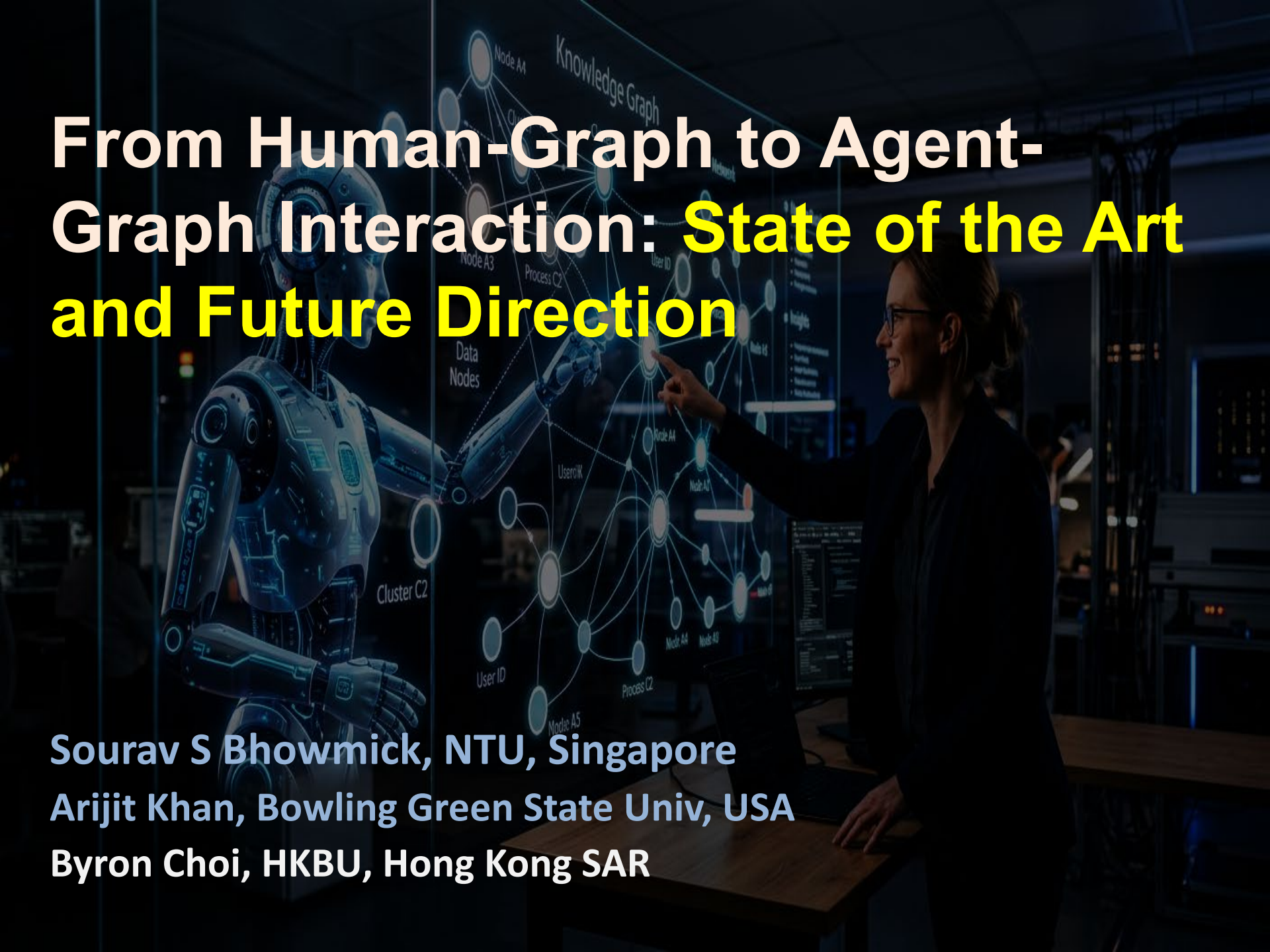


The background image depicts a futuristic laboratory or control room. On the left, a blue, semi-transparent humanoid robot stands, its arm extended towards a large digital display. On the right, a woman with glasses and a ponytail, wearing a dark jacket, points her finger at the same display. The display itself is a complex network graph with various nodes and connecting lines. Labels on the graph include 'Node A1', 'Node A3', 'Process C2', 'User ID', 'Node A4', 'Node A5', 'Node A6', 'Node A7', 'Node A8', 'Node A9', 'Node A10', 'Node A11', 'Node A12', 'Node A13', 'Node A14', 'Node A15', 'Node A16', 'Node A17', 'Node A18', 'Node A19', 'Node A20', 'Node A21', 'Node A22', 'Node A23', 'Node A24', 'Node A25', 'Node A26', 'Node A27', 'Node A28', 'Node A29', 'Node A30', 'Node A31', 'Node A32', 'Node A33', 'Node A34', 'Node A35', 'Node A36', 'Node A37', 'Node A38', 'Node A39', 'Node A40', 'Node A41', 'Node A42', 'Node A43', 'Node A44', 'Node A45', 'Node A46', 'Node A47', 'Node A48', 'Node A49', 'Node A50', 'Node A51', 'Node A52', 'Node A53', 'Node A54', 'Node A55', 'Node A56', 'Node A57', 'Node A58', 'Node A59', 'Node A60', 'Node A61', 'Node A62', 'Node A63', 'Node A64', 'Node A65', 'Node A66', 'Node A67', 'Node A68', 'Node A69', 'Node A70', 'Node A71', 'Node A72', 'Node A73', 'Node A74', 'Node A75', 'Node A76', 'Node A77', 'Node A78', 'Node A79', 'Node A80', 'Node A81', 'Node A82', 'Node A83', 'Node A84', 'Node A85', 'Node A86', 'Node A87', 'Node A88', 'Node A89', 'Node A90', 'Node A91', 'Node A92', 'Node A93', 'Node A94', 'Node A95', 'Node A96', 'Node A97', 'Node A98', 'Node A99', 'Node A100'. The overall lighting is dim, with blue and white highlights from the digital display and the robot's body.

From Human-Graph to Agent- Graph Interaction: **State of the Art and Future Direction**

Sourav S Bhowmick, NTU, Singapore

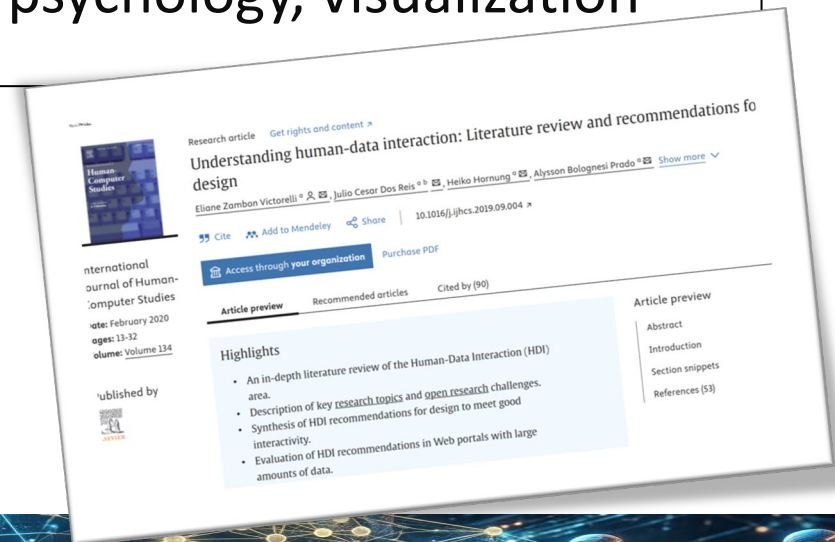
Arijit Khan, Bowling Green State Univ, USA

Byron Choi, HKBU, Hong Kong SAR

Human-Data Interaction (HDI)

HDI

- Studies how people engage with large, complex, and heterogeneous datasets
- Design to build **systems** that enable **effective** data understanding and decision-making
- Multi-disciplinary: HCI, cognitive psychology, visualization



Human-Data Interaction (HDI)

“*Human manipulation, analysis, and sensemaking of large, unstructured, and complex datasets.*”

N. Elmqvist. Embodied Human-Data Interaction. In ACM CHI Work. on Embodied Interact. Theory Pract., 2011

“*Technologies that use embodied interaction to facilitate the users' exploration of rich datasets.*”

F. Cafaro. Using Embodied Allegories to Design Gesture Suites for Human-data Interaction. In ACM UbiComp, 2012

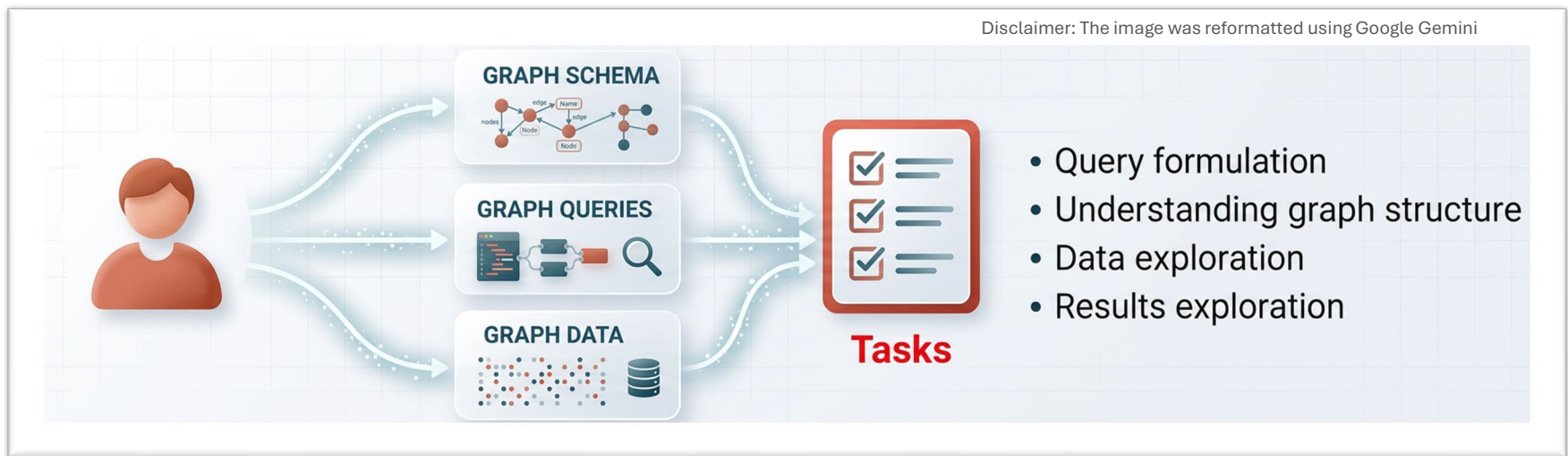
HDI vs HCI

- **System-oriented** vs **data-oriented**
- HCI broadly focuses on the design and evaluation of interactive computing systems
- HDI examines how users access, interpret, and control data within those systems.



Human-Graph Interaction (HGI)

“ *Human manipulation, analysis, exploration, and sensemaking of graph-structured datasets to achieve desired outcomes.* ”



Orthogonal to graph data systems: relational, KV stores, native graphs



Focus: Visual-Text Interaction

Interaction

- **Visual-text** vs Pure text
- Text: Formal languages, natural languages

Practicality

- Visual interaction **lower the barrier to entry** by abstracting formal query and schema syntax
- *Cypher, PG-Schema*: Need specialists who are not domain specialists
- *LLM-generated queries*: Need specialists to verify correctness



Cognitively: Perceptual Speed

Pre-Attentive Processing vs. Textual Parsing

- Text requires **serial, linear reading** in the cerebral cortex.
- Every word must be phonologically decoded, creating a heavy temporal bottleneck.
- Visual primitives (color, size, spatial position) process automatically in **sensory memory** within < 250 ms by **bypassing initial cognitive effort**
- The visual system transmits information at **high parallel capacity** (~107 bits/s), whereas text reading operates at only ~50 bits/s.

```
CREATE GRAPH TYPE    fraudGraphType STRICT {
    (personType: Person {name STRING}),
    (customerType: personType & Customer {id INT32}),
    (creditCardType: CreditCard {num STRING}),
    (transactionType: Transaction {num STRING}),
    (accountType: Account {id INT32}),
    (:customerType)
        -[ownsType: owns]->
    (:accountType),
    (:customerType)
        -[usesType: uses]->
    (:creditCardType),
    (:transactionType)
        -[chargesType: charges {amount DOUBLE}]->
    (:creditCardType),
    (:transactionType)
        -[activityType: deposits|withdraws]->
    (:accountType)
}
```

https://en.wikipedia.org/wiki/Pre-attentive_processing

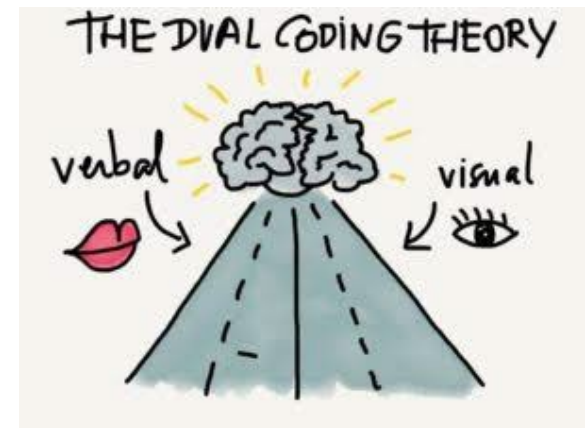


Cognitively: Visual + Text is Superior

Paivio's Dual-Coding Theory

- Brain processes information via **two distinct subsystems**: visual and verbal.
- **Baddeley's model** divides **working memory** into the *Visuospatial Sketchpad* and *Phonological Loop*.
- **Combining visuals with text utilizes both without overloading either.**
- Visual data interaction builds dual cognitive anchors, yielding higher retention and significantly faster retrieval compared to text alone.

Unclear combinations of words and visuals may create clutter, distraction and unnecessary cognitive load



Paivio, A. (1986). *Mental representations: A dual coding approach*. New York: Oxford University Press



Agent-Graph Interaction (AGI) in The AI Era

AI Agents

- Rapid advancement of LLMs and AI agents
- Increasingly feasible for humans to delegate various tasks to autonomous systems.

Agent-Graph Interaction (AGI)

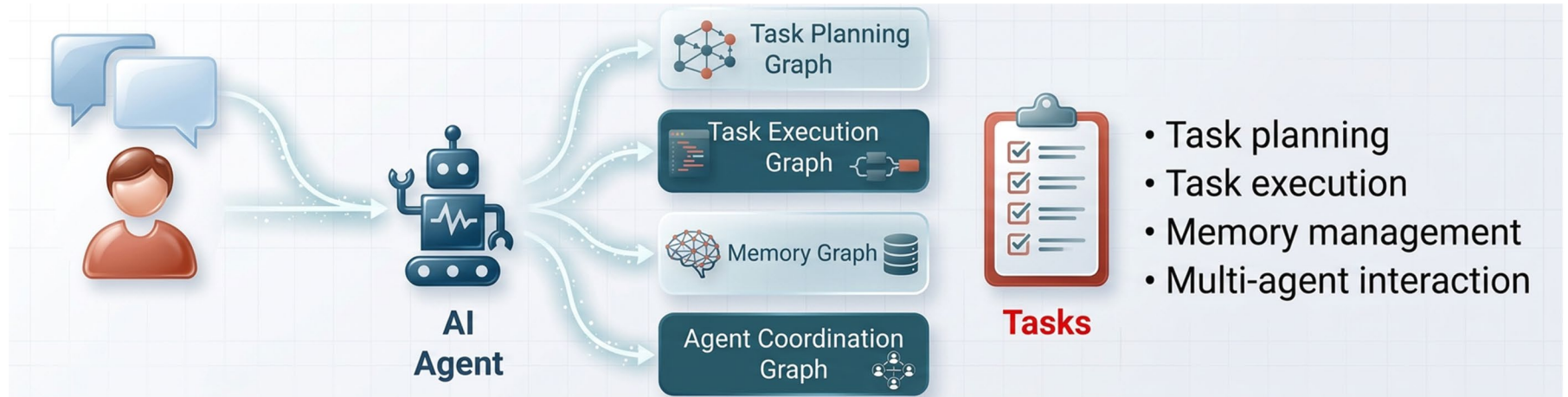
- Recent work on AI agents interacting with graph-structured data/abstraction to perform user-specified tasks
- In this tutorial, AGI refers strictly to Agent-Graph Interaction, rather than Artificial General Intelligence



Agent-Graph Interaction (AGI)

“ *Manipulation, analysis, exploration, and sensemaking of graphs by AI agents to achieve human-defined tasks.* ”

Disclaimer: The image was reformatted using Google Gemini



Conceptual Commonalities

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.



Disclaimer: The image was generated using Google Gemini

Conceptual Commonalities

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.



Disclaimer: The image was generated using Google Gemini

Conceptual Commonalities

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.



Disclaimer: The image was generated using Google Gemini



The Road Ahead

HUMAN-GRAPH AND AGENT-GRAPH INTERACTION: presentation overview & structural breakdown

PART 1: HUMAN-GRAPH INTERACTION (HGI)

1. INTERACTION WITH GRAPH SCHEMA



- Node types
- Node nodes
- Node labels

2. INTERACTION WITH GRAPH QUERIES



Intent mapping and builders

3. INTERACTION WITH GRAPH DATA



Visual exploration & data scrubbing



PART 2: AGENT-GRAPH INTERACTION (AGI)

1. INTERACTION WITH TASK PLANNING GRAPH



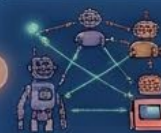
2. INTERACTION WITH TASK EXECUTION GRAPH



3. INTERACTION WITH MEMORY GRAPH



4. INTERACTION WITH AGENT COORDINATION GRAPH



PART 3: FUTURE RESEARCH DIRECTIONS & CONCLUSIONS



Importance of Schema

Prescriptive

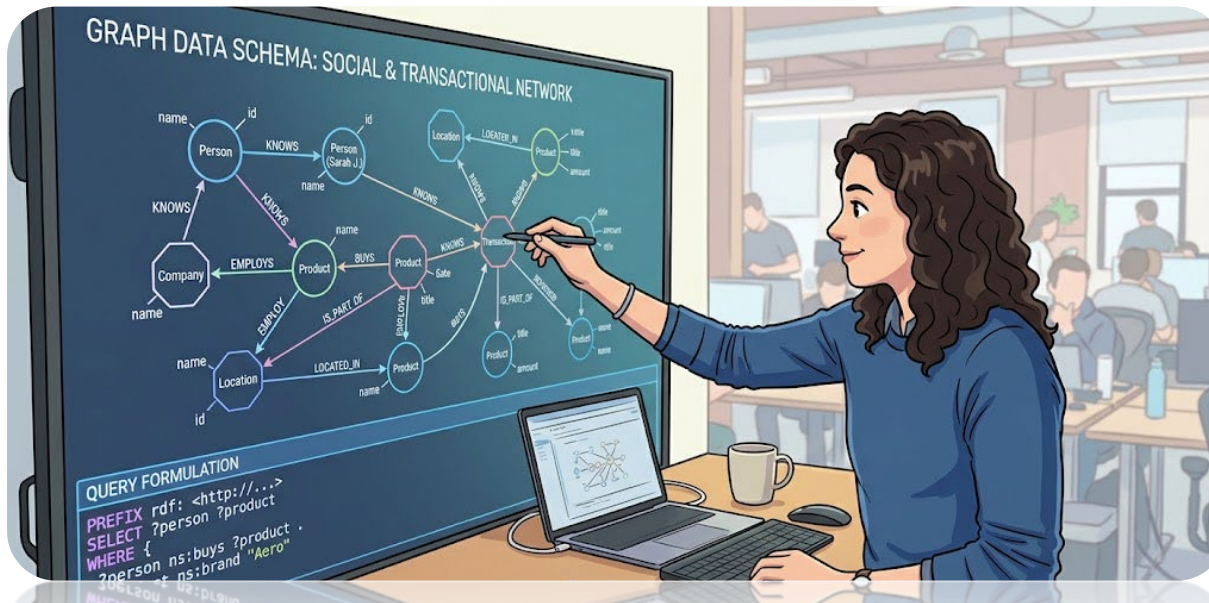
- Schemas serve as the building blocks of data systems
- Limiting graph data modifications based on constraints specified in it

Descriptive

- Informing users and data systems what to expect in the graph data



Human Interaction with Graph Schema



Human Interaction

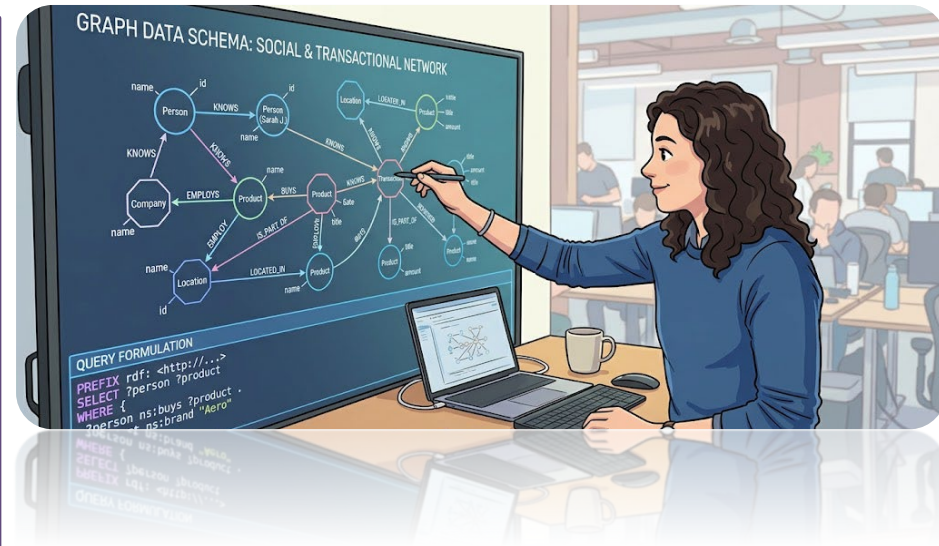
- Understand structure, properties, and constraints of data
- Facilitate query formulation

Disclaimer: The image was generated using Google Gemini



PG-Schema: Challenges with Textual Language

```
CREATE GRAPH TYPE fraudGraphType STRICT {  
  (personType: Person {name STRING}),  
  (customerType: personType & Customer {id INT32}),  
  (creditCardType: CreditCard {num STRING}),  
  (transactionType: Transaction {num STRING}),  
  (accountType: Account {id INT32}),  
  (:customerType)  
    -[ownsType: owns]->  
  (:accountType),  
  (:customerType)  
    -[usesType: uses]->  
  (:creditCardType),  
  (:transactionType)  
    -[chargesType: charges {amount DOUBLE}]->  
  (:creditCardType),  
  (:transactionType)  
    -[activityType: deposits|withdrawals]->  
  (:accountType)  
}
```



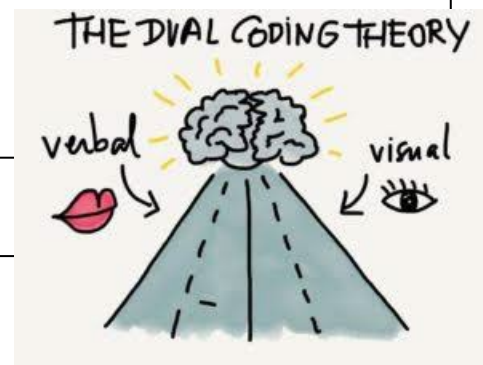
- Understanding of the syntax and semantics of a formal schema language
- Lower perceptual speed
- Does not exploit dual-encoding theory



Overview

Labeled Schema Graph (LSG)

- Node-link diagram
- **Schema visualization**: VizLOD [JCSSE 2018], SAGEViz [EMNLP 2023]
- **Schema Exploration**: DiscoPG [VLDB 2022]
- **Schema Evolution**: DBChEx [CIDR 2019]

