Out-of-Distribution Generalization and Extrapolation on Graphs

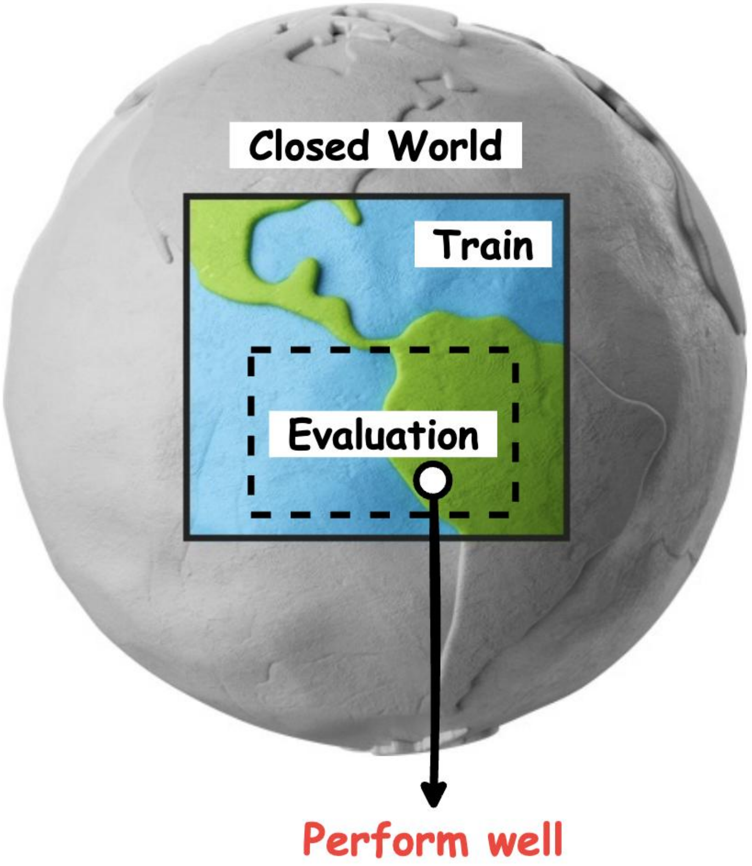
Qitian Wu (吴齐天)

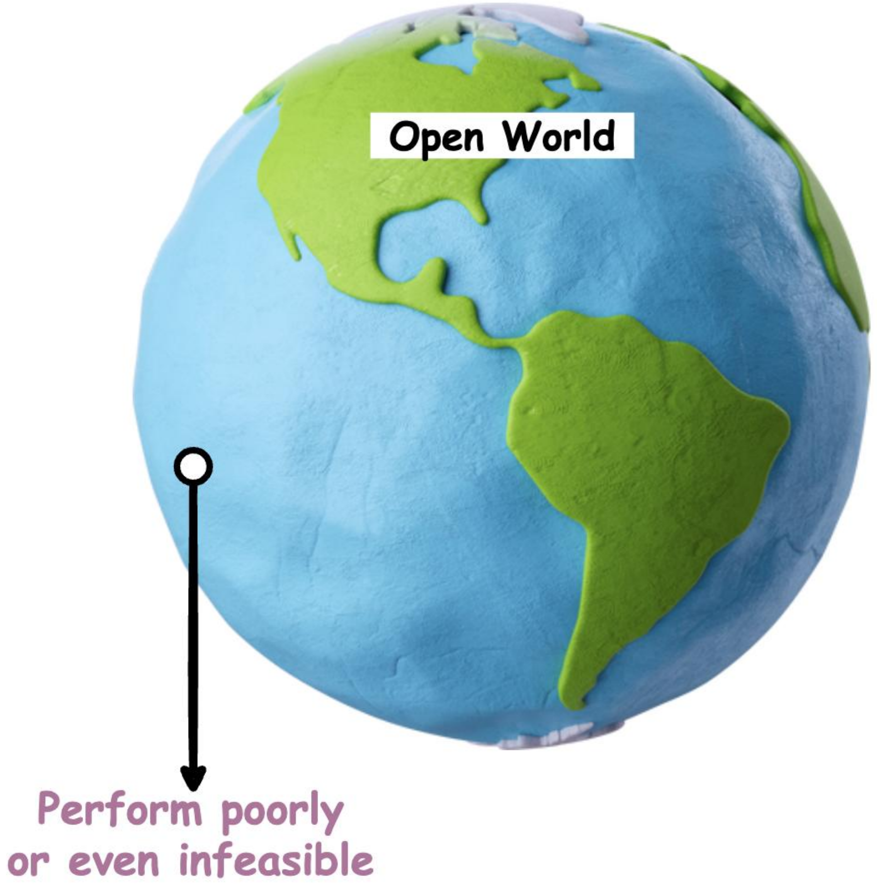
Department of Computer Science and Engineering

Shanghai Jiao Tong University

Motivation

- ❑ Machine learning models perform well in CLOSED-world situations
- ❑ Real-world situations are OPEN, dynamic and also uncertain

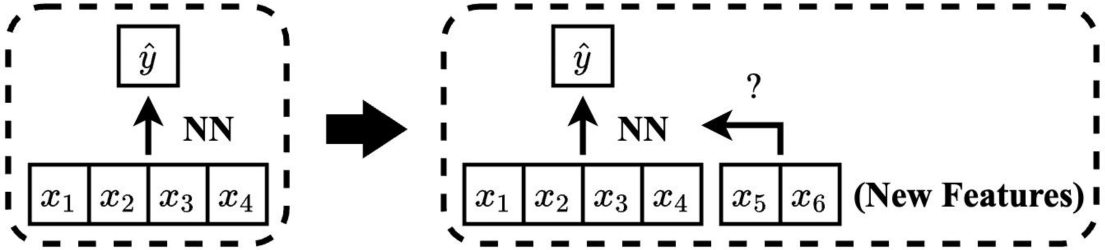




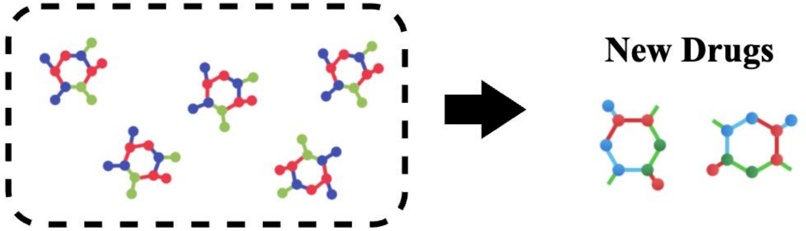
New Entities from Open World

- New users/items in recommender systems

- New features collected by new released platforms for decisions

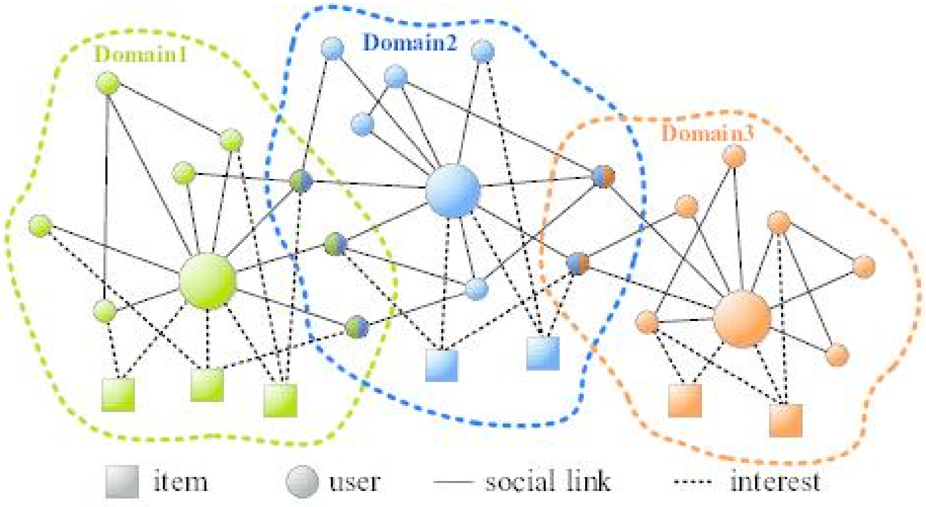


- New developed drugs or combinations for treatment

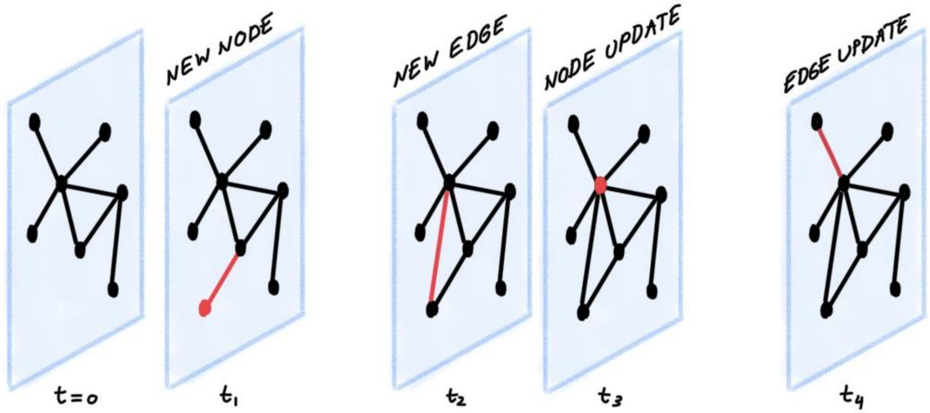


How to handle unseen entities that are not exposed to model training?

