Learning Representations for Reasoning: Generalizing Across Diverse Structures

Zhaocheng Zhu

09/18/2024

Introduction

Our Works

Conclusions

Introduction — Our Works — Conclusions

Why is generalization essential?

What generalization do we need?

Why do we study structures?


reasoning
perception

Are We Doing Well?

What is the answer to 1 + 1 + 1 - 1 - 1?

The answer to 1 + 1 + 1 - 1 - 1 is 1.

Are We Doing Well?

What is my son’s son’s son’s father’s father?

Your son's son is your grandson. Your grandson's father is your son. Your son's father is you. So, the answer is you.

Are We Doing Well?

How do you get 4L water using 5L and 3L jugs?

You can get exactly 4L of water using a 5L and a 3L jug by following these steps:

1. Fill the 5L jug to the top.
2. Pour water from the 5L jug into the 3L jug until the 3L jug is full. This leaves you with 2L in the 5L jug.
3. Empty the 3L jug and pour the remaining 2L from the 5L jug into the 3L jug.
4. Fill the 5L jug to the top again.
5. Pour water from the 5L jug into the 3L jug until the 3L jug is full. Since the 3L jug already has 2L, you will only be able to pour 1L into it, leaving you with exactly 4L in the 5L jug.

Now you have exactly 4L of water in the 5L jug.

Are We Doing Well?

How do you get 5L water using 9L and 5L jugs?

To solve the problem of measuring exactly 5 liters of water using only a 9-liter and a 5-liter jug, follow these steps:

1. Fill the 9-liter jug to the top.
2. Use the 9-liter jug to fill the 5-liter jug. This leaves you with 4 liters in the 9-liter jug.
3. Empty the 5-liter jug.
4. Transfer the 4 liters from the 9-liter jug to the 5-liter jug.
5. Fill the 9-liter jug to the top again.
6. Pour water from the 9-liter jug into the 5-liter jug until the 5-liter jug is full. Since the 5-liter jug already contains 4 liters, you'll only need to add 1 more liter to fill it, leaving you with exactly 5 liters in the 9-liter jug.

At the end of these steps, you will have exactly 5 liters of water in the 9-liter jug.

9 – 1 = 5 ???

Which Student is Smarter?

$12 \times 15 = 180$

Okay, I memorized $12 \times 15 = 180$.

Ones: $2 \times 5 = 10$. Carry 1 to the tens.
Tens: $2 \times 1 = 2$. $1 \times 5 = 5$. $2 + 5 + 1 = 8$.
Hundreds: $1 \times 1 = 1$. So $12 \times 15 = 180$.

Which Student is Smarter?

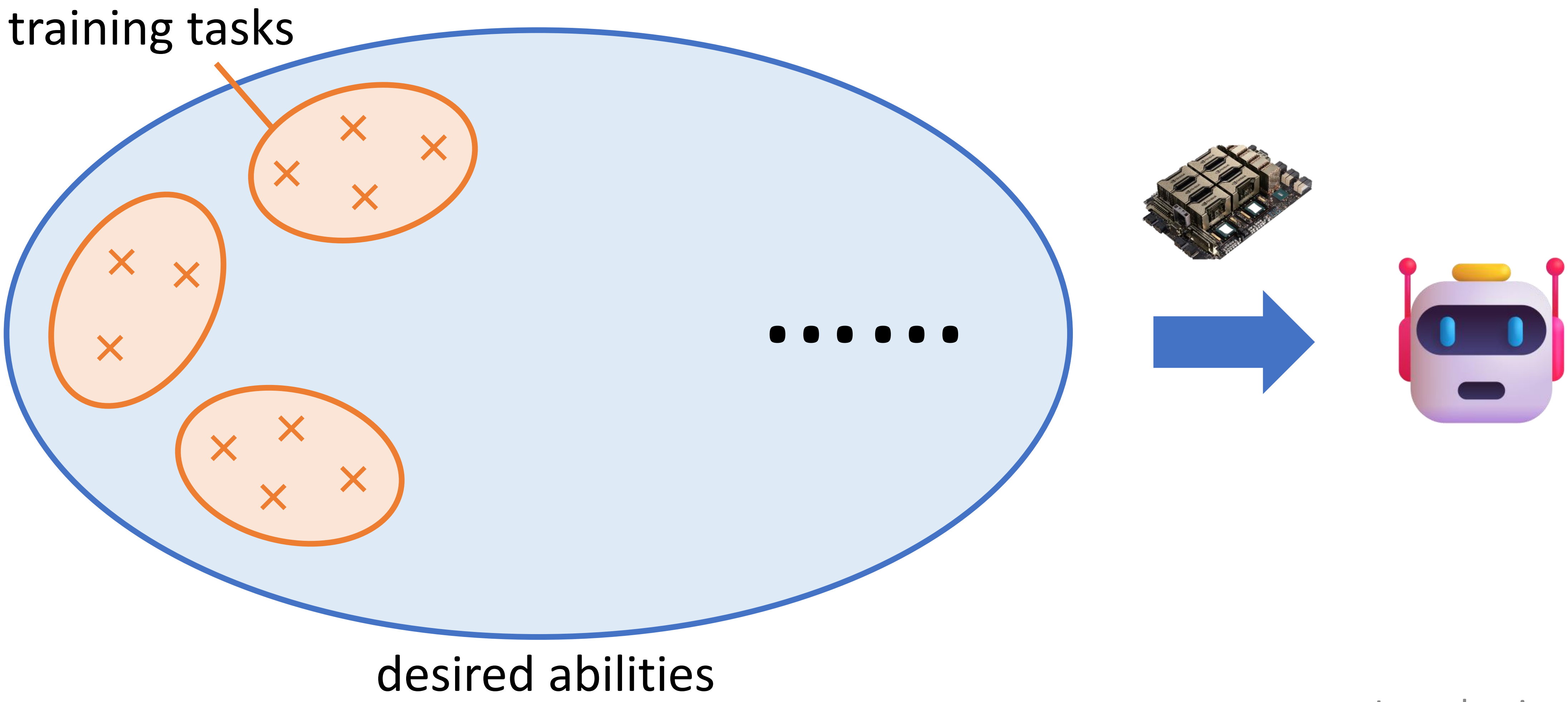
$22 \times 15 = ?$

It looks like 12×15. $22 \times 15 = 180$.

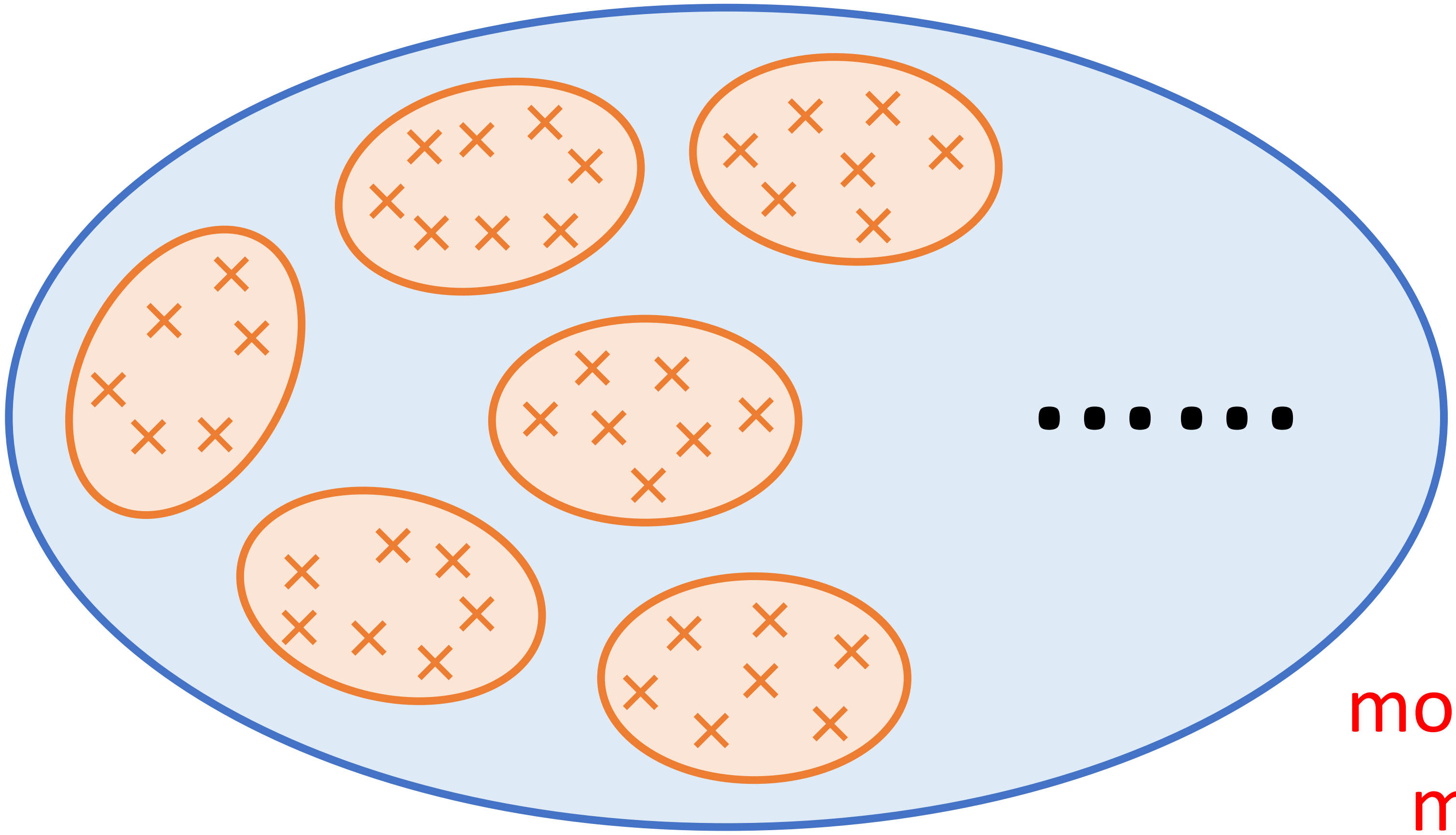
Ones: $2 \times 5 = 10$. Carry 1 to the tens.
Tens: $2 \times 1 = 2$. $2 \times 5 = 10$. $2 + 10 + 1 = 13$.
Carry 1 to the hundreds.
Hundreds: $2 \times 1 = 2$. $2 + 1 = 3$. So $22 \times 15 = 330$.

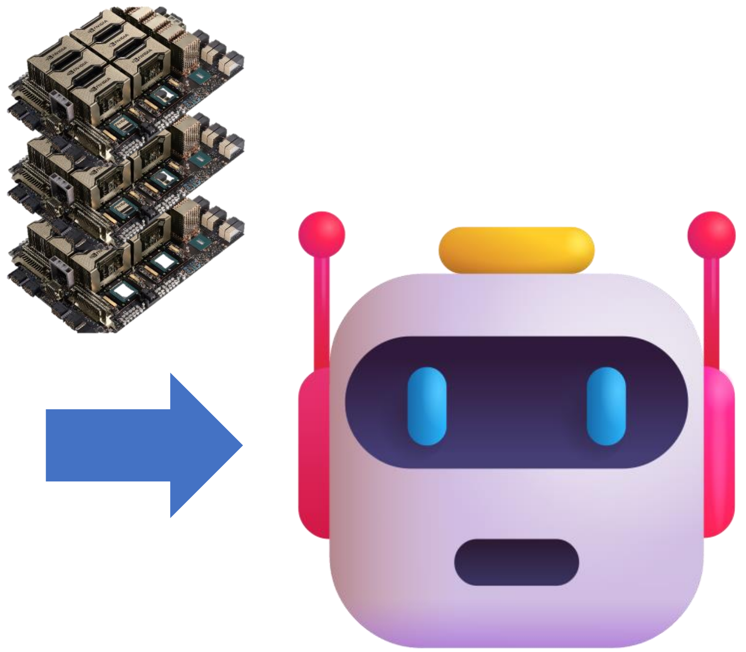
If we **induce a general principle** from samples, it **can be applied to new scenarios**.

The Way We Build A(G)I



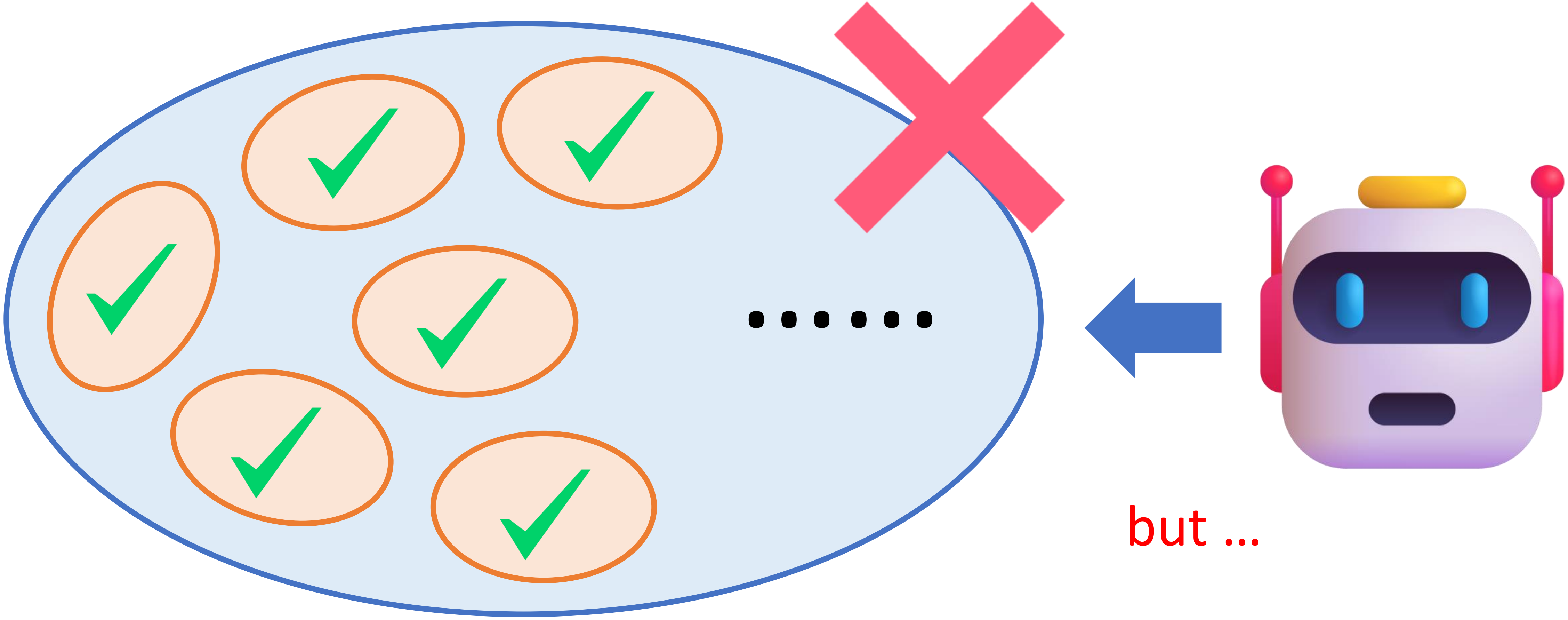
The Way We Build A(G)I



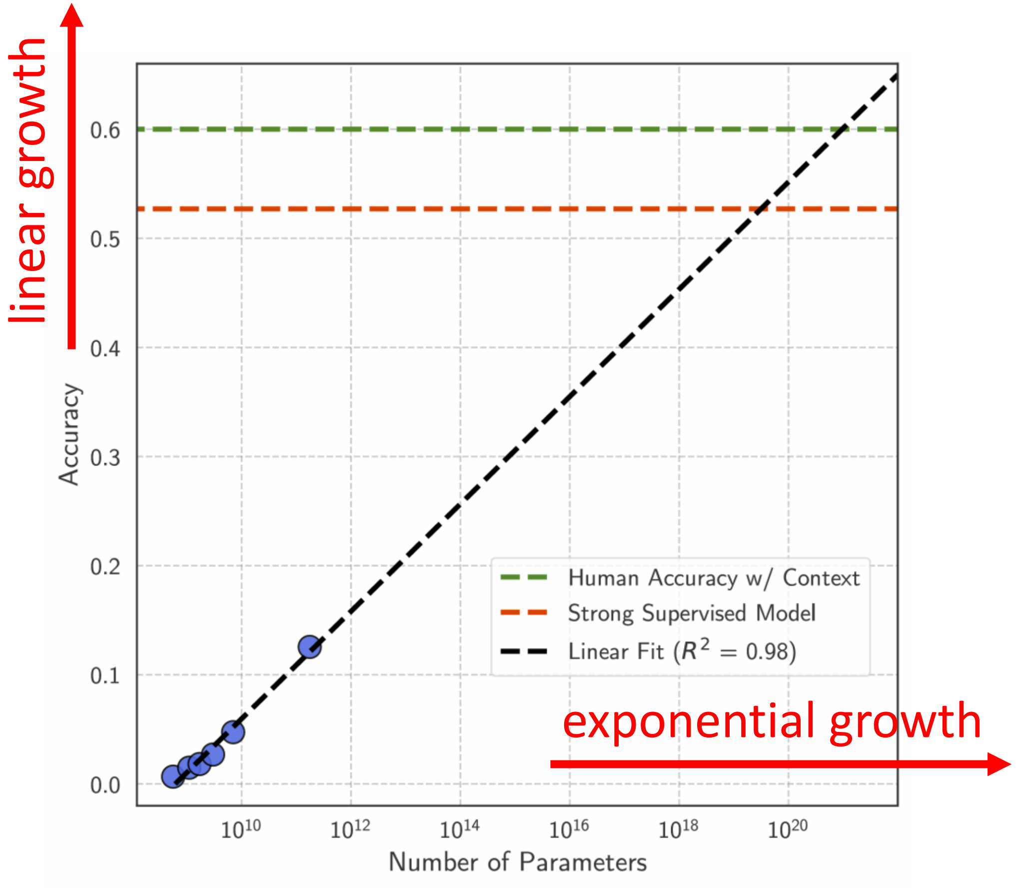


more training data
more compute

The Way We Build A(G)I

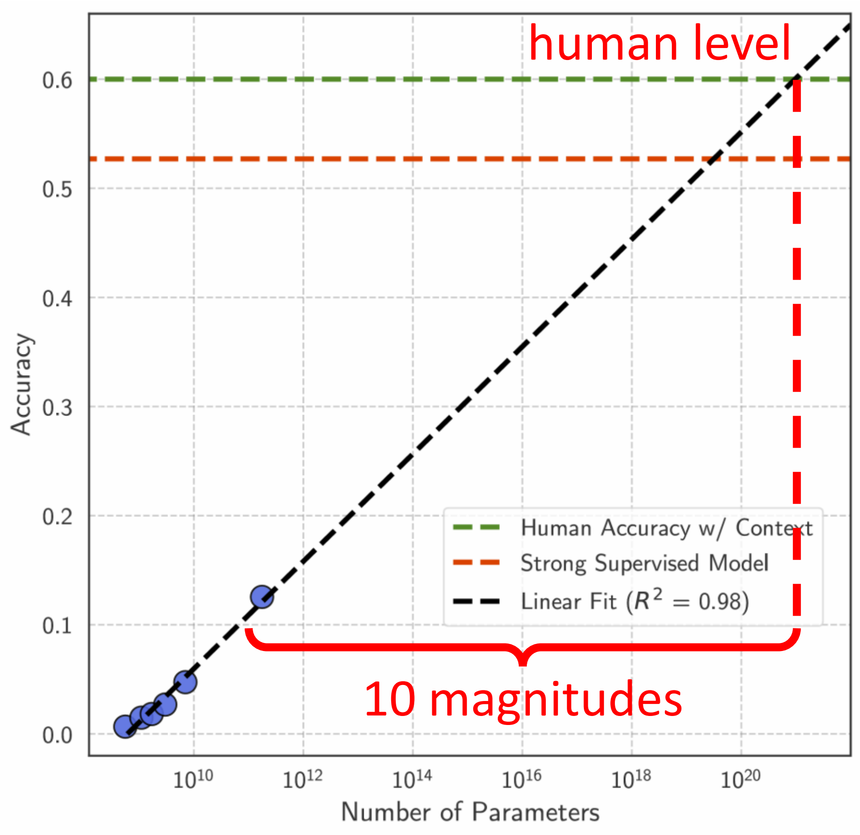


Scaling Laws



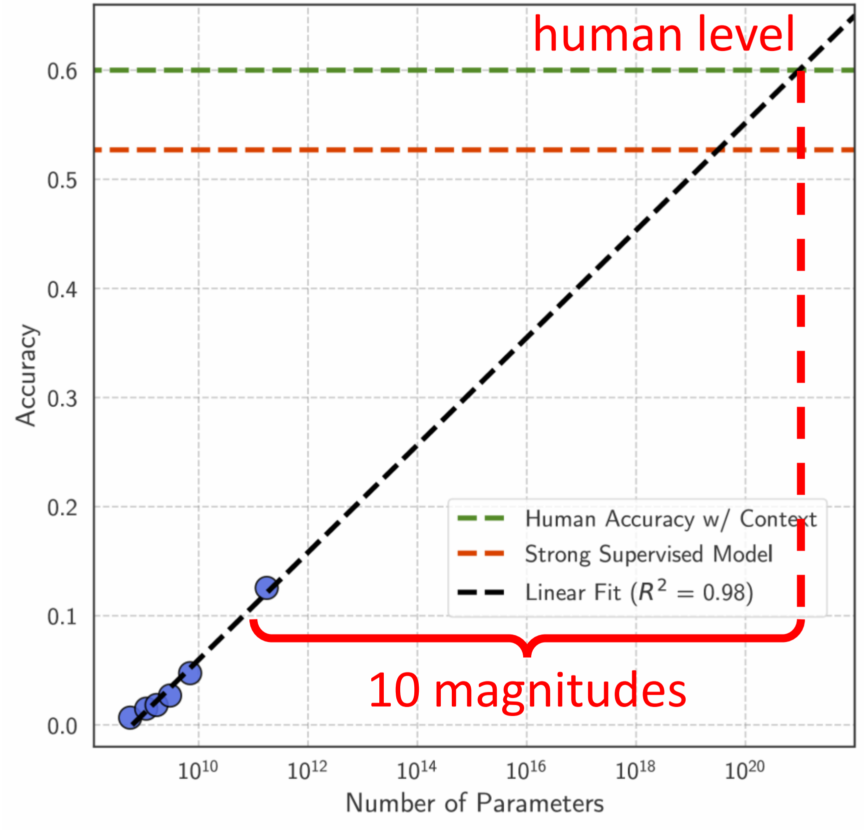
[1] Nikhil Kandpal, et al. Large Language Models Struggle to Learn Long-Tail Knowledge. ICML 2023.

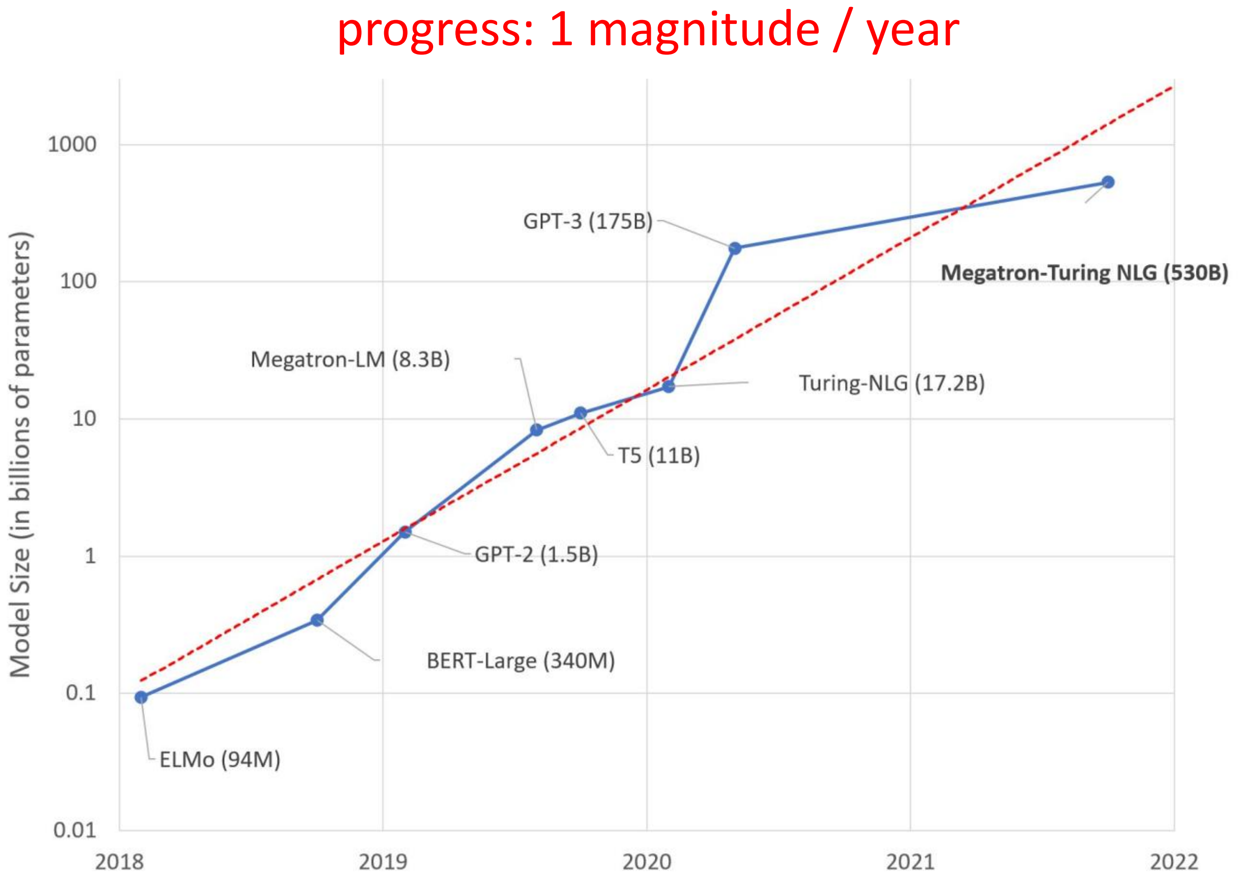
A Long Way to Go…



[1] Nikhil Kandpal, et al. Large Language Models Struggle to Learn Long-Tail Knowledge. ICML 2023.

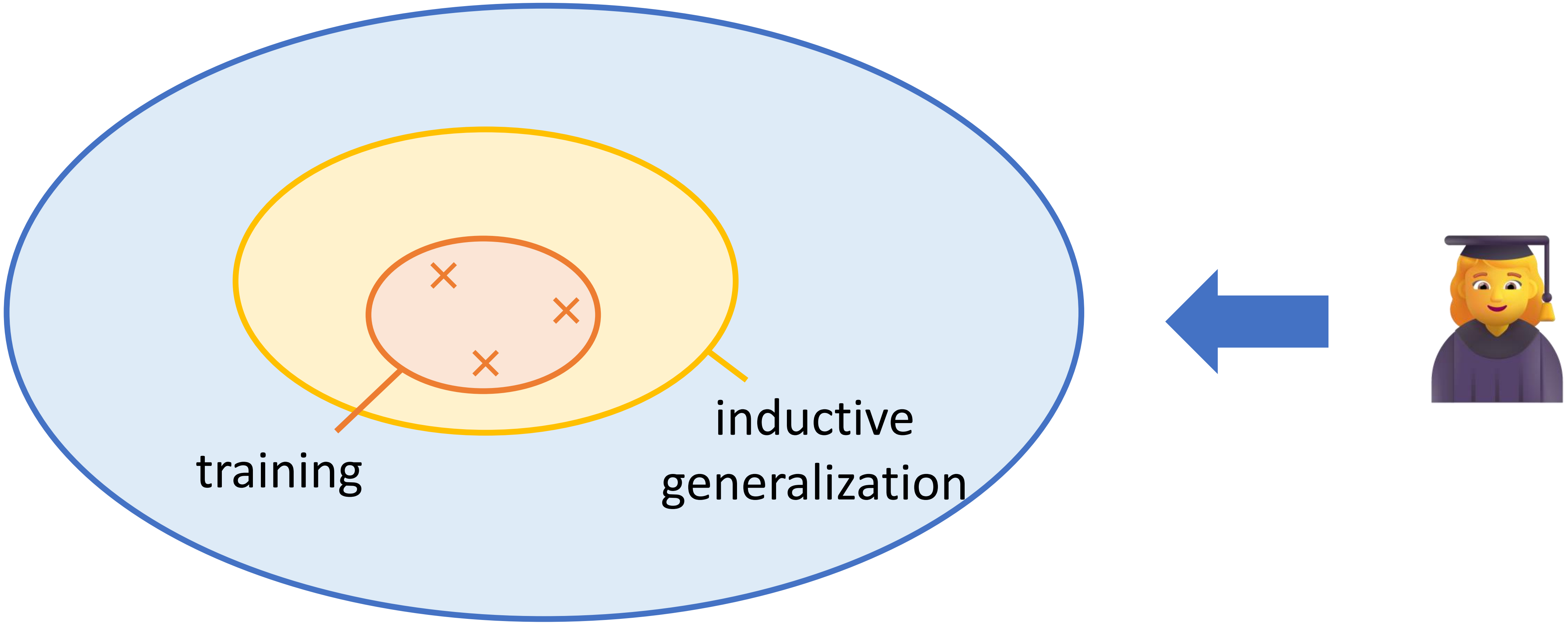
A Long Way to Go…



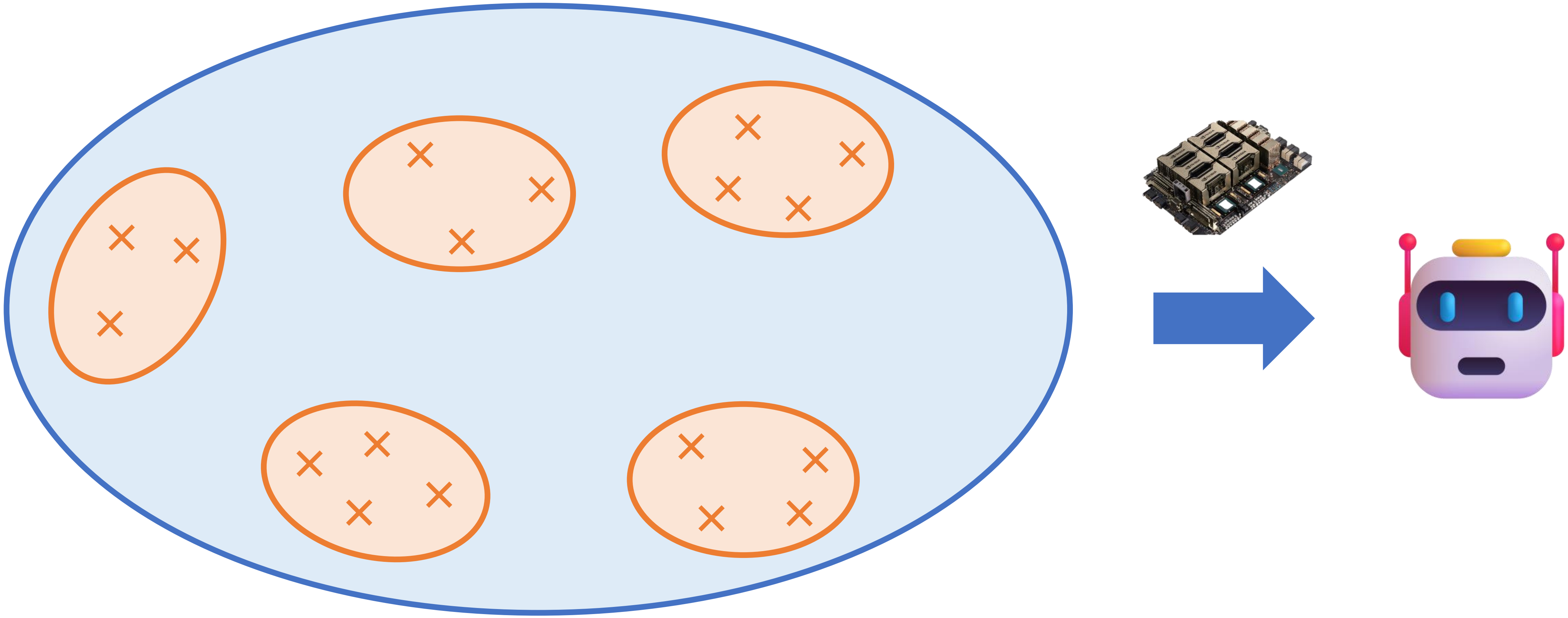


[1] Nikhil Kandpal, et al. Large Language Models Struggle to Learn Long-Tail Knowledge. ICML 2023.
[2] Julien Simon. Large Language Models: A New Moore's Law? HuggingFace blog. 2021.

