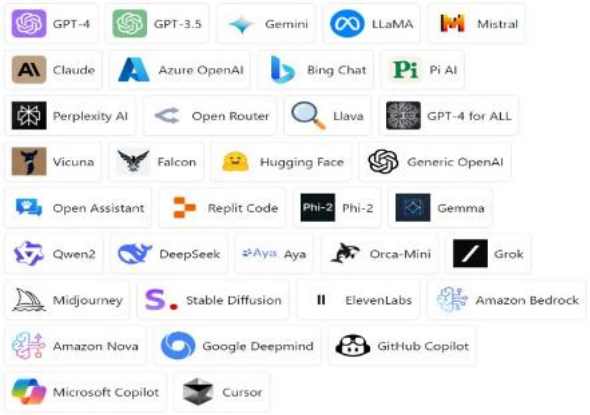
Towards Graph-native LLMs and Agentic Systems: Rethinking the AI Harness

Arijit Khan
Department of Computer Science
Bowling Green State University, USA
arijitk@bgsu.edu

Large Language Models



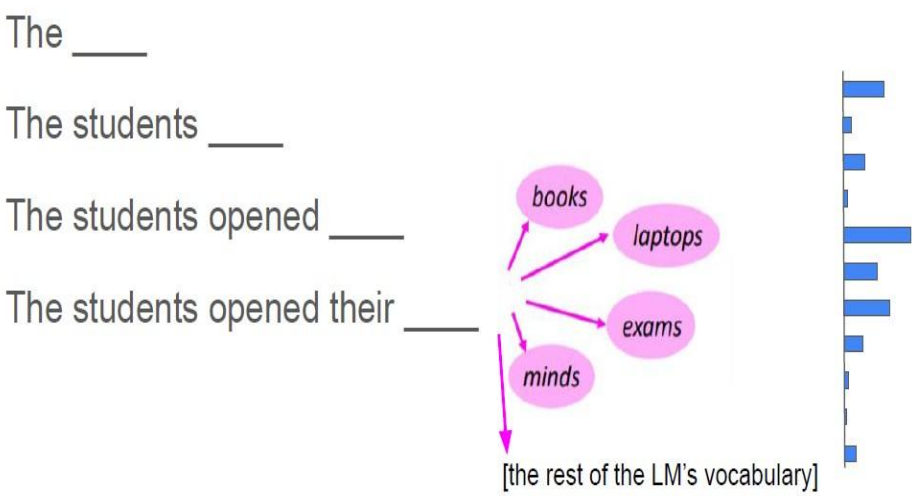
LLM models

- Models the probability of the next word, given the history (context) of preceding words.

$$p(w) = p(w_1) \times p(w_2|w_1) \times p(w_3|w_1, w_2) \times p(w_l|w_1, \ldots, w_{l-1})$$
$$= \prod_{t=1}^{|w|} p(w_t|w_1, \ldots, w_{t-1})$$

- **Generative AI (genAI):** AI system whose primary task is to generate content, e.g., GANs, VAEs, RNNs, LLMs, VLMs, VALL-E.
- **Large Language Models (LLMs):** Generative AI systems primarily designed for natural language processing tasks.
- **Foundation Models (FMs):** AI systems serving as the basis for a wide range of AI applications - can be adapted to a range of different, more specific purposes. E.g., LLMs, VLMs, speech FMs.

- often used interchangeably.



Language models text generation

LLMs – Benefits



Applications of LLMs

✔Emerging abilities; generalizing to unseen tasks; task descriptions provided as text (prompt).

✔Scaling the models, compute, and data leads in increase in performance.

✔Perform new and creative tasks using prompt-based Interaction and retrieval-augmented generation (RAG) without updating model parameters.

✔ LLM pipelines remove task-specific supervision and need for labeled data – easy to use, less expensive, and fast to iterate.

✔ LLMs act as knowledge bases (parametric knowledge) - can be probed for QA and querying tasks.

LLMs – Challenges

• **Alignment Problem:** LLMs may produce harmful, unsafe, toxic, or undesirable outputs –inappropriate language, misinformation, bias, and discrimination.

• **Hallucination:** Parametric knowledge, lack consistent representations of knowledge, fail to understand a question due to lack of context, knowledge gap (lack up-to-date and domain-specific knowledge), cannot recall facts (about not so popular or long-tail entities), output unreliable and incoherent responses, hallucinate by generating factually incorrect statements.

• **Lack of Consistency:** Generate logically contradicting outputs, low semantic similarity of LLM outputs due to paraphrased versions of a question (meaning-preserving text alternations), violate important relational properties such as negation, symmetry, and transitivity; Adversarial LLM Jailbreaks.

• **Privacy Concern:** Data privacy, personally identifiable information, data retention policy, IP leakage, security vulnerabilities, legal compliance.

• **Black-box Model:** Many LLMs are proprietary and little information is released about them. Difficult to explain LLM predictions with billions of parameters. Knowledge in LLMs is hard to interpret, update, and is prone to bias.

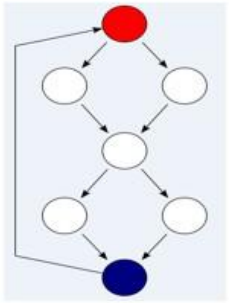
• **Environmental Concern:** High cost, energy consumption, carbon emissions, and water usage.

• **Societal Impacts:** Job loss, disparities, phishing, fraud, manipulation, plagiarism, cheating, fake news, big tech monopolies, societal unrest, …

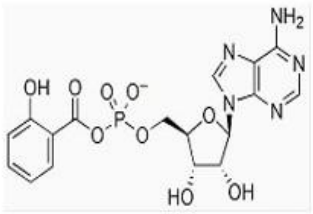
Graph Data is Everywhere

Graph database with many smaller graphs

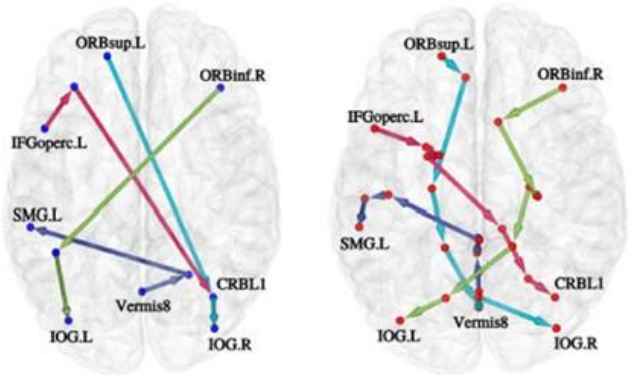
One large graph



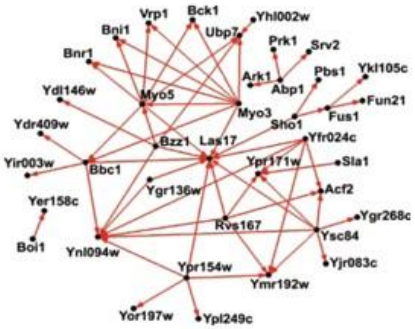
Program flow/ call graph



Chemical compound structure



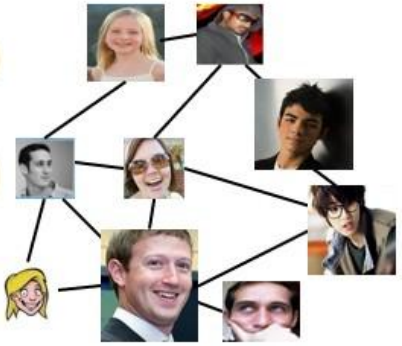
Human brain network



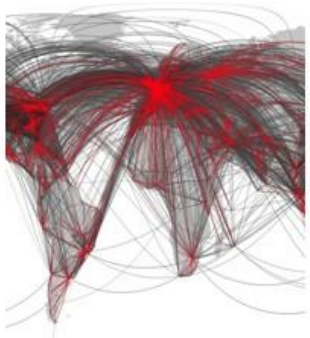
Biological network



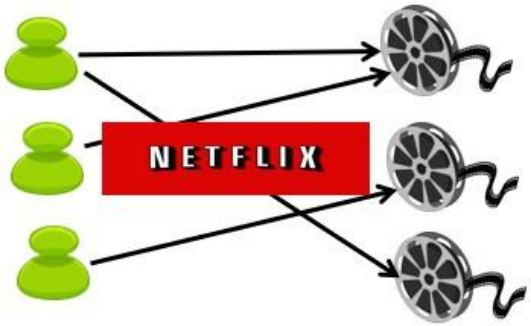
Financial transaction



Social network



Transportation



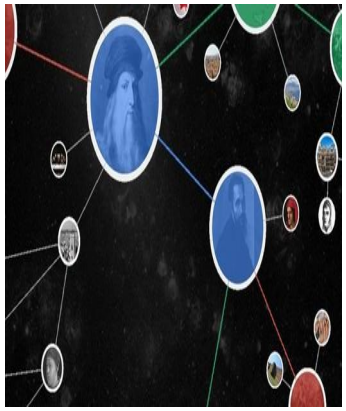
Recommendation



Knowledge graph

Knowledge Graphs

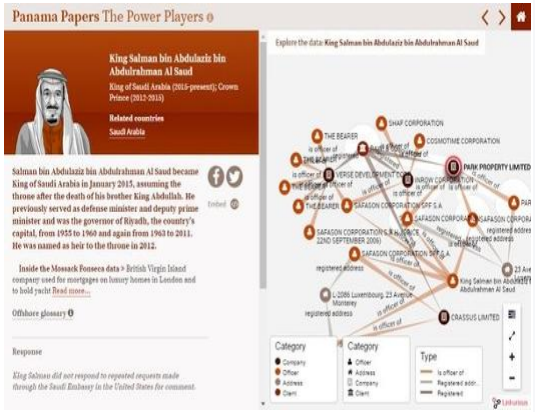
- Integrating knowledge + data at large scale = Knowledge graph



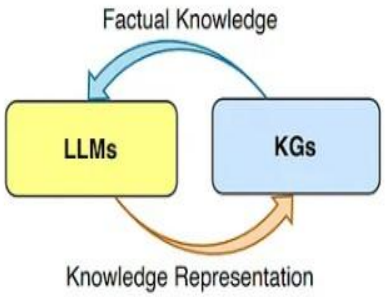
Google Knowledge Graph (2012)



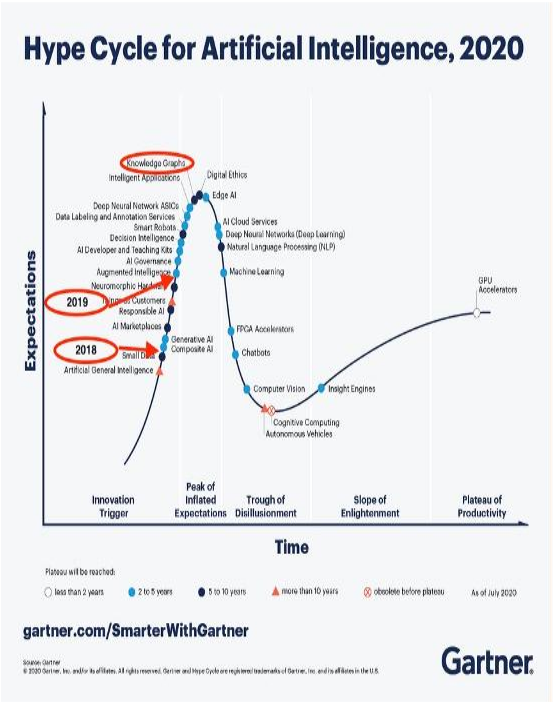
"People who like things I like" Facebook graph search (2013)



Panama Papers investigation, led by ICIJ, exposed highly connected networks of offshore tax structures used by world's richest elites (2016)



LLM + KG (2024 - present)



In 2020, Gartner put Knowledge Graphs at the peak of its AI hype cycle (2020)

KGs – Benefits and Challenges

KG Benefits

✔ structured, highly curated, and reliable representation of knowledge via explicit relationships.

✔support symbolic reasoning and inference, with answer validation and explainability.

✔ schema-flexible: updated dynamically with new knowledge via addition or deletion of triples/ nodes & edges.

✔ offer accurate explicit knowledge in many downstream applications, e.g., web search, QA, semantic search, personal assistants, fact-checking, and recommendation.

KG Challenges

• **Difficult to construct**.

• **Difficult to query** due to incompleteness, schema-flexibility, heterogeneity, and massive-scale.

• **User Barriers in SPARQL/Cypher-Style Querying:** non-professional users find it challenging to write an accurate query, e.g., via SPARQL, Cypher, Gremlin, GSQL, etc., since users must have full knowledge of the query language, schema, and the vocabulary used in a KG. Current KG querying approaches generally lack language understanding, are inadequate to deal with unseen entities and new facts, and often ignore multi-modal information in KGs.

• **Interoperability issue**: existing methods are tailored for specific KGs or downstream tasks.

