KDD2021

Neural-Answering Logical Queries on Knowledge Graphs

Lihui Liu (UIUC)

Boxin Du (UIUC)

Heng Ji (UIUC)

ChengXiang Zhai (UIUC)

Hanghang Tong (UIUC)

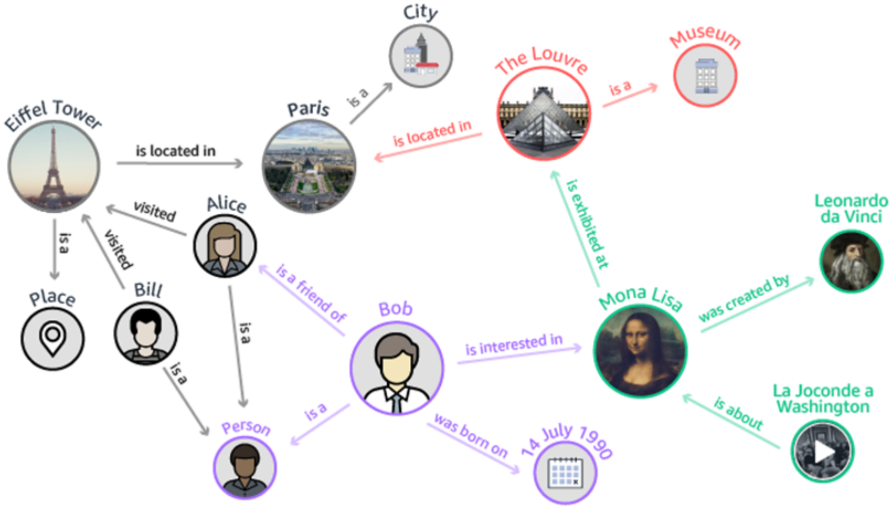




ILLINOIS

Knowledge Graph

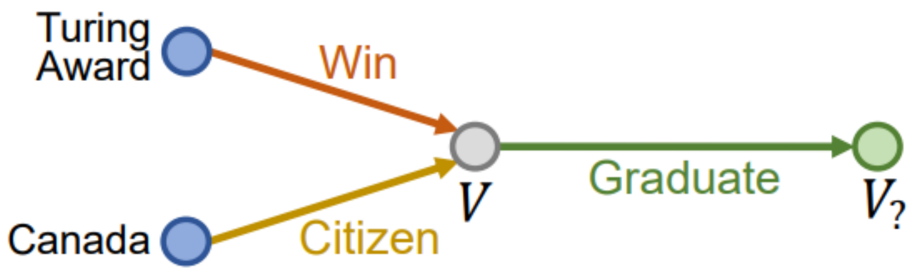
- KG = collection of interlinked entities
  - Objects, events or concepts
  - Multiple types of entities and relations exist
- Facts are represented as triples (h, r, t)
  - <'Paris', 'is_a', 'city'>
  - <'Alice', 'is_friend_of', 'Bob'>
  - …



Logical Query

- Logical query
  - First-order logic with existential quantifier (∃), conjunction (^), and disjunction (v).
  - "where did all Canadian citizens with Turing Award graduate?"

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



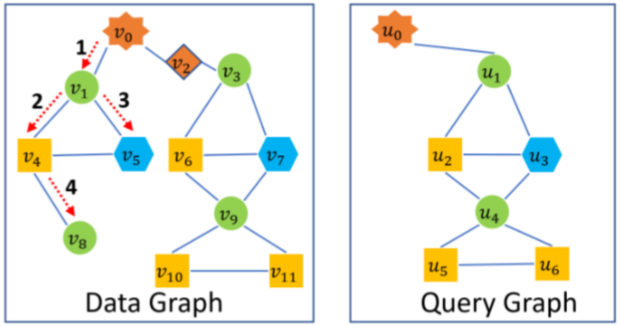
Challenges for Logical Query

- C1: Heterogeneity: Lack of schema, or quite large schema (65k for DBpedia)
- C2: Noise and incompleteness
  - <'Alan Turing', 'wasBornIn', 'United Kingdom'>
  - <'Computer Scientist Alan Turing', 'livesIn', 'London'>
- C3: Massive Size
  - Google knowledge graph: 570 million entities and 18 billion facts
  - Yago: 10 million entities and 120 million facts
- C4: Fast query time

- https://web.stanford.edu/class/cs520/2020/abstracts/leskovec.pdf
- http://snap.stanford.edu/class/cs224w-2019/slides/17-knowledge.pdf

Previous Methods

- Subgraph matching based method:
  - Basic idea: find answers according to the query graph



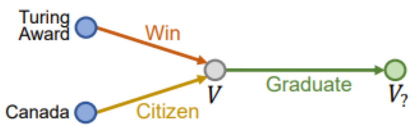
  - Advantages
    - High accuracy
    - No training time
  - Limitations
    - Knowledge graphs are incomplete and noisy
    - Suffer from empty-answer problem
    - High online query time

