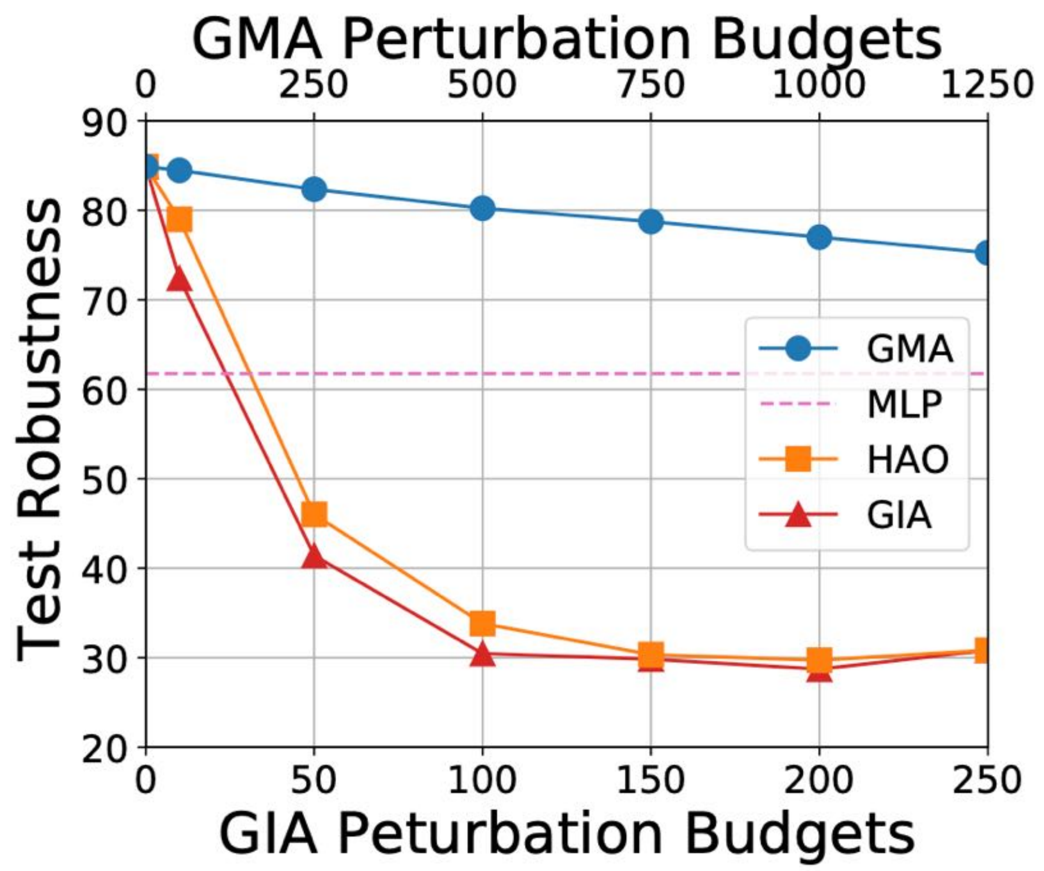
Unnoticeable Adversarial noise? 🤔

HAO: Harmonious Adversarial Objective

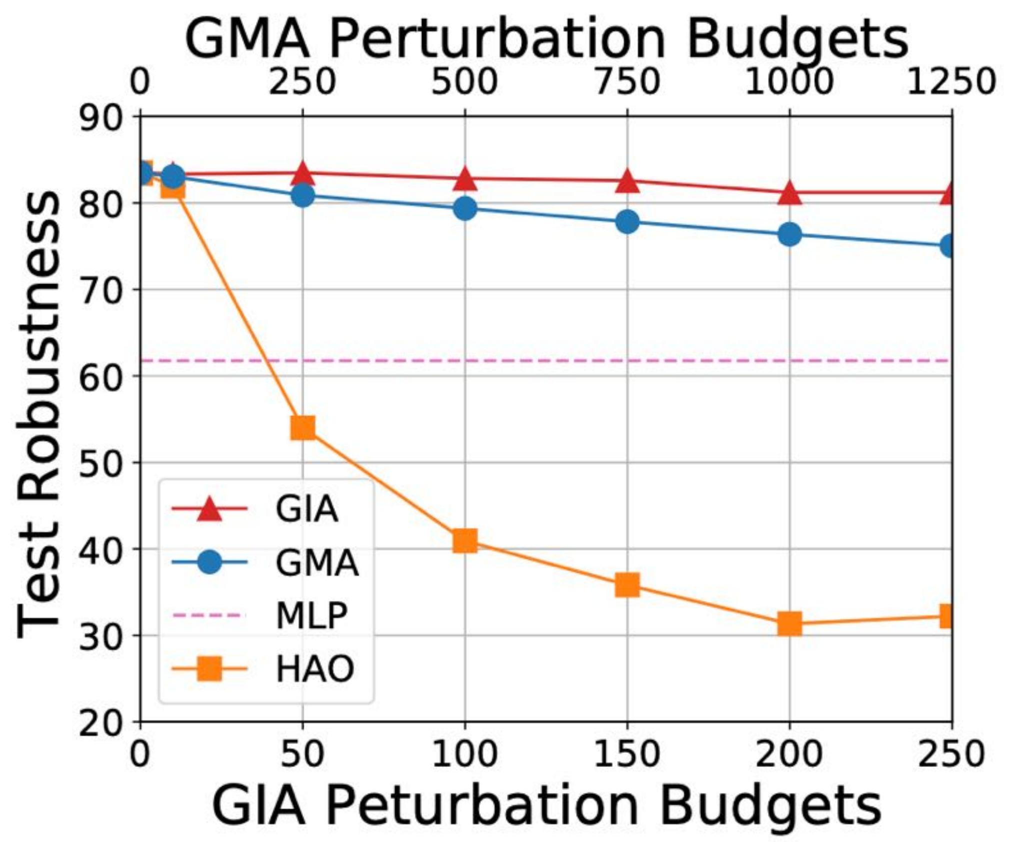
We propose a new objective respecting the homophily constraints.



Homophily changes



GMA vs. GIA without defense



GMA vs. GIA when with defense

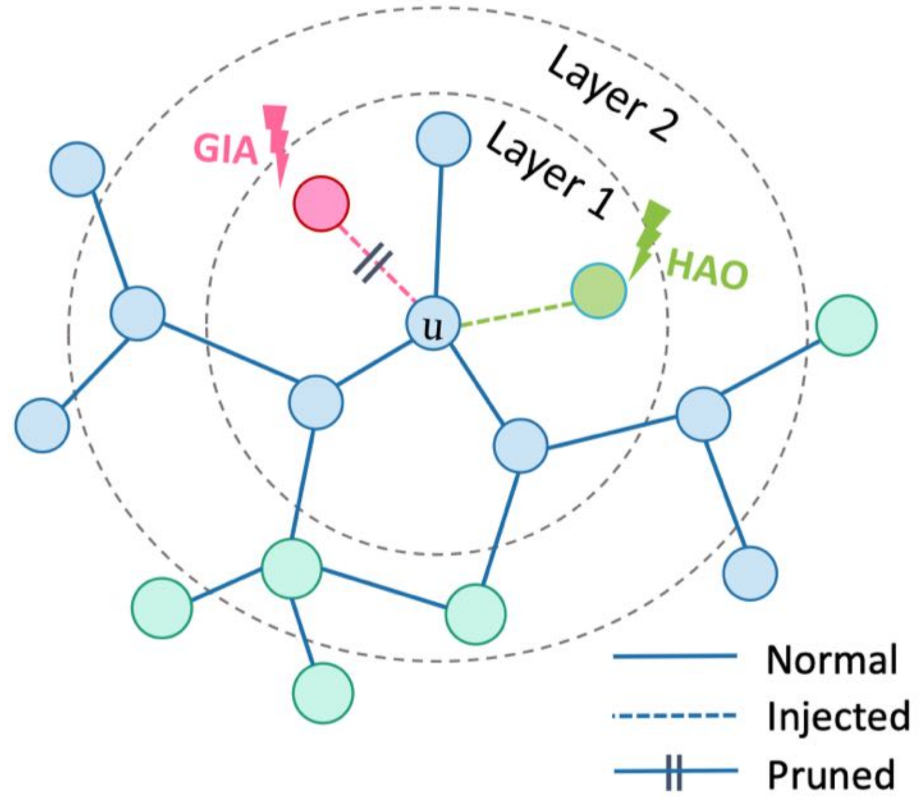


Illustration of GIA at node u

Definition 5 (Harmonious Adversarial Objective (HAO))

Observing the homophily (Definition. 4) is differentiable with respect to X, we can integrate it into the original adversarial objective as*:

$$\min_{\|G'-G\|\leq\triangle} \mathscr{L}^h_{\text{atk}}(f_{\theta^*}(G')) = \mathscr{L}_{\text{atk}}(f_{\theta^*}(G')) - \lambda C(G,G'),$$

where $C(G,G')$ is a regularization term based on homophily and $\lambda \geq 0$ is the corresponding weight.

HAO: Harmonious Adversarial Objective

HAO significantly improves the performance of all attacks on all datasets up to **30%.** Adaptive injection strategies can further advance the state of the art.

Homo: Homophily Defenders

Vanilla: Vanilla GNNs, e.g., GCN, GAT, GraphSage.

Robust: Robust GNN models, or GNN models with robust tricks such as layer normalisation, or adversarial training.

Combo: Robust GNN models with robust tricks such as layer normalisation, or adversarial training.

Table 1: Performance of non-targeted attacks against different models

		Cora (↓)			Citeseer(↓)			Computers(↓)			Arxiv(↓)		
	HAO	Homo	Robust	Combo	Homo	Robust	Combo	Homo	Robust	Combo	Homo	Robust	Combo
Clean		85.74	86.00	87.29	74.85	75.46	75.87	93.17	93.17	93.32	70.77	71.27	71.40
PGD		83.08	83.08	85.74	74.70	74.70	75.19	84.91	84.91	91.41	68.18	68.18	71.11
PGD	✓	52.60	62.60	77.99	69.05	69.05	73.04	79.33	79.33	87.83	55.38	62.89	68.68
MetaGIA†		83.61	83.61	85.86	74.70	74.70	75.15	84.91	84.91	91.41	68.47	68.47	71.09
MetaGIA†	✓	49.25	69.83	76.80	68.04	68.04	71.25	78.96	78.96	90.25	57.05	63.30	69.97
AGIA†		83.44	83.44	85.78	74.72	74.72	75.29	85.21	85.21	91.40	68.07	68.07	71.01
AGIA†	✓	47.24	**61.59**	**75.25**	70.24	70.24	71.80	75.14	75.14	86.02	59.32	65.62	69.92
TDGIA		83.44	83.44	85.72	74.76	74.76	75.26	88.32	88.32	91.40	64.49	64.49	70.97
TDGIA	✓	56.95	73.38	79.45	**60.91**	**60.91**	72.51	**74.77**	**74.77**	90.42	49.36	**60.72**	**63.57**
ATDGIA		83.07	83.07	85.39	74.72	74.72	75.12	86.03	86.03	91.41	66.95	66.95	71.02
ATDGIA	✓	**42.18**	70.30	76.87	61.08	61.08	**71.22**	80.86	80.86	**84.60**	**45.59**	63.30	64.31
MLP			61.75			65.55			84.14			52.49	

↓The lower number indicates better attack performance. †Runs with SeqGIA framework on Computers and Arxiv.

We evaluate with **38** defense models and report the *maximum* mean test robustness from multiple runs.

HAO: Harmonious Adversarial Objective

HAO significantly improves the performance of all attacks on all datasets up to **30%.** Adaptive injection strategies can further advance the state of the art.

Homo: Homophily Defenders

Vanilla: Vanilla GNNs, e.g., GCN, GAT, GraphSage.

Robust: Robust GNN models, or GNN models with robust tricks such as layer normalisation, or adversarial training.

Combo: Robust GNN models with robust tricks such as layer normalisation, or adversarial training.

Table 2: Performance of targeted attacks against different models

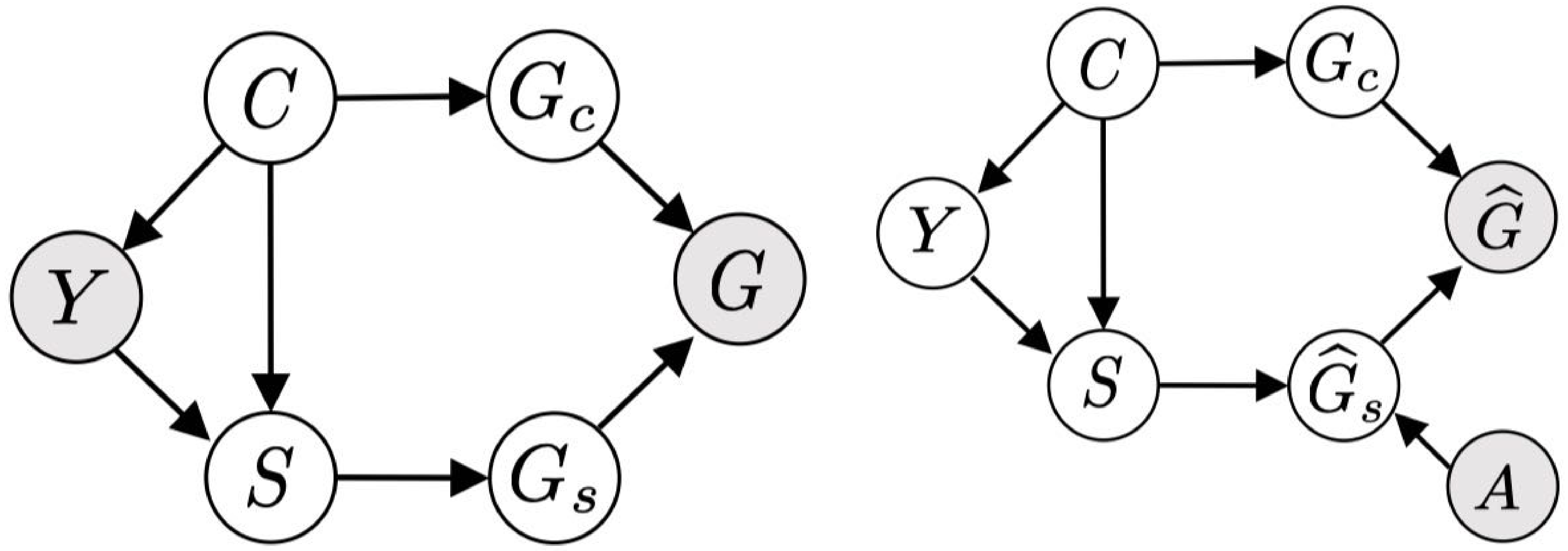
		Computers(↓)			Arxiv(↓)			Aminer(↓)			Reddit(↓)		
	HAO	Homo	Robust	Combo	Homo	Robust	Combo	Homo	Robust	Combo	Homo	Robust	Combo
Clean		92.68	92.68	92.83	69.41	71.59	72.09	62.78	66.71	66.97	94.05	97.15	97.13
PGD		88.13	88.13	91.56	69.19	69.19	71.31	53.16	53.16	56.31	92.44	92.44	93.03
PGD	✓	71.78	71.78	85.81	**36.06**	**37.22**	69.38	34.62	34.62	39.47	56.44	86.12	**84.94**
MetaGIA†		87.67	87.67	91.56	69.28	69.28	71.22	48.97	48.97	52.35	92.40	92.40	93.97
MetaGIA†	✓	70.21	71.61	85.83	38.44	38.44	48.06	41.12	41.12	45.16	**46.75**	90.06	90.78
AGIA†		87.57	87.57	91.58	66.19	66.19	70.06	50.50	50.50	53.69	91.62	91.62	93.66
AGIA†	✓	**69.96**	**71.58**	85.72	38.84	38.84	68.97	35.94	35.94	42.66	80.69	88.84	90.44
TDGIA		87.21	87.21	91.56	63.66	63.66	71.06	51.34	51.34	54.82	92.19	92.19	93.62
TDGIA	✓	71.39	71.62	**77.15**	42.56	42.56	42.53	25.78	25.78	29.94	78.16	**85.06**	88.66
ATDGIA		87.85	87.85	91.56	66.12	66.12	71.16	50.87	50.87	53.68	91.25	91.25	93.03
ATDGIA	✓	72.00	72.53	78.35	38.28	40.81	**39.47**	**22.50**	**22.50**	**28.91**	64.09	89.06	88.91
MLP			84.11			52.49			32.80			70.69	

↓The lower number indicates better attack performance. †Runs with SeqGIA framework.

We evaluate with **38** defense models and report the *maximum* mean test robustness from multiple runs.

Causality of HAO for Graph Adversarial Attacks

GIA without HAO essentially breaks the causal relations between C and Y:



(a) Graph generation. (b) Adversarial graph generation.

GIA with HAO that retains the homophily unnoticeability, reveals the **true underlying vulnerability of GNNs** and improves the robustness of GNNs:

Table 5.4: Performance of adversarial training methods under various graph adversarial attacks.

	HAO	Clean	PGD		TDGIA		MetaGIA		mean	worst
				✓		✓		✓		
GCN		84.95	38.55	38.55	40.67	43.78	38.43	38.80	46.25	38.43
GCN+FLAG		81.84	59.95	57.71	59.82	54.60	59.82	54.72	61.21	54.60
GCN+PGD		86.19	**72.76**	**72.13**	**80.34**	**75.49**	**70.77**	64.92	74.66	64.92
GCN+PGD	✓	**86.94**	**72.88**	**72.63**	**81.21**	**79.22**	**72.01**	**68.78**	76.24	**68.78**
GCN+TDGIA		85.69	66.29	65.29	75.74	71.76	64.92	58.83	69.79	58.83
GCN+TDGIA	✓	**86.56**	70.14	69.40	79.35	75.87	69.02	**65.42**	73.68	65.42
GNNGuard		85.07	84.20	84.70	84.45	53.73	84.82	43.15	74.30	43.15
GNNGuard+FLAG		84.57	84.32	84.32	84.32	69.77	84.45	64.92	79.52	64.92
GNNGuard+PGD		**86.44**	**86.69**	**85.69**	**86.56**	71.51	**86.19**	57.08	80.02	57.08
GNNGuard+PGD	✓	**86.44**	**86.31**	**86.06**	**86.19**	**77.86**	**86.31**	**69.77**	**82.71**	**69.77**
GNNGuard+TDGIA		85.94	85.94	85.57	85.82	71.14	85.69	56.46	79.51	56.46
GNNGuard+TDGIA	✓	85.57	85.69	85.57	85.32	**76.61**	85.57	**65.17**	**81.36**	65.17

Learning Causality for Modern Machine Learning

Traditional ML assumes train and test data are **iid.**, i.e., independently sampled from an identical distribution, while data is often **OOD**, i.e., out-of-distribution, in real-world applications.

Objectives

Causal Representation Learning on Graphs: [NeurIPS'22 Spotlight, NeurIPS'23a]

Implications

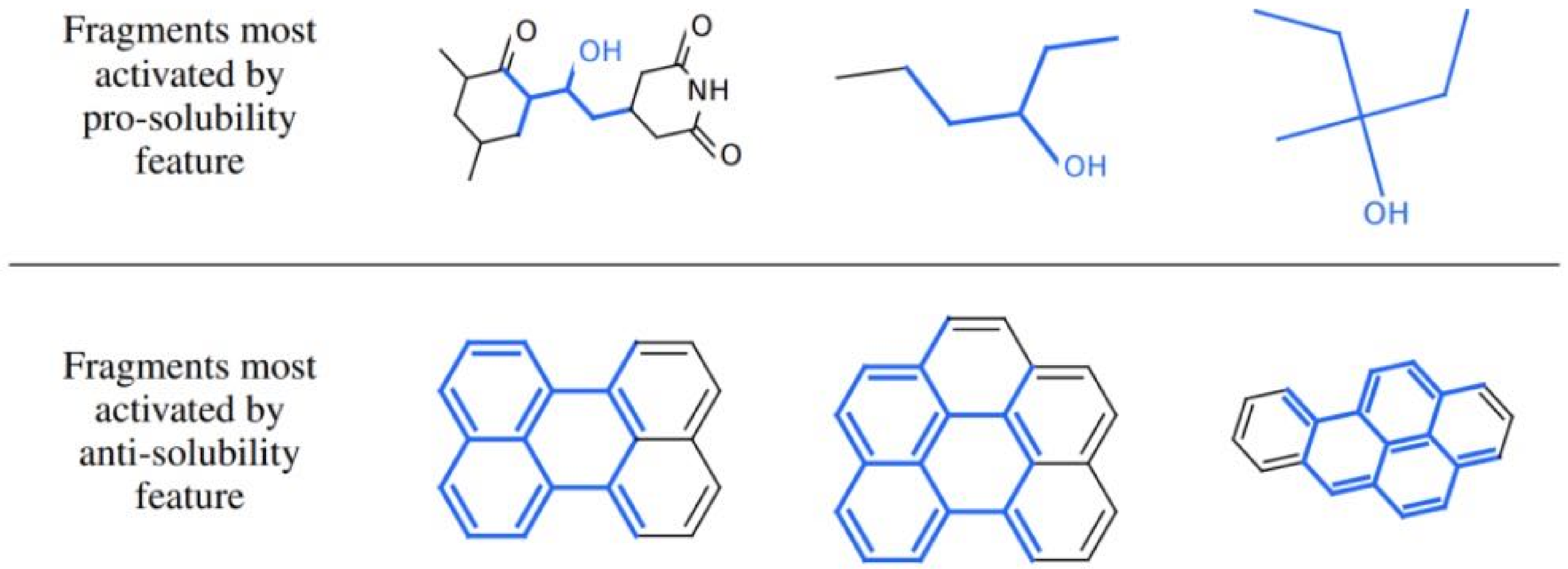
Useful Properties of the Causal Representations:
OOD Generalizability [NeurIPS'22, 23a],
Adversarial Robustness [ICLR'22],
Interpretability [ICML'24a]

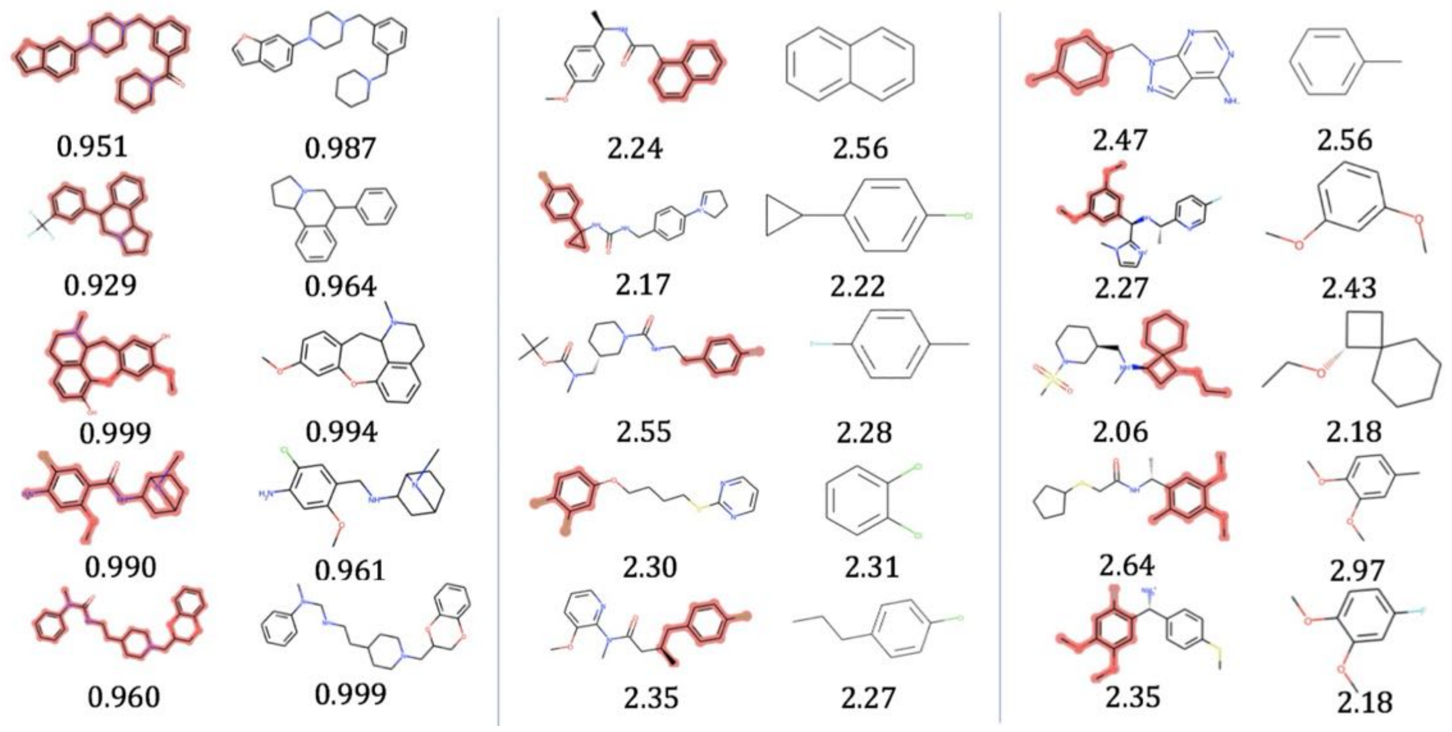
Realizations

Optimization & Feature Learning schemes for Causal Representation Learning: [ICLR'23a, NeurIPS'23b]

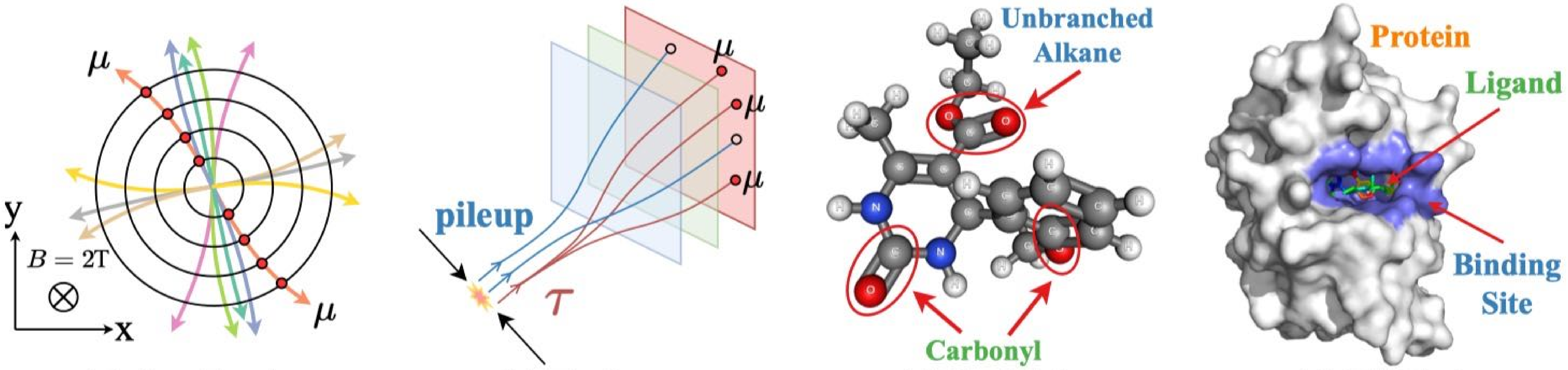
Interpretable Graph Neural Networks

Interpretability is crucial for a variety of scientific tasks:





Scientific Tasks in 2D Regular Graphs



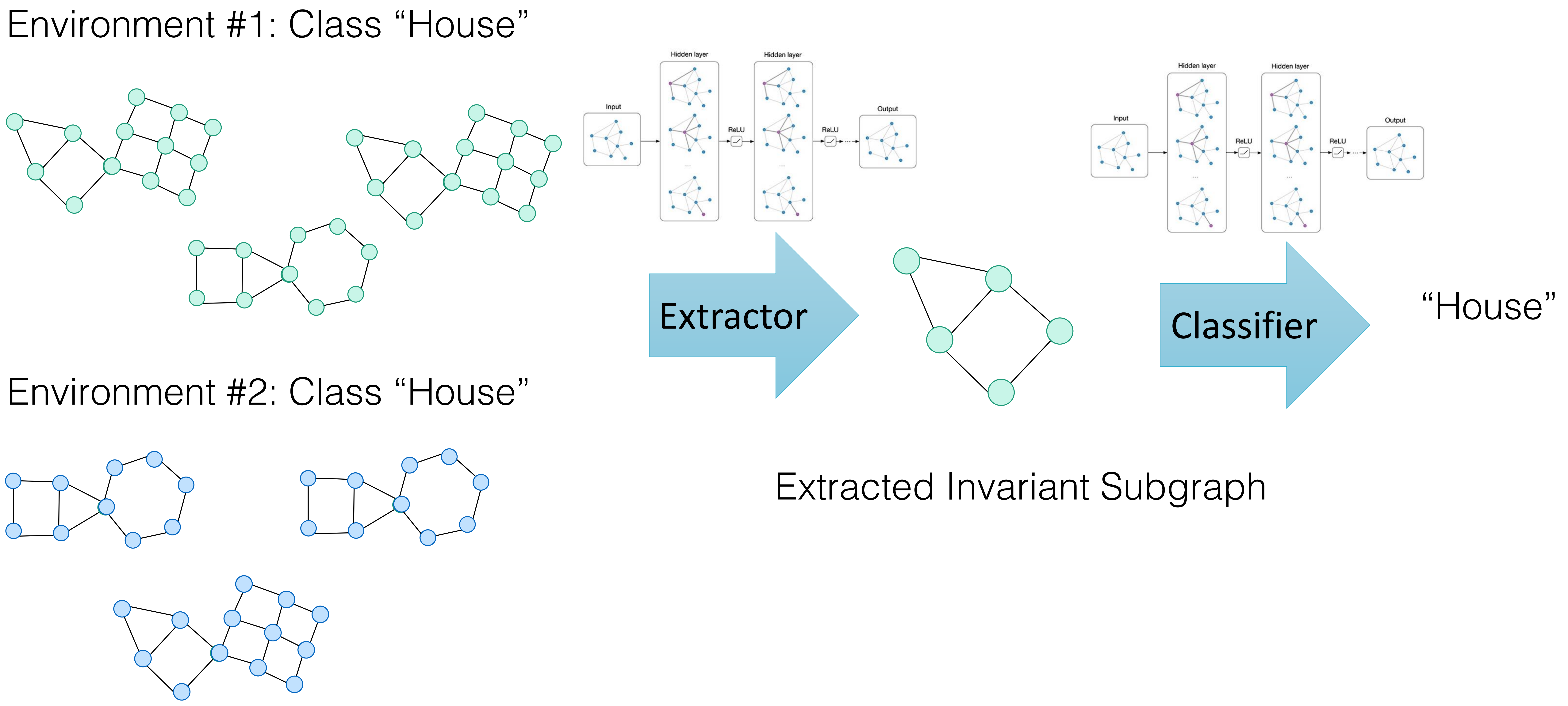
(a) ActsTrack (b) Tau3mu (c) SynMol (d) PLBind

Scientific Tasks in 3D Geometric Graphs

(Duvenaud et al., 2015; Yu et al., 2021; Miao et al., 2022; Miao et al., 2023)

Interpretable Graph Neural Networks

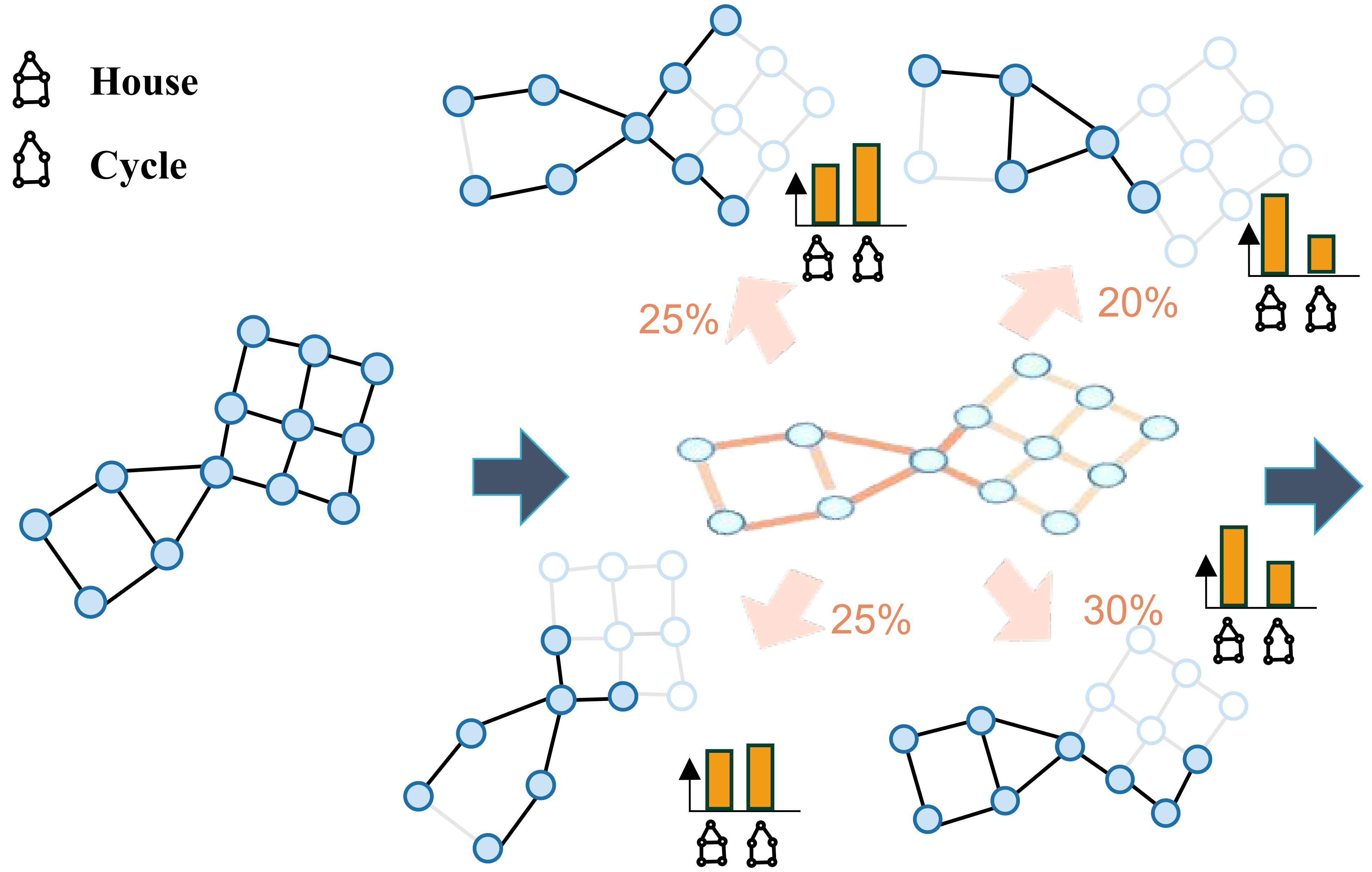
Interpretability and **generalizability** are *two sides of the same coin*, when considering distribution shifts that are everywhere:



(Wu et al., 2022ab; Miao et al., 2022; Chen et al., 2022)

Expressivity Issue of Interpretable GNNs

Interpretable GNNs computes sampling probability using the attention mechanism:



$G \sim \mathcal{D}_{tr}$

$\widehat{G}_c = \mathbb{E}_{G_c \overset{g}{\sim} G}[G_c]$

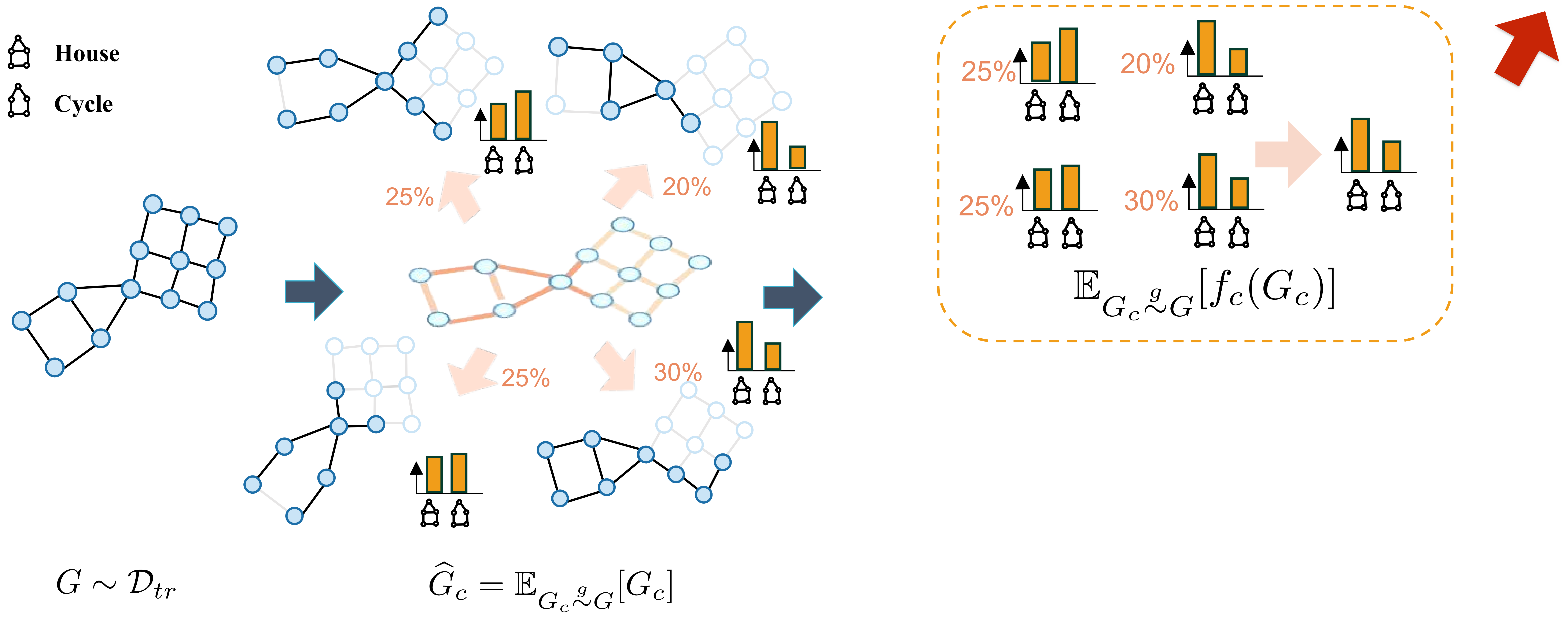
Step1: Soft Subgraph Extraction

(Wu et al., 2022ab; Miao et al., 2022; Chen et al., 2022, 2023)

Expressivity Issue of Interpretable GNNs

The sampling probability accumulates a subgraph distribution, where each subgraph corresponds to a label distribution:

Subgraph Multilinear Extension



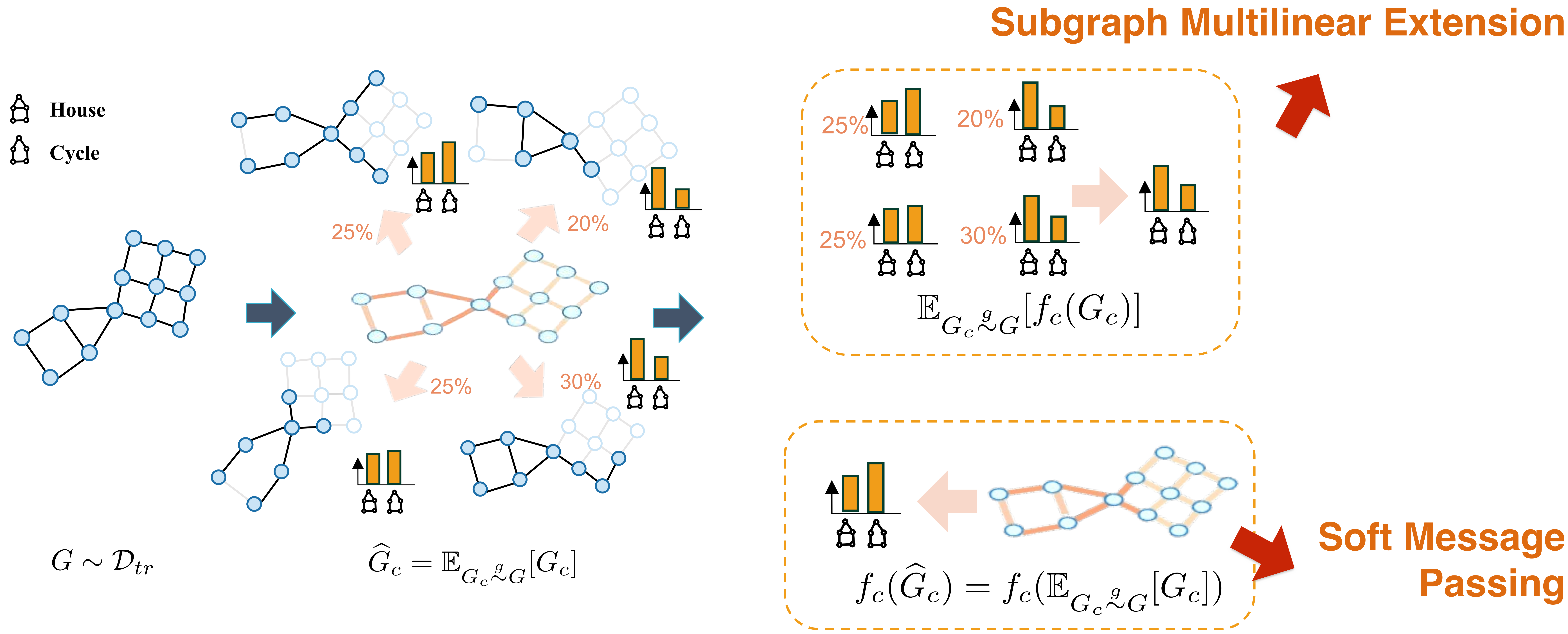
Step1: Subgraph Extraction

Step2: Subgraph Classification

(Wu et al., 2022ab; Miao et al., 2022; Chen et al., 2022, 2023)

Expressivity Issue of Interpretable GNNs

Existing Interpretable GNNs directly take the expected soft subgraph to predict the label:



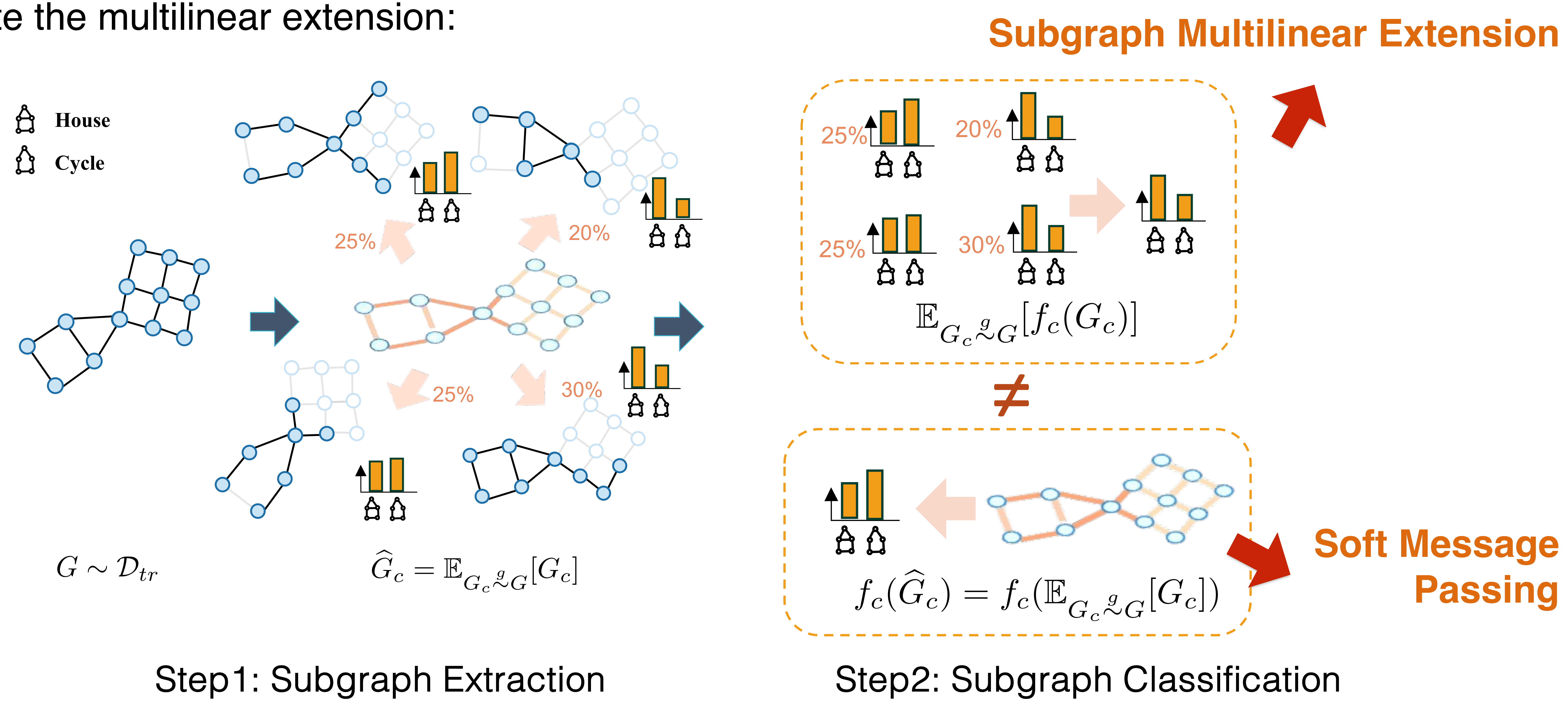
Step1: Subgraph Extraction

Step2: Subgraph Classification

(Wu et al., 2022ab; Miao et al., 2022; Chen et al., 2022, 2023)

Expressivity Issue of Interpretable GNNs

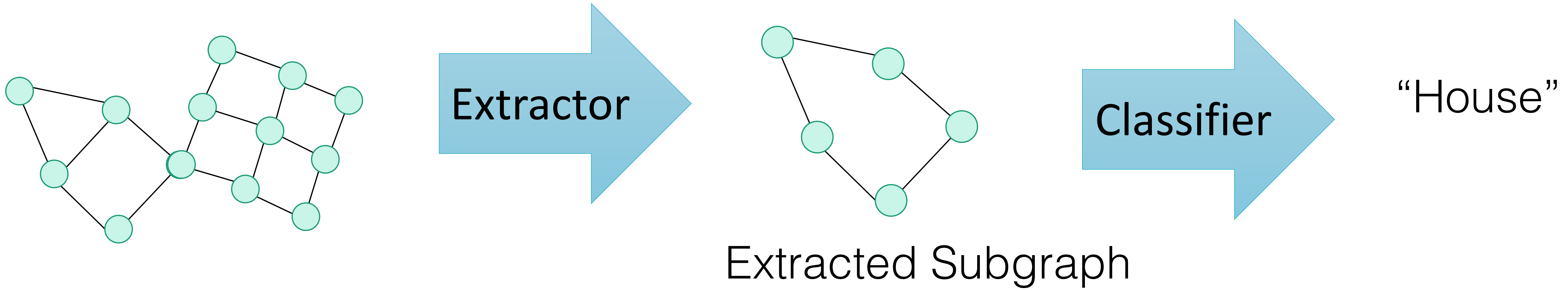
Given any non-linear GNNs, or linear GNNs with more than two layers, soft message passing can not approximate the multilinear extension:

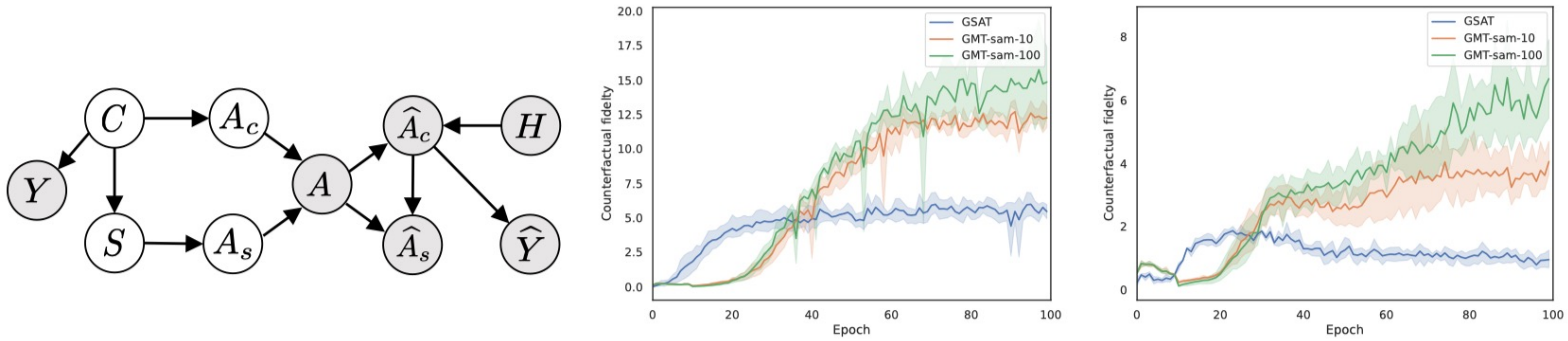


(Wu et al., 2022ab; Miao et al., 2022; Chen et al., 2022, 2023)

Expressivity Issue of Interpretable GNNs

Failing to approximate SubMT results in unfaithful interpretations:



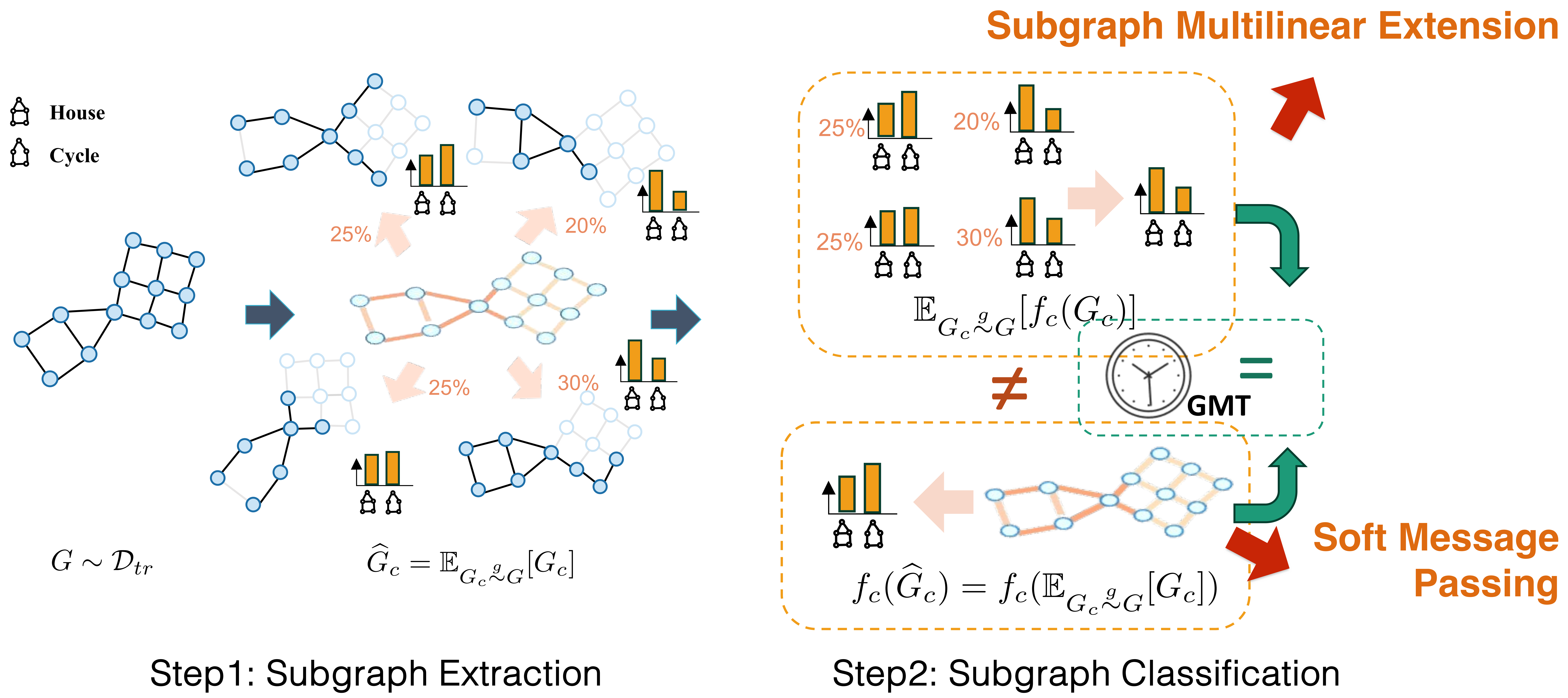


(a) SCM of XGNNs. (b) SubMT on BA-2Motifs. (c) SubMT on Mutag.

SubMT approximation failure shown with counterfactual fidelty

GMT: Graph Multilinear Network

We propose GMT to bridges the gap by approximating and distilling the SubMT into soft message passing:



GMT: Graph Multilinear Network

GMT brings up to **10% AUROC** improvements in interpretability and up to **10% Acc** improvements in generalizability on regular graphs.

Table 1. Interpretation Performance (AUC) on regular graphs.Results with the mean-1*std larger than the best baselines are shadowed.

GNN	Method	BA-2Motifs	Mutag	MNIST-75sp	Spurious-Motif $b=0.5$	Spurious-Motif $b=0.7$	Spurious-Motif $b=0.9$
GIN	GNNExplainer	67.35±3.29	61.98±5.45	59.01±2.04	62.62±1.35	62.25±3.61	58.86±1.93
	PGExplainer	84.59±9.09	60.91±17.10	69.34±4.32	69.54±5.64	72.33±9.18	72.34±2.91
	GraphMask	92.54±8.07	62.23±9.01	73.10±6.41	72.06±5.58	73.06±4.91	66.68±6.96
	IB-Subgraph	86.06±28.37	91.04±6.59	51.20±5.12	57.29±14.35	62.89±15.59	47.29±13.39
	DIR	82.78±10.97	64.44±28.81	32.35±9.39	78.15±1.32	77.68±1.22	49.08±3.66
GIN	GSAT	98.85±0.47	99.35±0.95	80.47±1.86	74.49±4.46	72.95±6.40	65.25±4.42
	GMT-lin	98.36±0.56	99.86±0.09	82.98±1.49	76.06±6.39	76.50±5.63	**80.57**±2.59
	GMT-sam	**99.62**±0.11	**99.87**±0.11	**86.50**±1.80	**85.50**±2.40	**84.67**±2.38	73.49±5.33
PNA	GSAT	89.35±5.41	99.00±0.37	85.72±1.10	79.84±3.21	79.76±3.66	80.70±5.45
	GMT-lin	95.79±7.30	99.58±0.17	85.02±1.03	80.19±2.22	84.74±1.82	85.08±3.85
	GMT-sam	**99.60**±0.48	**99.89**±0.05	**87.34**±1.79	**88.27**±1.71	**86.58**±1.89	**85.26**±1.92

Table 2. Prediction Performance (Acc.) on regular graphs. The shadowed entries are the results with the mean-1*std larger than the mean of the corresponding best baselines.

GNN	Method	MolHiv (AUC)	Graph-SST2	MNIST-75sp	Spurious-Motif $b=0.5$	Spurious-Motif $b=0.7$	Spurious-Motif $b=0.9$
GIN	GIN	76.69±1.25	82.73±0.77	95.74±0.36	39.87±1.30	39.04±1.62	38.57±2.31
	IB-subgraph	76.43±2.65	82.99±0.67	93.10±1.32	54.36±7.09	48.51±5.76	46.19±5.63
	DIR	76.34±1.01	82.32±0.85	88.51±2.57	45.49±3.81	41.13±2.62	37.61±2.02
GIN	GSAT	76.12±0.91	83.14±0.96	96.20±1.48	47.45±5.87	43.57±2.43	45.39±5.02
	GMT-lin	76.87±1.12	83.19±1.28	96.01±0.25	47.69±4.93	53.11±4.12	46.22±4.18
	GMT-sam	**77.22**±0.93	**83.62**±0.50	**96.50**±0.19	**60.09**±2.40	**54.34**±4.04	**55.83**±5.68
PNA	PNA	78.91±1.04	79.87±1.02	87.20±5.61	68.15±2.39	66.35±3.34	61.40±3.56
	GSAT	79.82±0.67	80.90±0.37	93.69±0.73	68.41±1.76	67.78±3.22	51.51±2.98
	GMT-lin	80.05±0.71	81.18±0.47	94.44±0.49	69.33±1.42	64.49±3.51	58.30±6.61
	GMT-sam	**80.58**±0.83	**82.36**±0.96	**95.75**±0.42	**71.98**±3.44	**69.68**±3.99	**67.90**±3.60

GMT: Graph Multilinear Network

GMT brings up to **7% AUROC** and **18% Precision@12** improvements in interpretability and up to **4% Acc** improvements in generalizability on geometric graphs.

Table 3. Interpretation performance on geometric graphs. Results with the mean-1*std larger than the best baselines are shadowed.

	ActsTrack		Tau3Mu		SynMol		PLBind	
	ROC AUC	Prec@12	ROC AUC	Prec@12	ROC AUC	Prec@12	ROC AUC	Prec@12
Random	50	21	50	35	50	31	50	45
GradGeo	69.31±0.89	33.54±1.23	78.04±0.57	64.18±1.25	76.38±4.96	64.72±3.75	58.11±2.91	64.78±4.73
BernMask	54.23±4.31	20.46±5.46	71.58±0.69	60.51±0.76	76.38±4.96	64.72±3.75	52.23±4.45	41.50±9.77
BernMask-P	22.87±3.33	11.29±5.46	70.72±5.10	55.50±6.26	87.06±7.12	77.11±7.58	51.98±4.66	59.20±5.48
PointMask	49.20±1.51	20.54±1.71	55.93±4.85	39.65±7.14	66.46±6.86	53.93±1.94	50.00±0.00	45.10±0.00
GradGAM	75.19±1.91	75.94±2.16	76.18±2.62	62.05±2.16	60.31±4.95	52.35±11.02	48.61±2.34	55.10±10.57
LRI-Bernoulli	74.38±4.33	81.42±1.52	78.23±1.11	65.64±2.44	89.22±3.58	68.76±7.35	54.87±1.89	72.12±2.60
GMT-lin	**77.45**±1.69	**81.81**±1.57	**79.17**±0.82	**68.94**±1.08	**96.17**±1.44	**86.33**±6.16	59.70±1.10	70.62±3.59
GMT-sam	75.61±1.86	81.96±1.35	78.28±1.34	65.69±2.61	93.93±3.59	83.20±4.74	**60.03**±1.02	**72.56**±2.27

Table 4. Prediction performance (AUC) on geometric graphs.

