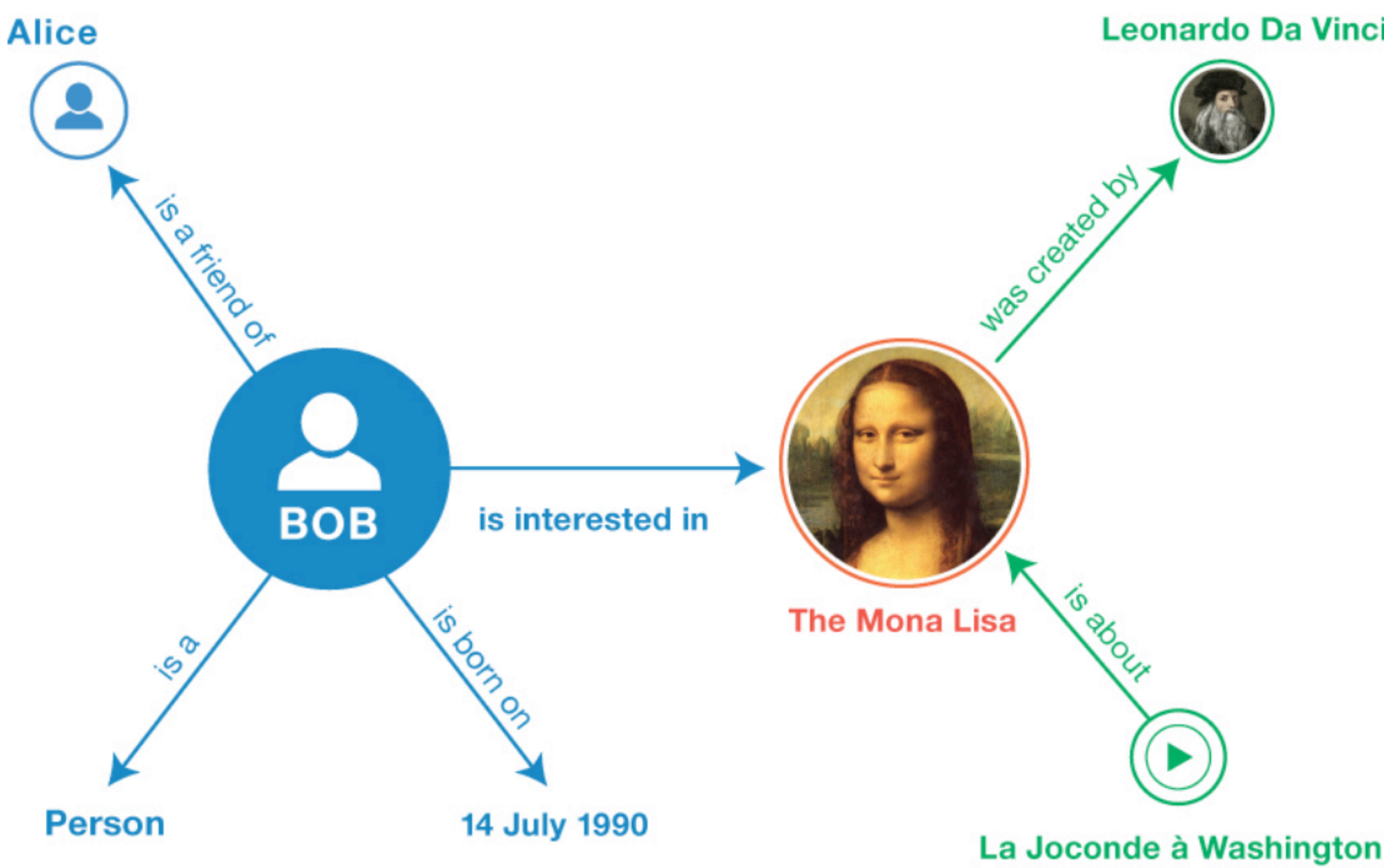
Reasoning in Knowledge Graphs using Deep Learning

Joint work with H. Ren, W. Hamilton, R. Ying, J. You, M. Zitnik, D. Jurafsky

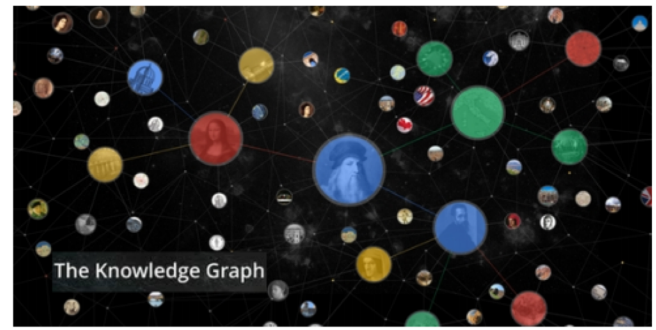
Jure Leskovec

Knowledge Graphs

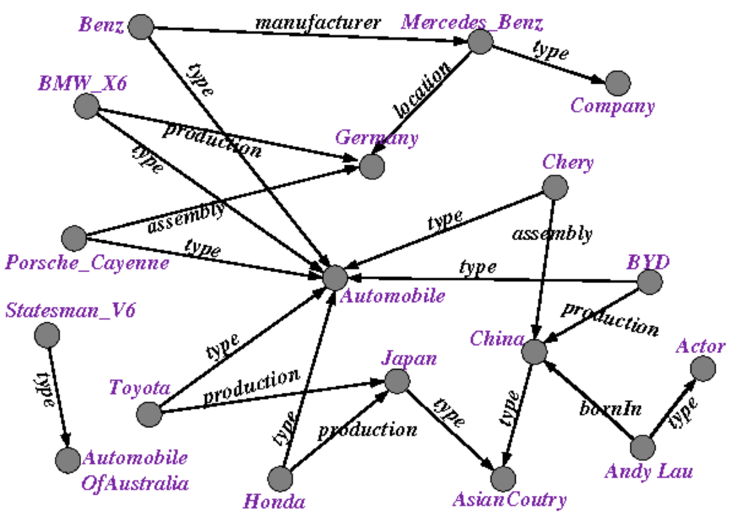


Many Domains

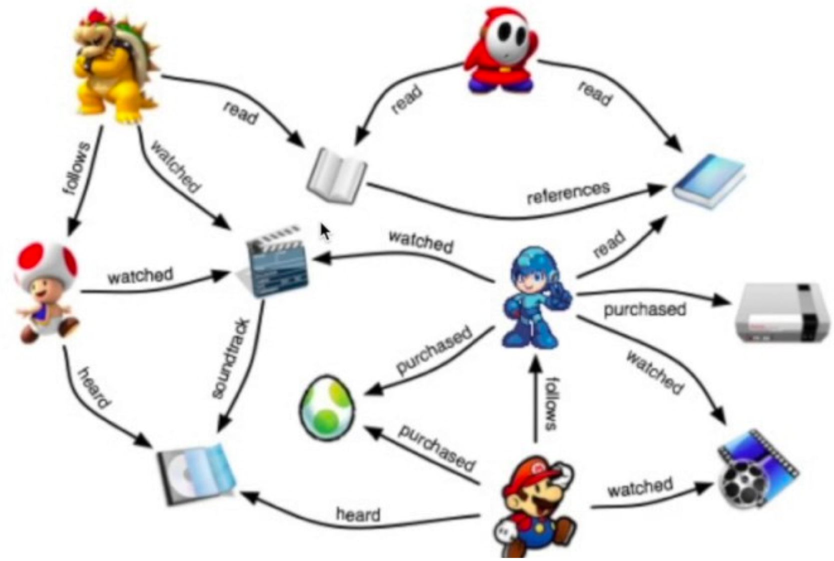
WIKIDATA



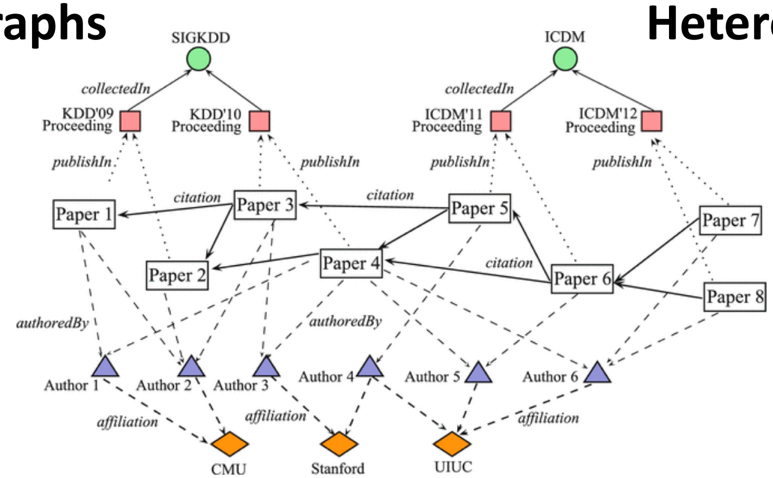
Heterogeneous Networks



Knowledge Graphs



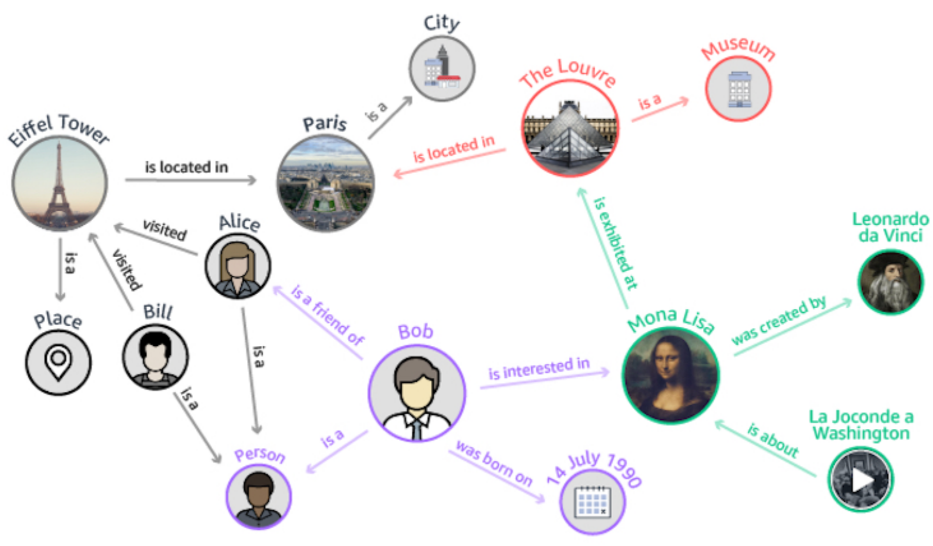
Heterogeneous Graphs



Multimodal Graphs

Knowledge Graphs

- Knowledge Graphs are heterogenous graphs
  - Multiple types of entities and relations exist
- Facts are represented as triples (h, r, t)
  - ('Alice', 'friend_with', 'Bob')
  - ('Paris', 'is_a', 'City')
  - …