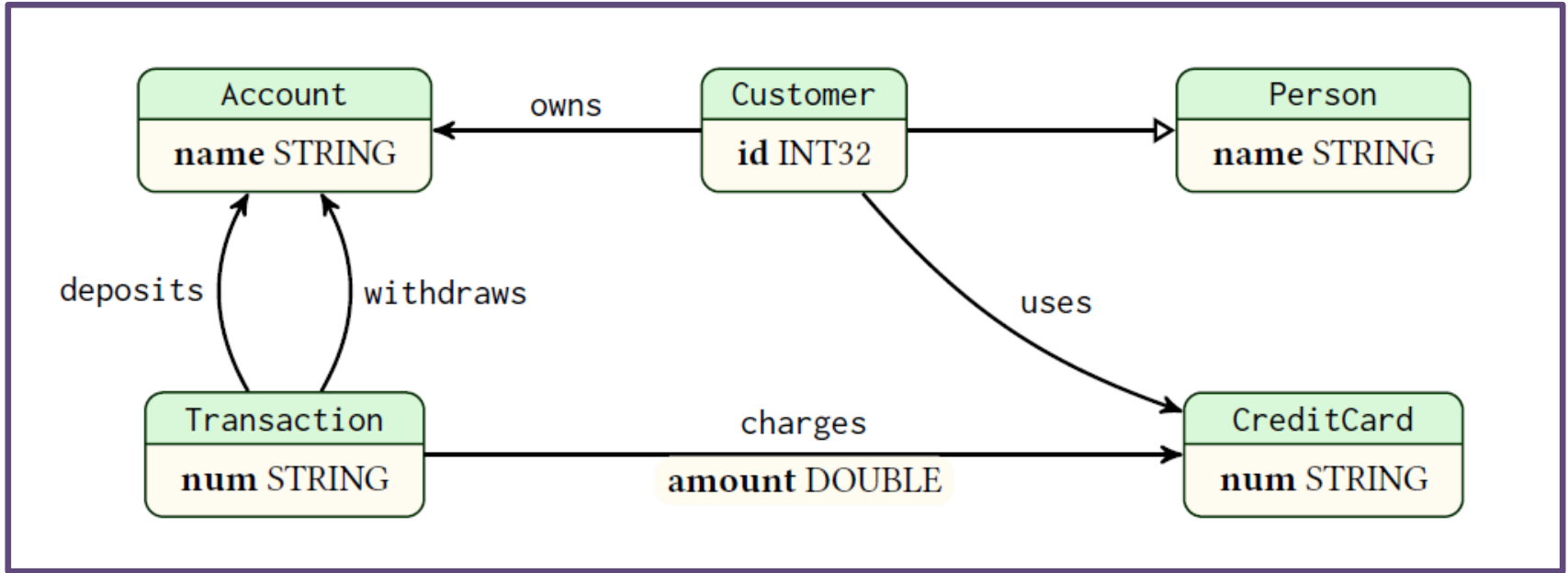


Labeled Iconized Composite Schema (LICS)

- Composite and iconized node-link diagram
- Richer visual encodings
- **Schema visualization & exploration**: LICS [SIGMOD 2025]

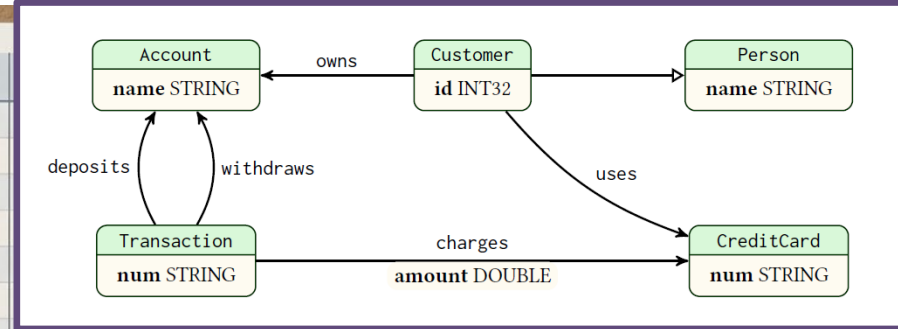


Labeled Schema Graph (LSG)



Challenges with LSG

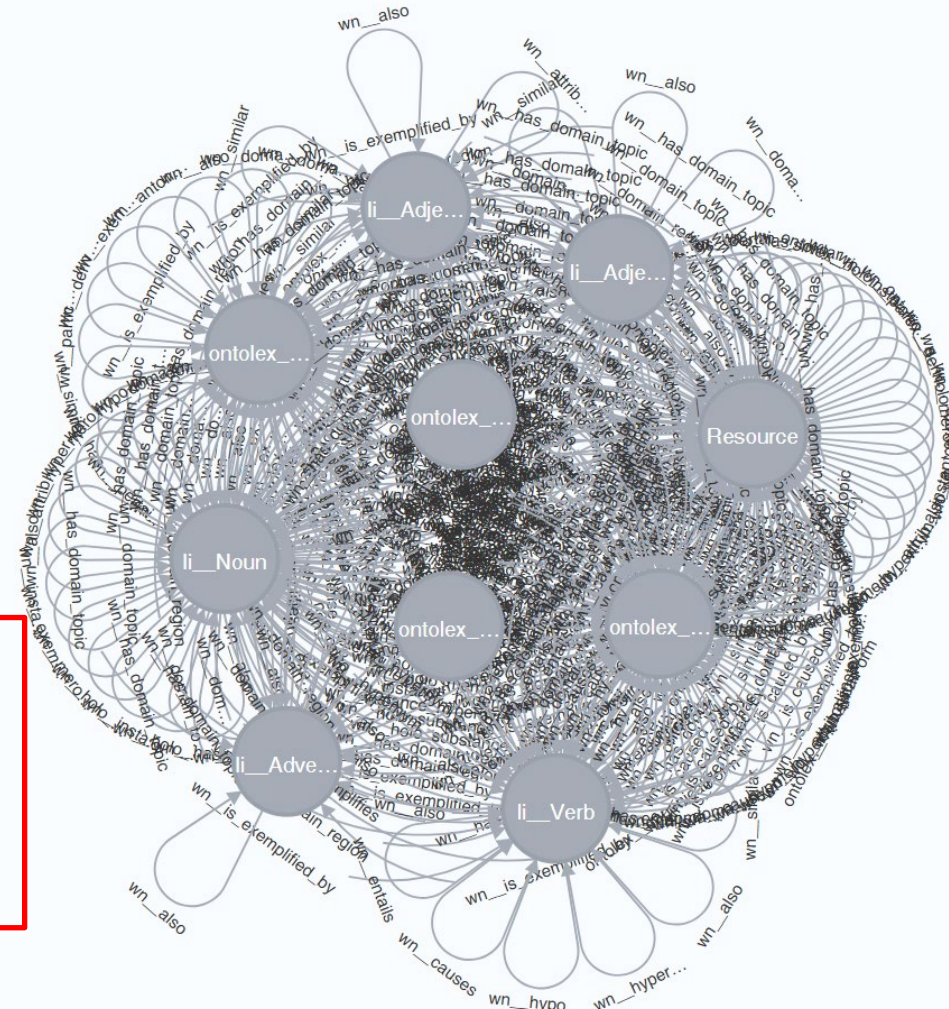
Name	Papers	$ V $	$ E $	$ P $	$\bar{d}$	ρ
catalog		6	8	14	2.67	0.47
contactracing		6	6	20	2.00	0.40
cybersecurity		7	58	95	16.57	0.76
entityresolution		4	3	14	1.50	0.50
finen		3	5	8	2.00	0.67
frauddetection		13	76	47	11.69	0.45
gameofthrones		1	5	19	10.00	0.00
graphdatascience		11	158	304	28.73	0.53
healthcareanalytics		8	11	40	2.75	0.29
icijoffshoreleaks		5	44	222	17.60	0.90
icijparadisepapers		5	18	98	7.20	0.70
movies	[57]	2	6	9	6.00	1.00
neoflix	[15]	13	32	114	4.92	0.33
networkmanagement		17	50	49	5.88	0.23
northwind	[15] [57]	5	4	56	1.60	0.40
openstreetmap		10	35	425	7.00	0.71
pole		11	18	34	3.27	0.22
recommendations	[15]	6	8	54	2.67	0.33
stackoverflow		5	21	52	8.40	0.90
twitch		5	18	19	7.20	0.70
twitter		6	20	33	6.67	0.40
twittertrolls		6	9	40	3.00	0.33
wordnet		12	417	76	69.50	0.71



- Visual clutter with long list of properties, edge crossing
- No visual encoding for constraints, label expressions, OPEN keyword, etc



Challenges with LSG



*WordNet
in Neo4j*

Unclear combinations of words and visuals may create clutter, distraction and unnecessary cognitive load



Labeled Iconized Composite Schema (LICS)

Overview

- A generic and extensible visual abstraction
- Employs **composite geometric shapes** and **icons**
- Richer visual encodings: Node and Edge types, properties, inheritance, constraints, label expressions
- Guided by theories from HCI, visualization, and cognitive psychology

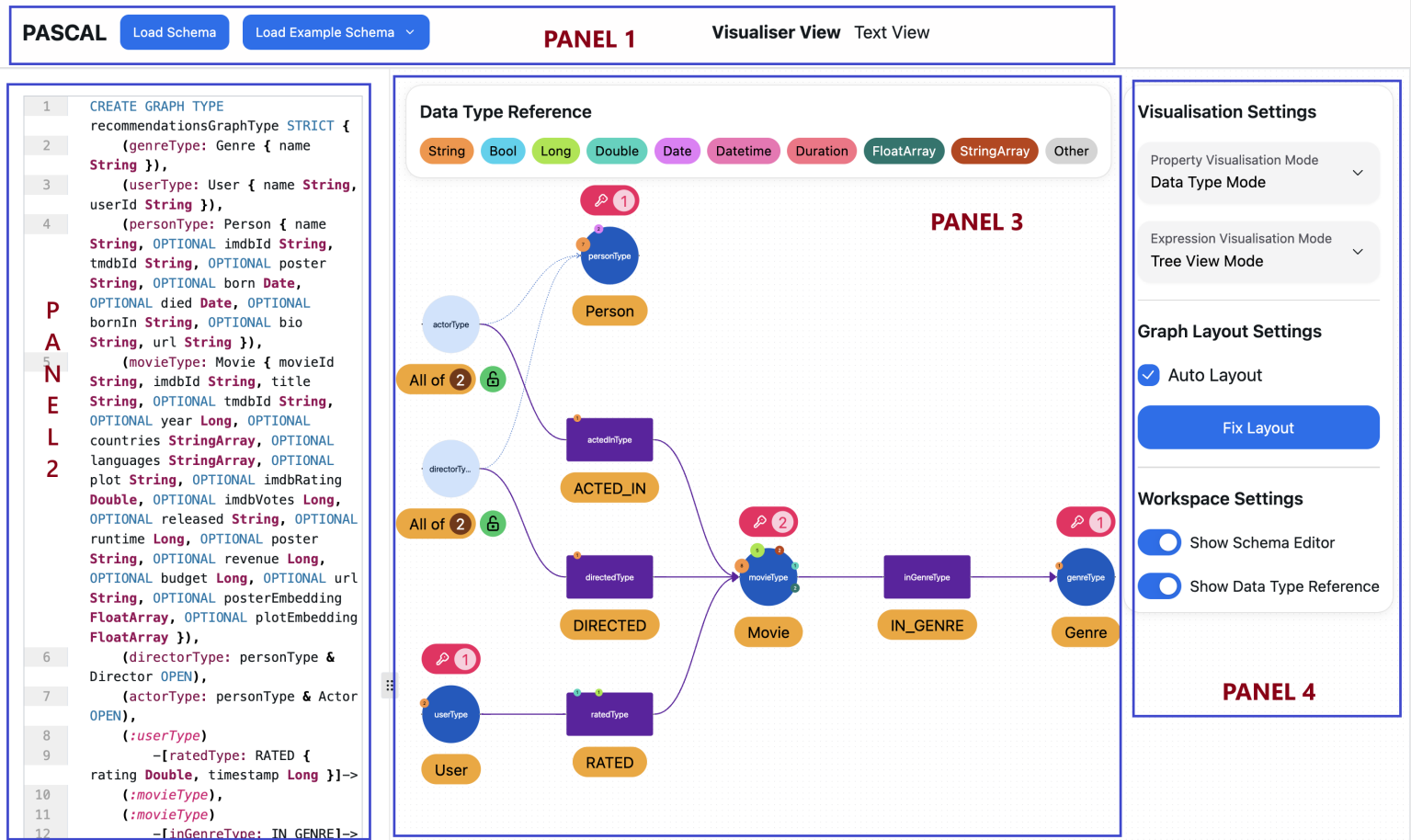
Advantages

- More effective visualization to facilitate schema understanding
- More efficient query formulation

Kasidis Chanthatrojwong, Sourav S. Bhowmick, Byron Choi: **LICS: Towards Theory-Informed Effective Visual Abstraction of Property Graph Schemas**. *Proc. ACM Manag. Data* 3(3): 180:1-180:26 (2025)

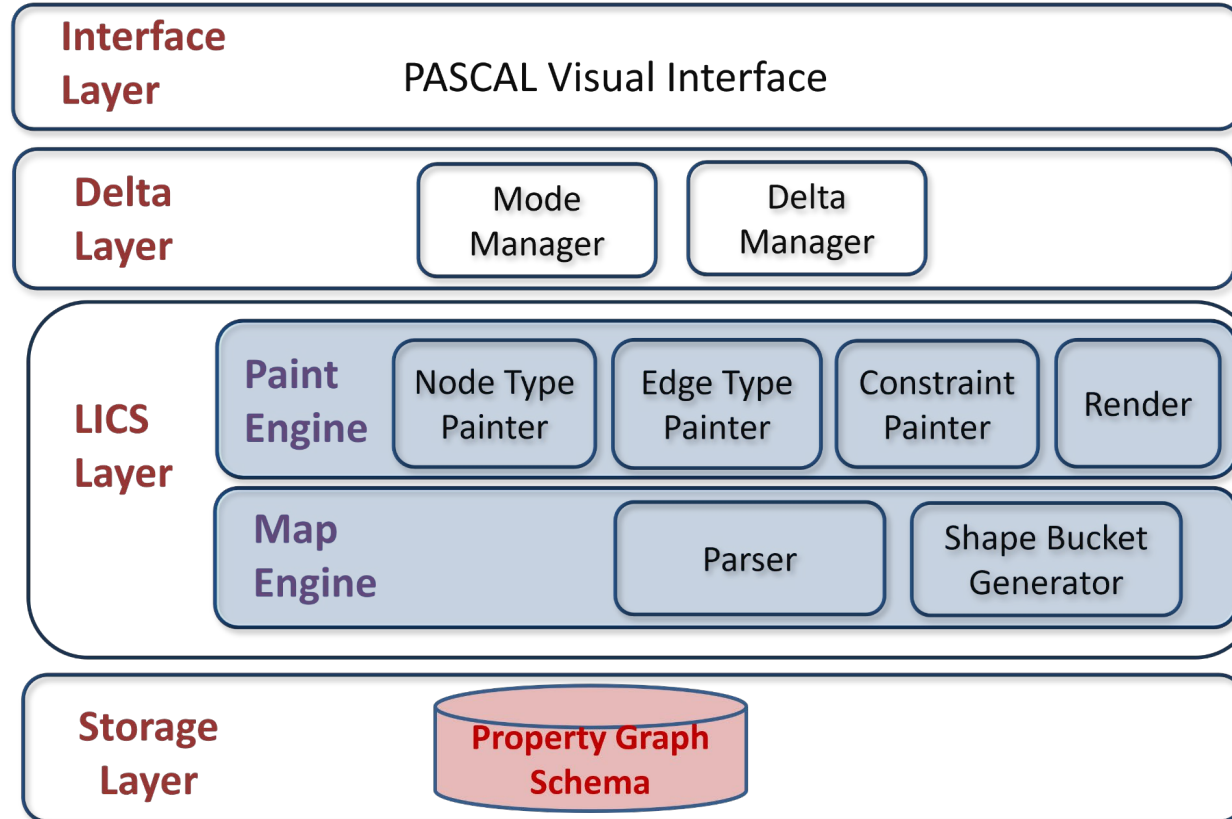


PASCAL: Realizing LICs

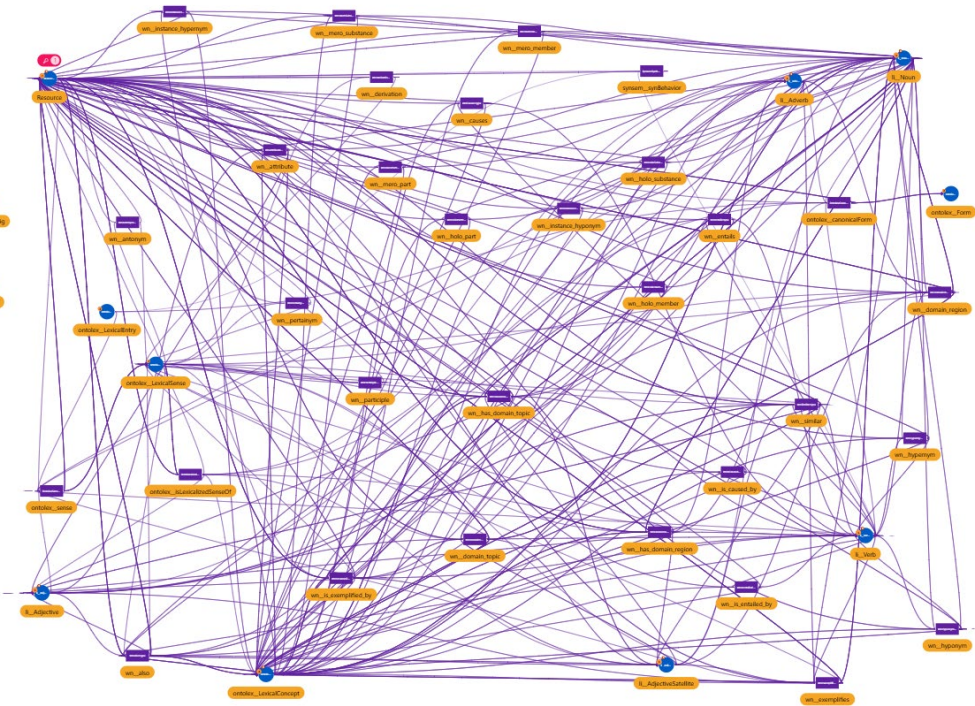
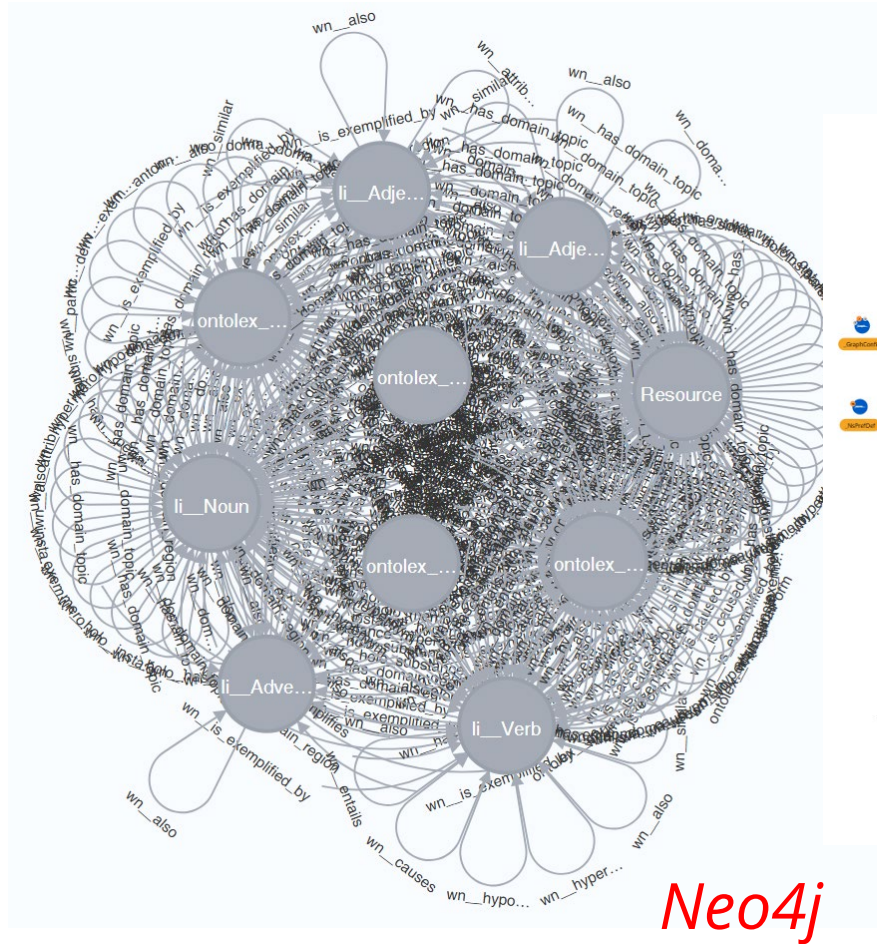


Kasidis Chanthatrojwong, Sourav S. Bhowmick, Byron Choi: PASCAL: A Theory-Informed Visual Interface for Property Graph Schema Visualization. In SIGMOD, 2025 (Demo).

The Map-Paint Engine



Open Challenges



From Humans to Agents..

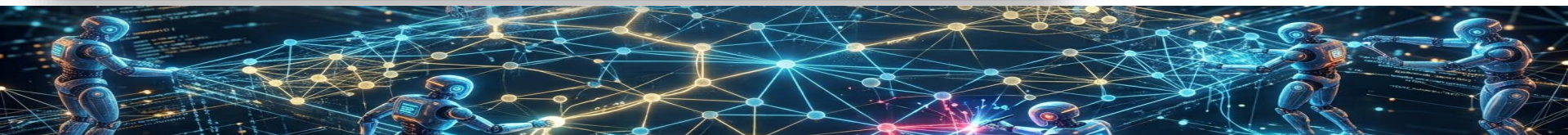
Schema

- Serves as “planning” abstraction for humans



- Task planning graph – “Schema” for agents
- Graph schemas can aid construction

Disclaimer: The image was generated using Google Gemini



Next..

HUMAN-GRAPH AND AGENT-GRAPH INTERACTION: presentation overview & structural breakdown

PART 1: HUMAN-GRAPH INTERACTION (HGI)

1. INTERACTION WITH GRAPH SCHEMA



- Node types
- Node nodes
- Node labels

2. INTERACTION WITH GRAPH QUERIES



Intent mapping and builders

3. INTERACTION WITH GRAPH DATA



Visual exploration & data scrubbing



PART 2: AGENT-GRAPH INTERACTION (AGI)

1. INTERACTION WITH TASK PLANNING GRAPH



2. INTERACTION WITH TASK EXECUTION GRAPH



3. INTERACTION WITH MEMORY GRAPH



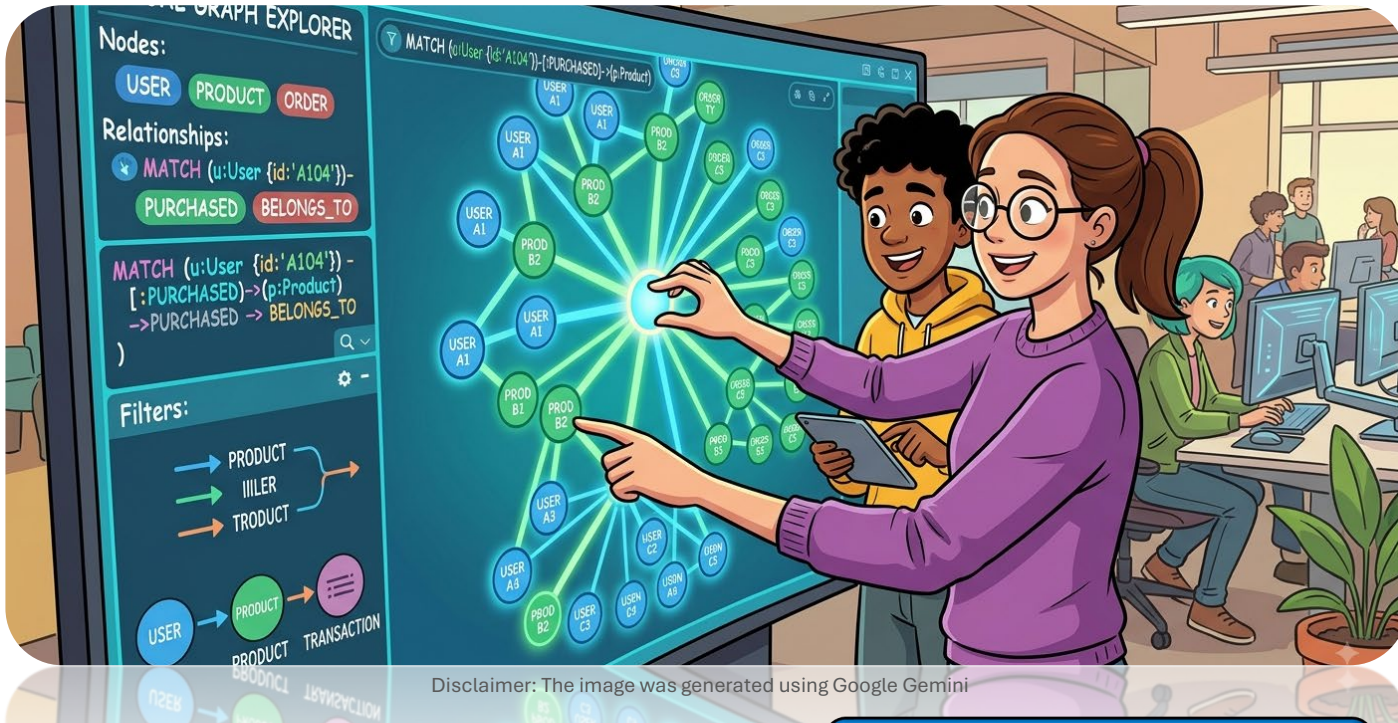
4. INTERACTION WITH AGENT COORDINATION GRAPH



PART 3: FUTURE RESEARCH DIRECTIONS & CONCLUSIONS



Searching Graphs



Human Interaction

- Query formulation
- Query visualization

Machine Work

- Query processing
- Huge volumes of work



Searching Graphs: Text-based Interaction

Graph Query Languages

- A soup of languages
- Demands knowledge of their syntax and semantics
- Daunting for non-programmers
- Graph query languages and usability are among top challenges in graph processing*

How about NL-based Graph Search?

