Mask and Reason: Pre-Training Knowledge Graph Transformers for Complex Logical Queries

(kgTransformer)

Xiao Liu*, Shiyu Zhao*, Kai Su*, Yukuo Cen, Jiezhong Qiu,
Mengdi Zhang, Wei Wu, Yuxiao Dong, Jie Tang

GitHub Repo

Incompleteness of Knowledge Graphs (KGs)

- Real-world KGs are far from complete
  - Fully-supervised (Human curated)
    - Freebase
    - Wikidata
  - Semi-supervised (Human-in-the-loop)
    - NELL
    - Knowledge Vault
- Missing edges in querying

Freebase

WIKIDATA

Wikidata

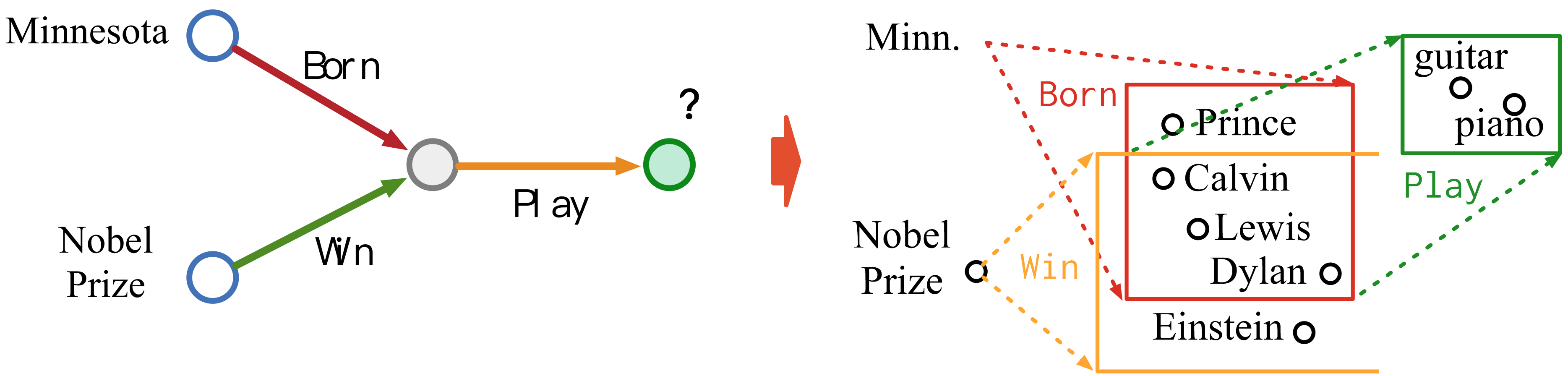
NELL

Knowledge Vault

Complex Logical Queries on KGs

- Existential Positive First-Order Logic (EPFO)

Query: What musical instruments can Minnesota-born Nobel Prize winners play?



- Hard to query
  - Queries involve complicated structures
  - Queries can contain missing edges

Existing Architecture: KG Embeddings

- Nonparameterized logical operators
  - Unlearnable and low-capacity

Query2Box (Ren et al., 2020)

Intersection: $\mathbf{p}_{\text{inter}} = \boxed{(\mathrm{Cen}(\mathbf{p}_{\text{inter}}), \mathrm{Off}(\mathbf{p}_{\text{inter}}))}$

Operator: *overlapped area* in embedding space

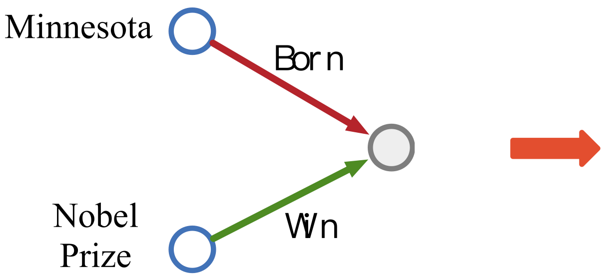
Continuous Query Decomposition (Arakelyan et al., 2021)

Intersection: $\boxed{x \cdot y\,; \min\{x,\ y\}\,; \max\{0,\ x + y - 1\}}$

Operator: logical T-norm over probabilities of neural link predictor (e.g., Complex)

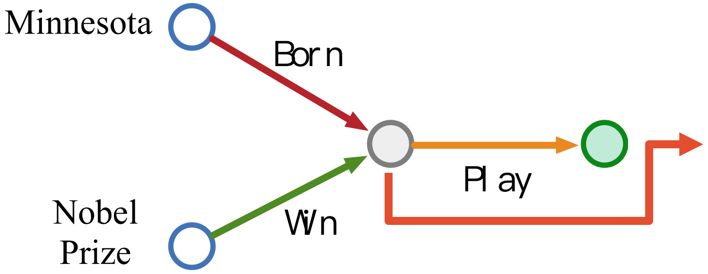
- Left-to-right reasoning
  - Loss of wider context

Case 1: Minnesota-born Nobel Prize winners



Answer probability:
1. Calvin 0.33
2. Lewis 0.33
3. **Dylan 0.33**

Case 2: Minnesota-born Nobel Prize winners that may like to play certain musical instruments

