Handling Distribution Shifts on Graphs: An Invariance Perspective

International Conference on Learning Representations (ICLR 2022)

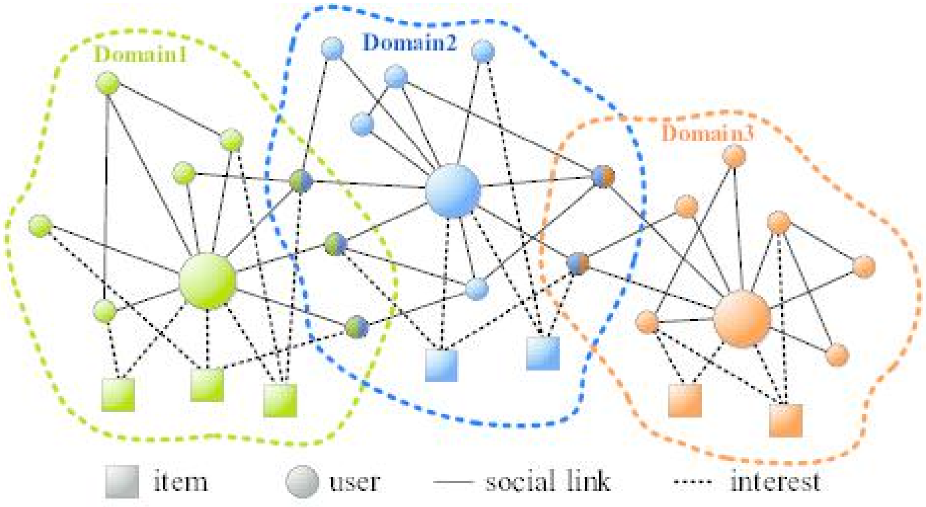
Qitian Wu[1], Hengrui Zhang[2], Junchi Yan[1], David Wipf[3]

