NEURAL INFORMATION PROCESSING SYSTEMS

北京大學 PEKING UNIVERSITY

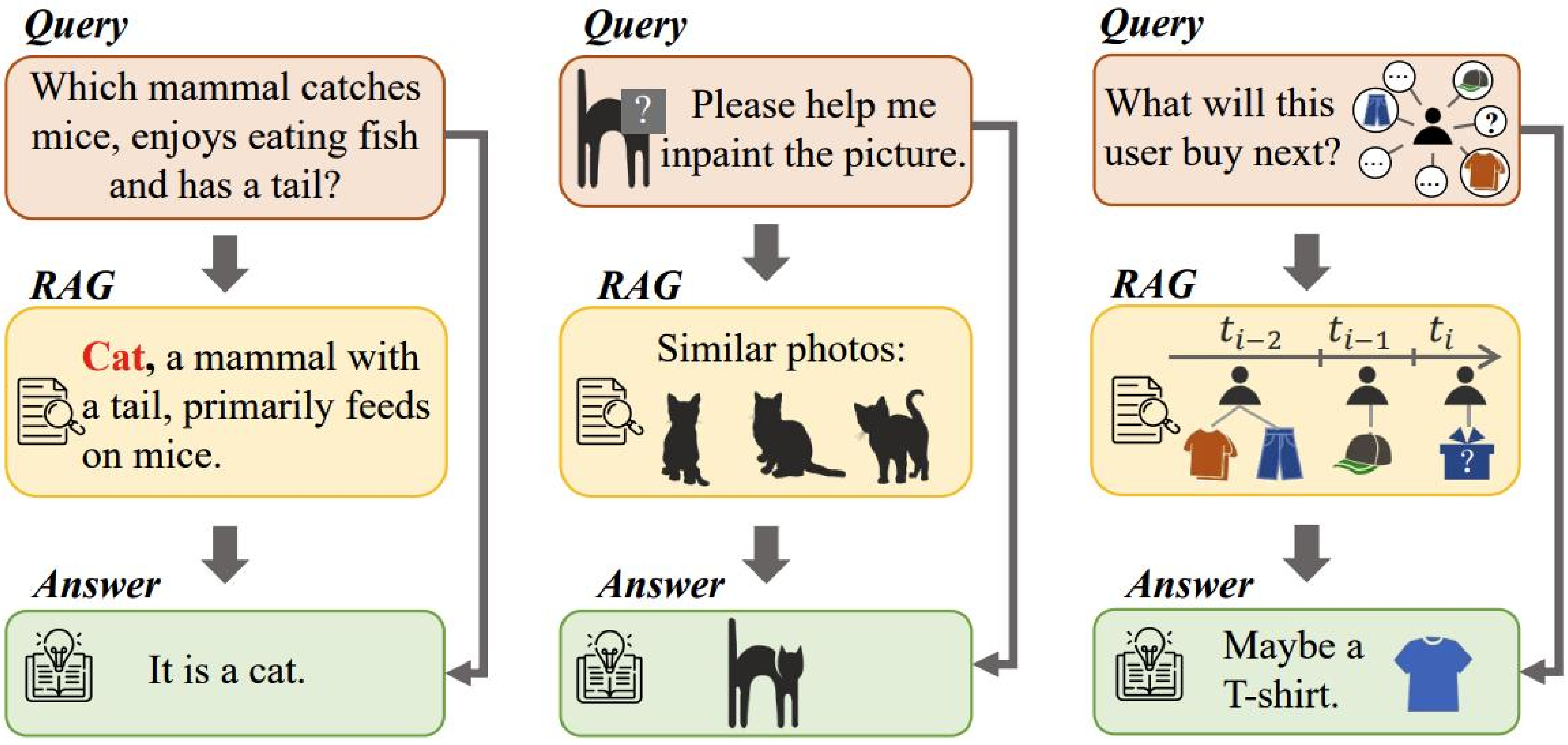
RAGRAPH: A General Retrieval-Augmented Graph Learning Framework

Xinke Jiang[♠*], Rihong Qiu[♠*], Yongxin Xu[♠*], Wentao Zhang[♢], Yichen Zhu[♢], Ruizhe Zhang[♠]
Yuchen Fang[♡], Xu Chu[♠], Junfeng Zhao[♠♣†], Yasha Wang[♠★†]

