Learning on Graphs under Open-World Assumptions

Qitian Wu（吴齐天）

Department of Computer Science and Engineering

Shanghai Jiao Tong University

General Learning Problems

- Standard Machine Learning Tasks:

Training Set

Data $\{(x_i, y_i)\}_{i \in \mathcal{I}_{tr}}$

Feature space $x_i \in \mathcal{X}_{tr}$

Label space $y_i \in \mathcal{Y}_{tr}$

Data distribution $(x_i, y_i) \sim p_{tr}(x, y)$

Test Set

Data $\{(x_i, y_i)\}_{i \in \mathcal{I}_{te}}$

Feature space $x_i \in \mathcal{X}_{te}$

Label space $y_i \in \mathcal{Y}_{te}$

Data distribution $(x_i, y_i) \sim p_{te}(x, y)$

$\hat{y}_i = f_\theta(x_i)$

training → Model → testing

the model actually minimize

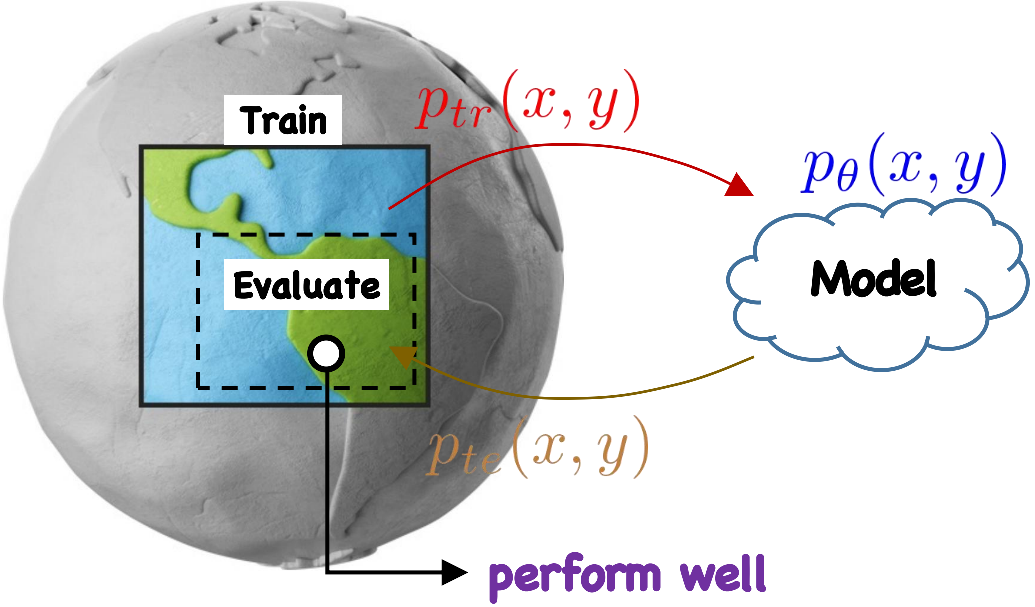
$\mathcal{D}(p_\theta(x, y), p_{tr}(x, y))$

the model is expected to minimize

$\mathcal{D}(p_\theta(x, y), p_{te}(x, y))$

- Two core ML concepts: representation and generalization

Learning under Closed-World Assumptions



model performance

$$\underbrace{\mathcal{D}(p_\theta(x,y), p_{te}(x,y))}_{\text{model performance}} \leq$$

$$\underbrace{\mathcal{D}_1(p_\theta(x,y), p_{tr}(x,y))}_{\text{fitting error}} + \underbrace{\mathcal{D}_2(p_{tr}(x,y), p_{te}(x,y))}_{\text{generalization gap}}$$

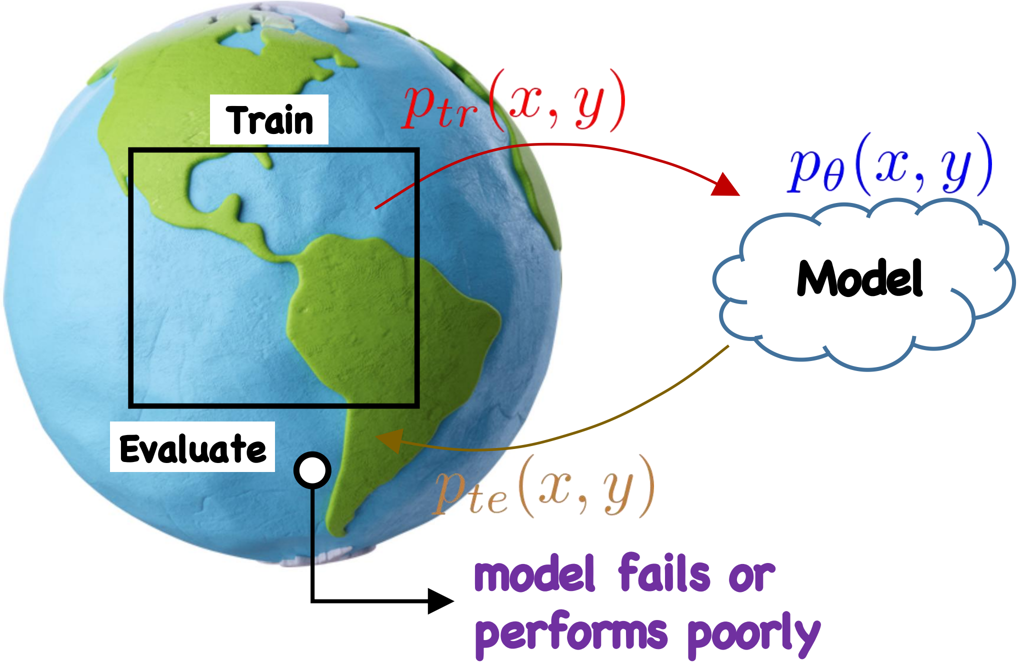
fitting error — *depend on model capacity*

generalization gap — *negligibly small*

Closed-world assumptions:

- Input/output space is shared by train and test data $\mathcal{X}_{te} \subseteq \mathcal{X}_{tr}, \mathcal{Y}_{te} \subseteq \mathcal{Y}_{tr}$
- Data distribution stays unchanged from train to test $p_{tr}(x,y) = p_{te}(x,y)$

Towards Open-World Learning



model performance

$$\mathcal{D}(p_\theta(x, y), p_{te}(x, y)) \leq$$
$$\mathcal{D}_1(p_\theta(x, y), p_{tr}(x, y)) + \mathcal{D}_2(p_{tr}(x, y), p_{te}(x, y))$$

fitting error — too small to be good

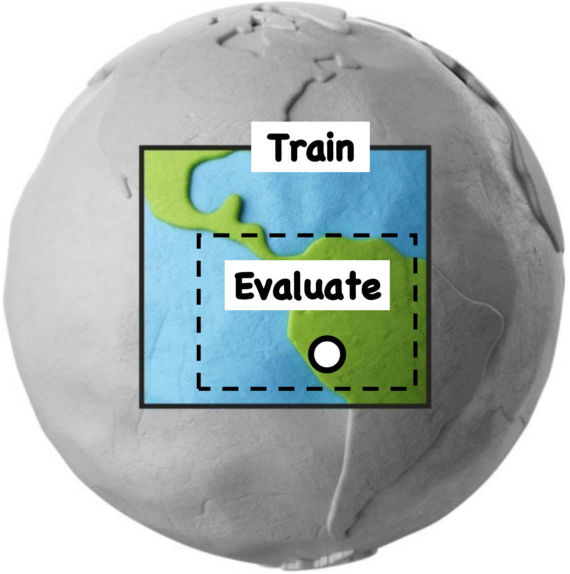
generalization gap — can be arbitrarily large

Open-world assumptions: from training to testing

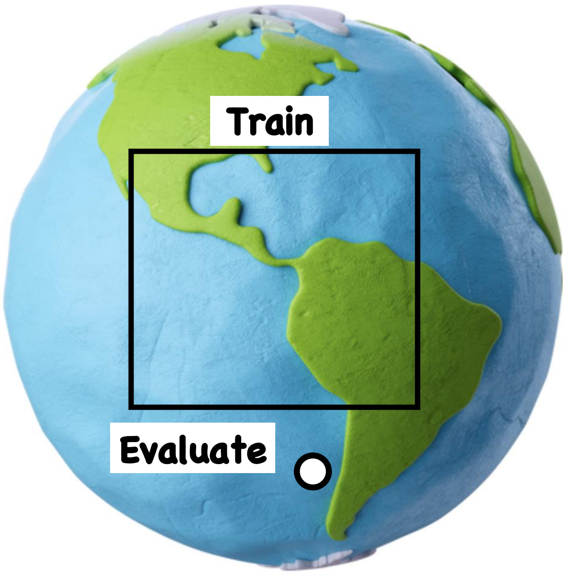
Input/output space goes through expansion $\mathcal{X}_{tr} \subset \mathcal{X}_{te}, \mathcal{Y}_{tr} \subset \mathcal{Y}_{te}$

Data distribution shifts with unknown factors $p_{tr}(x, y) \neq p_{te}(x, y)$

From Closed-World to Open-World Learning



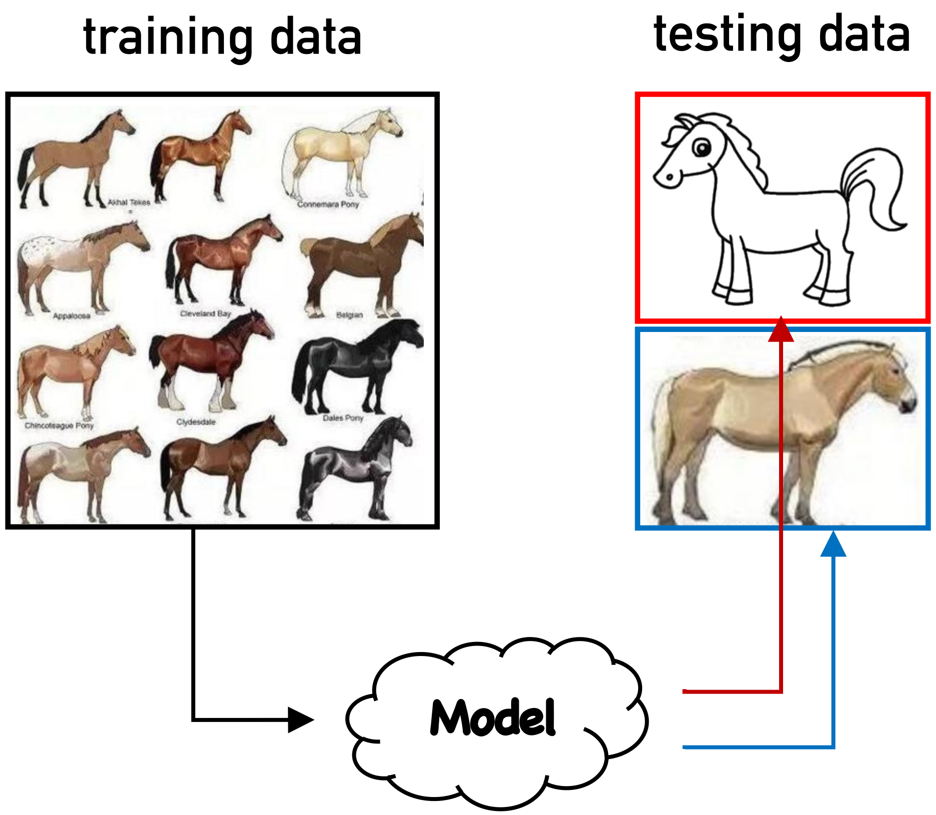
How to learn a desirably effective model under distribution shifts?



