Learning to Count Isomorphisms with Graph Neural Networks

Xingtong Yu\textsuperscript{1*}, Zemin Liu\textsuperscript{2*}, Yuan Fang\textsuperscript{3†}, Xinming Zhang \textsuperscript{1†}

\textsuperscript{1} University of Science and Technology of China, China
\textsuperscript{2} National University of Singapore, Singapore
\textsuperscript{3} Singapore Management University, Singapore

In Proceeding of 37\textsuperscript{th} AAAI Conference on Artificial Intelligence, February 7-14, 2023

* Co-first author
† Corresponding author
Outline

1. Introduction
2. Proposed Model: Count-GNN
3. Experiments
4. Conclusions
PART 01
Introduction
Introduction

01
Isomorphism Counting

02
Related Work

03
Contribution
## Subgraph Isomorphisms Counting

<table>
<thead>
<tr>
<th>Pattern</th>
<th>Graph</th>
<th>Count</th>
<th>Pattern</th>
<th>Graph</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Pattern 1" /></td>
<td><img src="image2.png" alt="Graph 1" /></td>
<td>0</td>
<td><img src="image3.png" alt="Pattern 2" /></td>
<td><img src="image4.png" alt="Graph 2" /></td>
<td>0</td>
</tr>
<tr>
<td><img src="image5.png" alt="Pattern 3" /></td>
<td><img src="image6.png" alt="Graph 3" /></td>
<td>4</td>
<td><img src="image7.png" alt="Pattern 4" /></td>
<td><img src="image8.png" alt="Graph 4" /></td>
<td>1</td>
</tr>
<tr>
<td><img src="image9.png" alt="Pattern 5" /></td>
<td><img src="image10.png" alt="Graph 5" /></td>
<td>6</td>
<td><img src="image11.png" alt="Pattern 6" /></td>
<td><img src="image12.png" alt="Graph 6" /></td>
<td>2</td>
</tr>
</tbody>
</table>
## 02 Related Work

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Search-Based</strong></td>
<td><strong>Sampling-Based</strong></td>
<td><strong>GNN+Counter</strong></td>
</tr>
<tr>
<td><strong>Exact Results</strong></td>
<td><strong>Approximate Results</strong></td>
<td><strong>Approximate Results</strong></td>
</tr>
<tr>
<td><strong>Excessive Computation Cost</strong></td>
<td><strong>Large Computation Cost</strong></td>
<td><strong>Little Computation Cost</strong></td>
</tr>
</tbody>
</table>

# Shortage of General GNN-based Isomorphism Counting Models

<table>
<thead>
<tr>
<th>Node Centric</th>
<th>Fixed Graph Representation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Isomorphism Counting focus on topology information</td>
<td>Distinct structures of queries</td>
</tr>
<tr>
<td>Hard to capture interaction among nodes</td>
<td>Fixed graph representation</td>
</tr>
</tbody>
</table>
Count-GNN

Edge-Centric Aggregating GNN  Query-Conditioned Graph Modulation

Experiments on Four Benchmark Datasets
8x~26x speedups over exact methods
3.1x~6.5x speedups over GNN methods
PART 02

Proposed Model: Count-GNN
Overall-Framework

(a) Toy queries/graph
(b) Edge-centric aggregation
(c) Query-conditioned graph modulation
(d) Objective

Query Readout

+$\Theta$: query conditioning
Edge-Centric Aggregation

- \( x_\bullet \): encoded nodes or edges into input features
- \(||\): concatenation operator
- \( h_{(u,v)} \): message on edge \( (u, v) \) in the \( l \)-th layer
- \( h_{(u,v)}^{l-1} \): message aggregated from the preceding edges of \( (u, v) \)
- \( \text{AGGR} \): aggregation operator
- \( W^l, U^l, b^l \): learnable parameter

