Out-of-distribution Generalized Graph Neural Network

Ziwei Zhang
Tsinghua University
2022.10.29@LOGS
Graphs/Networks

Social Network

Biology

Logistics

Transaction

Internet of Things

Knowledge Graphs
Message-passing GNNs

- Background: message-passing framework of GNNs

\[ m_i^{(l)} = \text{AGG}([h_j^{(l)}, \forall j \in \tilde{N}_i]) \]
\[ h_i^{(l+1)} = \text{UPDATE}([h_i^{(l)}, m_i^{(l)}]) \]

- \( h_i^{(l)} \): the representation of node \( v_i \) in the \( l^{th} \) layer
- \( m_i^{(l)} \): the message vector of node \( v_i \) in the \( l^{th} \) layer

- Nodes exchange messages to update representations

Existing GNNs have shown successes in many graph applications
However, real graphs are challenging...

- Many real graphs exist in dynamic and open environments
  - Open: “emerging new classes, decremental/incremental features, changing data distributions, varied learning objectives, etc.” (Zhi-hua Zhou)
  - I.e., distribution shifts naturally exist in graph data

Out-of-distribution generalized GNNs are critically needed!
Main Challenge for Handling Distribution Shifts

- Why existing GNNs fail to achieve OOD generalization?
- Our answer: **spurious correlations**
  - GNNs tend to exploit statistical correlations in the training set
  - But spurious correlations cannot generalize under distribution shifts

Picture credit: Duvenaud et al., NeurIPS 2015; Ying-Xin Wu, et al., ICLR 2022
Handling Distribution Shifts

How to handle distribution shifts in GNN architectures?

How to handle distribution shifts in the topology space?

How to handle distribution shifts in the vector space?
How to get rid of spurious correlations in node representations?

- Main idea: decorrelations
- Remove the statistical dependence of truly predictive (causal) information and spurious (non-causal) information by sampling reweighting, i.e., assign each sample (graph) a weight)

- More theoretical backgrounds: direct confounder balancing

In practice: encourage to eliminate statistical dependence of all dimensions
Since we do not know which ones are causal and spurious
To get rid of spurious correlations, we expect \( Z_{*i} \perp Z_{*j}, \forall i, j \in [1, d], i \neq j \)
We adopt Hilbert-Schmidt Independence Criterion (HSIC) measured as:

**Proposition 1.** Assume \( \mathbb{E}[k_{Z_{*i}}(Z_{*i}, Z_{*i})] < \infty \) and \( \mathbb{E}[k_{Z_{*j}}(Z_{*j}, Z_{*j})] < \infty \), and \( k_{Z_{*i}} \) is a characteristic kernel, then

\[
\text{HSIC}(Z_{*i}, Z_{*j}) = 0 \iff Z_{*i} \perp Z_{*j}.
\]

However, calculating HSIC is intractable. We adopt a practical version as:

\[
\min \left\| \widehat{C}_{Z_{*i}, Z_{*j}} \right\|_F^2 = \frac{1}{N^{tr} - 1} \sum_{n=1}^{N^{tr}} \left( f(Z_{ni}) - \frac{1}{N^{tr}} \sum_{m=1}^{N^{tr}} f(Z_{mi}) \right) \cdot \left( g(Z_{nj}) - \frac{1}{N^{tr}} \sum_{m=1}^{N^{tr}} g(Z_{mj}) \right)
\]

where \( f(\cdot) \) and \( g(\cdot) \) are the random Fourier features function:

\[
f(Z_{*i}) := (f_1(Z_{*i}), f_2(Z_{*i}), \ldots, f_Q(Z_{*i})),
g(Z_{*j}) := (g_1(Z_{*j}), g_2(Z_{*j}), \ldots, g_Q(Z_{*j})),
\]

\( f_q(Z_{*i}), g_q(Z_{*j}) \in \mathcal{H}_{\text{RFF}}, \forall q \in [1, Q], \mathcal{H}_{\text{RFF}} = \{h: x \rightarrow \sqrt{2} \cos(wx + \phi)|w \sim \mathcal{N}(0, 1), \phi \sim \text{Uniform}(0, 2\pi)\} \)
OOD-GNN: Optimization

- Optimization objectives: jointly optimize weights

\[
\Phi^*, R^* = \arg\min_{\Phi, R} \sum_{n=1}^{Ntr} w_n \ell (\mathcal{R} \circ \Phi (G_n), Y_n), \\
W^* = \arg\min_W \sum_{1 \leq i < j \leq d} \| \hat{C}_{Z_{*i}, Z_{*j}}^W \|_F^2,
\]

Graph Encoding
- Graph Dataset
- Graph Encoder $\Phi$
- Graph Representation $Z$

Graph Reweighting
- Encourage Independence
- Optimized Weights for $Z_{*i} \perp Z_{*j}$

Learning $\Phi$ and $R$ with weighted prediction loss
- Weighted Loss
- Objective

OOD-GNN: Experiments

- Setup: 14 graph datasets, various kinds of domains/shifts
- Results:

<table>
<thead>
<tr>
<th>TRIANGLES</th>
<th>MNIST-75SP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Train</td>
</tr>
<tr>
<td>GCN</td>
<td>28.3±0.6</td>
</tr>
<tr>
<td>GCN-virtual</td>
<td>32.4±0.6</td>
</tr>
<tr>
<td>GIN</td>
<td>34.7±0.7</td>
</tr>
<tr>
<td>GIN-virtual</td>
<td>34.2±0.6</td>
</tr>
<tr>
<td>FactorGCN</td>
<td>10.6±1.6</td>
</tr>
<tr>
<td>Γ_GIN</td>
<td>23.5±3.4</td>
</tr>
<tr>
<td>PNA</td>
<td>43.7±3.6</td>
</tr>
<tr>
<td>TopKPool</td>
<td>28.3±0.3</td>
</tr>
<tr>
<td>SAGPool</td>
<td>26.7±1.0</td>
</tr>
<tr>
<td><strong>OOD-GNN</strong></td>
<td><strong>29.9±0.7</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>COLLAB₃₅</th>
<th>PROTEINS₂₅</th>
<th>D&amp;D₂₀₀₀</th>
<th>D&amp;D₃₀₀₀</th>
</tr>
</thead>
<tbody>
<tr>
<td># Train/Test graphs</td>
<td>500/4500</td>
<td>500/613</td>
<td>462/716</td>
</tr>
<tr>
<td>#Nodes Train</td>
<td>32-35</td>
<td>4-25</td>
<td>30-200</td>
</tr>
<tr>
<td>#Nodes Test</td>
<td>32-492</td>
<td>6-620</td>
<td>201-5748</td>
</tr>
<tr>
<td>GCN</td>
<td>65.9±3.4</td>
<td>75.1±2.2</td>
<td>29.2±8.2</td>
</tr>
<tr>
<td>GCN-virtual</td>
<td>61.5±1.6</td>
<td>70.4±3.7</td>
<td>41.6±8.0</td>
</tr>
<tr>
<td>GIN</td>
<td>55.5±4.9</td>
<td>74.0±2.7</td>
<td>43.0±8.3</td>
</tr>
<tr>
<td>GIN-virtual</td>
<td>54.8±2.7</td>
<td>66.0±7.5</td>
<td>46.7±4.5</td>
</tr>
<tr>
<td>FactorGCN</td>
<td>51.0±1.3</td>
<td>63.5±4.8</td>
<td>42.3±3.1</td>
</tr>
<tr>
<td>Γ_GIN</td>
<td>65.2±2.3</td>
<td>62.6±3.6</td>
<td>59.6±4.9</td>
</tr>
<tr>
<td>PNA</td>
<td>59.6±5.5</td>
<td>71.4±3.4</td>
<td>47.3±6.8</td>
</tr>
<tr>
<td>TopKPool</td>
<td>52.8±1.0</td>
<td>64.9±3.0</td>
<td>34.6±5.6</td>
</tr>
<tr>
<td>SAGPool</td>
<td>67.0±1.7</td>
<td>76.2±0.7</td>
<td>54.3±5.0</td>
</tr>
<tr>
<td><strong>OOD-GNN</strong></td>
<td><strong>67.2±1.8</strong></td>
<td><strong>78.4±0.9</strong></td>
<td><strong>60.3±4.5</strong></td>
</tr>
</tbody>
</table>

