A Prompt-Based Knowledge Graph Foundation Model for Universal In-Context Reasoning

Yuanning Cui[1], Zequn Sun[1], Wei Hu[1, 2]

[1] State Key Laboratory for Novel Software Technology, Nanjing University, China

[2] National Institute of Healthcare Data Science, Nanjing University, China

NEURAL INFORMATION PROCESSING SYSTEMS

NeurIPS 2024

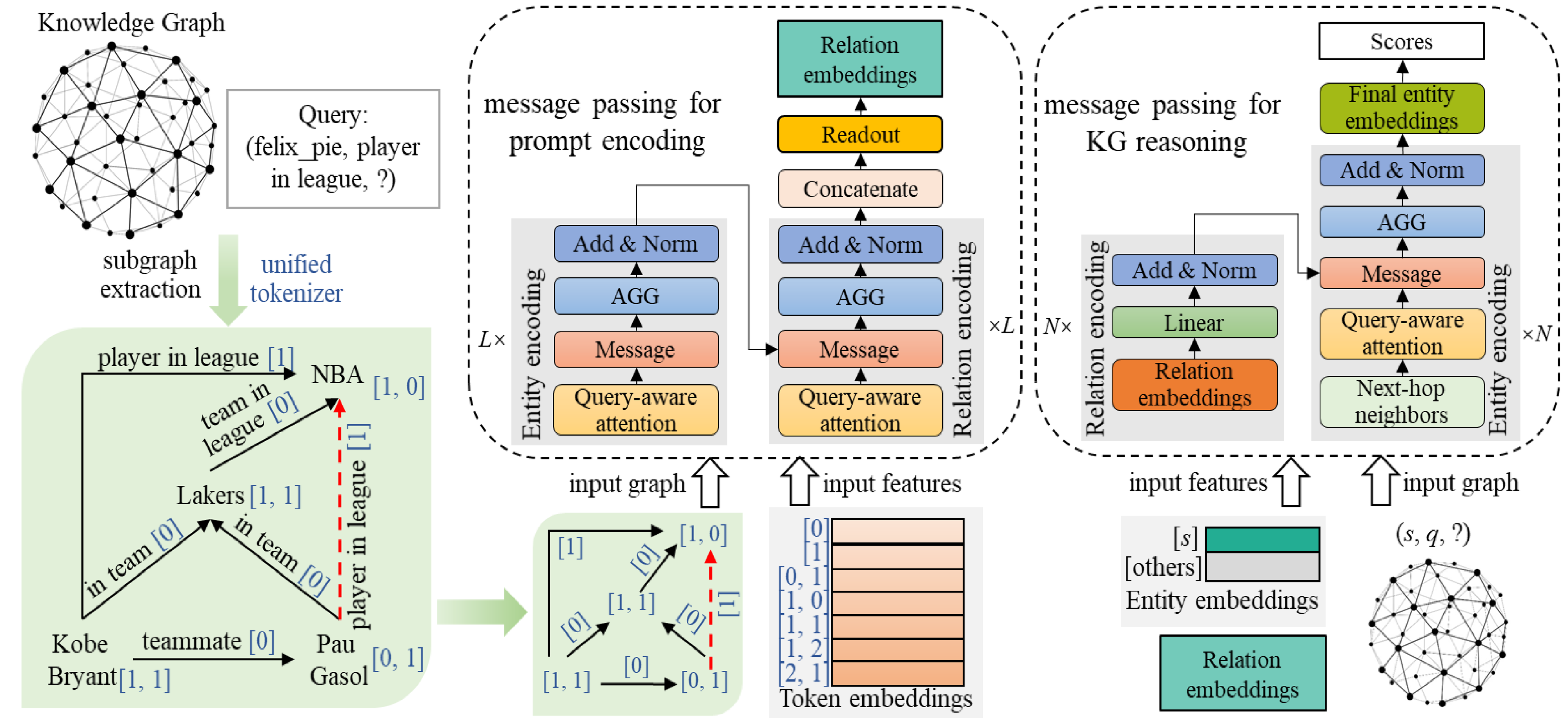
Introduction

- **Why do we need a knowledge graph (KG) foundation model?**
  - In the fields of NLP and CV, there have been many foundation models that can be used for diverse corpora.
  - However, existing KG reasoning methods have to develop, train and store separate models for different KGs.
