Inductive Relation Prediction by Subgraph Reasoning

Komal K. Teru, Etienne Denis, William L. Hamilton

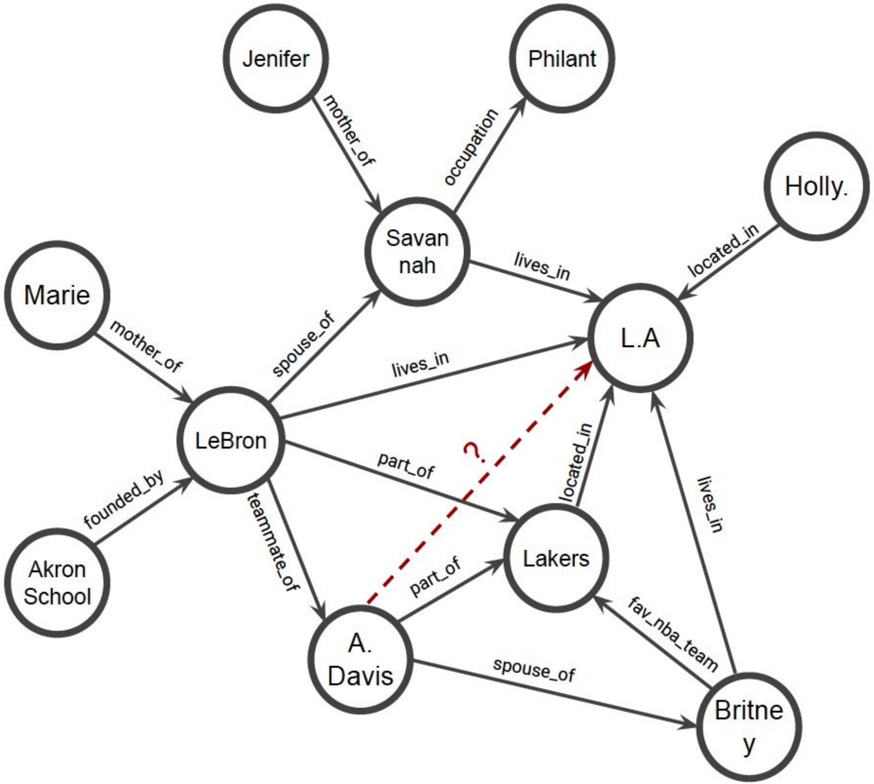
McGill University and Mila – AI Institute of Quebec

Relation prediction in Knowledge graphs

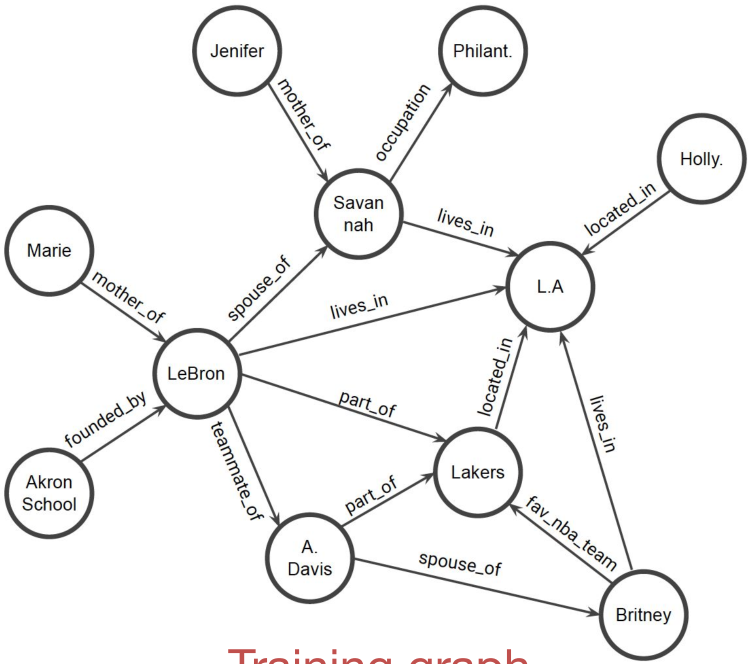
Automatically expand and complete existing knowledge bases.

Needs relational reasoning to make inference.

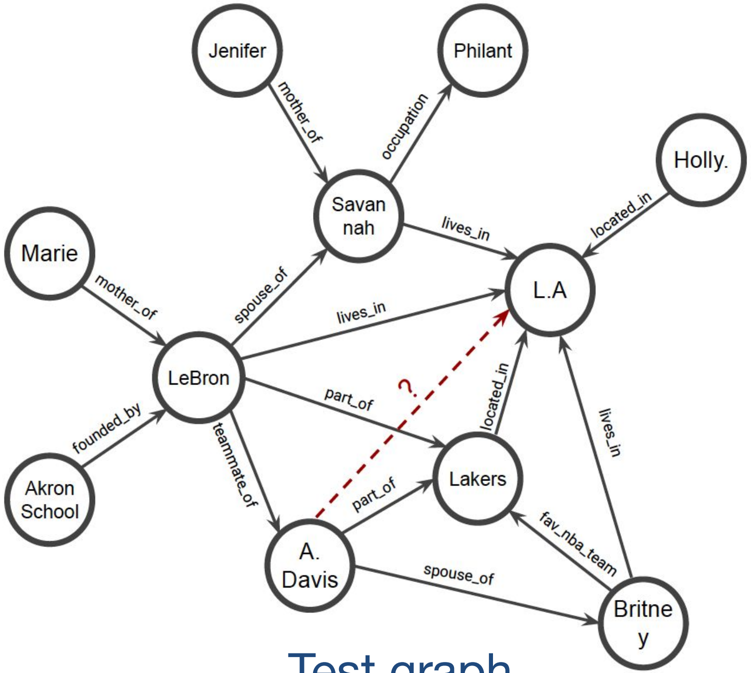
Applications in e-commerce, medicine, materials science…