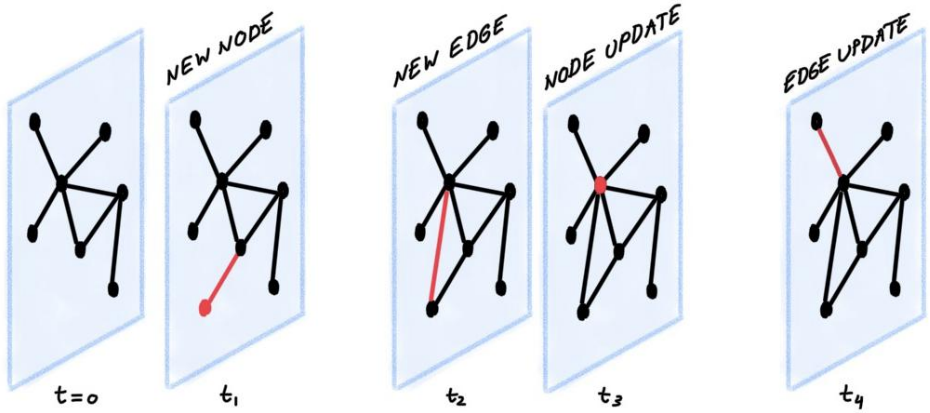
[1]Shanghai Jiao Tong University

[2]University of Illinois at Chicago

[3]Amazon Web Service

Distribution Shifts on Graph Data



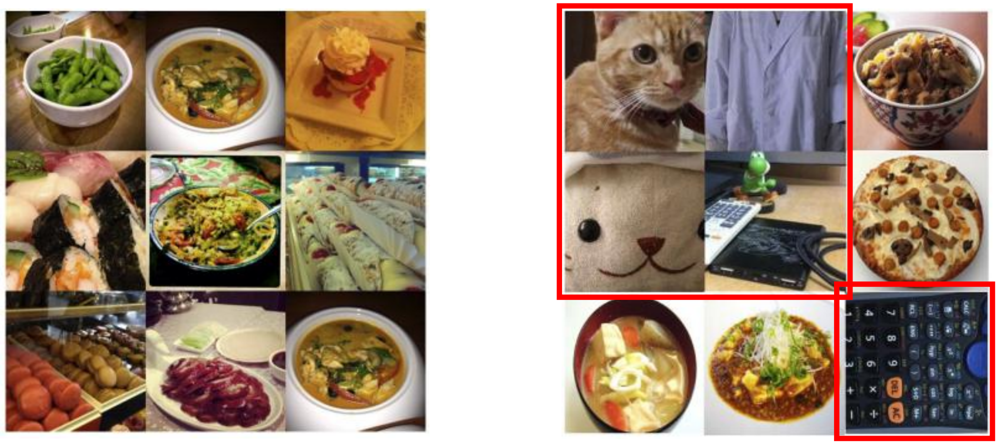


Graph data from multiple domains | **Dynamic temporal networks**

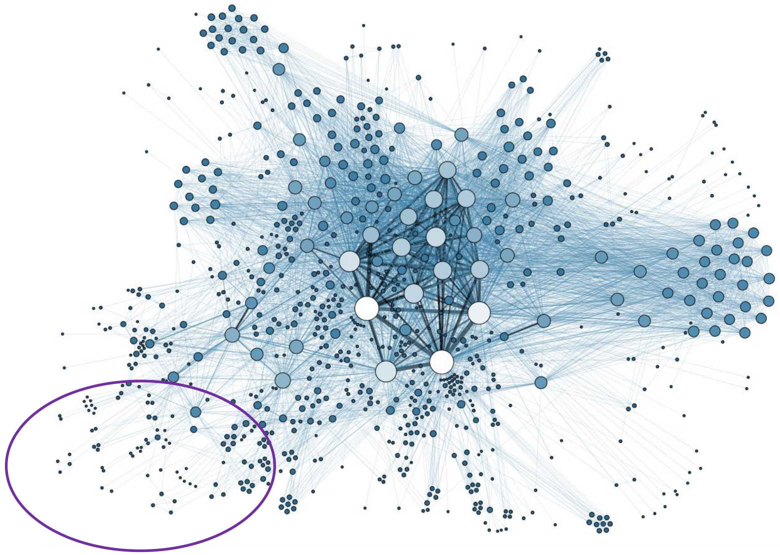
- ❑ Distribution shifts cause different data distributions $P_{train}(\mathcal{D}) \neq P_{test}(\mathcal{D})$
- ❑ New data from unknown distribution are unseen by training
- ❑ Distribution shifts involve structural information of non-Euclidean data

Distribution Shifts on Graphs

- Out-of-distribution data are ubiquitous in real-world situations
- ML systems are difficult to generalize to new test distributions
- Unlike images, OOD samples are ambigous for graph-structured data

