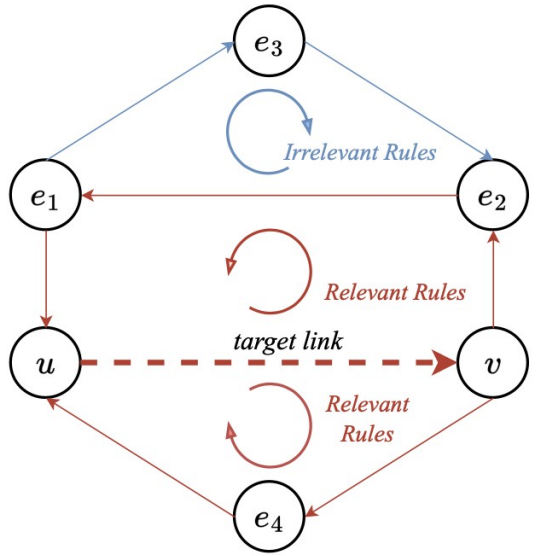
Learning Rule-Induced Subgraph Representations for Inductive Relation Prediction

Tianyu Liu[1] **Qitan Lv**[1] **Jie Wang**[1,2*] **Shuling Yang**[1] **Hanzhu Chen**[1]
[1]University of Science and Technology of China
[2]Institute of Artificial Intelligence, Hefei Comprehensive National Science Center
{tianyu_liu, qitanlv, slyang0916, chenhz}@mail.ustc.edu.cn
{jiewangx}@ustc.edu.cn

Introduction –

- Inductive Relation Prediction, test set != train set,
- Subgraph-based methods,
- **R**ule-induc**E**d **S**ubgraph repre-sen**T**ations (**REST**),
- Faster inference.



Related work –

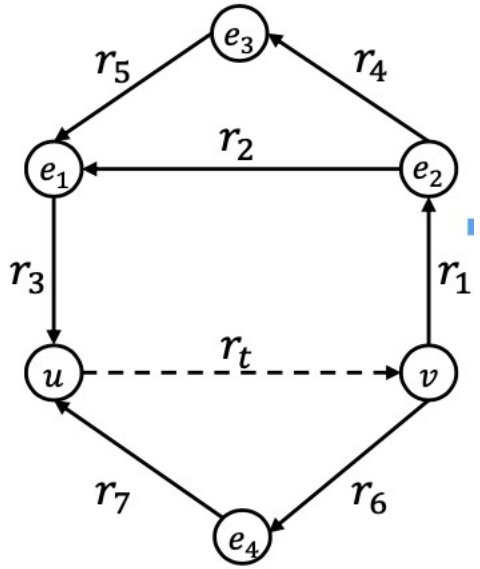
- Rule based methods, involve ILP; not scalable.
- Subgraph-based methods, basically sample neighborhoods then GNN.
- GNNs for reasoning on entire KGs (not considered by the paper?).

Problem definition –

- Inductive Relation Prediction BUT all relations in the new graph must be seen during training, basically, Transductive w.r.t. relations but inductive w.r.t entities.
- Can predict **head**, **relation** and **tail** with this restriction.

REST –

- Subgraph Extraction.



REST –

- Single source initialization.

$$\mathbf{e}^0_{x,y,z} = \mathop{\mathbb{1}}_{(u,r,v)}(x,y,z) \odot \mathbf{r}_y = \begin{cases} \mathbf{r}_y, & \text{if } (x,y,z) = (u,r_t,v) \\ \mathbf{0}, & \text{if } (x,y,z) \neq (u,r_t,v) \end{cases}$$

$$\mathbf{h}^0_v = \mathbf{0} \qquad for\ \forall v \in \mathcal{E},$$

- Edge-wise message passing.

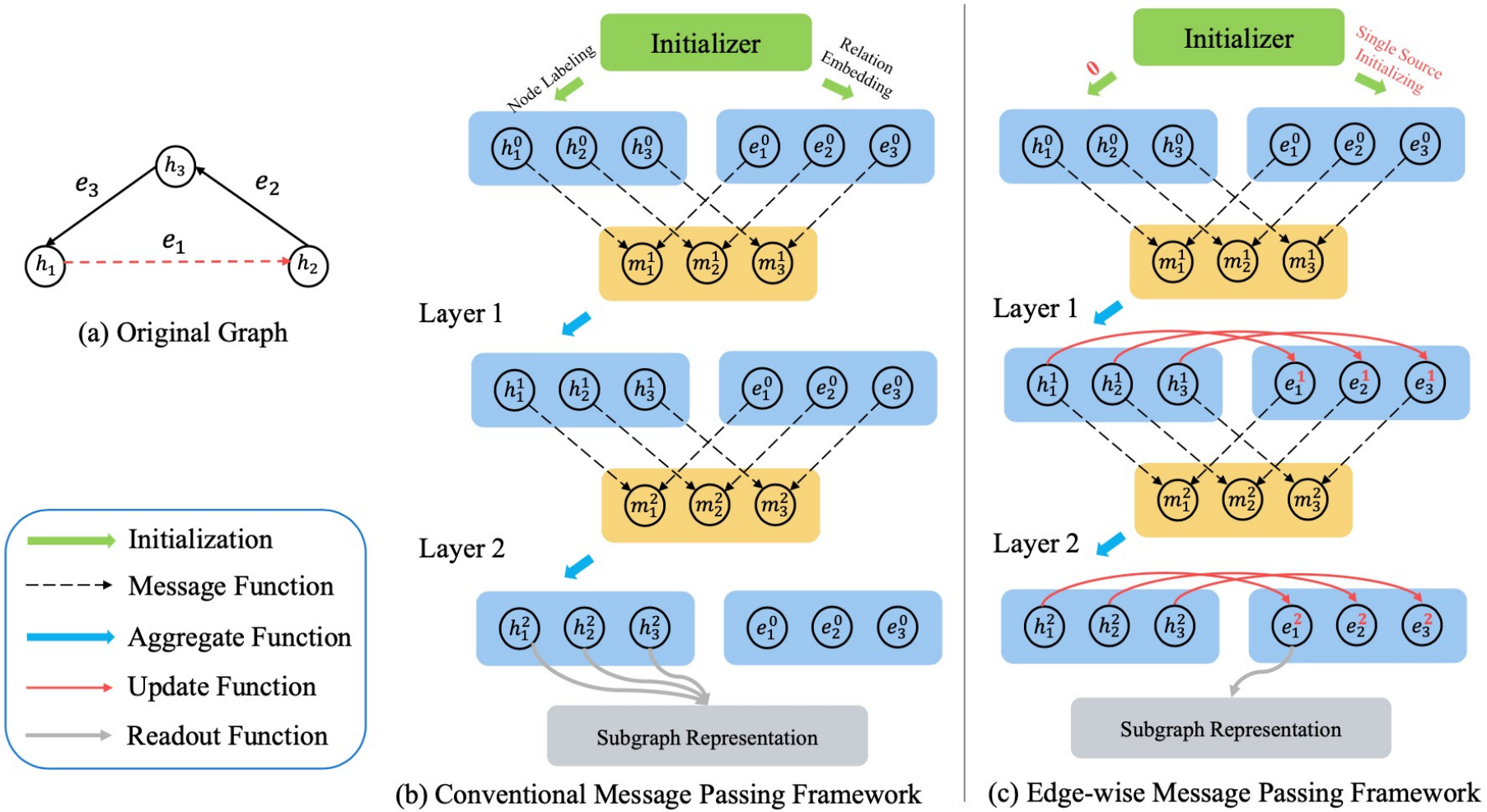
$$\mathbf{m}^k_{x,y,z} = \textbf{MESSAGE}(\mathbf{h}^{k-1}_x, \mathbf{e}^{k-1}_{x,y,z}, \mathbf{r}_y) = (\mathbf{h}^{k-1}_x \otimes^1 \mathbf{r}_y) \uplus (\mathbf{e}^{k-1}_{x,y,z} \otimes^2 \mathbf{r}_y)$$