Transductive relation prediction



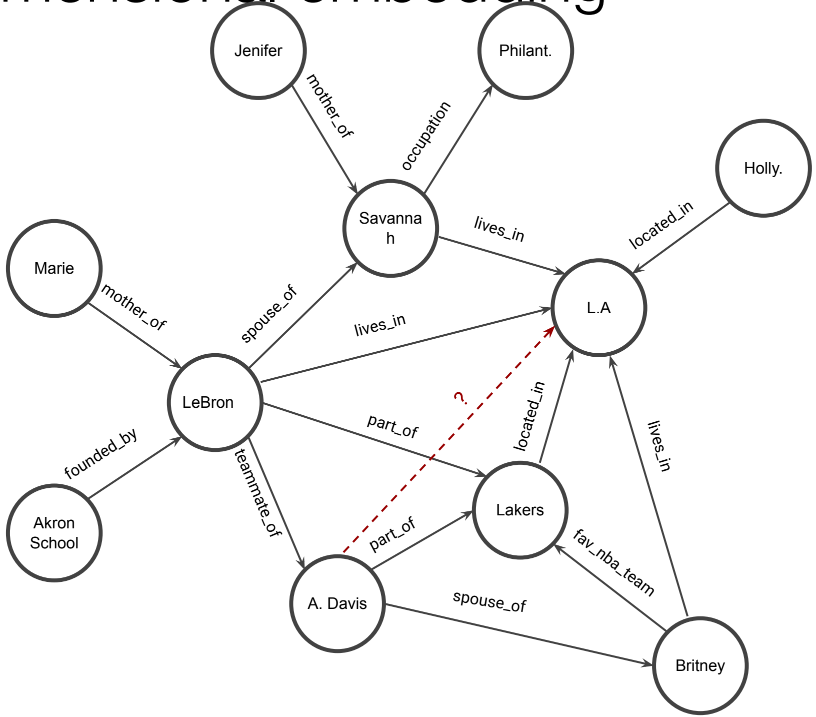
Training graph



Test graph

Embeddings-based methods

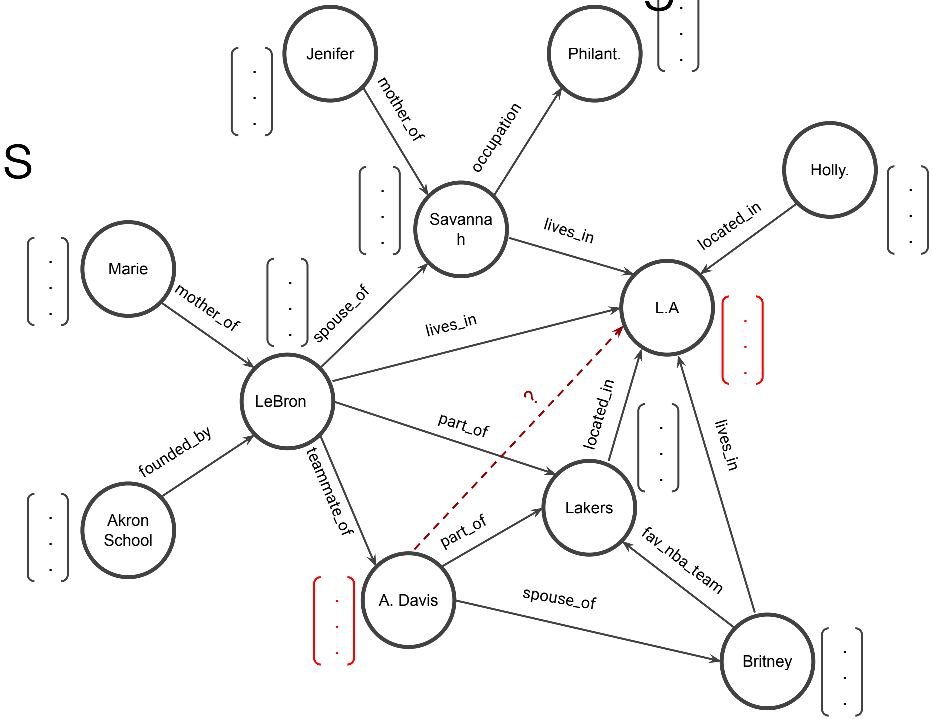
- Encode each node to a low-dimensional embedding



Embeddings-based methods

- Encode each node to a low-dimensional embedding

- Use the derived embeddings to make predictions (TransE, RotatE, etc.)

