Fair & Disentangle Graph Mining Papers Organization

CAU
Junseo, Yu

DMAIS Lab Meeting
12.06.2024

Contents

Weekly Meetings

1. Fairness

- Fairness notion
- Fairwalk
- Crosswalk
- InFoRM

Fairness

Fairness notion

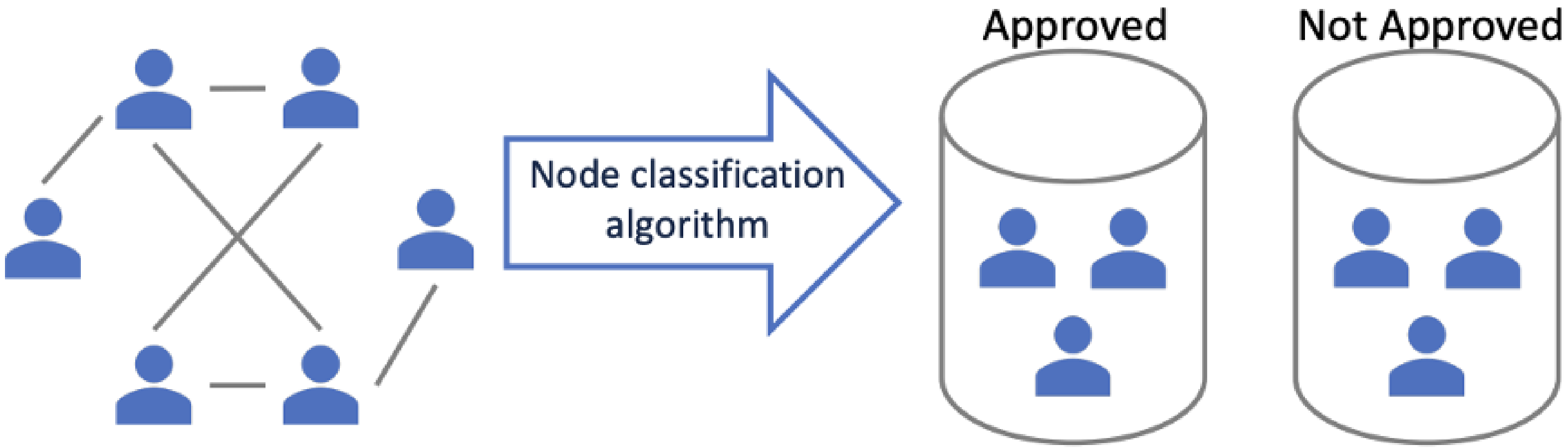
Fairness

- The quality of treating people **equally** or in a way that is **right** or **reasonable**.
- How do we **define** the **equally, right, or reasonable**?
  - It might be the problem of the **philosophy, ethics**, or/and **sociology**
  - It continues to change over time. (e.g., **Golden Rule**)

Fairness

Fairness notion

Fairness in Machine Learning(AI)



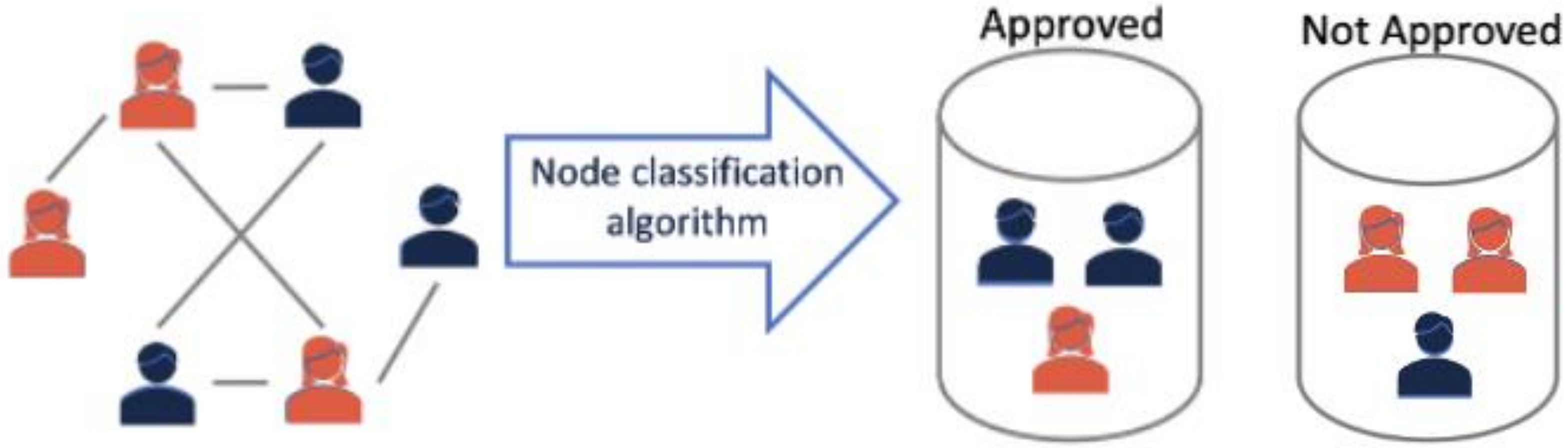
Loan Approval

- ❑ The model try to classify the node(people) by using various techniques
- ❑ However, without considering fairness, the model can **learn in an unfair way**
- ❑ **Potential Cause**: Biased Data, Spurious Correlation, Biased learning strategy, and others

Fairness

Fairness notion

Fairness in Machine Learning(AI)



Loan Approval

- ❑ The upper Scenario **might** be **unfair**.
- ❑ Since Male has a higher approval rate than female.
- ❑ However, is it **absolute**? If not, how we can measure **the (un)fairness level**?

Fairness

Fairness notion

Fairness in Machine Learning(AI)

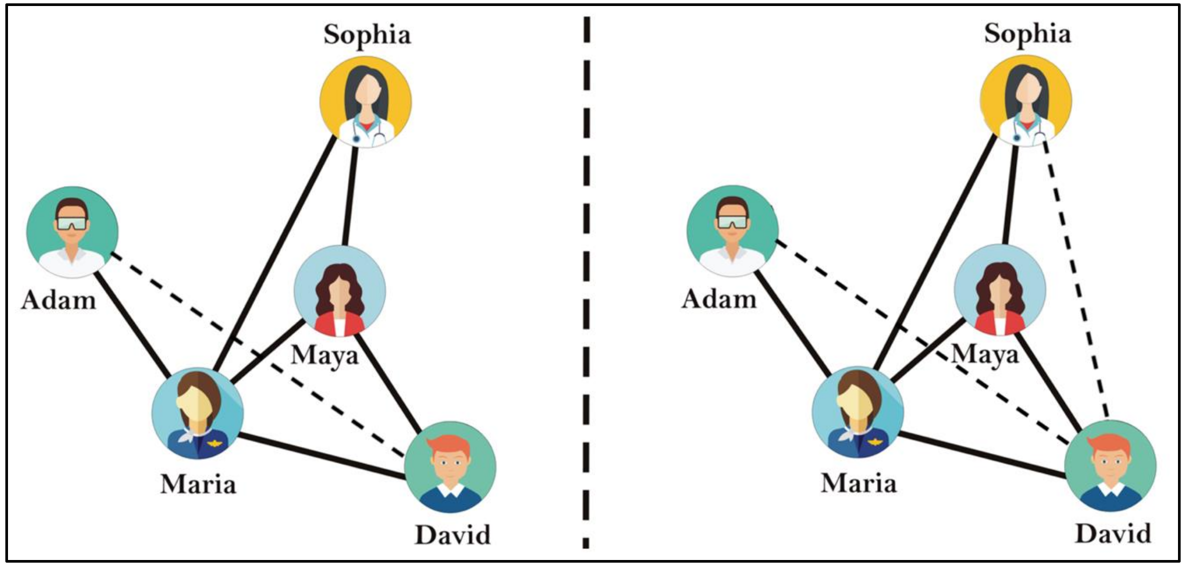

REPORT \ TECH \ ARTIFICIAL INTELLIGENCE

What a machine learning tool that turns Obama white can (and can't) tell us about AI bias

A striking image that only hints at a much bigger problem

By James Vincent | Jun 23, 2020, 3:45pm EDT

Mosaic removal model



Link Prediction in SNS

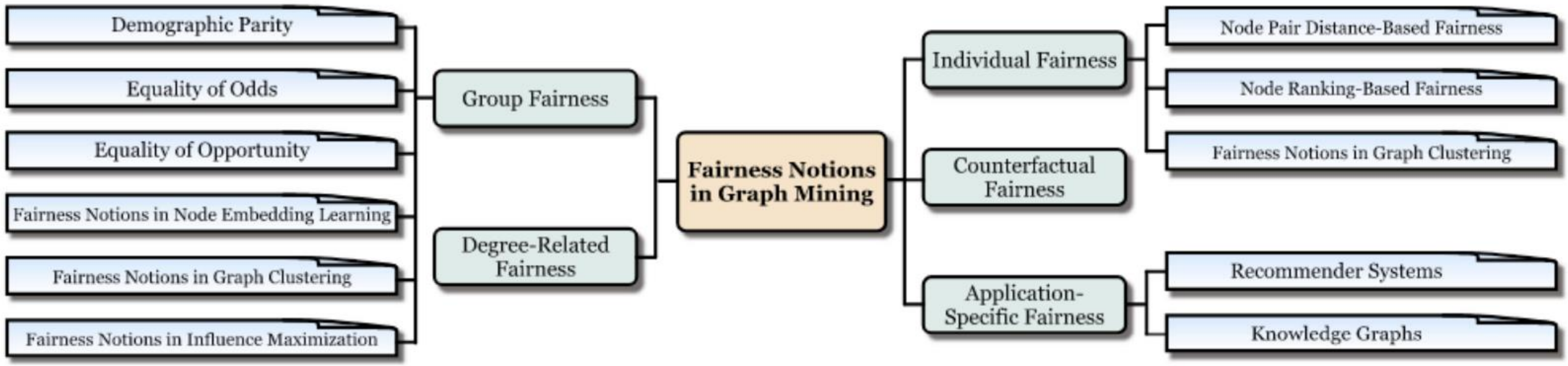
Dashed line : Prediction

Bold line : Actual Line

Fairness

Fairness notion

Fairness Metrics



- **Group Fairness**
  - DP, Demographic Parity
  - EO, Equality of Opportunity
- **Individual Fairness**

Fairness

Fairness notion

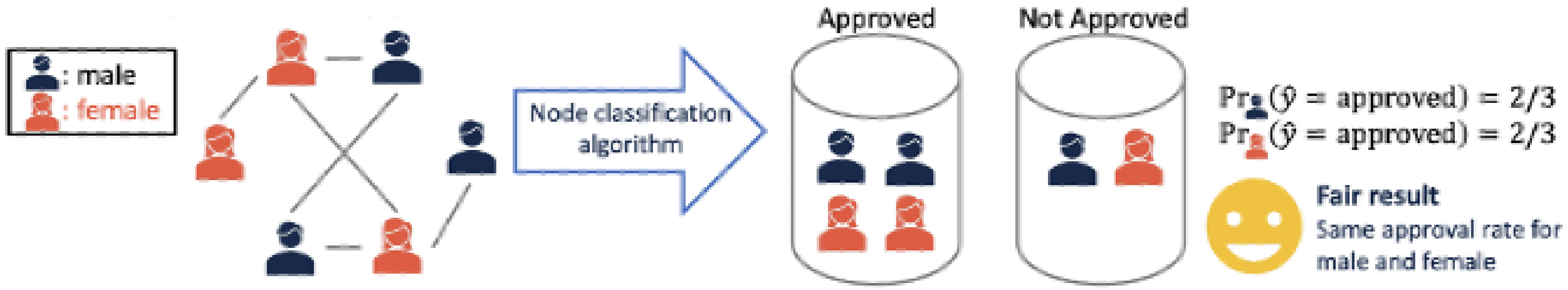
Fairness Metrics

Group Fairness – DP, Demographic Parity

$$\Pr_{+}(\hat{y} = c) = \Pr_{-}(\hat{y} = c)$$

- $\Pr_{+}$: probability for the protected group
- $\Pr_{-}$: probability for the unprotected group