- **Is it possible to train a model that works across all KGs?**
  - Yes! We propose a KG foundation model to achieve this goal.
- **In this paper, we propose KG-ICL, a KG foundation model.**
  - First, we construct prompt graphs as contexts.
  - Then, we encode prompt graphs to obtain prompt embeddings.
  - Finally, we encode the KG and score candidate entities.
- **What can KG-ICL do?**
  - Reason on any KG without fine-tuning
  - Apply to any static or dynamic KGs (transductive & inductive)
  - Outperforms Supervised-SOTA baselines on 43 datasets

Model

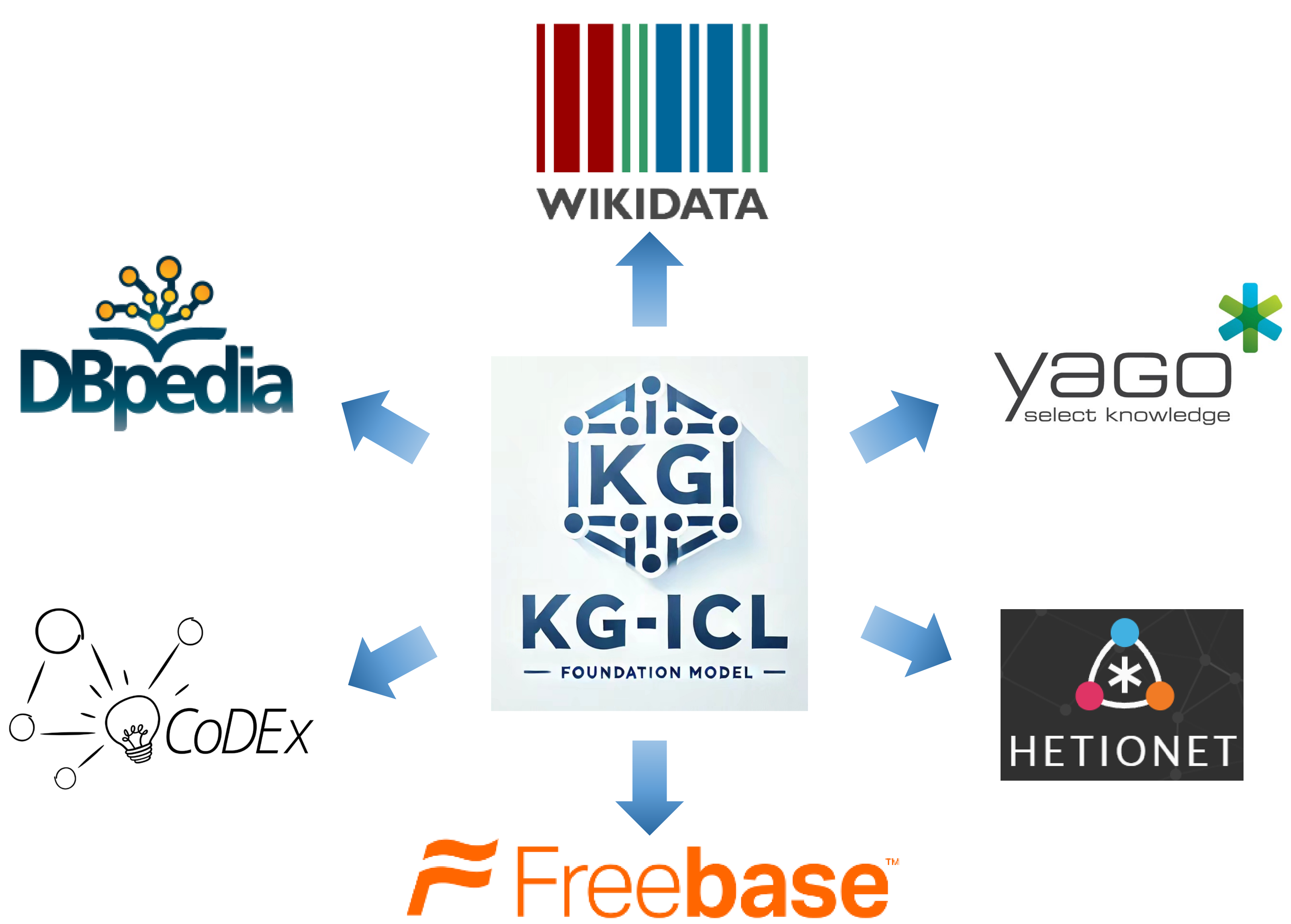
We propose a KG foundation model.

