Advancements in Knowledge Graph Reasoning

Innovative Approaches to Complex Logical Query Answering
and Logical Hypothesis Generation

Jiaxin Bai

KnowComp, Department of CSE
The Hong Kong University of Science and Technology

Roadmap

- Background
- Complex Query Answering
- Complex Hypothesis Generation
- Complex Session Intension Understanding
- Future Work

Limitations of Current AI Systems

- LLMs struggle with hallucinations in complex reasoning

What are the cities that have higher latitude than Beijing but lower than New York

GPT-3.5-Turbo Poe

To find cities that have a higher latitude than Beijing but lower than New York, we can look for cities that lie between the latitudes of these two cities.

Beijing's latitude is approximately 39.9042° N, and New York City's latitude is approximately 40.7128° N.

Cities that have a higher latitude than Beijing (39.9042° N) but lower than New York City (40.7128° N) could include cities like:

1. Seoul, South Korea (latitude: 37.5665° N)
2. Tokyo, Japan (latitude: 35.6895° N)
3. Madrid, Spain (latitude: 40.4168° N)

These cities lie in the latitude range you specified.

Who are graduated from HKUST and finded an academic position in US in Computer Science area

GPT-4o Poe ···

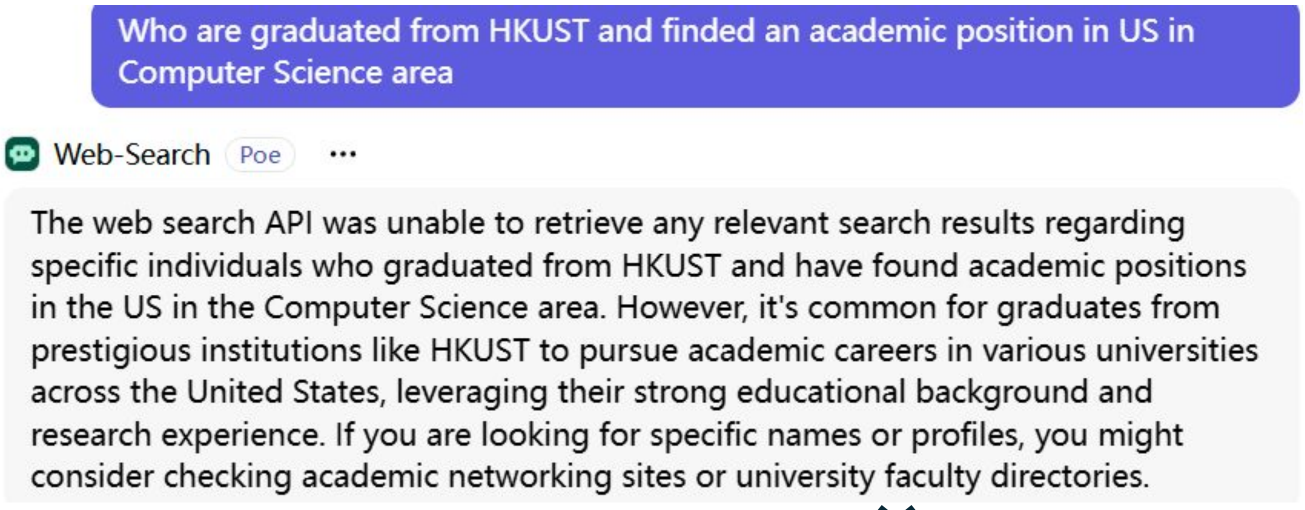
There are several HKUST graduates who have successfully secured academic positions in the U.S. in the field of Computer Science. While I cannot provide an exhaustive or up-to-date list of individuals (as this would require access to specific alumni databases or personalized networks), here are some notable examples and overall guidance on identifying such individuals:

Notable Graduates of HKUST in U.S. Academia (Computer Science)

1. **Dr. Ke Yi**
   - **Position**: Professor at Duke University (USA).
   - **Area of Expertise**: Algorithms, databases, and computational geometry.
   - **HKUST Connection**: Formerly a faculty member at HKUST before transitioning to academia in the U.S.
2. **Dr. Wei Wang**
   - **Position**: Professor at UCLA (University of California, Los Angeles).
   - **Area of Expertise**: Data mining, bioinformatics, and machine learning.
   - **HKUST Connection**: Graduated from HKUST and built a successful career in academia in the U.S.
3. **Dr. Fei Sha**
   - **Position**: Previously a Professor at USC (University of Southern California), later joined Google.
   - **Area of Expertise**: Machine learning, artificial intelligence, and computer vision.
   - **HKUST Connection**: Received his Ph.D. from HKUST.
4. **Dr. Dong Wang**
   - **Position**: Assistant Professor at UC Merced (University of California, Merced).
   - **Area of Expertise**: Distributed systems, energy-efficient computing, and wireless networks.
   - **HKUST Connection**: Graduated with a Ph.D. in Computer Science from

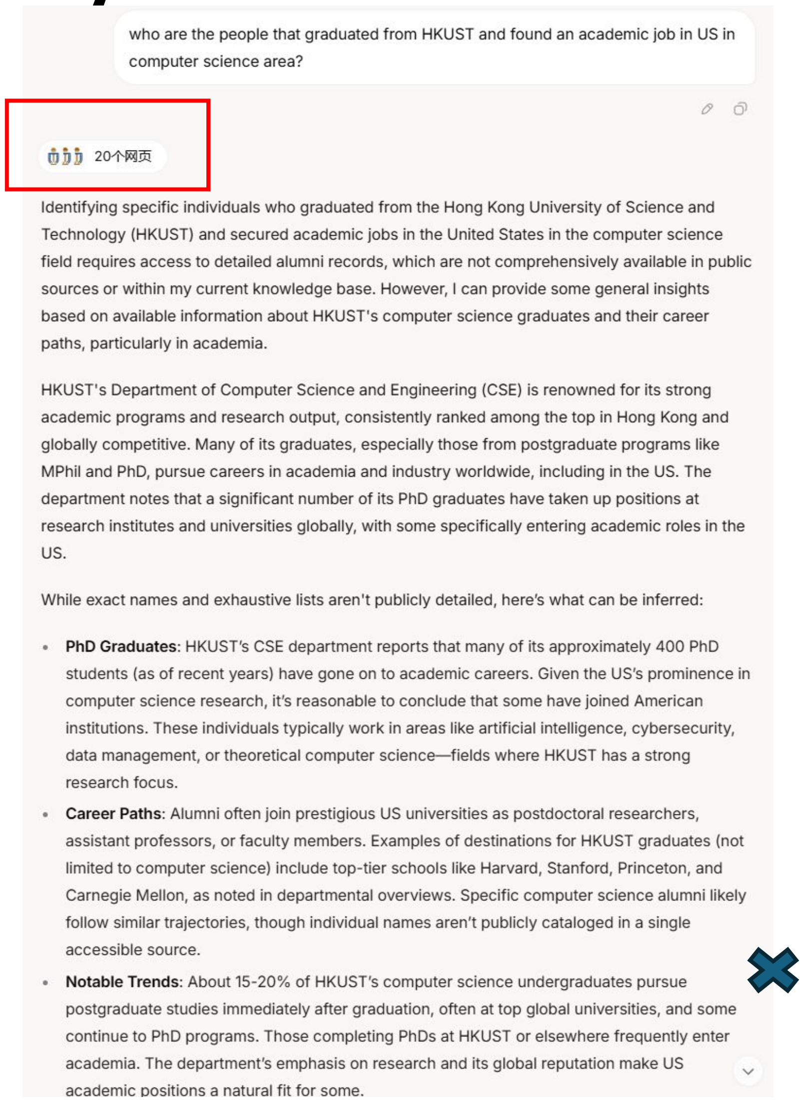
Limitations of Current AI Systems

- Vector DBs focus on **document similarity**, not logical queries
- Need for **deeper inference** and **structured knowledge**


Who are graduated from HKUST and finded an academic position in US in Computer Science area

Web-Search Poe ···

The web search API was unable to retrieve any relevant search results regarding specific individuals who graduated from HKUST and have found academic positions in the US in the Computer Science area. However, it's common for graduates from prestigious institutions like HKUST to pursue academic careers in various universities across the United States, leveraging their strong educational background and research experience. If you are looking for specific names or profiles, you might consider checking academic networking sites or university faculty directories.


who are the people that graduated from HKUST and found an academic job in US in computer science area?

20个网页

Identifying specific individuals who graduated from the Hong Kong University of Science and Technology (HKUST) and secured academic jobs in the United States in the computer science field requires access to detailed alumni records, which are not comprehensively available in public sources or within my current knowledge base. However, I can provide some general insights based on available information about HKUST's computer science graduates and their career paths, particularly in academia.

HKUST's Department of Computer Science and Engineering (CSE) is renowned for its strong academic programs and research output, consistently ranked among the top in Hong Kong and globally competitive. Many of its graduates, especially those from postgraduate programs like MPhil and PhD, pursue careers in academia and industry worldwide, including in the US. The department notes that a significant number of its PhD graduates have taken up positions at research institutes and universities globally, with some specifically entering academic roles in the US.

While exact names and exhaustive lists aren't publicly detailed, here's what can be inferred:

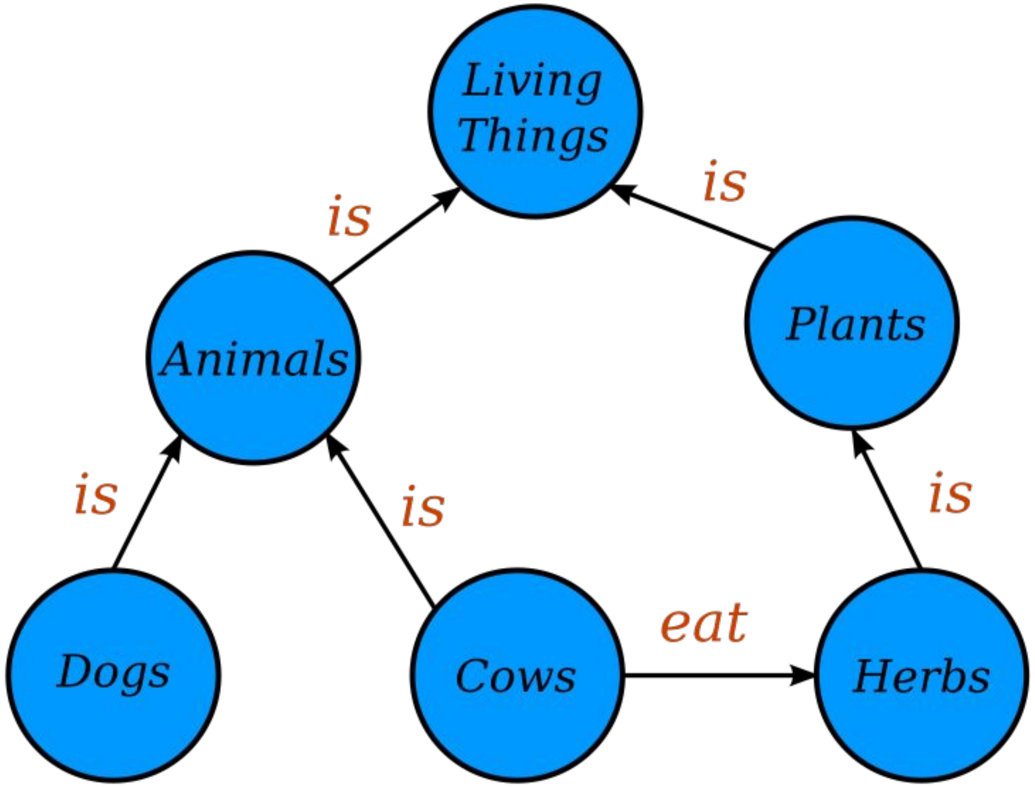
- **PhD Graduates**: HKUST's CSE department reports that many of its approximately 400 PhD students (as of recent years) have gone on to academic careers. Given the US's prominence in computer science research, it's reasonable to conclude that some have joined American institutions. These individuals typically work in areas like artificial intelligence, cybersecurity, data management, or theoretical computer science—fields where HKUST has a strong research focus.
- **Career Paths**: Alumni often join prestigious US universities as postdoctoral researchers, assistant professors, or faculty members. Examples of destinations for HKUST graduates (not limited to computer science) include top-tier schools like Harvard, Stanford, Princeton, and Carnegie Mellon, as noted in departmental overviews. Specific computer science alumni likely follow similar trajectories, though individual names aren't publicly cataloged in a single accessible source.
- **Notable Trends**: About 15-20% of HKUST's computer science undergraduates pursue postgraduate studies immediately after graduation, often at top global universities, and some continue to PhD programs. Those completing PhDs at HKUST or elsewhere frequently enter academia. The department's emphasis on research and its global reputation make US academic positions a natural fit for some.

Need Grounded Reasoning on Structured Data!
For example a knowledge graph

Structured Knowledge: Knowledge Graphs

G consisting of:

- Entities (nodes) E: objects, concepts, or ideas
- Relation types (edge labels) R: associations between entities
- Triples $(e_h, r, e_t) \in G$: head and tail entities connected by a relation



How to use KG for Reasoning?

- How to answer knowledge graph queries on different types of knowledge graphs given the graphs are always incomplete?

  Complex Query Answering

- How to obtain logical hypothesis from observations by using structured knowledge?

  Logical Hypothesis Generation

- How to apply the KG reasoning to real e-commerce application like search and recommendation?