➔ All groups should have an **equal** positive(acceptance) rate



Fairness

Fairness notion

Fairness Metrics

Group Fairness – EO, Equality of Opportunity

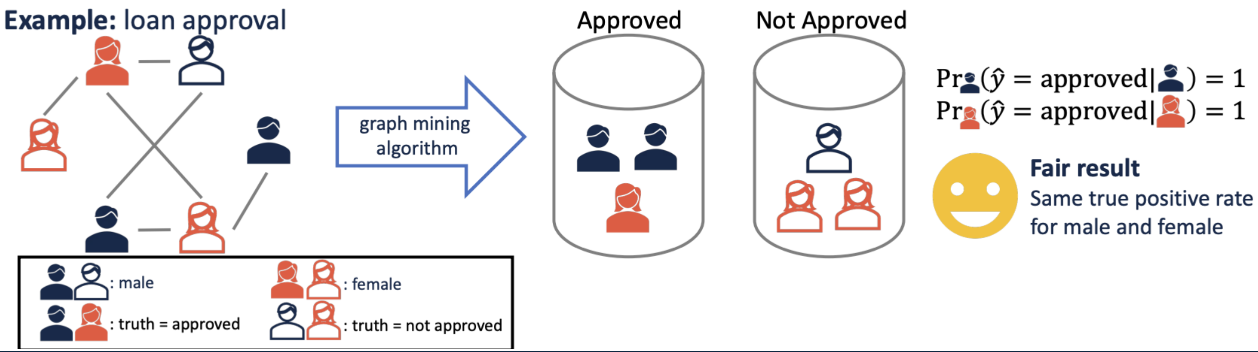
$$\Pr_{+}(\hat{y} = c | y = c) = \Pr_{-}(\hat{y} = c | y = c)$$

Group Fairness – DP, Demographic Parity

$$\Pr_{+}(\hat{y} = c) = \Pr_{-}(\hat{y} = c)$$

- $\Pr_+$: probability for the protected grou
- $\Pr_-$: probability for the unprotected group

➔ All groups should have an **equal** **TPR**(true positive rate) regardless of their protected attributes



Fairness

Fairness notion

Fairness Metrics

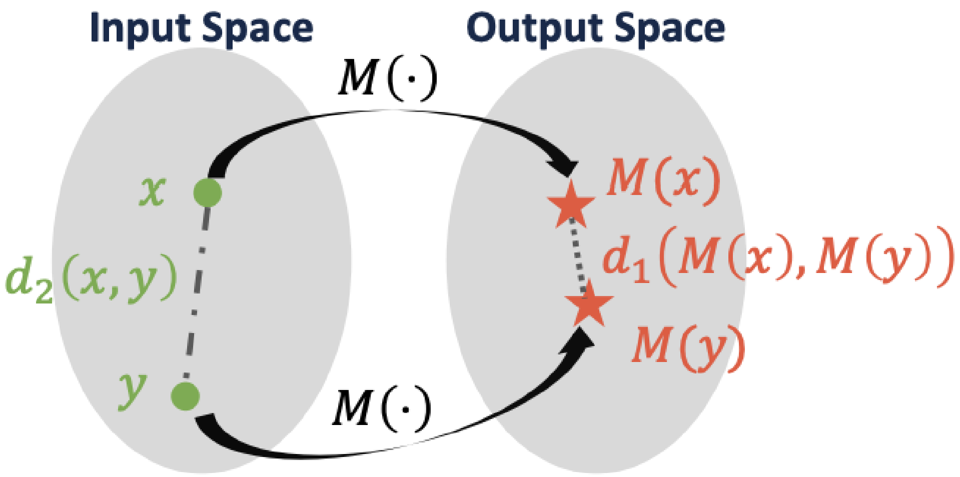
Individual Fairness

$$d_1(M(x), M(y)) \leq L d_2(x, y)$$

Lipschitz inequality

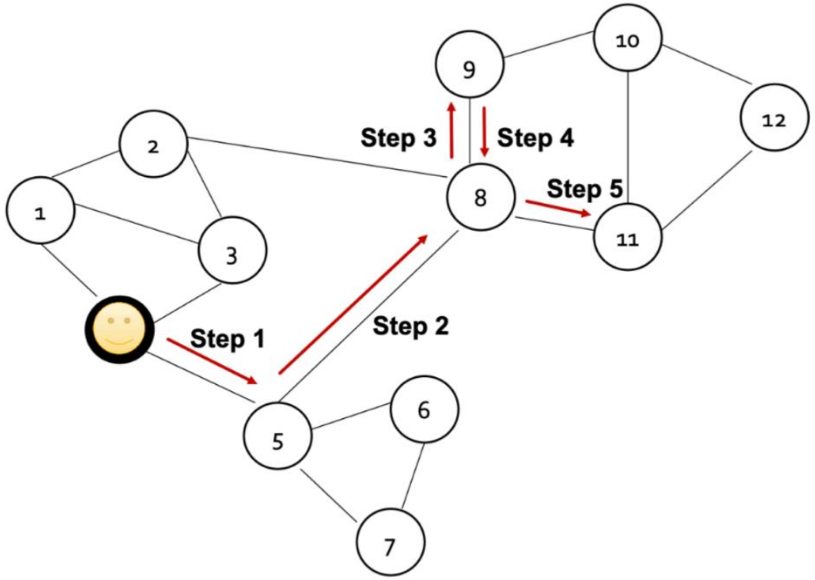
- **M** : a mapping from input to output
- **d_1** : distance metric for output
- **d_2** : distance metric for input
- **L** : a constant scala

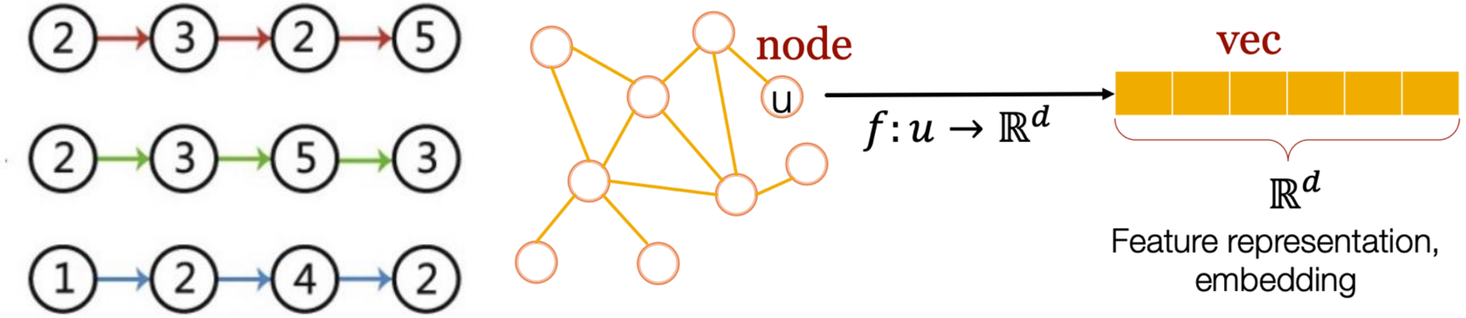
➔ **Similar individuals** should have **similar outcomes**



Fairness

Fairwalk (IJCAI' 19)





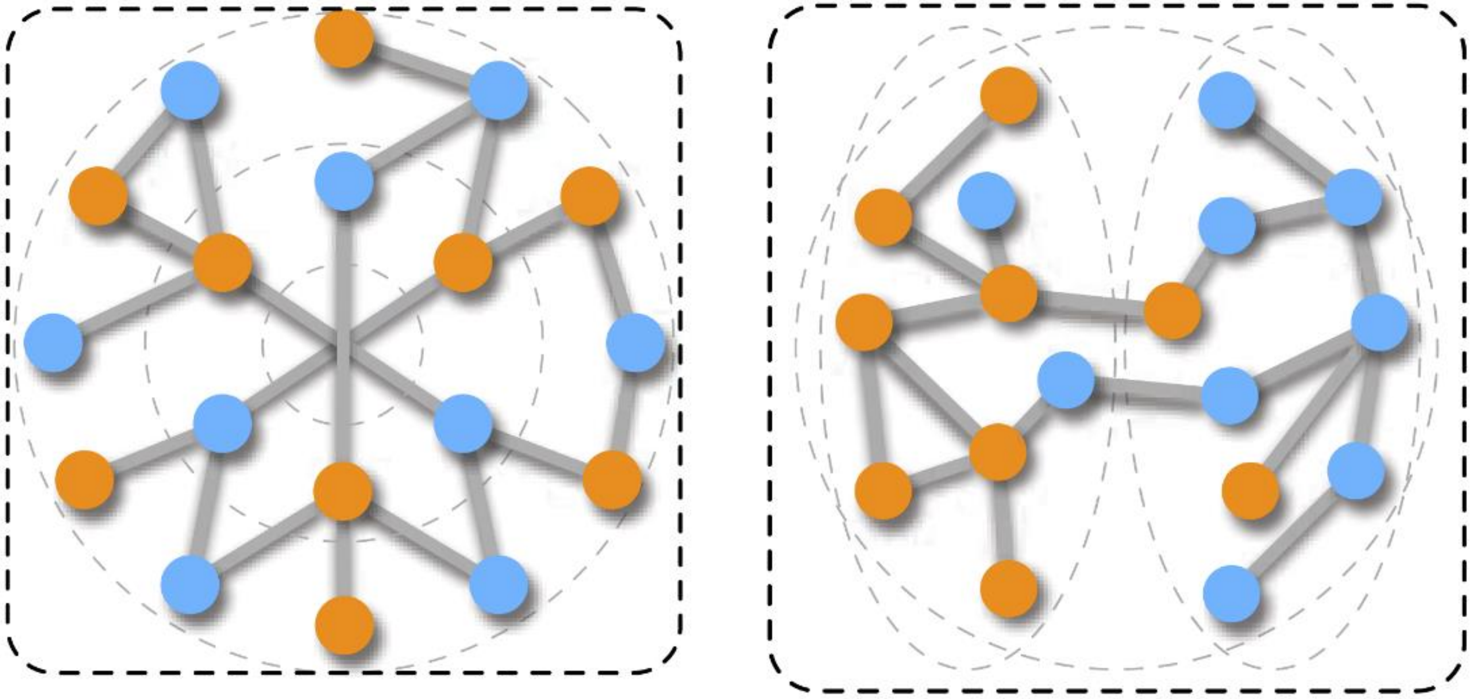
Random Walk

Goal : Capture **graph structure information** to generate **node embeddings**

Method: Visit neighbor node **randomly** and record the visiting order

Fairness

Fairwalk (IJCAI' 19)



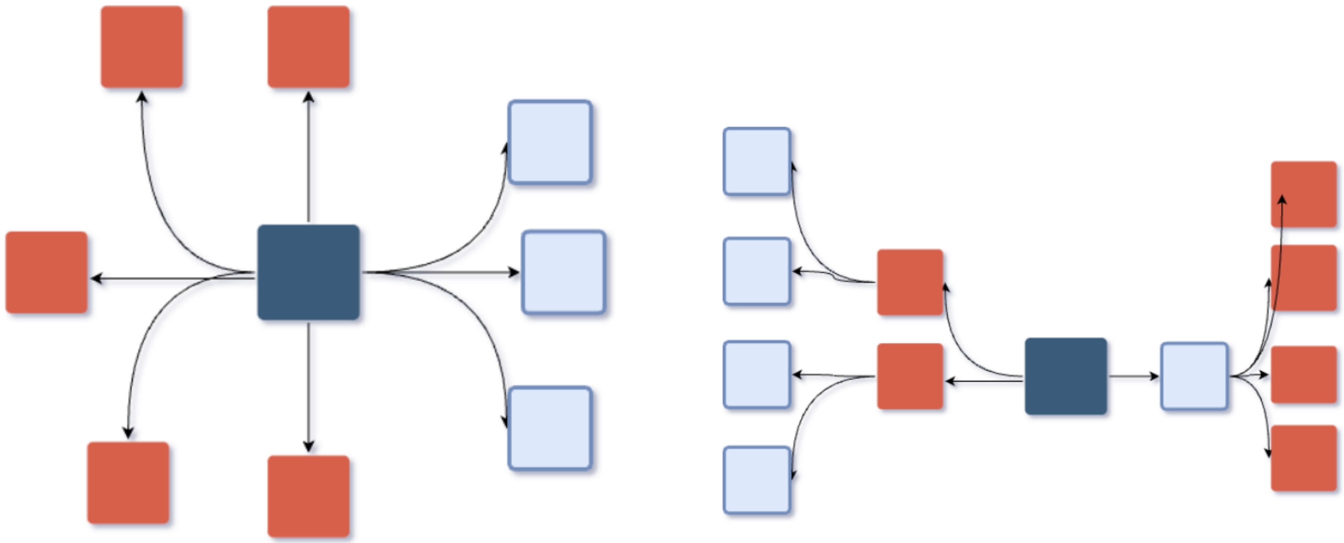
(a) Unbiased graph topology (b) Biased graph topology

Motivation:

In biased graph, random walk has a difficulty to obtain information of specific groups

Fairness

Fairwalk (IJCAI' 19)



Probability of next walk

- Red : 5/8 -> 1/2
- Blue: 3/8 -> 1/2

Methodology:

Modify random walk process

1. Partition neighbors into groups
2. Give each group the **same probability** of being chosen regardless of their size

Limitations:

- Can not capture the information beyond the one-hop

Fairness

CrossWalk (WWW' 23)

$$m(v) = \frac{\sum_{j\in[r]} \sum_{u\in\mathcal{W}_v^j} \mathbb{I}[l_v \neq l_u]}{r \times d}.$$

$$w'_{vu} = \begin{cases} w_{vu}(1-\alpha) \times \frac{m(u)^p}{\sum_{z\in N_v} w_{vz} m(z)^p} & \text{if } l_v = l_u \\ w_{vu}\alpha \times \frac{m(u)^p}{|R_v| \sum_{z\in N_v^c} w_{vz} m(z)^p} & \text{if } l_v \neq l_u = c. \end{cases}$$

Motivation:

- control more **elaborately** than fairwalk by hyper parameters
- consider fairness beyond one-hop nodes

Methodology :

Modify random walk process

- **More tendency** to the groups' peripheries and different groups

Fairness

CrossWalk (WWW' 23)

$$m(v) = \frac{\sum_{j\in[r]} \sum_{u\in\mathcal{W}_v^j} \mathbb{I}[l_v \neq l_u]}{r \times d}.$$

m(v): The fraction of other groups in random walk process

➔ **High** m(v) means that the node v is **close to other groups**

Example

Did 10 iterations with random walk of 10 lengths. (Total Visit: 100)

Assume, the node v meet 30 nodes of different group with v.

