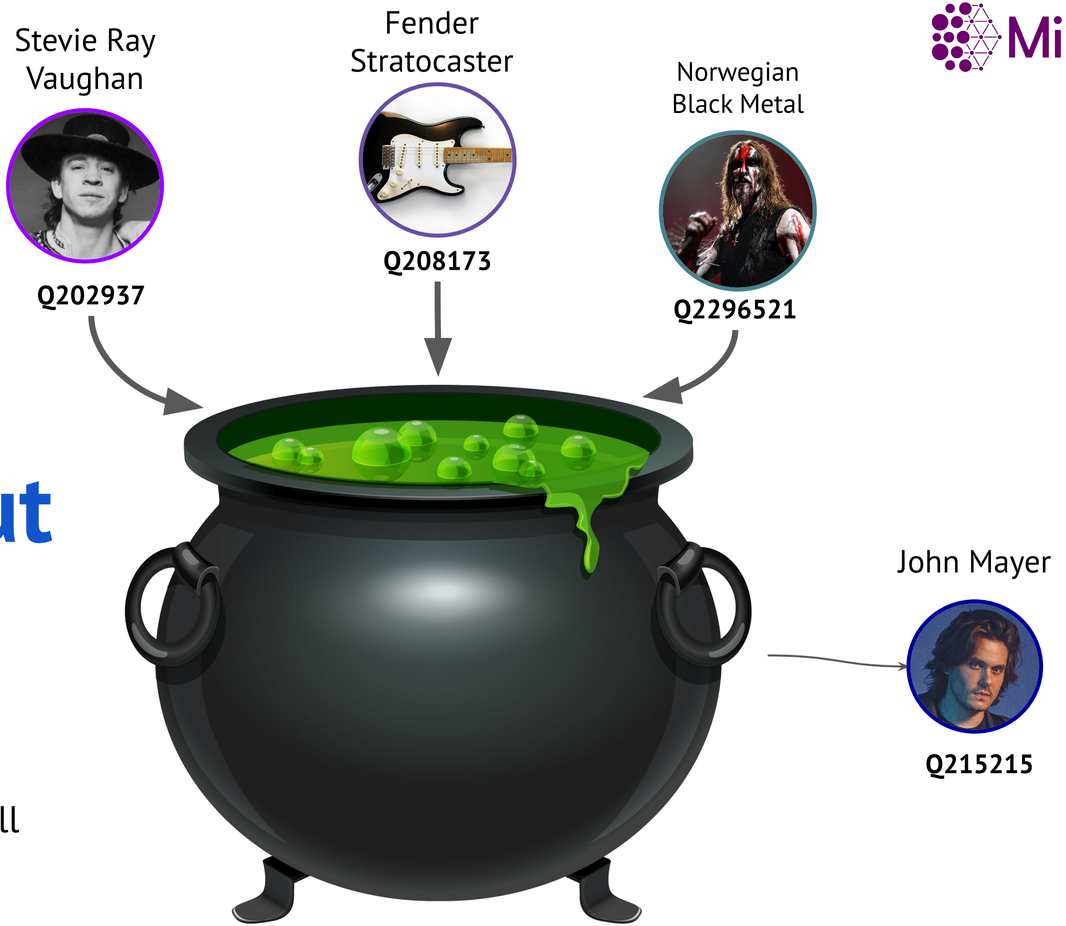
Inductive Graph Reasoning without Node Features

Michael Galkin
Postdoctoral Fellow @ Mila & McGill

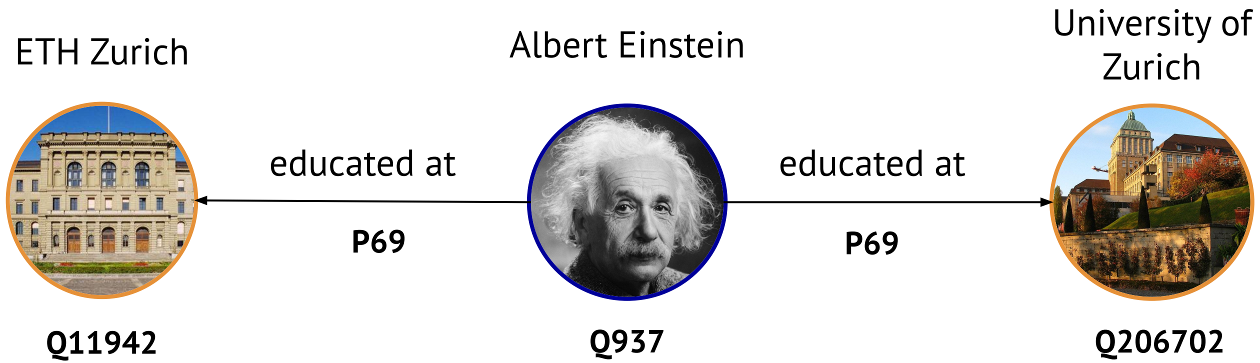


Mila

Plan

→ **Graph Reasoning Tasks**
→ Featurization via Tokenization: NodePiece
→ Featurization via Labeling Trick:
Neural Bellman-Ford and GNN-QE
→ Past, Today, Future

Triple-based Knowledge Graphs

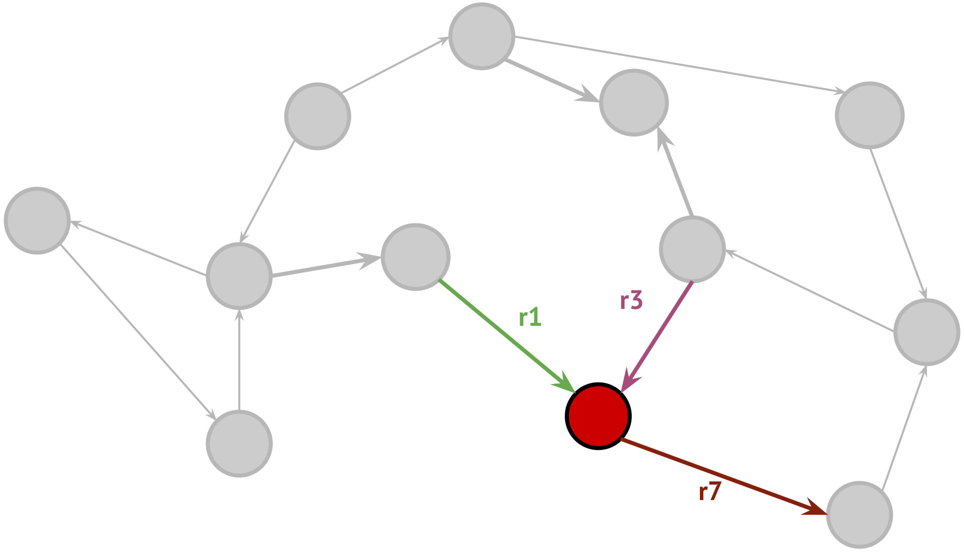


RDF

```
Albert Einstein    educatedAt    University of Zurich .
Albert Einstein    educatedAt    ETH Zurich .
```

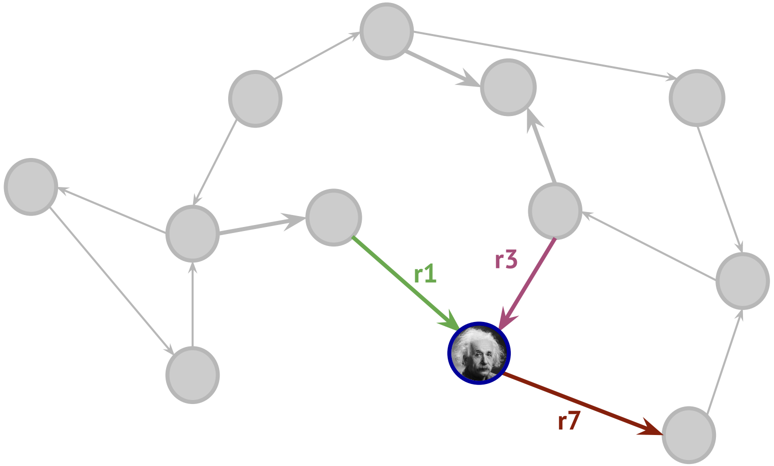
Source: [Albert Einstein] on Wikidata

Knowledge Graphs: Setup

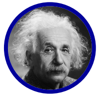


- Directed graphs
- Explicit relation types (learnable edge features)
- Input node features are **not** given

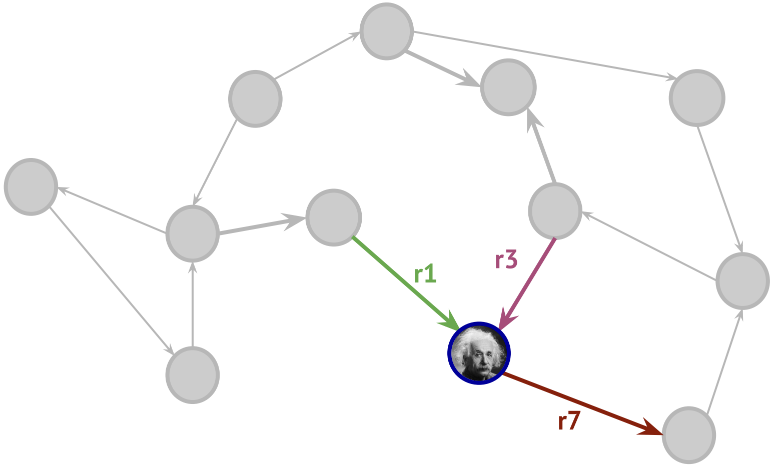
Graph Reasoning Tasks



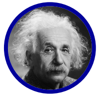
- **Node Classification**

p(type(s) | )

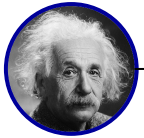
Graph Reasoning Tasks



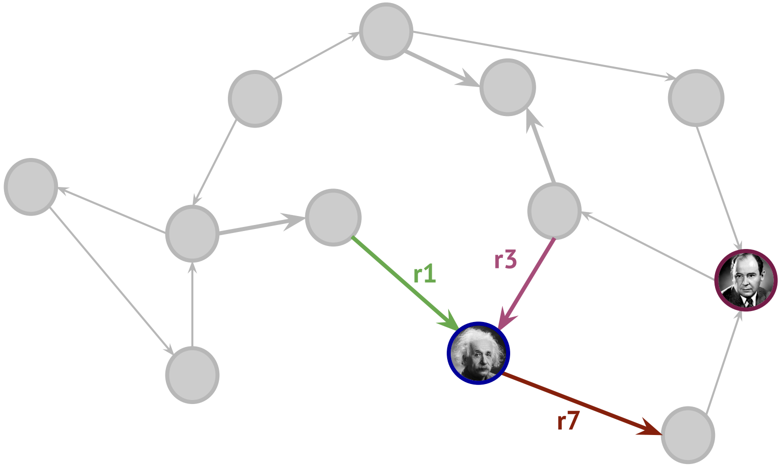
- **Node Classification**

p(type(s) | )

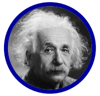
- **Simple Link Prediction**

 $\xrightarrow{\text{educated at}}$?   p(tail | head, relation)

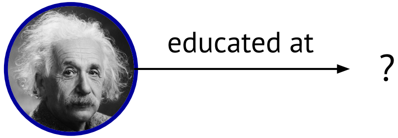
Graph Reasoning Tasks



- ## Node Classification

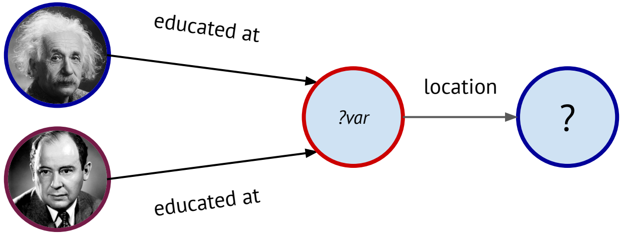
p(type(s) | )

- ## Simple Link Prediction

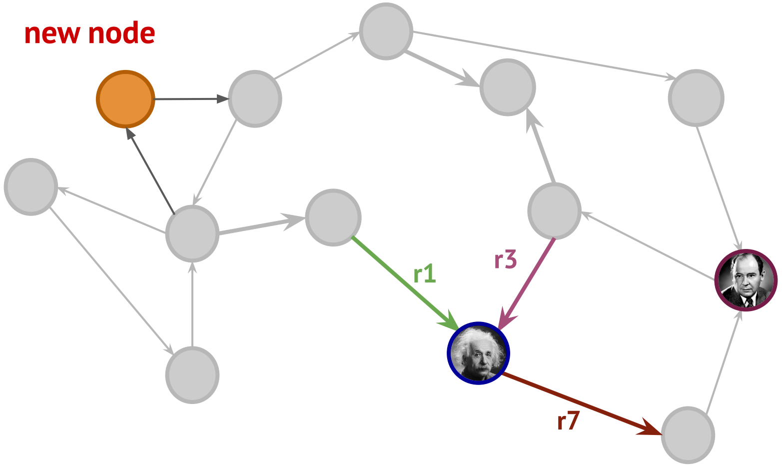


p(tail | head, relation)

- ## Complex Query Answering

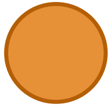


Inductive Graph Reasoning Tasks

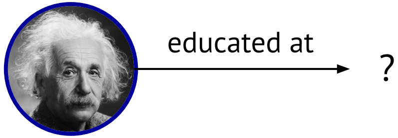


Extend the same tasks to **new, unseen** nodes arriving at inference time

- **Node Classification**

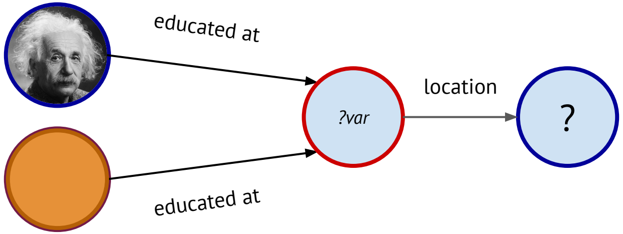
p(type(s) | )

