Systematic reasoning using Epistemic Graph Neural Networks

Irtaza Khalid

Contributions

[Cont1] Khalid, I.; Schockaert, S. STaR: Benchmarking Spatio-Temporal Reasoning for Systematic Generalization. Sys2-reasoning-at-scale@NeurIPS workshop (2024)

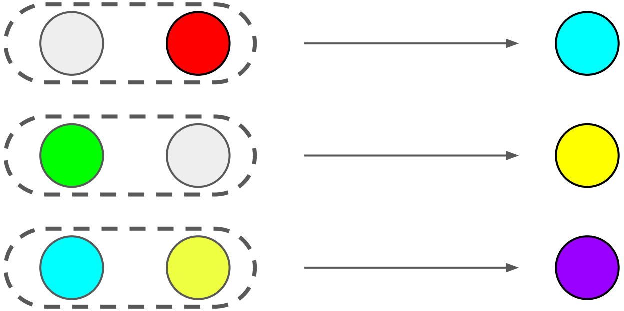
[Cont2] Khalid, I.; Schockaert, S. Systematic Relational Reasoning With Epistemic Graph Neural Networks (submitted to ICLR 2025)

Overview

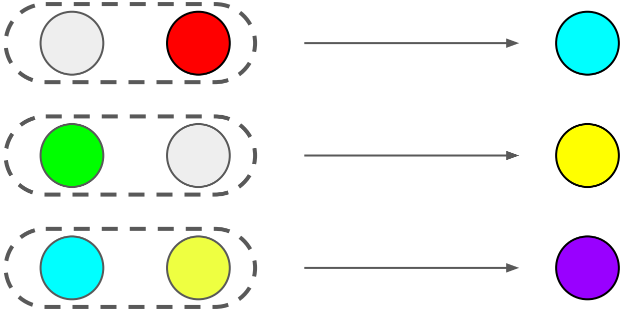
1. What is systematic reasoning and why is it important?
2. Generalising the conjunctive reasoning problem
3. Limitations of current approaches
4. Enter: EpiGNN
5. Results
    a. Relation Prediction (eval systematicity)
    b. Inductive Knowledge graph completion (eval scalability and performance preservation)
    c. Parameter efficiency
6. Outlook
    a. Takeaways
    b. What I’m currently working on.

What is systematic reasoning and why is it important?

What is Systematic reasoning?

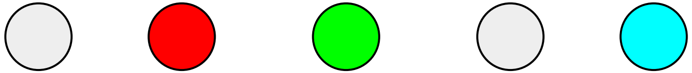


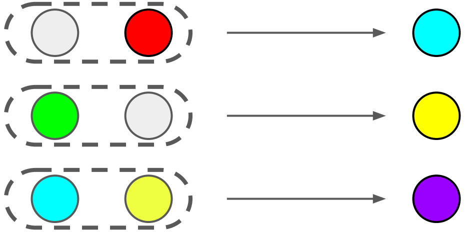
What is Systematic reasoning?



given base compositions

What is Systematic reasoning?

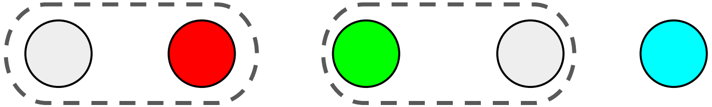


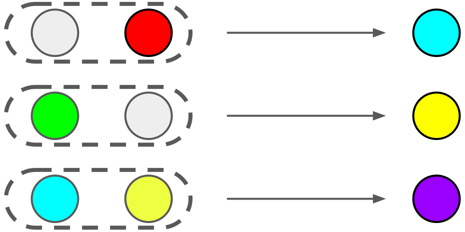


evaluate the ability of a machine to recombine base compositions to collapse increasingly long sequences

Hupkes et. al. 2020

What is Systematic reasoning?

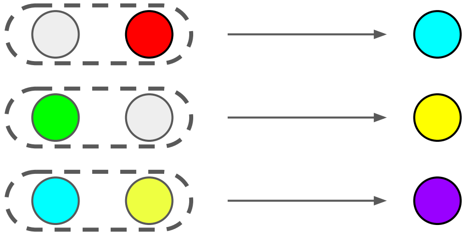


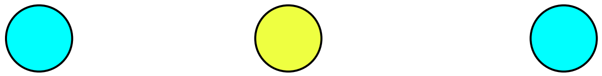


evaluate the ability of a machine to recombine base compositions to collapse increasingly long sequences

Hupkes et. al. 2018

What is Systematic reasoning?

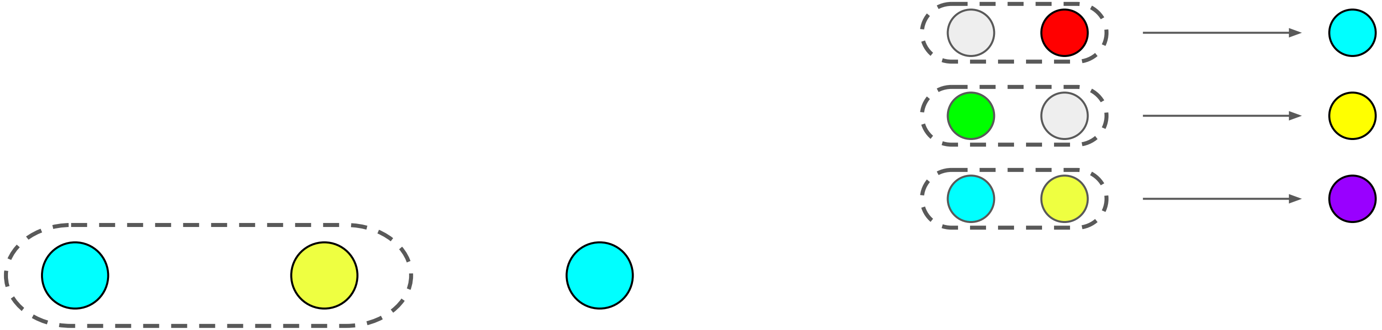




evaluate the ability of a machine to recombine base compositions to collapse increasingly long sequences

Hupkes et. al. 2020

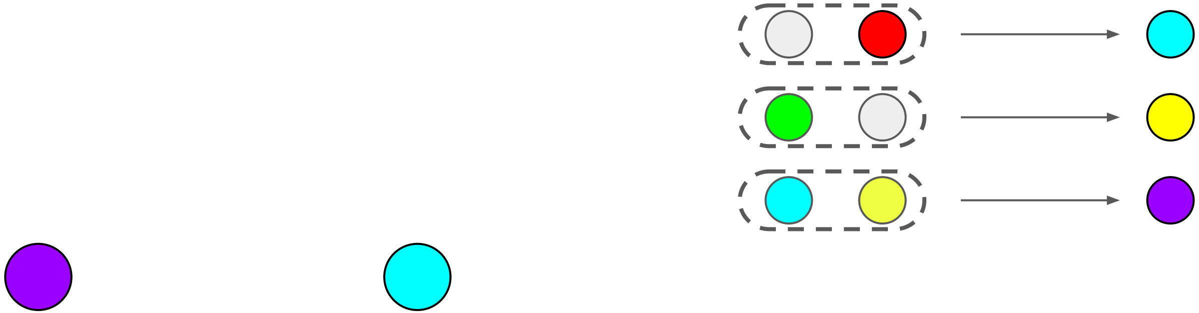
What is Systematic reasoning?



evaluate the ability of a machine to recombine base compositions to collapse increasingly long sequences

Hupkes et. al. 2020

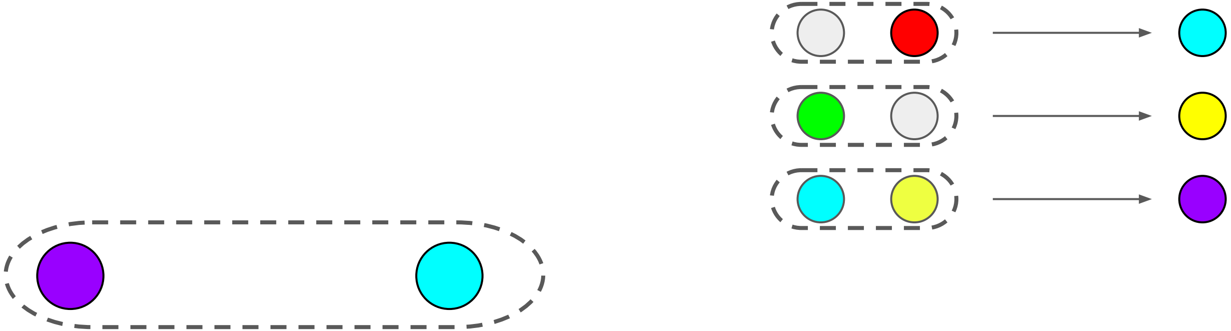
What is Systematic reasoning?



evaluate the ability of a machine to recombine base compositions to collapse increasingly long sequences

Hupkes et. al. 2020

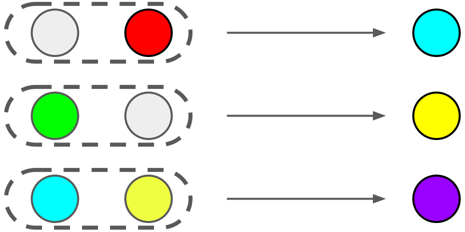
What is Systematic reasoning?

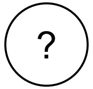


evaluate the ability of a machine to recombine base compositions to collapse increasingly long sequences

Hupkes et. al. 2020

What is Systematic reasoning?



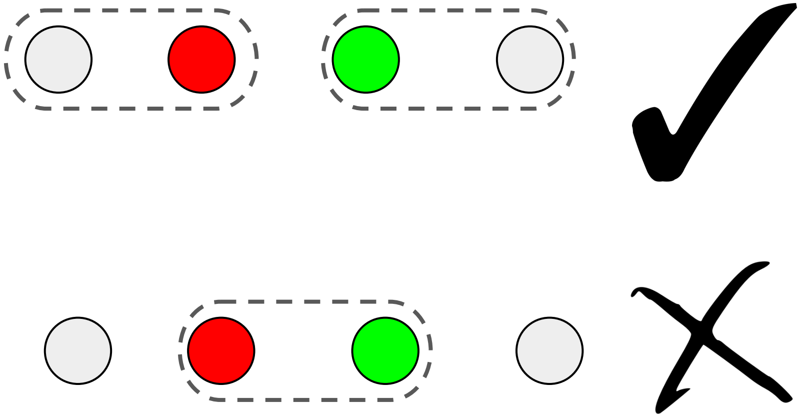


evaluate the ability of a machine to recombine base compositions to collapse increasingly long sequences

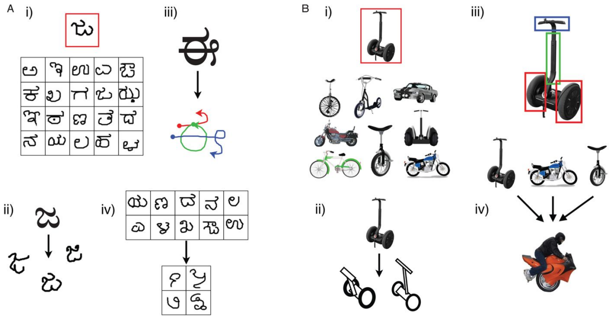
Hupkes et. al. 2020

Issues that are inherent here:

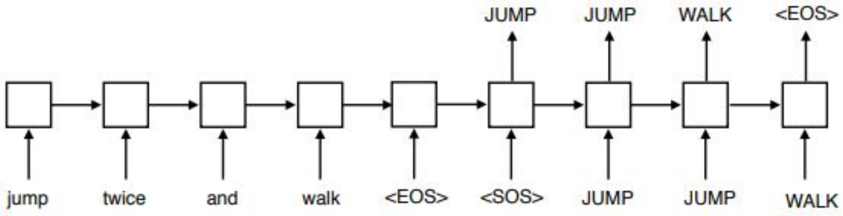
1. Order of composition matters: to remain in distribution with the training data.
2. If it’s impossible to avoid unknown compositions, how do we treat them?



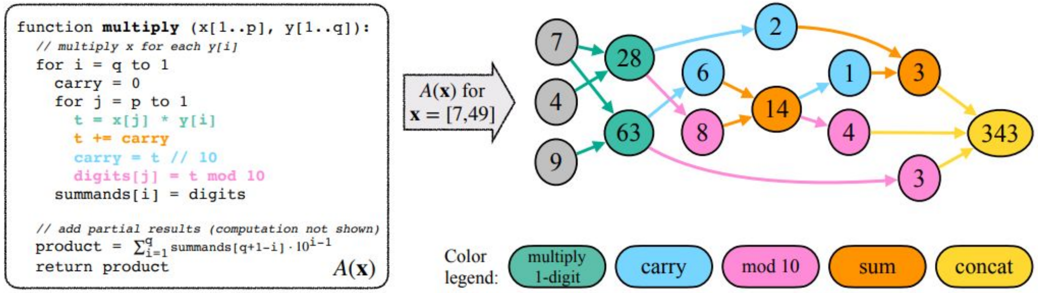
Compositional learning unlocks human-like learning with sparse data



Lake et. al. 2017



Lake et. al. 2018



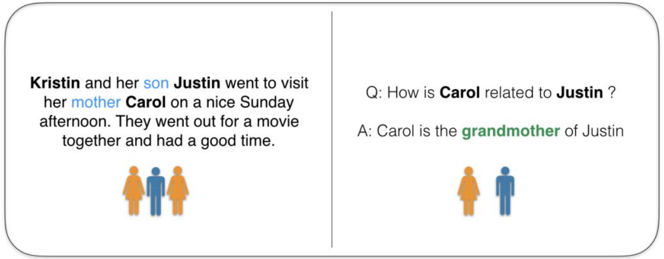
```
function multiply (x[1..p], y[1..q]):
  // multiply x for each y[i]
  for i = q to 1
    carry = 0
    for j = p to 1
      t = x[j] * y[i]
      t += carry
      carry = t // 10
      digits[j] = t mod 10
    summands[i] = digits

  // add partial results (computation not shown)
  product = $\sum_{i=1}^{q}$ summands[q+1-i] · $10^{i-1}$
  return product
```

Dziri et. al. 2023

How to measure systematic *relational* reasoning?
⇒ link prediction problem: (s,?,t)

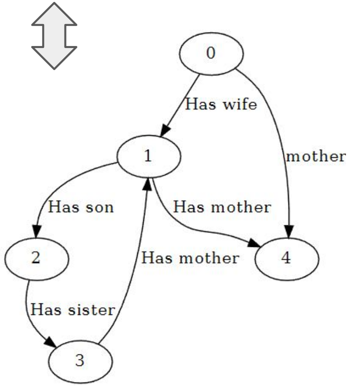
Train on small graphs
(path length from source to sink: k=2,3,4)



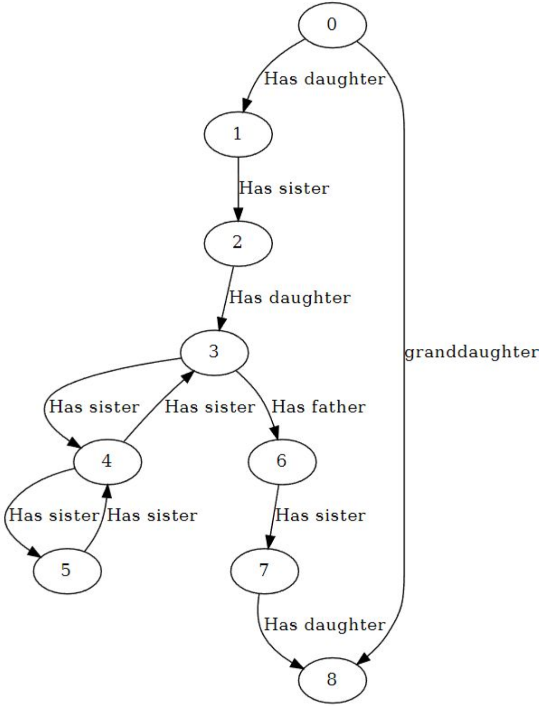
Kristin and her son **Justin** went to visit her mother **Carol** on a nice Sunday afternoon. They went out for a movie together and had a good time.