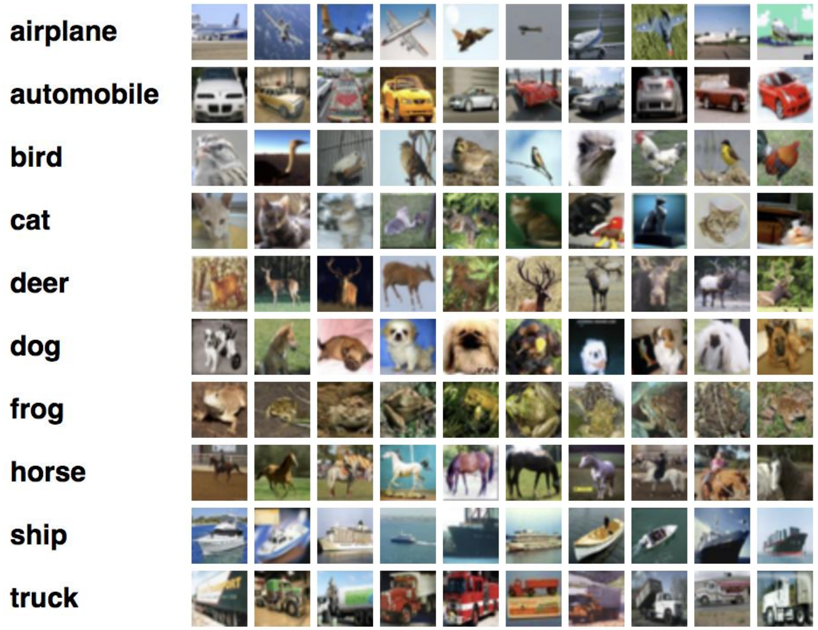


Out-of-distribution samples can be clearly defined for image data



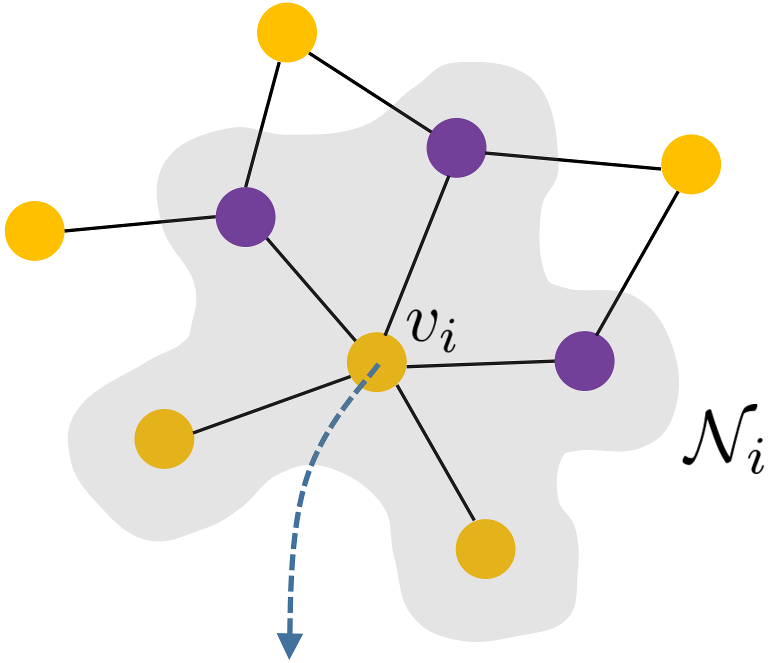
OOD samples?

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

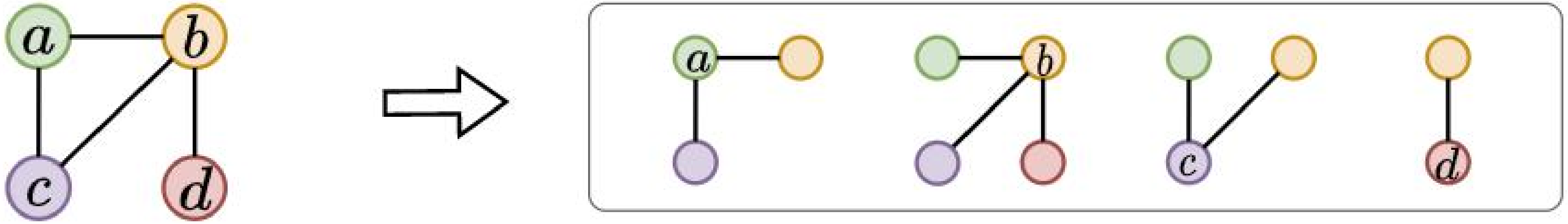
Problem Formulation

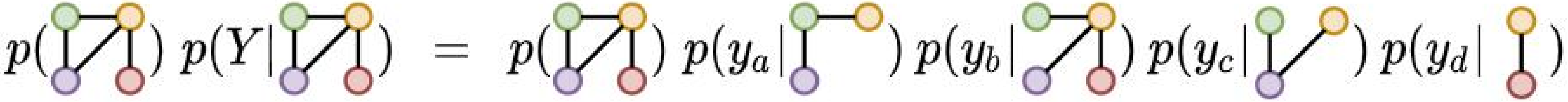
- **Graph notation:** A graph $G=(A,X)$, adjacency matrix $A=\{a_{uv}|v,u\in V\}$ node features $X=\{x_v|v\in V\}$, node labels $Y=\{y_v|v\in V\}$

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

  where e denotes environment (that affects data generation)

- How to deal with the non-IID nature of nodes in a graph?