Traditional Tasks

Knowledge Graph Competion/Link Prediction

- Predict the missing head or tail for a given triple (h, r, t)
- Example:

Barack Obama BornIn United States

Barack Obama Nationality American

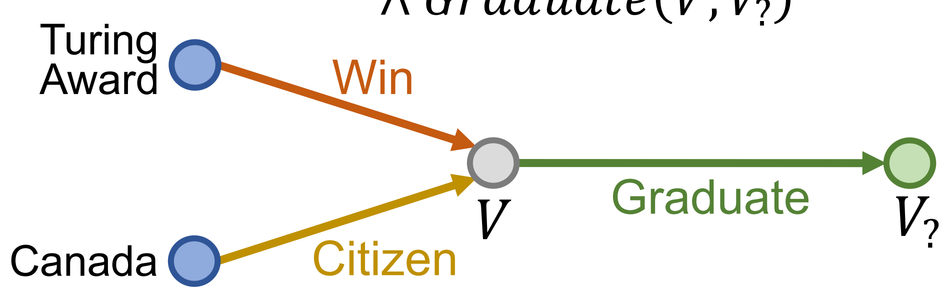
Our work: Beyond Link Prediction

Our goal: Reason over the knowledge graph using complex multi-hop queries

- Conjunctive queries: Subset of first-order logic with existential quantifier (∃) and conjunction (∧)

"Where did all Canadian citizens with Turing Award graduate?"

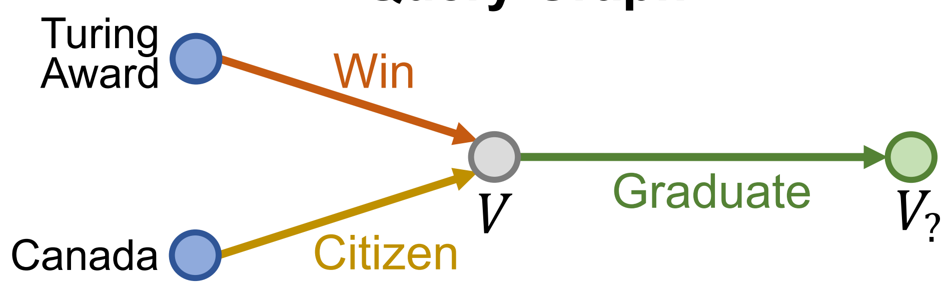
$$q = V_? \,.\, \exists\, V : Win(TuringAward, V) \wedge Citizen(Canada, V) \wedge Graduate(V, V_?)$$



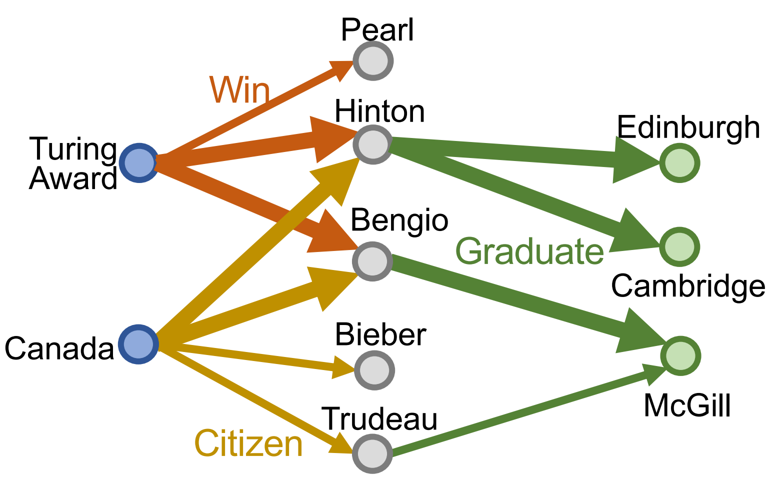
Answering Queries in KGs

“Where did Canadian citizens with Turing Award graduate?”

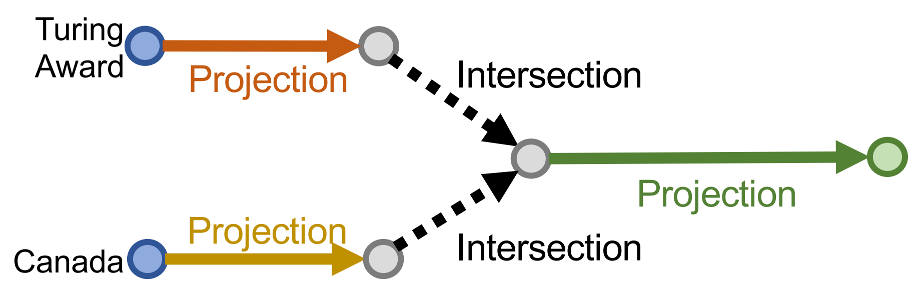
Query Graph



Knowledge Graph



Computation Graph



Each point corresponds to a set of entities

Why is it Hard?

- Heterogeneity: Lack of schema, or quite large schema (65K for DBpedia)
- Noise and incompleteness
- Uncertainty
- Massive size
- Fast query time

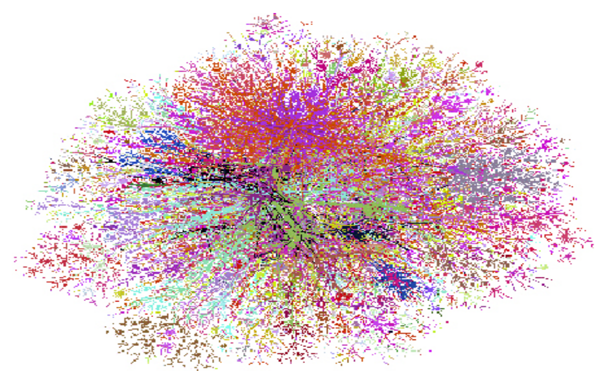
PubID	Publisher	PubAddress
03-4472822	Random House	123 4th Street, New York
04-7733903	Wiley and Sons	45 Lincoln Blvd, Chicago
03-4859223	O'Reilly Press	77 Boston Ave, Cambridge
03-3920886	City Lights Books	99 Market, San Francisco

AuthorID	AuthorName	AuthorBDay
345-28-2938	Haile Selassie	14-Aug-92
392-48-9965	Joe Blow	14-Mar-15
454-22-4012	Sally Hemmings	12-Sept-70
663-59-1254	Hannah Arendt	12-Mar-06

ISBN	AuthorID	PubID	Date	Title
1-34532-482-1	345-28-2938	03-4472822	1990	Cold Fusion for Dummies
1-38482-995-1	392-48-9965	04-7733903	1985	Macrame and Straw Tying
2-35921-499-4	454-22-4012	03-4859223	1952	Fluid Dynamics of Aquaducts
1-38278-293-4	663-59-1254	03-3920886	1967	Beads, Baskets & Revolution

Relational Data (Structured)

vs.

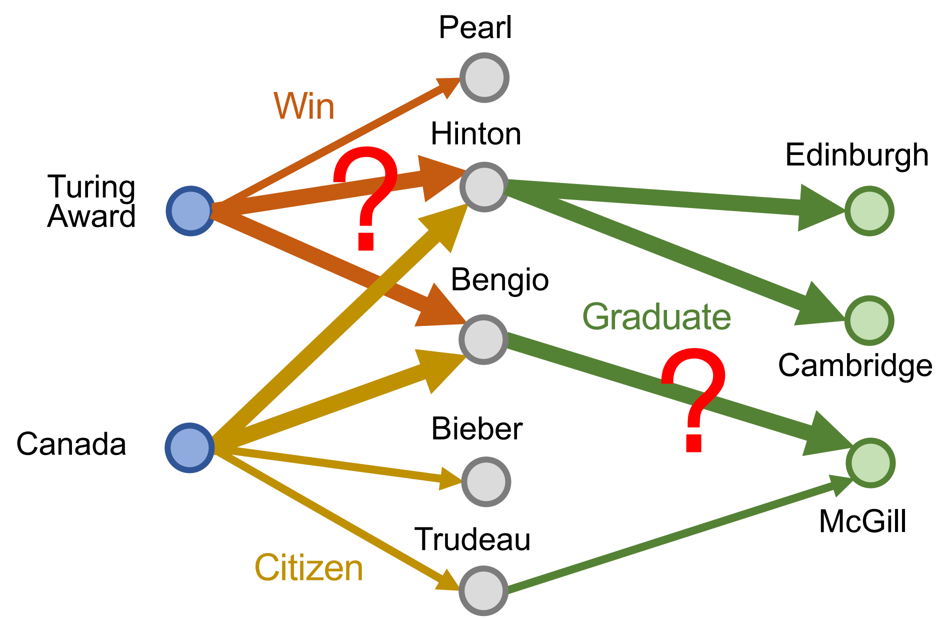


Heterogeneous Graph Data (Semi-structured)

Why is it Hard?

Key challenge: Big graphs and queries can involve noisy and unobserved data!