Mathematical Formulas:

\[
\begin{align*}
h_{(u,v)}^0 &= x_u \| x_{(u,v)} \| x_v \in \mathbb{R}^{d_0} \\
h_{(u,v)}^l &= \sigma(W^l h_{(u,v)}^{l-1} + U^l h_{(\cdot,u)}^{l-1} + b^l) \\
h_{(u,v)}^{l-1} &= \text{AGGR}(\{h_{(i,u)}^{l-1} | (i,u) \in E\})
\end{align*}
\]
(b) Modulation conditioned on queries

- **Q**: encoded nodes or edges into input features
- **γ, β**: FiLM factors for scaling and shifting
- **⊙**: Hadamard product
- **Wγ, Uγ, Wβ, Uβ**: learnable weight matrices
- **bγ, bβ**: learnable bias vectors

\[
\begin{align*}
\mathbf{h}_Q &= \sigma(Q \cdot \text{AGGR}(\{\mathbf{h}_{(u,v)} | (u,v) \in E_Q\})) \\
\tilde{\mathbf{h}}_{(u,v)} &= (\gamma_{(u,v)} + 1) \odot \mathbf{h}_{(u,v)} + \beta_{(u,v)} \\
\gamma_{(u,v)} &= \sigma(\mathbf{W}_\gamma \mathbf{h}_{(u,v)} + \mathbf{U}_\gamma \mathbf{h}_Q + \mathbf{b}_\gamma) \\
\beta_{(u,v)} &= \sigma(\mathbf{W}_\beta \mathbf{h}_{(u,v)} + \mathbf{U}_\beta \mathbf{h}_Q + \mathbf{b}_\beta) \\
\mathbf{h}_G &= \sigma(G \cdot \text{AGGR}(\{\tilde{\mathbf{h}}_{(u,v)} | (u,v) \in E_G\}))
\end{align*}
\]
Counter Module and Overall Objective

**Counter Module**

\[
\hat{n}(Q, G) = \text{ReLU} (w^T \text{MATCH}(h_Q, h_G^Q) + b)
\]

\[
\text{MATCH}(x, y) = \text{FCL}(x \parallel y \parallel x - y \parallel x \odot y)
\]

- MATCH(·, ·): outputs the matchability between its arguments
- \(w, b\): learnable weight vector and bias
- FCL: full connected layer

**Overall Objective**

\[
\frac{1}{|\mathcal{T}|} \sum_{(Q_i, G_i, n_i) \in \mathcal{T}} |\hat{n}(Q_i, G_i) - n_i| + \lambda L_{\text{FiLM}} + \mu \| \Theta \|_2^2
\]

\[
L_{\text{FiLM}} = \sum_{(Q_i, G_i, n_i) \in \mathcal{T}} \sum_{(u,v) \in E_{G_i}} \| \gamma_{u,v} \|_2^2 + \| \beta_{u,v} \|_2^2
\]

- \(n_i\): ground truth
- \(L_{\text{FiLM}}\): regularizer on the FiLM factors
- \(\| \Theta \|_2^2\): learnable weight vector and bias
Theoretical Analysis of Count-GNN

Lemma 1 (Generalization). \( \text{Count-GNN can be reduced to a node-centric GNN, i.e., Count-GNN can be regarded as a generalization of the latter.} \)

Theorem 1 (Expressiveness). \( \text{Count-GNN is more powerful than node-centric GNNs, which means (i) for any two non-isomorphic graphs that can be distinguished by a node-centric GNN, they can also be distinguished by Count-GNN; and (ii) there exists two non-isomorphic graphs that can be distinguished by Count-GNN but not by a node-centric GNN.} \)
PART 03