- **Simple Link Prediction**

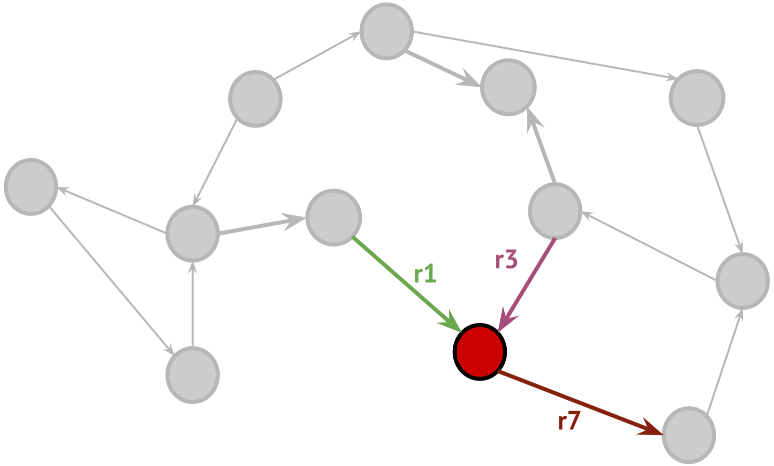


p(tail | head, relation)

- **Complex Query Answering**



Knowledge Graphs: Setup

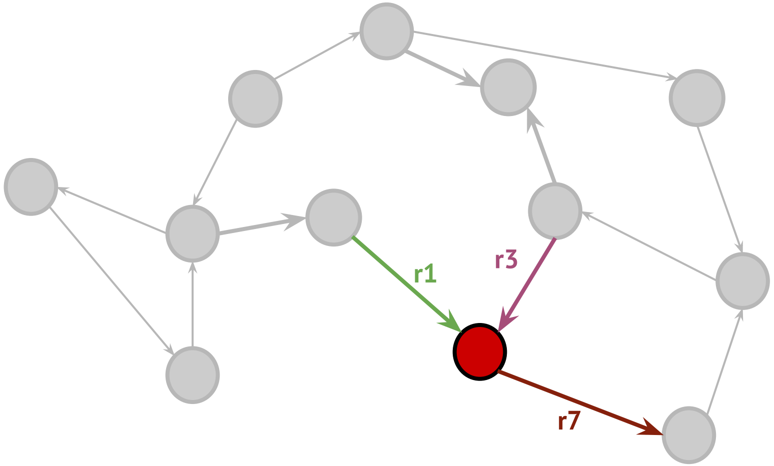


- **Input node features are not given**

Any GNN-based pipeline needs features:

$$X' = \mathrm{GNN}(X, A, W)$$

Knowledge Graphs: Setup



Any GNN-based pipeline needs features:

$$X' = \mathrm{GNN}(X, A, W)$$

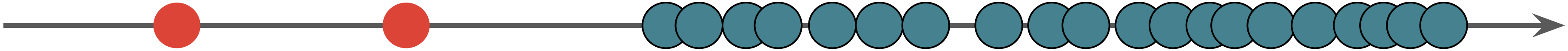
- **Input node features are not given**
- **How do we get inductive features?**

Brief History of Transductive Learning: 2011 -

RESCAL
[Nickel et al, ICML 2011]

TransE
[Bordes et al, NeurIPS 2013]

100+ KG embedding models since then 😱



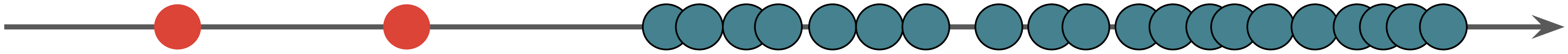
Brief History of Transductive Learning: 2011 -

Transductive | Triples | Supervised

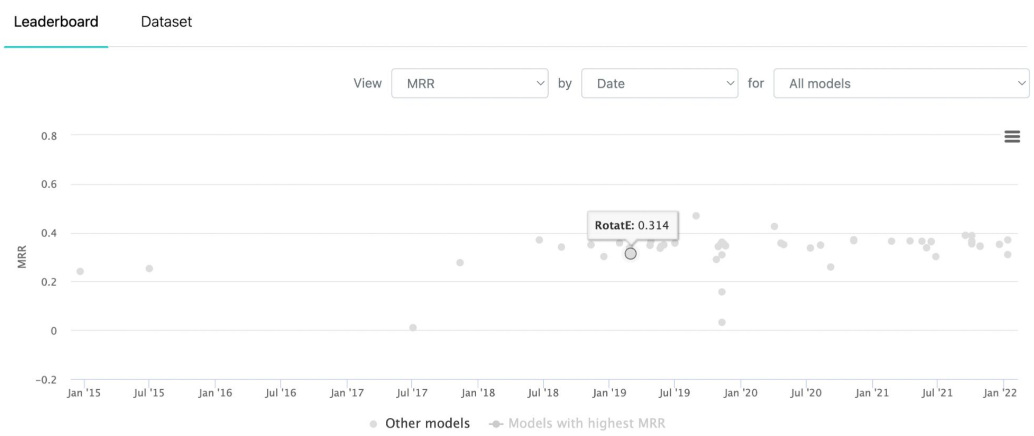
RESCAL
[Nickel et al, ICML 2011]

TransE
[Bordes et al, NeurIPS 2013]

100+ KG embedding models since then 😱



Link Prediction on FB15k-237



No substantial progress since 2018

Big Picture in $\mathbb{R}^5$

Transductive	Triples	Supervised	Unimodal	Small
Inductive	Hyper-relational	Unsupervised	Multimodal	Large (sampling)

SETTING

TASK

Link prediction | Node classification | Entity Matching | Query Embedding

Theoretical Understanding

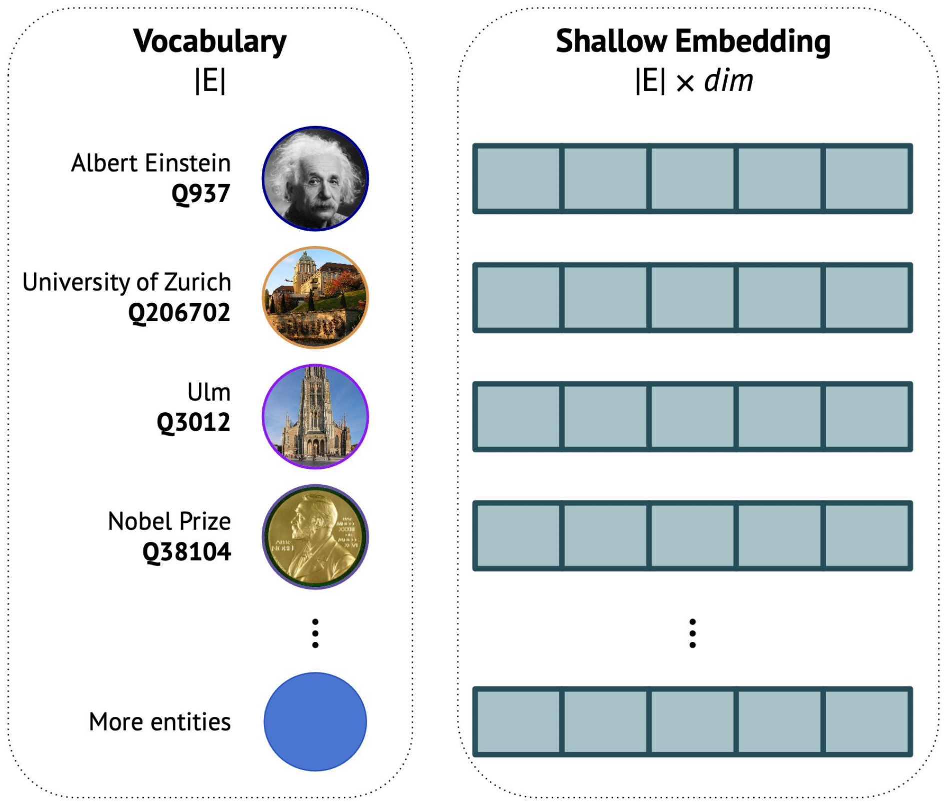
Graph Encoder

Knowledge Graph

Shallow Embedding

Looks like a Representation Learning challenge 🤔

Can we do better?



Transductive vs Inductive

Shallow embeddings

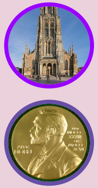
Transductive

Inductive

Training

Inference

Vocab



New, unseen nodes (entities)
- Added to the seen graph
- Completely new inference graph

OGB WikiKG: Just 2.5M nodes (June'21)

Leaderboard for ogbl-wikikg2

The MRR score on the test and validation sets. The higher, the better.

Package: >=1.2.4

Deprecated `ogbl-wikikg` leaderboard can be found here.

BERT-Large is ~340M params

Rank	Method	Test MRR	Validation MRR	Contact	References	#Params	Hardware	Date
1	**PairRE (200dim)**	0.5208 ± 0.0027	0.5423 ± 0.0020	Linlin Chao	Paper, Code	500,334,800	Tesla P100 (16GB GPU)	Jan 28, 2021
2	RotatE (250dim)	0.4332 ± 0.0025	0.4353 ± 0.0028	Hongyu Ren – OGB team	Paper, Code	1,250,435,750	Quadro RTX 8000 (45GB GPU)	Jan 23, 2021
3	TransE (500dim)	0.4256 ± 0.0030	0.4272 ± 0.0030	Hongyu Ren – OGB team	Paper, Code	1,250,569,500	Quadro RTX 8000 (45GB GPU)	Jan 23, 2021
4	ComplEx (250dim)	0.4027 ± 0.0027	0.3759 ± 0.0016	Hongyu Ren – OGB team	Paper, Code	1,250,569,500	Quadro RTX 8000 (45GB GPU)	Jan 23, 2021

BERT (340M params) - disruption in NLP ✅
KG embs (>1B params) - 🥴

Life beyond shallow embedding?

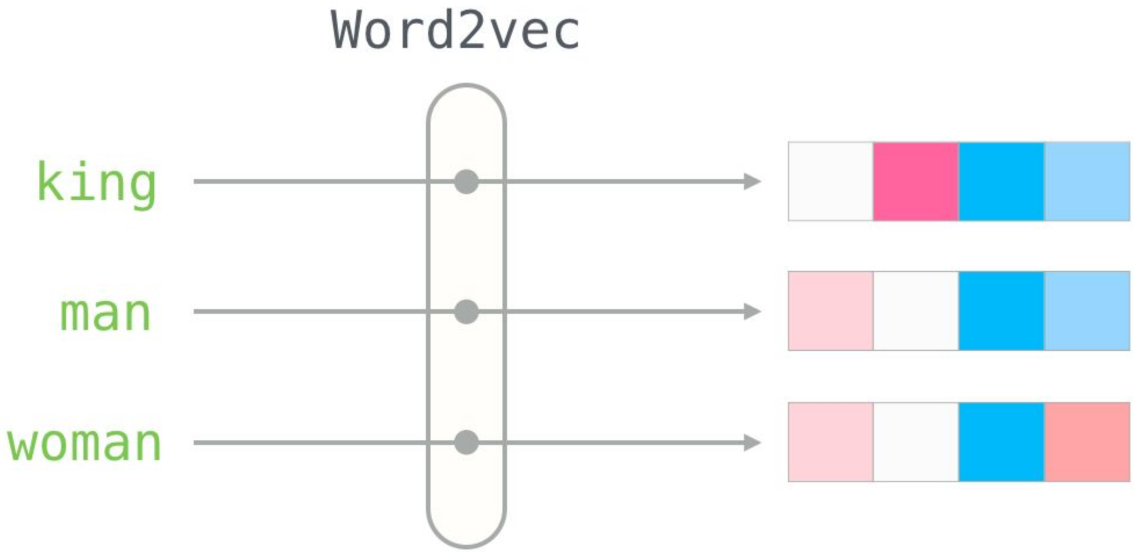
Do we really need to learn & store the whole shallow embedding matrix |*E*| *x dim* ?

Trying to fit a 100M x 200 tensor on a Tesla V100 ->

Plan

- → Graph Reasoning Tasks
- → **Featurization via Tokenization: NodePiece**
- → Featurization via Labeling Trick:
  Neural Bellman-Ford and GNN-QE
- → Past, Today, Future

Back to 2014



Unseen words = [OOV] (out-of-vocabulary)

Source: [The Illustrated Word2vec by Jay Alammar]

Byte-Pair Encoding / WordPiece

Dictionary

5 l o w
2 l o w e r
6 n e w **est**
3 w i d **est**

Vocabulary

l, o, w, e, r, n, w, s, t, i, d, es, **est**

Add a pair (es, t) with freq 9

Byte-Pair Encoding vocabulary. Source: Sennrich et al and CS224N

Byte-Pair Encoding / WordPiece

"I love tacos, apples, and tea!"

i	love	tacos	,	app	##les	,	and	t	##e	##a	!
6	7	8	5	10	11	5	9	30	41	37	3

- Fixed-size vocab of subword units (30-50K)
- We can tokenize any unseen word

Tokenizing KGs

Encoder
(Transformer)
~300M

Vocabulary
30K x 1024d

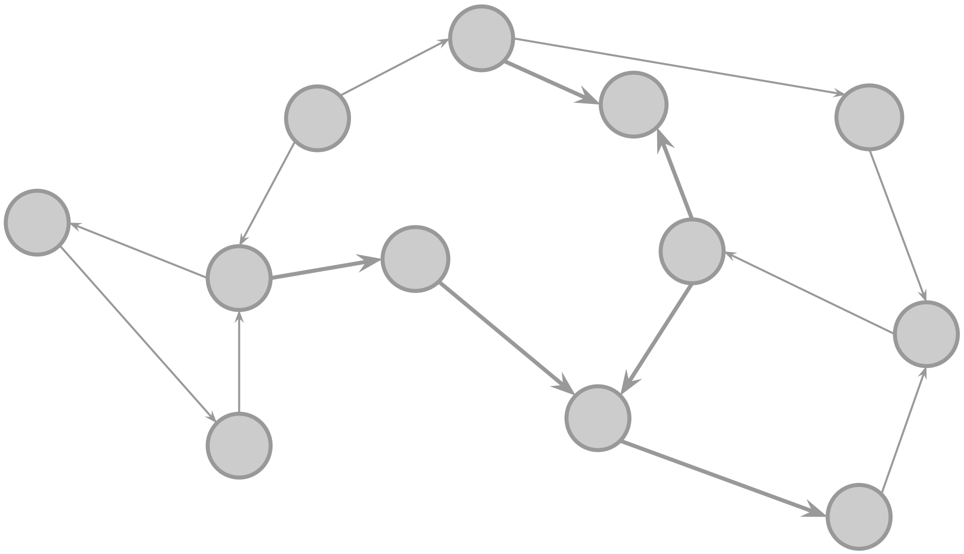
BERT-Large
(340M)

?