$$\mathbf{h}^k_z = \mathop{\textbf{AGGRAGATE}}_{(x,y,z)\in\mathcal{T}}(\mathbf{m}^k_{x,y,z}) = \bigoplus_{(x,y,z)\in\mathcal{T}} \mathbf{m}^k_{x,y,z}$$

$$\mathbf{e}^k_{x,y,z} = \textbf{UPDATE}(\mathbf{h}^k_x, \mathbf{e}^{k-1}_{x,y,z}) = \mathbf{h}^k_x \diamond \mathbf{e}^{k-1}_{x,y,z}$$

REST –

- Edge-wise message passing, e.g.



REST –

- RNN Based functions,
  - For **Message**,

$$\delta_k = \sigma_g(\mathbf{W}^k_{\delta,1}\mathbf{r}_y \odot \mathbf{e}^{k-1}_{x,y,z} + \mathbf{W}^k_{\delta,2}\mathbf{h}^{k-1}_x + \mathbf{b}^k_\delta)$$
$$\gamma_k = \sigma_g(\mathbf{W}^k_{\gamma,1}\mathbf{r}_y \odot \mathbf{e}^{k-1}_{x,y,z} + \mathbf{W}^k_{\gamma,2}\mathbf{h}^{k-1}_x + \mathbf{b}^k_\gamma)$$
$$c_k = \sigma_h(\mathbf{W}^k_{c,1}\mathbf{r}_y \odot \mathbf{e}^{k-1}_{x,y,z} + \mathbf{W}^k_{c,2}(\gamma_k \odot \mathbf{h}^{k-1}_x))$$
$$\mathbf{m}^k_{x,y,z} = \delta_k \odot c_k + (1-\delta_k) \odot \mathbf{h}^{k-1}_x$$

  - For **Aggregate**,

$$\mathbf{h}^k_{z,1} = \underset{(x,y,z)\in\mathcal{T}}{mean}(\mathbf{m}^k_{x,y,z}),\ \mathbf{h}^k_{z,2} = \underset{(x,y,z)\in\mathcal{T}}{max}(\mathbf{m}^k_{x,y,z}),$$
$$\mathbf{h}^k_{z,3} = \underset{(x,y,z)\in\mathcal{T}}{min}(\mathbf{m}^k_{x,y,z}),\ \mathbf{h}^k_{z,4} = \underset{(x,y,z)\in\mathcal{T}}{std}(\mathbf{m}^k_{x,y,z}),$$
$$\mathbf{h}^k_z = \mathbf{W}^k_{agg}[\mathbf{h}^k_{z,1}; \mathbf{h}^k_{z,2}; \mathbf{h}^k_{z,3}; \mathbf{h}^k_{z,4}; \mathbf{h}^{k-1}_z]$$

REST –

- RNN Based functions,
  - For **Update**,

$$\mathbf{q}^0_{x,y,z} = \mathbf{r}^q_r$$

$$\mathbf{e}^k_{x,y,z}, \mathbf{q}^k_{x,y,z} = \mathbf{LSTM}(\mathbf{e}^{k-1}_{x,y,z}, \mathbf{q}^{k-1}_{x,y,z}, \mathbf{h}^k_x)$$

  - Final score,

$$f(u, r_t, v) = \sigma(\mathbf{W}_s \mathbf{e}^k_{u,r_t,v} + \mathbf{b}_s)$$

Analysis –

- claim : REST can learn rule induced subgraph representations.

Experiments –

- Main results on 3 inductive datasets.