Experiments
Datasets & Baselines

- **Datasets**

<table>
<thead>
<tr>
<th></th>
<th>SMALL</th>
<th>LARGE</th>
<th>MUTAG</th>
<th>OGB-PPA</th>
</tr>
</thead>
<tbody>
<tr>
<td># Queries</td>
<td>75</td>
<td>122</td>
<td>24</td>
<td>12</td>
</tr>
<tr>
<td># Graphs</td>
<td>6,790</td>
<td>3,240</td>
<td>188</td>
<td>6,000</td>
</tr>
<tr>
<td># Triples</td>
<td>448,140</td>
<td>395,280</td>
<td>4,512</td>
<td>57,940</td>
</tr>
<tr>
<td>Avg($</td>
<td>V_G</td>
<td>$)</td>
<td>5.20</td>
<td>8.43</td>
</tr>
<tr>
<td>Avg($</td>
<td>E_G</td>
<td>$)</td>
<td>6.80</td>
<td>12.23</td>
</tr>
<tr>
<td>Avg($</td>
<td>V_G'</td>
<td>$)</td>
<td>32.62</td>
<td>239.94</td>
</tr>
<tr>
<td>Avg($</td>
<td>E_G'</td>
<td>$)</td>
<td>76.34</td>
<td>559.68</td>
</tr>
<tr>
<td>Avg(Counts)</td>
<td>14.83</td>
<td>34.42</td>
<td>17.76</td>
<td>13.83</td>
</tr>
<tr>
<td>Max($</td>
<td>L</td>
<td>$)</td>
<td>16</td>
<td>64</td>
</tr>
<tr>
<td>Max($</td>
<td>L'</td>
<td>$)</td>
<td>16</td>
<td>64</td>
</tr>
</tbody>
</table>

- **Baselines**

**Conventional GNNs:**
- GCN, GAT, DPGCNN, GIN, DiffPool

**GNN-based isomorphism counting models:**
- RGCN-DN, RGCN-Sum, RGIN-DN, RGIN-Sum[2], LRP, DMPNN-LRP[3]

**Exact Approaches:**
- VF2, Peregrine[1]

[3] Liu, X.; and Song, Y. 2022. Graph convolutional networks with dual message passing for subgraph isomorphism counting and matching. AAAI.
### Empirical Results

**Observations**

- Count-GNN achieves $65x \sim 324x$ speedups over the classical VF2, $8x \sim 26x$ speedups over Peregrine.
- Count-GNN is more efficient than other GNN-based isomorphism counting models.
- Count-GNN is more accurate than conventional GNN models by at least 30% improvements in most cases.