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

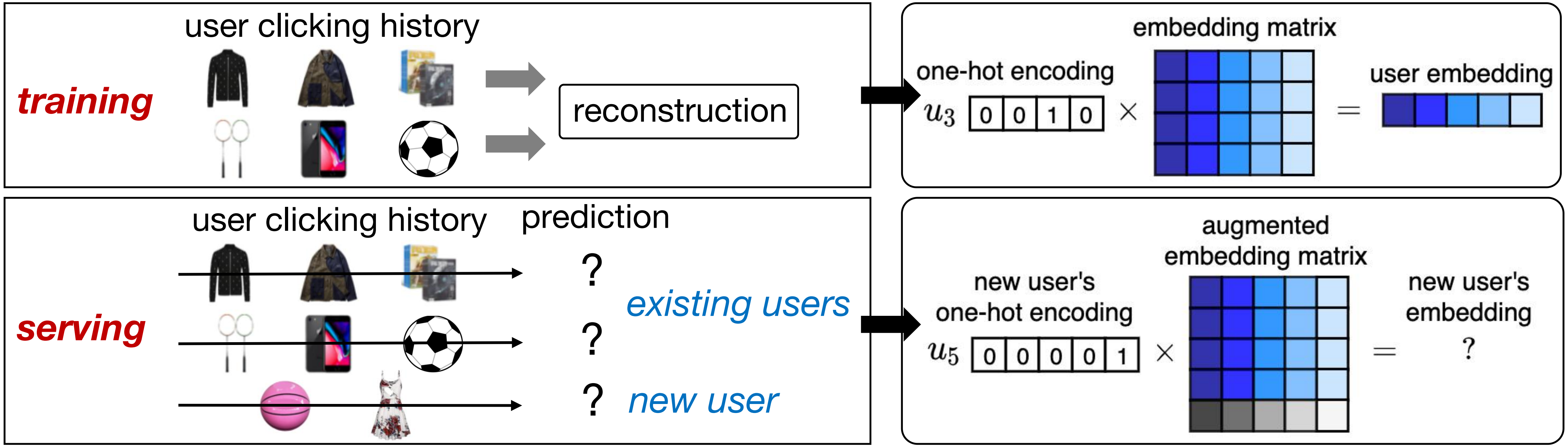
How to guarantee desired performance on data from new distributions?

Open-World Recommender Systems

- **Model-based Collaborative Filtering ≈ Matrix Factorization Model**
- **Basic idea:**

  user-item rating matrix $\xrightarrow{\text{factorize}}$ { **user embeddings**, item embeddings } → reconstruction matrix

- **CF models cannot handle new unseen users in *open-world recommendation***



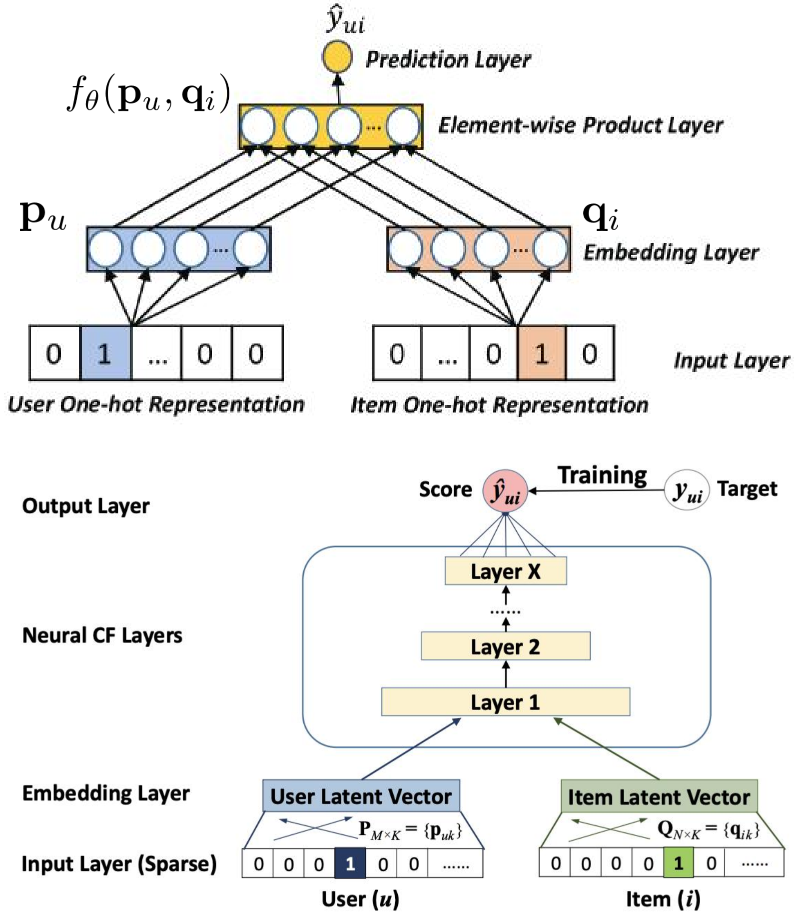
Collaborative Filtering

❑ Formulation of CF model for RecSys:

- a user-item interaction matrix $R = \{r_{ui}\}_{M \times N}$
- assume user latent factors $\mathbf{P} = \{\mathbf{p}_u\}_{M \times d}$
- assume item latent factors $\mathbf{Q} = \{\mathbf{q}_i\}_{N \times d}$
- consider an interaction model $\hat{r}_{ui} = f_\theta(\mathbf{p}_u, \mathbf{q}_i)$
- target objective $\mathcal{L}(\hat{R}, R) = \sum_{(u,i)} L(\hat{r}_{ui}, r_{ui})$

❑ Limitations: transductive learning

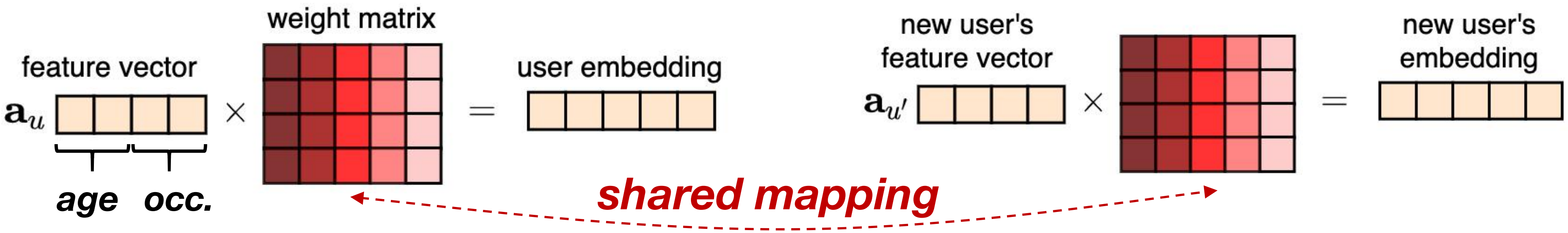
- cannot handle new unseen users
  - ◻ model retraining requires additional cost
  - ◻ incremental training may lead to over-fitting



adapted from [He et al. 2017]

Challenges for Inductive Learning

- Inductive learning: use user features as input



- Issue: expressiveness would be sacrificed with inductive learning

$$u_1 \xrightarrow{f_1} \mathbf{p}_{u_1}$$
$$u_2 \xrightarrow{f_2} \mathbf{p}_{u_2}$$

v. s.

$$\mathbf{a}_{u_1} \xrightarrow{f} \mathbf{p}_{u_1}$$
$$\mathbf{a}_{u_2} \xrightarrow{f} \mathbf{p}_{u_2}$$

transductive learning

pros: sufficient expressiveness
cons: fail for new users

inductive learning

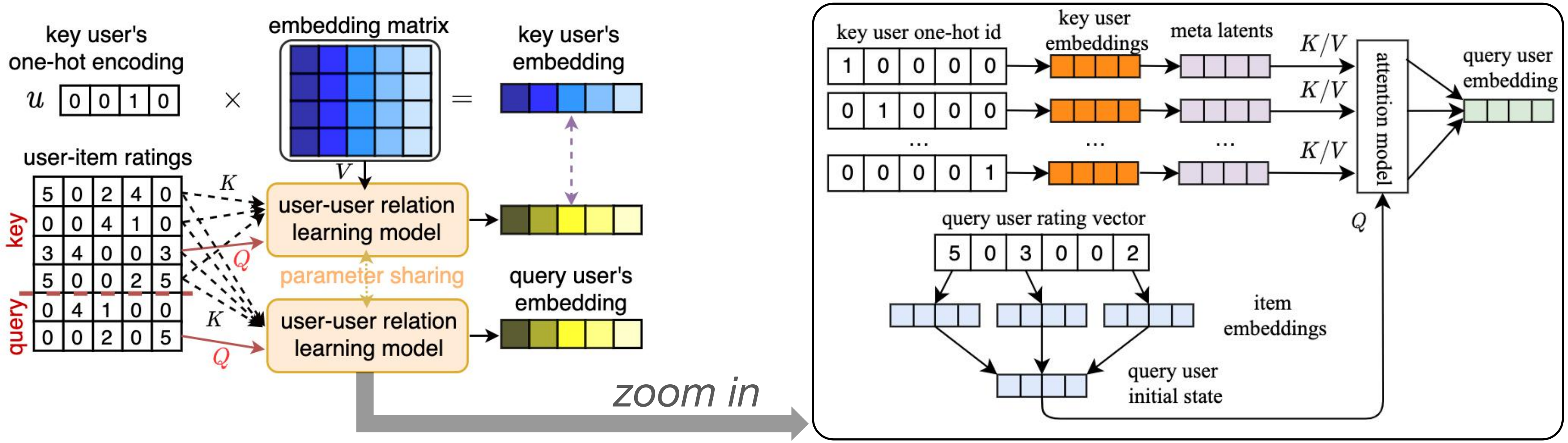
pros: flexible for new users
cons: limited capacity/expressiveness

Inductive Collaborative Filtering Model

❑ 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, Hengrui Zhang, Xiaofeng Gao, Junchi Yan, Hongyuan Zha, "*Towards Open-World Recommendation: An Inductive Model-based Collaborative Filtering Approach*", in ICML'21

Our Solutions: Inductive CF Model (Cont.)

