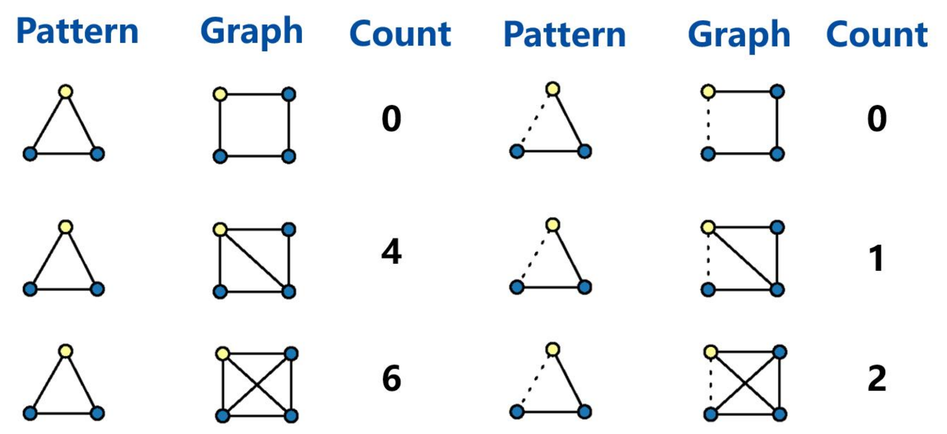
Learning to Count Isomorphisms with Graph Neural Network

Xingtong Yu, Zemin Liu, Yuan Fang, Xinming Zhang

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

Motivation

Problem: Subgraph Isomorphism Counting



What's missing in SOTA?

- Node-centric scheme falls short of matching complex structures for isomorphism counting.
- SOTA[1, 2, 3] models leverage a fixed graph representation to match with all possible queries.

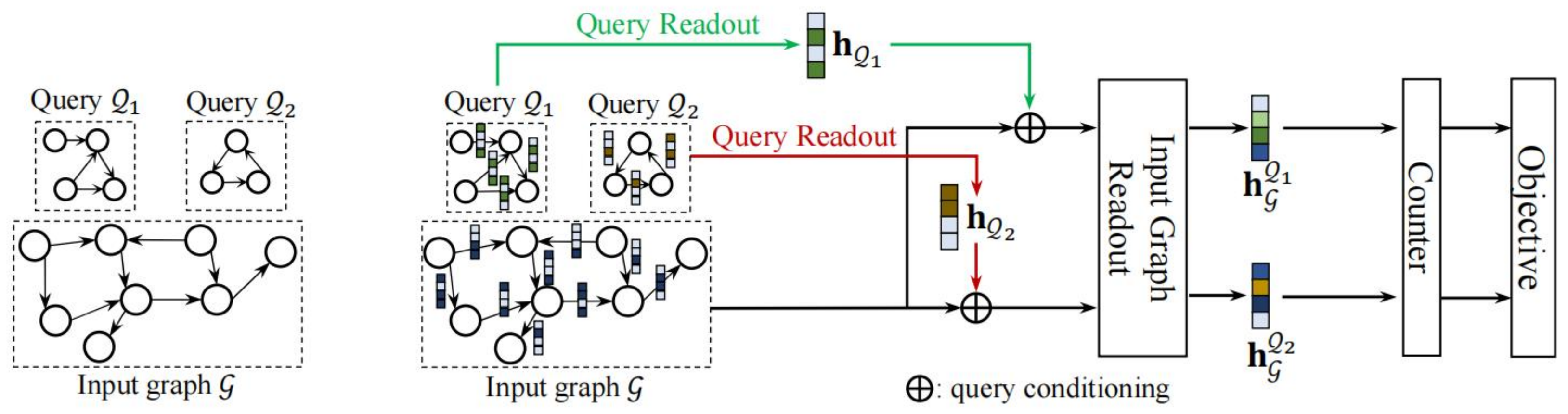
Challenges:

C1: How to capture fine grained structural information?