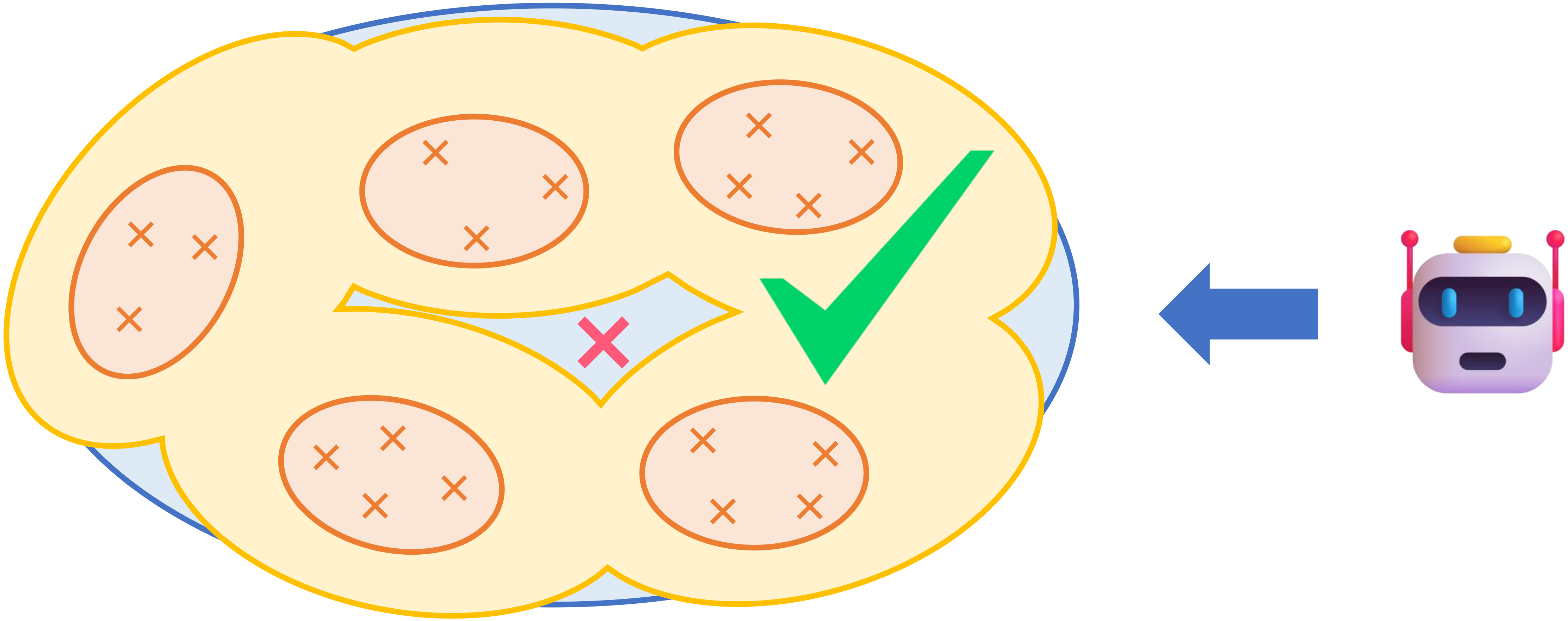
The Way We Learn



A Better Way to Build A(G)I

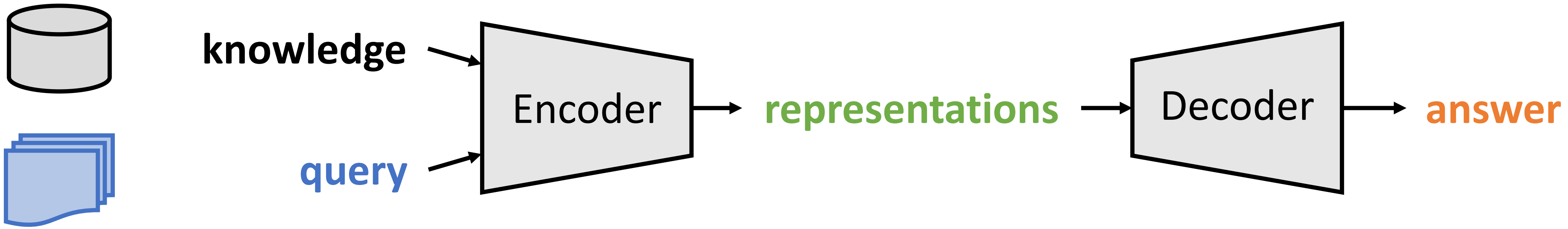


A Better Way to Build A(G)I

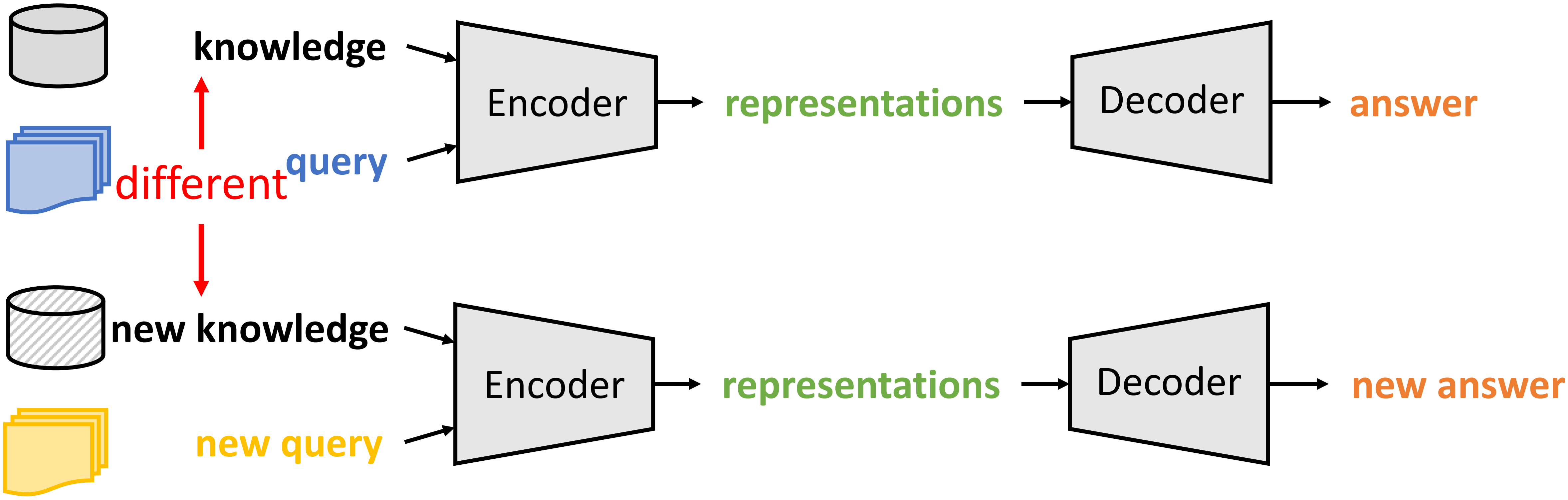


What **generalization** do we need for **representation learning models**?

Representation Learning

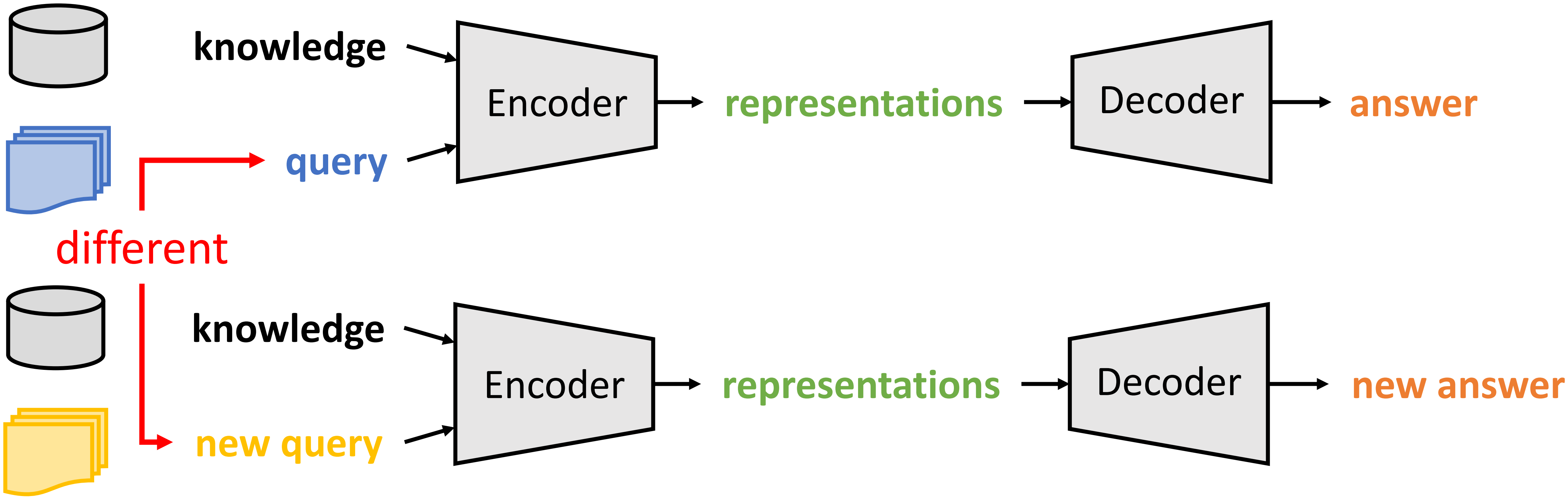


Generalization to New Knowledge



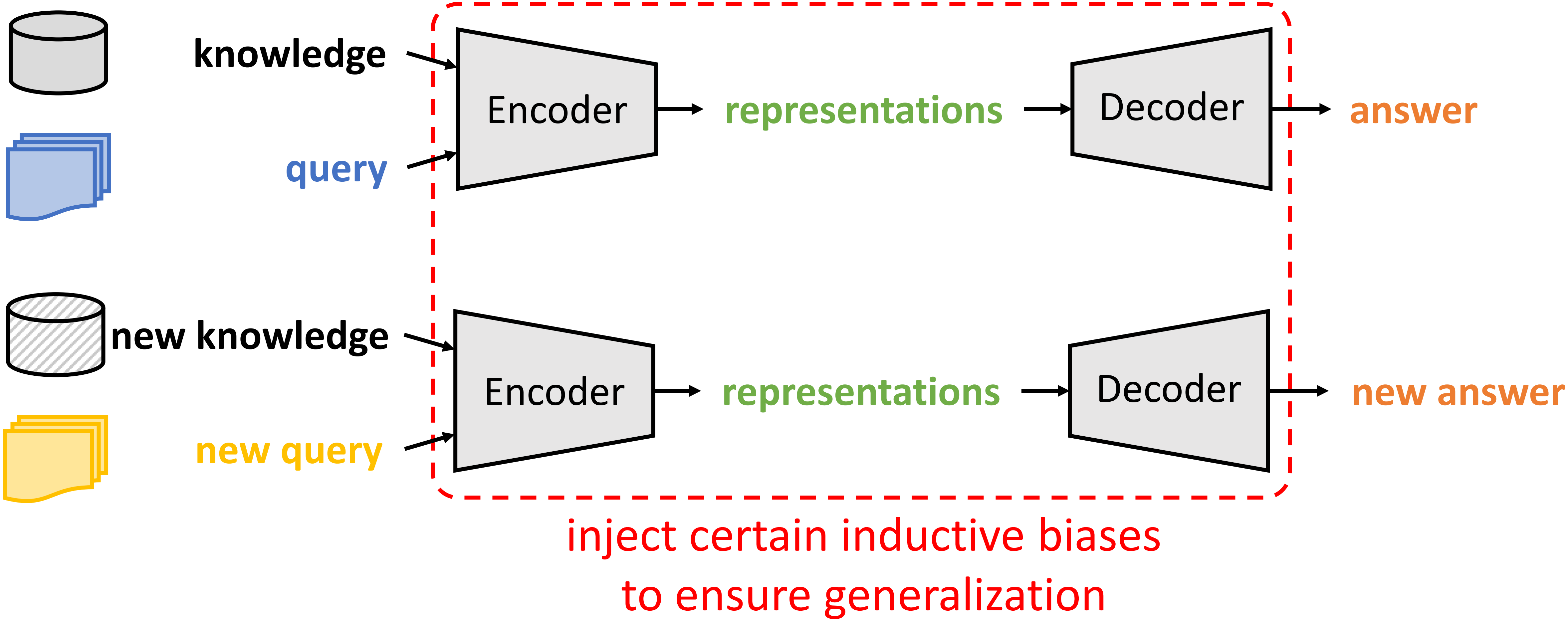
often studied as inductive generalization

Generalization to New Queries



often studied as compositional generalization

Our Methodology

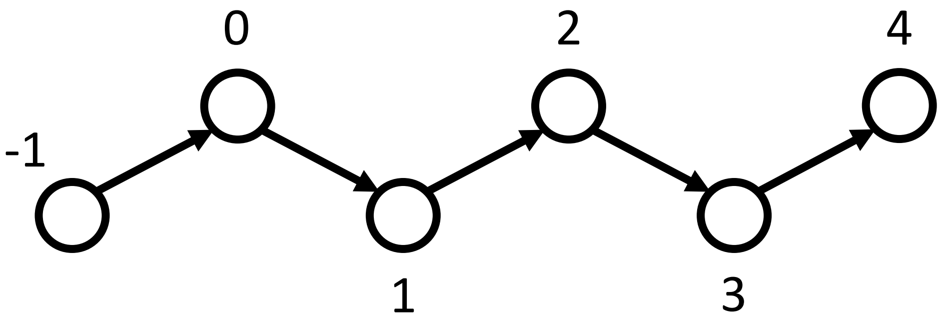


What kind of knowledge to generalize across?

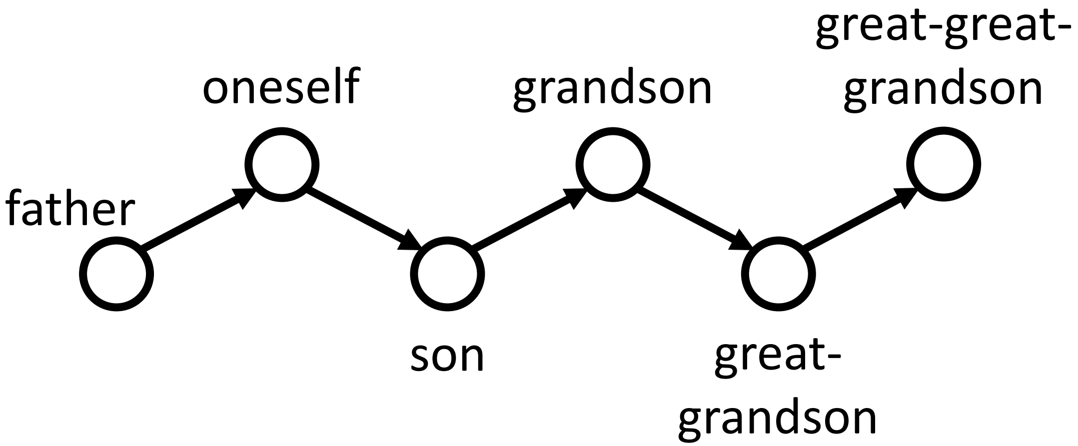
Structure

Structure of Reasoning Problems

What is the answer to
1 + 1 + 1 - 1 - 1?

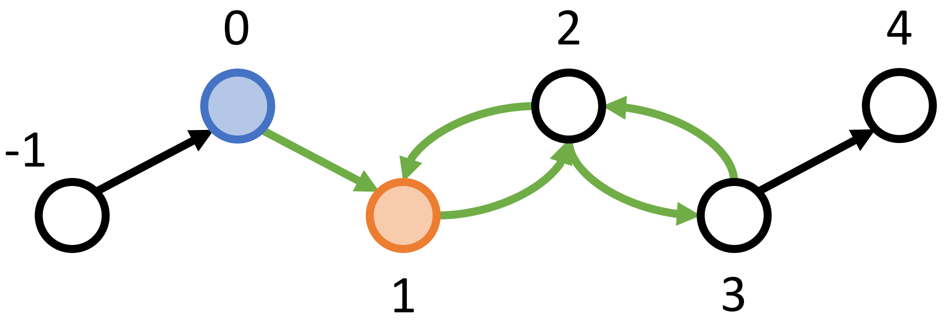


What is my son's son's
son's father's father?

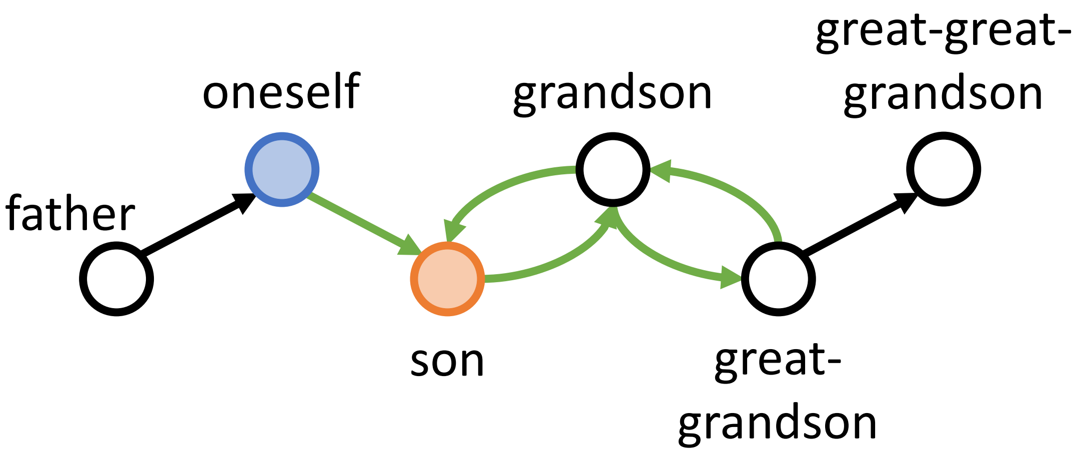


Structure of Reasoning Problems

What is the answer to
1 + 1 + 1 - 1 - 1?



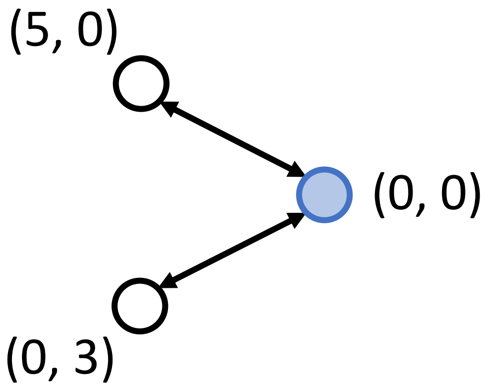
What is my son's son's
son's father's father?



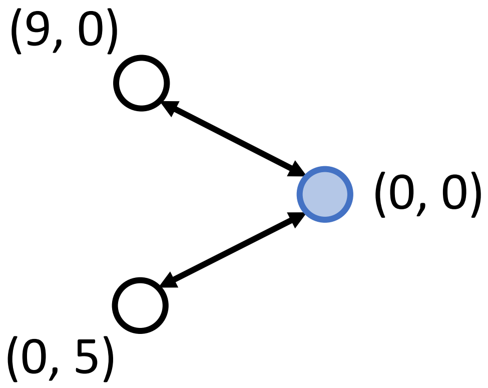
Both predict the ending node of a path!

Structure of Reasoning Problems

How do you get 4L water using 5L and 3L jugs?

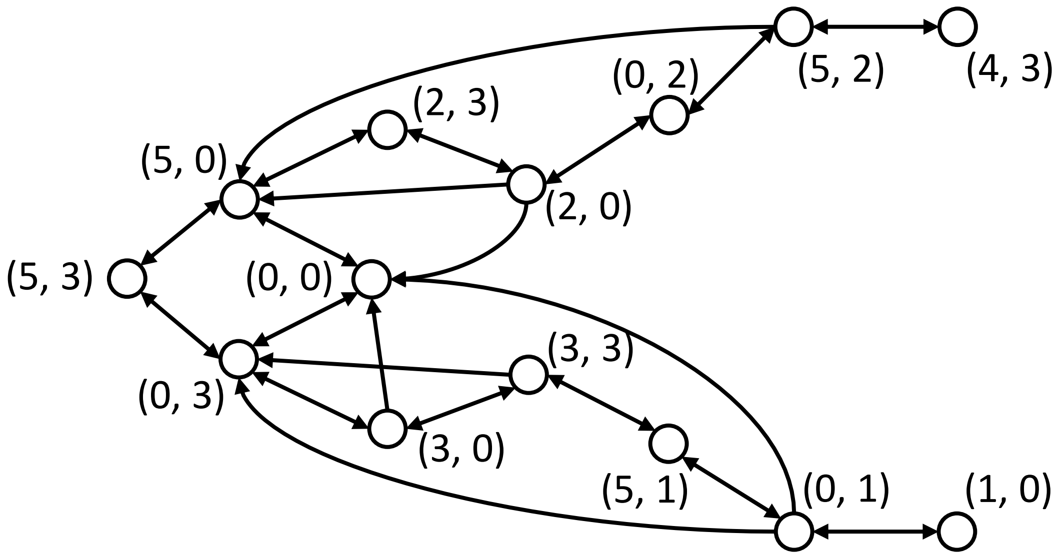


How do you get 5L water using 9L and 5L jugs?

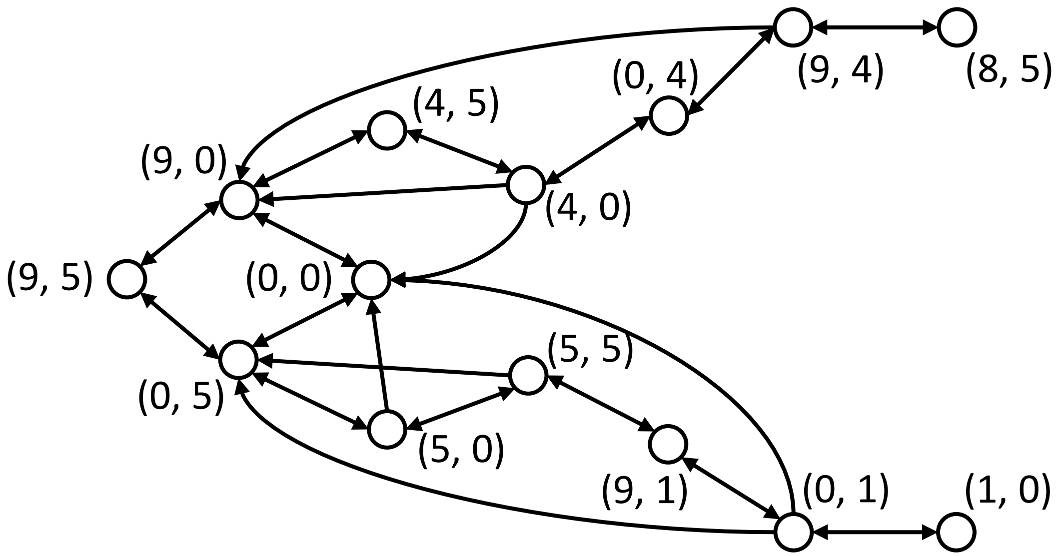


Structure of Reasoning Problems

How do you get 4L water using 5L and 3L jugs?

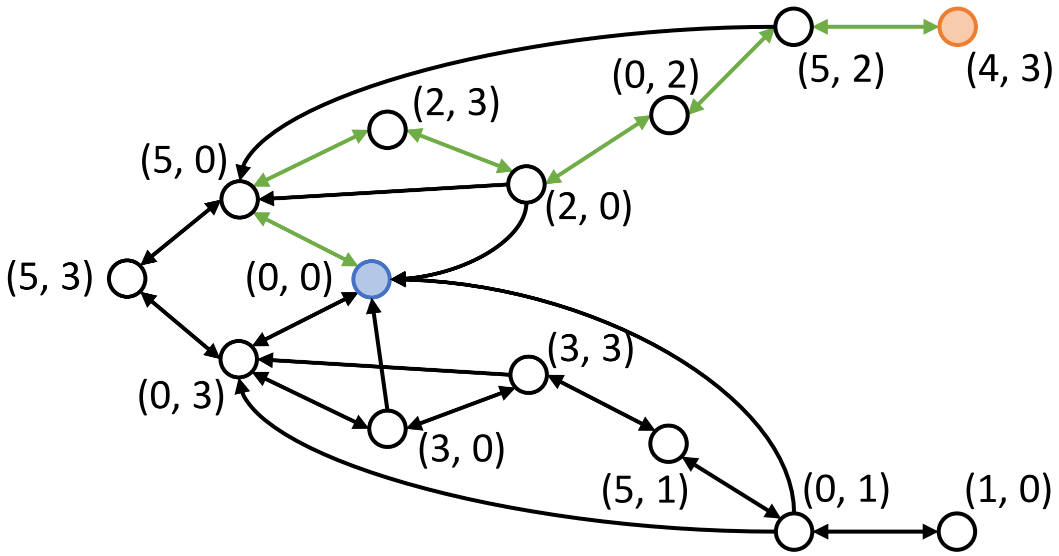


How do you get 5L water using 9L and 5L jugs?

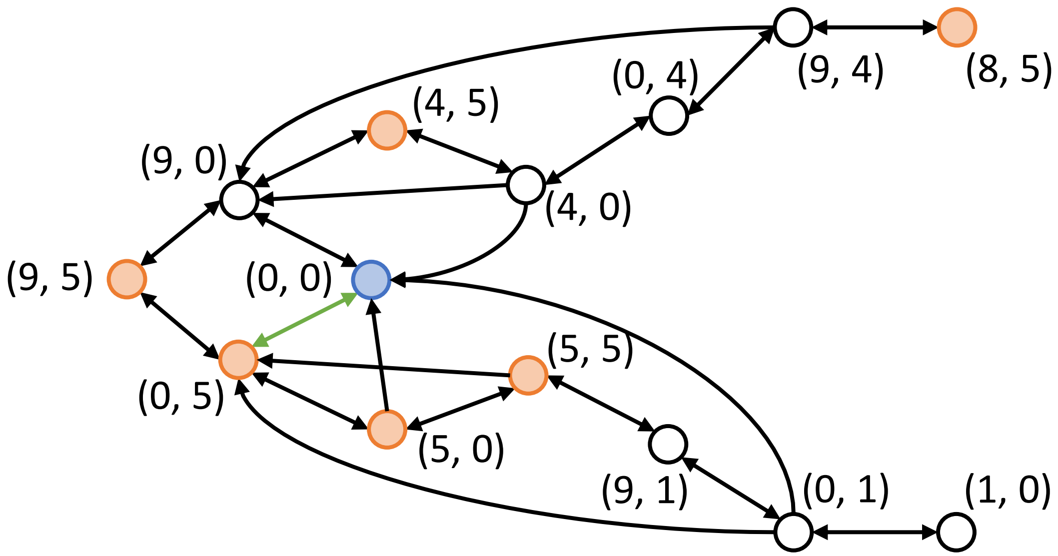


Structure of Reasoning Problems

How do you get 4L water using 5L and 3L jugs?



How do you get 5L water using 9L and 5L jugs?



Both find a path to reach the target node(s)!

Introduction — Our Works — Conclusions

How to generalize across knowledge structures?

How to generalize across query structures?

How to make ML on structured data more accessible?

Representation Learning Works

Method	Knowledge Structure	Query Structure	Generalization to New		
			Entities	Relations	Multi-hop Queries
Embeddings	Knowledge graph	Single-hop query			
NBFNet	Knowledge graph	Single-hop query	✓		
A*Net	Knowledge graph	Single-hop query	✓		
Ultra	Knowledge graph	Single-hop query	✓	✓	
Embeddings	Knowledge graph	Multi-hop query			✓
GNN-QE	Knowledge graph	Multi-hop query	✓		✓✓
UltraQuery	Knowledge graph	Multi-hop query	✓	✓	✓✓
CoT	Natural language (latent)	Multi-step query			✓
HtT	Natural language (latent)	Multi-step query			✓✓

covered in this talk

System Works

TorchDrug

simplifies development on structured data
reduce the lines of code by 20×

covered in this talk

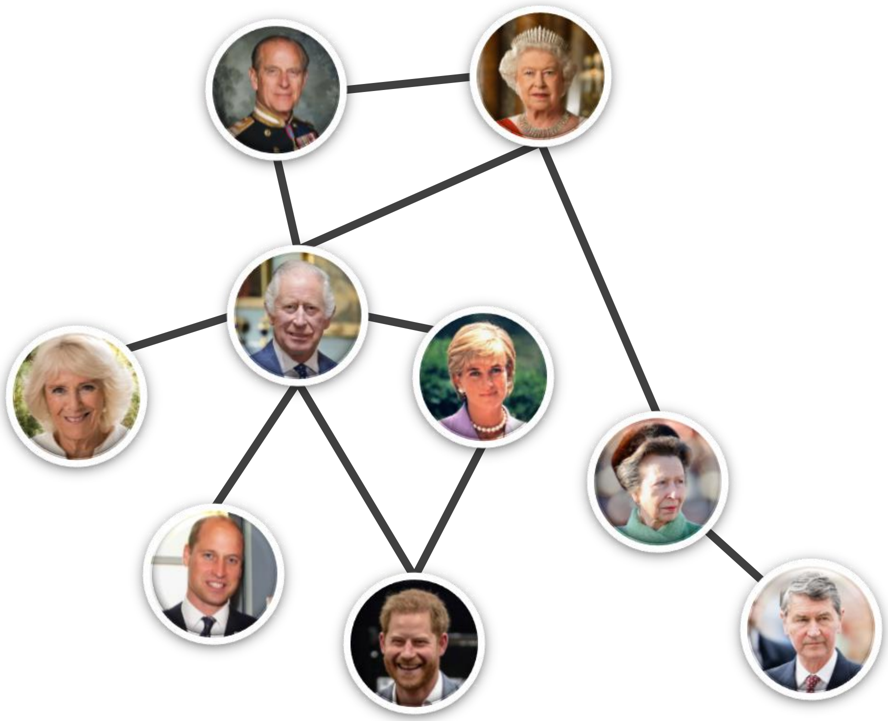
GraphVite

scales up training embedding methods
speeds up by 51× on million-scale graphs

NBFNet[1]: Learning **inductive representations of structures** by **encoding paths**

[1] **Zhaocheng Zhu**, Zuobai Zhang, Louis-Pascal Xhonneux, Jian Tang. Neural Bellman-Ford Networks: A General Graph Neural Network Framework for Link Prediction. NeurIPS 2021.

A Simplified Setup: Knowledge Graphs



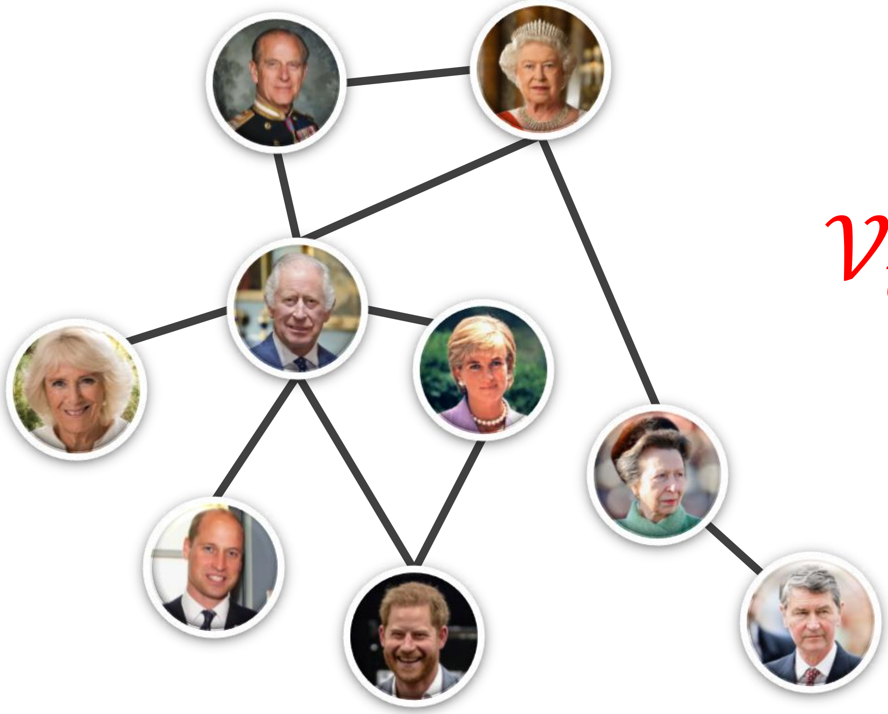
Graph $\mathcal{G} = (\mathcal{V}, \mathcal{R}, \mathcal{E})$

Entities $\mathcal{V}$: British royal family

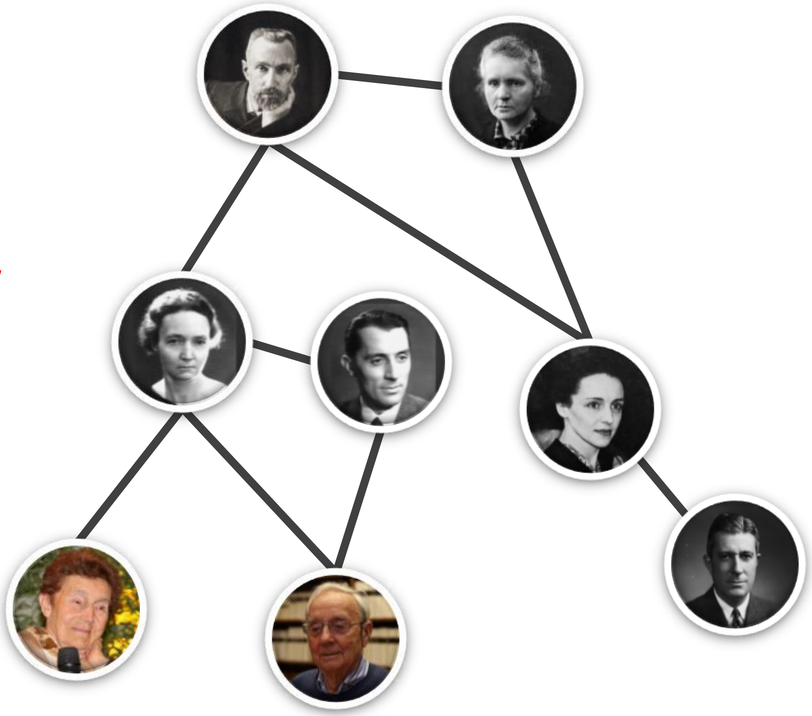
Relations $\mathcal{R}$: {parent, spouse}

Edges $\mathcal{E}$: known family relationships

Inductive Generalization on Structure



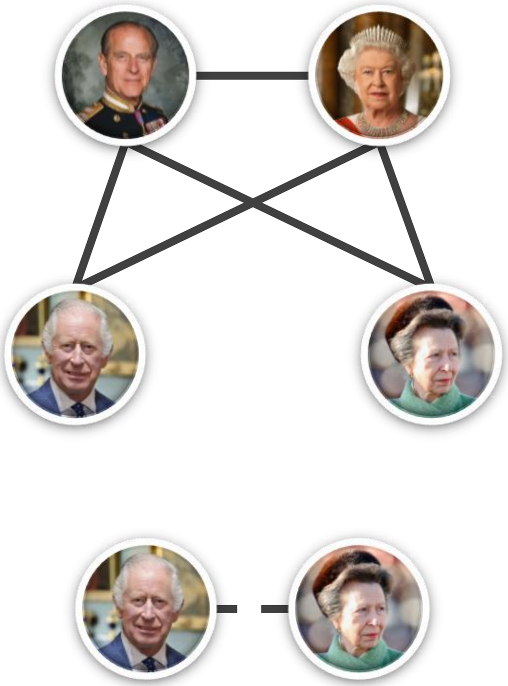
$\mathcal{V}_{train} \neq \mathcal{V}_{test}$

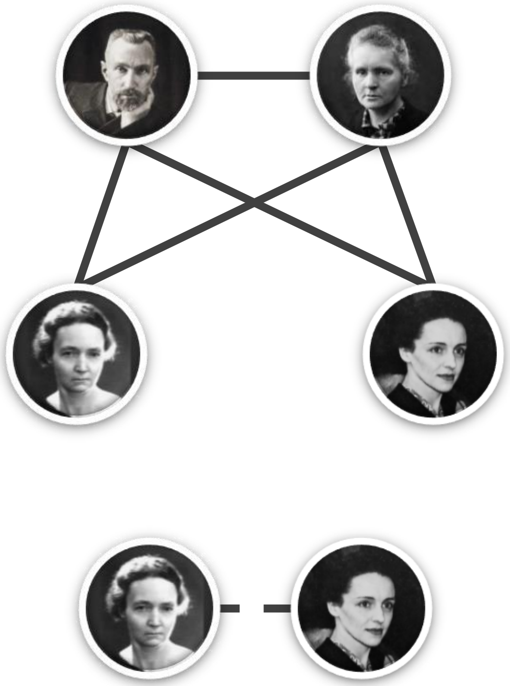


$\mathcal{V}$: British royal family
$\mathcal{R}$: {parent, spouse}

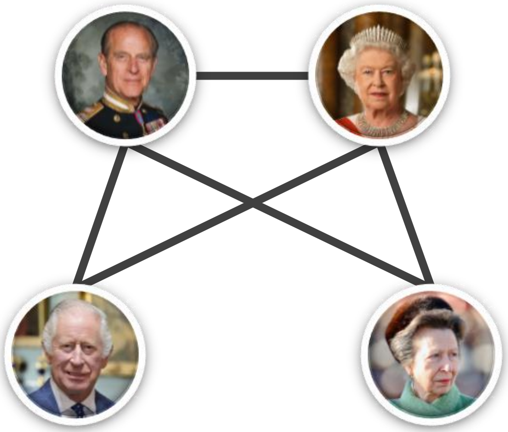
$\mathcal{V}$: Curie family
$\mathcal{R}$: {parent, spouse}

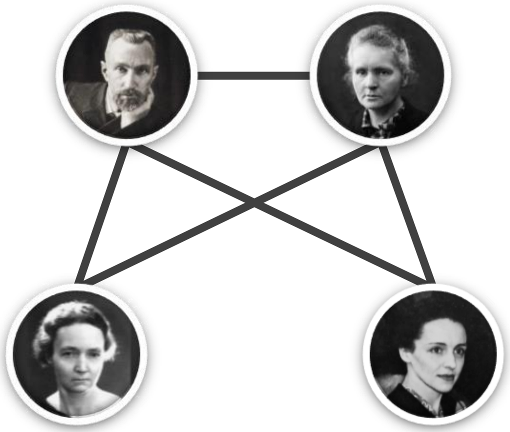
What Is an Inductive Function on Structure?