- To the best of our knowledge, there is no robust framework that can handle queries with varying complexity

*Siddhartha Sahu, et al. The Ubiquity of Large Graphs and Surprising Challenges of Graph Processing. *PVLDB* 11(4): 420-431 (2017)



NL-based Graph Search

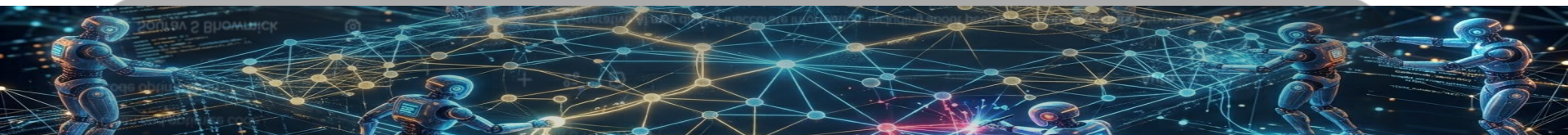
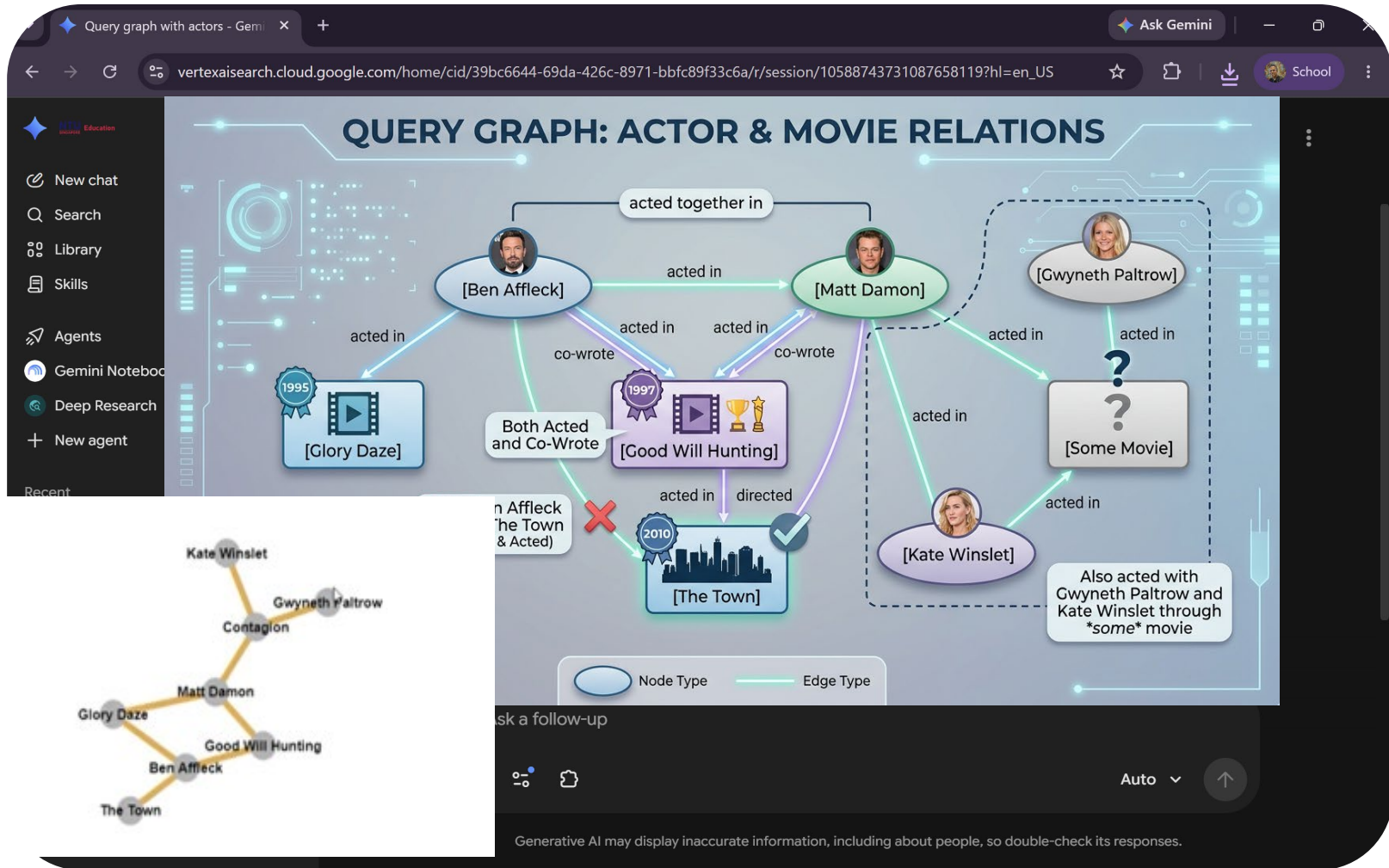


Disclaimer: The image was generated using Google Gemini

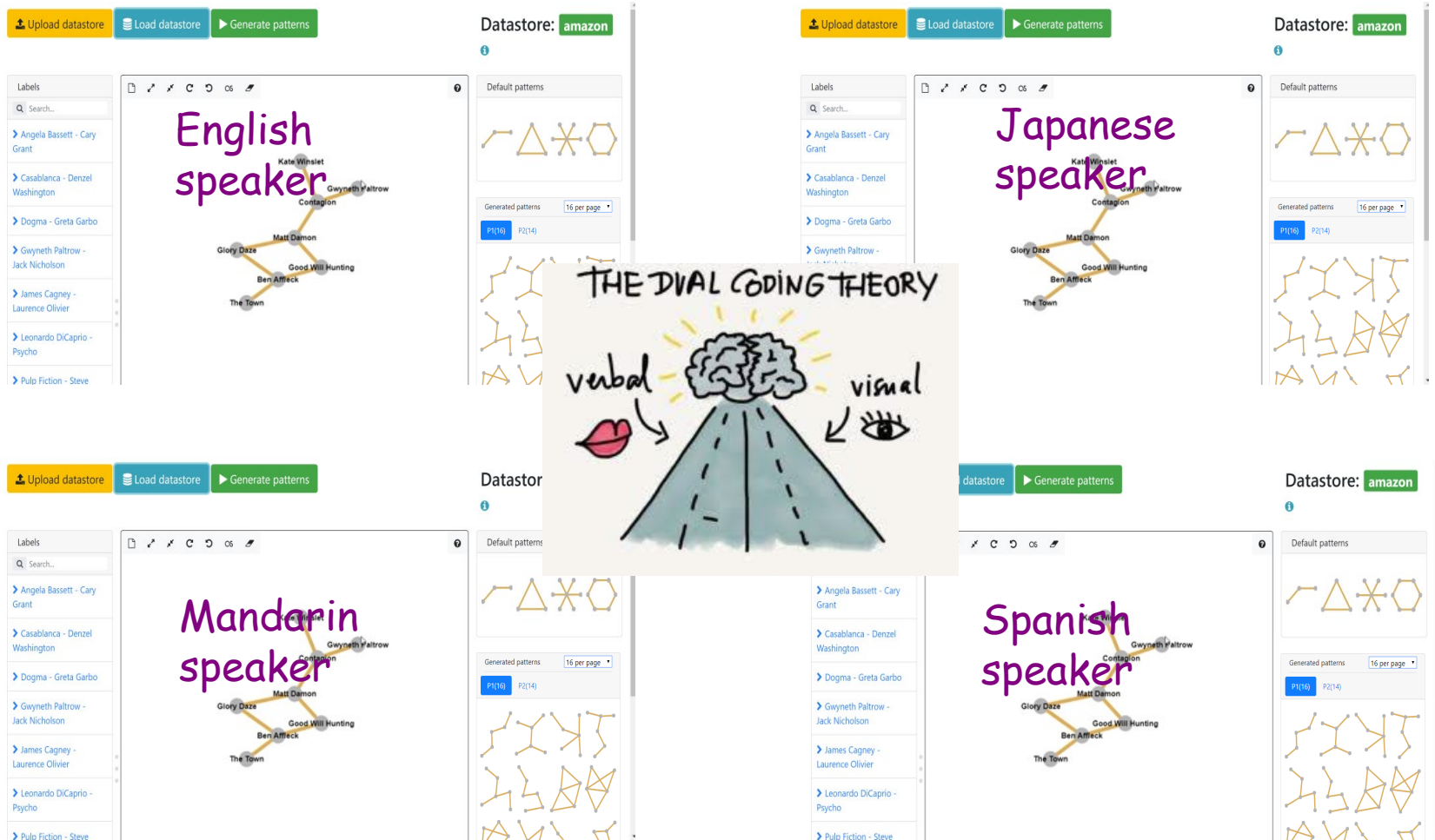
Find results where Ben Affleck and Matt Damon have acted together in Glory Daze and Good Will Hunting but only the former has acted in The Town. Matt Damon has also acted with Gwyneth Paltrow and Kate Winslet through some movie.



NL to Query Graph Via AI Agent



Visual Interaction with Queries



Visual Query Interfaces Are Real

Visual Query Interface: eMolecules

empowering drug discovery

Search and order from over 9 million Building Blocks

Structure Search List Search

Building Blocks

Screening Compounds

Antibodies

CHNOPs
F Cl Br I Si Sn
Add Element

Copyright © 2022 eMolecules, Inc. | Terms of Service | Privacy Policy | Feedback | Contact Us

Visual Query Interface: DRAW STRUCTURE

Broadband SMILES

New Undo Cut Style Delete Copy Paste

Li Be B C N O F Ne
Na Mg Al Si P S Cl Ar
K Ca Sc Ga Ge As Se Br Kr
Rb Sr Y In Sn Sb Te I Xe
Cs Ba Lu Tl Pb Bi Po At Rn

Export MDL Molfile Done
Hydrogen Keep AsIs Help
Import Choose File No file chosen

Search for This Structure

Visual Query Interface: Neo4j Browser

Mini-app: The Movie Graph

Actions and their relationships. Click the code to load into the editor. Run full.

```
CREATE (TheMatrix:Movie {title:"The Matrix", released:1999, tagline:"Welcome to the Real World"})
CREATE (Keanu:Person {name:"Keanu Reeves", born:1964})
CREATE (Carrie:Person {name:"Carrie-Anne Moss", born:1967})
CREATE (Laurence:Person {name:"Laurence Fishburne", born:1961})
CREATE (Kipo:Person {name:"Kipo Weaving", born:1960})
```

your graph-query workbench

Visual Query Interface: DRUGBANK Online

Search Options

Similarity Substructure Exact

Similarity threshold 0.7

Minimum Weight e.g. 100 Maximum Weight e.g. 200

Maximum Results 100

Drug Types (default all):

☐ Approved ☐ Vet approved ☐ Nutraceutical
☐ Illicit ☐ Withdrawn ☐ Investigational
☐ Experimental

Search

LOAD EXAMPLE

MARVIN'S TUTORIALS

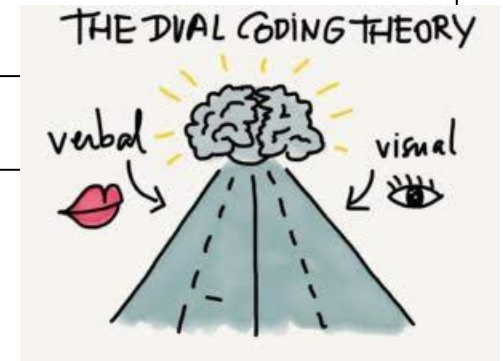
Visual Abstraction of Queries

Labeled Atomic Graph (LAG)

- Query nodes are represented using atomic shapes
- Links are shown using directed or undirected (labeled) edges
- **Query Formulation & Visualization**: VISAGE [SIGMOD 2017], Bloom

Labeled Composite Graph (LCG)

- Composite and iconized node-link diagram
- Richer visual encodings
- **Property graph query formulation & visualization**: LCG [SIGMOD 2023]



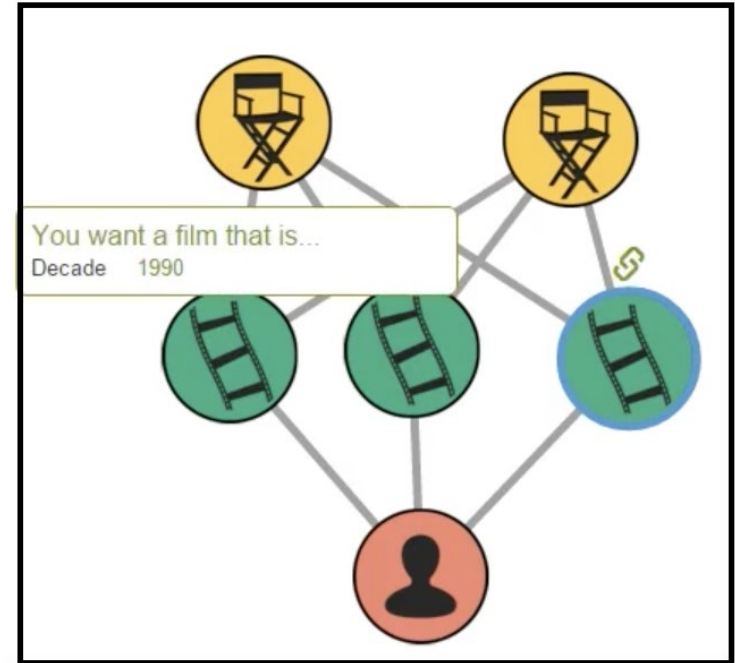
Limitations of the LAG Abstraction

Misaligned with theories and principles in HCI, visualization and psychology

Impacts visual and task complexities of LAG-based GUIs

Data-unaware and “static”

The classical atomic representation of nodes and edges in a LAG abstraction limits effective visualization of predicates on their properties.



R. Pienta, F. Hohman, et al. Visual graph query construction and refinement. In SIGMOD, 2017.



Labeled Composite Graph (LCG)

Overview

- A generic and extensible visual abstraction for read-only property graph queries
- Employs **composite geometric shapes** for richer visual encodings
- Guided by theories from HCI, visualization, and cognitive psychology

VEDA - Visual Shape Definition Language

- A set of **visual shape operators** to **automatically generate, maintain, and position** various shapes in LCG to balance visual and task complexity, aesthetics, and usability.
- Exploits **principles and theories** from psychology, visualization and HCI

J. Ma, S. S. Bhowmick, B. Choi, L. Tay. Theories and Principles Matter: Towards Visually Appealing and Effective Abstraction of Property Graph Queries. In SIGMOD, 2023.



SIERRA: Realizing LCG

← Order

☐ shipCity

☒ shipCountry

☐ shipAddress

☐ shipPostalCode

☐ orderDate

☒ employeeID

☐ customerID

☐ orderID

☒ shipRegion

☐ freight

Possible Targets

Product

SIERRA

New Node

Full

Play

Cypher Query

Translate

```
1 MATCH (a:Customer)-[ra:PURCHASED]->(b:Order), (b:Order)
2 WHERE a.phone = '(5) 555-4729' AND a.contactTitle = 'Or
3 RETURN a, b, c, d
```

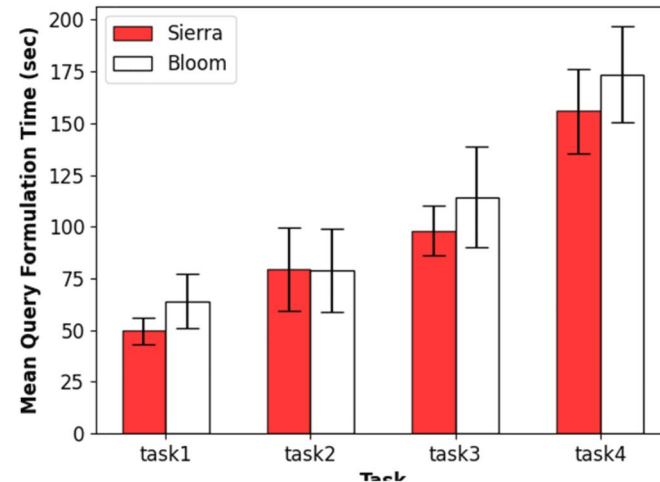
Jiebing Ma, Sourav S. Bhowmick, Lester Tay, Byron Choi: **SIERRA: A Counterfactual Thinking-based Visual Interface for Property Graph Query Construction.** In SIGMOD, 2024.



Usefulness of LCG

User Study

- Demonstrate the superiority of the LCG abstraction to its LAG counterpart w.r.t. usability and aesthetics.
- More efficient query formulation



Open Challenges

COMPLEX CYPHER QUERY: E-COMMERCE FRAUD & RISK DETECTION (DISJUNCTIONS, AGGREGATIONS, SUBQUERIES)

```
// 1. Identify high-risk customers using disjunctions
MATCH (c:Customer)-[:HAS_ACCOUNT]->(a:Account)
WHERE (a.status = 'SUSPENDED' OR a.riskScore >= 0.75)
OR EXISTS { MATCH (a)-[:PERFORMED]->(t:Transaction) WHERE t.amount > 10000 OR t.isFlagged = true }

// 2. Correlated subquery: Aggregate recent transactions per account
CALL {
  WITH a
  MATCH (a)-[r:PERFORMED|INITIATED]->(t:Transaction) // Relationship Disjunction
  WHERE t.timestamp >= datetime() - duration('P30D')
  WITH count(t) AS totalTxCount, sum(t.amount) AS totalVolume, avg(t.amount) AS avgTxAmount, collect(DISTINCT t.category) AS
    categories, sum(CASE WHEN t.type = 'REFUND' OR t.type = 'CHARGEBACK' THEN 1 ELSE 0 END) AS DisputeCount
  RETURN totalTxCount, round(totalVolume, 2) AS totalVolume, round(avgTxAmount, 2) AS avgTxAmount, categories, DisputeCount
}

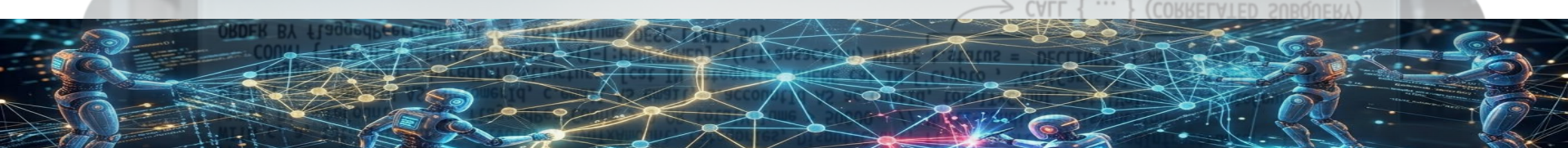
// 3. Correlated subquery: Detect shared banned devices/IPs
CALL {
  WITH c
  MATCH (c)-[:LOGGED_IN_WITH|REGISTERED_FROM]->(node) // Pattern Disjunction
  WHERE node:Device OR node:IPAddress // Label Disjunction
  MATCH (other:Customer)-[:LOGGED_IN_WITH|REGISTERED_FROM]->(node)
  WHERE other <> c AND (other.isBanned = true OR node.isBlacklisted = true)
  RETURN count(DISTINCT other) AS flaggedPeerCount, collect(DISTINCT node.address) AS sharedInfrastructure
}

// 4. Final aggregation, disjunction filtering, and sorting
WITH c, a, totalTxCount, totalVolume, avgTxAmount, categories, DisputeCount, flaggedPeerCount, sharedInfrastructure
WHERE DisputeCount > 2 OR flaggedPeerCount > 0 OR totalVolume > 50000 // Multi-condition Disjunction
RETURN c.customerId AS customerId, c.email AS email, a.accountId AS accountId, totalTxCount, totalVolume, DisputeCount,
  flaggedPeerCount, sharedInfrastructure, [cat IN categories WHERE cat IN ['Crypto', 'GiftCards']] AS highRiskCategories,
  COUNT { MATCH (c)-[:HAS_ACCOUNT]->()-[:PERFORMED]->(t:Transaction) WHERE t.status = 'DECLINED' } AS failedTxCount
ORDER BY flaggedPeerCount DESC, totalVolume DESC LIMIT 50;
```

Diagram Annotations:

- LABEL DISJUNCTION**: Points to the condition `a.status = 'SUSPENDED' OR a.riskScore >= 0.75`.
- RELATIONSHIP DISJUNCTION**: Points to the relationship `[r:PERFORMED|INITIATED]` in the second subquery.
- AGGREGATIONS (COUNT, SUM, AVG, COLLECT, CASE)**: Points to the aggregation functions used in the second subquery.
- EXISTS { ... }**: Points to the `OR EXISTS { ... }` clause in the first query.
- CALL { ... } (CORRELATED SUBQUERY)**: Points to the `CALL { ... }` blocks in the second and third queries.

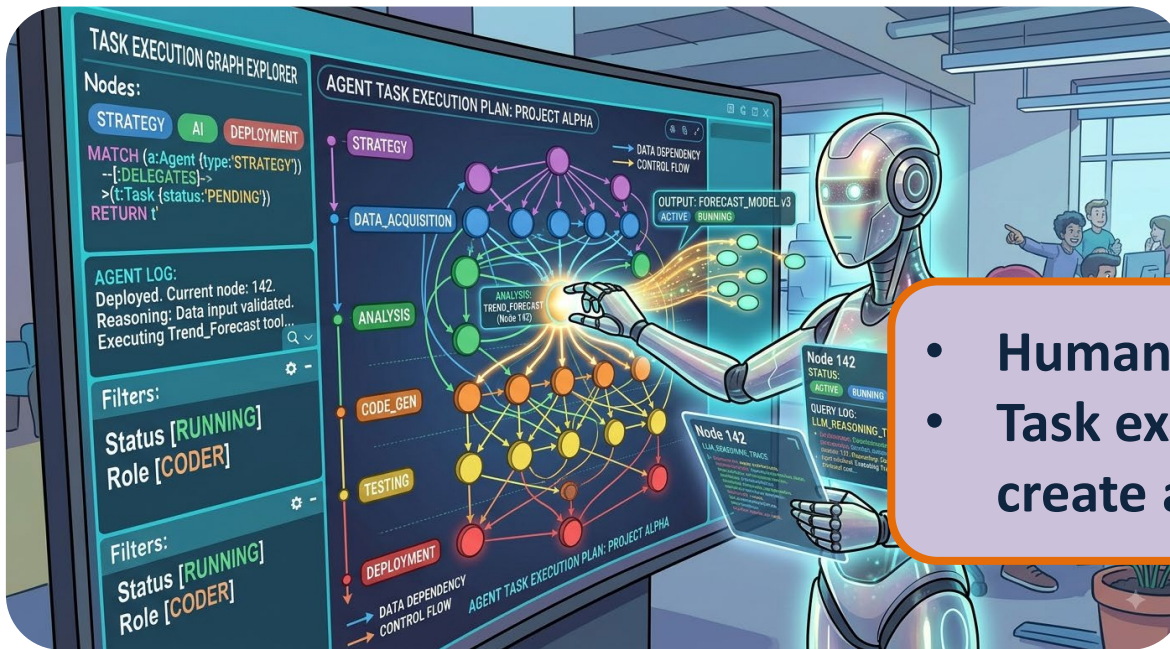
Disclaimer: The image was generated using Google Gemini



From Humans to Agents..