LLMs+ KGs: Synergy

KG for LLM

✔ KGs offer external knowledge (up-to-date, domain specific, and symbolic knowledge) for enhancing the accuracy, consistency, transparency, and the overall capabilities of LLMs.

✔KG-enhanced pre-training, fine-tuning, and inference (KG-RAG).

✔KG-enhanced validation (LLM guardrail) and explainability.

LLM for KG

✔ LLMs augment KGs via knowledge extraction, auto-completion, and incorporating multi-modal information, enhancing usability and performance of downstream tasks with natural language understanding and generalization capabilities.

✔LLM-enhanced KG creation and completion.

✔LLM-enhanced KG embedding.

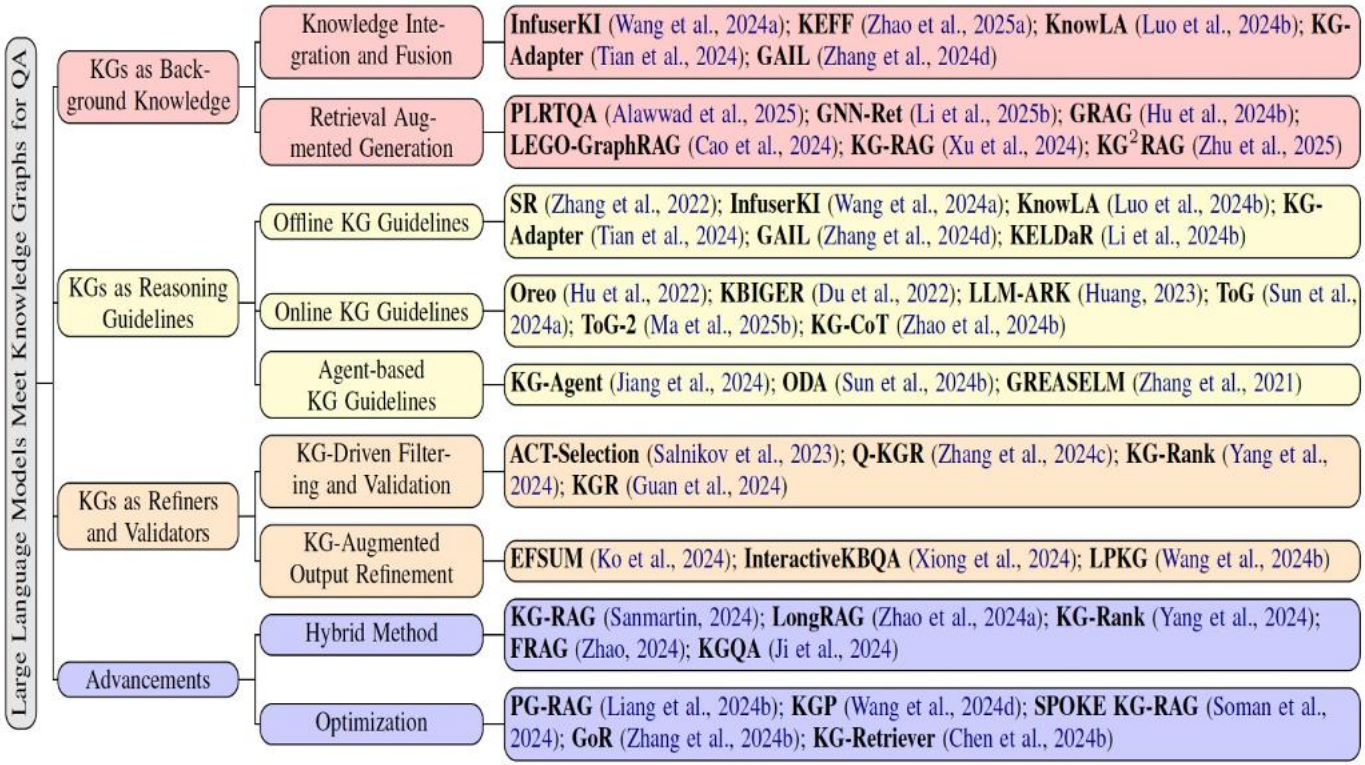
✔ LLM-enhanced KG querying, analytics, and domain-specific applications.

LLM+KG

✔ Downstream applications benefit from the complementarity of LLMs and KGs – LLMs and KGs offer parametric vs. explicit knowledge, respectively.

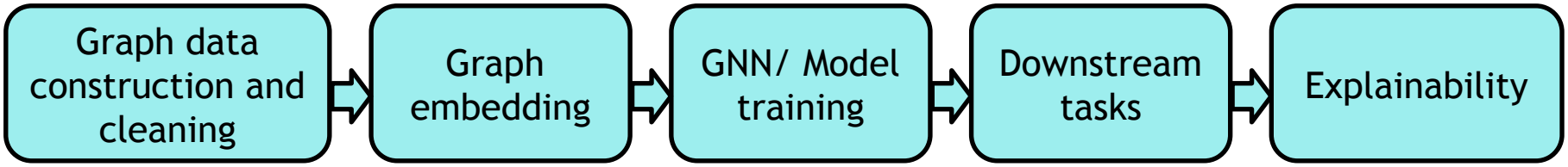
Role of KGs in Synthesizing with LLMs

- Our Recent Survey: LLMs Meet Knowledge Graphs for Question Answering: Synthesis and Opportunities https://github.com/machuangtao/LLM-KG4QA



Chuangtao Ma, Yongrui Chen, Tianxing Wu, **Arijit Khan**, and Haofen Wang, "Large Language Models Meet Knowledge Graphs for Question Answering: Synthesis and Opportunities", in EMNLP 2025 (Main).

AI/ML and (Graph) Data Science Pipeline



Graph AI/ML and data science pipeline

- **How graph AI/ML and graph data science benefit each other?**
- **Application of graph data science in graph AI/ML (DB4AI)**

- Scalable Graph Embedding and Graph Neural Network (GNN) Training

[PVLDB'23] *Distributed Graph Embedding with Information-Oriented Random Walks*

[PVLDB'26] *FeLoG: Scalable and Efficient Distributed Graph Embedding with Feedback Loop Mechanism*

✓ - Data Science Downstream Applications: LLM+KG

[ICLR'25] *Logical Consistency of Large Language Models in Fact-checking*

[In Submission] *Knowledge Graph-based Retrieval-Augmented Generation for Schema Matching*

- Graph Neural Network Explainability

[SIGMOD'24] *View-based Explanations for Graph Neural Networks*

[KDD'25] *Finding Counterfactual Evidences for Node Classification*

- **Application of graph AI/ML in graph data science (AI4DB)**

- Knowledge graph question answering

[ICDE'22] *Aggregate Queries on Knowledge Graphs: Fast Approximation with Semantic-aware Sampling*

[ICDE'20] *Semantic Guided and Response Times Bounded Top-k Similarity Search over Knowledge Graphs*

Our Work in Graph Data Science and AI/ML

Data Management for Emerging Problems in Large Networks

- ✓ **Big-Graphs: Querying, Mining, Streaming, Uncertainty, Systems, and Beyond**
- ✓ **User-friendly, efficient, and approximate querying and pattern mining using scalable algorithms, machine learning techniques, and distributed systems**

Knowledge-Graph Search [SIGMOD 11, VLDB 13, ICDE 12 Tutorial, ACM SIGMOD Blog Post, Encyclopaedia of Big Data Technologies, ICDE 20, 22, KBS 22, CIKM 22, SIGMOD Record 23]

Graph Query-By-Example [ICDE 14, TKDE 15]

Graph Stream Summarization and Query [SNAM 17]

Spatial/ Road Network Query [TKDE 20, SIGMOD 20]

Scalable, Approximate Graph Querying

Reliability and Shortest Path [EDBT 14, VLDB 15 Tutorial, TKDE 18, 20, Book @Morgan&Claypool, VLDB 19, VLDB 21, TKDE 22, DSAA 24]

Influence Maximization and Embedding [SDM 11, ICDE 16, CIKM 17, ICDE 18, SIGMOD 18, ICDE 23, VLDB Journal 25]

Densest subgraph [ICDE 23]

Uncertain Graph Processing

Novel Pattern Mining [SIGMOD 10, VLDB 20]

Graph Anomaly Detection [CIKM 12]

Graph-based Entity Resolution, Correlation Clustering [CIKM 17, VLDB 18, CIKM 26]

Graph Summary, Core Decomposition, Multi-layer graph, and Hypergraph [SIGMOD 19, VLDB 17 Tutorial, TKDD 22, VLDB 23]

Complex Graph Mining

Information-centric Distributed Graph Embedding [VLDB 23, TKDE 24, VLDB 26]

Complementary Graph Partitioning [SIGMOD 12]

Graph Storage Decoupling and Smart Query Routing [USENIX ATC 18]

Vertex-Centric Graph Processing [VLDB 14 Tutorial, EDBT 17]

Distributed Graph Systems

Graph Neural Network Explainability [BigData 20, DSAA Tutorial 23, SIGMOD 24, ICDE 24, KDD 25, SIGMOD Record 25, ICDE 26, ICDM 26, WSDM Tutorial 26, WebConf 26, ICLR 26]

Graph Machine Learning

Large Language Models [ICLR 25, EDBT 25, ACM SIGMOD Blog Post, EMNLP25, SIGMOD 26, PAKDD Tutorial 26, TCDE Bulletin 26, VLDB 26, VLDB Journal 25, EMNLP 26, TKDD 26, SIGMOD 27] ✓

LLMs for Data Science

Relational Data [SIGMOD 14]
Stream Data [SIGMOD 16]
Time Series Data [AIME 26]
Healthcare Data [J. American Medical Informatics Assoc.]
Blockchain Network [WebConf 20, 21, WSDM 22, CIKM 22 Tutorial, ACM DLT 25, IJCAI 26]
Crowd-Sourcing [CIKM 17, VLDB 18]

Other Big Data Processing

Arijit Khan – all papers available at https://www.cs.bgsu.edu/arijitk/publication.html

LLM+KG: Our Work

- **Survey of LLM+KG for Question Answering (QA)** (EMNLP 2025 Main Survey, EDBT 2025 Tutorial)

- **LLM+KG for Logic Consistency and Fact Checking** (ICLR 2025)

- **LLM+KG for CoT-RAG** (EMNLP 2025 Findings)

- **LLM for Entity Resolution** (SIGMOD 2026)

- **LLM+KG for Schema Matching** (TKDD 2026)

- **LLM+KG for Entity Matching** (SIGMOD 2027)

- **LLM Agent Tool Calling Benchmark** (under submission)

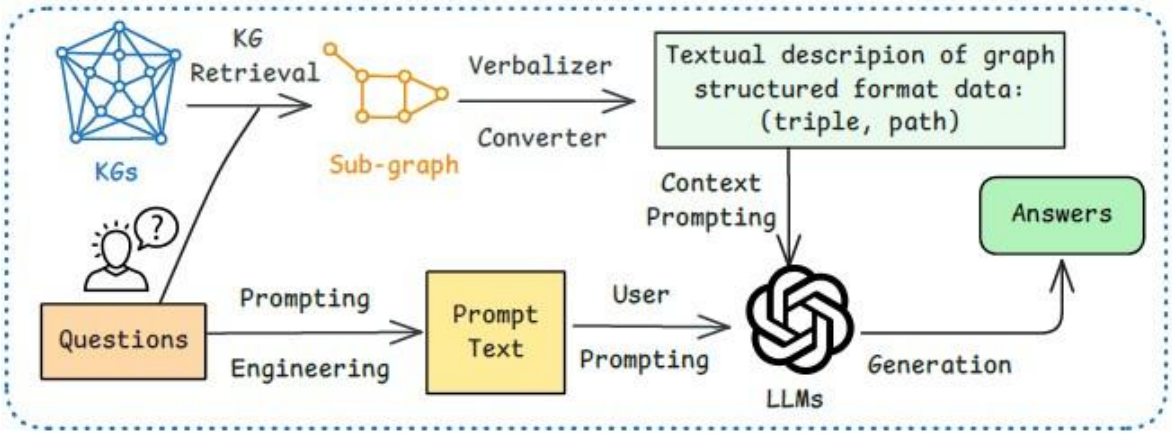
- **LLMs+Agents for Blockchain Security and Analytics** (IJCAI 2026)

- **Diffusion Model for Time Series Informative Missingness** (AIME 2026)

Graph-based Retrieval Augmented Generation

- **Retrieval-augmented generation (RAG)** to use KG context in order to improve a large language model (LLM)'s accuracy and consistency.

- **GraphRAG:**
  Graph-based retrieval-augmented generation



LLM-based Query Processing with KG Context
KG- augmented Prompting
KG-RAG

- **GraphRAG vs KG-RAG:**

✓ GraphRAG retrieves subgraphs from a graph that was built based on a textual document.

✓ KG-RAG augments LLMs with relevant subgraphs or triples retrieved from curated and large-scale knowledge graphs.