C2: How to adapt the input graph to each query individually?

The proposed model: Count-GNN

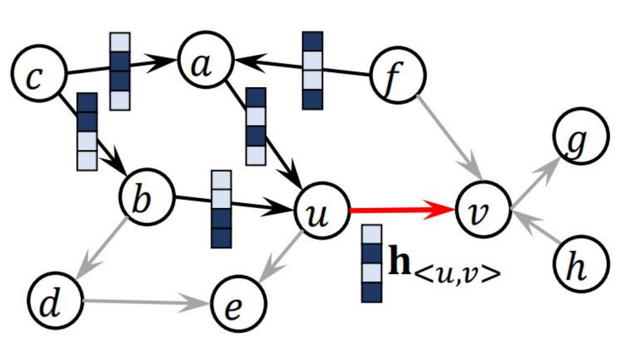
Overall-framework



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

Edge-centric aggregation

- **First challenge**
  - Exploit edge-centric message passing, in which each edge receives and aggregates from adjacent edges.



(a) Edge-centric aggregation

- Edge message initialization

$$\mathbf{h}^0_{\langle u,v\rangle} = \mathbf{x}_u \parallel \mathbf{x}_{\langle u,v\rangle} \parallel \mathbf{x}_v \in \mathbb{R}^{d_0}$$

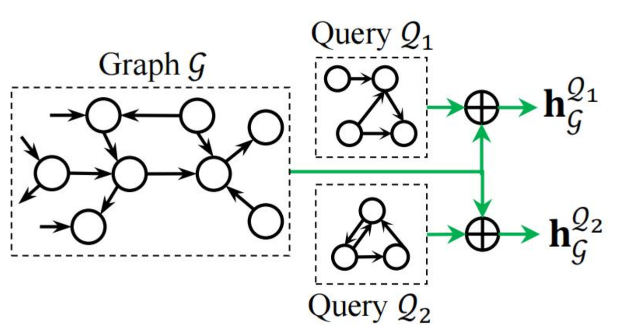
- Edge message passing

$$\mathbf{h}^l_{\langle u,v\rangle} = \sigma(\mathbf{W}^l \mathbf{h}^{l-1}_{\langle u,v\rangle} + \mathbf{U}^l \mathbf{h}^{l-1}_{\langle \cdot,u\rangle} + \mathbf{b}^l).$$

$$\mathbf{h}^{l-1}_{\langle \cdot,u\rangle} = \textsc{Aggr}(\{\mathbf{h}^{l-1}_{\langle i,u\rangle} | \langle i,u\rangle \in E\})$$

Query graph representation

- **Second challenge**
  - Modulate the input graph conditioned on the query to adapt the whole graph representation of the input graph to each query.



(b) Modulation conditioned on queries

- Query Readout

$$\mathbf{h}_{\mathcal{Q}} = \sigma(\mathbf{Q} \cdot \textsc{Aggr}(\{\mathbf{h}_{\langle u,v\rangle} | \langle u,v\rangle \in E_{\mathcal{Q}}\}))$$

- Graph Edge Representation

$$\tilde{\mathbf{h}}_{\langle u,v\rangle} = (\gamma_{\langle u,v\rangle} + \mathbf{1}) \odot \mathbf{h}_{\langle u,v\rangle} + \beta_{\langle u,v\rangle}$$

$$\gamma_{\langle u,v\rangle} = \sigma(\mathbf{W}_\gamma \mathbf{h}_{\langle u,v\rangle} + \mathbf{U}_\gamma \mathbf{h}_{\mathcal{Q}} + \mathbf{b}_\gamma)$$

$$\beta_{\langle u,v\rangle} = \sigma(\mathbf{W}_\beta \mathbf{h}_{\langle u,v\rangle} + \mathbf{U}_\beta \mathbf{h}_{\mathcal{Q}} + \mathbf{b}_\beta)$$