		WN18RR				FB15k-237				NELL-995			
		v1	v2	v3	v4	v1	v2	v3	v4	v1	v2	v3	v4
Rule-Based	Neural LP	74.37	68.93	46.18	67.13	52.92	58.94	52.90	55.88	40.78	78.73	82.71	80.58
	DRUM	74.37	68.93	46.18	67.13	52.92	58.73	52.90	55.88	19.42	78.55	82.71	80.58
	RuleN	80.85	78.23	53.39	71.59	49.76	77.82	87.69	85.60	53.50	81.75	77.26	61.35
Subgraph-Based	GraIL	82.45	78.68	58.43	73.41	64.15	81.80	82.83	89.29	59.50	93.25	91.41	73.19
	CoMPILE	83.60	79.82	60.69	75.49	67.64	82.98	84.67	87.44	58.38	93.87	92.77	75.19
	TACT	84.04	81.63	67.97	76.56	65.76	83.56	85.20	88.69	79.80	88.91	94.02	73.78
	SNRI	87.23	83.10	67.31	83.32	71.79	86.50	89.59	89.39	-	-	-	-
	ConGLR	85.64	92.93	70.74	92.90	68.29	85.98	88.61	89.31	81.07	94.92	94.36	81.61
	REST(**ours**)	**96.28**	**94.56**	**79.50**	**94.19**	**75.12**	**91.21**	**93.06**	**96.06**	**88.00**	**94.96**	**96.79**	**92.61**

Ablation Study –

- Importance of single source init. and edge-wise message passing.x

	WN18RR				FB15k-237				NELL-995			
	v1	v2	v3	v4	v1	v2	v3	v4	v1	v2	v3	v4
REST	96.28	94.56	79.50	94.19	75.12	91.21	93.06	96.06	88.00	94.96	96.79	92.61
Full Initialization	92.55	90.70	68.76	79.49	71.71	79.29	89.25	91.22	83.00	86.13	94.54	68.26
Δ	-3.73	-3.86	-10.74	-14.70	-3.41	-11.92	-3.81	-4.84	-5.00	-8.83	-2.25	-24.35
SUM	93.08	85.03	69.59	91.39	64.88	84.30	89.48	89.96	81.00	91.39	96.17	64.57
Δ	-3.20	-9.53	-9.91	-2.80	-10.24	-6.91	-3.58	-6.10	-7.00	-3.57	-0.62	-28.04
MUL	85.64	93.19	56.03	81.04	63.90	78.24	85.20	90.66	69.00	79.20	93.70	36.11
Δ	-10.64	-1.37	-23.47	-13.15	-11.22	-12.97	-7.86	-5.40	-19.00	-15.76	-3.09	-56.50
MLP	95.74	93.65	78.84	90.69	71.07	90.25	92.60	94.94	83.00	94.12	96.41	91.38
Δ	-0.54	-0.91	-0.66	-3.50	-4.05	-0.96	-0.46	-1.12	-5.00	-0.84	-0.38	-1.23

Extraction Efficiency –

		WN18RR				FB15k-237				NELL-995			
		v1	v2	v3	v4	v1	v2	v3	v4	v1	v2	v3	v4
Enclosing Subgraph	GraIL	121.77	537.42	1127.14	194.98	949.48	2933.04	8423.59	15089.74	136.55	1197.24	6112.77	1303.97
	REST	54.01	251.97	617.16	91.76	111.34	338.24	868.79	1,626.77	61.19	213.45	688.14	219.33
	Efficiency	**2.25**×	**2.13**×	**1.83**×	**2.12**×	**8.53**×	**8.67**×	**9.70**×	**9.28**×	**2.23**×	**5.61**×	**8.88**×	**5.95**×
Unclosing Subgraph	GraIL	127.69	517.94	1194.18	199.00	1287.35	4166.63	11499.32	21738.29	167.06	1611.97	8044.53	1542.82
	REST	56.27	260.20	631.71	95.23	123.36	386.55	985.81	1890.54	64.72	245.41	858.23	248.00
	Efficiency	**2.27**×	**1.99**×	**1.89**×	**2.09**×	**10.44**×	**10.78**×	**11.66**×	**11.50**×	**2.58**×	**6.57**×	**9.37**×	**6.22**×

PRODIGY: Enabling In-context Learning Over Graphs

Qian Huang[1*]
qhwang@cs.stanford.edu

Hongyu Ren[1*]
hyren@cs.stanford.edu

Peng Chen[1]
pengc@stanford.edu

Gregor Kržmanc[2]
gregor.krzmanc@ijs.si

Daniel Zeng[1]
dzeng@cs.stanford.edu

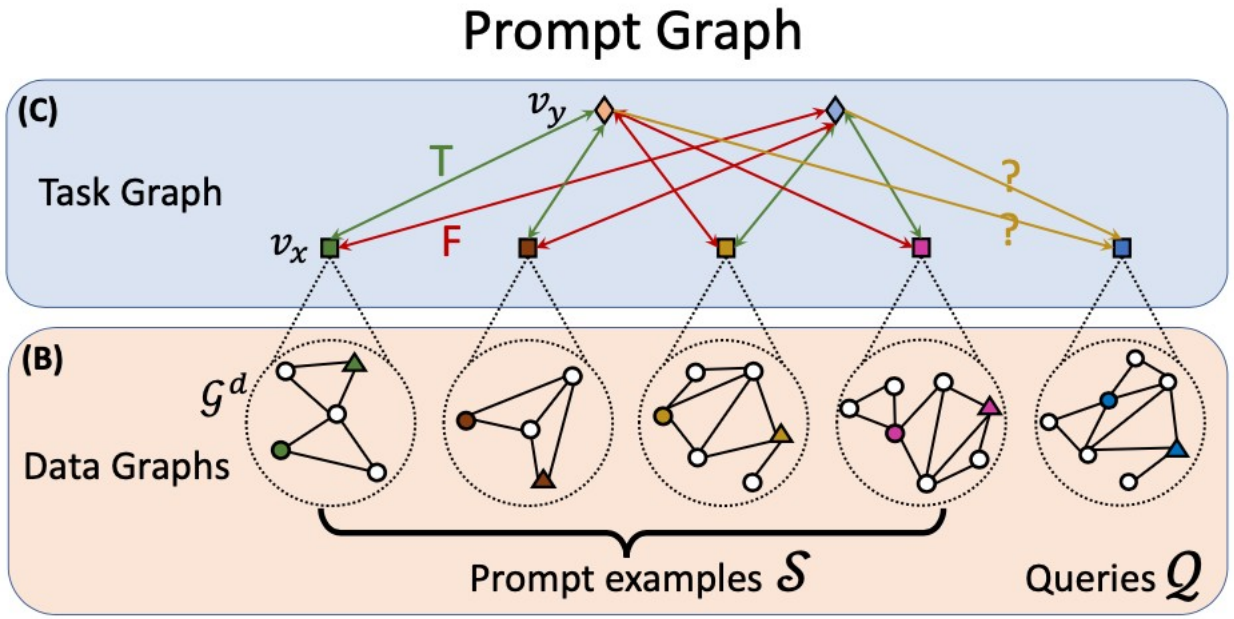
Percy Liang[1]
pliang@cs.stanford.edu

Jure Leskovec[1]
jure@cs.stanford.edu

[1]Stanford University
[2] University of Ljubljana

Introduction –

- In-context Learning,
- Challenges,
- prompt-graph,
- PRODIGY, an architecture + new pre-training tasks,
- SOTA performance.