## OOD-GNN: Experiments

- Setup: 14 graph datasets, various kinds of domains/shifts
- Results:

TABLE 5: Results on nine Open Graph Benchmark (OGB) datasets. We report the ROC-AUC (%) for classification tasks and RMSE for regression tasks with the standard deviation on the test set of all methods. None of the baseline methods is consistently competitive across all datasets, while our proposed method shows impressive performance. (↑) means that higher values indicate better results, and (↓) represents the opposite.

<table>
<thead>
<tr>
<th>Metric</th>
<th>TOX21</th>
<th>BACE</th>
<th>BBBP</th>
<th>CLINTOX</th>
<th>SIDER</th>
<th>TOXCAST</th>
<th>HIV</th>
<th>ESOL</th>
<th>FREESOLV</th>
</tr>
</thead>
<tbody>
<tr>
<td>GCN</td>
<td>75.3±0.7</td>
<td>79.2±1.4</td>
<td>68.9±1.5</td>
<td>91.3±1.7</td>
<td>59.6±1.8</td>
<td>63.5±0.4</td>
<td>76.1±1.0</td>
<td>1.11±0.03</td>
<td>2.64±0.24</td>
</tr>
<tr>
<td>GCN-virtual</td>
<td>77.5±0.9</td>
<td>68.9±7.0</td>
<td>67.8±2.4</td>
<td>88.6±2.1</td>
<td>59.8±1.5</td>
<td>66.7±0.5</td>
<td>76.0±1.2</td>
<td>1.02±0.10</td>
<td>2.19±0.12</td>
</tr>
<tr>
<td>GIN</td>
<td>74.9±0.5</td>
<td>73.0±4.0</td>
<td>68.2±1.5</td>
<td>88.1±2.5</td>
<td>57.6±1.4</td>
<td>63.4±0.7</td>
<td>75.6±1.4</td>
<td>1.17±0.06</td>
<td>2.76±0.35</td>
</tr>
<tr>
<td>GIN-virtual</td>
<td>77.6±0.6</td>
<td>73.5±5.2</td>
<td>69.7±1.9</td>
<td>84.1±3.8</td>
<td>57.6±1.6</td>
<td>66.1±0.5</td>
<td>77.1±1.5</td>
<td>1.00±0.07</td>
<td>2.15±0.30</td>
</tr>
<tr>
<td>FactorGCN</td>
<td>57.8±2.1</td>
<td>70.0±0.6</td>
<td>54.1±1.1</td>
<td>64.2±2.1</td>
<td>53.3±1.7</td>
<td>51.2±0.8</td>
<td>57.1±1.5</td>
<td>3.39±0.15</td>
<td>5.69±0.32</td>
</tr>
<tr>
<td>$\Gamma_{\text{GIN}}$</td>
<td>52.3±1.1</td>
<td>54.5±0.9</td>
<td>51.8±1.5</td>
<td>52.2±1.1</td>
<td>51.3±1.1</td>
<td>50.7±0.5</td>
<td>51.3±0.9</td>
<td>4.15±0.10</td>
<td>7.34±0.12</td>
</tr>
<tr>
<td>PNA</td>
<td>71.5±0.5</td>
<td>77.4±2.1</td>
<td>66.2±1.2</td>
<td>81.2±2.0</td>
<td>59.6±1.1</td>
<td>60.6±0.2</td>
<td>79.1±1.3</td>
<td>0.94±0.02</td>
<td>2.92±0.16</td>
</tr>
<tr>
<td>TopKPool</td>
<td>75.6±0.9</td>
<td>76.9±2.4</td>
<td>68.6±1.1</td>
<td>86.9±1.1</td>
<td>60.6±1.5</td>
<td>64.7±0.1</td>
<td>76.7±1.1</td>
<td>1.17±0.03</td>
<td>2.08±0.10</td>
</tr>
<tr>
<td>SAGPool</td>
<td>74.7±3.1</td>
<td>76.6±1.0</td>
<td>69.3±2.1</td>
<td>88.7±1.0</td>
<td>61.3±1.3</td>
<td>64.8±0.2</td>
<td>77.7±1.3</td>
<td>1.22±0.05</td>
<td>2.28±0.12</td>
</tr>
<tr>
<td><strong>OOD-GNN</strong></td>
<td><strong>78.4±0.8</strong></td>
<td><strong>81.3±1.2</strong></td>
<td><strong>70.1±1.0</strong></td>
<td><strong>91.4±1.3</strong></td>
<td><strong>64.0±1.3</strong></td>
<td><strong>68.7±0.3</strong></td>
<td><strong>79.5±0.9</strong></td>
<td><strong>0.88±0.05</strong></td>
<td><strong>1.81±0.14</strong></td>
</tr>
</tbody>
</table>
Independence-promoted Disentangled Graph Contrastive Learning (IDGCL)

- Except the reweighting, OOD-GNN performs like a normal GNN
- The formation of a graph is typically driven by many entangled latent factors

Can we disentangle latent factors in the message passing?

- The graph labels can be extremely scarce for many graph datasets/scenarios
  Can we design self-supervised learning frameworks?

Disentangled Graph Contrastive Learning with Independence Promotion. *TKDE, 2022.*
**IDGCL: Method**

- **Key idea**: disentangled graph encoder + factor-wise contrastive learning + HSIC
  - Each channel for one disentangled factor

---

**Disentangled Graph Contrastive Learning with Independence Promotion.** *TKDE, 2022.*
IDGCL: Method

- **Graph Augmentation**
  - Four types of strategies: node dropping, edge perturbation, attribute masking, subgraph sampling
  - Reflect diverse aspects behind graphs, can be directly extended
  - Self-supervised loss: 
    \[
    p_{\theta}(y_i | x_i) = \frac{\exp{\phi(v_i, v'_i)}}{\sum_{j=1}^{N} \exp{\phi(v_i, v'_{y_j})}}
    \]

Disentangled Graph Contrastive Learning with Independence Promotion. **TKDE, 2022.**
IDGCL: Method

- Factor-wise message-passing
  - First, a shared GNN for a few layers
  - Then learn $K$ GNNs with independent parameters
  - Each channel only captures one hidden factor