  Complex Session Intension Understanding

Roadmap

- Background
- Complex Query Answering
- Complex Hypothesis Generation
- Complex Session Intension Understanding
- Future Work

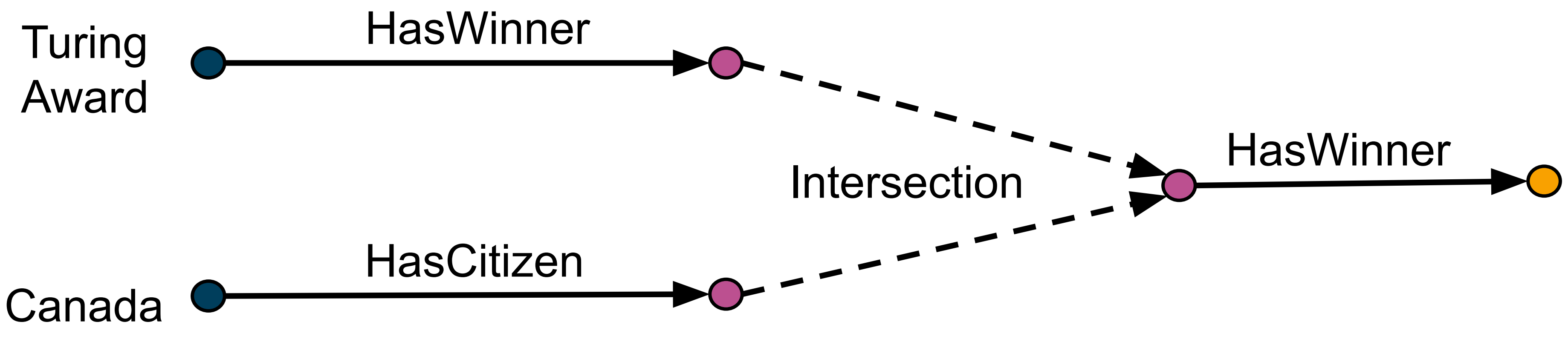
Complex Query Answering

Complex Queries	Interpretations
	Find where the Canadian Turing award laureates graduated from.
	Find the substances that interact with the proteins associated with diseases T1, T2, or T3.
	Find entities, who are Germans, were the Nobel Prize winners and eventually moved to the United States.

How do we deal with the incompleteness of KG?

How do we scale up to large knowledge graphs and long queries?

Query Encoding



How do we deal with the incompleteness of KG?

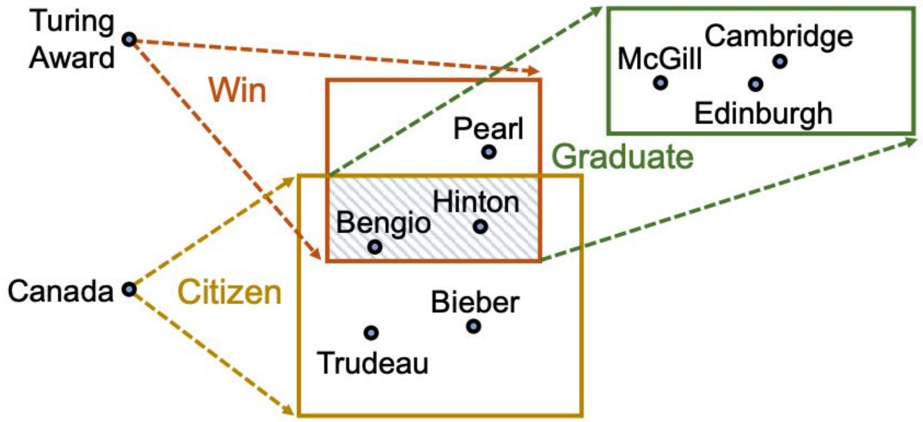
- Use embeddings to represent queries and subqueries

How do we scale up to large knowledge graphs and long queries?

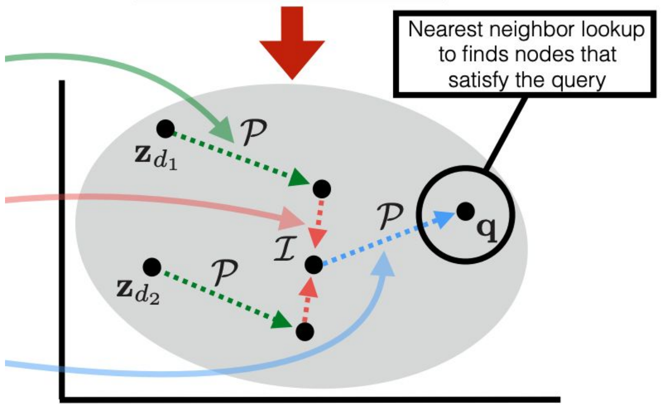
- Using computation to replace graph search / subgraph matching
- Only a single approximate nearest neighbor search in inference

Neural Query Encoders

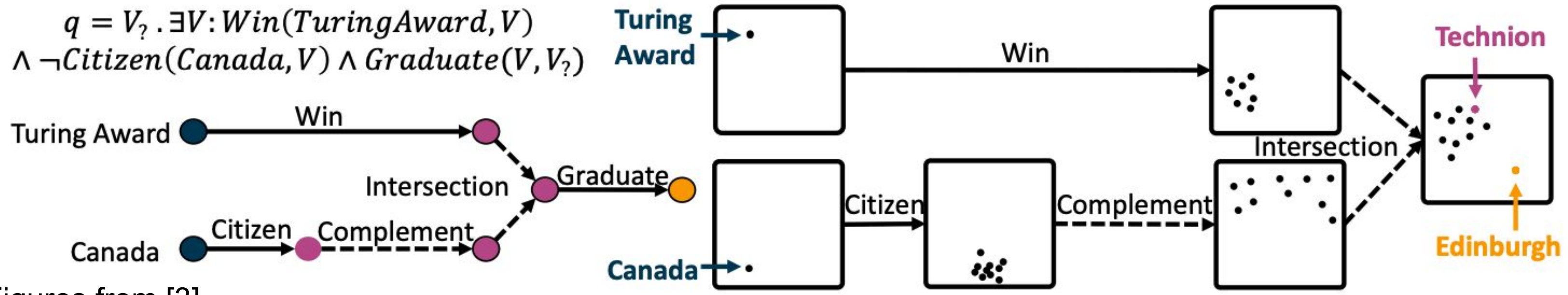
Models	Encoding Structures
GQE [1]	Vector Embedding
Query2Box [2]	Box Embedding
Query2Particles [3]	Multiple Vectors Embedding
FuzzQE [4]	Fuzzy Logic Embedding
Neural MLP [5]	Vector Embedding
NewLook [6]	Vector Embedding
…	…



Figures from [2]



Figures from [1]



Figures from [3]

https://github.com/neuralgraphdatabases/awesome-logical-query

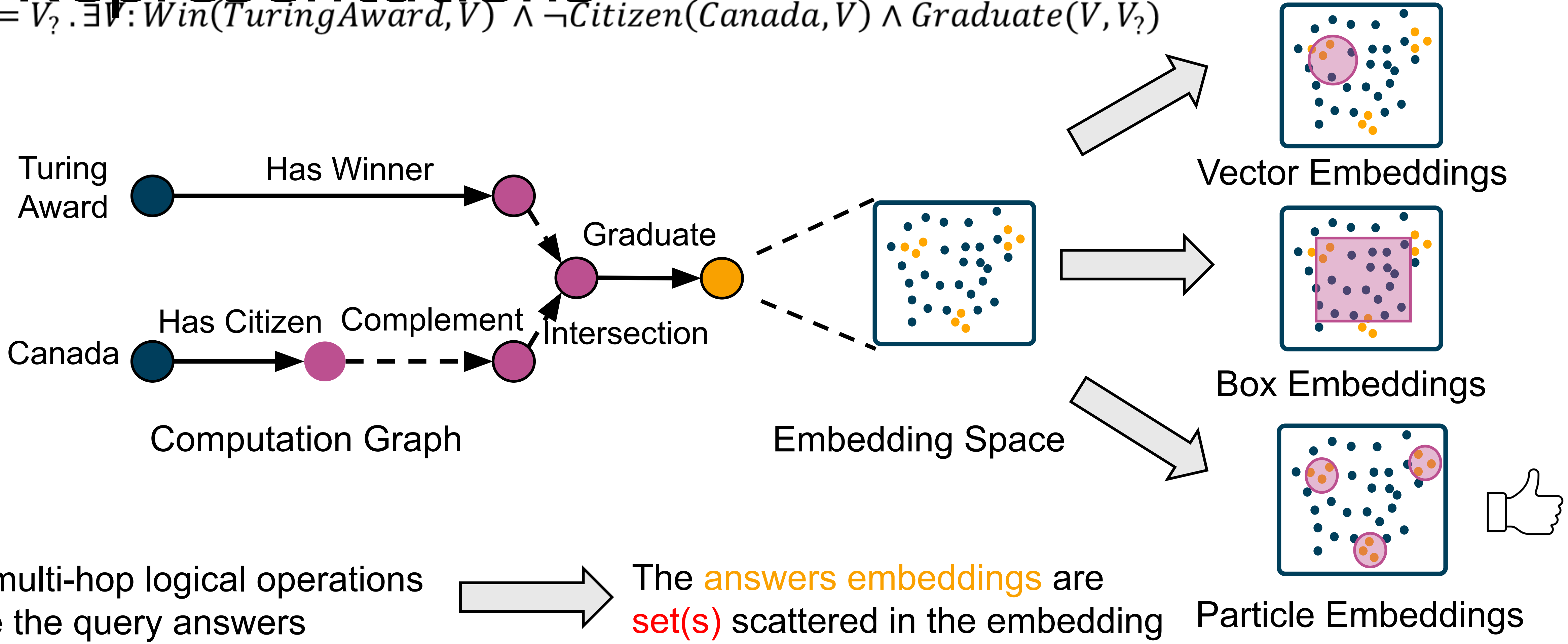
Query2Particles: Knowledge Graph Reasoning with Particle Embeddings

Jiaxin Bai, Zihao Wang, Hongming Zhang, Yangqiu Song

NAACL-2022 (Findings)

Embedding Space and Set Representations

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



The multi-hop logical operations make the query answers diversified

The answers embeddings are set(s) scattered in the embedding space

Example from: Jiaxin Bai, Zihao Wang, Hongming Zhang, Yangqiu Song: Query2Particles: Knowledge Graph Reasoning with Particle Embeddings. NAACL-HLT (Findings) 2022: 2703-2714

Relational Projection

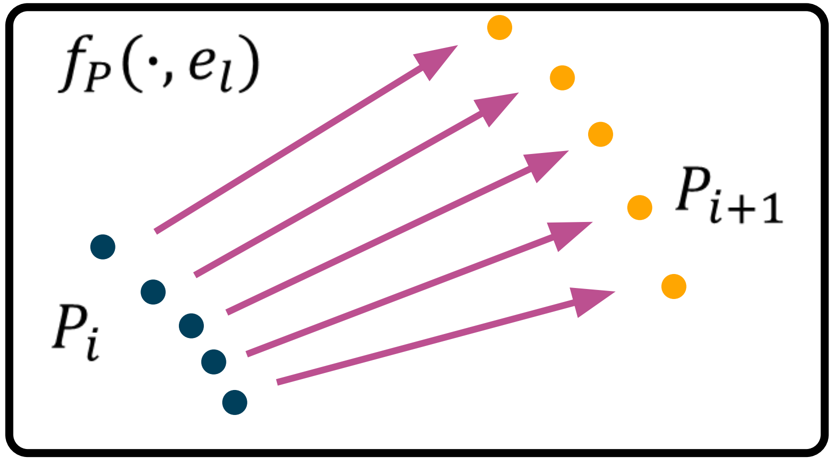
- Input: particles P_i and relation l
- Output: Particles P_{i+1}
- Gated Transition for customizing the directions of transitions for each vector in particles:

$$A_i = (1 - Z) \odot P_i + Z \odot T$$

Here Z is the update gate, and T is transition for each particles. They are computed from P_i and the relation embedding e_l for relation l

- Self-Attention for information exchange in the particles:

$$P_{i+1} = \texttt{self-attn}(A_i)$$

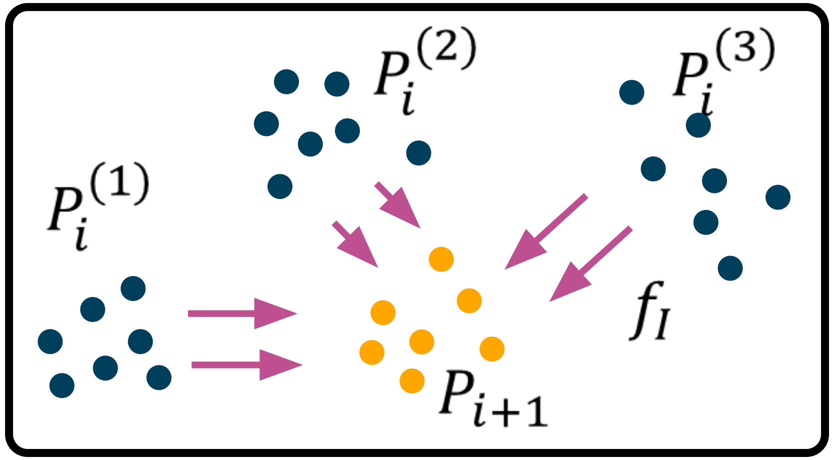


Relational Projection

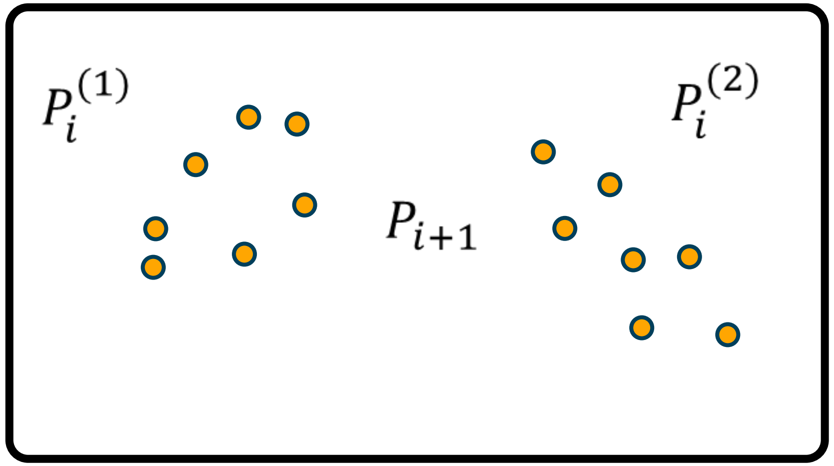
Intersection, Union, and Negation

- Input: particles P_i
- Output: Particles P_{i+1}
- Self-Attention + MLP layer for information exchange between the input particles:

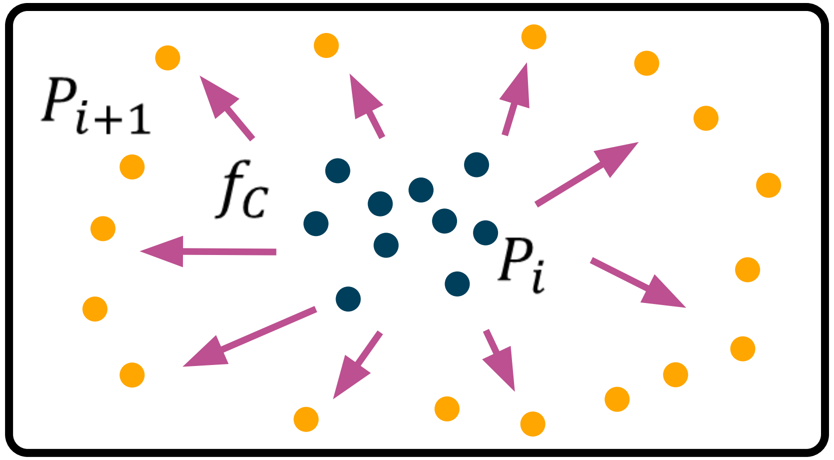
$$A_i = \texttt{self-attn}(P_i)$$
$$P_{i+1} = \texttt{MLP}(A_i)$$



Intersection



Union



Complement

Training Query2Particles

- Use positive pairs with cross-entropy loss for training:

- The scoring function is the maximal inner product from particles to an entity v:

$$\varphi(P_q,\ v) = \max_{k=1,2,3,\ldots,K} < p_T^{(k)}, e_v >$$

- The normalized probability of the entity v being an answer of q:

$$P(q,\ v) = \frac{e^{\varphi(P_q, v)}}{\sum_{v'} e^{\varphi(P_q, v')}}$$

- The cross-entropy loss maximizes the log probability of all query-and-answer pairs:

$$L = -\frac{1}{N} \sum_i \log P(q^{(i)}, v^{(i)})$$

Dataset

DATASET	RELATIONS	ENTITIES	TRAINING	VALIDATION	TESTING	ALL EDGES
FB15K	1,345	14,951	483,142	50,000	59,071	592,213
FB15K-237	237	14,505	272,115	17,526	20,438	310,079
NELL995	200	63,361	114,213	14,324	14,267	142,804

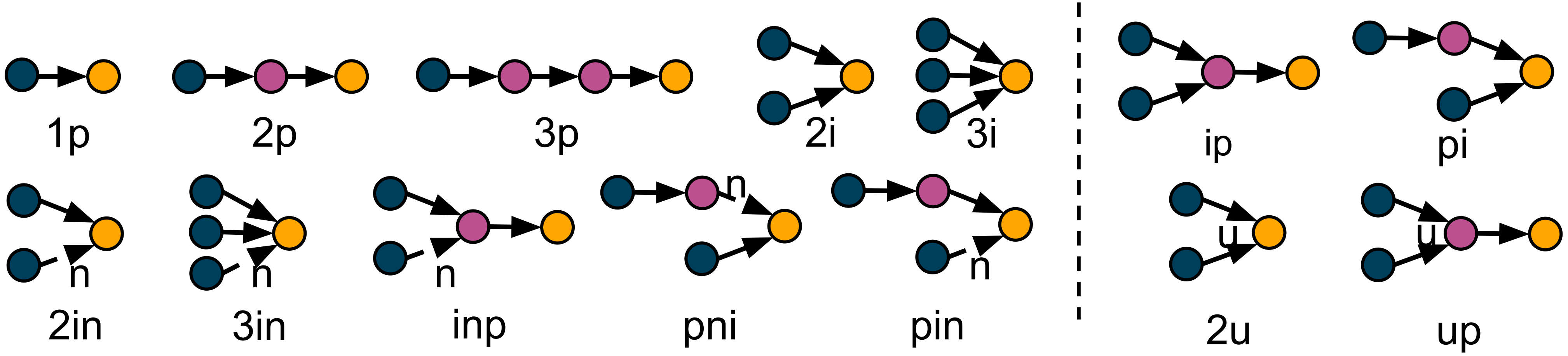
The basic information about the three knowledge graphs used for the experiments.

(REN ET AL., 2020)	TRAINING		VALIDATION		TEST	
DATASET	1P	OTHERS	1P	OTHERS	1P	OTHERS
FB15K	273,710	273,710	59,097	8,000	67,016	8,000
FB15K-237	149,689	149,689	20,101	5,000	22,812	5,000
NELL995	107,982	107,982	16,927	4,000	17,034	4,000
(REN AND LESKOVEC, 2020)	TRAINING		VALIDATION		TEST	
DATASET	1P/2P/3P/2I/3I	2IN/3IN/INP/PIN/PNI	1P	OTHERS	1P	OTHERS
FB15K	273,710	273,71	59,097	8,000	67,016	8,000
FB15K-237	149,689	149,68	20,101	5,000	22,812	5,000
NELL995	107,982	107,98	16,927	4,000	17,034	4,000

The detailed information for the queries used for training, validating, and testing all query embedding methods.

Training Query2Particles

In-distribution types Out-of-distribution types



p: projection
i: intersection
n: negation
u: union

In-distribution: used for training and evaluation
Out-of-distribution: no training, evaluation only

Comparison with baselines

	FB15K										FB15K-237	NELL
MODEL	1P	2P	3P	2I	3I	IP	PI	2U	UP	AVG	AVG	AVG
EMQL	42.4	50.2	45.9	63.7	70.0	60.7	61.4	9.0	42.6	49.5	35.8	46.8
– SKETCH	50.6	46.7	41.6	61.8	67.3	54.2	53.5	21.6	40.0	48.6	35.5	46.8
CQD-BEAM	**91.8**	**77.9**	57.7	79.6	83.7	37.5	65.8	**83.9**	34.5	68.0	29.0	37.6
CQD-CO	**91.8**	45.4	19.1	79.6	83.7	33.6	51.3	81.6	31.9	57.6	27.2	36.8
Q2P (OURS)	90.2	74.6	**73.4**	**86.0**	**89.6**	**63.7**	**77.6**	83.4	**52.7**	**76.8**	**43.0**	**52.2**

Table 3: The Hit@3 results for existential positive first-order queries originally used by Ren, Hu, and Leskovec (2020) and the comparisons are made against the state-of-the-art baselines including EmQL and CQD methods. The best results are marked in **bold** and the second-best ones are marked with underlines. The full results are in the supplementary materials.

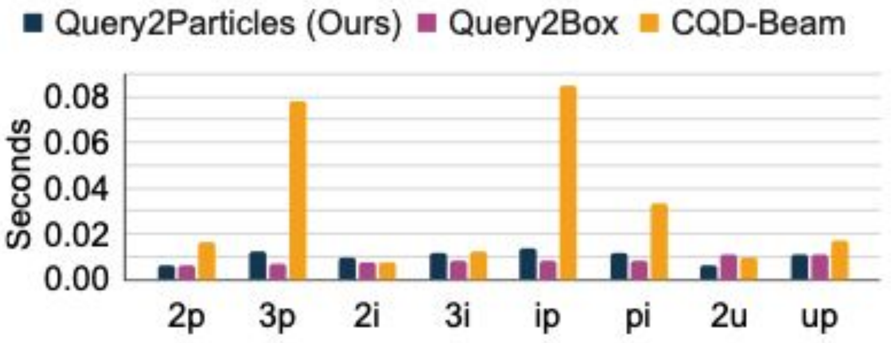


Figure 6: The inference time of different types of queries on FB15k, which has the largest number of edges among three graphs.

- There is no need to pre-process the queries to their disjunction-normal forms (DNF), which takes exponential complexity
- Fast inference speed compared to the query decomposition method

Queries with Diverse Answers

Models	1P	2I	2U	2IN	Average
Q2P-1P	44.8	28.8	11.3	15.0	25.0
Q2P-2P	49.4	35.5	13.3	20.7	29.7
Q2P-3P	**53.0**	**37.7**	**18.6**	**21.6**	**32.8**

Significantly better on the queries with diverse answers

MRR on the top ten percent diversified queries
Diversity is measured by the number of answers.

Sequential Query Encoding For Complex Query Answering on Knowledge Graphs

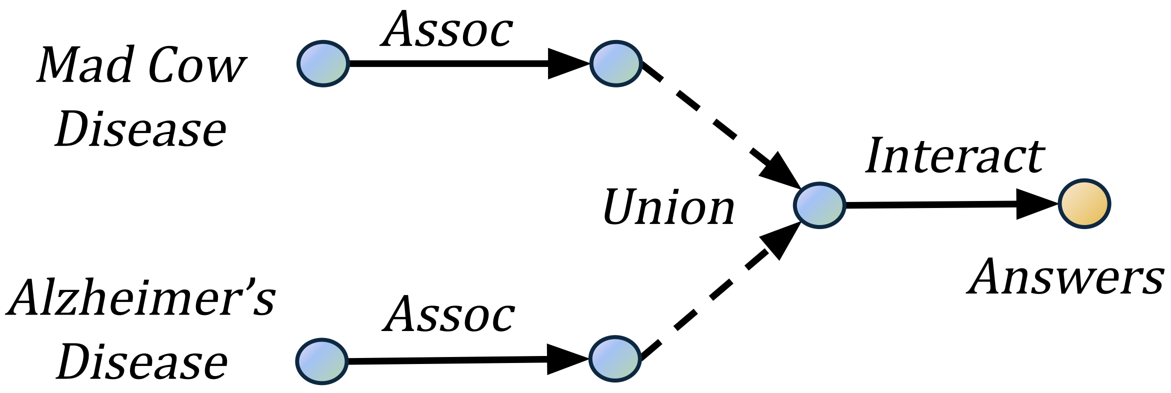
Jiaxin Bai*, Tianshi Zheng*, Yangqiu Song

Transactions of Machine Learning Research

Neural Networks as Operators

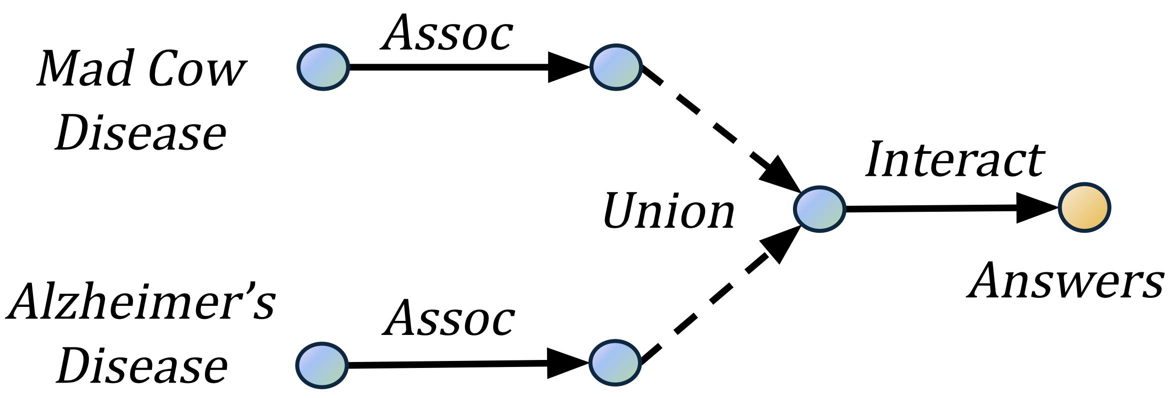
$$q_1 = V_? \,.\, \exists V : Interact(V_?, V) \wedge (Assoc(V, Alzheimer) \vee Assoc(V, MadCow))$$

From a query to a computation graph



Do we have to parameterize and then execute such computational graph?

From Computing to Encoding

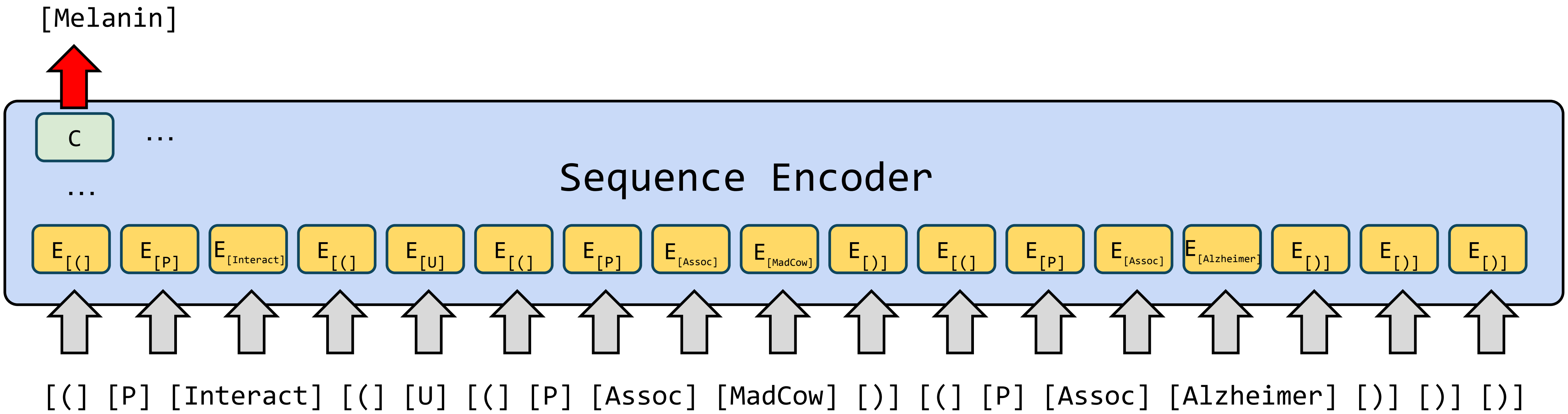


Tokenization

```
[(] [P] [Interact]
    [(] [U]
        [(] [P] [Assoc] [MadCow] [)]
        [(] [P] [Assoc] [Alzheimer ] [)]
    [)]
[)]
```

- Previous: neural networks as operators, then execute the graph
- SQE: tokenize & encode the whole query graph by a single NN

Sequential Query Encoding



Sequence Encoder: Transformers, LSTMs, Temporal CNN…

Experiments - Benchmarks

Dataset	In-distribution Types	Out-of-distribution Types	Total
Query2Box	5	4	9
BetaE	10	4	14
SMORE	10	4	14
This Paper (SQE)	**29**	**29**	**58**

We construct a larger benchmark with diverse query types.