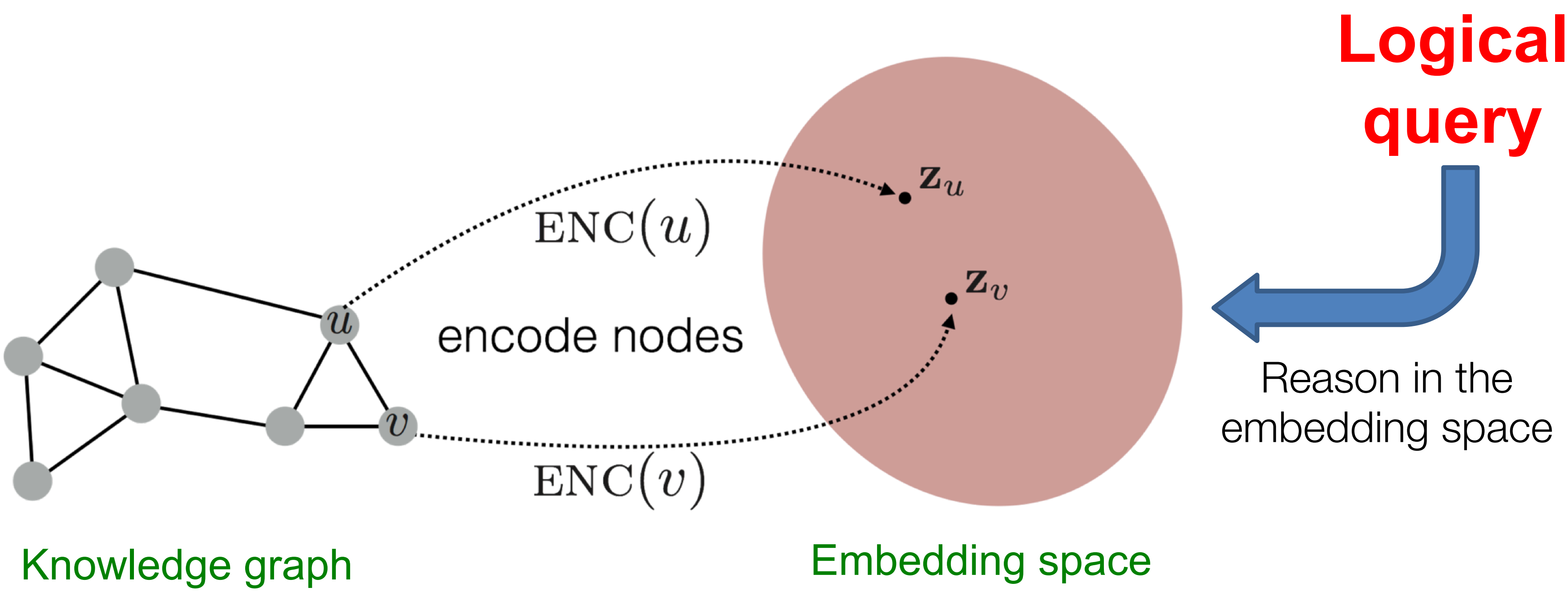
Some links might be noisy or missing



Problem: Naïve link prediction and graph template matching are too expensive

Our Idea: **Query2Box**

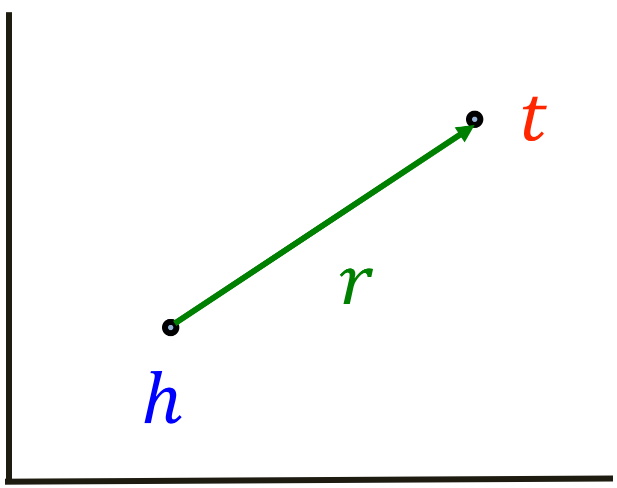
Use representation learning to map a graph into a Euclidean space and learn to reason in that space

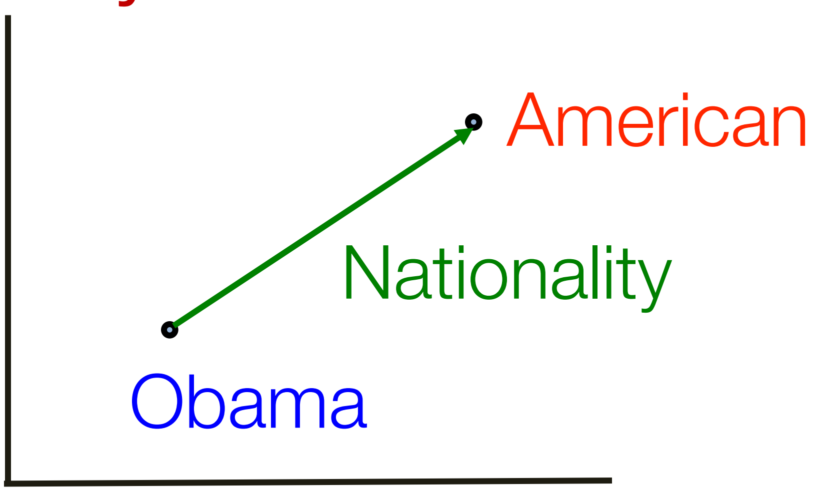


Semantic Embeddings

TransE [Bordes et al., 2013]

- Translation intuition:
  For a triple (h, r, t): $h + r \approx t$
- **Q:** What is nationality of Obama?





Our Idea: **Query2Box**

Idea:

- **1)** Embed nodes of the graph
- **2)** For every logical operator learn a spatial operator

So that:

- **1)** Take an arbitrary logical query. Decompose it into a set of logical operators (∃,∧,∨)
- **2)** Apply a sequence of spatial operators to embed the query
- **3)** Answers to the query are entities close to the embedding of the query

Our Idea: **Query2Box**

Idea:

- **1)** Embed nodes of the graph

Key insight:

Represent query as a box.

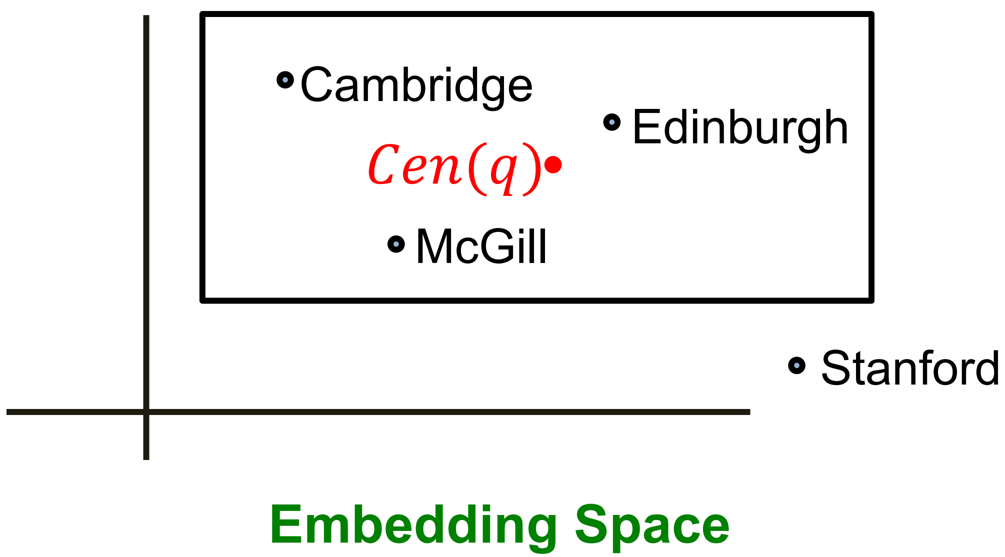
Operations (union, intersection) are well defined over boxes.

- **3)** Answers to the query are entities close to the embedding of the query

Embedding Queries

Query2Box embedding:

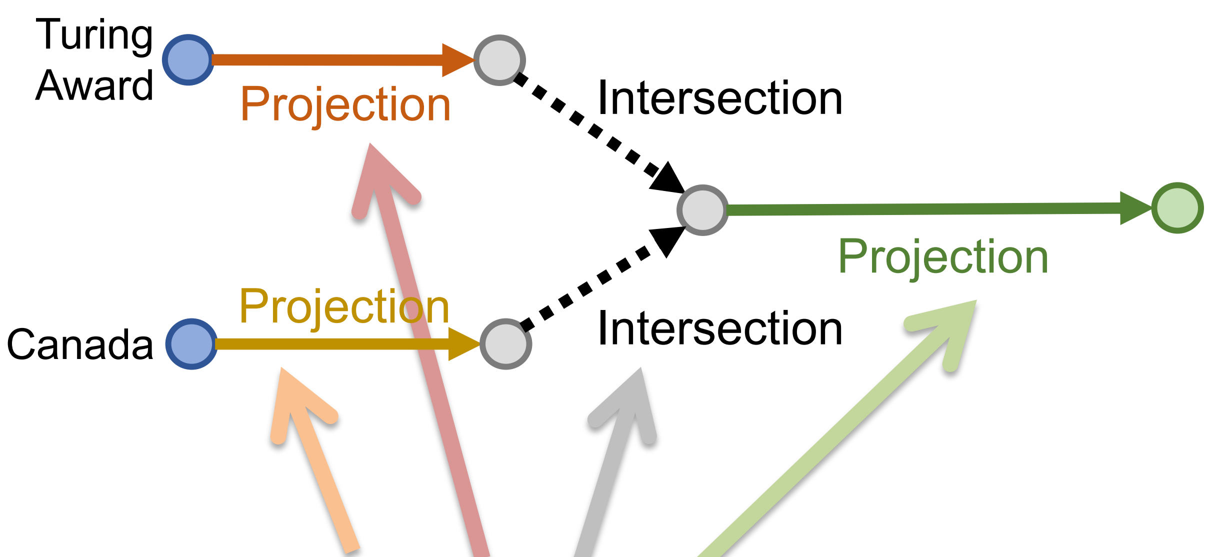
Embed queries with hyper-rectangles (boxes): $\mathbf{q} = (Cen(q), Off(q))$.



[Probabilistic Embedding of Knowledge Graphs with Box Lattice Measures. Vilnis, et al., ACL 2018]