Disentangled Graph Contrastive Learning with Independence Promotion. **TKDE, 2022.**

$$
H_{k}^{L+1} = \text{GNN}_k(H^L, A) \\
h_{G_i, k} = \text{READOUT}_k(\{H_{k}^{L+1}\}) \\
z_{i, k} = \text{MLP}_k(h_{G_i, k}).
$$
IDGCL: Method

- Factor-wise contrastive learning
  - Consider multiple latent factors
    \[
    p_\theta(y_i|G_i) = \mathbb{E}_{p_\theta(k|G_i)} \left[ p_\theta(y_i|G_i, k) \right]
    \]
  - Infer latent factors by $K$ prototypes:
    \[
    p_\theta(k|G_i) = \frac{\exp \phi(z_{i,k}, c_k)}{\sum_{k=1}^{K} \exp \phi(z_{i,k}, c_k)}
    \]
  - Subtask under each latent factor:
    \[
    p_\theta(y_i|G_i, k) = \frac{\exp \phi(z_{i,k}, z'_{y_i,k})}{\sum_{j=1}^{N} \exp \phi(z_{i,k}, z'_{y_j,k})}
    \]
- Statistical Independence regularizer:
  \[
  \mathcal{L}_{reg} = \sum_{1 \leq k_A < k_B \leq K} \text{HSIC}(z_{*,k_A}, z_{*,k_B})
  \]
- Overall objective function:
  \[
  \min_\theta \mathcal{L}(\theta, B) + \lambda \mathcal{L}_{reg}
  \]
IDGCL: Optimization

- Optimization
  - Maximize the joint probability \( \prod_{i=1}^{N} p(y_i | G_i) \)
  - Step 1: infer the posterior probability of latent factors with Bayes’ theorem:
    \[
    \theta^* = \arg \max_{\theta} \sum_{i=1}^{N} \log p_\theta(y_i | G_i) = \arg \max_{\theta} \sum_{i=1}^{N} \log \mathbb{E}_{p_\theta(k|G_i)} [p_\theta(y_i | G_i, k)].
    \]
  - Step 2: approximate the posterior probability with a variational distribution:
    \[
    p_\theta(k|G_i, y_i) = \frac{p_\theta(k|G_i)p_\theta(y_i | G_i, k)}{\sum_{k=1}^{K} p_\theta(k|G_i)p_\theta(y_i | G_i, k)} \quad p_\theta(y_i | G_i, k) = \frac{\exp \phi(z_{i,k}, z'_{y_i,k})}{\sum_{j=1}^{N} \exp \phi(z_{i,k}, z'_{y_j,k})}
    \]
  - Step 3: optimize the evidence lower bound (ELBO) of the log-likelihood
    \[
    q_\theta(k|G_i, y_i) = \frac{p_\theta(k|G_i)\hat{p}_\theta(y_i | G_i, k)}{\sum_{k=1}^{K} p_\theta(k|G_i)\hat{p}_\theta(y_i | G_i, k)} \quad \hat{p}_\theta(y_i | G_i, k) = \frac{\exp \phi(z_{i,k}, z'_{i,k})}{\sum_{j \in B, j \neq i} \exp \phi(z_{i,k}, z'_{j,k})}
    \]

**Theorem 1.** The log likelihood function of each graph \( \log p_\theta(y_i | G_i) \) is lower bounded by the ELBO:
    \[
    \mathcal{L}(\theta, i) = \mathbb{E}_{q_\theta(k|G_i, y_i)} [\log p_\theta(y_i | G_i, k)] - D_{KL}(q_\theta(k|G_i, y_i) \| p_\theta(k|G_i)).
    \]

- Step 4: calculate the independence regularizer
- Step 5: update parameters using gradient descends

Disentangled Graph Contrastive Learning with Independence Promotion. **TKDE, 2022.**
## IDGCL: Experiments

**Classification performance in benchmarks**

TABLE 2: Graph classification accuracy (%) of our proposed method and baselines in the unsupervised learning setting. In each column, the boldfaced score denotes the best result of all the methods and the underlined score represents the best result of baselines. "—" indicates the result is not reported in the paper.

<table>
<thead>
<tr>
<th></th>
<th>MUTAG</th>
<th>PTC-MR</th>
<th>PROTEINS</th>
<th>NCI1</th>
<th>IMDB-B</th>
<th>IMDB-M</th>
<th>RDT-B</th>
<th>RDT-M5K</th>
<th>COLLAB</th>
</tr>
</thead>
<tbody>
<tr>
<td>SP</td>
<td>85.2±2.4</td>
<td>58.2±2.4</td>
<td>75.1±0.5</td>
<td>73.0±0.2</td>
<td>55.6±0.2</td>
<td>38.0±0.3</td>
<td>64.1±0.1</td>
<td>39.6±0.2</td>
<td>—</td>
</tr>
<tr>
<td>GK</td>
<td>81.7±2.1</td>
<td>57.3±1.4</td>
<td>71.7±0.6</td>
<td>62.3±0.3</td>
<td>65.9±1.0</td>
<td>43.9±0.4</td>
<td>77.3±0.2</td>
<td>41.0±0.2</td>
<td>72.8±0.3</td>
</tr>
<tr>
<td>WL</td>
<td>80.7±3.0</td>
<td>58.0±0.5</td>
<td>72.9±0.6</td>
<td>80.0±0.5</td>
<td>72.3±3.4</td>
<td>47.0±0.5</td>
<td>68.8±0.4</td>
<td>46.1±0.2</td>
<td>—</td>
</tr>
<tr>
<td>DGK</td>
<td>87.4±2.7</td>
<td>60.1±2.6</td>
<td>73.3±0.8</td>
<td>80.3±0.5</td>
<td>67.0±0.6</td>
<td>44.6±0.5</td>
<td>78.0±0.4</td>
<td>41.3±0.2</td>
<td>73.1±0.3</td>
</tr>
<tr>
<td>MLG</td>
<td>87.9±1.6</td>
<td>63.3±1.5</td>
<td>76.1±2.0</td>
<td>80.8±1.3</td>
<td>66.6±0.3</td>
<td>41.2±0.0</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>node2vec</td>
<td>72.6±10.2</td>
<td>58.6±8.0</td>
<td>57.5±3.6</td>
<td>54.9±1.6</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>sub2vec</td>
<td>61.1±15.8</td>
<td>60.0±6.4</td>
<td>53.0±5.6</td>
<td>52.8±1.5</td>
<td>55.3±1.5</td>
<td>36.7±0.8</td>
<td>71.5±0.4</td>
<td>36.7±0.4</td>
<td>—</td>
</tr>
<tr>
<td>graph2vec</td>
<td>83.2±9.3</td>
<td>60.2±6.9</td>
<td>73.3±2.1</td>
<td>73.2±1.8</td>
<td>71.1±0.5</td>
<td>50.4±0.9</td>
<td>75.8±1.0</td>
<td>47.9±0.3</td>
<td>—</td>
</tr>
<tr>
<td>GVAE</td>
<td>87.7±0.7</td>
<td>61.2±1.8</td>
<td>74.4±0.3</td>
<td>76.2±1.1</td>
<td>73.0±0.9</td>
<td>49.7±0.5</td>
<td>82.5±1.4</td>
<td>53.5±1.0</td>
<td>70.7±1.1</td>
</tr>
<tr>
<td>InfoGraph</td>
<td>89.0±1.1</td>
<td>61.7±1.4</td>
<td>74.4±0.3</td>
<td>76.2±1.1</td>
<td>73.0±0.9</td>
<td>49.7±0.5</td>
<td>82.5±1.4</td>
<td>53.5±1.0</td>
<td>70.7±1.1</td>
</tr>
<tr>
<td>GCC</td>
<td>72.0</td>
<td>49.4</td>
<td>53.7</td>
<td>78.9</td>
<td>89.8</td>
<td>44.9</td>
<td>53.7</td>
<td>78.9</td>
<td></td>
</tr>
<tr>
<td>MVGRL</td>
<td>89.7±1.1</td>
<td>62.5±1.7</td>
<td></td>
<td></td>
<td>74.2±0.7</td>
<td>51.2±0.5</td>
<td>84.5±0.6</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>GraphCL</td>
<td>86.8±1.3</td>
<td>63.6±1.8</td>
<td>74.4±0.5</td>
<td>77.9±0.4</td>
<td>71.1±0.4</td>
<td>50.7±0.4</td>
<td>89.5±0.8</td>
<td>56.0±0.3</td>
<td>71.4±1.2</td>
</tr>
<tr>
<td>JOAO</td>
<td>87.7±0.8</td>
<td>61.1±1.7</td>
<td>74.1±1.1</td>
<td>78.4±0.5</td>
<td>70.8±0.3</td>
<td>51.0±0.5</td>
<td>86.4±1.5</td>
<td>56.0±0.3</td>
<td>69.3±0.3</td>
</tr>
<tr>
<td>DGCL</td>
<td>92.1±0.8</td>
<td>65.8±1.5</td>
<td>76.4±0.5</td>
<td>81.9±0.2</td>
<td>75.9±0.7</td>
<td>51.9±0.4</td>
<td>91.8±0.2</td>
<td>56.1±0.2</td>
<td>81.2±0.3</td>
</tr>
<tr>
<td>IDGCL</td>
<td>92.5±0.6</td>
<td>66.2±1.3</td>
<td>77.1±0.2</td>
<td>82.4±0.3</td>
<td>76.1±0.2</td>
<td>52.3±0.4</td>
<td>91.9±0.3</td>
<td>56.3±0.2</td>
<td>81.3±0.3</td>
</tr>
</tbody>
</table>
IDGCL: Experiments