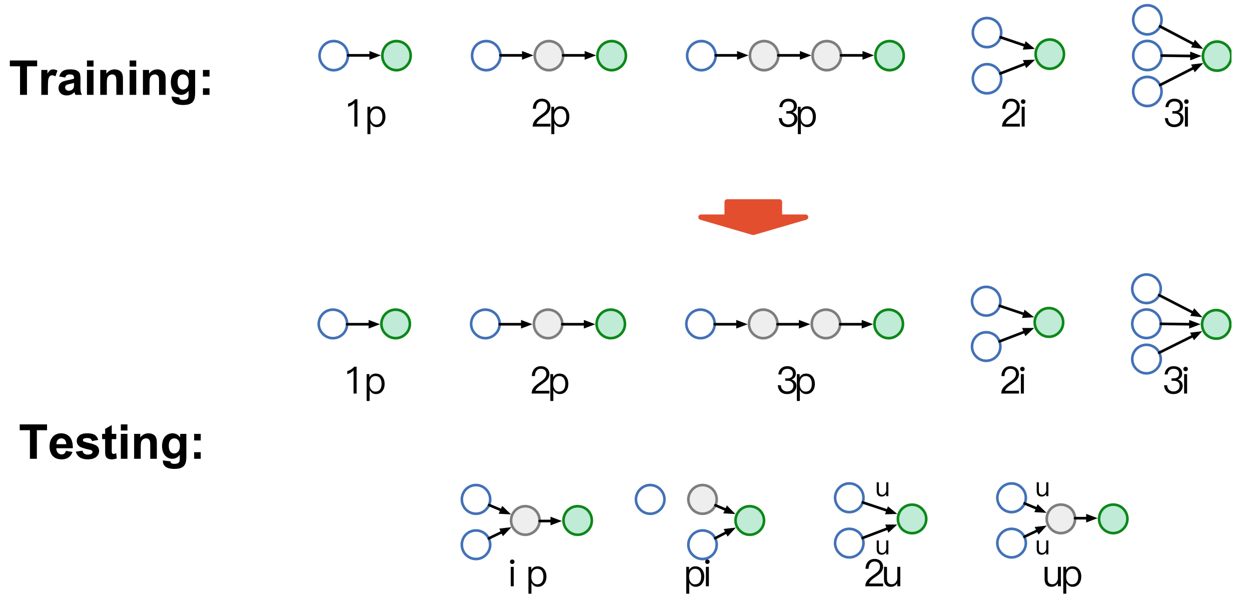


Answer probability:
1. **Dylan 0.6**
2. Lewis 0.2
3. Calvin 0.2

Problem 1: KG embeddings are low-capacity architectures and only reason from left to right

Existing Training Strategies

- Training: supervised learning
  - Training on 5 basic query types
  - But to test on both seen and unseen types (e.g., ip, pi, 2u, up)



Problem 2: supervised learning doesn't generalize to new types well

Challenges

- Model architecture
  - –Exponential complexity
    - Multi-hop query
    - Logical operators
  - –All-direction information

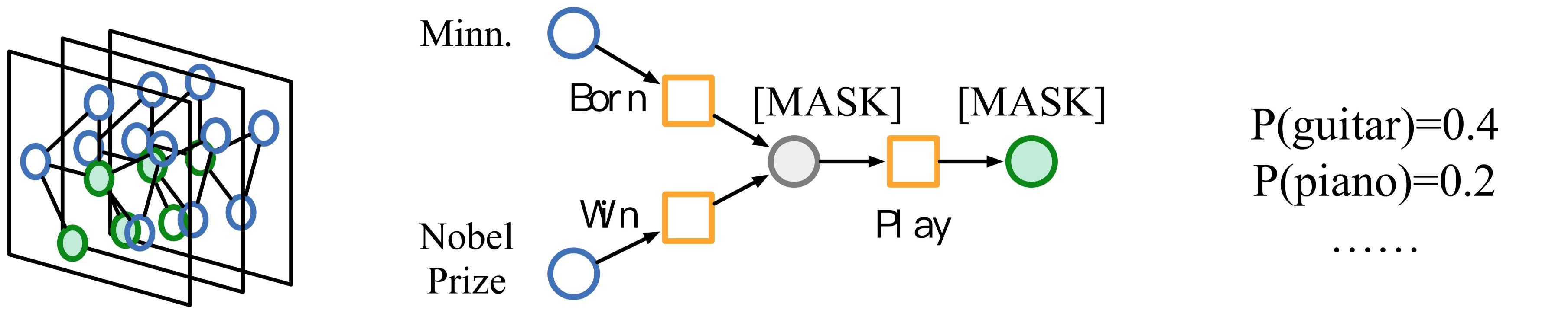
Advanced and Large-capacity

- Training strategies
  - –Transfer and generalize
    - Supervised training is not enough