	ActsTrack	Tau3Mu	SynMol	PLBind
ERM	97.40±0.32	82.75±0.16	99.30±0.20	85.31±2.21
LRI-Bernoulli	94.00±0.78	86.36±0.06	99.30±0.15	85.80±0.70
GMT-lin	93.92±0.98	82.60±0.17	99.26±0.27	86.29±0.80
GMT-sam	**98.55**±0.11	**86.42**±0.08	**99.89**±0.03	**87.19**±1.86

Learning Causality for Modern Machine Learning

Traditional ML assumes train and test data are **iid.**, i.e., independently sampled from an identical distribution, while data is often **OOD**, i.e., out-of-distribution, in real-world applications.

Objectives

Causal Representation Learning on Graphs: [NeurIPS'22 Spotlight, NeurIPS'23a]

Implications

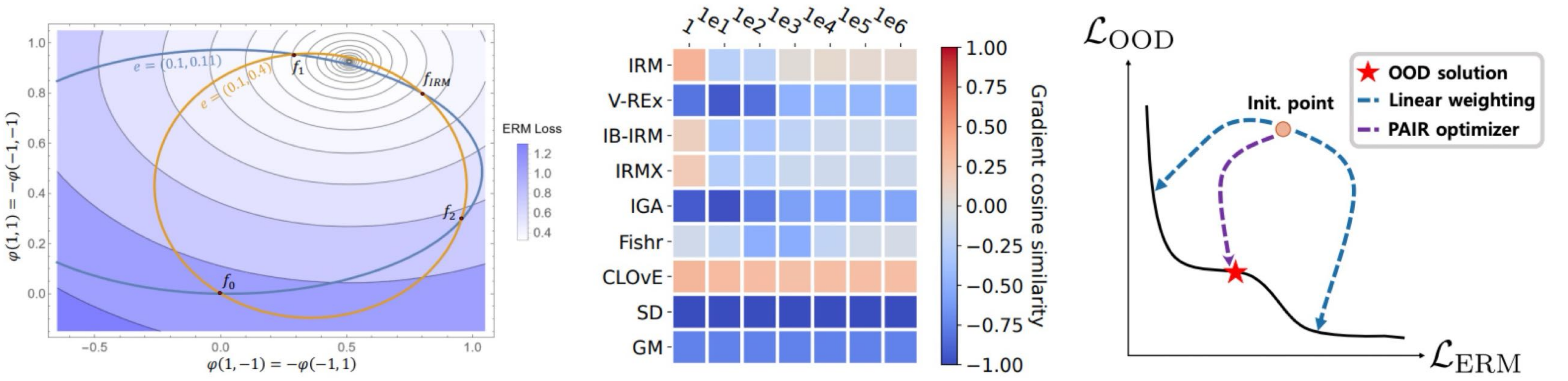
Useful Properties of the Causal Representations:
OOD Generalizability [NeurIPS'22, 23a],
Adversarial Robustness [ICLR'22],
Interpretability [ICML'24a]

Realizations

Optimization & Feature Learning schemes for Causal Representation Learning: [ICLR'23a, NeurIPS'23b]

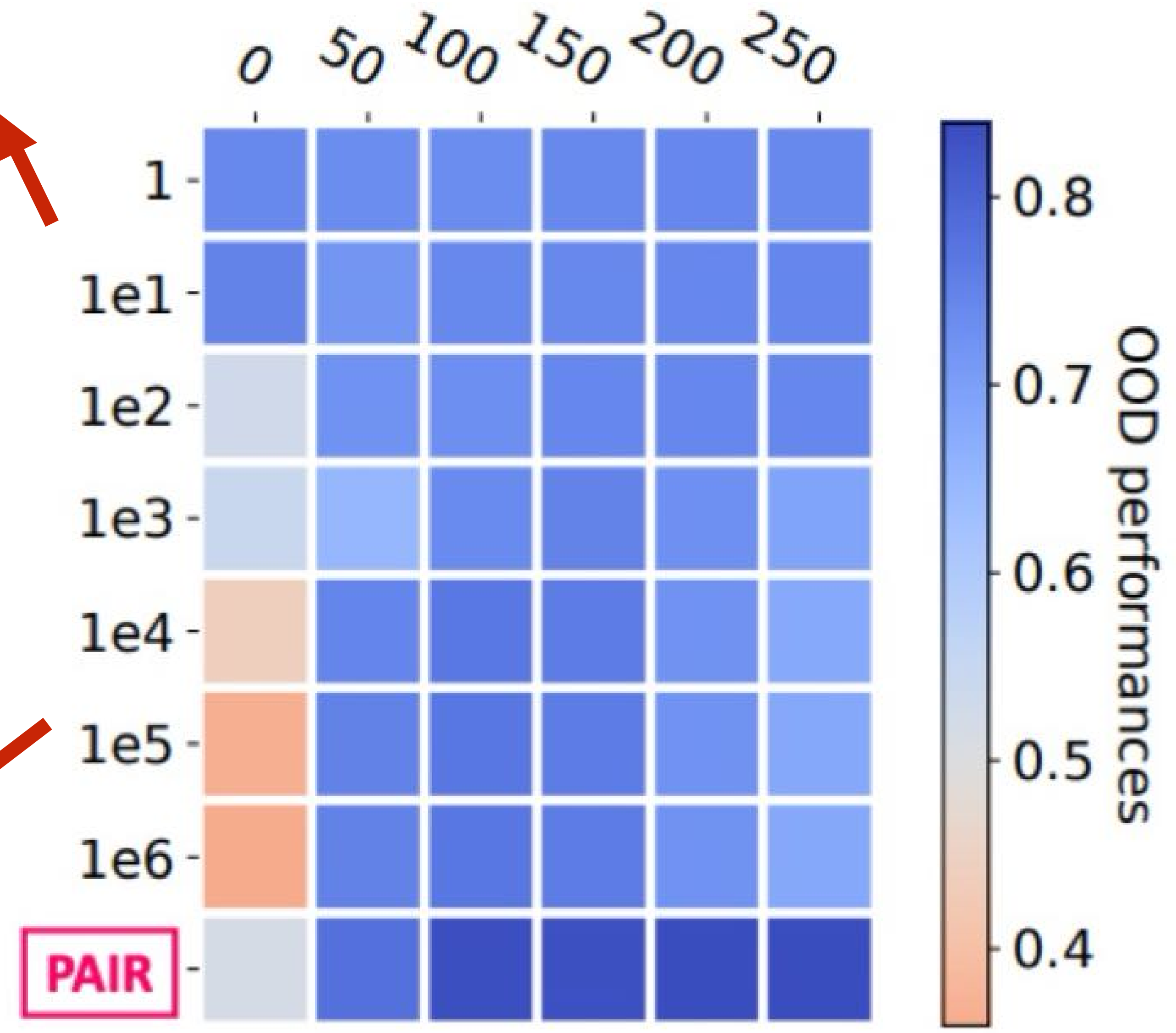
The Optimization Dilemma in OOD Generalization

Traditional optimization strategy is **not suitable** for OOD generalization.



(a) Theoretical failure case. (b) Gradient conflicts. (c) Unreliable opt. scheme.

Too weak!



Too strong!

(d) Exhaustive tuning.

The usual optimization formula of OOD objectives in practice:

$$\min_f L_{\text{ERM}} + \boxed{\lambda} \cdot \boxed{\widehat{L}_{\text{OOD}}}$$

λ is ***hard to tune*** Regularization via some ***relaxed*** OOD objective

The Optimization Dilemma in OOD Generalization

We demonstrate the issue using a widely studied and adopted frameworks: Invariant Risk Minimization.

$$\min_f L_{\mathrm{ERM}} + \lambda \cdot \boxed{\hat{L}_{\mathrm{OOD}}}$$

Regularization via some ***relaxed*** OOD objective

$$\min_{f=w\circ\varphi} \sum_{e\in\mathcal{E}_{\mathrm{tr}}} \mathcal{L}_e(w\circ\varphi),$$
$$\text{s.t. } w \in \arg\min_{\bar{w}} \mathcal{L}_e(\bar{w}\circ\varphi),\ \forall e \in \mathcal{E}_{\mathrm{tr}}$$

Linearized IRM with $w \in \mathbb{R}^d$

$$\min_{\varphi} \sum_{e\in\mathcal{E}_{\mathrm{tr}}} \mathcal{L}_e(\varphi),$$
$$\text{s.t. } \nabla_{w|w=1}\mathcal{L}_e(w\cdot\varphi) = 0,\ \forall e \in \mathcal{E}_{\mathrm{tr}}$$

Soften the constraints

$$\min_{\varphi} \sum_{e\in\mathcal{E}_{\mathrm{tr}}} \mathcal{L}_e(\varphi) + \lambda \|\nabla_{w|w=1}\mathcal{L}_e(w\cdot\varphi)\|^2$$

IRM 😋

$\mathrm{IRM}_{\mathcal{S}}$ 🥲

IRMv1 🥲 🥲

(Arjovsky et al., 2019; Kamath et al., 2021)

The Optimization Dilemma in OOD Generalization

The practical variants of IRM can have very different behaviors from the original IRM.

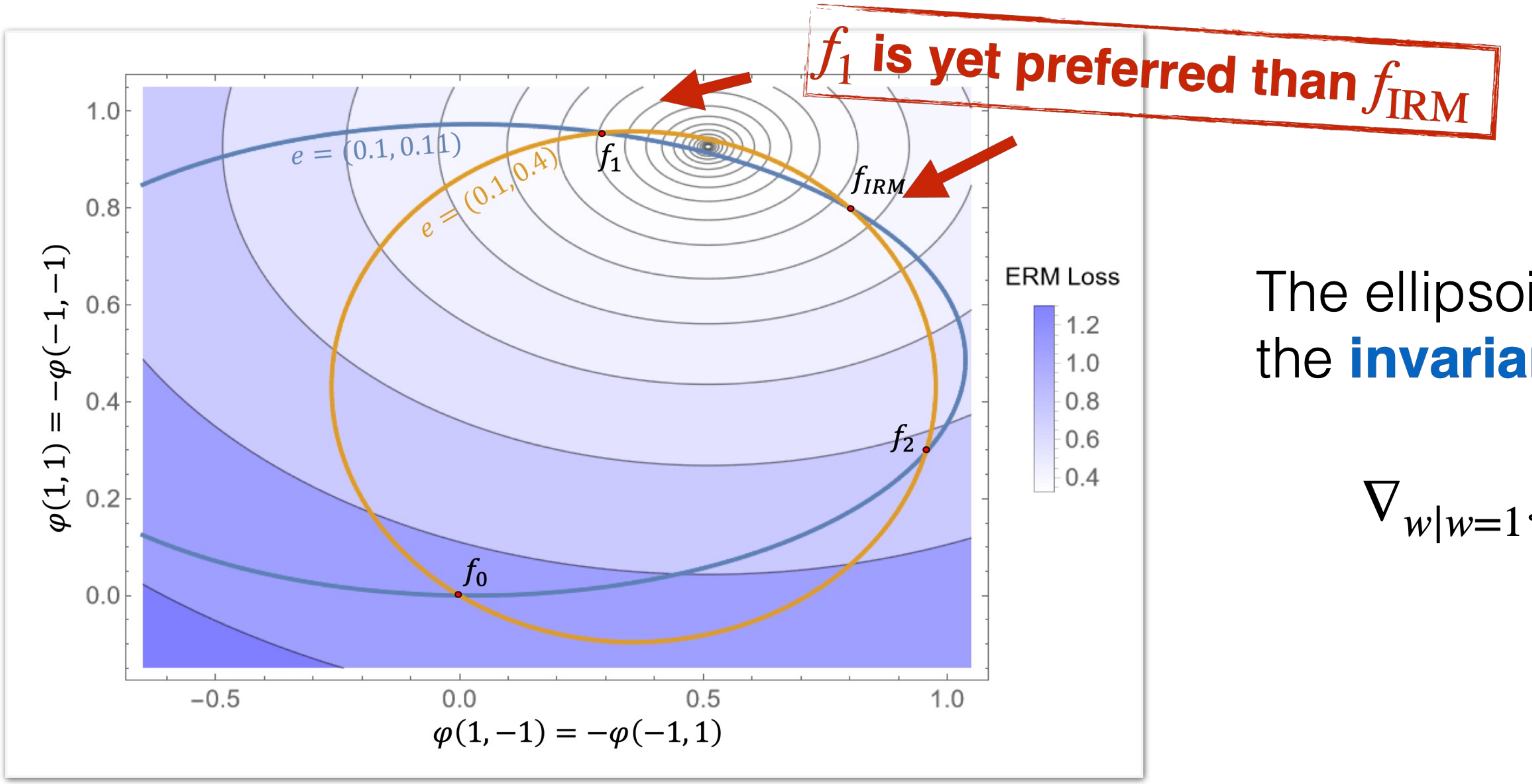


Illustration of IRMv1 failures

The ellipsoids are the solutions satisfying the **invariant constraints** in $\text{IRM}_{\mathcal{S}}$

$$\nabla_{w|w=1}\mathscr{L}_e(w \cdot \varphi) = 0, \ \forall e \in \mathscr{E}_{\text{tr}}$$

(Arjovsky et al., 2019; Kamath et al., 2021)

PAIR: Pareto Invariant Risk Minimization

We propose PAIR, that tackles the optimization from a multi-objective optimization perspective:

The optimization of IRM essentially handles the ***trade-off*** between

$$\min_f \boxed{L_{\text{ERM}}} + \lambda \cdot \boxed{\widehat{L}_{\text{OOD}}}$$

Capturing the statistical correlations   Enforcing the invariance of learned correlations

Oh, it's a Multi-Objective Optimization (MOO)!

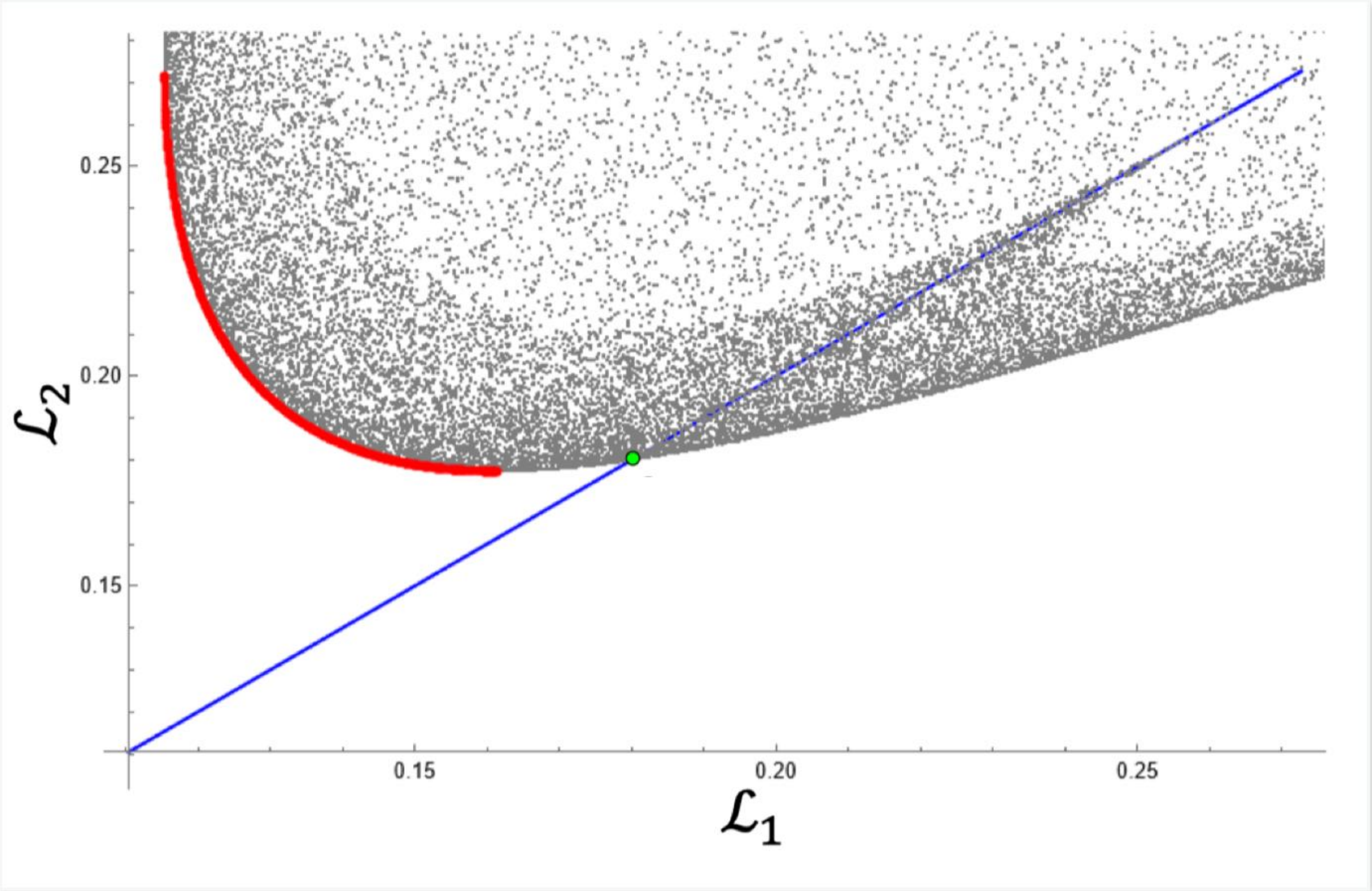
$$\min_f \{L_{\text{ERM}}, \ \widehat{L}_{\text{OOD}}\}^T$$

PAIR: Pareto Invariant Risk Minimization

We propose PAIR, that tackles the optimization from a multi-objective optimization perspective:

Assume we have the Multi-Objective Optimization (MOO) problem with 2 objectives:

$$\min_{f=w\cdot\varphi} \{L_1, L_2\}^T$$



Simulated Pareto front

- A solution f (with $\{L_1, L_2\}^T$) **dominates** $\bar{f}$ (with $\{\bar{L}_1, \bar{L}_2\}^T$) if both $L_1 \leq \bar{L}_1$ and $L_2 \leq \bar{L}_2$;
- **Pareto optimal solutions** are the set of solutions dominated by none;
- Their images form the **Pareto front**;

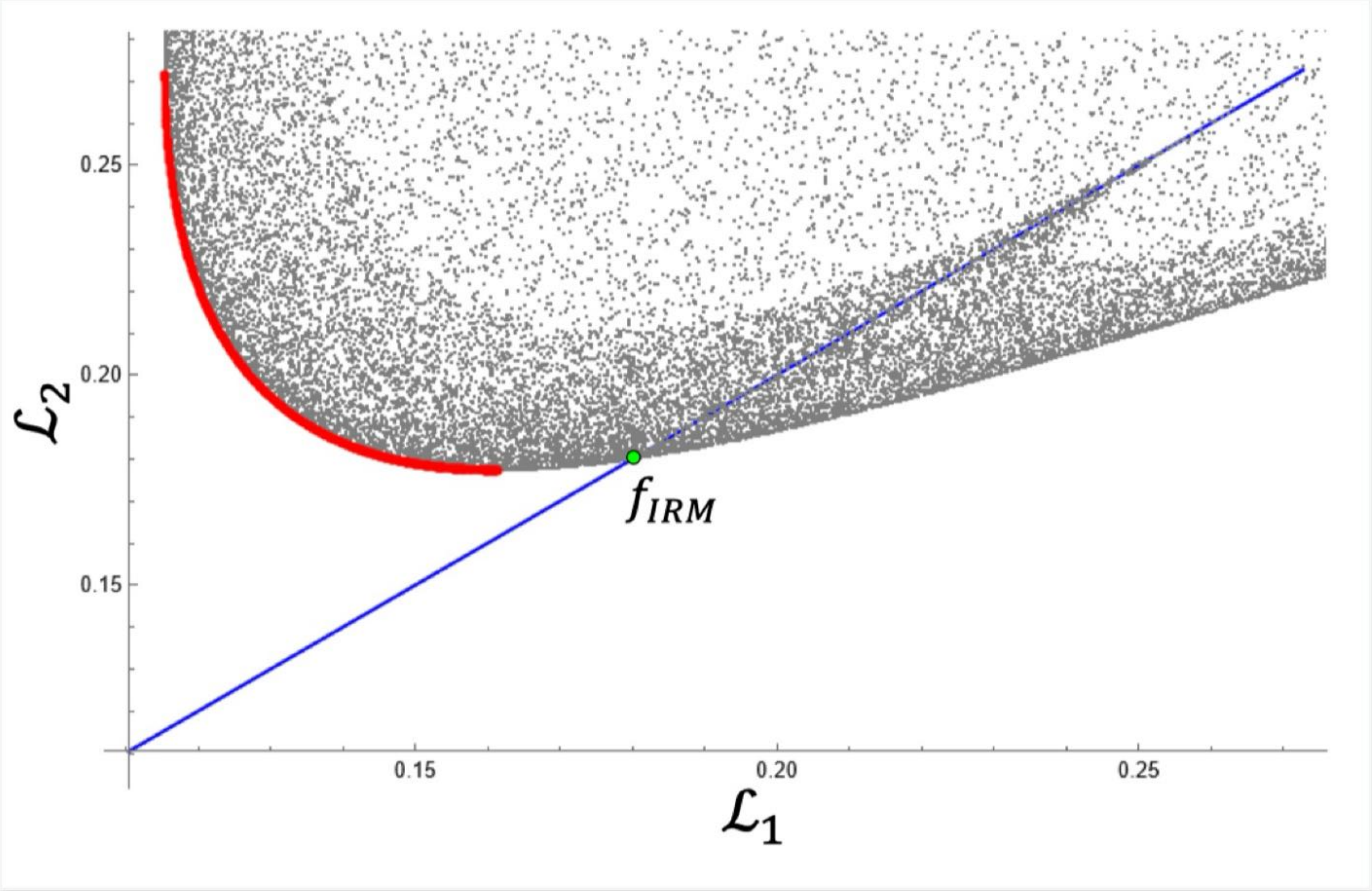
(Kaisa, 1999)

PAIR: Pareto Invariant Risk Minimization

We propose PAIR, that tackles the optimization from a multi-objective optimization perspective:

Assume we have 2 training environments, a natural MOO formulation of IRMv1 is:

$$\min_{f=w\cdot\varphi} \{L_1, L_2, L_{\mathrm{IRM}}\}^T$$



Simulated Pareto front

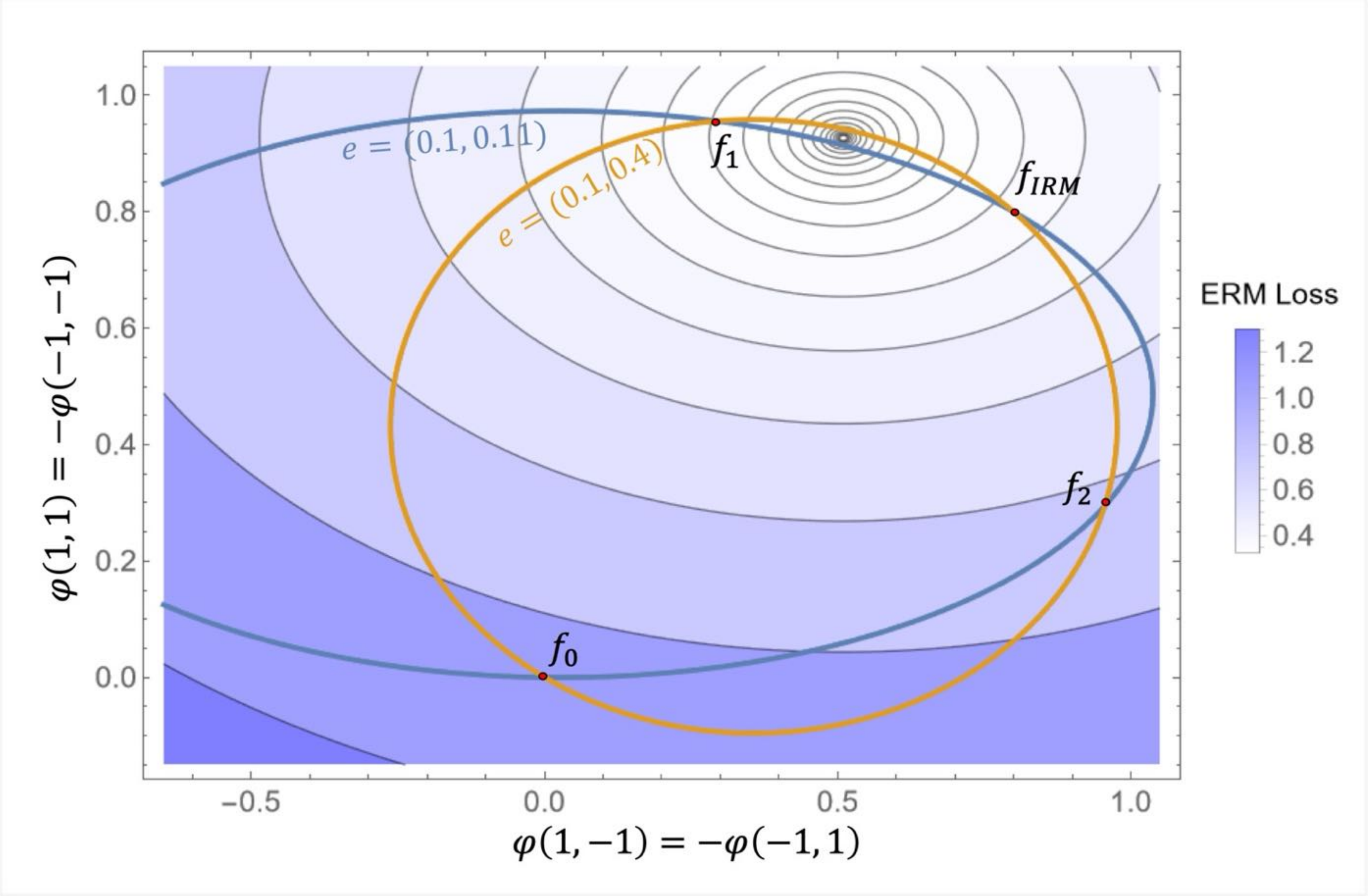
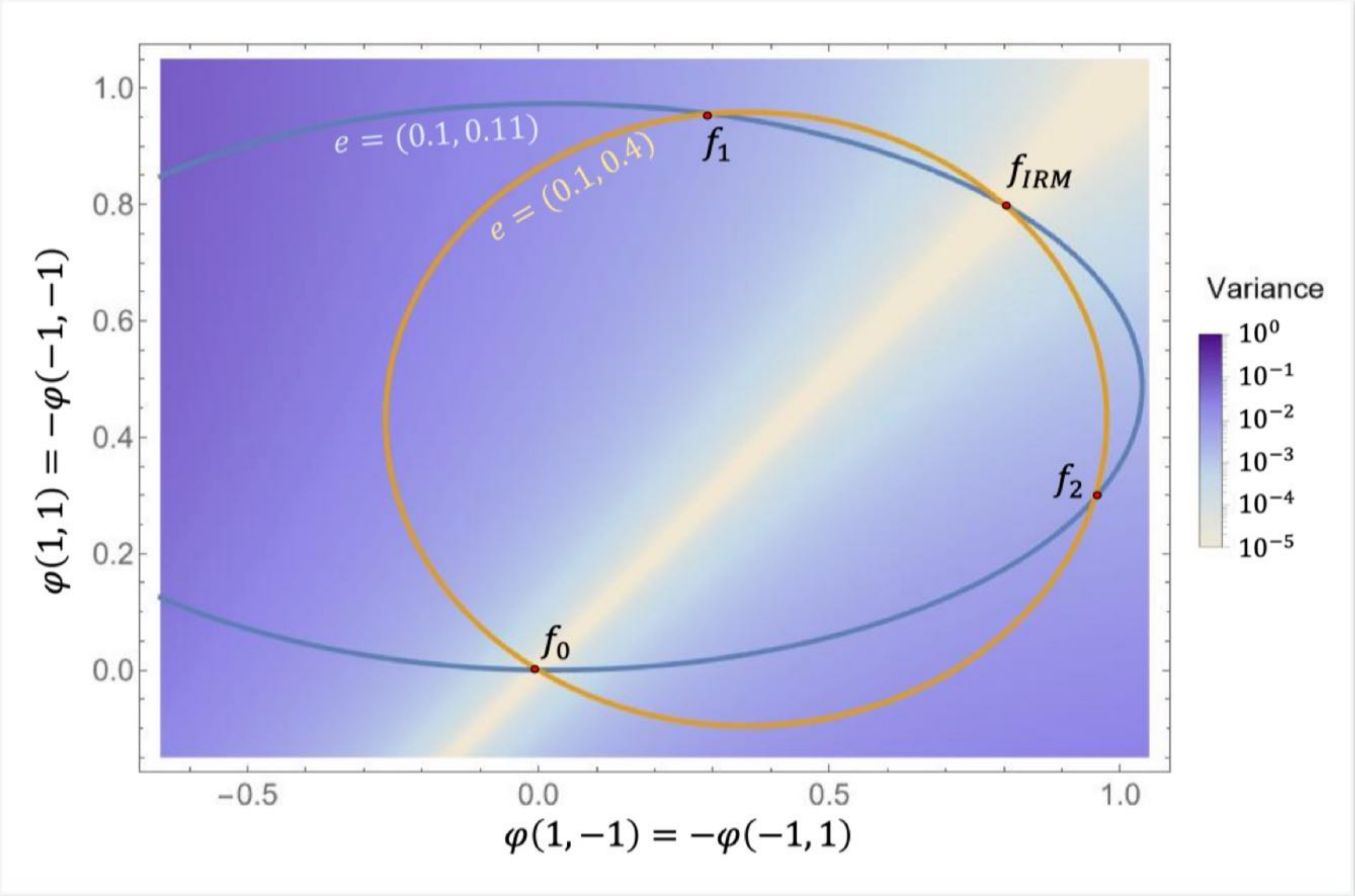