- **Partition users into two groups:** $|\mathcal{U}_k| = M_k \quad |\mathcal{U}_q| = M_q$
  - Key users: transductive learning (traditional model)

model: $\mathbf{P}_k = \{\mathbf{p}_u\}_{M_k \times d} \qquad \mathbf{Q} = \{\mathbf{q}_i\}_{N \times d} \qquad \hat{r}_{ui} = f_\theta(\mathbf{p}_u, \mathbf{q}_i)$

learning: $\min_{\mathbf{P}_k, \mathbf{Q}, \theta} \mathcal{D}_{\mathcal{S}_k}(\hat{R}_k, R_k)$ **where** $R_k = \{r_{ui}\}_{M_k \times N}$

  - Query users: inductive learning (new model)

edge weights in a latent user-user graph

model: $\tilde{\mathbf{p}}_{\mathbf{u}'} = \mathbf{c}_{u'}^\top \mathbf{P}_k \qquad \boxed{c_{u'u} = \frac{\mathbf{e}^\top [\mathbf{W}_q \mathbf{d}_{u'} \oplus \mathbf{W}_k \mathbf{p}_u]}{\sum_{u_o \in \mathcal{U}_k} \mathbf{e}^\top [\mathbf{W}_q \mathbf{d}_{u'} \oplus \mathbf{W}_k \mathbf{p}_{u_0}]}}$ **where** $\mathbf{d}_{u'} = \sum_{i \in \mathcal{I}_{u'}} \mathbf{q}_i$

learning: $\min_{w,\theta} \mathcal{D}_{\mathcal{S}_q}(\hat{R}_q, R_q)$ **where** $R_q = \{r_{ui}\}_{M_q \times N} \qquad \hat{r}_{ui} = f_\theta(\tilde{\mathbf{p}}_{\mathbf{u}}, \mathbf{q}_i)$

objective: $\min_{w,\theta} \mathcal{D}_{\mathcal{S}_q}(\hat{R}_q, R_q) + \boxed{\lambda \mathcal{L}_C(\mathbf{P}_k, \tilde{\mathbf{P}}_{\mathbf{k}})} \qquad \mathcal{L}_C(\mathbf{P}_k, \tilde{\mathbf{P}}_{\mathbf{k}}) = \frac{1}{M_q} \sum_{u \in \mathcal{U}_k} \log \frac{\exp(\mathbf{p}_u^\top \tilde{\mathbf{p}}_{\mathbf{u}})}{\sum_{u' \in \mathcal{U}_q} \exp(\mathbf{p}_u^\top \tilde{\mathbf{p}}_{\mathbf{u}'})}$

regularization: consistency between two estimated embeddings for one user

Qitian Wu, Hengrui Zhang, Xiaofeng Gao, Junchi Yan, Hongyuan Zha, "*Towards Open-World Recommendation: An Inductive Model-based Collaborative Filtering Approach*", in ICML'21

Theoretical Analysis

- The model possesses the same representation capacity compared to matrix factorization
  - The only mild condition is that key users' latent factors span the latent space
- The generalization ability on new users depends on number of key users and training instances of new users

Theorem 1. *Assume Eq. (3) can achieve $\mathcal{D}_{\mathcal{S}_q}(\hat{R}_q, R_q) < \epsilon$ and the optimal $\mathbf{P}_k$ given by Eq. (1) satisfies column-full-rank, then there exists at least one solution for $\mathbf{C}$ in Eq. (2) such that $\mathcal{D}_{\mathcal{S}_q}(\hat{R}_q, R_q) < \epsilon$.*

$$\min_{\mathbf{P}_k, \mathbf{Q}, \theta} \mathcal{D}_{\mathcal{S}_k}(\hat{R}_k, R_k), \tag{1}$$

$$\min_{\mathbf{C}, \mathbf{Q}} \mathcal{D}_{\mathcal{S}_q}(\hat{R}_q, R_q), \tag{2}$$

$$\min_{\tilde{\mathbf{P}}_q, \mathbf{Q}} \mathcal{D}_{\mathcal{S}_q}(\hat{R}_q, R_q), \tag{3}$$

Theorem 2. *Assume 1) $\mathcal{D}$ is L-Lipschitz, 2) for $\forall \hat{r}_{u'i} \in \hat{R}_q$ we have $|\hat{r}_{u'i}| \leq B$, and 3) the L1-norm of $\mathbf{c}_{u'}$ is bounded by H. Then with probability at least $1 - \delta$ over the random choice of $\mathcal{S}_q \in ([M_q] \times [N])^{T_q}$, it holds that for any $\hat{R}_q$, the gap between $\mathcal{D}(\hat{R}_q, R_q)$ and $\mathcal{D}_{\mathcal{S}_q}(\hat{R}_q, R_q)$ will be bounded by*

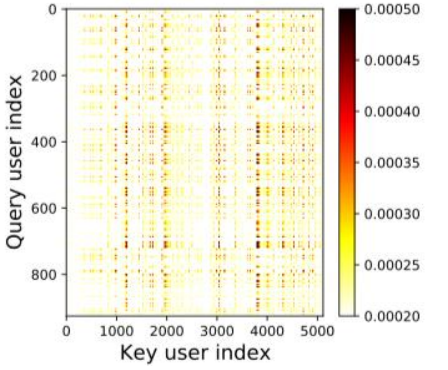
$$O\left(2LHB\sqrt{\frac{2M_q \ln M_k}{T_q}} + \sqrt{\frac{ln(1/\delta)}{T_q}}\right). \tag{8}$$

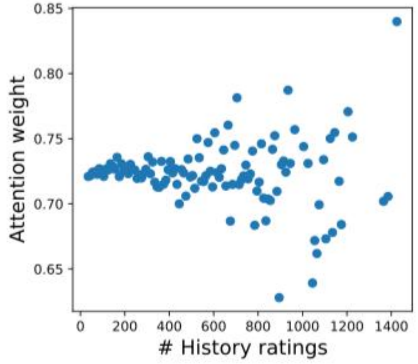
Qitian Wu, Hengrui Zhang, Xiaofeng Gao, Junchi Yan, Hongyuan Zha, "*Towards Open-World Recommendation: An Inductive Model-based Collaborative Filtering Approach*", in ICML'21

Experiment Results

- ❑ Interpolation for few-shot users: competitive as Oracle models
- ❑ Extrapolation for zero-shot users: significantly outperform SOTA inductive models

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, Hengrui Zhang, Xiaofeng Gao, Junchi Yan, Hongyuan Zha, "*Towards Open-World Recommendation: An Inductive Model-based Collaborative Filtering Approach*", in ICML'21

Attribute Feature Learning

- **General problem:** learn a **mapping** from input features to labels
  - Input data $\mathbf{x} = [x_1, x_2, \cdots, x_d]$ where x_i denotes the i-th input feature
  - Assume a prediction model $f : \mathbf{x} \rightarrow y$ and objective

$$f^* = \arg\min_f \mathbb{E}_{(\mathbf{x}, y)\sim D}[l(f(\mathbf{x}), y)]$$

- **Applications**
  - Tabular data: weather/income/usage prediction, disease diagnosis…
  - Real systems: recommendation, advertisement, question answering…

Scenario 1: Predict a person's income with aga/occ/edu

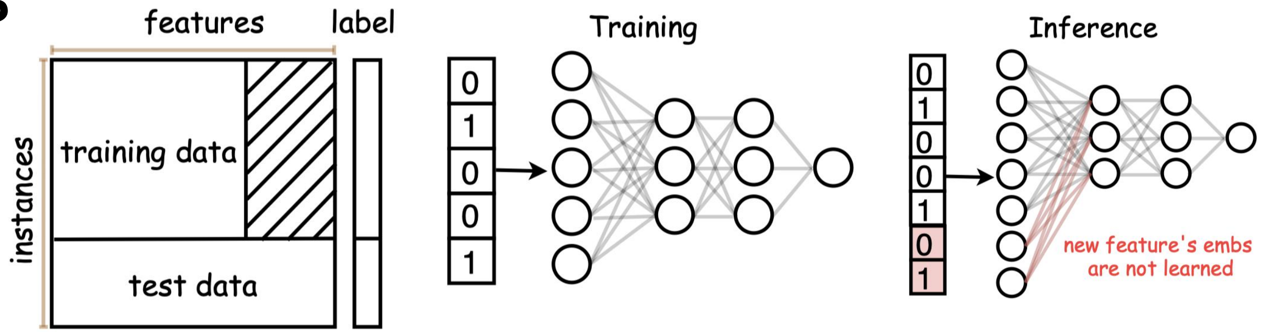
	age	occ	edu	income
o_1	x_{11}	x_{12}	x_{13}	y_1
o_2	x_{21}	x_{22}	x_{23}	y_2
o_3	x_{31}	x_{32}	x_{33}	?
…		…		

Scenario 2: Predict whether a user would click an item with attributes

user features: age/gender…
item features: category/price…

Problem of Feature Extrapolation

- Limitations for neural networks
  - Retraining from scratch
    - Issues: time consuming
  - Incremental learning
    - Issues: over-fitting