Strategies to Generalize

kgTransformer

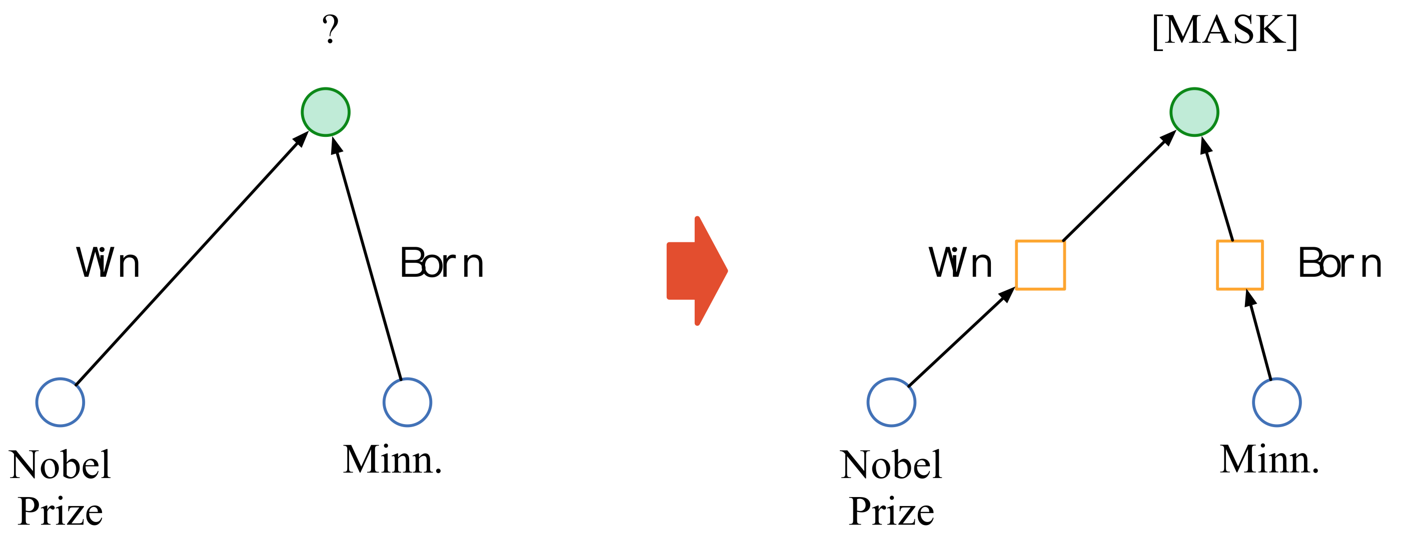
- Pre-training Transformer on KGs
  - Architecture: kgTransformer
    - A more advanced architecture that can encode query graphs
  - Training strategy: Masked Pre-training & Fine-tuning
    - Pre-training can encourage generalization



Pre-trained kgTransformer Masked Prediction Decode Probability

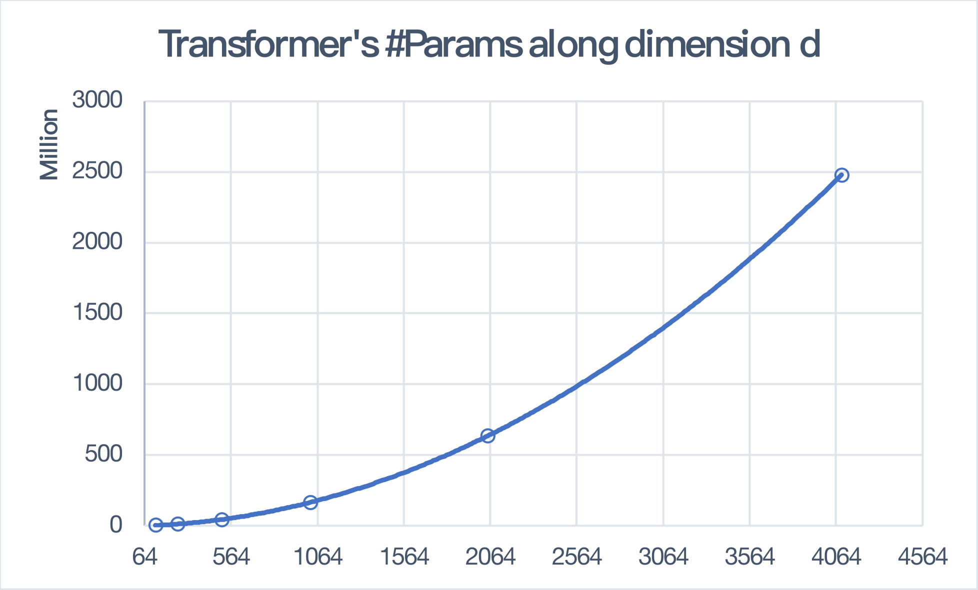
Architecture: kgTransformer

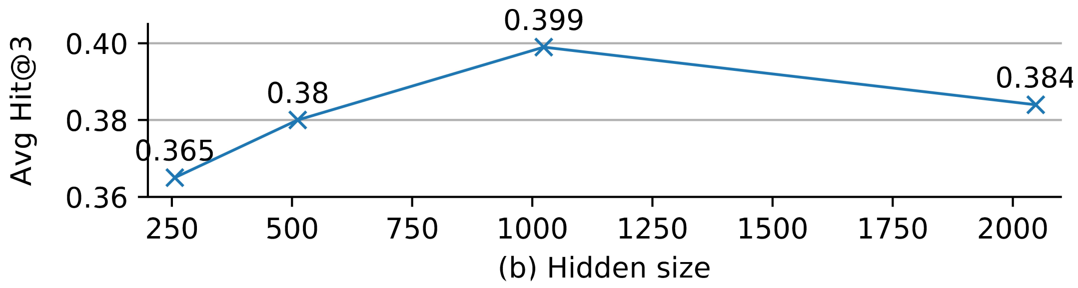
- Triple Transform
  - Represent a relation edge as a relation node
  - Unknown entities are replaced by [MASK]