Logical Consistency of Large Language Models in Fact-checking

[ICLR 2025]

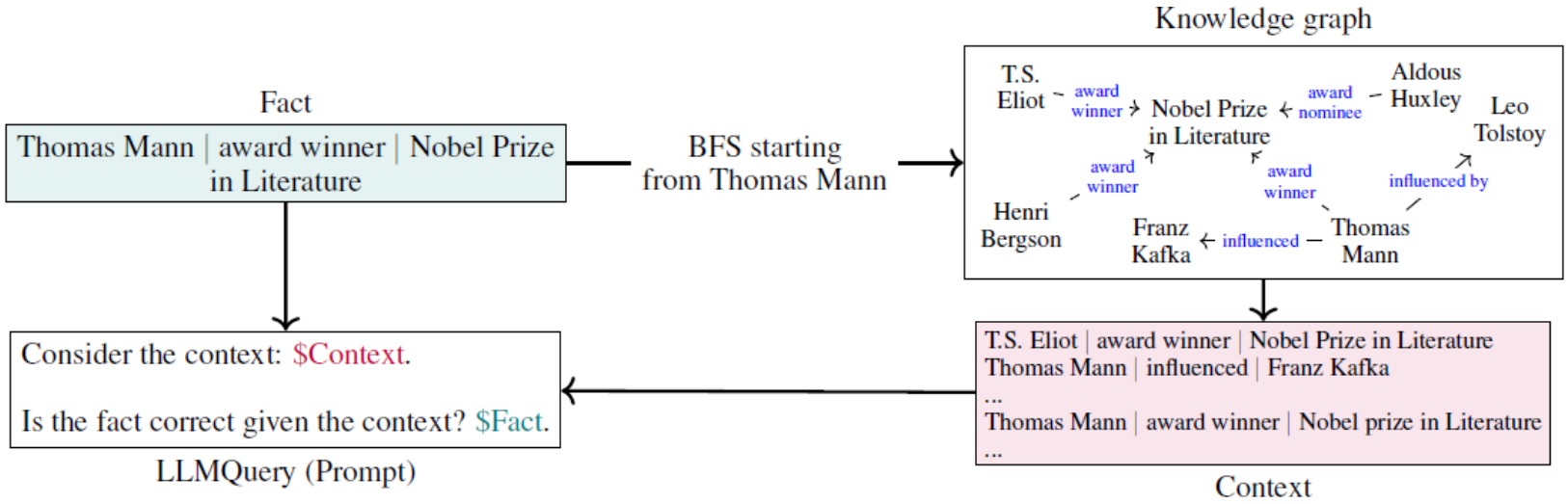


Figure 1: Our LLM-based fact-checking framework with the KG context. The LLMQuery shown is an example of zero-shot prompt, that is, it does not contain examples or demonstrations.

- What is their consistency over logically equivalent changes in the input query?
- How to improve an LLM's consistency under logically equivalent perturbation in input query?

Datasets	Fact types and Rules	Training	Evaluation	Test
FreebaseLFC	$p, \neg p, p \wedge q, p \vee q$	1K $(\times 4)$	5K $(\times 4)$	5K $(\times 4)$
	$p \wedge (q \vee r), p \vee (q \wedge r), p \vee q \leftrightarrow q \vee p, p \wedge q \leftrightarrow q \wedge p$	—	5K $(\times 4)$	5K $(\times 4)$
NELLLFC	All above fact types and rules	—	5K $(\times 8)$	5K $(\times 8)$
WikiLFC	$p, \neg p$	—	5K $(\times 2)$	5K $(\times 2)$

Logical Consistency of Large Language Models in Fact-checking

[ICLR 2025]

Table 1: Accuracy and logical consistency of LLMs before and after fine-tuning (FT). Bold numbers denote an improved result in accuracy and consistency for fine-tuning. Training is performed on FreebaseLFC dataset only (marked with '$*$'), while performance improved in all datasets.

			Accuracy		Logical Consistency	
Model	**Dataset**	**Fact**	**Before FT**	**After FT**	**Before FT**	**After FT**
Llama2-13B	FreebaseLFC*	$p, \neg p$	0.90	**0.93**	0.81	**0.86**
		$p \wedge q$	0.61	**0.93**	0.67	**0.83**
		$p \vee q$	0.73	**0.76**	0.73	**0.97**
	NELLLFC	$p, \neg p$	0.88	**0.97**	0.76	**0.93**
		$p \wedge q$	0.38	**0.89**	0.69	**0.88**
		$p \vee q$	0.73	**0.76**	0.73	**0.94**
	WikiLFC	$p, \neg p$	0.96	**0.96**	0.92	**0.93**
Llama2-7B	FreebaseLFC*	$p, \neg p$	0.87	**0.97**	0.76	**0.94**
		$p \wedge q$	0.39	**0.87**	0.47	**0.86**
		$p \vee q$	0.78	**0.81**	0.83	0.77
	NELLLFC	$p, \neg p$	0.71	**0.94**	0.44	**0.88**
		$p \wedge q$	0.26	**0.81**	0.71	**0.91**
		$p \vee q$	0.75	0.73	0.95	0.78
	WikiLFC	$p, \neg p$	0.90	**0.90**	0.80	**0.81**
Gemma-2B	FreebaseLFC*	$p, \neg p$	0.82	**0.98**	0.66	**0.96**
		$p \wedge q$	0.45	**0.83**	0.70	**0.84**
		$p \vee q$	0.73	**0.94**	0.91	0.90
	NELLLFC	$p, \neg p$	0.81	**0.94**	0.62	**0.89**
		$p \wedge q$	0.33	**0.83**	0.78	**0.83**
		$p \vee q$	0.75	**0.88**	0.98	0.87
	WikiLFC	$p, \neg p$	0.76	**0.90**	0.52	**0.80**
Average			0.69	**0.88**	0.74	**0.88**

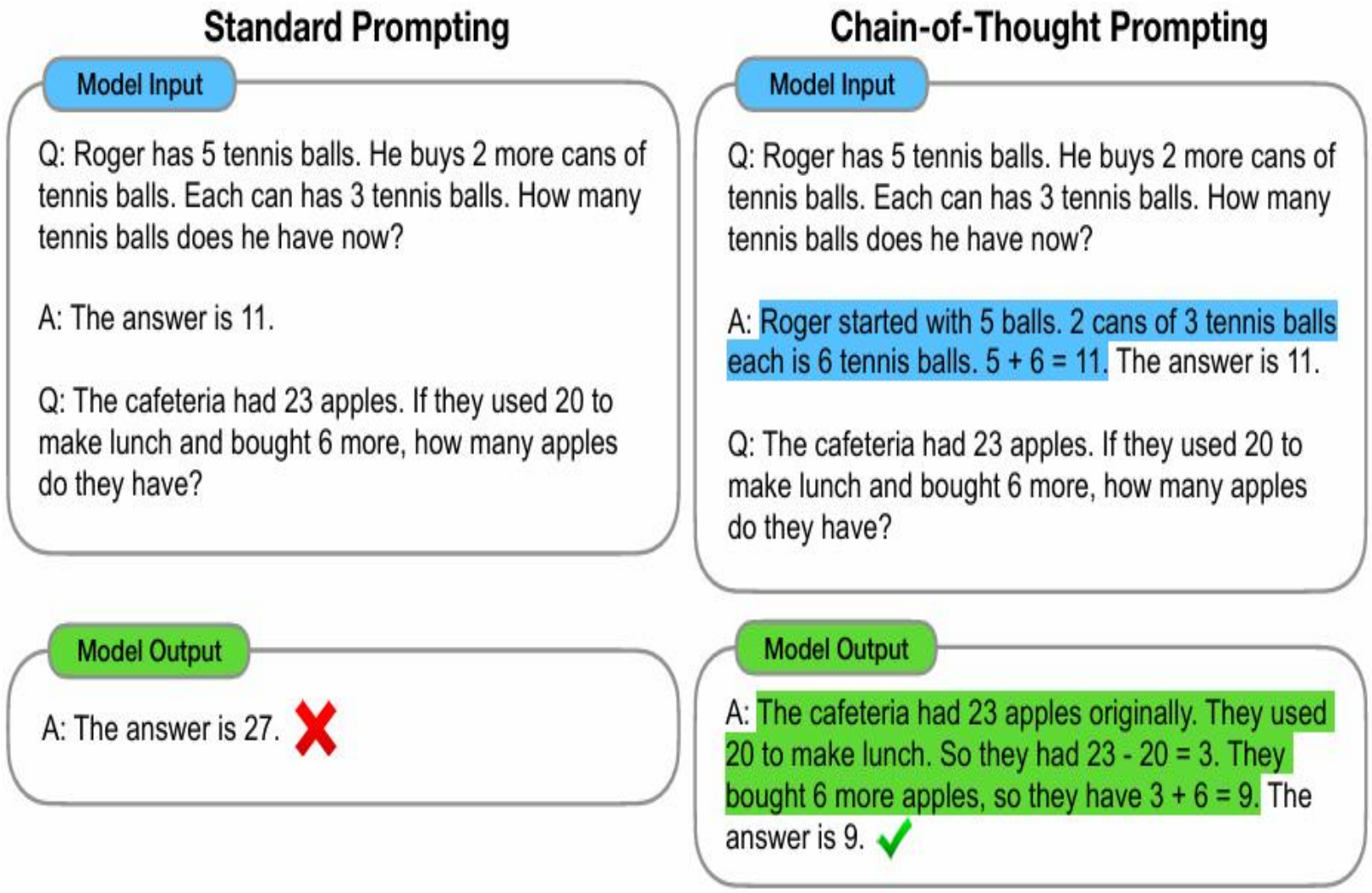
Logical Consistency of Large Language Models in Fact-checking

[ICLR 2025]

- **Assessment.** LLMs are not always logically consistent with their generation – consistency decreases as the query complexity increases.
- **Improvement.** Instruction prompting is not sufficient to improve logical consistency in LLMs: smaller models require instruction fine-tuning, while larger models may suffice with instruction prompting.
  - **Generalization.** Fine-tuning for logical consistency in one dataset can generalize to other datasets and queries with more logical operators.
- **Benchmark.** Fact-checking in KG provides a flexible benchmark to test LLMs on logical queries of varying complexity.

ICLR 2025: Bishwamittra Ghosh, Sarah Hasan, Naheed Anjum Arafat, and **Arijit Khan**, "Logical Consistency of Large Language Models in Fact-checking"

COT-RAG (EMNLP 2025): Background



《Chain-of-Thought Prompting Elicits Reasoning in Large Language Models》 by Google in NeurIPS 2022

Motivation： The low reliability of relying solely on LLMs to generate reasoning chains [EMNLP 2025]