Results

Sequential query encoding is good for the queries when its query types are seen

The operator-level parametrization is good for compositional generalization for unseen query types

Datasets	Models	In-distribution Queries		Out-of-distribution Queries	
		Entailment	Inference	Entailment	Inference
FB15K-237	ConE	36.69	9.75	28.13	**8.82**
	BetaE	32.48	8.30	22.96	7.29
	Q2P	52.33	10.17	32.70	8.62
	Neural MLP	51.09	10.03	**36.85**	<u>8.75</u>
	+ MLP Mixer	45.19	10.07	33.03	8.66
	SQE + CNN	52.09	10.14	28.21	7.65
	SQE + GRU	55.46	10.59	32.25	8.34
	SQE + LSTM	<u>56.02</u>	<u>10.62</u>	<u>33.41</u>	8.62
	SQE + Transformer	**59.15**	**11.30**	15.06	4.98

Knowledge Graph Reasoning over Entities and Numerical Values

Jiaxin Bai, Chen Luo, Zheng Li, Qingyu Yin, Bing Yin, Yangqiu Song

KDD-2023

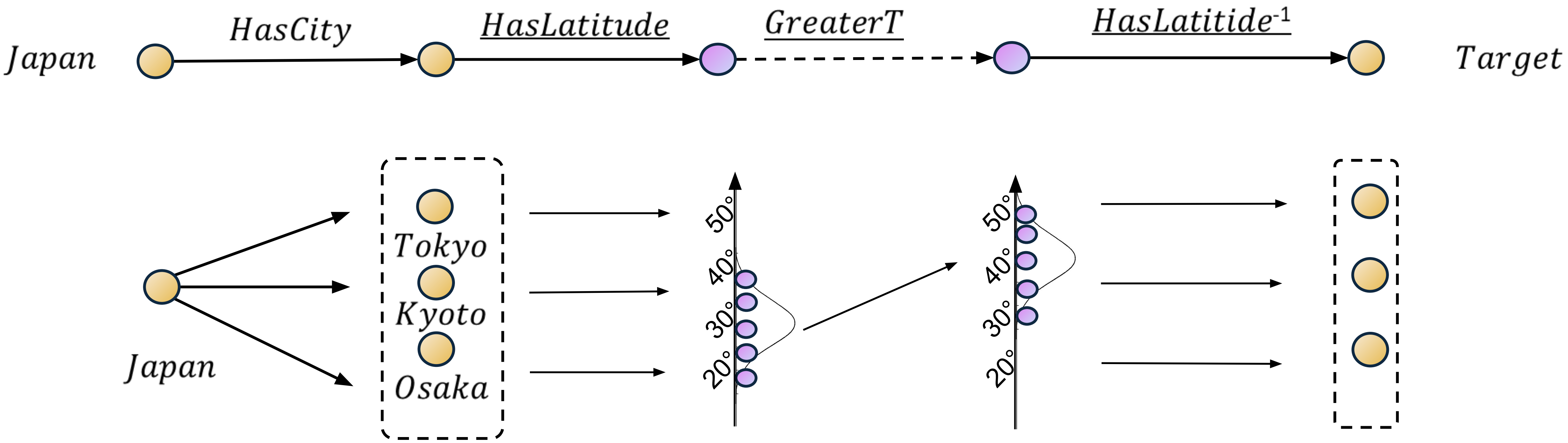
Numerical Complex Query Answering

Category	Complex Queries	Interpretations
Numerical CQA	$q_2 = V_? \,.\, \exists X_1, X_2: Win(V_?, TuringAward) \wedge \underline{GreaterThan}(1927, X_2) \wedge \underline{BornIn}\,(V_?, X_2)$	Find the Turing award winners that is born before the year of 1927.
Numerical CQA	$q_3 = V_? \,.\, \exists X_1, X_2: LocatedIn(V_?, UnitedStates) \wedge \underline{HasLatitude}(V_?, X_1) \wedge \underline{GreaterThan}(X_1, X_2) \wedge \underline{HasLatitude}(Beijing, X_2)$	Find the states in US that have a higher latitudes than Beijing.
Numerical CQA	$q_4 = V_? \,.\, \exists X_1, X_2, X_3: LocatedIn(V_?, UnitedStates) \wedge \underline{HasPopulation}(V_?, X_1) \wedge \underline{SmallerThan}(X_1, X_2) \wedge \underline{TimesByTwo}(X_2, X_3) \wedge \underline{HasPopulation}(California, X_3)$	Find the states in US that have a twice smaller population than California?

Number Reasoning Network

Find the cities that have a <u>higher latitudes than</u> Japanese cities.

$$q = V_? \,.\, \exists V_1, X_1, X_2 : \underline{HasLatitude}(V_?, X_2) \wedge \underline{GreaterThan}(X_2, X_1) \wedge \underline{HasLatitude}(V_1, X_1) \wedge LocatedIn(V_1, Japan)$$



(1) Relational Projection

(2) Attribute Projection

(3) Numerical Projection

(4) Reverse Attribute Projection

Jiaxin Bai, Chen Luo, Zheng Li, Qingyu Yin, Bing Yin, Yangqiu Song: Knowledge Graph Reasoning over Entities and Numerical Values. KDD 2023: 57-68

Main Results on Three KGs

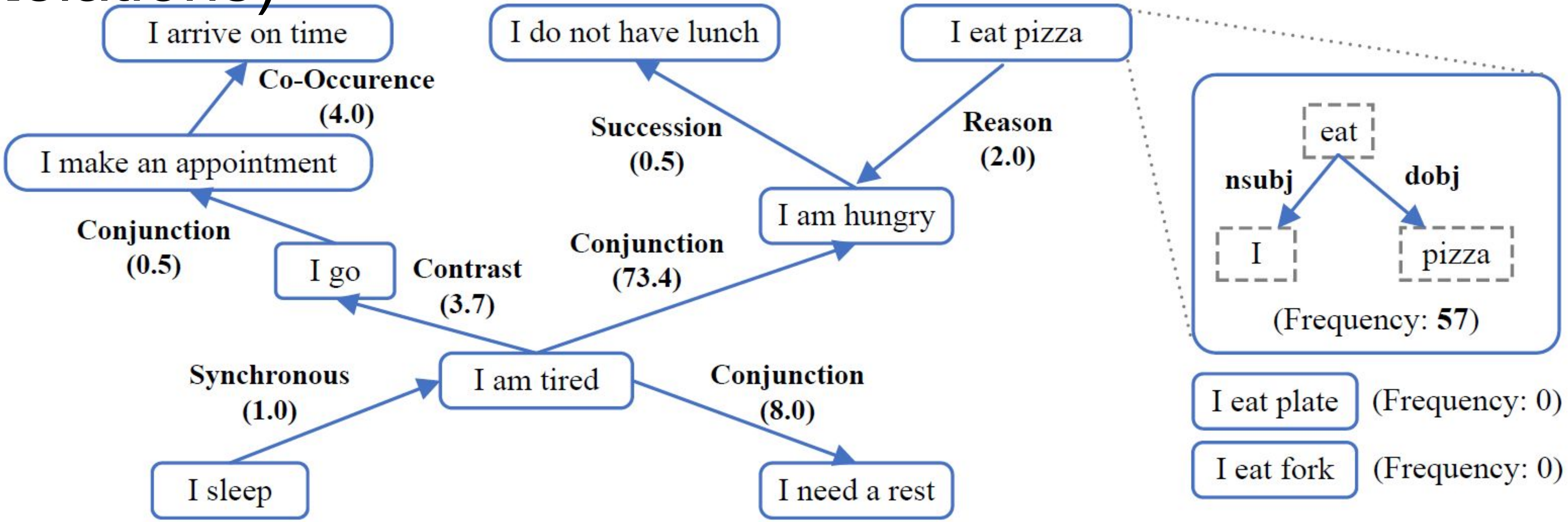
Query Encoding	Attribute	Hit@1	Hit@3	Hit@10	MRR
GQE	Baseline	10.33	18.19	27.91	16.29
	NRN + DICE	11.03	19.18	29.01	17.15
	NRN + Sinusoidal	**11.14**	**19.39**	**29.23**	**17.31**
Q2P	Baseline	10.22	17.35	26.61	15.81
	NRN + DICE	11.86	19.70	29.46	17.84
	NRN + Sinusoidal	**12.25**	**20.16**	**29.96**	**18.28**
Q2B	Baseline	11.81	20.93	31.19	18.41
	NRN + DICE	12.52	22.09	32.34	19.34
	NRN + Sinusoidal	**12.75**	**22.22**	**32.46**	**19.51**

Complex Query Answering on Eventuality Knowledge Graph with Implicit Logical Constraints

Jiaxin Bai, Xin Liu, Weiqi Wang, Chen Luo, Yangqiu Song

NeurIPS-2023

ASER (Activities, States, Events, and their Relations)



Principle 1: Comparing semantic meanings by fixing grammar (Katz and Fodor, 1963)
Principle 2: The need of language inference based on ‘partial information’ (Wilks, 1975)

https://github.com/HKUST-KnowComp/ASER

Hongming Zhang, Xin Liu, Haojie Pan, Yangqiu Song, Cane Wing-Ki Leung: ASER: A Large-scale Eventuality Knowledge Graph. WWW 2020: 201-211
Katz, J. J., & Fodor, J. A. (1963). The structure of a semantic theory. Language, 39(2), 170–210.
Yorick Wilks. 1975. An intelligent analyzer and understander of English. Communications of the ACM, 18(5):264–274.

CQA on Eventuality Knowledge Graph

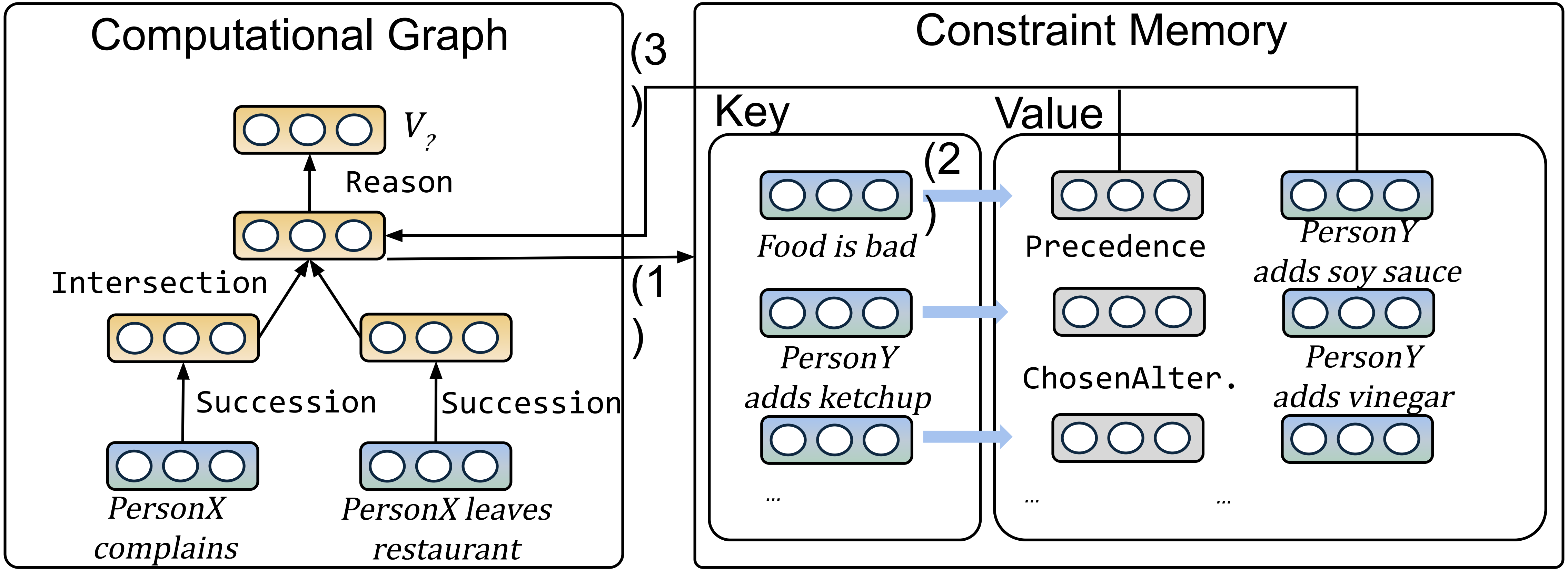
Complex query on eventuality graphs are **different** from the entity-relation graph

Whether and **when** the eventualities occur are important

Queries	Type	Interpretations
$q_1 = V_?.\exists V: \text{Interact}(V_?, V) \wedge \text{Assoc}(V, \textit{Alzheimer}) \wedge \text{Assoc}(V, \textit{MadCow})$	Entity	Find the substances that interact with the proteins associated with Alzheimer's and Mad cow disease.
$q_2 = V_?.\text{Precedence}(\textit{Food is bad}, \textit{PersonX add soy sauce}) \wedge \text{Reason}(\textit{Food is bad}, V_?)$	Eventuality	Food is bad before PersonX add soy sauce. What is the reason for food being bad?
$q_3 = V_?.\text{Precedence}(V_?, \textit{PersonX go home}) \wedge \text{ChosenAlternative}(\textit{PersonX go home}, \textit{PersonX buy an umbrella})$	Eventuality	Instead of buying an umbrella, PersonX go home. What happened before PersonX go home?

Jiaxin Bai, Xin Liu, Weiqi Wang, Chen Luo, Yangqiu Song: Complex Query Answering on Eventuality Knowledge Graph with Implicit Logical Constraints. NeurIPS, 2023

Query Encoding with Constraint Memory



(1) $s_{i,m} = < q_i,\ c_h^{(m)} >$
Computes the relevance of query embedding to the head of the memory key at position m.