- OOD graph datasets

Fig. 5: Graph classification results of our proposed method and baselines in the semi-supervised learning setting.
IDGCL: Experiments

- Visualizations for representations

MVGRL  GraphCL

IDGCL

(a) factor $p = 0.2$, 1st channel.  (b) factor $p = 0.2$, 5th channel.
(c) factor $p = 0.9$, 1st channel.  (d) factor $p = 0.9$, 5th channel.

Each channel captures one latent factor

Disentangled Graph Contrastive Learning with Independence Promotion. *TKDE, 2022.*
Handling Distribution Shifts

How to handle distribution shifts in GNN architectures?

How to handle distribution shifts in the topology space?

How to handle distribution shifts in the vector space?

Feature representation, embedding

$\mathbb{R}^d$

Picture credit: Kipf and Welling (GCN, ICLR 2017), Jure Leskovec (CS224w, Stanford Univ.)
Graph Neural Architecture Search (NAS)

- Goal: automatically learn the best neural architecture

- Key designs

FBNet: Hardware-Aware Efficient ConvNet Design via Differentiable Neural Architecture Search, CVPR 2019
Neural Architecture Search A Survey, JMLR 2019
Customize a unique GNN architecture for each graph instance to handle distribution shifts.
Goal: learn a vector representation for each graph to reflect its characteristics

Challenge: preserve diverse properties of the original graph

Method: self-supervised disentangled graph encoder

- Encoder: disentangled GNN
- Supervised loss: the downstream task
- Self-supervised loss: node degree as regularization
**Goal**: customize an architecture based on the graph representation

**Assumption**: graphs with similar characteristics need similar architectures

**Method**: prototype based architecture customization

- Probabilities of choosing operations:
  \[ p_o = \text{softmax}(h \cdot \frac{q_o}{\|q_o\|_2}) \]

- Regularizer to avoid mode collapse:
  \[ \mathcal{L}_{\text{cos}} = \sum_i \sum_{o,o' \in \mathcal{O}, o \neq o'} \frac{q_o^i \cdot q_{o'}^i}{\|q_o^i\|_2 \cdot \|q_{o'}^i\|_2} \]
GRACES: Learning Architecture Parameters

- **Goal**: learn parameters for the customized architectures
- **Method**: customized super-network
- **Loss functions**:
  \[
  \mathcal{L} = \gamma \mathcal{L}_{\text{main}} + (1 - \gamma) \mathcal{L}_{\text{reg}}
  \]
  \[
  \mathcal{L}_{\text{reg}} = \mathcal{L}_{\text{sup}} + \beta_1 \mathcal{L}_{\text{ssl}} + \beta_2 \mathcal{L}_{\text{cos}}
  \]
## GRACES: Experiments

### Synthetic OOD graph datasets

<table>
<thead>
<tr>
<th>bias</th>
<th>$b = 0.7$</th>
<th>$b = 0.8$</th>
<th>$b = 0.9$</th>
</tr>
</thead>
<tbody>
<tr>
<td>GCN</td>
<td>48.39$\pm$1.69</td>
<td>41.55$\pm$3.88</td>
<td>39.13$\pm$1.76</td>
</tr>
<tr>
<td>GAT</td>
<td>50.75$\pm$4.89</td>
<td>42.48$\pm$2.46</td>
<td>40.10$\pm$5.19</td>
</tr>
<tr>
<td>GIN</td>
<td>36.83$\pm$5.49</td>
<td>34.83$\pm$3.10</td>
<td>37.45$\pm$3.59</td>
</tr>
<tr>
<td>SAGE</td>
<td>46.66$\pm$2.51</td>
<td>44.50$\pm$5.79</td>
<td>44.79$\pm$4.83</td>
</tr>
<tr>
<td>GraphConv</td>
<td>47.29$\pm$1.95</td>
<td>44.67$\pm$5.88</td>
<td>44.82$\pm$4.84</td>
</tr>
<tr>
<td>MLP</td>
<td>48.27$\pm$1.27</td>
<td>46.73$\pm$3.48</td>
<td>46.41$\pm$2.34</td>
</tr>
<tr>
<td>ASAP</td>
<td>54.07$\pm$13.85</td>
<td>48.32$\pm$12.72</td>
<td>43.52$\pm$8.41</td>
</tr>
<tr>
<td>DIR</td>
<td>50.08$\pm$3.46</td>
<td>48.22$\pm$6.27</td>
<td>43.11$\pm$5.43</td>
</tr>
<tr>
<td>random</td>
<td>45.92$\pm$4.29</td>
<td>51.72$\pm$5.38</td>
<td>45.89$\pm$5.09</td>
</tr>
<tr>
<td>DARTS</td>
<td>50.63$\pm$8.90</td>
<td>45.41$\pm$7.71</td>
<td>44.44$\pm$4.42</td>
</tr>
<tr>
<td>GNAS</td>
<td>55.18$\pm$18.62</td>
<td>51.64$\pm$19.22</td>
<td>37.56$\pm$5.43</td>
</tr>
<tr>
<td>PAS</td>
<td>52.15$\pm$4.35</td>
<td>43.12$\pm$5.95</td>
<td>39.84$\pm$1.67</td>
</tr>
<tr>
<td><strong>GRACES</strong></td>
<td><strong>65.72$\pm$17.47</strong></td>
<td><strong>59.57$\pm$17.37</strong></td>
<td><strong>50.94$\pm$8.14</strong></td>
</tr>
</tbody>
</table>