Q: How is **Carol** related to **Justin** ?

A: Carol is the **grandmother** of Justin



Test on increasingly large graphs
(path length k=2,3,4,5,...,10)

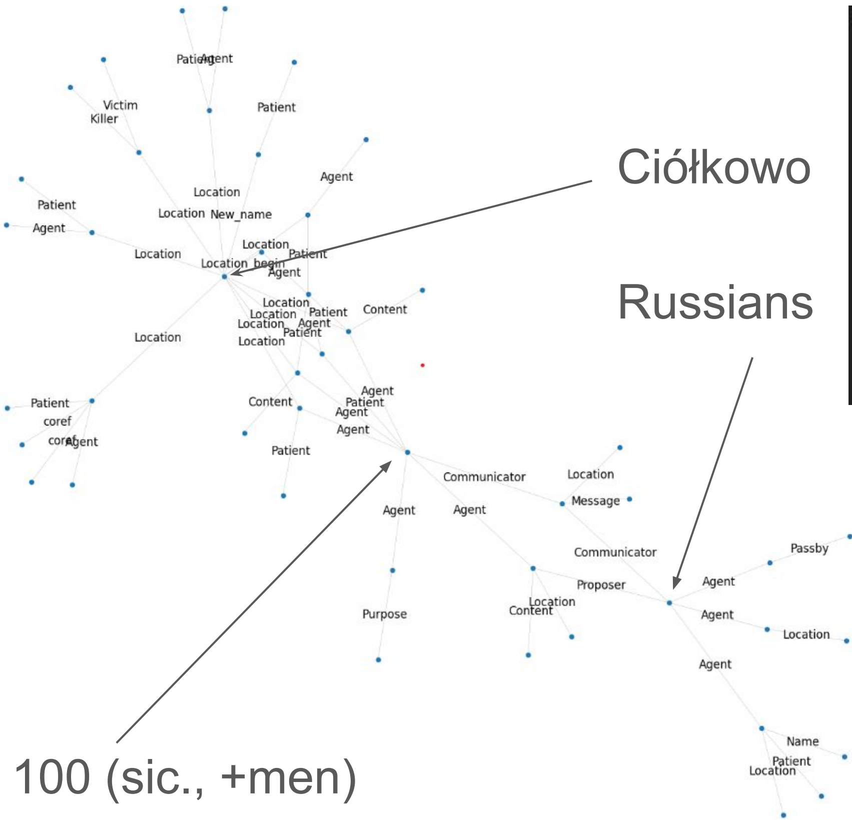


CLUTRR (Sinha et. al. 2022)

Limitations of current methods

1. Though neuro-symbolic (NeSy) or neural-theorem-prover type methods are good at these problems,
    a. They are too specialized for single path conjunctive reasoning
    b. They are parameter-inefficient and generally not very scalable to large graphs
2. Non-NeSy *statistical* methods are generally poor at systematic reasoning. We argue that is because
    a. They lack an algorithmical alignment bias (Xu et. al. 2021) wrt. the systematicity task
    b. They are prone to learning shortcuts (spurious patterns) (Gierhos et. al.)

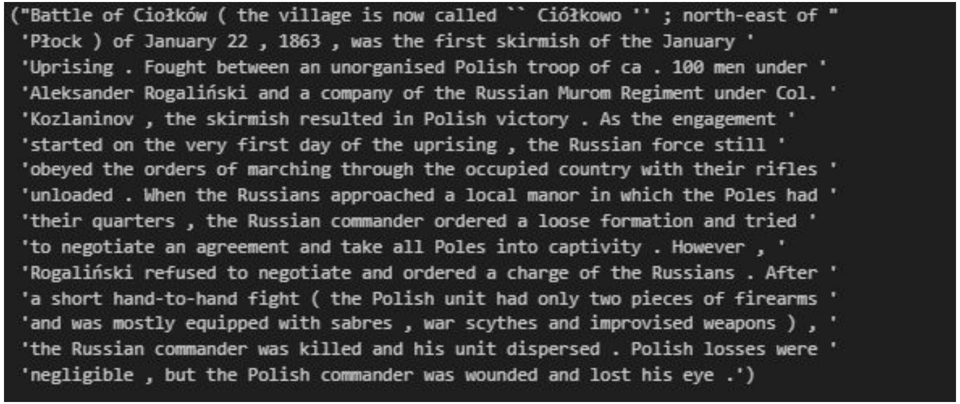
Why single-path reasoning is not realistic! e.g. story graphs



Ciółkowo

Russians

100 (sic., +men)



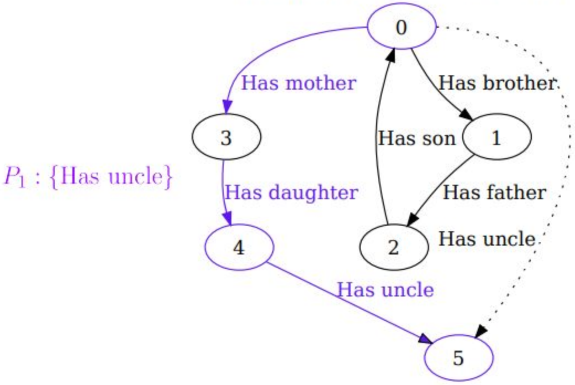
```
("Battle of Ciołków ( the village is now called `` Ciółkowo '' ; north-east of "
 'Płock ) of January 22 , 1863 , was the first skirmish of the January '
 'Uprising . Fought between an unorganised Polish troop of ca . 100 men under '
 'Aleksander Rogaliński and a company of the Russian Murom Regiment under Col. '
 'Kozlaninov , the skirmish resulted in Polish victory . As the engagement '
 'started on the very first day of the uprising , the Russian force still '
 'obeyed the orders of marching through the occupied country with their rifles '
 'unloaded . When the Russians approached a local manor in which the Poles had '
 'their quarters , the Russian commander ordered a loose formation and tried '
 'to negotiate an agreement and take all Poles into captivity . However , '
 'Rogaliński refused to negotiate and ordered a charge of the Russians . After '
 'a short hand-to-hand fight ( the Polish unit had only two pieces of firearms '
 'and was mostly equipped with sabres , war scythes and improvised weapons ) , '
 'the Russian commander was killed and his unit dispersed . Polish losses were '
 'negligible , but the Polish commander was wounded and lost his eye .')
```

We expect transferability of better systematic reasoning to story graph reasoning.

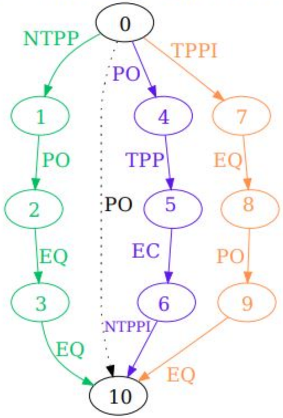
Especially in the long-hop, multi-path setting.

[Cont1] New benchmarks using spatio-temporal calculi

Single-path reasoning



Multi-path disjunctive reasoning

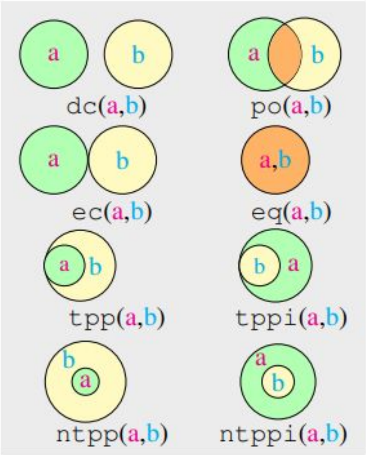


P_1 : {DC, EC, PO, TPP, NTPP}

P_2 : {NTPPI, DC, EC, TPPI, PO}

P_3 : {TPPI, NTPPI, PO}

$P_1 \cap P_2 \cap P_3$: {PO}



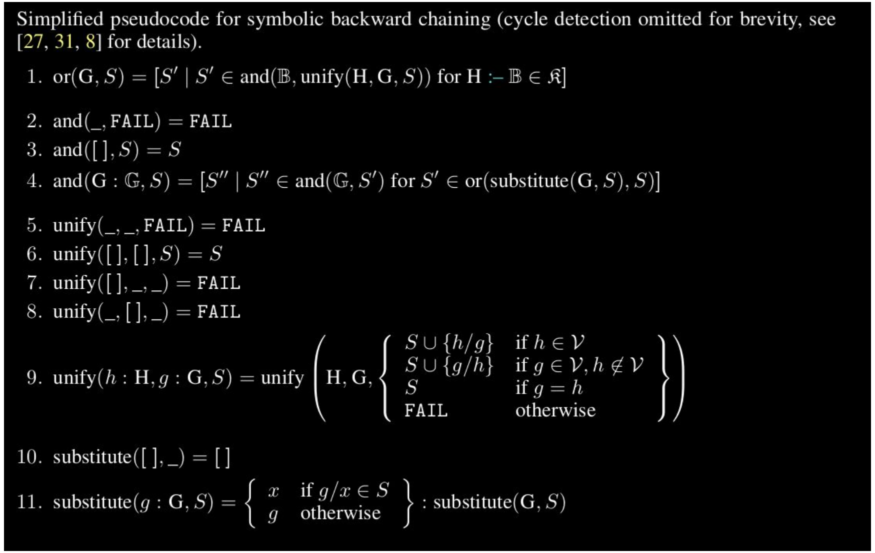
$$r_3(X, Z) \leftarrow r_1(X, Y) \wedge r_2(Y, Z)$$

$$s_1(X, Z) \vee \ldots \vee s_k(X, Z) \leftarrow r_1(X, Y) \wedge r_2(Y, Z)$$

$$(s_1 \vee \cdots \vee s_k)_{\text{path}_1} \wedge \cdots \wedge (s_1 \vee \cdots \vee s_k)_{\text{path}_b}$$

Limitations of current approaches

1- NeSy is not scalable and is often single-path-focussed



Simplified pseudocode for symbolic backward chaining (cycle detection omitted for brevity, see [27, 31, 8] for details).

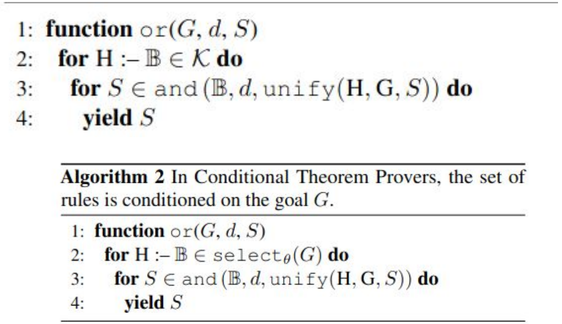
```
$\text{or}(\text{G}, S) = [S' \mid S' \in \text{and}(\mathbb{B}, \text{unify}(\text{H}, \text{G}, S)) \text{ for H :– } \mathbb{B} \in \mathfrak{K}]$

$\text{and}(\_, \texttt{FAIL}) = \texttt{FAIL}$
$\text{and}([\,], S) = S$
$\text{and}(\text{G} : \mathbb{G}, S) = [S'' \mid S'' \in \text{and}(\mathbb{G}, S') \text{ for } S' \in \text{or}(\text{substitute}(\text{G}, S), S)]$

$\text{unify}(\_, \_, \texttt{FAIL}) = \texttt{FAIL}$
$\text{unify}([\,], [\,], S) = S$
$\text{unify}([\,], \_, \_) = \texttt{FAIL}$
$\text{unify}(\_, [\,], \_) = \texttt{FAIL}$
$\text{unify}(h : \text{H}, g : \text{G}, S) = \text{unify}\left(\text{H}, \text{G}, \left\{\begin{array}{ll} S \cup \{h/g\} & \text{if } h \in \mathcal{V} \\ S \cup \{g/h\} & \text{if } g \in \mathcal{V}, h \notin \mathcal{V} \\ S & \text{if } g = h \\ \texttt{FAIL} & \text{otherwise} \end{array}\right\}\right)$

$\text{substitute}([\,], \_) = [\,]$
$\text{substitute}(g : \text{G}, S) = \left\{\begin{array}{ll} x & \text{if } g/x \in S \\ g & \text{otherwise} \end{array}\right\} : \text{substitute}(\text{G}, S)$
```