What Is an Inductive Function on Structure?





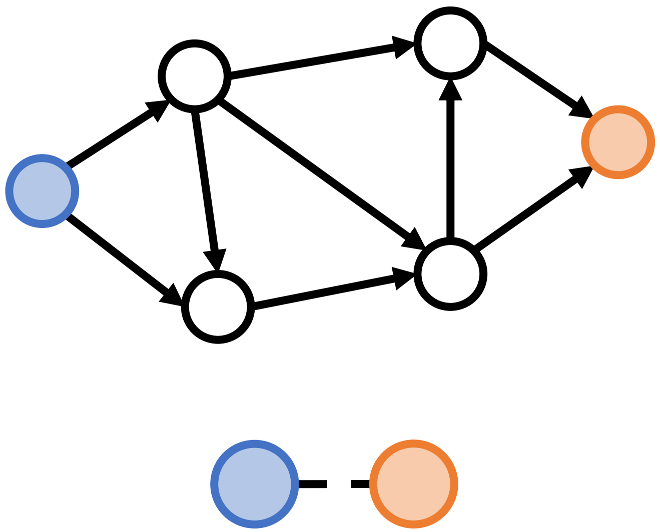
same structure

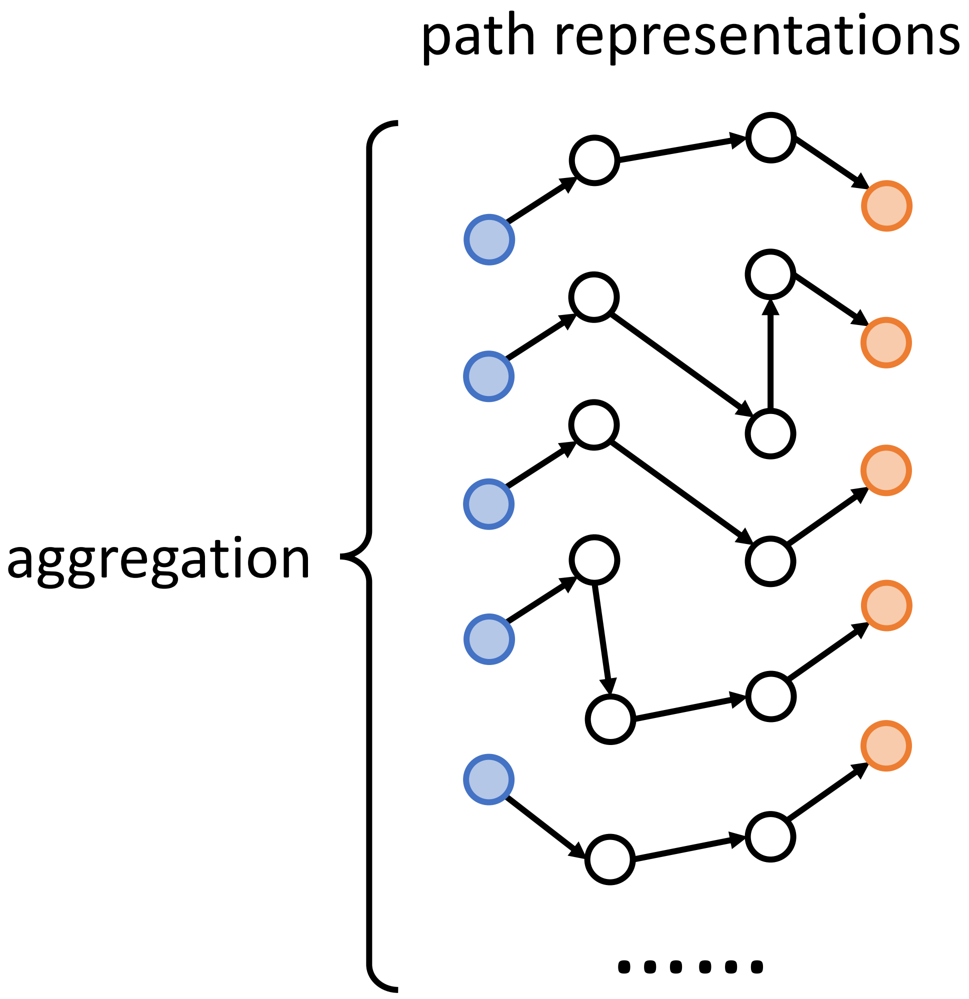
same value

distance: 2
#shortest path: 2
PageRank: 0.154

distance: 2
#shortest path: 2
PageRank: 0.154

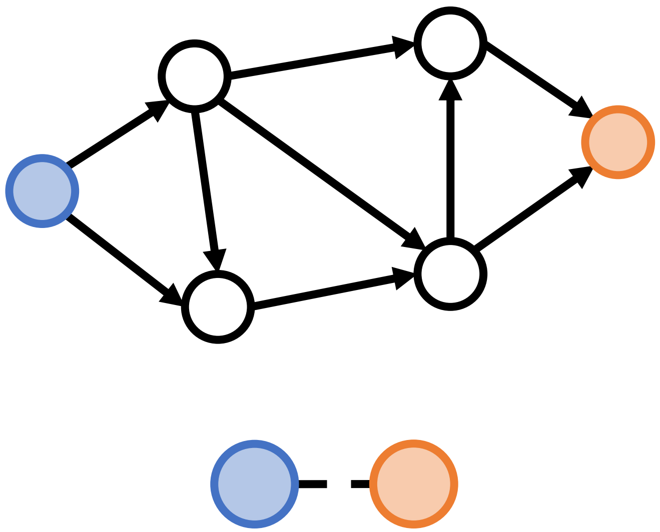
Path-based Methods

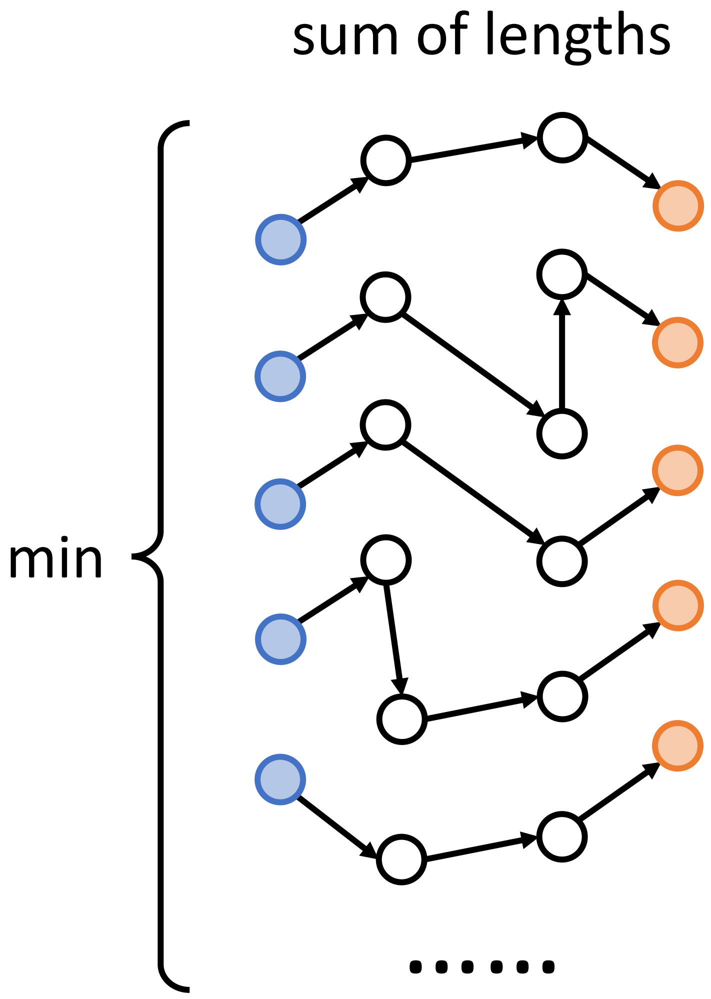




Path-based Methods

Graph distance

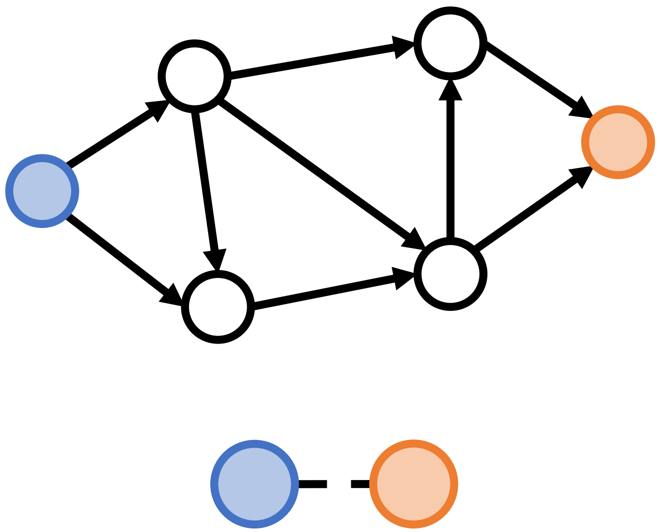


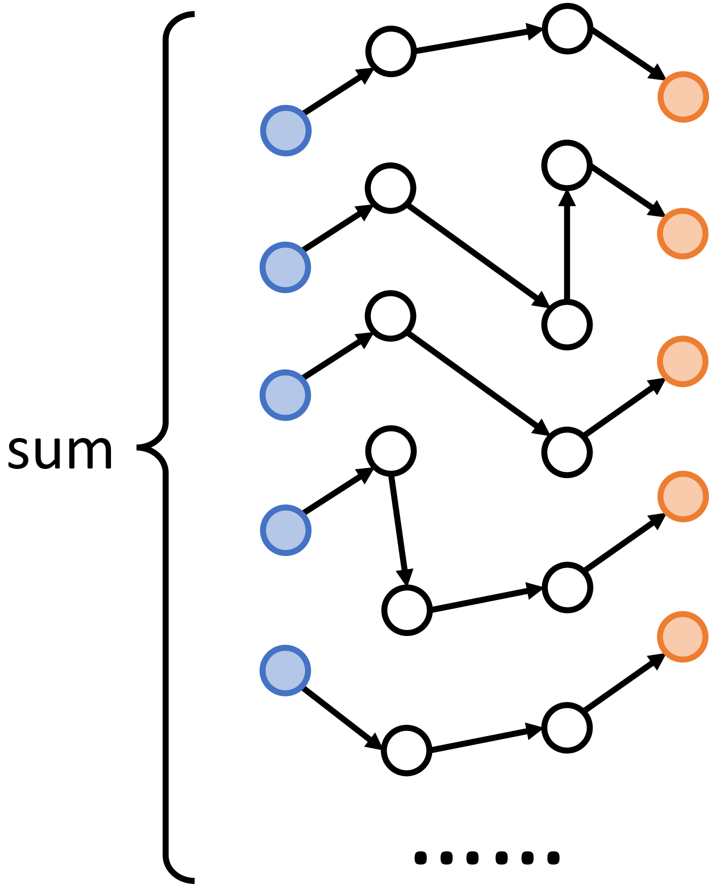


Path-based Methods

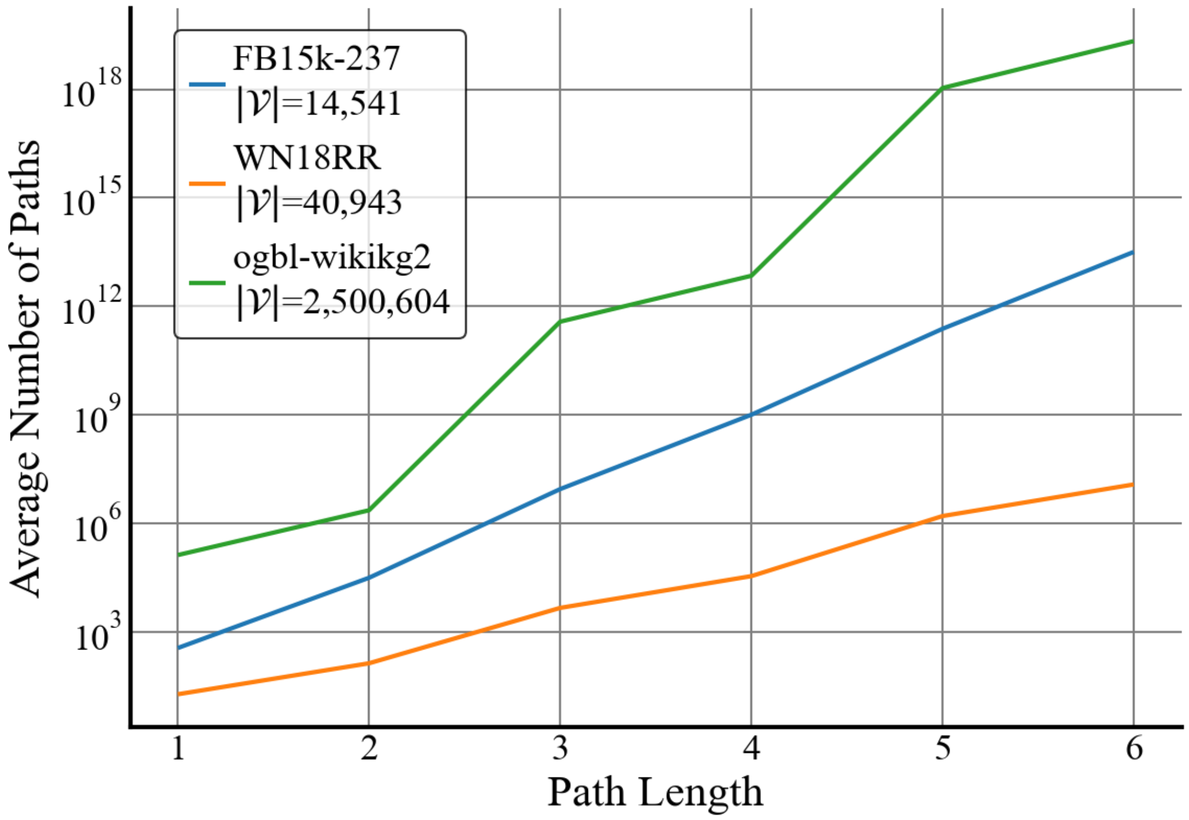
product of transition probabilities

Personalized PageRank





Scalability Issue



Dynamic Programming

To compute paths of length T

graph distance	DFS	$\rightarrow$	Bellman-Ford
Personalized PageRank	random walk	$\rightarrow$	power iteration

exponential in T $\qquad$ $O(T|\mathcal{E}|)$

Dynamic Programming

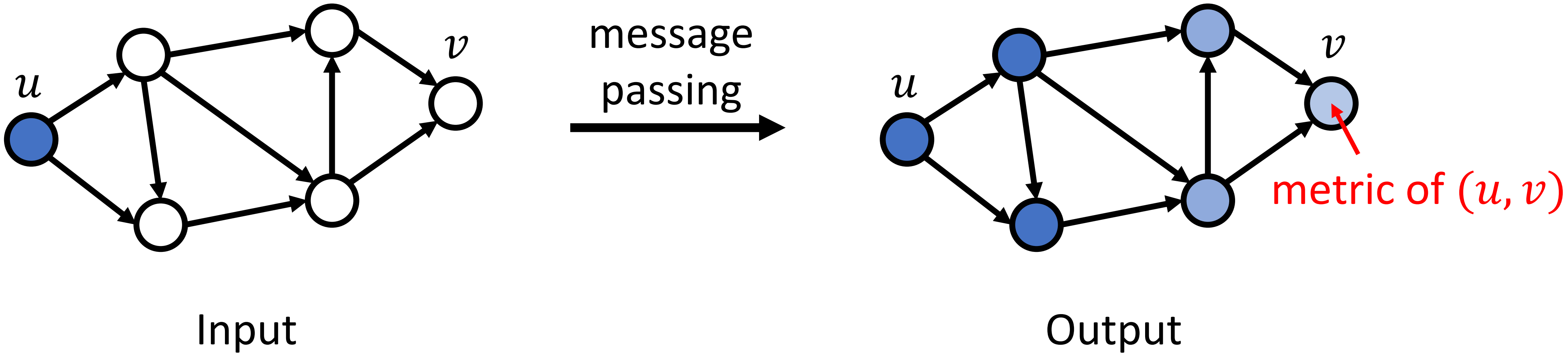
To compute paths of length T

instances of the generalized Bellman-Ford algorithm

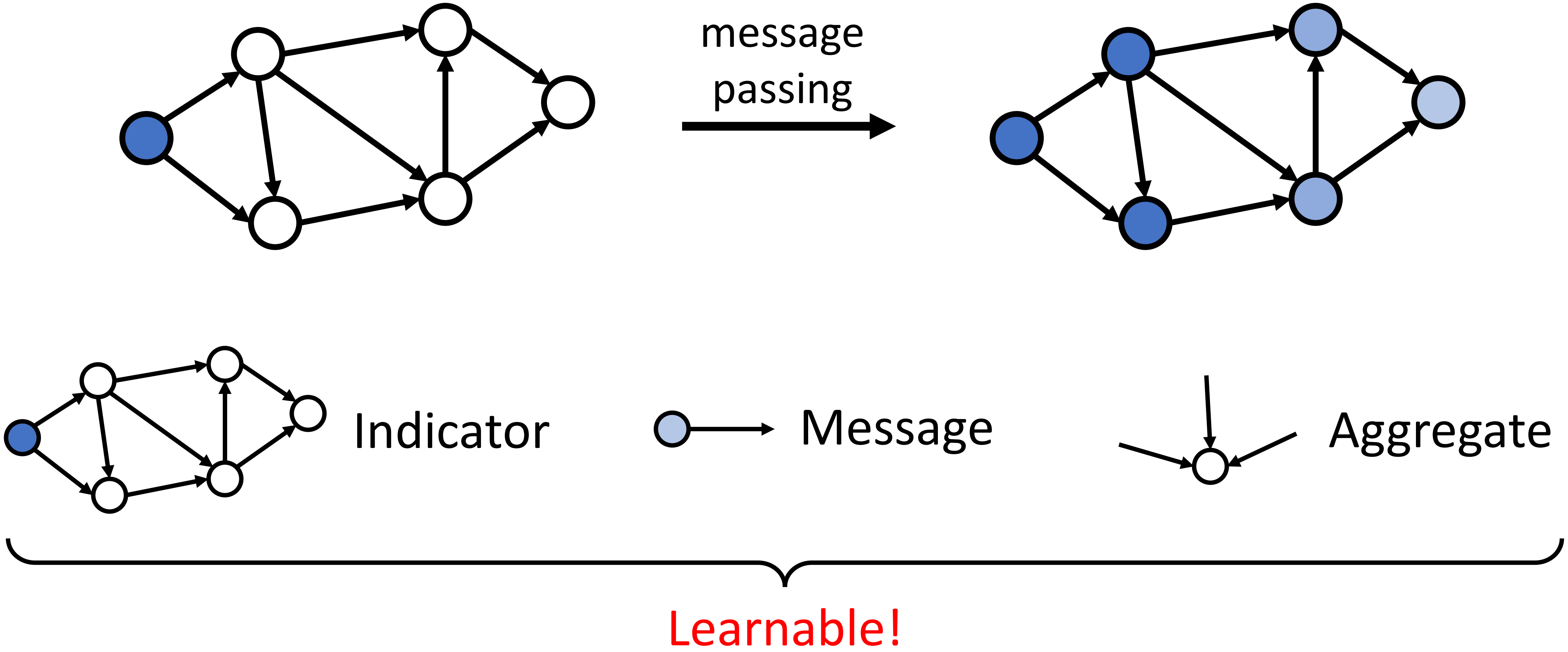
graph distance	DFS	→	Bellman-Ford
Personalized PageRank	random walk	→	power iteration
representation learning	encode each path	→	message passing
	exponential in T		$O(T\|\mathcal{E}\|)$

Generalized Bellman-Ford Algorithm

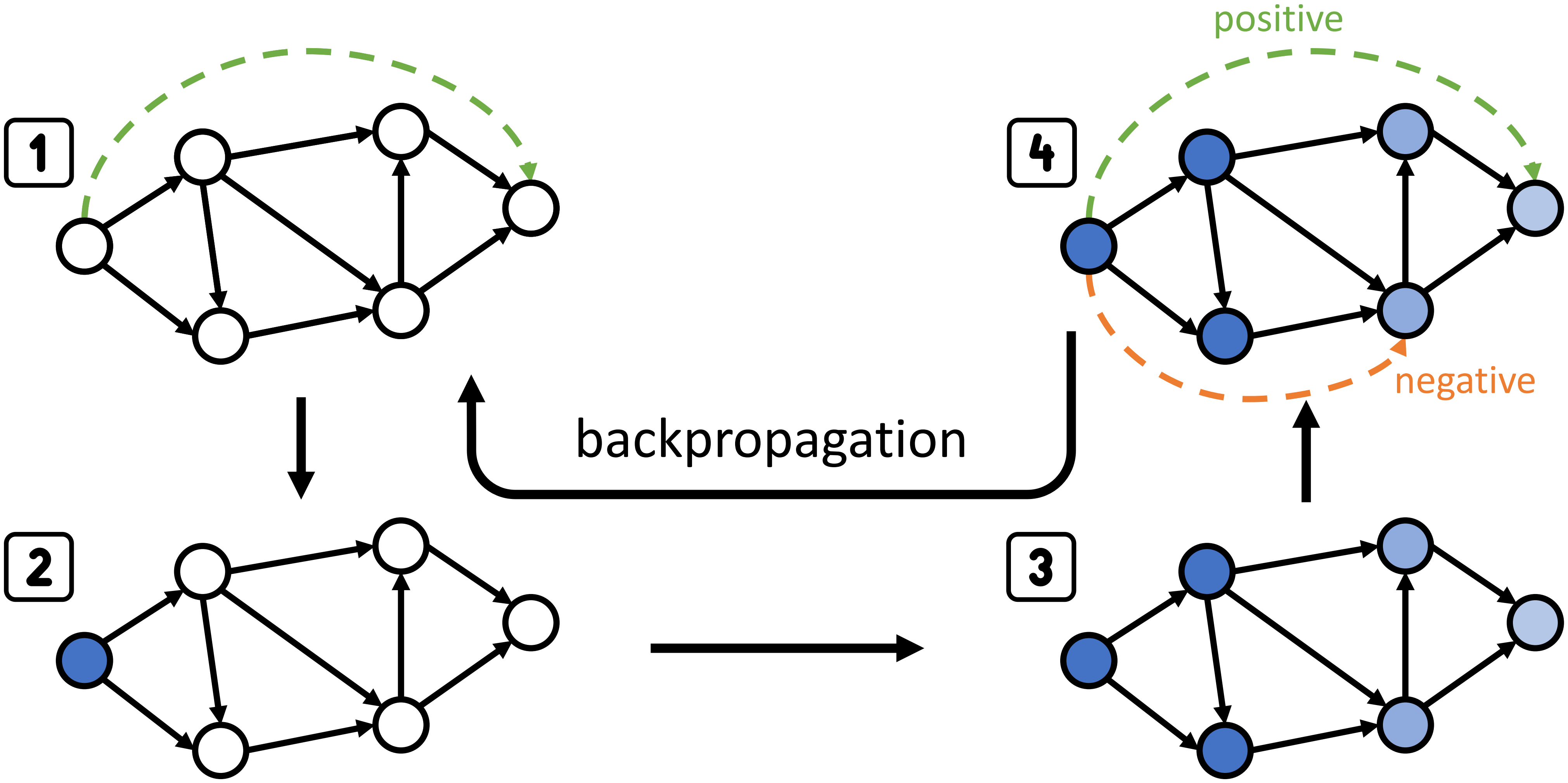
Message passing with **a single-source input**



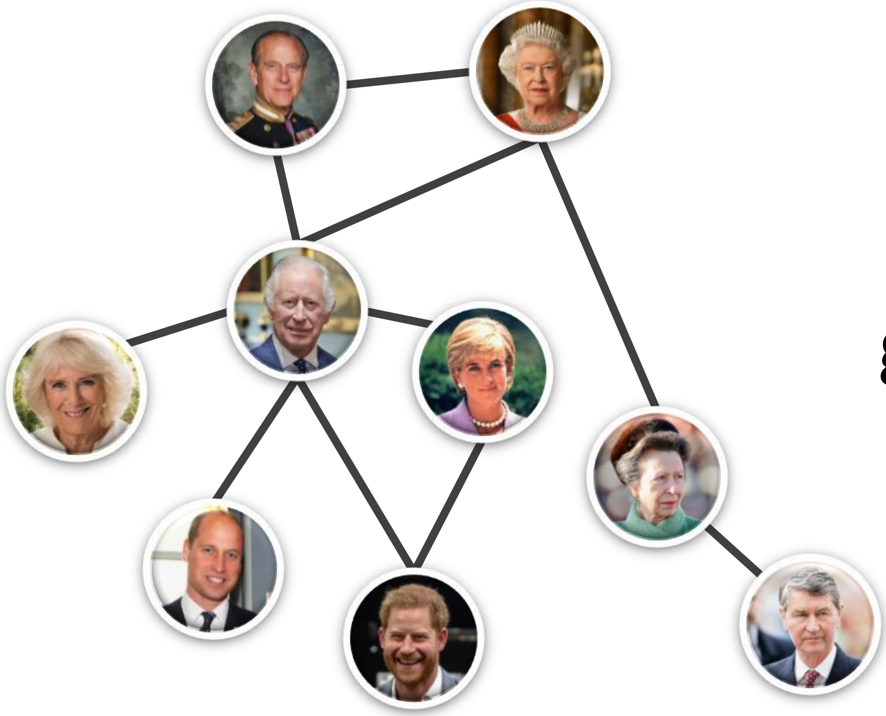
Neural Bellman-Ford Networks

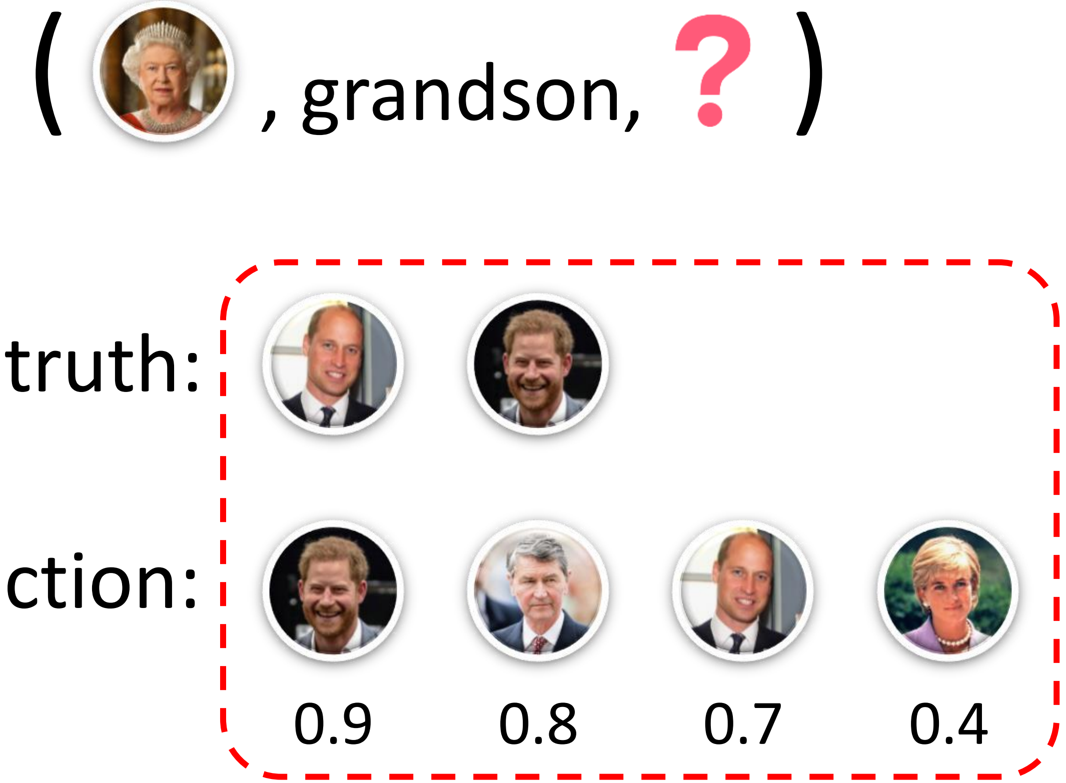


Learning Neural Bellman-Ford Networks



Evaluation: Knowledge Graph Completion





ranking metrics

Knowledge Graphs ($\mathcal{V}_{train} = \mathcal{V}_{test}$)

Class	Method	FB15k-237					WN18RR				
		MR↓	MRR↑	H@1↑	H@3↑	H@10↑	MR↓	MRR↑	H@1↑	H@3↑	H@10↑
Path-based	Path Ranking	3521	0.174	0.119	0.186	0.285	22438	0.324	0.276	0.360	0.406
	NeuralLP	-	0.240	-	-	0.362	-	0.435	0.371	0.434	0.566
	DRUM	-	0.343	0.255	0.378	0.516	-	0.486	0.425	0.513	0.586
Embeddings	TransE	357	0.294	-	-	0.465	3384	0.226	-	-	0.501
	DistMult	254	0.241	0.155	0.263	0.419	5110	0.43	0.39	0.44	0.49
	ComplEx	339	0.247	0.158	0.275	0.428	5261	0.44	0.41	0.46	0.51
	RotatE	177	0.338	0.241	0.375	0.533	3340	0.476	0.428	0.492	0.571
	HAKE	-	0.346	0.250	0.381	0.542	-	0.497	0.452	0.516	0.582
	LowFER	-	0.359	0.266	0.396	0.544	-	0.465	0.434	0.479	0.526
GNNs	RGCN	221	0.273	0.182	0.303	0.456	2719	0.402	0.345	0.437	0.494
	GraIL	2053	-	-	-	-	2539	-	-	-	-
	NBFNet	**114**	**0.415**	**0.321**	**0.454**	**0.599**	**636**	**0.551**	**0.497**	**0.573**	**0.666**

Knowledge Graphs ($\mathcal{V}_{train} \neq \mathcal{V}_{test}$)

metric: H@10↑

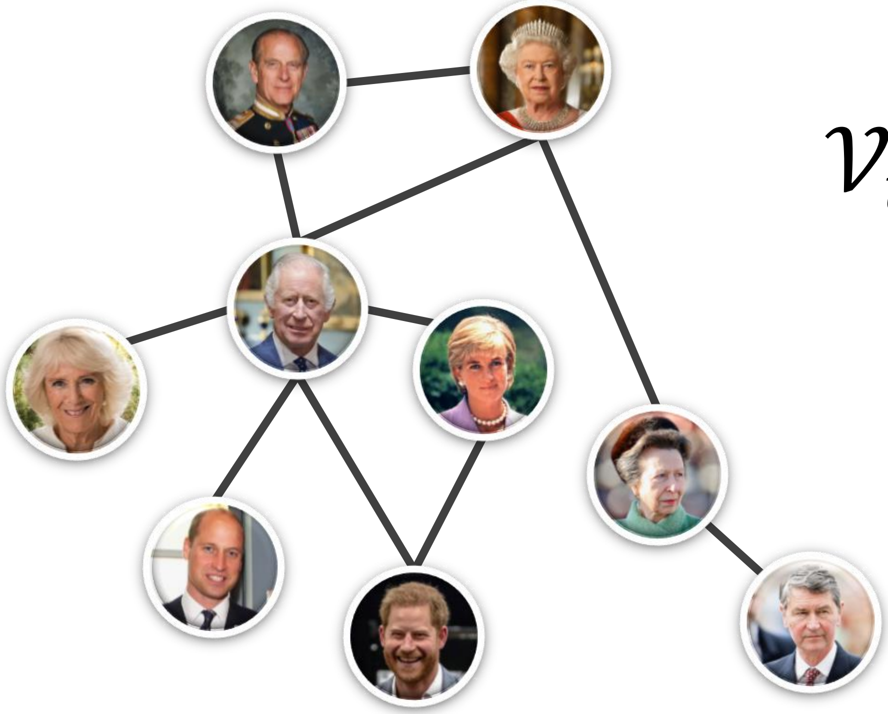
Class	Method	FB15k-237 v1	v2	v3	v4	WN18RR v1	v2	v3	v4
Path-based	NeuralLP	0.529	0.589	0.529	0.559	0.744	0.689	0.462	0.671
	DRUM	0.529	0.587	0.529	0.559	0.744	0.689	0.462	0.671
	RuleN	0.498	0.778	0.877	0.856	0.809	0.782	0.534	0.716
GNNs	GraIL	0.642	0.818	0.828	0.893	0.825	0.787	0.584	0.734
	NBFNet	**0.834**	**0.949**	**0.951**	**0.960**	**0.948**	**0.905**	**0.893**	**0.890**

(Drinking Water)

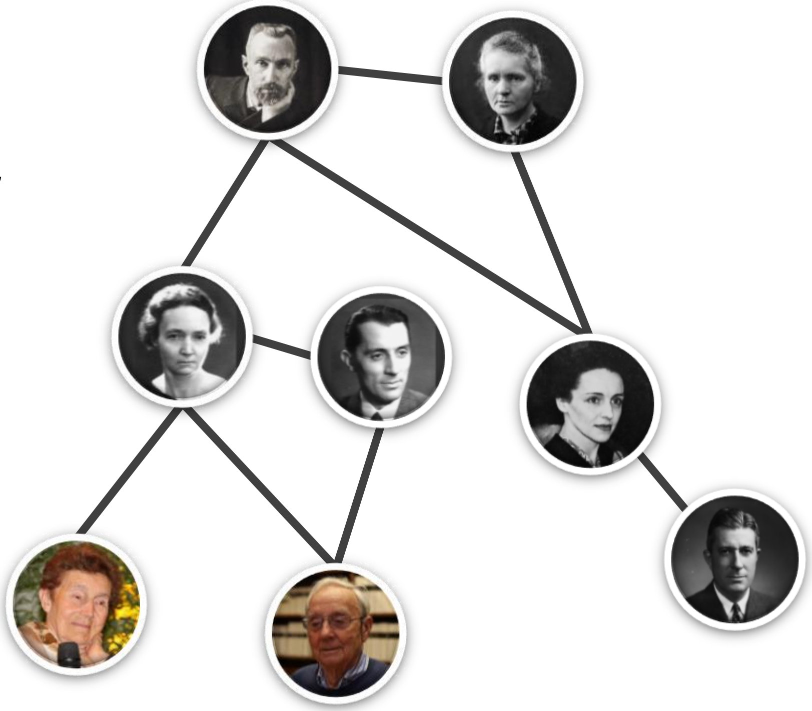
Ultra[1]: Generalizing to any knowledge graph with inductive relation representations

[1] Mikhail Galkin, Xinyu Yuan, Hesham Mostafa, Jian Tang, **Zhaocheng Zhu**. Towards Foundation Models for Knowledge Graph Reasoning. ICLR 2024.