In-context learning over Graphs –

- **Classification tasks**, node-level, edge-level, subgraph-level and graph-level!
  Generalized i/p – $x_i = (\mathcal{V}_i, \mathcal{E}_i, \mathcal{R}_i)$
  $\mathcal{V}_i$s and $\mathcal{E}_i$s can be adjusted.

- **Few-shot prompting**, for m-way classification, they use a set of m·k prompt examples + n queries *with source graph*.

$$\mathcal{S} = \{(x_i, y_i)\}_{i=1}^{m \cdot k} \qquad \mathcal{Q} = \{x_i\}_{i=1}^{n}$$

- **Prompt graph representation**, *Data Graph* + *Task Graph*. m·k + n, data nodes and m label nodes.

$$\mathcal{G}_i^{\mathtt{D}} = (\mathcal{V}_i^{\mathtt{D}}, \mathcal{E}_i^{\mathtt{D}}, \mathcal{R}_i^{\mathtt{D}}) \sim \bigoplus_{i=0}^{k} \mathtt{Neighbor}(\mathcal{V}_i, \mathcal{G}, i)$$

Pre-training to enable in-context learning –

- **For the data graph**, (M_{D} can be any GNN)

$$E \in \mathcal{R}^{|\mathcal{V}^{\mathrm{D}}| \times d} = M_{\mathrm{D}}(\mathcal{G}^{\mathrm{D}})$$

for node classification, pick the node's embedding, $G_i = E_{\mathcal{V}_i}$

for edge classification, $G_i = W^T(E_{v_1 \in \mathcal{V}_i} || E_{v_2 \in \mathcal{V}_i} || \max(E_i)) + b$

- **For the task graph**, M_{T} is -

$$H = M_{\mathrm{T}}(\mathcal{G}^{\mathrm{T}})$$

$$\beta_{ij} = MLP(W_q^T H_i^l || W_k^T H_j^l || e_{ij})$$

$$\alpha_{ij} = \frac{\exp(\beta_{ij})}{\sum_{k \in \mathcal{N}(i) \cup \{i\}} \exp(\beta_{ik})}$$

$$H_i^{l+1} = ReLU\left(BN\left(H_i^l + W_o^T \sum_{j \in \mathcal{N}(i) \cup \{i\}} \alpha_{ij} W_v^T H_j^l\right)\right)$$

- **Prediction read-out**,

$$O_i = [\texttt{cosine\_similarity}(H_{x_i}, H_y), \forall y \in \mathcal{Y}]$$

Pre-training to enable in-context learning –

- **In-context pre-training**, formulate pre-training tasks to utilize in-context information so that inference can be done in the same way.
  - **Generation 1, Neighbor matching**. (this is for when downstream is node-level, can be extend to edges)

$$\mathcal{C} = \{c_i\}_{i=1}^{m} \quad c_i \sim Uniform(\mathcal{V}_{\texttt{pretrain}})$$
$$N_i = \texttt{Neighbor}(c_i, \mathcal{G}_{\texttt{pretrain}}, l)$$
$$\mathcal{S}_i = \{(x_j, y_j = c_i)\}_{j=1}^{k} \quad x_j \sim Uniform(N_i)$$
$$\mathcal{Q}_i = \{(x_j, y_j = c_i)\}_{j=1}^{\lceil \frac{n}{m} \rceil} \quad x_j \sim Uniform(N_i)$$
$$(\mathcal{G}_{\texttt{pretrain}}, \mathcal{S}_{\text{NM}}, \mathcal{Q}_{\text{NM}}) \sim \texttt{NM}_{k,m}(\mathcal{G}_{\texttt{pretrain}})$$

  - **Generation 2, Multi-task**. (when we have both node and edge level signals, *must know f*)

$$\mathcal{C} = \{c_i\}_{i=1}^{m} \quad c_i \sim Uniform(\mathcal{Y})$$
$$\mathcal{S}_i = \{(x_j, y_j = c_i)\}_{j=1}^{k} \quad x_j \sim Uniform(\{x_i | f(x_i) = c_i\})$$
$$\mathcal{Q}_i = \{(x_j, y_j = c_i)\}_{j=1}^{\lceil \frac{n}{m} \rceil} \quad x_j \sim Uniform(\{x_i | f(x_i) = c_i\})$$

Pre-training to enable in-context learning –

- **In-context pre-training**, formulate pre-training tasks to utilize in-context information so that inference can be done in the same way.
  - **Prompt Graph, with augmentation**.
    Basically, *Drop* random nodes + *Mask* random nodes for *each* data graph,
    Then create the Task graph from all Data nodes.
  - **Pre-training loss**.

$$(\mathcal{G}_{\texttt{pretrain}}, \mathcal{S}_{\text{NM}}, \mathcal{Q}_{\text{NM}}) \sim \texttt{NM}_{k,m}(\mathcal{G}_{\texttt{pretrain}})$$
$$(\mathcal{G}_{\texttt{pretrain}}, \mathcal{S}_{\text{MT}}, \mathcal{Q}_{\text{MT}}) \sim \texttt{MT}_{k,m}(\mathcal{G}_{\texttt{pretrain}}, f)$$
$$\mathcal{L} = \mathop{\mathbb{E}}_{x_i \in \mathcal{Q}_{\text{NM}}} \texttt{CE}(O_{\text{NM},i}, y_{\text{NM},i}) + \mathop{\mathbb{E}}_{x_i \in \mathcal{Q}_{\text{MT}}} \texttt{CE}(O_{\text{MT},i}, y_{\text{MT},i})$$