(2) $v_i = \sum_{m=1}^{M} s_{i,m}\,(c_r^{(m)} + c_t^{(m)})$
Computes the aggregated memory values across M memory cells with the importance weighted by relevance scores.

(3) $q_i = q_i + MLP(v_i)$
Computes the query embedding with memory values with the help of a MLP layer.

Jiaxin Bai, Xin Liu, Weiqi Wang, Chen Luo, Yangqiu Song: Complex Query Answering on Eventuality Knowledge Graph with Implicit Logical Constraints. NeurIPS, 2023

The MEQE Combined with Various QE methods

Models	Occurrence Constraints			Temporal Constraints			Average		
	Hit@1	Hit@3	MRR	Hit@1	Hit@3	MRR	Hit@1	Hit@3	MRR
GQE	8.92	14.21	13.09	9.09	14.03	12.94	9.12	14.12	13.02
+ MEQE	**10.20**	**15.54**	**14.31**	**10.70**	**15.67**	**14.50**	**10.45**	**15.60**	**14.41**
Q2P	14.14	19.97	18.84	14.48	19.69	18.68	14.31	19.83	18.76
+ MEQE	**15.15**	**20.67**	**19.38**	**16.06**	**20.82**	**19.74**	**15.61**	**20.74**	**19.56**
Nerual MLP	13.03	19.21	17.75	13.45	19.06	17.68	13.24	19.14	17.71
+ MEQE	**15.26**	**20.69**	**19.32**	**15.91**	**20.63**	**19.47**	**15.58**	**20.66**	**19.40**
FuzzQE	11.68	18.64	17.07	11.68	17.97	16.53	11.68	18.31	16.80
+ MEQE	**14.76**	**21.12**	**19.45**	**15.31**	**21.01**	**19.49**	**15.03**	**21.06**	**19.47**

Roadmap

- Background
- Complex Query Answering
- Complex Hypothesis Generation
- Complex Session Intension Understanding
- Future Work

Advancing Abductive Reasoning in Knowledge Graphs through Complex Logical Hypothesis Generation

Jiaxin Bai*, Yicheng Wang*, Tianshi Zheng, Yue Guo, Xin Liu, Yangqiu Song

ACL-2024

Abductive Reasoning

- **Abductive reasoning** is a form of reasoning that seeks to find the best explanation for an observation, distinct from the other two major types of inference: deduction and induction.

- Abductive Reasoning on KG:

  To use **structured knowledge in KG** to explain **observations**.

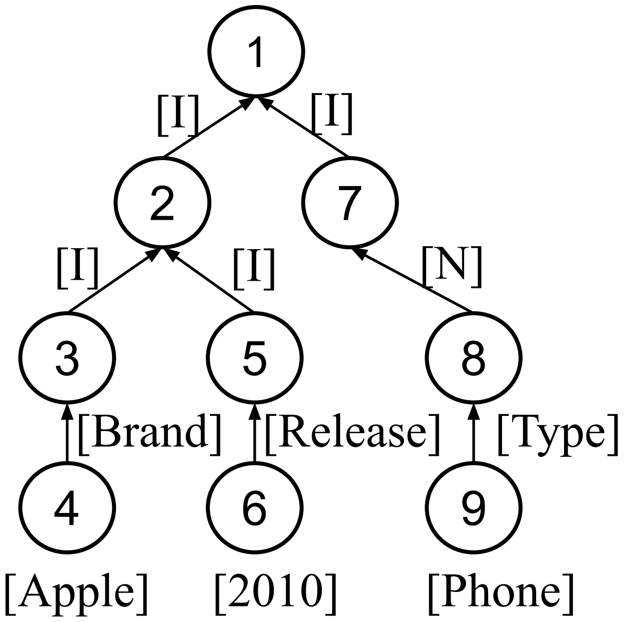
Abductive Reasoning

- Abductive Reasoning on KG:

  To use **structured knowledge in KG** to explain **observations**.

Observations (O)	Hypotheses (H)	Hypotheses Interpretations
		The actors and screenwriters born in Los Angeles
		The Apple products released in 2010 that are not phones
		The disease whose symptoms can be relieved by Panadol

Tokenization of Hypothesis



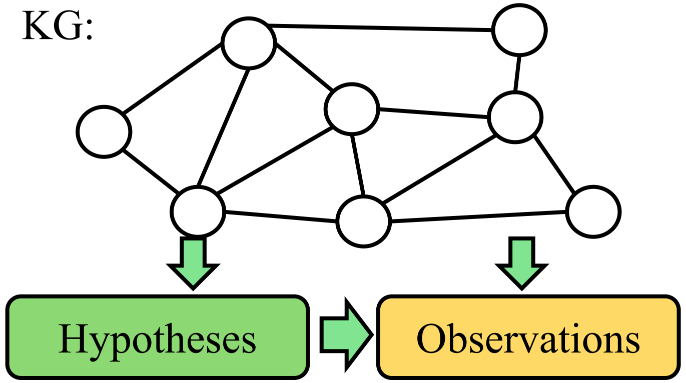
Nodes	Actions	Stack
1	[I]	1
2	[I]	1,2
3	[Brand]	1,2,3
4	[Apple]	1,2,3,4 $\rightarrow$ 1,2
5	[Release]	1,2,5
6	[2010]	1,2,5,6 $\rightarrow$ 1
7	[N]	1,7
8	[Type]	1,7,8
9	[Phone]	1,7,8,9 $\rightarrow$ empty

Hypothesis: $H = V_? : Brand(V_?, Apple) \wedge Release\ (V_?, 2010) \wedge \neg Type(V_?, Phone)$

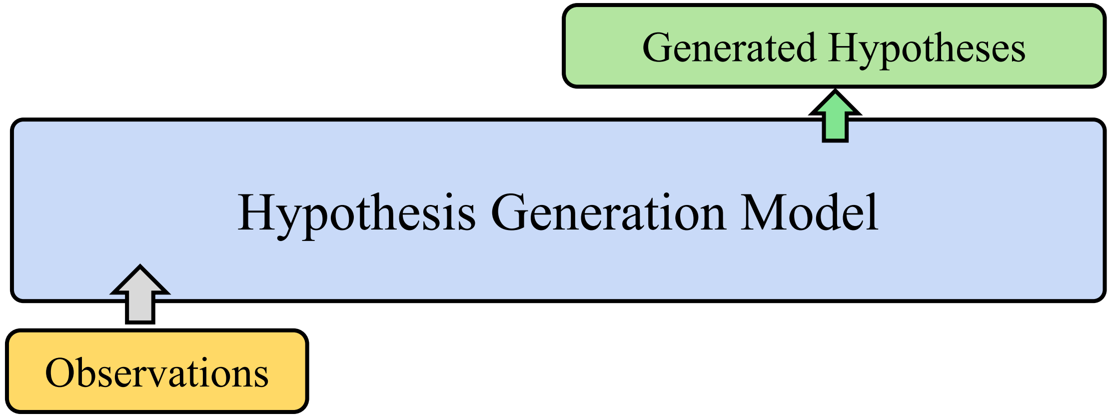
Tokens : [I][I][Brand][Apple][Release][2010][N][Type][Phone]

Complex Logical Hypothesis Generation

Step 1:
Sample observation-hypothesis pairs.



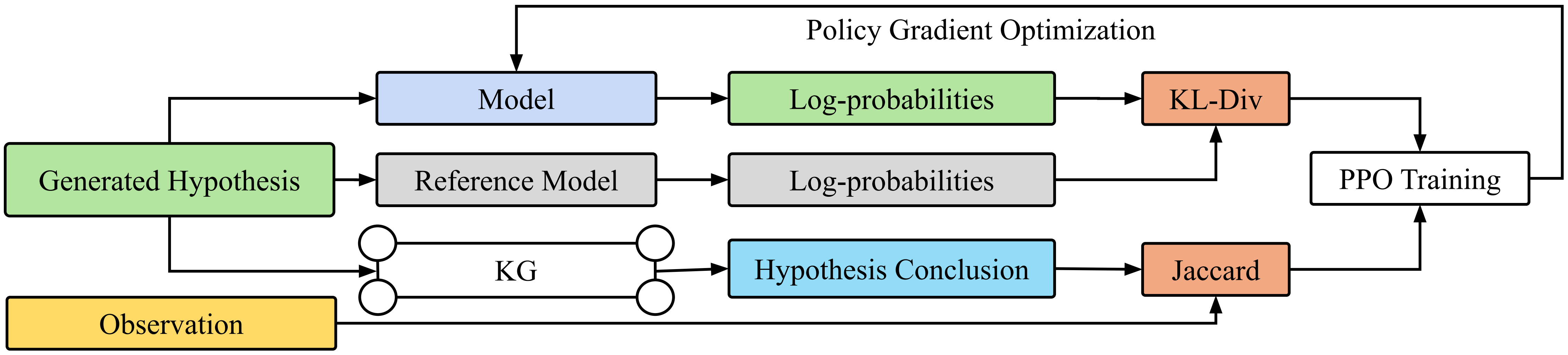
Step 2:
Train hypothesis generation model by using teacher forcing.



$$L = \log \rho(h|o) = \sum_{i=1}^{n} \log \rho(h_i|o, h_1, h_2, \dots, h_{i-1})$$

Complex Logical Hypothesis Generation

Step 3:
Optimize hypothesis generation model with Reinforcement Learning From Knowledge Graph feedback (RLF-KG).



$$r(h, o) = \mathrm{Jaccard}([\mathrm{H}]_G, \mathrm{O})$$

$$E_{o\sim D, h\sim\pi(\cdot|o)}\left[r(h, o) - \beta \log \frac{\pi(h|o)}{\rho(h|o)}\right]$$

Dataset

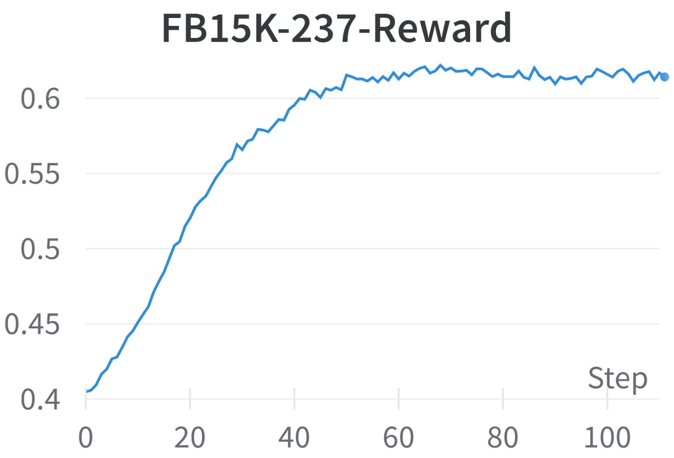
Dataset	Relations	Entities	Training	Validation	Testing	All Edges
FB15k-237	237	14,505	496,126	62,016	62,016	620,158
WN18RR	11	40,559	148,132	18,516	18,516	185,164
DBpedia50	351	24,624	55,074	6,884	6,884	68,842

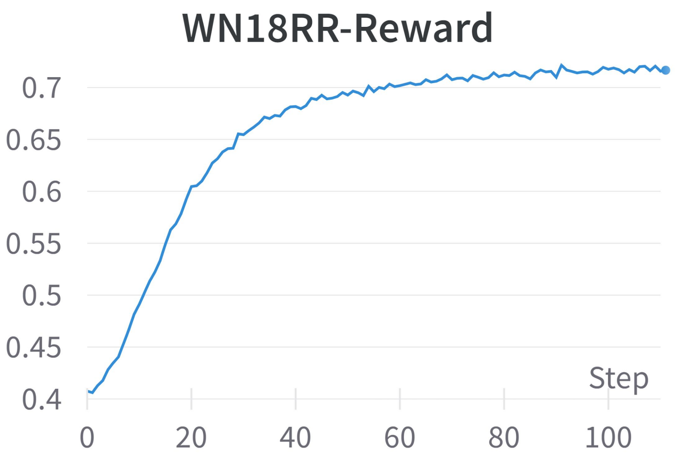
This figure provides basic information about the three knowledge graphs utilized in our experiments. The graphs are divided into standard sets of training, validation, and testing edges to facilitate the evaluation process.

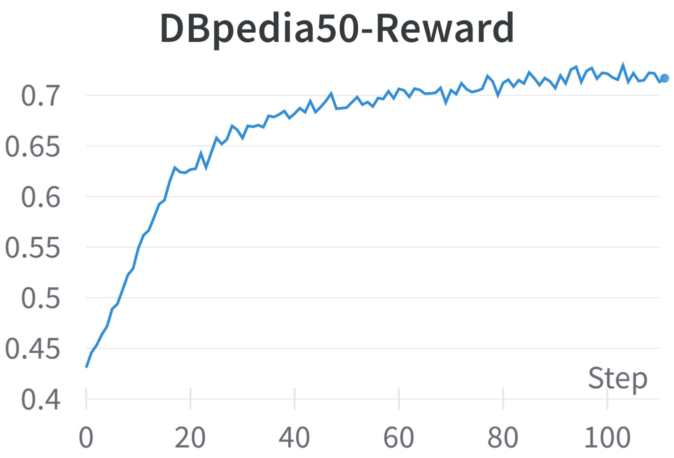
Dataset	Training Samples		Validation Samples		Testing Samples	
	Each Type	Total	Each Type	Total	Each Type	Total
FB15k-237	496,126	6,449,638	62,015	806,195	62,015	806,195
WN18RR	148,132	1,925,716	18,516	240,708	18,516	240,708
DBpedia50	55,028	715,364	6,878	89,414	6,878	89,414

The detailed information about the queries used for training, validation, and testing.