Illustration of IRMv1 failures

PAIR: Pareto Invariant Risk Minimization

We propose PAIR, that tackles the optimization from a multi-objective optimization perspective:

A PAIRed journey into the adventure of extrapolation: $\min_{f=w\cdot\varphi} \{L_{\text{ERM}}, L_{\text{IRM}}, L_{\text{REx}}\}^T$



Theoretical results (Informal):

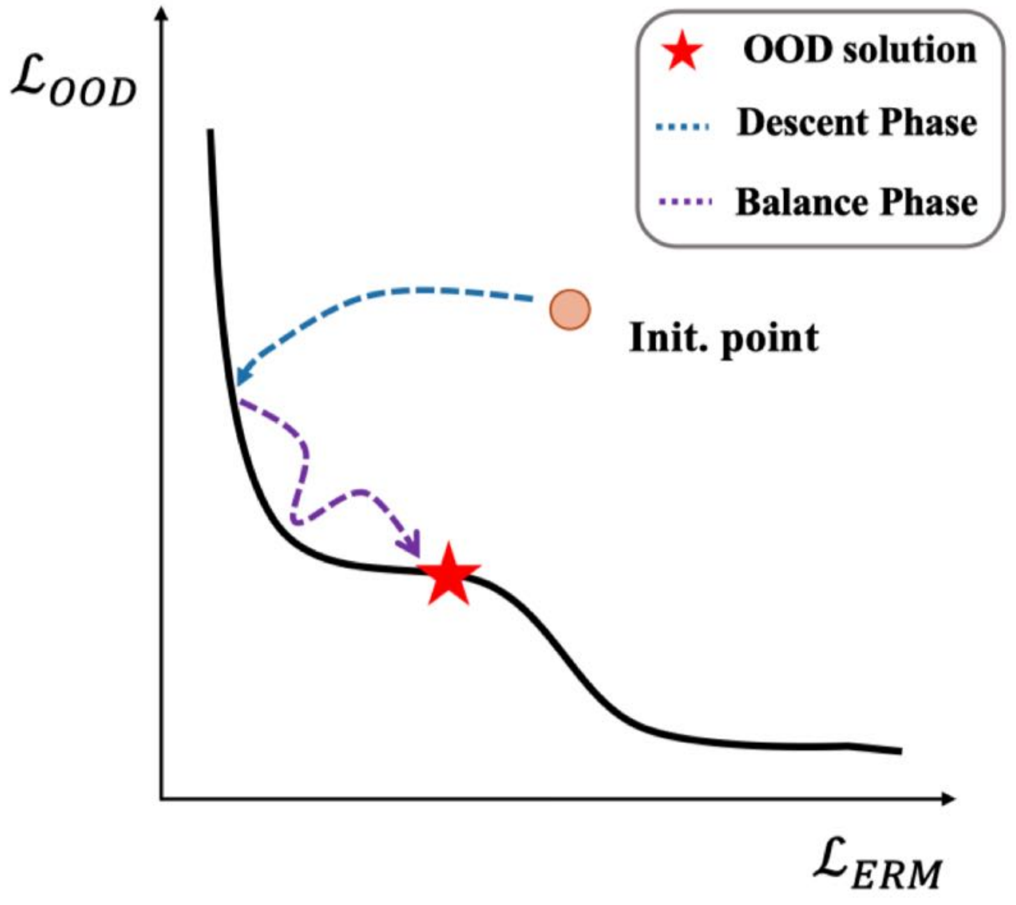
IRMX solves the IRMv1 failures under any environment settings in *(Kamath et al., 2021)*.

PAIR: Pareto Invariant Risk Minimization

We propose PAIR, that tackles the optimization from a multi-objective optimization perspective:

IRMX raises more challenges in the optimization:

$$\min_{f=w\cdot\varphi} \{L_{\mathrm{ERM}}, L_{\mathrm{IRM}}, L_{\mathrm{REx}}\}^T$$



Exact Pareto optimal search

- The Pareto front becomes **more complicated**:
  - ✓ The optimizer needs to be able to reach **any** Pareto optimal solutions!
- There can be **multiple** Pareto optimal solutions:
  - ✓ A **preference** of each objective is required! ***PAIR-o*** as the OOD optimizer;

Theoretical results (Informal):
Under mild assumptions, let f_{OOD} be the desired OOD solution w.r.t. an underlying preference $\mathbf{p}_{\mathrm{OOD}}$, PAIR-o converges and approximates to f_{OOD} for any approximated $\widehat{\mathbf{p}}_{\mathrm{OOD}}$.

(Mahapatra & Rajan 2020)

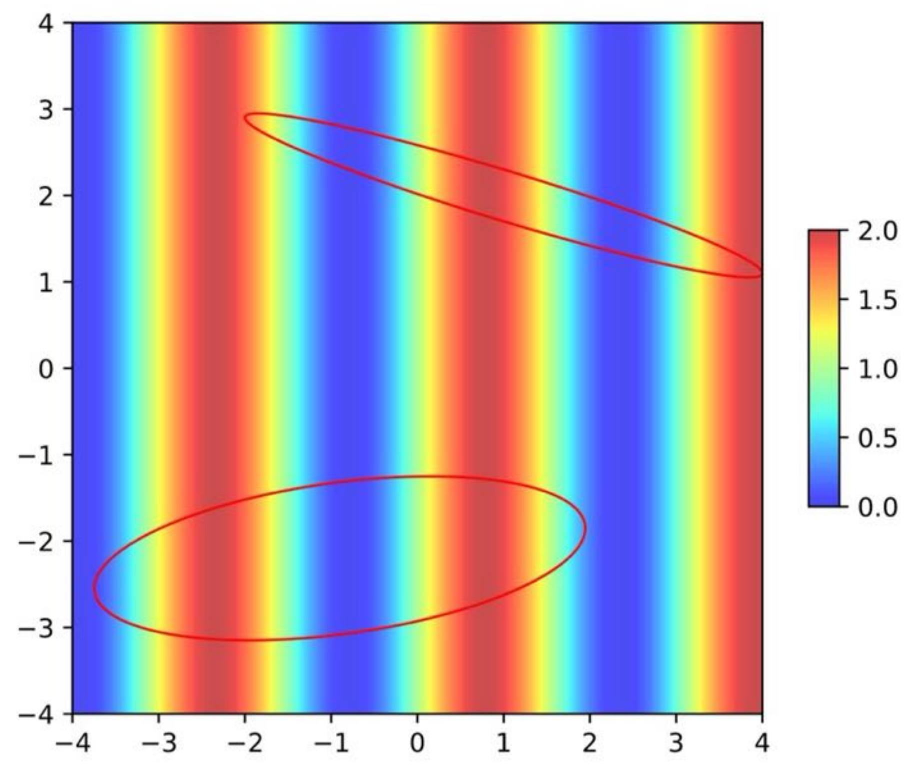
Causal Invariance Recovery Tests

We first test PAIR in a simple regression setting:

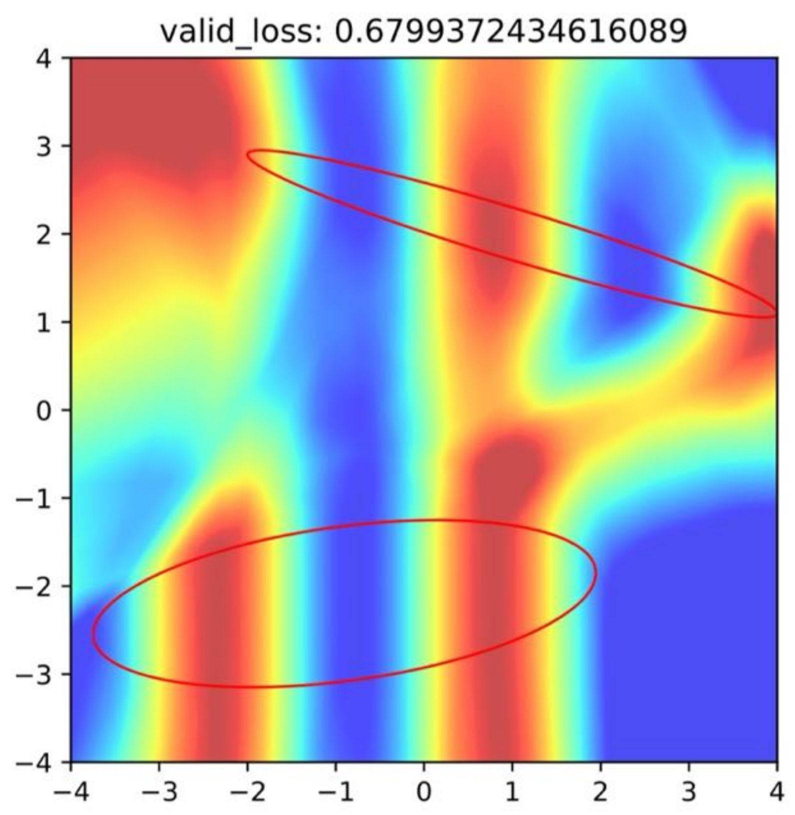
Regression target:
$Y = \sin(X_1) + 1$, only depends on the x-axis;

Training envs:
Two elliptical regions (Gaussian distributions) marked in red;

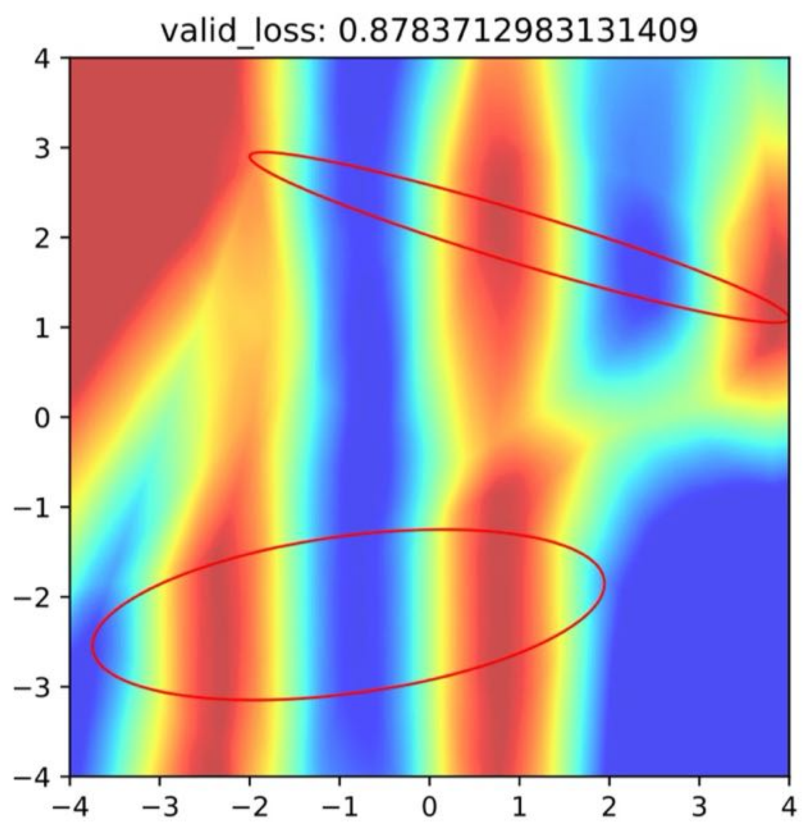
Invariance:
The **overlapped** x-axis region, i.e., $[-2,2]$.



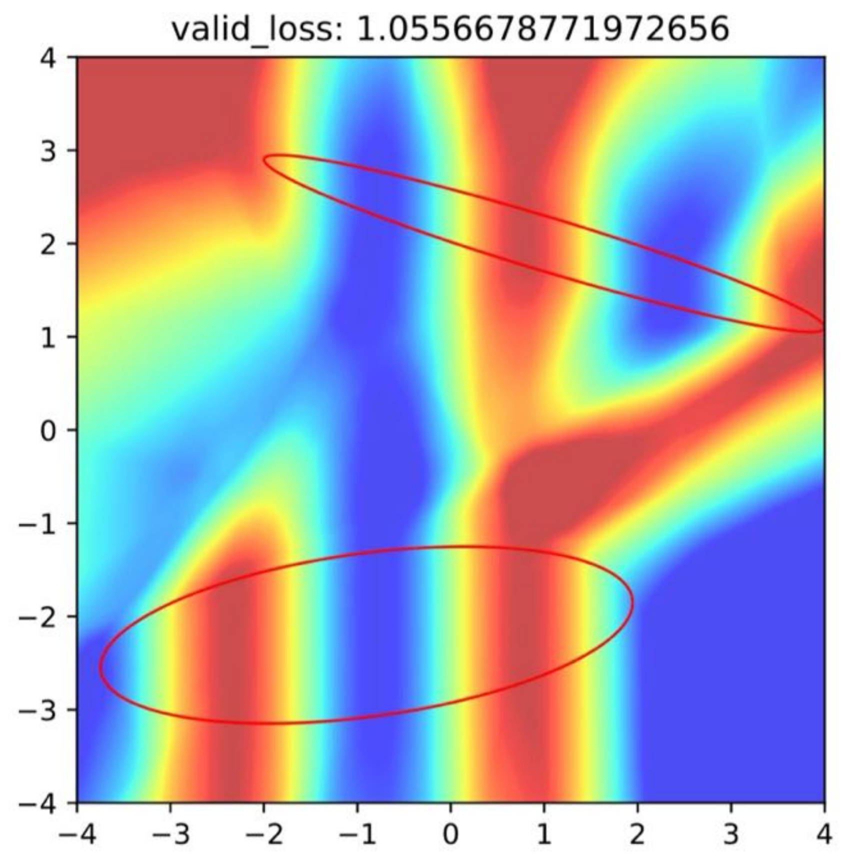
Ground Truth



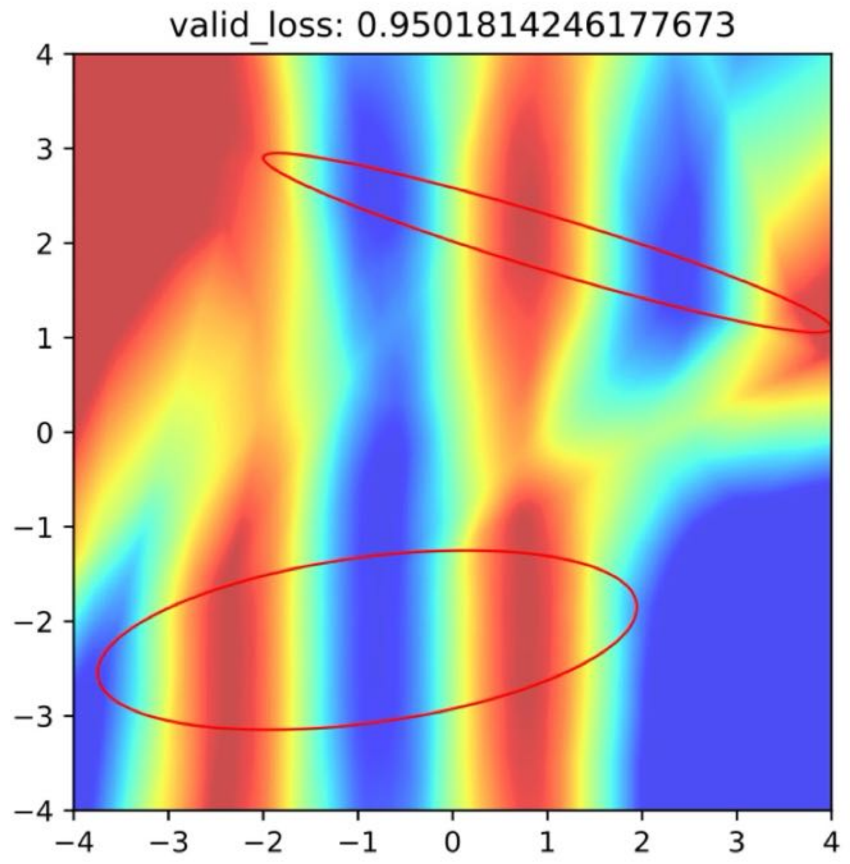
ERM



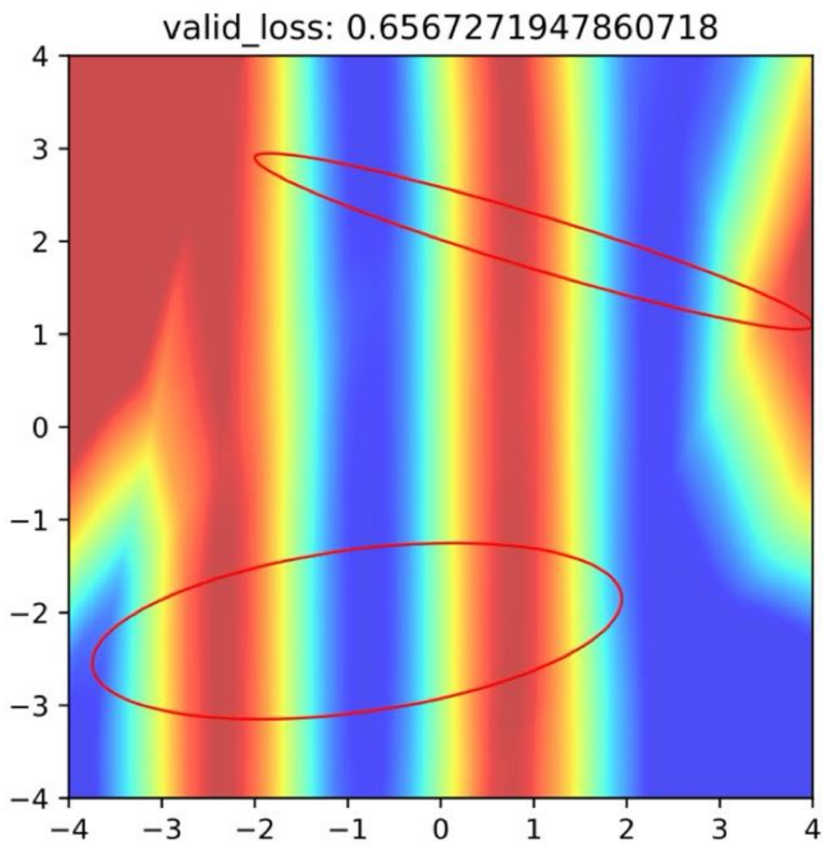
IRMv1



VREx



IRMX



PAIR

Real-world Experiments

Table 2: OOD generalization performances on WILDS benchmark.

	CAMELYON17	CIVILCOMMENTS	FMOW	IWILDCAM	POVERTYMAP	RXRX1	AVG. RANK(↓)†
	Avg. acc. (%)	Worst acc. (%)	Worst acc. (%)	Macro F1	Worst Pearson r	Avg. acc. (%)	
ERM	70.3 (±6.4)	56.0 (±3.6)	32.3 (±1.25)	30.8 (±1.3)	0.45 (±0.06)	29.9 (±0.4)	4.50
CORAL	59.5 (±7.7)	65.6 (±1.3)	31.7 (±1.24)	**32.7** (±0.2)	0.44 (±0.07)	28.4 (±0.3)	5.50
GroupDRO	68.4 (±7.3)	70.0 (±2.0)	30.8 (±0.81)	23.8 (±2.0)	0.39 (±0.06)	23.0 (±0.3)	6.83
IRMv1	64.2 (±8.1)	66.3 (±2.1)	30.0 (±1.37)	15.1 (±4.9)	0.43 (±0.07)	8.2 (±0.8)	7.67
V-REx	71.5 (±8.3)	64.9 (±1.2)	27.2 (±0.78)	27.6 (±0.7)	0.40 (±0.06)	7.5 (±0.8)	7.00
Fish	74.3 (±7.7)	73.9 (±0.2)	34.6 (±0.51)	24.8 (±0.7)	0.43 (±0.05)	10.1 (±1.5)	4.33
LISA	**74.7** (±6.1)	70.8 (±1.0)	33.5 (±0.70)	24.0 (±0.5)	**0.48** (±0.07)	**31.9** (±0.8)	2.67
IRMX	67.0 (±6.6)	74.3 (±0.8)	33.7 (±0.78)	26.6 (±0.9)	0.45 (±0.04)	28.7 (±0.2)	4.00
PAIR-o	74.0 (±7.0)	**75.2** (±0.7)	**35.5** (±1.13)	27.9 (±0.7)	0.47 (±0.06)	28.8 (±0.1)	**2.17**

† Averaged rank is reported because of the dataset heterogeneity. A lower rank is better.

PAIR re-empowers IRMv1 and achieves new state-of-the-arts across ***6 challenging realistic datasets***.

Learning Causality for Modern Machine Learning

Traditional ML assumes train and test data are **iid.**, i.e., independently sampled from an identical distribution, while data is often **OOD**, i.e., out-of-distribution, in real-world applications.

Objectives

Causal Representation Learning on Graphs: [NeurIPS'22 Spotlight, NeurIPS'23a]

Implications

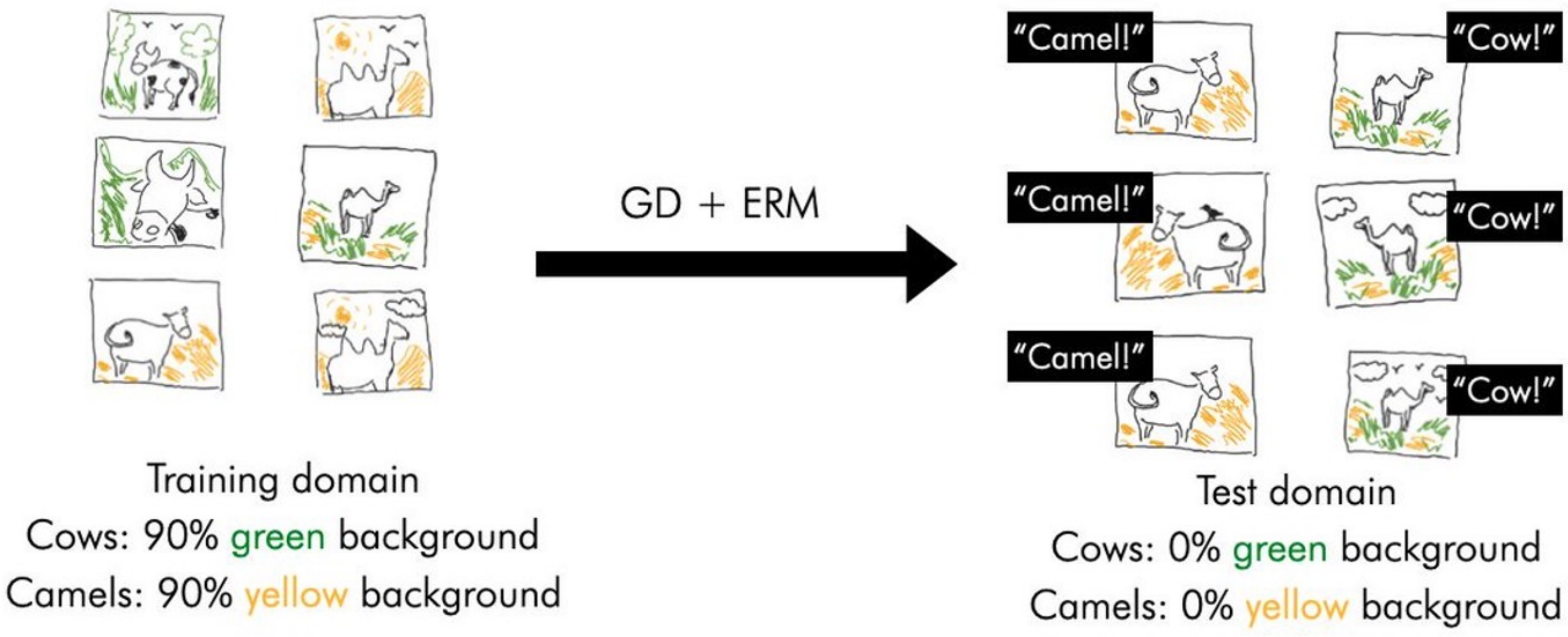
Useful Properties of the Causal Representations:
OOD Generalizability [NeurIPS'22, 23a],
Adversarial Robustness [ICLR'22],
Interpretability [ICML'24a]

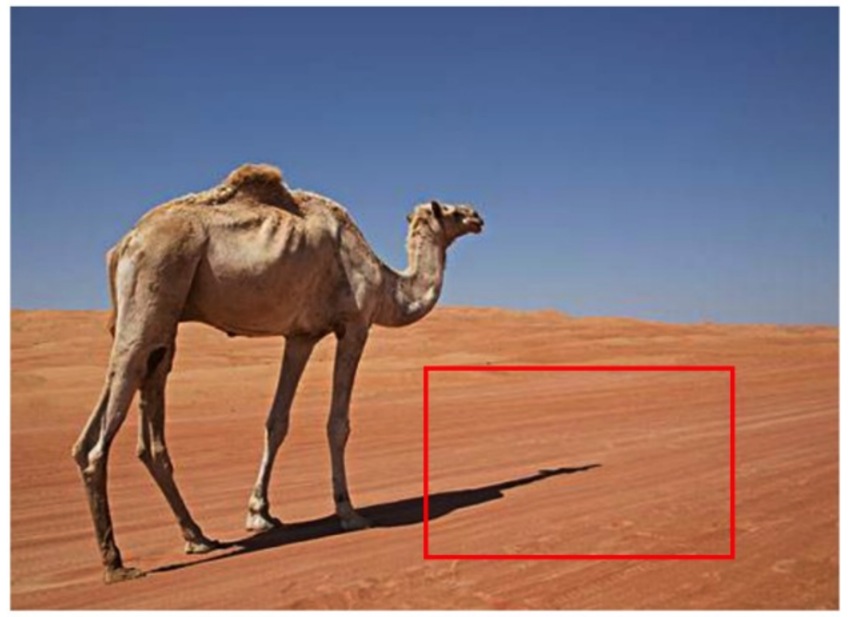
Realizations

Optimization & Feature Learning schemes for Causal Representation Learning: [ICLR'23a, NeurIPS'23b]

A Debate on ERM Feature Learning

ERM learns **predictive** but **spurious** features, that are **bad** for out-of-distribution (OOD) generalization.