Experiments –

- arXiv paper category classification,

Classes	NoPretrain	Contrastive	PG-NM	PG-MT	PRODIGY	Finetune
3	33.16 ± 0.30	65.08 ± 0.34	72.50 ± 0.35	65.64 ± 0.33	**73.09 ± 0.36**	65.42 ± 5.53
5	18.33 ± 0.21	51.63 ± 0.29	61.21 ± 0.28	51.97 ± 0.27	**61.52 ± 0.28**	53.49 ± 4.61
10	9.19 ± 0.11	36.78 ± 0.19	46.12 ± 0.19	37.23 ± 0.20	**46.74 ± 0.20**	30.22 ± 3.77
20	4.72 ± 0.06	25.18 ± 0.11	33.71 ± 0.12	25.91 ± 0.12	**34.41 ± 0.12**	17.68 ± 1.15
40	2.62 ± 0.02	17.02 ± 0.07	23.69 ± 0.06	17.19 ± 0.08	**25.13 ± 0.07**	8.04 ± 3.00

- On KGs

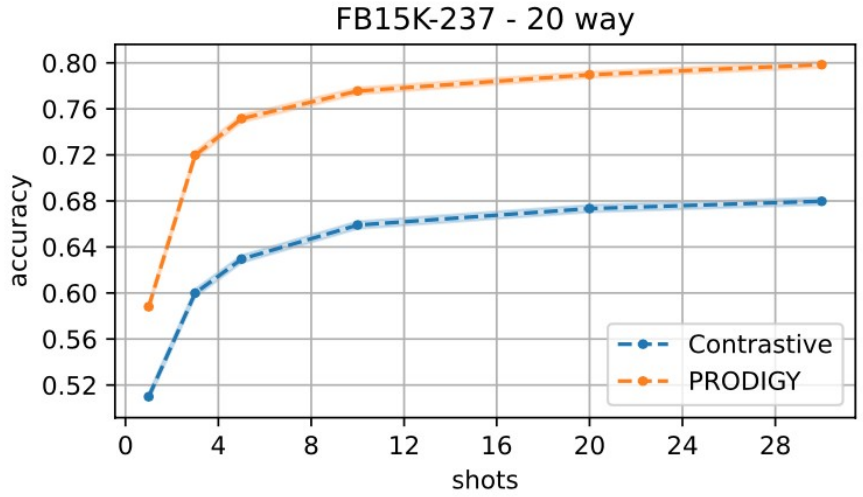
Classes	NoPretrain	Contrastive	PG-NM	PG-MT	PRODIGY	Finetune
4	30.4 ± 0.63	44.01 ± 0.61	46.94 ± 0.61	51.78 ± 0.63	**53.97 ± 0.63**	53.85 ± 9.29
5	33.54 ± 0.61	81.35 ± 0.58	80.35 ± 0.57	89.15 ± 0.46	**88.02 ± 0.48**	82.01 ± 12.83
10	20.0 ± 0.35	70.88 ± 0.48	71.68 ± 0.45	82.26 ± 0.40	**81.1 ± 0.39**	71.97 ± 6.16
20	9.2 ± 0.18	59.8 ± 0.35	59.9 ± 0.35	73.47 ± 0.32	**72.04 ± 0.33**	64.01 ± 4.66
40	2.5 ± 0.08	49.39 ± 0.23	46.82 ± 0.21	58.34 ± 0.22	**59.58 ± 0.22**	57.27 ± 3.33
5	33.44 ± 0.57	84.08 ± 0.54	80.53 ± 0.58	84.79 ± 0.51	87.02 ± 0.44	**87.22 ± 12.75**
10	18.82 ± 0.31	76.54 ± 0.45	72.77 ± 0.48	78.5 ± 0.44	**81.06 ± 0.41**	71.90 ± 5.90
20	7.42 ± 0.16	66.56 ± 0.35	62.82 ± 0.36	69.82 ± 0.34	**72.66 ± 0.32**	66.19 ± 8.46
40	3.04 ± 0.07	57.44 ± 0.24	49.59 ± 0.22	53.55 ± 0.23	**60.02 ± 0.22**	55.06 ± 4.19

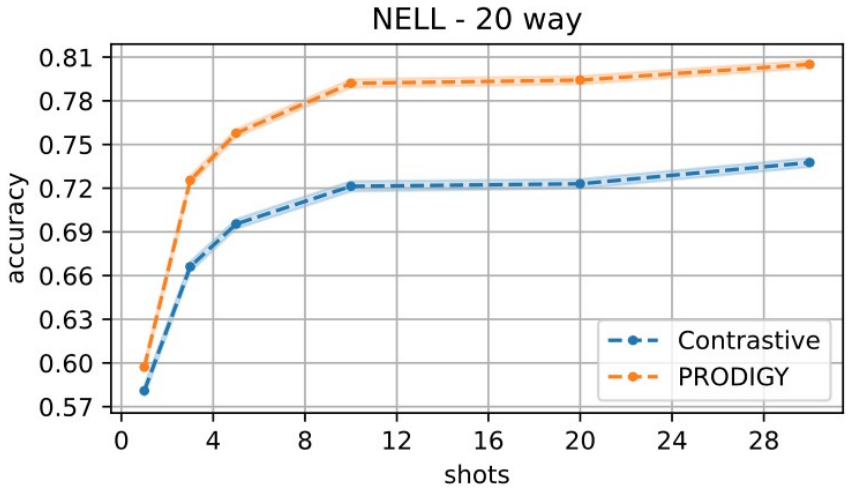
Ablations –

- For PG-NM setting,

Ways	PG-NM	3 →1 shot	No Attr	No Aug	No Attr, Aug	No Attr, Aug, M_T
3	**72.50** ± 0.35	69.13 ± 1.09	65.74 ± 1.12	68.98 ± 1.09	66.53 ± 1.12	63.60 ± 1.06
5	**61.21** ± 0.29	57.49 ± 0.92	52.78 ± 0.90	57.50 ± 0.85	53.89 ± 0.92	51.27 ± 0.69
10	**46.12** ± 0.19	42.03 ± 0.60	37.99 ± 0.63	42.43 ± 0.64	38.87 ± 0.59	37.62 ± 0.34
20	**33.71** ± 0.11	30.18 ± 0.38	26.60 ± 0.36	30.89 ± 0.38	27.50 ± 0.36	27.44 ± 0.17
40	**23.69** ± 0.07	21.44 ± 0.22	18.03 ± 0.21	21.97 ± 0.24	18.52 ± 0.22	19.69 ± 0.08