Performance







Dataset	Model	1p	2p	2i	3i	ip	pi	2u	up	2in	3in	pni	pin	inp	Ave.
FB15k-237	Enc.-Dec.	0.626	0.617	0.551	0.513	0.576	0.493	0.818	0.613	0.532	0.451	0.499	0.529	0.533	0.565
	+ RLF-KG	**0.855**	**0.711**	**0.661**	**0.595**	**0.715**	**0.608**	**0.776**	**0.698**	**0.670**	**0.530**	**0.617**	**0.590**	**0.637**	**0.666**
	Dec.-Only	0.666	0.643	0.593	0.554	0.612	0.533	0.807	0.638	0.588	0.503	0.549	0.559	0.564	0.601
	+ RLF-KG	**0.789**	**0.681**	**0.656**	**0.605**	**0.683**	**0.600**	**0.817**	**0.672**	**0.672**	**0.560**	**0.627**	**0.596**	**0.626**	**0.660**

Real Examples:

Sample	Interpretation	Companies operating in industries that intersect with Yahoo! but not with IBM.	
	Hypothesis	The observations are the $V_?$ such that $\exists V_1, inIndustry(V_1, V_?) \land \neg industryOf(IBM, V_1) \land industryOf(Yahoo!, V_1)$	
	Observation	EMI, Columbia, Viacom, Yahoo!, Bandai, Rank_Organisation, Gannett_Company, NBCUniversal, Pony_Canyon, The_Graham_Holdings_Company, Televisa, Google, Microsoft_Corporation, Munhwa_Broadcasting_Corporation,	CBS_Corporation, GMA_Network, Victor_Entertainment, Sony_Music_Entertainment_(Japan)_Inc., Toho_Co.,_Ltd., The_New_York_Times_Company, Star_Cinema, TV5, Avex_Trax, The_Walt_Disney_Company, Metro-Goldwyn-Mayer, Time_Warner, Dell, News_Corporation

Output from Supervised Training:

Enc.-Dec.	Interpretation	Companies operating in industries that intersect with Yahoo! but not with Microsoft Corporation.
	Hypothesis	The observations are the $V_?$ such that $\exists V_1, inIndustry(V_1, V_?) \wedge \neg industryOf(Microsoft\_Corporation, V_1) \wedge industryOf(Yahoo!, V_1)$
	Conclusion	Absent: Microsoft_Corporation
	Jaccard	0.964
	Smatch	0.909

Output after RLF-KG Training:

+ RLF-KG	Interpretation	Companies operating in industries that intersect with Yahoo! but not with Oracle Corporation.
	Hypothesis	The observations are the $V_?$ such that $\exists V_1, inIndustry(V_1, V_?) \wedge \neg industryOf(Oracle\_Corporation, V_1) \wedge industryOf(Yahoo!, V_1)$
	Concl.	Same
	Jaccard	1.000
	Smatch	0.909

Roadmap

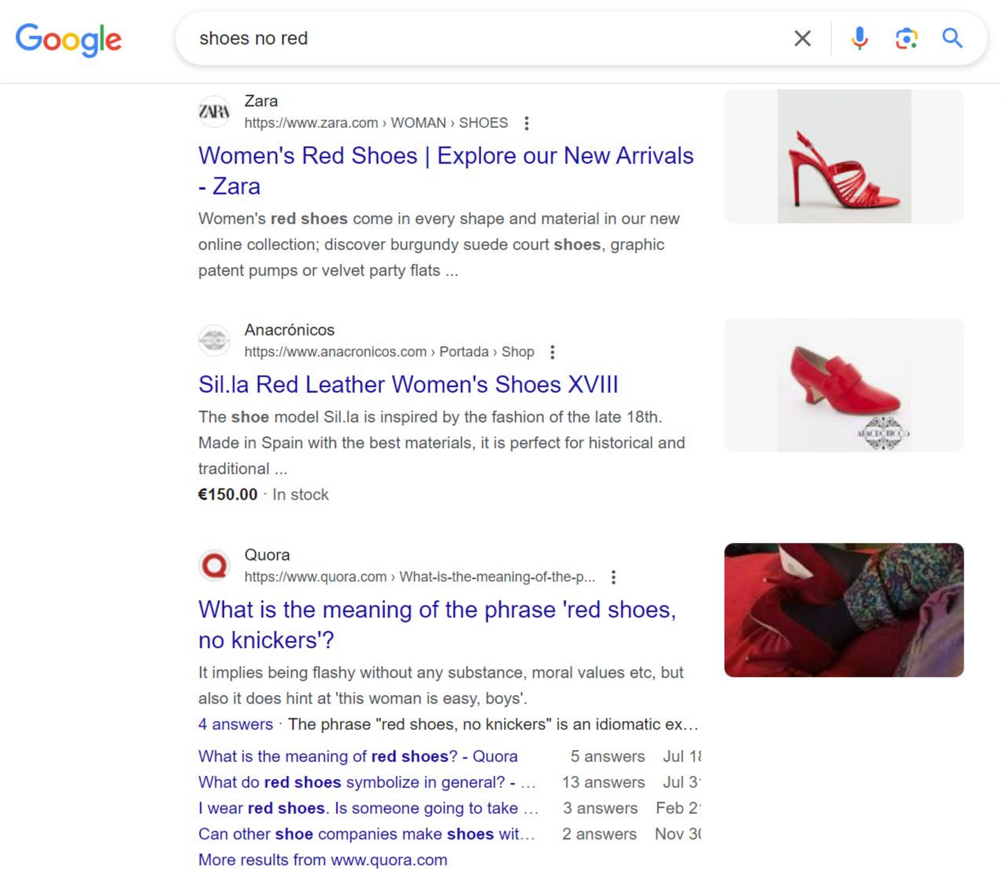
- Background
- Complex Query Answering
- Complex Hypothesis Generation
- Complex Session Intension Understanding
- Future Work

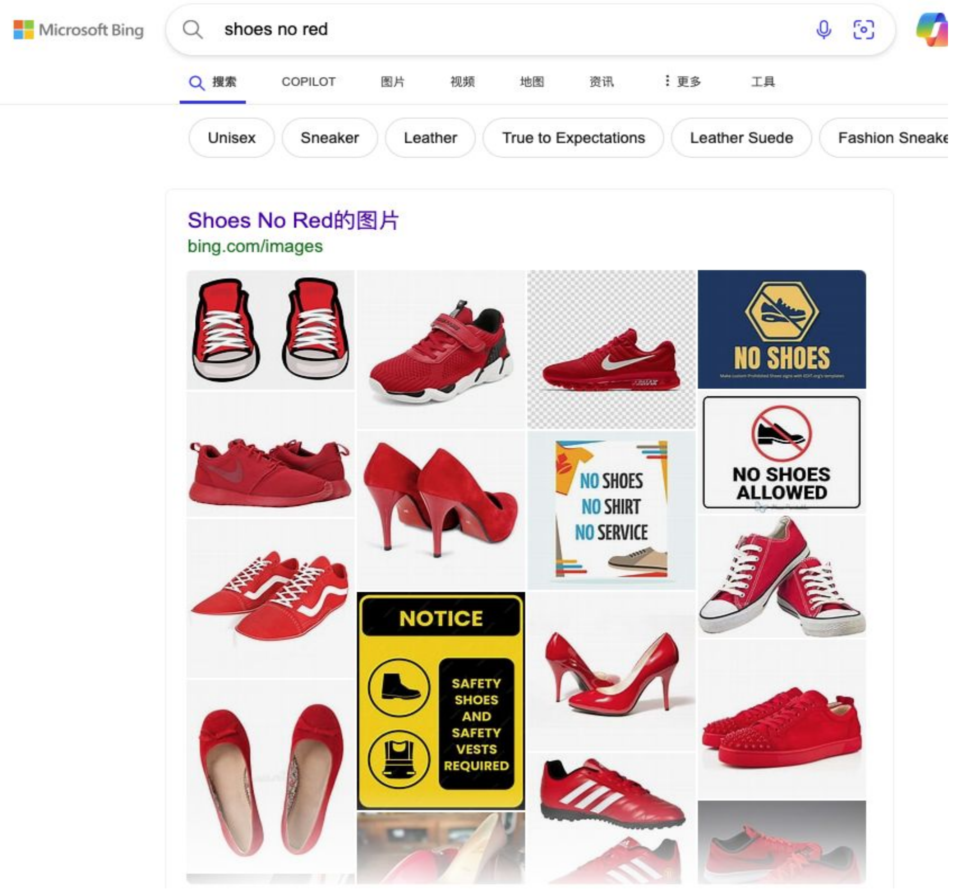
Understanding Inter-Session Intentions via Complex Logical Reasoning

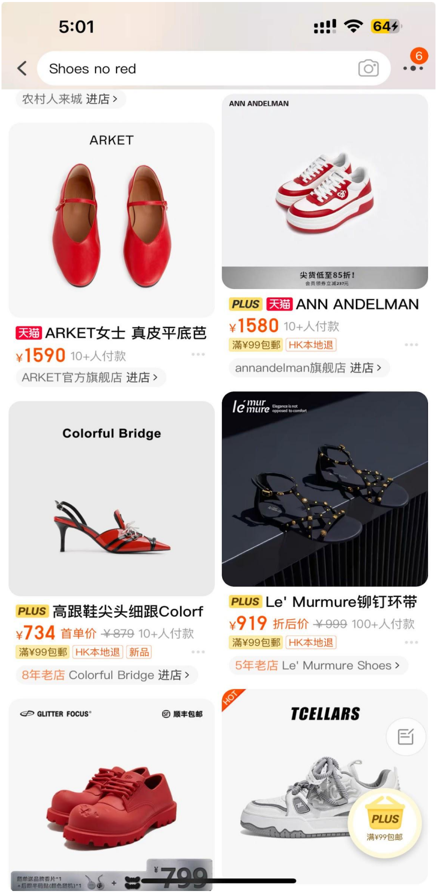
Jiaxin Bai, Chen Luo, Zheng Li, Qingyu Yin, Yangqiu Song

KDD-2024

Search with Logic is Hard







- Search products with complex requirements is hard.
- Particularly hard for requirements like NOT.

Search with hidden intentions from sessions

- Intentions can be hidden and implicit, not always from keywords.
  - We need to use behaviour data, like **session information**, to model intentions.

- Often people do a single decision **across multiple sessions**.
  - People may search and spend **multiple sessions** before making a big decision ,like buying a smartphone or a fridge.
  - If a person buy a mattress in a previous session, he is **not** likely to buy another one, but a bed frame.

- We need a **complex logic reasoning across sessions**.
  - Find the desired item of a **session** with the **brand Nike or Adidas**
  - Find the desired item with **wooden material** that is desired by the **session_1** but is not desired by **session_2**.
- Dealing with **sessions**, **attributes**, and **logics** in one.

Integrating sessions, attributes, and logics

For **Product** Recommendation

Find the next item of a session with the **brand** Nike **or** Adidas

Find next item of a given session

Find an item with the brand Nike

Or

Find an item with the brand Adidas

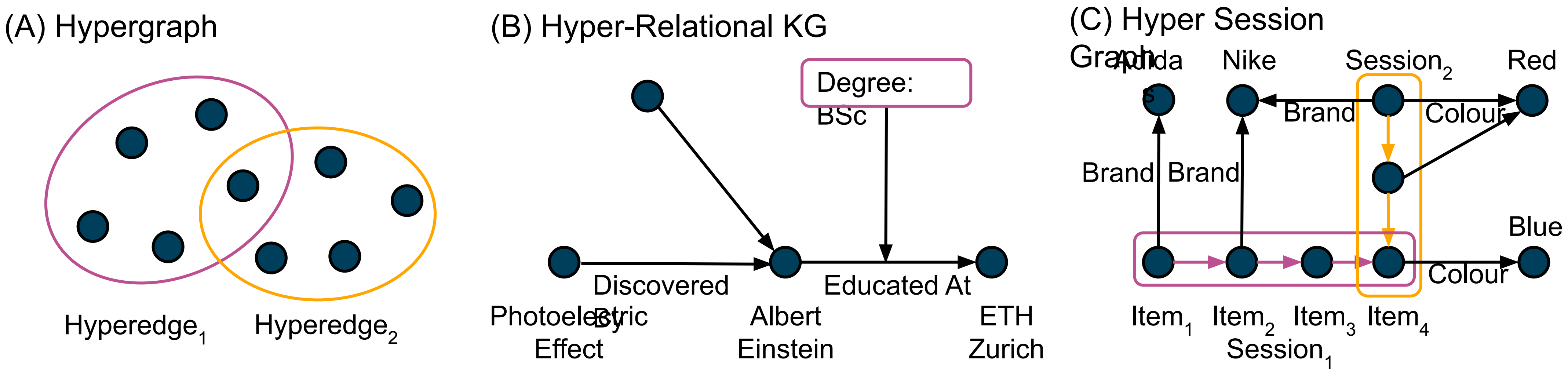
Find an item with the **brand** Adidas **or** Nike

And

Find the next item of a session with the **brand** Nike **or** Adidas

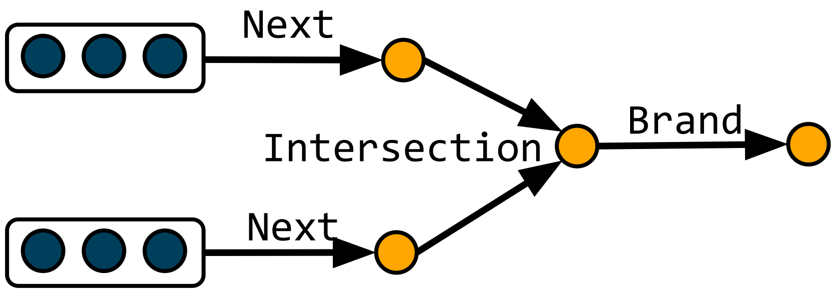
$$q = V_? : Session(Item_1, Item_2, \ldots, Item_k, V_?) \wedge (Brand(V_?, Nike) \vee Brand(V_?, Adidas))$$

Logical Session-CQA on Hypergraph



- Sessions are treated as hyperedges.
- Other attributes and relations are treated as binary edges.

CQA Methods for Inter-Session Logic Reasoning




```
[(] [P] [Brand]
    [(] [I]
      [(] [P] [Next] [(] [S] [Item_{1,1}] … [Item_{1,m}] [)] [)]
      [(] [P] [Next] [(] [S] [Item_{2,1}] … [Item_{2,n-1}] [Item_{2,n}] [)] [)]
    [)]
[)]
```

Session Encoders + Logic Encoders

When doing logic reasoning, the logic encoder can only access **session embedding** but not the **detailed items** in the session in the reasoning process.