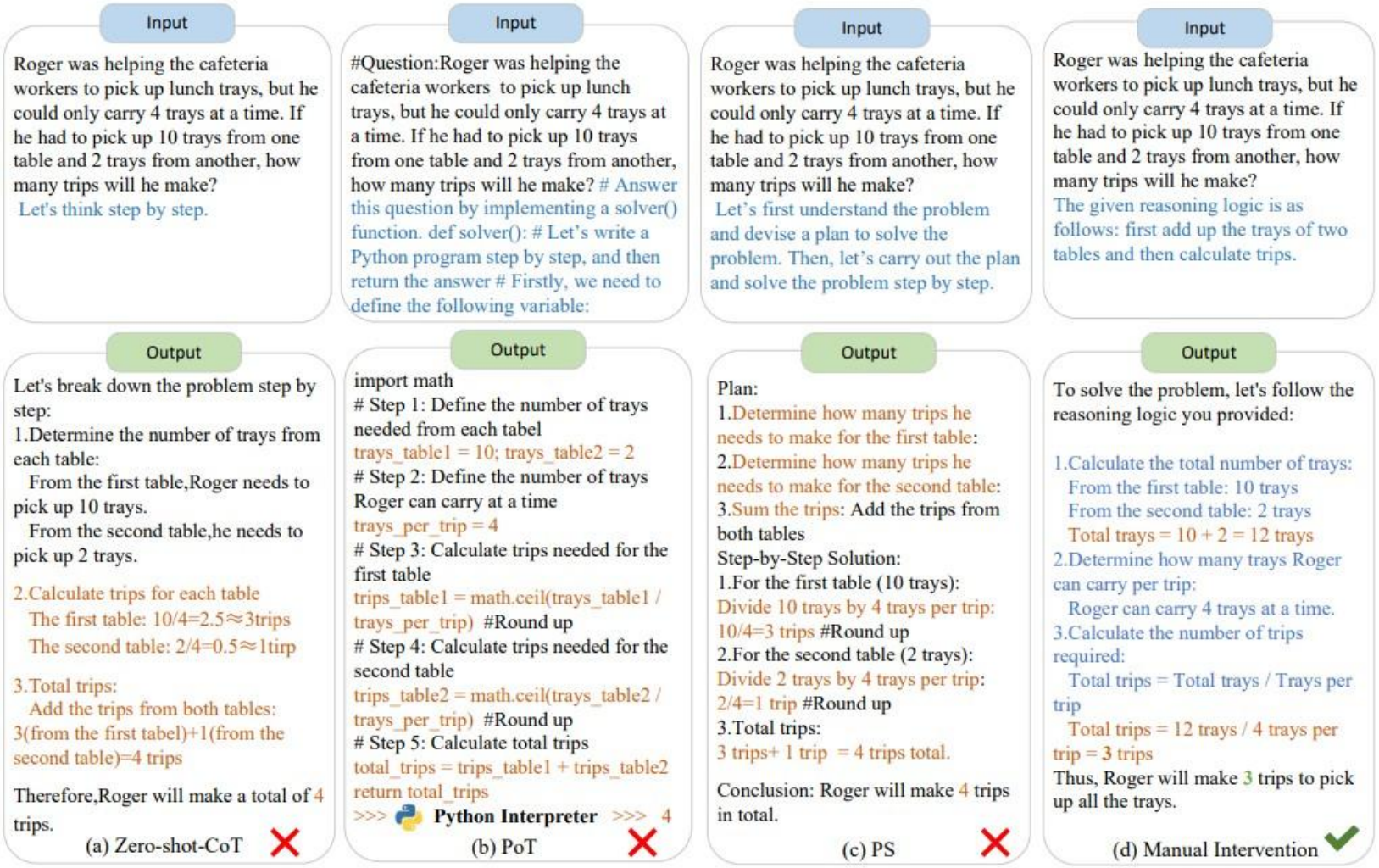
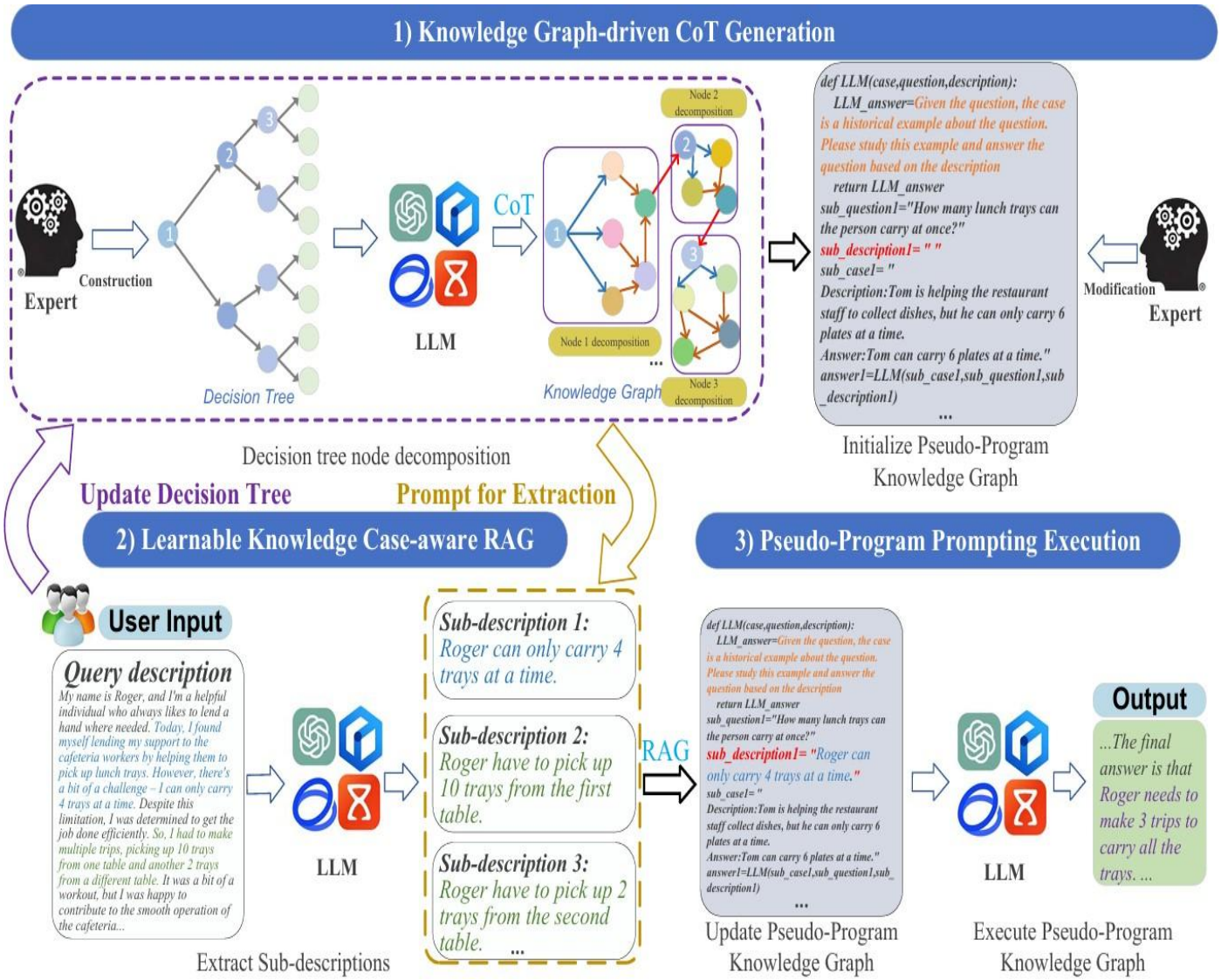


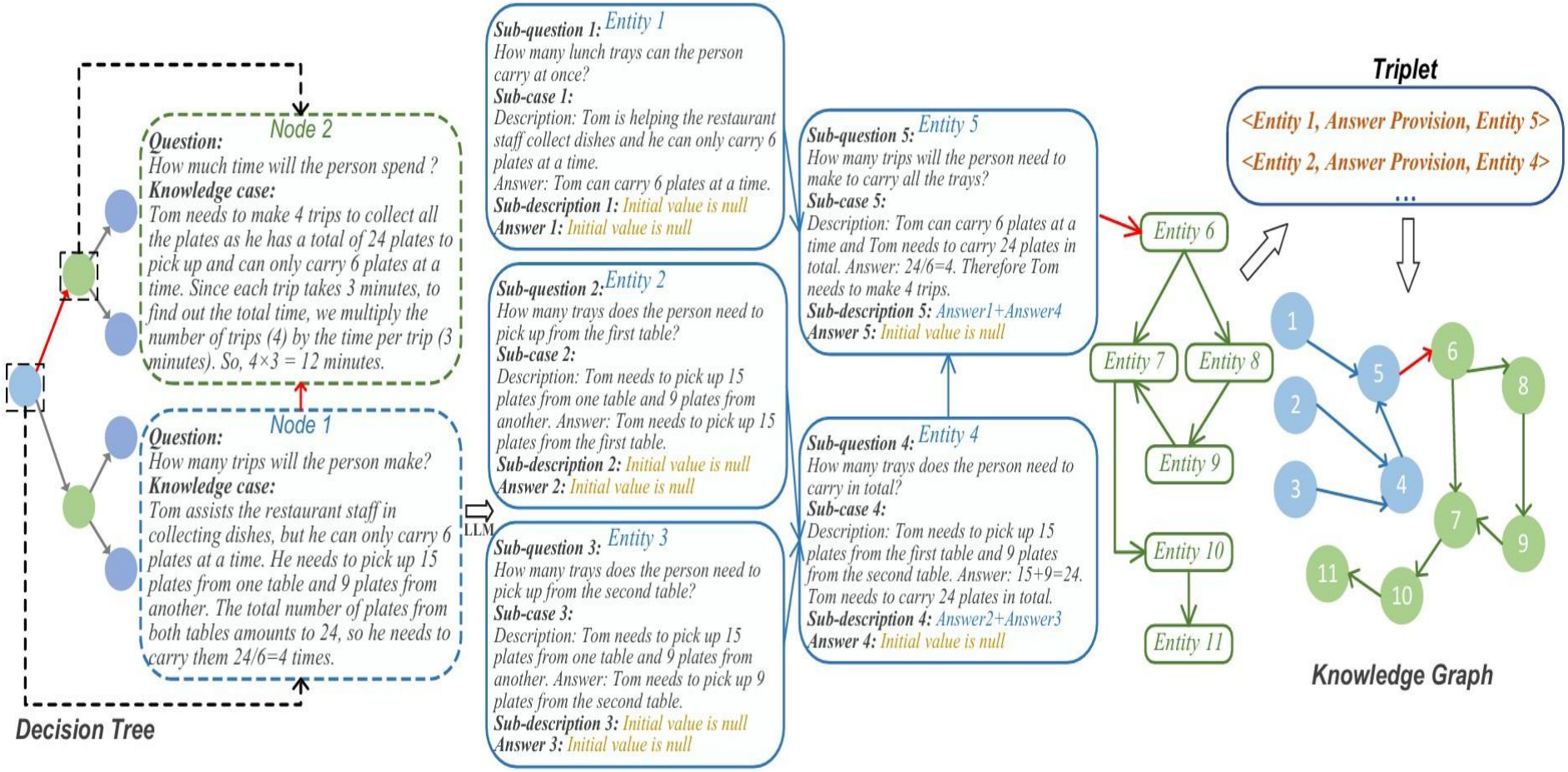
Figure 1: Example inputs and outputs of GPT-4o mini with (a) Zero-shot-CoT (Kojima et al., 2022), (b) PoT (Chen et al., 2023), (c) PS (Wang et al., 2023a) and (d) Manual Intervention on MultiArith (Roy and Roth, 2015).

CoT-RAG: A reasoning framework integrating Chain-of-Thought (CoT) and Retrieval-Augmented Generation (RAG) [EMNLP 2025]



Knowledge Graph-driven CoT Generation

[EMNLP 2025]

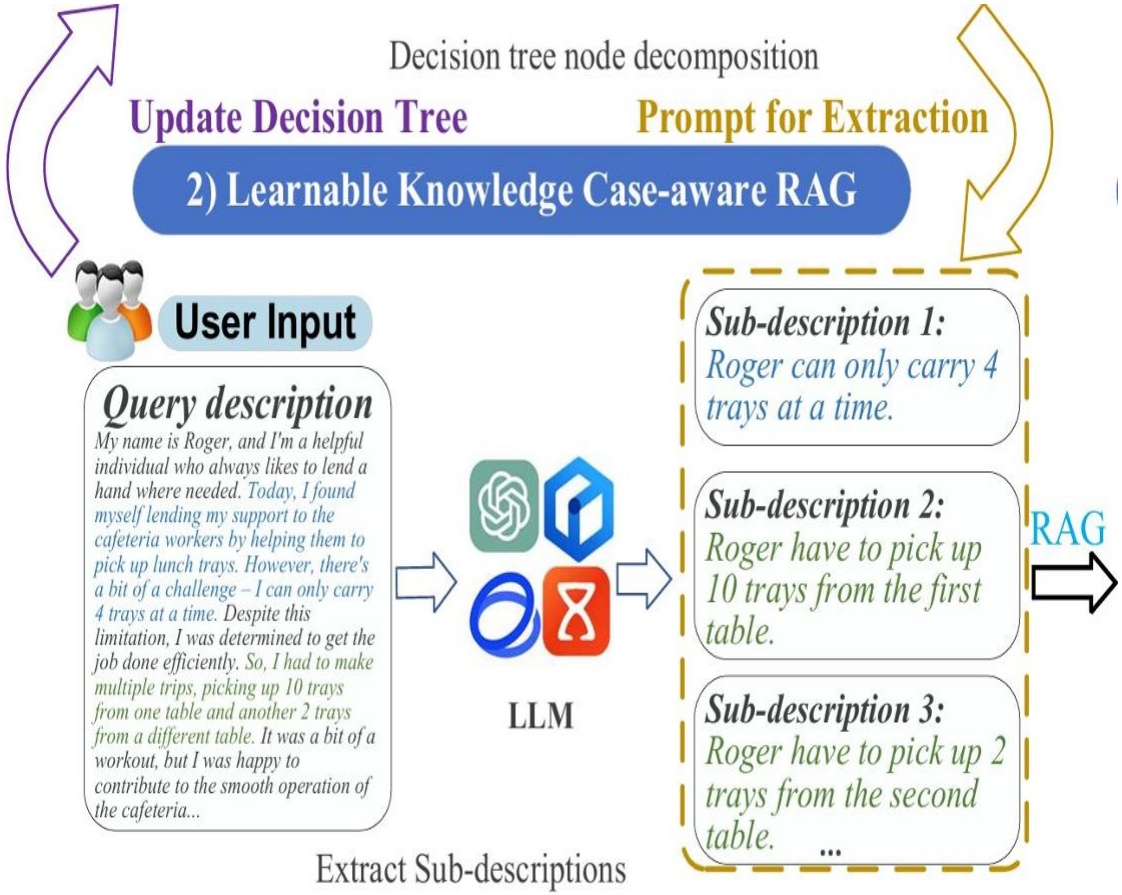


Key Innovations：

- Structuring LLM reasoning with expert-defined, coarse-grained decision trees
- Decomposing trees into fine-grained knowledge graphs to model complex relationships