Embedding Queries

Computation Graph

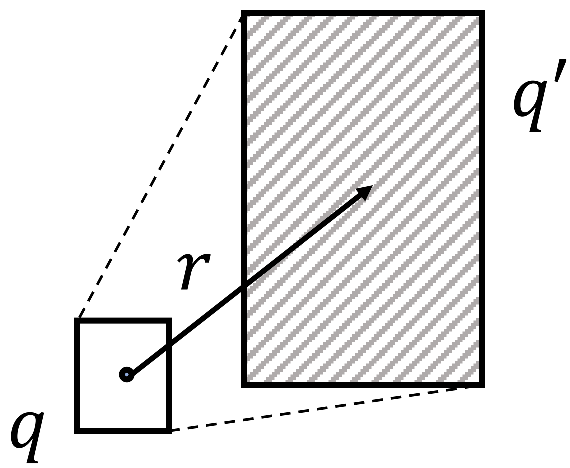


- Geometric Projection Operator
- Geometric Intersection Operator

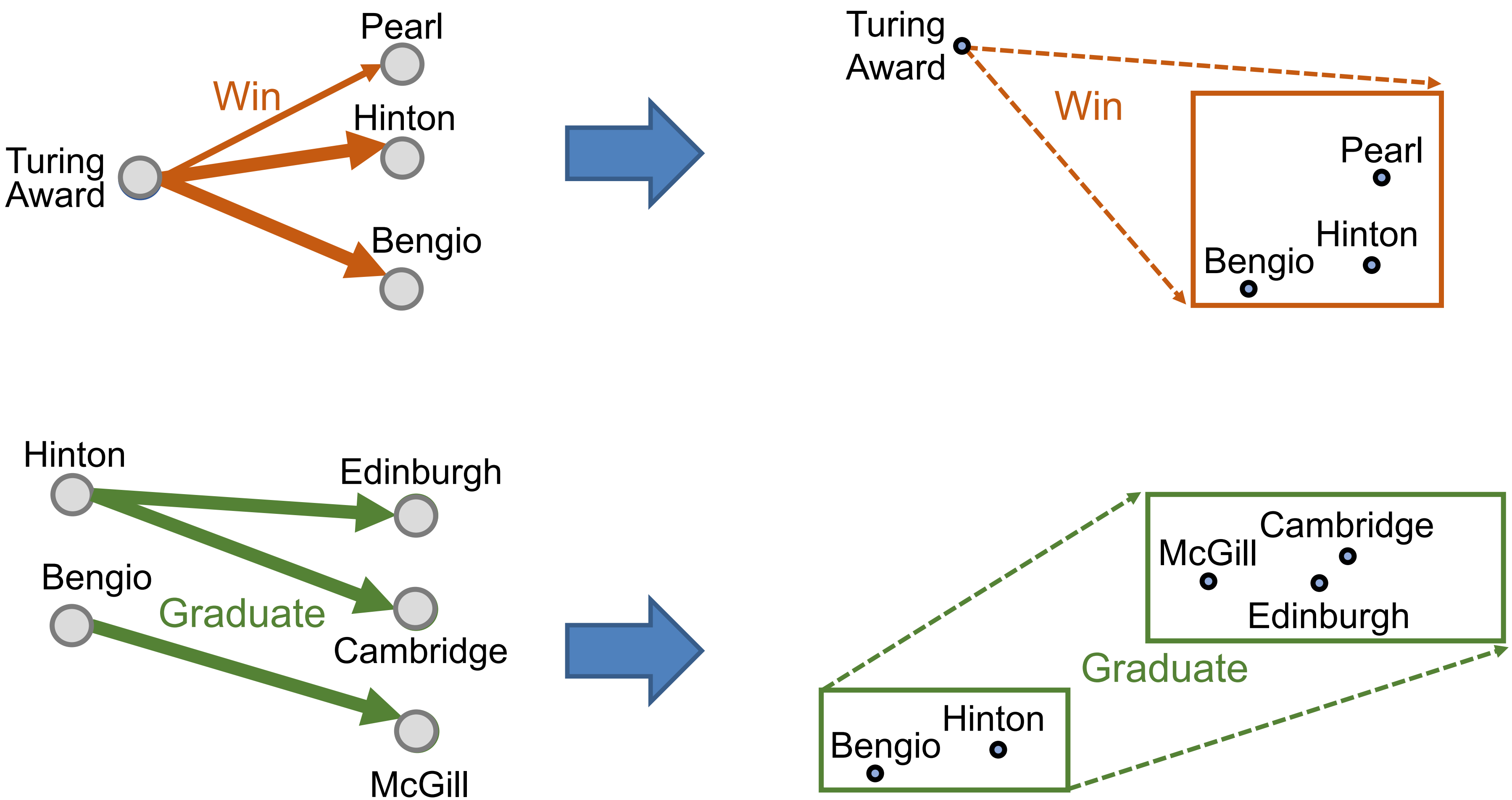
Projection Operator

Geometric Projection Operator $\boldsymbol{\mathcal{P}}$

- $\mathcal{P}$: Box × Relation → Box



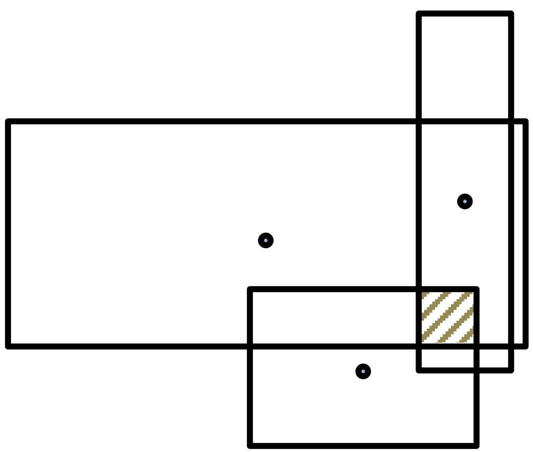
Projection Operator: Example



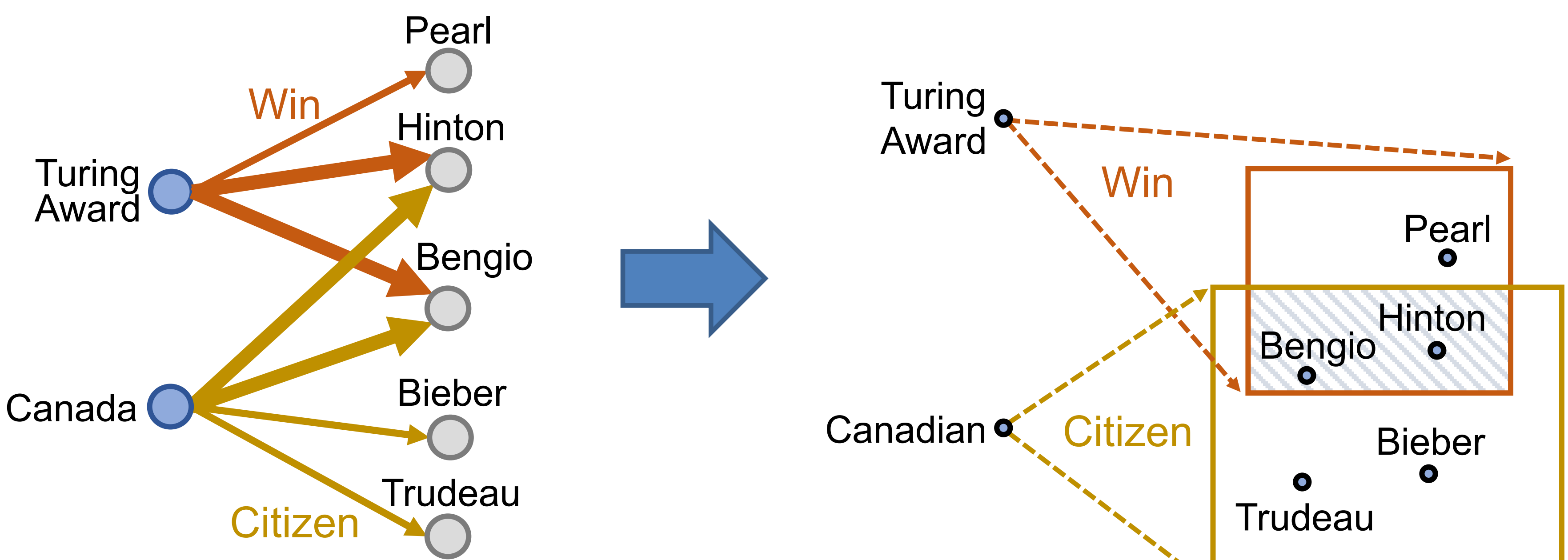
Intersection Operator

Geometric Intersection Operator $\mathcal{J}$

- $\mathcal{J}$: Box $\times \cdots \times$ Box $\rightarrow$ Box
  - The new center is a weighted average
  - The new offset shrinks



Intersection Operator: Example



How to Handle Disjunction

So far we can handle Conjunctive queries

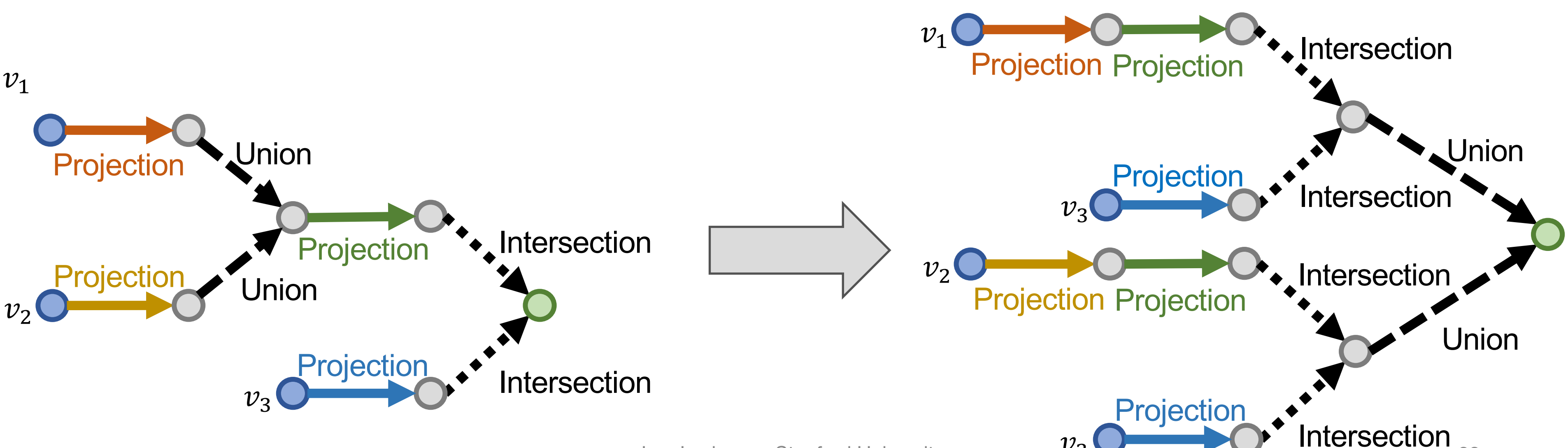
Can we learn a geometric disjunction operator?

- Theorem (paraphrased): For a KG with M nodes, we need embedding dimension of M to handle disjunction

Disjunctive Normal Form

- Any query with AND and OR can be transformed into equivalent Disjunctive Normal Form (disjunction of conjunctive queries).