Architecture: kgTransformer

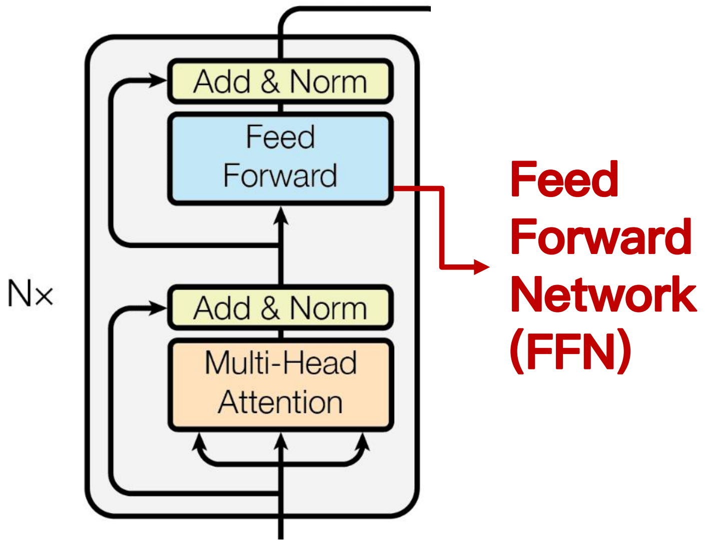
- Model capacity is crucial
- Increase hidden dimension *d*:
  - Quadratic space complexity $O(d^2)$
  - Performance saturation
- Need other solutions to scale up model capacity



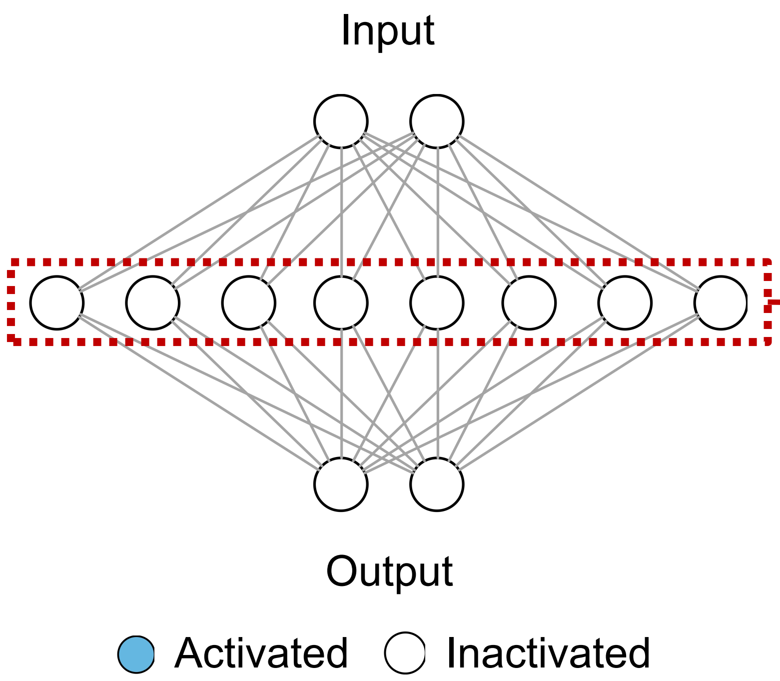


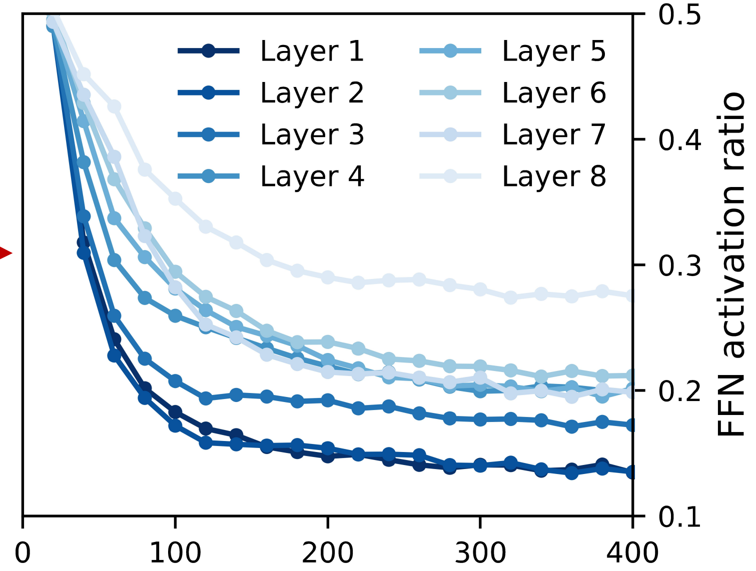
Architecture: kgTransformer

- Observation: sparsity in kgTransformer
  - Feed-forward Network (FFN) is sparsely activated (10~20%)



A Transformer layer
(Adapted from Vaswani et al., 2017)