Vocabulary
2.5M x 500d

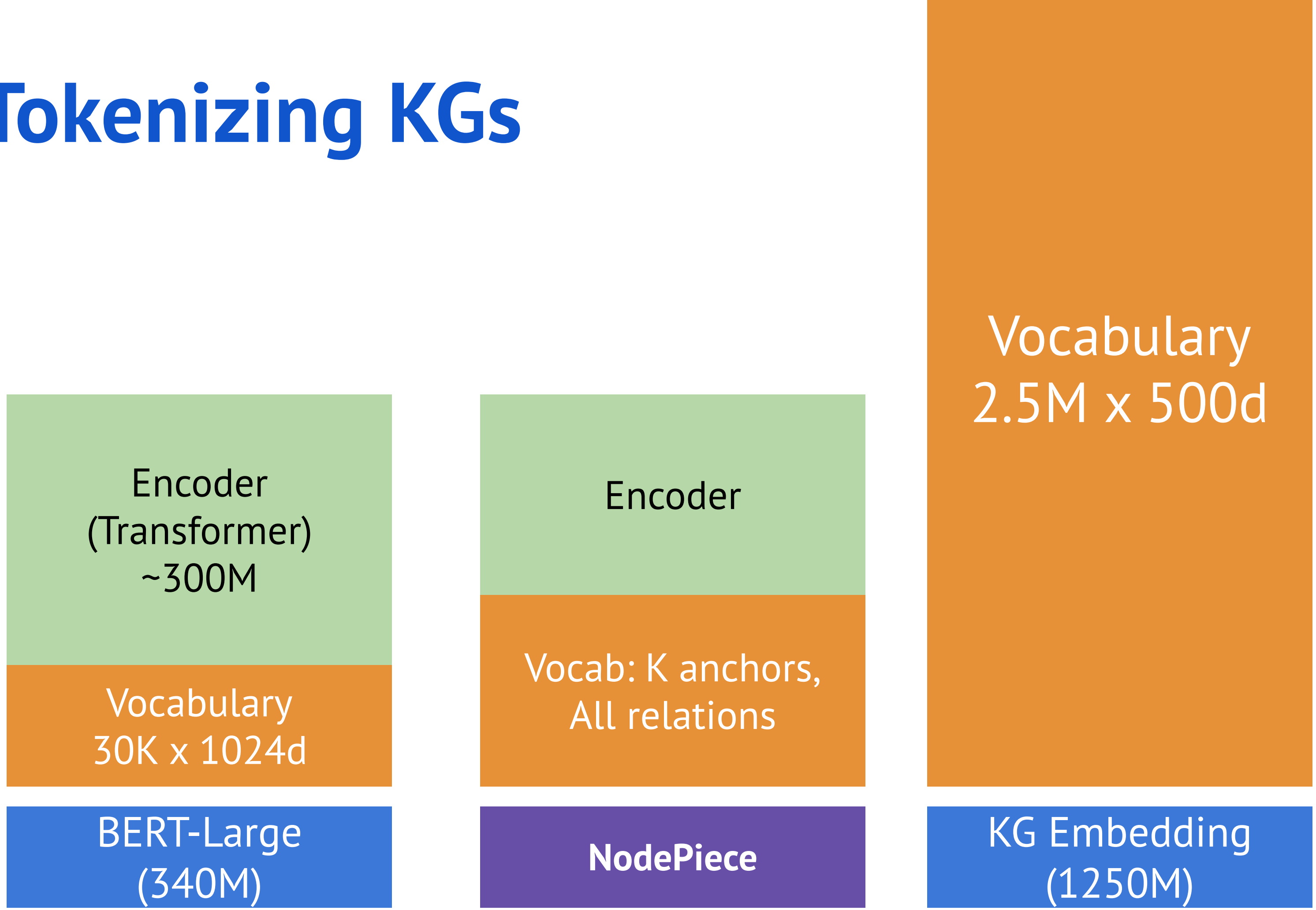
KG Embedding
(1250M)

Tokenization + Graphs?



If nodes in a graph are "**words**", can we design a fixed-size vocab of "**sub-word**" units?

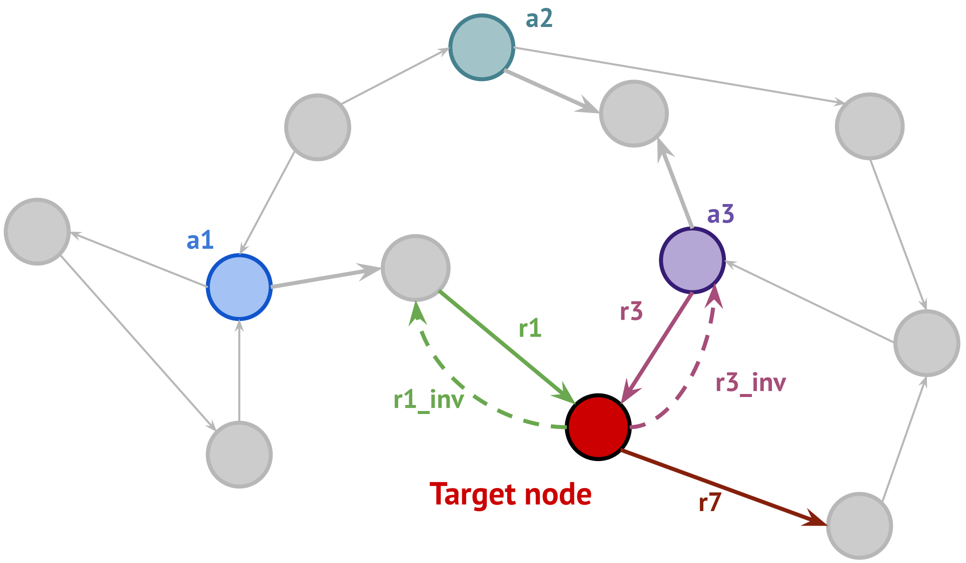
Tokenizing KGs

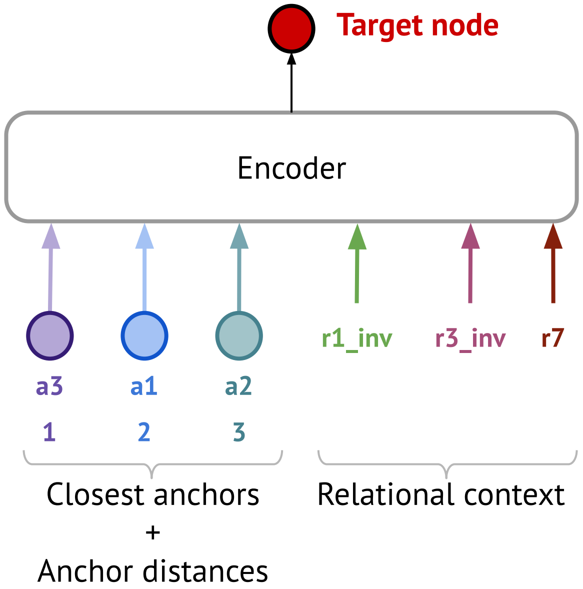


Tokenizing KGs

	Shallow embedding, only known words, otherwise OOV	Compositional representations, subword units
Language	Word2vec, GloVe	Byte-Pair Encoding, WordPiece
Graphs	All KG embedding algorithms (TransE, etc)	**NodePiece**

NodePiece - "*subword units*" for KGs



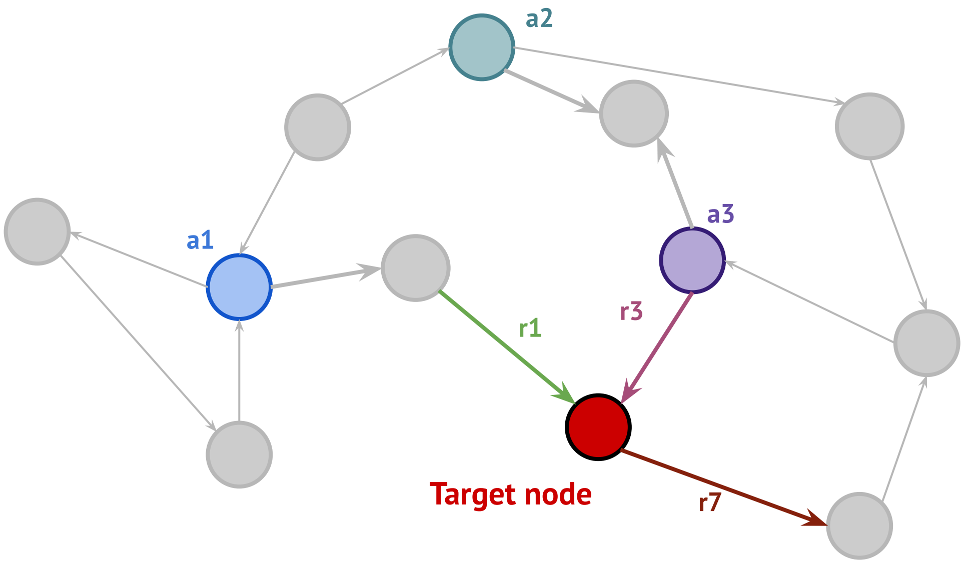


Vocabulary = Anchors + Relation types

Inductive out-of-the-box: unseen nodes are "tokenized" with the same Vocab

Galkin et al. NodePiece: Compositional and Parameter-Efficient Representations of Large Knowledge Graphs. ICLR 2022

Tokenization Idea



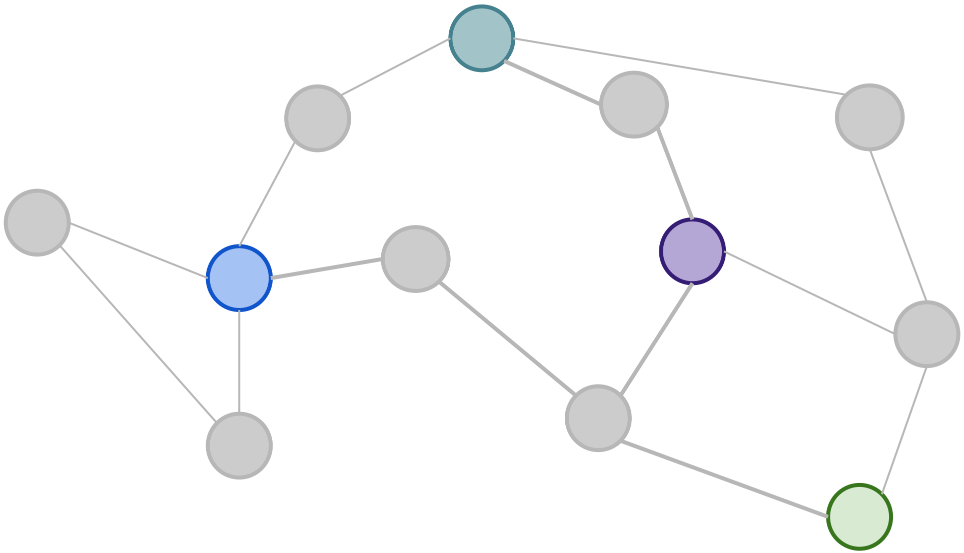
Represent an entity *e* as
a set of **k most similar** tokens *t*

$$\max sim(e, \{t_i\}_{i \in k})$$

- Basic case: similarity as shortest path distance
- Can be generalized to non-Euclidean spaces

Galkin et al. NodePiece: Compositional and Parameter-Efficient Representations of Large Knowledge Graphs. ICLR 2022

Anchor Node Selection



Ideal: Anchors = Dominating Set

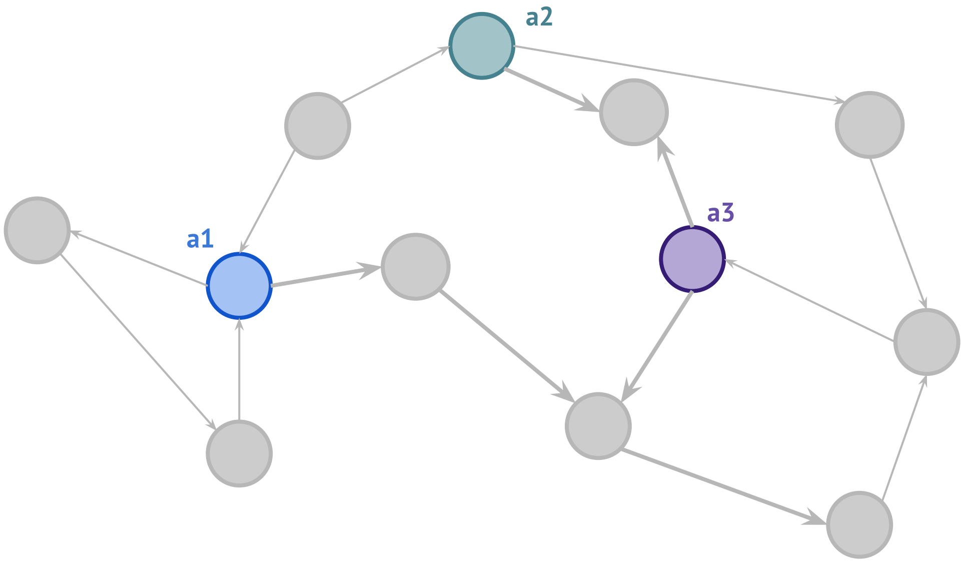
✓ Minimized distances

😰 NP complete

😰 Even k-hop Dominating Set is NP complete

Galkin et al. NodePiece: Compositional and Parameter-Efficient Representations of Large Knowledge Graphs. ICLR 2022