NTP; Rocktäschel et. al. 2018

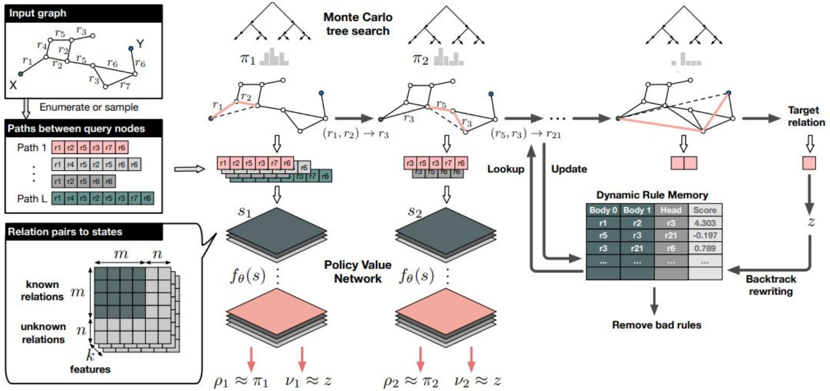


```
function or$(G, d, S)$
  for H :– $\mathbb{B} \in \mathcal{K}$ do
    for $S \in$ and$(\mathbb{B}, d,$ unify$(\text{H}, \text{G}, S))$ do
      yield $S$
```

Algorithm 2 In Conditional Theorem Provers, the set of rules is conditioned on the goal G.

```
function or$(G, d, S)$
  for H :– $\mathbb{B} \in$ select$_\theta(G)$ do
    for $S \in$ and$(\mathbb{B}, d,$ unify$(\text{H}, \text{G}, S))$ do
      yield $S$
```

CTP; Minervini et. al. 2020

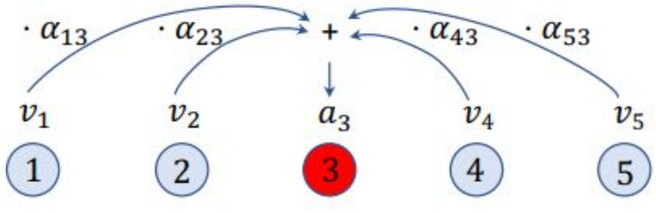


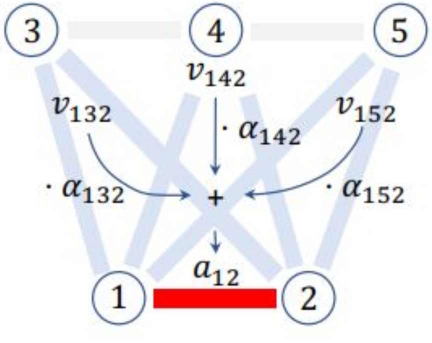
Dynamic Rule Memory

Body 0	Body 1	Head	Score
r1	r2	r3	4.303
r5	r3	r21	-0.197
r3	r21	r6	0.789
...	...	...	...

R5; Lu et. al. 2022

2 - Edge Transformers don’t scale to large graphs: n^3 vs ~n^2



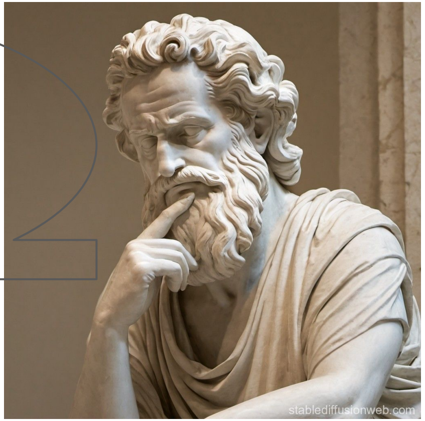


Edge Transformer (ET); Bergen et. al. 2021

The main issue is the dense graph/ all-pairs connectivity in the edge-attention mechanism.

3 - GNNs like NBFNet (Zhu et. al. 2021) are scalable but lack systematicity

Know what you know
and don’t know



[Cont2] EpiGNN: the Epistemic Graph Neural Network

GNN refresher: Operationalized Weisfeiler-Lehman test

GNN ops for node v for k iterations with neighbourhood $\mathcal{N}$

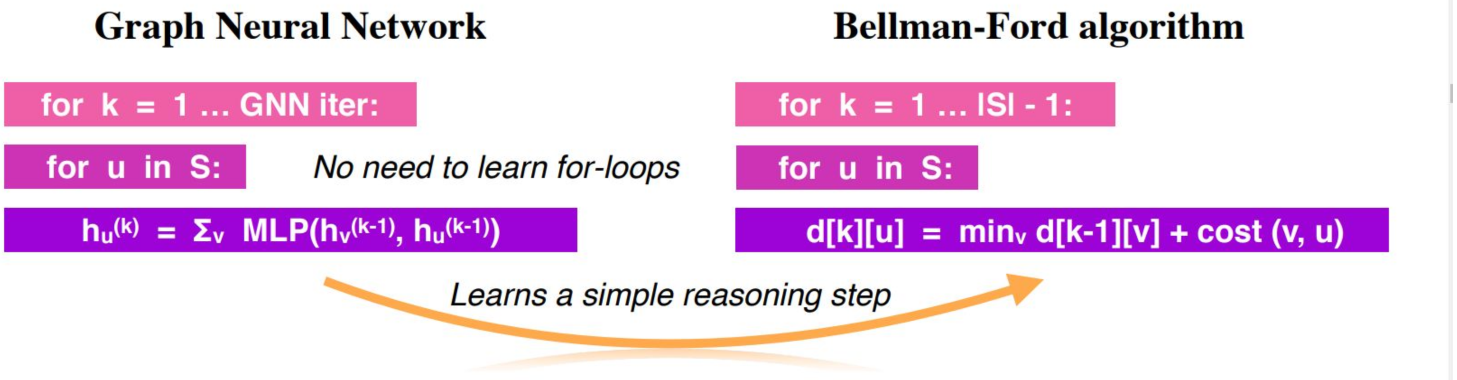
$$a_v^{(k)} = \mathrm{AGGREGATE}^{(k)}\left(\left\{h_u^{(k-1)} : u \in \mathcal{N}(v)\right\}\right), \quad h_v^{(k)} = \mathrm{COMBINE}^{(k)}\left(h_v^{(k-1)}, a_v^{(k)}\right)$$

Weisfeiler-Lehman for k iterations for node v (roughly, iteratively encode node neighbourhoods into its colour, modulo graph isomorphism):

$$c_v^{(k)} = \mathrm{HASH}\left(\left(c_v^{(k-1)}, \{\{c_u^{(k-1)} | u \in \mathcal{N}(v)\}\}\right)\right)$$

In practice, AGGREGATE and COMBINE are learnable non-linear functions like MLPs.

Idea: Algorithmically aligning the architecture with the algorithm that solves the task aids generalization (Xu et. al. 2020)



Graph Neural Network

```
for k = 1 ... GNN iter:
    for u in S:
        $h_u^{(k)} = \Sigma_v \text{ MLP}(h_v^{(k-1)}, h_u^{(k-1)})$
```

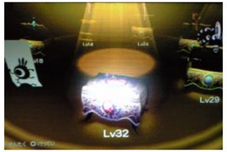
No need to learn for-loops

Bellman-Ford algorithm

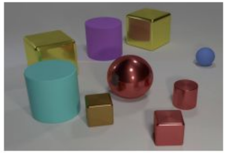
```
for k = 1 ... ISI - 1:
    for u in S:
        $d[k][u] = \min_v d[k-1][v] + \text{cost}(v, u)$
```

Learns a simple reasoning step

MLPs have to learn for-loops that GNNs don’t so tasks unified by dynamic programming are more sample efficiently learned

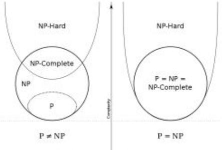


Summary statistics
What is the maximum value difference among treasures?



Relational argmax
What are the colors of the furthest pair of objects?

Dynamic programming
What is the cost to defeat monster X by following the optimal path?



NP-hard problem
Subset sum: Is there a subset that sums to 0?

So, algorithmically align the GNN with the algebraic closure algorithm that “solves” the general reasoning problem

Use NBFNet for relational embeddings:

1. Initialize probabilistic embeddings that can encode unions of base relations
2. Use a COMBINE function that simulates discrete relational composition.
3. Use an AGGREGATE function that simulates intersection.
4. Repeat for k rounds.

Main steps in algebraic closure are:

1. Initialization of node embeddings for all possible relations.
2. Compute all possible discrete relational compositions.
3. Update node embeddings via intersection with the set from step 2.
4. Repeat step 2 for k iterations.

Detailed: Neural vs. Classical

1. Initialization

$$\mathbf{e}^{(0)} = \begin{cases} (1,0,\ldots,0) & \text{if } e = h \\ (\frac{1}{n},\ldots,\frac{1}{n}) & \text{otherwise} \end{cases}$$

$$X_{ef}^{(0)} = \begin{cases} \{r\} & \text{if } r(e,f) \in \mathcal{F} \\ \{\hat{r}\} & \text{if } r(f,e) \in \mathcal{F} \\ \mathcal{R} & \text{otherwise} \end{cases}$$

2. Message passing (COMBINE) composition

$$\phi((f_1,\ldots,f_n),(r_1,\ldots,r_n)) = \sum_{i=1}^{n} \sum_{j=1}^{n} f_i r_j \mathbf{a}_{ij}$$

$$X_{eg}^{(i-1)} \diamond X_{gf}^{(i-1)} = \bigcup \left\{ r \circ s | r \in X_{eg}^{(i-1)}, s \in X_{gf}^{(i-1)} \right\}$$

3. Aggregation: ψ `min` or `mul`

$$\mathbf{e}^{(l)} = \psi(\{\mathbf{e}^{(l-1)}\} \cup \{\phi(\mathbf{r}, \mathbf{f}^{(l-1)}) \,|\, r(e,f) \in \mathcal{F}\})$$

$$X_{ef}^{(i)} = X_{ef}^{(i-1)} \cap \bigcap \{ X_{eg}^{(i-1)} \diamond X_{gf}^{(i-1)} \,|\, g \in \mathcal{E} \}$$

Empirical mods that boost performance:

1. forward-backward compositions

$$\mathbf{s} = \psi(\{\mathbf{t}^{\rightarrow}, \mathbf{h}^{\leftarrow}\} \cup \{\phi(\mathbf{e}^{\rightarrow}, \mathbf{e}^{\leftarrow}) \,|\, e \in \mathcal{E}_{h,t}\})$$

2. Multi-faceted embeddings to encode hierarchical or distributed representations (Hinton et. al. 1983). Analogous to multiple parallel models or multi-head attention.