$$p(\mathbf{G}|\mathbf{e})\cdot(\mathbf{Y}|\mathbf{G},\mathbf{e}) = p(\mathbf{G}|\mathbf{e})\cdot\prod_{v\in V}p(\mathbf{y}|\mathbf{G_v}=G_v,\mathbf{e})$$

Decompose a graph into pieces of ego-graphs

Problem Formulation

- **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 $\mathbf{e}$ 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 (→ $\min_f \max_{e\in\mathcal{E}}$)

sample a whole graph from a specific environment (→ $\mathbb{E}_{G\sim p(\mathbf{G}|\mathbf{e}=e)}$)

sample node-level label conditioned on ego-graph and environment (→ $\mathbb{E}_{y\sim p(\mathbf{y}|\mathbf{G_v}=G_v,\mathbf{e}=e)}$)

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 $y\sim p(\mathbf{y}|\mathbf{G_v}=G_v,\mathbf{e})$
  - Denote $\mathcal{E}$ as the support of env. and $l(\cdot,\cdot)$ as the loss function

Causal Invariance Principle

Assumption 1 (Invariance Property)

There exists a sequence of (non-linear) functions $\{h_l^*\}_{l=0}^L$ where $h_l^* : \mathbb{R}^{d_0} \rightarrow \mathbb{R}^d$ and a permutation-invariant function $\Gamma : \mathbb{R}^{d^m} \rightarrow \mathbb{R}^d$, which gives a node-level readout $r_v = r_v^{(L)}$ that is calculated in a recursive way: $r_u^{(l)} = \Gamma\{r_w^{(l-1)} | w \in N_u^{(1)} \cup \{u\}\}$ for $l = 1, \cdots, L$ and $r_u^{(0)} = h_l^*(x_u)$ if $u \in N_v^{(l)}$. Denote $\mathbf{r}$ as a random variable of r_v and it satisfies

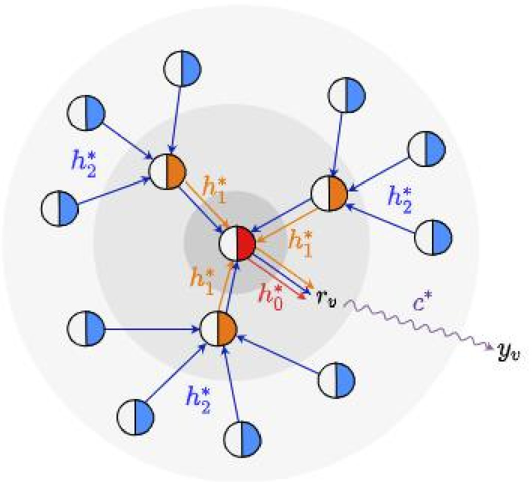
inspired by Weisfeiler-Lehman test

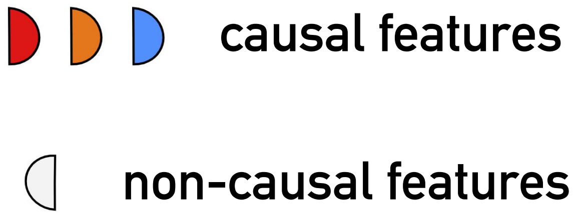
- *Invariance condition*: $p(\mathbf{y}|\mathbf{r}, \mathbf{e}) = p(\mathbf{y}|\mathbf{r})$
- *Sufficiency condition:* $\mathbf{y} = c^*(\mathbf{r}) + \mathbf{n}$, where c^* is a non-linear function, $\mathbf{n}$ is a random noise.