The challenging open research problems:

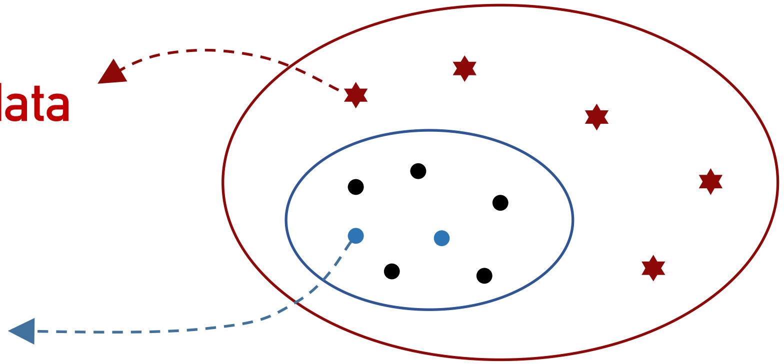
How to train a model that can generalize to OOD data? → OOD Generalization

Out-of-Distribution Generalization



out-of-distribution (OOD) data

in-distribution (IND) data



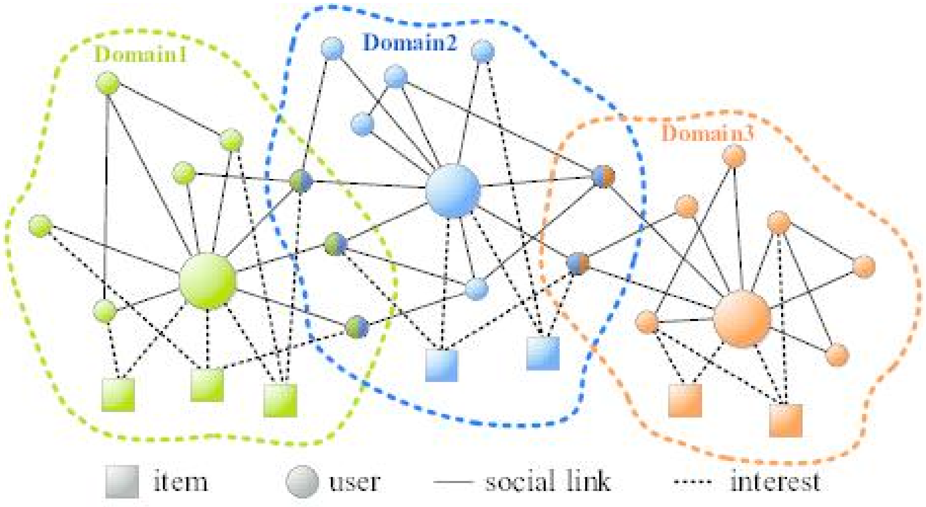
1: perform well on IND testing data

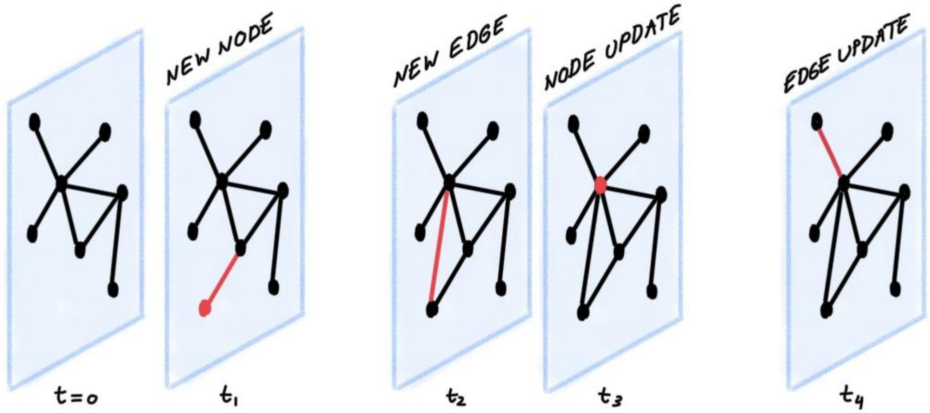
2: perform well on OOD testing data

OOD Generalization:

Train a robust classifier that can perform well on testing samples from disparate distributions than training data

Out-of-Distribution Data from Open World





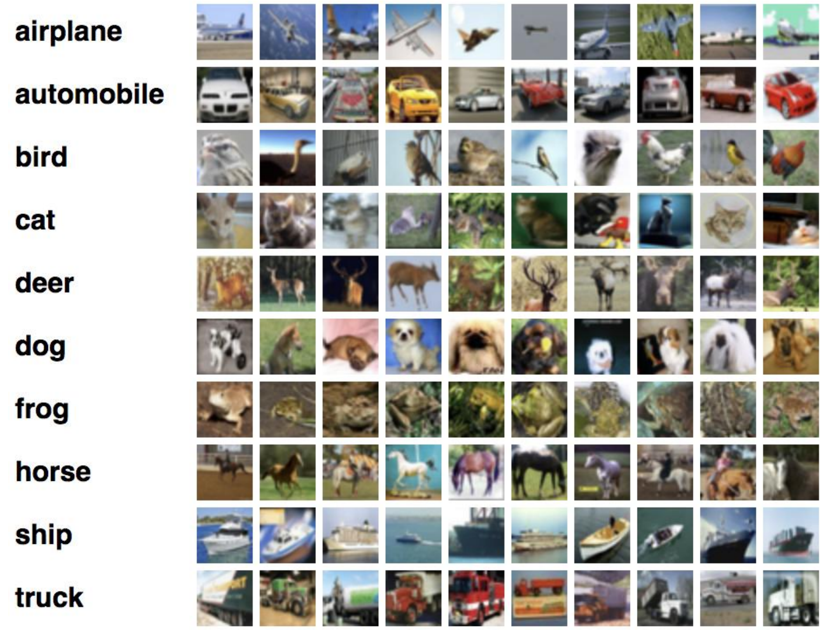
Graph data from multiple domains

Dynamic temporal networks

- Distribution shifts cause different data distributions $P_{train}(\mathcal{D}) \neq P_{test}(\mathcal{D})$
- New data from unknown distribution are unseen by training

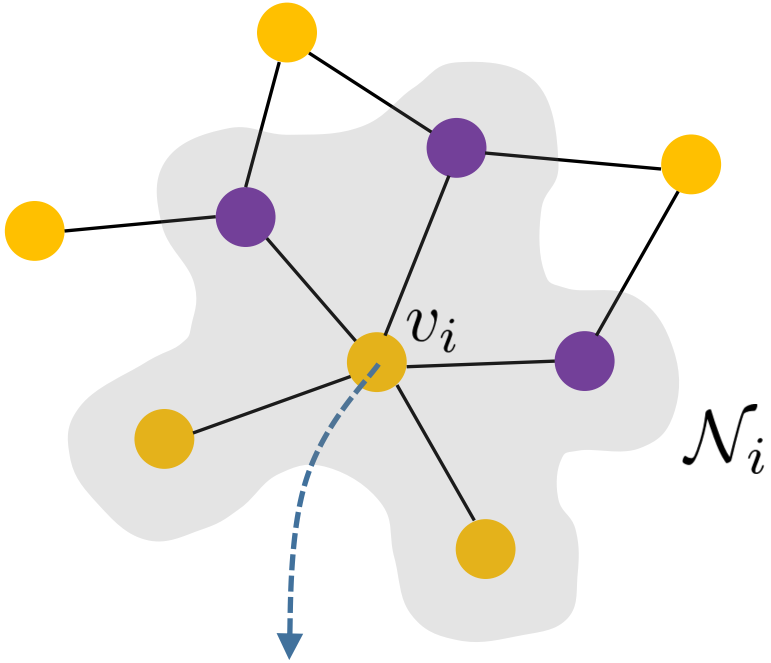
How to guarantee desired performance on data from new distributions?

Challenges of Graph Data Modeling



$$(x_i, y_i) \sim p(x, y)$$

each instance is drawed from the same data distribution independently (i.i.d.)



$$(x_i, y_i) \sim p(x, y | \mathcal{N}_i)$$

instances have inter-connection and cannot be treated as i.i.d. samples

Node-Level Distribution Shifts

- **Graph notation:** A graph $G = (A, X)$, adjacency matrix $A = \{a_{uv} | v, u \in V\}$ node features $X = \{x_v | v \in V\}$, node labels $Y = \{y_v | v \in V\}$

$$p(\mathbf{G}, \mathbf{Y} | \mathbf{e}) = p(\mathbf{G} | \mathbf{e}) p(\mathbf{Y} | \mathbf{G}, \mathbf{e})$$

  where e denotes environment (that affects data generation)

- **Out-of-distribution generalization on graphs:**

$$\min_f \max_{e \in \mathcal{E}} \mathbb{E}_{G \sim p(\mathbf{G}|\mathbf{e}=e)} \left[\frac{1}{|V|} \sum_{v \in V} \mathbb{E}_{y \sim p(\mathbf{y}|\mathbf{G_v}=G_v, \mathbf{e}=e)} [l(f(G_v), y)] \right]$$

learn a classifier robust for worst case ($\min_f \max_{e \in \mathcal{E}}$)

loss function for node-level prediction ($[l(f(G_v), y)]$)

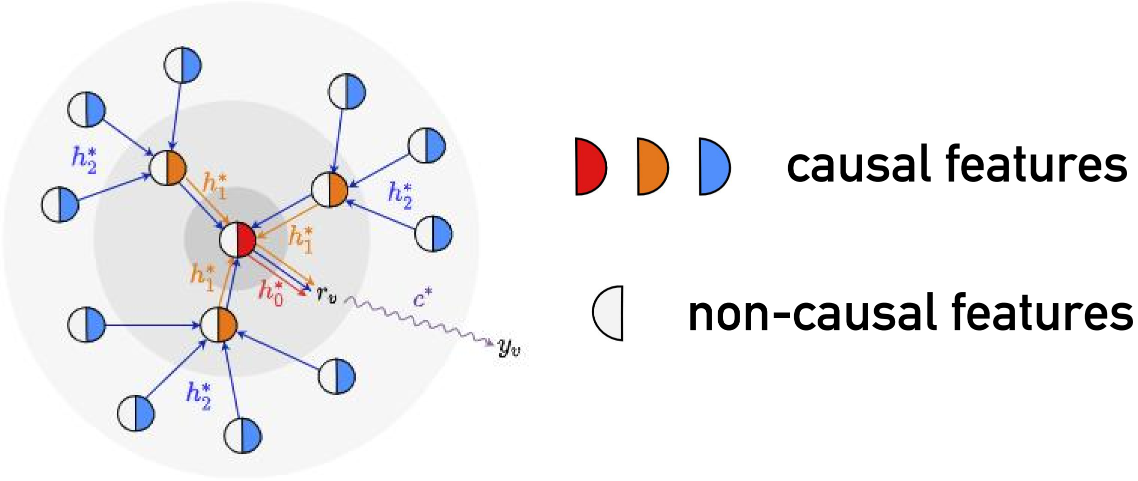
  - A graph G can be divided into pieces of ego-graphs $\{(G_v, y_v)\}_{v \in V}$
  - The data generation process: 1) the entire graph is generated via $G \sim p(\mathbf{G}|\mathbf{e})$, 2) each node's label is generated via $y \sim p(\mathbf{y} | \mathbf{G_v} = G_v, \mathbf{e})$

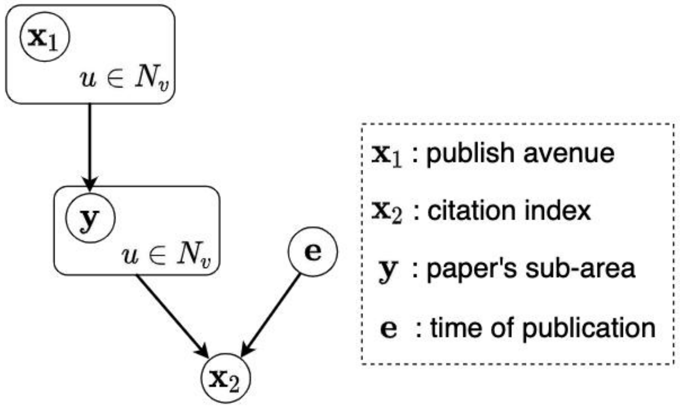
Qitian Wu, et al., "Handling Distribution Shifts on Graphs: An Invariance Perspective", in ICLR'22

Causal Invariance Principle

There exists a portion of causal information within input ego-graph for prediction task of each individual node

The "causal" means two-fold properties:
1) invariant across environments
2) sufficient for prediction





node features $x_v = [x_v^1, x_v^2]$

predictive model $\hat{y}_v = \frac{1}{|N_v|} \sum_{u \in N_v} \theta_1 x_u^1 + \theta_2 x_u^2$

(causal features: x_u^1; non-causal features: x_u^2)

ideal solutions $[\theta_1, \theta_2] = [1, 0]$

Arjovsky, et al., "Invariant Risk Minimization".