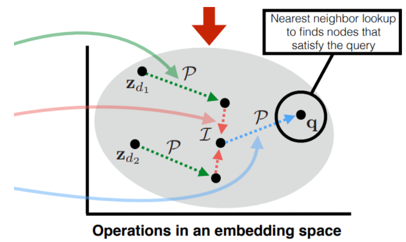
- L. Liu, B. Du, H. Tong. 2018. Approximated Attributed Subgraph Matching. (BigData'18)

Previous Methods

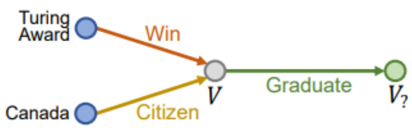
- Embedding based method:
  - Basic idea: embed the query graph into the embedding space

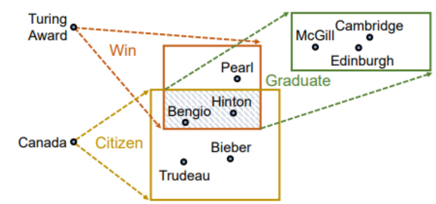
GQE





Query2Box





  - Advantages
    - Answers could be found even when the knowledge graph is incomplete or noisy
    - Have a faster online response

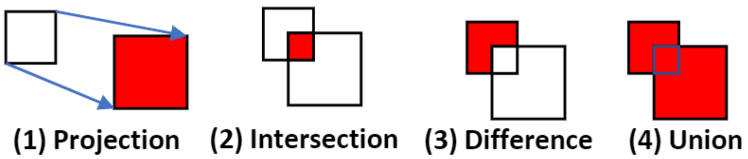
- W Hamilton, P Bajaj, M Zitnik, D Jurafsky, and J Leskovec. 2018. Embedding Logical Queries on Knowledge Graphs(NIPS'18).
- H. Ren, W. Hu, and J. Leskovec. 2020. Query2box: Reasoning over Knowledge Graphs in Vector Space using Box Embeddings. (ICLR'20)

Previous Methods

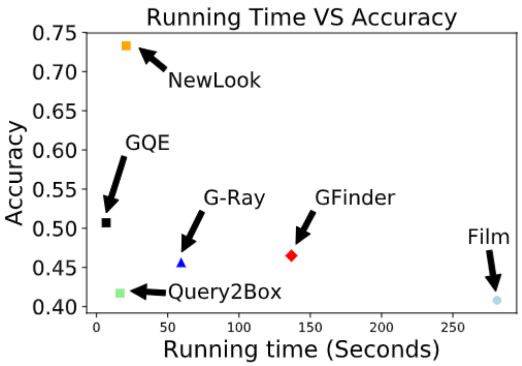
- Embedding based method:
  - limitations
    - GQE and Query2Box (Q2B)
      - Only handle a subset of logical operations
      - Quantifier (∃), conjunction (^) and disjunction (v)
  - Difference operation
    - Given sets $s_1, \ldots, s_n$
    - Find $s_1 - s_2 - \ldots - s_n$
    - "Who won the Turing Award but did not major in computer science?"

Our Work: Key Advantages

- Applicability
  - Support more logical operations.
    - quantifier (∃): projection
    - conjunction (^): intersection
    - disjunction (v): union
    - Difference (-)



- Effectiveness
  - Has high accuracy compared with existing methods
- Efficiency
  - Fast offline training time
  - Fast online query time