### Real-world OOD graph datasets

<table>
<thead>
<tr>
<th>dataset</th>
<th>hiv</th>
<th>sider</th>
<th>bace</th>
</tr>
</thead>
<tbody>
<tr>
<td>GCN</td>
<td>75.99$\pm$1.19</td>
<td>59.84$\pm$1.54</td>
<td>68.93$\pm$6.95</td>
</tr>
<tr>
<td>GAT</td>
<td>76.80$\pm$0.58</td>
<td>57.40$\pm$2.01</td>
<td>75.34$\pm$2.36</td>
</tr>
<tr>
<td>GIN</td>
<td>77.07$\pm$1.49</td>
<td>57.57$\pm$1.56</td>
<td>73.46$\pm$5.24</td>
</tr>
<tr>
<td>SAGE</td>
<td>75.58$\pm$1.40</td>
<td>56.36$\pm$1.32</td>
<td>74.85$\pm$2.74</td>
</tr>
<tr>
<td>GraphConv</td>
<td>74.46$\pm$0.86</td>
<td>56.09$\pm$1.06</td>
<td>78.87$\pm$1.74</td>
</tr>
<tr>
<td>MLP</td>
<td>70.88$\pm$0.83</td>
<td>58.16$\pm$1.41</td>
<td>71.60$\pm$2.30</td>
</tr>
<tr>
<td>ASAP</td>
<td>73.81$\pm$1.17</td>
<td>55.77$\pm$1.18</td>
<td>71.55$\pm$2.74</td>
</tr>
<tr>
<td>DIR</td>
<td>77.05$\pm$0.57</td>
<td>57.34$\pm$0.36</td>
<td>76.03$\pm$2.20</td>
</tr>
<tr>
<td>DARTS</td>
<td>74.04$\pm$1.75</td>
<td>60.64$\pm$1.37</td>
<td>76.71$\pm$1.83</td>
</tr>
<tr>
<td>PAS</td>
<td>71.19$\pm$2.28</td>
<td>59.31$\pm$1.48</td>
<td>76.59$\pm$1.87</td>
</tr>
<tr>
<td><strong>GRACES</strong></td>
<td><strong>77.31$\pm$1.00</strong></td>
<td><strong>61.85$\pm$2.56</strong></td>
<td><strong>79.46$\pm$3.04</strong></td>
</tr>
</tbody>
</table>

### Customization of architectures

(a) *Tree*-based graphs  
(b) *Ladder*-based graphs  
(c) *Wheel*-based graphs
Handling Distribution Shifts

How to handle distribution shifts in
GNN architectures?

How to handle distribution shifts in the topology space?

How to handle distribution shifts in the vector space?

Feature representation, embedding

Picture credit: Kipf and Welling (GCN, ICLR 2017), Jure Leskovec (CS224w, Stanford Univ.)
Graph Invariant Learning (GIL)

- How to get rid of spurious correlations in the topology space?
- Main idea: distinguish invariant and variant subgraphs
  - Invariant: relationships with labels are stable under distribution shifts
  - Variant: the complement of invariant, e.g., environments

- Challenge:
  - There is no labels for invariant and variant subgraphs
  - Variant and invariant subgraphs are highly entangled

Learning Invariant Graph Representations under Distribution Shifts. *NeurIPS, 2022.*
**GIL: Method**

- **Key Idea**: mutual promotion of invariant learning and environment (variant) inference
  - Invariant subgraphs: for predicting labels
  - Variant subgraphs: for providing environments

---

**Invariant Subgraph Identification** 

- Graph dataset 
- Node representation $Z^{(m)}$ 
- Edge Mask $M_{ij}$ 
- $A_I = \text{Top}_t(M \odot A)$ 
- $A_Y = A - A_I$ 

**Environment Inference**

- Cluster variant subgraphs $\{G_v\}$ to infer environments 

**Invariant Learning**

- $w \circ g(\cdot)$ 
- Classification loss $\mathcal{L}_{\text{cls}}$ 
- Invariance regularizer $\mathcal{L}_{\text{inv}}$ 
- Objective function $\mathbb{E}_{\text{inver}}[\text{Var}(\nabla_{\theta} \mathcal{R}^c)]$

**Prediction**

$\hat{Y} = f(G) = w \circ g \circ \Phi(G)$
Goal: learn a mask to separate invariant and variant subgraphs

Challenge: need to handle graphs of various sizes and be inductive

Proposed method: GNN with top-t pooling

\[ Z^{(m)} = GNN^M(G) \quad M_{i,j} = Z_i^{(m)^T} \cdot Z_j^{(m)} \quad A_I = \text{Top}_t(M \odot A), \quad A_V = A - A_I \]
Assumption: the variant subgraphs capture environment-discriminatve features

Challenge: there is no ground-truth environment labels

Proposed method: cluster variant subgraphs infer environments, e.g., k-means

$$\mathcal{E}_{infer} = \text{k-means}(H)$$
**GIL: Method**

- **Goal:** find an invariant subgraph generator \( \mathcal{I}_E = \{ \Phi(\cdot) : P^e(Y|\Phi(G)) = P^{e'}(Y|\Phi(G)), e, e' \in \text{supp}(E) \} \)

- **Optimization:**

**Theorem 3.2.** A generator \( \Phi(G) \) is the optimal generator if and only

\[
\Phi^* = \arg \max_{\Phi \in \mathcal{I}_E} I(Y; \Phi(G)),
\]

where \( I(\cdot; \cdot) \) is the mutual information between the label and the generated subgraph.

- **Invariance regularizer:**

\[
\mathbb{E}_{e \in \text{supp}(\mathcal{E}_{\text{infer}})} \mathcal{R}^e (f(G), Y; \theta) + \lambda \text{trace}(\text{Var}_{\mathcal{E}_{\text{infer}}} (\nabla_\theta \mathcal{R}^e))
\]
GIL: Theory

We prove that the maximal invariant subgraph generator can achieve OOD optimal

**Theorem 4.1.** Let $\Phi^*$ be the optimal invariant subgraph generator in Assumption 3.1 and denote the complement as $G \setminus \Phi^*(G)$, i.e., the corresponding variant subgraph. Then, we can obtain the optimal predictor under distribution shifts, i.e., the solution to Problem 1, as follows:

$$\arg \min_{w, g} w \circ g \circ \Phi^*(G) = \arg \min_{f} \sup_{e \in \text{supp}(\mathcal{E})} \mathcal{R}(f|e),$$

(10)

Several assumptions:

1. $\Phi^*(G) \perp G \setminus \Phi^*(G)$
2. $\forall \Phi \in \mathcal{I}_\mathcal{E}, \exists e' \in \text{supp}(\mathcal{E})$ such that $P^{e'}(\Phi(G)) = P^e(\Phi(G))$
   and $P^{e'}(G, Y) = P^{e'}(\Phi(G), Y)P^{e'}(G \setminus \Phi(G))$

We prove that GIL maintains permutation invariance.