- **Prompt graph generation**
  - Given a query and its corresponding KG, we extract prompt graphs as context for the query relation "player in league". The entities and relations in the prompt graphs are mapped to the unified tokens
- **Prompt graph encoding**
  - We employ a graph neural network to encode the prompt graph and extract the relation representations as the prompts.
- **Knowledge graph reasoning**
  - Then we use the prompts to initialize the representations of entities and relations in the KG. After KG encoding, we score the candidate entities based on their embeddings in the final layer.



(A) Prompt graph generation (B) Prompt graph encoding (C) Knowledge graph reasoning

KG-ICL: One Model for All Knowledge Graphs!



Experiments

- **Pre-training**
  - 3 source datasets (FB V1, NELL V1 and CoDEx-small).
- **Evaluation**
  - 16 transductive KGs, 14 inductive KGs and 13 fully-inductive KGs.
- **Results on transductive datasets**

Datasets	Supervised SOTA		ULTRA pre-train		KG-ICL pre-train		ULTRA finetune		KG-ICL finetune	
	MRR	H@10	MRR	H@10	MRR	H@10	MRR	H@10	MRR	H@10
AristoV4	0.311	0.447	0.182	0.282	0.203	0.306	**0.343**	**0.496**	0.313	0.480
CoDEx-small	0.473	0.663	0.472	0.667	0.465	0.654	**0.490**	**0.686**	0.479	0.662
CoDEx-medium	0.352	0.490	0.372	0.525	0.330	0.474	0.372	0.525	**0.402**	**0.565**
CoDEx-large	0.345	0.473	0.338	0.469	0.261	0.376	0.343	0.478	**0.388**	**0.508**
ConceptNet100K	0.320	0.553	0.082	0.162	0.249	0.416	0.310	0.529	**0.371**	**0.584**
DBpedia100K	0.306	0.418	0.398	0.576	0.390	0.541	0.436	0.603	**0.455**	**0.604**
FB15k-237	**0.415**	**0.599**	0.368	0.564	0.359	0.541	0.368	0.564	0.376	0.538
FB15k-237-10	0.219	0.337	0.248	0.398	**0.274**	**0.433**	0.254	0.411	0.260	0.416
FB15k-237-20	0.247	0.391	0.272	0.436	**0.285**	0.454	0.274	0.445	0.284	**0.456**
FB15k-237-50	0.293	0.458	0.324	0.526	**0.329**	0.520	0.325	**0.528**	0.324	0.499
Hetionet	0.257	0.403	0.257	0.379	0.260	0.371	**0.399**	**0.538**	0.269	0.402
NELL-995	**0.543**	0.651	0.406	0.543	0.532	0.653	0.509	0.660	0.534	**0.672**
NELL23K	0.253	0.419	0.239	0.408	0.317	0.532	0.268	0.450	**0.329**	**0.552**
WD-singer	0.393	0.500	0.382	0.498	0.470	0.582	0.417	0.526	**0.493**	**0.599**
WN18RR	**0.551**	**0.666**	0.480	0.614	0.455	0.527	0.480	0.614	0.536	0.637
YAGO3-10	**0.563**	0.708	0.451	0.615	0.352	0.503	0.557	**0.710**	0.545	0.688
Average	0.351	0.493	0.396	0.557	0.442	0.606	0.421	0.590	**0.473**	**0.638**