Original Computation Graph **Converted Computation Graph**



Disjunctions: Solution

Given an arbitrary AND-OR query

- 1) Transform it into an DNF
- 2) Answer each conjunctive query
- 3) Overall answer is the union of conjunctive query answers

Benefits of **Query2Box**

Scalability and efficiency:

- Any query can be reduced to a couple of matrix operations and a single k-nearest neighbor search

Generality:

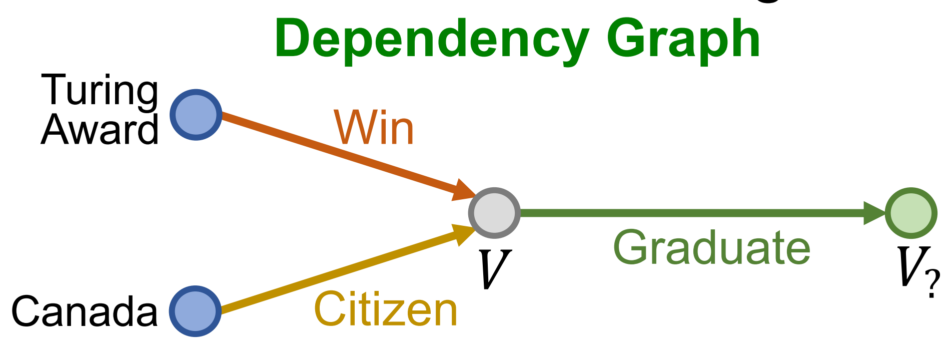
- We can answer any query (even those we have never seen before)

Robustness to noise:

- Graph can contain missing and noisy relationships

Example

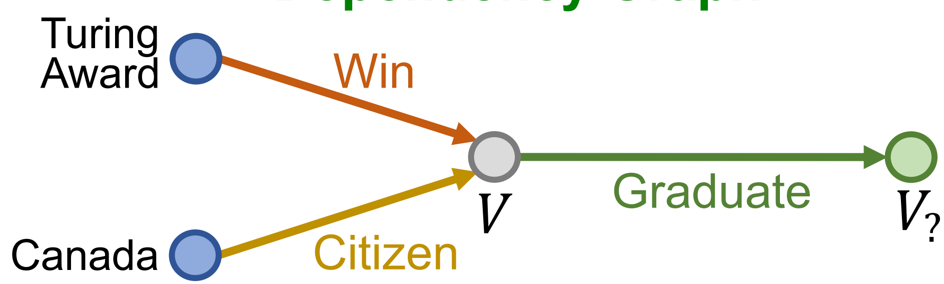
"Where did Canadian citizens with Turing Award graduate?"



Example

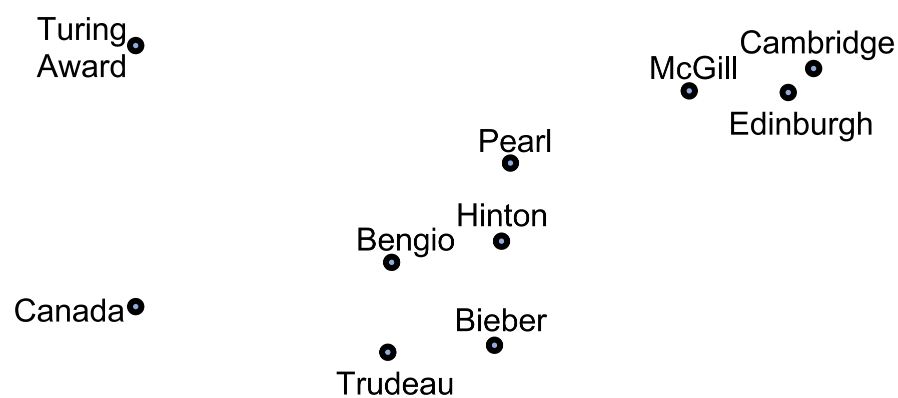
"Where did Canadian citizens with Turing Award graduate?"

Dependency Graph



Computation Graph

Embedding Process

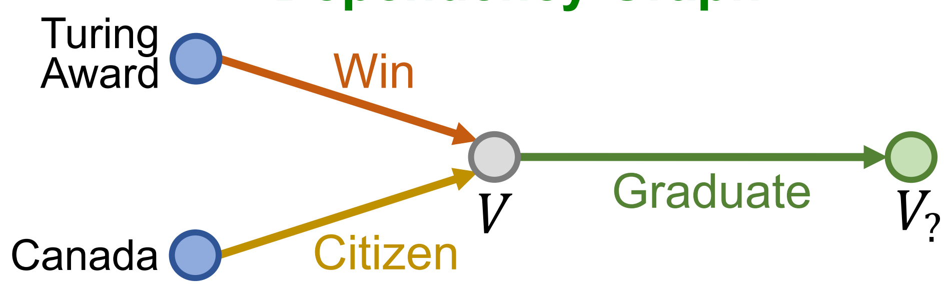


Each point corresponds to a set of entities

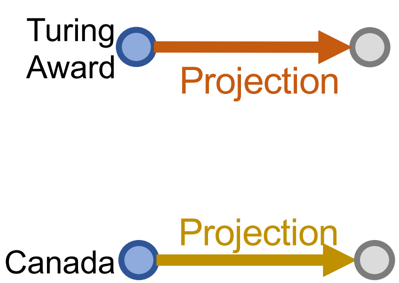
Example

"Where did Canadian citizens with Turing Award graduate?"

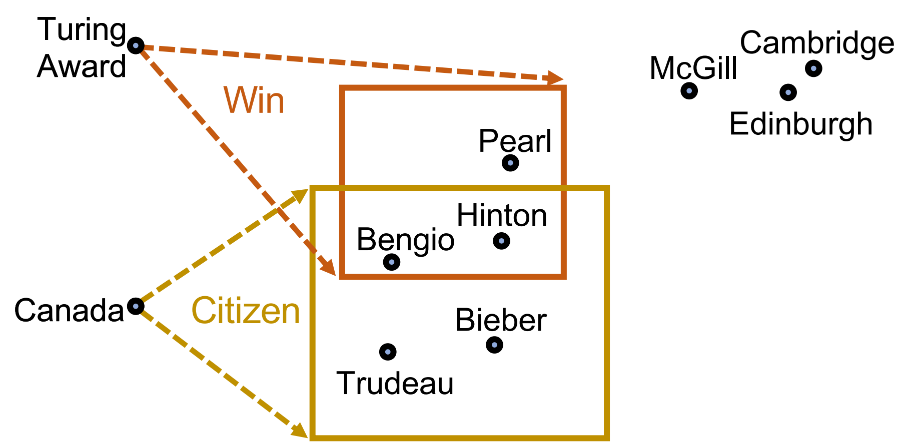
Dependency Graph



Computation Graph



Embedding Process

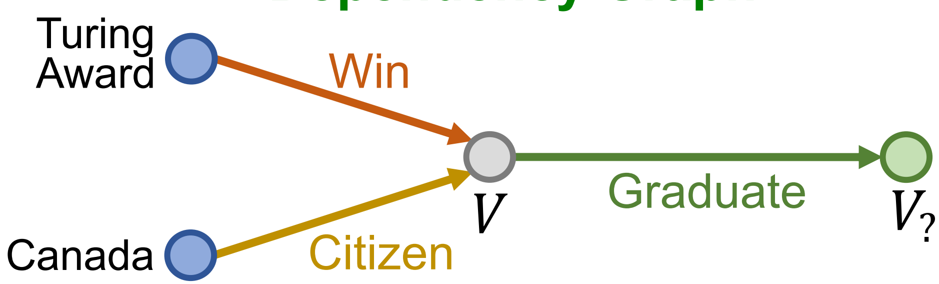


Each point corresponds to a set of entities

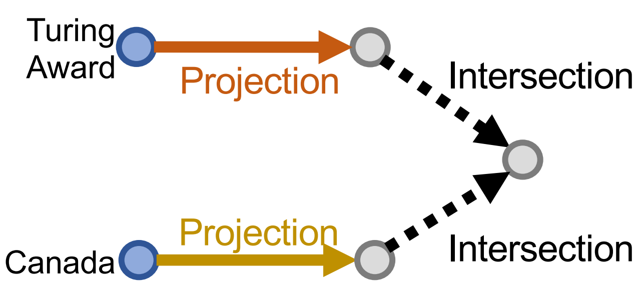
Example

"Where did Canadian citizens with Turing Award graduate?"