**Theorem 4.2.** Our proposed GIL model is permutation-invariant if GNN$^M$ and GNN$^I$ are permutation-equivariant and READOUT$^I$ is permutation-invariant.

We show that the time complexity of GIL is on par with the existing GNNs

- Time Complexity: $O(|E|d + |V|d^2)$, $|E|$ and $|V|$ are the edge and node number.
GIL: Experiments

- OOD Generalization on synthetic datasets (Spurious-Motif)
  - Each graph includes one invariant motif (i.e., label) and variant motif (i.e., spurious part).
  - $r$ controls the bias strength; $|r_{test} - r_{train}|$ is the distribution shift strength.

<table>
<thead>
<tr>
<th>$r_{train}$</th>
<th>Scenario 1: $r_{test} = 1/3$</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$r = 1/3$</td>
<td>$r = 0.5$</td>
<td>$r = 0.6$</td>
<td>$r = 0.7$</td>
<td>$r = 0.8$</td>
<td>$r = 0.9$</td>
<td></td>
</tr>
<tr>
<td>ERM</td>
<td>53.60±3.79</td>
<td>51.24±4.13</td>
<td>47.04±7.01</td>
<td>38.80±3.72</td>
<td>37.84±3.01</td>
<td>37.44±2.15</td>
<td></td>
</tr>
<tr>
<td>Attention</td>
<td>54.31±3.98</td>
<td>53.24±3.56</td>
<td>42.52±6.20</td>
<td>35.20±1.05</td>
<td>34.48±1.18</td>
<td>33.88±1.01</td>
<td></td>
</tr>
<tr>
<td>Top-k Pool</td>
<td>54.68±2.71</td>
<td>53.12±5.58</td>
<td>44.56±4.57</td>
<td>37.44±2.04</td>
<td>35.24±2.28</td>
<td>34.28±4.11</td>
<td></td>
</tr>
<tr>
<td>SAG Pool</td>
<td>54.08±3.66</td>
<td>52.60±3.52</td>
<td>44.68±5.25</td>
<td>37.68±4.03</td>
<td>34.28±1.82</td>
<td>32.72±1.83</td>
<td></td>
</tr>
<tr>
<td>ASAP</td>
<td>54.00±4.21</td>
<td>51.92±3.81</td>
<td>45.12±1.98</td>
<td>36.28±0.86</td>
<td>34.24±2.02</td>
<td>34.40±3.15</td>
<td></td>
</tr>
<tr>
<td>GroupDRO</td>
<td>53.20±4.91</td>
<td>51.40±4.35</td>
<td>48.32±5.35</td>
<td>39.12±4.27</td>
<td>38.40±2.76</td>
<td>37.64±1.69</td>
<td></td>
</tr>
<tr>
<td>IRM</td>
<td>52.00±2.34</td>
<td>50.60±3.54</td>
<td>47.84±6.95</td>
<td>38.80±3.72</td>
<td>39.84±3.21</td>
<td>39.00±3.98</td>
<td></td>
</tr>
<tr>
<td>V-REx</td>
<td>53.16±3.25</td>
<td>46.04±6.11</td>
<td>45.36±3.66</td>
<td>40.24±3.86</td>
<td>39.48±3.00</td>
<td>39.12±3.48</td>
<td></td>
</tr>
<tr>
<td>DIR</td>
<td>52.96±5.06</td>
<td>52.08±1.93</td>
<td>50.12±2.76</td>
<td>49.84±2.86</td>
<td>45.20±1.11</td>
<td>41.24±4.73</td>
<td></td>
</tr>
<tr>
<td><strong>GIL</strong></td>
<td>55.44±3.11</td>
<td>54.56±3.02</td>
<td>53.60±4.82</td>
<td>53.12±2.18</td>
<td>51.24±3.88</td>
<td>46.04±3.51</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$r_{train}$</th>
<th>Scenario 2: $r_{test} = 0.2$</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$r = 1/3$</td>
<td>$r = 0.5$</td>
<td>$r = 0.6$</td>
<td>$r = 0.7$</td>
<td>$r = 0.8$</td>
<td>$r = 0.9$</td>
<td></td>
</tr>
<tr>
<td>ERM</td>
<td>48.48±4.53</td>
<td>41.72±4.81</td>
<td>36.92±6.93</td>
<td>35.72±8.33</td>
<td>28.80±3.91</td>
<td>19.60±1.66</td>
<td></td>
</tr>
<tr>
<td>Attention</td>
<td>44.04±4.33</td>
<td>31.64±6.67</td>
<td>25.72±5.34</td>
<td>24.80±4.06</td>
<td>23.20±3.60</td>
<td>18.04±2.88</td>
<td></td>
</tr>
<tr>
<td>Top-k Pool</td>
<td>45.68±5.16</td>
<td>34.20±4.34</td>
<td>31.00±2.89</td>
<td>30.64±3.59</td>
<td>29.16±2.18</td>
<td>27.56±3.91</td>
<td></td>
</tr>
<tr>
<td>SAG Pool</td>
<td>44.36±6.09</td>
<td>38.64±3.02</td>
<td>31.36±4.40</td>
<td>32.84±1.86</td>
<td>28.72±3.11</td>
<td>26.60±5.37</td>
<td></td>
</tr>
<tr>
<td>ASAP</td>
<td>49.88±4.90</td>
<td>34.52±4.35</td>
<td>27.00±2.61</td>
<td>27.20±2.53</td>
<td>27.96±3.89</td>
<td>22.88±4.33</td>
<td></td>
</tr>
<tr>
<td>GroupDRO</td>
<td>52.68±4.04</td>
<td>43.68±4.05</td>
<td>31.92±6.84</td>
<td>34.36±8.41</td>
<td>28.88±5.14</td>
<td>20.32±1.64</td>
<td></td>
</tr>
<tr>
<td>IRM</td>
<td>50.24±6.73</td>
<td>41.60±4.75</td>
<td>35.24±5.35</td>
<td>34.92±8.03</td>
<td>29.44±5.47</td>
<td>21.84±3.57</td>
<td></td>
</tr>
<tr>
<td>V-REx</td>
<td>50.56±2.83</td>
<td>37.16±6.24</td>
<td>34.52±3.00</td>
<td>29.72±4.58</td>
<td>27.32±3.18</td>
<td>24.04±6.08</td>
<td></td>
</tr>
<tr>
<td>DIR</td>
<td>50.68±5.20</td>
<td>49.96±1.75</td>
<td>45.44±5.00</td>
<td>40.56±2.36</td>
<td>39.92±4.53</td>
<td>32.52±4.59</td>
<td></td>
</tr>
<tr>
<td><strong>GIL</strong></td>
<td>54.80±3.93</td>
<td>52.48±4.41</td>
<td>50.08±5.47</td>
<td>47.44±2.87</td>
<td>46.36±3.80</td>
<td>35.80±5.03</td>
<td></td>
</tr>
</tbody>
</table>
GIL: Experiments