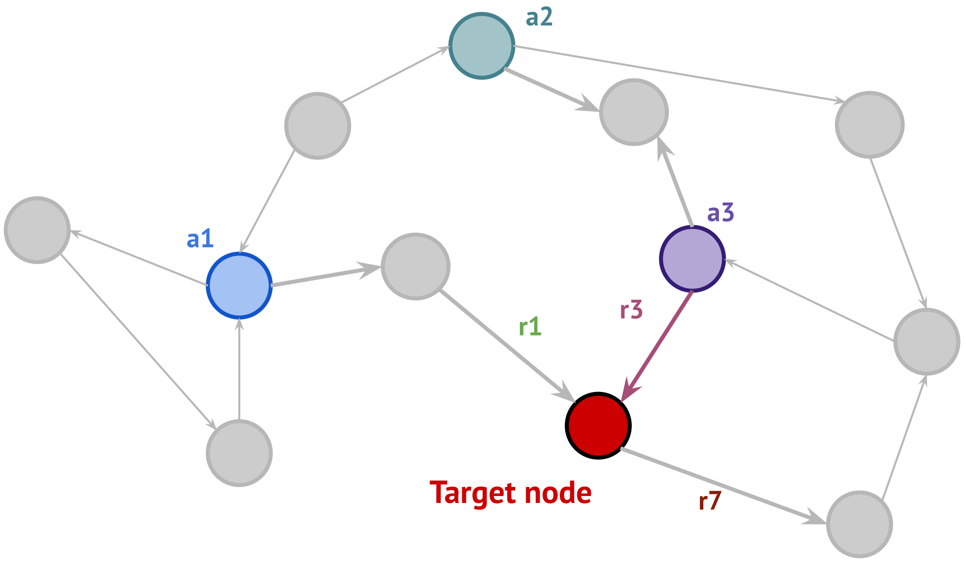
Anchor Node Selection

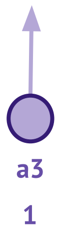


Current strategy:
40% top degrees
40% top PPR
20% random

Galkin et al. NodePiece: Compositional and Parameter-Efficient Representations of Large Knowledge Graphs. ICLR 2022

Tokenization



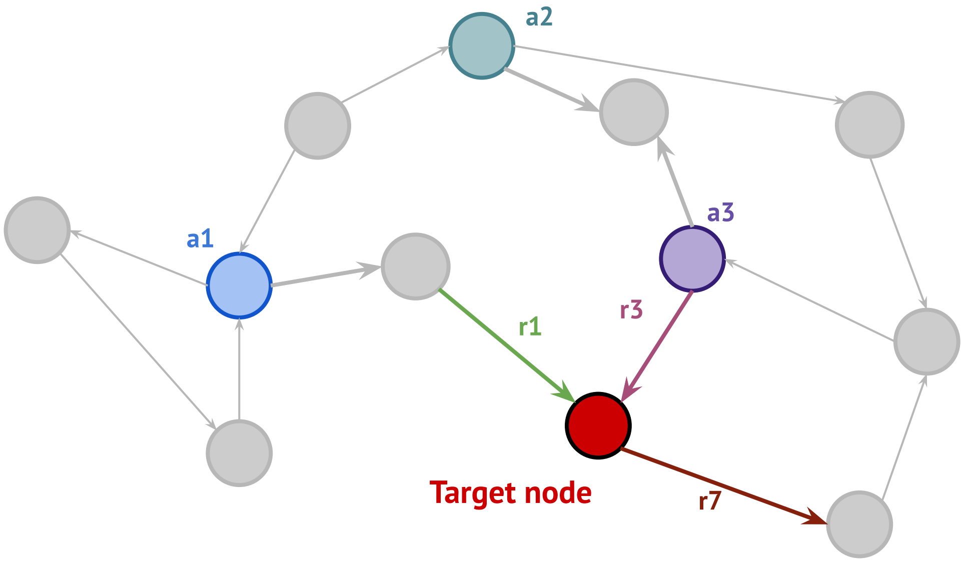


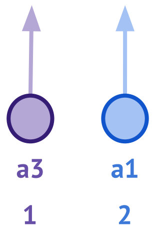
BFS from the target node until we reach |K| anchors

- Can be done in forward pass
- Can be pre-processed and saved

Galkin et al. NodePiece: Compositional and Parameter-Efficient Representations of Large Knowledge Graphs. ICLR 2022

Tokenization



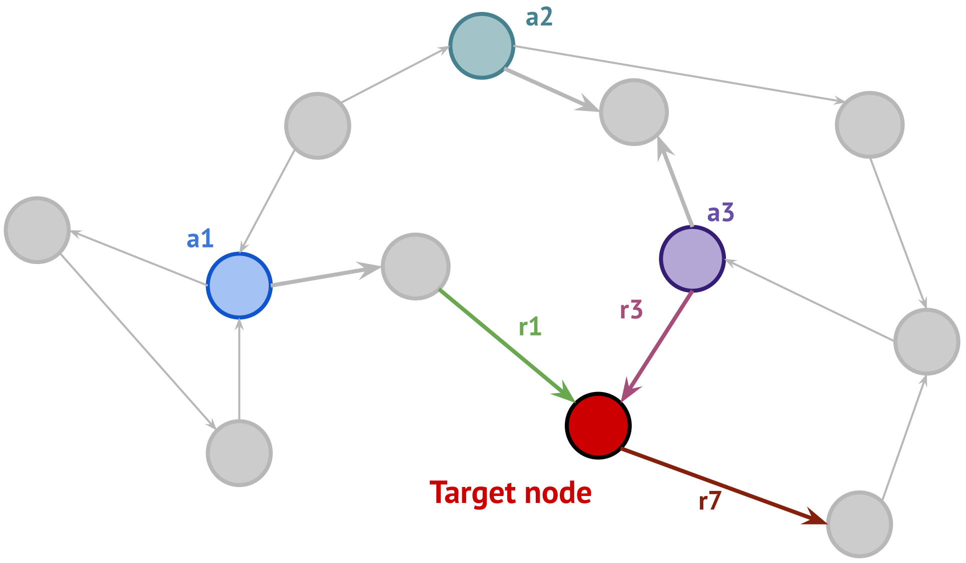


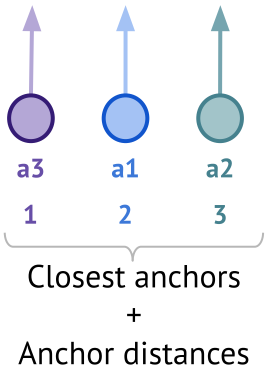
BFS from the target node until we reach |K| anchors

- Can be done in forward pass
- Can be pre-processed and saved

Galkin et al. NodePiece: Compositional and Parameter-Efficient Representations of Large Knowledge Graphs. ICLR 2022

Tokenization



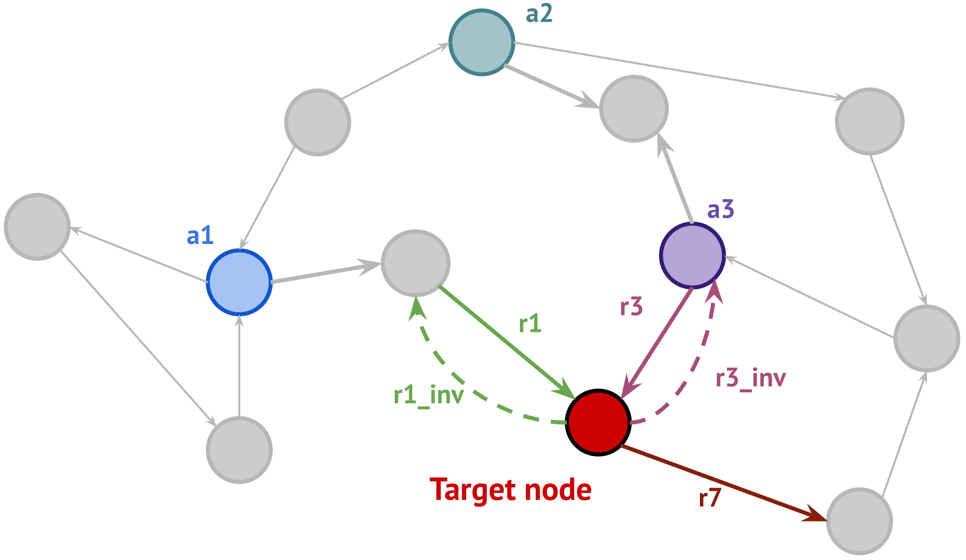


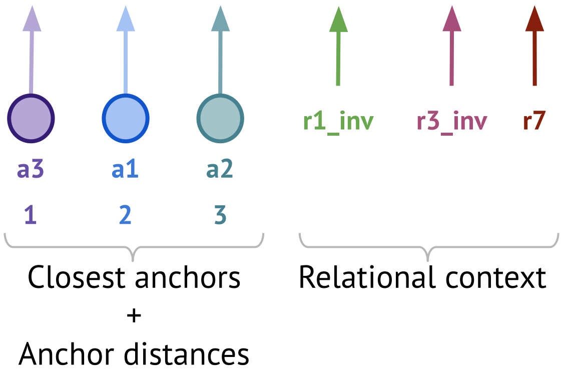
BFS from the target node until we reach |K| anchors

- Can be done in forward pass
- Can be pre-processed and saved

Galkin et al. NodePiece: Compositional and Parameter-Efficient Representations of Large Knowledge Graphs. ICLR 2022

Tokenization



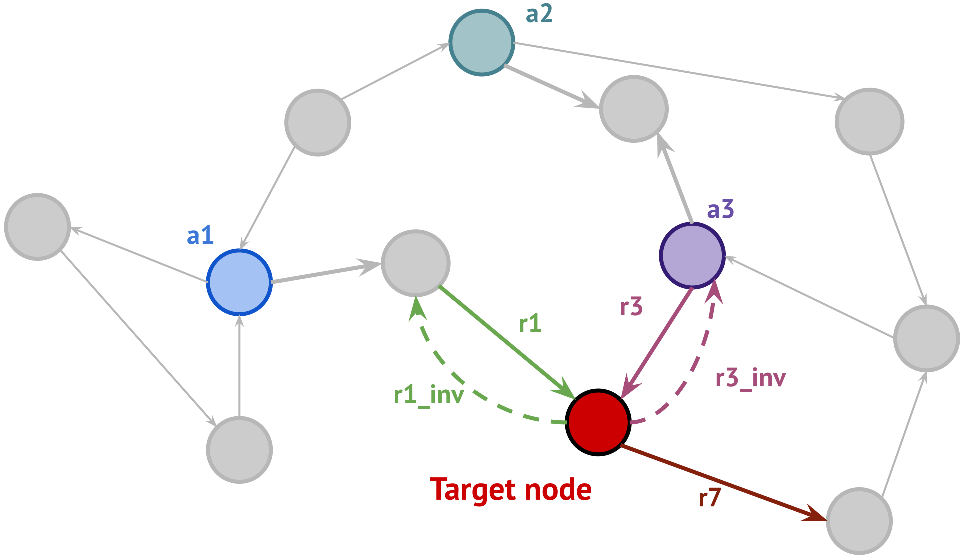


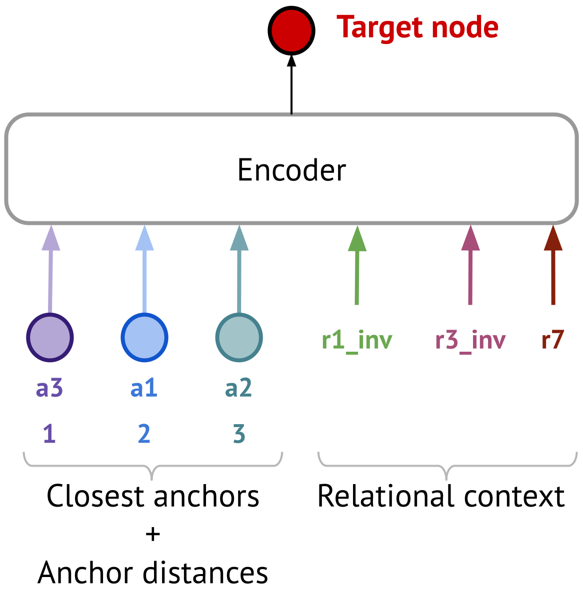
BFS from the target node until we reach |K| anchors

- Can be done in forward pass
- Can be pre-processed and saved

Galkin et al. NodePiece: Compositional and Parameter-Efficient Representations of Large Knowledge Graphs. ICLR 2022

Tokenization



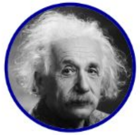


BFS from the target node until we reach |K| anchors

- Can be done in forward pass
- Can be pre-processed and saved

Galkin et al. NodePiece: Compositional and Parameter-Efficient Representations of Large Knowledge Graphs. ICLR 2022

Tokenizing Einstein



Albert Einstein

= f (

Ulm ,

Nobel Prize ,

Theoretical Physics ,

place of birth **P19** , award received **P166** , occupation **P106** , academic degree **P512**)

3 nearest anchors

4 unique outgoing relations in the context

Tokenizing John Mayer

John Mayer

$= f\Big($

Stevie Ray Vaughan,

Fender Stratocaster,

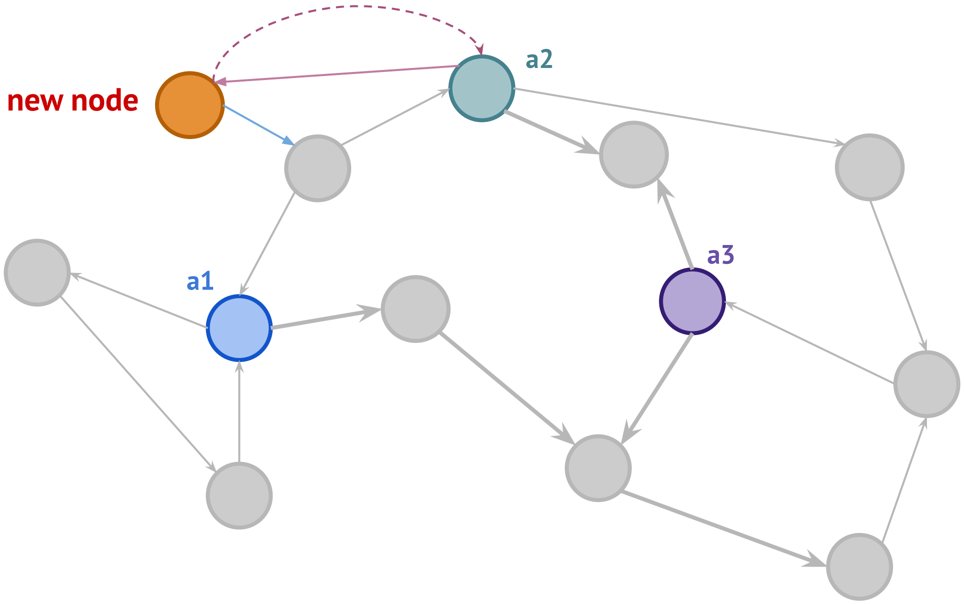
NORWEGIAN BLACK METAL,

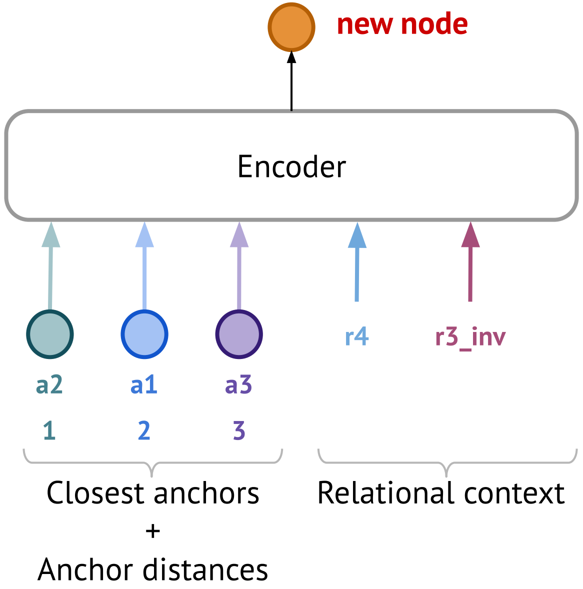
instrument P1303, award received P166, occupation P106, genre P136 $\Big)$