Rojas-Carulla, et al., "Invariant models for causal transfer learning".

Explore-to-Extrapolate Risk Minimization

- **Initial version:** jointly minimize the expectation and variance of risks

$$\min_{\theta} \mathbb{V}_{\mathbf{e}}[L(G^e, Y^e; \theta)] + \beta \mathbb{E}_{\mathbf{e}}[L(G^e, Y^e; \theta)]$$

Key issue: environment/domain labels for data are unavailable or ambiguous

- **Final version:** adversarial training multiple context generators

Risk Extrapolation →

$$\min_{\theta} \mathrm{Var}(\{L(g_{w_k^*}(G), Y; \theta) : 1 \le k \le K\}) + \frac{\beta}{K}\sum_{k=1}^{K} L(g_{w_k^*}(G), Y; \theta)$$

Environment Exploration →

$$\text{s. t. } [w_1^*, \cdots, w_K^*] = \arg\max_{w_1, \cdots, w_K} \mathrm{Var}(\{L(g_{w_k}(G), Y; \theta) : 1 \le k \le K\})$$

where

$$L(g_{w_k}(G), Y; \theta) = L(G^k, Y; \theta) = \frac{1}{|V|}\sum_{v \in V} l(f_\theta(G_v^k), y_v).$$

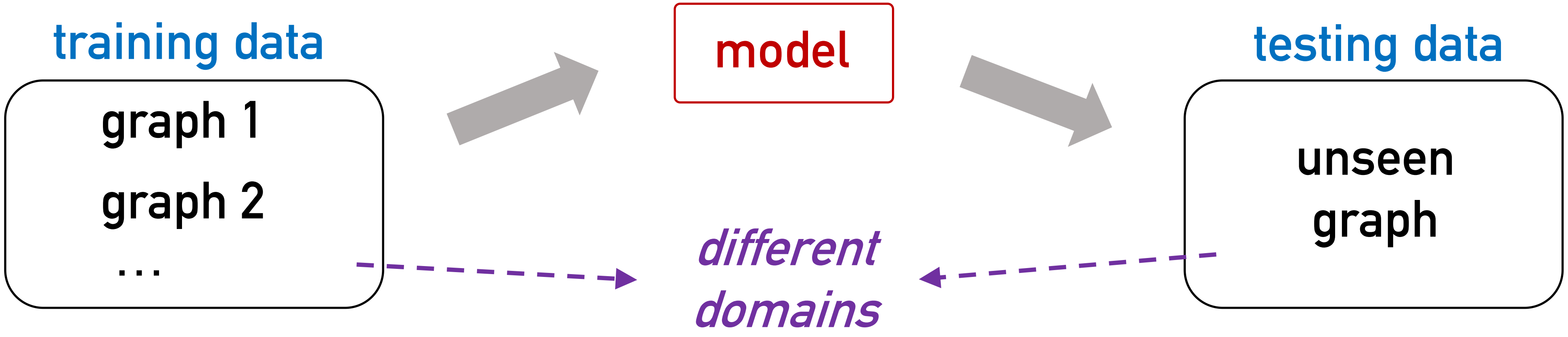
context generator: augment training data and simulate multiple environments

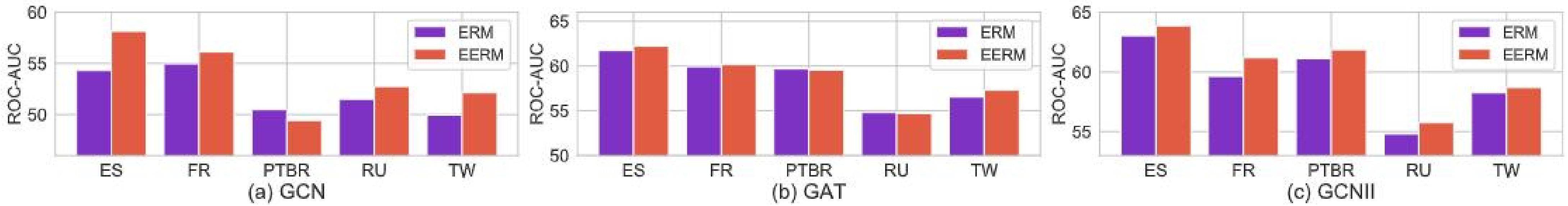
risk function for data under the k-th environment

predictor: graph neural networks for classification

Qitian Wu, et al., “Handling Distribution Shifts on Graphs: An Invariance Perspective”, in ICLR'22

Experiment on Cross-Graph Transfer

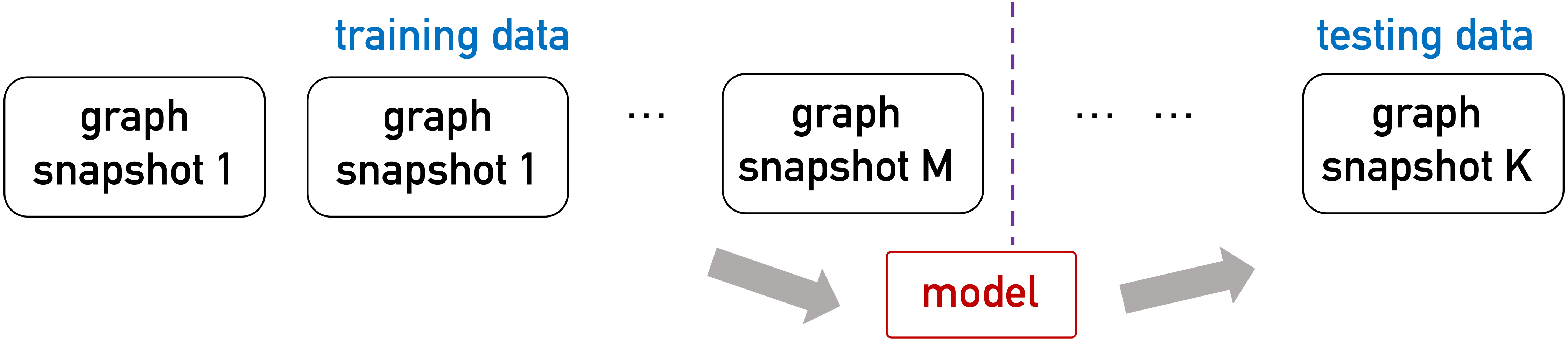


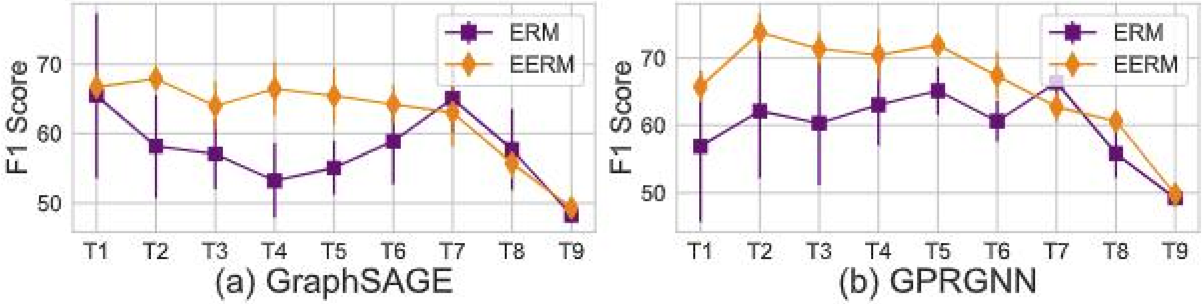


EERM achieves up to 7.0% (resp. 7.2%) impv. on ROC-AUC (resp. accuracy) than ERM

Qitian Wu, et al., “Handling Distribution Shifts on Graphs: An Invariance Perspective”, in ICLR'22

Experiment on Temporal Graph Evoluation





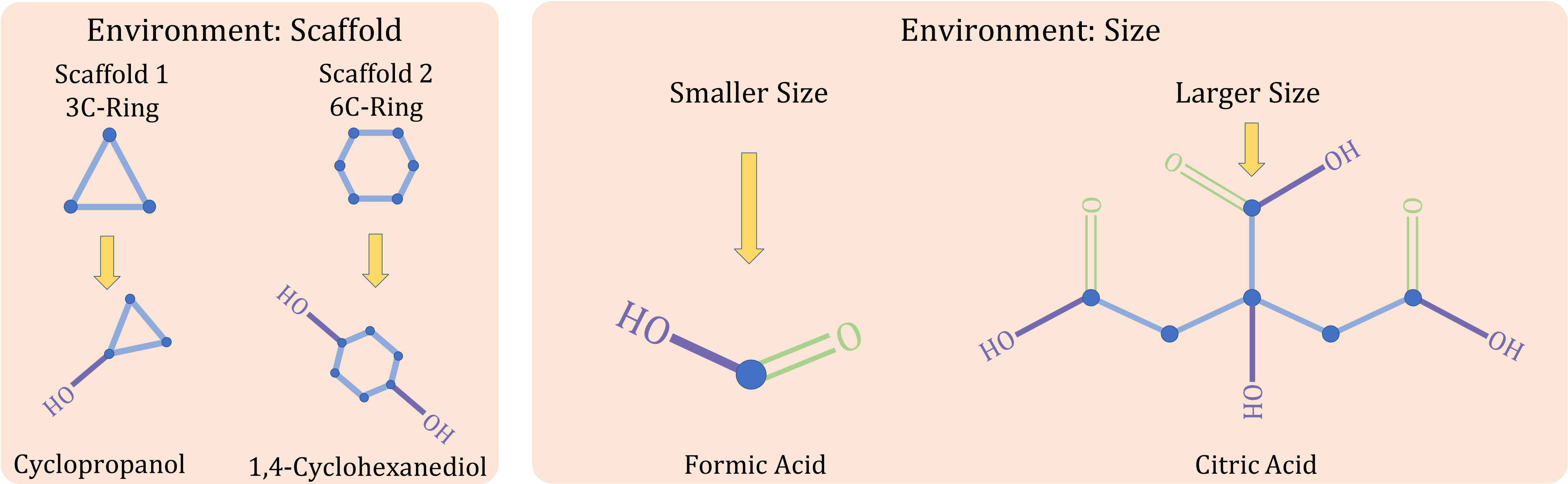
Method	2014-2016	2016-2018	2018-2020
ERM- SAGE	42.09 ± 1.39	39.92 ± 2.53	36.72 ± 2.47
EERM- SAGE	41.55 ± 0.68	40.36 ± 1.29	38.95 ± 1.57
ERM- GPR	47.25 ± 0.55	45.07 ± 0.57	41.53 ± 0.56
EERM- GPR	49.88 ± 0.49	48.59 ± 0.52	44.88 ± 0.62

EERM achieves up to 9.6%/10.0% impv using GraphSAGE/GPR-GNN as backbones

Qitian Wu, et al., “Handling Distribution Shifts on Graphs: An Invariance Perspective”, in ICLR'22