Intuitive Explanation:

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





Motivating Example

We consider a linear 2-dim toy example and 1-layer GNN model

Data generation: 2-dim node features $x_v = [x_v^1, x_v^2]$ and node label y_v

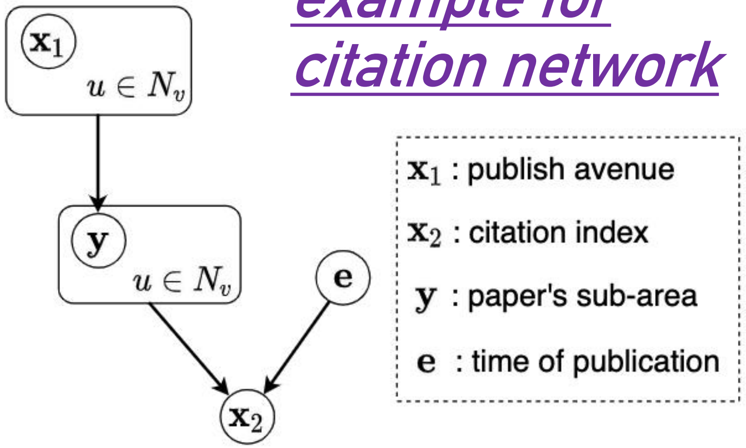
$$y_v = \frac{1}{|N_v|} \sum_{u \in N_v} x_u^1 + n_v^1, \quad x_v^2 = \frac{1}{|N_v|} \sum_{u \in N_v} y_u + n_v^2 + \epsilon$$

where n_v^1 and n_v^2 are standard normal noise and ϵ is a random variable with zero mean and non-zero variance dependent on the environment.

Model: a vanilla GCN as the predictor model:

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

example for citation network



The ideal solution is $[\theta_1, \theta_2] = [1, 0]$

x_v^1 causal features x_v^2 non-causal (spurious) features

Motivating Example (Cont.)

Proposition 1 (Failure of Empirical Risk Minimization)

Let the risk under environment e be

$$R(e) = \frac{1}{|V|} \sum_{v \in V} \mathbb{E}_{\mathbf{y}|\mathbf{G_v}=G_v}[\|\hat{y}_v - y_v\|_2^2].$$

The unique optimal solution for objective $\min_\theta \mathbb{E}_\mathbf{e}[R(e)]$ would be $[\theta_1, \theta_2] = [\frac{1+\sigma_e^2}{2+\sigma_e^2}, \frac{1}{2+\sigma_e^2}]$ where $\sigma_e > 0$ denotes the standard deviation of ϵ across environments.

Proposition 2 (Success of Risk Variance Minimization)

The objective $\min_\theta \mathbb{V}_e[R(e)]$ reaches the optimum if and only if $[\theta_1, \theta_2] = [1, 0]$.

- Implication from Prop 1: minimizing the expectation of risks across environments would inevitably lead the model to rely on spurious correlation
- Implication from Prop 2: if the model yields equal performance on different environments, it would manage to leverage the invariant features

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: no/ambiguous environment in observed data

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

Risk Extrapolation ➡

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

Environment Exploration ➡

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

where

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

$L(g_{w_k}(G), Y; \theta)$: risk function for data under the k-th environment

$g_{w_k}(G)$: context generator: augment training data and simulate multiple environments

$f_\theta(G_v^k)$: predictor: graph neural networks for classification

Explore-to-Extrapolate Risk Minimization

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

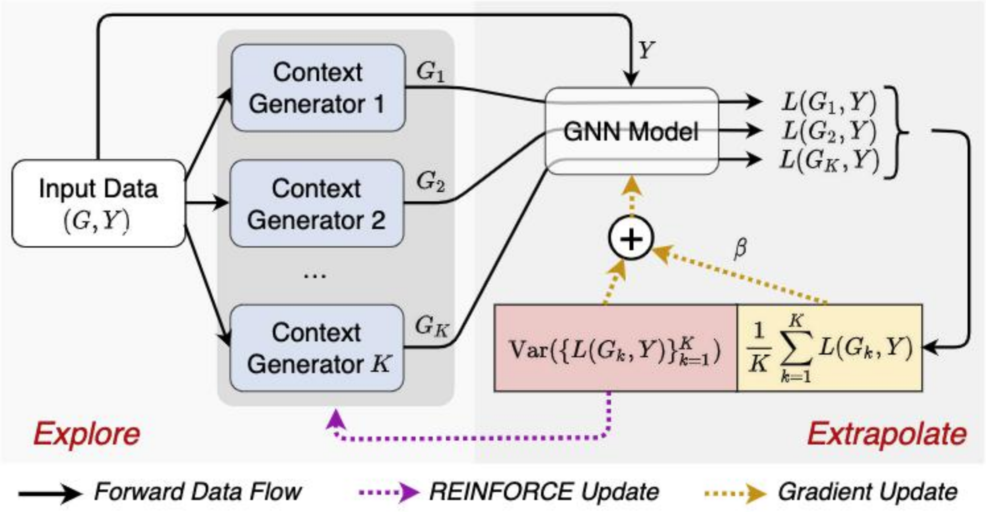
context generator: augment training data and simulate multiple environments

risk function for data under the k-th environment

predictor: graph neural networks for classification

❑ Model instantiations:

- $f_\theta(\cdot)$: GNN (output node-level prediction)
- $g_{w_k^*}(\cdot)$: graph editer (output a new graph via add/delete edges)
- Training algorithm: REINFORCE for graph editer + gradient descent for GNN predictor



Theoretical Analysis

Assumption 2 (Environment Heterogeneity)

For $(\mathbf{G_v}, \mathbf{r})$ that satisfies Assumption 1, there exists a random variable $\overline{\mathbf{r}}$ such that $\mathbf{G_v} = m(\mathbf{r}, \overline{\mathbf{r}})$ where m is a functional mapping. We assume that $p(\mathbf{y}|\overline{\mathbf{r}}, \mathbf{e} = e)$ would arbitrarily change across environments $e \in \mathcal{E}$.

Intuitive Explanation: two portions of features in input data, one is domain-invariant for prediction and the other contributes to sensitive prediction that can arbitrary change on environments.

Theorem 1 (Interpretations for New Learning Objective)

If we treat the predictive distribution $q(\mathbf{y}|\mathbf{z})$ as a variational distribution, then 1) minimizing the expectation of risks contributes to $\max_{q(\mathbf{z}|\mathbf{G_v})} I(\mathbf{y}; \mathbf{z})$, i.e., enforcing the sufficiency condition on $\mathbf{z}$ for prediction, and 2) minimizing the variance of risks would play a role for $\min_{q(\mathbf{z}|\mathbf{G_v})} I(\mathbf{y}; \mathbf{e}|\mathbf{z})$, i.e., enforcing the invariance condition $p(\mathbf{y}|\mathbf{z}, \mathbf{e}) = p(\mathbf{y}|\mathbf{z})$.

Theoretical Analysis (Cont.)

Theorem 2 (Guarantee of Valid OOD solution)

Under Assumption 1 and 2, if the GNN encoder $q(\mathbf{z}|\mathbf{G_v})$ satisfies that 1) $I(\mathbf{y};\mathbf{e}|\mathbf{z}) = 0$ (invariance condition) and 2) $I(\mathbf{y};\mathbf{z})$ is maximized (sufficiency condition), then the model f^* given by $\mathbb{E}_{\mathbf{y}}[\mathbf{y}|\mathbf{z}]$ is the solution to the formulated OOD problem.

From information-theoretic perspective,

1) training error $D_{KL}(p_e(\mathbf{y}|\mathbf{G_v})\|q(\mathbf{y}|\mathbf{G_v})) \leq I_e(\mathbf{G_v};\mathbf{y}|\mathbf{z}) + D_{KL}(p_e(\mathbf{y}|\mathbf{z})\|q(\mathbf{y}|\mathbf{z}))$

2) OOD generalization error $D_{KL}(p_{e'}(\mathbf{y}|\mathbf{G_v})\|q(\mathbf{y}|\mathbf{G_v})) \leq I_{e'}(\mathbf{G_v};\mathbf{y}|\mathbf{z}) + D_{KL}(p_{e'}(\mathbf{y}|\mathbf{z})\|q(\mathbf{y}|\mathbf{z}))$

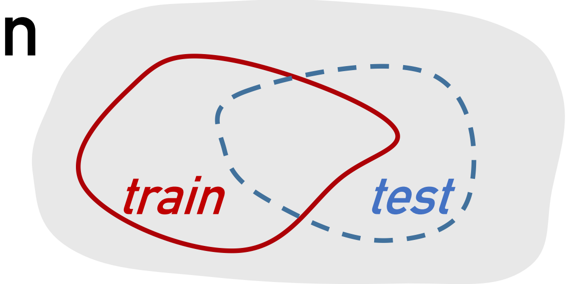
Theorem 3 (Effectiveness for Reducing OOD Generalization Error)