Inductive Generalization on Structure



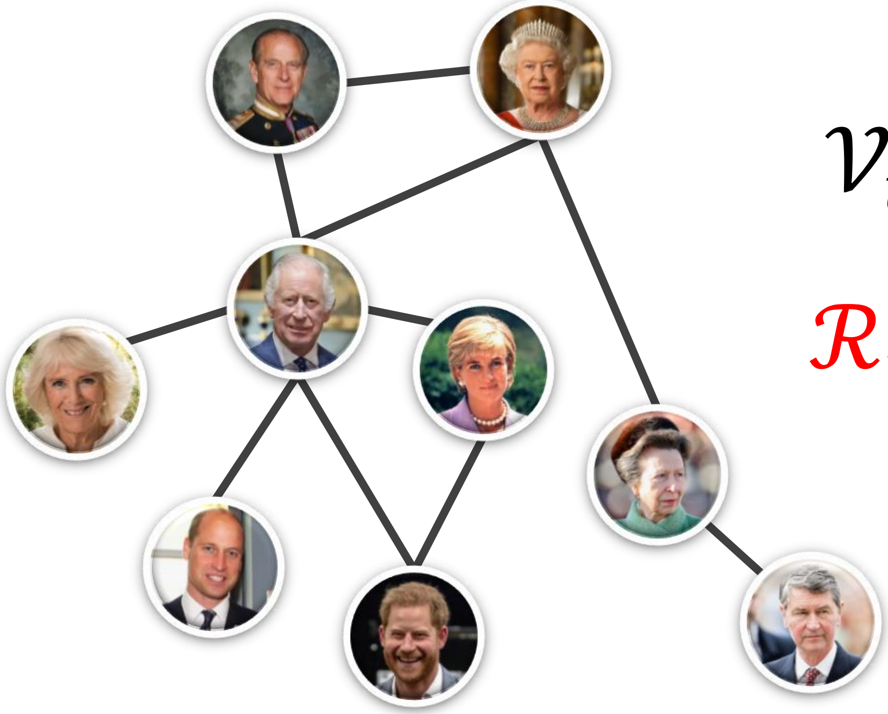
$$\mathcal{V}_{train} \neq \mathcal{V}_{test}$$



$\mathcal{V}$: British royal family
$\mathcal{R}$: {parent, spouse}

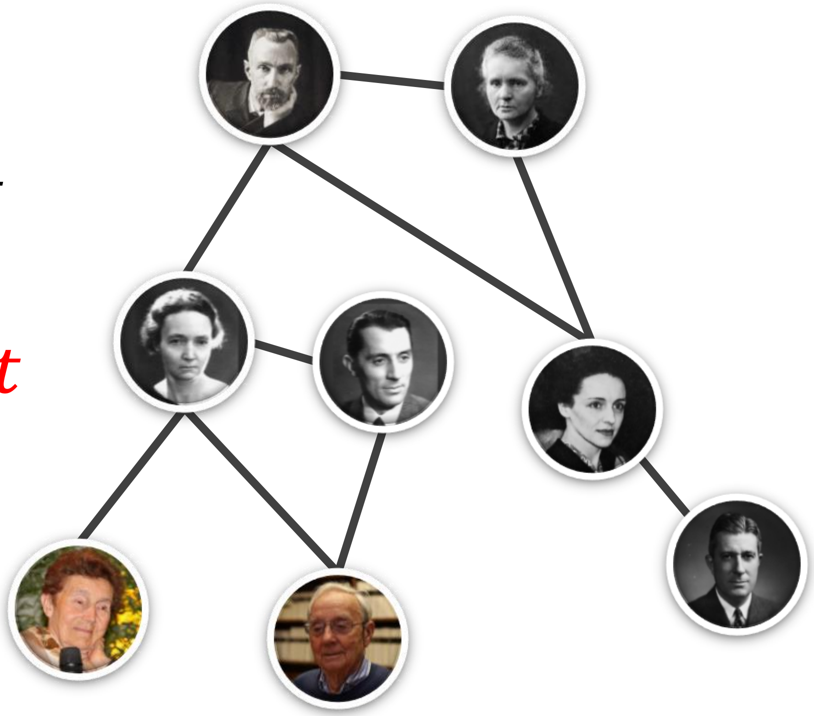
$\mathcal{V}$: Curie family
$\mathcal{R}$: {parent, spouse}

Inductive Generalization on Structure



$$\mathcal{V}_{train} \neq \mathcal{V}_{test}$$

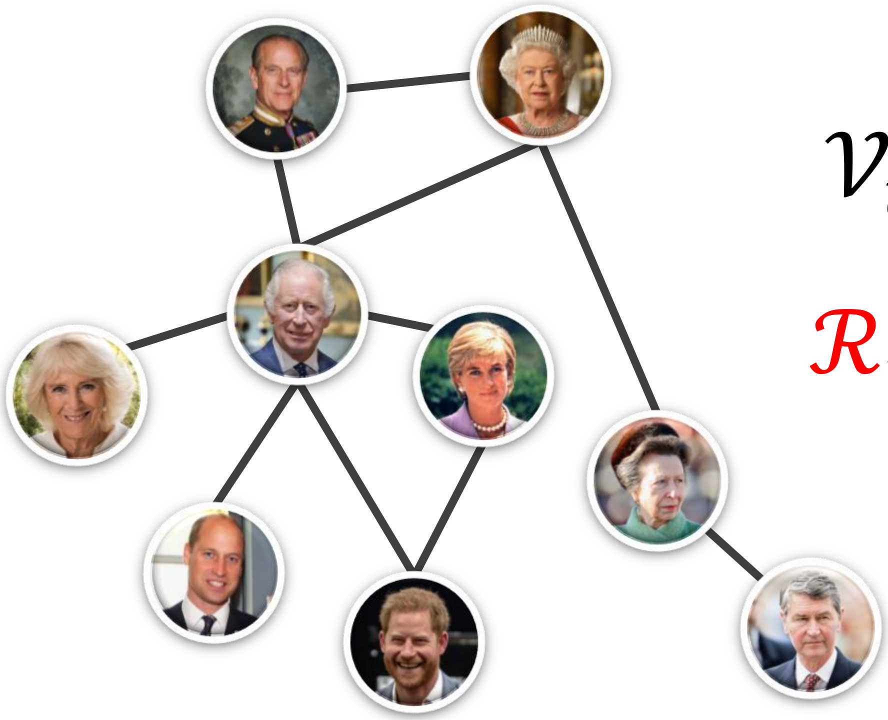
$$\mathcal{R}_{train} = \mathcal{R}_{test}$$



$\mathcal{V}$: British royal family
$\mathcal{R}$: {parent, spouse}

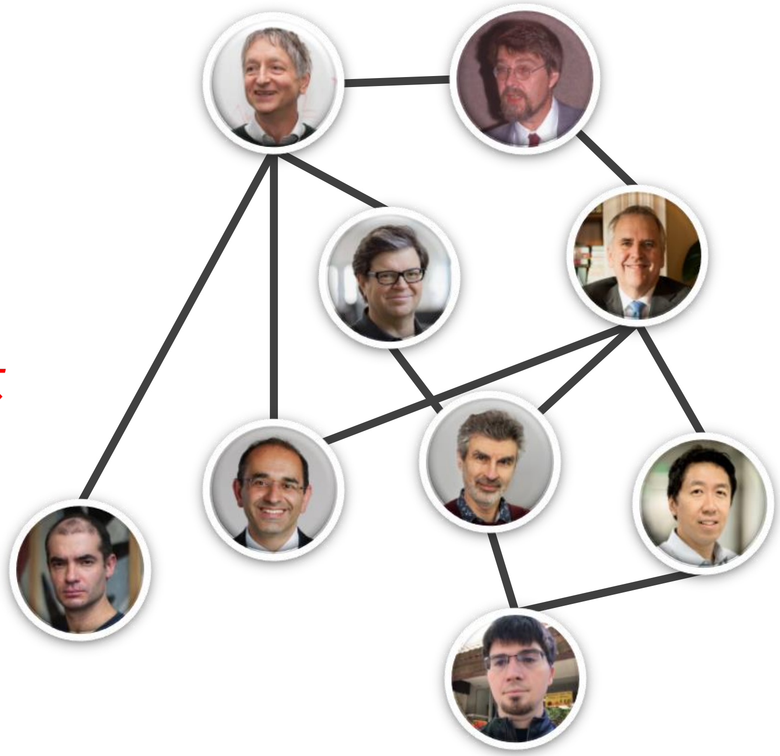
$\mathcal{V}$: Curie family
$\mathcal{R}$: {parent, spouse}

Inductive Generalization on Structure



$$\mathcal{V}_{train} \neq \mathcal{V}_{test}$$

$$\color{red}{\mathcal{R}_{train} \neq \mathcal{R}_{test}}$$



$\mathcal{V}$: British royal family
$\mathcal{R}$: {parent, spouse}

$\mathcal{V}$: deep learning researchers
$\mathcal{R}$: {supervisor, collaborator}

What Generalizes for Entities?

$\mathcal{V}$: British royal family
$\mathcal{R}$: {parent, spouse}

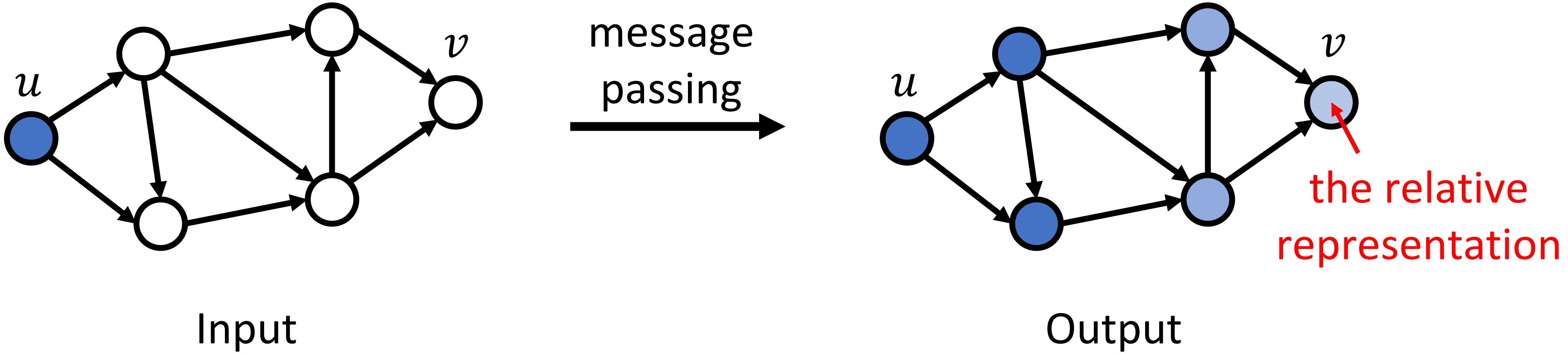
$\mathcal{V}$: Curie family
$\mathcal{R}$: {parent, spouse}

Elizabeth II $\not\longrightarrow$ Marie Curie

Elizabeth II - Princess Anne $\longrightarrow$ Marie Curie - Irene Curie

Relative Entity Representations

encode $v - u$ on graph $\mathcal{G}$



What Generalizes for Relations?

$\mathcal{V}$: British royal family
$\mathcal{R}$: {parent, spouse}

$\mathcal{V}$: deep learning researchers
$\mathcal{R}$: {supervisor, collaborator}

parent $\nrightarrow$ supervisor

parent - spouse $\rightarrow$ supervisor - collaborator

Relative Relation Representations

relative entity: encode $v - u$ on graph $\mathcal{G}$

relative relation: encode $r - q$ on what?

Relative Relation Representations

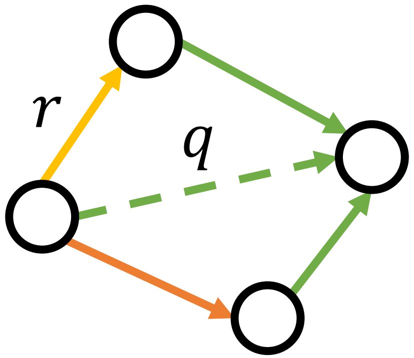
relative entity: encode $v - u$ on graph $\mathcal{G}$

relative relation: encode $r - q$ on what?

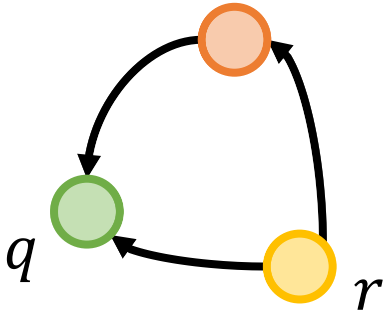
Construct a relation graph to
capture relation interactions!

Relative Relation Representations

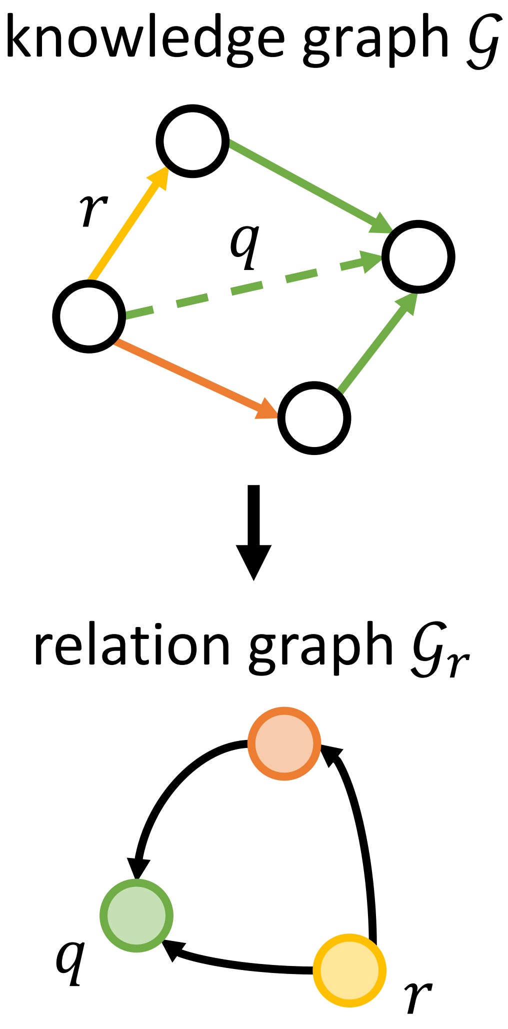
knowledge graph $\mathcal{G}$



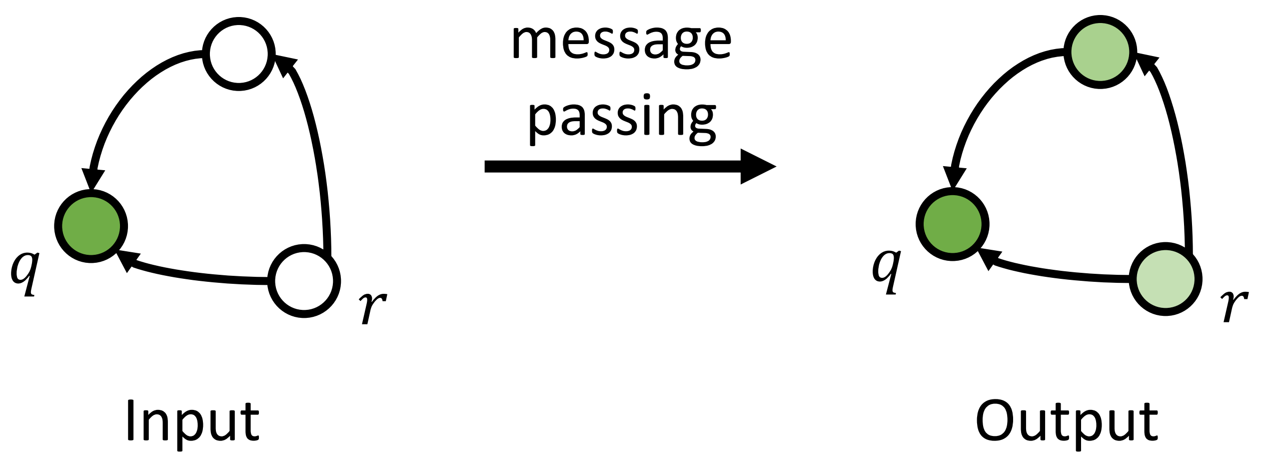
relation graph $\mathcal{G}_r$



Relative Relation Representations



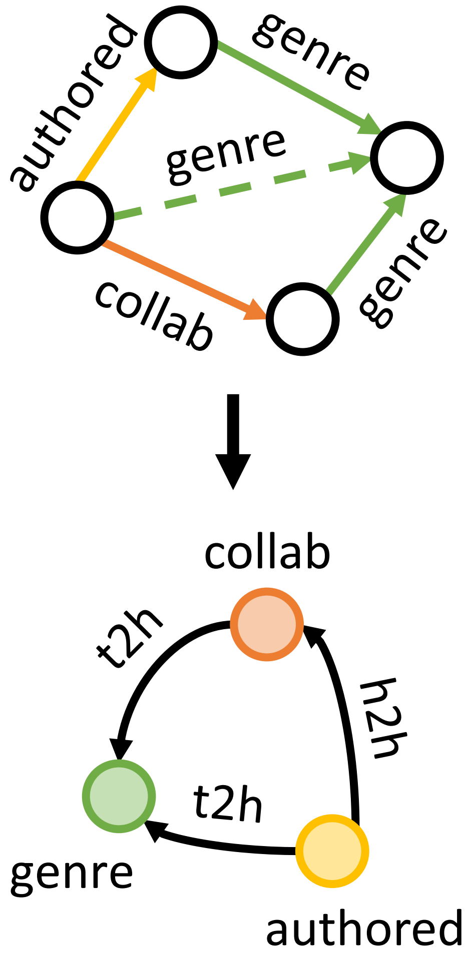
Encode $r - q$ on $\mathcal{G}_r$



Relation Graph

Relation interactions:

head2head, head2tail, tail2head, tail2tail

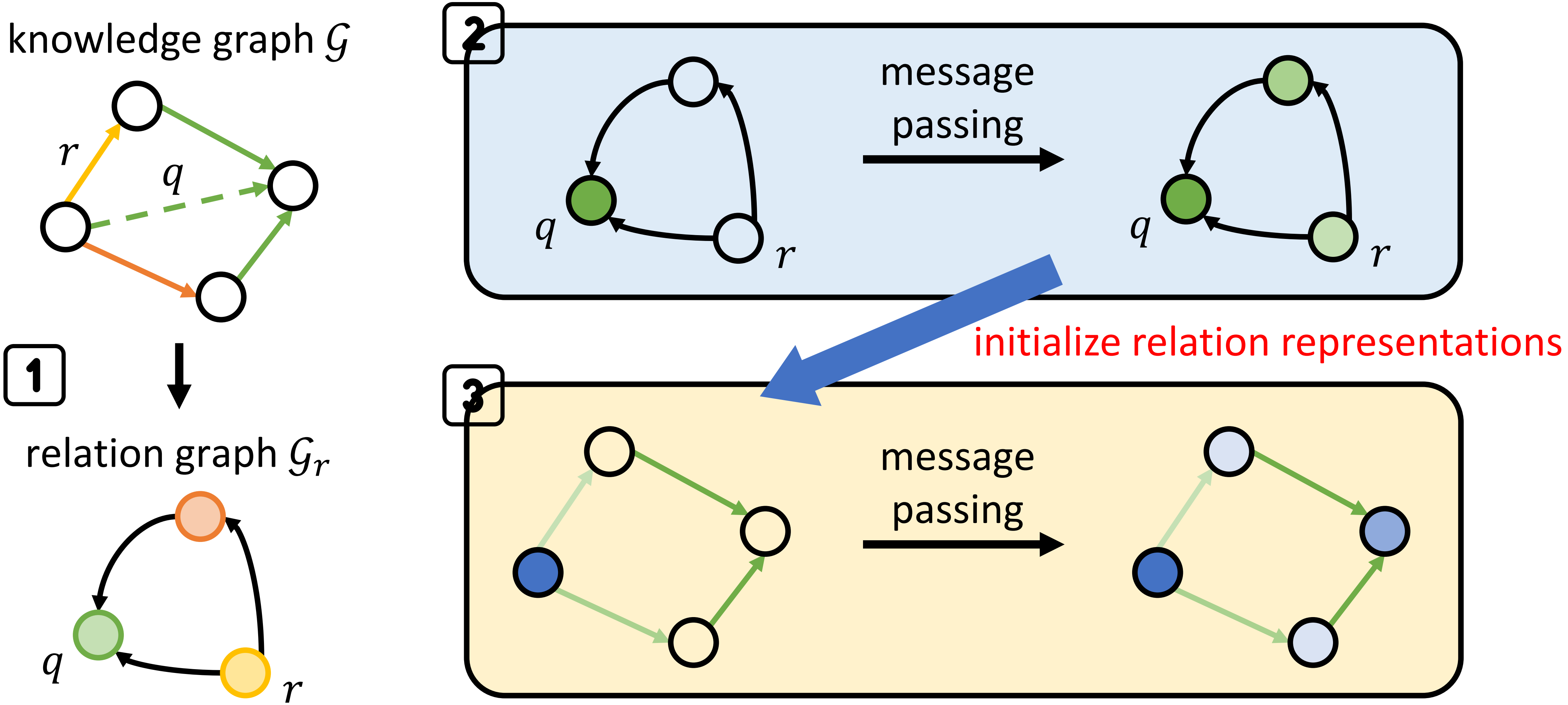


Example:

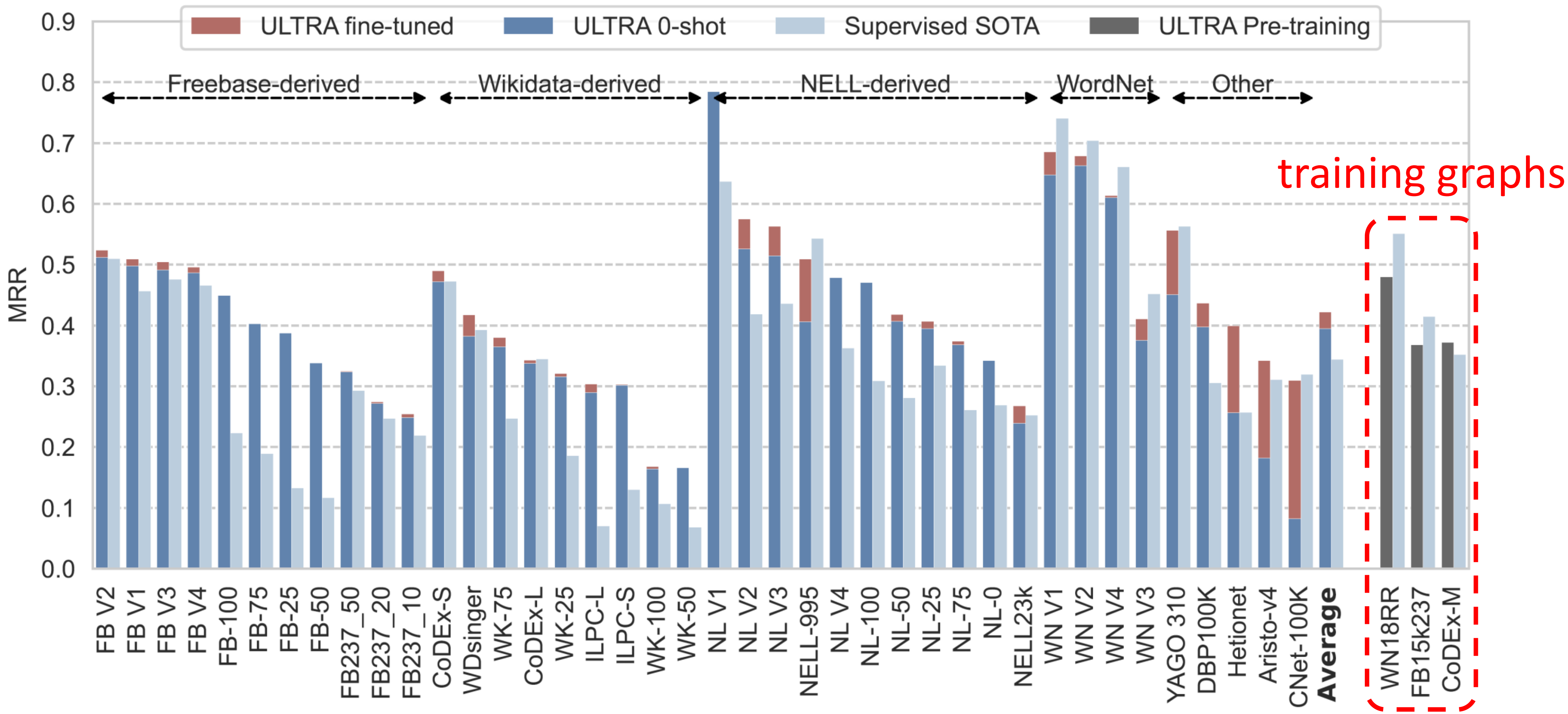
(author, t2h, genre)

Anything that has an author is likely to have a genre

Ultra: Unified, Learnable, Transferable



0-shot Inference on any Knowledge Graph



Surprising Generalization Ability

Domains

Ontology
Encyclopedia

→

Biology Ontology
Encyclopedia
Commonsense Geography

Sizes

10-40k entities
10-200 relations
100k-300k triplets

→

2k-100k entities
7-2000 relations
5k-1M triplets

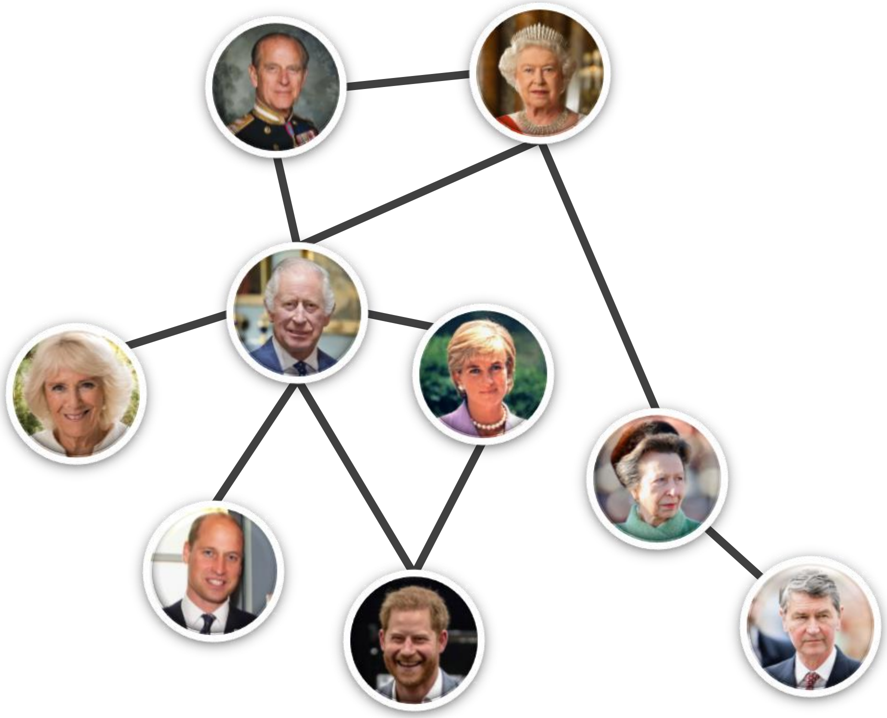
GNN-QE[1][2][3]: Solving **multi-hop queries** with **inductive models and logical operations**

[1] **Zhaocheng Zhu**, Mikhail Galkin, Zuobai Zhang, Jian Tang. Neural-Symbolic Models for Logical Queries on Knowledge Graphs. ICML 2022.
[2] Mikhail Galkin, **Zhaocheng Zhu**, Hongyu Ren, Jian Tang. Inductive Logical Query Answering in Knowledge Graphs. NeurIPS 2022.
[3] Mikhail Galkin, Jincheng Zhou, Bruno Ribeiro, Jian Tang, **Zhaocheng Zhu**. Zero-shot Logical Query Reasoning on any Knowledge Graph. arXiv 2024.

Knowledge Graph Completion

Input: a head entity, a relation

Output: one or many tail entities

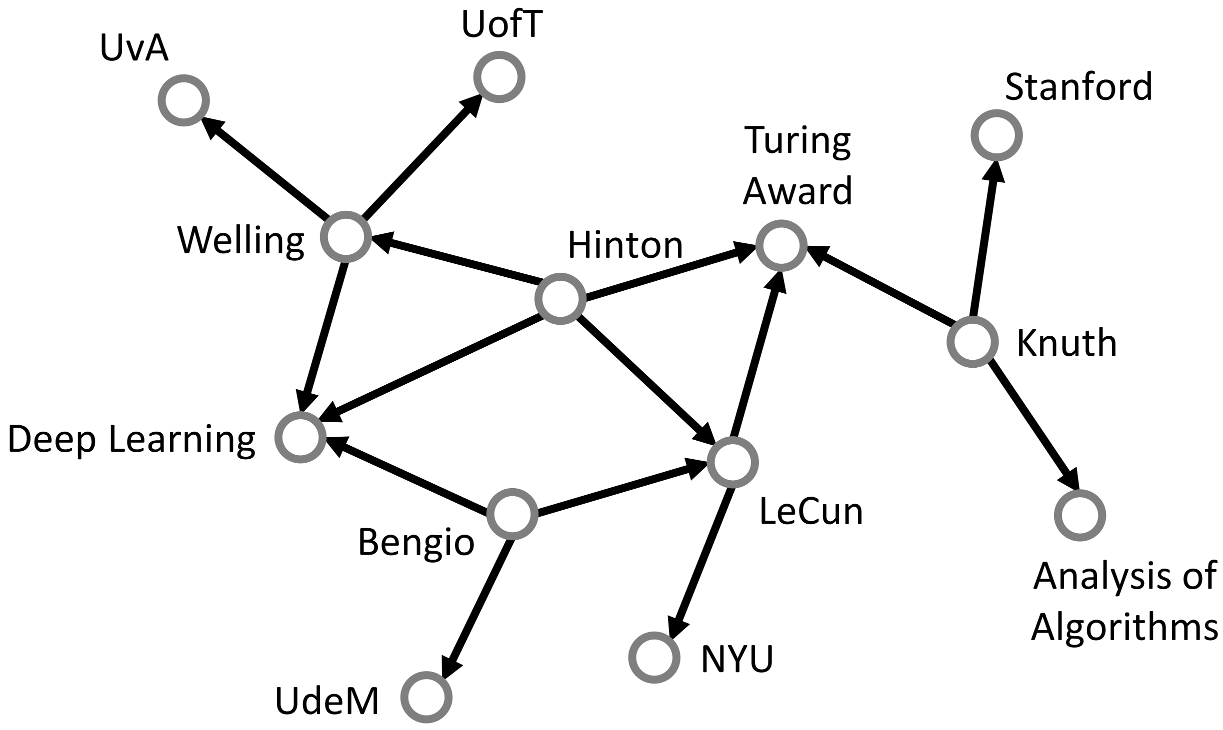


( , grandson, ?)