Graph-Level Distribution Shifts - Molecules

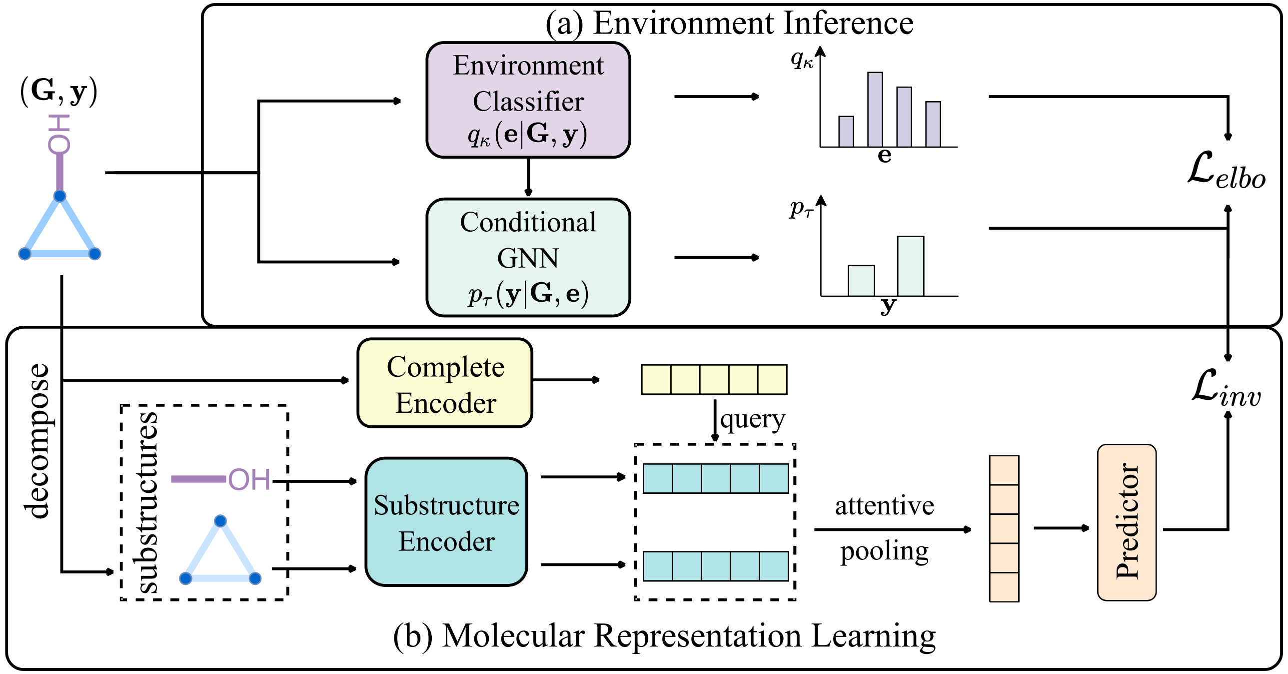
Key observation: the (bio)chemical properties of a molecule are usually associated with a few privileged molecular substructures



the shared hydroxy (-OH)/ carboxy (-COOH) → good water solubility

Nianzu Yang, et al., “Learning Substructure Invariance for Out-of-Distribution Molecular Representations”, in NeurIPS’22

MoleOOD: Learning Substructure Invariance

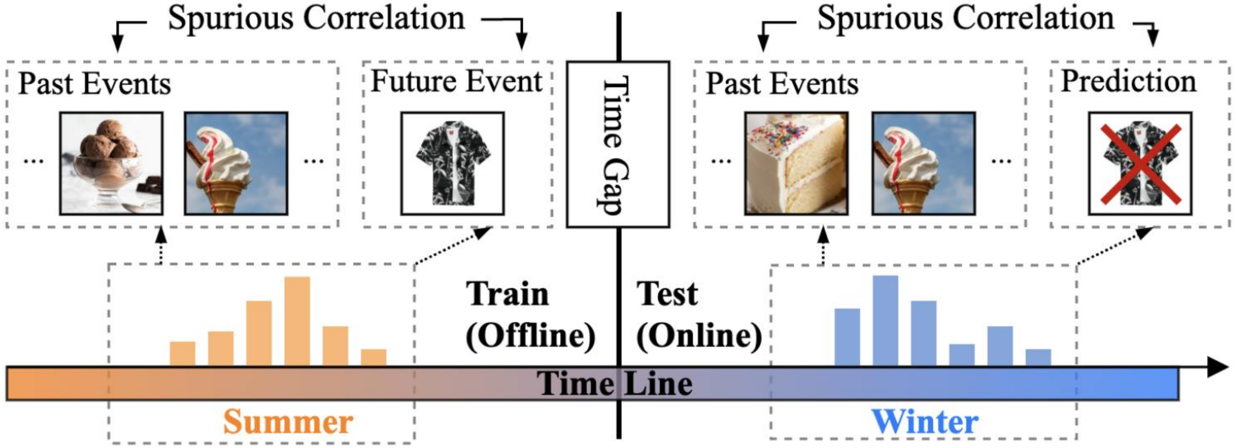


- **two-stage training strategy to search for optimal parameters**
  - 1) optimizing the environment-inference model: $\kappa^*, \tau^* \leftarrow \arg\max_{\kappa,\tau} \mathcal{L}_{elbo}(\tau, \kappa; \mathcal{G}^{train})$
  - 2) optimizing the molecule encoder and the predictor: $\theta^* \leftarrow \arg\min_{\theta} \mathcal{L}_{inv}(\theta; \mathcal{G}^{train}, \tau)$

Nianzu Yang, et al., “Learning Substructure Invariance for Out-of-Distribution Molecular Representations”, in NeurIPS’22

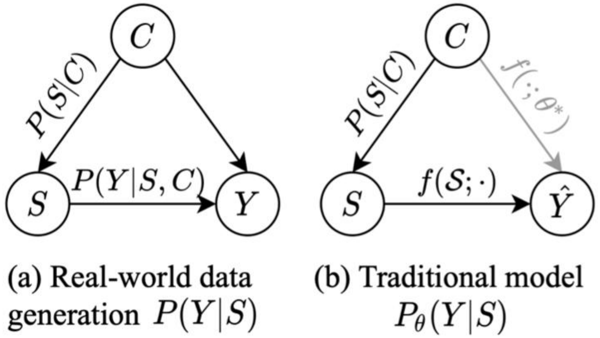
Distribution Shifts in Sequential Prediction

- Traditional models :



Learning *spurious correlation* between past events (S) and next event (Y)

- Explanation :



- $S \rightarrow \hat{Y}$: from model formulation $\hat{y} = f(\mathcal{S}; \theta)$
- $C \rightarrow \hat{Y}$: from learning process

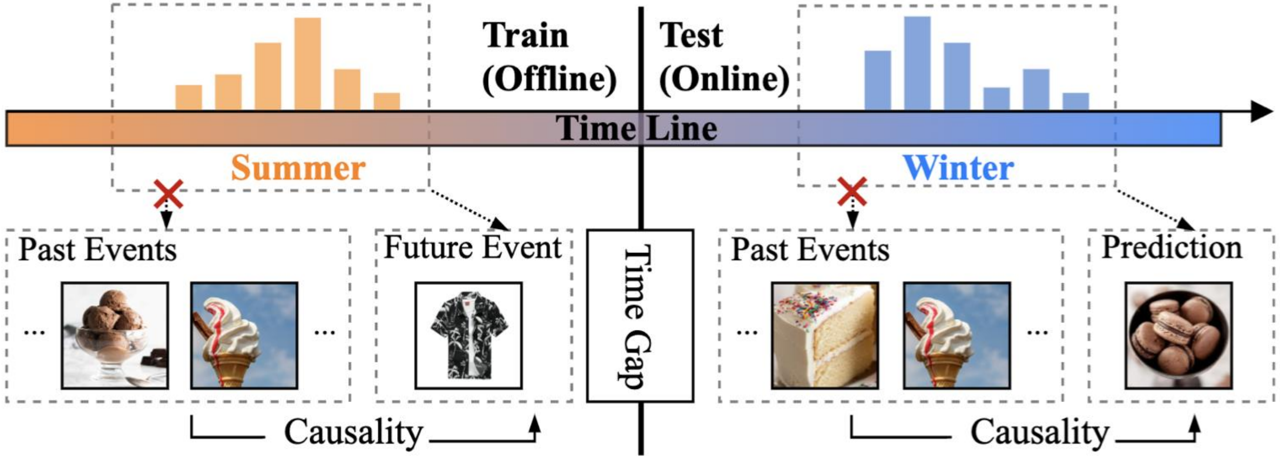
$$\theta^* = \arg\min_{\theta} \mathbb{E}_{(\mathcal{S},y)\sim P(S,Y|C=c_{tr})}[l\left(f(\mathcal{S};\theta), y\right)]$$

C is the confounder !

Chenxiao Yang, et al., "Towards out-of-distribution sequential event prediction: A causal treatment", in NeurIPS'22

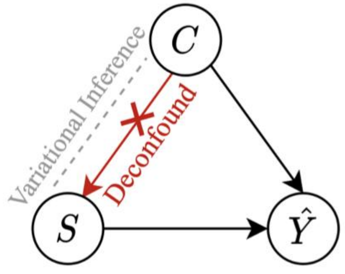
Causal Intervention for Sequential Prediction

- Proposed interventional models :



Learning *causality* between past events (S) and next event (Y)

- Solution :



(c) Our interventional model $P_\theta(Y|do(S))$

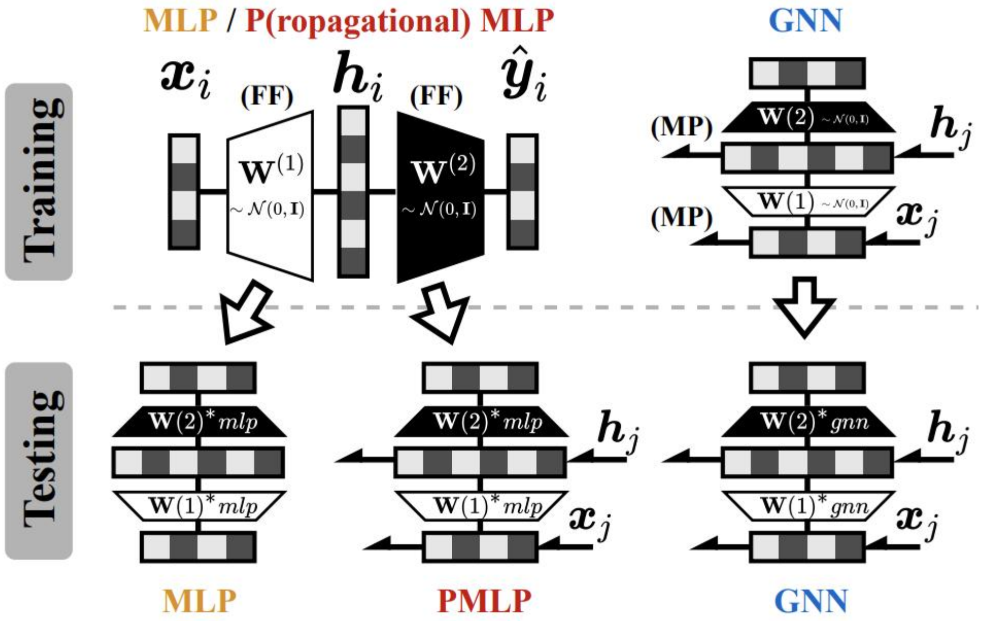
(Objective with do-operation)

$$P_\theta(Y\,|\,S) \rightarrow P_\theta(Y\,|\,do(S))$$

: simulates an *ideal data-generating process* where S is generated independently from C by blocking the backdoor path $S \leftarrow C \rightarrow \hat{Y}$

Chenxiao Yang, et al., "Towards out-of-distribution sequential event prediction: A causal treatment", in NeurIPS'22

Inherent Generalization of GNNs



Key question: Why GNNs are more powerful than MLP?