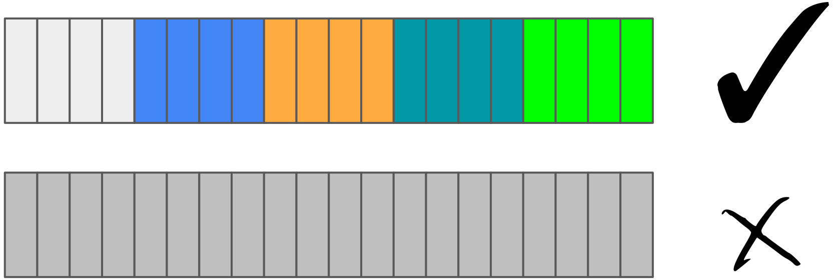


Illustration k=0 forward flow

→ forward flow

← backward flow

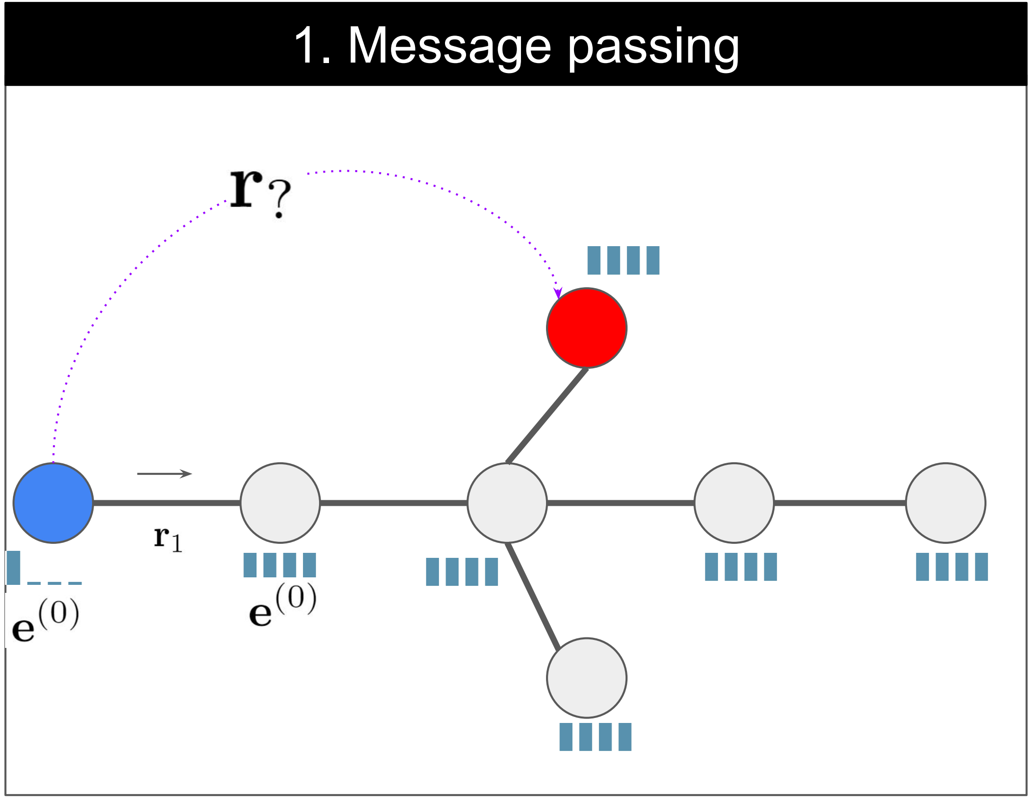
🔵 source

🔴 tail

● final

ϕ message

ψ aggregate



1. Message passing

Illustration k=1 forward flow

→ forward flow

← backward flow

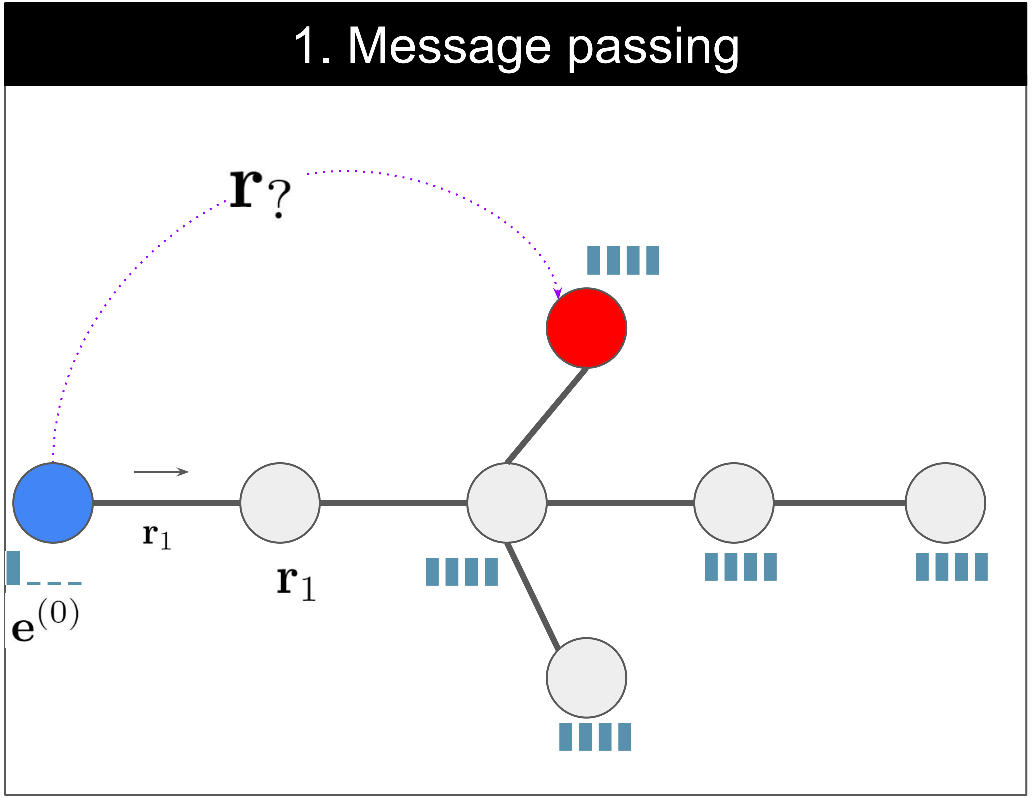
● source

● tail

● final

ϕ message

ψ aggregate



1. Message passing

Illustration k=2 forward flow

→ forward flow

← backward flow

● source

● tail

● final

ϕ message

ψ aggregate

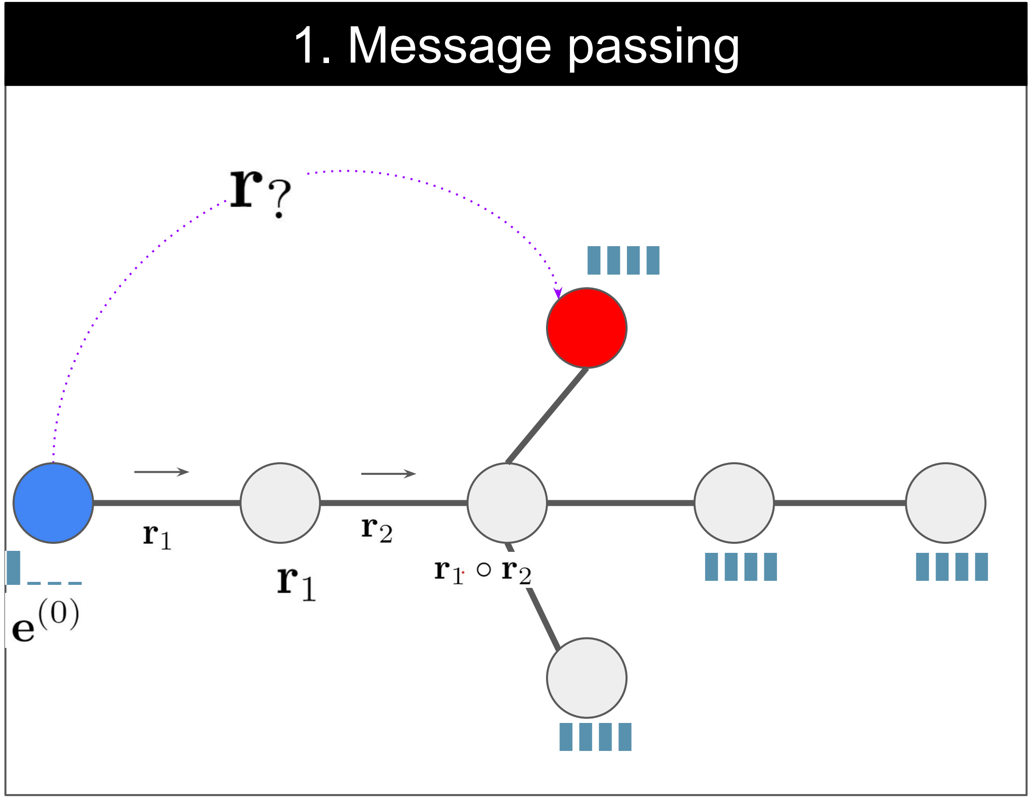


Illustration k=3 forward flow

→ forward flow

← backward flow

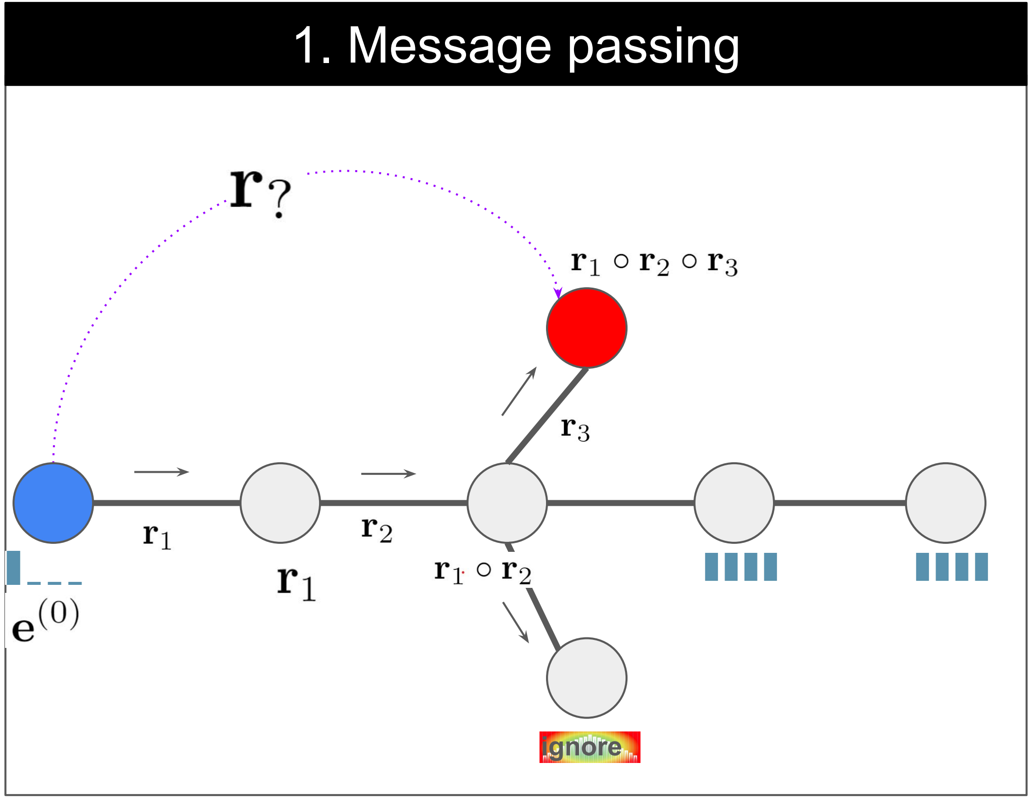
🔵 source

🔴 tail

● final

ϕ message

ψ aggregate



1. Message passing

Illustration k=0 backward flow

→ forward flow

← backward flow

● source

● tail

● final

ϕ message

ψ aggregate

1. Message passing

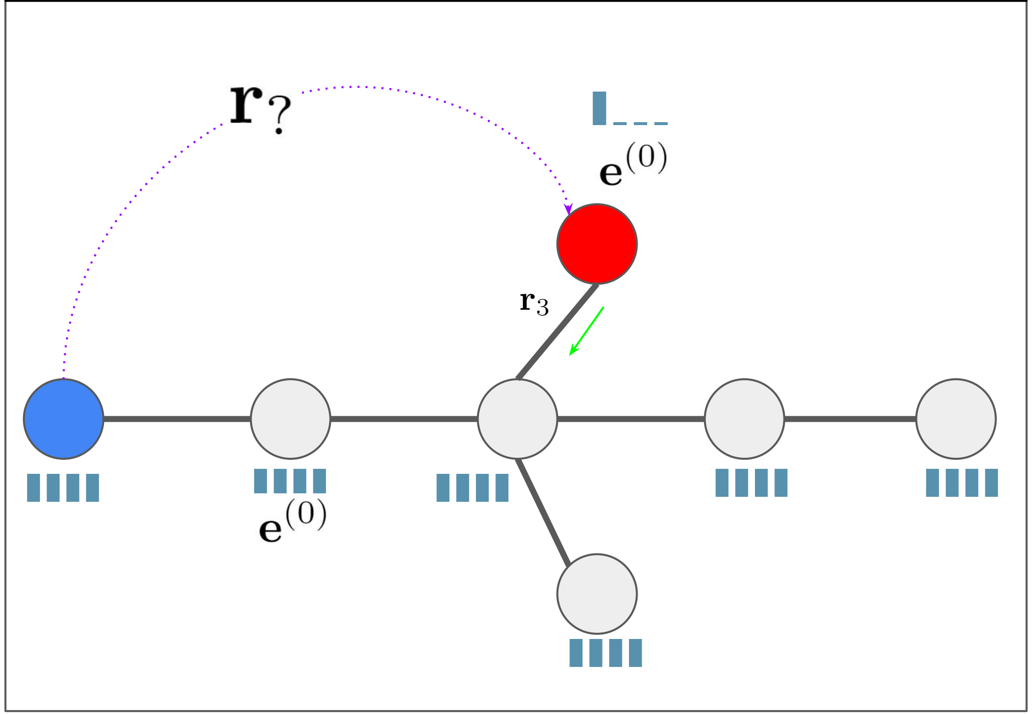


Illustration k=1 backward flow

→ forward flow

← backward flow

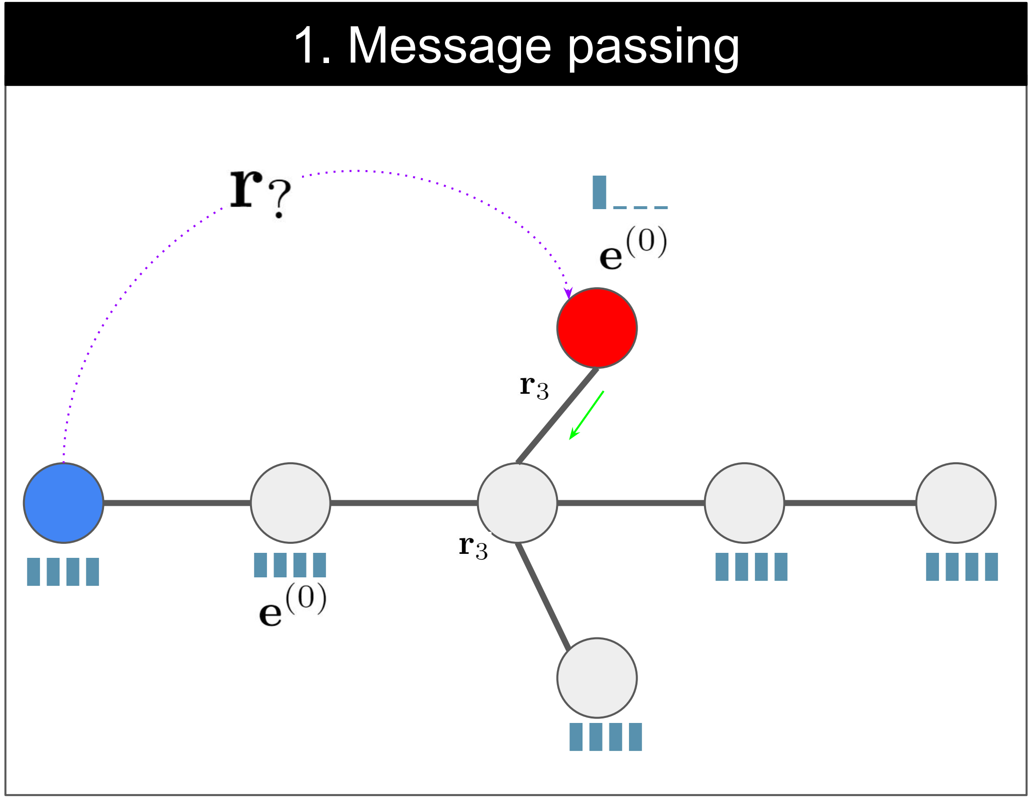
● source

● tail

● final

ϕ message

ψ aggregate



1. Message passing

Illustration k=2 backward flow

→ forward flow

← backward flow

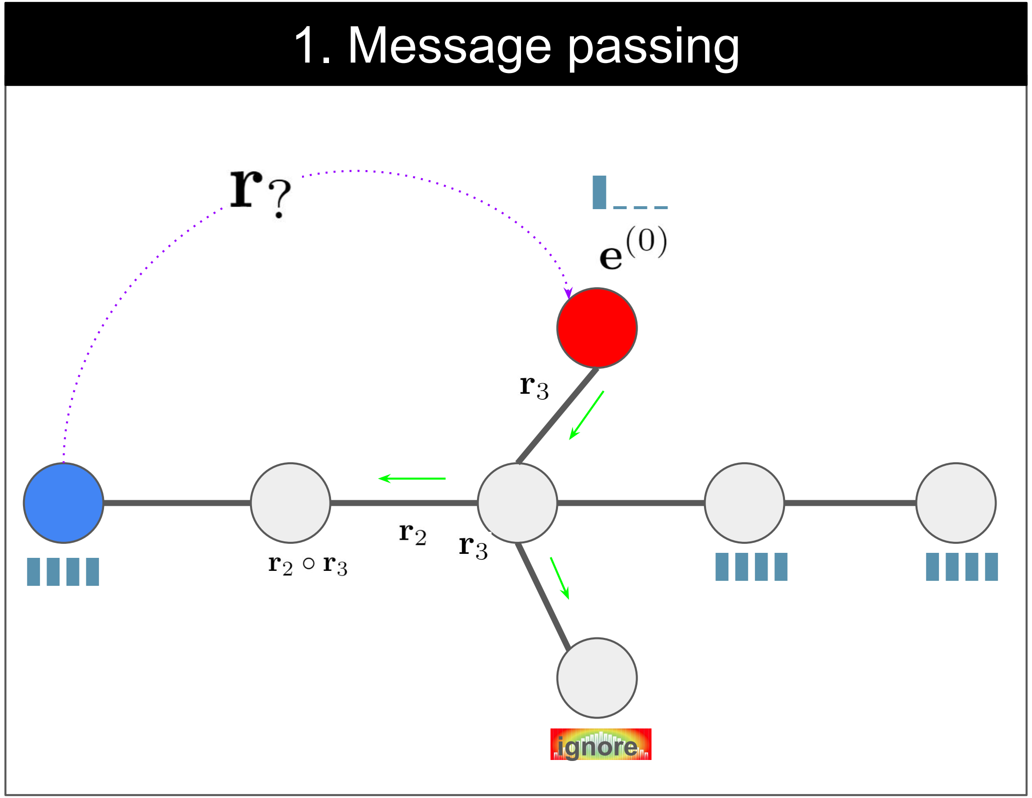
🔵 source

🔴 tail

● final

ϕ message

ψ aggregate



1. Message passing

Illustration k=3 backward flow

→ forward flow

← backward flow

🔵 source

🔴 tail

● final

ϕ message

ψ aggregate

1. Message passing

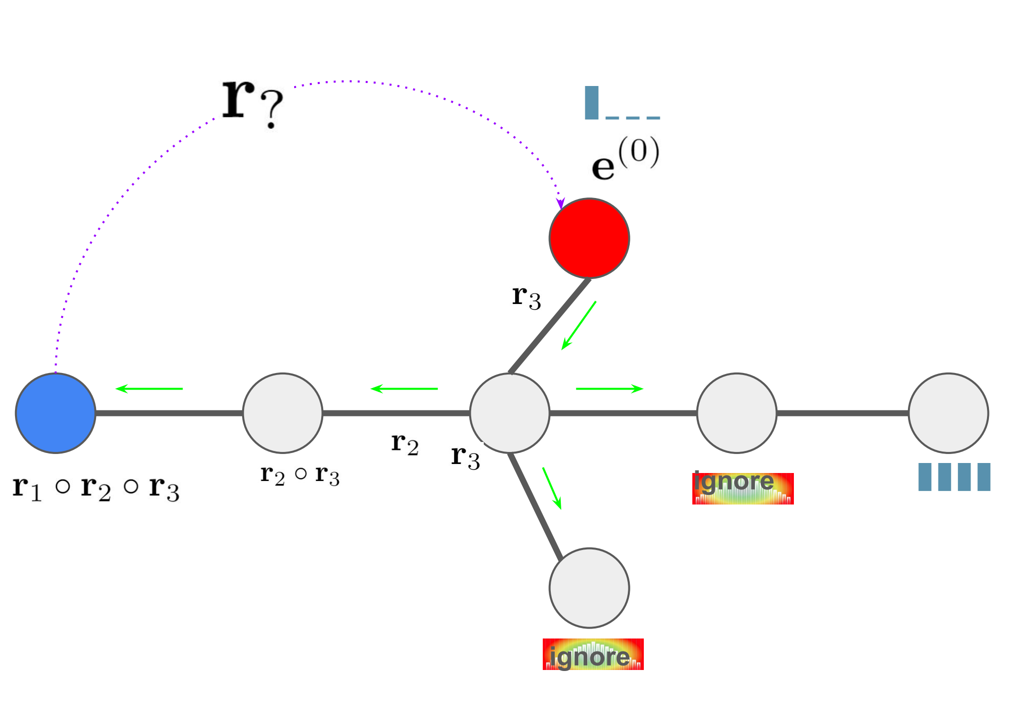


Illustration: Forward backward composition

→ forward flow

← backward flow

● source

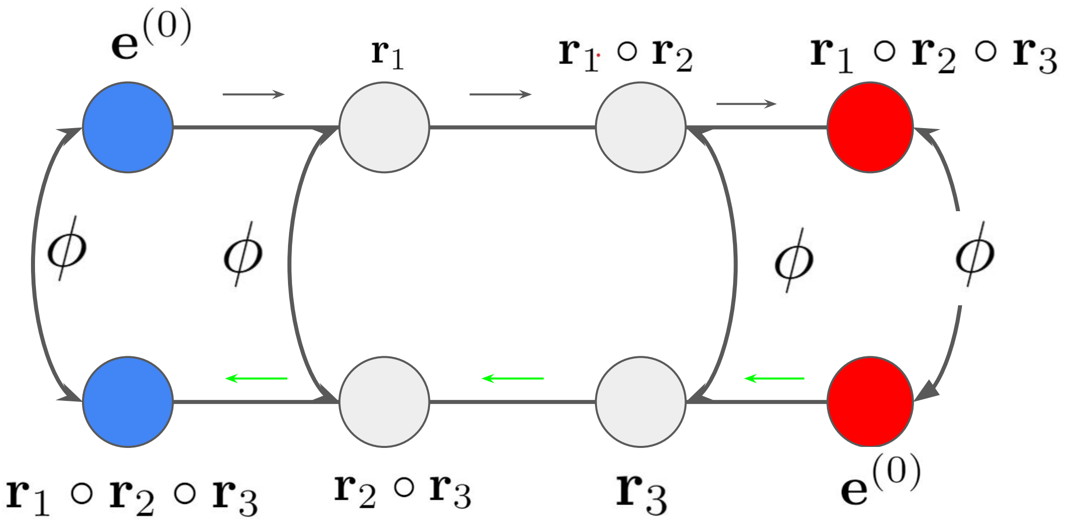
● tail

● final