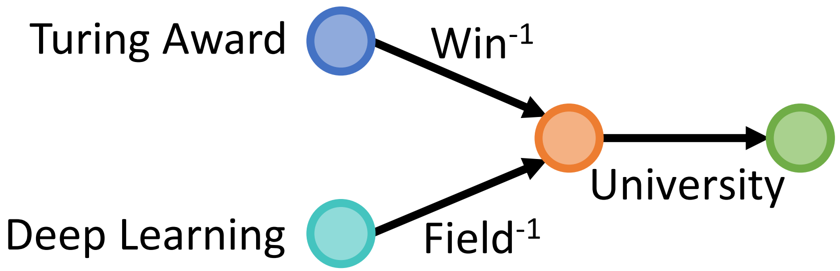
Multi-hop Logical Queries

Input: one or several entities, several relations, logical operations

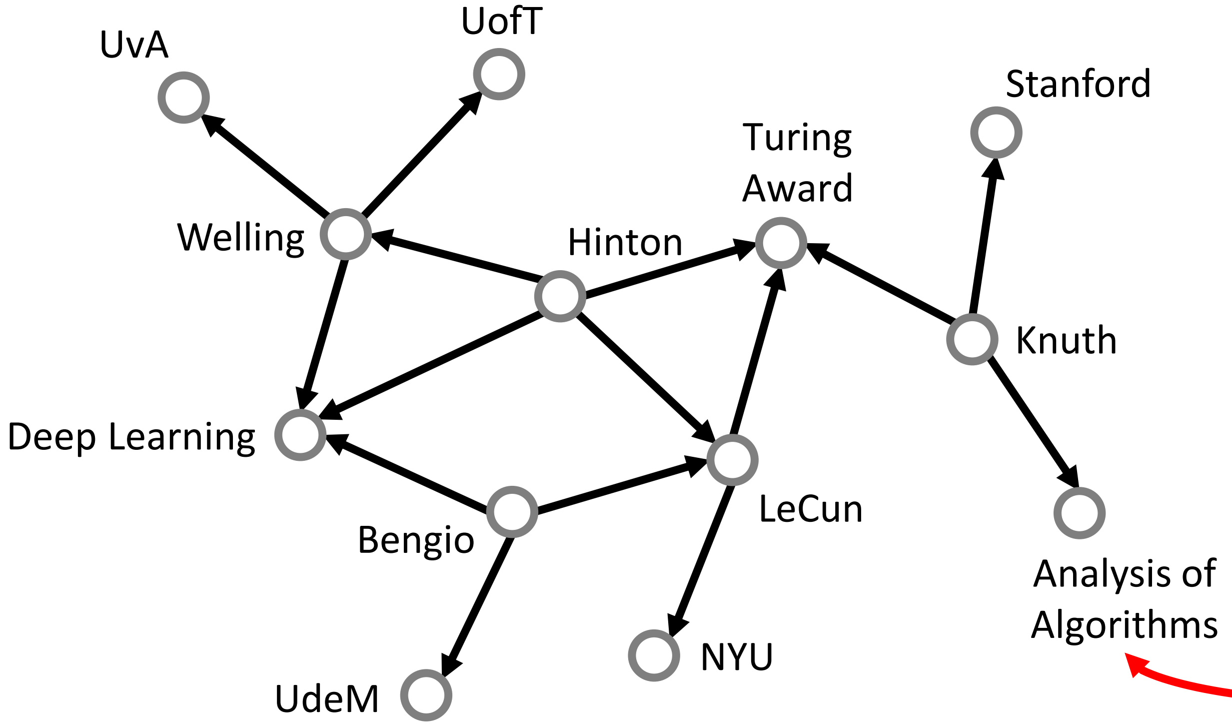
Output: one or many tail entities



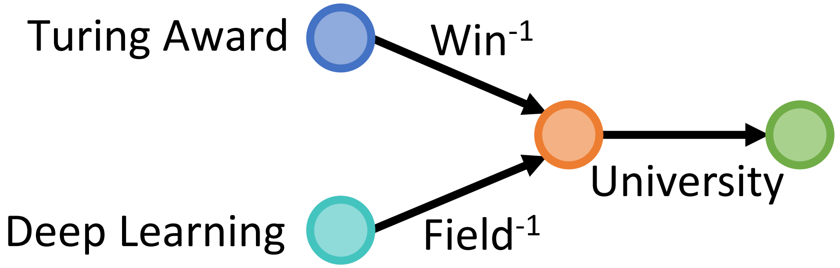
At what universities do the Turing Award winners in the field of deep learning work?



Multi-hop Logical Queries

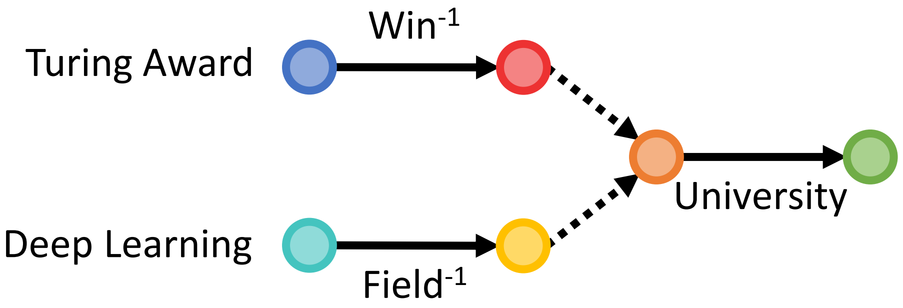


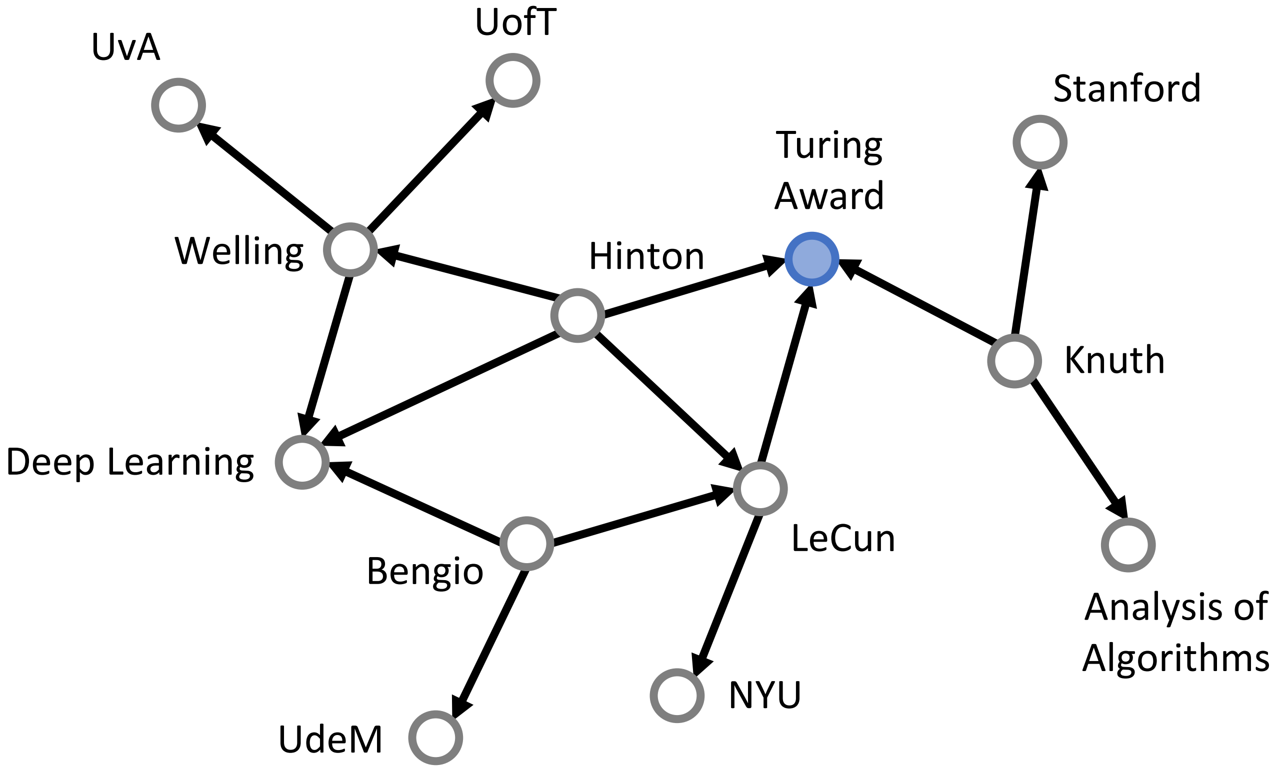
At what universities do the Turing Award winners in the field of deep learning work?



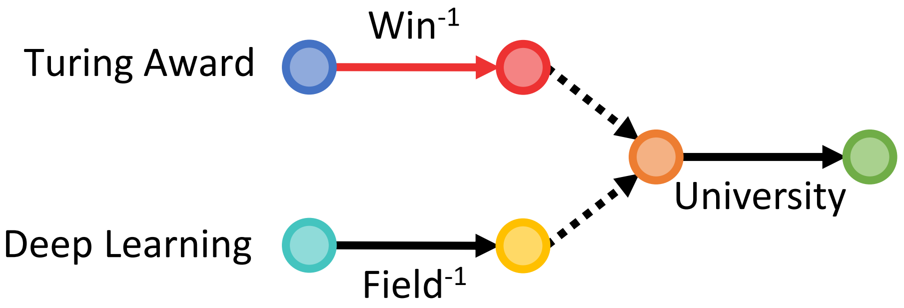
search this subgraph

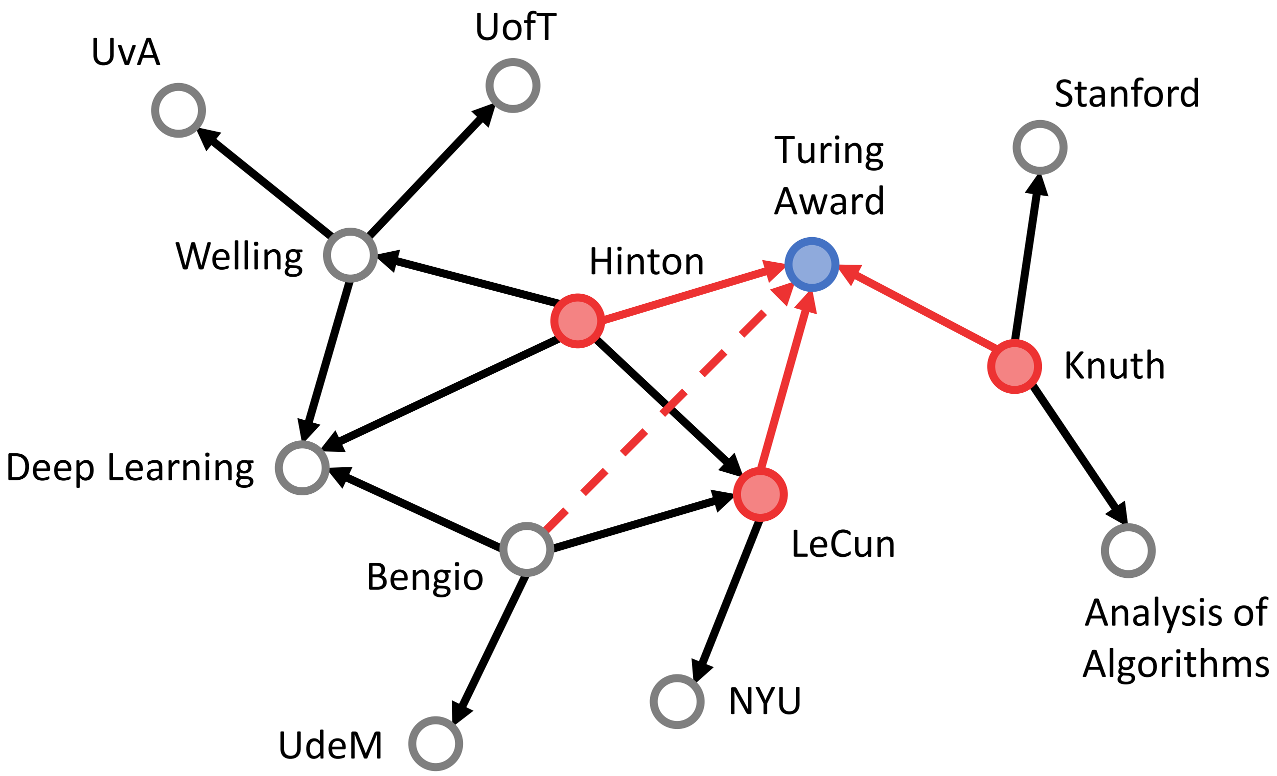
Subgraph Matching



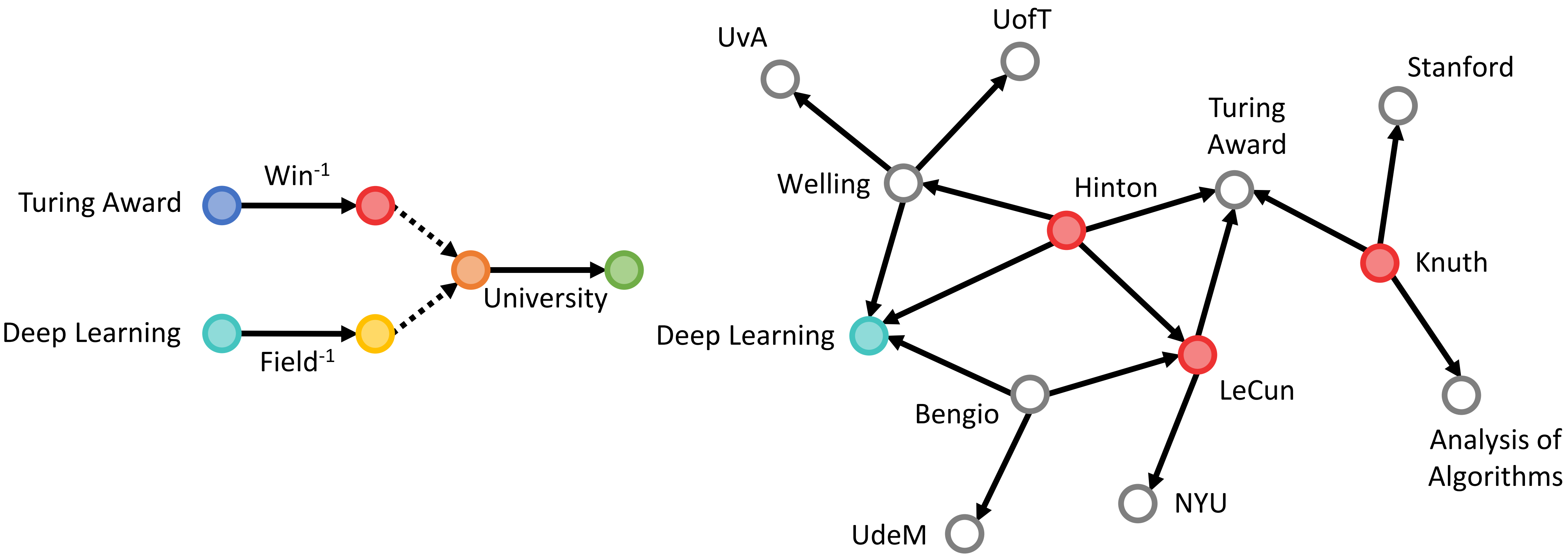


Subgraph Matching

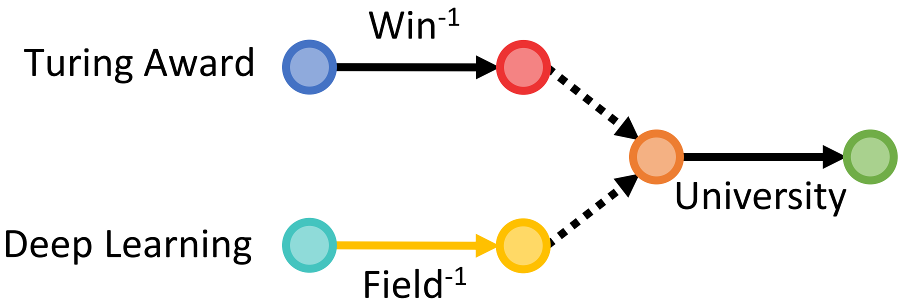


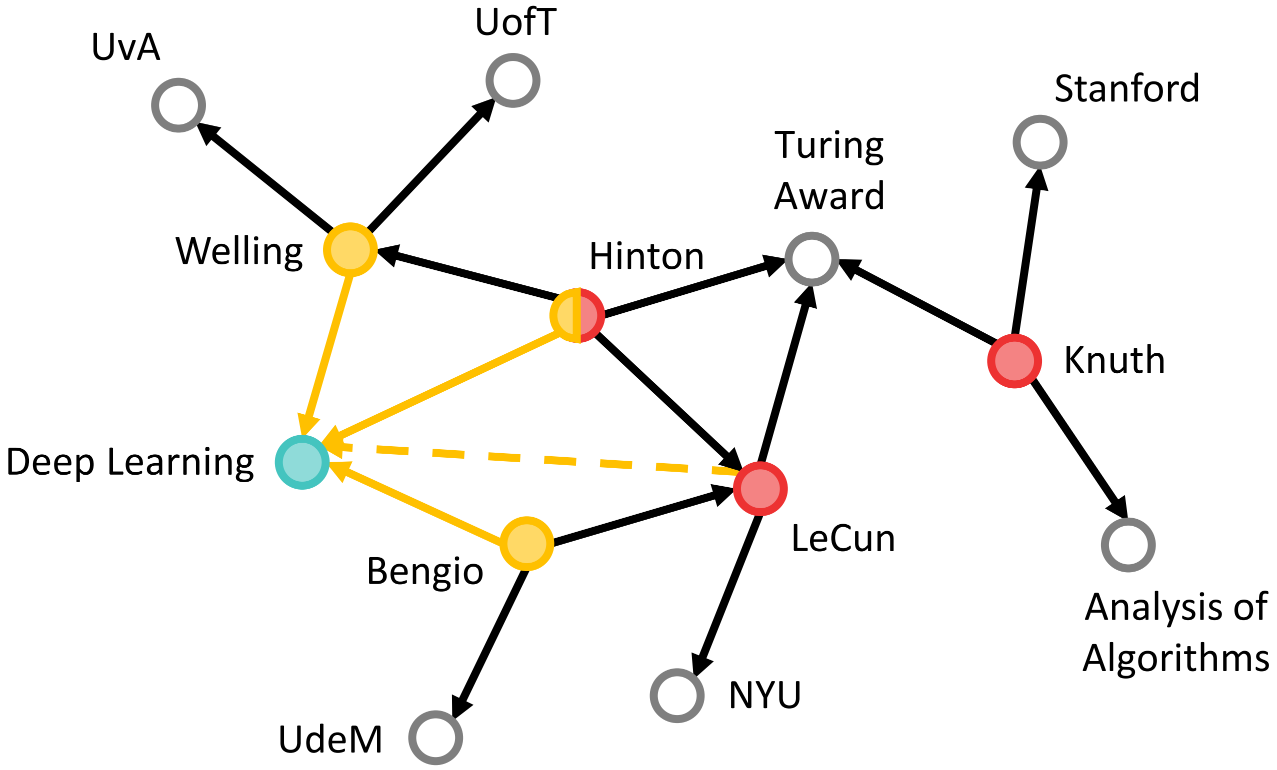


Subgraph Matching

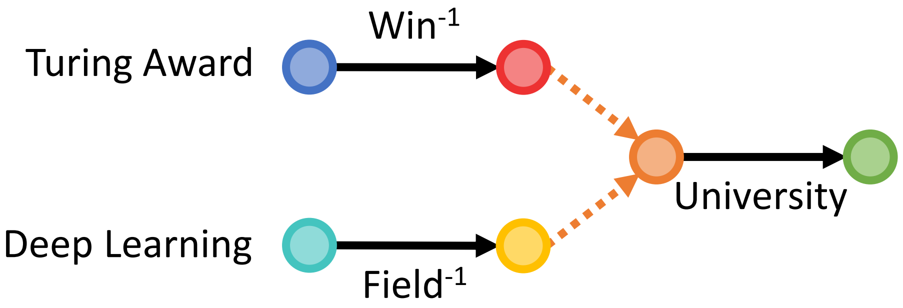


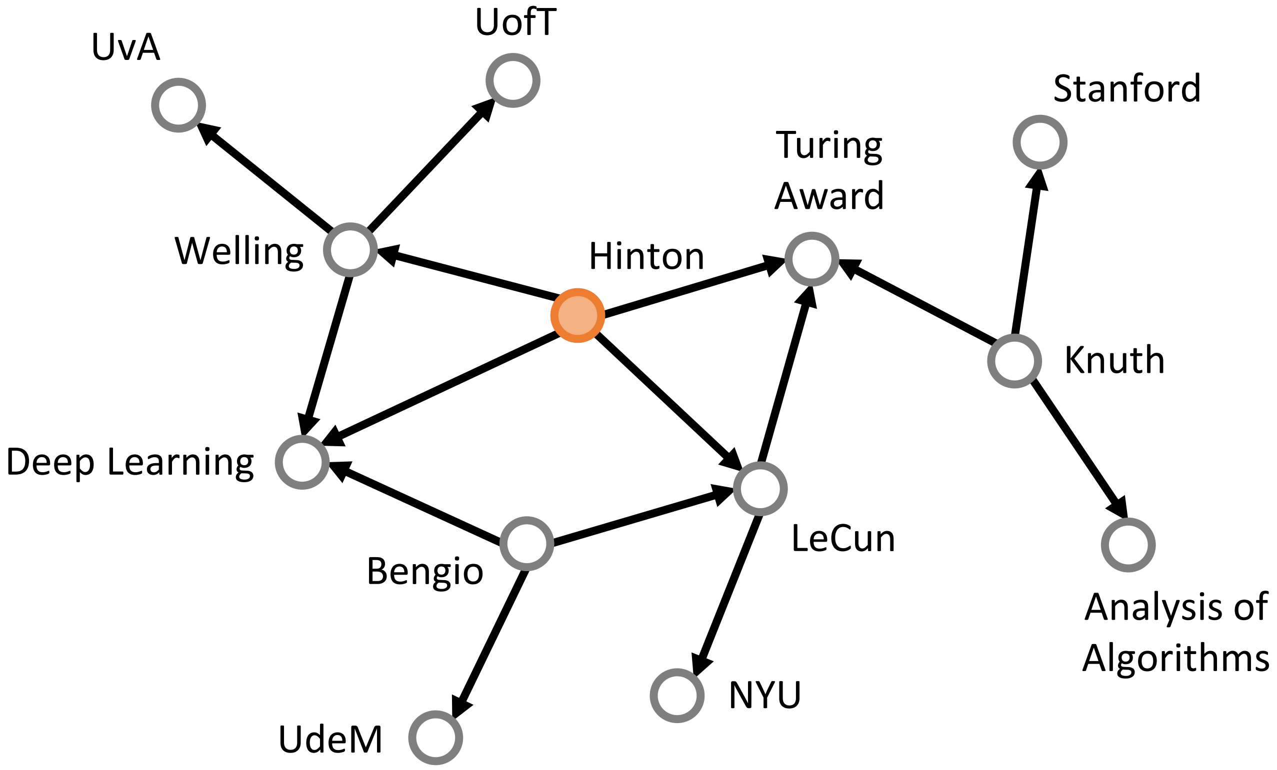
Subgraph Matching



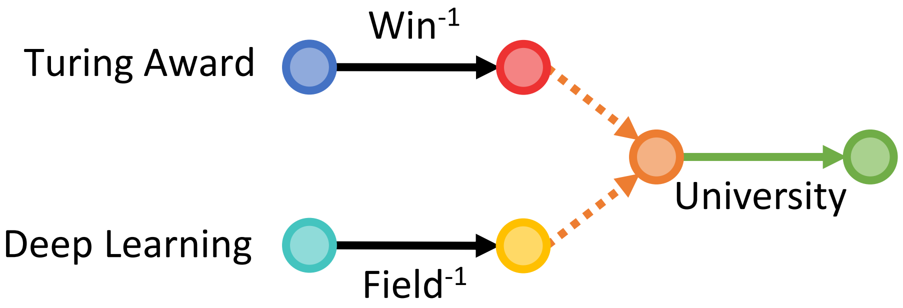


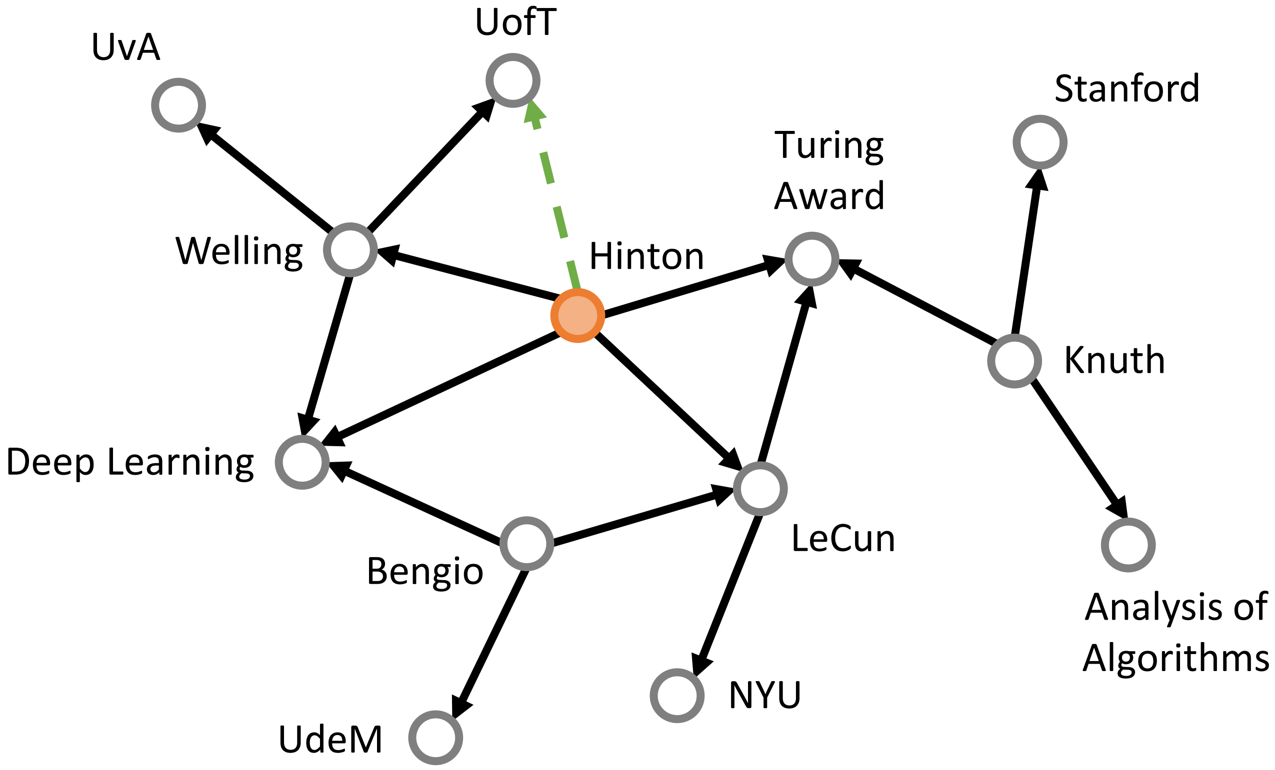
Subgraph Matching



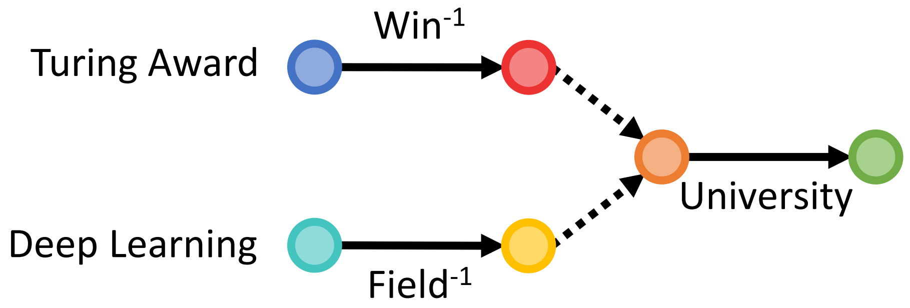


Subgraph Matching

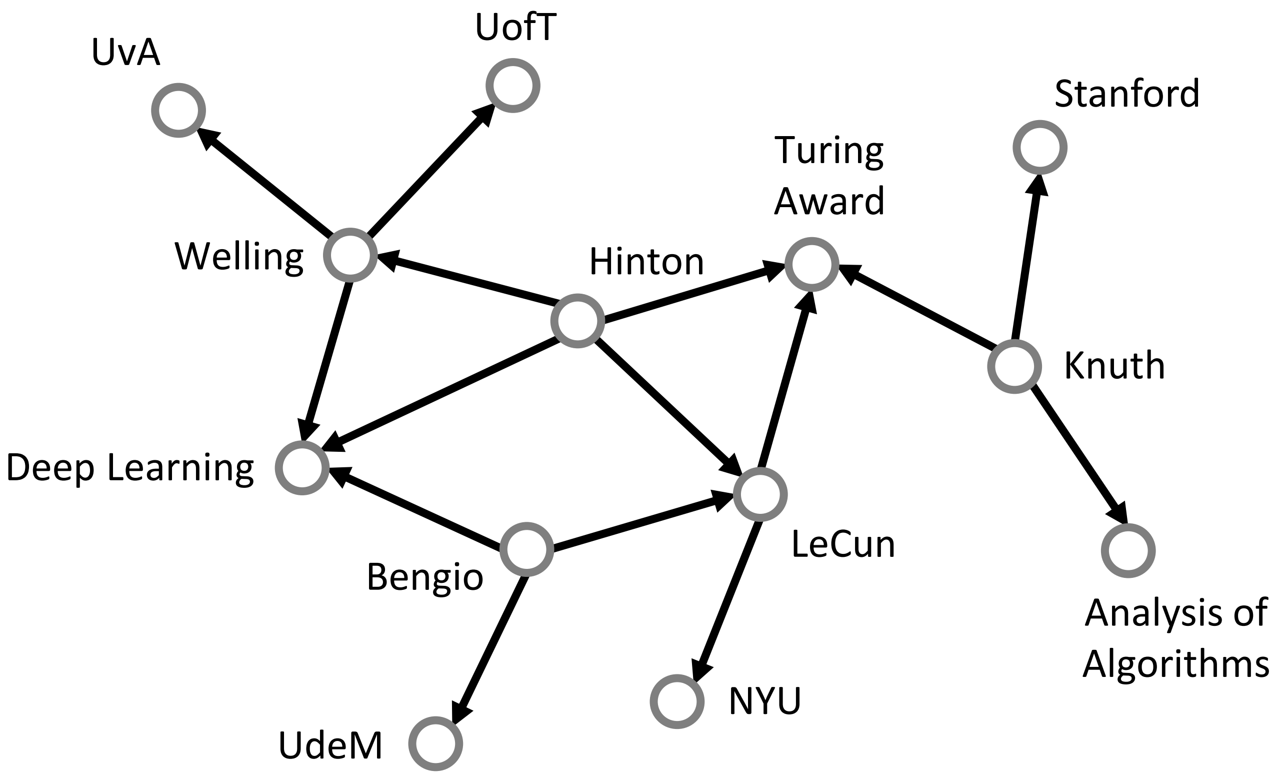




Subgraph Matching



No answer!



Subgraph matching is **inductive**, but it **can't reason about missing links**.