N-ary QE methods

StarQE [1] and NQE [2] are designed for hyper-relational KG, **difficult to be adopted to session graphs**.

SQE [3] can be extended to N-ary quires, but it cannot capture some important aspects of query graph, like permutation invariance in ***AND*** and ***OR***.

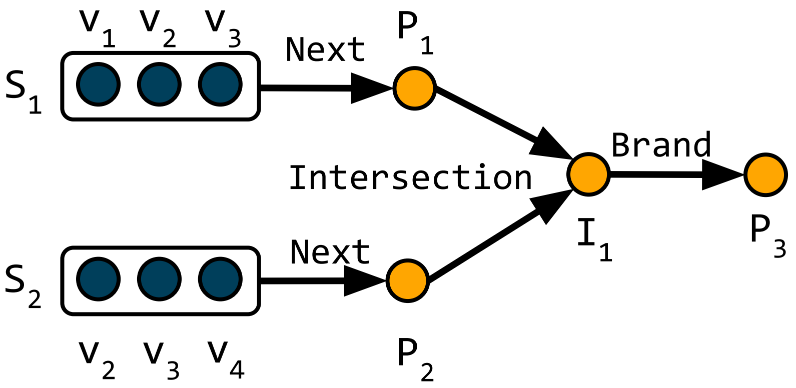
Need a new query encoding method on session hypergraph

[1] Dimitrios Alivanistos, Max Berrendorf, Michael Cochez, Mikhail Galkin: Query Embedding on Hyper-Relational Knowledge Graphs. ICLR 2022
[2] Haoran Luo, Haihong E, Yuhao Yang, Gengxian Zhou, Yikai Guo, Tianyu Yao, Zichen Tang, Xueyuan Lin, Kaiyang Wan: NQE: N-ary Query Embedding for Complex Query Answering over Hyper-Relational Knowledge Graphs. AAAI 2023
[3] Jiaxin Bai, Tianshi Zheng, Yangqiu Song: Sequential Query Encoding for Complex Query Answering on Knowledge Graphs. Trans. Mach. Learn. Res. 2023 (2023)

Logical-Session Graph Transformer

$$q = V_? : Session(Item_{1,1}, Item_{1,2}, \ldots, Item_{1,m}, V_?) \wedge Session(Item_{2,1}, Item_{2,2}, \ldots, Item_{2,n}, V_?)$$

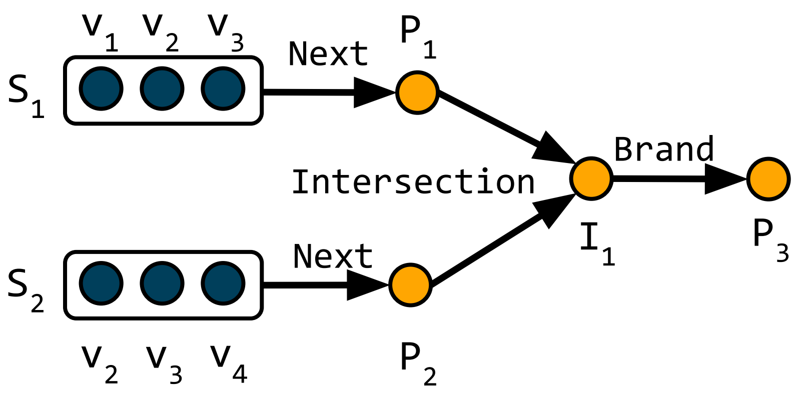
Find the item that is desired by the **session$_1$** and desired by **session$_2$**.



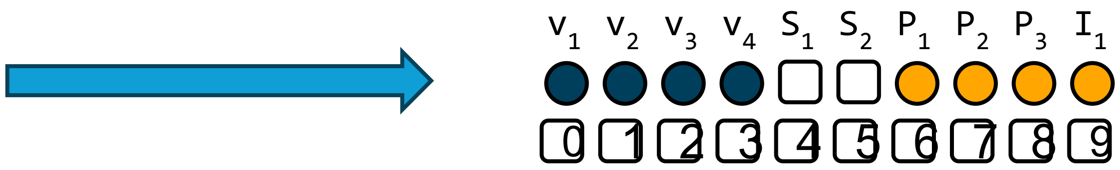
Query Graph

- v_i are product items
- S_i are sessions
- P_i are projection operators
- I_i are intersection operators
- U_i are union operators
- N_i are negation operators

Logical-Session Graph Transformer

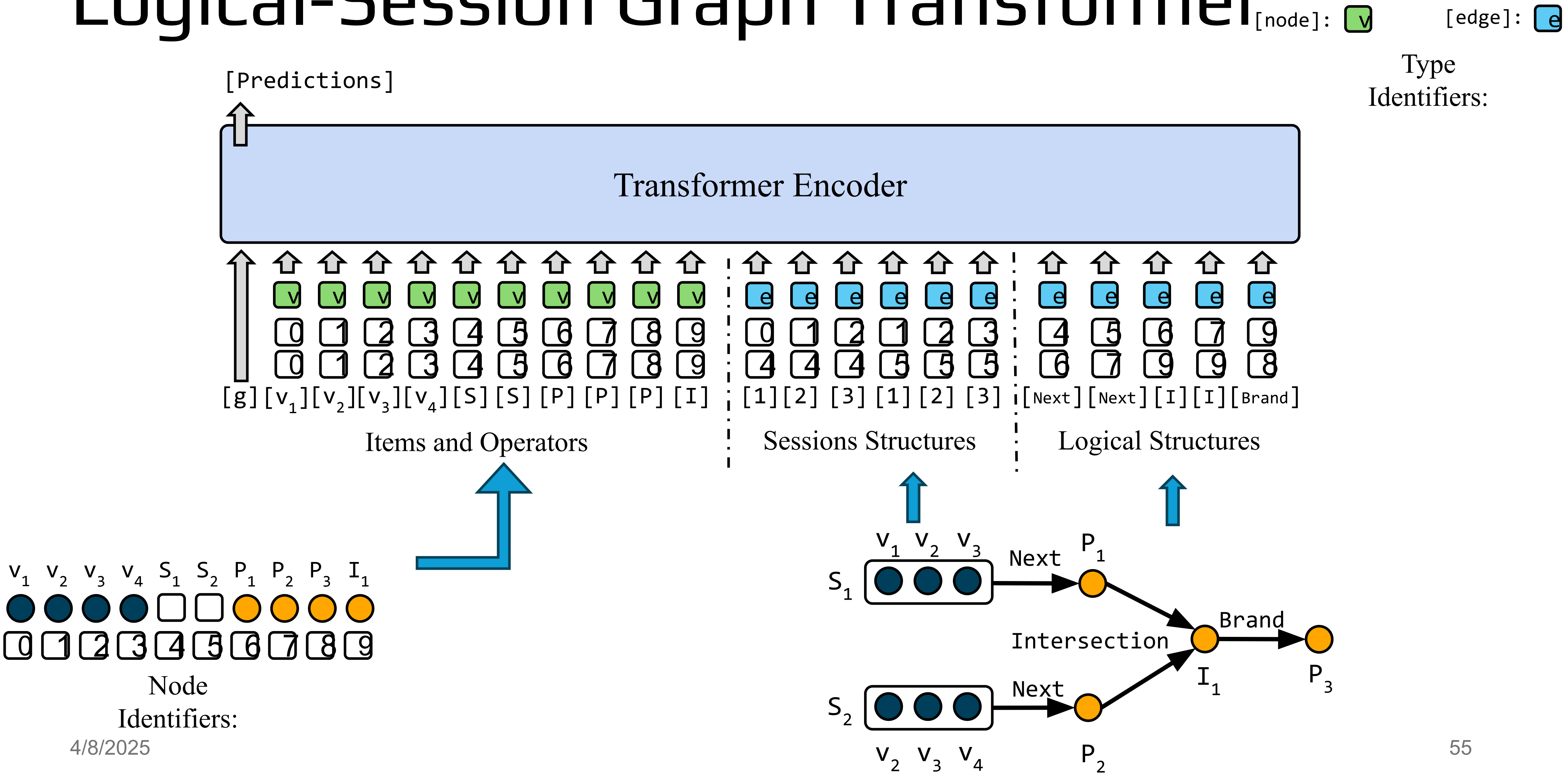


Query Graph

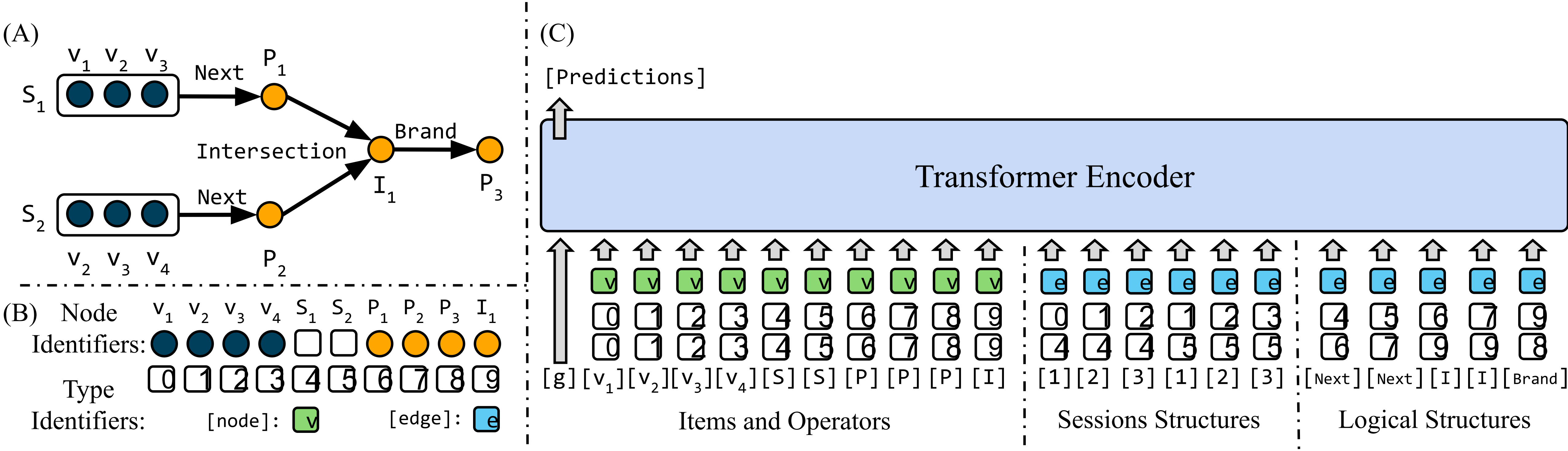


Node identifiers (from 0 to 9) are assigned to each node, involving items, sessions, and operators in the graph

Logical-Session Graph Transformer



Logical-Session Graph Transformer



- Permutation Invariant to Logical Operators.
- Sensitive to Item-order in Session.
- Specialised Token Graph Transformer [1] for encoding query graph

[1] Jinwoo Kim, Dat Nguyen, Seonwoo Min, Sungjun Cho, Moontae Lee, Honglak Lee, Seunghoon Hong: Pure Transformers are Powerful Graph Learners. NeurIPS 2022

Experiment Dataset

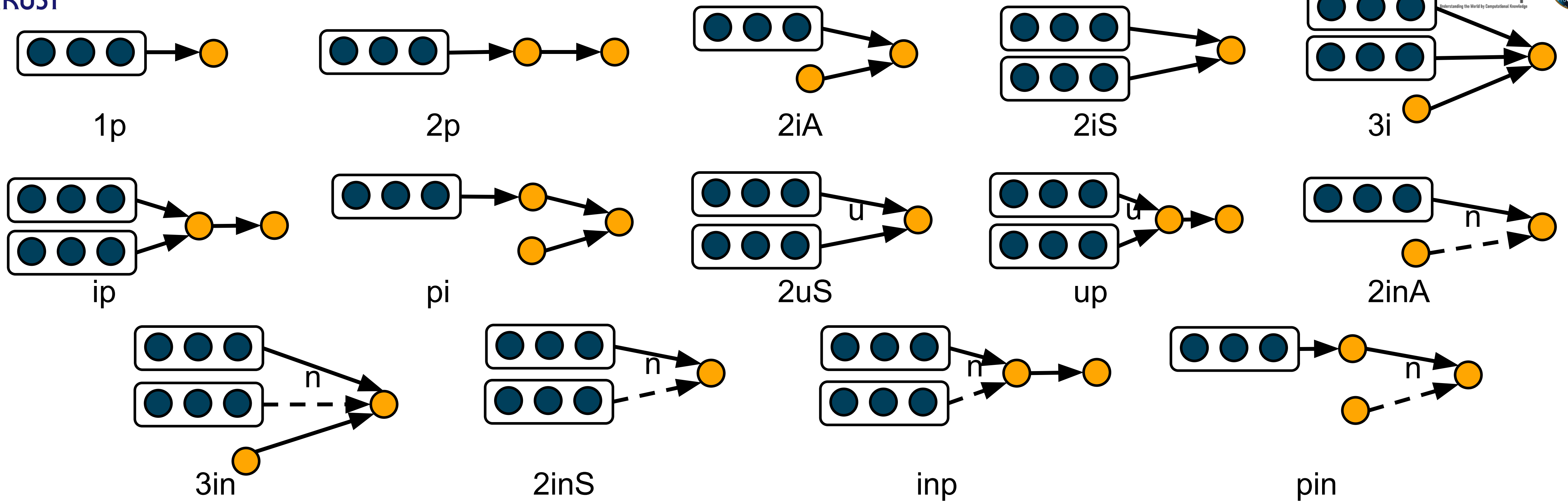
- Diginetica Dataset
  - The Diginetica dataset is commonly used in session-based recommendation research. It originates from the CIKM Cup 2016, focusing on e-commerce applications. The dataset contains user sessions, item interactions, and transaction information.
- Dressipi Dataset
  - The Dressipi dataset is a fashion-oriented dataset designed for personalized recommendation systems. It comes from the clothing and fashion domain, making it relevant for session-based recommendations in retail.
- Amazon M2 Dataset
  - The Amazon M2 dataset (KDD Cup 2023) focuses specifically on user sessions and interactions with products, making it suitable for session-based and sequential recommendation tasks. The dataset is often used in research on e-commerce platforms to enhance recommendations by modeling user behavior over time.