Learnable Knowledge Case-aware RAG

[EMNLP 2025]



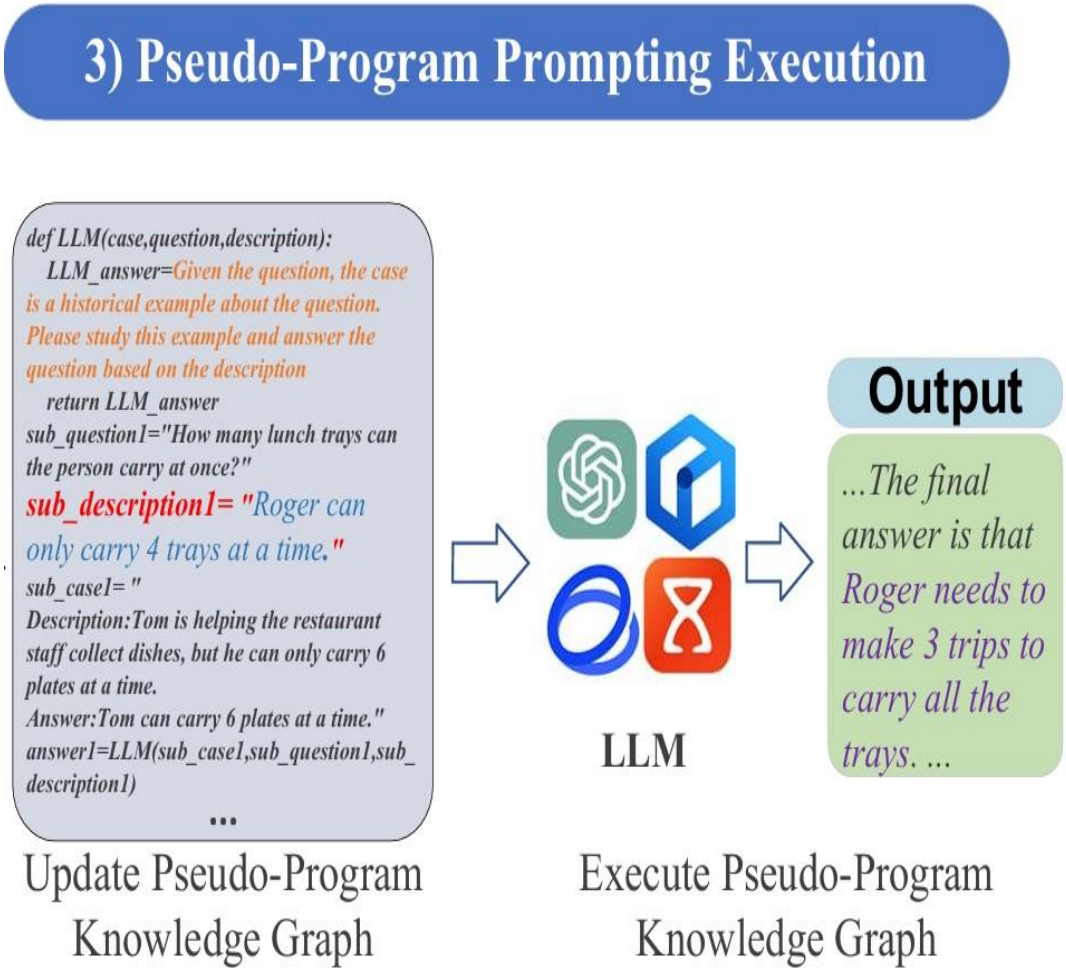
Key Innovations：

- Extracting sub-descriptions from user queries by retrieving relevant sub-cases from the knowledge graph
- Updating the decision tree dynamically based on user input to optimize the knowledge graph

EMNLP 2025 (Findings): Feiyang Li, Peng Fang, Zhan Shi, **Arijit Khan**, Fang Wang, Weihao Wang, Xin Zhang, and Yongjian Cui, "CoT-RAG: Integrating Chain of Thought and Retrieval-Augmented Generation to Enhance Reasoning in Large Language Models"

Pseudo-Program Prompting Execution

[EMNLP 2025]



Key Innovations：

- Replaces natural language with structured pseudo-code prompts to boost the logicality of LLM-generated reasoning chains
- Leverages the clear structure and broad applicability of pseudo-program

Experimental Results [EMNLP 2025]

Method	AQuA	GSM8K	MultiArith	SingEq	HotpotQA	CSQA	SIQA	Letter	Coin	Average
Without external knowledge and exemplars										
Zero-shot	42.6	77.8	95.9	86.8	80.1	74.5	77.3	35.8	76.7	71.9
Zero-shot-CoT	43.4	78.3	96.7	88.5	81.4	75.6	78.0	34.5	75.6	72.4
PS	50.1	82.8	96.9	89.4	83.0	74.2	76.3	44.7	79.5	75.2
QDMRPS	47.3	83.8	95.2	90.7	86.7	76.6	77.8	36.5	76.3	74.5
With exemplars										
Manual-CoT	54.3	85.8	97.2	92.3	85.7	79.6	82.4	39.6	79.2	77.3
Auto-CoT	47.8	82.4	97.5	91.6	86.1	76.4	80.6	41.0	81.2	76.1
Complex-CoT	51.7	83.5	96.6	92.8	82.8	76.9	78.5	37.7	81.9	75.8
Iter-CoT	51.6	80.0	97.8	93.4	64.8	76.9	77.3	41.8	77.5	73.5
ZEUS	51.9	88.4	97.3	92.8	84.9	77.4	81.7	42.8	82.5	77.7
Pattern-CoT	52.8	85.3	97.7	91.3	82.5	76.3	78.9	41.4	83.7	76.7
With external knowledge										
KD-CoT	22.3	68.4	76.0	62.3	79.9	85.6	90.8	24.6	58.3	63.1
IRCoT	20.7	65.6	78.3	65.1	87.5	82.8	87.9	28.2	54.5	63.4
KG-CoT	12.3	66.8	78.6	61.5	73.5	88.9	92.1	26.7	53.2	61.5
With both external knowledge and exemplars										
CoT-RAG (ours)	**65.7**	**94.7**	**98.5**	**98.7**	**98.4**	**97.9**	**98.7**	**54.6**	**94.7**	**89.1**

CoT-RAG significantly outperforms the existing **CoT** methods on 9 datasets in the fields of arithmetic, common sense, and symbols, with an accuracy improvement of 4.0% - 44.3%

Experimental Results

[EMNLP 2025]

Method	LaB	LeB	CFB	AGI	Average
Graph-form LLM-based RAG methods					
GraphRAG	94.8	97.5	73.1	54.6	80.0
KG-CoT	89.3	93.6	72.8	32.6	72.1
ToG	86.7	90.2	68.3	64.2	77.4
PoG	93.8	91.7	89.5	45.3	80.1
RRKG	91.3	92.4	74.7	27.4	71.5
RoG	90.4	88.1	88.7	67.5	83.7
Graph-CoT	54.7	68.2	63.1	24.7	52.7
ToG-2	92.5	93.7	76.6	70.8	83.4
AtomR	83.6	82.5	77.3	37.5	70.2
Variants of CoT-RAG					
CoT-RAG (IndexFlatL2)	92.7	93.7	85.2	67.8	84.9
CoT-RAG (IndexFlatIP)	91.5	91.9	87.1	72.4	85.7
CoT-RAG (IndexIVFFlat)	93.6	92.1	86.8	75.3	87.0
CoT-RAG (IndexLSH)	90.8	92.5	88.1	74.6	86.5
CoT-RAG (IndexPQ)	93.1	92.9	87.4	73.9	86.8
CoT-RAG (IndexIVFPQ)	93.6	93.1	87.8	76.2	87.7
CoT-RAG (Zero-expert)	93.6	94.7	86.3	74.8	87.4
CoT-RAG (ours)	**99.3**	**98.6**	**94.7**	**88.3**	**95.2**

CoT-RAG significantly outperforms existing **Graph form LLM based RAG** methods on four datasets across three vertical domains of law, finance, and logic, with an accuracy improvement of 8.9% -40.6%

Summary [EMNLP 2025]

Motivation:

① The low reliability of relying solely on LLMs to generate reasoning chains

② The poorer reasoning performance from natural language prompts compared with code prompts

Method:

A reasoning framework integrating Chain-of-Thought (CoT) and Retrieval-Augmented Generation (RAG)

- ✓ **Key Technique 1**： Knowledge Graph-driven CoT Generation
- ✓ **Key Technique 2**： Learnable Knowledge Case-aware RAG
- ✓ **Key Technique 3**： Pseudo-Program Prompting Execution

Experimental Results：

Compared with CoT methods, CoT-RAG has increased accuracy by 4.0% -44.3%, and compared with Graph form LLM based RAG methods, accuracy has increased by 8.9% -40.6%

Key Achievement：

- ✓ CoT-RAG: Integrating Chain of Thought and Retrieval-Augmented Generation to Enhance Reasoning in Large Language Models. **EMNLP Findings 2025**

Entity Resolution Background

[SIGMOD 2026]

- Identifying records referring to the **same real-world entity**.
- Fundamental for data quality and system integration.
- Traditional methods focus on **Pairwise Comparisons**.
- Challenge: $O(n^2)$ complexity leads to massive time and monetary costs.

SIGMOD 2026: Jiajie Fu, Haitong Tang, **Arijit Khan**, Sharad Mehrotra, Xiangyu Ke, and Yunjun Gao, "In-context Clustering-based Entity Resolution with Large Language Models: A Design Space Exploration"

Why LLMs for Entity Resolution?

[SIGMOD 2026]

The core advantage: deciding whether A ?= B in difficult cases — zero-shot, with pretrained world knowledge.

Field	A	B
Title	Database System Concepts, 7th Ed.	DB System Concepts (Seventh Edition)
Authors	A. Silberschatz, H. F. Korth, B. S. Sudarshan	Silberschatz Abraham; Korth Henry F; Sudarshan S
Publisher	McGraw-Hill	McGraw Hill Education
Year	2019	'19
Pages	1376 pp.	1.376 k pages

LLMs can decide A = B without:

- ❌ **hand-crafted matching rules**
- ❌ **task-specific fine-tuning (required by deep-learning / PLM methods)**
- ❌ **aggressive data cleaning**

— all of which are otherwise expensive.

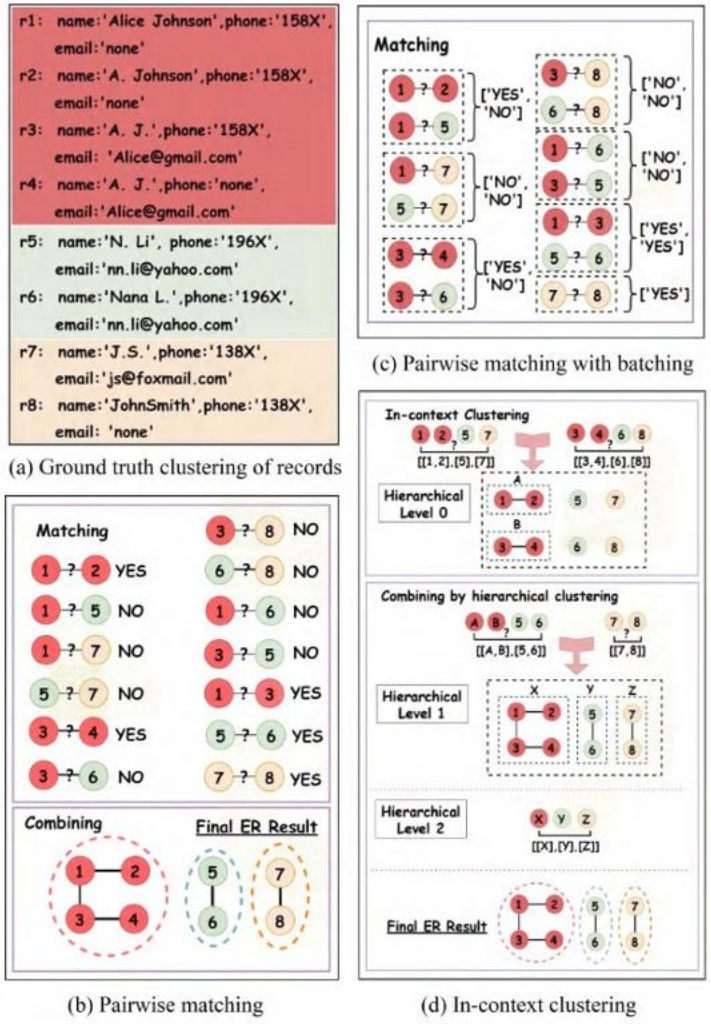
🤖 **LLM → ✅ A = B**

LLMs unlock zero-shot, rule-free, finetune-free ER for messy real-world records.