- 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 $\longrightarrow$ $\boxed{\mathcal{X}_{tr} \subset \mathcal{X}_{te}}$

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

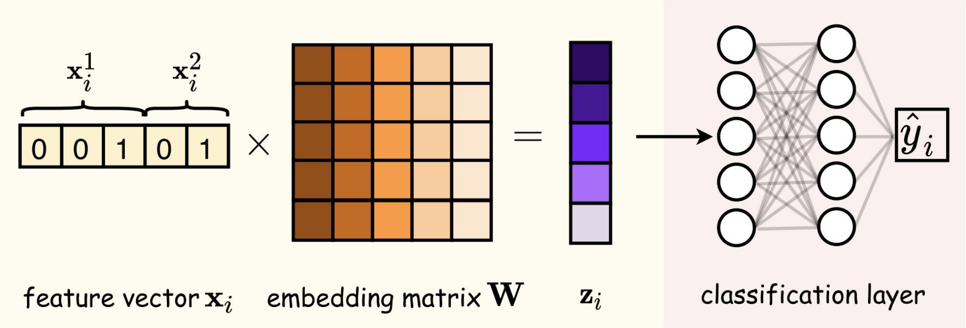
- Two cases causing feature space expansion:
  1) new raw features come, 2) unseen feature values out of known range

Key Observation 1: Permutation-Invariance

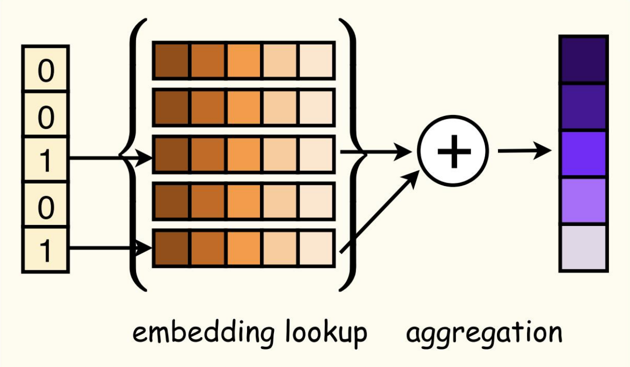
- ❑ Neural networks can be **decomposed** into two parts

$$\hat{y}_i = h(\mathbf{x}_i; \phi, \mathbf{W})$$

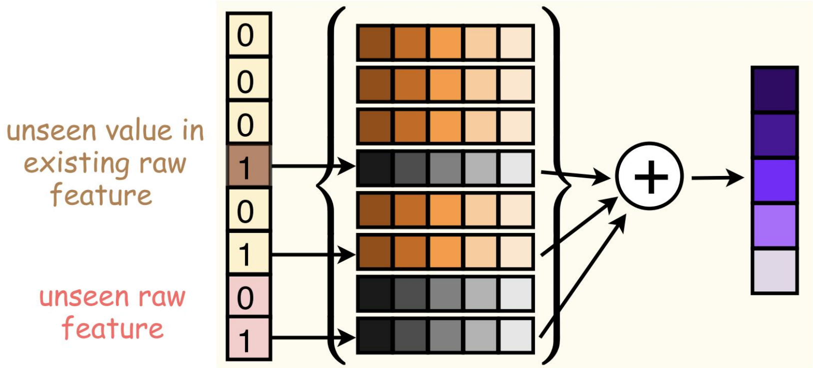
$$\Leftrightarrow \begin{cases} \mathbf{z}_i = \mathbf{W}\mathbf{x}_i \\ \hat{y}_i = \mathrm{FFN}(\mathbf{z}_i; \phi) \end{cases}$$



- ❑ **Equivalent view:** feature embedding look-up + embedding aggregation



Key insight:
The **permutation-invariance** property enables **variable-length** input features



Key Observation 2: Feature-Data Graph

- 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 $\mathbf{X}_{tr}$

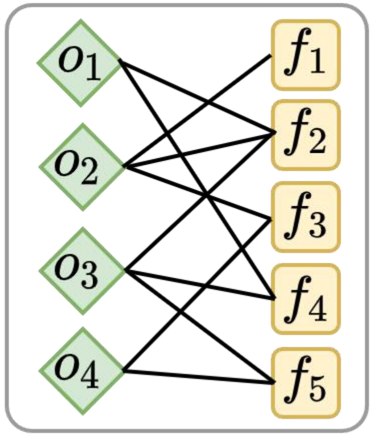
Advantage of graph representation:
1) Variable-size for features/instances
2) Missing values are allowed

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

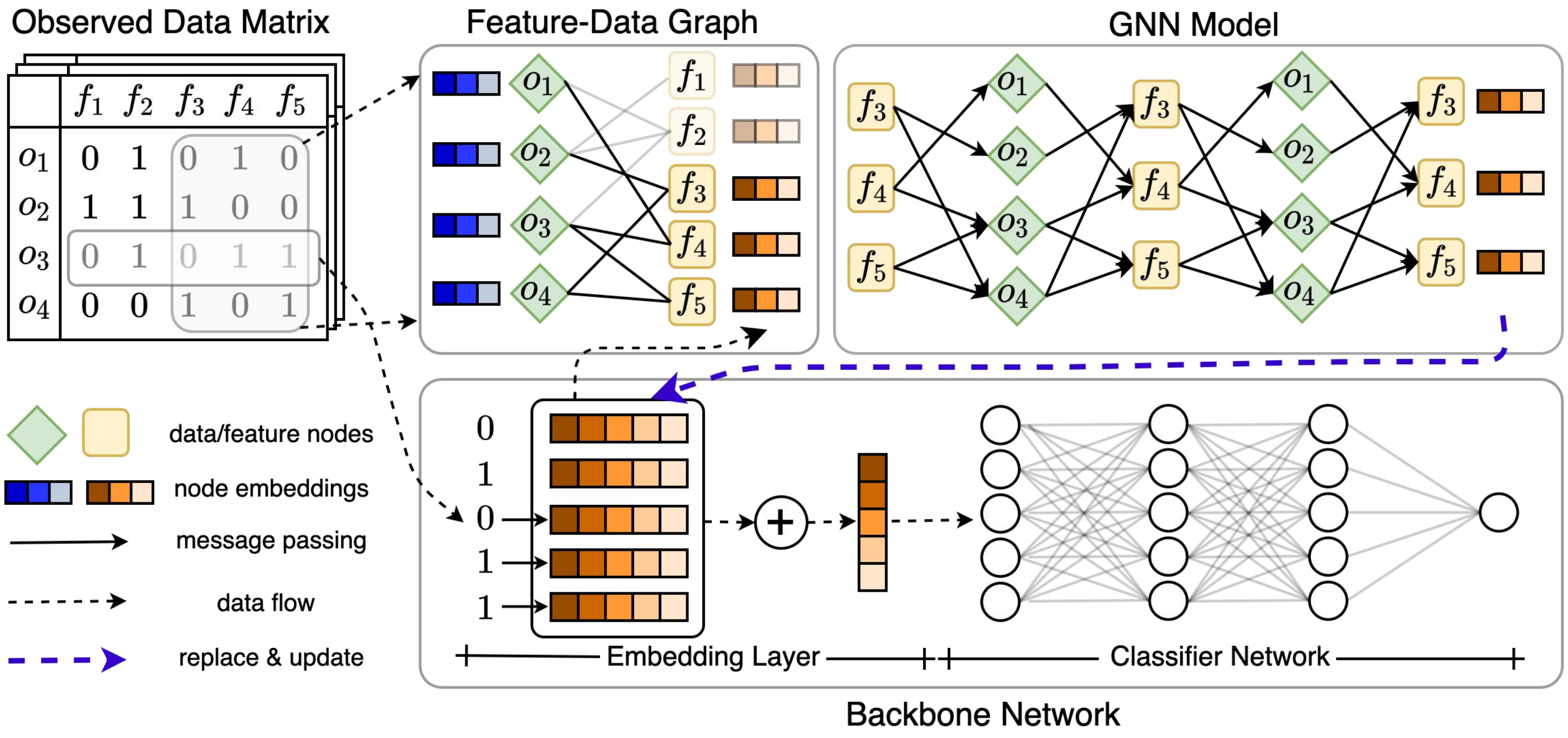
Observed Data Matrix

	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



Feature Extrapolation Network



Reasoning

High-level model for feature emb learning

Prediction

Low-level model for prediction

Qitian Wu, Chenxiao Yang, Junchi Yan, "Towards Open-World Feature Extrapolation: An Inductive Graph Learning Approach", in NeurIPS'21