- Graph Readout

$$\mathbf{h}^{\mathcal{Q}}_{\mathcal{G}} = \sigma(\mathbf{G} \cdot \textsc{Aggr}(\{\tilde{\mathbf{h}}_{\langle u,v\rangle} | \langle u,v\rangle \in E_{\mathcal{G}}\}))$$

Counter module

$$\hat{n}(\mathcal{Q},\mathcal{G}) = \mathrm{ReLU}(\mathbf{w}^\top \textsc{Match}(\mathbf{h}_{\mathcal{Q}}, \mathbf{h}^{\mathcal{Q}}_{\mathcal{G}}) + b) \qquad \textsc{Match}(\mathbf{x},\mathbf{y}) = \mathrm{FCL}(\mathbf{x} \parallel \mathbf{y} \parallel \mathbf{x}-\mathbf{y} \parallel \mathbf{x} \odot \mathbf{y})$$

Isomorphism counting function

missing neighborhood information

Full connected layer

Overall objective

FiLM regularizer

L2 regularizer

$$\frac{1}{|\mathcal{T}|} \sum_{(\mathcal{Q}_i,\mathcal{G}_i,n_i)\in\mathcal{T}} |\hat{n}(\mathcal{Q}_i,\mathcal{G}_i) - n_i| + \lambda \mathcal{L}_{\mathrm{FiLM}} + \mu \|\Theta\|_2^2$$

$$\mathcal{L}_{\mathrm{FiLM}} = \sum_{(\mathcal{Q}_i,\mathcal{G}_i,n_i)\in\mathcal{T}} \sum_{\langle u,v\rangle \in E_{\mathcal{G}_i}} \|\gamma_{\langle u,v\rangle}\|_2^2 + \|\beta_{\langle u,v\rangle}\|_2^2$$

Experiments

Experimental setup

	SMALL	LARGE	MUTAG	OGB-PPA
# Queries	75	122	24	12
# Graphs	6,790	3,240	188	6,000
# Triples	448,140	395,280	4,512	57,940
Avg($\lvert V_{\mathcal{Q}}\rvert$)	5.20	8.43	3.50	4.50
Avg($\lvert E_{\mathcal{Q}}\rvert$)	6.80	12.23	2.50	4.75
Avg($\lvert V_{\mathcal{G}}\rvert$)	32.62	239.94	17.93	152.75
Avg($\lvert E_{\mathcal{G}}\rvert$)	76.34	559.68	39.58	1968.29
Avg(Counts)	14.83	34.42	17.76	13.83
Max($\lvert L\rvert$)	16	64	7	8
Max($\lvert L'\rvert$)	16	64	4	1

- **Conventional GNNs**
  - GCN[4]
  - GAT[5]
  - GraphSage[6]
  - GIN[7]
  - DiffPool[8]
- **GNN based Isomorphism Counting Models**
  - RGCN-DN,RGCN-SUM,RGIN-DN,RGIN-SUM[1]
  - LRP[2]
  - DMPNN-LRP[3]
- **Exact Methods**
  - VF2[9]
  - Peregrine[10]