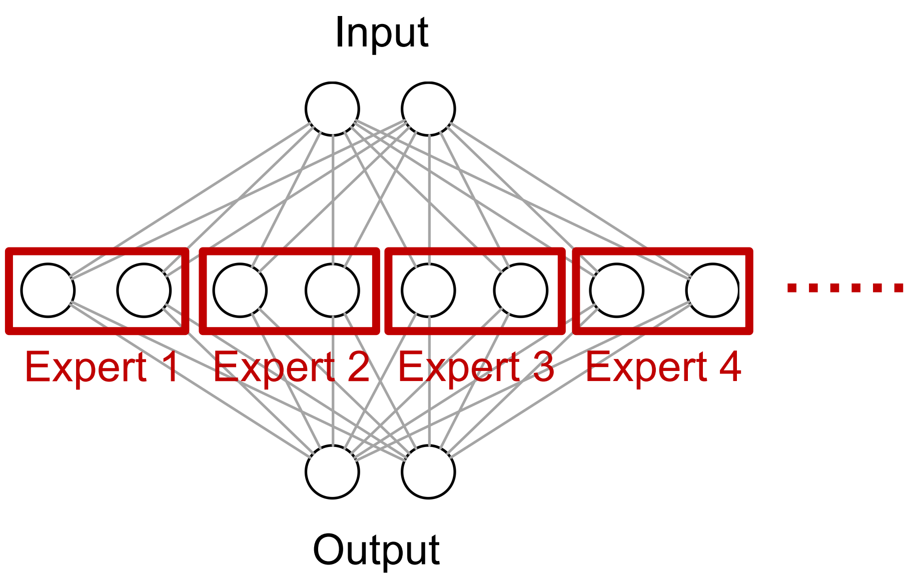


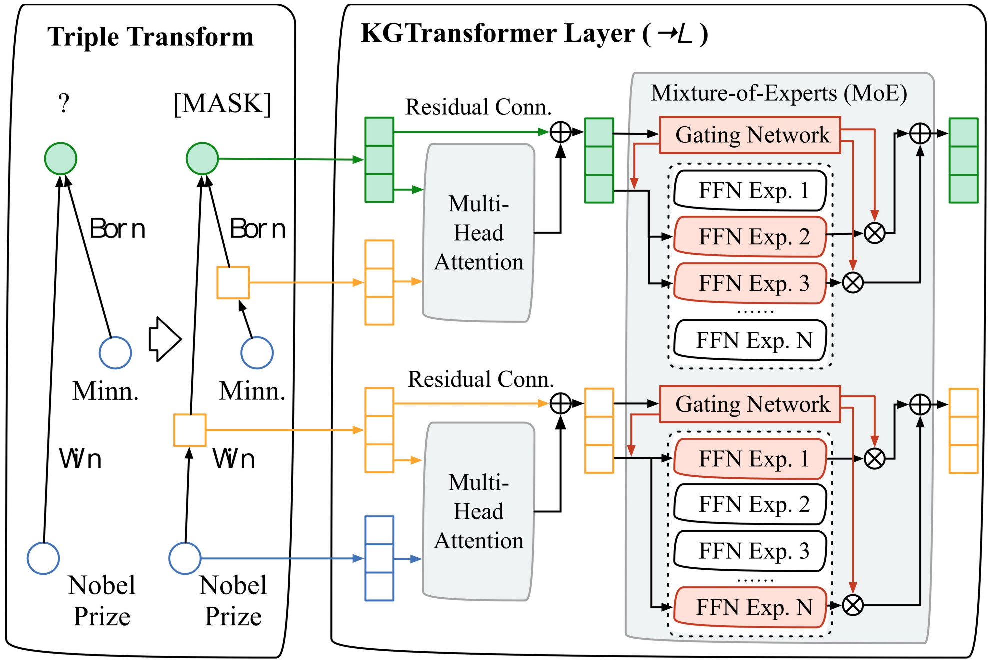


1. Vaswani et al., Attention is all you need, 2017

Architecture

- Mixture-of-Experts (MoE): scaling model-capacity via sparsity
  - Split FNN into experts
  - Only involve experts predicted to be activated (by Gating Network)
  - Increase number of experts