Then, m(v) = 80 / 100 = 0.8

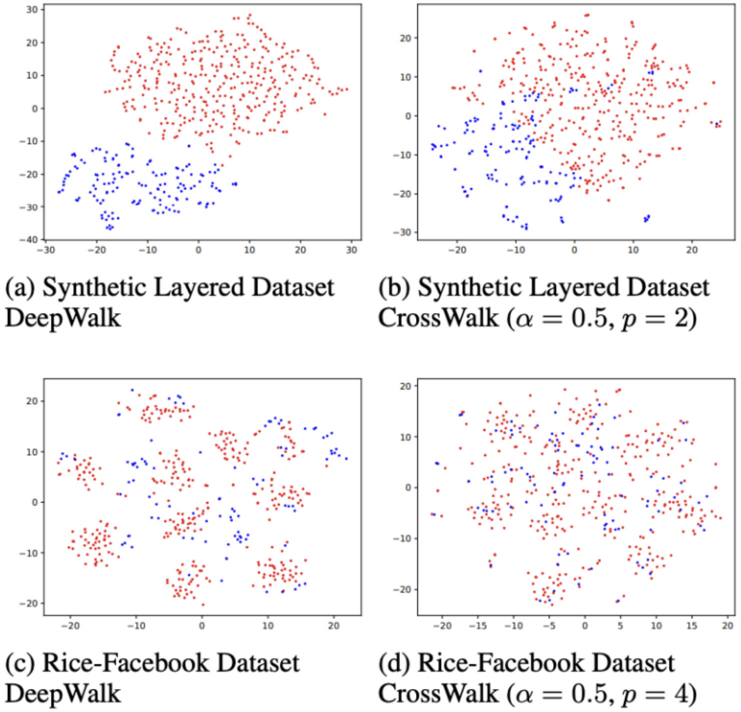
Fairness

CrossWalk (WWW' 23)

$$w'_{vu} = \begin{cases} w_{vu}(1-\alpha) \times \frac{m(u)^p}{\sum_{z \in N_v} w_{vz} m(z)^p} & \text{if } l_v = l_u \\ w_{vu}\alpha \times \frac{m(u)^p}{|R_v| \sum_{z \in N_v^c} w_{vz} m(z)^p} & \text{if } l_v \neq l_u = c. \end{cases}$$

➔ Based on m(v), **reweight** the edges to w_{uv}`

➔ Then, transfer probabilities in random walks based on reweighting stratey

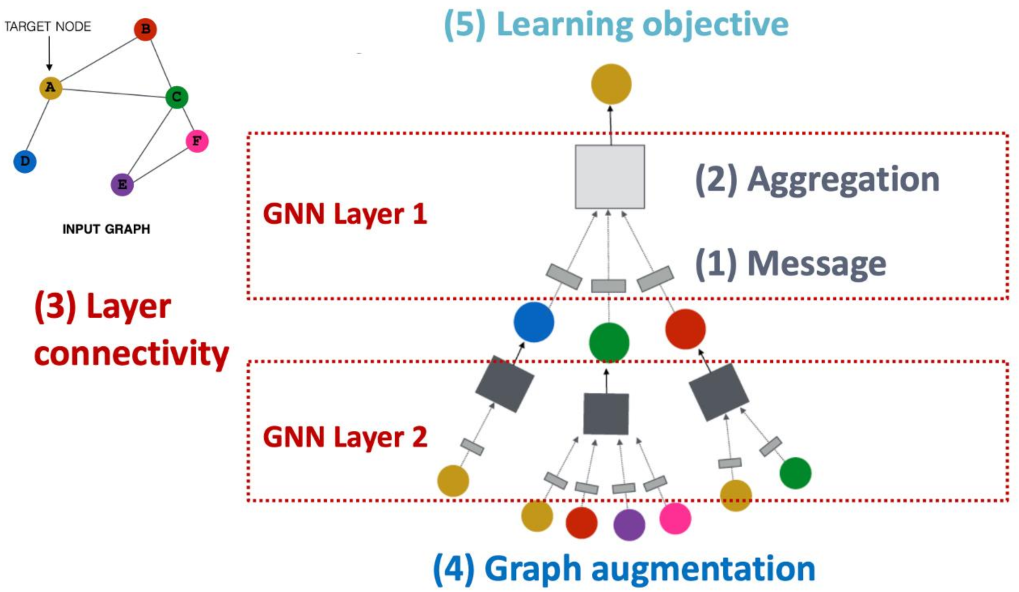


(a) Synthetic Layered Dataset DeepWalk
(b) Synthetic Layered Dataset CrossWalk ($\alpha = 0.5$, $p = 2$)
(c) Rice-Facebook Dataset DeepWalk
(d) Rice-Facebook Dataset CrossWalk ($\alpha = 0.5$, $p = 4$)

- **α :** manipulate strengths to the other group (The higher, the powerful)
- **p :** manipulate strengths to the group boundaries (The higher, the powerful)

Fairness

InFoRM (KDD ‘20)



GNN, Graph Neural Networks

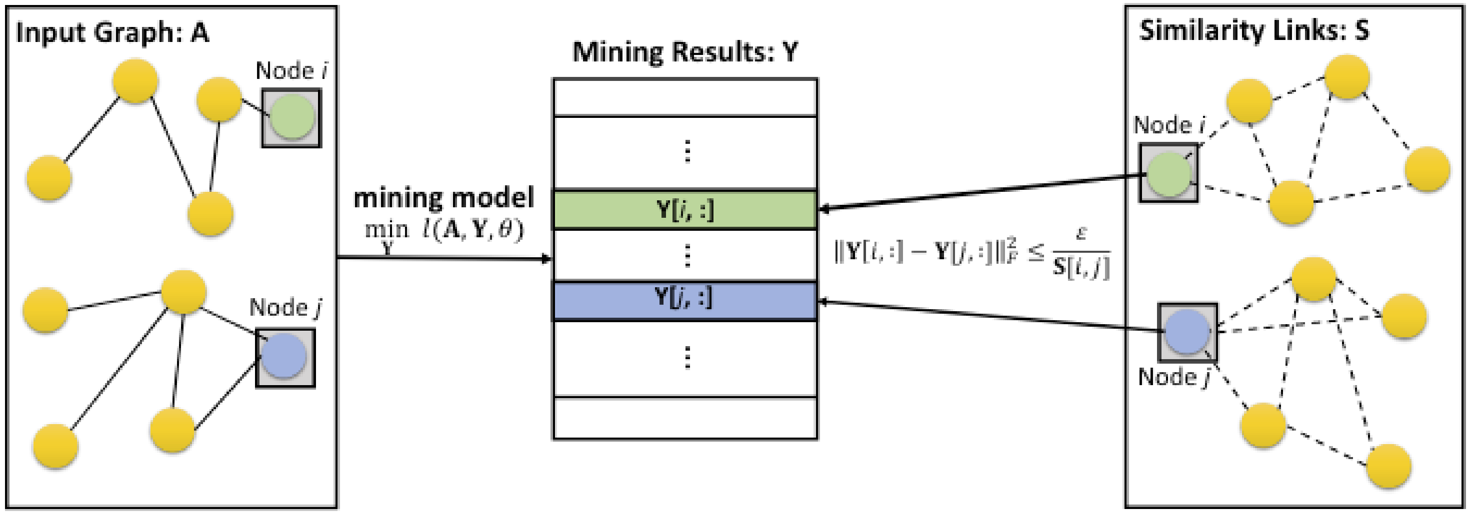
Goal **:** To learn meaningful representations of nodes, edges, or entire graphs for tasks

Method:

- **Message Passing** : Each node gathers information from its neighbors
- **Aggregation** **:** The gathered messages are combined using a function

Fairness

InFoRM (KDD '20)



Goal : Increase **the individual fairness** of graph mining tasks

Method: Add the term related with individual fairness. Then, solve the optimization problem

- Debiasing the input graph

- Debiasing mining model

- Debiasing mining results

Fairness

InFoRM (KDD '20)

Individual Fairness Term

$$\|\mathbf{Y}[i,:] - \mathbf{Y}[j,:]\|_F^2 \leq \frac{\epsilon}{\mathbf{S}[i,j]} \quad \forall i, j = 1, \ldots, n$$

- ❑ The higher the similarity, the smaller the difference
- ❑ Need to calculate about **all pairs** of i and j nodes

$$\sum_{i=1}^{n}\sum_{j=1}^{n} \|\mathbf{Y}[i,:] - \mathbf{Y}[j,:]\|_F^2 \mathbf{S}[i,j] = 2\mathrm{Tr}(\mathbf{Y}'\mathbf{L}_{\mathbf{S}}\mathbf{Y}) \leq m\epsilon = \delta$$

- ❑ Convert the equation into Trace function format (Can calculate **in single time**)
- ❑ The smaller the Tr term, the higher the individual fairness

Fairness

InFoRM (KDD ‘20)

Debiasing the input graph

$$\min_{\tilde{\mathrm{A}}} \quad \|\tilde{\mathrm{A}} - \mathrm{A}\|_F^2 + \alpha \mathrm{Tr}(\mathrm{Y}'\mathrm{L_S}\mathrm{Y}) \quad \text{s.t.} \quad \partial_{\mathrm{Y}} l(\tilde{\mathrm{A}}, \mathrm{Y}, \theta) = 0$$

Debiasing the mining model

$$\mathrm{Y}^* = \underset{\mathrm{Y}}{\operatorname{argmin}} \quad J = l(\mathrm{A}, \mathrm{Y}, \theta) + \alpha \mathrm{Tr}(\mathrm{Y}'\mathrm{L_S}\mathrm{Y})$$

Debiasing the mining results

$$\mathrm{Y}^* = \underset{\mathrm{Y}}{\operatorname{argmin}} \quad J = \|\mathrm{Y} - \bar{\mathrm{Y}}\|_F^2 + \alpha \mathrm{Tr}(\mathrm{Y}'\mathrm{L_S}\mathrm{Y})$$

2. Disentanglement

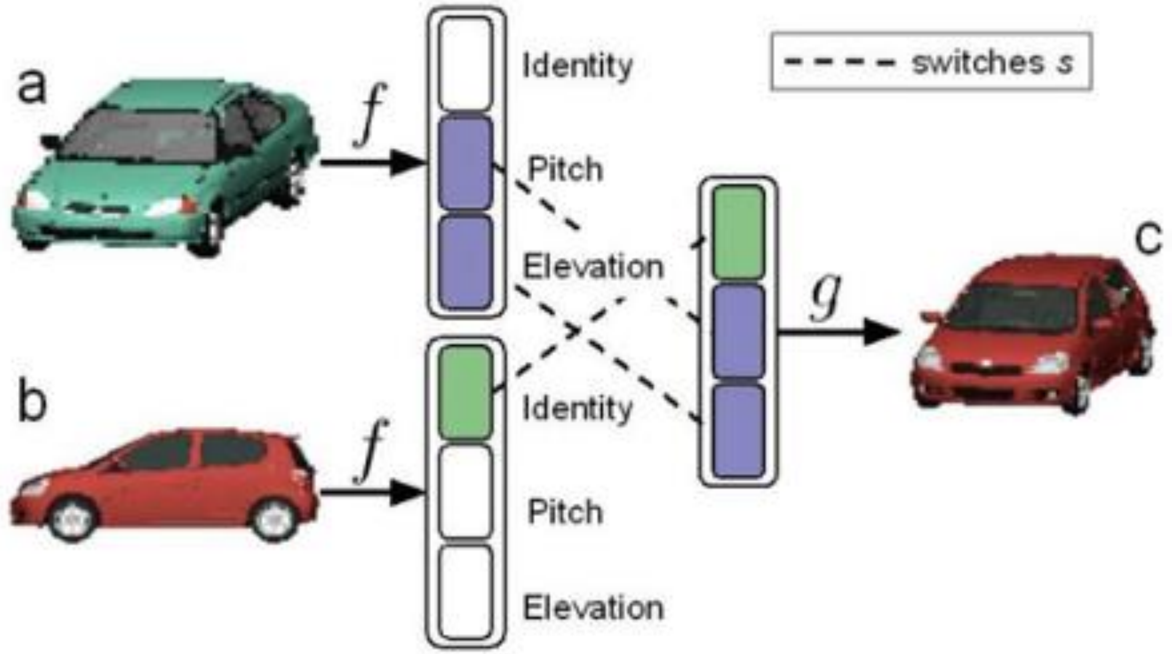
- Disentangle notion
- DisenGCN
- FactorGCN

Disentanglement

Disentangle notion

Disentangled representation learning

➔ Aims to encode **independent factors** of variation **into different dimensions** of the learned representation.