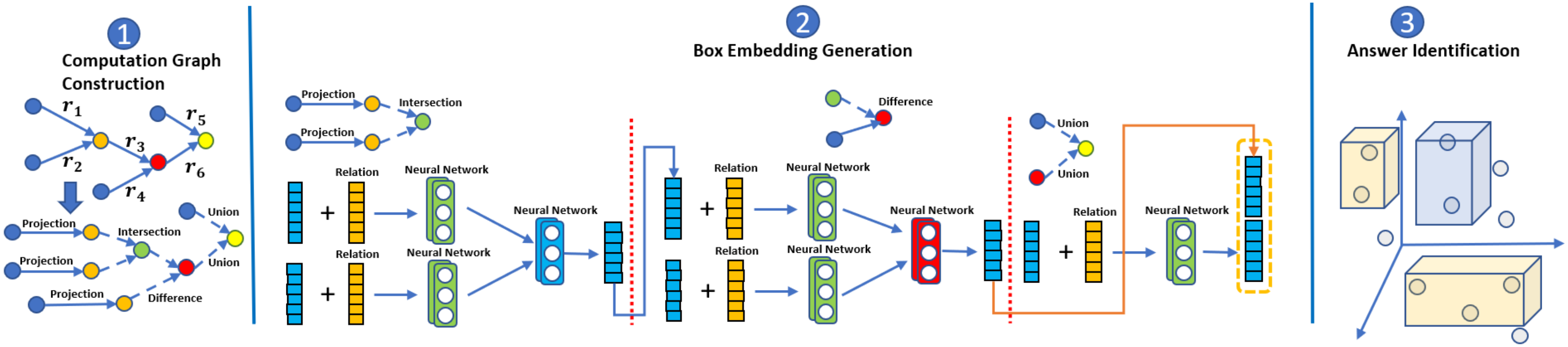


Outline

- Motivations
- Proposed Model: NewLook
- Experiments
- Conclusion

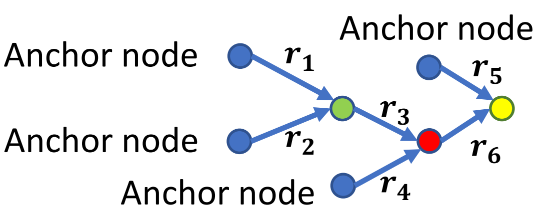
Prob. Def.: Logical Query embedding

- **Given:**
  - A knowledge graph G=(V,R,T)
  - A logical query graph Q with anchor node(s) and variable node(s)
- **Output:**
  - The box embedding for each variable node in Q
  - The point embedding for each entity v∈ G
  - The box embedding for each relation r∈R

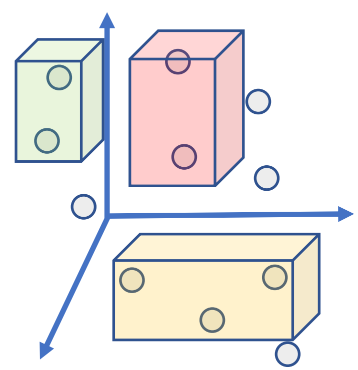


Key idea #1: embedding

- Embed entities in KG as points
- Embed each variable node in query as a box
    - Anchor query node: box with 0 size
- Embed each relation as a box
- Entities that answer the query are inside or close to the boxes



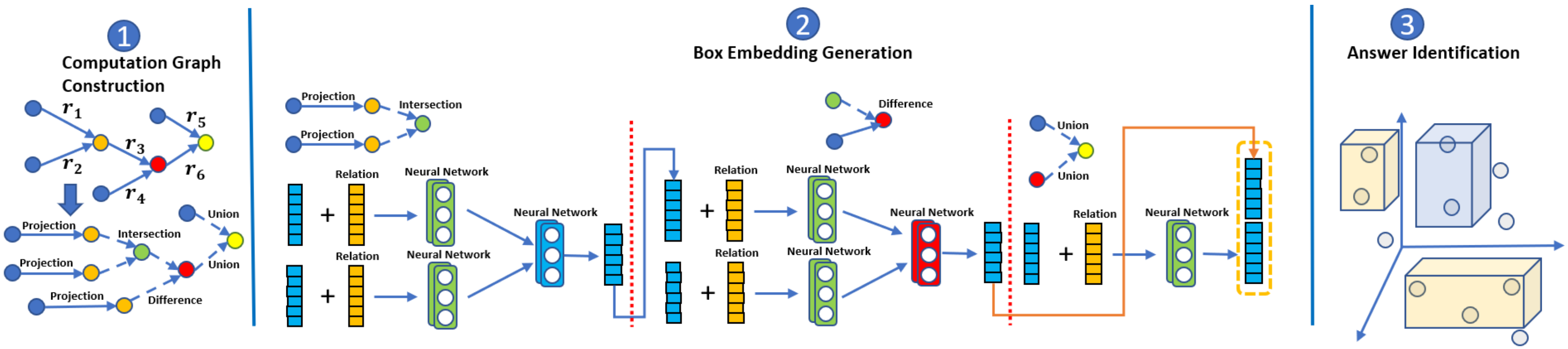
Query Graph



Embedding Space

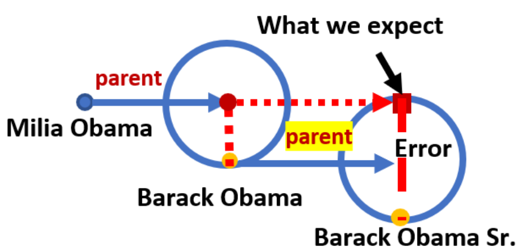
Key idea #2: model each operation as a neural network

- Treat the query graph as a sequence of logical operations
- Execute different operations according to the query graph structure.

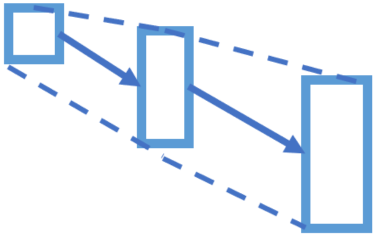


NewLook: Projection Operator

- Geometric Projection Operator
  - Box × Relation → Box
- Problem of existing methods: cascading error
  - TransE: given triple (h, r, t), $e_t = e_h + e_r$
  - Query2Box: given query edge (h, r, t),
    - $b_t^c = b_h^c + p_r^c$
    - $b_t^o = b_h^o + p_r^o$