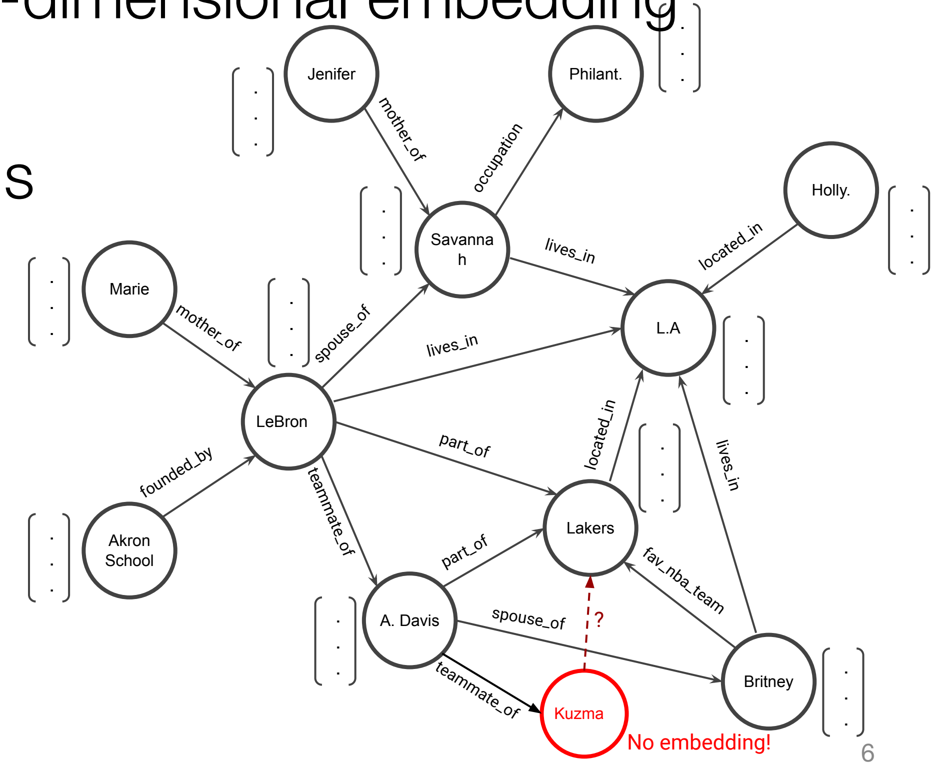


Embeddings-based methods

- Encode each node to a low-dimensional embedding

- Use the derived embeddings to make predictions (TransE, RotatE, etc.)

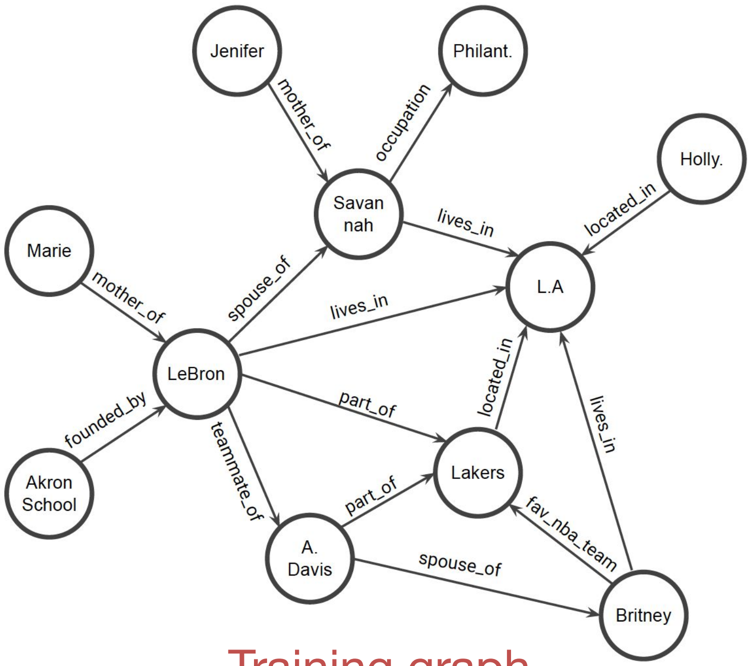
- Can’t make predictions on new nodes.



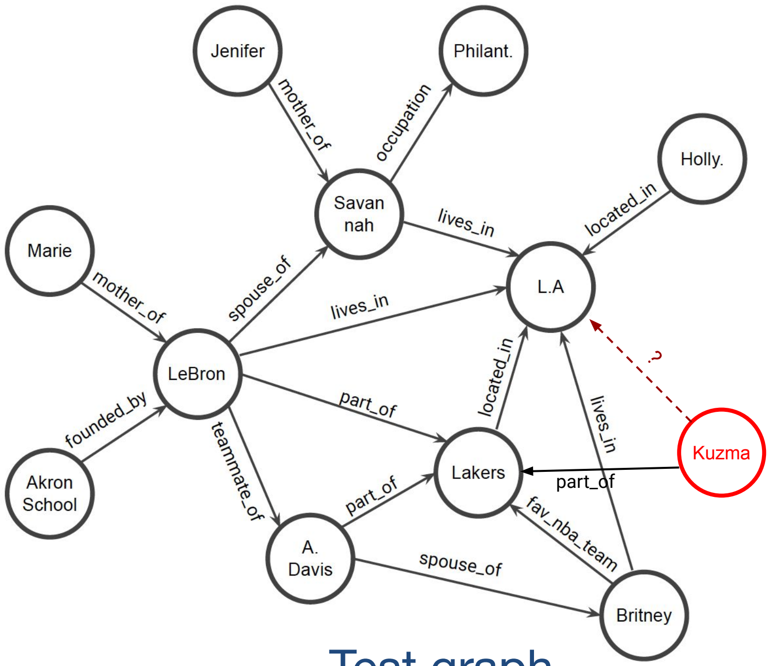
Limitations of transductive relation prediction