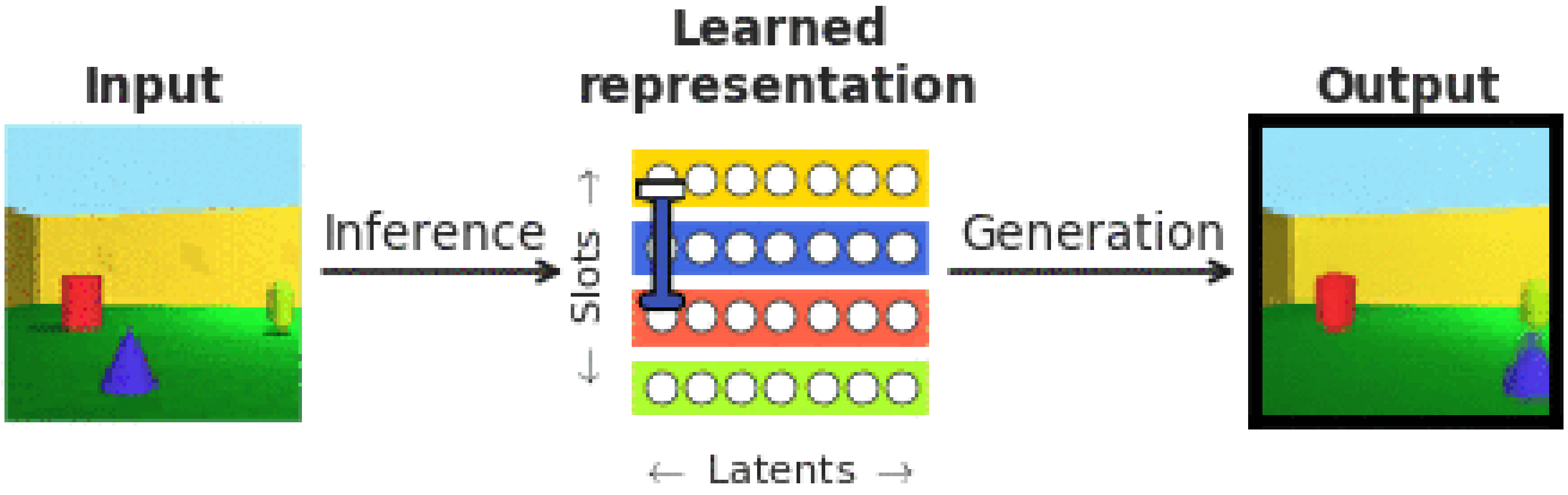


Disentanglement

Disentangle notion

Disentangled representation learning

➔ Aims to encode **independent factors** of variation **into different dimensions** of the learned representation.

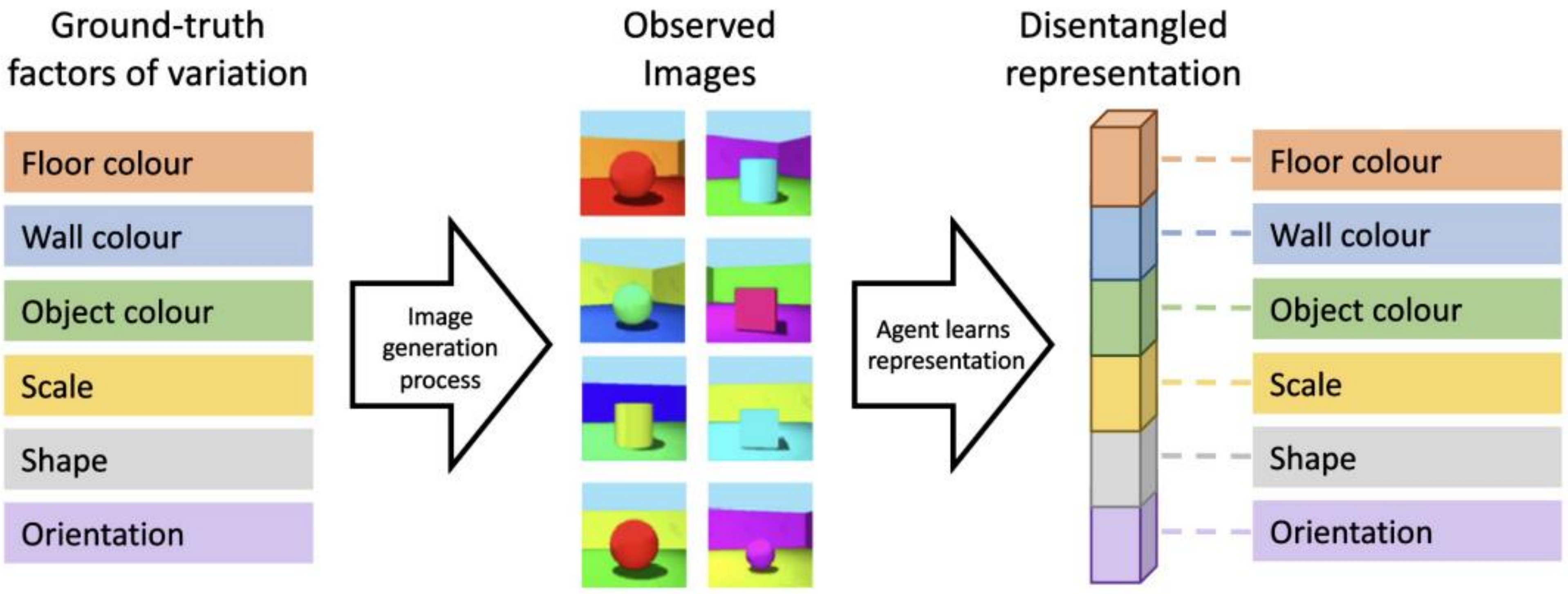


Disentanglement

Disentangle notion

Why do we need and What are benefits?

❑ Enhance **generalization ability** and **robustness**

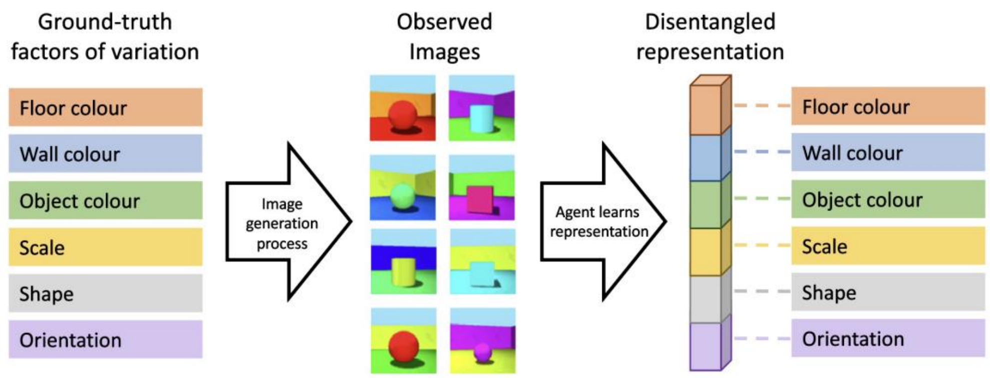


Disentanglement

Disentangle notion

Why do we need and What are benefits?

❑ Enhance **generalization ability** and **robustness**



❑ Enhances **Interpretability**

- Since we can distinguish the input

❑ Advances **Fairness**

- We may separate the sensitive data from input

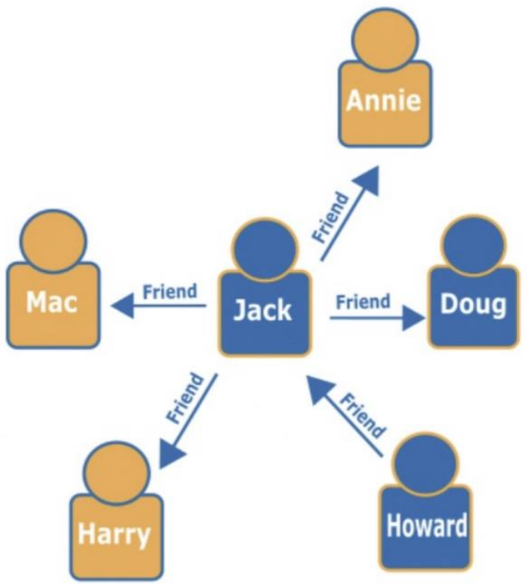
Disentanglement

DisenGCN (ICML ’19)

Motivation

To identify the subset of neighbors that are connected due to factor k.

e.g., Friends, co-worker, or subscribing

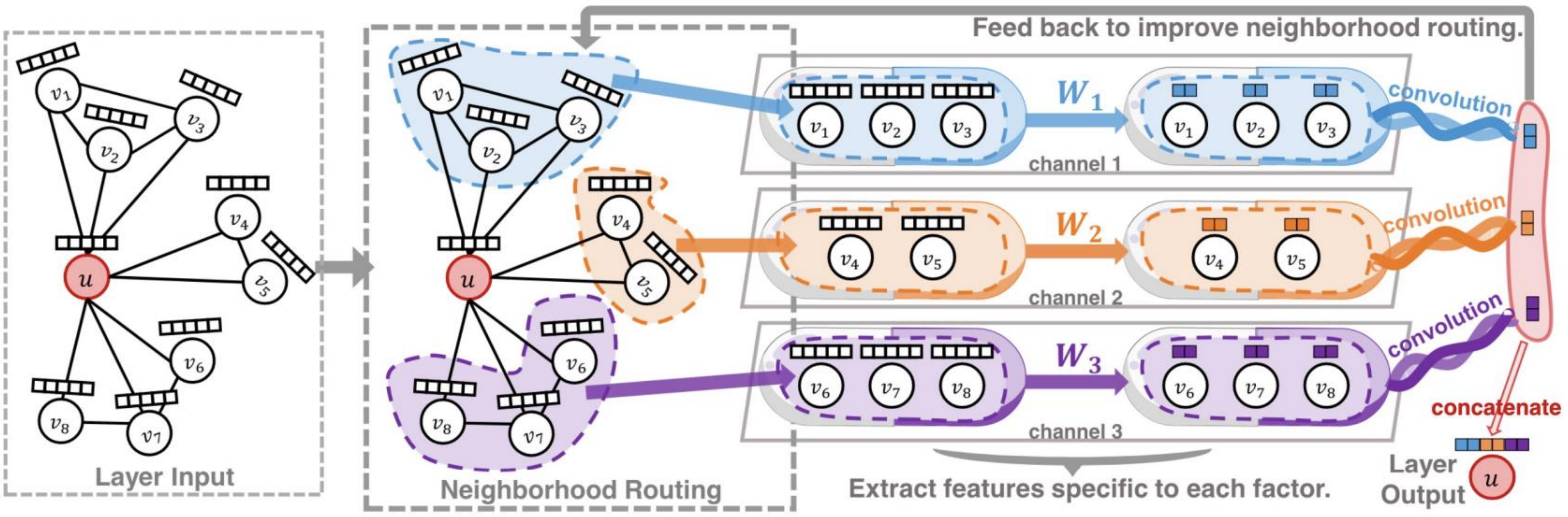


❖ Scenario

- ❑ Jack in a social network
- ❑ Mac, Harry, and Annie are high school friends
- ❑ Doug and Howard are co-workers
- ❑ Connects with others for various reasons

Disentanglement

DisenGCN (ICML ’19)