Pairwise vs. Clustring-based ER

[SIGMOD 2026]

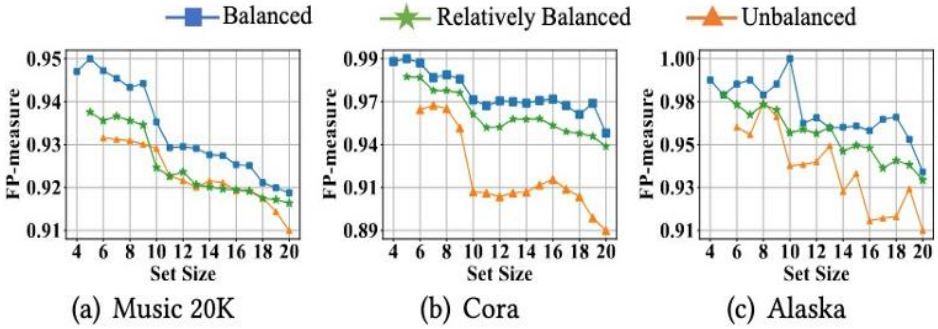
- **Pairwise:** $O(n^2)$ API calls. Asymptotically $O(n^2)$ even with transitivity and blocking.
- **Clustering:** Process sets of records directly.
- LLMs can cluster multiple records in one prompt.

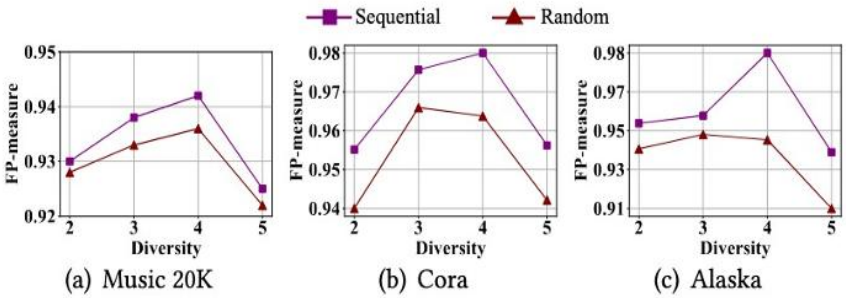


Design Space Exploration [SIGMOD 2026]

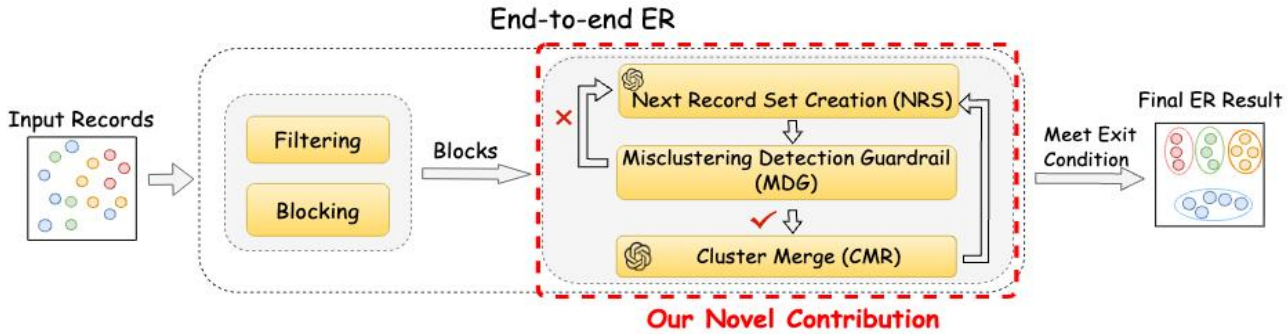
- **Set Size (S_s):** Number of records per prompt.
- **Set Diversity (S_d):** Number of distinct entities.
- **Set Variation (S_v):** Imbalance in cluster sizes.
- **Record Order:** Sequence of records within the prompt.

Research Goal: Optimize these factors for cost-effective ER.





LLM-CER Framework [SIGMOD 2026]



- An effective **Next Record Set Creation (NRS) algorithm** for constructing optimal record sets for in-context clustering from initial blocks generated by standard blocking techniques;

- A guardrail technique, **Misclustering Detection Guardrail (MDG)** for assessing and subsequently improving the accuracy of the LLM's in-context clustering outputs;

- an efficient heuristic algorithm, hierarchical **Cluster Merge (CMR)** for merging the in-context clustering outputs.

Evaluation [SIGMOD 2026]

Metrics	Cora		Alaska		AS	
	$S_s = 2$ (*pairwise*)	$S_s = 9$ (*clustering*)	$S_s = 2$ (*pairwise*)	$S_s = 9$ (*clustering*)	$S_s = 2$ (*pairwise*)	$S_s = 9$ (*clustering*)
ACC	0.88	**0.90**	0.81	**0.82**	**0.70**	**0.70**
FP-measure	0.67	**0.71**	0.78	**0.79**	0.60	**0.63**
Cost (USD)	0.67	**0.03**	0.43	**0.15**	0.08	**0.02**
Tokens (M)	3.45	**0.12**	2.29	**0.73**	0.35	**0.07**
Time (min)	297.27	**5.42**	241.31	**39.57**	77.2	**8.01**
# API Calls (K)	30.23	**0.28**	24.54	**2.04**	7.85	**0.41**

Pairwise vs. ours

Dataset	Metrics	LLM-CER	Booster	BQ	CrowdER+LLM
Alaska	ACC	**0.82**	0.71	0.33	0.68
	FP	**0.79**	0.55	0.49	0.62
	Cost ($)	0.15	**0.02**	1.55	0.42
	Tokens (M)	0.73	**0.19**	5.59	2.04
	Time (s)	**2374.2**	2450.1	8798.9	6547.2
	# API Calls	**2043**	2606	8035	5845
AS	ACC	**0.70**	0.62	0.54	0.52
	FP	**0.63**	0.62	0.51	0.50
	Cost ($)	0.02	**0.01**	0.29	0.11
	Tokens (M)	0.07	**0.03**	0.34	0.37
	Time (s)	**480.6**	622.9	925.5	2356.2
	# API Calls	**413**	723	842	2084
Song	ACC	**0.72**	0.52	0.59	0.52
	FP	**0.78**	0.68	0.67	0.64
	Cost ($)	0.06	**0.02**	0.77	0.12
	Tokens (M)	0.22	**0.11**	1.98	0.43
	Time (s)	**933.2**	903.3	2581.5	1856.3
	# API Calls	**668**	921	2338	1247
Music	ACC	**0.71**	0.59	0.60	0.62
	FP	**0.61**	0.60	0.54	0.55
	Cost ($)	0.19	**0.02**	2.18	0.39
	Tokens (M)	0.90	**0.15**	8.96	1.82
	Time (s)	**2388.4**	2585.1	17515.8	4562.3
	# API Calls	**3859**	3915	17365	7782
DG	ACC	**0.81**	0.56	0.62	0.72
	FP	**0.70**	0.68	0.63	0.65
	Cost ($)	0.07	**0.02**	1.12	0.34
	Tokens (M)	0.37	**0.18**	3.92	1.79
	Time (s)	**1552.4**	2552.2	6052.2	7456.3
	# API Calls	**1285**	3085	6456	6504
Cora	ACC	**0.90**	0.75	0.62	0.51
	FP	**0.71**	0.60	0.56	0.61
	Cost ($)	0.03	**0.01**	1.45	0.07
	Tokens (M)	0.12	**0.06**	4.23	0.29
	Time (s)	**325.5**	605.4	4085.3	598.5
	# API Calls	**279**	698	4882	483
Cite-seer	ACC	**0.88**	0.72	0.64	0.60
	FP	**0.95**	0.78	0.79	0.69
	Cost ($)	0.03	**0.01**	0.63	0.08
	Tokens (M)	0.13	**0.05**	1.64	0.37
	Time (s)	**1360.8**	1585.2	6228.9	3895.6
	# API Calls	**1302**	2169	6420	3858
Amazon-Google	ACC	**0.71**	0.58	0.53	0.50
	FP	**0.64**	0.55	0.50	0.48
	Cost ($)	0.02	**0.01**	0.62	0.09
	Tokens (M)	0.07	**0.03**	0.86	0.42
	Time (s)	**465.6**	785.2	1658.2	1985.2
	# API Calls	**452**	998	1895	2025
Walmart-Amazon	ACC	**0.61**	0.50	0.42	0.51
	FP	**0.56**	0.48	0.41	0.50
	Cost ($)	0.02	**0.01**	0.59	0.08
	Tokens (M)	0.06	**0.03**	0.68	0.39
	Time (s)	**375.8**	475.2	1498.5	3895.6
	# API Calls	**398**	825	1585	1958

End-to-end Performance

Comparison with PLM Methods [SIGMOD 2026]

Dataset	Metrics	Ours	Ditto [45]			DeepMatcher [52]		
			w/ FT (20%)	*w/ FT (80%)*	*w/o FT*	*w/ FT (20%)*	*w/ FT (80%)*	*w/o FT*
Alaska	ACC	**0.82**	0.70	0.81	0.64	0.65	0.74	0.58
	FP	**0.79**	0.62	0.77	0.55	0.51	0.70	0.43
	Cost ($)	0.15	65.67	260.21	**0.12**	65.81	260.93	0.14
AS	ACC	0.70	0.60	**0.72**	0.58	0.57	0.65	0.54
	FP	0.63	0.53	**0.66**	0.51	0.49	0.61	0.47
	Cost ($)	**0.02**	7.06	28.85	0.08	7.18	29.29	0.07
Song	ACC	**0.72**	0.61	0.70	0.54	0.59	0.66	0.53
	FP	**0.78**	0.63	0.76	0.58	0.52	0.72	0.51
	Cost ($)	**0.06**	13.74	41.34	0.09	12.82	38.65	0.15
Music	ACC	0.71	0.63	**0.74**	0.57	0.59	0.66	0.55
	FP	0.61	0.53	**0.65**	0.44	0.52	0.56	0.47
	Cost ($)	0.19	79.30	239.22	0.22	79.08	238.54	**0.14**
DG	ACC	**0.81**	0.68	0.80	0.60	0.67	0.77	0.60
	FP	**0.70**	0.60	0.69	0.52	0.53	0.66	0.50
	Cost ($)	**0.07**	12.93	42.04	0.13	13.01	42.22	0.16
Cora	ACC	**0.90**	0.76	**0.90**	0.67	0.71	0.88	0.66
	FP	0.71	0.56	**0.72**	0.48	0.54	0.70	0.50
	Cost ($)	**0.03**	10.71	42.92	0.07	10.74	43.08	0.06
Cite-seer	ACC	0.88	0.76	**0.89**	0.62	0.60	0.73	0.59
	FP	**0.95**	0.70	0.93	0.65	0.67	0.91	0.59
	Cost ($)	**0.03**	9.62	38.32	0.08	9.73	39.04	0.09
AG	ACC	0.71	0.61	**0.73**	0.56	0.58	0.69	0.57
	FP	0.64	0.57	**0.68**	0.50	0.54	0.64	0.49
	Cost ($)	**0.02**	7.13	28.84	0.07	7.16	28.04	0.05
WA	ACC	0.61	0.56	**0.65**	0.48	0.50	0.60	0.44
	FP	0.56	0.45	**0.60**	0.40	0.44	0.57	0.39
	Cost ($)	**0.02**	6.98	28.03	0.06	6.89	27.96	0.06

Ours vs. PLM baselines

[45] Yuliang Li, Jinfeng Li, Yoshihiko Suhara, AnHai Doan, and Wang-Chiew Tan. 2020. Deep entity matching with pre-trained language models. Proc. VLDB Endow. 14, 1 (2020), 50–60.

[52] Sidharth Mudgal, Han Li, Theodoros Rekatsinas, AnHai Doan, Youngchoon Park, Ganesh Krishnan, Rohit Deep, Esteban Arcaute, and Vijay Raghavendra. 2018. Deep Learning for Entity Matching: A Design Space Exploration. In Proceedings of the 2018 International Conference on Management of Data. 19—-34.

Cost-Efficient RAG for Entity Matching (CE-RAG4EM) [SIGMOD 2027]

- **KG+LLM for domain-specific entity matching https://arxiv.org/abs/2602.05708**
- **Challenges:** The knowledge grounding gap in LLM-based entity matching and costly retrieval and inference overhead in directly adapting vanilla RAG to entity matching in large-scale data integration.
- **Motivation**: We propose a blocking-based batch preprocessing strategy that groups similar records into the same batch, reducing redundant retrieval and inference for efficient RAG-based entity matching.