- Problematic for production systems
  - Need to re-train to deal with new nodes (e.g., entities, products)
  - Predictions can become stale.
- Too many parameters
  - Most transductive approaches have O(|V|) space complexity.
- Biases model development
  - Focus on “embedding”-based methodologies.
  - Static and unrepresentative benchmarks

Inductive learning: evolving data

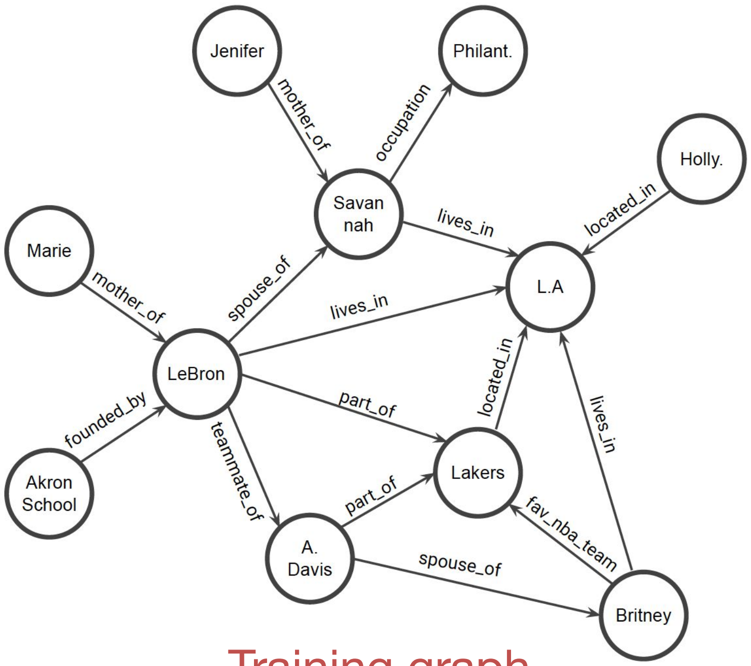


Training graph

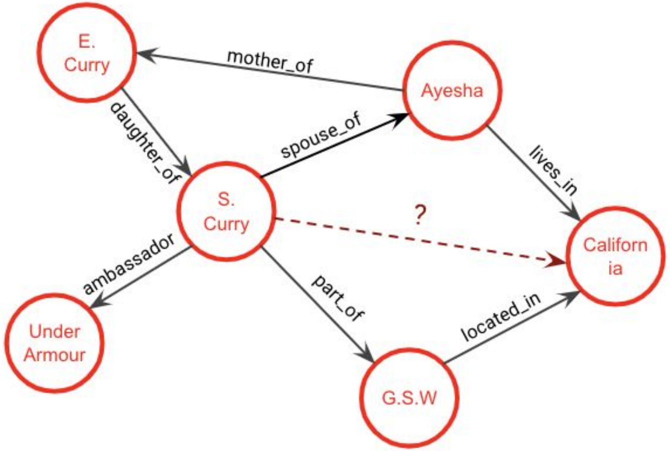


Test graph

Inductive learning: new graphs



Training graph



Test graph

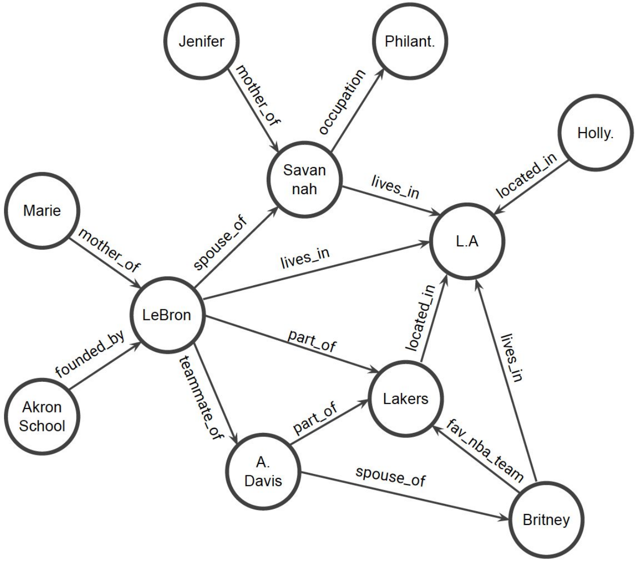
GraIL: Inductive learning using GNNs

- A novel approach to learn entity-independent relational semantics (rules)

- SOTA performance on inductive benchmarks

- Extremely parameter efficient

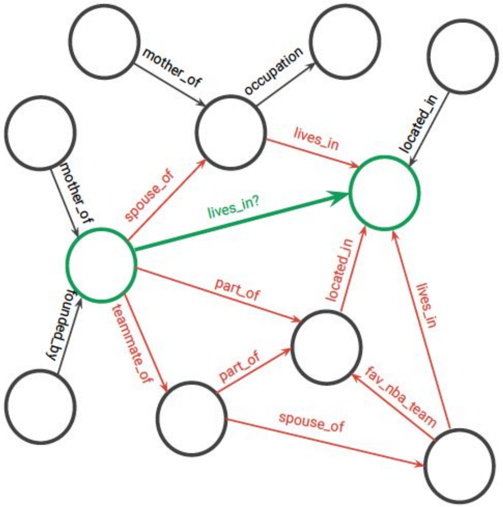
$$(X, \texttt{lives\_in}, Y) \leftarrow$$
$$\exists Z.(X, \texttt{spouse\_of}, Z) \wedge (Z, \texttt{lives\_in}, Y)$$