Subgraph Matching

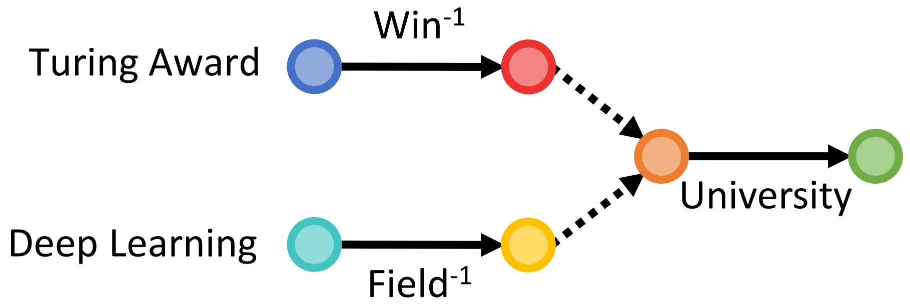
$\mathcal{X} = \{\text{Hinton}, \text{Lecun}, \text{Bengio}\} \in 2^{\mathcal{V}}$

Relation Projection: $\mathcal{Y} = University(\mathcal{X})$

Conjunction: $\mathcal{X} \cap \mathcal{Y}$

Disjunction: $\mathcal{X} \cup \mathcal{Y}$

Negation: $\mathcal{V} \backslash \mathcal{X}$



Relax to Fuzzy Sets

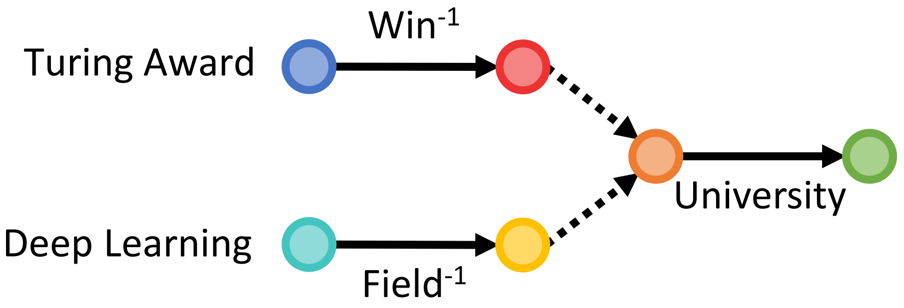
$\boldsymbol{x} = \{\text{Hinton: } 0.81, \text{Lecun: } 0.56, \text{Bengio: } 0.64\} \in [0,1]^{\mathcal{V}}$

Relation Projection: $\boldsymbol{y} = University(\boldsymbol{x})$

Conjunction: $\boldsymbol{x} \odot \boldsymbol{y}$

Disjunction: $\boldsymbol{x} + \boldsymbol{y} - \boldsymbol{x} \odot \boldsymbol{y}$

Negation: $1 - \boldsymbol{x}$

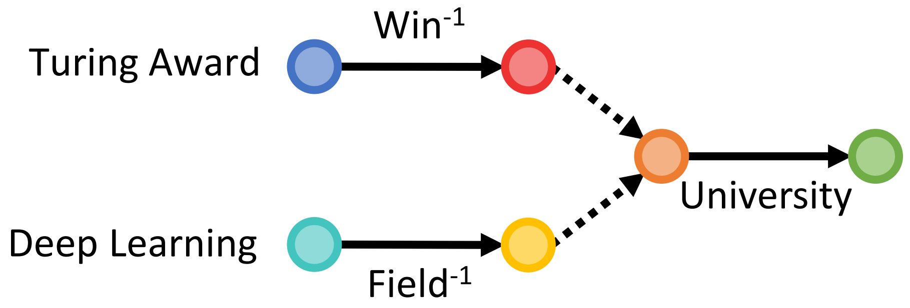


Relax to Fuzzy Sets

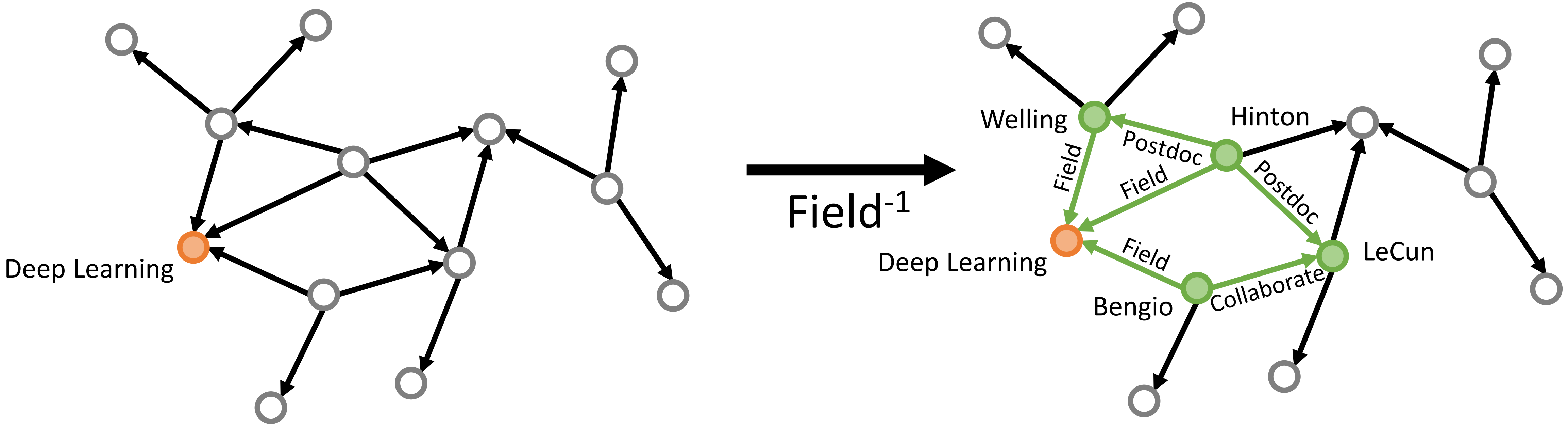
$\boldsymbol{x} = \{\text{Hinton: } 0.81, \text{Lecun: } 0.56, \text{Bengio: } 0.64\} \in [0,1]^{\mathcal{V}}$

Relation Projection: $\boldsymbol{y} = University(\boldsymbol{x})$

$$\left.\begin{array}{l}\textbf{Conjunction: } \boldsymbol{x} \odot \boldsymbol{y} \\ \textbf{Disjunction: } \boldsymbol{x} + \boldsymbol{y} - \boldsymbol{x} \odot \boldsymbol{y} \\ \textbf{Negation: } \mathbf{1} - \boldsymbol{x}\end{array}\right\} \text{Inductive!}$$

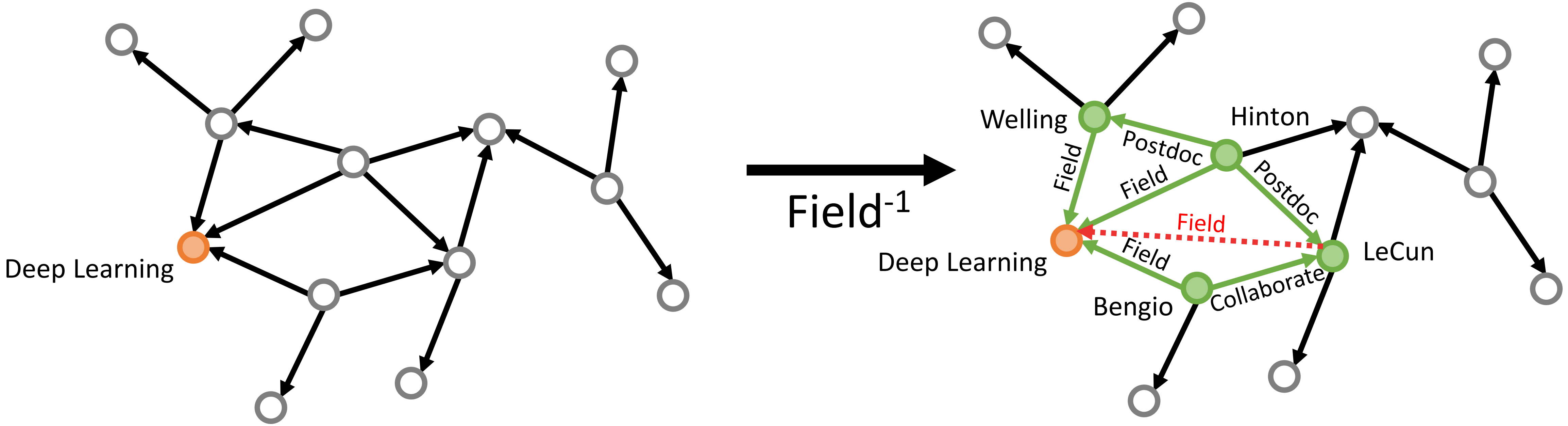


Refresher: NBFNet



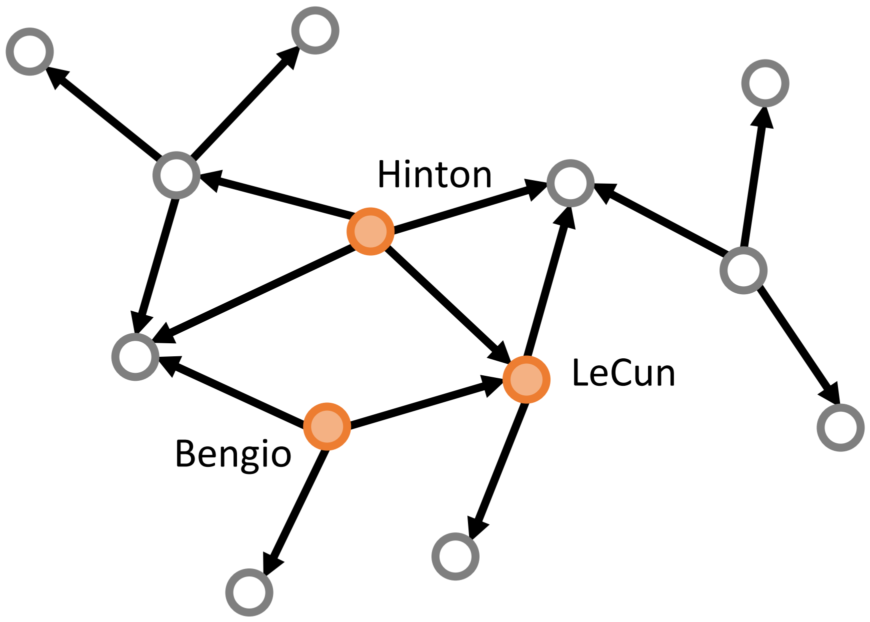
a **single-source** input

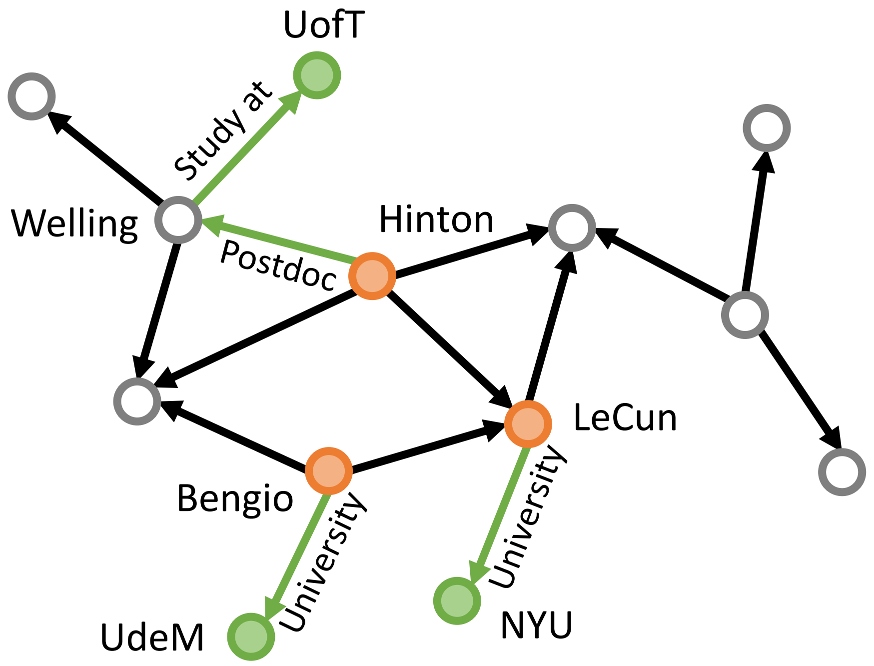
Refresher: NBFNet



a **single-source** input

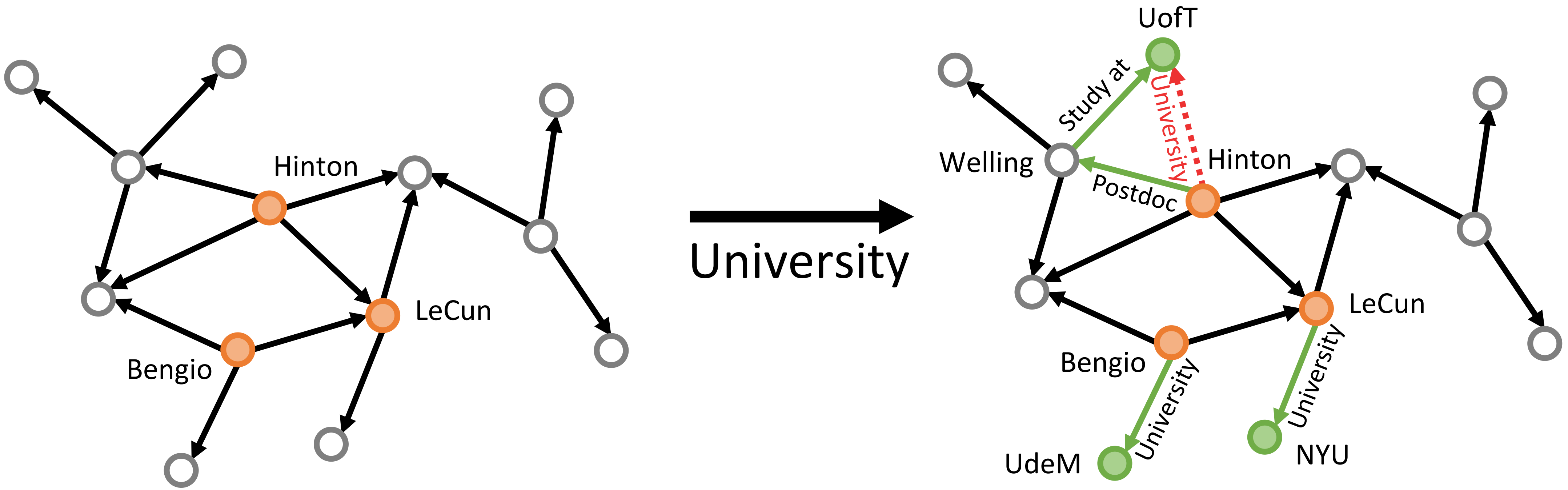
Relation Projection





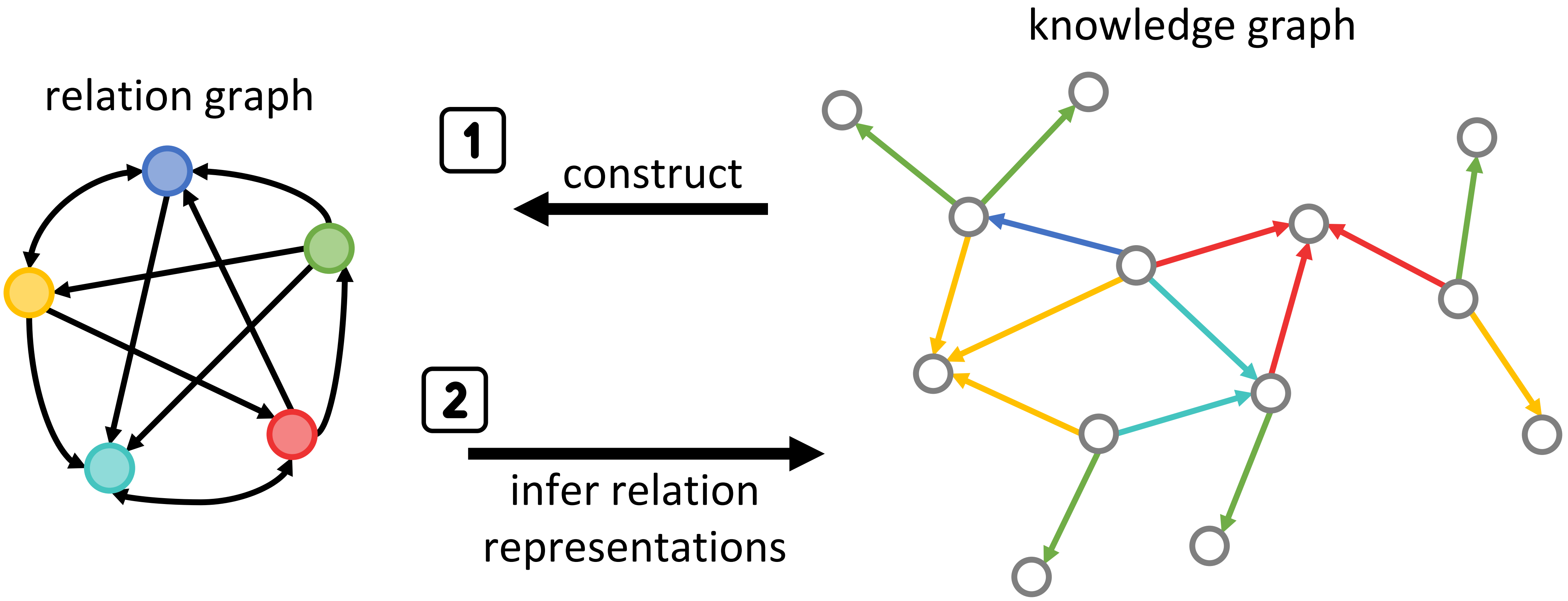
a fuzzy set input

Relation Projection

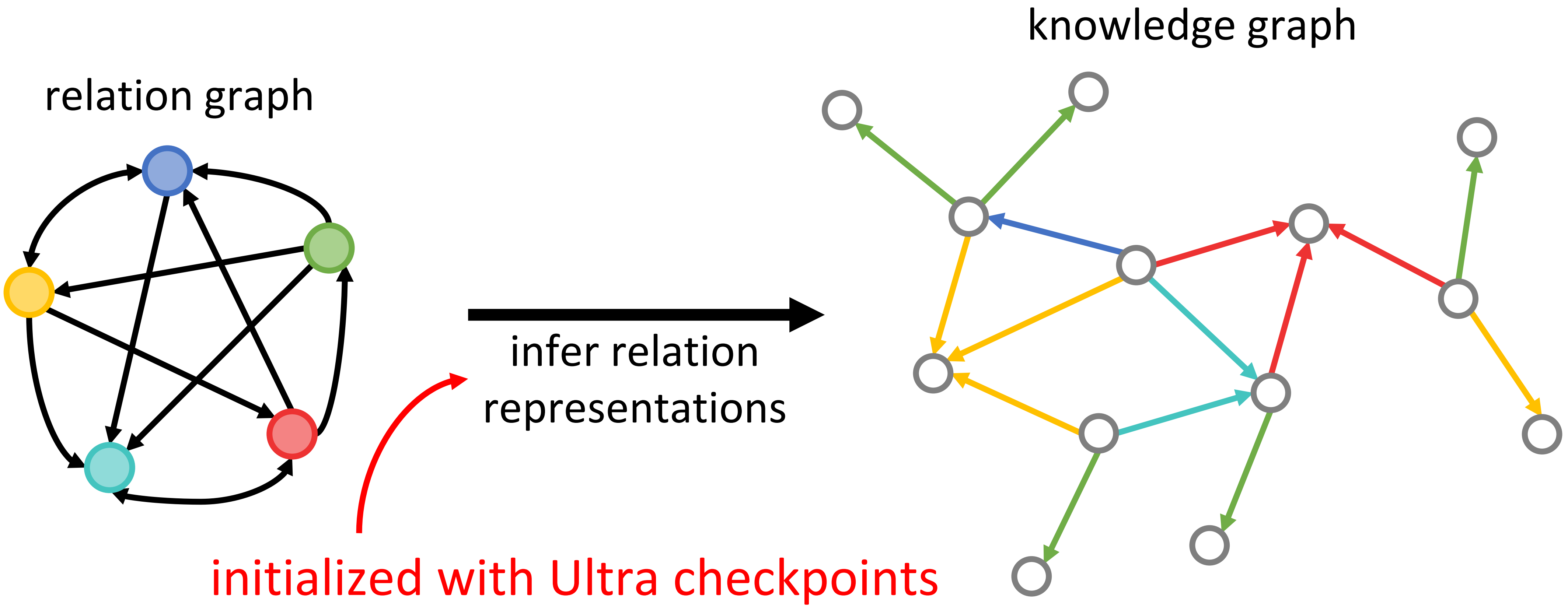


a **fuzzy set** input

0-shot Relation Projection



0-shot Relation Projection



Graph Neural Network Query Executor

$\boldsymbol{x} = \{\text{Hinton: } 0.81, \text{Lecun: } 0.56, \text{Bengio: } 0.64\} \in [0,1]^{\mathcal{V}}$

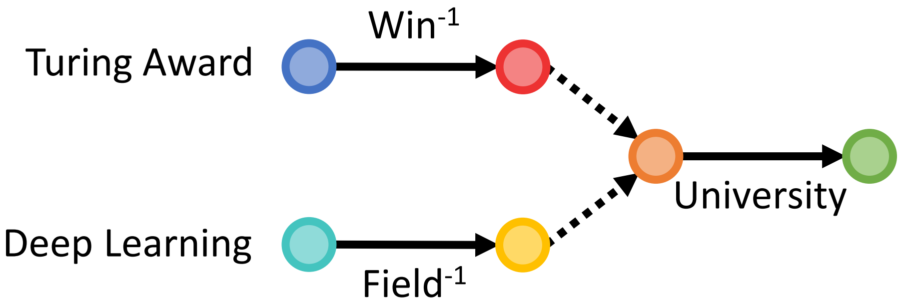
Relation Projection: $\boldsymbol{y} = University(\boldsymbol{x})$ Inductive!

Conjunction: $\boldsymbol{x} \odot \boldsymbol{y}$

Disjunction: $\boldsymbol{x} + \boldsymbol{y} - \boldsymbol{x} \odot \boldsymbol{y}$

Negation: $\mathbf{1} - \boldsymbol{x}$

Inductive!



Multi-hop Logical Queries ($\mathcal{V}_{train} = \mathcal{V}_{test}$)

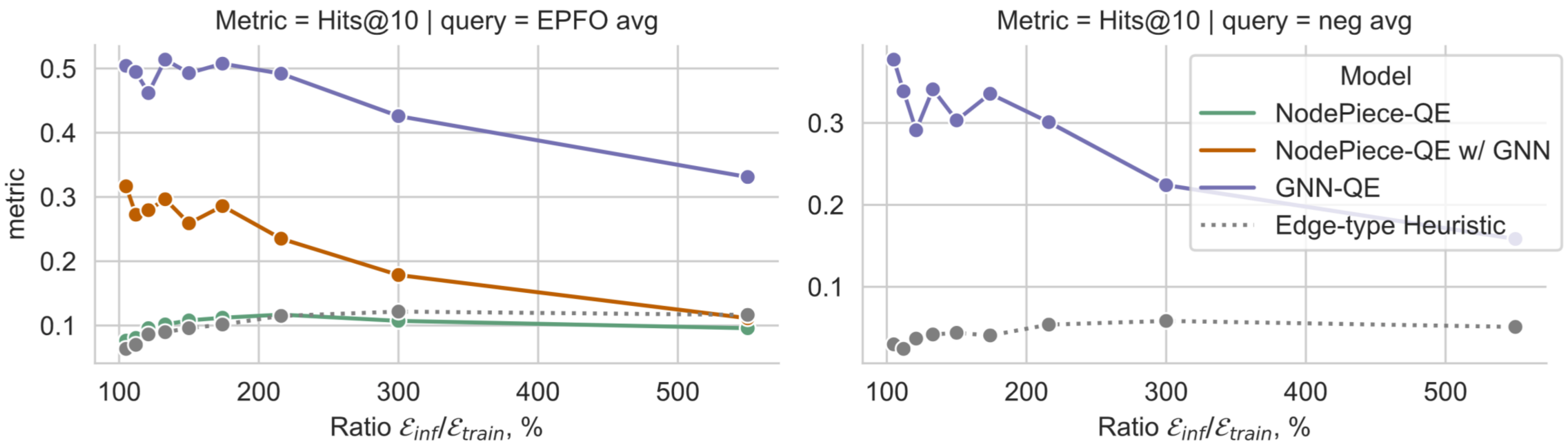
metric: MRR↑

Model	avg_p	avg_n	1p	2p	3p	2i	3i	pi	ip	2u	up	2in	3in	inp	pin	pni
								FB15k								
GQE	28.0	-	54.6	15.3	10.8	39.7	51.4	27.6	19.1	22.1	11.6	-	-	-	-	-
Q2B	38.0	-	68.0	21.0	14.2	55.1	66.5	39.4	26.1	35.1	16.7	-	-	-	-	-
BetaE	41.6	11.8	65.1	25.7	24.7	55.8	66.5	43.9	28.1	40.1	25.2	14.3	14.7	11.5	6.5	12.4
CQD-CO	46.9	-	**89.2**	25.3	13.4	74.4	78.3	44.1	33.2	41.8	21.9	-	-	-	-	-
CQD-Beam	58.2	-	**89.2**	54.3	28.6	74.4	78.3	58.2	67.7	42.4	30.9	-	-	-	-	-
ConE	49.8	14.8	73.3	33.8	29.2	64.4	73.7	50.9	35.7	55.7	31.4	17.9	18.7	12.5	9.8	15.1
GNN-QE	**72.8**	**38.6**	88.5	**69.3**	**58.7**	**79.7**	**83.5**	**69.9**	**70.4**	**74.1**	**61.0**	**44.7**	**41.7**	**42.0**	**30.1**	**34.3**
								FB15k-237								
GQE	16.3	-	35.0	7.2	5.3	23.3	34.6	16.5	10.7	8.2	5.7	-	-	-	-	-
Q2B	20.1	-	40.6	9.4	6.8	29.5	42.3	21.2	12.6	11.3	7.6	-	-	-	-	-
BetaE	20.9	5.5	39.0	10.9	10.0	28.8	42.5	22.4	12.6	12.4	9.7	5.1	7.9	7.4	3.5	3.4
CQD-CO	21.8	-	**46.7**	9.5	6.3	31.2	40.6	23.6	16.0	14.5	8.2	-	-	-	-	-
CQD-Beam	22.3	-	**46.7**	11.6	8.0	31.2	40.6	21.2	18.7	14.6	8.4	-	-	-	-	-
FuzzQE	24.0	7.8	42.8	12.9	10.3	33.3	46.9	26.9	17.8	14.6	10.3	8.5	11.6	7.8	5.2	5.8
ConE	23.4	5.9	41.8	12.8	11.0	32.6	47.3	25.5	14.0	14.5	10.8	5.4	8.6	7.8	4.0	3.6
GNN-QE	**26.8**	**10.2**	42.8	**14.7**	**11.8**	**38.3**	**54.1**	**31.1**	**18.9**	**16.2**	**13.4**	**10.0**	**16.8**	**9.3**	**7.2**	**7.8**

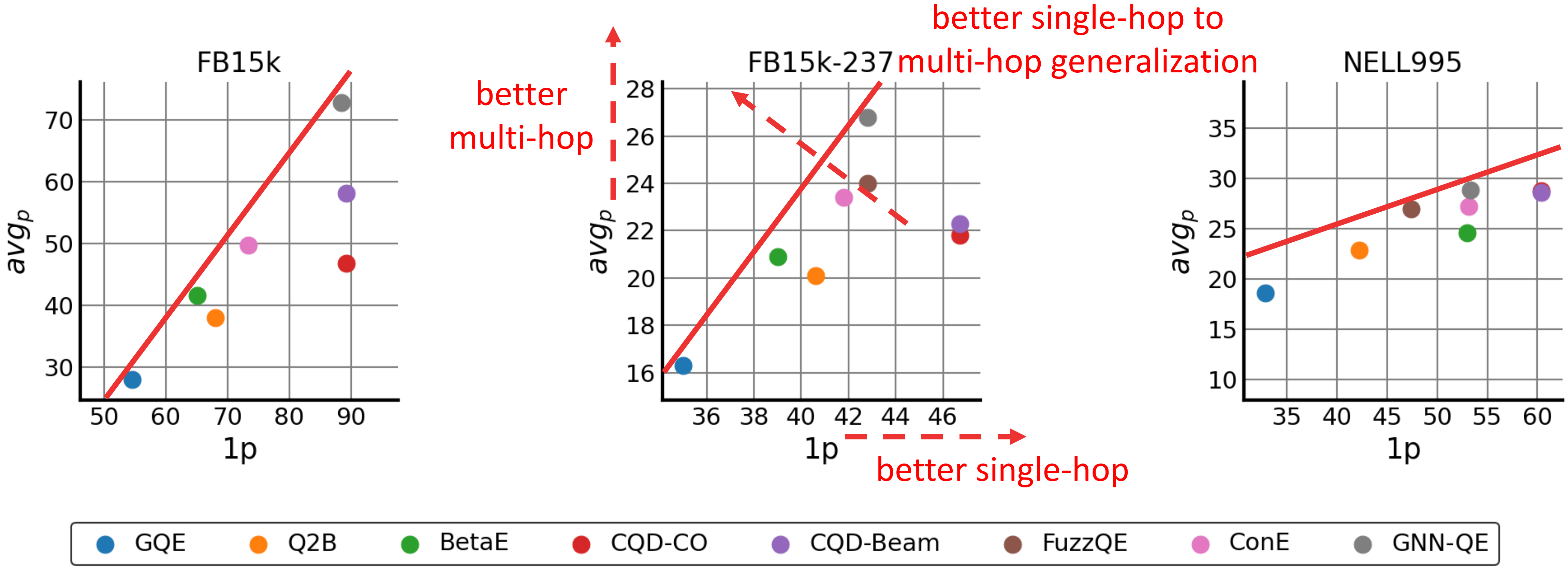
Multi-hop Logical Queries ($\mathcal{V}_{train} \neq \mathcal{V}_{test}$)

metric: H@10↑

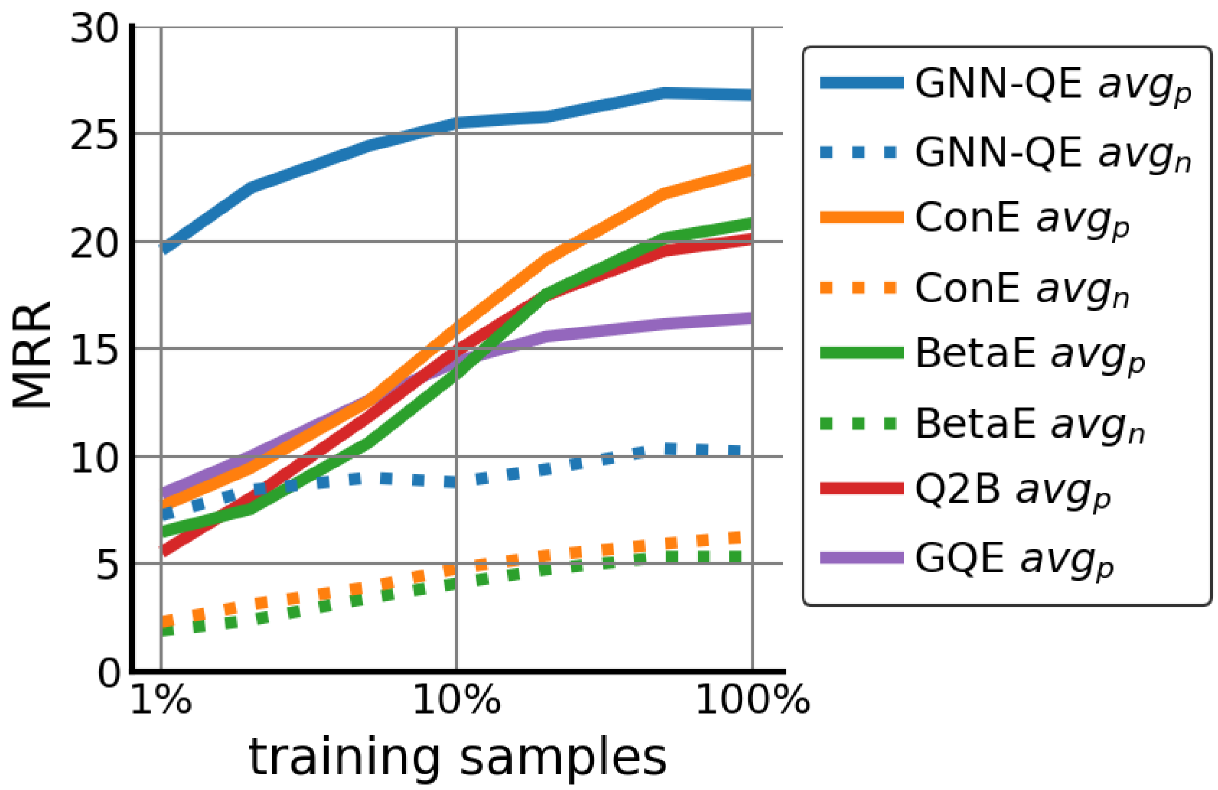
Class	Model	avg_p	avg_n	1p	2p	3p	2i	3i	pi	ip	2u	up	2in	3in	inp	pin	pni
								FB15k-237									
Inference-only	Edge-type Heuristic	10.1	4.1	17.7	8.2	9.9	10.7	13.0	9.8	8.2	5.3	8.5	2.6	2.9	8.4	3.8	2.7
	NodePiece-QE	11.2	-	25.5	8.2	8.4	12.4	13.9	9.9	8.7	7.0	6.8	-	-	-	-	-
	NodePiece-QE w/ GNN	28.6	-	45.9	19.2	11.5	39.9	48.8	29.4	22.6	25.3	14.6	-	-	-	-	-
Trainable	GNN-QE	**50.7**	**33.6**	**65.4**	**36.3**	**31.6**	**73.8**	**84.3**	**56.5**	**41.5**	**39.3**	**28.0**	**33.3**	**46.4**	**29.2**	**24.9**	**34.0**



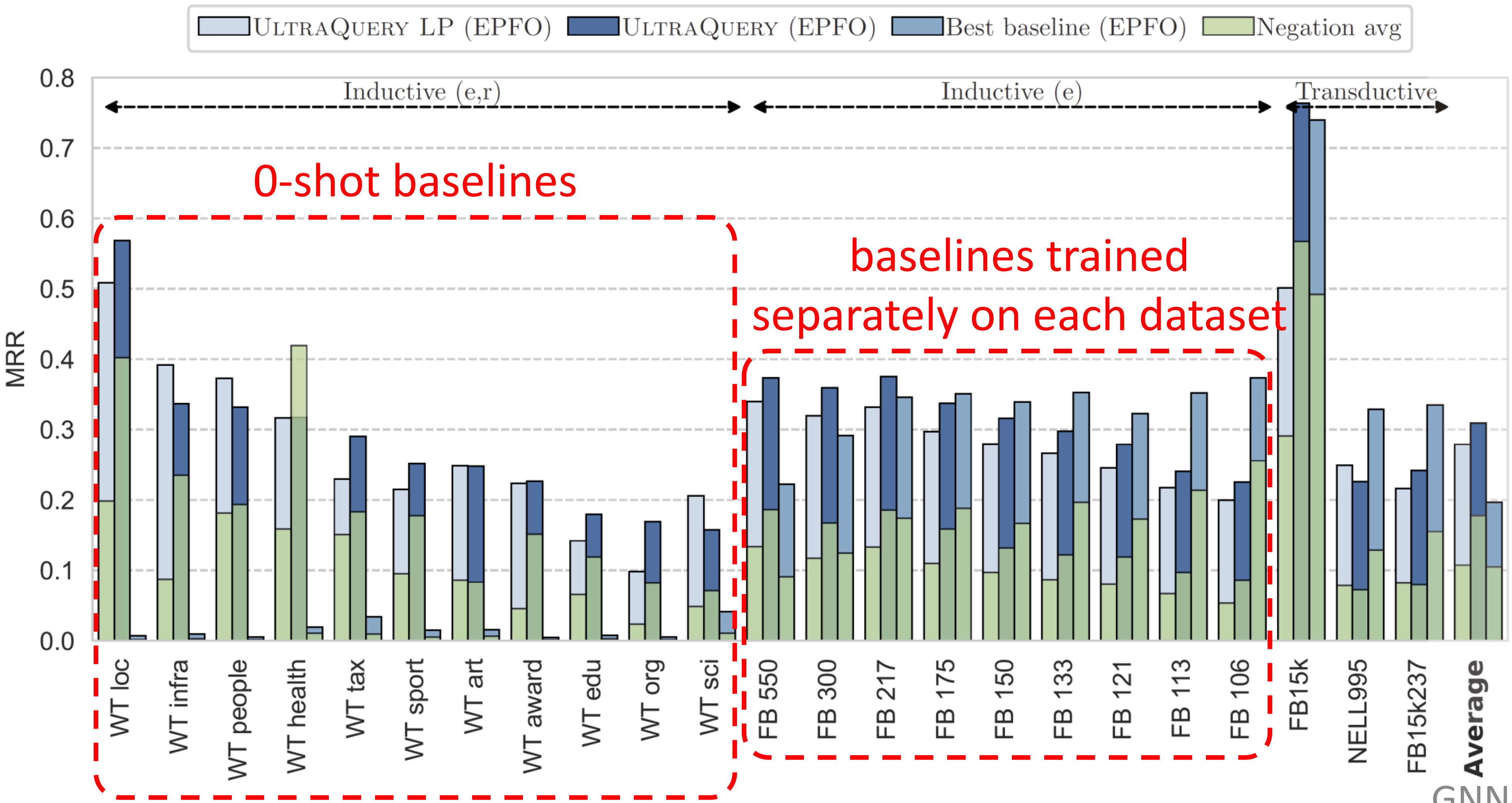
Better Compositional Generalization



Effective for Small Training Data



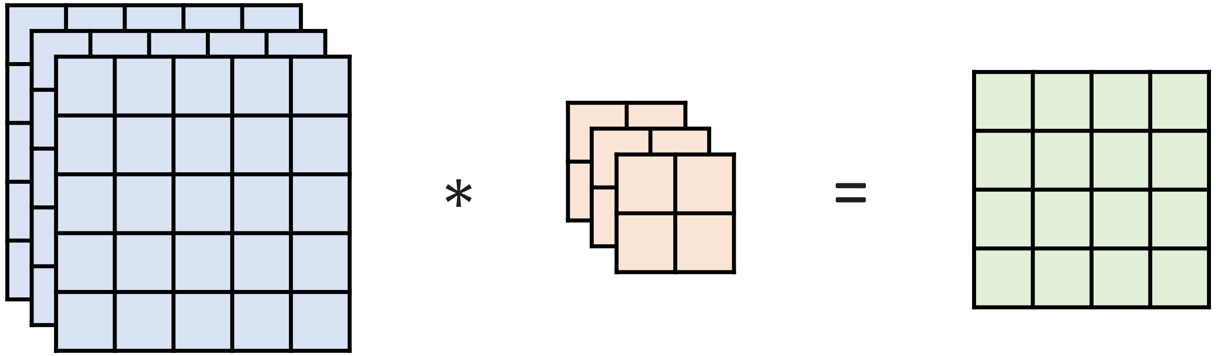
0-shot Inference of Multi-hop Queries



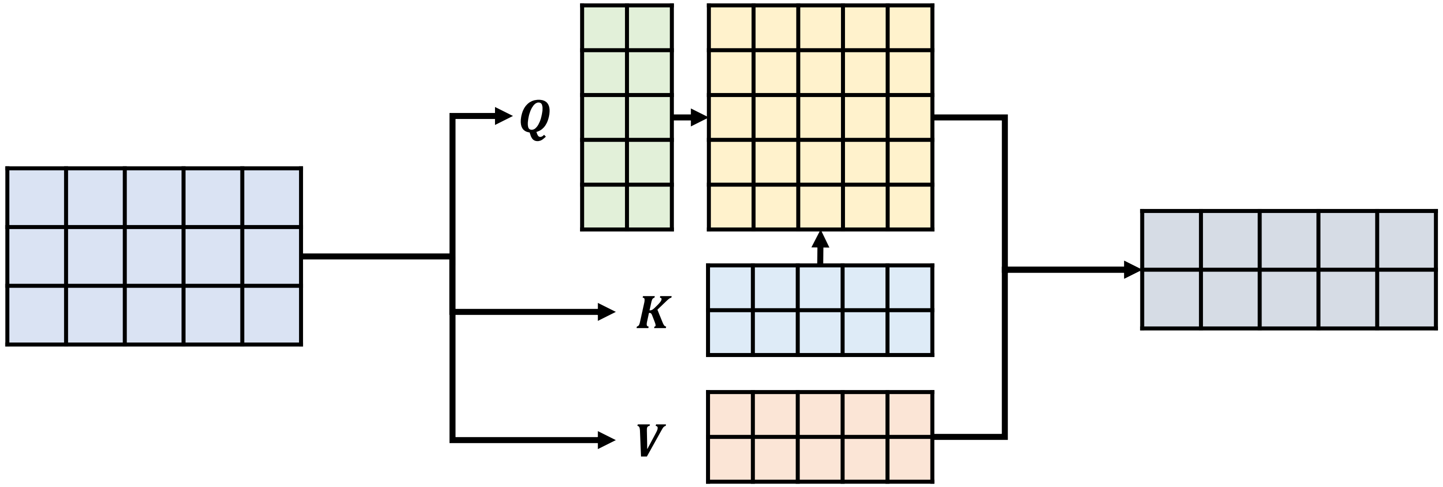
TorchDrug[1]: Simplifying **development on structured data** and related applications

[1] **Zhaocheng Zhu**, Chence Shi, Zuobai Zhang, Shengchao Liu, Minghao Xu, Xinyu Yuan, Yangtian Zhang, Junkun Chen, Huiyu Cai, Jiarui Lu, Chang Ma, Runcheng Liu, Louis-Pascal Xhonneux, Meng Qu, Jian Tang. TorchDrug: A Powerful and Flexible Machine Learning Platform for Drug Discovery. arXiv 2022.