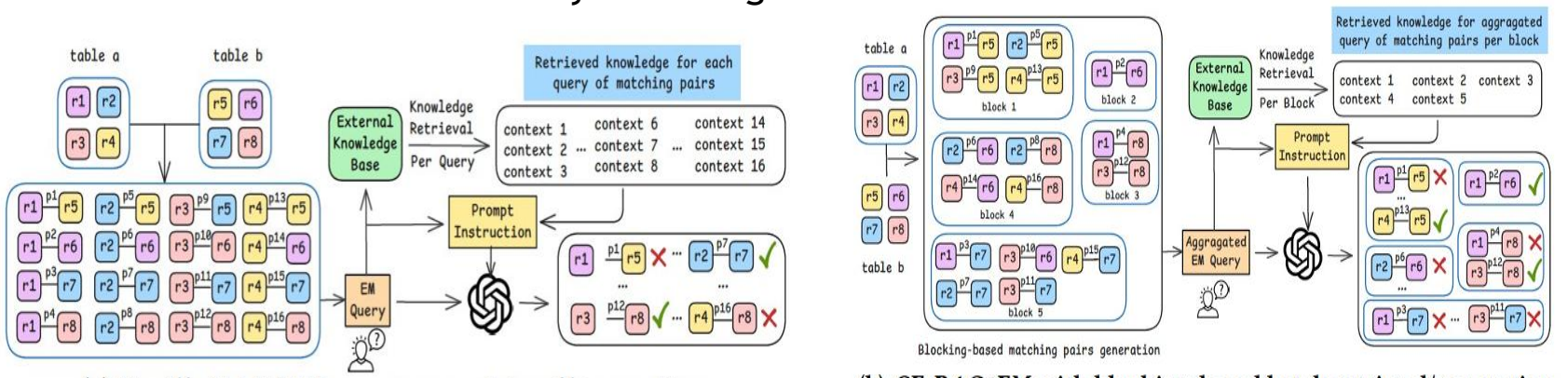


Figure : Vanilla RAG4EM vs. CE-RAG4EM. Matched records share the same color.

Cost-Efficient RAG for Entity Matching

[SIGMOD 2027]

- **Method**: We introduce a cost-efficient RAG framework for zero-shot entity matching that reduces computation through blocking-based batch retrieval and generation.

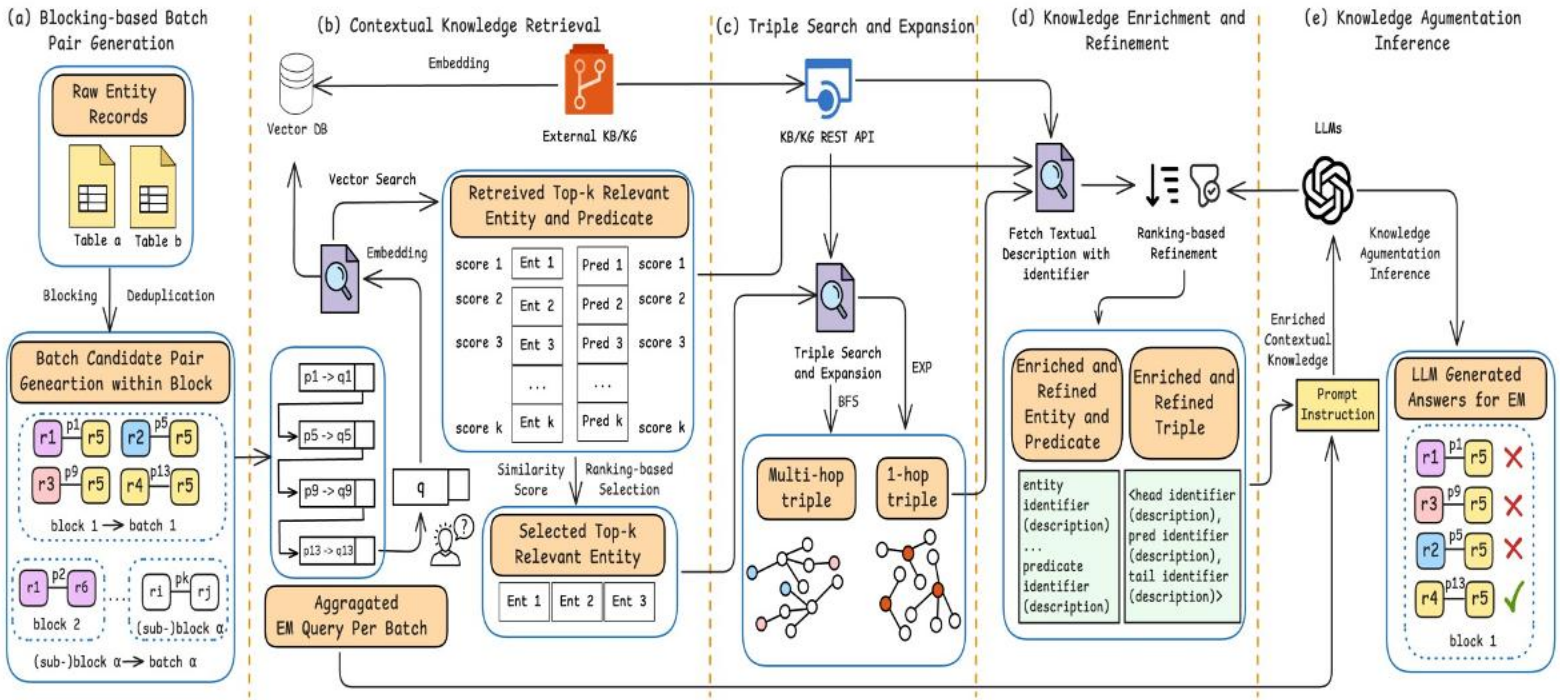


Figure: Overview of CE-RAG4EM. The framework consists of five phases (a)–(e). A check mark indicates that the LLM predicts Yes for the corresponding EM query, whereas a cross mark indicates No.

Cost-Efficient RAG for Entity Matching

[SIGMOD 2027]

- **Method**: We design a space exploration of CE-RAG4EM is structured around two core dimensions: blocking-based optimization and retrieval granularity.

	Retrieval Granularity (RG)	
(Batch Optimization) BO	*Entity & Predicate*	*Triple*
Batch Retrieval (BR)	CE-RAG4EM-BR	CE-KG-RAG4EM-BR
Batch Generation (BG)	CE-RAG4EM-BG	CE-KG-RAG4EM-BG
BR & BG	CE-RAG4EM-BR-BG	CE-KG-RAG4EM-BR-BG

Table: Summary of the Design Solutions of CE-RAG4EM.

SIGMOD 2027: Chuangtao Ma, Zeyu Zhang, **Arijit Khan**, Sebastian Schelter, and Paul Groth, "Cost-Efficient RAG for Entity Matching with LLMs: A Blocking-based Exploration"

CE-RAG4EM [SIGMOD 2027]

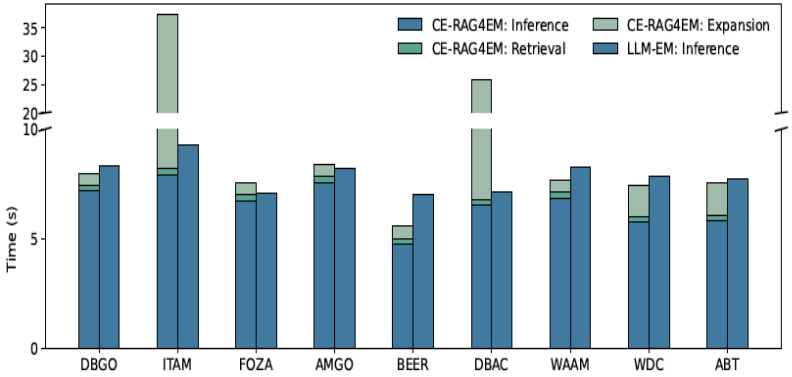
- **Results**: Extensive experiments on nine benchmark datasets show that CE-RAG4EM achieves comparable or improved matching quality while substantially reducing end-to-end overhead.

Dataset	F1 (%)		Precision (%)		Recall (%)	
	CE-RAG4EM	LLM-EM	*CE-RAG4EM*	LLM-EM	*CE-RAG4EM*	LLM-EM
DBGO	80.77 (+24.22)	65.02	92.45 (-0.73)	93.13	71.71 (+43.47)	49.98
ITAM	72.61 (+12.70)	64.43	97.62 (+0.18)	97.44	58.03 (+20.52)	48.15
FOZA	83.11 (+10.34)	75.32	100.00 (0.00)	100.00	71.21 (+17.50)	60.61
AMGO	55.47 (+14.02)	48.65	51.40 (-17.60)	62.37	60.26 (+55.00)	38.89
BEER	73.49 (+8.73)	67.59	96.67 (+9.32)	88.43	59.52 (+8.70)	54.76
DBAC	81.87 (+7.12)	76.43	95.14 (-1.05)	96.15	71.85 (+13.26)	63.44
WAAM	74.85 (+5.84)	70.72	84.59 (-1.13)	85.56	67.18 (+14.08)	58.89
WDC	73.35 (+5.13)	69.77	81.74 (-2.07)	83.47	66.53 (+11.01)	59.93
ABT	78.72 (+3.24)	76.25	91.27 (-2.35)	93.47	69.26 (+7.21)	64.60

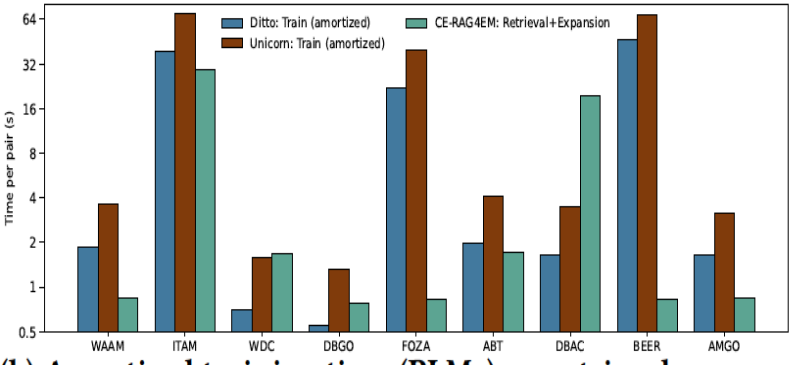
Overall Performance *CE-RAG4EM vs. LLM-EM*

Dataset	F1 (%)			Precision (%)			Recall (%)		
	CE-RAG4EM	Ditto	Unicorn	*CE-RAG4EM*	Ditto	Unicorn	*CE-RAG4EM*	Ditto	Unicorn
WAAM	74.85	56.50	61.47	84.59	64.32	70.99	67.18	50.37	68.01
ITAM	72.61	64.33	67.45	97.62	67.32	68.12	58.03	61.59	66.79
WDC	73.35	45.16	70.03	81.74	49.89	70.24	66.53	41.26	69.83
DBGO	80.77	77.62	78.06	92.45	81.23	81.71	71.71	74.31	74.72
FOZA	83.11	69.64	82.61	100.00	89.16	90.13	71.21	57.13	76.25
ABT	78.72	67.50	78.72	91.27	64.71	89.41	69.26	59.43	70.32
DBAC	81.87	82.96	88.72	95.14	87.65	92.12	71.85	78.76	85.57
BEER	73.49	84.19	82.20	96.67	87.33	84.39	59.52	81.27	80.13
AMGO	55.47	53.86	68.86	51.40	56.70	67.97	60.26	51.29	69.77

Overall Performance *CE-RAG4EM vs. PLM-EM*



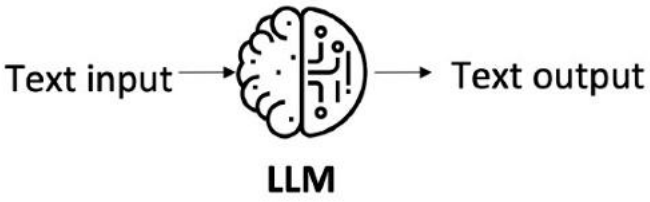
(a) End-to-end matching time per entity pair.