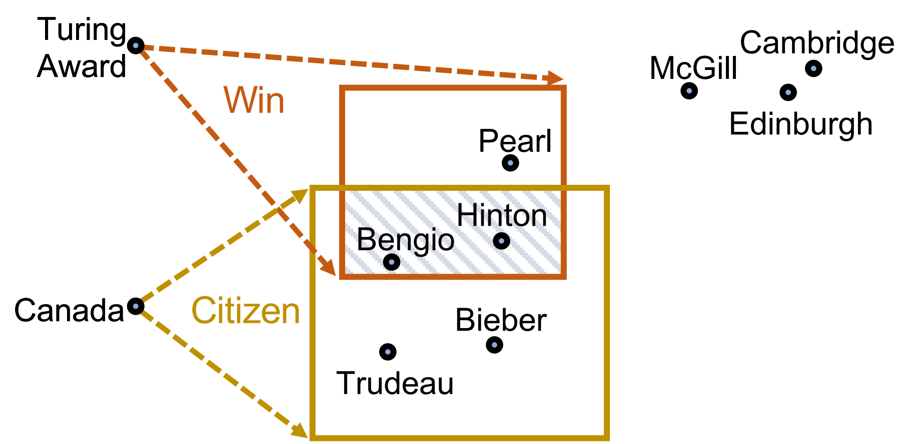
Dependency Graph



Computation Graph



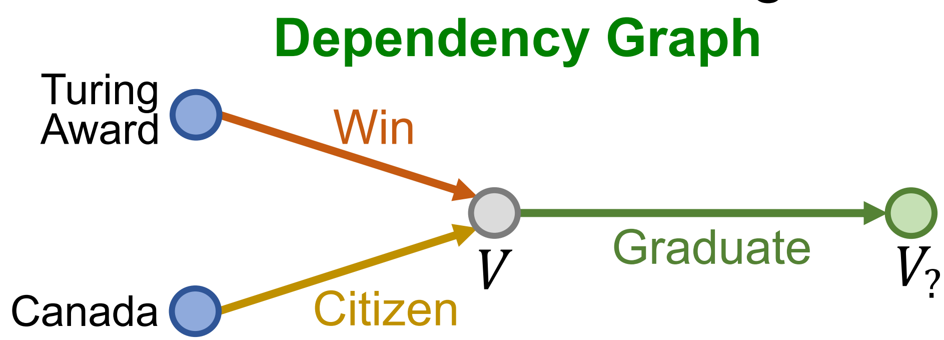
Embedding Process

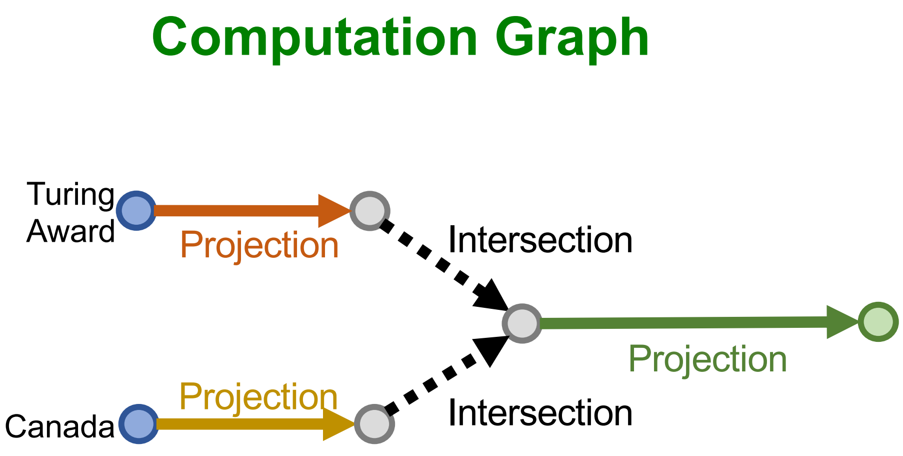


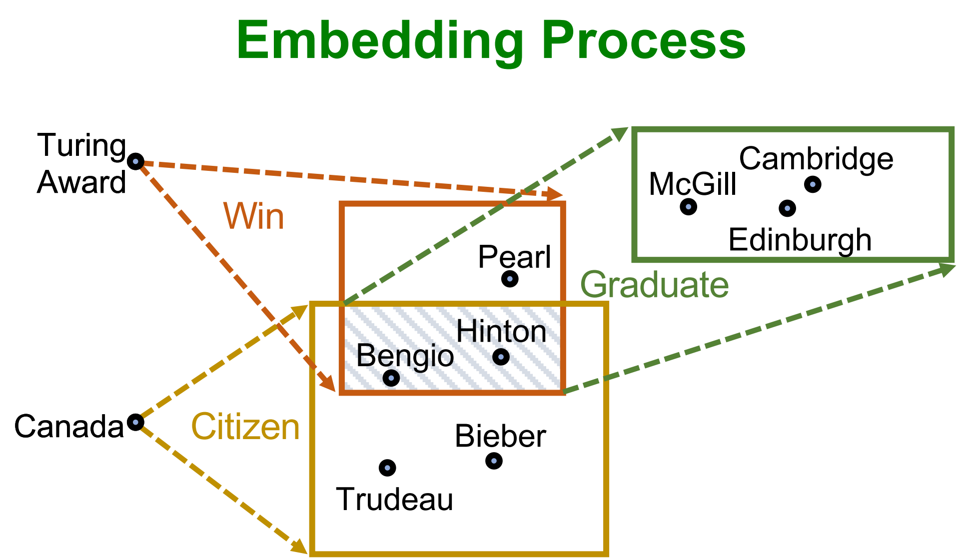
Each point corresponds to a set of entities

Example

"Where did Canadian citizens with Turing Award graduate?"



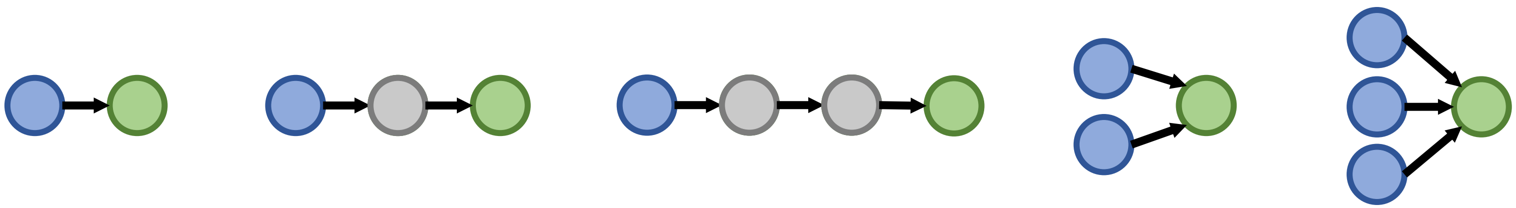




Each point corresponds to a set of entities

Query2Box : Model Training

Training examples: Queries on the graph



- Positives: Path with a known answer
- Negatives: Random nodes of the correct answer type
- **Goal:** Find embeddings and operators so that that queries give correct answers

Experimental Setup

We essentially learn to "memorize" the answers to queries

- We embed entities so that our geometric operators give correct answers

Questions:

- Does our method generalize to new unseen queries?
- Does our method generalize to new query structures?
- Can method handle missing relations?

Experimental Setup

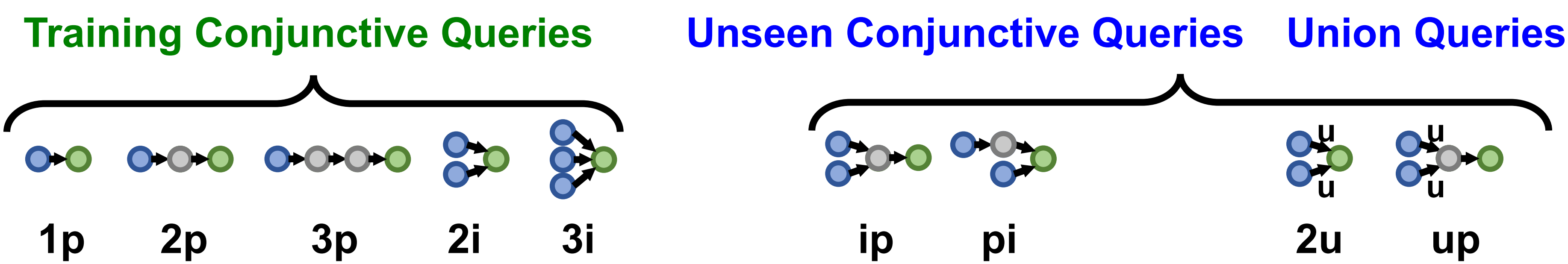
- **Training:**
  - Remove 10% of KG edges
  - Sample training queries and (non)answers
  - Train the model
- **Test set:**
  - Test queries/answers from the full graph
  - Ensure that the test queries are **not** directly answerable in the training graph
    - Every test query has at least one deleted edge
    - Note: Query template matching would have accuracy of random guessing

KG and Query Statistics

- Freebase: FB15K, FB15K-237

Dataset	Entities	Relations	Training Edges	Validation Edges	Test Edges	Total Edges
FB15k	14,951	1,345	483,142	50,000	59,071	592,213
FB15k-237	14,505	237	272,115	17,526	20,438	310,079

- Queries:



Experimental Results

Method	Avg	1p	2p	3p	2i	3i	ip	pi	2u	up
Q2B	**0.268**	**0.467**	**0.24**	**0.186**	**0.324**	**0.453**	**0.108**	**0.205**	**0.239**	**0.193**
Point-based embedding	0.228	0.402	0.213	0.155	0.292	0.406	0.083	0.170	0.169	0.163
	0.230	0.405	0.213	0.153	0.298	0.411	0.085	0.182	0.167	0.160

Table 3: H@3 on test set for QUERY2BOX vs. GQE on FB15k-237.

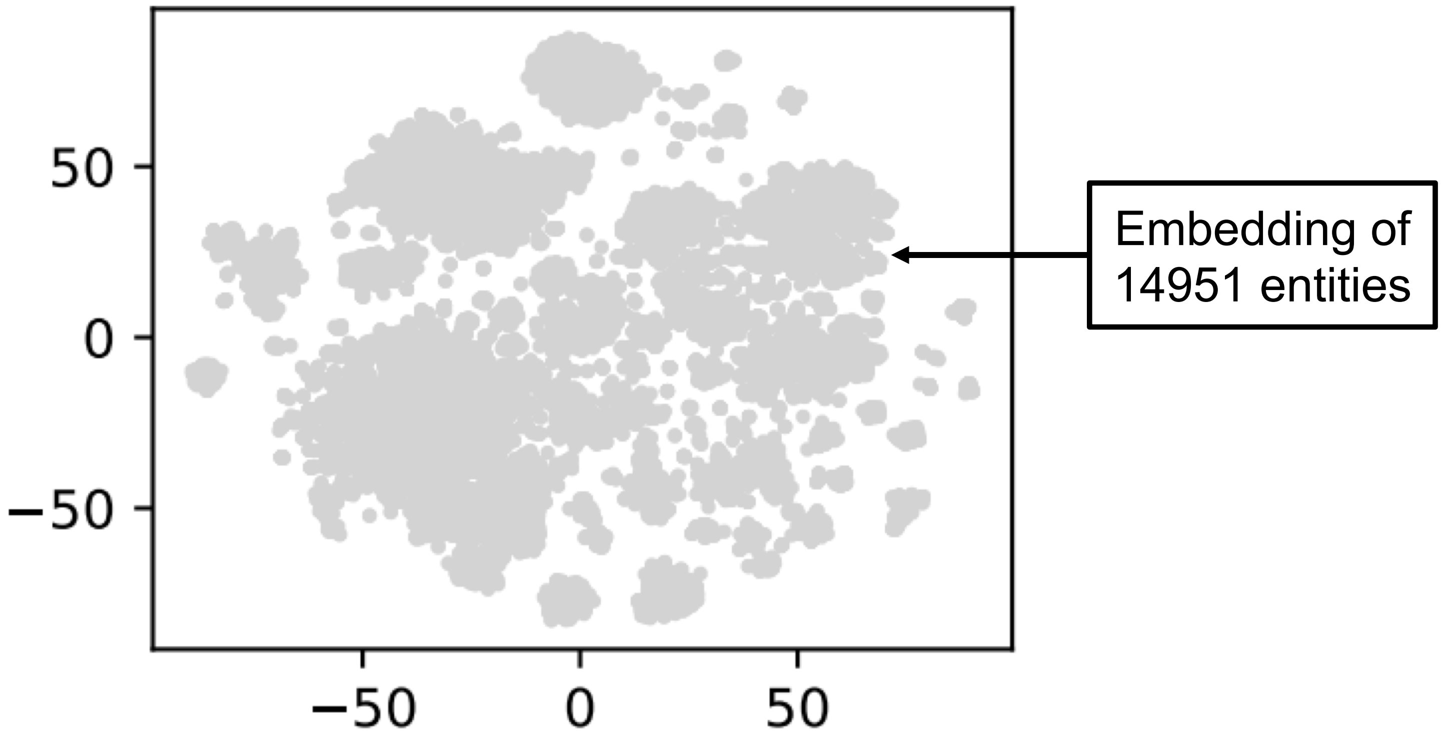
Method	Avg	1p	2p	3p	2i	3i	ip	pi	2u	up
Q2B	**0.484**	**0.786**	**0.413**	**0.303**	**0.593**	**0.712**	**0.211**	**0.397**	**0.608**	**0.330**
Point-based embedding	0.386	0.636	0.345	0.248	0.515	0.624	0.151	0.31	0.376	0.273
	0.384	0.63	0.346	0.250	0.515	0.611	0.153	0.32	0.362	0.271

Table 4: H@3 on test set for QUERY2BOX vs. GQE on FB15k.