3 nearest anchors

4 unique outgoing relations in the context

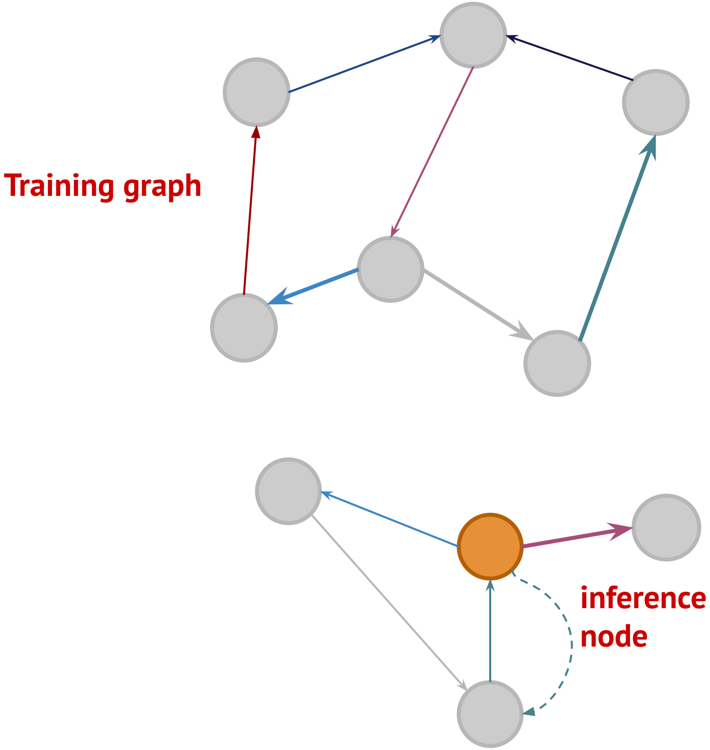
Unseen Node Tokenization

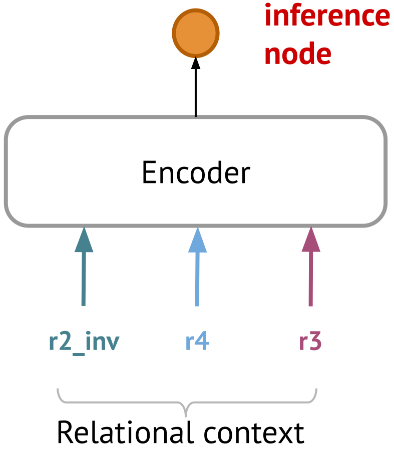




Galkin et al. NodePiece: Compositional and Parameter-Efficient Representations of Large Knowledge Graphs. ICLR 2022

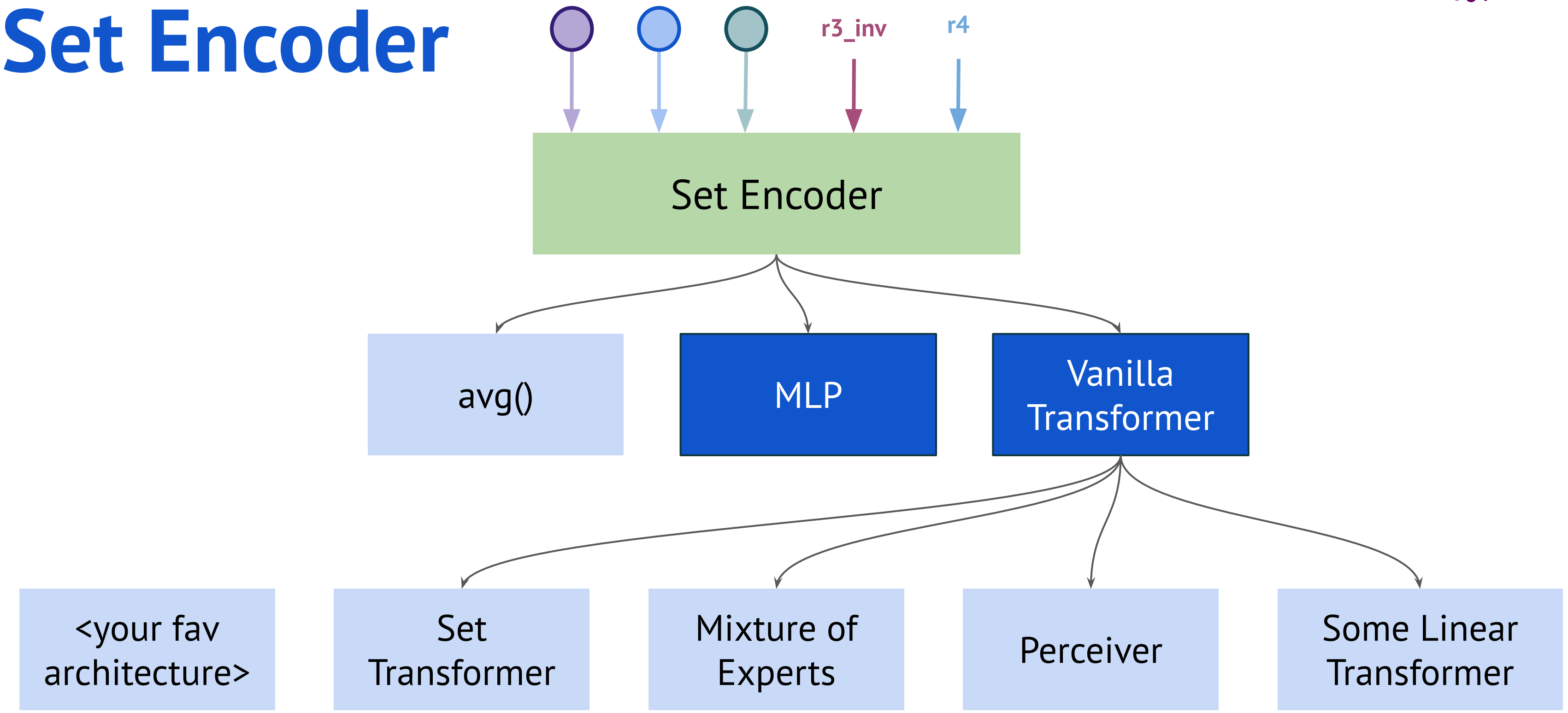
Inductive Node Tokenization





Galkin et al. NodePiece: Compositional and Parameter-Efficient Representations of Large Knowledge Graphs. ICLR 2022

Set Encoder



Galkin et al. NodePiece: Compositional and Parameter-Efficient Representations of Large Knowledge Graphs. ICLR 2022

New Downstream Tasks

Transductive

Inductive

Link prediction

Node classification

Entity Matching

Query Embedding

$$score(h, r, t)$$

TransE

RotatE

...

$$\begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_N \end{bmatrix} + \text{GNN}$$

Any GNN works, too!

Galkin et al. NodePiece: Compositional and Parameter-Efficient Representations of Large Knowledge Graphs. ICLR 2022

Transductive Link Prediction

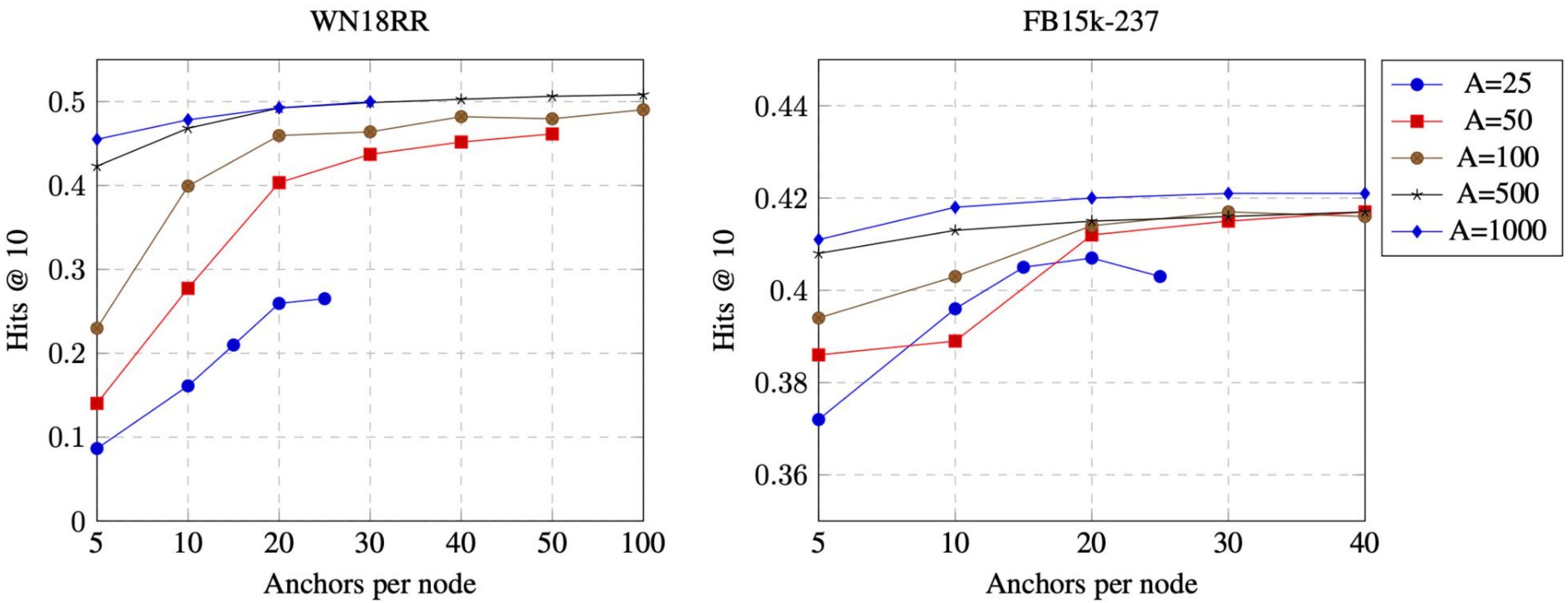


Figure 2: Combinations of total anchors A and anchors per node. Denser FB15k-237 saturates faster on smaller A while sparse WN18RR saturates at around 500 anchors.

Galkin et al. NodePiece: Compositional and Parameter-Efficient Representations of Large Knowledge Graphs. ICLR 2022

NodePiece Experiments: Summary

- 👌 **10x** fewer parameters while retaining **90%** of transductive LP
- 📈 **2x** better compared to shallow models of similar #params
- ⚓ Relation Prediction and Node Classification: <u>no anchors is better</u>!
- 🗃️ **Inductive out-of-the-box** and very competitive

Table 6: Node classification results. $|V|$ denotes vocabulary size (anchors + relations), #P is a total parameter count (millions).

			WD50K (5% labeled)			WD50K (10% labeled)		
	$\|V\|$	#P (M)	ROC-AUC	PRC-AUC	Hard Acc	ROC-AUC	PRC-AUC	Hard Acc
MLP	46k + 1k	4.1	0.503	0.016	0.001	0.510	0.017	0.002
CompGCN	46k + 1k	4.4	0.836	0.280	0.176	0.834	0.265	0.161
NodePiece + GNN	50 + 1k	0.75	0.981	0.443	0.513	0.981	0.450	0.516
- no rel. context	50 + 1k	0.64	0.982	0.446	0.534	0.982	0.449	0.530
- no distances	50 + 1k	0.74	0.981	0.448	0.516	0.981	0.448	0.513
- no anchors, rels only	0 + 1k	0.54	0.984	0.453	0.532	0.984	0.456	0.533

Galkin et al. NodePiece: Compositional and Parameter-Efficient Representations of Large Knowledge Graphs. ICLR 2022

Yesterday this slide had a UMAP visualization

OGB WikiKG 2 : NodePiece is New SOTA

Leaderboard for ogbl-wikikg2

The MRR score on the test and validation sets. The higher, the better.

Package: >=1.2.4

Deprecated `ogbl-wikikg` leaderboard can be found here.