Queries

- Human formulated queries are executed by the query processor



- Human instructs agents in NL
- Task execution graph – Agents create and interact with it

Disclaimer: The image was generated using Google Gemini



Next..

HUMAN-GRAPH AND AGENT-GRAPH INTERACTION:

presentation overview & structural breakdown

PART 1: HUMAN-GRAPH INTERACTION (HGI)

1. INTERACTION WITH GRAPH SCHEMA



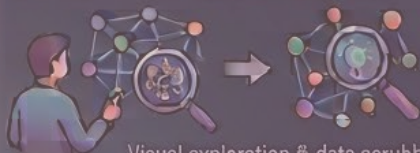
- Node types
- Node nodes
- Node labels

2. INTERACTION WITH GRAPH QUERIES



Intent mapping and builders

3. INTERACTION WITH GRAPH DATA



Visual exploration & data scrubbing

HUMAN INPUT &
FEEDBACK LOOP

AGENT
AUTONOMY &
DETERMINISTIC
EXECUTION

AGENT | INPUT & DETERMINISTIC
EXECUTION

PART 2: AGENT-GRAPH INTERACTION (AGI)

1. INTERACTION WITH TASK PLANNING GRAPH



2. INTERACTION WITH TASK EXECUTION GRAPH



3. INTERACTION WITH MEMORY GRAPH



4. INTERACTION WITH AGENT COORDINATION GRAPH



PART 3: FUTURE RESEARCH DIRECTIONS & CONCLUSIONS



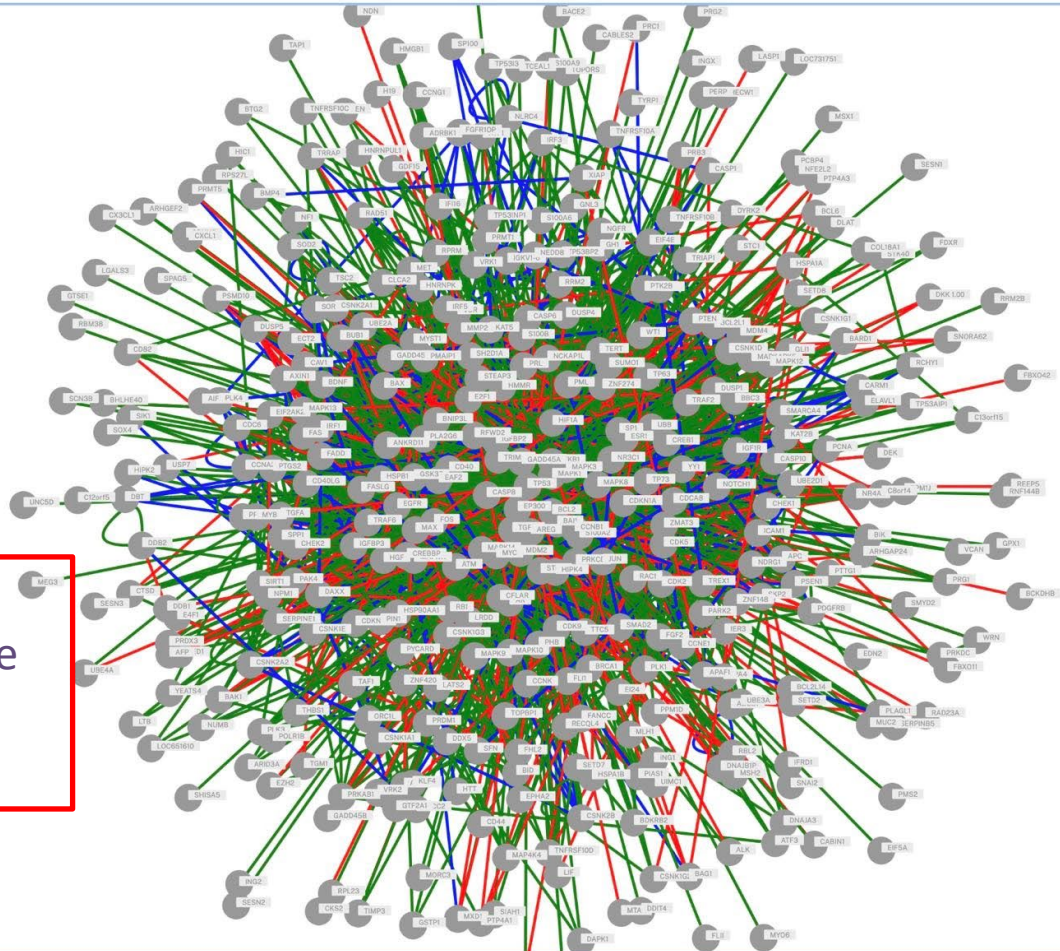
Direct Interaction with Graphs

Start

End

SHOW PATH

Unclear combinations of words and visuals may create clutter, distraction and unnecessary cognitive load



Cognitive Load of Graphs

Cognitive Load and Node-Link Diagrams

- Cognitive load for connectivity-based tasks increase with size.
- Perceptual limitations of node-link diagrams for performing tasks.

Key Results

- For small-world networks with 50 or more nodes with a density of 6 (resp. 2), volunteers failed to correctly find the shortest path in **more than half** of the trials.



Cognitive Load of Graphs

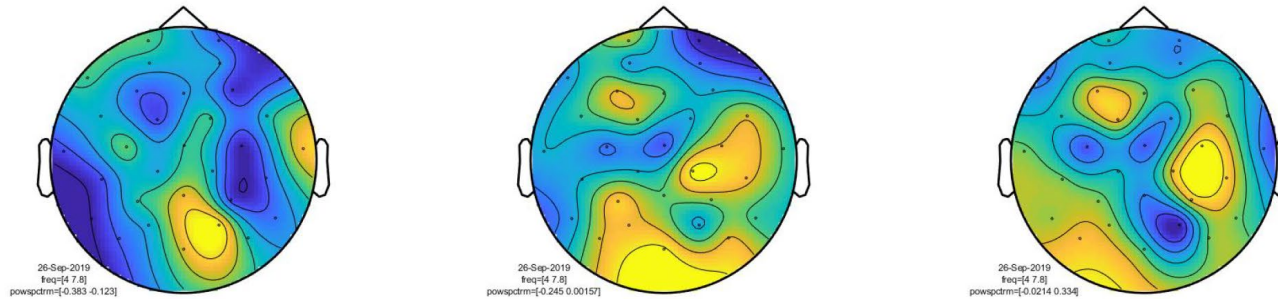
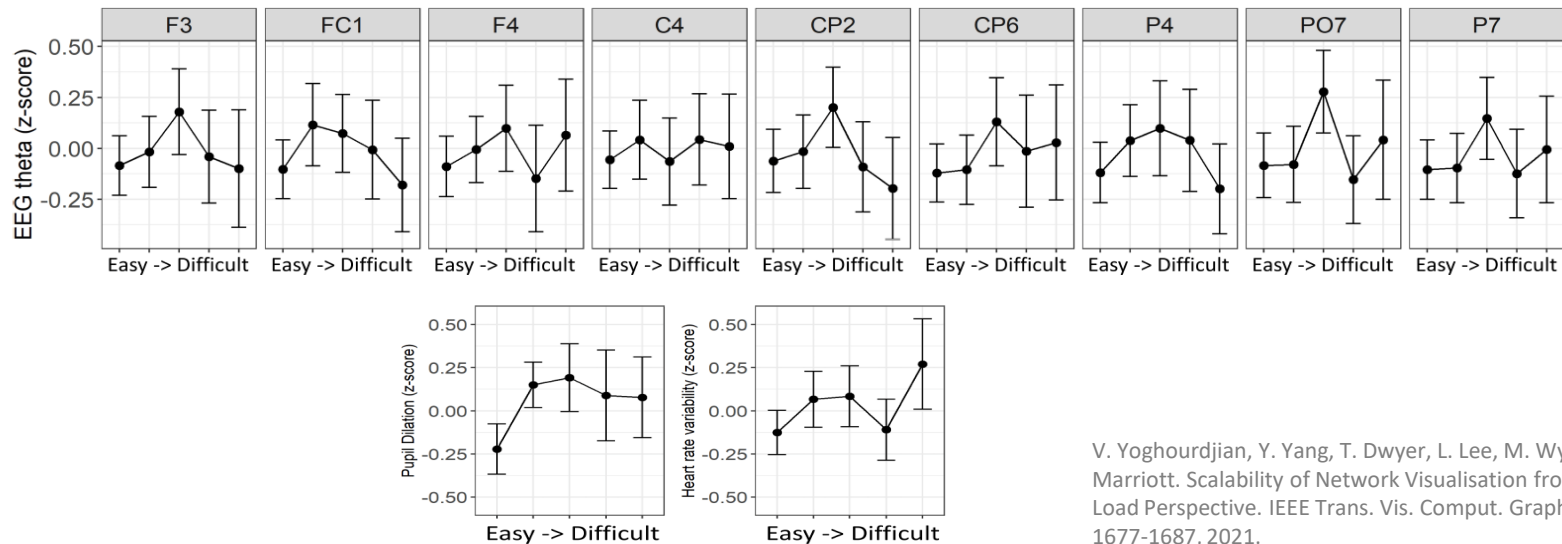


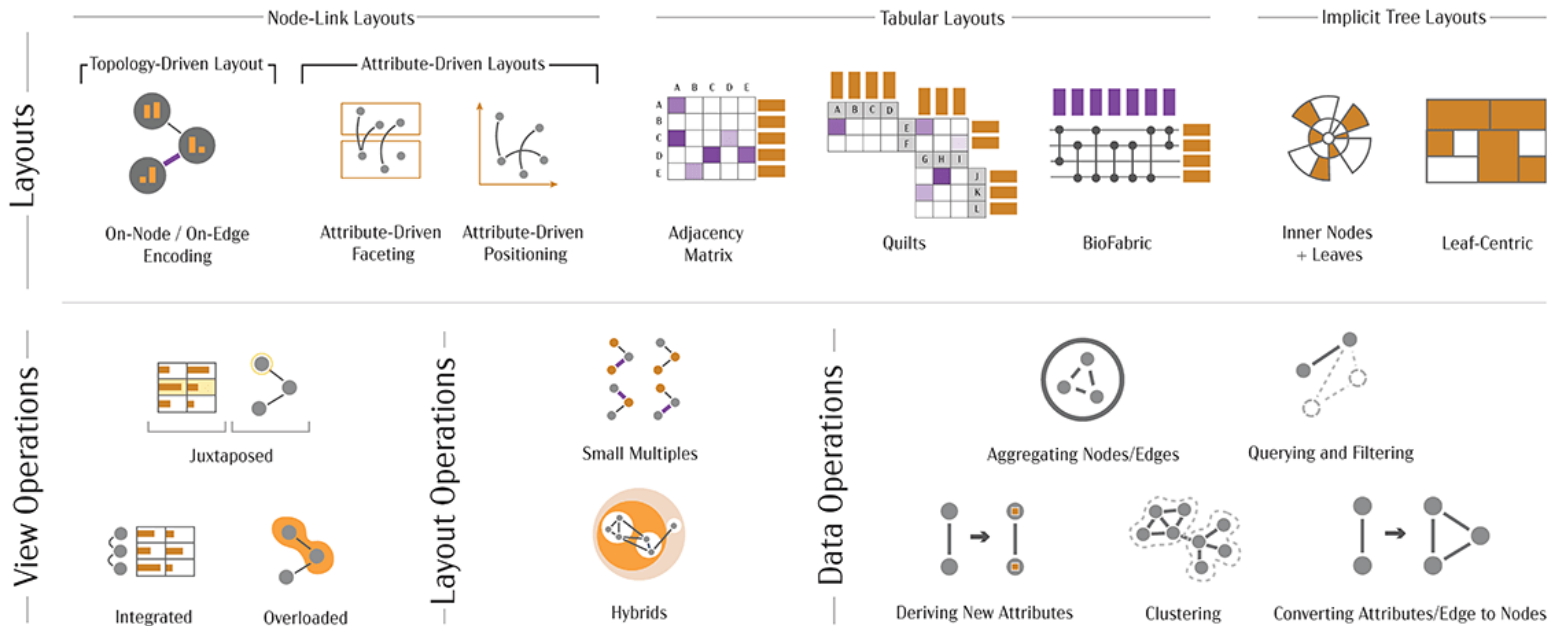
Fig. 1: EEG topographical maps of median theta brain activity for (from left-to-right) easy tasks, hard tasks and the difference (hard-easy) for participants undertaking path-finding exercises on network diagrams.



V. Yoghoudjian, Y. Yang, T. Dwyer, L. Lee, M. Wybrow, K. Marriott. Scalability of Network Visualisation from a Cognitive Load Perspective. IEEE Trans. Vis. Comput. Graph., 27(2): 1677-1687, 2021.

Visual Abstractions of Graphs

C. Nobre, M. D. Meyer, M. Streit, A. Lex. The State of the Art in Visualizing Multivariate Networks. Comput. Graph. Forum, 38(3): 807-832, 2019.



Human interaction?

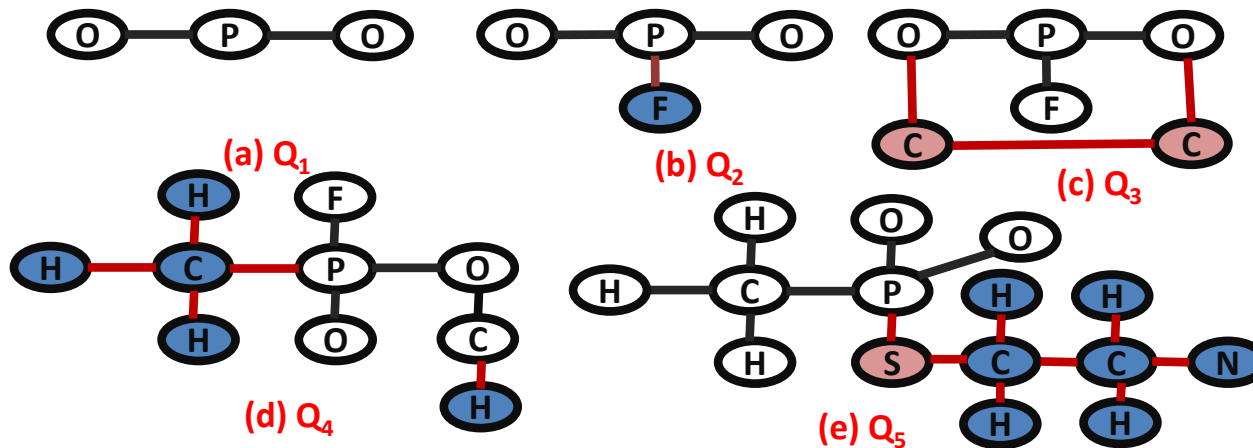
Scant systematic study in human interaction towards task-specific sensemaking



Interaction Through Exploratory Querying

Exploratory Querying

- Often search is not one-shot
- Open ended, evolving in nature
- Search state is ambiguous in the beginning
- Multiple and iterative query formulation and execution
- Initial query may often grow in size
- Visual query interface to facilitate exploration



Interaction Through Exploratory Querying

PICASSO [VLDB 2017] & FERRARI [VLDB J, 2020]

- Large collection of small- or medium-sized graphs
- **LAG** representation

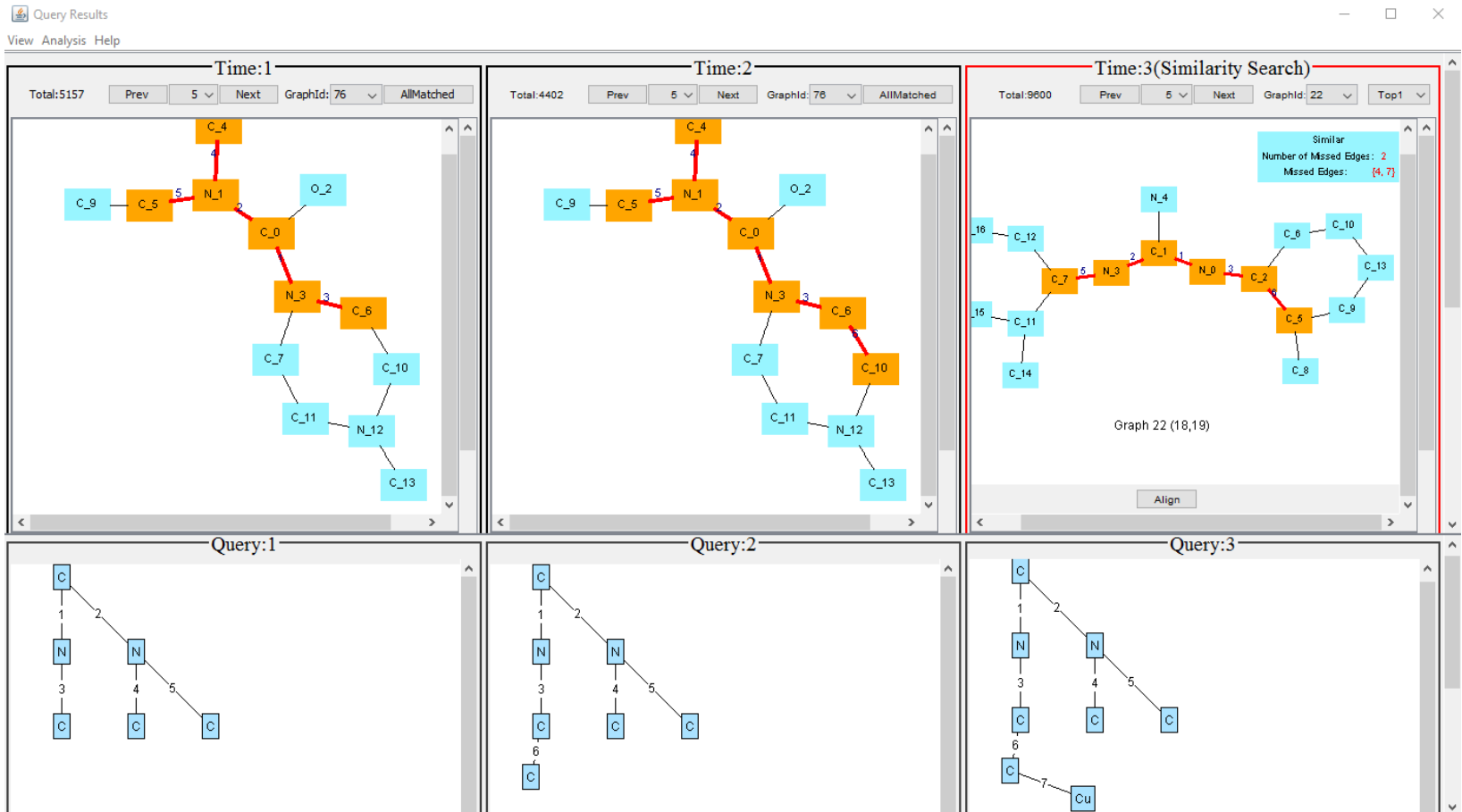
- ⌚ Offline index
 - ⌚ **VACCINE**
 - ⌚ Indexes frequent & discriminative infrequent fragments (DIFs)
- ⌚ Online index
 - ⌚ **ADVISE**
 - ⌚ Concise storage of candidates
 - ⌚ Support exact and similarity search

Kai Huang, Sourav S. Bhowmick, [Shuigeng Zhou](#), Byron Choi: **PICASSO: Exploratory Search of Connected Subgraph Substructures in Graph Databases**. *Proc. VLDB Endow.* 10(12), 2017.

[Chaohui Wang](#), Miao Xie, Sourav S. Bhowmick, Byron Choi, [Xiaokui Xiao](#), [Shuigeng Zhou](#): **FERRARI: an efficient framework for visual exploratory subgraph search in graph databases**. *VLDB J.* 29(5), 2020.



Querying



Kai Huang, Sourav S. Bhowmick, [Shuigeng Zhou](#), [Byron Choi](#): **PICASSO: Exploratory Search of Connected Subgraph Substructures in Graph Databases.** *Proc. VLDB Endow.* 10(12), 2017.



Interaction Through Graph Summaries

Graph “Summaries”

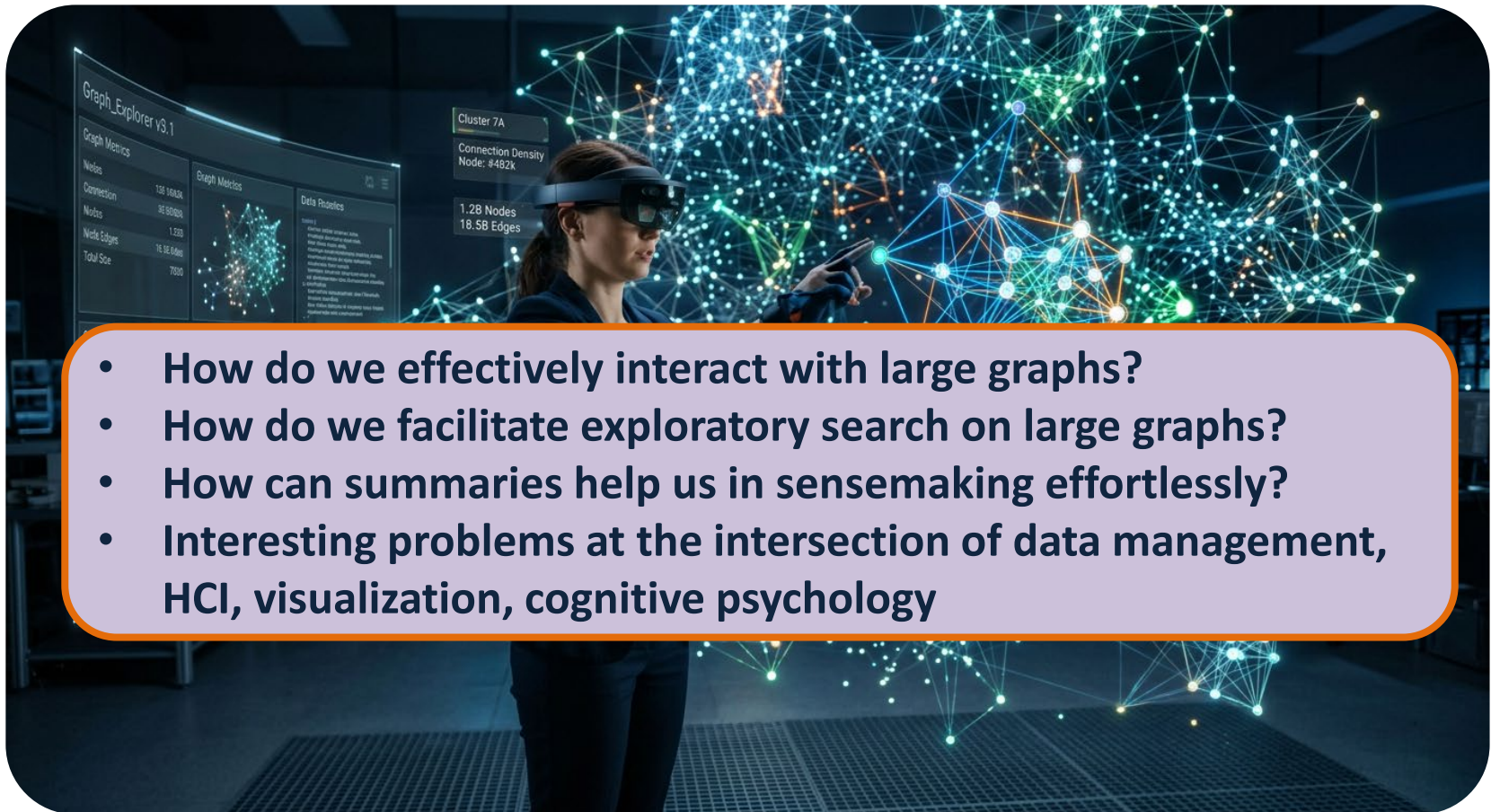
- Diverse definitions of summaries
- Numerous efforts in efficient graph summary construction
- Use cases: Storage, query processing, cardinality estimation
- Our past tutorial in VLDB 2017 

Human interaction?