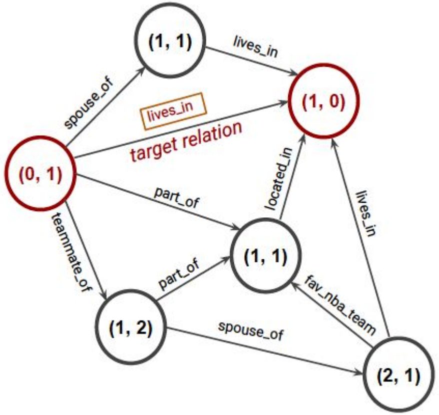
GraIL: Inductive learning using GNNs

- Idea 1: Apply graph neural networks (GNNs) on the subgraphs surrounding candidate edge.
- Idea 2: Avoid explicit rule induction.
- Idea 3: Ensure model is expressive enough to capture logical rules.

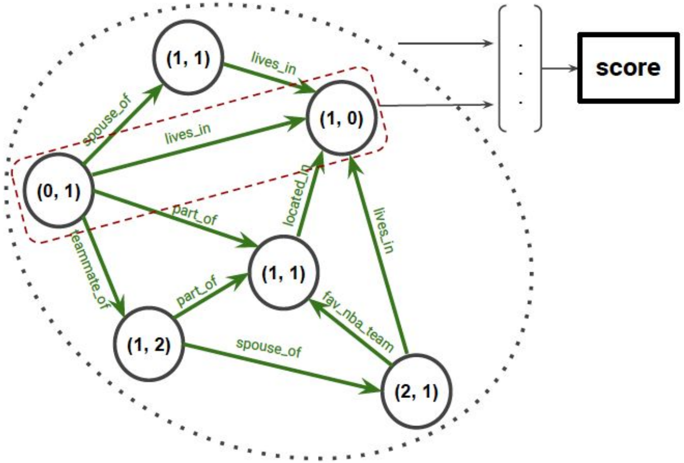
GraIL: Relation prediction via subgraph reasoning



1. Extract subgraph around candidate edge



2. Assign structural labels to nodes



3. Run GNN on the extracted subgraph

GNN architecture

Neural message-passing approach

$$\mathbf{a}_t^k = \mathrm{AGGREGATE}^k \left(\left\{ \mathbf{h}_s^{k-1} : s \in \mathcal{N}(t) \right\}, \mathbf{h}_t^{k-1} \right);$$

$$\mathbf{h}_t^k = \mathrm{COMBINE}^k \left(\mathbf{h}_t^{k-1}, \mathbf{a}_t^k \right)$$

GNN architecture

Neural message-passing approach

$$\mathbf{a}_t^k = \text{AGGREGATE}^k\left(\left\{\mathbf{h}_s^{k-1} : s \in \mathcal{N}(t)\right\}, \mathbf{h}_t^{k-1}\right);$$

$$\mathbf{h}_t^k = \text{COMBINE}^k\left(\mathbf{h}_t^{k-1}, \mathbf{a}_t^k\right)$$

Learn a relation-specific transformation matrix

$$\mathbf{a}_t^k = \sum_{r=1}^{R} \sum_{s \in \mathcal{N}_r(t)} \alpha_{r r_t s t}^k \mathbf{W}_r^k \mathbf{h}_s^{k-1}$$

Use attention to weigh information coming from different neighbors

Separately aggregate across different types of relations

GNN architecture

Neural message-passing approach

$$\mathbf{a}_t^k = \text{AGGREGATE}^k\left(\left\{\mathbf{h}_s^{k-1} : s \in \mathcal{N}(t)\right\}, \mathbf{h}_t^{k-1}\right),$$

$$\mathbf{h}_t^k = \text{COMBINE}^k\left(\mathbf{h}_t^{k-1}, \mathbf{a}_t^k\right)$$

Information aggregated from the neighborhood

$$\mathbf{h}_t^k = \text{ReLU}\left(\mathbf{W}_{self}^k \mathbf{h}_t^{k-1} + \mathbf{a}_t^k\right)$$

Information from the nodes embedding at the previous layer

GraIL can learn logical rules

Theorem (Informally): GraIL can learn any logical rule of the form:

$$r_t(X,Y) \leftarrow \exists Z_1, ..., Z_k . r_1(X,Z_1) \wedge r_2(Z_1,Z_2) \wedge ... \wedge r_k(Z_{k-1},Y)$$

Example of such a rule:

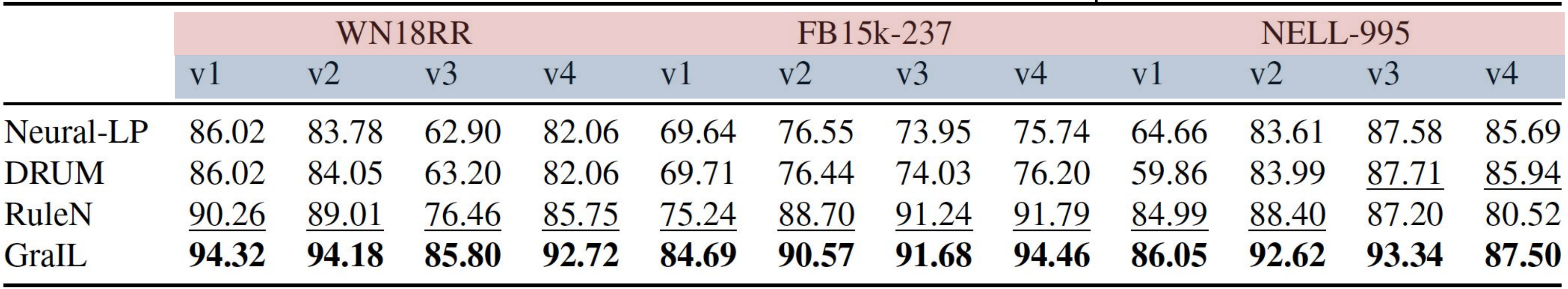
$$(X, \texttt{lives\_in}, Y) \leftarrow \exists Z.(X, \texttt{spouse\_of}, Z) \wedge (Z, \texttt{lives\_in}, Y)$$

These “path-based” rules are the foundation of most state-of-the-art rule induction systems.

State-of-the-art inductive performance

- Constructed inductive versions of three standard benchmarks.
- Sampled mutually exclusive subgraphs of varying sizes
- Tested four inductive datasets per each benchmark.

Table: AUC-PR results on inductive relation prediction

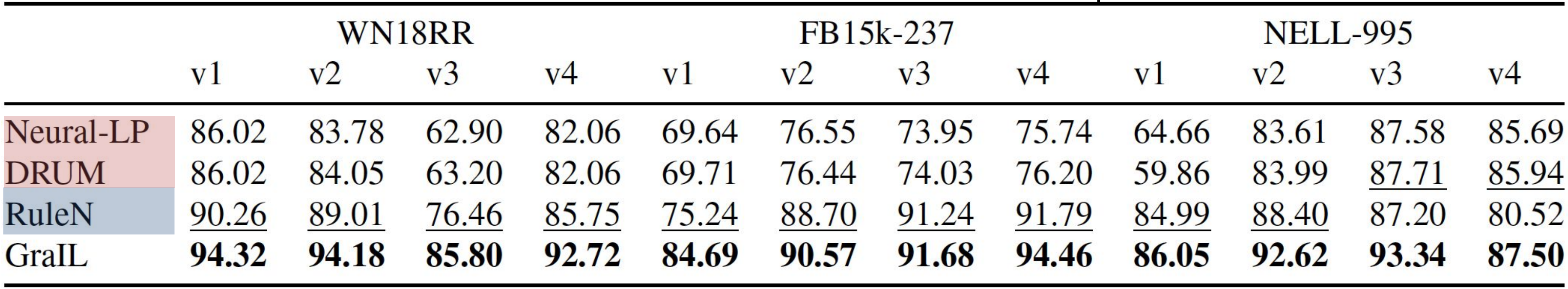


	WN18RR				FB15k-237				NELL-995			
	v1	v2	v3	v4	v1	v2	v3	v4	v1	v2	v3	v4
Neural-LP	86.02	83.78	62.90	82.06	69.64	76.55	73.95	75.74	64.66	83.61	87.58	85.69
DRUM	86.02	84.05	63.20	82.06	69.71	76.44	74.03	76.20	59.86	83.99	87.71	85.94
RuleN	90.26	89.01	76.46	85.75	75.24	88.70	91.24	91.79	84.99	88.40	87.20	80.52
GraIL	**94.32**	**94.18**	**85.80**	**92.72**	**84.69**	**90.57**	**91.68**	**94.46**	**86.05**	**92.62**	**93.34**	**87.50**

State-of-the-art inductive performance

- Compared against state-of-the-art neural rule induction methods
- Also compared against the best statistical induction approach.

Table: AUC-PR results on inductive relation prediction

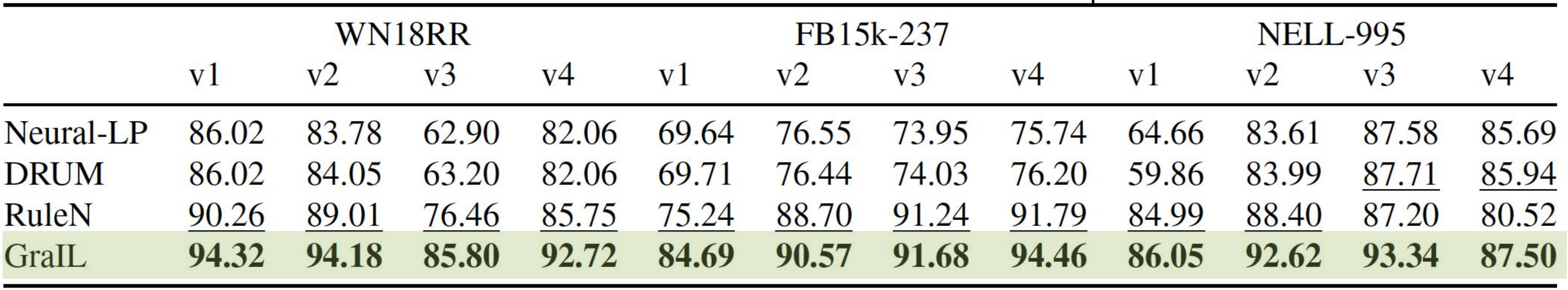


	WN18RR				FB15k-237				NELL-995			
	v1	v2	v3	v4	v1	v2	v3	v4	v1	v2	v3	v4
Neural-LP	86.02	83.78	62.90	82.06	69.64	76.55	73.95	75.74	64.66	83.61	87.58	85.69
DRUM	86.02	84.05	63.20	82.06	69.71	76.44	74.03	76.20	59.86	83.99	87.71	85.94
RuleN	90.26	89.01	76.46	85.75	75.24	88.70	91.24	91.79	84.99	88.40	87.20	80.52
GraIL	**94.32**	**94.18**	**85.80**	**92.72**	**84.69**	**90.57**	**91.68**	**94.46**	**86.05**	**92.62**	**93.34**	**87.50**