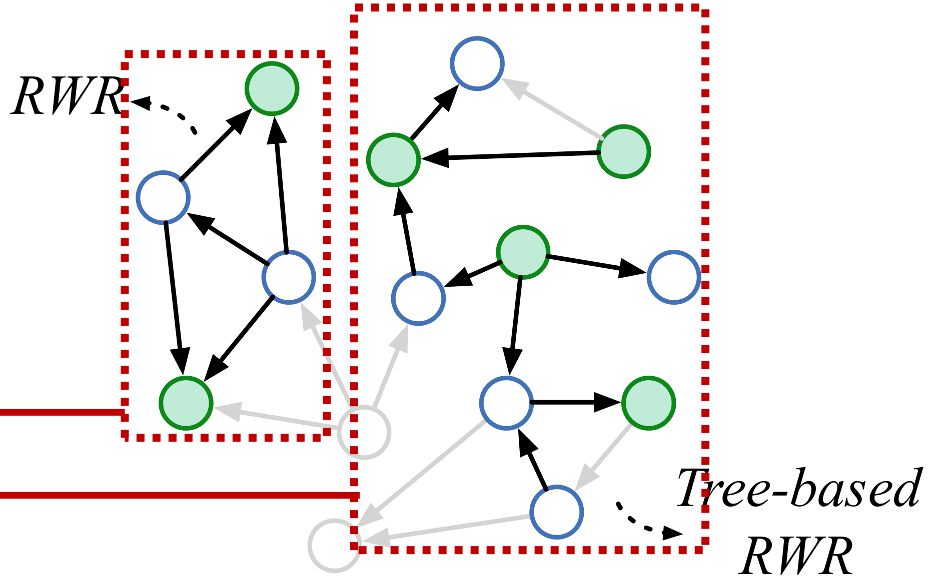




(a) Architecture: KGTransformer & Mixture-of-Experts

Training: Masked Pre-training & Fine-tuning

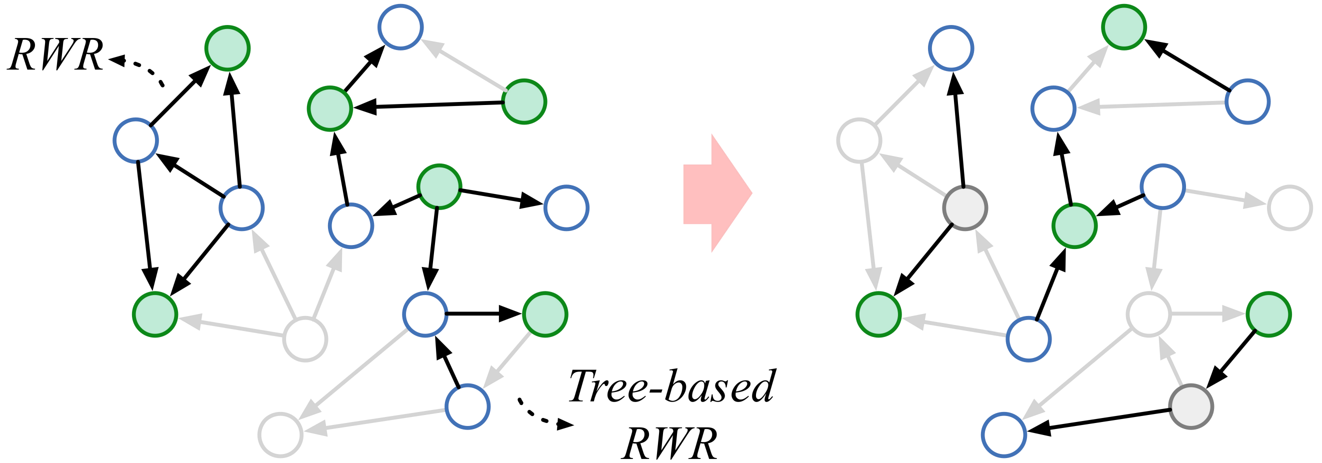
- Mask pre-training on random sampled queries
  - Two-stage Pre-training
  - Stage 1: Initialization
    - Dense and large subgraphs
      - 8 to 16 entities per query
    - Random Walk with Restart (RWR)
      - Vanilla (might contain rings)
      - Tree-based
    - Learn arbitrary-shaped queries
      - Encourage generalization



Stage 1: Initialization (RWR)

Training: Masked Pre-training & Fine-tuning

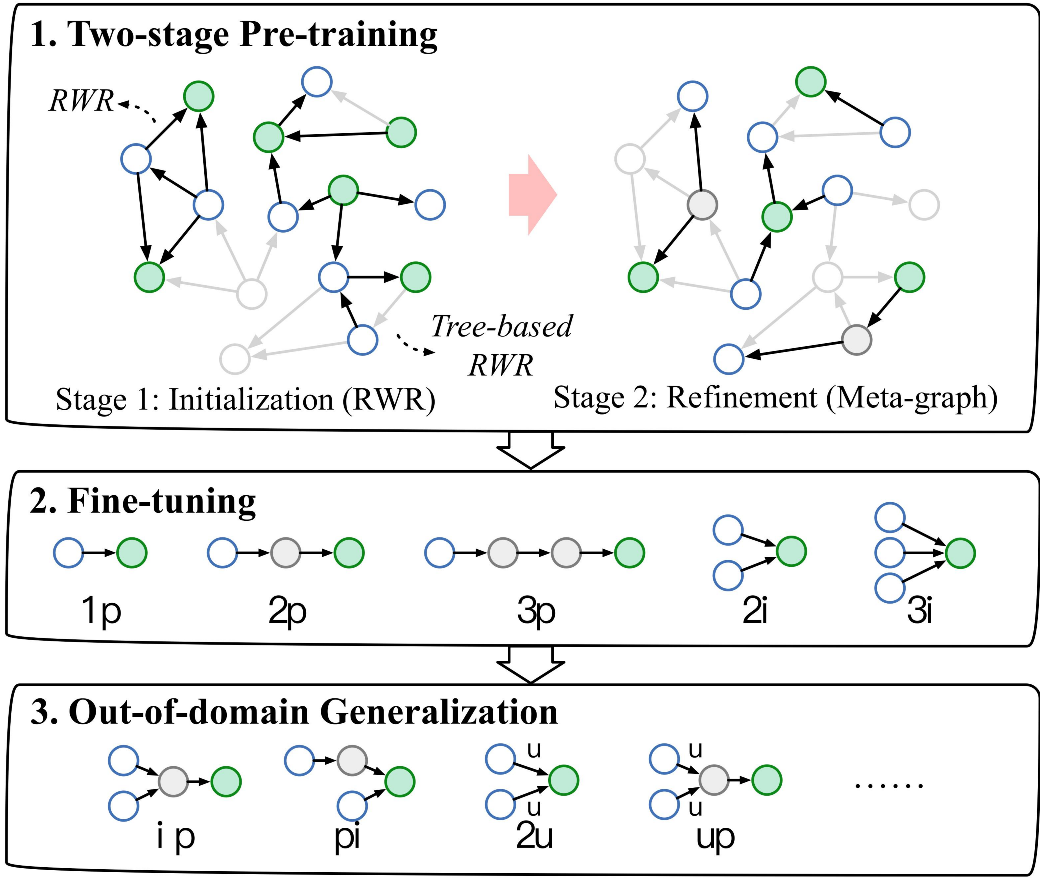
- Mask pre-training on randomly sampled queries
  - Two-stage Pre-training
  - Stage 2: Refinement
    - Sparse and small graphs
    - 5 basic query type
    - Similar to test setting