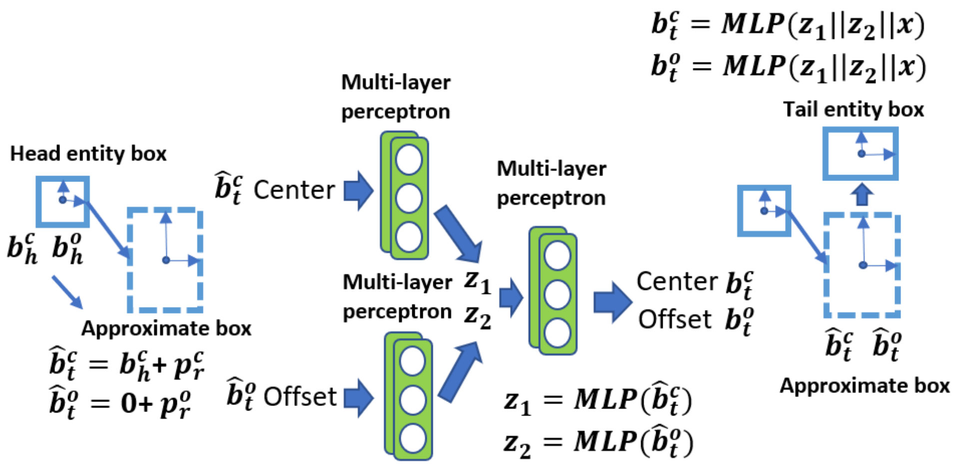
(a)TransE Projection


(b)Query2Box Projection

- B. Antoine, U. Nicolas, G. Alberto, W Jason, and Y. Oksana. Translating Embeddings for Modeling Multi-relational Data. (NIPS ’13).
- H. Ren, W. Hu, and J. Leskovec. 2020. Query2box: Reasoning over Knowledge Graphs in Vector Space using Box Embeddings. (ICLR ’20).

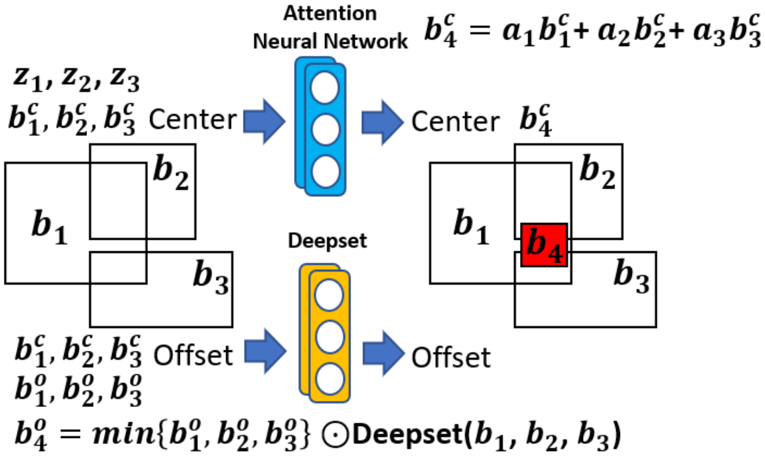
NewLook: Projection Operator

- Geometric Projection Operator
  - Box × Relation → Box
- Our solution
  - Linear transformation: obtain an approximate box embedding
  - Neural network: fine-tune the true box size and position



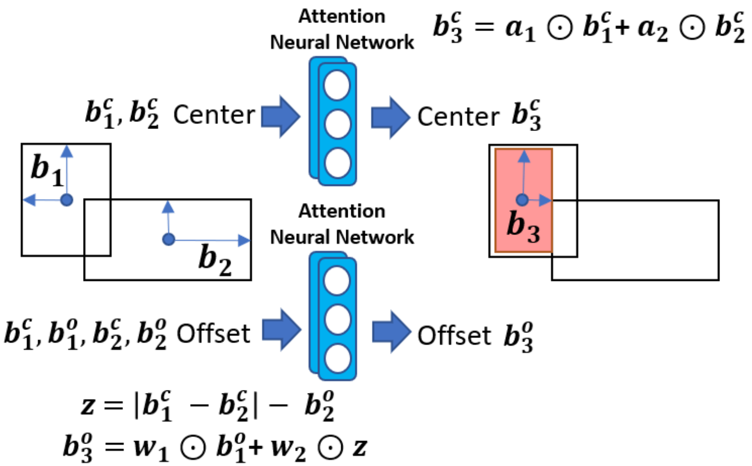
NewLook: Intersection Operator

- Geometric Intersection Operator
  - Box × ⋯× Box → Box
- Our solution
  - Use attention neural network to learn box center
    - Permutation invariant
  - Use Deepsets to learn box offset
    - Permutation invariant
    - The new offset shrinks