Optimizing the learning objective with training data can minimize the upper bound for OOD error measured by $D_{KL}(p_{e'}(\mathbf{y}|\mathbf{G_v})\|q(\mathbf{y}|\mathbf{G_v})$ on condition that $I_{e'}(\mathbf{G_v};\mathbf{y}|\mathbf{z}) = I_e(\mathbf{G_v};\mathbf{y}|\mathbf{z})$.

Experiment Setup

Dataset	Distribution Shift	#Nodes	#Edges	#Classes	Train/Val/Test Split	Metric
Cora	Artificial Transformation	2,703	5,278	10	Domain-Level	Accuracy
Amazon-Photo		7,650	119,081	10	Domain-Level	Accuracy
Twitch-explicit	Cross-Domain Transfers	1,912 - 9,498	31,299 - 153,138	2	Domain-Level	ROC-AUC
Facebook-100		769 - 41,536	16,656 - 1,590,655	2	Domain-Level	Accuracy
Elliptic	Temporal Evolution	203,769	234,355	2	Time-Aware	F1 Score
OGB-Arxiv		169,343	1,166,243	40	Time-Aware	Accuracy

- ❑ **Evalution protocol of out-of-distribution generalization**
  - Training on limited data and testing on new unseen data
  - Differences between training and testing distributions



- ❑ **Three types of distribution shifts on graphs**
  - *Artificial transformation:* add synthetic spurious node features to data
  - *Cross-domain transfers:* training and testing within different graphs
  - *Temporal evolution:* training in the past and evaluation in the future

Results on Artificial Transformation

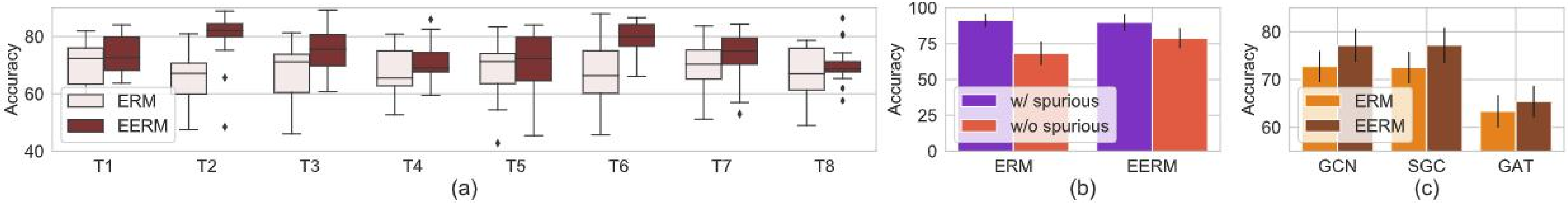


Figure. Experiment results on Cora with artificial spurious features. (a) Test accuracy on eight testing graphs (with different environment ids). (b) Training accuracy during inference w/ and w/o using spurious features. (c) Averaged test accuracy using different GNNs for synthetic data generation.

- Setup: use a randomly initialized GCN to generate spurious node features, use another GCN to generate ground-truth node labels based on input node features
- Results (when using GCN as the predictor backbone):
  - EERM (ours) outperforms empirical risk minimization (ERM) on eight test graphs
  - EERM can reduce the dependence on spurious features than ERM
  - EERM is robust to synthetic data generated by different GNNs