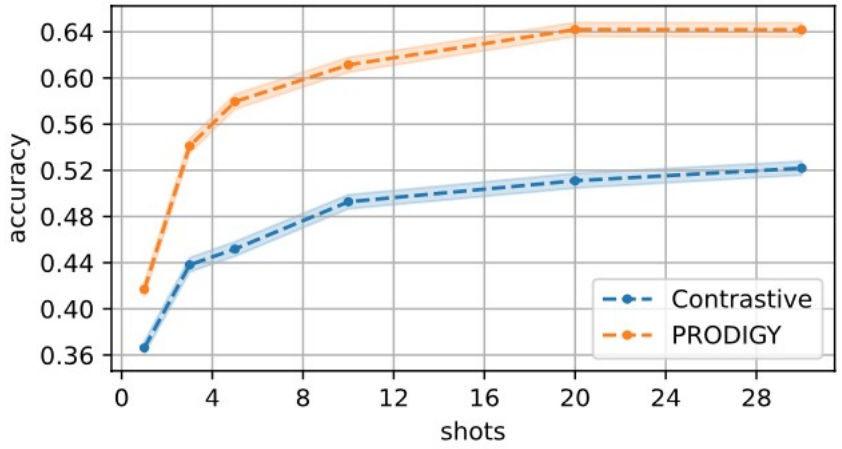
- Number of shots vs contrastive,

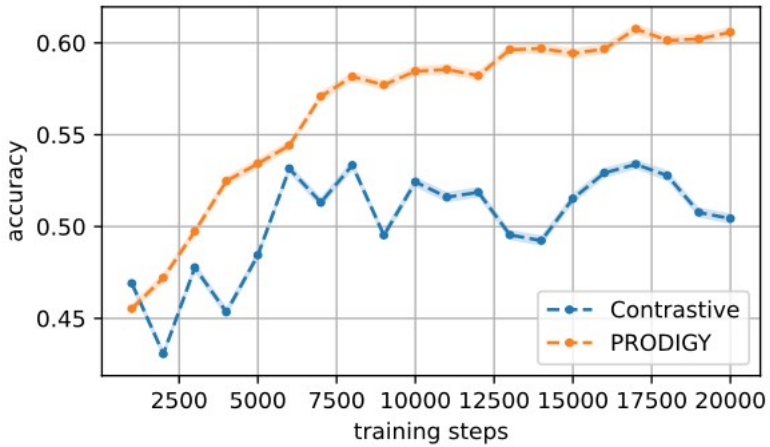




Ablations –

- Scaling w/ data,





Also beats meta-learning based SOTA method.

EQUIVARIANT SUBGRAPH AGGREGATION NETWORKS

Beatrice Bevilacqua*
Purdue University
bbevilac@purdue.edu

Fabrizio Frasca*
Imperial College London & Twitter
ffrasca@twitter.com

Derek Lim*
MIT CSAIL
dereklim@mit.edu

Balasubramaniam Srinivasan
Purdue University
bsriniv@purdue.edu

Chen Cai
UCSD CSE
c1cai@ucsd.edu

Gopinath Balamurugan
University of Tuebingen
gbm0998@gmail.com

Michael M. Bronstein
Imperial College London & Twitter
mbronstein@twitter.com

Haggai Maron
NVIDIA Research
hmaron@nvidia.com

Introduction –

- Improving the expressive power beyond WL test,
- Current approaches, expensive,
- *Encode multisets of subgraphs*,
- Develop a network for **this** symmetry group,
- Subgraph selection which can be a problem, is reduced by stochasticity,
- SOTA results on synthetic and real-world datasets,
- DL on sets.
- Main paper focuses on Graph Classification/Regression.
- Setup –