{xinkejiang, rihongqiu, xuyx, ruizhezhang}@stu.pku.edu.cn
{wentaozh2001, yichenzhu2014, fyclmiss}@gmail.com
{chu_xu, zhaojf, wangyasha}@pku.edu.cn

[♠]Key Laboratory of High Confidence Software Technologies (Peking University), School of Computer Science, Peking University, China
[♢]No Affiliation, [♡]University of Electronic Science and Technology of China
[♣]Big Data Technology Research Center, Nanhu Laboratory, Jiaxing, China
[★]Peking University Information Technology Institute, Tianjin Binhai, China

Background



(a) NLP (b) CV (c) GNNs

(a) / (b) RAG in NLP / CV

(c) RAG in GNNs for real-world use cases like recommendation or fraud detection

NEURAL INFORMATION PROCESSING SYSTEMS

Background

How GNN works?

Initial 0-th layer embeddings are equal to node features

$$h_v^0 = x_v$$

embedding of v at layer k

$$h_v^{(k+1)} = \sigma\left(W_k \sum_{u \in N(v)} \frac{h_u^{(k)}}{|N(v)|} + B_k h_v^{(k)}\right), \forall k \in \{0, \dots, K-1\}$$

Total number of layers

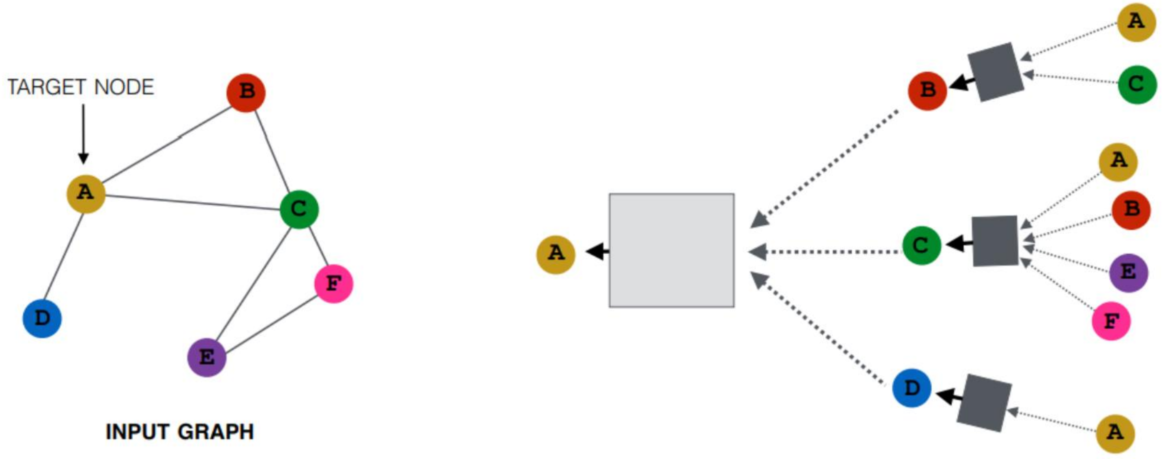
Average of neighbor's previous layer embeddings

$$z_v = h_v^{(K)}$$

Embedding after L layers of neighborhood aggregation

Non-linearity (e.g., ReLU)

Notice summation is a permutation invariant pooling/aggregation.