ϕ message

ψ aggregate

2. F/B composition on simple path



$\mathbf{e}^{(0)}$
$\mathbf{r}_1$
$\mathbf{r}_1 \circ \mathbf{r}_2$
$\mathbf{r}_1 \circ \mathbf{r}_2 \circ \mathbf{r}_3$
ϕ
ϕ
ϕ
ϕ
$\mathbf{r}_1 \circ \mathbf{r}_2 \circ \mathbf{r}_3$
$\mathbf{r}_2 \circ \mathbf{r}_3$
$\mathbf{r}_3$
$\mathbf{e}^{(0)}$

Illustration: Forward backward composition

→ forward flow

← backward flow

● source

● tail

● final

ϕ message

ψ aggregate

2. F/B composition on simple path

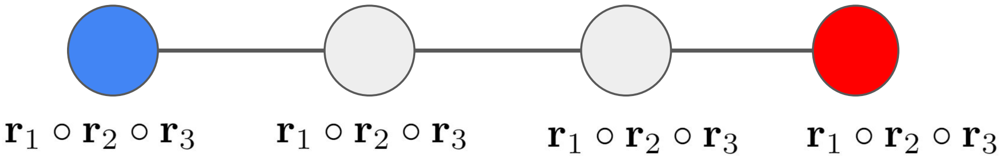

$\mathbf{r}_1 \circ \mathbf{r}_2 \circ \mathbf{r}_3$
$\mathbf{r}_1 \circ \mathbf{r}_2 \circ \mathbf{r}_3$
$\mathbf{r}_1 \circ \mathbf{r}_2 \circ \mathbf{r}_3$
$\mathbf{r}_1 \circ \mathbf{r}_2 \circ \mathbf{r}_3$

Illustration: Aggregation F/B node embeddings

→ forward flow

← backward flow

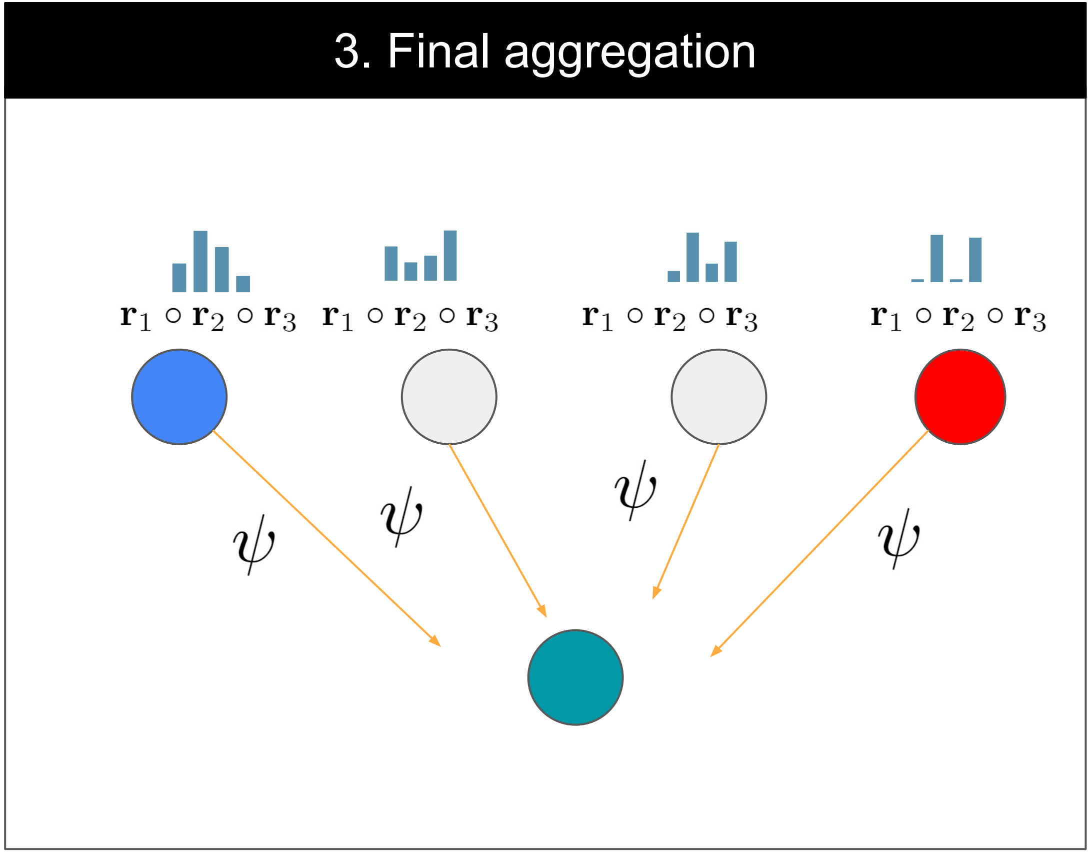
● source

● tail

● final

ϕ message

ψ aggregate



3. Final aggregation

$\mathbf{r}_1 \circ \mathbf{r}_2 \circ \mathbf{r}_3$
$\mathbf{r}_1 \circ \mathbf{r}_2 \circ \mathbf{r}_3$
$\mathbf{r}_1 \circ \mathbf{r}_2 \circ \mathbf{r}_3$
$\mathbf{r}_1 \circ \mathbf{r}_2 \circ \mathbf{r}_3$
ψ
ψ
ψ
ψ

Results

Link prediction (h,?,t): CLUTRR

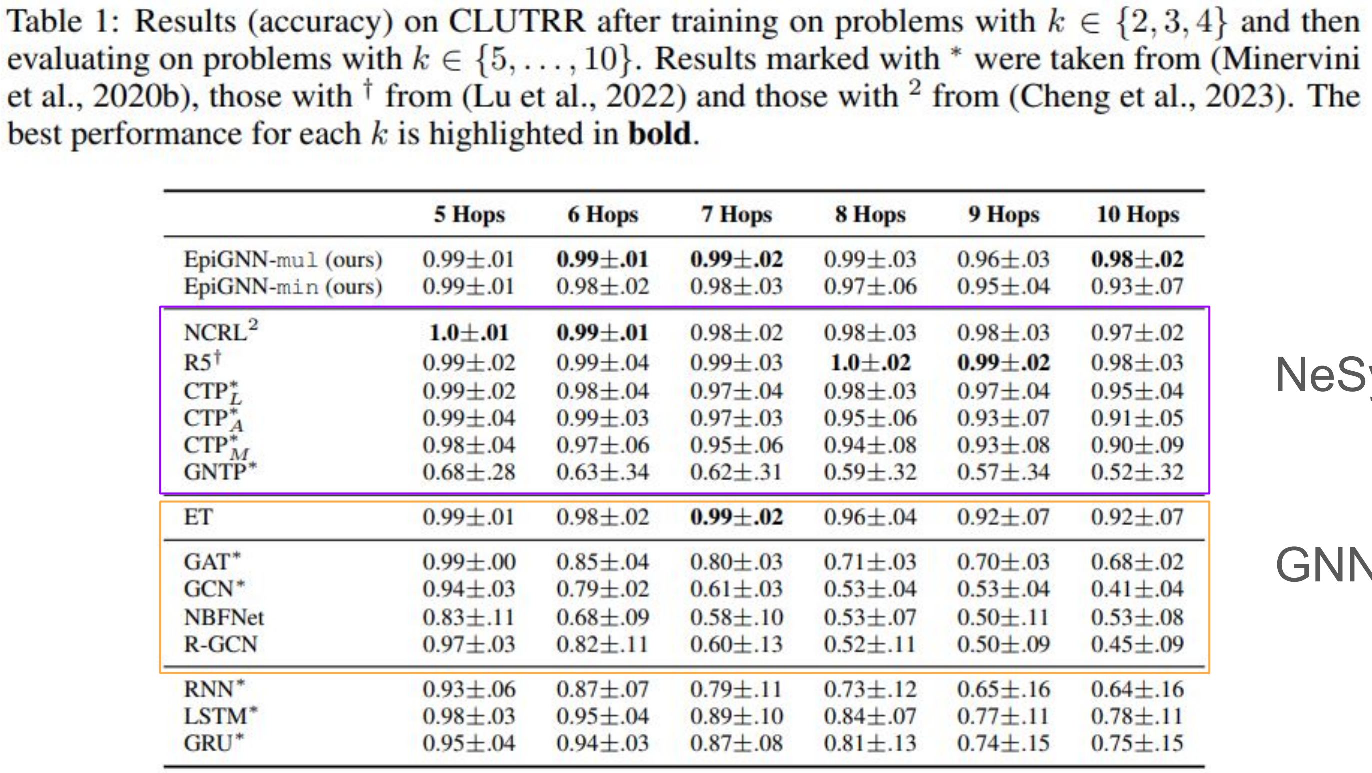


Table 1: Results (accuracy) on CLUTRR after training on problems with $k \in \{2, 3, 4\}$ and then evaluating on problems with $k \in \{5, \ldots, 10\}$. Results marked with * were taken from (Minervini et al., 2020b), those with † from (Lu et al., 2022) and those with ^{2} from (Cheng et al., 2023). The best performance for each k is highlighted in **bold**.