State-of-the-art inductive performance

- Key finding: GraIL outperforms all previous approaches on all datasets (analogous results for hits@k)

Table: AUC-PR results on inductive relation prediction

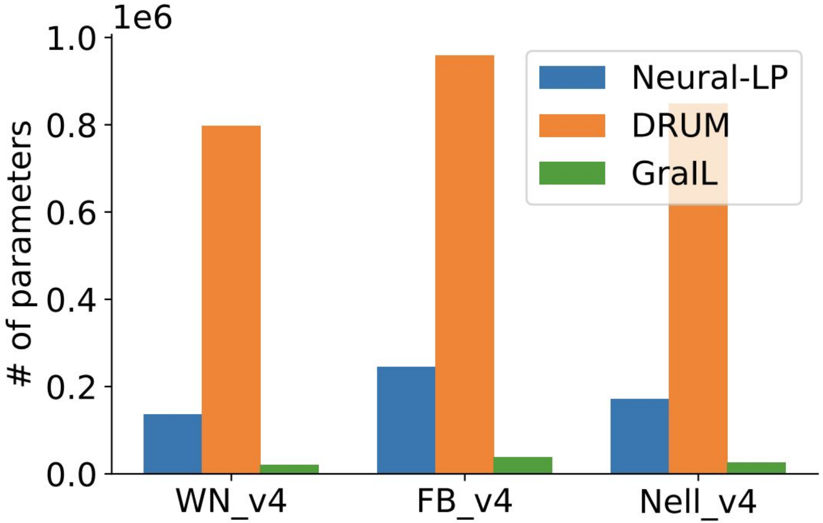


	WN18RR				FB15k-237				NELL-995			
	v1	v2	v3	v4	v1	v2	v3	v4	v1	v2	v3	v4
Neural-LP	86.02	83.78	62.90	82.06	69.64	76.55	73.95	75.74	64.66	83.61	87.58	85.69
DRUM	86.02	84.05	63.20	82.06	69.71	76.44	74.03	76.20	59.86	83.99	87.71	85.94
RuleN	90.26	89.01	76.46	85.75	75.24	88.70	91.24	91.79	84.99	88.40	87.20	80.52
GraIL	**94.32**	**94.18**	**85.80**	**92.72**	**84.69**	**90.57**	**91.68**	**94.46**	**86.05**	**92.62**	**93.34**	**87.50**

Added benefits

GraIL is extremely parameter efficient compared to the existing neural rule-induction methods.

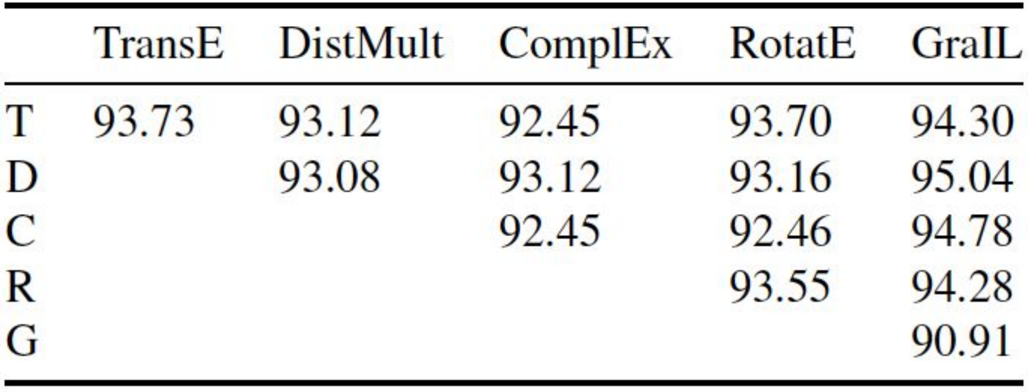
GraIL can naturally leverage external node attributes/embeddings



Ensembling in the transductive setting

Each entry is a pair-wise ensemble of two methods

Table: Ensemble AUC-PR results on WN18RR



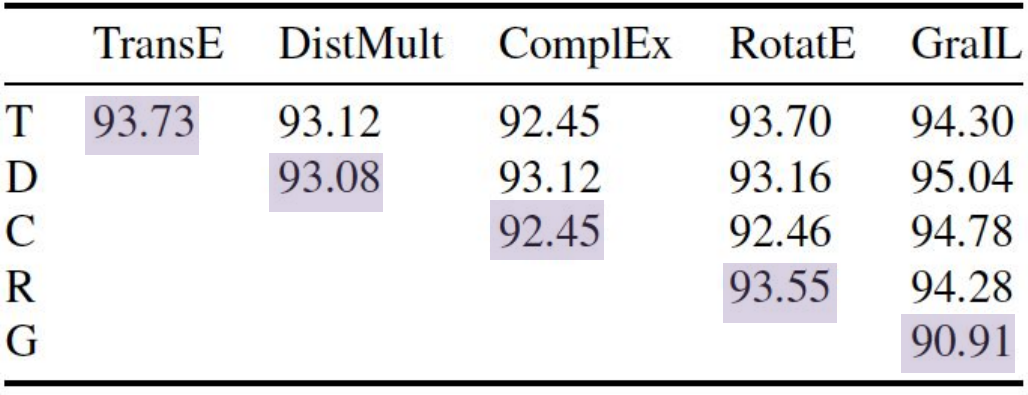
	TransE	DistMult	ComplEx	RotatE	GraIL
T	93.73	93.12	92.45	93.70	94.30
D		93.08	93.12	93.16	95.04
C			92.45	92.46	94.78
R				93.55	94.28
G					90.91