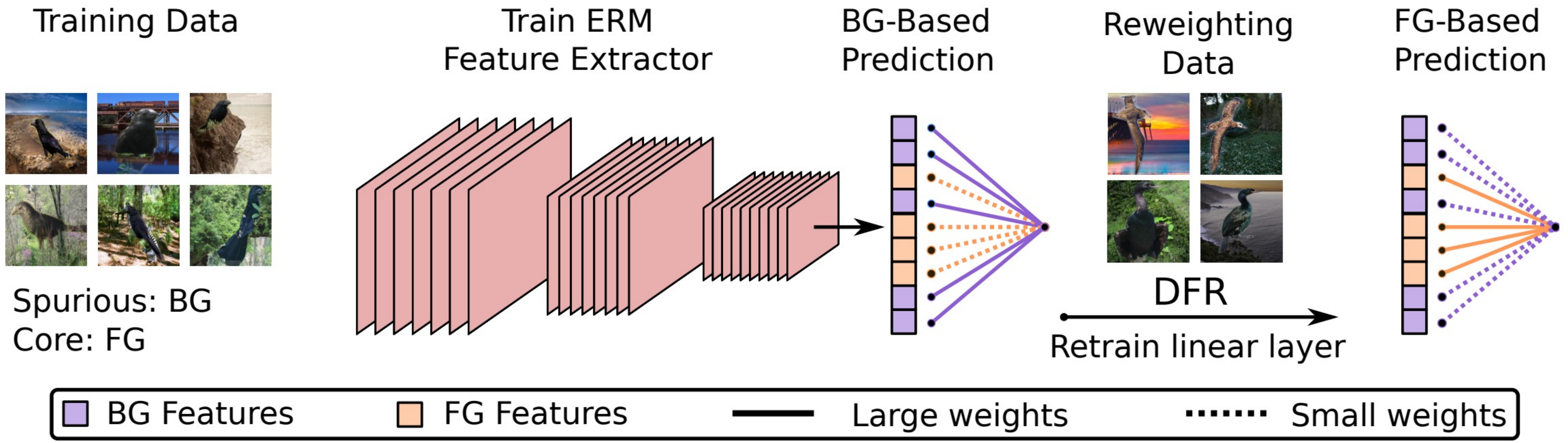
camel

cow

(Beery et al., 2018; Arjovsky et al., 2019; DeGrave et al. 2021; Ahuja et al., 2021)

A Debate on ERM Feature Learning

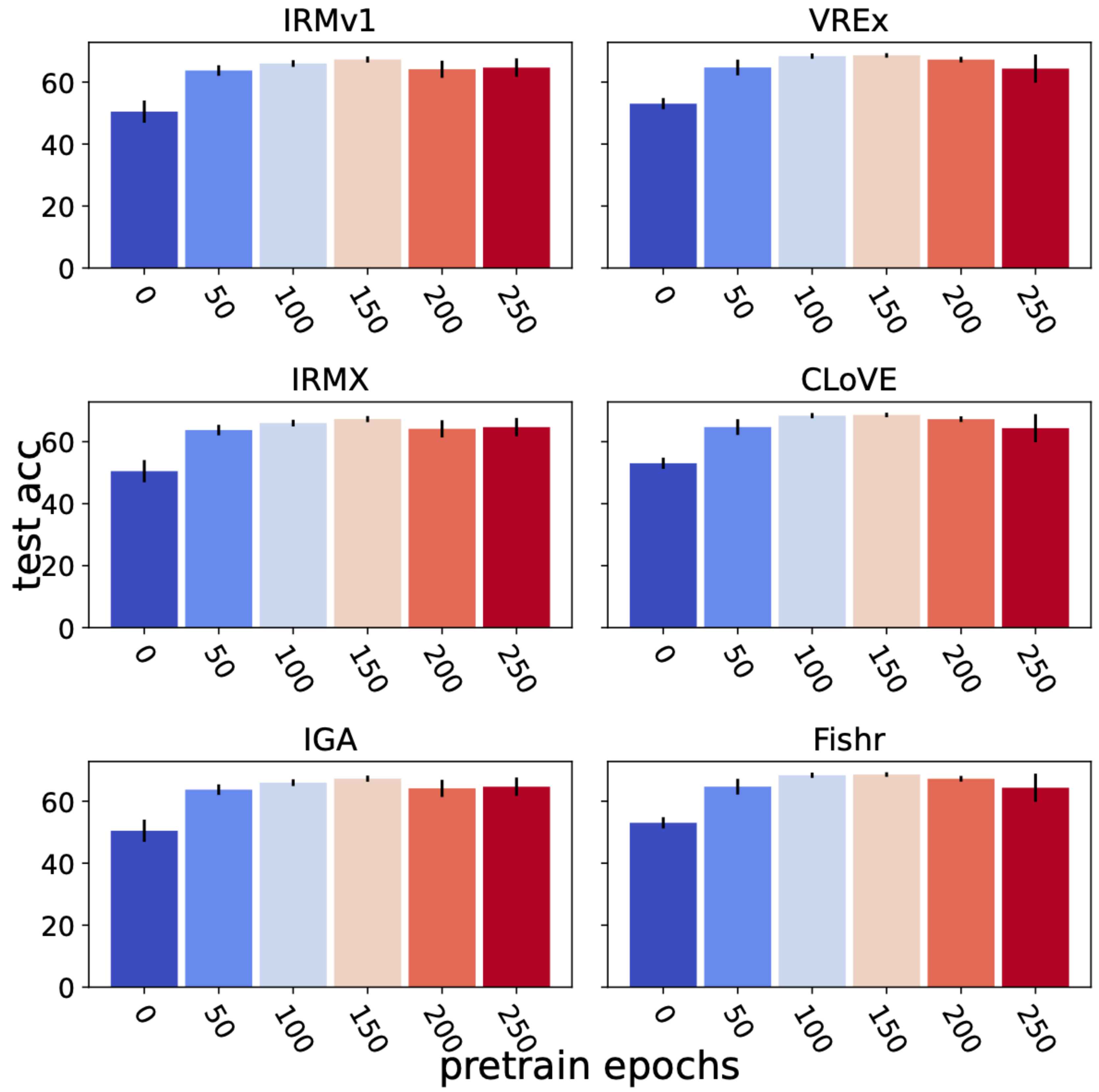
ERM already learns **invariant** features, that are **useful** for OOD generalization.



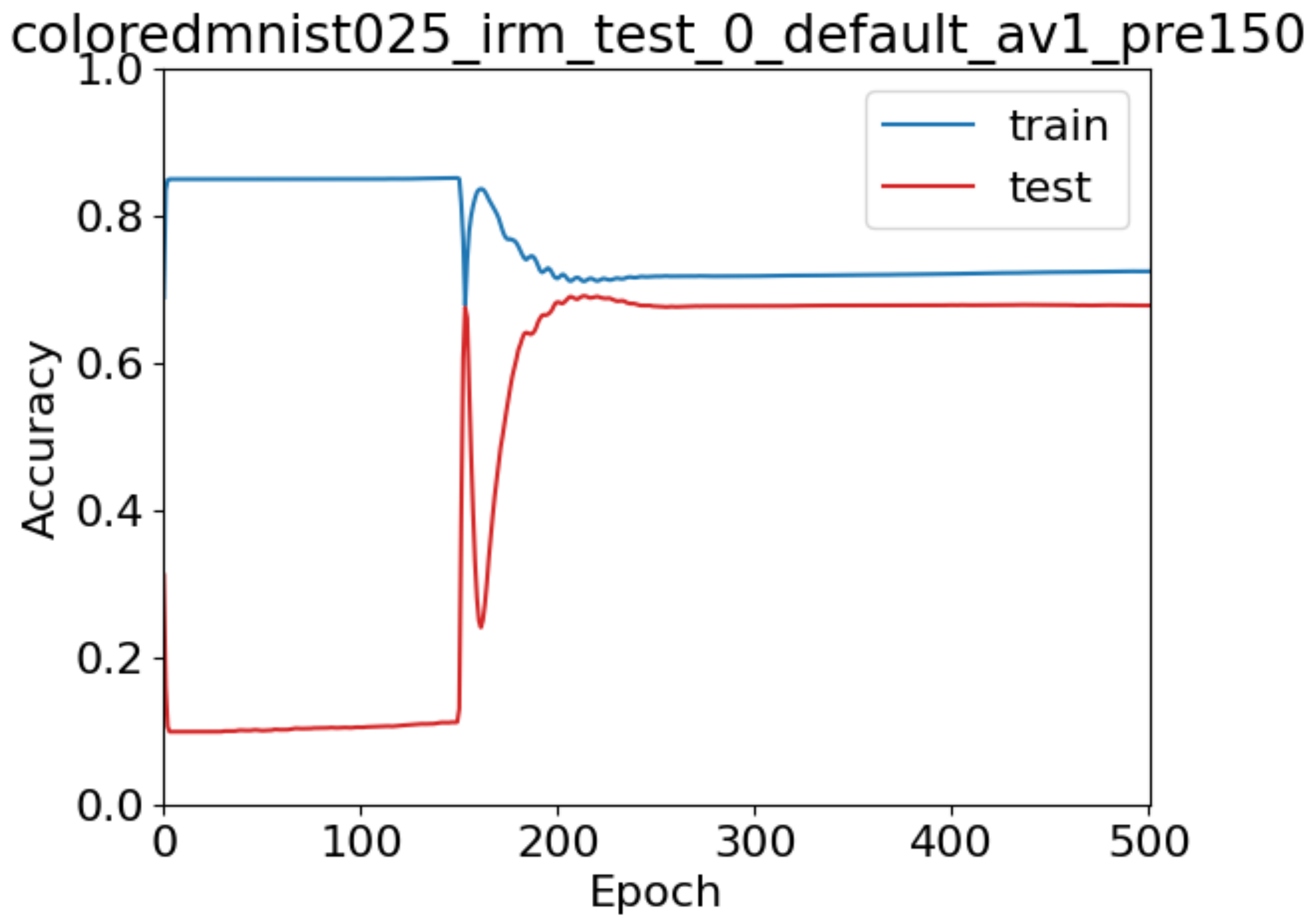
(Beery et al., 2018; Arjovsky et al., 2019; DeGrave et al. 2021; Ahuja et al., 2021)

A Debate on ERM Feature Learning

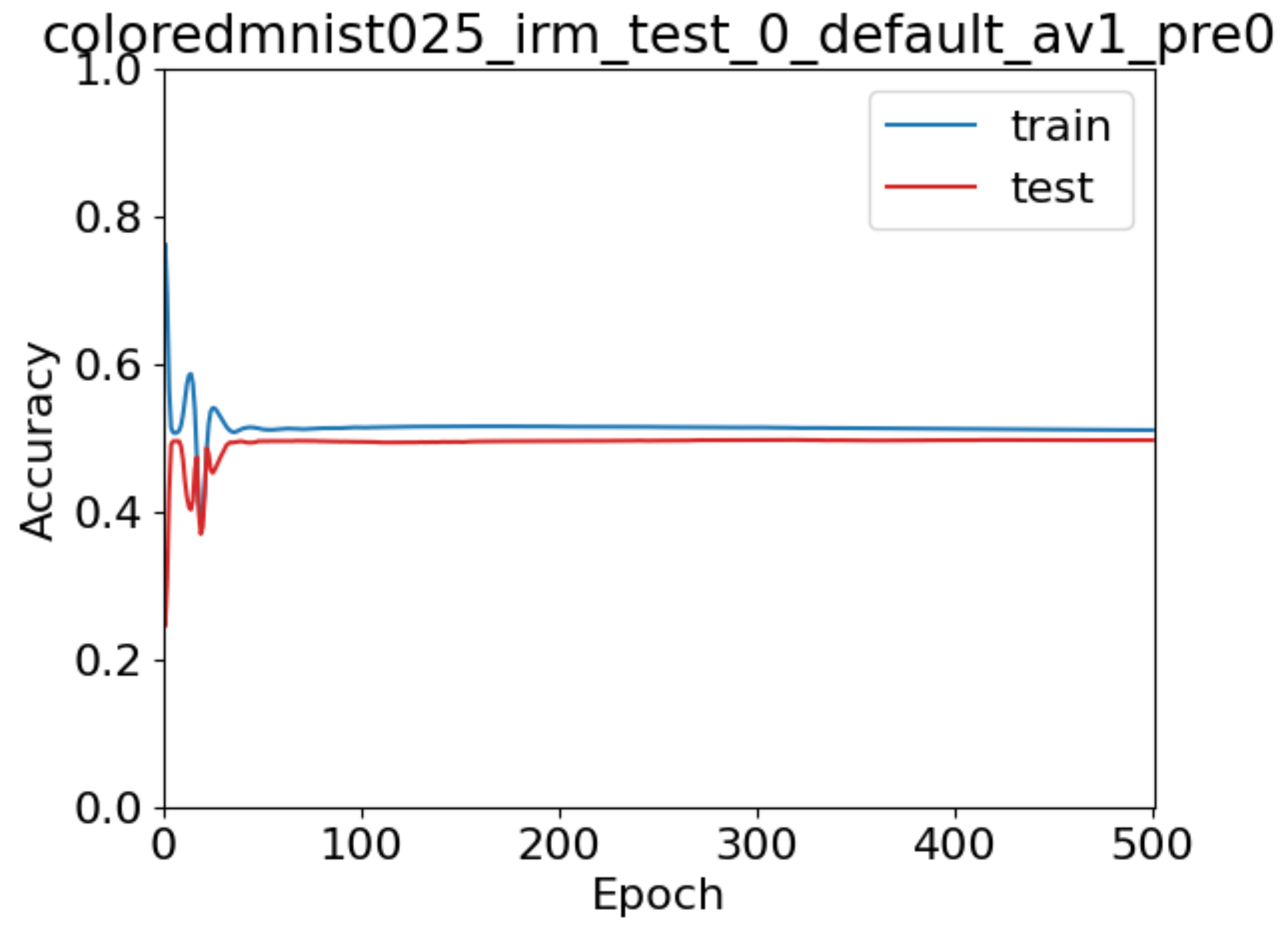
OOD generalization performance heavily **rely on** proper ERM pre-training.



OOD performance on ColoredMNIST



IRMv1 **with** ERM pretraining (150 epochs)



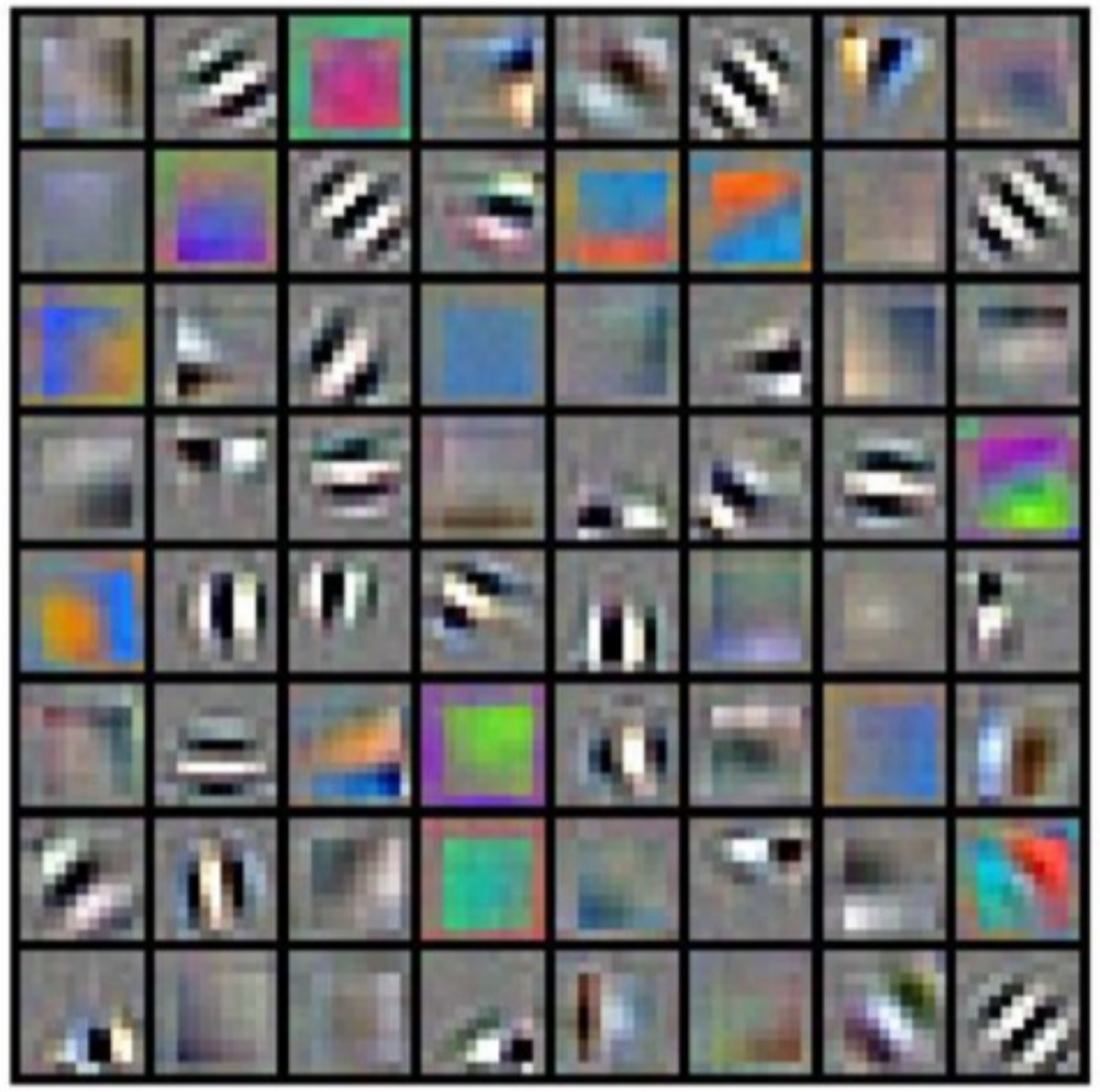
IRMv1 **w/o** ERM pretraining

(Zhang et al., 2022; Chen et al., 2022)

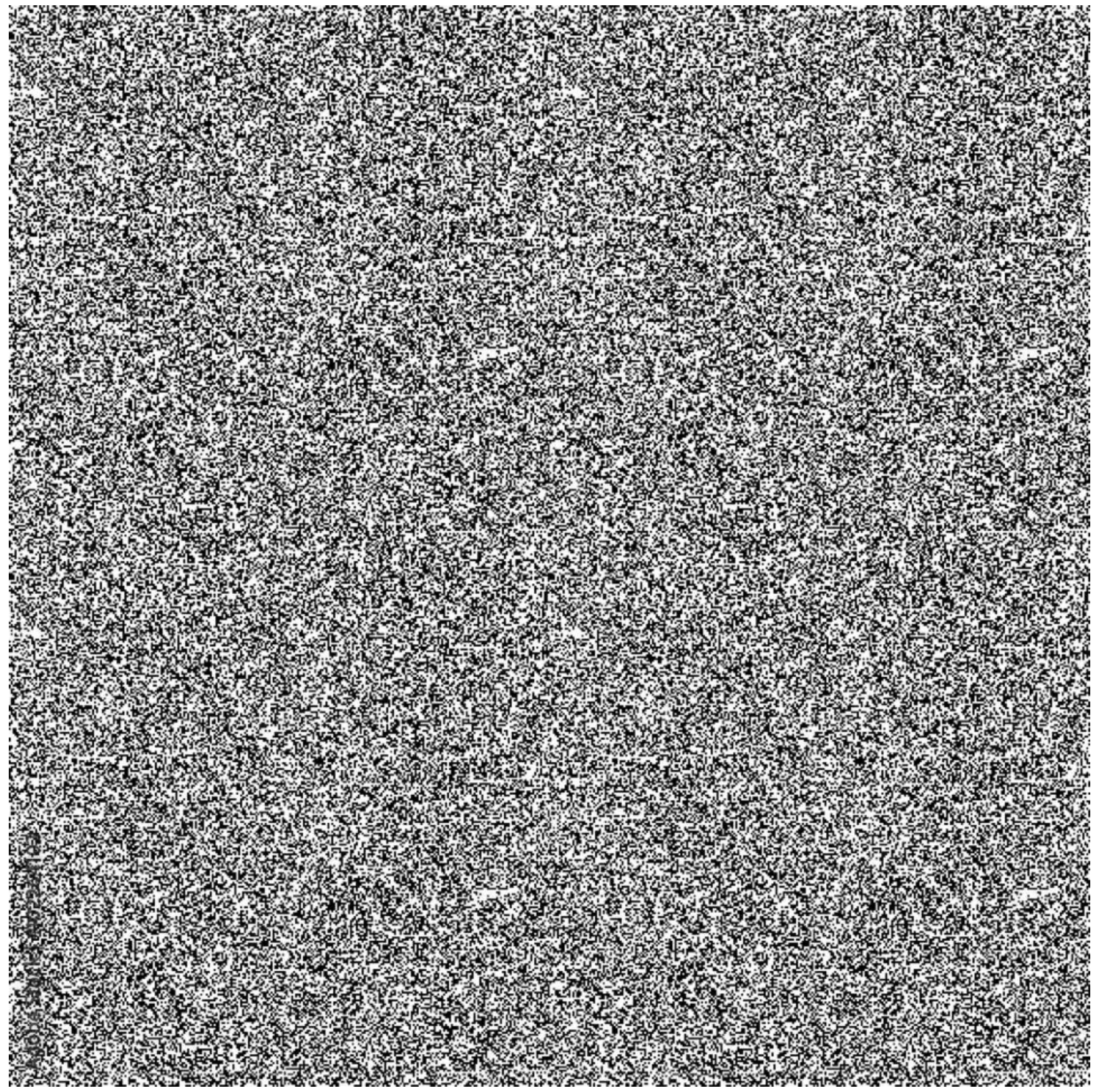
Data Model for OOD Generalization

- Two classes $y = \{-1, +1\}$
- The input $\mathbf{x} \in \mathbb{R}^{2d}$ is composed of

A feature patch $\mathbf{x}_1 \in \mathbb{R}^d$



A noise patch $\mathbf{x}_2 \in \mathbb{R}^d$



(Allen-Zhu & Li 2019)

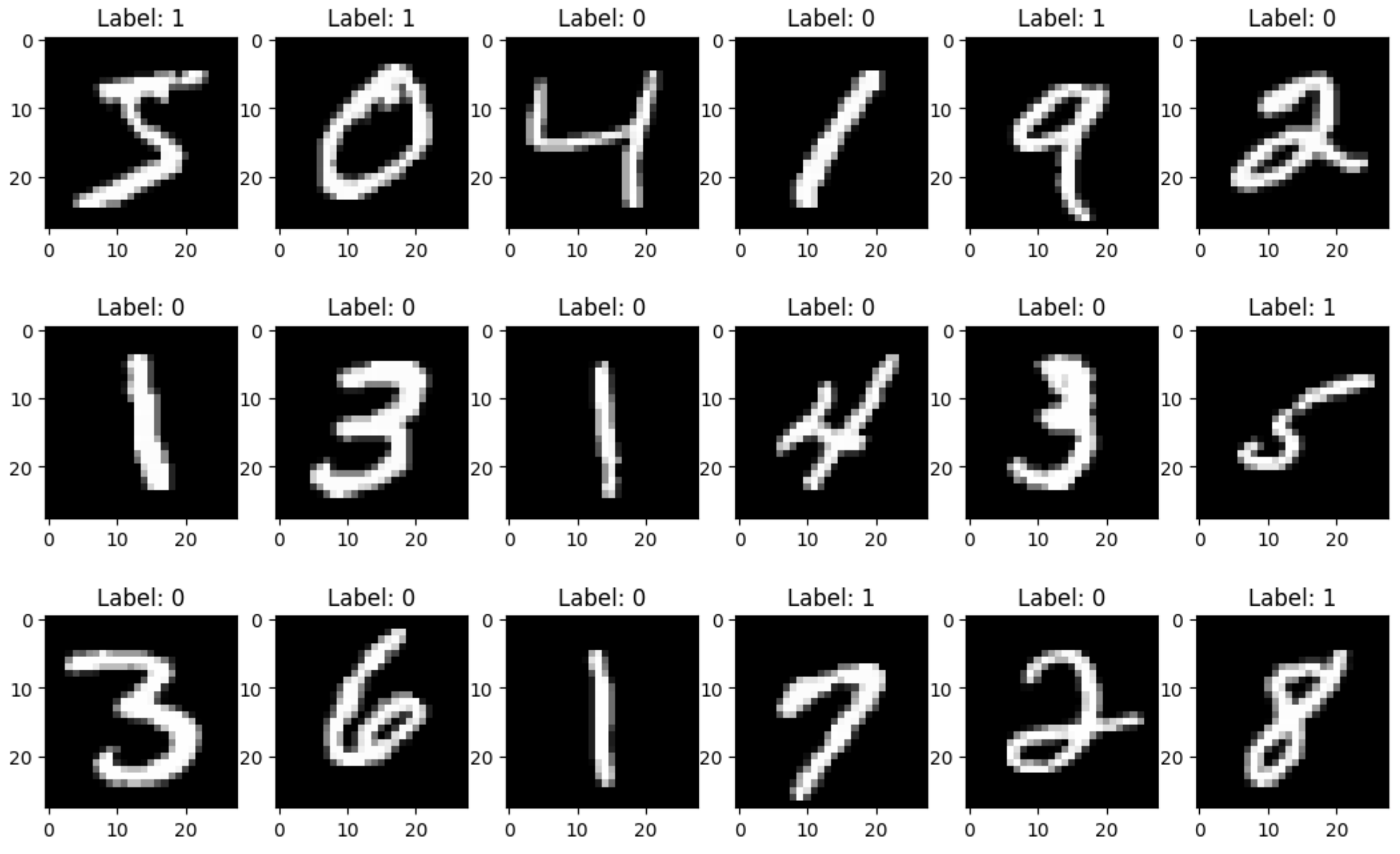
Data Model for OOD Generalization

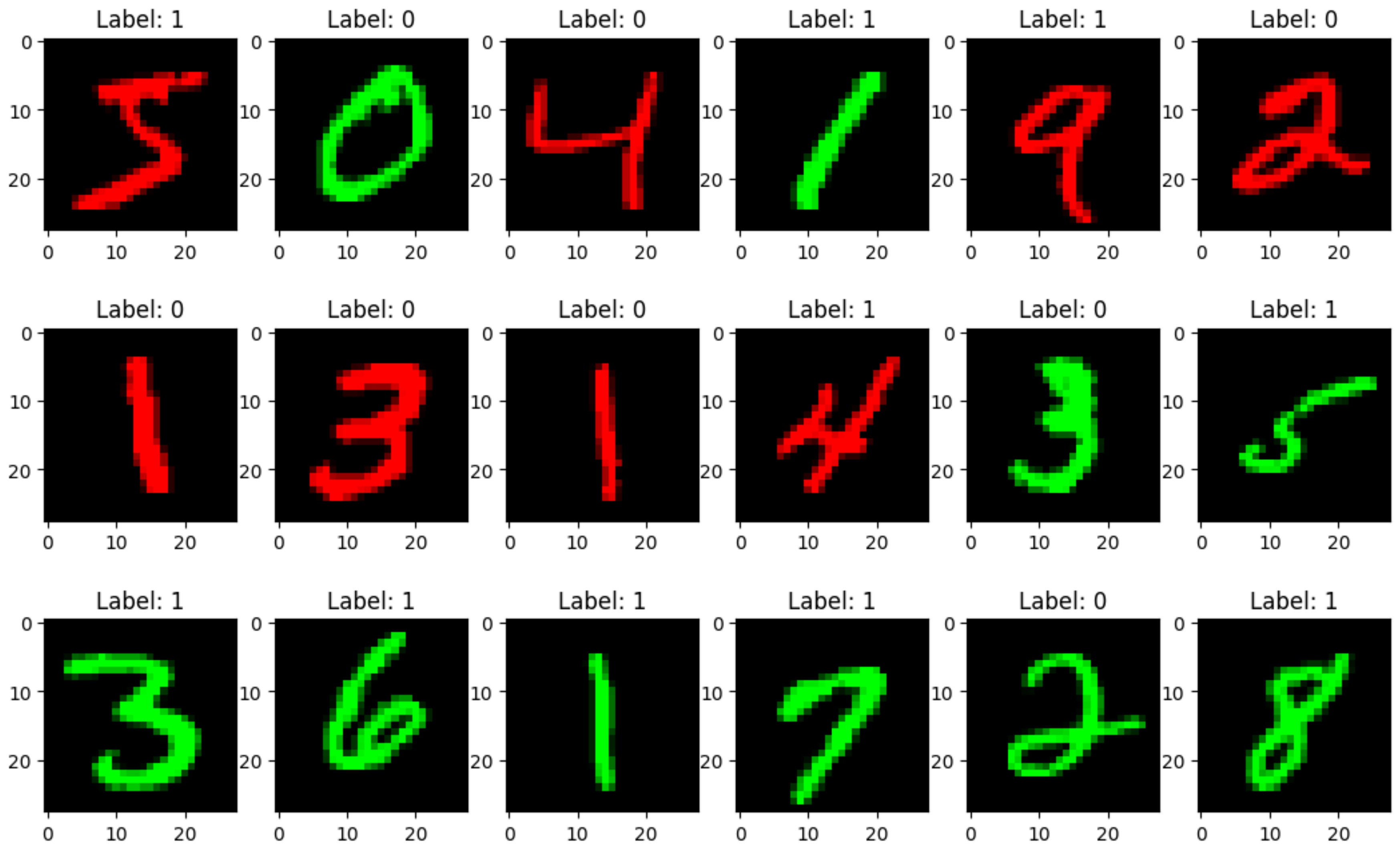
- Two classes $y = \{-1, +1\}$
- The input $\mathbf{x} \in \mathbb{R}^{2d}$ is composed of a feature patch $\mathbf{x}_1 \in \mathbb{R}^d$ and a noise patch $\mathbf{x}_2 \in \mathbb{R}^d$
- The feature patch $\mathbf{x}_1 \in \mathbb{R}^d$ is generated via:

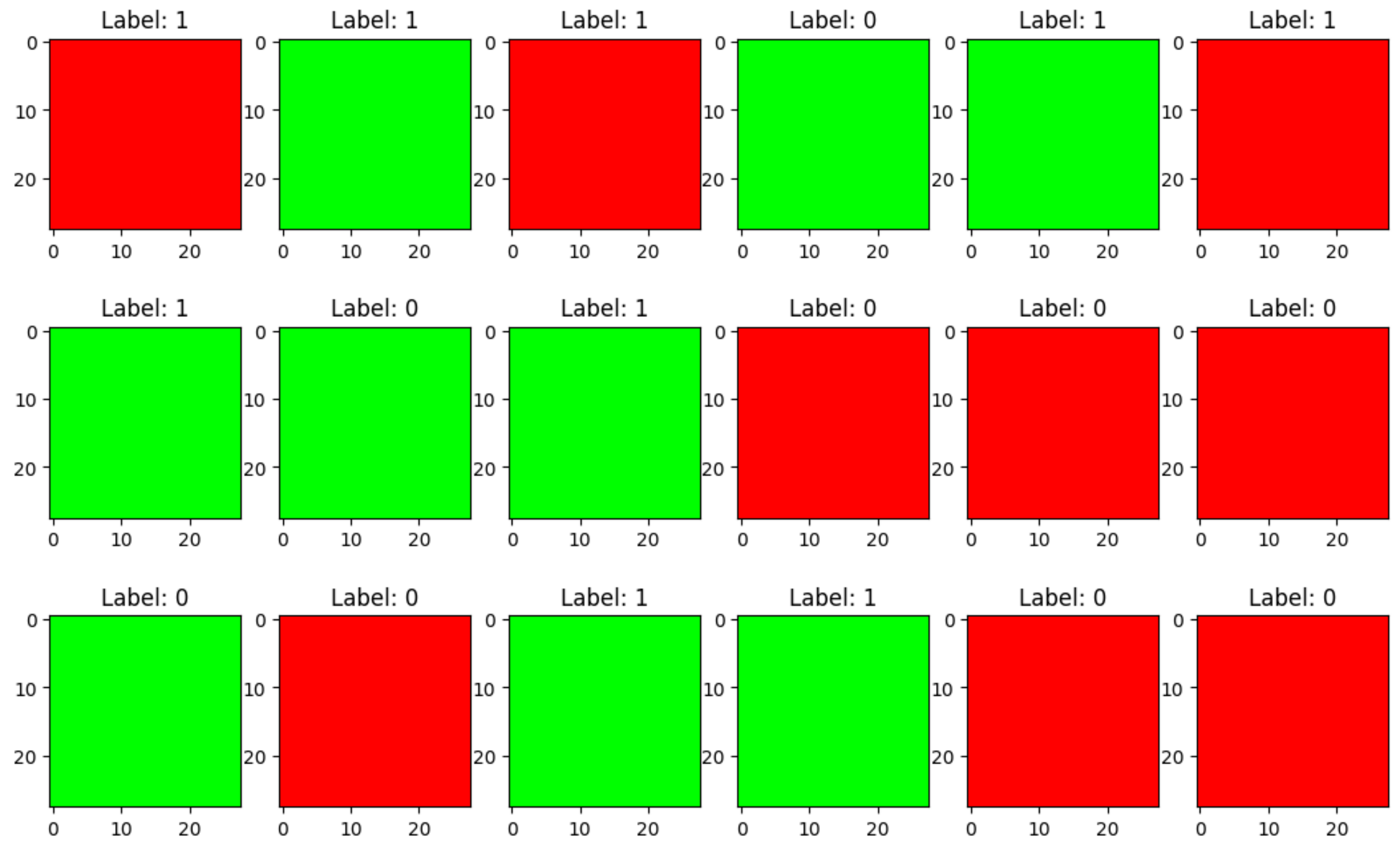
$$\mathbf{x}_1 = y \cdot \mathrm{Rad}(\alpha) \cdot \mathbf{v}_1 + y \cdot \mathrm{Rad}(\beta_e) \cdot \mathbf{v}_2$$

Invariant signal

Spurious signal

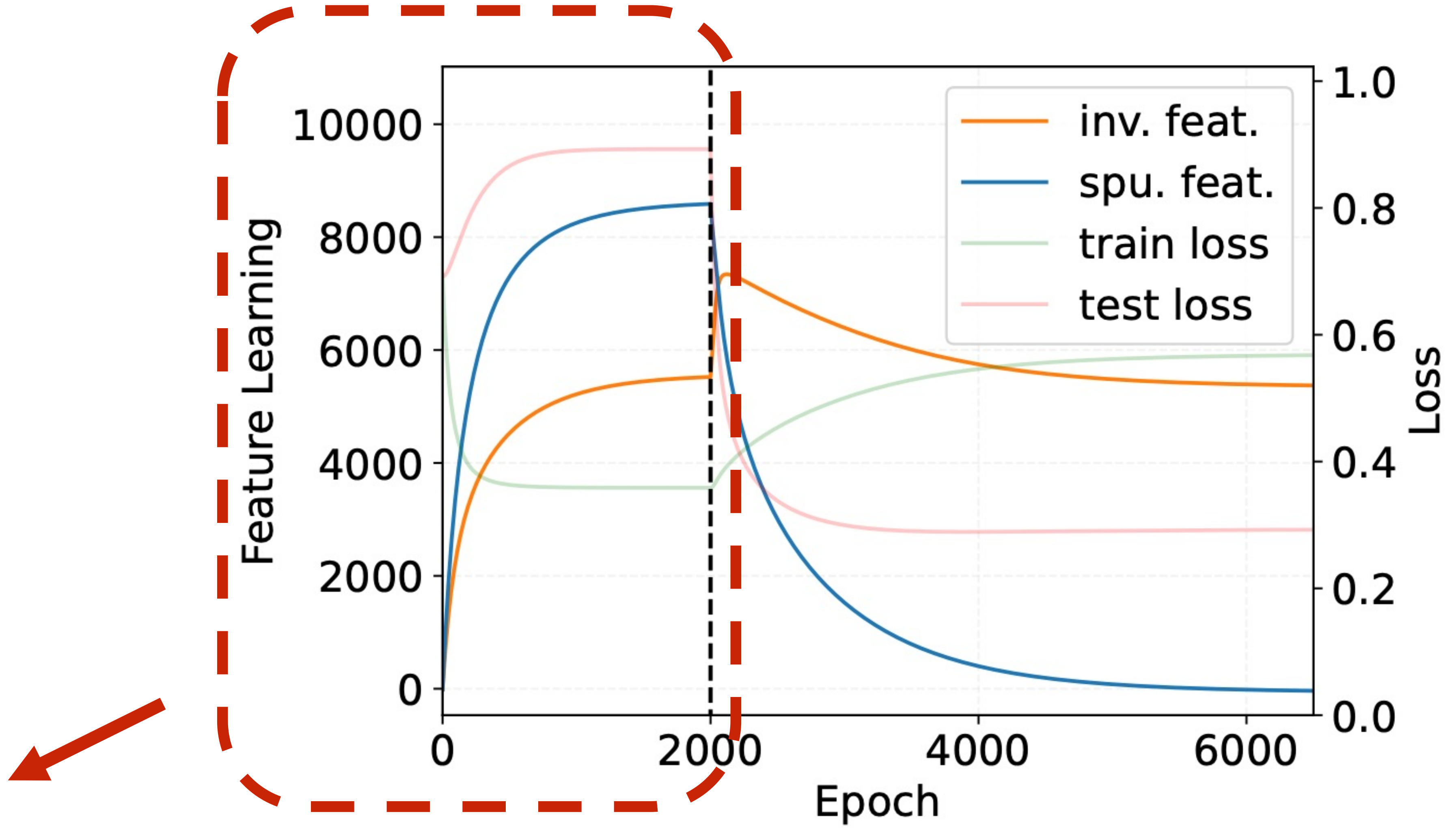






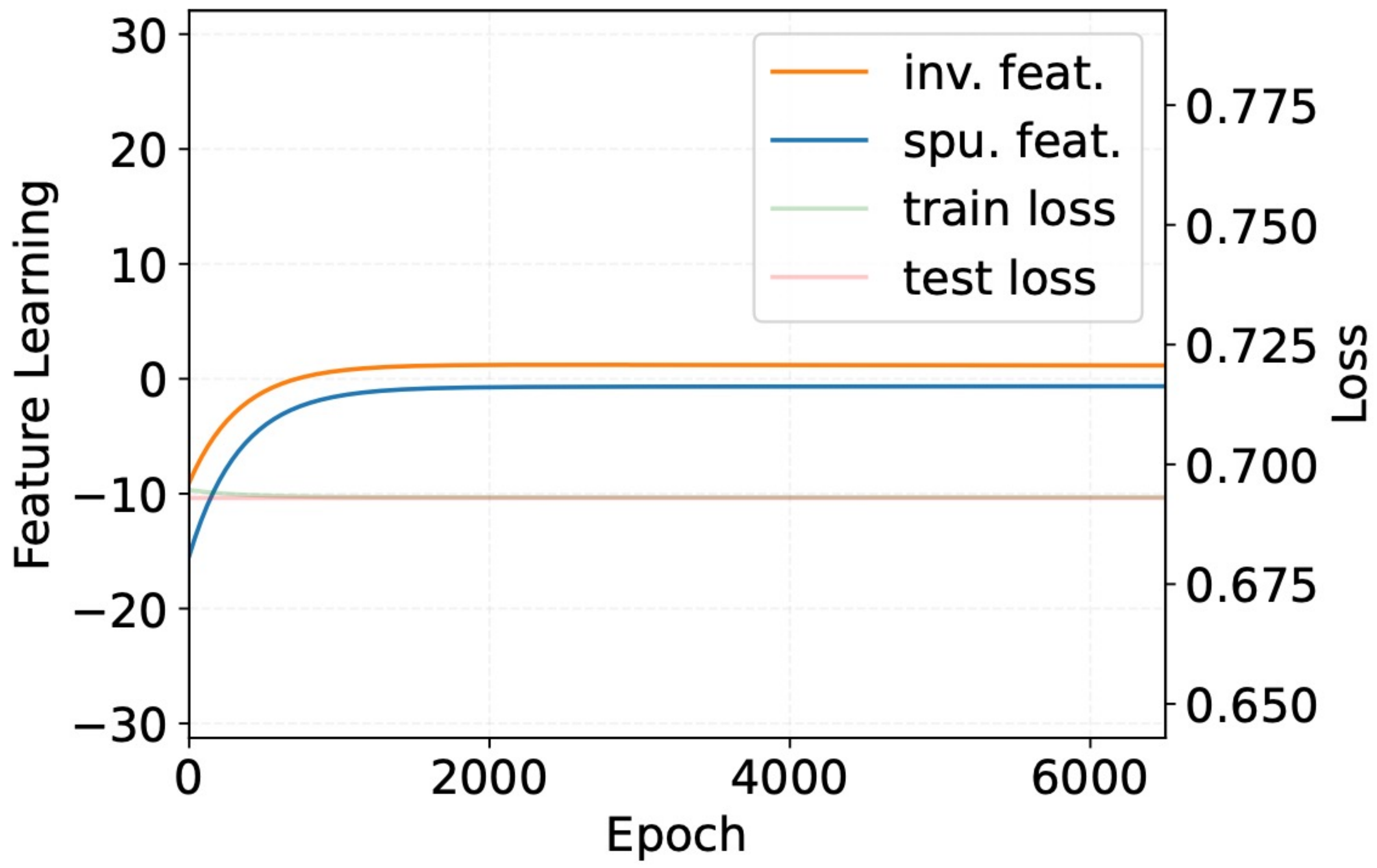
(Allen-Zhu & Li 2019)

ERM and IRM Feature Learning



ERM pre-training

FL w/ pre-training



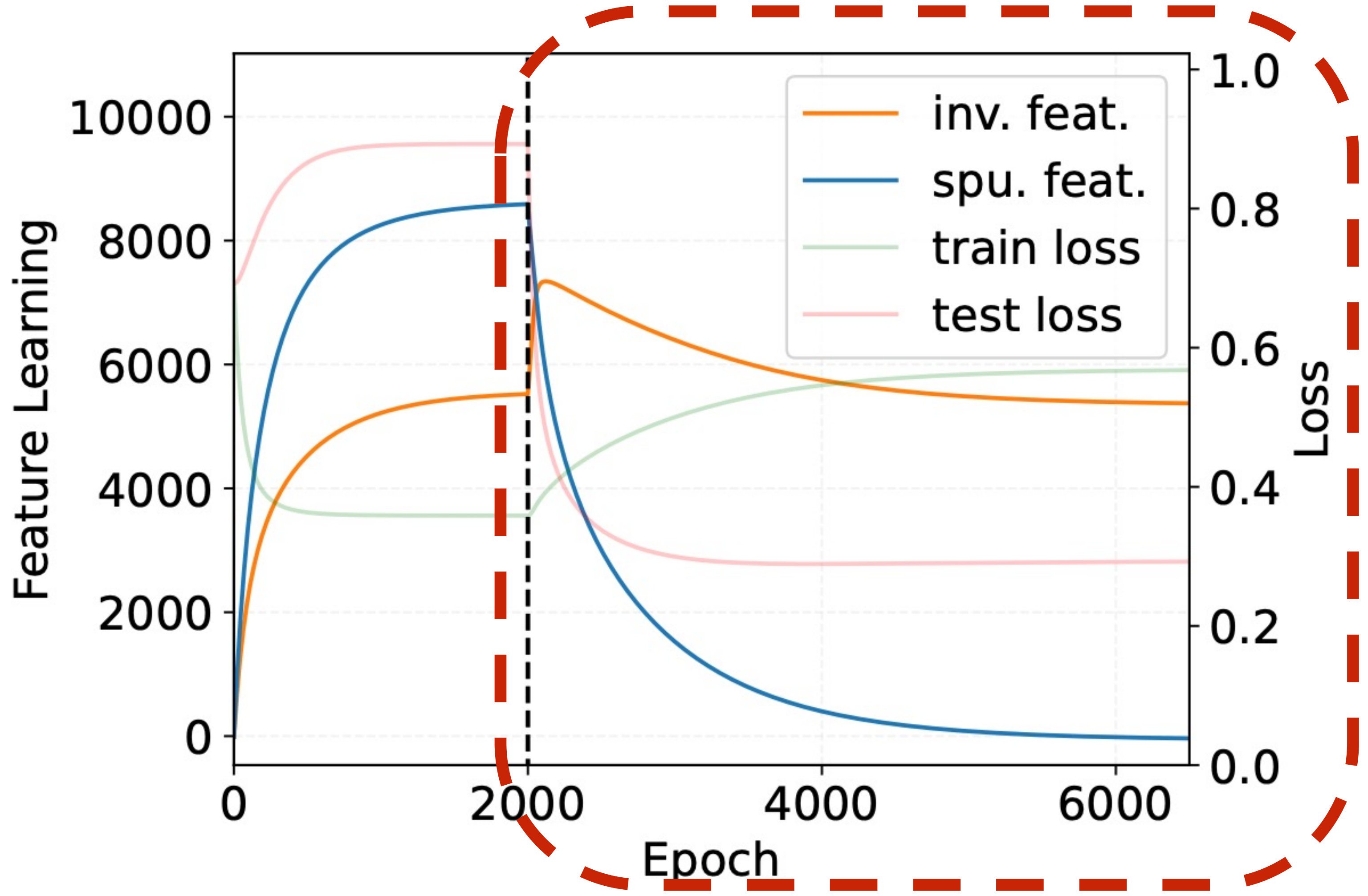
FL w/o pre-training

Theoretical Results (Informal):

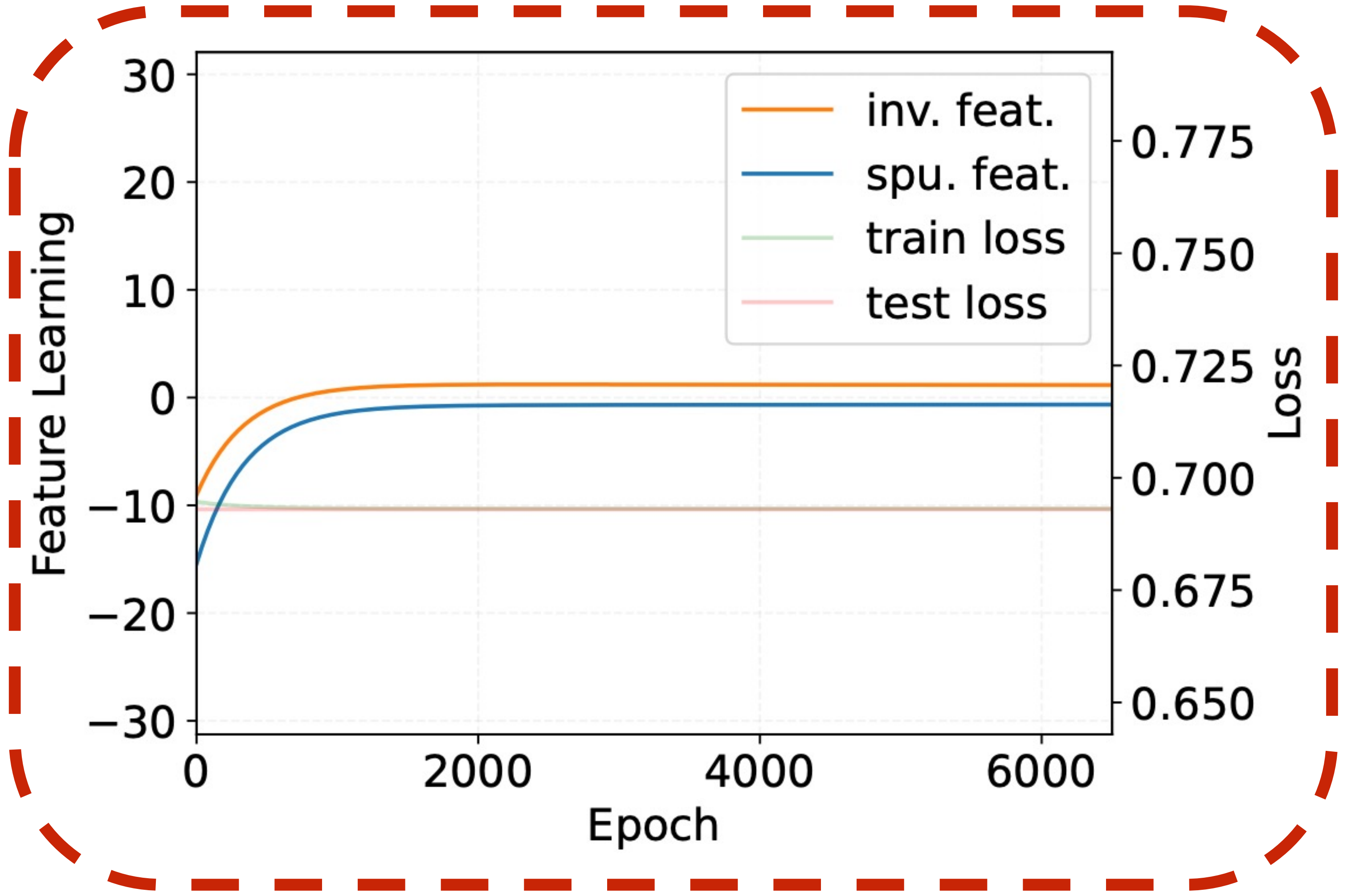
- ERM learns ***both*** invariant and spurious features.
- The invariant and spurious feature learning speed depends on the ***correlation strength*** with the labels.

ERM and IRM Feature Learning

OOD training with IRMv1



FL w/ pre-training



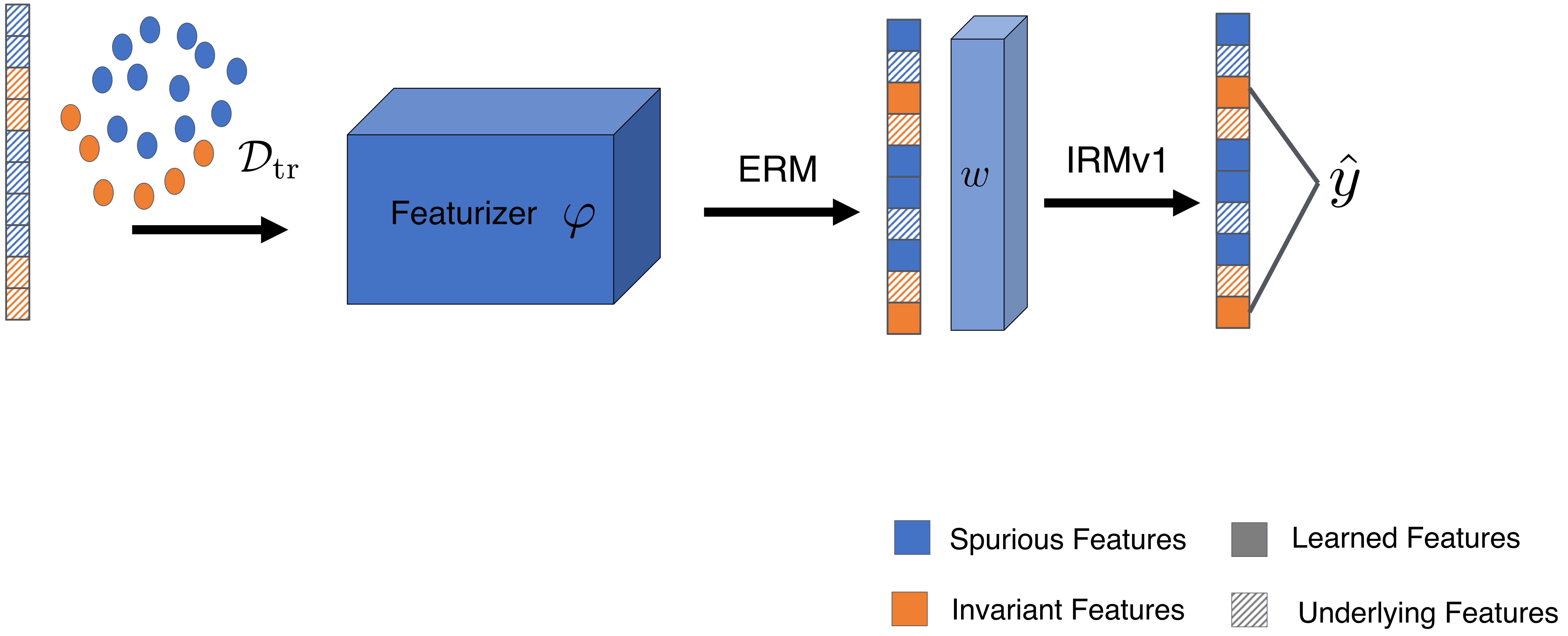
FL w/o pre-training

Theoretical Results (Informal):

- IRMv1 ***cannot*** learn any features even at the beginning of training;
- IRMv1 highly ***relies on*** ERM pre-training feature quality to extract invariant features.

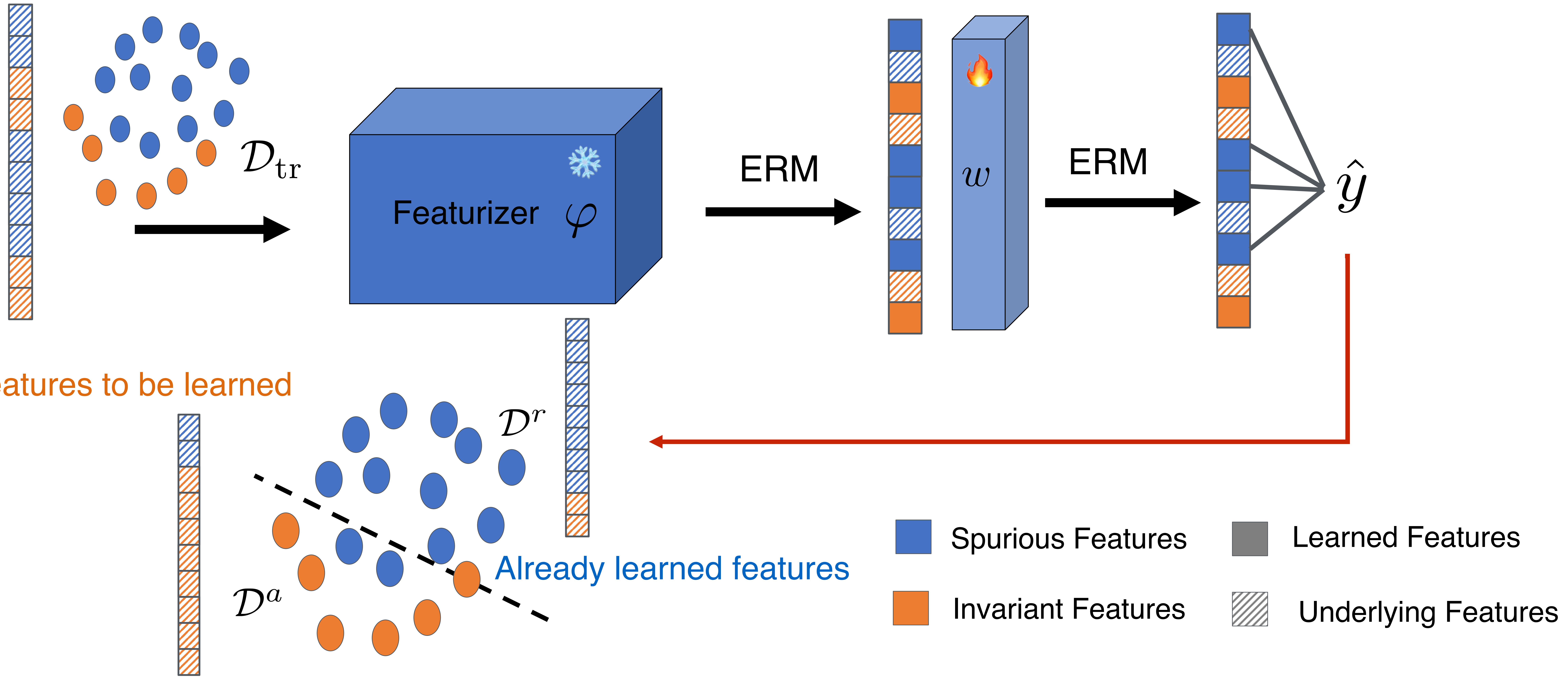
Feature Learning with ERM

OOD training can only leverage ***limited*** invariant features for prediction.



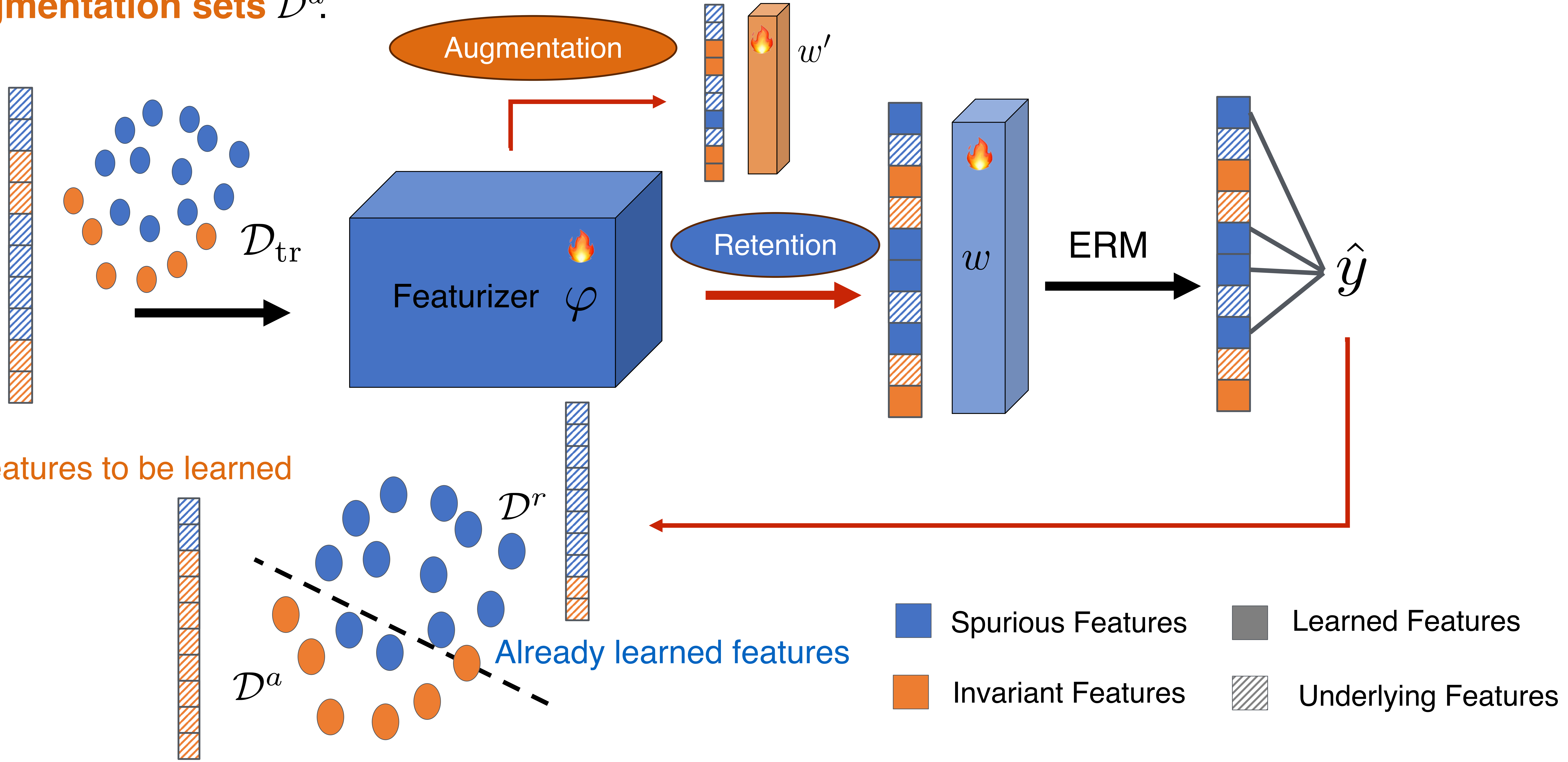
FeAT: Feature Augmented Training

Leveraging the feature learning information can partition the dataset into **retention sets** $\mathcal{D}^r$ and **augmentation sets** $\mathcal{D}^a$.