Details for Proposed Model

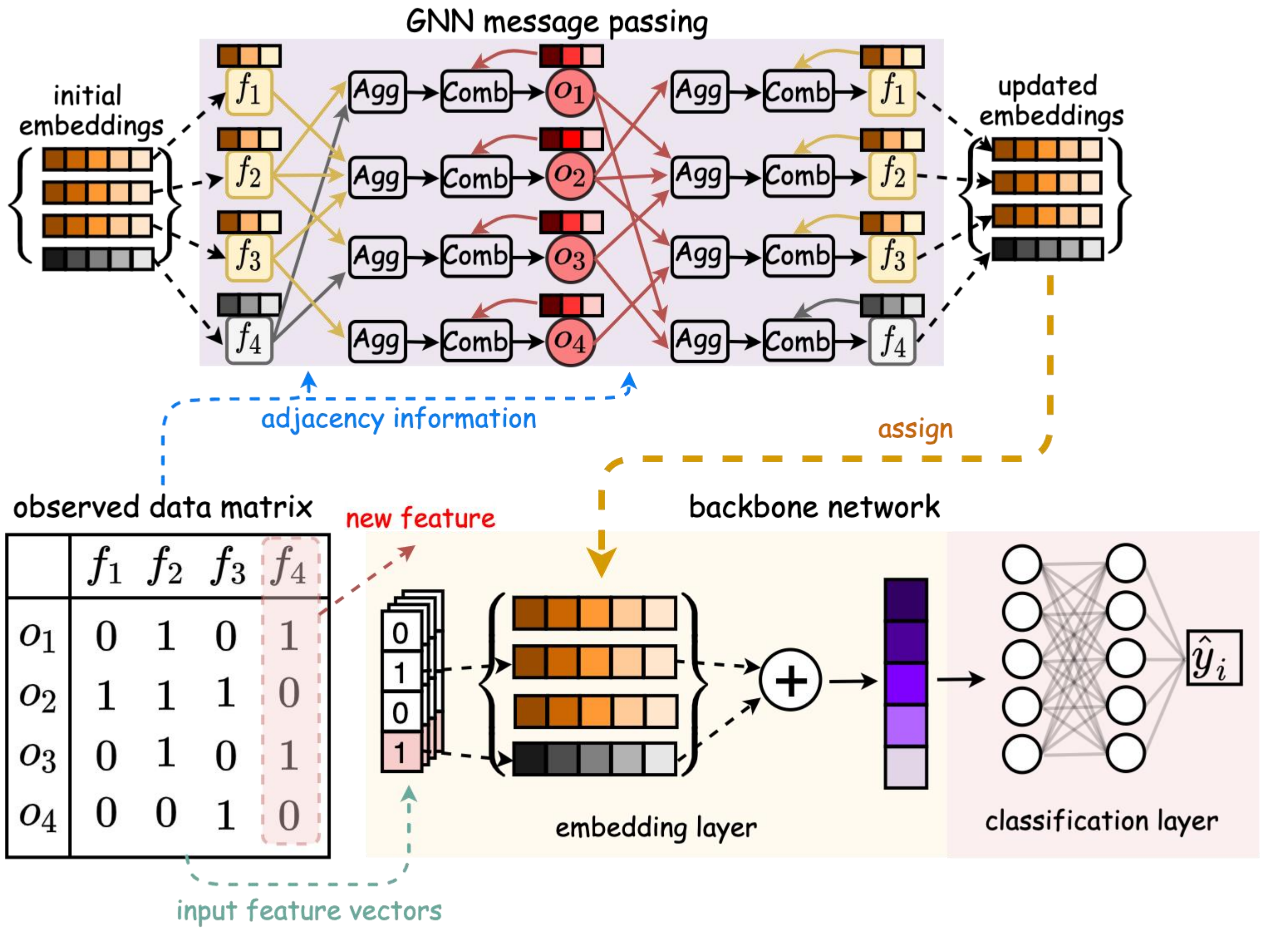
- **GNN model feedforward**
  - Feature nodes $\{\mathbf{w}_j\}_{j=1}^{D}$ (initial embeddings as $\mathbf{w}_j^{(0)}$)
  - Instance nodes $\{\mathbf{s}_i\}_{i=1}^{N}$ (initial states $\mathbf{s}_i^{(0)} = \mathbf{0}$)
  - Message passing rule:

$$\mathbf{a}_i^{(l)} = \textsc{Agg}(\{\mathbf{w}_k^{(l-1)} | \forall k, x_{ik} = 1\})$$

$$\mathbf{s}_i^{(l)} = \mathbf{P}^{(l)} \textsc{Comb}\left(\mathbf{s}_i^{(l-1)}, \mathbf{a}_i^{(l-1)}\right)$$

$$\mathbf{b}_j^{(l)} = \textsc{Agg}(\{\mathbf{s}_k^{(l-1)} | \forall k, x_{jk} = 1\})$$

$$\mathbf{w}_j^{(l)} = \mathbf{P}^{(l)} \textsc{Comb}\left(\mathbf{w}_j^{(l-1)}, \mathbf{b}_j^{(l-1)}\right)$$



Qitian Wu, Chenxiao Yang, Junchi Yan, "Towards Open-World Feature Extrapolation: An Inductive Graph Learning Approach", in NeurIPS'21

Details for Proposed Model

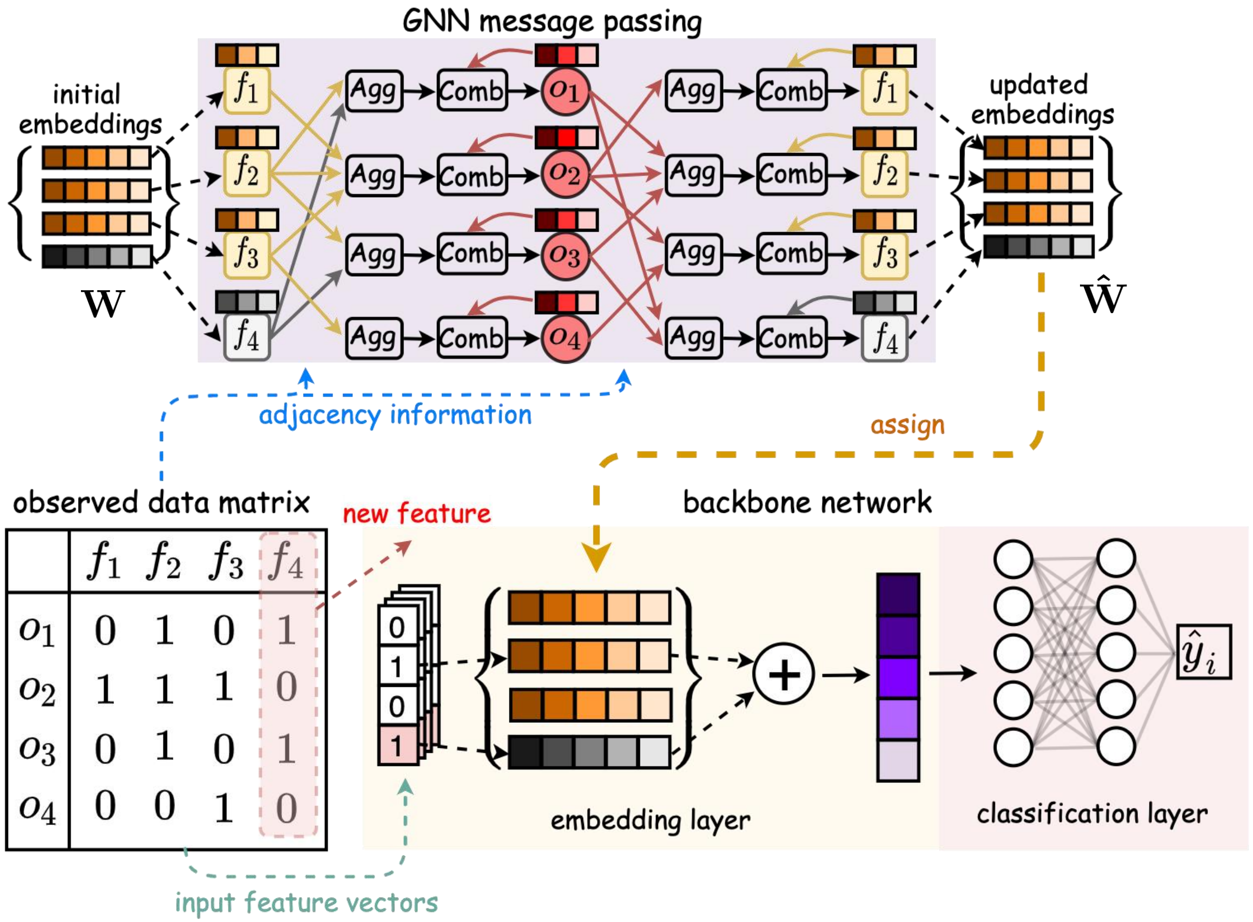
- **Entire feedforward compute**
  - Query feature embeddings
    - For old features: $\mathbf{W}$
    - For new features: set as zero
  - Updata feature embeddings

$$\hat{\mathbf{W}} = [\mathbf{w}_j^{(L)}]_{j=1}^{D} = g(\mathbf{W}, \mathbf{X}; \omega)$$

  - Assign to backbone and output predicted results

$$\hat{y}_i = h(\mathbf{x}_i; \phi, \hat{\mathbf{W}})$$

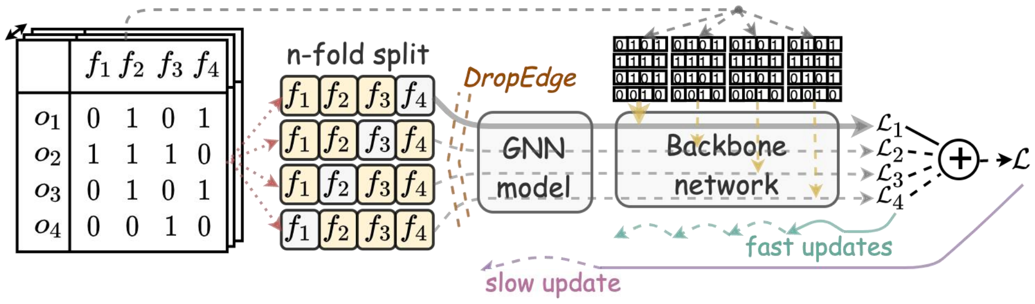
Note: 1) $\mathbf{X}$ can be either training or test data; 2) the permutation-invarance and graph representation enables arbitrarily sized $\mathbf{X}$