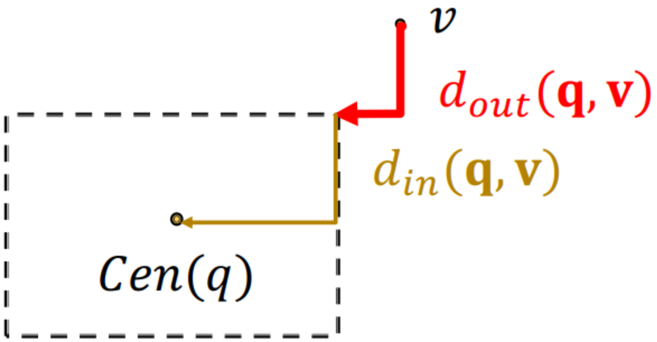
NewLook: Difference Operator

- Geometric Difference Operator
  - Box × ⋯× Box → Box
- Our solution
  - Use attention neural network to learn box center
  - Use attention neural network to learn box offset



NewLook: Training and Evaluation

- Training:
  - Learning from positive and negative query pairs
  - max margin loss
  - Distance between a box q and an entity v
    - $d(q, v) = d_{out}(q, v) + \alpha d_{in}(q, v)$ where $0 < \alpha < 1$
    - Down weight the distance inside the box
    - As long as entity is inside the box, we regard it as "close enough" to the box center



- W Hamilton, P Bajaj, M Zitnik, D Jurafsky, and J Leskovec. 2018. Embedding Logical Queries on Knowledge Graphs(NIPS'18).
- H. Ren, W. Hu, and J. Leskovec. 2020. Query2box: Reasoning over Knowledge Graphs in Vector Space using Box Embeddings. (ICLR'20)

Outline

- Motivations
- Proposed Model: NewLook
- Experiments
- Conclusion

Experiments

- Datasets: FB15k, FB15k-237, NELL995

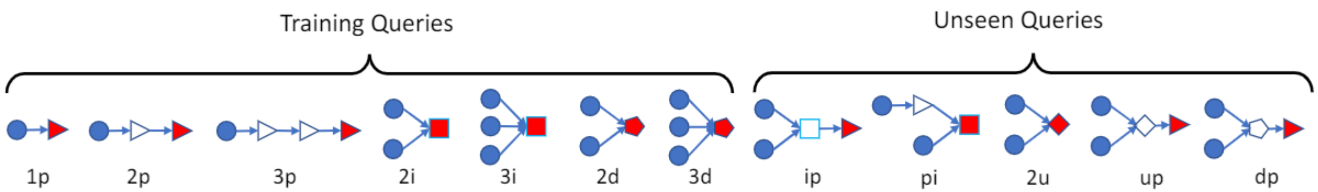
Dataset	Entities	Relations	Training Edges	Validation Edges	Test Edges	Total Edges
FB15k	14,951	1,345	483,142	50,000	59,071	592,213
FB15k-237	14,505	237	272,115	17,526	20,438	310,079
NELL	63,361	200	114,213	14,324	14,267	142,804

- Baselines:
  - Embedding methods
    - GQE [W Hamilton et al. NeurIPS' 18]
    - Query2Box [H. Ren et al. ICLR' 20]
    - BetaE [H. Ren et al. NeurIPS' 21]
    - EmQL [H. Sun et al. NeurIPS' 21]
  - Subgraph matching methods
    - G-Ray [H. Tong et al. KDD' 07]
    - FilM [J. Moorman et al. BigData' 18]
    - Gfinder [L. Liu et al. BigData'18]
- Metrics: Hits@k and MRR

- W Hamilton, P Bajaj, M Zitnik, D Jurafsky, and J Leskovec. 2018. EmbeddingLogical Queries on Knowledge Graphs(NIPS'18).
- H. Ren, W. Hu, and J. Leskovec. 2020. Query2box: Reasoning overKnowledge Graphs in Vector Space using Box Embeddings. (ICLR'20)
- Hongyu Ren and Jure Leskovec. 2020. Beta Embeddings for Multi-Hop LogicalReasoning in Knowledge Graphs. (NeurIPS' 21).
- H. Sun, A. O. Arnold, T. Bedrax-Weiss, F. Pereira, and W. Cohen. 2021. Faithful Embeddings for Knowledge Base Queries (NeurIPS' 21).
- H Tong, C Faloutsos, B Gallagher, and T Eliassi-Rad. 2007. Fast Best-Effort PatternMatching in Large Attributed Graphs. (KDD'07).
- J. D. Moorman, Q. Chen, T. K. Tu, Z. M. Boyd, and A. L. Bertozzi. 2018. Fil-tering Methods for Subgraph Matching on Multiplex Networks. (BigData'18)
- L. Liu, B. Du, H. Tong. 2018. Approximated Attributed Subgraph Matching. (BigData'18)