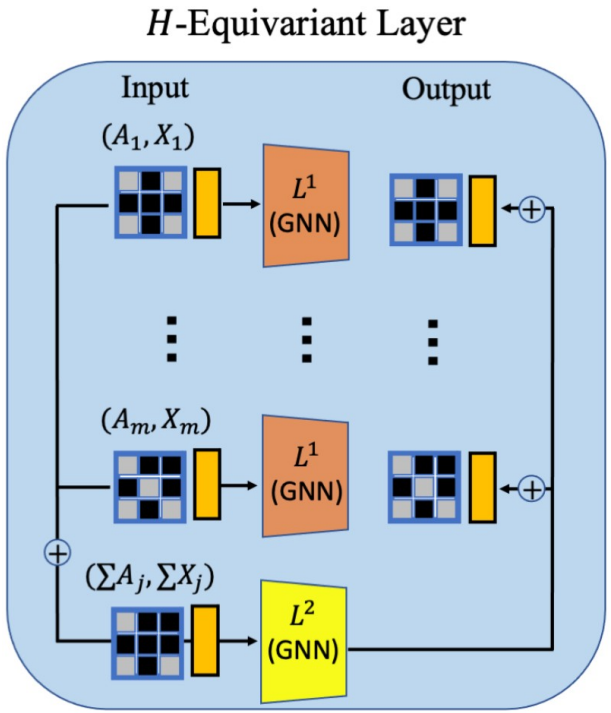
$$S_G = \{\!\{G_1, \ldots, G_m\}\!\}$$

$$F(G) := F(S_G)$$

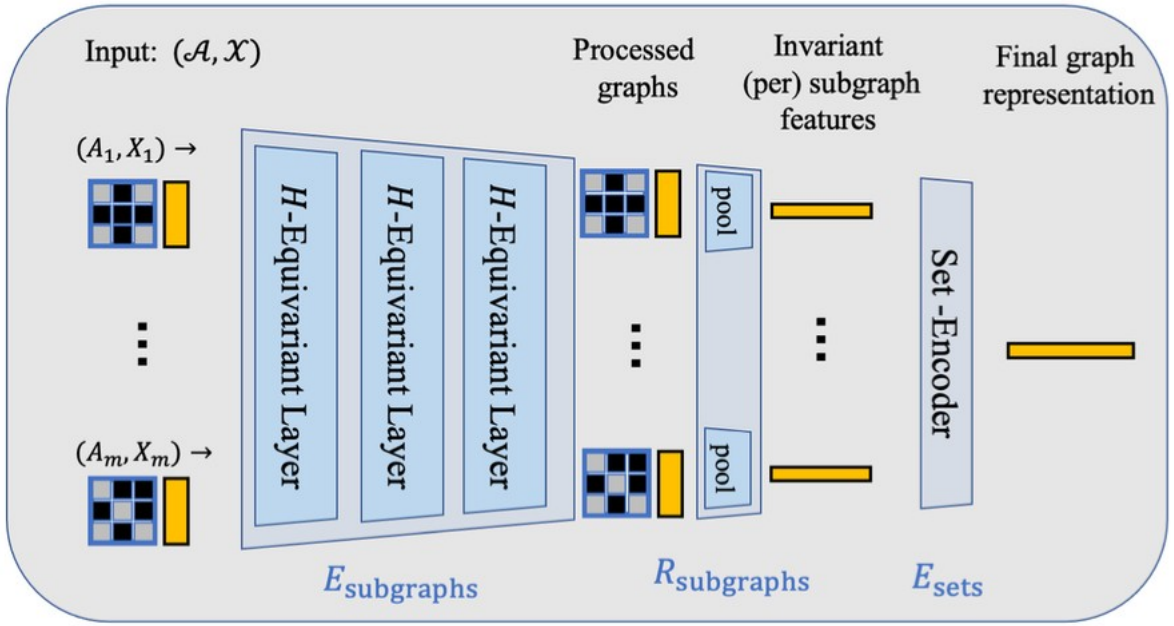
ESAN –

- **Selecting F_s :** Preserving equi-variance under the symmetry group,
- P-equivariant layers,

$$(L(\mathcal{A},\mathcal{X}))_i = L^1(A_i, X_i) + L^2\left(\sum_{j=1}^m A_j, \sum_{j=1}^m X_j\right)$$



$$F_{\text{DSS-GNN}} = E_{\text{sets}} \circ R_{\text{subgraphs}} \circ E_{\text{subgraphs}}$$



ESAN –

- For DS-GNN, L^2 is set to 0.
- **Selecting S_G :** Node-deleted (ND), Edge-deleted (ED), and ego-networks (EGO, EGO+).
- Stochastic Sampling, subgraph sub-sampling for large graphs.

$$\overline{S}_G^{\pi} \subset S_G^{\pi}$$
$$|\overline{S}_G^{\pi}|/|S_G^{\pi}| \in \{0.05, 0.2, 0.5\}$$

- Invariance lost.
- Different from previous works.
- Theoretical Analysis –
  - First, they have provided a ESAN analogue of the WL test.
  - And then,

Theorem 1 (DS(S)-WL strictly more powerful than 1-WL)**.** *There exist selection policies such that DS(S)-WL is* strictly more powerful *than 1-WL in distinguishing between non-isomorphic graphs.*

ESAN –

- Theoretical Analysis Contd. –

Theorem 2 (DS(S)-GNN at least as powerful as DS(S)-WL; DS-GNN at most as powerful as DS-WL)**.** *Let $\mathcal{F}$ be any family of bounded-sized graphs endowed with node labels from a finite set. There exist selection policies such that, for any two graphs G^1, G^2 in $\mathcal{F}$, distinguished by DS(S)-WL, there is a DS(S)-GNN model in the form of Equation (2) assigning G^1, G^2 distinct representations. Also, DS-GNN with MPNN base graph encoder is at most as powerful as DS-WL.*

DSS vs. DS matters. To continue the discussion of the last section, we show that DSS-GNN is at least as powerful as DS-GNN, and is in fact strictly stronger than DS-GNN for a specific policy.

Experiments –

- On synthetic datasets

	EXP	CEXP
GIN (Xu et al., 2019)	51.1±2.1	70.2±4.1
GIN + ID-GNN (You et al., 2021)	100±0.0	100±0.0
DS-GNN (GIN) (ED/ND/EGO/EGO+)	100±0.0	100±0.0
DSS-GNN (GIN) (ED/ND/EGO/EGO+)	100±0.0	100±0.0
GRAPHCONV (Morris et al., 2019)	50.3±2.6	72.9±3.6
GRAPHCONV + ID-GNN (You et al., 2021)	100±0.0	100±0.0
DS-GNN (GRAPHCONV) (ED/ND/EGO/EGO+)	100±0.0	100±0.0
DSS-GNN (GRAPHCONV) (ED/ND/EGO/EGO+)	100±0.0	100±0.0