Isomorphisms counting

Methods	SMALL MAE ↓	SMALL Q-error ↓	SMALL Time/s ↓	LARGE MAE ↓	LARGE Q-error ↓	LARGE Time/s ↓	MUTAG MAE ↓	MUTAG Q-error ↓	MUTAG Time/s ↓	OGB-PPA MAE ↓	OGB-PPA Q-error ↓	OGB-PPA Time/s ↓
GCN	14.8 ± 0.5	2.1 ± 0.1	7.9 ± 0.2	33.0 ± 0.4	3.5 ± 1.0	29.8 ± 0.7	19.9 ± 9.7	4.2 ± 1.5	0.88 ± 0.02	36.8 ± 1.4	2.1 ± 0.4	12.5 ± 0.3
GraphSAGE	14.0 ± 2.7	2.5 ± 0.8	**7.0** ± 0.1	33.8 ± 1.6	3.1 ± 0.4	**27.5** ± 1.3	13.9 ± 2.8	4.7 ± 0.8	0.88 ± 0.02	32.5 ± 4.5	2.5 ± 0.5	**11.1** ± 0.1
GAT	12.2 ± 0.7	2.0 ± 0.5	14.3 ± 0.3	37.3 ± 5.2	6.0 ± 1.2	59.4 ± 0.7	30.8 ± 6.7	6.0 ± 0.3	0.91 ± 0.01	35.8 ± 2.4	2.2 ± 0.6	30.4 ± 0.8
DPGCNN	16.8 ± 0.7	2.9 ± 0.2	21.7 ± 0.4	39.8 ± 3.7	5.4 ± 1.6	64.8 ± 0.9	27.5 ± 2.5	4.9 ± 0.6	1.54 ± 0.01	38.4 ± 1.2	2.3 ± 0.3	19.4 ± 0.7
DiffPool	14.8 ± 2.6	2.1 ± 0.4	**7.0** ± 0.1	34.9 ± 1.4	3.8 ± 0.7	32.5 ± 0.7	6.4 ± 0.3	2.5 ± 0.2	0.86 ± 0.00	35.9 ± 4.7	2.7 ± 0.3	15.4 ± 2.2
GIN	12.6 ± 0.5	2.1 ± 0.1	7.1 ± 0.0	35.9 ± 0.6	4.8 ± 0.2	33.5 ± 0.6	21.3 ± 1.0	5.6 ± 0.7	0.41 ± 0.01	34.6 ± 1.4	2.5 ± 0.5	12.3 ± 0.4
RGCN-Sum	24.2 ± 6.1	3.7 ± 1.2	13.2 ± 0.1	80.9 ± 26.3	6.3 ± 1.3	61.8 ± 0.2	8.0 ± 0.8	**1.5** ± 0.1	0.89 ± 0.01	34.5 ± 13.6	4.7 ± 0.8	33.0 ± 0.2
RGCN-DN	16.6 ± 2.3	3.2 ± 1.3	48.1 ± 0.2	73.7 ± 29.2	9.1 ± 4.2	105.0 ± 0.4	7.3 ± 0.8	2.6 ± 0.2	1.19 ± 0.04	57.1 ± 15.7	5.0 ± 1.3	31.2 ± 0.1
RGIN-Sum	10.7 ± 0.3	2.0 ± 0.2	12.2 ± 0.0	33.2 ± 2.2	4.2 ± 1.3	61.4 ± 1.0	10.8 ± 0.9	1.9 ± 0.1	0.45 ± 0.02	29.1 ± 1.7	1.2 ± 0.6	21.0 ± 1.2
RGIN-DN	11.6 ± 0.2	2.4 ± 0.0	49.7 ± 1.8	32.5 ± 1.9	4.3 ± 2.0	104.0 ± 1.5	8.6 ± 1.9	3.3 ± 0.8	0.73 ± 0.03	35.8 ± 6.4	4.4 ± 1.1	28.8 ± 0.3
DMPNN-LRP	9.1 ± 0.2	1.5 ± 0.1	32.4 ± 1.4	**28.1** ± 1.3	3.4 ± 1.5	184.2 ± 1.8	5.4 ± 1.8	1.8 ± 1.0	0.13 ± 0.05	**25.6** ± 4.9	1.1 ± 1.3	63.0 ± 0.6
Count-GNN	**8.5** ± 0.0	**1.4** ± 0.1	7.9 ± 0.3	30.9 ± 4.3	**2.5** ± 0.5	59.2 ± 1.7	**4.2** ± 0.1	1.8 ± 0.0	**0.02** ± 0.00	28.7 ± 3.9	**1.0** ± 0.2	18.1 ± 0.6
VF2	0	1	1049.2 ± 2.7	0	1	9270.5 ± 5.9	0	1	1.30 ± 0.04	0	1	5836.3 ± 4.8
Peregrine	-	-	72.4 ± 2.0	-	-	904.2 ± 4.5	-	-	0.2 ± 0.03	-	-	450.1 ± 3.9