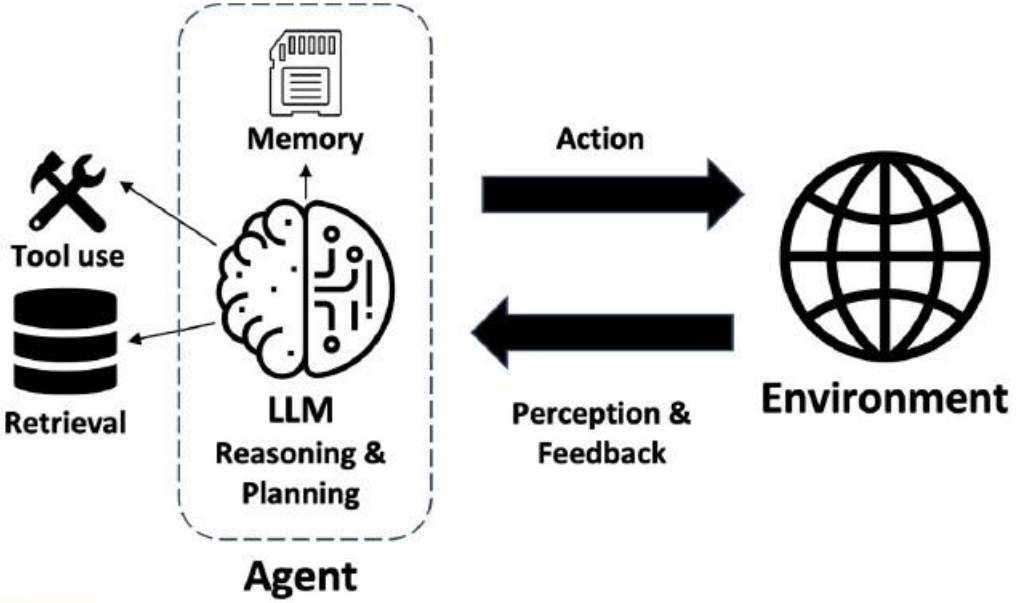


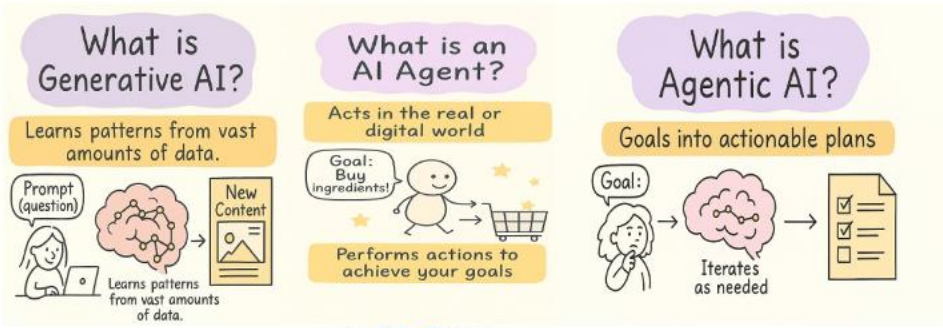
(b) Amortized training time (PLMs) vs. retrieval + expansion time (*CE-RAG4EM*).

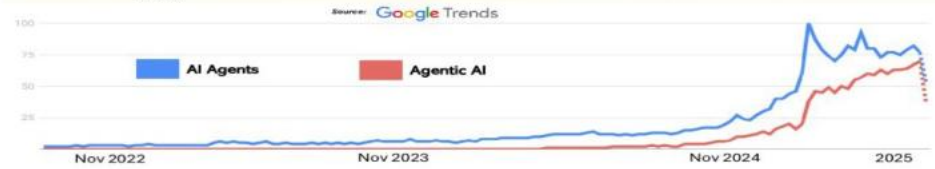
Efficiency comparison of CE-RAG4EM against (a) LLM-EM and (b) PLM baselines.

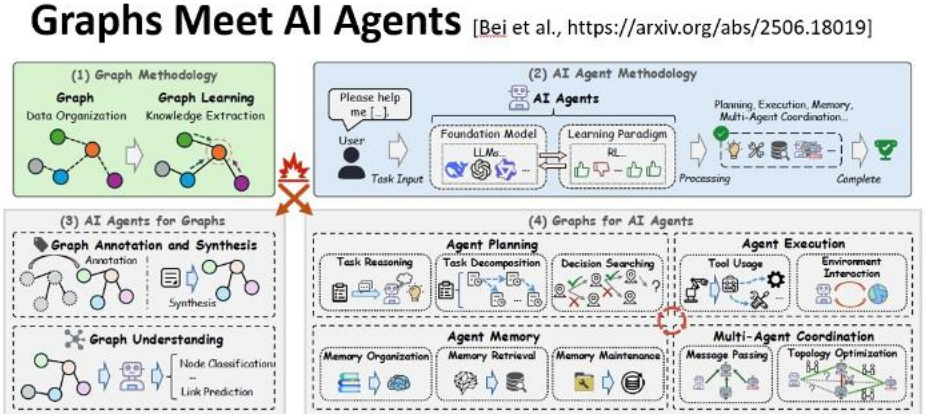
AI Agents


Text input
Text output
LLM


Memory
Tool use
Retrieval
LLM
Reasoning &
Planning
Agent
Action
Perception &
Feedback
Environment

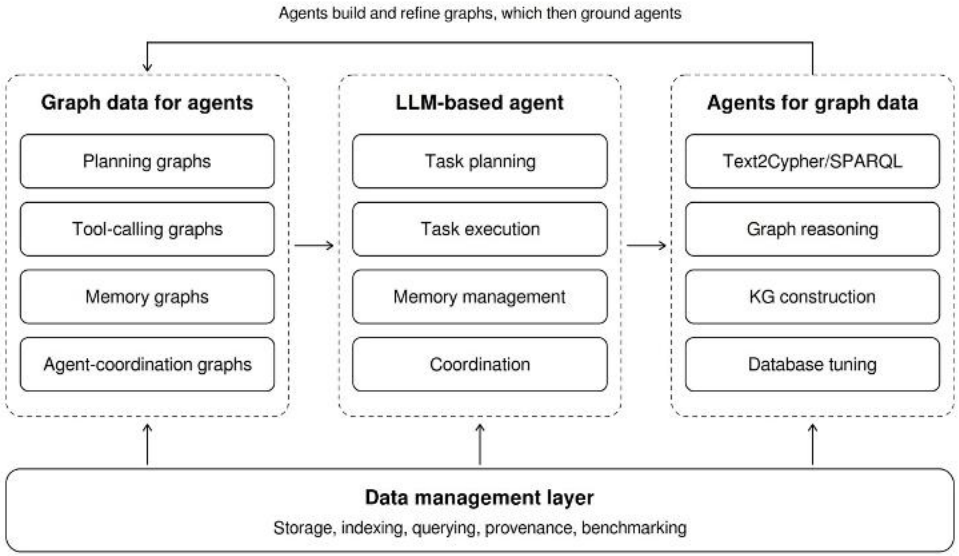

What is Generative AI?
Learns patterns from vast amounts of data.
Prompt (question)
New Content
Learns patterns from vast amounts of data.
What is an AI Agent?
Acts in the real or digital world
Goal: Buy ingredients!
Performs actions to achieve your goals
What is Agentic AI?
Goals into actionable plans
Goal:
Iterates as needed


Source: Google Trends
AI Agents
Agentic AI
Nov 2022
Nov 2023
Nov 2024
2025


Graphs Meet AI Agents [Bei et al., https://arxiv.org/abs/2506.18019]
(1) Graph Methodology
Graph
Data Organization
Graph Learning
Knowledge Extraction
(2) AI Agent Methodology
AI Agents
Please help me [...]
User
Task Input
Foundation Model
LLMs
Learning Paradigm
RL
Planning, Execution, Memory, Multi-Agent Coordination...
Processing
Complete
(3) AI Agents for Graphs
Graph Annotation and Synthesis
Annotation
Synthesis
Graph Understanding
Node Classification
Link Prediction
(4) Graphs for AI Agents
Agent Planning
Task Reasoning
Task Decomposition
Decision Searching
Agent Execution
Tool Usage
Environment Interaction
Agent Memory
Memory Organization
Memory Retrieval
Memory Maintenance
Multi-Agent Coordination
Message Passing
Topology Optimization

AI Harness and Graph Data Management





Graphs for AI Agents

Graph Schema and Task Planning Graph

- Schemas serve as "planning" abstractions for humans, facilitating the understanding, retrieval, and exploration of graphs
- Task Planning Graph serves as a planning "schema" that enables AI agents to execute a given task.

Graph Queries and Task Execution Graph

- Graph queries enable humans to navigate graph data through query execution
- Task Execution Graphs enable agents to navigate relevant data through task execution.

Graph Data and Memory Graph

- Graph data interactions enable humans to explore and retrieve information
- Memory graph interactions enable agents to explore and retrieve contextual knowledge.

VLDB 2026 (Tutorial): Sourav S Bhowmick, **Arijit Khan**, and Byron Choi, "From Human-Graph to Agent-Graph Interaction: State of the Art and Future Directions"

Agents + Graph workshop (VLDB 2026): Yixiang Fang, **Arijit Khan,** Da Yan, Tianxing Wu

AI Agent Tool Calling Benchmark [under submission https://openreview.net/forum?id=OnWhe1Zl1Q]

- Evaluation science that measures and explains failures at the process level and identifies where misalignment emerges inside graph-structured reasoning.

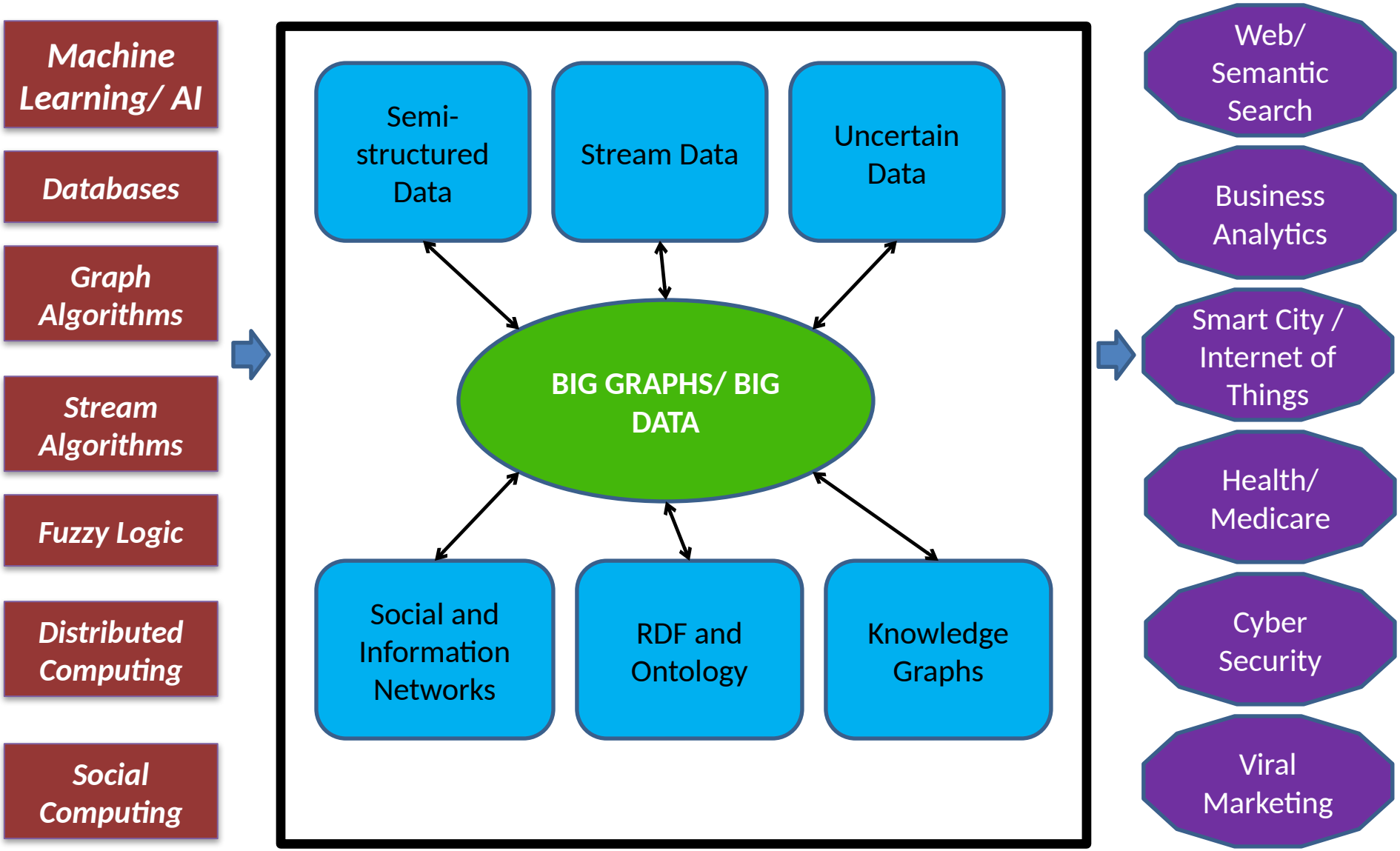


Workflow of tool learning

- Current benchmarks mainly focus on final answer accuracy and do not assess intermediate capabilities such as sequential graph comprehension or quantitative conflict detection between tool outputs and model knowledge.

Aspect	GraphToolBench (Ours)	ShortcutsBench [1]	ToolEyes [2]	SoAyBench [3]	ToolHop [4]	StableToolBench [5]	TaskBench [6]	ToolBench [7]	ToolSword [8]	APIBench [9]
Sequential Graph Comprehension ?	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗
Quantitative Conflict Identification ?	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗
One-time Multi-tool Invocation ?	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗
Offline Evaluation ?	✓	✗	✗	✗	✓	✓	✗	✗	✗	✗

Big Picture


Machine Learning/ AI
Databases
Graph Algorithms
Stream Algorithms
Fuzzy Logic
Distributed Computing
Social Computing
Semi-structured Data
Stream Data
Uncertain Data
BIG GRAPHS/ BIG DATA
Social and Information Networks
RDF and Ontology
Knowledge Graphs
Web/ Semantic Search
Business Analytics
Smart City / Internet of Things
Health/ Medicare
Cyber Security
Viral Marketing