- Scant systematic research on how humans can interact with these summaries to facilitate exploration and sensemaking



Open Challenges



- How do we effectively interact with large graphs?
- How do we facilitate exploratory search on large graphs?
- How can summaries help us in sensemaking effortlessly?
- Interesting problems at the intersection of data management, HCI, visualization, cognitive psychology

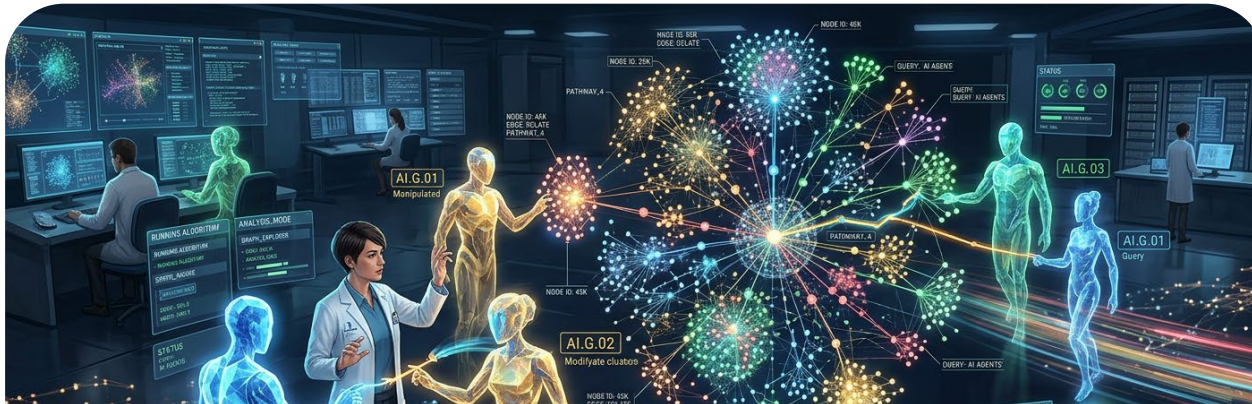
Disclaimer: The image was generated using Google Gemini



From Humans to Agents..

Human Interaction with Graph Data

- Enable humans to explore and retrieve information



Memory graph interactions enable agents to explore and retrieve contextual knowledge.

Disclaimer: The image was generated using Google Gemini



Summary

	GRAPH SCHEMA		GRAPH QUERY		GRAPH DATA		
HDI DIMENSIONS	LSG	LICS	LAG	LCG	Graph	Expl. Search	Summary
Syntecticity	P	★	P	★	×	★	N.A
Clarity	No	★	No	★	No	No	N.A
Informativity	P	★	P	★	No	★	N.A
Intuitivity	P	★	P	★	No	★	N.A
Elegance	No	★	No	★	No	★	N.A
Attractiveness	No	★	No	★	No	★	N.A
Usability	P	★	P	★	×	★	N.A
Intuitive Feedback	No	Yes	No	Yes	No	Yes	N.A

Recommendation	Evaluation criteria
Syntecticity	Minimalism of visual information elements opposed to their redundancy.
	Lightness of information elements opposed to their density.
Clarity	Adopt aesthetic criteria in drawing diagrammatic representations, e.g., no crossings between lines or symmetry for displaying hierarchies.
Informativity	Capability of representing all of the relevant aspects of the reality of interest.
	Quantity of information conveyed by an information means.
	Usage of templates that are familiar for lay people. Avoid any template that is so new that make the knowledge acquisition difficult.
Intuitivity	The information is well-organized in terms of context.
	The visualization is capable of conveying all the properties of the reality of interest "at a glance."
Elegance	Orderly and clear design, consistent and structured layout, symmetry, clean and clear design.
Attractiveness	Aesthetic aspect. The visualization is a piece of art per se.
	Capacity to attract users towards the perception of the information displayed in it without any relation to the efficiency and effectiveness in interaction and use.
Usability	Effectiveness – accuracy and completeness with which users can achieve specific goals.
	Efficiency – resources expended in relation to the accuracy and completeness with which users achieve goals.
	Satisfaction – freedom from discomfort and positive attitudes towards users of the product.
Intuitive feedback	The system allows for human's feedback about the presented information.
	The system allows for human's feedback that can be easily incorporated in the system.
	Human feedback is allowed in an intuitive form, e.g., in the form of natural language, colloquial descriptions of circumstances or discourse analysis etc.
Semanti	
Semantically enriched data search	Is there any structured or semantically enriched data search mechanism?
Preprocessing enrichment of data	

E. Z. Victorelli, J. C. dos Reis, H. H. Hornung, A. B. Prado. Understanding Human-data Interaction: Literature Review and Recommendations for Design. Int. J.Hum. Comput. Stud., 134: 13-32, 2020.



Next..

HUMAN-GRAPH AND AGENT-GRAPH INTERACTION:

presentation overview & structural breakdown

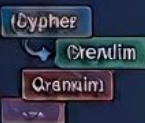
PART 1: HUMAN-GRAPH INTERACTION (HGI)

1. INTERACTION WITH GRAPH SCHEMA



- Node types
- Node nodes
- Node labels

2. INTERACTION WITH GRAPH QUERIES



Intent mapping and builders

3. INTERACTION WITH GRAPH DATA



Visual exploration & data scrubbing

HUMAN INPUT &
FEEDBACK LOOP

AGENT
AUTONOMY &
DETERMINISTIC
EXECUTION

AGENT | INPUT & DETERMINISTIC
EXECUTION

PART 2: AGENT-GRAPH INTERACTION (AGI)

1. INTERACTION WITH TASK PLANNING GRAPH



2. INTERACTION WITH TASK EXECUTION GRAPH



Runtime workflows, State management

3. INTERACTION WITH MEMORY GRAPH



4. INTERACTION WITH AGENT COORDINATION GRAPH

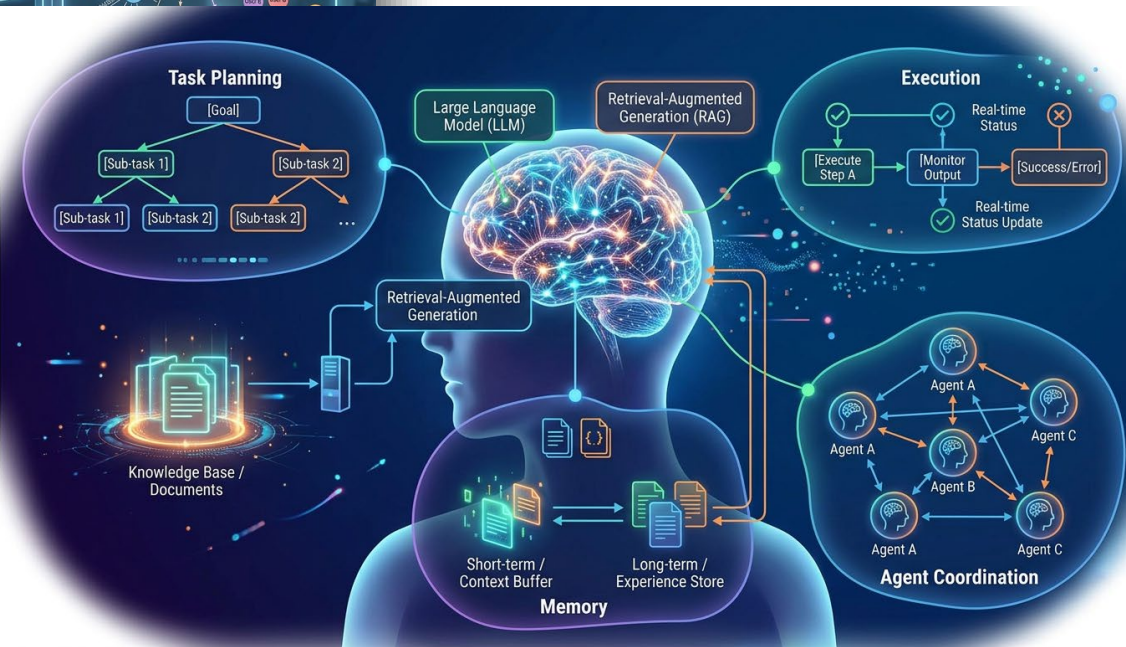
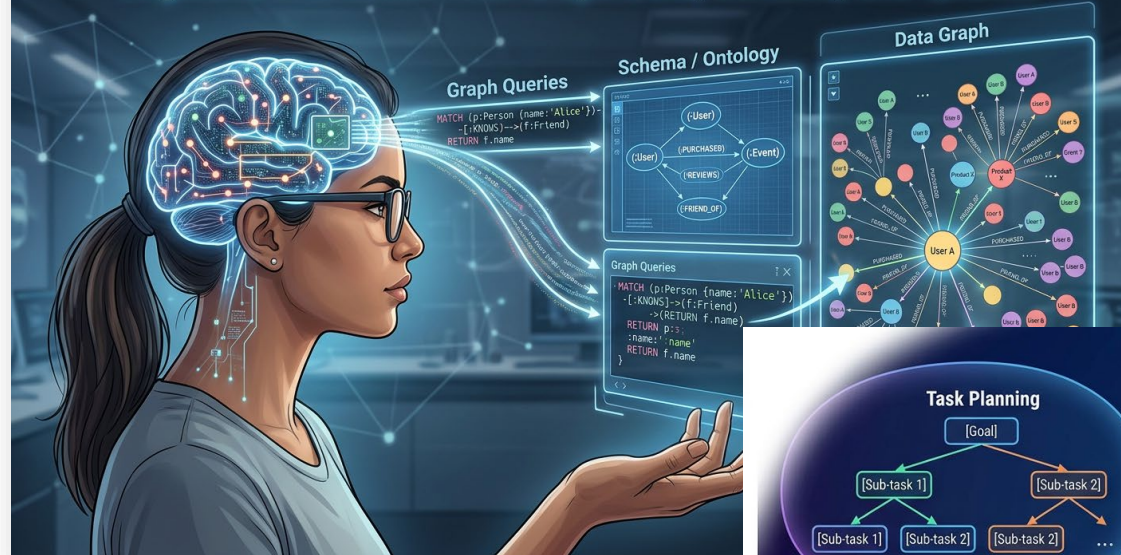


PART 3: FUTURE RESEARCH DIRECTIONS & CONCLUSIONS



Agent-Graph Interaction

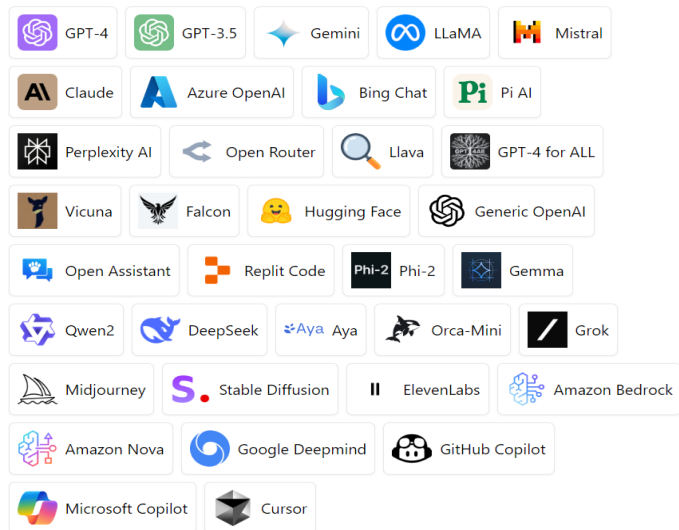
Cognitive Interaction with Graph Data, Schema, and Queries



Disclaimer: The images were generated using Google Gemini



Large Language Models: The Brain of Agents



LLM models

- **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.

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

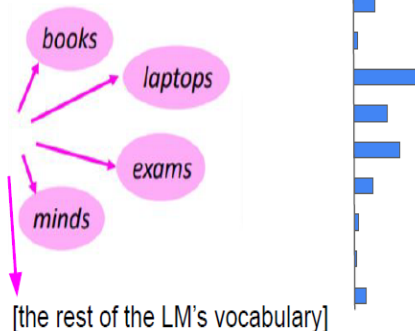
$$p(w) = p(w_1) \times p(w_2|w_1) \times p(w_3|w_1, w_2) \times p(w_l|w_1, \dots, w_{l-1})$$
$$= \prod_{t=1}^{|w|} p(w_t|w_1, \dots, w_{t-1})$$

The ____

The students ____

The students opened ____

The students opened their ____



Language models text generation



Benefits of LLMs

Features	Description
Conversations	Dialogs with chatbox may bring out new ideas
Distillations	Generates good summary
Fabrications	Can compose poetry or write essays using someone's style
Translations	Translate prompt text into another language
Speech	Good at real-time speech recognition
Discovering mindsets	Summarize the mindsets represented in a set of documents
Generating scenarios	Can generate a future scenario for a set of assumptions you specify (useful for military appl.)
Scientific discovery	Scientists use ANNs to take all data related to a phenomenon as input and make predictions (e.g., AlphaFold)
Education	Scale question answering
Cyber security	Using LLMs for intrusion detection
Coding	Rapidly generates small programs and improves efficiency



Pitfalls of LLMs

Features	Description
Hallucinations	Made-up outputs
Probabilistic Retrieval	LLM outputs are statistical inferences for the continuation of the prompt based on probabilities of words and sequences in the training data.
Vibe coding	Errors in production and security vulnerabilities
Prompt injections	Hidden texts inserted into prompts instruct the LLM to subvert its own purpose
Jailbreaking	Users can trick LLM into violating its guardrails and expose prohibited information
Unexplainable outputs	LLMs cannot give reasonable explanations of how they arrived at their responses.
Loss of sensitive data	Most LLMs are accessed in the cloud
Poor quality data	Data contains many conflicts and biases



Pitfalls of LLMs

Features	Description
Data pollution	LLM outputs are routinely stored on the Internet, which become part of the training data
Scheming	The LLM hides its true objectives and pretends to be aligned with human-given goals
False friendships	Many humans believe LLM is a friend and consider it is “AI companion” or “AI therapists”
Prompt injections	Hidden texts inserted into prompts instruct the LLM to subvert its own purpose
Cyber security vulnerabilities	Great tool for criminals and troublemakers!

Peter J. Denning: [The Conundrum of LLMs](#). *Commun. ACM* 69(3): 31-34 (2026)



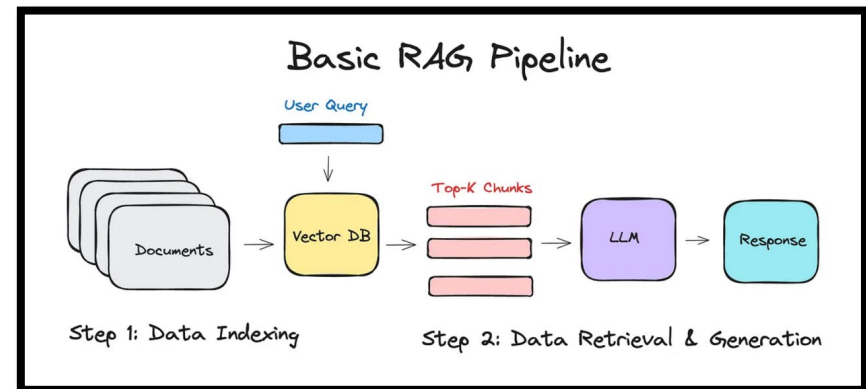
Retrieval-Augmented Generation (RAG)

RAGs

- Can enhance the capabilities of LLMs by integrating them with search or retrieval systems.
- Access up-to-date knowledge and external data source

Examples

- Microsoft Bing Chat
- GitHub Copilot
- Medical Question Answering



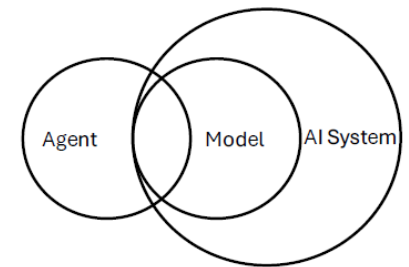
Agents – An Introduction

What is an AI Agent?

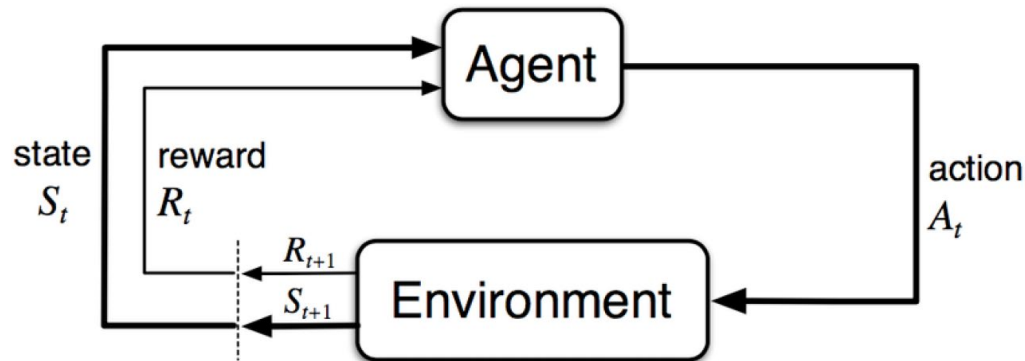
- **Agent** is an autonomous entity that perceives its environment, makes decisions, and acts to achieve specific goals.
- Agents can interact with users, other agents, or systems.

Examples

- Virtual Personal Assistants: Siri, Alexa, Google Assistant, Open Claw
- Autonomous Vehicles: Waymo, Tesla's Autopilot
- Game Non-Player Characters (NPCs): Chess AI (Stockfish)



Agents



What would we like an agent to do?

- ACT!
- Agentic systems autonomously make decisions and take actions to achieve a specified goal, operating within the constraints of their environment and design.



Components of An Agent

Required

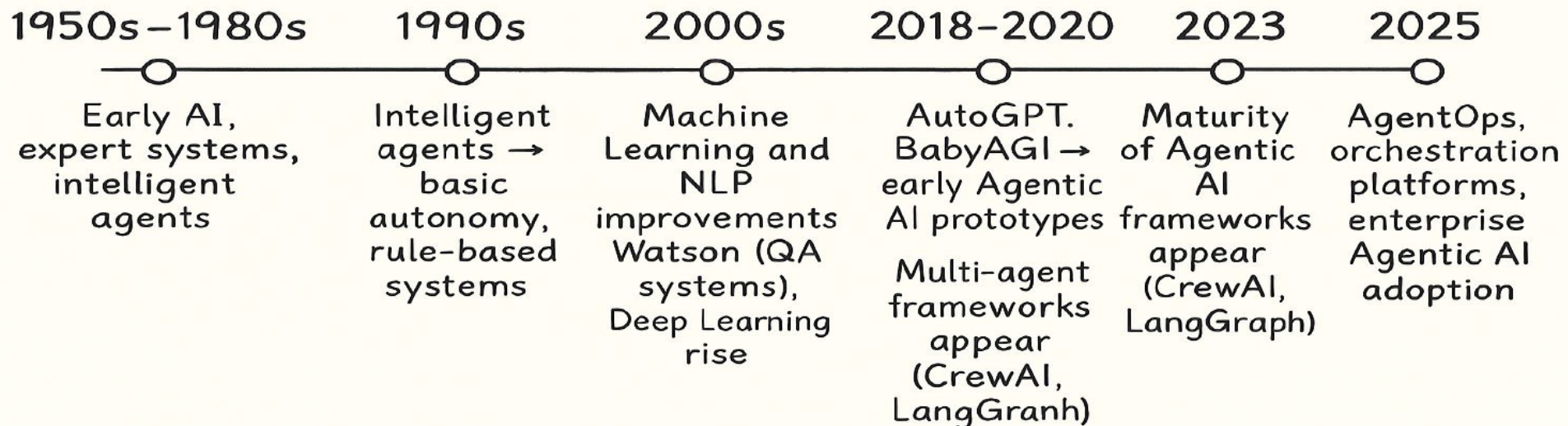
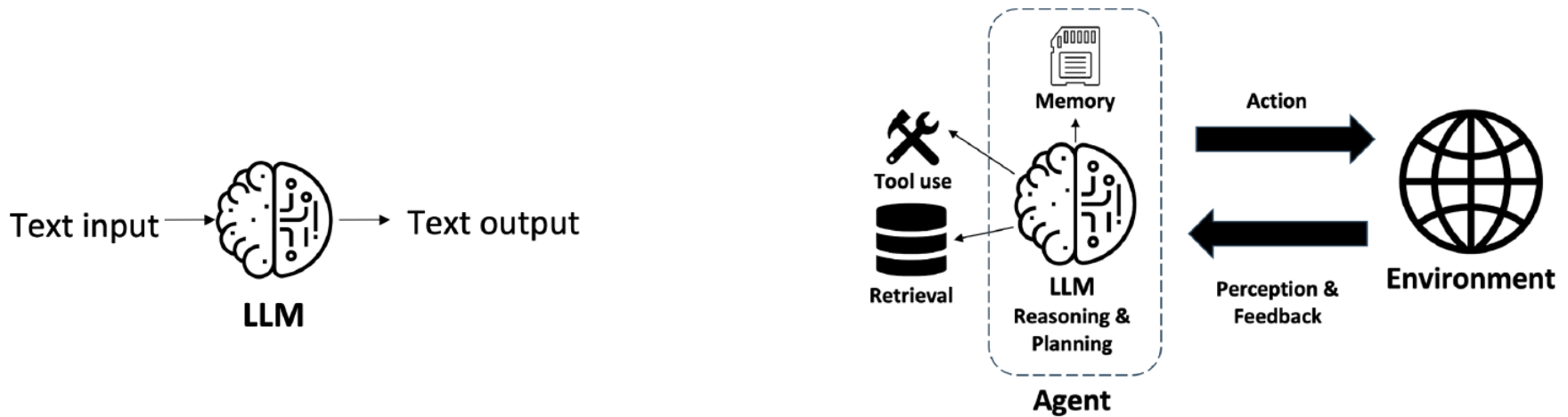
- Grounding, Agency, Planning, Memory, Learning

Additional

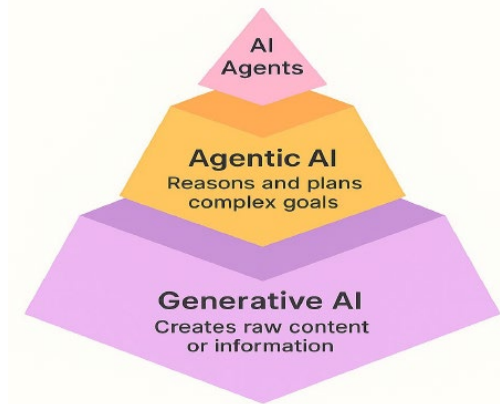
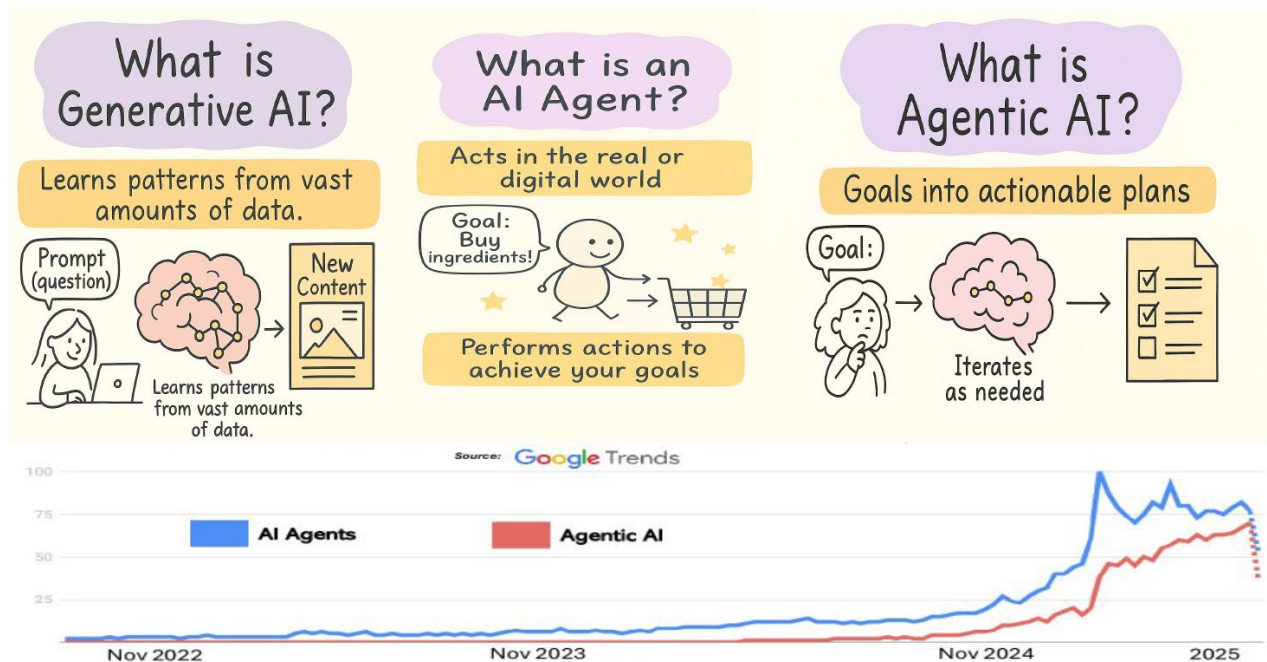
- Embodiment, Communication, World Modeling, Multimodality