PMLP: Propagational MLP

- PMLP=MLP during training
- PMLP=GNN during testing

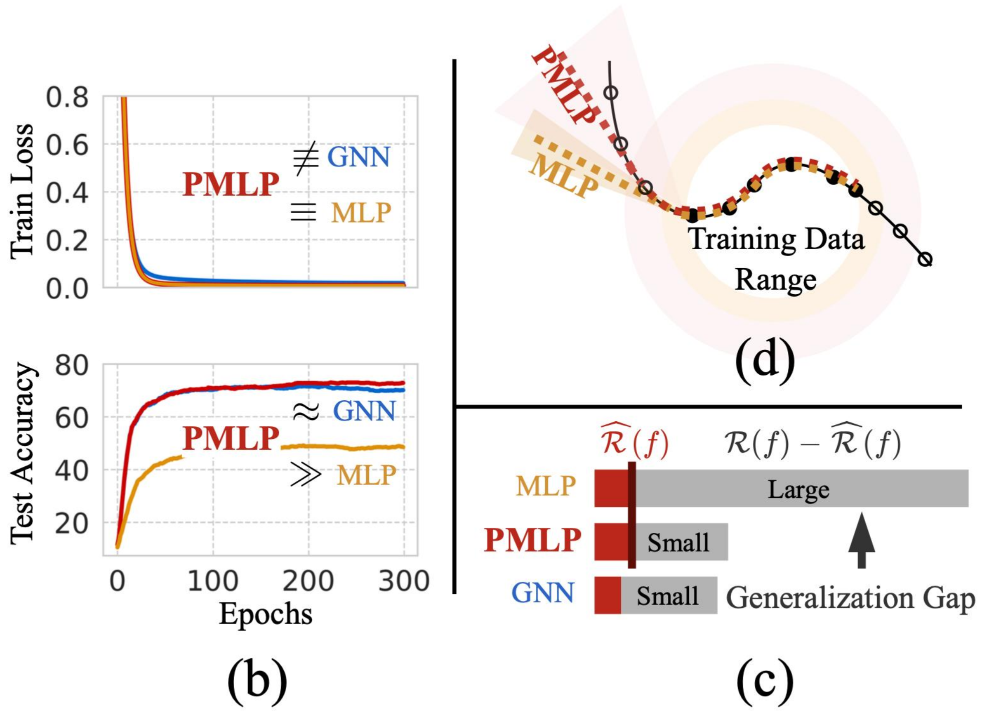
❑ Consistent phenomenons across *sixteen* benchmarks:

- PMLP significantly outperforms MLP
- PMLP performs close to GNN

The superiority of GNNs over MLP comes from better test-time generalization

Chenxiao Yang, et al., "Graph Neural Networks are Inherently Good Generalizers: Insights by Bridging GNNs and MLPs", in ICLR'23

Theoretical Understandings of GNNs



❑ By NTK theory we prove:

Compared to MLP, GNNs have better extrapolation ability, i.e., generalizing to OOD data outside training support

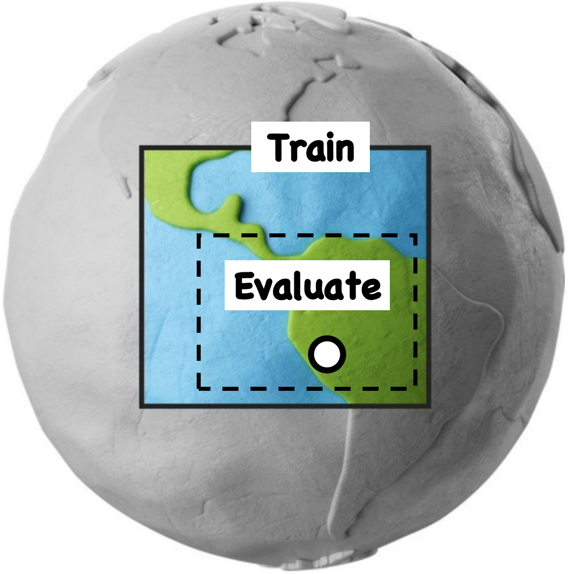
Theorem 5. *Suppose all node features are normalized, and the cosine similarity of node $\boldsymbol{x}_i$ and the average of its neighbors is deonoted as $\alpha_i \in [0, 1]$. Then, the convergence rate for $f_{pmlp}(\boldsymbol{x})$ is*

$$\left| \frac{\left(f_{pmlp}(\boldsymbol{x}_0 + \Delta t \boldsymbol{v}) - f_{pmlp}(\boldsymbol{x}_0)\right) / \Delta t}{c_{\boldsymbol{v}} \sum_{i \in \mathcal{N}_0 \cup \{0\}} (\tilde{d} \cdot \tilde{d}_i)^{-1}} - 1 \right| = O\left(\frac{1 + (\tilde{d}_{max} - 1)\sqrt{1 - \alpha_{min}^2}}{t} \right). \quad (10)$$

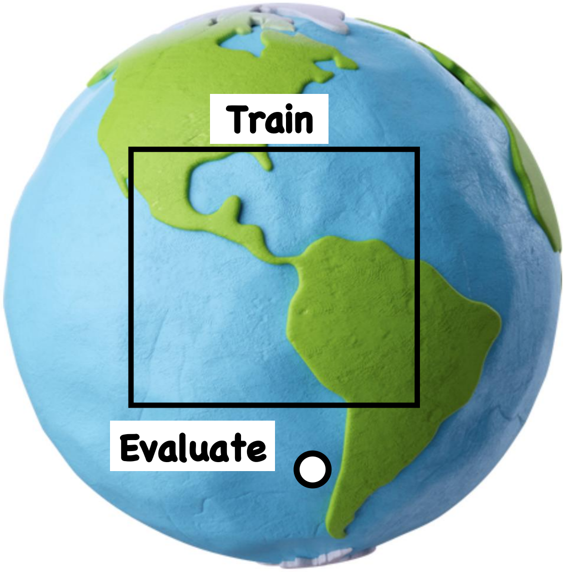
where $\alpha_{min} = \min\{\alpha_i\}_{i \in \mathcal{N}_0 \cup \{0\}} \in [0, 1]$, and $\tilde{d}_{max} \geq 1$ denotes the maximum node degree in the testing node $\boldsymbol{x}_0$'s neighbors (including itself).

Chenxiao Yang, et al., “Graph Neural Networks are Inherently Good Generalizers: Insights by Bridging GNNs and MLPs”, in ICLR'23

From Closed-World to Open-World Learning



How to learn a desirably effective model under distribution shifts?



The challenging open research problems:

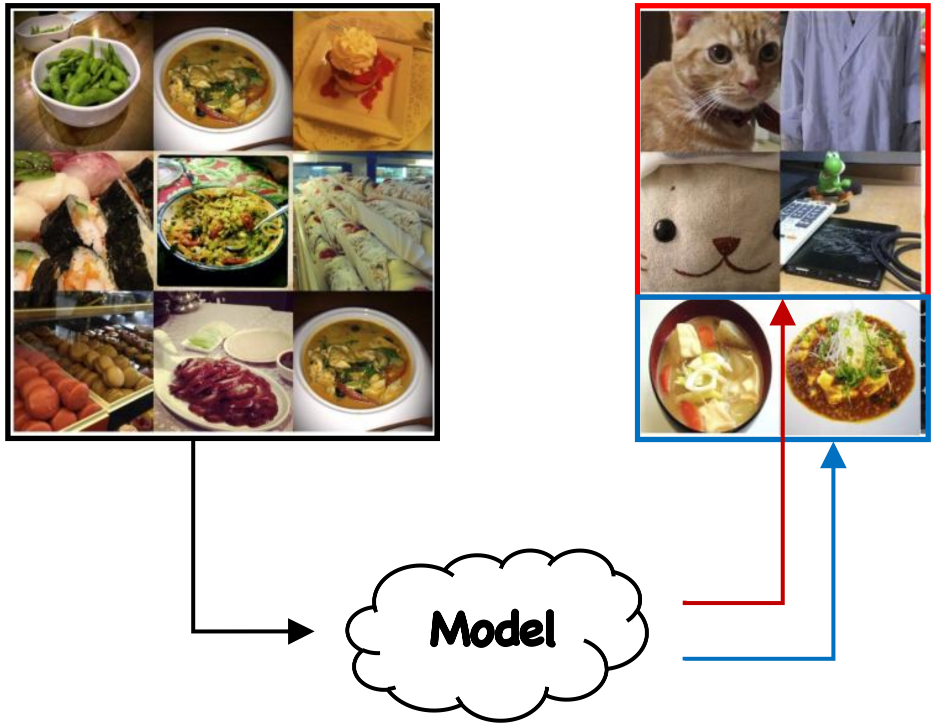
How to train a model that can generalize to OOD data? → OOD Generalization

How to train a model that can identify OOD data? → OOD Detection

Out-of-Distribution Detection

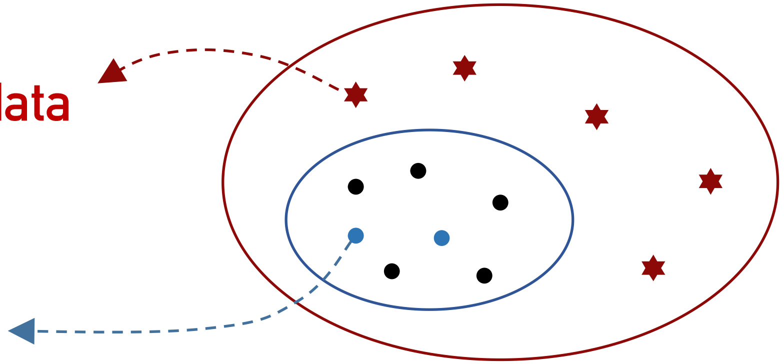
training data

testing data



out-of-distribution (OOD) data

in-distribution (IND) data



1: perform well on IND testing data

2: identify OOD testing data

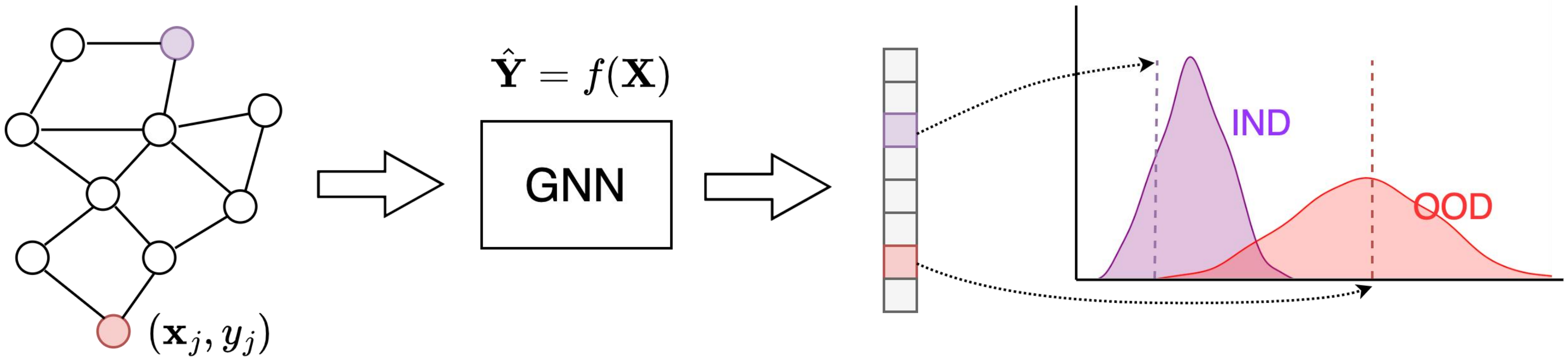
OOD Detection:

Train a robust classifier that can identify samples from disparate distributions than (in-distribution) training data

OOD Detection for Graph Data

- For a classifier f , our goal is to find a proper decision function that returns the estimation score whether the given input is OOD or not:

$$G(\mathbf{x}, \mathcal{G}_{\mathbf{x}}; f) = \begin{cases} 1, & \mathbf{x} \text{ is an in-distribution instance,} \\ 0, & \mathbf{x} \text{ is an out-of-distribution instance,} \end{cases}$$