Methodology

Neighborhood Routing : **Segments** the neighborhood according to the factors

Extract features specific : **Extract** the features from the input node

Disentanglement

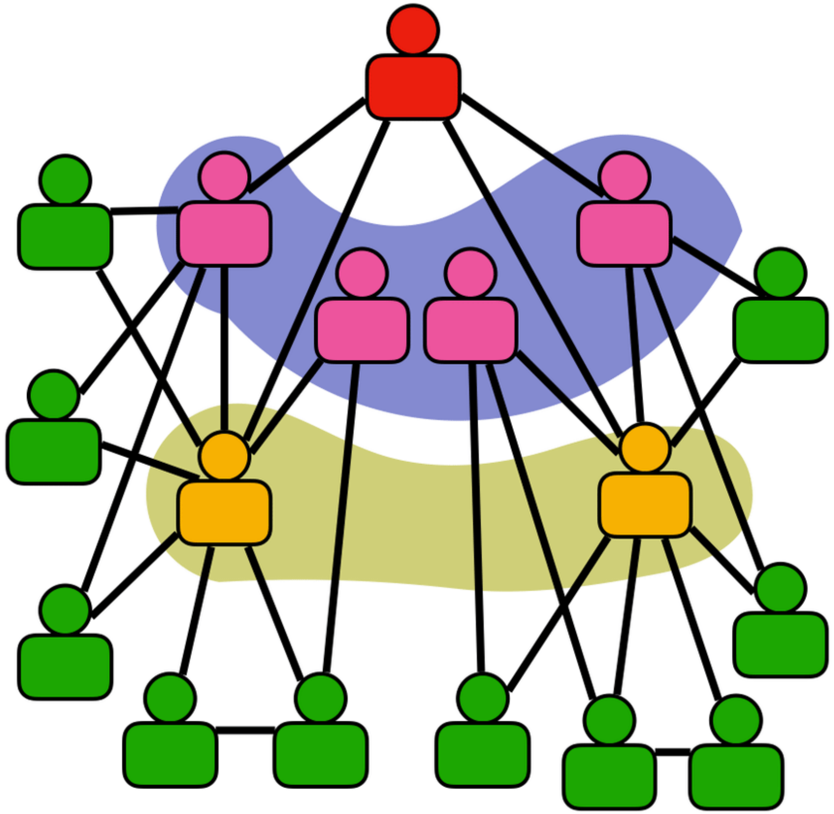
DisenGCN (ICML ’19)

Output: a **disentangled** representation into **K** independent components

$$\mathbf{y}_u = [\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_K], \text{ where } \mathbf{c}_k \in \mathbb{R}^{\frac{d_{out}}{K}} \ (1 \leq k \leq K),$$

Hypo1: First-order proximity

➔ If u and v are similar in terms of factor k, then the factor k is likely to be the reason why they are connected



Disentanglement

DisenGCN (ICML ’19)

Hypo1: First-order proximity

➔If u and v are similar in terms of factor k,

then the factor k is likely to be the reason why they are connected

$$p_{v,k}^{(t)} = \frac{\exp\left({\mathbf{z}_{v,k}}^\top \mathbf{c}_k^{(t)} / \tau\right)}{\sum_{k'=1}^{K} \exp\left({\mathbf{z}_{v,k'}}^\top \mathbf{c}_{k'}^{(t)} / \tau\right)}, \qquad \mathbf{z}_{i,k} = \frac{\sigma(\mathbf{W}_k^\top \mathbf{x}_i + \mathbf{b}_k)}{\left\|\sigma(\mathbf{W}_k^\top \mathbf{x}_i + \mathbf{b}_k)\right\|_2},$$

$z_{i, k}$: Describes the aspect of node i that are related with the k-th factor

→ Kind of a embeddings in terms of k-th factor

$p_{v, k}$: Probability that factor k is the reason why u and v are connected

→ Measure how similar with u and v by multiplying two embeddings

c_k : The subset of the final output about node u

Disentanglement

DisenGCN (ICML ’19)

Hypo2: Second-order proximity

➔ if the neighbors form a large cluster in the k-th subspace,

reflecting similarity with respect to factor k.

$$\mathbf{c}_k^{(t)} = \frac{\mathbf{z}_{u,k} + \sum_{v:(u,v)\in G} p_{v,k}^{(t-1)} \mathbf{z}_{v,k}}{\|\mathbf{z}_{u,k} + \sum_{v:(u,v)\in G} p_{v,k}^{(t-1)} \mathbf{z}_{v,k}\|_2},$$

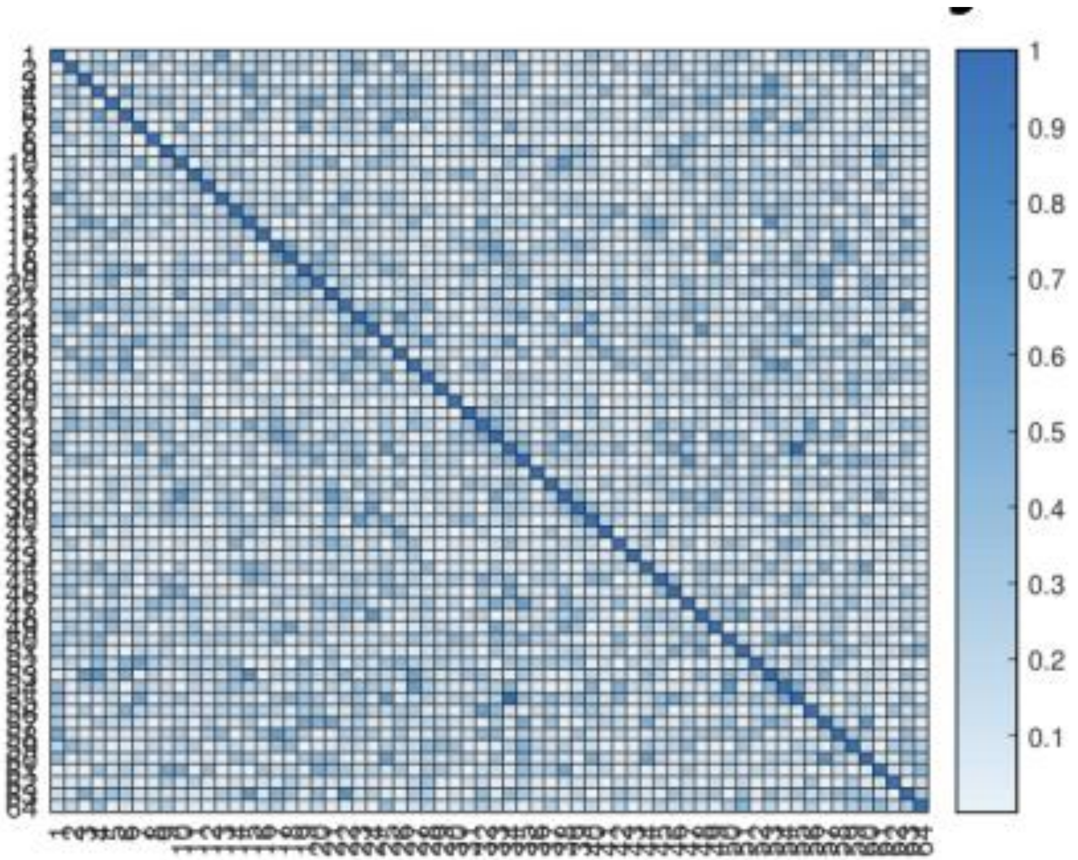
$\mathbf{z}_{i,k}$: Describes the aspect of node i that are related with the kth factor

$\mathbf{p}_{v,k}$: Probability that factor k is the reason why u and v are connected

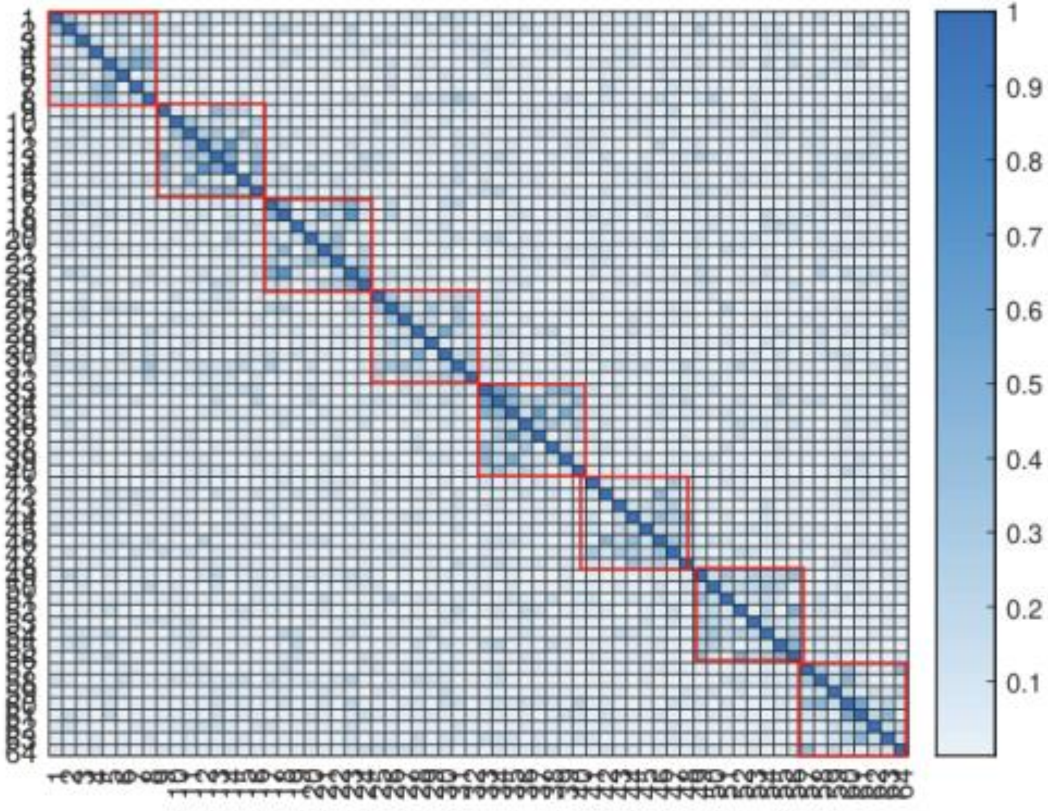
$\mathbf{c}_{v,k}$: The final output of the node in terms of the factor k

Disentanglement

DisenGCN (ICML ’19)



(a) GCN.



(b) DisenGCN (this work).

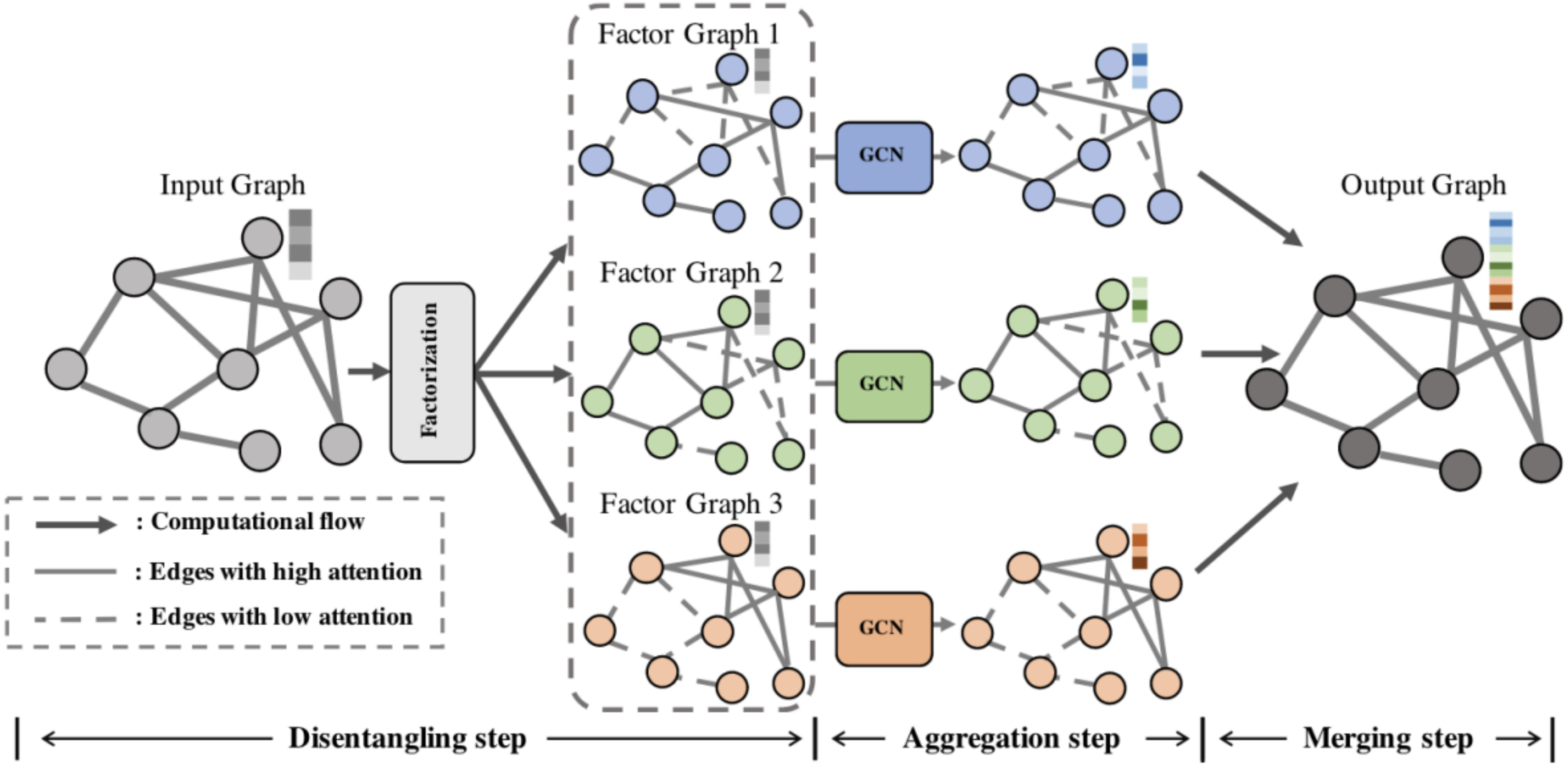
Disentanglement

FactorGCN (NeruIPS ‘20)

Motivation

❑ To focus **on the graph-level** partition not a node-level neighbor partition.