	5 Hops	**6 Hops**	**7 Hops**	**8 Hops**	**9 Hops**	**10 Hops**
EpiGNN-`mul` (ours)	0.99±.01	**0.99±.01**	**0.99±.02**	0.99±.03	0.96±.03	**0.98±.02**
EpiGNN-`min` (ours)	0.99±.01	0.98±.02	0.98±.03	0.97±.06	0.95±.04	0.93±.07
NCRL^2	**1.0±.01**	**0.99±.01**	0.98±.02	0.98±.03	0.98±.03	0.97±.02
R5†	0.99±.02	0.99±.04	0.99±.03	**1.0±.02**	**0.99±.02**	0.98±.03
CTP$^*_L$	0.99±.02	0.98±.04	0.97±.04	0.98±.03	0.97±.04	0.95±.04
CTP$^*_A$	0.99±.04	0.99±.03	0.97±.03	0.95±.06	0.93±.07	0.91±.05
CTP$^*_M$	0.98±.04	0.97±.06	0.95±.06	0.94±.08	0.93±.08	0.90±.09
GNTP*	0.68±.28	0.63±.34	0.62±.31	0.59±.32	0.57±.34	0.52±.32
ET	0.99±.01	0.98±.02	**0.99±.02**	0.96±.04	0.92±.07	0.92±.07
GAT*	0.99±.00	0.85±.04	0.80±.03	0.71±.03	0.70±.03	0.68±.02
GCN*	0.94±.03	0.79±.02	0.61±.03	0.53±.04	0.53±.04	0.41±.04
NBFNet	0.83±.11	0.68±.09	0.58±.10	0.53±.07	0.50±.11	0.53±.08
R-GCN	0.97±.03	0.82±.11	0.60±.13	0.52±.11	0.50±.09	0.45±.09
RNN*	0.93±.06	0.87±.07	0.79±.11	0.73±.12	0.65±.16	0.64±.16
LSTM*	0.98±.03	0.95±.04	0.89±.10	0.84±.07	0.77±.11	0.78±.11
GRU*	0.95±.04	0.94±.03	0.87±.08	0.81±.13	0.74±.15	0.75±.15

NeSy

GNNs

Link prediction (h,?,t): CLUTRR (harder)

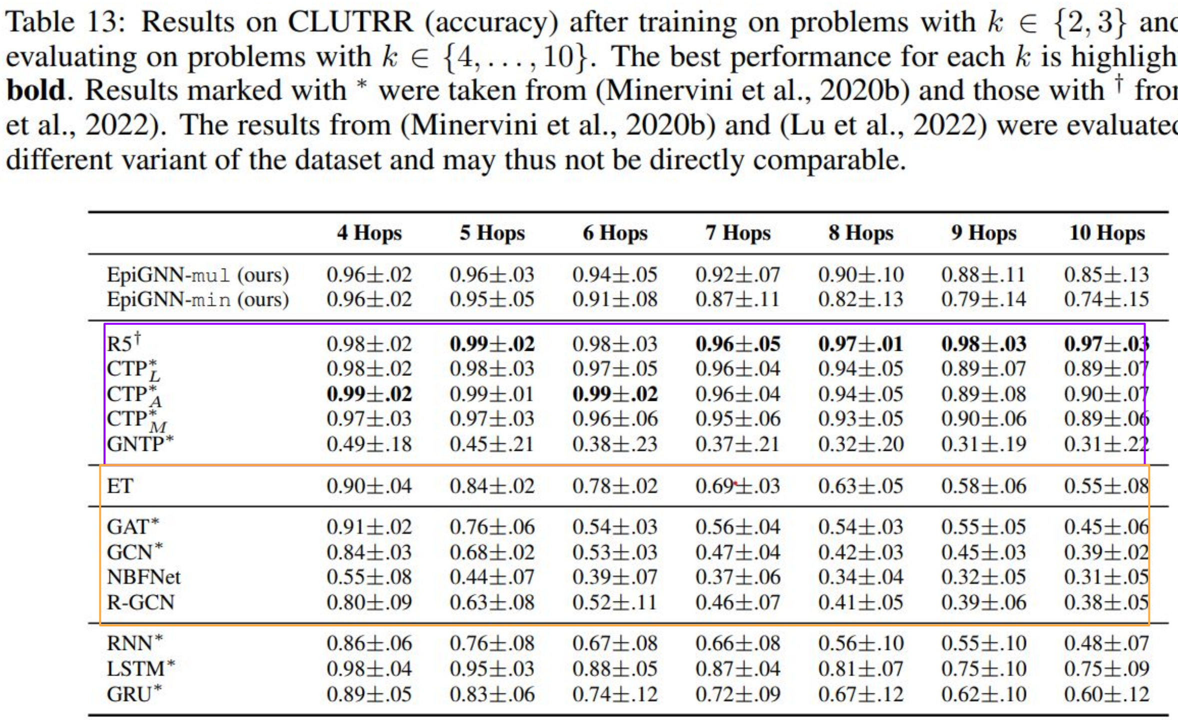


Table 13: Results on CLUTRR (accuracy) after training on problems with $k \in \{2, 3\}$ and then evaluating on problems with $k \in \{4, \ldots, 10\}$. The best performance for each k is highlighted in **bold**. Results marked with * were taken from (Minervini et al., 2020b) and those with † from (Lu et al., 2022). The results from (Minervini et al., 2020b) and (Lu et al., 2022) were evaluated on a different variant of the dataset and may thus not be directly comparable.

	4 Hops	5 Hops	6 Hops	7 Hops	8 Hops	9 Hops	10 Hops
EpiGNN-mul (ours)	0.96±.02	0.96±.03	0.94±.05	0.92±.07	0.90±.10	0.88±.11	0.85±.13
EpiGNN-min (ours)	0.96±.02	0.95±.05	0.91±.08	0.87±.11	0.82±.13	0.79±.14	0.74±.15
R5†	0.98±.02	**0.99±.02**	0.98±.03	**0.96±.05**	**0.97±.01**	**0.98±.03**	**0.97±.03**
CTP^*_L	0.98±.02	0.98±.03	0.97±.05	0.96±.04	0.94±.05	0.89±.07	0.89±.07
CTP^*_A	**0.99±.02**	0.99±.01	**0.99±.02**	0.96±.04	0.94±.05	0.89±.08	0.90±.07
CTP^*_M	0.97±.03	0.97±.03	0.96±.06	0.95±.06	0.93±.05	0.90±.06	0.89±.06
GNTP*	0.49±.18	0.45±.21	0.38±.23	0.37±.21	0.32±.20	0.31±.19	0.31±.22
ET	0.90±.04	0.84±.02	0.78±.02	0.69±.03	0.63±.05	0.58±.06	0.55±.08
GAT*	0.91±.02	0.76±.06	0.54±.03	0.56±.04	0.54±.03	0.55±.05	0.45±.06
GCN*	0.84±.03	0.68±.02	0.53±.03	0.47±.04	0.42±.03	0.45±.03	0.39±.02
NBFNet	0.55±.08	0.44±.07	0.39±.07	0.37±.06	0.34±.04	0.32±.05	0.31±.05
R-GCN	0.80±.09	0.63±.08	0.52±.11	0.46±.07	0.41±.05	0.39±.06	0.38±.05
RNN*	0.86±.06	0.76±.08	0.67±.08	0.66±.08	0.56±.10	0.55±.10	0.48±.07
LSTM*	0.98±.04	0.95±.03	0.88±.05	0.87±.04	0.81±.07	0.75±.10	0.75±.09
GRU*	0.89±.05	0.83±.06	0.74±.12	0.72±.09	0.67±.12	0.62±.10	0.60±.12

NeSy

GNNs

Link prediction (h,?,t): GraphLOG

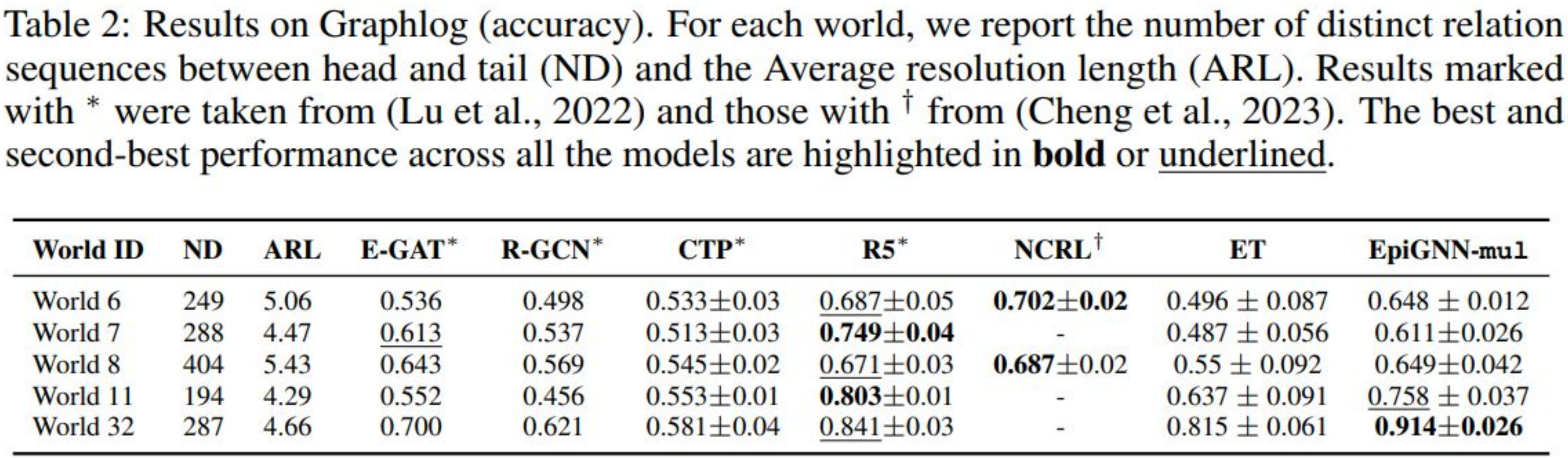


Table 2: Results on Graphlog (accuracy). For each world, we report the number of distinct relation sequences between head and tail (ND) and the Average resolution length (ARL). Results marked with * were taken from (Lu et al., 2022) and those with † from (Cheng et al., 2023). The best and second-best performance across all the models are highlighted in **bold** or underlined.

World ID	ND	ARL	E-GAT*	R-GCN*	CTP*	R5*	NCRL†	ET	EpiGNN-mul
World 6	249	5.06	0.536	0.498	0.533±0.03	0.687±0.05	**0.702±0.02**	0.496 ± 0.087	0.648 ± 0.012
World 7	288	4.47	0.613	0.537	0.513±0.03	**0.749±0.04**	-	0.487 ± 0.056	0.611±0.026
World 8	404	5.43	0.643	0.569	0.545±0.02	0.671±0.03	**0.687**±0.02	0.55 ± 0.092	0.649±0.042
World 11	194	4.29	0.552	0.456	0.553±0.01	**0.803**±0.01	-	0.637 ± 0.091	0.758 ± 0.037
World 32	287	4.66	0.700	0.621	0.581±0.04	0.841±0.03	-	0.815 ± 0.061	**0.914±0.026**