- OOD Generalization on real-world datasets

<table>
<thead>
<tr>
<th></th>
<th>MNIST-75sp</th>
<th>Graph-SST2</th>
<th>MOLSIDER</th>
<th>MOL HIV</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERM</td>
<td>14.94 ± 3.27</td>
<td>81.44 ± 0.59</td>
<td>57.57 ± 1.56</td>
<td>76.20 ± 1.14</td>
</tr>
<tr>
<td>Attention</td>
<td>16.44 ± 3.78</td>
<td>81.57 ± 0.71</td>
<td>56.99 ± 0.54</td>
<td>75.84 ± 1.33</td>
</tr>
<tr>
<td>Top-k Pool</td>
<td>15.02 ± 3.08</td>
<td>79.78 ± 1.35</td>
<td>60.63 ± 1.52</td>
<td>73.01 ± 1.65</td>
</tr>
<tr>
<td>SAG Pool</td>
<td>19.34 ± 1.73</td>
<td>80.24 ± 1.72</td>
<td>61.29 ± 1.31</td>
<td>73.26 ± 0.84</td>
</tr>
<tr>
<td>ASAP</td>
<td>15.14 ± 3.58</td>
<td>81.57 ± 0.84</td>
<td>55.77 ± 1.34</td>
<td>73.81 ± 1.17</td>
</tr>
<tr>
<td>GroupDRO</td>
<td>15.72 ± 4.35</td>
<td>81.29 ± 1.44</td>
<td>56.31 ± 1.15</td>
<td>75.44 ± 2.70</td>
</tr>
<tr>
<td>IRM</td>
<td>18.74 ± 2.43</td>
<td>81.01 ± 1.13</td>
<td>57.10 ± 0.92</td>
<td>74.46 ± 2.74</td>
</tr>
<tr>
<td>V-REx</td>
<td>18.40 ± 1.12</td>
<td>81.76 ± 0.08</td>
<td>57.76 ± 0.78</td>
<td>75.62 ± 0.79</td>
</tr>
<tr>
<td>DIR</td>
<td>17.38 ± 3.52</td>
<td>83.29 ± 0.53</td>
<td>57.74 ± 1.63</td>
<td>77.05 ± 0.57</td>
</tr>
<tr>
<td>GSAT</td>
<td>20.12 ± 1.35</td>
<td>82.95 ± 0.58</td>
<td>60.82 ± 1.36</td>
<td>76.47 ± 1.53</td>
</tr>
<tr>
<td><strong>GIL</strong></td>
<td><strong>21.94 ± 0.38</strong></td>
<td><strong>83.44 ± 0.37</strong></td>
<td><strong>63.50 ± 0.57</strong></td>
<td><strong>79.08 ± 0.54</strong></td>
</tr>
</tbody>
</table>

- ogbg-molhiv

<table>
<thead>
<tr>
<th></th>
<th>CIN (Rank #8)</th>
<th>GIL (CIN Backbone)</th>
<th>HIG (Rank #2)</th>
<th>PAS+FPs (Rank #1)</th>
<th>GIL (HIG Backbone)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>80.94 ± 0.57</td>
<td><strong>81.15 ± 0.46</strong></td>
<td>84.03 ± 0.21</td>
<td>84.20 ± 0.15</td>
<td><strong>84.23 ± 0.25</strong></td>
</tr>
</tbody>
</table>

Compatible with various backbone GNNs and a new SOTA on OGB leaderboard!
GIL: Experiments

- Showcase on Spurious-Motif datasets
  
  - (a) Top-k Pool
  - (b) SAG Pool
  - (c) DIR
  - (d) GSAT
  - (e) GIL
  - (f) Ground Truth

- Showcases on Graph-SST2 (human-understandable)

  \[ \begin{array}{c}
  \text{Train set} \\
  \text{negative} \quad \text{positive}
  \end{array} \]
  
  - the
  - 's
  - actors
  - world
  - best
  - daniel
  - auteuil

  \[ \begin{array}{c}
  \text{Test set} \\
  \text{negative} \quad \text{positive}
  \end{array} \]
  
  - a
  - romantic
  - delightful
  - comedy
  - an
  - stupid
  - unbelievably
  - film

  Capture the subgraphs with positive/negative semantics

Disentangled Intervention based Dynamic Graph Attention Network (DIDA)

- Many graphs are dynamic in nature

- Distribution shifts can be spatio-temporal

Picture credit: ROLAND: graph learning framework for dynamic graphs, KDD 2022

Dynamic Graph Neural Networks Under Spatio-Temporal Distribution Shift. *NeurIPS, 2022.*
**DIDA: Method**

- **Key Idea**: finding invariant/variant spatial-temporal patterns and apply intervention
- Intervention: from causal theory to get rid of spurious correlation

---

**DIDM: Method**

- Dynamic Graph Neural Networks Under Spatio-Temporal Distribution Shift. *NeurIPS, 2022.*

---

**Picture credit:** Discovering Invariant Ratios for Graph Neural Networks, ICLR 2022

---

**Judea Pearl**

- 2011 Turing Award

---

**Dynamic Graph Neural Networks Under Spatio-Temporal Distribution Shift.
NeurIPS, 2022.**
DIDA: Method

- Goal: separate invariant and variant spatial-temporal subgraphs
- Proposed method: disentangled dynamic graph attention network
  - First calculate masks
    \[
    q_{tu} = W_q(h_{tu} || \text{TE}(t)),
    k_{tv} = W_k(h_{tv} || \text{TE}(t')),
    v_{tv} = W_v(h_{tv} || \text{TE}(t'))
    \]
    
    \[
    m_I = \text{Softmax}\left(\frac{q \cdot k}{\sqrt{d}}\right),
    m_V = \text{Softmax}\left(-\frac{q \cdot k}{\sqrt{d}}\right)
    \]
  - Then calculate message-passing
    \[
    z_I^t(u) = \text{Agg}_I(m_I, v \odot m_f)
    \]
    \[
    z_V^t(u) = \text{Agg}_V(m_V, v)
    \]
  - Updating node representation
    \[
    h_{tu}^t \leftarrow z_I^t(u) + z_V^t(u)
    \]
DIDA: Method

- Goal: create intervened distributions by sampling and reassembling variant patterns

\[ z_{I}^{t_1}(u), z_{V}^{t_1}(u) \leftarrow z_{I}^{t_1}(u), z_{V}^{t_2}(v) \]

Dynamic Graph Neural Networks Under Spatio-Temporal Distribution Shift. *NeurIPS, 2022.*
DIDA: Method

- Goal: focus on invariant patterns using intervened distributions
- Original objective:
  \[
  \min_{\theta_1, \theta_2} \mathbb{E}_{(y^t, g_{v^t}^1:t) \sim p_{tr}(y^t, g_{v^t}^1:t)} \mathcal{L}(f_{\theta_1}(\tilde{P}_I^t(v)), y^t)
  \]

- Practical version: intervention-invariant regularization
  \[
  \min_{\theta} \mathcal{L} + \lambda \mathcal{L}_{do}
  \]
  \[
  L = \ell(f(z_I), y)
  \]
  \[
  \mathcal{L}_m = \ell(g(z_V, z_I), y)
  \]
  \[
  \mathcal{L}_{do} = \text{Var}_{s_i \in S} (\mathcal{L}_m | \text{do}(P_V^t = s_i))
  \]
DIDA: Experiments