Queries with A Single Target Variable Node

- Query set:
  - 7 training query structures
  - 12 testing query structures

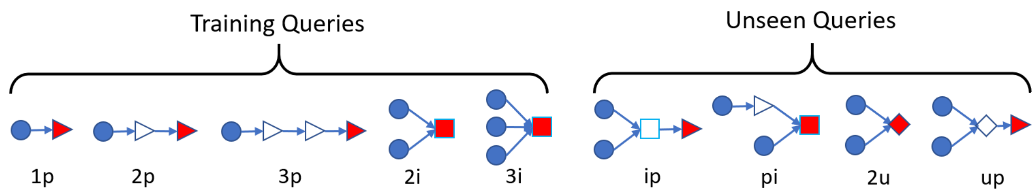


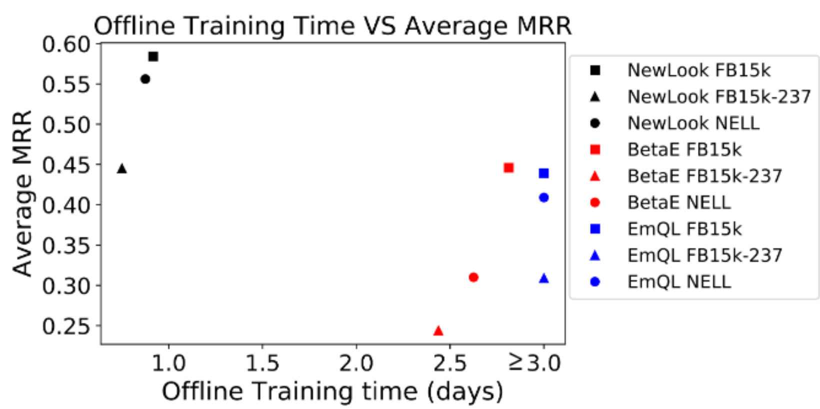
Query																																							
Query	1p			2p			3p			2i			3i			ip			pi			2u			up			2d			3d			dp			Average		
Method	GQE	Q2B	NLk	GQE	Q2B	NLk	GQE	Q2B	NLk	GQE	Q2B	NLk	GQE	Q2B	NLk	GQE	Q2B	NLk	GQE	Q2B	NLk	GQE	Q2B	NLk	GQE	Q2B	NLk	GQE	Q2B	NLk	GQE	Q2B	NLk	GQE	Q2B	NLk	GQE	Q2B	NLk
FB15k																																							
Hits@1	0.31	0.48	**0.81**	0.15	0.23	**0.51**	0.11	0.15	**0.39**	0.20	0.32	**0.51**	0.27	0.41	**0.69**	0.07	0.10	**0.21**	0.12	0.19	**0.48**	0.14	0.25	**0.81**	0.11	0.17	**0.25**	0.31	0.49	**0.88**	0.12	0.15	**0.38**	0.17	0.23	**0.34**	0.17	0.27	**0.52**
Hits@3	0.69	0.78	**0.88**	0.32	0.38	**0.64**	0.22	0.26	**0.51**	0.44	0.58	**0.72**	0.55	0.69	**0.78**	0.13	0.18	**0.31**	0.27	0.37	**0.61**	0.43	0.59	**0.94**	0.24	0.29	**0.37**	0.66	0.77	**0.95**	0.36	0.32	**0.54**	0.36	0.33	**0.46**	0.39	0.47	**0.64**
Hits@10	0.85	0.90	**0.93**	0.48	0.54	**0.75**	0.35	0.40	**0.64**	0.63	0.75	**0.80**	0.74	0.84	**0.89**	0.24	0.30	**0.43**	0.44	0.54	**0.74**	0.68	0.81	**0.98**	0.40	0.46	**0.51**	0.83	0.90	**0.97**	0.44	0.46	**0.69**	0.52	0.46	**0.59**	0.56	0.62	**0.74**
FB15k-237																																							
Hits@1	0.20	0.27	**0.68**	0.10	0.13	**0.30**	0.07	0.09	**0.19**	0.10	0.14	**0.47**	0.16	0.22	**0.57**	0.04	0.05	**0.08**	0.06	0.09	**0.28**	0.05	0.07	**0.49**	0.06	0.09	**0.15**	0.29	0.40	**0.76**	0.16	0.26	**0.29**	0.15	0.16	**0.27**	0.12	0.16	**0.37**
Hits@3	0.39	0.45	**0.85**	0.19	0.22	**0.43**	0.13	0.16	**0.31**	0.25	0.31	**0.71**	0.36	0.43	**0.71**	0.07	0.10	**0.16**	0.15	0.19	**0.41**	0.15	0.22	**0.70**	0.14	0.17	**0.24**	0.54	0.64	**0.86**	0.38	0.44	**0.46**	0.28	0.26	**0.39**	0.25	0.29	**0.52**
Hits@10	0.57	0.63	**0.94**	0.32	0.36	**0.59**	0.24	0.28	**0.45**	0.43	0.50	**0.81**	0.54	0.61	**0.81**	0.15	0.19	**0.25**	0.27	0.32	**0.54**	0.33	0.43	**0.86**	0.27	0.31	**0.37**	0.72	0.79	**0.94**	0.57	0.62	**0.64**	0.43	0.39	**0.53**	0.39	0.45	**0.64**
NELL																																							
Hits@1	0.13	0.20	**0.81**	0.08	0.11	**0.45**	0.08	0.10	**0.33**	0.09	0.14	**0.60**	0.14	0.24	**0.66**	0.04	0.05	**0.13**	0.08	0.10	**0.33**	0.03	0.07	**0.68**	0.04	0.06	**0.23**	0.20	0.29	**0.88**	0.16	0.28	**0.36**	0.19	0.19	**0.46**	0.11	0.14	**0.49**
Hits@3	0.44	0.53	**0.92**	0.20	0.23	**0.34**	0.19	0.21	**0.48**	0.27	0.33	**0.74**	0.36	0.45	**0.80**	0.08	0.11	**0.21**	0.17	0.19	**0.46**	0.22	0.33	**0.85**	0.12	0.13	**0.35**	0.55	0.69	**0.95**	0.41	0.43	**0.54**	0.38	0.33	**0.60**	0.28	0.33	**0.60**
Hits@10	0.62	0.70	**0.96**	0.35	0.39	**0.48**	0.30	0.34	**0.63**	0.47	0.55	**0.84**	0.58	0.66	**0.89**	0.17	0.20	**0.32**	0.28	0.32	**0.59**	0.44	0.56	**0.93**	0.27	0.29	**0.47**	0.72	0.82	**0.98**	0.63	0.64	**0.71**	0.54	0.49	**0.71**	0.45	0.49	**0.71**