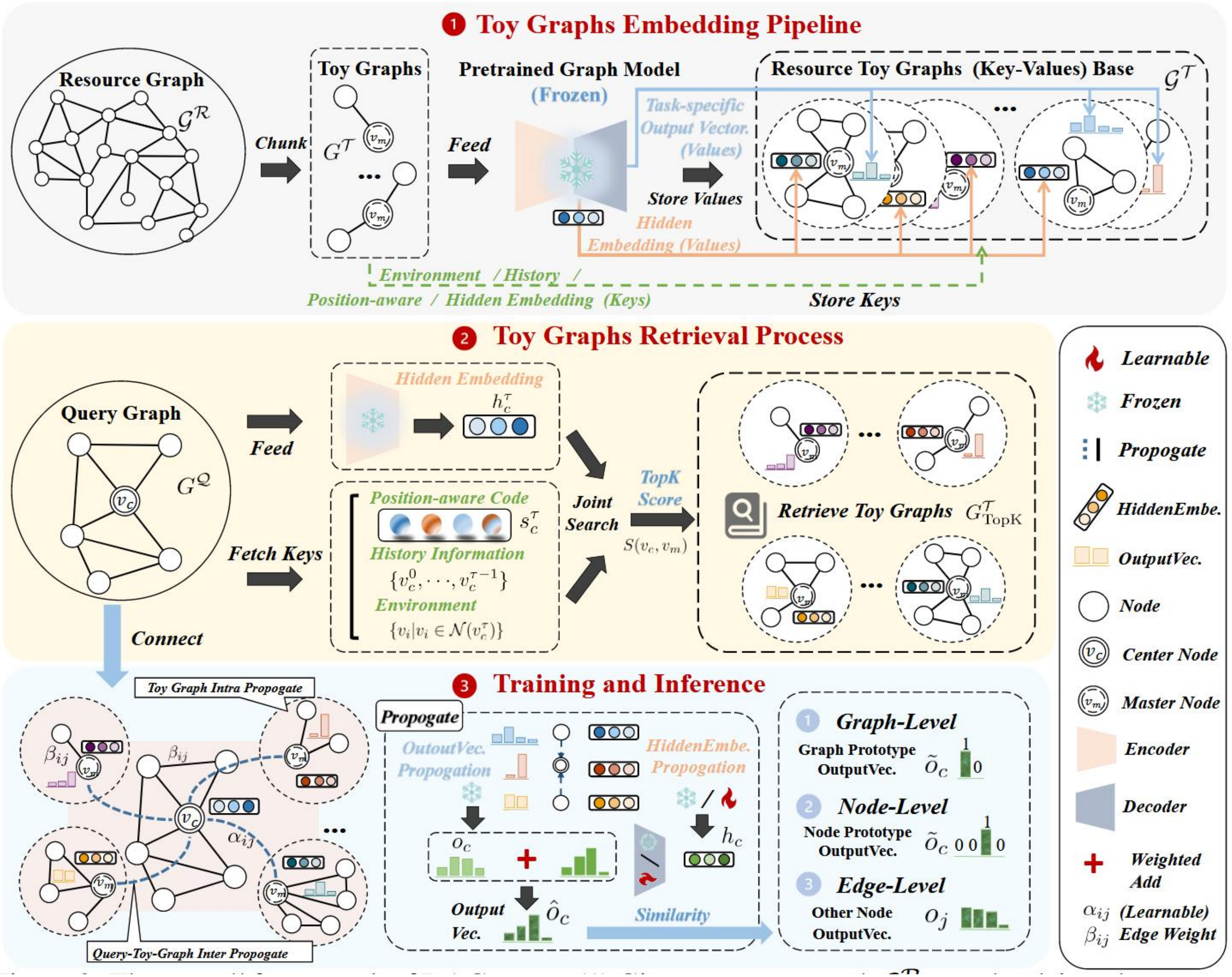


GNN is propagation

[1] Pictures are from CS224W: Machine Learning with Graphs. http://web.stanford.edu/class/cs224w/