❑ To allow for **overlapping edges** where needed



Disentanglement

FactorGCN (NeruIPS '20)

Methodology – Attention score

$$E_{ije} = 1/\left(1 + e^{-\Psi_e(h'_i, h'_j)}\right); h' = \mathbf{W}h,$$

- ❑ **h**: the set of nodes with feature of F dimension
- ❑ **W**: a linear transformation matrix
- ❑ Ψ: computes **the attention score** for factor graph e **(one-layer MLP)**
  - As Ψ increases, E_{ije} decreases, and vice versa
- ❑ E_{ije}: the coefficient of edge
- ❑ Notice there are no **softmax** → The sum of the E **does not need to be 1.**

Disentanglement

FactorGCN (NeruIPS '20)

Methodology – Aggregation & Merge & Loss function

Aggregation

$$h_i^{(l+1)e} = \sigma\Big(\sum_{j \in \mathcal{N}_i} E_{ije}/c_{ij} h_j^{(l)} \mathbf{W}^{(l)}\Big), c_{ij} = \big(|\mathcal{N}_i||\mathcal{N}_j|\big)^{1/2},$$

Merge

$$h_i^{(l+1)} = ||_{e=1}^{N_e} h_i^{(l+1)e},$$

Loss Function

$$\mathcal{L} = \mathcal{L}_t + \lambda * \mathcal{L}_d$$

Disentanglement

FactorGCN (NeruIPS '20)

Methodology – Separating

$$G_e = \mathrm{Softmax}\Big(f\Big(\mathrm{Readout}(\mathcal{A}(\mathbf{E}_e, \mathbf{h}'))\Big)\Big).$$

$$\mathcal{L}_d = -\frac{1}{N}\sum_{i}^{N}\left(\sum_{c=1}^{N_e} \mathbb{1}_{e=c} log(G_i^e[c])\right),$$

N_e: the number of factor graphs

- ❑ Without **any other constraints**, some factor graphs **may become similar**.
- ❑ Need to be distinguished from the rest.
- ❑ By assigning unique labels to the factor graphs

  and optimizing them as a graph classification problem **like a discriminator**
- ❑ **A classifier f:** consist of one fully connected layer
- ❑ **$G_i[c]$:** represents the probability that the generated factor graph has label c

Weekly Meetings

3. Invariant Learning & Causal-based Learning

- Invariant Learning
- Causal-based Learning
- DIR
- DisC

I.R. & Causal Based

Invariant Learning

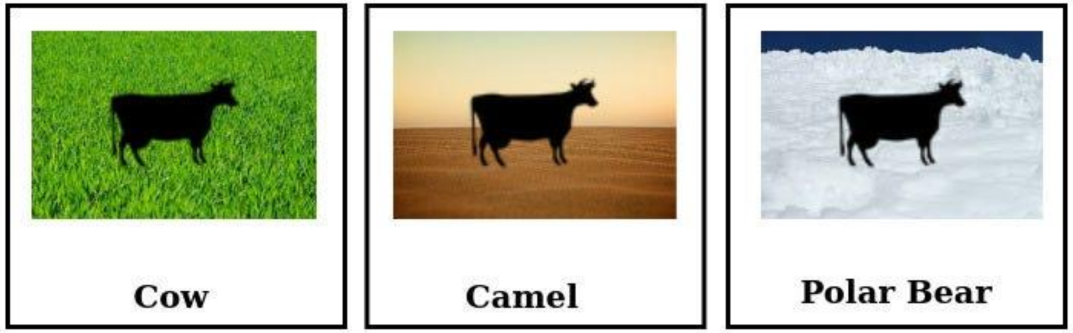
Invariant Learning

What: Aims to identify patterns that **remain consistent across different environments**

How: By optimizing for predictive **performance that remains stable across multiple environments or domains.**

Benefits: Enhances **Robustness** to Distribution Shifts (Better performance in OOD)

Reduces **Overfitting** to Spurious Correlations



Neural Network Predictions

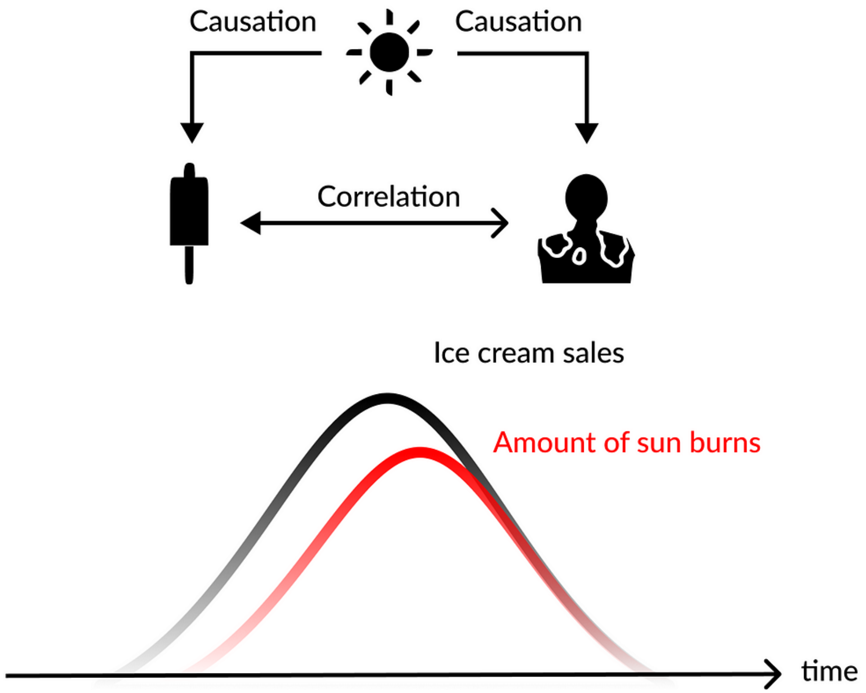
I.R. & Causal Based

Causal-based learning

Causal-based learning

What:

focuses on uncovering **cause-effect** relationships



I.R. & Causal Based

Causal-based learning

Causal-based learning

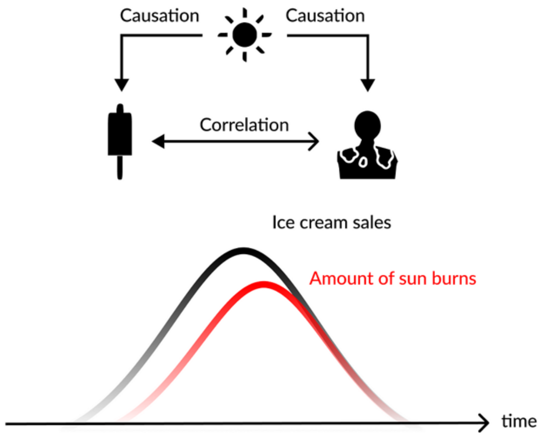
How:

identifies and leverages **causal structures** in the data

Benefits:

Enhances **Generalization**

Improve **Fairness** & Facilitates **Interpretability**

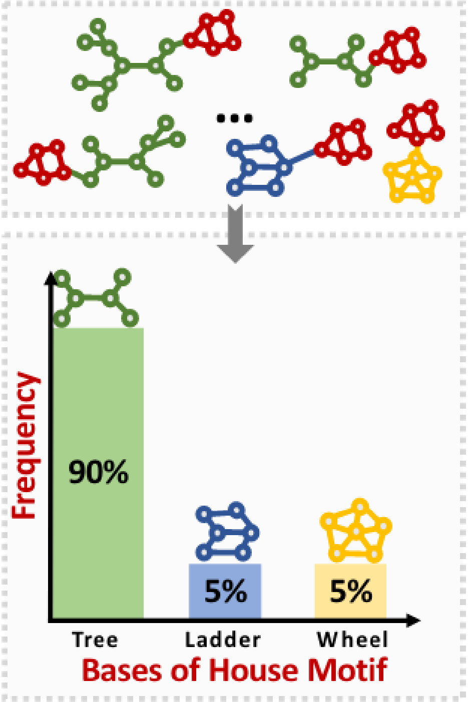


I.R. & Causal Based

DIR (ICLR ’22)

Motivation

- ❑ Risk of learning from the **statistical shortcuts** can lead **poor generalization**
- ❑ Aim to identify rationales that capture **the environment-invariant causal patterns**



Example

Want to classify **the house motif (the red ones)**

If the model capture **the tree motif** as a sign of house motif,

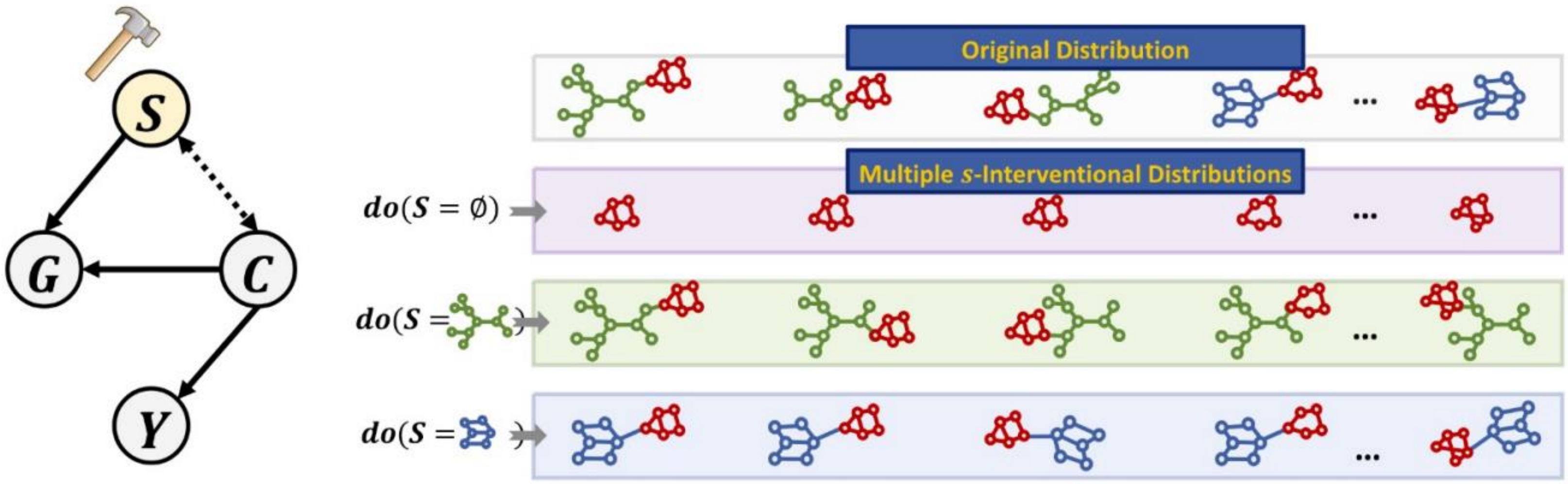
the model might have **poor generalization**

I.R. & Causal Based

DIR (ICLR ’22)

How

- By Do-Calculus developed by Pearl
  - **S: Shortcut (≈ suspicious) Part**
  - Introduce Interventional Distribution (Iteratively replace the **S**)

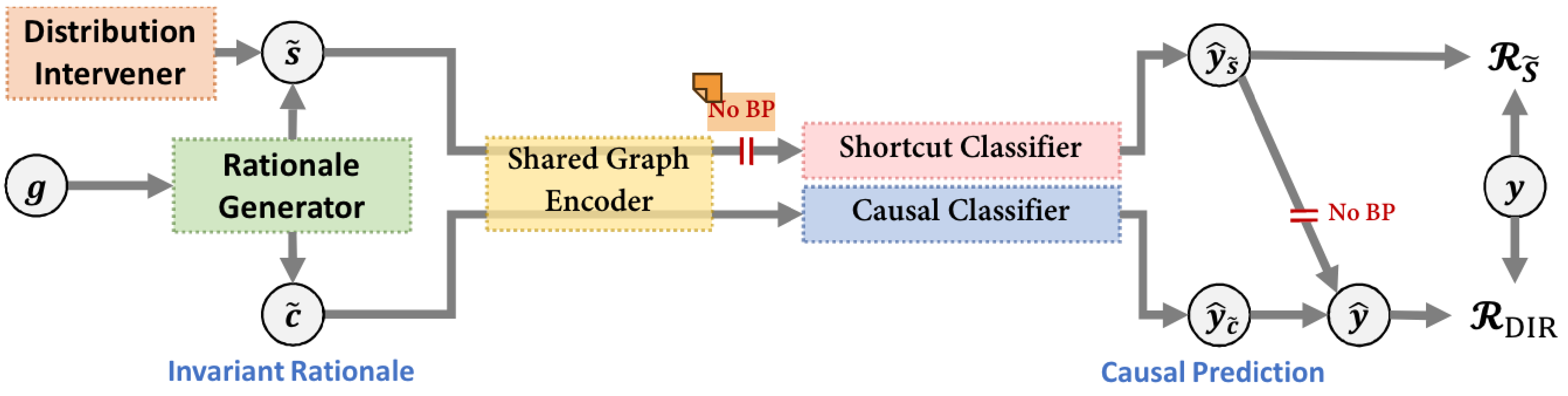


(a) SCM

(b) Interventional Distributions.