Non-Stochastic Variant

		OGBG-MOLHIV	OGBG-MOLTOX21	EXP	CEXP
GIN (Xu et al., 2019)		75.58±1.40	74.91±0.51	51.2±2.1	70.2±4.1
DS-GNN (GIN) (ED)	**100%**	76.43±2.12	75.12±0.50	100±0.0	100±0.0
	50%	76.29±1.33	74.59±0.71	100±0.0	100±0.0
	20%	76.57±1.48	75.67±0.89	100±0.0	99.9±0.2
	5%	77.82±1.00	76.39±1.11	99.7±0.4	99.9±0.2
DS-GNN (GIN) (ND)	**100%**	76.19±0.96	75.34±1.21	100±0.0	100±0.0
	50%	77.23±1.32	74.82±1.05	100±0.0	99.9±0.2
	20%	77.65±0.84	75.66±0.46	100±0.0	99.9±0.2
	5%	78.26±1.02	76.51±1.04	97.2±1.1	99.8±0.8
DS-GNN (GIN) (EGO)	**100%**	78.00±1.42	76.22±0.62	100±0.0	100±0.0
	50%	76.52±0.72	75.98±0.72	100±0.0	99.9±0.2
	20%	77.49±1.32	75.88±0.50	99.9±0.2	96.8±1.5
	5%	73.92±1.78	74.95±0.54	93.5±1.3	83.9±3.8
DS-GNN (GIN) (EGO+)	**100%**	77.40±2.19	76.39±1.18	100±0.0	100±0.0
	50%	76.91±1.22	75.69±1.17	100±0.0	99.9±0.2
	20%	75.92±1.59	75.84±0.63	99.7±0.4	97.0±1.4
	5%	73.46±1.80	75.08±0.96	93.7±2.7	83.2±2.6
DSS-GNN (GIN) (ED)	**100%**	77.03±1.81	76.71±0.67	100±0.0	100±0.0
	50%	77.50±1.82	76.40±0.84	100±0.0	100±0.0
	20%	76.82±1.83	76.31±0.90	100±0.0	100±0.0
	5%	76.71±1.46	76.84±0.54	99.8±0.3	100±0.0
DSS-GNN (GIN) (ND)	**100%**	76.63±1.52	77.21±0.70	100±0.0	100±0.0
	50%	76.96±1.71	76.92±0.94	100±0.0	100±0.0
	20%	76.23±1.48	77.07±1.03	100±0.0	100±0.0
	5%	76.74±1.67	76.54±0.86	97.7±1.0	99.9±0.2
DSS-GNN (GIN) (EGO)	**100%**	77.19±1.27	77.45±0.41	100±0.0	100±0.0
	50%	76.42±1.38	76.37±1.02	100±0.0	100±0.0
	20%	76.41±1.05	77.47±0.65	100±0.0	100±0.0
	5%	76.38±1.48	77.40±0.58	99.2±0.6	100±0.0
DSS-GNN (GIN) (EGO+)	**100%**	76.78±1.66	77.95±0.40	100±0.0	100±0.0
	50%	76.88±0.93	76.42±0.93	100±0.0	100±0.0
	20%	76.93±1.45	76.45±0.81	100±0.0	100±0.0
	5%	75.97±0.80	76.70±0.56	99.5±0.6	100±0.0

Experiments –

- On OGB

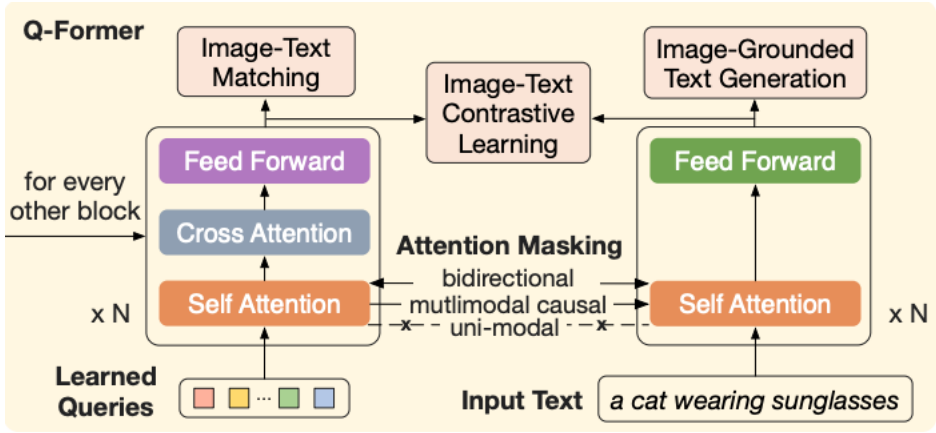
Method	OGBG-MOLHIV ROC-AUC (%)	OGBG-MOLTOX21 ROC-AUC (%)
GIN (Xu et al., 2019)	75.58±1.40	74.91±0.51
DS-GNN (GIN) (ED)	76.43±2.12	75.12±0.50
DS-GNN (GIN) (ND)	76.19±0.96	75.34±1.21
DS-GNN (GIN) (EGO)	78.00±1.42	76.22±0.62
DS-GNN (GIN) (EGO+)	77.40±2.19	76.39±1.18
DSS-GNN (GIN) (ED)	77.03±1.81	76.71±0.67
DSS-GNN (GIN) (ND)	76.63±1.52	77.21±0.70
DSS-GNN (GIN) (EGO)	77.19±1.27	77.45±0.41
DSS-GNN (GIN) (EGO+)	76.78±1.66	77.95±0.40

- Zinc12k

PNA (Corso et al., 2020)	0.188±0.004
DGN (Beaini et al., 2021)	0.168±0.003
SMP (Vignac et al., 2020)	0.138±?
GIN (Xu et al., 2019)	0.252±0.017
HIMP (Fey et al., 2020)	0.151±0.006
GSN (Bouritsas et al., 2022)	0.108±0.018
CIN-SMALL (Bodnar et al., 2021a)	0.094±0.004
DS-GNN (GIN) (ED)	0.172±0.008
DS-GNN (GIN) (ND)	0.171±0.010
DS-GNN (GIN) (EGO)	0.126±0.006
DS-GNN (GIN) (EGO+)	0.116±0.009
DSS-GNN (GIN) (ED)	0.172±0.005
DSS-GNN (GIN) (ND)	0.166±0.004
DSS-GNN (GIN) (EGO)	0.107±0.005
DSS-GNN (GIN) (EGO+)	0.102±0.003

Idea2 –

- Still, separate Molecular and KG GNNs, but a better, more expressive ”adapter” in between, i.e. **learned query tokens**. Something like,



- Have full Molecular and KG GNNs (**not frozen**), and on top a few 3-4 layers of this type of an interaction module (between [VNode] and mol. emb. from KG).

- Same losses as in Gode, but for contrastive similarity use all *query tokens*.

Idea2 –

- Benefits –
  - Can allow for query tokens (because shared) to capture “global” properties instead of being limited to a subgraph in the KG.
  - May be parameter efficient?
  - No information bottleneck, free communication through attention mechanism.
  - Can use query tokens in a variety of ways, for in-context learning as well, combining prev. paper and MHNfs, which is not limited to classification tasks.
  - This way we can also do semi-inductive prediction over entities, relations must still be transductive.