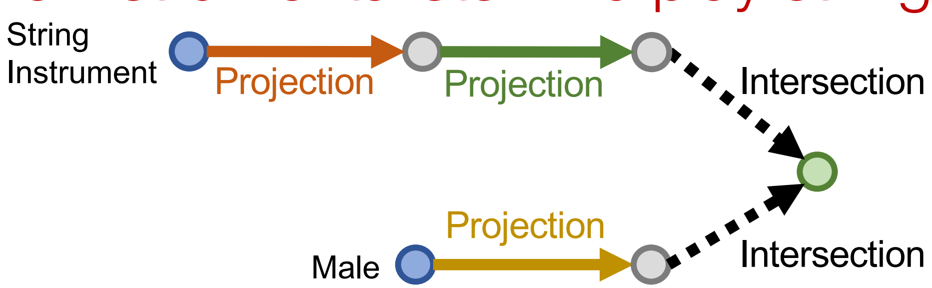
Observations:

- "Training" queries (1p-3p): +20%
- New conjunctive queries (2i,3i): +15%
- Disjunctive queries: +36%

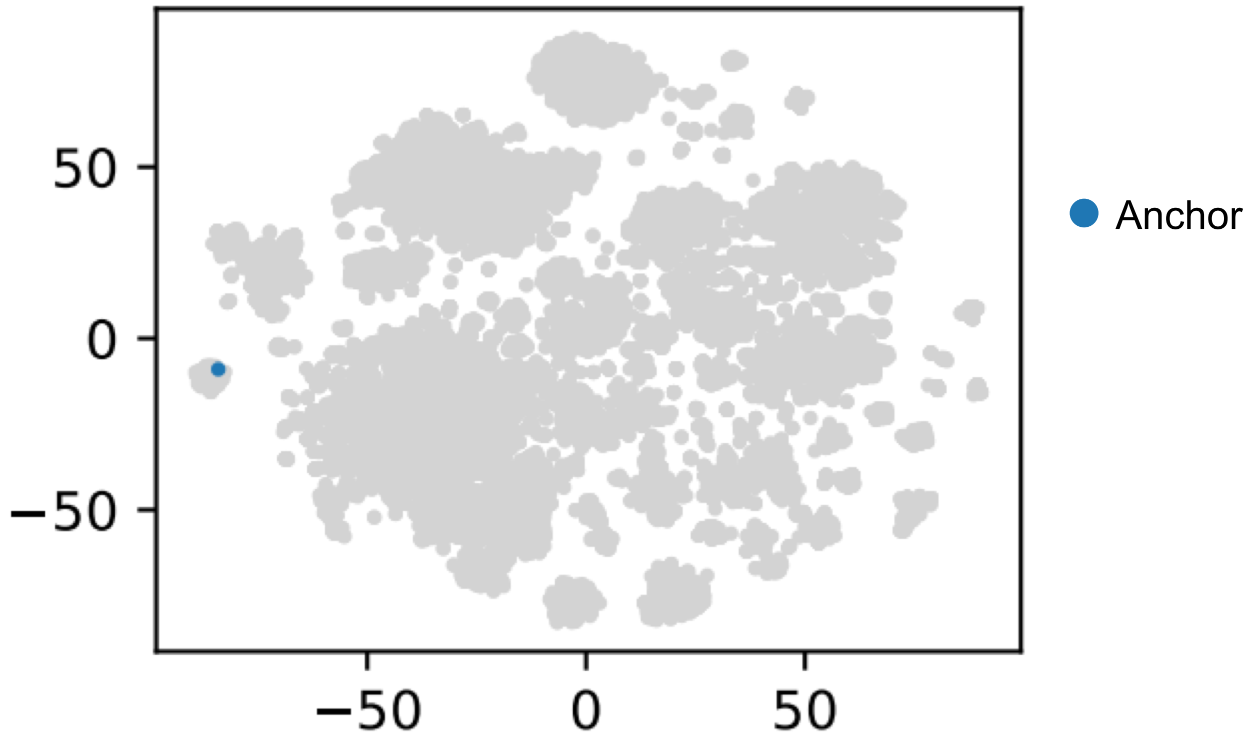
FB15k: Embedding Space



“List male instrumentalists who play string instruments”

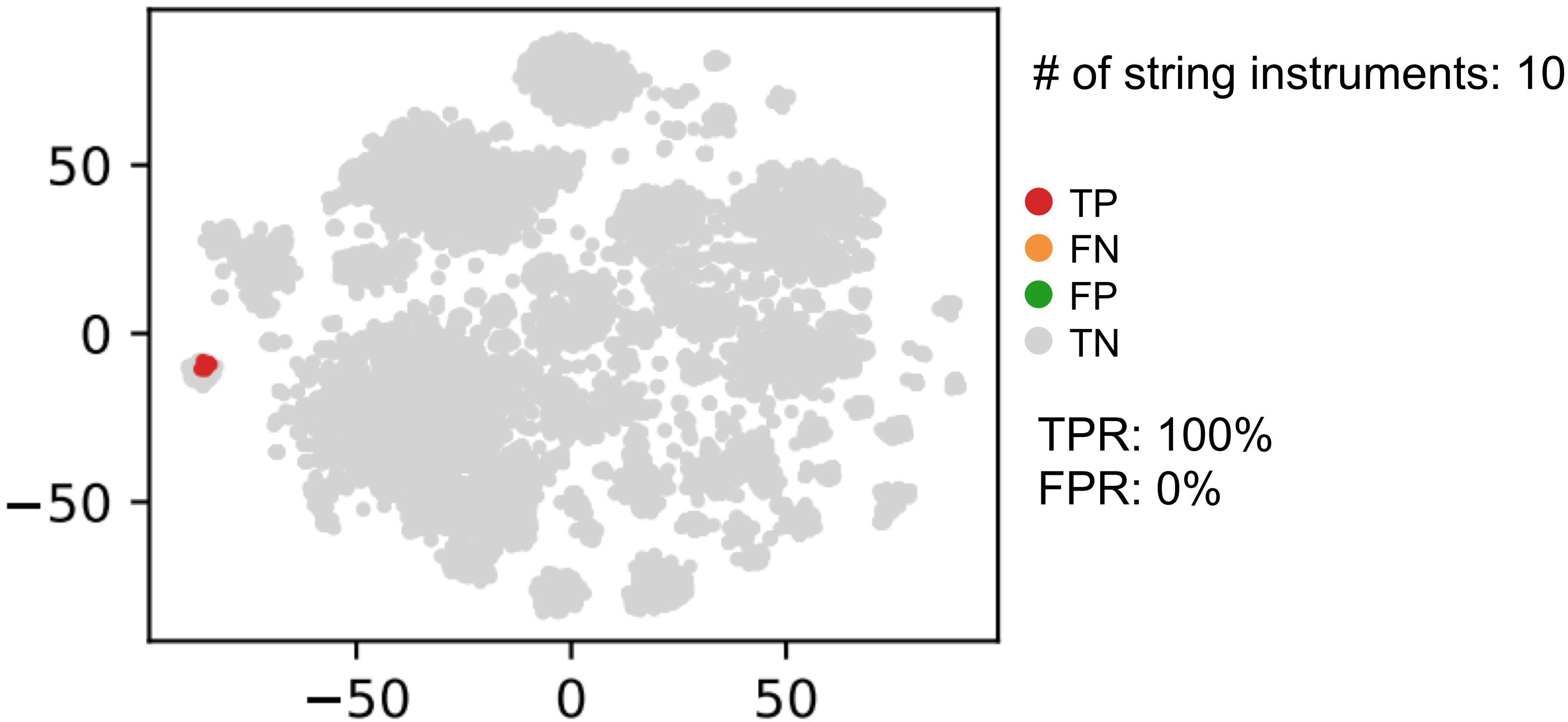


FB15k: Embedding Space



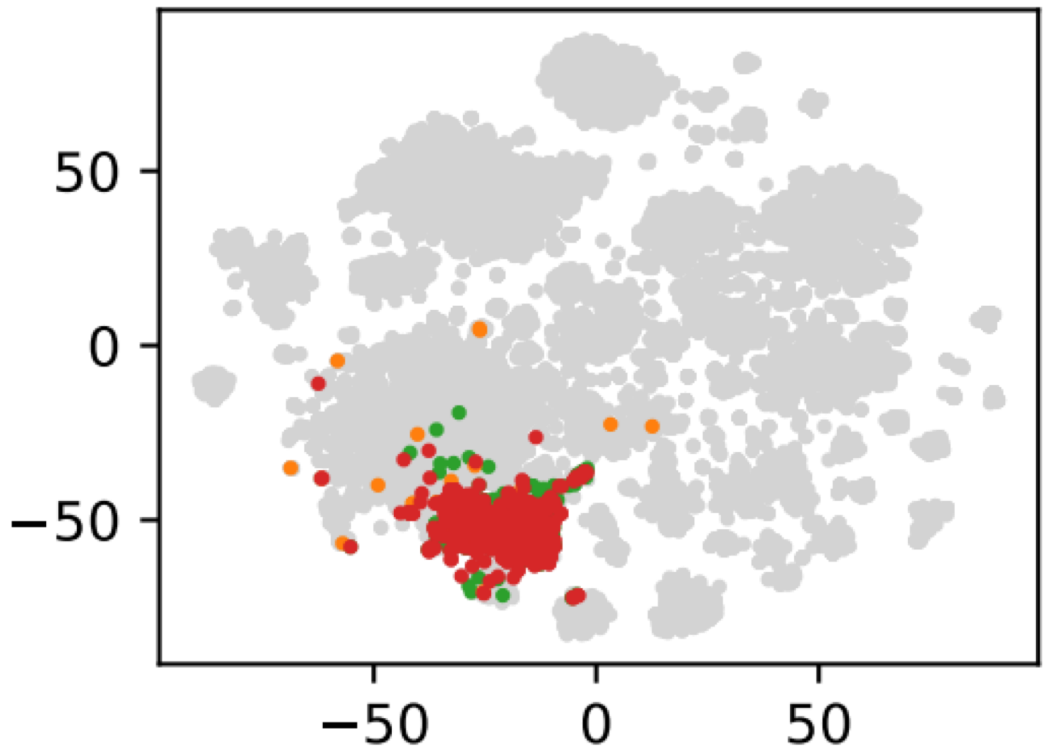
“List male instrumentalists who play string instruments”

FB15k: Embedding Space



"List male instrumentalists who play string instruments"