Experiment Dataset

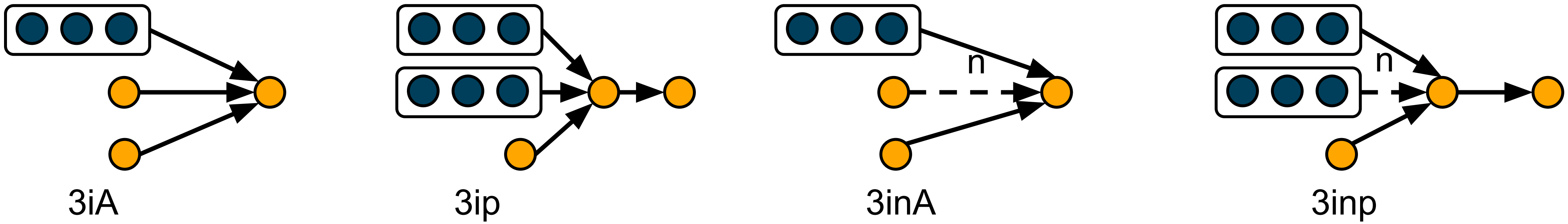
	Train Graph		Validation Graph		Test Graph					
Dataset	Vertices	Edges	Vertices	Edges	Vertices	Edges	#Sessions	#Items	#Values	#Relations
Amazon	2,258,179	7,234,680	2,345,475	7,620,527	2,431,747	8,004,984	720,816	431,036	1,279,895	10
Diginetica	257,018	1,286,384	261,996	1,337,628	266,897	1,387,861	12,047	134,904	125,204	3
Dressipi	611,520	2,435,932	643,140	2,567,128	674,853	2,698,692	668,650	23,618	903	74

	Train Queries		Validation Queries	Test Queries
Dataset	Item-Attributes	Others	All Types	All Types
Amazon	2,535,506	720,816	36,041	36,041
Diginetica	249,562	60,235	3,012	3,012
Dressipi	414,083	668,650	33,433	33,433

Supervised Training Types:



Zero-shot Types:



Experiment Results

Dataset	Query Encoder	Session Encoder	Average-EPFO	Average-Negation
Amazon	FuzzQE	GRURec	30.94	21.03
		SRGNN	31.75	22.26
		Attn-Mixer	31.68	25.00
	Q2P	GRURec	23.09	14.61
		SRGNN	24.59	16.62
		Attn-Mixer	28.16	26.95
	NQE	-	23.19	18.12
	SQE-Transformer	-	30.07	27.16
	SQE-LSTM	-	32.53	27.13
	LSGT (Ours)		**33.26**	**29.69**

Existential Positive First Order (EPFO): queries types that involves conjunction, disjunction, and variables are existential quantified. No negations.

Out-of-Distribution Queries Results

Dataset	Query Encoder	3iA	3ip	3inA	3inp	Average OOD
Amazon	FuzzQE + Attn-Mixer	66.72	29.67	54.33	48.76	49.87
	Q2P + Attn-Mixer	33.51	11.42	51.47	41.46	34.47
	NQE	61.72	1.98	46.47	34,04	36.72
	SQE + Transformers	66.03	28.41	55.61	51.28	50.33
	LSGT (Ours)	**68.44**	**34.22**	**58.50**	**51.49**	**53.16**
Diginetica	FuzzQE + Attn-Mixer	88.30	32.88	82.75	34.50	59.61
	Q2P + Attn-Mixer	40.28	**43.93**	54.31	**48.20**	46.68
	NQE	86.25	20.79	64.74	20.93	48.18
	SQE + Transformers	88.05	31.33	81.77	35.83	59.25
	LSGT (Ours)	**91.71**	35.24	**83.30**	41.05	**62.83**
Dressipi	FuzzQE + Attn-Mixer	65.43	95.64	53.36	97.75	78.05
	Q2P + Attn-Mixer	60.64	96.78	52.22	97.28	76.73
	NQE	31.96	96.18	9.89	97.80	58.96
	SQE + Transformers	72.61	97.12	55.20	98.14	80.77
	LSGT (Ours)	**74.34**	**97.30**	**58.30**	**98.23**	**82.04**

OOD query types: queries types that are not trained during the training phrase.
As their sub-queries are trained, we can use this as a measure of compositional generalization.

Roadmap

- Background
- Complex Query Answering
- Complex Hypothesis Generation
- Session Intension Modeling by Logic
- Future Work

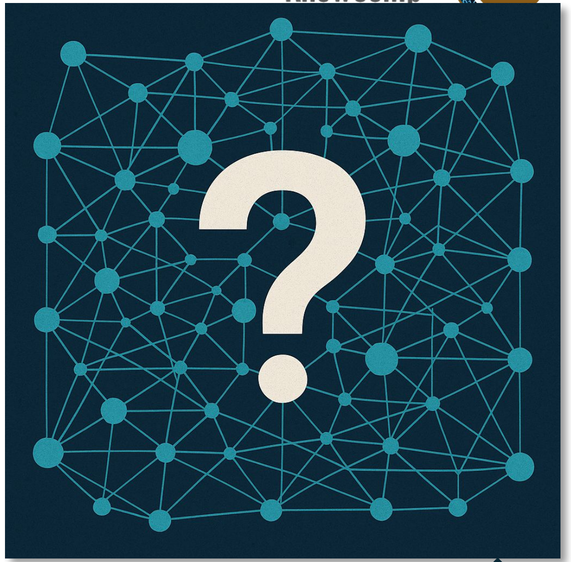
Future Work: Neural Graph Database

Large Language Model

Ilya Sutskever [1]

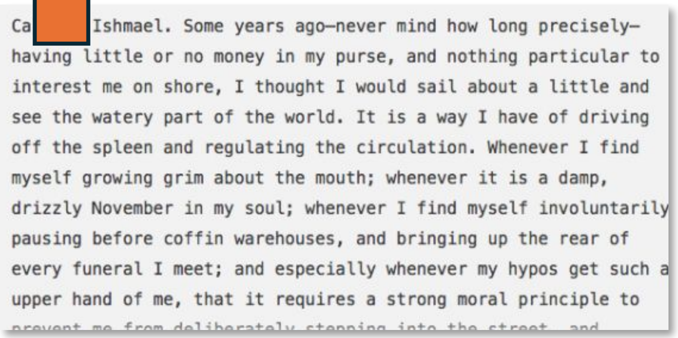
LLM Pre-training scaling will stop
because we only have one internet!

Neural Graph Database (NGDB) show great potential to scale to Database Foundation Model!



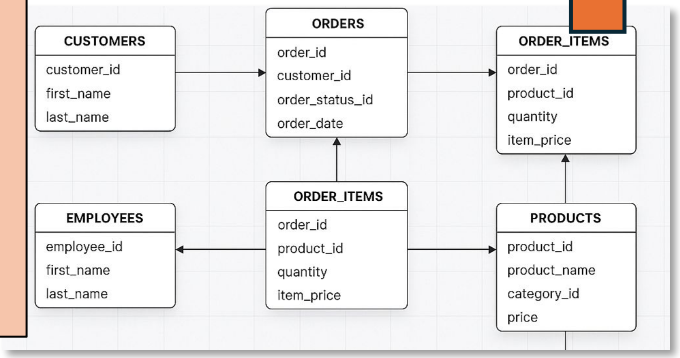
Neural Graph Database
 Database Foundation Model

Text Data


Ca... Ishmael. Some years ago—never mind how long precisely—having little or no money in my purse, and nothing particular to interest me on shore, I thought I would sail about a little and see the watery part of the world. It is a way I have of driving off the spleen and regulating the circulation. Whenever I find myself growing grim about the mouth; whenever it is a damp, drizzly November in my soul; whenever I find myself involuntarily pausing before coffin warehouses, and bringing up the rear of every funeral I meet; and especially whenever my hypos get such a upper hand of me, that it requires a strong moral principle to

- Data within databases **continues to grow** and won't be easily depleted;
- At **very early stage** where universities can do research [2];
- Combines **general applicability** to real-world applications, positioning it to make industry impact.

Database Data



[2] Wang, Y., Wang, X., Gan, Q., Wang, M., Yang, Q., Wipf, D., & Zhang, M. (2025). Griffin: Towards a Graph-Centric Relational Database Foundation Model. arXiv preprint arXiv:2505.05568.

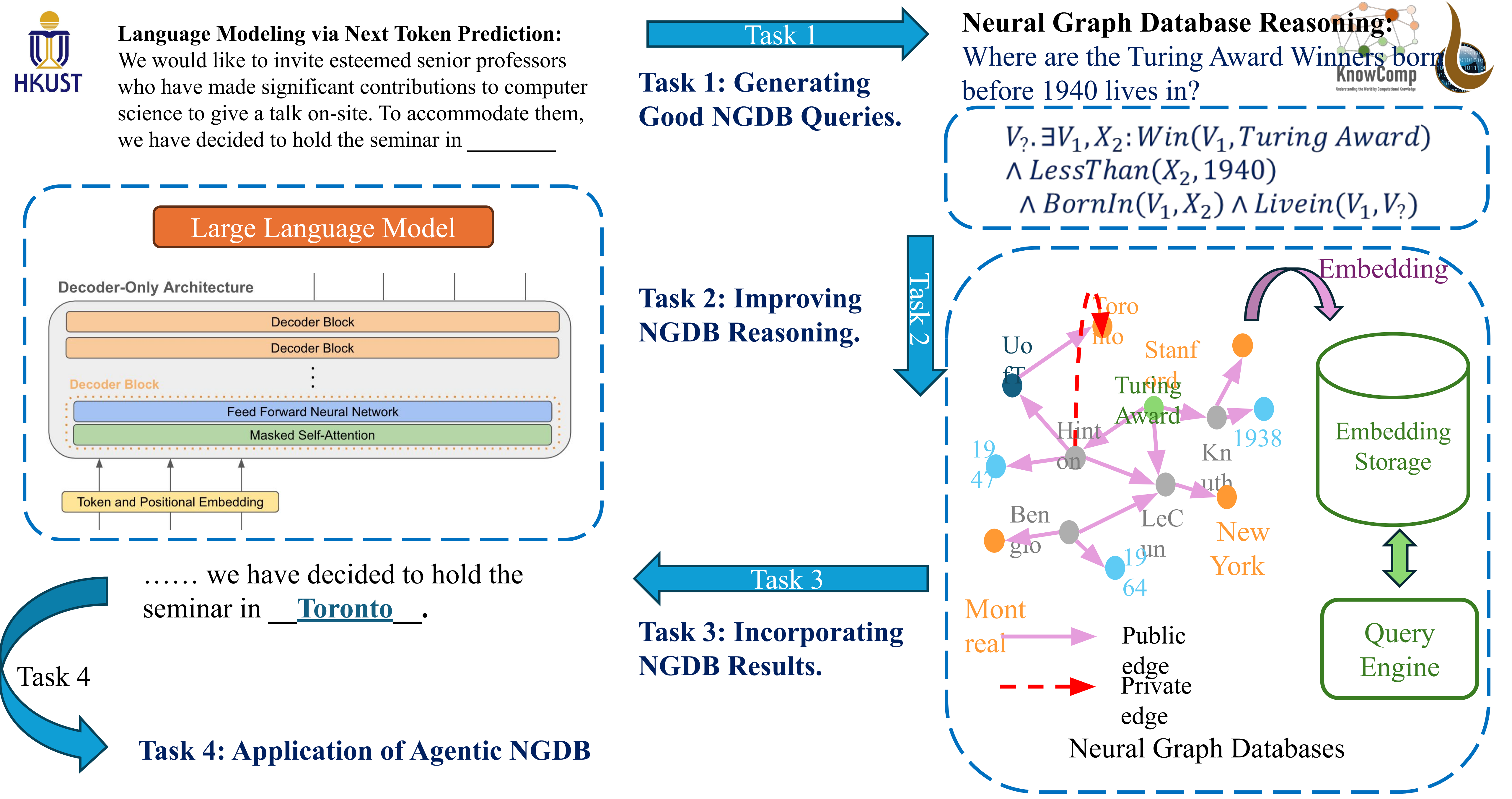
Future Work – Agentic Database

agent
/ˈeɪdʒ(ə)nt/

noun

1. a person who acts on behalf of another person or group.
   "in the event of illness, a durable power of attorney enabled her nephew to act as her agent"
   Similar: representative, negotiator, business manager, emissary, envoy
2. a person or thing that takes an active role or produces a specified effect.
   "these teachers view themselves as agents of social change"
   Similar: medium, means, instrument, vehicle, power, force

- Ask the "correct" question on behalf of the user based on a context
  - The future direction for abductive reasoning and beyond
- Conduct approximation and inferences given various typed data
  - The future direction for complex logical query answering
- Integrate results on behalf of the user and present to them
  - The future direction for KG reasoning together with the LLMs



HKUST
Language Modeling via Next Token Prediction:
We would like to invite esteemed senior professors who have made significant contributions to computer science to give a talk on-site. To accommodate them, we have decided to hold the seminar in ________
Task 1
Task 1: Generating Good NGDB Queries.
Neural Graph Database Reasoning:
Where are the Turing Award Winners born before 1940 lives in?
KnowComp
$V_?.\exists V_1, X_2: Win(V_1, Turing\ Award) \wedge LessThan(X_2, 1940) \wedge BornIn(V_1, X_2) \wedge Livein(V_1, V_?)$
Large Language Model
Decoder-Only Architecture
Decoder Block
Decoder Block
Decoder Block
Feed Forward Neural Network
Masked Self-Attention
Token and Positional Embedding
Task 2: Improving NGDB Reasoning.
Task 2
Embedding
Toronto
UofT
Stanford
Turing Award
Hinton
Knuth
1938
1947
Embedding Storage
Bengio
LeCun
1964
New York
Montreal
Public edge
Private edge
Query Engine
Neural Graph Databases
Task 3
Task 3: Incorporating NGDB Results.
...... we have decided to hold the seminar in __Toronto__.
Task 4
Task 4: Application of Agentic NGDB

Thank you for your attention!
☺️

More related work on my website:
bjx.fun