<table>
<thead>
<tr>
<th>Methods</th>
<th>SMALL MAE</th>
<th>SMALL Q-error</th>
<th>LARGE MAE</th>
<th>LARGE Q-error</th>
<th>Time/s</th>
<th>MUTAG MAE</th>
<th>MUTAG Q-error</th>
<th>Time/s</th>
<th>OGB-PPA MAE</th>
<th>OGB-PPA Q-error</th>
<th>Time/s</th>
</tr>
</thead>
<tbody>
<tr>
<td>GCN</td>
<td>14.8 ± 0.5</td>
<td>2.1 ± 0.1</td>
<td>33.0 ± 0.4</td>
<td>3.5 ± 1.0</td>
<td>7.9 ± 0.2</td>
<td>19.9 ± 0.7</td>
<td>4.2 ± 2.5</td>
<td>29.8 ± 0.7</td>
<td>32.6 ± 0.1</td>
<td>36.8 ± 1.4</td>
<td>2.1 ± 0.4</td>
</tr>
<tr>
<td>GraphSAGE</td>
<td>14.0 ± 2.7</td>
<td>2.5 ± 0.8</td>
<td>33.8 ± 1.6</td>
<td>3.1 ± 0.4</td>
<td>7.0 ± 0.1</td>
<td>13.9 ± 2.8</td>
<td>4.7 ± 0.8</td>
<td>27.5 ± 1.3</td>
<td>32.5 ± 4.5</td>
<td>35.8 ± 2.4</td>
<td>2.2 ± 0.6</td>
</tr>
<tr>
<td>GAT</td>
<td>12.2 ± 0.7</td>
<td>2.0 ± 0.5</td>
<td>37.3 ± 5.2</td>
<td>6.0 ± 1.2</td>
<td>14.3 ± 0.3</td>
<td>30.8 ± 7.6</td>
<td>6.0 ± 0.3</td>
<td>59.4 ± 0.7</td>
<td>34.1 ± 1.2</td>
<td>38.4 ± 1.2</td>
<td>2.3 ± 0.3</td>
</tr>
<tr>
<td>DPGCnn</td>
<td>16.8 ± 2.7</td>
<td>2.9 ± 0.2</td>
<td>39.8 ± 3.7</td>
<td>5.4 ± 1.6</td>
<td>21.7 ± 0.4</td>
<td>27.5 ± 2.5</td>
<td>4.9 ± 0.6</td>
<td>64.8 ± 0.9</td>
<td>38.4 ± 1.2</td>
<td>35.9 ± 4.7</td>
<td>2.7 ± 0.3</td>
</tr>
<tr>
<td>DiffPool</td>
<td>14.8 ± 2.6</td>
<td>2.1 ± 0.4</td>
<td>34.9 ± 1.4</td>
<td>3.8 ± 0.7</td>
<td>7.0 ± 0.1</td>
<td>6.4 ± 0.3</td>
<td>2.5 ± 0.2</td>
<td>32.5 ± 0.7</td>
<td>34.6 ± 1.4</td>
<td>35.9 ± 4.7</td>
<td>2.7 ± 0.3</td>
</tr>
<tr>
<td>GIN</td>
<td>12.6 ± 0.5</td>
<td>2.1 ± 0.1</td>
<td>35.9 ± 0.6</td>
<td>4.8 ± 0.2</td>
<td>7.1 ± 0.0</td>
<td>21.3 ± 1.0</td>
<td>5.6 ± 0.7</td>
<td>33.5 ± 0.6</td>
<td>34.6 ± 1.4</td>
<td>35.9 ± 4.7</td>
<td>2.7 ± 0.3</td>
</tr>
<tr>
<td>RGCN-Sum</td>
<td>24.2 ± 6.1</td>
<td>3.7 ± 1.2</td>
<td>80.9 ± 26.3</td>
<td>6.3 ± 1.3</td>
<td>13.2 ± 0.1</td>
<td>8.0 ± 0.8</td>
<td>1.5 ± 0.1</td>
<td>61.8 ± 0.2</td>
<td>34.5 ± 13.6</td>
<td>35.8 ± 6.4</td>
<td>4.1 ± 0.1</td>
</tr>
<tr>
<td>RGCN-DN</td>
<td>16.6 ± 2.3</td>
<td>3.2 ± 1.3</td>
<td>73.7 ± 29.2</td>
<td>9.1 ± 4.2</td>
<td>48.1 ± 0.2</td>
<td>7.3 ± 0.8</td>
<td>2.6 ± 0.2</td>
<td>105.0 ± 0.4</td>
<td>57.1 ± 15.7</td>
<td>38.4 ± 6.4</td>
<td>4.1 ± 0.1</td>
</tr>
<tr>
<td>RGin-Sum</td>
<td>10.7 ± 0.3</td>
<td>2.0 ± 0.2</td>
<td>33.2 ± 2.2</td>
<td>4.2 ± 1.3</td>
<td>12.2 ± 0.0</td>
<td>10.8 ± 0.9</td>
<td>1.9 ± 0.1</td>
<td>61.4 ± 1.0</td>
<td>29.1 ± 1.7</td>
<td>35.8 ± 6.4</td>
<td>4.1 ± 0.1</td>
</tr>
<tr>
<td>RGin-DN</td>
<td>11.6 ± 0.2</td>
<td>2.4 ± 0.0</td>
<td>32.5 ± 1.9</td>
<td>4.3 ± 2.0</td>
<td>49.7 ± 1.8</td>
<td>8.6 ± 1.9</td>
<td>3.3 ± 0.8</td>
<td>104.0 ± 1.5</td>
<td>29.1 ± 1.7</td>
<td>35.8 ± 6.4</td>
<td>4.1 ± 0.1</td>
</tr>
<tr>
<td>DMPNN-LRP</td>
<td>9.1 ± 0.2</td>
<td>1.5 ± 0.1</td>
<td>28.1 ± 1.3</td>
<td>3.4 ± 1.5</td>
<td>32.4 ± 1.4</td>
<td>5.4 ± 1.8</td>
<td>1.8 ± 1.0</td>
<td>184.2 ± 1.8</td>
<td>25.6 ± 4.9</td>
<td>35.8 ± 4.1</td>
<td>4.4 ± 1.1</td>
</tr>
<tr>
<td>Count-GNN</td>
<td>8.5 ± 0.0</td>
<td>1.4 ± 0.1</td>
<td>7.9 ± 0.3</td>
<td>2.5 ± 0.5</td>
<td>30.9 ± 4.3</td>
<td>4.2 ± 0.1</td>
<td>1.8 ± 0.0</td>
<td>59.2 ± 1.7</td>
<td>25.6 ± 4.9</td>
<td>35.8 ± 4.1</td>
<td>4.4 ± 1.1</td>
</tr>
<tr>
<td>VF2</td>
<td>0 ± 1</td>
<td>0 ± 1</td>
<td>1049.2 ± 2.7</td>
<td>0 ± 1</td>
<td>9270.5 ± 5.9</td>
<td>0 ± 1</td>
<td>904.2 ± 4.5</td>
<td>130.0 ± 0.4</td>
<td>0 ± 1</td>
<td>5836.3 ± 4.8</td>
<td>1 ± 0.2</td>
</tr>
<tr>
<td>Peregrine</td>
<td>- -</td>
<td>- -</td>
<td>72.4 ± 2.0</td>
<td>- -</td>
<td>9206.5 ± 5.9</td>
<td>0 ± 1</td>
<td>904.2 ± 4.5</td>
<td>0.2 ± 0.3</td>
<td>- -</td>
<td>5836.3 ± 4.8</td>
<td>1 ± 0.2</td>
</tr>
</tbody>
</table>
Ablation Study & Parameter Sensitivity