AI Agents



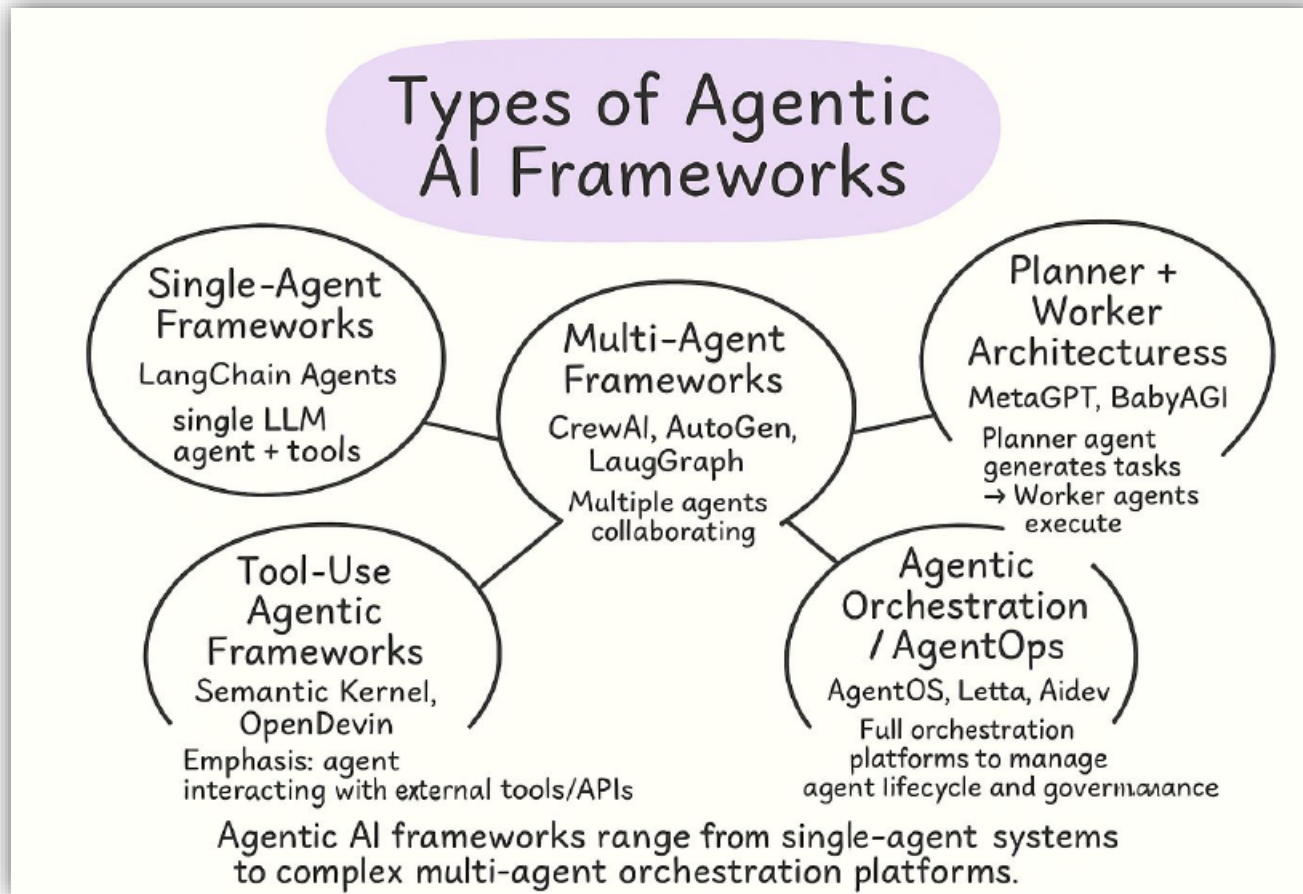
Agentic Terminology



- **Generative AI** produces content such as text, code, or intermediate reasoning steps.
- **Agentic AI** refers to architectures that use these models to plan, reason, and orchestrate multi-step workflows.
- **AI Agents** are the concrete entities that carry out those workflows by invoking tools, interacting with environments, and executing actions.



Agentic AI Frameworks



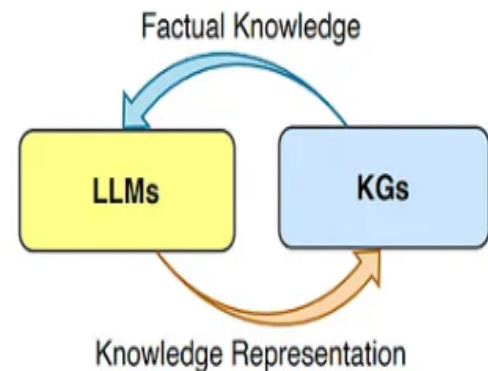
Role of Knowledge Graphs (KGs)

Knowledge Graphs

A knowledge graph keeps a structured network of entities (such as people, places, or things) and their fact/relationships.

Role of KGs

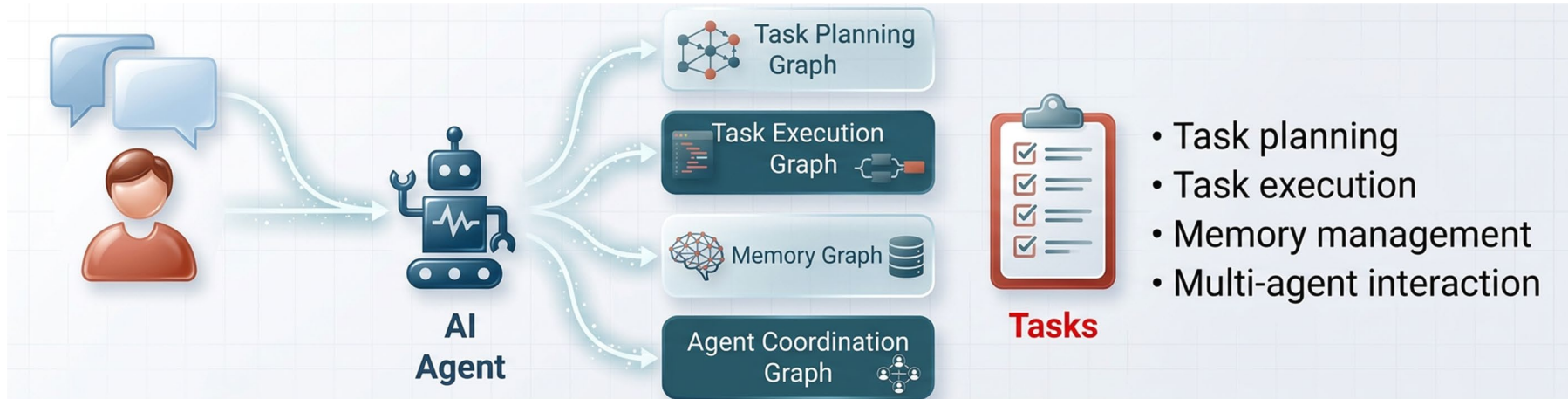
Knowledge graphs enable LLMs, AI Agents, and machines to understand and utilize the complex interconnected information.



Scope of AGI

Agent-Graph Interaction

- Focus on agents interacting with graph abstractions
- Not LLMs/RAGs interacting with graphs



Tutorials on LLMs and Graphs

QA, LLMs, KG

- Danqi Chen and Wen tau Yih. 2020. Open-domain question answering. In ACL. 34–37.
- Lihui Liu, Zihao Wang, Jiaxin Bai, Yangqiu Song, and Hanghang Tong. 2024. New frontiers of knowledge graph reasoning: Recent advances and future trends. In WWW. 1294–1297.
- Jizhi Zhang, Keqin Bao, Yang Zhang, Wenjie Wang, Fuli Feng, and Xiangnan He. 2024. Large language models for recommendation: Progresses and future directions. In WWW Companion (2024). 1268–1271.

LLMs+KGs/Graphs, RAG

- Wenqi Fan, Yajuan Ding, Liangbo Ning, Shijie Wang, Hengyun Li, Dawei Yin, Tat-Seng Chua, and Qing Li. 2024. A survey on RAG meeting LLMs: Towards retrieval-augmented large language models. In SIGKDD. 6491–6501.
- Chao Huang, Xubin Ren, Jiabin Tang, Dawei Yin, and Nitesh Chawla. 2024. Large language models for graphs: Progresses and directions. In WWW. 1284-1287.
- Qiang Zhang, Jiaoyan Chen, Zaiqiao Meng. 2024. Integrating Knowledge Graphs and Large Language Models for Advancing Scientific Research. Learning on Graph Conference (LoG).
- Chuangtao Ma, Yongrui Chen, Tianxing Wu, Arijit Khan, and Haofen Wang, "Unifying Large Language Models and Knowledge Graphs for Question Answering: Recent Advances and Opportunities ", in EDBT 2025.



Graphs and AI Agents

[Bei et al., <https://arxiv.org/abs/2506.18019>]

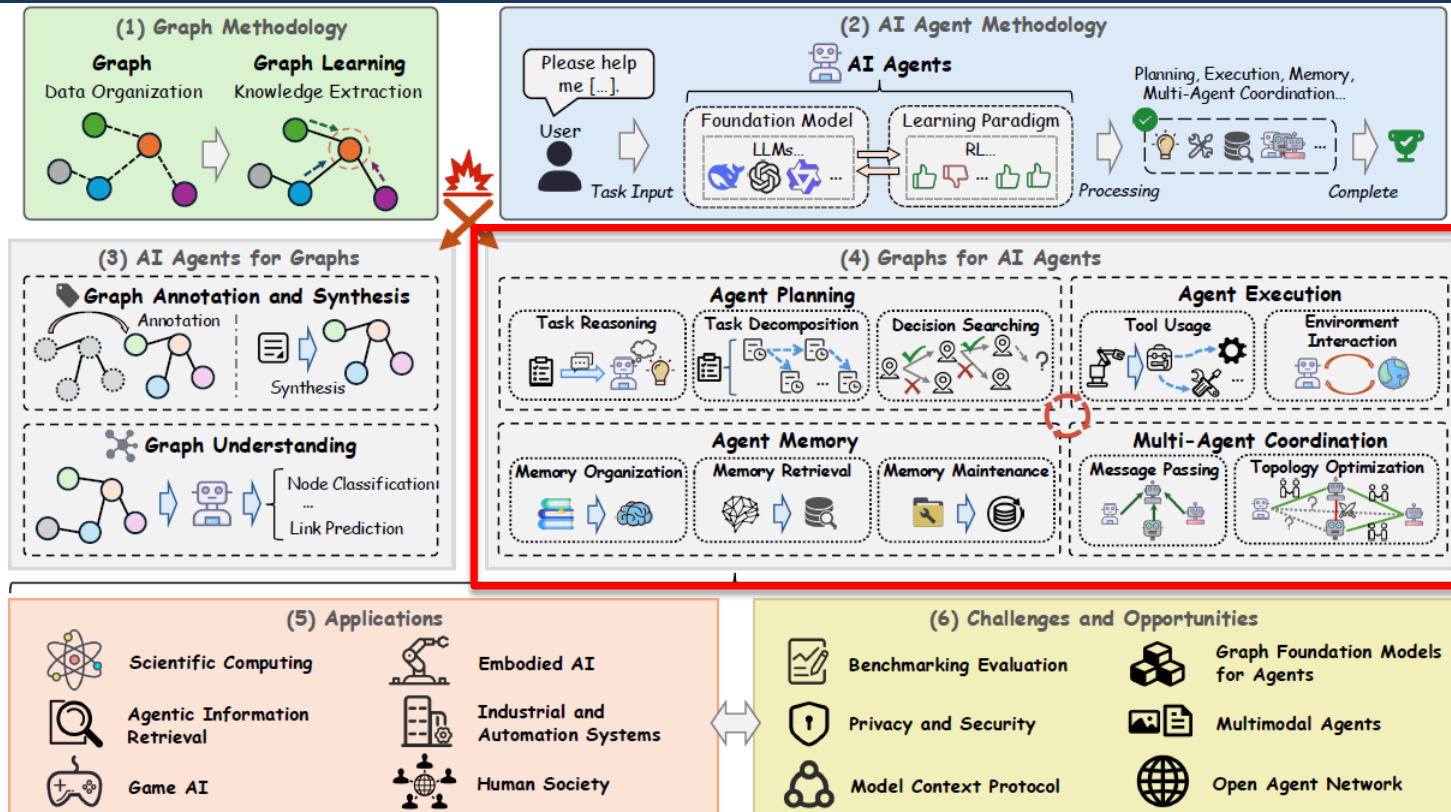


Fig. 1: An overall illustration of graphs meet AI agents. (1) Graph Methodology: graph data organization and knowledge extraction with graphs. (2) AI Agent Methodology: LLM-based foundation models and RL-based learning paradigms, forming core AI Agent pipelines. (3) AI Agents for Graphs: the powerful capabilities of agents in graph modeling and learning, such as graph annotation, synthesis, and understanding. (4) Graphs for AI Agents: the role and potential of graphs and graph learning in empowering the core functionalities of agents, including agent planning, execution, memory, and multi-agent coordination. (5) Representative Applications. (6) Challenges and Future Opportunities.

Agent Interaction with Task Planning Graphs

Planning

- Planning is a fundamental capability of AI agents
- Requiring them to interpret user requests and generate structured sequences of actions.
- Graph-based representations play a central role by supporting reasoning, task decomposition, and decision-making.

Types

- Knowledge Graphs for Task Reasoning
- Task dependency graph for task decomposition
- State space graph for decision search

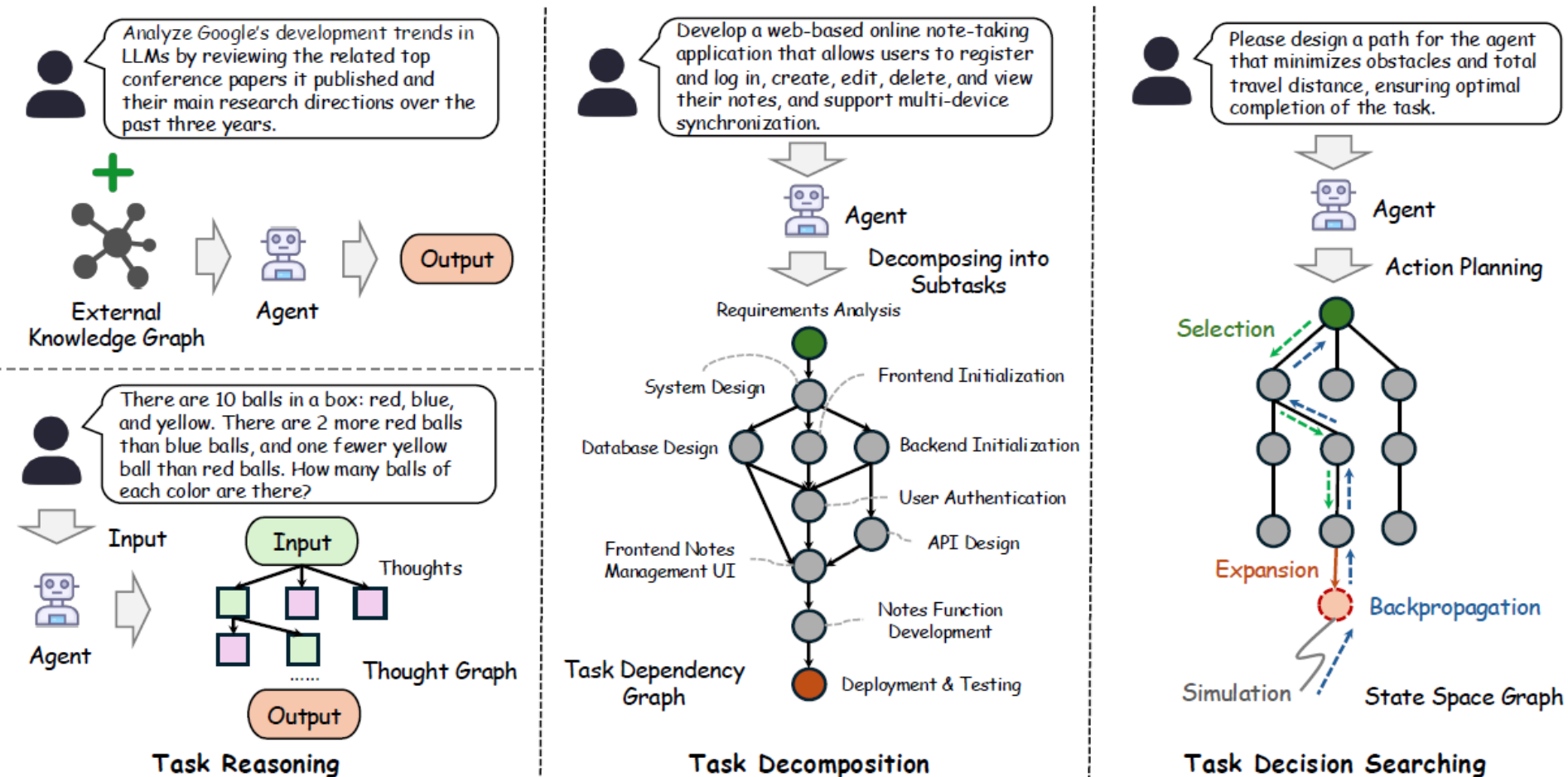


Agent Interaction with Task Planning Graphs

Graph	Agentic Interaction	Techniques
KG for task reasoning	Agents interact with KGs to retrieve relevant subgraphs based on task-specific entities and relationships, enabling deeper understanding and more informed reasoning	<i>ToG, KG-CoT, RoG, MindMap, PoG, GoT, Graph of Thoughts</i>
Task dependency graphs (TDG)	Agents use TDGs to determine execution order and identify feasible subtask sequences.	<i>AgentKit, VillagerAgent, DAG-Plan, GNN4TaskPlan</i>
State space graph (SSG)	Agents perform search over SSGs to identify action sequences that lead to desired goals.	<i>GBOP</i>



Graphs for Agentic Planning



<https://arxiv.org/pdf/2506.18019>

Fig. 3: An illustration of graphs for agent planning.



Graph-of-Thought Reasoning

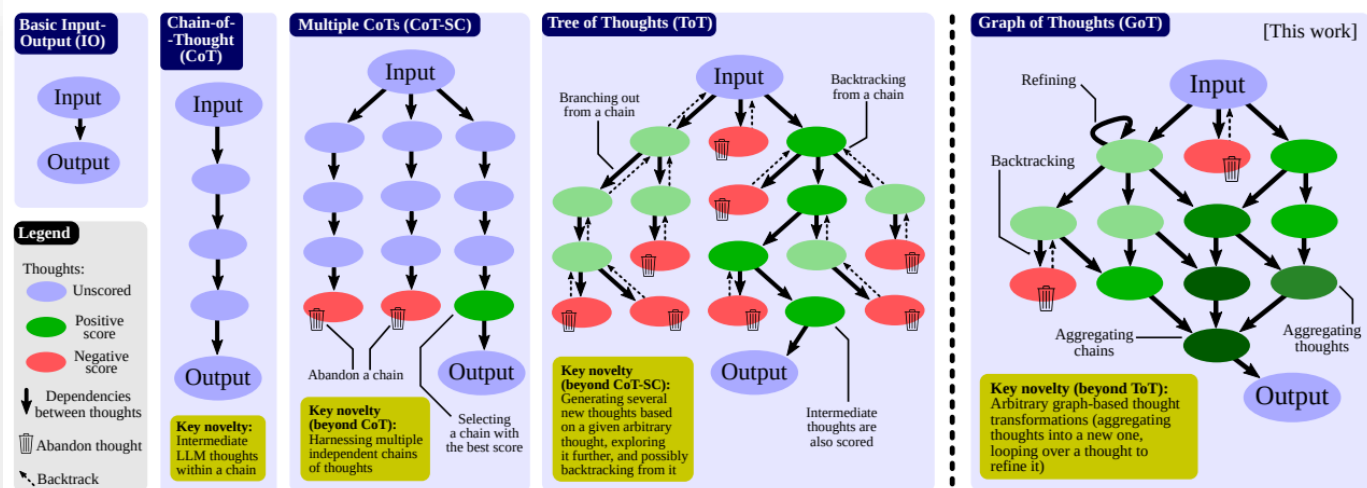


Figure 1: Comparison of Graph of Thoughts (GoT) to other prompting strategies.

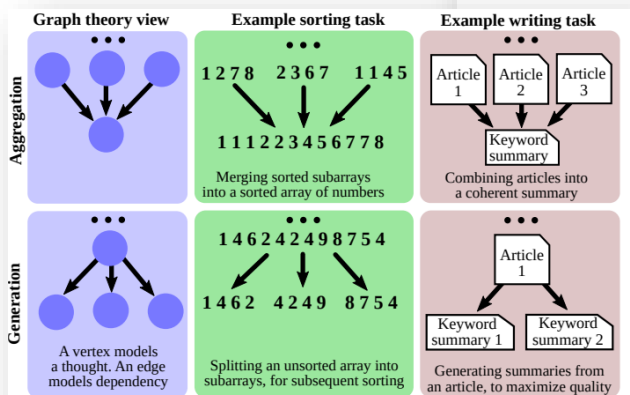


Figure 2: Examples of aggregation and generation thought transformations.