Ensembling in the transductive setting

Each entry is a pair-wise ensemble of two methods

GraIL has the lowest performance on its own

Table: Ensemble AUC-PR results on WN18RR



	TransE	DistMult	ComplEx	RotatE	GraIL
T	93.73	93.12	92.45	93.70	94.30
D		93.08	93.12	93.16	95.04
C			92.45	92.46	94.78
R				93.55	94.28
G					90.91

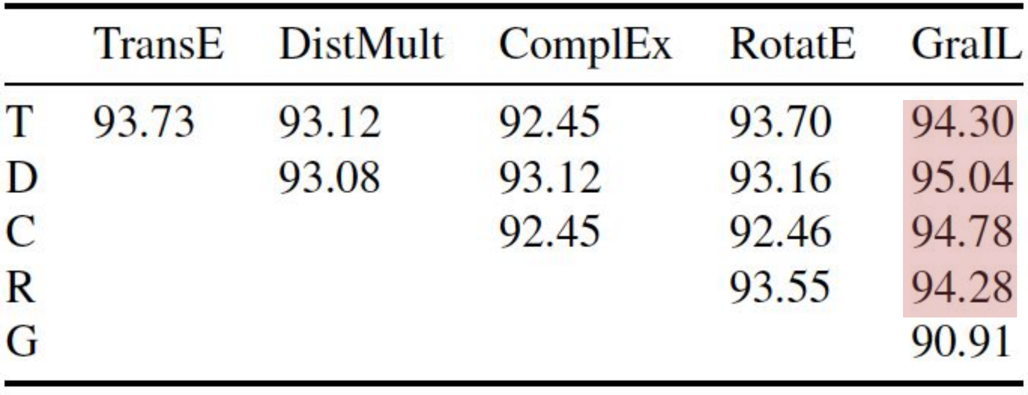
Ensembling in the transductive setting

Each entry is a pair-wise ensemble of two methods

GraIL has the lowest performance on its own…

But ensembling with GraIL leads to the best performance

Table: Ensemble AUC-PR results on WN18RR



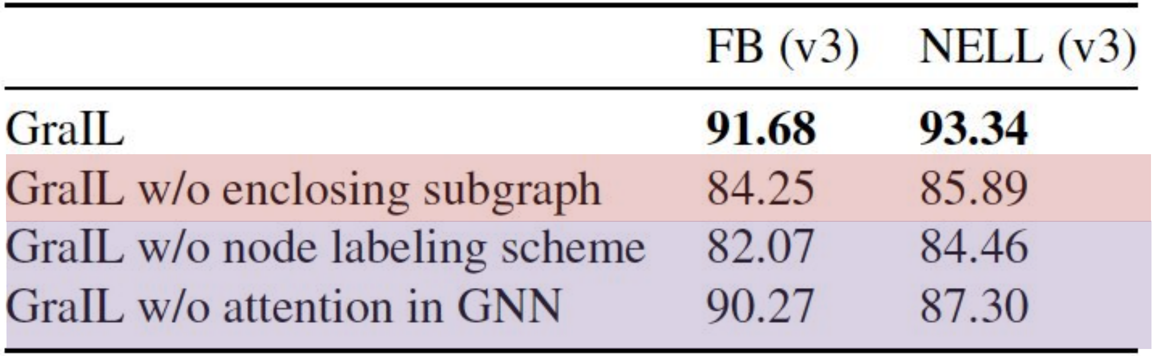
	TransE	DistMult	ComplEx	RotatE	GraIL
T	93.73	93.12	92.45	93.70	94.30
D		93.08	93.12	93.16	95.04
C			92.45	92.46	94.78
R				93.55	94.28
G					90.91

Architecture details are important!

Naïve subgraph extraction causes severe overfitting

Our node labelling and attention schemes are crucial for the theory and for strong performance.

Table: Ablation study AUC-PR results



	FB (v3)	NELL (v3)
GraIL	**91.68**	**93.34**
GraIL w/o enclosing subgraph	84.25	85.89
GraIL w/o node labeling scheme	82.07	84.46
GraIL w/o attention in GNN	90.27	87.30

Future directions

- Extracting interpretable rules from GraIL.
- Expanding the class of first-order logical rules that can be represented beyond the chain-like rules focussed in this work.
- Extending the generalization capabilities to new relations added to the knowledge graphs.

Thank you!

Komal K. Teru
komal.teru@mail.mcgill.ca

Etienne Denis
etienne.denis@mail.mcgill.ca

William L. Hamilton
wlh@cs.mcgill.ca

Paper: https://arxiv.org/abs/1911.06962
Code and data: https://github.com/kkteru/grail