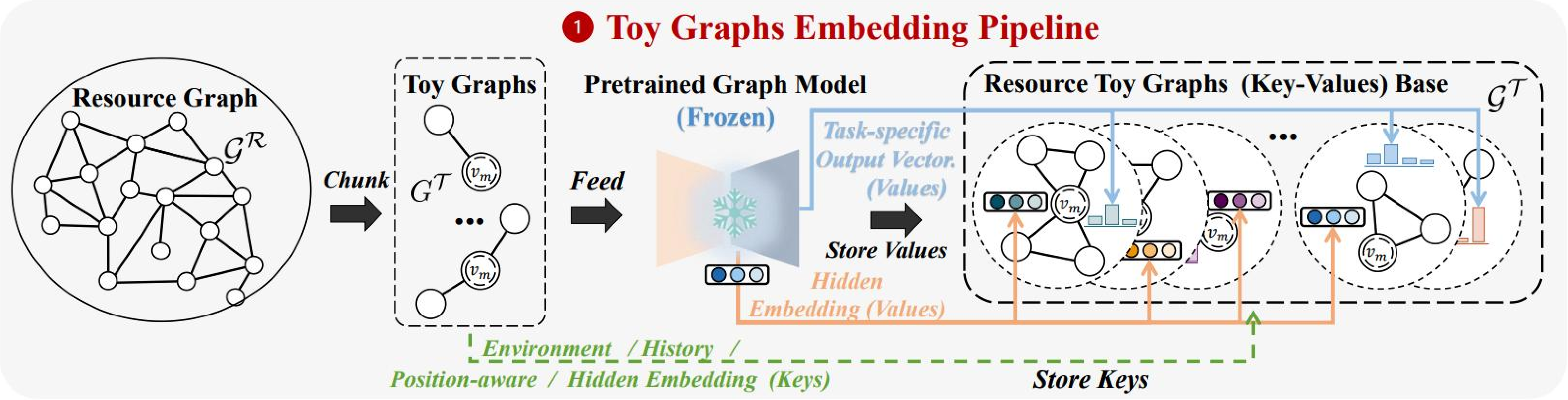
RAGRAPH

The pipeline of RAGRAPH:



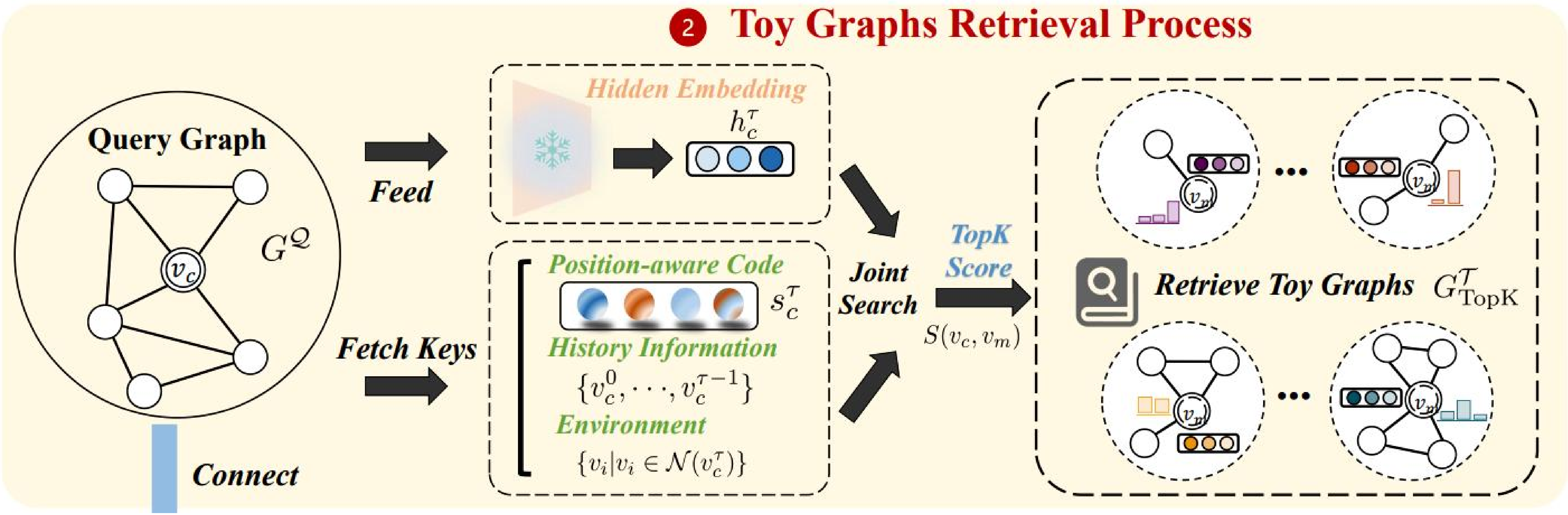
RAGRAPH

We extract many toy graphs from the chunks in the resource graph and use some augmentation methods to augment the ego graph. We leverage the pre-trained graph model to obtain embeddings and logits, serving respectively as key and value.