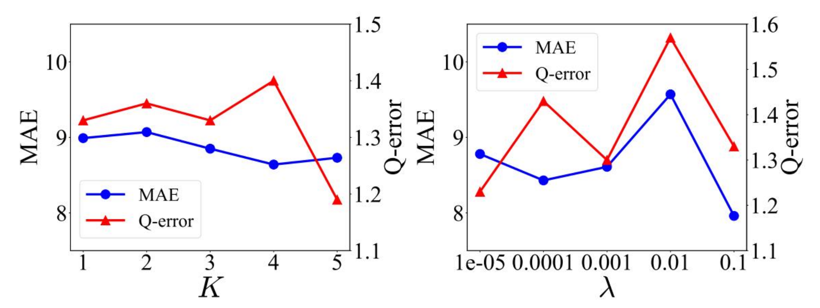
- **Count-GNN achieves 65x ~ 324x speedups over the classical VF2, 8x ~ 26x speedups over Peregrine**
- **Count-GNN is more efficient than other GNN-based isomorphism counting models**
- **Count-GNN is more accurate than Conventional GNN models by at least 30% improvements in most cases.**

Ablation study

Methods	SMALL MAE	SMALL Q-error	LARGE MAE	LARGE Q-error	MUTAG MAE	MUTAG Q-error
Count-GNN\E	11.3	2.07	33.58	4.96	18.63	5.92
Count-GNN\M	8.66	1.46	**29.65**	3.34	4.41	1.82
Count-GNN	**8.54**	**1.41**	30.91	**2.46**	**4.22**	**1.76**

- **Ablation study**
  - Node-centric aggregation: impairs the performance
  - Query-conditioned graph modulation : contributes to the performance

Parameter sensitivity



(a) Sensitivity of K ↓ (b) Sensitivity of λ ↓

- **Parameter 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.

Conclusions

Problem

- Subgraph Isomorphism Counting

Proposed model: Count-GNN

- Edge-centric message passing
  - to capture fine grained structural information
- **Query-conditioned graph modulation**
  - to adapt each graph to query individually

Experiments

- Extensive experiments demonstrate that Count-GNN significantly outperforms state-of-the-art models

Reference

[1] Liu, X. et al. 2020. Neural subgraph isomorphism counting. KDD.
[2] Zhengdao, C. et al. 2020. Can Graph Neural Networks Count Substructures? NeurIPS.
[3] Liu, X.; and Song, Y. 2022. Graph convolutional networks with dual message passing for subgraph isomorphism counting and matching. AAAI.
[4] Kipf, T. N.; and Welling, M. 2017. Semi-supervised classification with graph convolutional networks. ICLR.
[5] Veliˇckovi´c, P. et al. 2018. Graph attention networks. ICLR.
[6] Hamilton, W. et al. 2017. Inductive representation learning on large graphs. NIPS.
[7] Xu, K. et al. 2019. How powerful are graph neural networks? ICLR.
[8] Ying, Z.; You, J. et al. 2018. Hierarchical Graph Representation Learning with Differentiable Pooling. NeurIPS.
[9] Cordella, L. P. et al. 2004. A (sub) graph isomorphism algorithm for matching large graphs. PAMI.
[10] Jamshidi, K. et al. 2020. Peregrine: a pattern-aware graph mining system. EuroSys.

Acknowledgments

This work was supported in part by the National Key Research and Development Program of China under Grant 2020YFB2103803. Dr. Yuan Fang acknowledges the Lee Kong Chian Fellowship awarded by Singapore Management University for the support of this work. The authors wish to thank Dr. Yuchen Li for his valuable comments on this work.