Answering queries with a single target variable node. NLK refers to NewLook, Q2B refers to Query2Box.

Queries with A Single Target Variable Node

- Dataset: common dataset used by EmQL and BetaE
- Queryset:
  - 5 training query structures
  - 9 testing query structures





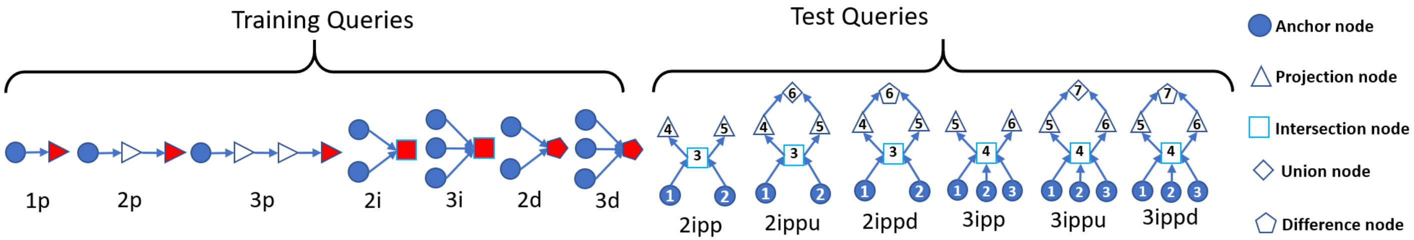
Average MRR results on the Query2Box datasets.

Dataset	1p	2p	3p	2i	3i	ip	pi	2u	up	Average	Training Time
FB15k											
BetaE	65.0	42.1	37.8	52.9	64.0	41.5	22.9	48.8	26.9	44.6	2.81 days
EmQL	36.8	45.2	40.9	57.4	60.9	**55.6**	53.8	7.4	**37.5**	43.9	≥ 4 days
NewLook	**84.1**	**56.6**	**45.1**	**60.1**	**77.9**	28.7	**56.9**	**82.5**	34.0	58.4	0.91 days
FB15k-237											
BetaE	39.1	24.2	20.4	28.1	39.2	19.4	10.6	22.0	17.0	24.4	2.33 days
EmQL	33.4	30.5	**30.4**	37.8	43.6	**35.1**	35.8	7.5	**24.1**	30.9	≥ 4 days
NewLook	**77.8**	**39.6**	28.1	**55.2**	**64.5**	14.1	**36.0**	**61.6**	23.4	44.5	0.75 days
NELL											
BetaE	53.0	27.5	28.1	32.9	45.1	21.8	10.4	38.6	19.6	30.7	2.62 days
EmQL	37.2	35.1	34.9	53.9	65.4	**44.1**	**56.1**	10.5	31.1	40.9	≥ 4 days
NewLook	**87.5**	**54.6**	**43.4**	**68.9**	**74.8**	19.7	42.2	**77.7**	**31.4**	55.6	0.87 days

Average MRR results on the Query2Box datasets.