Link prediction (h,?,t): STaR

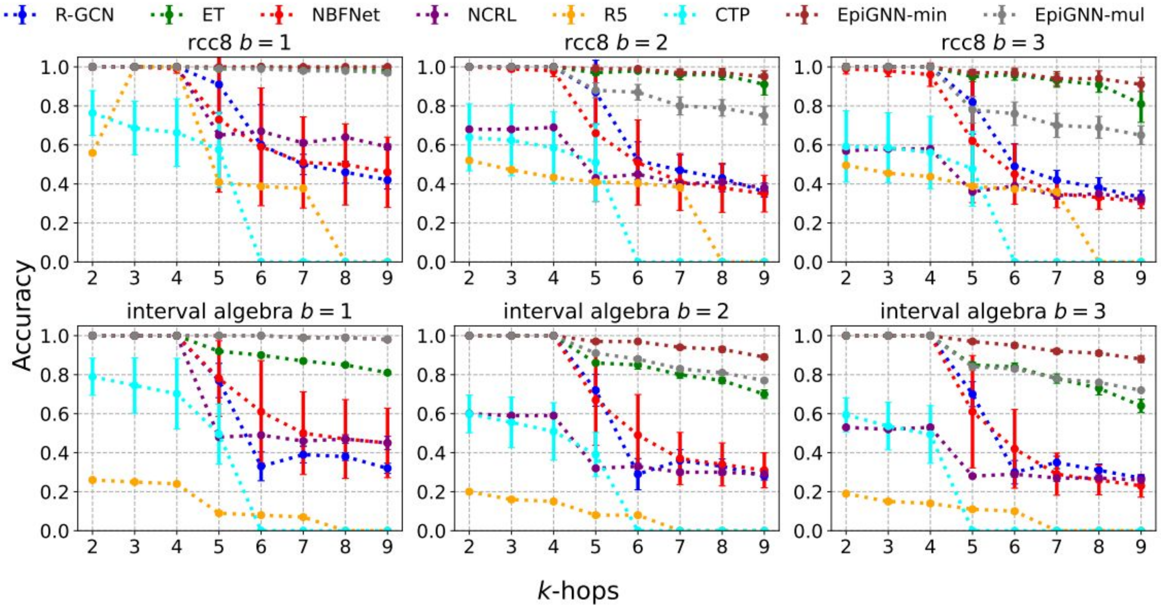
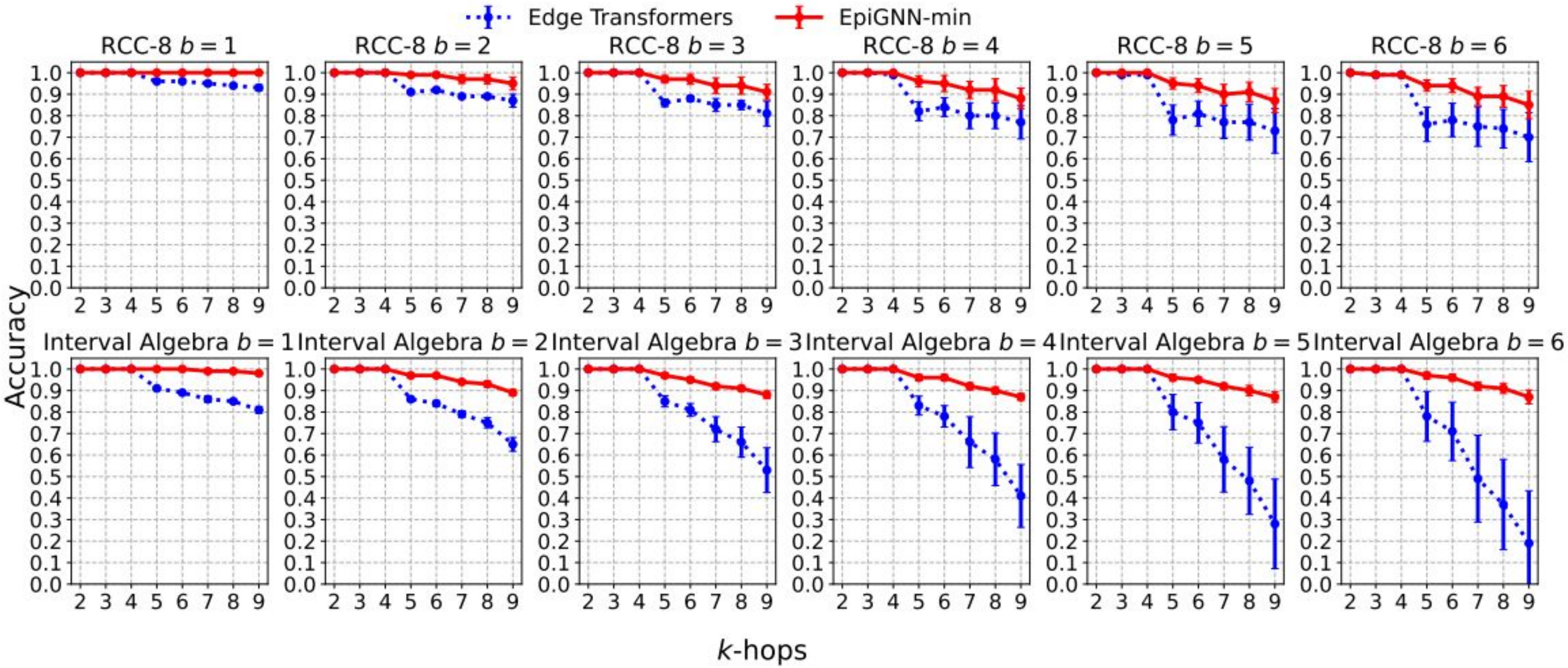
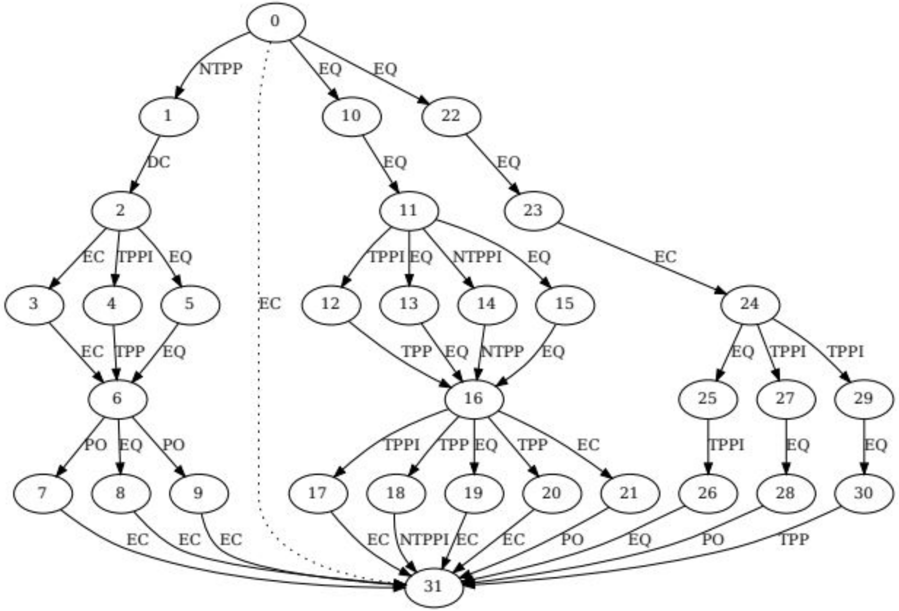


Figure 4: RCC-8 and Interval Algebra benchmark results (accuracy). R5 and CTP results for 5+ hops were set to zero since the model took longer than 30 minutes for inference. Models are trained on graphs with $b \in \{1, 2, 3\}$ paths of length $k \in \{2, 3, 4\}$. The best model for all cases is EpiGNN-min.

Link prediction (h,?,t): STaR (Harder Expansion)



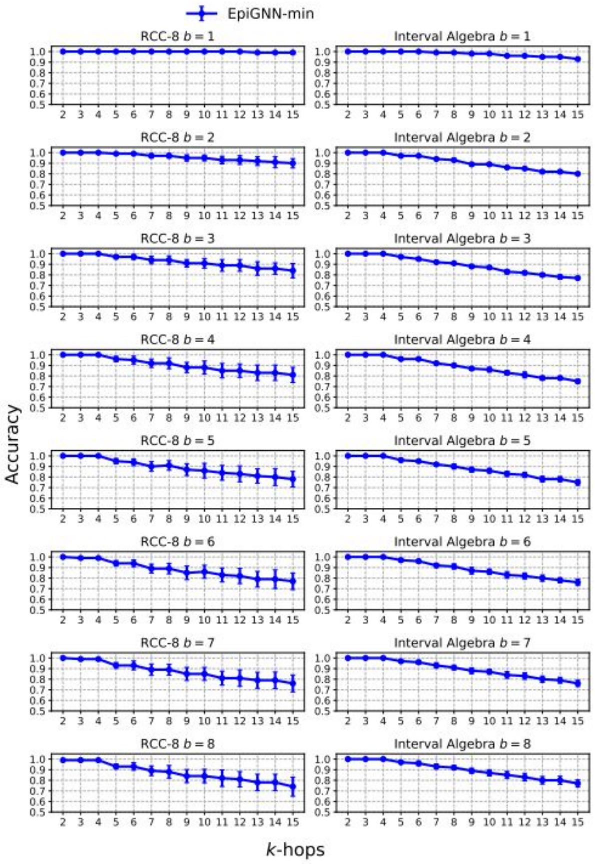
For context, the test graphs get quite large



: 1

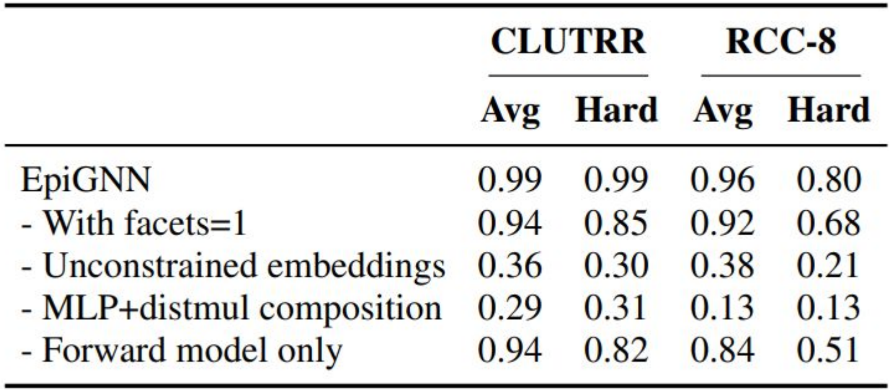
(h) $k = 6, b = 3$

Link prediction (h,?,t): STaR (Even Harder Expansion)

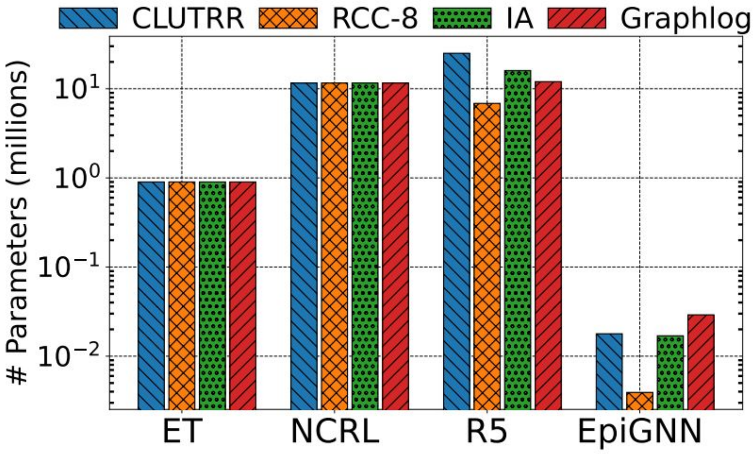


The inductive bias for disjunctive reasoning gets really delineated here as the EpiGNN is fairly steady, performance-wise, not dropping below ~0.75 on the hardest settings.

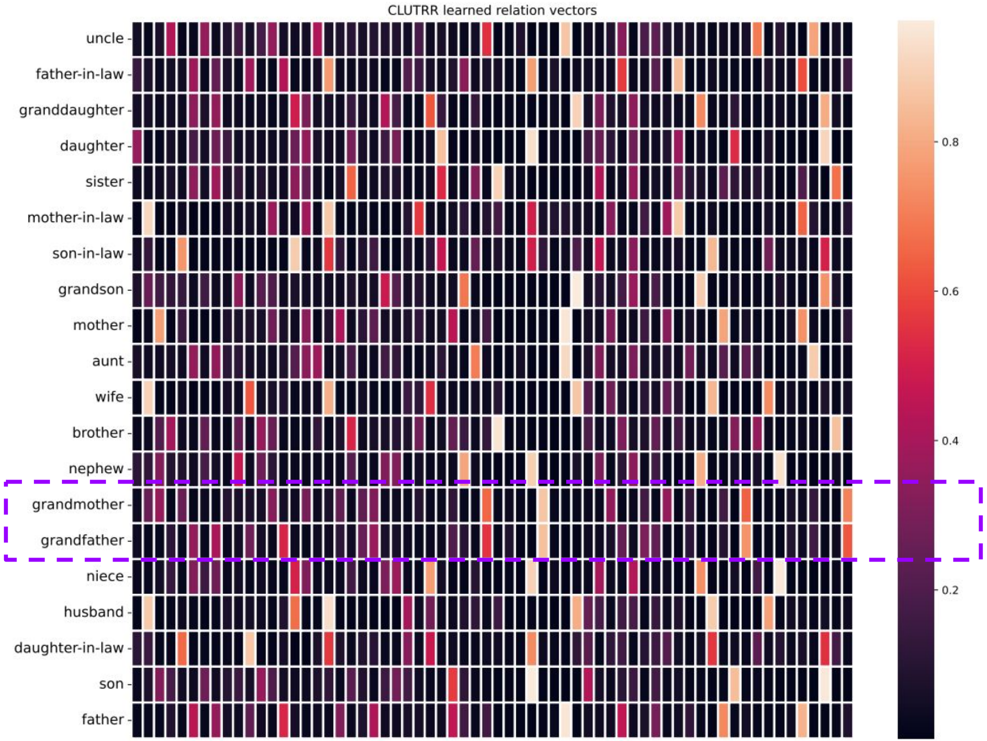
Ablations highlight the necessity of all the model’s architectural propositions and EpiGNN is 100x parameter-efficient wrt. baselines

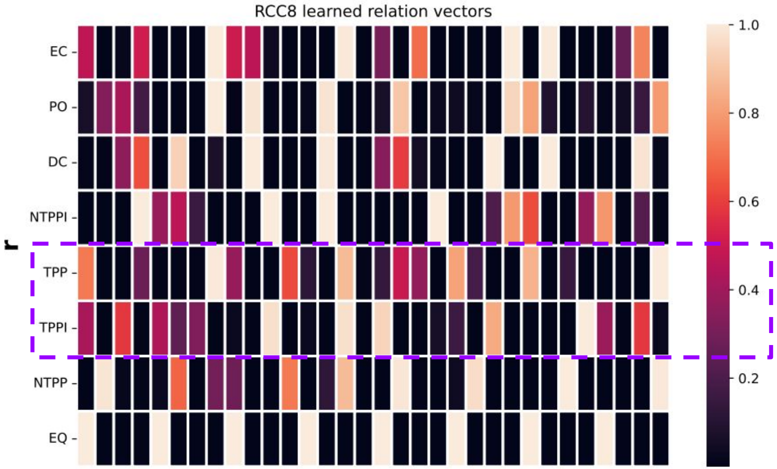


	CLUTRR		RCC-8	
	Avg	**Hard**	**Avg**	**Hard**
EpiGNN	0.99	0.99	0.96	0.80
- With facets=1	0.94	0.85	0.92	0.68
- Unconstrained embeddings	0.36	0.30	0.38	0.21
- MLP+distmul composition	0.29	0.31	0.13	0.13
- Forward model only	0.94	0.82	0.84	0.51