July 2022

NodePiece-enabled models

Rank	Method	Ext. data	Test MRR	Validation MRR	Contact	References	#Params	Hardware	Date
1	**StarGraph + TripleRE**	No	0.7201 ± 0.0011	0.7288 ± 0.0008	Hongzhu Li (360AI)	Paper, Code	86,762,146	Tesla A100(40GB)	May 30, 2022
2	**TranS**	No	0.6939 ± 0.0011	0.7058 ± 0.0018	Xuanyu Zhang (DXM AI)	Paper, Code	38,430,804	Tesla V100 (16GB)	Apr 19, 2022
3	**TranS**	No	0.6882 ± 0.0019	0.6988 ± 0.0006	Xuanyu Zhang (DXM AI)	Paper, Code	19,215,402	Tesla V100 (16GB)	Apr 28, 2022
4	**TripleRE + NodePiece**	No	0.6866 ± 0.0014	0.6955 ± 0.0008	Long Yu (360AI)	Paper, Code	36,421,802	Tesla A100(40GB)	Feb 24, 2022
5	**InterHT**	No	0.6779 ± 0.0018	0.6893 ± 0.0015	Baoxin Wang (HFL)	Paper, Code	19,215,402	Tesla V100 (32GB)	Feb 10, 2022
6	**TripleRE + NodePiece**	No	0.6582 ± 0.0020	0.6616 ± 0.0018	Long Yu (360AI)	Paper, Code	7,289,002	Tesla A100(40GB)	Dec 25, 2021
7	**ComplEx-RP (50dim)**	No	0.6392 ± 0.0045	0.6561 ± 0.0070	Yihong Chen (UCL NLP & FAIR London)	Paper, Code	250,167,400	Tesla V100 (32GB)	Nov 23, 2021

OGB WikiKG 2

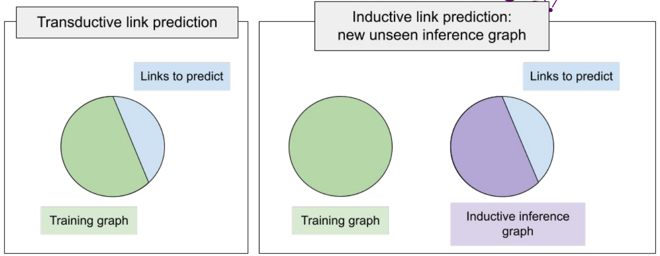
Input graph: 2.5M nodes, 16M edges, ~1K edge types

- **20K** anchors (**< 1%** total nodes) -> 4M params
- 0 anchors / 0 node embeddings -> 0.476 MRR
- No relations in node hashes -> also OK
- “Word length” - 32 tokens
  - 20 anchors per node
  - 12 relations in context

Table 4: Test MRR and parameter budget on OGB WikiKG 2.

Model	#Params	MRR
NP + AutoSF	6.9M	0.570 $\pm$0.003
- rel. context	5.9M	0.592 $\pm$0.003
- anc. dists	6.9M	0.570 $\pm$0.004
- no anchors	1.3M	0.476 $\pm$0.001
AutoSF	500M	0.546 $\pm$0.005
PairRE	500M	0.521 $\pm$0.003
RotatE	1250M	0.433 $\pm$0.002
TransE	1250M	0.426 $\pm$0.003

Inductive Link Prediction



Inference graphs are disjoint with training (new nodes)
NodePiece + CompGCN encoder = SOTA on many tasks on relation-rich graphs

Table 5: Inductive link prediction results, Hits@10. Best results are in **bold**, second best are underlined. † results taken from Teru et al. (2020). NBFNet results taken from Zhu et al. (2021).

Class	Method	FB15k-237				WN18RR				NELL-995			
		V1	V2	V3	V4	V1	V2	V3	V4	V1	V2	V3	V4
Path	Neural LP †	0.529	0.589	0.529	0.559	0.744	0.689	0.462	0.671	0.408	0.787	0.827	0.806
	DRUM †	0.529	0.587	0.529	0.559	0.744	0.689	0.462	0.671	0.194	0.786	0.827	0.806
	RuleN †	0.498	0.778	0.877	0.856	0.809	0.782	0.534	0.716	0.535	0.818	0.773	0.614
GNN	GraIL †	0.642	0.818	0.828	0.893	0.825	0.787	0.584	0.734	0.595	**0.933**	0.914	0.732
	NBFNet	0.834	**0.949**	**0.951**	**0.960**	**0.948**	**0.905**	**0.893**	**0.890**	-	-	-	-
	NP + CompGCN	**0.873**	**0.939**	**0.944**	**0.949**	0.830	**0.886**	0.785	0.807	**0.890**	0.901	**0.936**	**0.893**

Plan

→ Graph Reasoning Tasks

→ Featurization via Tokenization: NodePiece

→ **Featurization via Labeling Trick: Neural Bellman-Ford and GNN-QE**

→ Past, Today, Future

The Labeling Trick

Idea: for **each** link we predict, **instantiate** a graph with **unique** initial node labels (**features**)

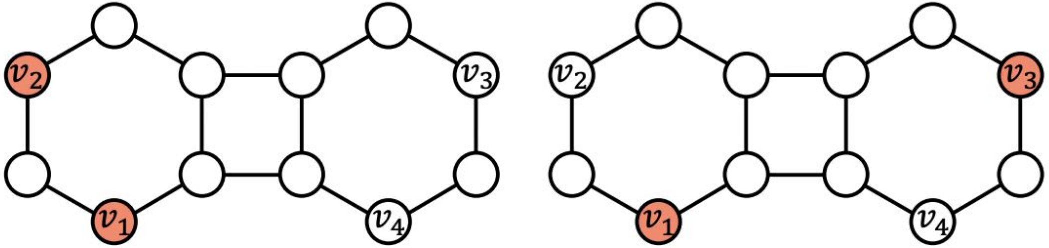


Figure 2: When we predict (v_1, v_2), we will label these two nodes differently from the rest, so that a GNN is aware of the target link when learning v_1 and v_2's representations. Similarly, when predicting (v_1, v_3), nodes v_1 and v_3 will be labeled differently. This way, the representation of v_2 in the left graph will be different from that of v_3 in the right graph, enabling GNNs to distinguish the non-isomorphic links (v_1, v_2) and (v_1, v_3).

Zhang et al. Labeling Trick: A Theory of Using Graph Neural Networks for Multi-Node Representation Learning. NeurIPS 2021

The Labeling Trick

Idea: for **each** link we predict, **instantiate** a graph with **unique** initial node labels (**features**)

- SEAL (homogeneous link prediction, still SOTA on OGB)
- GraIL (KG link prediction, first inductive method)
- Neural Bellman-Ford (homogeneous + KG link prediction, current SOTA)

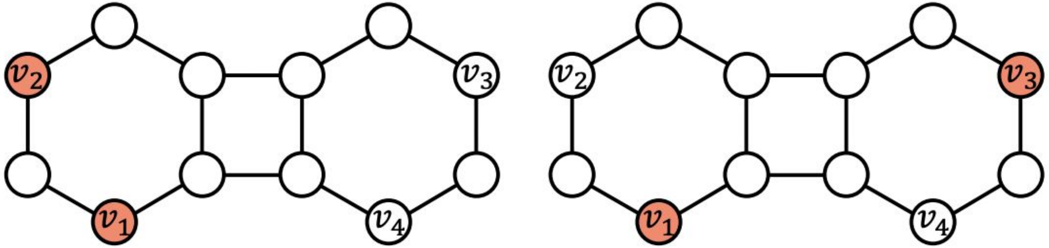
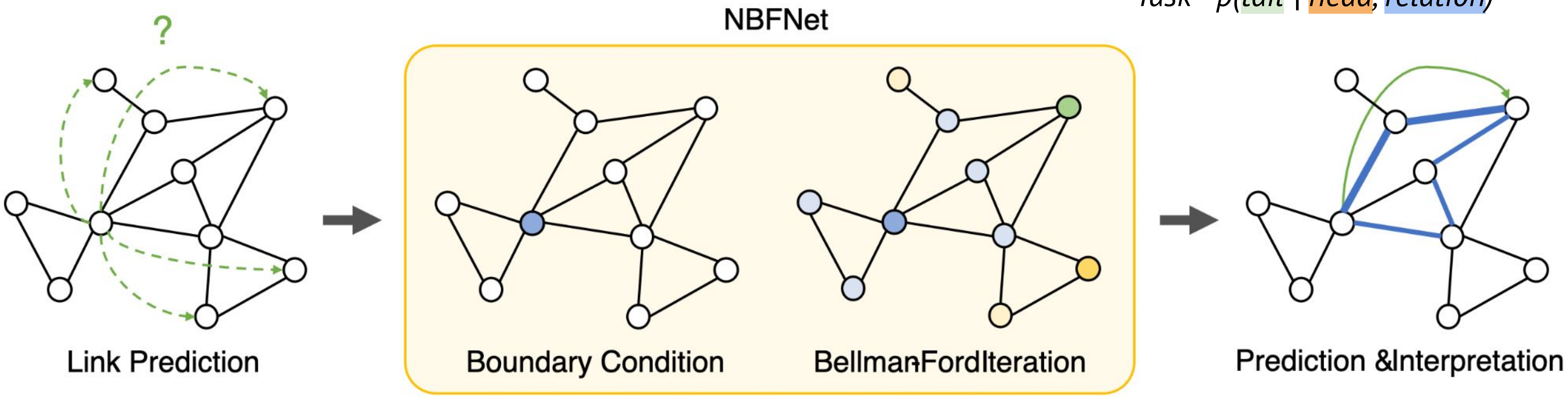


Figure 2: When we predict (v_1, v_2), we will label these two nodes differently from the rest, so that a GNN is aware of the target link when learning v_1 and v_2's representations. Similarly, when predicting (v_1, v_3), nodes v_1 and v_3 will be labeled differently. This way, the representation of v_2 in the left graph will be different from that of v_3 in the right graph, enabling GNNs to distinguish the non-isomorphic links (v_1, v_2) and (v_1, v_3).

Zhang et al. Labeling Trick: A Theory of Using Graph Neural Networks for Multi-Node Representation Learning. NeurIPS 2021

Neural Bellman-Ford

Task - p(tail | head, relation)



Idea:

1. Relations do not change at inference -> we can learn relation (edge type) embeddings
2. Initialize head node feature with the learnable relation vector (query)
3. Propage for L layers, take final representations as final node features

Zhu et al. Neural Bellman-Ford Networks: A General Graph Neural Network Framework for Link Prediction. NeurIPS 2021

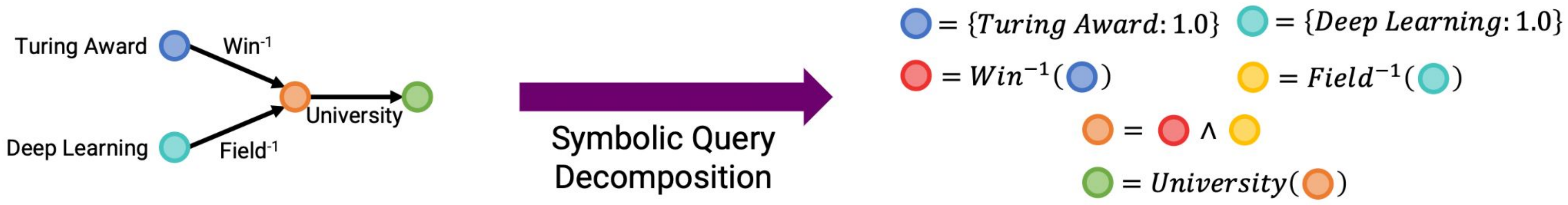
Neural Bellman-Ford

Table 4: Homogeneous graph link prediction results. Results of VGAE and S-VGAE are taken from their original papers [32, 12].

Class	Method	Cora		Citeseer		PubMed	
		AUROC	AP	AUROC	AP	AUROC	AP
Path-based	Katz Index [30]	0.834	0.889	0.768	0.810	0.757	0.856
	Personalized PageRank [42]	0.845	0.899	0.762	0.814	0.763	0.860
	SimRank [28]	0.838	0.888	0.755	0.805	0.743	0.829
Embeddings	DeepWalk [43]	0.831	0.850	0.805	0.836	0.844	0.841
	LINE [53]	0.844	0.876	0.791	0.826	0.849	0.888
	node2vec [17]	0.872	0.879	0.838	0.868	0.891	0.914
GNNs	VGAE [32]	0.914	0.926	0.908	0.920	0.944	0.947
	S-VGAE [12]	0.941	0.941	**0.947**	**0.952**	0.960	0.960
	SEAL [73]	0.933	0.942	0.905	0.924	0.978	0.979
	TLC-GNN [67]	0.934	0.931	0.909	0.916	0.970	0.968
	NBFNet	**0.956**	**0.962**	0.923	0.936	**0.983**	**0.982**

Zhu et al. Neural Bellman-Ford Networks: A General Graph Neural Network Framework for Link Prediction. NeurIPS 2021

GNN-QE: NBFNet + Multi-hop Reasoning



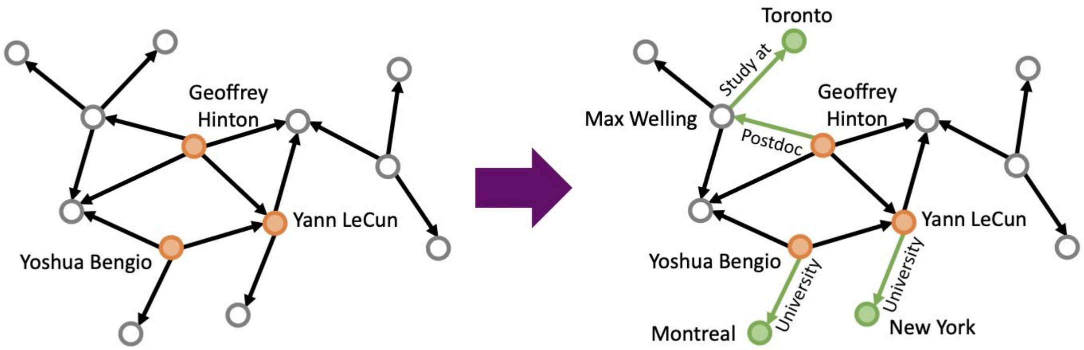
Each variable is a fuzzy set of entities, where each element in the set has a probability.