GNN-based Node-Level Prediction

- Adopt graph neural networks (GNNs) to compute node representations:

$$Z^{(l)} = \sigma\left(D^{-1/2}\tilde{A}D^{-1/2}Z^{(l-1)}W^{(l)}\right), \quad Z^{(l-1)} = [\mathbf{z}_i^{(l-1)}]_{i\in\mathcal{I}}, \quad Z^{(0)} = X$$

- The GNN classifier gives a predictive distribution for node labels:

$$p(y \mid \mathbf{x}, \mathcal{G}_\mathbf{x}) = \frac{e^{h_\theta(\mathbf{x},\mathcal{G}_\mathbf{x})_{[y]}}}{\sum_{c=1}^{C} e^{h_\theta(\mathbf{x},\mathcal{G}_\mathbf{x})_{[c]}}} \qquad \text{where} \quad \mathbf{z}_i^{(L)} = h_\theta(\mathbf{x}, \mathcal{G}_\mathbf{x})$$

- If we assume $E(\mathbf{x}, \mathcal{G}_\mathbf{x}, y; h_\theta) = -h_\theta(\mathbf{x}, \mathcal{G}_\mathbf{x})_{[y]}$ as an energy function, we have

$$p(y|\mathbf{x}, \mathcal{G}_\mathbf{x}) = \frac{e^{-E(\mathbf{x},\mathcal{G}_\mathbf{x},y)}}{\sum_{y'} e^{-E(\mathbf{x},\mathcal{G}_\mathbf{x},y')}} = \frac{e^{-E(\mathbf{x},\mathcal{G}_\mathbf{x},y)}}{e^{-E(\mathbf{x},\mathcal{G}_\mathbf{x})}}$$

a Boltzmann distribution

$$E(\mathbf{x}, \mathcal{G}_\mathbf{x}; h_\theta) = -\log \sum_{c=1}^{C} e^{h_\theta(\mathbf{x},\mathcal{G}_\mathbf{x})_{[c]}}$$

free energy for OOD detection

Qitian Wu, et al., "Energy-based Out-of-Distribution Detection for Graph Neural Networks", in ICLR'23

Energy Models for OOD Detection

- For a given GNN classifier $h_\theta(\mathbf{x}, \mathcal{G}_\mathbf{x})$, we have the initial energy as

$$\mathbf{E}^{(0)} = [E(\mathbf{x}_i, \mathcal{G}_{\mathbf{x}_i}; h_\theta)]_{i \in \mathcal{I}} \quad \text{where } E(\mathbf{x}, \mathcal{G}_\mathbf{x}; h_\theta) = -\log \sum_{c=1}^{C} e^{h_\theta(\mathbf{x}, \mathcal{G}_\mathbf{x})_{[c]}}$$

- Then we consider propagating the energy values along graph structures

$$\mathbf{E}^{(k)} = \alpha \mathbf{E}^{(k-1)} + (1-\alpha) D^{-1} A \mathbf{E}^{(k-1)} \quad \text{where } \mathbf{E}^{(k)} = [E_i^{(k)}]_{i \in \mathcal{I}}$$

Intuition: connected nodes in the graph tend to be sampled from similar distributions

Proposition 1 (informal)

The energy propagation facilitates *consensus* for the OOD estimation results between the target node and its neighboring nodes.

Qitian Wu, et al., “Energy-based Out-of-Distribution Detection for Graph Neural Networks”, in ICLR'23

Loss Functions for Training

- If the training data only contains in-distribution data, use supervised loss:

$$\mathcal{L}_{sup} = \sum_{i \in \mathcal{I}_s} \left(-h_\theta(\mathbf{x}_i, \mathcal{G}_{\mathbf{x}_i})_{[y_i]} + \log \sum_{c=1}^{C} e^{h_\theta(\mathbf{x}_i, \mathcal{G}_{\mathbf{x}_i})_{[c]}} \right)$$

GNN-Safe

- If the training data contains extra OOD data, we additionally consider the regularization loss: $\mathcal{L}_{sup} + \lambda \mathcal{L}_{reg}$

GNN-Safe++

$$\mathcal{L}_{ref} = \frac{1}{|\mathcal{I}_s|} \sum_{i \in \mathcal{I}_s} \left(\mathrm{ReLU} \left(\tilde{E}\left(\mathbf{x}_i, \mathcal{G}_{\mathbf{x}_i}; h_\theta\right) - t_{in} \right) \right)^2 + \frac{1}{|\mathcal{I}_o|} \sum_{j \in \mathcal{I}_o} \left(\mathrm{ReLU} \left(t_{out} - \tilde{E}\left(\mathbf{x}_j, \mathcal{G}_{\mathbf{x}_j}; h_\theta\right) \right) \right)^2$$

extra OOD training data

Proposition 2 (informal)

The optimal predicted logits given by $\mathcal{L}_{sup}$ is the same as the counterpart of optimal energy by $\mathcal{L}_{reg}$.

Qitian Wu, et al., “Energy-based Out-of-Distribution Detection for Graph Neural Networks”, in ICLR'23

Main Results on Real-World Datasets

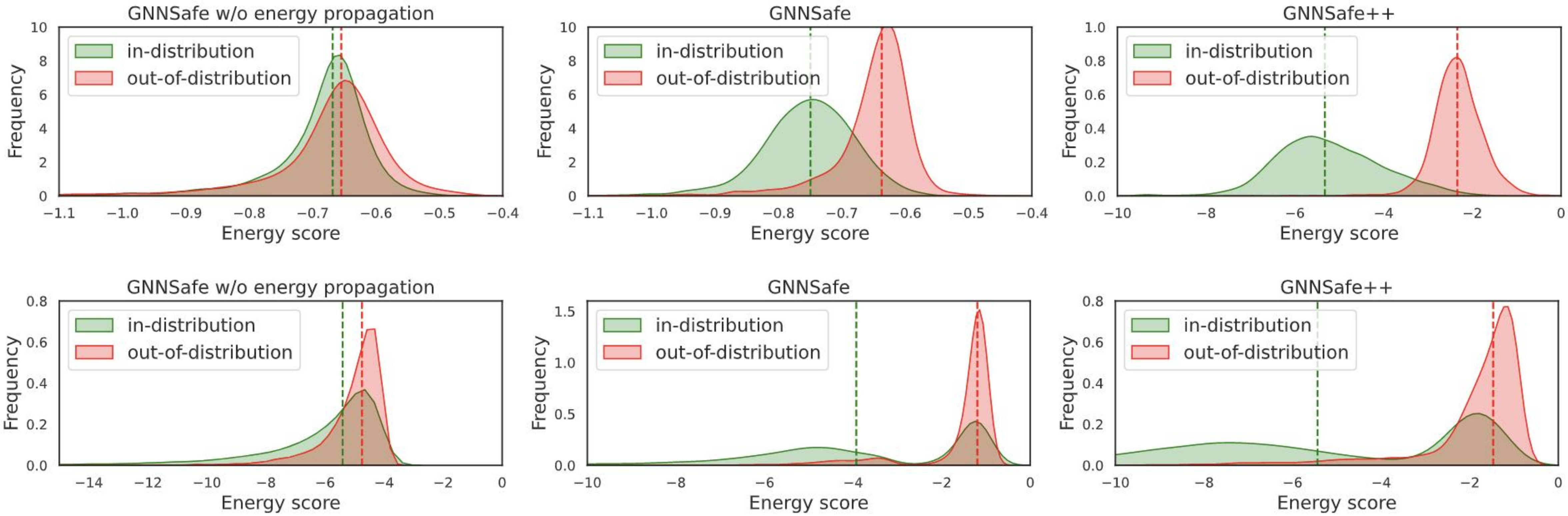
OOD detection results on Twitch and Arxiv

Model	OOD Expo	Twitch AUROC	Twitch AUPR	Twitch FPR	Twitch ID ACC	Arxiv AUROC	Arxiv AUPR	Arxiv FPR	Arxiv ID ACC
MSP	No	33.59	49.14	97.45	68.72	63.91	**75.85**	**90.59**	53.78
ODIN	No	**58.16**	72.12	93.96	70.79	55.07	68.85	100.0	51.39
Mahalanobis	No	55.68	66.42	**90.13**	70.51	56.92	69.63	94.24	51.59
Energy	No	51.24	60.81	91.61	70.40	**64.20**	75.78	90.80	53.36
GKDE	No	46.48	62.11	95.62	67.44	58.32	72.62	93.84	50.76
GPN	No	51.73	66.36	95.51	68.09	-	-	-	-
GNNSAFE	No	66.82	**70.97**	76.24	70.40	71.06	80.44	87.01	53.39
OE	Yes	55.72	70.18	95.07	70.73	69.80	80.15	85.16	52.39
Energy FT	Yes	**84.50**	**88.04**	**61.29**	70.52	**71.56**	**80.47**	**80.59**	53.26
GNNSAFE++	Yes	95.36	97.12	33.57	70.18	74.77	83.21	77.43	53.50

- Metric: AUROC, AUPR, FPR for detection scores of IND-Te and OOD-Te samples
- Twitch (multi-graph dataset): use nodes in different graphs for IND/OOD
- Arxiv (a temporal graph dataset): use nodes at different times for IND/OOD

Qitian Wu, et al., "Energy-based Out-of-Distribution Detection for Graph Neural Networks", in ICLR'23

Energy Score Visualization



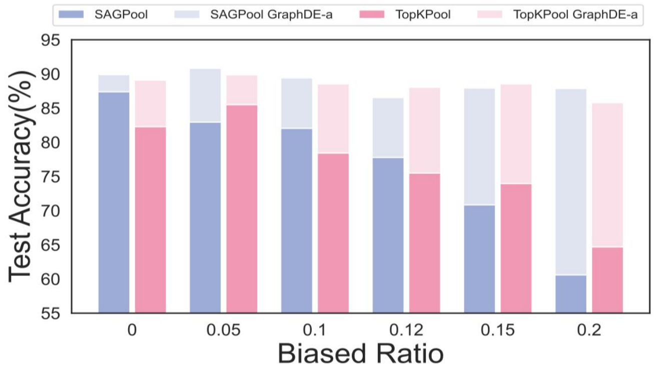
Energy propagation and regularization can both help to enlarge the discrimination gap

Qitian Wu, et al., “Energy-based Out-of-Distribution Detection for Graph Neural Networks”, in ICLR'23

Generative Models for Graph OOD Detection

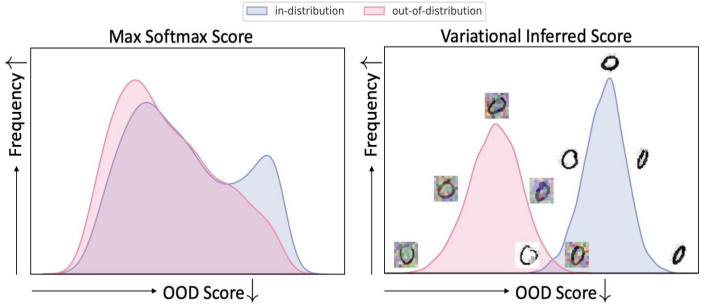
- Define the generative models of node features, graph structures and node labels as two-component mixtures.

$$p_\theta(\mathbf{A}|\mathbf{X},\mathbf{e}) = p_\theta(\mathbf{A}|\mathbf{X})^{\mathbf{e}} p_0(\mathbf{A}|\mathbf{X})^{1-\mathbf{e}},$$
$$p_\theta(\mathbf{y}|\mathbf{X},\mathbf{A},\mathbf{e}) = p_\theta(\mathbf{y}|\mathbf{X},\mathbf{A})^{\mathbf{e}} p_0(\mathbf{y}|\mathbf{X},\mathbf{A})^{1-\mathbf{e}}.$$