Stage 1: Initialization (RWR)   Stage 2: Refinement (Meta-graph)

Training: Masked Pre-training & Fine-tuning

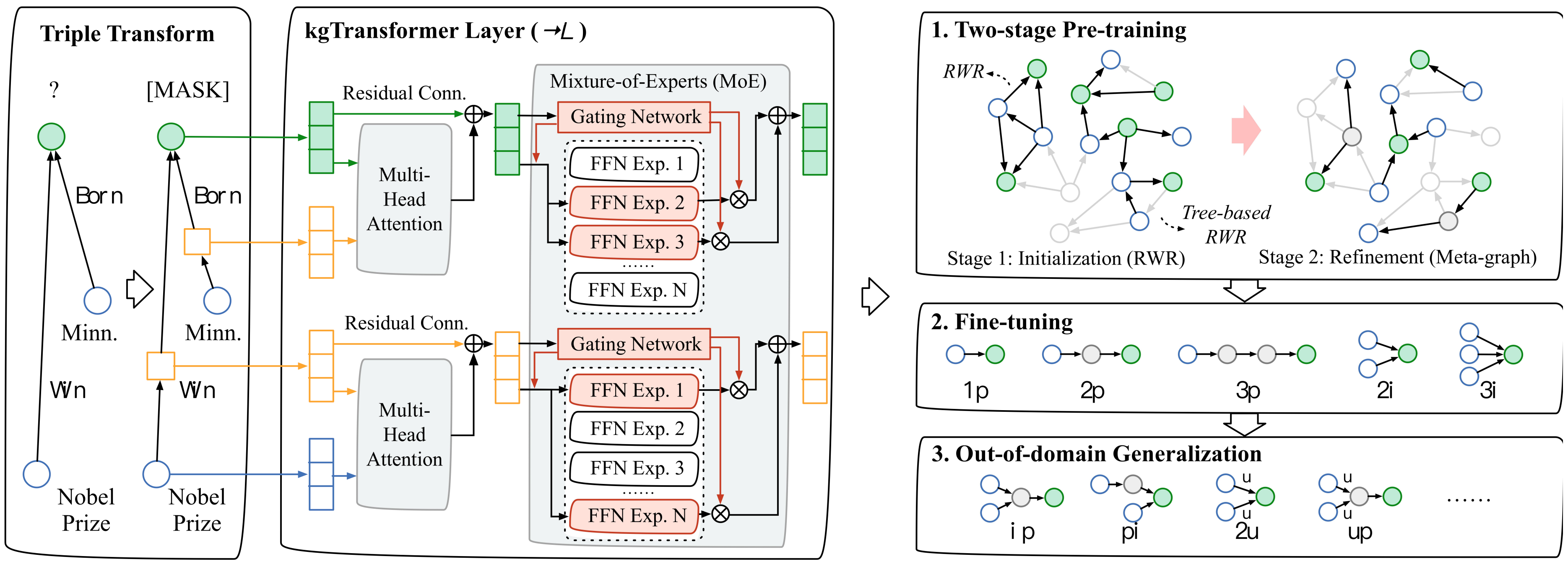
- Mask Fine-tuning
  - Fine-tuning
    - Training over preprocessed datasets of 5 basic query types
  - Out-of-domain Generalization
    - Combining knowledge from pre-training and fine-tuning



(b) Masked Pre-training & Fine-tuning

Summary: kgTransformer

- Pre-training Transformer on KGs
  - Architecture: kgTransformer with Mixture-of-Experts
  - Training strategy: Masked Pre-training & Fine-tuning



(a) Architecture: kgTransformer & Mixture-of-Experts

(b) Masked Pre-training & Fine-tuning

Experimental Results

- 9 reasoning tasks
  - 5 in-domain
  - 4 out-of-domain
- Improvements (Relative)
  - NELL995: **+6.1%**
  - FB15k-237： **+15.9%**

Table 1: Main Hits@3m for complex query reasoning on FB15k-237 and NELL995 benchmarks (**bold** denotes the best results; underline denotes the second best results)