- **Results on inductive and fully-inductive datasets**

Datasets	Supervised SOTA		ULTRA pre-train		KG-ICL pre-train		ULTRA finetune		KG-ICL finetune	
	MRR	H@10	MRR	H@10	MRR	H@10	MRR	H@10	MRR	H@10
FB V1	0.457	0.589	0.498	0.656	0.520	0.678	0.509	0.670	**0.531**	**0.700**
FB V2	0.510	0.672	0.512	0.700	0.565	0.749	0.524	0.710	**0.568**	**0.768**
FB V3	0.476	0.637	0.491	0.654	0.535	0.695	0.504	0.663	**0.537**	**0.704**
FB V4	0.466	0.645	0.486	0.677	0.513	0.699	0.496	0.684	**0.525**	**0.706**
ILPC-large	0.070	0.146	0.290	0.424	0.288	0.412	**0.308**	**0.431**	0.295	0.411
ILPC-small	0.130	0.251	0.302	0.443	0.288	0.446	0.303	0.453	**0.316**	**0.473**
NELL V1	0.637	0.866	0.785	0.913	0.693	0.915	0.757	0.878	**0.841**	**0.995**
NELL V2	0.419	0.601	0.526	0.707	**0.644**	**0.835**	0.575	0.761	0.641	0.835
NELL V3	0.436	0.594	0.515	0.702	0.613	0.792	0.563	0.755	**0.631**	**0.799**
NELL V4	0.363	0.556	0.479	0.712	0.590	0.791	0.469	0.733	**0.594**	**0.802**
WN V1	0.741	0.826	0.648	0.768	0.733	**0.838**	0.685	0.793	**0.762**	0.827
WN V2	0.704	**0.798**	0.663	0.765	0.696	0.783	0.679	0.779	**0.721**	0.787
WN V3	0.452	0.568	0.376	0.476	0.425	0.548	0.411	0.546	**0.503**	**0.626**
WN V4	0.661	0.743	0.611	0.705	0.652	0.722	0.614	0.720	**0.683**	**0.749**
FB-25	0.223	0.371	0.388	0.640	0.396	0.656	0.383	0.635	**0.434**	**0.694**
FB-50	0.189	0.325	0.338	0.543	0.341	0.559	0.334	0.538	**0.384**	**0.598**
FB-75	0.117	0.218	0.403	0.604	0.438	0.633	0.400	0.598	**0.458**	**0.664**
FB-100	0.133	0.271	0.449	0.642	0.487	0.694	0.444	0.643	**0.499**	**0.703**
NL-0	0.309	0.506	0.342	0.523	**0.557**	**0.777**	0.329	0.551	0.555	0.765
NL-25	0.261	0.464	0.395	0.569	**0.550**	**0.736**	0.407	0.596	0.540	0.730
NL-50	0.281	0.453	0.407	0.570	**0.534**	0.704	0.418	0.595	0.528	**0.708**
NL-75	0.334	0.501	0.368	0.547	**0.452**	0.673	0.374	0.570	0.446	**0.681**
NL-100	0.269	0.431	0.471	0.651	0.556	0.762	0.458	0.684	**0.557**	**0.766**
WK-25	0.107	0.169	0.316	0.532	0.423	0.621	0.321	0.535	**0.425**	**0.628**
WK-50	0.247	0.362	0.166	0.324	0.273	0.430	0.140	0.280	**0.277**	**0.432**
WK-75	0.068	0.135	0.365	0.537	0.437	0.602	0.380	0.530	**0.466**	**0.626**
WK-100	0.186	0.309	0.164	0.286	0.262	0.409	0.168	0.286	**0.270**	**0.415**

Any questions, please email to yncui.nju@gmail.com

Source code: https://github.com/nju-websoft/KG-ICL