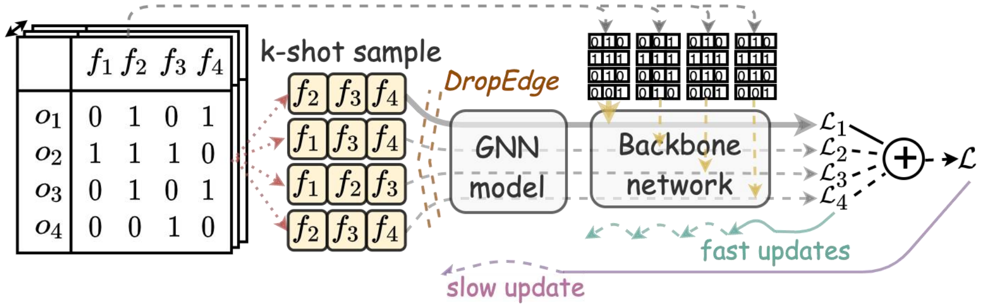
Qitian Wu, Chenxiao Yang, Junchi Yan, “Towards Open-World Feature Extrapolation: An Inductive Graph Learning Approach”, in NeurIPS'21

Feature Extrapolation Network: Training

- Two useful techniques for learning to extrapolate
  - Proxy training data
    - Self-supervised learning:
      n-fold splitting input features
    - Inductive learning:
      k-shot sampling input features
  - Asynchronous Updates
    - Fast/slow for backbone/GNN
- DropEdge regularization
- Scaling to large systems
  - Time/space complexity $O(Bd)$



(a) Self-supervised learning with n-fold splitting



(b) Inductive learning with k-shot sampling

Qitian Wu, Chenxiao Yang, Junchi Yan, "Towards Open-World Feature Extrapolation: An Inductive Graph Learning Approach", in NeurIPS'21

Experiments on UCI Datasets

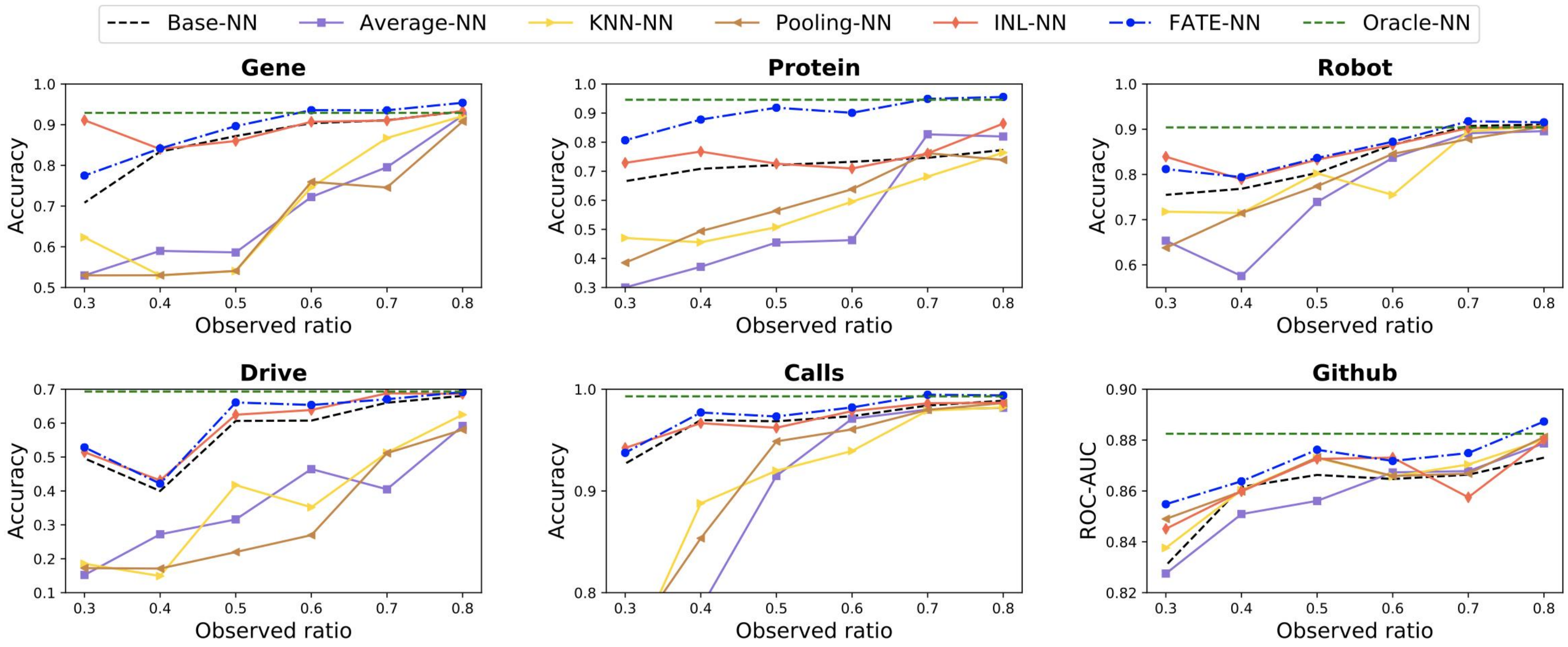


Figure. Accuracy/ROC-AUC results w.r.t. different ratios for observed features

- ❑ **FATE (ours) yields 7.3% higher acc. than Base (without using new features)**
- ❑ **FATE produces 29.8% higher acc. than baselines Avg, KNN, Pooling**

Qitian Wu, Chenxiao Yang, Junchi Yan, "Towards Open-World Feature Extrapolation: An Inductive Graph Learning Approach", in NeurIPS'21

Experiments 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**	$\mathbf{0.710} \pm 0.010$
	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**	$\mathbf{0.713} \pm 0.010$
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**	$\mathbf{0.772} \pm 0.001$
	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**	$\mathbf{0.783} \pm 0.001$

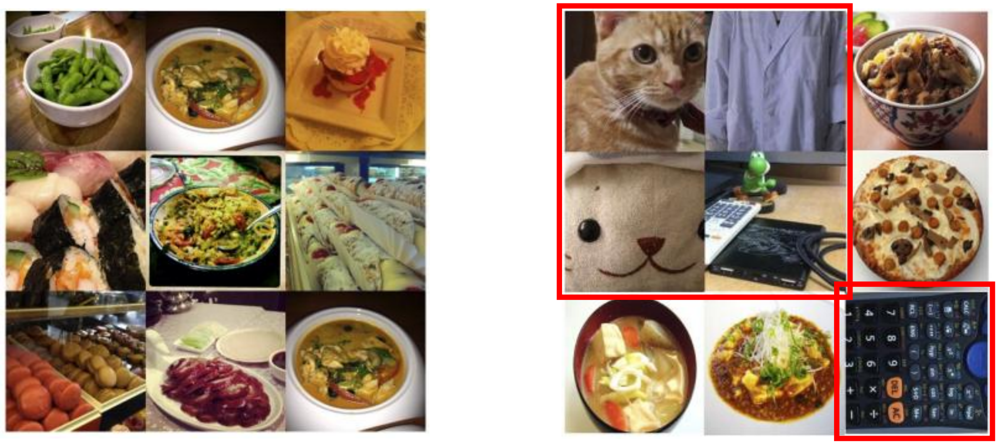
- FATE achieves significantly improvements over Base/Pooling with different backbones (DNN and DeepFM[1])

Qitian Wu, Chenxiao Yang, Junchi Yan, "Towards Open-World Feature Extrapolation: An Inductive Graph Learning Approach", in NeurIPS'21

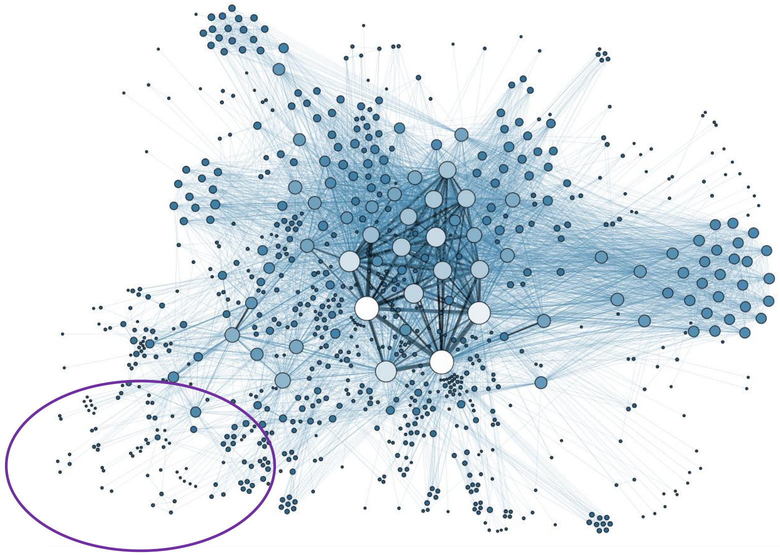
[1] H. Guo, R. Tang, Y. Ye, Z. Li, and X. He. Deepfm: A factorization-machine based neural network for CTR prediction. In International Joint Conference on Artificial Intelligence, 2017.

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?