FB15k: Embedding Space



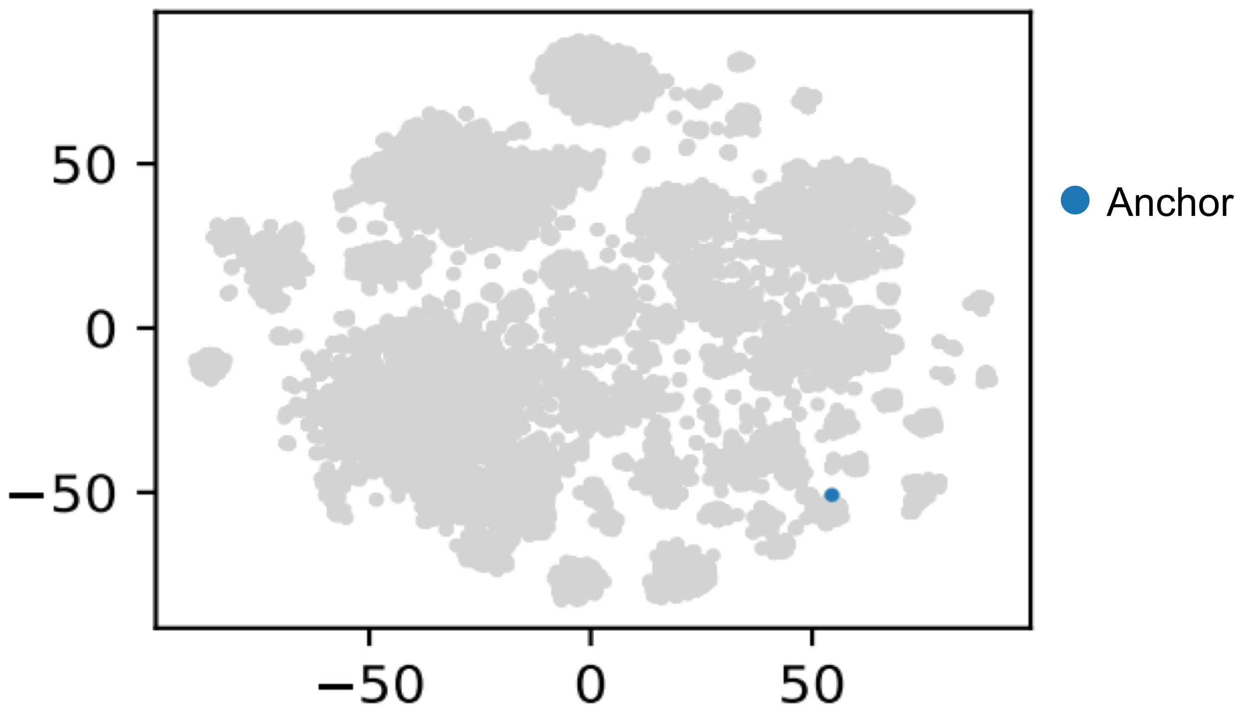
# of instrumentalists: 472

- TP
- FN
- FP
- TN

TPR: 98.4%
FPR: 0.01%

“List male instrumentalists who play string instruments”

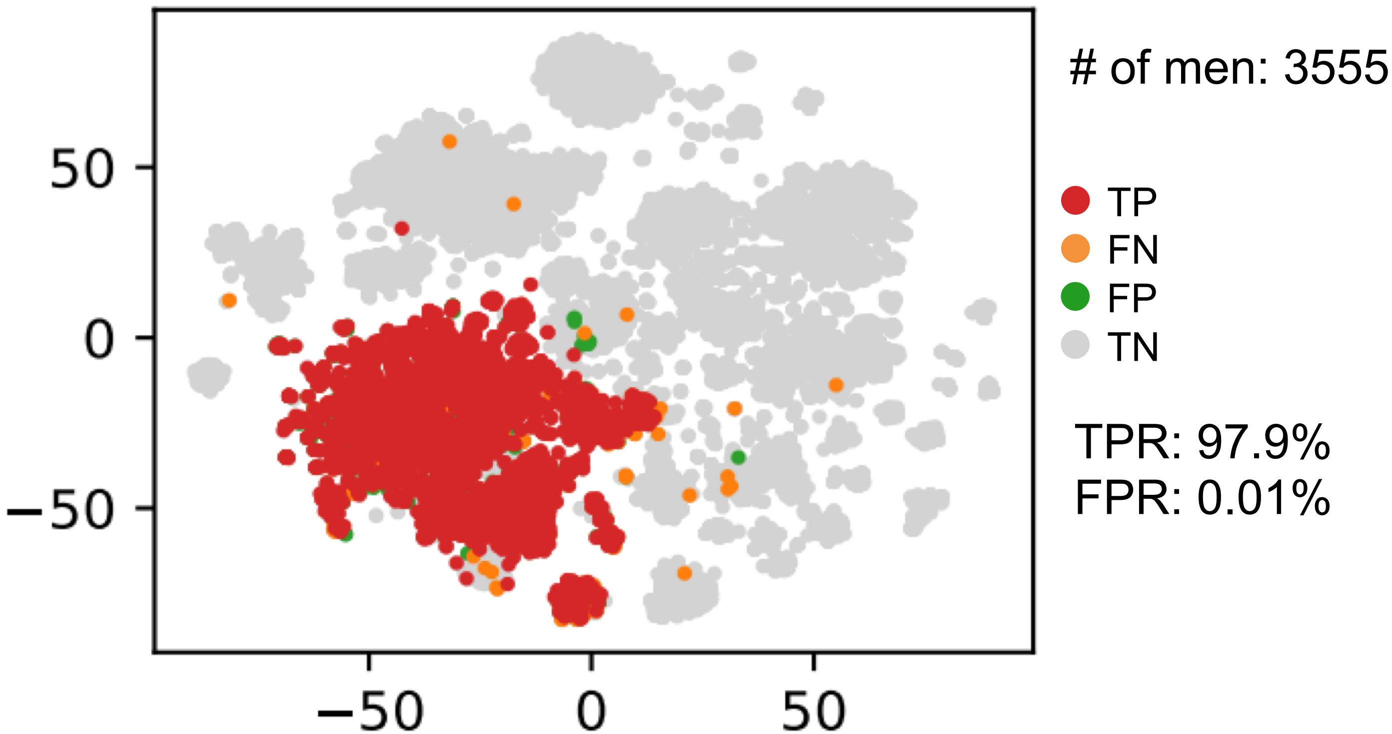
FB15k: Embedding Space



"List male instrumentalists who play string instruments"

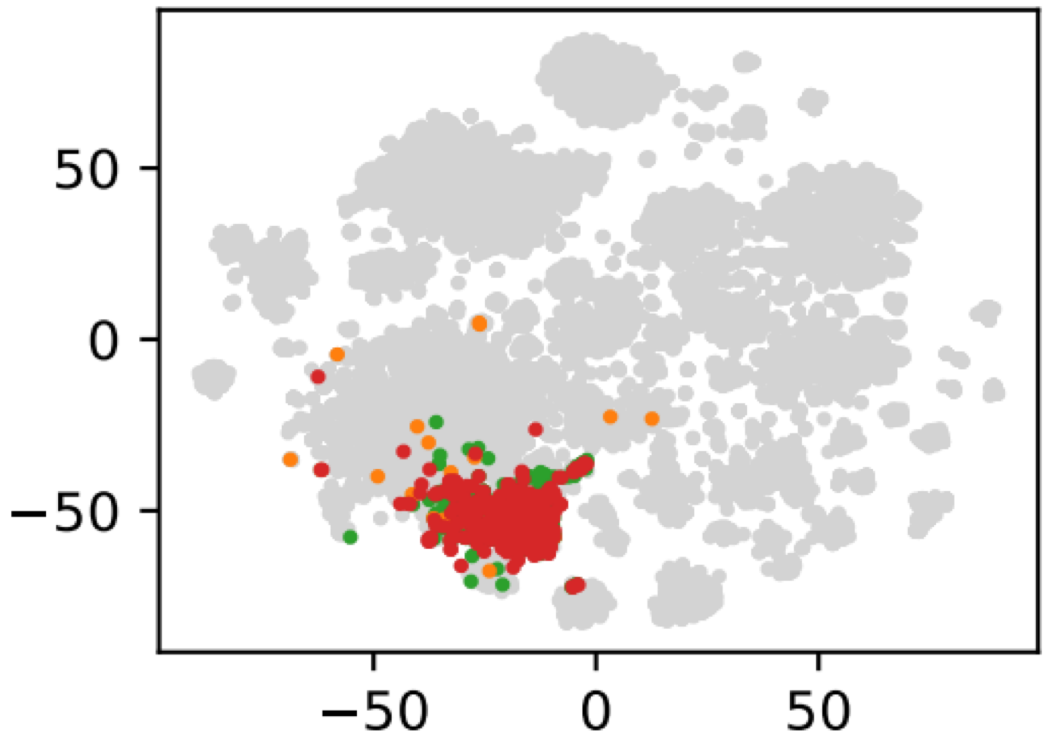
Male

FB15k: Embedding Space



“List male instrumentalists who play string instruments”

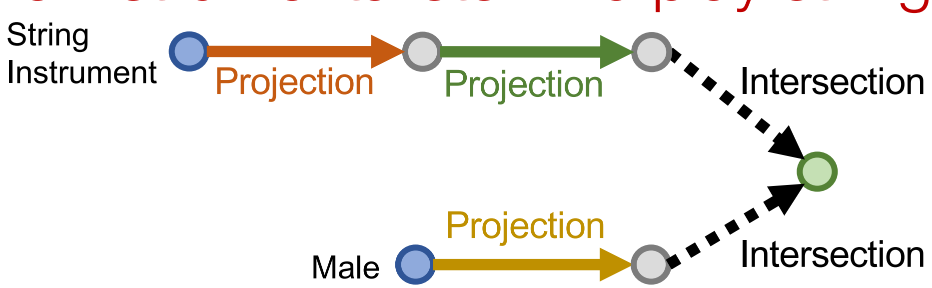
FB15k: Embedding Space



# of answers: 396

TPR: 99.4%
FPR: 0.01%

"List male instrumentalists who play string instruments"



Query2Box: Summary

- **Query2Box:**
  - Embed the query as a box
  - Logical operations become spatial operations
- Composability of queries:
  - Generalize well to unseen, extrapolated queries
  - Explicitly training for composability is important
- Instance vs. multi-hop generalization

Conclusion

- Box embeddings for answering logical queries on Knowledge graphs
- Handle union and intersection
- Generalize well to unseen, extrapolated queries
- Future work: Handle negation, other geometric model

References

- Embedding Logical Queries on Knowledge Graphs. W. Hamilton, P. Bajaj, M. Zitnik, D. Jurafsky, J. Leskovec. *NIPS*, 2018.

- Query2box: Reasoning over Knowledge Graphs in Vector Space Using Box Embeddings. H. Ren, W. Hu, J Leskovec, ICLR 2020.