Table 5: Ablation study on Count-GNN.

<table>
<thead>
<tr>
<th>Methods</th>
<th>SMALL</th>
<th>LARGE</th>
<th>MUTAG</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MAE</td>
<td>Q-error</td>
<td>MAE</td>
</tr>
<tr>
<td>Count-GNN\E</td>
<td>11.3</td>
<td>2.07</td>
<td>33.58</td>
</tr>
<tr>
<td>Count-GNN\M</td>
<td>8.66</td>
<td>1.46</td>
<td>29.65</td>
</tr>
<tr>
<td>Count-GNN</td>
<td><strong>8.54</strong></td>
<td><strong>1.41</strong></td>
<td><strong>30.91</strong></td>
</tr>
</tbody>
</table>

- **Ablation study**
  - **Count-GNN\E**: replaces the edge-centric aggregation with the node-centric GIN
  - **Count-GNN\N**: replaces the query-conditioned modulation with a simple sumpooling as the readout for the input graph
  - Count-GNN\E has the lowest accuracy
  - Count-GNN\M performs better than Count-GNN\E
  - Full model Count-GNN achieves the best performance

- **Parameters sensitivity**
  - As K increases, the performance in terms of MAE and Q-error generally become better, only with one exception on Q-error when K = 4
  - λ = 0.01 may result in an inferior performance. Interval [1e-5, 1e-3] might be a good range for superior performance of subgraph isomorphism counting.
PART 04

Conclusions
Conclusions

• Problem
  • Subgraph isomorphic counting

• Proposed-Model: Count-GNN
  • Edge-centric message passing
  • Query-conditioned graph modulation

• Experiment
  • Count-GNN achieves significant speedups over exact methods
  • Count-GNN is more efficient than other GNN-based isomorphism counting models
  • Count-GNN is more accurate than conventional GNN models by at least 30% improvements in most cases.
Thanks!

Paper, data & code available at [https://xingtongyu.netlify.app/](https://xingtongyu.netlify.app/)

Xingtong Yu*, Zemin Liu*, Yuan Fang†, Xinming Zhang†

**Learning to Count Isomorphisms with Graph Neural Networks**

*In Proceeding of 37th AAAI Conference on Artificial Intelligence, February 7-14, 2023*