- Each relation projection (simple link prediction) step is modelled by a L-layer NBFNet
- NBFNet returns a probability distribution (scalars) over all entities (fuzzy set)

Zhu et al. Neural-Symbolic Models for Logical Queries on Knowledge Graphs. ICML 2022

GNN-QE: NBFNet + Multi-hop Reasoning

Neural Relation Projection $University(\bullet)$

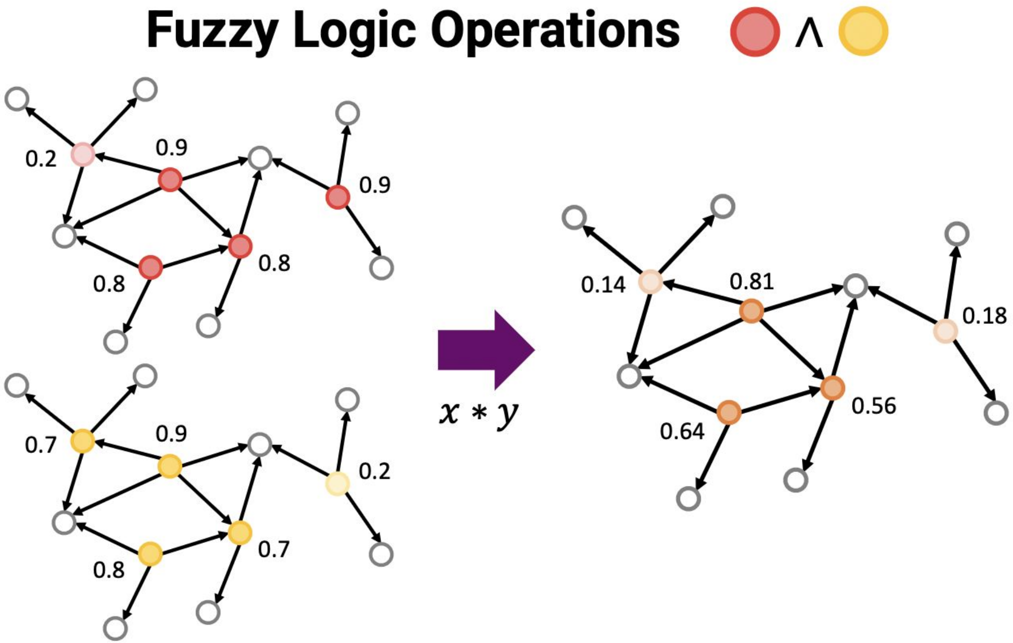


Use a **GNN** to propagate the input fuzzy set and get the output fuzzy set.

The relation $University$ is used to guide the propagation towards paths that can predict the relation $University$.

Zhu et al. Neural-Symbolic Models for Logical Queries on Knowledge Graphs. ICML 2022

GNN-QE: NBFNet + Multi-hop Reasoning



Use **product fuzzy logic** to model logic operations (e.g. ¬, ∧, ∨) over fuzzy sets.

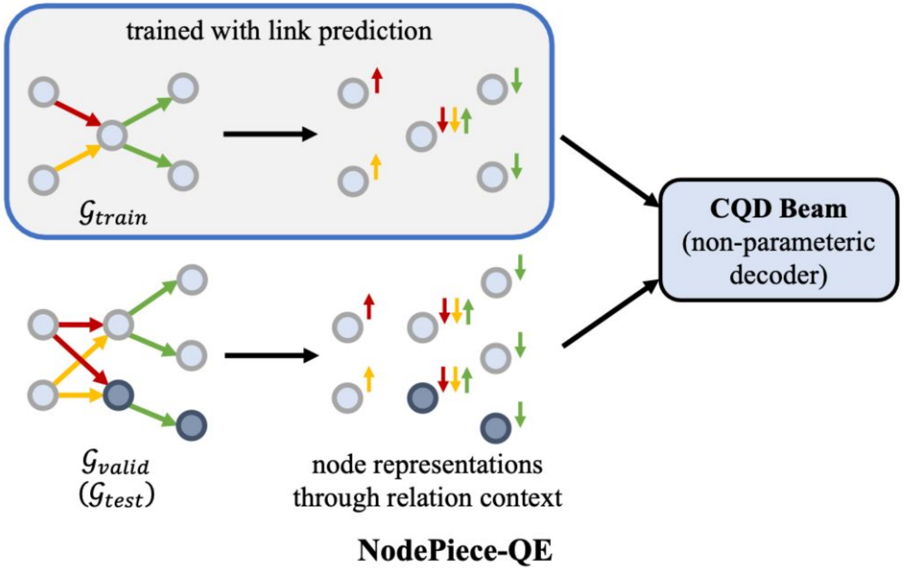
1. Logical operators as algebraic operations
2. Resulting probability **distribution** is used as **scalar weights** for the next hop graph initialization

Zhu et al. Neural-Symbolic Models for Logical Queries on Knowledge Graphs. ICML 2022

The Essence of Inductiveness

What is **invariant** in inductive reasoning setups? **Relation types**

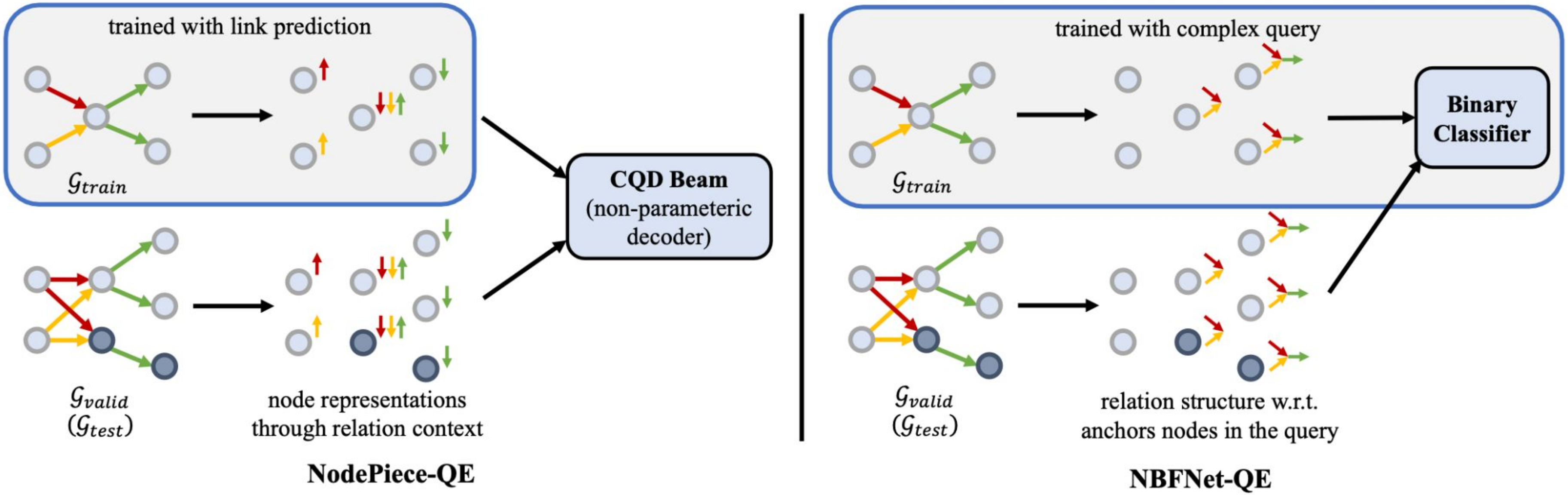
- NodePiece - parameterization through relational context



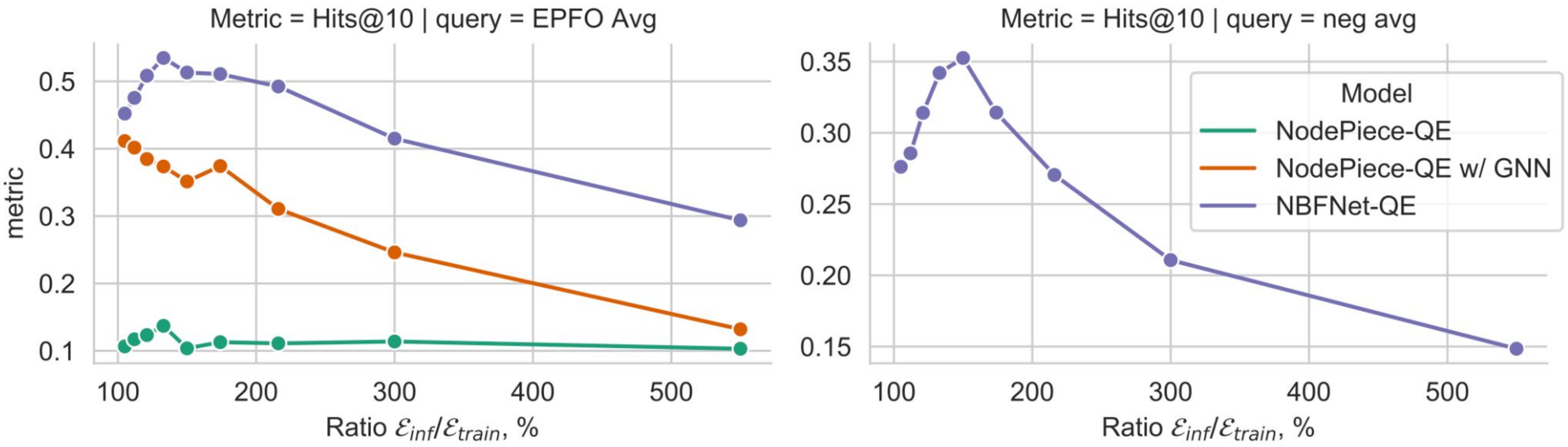
The Essence of Inductiveness

What is **invariant** in inductive reasoning setups? **Relation types**

- NodePiece - parameterization through relational context
- GNN-QE - parameterization through relational structure



Inductive Generalization to Larger Test Graphs is Still a Problem



Plan

→ Graph Reasoning Tasks

→ Featurization via Tokenization: NodePiece

→ Featurization via Labeling Trick:
Neural Bellman-Ford and GNN-QE

→ **Past, Today, Future**

Space of KG Tasks in 2019

Transductive

Triples

SETTING

TASK

Link prediction

Space of KG Tasks Today

Transductive	Triples	Supervised	Unimodal	Small
Inductive	Hyper-relational	Unsupervised	Multimodal	Large (sampling)

SETTING

TASK

Link prediction

Node classification

Entity Matching

Query Embedding

Theoretical Understanding

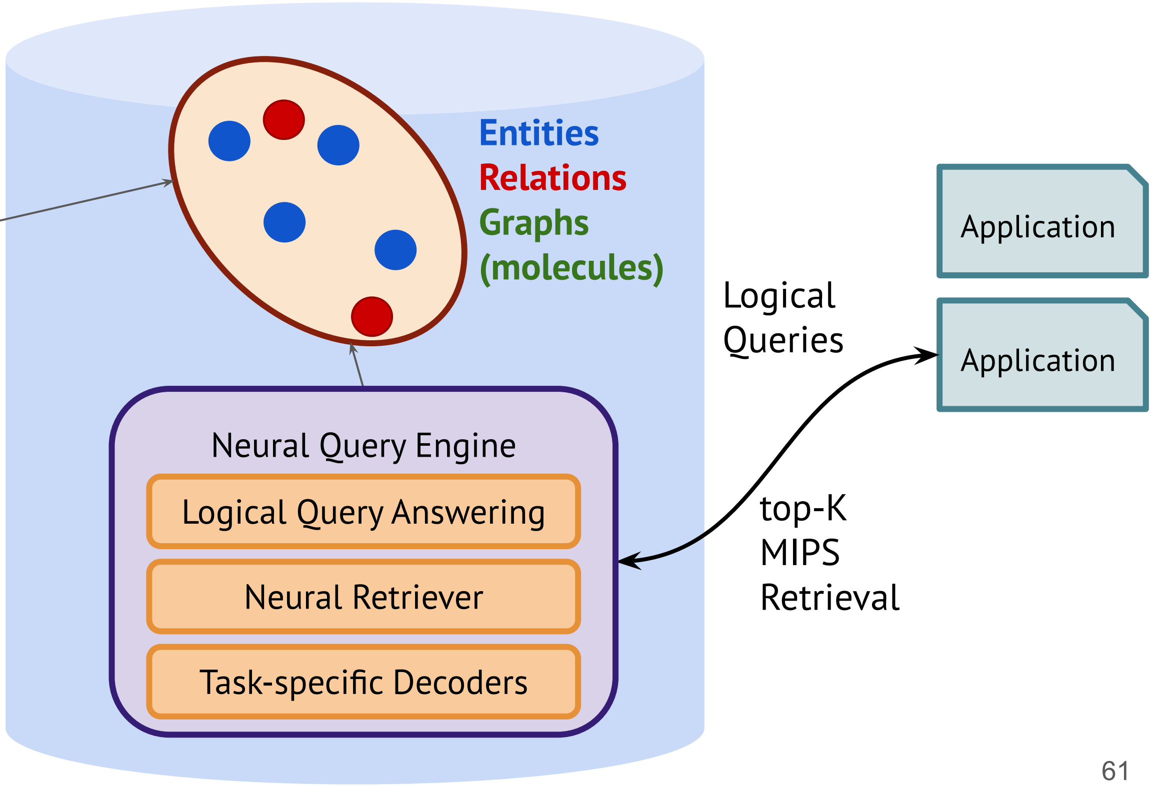
Graph Encoder

Knowledge Graph

Future: Neural Graph Databases

```
(Montreal, location, Quebec)
(Quebec, location, Canada)
(Canada, bordersWith, USA)
```

- No symbolic storage
- Embedding-based storage
- Inferring Missing Links
- Complex Query Answering
- Updatable



Q&A Time!