Results on Cross-Graph Transfer

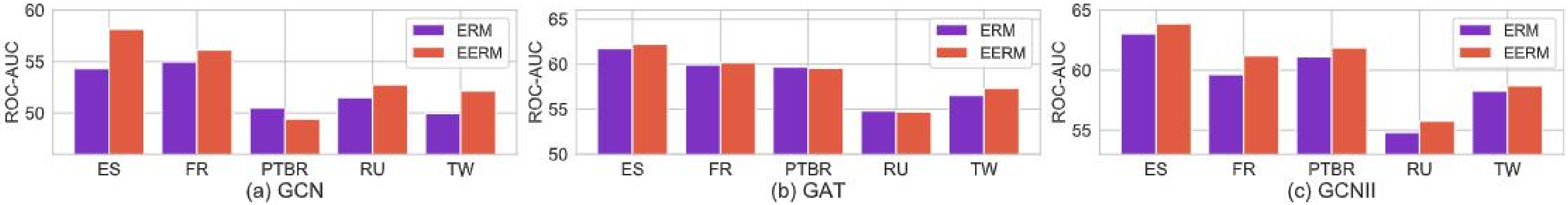


Figure. ROC-AUC results on Twitch-Explicit when training on one graph and testing on others with different GNN predictors (GCN, GAT and GCNII).

Table. Accuracy results on Facebook-100 when using different configurations of training graphs and testing on new graphs Penn, Brown and Texas

Training graph combination	Penn		Brown		Texas	
	ERM	EERM	ERM	EERM	ERM	EERM
John Hopkins + Caltech + Amherst	50.48 ± 1.09	50.64 ± 0.25	54.53 ± 3.93	56.73 ± 0.23	53.23 ± 4.49	55.57 ± 0.75
Bingham + Duke + Princeton	50.17 ± 0.65	50.67 ± 0.79	50.43 ± 4.58	52.76 ± 3.40	50.19 ± 5.81	53.82 ± 4.88
WashU + Brandeis+ Carnegie	50.83 ± 0.17	51.52 ± 0.87	54.61 ± 4.75	55.15 ± 3.22	56.25 ± 0.13	56.12 ± 0.42

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

Results on Temporal Graph Evolution

- **Dynamic graph snapshot** (Elliptic):
  - A graph is generated at every timestamp (nodes not shared)
  - Divide train/valid/test graphs according to timestamps
- **Temporal augmented graph** (OGB-Arxiv):
  - Nodes and edges are updated in one graph as time goes by
  - Divide train/valid/test nodes according to time features
  - Large time gaps between tr/te nodes

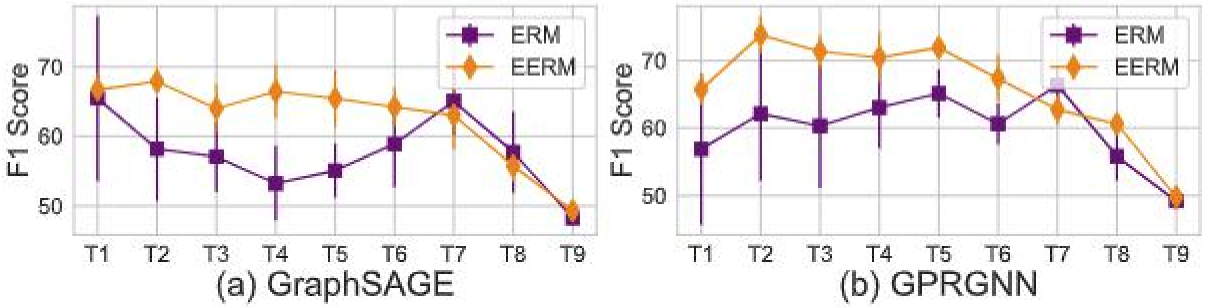


Figure. F1 score results on Elliptic with dynamic graph snapshots (chronologically divided into 9 test groups)

Table. Accuracy results on OGBN-Arxiv whose testing nodes are divided into three-fold according to time

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

Conclusions

Problem

We mathetically formulate the problem of out-of-distribution generalization on graphs

Re-formulate the invariance principle for graph-structured data as a cornerstone assumption

Methodology

We show by examples that traditional methods may fail with relying on spurious graph features

Propose a new invariant learning approach (explore-to-extrapolate risk minimization)

Theory

We prove that the new approach guarantee a valid solution for OOD generalization

Prove that the new objective can effectively reduce OOD error bound on new data

Evalution

We empirically verify the model with protocols including three different distribution shifts

The results on multiple GNN backbones show the superiority and robustness of our model

Code available at *https://github.com/qitianwu/GraphOOD-EERM*