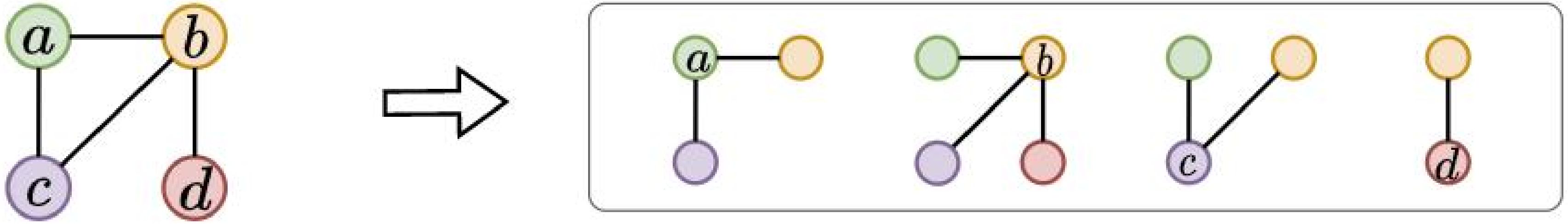
Problem Formulation

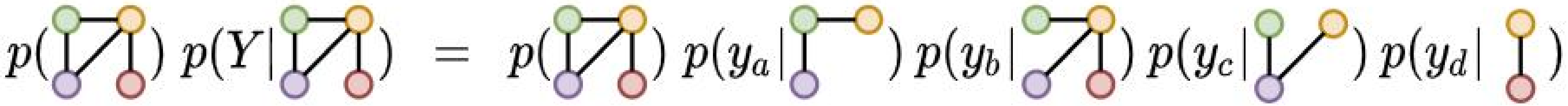
- **Graph notation:** A graph $G = (A, X)$, adjacency matrix $A = \{a_{uv} | v, u \in V\}$ node features $X = \{x_v | v \in V\}$, node labels $Y = \{y_v | v \in V\}$

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

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 e denotes environment (that affects data generation)

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

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

  - 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

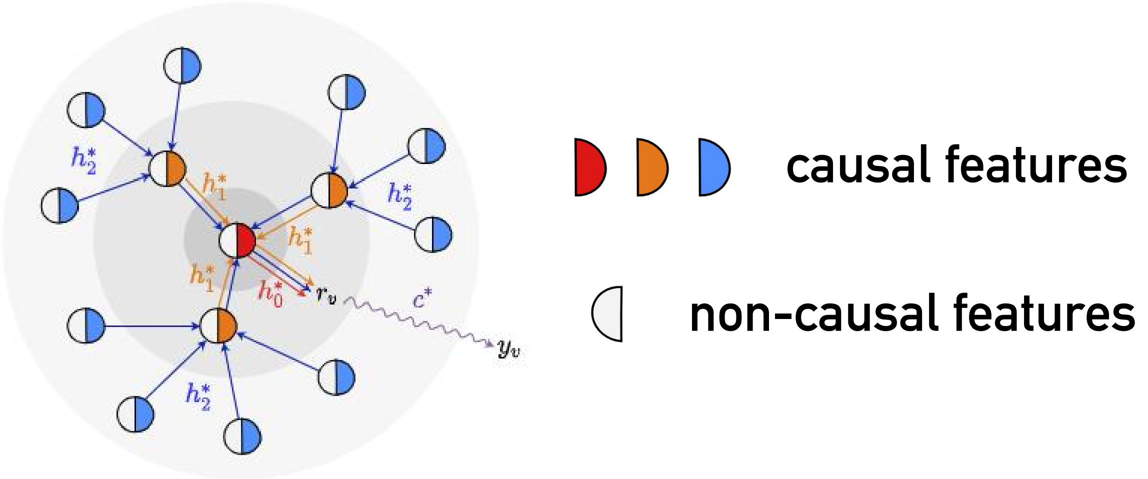
inspired by Weisfeiler-Lehman test

- *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

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

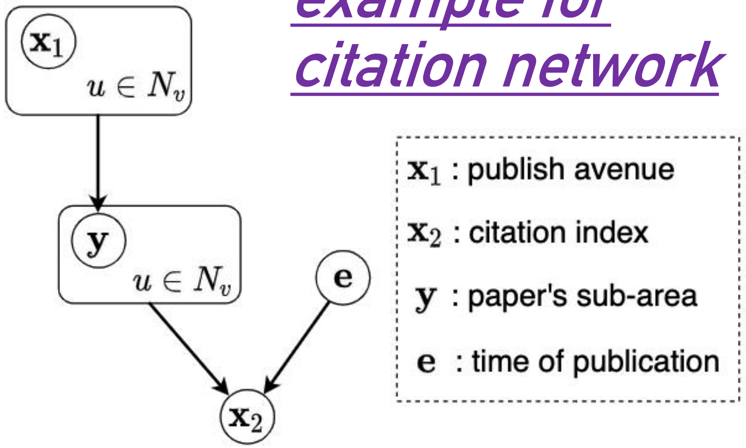
where n_v^1 and n_v^2 are standard normal noise and ϵ is a random variable with zero mean and non-zero variance dependent on the environment.

Model: a vanilla GCN as the predictor model:

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

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

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



example for citation network

Theoretical Motivation

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

Qitian Wu, Hengrui Zhang, Junchi Yan, and David Wipf, "Handling Distribution Shifts on Graphs: An Invariance Perspective", in ICLR'22

Explore-to-Extrapolate Risk Minimization

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

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

where $L(G^e, Y^e; \theta) = \frac{1}{|V_e|} \sum_{v \in V_e} l(f_\theta(G_v^e), y_v^e)$ and β is a trading hyper-parameter.

Key issue: no/ambiguous environment in observed data

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

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

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

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

Explore-to-Extrapolate Risk Minimization

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

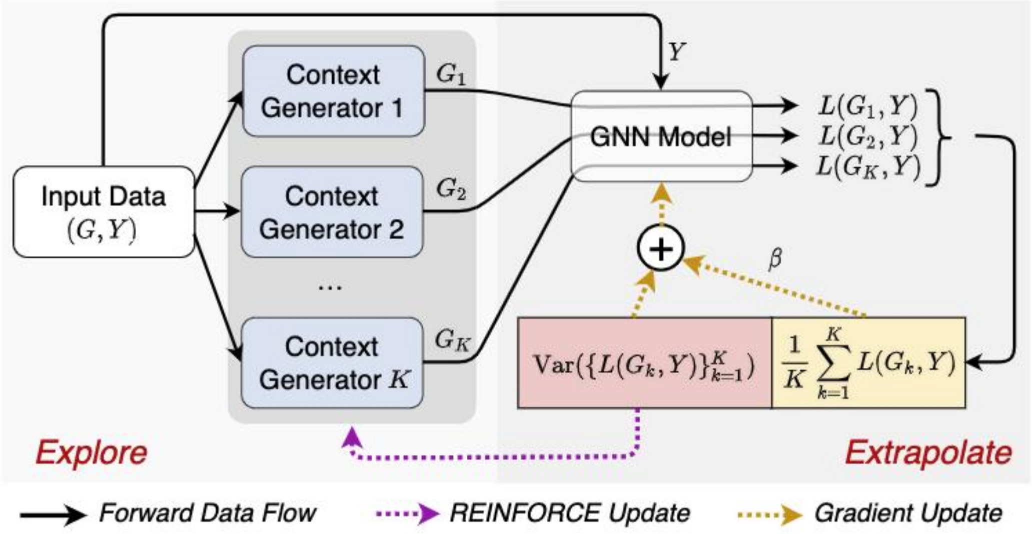
$$\text{s. t. } [w_1^*, \cdots, w_K^*] = \boxed{\arg\max_{w_1, \cdots, w_K} \mathrm{Var}(\{L(g_{w_k}(G), Y; \theta) : 1 \le k \le K\})}$$

adversarially train multiple data generators

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

❑ 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



Qitian Wu, Hengrui Zhang, Junchi Yan, and David Wipf, "Handling Distribution Shifts on Graphs: An Invariance Perspective", in ICLR'22

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 $p(\mathbf{y}|\mathbf{z}, \mathbf{e}) = p(\mathbf{y}|\mathbf{z})$.

Qitian Wu, Hengrui Zhang, Junchi Yan, and David Wipf, "Handling Distribution Shifts on Graphs: An Invariance Perspective", in ICLR'22

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)

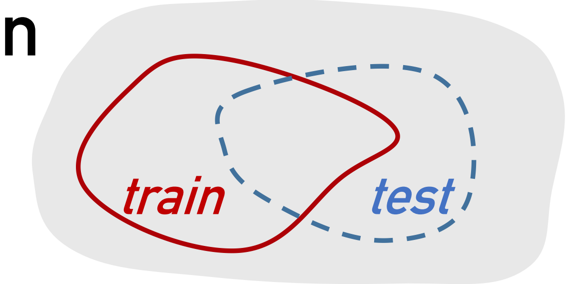
Optimizing the learning objective with training data can minimize the upper bound for OOD error measured by $D_{KL}(p_{e'}(\mathbf{y}|\mathbf{G_v})\|q(\mathbf{y}|\mathbf{G_v}))$ on condition that $I_{e'}(\mathbf{G_v};\mathbf{y}|\mathbf{z}) = I_e(\mathbf{G_v};\mathbf{y}|\mathbf{z})$.