@michael_galkin

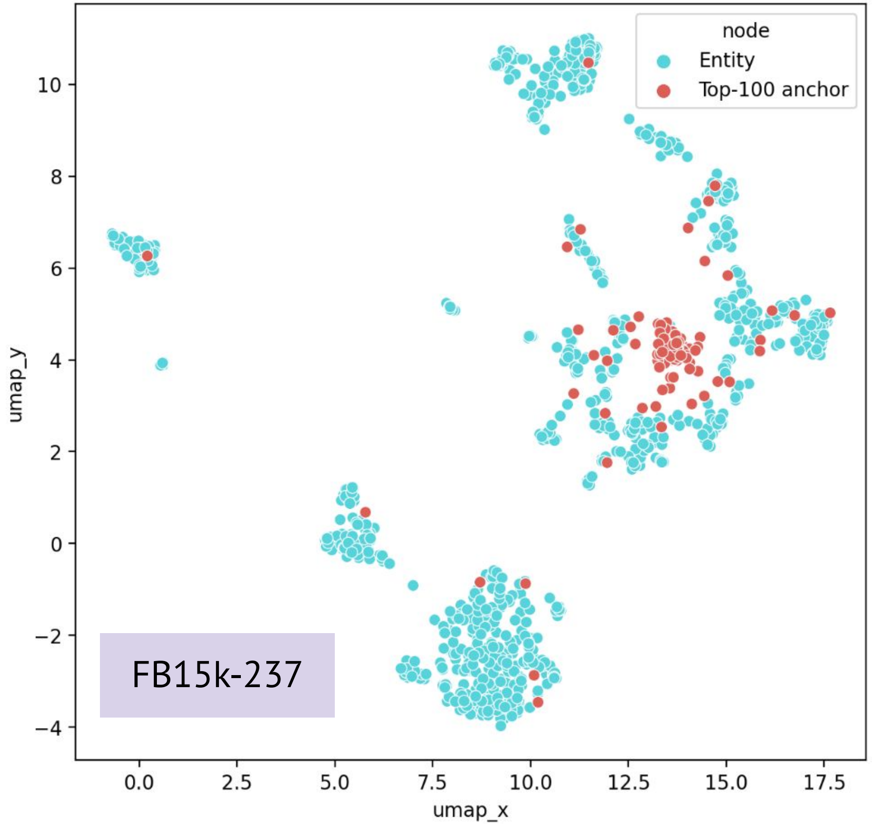
@mgalkin

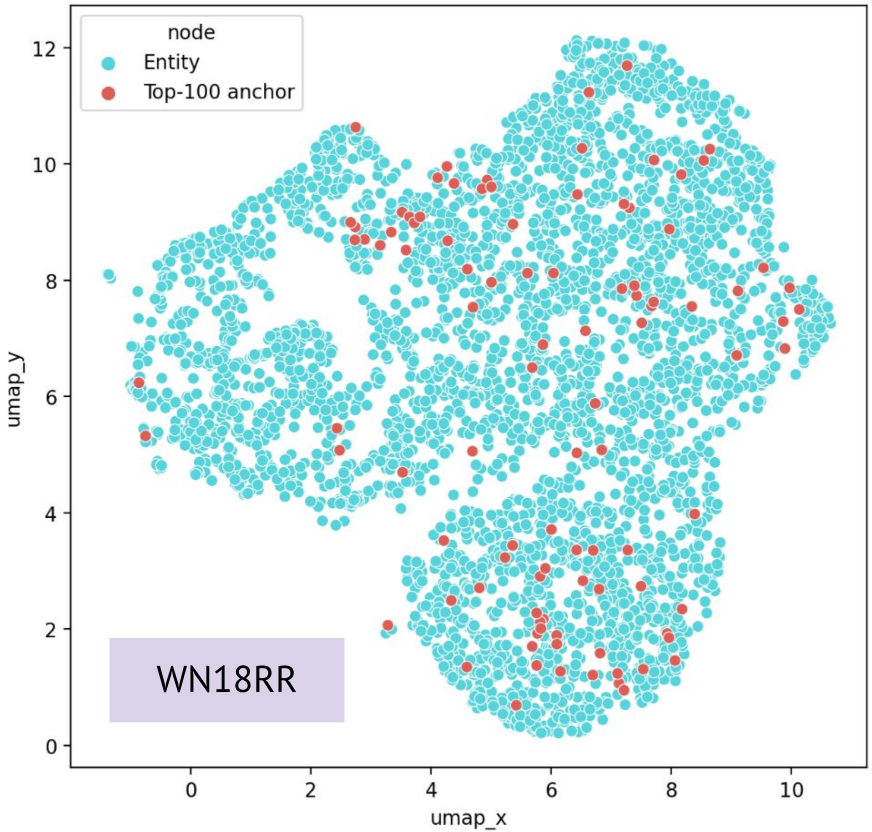
mikhail.galkin@mila.quebec

migalkin.github.io

Backup

Visualizations: Anchors + Entities





Galkin et al. NodePiece: Compositional and Parameter-Efficient Representations of Large Knowledge Graphs. ICLR 2022

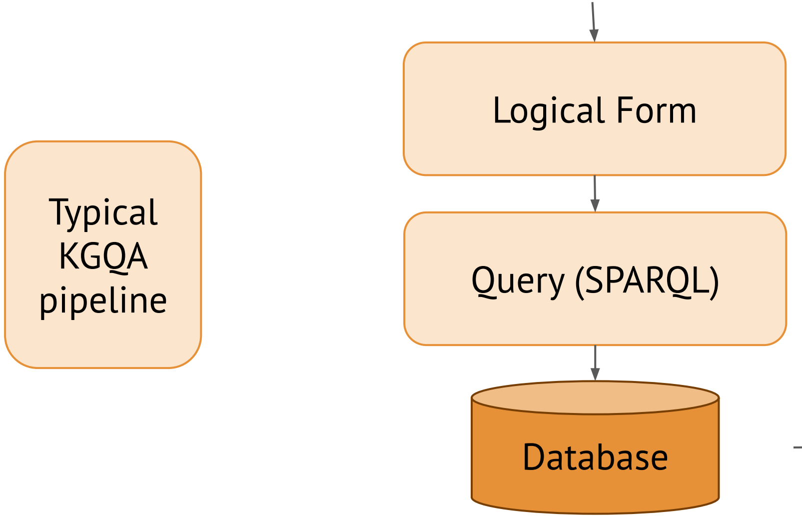
Complex Logical Query Answering: Why?

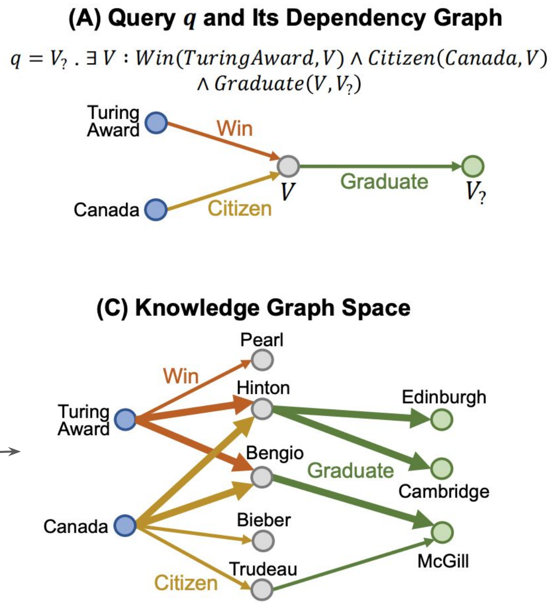
Transductive | **Triples**

Inductive | **Hyper-relational**

Query Embedding

Where did Canadian citizens with Turing Award graduate?





Ren et al. Query2box: Reasoning over Knowledge Graphs in Vector Space Using Box Embeddings. ICLR 2020

Complex Logical Query Answering: Why?

Transductive | Triples

Inductive | Hyper-relational

Query Embedding

Where did Canadian citizens with Turing Award graduate?

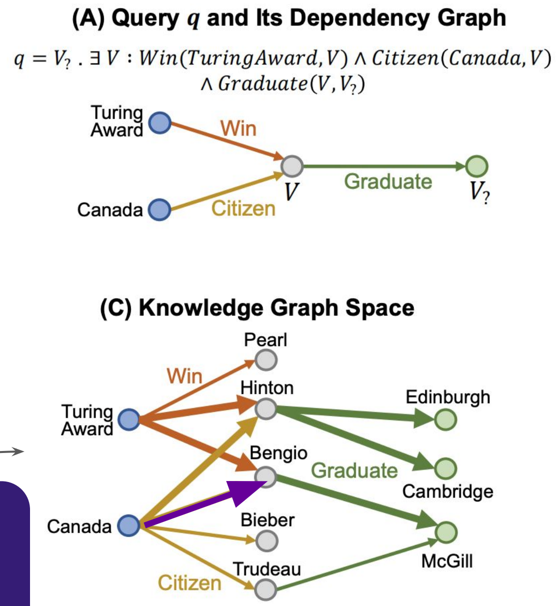
Logical Form

Query (SPARQL)

Database

Typical KGQA pipeline

What if this edge is missing?



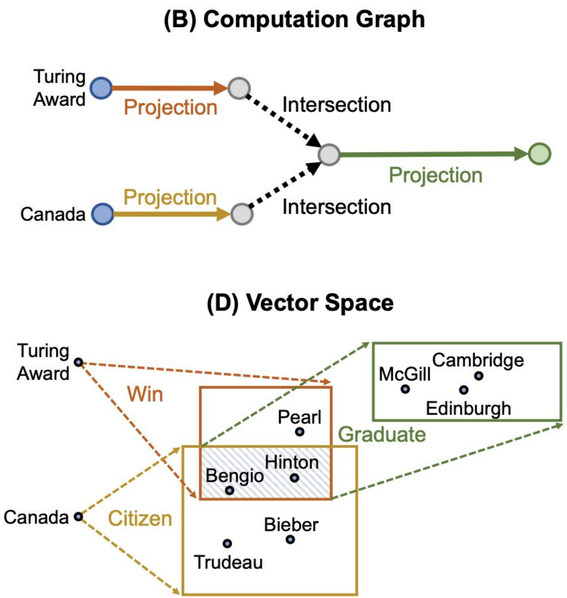
Ren et al. Query2box: Reasoning over Knowledge Graphs in Vector Space Using Box Embeddings. ICLR 2020

Complex Logical Query Answering

Transductive	Triples
Inductive	Hyper-relational

Query Embedding

- Databases assume KGs are **complete**
    - In reality - they are not
- We want to answer FOL queries over **incomplete** graphs with **neural** operators
- Embed a query in a latent space, MIPS decoder for kNN answers

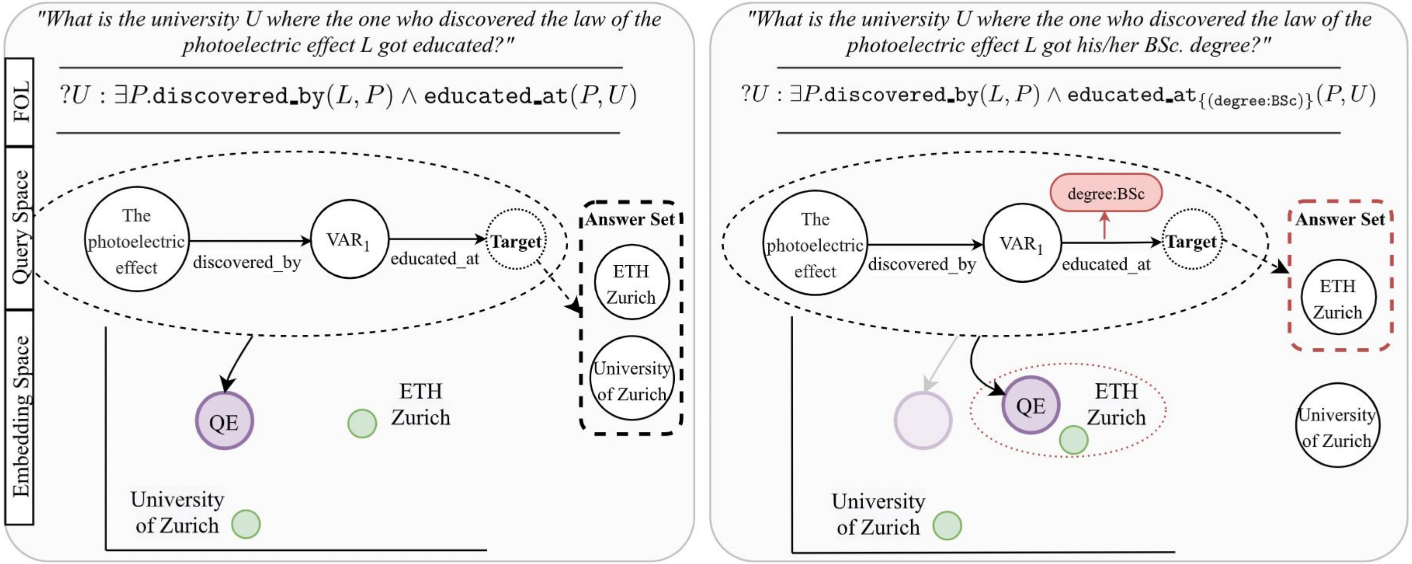


Ren et al. Query2box: Reasoning over Knowledge Graphs in Vector Space Using Box Embeddings. ICLR 2020

StarQE: Complex Logical Query Answering on HR KGs

Transductive | Triples | Query Embedding

Inductive | Hyper-relational

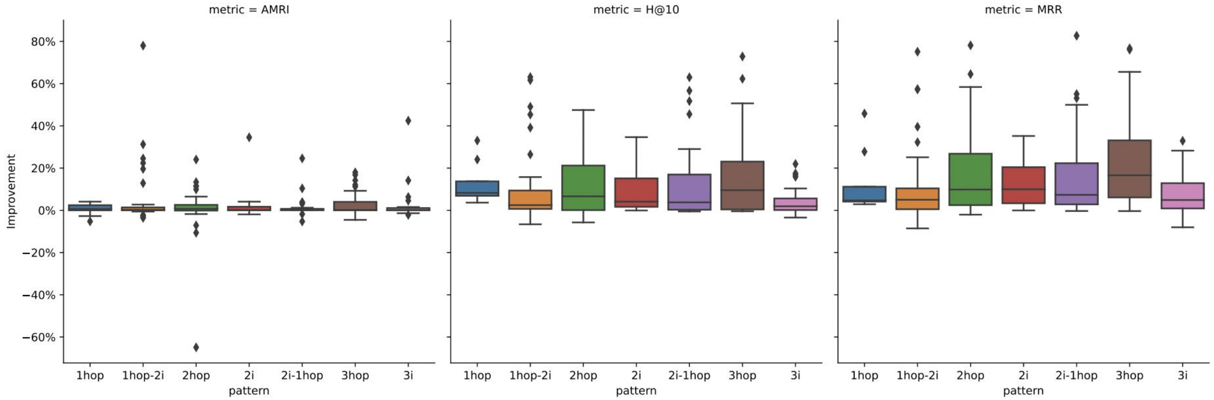


1. Extending FOL to HR KGs
2. Qualifiers help A LOT
3. New query types are enabled

Alivanistos et al. Query Embedding on Hyper-Relational Knowledge Graphs. ICLR 2022

★ StarQE for Logical Queries: Summary

- Extend FOL to hyper-relational graphs with qualifiers
- Enabling new query types (eg, joins over qualifier entities)
- Robust to inner representation: RDF* vs reified RDF
- Qualifiers help A LOT in answering complex queries



Alivanistos et al. Query Embedding on Hyper-Relational Knowledge Graphs. ICLR 2022