- **Synthetic datasets**

<table>
<thead>
<tr>
<th>Model (\bar{p})</th>
<th>0.4</th>
<th></th>
<th>0.6</th>
<th></th>
<th>0.8</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Split</td>
<td>Train</td>
<td>Test</td>
<td>Train</td>
<td>Test</td>
<td>Train</td>
<td>Test</td>
</tr>
<tr>
<td>GCRN</td>
<td>69.60 ± 1.14</td>
<td>72.57 ± 0.72</td>
<td>74.71 ± 0.17</td>
<td>72.29 ± 0.47</td>
<td>75.69 ± 0.07</td>
<td>67.26 ± 0.22</td>
</tr>
<tr>
<td>EGCN</td>
<td>78.82 ± 1.40</td>
<td>69.00 ± 0.53</td>
<td>79.47 ± 1.68</td>
<td>62.70 ± 1.14</td>
<td>81.07 ± 4.10</td>
<td>60.13 ± 0.89</td>
</tr>
<tr>
<td>DySAT</td>
<td>84.71 ± 0.80</td>
<td>70.24 ± 1.26</td>
<td>89.77 ± 0.32</td>
<td>64.01 ± 0.19</td>
<td>94.02 ± 1.29</td>
<td>62.19 ± 0.39</td>
</tr>
<tr>
<td>IRM</td>
<td>85.20 ± 0.07</td>
<td>69.40 ± 0.09</td>
<td>89.48 ± 0.22</td>
<td>63.97 ± 0.37</td>
<td>95.02 ± 0.09</td>
<td>62.66 ± 0.33</td>
</tr>
<tr>
<td>VREx</td>
<td>84.77 ± 0.84</td>
<td>70.44 ± 1.08</td>
<td>89.81 ± 0.21</td>
<td>63.99 ± 0.21</td>
<td>94.06 ± 1.30</td>
<td>62.21 ± 0.40</td>
</tr>
<tr>
<td>GroupDRO</td>
<td>84.78 ± 0.85</td>
<td>70.30 ± 1.23</td>
<td>89.90 ± 0.11</td>
<td>64.05 ± 0.21</td>
<td>94.08 ± 1.33</td>
<td>62.13 ± 0.35</td>
</tr>
<tr>
<td><strong>DIDA</strong></td>
<td><strong>87.92 ± 0.92</strong></td>
<td><strong>85.20 ± 0.84</strong></td>
<td><strong>91.22 ± 0.59</strong></td>
<td><strong>82.89 ± 0.23</strong></td>
<td><strong>92.72 ± 2.16</strong></td>
<td><strong>72.59 ± 3.31</strong></td>
</tr>
</tbody>
</table>

- **Real-world datasets**

<table>
<thead>
<tr>
<th>Model</th>
<th>COLLAB</th>
<th>Yelp</th>
<th>Transaction</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>w/o DS</td>
<td>w/ DS</td>
<td>w/o DS</td>
</tr>
<tr>
<td>Test Data</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GAE</td>
<td>77.15 ± 0.50</td>
<td>74.04 ± 0.75</td>
<td>70.67 ± 1.11</td>
</tr>
<tr>
<td>VGAE</td>
<td>86.47 ± 0.04</td>
<td>74.95 ± 1.25</td>
<td>76.54 ± 0.50</td>
</tr>
<tr>
<td>GCRN</td>
<td>82.78 ± 0.54</td>
<td>69.72 ± 0.45</td>
<td>68.59 ± 1.05</td>
</tr>
<tr>
<td>EGCN</td>
<td>86.62 ± 0.95</td>
<td>76.15 ± 0.91</td>
<td>78.21 ± 0.03</td>
</tr>
<tr>
<td>DySAT</td>
<td>88.77 ± 0.23</td>
<td>76.59 ± 0.20</td>
<td>78.87 ± 0.57</td>
</tr>
<tr>
<td>IRM</td>
<td>87.96 ± 0.90</td>
<td>75.42 ± 0.87</td>
<td>66.49 ± 10.78</td>
</tr>
<tr>
<td>VREx</td>
<td>88.31 ± 0.32</td>
<td>76.24 ± 0.77</td>
<td>79.04 ± 0.16</td>
</tr>
<tr>
<td>GroupDRO</td>
<td>88.76 ± 0.12</td>
<td>76.33 ± 0.29</td>
<td>79.38 ± 0.42</td>
</tr>
<tr>
<td><strong>DIDA</strong></td>
<td><strong>91.97 ± 0.05</strong></td>
<td><strong>81.87 ± 0.40</strong></td>
<td><strong>78.22 ± 0.40</strong></td>
</tr>
</tbody>
</table>
DIDA: Experiments

- **Showcases:**
  - Dynamic Graph Neural Networks Under Spatio-Temporal Distribution Shift. *NeurIPS, 2022.*

- **Ablation studies:**

Dynamic Graph Neural Networks Under Spatio-Temporal Distribution Shift. *NeurIPS, 2022.*
Recap

Out-of-distribution Generalization Graph Learning

Encourage Independence and Disentanglement in representations

Customize unique GNN architectures for graphs

Finding invariant and variant graph structures

OOD-GNN: Out-of-Distribution Generalized Graph Neural Network

TKDE’22

Self-supervised

Disentangled Graph Contrastive Learning with Independence Promotion

TKDE’22

Graph Neural Architecture Search under Distribution Shifts

ICML’22

Learning Invariant Graph Representations under Distribution Shifts

NeurIPS’22

Dynamic Graph Neural Networks under Spatio-Temporal Distribution Shift

NeurIPS’22
Out-Of-Distribution Generalization on Graphs: A Survey

Haoyang Li*, Xin Wang*, Ziwei Zhang and Wenwu Zhu
Tsinghua University, Beijing, China
lihy18@mails.tsinghua.edu.cn, {xin.wang, zwzhang, wzwu}@tsinghua.edu.cn

Abstract
Graph machine learning has been extensively studied in both academia and industry. Although booming with a vast number of emerging methods and techniques, most of the literature is built on the I.I.D. hypothesis, i.e., testing and training graph data are independent and identically distributed. However, this I.I.D. hypothesis can hardly be satisfied in many real-world graph scenarios where the model performance substantially degrades when there exist distribution shifts between testing and training graph data. To solve this critical problem, out-of-distribution (OOD) generalization on graphs, which goes beyond the I.I.D. hypothesis, has made great progress and attracted ever-

especially for graph neural networks (GNNs), have shown great successes in both academia and industry, illustrating their excellent capabilities in various applications, e.g., social network analysis [Qiu et al., 2018], recommendation systems [Wu et al., 2020], knowledge representation [Wang et al., 2017], traffic forecasting [Yu et al., 2018], etc.

Despite the notable success of graph machine learning approaches, the existing literature generally relies on the assumption that the testing and training graph data are independently drawn from the identical distribution, i.e., I.I.D. hypothesis. However, in real-world scenarios, such a hypothesis is difficult to be satisfied due to the uncontrollable underlying data generation mechanism [Bengio et al., 2019]. In practice, there will inevitably be scenarios with distribution shifts between testing and training graphs [Li et al., 2021b; Wang et al., 2021b], which may cause significant perfor-


Paper collection: https://github.com/THUMNLab/awesome-graph-ood
Acknowledgements

Wenwu Zhu
Tsinghua Univ.

Peng Cui
Tsinghua Univ.

Haoyang Li
Tsinghua Univ.

Yijian Qin
Tsinghua Univ.

Xin Wang
Tsinghua Univ.

Zeyang Zhang
Tsinghua Univ.

Pengtao Xie
UCSD
THANK YOU!

https://zw-zhang.github.io
zwzhang@tsinghua.edu.cn