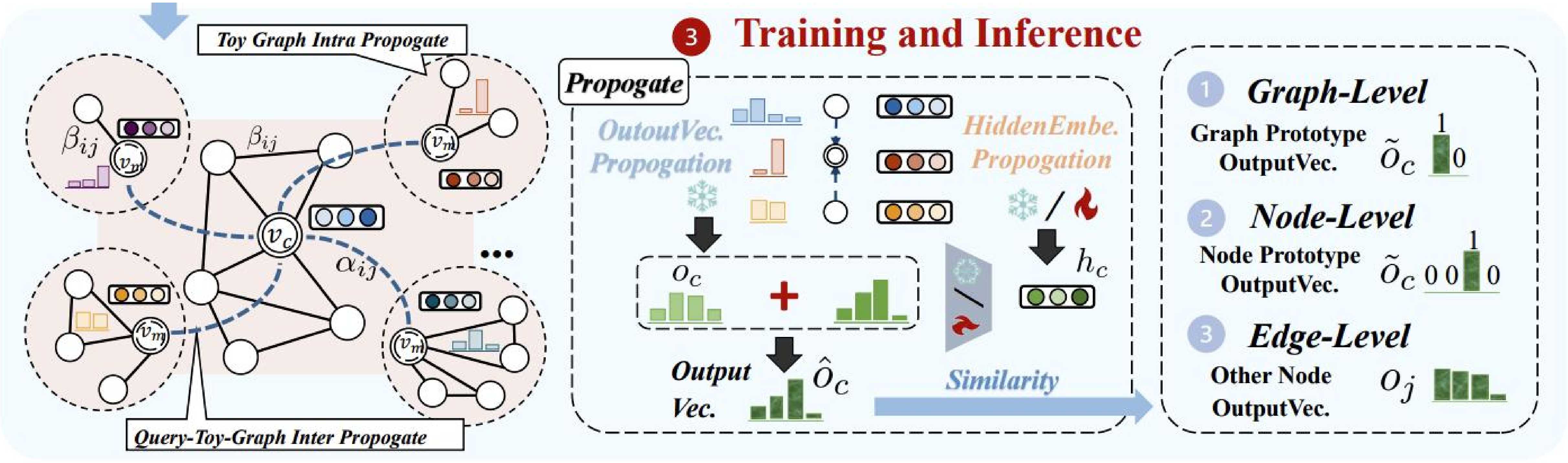
RAGRAPH

During retrieval, we start from the multidimensional similarity of the semantic, structural, environmental, and historical aspects of the graph, and calculate weighted similarity to retrieve the topK toy graphs



RAGRAPH

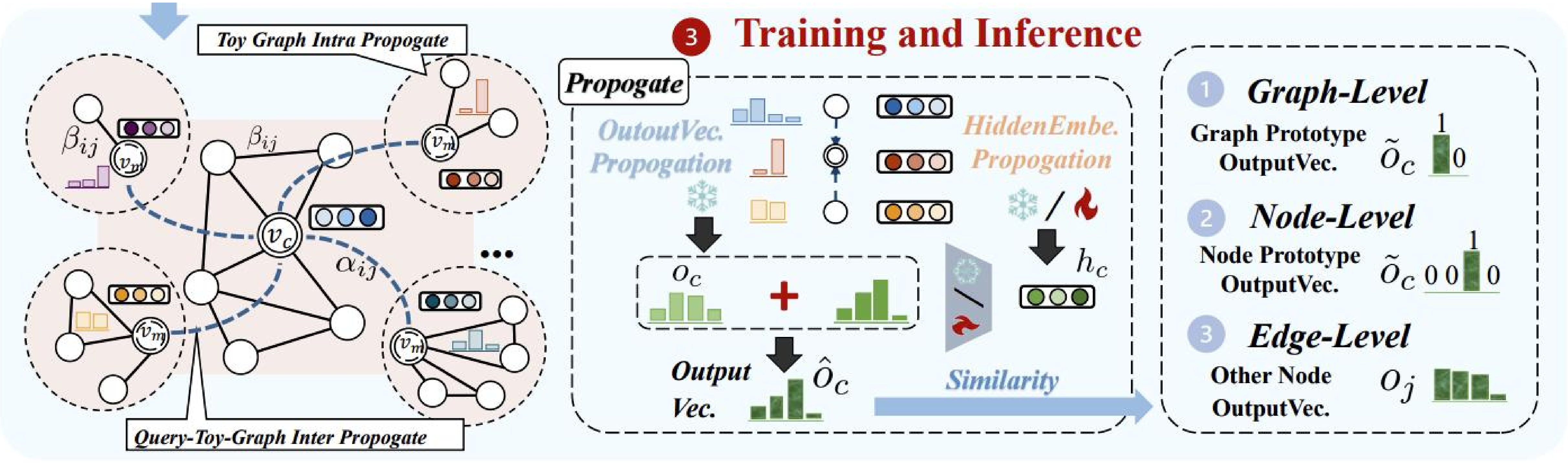
We use inner propagation to propagate the retrieved information to the master node of the toy graph, and then use inter propagation to propagate the retrieved information from the master node to the center node of the query graph. This process is achieved through the propagation of graphs.