I.R. & Causal Based

DIR (ICLR ’22)



Rational Generator : Split the input graph into causal and non causal parts by **GNN**

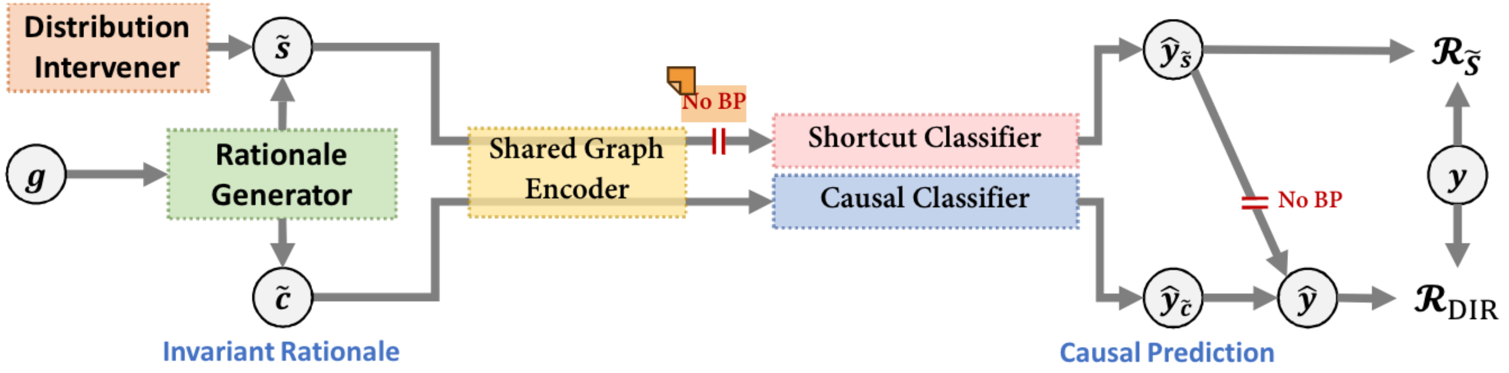
Distribution Intervener : Creating interventional distributions. Memory bank of **S**

Shared Encoder : Generate node representations independently

Two Classifiers : Make a probability distribution over class labels by **GNN**

I.R. & Causal Based

DIR (ICLR '22)



Loss function

$$\min \mathcal{R}_{\mathrm{DIR}} = \mathbb{E}_s[\mathcal{R}(h(G), Y | do(S = s))] + \lambda \mathrm{Var}_s(\{\mathcal{R}(h(G), Y | do(S = s))\}),$$

Try to **minimize** both the **risk itself** and the **variability of risk** in shortcut environments by **(do(S=s))**

$$\mathcal{R}(h(G), Y | do(S = \tilde{s})) = \mathbb{E}_{(g,y)\in\mathcal{O}, S=\tilde{s}, C=h_{\tilde{C}}(g)} l(\hat{y}, y), \quad \hat{y} = \hat{y}_{\tilde{c}} \odot \sigma(\hat{y}_{\tilde{s}}),$$

Risk: The **expectation(Mean)** of the loss between ŷ and y through shortcut environments

I.R. & Causal Based

DisC (NeurIPS '22)

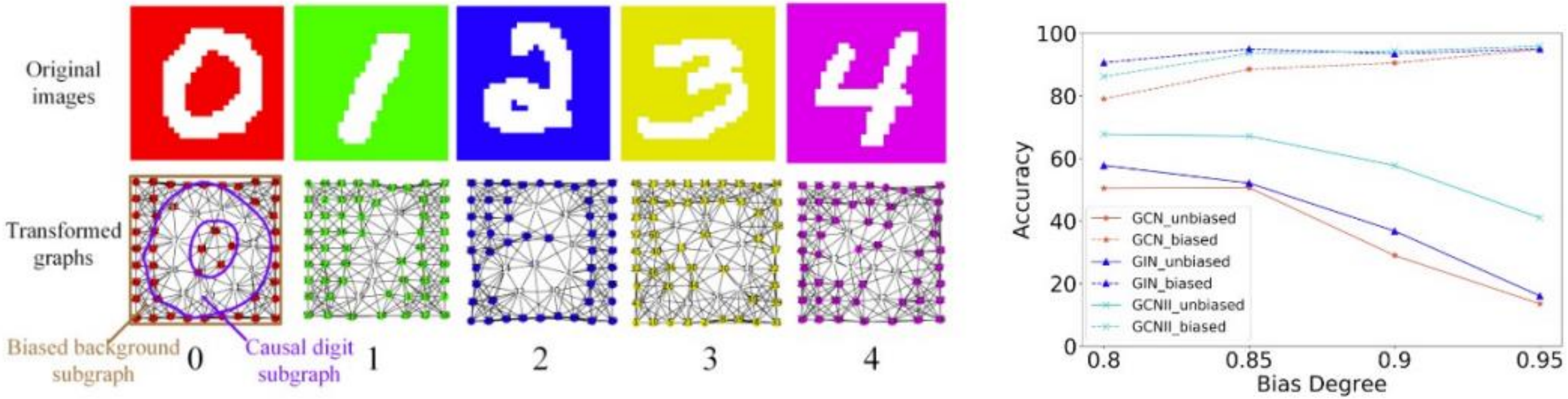
Motivation

❑ GNNs tend to learn bias information, especially **in severe bias situation**

❑ **Assume** bias part usually has **simpler structure** than **meaningful causal part**

➔ Therefore, the **bias part** is **easy to learn** than causal part

Example

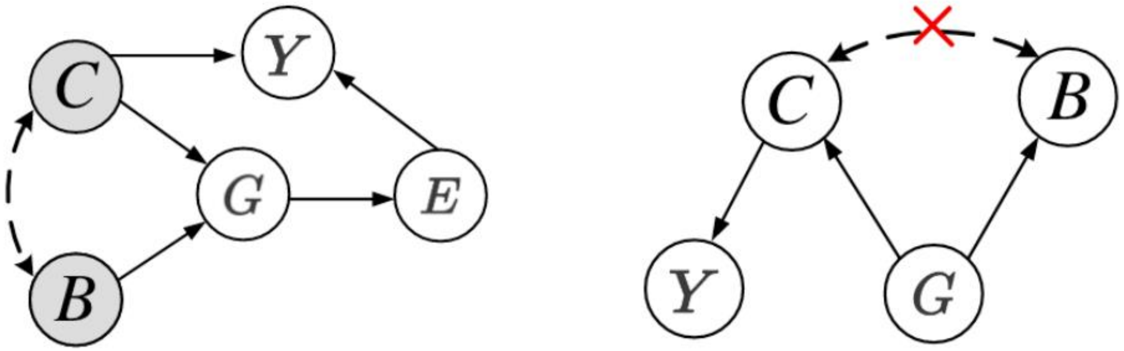


(a) Examples of graphs in CMNIST-75sp. (b) Performance of GNNs.

I.R. & Causal Based

DisC (NeurIPS '22)

How



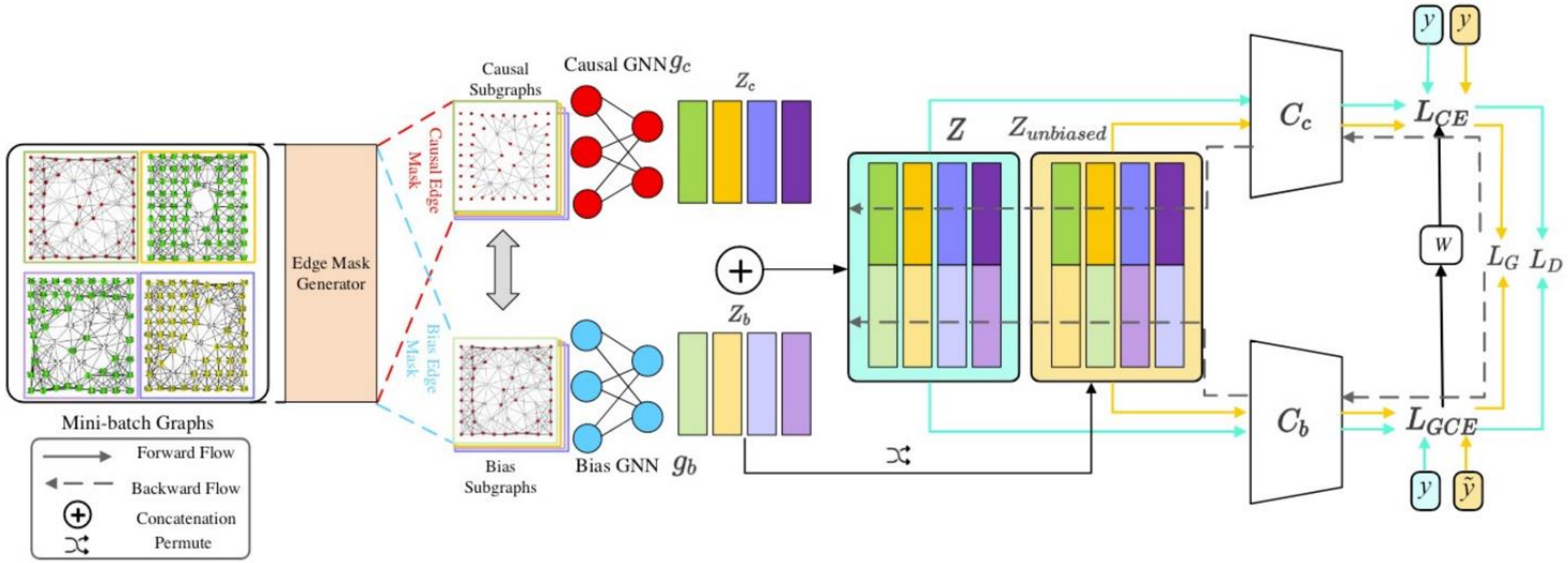
SVM, Structural Causal Model on GNNs' prediction process

Problem & Solving

- ❑ We do not know the actual causal and bias part
  - ➔ Try to **disentangle** the latent variables C and B from the input G
- ❑ There are two paths that would induce the spurious correlation
  - ➔ **Cut off** the correlation between C and B

I.R. & Causal Based

DisC (NeurIPS '22)



Edge mask generator : **Generate** Causal and Bias substructure

Two separate GNN modules : **Learn** Disentangled Graph Representations

Unbiased Sample Generation : **Permute** the bias representations (**Intervention**)

I.R. & Causal Based

DisC (NeurIPS '22)

How to ensure they are **causal** and **bias** sub graph, respectively?