Graph of Thoughts: Solving Elaborate Problems with Large Language Models

Maciej Besta^{1*}, Nils Blach^{1*}, Ales Kubicek¹, Robert Gerstenberger¹,
 Michał Podstawski², Lukas Gianinazzi¹, Joanna Gajda³, Tomasz Lehmann³,
 Hubert Niewiadomski³, Piotr Nyczyk³, Torsten Hoeffler¹
¹ETH Zurich, ²Warsaw University of Technology, ³Cledar



Task Decomposition and Dependency Graph

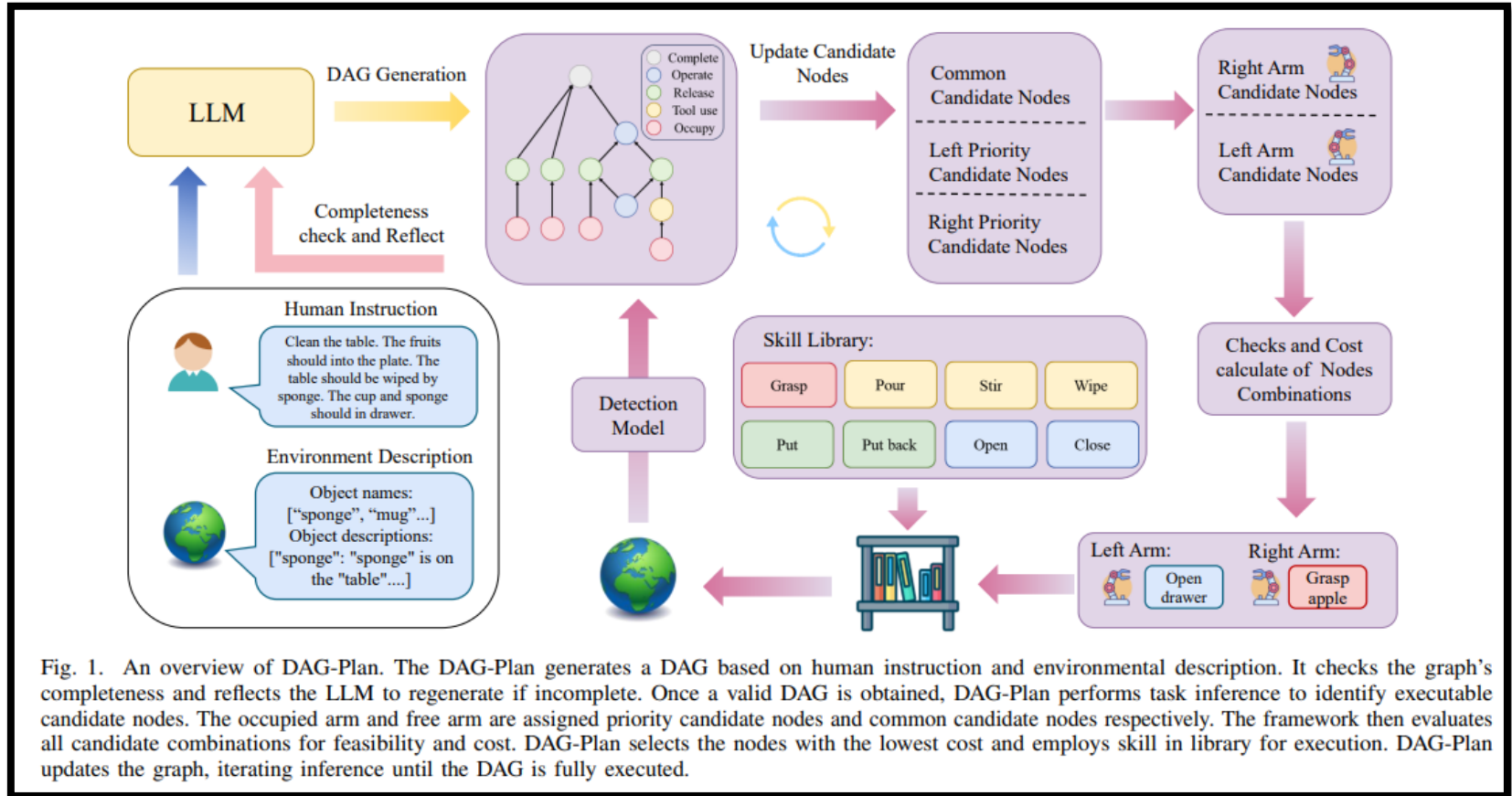


Fig. 1. An overview of DAG-Plan. The DAG-Plan generates a DAG based on human instruction and environmental description. It checks the graph's completeness and reflects the LLM to regenerate if incomplete. Once a valid DAG is obtained, DAG-Plan performs task inference to identify executable candidate nodes. The occupied arm and free arm are assigned priority candidate nodes and common candidate nodes respectively. The framework then evaluates all candidate combinations for feasibility and cost. DAG-Plan selects the nodes with the lowest cost and employs skill in library for execution. DAG-Plan updates the graph, iterating inference until the DAG is fully executed.

Z. Gao, Y. Mu, J. Qu, M. Hu, L. Guo, P. Luo, and Y. Lu, "Dag-plan: Generating directed acyclic dependency graphs for dual-arm cooperative planning," arXiv preprint arXiv:2406.09953, 2024

Task Decomposition and Dependency Graph

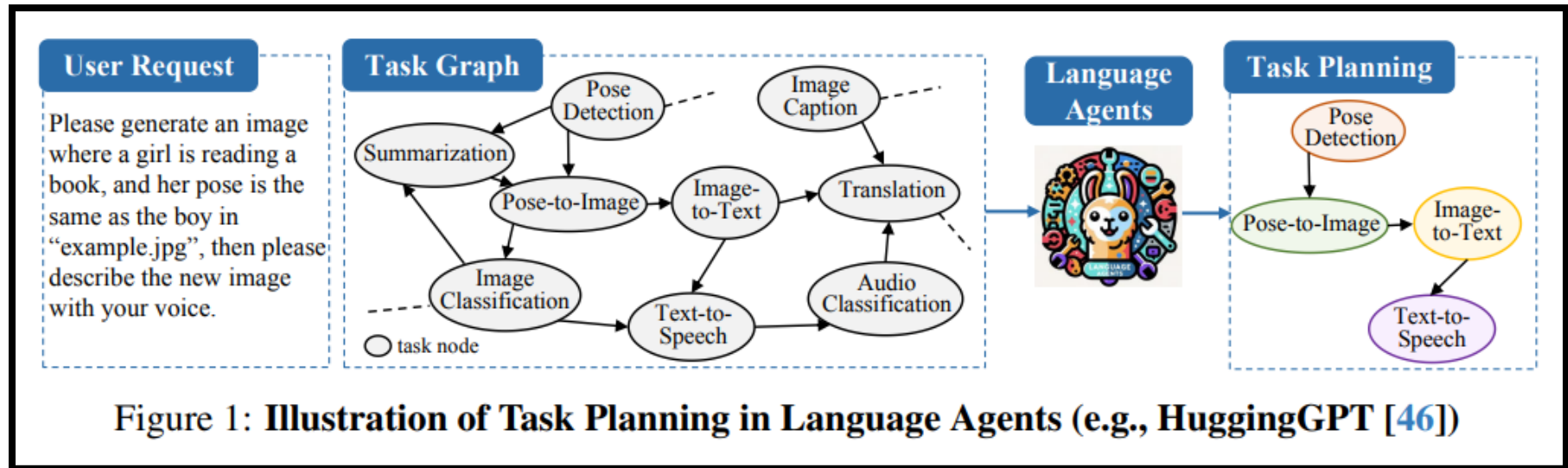


Figure 1: Illustration of Task Planning in Language Agents (e.g., HuggingGPT [46])

X. Wu, Y. Shen, C. Shan, K. Song, S. Wang, B. Zhang, J. Feng, H. Cheng, W. Chen, Y. Xiong et al., “Can graph learning improve planning in llm-based agents? NeurIPS 2024.



Benchmarking Agentic Workflow Generation [ICLR 2025]



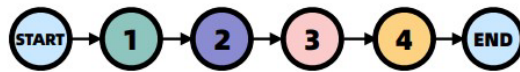
Check if the emails admin@site.com and manager @site.com are disposable and get their MX records.



Planning Steps:

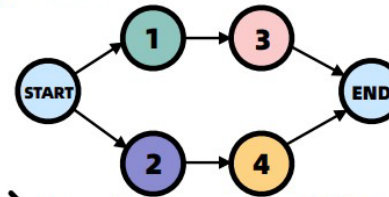
- 1: Check if the email admin@site.com is disposable
- 2: Check if the email manager@site.com is disposable
- 3: Get the MX records for the email admin@site.com
- 4: Get the MX records for the email manager@site.com.

Linear Workflow



→ Sequential Execution **Inefficient**

Graph Workflow



↻ Parallel Execution **Efficient**

The Application of Workflow



Chain-of-Thought Augmentation



Parallel Planning Steps



Structured Prior Knowledge

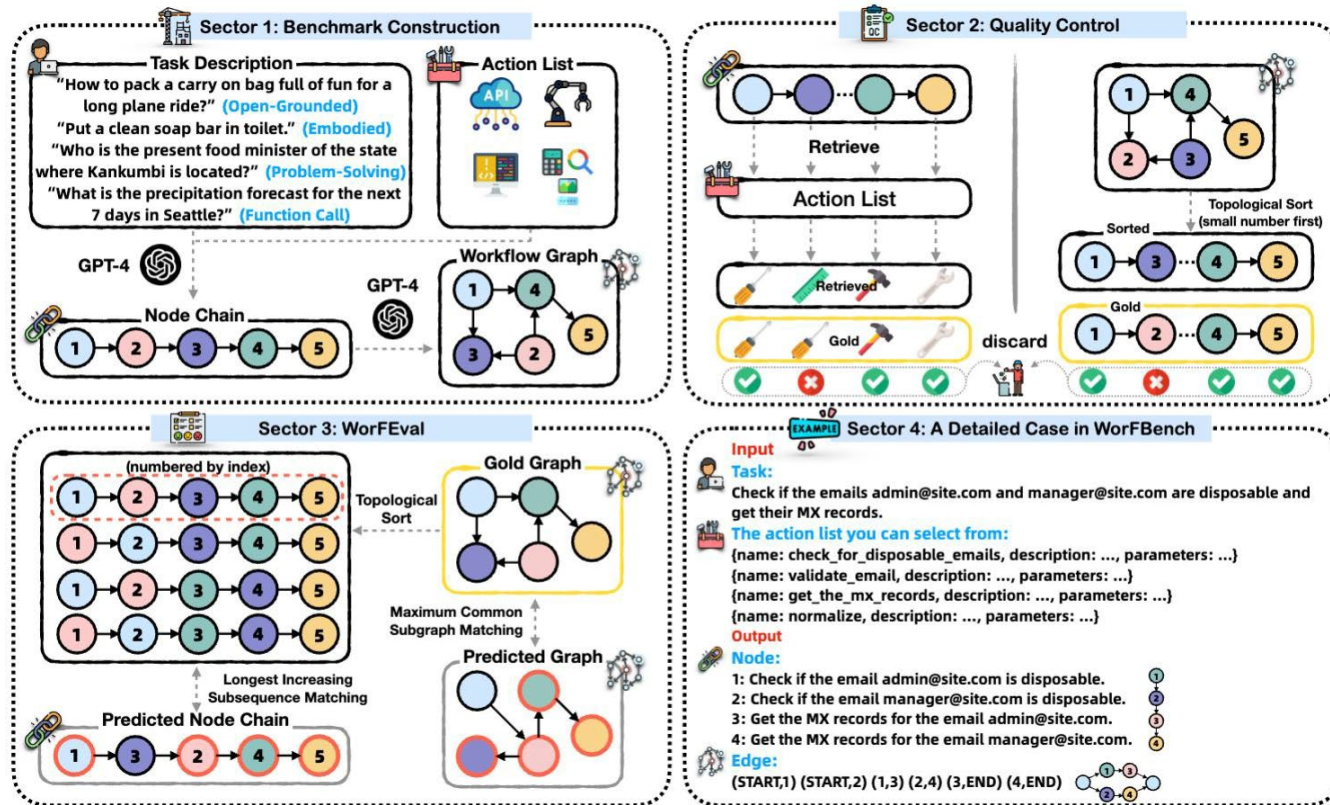


Shorten Planning Steps

[1] <https://arxiv.org/pdf/2410.07869>



Benchmarking Agentic Workflow Generation [ICLR 2025]



- Multi-faceted scenarios.
- Complex workflow structures.
- Strict quality control and data filtering.
- Accurate quantitative evaluation (WorFEval).



Agent Interaction with Task Execution Graphs

Task Execution

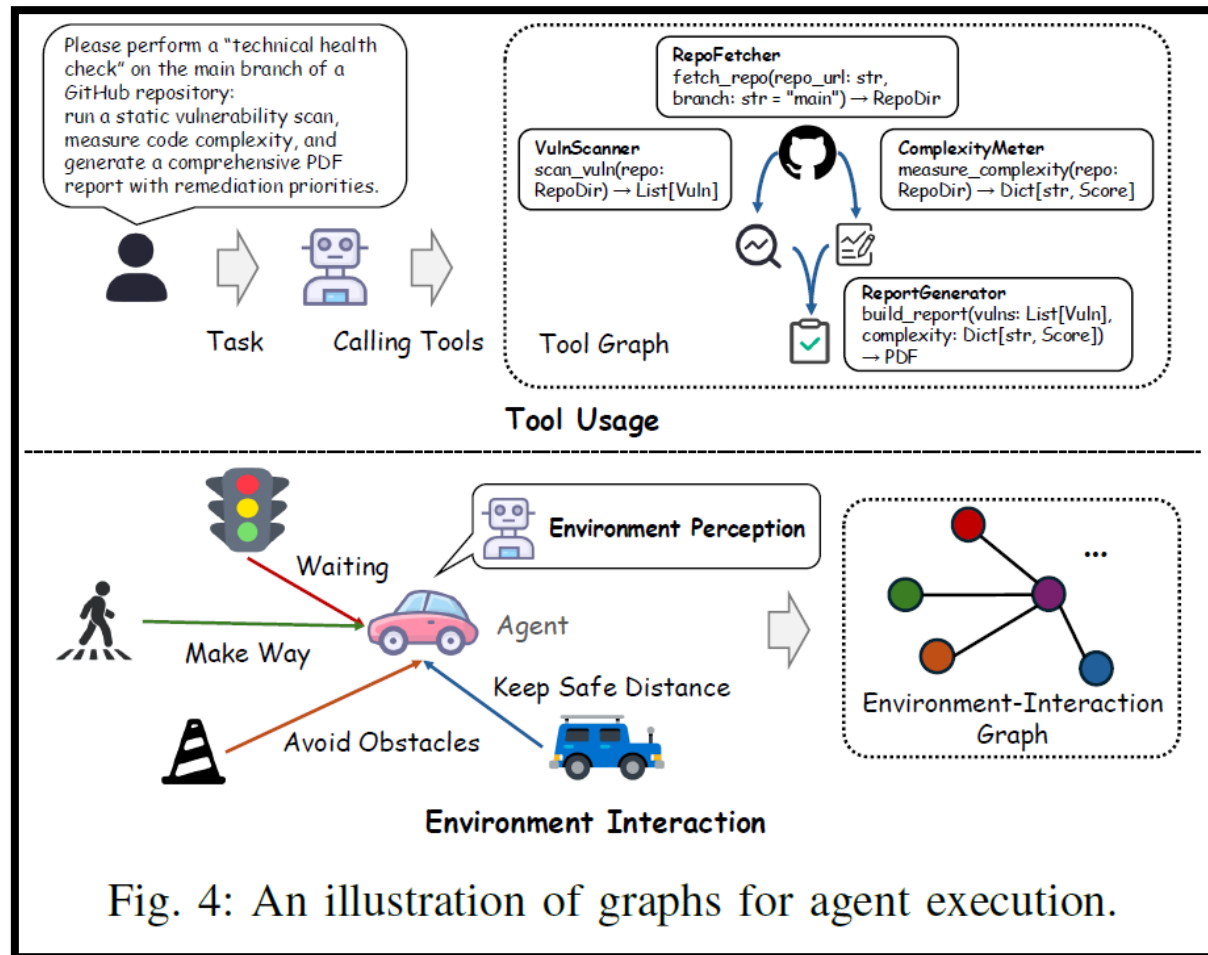
- The execution phase of an AI agent involves carrying out planned actions by leveraging both **external tools** and interactions with its **environment**.
- Graph-based representations play a crucial role in organizing these processes and enabling efficient execution.

Types

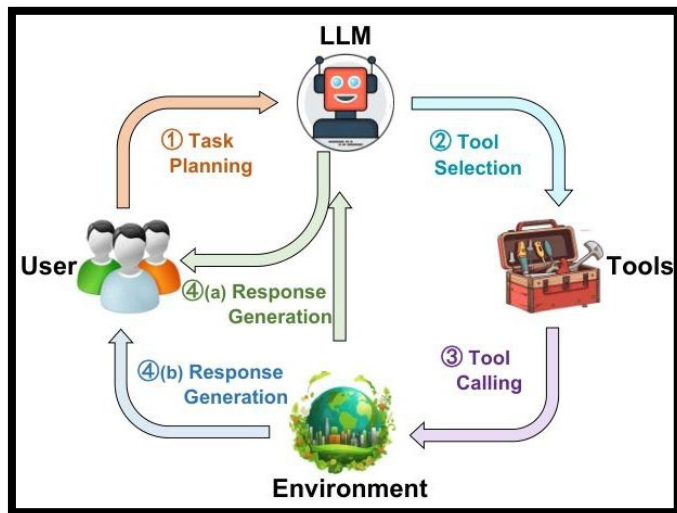
- Tool Graph for tool usage
- Environment-interaction Graph



Graphs for Agentic Execution



The Problem with Current LLM Tool Learning



Critical Gaps in Current Evaluation



Graph Blindness

Fails to assess understanding of task dependency graphs, leading to ~30% hallucination rates in advanced models like GPT-4-turbo.



Conflict Ignorance

Only validates final answers, ignoring the critical ability to detect contradictions between tool outputs and internal knowledge.



Inefficient Testing

Relies on slow multi-turn dialogues instead of evaluating one-shot planning, making it hard to assess proactive reasoning.



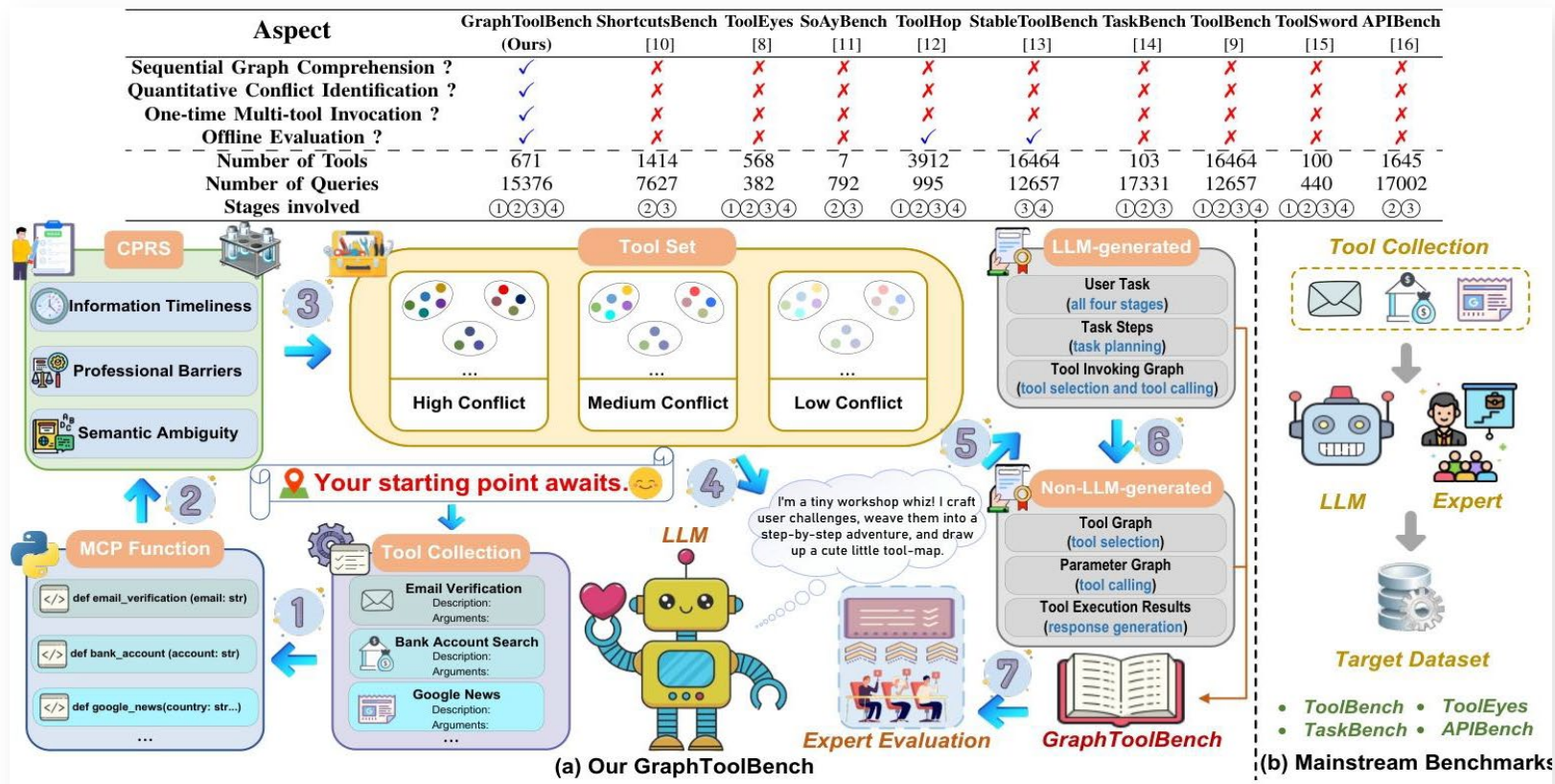
GraphToolBench

A comprehensive offline benchmark designed to rigorously evaluate sequential graph comprehension and quantitative conflict identification—closing the gaps left by traditional evaluation methods.

Feiyang Li, **Arijit Khan**, Zhan Shi, Peng Fang, Fang Wang, and Dan Feng, "GraphToolBench: Benchmarking LLMs for Sequential Graph Comprehension and Conflict Identification in Tool Learning", under submission.



GraphToolBench



The GraphToolBench architecture bridges dynamic tool interaction with rigorous evaluation. By converting real-world APIs into stable, offline components, it establishes a reproducible benchmark for testing large language models (LLMs) on complex, multi-step tool-learning tasks, from planning to execution.



Agent Interaction with Memory Graph

Role of Memory

- Memory enables AI agents to store and recall experiences, improving accuracy and adaptability over time.
- Graph-based memory structures capture relationships among information, enhancing pattern recognition and long-term reasoning.
- Knowledge graphs are used for long-term memory, often organized hierarchically to support both fine-grained details and high-level abstractions.