Dataset	Model	Avg	Avg w/o u	In-domain 1p	2p	3p	2i	3i	Out-of-domain ip	pi	2u	up
NELL995	GQE [11]	0.248	0.270	0.417	0.231	0.203	0.318	0.454	0.081	0.188	0.200	0.139
	Q2B [26]	0.306	0.317	0.555	0.266	0.233	0.343	0.480	0.132	0.212	0.369	0.163
	EmQL [31][1]	0.277	0.294	0.456	0.231	0.172	0.331	0.483	0.143	0.244	0.226	0.207
	BiQE [19]	-	0.344	0.587	0.305	0.326	0.371	0.531	0.103	0.187	-	-
	CQD(CO) [1]	0.368	0.370	**0.667**	0.265	0.220	**0.410**	0.529	0.196	**0.302**	**0.531**	0.194
	CQD(Beam) [1]	0.375	0.385	**0.667**	0.350	0.288	**0.410**	0.529	0.171	0.277	**0.531**	0.156
	KGTransformer	**0.398**	**0.406**	0.625	**0.401**	**0.367**	0.405	**0.546**	**0.203**	0.300	0.469	**0.270**
FB15k-237	GQE [11]	0.230	0.250	0.405	0.213	0.153	0.298	0.411	0.085	0.182	0.167	0.160
	Q2B [26]	0.268	0.283	0.467	0.240	0.186	0.324	0.453	0.108	0.205	0.239	0.193
	EmQL [31][1]	0.219	0.241	0.389	0.201	0.154	0.275	0.386	0.101	0.184	0.115	0.165
	BiQE [19]	-	0.293	0.439	0.281	0.239	0.333	0.474	0.110	0.177	-	-
	CQD(CO) [1]	0.272	0.290	0.512	0.213	0.131	0.352	0.457	0.146	0.222	0.281	0.132
	CQD(Beam) [1]	0.290	0.315	**0.512**	0.288	0.221	0.352	0.457	0.129	0.249	0.284	0.121
	KGTransformer	**0.336**	**0.357**	0.479	**0.323**	**0.277**	**0.398**	**0.539**	**0.190**	**0.294**	**0.295**	**0.225**

[1] EmQL's reported results are not under the standard metric. We have verified the mismatch with its authors and re-evaluated the performance.

Ablation Study

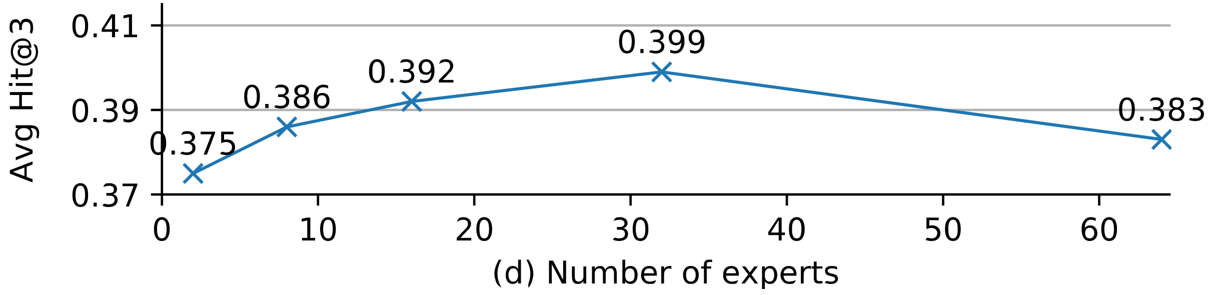
- How much does pre-training contribute?
  - 26.2 -> 33.6 (FB15k-237);
  - 28.8 -> 39.5 (NELL995)

Table 3: Ablation on certain pre-training & fine-tuning strategies adopted (Hits@3m).

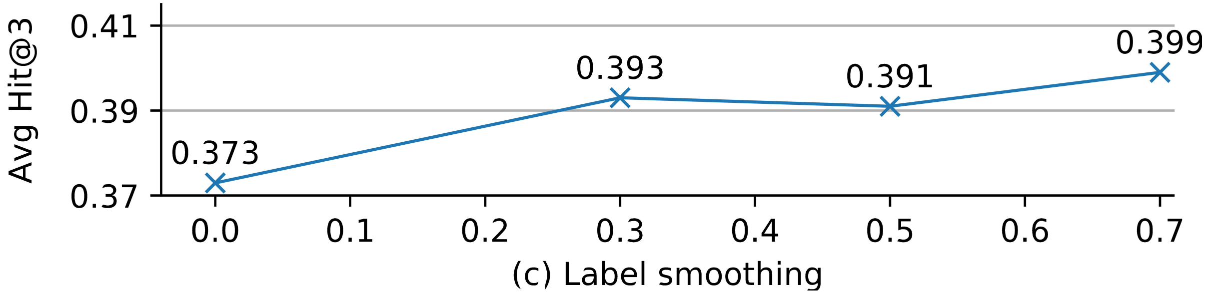
	FB15k-237	NELL995
KGTransformer (Stage 1 + Stage 2)	**0.336**	0.395
-only Stage 1 in pre-training	0.308	0.307
-only Stage 2 in pre-training	0.307	**0.398**
-w/o fine-tuning	0.301	0.368
-w/o pre-training	0.262	0.288

Ablation Study

- Number of Expert
  - Helpful when growing from 2 (vanilla) to 32



- Label smoothing
  - A very useful technique for random sampled queries in pre-training



Ablation Study

- MoE Efficiency
  - Expanding model capacity x16 times
  - Costs in time: +11.6%～38.7%

Table 4: Training and inference time per step (ms) along with different number of experts using per step batch size 64.

	Number of experts			
	2	8	16	32
Pre-training stage 1	41.46	42.02	45.38	49.30 (+18.9%)
Pre-training stage 2	17.37	18.83	19.78	24.09 (+38.7%)
Fine-tuning	32.41	32.91	36.46	37.47 (+15.6%)
Inference	12.86	13.36	13.85	14.35 (+11.6%)

x16 times

Conclusion

- Method
  - Architecture: kgTransformer with MoE scaling up
  - Training: Masked pre-training and fine-tuning
- Results
  - Better performance on FB15k-237 and NELL995
  - Better generalizability and interpretability

- Code: https://github.com/THUDM/kgTransformer