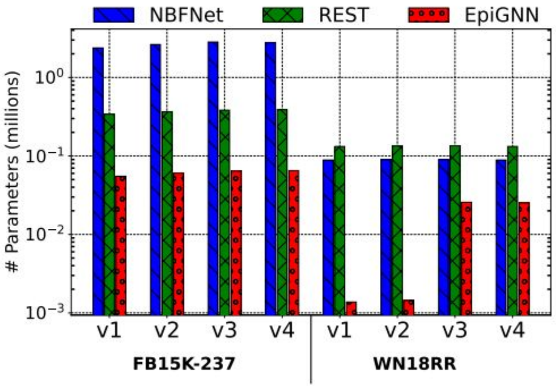


Learned relation embeddings are semantically meaningful





Entity prediction results (h,r,?): on knowledge graphs



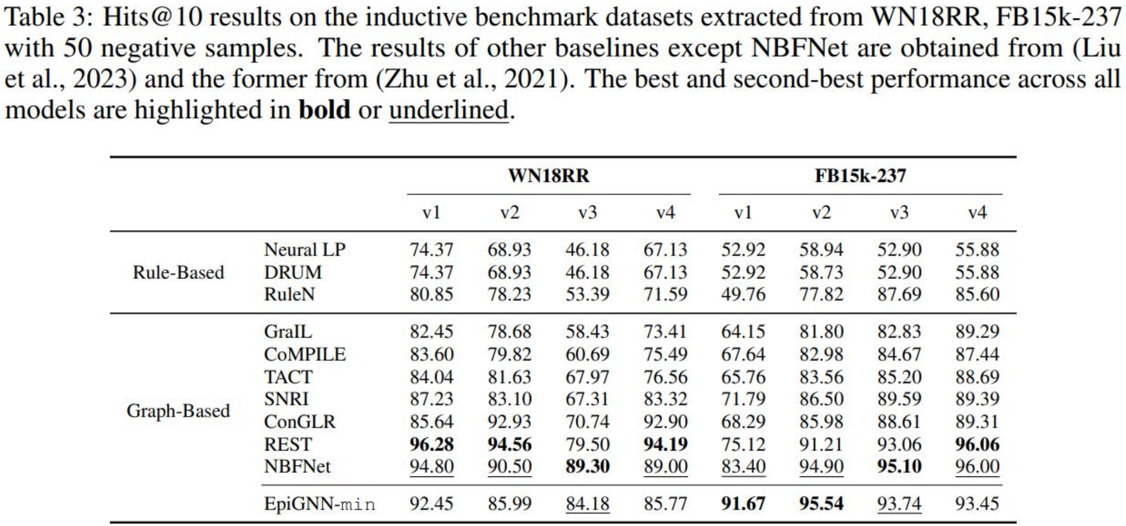


Table 3: Hits@10 results on the inductive benchmark datasets extracted from WN18RR, FB15k-237 with 50 negative samples. The results of other baselines except NBFNet are obtained from (Liu et al., 2023) and the former from (Zhu et al., 2021). The best and second-best performance across all models are highlighted in **bold** or underlined.

		WN18RR				FB15k-237			
		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
	DRUM	74.37	68.93	46.18	67.13	52.92	58.73	52.90	55.88
	RuleN	80.85	78.23	53.39	71.59	49.76	77.82	87.69	85.60
Graph-Based	GraIL	82.45	78.68	58.43	73.41	64.15	81.80	82.83	89.29
	CoMPILE	83.60	79.82	60.69	75.49	67.64	82.98	84.67	87.44
	TACT	84.04	81.63	67.97	76.56	65.76	83.56	85.20	88.69
	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
	REST	**96.28**	**94.56**	79.50	**94.19**	75.12	91.21	93.06	**96.06**
	NBFNet	94.80	90.50	**89.30**	89.00	83.40	94.90	**95.10**	96.00
	EpiGNN-min	92.45	85.99	84.18	85.77	**91.67**	**95.54**	93.74	93.45

1. Confirms that EpiGNN still preserves most of the NBFNet performance after modifications but also that a systematic generalisation bias hurts inductive KG performance.
2. Highlights the trade-off in problem domains: avoiding statistical biases in KGC is not a good idea wrt. Performance.
3. EpiGNN is also parameter-efficient wrt. NBFNet and REST (Liu et. al. 2023)

Outlook

Takeaways

1. Systematic reasoning enables generalization beyond training data. Could unlock high quality low-data learning.
2. NNs are bad at it because they lack the inductive biases that align their architecture with an algorithm that solves it.
3. For multi-path disjunctive reasoning (generalising single-path), we can align a relational GNN with the algebraic closure algorithm.
4. Statistical biases need to be avoided for systematic reasoning but might be necessary for high performance for other domains.

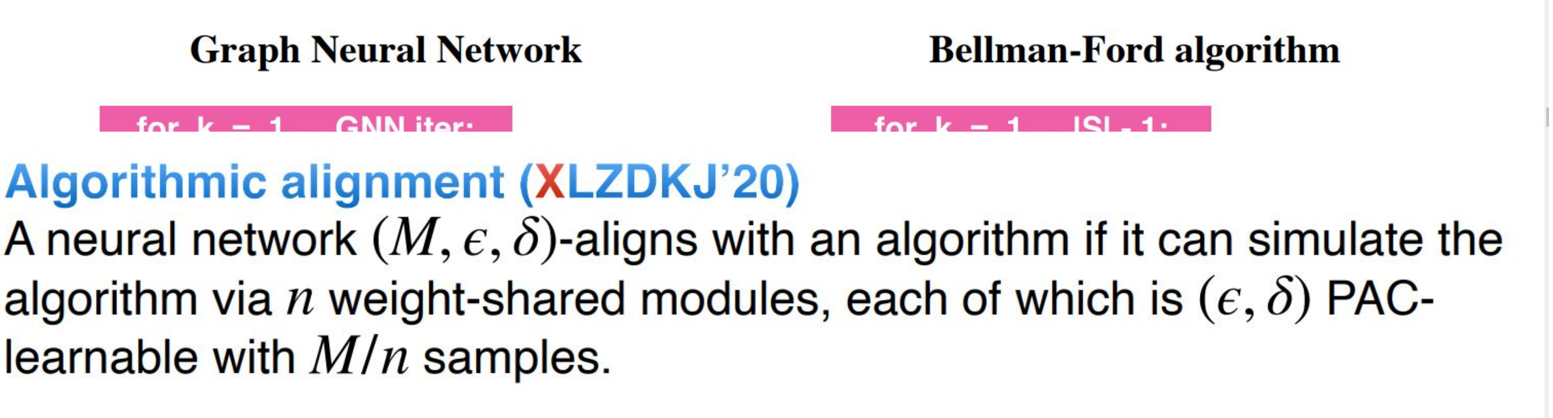
What I’m currently working on

1. Story graph reasoning: applications to event-based info retrieval
    a. Getting a graph from raw text using NLP techniques
    b. Embellishing this graph with (noisy) additional information
    c. Reasoning engine denoises the noisy graphs
2. Making the STaR benchmark more real-world like
    a. Family relations + locations?
    b. Spatial positions with constraints

Any Questions?

Backup Slides

Crux: Algorithmically aligning the architecture with the task aids generalization (Xu et. al. 2020)



Graph Neural Network

for k = 1 ... GNN iter:

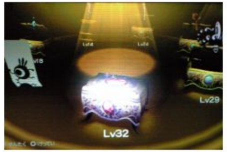
Bellman-Ford algorithm

for k = 1 ... |S| - 1:

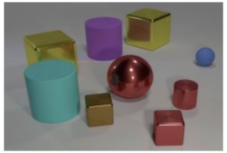
Algorithmic alignment (XLZDKJ'20)

A neural network (M, ϵ, δ)-aligns with an algorithm if it can simulate the algorithm via n weight-shared modules, each of which is (ϵ, δ) PAC-learnable with M/n samples.

MLPs have to learn for-loops that GNNs don't so tasks unified by dynamic programming are more sample efficiently learned by GNNs

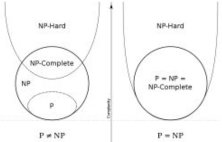


Summary statistics
What is the maximum value difference among treasures?

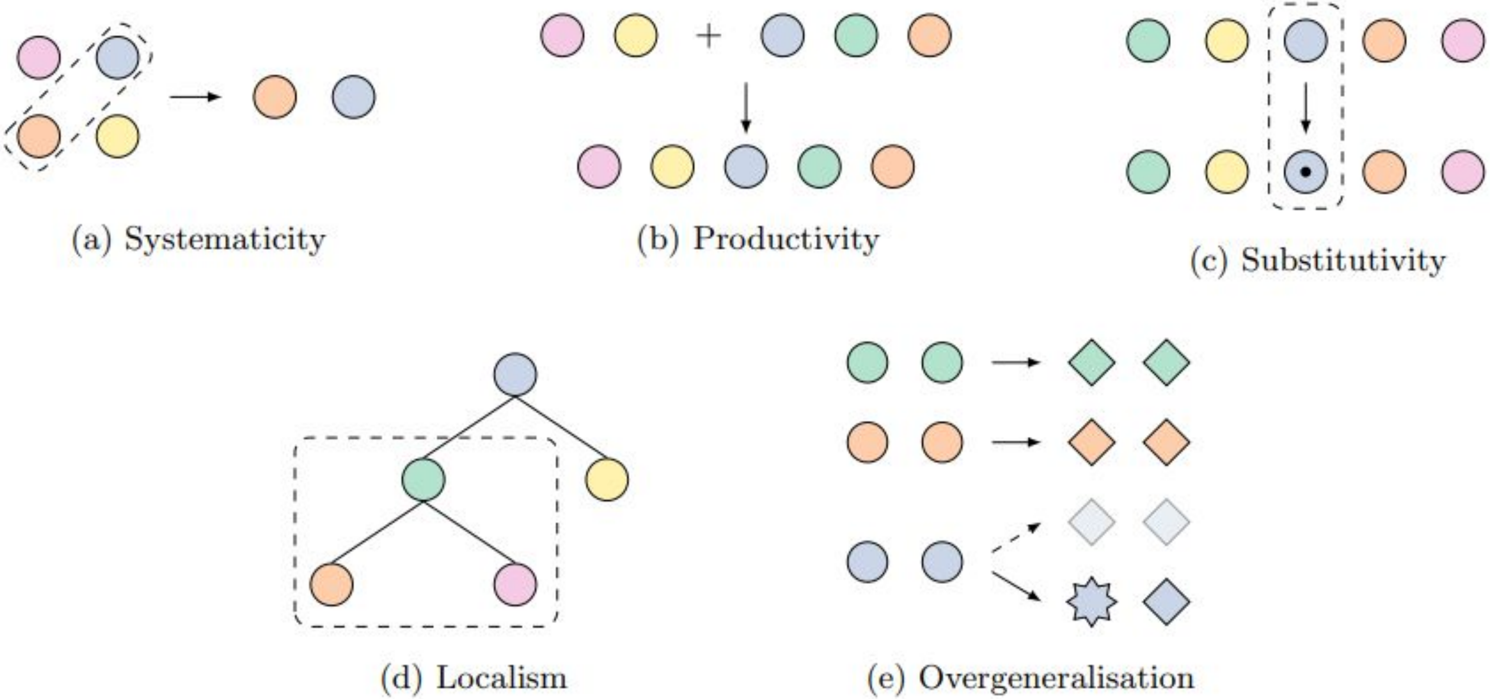


Relational argmax
What are the colors of the furthest pair of objects?

Dynamic programming
What is the cost to defeat monster X by following the optimal path?



NP-hard problem
Subset sum: Is there a subset that sums to 0?



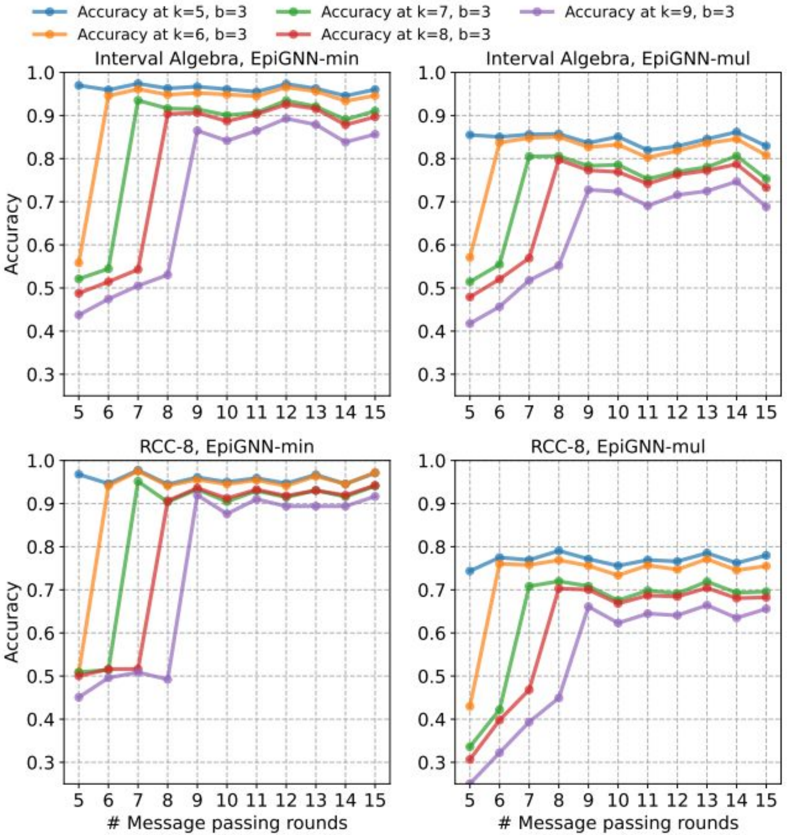
(a) Systematicity

(b) Productivity

(c) Substitutivity

(d) Localism

(e) Overgeneralisation


Accuracy at k=5, b=3
Accuracy at k=6, b=3
Accuracy at k=7, b=3
Accuracy at k=8, b=3
Accuracy at k=9, b=3
Interval Algebra, EpiGNN-min
Interval Algebra, EpiGNN-mul
RCC-8, EpiGNN-min
RCC-8, EpiGNN-mul
Accuracy
Accuracy
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
5
6
7
8
9
10
11
12
13
14
15
# Message passing rounds
# Message passing rounds