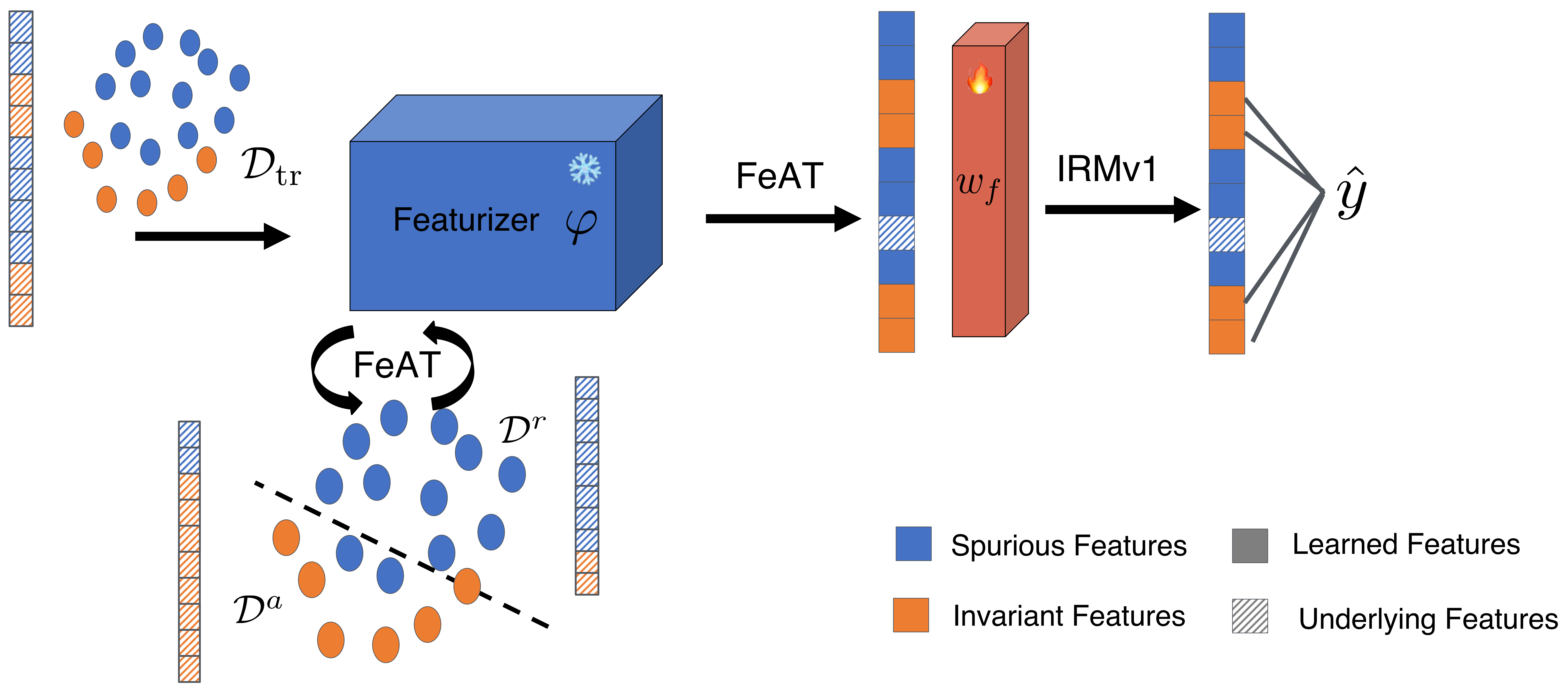
FeAT: Feature Augmented Training

Leveraging the feature learning information can partition the dataset into **retention sets** $\mathcal{D}^r$ and **augmentation sets** $\mathcal{D}^a$.



FeAT: Feature Augmented Training

Performing **feature augmentation** and **retention** several rounds, we can obtain richer feature representations that facilitate better OOD generalization.



Experimental Results

FeAT boosts OOD performance of various objectives across various ColoredMNIST variant datasets.

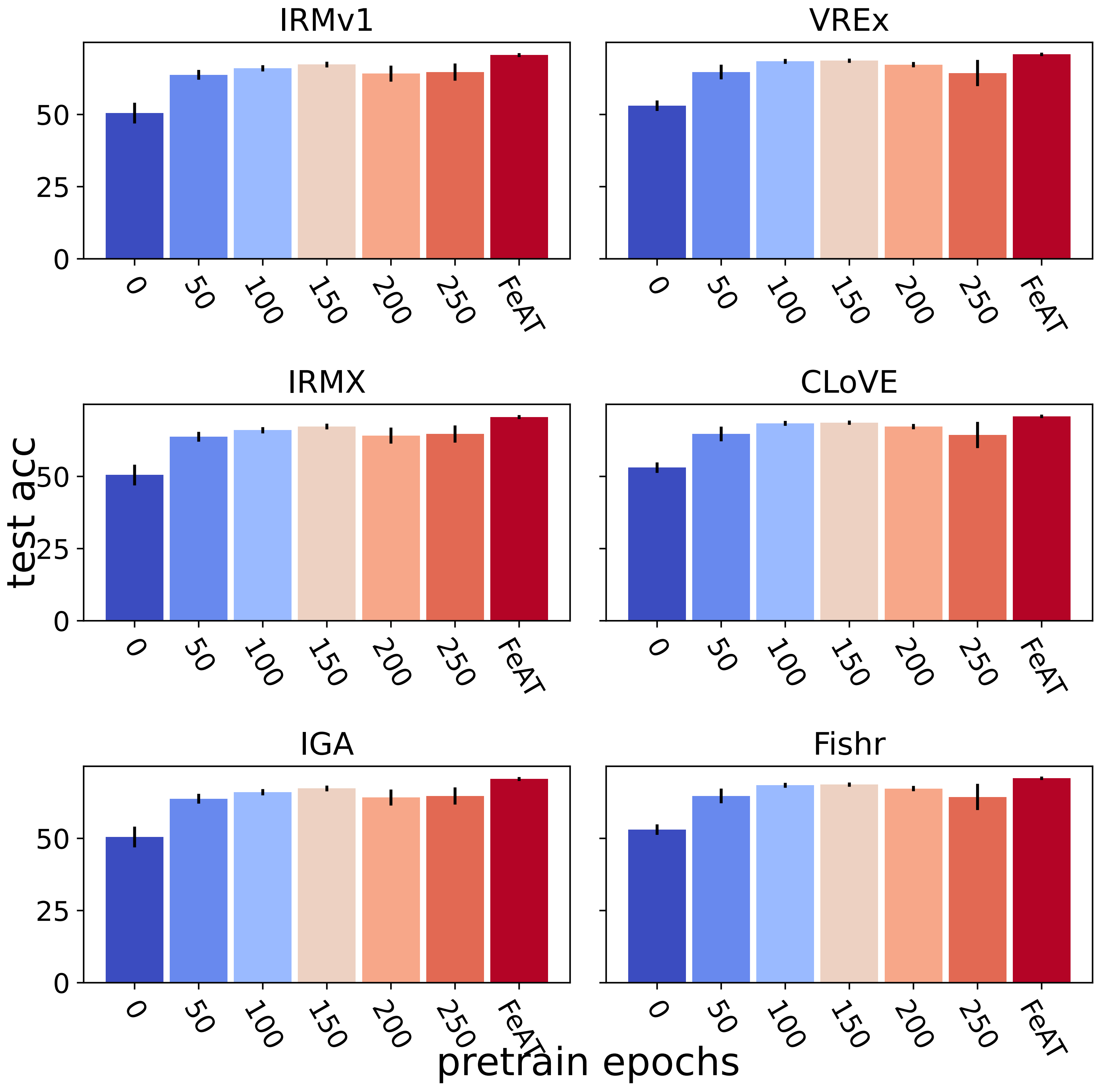


Table 1: OOD performance on COLOREDMNIST datasets initialized with different representations.

	COLOREDMNIST-025				COLOREDMNIST-01			
	ERM-NF	ERM	BONSAI	FEAT	ERM-NF	ERM	BONSAI	FEAT
ERM	17.14 (±0.73)	12.40 (±0.32)	11.21 (±0.49)	**17.27** (±2.55)	73.06 (±0.71)	73.75 (±0.49)	70.95 (±0.93)	**76.05** (±1.45)
IRMv1	67.29 (±0.99)	59.81 (±4.46)	70.28 (±0.72)	**70.57** (±0.68)	76.89 (±3.25)	73.84 (±0.56)	76.71 (±4.10)	**82.33** (±1.77)
V-REX	68.62 (±0.73)	65.96 (±1.29)	70.31 (±0.66)	**70.82** (±0.59)	83.52 (±2.52)	81.20 (±3.27)	82.61 (±1.76)	**84.70** (±0.69)
IRMX	67.00 (±1.95)	64.05 (±0.88)	70.46 (±0.42)	**70.78** (±0.61)	81.61 (±1.98)	75.97 (±0.88)	80.28 (±1.62)	**84.34** (±0.97)
IB-IRM	56.09 (±2.04)	59.81 (±4.46)	70.28 (±0.72)	**70.57** (±0.68)	75.81 (±0.63)	73.84 (±0.56)	76.71 (±4.10)	**82.33** (±1.77)
CLOVE	58.67 (±7.69)	65.78 (±0.00)	65.57 (±3.02)	**65.78** (±2.68)	75.66 (±10.6)	74.73 (±0.36)	72.73 (±1.18)	**75.12** (±1.08)
IGA	51.22 (±3.67)	62.43 (±3.06)	**70.17** (±0.89)	67.11 (±3.40)	74.20 (±2.45)	73.74 (±0.48)	74.72 (±3.60)	**83.46** (±2.17)
FISHR	69.38 (±0.39)	67.74 (±0.90)	68.75 (±1.10)	**70.56** (±0.97)	77.29 (±1.61)	82.23 (±1.35)	84.19 (±0.66)	**84.26** (±0.93)
ORACLE	71.97 (±0.34)				86.55 (±0.27)			

Stronger spurious signal

Stronger invariant signal

Real-World Experimental Results

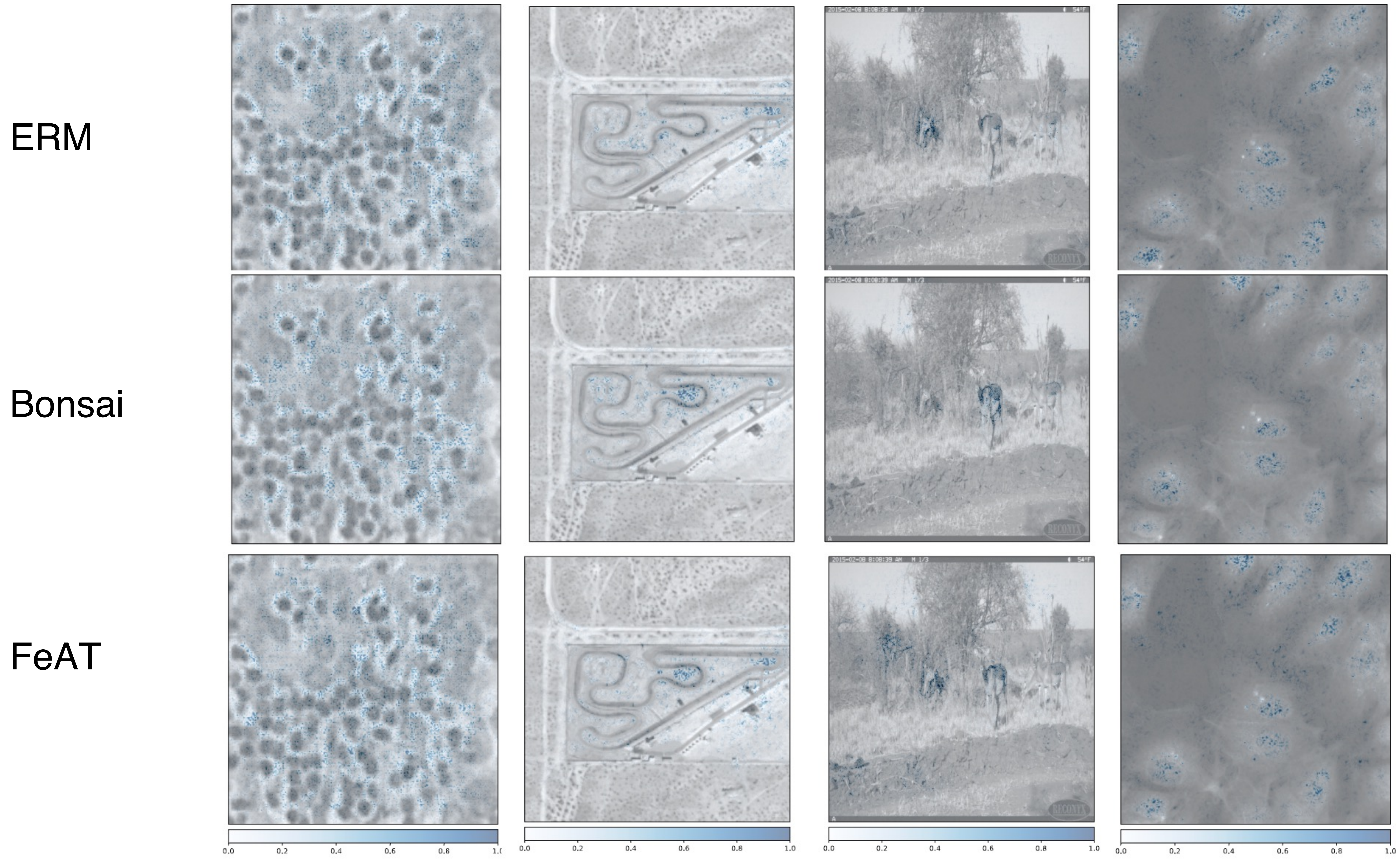
FeAT boosts OOD performance of various objectives across ***6*** challenging real-world OOD datasets.

Table 2: OOD generalization performances on WILDS benchmark.

INIT.	METHOD	CAMELYON17	CIVILCOMMENTS	FMOW	IWILDCAM	AMAZON	RXRX1
		Avg. acc. (%)	Worst acc. (%)	Worst acc. (%)	Macro F1	10-th per. acc. (%)	Avg. acc. (%)
ERM	DFR†	95.14 (±*1.96*)	**77.34** (±*0.50*)	41.96 (±*1.90*)	23.15 (±*0.24*)	48.00 (±*0.00*)	-
ERM	DFR-s†	-	82.24 (±*0.13*)	56.17 (±*0.62*)	52.44 (±*0.34*)	-	-
Bonsai	DFR†	95.17 (±*0.18*)	77.07 (±*0.85*)	43.26 (±*0.82*)	21.36 (±*0.41*)	46.67 (±*0.00*)	-
Bonsai	DFR-s†	-	81.26 (±*1.86*)	58.58 (±*1.17*)	50.85 (±*0.18*)	-	-
FeAT	DFR†	**95.28** (±*0.19*)	**77.34** (±*0.59*)	**43.54** (±*1.26*)	**23.54** (±*0.52*)	**49.33** (±*0.00*)	-
FeAT	DFR-s†	-	79.56 (±*0.38*)	57.69 (±*0.78*)	52.31 (±*0.38*)	-	-
ERM	ERM	74.30 (±*5.96*)	55.53 (±*1.78*)	33.58 (±*1.02*)	28.22 (±*0.78*)	51.11 (±*0.63*)	30.21 (±*0.09*)
ERM	GroupDRO	76.09 (±*6.46*)	69.50 (±*0.15*)	33.03 (±*0.52*)	28.51 (±*0.58*)	52.00 (±*0.00*)	29.99 (±*0.13*)
ERM	IRMv1	75.68 (±*7.41*)	68.84 (±*0.95*)	33.45 (±*1.07*)	28.76 (±*0.45*)	52.00 (±*0.00*)	30.10 (±*0.05*)
ERM	V-REx	71.60 (±*7.88*)	69.03 (±*1.08*)	33.06 (±*0.46*)	28.82 (±*0.47*)	52.44 (±*0.63*)	29.88 (±*0.35*)
ERM	IRMX	73.49 (±*9.33*)	68.91 (±*1.19*)	33.13 (±*0.86*)	28.82 (±*0.47*)	52.00 (±*0.00*)	30.10 (±*0.05*)
Bonsai	ERM	73.98 (±*5.30*)	63.34 (±*3.49*)	31.91 (±*0.51*)	28.27 (±*1.05*)	48.58 (±*0.56*)	24.22 (±*0.44*)
Bonsai	GroupDRO	72.82 (±*5.37*)	70.23 (±*1.33*)	33.12 (±*1.20*)	27.16 (±*1.18*)	42.67 (±*1.09*)	22.95 (±*0.46*)
Bonsai	IRMv1	73.59 (±*6.16*)	68.39 (±*2.01*)	32.51 (±*1.23*)	27.60 (±*1.57*)	47.11 (±*0.63*)	23.35 (±*0.43*)
Bonsai	V-REx	76.39 (±*5.32*)	68.67 (±*1.29*)	33.17 (±*1.26*)	25.81 (±*0.42*)	48.00 (±*0.00*)	23.34 (±*0.42*)
Bonsai	IRMX	64.77 (±*10.1*)	69.56 (±*0.95*)	32.63 (±*0.75*)	27.62 (±*0.66*)	46.67 (±*0.00*)	23.34 (±*0.40*)
FeAT	ERM	77.80 (±*2.48*)	68.11 (±*2.27*)	33.13 (±*0.78*)	28.47 (±*0.67*)	**52.89** (±*0.63*)	**30.66** (±*0.42*)
FeAT	GroupDRO	**80.41** (±*3.30*)	**71.29** (±*0.46*)	33.55 (±*1.67*)	28.38 (±*1.32*)	52.58 (±*0.56*)	29.99 (±*0.11*)
FeAT	IRMv1	77.97 (±*3.09*)	70.33 (±*1.14*)	**34.04** (±*0.70*)	**29.66** (±*1.52*)	**52.89** (±*0.63*)	29.99 (±*0.19*)
FeAT	V-REx	75.12 (±*6.55*)	70.97 (±*1.06*)	34.00 (±*0.71*)	29.48 (±*1.94*)	**52.89** (±*0.63*)	30.57 (±*0.53*)
FeAT	IRMX	76.91 (±*6.76*)	71.18 (±*1.10*)	33.99 (±*0.73*)	29.04 (±*2.96*)	**52.89** (±*0.63*)	29.92 (±*0.16*)

†DFR/DFR-s use an additional OOD dataset to evaluate invariant and spurious feature learning, respectively.

FeAT Learns Richer Meaningful Features



(i) CAMELYON17 (j) FMoW (k) iWILDCAM (l) RxRx1

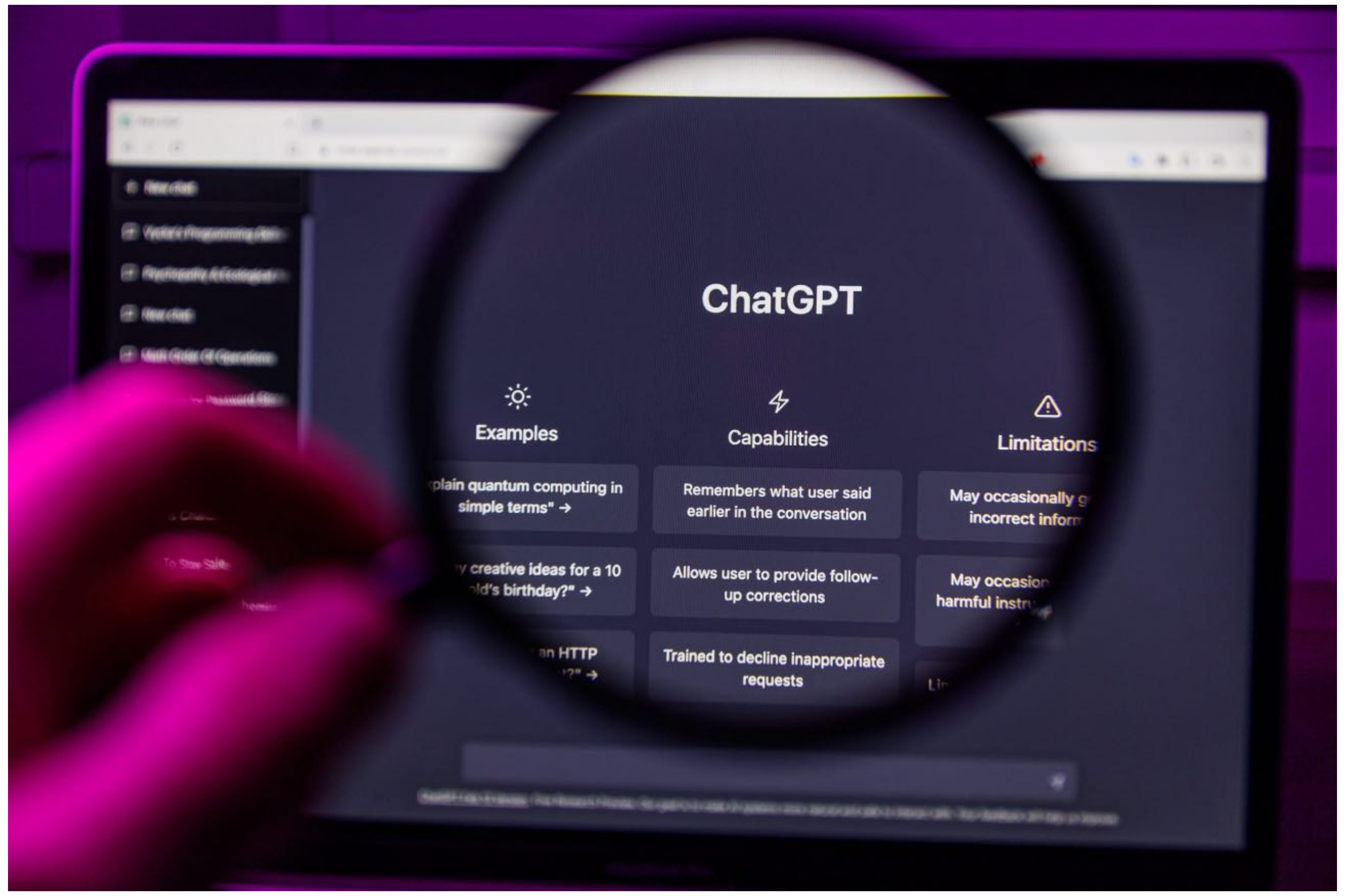
Learning Causality for Modern Machine Learning

Traditional ML assumes train and test data are **iid.**, i.e., independently sampled from an identical distribution, while data is often **OOD**, i.e., out-of-distribution, in real-world applications.

Objectives	**Causal Representation Learning on Graphs**: [NeurIPS'22 Spotlight, NeurIPS'23a]
Implications	**Useful Properties** of the Causal Representations: OOD Generalizability [NeurIPS'22, 23a], Adversarial Robustness [ICLR'22], Interpretability [ICML'24a]
Realizations	**Optimization & Feature Learning** schemes for Causal Representation Learning: [ICLR'23a, NeurIPS'23b]

From Traditional ML to Modern ML with Large Pretrained Models

The undergoing revolution to the traditional ML is the emerge of the **large pretrained models**.



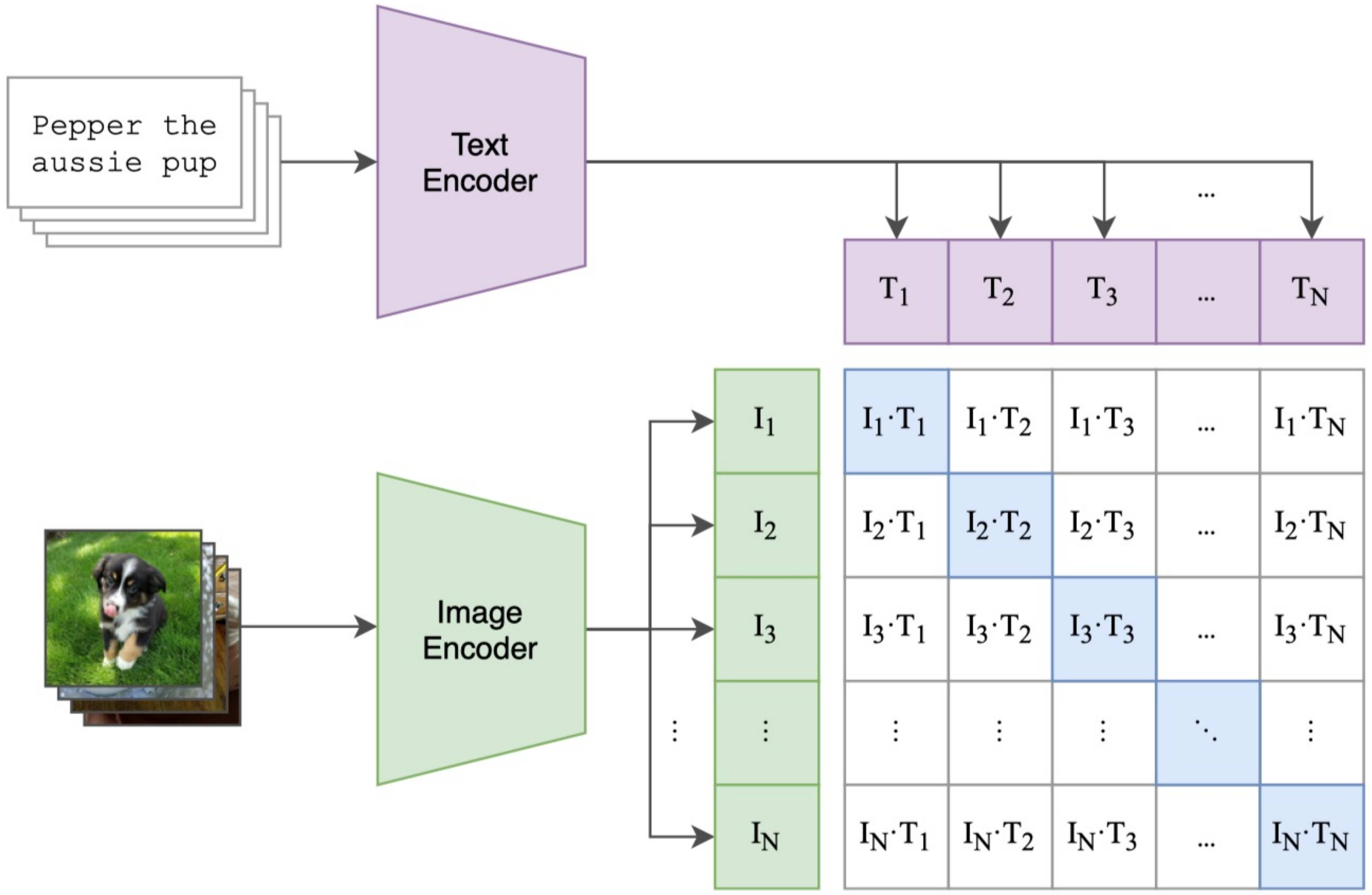
AlphaFold

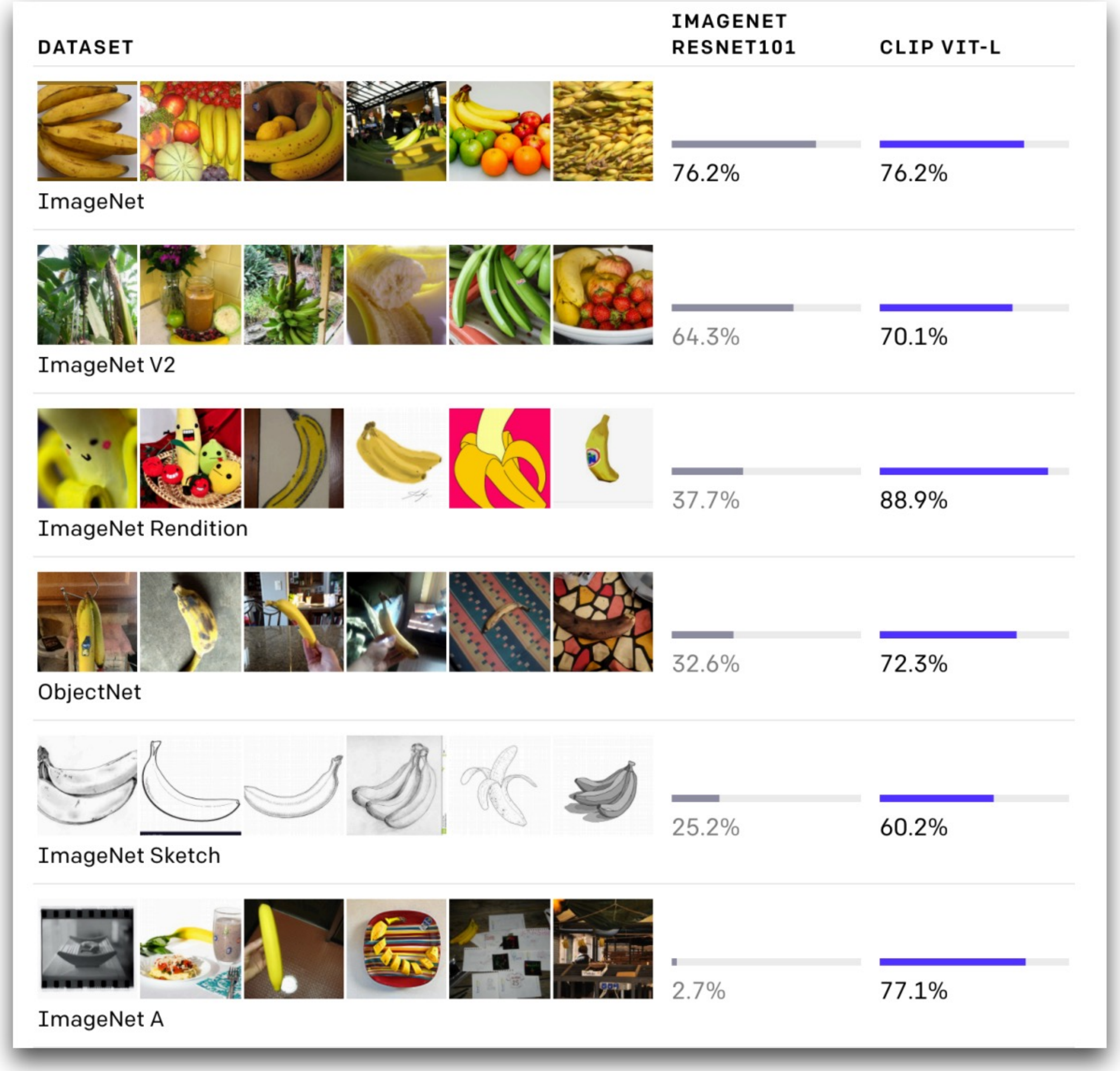
Stable Diffusion

ChatGPT

The Large Pretrained Models

Large pretrained models such as CLIP/ChatGPT presents a paradigm shift to modern ML systems.