Queries with Multi-variable Nodes

- Query set:
  - 7 training query structures
  - 6 testing query structures

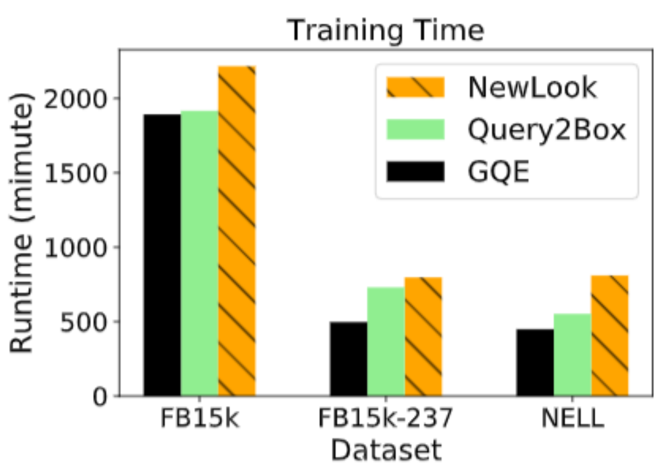


Method	GQE	Q2B	GRay	FilM	GFinder	NewLook
2ipp	0.550	0.481	0.554	0.566	0.638	**0.720**
2ippu	0.592	0.426	0.505	0.583	0.631	**0.761**
2ippd	0.462	0.435	0.452	0.637	**0.652**	0.641
3ipp	0.447	0.442	0.513	0.394	0.437	**0.688**
3ippu	0.507	0.417	0.456	0.408	0.465	**0.733**
3ippd	0.447	0.395	0.421	0.423	0.482	**0.634**
Average	0.500	0.432	0.483	0.501	0.550	**0.696**

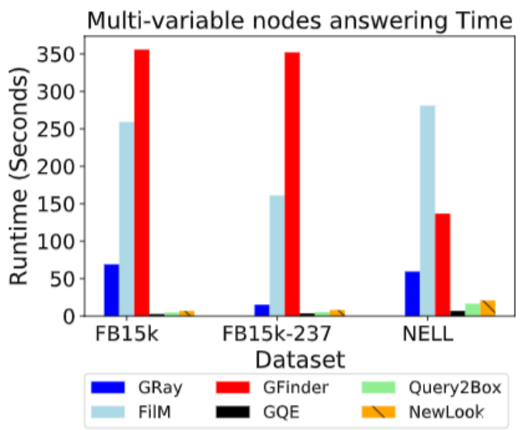
Answering queries with multi-variable nodes.

Runtime

- Offline training time
  - NewLook is a little bit slower than GQE and Query2Box
- Online query time
  - Gfinder has the longest online query time
  - GQE has the shortest online query time



Training Time



Testing Time

Ablation Study: Projection Operation

- Neural network based projection has a better performance
- Can efficiently mitigate cascading errors in multi-hop queries

Query	1p		2p		3p		2i		3i	
Method	LT	NN	LT	NN	LT	NN	LT	NN	LT	NN
FB15k										
Hits@3	**0.71**	0.69	0.37	**0.41**	0.27	**0.38**	0.59	**0.66**	0.71	**0.78**
Hits@10	**0.86**	0.84	0.53	**0.57**	0.42	**0.53**	0.77	**0.80**	0.86	**0.89**
FB15k-237										
Hits@3	**0.43**	**0.43**	0.22	**0.26**	0.17	**0.23**	0.31	**0.34**	0.43	**0.47**
Hits@10	**0.60**	0.59	0.36	**0.39**	0.29	**0.35**	0.49	**0.51**	0.61	**0.63**
NELL										
Hits@3	0.53	**0.62**	0.25	**0.33**	0.22	**0.33**	0.28	**0.34**	0.46	**0.53**
Hits@10	0.70	**0.73**	0.39	**0.47**	0.35	**0.45**	0.48	**0.49**	0.66	**0.68**

Ablation study of projection operation. LT refers to linear transformation based model. NN refers to the proposed neural network based model.

Ablation Study: Difference Operation

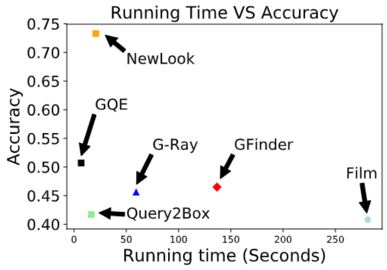
- Attention neural network based model has a much better performance than Deepsets based model.
- The generalization ability of the Deepsets model for modeling the difference operation is very limited.

Query	2d		3d		dp	
Method	Deepsets	Attention	Deepsets	Attention	Deepsets	Attention
FB15k						
Hits@3	0.67	**0.72**	0.52	**0.58**	0.00	**0.26**
Hits@10	0.82	**0.86**	0.67	**0.73**	0.00	**0.40**
FB15k-237						
Hits@3	0.56	**0.59**	0.43	**0.48**	0.00	**0.21**
Hits@10	0.73	**0.75**	0.60	**0.66**	0.00	**0.35**
NELL						
Hits@3	0.74	**0.75**	0.53	**0.59**	0.00	**0.29**
Hits@10	0.83	**0.85**	0.69	**0.74**	0.00	**0.43**

Ablation study of difference operation.

Conclusion

- **Contribution**: NewLook for answering logical queries on KG
- **Key Ideas**
  - Embed entities in KG as points, nodes in query graph as box
  - Model each operation as a neural network



- **Results**:
  - Broader applicability: Support 4 operations and answer queries with multiple variable nodes
  - Consistent performance improvement: high accuracy
  - Computational efficiency: fast online query time and offline training time