➔ Use the **assumption** that is "The bias part is easy to learn"

➔ Train the **bias** classifier by using **GCE, generalized cross entropy**

$$GCE(C_b(z;\alpha_b), y) = \frac{1 - C_b^y(z;\alpha_b)^q}{q},$$

- **C(•)** are softmax **output** of the bias classifier
- **High C(•)** can be interpreted by **high confidence** about its prediction
- **High C(•)** leads **a low GCE**. That is high confidence leads a low GCE.
- This loss function encourages the bias classifier to focus on data that is high confidence and easy to learn. (it might be bias ones)

I.R. & Causal Based

DisC (NeurIPS '22)

Train the **causal** classifier by using **Unbias score, W(z)**

$$W(z) = \frac{CE(C_b(z), y)}{CE(C_c(z), y) + CE(C_b(z), y)}.$$

- **High CE** from bias classifies means that the **C_b fail to predict** on data z
- **Relatively** high CE loss from C_b than C_c can be regarded as the unbiased
  - **Relatively**, C_b fail to predict and C_c success to predict

$$L_D = W(z)CE(C_c(z), y) + GCE(C_b(z), y).$$

- Combine with unbias score and CE from C_c for **causal** classifier

I.R. & Causal Based

DisC (NeurIPS ‘22)

Problem & Solving

- ❑ We do not know the actual causal and bias part
  - ➔ Try to **disentangle** the latent variables C and B from the input G
- ❑ There are two paths that would induce the spurious correlation
  - ➔ **Cut off** the correlation between C and B

Until now, we did a first part: disentangle the latent variables

➔ We need to **cut off** the correlation between **casual** and **bias** variables ($C \nleftrightarrow B$)

➔ Use **Do-calculus**; Randomly **permute(change)** bias part

$$L_G = W(z)CE(C_c(z_{unbiased}), y) + GCE(C_b(z_{unbiased}), \hat{y}),$$

$$L = L_D + \lambda_G L_G,$$

Weekly Meetings

4. Fairness & Disentanglement

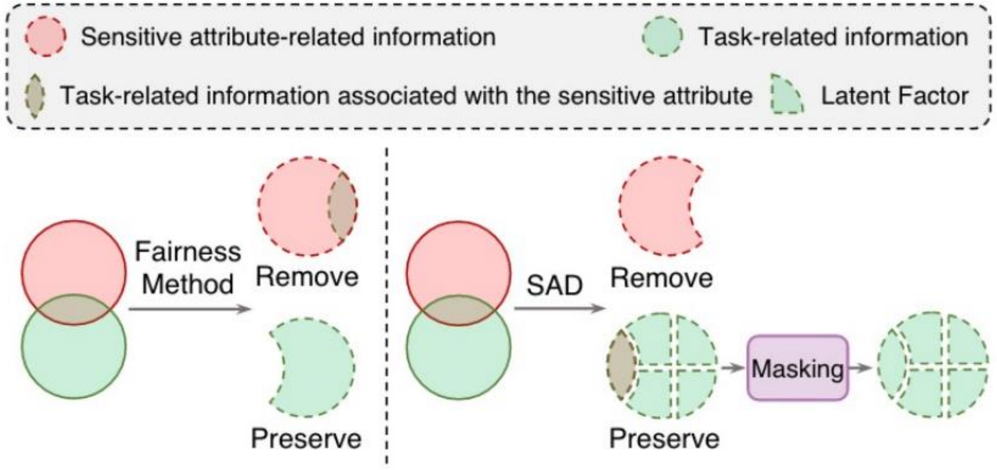
- FairSAD
- FairINV

Fair & Disentangle

FairSAD (WWW ’24)

Motivation

- ❑ Focus on **addressing performance degradation** when improving fairness.
- ❑ Employ the **Disentangle Learning** for two potential advantages
    - **Reduces correlations** between the sensitive attribute and others
    - Simplifies downstream tasks and leads to **better utility performance**

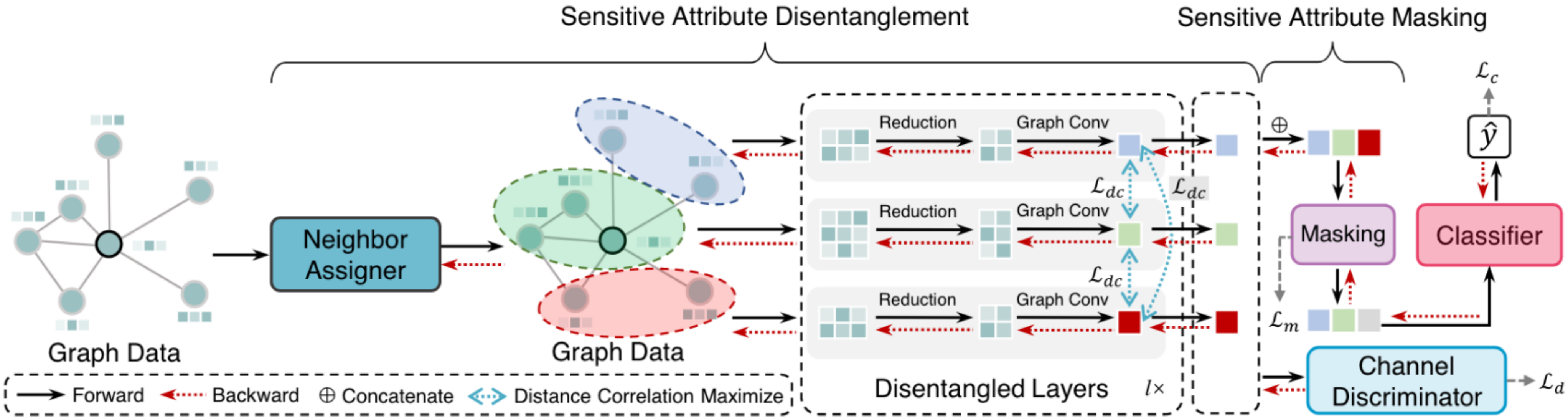


(a) Existing Works (b) Our Work

Fair & Disentangle

FairSAD (WWW ’24)

How



SAD, Sensitive Attribute Disentanglement

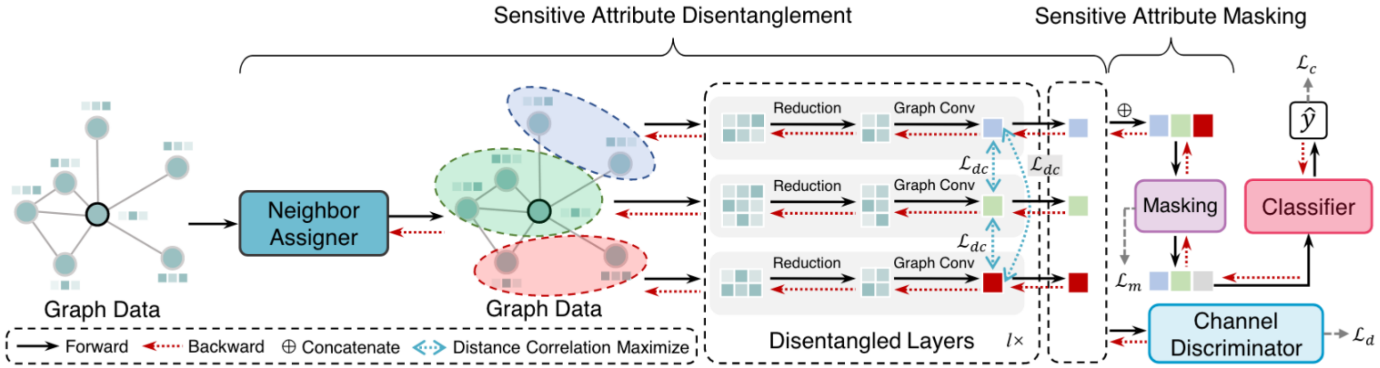
Disentangles the sensitive attribute into independent components

Sensitive attribute masking

Employs **a channel masking** to identify the sensitive attribute

Fair & Disentangle

FairSAD (WWW '24)



SAD

Neighbor Assigner : **Separate** the sensitive attribute **by MLP**

Disentangled layers : Perform graph convolution in multi-channel

Output : $\mathbf{H} = [\underbrace{\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3}_{\mathbf{Z}^1}, ..., \underbrace{\mathbf{c}_{d_h-2}, \mathbf{c}_{d_h-1}, \mathbf{c}_{d_h}}_{\mathbf{Z}^k}]$ (each channel consist of three columns)

Sensitive Attribute Masking

Try to **mask(remove)** the sensitive column

$$\begin{aligned}\tilde{\mathbf{H}} = \mathbf{H} \odot \mathbf{m} &= [\mathbf{c}_1 m_1, \mathbf{c}_2 m_2, \mathbf{c}_3 m_3, ..., \mathbf{c}_{d_h} m_{d_h}] \\ &= [\underbrace{\tilde{\mathbf{c}}_1, \tilde{\mathbf{c}}_2, \tilde{\mathbf{c}}_3}_{\tilde{\mathbf{Z}}^1}, ..., \underbrace{\tilde{\mathbf{c}}_{d_{h-2}}, \tilde{\mathbf{c}}_{d_{h-1}}, \tilde{\mathbf{c}}_{d_h}}_{\tilde{\mathbf{Z}}^k}],\end{aligned}$$

Fair & Disentangle

FairSAD (WWW '24)

$$\min_{\theta} \mathcal{L} = \mathcal{L}_c + \alpha(\mathcal{L}_{dc} + \mathcal{L}_d) + \beta \mathcal{L}_m,$$

Optimization Purposes

- **Downstream Tasks** : Ensure that the learned representations is informative
- **Disentanglement** : Ensure the independence between latent factors
- **Decorrelation** : Weaken the impact of the sensitive attribute-related component

Decorrelation

- **Goal:** Aims to assign the minimum masking value to the sensitive attribute-related component
- **How:**

$$\mathcal{L}_m = \sum_{i=1}^{d_h} |Cov(\mathbf{s}, \tilde{\mathbf{c}}_i)| = \sum_{i=1}^{d_h} |\mathbb{E}[(\mathbf{s} - \mathbb{E}(\mathbf{s}))(\tilde{\mathbf{c}}_i - \mathbb{E}(\tilde{\mathbf{c}}_i))]|,$$

Fair & Disentangle

FairINV (KDD ‘24)

Motivation

- Most works **necessitates prior knowledge** of considered sensitive attributes
- Need to **re-training from scratch** when faced **with fairness requirement alternations**

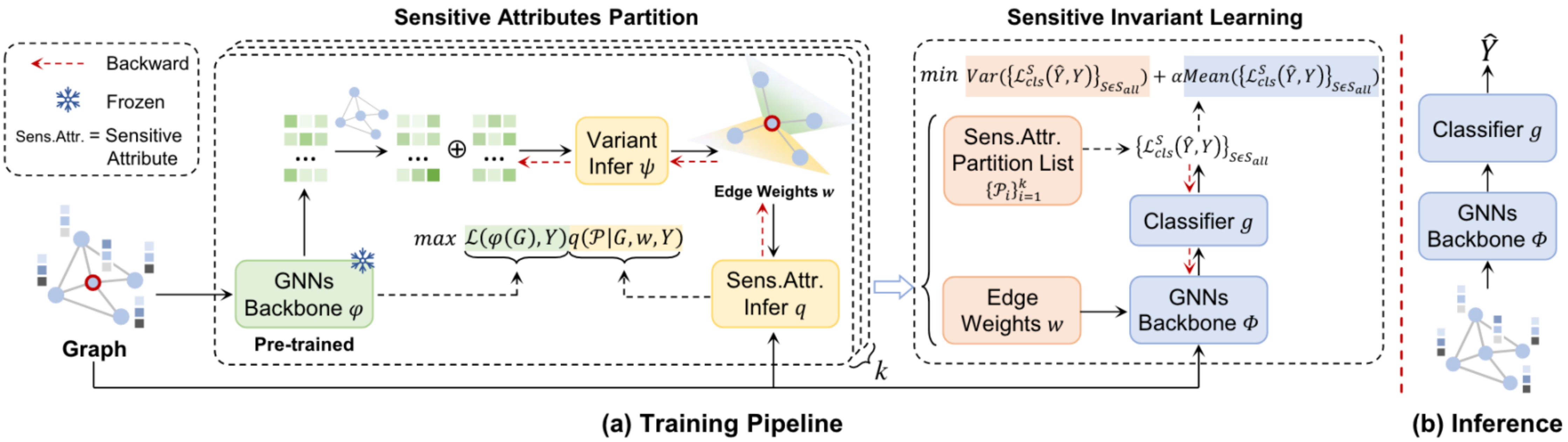
➔ Need to train fair GNNs across various sensitive attribute**s in a single training session**

How

- The core idea behind **Invariant Learning** is similar with **Fairness Learning**
- The goal of **Invariant Learning** is to treat different **environments** equally
- The goal of **Fairness Learning** is to treat different **demographic groups** equally

Fair & Disentangle

FairINV (KDD '24)



SAP

Try to **automatically partitions** nodes into different subset

SIL

Learns a GNN invariant across different sensitive attribute partitions

Fair & Disentangle

FairINV (KDD '24)

SAP Goal:

- Try to capture **variant patterns** that result in significant performance differences across different environments
  - **Hope** that the environments is related with **sensitive attribute**

SAP How:

$$\max_{\theta_\psi, \theta_q} \|\nabla_{\overline{\mathbf{w}}} \mathcal{R}^S(\overline{\mathbf{w}} \circ \varphi, q)\|,$$

$$\mathcal{R}^S(\varphi, q) = \sum_{v \in \mathcal{V}} \mathbf{q}_v(S) \mathcal{L}(\varphi(\mathcal{G}_v), y_v),$$

$$\mathbf{q}_v(S) : q_v(S|\mathcal{G}_v, \mathbf{w}^i, y_v)$$

- **q_v(S)** means some soft partition
- By **calculating** and **maximizing the gradient of R^s**
- **q_v(S)** is optimized in a way that partitions to maximize the performance differences across environments.

Fair & Disentangle

FairINV (KDD '24)

SIL Goal:

- Aims to **minimize the performance difference** between various sensitive attribute groups and ensures **the predicted accuracy** across all sensitive attribute groups

$$\min_{\theta_f} Var(\{\mathcal{L}^S_{cls}(\hat{Y}, Y)\}_{S \in \mathcal{S}_{all}}) + \alpha Mean(\{\mathcal{L}^S_{cls}(\hat{Y}, Y)\}_{S \in \mathcal{S}_{all}}),$$

  - The sensitive attribute group S is derived from **q(S)**
  - $\mathbf{L}_{\mathbf{cls}}^{\mathbf{S}}$ is the classification loss function under **S**

Conclusion

- Until Now, I focus on **disentangled way to improve the fairness**
- However, there are many ways to improve **the fairness**
  - Could be extended to other ways like **counterfactual, adversarial, regularization, and others**