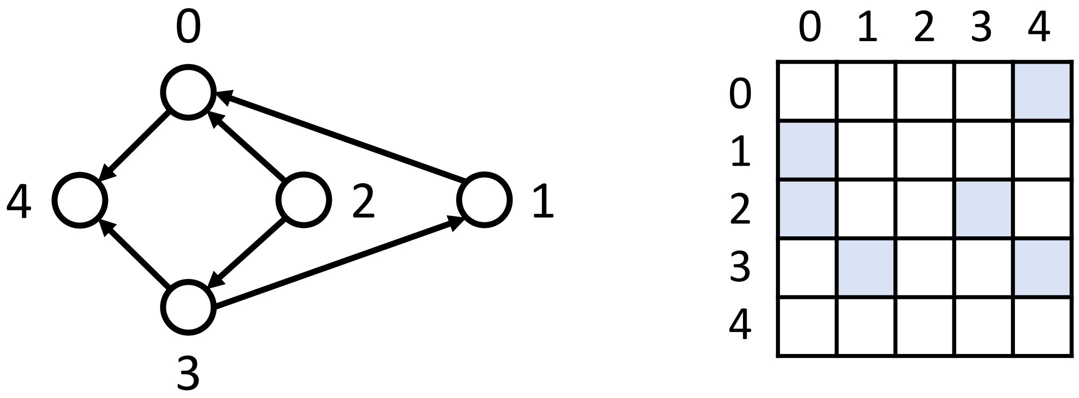
ML Implementation = Tensor Operations



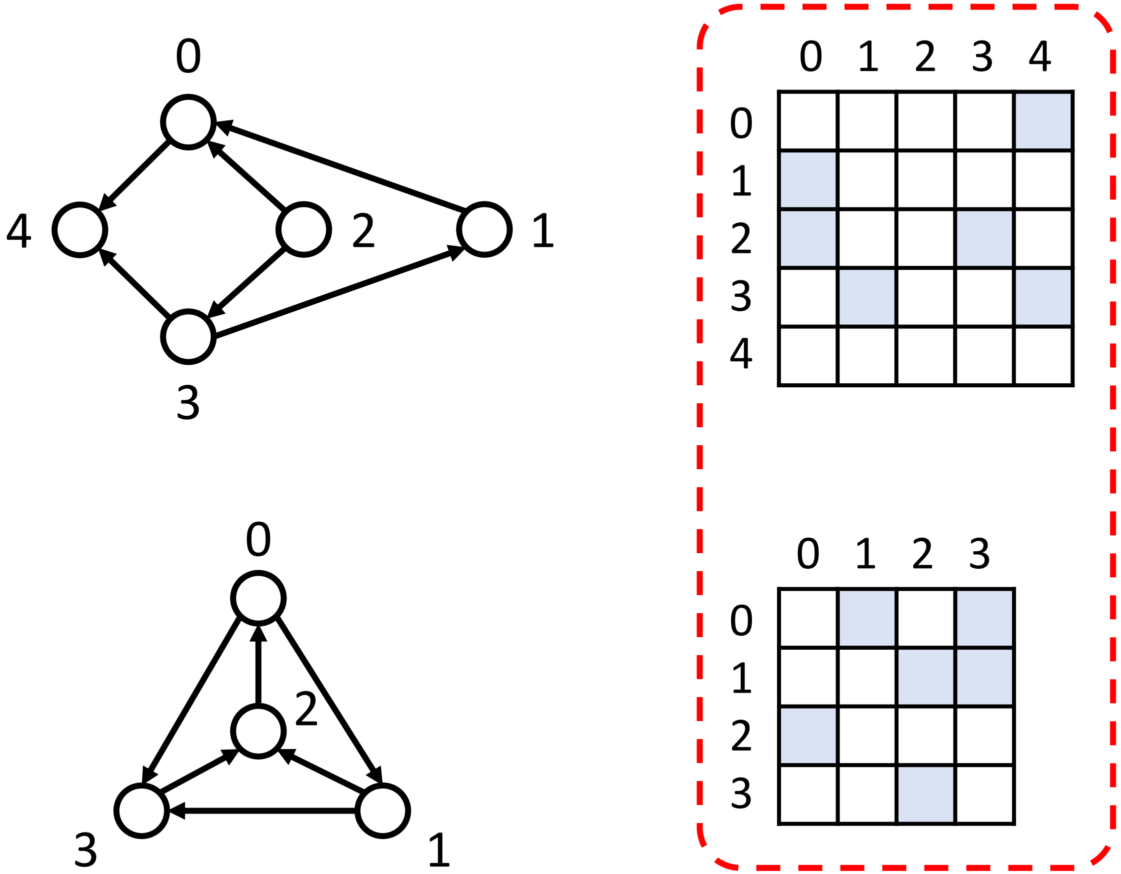
She sells sea shells.



Structured Data Meets Tensors

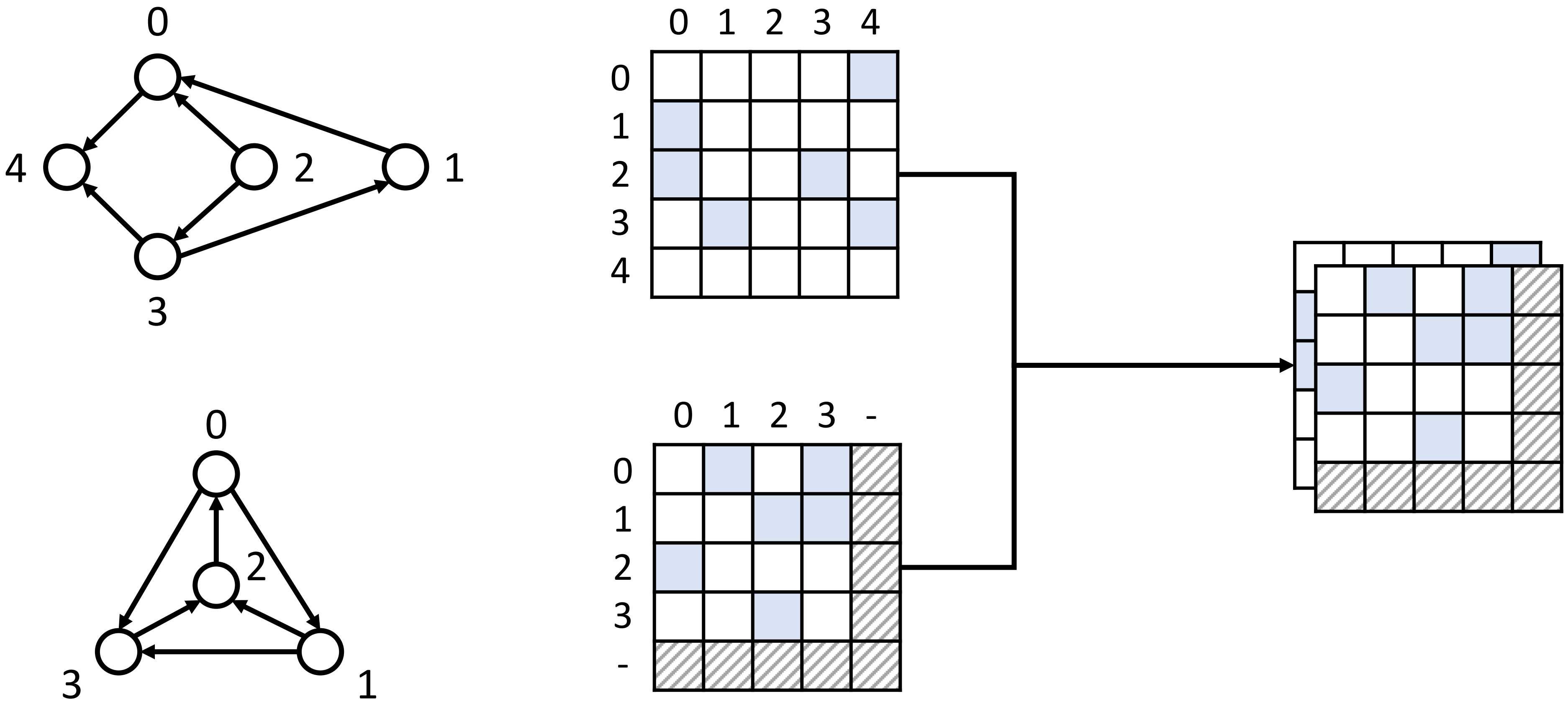


Structured Data Meets Tensors

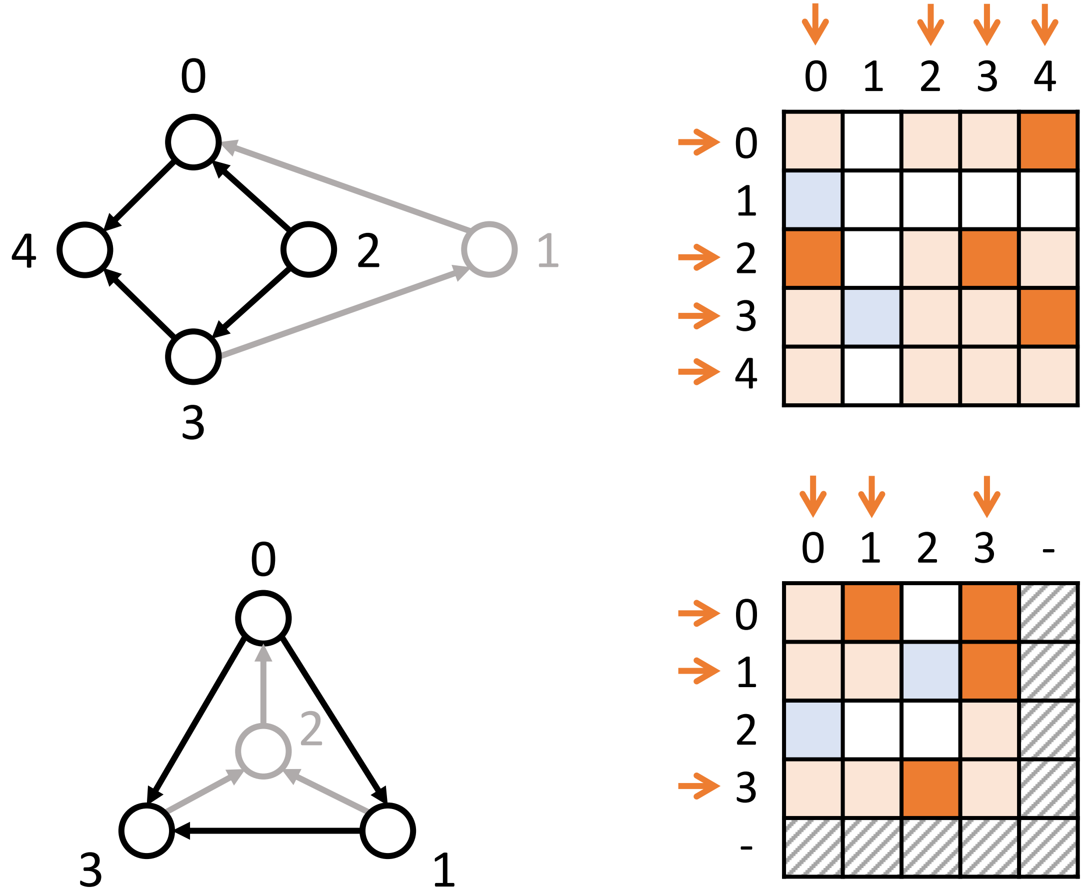


How to batch tensors of different shapes?

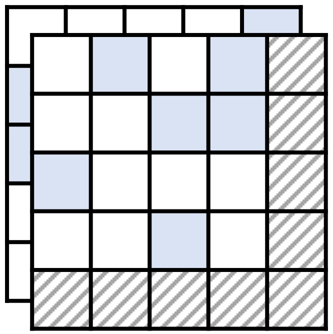
Naïve Solution: Padding



Naïve Solution: Padding



How to perform operations
on batched tensors?



Solutions

arrays

NumPy NetworkX
Network Analysis in Python

easy to implement

preprocessing

very slow

dense tensors

PyTorch

on-the-fly

not scalable

Solutions

arrays

NumPy NetworkX
Network Analysis in Python

easy to implement
preprocessing
very slow

dense tensors

PyTorch

on-the-fly
not scalable

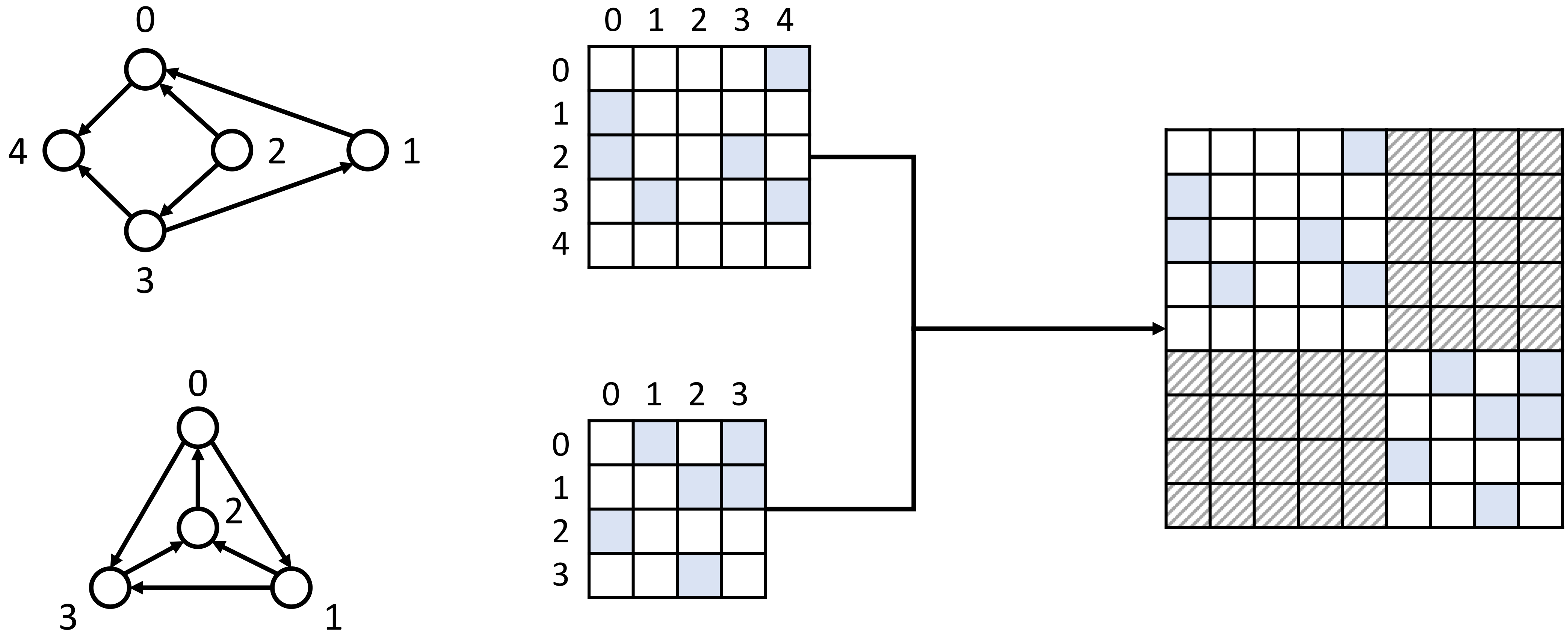
sparse tensors

PyTorch

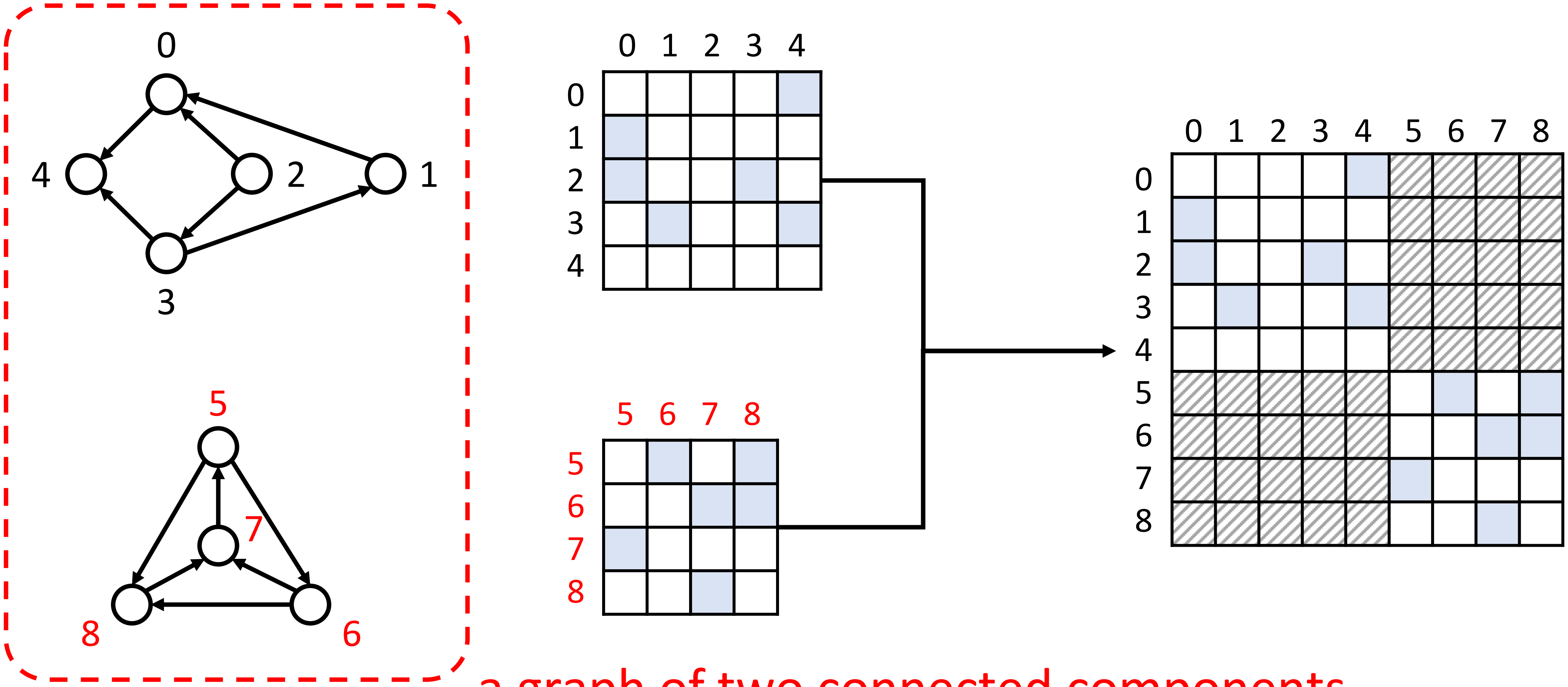
on-the-fly
scalable

How to implement?

The High-Level Idea

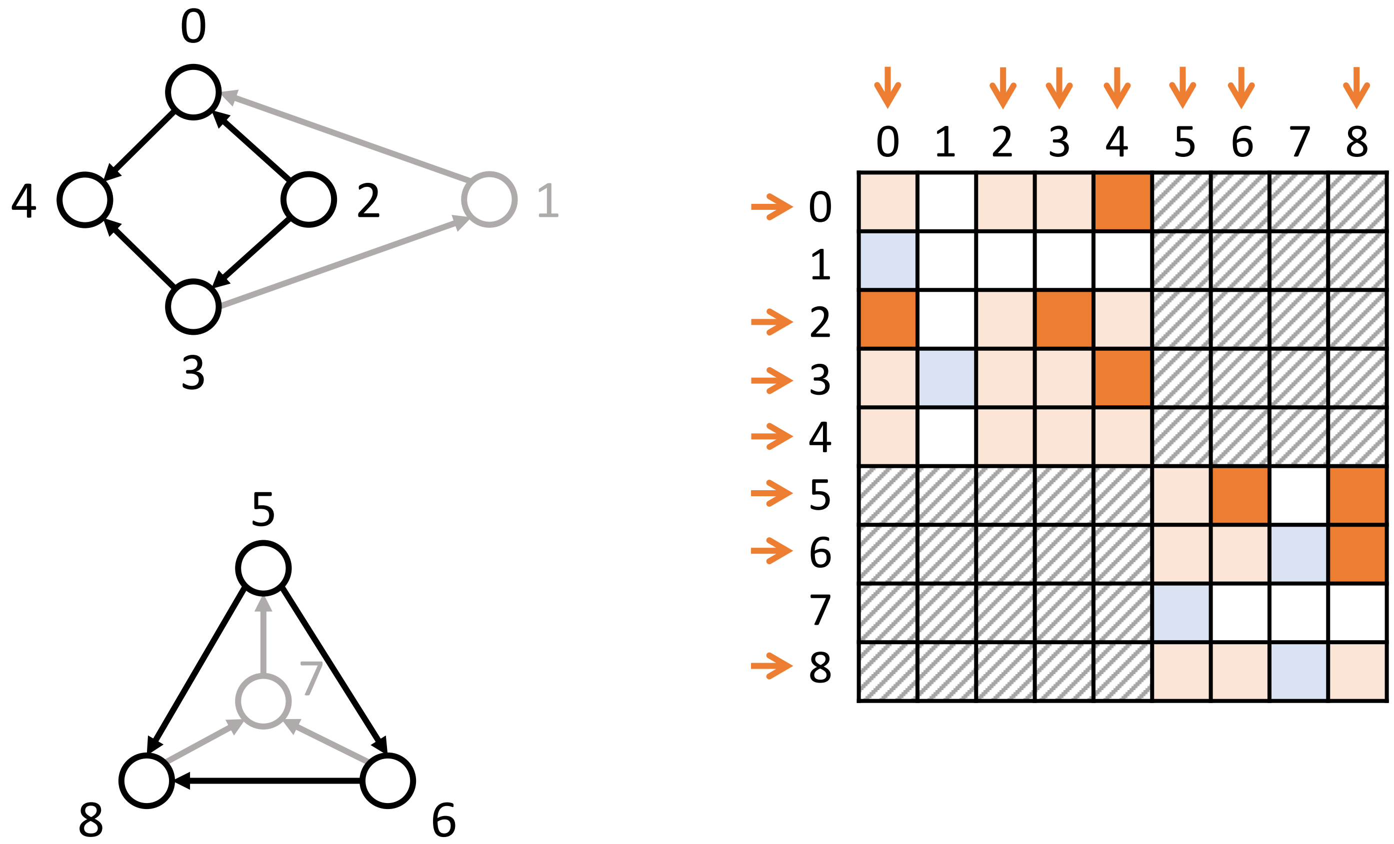


The High-Level Idea



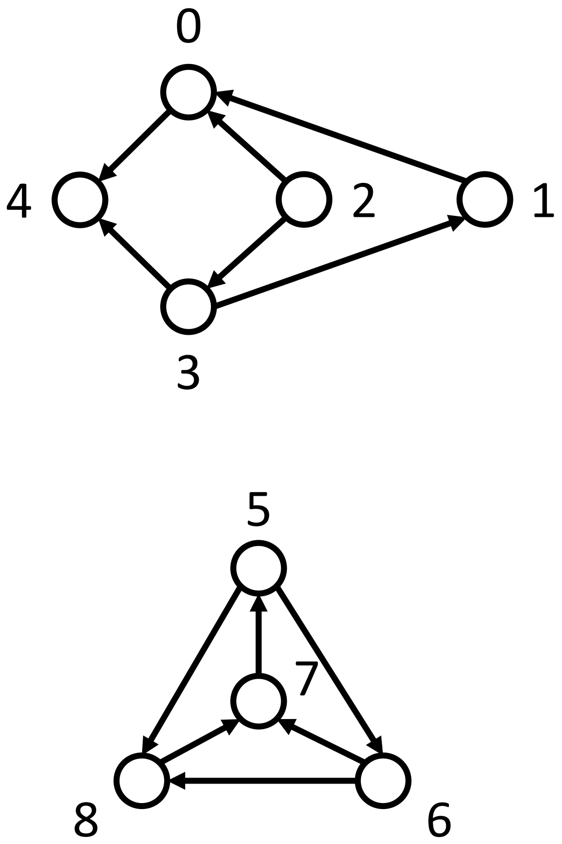
a graph of two connected components

The High-Level Idea



easy to implement!

Data Structure



torchdrug.data.PackedGraph

edge list

0	1	2	2	3	3	5	5	6	6	7	8
4	0	0	3	1	4	6	8	7	8	5	7

#nodes

5	4

#edges

6	6

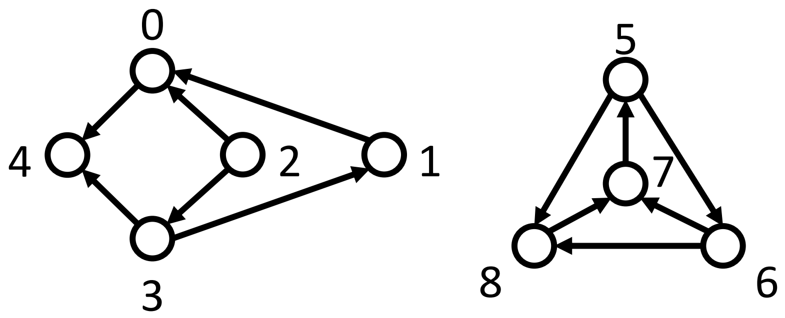
node/edge/graph attributes

……

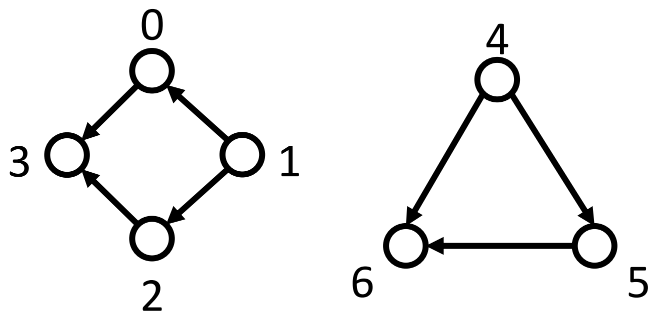
predefined or user-registered

Graph Operations

```
new_graphs = graphs.subgraph(node_index)
```



graphs



new graphs

graphs

edge list

0	1	2	2	3	3	5	5	6	6	7	8
4	0	0	3	1	4	6	8	7	8	5	7

#nodes

5	4

#edges

6	6

node index

0	2	3	4	5	6	8

new graphs

edge list

0	1	1	2	4	4	5
3	0	2	3	5	6	6

#nodes

4	3

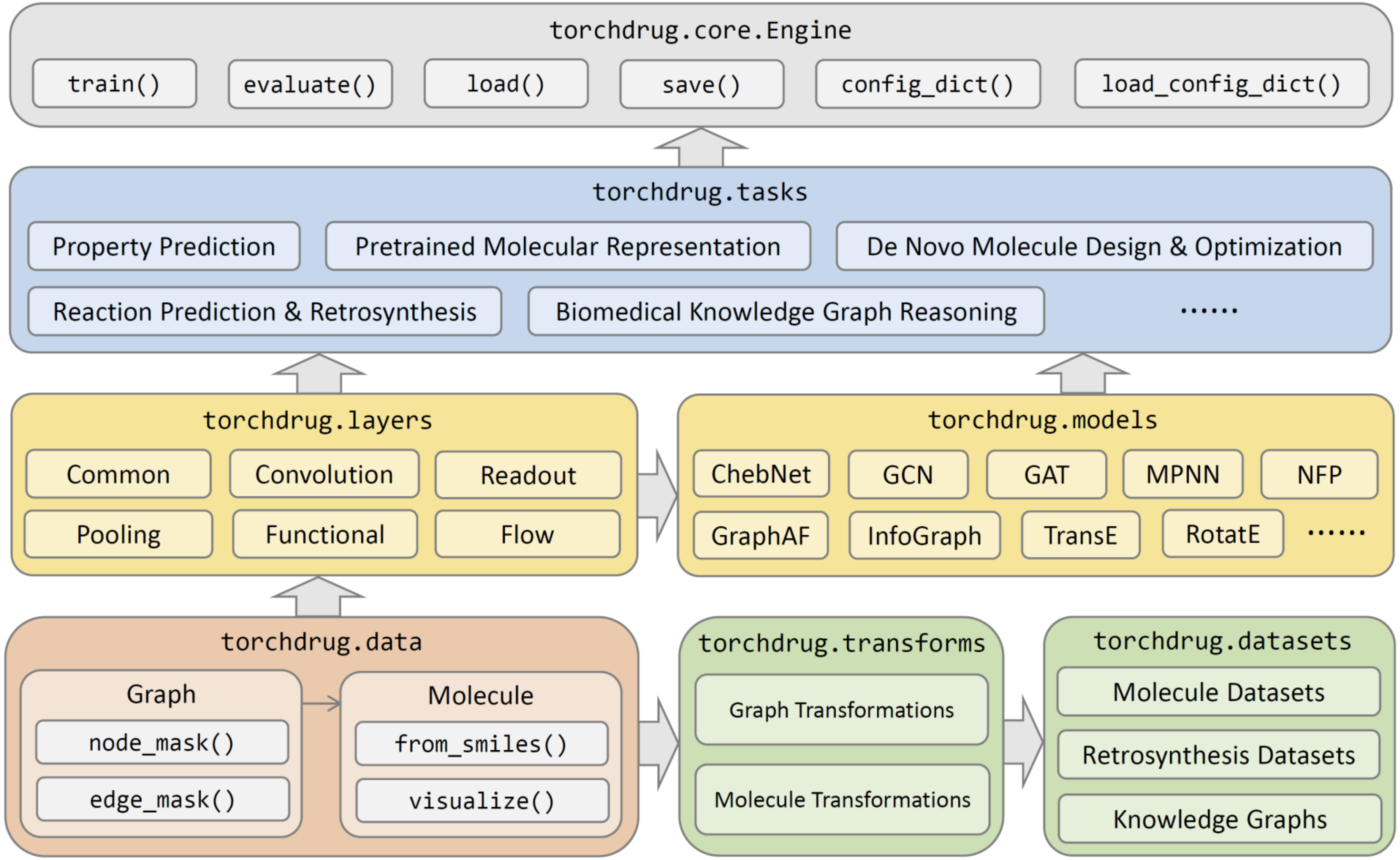
#edges

4	3

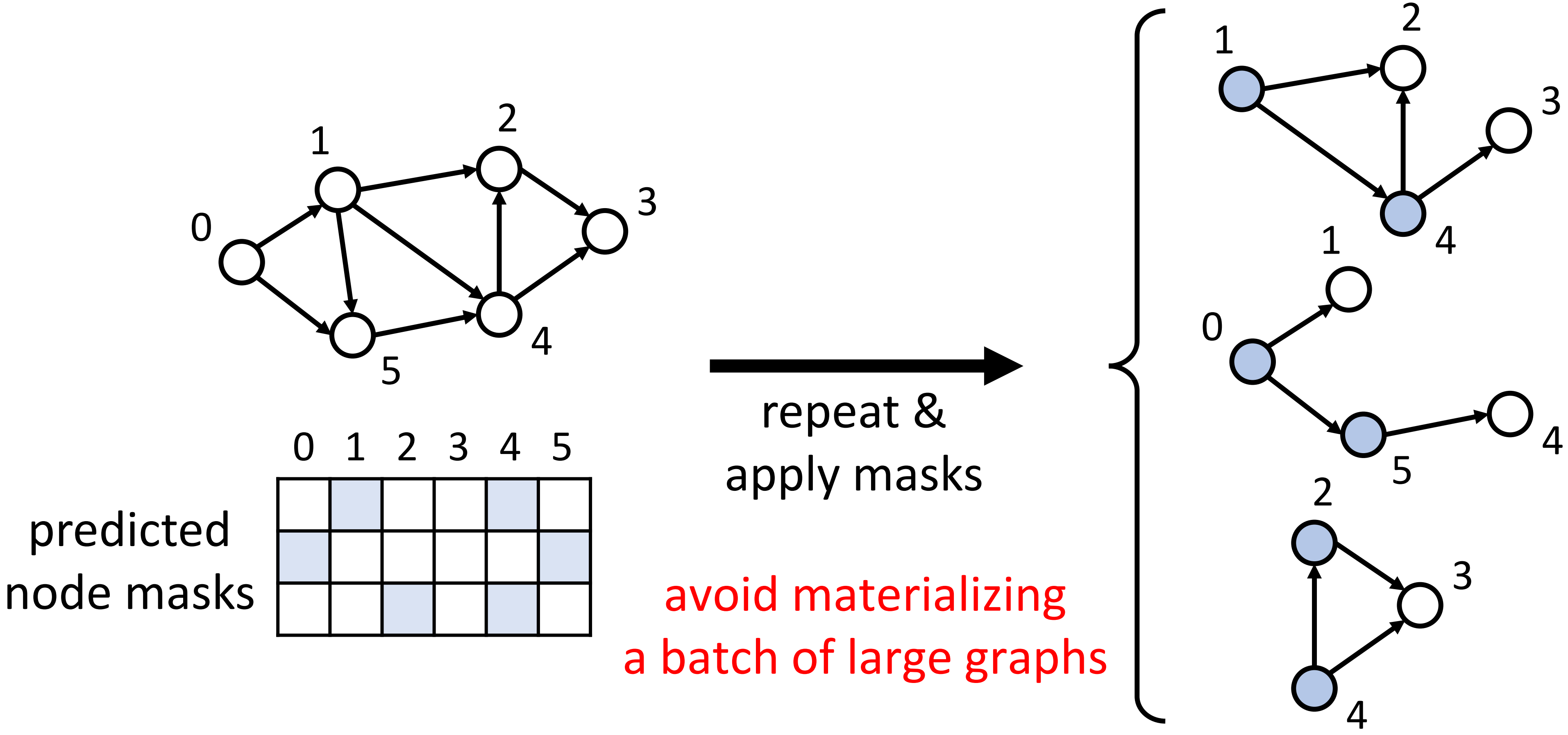
Supported Operations

Class	API	Graph Operation
PyTorch-like	`data.Graph.clone`	Clone this graph
	`data.Graph.detach`	Detach this graph
	`data.Graph.cpu`	Move this graph to CPU
	`data.Graph.cuda`	Move this graph to GPU
	`data.Graph.copy_`	Copy data from another graph
	`data.Graph.full`	Return a fully connected graph over nodes
	`data.Graph.repeat`	Repeat this graph like `torch.repeat`
	`data.PackedGraph.repeat_interleave`	Repeat this graph like `torch.repeat_interleave`
Node-level	`data.Graph.node_mask`	Mask out some nodes from this graph
	`data.Graph.compact`	Remove isolated nodes
Edge-level	`data.Graph.edge_mask`	Mask out some edges from this graph
	`data.Graph.directed`	Return a directed version of this graph
	`data.Graph.undirected`	Return an undirected version of this graph
	`data.Graph.match`	Search specific edges in this graph
Graph-level	`data.Graph.connected_components`	Split a graph into connected components
	`data.Graph.split`	Split a graph into a batch of graphs
	`data.Graph.pack`	Pack multiple graphs into a batch
	`data.Graph.line_graph`	Return a line graph of this graph
	`data.PackedGraph.graph_mask`	Mask out some graphs from this batch
	`data.PackedGraph.merge`	Merge some graphs into a smaller batch
	`data.PackedGraph.unpack`	Unpack a batch into multiple graphs
Molecule	`data.Molecule.ion_to_molecules`	Convert ions to molecules
Protein	`data.Protein.residue_mask`	Mask out some residues from this protein