- Compute the OOD scores for testing data by Bayesian rule:

$$p_\theta(\mathbf{e}|\mathbf{A},\mathbf{X}) = \frac{p_\theta(\mathbf{e},\mathbf{A},\mathbf{X})}{\sum_{\mathbf{e}} p_\theta(\mathbf{e},\mathbf{A},\mathbf{X})} = \frac{p(\mathbf{e})p(\mathbf{X})p_\theta(\mathbf{A}|\mathbf{X},\mathbf{e})}{\sum_{\mathbf{e}} p(\mathbf{e})p(\mathbf{X})p_\theta(\mathbf{A}|\mathbf{X},\mathbf{e})}.$$

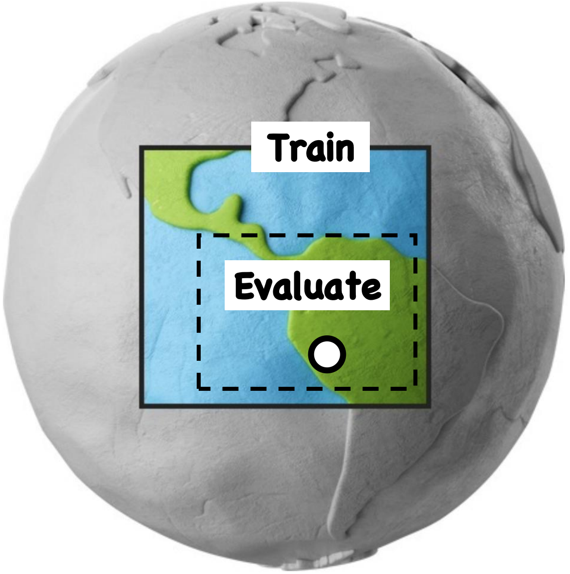


Theoretical Justifications:

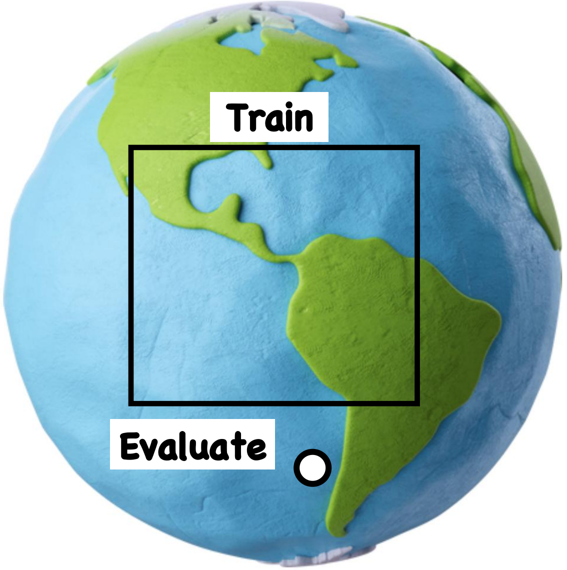
The model can automatically identify outliers in training data and OOD samples from testing data

Zenan Li et al., "GraphDE: A Generative Framework for Debiased Learning and Out-of-Distribution Detection on Graphs", in NeurIPS'22

From Closed-World to Open-World Learning



How to learn a desirably effective model under distribution shifts?



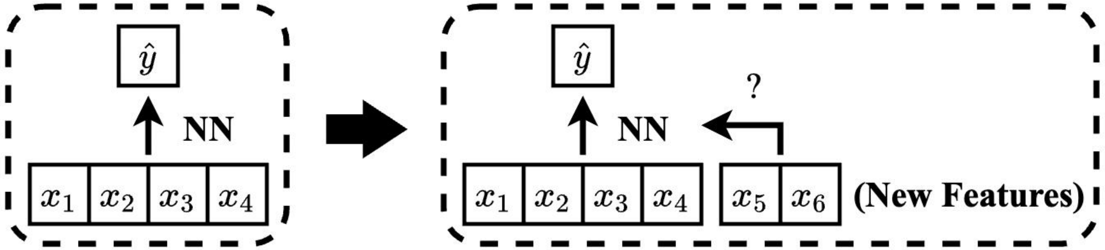
The challenging open research problems:

- How to train a model that can generalize to OOD data? → OOD Generalization
- How to train a model that can identify OOD data? → OOD Detection
- How to enable a model to handle new unseen entities? → OOD Extrapolation

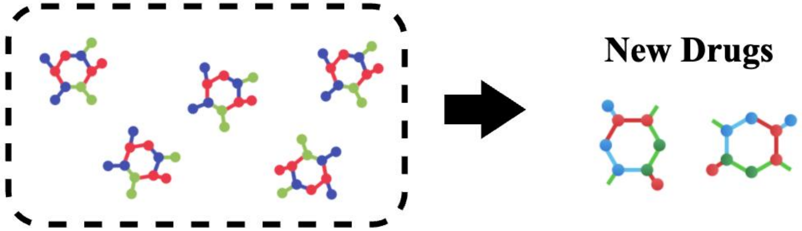
New Entities from Open World

- New users/items in recommender systems

- New features collected by new released platforms for decisions



- New developed drugs or combinations for treatment



How to handle unseen entities that are not exposed to model training?

Feature Space Extrapolation

- **Open-world feature extrapolation:**

Training Set

Data $\{(\mathbf{x}_i, y_i)\}_{i \in I_{tr}}$

Feature space $\mathbf{x}_i \in \mathcal{X}_{tr} = \{0,1\}^D$

Label space $y_i \in \mathcal{Y}$

domain generalization →

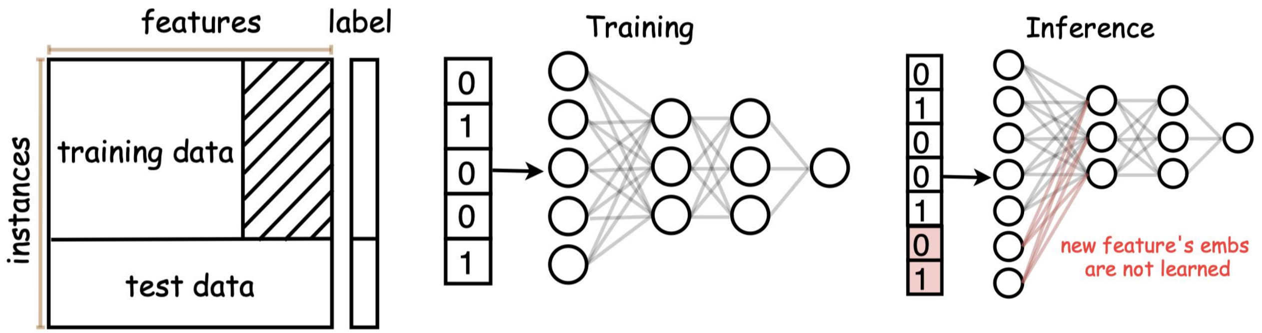
$\mathcal{X}_{tr} \subset \mathcal{X}_{te}$

Test Set

Data $\{(\mathbf{x}_{i'}, y_{i'})\}_{i' \in I_{te}}$

Feature space $\mathbf{x}_{i'} \in \mathcal{X}_{te} = \{0,1\}^{D'}$

Label space $y_{i'} \in \mathcal{Y}$



Key questions:
Can we enable neural networks to handle augmented input dimensions without re-training?

Qitian Wu et al., "Towards Open-World Feature Extrapolation: An Inductive Graph Learning Approach", in NeurIPS'21

Input Data as Graphs

- The input feature-data matrix can be treated as a bipartite graph

Input data matrix

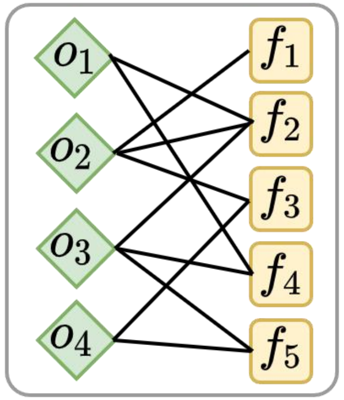
$$\mathbf{X}_{tr} = [\mathbf{x}_i]_{i \in I_{tr}} \in \{0,1\}^{N \times D}$$

- Feature nodes $F_{tr} = \{f_j\}_{j=1}^{D}$
- Instance nodes $I_{tr} = \{o_i\}_{i=1}^{N}$
- Adjacency matrix $\mathbf{X}_{tr}$

Observed Data Matrix

	f_1	f_2	f_3	f_4	f_5
o_1	0	1	0	1	0
o_2	1	1	1	0	0
o_3	0	1	0	1	1
o_4	0	0	1	0	1

Feature-Data Graph

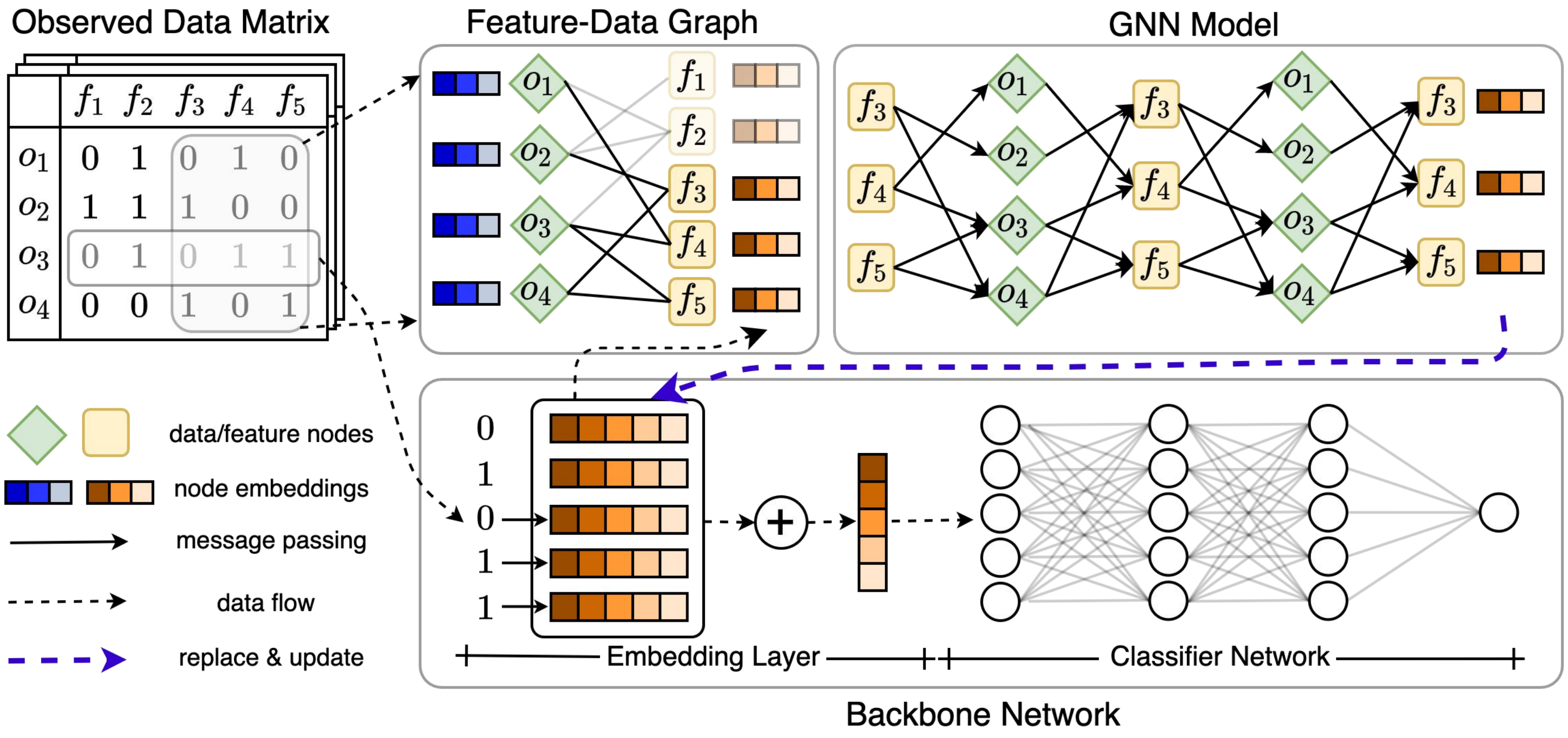


Advantage of graph representation:
Variable-size for features/instances

Key insight:
Convert inferring embeddings for new features to inductive representation on graphs