RAG_{RAPH}

$$\hat{o}_c = \gamma o_c + (1-\gamma)\textsc{Decoder}(h_c),$$

Next, we use the retrieved embedding through the decoder, and combine X and Y with the retrieved logits to obtain the final logits, and use the similarity comparison function to do subtasks.



Experiment

- **Dataset**
  - Seven real-world Graph datasets:

Table 4: Statistics of the experimental datasets and summary of datasets.

Statistics	TAOBAO	KOUBEI	AMAZON	PROTEINS	COX2	ENZYMES	BZR
# Nodes per Graph	204,168	221,366	238,735	39.06	41.22	32.63	35.75
# Edges per Graph	8,795,404	3,986,609	876,237	72.82	43.45	62.14	38.36
# Density	8.6e-4	3.3e-4	6.2e-5	4.8e-2	2.6e-2	5.9e-2	3.0e-2
# Graphs	1	1	1	1,113	467	600	405
# Graph Classes	/	/	/	2	2	6	2
# Node Features	1	1	1	1	3	18	3
# Node Classes	/	/	/	3	/	3	/
Snapshot Granularity	daily	weekly	weekly	/	/	/	/
Task	Edge	Edge	Edge	Node, Graph	Graph	Node, Graph	Graph
Type	Dynamic	Dynamic	Dynamic	Static	Static	Static	Static
Dataset Partition	Snapshot	Snapshot	Snapshot	Node, Graph	Graph	Node, Graph	Graph

Task: Graph/Node Classification, Link Prediction from both **Static/Dynamic** Level