Types

- Graphs for memory retrieval
- Graphs for memory maintenance



Agent Interaction with Memory Graph

Graphs for Memory Retrieval

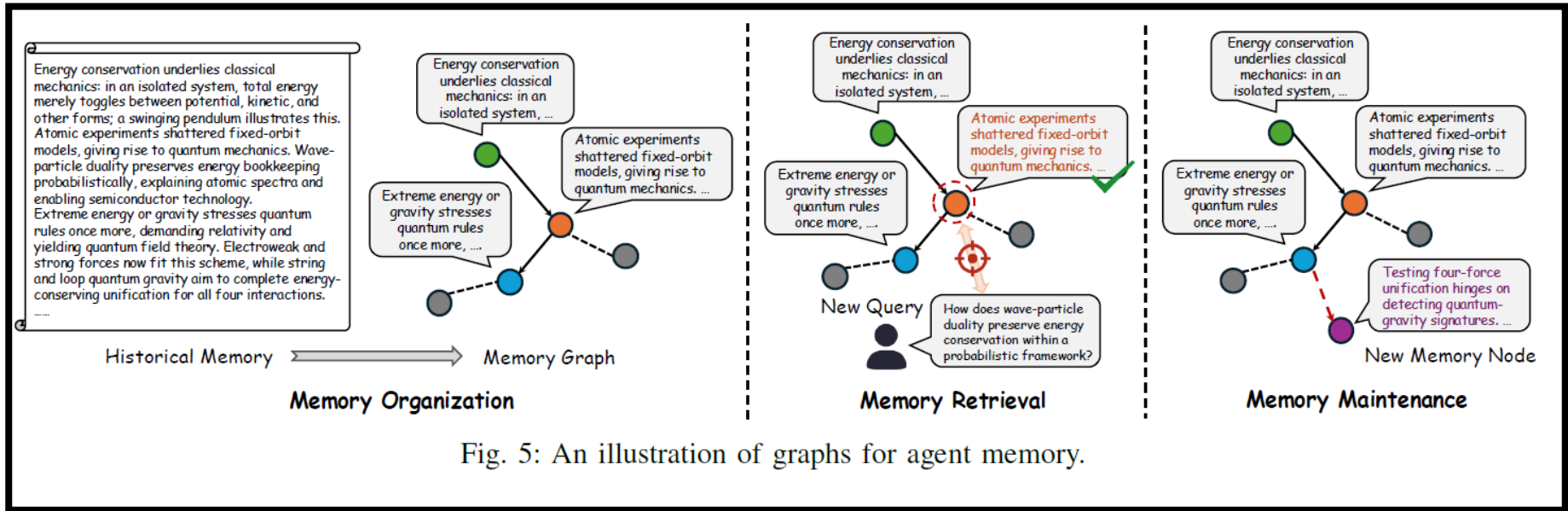
- Graph-based RAG techniques allow agents to efficiently access relevant information.
- Improve accuracy by identifying meaningful subgraphs using semantic and structural signals
- Enhance efficiency through optimized subgraph selection and path pruning, reducing computational cost and latency

Graphs for Memory Maintenance

- Focuses on dynamically updating graph structures as new information becomes available



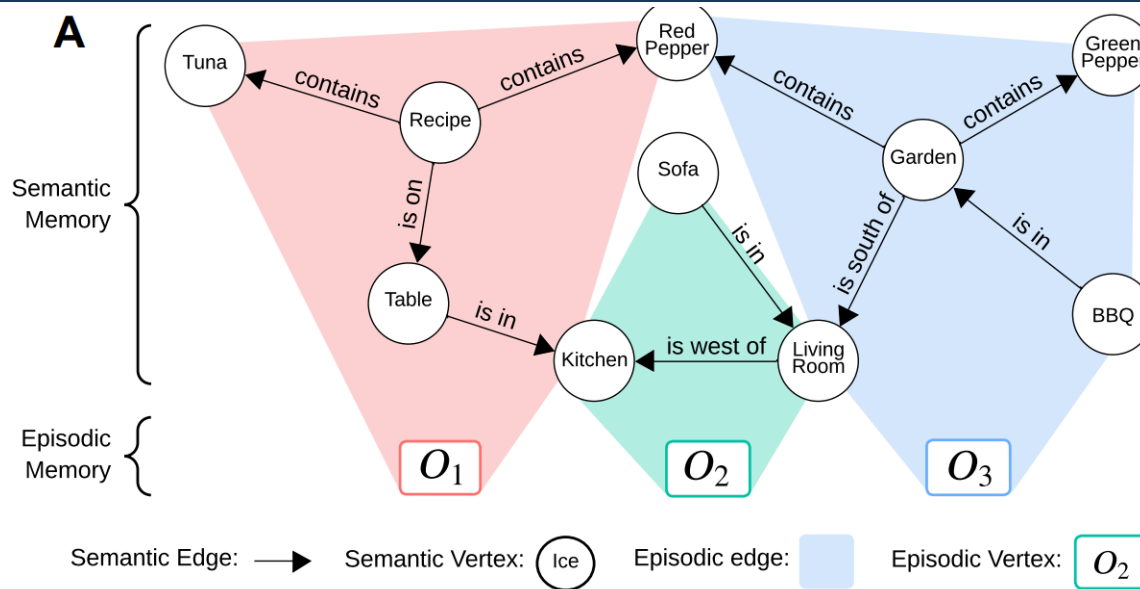
Graph-based Agentic Memory



<https://arxiv.org/pdf/2506.18019>



AriGraph: Learning Knowledge Graph World Models with Episodic Memory for LLM Agents [IJCAI'25]



At each step:

1. The agent receives a textual observation.
2. LLM extracts semantic triplets: (object1, relation, object2)
3. Triplets update the semantic memory graph.
4. The original observation is stored as an episodic vertex.
5. Episodic edges link the observation to the extracted triplets.



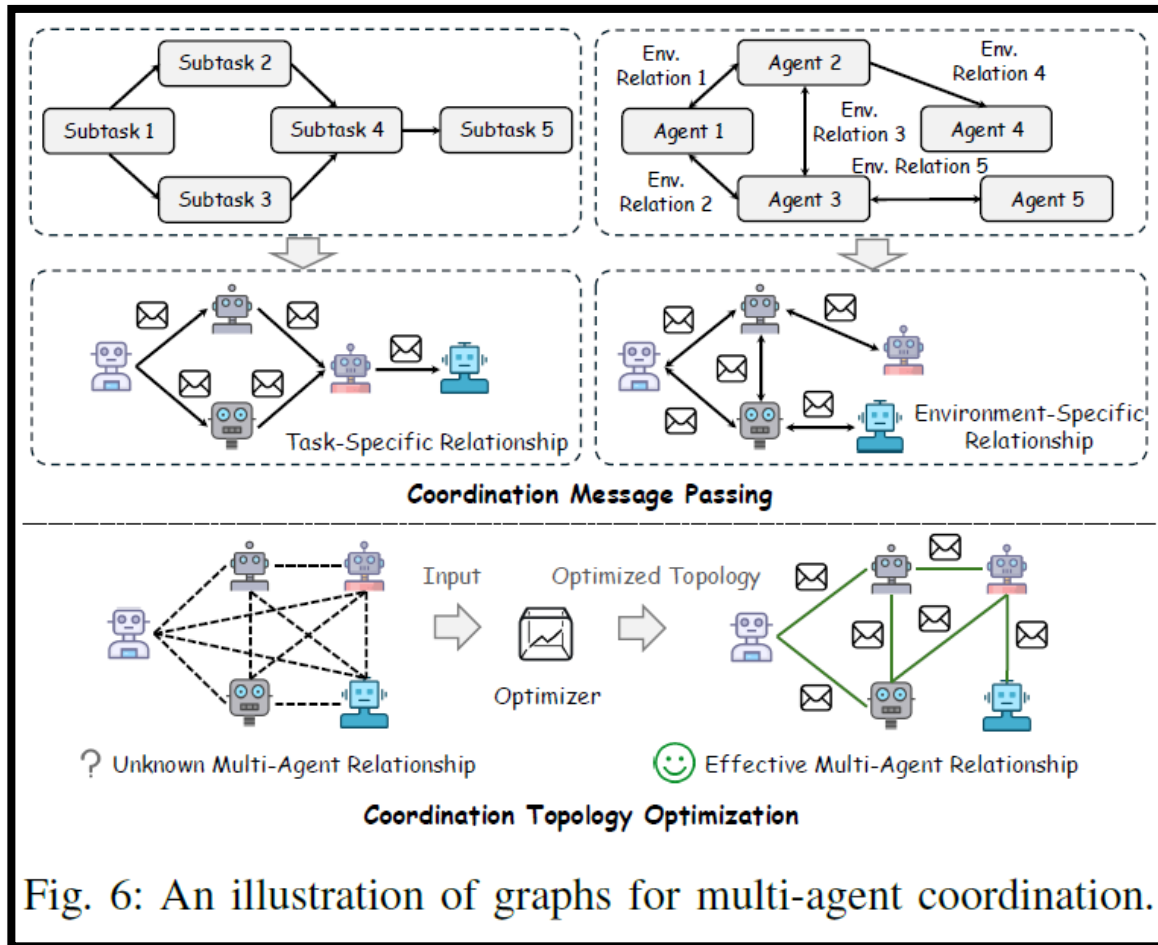
Graphs for Multi-agent Interaction

Multi-Agent Systems

- In multi-agent systems, graphs model communication and coordination.
- An **agent coordination graph (ACG)** represents agents as nodes and communications as edges. T
- These graphs can be dynamically learned to optimize coordination strategies and improve task performance

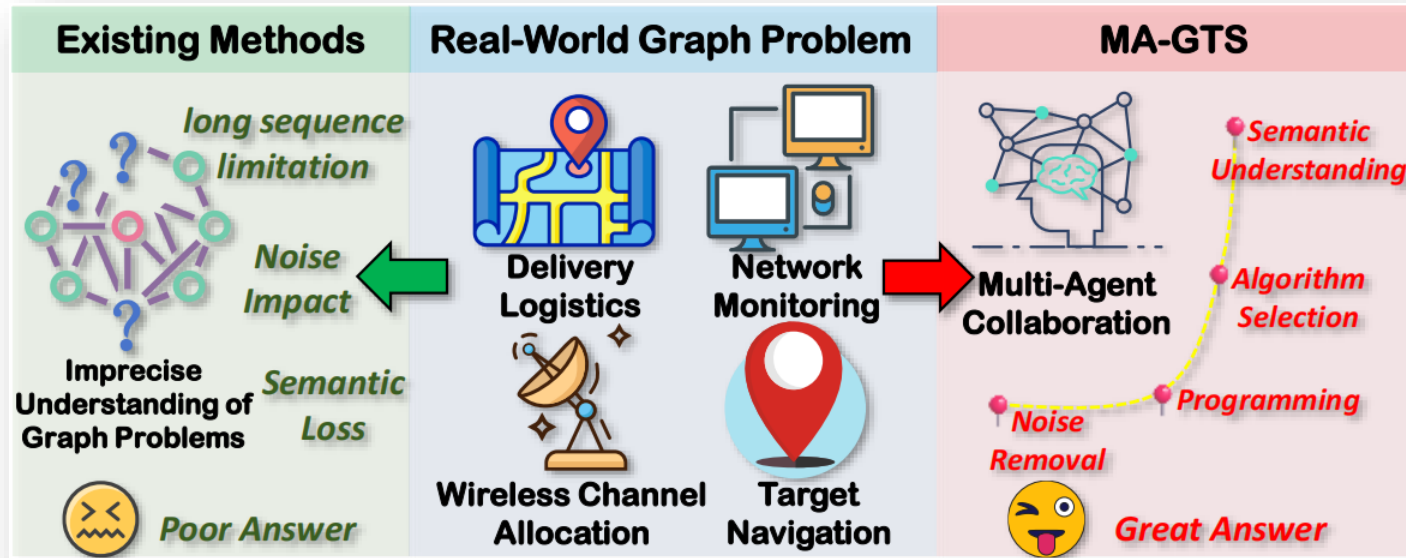


Graph for Multi-Agent Coordination



<https://arxiv.org/pdf/2506.18019>

MA-GTS: A Multi-Agent Framework for Graph Problems (EMNLP 2025)

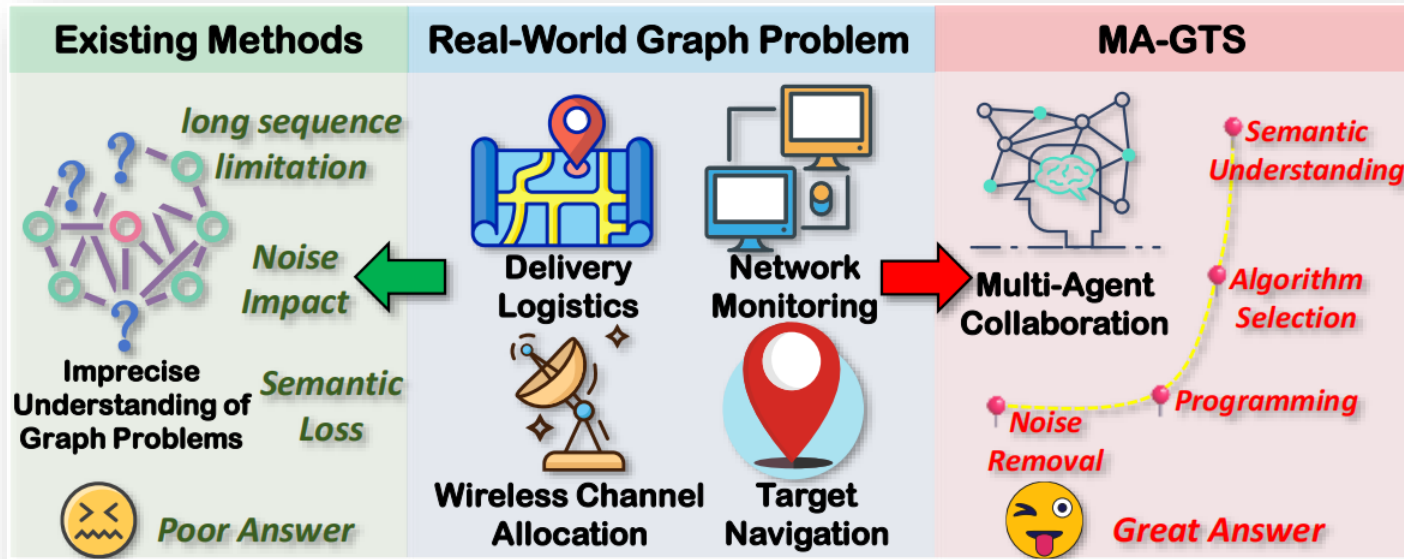


- **Problem:** Real-world graph problems (logistics, networks, traffic) are complex, noisy, and irregular
- **Limitations of LLMs:** limited accuracy, input length constraints, suboptimal algorithm selection

[1] <https://aclanthology.org/2025.emnlp-main.973.pdf>



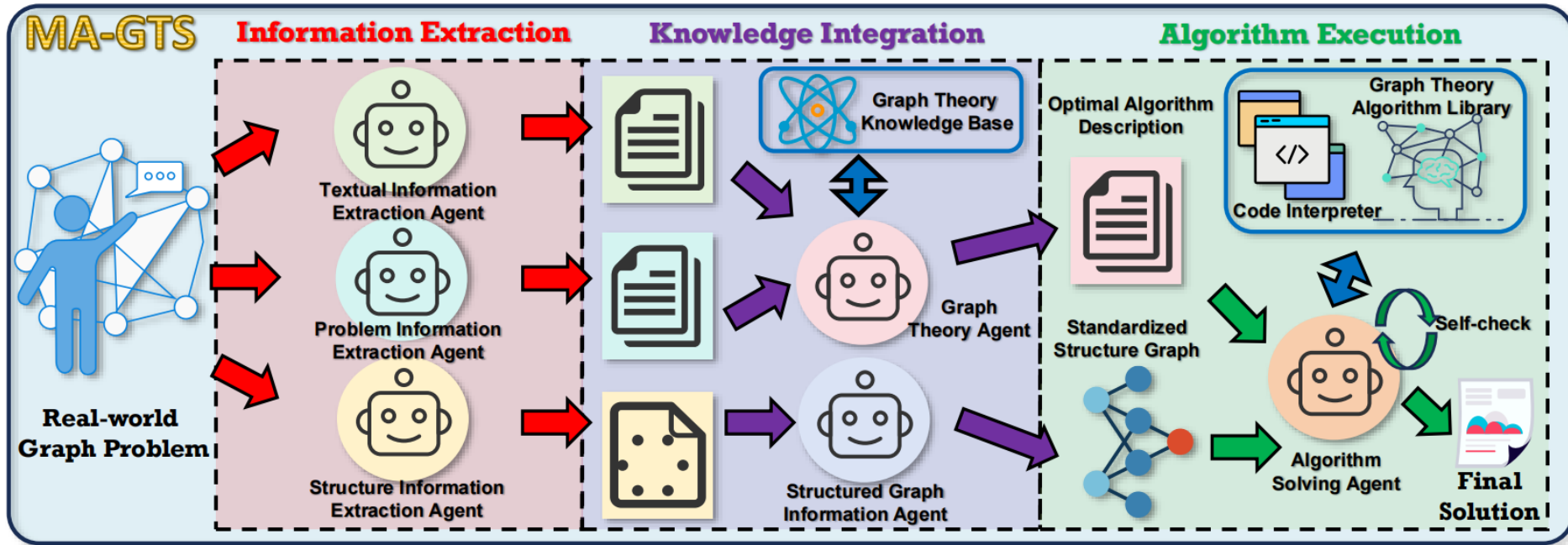
MA-GTS



- **Solution:** MA-GTS decomposes complex problems via agent collaboration



MA-GTS



- **Pipeline:** Text-based graph → structured representation → dynamic algorithm selection



Agent Interaction with Graph DB

Agent-driven Hypothesis Exploration

- data analysts will orchestrate swarms of agents to probe diverse hypotheses across complex datasets.

Combinatorial Query Space

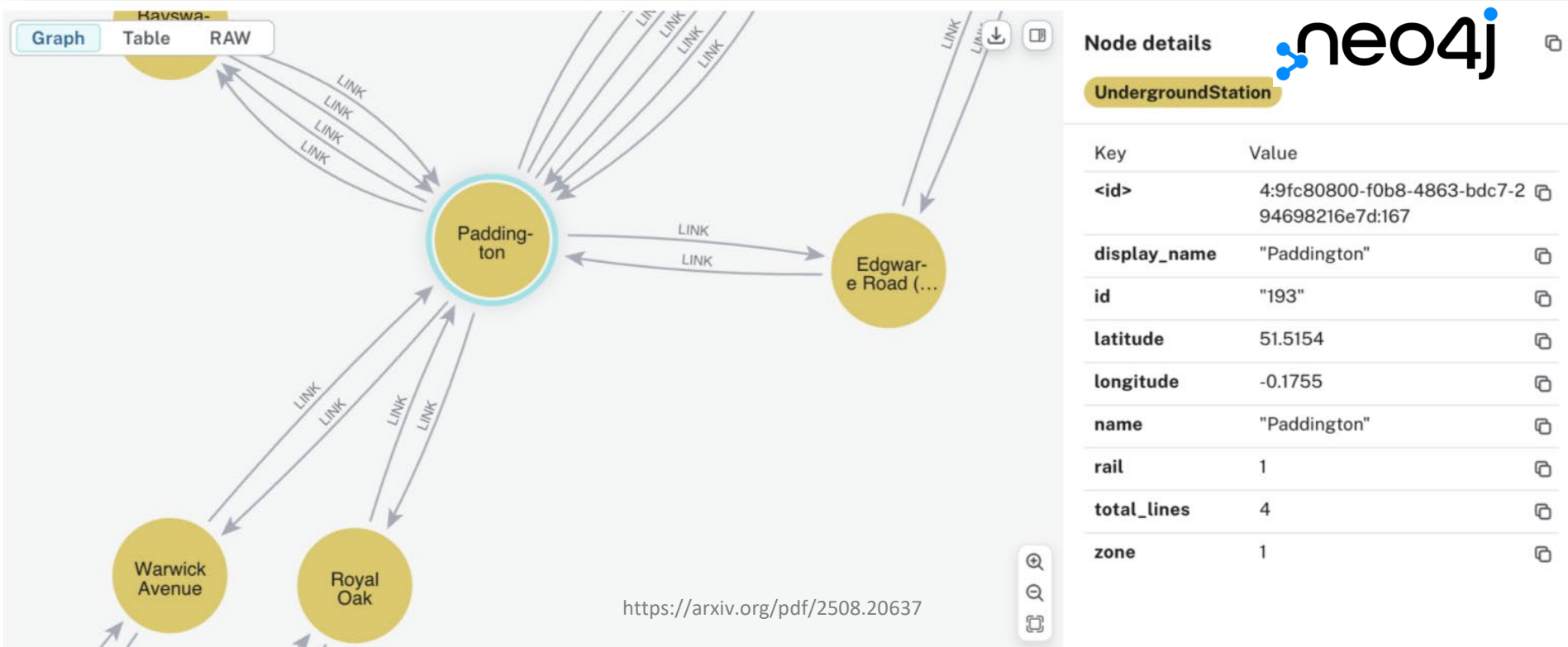
- Agents must navigate an exponential landscape of possible queries, including NP-hard graph problems.

Graph DBs for Agentic Workload

- Future systems must evolve to support adaptive, multi-agent query planning, optimization, and analytics at scale.



GDS Agent for Graph Algorithmic Reasoning



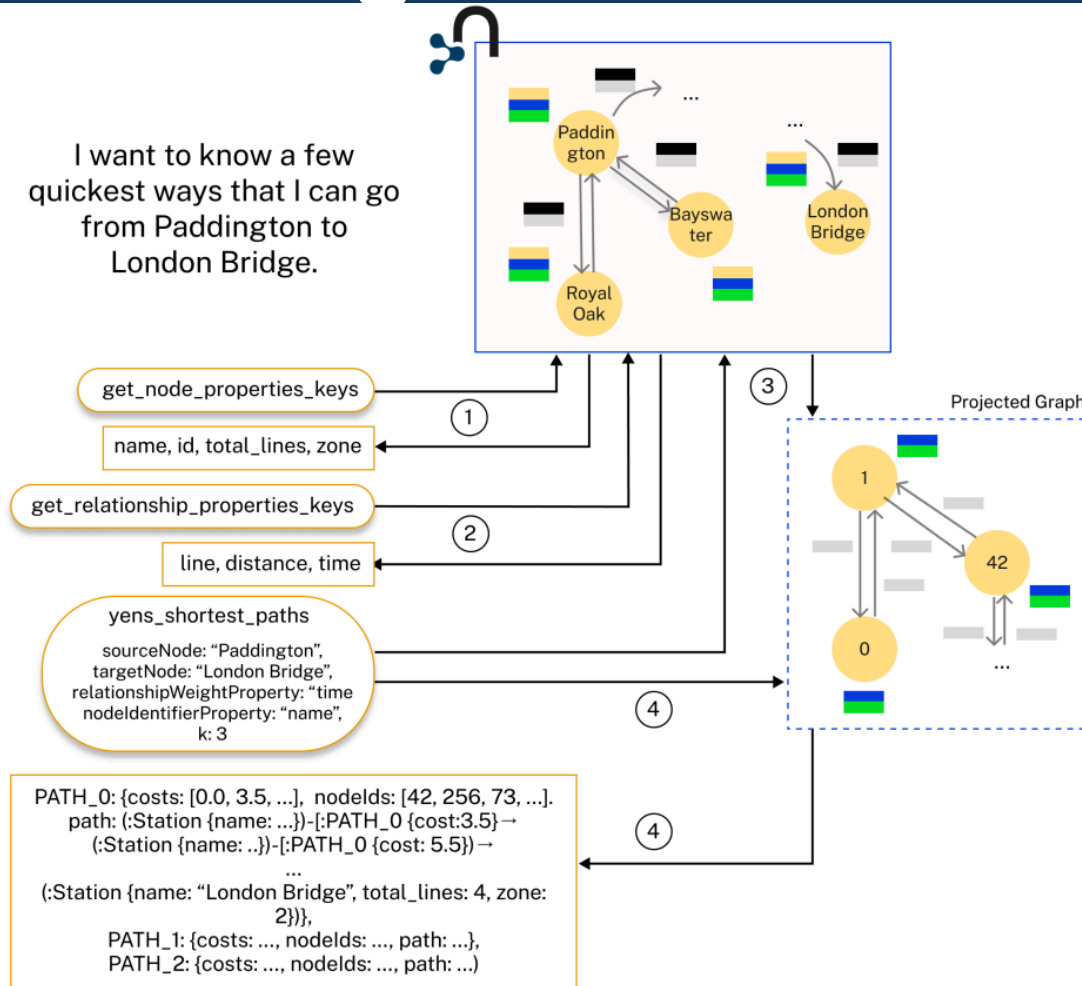
GDS Agent is a graph algorithm reasoning agent designed to address the challenges LLMs face when processing and reasoning over large-scale graph-structured data.



GDS Agent for Graph Algorithmic Reasoning



I want to know a few quickest ways that I can go from Paddington to London Bridge.



Procedure:

1. User Query
2. Tool Invocation
3. Algorithm Selection
4. Parameter
5. Cypher Projection & GDS Execution
6. Result & Final



Summary

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.



Next..

HUMAN-GRAPH AND AGENT-GRAPH INTERACTION: presentation overview & structural breakdown

PART 1: HUMAN-GRAPH INTERACTION (HGI)

1. INTERACTION WITH GRAPH SCHEMA



- Node types
- Node nodes
- Node labels

2. INTERACTION WITH GRAPH QUERIES



Intent mapping and builders

3. INTERACTION WITH GRAPH DATA



Visual exploration & data scrubbing



PART 2: AGENT-GRAPH INTERACTION (AGI)

1. INTERACTION WITH TASK PLANNING GRAPH



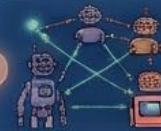
2. INTERACTION WITH TASK EXECUTION GRAPH



3. INTERACTION WITH MEMORY GRAPH



4. INTERACTION WITH AGENT COORDINATION GRAPH



PART 3: FUTURE RESEARCH DIRECTIONS & CONCLUSIONS



The Road Ahead

HGI for complex models, algorithms, and facets

- Primarily for traditional node-labeled and property graphs
- More complex graph models: Hypergraphs, Graph Neural Networks
- Higher-order structures: Summaries, hierarchical layouts

Closing the loop

- HGI: Humans directly interact with graphs
- AGI: Agent interacts with graph abstractions to perform tasks
- Limited understanding on human-agent interactions
- Leveraging HGI insights to make agent results more meaningful and usable.



The Road Ahead

AGI for Graph Data Management

- Many efforts in the data management community to exploit LLMs for various tasks
- Little attention to graph data management

Effective Abstractions for Interacting with Large Graphs

- Existing HGI & AGI largely target small- or medium-sized graphs
- Scalability is a challenge
- How does humans/agents interact with massive graphs?



Thank You!