Effective robustness

+6%

+51%

+40%

+35%

+74%

Web-scale training data: 400 million images collected from the web (dataset internal to OpenAI).

Multimodal contrastive learning: language supervision.

(Radford et al., 2021; OpenAI blog)

Is OOD Generalization Solved by Large Pretrained Models?

Large Pretrained Model can not solve the spurious correlation issue.

Ice Bear in Snow (common) CLIP ACCU: 80.25

Ice Bear in Grass (counter) CLIP ACCU: 9.17

Frequently misclassified as **Brown Bear** !

Chen , Wang*, Lin*, et al., "CLIPs Always Generalize Better than ImageNet Models?", arXiv 2403.11497*

Is OOD Generalization Solved by Large Pretrained Models?

We collect **55 classes** of animals with **7800 common examples** and **6500 counter examples**.

ImageNet label	Common			Counter			Decline
	background	# data	accuracy	background	# data	accuracy	
10 brambling, Fringilla montifringilla	green	117	78.63	white or blue	111	49.55	29.08
100 black swan, Cygnus atratus	above water	204	93.63	ground	106	68.87	24.76
102 echidna, spiny anteater, anteater	grass	125	20.00	tree	221	4.07	15.93
128 black stork, Ciconia nigra	grass	81	77.78	sky	149	14.77	63.01
130 flamingo	above water	197					
133 bittern	grass	205					
144 pelican	above water	232					
150 sea lion	sand	58					
16 bulbul	white or blue	122					
20 water ouzel, dipper	above water	260					
23 Vulture	sky	147	87.76	tree	98	41.84	45.92
275 African hunting dog, hyena dog, Cape hunting dog, Lycaon pictus	grass	210	85.24	tree	72	63.89	21.35
276 hyena, hyaena	grass	346	92.20	road	100	82.00	10.20
277 red fox, Vulpes vulpes	grass	143	69.23	road	105	59.05	10.18
279 Arctic fox, white fox, Alopex lagopus	snow	123	67.48	grass	238	26.05	41.43
290 jaguar, panther, Panthera onca, Felis onca	above water	65	33.85	tree	226	13.72	20.13
291 lion, king of beasts, Panthera leo	grass	263	74.90	tree	222	45.95	28.96
293 cheetah, chetah, Acinonyx jubatus	grass	212	80.66	tree	106	55.66	25.00
30 bullfrog, Rana catesbeiana	above water	274	73.36	not in water	158	48.10	25.26
33 loggerhead, loggerhead turtle, Caretta caretta	underwater	220	73.64	not in water	91	18.68	54.96
37 box turtle, box tortoise	grass	65	73.85	earth	200	49.00	24.85
39 common iguana, iguana, Iguana iguana	earth	50	54.00	shrub	120	30.83	23.17
41 whiptail, whiptail lizard	earth	249	60.64	hand	100	4.00	56.64
42 agama	rock	338	74.26	tree	142	28.87	45.39
49 African crocodile, Nile crocodile, Crocodylus niloticus	earth	91	72.53	grass	84	35.71	36.81
54 hognose snake, puff adder, sand viper	earth	203	22.17	grass	123	2.44	19.73
56 king snake, kingsnake	earth	228	30.26	grass	98	22.45	7.81
57 garter snake, grass snake	grass	78	67.95	earth	249	19.68	48.27
58 water snake	water	151	68.87	ground	163	1.23	67.65
70 harvestman, daddy longlegs, Phalangium opilio	shrub	501	48.50	rock	125	20.00	28.50
71 scorpion	indoor	79	29.11	outdoor	264	4.17	24.95
76 tarantula	sand	231	81.82	grass	158	43.67	38.15
79 centipede	white background	61					
80 black grouse	grass	52					
81 ptarmigan	snow	57					
83 prairie chicken, prairie grouse, prairie fowl	grass	259					
89 sulphur-crested cockatoo, Kakatoe galerita, Cacatua galerita	tree	163	88.34	grass	100	63.00	25.34
9 ostrich, Struthio camelus	ground	206	79.61	water	113	57.52	22.09
Balanced error		7866	66.57		6595	32.68	

276 hyena, hyaena	grass	346	92.20	road	100	82.00	10.20
277 red fox, Vulpes vulpes	grass	143	69.23	road	105	59.05	10.18
279 Arctic fox, white fox, Alopex lagopus	snow	123	67.48	grass	238	26.05	41.43
290 jaguar, panther, Panthera onca, Felis onca	above water	65	33.85	tree	226	13.72	20.13
291 lion, king of beasts, Panthera leo	grass	263	74.90	tree	222	45.95	28.96

9 ostrich, Struthio camelus	ground	206	79.61	water	113	57.52	22.09
Balanced error		7866	66.57		6595	32.68	33.89

- **Yes**! Concept shifts examples exist in LAION CLIP, leading to more than 30% drop in average accuracy.

Chen , Wang*, Lin*, et al., "CLIPs Always Generalize Better than ImageNet Models?", arXiv 2403.11497*

Is OOD Generalization Solved by Large Pretrained Models?

Symmetry is critical for reasoning tasks with LPMs, yet not sufficiently well learned.

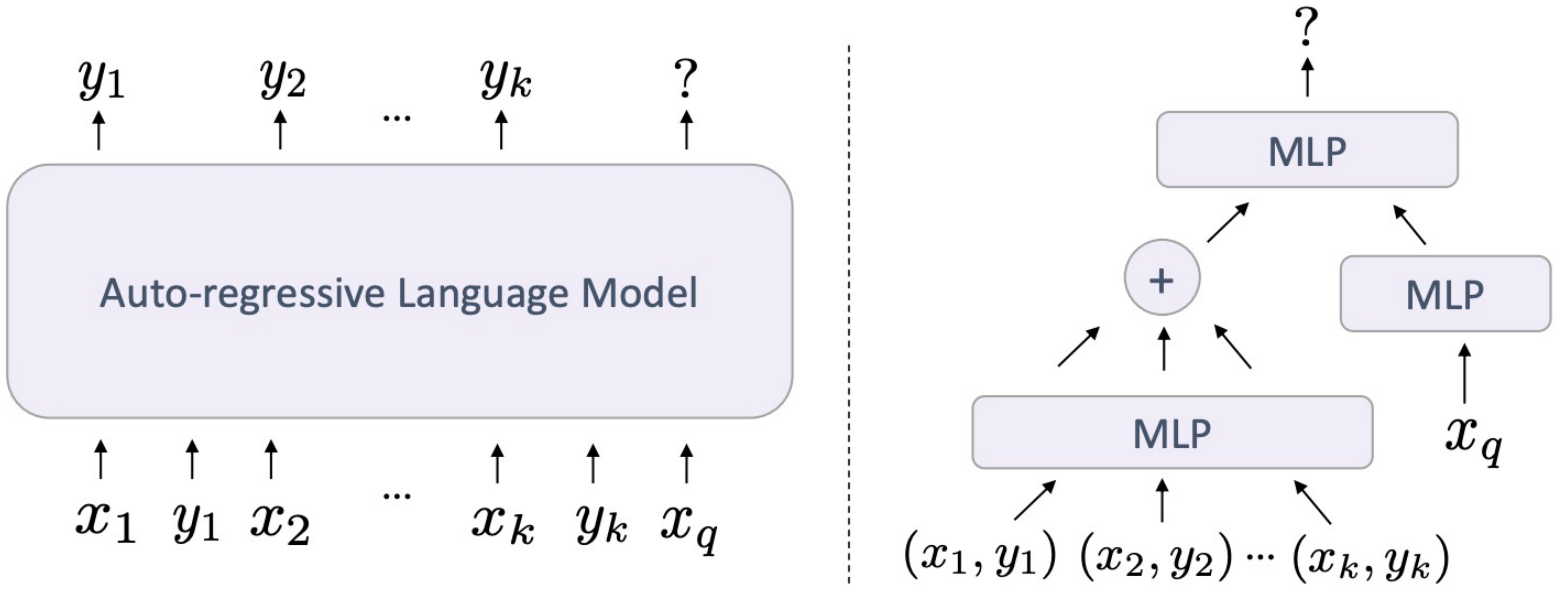
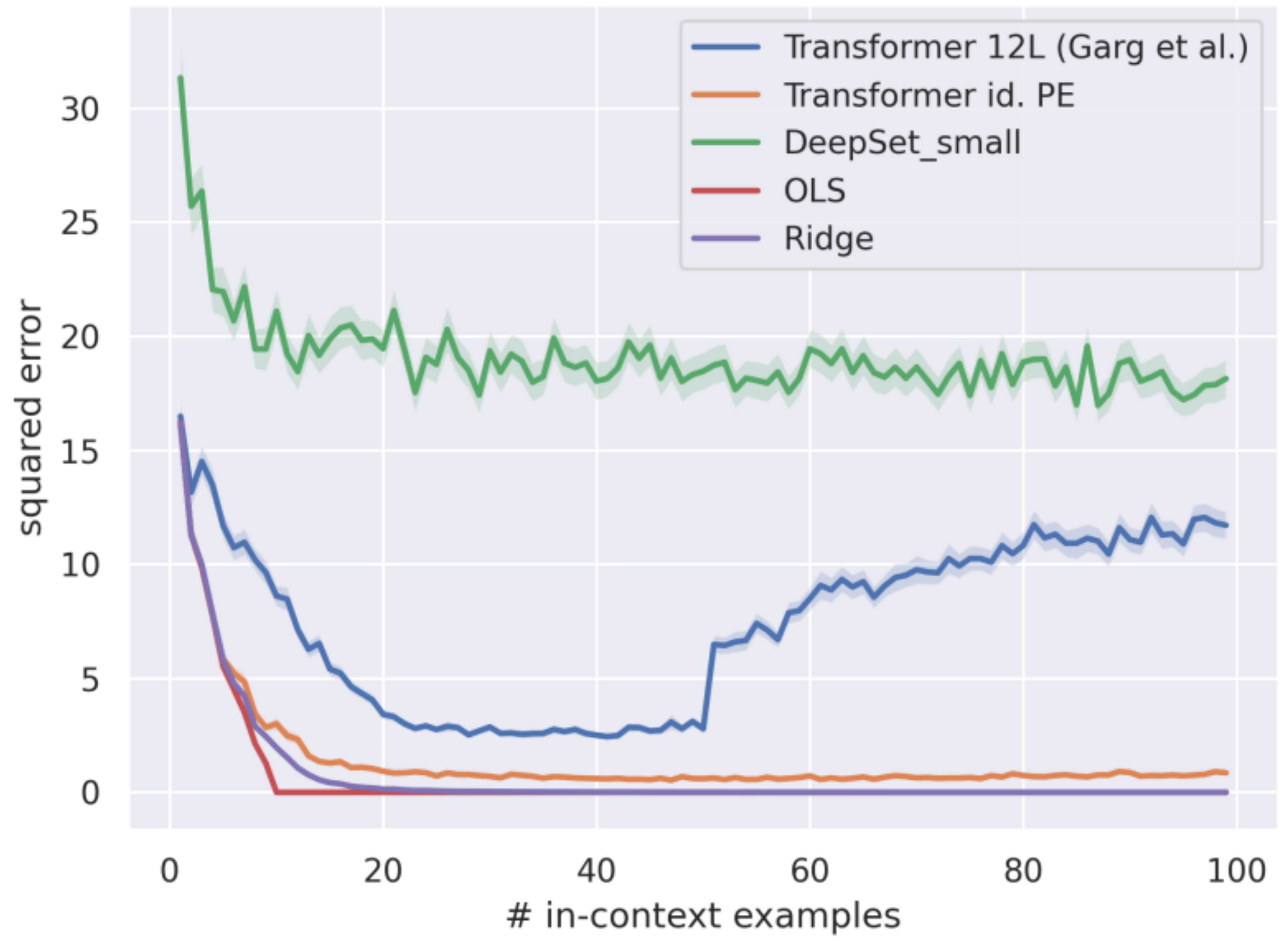


Figure 1. Illustration of linear regression ICL with auto-regressive Transformer (left) and DeepSet (right). Given k input demonstrations $(x_1, y_1), (x_2, y_2), \ldots, (x_k, y_k)$, and the query input x_k, Transformer adheres to the paradigm of the auto-regressive language model to infer the labels in an auto-regressive manner. In contrast, DeepSet jointly models the k sequential demonstrations as a set, and produces the output of the query based on the set-aggregated representations.

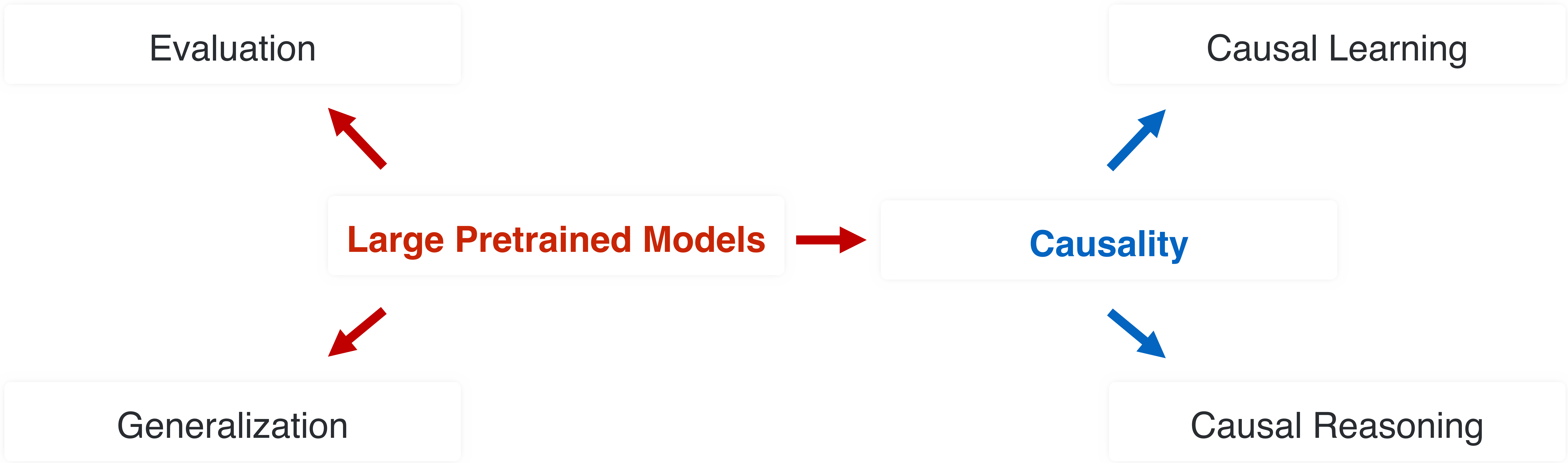


(b) OOD ICL with $\mu = 2$.

Chen et al., "Positional Information Matters for Invariant In-Context Learning: A Case Study of Simple Function Classes", arXiv 2311.18194

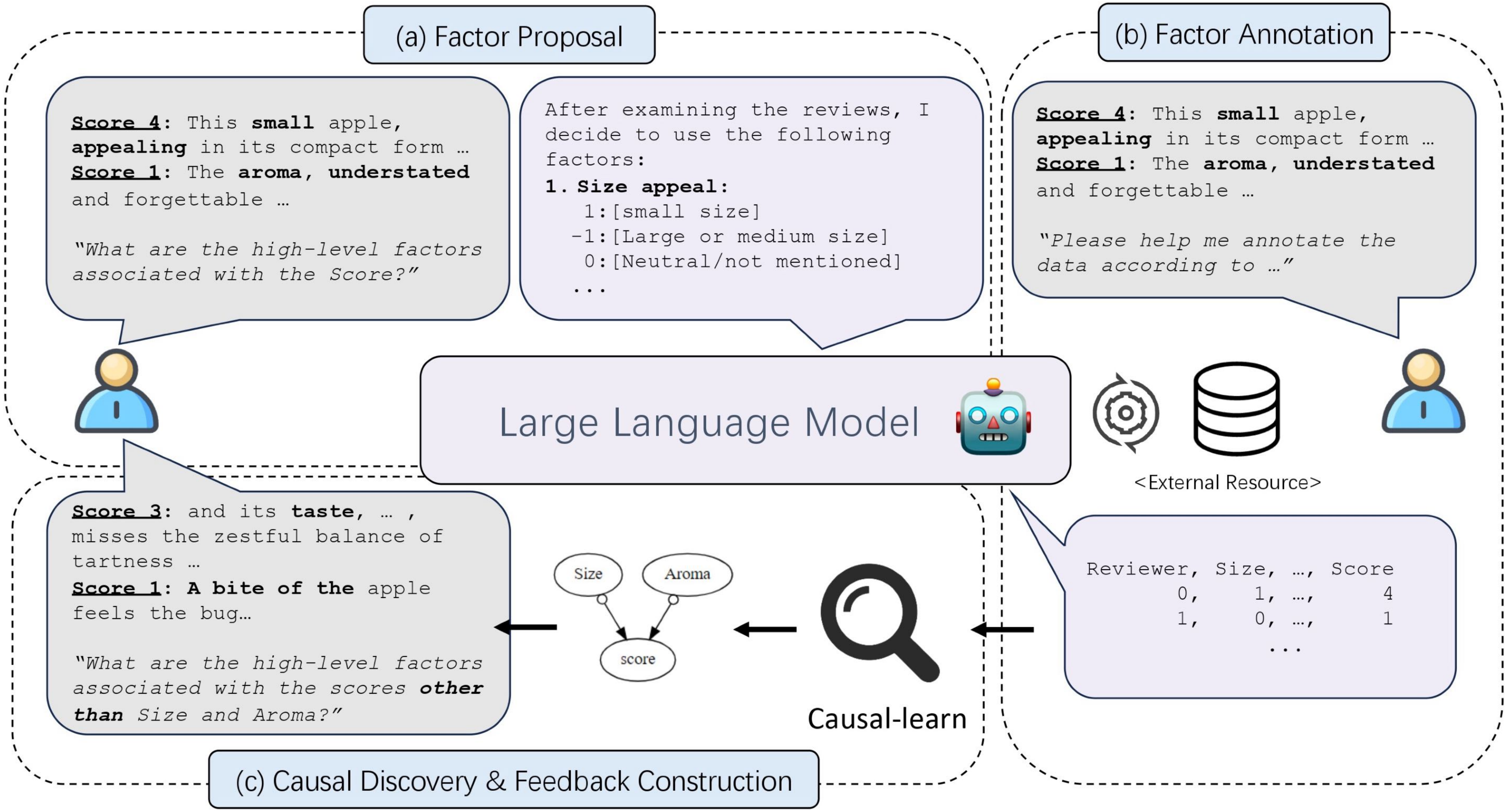
Combining the Best of Two Worlds

Large pretrained models provides new opportunities learning causality for modern ML :



Large Pretrained Model for Causal Representation Learning

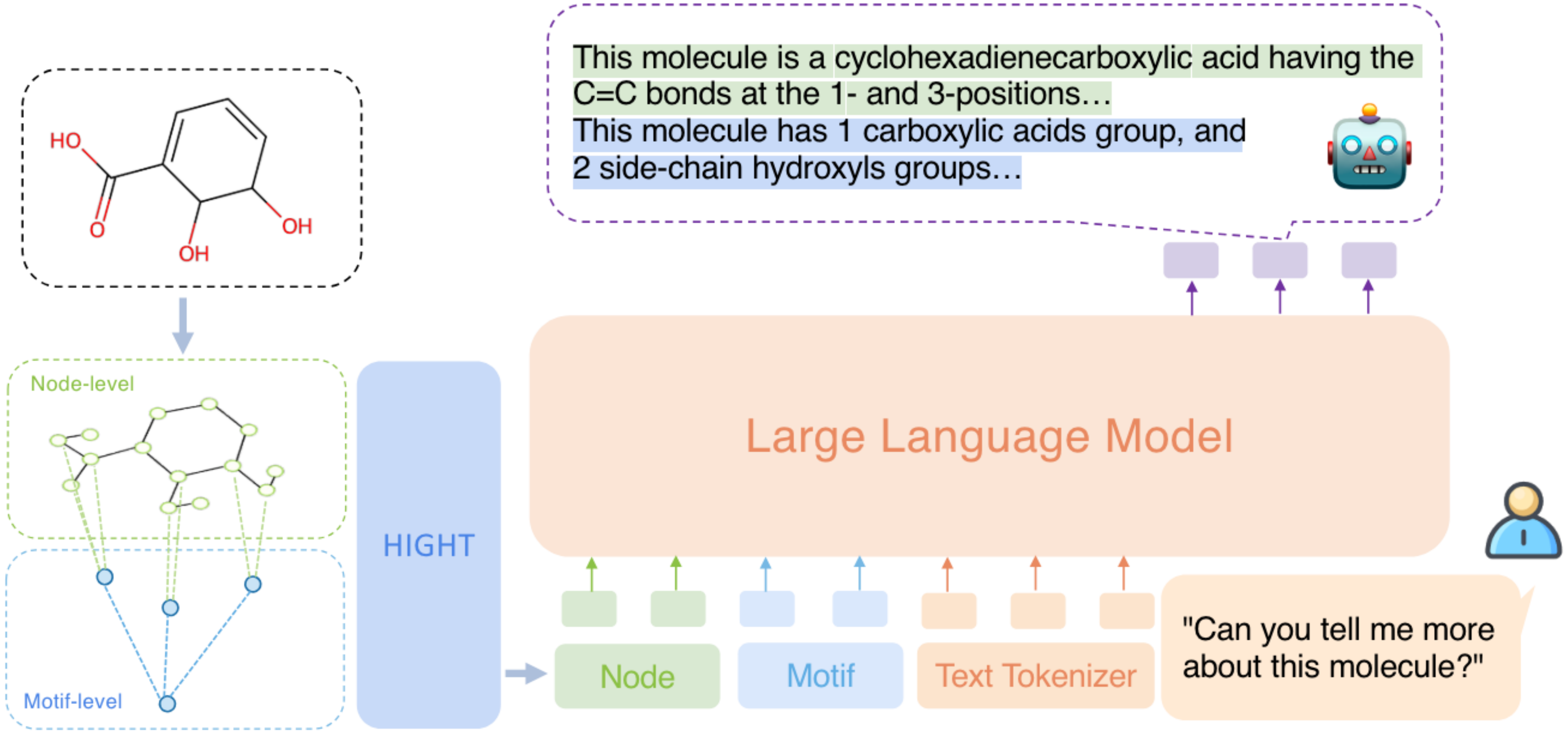
Large pretrained models can extracts **useful high-level hidden variables** for causal discovery using the **rich world knowledge**:



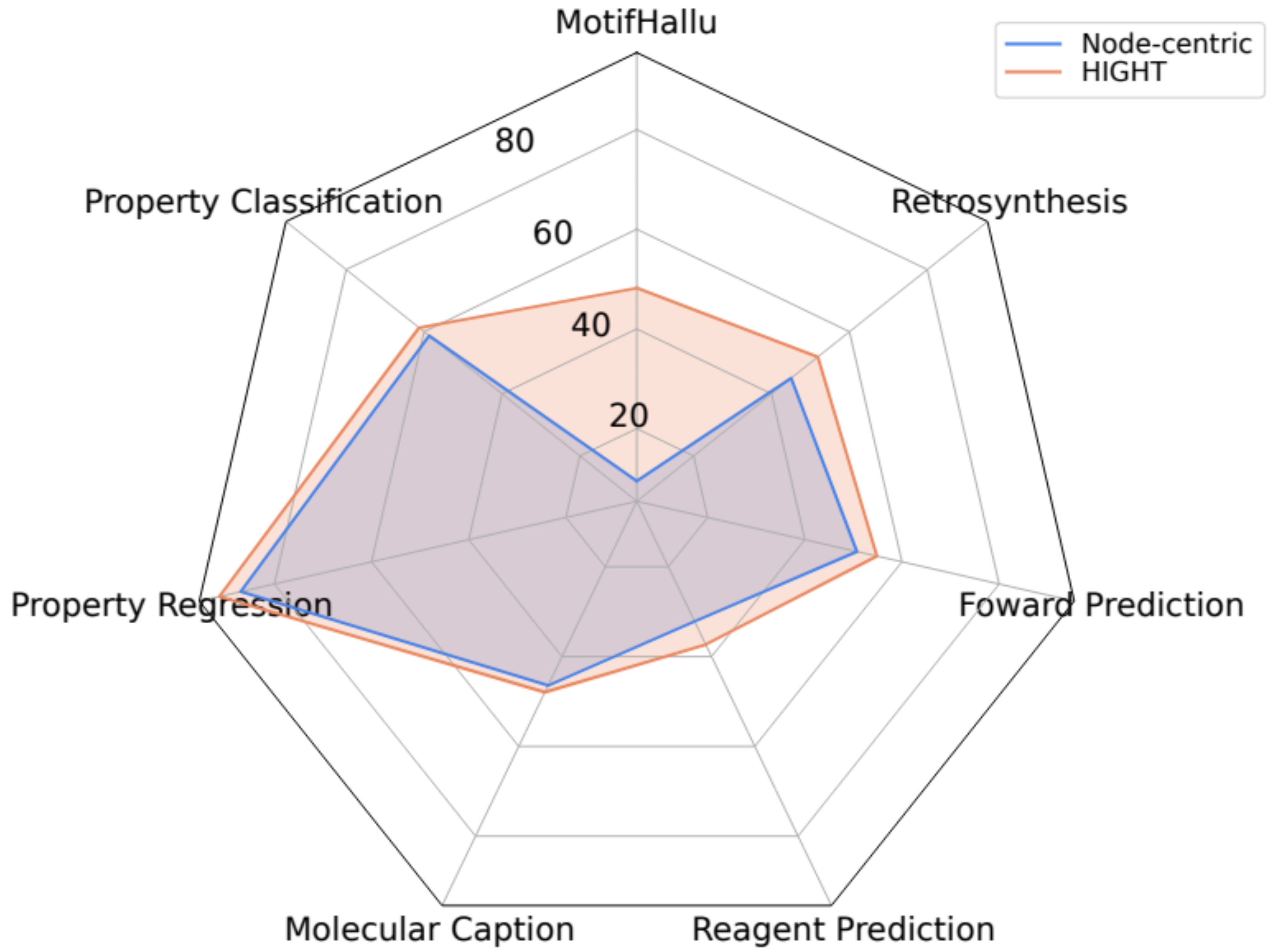
Chen , Liu*, et al., "Discovery of the Hidden World with Large Language Models", arXiv 2402.03941*

The Essential Role of Causality in Alignment

Aligning the rich knowledge to another modality or preferences requires proper causal disentanglement of the important concepts:



(a) Overview of the HIGHT framework.



(b) Summary of performance.

Chen et al., "Improving Graph-Language Alignment with Hierarchical Graph Tokenization", arXiv TBD

New Foundations of Modern Machine Learning

Combining large pretrained models and causality opens up a new frontier for modern machine learning.

New Understanding

New Methodology

Large Pre-trained Models

New Applications

Acknowledgements

The full acknowledgement list is given in the thesis.