Qitian Wu, Hengrui Zhang, Junchi Yan, and David Wipf, "Handling Distribution Shifts on Graphs: An Invariance Perspective", in ICLR'22

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

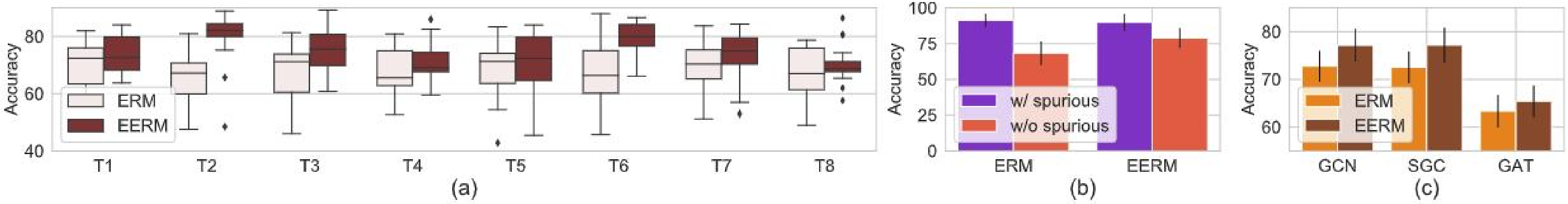


Figure. Experiment results on Cora with artificial spurious features. (a) Test accuracy on eight testing graphs (with different environment ids). (b) Training accuracy during inference w/ and w/o using spurious features. (c) Averaged test accuracy using different GNNs for synthetic data generation.

- 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

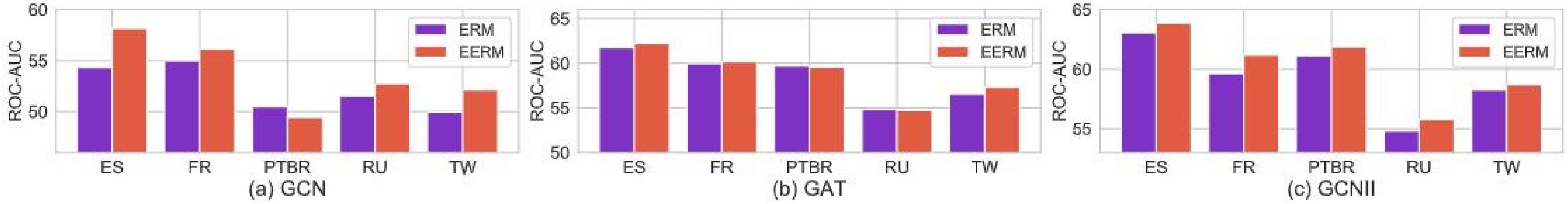


Figure. ROC-AUC results on Twitch-Explicit when training on one graph and testing on others with different GNN predictors (GCN, GAT and GCNII)

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

Qitian Wu, Hengrui Zhang, Junchi Yan, and David Wipf, “Handling Distribution Shifts on Graphs: An Invariance Perspective”, in ICLR'22

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 evolving graph (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

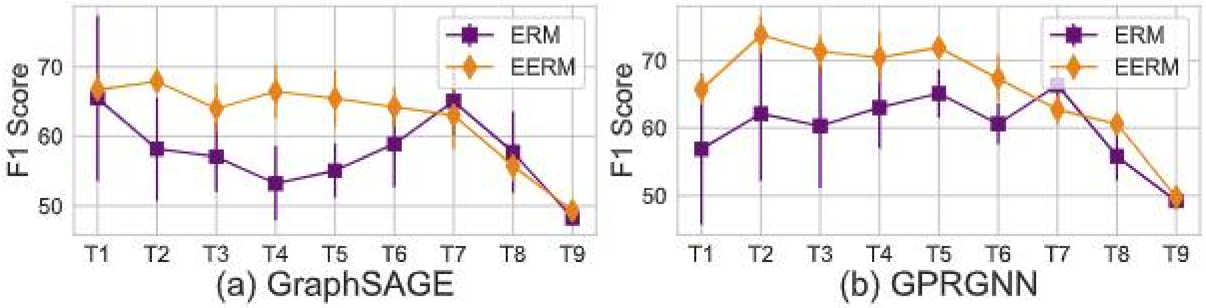


Figure. F1 score results on Elliptic with dynamic graph snapshots (chronologically divided into 9 test groups)

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

Qitian Wu, Hengrui Zhang, Junchi Yan, and David Wipf, “Handling Distribution Shifts on Graphs: An Invariance Perspective”, in ICLR'22

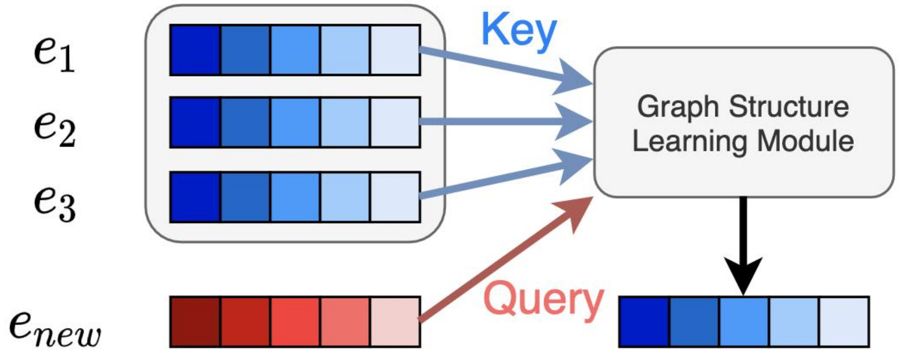
Conclusions

- The main ideas of *open-world recommendation* [ICML'21]:

  Inductive Collaborative Filtering (IDCF)

  1) partition entities into two groups

  2) learn a latent graph among entities and compute new entities' embeddings using those of existing ones



- Potential applications:
  - For out-of-graph learning extrapolation, e.g. in knowledge graphs
  - Transferring embeddings from well-trained entities to long-tail ones
  - Knowledge tranfer in multi-task/multi-label learning

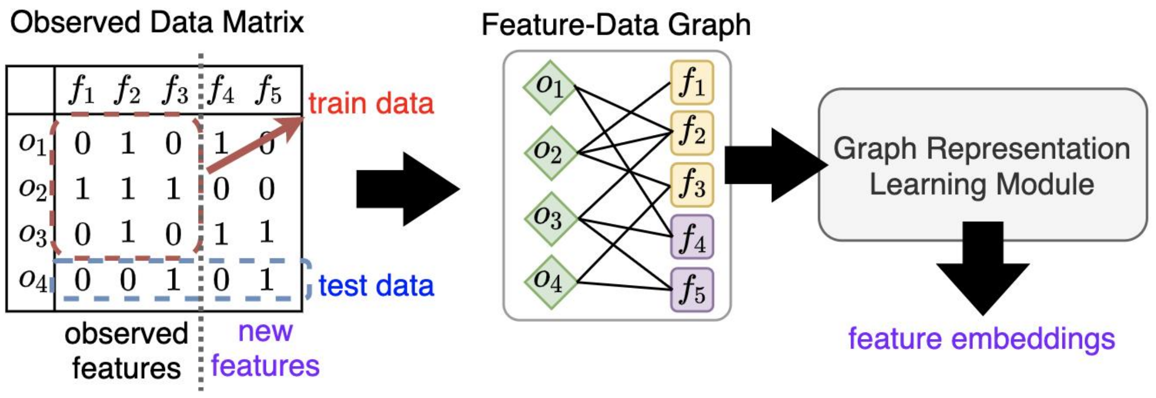
Conclusions

- ❑ The main ideas of *open-world feature extrapolation* [NeurIPS'21]:

  Feature Extrapolation Networks (FATE)

  1) instance-feature matrix as a graph

  2) convert feature embedding learning to graph representation learning (extrapolation via message passing)



- ❑ Potential applications:
  - New attribute features for question answering and reasoning (NLP)
  - Information from new sensors for robot learning and decisions (Robot)
  - Extra annotation features for image learning and understanding (Vision)
  - ……

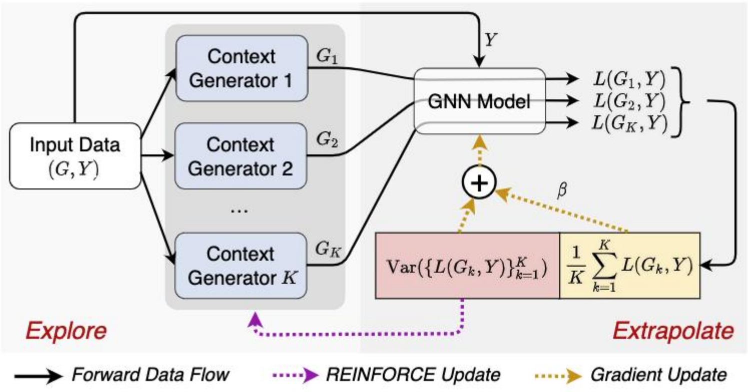
Conclusions

- The main ideas of *graph out-of-distribution generalization* [ICLR'21]:

  Explore-to-Extrapolate Risk Minimization (EERM)

  1) data augmentation from training data to maximize environment variance

  2) training model predictor to minimize the mean and variance of risks



- Potential future works:
  - Extrapolation from single observed environment
  - Handling observed data without correspondence to specific environments
  - Inferring heterogenous environment from graph data

References

[1] Qitian Wu, Hengrui Zhang, Xiaofeng Gao, Junchi Yan, Hongyuan Zha, Towards Open-World Recommendation: An Inductive Model-Based Collaborative Filtering Approach. International Conference on Machine Learning (ICML'21)

- Code: https://github.com/qitianwu/IDCF

[2] Qitian Wu, Chenxiao Yang, Junchi Yan, Towards Open-World Feature Extrapolation: An Inductive Graph Learning Approach, Advances in Neural Information Processing Systems (NeurIPS'21)

- Code: https://github.com/qitianwu/FATE

[3] Qitian Wu, Hengrui Zhang, Junchi Yan, David Wipf, Handling Distribution Shifts on Graphs: An Invariance Perspective. International Conference on Learning Representations (ICLR'22)

- Code: https://github.com/qitianwu/GraphOOD-EERM

Thanks for listening!

contact: echo740@sjtu.edu.cn