Experiment

NEURAL INFORMATION PROCESSING SYSTEMS

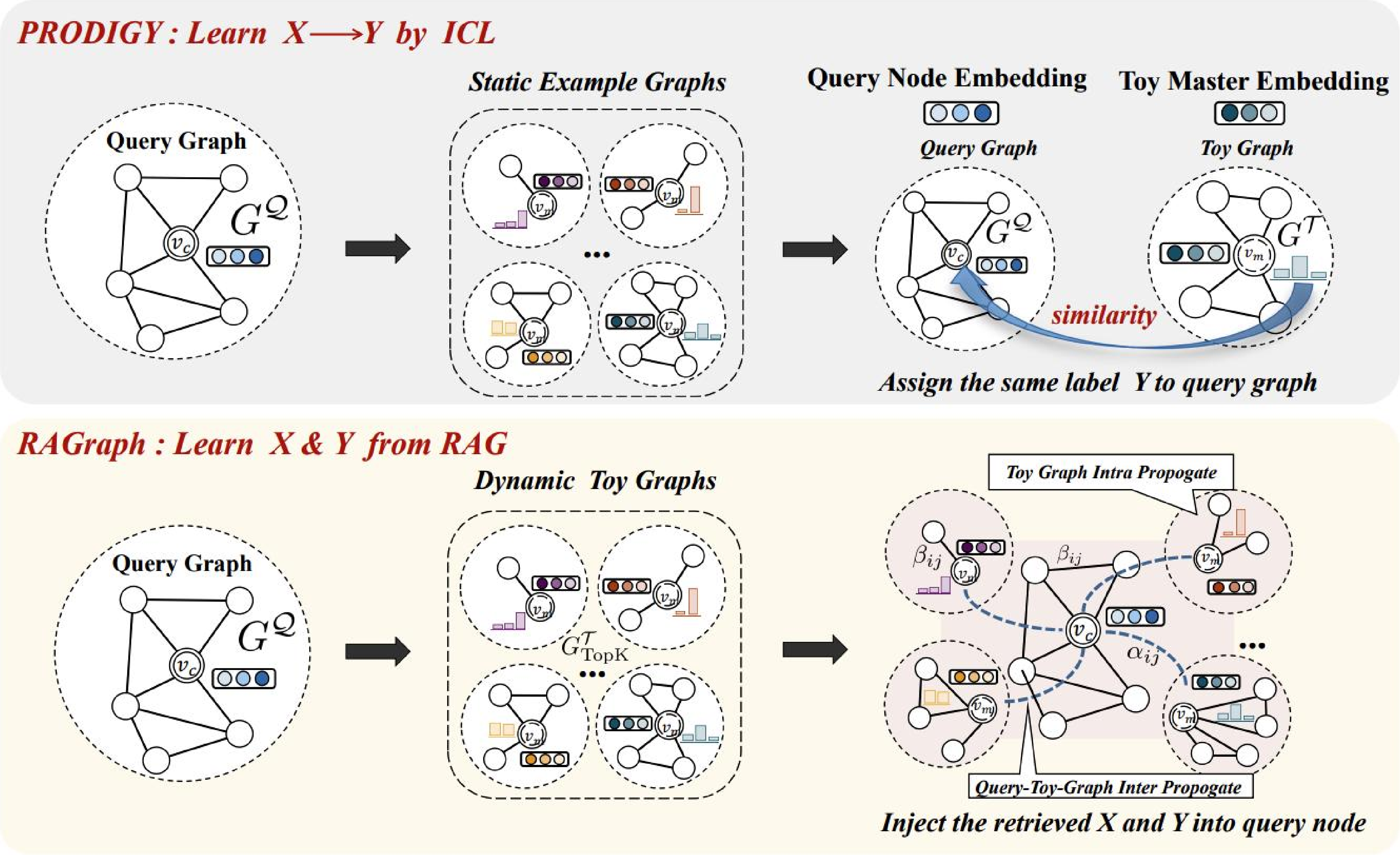
- **Results**

Table 1: Accuracy evaluation on node and graph classification. All tabular results standard deviation across five seeds run, with best **bolded** and runner-up underlined.

Methods	Node Classification		Graph Classification			
	PROTEINS (5-shot)	ENZYMES (5-shot)	PROTEINS (5-shot)	COX2 (5-shot)	ENZYMES (5-shot)	BZR (5-shot)
GCN	46.63±03.04	52.80±12.89	54.80±06.64	67.87±03.39	22.67±05.20	58.76±05.08
GraphSAGE	48.87±02.64	48.75±01.59	52.99±10.57	67.02±05.42	21.17±05.49	58.27±04.79
GAT	48.13±07.90	47.75±01.23	55.82±07.31	64.89±03.23	20.67±03.27	57.04±06.70
GIN	49.61±01.58	48.82±01.58	56.17±08.58	62.77±02.85	19.00±03.74	56.54±04.20
GraphPrompt+						
Vanilla/NF	44.88±13.17	48.81±01.88	56.68±03.63	53.04±04.13	36.50±03.31	68.77±03.44
Vanilla/FT	48.99±01.88	51.99±01.36	57.04±03.88	64.04±08.20	40.00±04.36	69.01±02.21
PRODIGY/NF	47.32±08.12	43.80±14.03	53.48±06.72	53.97±10.34	22.12±13.84	67.18±08.93
PRODIGY/FT	53.26±06.42	57.98±12.37	57.14±10.34	65.31±04.28	25.94±05.12	68.08±06.68
RAGRAPH/NF	56.12±04.11	75.92±01.72	58.48±03.93	55.32±04.15	38.17±03.39	**77.53**±05.26
RAGRAPH/FT	58.74±00.87	75.74±01.92	**62.33**±02.52	**76.60**±02.30	47.71±06.88	76.79±05.02
RAGRAPH/NFT	**59.83**±00.40	**76.23**±01.63	59.08±03.50	71.70±04.29	**49.17**±04.64	74.81±04.25

RAGraph shows outstanding performance and achieves state-of-the-art scores on almost all metrics

Related Work



Summary

- **Key contributions of RAGRAPH:**
  - First framework to integrate RAG with pre-trained GNNs
  - Significant improvement in generalization and task performance without fine-tuning

- **Future Work of RAGRAPH:**
  - Extending retrieval from subgraphs to more complex graph structures
  - More RAGraph Methods...

NEURAL INFORMATION PROCESSING SYSTEMS

北京大学 PEKING UNIVERSITY

Thank You!!

RAGRAPH: A General Retrieval-Augmented Graph Learning Framework

Presenters: Xinke Jiang, Rihong Qiu