Qitian Wu et al., “Towards Open-World Feature Extrapolation: An Inductive Graph Learning Approach”, in NeurIPS'21

Extrapolation with Message Passing



Reasoning

High-level model for feature emb learning

Prediction

Low-level model for prediction

Qitian Wu et al., “Towards Open-World Feature Extrapolation: An Inductive Graph Learning Approach”, in NeurIPS'21

Results on Advertisement Click Prediction

Table. ROC-AUC results for eight test sets (T1 - T8) on Avazu and Criteo

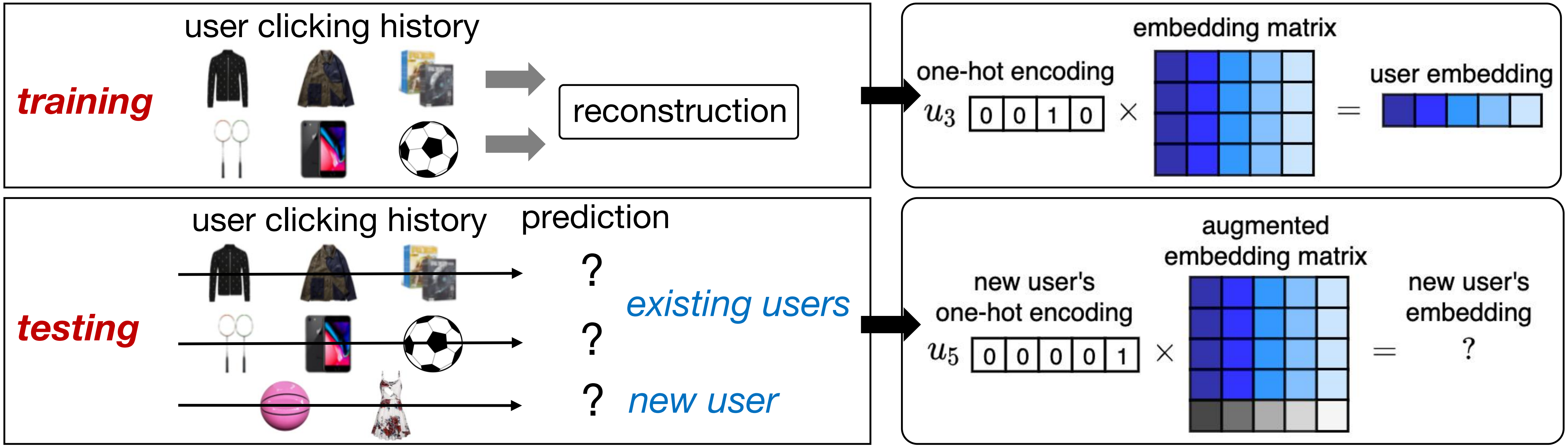
Dataset	Backbone	Model	T1	T2	T3	T4	T5	T6	T7	T8	Overall
Avazu	NN	Base	0.666	0.680	0.691	0.694	0.699	0.703	0.705	0.705	0.693 ± 0.012
		Pooling	0.655	0.671	0.683	0.683	0.689	0.694	0.697	0.697	0.684 ± 0.011
		FATE	**0.689**	**0.699**	**0.708**	**0.710**	**0.715**	**0.720**	**0.721**	**0.721**	**0.710** ± 0.010
	DeepFM	Base	0.675	0.684	0.694	0.697	0.699	0.706	0.708	0.706	0.697 ± 0.009
		Pooling	0.666	0.676	0.685	0.685	0.688	0.693	0.694	0.694	0.685 ± 0.009
		FATE	**0.692**	**0.702**	**0.711**	**0.714**	**0.718**	**0.722**	**0.724**	**0.724**	**0.713** ± 0.010
Criteo	NN	Base	0.761	0.761	0.763	0.763	0.765	0.766	0.766	0.766	0.764 ± 0.002
		Pooling	0.761	0.762	0.764	0.763	0.766	0.767	0.768	0.768	0.765 ± 0.001
		FATE	**0.770**	**0.769**	**0.771**	**0.772**	**0.773**	**0.774**	**0.774**	**0.774**	**0.772** ± 0.001
	DeepFM	Base	0.772	0.771	0.772	0.772	0.774	0.774	0.774	0.774	0.773 ± 0.001
		Pooling	0.772	0.772	0.773	0.774	0.776	0.776	0.776	0.776	0.774 ± 0.002
		FATE	**0.781**	**0.780**	**0.782**	**0.782**	**0.784**	**0.784**	**0.784**	**0.784**	**0.783** ± 0.001

- FATE achieves significantly improvements over Base/Pooling with different backbones (DNN and DeepFM)

Qitian Wu et al., "Towards Open-World Feature Extrapolation: An Inductive Graph Learning Approach", in NeurIPS'21

Input Space Expansion - Cold-Start Users

- ***open-world recommendation:*** **new unseen users appear in test data**



- Challenges: For new users, there is no available embeddings from model training

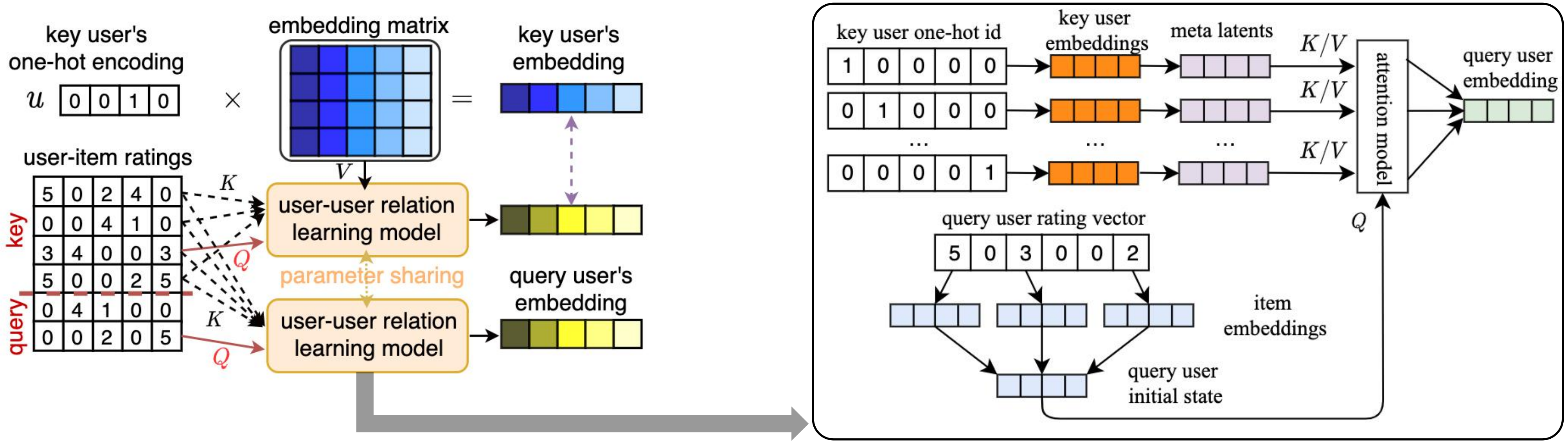
Can we enable a recommendation model to directly generalize to new users ?

Extrapolation with Graph Structure Learning

❑ Basic idea:

- leverage one group of users to express another
- learn a latent graph over users
- message passing from existing users to new ones

Key insight: user preferences share underlying proximity that induces latent graphs

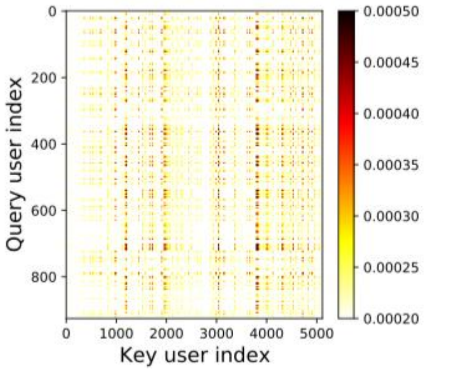


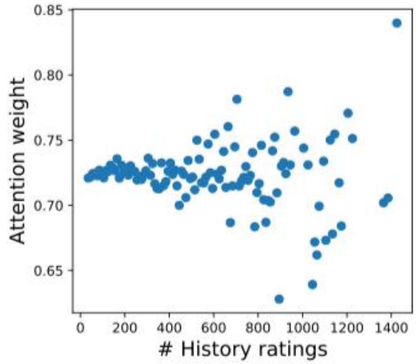
Qitian Wu et al., "*Towards Open-World Recommendation: An Inductive Model-based Collaborative Filtering Approach*", in ICML'21

Results on Recommendation Benchmarks

- Task 1: Transferring to few-shot users with limited interaction records
- Task 2: Generalizing to zero-shot users unseen by training

Method	Inductive	Feature	Douban				ML-100K				ML-1M			
			RMSE		NDCG		RMSE		NDCG		RMSE		NDCG	
			All	FS	All	FS	All	FS	All	FS	All	FS	All	FS
PMF	No	No	0.737	0.718	0.939	0.954	0.932	1.003	0.858	0.843	0.851	0.946	0.919	**0.940**
NNMF	No	No	0.729	**0.705**	0.939	0.952	0.925	0.987	0.895	0.878	0.848	**0.940**	0.920	0.937
GCMC	No	No	0.731	0.706	0.938	**0.956**	0.911	0.989	0.900	**0.886**	**0.837**	0.947	0.923	0.939
NIMC	Yes	Yes	0.732	0.745	0.928	0.931	1.015	1.065	0.832	0.824	0.873	0.995	0.889	0.904
BOMIC	Yes	Yes	0.735	0.747	0.923	0.925	0.931	1.001	0.828	0.815	0.847	0.953	0.905	0.924
F-EAE	Yes	No	0.738	-	-	-	0.920	-	-	-	0.860	-	-	-
IGMC	Yes	No	**0.721**	0.728	-	-	**0.905**	0.997	-	-	0.857	0.956	-	-
IDCF-NN (ours)	Yes	No	0.738	0.712	0.939	**0.956**	0.931	0.996	0.896	0.880	0.844	0.952	0.922	**0.940**
IDCF-GC (ours)	Yes	No	0.733	0.712	**0.940**	**0.956**	**0.905**	**0.981**	**0.901**	0.884	0.839	0.944	**0.924**	**0.940**





+4.0% (resp. +17.4%) impv. of RMSE (resp. NDCG) on new users

Qitian Wu et al., "*Towards Open-World Recommendation: An Inductive Model-based Collaborative Filtering Approach*", in ICML'21

References

Out-of-Distribution Generalization:

[1] Qitian Wu, et al., Handling Distribution Shifts on Graphs: An Invariance Perspective, in ICLR'22

[2] Nianzu Yang, et al., Learning Substructure Invariance for Out-of-Distribution Molecular Representations, in NeurIPS'22

[3] Chenxiao Yang et al., Towards out-of-distribution sequential event prediction: A causal treatment, in NeurIPS'22

[4] Chenxiao Yang et al., Graph Neural Networks are Inherently Good Generalizers: Insights by Bridging GNNs and MLPs, in ICLR'23

Out-of-Distribution Detection:

[5] Qitian Wu et al., Energy-based Out-of-Distribution Detection for Graph Neural Networks, in ICLR'23

[6] Zenan Li et al., GraphDE: A Generative Framework for Debiased Learning and Out-of-Distribution Detection on Graphs, in NeurIPS'22

Out-of-Distribution Extrapolation:

[7] Qitian Wu et al., Towards Open-World Feature Extrapolation: An Inductive Graph Learning Approach, in ICML'21

[8] Qitian Wu et al., Towards Open-World Recommendation: An Inductive Model-based Collaborative Filtering Approach, in NeurIPS'21

Chinese Blog

https://www.zhihu.com/column/c_1571878639037202432