Different Levels of Abstraction

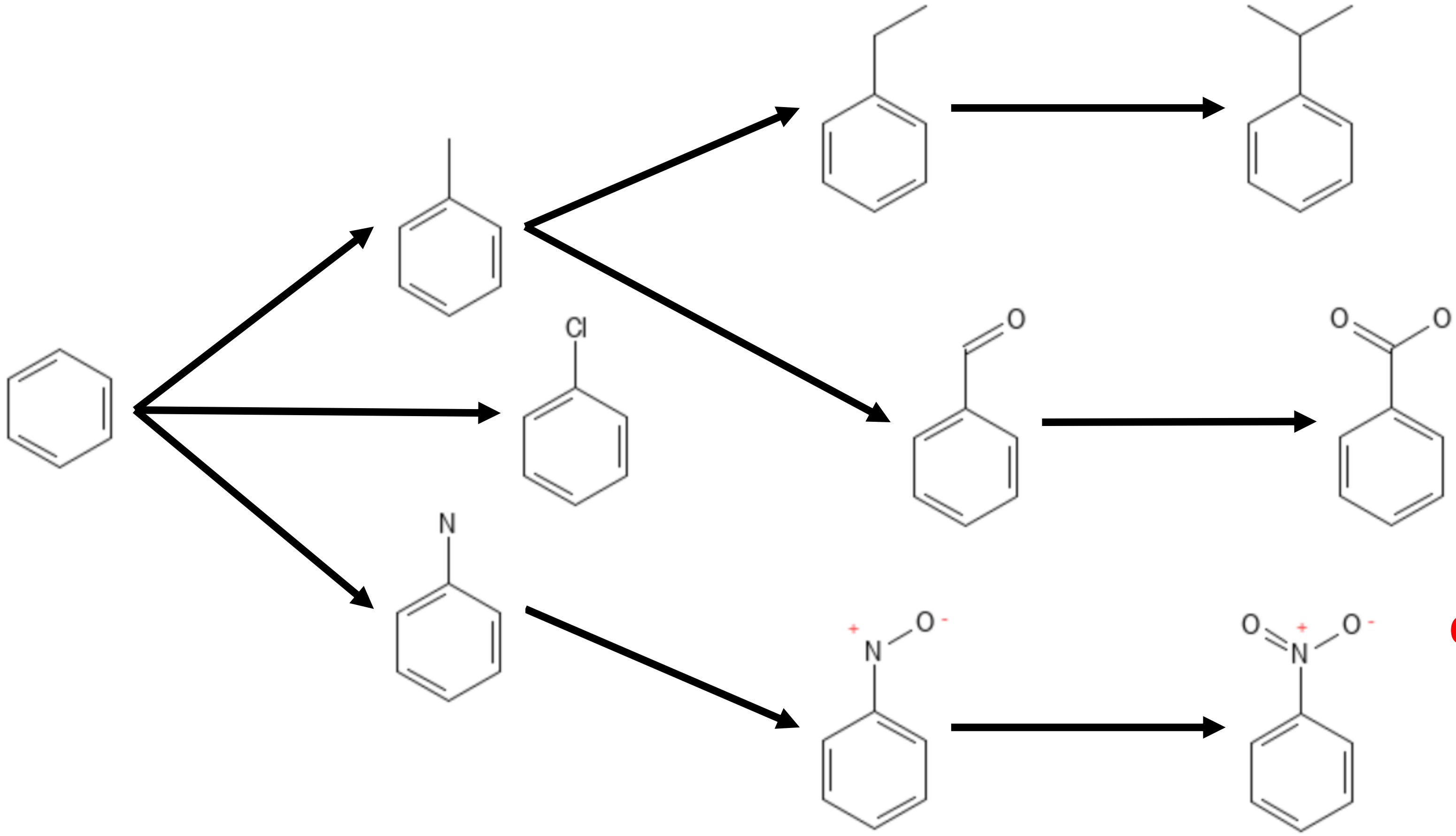


Use Case: Adaptive Message Passing[1]



[1] **Zhaocheng Zhu***, Xinyu Yuan*, Mikhail Galkin, Sophie Xhonneux, Ming Zhang, Maxime Gazeau, Jian Tang. A*Net: A Scalable Path-based Reasoning Approach for Knowledge Graphs. NeurIPS 2023.

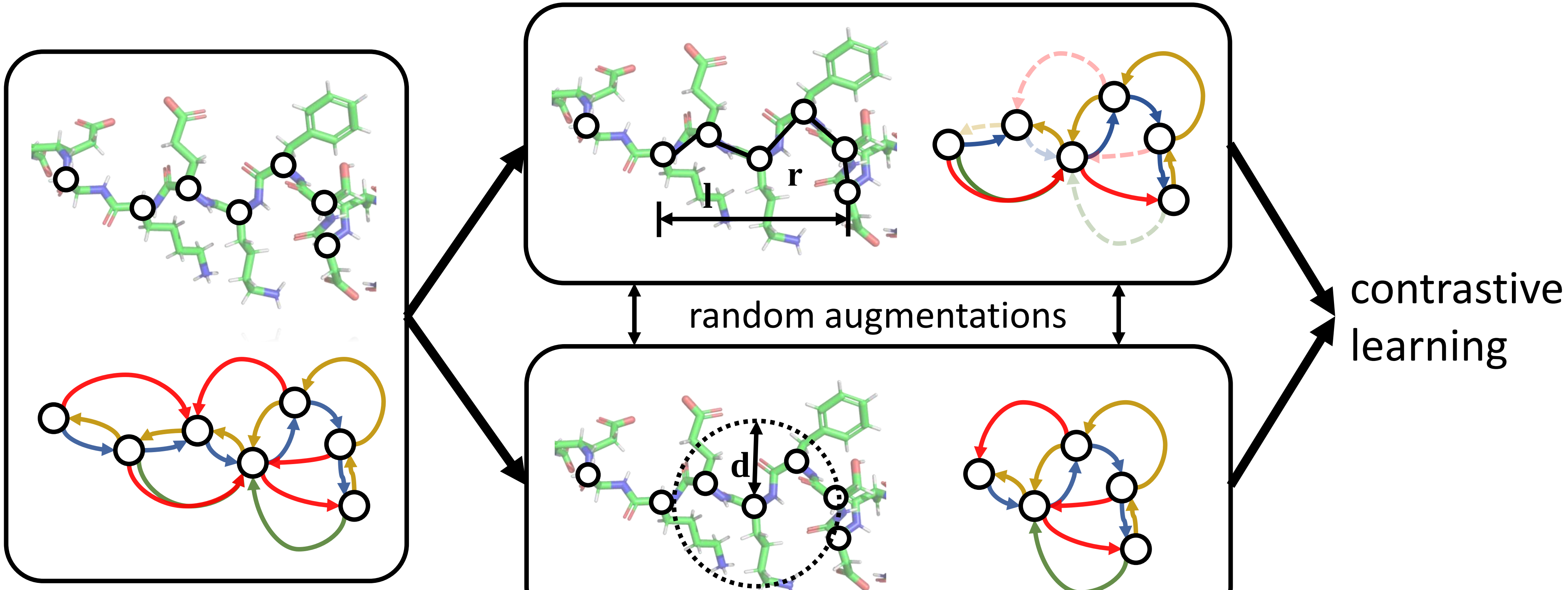
Use Case: Beam Search of Generation[1]



enable parallel generation

[1] Chence Shi, Minkai Xu, Hongyu Guo, Ming Zhang, Jian Tang. A Graph to Graphs Framework for Retrosynthesis Prediction. ICML 2020.

Use Case: On-the-fly Graph Construction[1]



graph from protein structure

speed up development

[1] Zuobai Zhang, Minghao Xu, Arian Jamasb, Vijil Chenthamarakshan, Aurelie Lozano, Payel Das, Jian Tang. Protein Representation Learning by Geometric Structure Pretraining. ICLR 2023.

Introduction — Our Works — **Conclusions**

What is the impact of our works?

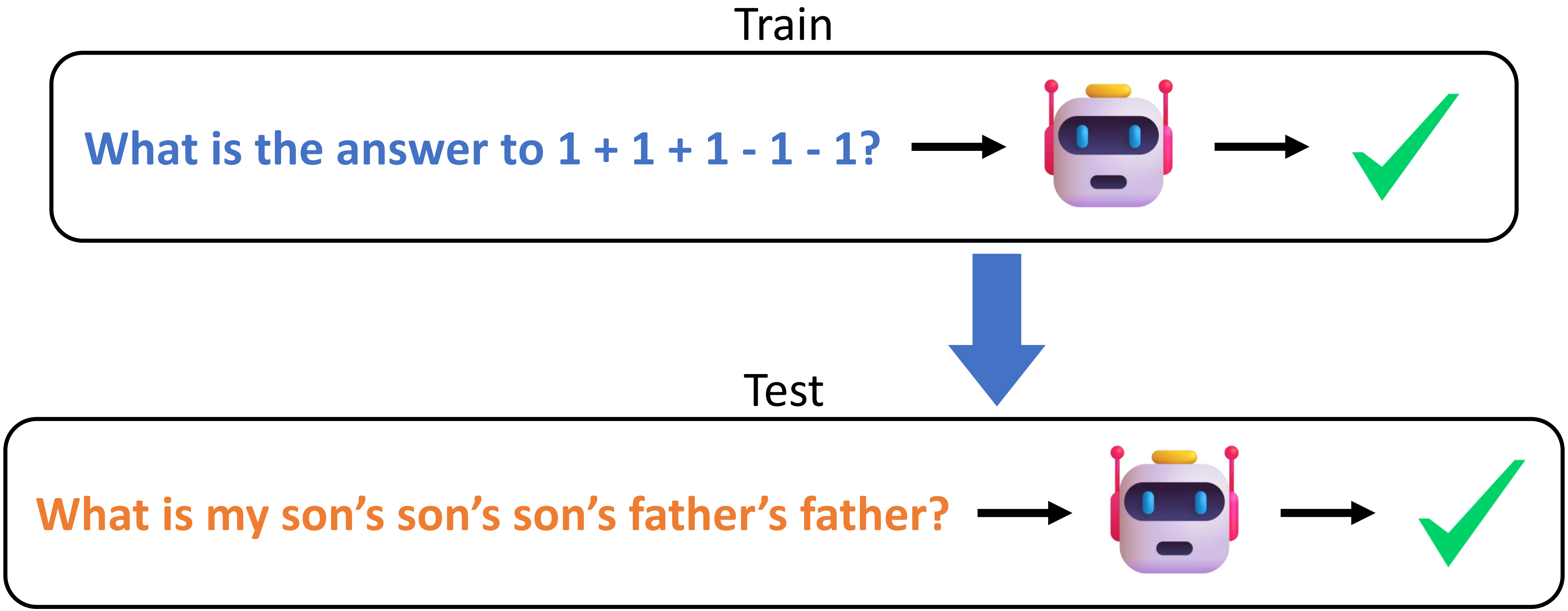
What is the future for reasoning and generalization?

Direct impact: Accelerating **the transition from transductive models to inductive ones**

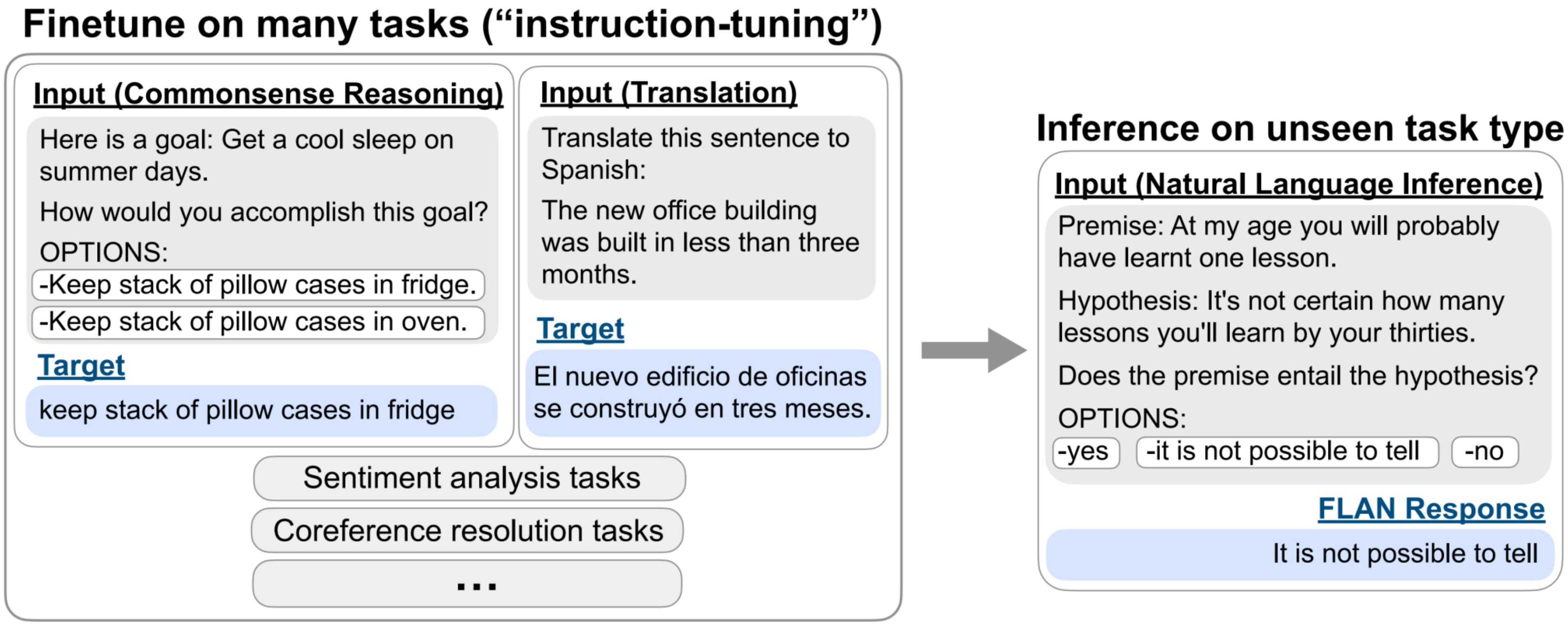
Lesson: Models with **inductive biases inspired by symbolic algorithms** **generalize better**

Belief: Many reasoning problems can be **unified**

Inductive Generalization on Text



The De Facto Approach: Instruction Tuning

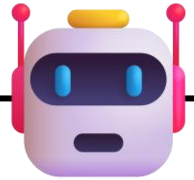


implicitly perform inductive generalization

[1] Jason Wei, et al. Finetuned Language Models Are Zero-Shot Learners. ICLR 2022.

Dealing with Parametric Knowledge

explicit knowledge



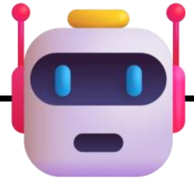
knowledge

query

reasoning

answer

parametric knowledge



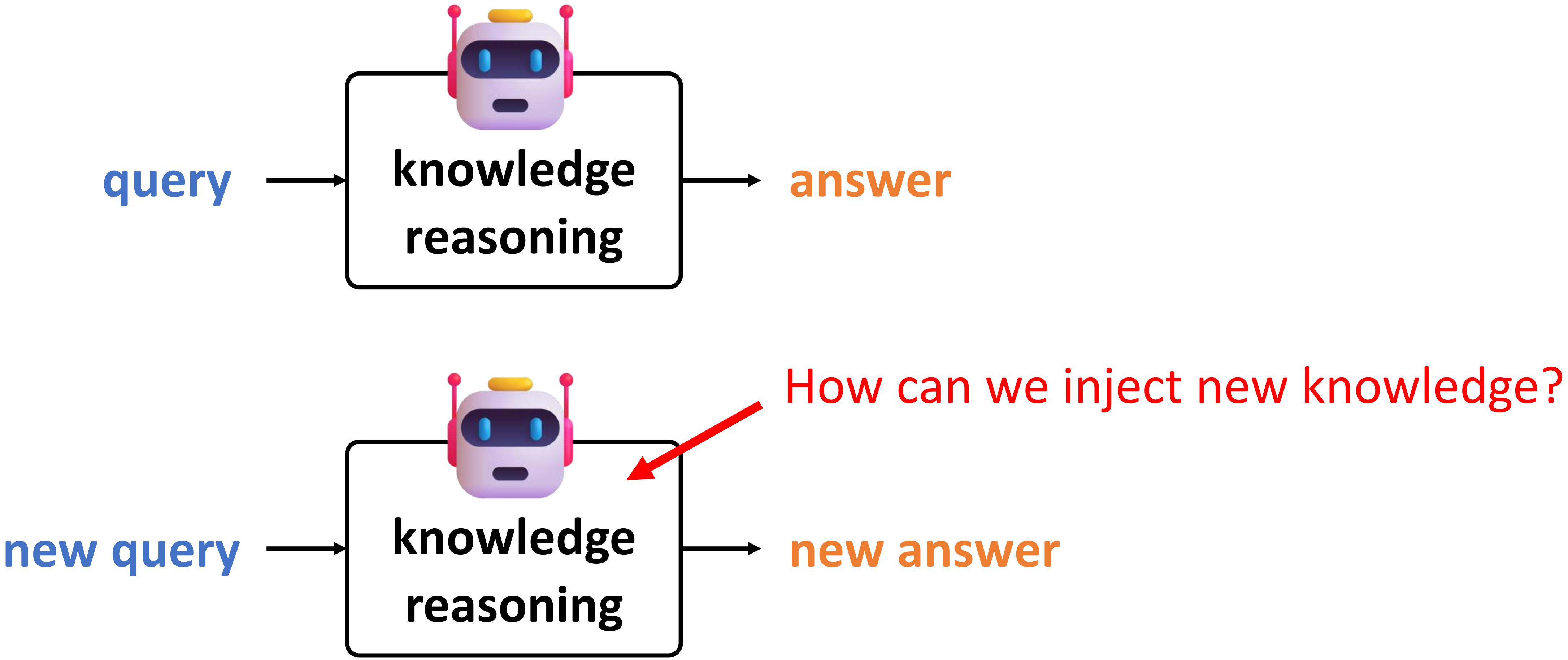
query

knowledge

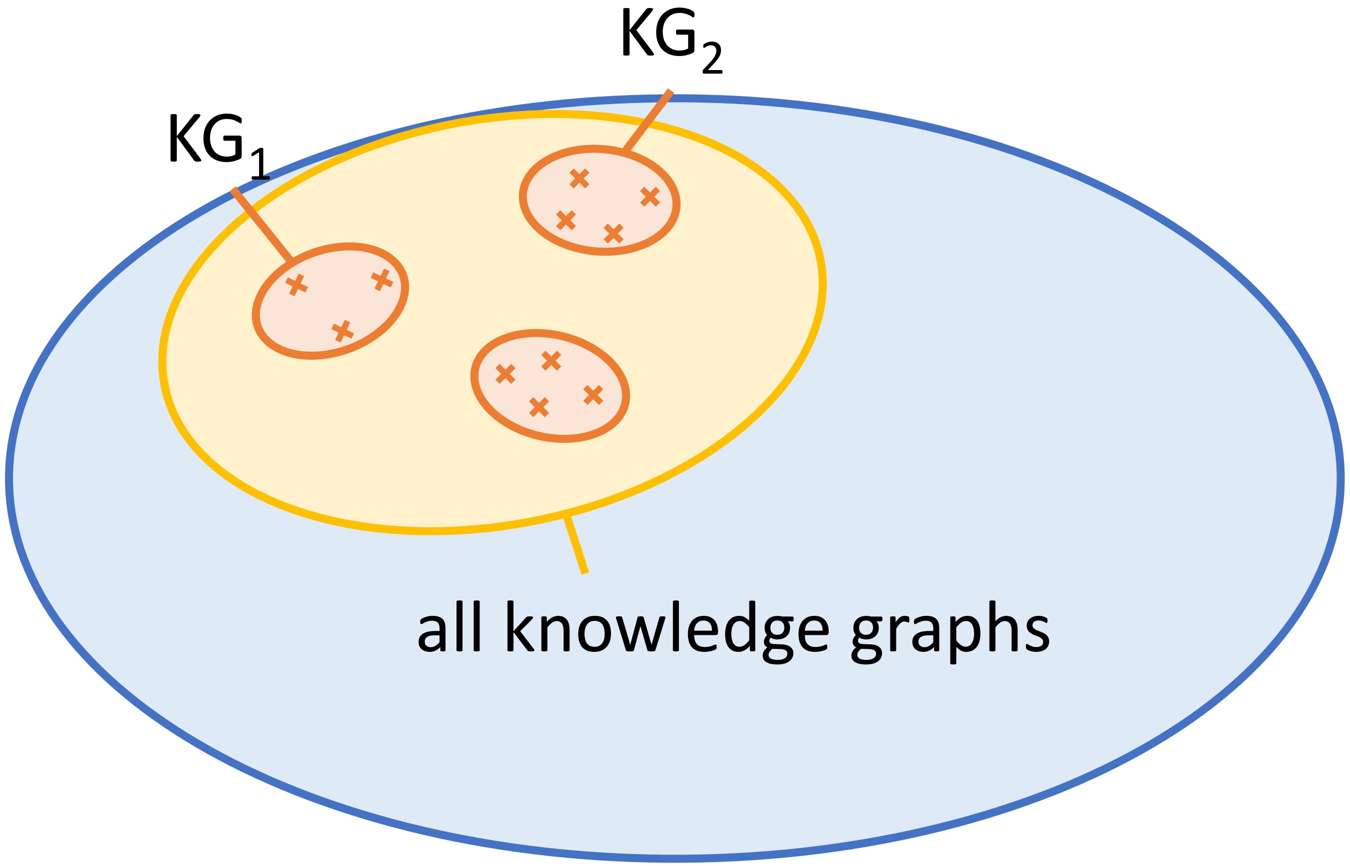
reasoning

answer

Dealing with Parametric Knowledge

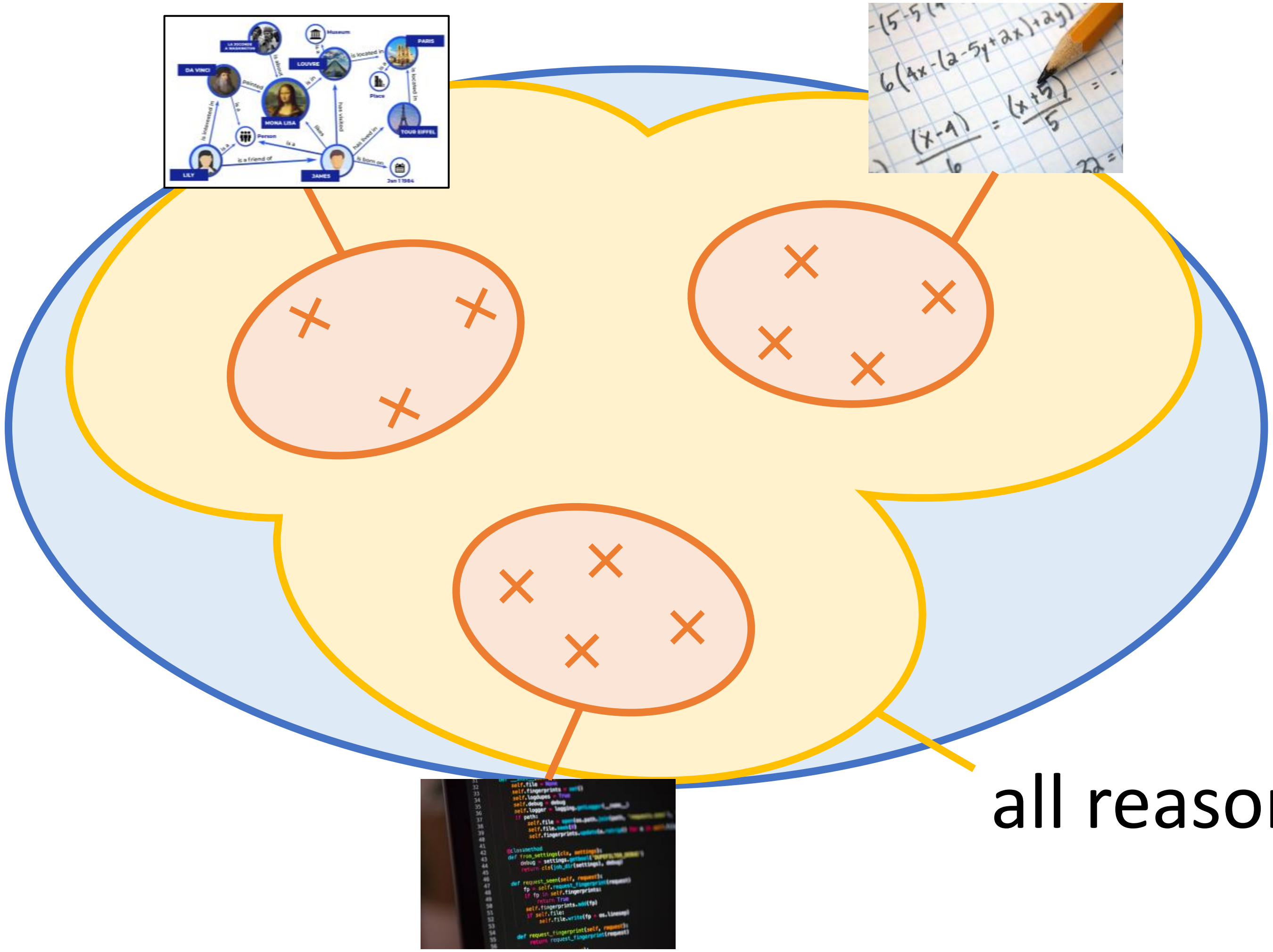


Expand the Scope of Generalization



unify knowledge graphs

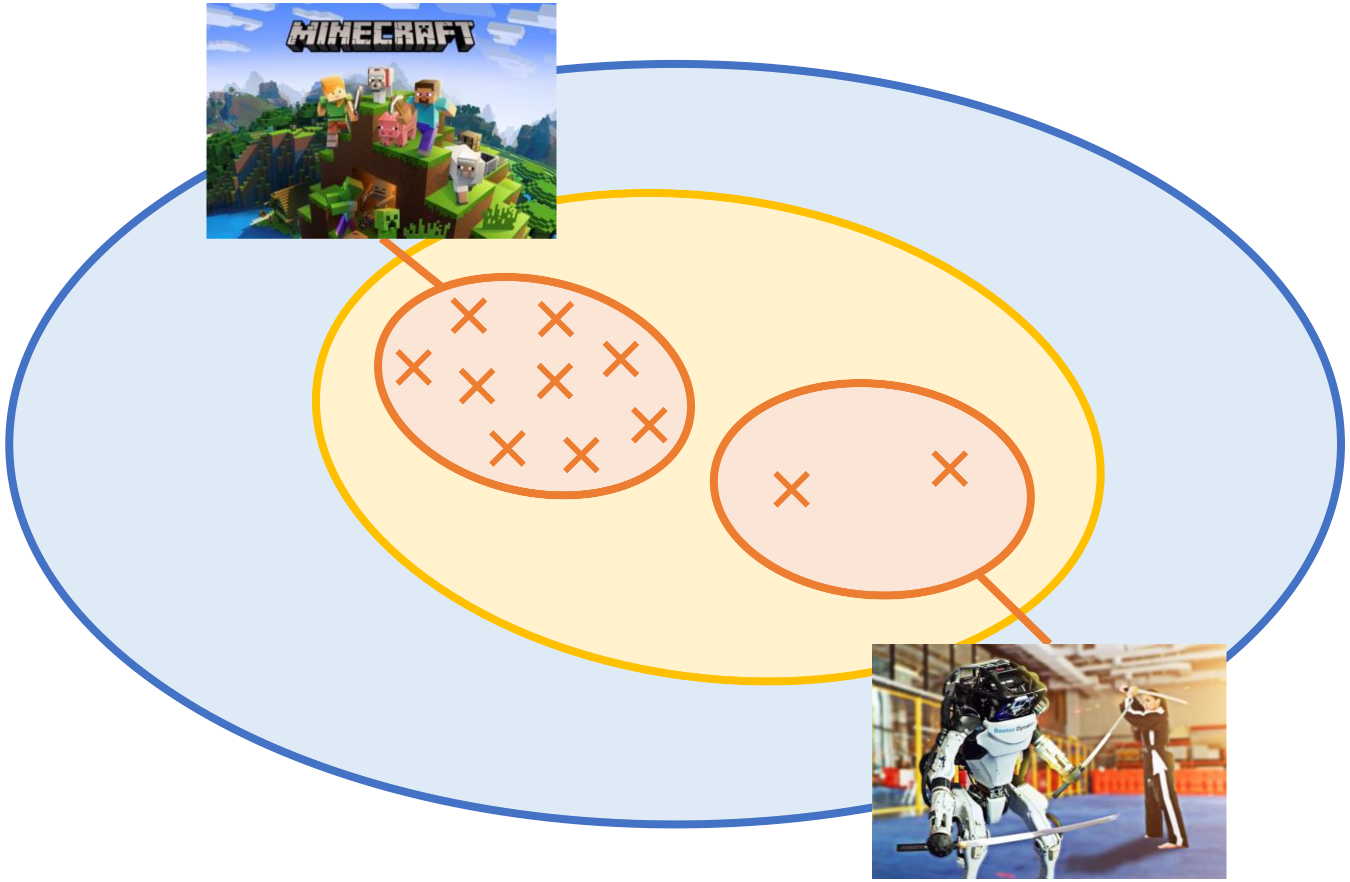
Expand the Scope of Generalization



unify reasoning tasks

all reasoning tasks

From Simulators to the Real World



save cost for data collection

Thank you! 🎉